Panduan Developer

AWS Key Management Service



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Key Management Service: Panduan Developer

Copyright © 2026 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 hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

AWS Key Management Service 1
Mengapa menggunakan AWS KMS? 1
AWS KMS in Wilayah AWS 2
AWS KMS harga 2
AWS KMS perjanjian tingkat layanan 2
Mengakses AWS Key Management Service 3
Konsol Manajemen AWS 3
Izin yang diperlukan untuk menggunakan AWS KMS konsol 3
AWS Command Line Interface 4
AWS KMS API REST 4
AWS SDK 4
Bekerja dengan AWS SDK 5
AWS Encryption SDK 6
AWS KMS eventual consistency 6
TLS pasca-kuantum hibrida 7
Cara menggunakannya 7
Konfigurasikan TLS pasca-kuantum hibrida 9
Pelajari selengkapnya 10
Connect ke AWS KMS melalui titik akhir VPC 11
Buat titik akhir VPC untuk AWS KMS 12
Connect ke titik akhir VPC 13
Gunakan titik akhir VPC untuk mengontrol akses ke sumber daya AWS KMS 14
AWS KMS Permintaan logging yang menggunakan titik akhir VPC 18
Dual-stack titik akhir 19
Fitur tidak tersedia melalui IPv6 19
Konsep 20
Pengantar 20
Tujuan desain 21
AWS KMS keys 22
Kunci yang dikelola pelanggan 25
Kunci yang dikelola AWS 26
Kunci milik AWS 27
AWS KMS key hierarki 28
Pengidentifikasi kunci (KeyId) 30

Kunci asimetris 33
Kunci HMAC 35
Kunci ML-DSA 37
Multi-Region kunci 38
Materi kunci yang diimpor 49
Kunci KMS di toko kunci CloudHSM 56
Kunci KMS di toko kunci eksternal 59
AWS KMS esensi kriptografi 62
Entropi dan pembangkitan bilangan acak 62
Operasi kunci simetris (enkripsi saja) 62
Operasi kunci asimetris (enkripsi, penandatanganan digital dan verifikasi tanda tangan) 63
Fungsi derivasi kunci 63
AWS KMS penggunaan internal tanda tangan digital 64
Enkripsi amplop 64
Operasi kriptografi 66
Algoritme yang didukung 68
Algoritma kunci simetris 68
Algoritma kunci asimetris 68
Akses dan izin kunci KMS 71
Kebijakan utama KMS 71
Hibah kunci KMS 72
Kebijakan utama 73
Membuat kebijakan utama 73
Kebijakan kunci default 80
Melihat kebijakan utama 96
Mengubah kebijakan utama 99
Izin untuk layanan AWS 102
Kebijakan IAM 103
Mengizinkan beberapa prinsipal IAM untuk mengakses kunci KMS 105
Praktik terbaik untuk kebijakan IAM 106
Menentukan kunci KMS dalam pernyataan kebijakan IAM 108
Contoh 112
Kebijakan kontrol sumber daya 118
Izin 121
Konsep hibah 123
Praktik terbaik 129

Mengontrol akses ke pemberian izin 130
Membuat pemberian izin 131
Melihat pemberian izin 143
Menggunakan token izin 145
Menghentikan dan mencabut pemberian izin 146
Kunci syarat 148
AWS kunci kondisi global 148
AWS KMS kunci kondisi 152
AWS KMS kunci kondisi untuk platform yang dibuktikan 228
Least-privilege izin 234
Menerapkan izin yang paling tidak memiliki hak istimewa 235
Kontrol akses berbasis atribut (ABAC) 238
Kunci kondisi ABAC untuk AWS KMS 239
Tag atau alias? 242
Memecahkan masalah ABAC untuk AWS KMS 244
Kontrol akses berbasis peran (RBAC) 248
Cross-account akses 251
Langkah 1: Menambahkan pernyataan kebijakan kunci di akun lokal 253
Langkah 2: Menambahkan kebijakan IAM di akun eksternal 256
Mengizinkan penggunaan kunci KMS eksternal dengan Layanan AWS 258
Menggunakan kunci KMS di akun lain 258
Kontrol akses ke tombol Multi-wilayah 259
Basic otorisasi untuk kunci multi-Wilayah 260
Mengotorisasi administrator dan pengguna kunci multi-Wilayah 261
Menentukan akses 266
Memeriksa kebijakan kunci 266
Memeriksa kebijakan IAM 270
Memeriksa pemberian izin 272
Konteks enkripsi 274
Aturan konteks enkripsi 275
Konteks enkripsi dalam kebijakan 275
Konteks enkripsi dalam hibah 276
Konteks enkripsi pencatatan 277
Menyimpan konteks enkripsi 277
Menguji izin Anda 277
Apa itu DryRun? 278

Menentukan DryRun dengan API 279
Pemecahan Masalah AWS KMS izin 280
Contoh 1: Pengguna ditolak akses ke kunci KMS di Akun AWS 282
Contoh 2: Pengguna mengasumsikan peran dengan izin untuk menggunakan kunci KMS dalam yang berbeda Akun AWS 284
Glosarium 288
Autentikasi 288
Otorisasi 288
Mengautentikasi dengan identitas 288
Mengelola akses menggunakan kebijakan 290
AWS KMS sumber daya 292
Buat kunci KMS 293
Izin untuk membuat kunci KMS 295
Memilih jenis kunci KMS apa yang akan dibuat 296
Buat kunci KMS enkripsi simetris 298
Buat kunci KMS asimetris 304
Buat kunci HMAC KMS 310
Buat kunci utama Multi-wilayah 315
Buat kunci replika Multi-wilayah 320
Langkah 1: Pilih Wilayah replika 321
Langkah 2: Buat kunci replika 322
Buat kunci KMS dengan bahan kunci yang diimpor 328
Izin untuk mengimpor material kunci 329
Persyaratan untuk bahan kunci yang diimpor 330
Langkah 1: Buat materi AWS KMS key tanpa kunci 333
Langkah 2: Unduh kunci publik pembungkus dan token impor 336
Langkah 3: Enkripsi material kunci 345
Langkah 4: Impor material kunci 354
Buat kunci KMS di toko AWS CloudHSM kunci 362
Buat kunci KMS baru di toko kunci CloudHSM Anda 363
Buat kunci KMS di toko kunci eksternal 369
Persyaratan untuk kunci KMS di toko kunci eksternal 371
Buat kunci KMS baru di toko kunci eksternal Anda 372
Identifikasi dan lihat kunci 380
Temukan ID kunci dan kunci ARN 380
Akses dan daftar rincian kunci KMS 382

Identifikasi berbagai jenis kunci ... 392
Identifikasi kunci KMS asimetris ... 392
Identifikasi kunci HMAC KMS ... 393
Identifikasi kunci KMS Multi-wilayah ... 394
Identifikasi kunci KMS dengan bahan kunci impor ... 395
Identifikasi kunci KMS di AWS CloudHSM toko-toko utama ... 395
Identifikasi kunci KMS di toko kunci eksternal ... 396
Sesuaikan tampilan konsol Anda ... 397
Urutkan dan filter kunci KMS Anda ... 398
Sesuaikan tabel kunci KMS Anda ... 400
Temukan kunci KMS dan bahan kunci di toko AWS CloudHSM kunci ... 402
Temukan kunci KMS di toko AWS CloudHSM kunci ... 403
Temukan semua kunci untuk toko AWS CloudHSM kunci ... 404
Temukan kunci KMS untuk kunci AWS CloudHSM ... 406
Temukan AWS CloudHSM kunci untuk kunci KMS ... 411
Aktifkan dan nonaktifkan kunci ... 414
Putar tombol ... 417
Mengapa harus merotasi kunci KMS? ... 419
Cara kerja rotasi kunci ... 420
Aktifkan rotasi tombol otomatis ... 425
Nonaktifkan rotasi tombol otomatis ... 427
Lakukan rotasi kunci sesuai permintaan ... 429
Memulai rotasi kunci sesuai permintaan (konsol) ... 430
Memulai rotasi kunci sesuai permintaan (AWS KMS API) ... 431
Daftar rotasi dan bahan utama ... 432
Daftar rotasi dan bahan utama (konsol) ... 433
Daftar rotasi dan bahan utama (AWS KMS API) ... 434
Putar tombol secara manual ... 435
Ubah kunci utama dalam satu set kunci Multi-wilayah ... 437
Perbarui Wilayah utama ... 439
Hapus kunci ... 442
Tentang masa tunggu ... 443
Pertimbangan khusus ... 444
Kontrol akses ke penghapusan kunci ... 447
Izinkan administrator kunci untuk menjadwalkan dan membatalkan penghapusan kunci ... 447
Jadwalkan penghapusan kunci ... 450

Batalkan penghapusan kunci ... 452
Membuat alarm ... 453
Hapus materi kunci yang diimpor ... 455
Hasilkan kunci data ... 458
Membuat kunci data ... 458
Bagaimana operasi kriptografi dengan kunci data bekerja ... 459
Mengenkripsi data dengan kunci data ... 459
Mendekripsi data dengan kunci data ... 460
Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data ... 461
Hasilkan pasangan kunci data ... 463
Membuat pasangan kunci data ... 463
Bagaimana operasi kriptografi dengan pasangan kunci data bekerja ... 464
Mengenkripsi data dengan pasangan kunci data ... 465
Mendekripsi data dengan pasangan kunci data ... 465
Menandatangani pesan dengan pasangan kunci data ... 466
Memverifikasi tanda tangan dengan pasangan kunci data ... 467
Dapatkan rahasia bersama dengan pasangan kunci data ... 468
Lakukan operasi offline dengan kunci publik ... 469
Pertimbangan khusus untuk mengunduh kunci publik ... 470
Unduh kunci publik ... 471
Contoh operasi offline ... 473
Memperoleh rahasia bersama secara offline ... 473
Verifikasi offline dengan pasangan ML-DSA kunci ... 474
Verifikasi offline dengan pasangan kunci SM2 (hanya Wilayah China) ... 477
Tombol monitor ... 483
Alat-alat pemantauan ... 484
Alat otomatis ... 484
Alat manual ... 485
Logging dengan AWS CloudTrail ... 486
Temuan AWS KMS entri log di CloudTrail ... 487
Tidak termasuk AWS KMS peristiwa dari jejak ... 488
Contoh dari AWS KMS entri log ... 489
Tombol monitor dengan CloudWatch ... 577
AWS KMS metrik dan dimensi ... 577
Buat CloudWatch alarm untuk kedaluwarsa materi kunci yang diimpor ... 586
Buat CloudWatch alarm untuk penyimpanan kunci eksternal ... 588

Menentukan penggunaan kunci KMS sebelumnya 592
Periksa izin kunci KMS untuk menentukan ruang lingkup penggunaan potensial 593
Periksa operasi kriptografi terakhir yang dilakukan dengan kunci KMS 593
Periksa AWS CloudTrail log untuk memeriksa penggunaan masa lalu 596
Tombol monitor dengan Amazon EventBridge 599
Rotasi CMK KMS 600
KMS Mengimpor Tanggal Kedaluwarsa Bahan Kunci 601
Penghapusan CMK KMS 601
Alias 603
Cara kerja alias 603
Mengontrol akses ke alias 607
km: CreateAlias 607
km: ListAliases 608
km: UpdateAlias 609
km: DeleteAlias 610
Membatasi izin alias 612
Buat alias 613
Temukan nama alias dan alias ARN 615
Perbarui alias 620
Hapus alias 621
Gunakan alias untuk mengontrol akses ke tombol KMS 622
km: RequestAlias 624
km: ResourceAliases 625
Pelajari cara menggunakan alias dalam aplikasi 627
Temukan alias di AWS CloudTrail log 628
Tag 631
Pengontrolan akses ke tanda 632
Izin tanda dalam kebijakan 633
Membatasi izin tanda 635
Menambahkan tag 637
Tambahkan tag saat membuat kunci KMS 637
Tambahkan tag ke kunci KMS yang ada 639
Edit tag 640
Hapus tag 641
Lihat tanda 642
Gunakan tag untuk mengontrol akses ke tombol KMS 644

Toko-toko utama 649
Memilih toko kunci yang tepat 649
AWS KMS toko kunci standar 650
Fitur 651
Operasi 651
Model tanggung jawab 651
AWS KMS toko kunci standar dengan bahan kunci impor 651
Fitur 652
Operasi 652
Model tanggung jawab 652
AWS KMS toko kunci kustom 652
Fitur yang didukung 653
AWS CloudHSM toko kunci 653
Toko kunci eksternal 654
AWS CloudHSM toko-toko utama 655
AWS CloudHSM konsep toko utama 658
Kontrol akses ke toko AWS CloudHSM kunci Anda 661
Buat toko AWS CloudHSM kunci 662
Lihat toko AWS CloudHSM kunci 670
Edit pengaturan penyimpanan AWS CloudHSM kunci 673
Connect toko AWS CloudHSM kunci 676
Putuskan sambungan toko AWS CloudHSM kunci 680
Hapus toko AWS CloudHSM kunci 683
Memecahkan masalah penyimpanan kunci kustom 686
Toko kunci eksternal 702
Konsep toko kunci eksternal 706
Cara kerja toko kunci eksternal 715
Kontrol akses ke toko kunci eksternal Anda 717
Pilih opsi konektivitas proxy 723
Buat penyimpanan kunci eksternal 736
Edit properti penyimpanan kunci eksternal 750
Lihat toko kunci eksternal 757
Pantau toko kunci eksternal 763
Connect dan lepaskan penyimpanan kunci eksternal 776
Hapus penyimpanan kunci eksternal 786
Memecahkan masalah toko kunci eksternal 788

Keamanan 815
Perlindungan data 816
Melindungi bahan utama 816
Enkripsi data 817
Privasi antar jaringan 819
Manajemen identitas dan akses 820
AWS kebijakan yang dikelola 820
Peran terkait layanan 824
Pencatatan log dan pemantauan 830
Validasi kepatuhan 831
Dokumen kepatuhan dan keamanan 832
Pelajari selengkapnya 832
Ketahanan 833
Isolasi regional 833
Desain multi-penyewa 834
Praktik terbaik ketahanan di AWS KMS 834
Keamanan infrastruktur 835
Isolasi pada Host Fisik 836
Kuota 838
Kuota sumber daya 838
AWS KMS keys: 100.000 839
Alias per kunci KMS: 50 839
Hibah per kunci KMS: 50.000 840
Kuota sumber daya penyimpanan kunci kustom: 10 840
On-demand rotasi: 25 840
Kuota permintaan 841
Minta kuota untuk masing-masing AWS KMS Operasi API 842
Menerapkan kuota permintaan 849
Kuota bersama untuk operasi kriptografis 850
Permintaan API yang dibuat atas nama Anda 851
Cross-account permintaan 851
Kuota permintaan penyimpanan kunci khusus 852
Melakukan throttling permintaan 853
Contoh kode 856
Hal-hal mendasar 857
Halo AWS KMS 858

Pelajari dasar-dasarnya 862
Tindakan 936
Skenario 1104
Bekerja dengan enkripsi tabel 1105
Dukungan pengesahan kriptografi di AWS KMS 1108
Cara membuat panggilan yang dibuktikan ke AWS KMS 1110
Memantau permintaan yang dibuktikan 1110
Memantau permintaan untuk kantong Nitro 1111
Memantau permintaan untuk NitroTPM 1116
Layanan enkripsi AWS 1121
Amazon Elastic Block Store (Amazon EBS) 1122
Enkripsi EBS Amazon 1122
Menggunakan kunci KMS dan kunci data 1123
Konteks enkripsi Amazon EBS 1123
Mendeteksi kegagalan Amazon EBS 1124
Penggunaan AWS CloudFormation untuk membuat volume Amazon EBS terenkripsi 1125
Amazon EMR 1125
Mengenkripsi data pada sistem file EMR (EMRFS) 1126
Mengenkripsi data pada volume penyimpanan simpul klaster 1129
Konteks enkripsi 1130
Amazon Redshift 1131
Enkripsi Amazon Redshift 1131
Konteks enkripsi 1132
Referensi 1133
Referensi status kunci 1134
Status kunci dan tipe kunci KMS 1134
Tabel status kunci 1135
Referensi tipe kunci 1144
Tabel tipe kunci 1144
Tabel fitur khusus 1151
Referensi spesifikasi kunci 1160
Spesifikasi kunci SYMMETRIC_DEFAULT 1162
Spesifikasi kunci RSA 1163
Spesifikasi kunci kurva elips 1167
Spesifikasi utama untuk kunci HMAC KMS 1170
Spesifikasi kunci ML-DSA 1171

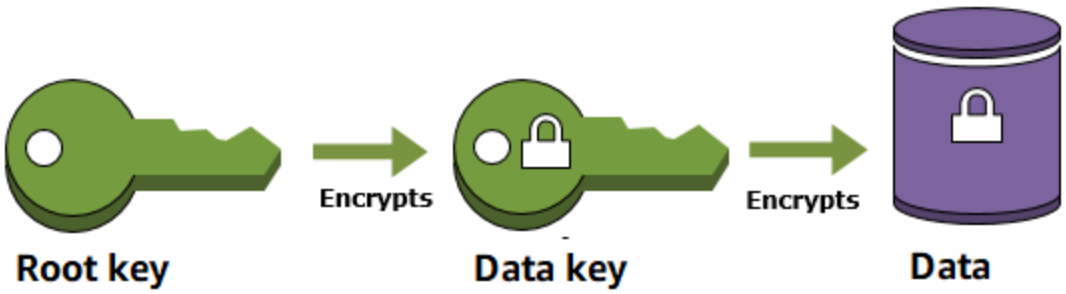
SM2 spesifikasi kunci (hanya Wilayah Tiongkok) 1171
Referensi izin 1172
Deskripsi kolom 1222
AWS KMS operasi internal 1224
Domain dan status domain 1225
Keamanan komunikasi internal 1229
Proses replikasi untuk kunci Multi-wilayah 1232
Perlindungan daya tahan 1233
Riwayat dokumen 1234
Pembaruan terkini 1234
Pembaruan lebih awal 1242
..... mccxlvii

AWS Key Management Service

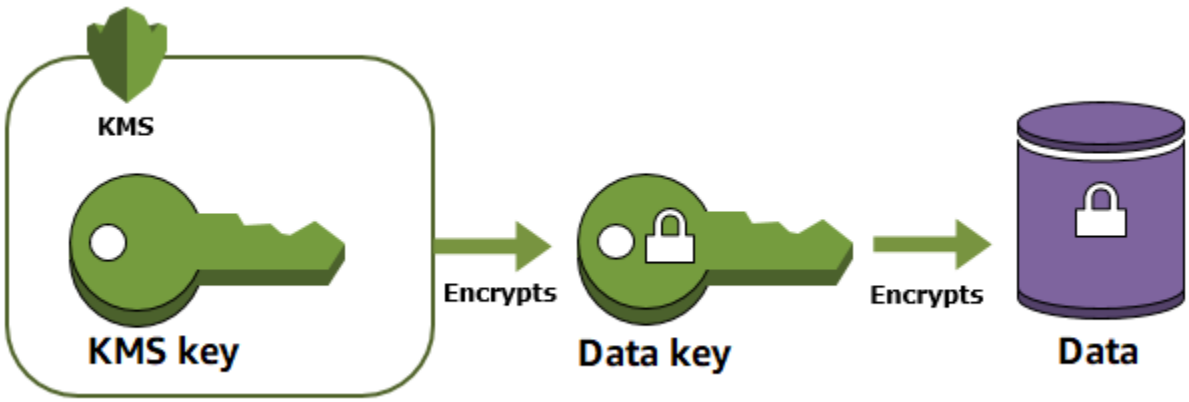
AWS Key Management Service (AWS KMS) adalah layanan AWS terkelola yang memudahkan Anda membuat dan mengontrol kunci yang digunakan untuk mengenkripsi dan menandatangani data Anda. AWS KMS keys Yang Anda buat dilindungi oleh FIPS 140-3 Security Level 3 divalidasi modul keamanan perangkat keras (HSM). AWS KMS Mereka tidak pernah meninggalkan yang AWS KMS tidak terenkripsi. Untuk menggunakan atau mengelola kunci KMS Anda, Anda berinteraksi dengan AWS KMS.

Mengapa menggunakan AWS KMS?

Saat Anda mengenkripsi data, Anda perlu melindungi kunci enkripsi Anda. Jika Anda mengenkripsi kunci Anda, Anda perlu melindungi kunci enkripsi. Akhirnya, Anda harus melindungi kunci enkripsi tingkat tertinggi (dikenal sebagai kunci root) dalam hierarki yang melindungi data Anda. Di situlah AWS KMS datang.



AWS KMS melindungi kunci root Anda. Kunci KMS dibuat, dikelola, digunakan, dan dihapus seluruhnya di dalam AWS KMS. Mereka tidak pernah meninggalkan layanan tanpa terenkripsi. Untuk menggunakan atau mengelola kunci KMS Anda, Anda menelepon AWS KMS.



Selain itu, Anda dapat membuat dan mengelola kebijakan utama AWS KMS, memastikan bahwa hanya pengguna tepercaya yang memiliki akses ke kunci KMS.

AWS KMS in Wilayah AWS

Yang didukung tercantum Wilayah AWS dalam AWS Key Management Service Endpoints dan Quotas. AWS KMS Jika AWS KMS fitur tidak didukung dalam Wilayah AWS yang AWS KMS mendukung, perbedaan regional dijelaskan dalam topik tentang fitur tersebut.

AWS KMS harga

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

AWS KMS perjanjian tingkat layanan

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

Mengakses AWS Key Management Service

Anda dapat bekerja AWS KMS dengan cara berikut:

Topik

- Konsol Manajemen AWS
- AWS Command Line Interface
- AWS KMS API REST
- AWS SDK
- Menggunakan layanan ini dengan AWS SDK
- AWS Encryption SDK
- AWS KMS eventual consistency
- Menggunakan TLS pasca-kuantum hibrida dengan AWS KMS
- Connect ke AWS KMS melalui titik akhir VPC
- Dual-stack dukungan titik akhir

Konsol Manajemen AWS

Konsol adalah antarmuka pengguna berbasis web untuk mengelola AWS KMS dan AWS sumber daya. Jika Anda telah mendaftar Akun AWS, Anda dapat mengakses AWS KMS konsol dengan masuk ke Konsol Manajemen AWS dan memilih AWS KMS dari halaman Konsol Manajemen AWS beranda.

Izin yang diperlukan untuk menggunakan AWS KMS konsol

Untuk bekerja dengan AWS KMS konsol, pengguna harus memiliki seperangkat izin minimum yang memungkinkan mereka 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 dari izin minimum yang diperlukan, konsol AWS KMS tersebut tidak akan berfungsi sebagaimana mestinya 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 mengizinkan pengguna bekerja dengan AWS KMS konsol untuk membuat dan mengelola kunci KMS, lampirkan kebijakan ter AWSKeyManagementServicePowerUser kelola ke pengguna, seperti yang dijelaskan dalamAWS kebijakan yang dikelola untuk AWS Key Management Service.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang bekerja dengan AWS KMS API melalui AWS SDK, AWS Command Line Interface, atau Alat AWS untuk PowerShell. Namun, Anda harus memberikan izin kepada pengguna ini untuk menggunakan API. Untuk informasi selengkapnya, lihat Referensi izin.

AWS Command Line Interface

Anda dapat menggunakan AWS CLI alat untuk mengeluarkan perintah atau membuat skrip di baris perintah sistem Anda untuk melakukan AWS (termasuk AWS KMS) tugas.

Untuk informasi selengkapnya tentang menggunakan AWS KMS melalui AWS CLI, lihat Referensi AWS CLI Perintah

AWS KMS API REST

Arsitektur AWS KMS dirancang untuk menjadi bahasa pemrograman netral. Ini REST API adalah antarmuka HTTP untuk AWS KMS. DenganREST API, Anda menggunakan permintaan HTTP standar untuk membuat, mengambil, dan menghapus kunci.

Untuk informasi lebih lanjut tentang penggunaan AWS KMS REST API, lihat Refer AWS Key Management Service ensi API

AWS SDK

AWS menyediakan SDK (perangkat pengembangan perangkat lunak) yang terdiri dari pustaka dan kode sampel untuk bahasa pemrograman umum dan platform (Java JavaScript,, C, Python, dan sebagainya). AWS SDK menyediakan cara mudah untuk membuat akses terprogram ke AWS KMS dan. AWS AWS KMS adalah sebuah REST layanan. Anda dapat mengirim permintaan AWS KMS menggunakan pustaka AWS SDK, yang membungkus tugas pemrograman yang mendasarinya AWS KMS REST API dan menyederhanakan. Untuk informasi tentang AWS SDKs, termasuk cara mengunduh dan menginstalnya, lihat Alat untuk Membangun AWS.

Ini Contoh kode untuk AWS KMS menggunakan AWS SDKs memberikan titik awal yang baik untuk AWS KMS digunakan melalui AWS SDKs.

Menggunakan layanan ini dengan AWS SDK

AWS perangkat pengembangan perangkat lunak (SDK) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK untuk C++	AWS SDK untuk C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK untuk Go	AWS SDK untuk Go contoh kode
AWS SDK untuk Java	AWS SDK untuk Java contoh kode
AWS SDK untuk JavaScript	AWS SDK untuk JavaScript contoh kode
AWS SDK untuk Kotlin	AWS SDK untuk Kotlin contoh kode
AWS SDK untuk .NET	AWS SDK untuk .NET contoh kode
AWS SDK untuk PHP	AWS SDK untuk PHP contoh kode
Alat AWS untuk PowerShell	Alat AWS untuk PowerShell contoh kode
AWS SDK untuk Python v4	AWS SDK untuk contoh kode Python v4
AWS SDK untuk Python (Boto3)	AWS SDK untuk Python (Boto3) contoh kode
AWS SDK untuk Ruby	AWS SDK untuk Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Ketersediaan contoh

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

AWS Encryption SDK

Ini AWS Encryption SDK adalah alat untuk menerapkan enkripsi sisi klien dalam aplikasi Anda. Ini tidak menyediakan akses penuh ke KMS, tetapi malah terintegrasi dengan AWS KMS, atau dapat digunakan sebagai SDK yang berdiri sendiri tanpa merujuk kunci KMS. Perpustakaan tersedia untuk Java, C JavaScript, Python, dan bahasa pemrograman lainnya.

Lihat informasi selengkapnya di Panduan Developer AWS Encryption SDK.

AWS KMS key kebijakan dan kebijakan IAM

AWS KMS eventual consistency

AWS KMS API mengikuti model konsistensi akhirnya karena sifat sistem yang terdistribusi. Akibatnya, perubahan pada AWS KMS sumber daya mungkin tidak segera terlihat oleh perintah berikutnya yang Anda jalankan.

Saat Anda melakukan panggilan AWS KMS API, 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 mendapatkan kesalahan tak terduga, seperti a `NotFoundException` atau an`InvalidStateException`, selama waktu ini. Misalnya, AWS KMS mungkin mengembalikan `NotFoundException` jika Anda menelepon `GetParametersForImport` segera setelah meneleponCreateKey.

Sebaiknya Anda mengonfigurasi strategi coba ulang pada AWS KMS klien Anda untuk secara otomatis mencoba kembali operasi setelah periode tunggu yang singkat. Untuk informasi selengkapnya, lihat Per ilaku Coba Ulang di Panduan Refer AWS ensi SDK dan Alat.

Untuk panggilan API terkait hibah, Anda dapat menggunakan token hibah untuk menghindari potensi penundaan dan segera menggunakan izin dalam hibah. Untuk informasi lebih lanjut, lihat Kon sistensi akhir (untuk hibah).

Menggunakan TLS pasca-kuantum hibrida dengan AWS KMS

Data yang Anda kirim ke AWS Key Management Service (AWS KMS) dilindungi saat transit oleh enkripsi yang disediakan oleh koneksi Transport Layer Security (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, rangkaian sandi klasik yang digunakan dalam mekanisme pertukaran kunci TLS akan rentan terhadap serangan ini.

Pada Agustus 2024, NIST menyelesaikan standar kriptografi pasca-kuantum pertamanya. Ini termasuk Module-Lattice-Based Key-Encapsulation Mekanisme (ML-KEM) yang ditentukan dalam FIPS 203 di situs web NIST. AWS KMS sekarang mendukung opsi pertukaran kunci pasca-kuantum hibrida untuk TLS yang menggabungkan Elliptic Curve Diffie-Hellman (ECDH) dengan. ML-KEM Pertukaran kunci hibrida ini mempertahankan kekuatan suite sandi klasik saat ini sambil menambahkan ketahanan terhadap musuh yang mampu kuantum di masa depan. Untuk informasi selengkapnya, lihat TLS ML-KEM pasca-kuantum sekarang didukung di AWS KMS, ACM, dan Manajer Rahasia di Blog Keamanan. AWS

Jika aplikasi Anda bergantung pada kerahasiaan jangka panjang data yang diteruskan melalui koneksi TLS, Anda harus menggunakan kriptografi pasca-kuantum. Jika tidak, musuh mungkin dapat menangkap data yang dienkripsi dengan suite sandi klasik hari ini dan mendekripsinya begitu komputer kuantum tersedia. Strategi ini dikenal sebagai panen sekarang, dekripsi nanti.

Menggunakan TLS pasca-kuantum hibrida dengan AWS KMS

Anda dapat menggunakan TLS pasca-kuantum hibrida untuk semua panggilan API Anda. AWS KMS Saat menyiapkan klien HTTP Anda, perhatikan informasi berikut.

AWS KMS Titik akhir

AWS KMS mendukung TLS pasca-kuantum hibrida di semua Wilayah AWS tempat tersedia, di semua titik akhir termasuk titik akhir yang divalidasi FIPS 140-3. Untuk daftar titik akhir AWS KMS untuk masing-masing Wilayah AWS, lihat titik akhir AWS Key Management Service dan kuota di. Referensi Umum Amazon Web Services Untuk informasi tentang titik akhir FIPS, lihat titik akhir FIPS di. Referensi Umum Amazon Web Services

Sistem yang Didukung

Banyak AWS SDK mendukung TLS pasca-kuantum hibrida, seperti yang dijelaskan dalam Meng aktifkan TLS pasca-kuantum hibrida di Panduan Referensi AWS SDK dan Alat. Untuk contoh Java, lihatKonfigurasikan TLS pasca-kuantum hibrida.

Memverifikasi Post-Quantum TLS Hibrida

Anda dapat mengonfirmasi bahwa panggilan AWS KMS API tertentu menggunakan TLS pasca-kuantum hibrida dengan memeriksa entri lognya CloudTrail . Temukan `tlsDetails` bagian di entri log untuk panggilan tersebut. Bid `keyExchange` ang tersebut menamai algoritma pertukaran kunci yang digunakan. Untuk koneksi pasca-kuantum hibrida, bidang ini menunjukkan algoritma hibrida seperti`X25519MLKEM768`. Untuk contoh entri log, lihatDekripsi dengan kunci enkripsi simetris standar melalui koneksi TLS pasca-kuantum.

> **Note**
>
> Bid `tlsDetails` ang tidak ada di entri log saat AWS layanan memanggil AWS KMS atas nama Anda. Untuk detail tentang bidang dalam entri CloudTrail log dan kapan mereka ada, lihat konten CloudTrail rek aman di Panduan AWS CloudTrail Pengguna.

Performa

Pertukaran kunci hibrida sedikit meningkatkan ukuran dan waktu pemrosesan beberapa pesan jabat tangan TLS, tetapi dampak kinerja keseluruhan tidak terlihat untuk sebagian besar beban kerja. Karena pesan ini lebih besar, kami sarankan Anda menguji panggilan AWS KMS API dari lokasi yang berbeda di jaringan Anda.

Bergantung pada jalur jaringan yang diambil permintaan Anda, host perantara lama, proxy, atau firewall dengan inspeksi paket mendalam (DPI) mungkin memblokir permintaan. Ini dapat dihasilkan dari grup pertukaran kunci baru di ClientHello bagian jabat tangan TLS, atau dari pesan pertukaran kunci yang lebih besar. Jika Anda mengalami kesulitan menyelesaikan masalah ini, bekerjasama dengan tim keamanan atau administrator TI untuk membuka blokir grup pertukaran kunci TLS yang baru.

Enkripsi Saat Data Berpindah

Suite cipher hybrid hanya digunakan untuk enkripsi dalam transit. Mereka melindungi data Anda saat bepergian dari klien Anda ke AWS KMS titik akhir. AWS KMS tidak menggunakan suite sandi 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 AES-GCM kunci 256-bit mengurangi keamanan efektif kunci menjadi 128 bit. Untuk informasi lebih lanjut, lihat makalah Kriptografi dan Keamanan Aman Quantum di situs web ETSI. Tingkat keamanan ini cukup untuk membuat serangan brute force pada AWS KMS ciphertext tidak mungkin dilakukan.

Konfigurasikan TLS pasca-kuantum hibrida

Dalam prosedur ini, tambahkan dependensi Maven untuk Com AWS mon Runtime HTTP Client. Selanjutnya, konfigurasikan klien HTTP yang lebih menyukai TLS pasca-kuantum. Kemudian, buat AWS KMS klien yang menggunakan klien HTTP.

Untuk melihat contoh kerja lengkap mengonfigurasi dan menggunakan TLS pasca-kuantum hibrida dengan AWS KMS, lihat aws-kms-pq-tls-example repositori aktif. GitHub

1. Tambahkan klien Com AWS mon Runtime ke dependensi Maven Anda. Sebaiknya gunakan versi terbaru yang tersedia.

   Misalnya, pernyataan ini menambahkan versi klien `2.30.22` Com AWS mon Runtime ke dependensi Maven Anda. Gunakan versi `2.30.22` atau yang lebih baru untuk mengaktifkan ML-KEM.

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

2. Untuk mengaktifkan suite sandi pasca-kuantum hibrida, tambahkan AWS SDK for Java 2.x ke proyek Anda dan inisialisasi. Kemudian aktifkan suite cipher 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 memilih rangkaian cipher pasca-kuantum hibrida yang direkomendasikan, ECDH dengan. ML-KEM Kemudian menggunakan klien HTTP yang dikonfigurasi untuk membangun instance klien AWS KMS asinkron,. `KmsAsyncClient` Setelah kode ini selesai, semua permintaan AWS KMS API 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.

   Saat Anda memanggil operasi AWS KMS API pada AWS KMS klien yang dikonfigurasi, panggilan Anda ditransmisikan ke AWS KMS titik akhir menggunakan TLS pasca-kuantum hibrida. Untuk menguji konfigurasi Anda, pang AWS KMS gil API, seperti`ListKeys`.

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

   Untuk mengonfirmasi bahwa panggilan Anda menggunakan TLS pasca-kuantum hibrida, periksa entri CloudTrail lognya seperti yang dijelaskan di. Memverifikasi Post-Quantum TLS Hibrida

Pelajari lebih lanjut tentang TLS pasca-kuantum di AWS KMS

Untuk informasi lebih lanjut tentang menggunakan TLS pasca-kuantum hibrida di AWS KMS, lihat sumber daya berikut.

- Untuk mempelajari tentang kriptografi pasca-kuantum di AWS, termasuk tautan ke posting blog dan makalah penelitian, lihat Post-Quantum Kriptografi.
- AWS KMS menggunakans2n-tls, implementasi TLS open source dari AWS, untuk sisi server dari koneksi TLS pasca-kuantum hibridanya. Untuk informasi tentangs2n-tls, lihat

Memperkenalkans2n-tls, Implementasi dan Penggunaan s2n-tls TLS Sumber Terbuka Baru. GitHub

- Untuk informasi tentang Com AWS mon Runtime HTTP Client, lihat Meng konfigurasi klien AWS CRT-based HTTP di Panduan AWS SDK for Java 2.x Pengembang.
- Untuk informasi tentang proyek kriptografi pasca-kuantum di Institut Nasional untuk Standar dan Teknologi (NIST), lihat Kriptografi. Post-Quantum
- Untuk informasi tentang standardisasi kriptografi pasca-kuantum NIST, lihat Post-Quantum Standarisasi Kriptografi.

Connect ke AWS KMS melalui titik akhir VPC

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 KMS mendukung titik akhir Amazon Virtual Private Cloud (Amazon VPC) yang didukung oleh. AWS PrivateLink Setiap titik akhir VPC diwakili oleh satu atau lebih Elastic Network Interfaces (ENIs) dengan alamat IP pribadi 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. AWS KMS

Daerah

AWS KMS mendukung kebijakan titik akhir VPC dan titik akhir VPC di semua yang didukung. Wilayah AWS AWS KMS

Pertimbangan untuk titik akhir AWS KMS VPC

Sebelum Anda menyiapkan titik akhir VPC antarmuka untuk AWS KMS, tinjau topik properti dan batasan titik akhir Antarmuka di Panduan.AWS PrivateLink

AWS KMS dukungan 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. Lihat perinciannya di AWS KMS Permintaan logging yang menggunakan titik akhir VPC.

Topik

- Buat titik akhir VPC untuk AWS KMS
- Connect ke titik AWS KMS akhir VPC
- Gunakan titik akhir VPC untuk mengontrol akses ke sumber daya AWS KMS
- AWS KMS Permintaan logging yang menggunakan titik akhir VPC

Buat titik akhir VPC untuk AWS KMS

Anda dapat membuat titik akhir VPC AWS KMS dengan menggunakan konsol VPC Amazon atau API VPC Amazon. Ikuti prosedur untuk Membuat titik akhir antarmuka menggunakan salah satu nilai berikut.

- Untuk membuat titik akhir VPC 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 KMS FIPS, 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. Itu AWS SDKs dan AWS CLI gunakan nama host AWS KMS DNS standar secara default, sehingga Anda tidak perlu menentukan URL titik akhir VPC dalam aplikasi dan perintah.

Untuk informasi selengkapnya, lihat Mengakses layanan melalui titik akhir antarmuka di Panduan.AWS PrivateLink

Connect ke titik AWS KMS akhir VPC

Anda dapat terhubung AWS KMS melalui titik akhir VPC dengan menggunakan AWS SDK, file, atau. AWS CLI Alat AWS untuk 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
```

Izin yang diperlukan

Agar AWS KMS permintaan yang menggunakan titik akhir VPC 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.

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 titik akhir VPC 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

Nama host pribadi

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. SDKs Gunakan AWS CLI dan gunakan nama host ini secara default, sehingga Anda dapat mulai menggunakan titik akhir VPC untuk terhubung ke AWS KMS titik akhir 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.

Gunakan titik akhir VPC untuk mengontrol akses ke sumber daya AWS KMS

Anda dapat mengontrol akses ke AWS KMS sumber daya dan operasi saat permintaan berasal dari VPC atau menggunakan titik akhir VPC. 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 titik akhir VPC atau VPC tertentu, permintaan dari AWS layanan terintegrasi yang menggunakan AWS KMS

sumber daya atas nama Anda mungkin gagal. Untuk bantuan, lihat Menggunakan kondisi titik akhir VPC dalam kebijakan dengan AWS KMS izin.
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. Saat pengguna membuat permintaan AWS KMS, ID titik akhir VPC dalam permintaan akan dibandingkan dengan nilai kunci `aws:sourceVpce` kondisi dalam kebijakan. Jika tidak cocok, permintaan ditolak.

Untuk menggunakan kebijakan seperti ini, ganti Akun AWS ID placeholder dan titik akhir VPC IDs dengan nilai yang valid untuk akun Anda.

JSON

```
{
    "Id": "example-key-1",
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EnableIAMpolicies",
            "Effect": "Allow",
            "Principal": {"AWS":["111122223333"]},
            "Action": ["kms:*"],
            "Resource": "*"
        },
        {
            "Sid": "Restrict usage to my VPC endpoint",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "kms:Encrypt",
```

```
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                "aws:sourceVpce": "vpce-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 Akun AWS ID placeholder dan titik akhir VPC IDs dengan nilai yang valid untuk akun Anda.

JSON

```
{
    "Id": "example-key-2",
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAdministrativeActionsFromVPC",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": [
                "kms:Create*",
```

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

```
            ],
            "Resource": "*"
        }
    ]
}
```

AWS KMS Permintaan logging yang menggunakan titik akhir VPC

AWS CloudTrail mencatat semua operasi yang menggunakan titik akhir VPC. Saat permintaan untuk AWS KMS menggunakan titik akhir VPC, ID titik akhir VPC muncul di entri AWS CloudTrail log yang mencatat permintaan tersebut. Anda dapat menggunakan ID titik akhir untuk mengaudit penggunaan titik akhir AWS KMS VPC Anda.

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. Selain itu, untuk melindungi VPC Anda, permintaan yang ditolak oleh kebijakan titik akhir VPC, tetapi sebaliknya diizinkan, tidak dicatat. 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"
}
```

Dual-stack dukungan titik akhir

AWS KMS menyediakan titik akhir publik dual-stack yang mendukung klien IPv4 dan IPv6. Titik akhir dual-stack memungkinkan klien untuk berkomunikasi AWS KMS menggunakan alamat IPv4 atau IPv6. Untuk informasi selengkapnya tentang titik AWS KMS akhir, lihat titik akhir dan AWS Key Management Service kuota.

T AWS KMS itik akhir publik dual-stack at `https://kms.`*`your-region`*`.api.aws` mendukung klien IPv4 dan IPv6. AWS KMS juga dapat diakses secara pribadi melalui IPv4 dan IPv6 dari virtual private cloud (VPC) Anda menggunakan. AWS PrivateLink Untuk informasi selengkapnya tentang membuat titik akhir VPC antarmuka pribadi AWS KMS, lihat. Connect ke AWS KMS melalui titik akhir VPC

Untuk informasi selengkapnya tentang pengalamatan IPv6 untuk VPC Anda, lihat Cara kerja Amazon VPC di Panduan Pengguna Amazon Virtual Private Cloud. Untuk informasi selengkapnya tentang cara mengonfigurasi VPC untuk mode dual-stack, lihat pengal amatan IP untuk VPC dan subnet Anda di Panduan Pengguna Amazon Virtual Private Cloud.

Fitur tidak tersedia melalui IPv6

AWS KMS tidak dapat berkomunikasi melalui IPv6 dengan AWS CloudHSM penyimpanan kunci atau penyimpanan kunci eksternal. Batasan ini tidak mencegah Anda memanggil AWS KMS API melalui IPv6.

AWS KMS konsep

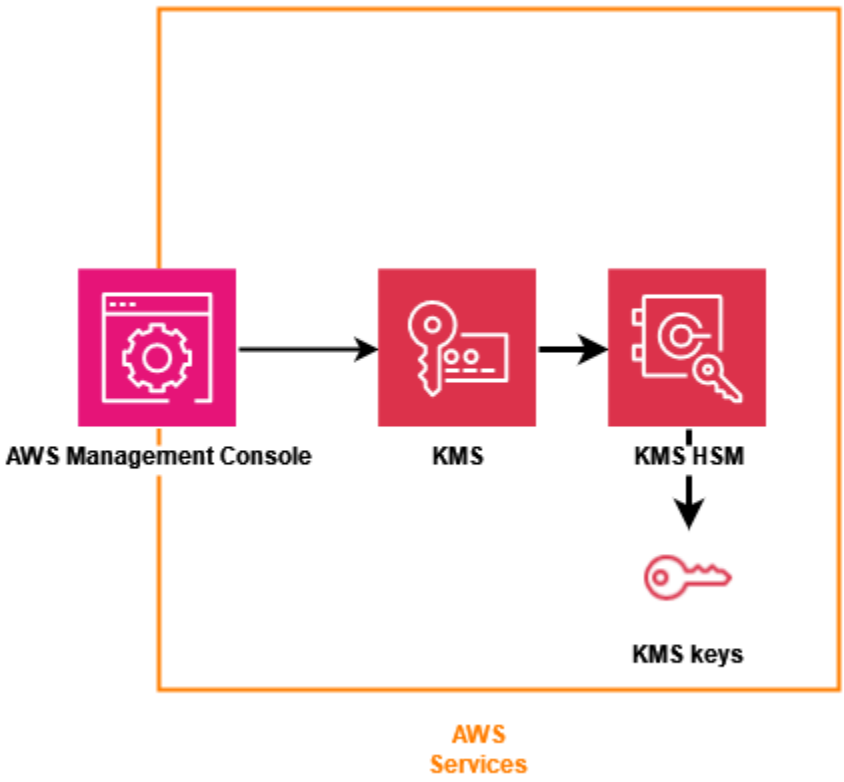
Pelajari istilah dan konsep dasar yang digunakan dalam AWS Key Management Service (AWS KMS) dan bagaimana mereka bekerja bersama untuk membantu melindungi data Anda.

Pengantar AWS KMS

AWS Key Management Service (AWS KMS) menyediakan antarmuka web untuk menghasilkan dan mengelola kunci kriptografi dan beroperasi sebagai penyedia layanan kriptografi untuk melindungi data. AWS KMS menawarkan layanan manajemen kunci tradisional yang terintegrasi dengan AWS layanan untuk memberikan pandangan yang konsisten tentang kunci pelanggan di seluruh AWS, dengan manajemen dan audit terpusat.

AWS KMS mencakup antarmuka web melalui antarmuka baris perintah Konsol Manajemen AWS, dan operasi RESTful API untuk meminta operasi kriptografi dari armada terdistribusi modul keamanan perangkat keras (HSM) tervalidasi FIPS 140-3. AWS KMS HSM adalah alat kriptografi perangkat keras mandiri multichip yang dirancang untuk menyediakan fungsi kriptografi khusus untuk memenuhi persyaratan keamanan dan skalabilitas. AWS KMS Anda dapat membuat hierarki HSM-based kriptografi Anda sendiri di bawah kunci yang Anda kelola sebagai AWS KMS keys. Kunci ini tersedia hanya di HSM dan hanya dalam memori untuk jangka waktu yang diperlukan untuk memproses permintaan kriptografi Anda. Anda dapat membuat beberapa kunci KMS, masing-masing diwakili oleh ID kuncinya. Hanya di bawah peran dan akun AWS IAM yang dikelola oleh setiap pelanggan, kunci KMS yang dikelola pelanggan dapat dibuat, dihapus, atau digunakan untuk mengenkripsi, mendekripsi, menandatangani, atau memverifikasi data. Anda dapat menentukan kontrol akses pada siapa yang dapat mengelola kunci and/or penggunaan KMS dengan membuat kebijakan yang dilampirkan ke kunci. Kebijakan tersebut memungkinkan Anda untuk menentukan penggunaan khusus aplikasi untuk kunci Anda bagi setiap operasi API.

Selain itu, sebagian besar AWS layanan mendukung enkripsi data saat istirahat menggunakan kunci KMS. Kemampuan ini memungkinkan pelanggan untuk mengontrol bagaimana dan kapan AWS layanan dapat mengakses data terenkripsi dengan mengontrol bagaimana dan kapan kunci KMS dapat diakses.



AWS KMS adalah layanan berjenjang yang terdiri dari host yang menghadap AWS KMS ke web dan tingkat HSM. Pengelompokan host berjenjang ini membentuk AWS KMS tumpukan. Semua permintaan AWS KMS harus dilakukan melalui protokol Transport Layer Security (TLS) dan diakhiri pada AWS KMS host. AWS KMS host hanya mengizinkan TLS dengan ciphersuite yang memberikan kerahasiaan maju yang sempurna. AWS KMS mengotentikasi dan mengotorisasi permintaan Anda menggunakan mekanisme kredenSIAL dan kebijakan AWS Identity and Access Management (IAM) yang sama yang tersedia untuk semua operasi API lainnya. AWS

AWS KMS tujuan desain

AWS KMS dirancang untuk memenuhi persyaratan berikut.

Daya tahan

Daya tahan kunci kriptografi dirancang untuk menyamai dengan layanan daya tahan tertinggi di AWS. Kunci kriptografi tunggal dapat mengenkripsi volume besar data Anda yang telah terakumulasi dalam waktu lama.

Dapat Dipercaya

Penggunaan kunci dilindungi oleh kebijakan kontrol akses yang Anda tetapkan dan kelola. Tidak ada mekanisme untuk mengekspor kunci KMS plaintext. Kerahasiaan kunci kriptografi Anda sangat penting. Beberapa karyawan Amazon dengan akses khusus peran untuk kontrol akses berbasis kuorum diperlukan untuk melakukan tindakan administratif pada HSM.

Low-latency dan throughput tinggi

AWS KMS menyediakan operasi kriptografi pada tingkat latensi dan throughput yang cocok untuk digunakan oleh layanan lain di AWS.

Daerah Independen

AWS menyediakan Wilayah independen untuk pelanggan yang perlu membatasi akses data di Wilayah yang berbeda. Penggunaan kunci dapat diisolasi dalam Wilayah AWS.

Sumber nomor acak yang aman

Karena kriptografi yang kuat bergantung pada generasi nomor acak yang benar-benar tidak dapat diprediksi, AWS KMS menyediakan sumber angka acak berkualitas tinggi dan tervalidasi.

Audit

AWS KMS mencatat penggunaan dan pengelolaan kunci kriptografi dalam AWS CloudTrail log. Anda dapat menggunakan AWS CloudTrail log untuk memeriksa penggunaan kunci kriptografi Anda, termasuk penggunaan kunci oleh AWS layanan atas nama Anda.

Untuk mencapai tujuan ini, AWS KMS sistem mencakup seperangkat AWS KMS operator dan operator host layanan (secara kolektif, “operator”) yang mengelola “domain.” Domain adalah kumpulan AWS KMS server, HSM, dan operator yang didefinisikan secara regional. Setiap AWS KMS operator memiliki token perangkat keras yang berisi pasangan kunci pribadi dan publik yang digunakan untuk mengotentikasi tindakannya. HSM memiliki pasangan kunci pribadi dan publik tambahan untuk membuat kunci enkripsi yang melindungi sinkronisasi status HSM.

AWS KMS keys

Kunci KMS yang Anda buat dan kelola untuk digunakan dalam aplikasi kriptografi Anda sendiri adalah jenis yang dikenal sebagai kunci yang dikelola pelanggan. Kunci yang dikelola pelanggan juga dapat digunakan bersama dengan AWS layanan yang menggunakan kunci KMS untuk mengenkripsi data yang disimpan layanan atas nama Anda. Kunci yang dikelola pelanggan

direkomendasikan untuk pelanggan yang menginginkan kontrol penuh atas siklus hidup dan penggunaan kunci mereka. Ada biaya bulanan untuk memiliki kunci yang dikelola pelanggan di akun Anda. Selain itu, permintaan penggunaan and/or mengelola kunci dikenakan biaya penggunaan. Lihat AWS Key Management Service Harga untuk lebih jelasnya.

Ada kasus di mana pelanggan mungkin menginginkan AWS layanan untuk mengenkripsi data mereka, tetapi mereka tidak menginginkan overhead pengelolaan kunci dan tidak ingin membayar kunci. An Kunci yang dikelola AWS adalah kunci KMS yang ada di akun Anda, tetapi hanya dapat digunakan dalam keadaan tertentu. Secara khusus, ini hanya dapat digunakan dalam konteks AWS layanan tempat Anda beroperasi dan hanya dapat digunakan oleh kepala sekolah di dalam akun yang kuncinya ada. Anda tidak dapat mengelola apa pun tentang siklus hidup atau izin kunci ini. Saat Anda menggunakan fitur enkripsi dalam AWS layanan, Anda mungkin melihat Kunci yang dikelola AWS; mereka menggunakan alias dari formulir “aws<service code>”. Misalnya, `aws/ebs` kunci hanya dapat digunakan untuk mengenkripsi volume EBS dan hanya untuk volume yang digunakan oleh prinsipal IAM di akun yang sama dengan kunci. Pikirkan Kunci yang dikelola AWS yang dicakup untuk digunakan hanya oleh pengguna di akun Anda untuk sumber daya di akun Anda. Anda tidak dapat berbagi sumber daya yang dienkripsi di bawah a Kunci yang dikelola AWS dengan akun lain. Meskipun Kunci yang dikelola AWS ada gratis di akun Anda, Anda akan dikenakan biaya untuk penggunaan jenis kunci ini oleh AWS layanan yang ditetapkan ke kunci tersebut.

Kunci yang dikelola AWS adalah jenis kunci lama yang tidak lagi dibuat untuk AWS layanan baru pada tahun 2021. Sebaliknya, AWS layanan baru (dan lama) menggunakan apa yang dikenal sebagai a Kunci milik AWS untuk mengenkripsi data pelanggan secara default. An Kunci milik AWS adalah kunci KMS yang ada di akun yang dikelola oleh AWS layanan, sehingga operator layanan memiliki kemampuan untuk mengelola siklus hidup dan izin penggunaannya. Dengan menggunakan Kunci milik AWS, AWS layanan dapat mengenkripsi data Anda secara transparan dan memungkinkan berbagi data lintas akun atau lintas wilayah yang mudah tanpa Anda perlu khawatir tentang izin utama. Gunakan Kunci milik AWS untuk beban kerja enkripsi berdasarkan default yang memberikan perlindungan data yang lebih mudah dan lebih otomatis. Karena kunci ini dimiliki dan dikelola oleh AWS, Anda tidak dikenakan biaya untuk keberadaan atau penggunaannya, Anda tidak dapat mengubah kebijakannya, Anda tidak dapat mengaudit aktivitas pada kunci ini, dan Anda tidak dapat menghapusnya. Gunakan kunci yang dikelola pelanggan ketika kontrol penting, tetapi gunakan Kunci milik AWS ketika kenyamanan paling penting.

	Kunci yang dikelola pelanggan	Kunci yang dikelola AWS	Kunci milik AWS

Kebijakan utama	Dikendalikan secara eksklusif oleh pelanggan	Dikendalikan oleh layanan; dapat dilihat oleh pelanggan	Dikendalikan secara eksklusif dan hanya dapat dilihat oleh AWS layanan yang mengenkripsi data Anda
Pencatatan log	CloudTrail jejak pelanggan atau penyimpanan data acara	CloudTrail jejak pelanggan atau penyimpanan data acara	Tidak dapat dilihat oleh pelanggan
Manajemen siklus hidup	Pelanggan mengelola rotasi, penghapusan, dan lokasi Regional	AWS KMS mengelola rotasi (tahunan), penghapusan, dan lokasi Regional	Layanan AWS mengelola rotasi, penghapusan, dan lokasi Regional
Harga	Biaya bulanan untuk keberadaan kunci (pro-rated per jam). Juga dikenakan biaya untuk penggunaan kunci	Tidak ada biaya bulanan; tetapi pemanggil dikenakan biaya untuk penggunaan API pada kunci ini	Tidak ada biaya kepada pelanggan

Kunci KMS yang Anda buat adalah kunci yang dikelola pelanggan. Layanan AWS yang 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	Dibuat di saya Akun AWS	Rotasi otomatis	Harga
Kunci yang dikelola pelanggan	Ya	Ya	Ya	Tidak wajib.	Biaya bulanan (per jam per jam)

Jenis kunci KMS	Dapat melihat metadata kunci KMS	Dapat mengelola kunci KMS	Dibuat di saya Akun AWS	Rotasi otomatis	Harga
					Per-use biaya
Kunci yang dikelola AWS	Ya	Tidak	Ya	Wajib. Setiap tahun (sekitar 365 hari).	Tidak ada biaya bulanan Per-use biaya (beberapa Layanan AWS membayar biaya ini untuk Anda)
Kunci milik AWS	Tidak	Tidak	Tidak	M Layanan AWS engelola strategi rotasi	Tidak ada biaya

AWS layanan yang terintegrasi dengan AWS KMS berbeda dalam dukungan mereka untuk kunci KMS. Beberapa AWS layanan mengenkripsi data Anda secara default dengan a Kunci milik AWS atau an Kunci yang dikelola AWS. Beberapa AWS layanan mendukung kunci yang dikelola pelanggan. AWS Layanan lain mendukung semua jenis kunci KMS untuk memungkinkan Anda kemudahan Kunci milik AWS, visibilitas Kunci yang dikelola AWS, atau kontrol kunci yang dikelola pelanggan. Untuk informasi terperinci tentang opsi enkripsi yang ditawarkan AWS layanan, lihat topik Enkripsi saat istirahat di panduan pengguna atau panduan pengembang 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 menetapkan dan memelihara kebijakan kunci, kebijakan IAM, dan hibah, mengaktifkan dan menonaktif kannya, memutar materi kriptografinya, menambahkan tag, membuat alias yang merujuk ke kunci KMS, dan men jadwalkan kunci KMS untuk dihapus.

Kunci yang dikelola pelanggan muncul di halaman kunci yang dikelola pelanggan Konsol Manajemen AWS untuk AWS KMS. Untuk mengidentifikasi kunci yang dikelola pelanggan secara definitif, gunakan DescribeKey operasi. 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 AWS layanan 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. Mereka dihitung terhadap ku AWS KMS ota 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 AWS layanan yang terintegrasi dengan AWS KMS.

Beberapa AWS layanan memungkinkan Anda memilih kunci Kunci yang dikelola AWS atau kunci yang dikelola pelanggan untuk melindungi sumber daya Anda dalam layanan tersebut. 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 memelihara kunci atau kebijakan utamanya, dan tidak pernah ada biaya bulanan untuk Kunci yang dikelola AWS.

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

Kebijakan kontrol sumber daya di organisasi Anda tidak berlaku untuk Kunci yang dikelola AWS.

Kunci yang dikelola AWS muncul di Kunci yang dikelola AWS halaman Konsol Manajemen AWS untuk AWS KMS. Anda juga dapat mengidentifikasi Kunci yang dikelola AWS dengan alias mereka, yang memiliki format`aws/`*`service-name`*, sepertiaws/redshift. Untuk mengidentifikasi secara definitif Kunci yang dikelola AWS, gunakan DescribeKey operasi. Untuk Kunci yang dikelola AWS, nilai `KeyManager` bidang `DescribeKey` respons adalah`AWS`.

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

> **Note**
>
> Pada bulan 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).

Tidak ada biaya bulanan untuk Kunci yang dikelola AWS. Mereka dapat dikenakan biaya untuk penggunaan melebihi tingkat gratis, tetapi beberapa AWS layanan menanggung 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 AWS jangan dihitung terhadap kuota sumber daya pada jumlah kunci KMS di setiap Wilayah akun Anda. Tetapi ketika digunakan atas nama kepala sekolah di akun Anda, kunci KMS dihitung terhadap kuota permintaan. Lihat perinciannya di Kuota.

Kunci milik AWS

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

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

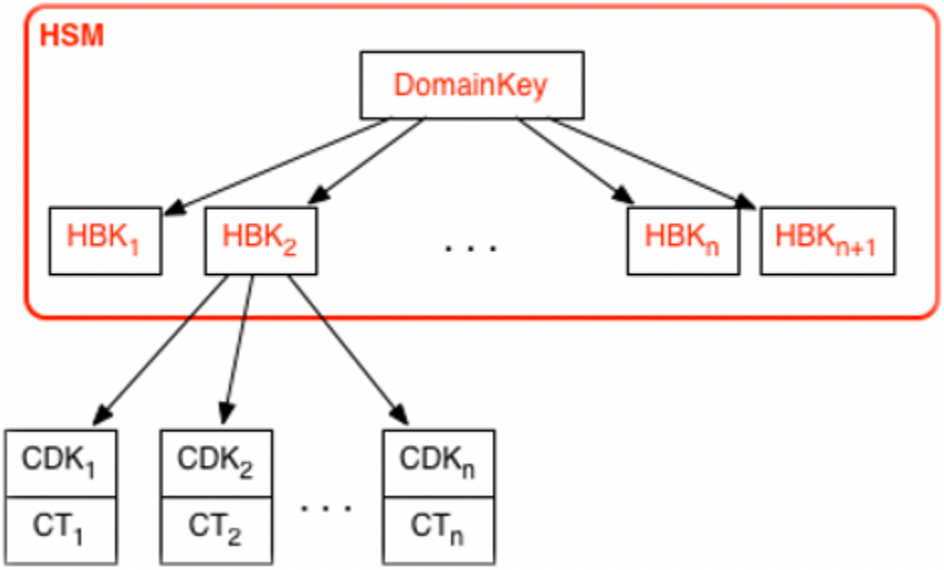
Rotasi Kunci milik AWS bervariasi antar layanan. Untuk informasi tentang rotasi tertentu Kunci milik AWS, lihat topik Enkripsi saat istirahat di panduan pengguna atau panduan pengembang untuk layanan.

AWS KMS key hierarki

Hirarki kunci Anda dimulai dengan kunci logis tingkat atas, an AWS KMS key. Kunci KMS mewakili wadah untuk materi kunci tingkat atas dan didefinisikan secara unik dalam namespace AWS layanan dengan Nama Sumber Daya Amazon (ARN). ARN menyertakan pengenal kunci yang dihasilkan secara unik, ID kunci. Kunci KMS dibuat berdasarkan permintaan yang diprakarsai pengguna melalui. AWS KMS Setelah penerimaan, AWS KMS meminta pembuatan kunci pendukung HSM awal (HBK) untuk ditempatkan ke dalam wadah kunci KMS. HBK dihasilkan pada HSM di domain dan dirancang agar tidak pernah diekspor dari HSM di plaintext. Sebaliknya, HBK diekspor terenkripsi di bawah kunci HSM-managed domain. HBK yang diekspor ini disebut sebagai token kunci yang diekspor (EKT).

EKT diekspor ke penyimpanan latensi rendah yang sangat berdaya tahan. Misalnya, Anda menerima ARN ke kunci KMS logis. Ini mewakili bagian atas hierarki kunci, atau konteks kriptografi, untuk Anda. Anda dapat membuat beberapa kunci KMS dalam akun Anda dan menetapkan kebijakan pada kunci KMS Anda seperti sumber daya AWS bernama lainnya.

Dalam hierarki kunci KMS tertentu, HBK dapat dianggap sebagai versi kunci KMS. Ketika Anda ingin memutar kunci KMS AWS KMS, HBK baru dibuat dan dikaitkan dengan kunci KMS sebagai HBK aktif untuk kunci KMS. HBK yang lebih lama disimpan dan dapat digunakan untuk mendekripsi dan memverifikasi data yang sebelumnya dilindungi. Tetapi hanya kunci kriptografi aktif yang bisa digunakan untuk melindungi informasi baru.



Anda dapat membuat permintaan melalui AWS KMS penggunaan kunci KMS Anda untuk melindungi informasi secara langsung atau meminta HSM-generated kunci tambahan yang dilindungi di bawah kunci KMS Anda. Kunci ini disebut kunci data pelanggan, atau CDK. CDK dapat dikembalikan dalam

kondisi dienkripsi sebagai ciphertext (CT), di plaintext, atau keduanya. Semua objek yang dienkripsi di bawah kunci KMS (baik data atau HSM-generated kunci yang disediakan pelanggan) dapat didekripsi hanya pada HSM melalui panggilan melalui. AWS KMS

Ciphertext yang dikembalikan, atau payload yang didekripsi, tidak pernah disimpan di dalamnya. AWS KMS Informasi dikembalikan kepada Anda melalui koneksi TLS Anda ke AWS KMS. Ini juga berlaku untuk panggilan yang dilakukan oleh AWS layanan atas nama Anda.

Hierarki kunci dan properti kunci tertentu dicantumkan dalam tabel berikut.

Kunci	Deskripsi	Siklus hidup
Kunci domain	Kunci 256-bit AES-GCM hanya dalam memori HSM yang digunakan untuk membungkus versi kunci KMS, kunci pendukung HSM.	Dirotasi setiap hari[1]
Kunci dukungan HSM	Kunci simetris 256-bit atau RSA atau kunci pribadi kurva eliptik, digunakan untuk melindungi data dan kunci pelanggan serta disimpan dalam kondisi dienkripsi berdasarkan kunci domain. Satu atau lebih kunci pendukung HSM terdiri dari kunci KMS, diwakili oleh keyID.	Dirotasi per tahun[2] (konfigurasi opsional)
Kunci enkripsi turunan	Kunci 256-bit hanya dalam memori HSM yang digunakan untuk mengenkripsi data dan AES-GCM kunci pelanggan. Berasal dari HBK untuk setiap enkripsi.	Digunakan sekali tiap enkripsi dan diregenerasi pada dekripsi
Kunci data pelanggan	User-dekunci simetris atau asimetris tetap diekspor dari HSM dalam plaintext dan ciphertext. Dienkripsi berdasarkan kunci cadangan HSM dan dikembalikan ke	Rotasi dan penggunaan dikendalikan oleh aplikasi

Kunci	Deskripsi	Siklus hidup
	pengguna yang diotorisasi melalui saluran TLS.	

[Satu] AWS KMS mungkin dari waktu ke waktu melonggarkan rotasi kunci domain menjadi paling banyak mingguan untuk memperhitungkan administrasi domain dan tugas konfigurasi.

[2 Def] ault Kunci yang dikelola AWS yang dibuat dan dikelola oleh AWS KMS atas nama Anda secara otomatis dirotasi setiap tahun.

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 mendefinisikan beberapa pengidentifikasi kunci. Saat Anda membuat kunci KMS, AWS KMS menghasilkan kunci ARN dan ID kunci, 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 di Konsol Manajemen AWS dan di AWS KMS API.

Di AWS KMS konsol, Anda dapat melihat dan memfilter kunci KMS berdasarkan kunci ARN, ID kunci, atau nama alias, dan mengurutkan berdasarkan ID kunci dan nama alias. Untuk bantuan menemukan pengidentifikasi kunci di konsol, lihatthe section called “Temukan 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 Refer AWS Key Management Service ensi API.

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

AWS KMS mendukung pengenal kunci berikut.

ARN kunci

Kunci ARN adalah Nama Sumber Daya Amazon (ARN) dari kunci KMS. Ini adalah pengidentifikasi unik dan memenuhi syarat untuk kunci KMS. Kunci ARN mencakup Akun AWS, Wilayah, dan ID kunci. Untuk bantuan menemukan kunci ARN dari kunci KMS, lihatthe section called “Temukan 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 wilayah tunggal.

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

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

```
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 “Temukan ID kunci dan kunci ARN”.

Berikut ini adalah contoh ID kunci untuk kunci KMS wilayah tunggal.

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

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

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

ARN Alias

Alias ARN adalah Nama Sumber Daya Amazon (ARN) dari sebuah AWS KMS alias. Ini adalah pengidentifikasi unik dan memenuhi syarat untuk alias, dan untuk kunci KMS yang diwakilinya. Alias ARN mencakup Akun AWS, Region, 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, lihatTemukan nama alias dan alias ARN untuk kunci KMS.

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. Di AWS KMS API, nama alias selalu dimulai dengan`alias/`. Untuk bantuan menemukan nama alias kunci KMS, lihatTemukan nama alias dan alias ARN untuk kunci KMS.

Format nama alias adalah sebagai berikut:

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

Contoh:

```
alias/ExampleAlias
```

A `aws/` walan 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) adalah sebagai berikut.

```
alias/aws/s3
```

Kunci 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 pribadi, Anda harus menelepon 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 menelepon AWS 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. AWS layanan yang terintegrasi dengan hanya AWS KMS menggunakan kunci KMS enkripsi simetris untuk mengenkripsi data Anda. Layanan ini tidak mendukung enkripsi dengan kunci KMS asimetris.

Saat menandatangani pesan yang lebih besar dari 4 KB dengan AWS KMS, Anda harus melakukan hash pesan di luar AWS KMS sebelum menandatangani. AWS KMS menyediakan tiga `MessageType` opsi untuk menangani input pesan: `RAW` untuk pesan teks biasa (di mana AWS KMS melakukan hashing), `DIGEST` untuk pesan pra-hash (di mana AWS KMS melewatkan langkah hashing), dan `EXTERNAL_MU` khusus untuk spesifikasi kunci KMS ML-DSA di mana inputnya adalah perwakilan 64-byte nilai μ. Untuk pesan besar yang melebihi batas 4 KB, hash pesan secara eksternal dan gunakan `MessageType:DIGEST`(atau `MessageType:EXTERNAL_MU`untuk kunci KMS ML-DSA) saat memanggil operasi Masuk dan Verifikasi. AWS KMSAWS KMS

AWS KMS mendukung beberapa jenis kunci KMS asimetris.

Kunci RSA KMS

Kunci KMS dengan key pair RSA untuk enkripsi dan dekripsi atau penandatanganan dan verifikasi (tetapi tidak keduanya). AWS KMS mendukung beberapa panjang kunci untuk persyaratan keamanan yang berbeda.

Untuk detail teknis tentang enkripsi dan algoritma penandatanganan yang AWS KMS mendukung kunci RSA KMS, lihat spesifikasi kunci RSA.

Elliptic Curve (ECC) KMS Kuncinya

Kunci KMS dengan elliptic curve key pair untuk penandatanganan dan verifikasi atau menurunkan rahasia bersama (tetapi tidak keduanya). AWS KMS mendukung beberapa kurva yang umum digunakan.

Untuk detail teknis tentang algoritma penandatanganan yang AWS KMS mendukung kunci ECC KMS, lihat Spesifikasi kunci kurva elips.

Kunci KMS ML-DSA

Kunci KMS dengan key pair ML-DSA untuk penandatanganan dan verifikasi. ML-DSA adalah standar kriptografi pasca-kuantum yang dikembangkan oleh Institut Standar dan Teknologi Nasional AS (NIST) untuk melindungi terhadap ancaman keamanan yang ditimbulkan oleh komputasi kuantum. ML-DSA adalah algoritma tanda tangan digital yang direkomendasikan untuk organisasi yang beralih dari algoritma tanda tangan digital RSA atau Elliptic Curve ke kriptografi aman pasca-kuantum.

AWS KMS mendukung beberapa panjang kunci untuk persyaratan keamanan yang berbeda. Untuk detail teknis tentang algoritma penandatanganan yang AWS KMS mendukung kunci KMS ML-DSA, lihat spesifikasi kunci ML-DSA.

SM2 Kunci KMS (hanya Wilayah Tiongkok)

Kunci KMS dengan SM2 key pair untuk enkripsi dan dekripsi, penandatanganan dan verifikasi, atau memperoleh rahasia bersama (Anda harus memilih satu jenis). Key usage

Untuk detail teknis tentang enkripsi dan algoritma penandatanganan yang AWS KMS mendukung kunci SM2 KMS (hanya Wilayah Tiongkok), lihat SM2 spesifikasi kunci.

Untuk bantuan memilih konfigurasi kunci asimetris Anda, lihatMemilih jenis kunci KMS apa yang akan dibuat.

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. Buat kunci KMS asimetris
- Untuk membuat kunci KMS asimetris Multi-wilayah, lihat. Buat kunci utama Multi-wilayah
- 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 KMSAWS
- Untuk mempelajari tentang pertimbangan khusus untuk menghapus kunci KMS asimetris, lihat. Deleting asymmetric KMS keys
- Untuk mengidentifikasi dan melihat kunci KMS asimetris, lihat. Identifikasi kunci KMS asimetris

Kunci HMAC di AWS KMS

Kunci KMS Kode Otentikasi Pesan Berbasis Hash (HMAC) adalah kunci simetris yang Anda gunakan untuk membuat dan memverifikasi di dalamnya. HMACs 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. HMACs 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 KMS GenerateMacOperasi menghasilkan tag HMAC standar. Kunci KMS HMAC dihasilkan dalam modul keamanan AWS KMS perangkat keras yang disertifikasi di bawah Program Validasi Modul Kriptografi FIPS 140-3 (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 HMACs dari perangkat lunak aplikasi karena materi utama dihasilkan dan digunakan sepenuhnya di dalam AWS 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.

Operasi kriptografi yang didukung

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 membuat kunci HMAC KMS, lihat. Buat kunci HMAC KMS
- Untuk membuat kunci KMS HMAC Multi-wilayah, lihat. Multi-Region kunci dalam 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 untuk operasi kriptografi”
- Untuk mengidentifikasi dan melihat kunci HMAC KMS, lihat. Identifikasi kunci HMAC KMS
- Untuk mempelajari cara menggunakan HMACs token web JSON, lihat Cara melindungi HMACs bagian dalam AWS KMS di Blog AWS Keamanan.
- Dengarkan podcast: Memperkenalkan HMACs AWS Key Management Service di The Official AWS Podcast.

Kunci ML-DSA di AWS KMS

AWS Key Management Service (AWS KMS) mendukung Module-Lattice Digital Signature Algorithm (ML-DSA) untuk tanda tangan kriptografi pasca-kuantum. Implementasi ini mengikuti standar Federal Information Processing Standards (FIPS) 204 untuk membantu melindungi terhadap ancaman komputasi kuantum masa depan. AWS KMS membuat dan melindungi semua kunci ML-DSA dan operasi tanda tangan di modul keamanan perangkat keras yang divalidasi FIPS 140-3 Security Level 3. Untuk membantu menyeimbangkan keamanan dengan kinerja, ML-DSA AWS KMS menawarkan tiga tingkat keamanan yang berbeda melalui spesifikasi kunci yang berbeda, ML_DSA_44, ML_DSA_65, dan ML_DSA_87.

AWS KMS mendukung tanda tangan kunci asimetris untuk pesan hingga 4 KB menggunakan jenis pesan. `RAW` Untuk pesan yang lebih besar, Anda harus menghitung secara eksternal representasi pesan 64-byte μ yang digunakan dalam penandatanganan ML-DSA seperti yang didefinisikan dalam NIST FIPS 204 bagian 6.2. Gunakan jenis `EXTERNAL_MU` pesan dalam operasi AWS KMS Tanda tangan untuk menentukan pesan 64-byte yang telah diproses sebelumnya ini. Tanda tangan yang dihasilkan oleh μ yang dihitung secara eksternal sama dengan tanda tangan saat menggunakan pesan dan `RAW` kunci pribadi yang sama. Perhatikan bahwa penandatanganan ini berbeda dari “pra-hash” ML-DSA atau HashML-DSA dari bagian 5.4 dari NIST FIPS 204.

Untuk informasi selengkapnya tentang penggunaan ML-DSA dan jenis pesan EXTERNAL_MU, lihat. Spesifikasi kunci ML-DSA

Untuk contoh menggunakan ML-DSA dan jenis pesan EXTERNAL_MU, lihat. Verifikasi offline dengan pasangan ML-DSA kunci

Multi-Region kunci dalam 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.

Multi-Region kunci adalah solusi yang fleksibel dan kuat untuk banyak 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.

Active-active aplikasi 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-region dengan pustaka enkripsi sisi klien, seperti, AWS Database Encryption SDK AWS Encryption SDK, dan enkripsi sisi klien Amazon S3.

Sebagian besar 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. Misalnya, replikasi lintas wilayah Amazon S3 mendekripsi dan mengenkripsi ulang kunci data yang digunakan untuk mengenkripsi data objek di bawah kunci KMS di Wilayah tujuan, bahkan ketika kunci KMS di kedua Wilayah adalah kunci Multi-wilayah terkait. Lihat dokumentasi khusus layanan untuk memahami bagaimana layanan mereplikasi data terenkripsi dan apakah layanan memperlakukan kunci Multi-region secara berbeda.

Multi-Region kunci tidak 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 AWS 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.

Di Wilayah Tiongkok, Anda dapat menggunakan fitur kunci Multi-wilayah untuk mereplikasi kunci KMS dalam partisi Wilayah Tiongkok (). `aws-cn` Misalnya, Anda dapat mereplikasi kunci dari Wilayah

Tiongkok (Beijing) ke Wilayah Tiongkok (Ningxia), atau sebaliknya. Dengan mereplikasi kunci dari satu wilayah Tiongkok ke wilayah lain, Anda setuju untuk menggunakan wilayah tujuan dan mematuhi semua ketentuan perjanjian yang berlaku untuk wilayah tujuan. AWS Key Management Service Anda tidak dapat mereplikasi kunci dari Wilayah Beijing dan Ningxia menjadi AWS Wilayah di luar partisi Wilayah Tiongkok. Demikian pula, Anda tidak dapat mereplikasi kunci dari wilayah di luar partisi Wilayah Tiongkok ke Wilayah Beijing dan Ningxia.

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

Multi-Region kunci didukung di semua Wilayah AWS yang AWS KMS mendukung.

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
- Kunci Asymmetric KMS
- Kunci HMAC KMS
- Kunci KMS dengan bahan kunci impor

Anda tidak dapat membuat kunci multi-Wilayah di penyimpanan kunci kustom.

Pelajari selengkapnya

- Untuk mempelajari cara mengontrol akses ke kunci KMS Multi-wilayah, lihat. Kontrol akses ke tombol Multi-wilayah
- Untuk membuat kunci KMS primer Multi-wilayah dari jenis apa pun, lihat. Buat kunci utama Multi-wilayah
- Untuk membuat kunci KMS replika Multi-wilayah, lihat. Buat kunci replika Multi-wilayah
- Untuk memperbarui Wilayah utama, lihatUbah kunci utama dalam satu set kunci Multi-wilayah.
- Untuk mengidentifikasi dan melihat kunci KMS Multi-wilayah, lihat. Identifikasi kunci HMAC KMS
- Untuk mempelajari tentang pertimbangan khusus untuk menghapus kunci KMS Multi-wilayah, lihat. Deleting multi-Region keys

Terminologi dan konsep

Istilah dan konsep berikut digunakan dengan kunci multi-Wilayah.

Multi-Region kunci

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 dan mereplikasi jika dan ketika 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. Untuk kunci KMS dengan `AWS_KMS` asal, dengan AWS KMS aman mengangkut materi kunci melintasi batas Wilayah dan mengaitkannya dengan kunci replika baru, semuanya di dalamnya. AWS KMS Untuk kunci KMS dengan `EXTERNAL` asal, Anda harus mengimpor materi kunci yang sama dengan yang Anda impor ke kunci Wilayah utama ke kunci Region replika jangkauan satu per satu.

Properti bersama

Properti bersama adalah properti kunci utama 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.)
- Materi kunci — Kunci primer dan replika dalam satu set kunci Multi-wilayah terkait berbagi materi kunci yang sama. Untuk kunci Multi-wilayah yang bahan utamanya dihasilkan oleh AWS KMS (`AWS_KMS`asal), dengan AWS KMS aman mengangkut semua bahan utama dari primer ke setiap replika saat replika dibuat atau ketika bahan kunci baru dibuat melalui rotasi otomatis atau sesuai permintaan. Untuk kunci Multi-region dengan material kunci impor (`EXTERNAL`asal), AWS KMS sinkronkan pengenal material kunci dari kunci utama tetapi Anda harus mengimpor materi kunci ke setiap kunci replika secara independen.
- 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. Lihat perinciannya di Rotating multi-Region keys.
- On-demand rotasi — Anda dapat melakukan rotasi sesuai permintaan hanya pada kunci utama. Untuk kunci multi-wilayah yang materi utamanya dihasilkan oleh AWS KMS (`AWS_KMS`asal), AWS KMS buat kunci replika dengan semua versi materi kunci bersama. Untuk kunci multi-wilayah dengan bahan kunci impor (`EXTERNAL`asal), AWS KMS menyebarkan Id material kunci dan deskripsi material kunci dari kunci utama ke kunci replika, tetapi bukan bahan kunci. Anda harus mengimpor bahan kunci yang benar ke setiap kunci replika satu per satu. Lihat perinciannya di Rotating multi-Region keys.

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 kunci Multi-region lainnya adalah properti independen, termasuk deskripsi kunci, kebijakan kunci, hibah, status kunci yang 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.

Pertimbangan keamanan untuk kunci multi-Wilayah

Gunakan tombol AWS KMS Multi-region hanya jika Anda membutuhkannya. Multi-Region kunci menyediakan solusi yang fleksibel dan dapat diskalakan untuk beban kerja yang memindahkan data terenkripsi di antara Wilayah AWS atau memerlukan 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 signifikan untuk Regionally-isolated layanan dan sumber daya. 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.

Multi-Region kunci 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

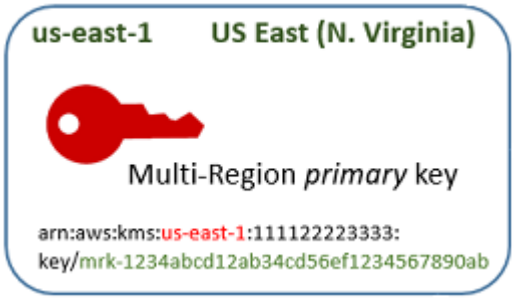
Untuk memudahkan Anda mengelola kontrol akses pada kunci Multi-region, izin untuk mereplikasi kunci Multi-region (kms: ReplicateKey) terpisah dari izin standar untuk membuat kunci (kms:). CreateKey Selain itu, AWS KMS mendukung beberapa kondisi kebijakan untuk kunci Multi-wilayah, termasuk`kms:MultiRegion`, 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. Lihat perinciannya di Kontrol akses ke tombol 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

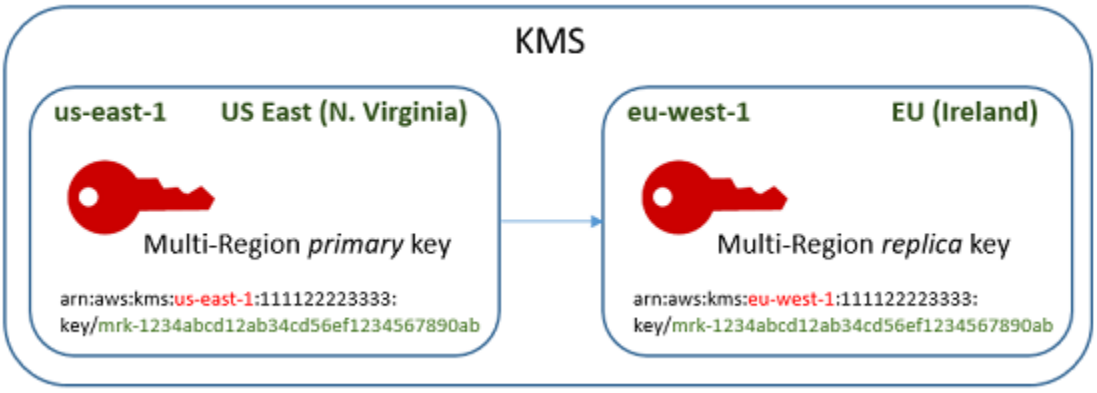
Wilayah tunggal atau multi-Wilayah hanya ketika Anda membuatnya; Anda tidak dapat mengubah properti ini nanti. Seperti halnya kunci KMS lainnya, Anda menetapkan kebijakan kunci untuk kunci Multi-region, dan Anda dapat membuat hibah, serta menambahkan alias dan tag untuk kategorisasi dan otorisasi. (Ini 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 `ketrue`. 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. 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

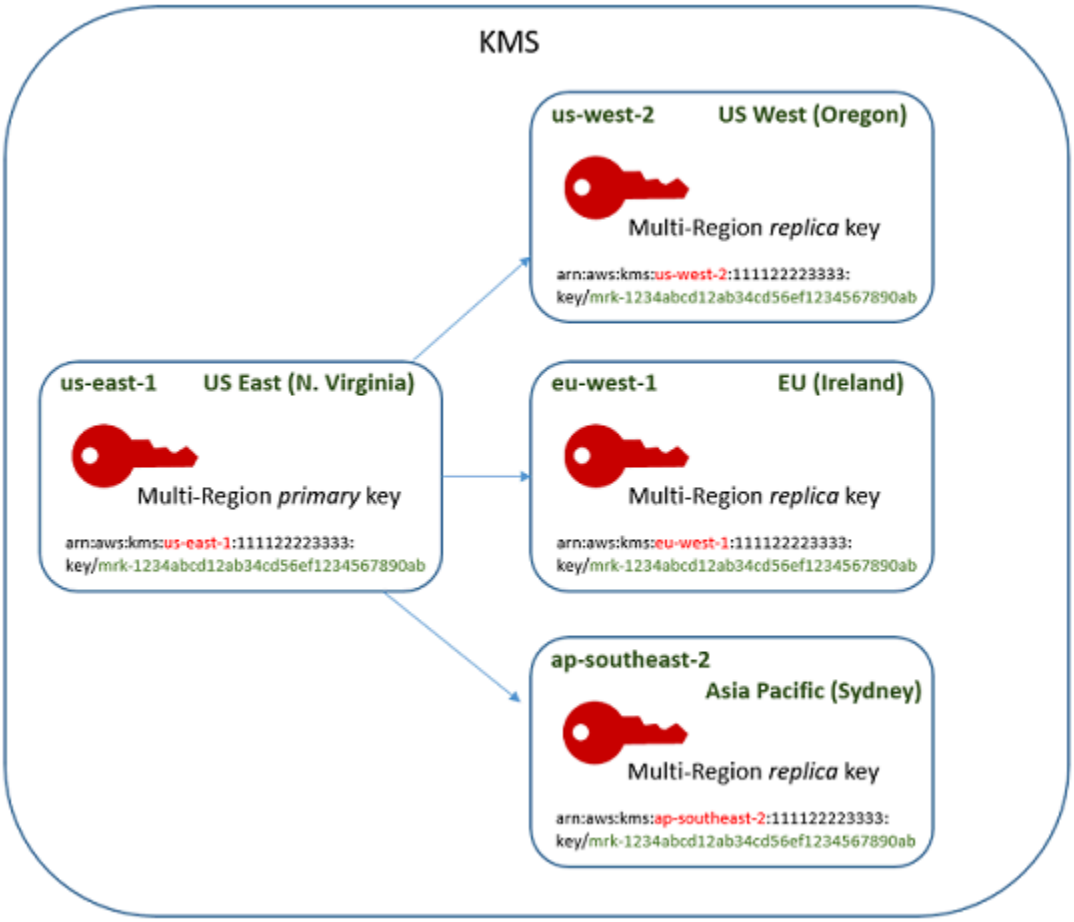
dengan deskripsi, kebijakan kunci, hibah, alias, dan tag sendiri. Mengaktifkan atau menonaktifkan kunci multi-Wilayah tidak berpengaruh pada kunci multi-Wilayah terkait. Anda dapat menggunakan kunci primer dan replika secara independen dalam operasi kriptografi atau mengoordinasikan penggunaannya. Misalnya, 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. Untuk kunci multi-wilayah dengan bahan kunci impor, Anda bertanggung jawab untuk mengimpor materi utama ke setiap kunci terkait satu per satu. 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 dengan `AWS_KMS` asal, 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. Lihat perinciannya di Rotating multi-Region keys.

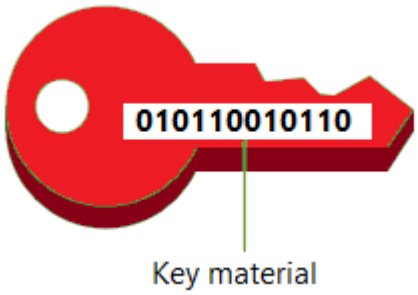
Mengubah kunci primer — Setiap set kunci multi-Wikayah harus memiliki tepatnya satu kunci utama. Parameter kunci primer adalah satu-satunya kunci yang dapat direplikasi. Ini juga merupakan sumber properti bersama dari kunci replika. Tapi Anda dapat mengubah kunci primer untuk replika dan mempromosikan 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. Lihat perinciannya di Ubah kunci utama dalam satu set kunci Multi-wilayah.

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 Deleting multi-Region keys.

Mengimpor bahan utama untuk AWS KMS keys

Anda dapat membuat AWS KMS keys (kunci KMS) dengan bahan kunci yang Anda berikan.

Kunci KMS adalah representasi logis dari kunci data. Metadata untuk kunci KMS mencakup ID materi kunci yang digunakan untuk melakukan operasi kriptografi. Saat Anda membuat kunci KMS, secara default, AWS KMS menghasilkan materi kunci untuk kunci KMS tersebut. Tetapi Anda dapat membuat kunci KMS tanpa materi kunci dan kemudian mengimpor materi kunci Anda sendiri ke kunci KMS itu, fitur yang sering dikenal sebagai “bawa kunci Anda sendiri” (BYOK).



> **Note**
>
> AWS KMS tidak mendukung dekripsi AWS KMS ciphertext apa pun yang dienkripsi oleh kunci KMS enkripsi simetris di luar AWS KMS, bahkan jika ciphertext dienkripsi di bawah kunci KMS dengan materi kunci yang diimpor. AWS KMS tidak menerbitkan format ciphertext yang diperlukan tugas ini, dan formatnya mungkin berubah tanpa pemberitahuan.

Ketika Anda menggunakan bahan kunci yang diimpor, 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 kunci dihasilkan menggunakan sumber entropi yang memenuhi kebutuhan Anda.
- Untuk menggunakan materi kunci dari infrastruktur Anda sendiri dengan AWS layanan, dan untuk digunakan AWS KMS untuk mengelola siklus hidup materi kunci di dalamnya AWS.
- Untuk menggunakan kunci yang sudah ada dan mapan AWS KMS, seperti kunci untuk penandatanganan kode, penandatanganan sertifikat PKI, dan aplikasi yang disematkan 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 materi kunci, dan menyimpannya di luar AWS untuk daya tahan tambahan dan pemulihan bencana selama siklus hidup lengkap materi kunci.
- Untuk kunci asimetris dan kunci HMAC, mengimpor menciptakan kunci yang kompatibel dan dapat dioperasikan yang beroperasi di dalam dan di luar. AWS

Jenis 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 penyimpanan kunci khusus.

- Kunci KMS enkripsi simetris
- Tombol KMS asimetris (kecuali ML-DSA tombol)
- Kunci HMAC KMS
- Multi-Region kunci dari semua jenis yang didukung.

Wilayah

Bahan kunci yang diimpor didukung di semua Wilayah AWS yang AWS KMS mendukung.

Di Wilayah China, persyaratan material utama untuk kunci KMS enkripsi simetris berbeda dari Wilayah lainnya. Lihat perinciannya di Langkah 3: Enkripsi material kunci.

Pelajari selengkapnya

- Untuk membuat kunci KMS dengan bahan kunci yang diimpor, lihatBuat kunci KMS dengan bahan kunci yang diimpor.
- Untuk membuat alarm yang memberi tahu Anda saat materi kunci yang diimpor dalam kunci KMS mendekati waktu kedaluwarsa, lihat. Buat CloudWatch alarm untuk kedaluwarsa materi kunci yang diimpor
- Untuk mengimpor kembali materi kunci ke kunci KMS, lihat. Impor ulang bahan kunci
- Untuk mengimpor materi kunci baru ke kunci KMS untuk rotasi sesuai permintaan, lihat Mengimpor materi kunci baru danLakukan rotasi kunci sesuai permintaan.
- Untuk mengidentifikasi dan melihat kunci KMS dengan materi kunci yang diimpor, lihatIdentifikasi kunci KMS dengan bahan kunci impor.
- Untuk mempelajari tentang pertimbangan khusus untuk menghapus kunci KMS dengan materi kunci yang diimpor, lihatDeleting KMS keys with imported key material.

Pertimbangan khusus untuk bahan kunci yang diimpor

Sebelum Anda memutuskan untuk mengimpor bahan kunci ke AWS KMS, Anda harus memahami karakteristik berikut dari bahan kunci yang diimpor.

Anda menghasilkan material kunci

Anda bertanggung jawab untuk menghasilkan materi kunci menggunakan sumber keacakan yang memenuhi persyaratan keamanan Anda.

Anda bertanggung jawab atas ketersediaan dan daya tahan

AWS KMS dirancang untuk menjaga bahan kunci yang diimpor sangat tersedia. Namun AWS KMS tidak mempertahankan daya tahan bahan kunci yang diimpor pada tingkat yang sama dengan bahan kunci yang AWS KMS dihasilkan. Lihat perinciannya di Melindungi bahan kunci yang diimpor.

Anda dapat menghapus materi kunci

Anda dapat menghapus materi kunci yang diimpor dari kunci KMS, segera membuat kunci KMS tidak dapat digunakan. Juga, saat Anda mengimpor materi kunci ke kunci KMS, Anda dapat menentukan apakah kunci kedaluwarsa dan mengatur waktu kedaluwarsa. Ketika waktu kedaluwarsa tiba AWS KMS , menghapus materi kunci. Tanpa materi kunci, kunci KMS tidak

dapat digunakan dalam operasi kriptografi apa pun. Untuk mengembalikan kunci, Anda harus mengimpor kembali materi kunci yang sama ke dalam kunci.

Anda tidak dapat mengubah materi kunci untuk kunci asimetris, dan HMAC

Saat Anda mengimpor materi kunci ke kunci KMS, kunci KMS secara permanen dikaitkan dengan materi kunci tersebut. Anda dapat mengimpor kembali materi kunci yang sama, tetapi Anda tidak dapat mengimpor materi kunci yang berbeda ke dalam kunci KMS tersebut. Selain itu, Anda tidak dapat mengaktifkan rotasi kunci otomatis untuk kunci KMS dengan bahan kunci yang diimpor. Namun, Anda dapat memutar kunci KMS secara manual dengan bahan kunci yang diimpor.

Anda dapat melakukan rotasi sesuai permintaan pada kunci enkripsi simetris

Kunci enkripsi simetris dengan bahan kunci yang diimpor mendukung rotasi sesuai permintaan. Anda dapat mengimpor beberapa bahan kunci ke dalam kunci ini dan menggunakan rotasi sesuai permintaan untuk memperbarui materi kunci saat ini. Materi kunci saat ini digunakan untuk enkripsi dan dekripsi tetapi bahan kunci (tidak arus) lainnya hanya dapat digunakan untuk dekripsi.

Anda tidak dapat mengubah asal bahan utama

Kunci KMS yang dirancang untuk bahan kunci yang diimpor 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 mengubah kunci KMS dengan bahan AWS KMS kunci menjadi kunci yang dirancang untuk bahan kunci yang diimpor.

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 yang diimpor di luar AWS, sebaiknya di pengelola kunci, seperti modul keamanan perangkat keras (HSM), sehingga Anda dapat mengimpor kembali materi kunci jika Anda menghapusnya atau kedaluwarsa.

Anda dapat membuat kunci multi-wilayah dengan bahan kunci yang diimpor

Multi-Region dengan bahan kunci yang diimpor memiliki fitur kunci KMS dengan bahan kunci yang diimpor, dan dapat saling beroperasi di antaranya. Wilayah AWS Untuk membuat kunci Multi-wilayah dengan materi kunci yang diimpor, Anda harus mengimpor materi kunci yang sama ke kunci KMS utama dan ke setiap kunci replika. Untuk detail selengkapnya tentang mengimpor bahan kunci untuk kunci Multi-wilayah, lihatMengimpor materi kunci baru.

Kunci asimetris dan kunci HMAC portabel dan dapat dioperasikan

Anda dapat menggunakan bahan kunci asimetris dan bahan kunci HMAC di luar AWS untuk berinteraksi dengan AWS KMS kunci dengan bahan kunci impor yang sama.

Tidak seperti cipher AWS KMS text simetris, yang terikat erat dengan kunci KMS yang digunakan dalam algoritma, AWS KMS menggunakan format HMAC standar dan asimetris untuk enkripsi, penandatanganan, dan pembuatan MAC. Akibatnya, kuncinya portabel dan mendukung skenario kunci escrow tradisional.

Ketika kunci KMS Anda telah mengimpor materi kunci, Anda dapat menggunakan materi kunci yang diimpor di luar AWS untuk melakukan operasi berikut.

- Kunci HMAC — Anda dapat memverifikasi tag HMAC yang dihasilkan oleh kunci HMAC KMS dengan materi kunci yang diimpor. Anda juga dapat menggunakan kunci HMAC KMS dengan materi 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.
- Kunci perjanjian kunci asimetris — Anda dapat menggunakan kunci KMS perjanjian kunci asimetris Anda dengan materi kunci yang diimpor untuk mendapatkan rahasia bersama dengan rekan di luar. AWS

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 materi kunci yang diimpor.

Kunci pribadi RSA

- AWS KMS membutuhkan kunci pribadi RSA yang diimpor untuk memiliki faktor utama yang sesuai dengan pengujian yang dijelaskan dalam FIPS 186-5, Bagian A. 1.3. Perangkat lunak atau perangkat lain mungkin menggunakan algoritma yang berbeda untuk memvalidasi faktor-

faktor utama kunci pribadi RSA ini. Dalam kasus yang jarang terjadi, kunci yang divalidasi menggunakan algoritma lain mungkin tidak diterima oleh AWS KMS.

Kunci enkripsi simetris tidak portabel atau dapat dioperasikan

Ciphertext simetris yang dihasilkan tidak portabel AWS KMS atau dapat dioperasikan. AWS KMS tidak menerbitkan format ciphertext simetris yang diperlukan portabilitas, dan formatnya mungkin berubah tanpa pemberitahuan.

- AWS KMS tidak dapat mendekripsi ciphertext simetris yang Anda enkripsi di lu AWS arnya, 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 dienkripsi data yang dapat mendekripsinya.

Selain itu, Anda tidak dapat menggunakan AWS alat apa pun, seperti enkripsi sisi AWS Encryption SDK klien Amazon S3, untuk mendekripsi ciphertext AWS KMS simetris.

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 teks sandi 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.

Melindungi bahan kunci yang diimpor

Bahan utama yang Anda impor dilindungi saat transit dan saat istirahat. Sebelum mengimpor materi kunci, Anda mengenkripsi (atau “membungkus”) materi kunci dengan kunci publik dari pasangan kunci RSA yang dihasilkan dalam modul keamanan AWS KMS perangkat keras (HSM) yang divalidasi di bawah Program Validasi Modul Kriptografi FIPS 140-3. 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 mendekripsi materi kunci dengan kunci pribadi yang sesuai dalam 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 digunakan dan hanya dalam AWS KMS HSM.

Penggunaan kunci KMS Anda dengan materi kunci yang diimpor 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 mem antaunya menggunakan layanan seperti AWS CloudTrail.

Namun, Anda menyimpan satu-satunya salinan yang aman dari materi kunci Anda. Sebagai imbalan atas ukuran kontrol ekstra ini, Anda bertanggung jawab atas daya tahan dan ketersediaan keseluruhan bahan utama yang diimpor. AWS KMS dirancang untuk menjaga bahan kunci yang diimpor sangat tersedia. Namun AWS KMS tidak mempertahankan daya tahan bahan kunci yang diimpor pada tingkat yang sama dengan bahan kunci yang AWS KMS dihasilkan.

Perbedaan daya tahan ini bermakna dalam kasus-kasus berikut:

- Saat Anda menetapkan waktu kedaluwar sa untuk materi kunci yang diimpor, AWS KMS menghapus materi kunci setelah kedaluwarsa. AWS KMS tidak menghapus kunci KMS atau metadatanya. Anda dapat membuat CloudWatch alarm Amazon yang memberi tahu Anda ketika materi kunci yang diimpor mendekati tanggal kedaluwarsa.

  Anda tidak dapat menghapus materi kunci yang dihasilkan AWS KMS untuk kunci KMS dan Anda tidak dapat mengatur materi AWS KMS kunci untuk kedaluwarsa.
- Saat Anda menghapus materi kunci yang diimpor secara AWS KMS manual, 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, metadata, dan materi kuncinya.
- Jika tidak mungkin terjadi kegagalan di seluruh wilayah tertentu yang mempengaruhi AWS KMS (seperti kehilangan daya total), materi kunci yang diimpor AWS KMS tidak dapat secara otomatis memulihkan. Namun, AWS KMS dapat mengembalikan kunci KMS dan metadatanya.

Anda harus menyimpan salinan materi kunci yang diimpor AWS di luar sistem yang Anda kendalikan. Kami menyarankan Anda menyimpan salinan materi kunci yang diimpor yang dapat diekspor dalam sistem manajemen kunci, seperti HSM. Sebagai praktik terbaik, Anda harus menyimpan referensi ke kunci KMS ARN dan ID materi kunci yang dihasilkan AWS KMS bersama salinan materi kunci yang dapat diekspor. Jika materi kunci yang diimpor dihapus atau kedaluwarsa, kunci KMS terkait menjadi tidak dapat digunakan sampai Anda mengimpor kembali materi kunci yang

sama. Jika materi kunci yang diimpor hilang secara permanen, ciphertext apa pun yang dienkripsi di bawah kunci KMS tidak dapat dipulihkan.

> **Important**
>
> Kunci enkripsi simetris dapat memiliki beberapa bahan kunci yang terkait dengannya. Seluruh kunci KMS menjadi tidak dapat digunakan segera setelah Anda menghapus salah satu materi kunci tersebut atau jika salah satu dari materi kunci tersebut kedaluwarsa (kecuali materi kunci yang dihapus atau kedaluwarsa adalah `PENDING_ROTATION` atau). `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` Anda harus mengimpor kembali materi kunci yang kedaluwarsa atau dihapus yang terkait dengan kunci tersebut sebelum kunci tersebut dapat digunakan untuk operasi kriptografi.

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 Bila Anda menggunakan kunci KMS di toko AWS CloudHSM kunci, operasi kriptografi dilakukan di HSMs dalam 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 tombol KMS untuk melakukan operasi kriptografi berikut:
  - Enkripsi
  - Dekripsi
  - GenerateDataKey

- GenerateDataKeyWithoutPlaintext
- ReEncrypt

Operasi yang menghasilkan pasangan kunci data asimetris, GenerateDataKeyPairdan GenerateDataKeyPairWithoutPlaintext, tidak didukung di penyimpanan kunci khusus.

- Gunakan kunci KMS dengan AWS layanan yang terintegrasi dengan AWS KMS dan mendukung kunci yang dikelola pelanggan.
- Lacak penggunaan kunci KMS Anda di AWS CloudTrail log dan alat CloudWatch pemantauan Amazon.

Fitur yang tidak didukung

- AWS CloudHSM toko 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 KMS menghasilkan 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.

Menggunakan kunci KMS di toko AWS CloudHSM kunci

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 KMS konsol atau `KeyState` bidang dalam DescribeKeyrespons.
- Toko AWS CloudHSM kunci harus terhubung ke AWS CloudHSM klasternya. Statusnya di AWS KMS konsol atau `ConnectionState` dalam DescribeCustomKeyStoresrespons harus`CONNECTED`.

- AWS CloudHSM Cluster yang terkait dengan penyimpanan kunci kustom harus berisi setidaknya satu HSM aktif. Untuk menemukan jumlah aktif HSMs di cluster, gunakan AWS KMS konsol, AWS CloudHSM konsol, atau DescribeClustersoperasi.
- AWS CloudHSM Cluster 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 `KMSInvalidStateException` pengecualian. 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 a`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 penyimpanan kunci khusus.

Pelajari selengkapnya

- Untuk mempelajari lebih lanjut tentang toko-toko AWS CloudHSM utama, lihatAWS CloudHSM toko-toko utama.
- Untuk membuat kunci KMS di toko AWS CloudHSM kunci, lihatBuat kunci KMS di toko AWS CloudHSM kunci.
- Untuk mengidentifikasi dan melihat kunci KMS di toko AWS CloudHSM kunci, lihatIdentifikasi kunci KMS di AWS CloudHSM toko-toko utama.
- Untuk menemukan kunci KMS dan bahan kunci di toko AWS CloudHSM kunci, lihatTemukan kunci KMS dan bahan kunci di toko AWS CloudHSM kunci.
- Untuk mempelajari tentang pertimbangan khusus untuk menghapus kunci KMS di toko kunci, lihat Menghapus AWS CloudHSM kunci KMS dari toko kunci. AWS CloudHSM

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 KMS tidak dapat membuat, melihat, memperbarui, atau menghapus kunci kriptografi apa pun di pengelola kunci eksternal Anda. AWS KMS tidak 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 tombol KMS untuk melakukan operasi kriptografi berikut:
  - Enkripsi
  - Dekripsi
  - GenerateDataKey
  - GenerateDataKeyWithoutPlaintext
  - ReEncrypt

  Operasi yang menghasilkan pasangan kunci data asimetris, GenerateDataKeyPairdan GenerateDataKeyPairWithoutPlaintext, tidak didukung di penyimpanan kunci khusus.
- Gunakan kunci KMS Layanan AWS yang 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::KMS::Key CloudFormation template 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.

Menggunakan kunci KMS di toko kunci eksternal

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 KMS konsol 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 petunjuk, lihat Connect dan lepaskan 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 independen dan tidak terpengaruh oleh perubahan status 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 `KMSInvalidStateException` pengecualian. Anda mungkin perlu menyambungkan kembali penyimpanan kunci eksternal atau menggunakan alat pengelola kunci eksternal 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 a`ThrottlingException`. Untuk detail tentang kuota permintaan toko kunci kustom, lihatKuota permintaan penyimpanan kunci khusus.

Pelajari selengkapnya

- Untuk mempelajari lebih lanjut tentang toko kunci eksternal, lihatToko kunci eksternal.
- Untuk mempelajari lebih lanjut tentang materi utama di toko kunci eksternal, lihatKunci eksternal.
- Untuk membuat kunci KMS di toko kunci eksternal, lihatBuat kunci KMS di toko kunci eksternal.
- Untuk mengidentifikasi dan melihat kunci KMS di penyimpanan kunci eksternal, lihatIdentifikasi kunci KMS di toko kunci eksternal.

- Untuk mempelajari tentang pertimbangan khusus untuk menghapus kunci KMS di penyimpanan kunci eksternal, lihat Menghapus kunci KMS dari penyimpanan kunci eksternal.

AWS KMS esensi kriptografi

AWS KMS dibangun di atas arsitektur crypto-agile, memungkinkan migrasi yang lancar dari satu algoritma atau mode yang disetujui ke yang lain saat standar berkembang. Kami memilih algoritma kriptografi yang kuat dan diperiksa secara luas dan memprioritaskan NIST-approved opsi untuk memastikan data pelanggan tetap terlindungi dari ancaman saat ini dan yang muncul.

Untuk informasi lebih lanjut tentang algoritma kriptografi yang disukai dan dapat diterima, lihat. Algoritme kriptografi yang didukung

Entropi dan pembangkitan bilangan acak

AWS KMS pembuatan kunci dilakukan di AWS KMS HSM. HSM menerapkan generator bilangan acak hibrida yang menggunakan NIST SP800-90A Deterministic Random Bit Generator (DRBG) CTR_DRBG using AES-2 56. Pembangkit tersebut ditempatkan dengan generator bit acak nondeterministik dengan 384-bit entropi dan diperbarui dengan entropi tambahan untuk memberikan prediksi resistansi pada setiap panggilan untuk bahan kriptografi.

Operasi kunci simetris (enkripsi saja)

Semua perintah enkripsi kunci simetris yang digunakan dalam HSMs menggunakan Standar Enkripsi Tingkat Lanjut (AES), di Mode Penghitung Galois (GCM) menggunakan kunci 256-bit. Panggilan analog untuk mendekripsi menggunakan fungsi invers.

AES-GCM adalah skema enkripsi yang diautentikasi. Selain mengenkripsi plaintext untuk menghasilkan ciphertext, AES-CGM mengomputasi tag autentikasi atas ciphertext dan data tambahan yang memerlukan autentikasi diperlukan (data autentikasi tambahan, atau AAD). Tag autentikasi membantu memastikan bahwa data berasal dari sumber yang dimaksudkan dan bahwa ciphertext dan AAD belum dimodifikasi.

Seringkali, AWS menghilangkan penyertaan AAD dalam deskripsi kami, terutama ketika mengacu pada enkripsi kunci data. Penghilangan ini tersirat oleh teks sekitarnya dalam kasus ini bahwa struktur yang akan dienkripsi dipartisi antara plaintext yang akan dienkripsi dan cleartext AAAD yang akan dilindungi.

AWS KMS menyediakan opsi bagi Anda untuk mengimpor materi kunci ke dalam alih- AWS KMS key alih mengandalkan AWS KMS untuk menghasilkan materi kunci. Bahan kunci yang diimpor ini dapat dienkripsi menggunakan RSAES-OAEP untuk melindungi kunci selama transportasi ke AWS KMS HSM. Pasangan kunci RSA dihasilkan pada AWS KMS HSM. Materi kunci yang diimpor didekripsi pada AWS KMS HSM dan dienkripsi ulang AES-GCM sebelum disimpan oleh layanan.

Operasi kunci asimetris (enkripsi, penandatanganan digital dan verifikasi tanda tangan)

AWS KMS mendukung penggunaan operasi kunci asimetris untuk enkripsi, tanda tangan digital, dan operasi perjanjian kunci. Operasi kunci asimetris bergantung pada kunci publik yang terkait secara matematis dan pasangan kunci pribadi yang dapat Anda gunakan untuk enkripsi dan dekripsi, verifikasi penandatanganan dan tanda tangan, atau memperoleh rahasia bersama. Kunci pribadi tidak pernah meninggalkan AWS KMS tanpa terenkripsi. Anda dapat menggunakan kunci publik di dalam AWS KMS dengan memanggil operasi AWS KMS API, atau mengunduh kunci publik dan menggunakannya di luar AWS KMS.

AWS KMS mendukung cipher asimetris berikut.

- RSA-OAEP (untuk enkripsi) & RSA-PSS dan RSA-PKCS - #1 -v1_5 (untuk penandatanganan dan verifikasi) - Mendukung panjang kunci RSA (dalam bit): 2048, 3072, dan 4096 untuk persyaratan keamanan yang berbeda.
- Kurva Elliptic (ECC) — Digunakan untuk penandatanganan dan verifikasi atau memperoleh rahasia bersama, tetapi tidak keduanya. Mendukung kurva ECC: NIST P256, P384, P521, SECP 256k1, Ed25519.
- ML-DSA— Digunakan untuk penandatanganan dan verifikasi. Spesifikasi ML-DSA utama yang didukung adalah: ML_DSA_44, ML_DSA_65, dan ML_DSA_87.
- SM2 (Khusus Wilayah China) — Digunakan untuk enkripsi dan dekripsi, penandatanganan dan verifikasi, atau memperoleh rahasia bersama, tetapi Anda harus memilih satu penggunaan kunci. Mendukung SM2PKE untuk enkripsi dan SM2DSA untuk penandatanganan.

Fungsi derivasi kunci

Fungsi derivasi kunci digunakan untuk mendapatkan kunci tambahan dari rahasia atau kunci awal. AWS KMS menggunakan fungsi derivasi kunci (KDF) untuk memperoleh kunci per panggilan untuk setiap enkripsi di bawah. AWS KMS key Semua operasi KDF menggunakan KDF dalam mode

counter yang menggunakan HMAC [FIPS197] dengan SHA256 [FIPS180]. Kunci turunan 256-bit digunakan AES-GCM untuk mengenkripsi atau mendekripsi data dan kunci pelanggan.

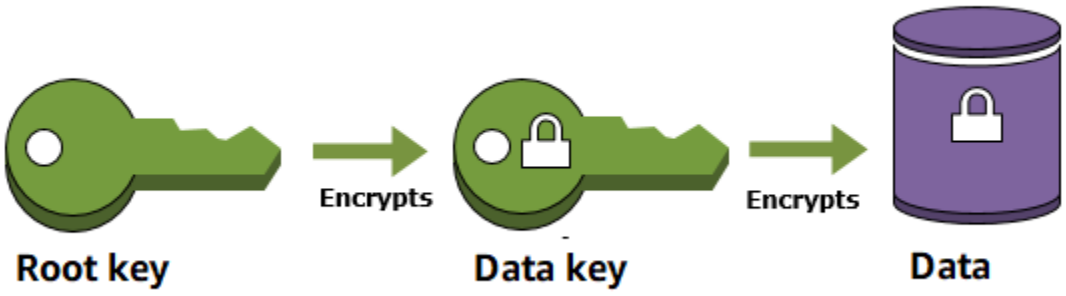
AWS KMS penggunaan internal tanda tangan digital

Tanda tangan digital juga digunakan untuk mengautentikasi perintah dan komunikasi antara entitas AWS KMS . Semua entitas layanan memiliki pasangan kunci algoritma tanda tangan digital kurva elips (ECDSA). Mereka melakukan ECDSA seperti yang didefinisikan dalam Penggunaan Algoritma Kriptografi Kurva Elliptik (ECC) dalam Sintaks Pesan Kriptografi (CMS) dan X9.62-2005: Kriptografi Kunci Pu blik untuk Industri Jasa Keuangan: Algoritma Tanda Tangan Digital Kurva Elliptik (ECDSA). Entitas menggunakan algoritma hash aman yang ditentukan dalam Publikasi Standar Pengolahan Informasi Federal, FIPS PUB 180-4, yang dikenal sebagai SHA384. Kunci dihasilkan pada kurva secp384r1 (). NIST-P384

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 KMS membantu Anda melindungi kunci enkripsi Anda dengan menyimpan dan mengelolanya dengan aman. Kunci root yang disimpan di AWS KMS, dikenal sebagai AWS KMS keys, jangan pernah meninggalkan modul keamanan perangkat keras yang AWS KMS divalidasi FIPS 140-3

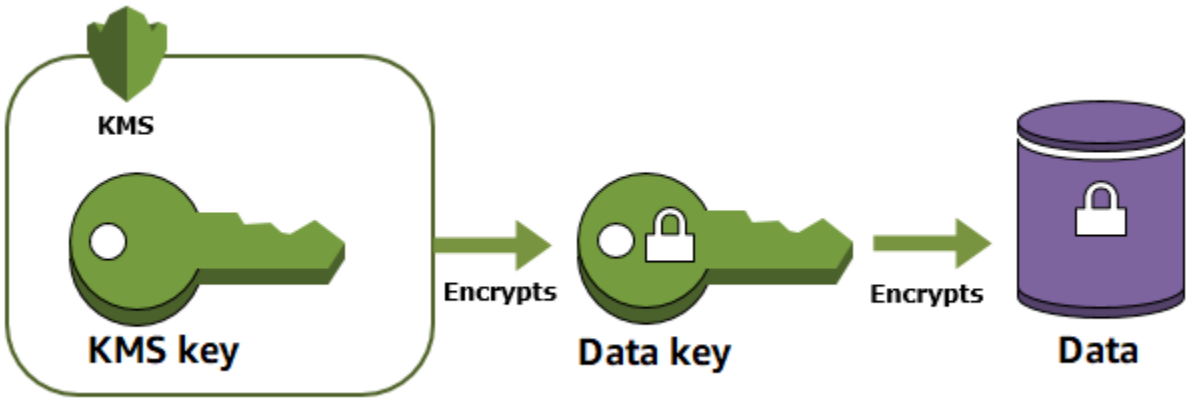
Security Level 3 tidak dienkripsi. Untuk menggunakan kunci KMS, Anda harus menelep AWS KMS on.

Konstruksi basic yang digunakan dalam banyak sistem kriptografi adalah enkripsi amplop. Enkripsi amplop menggunakan dua kunci kriptografi atau lebih untuk mengamankan pesan. Biasanya, satu kunci berasal dari kunci statis jangka panjang k, dan kunci lainnya adalah kunci per-pesan, msgKey, yang dihasilkan untuk mengenkripsi pesan. Amplop dibentuk dengan mengenkripsi pesan: ciphertext = Encrypt(MsgKey, message) . Kemudian kunci pesan dienkripsi dengan kunci statis jangka panjang: encKey = Encrypt(k, MsgKey) . Akhirnya, dua nilai (EnCKey, ciphertext) dikemas ke dalam satu struktur, atau pesan terenkripsi amplop.

Penerima, dengan akses ke k, dapat membuka pesan yang disambungkan dengan terlebih dahulu mendekripsi kunci terenkripsi dan kemudian mendekripsi pesan.

AWS KMS menyediakan kemampuan untuk mengelola kunci statis jangka panjang ini dan mengotomatiskan proses enkripsi amplop data Anda.

Selain kemampuan enkripsi yang disediakan dalam AWS KMS layanan, En AWS kripsi SDK menyediakan pustaka enkripsi amplop sisi klien. Anda dapat menggunakan perpustakaan ini untuk melindungi data Anda dan kunci enkripsi yang digunakan untuk mengenkripsi data tersebut.



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.

Operasi kriptografi

Dalam AWS KMS, operasi kriptografi adalah operasi API yang menggunakan kunci KMS untuk melindungi data. Karena kunci KMS tetap berada di dalam AWS 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. Alat AWS untuk PowerShell Anda tidak dapat melakukan operasi kriptografi di konsol AWS KMS . Untuk contoh memanggil operasi kriptografi dalam beberapa bahasa pemrograman, lihat Contoh kode untuk AWS KMS menggunakan AWS SDKs.

Tabel berikut mencantumkan operasi AWS KMS kriptografi. 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`
DeriveSharedSecret	Asimetris	`KEY_AGREEMENT`
Enkripsi	Simetris atau asimetris	`ENCRYPT_DECRYPT`
GenerateDataKey	Simetris	`ENCRYPT_DECRYPT`
GenerateDataKeyPair	Simetris [1]	`ENCRYPT_DECRYPT`

Operasi	Tipe Kunci	Penggunaan kunci
	Tidak didukung pada kunci KMS di penyimpanan kunci khusus.	
GenerateDataKeyPairWithoutPlaintext	Simetris [1] Tidak didukung pada kunci KMS di penyimpanan kunci khusus.	ENCRYPT_DECRYPT
GenerateDataKeyWithoutPlaintext	Simetris	ENCRYPT_DECRYPT
GenerateMac	HMAC	GENERATE_ VERIFY_MAC
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 pasangan kunci data 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, tetapkan AWS KMS kuota pada jumlah operasi kriptografi yang dipanggil setiap detik. Lihat perinciannya di the section called “Kuota bersama untuk operasi kriptografis”.

Algoritme kriptografi yang didukung

AWS KMS mendukung algoritma kriptografi berikut untuk kunci KMS. Algoritma yang dapat Anda gunakan bergantung pada spesifikasi kunci dan penggunaan kunci kunci KMS. Untuk deskripsi rinci dari setiap spesifikasi kunci dan algoritma yang didukungnya, lihat. Referensi spesifikasi kunci

Untuk panduan tentang algoritma mana yang akan digunakan dan kapan, lihat Algoritma kriptografi dan. Layanan AWS

Algoritma kunci simetris

AWS KMS mendukung algoritma berikut untuk kunci KMS simetris.

Algoritma yang didukung untuk kunci KMS simetris

Algoritme	Penggunaan kunci	Spesifikasi kunci
AES-256-GCM	Enkripsi dan dekripsi	SYMMETRIC_DEFAULT
SM4-128 (Hanya Wilayah Tiongkok)	Enkripsi dan dekripsi	SYMMETRIC_DEFAULT
HMAC_SHA_224	Hasilkan dan verifikasi MAC	HMAC_224
HMAC_SHA_256	Hasilkan dan verifikasi MAC	HMAC_256
HMAC_SHA_384	Hasilkan dan verifikasi MAC	HMAC_384
HMAC_SHA_512	Hasilkan dan verifikasi MAC	HMAC_512

Algoritma kunci asimetris

Setiap kunci KMS asimetris memiliki penggunaan kunci tunggal yang menentukan algoritma mana yang dapat Anda gunakan.

Algoritma yang didukung untuk kunci KMS asimetris

Algoritme	Penggunaan kunci	Spesifikasi kunci
RSAES_OAEP_SHA_1, RSAES_OAEP_SHA_256	Enkripsi dan dekripsi	RSA_2048, RSA_3072, RSA_4096
RSASSA_PSS_SHA_256, RSASSA_PSS_SHA_384, RSASSA_PSS_SHA_512, RSASSA_PKCS1_V1_5_SHA_256, RSASSA_PKCS1_V1_5_SHA_384, RSASSA_PKCS1_V1_5_SHA_512	Tanda tangan dan verifikasi	RSA_2048, RSA_3072, RSA_4096
ECDSA_SHA_256	Tanda tangan dan verifikasi	ECC_NIST_P256 (secp256r1)
ECDSA_SHA_384	Tanda tangan dan verifikasi	ECC_NIST_P384 (secp384r1)
ECDSA_SHA_512	Tanda tangan dan verifikasi	ECC_NIST_P521 (secp521r1)
ECDSA_SHA_256	Tanda tangan dan verifikasi	ECC_SECG_P256K1 (secp256k1)
ED25519_SHA_512, ED25519_PH_SHA_512	Tanda tangan dan verifikasi	ECC_NIST_EDWARDS25519 (ed25519)
ML_DSA_SHAKE_256	Tanda tangan dan verifikasi	ML_DSA_44, ML_DSA_65, ML_DSA_87
ECDH	Perjanjian kunci	ECC_NIST_P256, ECC_NIST_P384, ECC_NIST_P521
SM2PKE (enkripsi), SM2DSA (penandatanganan), ECDH (perjanjian kunci)	Mengenkripsi dan mendekripsi, menandatangani dan	SM2 (Hanya Wilayah Tiongkok)

Algoritme	Penggunaan kunci	Spesifikasi kunci
	memverifikasi, atau perjanjian kunci	

Akses dan izin kunci KMS

Untuk menggunakannya AWS KMS, Anda harus memiliki kredensi yang AWS dapat digunakan untuk mengautentikasi permintaan Anda. Kredensialnya harus menyertakan izin untuk mengakses AWS sumber daya: AWS KMS keys dan 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.

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.

Kebijakan utama 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 KMS kebijakan sumber daya untuk kunci KMS disebut kebijakan kunci.

Semua kunci KMS memiliki kebijakan kunci. Jika Anda tidak menyediakannya, AWS KMS buatlah satu untuk Anda. Kebijakan kunci default yang AWS KMS digunakan berbeda-beda tergantung pada apakah Anda membuat kunci di AWS KMS konsol atau Anda menggunakan AWS KMS API. Kami menyarankan Anda mengedit kebijakan kunci default agar selaras dengan persyaratan organisasi Anda untuk izin hak istimewa paling sedikit.

Anda dapat menggunakan kebijakan kunci sendiri untuk mengontrol akses jika kunci dan prinsipal IAM berada di AWS akun yang sama, yang berarti cakupan penuh akses ke kunci KMS didefinisikan dalam satu dokumen (kebijakan kunci). Namun, ketika penelepon di satu akun harus mengakses kunci di akun yang berbeda, Anda tidak dapat menggunakan kebijakan kunci sendiri untuk memberikan akses. Dalam skenario lintas akun, kebijakan IAM harus dilampirkan ke pengguna atau peran pemanggil yang secara eksplisit memungkinkan pemanggil untuk melakukan panggilan API.

Anda juga dapat menggunakan kebijakan IAM dalam kombinasi dengan kebijakan utama dan hibah untuk mengontrol akses ke kunci KMS. Untuk menggunakan kebijakan IAM untuk mengontrol akses ke kunci KMS, kebijakan kunci harus memberikan izin akun untuk menggunakan kebijakan IAM. Anda dapat menentukan pernyataan kebijakan kunci yang mengaktifkan kebijakan IAM, atau Anda dapat secara eksplisit menentukan prinsip yang diizinkan dalam kebijakan utama.

Saat menulis kebijakan, pastikan Anda memiliki kontrol kuat yang membatasi siapa yang dapat melakukan tindakan berikut:

- Memperbarui, membuat, dan menghapus kebijakan kunci IAM dan KMS
- Lampirkan dan lepaskan kebijakan IAM dari pengguna, peran, dan grup
- Pasang dan lepaskan kebijakan kunci KMS dari kunci KMS Anda

Hibah kunci KMS

Selain IAM dan kebijakan utama, AWS KMS mendukung hibah. Hibah menyediakan cara yang fleksibel dan ampuh untuk mengelola prinsipal IAM atau kepala AWS layanan mana yang dapat mengakses kunci tanpa memodifikasi kebijakan utama. Ini dapat mencakup kasus penggunaan di mana izin hanya diperlukan selama beberapa pekerjaan atau sumber daya lain. Anda memanggil API untuk membuat hibah dan untuk menghentikan atau mencabut hibah ketika tidak lagi diperlukan.

Lihat informasi yang lebih lengkap di Hibah di AWS KMS.

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 diMemungkinkan akses ke Akun AWS dan memungkinkan kebijakan IAM.

Tidak seperti kebijakan IAM, yang bersifat global, kebijakan kunci bersifat Regional. Kebijakan utama 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 utama
- Mengubah kebijakan utama
- Izin untuk AWS layanan dalam kebijakan utama

Membuat kebijakan utama

Anda dapat membuat dan mengelola kebijakan utama di AWS KMS konsol atau dengan menggunakan operasi AWS KMS API, seperti CreateKey, ReplicateKey, dan PutKeyPolicy.

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.

JSON

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

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 Latin Dasar dan Latin-1 Suplemen
- 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.

Principal

(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.

> **Note**
>
> Jika `Action` elemen yang diperlukan hilang dari pernyataan kebijakan kunci, pernyataan kebijakan tidak berpengaruh. Pernyataan kebijakan kunci tanpa `Action` elemen tidak berlaku untuk kunci KMS apa pun.
> Ketika pernyataan kebijakan kunci kehilangan `Action` elemennya, AWS KMS konsol melaporkan kesalahan dengan benar, tetapi PutKeyPolicyAPI CreateKeydan berhasil, meskipun pernyataan kebijakan tidak efektif.

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.

Kondisi

(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.

JSON

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

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

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 digunakan berbeda-beda 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 Alat AWS untuk 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 Memungkinkan akses ke Akun AWS dan memungkinkan kebijakan IAM.

Kebijakan kunci default saat Anda membuat kunci KMS dengan Konsol Manajemen AWS

Saat Anda membuat kunci KMS dengan Konsol Manajemen AWS, 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, peran IAM, dan Akun AWS siapa administrator kunci dan mereka yang merupakan pengguna kunci (atau keduanya).

Izin

- Memungkinkan akses ke Akun AWS dan memungkinkan 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 AWS layanan

Memungkinkan akses ke Akun AWS dan memungkinkan 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 utama yang ditunjukkan di atas memberikan izin pokok akun untuk mengontrol kunci. Prinsipal akun 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

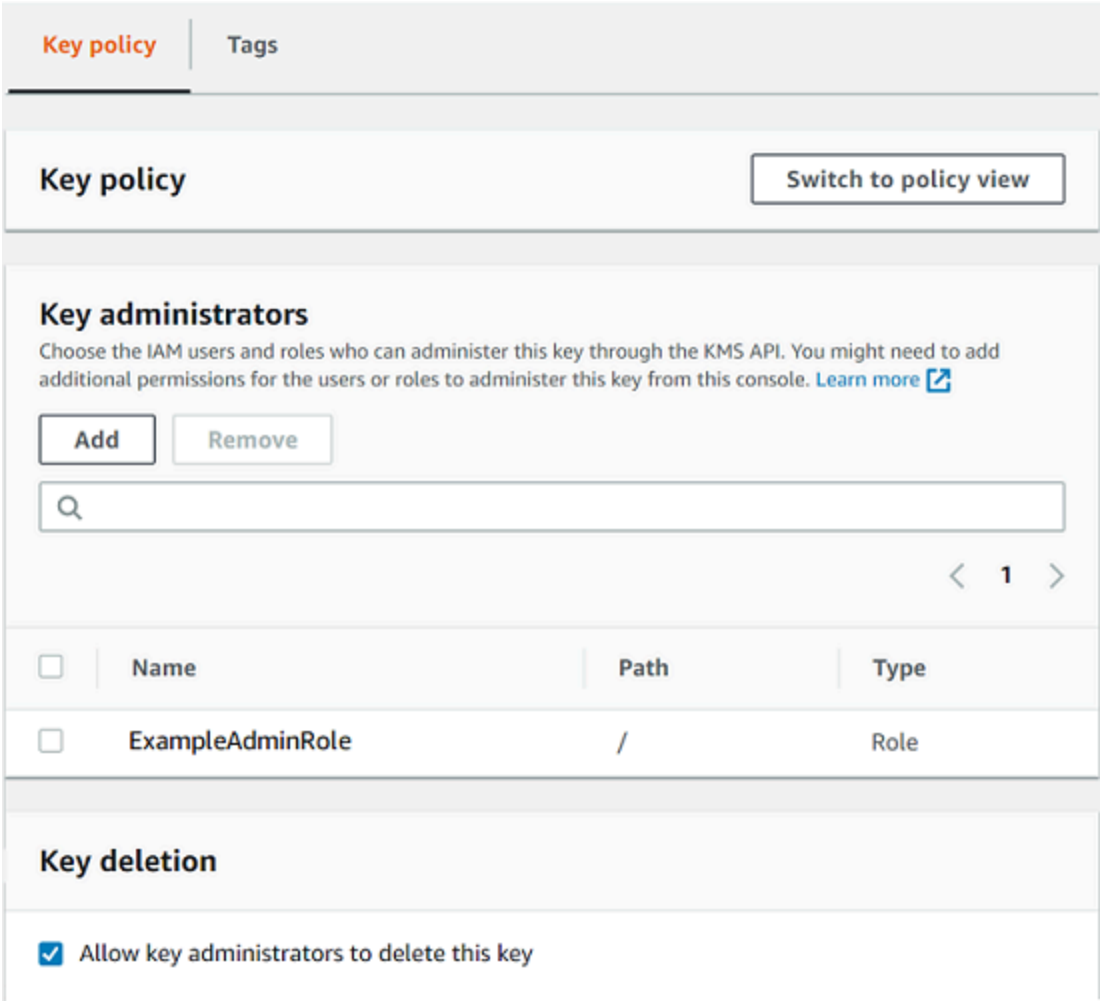
Kebijakan kunci default yang dibuat oleh konsol tersebut mengizinkan Anda memilih pengguna IAM dan peran dalam akun dan membuatnya administrator kunci. Pernyataan ini disebut pernyataan administrator kunci. Administrator kunci memiliki izin untuk mengelola kunci KMS, tetapi tidak memiliki izin untuk 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.

> **Note**
>
> AWS KMS Konsol menambahkan administrator kunci ke kebijakan kunci di bawah pengenal `"Allow access for Key Administrators"` pernyataan. Memodifikasi pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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

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 Anda 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. Konsol Manajemen AWS

`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 Konsol Manajemen AWS, 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 bahkan jika 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. Konsol Manajemen AWS

`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, hapus 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, hapus opsi Izinkan administrator kunci untuk menghapus opsi kunci ini.

`kms:RotateKeyOnDemand`

Memungkinkan `kms:RotateKeyOnDemand`, yang memungkinkan administrator kunci untuk melakukan rotasi sesuai permintaan dari materi utama dalam kunci KMS 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.

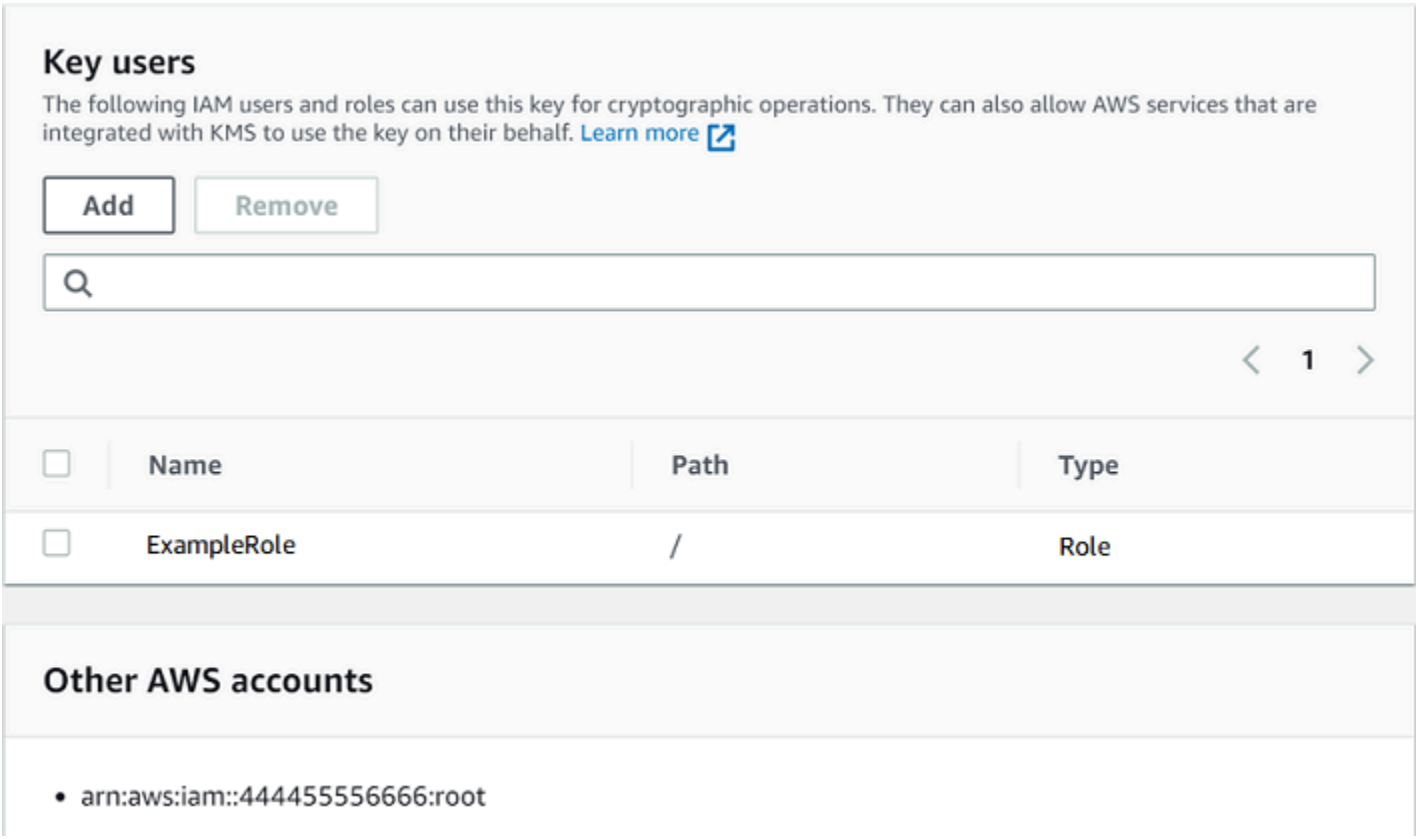
Konsol tersebut menambahkan dua pernyataan kebijakan untuk kebijakan kunci bagi pengguna kunci.

- 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 menggunakan kunci KMS, lihat[Memungkinkan 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 akan menambahkan pengguna dan peran yang Anda tentukan ke `Principal` elemen di setiap pernyataan pengguna kunci.

> **Note**
>
> AWS KMS Konsol menambahkan pengguna kunci ke kebijakan kunci di bawah pengidentifikasi pernyataan `"Allow use of the key"` dan`"Allow attachment of persistent resources"`. Memodifikasi pengidentifikasi pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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

Kunci KMS asimetris untuk mendapatkan rahasia bersama

Konsol menambahkan pernyataan berikut ke kebijakan kunci untuk kunci KMS asimetris dengan penggunaan kunci perjanjian Kunci.

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

Memungkinkan pengguna kunci untuk mendapatkan rahasia bersama 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 untuk 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 AWS layanan

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 GrantIsForAWSResource 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 dengan 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 Bagaimana 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 utama

Anda dapat melihat kebijakan kunci untuk kunci yang dikelola AWS KMS pelanggan atau Kunci yang dikelola AWSdi akun Anda dengan menggunakan AWS KMS konsol atau GetKeyPolicyoperasi di AWS KMS API. Anda tidak dapat menggunakan teknik ini untuk melihat kebijakan kunci kunci KMS secara berbeda Akun AWS.

Untuk mempelajari lebih lanjut tentang kebijakan AWS KMS utama, lihatKebijakan utama di AWS KMS. Untuk mempelajari cara menentukan pengguna dan peran mana yang memiliki akses ke kunci KMS, lihatthe section called “Menentukan akses”.

Menggunakan AWS KMS konsol

Pengguna yang berwenang dapat melihat kebijakan kunci untuk kunci terkelola Kunci yang dikelola AWSatau pelanggan pada tab Kebijakan kunci pada Konsol Manajemen AWS.

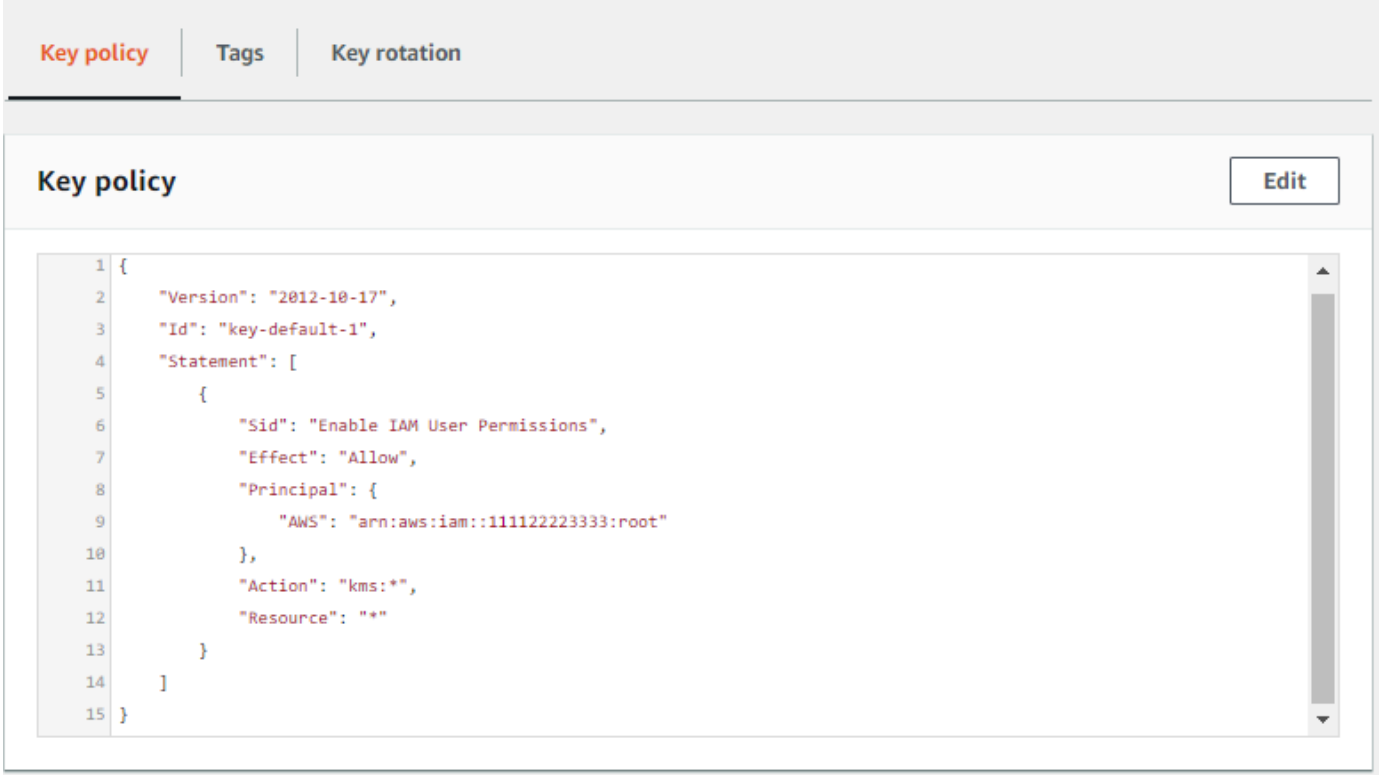
Untuk melihat kebijakan kunci untuk kunci KMS di Konsol Manajemen AWS, Anda harus memiliki izin kms:, kms: ListAliasesDescribeKey, dan kms:. GetKeyPolicy

1. Masuk ke Konsol Manajemen AWS 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 Anda yang AWS membuat dan mengelola untuk Anda, di panel navigasi, pilih kunci AWS terkelola. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih CMK.
4. Dalam daftar kunci KMS, pilih alias atau ID kunci dari kunci KMS yang ingin Anda periksa.
5. Pilih tab Kebijakan kunci.

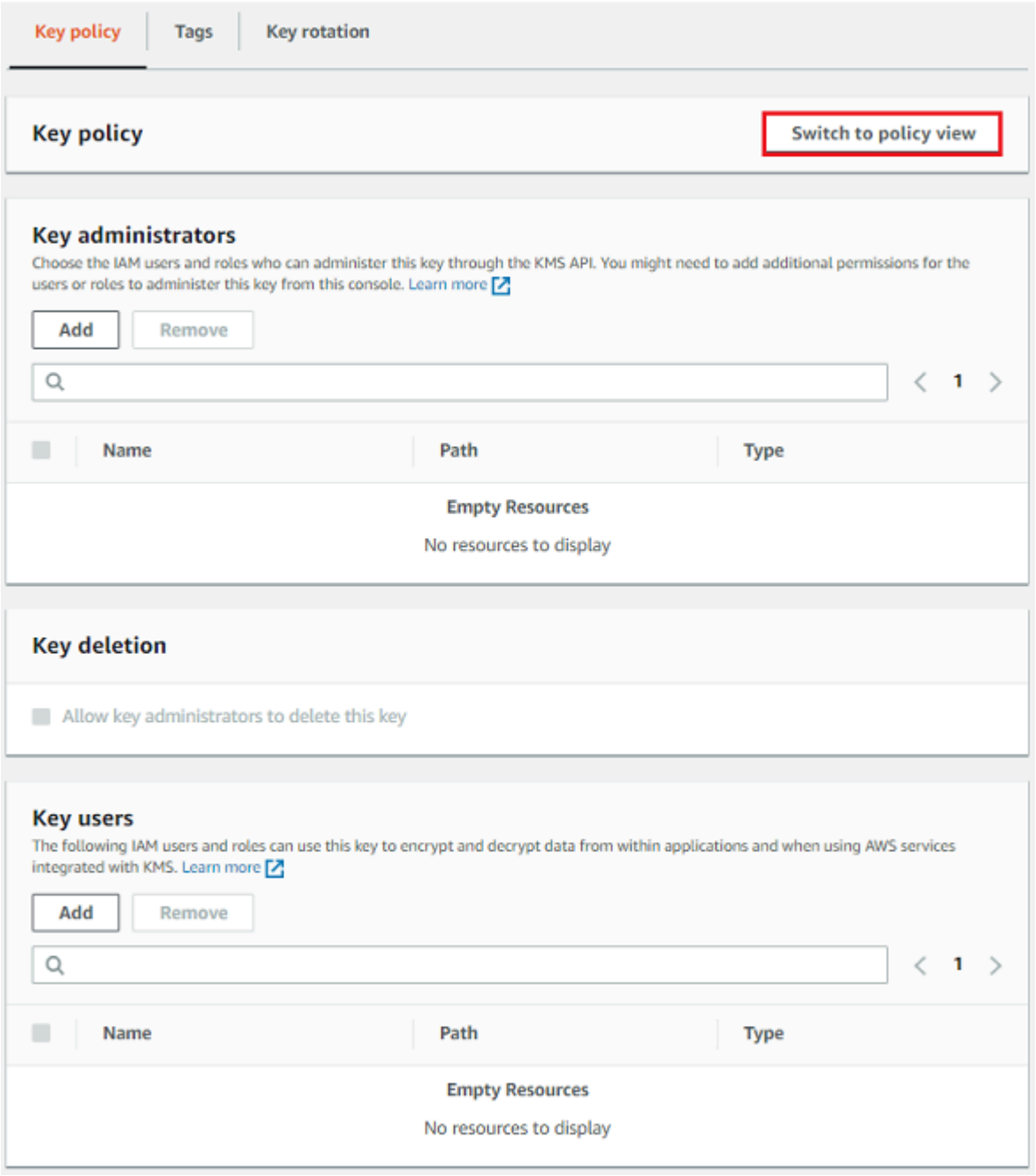
   Pada tab Kebijakan kunci, Anda mungkin melihat dokumen kebijakan kunci. Ini adalah tampilan kebijakan. Dalam pernyataan kebijakan utama, Anda dapat melihat kepala sekolah yang telah diberi akses ke kunci KMS oleh kebijakan utama, dan Anda dapat melihat tindakan yang dapat mereka lakukan.

   Contoh berikut menunjukkan tampilan kebijakan untuk kebijakan kunci default.



   Atau, jika Anda membuat kunci KMS di Konsol Manajemen AWS, 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.



Menggunakan AWS KMS API

Untuk mendapatkan kebijakan kunci untuk kunci KMS di Anda Akun AWS, gunakan GetKeyPolicyoperasi di AWS KMS API. Anda tidak dapat menggunakan operasi ini untuk melihat kebijakan kunci di akun yang berbeda.

Contoh berikut menggunakan perintah get-key-policy 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.

JSON

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

Mengubah kebijakan utama

Anda dapat mengubah kebijakan kunci untuk kunci KMS di Anda Akun AWS dengan menggunakan Konsol Manajemen AWS atau PutKeyPolicyoperasi. Anda tidak dapat menggunakan teknik ini untuk mengubah kebijakan kunci kunci KMS dengan cara yang berbeda Akun 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 eksternal Akun AWS ke kebijakan utama, Anda juga harus menggunakan kebijakan IAM di akun eksternal untuk memberikan izin kepada pengguna, grup, atau peran 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).

Cara mengubah kebijakan kunci

Anda dapat mengubah kebijakan kunci dalam tiga cara berbeda seperti yang dijelaskan di bagian berikut.

Topik

- Menggunakan tampilan Konsol Manajemen AWS default
- Menggunakan tampilan Konsol Manajemen AWS kebijakan
- Menggunakan AWS KMS API

Menggunakan tampilan Konsol Manajemen AWS default

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 Konsol Manajemen AWS kebijakan atau Menggunakan AWS KMS API.

1. Lihat kebijakan kunci untuk kunci terkelola pelanggan seperti yang dijelaskan dalamMenggunakan AWS KMS konsol. (Anda tidak dapat mengubah kebijakan utama Kunci 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 Konsol Manajemen AWS kebijakan

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 dalamMenggunakan AWS KMS konsol. (Anda tidak dapat mengubah kebijakan utama Kunci yang dikelola AWS.)
2. Di bagian Kebijakan Kunci, pilih Beralih ke tampilan kebijakan.
3. Pilih Edit.
4. Menentukan hal yang harus diubah.

   - Untuk menambahkan pernyataan baru, pilih Tambahkan pernyataan baru. Kemudian, Anda dapat memilih tindakan, prinsip, dan kondisi untuk pernyataan kebijakan kunci baru Anda dari opsi yang tercantum di panel pembuat pernyataan, atau memasukkan elemen pernyataan kebijakan secara manual.
   - Untuk menghapus pernyataan dari kebijakan utama Anda, pilih pernyataan, lalu pilih Hapus. Tinjau pernyataan kebijakan yang dipilih dan konfirmasikan bahwa Anda ingin menghapusnya. Jika Anda memutuskan bahwa Anda tidak ingin melanjutkan dengan menghapus pernyataan, pilih Batal.

- Untuk mengedit pernyataan kebijakan kunci yang ada, pilih pernyataan tersebut. Kemudian, Anda dapat menggunakan panel pembuat pernyataan untuk memilih elemen tertentu yang ingin Anda modifikasi, atau mengedit pernyataan secara manual.

5. Pilih Simpan perubahan.

Menggunakan AWS KMS API

Anda dapat menggunakan PutKeyPolicyoperasi untuk mengubah kebijakan kunci kunci KMS di Anda Akun AWS. Anda tidak dapat menggunakan API ini pada kunci KMS yang berbeda Akun 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 Gunakan `GetKeyPolicy` dengan AWS SDK atau CLI.
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 Gunakan `PutKeyPolicy` dengan AWS SDK atau CLI.

Untuk contoh menyalin kebijakan kunci dari satu kunci KMS ke kunci lainnya, lihat GetKeyPolicy contoh di AWS CLI Command Reference.

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 terkelola pelanggan, lihat dokumentasi enkripsi untuk layanan tersebut. Daftar berikut mencakup tautan ke beberapa dokumentasi layanan:

- AWS CloudTrailizin - Konfigurasikan kebijakan AWS KMS utama untuk CloudTrail
- Izin Amazon Elastic Block Store - Panduan Pengguna Amazon EC2 dan Panduan Pengguna Amazon EC2
- AWS Lambdaizin - Enkripsi data saat istirahat untuk Lambda
- Izin Amazon Q - Enkripsi data untuk Amazon Q
- Izin Layanan Amazon Relational Database Service - manajemen AWS KMS kunci
- AWS Secrets Managerizin - Otorisasi penggunaan kunci KMS
- Izin Layanan Antrian Sederhana Amazon - Manajemen Kunci Amazon SQS

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 kebijakan titik akhir VPC.

Untuk mendapatkan bantuan mengenai cara menulis dan memformat dokumen kebijakan JSON, lihat Referensi Kebijakan IAM JSON dalam Panduan Pengguna 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.

JSON

```
{
  "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.

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 tentang cara kerja kebijakan AWS KMS utama dan kebijakan IAM, lihatPemecahan Masalah AWS KMS 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 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.

JSON

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

```
      "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn:aws:kms:us-
west-2:111122223333:key/01234abcd-12ab-34cd-56ef-1234567890ab"
    ]
  }
}
```

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.

JSON

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

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.

JSON

```
{
  "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 hanya memungkinkan operasi dan operasi read-only yang kurang sensitif yang tidak berlaku untuk kunci KMS tertentu.

JSON

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

JSON

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

```
}
```

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

JSON

```
{
  "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` 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.

JSON

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

```
    }
  ]
}
```

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

JSON

```
{
  "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).

JSON

```
{
  "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.

JSON

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

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 Pemecahan Masalah AWS KMS izin.

JSON

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

Kebijakan pengendalian sumber daya di AWS KMS

Kebijakan pengendalian sumber daya (RCPs) adalah jenis kebijakan organisasi yang dapat Anda gunakan untuk menegakkan kontrol preventif pada AWS sumber daya di organisasi Anda. RCPs membantu Anda membatasi akses eksternal ke AWS sumber daya Anda secara terpusat dalam skala besar. RCPs melengkapi kebijakan kontrol layanan (SCPs). Sementara, SCPs dapat digunakan untuk mengatur izin maksimum secara terpusat pada peran IAM dan pengguna di organisasi Anda, RCPs dapat digunakan untuk secara terpusat menetapkan izin maksimum pada AWS sumber daya di organisasi Anda.

Anda dapat menggunakan RCPs untuk mengelola izin ke kunci KMS yang dikelola pelanggan di organisasi Anda. RCPs saja tidak cukup dalam memberikan izin ke kunci yang dikelola pelanggan Anda. Tidak ada izin yang diberikan oleh RCP. RCP mendefinisikan pagar pembatas izin, atau menetapkan batasan, pada tindakan yang dapat diambil identitas terhadap sumber daya di akun yang terpengaruh. Administrator harus tetap melampirkan kebijakan berbasis identitas ke peran IAM atau pengguna, atau kebijakan utama untuk benar-benar memberikan izin.

> **Note**
>
> Kebijakan pengendalian sumber daya di organisasi Anda tidak berlaku Kunci yang dikelola AWS.

Kunci yang dikelola AWS dibuat, dikelola, dan digunakan atas nama Anda oleh suatu AWS layanan, Anda tidak dapat mengubah atau mengelola izin mereka.

Pelajari selengkapnya

- Untuk informasi lebih umum tentang RCPs, lihat Kebijakan kontrol sumber daya di Panduan AWS Organizations Pengguna.
- Untuk detail tentang cara mendefinisikan RCPs, termasuk contoh, lihat sintaks RCP di AWS Organizations Panduan Pengguna.

Contoh berikut menunjukkan cara menggunakan RCP untuk mencegah prinsipal eksternal mengakses kunci yang dikelola pelanggan di organisasi Anda. Kebijakan ini hanyalah contoh, dan Anda harus menyesuaikannya untuk memenuhi kebutuhan bisnis dan keamanan unik Anda. Misalnya, Anda mungkin ingin menyesuaikan kebijakan Anda untuk mengizinkan akses oleh mitra bisnis Anda. Untuk detail selengkapnya, lihat repositori contoh kebijakan perimeter data.

Note

`kms:RetireGrant`Izin tidak efektif dalam RCP, bahkan jika `Action` elemen menentukan tanda bintang (*) sebagai wildcard.
Untuk informasi selengkapnya tentang cara izin `kms:RetireGrant` ditentukan, lihatMenghentikan dan mencabut pemberian izin.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "RCPEnforceIdentityPerimeter",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "kms:*",
            "Resource": "*",
            "Condition": {
                "StringNotEqualsIfExists": {
```

```
                    "aws:PrincipalOrgID": "my-org-id"
                },
                "Bool": {
                    "aws:PrincipalIsAWSService": "false"
                }
            }
        }
    ]
}
```

Contoh RCP berikut mengharuskan prinsipal AWS layanan hanya dapat mengakses kunci KMS yang dikelola pelanggan Anda saat permintaan berasal dari organisasi Anda. Kebijakan ini menerapkan kontrol hanya pada permintaan yang `aws:SourceAccount` ada. Ini memastikan bahwa integrasi layanan yang tidak memerlukan penggunaan `aws:SourceAccount` tidak terpengaruh. Jika `aws:SourceAccount` ada dalam konteks permintaan, `Null` kondisi akan dievaluasi`true`, menyebabkan `aws:SourceOrgID` kunci ditegakkan.

Untuk informasi lebih lanjut tentang masalah wakil yang bingung, lihat Masalah wakil yang bingung di Panduan Pengguna IAM.

JSON

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

```
                    "aws:SourceAccount": "false"
                }
            }
        }
    ]
}
```

Hibah di AWS KMS

Hi bah adalah instrumen kebijakan yang memungkinkan kepala sekolah atau AWS prinsipal AWS layanan untuk menggunakan kunci KMS dalam operasi kriptografi. 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.

Hibah biasanya digunakan oleh AWS layanan yang terintegrasi dengan AWS KMS untuk mengenkripsi data Anda saat istirahat. Layanan akan membuat pemberian izin atas nama pengguna di akun, menggunakan izinnya, dan menghentikan pemberian izin begitu tugasnya selesai. Untuk detail tentang cara AWS layanan, menggunakan hibah, lihat topik Enkripsi saat istirahat di panduan pengguna layanan atau panduan pengembang.

Pemberian izin adalah mekanisme kontrol akses yang sangat fleksibel dan berguna. Saat Anda membuat hibah untuk kunci KMS, hibah memungkinkan prinsipal 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 dengan cara yang berbeda Akun AWS.
- Hibah dapat mengizinkan akses ke kunci KMS, tetapi tidak menolak akses.
- Setiap hibah memiliki satu penerima hibah, yang dapat berupa prinsipal penerima hi bah (identitas AWS IAM) atau prinsipal layanan penerima hibah (prinsipal layanan). AWS Anda dapat menentukan `GranteePrincipal` atau `GranteeServicePrincipal`, tetapi bukan keduanya.
- Hibah hanya dapat mengizinkan operasi hibah. Operasi hibah harus didukung oleh kunci KMS dalam hibah. Jika Anda menentukan operasi yang tidak didukung, CreateGrant permintaan gagal dengan peng `ValidationError` ecualian.

- Kepala sekolah penerima hibah dapat menggunakan izin yang diberikan hibah kepada mereka tanpa menentukan hibah, sama seperti yang mereka lakukan jika izin berasal dari kebijakan utama atau kebijakan IAM. Namun, karena AWS KMS API mengikuti model konsist ensi 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 harus ditambahkan atau dihapus untuk membatalkan hibah.
- AWS KMS membatasi jumlah hibah pada setiap kunci KMS. Untuk detailnya, lihat Hi Hibah per kunci KMS: 50.000 kunci KMS.

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 mengiz inkan PutKeyPolicy izin kms: untuk menetapkan kebijakan.

- Pengguna dengan izin untuk membuat hibah untuk kunci KMS (`kms:CreateGrant`) dapat menggunakan hibah untuk mengizinkan pengguna dan peran, termasuk AWS layanan, menggunakan kunci KMS. Prinsipal dapat berupa identitas Anda sendiri Akun AWS atau identitas di akun atau organisasi yang berbeda.
- Hibah hanya dapat mengizinkan sebagian AWS KMS operasi. Anda dapat menggunakan hibah untuk mengizinkan kepala sekolah melihat kunci KMS, menggunakannya dalam operasi kriptografi, dan membuat dan menghentikan hibah. Untuk detailnya, lihat Operasi pemberian izin. Anda juga dapat menggunakan batas pemberian izin untuk membatasi izin dalam pemberian izin.
- 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 hi bah apa pun pada kunci KMS. Para kepala sekolah 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. Prinsipal ini hanya dapat mendelegasikan izin yang diberikan kepada mereka, bahkan jika mereka memiliki izin lain dari kebijakan. Lihat perinciannya di Memberikan CreateGrant izin.

Konsep hibah

Untuk menggunakan pemberian izin secara efektif, Anda harus memahami istilah dan konsep yang digunakan AWS KMS .

Batas pemberian izin

Kondisi yang membatasi izin dalam hibah. AWS KMS mendukung dua jenis kendala hibah:

- Kendala konteks enkripsi — Membatasi izin berdasarkan konteks enkripsi yang diser takan dalam permintaan untuk operasi kriptografi. Kendala ini hanya berfungsi dengan kunci KMS enkripsi simetris.
- SourceArn batasan — Membatasi izin pemberian untuk permintaan yang dibuat atas nama sumber daya tertentu AWS . Ini secara efektif menempatkan kunci kondisi aws:SourceArn global ke dalam hibah. Kendala ini bekerja dengan semua jenis kunci KMS.

Lihat perinciannya di 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 RevokeGrant permintaan RetireGrant atau.

Operasi pemberian izin

AWS KMS Operasi yang dapat Anda izinkan dalam hibah. Jika Anda menentukan operasi lain, CreateGrant permintaan gagal dengan peng `ValidationError` ecualian. 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
  - DeriveSharedSecret
  - 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, CreateGrant permintaan gagal dengan peng `ValidationError` ecualian. Misalnya, hibah untuk kunci KMS enkripsi simetris tidak dapat mengizinkan Penandatang anan, Verifikasi, `GenerateMac` atau `VerifyMac` operasi. Hibah untuk kunci KMS asimetris tidak dapat mengizinkan operasi apa pun yang menghasilkan kunci data atau pasangan kunci data.

Token izin

  AWS KMS API mengikuti model konsistensi akhirnya. Saat Anda membuat 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 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, berdurasi variabel, yang dikodekan 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 diperbanyak sepenuhnya AWS KMS. Setelah itu, perwakilan penerima dapat menggunakan izin dalam pemberian izin tanpa memberikan token izin atau bukti izin lain. Anda dapat menggunakan token hibah kapan saja, tetapi setelah hibah pada akhirnya konsisten, AWS KMS gunakan hibah untuk menentukan izin, bukan token hibah.

Misalnya, perintah berikut memanggil GenerateDataKey operasi. Ini menggunakan token hibah untuk mewakili hibah yang memberikan izin kepada pemanggil (prinsipal penerima hibah) untuk memanggil kunci KMS `GenerateDataKey` 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 RetireGrant operasi.

```
$ 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 CloudTrail peristiwa log untuk CreateGrant operasi tersebut. ListRetirableGrantsOper ListGrants asi dan mengembalikan ID hibah, tetapi bukan token hibah.

Lihat perinciannya di Menggunakan token izin.

Perwakilan penerima

Prinsi AWS pal (identitas IAM) yang mendapatkan izin yang ditentukan dalam hibah.

Prinsip penerima hibah dapat berupa prinsipal AWS apa pun, termasuk Akun AWS (root), pengguna IAM, peran IAM, peran atau pengguna federasi, atau pengguna peran yang diasum sikan. Prinsip 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 AWS organisasi.

Untuk menentukan prinsipal penerima hibah, gunakan `GranteePrincipal` parameter dalam permintaan. CreateGrant

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

Kepala layanan penerima hibah

Prinsi AWS p layanan yang mendapatkan izin yang ditentukan dalam hibah. Untuk menentukan prinsipal layanan penerima hibah, gunakan `GranteeServicePrincipal` parameter dalam permintaan. CreateGrant

Saat Anda membuat hibah dengan a`GranteeServicePrincipal`, Anda juga harus menyertakan a `SourceArn grant constraint`. Ken `SourceArn` dala memastikan bahwa prinsipal layanan dapat menggunakan kunci KMS hanya ketika permintaan dibuat atas nama sumber daya yang ditentukan AWS .

Ketika Anda menentukan a`GranteeServicePrincipal`, Anda juga harus menentukan a `RetiringPrincipal` atau a `RetiringServicePrincipal`.

> **Note**
>
> Anda dapat menentukan `GranteePrincipal` atau `GranteeServicePrincipal`, tetapi bukan keduanya.

Penghentian (pemberian izin)

Mengakhiri pemberian izin. Anda menghentikan hibah setelah 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. Prinsip pensiun dapat berupa prinsipal apa pun AWS , termasuk, pengguna IAM Akun AWS, peran IAM, pengguna federasi, dan pengguna peran yang diasumsikan. Prinsip pensiun dapat berada di akun yang sama dengan kunci KMS atau akun yang berbeda.

Untuk menentukan prinsipal pensiun, gunakan `RetiringPrincipal` parameter dalam CreateGrant permintaan.

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

Kepala layanan pensiun

Kepala AWS sekolah layanan yang memiliki izin untuk pensiun hibah. Untuk menentukan prinsipal layanan pensiun, gunakan `RetiringServicePrincipal` parameter dalam CreateGrant permintaan.

> **Note**
>
> Anda dapat menentukan salah satu dari `RetiringPrincipal` atau `RetiringServicePrincipal`, bukan keduanya.

Untuk informasi lebih lanjut tentang siapa yang dapat pensiun hibah, lihatMenghentikan 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 KMS API mengikuti model konsistensi akhirnya. Saat Anda membuat, menghentikan, 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 hibah baru atau menggunakan izin dalam hibah baru sebelum hibah diketahui secara keseluruhan AWS KMS, Anda mungkin mendapatkan 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 tipikal adalah mencoba kembali permintaan, dan beberapa AWS SDK menyertakan logika backoff dan coba ulang otomatis.

AWS KMS memiliki fitur untuk 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 petunjuk, lihat Menggunakan token izin.
- CreateGrantOperasi memiliki `Name` parameter yang mencegah operasi percobaan ulang 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 AWS KMS konsistensi akhir.

Praktik terbaik untuk AWS KMS hadiah

AWS KMS merekomendasikan praktik terbaik berikut saat membuat, menggunakan, dan mengelola hibah.

- Batasi izin dalam hibah untuk yang dibutuhkan penerima hibah. Gunakan prinsip akses yang paling tidak istimewa. Jadilah sespesifik mungkin dalam semua parameter hibah:
  - Tentukan hanya operasi hibah yang dibutuhkan penerima hibah.
  - Gunakan prinsip penerima hibah tertentu, seperti peran IAM, atau prinsipal layanan penerima hibah tertentu.
  - Gunakan batasan hibah untuk membatasi hibah lebih lanjut.
- Gunakan batasan pemberian konteks enkripsi untuk memastikan bahwa pemanggil menggunakan kunci KMS untuk tujuan yang dimaksudkan. 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 AWS Keamanan.

  > **Tip**
  >
  > Gunakan batasan EncryptionContextEqual hibah bila memungkinkan. Kendala EncryptionContextSubset hibah lebih sulit digunakan dengan benar. Jika Anda perlu menggunakannya, baca dokumentasi dengan seksama dan uji batasan hibah untuk memastikannya berfungsi sebagaimana dimaksud.

- Saat membuat hibah untuk AWS layanan yang didukung, gunakan `SourceArn grant constraint` untuk membatasi hibah ke sumber daya tertentu.
- Waspadai hibah duplikat. Hibah duplikat memiliki parameter yang sama kecuali untuk nama hibah. Duplikasi yang tidak perlu dapat menyebabkan Anda mencapai kuota kunci Hibah per KMS Hi Untuk menghindari duplikasi hibah saat mencoba kembali `CreateGrant` permintaan, gunakan parameter. `Name` Untuk mendeteksi hibah duplikat, gunakan ListGrants operasi.

  > **Note**
  >
  > Beberapa AWS layanan membuat hibah untuk sumber daya berbeda yang mungkin tampak sebagai duplikat. Hibah ini memiliki siklus hidup yang terkait dengan sumber daya

yang berbeda. Menghapus hibah yang dibuat oleh AWS layanan dapat mengganggu dan memerlukan tindakan pencegahan ekstra.

- 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.

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 Anda menggunakan kebijakan utama atau kebijakan IAM untuk mengontrol akses ke operasi yang membuat dan mengelola hibah, Anda dapat menggunakan satu atau beberapa kondisi kebijakan berikut untuk membatasi izin. AWS KMS mendukung semua kunci kondisi terkait hibah berikut. Untuk informasi dan contoh mendetail, lihat AWS KMS kunci kondisi.

km: GrantConstraintSourceArn

Mengizinkan kepala sekolah untuk membuat hibah hanya jika hibah menyertakan `SourceArn` batasan dengan nilai yang ditentukan.

km: GrantConstraintType

Memungkinkan perwakilan membuat pemberian izin hanya jika pemberian izin menyertakan batas pemberian izin yang ditentukan.

km: GranteePrincipal

Memungkinkan kepala sekolah membuat pemberian izin hanya untuk perwakilan penerima yang ditentukan.

km: GranteeServicePrincipal

Mengizinkan kepala sekolah untuk membuat hibah hanya ketika hibah menentukan prinsipal layanan penerima hibah tertentu.

km: GrantIsForAWSResource

Memungkinkan kepala sekolah untuk menelepon `CreateGrant`, `ListGrants`, atau `RevokeGrant` hanya ketika AWS layanan yang terintegrasi dengan AWS KMS mengirimkan permintaan atas nama kepala sekolah.

km: GrantOperations

Memungkinkan perwakilan membuat pemberian izin, tetapi membatasi pemberian izin untuk operasi yang ditentukan.

km: RetiringPrincipal

Memungkinkan perwakilan membuat pemberian izin hanya jika pemberian izin menentukan perwakilan penghentian tertentu.

km: RetiringServicePrincipal

Mengizinkan kepala sekolah untuk membuat hibah hanya ketika hibah menentukan prinsipal layanan pensiun tertentu.

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
- Memberikan CreateGrant izin

Membuat pemberian izin

Untuk membuat hibah, panggil CreateGrant operasi. Tentukan kunci KMS, daftar operasi hibah yang diizinkan, dan kepala sekolah penerima hibah ??? atau kepala sekolah layanan penerima hibah.

Saat Anda menentukan prinsipal penerima hibah, Anda dapat secara opsional menunjuk kepala sekolah yang pensiun atau prinsipal layanan pensiun, dan menggunakan `Constraints` parameter untuk menentukan batasan hibah. https://docs.aws.amazon.com/kms/latest/APIReference/API_GrantConstraints.html Saat Anda menentukan prinsipal layanan penerima hibah, Anda harus menyertakan batasan hi `SourceArn` bah. Anda juga harus menentukan a `RetiringServicePrincipal` atau a`RetiringPrincipal`.

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 Kon sistensi akhir (untuk hibah).

Misalnya, `CreateGrant` perintah berikut membuat hibah yang memungkinkan pengguna yang diberi wewenang untuk mengambil `keyUserRole` peran untuk memanggil operasi Decrypt 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 `CreateGrant` permintaan hibah dengan properti yang sama dengan hibah yang ada, termasuk nama, itu mengakui permintaan

tersebut sebagai percobaan ulang, dan tidak membuat hibah 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 menunjukkan cara membuat hibah dalam beberapa bahasa pemrograman, lihat[Gunakan `CreateGrant` dengan AWS SDK atau CLI](#).

Menggunakan batas pemberian izin

Batasan hibah menetapkan kondisi pada izin yang diberikan hibah kepada prinsipal penerima hibah. Batas pemberian izin menggantikan kunci syarat dalam kebijakan kunci atau kebijakan IAM.

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

AWS KMS mendukung dua jenis kendala hibah: batasan konteks enkripsi dan batasan. `SourceArn`

Kendala konteks enkripsi

Kendala konteks enkripsi menetapkan persyaratan untuk konteks enkripsi dalam permintaan untuk operasi kriptografi. Setiap nilai batasan pemberian konteks enkripsi dapat mencakup hingga 8 pasangan konteks enkripsi. Nilai konteks enkripsi di setiap batas tidak boleh lebih dari 384 karakter.

- Batasan konteks enkripsi hanya valid dalam hibah untuk kunci KMS enkripsi simetris. Operasi kriptografi dengan kunci KMS lainnya tidak mendukung konteks enkripsi.

- Batasan konteks enkripsi diabaikan untuk `RetireGrant` operasi `DescribeKey` dan. `DescribeKey`dan `RetireGrant` tidak memiliki parameter konteks enkripsi, tetapi Anda dapat menyertakan operasi ini dalam hibah yang memiliki batasan konteks enkripsi.
- Anda dapat menggunakan batasan konteks enkripsi dalam hibah untuk `CreateGrant` operasi. Kendala konteks enkripsi mensyaratkan bahwa setiap hibah yang dibuat dengan `CreateGrant` izin memiliki batasan konteks enkripsi yang sama ketat atau lebih ketat.

AWS KMS mendukung dua batasan hibah konteks enkripsi.

EncryptionContextEquals

Gunakan `EncryptionContextEquals` untuk menentukan konteks enkripsi yang tepat untuk permintaan yang diizinkan.

`EncryptionContextEquals`mensyaratkan bahwa 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 pemberian `EncryptionContextEquals` memerlukan pasangan konteks `"Department": "IT"` enkripsi, hibah mengizinkan permintaan dari jenis yang ditentukan hanya jika konteks enkripsi dalam permintaan tepat`"Department": "IT"`.

EncryptionContextSubset

Gunakan `EncryptionContextSubset` untuk mengharuskan permintaan menyertakan pasangan konteks enkripsi tertentu.

`EncryptionContextSubset`mengharuskan permintaan menyertakan semua pasangan konteks enkripsi dalam batasan hibah (kecocokan yang tepat dan peka huruf besar-kecil), tetapi permintaan 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 pemberian `EncryptionContextSubset` memerlukan pasangan konteks `Department=IT` enkripsi, hibah mengizinkan permintaan dari jenis yang ditentukan ketika konteks enkripsi dalam permintaan`"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 Pemberian yang dibuat perintah ini memberi pengguna yang berwenang untuk mengambil `keyUserRole` peran izin untuk memanggil operasi Dec rypt. Tetapi izin itu hanya efektif jika 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 kepada `keyUserRole` peran hanya efektif bila `Decrypt` permintaan menggunakan pasangan konteks enkripsi yang sama yang ditentukan dalam batasan pemberian. Untuk menemukan hibah pada kunci KMS, gunakan ListGrants operasi.

```
$ 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 `EncryptionContextEquals` batasan hibah, konteks enkripsi dalam permintaan `Decrypt` operasi harus ber `"Department": "IT"` pasangan. 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 pemberian adalah`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 mensyaratkan bahwa salah satu pasangan konteks enkripsi dalam permintaan adalah`"Deparment": "IT"`.

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

Permintaan berikut dari kepala sekolah penerima hibah akan memenuhi batasan `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 kepala sekolah penerima hibah akan memenuhi batasan `EncryptionContextSubset` hibah, tetapi itu akan gagal dalam batasan 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
```

AWS layanan sering menggunakan batasan konteks enkripsi dalam hibah yang memberi mereka izin untuk menggunakan kunci KMS di Anda Akun AWS. Misalnya, Amazon DynamoDB menggunakan hibah seperti yang berikut untuk mendapatkan izin menggunakan Kunci yang dikelola AWS for 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 menggunakan kunci KMS yang ditentukan hanya untuk tabel tertentu di Anda. Akun AWS

Untuk mendapatkan output ini, jalankan ListGrants operasi 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"
        }
    ]
}
```

SourceArn kendala

SourceArn

Bat `SourceArn` asan membatasi izin pemberian untuk permintaan yang dibuat atas nama AWS sumber daya tertentu, yang diidentifikasi oleh Nama Sumber Daya Amazon (ARN). Kendala `SourceArn` hibah secara efektif menempatkan kunci kondisi aws:SourceArn global ke dalam hibah dan hanya dievaluasi ketika permintaan dibuat oleh AWS layanan.

Ken `SourceArn` dala selalu diperlukan ketika hibah menentukan a`GranteeServicePrincipal`. Secara opsional dapat digunakan dengan`GranteePrincipal`.

Tidak seperti batasan konteks enkripsi`SourceArn`, batasan didukung pada hibah untuk semua jenis kunci KMS.

Untuk menentukan `SourceArn` batasan dalam hibah, gunakan `Constraints` parameter dalam CreateGrant operasi. Contoh berikut membuat hibah yang memungkinkan prinsi AWS pal layanan untuk memanggil `Decrypt` operasi `Encrypt` dan, tetapi hanya ketika permintaan dibuat atas nama tabel DynamoDB yang ditentukan. Ken `SourceArn` dala diperlukan ketika hibah menentukan a`GranteeServicePrincipal`.

```
$ aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-service-principal service-name.amazonaws.com \
    --operations Encrypt Decrypt \
    --retiring-service-principal service-name.amazonaws.com
 \
    --constraints '{"SourceArn":"arn:aws:dynamodb:us-east-1:111122223333:table/
ExampleTable"}'
```

Hibah yang dihasilkan terlihat seperti berikut ini. Izin dalam hibah hanya efektif jika permintaan menyertakan ARN sumber yang cocok dengan `SourceArn` nilai dalam batasan hibah.

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

```
{
    "Grants": [
        {
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Operations": [
                "Encrypt",
                "Decrypt"
            ],
            "GranteeServicePrincipal": "service-name.amazonaws.com",
            "Constraints": {
                "SourceArn": "arn:aws:dynamodb:us-east-1:111122223333:table/
ExampleTable"
            },
            "CreationDate": 1718567315.0,
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "RetiringServicePrincipal": "service-name.amazonaws.com"
        }
    ]
}
```

Anda juga dapat menggabungkan beberapa batasan dalam satu hibah. Namun, Anda dapat menentukan salah satu `EncryptionContextEquals` atau`EncryptionContextSubset`, tetapi tidak keduanya.

Ketika `GranteeServicePrincipal` ditentukan, batasan `SourceArn` diperlukan, sehingga kombinasi yang valid untuk membuat hibah dengan beberapa batasan adalah:

- `EncryptionContextEquals` dengan `SourceArn`
- `EncryptionContextSubset` dengan `SourceArn`

Contoh berikut membuat hibah dengan `GranteeServicePrincipal` dan`RetiringServicePrincipal`, dan menyertakan `EncryptionContextSubset` kendala dan ken `SourceArn` dala.

```
$ aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-service-principal service-name.amazonaws.com \
```

```
    --retiring-service-principal service-name.amazonaws.com \
    --operations Encrypt Decrypt GenerateDataKey DescribeKey \
    --constraints '{"EncryptionContextSubset":
{"Department":"IT"},"SourceArn":"arn:aws:dynamodb:us-east-1:111122223333:table/
ExampleTable"}'
```

Hibah yang dihasilkan terlihat seperti berikut ini. Perhatikan bahwa respons mencakup `GranteeServicePrincipal` dan `RetiringServicePrincipal` bidang, dan `Constraints` bidang berisi keduanya `EncryptionContextSubset` dan`SourceArn`.

```
$ aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Grants": [
        {
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Operations": [
                "Encrypt",
                "Decrypt",
                "GenerateDataKey",
                "DescribeKey"
            ],
            "GranteeServicePrincipal": "service-name.amazonaws.com",
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                },
                "SourceArn": "arn:aws:dynamodb:us-east-1:111122223333:table/
ExampleTable"
            },
            "CreationDate": 1718567315.0,
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "RetiringServicePrincipal": "service-name.amazonaws.com"
        }
    ]
}
```

Memberikan CreateGrant izin

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.

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

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": {
                "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.

Melihat pemberian izin

Untuk melihat hibah, gunakan ListGrants operasi. Anda harus menentukan kunci KMS tempat hibah berlaku. Anda juga dapat memfilter daftar hibah berdasarkan ID hibah, pokok penerima hibah, atau prinsipal layanan penerima hibah. Untuk contoh lainnya, lihat Gunakan `ListGrants` dengan AWS SDK atau CLI.

Untuk melihat semua hibah di Akun AWS dan Wilayah dengan pokok pensiun tertentu atau prinsipal layanan pensiun, gunakan. ListRetirableGrants Respons meliputi detail tentang setiap pemberian izin.

> **Note**
>
> Kolom `GranteePrincipal` dalam respons `ListGrants` biasanya berisi perwakilan penerima pemberian izin. Namun, ketika prinsipal penerima hibah dalam hibah adalah AWS layanan, `GranteePrincipal` bidang tersebut berisi prinsipal layanan, yang mungkin mewakili beberapa kepala sekolah penerima hibah yang berbeda.

> **Note**
>
> Ketika hibah dibuat dengan `GranteeServicePrincipal` parameter, `ListGrants` respons menyertakan `GranteeServicePrincipal` bidang, bukan`GranteePrincipal`. Ini membedakan hibah yang secara eksplisit dibuat `GranteeServicePrincipal` untuk AWS kepala sekolah layanan dari hibah di mana AWS layanan diwakili di `GranteePrincipal` lapangan.

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"
            ]
        }
    ]
}
```

Contoh berikut menunjukkan `ListGrants` respons untuk hibah yang dibuat dengan`GranteeServicePrincipal`. Perhatikan bahwa respons menyertakan `GranteeServicePrincipal` bidang`GranteePrincipal`, bukan, dan `Constraints` bidang berisi `SourceArn` nilai.

```
$  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": 1718567315.0,
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Constraints": {
```

```
                "EncryptionContextSubset": {
                    "Department": "IT"
                },
                "SourceArn": "arn:aws:dynamodb:us-east-1:111122223333:table/
ExampleTable"
            },
            "RetiringServicePrincipal": "service-name.amazonaws.com",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteeServicePrincipal": "service-name.amazonaws.com",
            "Operations": [
                "Encrypt",
                "Decrypt",
                "GenerateDataKey"
            ]
        }
    ]
}
```

Menggunakan token izin

AWS KMS API mengikuti model konsistensi akhirnya. Saat Anda membuat pemberian izin, mungkin tidak langsung efektif. 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. Setelah perubahan sepenuhnya menyebar ke seluruh sistem, prinsipal penerima hibah dapat menggunakan izin dalam hibah tanpa menentukan token hibah atau bukti hibah apa pun. Namun, jika hibah yang begitu baru sehingga belum diketahui semua orang AWS KMS, permintaan mungkin gagal dengan `AccessDeniedException` kesalahan.

Untuk segera menggunakan izin dalam pemberian izin baru, gunakan token izin untuk pemberian izin. Simpan token hibah yang dikem CreateGrant balikan oleh operasi. Kemudian kirimkan token hibah dalam permintaan untuk AWS KMS operasi. Anda dapat mengirimkan token hibah ke operasi AWS KMS hibah apa pun dan Anda dapat mengirimkan beberapa token hibah dalam permintaan yang sama.

Contoh berikut menggunakan `CreateGrant` operasi untuk membuat hibah yang memungkinkan operasi GenerateDataKey dan Dek ripsi. 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 menghentikan hibah baru bahkan sebelum hibah tersedia secara keseluruhan. 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.

RevokeGrantOper RetireGrant asi dan 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 AWS KMS operasi, akses ke `RevokeGrant` operasi dikendalikan melalui kebijakan utama 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.

Hibah dapat pensiun dengan salah satu dari yang berikut:

- Prinsipal pensiun atau kepala sekolah layanan pensiun yang ditentukan dalam hibah.
- Kepala sekolah penerima hibah atau kepala sekolah layanan penerima hibah, jika hibah mencakup operasi. `RetireGrant`
- Akun AWS Di mana hibah itu dibuat.

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. Iz `kms:RetireGrant` in tidak efektif dalam kebijakan utama atau kebijakan kontrol sumber daya.

- Untuk menolak izin untuk menghentikan hibah, Anda dapat menggunakan `Deny` tindakan dengan `kms:RetireGrant` izin dalam kebijakan IAM Anda.
- Akun AWS Yang memiliki kunci KMS dapat mendelegasikan `kms:RetireGrant` izin ke prinsipal IAM di akun.
- Jika prinsipal pensiun berbeda Akun AWS, administrator di akun lain dapat menggunakannya `kms:RetireGrant` untuk mendelegasikan izin untuk menghentikan hibah ke prinsipal IAM di akun tersebut.

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 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 hibah tersedia AWS KMS, gunakan token hibah untuk menghentikan hibah tersebut. Anda tidak dapat menggunakan token izin untuk mencabut pemberian izin.

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 platform yang dibuktikan

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: kunci SourceAccount` kondisi.

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 kondisi alamat IP dalam kebijakan dengan AWS KMS izin
- Menggunakan kondisi titik akhir VPC dalam kebijakan dengan AWS KMS izin
- Menggunakan alamat IPv6 di IAM dan AWS KMS kebijakan utama

Menggunakan kondisi alamat IP dalam kebijakan dengan AWS KMS izin

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 kondisi titik akhir VPC dalam kebijakan dengan AWS KMS izin

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. Lihat perinciannya di Gunakan titik akhir VPC untuk mengontrol akses ke sumber daya AWS KMS.

- `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.

Menggunakan alamat IPv6 di IAM dan AWS KMS kebijakan utama

Sebelum mencoba mengakses AWS KMS melalui IPv6, pastikan kebijakan kunci dan IAM yang berisi batasan alamat IP diperbarui untuk menyertakan rentang alamat IPv6. Kebijakan berbasis IP yang tidak diperbarui untuk menangani alamat IPv6 dapat mengakibatkan klien salah kehilangan atau mendapatkan akses ketika mereka mulai menggunakan IPv6. Untuk panduan umum tentang kontrol akses KMS, lihatAkses dan izin kunci KMS. Untuk mempelajari tentang KMS dan dukungan dual stack, lihatDual-stack dukungan titik akhir.

> **Important**
>
> Pernyataan ini tidak mengizinkan tindakan apa pun. Gunakan pernyataan ini dalam kombinasi dengan pernyataan lain yang memungkinkan tindakan tertentu.

Pernyataan berikut secara eksplisit menolak akses ke semua izin KMS untuk permintaan yang berasal dari kisaran alamat IPv4. `192.0.2.*` Alamat IP apa pun di luar rentang ini tidak secara eksplisit ditolak izin KMS. Karena semua alamat IPv6 berada di luar rentang yang ditolak, pernyataan ini tidak secara eksplisit menolak izin KMS untuk alamat IPv6 apa pun.

```
{
     "Sid": "DenyKMSPermissions",
     "Effect": "Deny",
    "Action": [
        "kms:*"
    ],
    "Resource": "*",
    "Condition": {
        "NotIpAddress": {
            "aws:SourceIp": [
                "192.0.2.0/24"
            ]
        }
    }
}
```

Anda dapat memodifikasi `Condition` elemen untuk menolak rentang alamat IPv4 (`192.0.2.0/24`) dan IPv6 (`2001:db8:1234::/32`) seperti yang ditunjukkan pada contoh berikut.

```
{
    "Sid": "DenyKMSPermissions",
    "Effect": "Deny",
    "Action": [
        "kms:*"
    ],
    "Resource": "*",
    "Condition": {
        "NotIpAddress": {
            "aws:SourceIp": [
                "192.0.2.0/24",
                "2001:db8:1234::/32"
            ]
        }
    }
}
```

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.

- Single-valued kunci kondisi 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.
- Multi-valuedkunci kondisi memiliki beberapa nilai dalam konteks otorisasi (permintaan atau sumber daya). Misalnya, karena setiap kunci KMS dapat memiliki beberapa alias, kms: ResourceAliases dapat memiliki beberapa nilai. Multi-valued kunci kondisi memerlukan operator yang ditetapkan.

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
- km:: EncryptionContextkunci-konteks
- km: EncryptionContextKeys
- km: ExpirationModel
- km: GrantConstraintType
- km: GrantIsForAWSResource
- km: GrantOperations
- km: GranteePrincipal
- km: KeyAgreementAlgorithm
- 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: GrantConstraintSourceArn
- km: GranteeServicePrincipal
- km: RetiringServicePrincipal
- km: RotationPeriodInDays
- km: ScheduleKeyDeletionPendingWindowInDays
- km: SigningAlgorithm
- km: TrailingDaysWithoutKeyUsage
- 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	Single-valued	`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	Single-valued	Operasi sumber daya utama KMS Operasi penyimpanan kunci kustom	Kebijakan kunci dan kebijakan IAM

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	Single-valued	`GenerateDataKeyPair` `GenerateDataKeyPairWithoutPlaintext`	Kebijakan kunci dan kebijakan IAM

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 “km:: EncryptionContextkunci-konteks”
- the section called “km: EncryptionContextKeys”

km: EncryptionAlgorithm

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan

`kms:EncryptionAlgorithm`	String	Single-valued	`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 yang ditentukan 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

km:: EncryptionContextkunci-konteks

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan

`kms:EncryptionContext:` *`context-key`*	String	Single-valued	`CreateGrant`
			`Encrypt`
			`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`
			`RetireGrant`
				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 *context-key* placeholder dengan kunci konteks enkripsi. Ganti *context-value* placeholder 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`. Lihat perinciannya di 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:EncryptionContextKeysdalam 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-valued	`CreateGrant`
			`Decrypt`
			`Encrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`
			`RetireGrant`
				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

- km:: EncryptionContextkunci-konteks
- km: GrantConstraintType

km: ExpirationModel

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:ExpirationModel`	String	Single-valued	`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	Single-valued	`CreateGrant` `RetireGrant`	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.

> **Note**
>
> Kunci `kms:GrantConstraintType` kondisi hanya mengevaluasi batasan hibah konteks enkripsi (dan). `EncryptionContextEquals` `EncryptionContextSubset` Untuk mengontrol akses berdasarkan batasan `SourceArn` hibah, gunakan kunci kondisi kms:GrantConstraintSourceArn.

> **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

- km:: EncryptionContextkunci-konteks
- km: EncryptionContextKeys
- km: GrantConstraintSourceArn
- km: GranteePrincipal
- km: GranteeServicePrincipal
- km: GrantIsForAWSResource
- km: GrantOperations
- km: RetiringPrincipal
- km: RetiringServicePrincipal

km: GrantIsForAWSResource

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan

`kms:GrantIsForAWSResource`	Boolean	Single-valued	`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: GrantConstraintSourceArn
- km: GrantConstraintType
- km: GranteePrincipal
- km: GranteeServicePrincipal
- km: GrantOperations
- km: RetiringPrincipal
- km: RetiringServicePrincipal

km: GrantOperations

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:GrantOperations`	String	Multi-valued	`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: GrantConstraintSourceArn
- km: GrantConstraintType
- km: GrantIsForAWSResource
- km: GranteePrincipal
- km: GranteeServicePrincipal
- km: RetiringPrincipal
- km: RetiringServicePrincipal

km: GranteePrincipal

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:GranteePrincipal`	String	Single-valued	`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: GrantConstraintSourceArn
- km: GrantConstraintType
- km: GrantIsForAWSResource
- km: GrantOperations
- km: GranteeServicePrincipal
- km: RetiringPrincipal
- km: RetiringServicePrincipal

km: KeyAgreementAlgorithm

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:KeyAgreementAlgorithm`	String	Single-valued	`DeriveSharedSecret`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol `kms:KeyAgreementAlgorithm` kondisi untuk mengontrol akses ke DeriveSharedSecretoperasi berdasarkan nilai `KeyAgreementAlgorithm` parameter dalam permintaan. Satu-satunya nilai yang valid untuk `KeyAgreementAlgorithm` adalah`ECDH`.

Misalnya, pernyataan kebijakan kunci berikut menggunakan kunci `kms:KeyAgreementAlgorithm` kondisi untuk menolak semua akses DeriveSharedSecret kecuali `KeyAgreementAlgorithm` is`ECDH`.

```
{
       "Effect": "Deny",
       "Principal": {
         "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
       },
       "Action": "kms:DeriveSharedSecret",
       "Resource": "*",
       "Condition": {
            "StringNotEquals": {
               "kms:KeyAgreementAlgorithm": "ECDH"
         }
```

```
        }
}
```

Lihat juga

- the section called “km: KeyUsage”

km: KeyOrigin

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:KeyOrigin`	String	Single-valued	`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` are`AWS_KMS`,`AWS_CLOUDHSM`,`EXTERNAL_KEY_STORE`, dan`EXTERNAL`.

Misalnya, Anda dapat membuat kunci KMS hanya ketika materi kunci dihasilkan di AWS KMS (`AWS_KMS`), hanya jika materi kunci dihasilkan dalam AWS CloudHSM klaster yang terkait dengan penyimpanan kunci kustom CloudHSM `AWS_CLOUDHSM` (), hanya jika materi kunci dihasilkan di penyimpanan kunci eksternal `EXTERNAL_KEY_STORE` (), 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.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": "kms:CreateKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:KeyOrigin": "AWS_KMS"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:GenerateDataKeyPair",
        "kms:GenerateDataKeyPairWithoutPlaintext",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "kms:KeyOrigin": "AWS_CLOUDHSM"
        }
      }
    }
  ]
}
```

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	Single-valued	`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	Single-valued	`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` are`ENCRYPT_DECRYPT`,`SIGN_VERIFY`,`GENERATE_VERIFY_MAC`, dan`KEY_AGREEMENT`.

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	Single-valued	`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.

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:MacAlgorithm": "HMAC_SHA_384"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:MacAlgorithm": "HMAC_SHA_512"
        }
      }
    }
  ]
}
```

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	Single-valued	`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` are`RAW`, `DIGEST` dan`EXTERNAL_MU`.

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	Single-valued	`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	Single-valued	`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)	Single-valued	`UpdatePrimaryRegion`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol kondisi ini untuk membatasi Wilayah tujuan dalam 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	Single-valued	`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)	Single-valued	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. Lihat perinciannya di 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 pengenal 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-valued	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)	Single-valued	`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)	Single-valued	`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: GrantConstraintSourceArn
- km: GrantConstraintType
- km: GranteePrincipal
- km: GranteeServicePrincipal
- km: GrantIsForAWSResource
- km: GrantOperations
- km: RetiringServicePrincipal

km: GrantConstraintSourceArn

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Tipe kebijakan
`kms:GrantConstraintSourceArn`	ARN	Single-valued	`CreateGrant`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol kondisi ini untuk mengontrol akses ke CreateGrantoperasi berdasarkan `SourceArn` nilai dalam parameter Constraints permintaan. Kendala `SourceArn` hibah secara efektif menempatkan kunci kondisi aws:SourceArnglobal ke dalam hibah.

Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:GrantConstraintSourceArn` kondisi untuk memungkinkan pembuatan hibah untuk kunci KMS hanya jika `SourceArn` kendala dalam hibah cocok dengan pola ARN tabel DynamoDB tertentu.

```
{
```

```
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "ArnLike": {
      "kms:GrantConstraintSourceArn": "arn:aws:dynamodb:us-east-1:111122223333:table/
ExampleTable*"
    }
  }
}
```

Lihat juga

- km: GrantConstraintType
- km: GranteePrincipal
- km: GranteeServicePrincipal
- km: GrantIsForAWSResource
- km: GrantOperations
- km: RetiringPrincipal
- km: RetiringServicePrincipal

km: GranteeServicePrincipal

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:GranteeServicePrincipal`	String	Single-valued	`CreateGrant`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol kondisi ini untuk mengontrol akses ke CreateGrantoperasi berdasarkan nilai GranteeServicePrincipalparameter dalam permintaan. Nilai adalah pokok AWS layanan yang ditentukan sebagai penerima hibah (misalnya,`service-name.amazonaws.com`).

Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:GranteeServicePrincipal` kondisi untuk memungkinkan pembuatan hibah untuk kunci KMS hanya ketika prinsipal layanan penerima hibah dalam hibah adalah. *`service-name`*`.amazonaws.com`

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GranteeServicePrincipal": "service-name.amazonaws.com"
    }
  }
}
```

Lihat juga

- km: GrantConstraintSourceArn
- km: GrantConstraintType
- km: GranteePrincipal
- km: GrantIsForAWSResource
- km: GrantOperations
- km: RetiringPrincipal
- km: RetiringServicePrincipal

km: RetiringServicePrincipal

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:RetiringServicePrincipal`	String	Single-valued	`CreateGrant`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol kondisi ini untuk mengontrol akses ke CreateGrantoperasi berdasarkan nilai RetiringServicePrincipalparameter dalam permintaan. Nilai adalah pokok AWS layanan yang ditentukan sebagai prinsipal pensiun dari hibah (misalnya,`service-name.amazonaws.com`).

Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:RetiringServicePrincipal` kondisi untuk memungkinkan pembuatan hibah untuk kunci KMS hanya ketika prinsipal layanan pensiun dalam hibah adalah. *`service-name`*`.amazonaws.com`

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:RetiringServicePrincipal": "service-name.amazonaws.com"
    }
  }
}
```

Lihat juga

- km: GrantConstraintSourceArn
- km: GrantConstraintType
- km: GranteePrincipal
- km: GranteeServicePrincipal
- km: GrantIsForAWSResource
- km: GrantOperations
- km: RetiringPrincipal

km: RotationPeriodInDays

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Tipe kebijakan
`kms:RotationPeriodInDays`	Numerik	Single-valued	`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	Tipe kebijakan
`kms:ScheduleKeyDeletionPendingWindowInDays`	Numerik	Single-valued	`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	Single-valued	`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: TrailingDaysWithoutKeyUsage

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Tipe kebijakan
`kms:TrailingDaysWithoutKeyUsage`	Numerik	Single-valued	`DisableKey` `ScheduleKeyDeletion`	Kebijakan kunci dan kebijakan IAM

Kunci `kms:TrailingDaysWithoutKeyUsage` kondisi mewakili jumlah hari trailing tanpa operasi kriptografi pada kunci KMS, dihitung dari operasi kriptografi terakhir yang berhasil, atau dari tanggal pembuatan kunci KMS atau `TrackingStartDate` jika kunci belum pernah digunakan. Untuk informasi selengkapnya tentang tanggal mulai pelacakan dan bagaimana perilakunya dapat memengaruhi kunci kondisi ini, lihatMemahami periode pelacakan penggunaan. Anda dapat menggunakan kunci kondisi ini dalam kebijakan utama dan kebijakan IAM untuk mengontrol akses ke ScheduleKeyDeletiondan DisableKeyoperasi.

Kunci kondisi ini adalah kondisi numerik, bernilai tunggal. Nilai trailing days selalu dibulatkan ke bawah. Misalnya, jika kunci terakhir digunakan 89,9 hari yang lalu, nilainya adalah 89.

`kms:TrailingDaysWithoutKeyUsage`Dihitung sebagai berikut:

- Untuk kunci yang dibuat pada atau setelah `TrackingStartDate` yang belum digunakan sejak pelacakan dimulai:

  Formula: Tanggal Sekarang - `KeyCreationDate`

  Misalnya, jika Tanggal Sekarang adalah 30 April 2026 dan kuncinya dibuat pada 20 April 2026, nilainya adalah 10.
- Untuk kunci yang dibuat sebelum `TrackingStartDate` yang belum digunakan sejak pelacakan dimulai:

  Formula: Tanggal Sekarang - `TrackingStartDate`

  Misalnya, jika Tanggal Sekarang adalah 30 April 2026 dan 15 April 2026, nilainya adalah 15, terlepas dari kapan kunci dibuat. `TrackingStartDate`
- Untuk kunci yang telah digunakan sejak pelacakan dimulai, terlepas dari kapan kunci dibuat:

Formula: Tanggal Sekarang - `Timestamp` dari `KeyLastUsage`

Contoh pernyataan kebijakan kunci berikut menyangkal `DisableKey` operasi `ScheduleKeyDeletion` dan ketika kunci telah digunakan dalam 90 hari terakhir, atau ketika kunci tidak memiliki penggunaan yang direkam dan dibuat 90 atau kurang hari yang lalu atau pelacakan dimulai 90 hari yang lalu atau kurang hari yang lalu. Ini mencegah penghapusan atau penonaktifan kunci yang digunakan secara aktif atau yang baru dibuat secara tidak sengaja.

```
{
  "Effect": "Deny",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:DisableKey"
  ],
  "Principal": "*",
  "Resource": "*",
  "Condition": {
    "NumericLessThanEquals": {
      "kms:TrailingDaysWithoutKeyUsage": "90"
    }
  }
}
```

Warning

Jangan hanya mengandalkan informasi penggunaan terakhir saat menghapus kunci yang tidak digunakan. Sebagai gantinya, nonaktifkan kunci terlebih dahulu dan pantau AWS CloudTrail `DisabledException` entri, yang menunjukkan upaya untuk menggunakan kunci saat dinonaktifkan. Ini membantu mengidentifikasi potensi dependensi dan kegagalan beban kerja. AWS CloudTrail tetap menjadi sumber otoritatif untuk semua panggilan API yang dilakukan ke kunci Anda.

Lihat juga

- Periksa operasi kriptografi terakhir yang dilakukan dengan kunci KMS
- km: ScheduleKeyDeletionPendingWindowInDays

km: ValidTo

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:ValidTo`	Stempel waktu	Single-valued	`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	Single-valued	Operasi sumber daya utama KMS	Kebijakan kunci dan kebijakan IAM

Kunci `kms:ViaService` kondisi membatasi penggunaan kunci KMS untuk permintaan dari yang ditentukan Layanan AWS. Kunci kondisi ini hanya berlaku untuk sesi akses Teruskan. 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 Penggunaan AWS KMS enkripsi dengan AWS layanan.
- 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 **`kms: kunci ViaService`** kondisi

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
Operasi Amazon AI	`aiops.`*`AWS_region`*` .amazonaws.com`
AWS App Runner	`apprunner.`*`AWS_region`*`.amazonaws.com`

Nama layanan	AWS KMS ViaService nama
AWS AppFabric	appfabric. *AWS_region* .amazonaws.com
Amazon AppFlow	appflow.*AWS_region* .amazonaws.com
AWS Transform MGN	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
Salinan Model Batuan Dasar Amazon	bedrock.*AWS_region* .amazonaws.com
Amazon Chime SDK	chimevoiceconnector. *AWS_region* .amazonaws.com
AWS Clean Rooms ML	cleanrooms-ml. *AWS_region* .amazonaws.com
AWS CodeArtifact	codeartifact. *AWS_region* .amazonaws.com
CodeGuru Peninjau Amazon	codeguru-reviewer. *AWS_region* .amazonaws.com
Amazon Comprehend	comprehend. *AWS_region* .amazonaws.com
Pelanggan Amazon Connect	connect.*AWS_region* .amazonaws.com

Nama layanan	AWS KMS ViaService nama
Profil Pelanggan Pelanggan Amazon Connect	`profile.`*`AWS_region`*` .amazonaws.com`
Amazon Q di Connect	`wisdom.`*`AWS_region`*` .amazonaws.com`
AWS Database Migration Service (AWS DMS)	`dms.`*`AWS_region`*` .amazonaws.com`
AWS DeepRacer	`deepracer.`*`AWS_region`*` .amazonaws.com`
AWS Directory Service	`directoryservice.`*`AWS_region`*` .amazonaws.com`
Amazon DocumentDB	`docdb-elastic.`*`AWS_region`*` .amazonaws.com`
Amazon DynamoDB	`dynamodb.`*`AWS_region`*` .amazonaws.com`
Amazon EC2 Systems Manager (SSM)	`ssm.`*`AWS_region`*` .amazonaws.com`
Amazon Elastic Block Store (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`

Nama layanan	AWS KMS ViaService nama
AWS Resolusi Entitas	entityresolution. *AWS_region* .amazonaws.com
Amazon EventBridge	events.*AWS_region* .amazonaws.com
Amazon FinSpace	finspace. *AWS_region* .amazonaws.com
Amazon Forecast	forecast. *AWS_region* .amazonaws.com
Amazon FSx	fsx.*AWS_region* .amazonaws.com
AWS Glue	glue.*AWS_region* .amazonaws.com
AWS Ground Station	groundstation. *AWS_region* .amazonaws.com
Amazon GuardDuty	malware-protection. *AWS_region* .amazonaws.com
AWS HealthLake	healthlake. *AWS_region* .amazonaws.com
AWS IoT SiteWise	iotsitewise. *AWS_region* .amazonaws.com
Amazon Kendra	kendra.*AWS_region* .amazonaws.com
Amazon Keyspaces (untuk Apache Cassandra)	cassandra. *AWS_region* .amazonaws.com
Amazon Kinesis	kinesis.*AWS_region* .amazonaws.com
Amazon Data Firehose	firehose. *AWS_region* .amazonaws.com

Nama layanan	AWS KMS ViaService nama
Amazon Kinesis Video Streams	kinesisvideo. *AWS_region* .amazonaws.com
AWS Lambda	lambda.*AWS_region* .amazonaws.com
Amazon Lex	lex.*AWS_region* .amazonaws.com
AWS License Manager	license-manager. *AWS_region* .amazonaws.com
Amazon Location Service	geo.*AWS_region* .amazonaws.com
Amazon Lookout 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
AWS Mainframe Modernization Pengujian Aplikasi	apptest.*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

Nama layanan	AWS KMS ViaService nama
Amazon MemoryDB	memorydb. *AWS_region* .amazonaws.com
Amazon Monitron	monitron. *AWS_region* .amazonaws.com
Amazon MQ	mq.*AWS_region* .amazonaws.com
Amazon Neptune	rds.*AWS_region* .amazonaws.com
Amazon Nimble Studio	nimble.*AWS_region* .amazonaws.com
AWS HealthOmics	omics.*AWS_region* .amazonaws.com
OpenSearch Layanan Amazon	es.*AWS_region* .amazonaws.com , aoss.*AWS_region* .amazonaws.com
Paket OpenSearch Kustom Amazon	custom-packages. *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

Nama layanan	AWS KMS ViaService nama
Amazon Relational Database Service (Amazon RDS)	rds.*AWS_region* .amazonaws.com
Toko Data Replikasi Amazon	ards.*AWS_region* .amazonaws.com
Amazon SageMaker AI	sagemaker. *AWS_region* .amazonaws.com
AWS Secrets Manager	secretsmanager. *AWS_region* .amazonaws.com
Amazon Security Lake	securitylake. *AWS_region* .amazonaws.com
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
Tabel Amazon S3	s3tables. *AWS_region* .amazonaws.com
AWS Snowball Edge	importexport. *AWS_region* .amazonaws.com
AWS Step Functions	states.*AWS_region* .amazonaws.com
AWS Storage Gateway	storagegateway. *AWS_region* .amazonaws.com
Manajer Insiden AWS Systems Manager	ssm-incidents. *AWS_region* .amazonaws.com

Nama layanan	AWS KMS ViaService nama
Manajer Insiden AWS Systems 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`
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	Single-valued	`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	Single-valued	`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 platform yang dibuktikan

AWS KMS menyediakan kunci kondisi untuk mendukung pengesahan kriptografi untuk AWS Nitro Enclave dan NitroTPM. 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. NitroTPM memperluas fungsionalitas pengesahan serupa ke instans EC2.

Saat Anda memanggil operasi Dekripsi,, DeriveSharedSecret, GenerateDataKey, GenerateDataKeyPair, atau GenerateRandomAPI dengan dokumen pengesahan yang ditandatangani, 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 Dukungan pengesahan kriptografi di AWS KMS.

> **Note**
>
> Jika Anda tidak memberikan kebijakan kunci saat membuat AWS KMS kunci, AWS buatlah untuk Anda. Kebijakan kunci default ini memberikan akses penuh kunci KMS ke kunci tersebut dan memungkinkan akun menggunakan kebijakan IAM untuk mengizinkan akses ke kunci tersebut. Akun AWS Kebijakan ini memungkinkan semua tindakan seperti Dekripsi. AWS merekomendasikan Least-privilege izin untuk menerapkan prinsip kebijakan kunci KMS Anda. Anda juga dapat membatasi akses dengan memodifikasi tindakan kebijakan kunci KMS untuk. `kms:*` `NotAction:` `kms:Decrypt`

Kunci syarat berikut memungkinkan Anda membatasi izin untuk operasi ini berdasarkan konten dokumen pengesahan yang ditandatangani. Sebelum mengizinkan operasi, AWS KMS bandingkan dokumen pengesahan dengan nilai dalam kunci kondisi ini AWS KMS .

Kunci kondisi untuk Nitro Enclave

Kunci kondisi berikut khusus untuk pengesahan Nitro Enclave:

km: RecipientAttestation:ImageSha384

AWS KMS Kunci Kondisi	Jenis Syarat	Jenis nilai	Operasi API	Jenis Kebijakan

`kms:RecipientAttestation:ImageSha384`	String	Single-valued	`Decrypt`
			`DeriveSharedSecret`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Kebijakan kunci dan kebijakan IAM

Kunci `kms:RecipientAttestation:ImageSha384` kondisi mengontrol akses ke`Decrypt`,`DeriveSharedSecret`,`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.

Misalnya, pernyataan kebijakan kunci berikut memungkinkan `data-processing` peran menggunakan kunci KMS untuk Dekripsi,,, DeriveSharedSecretGenerateDataKeyGenerateDataKeyPair, 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:DeriveSharedSecret",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
 "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

kms: PCR_ID RecipientAttestation:PCR **< >**

AWS KMS Kunci Kondisi	Jenis Syarat	Jenis nilai	Operasi API	Jenis Kebijakan

`kms:RecipientAttestation:PCR<PCR_ID>`	String	Single-valued	`Decrypt`
			`DeriveSharedSecret`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Kebijakan kunci dan kebijakan IAM

Kunci `kms:RecipientAttestation:PCR<PCR_ID>` kondisi mengontrol akses ke`Decrypt`,,`DeriveSharedSecret`, `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 enclave Nitro. AWS

Nilai ini juga termasuk dalam CloudTrailperistiwa yang mewakili permintaan AWS KMS untuk kantong Nitro.

Untuk menentukan nilai PCR, gunakan format berikut. Gabungkan ID PCR ke nama kunci syarat. Anda dapat menentukan ID PCR yang mengidentifikasi salah satu dari enam pengukuran enclave atau ID PCR khusus yang Anda tentukan untuk kasus penggunaan tertentu. 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:
 "abc1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcde
```

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":
 "abc1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af79355814748447674530 82c6f1
    }
  }
}
```

Kunci kondisi untuk NitroTPM

Kunci kondisi berikut khusus untuk pengesahan NitroTPM:

kms: PCR_ID RecipientAttestation:NitroTPMPCR **< >**

AWS KMS Kunci Kondisi	Jenis Syarat	Jenis nilai	Operasi API	Jenis Kebijakan

`kms:RecipientAttestation:NitroTPMPCR<PCR_ID>`	String	Single-valued	`Decrypt`
			`DeriveSharedSecret`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Kebijakan kunci dan kebijakan IAM

Kunci `kms:RecipientAttestation:NitroTPMPCR<PCR_ID>` kondisi mengontrol akses ke`Decrypt`,,`DeriveSharedSecret`, `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 NitRotPM.

Nilai ini juga termasuk dalam CloudTrailperistiwa yang mewakili permintaan AWS KMS untuk NitRotPM.

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:NitroTPMPCRPCR_ID": "PCR_value"
```

Misalnya, kunci kondisi berikut menentukan nilai tertentu untuk PCR4:

```
kms:RecipientAttestation:NitroTPMPCR4:
 "abc1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcde
```

Contoh pernyataan kebijakan kunci berikut memungkinkan `data-processing` peran untuk menggunakan kunci KMS untuk operasi Dekripsi.

Kunci `kms:RecipientAttestation:NitroTPMPCR` kondisi dalam pernyataan ini memungkinkan operasi hanya jika nilai PCR4 dalam dokumen pengesahan yang ditandatangani dalam permintaan cocok dengan `kms:RecipientAttestation:NitroTPMPCR4` nilai dalam kondisi. 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 NitroTPM data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:NitroTPMPCR4":
 "abc1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af79355814748447674530 82c6f1
    }
```

```
  }
}
```

Least-privilege izin

Karena kunci KMS Anda melindungi informasi sensitif, kami sarankan mengikuti prinsip akses yang paling tidak memiliki hak istimewa. Mendelegasikan izin minimum yang diperlukan untuk melakukan tugas saat Anda menentukan kebijakan utama Anda. Hanya izinkan semua tindakan (`kms:*`) pada kebijakan kunci KMS jika Anda berencana untuk membatasi izin lebih lanjut dengan kebijakan IAM tambahan. Jika Anda berencana mengelola izin dengan kebijakan IAM, batasi siapa yang memiliki kemampuan untuk membuat dan melampirkan kebijakan IAM ke prinsipal IAM dan pantau perubahan kebijakan.

Jika Anda mengizinkan semua tindakan (`kms:*`) di kebijakan utama dan kebijakan IAM, prinsipal memiliki izin administratif dan penggunaan ke kunci KMS. Sebagai praktik terbaik keamanan, sebaiknya hanya mendelegasikan izin ini ke prinsipal tertentu. Anda dapat melakukan ini dengan secara eksplisit menamai prinsipal dalam kebijakan utama atau dengan membatasi prinsipal mana yang dilampirkan kebijakan IAM. Anda juga dapat menggunakan kunci kondisi untuk membatasi izin. Misalnya, Anda dapat menggunakan `aws:PrincipalTag` untuk mengizinkan semua tindakan jika prinsipal yang membuat panggilan API memiliki tag yang ditentukan dalam aturan kondisi.

Untuk bantuan memahami bagaimana pernyataan kebijakan dievaluasi di AWS, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM. Sebaiknya tinjau topik ini sebelum menulis kebijakan untuk mengurangi kemungkinan kebijakan Anda memiliki efek yang tidak diinginkan, seperti menyediakan akses ke kepala sekolah yang seharusnya tidak memiliki akses.

> **Tip**
>
> Saat menguji aplikasi di lingkungan non-produksi, gunakan Penganalisis Akses IAM untuk membantu Anda menerapkan hak istimewa terkecil ke kebijakan IAM Anda.

Jika Anda menggunakan pengguna IAM alih-alih peran IAM, sebaiknya aktifkan otentikasi AWS multi-faktor (MFA) untuk mengurangi kerentanan kredentif jangka panjang. Anda dapat menggunakan MFA untuk melakukan hal berikut:

- Memwajibkan pengguna memvalidasi kredensialnya dengan MFA sebelum melakukan tindakan istimewa, seperti menjadwalkan penghapusan kunci.

- Pisahkan kepemilikan kata sandi akun administrator dan perangkat MFA antara individu untuk menerapkan otorisasi terpisah.

Pelajari selengkapnya

- AWS kebijakan terkelola untuk fungsi pekerjaan
- Teknik untuk menulis kebijakan IAM dengan hak akses paling rendah

Menerapkan izin yang paling tidak memiliki hak istimewa

Saat Anda memberikan izin AWS layanan untuk menggunakan kunci KMS, pastikan izin tersebut hanya valid untuk sumber daya yang harus diakses layanan atas nama Anda. Strategi hak istimewa terendah ini membantu mencegah penggunaan kunci KMS yang tidak sah saat permintaan diteruskan antar AWS layanan.

Untuk menerapkan strategi hak istimewa yang paling sedikit, 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 menyertakan kunci kms:EncryptionContextKeys kondisi kms:EncryptionContext:context-key atau dalam kebijakan yang memungkinkan prinsipal memanggil operasi AWS KMS kriptografi. Kunci kondisi ini sangat efektif karena mengaitkan izin dengan konteks en kripsi yang terikat ke ciphertext saat sumber daya dienkripsi.

Gunakan kunci kondisi konteks enkripsi hanya jika tindakan dalam pernyataan kebijakan adalah CreateGrant atau operasi kriptografi AWS KMS simetris yang mengambil `EncryptionContext` parameter, seperti operasi seperti GenerateDataKey atau Dekripsi. (Untuk daftar operasi yang didukung, lihat kms:EncryptionContext:context-key atau 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 en kripsi untuk AWS Proton mengidentifikasi sumber daya AWS Proton dan template terkaitnya. Kon AWS Secrets Manager teks enkripsi mengidentifikasi rahasia dan versinya. Konteks enkripsi untuk Lokasi Amazon mengidentifikasi pelacak atau koleksi.

Contoh pernyataan kebijakan utama berikut memungkinkan Layanan Lokasi Amazon 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: kunci SourceAccount`** kondisi

Jika prinsipal dalam pernyataan kebijakan utama adalah AWS prinsipal layanan, kami sangat menyarankan Anda menggunakan kunci kondisi aws:SourceArn atau aws:SourceAccount global, selain kunci `kms:EncryptionContext:`*`context-key`* kondisi. Nilai ARN dan akun disertakan dalam konteks otorisasi hanya ketika permintaan datang AWS KMS dari AWS layanan lain. Kombinasi kondisi ini menerapkan izin yang paling tidak istimewa dan menghindari skenario wakil yang mungkin membingungkan. Prinsip layanan biasanya tidak digunakan sebagai prinsipal dalam kebijakan utama, tetapi beberapa AWS layanan, seperti AWS CloudTrail, memerlukannya.

Untuk menggunakan kunci kondisi `aws:SourceArn` atau `aws:SourceAccount` global, tetapkan nilai ke Nama Sumber Daya Amazon (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 sebagai `aws:SourceAccount` 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 `aws:SourceArn` kondisi. Sebagai gantinya, gunakan kunci `aws:SourceAccount` kondisi. Untuk detail tentang aturan dokumen kebijakan utama, lihatFormat kebijakan utama.

Dalam contoh kebijakan kunci berikut, kepala sekolah yang mendapatkan izin adalah prinsi AWS CloudTrail pal layanan,`cloudtrail.amazonaws.com`. Untuk menerapkan hak istimewa terkecil, kebijakan ini menggunakan kunci `kms:EncryptionContext:`*`context-key`* kondisi `aws:SourceArn` dan. Pernyataan kebijakan memungkinkan CloudTrail untuk menggunakan kunci KMS untuk menghasilkan kunci data yang digunakan untuk mengenkripsi jejak. Kondisi `aws:SourceArn` dan `kms:EncryptionContext:`*`context-key`* kondisi dievaluasi secara independen. Setiap permintaan untuk menggunakan kunci KMS untuk operasi yang ditentukan harus memenuhi kedua kondisi tersebut.

Untuk membatasi izin layanan ke `finance` jejak di akun contoh (111122223333) dan Wil `us-west-2` ayah, pernyataan kebijakan ini menetapkan kunci `aws:SourceArn` kondisi ke ARN dari jejak tertentu. Pernyataan kondisi menggunakan ArnEquals operator 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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudTrailToEncryptLogs",
      "Effect": "Allow",
```

```
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "kms:GenerateDataKey",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:cloudtrail:us-west-2:111122223333:trail/finance"
          ]
        },
        "StringLike": {
          "kms:EncryptionContext:aws:cloudtrail:arn": [
            "arn:aws:cloudtrail:*:111122223333:trail/*"
          ]
        }
      }
    }
  ]
}
```

ABAC untuk AWS KMS

Attribute-based access control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. AWS KMS mendukung 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, berhati-hatilah dalam memberikan izin kepada prinsipal 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 AWS KMS dukungan untuk ABAC, termasuk contoh, lihat Gunakan alias untuk mengontrol akses ke tombol KMS danGunakan tag untuk mengontrol akses ke tombol KMS.
- Untuk informasi lebih umum tentang penggunaan tag untuk mengontrol akses ke AWS sumber daya, lihat Untuk apa ABAC? AWS dan Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di Panduan Pengguna IAM.

Kunci kondisi 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	Tipe kebijakan	AWS KMS operasi
aws: ResourceTag	Tag (kunci dan nilai) pada kunci KMS cocok dengan tag (kunci dan nilai)	Khusus kebijakan IAM	Operasi sumber daya utama KMS 2

Kunci syarat ABAC	Deskripsi	Tipe kebijakan	AWS KMS operasi
	atau pola tag dalam kebijakan		
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
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 dari kebijakan yang mengandalkan tag: Meskipun Anda dapat menggunakan pola alias dalam kebijakan, setiap alias harus unik di Akun AWS dan Wilayah. Ini memungkinkan Anda menerapkan kebijakan ke kumpulan kunci KMS tertentu tanpa mencantumkan kunci kunci ARNs 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 kunci kunci ARNs 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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Test"
        }
      }
    }
  ]
}
```

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 Gunakan alias untuk mengontrol akses ke tombol KMS dan Gunakan 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 AWS penggunaan khusus Anda. 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 berbagai jenis AWS sumber daya.

  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 ARNs untuk merujuk ke kunci KMS.
- Berikan izin yang sangat terbatas.

  Alias harus unik di wilayah Akun AWS dan. 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, sepertialias/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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
```

```
  ]
}
```

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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:Describe*",
```

```
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "kms:ResourceAliases": [
            "alias/ProjectAlpha",
            "alias/ProjectAlpha_Test",
            "alias/ProjectAlpha_Dev"
          ]
        }
      }
    }
  ]
}
```

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 kemungkinan akan lebih lama daripada penundaan konsistensi akhirnya singkat yang mempengaruhi sebagian besar AWS KMS operasi.

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.

RBAC untuk AWS KMS

Kontrol akses berbasis peran (RBAC) adalah strategi otorisasi yang hanya memberi pengguna izin yang diperlukan untuk melakukan tugas pekerjaan mereka, dan tidak lebih. AWS KMS mendukung RBAC dengan memungkinkan Anda mengontrol akses ke kunci Anda dengan menentukan izin terperinci tentang penggunaan kunci dalam kebijakan utama. Kebijakan kunci menentukan sumber daya, tindakan, efek, prinsip, dan kondisi opsional untuk memberikan akses ke kunci.

Untuk mengimplementasikan RBAC di AWS KMS, sebaiknya pisahkan izin untuk pengguna kunci dan administrator kunci.

Key users

Contoh kebijakan kunci berikut memungkinkan peran `ExampleUserRole` IAM untuk menggunakan kunci KMS.

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

Pengguna kunci Anda mungkin memerlukan izin yang lebih sedikit daripada pengguna dalam contoh ini. Hanya tetapkan izin yang dibutuhkan pengguna. Gunakan pertanyaan berikut untuk membantu Anda menyempurnakan izin lebih lanjut.

- Prinsipal IAM (peran atau pengguna) mana yang membutuhkan akses ke kunci?
- Tindakan apa yang harus dilakukan setiap kepala sekolah dengan kunci? Misalnya, apakah prinsipal hanya memerlukan izin Enkripsi dan Tanda tangani?
- Apakah pengguna itu manusia atau AWS layanan? Jika ini adalah AWS layanan, Anda dapat menggunakan kunci kondisi untuk membatasi penggunaan kunci ke AWS layanan tertentu.

Key administrators

Contoh kebijakan kunci berikut memungkinkan peran `ExampleAdminRole` IAM untuk mengelola kunci KMS.

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

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

Administrator kunci Anda mungkin memerlukan izin yang lebih sedikit daripada administrator dalam contoh ini. Hanya tetapkan izin yang dibutuhkan administrator kunci Anda.

Hanya berikan pengguna izin yang mereka butuhkan untuk memenuhi peran mereka. Izin pengguna dapat bervariasi tergantung pada apakah kunci digunakan dalam lingkungan pengujian atau produksi. Jika Anda menggunakan izin yang tidak terlalu ketat di lingkungan non-produksi tertentu, terapkan proses untuk menguji kebijakan sebelum dirilis ke produksi.

Pelajari selengkapnya

- Identitas IAM (pengguna, grup pengguna, dan peran)
- Jenis kontrol akses

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. Cross-account akses memerlukan izin dalam kebijakan kunci kunci KMS dan dalam kebijakan IAM di akun pengguna eksternal.

Cross-account izin 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 Cross-account penggunaan di AWS KMS izin danMenggunakan kunci KMS di akun lain. Ada juga bagian Cross-account penggunaan di setiap deskripsi API di Referensi AWS Key Management Service API.

> **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 Konsol Manajemen AWS atau menggunakan PutKeyPolicyoperasi CreateKeyatau.

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 Anda AWS 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

- 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 contohSid,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. Misalnya, Anda mungkin ingin memberi pengguna izin untuk mendekripsi tetapi tidak mengenkripsi, atau izin untuk melihat kunci KMS tetapi tidak menggunakannya. Untuk daftar izin pada kunci KMS, lihat. AWS KMS izin

Menyetel kebijakan kunci saat Anda membuat kunci KMS

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.

Saat Anda membuat kunci KMS di Konsol Manajemen AWS, 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 akun eksternal, AWS KMS tambahkan dua pernyataan ke kebijakan utama. 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. Pernyataan kebijakan kunci kedua memungkinkan akun eksternal untuk membuat, melihat, dan mencabut hibah pada kunci KMS, tetapi hanya jika permintaan berasal dari AWS layanan 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 layanan AWS

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 AWS layanan yang terintegrasi dengannya AWS KMS, Anda mungkin perlu menambahkan izin ke kebijakan kunci atau kebijakan IAM. Lihat perinciannya di Mengizinkan penggunaan kunci KMS eksternal dengan Layanan AWS.

Untuk informasi selengkapnya tentang bekerja dengan kebijakan IAM, lihat Kebijakan IAM.

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 dengannya AWS 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 berbeda Akun AWS, Anda dapat menggunakan kunci KMS di Konsol Manajemen AWS, AWS SDK, AWS CLI dan. Alat AWS untuk 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 KMS Konsol 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.

Kontrol akses ke tombol 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 kepala sekolah untuk mereplikasi kunci Multi-wilayah hanya ke yang membutuhkannya. Wilayah AWS Berikan izin kunci multi-Wilayah hanya kepada perwakilan yang membutuhkan dan hanya untuk tugas-tugas yang memerlukannya.

Anda dapat menggunakan kebijakan utama, kebijakan IAM, dan hibah untuk mengizinkan prinsipal IAM mengelola dan menggunakan kunci Multi-wilayah di Anda. Akun AWS 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.

Untuk mendukung kunci Multi-region, AWS KMS gunakan peran terkait layanan IAM. Peran ini memberi AWS KMS izin yang dibutuhkan untuk menyinkronkan properti bersama. Untuk informasi selengkapnya, lihat Otorisasi AWS KMS untuk menyinkronkan kunci Multi-region.

Topik

- Basic otorisasi untuk kunci multi-Wilayah
- Mengotorisasi administrator dan pengguna 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 utama bukan properti bersama dari kunci Multi-wilayah. AWS KMS tidak menyalin atau menyinkronkan kebijakan utama di antara kunci Multi-wilayah terkait.

  Saat Anda membuat kunci replika di AWS KMS konsol, konsol akan menampilkan kebijakan kunci saat ini dari kunci utama sebagai kenyamanan. Anda dapat menggunakan kebijakan kunci ini, mengeditnya, atau menghapus serta menggantinya. Tetapi bahkan jika Anda menerima kebijakan kunci utama 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 secara Wilayah AWS 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.
- Hibah — AWS KMS hibah 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 kunci ARNs Multi-region terkait memiliki partisi, akun, dan ID kunci yang sama, tetapi 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 Anda Akun 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 kunci Multi-wilayah terkait, gunakan AWS KMS peran terkait layanan IAM. AWS KMS membuat peran terkait layanan di Akun AWS setiap kali Anda membuat kunci utama Multi-wilayah. (Jika perannya ada, AWS KMS akan membuatnya ulang, yang tidak memiliki efek berbahaya.) Peran ini berlaku di semua Wilayah. AWS KMS Untuk 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 kunci Multi-wilayah, melampirkan kebijakan untuk kunci tersebut, dan peran terkait layanan untuk kunci Multi-wilayah.

JSON

```
{
  "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.

JSON

```
{
  "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 memungkinkan prinsipal untuk memperbarui Region utama kunci Multi-region di Akun AWS, tetapi hanya jika Region primer baru adalah salah satu 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"
  }
}
```

Menentukan akses ke AWS KMS keys

Untuk menentukan sepenuhnya siapa atau apa yang saat ini memiliki akses ke suatu AWS 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

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 Konsol Manajemen AWS 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 utama”.

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 dengan Konsol Manajemen AWS, 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 yang disebutkan KMSKeyAdmins dalam 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 AWS layanan 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 lebih lanjut tentang kebijakan AWS KMS utama, lihatKebijakan 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 Pemecahan Masalah AWS KMS izin.

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 Konsol Manajemen AWS dan kemudian buka IAM Policy Simulator dihttps://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 AWS KMS tindakan tertentu, untuk Pilih tindakan, pilih tindakan yang akan disimulasikan. Untuk mensimulasikan semua AWS KMS tindakan, 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 dalam Akun 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 `ActionNames` daftar, tentukan setiap tindakan AWS KMS API yang akan disimulasikan. Untuk mensimulasikan semua tindakan AWS KMS API, gunakan`kms:*`. Untuk menguji tindakan AWS KMS API individual, mendahului 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 menyertakan nama operasi AWS KMS API 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. Untuk hibah yang dibuat dengan`GranteePrincipal`, cari `"GranteePrincipal"` bidang `ListGrants` sebagai tanggapan. Untuk hibah yang dibuat dengan`GranteeServicePrincipal`, cari `"GranteeServicePrincipal"` bidang `ListGrants` sebagai tanggapan.

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. Dalam contoh berikut, pokok penerima hibah adalah yang lain. Akun AWS

```
{"Grants": [{
  "Operations": ["Encrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "",
  "GranteePrincipal": "arn:aws:iam::444455556666:root",
  "GrantId": "f271e8328717f8bde5d03f4981f06a6b3fc18bcae2da12ac38bd9186e7925d11",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151269E9
}]}
```

Contoh berikut menunjukkan hibah yang dibuat dengan`GranteeServicePrincipal`. Perhatikan bahwa respons menyertakan `GranteeServicePrincipal` bidang, bukan`GranteePrincipal`, dan `Constraints` bidang berisi `SourceArn` nilai yang membatasi hibah ke AWS sumber daya tertentu.

```
{"Grants": [{
  "Operations": ["Encrypt", "Decrypt", "GenerateDataKey", "DescribeKey"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "",
  "GranteeServicePrincipal": "service-name.amazonaws.com",
  "GrantId": "a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "RetiringServicePrincipal": "service-name.amazonaws.com",
  "Constraints": {
    "SourceArn": "arn:aws:dynamodb:us-east-1:111122223333:table/ExampleTable"
  },
  "CreationDate": 1.718567315E9
}]}
```

Konteks enkripsi

> **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.

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. Anda dapat menyisipkan konteks enkripsi dalam `Encrypt` operasi AWS KMS untuk meningkatkan otorisasi dan auditabilitas panggilan dekripsi AWS KMS API Anda. AWS KMS menggunakan konteks enkripsi sebagai data otentikasi tambahan (AAD) untuk mendukung enkripsi yang diautentikasi. Konteks enkripsi terikat secara kriptografis ke ciphertext sehingga konteks enkripsi yang sama diperlukan untuk mendekripsi data.

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. Kunci konteks enkripsi dan nilainya dapat berupa string

arbitrer dengan. `aws` Nilai-nilai ini kontras tag yang AWS dihasilkan seperti aws:cloudformation:stack-name. Untuk informasi selengkapnya, lihat km:: EncryptionContextkunci-konteks

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

Aturan konteks enkripsi

AWS KMS memberlakukan 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 membuat hibah, Anda dapat menyertakan batasan hibah yang menetapkan ketentuan untuk izin hibah. AWS KMS mendukung dua kendala hibah konteks enkripsi, `EncryptionContextEquals` dan`EncryptionContextSubset`, 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 KMS digunakan 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.

Menguji izin Anda

Untuk menggunakannya AWS 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 ke AWS 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 KMS mendukung penggunaan `DryRun` parameter dalam tindakan API tertentu:

- CreateGrant
- Dekripsi
- DeriveSharedSecret
- 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 selengkapnya tentang AWS KMS harga, lihat AWS Key Management Service Harga.

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 Pencatatan log AWS KMS Panggilan 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 KMS panggilan 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 \
```

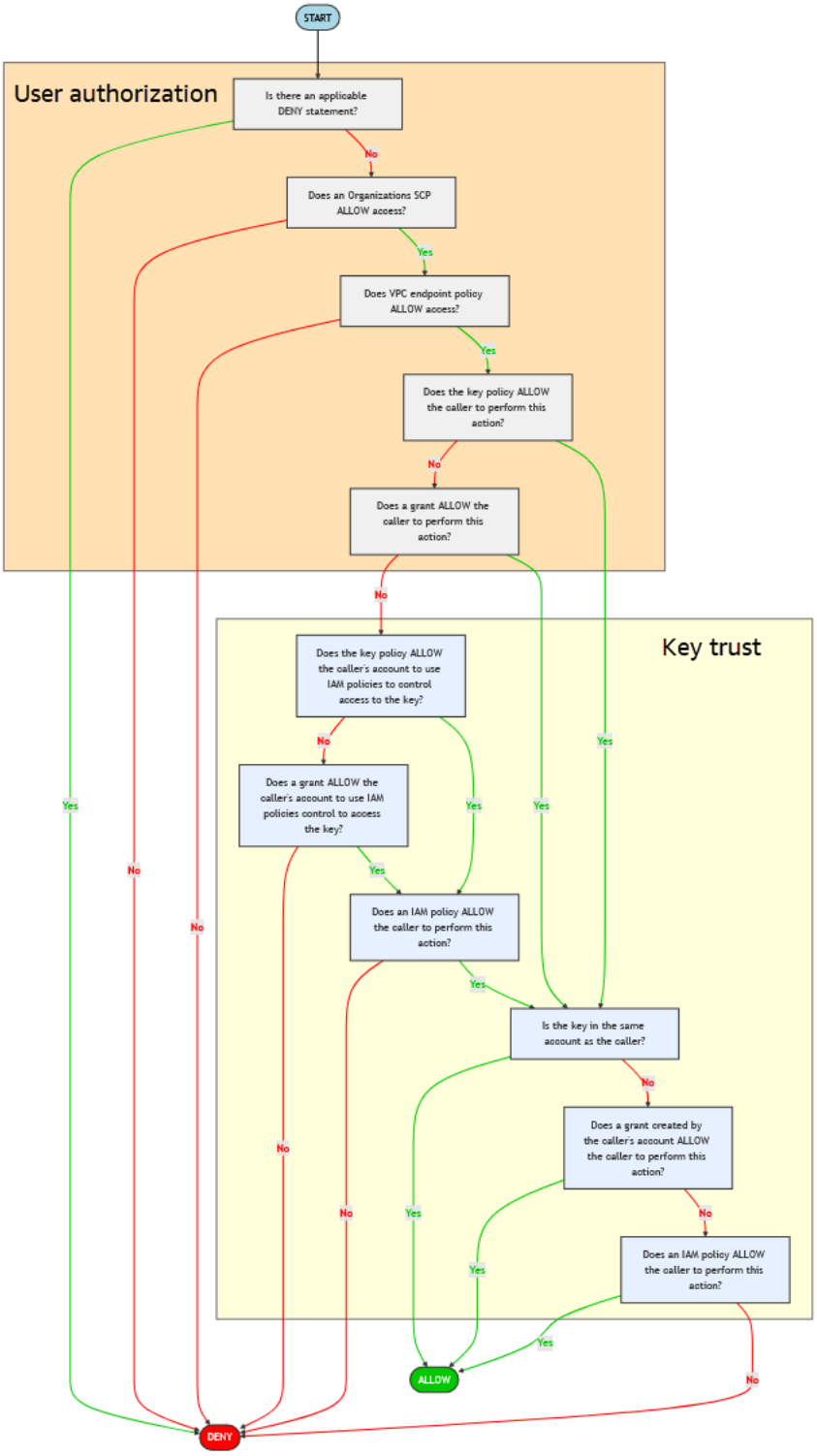
```
    --operations Decrypt \
    --dry-run
```

Pemecahan Masalah AWS KMS izin

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 KMS mengevaluasi mekanisme kebijakan ini bersama-sama untuk menentukan apakah akses ke kunci KMS diperbolehkan atau ditolak. Untuk melakukan ini, AWS KMS gunakan proses yang mirip dengan yang digambarkan dalam diagram 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 di Anda 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.

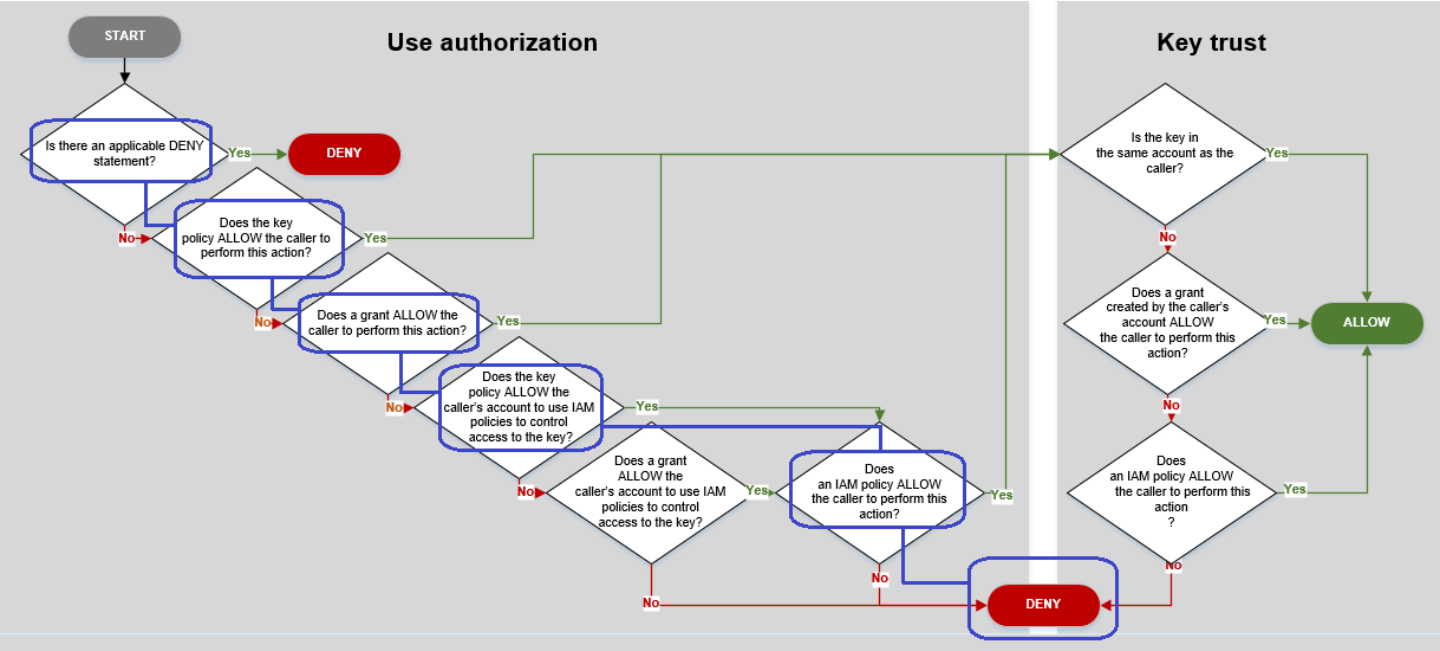
Contoh Izin

- 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 111122223333. Akun AWS Dia ditolak akses ke kunci KMS dalam hal yang sama Akun 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.

  JSON

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

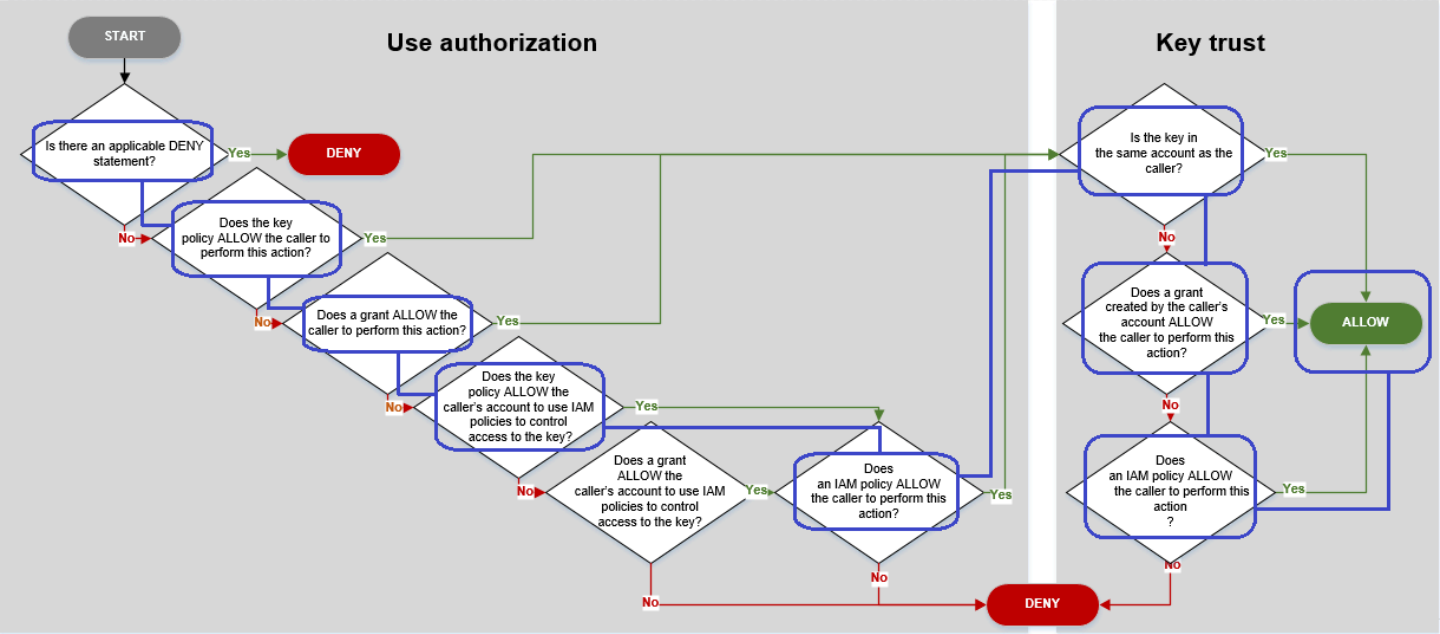
- Namun, tidak ada kebijakan kunci, kebijakan IAM, atau hibah yang memberikan izin kepada Alice untuk menggunakan kunci KMS. Oleh karena itu, Alice ditolak izinnya untuk menggunakan kunci KMS.

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?.

  JSON

```
{
    "Id": "key-policy-acct-2",
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PermissionUseIAMpolicies",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::444455556666:root"
            },
```

```
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "AllowAccount1UseThisKMSKeys",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

- 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`.

  Cross-account Kebijakan IAM seperti ini hanya efektif ketika 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?.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:DescribeKey"
            ],
            "Resource": [
            "arn:aws:kms:us-
west-2:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            ]
        }
    ]
}
```

- 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 glosarium kontrol akses

Topik berikut menjelaskan istilah dan konsep penting dalam kontrol AWS KMS akses.

Autentikasi

Otentikasi adalah proses verifikasi identitas Anda. Untuk mengirim permintaan AWS KMS, Anda harus masuk AWS menggunakan AWS kredensil 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

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensil dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensional. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat Cara masuk ke Akun AWS Anda dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensil dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat Apa itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara di Panduan Pengguna IAM.

Grup IAM menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan beralih dari pengguna ke peran IAM (konsol) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat Pilih 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.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

AWS KMS sumber daya

Pada tahun AWS KMS, sumber daya utama adalah AWS KMS key. AWS KMS juga mendukung alias, sumber daya independen yang menyediakan nama ramah untuk kunci KMS. Beberapa AWS KMS operasi memungkinkan Anda untuk menggunakan alias untuk mengidentifikasi kunci KMS.

Setiap instance kunci atau alias KMS memiliki Amazon Resource Name (ARN) unik dengan format standar. Dalam AWS KMS sumber daya, nama AWS layanannya 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 serangkaian operasi API untuk bekerja dengan AWS KMS sumber daya Anda. Untuk informasi selengkapnya tentang mengidentifikasi kunci KMS dalam operasi Konsol Manajemen AWS dan AWS KMS API, lihatPengidentifikasi kunci (KeyId). Untuk daftar AWS KMS operasi, lihat Referensi AWS Key Management Service API.

Buat kunci KMS

Anda dapat membuat AWS KMS keys di Konsol Manajemen AWS, atau dengan menggunakan CreateKeyoperasi atau AWS::KMS::Key AWS CloudFormation sumber daya. Selama proses ini, Anda menetapkan kebijakan kunci untuk kunci KMS, yang dapat Anda ubah kapan saja. Anda juga memilih nilai berikut yang menentukan jenis kunci KMS yang Anda buat. Anda tidak dapat mengubah properti ini setelah kunci KMS dibuat.

Jenis kunci KMS

Tipe kunci adalah properti yang menentukan jenis kunci kriptografi apa yang dibuat. AWS KMS menawarkan tiga jenis kunci untuk melindungi data:

- Kunci simetris Advanced Encryption Standard (AES)

  Tombol 256-bit yang digunakan di bawah mode Galois Counter Mode (GCM) AES untuk memberikan otentikasi data di bawah encryption/decryption ukuran 4KB. Ini adalah jenis kunci yang paling umum dan digunakan untuk melindungi kunci enkripsi data lain yang digunakan dalam aplikasi Anda dan dengan Layanan AWS itu mengenkripsi data Anda atas nama Anda.

- RSA, kurva elips, atau (hanya Wilayah SM2 Tiongkok) kunci asimetris

  Tombol ini tersedia dalam berbagai ukuran dan mendukung banyak algoritma. Mereka dapat digunakan untuk enkripsi dan dekripsi, menandatangani dan memverifikasi, atau memperoleh operasi rahasia bersama tergantung pada pilihan algoritme.

- Kunci simetris untuk melakukan operasi kode otentikasi pesan berbasis hash (HMAC)

  Kunci ini adalah kunci 256-bit yang digunakan untuk menandatangani dan memverifikasi operasi.

  Kunci KMS tidak dapat diekspor dari layanan dalam teks biasa. Mereka dihasilkan oleh dan hanya dapat digunakan dalam modul keamanan perangkat keras (HSMs) yang digunakan oleh layanan. Ini adalah properti keamanan dasar AWS KMS untuk memastikan bahwa kunci tidak dikompromikan.

Penggunaan kunci

Penggunaan kunci adalah properti yang menentukan operasi kriptografi yang didukung kunci. Kunci KMS dapat memiliki penggunaan kunci`ENCRYPT_DECRYPT`,, `SIGN_VERIFY`GENERATE_VERIFY_MAC, atau`KEY_AGREEMENT`. Setiap kunci KMS hanya

dapat memiliki satu penggunaan kunci. Ini mengikuti praktik terbaik penggunaan utama menurut National Institute of Standards and Technology (NIST) Publikasi Khusus 800-57 Rekomendasi untuk Manajemen Utama, Bagian 5.2, Penggunaan utama. Menggunakan kunci KMS untuk lebih dari satu jenis operasi membuat produk dari kedua operasi lebih rentan terhadap serangan.

Spesifikasi kunci

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

Untuk kunci KMS, spesifikasi kunci menentukan apakah kunci KMS simetris atau asimetris. Ini juga menentukan jenis bahan utamanya, dan algoritma yang didukungnya.

Spesifikasi kunci default, SYMMETRIC_DEFAULT, mewakili kunci enkripsi simetris 256-bit. Untuk penjelasan rinci tentang semua spesifikasi utama yang didukung, lihatReferensi spesifikasi kunci.

Asal bahan utama

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.

Setiap kunci KMS menyertakan referensi ke materi utamanya dalam metadatanya. Asal material kunci kunci KMS enkripsi simetris dapat bervariasi. Anda dapat menggunakan materi kunci yang AWS KMS menghasilkan, materi kunci yang dihasilkan di toko kunci khusus, atau mengimpor materi kunci Anda sendiri.

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

Kunci KMS dapat memiliki salah satu nilai asal material utama berikut:`AWS_KMS`, `EXTERNAL` (bahan kunci impor), `AWS_CLOUDHSM` (kunci KMS di toko AWS CloudHSM kunci), atau `EXTERNAL_KEY_STORE` (kunci KMS di toko kunci eksternal).

Topik

- Izin untuk membuat kunci KMS
- Memilih jenis kunci KMS apa yang akan dibuat
- Buat kunci KMS enkripsi simetris

- Buat kunci KMS asimetris
- Buat kunci HMAC KMS
- Buat kunci utama Multi-wilayah
- Buat kunci replika Multi-wilayah
- Buat kunci KMS dengan bahan kunci yang diimpor
- Buat kunci KMS di toko AWS CloudHSM kunci
- Buat kunci KMS di toko kunci eksternal

Izin untuk membuat kunci KMS

Untuk membuat kunci KMS di konsol atau dengan menggunakan APIs, 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. Lihat perinciannya di 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. Lihat perinciannya di Kontrol akses ke tombol 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.

Memilih jenis kunci KMS apa yang akan dibuat

Jenis kunci KMS yang Anda buat sangat tergantung pada bagaimana Anda berencana untuk menggunakan kunci KMS, persyaratan keamanan Anda, dan persyaratan otorisasi Anda. Jenis kunci dan penggunaan kunci dari kunci KMS menentukan operasi kriptografi apa yang dapat dilakukan oleh kunci. 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.

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.

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 menelepon AWS 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, key pair ML-DSA, atau key pair (hanya Wilayah China). SM2 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. Gunakan key pair ML-DSA saat bermigrasi dari kunci RSA atau ECC ke kunci pasca-kuantum. Namun, Anda mungkin perlu menggunakan spesifikasi kunci tertentu dan algoritma penandatanganan untuk mendukung pengguna yang memverifikasi tanda tangan di luar. AWS

Enkripsi dengan pasangan kunci asimetris

Untuk mengenkripsi data dengan asymmetric key pair, Anda harus menggunakan kunci KMS asimetris dengan spesifikasi kunci RSA atau spesifikasi kunci (khusus Wilayah Tiongkok SM2). 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 AWS KMS kunci atau hibah 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, hubungi 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. Referensi spesifikasi kunci

Untuk menghindari kesalahan ini, siapa pun yang menggunakan kunci publik di luar AWS KMS harus menyimpan konfigurasi kunci. AWS KMS Konsol dan GetPublicKeyrespons memberikan informasi yang harus Anda sertakan saat Anda membagikan kunci publik.

Turunkan rahasia bersama

Untuk mendapatkan rahasia bersama, gunakan kunci KMS dengan kurva elips standar NIST atau bahan kunci SM2(hanya Wilayah Tiongkok). AWS KMS menggunakan Elliptic Curve Cryptography Cofactor Diffie-Hellman Primitive (ECDH) untuk membuat kesepakatan kunci antara dua rekan dengan memperoleh rahasia bersama dari kurva elips pasangan kunci publik-pribadi mereka. Anda dapat menggunakan rahasia bersama mentah yang dikembalikan DeriveSharedSecretoperasi untuk mendapatkan kunci simetris yang dapat mengenkripsi dan mendekripsi data yang dikirim antara dua pihak, atau menghasilkan dan memverifikasi. HMACs AWS KMS merekomendasikan agar Anda mengikuti rekomendasi NIST untuk derivasi kunci saat menggunakan rahasia bersama mentah untuk mendapatkan kunci simetris.

Hasilkan dan verifikasi kode HMAC

Untuk membuat dan memverifikasi kode otentikasi pesan berbasis hash, gunakan kunci HMAC KMS. Saat Anda membuat kunci HMAC AWS KMS, AWS KMS membuat dan melindungi materi kunci Anda dan memastikan bahwa Anda menggunakan algoritme 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 AWS layanan yang terintegrasi dengan AWS KMS, lihat dokumentasi untuk layanan tersebut. AWS layanan 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.

Buat kunci KMS enkripsi simetris

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 file Layanan AWS.

Anda dapat membuat kunci KMS enkripsi simetris di AWS KMS konsol, dengan menggunakan CreateKeyAPI, atau dengan menggunakan template. AWS::KMS::Key CloudFormation

Spesifikasi kunci default, SYMMETRIC_DEFAULT, adalah spesifikasi kunci untuk kunci KMS enkripsi simetris. Ketika Anda memilih jenis kunci simetris dan Enkripsi dan dekripsi penggunaan kunci di AWS KMS konsol, itu 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.

Untuk informasi tentang kuota yang berlaku untuk kunci KMS, lihat. Kuota

Menggunakan AWS KMS konsol

Anda dapat menggunakan Konsol Manajemen AWS 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 Konsol Manajemen AWS 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.
6. Dalam Penggunaan kunci, opsi Enkripsi dan dekripsi dipilih untuk Anda.
7. Pilih Berikutnya.
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 Gunakan 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. Konsol Manajemen AWS 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 pada halaman detail untuk kunci KMS di Konsol Manajemen AWS 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 Gunakan tag untuk mengontrol akses ke tombol KMS.

    Saat Anda menambahkan tag ke AWS sumber daya Anda, AWS buat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan berdasarkan tag. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tag di AWS KMS dan. ABAC untuk AWS KMS

11. Pilih Berikutnya.

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

> **Catatan**
>
> 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. Lihat perinciannya di the section called “Kebijakan kunci default”.
> 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.
> AWS KMS Konsol menambahkan administrator kunci ke kebijakan kunci di bawah pengenal `"Allow access for Key Administrators"` pernyataan. Memodifikasi pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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

> **Catatan**
>
> 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.
> AWS KMS Konsol menambahkan pengguna kunci ke kebijakan kunci di bawah pengidentifikasi pernyataan `"Allow use of the key"` dan`"Allow attachment of persistent resources"`. Memodifikasi pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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 nomor Akun AWS identifikasi 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 Berikutnya.
18. Tinjau pernyataan kebijakan utama untuk kunci tersebut. Untuk membuat perubahan pada kebijakan utama, pilih Edit.
19. Pilih Berikutnya.
20. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
21. Pilih Selesai untuk membuat kunci KMS.

Menggunakan AWS KMS API

Anda dapat menggunakan CreateKeyoperasi untuk membuat semua AWS KMS keys jenis. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI). Untuk contoh dalam beberapa bahasa pemrograman, lihat Gunakan `CreateKey` dengan AWS SDK atau CLI.

> **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 membuat 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 impor, dan kunci di toko kunci khusus. Untuk membuat kunci data untuk enkripsi sisi klien, gunakan operasi. GenerateDataKey

`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 Akses dan izin kunci KMS

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

JSON

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

Buat kunci KMS asimetris

Anda dapat membuat kunci KMS asimetris di AWS KMS konsol, dengan menggunakan CreateKeyAPI, atau dengan menggunakan template. AWS::KMS::Key CloudFormation Kunci KMS asimetris mewakili key pair publik dan pribadi yang dapat digunakan untuk enkripsi, penandatanganan, atau menurunkan rahasia bersama. Kunci privat tetap ada di dalam AWS KMS. Untuk mengunduh kunci publik untuk digunakan di luar AWS KMS, lihatUnduh kunci publik.

Saat Anda membuat kunci KMS asimetris, Anda harus memilih spesifikasi kunci. 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 penjelasan rinci tentang semua spesifikasi utama yang didukung, lihatReferensi spesifikasi kunci.

AWS layanan yang terintegrasi dengan AWS KMS tidak mendukung kunci KMS asimetris. Jika Anda ingin membuat kunci KMS yang mengenkripsi data yang Anda simpan atau kelola dalam suatu AWS layanan, buat kunci KMS enkripsi simetris.

Untuk informasi tentang izin yang diperlukan untuk membuat kunci KMS, lihat. Izin untuk membuat kunci KMS

Menggunakan AWS KMS konsol

Anda dapat menggunakan tombol Konsol Manajemen AWS 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 Konsol Manajemen AWS 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.
6. Untuk membuat kunci KMS asimetris untuk enkripsi kunci publik, dalam penggunaan Kunci, pilih Enkripsi dan dekripsi.

   Untuk membuat kunci KMS asimetris untuk menandatangani pesan dan memverifikasi tanda tangan, dalam Penggunaan kunci, pilih Masuk dan verifikasi.

   Untuk membuat kunci KMS asimetris untuk mendapatkan rahasia bersama, dalam penggunaan Kunci, pilih Perjanjian kunci.

   Untuk bantuan memilih nilai penggunaan kunci, lihat Memilih jenis kunci KMS apa yang akan dibuat.
7. Pilih spesifikasi (Spesifikasi kunci) untuk kunci KMS asimetris Anda.
8. Pilih Berikutnya.
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 beberapa. AWS KMS APIs 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. Konsol Manajemen AWS Anda tidak dapat menentukan alias ketika Anda menggunakan CreateKeyoperasi, tetapi Anda dapat menggunakan konsol atau CreateAliasoperasi untuk membuat alias untuk kunci KMS yang ada. Lihat perinciannya di Alias di AWS KMS.

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 kuncinya adalah `Pending Deletion` atau`Pending Replica Deletion`. Untuk menambah, mengubah, atau menghapus deskripsi kunci terkelola pelanggan yang ada, edit deskripsi pada halaman detail untuk kunci KMS di Konsol Manajemen AWS 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 tag ke AWS sumber daya Anda, AWS buat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan berdasarkan tag. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tag di AWS KMS dan. ABAC untuk AWS KMS

12. Pilih Berikutnya.

13. Pilih pengguna IAM dan peran yang dapat mengelola kunci KMS.

> **Catatan**
>
> 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. Lihat perinciannya di the section called “Kebijakan kunci default”.
> 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.
> AWS KMS Konsol menambahkan administrator kunci ke kebijakan kunci di bawah pengenal `"Allow access for Key Administrators"` pernyataan. Memodifikasi

pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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.

> **Catatan**
>
> 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.
> AWS KMS Konsol menambahkan pengguna kunci ke kebijakan kunci di bawah pengidentifikasi pernyataan `"Allow use of the key"` dan`"Allow attachment of persistent resources"`. Memodifikasi pengidentifikasi pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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 Berikutnya.
19. Tinjau pernyataan kebijakan utama untuk kunci tersebut. Untuk membuat perubahan pada kebijakan utama, pilih Edit.
20. Pilih Berikutnya.

21. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
22. Pilih Selesai untuk membuat kunci KMS.

Menggunakan AWS KMS API

Anda dapat menggunakan CreateKeyoperasi untuk membuat asimetris AWS 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 `KeyUsage` nilai ENCRYPT_DECRYPT, SIGN_VERIFY, atau KEY_AGREEMENT. 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.

Buat key pair KMS asimetris untuk enkripsi publik

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
    }
}
```

Buat key pair KMS asimetris untuk penandatanganan dan verifikasi

Contoh perintah berikut membuat kunci KMS asimetris yang mewakili sepasang kunci ECC 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"
    }
}
```

Buat key pair KMS asimetris untuk mendapatkan rahasia bersama

Contoh perintah berikut membuat kunci KMS asimetris yang mewakili sepasang kunci ECDH yang digunakan untuk menurunkan rahasia bersama. Anda tidak dapat membuat pasangan kunci kurva elips untuk enkripsi dan dekripsi.

```
$ aws kms create-key --key-spec ECC_NIST_P256 --key-usage KEY_AGREEMENT
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "CreationDate": "2023-12-27T19:10:15.063000+00:00",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "KEY_AGREEMENT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "ECC_NIST_P256",
        "KeySpec": "ECC_NIST_P256",
        "KeyAgreementAlgorithms": [
            "ECDH"
        ],
        "MultiRegion": false
    }
}
```

Buat kunci HMAC KMS

Anda dapat membuat kunci HMAC KMS di AWS KMS konsol, dengan menggunakan CreateKeyAPI, atau dengan menggunakan template. AWS::KMS::Key CloudFormation

Saat Anda membuat kunci HMAC KMS, Anda harus memilih spesifikasi kunci. AWS KMS mendukung 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.

Untuk informasi tentang izin yang diperlukan untuk membuat kunci KMS, lihat. Izin untuk membuat kunci KMS

Menggunakan AWS KMS konsol

Anda dapat menggunakan tombol Konsol Manajemen AWS 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 Konsol Manajemen AWS 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.

   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. Konsol Manajemen AWS Anda tidak dapat menentukan alias ketika Anda menggunakan CreateKeyoperasi, tetapi Anda dapat menggunakan konsol atau CreateAliasoperasi untuk membuat alias untuk kunci KMS yang ada. Lihat perinciannya di Alias di AWS KMS.

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 pada halaman detail untuk kunci KMS di Konsol Manajemen AWS dalam Konsol Manajemen AWS atau menggunakan 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 tag ke AWS sumber daya Anda, AWS buat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan berdasarkan tag. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tag di AWS KMS dan. ABAC untuk AWS KMS

13. Pilih Berikutnya.

14. Pilih pengguna IAM dan peran yang dapat mengelola kunci KMS.

> **Catatan**
>
> 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. Lihat perinciannya di 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.
> AWS KMS Konsol menambahkan administrator kunci ke kebijakan kunci di bawah pengenal `"Allow access for Key Administrators"` pernyataan. Memodifikasi pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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.

> **Catatan**
>
> 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.
> AWS KMS Konsol menambahkan pengguna kunci ke kebijakan kunci di bawah pengidentifikasi pernyataan `"Allow use of the key"` dan`"Allow attachment of persistent resources"`. Memodifikasi pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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 Berikutnya.
20. Tinjau pernyataan kebijakan utama untuk kunci tersebut. Untuk membuat perubahan pada kebijakan utama, pilih Edit.
21. Pilih Berikutnya.
22. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
23. Pilih Selesai untuk membuat kunci HMAC KMS.

Menggunakan AWS KMS API

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
    }
}
```

Buat kunci utama Multi-wilayah

Anda dapat membuat kunci utama Multi-region di AWS KMS konsol atau dengan menggunakan AWS KMS API. Anda dapat membuat kunci utama di Wilayah AWS mana saja yang 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.

> **Note**
>
> Saat membuat kunci utama Multi-wilayah, pertimbangkan dengan cermat pengguna dan peran IAM yang Anda pilih untuk dikelola dan gunakan kunci tersebut. 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.

Menggunakan AWS KMS 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 Buat 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 Konsol Manajemen AWS 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, lihatBuat kunci KMS,the section called “Buat kunci KMS asimetris”, atauthe section called “Buat kunci HMAC KMS”.
7. Perluas Opsi lanjutan.

8. Di bawah Key material origin, untuk 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 Regionalitas, pilih Multi-Region key.

   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 KMS tidak 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 Gunakan 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 KMS tidak 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 KMS tidak 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 Gunakan tag untuk mengontrol akses ke tombol KMS.
13. Pilih pengguna IAM dan peran yang dapat mengelola kunci utama.

> **Catatan**
>
> - 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 KMS tidak menyinkronkan kebijakan kunci kunci Multi-region. Anda dapat mengubah kebijakan kunci KMS kapan saja.
> - Saat membuat kunci primer Multi-wilayah, pertimbangkan untuk menggunakan kebijakan kunci default yang dihasilkan oleh konsol. Jika Anda mengubah kebijakan ini, konsol tidak akan memberikan langkah-langkah untuk memilih administrator kunci dan pengguna saat membuat kunci replika, juga tidak akan menambahkan pernyataan kebijakan terkait. Akibatnya, Anda harus menambahkan ini secara manual.
> - AWS KMS Konsol menambahkan administrator kunci ke kebijakan kunci di bawah pengenal `"Allow access for Key Administrators"` pernyataan. Memodifikasi pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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.

> **Catatan**
>
> AWS KMS Konsol menambahkan pengguna kunci ke kebijakan kunci di bawah pengidentifikasi pernyataan `"Allow use of the key"` dan`"Allow attachment of persistent resources"`. Memodifikasi pengidentifikasi pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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 Berikutnya.
19. Tinjau pernyataan kebijakan utama untuk kunci tersebut. Untuk membuat perubahan pada kebijakan utama, pilih Edit.
20. Pilih Berikutnya.
21. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
22. Pilih Selesai untuk membuat kunci utama Multi-wilayah.

Menggunakan AWS KMS API

Untuk membuat kunci primer Multi-wilayah, gunakan CreateKeyoperasi. Gunakan parameter `MultiRegion` dengan nilai `True`.

Misalnya, perintah berikut membuat kunci utama Multi-region di pemanggil (Wilayah 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": [ ]
      }
    }
}
```

Buat kunci replika Multi-wilayah

Anda dapat membuat kunci replika Multi-region di AWS KMS konsol, dengan menggunakan ReplicateKeyoperasi, atau dengan menggunakan template. AWS::KMS::ReplicaKey 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` jika Anda membuat kunci Multi-wilayah

dengan materi kunci yang diimpor) setelah beberapa detik saat 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.

Untuk menggunakan AWS CloudFormation template untuk membuat kunci replika, lihat AWS::KMS::ReplicaKeydi Panduan AWS CloudFormation Pengguna.

Langkah 1: Pilih Wilayah replika

Anda biasanya memilih untuk mereplikasi kunci Multi-wilayah menjadi Wilayah AWS berdasarkan 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 AWS KMS persyaratan untuk Wilayah replika. 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.

Langkah 2: Buat kunci replika

> **Note**
>
> Saat membuat kunci replika, pertimbangkan dengan cermat pengguna dan peran IAM yang Anda pilih untuk dikelola dan gunakan tombol replika. 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.

Menggunakan AWS KMS konsol

Di AWS KMS konsol, Anda dapat membuat satu atau banyak replika kunci utama 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 kunci primer, AWS KMS tidak menjaga nilai-nilai ini disinkronkan.

> **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 Konsol Manajemen AWS 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 AWS partisi yang sama dengan kunci utama. 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 KMS tidak 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 Gunakan 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 KMS tidak 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 KMS tidak menyinkronkan tag kunci Multi-wilayah.

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

11. Pilih pengguna IAM dan peran yang dapat mengelola kunci replika.

> **Catatan**
>
> - Jika Anda mengubah kebijakan kunci default saat membuat kunci primer Multi-wilayah, konsol tidak akan meminta Anda untuk memilih administrator kunci atau pengguna kunci (langkah 11-15) selama pembuatan kunci replika. Dalam hal ini, Anda harus menambahkan izin yang diperlukan secara manual untuk administrator kunci dan pengguna ke kebijakan kunci dengan memilih Edit di langkah kebijakan kunci Edit (Langkah 17).
> - 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 KMS tidak menyinkronkan kebijakan utama. Anda dapat mengubah kebijakan kunci kunci KMS kapan saja.
> - AWS KMS Konsol menambahkan administrator kunci ke kebijakan kunci di bawah pengenal `"Allow access for Key Administrators"` pernyataan. Memodifikasi pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

12. (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.
13. Pilih Berikutnya.
14. Pilih pengguna IAM dan peran yang dapat menggunakan kunci KMS untuk operasi kriptografi.

    > **Catatan**
    >
    > AWS KMS Konsol menambahkan pengguna kunci ke kebijakan kunci di bawah pengidentifikasi pernyataan `"Allow use of the key"` dan`"Allow attachment of persistent resources"`. Memodifikasi pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

15. (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.

16. Pilih Berikutnya.
17. Tinjau pernyataan kebijakan utama untuk kunci tersebut. Untuk membuat perubahan pada kebijakan utama, pilih Edit.
18. Pilih Berikutnya.
19. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
20. Pilih Selesai untuk membuat kunci replika Multi-wilayah.

Menggunakan AWS KMS API

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 tidak ada 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, sepertiKeyId,, KeySpecKeyUsage, 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": []
```

```
}
```

Buat kunci KMS dengan bahan kunci yang diimpor

Bahan kunci yang diimpor memungkinkan Anda melindungi AWS sumber daya Anda di bawah kunci kriptografi yang Anda hasilkan. Gambaran umum berikut menjelaskan cara mengimpor material kunci Anda ke dalam AWS KMS. Untuk detail lebih lanjut 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 AWS KMS kunci terkait. Untuk informasi selengkapnya tentang kompatibilitas, lihatthe section called “Persyaratan untuk bahan kunci yang diimpor”.
2. Unduh kunci publik pembungkus dan token impor — Setelah menyelesaikan langkah 1, unduh kunci publik pembungkus dan token impor. Barang-barang ini melindungi bahan utama Anda saat diimpor ke AWS KMS.

   Pada langkah ini, Anda memilih jenis (“spesifikasi kunci”) kunci pembungkus RSA dan algoritma pembungkus yang akan Anda gunakan untuk mengenkripsi data Anda saat transit ke. AWS KMS Anda dapat memilih spesifikasi kunci pembungkus yang berbeda dan algoritma kunci pembungkus setiap kali Anda mengimpor atau mengimpor kembali bahan kunci yang sama.
3. Enkripsi materi kunci — Gunakan kunci publik pembungkus yang Anda unduh pada 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 kedaluwar AWS KMS sa, menghapusnya, dan kunci KMS menjadi tidak dapat digunakan. Untuk terus menggunakan kunci KMS, Anda harus mengimpor kembali materi kunci yang sama.

   Ketika operasi impor selesai dengan sukses, status kunci kunci KMS berubah dari `PendingImport` menjadi`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 merekam entri saat Anda menghapus materi kunci yang diimpor atau ketika menghapus materi kunci yang AWS KMS kedaluwarsa.

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 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 materi kunci yang diimpor, kepala sekolah 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 algoritma pembungkus tertentu dan spesifikasi kunci pembungkus, gunakan ketentuan kebijakan kms: WrappingAlgorithm dan kms: WrappingKeySpec.
- kms: ImportKeyMaterial (Kebijakan utama atau kebijakan IAM)

- Untuk mengizinkan atau melarang materi kunci yang kedaluwarsa dan mengontrol tanggal kedaluwarsa, gunakan ketentuan ValidTo kebijakan kms: ExpirationModel dan kms:.

Untuk mengimpor kembali materi kunci yang diimpor, kepala sekolah memerlukan izin kms: GetParametersForImport dan kms:. ImportKeyMaterial

Untuk menghapus materi kunci yang diimpor, kepala sekolah memerlukan DeleteImportedKeyMaterial izin kms:.

Misalnya, untuk memberikan `KMSAdminRole` izin contoh untuk mengelola semua aspek kunci KMS dengan materi kunci yang diimpor, sertakan pernyataan kebijakan kunci seperti yang berikut ini dalam kebijakan kunci 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 yang diimpor

Materi kunci yang Anda impor harus kompatibel dengan spesifikasi kunci kunci KMS terkait. Untuk pasangan kunci asimetris, impor hanya kunci pribadi pasangan. AWS KMS mengambil kunci publik dari kunci pribadi.

AWS KMS mendukung spesifikasi kunci berikut untuk kunci KMS dengan bahan kunci yang diimpor.

- Kunci enkripsi simetris
  - Spesifikasi kunci:
    - SIMETRIS_DEFAULT.
  - Persyaratan:

    - 256-bit (32 byte) data biner.
    - Di Wilayah China, itu harus berupa data biner 128 bit (16 byte).
- Kunci HMAC
    - Spesifikasi utama:
        - HMAC_224
        - HMAC_256
        - HMAC_384
        - HMAC_512
    - Persyaratan:
        - Bahan kunci HMAC harus sesuai dengan RFC 2104.
        - Panjang kunci harus setidaknya sama dengan panjang yang ditentukan oleh spesifikasi kunci. Panjang kunci maksimum adalah 1024-bit.
        - Jika materi kunci Anda melebihi 1024 bit, Anda dapat hash materi kunci dan mengimpor output hash. Algoritma hashing harus sesuai dengan spesifikasi kunci dari kunci HMAC KMS yang Anda buat.
    - Contoh:
        - Untuk mengimpor 2048 bit materi kunci ke dalam kunci HMAC_256, pertama-tama hitung SHA-256 hash dari materi kunci 2048-bit, kemudian impor output hash 256-bit yang dihasilkan ke kunci KMS.
    - Panjang kunci yang valid:
        - HMAC_224:224—1024 bit
        - HMAC_256:256—1024 bit
        - HMAC_384:384—1024 bit
        - HMAC_512:512—1024 bit
- Kunci pribadi asimetris RSA
    - Spesifikasi utama:
        - RSA_2048
        - RSA_3072
        - RSA_4096
    - Persyaratan:

    - Kunci pribadi asimetris RSA yang Anda impor harus menjadi bagian dari pasangan kunci 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 BER-encoded DER-encoded dalam format Standar Public-Key Kriptografi (PKCS) #8 yang sesuai dengan RFC 5208.
- Kurva elips kunci pribadi asimetris
    - Spesifikasi utama:
        - ECC_NIST_P256 (secp256r1)
        - ECC_NIST_P384 (secp384r1)
        - ECC_NIST_P521 (secp521r1)
        - ECC_SECG_P256K1 (secp256k1)
        - ECC_NIST_EDWARDS25519 (ed25519)
    - Persyaratan:
        - Kunci pribadi asimetris ECC yang Anda impor harus menjadi bagian dari pasangan kunci yang sesuai dengan RFC 5915.
        - Kurva: NIS T P-256, NIST, NIST P-384, SECP256k1 P-521, NIST Ed25519.
        - Parameter: Kurva bernama saja (kunci ECC dengan parameter eksplisit ditolak).
        - Koordinat titik publik: Dapat dikompresi, tidak terkompresi, atau proyektif.
        - Materi kunci asimetris harus BER-encoded DER-encoded dalam format Standar Public-Key Kriptografi (PKCS) #8 yang sesuai dengan RFC 5208.
- Kunci pribadi asimetris SM2 (hanya Wilayah China)
    - Persyaratan:
        - Kunci pribadi asimetris SM2 yang Anda impor harus menjadi bagian dari pasangan kunci yang sesuai dengan 0003. GM/T
        - Kurva: SM2.
        - Parameter: Kurva bernama saja (kunci SM2 dengan parameter eksplisit ditolak).
        - Koordinat titik publik: Dapat dikompresi, tidak terkompresi, atau proyektif.
        - Materi kunci asimetris harus BER-encoded DER-encoded dalam format Standar Public-Key Kriptografi (PKCS) #8 yang sesuai dengan RFC 5208.

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 asal`EXTERNAL`. 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 bahan kunci adalah`PendingImport`. 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 KMS merekam peristiwa di AWS CloudTrail log Anda saat Anda membuat kunci KMS, mengunduh kunci publik dan token impor, dan mengimpor materi kunci. AWS KMS juga mencatat CloudTrail peristiwa saat Anda menghapus materi kunci yang diimpor atau saat AWS KMS menghapus materi kunci yang kedaluwarsa.

Topik

- Membuat kunci KMS tanpa bahan kunci (konsol)
- Membuat kunci KMS tanpa materi kunci (AWS KMS API)

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. Lihat perinciannya di 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 materi kunci impor, mulailah dengan mengikuti petunjuk 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 bahan kunci yang diimpor.
4. Opsional: Untuk membuat kunci KMS Multi-wilayah dengan bahan kunci yang diimpor, di bawah Regionalitas pilih Multi-Region key.
5. 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 KMS API)

Untuk menggunakan AWS KMS API 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 KMS Kuncinya `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.

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 (disarankan)
- 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 (hanya Wilayah Tiongkok)

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 Tiongkok)	Algoritma pembungkus: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Algoritma pembungkus yang tidak digunakan lagi: RSAES_PKCS1_V1 **Note** Per 10 Oktober 2023, AWS KMS tidak mendukung algoritma pembungkus RSAES_PKCS1_V1_5. Spesifikasi kunci pembungkus: RSA_2048 RSA_3072

Material kunci	Algoritma dan spesifikasi pembungkus yang didukung
	RSA_4096
Kunci pribadi RSA asimetris	Algoritma pembungkus: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 SM2PKE (hanya Wilayah Tiongkok) Spesifikasi kunci pembungkus: RSA_2048 RSA_3072 RSA_4096 SM2 (Hanya Wilayah Tiongkok)
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 Tiongkok) Spesifikasi kunci pembungkus: RSA_2048 RSA_3072 RSA_4096 SM2 (Hanya Wilayah Tiongkok)

Material kunci	Algoritma dan spesifikasi pembungkus yang didukung
Kunci pribadi SM2 asimetris (hanya Wilayah Tiongkok)	Algoritma pembungkus: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (hanya Wilayah Tiongkok) Spesifikasi kunci pembungkus: RSA_2048 RSA_3072 RSA_4096 SM2 (Hanya Wilayah Tiongkok)
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 Tiongkok.

- `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 Tiongkok, 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 Tiongkok, 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 Tiongkok) — Algoritma enkripsi berbasis kurva elips yang ditentukan oleh GM/T OSCCA pada 0003.4-2012.

Topik

- Mengunduh kunci publik pembungkus dan token impor (konsol)
- Mengunduh kunci publik pembungkus dan token impor (AWS KMS API)

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 10
2. Masuk ke Konsol Manajemen AWS 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 sedang 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 utama untuk AWS KMS keys.
7. Pilih tab yang sesuai berdasarkan jenis kunci Anda.
   - Untuk tombol asimetris dan HMAC, pilih tab Bahan kunci.
   - Untuk kunci enkripsi simetris, pilih tab Bahan kunci dan rotasi.
8. Pilih tindakan impor.
   - Untuk kunci asimetris dan HMAC, pilih Impor bahan kunci.
   - Untuk kunci enkripsi simetris, pilih salah satu dari berikut ini:
     - Impor bahan kunci awal (jika belum ada bahan kunci yang diimpor)
     - Impor bahan kunci baru (untuk menambahkan material baru untuk rotasi)
     - Impor ulang materi kunci (tersedia dari menu Tindakan di tabel bahan utama)

> **Note**
>
> Untuk kunci Multi-region, Anda harus terlebih dahulu mengimpor materi kunci baru ke kunci Region primer. Kemudian, impor bahan kunci yang sama ke setiap kunci Region replika.
> Untuk kunci Multi-wilayah utama, tabel Bahan kunci menyertakan kolom status impor Replika yang menampilkan status impor di semua wilayah replika (misalnya, “0 dari 3 yang diimpor”). Pilih nilai status impor replika untuk membuka modal yang menunjukkan status impor untuk setiap wilayah replika. Modal menyediakan tautan materi kunci Impor untuk wilayah replika di mana materi kunci baru belum diimpor.

9. Untuk Pilih spesifikasi tombol pembungkus, pilih konfigurasi untuk kunci KMS Anda. Setelah Anda membuat kunci ini, Anda tidak dapat mengubah spesifikasi kunci.
10. Untuk Memilih algoritme pembungkus, pilih opsi yang akan Anda gunakan untuk mengenkripsi material kunci Anda. Untuk informasi selengkapnya tentang opsi, lihat Pilih Algoritme Pembungkus.
11. 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.
12. 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.
13. 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`

> **Note**
>
> Anda tidak dapat mengimpor materi kunci untuk kunci ML-DSA KMS.

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 *`public key (base64 encoded)`* dalam contoh output), 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 *`import token (base64 encoded)`* dalam contoh output), tempelkan 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.

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 yang diimpor.

> **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 Tiongkok.

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 Tiongkok, 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
```

- SM2 kunci pribadi (hanya Wilayah Tiongkok)

```
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 SM2 publik (khusus Wilayah Tiongkok), gunakan kelas tersebut`SM2OfflineOperationHelper`. Untuk informasi selengkapnya tentang jenis bahan utama yang didukung oleh setiap algoritme pembungkus, lihatthe section called “Pilih algoritme pembungkus”.

> **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, dan kunci HMAC. SM2

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, dan kunci HMAC. SM2

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.

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.

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.

Langkah 4: Impor material kunci

Setelah Anda mengenkripsi materi kunci Anda, Anda dapat mengimpor materi kunci untuk digunakan dengan file AWS 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. Setelah mengimpor materi kunci, Anda tidak dapat mengatur, mengubah, atau membatalkan tanggal kedaluwarsa untuk impor saat ini. Untuk mengubah nilai-nilai ini, Anda harus mengimpor ulang materi kunci yang sama.

Untuk semua kunci KMS dengan `EXTERNAL` asal, bahan kunci pertama yang diimpor ke dalamnya menjadi terkini dan secara permanen terkait dengannya. Kunci enkripsi simetris dengan dukungan `EXTERNAL` asal rotasi sesuai permintaan. Anda dapat mengaitkan beberapa materi utama dengan kunci impor yang mendukung rotasi sesuai permintaan. Proses untuk mengimpor bahan kunci baru berbeda untuk kunci Single-region dan Multi-region seperti yang dijelaskan di bagian Impor materi kunci baru. Anda harus mengatur `importType` parameter `NEW_KEY_MATERIAL` dengan ImportKeyMaterialtindakan untuk mengaitkan materi kunci baru dengan kunci KMS. Nilai default dari `ImportType` parameter opsional adalah`EXISTING_KEY_MATERIAL`. Ketika Anda menghilangkan `ImportType` parameter atau menentukannya sebagai`EXISTING_KEY_MATERIAL`, Anda harus mengimpor bahan kunci yang sebelumnya dikaitkan dengan kunci KMS.

Untuk kunci asimetris, atau HMAC KMS dengan `EXTERNAL` asal, hanya satu bahan kunci yang dapat dikaitkan dengan kunci tersebut. AWS KMS akan menolak permintaan ImportKeyMaterialAPI dengan `ImportType` parameter.

Ketika semua materi kunci yang secara permanen terkait dengan kunci KMS diimpor, kunci KMS tersedia untuk digunakan dalam operasi kriptografi. Jika salah satu dari materi utama ini dihapus atau dibiarkan kedaluwarsa, status kunci KMS berubah menjadi `PendingImport` dan kuncinya tidak dapat digunakan untuk operasi kriptografi.

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 AWS SDKs, AWS Command Line Interfaceatau Alat AWS untuk 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 impor kedaluwarsa, AWS KMS menghapusnya. 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.

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. Namun Anda dapat mengaktifkan dan menonaktifkan kedaluwarsa, atau mengatur waktu kedaluwarsa baru dengan mengimpor ulang materi utama. 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.

Tetapkan deskripsi bahan utama

Kunci enkripsi simetris dengan `EXTERNAL` asal dapat memiliki beberapa bahan utama yang terkait dengannya. Anda dapat menentukan deskripsi materi kunci opsional saat mengimpor materi kunci ke kunci tersebut. Deskripsi dapat digunakan untuk melacak di mana bahan kunci yang sesuai dipertahankan dengan tahan lama di luar AWS KMS.

Untuk kunci Multi-wilayah, Anda dapat mengatur atau mengubah deskripsi materi kunci hanya pada kunci Wilayah utama. AWS KMS secara otomatis menyebarkan deskripsi material kunci ke kunci Region replika.

Mengimpor materi kunci baru

Untuk melakukan rotasi sesuai permintaan pada kunci KMS enkripsi simetris dengan materi kunci yang diimpor, Anda harus terlebih dahulu mengimpor materi kunci baru, yang sebelumnya tidak terkait dengan kunci.

- Kunci Wilayah Tunggal
  - Gunakan ImportKeyMaterialoperasi dengan `ImportType` parameter yang disetel `NEW_KEY_MATERIAL` untuk menyelesaikan tugas ini. Materi kunci ini tidak secara permanen terkait dengan kunci sampai Anda melakukan RotateKeyOnDemandoperasi atau memutar kunci di Konsol Manajemen AWS. Sampai saat itu, bahan utama ini dalam `PENDING_ROTATION` keadaan. Kunci KMS dapat memiliki paling banyak satu materi kunci dalam `PENDING_ROTATION` keadaan kapan saja. Materi kunci dalam `PENDING_ROTATION` keadaan dapat dihapus tanpa memengaruhi kegunaan kunci dalam operasi kriptografi.
- Kunci Multi-Region
  - Untuk mengimpor materi kunci ke kunci Multi-wilayah, Anda harus terlebih dahulu mengimpor materi kunci baru ke kunci Wilayah utama. Anda tidak dapat langsung mengimpor materi kunci baru ke kunci Region replika. Setelah mengimpor materi kunci baru ke kunci Region primer, Anda dapat mengimpor bahan kunci yang sama ke dalam kunci Region replika.
  - Gunakan `ImportKeyMaterial`operasi dengan `ImportType` parameter yang disetel ke **`NEW_KEY_MATERIAL`** kunci Region primer untuk menyelesaikan tugas ini. Untuk kunci Region replika, gunakan **`EXISTING_KEY_MATERIAL`** parameter `ImportType` untuk `ImportKeyMaterial` operasi.
  - Materi kunci untuk enkripsi simetris Kunci multi-wilayah harus diimpor ke semua kunci Region replika dan kunci Region primer sebelum status material utama berubah menjadi status. `PENDING_ROTATION` Sampai saat itu, keadaan materi

kunci baru adalah`PENDING_MULTI_REGION_IMPORT_AND_ROTATION`. Kunci KMS dapat memiliki paling banyak satu materi kunci dalam `PENDING_ROTATION` atau `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` status kapan saja (lihat `KeyMaterialState` deskripsi di RotationsListEntry). Materi kunci dalam `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` atau `PENDING_ROTATION` status tidak secara permanen terkait dengan kunci dan dapat dihapus tanpa memengaruhi kegunaan kunci dalam operasi kriptografi.

Impor ulang bahan kunci

Jika Anda mengelola kunci KMS dengan materi kunci impor, 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.

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.

Prosedur untuk mengimpor ulang materi kunci adalah prosedur yang sama yang Anda gunakan untuk mengimpor materi kunci pertama kali, 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. Untuk kunci enkripsi simetris, Anda juga dapat mengubah deskripsi materi utama.

  Untuk kunci Multi-wilayah, Anda dapat mengatur atau mengubah deskripsi materi kunci hanya pada kunci Wilayah utama. AWS KMS secara otomatis menyebarkan deskripsi material kunci ke kunci Region replika.

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.

Material kunci impor (konsol)

Anda dapat menggunakan bahan kunci Konsol Manajemen AWS untuk mengimpor.

1. Jika Anda berada di halaman Unggah materi kunci yang dibungkus, lewati keStep 10.
2. Masuk ke Konsol Manajemen AWS 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 Origin of External (Import key material). Untuk informasi tentang membuat kunci KMS dengan materi kunci impor, lihatMengimpor bahan utama untuk AWS KMS keys.
7. Pilih tab yang sesuai berdasarkan jenis kunci Anda.

   - Untuk tombol asimetris dan HMAC, pilih tab Bahan kunci.
   - Untuk kunci enkripsi simetris, pilih tab Bahan kunci dan rotasi.
8. Pilih tindakan impor.

   - Untuk kunci asimetris dan HMAC, pilih Impor bahan kunci.
   - Untuk kunci enkripsi simetris, pilih salah satu dari berikut ini:
     - Impor bahan kunci awal (jika belum ada bahan kunci yang diimpor)
     - Impor bahan kunci baru (untuk menambahkan material baru untuk rotasi)
     - Impor ulang materi kunci (tersedia dari menu Tindakan di tabel bahan utama)

> **Note**
>
> Untuk kunci Multi-Region, Anda harus terlebih dahulu mengimpor materi kunci baru ke kunci Region primer. Kemudian, impor bahan kunci yang sama ke setiap kunci Region replika.
> Untuk kunci Multi-wilayah utama, tabel Bahan kunci menyertakan kolom status impor Replika yang menampilkan status impor di semua wilayah replika (misalnya, “0 dari 3

yang diimpor"). Pilih nilai status impor replika untuk membuka modal yang menunjukkan status impor untuk setiap wilayah replika. Modal menyediakan tautan materi kunci Impor untuk wilayah replika di mana materi kunci baru belum diimpor.

9. Jika Anda mengunduh materi kunci, mengimpor token, dan mengenkripsi materi kunci, pilih Berikutnya.

> **Note**
>
> Untuk kunci Multi-Region, Anda harus terlebih dahulu mengimpor materi kunci baru ke kunci Region primer. Kemudian Anda dapat mengimpor bahan kunci yang sama ke dalam kunci Region replika.

10. 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.
11. 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.
12. Untuk kunci enkripsi simetris, Anda dapat secara opsional menentukan deskripsi untuk materi kunci yang diimpor.
13. Pilih Impor bahan kunci.

Impor bahan kunci (AWS KMS API)

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
```

5. Jika Anda ingin mengimpor materi kunci baru, yang sebelumnya tidak terkait dengan kunci KMS, atur `ImportType` parameternya ke`NEW_KEY_MATERIAL`. Opsi ini hanya dapat digunakan dengan kunci enkripsi simetris. Untuk kunci ini, Anda juga dapat menggunakan `KeyMaterialDescription` parameter opsional untuk mengatur deskripsi untuk materi kunci yang diimpor dalam contoh baris perintah berikut:

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_EXPIRES \
    --valid-to 2023-06-17T12:00:00-08:00 \
    --import-type NEW_KEY_MATERIAL \
    --key-material-description "Q2 2025 Rotation"
```

6. Untuk kunci Multi-wilayah, Anda dapat mengatur atau mengubah deskripsi materi kunci hanya pada kunci Wilayah utama. AWS KMS secara otomatis menyebarkan deskripsi material kunci ke kunci Region replika.

> **Tip**
>
> Jika perintah tidak berhasil, Anda mungkin melihat a `KMSInvalidStateException` atau a`NotFoundException`. Anda dapat mencoba kembali permintaan tersebut.

Buat kunci KMS di toko AWS CloudHSM kunci

Setelah Anda membuat toko AWS CloudHSM kunci, Anda dapat membuat AWS KMS keys di 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. KMS tidak dapat berkomunikasi IPv6 dengan toko-toko AWS CloudHSM utama.

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 aktif HSMs di Availability Zone yang berbeda. Untuk menemukan status koneksi dan jumlah HSMs, lihat halaman toko AWS CloudHSM utama di Konsol Manajemen AWS. Saat menggunakan operasi API, gunakan DescribeCustomKeyStoresoperasi untuk memverifikasi bahwa penyimpanan AWS CloudHSM kunci terhubung. Untuk memverifikasi jumlah aktif HSMs 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 cluster sebagai `kmsuser`CU yang Anda buat. Kemudian ia menciptakan kunci simetris Advanced Encryption Standard (AES) 256-bit yang persisten, tidak dapat diekstraksi, di cluster. AWS KMS menetapkan 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 .

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 aktif HSMs 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.

Buat kunci KMS baru di toko kunci CloudHSM Anda

Anda dapat membuat kunci KMS enkripsi simetris di toko AWS CloudHSM kunci Anda di AWS KMS konsol atau dengan menggunakan operasi. CreateKey

Menggunakan AWS KMS 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 Konsol Manajemen AWS 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. AWS CloudHSM Cluster yang terkait harus aktif dan berisi setidaknya dua aktif HSMs di Availability Zone yang berbeda.

    Untuk bantuan menghubungkan toko AWS CloudHSM kunci, lihatPutuskan sambungan toko AWS CloudHSM kunci. Untuk bantuan menambahkan HSMs, lihat Menambahkan HSM di Panduan AWS CloudHSM Pengguna.

11. Pilih Berikutnya.

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 menambahkan tag ke AWS sumber daya, buat AWS laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan berdasarkan tag. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tag di AWS KMS 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.

> **Catatan**
>
> 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.
> AWS KMS Konsol menambahkan administrator kunci ke kebijakan kunci di bawah pengenal `"Allow access for Key Administrators"` pernyataan. Memodifikasi

pengenal pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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.

> **Catatan**
>
> 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.
> AWS KMS Konsol menambahkan pengguna kunci ke kebijakan kunci di bawah pengidentifikasi pernyataan `"Allow use of the key"` dan`"Allow attachment of persistent resources"`. Memodifikasi pengidentifikasi pernyataan ini dapat memengaruhi cara konsol menampilkan pembaruan yang Anda buat pada pernyataan tersebut.

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 Berikutnya.
21. Tinjau pernyataan kebijakan utama untuk kunci tersebut. Untuk membuat perubahan pada kebijakan utama, pilih Edit.

22. Pilih Berikutnya.
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 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. Lihat perinciannya di Sesuaikan tampilan konsol Anda.

Menggunakan AWS KMS 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 aktif. HSMs Jika cluster memiliki kurang dari dua HSMs, `CreateKey` operasi 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`.

Responsnya IDs mencakup penyimpanan kunci khusus dan AWS CloudHSM cluster.

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"
    ]
  }
}
```

Buat kunci KMS di toko kunci eksternal

Setelah Anda membuat dan menghubungkan toko kunci eksternal Anda, Anda dapat membuat AWS KMS keys di 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 KMS menyediakan ID kunci eksternal ke proxy penyimpanan kunci eksternal Anda dalam permintaan untuk operasi enkripsi dan dekripsi. AWS KMS tidak 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 Lihat perinciannya di 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 baru di toko kunci eksternal Anda

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, lihatLihat toko kunci eksternal. Untuk menghubungkan toko kunci eksternal Anda, lihatConnect dan lepaskan 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 KMS juga merekomendasikan agar kunci eksternal digunakan secara eksklusif untuk penyimpanan kunci eksternal. Pembatasan ini membuatnya lebih mudah 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 KMS merekomendasikan agar kunci eksternal disiapkan untuk menangani hingga 1800 permintaan per detik.

Buat kunci KMS baru di toko kunci eksternal Anda

Anda dapat membuat kunci KMS baru di toko kunci eksternal Anda di AWS KMS konsol atau dengan menggunakan CreateKeyoperasi.

Menggunakan AWS KMS 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 KMS Konsol 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 Konsol Manajemen AWS 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 IDs, 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 tag ke AWS sumber daya Anda, AWS buat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan berdasarkan tag. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tag di AWS KMS dan. ABAC untuk AWS KMS
11. Pilih Berikutnya.

12. Di bagian Key Administrators, 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 kredensyal 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 Pertimbangan khusus untuk menghapus kunci.

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 kredensyal 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 Konsol Manajemen AWS 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. Lihat perinciannya di Menggunakan AWS KMS 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 IDs, 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 tag ke AWS sumber daya Anda, AWS buat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan berdasarkan tag. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tag di AWS KMS dan. ABAC untuk AWS KMS

16. Pilih Berikutnya.

17. Di bagian Key Administrators, 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. Hapus sebuah AWS KMS key

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. Lihat perinciannya di Sesuaikan tampilan konsol Anda.

Menggunakan AWS KMS API

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.

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"
    }
  }
}
```

Identifikasi dan lihat kunci

Anda dapat menggunakan Konsol Manajemen AWSatau 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

- Temukan ID kunci dan kunci ARN
- Akses dan daftar rincian kunci KMS
- Identifikasi berbagai jenis kunci
- Sesuaikan tampilan konsol Anda
- Temukan kunci KMS dan bahan kunci di toko AWS CloudHSM kunci

Temukan ID kunci dan kunci ARN

Untuk mengidentifikasi AWS 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.

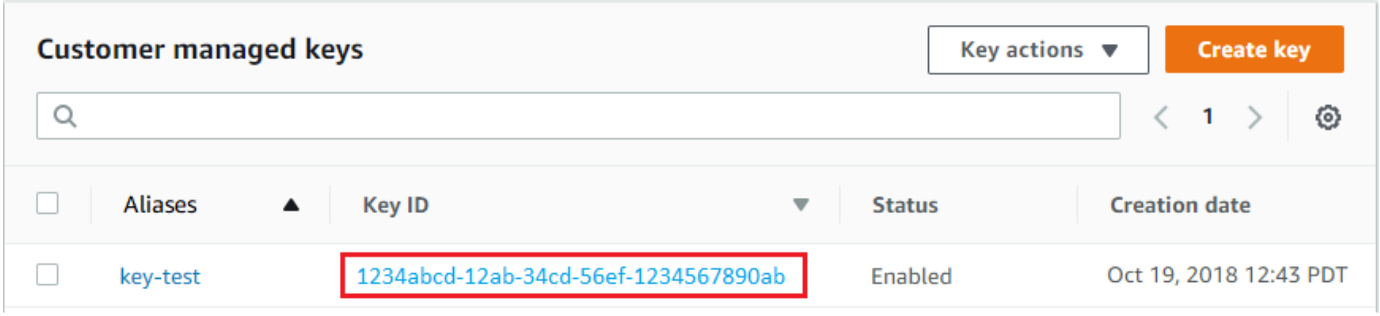
Anda dapat menggunakan AWS KMS konsol atau ListKeysoperasi untuk mengidentifikasi ID kunci dan kunci ARN dari setiap kunci KMS di akun dan Wilayah Anda.

Untuk informasi rinci tentang pengidentifikasi kunci KMS yang didukung oleh AWS KMS, lihat. Pengidentifikasi kunci (KeyId) Untuk bantuan menemukan nama alias dan alias ARN, lihat Temukan nama alias dan alias ARN untuk kunci KMS.

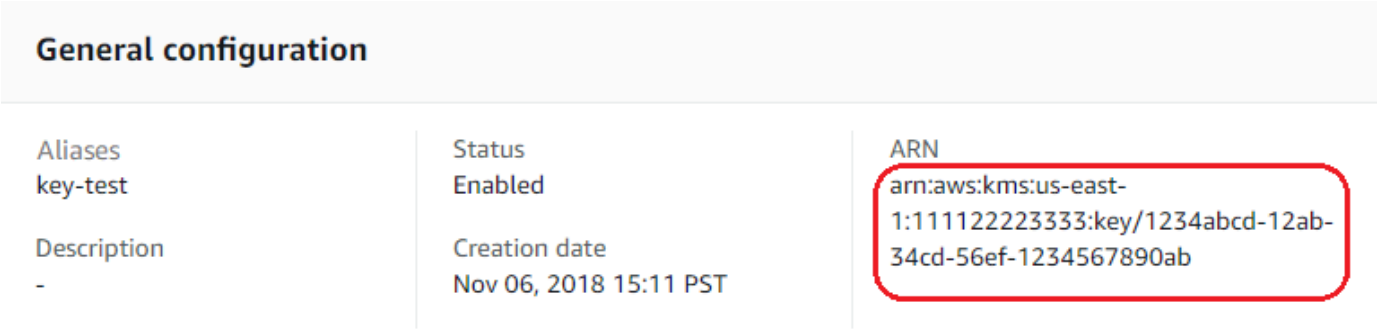
Menggunakan AWS KMS konsol

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. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih CMK. Untuk melihat kunci di akun yang AWS membuat dan mengelola untuk Anda, di panel navigasi, pilih kunci AWS terkelola.
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 “Sesuaikan tampilan konsol 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.



Menggunakan AWS KMS API

Untuk menemukan ID kunci dan kunci ARN dari sebuah AWS KMS key, gunakan operasi. ListKeys

`ListKeys`Operasi mengembalikan ID kunci dan Nama Sumber Daya Amazon (ARN) dari semua kunci KMS di akun dan Wilayah pemanggil.

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 Gunakan `ListKeys` dengan AWS SDK atau CLI.

```
$ aws kms list-keys
{
    "Keys": [
        {
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

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

Akses dan daftar rincian kunci KMS

Anda dapat menggunakan AWS KMS konsol atau DescribeKeyoperasi untuk mengakses dan mencantumkan informasi terperinci tentang kunci KMS di akun dan Wilayah.

Prosedur berikut menunjukkan cara mengakses detail kunci KMS, seperti ID kunci, spesifikasi kunci, penggunaan kunci, dan banyak lagi.

Menggunakan AWS KMS konsol

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 AWS atau halaman kunci yang dikelola Pelanggan, pilih alias atau ID kunci kunci kunci KMS.

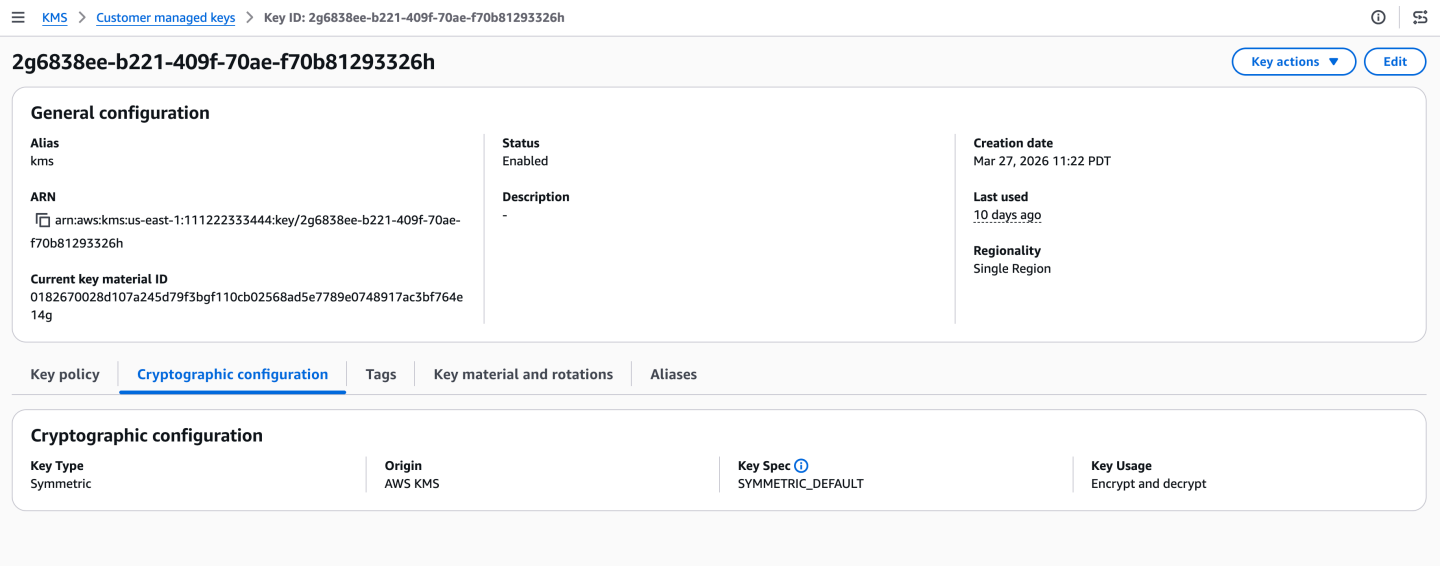
Halaman detail untuk kunci KMS mencakup bagian Konfigurasi Umum yang menampilkan properti dasar kunci KMS. Ini juga mencakup tab di mana Anda dapat melihat dan mengedit properti kunci KMS, seperti kebijakan Kunci, konfigurasi Kriptografi, Tag, Materi kunci dan rotasi (untuk kunci KMS yang mendukung rotasi otomatis atau sesuai permintaan), Regionalitas (untuk kunci Multi-wilayah), dan kunci Publik (untuk kunci KMS asimetris).

> **Note**
>
> AWS KMS Konsol 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.

Untuk menavigasi ke halaman detail utama untuk kunci KMS.

1. Masuk ke Konsol Manajemen AWS 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 CMK. Untuk melihat kunci di akun Anda yang AWS membuat dan mengelola untuk Anda, di panel navigasi, pilih kunci AWS terkelola.
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.



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. Lihat perinciannya di Alias di AWS KMS.

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 CloudHSM Cluster ID, itu akan membuka halaman Clusters di konsol. AWS CloudHSM

Bahan kunci saat ini

Di mana: Bagian konfigurasi umum

Kunci enkripsi simetris dengan `AWS_KMS` asal mendukung rotasi otomatis dan sesuai permintaan. Kunci enkripsi simetris dengan dukungan `EXTERNAL` asal rotasi sesuai permintaan. Kunci-kunci ini dapat memiliki beberapa bahan utama yang terkait dengan kunci. Bahan kunci yang paling baru diputar dapat digunakan untuk enkripsi dan dekripsi. Materi kunci ini diidentifikasi sebagai bahan kunci saat ini. Bahan kunci lainnya hanya dapat digunakan untuk dekripsi. Rotasi kunci otomatis atau sesuai permintaan dari kunci KMS mengubah bahan kuncinya saat ini.

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 toko kunci khusus, itu akan membuka halaman toko kunci kustom di AWS KMS konsol.

Nama penyimpanan kunci kustom

Di mana: Tab konfigurasi kriptografi

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

Jenis toko kunci kustom

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 algoritma enkripsi yang AWS KMS mendukung, lihat Spesifikasi kunci SYMMETRIC_DEFAULT danSpesifikasi 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. Kunci enkripsi simetris dapat memiliki beberapa bahan utama yang terkait dengannya. Untuk kunci tersebut, bidang ini menunjukkan tanggal dan waktu paling awal ketika salah satu materi utama terkait 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 toko 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 toko 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 toko kunci eksternal.

Terakhir digunakan

Di mana: Bagian konfigurasi umum

Terakhir kali kunci KMS digunakan dalam operasi kriptografi, ditampilkan sebagai nilai relatif seperti 5 hari yang lalu. Nilai ini hanya mencakup penggunaan sejak AWS KMS mulai melacak penggunaan untuk kunci ini. Arahkan kursor ke nilai untuk melihat stempel waktu yang tepat, tanggal saat AWS KMS mulai melacak penggunaan, AWS KMS operasi yang dilakukan, dan tautan ke peristiwa terkait. AWS CloudTrail

Bidang ini hanya melacak operasi kriptografi. Non-cryptographic operasi seperti DescribeKeyatau tidak GetKeyPolicydilacak.

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. Lihat perinciannya di the section called “Kebijakan utama”.

Bahan utama dan rotasi

Dimana: Materi kunci dan tab rotasi

Tab ini hanya muncul untuk kunci enkripsi simetris dengan `AWS_KMS` asal (yang mendukung rotasi otomatis dan sesuai permintaan) serta kunci enkripsi simetris wilayah tunggal dengan `EXTERNAL` asal (yang mendukung rotasi sesuai permintaan).

Tab ini memiliki tiga panel:

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

On-demand rotasi: Memulai rotasi sesuai permintaan dari bahan utama dalam kunci yang dikelola pelanggan. Untuk kunci yang diimpor, harus sudah ada materi kunci yang diimpor dalam `PENDING_ROTATION` status agar opsi Putar sekarang tersedia.

Materi utama: Daftar semua materi utama yang terkait dengan kunci KMS. Setiap bahan kunci memiliki pengenal unik dan barisnya menampilkan informasi tambahan tentang materi utama seperti tanggal rotasi ketika bahan kunci tersedia untuk digunakan di KMS. Untuk kunci yang diimpor, setiap baris juga memiliki menu Tindakan yang dapat digunakan untuk menghapus materi kunci tertentu atau mengimpornya kembali ke kunci KMS.

Spesifikasi kunci

Di mana: Tab konfigurasi kriptografi

Jenis bahan kunci dalam kunci KMS. AWS KMS mendukung 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. Lihat perinciannya di Key spec.

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. Lihat perinciannya di Key usage.

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 CloudHSM kunci
- 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 algoritma penandatanganan yang AWS KMS mendukung, lihat Spesifikasi kunci RSA untuk penandatanganan dan verifikasi danSpesifikasi 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. Negara-negara kunci AWS KMS keys

Tag

Di mana: Tab Tanda

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

Saat Anda menambahkan tag ke AWS sumber daya Anda, AWS buat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan berdasarkan tag. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tag di AWS KMS dan. ABAC untuk AWS KMS

Menggunakan AWS KMS API

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 key spesifikasi, status kunci, dan asal material utama. Untuk contoh dalam beberapa bahasa pemrograman, lihat Gunakan `DescribeKey` dengan AWS SDK atau CLI.

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

{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
```

```
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1499988169.234,
        "MultiRegion": false,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "CurrentKeyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    }
}
```

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"
```

```
        ]
    }
}
```

Identifikasi berbagai jenis kunci

Topik berikut menjelaskan cara mengidentifikasi berbagai jenis kunci di AWS KMS konsol dan DescribeKeytanggapan.

Untuk bantuan menavigasi ke tab konfigurasi Kriptografi pada halaman detail untuk kunci KMS, lihat. the section called “Akses dan daftar rincian kunci KMS”

Topik

- Identifikasi kunci KMS asimetris
- Identifikasi kunci HMAC KMS
- Identifikasi kunci KMS Multi-wilayah
- Identifikasi kunci KMS dengan bahan kunci impor
- Identifikasi kunci KMS di AWS CloudHSM toko-toko utama
- Identifikasi kunci KMS di toko kunci eksternal

Identifikasi kunci KMS asimetris

Di AWS KMS konsol

Kolom Key type dari tabel Customer managed keys menunjukkan apakah setiap kunci KMS simetris atau asimetris. Anda dapat memfilter tabel dengan nilai Key type untuk menampilkan hanya tombol KMS asimetris. Untuk mengetahui informasi selengkapnya, lihat the section called “Urutkan dan filter kunci KMS Anda”.

Tab konfigurasi kriptografi pada halaman detail untuk kunci KMS menampilkan Jenis Kunci, yang menunjukkan apakah kunci simetris atau asimetris. Ini juga menampilkan Penggunaan Kunci, yang menunjukkan apakah kunci KMS asimetris Anda digunakan untuk enkripsi dan dekripsi, penandatanganan dan verifikasi, atau memperoleh rahasia bersama.

Dalam DescribeKey tanggapan

Saat Anda memanggil `DescribeKey` operasi pada kunci KMS asimetris, respons mencakup `KeyUsage` nilai `KeySpec` dan, yang dapat digunakan untuk menentukan apakah kunci KMS simetris atau asimetris.

Jika `KeySpec` nilainya`SYMMETRIC_DEFAULT`, kuncinya adalah kunci KMS enkripsi simetris. Untuk detail tentang spesifikasi kunci asimetris, lihat. Referensi spesifikasi kunci

Jika `KeyUsage` nilainya `SIGN_VERIFY` atau`KEY_AGREEMENT`, kuncinya adalah kunci KMS asimetris.

`DescribeKey`Operasi ini juga mengembalikan rincian berikut untuk kunci KMS asimetris.

- Untuk kunci KMS asimetris dengan `KeyUsage` nilai`ENCRYPT_DECRYPT`, operasi mengembalikan`EncryptionAlgorithms`, yang mencantumkan algoritma enkripsi yang valid untuk kunci tersebut.
- Untuk kunci KMS asimetris dengan `KeyUsage` nilai, operasi mengembalikan`SIGN_VERIFY`, yang mencantumkan algoritma penandatanganan yang valid untuk kunci tersebut. `SigningAlgorithms`
- Untuk kunci KMS asimetris dengan `KeyUsage` nilai`KEY_AGREEMENT`, operasi mengembalikan`KeyAgreementAlgorithms`, yang mencantumkan algoritma perjanjian kunci yang valid untuk kunci tersebut.

Untuk informasi lebih lanjut tentang kunci KMS asimetris, lihat. the section called “Kunci asimetris”

Identifikasi kunci HMAC KMS

Di AWS KMS konsol

Kunci KMS HMAC disertakan dalam tabel kunci terkelola Pelanggan, tetapi Anda tidak dapat mengurutkan atau memfilter tabel ini berdasarkan spesifikasi kunci atau nilai penggunaan kunci yang mengidentifikasi kunci HMAC. Untuk mempermudah menemukan kunci HMAC Anda, tetapkan alias atau tag yang berbeda. Kemudian Anda dapat mengurutkan atau memfilter berdasarkan alias atau tag.

Tab konfigurasi kriptografi pada halaman detail untuk kunci KMS menampilkan Jenis Kunci, yang menunjukkan apakah kunci simetris atau asimetris. Kunci HMAC KMS simetris. Tab konfigurasi kriptografi juga menampilkan Key Usage. Untuk kunci HMAC KMS, nilai penggunaan kunci selalu Menghasilkan dan memverifikasi MAC.

Dalam DescribeKey tanggapan

Saat Anda memanggil `DescribeKey` operasi pada kunci HMAC KMS, respons mencakup `KeySpec` dan `KeyUsage` nilai. Untuk kunci HMAC KMS nilai penggunaan kunci selalu `GENERATE_VERIFY_MAC` dan nilai spesifikasi kunci selalu dimulai dengan. `HMAC_`

Untuk informasi selengkapnya tentang kunci HMAC KMS, lihat. the section called “Kunci HMAC”

Identifikasi kunci KMS Multi-wilayah

Di AWS KMS konsol

Tabel kunci terkelola Pelanggan hanya menampilkan kunci KMS di Wilayah yang dipilih. Anda dapat melihat kunci primer dan replika multi-Wilayah di Wilayah yang dipilih. Untuk mengubah AWS Region, gunakan pemilih Region di sudut kanan atas konsol.

Untuk mempermudah mengidentifikasi kunci Multi-region di tabel Customer managed keys, tambahkan kolom Regionality ke tabel Anda. Untuk bantuan, lihat the section called “Sesuaikan tabel kunci KMS Anda”.

Halaman detail untuk kunci KMS Multi-wilayah mencakup 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 Terkait, AWS KMS konsol akan berubah menjadi Wilayah dari kunci yang dipilih dan membuka halaman detail untuk kunci tersebut. Misalnya, jika Anda memilih kunci replika di `sa-east-1` Wilayah dari contoh bagian Kunci Multi-wilayah Terkait di bawah ini, AWS KMS konsol akan berubah ke `sa-east-1` Wilayah untuk menampilkan halaman detail untuk kunci replika tersebut. 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.

Dalam DescribeKey tanggapan

Secara default, operasi AWS KMS API bersifat Regional dan hanya mengembalikan sumber daya di Wilayah saat ini atau yang ditentukan. Namun, saat Anda memanggil `DescribeKey` operasi pada kunci KMS Multi-wilayah, responsnya mencakup semua kunci Multi-wilayah terkait di Wilayah lain dalam elemen AWS tersebut. `MultiRegionConfiguration`

Untuk informasi selengkapnya tentang kunci KMS Multi-wilayah, lihat. the section called “Multi-Region kunci”

Identifikasi kunci KMS dengan bahan kunci impor

Di AWS KMS konsol

Untuk mempermudah mengidentifikasi kunci KMS dengan materi kunci yang diimpor di tabel kunci yang dikelola Pelanggan, tambahkan kolom Origin ke tabel Anda. Kolom Origin memudahkan untuk mengidentifikasi kunci KMS dengan nilai properti asal Eksternal (Import Key material). Untuk bantuan, lihat the section called “Sesuaikan tabel kunci KMS Anda”.

Tab konfigurasi kriptografi pada halaman detail untuk kunci KMS menampilkan Origin, yang mengidentifikasi sumber bahan kunci untuk kunci KMS. Untuk kunci KMS dengan bahan kunci impor, nilai asal selalu Eksternal (bahan Import Key). Halaman detail juga menyertakan tab Materi kunci yang memberikan informasi rinci tentang materi kunci yang diimpor. Kunci enkripsi simetris dengan `EXTERNAL` asal mendukung rotasi sesuai permintaan dan dapat memiliki beberapa materi utama yang terkait dengannya. Untuk kunci seperti itu, tab diberi label Bahan kunci dan rotasi.

Dalam DescribeKey tanggapan

Saat Anda memanggil `DescribeKey` operasi pada kunci KMS dengan materi kunci yang diimpor, respons mencakup`Origin`,`ExpirationModel`, dan `ValidTo` nilai. Untuk kunci KMS dengan bahan kunci impor nilai asal selalu`EXTERNAL`. `ExpirationModel`Nilai menunjukkan apakah bahan kunci diatur untuk kedaluwarsa atau tidak, dan `ValidTo` nilai menunjukkan kapan bahan kunci akan kedaluwarsa. Ketika beberapa bahan kunci dikaitkan dengan kunci, `ValidTo` nilainya menunjukkan waktu kedaluwarsa paling awal di semua bahan utama (kecuali untuk satu rotasi yang tertunda) dan `ExpirationModel` diatur `DOES_NOT_EXPIRE` hanya jika tidak ada bahan utama ini yang ditetapkan untuk kedaluwarsa. Untuk informasi selengkapnya, lihat Mengatur waktu kedaluwarsa (opsional).

Untuk informasi lebih lanjut tentang kunci KMS dengan materi kunci impor, lihatthe section called “Materi kunci yang diimpor”.

Identifikasi kunci KMS di AWS CloudHSM toko-toko utama

Di AWS KMS konsol

Untuk mempermudah mengidentifikasi kunci KMS di toko AWS CloudHSM utama di tabel Customer managed keys, tambahkan kolom Origin ke tabel Anda. Kolom Origin memudahkan

untuk mengidentifikasi kunci KMS dengan nilai properti AWS CloudHSMasal. Untuk bantuan, lihat the section called “Sesuaikan tabel kunci KMS Anda”.

Tab konfigurasi kriptografi pada halaman detail untuk kunci KMS menampilkan Origin, yang mengidentifikasi sumber bahan kunci untuk kunci KMS. Untuk kunci KMS di toko-toko AWS CloudHSM utama, nilai asal selalu AWS CloudHSM.

Untuk kunci KMS di toko AWS CloudHSM kunci, tab konfigurasi Kriptografi mencakup bagian tambahan, toko kunci khusus, yang menyediakan informasi tentang penyimpanan AWS CloudHSM kunci dan AWS CloudHSM cluster yang terkait dengan kunci KMS.

Dalam DescribeKey tanggapan

Ketika Anda memanggil `DescribeKey` operasi pada kunci KMS di penyimpanan AWS CloudHSM kunci, respons termasuk`Origin`, yang mengidentifikasi sumber materi kunci. Untuk kunci KMS dalam penyimpanan AWS CloudHSM kunci, nilai asal selalu`AWS_CLOUDHSM`. Operasi ini juga mengembalikan bidang khusus berikut untuk kunci KMS di toko-toko AWS CloudHSM utama:

- `CloudHsmClusterId`
- `CustomKeyStoreId`

Untuk informasi lebih lanjut tentang toko-toko AWS CloudHSM utama, lihatthe section called “AWS CloudHSM toko-toko utama”.

Identifikasi kunci KMS di toko kunci eksternal

Di AWS KMS konsol

Untuk mempermudah mengidentifikasi kunci KMS di toko kunci eksternal di tabel Customer managed keys, tambahkan kolom Origin ke tabel Anda. Kolom Origin memudahkan untuk mengidentifikasi kunci KMS dengan nilai properti asal penyimpanan kunci Eksternal. Untuk bantuan, lihat the section called “Sesuaikan tabel kunci KMS Anda”.

Tab konfigurasi kriptografi pada halaman detail untuk kunci KMS menampilkan Origin, yang mengidentifikasi sumber bahan kunci untuk kunci KMS. Untuk kunci KMS di toko kunci eksternal, nilai asal selalu Penyimpanan kunci eksternal.

Untuk kunci KMS di toko kunci eksternal, tab konfigurasi Kriptografi mencakup dua bagian tambahan, penyimpanan kunci khusus dan kunci Eksternal. Tabel penyimpanan kunci Kustom

memberikan informasi tentang penyimpanan kunci eksternal yang terkait dengan kunci KMS. Tabel 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.

Dalam DescribeKey tanggapan

Ketika Anda memanggil `DescribeKey` operasi pada kunci KMS di penyimpanan kunci eksternal, respons termasuk`Origin`, yang mengidentifikasi sumber bahan kunci. Untuk kunci KMS dalam penyimpanan AWS CloudHSM kunci, nilai asal selalu`EXTERNAL_KEY_STORE`. Operasi ini juga mengembalikan `CustomKeyStoreId` elemen, yang mengidentifikasi penyimpanan kunci eksternal yang terkait dengan kunci KMS.

Untuk informasi selengkapnya tentang toko kunci eksternal, lihatthe section called “Toko kunci eksternal”.

Sesuaikan tampilan konsol Anda

Anda dapat menyesuaikan tampilan AWS KMS konsol untuk membuatnya lebih mudah untuk menemukan kunci KMS Anda. Sesuaikan tabel yang muncul di Kunci yang dikelola AWShalaman kunci yang dikelola Pelanggan untuk menampilkan informasi yang paling Anda butuhkan, atau urutkan dan filter kunci KMS yang dikembalikan dalam tabel.

Topik

- Urutkan dan filter kunci KMS Anda
- Sesuaikan tabel kunci KMS Anda

Urutkan dan filter kunci KMS Anda

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

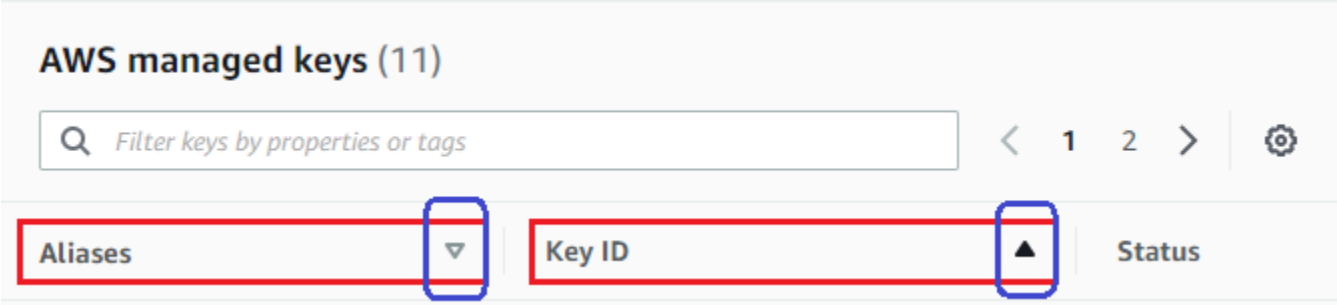
Urutkan

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.

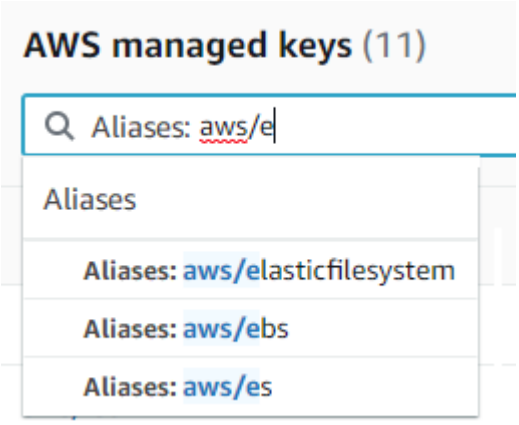
Properti yang dapat disaring tercantum dalam kotak filter. Pada Kunci yang dikelola AWS halaman, Anda dapat memfilter berdasarkan alias dan ID kunci. Pada halaman kunci terkelola

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.



Filter tabel kunci KMS yang disarankan

Filter untuk tombol KMS asimetris

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.

Filter untuk tombol Multi-wilayah

Untuk hanya menampilkan kunci Multi-region, pada halaman Customer managed keys, pilih kotak filter, pilih Regionality, lalu pilih Regionality:. Multi-Region Multi-RegionOpsi hanya muncul ketika Anda memiliki kunci Multi-wilayah dalam tabel.

Filter untuk tag

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.

Filter berdasarkan teks

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.

Sesuaikan tabel kunci KMS Anda

Anda dapat menyesuaikan tabel yang muncul di halaman kunci yang dikelola Pelanggan Kunci yang dikelola AWSdan Konsol Manajemen AWS 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 Menggunakan AWS KMS konsol.

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 oleh AWS 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.

Multi-Region kunci

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.

Temukan kunci KMS dan bahan kunci di toko AWS CloudHSM kunci

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 kunci, Anda masih mempertahankan kendali dan tanggung jawab untuk pengelolaan AWS CloudHSM klaster Anda, serta pencadangan HSMs dan kunci di. HSMs Anda mungkin perlu mengidentifikasi kunci untuk mengaudit materi kunci, melindunginya dari penghapusan yang tidak disengaja, atau menghapusnya dari HSMs dan cluster backup setelah menghapus kunci KMS.

Semua materi kunci untuk kunci KMS di toko AWS CloudHSM kunci Anda dimiliki oleh pengguna `kmsuser` kripto (CU). AWS KMS menetapkan atribut label kunci, yang hanya dapat dilihat di AWS 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 untuk melacak operasi kunci KMS di AWS CloudTrail log, memprediksi efek pemutusan penyimpanan kunci kustom pada kunci KMS, 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 berikut, ganti nilai ID penyimpanan kunci kustom fiktif dengan nilai yang 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 perintah daftar kunci di CloudHSM CLI.

Anda juga dapat menggunakan perintah daftar kunci untuk menemukan AWS CloudHSM kunci AWS KMS untuk. 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. Perintah daftar kunci mengembalikan `key-reference` dan`label`.

> **Catatan**
>
> Prosedur berikut menggunakan alat baris perintah AWS CloudHSM Client SDK 5, CloudHSM CLI. CloudHSM CLI `key-handle` menggantikan dengan. `key-reference`

Pada tanggal 1 Januari 2025, AWS CloudHSM akan mengakhiri dukungan untuk alat baris perintah Client SDK 3, CloudHSM Management Utility (CMU) dan Key Management Utility (KMU). Untuk informasi selengkapnya tentang perbedaan antara alat baris perintah Client SDK 3 dan alat baris perintah Client SDK 5, lihat Memigrasi dari Client SDK 3 CMU dan KMU ke Client SDK 5 CloudHSM CLI di Panduan Pengguna.AWS CloudHSM

Untuk menjalankan prosedur ini, Anda harus memutuskan sambungan penyimpanan AWS CloudHSM kunci sementara sehingga Anda dapat masuk sebagai `kmsuser` CU.

1. Putuskan sambungan toko AWS CloudHSM kunci, jika belum terputus, lalu masuk seperti `kmsuser` yang dijelaskan. Cara memutuskan koneksi dan login

   > **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.

2. Gunakan perintah daftar kunci di CloudHSM CLI untuk menemukan semua kunci untuk pengguna saat ini yang ada di cluster Anda. AWS CloudHSM

   Secara default, hanya 10 kunci dari pengguna yang saat ini masuk yang ditampilkan, dan hanya `key-reference` dan `label` ditampilkan sebagai output. Untuk opsi lainnya, lihat daftar kunci di Panduan AWS CloudHSM Pengguna.

   ```
   aws-cloudhsm > key list
   {
     "error_code": 0,
     "data": {
       "matched_keys": [
         {
           "key-reference": "0x0000000000000123",
           "attributes": {
             "label": "arn:aws:kms:us-
   west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
           }
         },
         {
   ```

```
        "key-reference": "0x0000000000000456",
        "attributes": {
          "label": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
      },.
      ...8 keys later...
    ],
    "total_key_count": 56,
    "returned_key_count": 10,
    "next_token": "10"
  }
}
```

3. Keluar dan sambungkan kembali toko AWS CloudHSM kunci seperti yang dijelaskan diCara logout dan menghubungkan kembali.

Temukan kunci KMS untuk kunci AWS CloudHSM

Jika Anda mengetahui referensi kunci atau ID kunci yang `kmsuser` dimiliki di cluster, Anda dapat menggunakan nilai tersebut untuk mengidentifikasi kunci KMS terkait di penyimpanan 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 daftar kunci di CloudHSM CLI untuk mengidentifikasi kunci KMS yang terkait dengan kunci. AWS CloudHSM

> **Catatan**
>
> Prosedur berikut menggunakan alat baris perintah AWS CloudHSM Client SDK 5, CloudHSM CLI. CloudHSM CLI `key-handle` menggantikan dengan. `key-reference`
> Pada tanggal 1 Januari 2025, AWS CloudHSM akan mengakhiri dukungan untuk alat baris perintah Client SDK 3, CloudHSM Management Utility (CMU) dan Key Management Utility (KMU). Untuk informasi selengkapnya tentang perbedaan antara alat baris perintah Client SDK 3 dan alat baris perintah Client SDK 5, lihat Memigrasi dari Client SDK 3 CMU dan KMU ke Client SDK 5 CloudHSM CLI di Panduan Pengguna.AWS CloudHSM

Untuk menjalankan prosedur ini, Anda perlu 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.

Topik

- Identifikasi kunci KMS yang terkait dengan referensi kunci
- Identifikasi kunci KMS yang terkait dengan ID kunci dukungan

Identifikasi kunci KMS yang terkait dengan referensi kunci

Prosedur berikut menunjukkan cara menggunakan perintah daftar kunci di CloudHSM CLI `key-reference` dengan filter atribut untuk menemukan kunci di cluster Anda yang berfungsi sebagai bahan kunci untuk kunci KMS tertentu di toko kunci Anda. AWS CloudHSM

1. Putuskan sambungan toko AWS CloudHSM kunci, jika belum terputus, lalu masuk seperti `kmsuser` yang dijelaskan. Cara memutuskan koneksi dan login
2. Gunakan perintah daftar kunci di CloudHSM CLI untuk memfilter berdasarkan atribut. `key-reference` Tentukan `verbose` argumen untuk menyertakan semua atribut dan informasi kunci untuk kunci yang cocok. Jika Anda tidak menentukan `verbose` argumen, operasi daftar kunci hanya mengembalikan atribut kunci-referensi dan label kunci yang cocok.

   Sebelum menjalankan perintah ini, ganti contoh `key-reference` dengan yang valid dari akun Anda.

```
aws-cloudhsm > key list --filter attr.key-reference="0x0000000000120034" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000120034",
        "key-info": {
          "key-owners": [
```

```
            {
              "username": "kmsuser",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "id": "0xbacking-key-id",
          "check-value": "0x29bbd1",
          "class": "my_test_key",
          "encrypt": true,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": false,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": false,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": true,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

3. Keluar dan sambungkan kembali toko AWS CloudHSM kunci seperti yang dijelaskan diCara logout dan menghubungkan kembali.

Identifikasi kunci KMS yang terkait dengan ID kunci dukungan

Semua entri CloudTrail log untuk operasi kriptografi dengan kunci KMS di toko AWS CloudHSM kunci menyertakan `additionalEventData` bidang dengan dan. `customKeyStoreId` `backingKeyId` Nilai yang dikembalikan di `backingKeyId` bidang berkorelasi dengan atribut kunci CloudHSM. `id` Anda dapat memfilter operasi daftar kunci dengan `id` atribut untuk mengidentifikasi kunci KMS yang terkait dengan spesifik`backingKeyId`.

1. Putuskan sambungan toko AWS CloudHSM kunci, jika belum terputus, lalu masuk seperti `kmsuser` yang dijelaskan. Cara memutuskan koneksi dan login
2. Gunakan perintah daftar kunci di CloudHSM CLI dengan filter atribut untuk menemukan kunci di cluster Anda yang berfungsi sebagai bahan kunci untuk kunci KMS tertentu di toko kunci Anda. AWS CloudHSM

   Contoh berikut menunjukkan bagaimana untuk memfilter oleh `id` atribut. AWS CloudHSM mengenali `id` nilai sebagai nilai heksadesimal. Untuk memfilter operasi daftar kunci berdasarkan `id` atribut, Anda harus terlebih dahulu mengonversi `backingKeyId` nilai yang Anda identifikasi dalam entri CloudTrail log Anda ke dalam format yang AWS CloudHSM mengenali.

   a. Gunakan perintah Linux berikut untuk mengubah `backingKeyId` menjadi representasi heksadesimal.

   ```
   echo backingKeyId | tr -d '\n' |  xxd -p
   ```

   Contoh berikut menunjukkan bagaimana mengkonversi array `backingKeyId` byte menjadi representasi heksadesimal.

   ```
   echo 5890723622dc15f699aa9ab2387a9f744b2b884c18b2186ee8ada4f556a2eb9d | tr -d
    '\n' |  xxd -p
   353839303732333632326463313566363939616139616232333837613966373434623262383834633138623
   ```

   b. Tambahkan representasi heksadesimal dari with. `backingKeyId` `0x`

   ```
   0x353839303732333632326463313566363939616139616232333837613966373434623262383834633138
   ```

   c. Gunakan `backingKeyId` nilai yang dikonversi untuk memfilter berdasarkan `id` atribut. Tentukan `verbose` argumen untuk menyertakan semua atribut dan informasi kunci untuk kunci yang cocok. Jika Anda tidak menentukan `verbose` argumen, operasi daftar kunci hanya mengembalikan atribut kunci-referensi dan label kunci yang cocok.

```
aws-cloudhsm > key list --filter
 attr.id="0x353839303732333632326463313566363939616139616232333837613966373434623262383
 --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000120034",
        "key-info": {
          "key-owners": [
            {
              "username": "kmsuser",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "id":
 "0x353839303732333632326463313566363939616139616232333837613966373434623262383834633133
          "check-value": "0x29bbd1",
          "class": "my_test_key",
          "encrypt": true,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": false,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": false,
          "trusted": false,
          "unwrap": true,
          "verify": false,
```

```
          "wrap": true,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

3. Keluar dan sambungkan kembali toko AWS CloudHSM kunci seperti yang dijelaskan diCara logout dan menghubungkan kembali.

Temukan AWS CloudHSM kunci untuk kunci KMS

Anda dapat menggunakan ID kunci KMS dari kunci KMS di penyimpanan AWS CloudHSM kunci untuk mengidentifikasi kunci di AWS CloudHSM cluster Anda yang berfungsi sebagai materi utamanya.

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 daftar kunci di CloudHSM CLI untuk menemukan sumber daya kunci dan id dari bahan kunci untuk kunci KMS.

Semua entri CloudTrail log untuk operasi kriptografi dengan kunci KMS di toko AWS CloudHSM kunci menyertakan `additionalEventData` bidang dengan dan. `customKeyStoreId` `backingKeyId` Nilai yang dikembalikan di `backingKeyId` bidang adalah atribut `id` AWS CloudHSM kunci. Anda dapat memfilter operasi AWS CloudHSM CLI daftar kunci dengan ARN kunci KMS untuk mengidentifikasi atribut kunci CloudHSM yang terkait dengan kunci KMS tertentu. `id`

Untuk menjalankan prosedur ini, Anda harus memutuskan sambungan penyimpanan AWS CloudHSM kunci sementara sehingga Anda dapat masuk sebagai `kmsuser` CU.

> **Catatan**
>
> Prosedur berikut menggunakan alat baris perintah AWS CloudHSM Client SDK 5, CloudHSM CLI. CloudHSM CLI `key-handle` menggantikan dengan. `key-reference`
> Pada tanggal 1 Januari 2025, AWS CloudHSM akan mengakhiri dukungan untuk alat baris perintah Client SDK 3, CloudHSM Management Utility (CMU) dan Key Management Utility

(KMU). Untuk informasi selengkapnya tentang perbedaan antara alat baris perintah Client SDK 3 dan alat baris perintah Client SDK 5, lihat Memigrasi dari Client SDK 3 CMU dan KMU ke Client SDK 5 CloudHSM CLI di Panduan Pengguna.AWS CloudHSM

1. Putuskan sambungan toko AWS CloudHSM kunci, jika belum terputus, lalu masuk seperti `kmsuser` yang dijelaskan. Cara memutuskan koneksi dan login

   > **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.

2. Gunakan perintah daftar kunci di CloudHSM CLI dan `label` filter untuk menemukan kunci KMS untuk kunci tertentu di cluster Anda. AWS CloudHSM Tentukan `verbose` argumen untuk menyertakan semua atribut dan informasi kunci untuk kunci yang cocok. Jika Anda tidak menentukan `verbose` argumen, operasi daftar kunci hanya mengembalikan atribut kunci-referensi dan label kunci yang cocok.

   Contoh berikut menunjukkan bagaimana memfilter dengan `label` atribut yang menyimpan ARN kunci KMS. Sebelum menjalankan perintah ini, ganti contoh kunci KMS ARN dengan yang valid dari akun Anda.

```
aws-cloudhsm > key list --filter attr.label="arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000120034",
        "key-info": {
          "key-owners": [
            {
              "username": "kmsuser",
              "key-coverage": "full"
            }
          ],
```

```
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "id": "0xbacking-key-id",
          "check-value": "0x29bbd1",
          "class": "my_test_key",
          "encrypt": true,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": false,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": false,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": true,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

3. Keluar dan sambungkan kembali toko AWS CloudHSM kunci seperti yang dijelaskan di[Cara logout dan menghubungkan kembali](#).

Aktifkan dan nonaktifkan 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 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.

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

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

Menggunakan AWS KMS konsol

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

1. Masuk ke Konsol Manajemen AWS 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.

Menggunakan AWS KMS API

EnableKeyOperasi memungkinkan dinonaktifkan AWS 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"
        ]
    }
}
```

Putar AWS KMS keys

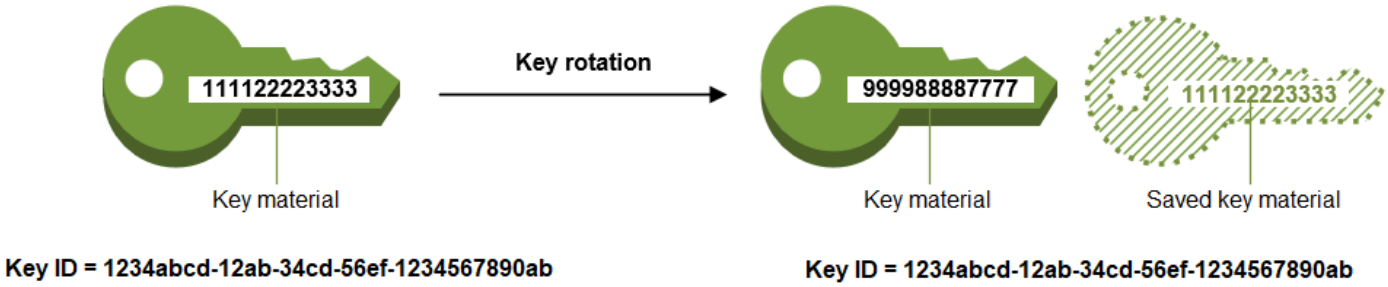
Untuk membuat materi kriptografi baru untuk kunci yang dikelola pelanggan Anda, Anda dapat membuat kunci KMS baru, dan kemudian mengubah aplikasi atau alias Anda untuk menggunakan kunci KMS baru. Atau, Anda dapat memutar materi kunci yang terkait dengan kunci KMS yang ada dengan mengaktifkan rotasi tombol 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. On-demand rotasi tidak mengubah jadwal rotasi otomatis yang ada.

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.

Rotasi kunci hanya mengubah materi kunci saat ini, yang merupakan rahasia kriptografi yang digunakan dalam operasi enkripsi. Saat Anda menggunakan kunci KMS yang diputar untuk mendekripsi ciphertext, AWS KMS gunakan bahan kunci yang digunakan untuk mengenkripsi itu. Anda tidak dapat memilih bahan kunci tertentu untuk operasi dekripsi, AWS KMS secara otomatis memilih bahan kunci 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

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 atau sesuai permintaan, termasuk kunci KMS asimetris, kunci KMS HMAC, dan kunciKMS di toko kunci khusus.

> **Note**
>
> Rotasi kunci 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. Rotasi kunci tidak akan mengurangi efek kunci data yang dikompromikan.

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 harus merotasi kunci KMS?
- Cara kerja rotasi kunci
- Aktifkan rotasi tombol otomatis
- Nonaktifkan rotasi tombol otomatis
- Lakukan rotasi kunci sesuai permintaan
- Daftar rotasi dan bahan utama
- Putar tombol secara manual
- Ubah kunci utama dalam satu set kunci Multi-wilayah

Mengapa harus merotasi kunci 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 dieksploitasi oleh aktor pintar untuk menemukan bit dalam kunci. Yang terbaik adalah menggunakan kunci data sekali, atau hanya beberapa kali, untuk mengurangi kelelahan kunci ini.

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, gunakan rotasi sesuai permintaan jika rotasi otomatis tidak didukung, dan rotasi tombol manual ketika rotasi kunci otomatis maupun sesuai permintaan tidak didukung.

Anda dapat mempertimbangkan untuk melakukan rotasi sesuai permintaan guna menunjukkan kemampuan rotasi materi kunci atau 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

AWS KMS rotasi kunci 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 kunci otomatis hanya didukung pada kunci KMS enkripsi simetris dengan bahan kunci yang AWS KMS menghasilkan (`AWS_KMS`asal).

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 Layanan AWS yang memiliki kunci.

On-demand rotasi

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

On-demand rotasi kunci didukung pada kunci KMS enkripsi simetris dengan bahan kunci yang AWS KMS menghasilkan (`AWS_KMS`asal) dan kunci KMS enkripsi simetris dengan bahan kunci impor (asal). `EXTERNAL`

Rotasi manual

Baik rotasi kunci otomatis maupun sesuai permintaan tidak didukung untuk jenis kunci KMS berikut, tetapi Anda dapat memutar tombol KMS ini secara manual.

- Kunci KMS Asimetris
- Kunci HMAC KMS

- Kunci KMS di toko kunci khusus

Mengelola materi utama

AWS KMS mempertahankan semua materi kunci untuk kunci KMS dengan `AWS_KMS` asal, bahkan jika rotasi kunci dinonaktifkan. AWS KMS menghapus materi kunci hanya ketika Anda menghapus kunci KMS.

Anda mengelola materi utama untuk kunci enkripsi simetris dengan `EXTERNAL` asal. Anda dapat menghapus materi kunci apa pun menggunakan DeleteImportedKeyMaterialoperasi atau mengatur waktu kedaluwarsa saat mengimpor materi. Kunci KMS menjadi tidak dapat digunakan segera setelah salah satu materinya kedaluwarsa atau dihapus.

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

Tanggal rotasi mencerminkan tanggal ketika bahan kunci saat ini untuk kunci KMS diperbarui baik sebagai hasil dari rotasi otomatis (terjadwal) atau rotasi kunci sesuai permintaan.

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).

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.

Memutar tombol Multi-wilayah

Perilaku rotasi berbeda tergantung pada apakah bahan kunci dihasilkan oleh AWS KMS (`AWS_KMS`asal) atau diimpor (`EXTERNAL`asal).

Multi-Region kunci dengan **`AWS_KMS`** asal

Anda dapat mengaktifkan dan menonaktifkan rotasi otomatis dan melakukan rotasi sesuai permintaan dari materi kunci dalam enkripsi simetris Kunci multi-wilayah dengan asal. `AWS_KMS` 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.

Multi-Region kunci dengan **`EXTERNAL`** asal

Anda dapat melakukan rotasi sesuai permintaan dari materi kunci dalam enkripsi simetris Kunci multi-wilayah dengan asal. `EXTERNAL` Rotasi kunci adalah properti bersama dari kunci Multi-wilayah.

Anda memulai rotasi sesuai permintaan hanya pada kunci utama setelah mengimpor materi kunci baru ke kunci utama dan setiap kunci replika.

- Saat Anda mengimpor materi kunci baru ke kunci utama, AWS KMS salin pengenal materi kunci dan deskripsi materi kunci dari kunci utama ke semua kunci replika terkait. Itu tidak menyalin materi kunci.
- Anda harus mengimpor bahan kunci yang sama ke setiap kunci replika satu per satu. Setelah materi kunci diimpor ke semua kunci Multi-wilayah terkait, Anda dapat memulai rotasi sesuai permintaan pada kunci utama. Ini memastikan bahwa AWS KMS tidak mengenkripsi data apa pun dengan materi kunci baru sampai materi kunci tersebut tersedia di kunci utama dan setiap kunci replika.
- Setiap materi kunci dalam kunci utama atau kunci replika apa pun dapat kedaluwarsa atau dihapus secara independen dari materi kunci lainnya dalam kunci yang sama atau kunci multi-wilayah terkait lainnya.

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 AWS layanan.

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.

Aktifkan rotasi tombol 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.

Anda dapat mengaktifkan rotasi tombol otomatis di AWS KMS konsol atau dengan menggunakan EnableKeyRotationoperasi. Untuk mengaktifkan rotasi tombol otomatis, Anda memerlukan `kms:EnableKeyRotation` izin. Untuk informasi selengkapnya tentang AWS KMS izin, lihat. Referensi izin

Menggunakan AWS KMS konsol

1. Masuk ke Konsol Manajemen AWS 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 yang diimpor, 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 Negara-negara kunci AWS KMS keys.
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.

Menggunakan AWS KMS API

Anda dapat menggunakan AWS Key Management Service (AWS KMS) API untuk mengaktifkan 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. 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.

```
$ 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"
}
```

Nonaktifkan rotasi tombol otomatis

Setelah mengaktifkan rotasi kunci otomatis pada kunci yang dikelola pelanggan, Anda dapat memilih untuk menonaktifkannya kapan saja.

Jika Anda menonaktifkan rotasi tombol otomatis, 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.

Menonaktifkan rotasi otomatis tidak memengaruhi kemampuan Anda untuk melakukan rotasi sesuai permintaan, juga tidak membatalkan rotasi sesuai permintaan yang sedang berlangsung.

Anda dapat menonaktifkan rotasi tombol otomatis di AWS KMS konsol atau dengan menggunakan DisableKeyRotationoperasi. Untuk menonaktifkan rotasi tombol otomatis, Anda memerlukan `kms:DisableKeyRotation` izin. Untuk informasi selengkapnya tentang AWS KMS izin, lihat. Referensi izin

Menggunakan AWS KMS konsol

1. Masuk ke Konsol Manajemen AWS 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 yang diimpor, 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 Nonaktifkan.

   > **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 detail selengkapnya, lihat Cara kerja rotasi kunci dan Negara-negara kunci AWS KMS keys.
8. Pilih Simpan.

Menggunakan AWS KMS API

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

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

Contoh berikut menonaktifkan rotasi kunci otomatis pada kunci KMS enkripsi simetris yang ditentukan dan menggunakan GetKeyRotationStatusoperasi untuk melihat hasilnya.

```
$ 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
}
```

Lakukan 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. On-demand rotasi 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 maksimum 25 kali per tombol KMS. Anda dapat menggunakan AWS KMS konsol untuk melihat jumlah rotasi sesuai permintaan yang tersisa yang tersedia untuk kunci KMS.

On-demand rotasi kunci hanya didukung pada kunci KMS enkripsi simetris. Anda tidak dapat melakukan rotasi sesuai permintaan kunci KMS asimetris, kunci KMS HMAC, 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 resmi dengan `kms:RotateKeyOnDemand` dan `kms:GetKeyRotationStatus` izin dapat menggunakan AWS KMS konsol dan AWS KMS API untuk memulai rotasi kunci sesuai permintaan dan melihat status rotasi kunci. Gunakan ListKeyRotationsuntuk melihat rotasi selesai untuk kunci KMS.

Topik

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

Memulai rotasi kunci sesuai permintaan (konsol)

1. Masuk ke Konsol Manajemen AWS 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 Bahan kunci dan rotasi.

   Tab Materi kunci dan rotasi hanya muncul di halaman detail kunci KMS enkripsi simetris yang mendukung rotasi otomatis atau sesuai permintaan. Ini termasuk kunci KMS dengan bahan kunci yang AWS KMS dihasilkan (asal AWS_KMS) dan kunci KMS dengan bahan kunci yang diimpor (asal EKSTERNAL)

   Anda tidak dapat melakukan rotasi sesuai permintaan kunci KMS asimetris, kunci KMS HMAC, atau kunci KMS di toko kunci khusus. Namun, Anda dapat memutarnya secara manual.
6. Pilih Rotate sekarang. Untuk kunci enkripsi simetris dengan bahan kunci yang diimpor, opsi Putar sekarang hanya tersedia jika Anda sebelumnya telah mengimpor materi kunci baru dan dalam status rotasi Tertunda.

> **Note**
>
> Untuk kunci Multi-region, hanya kunci Region primer yang dapat diputar.

7. Baca dan pertimbangkan peringatan dan informasi tentang jumlah rotasi sesuai permintaan yang tersisa untuk kunci tersebut. Anda juga akan melihat informasi seperti ID, deskripsi, dan waktu kedaluwarsa materi kunci yang akan menjadi terkini setelah rotasi. Jika Anda memutuskan bahwa Anda tidak ingin melanjutkan rotasi sesuai permintaan, pilih Batalkan.
8. Pilih tombol Putar untuk mengonfirmasi rotasi sesuai permintaan.

> **Note**
>
> On-demand rotasi 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 (AWS KMS API)

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 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. Dalam contoh ini, kunci KMS juga memiliki rotasi otomatis yang diaktifkan dengan jangka waktu 365 hari.

```
$ 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
}
```

Jika tombol KMS tidak mendukung rotasi otomatis atau tidak mengaktifkan rotasi otomatis, `kms:GetKeyRotationStatus` respons akan memiliki lebih sedikit bidang seperti yang ditunjukkan pada contoh berikut:

```
$ aws kms rotate-key-on-demand --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyRotationEnabled": false,
    "OnDemandRotationStartDate": "2024-02-24T18:44:48.587000+00:00"
}
```

Daftar rotasi dan bahan utama

Kunci KMS yang mendukung rotasi kunci otomatis atau sesuai permintaan dapat memiliki beberapa bahan utama yang terkait dengannya. Kunci-kunci ini memiliki bahan kunci awal dan satu bahan kunci tambahan untuk setiap rotasi otomatis atau sesuai permintaan.

Pengguna yang berwenang dengan `kms:ListKeyRotations` izin dapat menggunakan AWS KMS konsol dan ListKeyRotationsAPI untuk mencantumkan semua materi utama yang terkait dengan kunci KMS, termasuk yang berasal dari rotasi otomatis dan sesuai permintaan yang telah selesai.

Topik

- Daftar rotasi dan bahan utama (konsol)

- Daftar rotasi dan bahan utama (AWS KMS API)

Daftar rotasi dan bahan utama (konsol)

1. Masuk ke Konsol Manajemen AWS 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.
5. Pilih tab Bahan kunci dan rotasi.

   - Tab Materi kunci dan rotasi hanya muncul di halaman detail kunci KMS enkripsi simetris yang mendukung rotasi otomatis atau sesuai permintaan. Ini termasuk kunci KMS dengan bahan utama yang AWS KMS dihasilkan (`AWS_KMS`asal) dan kunci KMS dengan bahan kunci impor (`EXTERNAL`asal).
   - Tabel bahan utama di tab Materi kunci dan rotasi mencantumkan semua bahan utama yang terkait dengan kunci KMS. Untuk setiap materi kunci, entri yang sesuai menampilkan pengenal unik yang ditetapkan oleh AWS KMS, tanggal rotasi, dan status material kunci. Tanggal rotasi mengidentifikasi kapan bahan kunci menjadi arus setelah rotasi kunci otomatis atau sesuai permintaan. Tidak ada tanggal rotasi yang terkait dengan materi pertama atau `Pending rotation` utama. Keadaan material utama menentukan bagaimana AWS KMS menggunakan bahan kunci. Materi kunci saat ini digunakan untuk enkripsi dan dekripsi. Non-currentbahan kunci hanya digunakan untuk dekripsi. Keadaan material utama `Pending rotation` menunjukkan material utama dipentaskan untuk rotasi. Bahan kunci ini tidak digunakan untuk operasi kriptografi apa pun sampai rotasi kunci sesuai permintaan menjadikannya bahan kunci saat ini. Informasi tambahan yang ditampilkan untuk materi kunci tergantung pada jenis kunci KMS.
   - Untuk kunci KMS enkripsi simetris dengan `AWS_KMS` asal, setiap baris juga menampilkan jenis rotasi — `On-demand` atau. `Automatic`
   - Kunci KMS enkripsi simetris dengan bahan kunci impor (`EXTERNAL`asal) hanya mendukung `On-demand` rotasi, sehingga tidak ada kolom tipe rotasi. Sebagai gantinya, setiap baris menampilkan status impor, deskripsi yang ditentukan pengguna, informasi kedaluwarsa, dan menu Tindakan. Status impor adalah Impor yang menunjukkan materi utama tersedia di dalam AWS KMS atau Impor tertunda yang menunjukkan materi utama tidak tersedia di dalamnya AWS KMS. Menu Tindakan dapat digunakan untuk menghapus materi kunci yang diimpor

atau mengimpor ulang materi kunci. Tindakan material kunci Hapus dinonaktifkan jika status impor materi kunci adalah Impor tertunda. Tindakan material kunci Reimport selalu tersedia. Anda tidak perlu menunggu materi kunci kedaluwarsa atau dihapus sebelum mengimpornya kembali.

Daftar rotasi dan bahan utama (AWS KMS API)

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 ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

ListKeyRotationsOperasi mencantumkan semua rotasi dan bahan utama untuk kunci KMS yang ditentukan. Untuk mengidentifikasi kunci KMS dalam operasi ini, gunakan ID kunci atau kunci ARN.

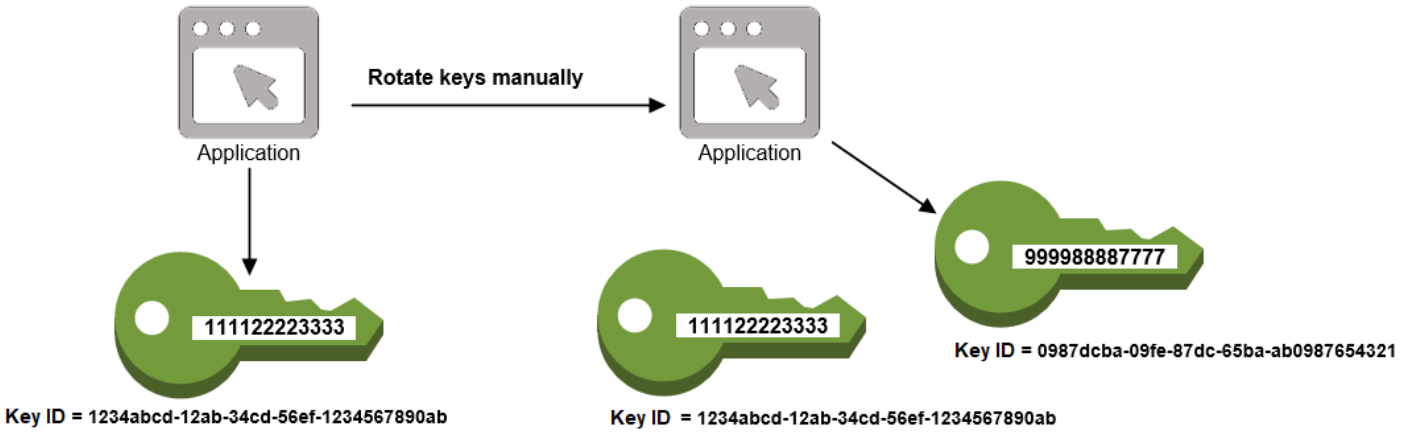
Operasi ini mendukung `IncludeKeyMaterial` parameter opsional. Nilai default dari parameter ini adalah`ROTATIONS_ONLY`. Jika Anda menghilangkan parameter ini, AWS KMS mengembalikan informasi tentang materi utama yang dibuat oleh rotasi kunci otomatis atau sesuai permintaan. Saat Anda menentukan nilai`ALL_KEY_MATERIAL`, AWS KMS tambahkan materi kunci pertama dan materi kunci impor apa pun yang menunggu rotasi ke respons. Parameter ini hanya dapat digunakan dengan kunci KMS yang mendukung rotasi kunci otomatis atau sesuai permintaan.

```
$ aws kms list-key-rotations --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --inlcude-key-material ALL_KEY_MATERIAL
{
    "Rotations": [
        {
            "KeyId": 1234abcd-12ab-34cd-56ef-1234567890ab,
            "KeyMaterialId":
 123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0,
            "KeyMaterialDescription": "KeyMaterialA",
            "ImportState": "PENDING_IMPORT",
            "KeyMaterialState": "NON_CURRENT"
        },
        {
            "KeyId": 1234abcd-12ab-34cd-56ef-1234567890ab,
            "KeyMaterialId":
 96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068,
            "ImportState": "IMPORTED",
            "KeyMaterialState": "CURRENT",
```

```
            "ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE",
            "RotationDate": "2025-05-01T15:50:51.045000-07:00",
            "RotationType": "ON_DEMAND"
        }
    ],
    "Truncated": false
}
```

Putar tombol secara manual

Anda mungkin ingin membuat kunci KMS baru dan menggunakannya sebagai pengganti kunci KMS saat ini alih-alih menggunakan rotasi kunci otomatis atau sesuai permintaan. 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 atau sesuai permintaan, seperti kunci KMS asimetris, kunci KMS HMAC, dan kunci KMS di toko kunci khusus.

> **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 Pelajari cara menggunakan alias dalam aplikasi.

> **Note**
>
> Alias yang mengarah ke versi terbaru dari kunci KMS yang diputar secara manual adalah solusi yang baik untuk DescribeKey, GetPublicKeydan operasi kriptografi seperti, Enkripsi DeriveSharedSecret,,,,, Tanda Tangan dan GenerateDataKeyPairVerifikasi GenerateMac. GenerateDataKeyVerifyMac 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 queryparameter 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
        },
    ]
}
```

Ubah kunci utama dalam satu set kunci Multi-wilayah

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.

Negara **`Updating`** kunci

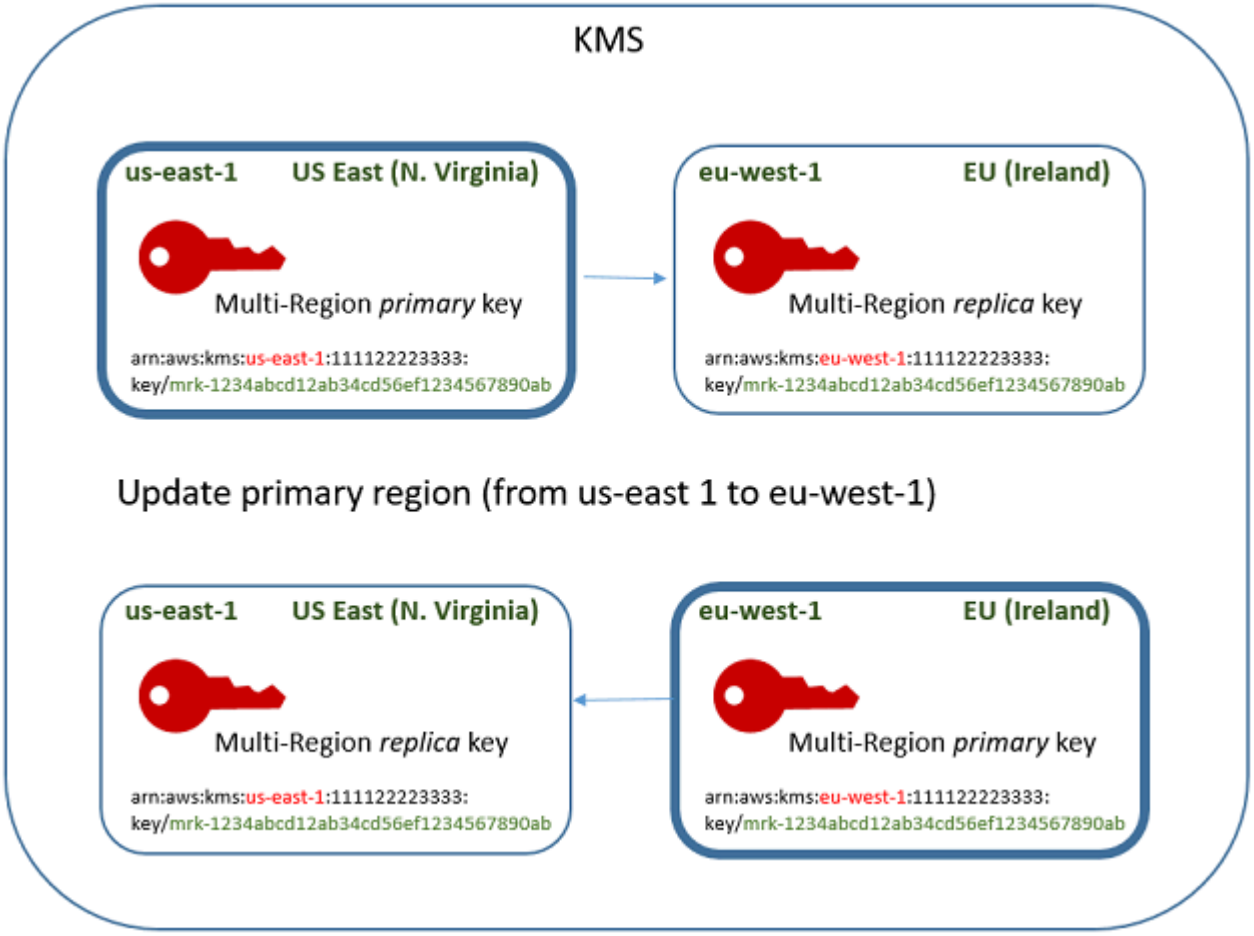
> Bahkan setelah `UpdatePrimaryRegion` operasi selesai, proses memperbarui Wilayah utama mungkin masih berlangsung selama beberapa detik lagi. Selama waktu ini, kunci primer lama dan baru memiliki status kunci sementara Memperbarui. Sementara status kunci adalah `Updating`, Anda dapat menggunakan kunci 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.

Untuk informasi tentang efek dari status kunci `Updating`, lihat Negara-negara kunci AWS KMS keys.

Cara kerjanya

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.

Perbarui Wilayah utama

Anda dapat mengonversi kunci replika menjadi kunci utama, yang mengubah kunci primer sebelumnya menjadi replika. Untuk memperbarui Wilayah utama, Anda memerlukan UpdatePrimaryRegion izin kms: di kedua Wilayah.

Anda dapat memperbarui Wilayah utama di AWS KMS konsol atau dengan menggunakan UpdatePrimaryRegionoperasi.

Menggunakan AWS KMS konsol

Anda dapat memperbarui kunci utama di AWS KMS konsol. Mulai pada halaman detail kunci untuk kunci primer saat ini.

1. Masuk ke Konsol Manajemen AWS 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.

Menggunakan AWS KMS API

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"
              }
           ]
        }
    }
}
```

Hapus sebuah AWS KMS key

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 atau Kunci milik AWS.

Sebelum menghapus kunci KMS, Anda mungkin ingin tahu berapa banyak ciphertext yang dienkripsi di bawah kunci KMS itu. AWS KMS tidak menyimpan informasi ini dan tidak menyimpan ciphertext apa pun. Untuk mendapatkan informasi ini, Anda harus menentukan penggunaan kunci KMS di masa lalu. Untuk bantuan, pergi keMenentukan penggunaan kunci KMS sebelumnya.

AWS KMS tidak 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 KMS mencatat entri di AWS CloudTrail log Anda ketika Anda menjadwalkan penghapusan kunci KMS dan ketika kunci KMS benar-benar dihapus.

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 KMS tidak 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. Lihat perinciannya di 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.

Pertimbangan khusus

Sebelum Anda menjadwalkan kunci Anda untuk dihapus, tinjau pertimbangan khusus berikut untuk menghapus kunci KMS tujuan khusus.

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 signifikan AWS 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, penghapusan penjadwalan 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. Lihat perinciannya di Deleting KMS keys with imported key material.

Menghapus tombol Multi-region

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.

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.

Menghapus kunci KMS dengan bahan 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.

Menghapus kunci KMS dari toko kunci AWS CloudHSM

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 lakukan upaya terbaik untuk menghapus materi kunci dari AWS CloudHSM cluster 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 materi kunci dari cadangan cluster. 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, gunakan DescribeKeyoperasi untuk mengidentifikasi 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 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.

Menghapus kunci KMS dari toko kunci eksternal

Menghapus kunci KMS dari penyimpanan kunci eksternal tidak berpengaruh pada kunci eksternal yang berfungsi sebagai bahan utamanya.

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 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.

Kontrol 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

Menggunakan AWS KMS konsol

Untuk memberikan izin kepada administrator kunci untuk menjadwalkan dan membatalkan penghapusan kunci.

1. Masuk ke Konsol Manajemen AWS 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.

Menggunakan AWS KMS API

Anda dapat menggunakan AWS Command Line Interface untuk menambahkan izin untuk penjadwalan dan pembatalan 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.

Jadwalkan penghapusan kunci

Prosedur berikut menjelaskan cara menjadwalkan penghapusan kunci dan membatalkan penghapusan kunci AWS KMS keys (kunci KMS) dalam AWS KMS menggunakan dan API. Konsol Manajemen AWS AWS KMS

> **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. Kontrol akses ke penghapusan kunci Anda juga dapat menggunakan kunci `kms:ScheduleKeyDeletionPendingWindowInDays`kondisi untuk lebih membatasi masa tunggu, seperti menegakkan masa tunggu minimum.

AWS KMS mencatat entri di AWS CloudTrail log Anda ketika Anda menjadwalkan penghapusan kunci KMS dan ketika kunci KMS benar-benar dihapus.

Menggunakan AWS KMS konsol

Di Konsol Manajemen AWS, Anda dapat menjadwalkan dan membatalkan penghapusan beberapa kunci KMS sekaligus.

Untuk menjadwalkan penghapusan kunci

1. Masuk ke Konsol Manajemen AWS 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. Peringatan tersebut mencakup jumlah kunci yang dipilih yang baru-baru ini digunakan dalam operasi kriptografi. Tabel menampilkan kolom Terakhir digunakan yang menunjukkan kapan setiap kunci KMS terakhir digunakan. 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. Pilih kotak centang di sebelah Saya mengonfirmasi bahwa saya ingin menjadwalkan kunci ini untuk dihapus. Kunci akan segera tidak dapat digunakan untuk operasi kriptografi. Setelah masa tunggu ***<number of days>*** sehari, kunci akan dihapus secara permanen.
10. Pilih Jadwalkan penghapusan.

Status kunci KMS berubah menjadi penghapusan Tertunda.

Menggunakan AWS KMS API

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 atau Kunci 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
}
```

Batalkan penghapusan kunci

Setelah Anda menjadwalkan kunci KMS untuk dihapus, Anda dapat membatalkan penghapusan kunci saat masih dalam status penghapusan tertunda. Anda dapat membatalkan penghapusan kunci di AWS KMS konsol atau dengan menggunakan operasi. CancelKeyDeletion Setelah Anda membatalkan penghapusan kunci KMS yang tertunda, status kunci dari kunci KMS adalah. `Disabled` Untuk informasi selengkapnya tentang mengaktifkan kunci KMS, lihat. Aktifkan dan nonaktifkan kunci

Menggunakan AWS KMS konsol

Untuk membatalkan penghapusan kunci

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. 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.

Menggunakan AWS KMS API

Gunakan `aws kms cancel-key-deletion`perintah untuk membatalkan penghapusan kunci dari AWS CLI seperti yang ditunjukkan pada 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.

Buat alarm yang mendeteksi penggunaan kunci KMS tertunda penghapusan

Anda dapat menggabungkan fitur AWS 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 AWS KMS API yang menampilkan pesan galat ini, lihatNegara-negara kunci AWS KMS keys.

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. Deleting asymmetric KMS keys

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 Buat 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.
Jenis ambang	Statis
Ketentuan	Setiap kali *`metric-name`* lebih besar/sama dari 1
Data menunjuk ke 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.

Hapus materi 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.

Kunci enkripsi simetris dapat memiliki beberapa bahan utama yang terkait dengannya. Untuk kunci ini, KMS memberikan pengenal unik untuk setiap materi kunci. Anda dapat menggunakan ListKeyRotationsAPI untuk melihat pengenal material utama ini dan status material kunci yang sesuai (lihat RotationsListEntry). Status material utama `PENDING_ROTATION` atau `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` menunjukkan materi kunci tidak terkait secara permanen dengan kunci KMS. Penghapusan atau kedaluwarsa materi kunci yang terkait secara permanen mengubah status kunci menjadi Impor tertunda. Anda dapat menghapus materi kunci tertentu dengan menentukan pengenalnya menggunakan `key-material-id` parameter di API. DeleteImportedKeyMaterial

Pertimbangan untuk kunci Multi-wilayah

- Saat Anda menghapus materi kunci kunci Wilayah utama yang ada di `PENDING_ROTATION` atau `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` status, Anda juga akan menghapus materi utama untuk kunci Region replika.
- Jika Anda menghapus materi kunci di kunci Region primer atau replika, hanya kunci spesifik yang terpengaruh dan kunci multi-wilayah terkait lainnya tetap tidak berubah. Kunci Wilayah primer atau replika apa pun yang memiliki semua materi kunci yang terkait secara permanen terus dapat digunakan dalam operasi kriptografi.

> **Warning**
>
> `key-material-id`Parameternya opsional dan jika Anda tidak menentukannya, AWS KMS akan menghapus materi kunci saat ini.

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 impor juga dengan cepat menghentikan penggunaan kunci KMS. Namun, jika penghapusan tidak dibatalkan

selama masa tunggu, kunci KMS, materi kunci terkait, dan semua metadata kunci akan dihapus secara permanen. Lihat perinciannya di Deleting KMS keys with imported key material.

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.

Bagaimana menghapus materi kunci memengaruhi layanan AWS

> Saat Anda menghapus materi kunci apa pun, 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 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.

Menggunakan AWS KMS konsol

Anda dapat menggunakan AWS KMS konsol untuk menghapus materi utama.

1. Masuk ke Konsol Manajemen AWS 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 tindakan berikut:
   - Pilih kotak centang untuk kunci KMS dengan bahan kunci impor. Pilih Tindakan kunci, Hapus material kunci. Untuk kunci enkripsi simetris yang memiliki beberapa materi utama yang terkait dengannya, ini akan menghapus materi kunci saat ini.
   - Untuk kunci KMS enkripsi simetris dengan bahan kunci yang diimpor, pilih alias atau ID kunci kunci KMS. Pilih tab Bahan kunci dan rotasi. Tabel bahan utama akan mencantumkan semua bahan utama yang terkait dengan kunci. Pilih Hapus materi kunci dari menu Tindakan di baris yang sesuai dengan materi kunci yang ingin Anda hapus.
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. Jika materi

kunci yang dihapus dalam `PENDING_ROTATION` status, tidak ada perubahan pada status kunci KMS.

Menggunakan 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. Perintah berikut menghapus materi kunci saat ini yang mungkin merupakan satu-satunya materi kunci yang terkait dengan kunci.

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

Untuk menghapus materi kunci tertentu, tentukan materi kunci yang diidentifikasi menggunakan `key-material-id` parameter. Ganti *123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0* dengan pengenal materi kunci yang ingin Anda hapus.

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-material-id 123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0
```

Hasilkan 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.

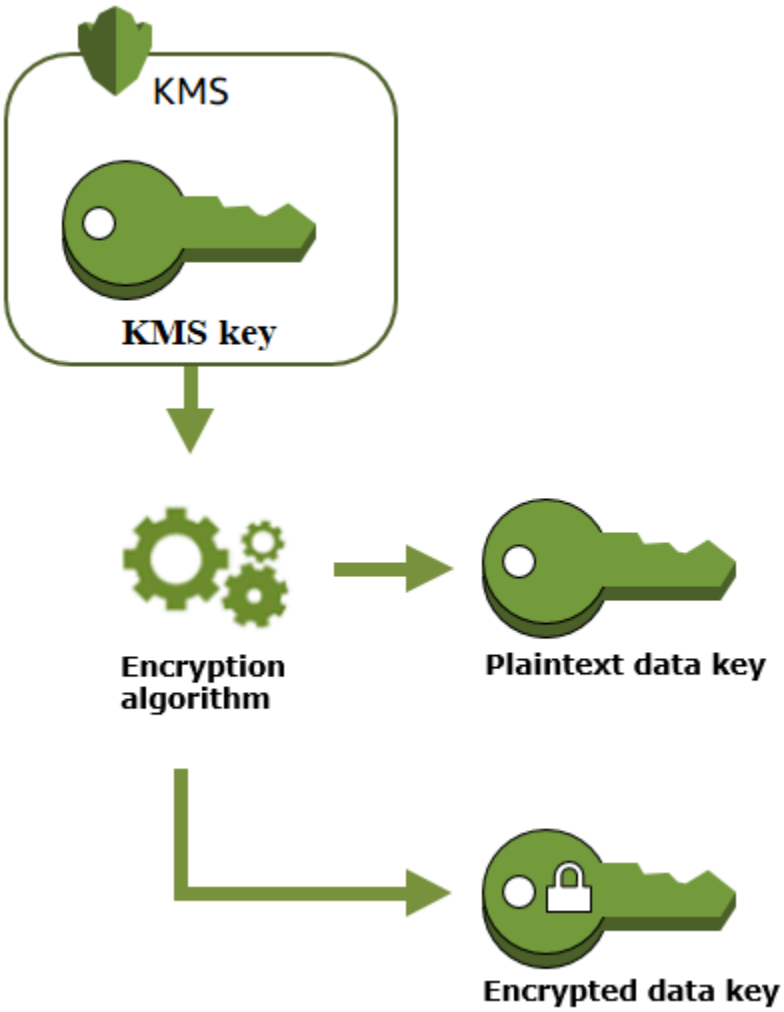
AWS KMS menghasilkan, mengenkripsi, dan mendekripsi kunci data. Namun, AWS KMS tidak menyimpan, mengelola, atau melacak kunci data Anda, atau melakukan operasi kriptografi dengan kunci data. Anda harus menggunakan dan mengelola kunci data di luar AWS KMS. Untuk bantuan menggunakan kunci data dengan aman,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

Topik

- Membuat kunci data
- Bagaimana operasi kriptografi dengan kunci data bekerja
- Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data

Membuat kunci data

Untuk membuat kunci data, panggil GenerateDataKeyoperasi. AWS KMS menghasilkan 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 KMS juga mendukung GenerateDataKeyWithoutPlaintextoperasi, yang mengembalikan hanya kunci data terenkripsi. Saat Anda perlu menggunakan kunci data, minta AWS KMS untuk mendekripsi.

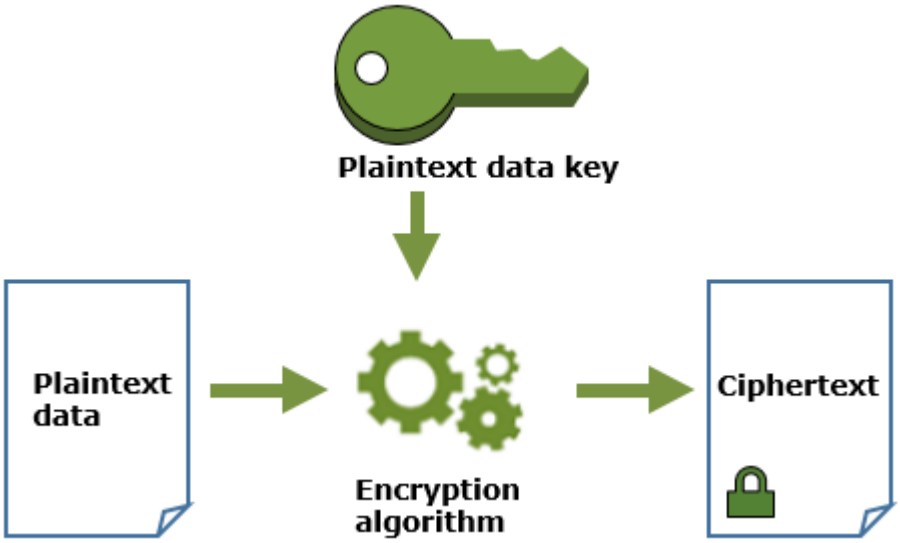
Bagaimana operasi kriptografi dengan kunci data bekerja

Topik berikut menjelaskan bagaimana kunci data yang dihasilkan oleh pekerjaan GenerateDataKeyatau GenerateDataKeyWithoutPlaintextoperasi.

Mengenkripsi data dengan kunci data

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

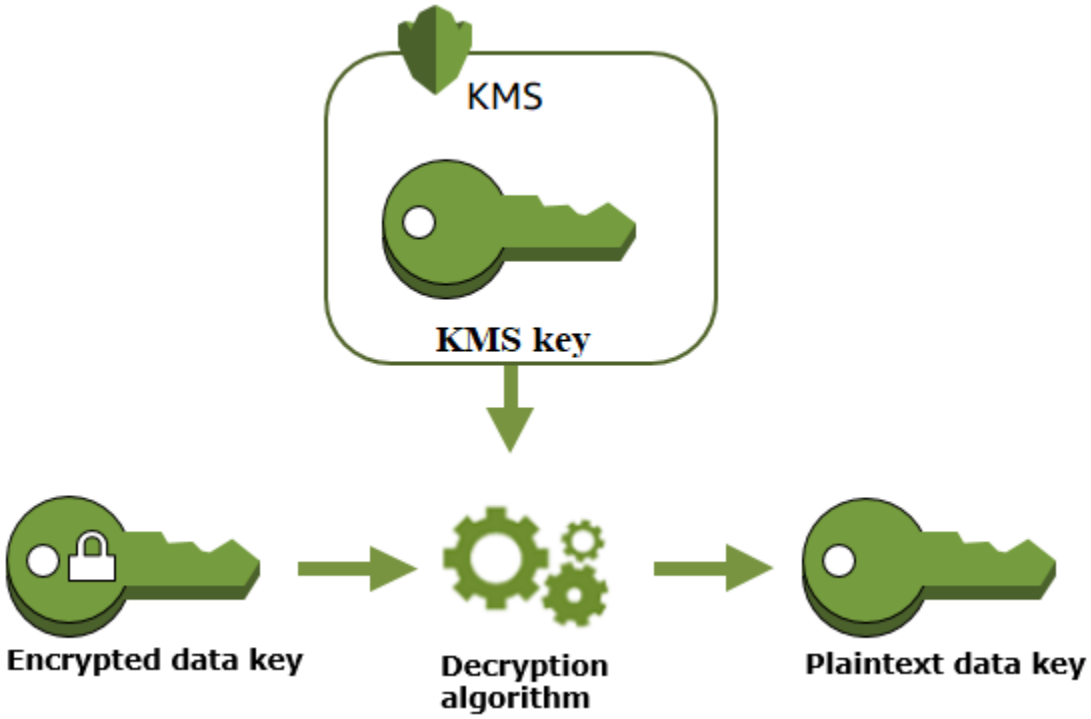
Setelah menggunakan kunci data plaintext untuk mengenkripsi data, hapus kunci data dari memori sesegera mungkin. Anda dapat dengan aman menyimpan kunci data terenkripsi dengan data terenkripsi sehingga tersedia untuk mendekripsi data.



Mendekripsi data dengan kunci data

Untuk mendekripsi data Anda, teruskan kunci data terenkripsi ke operasi Dekripsi. AWS KMS menggunakan 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 impor kedaluwarsa. Jika kunci KMS dengan `EXTERNAL` asal memiliki beberapa bahan kunci yang terkait, penghapusan atau kedaluwarsa materi kunci apa pun akan menyebabkan kunci menjadi tidak dapat digunakan.

- 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 bagian Layanan 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.

Hasilkan pasangan kunci data

Kunci KMS asimetris mewakili data key pair. 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 dan dekripsi sisi klien, penandatanganan dan verifikasi di luar AWS KMS, atau untuk membangun rahasia bersama antara dua rekan.

Berbeda dengan pasangan kunci data yang dihasilkan alat seperti OpenSSL AWS 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.

Topik

- Membuat pasangan kunci data
- Bagaimana operasi kriptografi dengan pasangan kunci data bekerja

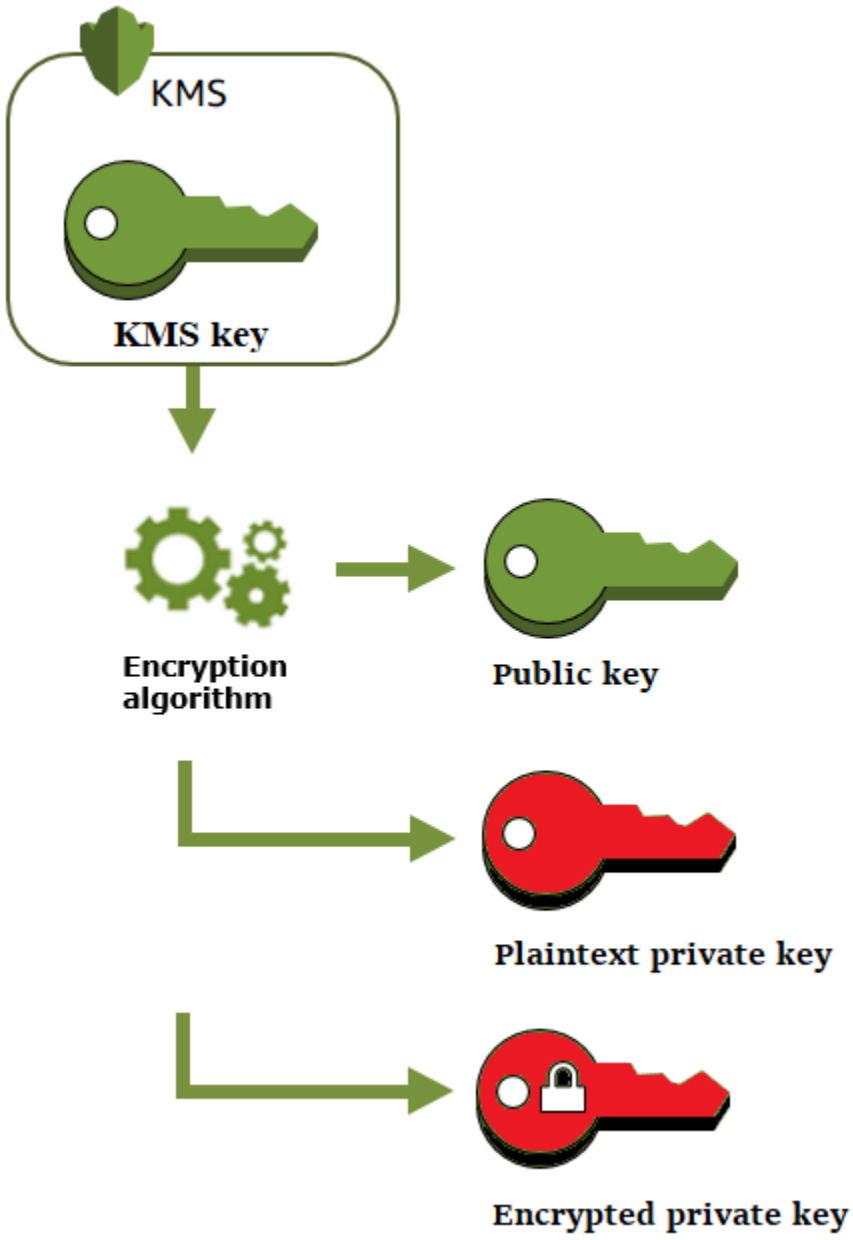
Membuat pasangan kunci data

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.



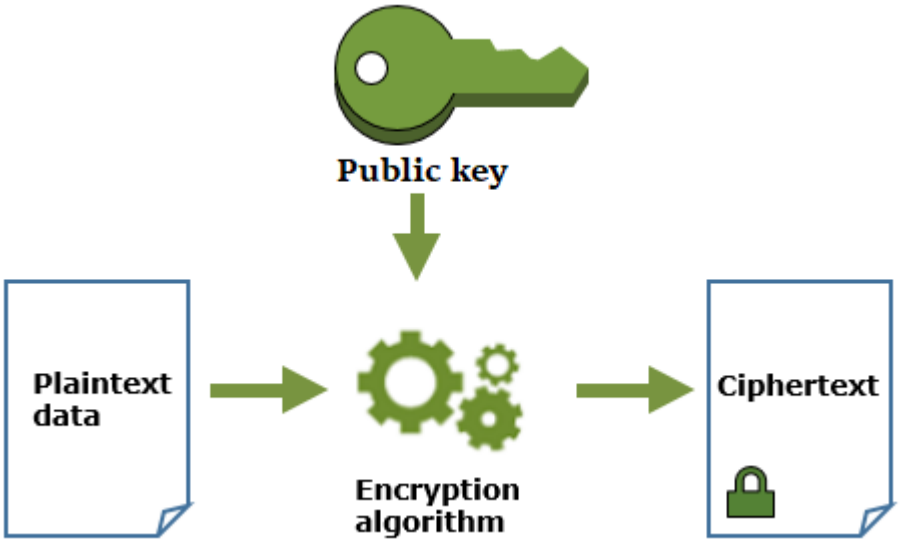
Bagaimana operasi kriptografi dengan pasangan kunci data bekerja

Topik berikut menjelaskan operasi kriptografi apa yang dapat Anda lakukan dengan pasangan kunci data yang dihasilkan oleh GenerateDataKeyPairWithoutPlaintextoperasi GenerateDataKeyPairatau atau cara kerjanya.

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.

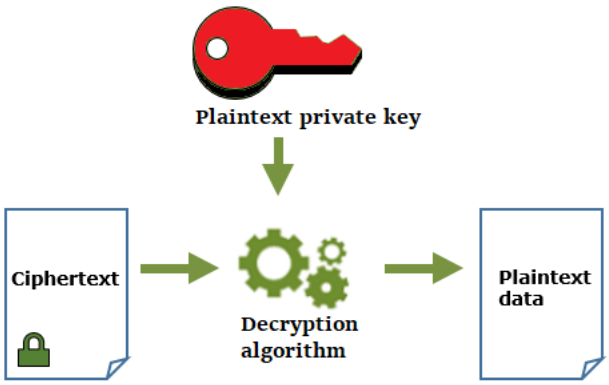


Mendekripsi data dengan pasangan kunci data

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.



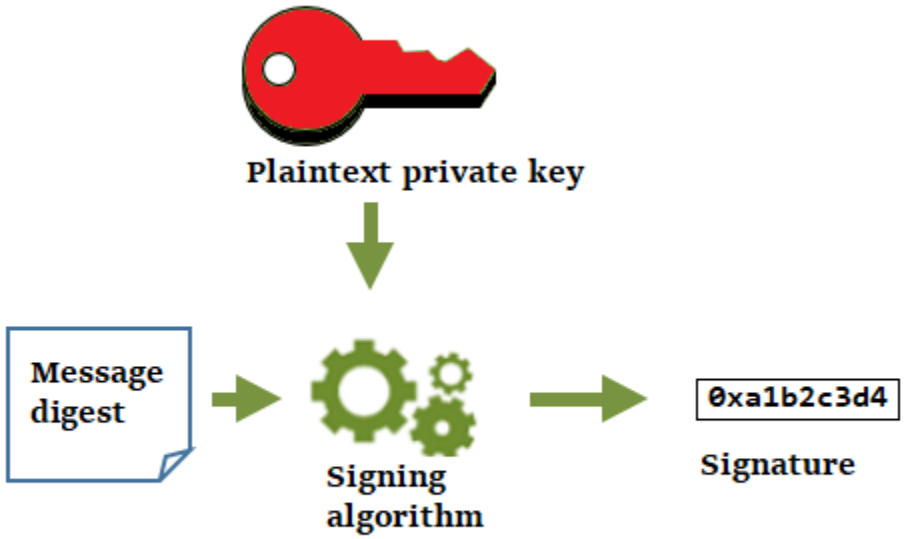
Menandatangani pesan dengan pasangan kunci data

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 KMS menggunakan 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.

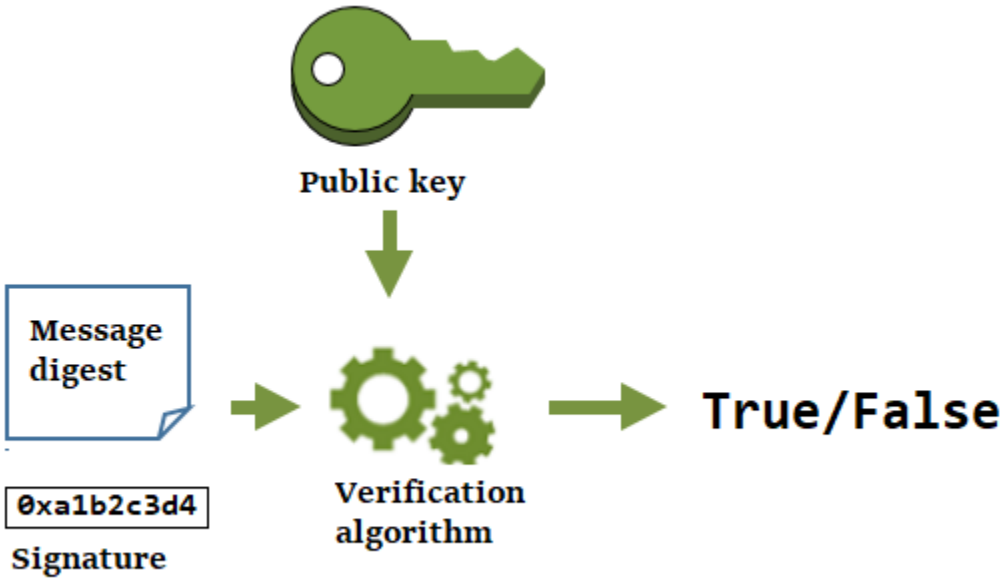


Memverifikasi tanda tangan dengan pasangan kunci data

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.



Dapatkan rahasia bersama dengan pasangan kunci data

Kesepakatan kunci memungkinkan dua rekan, masing-masing memiliki elliptic-curve public-private key pair, untuk membangun rahasia bersama melalui saluran yang tidak aman. Untuk mendapatkan rahasia bersama, kedua rekan harus bertukar kunci publik mereka melalui saluran komunikasi yang tidak aman (seperti internet). Kemudian, masing-masing pihak menggunakan kunci pribadi mereka dan kunci publik rekan mereka untuk menghitung rahasia bersama yang sama menggunakan algoritma perjanjian kunci. Anda dapat menggunakan nilai rahasia bersama untuk mendapatkan kunci simetris yang dapat mengenkripsi dan mendekripsi data yang dikirim antara dua rekan, atau yang dapat menghasilkan dan memverifikasi HMAC.

> **Note**
>
> AWS KMS sangat menyarankan untuk memverifikasi bahwa kunci publik yang Anda terima berasal dari pihak yang diharapkan sebelum menggunakannya untuk mendapatkan rahasia bersama.

Lakukan operasi offline dengan kunci publik

Dalam kunci KMS asimetris, kunci pribadi dibuat AWS KMS dan tidak pernah meninggalkan AWS KMS tanpa terenkripsi. Untuk menggunakan kunci pribadi, Anda harus menelepon AWS KMS. Anda dapat menggunakan kunci publik di dalam AWS KMS dengan memanggil operasi AWS KMS API. Atau, Anda dapat mengunduh kunci publik dan berbagi untuk digunakan di luar AWS KMS.

Anda dapat membagikan kunci publik untuk membiarkan orang lain mengenkripsi data di luar AWS KMS itu Anda dapat mendekripsi hanya dengan kunci pribadi Anda. Atau, untuk mengizinkan orang lain memverifikasi tanda tangan digital di luar AWS KMS yang telah Anda hasilkan dengan kunci privat Anda. Atau, untuk membagikan kunci publik Anda dengan rekan untuk mendapatkan rahasia bersama.

Saat Anda menggunakan kunci publik di kunci KMS asimetris di dalam AWS KMS, Anda mendapat manfaat dari otentikasi, otorisasi, dan pencatatan yang merupakan bagian dari setiap AWS KMS operasi. Anda juga mengurangi risiko mengenkripsi data yang tidak dapat didekripsi. Fitur-fitur ini tidak efektif di luar AWS KMS. Lihat perinciannya di Pertimbangan khusus untuk mengunduh kunci publik.

> **Tip**
>
> Mencari kunci data atau kunci SSH? Topik ini menjelaskan cara mengelola kunci asimetris di AWS 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 mengunduh kunci publik yang terkait dengan instans Amazon EC2, lihat Mengambil kunci publik di Panduan Pengguna Amazon EC2 dan Panduan Pengguna Amazon EC2.

Topik

- Pertimbangan khusus untuk mengunduh kunci publik
- Unduh kunci publik
- Contoh operasi offline

Pertimbangan khusus untuk mengunduh kunci publik

Untuk melindungi kunci KMS Anda, AWS KMS menyediakan kontrol akses, enkripsi yang diautentikasi, dan log terperinci dari setiap operasi. AWS KMS juga memungkinkan Anda untuk mencegah penggunaan kunci KMS, sementara atau permanen. Akhirnya, AWS KMS operasi dirancang untuk meminimalkan risiko mengenkripsi data yang tidak dapat didekripsi. Fitur-fitur ini tidak tersedia saat 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. Pasangan kunci RSA secara struktural sama apakah penggunaan kuncinya atau. `ENCRYPT_DECRYPT` `SIGN_VERIFY` Jadi Anda dapat menggunakan kunci publik yang diunduh dari kunci `SIGN_VERIFY` RSA KMS untuk mengenkripsi data di luar. AWS KMS Namun, tidak AWS KMS dapat mendekripsi data tersebut menggunakan kunci pribadi yang sesuai. AWS KMS memberlakukan penggunaan kunci untuk operasi yang dilakukannya, sehingga menolak permintaan Dek ripsi pada `SIGN_VERIFY` kunci.

Pembatasan algoritme

Pembatasan pada algoritma enkripsi dan penandatanganan yang AWS KMS mendukung tidak efektif di luar AWS KMS. Jika Anda mengenkripsi data dengan kunci publik dari kunci KMS di luar AWS KMS, dan menggunakan algoritma enkripsi yang AWS KMS tidak mendukung, data tidak dapat didekripsi.

Menonaktifkan dan menghapus kunci KMS

Tindakan yang dapat Anda ambil untuk mencegah penggunaan kunci KMS dalam operasi kriptografi di dalamnya 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 dipuli AWS KMS hkan.

Pencatatan log

AWS CloudTrail log yang mencatat setiap AWS KMS operasi, termasuk permintaan, respons, tanggal, waktu, dan pengguna yang berwenang, 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 (hanya Wilayah China).

Unduh kunci publik

Anda dapat mengunduh kunci publik dari pasangan kunci KMS asimetris di AWS KMS konsol atau dengan menggunakan GetPublicKey operasi. Untuk mengunduh kunci publik, Anda harus memiliki `kms:GetPublicKey` izin pada kunci KMS asimetris.

Kunci publik yang dikem AWS KMS balikan adalah kunci DER-encoded X.509 publik, juga dikenal sebagai `SubjectPublicKeyInfo` (SPKI), seperti yang didefinisikan dalam RFC 5280. Saat Anda menggunakan HTTP API atau AWS CLI, nilainya adalah Base64-encoded. Kalau tidak, tidak Base64-encoded.

Untuk mengunduh kunci publik dari pasangan kunci KMS asimetris, Anda memerlukan `kms:GetPublicKey` izin. Untuk informasi selengkapnya tentang AWS KMS izin, lihatReferensi izin.

Menggunakan AWS KMS konsol

Anda dapat menggunakan Konsol Manajemen AWS untuk melihat, menyalin, dan mengunduh kunci publik dari kunci KMS asimetris di Anda Akun AWS. Untuk mengunduh kunci publik dari kunci KMS asimetris secara berbeda Akun AWS, gunakan AWS KMS API.

1. Masuk ke Konsol Manajemen AWS 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 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.

Menggunakan AWS KMS API

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 algoritma 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, kunci ARN, nama alias, atau alias ARN. Bila menggunakan nama alias, beri prefiks dengan alias/. Untuk menentukan kunci KMS yang berbeda Akun AWS, Anda harus menggunakan kunci ARN atau alias ARN.

Sebelum menjalankan perintah ini, ganti nama alias contoh dengan pengidentifikasi 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..."
}
```

Contoh operasi offline

Setelah mengunduh kunci publik dari pasangan kunci KMS asimetris Anda, Anda dapat membagikannya dengan orang lain dan menggunakannya untuk melakukan operasi offline.

AWS CloudTrail log yang mencatat setiap AWS KMS operasi, termasuk permintaan, respons, tanggal, waktu, dan pengguna yang berwenang, tidak mencatat penggunaan kunci publik di luar AWS KMS.

Topik ini memberikan contoh operasi offline dan detail yang AWS KMS disediakan alat untuk mempermudah operasi offline.

Topik

- Memperoleh rahasia bersama secara offline
- Verifikasi offline dengan pasangan ML-DSA kunci
- Verifikasi offline dengan pasangan kunci SM2 (hanya Wilayah China)

Memperoleh rahasia bersama secara offline

Anda dapat mengunduh kunci publik dari pasangan kunci ECC Anda untuk digunakan dalam operasi offline, yaitu operasi di luar. AWS KMS

Panduan OpenSSL berikut menunjukkan satu metode untuk memperoleh rahasia bersama di luar AWS KMS menggunakan kunci publik dari pasangan kunci ECC KMS dan kunci pribadi yang dibuat dengan OpenSSL.

1. Buat pasangan kunci ECC di OpenSSL dan persiapkan untuk digunakan dengan. AWS KMS

```
// Create an ECC key pair in OpenSSL and save the private key in
 openssl_ecc_key_priv.pem
export OPENSSL_CURVE_NAME="P-256"
export KMS_CURVE_NAME="ECC_NIST_P256"

export OPENSSL_KEY1_PRIV_PEM="openssl_ecc_key1_priv.pem"
openssl ecparam -name ${OPENSSL_CURVE_NAME} -genkey -out ${OPENSSL_KEY1_PRIV_PEM}


// Derive the public key from the private key
export OPENSSL_KEY1_PUB_PEM="openssl_ecc_key1_pub.pem"
```

```
openssl ec -in ${OPENSSL_KEY1_PRIV_PEM} -pubout -outform pem \
    -out ${OPENSSL_KEY1_PUB_PEM}

// View the PEM file containing the public key and extract the public key as a
// Base64 encoded string into OPENSSL_KEY1_PUB_BASE64 for use with AWS KMS
export OPENSSL_KEY1_PUB_BASE64=`cat ${OPENSSL_KEY1_PUB_PEM} | \
    tee /dev/stderr | grep -v "PUBLIC KEY" | tr -d "\n"`
```

2. Buat pasangan kunci perjanjian kunci ECC AWS KMS dan persiapkan untuk digunakan dengan OpenSSL.

```
// Create a KMS key on the same curve as the key pair from step 1
// with a key usage of KEY_AGREEMENT
// Save its ARN in KMS_KEY1_ARN.
export KMS_KEY1_ARN=`aws kms create-key --key-spec ${KMS_CURVE_NAME} \
    --key-usage KEY_AGREEMENT | tee /dev/stderr | jq -r .KeyMetadata.Arn`

// Download the public key and save the Base64-encoded version in KMS_KEY1_PUB_BASE64

export KMS_KEY1_PUB_BASE64=`aws kms get-public-key --key-id ${KMS_KEY1_ARN} | \
    tee /dev/stderr | jq -r .PublicKey`

// Create a PEM file for the public KMS key for use with OpenSSL
export KMS_KEY1_PUB_PEM="aws_kms_ecdh_key1_pub.pem"
echo "-----BEGIN PUBLIC KEY-----" > ${KMS_KEY1_PUB_PEM}
echo ${KMS_KEY1_PUB_BASE64} | fold -w 64 >> ${KMS_KEY1_PUB_PEM}
echo "-----END PUBLIC KEY-----" >> ${KMS_KEY1_PUB_PEM}
```

3. Dapatkan rahasia bersama di OpenSSL menggunakan kunci pribadi di OpenSSL dan kunci KMS publik.

```
export OPENSSL_SHARED_SECRET1_BIN="openssl_shared_secret1.bin"
openssl pkeyutl -derive -inkey ${OPENSSL_KEY1_PRIV_PEM} \
    -peerkey ${KMS_KEY1_PUB_PEM} -out ${OPENSSL_SHARED_SECRET1_BIN}
```

Verifikasi offline dengan pasangan ML-DSA kunci

AWS KMS mendukung varian ML-DSA penandatanganan lindung nilai, seperti yang dijelaskan dalam Standar Pemrosesan Informasi Federal (FIPS) 204 bagian 3.4 standar untuk pesan hingga 4 KB byte.

Untuk menandatangani pesan yang lebih besar dari 4 KB, Anda melakukan langkah pra-pemrosesan pesan di luar. AWS KMS Langkah hashing ini membuat perwakilan pesan 64-byte μ, seperti yang didefinisikan dalam NIST FIPS 204, bagian 6.2.

AWS KMS memiliki jenis pesan yang dipanggil `EXTERNAL_MU` untuk pesan yang lebih besar dari 4 KB. Bila Anda menggunakan ini sebagai pengganti jenis `RAW` pesan, AWS KMS:

- Mengasumsikan Anda telah melakukan langkah hashing
- Melewati proses hashing internalnya
- Bekerja dengan pesan dari berbagai ukuran

Saat Anda memverifikasi pesan, metode yang Anda gunakan bergantung pada batasan ukuran sistem atau pustaka eksternal dan apakah metode tersebut mendukung perwakilan pesan 64-byte μ:

- Jika pesan lebih kecil dari batasan ukuran, gunakan jenis `RAW` pesan.
- Jika pesan lebih besar dari batasan ukuran, gunakan perwakilan μ di sistem eksternal.

Bagian berikut menunjukkan cara menandatangani pesan menggunakan AWS KMS dan memverifikasi pesan menggunakan OpenSSL. Kami memberikan contoh untuk pesan di bawah dan di atas batas ukuran pesan 4 KB yang diberlakukan oleh AWS KMS. OpenSSL tidak memberlakukan batasan ukuran pesan untuk verifikasi.

Untuk kedua contoh, pertama-tama dapatkan kunci publik dari AWS KMS. Gunakan perintah AWS CLI berikut:

```
aws kms get-public-key \
    --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
    --output text \
    --query PublicKey | base64 --decode > public_key.der
```

Ukuran pesan kurang dari 4KB

Untuk pesan di bawah 4 KB, gunakan jenis `RAW` pesan dengan AWS KMS. Meskipun Anda dapat menggunakan`EXTERNAL_MU`, itu tidak perlu untuk pesan dalam batas ukuran.

Gunakan AWS CLI perintah berikut untuk menandatangani pesan:

```
aws kms sign \
```

```
    --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
    --message 'your message' \
    --message-type RAW \
    --signing-algorithm ML_DSA_SHAKE_256 \
    --output text \
    --query Signature | base64 --decode > ExampleSignature.bin
```

Untuk memverifikasi pesan ini menggunakan OpenSSL gunakan perintah berikut:

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature
 ExampleSignature.bin
```

Ukuran pesan lebih dari 4KB

Untuk menandatangani pesan yang lebih besar dari 4KB, gunakan jenis `EXTERNAL_MU` pesan. Saat Anda menggunakan`EXTERNAL_MU`, Anda melakukan pre-hash pesan secara eksternal ke perwakilan 64-byte μ seperti yang didefinisikan dalam NIST FIPS 204 bagian 6.2 dan meneruskannya ke operasi penandatanganan atau verifikasi. Perhatikan bahwa ini berbeda dari "Pre-hash MLDSA" atau HashML-DSA didefinisikan dalam NIST FIPS 204 bagian 5.4.

1. Pertama, buat awalan pesan. Awalan berisi pemisah domain, panjang konteks apa pun, dan konteksnya. Default untuk pemisah domain dan panjang konteks adalah nol.
2. Tambahkan awalan pesan ke pesan.
3. Gunakan SHAKE256 untuk hash kunci publik dan menambahkannya ke hasil langkah 2.
4. Akhirnya, hash hasil langkah 3 untuk menghasilkan 64-byte`EXTERNAL_MU`.

Contoh berikut menggunakan OpenSSL 3.5 untuk membangun: `EXTERNAL_MU`

```
{
    openssl asn1parse -inform DER -in public_key.der -strparse 17 -noout -out - 2>/dev/
null |
    openssl dgst -provider default -shake256 -xoflen 64 -binary;
    printf '\x00\x00';
    echo -n "your message"
} | openssl dgst -provider default -shake256 -xoflen 64 -binary > mu.bin
```

Setelah Anda membuat `mu.bin` file, panggil AWS KMS API dengan perintah berikut untuk menandatangani pesan:

```
aws kms sign \
    --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
    --message fileb://mu.bin \
    --message-type EXTERNAL_MU \
    --signing-algorithm ML_DSA_SHAKE_256 \
    --output text \
    --query Signature | base64 --decode > ExampleSignature.bin
```

Tanda tangan yang dihasilkan sama dengan RAW tanda tangan pada pesan asli. Anda dapat menggunakan perintah OpenSSL 3.5 yang sama untuk memverifikasi pesan:

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature
 ExampleSignature.bin
```

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, `MessageType:RAW`, ke S ign API, AWS KMS menggunakan ID pembeda default,1234567812345678, yang ditentukan oleh OSCCA di GM/T 0009-2012. Anda tidak dapat menentukan ID pembeda Anda sendiri di dalamnya AWS KMS.

Namun, jika Anda membuat ringkasan pesan di luar AWS, Anda dapat menentukan ID pembeda Anda sendiri, lalu meneruskan ringkasan pesan, `MessageType:DIGEST`, AWS KMS untuk ditandatangani. Untuk melakukan ini, ubah `DEFAULT_DISTINGUISHING_ID` nilai di `SM2OfflineOperationHelper` kelas. ID pembeda yang Anda tentukan dapat berupa string apa pun dengan panjang hingga 8.192 karakter. Setelah AWS KMS menandatangani ringkasan pesan, Anda memerlukan ringkasan pesan atau pesan dan ID pembeda yang digunakan untuk menghitung ringkasan untuk memverifikasinya secara offline.

> **Important**
>
> Kode `SM2OfflineOperationHelper` referensi dirancang agar kompatibel dengan Bouncy Castle versi 1.68. Untuk bantuan dengan versi lain, hubungi bouncy castle.org.

`SM2OfflineOperationHelper` kelas

Untuk membantu Anda dengan operasi offline dengan kunci SM2, `SM2OfflineOperationHelper` kelas untuk Java memiliki metode yang melakukan tugas untuk Anda. Anda dapat menggunakan kelas pembantu ini sebagai model untuk penyedia kriptografi lainnya.

Di dalamnya AWS KMS, konversi ciphertext mentah dan perhitungan ringkasan 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 KMS mengkonfirmasi 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 ringkasan pesan

  Untuk membuat ringkasan pesan offline yang dapat Anda gunakan untuk verifikasi offline, atau yang dapat Anda lanjutkan AWS KMS untuk menandatangani, gunakan `calculateSM2Digest` metode ini. `calculateSM2Digest`Metode ini menghasilkan ringkasan 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 default1234567812345678, tetapi Anda dapat mengatur ID pembeda Anda sendiri dengan mengubah nil `DEFAULT_DISTINGUISHING_ID` ainya.

- Verifikasi

  Untuk memverifikasi tanda tangan secara offline, gunakan `offlineSM2DSAVerify` metode ini. `offlineSM2DSAVerify`Metode ini menggunakan ringkasan 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 diurai 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 `offlineSM2PKEEncrypt` metode ini. Metode ini memastikan ciphertext dalam format AWS KMS dapat mendekripsi.

`offlineSM2PKEEncrypt`Metode mengenkripsi plaintext, dan kemudian mengubah 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 membuat ringkasan 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");
    }
}
```

Monitor 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:

- AWS KMS Aktivitas API untuk operasi pesawat data. Ini adalah operasi kriptografi yang menggunakan kunci KMS, seperti Dekripsi, Enkripsi, dan. ReEncryptGenerateDataKey
- AWS KMS Aktivitas API untuk operasi pesawat 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 KMS Metrik 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

AWS menyediakan 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 CloudTrail Pemantauan 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 Pantau tombol KMS dengan Amazon CloudWatch.
- 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 Pantau tombol KMS dengan Amazon EventBridge 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. AWS Dasbor AWS KMS CloudWatch AWS Trusted Advisor,,, dan lainnya memberikan at-a-glance pandangan tentang keadaan AWS lingkungan Anda.

Anda dapat menyesuaikan halaman kunci yang dikelola Pelanggan di AWS KMS konsol 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 kustom)

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
- Data metrik grafik untuk memecahkan masalah dan mengungkap tren
- Cari dan telusuri semua metrik AWS sumber daya Anda
- Membuat dan mengedit alarm untuk menerima notifikasi terkait masalah

AWS Trusted Advisor dapat membantu Anda memantau AWS sumber daya Anda untuk meningkatkan kinerja, keandalan, keamanan, dan efektivitas biaya. Empat Trusted Advisor cek tersedia untuk semua pengguna; lebih dari 50 cek tersedia untuk pengguna dengan paket dukungan Bisnis atau Perusahaan. Lihat informasi yang lebih lengkap di AWS Trusted Advisor.

Pencatatan log AWS KMS Panggilan 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, CloudFormation template, AWS Command Line Interface (AWS CLI), dan Alat AWS untuk 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 semua operasi yang berhasil dan, dalam beberapa skenario, percobaan panggilan yang gagal, seperti ketika pemanggil ditolak akses ke sumber daya. Cross-account operasi 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 peristiwa dari jejak.

Pelajari lebih lanjut:

- Untuk contoh CloudTrail log AWS KMS operasi untuk platform yang dibuktikan, lihatMemantau permintaan yang dibuktikan.

Topik

- Temuan AWS KMS entri log di CloudTrail
- Tidak termasuk AWS KMS peristiwa dari jejak

- Contoh dari AWS KMS entri log

Temuan AWS KMS entri log 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` danEnableKey , tetapi tidak`ListKeys`.
Nama sumber daya (`ResourceName`)	ARN kunci (atau ID kunci dan kunci ARN)	Operasi manajemen yang mengubah kunci KMS tertentu, seperti `CreateKey` danEnableKey , 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 peristiwa 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 Konsol Manajemen AWS 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 kekms.amazonaws.com.

Contoh dari AWS KMS entri 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 dari platform yang AWS KMS dibuktikan, lihat. Memantau permintaan yang dibuktikan

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
- GetKeyLastUsage
- 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
- Amazon EC2 contoh satu
- Amazon EC2 contoh dua

CancelKeyDeletion

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil CancelKeyDeletionoperasi. Untuk informasi tentang menghapus AWS KMS keys, lihatHapus sebuah AWS KMS key.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T21:53:17Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "CancelKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "e3452e68-d4b0-4ec7-a768-7ae96c23764f",
    "eventID": "d818bf03-6655-48e9-8b26-f279a07075fd",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


ConnectCustomKeyStore

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil `ConnectCustomKeyStore`operasi. Untuk informasi tentang menghubungkan toko kunci kustom, lihatPutuskan 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 membuat alias di AWS KMS, lihatBuat 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, lihatBuat 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 membuat hibah di AWS KMS, lihatHibah 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 kunci ARN.

Contoh berikut menunjukkan entri `CreateGrant` log untuk hibah dengan kendala konteks 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": "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": "service-name.amazonaws.com"
  },
  "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"
}
```

Contoh berikut menunjukkan entri `CreateGrant` log untuk hibah utama layanan. Hibah ini menggunakan `GranteeServicePrincipal` parameter untuk menentukan pokok AWS layanan sebagai penerima hibah, dan menyertakan batasan `SourceArn` hibah.

```
{
  "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": "2026-03-04T18:22:45Z",
  "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": {
          "sourceArn": "arn:aws:dynamodb:us-east-1:111122223333:table/ExampleTable"
      },
      "operations": [
        "Encrypt",
        "Decrypt",
        "GenerateDataKey"
      ],
      "granteeServicePrincipal": "service-name.amazonaws.com",
      "retiringServicePrincipal": "service-name.amazonaws.com"
  },
  "responseElements": {
      "grantId": "a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2",
      "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "readOnly": false,
  "resources": [{
```

```
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

> **Note**
>
> Ketika hibah dibuat dengan `GranteeServicePrincipal` parameter, entri CloudTrail log untuk `CreateGrant` operasi menyertakan `granteeServicePrincipal` bidang, bukan`granteePrincipal`. Demikian pula, jika a `RetiringServicePrincipal` ditentukan, entri log menyertakan `retiringServicePrincipal` bidang, bukan`retiringPrincipal`. Ini membedakan hibah yang secara eksplisit dibuat dengan `GranteeServicePrincipal` untuk prinsipal AWS layanan dari hibah di mana AWS layanan diwakili di lapangan. `granteePrincipal`

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, lihatBuat kunci KMS.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-08-10T22:38:27Z",
    "eventSource": "kms.amazonaws.com",
```

```
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "description": "",
        "origin": "EXTERNAL",
        "bypassPolicyLockoutSafetyCheck": false,
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "keySpec": "SYMMETRIC_DEFAULT",
        "keyUsage": "ENCRYPT_DECRYPT"
    },
    "responseElements": {
        "keyMetadata": {
            "AWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Aug 10, 2022, 10:38:27 PM",
            "enabled": false,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "PendingImport",
            "origin": "EXTERNAL",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false
        }
    },
    "requestID": "1aef6713-0223-4ff7-9a6d-781360521930",
    "eventID": "36327b37-f4f6-40a9-92ab-48064ec905a2",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
```

```
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Contoh berikut menunjukkan CloudTrail log `CreateKey` operasi yang menciptakan kunci KMS enkripsi simetris di 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": "2021-10-14T17:39:50Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyUsage": "ENCRYPT_DECRYPT",
        "bypassPolicyLockoutSafetyCheck": false,
        "origin": "AWS_CLOUDHSM",
        "keySpec": "SYMMETRIC_DEFAULT",
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "customKeyStoreId": "cks-1234567890abcdef0",
        "description": ""
    },
    "responseElements": {
        "keyMetadata": {
            "aWSAccountId": "111122223333",
            "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
            "creationDate": "Oct 14, 2021, 5:39:50 PM",
            "enabled": true,
            "description": "",
```

```
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Enabled",
            "origin": "AWS_CLOUDHSM",
            "customKeyStoreId": "cks-1234567890abcdef0",
            "cloudHsmClusterId": "cluster-1a23b4cdefg",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false
        }
    },
    "additionalEventData": {
        "backingKey": "{\"backingKeyId\":\"backing-key-id\"}"
    },
    "requestID": "4f0b185c-588c-4767-9e90-c618f7e13cad",
    "eventID": "c73964b8-703d-49e4-bd9e-f773d0ee1e65",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

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 melalui koneksi TLS pasca-kuantum

Dekripsi dengan kunci enkripsi simetris standar

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

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:45:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
```

```
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

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 dan. `customKeyStoreId` `backingKeyId` Nilai yang dikembalikan di `backingKeyId` bidang adalah atribut kunci `id` CloudHSM. `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"
}
```

Dekripsi dengan kunci enkripsi simetris standar melalui koneksi TLS pasca-kuantum

Berikut ini adalah contoh entri CloudTrail log untuk `Decrypt` operasi dengan kunci enkripsi simetris standar melalui koneksi TLS pasca-kuantum. Bidang KeyExchange di `tlsDetails` bagian menyebutkan`X25519MLKEM768`. Ini adalah algoritma hibrida yang menggabungkan Elliptic Curve Diffie-Hellman (ECDH) melalui Curve25519, algoritma pertukaran kunci klasik yang digunakan saat ini di TLS, dengan Module-Lattice-Based Key-Encapsulation Mechanism (), enkripsi kunci publik dan algoritma pembentukan kunci yang telah ditetapkan oleh National Institute for Standards and Technology (NISTML-KEM) sebagai algoritma kesepakatan kunci pasca-kuantum standar pertamanya.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-11-12T15:16:26Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
```

```
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-sdk-java/2.30.22 md/io#async md/http#AwsCommonRuntime ...",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com",
        "keyExchange": "X25519MLKEM768"
    }
}
```

DeleteAlias

Contoh berikut menunjukkan entri AWS CloudTrail log untuk DeleteAliasoperasi. Untuk informasi tentang cara menghapus alias, lihat Hapus 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": "1.02",
    "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, lihatHapus 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 KMS mencatat 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 utama untuk AWS KMS keys.

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan saat AWS KMS menghapus materi kunci kedaluwarsa.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2025-05-22T19:55:11Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteExpiredKeyMaterial",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "cfa932fd-0d3a-4a76-a8b8-616863a2b547",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
```

```
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "eventCategory": "Management"
}
```

DeleteImportedKeyMaterial

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. Lihat perinciannya di Hapus materi kunci yang diimpor.

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan untuk `DeleteImportedKeyMaterial` operasi.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:45:08Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteImportedKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "responseElements": {
```

```
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "dcf0e82f-dad0-4622-a378-a5b964ad42c1",
    "eventID": "2afbb991-c668-4641-8a00-67d62e1fecbd",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```


DeleteKey

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. Hapus sebuah AWS KMS key

Contoh entri CloudTrail log berikut mencatat `DeleteKey` operasi kunci KMS dengan materi kunci di AWS KMS.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2020-07-31T00:07:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "b25f9cda-74e1-4458-847b-4972a0bf9668",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
     "managementEvent": true,
    "eventCategory": "Management"
}
```


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": "[{\"backingKeyId\":\"backing-key-id\"}]",
        "backingKeysDeletionStatus": "[{\"backingKeyId\":\"backing-key-id\",
\"deletionStatus\":\"SUCCESS\"}]"
    },
    "eventID": "1234585c-4b0c-4340-ab11-662414b79239",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
}
```


DescribeCustomKeyStores

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil `DescribeCustomKeyStores`operasi. Untuk informasi tentang melihat toko kunci khusus, lihatLihat 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 KMS merekam 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, lihat AWS KMS keys . AWS KMSAktifkan dan nonaktifkan 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, lihatPutar 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, lihatPutuskan 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, lihat AWS KMS keys .. AWS KMSAktifkan dan nonaktifkan 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, lihatPutar 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 AWS CloudTrail log untuk operasi Enkripsi.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Encrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "encryptionContext": {
            "Department": "Engineering"
        },
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "f3423043-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "91235988-eb87-476a-ac2c-0cdc244e6dca",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
```

```
}
```

GenerateDataKey

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GenerateDataKeyoperasi.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
```

```
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
}
```

GenerateDataKeyPair

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": "1.11",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
```

```
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

GenerateDataKeyPairWithoutPlaintext

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": "1.11",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
```

```
    "eventName": "GenerateDataKeyPairWithoutPlaintext",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_4096",
        "encryptionContext": {
            "Index": "5"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```


GenerateDataKeyWithoutPlaintext

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GenerateDataKeyWithoutPlaintextoperasi.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "d6b8e411-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "f7734272-9ec5-4c80-9f36-528ebbe35e4a",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
```

```
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
}
```

GenerateMac

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 menggunakan AWS 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"
}
```


GetKeyLastUsage

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GetKeyLastUsageoperasi. AWS KMS merekam entri seperti yang berikut saat Anda memanggil `GetKeyLastUsage` operasi, atau melihat detail kunci KMS di AWS KMS konsol.


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
```

```
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2026-04-27T17:11:06Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetKeyLastUsage",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "93746dd6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "4aa7e4d5-d047-452a-a5a6-2cce282a7e82",
    "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"
}
```


GetKeyPolicy

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GetKeyPolicyoperasi. Untuk informasi tentang melihat kebijakan kunci untuk kunci KMS, lihatMelihat kebijakan utama.


```
{
    "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. Putar 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 file AWS 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": "1.11",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2025-05-21T05:42:31Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ImportKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "validTo": "May 21, 2025, 5:47:45 AM",
        "expirationModel": "KEY_MATERIAL_EXPIRES",
        "importType": "NEW_KEY_MATERIAL",
        "keyMaterialDescription": "ExampleKeyMaterialA"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "89e10ee7-a612-414d-95a2-a128346969fd",
    "eventID": "c7abd205-a5a2-4430-bbfa-fc10f3e2d79f",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
```

```
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

ListAliases

Contoh berikut menunjukkan entri AWS CloudTrail log untuk ListAliasesoperasi. Karena operasi ini tidak menggunakan alias tertentu atau AWS KMS key, `resources` bidang kosong. Untuk informasi tentang melihat alias di AWS KMS, lihatTemukan nama alias dan alias ARN untuk kunci KMS.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:51:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListAliases",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "limit": 5,
        "marker":
 "eyJiIjoiYWxpYXMvZTU0Y2MxOTMtYTMwNC00YzEwLTliZWItYTJjZjA3NjA2OTJhIiwiYSI6ImFsaWFzL2U1NGNjMTkzL
    },
    "responseElements": null,
    "requestID": "bfe6c190-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a27dda7b-76f1-4ac3-8b40-42dfba77bcd6",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ListGrants

Contoh berikut menunjukkan entri AWS CloudTrail log untuk ListGrantoperasi. Untuk informasi tentang hibah di AWS KMS, lihatHibah 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. Putar AWS KMS keys


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-21T05:42:35Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListKeyRotations",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "includeKeyMaterial": "ALL_KEY_MATERIAL"
    },
    "responseElements": null,
    "requestID": "99c88d32-f2db-455e-8a9a-23855258a452",
    "eventID": "8ce0e74b-b9c7-45a2-96ef-83136d38068e",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
```

```
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
}
```

PutKeyPolicy

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil PutKeyPolicyoperasi. Untuk informasi tentang memperbarui kebijakan utama, lihatMengubah kebijakan utama.

```
{
    "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 dua AWS KMS keys, kunci KMS sumber dan kunci KMS tujuan, dalam urutan itu.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-22T19:34:55Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReEncrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "sourceEncryptionContext": {
            "Project": "Alpha",
            "Department": "Engineering"
        },
        "destinationKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "destinationEncryptionContext": {
```

```
            "Level": "3A"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "destinationKeyMaterialId":
 "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
        "sourceKeyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "03769fd4-acf9-4b33-adf3-2ab8ca73aadf",
    "eventID": "542d9e04-0e8d-4e05-bf4b-4bdeb032e6ec",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```


ReplicateKey

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 Buat 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. Putar 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, lihatPutar AWS KMS keys.

Untuk contoh entri CloudTrail log yang mencatat `EnableKeyRotation` operasi, lihatEnableKeyRotation.


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2025-05-20T20:44:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RotateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "rotationType": "AUTOMATIC",
        "keyOrigin": "AWS_KMS",
```

```
        "previousKeyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
        "currentKeyMaterialId":
 "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "eventCategory": "Management"
}
```

Contoh berikut menunjukkan entri CloudTrail log untuk rotasi sesuai permintaan yang diprakarsai oleh operasi. RotateKeyOnDemand Untuk informasi tentang memutar kunci KMS enkripsi simetris sesuai permintaan, lihat. Lakukan rotasi kunci sesuai permintaan

Untuk contoh entri CloudTrail log yang mencatat `RotateKeyOnDemand` operasi, lihatRotateKeyOnDemand.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2025-05-20T20:44:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RotateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
```

```
    "serviceEventDetails": {
        "rotationType": "ON_DEMAND",
        "keyOrigin": "EXTERNAL",
        "previousKeyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
        "currentKeyMaterialId":
 "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "eventCategory": "Management"
}
```

RotateKeyOnDemand

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. Lakukan 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 menghapus AWS KMS keys, lihatHapus sebuah AWS KMS key.

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": "[{\"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 AWS CloudTrail log yang dihasilkan saat AWS KMS menyinkronkan kunci Multi-region. Sinkronisasi melibatkan panggilan lintas wilayah untuk menyalin properti bersama dari kunci utama Multi-wilayah ke kunci replika. AWS KMS menyinkronkan kunci Multi-region secara berkala untuk memastikan bahwa semua kunci Multi-region 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 penandaan AWS KMS keys, lihatTag di AWS KMS.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-01T21:19:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "tags": [
            {
                "tagKey": "Department",
                "tagValue": "IT"
            }
```

```
        ]
    },
    "responseElements": null,
    "requestID": "b942584a-f77d-4787-9feb-b9c5be6e746d",
    "eventID": "0a091b9b-0df5-4cf9-b667-6f2879532b8f",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

UntagResource

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 penandaan AWS KMS keys, lihatTag di AWS KMS.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-01T21:19:19Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UntagResource",
```

```
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "tagKeys": [
            "Dept"
        ]
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "cb1d507b-6015-47f4-812b-179713af8068",
    "eventID": "0b00f4b0-036e-411d-aa75-87eb4a35a4b3",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

UpdateAlias

Contoh berikut menunjukkan entri AWS CloudTrail log untuk UpdateAliasoperasi. `resources`Elemen ini mencakup bidang untuk alias dan sumber daya kunci KMS. Untuk informasi tentang membuat alias di AWS KMS, lihatBuat 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, lihatEdit pengaturan penyimpanan 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-west-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"
}
```

Amazon EC2 contoh satu

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 file EC2 log mencakup `volumeId` bidang dengan nilai`"vol-13439757"`. AWS KMS Catatan berisi `encryptionContext` bidang 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"
    }
  ]
}
```

Amazon EC2 contoh dua

Dalam contoh berikut, prinsipal IAM yang menjalankan EC2 instance Amazon membuat dan memasang volume data yang dienkripsi di bawah kunci KMS. Tindakan ini menghasilkan beberapa catatan CloudTrail log. Untuk informasi selengkapnya tentang Amazon EBS dan enkripsi, lihat Persyaratan enkripsi Amazon EBS.

Saat volume dibuat, Amazon EC2, bertindak atas nama pelanggan, mendapatkan kunci data terenkripsi dari AWS KMS ()`GenerateDataKeyWithoutPlaintext`. Kemudian, volume membuat izin (`CreateGrant`) yang memungkinkannya untuk mendekripsi kunci data. Saat volume dipasang, Amazon EC2 memanggil AWS KMS untuk mendekripsi kunci data ()`Decrypt`.

EC2 Contoh Amazon,`"i-81e2f56c"`, muncul di `RunInstances` acara tersebut. `instanceId` 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"
    }
  ]
}
```

Pantau tombol KMS dengan Amazon CloudWatch

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.

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 Konsol Manajemen AWS 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.

Anda dapat melihat AWS KMS metrik menggunakan CloudWatch API Amazon Konsol Manajemen AWS dan Amazon. Untuk informasi selengkapnya, lihat Melihat metrik yang tersedia di Panduan CloudWatch Pengguna Amazon.

Topik

- SuccessfulRequest
- SecondsUntilKeyMaterialExpiration
- Awan HSMKey StoreThrottle
- ExternalKeyStoreThrottle
- XksProxyCertificateDaysToExpire
- XksProxyCredentialAge
- XksProxyErrors
- XksExternalKeyManagerStates
- XksProxyLatency

SuccessfulRequest

Jumlah permintaan yang berhasil untuk operasi kriptografi pada kunci KMS tertentu. Dengan menggunakan `SuccessfulRequest` metrik, Anda dapat menerapkan pemfilteran tingkat kunci ke penggunaan AWS KMS API di. CloudWatch `Sum`Statistik untuk metrik ini mendefinisikan jumlah total permintaan yang berhasil selama periode tersebut.

Gunakan metrik ini untuk mengidentifikasi kunci KMS mana yang menggunakan porsi terbesar dari kuota permintaan Anda atau berkontribusi paling banyak terhadap biaya API. Anda juga dapat membuat CloudWatch alarm berdasarkan `SuccesfulRequest` metrik untuk memberi tahu Anda tentang pola penggunaan AWS KMS API yang tidak normal. Peringatan ini dapat membantu mengidentifikasi alur kerja yang tidak efisien yang mungkin secara tidak sengaja melebihi kuota permintaan Anda atau menimbulkan biaya tak terduga.

Dimensi untuk **`SuccessfulRequest`**

Dimensi	Deskripsi
KeyArn	Nilai untuk setiap kunci KMS.

Dimensi	Deskripsi
Operasi	Nilai untuk setiap operasi AWS KMS API. Metrik ini hanya berlaku untuk operasi kriptografi.

Untuk ReEncryptoperasi, `SuccessfulRequest` metrik mencakup dimensi untuk kunci KMS sumber dan tujuan.

Dimensi	Deskripsi
SourceKeyArn	Nilai untuk kunci KMS yang mendekripsi ciphertext.
DestinationKeyArn	Nilai untuk kunci KMS yang mengenkripsi ulang data.
Operasi	Nilai untuk setiap operasi AWS KMS API, dalam hal ini, ReEncrypt.

SecondsUntilKeyMaterialExpiration

Jumlah detik yang tersisa hingga bahan kunci impor yang paling awal kedaluwarsa dalam kunci KMS. Metrik ini hanya berlaku untuk kunci KMS dengan bahan kunci impor (asal bahan utama`EXTERNAL`) dan tanggal kedaluwarsa.

Gunakan metrik ini untuk melacak berapa banyak waktu yang tersisa sebelum materi kunci impor Anda yang paling awal kedaluwarsa berakhir. Ketika waktu itu jatuh di bawah ambang batas yang Anda tentukan, Anda harus mengimpor ulang materi kunci dengan tanggal kedaluwarsa baru agar kunci KMS tetap dapat digunakan. `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, lihatBuat 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.

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 Buat alarm yang mendeteksi penggunaan kunci KMS tertunda penghapusan.

Awan HSMKey StoreThrottle

Jumlah permintaan untuk operasi kriptografi pada kunci KMS di setiap penyimpanan AWS CloudHSM kunci yang AWS KMS melambat (merespons dengan a). `ThrottlingException` Metrik ini hanya berlaku untuk toko AWS CloudHSM utama.

`CloudHSMKeyStoreThrottle`Metrik hanya berlaku untuk kunci KMS di penyimpanan AWS CloudHSM kunci dan hanya untuk permintaan operasi kriptografi. AWS KMS membatasi permintaan ini ketika tingkat permintaan melebihi kuota permintaan toko kunci kustom untuk toko kunci Anda AWS CloudHSM . Metrik ini juga mencakup pelambatan oleh cluster. AWS CloudHSM

Nama grup dimensi: Keystore Throttle Metrics

Dimensi	Deskripsi
CustomKey StoreId	Nilai untuk setiap toko AWS CloudHSM kunci.
KmsOperation	Nilai untuk setiap operasi AWS KMS API. Metrik ini hanya berlaku untuk operasi kriptografi pada kunci KMS di toko AWS CloudHSM kunci.
KeySpec	Nilai untuk setiap jenis kunci KMS. Satu-satunya spesifikasi kunci yang didukung untuk kunci KMS di toko AWS CloudHSM kunci adalah SYMMETRIC_DEFAULT.

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. 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 Dukungan 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 pada kunci KMS di penyimpanan kunci eksternal.

Dimensi	Deskripsi
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.

Anda dapat membuat CloudWatch alarm berdasarkan metrik untuk penyimpanan kunci eksternal dan kunci KMS di toko kunci eksternal. Untuk petunjuk, lihat Pantau toko kunci eksternal.

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.

Anda dapat membuat CloudWatch alarm berdasarkan metrik untuk penyimpanan kunci eksternal dan kunci KMS di toko kunci eksternal. Untuk petunjuk, lihat Pantau toko 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, lihatLihat toko kunci eksternal. Untuk mengedit pengaturan penyimpanan kunci eksternal Anda, lihatEdit 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 berhasil; tidak ada kesalahan
ExceptionName	Nilai: • Nama pengecualian • Tidak ada: Permintaan berhasil; tidak ada kesalahan

Anda dapat membuat CloudWatch alarm berdasarkan metrik untuk penyimpanan kunci eksternal dan kunci KMS di toko kunci eksternal. Untuk petunjuk, lihat Pantau toko kunci eksternal.

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.

Anda dapat membuat CloudWatch alarm berdasarkan metrik untuk penyimpanan kunci eksternal dan kunci KMS di toko kunci eksternal. Untuk petunjuk, lihat Pantau toko kunci eksternal.

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 habis 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.

Anda dapat membuat CloudWatch alarm berdasarkan metrik untuk penyimpanan kunci eksternal dan kunci KMS di toko kunci eksternal. Untuk petunjuk, lihat Pantau toko kunci eksternal.

Buat 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.
- 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

Bidang	Nilai
Jenis ambang	Statis
Setiap kali...	Setiap kali *metric-name* lebih besar dari 1

Buat CloudWatch alarm untuk penyimpanan kunci eksternal

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 Amazon EC2 Auto Scaling. 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

- Buat alarm untuk kedaluwarsa sertifikat
- Buat alarm untuk batas waktu respons
- Buat alarm untuk kesalahan yang dapat dicoba ulang
- Buat alarm untuk kesalahan yang tidak dapat dicoba ulang

Buat alarm 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
Setiap kali...	Kapan pun XksProxyCertificateDaysToExpire`Lower`lebih dari`10`.

Buat alarm 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 KMS mengharapkan 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
Setiap kali...	Kapan pun XksProxyLatency`Greater`lebih dari`200`.

Buat alarm 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)` nama Metrik.

Bidang	Nilai
	Masukkan `ErrorType = Retryable` untuk Filter oleh. Pilih Jalankan. Kemudian pilih Pilih metrik.
Label	*Retryable errors*
Periode	1 menit
Jenis ambang	Statis
Setiap kali...	Setiap kali q1 adalah `Greater` dari5.

Buat alarm 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)` nama Metrik. Masukkan `ErrorType = Non-retryable` untuk Filter oleh.

Bidang	Nilai
	Pilih Jalankan. Kemudian pilih Pilih metrik.
Label	*Non-retryable errors*
Periode	1 menit
Jenis ambang	Statis
Setiap kali...	Setiap kali q1 adalah `Greater` dari5.

Menentukan penggunaan kunci KMS sebelumnya

Untuk tujuan pemantauan dan audit, Anda mungkin ingin mengetahui bagaimana kunci KMS telah digunakan di masa lalu. Misalnya, Anda mungkin ingin menentukan apakah kunci KMS masih digunakan secara aktif sebelum menonaktifkan atau menjadwalkannya untuk dihapus, atau mengidentifikasi kunci yang tidak digunakan di akun Anda. Strategi berikut dapat membantu Anda menentukan penggunaan kunci KMS di masa lalu.

> **Warning**
>
> Strategi ini untuk menentukan penggunaan masa lalu hanya efektif untuk prinsi AWS pal dan AWS KMS operasi. Mereka tidak dapat mendeteksi penggunaan kunci yang tidak melibatkan panggilan API ke AWS KMS. Setelah kunci data dihasilkan dari kunci KMS simetris, penggunaan selanjutnya untuk enkripsi lokal atau dekripsi di luar tidak AWS KMS tercermin dalam informasi penggunaan terakhir. Demikian pula, strategi 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. Deleting asymmetric KMS keys

Topik

- Periksa izin kunci KMS untuk menentukan ruang lingkup penggunaan potensial
- Periksa operasi kriptografi terakhir yang dilakukan dengan kunci KMS
- Periksa AWS CloudTrail log untuk memeriksa penggunaan masa lalu

Periksa 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 kunci tersebut masih diperlukan. Untuk menentukan siapa atau apa yang saat ini memiliki akses ke kunci KMS, lihatMenentukan akses ke AWS KMS keys.

Periksa operasi kriptografi terakhir yang dilakukan dengan kunci KMS

AWS KMS memberikan informasi penggunaan pada operasi kriptografi terakhir yang berhasil dilakukan dengan setiap kunci KMS bersama dengan ID CloudTrail peristiwa terkait. Ini dapat memudahkan proses mengidentifikasi kunci KMS yang tidak digunakan. Anda juga dapat menggunakan kunci TrailingDaysWithoutKeyUsage kondisi kms: dalam kebijakan utama untuk mencegah kunci yang baru saja digunakan dari penonaktifan atau penjadwalan penghapusan secara tidak sengaja.

Anda dapat melihat operasi kriptografi terakhir yang berhasil dilakukan dengan kunci KMS menggunakan Konsol Manajemen AWS, AWS CLI, atau AWS KMS API.

> **Note**
>
> AWS Layanan tertentu membuat sumber daya yang bergantung pada kunci KMS untuk perlindungan data tetapi tidak sering memanggil operasi kriptografi pada kunci tersebut. Misalnya, layanan Amazon EC2 memanggil AWS KMS untuk mendekripsi kunci data untuk volume Amazon EBS terenkripsi hanya ketika volume dilampirkan ke instans. Dalam kasus ini, Anda tidak boleh mengandalkan informasi penggunaan terakhir saja untuk menentukan apakah kunci KMS dapat dihapus. Jika kunci KMS yang melindungi volume Amazon EBS dihapus, tidak akan ada gangguan pada volume Amazon EBS yang sudah terpasang, tetapi upaya selanjutnya untuk melampirkan volume Amazon EBS terenkripsi tersebut ke instans Amazon EC2 lain akan gagal.

Memahami periode pelacakan penggunaan

AWS KMS melacak hanya operasi kriptografi terakhir yang berhasil dilakukan dengan setiap kunci KMS. Mungkin ada penundaan hingga satu jam antara waktu operasi kriptografi terjadi dan waktu penggunaan dicatat.

Saat Anda memeriksa informasi penggunaan terakhir untuk kunci KMS, respons menyertakan tanggal mulai pelacakan. Ini `TrackingStartDate` adalah tanggal dari mana AWS KMS mulai merekam aktivitas kriptografi untuk kunci itu. Gunakan tanggal ini bersama dengan tanggal pembuatan kunci untuk menentukan riwayat penggunaannya dengan membandingkan tanggal pembuatan kunci dengan tanggal mulai pelacakan:

- Jika informasi penggunaan terakhir ada, kunci telah digunakan untuk operasi kriptografi sejak pelacakan dimulai. Respons mencakup jenis operasi, stempel waktu, dan ID AWS CloudTrail peristiwa terkait.
- Jika informasi penggunaan terakhir kosong, kunci tidak memiliki operasi kriptografi yang direkam sejak pelacakan dimulai. Bandingkan tanggal pembuatan kunci dengan `TrackingStartDate` untuk menentukan apa artinya ini:
  - Jika kunci dibuat pada atau setelah `TrackingStartDate`, kunci tersebut belum digunakan untuk operasi kriptografi sejak dibuat.
  - Jika kunci dibuat sebelum `TrackingStartDate`, tidak ada catatan kunci yang digunakan sejak pelacakan dimulai. Namun, kunci mungkin telah digunakan sebelum pelacakan dimulai. Untuk menentukan apakah kunci telah digunakan sebelumnya, periksa AWS CloudTrail log masa lalu Anda.

> **Warning**
>
> Jangan hanya mengandalkan informasi penggunaan terakhir saat menghapus kunci yang tidak digunakan. Sebagai gantinya, non aktifkan kunci terlebih dahulu dan AWS CloudTrail pan `DisabledException` tau entri, yang menunjukkan upaya untuk menggunakan kunci saat dinonaktifkan. Ini membantu mengidentifikasi potensi dependensi dan kegagalan beban kerja. AWS CloudTrail tetap menjadi sumber otoritatif untuk semua panggilan API yang dilakukan ke kunci Anda.

Operasi kriptografi yang dilacak

Hanya operasi kriptografi yang berhasil berikut yang dilacak dan dicatat untuk melaporkan informasi penggunaan terakhir. Non-cryptographic Operasi dikecualikan.

- `Decrypt`
- `DeriveSharedSecret`

- `Encrypt`
- `GenerateDataKey`
- `GenerateDataKeyPair`
- `GenerateDataKeyPairWithoutPlaintext`
- `GenerateDataKeyWithoutPlaintext`
- `GenerateMac`
- `ReEncrypt`
- `Sign`
- `Verify`
- `VerifyMac`

Pertimbangan lainnya

Ingatlah hal-hal berikut saat Anda menggunakan informasi penggunaan:

- Multi-Region Kunci KMS — Untuk kunci KMS multi-wilayah, kunci primer dan replika melacak informasi penggunaan terakhir secara independen. Setiap kunci dalam kumpulan kunci Multi-wilayah mempertahankan informasi penggunaan terakhirnya sendiri.
- ReEncrypt Oper asi — `ReEncrypt` Operasi menggunakan dua kunci: kunci sumber untuk dekripsi dan kunci tujuan untuk enkripsi. Informasi penggunaan terakhir dicatat untuk kedua kunci secara independen, masing-masing dengan ID CloudTrail peristiwa dari akun pemilik kunci masing-masing.

Anda dapat melihat informasi penggunaan terakhir menggunakan metode berikut:

Penggunaan AWS KMS konsol

Anda dapat melihat operasi kriptografi terakhir yang berhasil dilakukan dengan kunci KMS pada halaman detail untuk setiap kunci KMS. Untuk prosedur tentang cara melihat halaman detail untuk kunci KMS, lihatAkses dan daftar rincian kunci KMS.

Penggunaan AWS KMS API

GetKeyLastUsageOperasi mengembalikan informasi penggunaan pada operasi kriptografi terakhir yang dilakukan dengan kunci KMS yang ditentukan. Untuk mengidentifikasi kunci KMS, gunakan ID kunci atau kunci ARN.

Misalnya, panggilan berikut untuk mengambil `GetKeyLastUsage` informasi penggunaan tentang kunci KMS dengan ID kunci. *1234abcd-12ab-34cd-56ef-1234567890ab*

```
$ aws kms get-key-last-usage --key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
{
    "KeyCreationDate": 1773253425.56,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "TrackingStartDate": 1773253425.56,
    "KeyLastUsage": {
        "Timestamp": 1773253497.0,
        "Operation": "Encrypt",
        "KmsRequestId": "040cce3e-9ef3-4651-b8cf-e47c9bafdc9b",
        "CloudTrailEventId": "2cfd5892-ea8c-4342-ad49-4b9594b06a8b"
    }
}
```

Sebaliknya, panggilan berikut untuk `GetKeyLastUsage` mengungkapkan tidak ada informasi penggunaan untuk kunci KMS dengan ID *0987dcba-09fe-87dc-65ba-ab0987654321* kunci.

```
$ aws kms get-key-last-usage --key-id "0987dcba-09fe-87dc-65ba-ab0987654321"
{
    "KeyCreationDate": 1672531200.0,
    "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "TrackingStartDate": 1773253425.56,
    "KeyLastUsage": {}
}
```

Periksa AWS CloudTrail log untuk memeriksa penggunaan masa lalu

Anda dapat menggunakan riwayat penggunaan kunci KMS untuk membantu menentukan apakah Anda memiliki ciphertext yang dienkripsi di bawah kunci KMS tertentu.

Semua aktivitas AWS KMS API dicatat dalam file AWS CloudTrail log. Jika Anda telah membuat CloudTrail jejak di wilayah tempat kunci KMS Anda 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 di riwayat CloudTrail acara Anda. Untuk detail tentang cara AWS KMS penggunaan CloudTrail, lihatPencatatan log AWS KMS Panggilan 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 Mel indungi 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 GenerateDataKey operasi 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 `Decrypt` permintaan AWS KMS untuk mendekripsi kunci data objek 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"
}
```

Dengan mengevaluasi entri log ini, Anda mungkin dapat menentukan penggunaan kunci KMS tertentu sebelumnya, dan ini mungkin 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, bukaPencatatan log AWS KMS Panggilan API dengan AWS CloudTrail. Untuk informasi selengkap CloudTrail nya, buka Panduan AWS CloudTrail Pengguna.

Pertimbangan biaya untuk CloudTrail kueri

Saat Anda menanyakan CloudTrail log menggunakan Amazon Log CloudWatch s Insights, Amazon CloudWatch Logs Insights menagih Anda berdasarkan jumlah data yang dipindai. Kueri besar dan tidak terbatas dapat mengakibatkan biaya tinggi yang tak terduga. Untuk mengelola biaya, lingkup setiap kueri ke rentang waktu relevan yang paling sempit. Filter berdasarkan nama acara, ARN sumber daya, atau ID utama. Sebelum mencoba melakukan query, pertimbangkan untuk memeriksa operasi kriptografi terakhir yang berhasil dilakukan dengan kunci KMS terlebih dahulu untuk menghindari menimbulkan biaya kueri yang tidak perlu. Anda juga dapat mengatur peringatan AWS anggaran di Anggaran untuk menangkap volume kueri yang tidak terduga. Untuk informasi selengkapnya tentang harga, lihat CloudWatch Harga Amazon di situs web Amazon Web Services.

Pantau tombol KMS dengan Amazon EventBridge

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 atau sesuai permintaan.
- Materi kunci yang diimpor dalam kunci KMS kedaluwarsa.
- Kunci KMS yang telah dijadwalkan untuk dihapus telah dihapus.

AWS KMS terintegrasi 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 read/write API, lihat EventBridge Panduan Pengguna Amazon.

Topik berikut menjelaskan EventBridge peristiwa yang AWS KMS dihasilkan.

Rotasi CMK KMS

AWS KMS mendukung rotasi otomatis dan sesuai permintaan dari bahan utama dalam kunci KMS enkripsi simetris.

Setiap kali AWS KMS memutar materi kunci, ia mengirimkan `KMS CMK Rotation` acara ke EventBridge. AWS KMS menghasilkan 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": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
    "key-origin": "AWS_KMS",
    "rotation-type": "ON_DEMAND",
    "previous-key-material-id":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
    "current-key-material-id":
 "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068"
  }
}
```

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 KMS menghasilkan 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": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "key-material-id":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  }
}
```

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 kirimkan `KMS CMK Deletion` acara ke. EventBridge AWS KMS menjamin 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": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Alias di AWS KMS

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. 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 merupakan bagian dari AWS KMS dukungan untuk kontrol akses berbasis atribut (ABAC). Lihat perinciannya di Gunakan 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. Sebagai contoh, lihat Pelajari cara menggunakan alias dalam aplikasi.

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

AWS KMS API 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

Cara kerja alias

Pelajari cara kerja alias di AWS KMS.

Alias adalah sumber daya independen AWS

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 Anda membuat alias, Anda menentukan nama alias. AWS KMS membuat alias ARN 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. Di AWS KMS API, nama alias selalu diawali oleh. `alias/` Awalan itu dihilangkan di 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. Lihat perinciannya di Pelajari cara menggunakan alias dalam aplikasi.

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

AWS membuat 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 AWS KMS API, lihat deskripsi `KeyId` parameter dalam Referensi AWS Key Management Service API.

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 Lihat perinciannya di ABAC untuk AWS KMS.

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 Gunakan alias untuk mengontrol akses ke tombol KMS.

Untuk informasi tentang mengontrol akses ke semua AWS KMS operasi, lihatReferensi 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.

JSON

```
{
```

```
  "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 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. Misalnya, pernyataan kebijakan berikut memungkinkan 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*"
}
```

Buat alias

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

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.

Anda juga dapat menggunakan AWS CloudFormation template untuk membuat alias untuk kunci KMS. Untuk informasi selengkapnya, lihat AWS: :KMS: :Alias di Panduan Pengguna.AWS CloudFormation

Menggunakan AWS KMS 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 Konsol Manajemen AWS 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 atau Kunci 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`.

Menggunakan AWS KMS API

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.

Dalam operasi AWS KMS API, nama alias harus dimulai dengan `alias/` diikuti dengan nama, sepertialias/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 sama Wilayah 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 Gunakan `CreateAlias` dengan AWS SDK atau CLI.

```
$ 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`. Lihat perinciannya di Menggunakan AWS KMS API.

Temukan nama alias dan alias ARN untuk kunci KMS

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.

Prosedur berikut menunjukkan cara melihat dan mengidentifikasi alias yang terkait dengan kunci KMS menggunakan AWS KMS konsol dan AWS KMS API. Contoh AWS KMS API menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Menggunakan AWS KMS konsol

AWS KMS Konsol menampilkan alias yang terkait dengan kunci KMS.

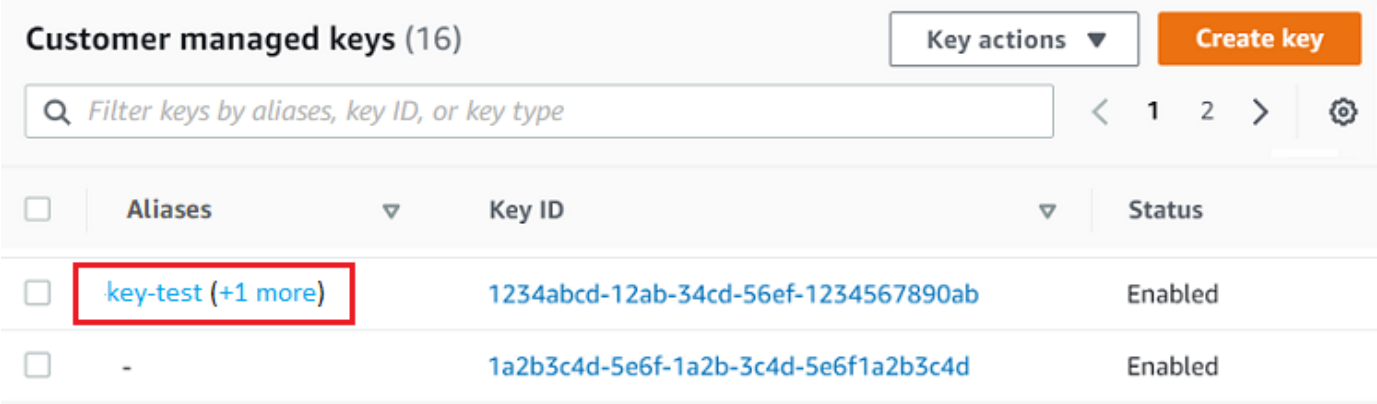
1. Buka AWS KMS konsol di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih CMK. Untuk melihat kunci di akun Anda yang AWS membuat dan mengelola untuk Anda, di panel navigasi, pilih kunci AWS terkelola.

4. Kolom Alias menampilkan alias untuk setiap tombol KMS. Jika kunci KMS tidak memiliki alias, tanda hubung (-) muncul di kolom Alias.

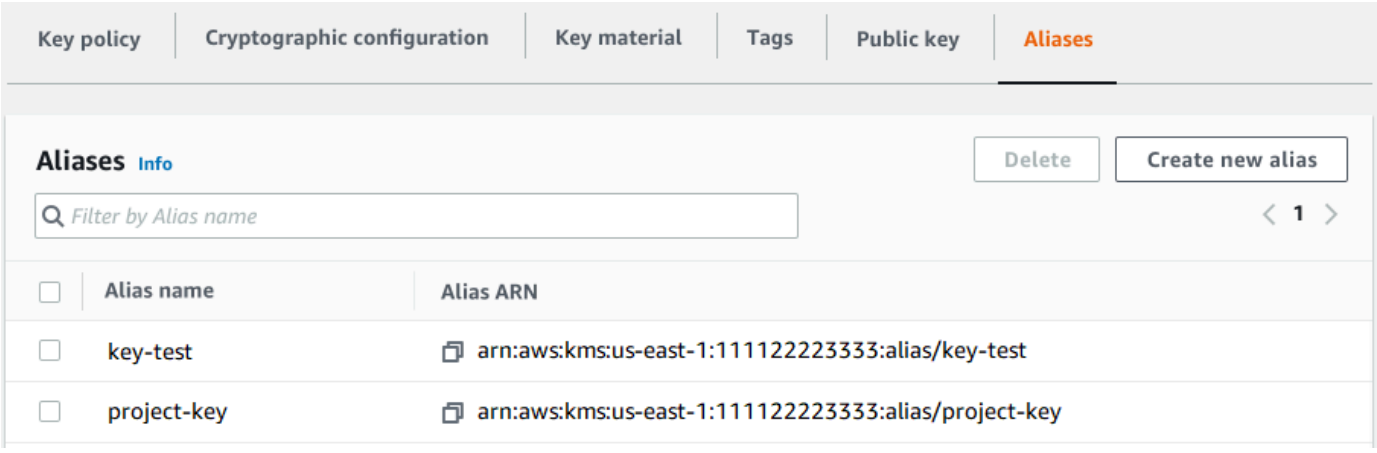
   Jika kunci KMS memiliki beberapa alias, kolom Alias juga memiliki ringkasan alias, seperti (+ n lebih). Misalnya, kunci KMS berikut memiliki dua alias, salah satunya adalah. `key-test`

   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.

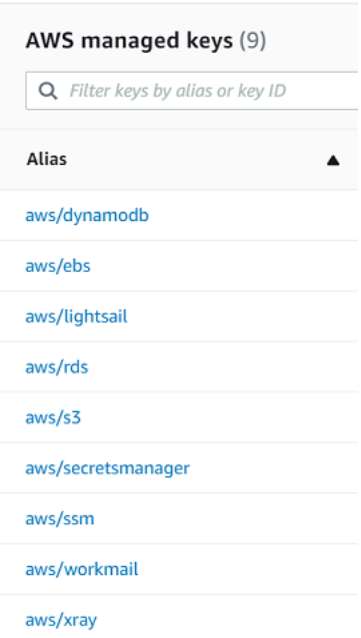


5. Tab Alias menampilkan nama alias dan alias ARN dari semua alias untuk kunci KMS. Anda juga dapat membuat dan menghapus alias untuk tombol KMS pada tab ini.



Kunci yang dikelola AWS

Anda dapat menggunakan alias untuk mengenali Kunci 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`



Menggunakan AWS KMS API

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 formatdari`aws/<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 untuk Kunci 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"
    }
]
```

Perbarui 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 AWS KMS konsol. 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. AWS CLI Perintah ini menggunakan `--query` parameter untuk mendapatkan hanya `test-key` alias. 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
    }
]
```

Hapus 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.

Menggunakan AWS KMS 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 Konsol Manajemen AWS 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 atau Kunci 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.

Menggunakan AWS KMS API

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 `--query` parameter dalam AWS CLI untuk mendapatkan hanya `test-key` alias. 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`]'
[]
```

Gunakan 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 merupakan bagian dari AWS KMS dukungan 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 kepada 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 kepala sekolah di Anda Akun AWS 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.

> **Important**
>
> Pernyataan `Deny` kebijakan dengan `kms:RequestAlias` kondisi tidak disarankan karena penelepon dapat memintasnya dengan menggunakan ID kunci, ARN kunci, atau alias lain untuk mengidentifikasi kunci KMS dalam permintaan. Untuk menolak akses berdasarkan alias yang terkait dengan kunci KMS, gunakan kunci ResourceAliases kondisi kms: sebagai gantinya.

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, sepertialias/test*, sehingga Anda dapat menggunakannya dalam kebijakan IAM untuk mengontrol akses ke beberapa kunci KMS di Wilayah yang sama. Ini berlaku untuk AWS KMS operasi apa pun yang menggunakan kunci KMS.

Misalnya, kebijakan IAM berikut memungkinkan prinsipal memanggil operasi yang ditentukan pada kunci KMS menjadi dua. Akun AWS Namun, izin hanya berlaku untuk kunci KMS yang terkait dengan alias yang dimulai dengan. restricted

JSON

```
{
```

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

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"
```

Pelajari cara menggunakan alias dalam aplikasi

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 berbeda Akun 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 Temukan nama alias dan alias ARN untuk kunci KMS.

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.

Temukan alias di AWS CloudTrail log

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 ARN kunci 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, lihatPencatatan log AWS KMS Panggilan API dengan AWS CloudTrail.

Tag di AWS KMS

Tag adalah label metadata opsional yang dapat Anda tetapkan (atau AWS dapat ditetapkan) 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 tag pada sumber daya harus memiliki kunci tag yang berbeda, tetapi Anda dapat menambahkan tag yang sama ke beberapa AWS sumber daya. 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 orang Layanan AWS, termasuk penagihan.

Di AWS 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:

- Identifikasi dan atur AWS sumber daya Anda. Banyak AWS layanan mendukung penandaan, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya 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.
- Lacak AWS biaya Anda. Saat menambahkan tag ke AWS sumber daya, buat AWS laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan berdasarkan tag. Anda dapat menggunakan fitur ini untuk melacak AWS KMS biaya 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 Billing Panduan Pengguna.

- Kontrol akses ke AWS sumber daya 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, lihatGunakan tag untuk mengontrol akses ke tombol KMS. Untuk informasi umum selengkapnya tentang penggunaan tag untuk mengontrol akses ke AWS sumber daya, lihat Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di Panduan Pengguna IAM.

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

Topik

- Pengontrolan akses ke tanda
- Tambahkan tag ke kunci KMS
- Edit tag yang terkait dengan kunci KMS
- Hapus tag yang terkait dengan kunci KMS
- Lihat tag yang terkait dengan kunci KMS
- Gunakan tag untuk mengontrol akses ke tombol KMS

Pengontrolan akses ke tanda

Untuk menambah, melihat, dan menghapus tag, baik di AWS KMS konsol atau dengan menggunakan API, prinsipal memerlukan 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 AWS global untuk tag. Dalam AWS 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 Gunakan 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.

JSON

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

```
    {
      "Sid": "EnableIAMUserPermissions",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "AllowAllTaggingPermissions",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:user/LeadAdmin",
        "arn:aws:iam::111122223333:user/SupportLead"
      ]},
      "Action": [
          "kms:TagResource",
          "kms:ListResourceTags",
          "kms:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowRolesViewTags",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:role/Administrator",
        "arn:aws:iam::111122223333:role/Developer"
      ]},
      "Action": "kms:ListResourceTags",
      "Resource": "*"
    }
  ]
}
```

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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKeys",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyTags",
      "Effect": "Allow",
      "Action": [
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ListResourceTags"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    }
  ]
}
```

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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKey",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyViewAllTags",
      "Effect": "Allow",
      "Action": "kms:ListResourceTags",
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
```

```
      "Sid": "IAMPolicyManageTags",
      "Effect": "Allow",
      "Action": [
        "kms:TagResource",
        "kms:UntagResource"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
          "ForAllValues:StringEquals": {"aws:TagKeys": "Project"}
      }
    }
  ]
}
```

Tambahkan tag ke kunci KMS

Tag membantu mengidentifikasi dan mengatur AWS sumber daya Anda. Anda dapat menambahkan tag ke kunci yang dikelola pelanggan saat membuat kunci KMS, atau menambahkan tag ke kunci KMS yang ada. Anda tidak dapat menandai Kunci yang dikelola AWS.

Prosedur berikut menunjukkan cara menambahkan tag ke kunci yang dikelola pelanggan menggunakan AWS KMS konsol dan AWS KMS API. Contoh AWS KMS API menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Topik

- Tambahkan tag saat membuat kunci KMS
- Tambahkan tag ke kunci KMS yang ada

Tambahkan tag saat membuat kunci KMS

Anda dapat menambahkan tag ke kunci KMS saat Anda membuat kunci menggunakan AWS KMS konsol atau CreateKeyoperasi. Untuk menambahkan tag saat membuat kunci KMS, Anda harus memiliki `kms:TagResource` izin dalam kebijakan IAM selain izin yang diperlukan untuk membuat kunci KMS. Minimal, izin harus mencakup semua kunci KMS di akun dan Wilayah. Lihat perinciannya di Pengontrolan akses ke tanda.

Menggunakan AWS KMS konsol

Untuk menambahkan tag saat membuat kunci KMS di konsol, Anda harus memiliki izin yang diperlukan untuk melihat kunci KMS di konsol selain izin yang diperlukan untuk menandai dan membuat kunci KMS. Minimal, izin harus mencakup semua kunci KMS di akun dan Wilayah.

1. Masuk ke Konsol Manajemen AWS 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 dari Kunci 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.

Menggunakan AWS KMS API

Untuk menentukan tag saat membuat kunci menggunakan CreateKeyoperasi, gunakan `Tags` parameter operasi.

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.

Tambahkan tag ke kunci KMS yang ada

Anda dapat menambahkan tag ke kunci KMS yang dikelola pelanggan yang ada di AWS KMS konsol atau dengan menggunakan TagResourceoperasi. Untuk menambahkan tag, 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.

Menggunakan AWS KMS konsol

1. Masuk ke Konsol Manajemen AWS 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 dari Kunci yang dikelola AWS)
4. Anda dapat menggunakan filter tabel untuk menampilkan hanya tombol KMS dengan tag tertentu. Untuk detailnya, lihat Melihat tag menggunakan AWS KMS konsol.
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.
8. Untuk menyimpan perubahan Anda, memilih Simpan perubahan.

Menggunakan AWS KMS API

TagResourceOperasi menambahkan satu atau lebih tag ke kunci KMS. Anda tidak dapat menggunakan operasi ini untuk menambahkan tag yang berbeda Akun AWS. Anda juga dapat menggunakan TagResource operasi untuk mengedit tag yang ada. Untuk informasi selengkapnya, lihat the section called “Edit tag”.

Untuk menambahkan tanda, tentukan kunci tanda baru dan nilai tanda. 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.

Edit tag yang terkait dengan kunci KMS

Tag membantu mengidentifikasi dan mengatur AWS sumber daya Anda. Anda dapat mengedit tag yang terkait dengan kunci KMS yang dikelola pelanggan Anda di AWS KMS konsol atau dengan menggunakan TagResourceoperasi. Anda tidak dapat mengedit tag dari file Kunci yang dikelola AWS.

Prosedur berikut menunjukkan cara mengedit tag yang terkait dengan kunci KMS. Contoh AWS KMS API menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Menggunakan AWS KMS konsol

1. Masuk ke Konsol Manajemen AWS 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 mengedit tag dari Kunci yang dikelola AWS)
4. Anda dapat menggunakan filter tabel untuk menampilkan hanya tombol KMS dengan tag tertentu. Untuk detailnya, lihat Melihat tag menggunakan AWS KMS konsol.
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 mengubah nama atau nilai tanda, pilih Edit, buat perubahan, lalu pilih Simpan.

8. Untuk menyimpan perubahan Anda, memilih Simpan perubahan.

Menggunakan AWS KMS API

TagResourceOperasi menambahkan satu atau beberapa tag ke kunci yang dikelola pelanggan;. Namun, Anda juga dapat menggunakan TagResourceuntuk mengubah nilai tag tag yang ada. Anda tidak dapat menggunakan operasi ini untuk menambah atau mengedit tanda dalam Akun AWS berbeda.

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.

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
```

Hapus tag yang terkait dengan kunci KMS

Tag membantu mengidentifikasi dan mengatur AWS sumber daya Anda. Anda dapat menghapus tag yang terkait dengan kunci KMS yang dikelola pelanggan Anda di AWS KMS konsol atau dengan menggunakan UntagResourceoperasi. Anda tidak dapat mengedit atau menghapus tag dari file Kunci yang dikelola AWS.

Prosedur berikut menunjukkan cara menghapus tag dari kunci KMS. Contoh AWS KMS API menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Menggunakan AWS KMS konsol

1. Masuk ke Konsol Manajemen AWS 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 dari Kunci yang dikelola AWS)
4. Anda dapat menggunakan filter tabel untuk menampilkan hanya tombol KMS dengan tag tertentu. Untuk detailnya, lihat Melihat tag menggunakan AWS KMS konsol.

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 menghapus sebuah tanda, pilih Edit. Pada baris tanda, pilih Hapus, lalu pilih Simpan.
8. Untuk menyimpan perubahan Anda, memilih Simpan perubahan.

Menggunakan AWS KMS API

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 berbeda Akun 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
```

Lihat tag yang terkait dengan kunci KMS

Tag membantu mengidentifikasi dan mengatur AWS sumber daya Anda. Anda dapat melihat tag yang terkait dengan kunci KMS yang dikelola pelanggan Anda di AWS KMS konsol atau dengan menggunakan ListResourceTagsoperasi.

Prosedur berikut menunjukkan cara menemukan tag yang terkait dengan kunci KMS tertentu. Contoh AWS KMS API menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Menggunakan AWS KMS konsol

1. Masuk ke Konsol Manajemen AWS 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 dari Kunci yang dikelola AWS)
4. Anda dapat menggunakan filter tabel untuk menampilkan hanya tombol KMS dengan tag tertentu.

   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.
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.

Menggunakan AWS KMS API

ListResourceTagsOperasi mendapatkan tag untuk kunci KMS. parameter `KeyId` diperlukan. Anda tidak dapat menggunakan operasi ini untuk melihat tag pada tombol KMS secara berbeda Akun 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"
      }
  ]
}
```

Gunakan 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 merupakan bagian dari AWS KMS dukungan untuk kontrol akses berbasis atribut (ABAC). Untuk informasi tentang penggunaan tag untuk mengontrol akses ke AWS sumber daya, lihat Untuk apa ABAC? AWS dan Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di 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 KMS mendukung 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 `aws:ResourceTag/`*`tag-key`* kondisi hanya didukung 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 prinsip-prinsip di Anda Akun AWS yang saat ini memiliki izin penandaan dan pembatalan tag dan sesuaikan, jika perlu. Misalnya, kebijakan kunci default konsol untuk administrator kunci menyertakan `kms:TagResource` dan `kms:UntagResource` izin pada kunci KMS tersebut. Kebijakan IAM mungkin mengizinkan 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 Anda Akun 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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

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"`

JSON

```
{
```

```
"Version":"2012-10-17",
"Statement": [
  {
    "Sid": "IAMAllowCryptographicOperations",
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:GenerateDataKey*",
      "kms:Decrypt",
      "kms:ReEncrypt*"
    ],
    "Resource": "arn:aws:kms:*:111122223333:key/*"
  },
  {
    "Sid": "IAMDenyOnTag",
    "Effect": "Deny",
    "Action": [
      "kms:Encrypt",
      "kms:GenerateDataKey*",
      "kms:Decrypt",
      "kms:ReEncrypt*"
    ],
    "Resource": "arn:aws:kms:*:111122223333:key/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/Type": "Reserved"
      }
    }
  },
  {
    "Sid": "IAMDenyNoTag",
    "Effect": "Deny",
    "Action": [
      "kms:Encrypt",
      "kms:GenerateDataKey*",
      "kms:Decrypt",
      "kms:ReEncrypt*"
    ],
    "Resource": "arn:aws:kms:*:111122223333:key/*",
    "Condition": {
      "Null": {
        "aws:ResourceTag/Type": "true"
      }
    }
```

```
    }
  ]
}
```

Toko-toko utama

Toko kunci adalah lokasi yang aman untuk menyimpan dan menggunakan kunci kriptografi. Penyimpanan kunci standar AWS KMS juga mendukung metode untuk menghasilkan dan mengelola kunci yang disimpannya. Materi kunci kriptografi untuk kunci KMS yang Anda buat dihasilkan dan dilindungi oleh modul keamanan perangkat keras (HSM) yang divalidasi FIPS 140-3 Level 3. Materi kunci untuk kunci KMS Anda tidak pernah meninggalkan HSM tidak terenkripsi, dan semua operasi kriptografi pada kunci Anda dilakukan dalam FIPS 140-3 HSM ini.

Untuk memenuhi persyaratan khusus, AWS KMS juga mendukung toko kunci khusus. Ada dua jenis: toko AWS CloudHSM utama, didukung oleh AWS CloudHSM cluster khusus milik pelanggan, dan toko kunci eksternal, didukung oleh HSM atau sistem manajemen kunci di luar. AWS Cloud Bagian berikut membantu Anda memilih toko kunci yang tepat untuk kebutuhan spesifik Anda.

Memilih toko kunci yang tepat

Bagi sebagian besar pelanggan, toko kunci AWS KMS standar memberikan keseimbangan optimal antara keamanan, kinerja, biaya, dan kesederhanaan operasional. AWS KMS HSM terus diaudit dan ditinjau terhadap berbagai standar keamanan dan sertifikasi kepatuhan di seluruh dunia, mencakup program global, regional, pemerintah, dan khusus industri. Untuk detailnya, lihat AWS KMS kepatuhan.

Untuk banyak persyaratan kepatuhan, kontrol yang Anda miliki atas akses kunci melalui kebijakan kunci KMS sudah cukup. Jika Anda perlu menunjukkan kontrol atas materi kunci itu sendiri, pertimbangkan untuk menggunakan kunci AWS KMS impor dengan toko kunci AWS KMS standar. Pendekatan ini memberikan manfaat kepatuhan dari materi kunci yang dikendalikan pelanggan sambil mempertahankan keunggulan operasional AWS KMS infrastruktur, termasuk kinerja dan ketersediaan yang tinggi.

Tabel berikut membandingkan kompleksitas operasional relatif dari opsi penyimpanan AWS KMS utama. Semua opsi sama-sama aman; perbedaannya ada pada trade-off operasional.

AWS KMS Key Store Options — Comparison

All options are equally secure and the difference is operational trade-offs

	AWS KMS Standard Key Store	AWS KMS Standard Key Store with Imported Key Material	AWS CloudHSM Key Store	External Key Store
Effort to manage	Low	Medium	Medium	High
Availability Risk	Low (99.999% SLA)	Low (99.999% SLA)	Medium	High
TPS Constraints	Low	Low	High	High
Latency	Low	Low	Medium	High

Pertimbangkan untuk membuat penyimpanan kunci khusus hanya jika organisasi Anda memiliki persyaratan peraturan yang secara eksplisit mengamanatkan perlindungan material kunci dalam HSM penyewa tunggal atau di HSM yang Anda kendalikan di luar. AWS Meski begitu, buat hanya jika risiko tidak memenuhi persyaratan tersebut melebihi biaya tambahan, kompleksitas operasional, dan penurunan kinerja toko kunci khusus.

AWS KMS toko kunci standar

Setiap kunci KMS adalah kunci yang didukung perangkat keras, dengan bahan utamanya yang dihasilkan dan dilindungi oleh HSM standar. AWS KMS Jenis HSM ini dapat dianggap sebagai armada penyewa agnostik HSM yang memungkinkan toko kunci yang paling skalabel, berbiaya terendah, dan termudah untuk dikelola. Jika Anda membuat kunci KMS untuk digunakan dalam satu atau lebih Layanan AWS sehingga layanan dapat mengenkripsi data Anda atas nama Anda, Anda membuat kunci simetris. Jika Anda menggunakan kunci KMS untuk desain aplikasi Anda sendiri, Anda dapat memilih untuk membuat kunci enkripsi simetris, kunci asimetris, atau kunci HMAC.

Dalam opsi penyimpanan kunci standar, AWS KMS buat kunci Anda, lalu enkripsi di bawah kunci yang dikelola layanan secara internal. AWS KMS kemudian menyimpan beberapa salinan terenkripsi kunci Anda dalam sistem yang dirancang untuk daya tahan. Menghasilkan dan melindungi bahan utama Anda dalam jenis toko kunci standar memungkinkan Anda memanfaatkan sepenuhnya

skalabilitas, ketersediaan, dan daya tahan AWS KMS dengan beban operasional dan biaya terendah dari toko-toko AWS utama.

Fitur

Toko kunci standar mendukung semua AWS KMS fitur.

Operasi

Toko kunci standar menyediakan tingkat transaksi per detik (TPS) tertinggi dengan latensi terendah, menjadikannya ideal untuk aplikasi throughput tinggi dan integrasi layanan. AWS AWS KMS menawarkan SLA publik sebesar 99,999% pada ketersediaan kunci di toko kunci standar dan penskalaan kinerja otomatis untuk memenuhi tuntutan beban kerja Anda.

Model tanggung jawab

AWS mengelola semua infrastruktur, penskalaan, dan pemeliharaan. Ini menghasilkan nol beban operasional untuk Anda, dengan skalabilitas penuh, ketersediaan, dan daya tahan. Anda tetap bertanggung jawab atas kebijakan kunci IAM dan KMS yang mengontrol kapan, di mana, siapa, dan bagaimana setiap kunci dapat digunakan. Anda juga bertanggung jawab atas tag kunci, alias, dan peristiwa siklus hidup seperti rotasi, mengaktifkan atau menonaktifkan kunci, dan menjadwalkannya untuk dihapus.

AWS KMS toko kunci standar dengan bahan kunci impor

Di toko kunci standar, kunci KMS menghasilkan materi utamanya menggunakan FIPS-validated AWS KMS HSM. Namun, Anda dapat membuat kunci KMS tanpa materi kunci dan mengimpor sendiri —terkadang disebut bring your own key (BYOK). Ini memberi Anda kendali atas bagaimana materi kunci dihasilkan, berapa lama bahan tersebut tetap tersedia AWS KMS, dan kapan materi tersebut dihapus. Anda dapat mengatur waktu kedaluwarsa saat impor atau panggilan `DeleteImportedKeyMaterial` untuk segera mencabut akses.

Mengimpor materi kunci berguna ketika Anda perlu membuat kunci menggunakan sistem atau sumber entropi tertentu, menyimpan salinan cadangan di luar AWS, atau menghapus materi kunci pada jadwal yang ditentukan.

Fitur

Materi kunci yang diimpor mendukung semua AWS KMS fitur kecuali rotasi kunci otomatis dan kunci pasca-kuantum (ML-DSA).

Operasi

Kunci dengan bahan kunci impor tidak dapat dibedakan dari kunci di penyimpanan kunci standar dalam hal kinerja, membuatnya cocok untuk aplikasi throughput tinggi dan integrasi layanan. AWS AWS KMS menawarkan SLA publik sebesar 99,999% pada ketersediaan kunci dengan bahan kunci impor dan penskalaan kinerja otomatis untuk memenuhi tuntutan beban kerja Anda.

Model tanggung jawab

Selain tanggung jawab toko kunci standar yang dijelaskan sebelumnya, Anda bertanggung jawab atas daya tahan bahan kunci impor Anda. AWS mengelola infrastruktur dan penskalaan untuk ketersediaan. Anda dapat segera menghapus materi kunci menggunakan `DeleteImportedKeyMaterial` API; setelah Anda menghapus materi kunci apa pun yang terkait dengan kunci KMS, itu menjadi tidak dapat digunakan untuk operasi kriptografi sampai Anda mengimpor ulang materi kunci yang terkait dengan kunci itu.

AWS KMS toko kunci kustom

Jika Anda memerlukan kepemilikan langsung dan pengelolaan HSM yang menyimpan dan melindungi materi utama Anda, Anda dapat membuat toko kunci khusus. Toko kunci khusus adalah toko kunci di dalamnya AWS KMS yang didukung oleh manajer kunci di luar AWS KMS, yang Anda miliki dan kelola. Toko kunci khusus menggabungkan dan memperluas antarmuka manajemen kunci yang sudah dikenal AWS KMS dengan kemampuan untuk menghasilkan dan menggunakan materi utama di HSM Anda sendiri. Bila Anda menggunakan kunci KMS di toko kunci kustom, operasi kriptografi dilakukan menggunakan bahan kunci yang tetap berada dalam HSM atau manajer kunci Anda.

Toko kunci khusus tidak lebih aman daripada toko kunci standar, tetapi mereka memiliki implikasi manajemen dan biaya yang berbeda (dan lebih tinggi). Terlepas dari apakah Anda menggunakan toko kunci standar atau toko kunci khusus, layanan ini dirancang sehingga tidak ada seorang pun, termasuk AWS karyawan, yang dapat mengambil kunci teks biasa Anda atau menggunakannya tanpa izin Anda.

AWS KMS mendukung dua jenis toko kunci khusus: toko AWS CloudHSM kunci dan toko kunci eksternal.

Fitur yang didukung

Toko kunci khusus hanya mendukung kunci KMS enkripsi simetris. AWS KMS Fitur lainnya—seperti kunci asimetris, kunci HMAC, rotasi kunci otomatis, kunci multi-wilayah, dan bahan kunci impor — tidak tersedia di toko kunci khusus.

AWS CloudHSM toko kunci

AWS Awalnya meluncurkan AWS CloudHSM produk untuk membantu pelanggan bermigrasi dari HSM lokal ke HSM penyewa tunggal di cloud. Dengan ekstensi, AWS KMS kemudian memperkenalkan toko kunci khusus untuk pelanggan yang sama untuk terus menggunakan AWS CloudHSM di seluruh aplikasi mereka sendiri dan Layanan AWS terintegrasi melalui AWS KMS.

Anda dapat mengonfigurasi AWS KMS untuk menggunakan penyimpanan AWS CloudHSM kunci, tempat kunci dihasilkan, disimpan, dan digunakan dalam AWS CloudHSM cluster yang Anda miliki dan kelola. Permintaan untuk AWS KMS diteruskan ke cluster Anda AWS CloudHSM . Meskipun AWS CloudHSM cluster di-host di dalam AWS, mereka menyediakan instance HSM penyewa tunggal yang dimiliki dan dikelola oleh Anda. Karena AWS CloudHSM cluster dikelola oleh Anda di luar AWS KMS, toko AWS CloudHSM utama menyediakan TPS rendah dengan ketersediaan, daya tahan, dan konsistensi kinerja tergantung pada AWS CloudHSM cluster Anda.

Jika Anda harus menggunakan toko kunci khusus, toko AWS CloudHSM kunci lebih disukai daripada toko kunci eksternal: koneksi antara AWS KMS dan HSM Anda tetap berada di dalam AWS jaringan, jadi AWS memiliki ketersediaan dan konsistensinya daripada jalur yang berjalan di luar AWS melalui proxy XKS Anda. Either way, Anda masih mengontrol fungsi yang mengisolasi materi kunci Anda dan dapat mengunci akses ke kunci Anda.

Operasi

AWS CloudHSM key store menyediakan TPS rendah dan latensi lebih tinggi daripada kunci standar, dengan ketersediaan tergantung pada konfigurasi cluster Anda dan tidak ada cakupan SLA. Mereka hanya cocok untuk beban kerja transaksi per detik (TPS) rendah seperti enkripsi volume Amazon Elastic Block Store atau enkripsi basis data Amazon RDS, dan tidak boleh digunakan untuk skenario TPS tinggi termasuk layanan analisis data, operasi Amazon Simple Storage Service volume tinggi, atau beban kerja DynamoDB.

Model tanggung jawab

Selain tanggung jawab penyimpanan kunci standar yang dijelaskan sebelumnya, Anda bertanggung jawab atas ketersediaan dan daya tahan kunci kriptografi dan skalabilitas operasi di cluster Anda AWS CloudHSM . Secara desain, AWS CloudHSM memiliki visibilitas terbatas ke dalam konfigurasi dan log cluster milik pelanggan. Ini membatasi kemampuan AWS untuk menyelesaikan masalah akses utama atas nama Anda.

Toko kunci eksternal

Anda dapat mengonfigurasi AWS KMS untuk menggunakan penyimpanan kunci eksternal (XKS), tempat materi kunci dihasilkan, disimpan, dan digunakan dalam sistem manajemen kunci di luar AWS. Permintaan untuk AWS KMS diteruskan ke sistem yang dihosting secara eksternal melalui proxy XKS di jaringan Anda. Spesifikasi API proxy XKS terbuka, dan banyak vendor manajemen kunci komersial mendukungnya. Karena proxy XKS di-host oleh Anda atau pihak ketiga di luar AWS, toko kunci eksternal menyediakan TPS terendah dan latensi tertinggi dari semua opsi penyimpanan utama, dengan ketersediaan, daya tahan, dan konsistensi kinerja tergantung pada infrastruktur eksternal Anda.

Toko kunci eksternal tidak disarankan. Pertimbangkan penyimpanan kunci eksternal hanya jika Anda memiliki persyaratan eksplisit dan tidak dapat diubah untuk menyimpan materi utama di luar. AWS

Operasi

Toko kunci eksternal menyediakan TPS terendah dan latensi tertinggi dari semua opsi penyimpanan utama, dengan ketersediaan, daya tahan, dan konsistensi kinerja tergantung pada infrastruktur eksternal Anda dan tidak ada cakupan SLA. Mereka hanya cocok untuk beban kerja transaksi per detik (TPS) rendah seperti enkripsi volume Amazon Elastic Block Store atau enkripsi basis data Amazon RDS, dan tidak boleh digunakan untuk skenario TPS tinggi termasuk layanan analisis data, operasi Amazon Simple Storage Service volume tinggi, atau beban kerja DynamoDB.

Model tanggung jawab

Selain tanggung jawab penyimpanan kunci standar yang dijelaskan sebelumnya, Anda bertanggung jawab atas ketersediaan dan daya tahan kunci kriptografi dan skalabilitas manajer kunci eksternal termasuk proxy XKS dan HSM. Karena jalur komunikasi antara AWS dan proxy XKS Anda berjalan di luar AWS jaringan, AWS memiliki visibilitas terbatas ke masalah jaringan di jalur itu dan tidak memiliki kemampuan untuk memecahkan masalah mereka atas nama Anda.

AWS CloudHSM toko-toko utama

Toko AWS CloudHSM kunci adalah toko kunci khusus yang didukung oleh AWS CloudHSM cluster. Saat Anda membuat AWS KMS key di 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.

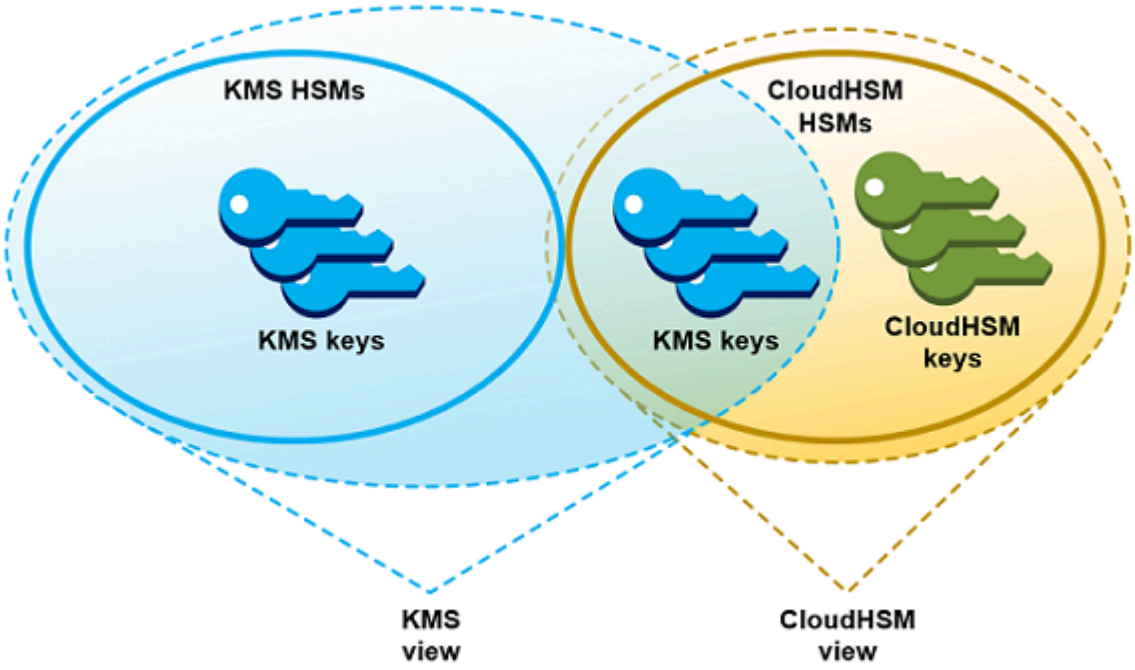
Apakah saya memerlukan toko AWS CloudHSM kunci?

Untuk memutuskan apakah toko AWS CloudHSM utama tepat untuk kebutuhan Anda, lihatMemilih toko kunci yang tepat.

Bagaimana cara kerja toko-toko AWS CloudHSM utama?

Setiap toko AWS CloudHSM kunci dikaitkan dengan AWS CloudHSM cluster di Anda Akun AWS. Saat Anda menghubungkan penyimpanan AWS CloudHSM kunci ke klasternya, AWS KMS buat infrastruktur jaringan untuk mendukung koneksi. Kemudian masuk ke AWS CloudHSM klien kunci di cluster menggunakan kredensional pengguna crypto khusus di cluster.

Anda membuat dan mengelola toko AWS CloudHSM utama Anda AWS KMS dan membuat serta mengelola klaster HSM Anda di. AWS CloudHSM Saat Anda membuat AWS KMS keys di toko AWS CloudHSM kunci, Anda melihat dan mengelola kunci KMS di AWS KMS. Tetapi Anda juga dapat melihat dan mengelola material kunci di AWS CloudHSM, seperti yang akan Anda lakukan untuk kunci lain dalam klaster.



Anda dapat membuat kunci KMS enkripsi simetris dengan materi kunci yang dihasilkan oleh AWS KMS di toko AWS CloudHSM kunci Anda. Kemudian gunakan teknik yang sama untuk melihat dan mengelola kunci KMS di toko AWS CloudHSM kunci 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 penyimpanan AWS CloudHSM kunci terputus, AWS KMS tidak dapat mengaksesnya, dan pengguna tidak dapat menggunakan kunci KMS di toko AWS CloudHSM kunci untuk operasi kriptografi. Lapisan kontrol tambahan ini membuat penyimpanan AWS CloudHSM kunci menjadi solusi yang kuat untuk organisasi yang membutuhkannya.

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.

Harga

Untuk informasi tentang biaya toko kunci AWS KMS kustom 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), Tiongkok (Beijing), Tiongkok (Ningxia), dan Eropa (Spanyol).

Fitur yang tidak didukung

AWS KMS tidak mendukung fitur berikut di toko kunci khusus.

- Kunci Asymmetric KMS
- Kunci HMAC KMS
- Kunci KMS dengan bahan kunci impor
- Rotasi kunci otomatis
- Multi-Region kunci

AWS CloudHSM konsep toko utama

Topik ini menjelaskan beberapa istilah dan konsep yang digunakan di toko-toko AWS CloudHSM utama.

AWS CloudHSM toko kunci

Toko AWS CloudHSM kunci adalah toko kunci khusus yang terkait dengan AWS CloudHSM cluster yang Anda miliki dan kelola. AWS CloudHSM cluster didukung oleh modul keamanan perangkat keras (HSM) yang disertifikasi di FIPS 140-2 atau FIPS 140-3Level 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 CloudHSM toko utama menggabungkan antarmuka manajemen kunci yang nyaman dan komprehensif AWS KMS dengan kontrol tambahan yang disediakan oleh AWS CloudHSM cluster di Anda Akun 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 AWS CloudHSM konsol, 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 dari AWS KMS, tetapi itu tidak mempengaruhi AWS CloudHSM cluster Anda.

AWS CloudHSM klaster

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 cluster 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 cluster baru atau menggunakan yang sudah ada. AWS KMS tidak perlu penggunaan cluster secara 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.

`kmsuser` Pengguna Crypto

Untuk membuat dan mengelola materi utama di AWS CloudHSM klaster terkait atas nama Anda, AWS KMS gunakan pengguna AWS CloudHSM kripto khusus (CU) di cluster 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 toko AWS CloudHSM kunci Anda, Anda membuat akun `kmsuser` CU di AWS CloudHSM cluster Anda menggunakan perintah user create di CloudHSM CLI. Kemudian ketika Anda membuat toko AWS CloudHSM kunci, Anda memberikan kata sandi `kmsuser` akun ke AWS KMS. Saat Anda menghubungkan toko kunci khusus, AWS KMS masuk ke cluster sebagai `kmsuser` CU dan memutar kata sandinya. AWS KMS mengenkripsi `kmsuser` kata sandi Anda sebelum menyimpannya dengan aman. Ketika kata sandi diputar, kata sandi baru dienkripsi dan disimpan dengan cara yang sama.

AWS KMS tetap 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 kunci yang `kmsuser` dimilikinya. 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 dalam sebuah AWS CloudHSM toko kunci

Anda dapat menggunakan AWS KMS API AWS KMS atau untuk membuat AWS KMS keys di 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 lainnya. 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 AWS layanan 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.

Kontrol akses ke toko AWS CloudHSM kunci 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.

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 Untuk informasi selengkapnya, lihat Otorisasi AWS KMS untuk mengelola AWS CloudHSM dan sumber daya Amazon EC2.

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 instance Amazon. EC2 Mereka mungkin juga perlu membuat dan menghapus HSMs, dan 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 KMSKebijakan kunci default untuk kunci KMS di penyimpanan AWS CloudHSM kunci identik dengan kebijakan kunci default untuk kunci KMS di. AWS KMSAttribute-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.

Buat 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.

> **Catatan**
>
> KMS tidak dapat berkomunikasi IPv6 dengan toko-toko AWS CloudHSM utama.
> 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 ada AWS CloudHSM .
> 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 baru

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 khusus (CU) di cluster HSMs yang AWS KMS dapat digunakan untuk membuat dan mengelola kunci atas nama Anda.

Sebelum Anda membuat toko AWS CloudHSM kunci, lakukan hal berikut:

Pilih AWS CloudHSM klaster

Setiap toko AWS CloudHSM kunci dikaitkan dengan tepat satu AWS CloudHSM cluster. Saat Anda membuat AWS KMS keys di toko AWS CloudHSM kunci, AWS KMS buat metadata kunci KMS, seperti ID dan Nama Sumber Daya Amazon (ARN) di. AWS KMS Kemudian menciptakan materi kunci di HSMs cluster terkait. Anda dapat membuat AWS CloudHSM cluster baru atau menggunakan yang sudah ada. AWS KMS tidak memerlukan akses eksklusif ke cluster.

AWS CloudHSM Cluster 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.

AWS CloudHSM Cluster yang Anda pilih harus memiliki karakteristik sebagai berikut:

- Klaster harus aktif.

  Anda harus membuat cluster, menginisialisasi, menginstal perangkat lunak AWS CloudHSM klien untuk platform Anda, dan kemudian mengaktifkan cluster. Untuk petunjuk terperinci, lihat Memulai AWS CloudHSM di Panduan AWS CloudHSM Pengguna.
- Mutual TLS (mTLS) tidak diaktifkan.

  KMS tidak mendukung mTL untuk cluster. Pengaturan ini tidak boleh diaktifkan.
- 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 khusus 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 kluster 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 KMS mengikat 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 tidak AWS CloudHSM didukung di semua Availability Zone, kami sarankan Anda membuat subnet pribadi 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 cluster, upaya untuk terhubung ke penyimpanan kunci kustom gagal dengan status kesalahan `SUBNET_NOT_FOUND` koneksi. Lihat perinciannya di 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 berakhir IPv4, 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.
- Cluster harus berisi setidaknya dua aktif HSMs di Availability Zone yang berbeda. Untuk memverifikasi jumlah HSMs, gunakan AWS CloudHSM konsol atau DescribeClustersoperasi. Jika perlu, Anda dapat menambahkan HSM.

Temukan sertifikat jangkar tepercaya

Saat membuat penyimpanan kunci khusus, Anda harus mengunggah sertifikat jangkar kepercayaan untuk AWS CloudHSM AWS KMS klaster. AWS KMS membutuhkan sertifikat jangkar kepercayaan untuk menghubungkan penyimpanan AWS CloudHSM kunci ke AWS CloudHSM klaster terkaitnya.

Setiap AWS CloudHSM cluster aktif memiliki sertifikat jangkar kepercayaan. Ketika menginisialisasi klaster, Anda menghasilkan sertifikat ini, menyimpannya dalam file `customerCA.crt`, dan menyalinnya ke host yang terhubung ke klaster.

Buat pengguna **`kmsuser`** crypto 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` untuk AWS KMS. Setiap kali Anda menghubungkan penyimpanan AWS CloudHSM kunci ke AWS CloudHSM cluster terkait AWS 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.

Catatan

Prosedur berikut menggunakan alat baris perintah AWS CloudHSM Client SDK 5, CloudHSM CLI. CloudHSM CLI `key-handle` menggantikan dengan. `key-reference` Pada tanggal 1 Januari 2025, AWS CloudHSM akan mengakhiri dukungan untuk alat baris perintah Client SDK 3, CloudHSM Management Utility (CMU) dan Key Management Utility (KMU). Untuk informasi selengkapnya tentang perbedaan antara alat baris perintah Client SDK 3 dan alat baris perintah Client SDK 5, lihat Memigrasi dari Client SDK 3 CMU dan KMU ke Client SDK 5 CloudHSM CLI di Panduan Pengguna.AWS CloudHSM

1. Ikuti prosedur memulai seperti yang dijelaskan dalam topik Memulai dengan CloudHSM Command Line Interface (CLI) dari Panduan Pengguna.AWS CloudHSM
2. Gunakan perintah user create untuk membuat CU bernama`kmsuser`.

   Kata sandi harus berisi 7–32 karakter alfanumerik. Kata sandi peka huruf besar/kecil dan tidak dapat berisi karakter khusus.

   Contoh perintah berikut menciptakan `kmsuser` CU.

```
aws-cloudhsm > user create --username kmsuser --role crypto-user
Enter password:
Confirm password:
{
 "error_code": 0,
 "data": {
   "username": "kmsuser",
   "role": "crypto-user"
 }
```

```
}
```

Buat toko AWS CloudHSM kunci baru

Setelah merakit prasyarat, Anda dapat membuat toko AWS CloudHSM kunci baru di AWS KMS konsol atau dengan menggunakan operasi. CreateCustomKeyStore

Menggunakan AWS KMS konsol

Saat Anda membuat penyimpanan AWS CloudHSM kunci di Konsol Manajemen AWS, Anda dapat menambahkan dan membuat prasyarat sebagai bagian dari alur kerja Anda. Namun, prosesnya lebih cepat saat Anda merakitnya terlebih dahulu.

1. Masuk ke Konsol Manajemen AWS 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 AWS CloudHSM cluster baru, pilih link Create an AWS CloudHSM cluster.

   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 ada AWS 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.

Menggunakan AWS KMS 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 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 kepercayaan, bukan isinya. 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 ada AWS CloudHSM .

Berikutnya: Untuk menggunakan toko AWS CloudHSM kunci, hubungkan ke AWS CloudHSM klaster nya.

Lihat toko AWS CloudHSM kunci

Anda dapat melihat toko AWS CloudHSM utama di setiap akun dan Wilayah dengan menggunakan AWS KMS konsol atau DescribeCustomKeyStoresoperasi.

Menggunakan AWS KMS konsol

Ketika Anda melihat toko-toko AWS CloudHSM utama di Konsol Manajemen AWS, Anda dapat melihat yang berikut:

- Nama dan ID toko kunci kustom
- ID AWS CloudHSM klaster terkait
- Jumlah HSMs dalam cluster
- 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 Konsol Manajemen AWS 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.

Menggunakan AWS KMS 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>"
```

```
      }
   ]
}
```

A `ConnectionState` `Disconnected` menunjukkan bahwa toko kunci khusus tidak pernah terhubung atau sengaja terputus dari klasternya AWS CloudHSM. 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>"
      }
   ]
}
```

Pelajari lebih lanjut:

- Lihat toko kunci eksternal
- Identifikasi kunci KMS di AWS CloudHSM toko-toko utama
- Pencatatan log AWS KMS Panggilan API dengan AWS CloudTrail

Edit pengaturan penyimpanan 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.

   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 Anda Akun AWS.

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

   ID cluster dari AWS CloudHSM cluster terkait.

   Edit nilai ini untuk menggantikan AWS CloudHSM cluster terkait dengan yang asli. Anda dapat menggunakan fitur ini untuk memperbaiki penyimpanan kunci khusus jika AWS CloudHSM klasternya rusak atau dihapus.

   Tentukan AWS CloudHSM klaster yang berbagi riwayat cadangan dengan klaster asli dan memenuhi persyaratan untuk asosiasi dengan penyimpanan kunci khusus, termasuk dua aktif HSMs 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`.

AWS KMS Memberitahu kata sandi `kmsuser` CU saat ini di AWS CloudHSM cluster. Tindakan ini tidak mengubah kata sandi `kmsuser` CU di AWS CloudHSM cluster.

Jika Anda mengubah kata sandi `kmsuser` CU di AWS CloudHSM cluster, gunakan fitur ini untuk memberi tahu AWS KMS `kmsuser` kata sandi baru. Jika tidak, AWS KMS tidak dapat login ke klaster dan semua upaya untuk menghubungkan penyimpanan kunci kustom untuk klaster mengalami kegagalan.

3. Sambungkan kembali penyimpanan kunci kustom ke klaster AWS CloudHSM -nya.

Edit pengaturan toko kunci Anda

Anda dapat mengedit pengaturan penyimpanan AWS CloudHSM kunci Anda di AWS KMS konsol atau dengan menggunakan UpdateCustomKeyStoreoperasi.

Menggunakan AWS KMS konsol

Saat Anda mengedit penyimpanan AWS CloudHSM kunci, Anda dapat mengubah salah satu atau nilai yang dapat dikonfigurasi.

1. Masuk ke Konsol Manajemen AWS 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.
   - Ketik ID cluster dari AWS CloudHSM cluster terkait.

- Ketik kata sandi pengguna `kmsuser` kripto saat ini di AWS CloudHSM cluster 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.

Menggunakan AWS KMS 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 mengatakan AWS KMS bahwa `kmsuser` kata sandi 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
```

Connect 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 CloudHSM toko 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 aktif HSMs. Untuk bantuan dengan kegagalan koneksi, lihat the section called “Memecahkan masalah penyimpanan kunci kustom”.

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 KMS tetap masuk ke AWS CloudHSM klien selama toko AWS CloudHSM kunci terhubung.

Untuk membuat koneksi, AWS KMS buat grup keamanan bernama `kms-<custom key store ID>` di virtual private cloud (VPC) cluster. Grup keamanan memiliki aturan tunggal yang memungkinkan lalu lintas masuk dari grup keamanan cluster. AWS KMS juga membuat elastic network interface (ENI) di setiap Availability Zone dari subnet pribadi untuk cluster. AWS KMS menambahkan ENIs ke grup `kms-<cluster ID>` keamanan dan grup keamanan untuk cluster. 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.

- AWS CloudHSM Cluster yang terkait harus mengandung setidaknya satu HSM aktif. Untuk menemukan jumlah HSMs 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.

Connect dan sambungkan kembali ke toko AWS CloudHSM utama Anda

Anda dapat menghubungkan, atau menghubungkan kembali, toko AWS CloudHSM kunci Anda di AWS KMS konsol atau dengan menggunakan ConnectCustomKeyStoreoperasi.

Menggunakan AWS KMS konsol

Untuk menghubungkan toko AWS CloudHSM kunci di Konsol Manajemen AWS, 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 Konsol Manajemen AWS 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 toko 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 “Buat kunci KMS di toko AWS CloudHSM kunci”.

Menggunakan AWS KMS API

Untuk menghubungkan toko AWS CloudHSM kunci yang terputus, gunakan ConnectCustomKeyStoreoperasi. AWS CloudHSM Cluster 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 hal ini, tidak AWS KMS dapat menemukan AWS CloudHSM cluster di akun Anda dengan ID cluster`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"
   ],
}
```

Putuskan 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.

> **Note**
>
> AWS CloudHSM toko 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 aktif HSMs. Untuk bantuan dengan kegagalan koneksi, lihat the section called “Memecahkan masalah penyimpanan kunci kustom”.

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 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.

> **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. Saat Anda memutuskan penyimpanan kunci khusus, AWS KMS keluar dari akun pengguna `kmsuser` kripto di AWS CloudHSM klien. 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 Konsol Manajemen AWS 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.

Putuskan sambungan toko AWS CloudHSM kunci Anda

Anda dapat memutuskan sambungan toko AWS CloudHSM kunci Anda di AWS KMS konsol atau dengan menggunakan DisconnectCustomKeyStoreoperasi.

Putuskan sambungan menggunakan konsol AWS KMS

Untuk memutuskan penyimpanan AWS CloudHSM kunci yang terhubung di AWS KMS konsol, mulailah dengan memilih toko AWS CloudHSM kunci dari halaman Toko Kunci Kustom.

1. Masuk ke Konsol Manajemen AWS 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 menggunakan API AWS KMS

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>"
   ],
}
```

Hapus 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 mempengaruhi AWS CloudHSM cluster, nya HSMs, 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 pun AWS 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, lihatthe section called “Identifikasi kunci KMS di AWS CloudHSM toko-toko utama”.
- Putuskan sambungan penyimpanan AWS CloudHSM kunci dari AWS CloudHSM klasternya.

Alih-alih menghapus penyimpanan AWS CloudHSM kunci, pertimbangkan untuk memutuskannya dari cluster terkait AWS CloudHSM . Sementara toko AWS CloudHSM kunci terputus, Anda dapat mengelola toko AWS CloudHSM kunci dan nya AWS 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.

Hapus toko AWS CloudHSM kunci Anda

Anda dapat menghapus toko AWS CloudHSM kunci Anda di AWS KMS konsol atau dengan menggunakan DeleteCustomKeyStoreoperasi.

Menggunakan AWS KMS konsol

Untuk menghapus toko AWS CloudHSM kunci di Konsol Manajemen AWS, mulailah dengan memilih toko AWS CloudHSM kunci dari halaman Toko kunci kustom.

1. Masuk ke Konsol Manajemen AWS 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.

Menggunakan AWS KMS 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 pun AWS KMS keys. Anda tidak dapat menghapus toko kunci khusus yang berisi kunci KMS. Perintah contoh pertama DescribeKeymenggunakan ListKeysdan mencari AWS KMS keys di toko AWS CloudHSM kunci dengan contoh ID toko kunci *cks-1234567890abcdef0* 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
```

Memecahkan masalah penyimpanan kunci kustom

AWS CloudHSM toko-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. Lihat perinciannya di Identifikasi dan lihat 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 id kunci dari bahan kunci untuk kunci KMS dan, jika perlu, cobalah untuk memulihkan materi kunci.
- Semua HSMs 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 HSMs dalam sebuah AWS CloudHSM cluster, gunakan AWS CloudHSM konsol atau DescribeClustersoperasi. Untuk menambahkan HSM ke cluster, gunakan AWS CloudHSM konsol atau CreateHsmoperasi.
- AWS CloudHSM Cluster 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 petunjuk, lihat Bagaimana memulihkan materi kunci yang dihapus untuk kunci KMS.
- AWS CloudHSM Cluster 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 AWS CloudHSM cluster dengan ID cluster 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 AWS CloudHSM cluster terkait tidak mengandung apapun HSMs. Untuk menghubungkan, klaster harus memiliki minimal satu HSM. Untuk menemukan jumlah HSMs di cluster, gunakan DescribeClustersoperasi. Untuk mengatasi kesalahan ini, tambahkan minimal satu HSM ke klaster. Jika Anda menambahkan beberapa HSMs, yang terbaik adalah membuatnya 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 pengaturan ulang `kmsuser` kata sandi, gunakan cadangan AWS CloudHSM klaster terbaru.

- `INTERNAL_ERROR`menunjukkan bahwa tidak AWS KMS 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 masuk ke AWS CloudHSM cluster terkait karena tidak memiliki kata sandi `kmsuser` akun 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 dalam konfigurasi AWS CloudHSM cluster 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 cluster dari cadangan terbaru dari AWS CloudHSM cluster 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 pengaturan ulang `kmsuser` kata sandi, gunakan cadangan AWS CloudHSM klaster terbaru.

- `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 perubahan kata sandi pengguna di CloudHSM CLI 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 `kmsuser` CU masuk ke AWS CloudHSM cluster terkait. Ini AWS KMS mencegah memutar kata sandi `kmsuser` akun dan masuk ke cluster. 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 `kmsuser` CU di AWS CloudHSM cluster 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 aturan Jaringan ACLs dan grup keamanan untuk VPC AWS CloudHSM klaster 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. Misalnya, KMS tidak dapat berkomunikasi dengan AWS CloudHSM cluster. IPv6 Untuk detail tentang prasyarat, lihat. Buat toko AWS CloudHSM kunci

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 kredensi 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 perubahan kata sandi pengguna di CloudHSM CLI untuk mengatur kata sandi ke nilai yang diketahui.
   - Jika `ConnectionErrorCode` nilainya`USER_LOCKED_OUT`, Anda harus menggunakan perintah perubahan kata sandi pengguna di CloudHSM CLI 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 KMS tidak mencoba menghapus materi kunci dari cadangan cluster.

AWS KMS melaporkan 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 ID `backing-key-id` () dari materi kunci.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-12-10T14:23:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
```

```
    "responseElements":  {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"backingKeyId\":\"backing-key-id\"}]",
        "backingKeysDeletionStatus": "[{\"backingKeyId\":\"backing-key-id\",
\"deletionStatus\":\"FAILURE\"}]"
    },
    "eventID": "c21f1f47-f52b-4ffe-bff0-6d994403cf40",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
}
```

> **Catatan**
>
> Prosedur berikut menggunakan alat baris perintah AWS CloudHSM Client SDK 5, CloudHSM CLI. CloudHSM CLI `key-handle` menggantikan dengan. `key-reference`
> Pada tanggal 1 Januari 2025, AWS CloudHSM akan mengakhiri dukungan untuk alat baris perintah Client SDK 3, CloudHSM Management Utility (CMU) dan Key Management Utility (KMU). Untuk informasi selengkapnya tentang perbedaan antara alat baris perintah Client SDK 3 dan alat baris perintah Client SDK 5, lihat Memigrasi dari Client SDK 3 CMU dan KMU ke Client SDK 5 CloudHSM CLI di Panduan Pengguna.AWS CloudHSM

Prosedur berikut menunjukkan cara menghapus materi kunci yatim piatu dari cluster terkait. AWS CloudHSM

1. Putuskan sambungan toko AWS CloudHSM kunci, jika belum terputus, lalu login, seperti yang dijelaskan di. Cara memutuskan koneksi dan login

   > **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.

2. Gunakan perintah hapus kunci di CloudHSM CLI untuk menghapus kunci HSMs dari dalam cluster.

   Semua entri CloudTrail log untuk operasi kriptografi dengan kunci KMS di toko AWS CloudHSM kunci menyertakan `additionalEventData` bidang dengan dan. `customKeyStoreId` `backingKey` Nilai yang dikembalikan di `backingKeyId` bidang adalah atribut kunci `id` CloudHSM. Kami menyarankan `id` untuk memfilter operasi penghapusan kunci dengan menghapus materi kunci yatim piatu yang Anda identifikasi di log Anda. CloudTrail

   AWS CloudHSM mengenali `backingKeyId` nilai sebagai nilai heksadesimal. Untuk memfilter`id`, Anda harus menambahkan dengan. `backingKeyId` `0x` Misalnya, jika `backingKeyId` di CloudTrail log Anda`1a2b3c45678abcdef`, Anda akan memfilter berdasarkan**`0x`**`1a2b3c45678abcdef`.

   Contoh berikut menghapus kunci dari HSMs dalam cluster Anda. `backing-key-id`Ini tercantum dalam entri CloudTrail log. Sebelum menjalankan perintah ini, ganti contoh `backing-key-id` dengan yang valid dari akun Anda.

   ```
   aws-cloudhsm key delete --filter attr.id="0x<backing-key-id>"
   {
     "error_code": 0,
     "data": {
       "message": "Key deleted successfully"
     }
   }
   ```

3. Keluar dan sambungkan kembali toko AWS CloudHSM kunci seperti yang dijelaskan diCara 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, dapat menghapus kunci, kecil kemungkinan kunci akan dihapus dari yang HSMs tidak sengaja.

Namun, jika materi kunci untuk kunci KMS dihapus dari HSMs dalam cluster, status kunci dari 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 `kmsuser` akun. Namun, untuk beberapa tugas, Anda perlu memutuskan penyimpanan AWS CloudHSM kunci, masuk ke cluster sebagai `kmsuser` CU, dan menggunakan CloudHSM Command Line Interface (CLI).

> **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 sebagai`kmsuser`, 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 perlu masuk ke cluster terkait sebagai pengguna `kmsuser` kripto.

> **Catatan**
>
> Prosedur berikut menggunakan alat baris perintah AWS CloudHSM Client SDK 5, CloudHSM CLI. CloudHSM CLI `key-handle` menggantikan dengan. `key-reference`
> Pada tanggal 1 Januari 2025, AWS CloudHSM akan mengakhiri dukungan untuk alat baris perintah Client SDK 3, CloudHSM Management Utility (CMU) dan Key Management Utility (KMU). Untuk informasi selengkapnya tentang perbedaan antara alat baris perintah Client SDK 3 dan alat baris perintah Client SDK 5, lihat Memigrasi dari Client SDK 3 CMU dan KMU ke Client SDK 5 CloudHSM CLI di Panduan Pengguna.AWS CloudHSM

1. Putuskan sambungan toko AWS CloudHSM kunci, jika belum terputus. Anda dapat menggunakan AWS KMS konsol atau AWS KMS API.

   Saat AWS CloudHSM kunci Anda AWS KMS terhubung, masuk sebagai file`kmsuser`. 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. Gunakan perintah login untuk login sebagai admin. Gunakan prosedur yang dijelaskan di bagian Menggunakan CloudHSM CLI dari Panduan Pengguna.AWS CloudHSM

   ```
   aws-cloudhsm > login --username admin --role admin
             Enter password:
   {
     "error_code": 0,
     "data": {
       "username": "admin",
   ```

```
    "role": "admin"
  }
}
```

3. Gunakan perintah perubahan kata sandi pengguna di CloudHSM CLI untuk mengubah kata sandi akun menjadi kata sandi yang Anda ketahui. `kmsuser` (AWS KMS memutar 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.
4. Login seperti `kmsuser` menggunakan kata sandi yang Anda tetapkan. Untuk petunjuk terperinci, lihat bagian Menggunakan CloudHSM CLI pada Panduan Pengguna.AWS CloudHSM

```
aws-cloudhsm > login --username kmsuser --role crypto-user
          Enter password:
{
  "error_code": 0,
  "data": {
    "username": "kmsuser",
    "role": "crypto-user"
  }
}
```

Cara logout dan menghubungkan kembali

Gunakan prosedur berikut setiap kali Anda harus keluar sebagai pengguna `kmsuser` kripto dan sambungkan kembali toko kunci Anda.

> **Catatan**
>
> Prosedur berikut menggunakan alat baris perintah AWS CloudHSM Client SDK 5, CloudHSM CLI. CloudHSM CLI `key-handle` menggantikan dengan. `key-reference`
> Pada tanggal 1 Januari 2025, AWS CloudHSM akan mengakhiri dukungan untuk alat baris perintah Client SDK 3, CloudHSM Management Utility (CMU) dan Key Management Utility (KMU). Untuk informasi selengkapnya tentang perbedaan antara alat baris perintah Client SDK 3 dan alat baris perintah Client SDK 5, lihat Memigrasi dari Client SDK 3 CMU dan KMU ke Client SDK 5 CloudHSM CLI di Panduan Pengguna.AWS CloudHSM

1. Lakukan tugas, lalu gunakan perintah logout di CloudHSM CLI untuk keluar. Jika Anda tidak keluar, upaya untuk menghubungkan kembali toko AWS CloudHSM kunci Anda akan gagal.

```
aws-cloudhsm  logout
{
  "error_code": 0,
  "data": "Logout successful"
}
```

2. Edit pengaturan kata sandi `kmsuser` untuk penyimpanan kunci kustom.

   Ini memberi tahu AWS KMS kata sandi saat ini untuk `kmsuser` di cluster. Jika Anda menghilangkan langkah ini, tidak AWS KMS akan dapat masuk ke cluster sebagai`kmsuser`, dan semua upaya untuk menghubungkan kembali toko 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 ubah `kmsuser` kata sandi 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.

Apakah saya memerlukan toko kunci eksternal?

Untuk memutuskan apakah toko kunci eksternal tepat untuk kebutuhan Anda, lihatMemilih toko kunci yang tepat.

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 detailnya, lihat 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.

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 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 Tiongkok (Beijing) dan Tiongkok (Ningxia).

Fitur yang tidak didukung

AWS KMS tidak mendukung fitur-fitur berikut di toko kunci khusus.

- Tombol Asymmetric KMS
- Kunci HMAC KMS
- Kunci KMS dengan bahan kunci impor
- Rotasi kunci otomatis
- Multi-Region kunci

Pelajari lebih lanjut:

- Mengumumkan Toko Kunci AWS KMS Eksternal di Blog AWS Berita.

Konsep toko kunci eksternal

Pelajari istilah dan konsep dasar yang digunakan di toko kunci eksternal.

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 pengelola 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 manajer 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 Anda sebagai bagian dari strategi perlindungan data Anda 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 menghubungkan ke pengelola kunci eksternal Anda, lihatPilih opsi konektivitas proxy penyimpanan 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.

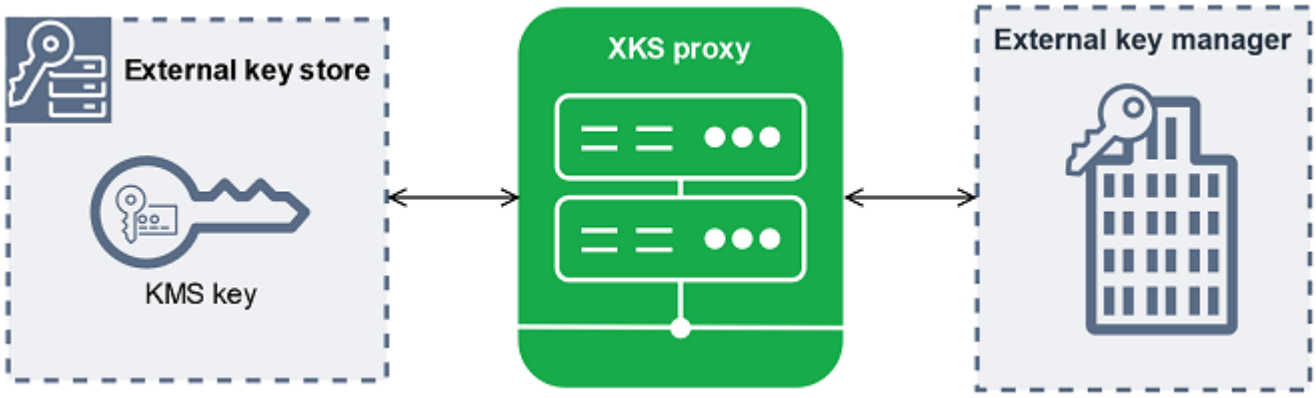
Selain kunci eksternal Anda, kunci KMS di toko kunci eksternal juga memiliki materi AWS KMS utama. Data yang dilindungi oleh kunci KMS dienkripsi terlebih dahulu dengan AWS KMS menggunakan materi AWS KMS kunci, dan kemudian oleh pengelola kunci eksternal Anda menggunakan kunci eksternal Anda. Proses 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 oleh 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 mentransmisikan 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.

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. Anda juga dapat menghubungkan toko kunci eksternal Anda ke layanan endpoint VPC Amazon yang dimiliki oleh orang lain. Akun AWS

Untuk detail tentang opsi konektivitas proxy penyimpanan kunci eksternal, lihatPilih opsi konektivitas proxy penyimpanan kunci eksternal.

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 otentikasi 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.

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 Pantau 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.

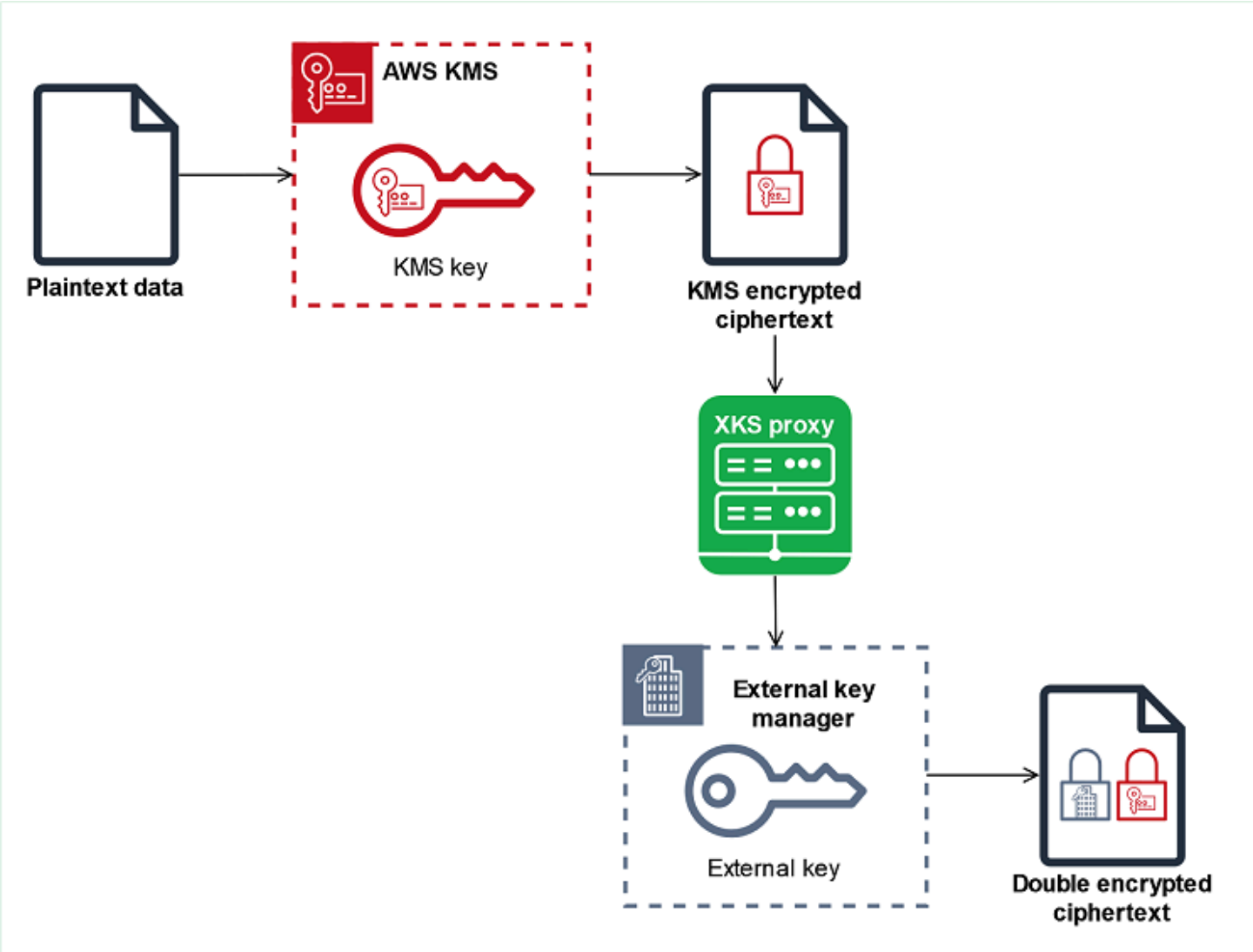
API Proksi	Deskripsi	AWS KMS Operasi terkait
Dekripsi	AWS KMS mengirimkan ciphertext untuk didekripsi, dan ID kunci eksternal untuk digunakan. Algoritma enkripsi yang diperlukan adalah AES_GCM.	Dekripsi, ReEncrypt
Enkripsi	AWS KMS mengirimkan data yang akan dienkripsi, dan ID kunci eksternal untuk digunakan. Algoritma enkripsi yang diperlukan adalah AES_GCM.	Enkripsi,, GenerateDataKey, GenerateDataKeyWithoutPlaintextReEncrypt
GetHealthStatus	AWS KMS meminta informasi tentang status proxy dan manajer kunci eksternal Anda.	CreateCustomKeyStore(untuk konektivitas titik akhir publik), ConnectCustomKeyStore(untuk konektivitas layanan titik akhir VPC)

API Proksi	Deskripsi	AWS KMS Operasi terkait
	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	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 dicabik-cabik 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 perangkat keras bersertifikat (HSM) AWS KMS FIPS 140-3 Security Level 3.
- 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.

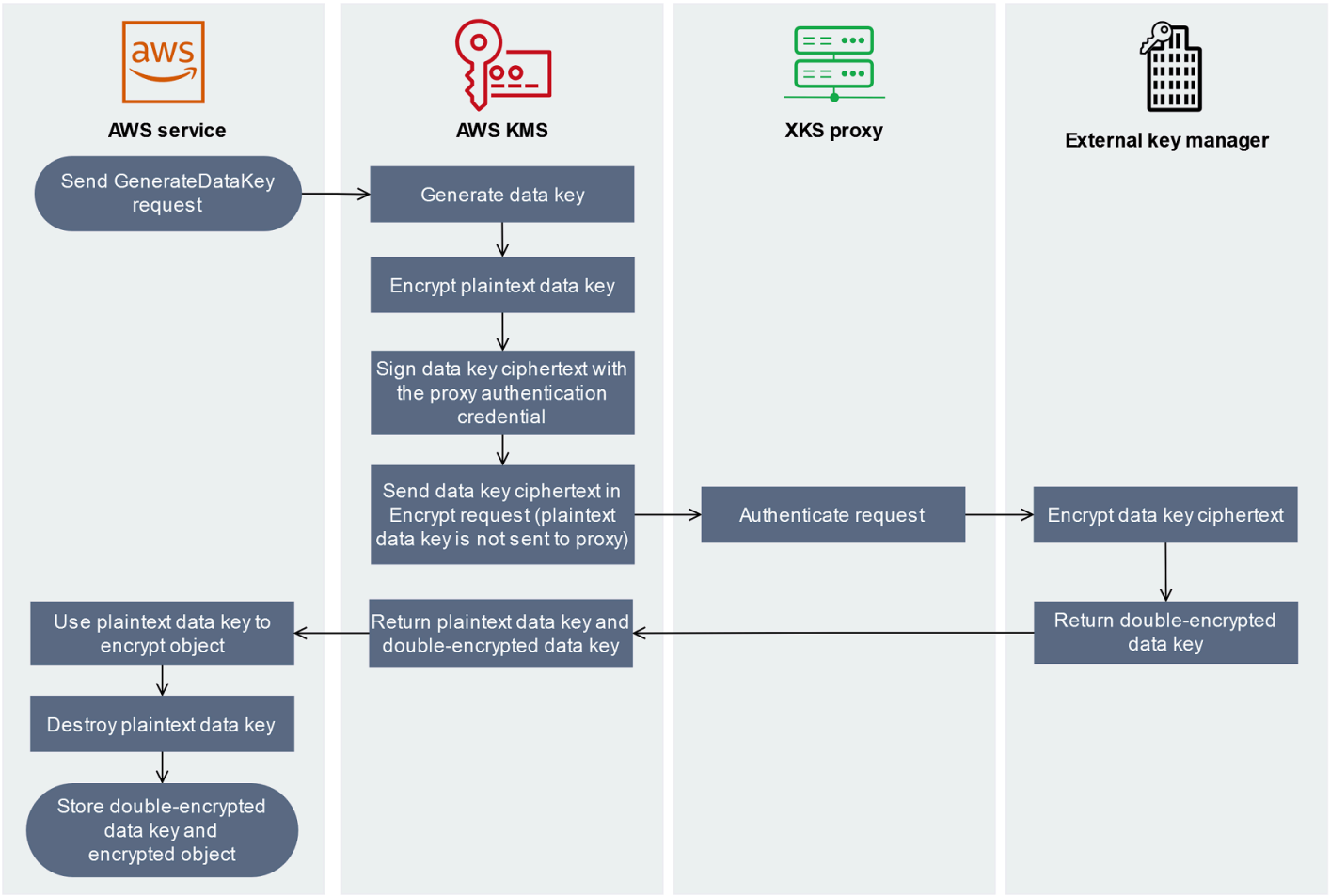
- Anda tidak dapat mendekripsi ciphertext apa pun yang dienkripsi oleh kunci KMS di penyimpanan kunci eksternal di luar AWS, bahkan jika Anda memiliki materi kunci eksternal.
- Anda tidak dapat membuat ulang kunci KMS yang telah dihapus dari penyimpanan kunci eksternal, bahkan jika Anda memiliki materi 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

Kontrol 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
- Mengotorisasi pengguna kunci KMS di toko kunci eksternal
- Otorisasi AWS KMS untuk berkomunikasi dengan proxy penyimpanan kunci eksternal Anda
- Otorisasi proxy penyimpanan kunci eksternal (opsional)
- Autentikasi mTLS (tidak digunakan lagi)

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`

Untuk membuat penyimpanan kunci eksternal dengan konektivitas layanan titik akhir Amazon VPC dan layanan titik akhir VPC dimiliki oleh yang berbeda Akun AWS, Anda juga memerlukan izin berikut:

- `ec2:DescribeVPCEndpointServices`

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 KMSKebijakan kunci default untuk kunci KMS di penyimpanan kunci eksternal identik dengan kebijakan kunci default untuk kunci KMS di. AWS KMSAttribute-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 di Akun 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 KMS berkomunikasi dengan pengelola kunci eksternal Anda hanya melalui proxy penyimpanan kunci eksternal yang Anda berikan. AWS KMS mengautentikasi 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 apa pun.

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 KMS Untuk memungkinkan membuat titik akhir antarmuka, gunakan konsol Amazon VPC atau ModifyVpcEndpointServicePermissionsoperasinya. Izinkan izin untuk prinsipal berikut:`cks.kms.<region>.amazonaws.com`.

Jika layanan titik akhir VPC Amazon Anda dimiliki oleh yang berbeda Akun AWS selain Akun AWS memiliki penyimpanan kunci eksternal (XKS), Anda juga harus mengizinkan akses XKS ke layanan titik akhir VPC. Untuk melakukannya, izinkan Akun AWS ID XKS sebagai prinsipal untuk layanan endpoint Amazon VPC.

Same Akun AWS

Ketika layanan titik akhir VPC Anda dimiliki Akun AWS sama dengan penyimpanan kunci eksternal Anda, Anda harus menambahkan AWS KMS ke daftar Izinkan prinsipal untuk layanan titik akhir VPC Anda.

Contoh berikut menggunakan `modify-vpc-endpoint-service-permissions` AWS CLI perintah untuk memungkinkan untuk terhubung AWS KMS ke layanan titik akhir VPC yang ditentukan di Wilayah AS Barat (Oregon) (us-west-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`

Cross Akun AWS

Ketika layanan titik akhir VPC Anda dimiliki oleh yang lain Akun AWS, Anda harus menambahkan keduanya AWS KMS dan penyimpanan kunci eksternal Anda ke daftar Izinkan prinsipal untuk layanan titik akhir VPC Anda.

Contoh berikut menggunakan `modify-vpc-endpoint-service-permissions` AWS CLI perintah untuk mengizinkan keduanya AWS KMS dan penyimpanan kunci eksternal (XKS) Anda terhubung ke layanan titik akhir VPC yang ditentukan di Wilayah AS Barat (Oregon) (us-west-2). Sebelum menggunakan perintah ini, ganti ID layanan Amazon VPC Wilayah AWS, dan ARN utama IAM dengan nilai yang valid untuk konfigurasi Anda. Prinsipal IAM harus diganti dengan prinsipal di pemilik XKS. Akun AWS

Dalam contoh ini, `arn:aws:iam::123456789012:role/cks_role` adalah prinsipal IAM di akun pemilik XKS, yang akan digunakan untuk membuat, memperbarui, atau menghubungkan XKS ke layanan titik akhir VPC Anda. Jika Anda ingin mengizinkan semua prinsipal di akun pemilik XKS untuk mengakses layanan titik akhir VPC Anda, Anda dapat menentukan. `arn:aws:iam::123456789012:root`

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
```

```
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com",
 "arn:aws:iam::123456789012:role/cks_role"]'
```

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 mengimplementasikan 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 memengaruhi 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.

Autentikasi mTLS (tidak digunakan lagi)

Versi sebelumnya dari panduan ini menyebutkan Mutual Transport Layer Security (mTLS) sebagai mekanisme otentikasi sekunder opsional untuk mengautentikasi permintaan dari. AWS KMS Dengan mTL, kedua belah pihak (AWS KMS sebagai klien dan proxy XKS sebagai server) berkomunikasi melalui saluran TLS menggunakan sertifikat untuk mengautentikasi satu sama lain.

Namun, perubahan pada Kebijakan Program Root Chrome (Bagian 4.2.2) melarang root tepercaya publik yang CAs termasuk dalam Toko Root Chrome menerbitkan sertifikat dengan ekstensi ClientAuth Extended Key Usage (EKU) setelah 15 Juni 2026. Akibatnya, tidak AWS KMS dapat lagi memperoleh sertifikat klien yang cocok untuk MTL dari Amazon Trust Services. Proxy XKS apa pun yang digunakan untuk membuat penyimpanan kunci eksternal baru AWS KMS setelah 16 Maret 2026 tidak boleh memerlukan mTL. Setelah 15 Juni 2026, proxy XKS apa pun yang dikonfigurasi untuk mewajibkan mTL tidak akan dapat berkomunikasi dengannya. AWS KMS Pelanggan harus mengandalkan otentikasi SiGv4 untuk memverifikasi bahwa permintaan berasal. AWS KMS Untuk informasi selengkapnya, lihat Kredensi otentikasi proxy penyimpanan kunci eksternal.

Pilih opsi konektivitas proxy penyimpanan 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.

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 mengonfigurasinya. AWS KMS mendukung opsi konektivitas berikut.

- Konektivitas titik akhir publik
- Konektivitas layanan titik akhir VPC

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.

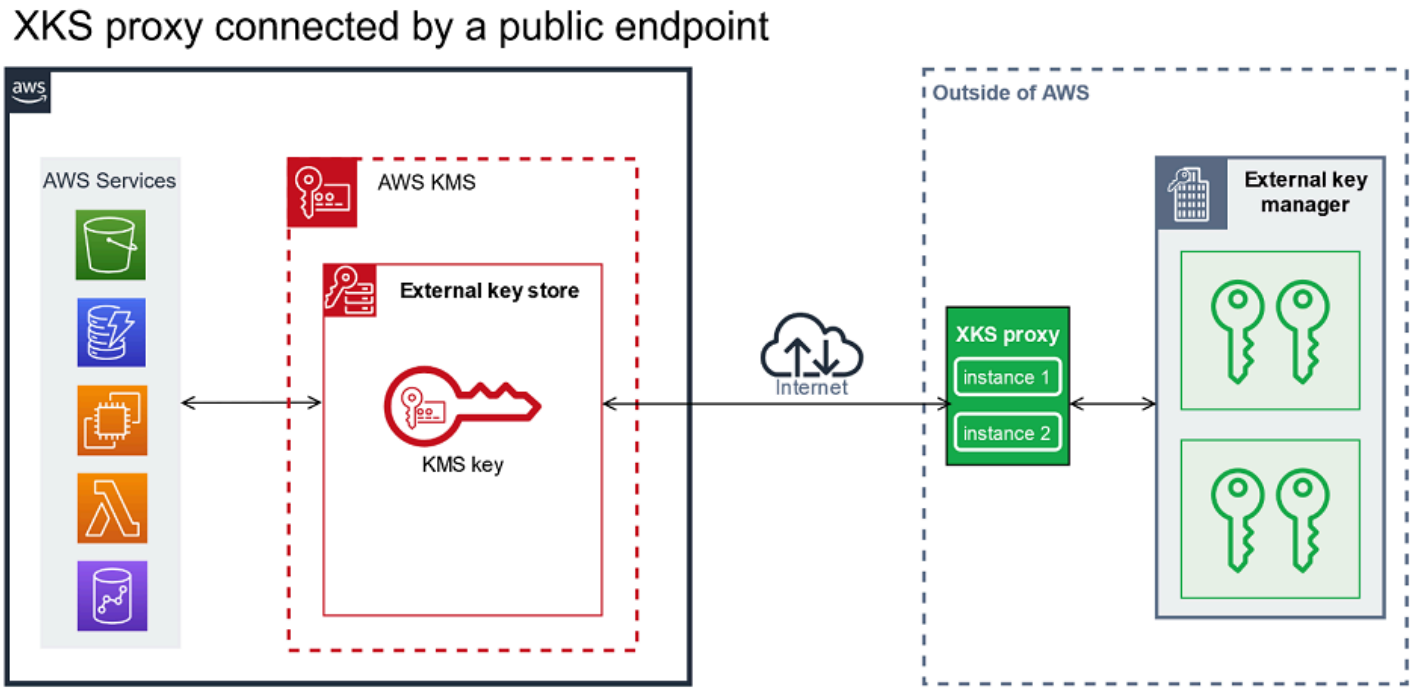
Pertimbangan-pertimbangan

- 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 KMS terhubung 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 sama Wilayah 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 KMS berkomunikasi di port 443 over. IPv4 Nilai ini tidak dapat dikonfigurasi.

Untuk semua persyaratan untuk penyimpanan kunci eksternal, lihat Merakit prasyarat.

Konektivitas layanan titik akhir VPC

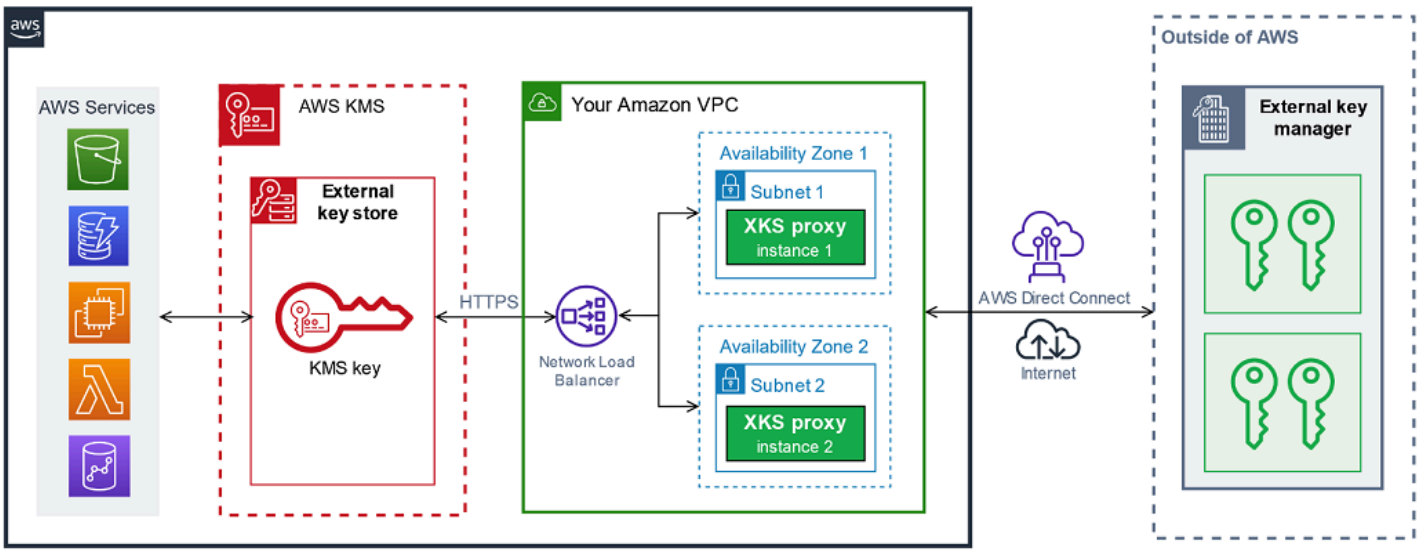
AWS KMS terhubung ke proxy penyimpanan kunci eksternal (proxy XKS) dengan membuat titik akhir antarmuka ke layanan titik akhir VPC Amazon yang Anda buat dan konfigurasikan. Anda bertanggung jawab untuk membuat layanan titik akhir VPC dan menghubungkan VPC Anda ke pengelola kunci eksternal Anda.

Layanan endpoint Anda dapat menggunakan salah satu opsi network-to-AmazonVPC yang didukung untuk komunikasi, termasuk. AWS Direct Connect

Opsi konektivitas ini lebih rumit untuk diatur dan dipelihara. Tetapi menggunakan AWS PrivateLink, yang memungkinkan AWS KMS untuk terhubung secara pribadi ke VPC Amazon Anda dan proxy toko kunci eksternal Anda tanpa menggunakan internet publik.

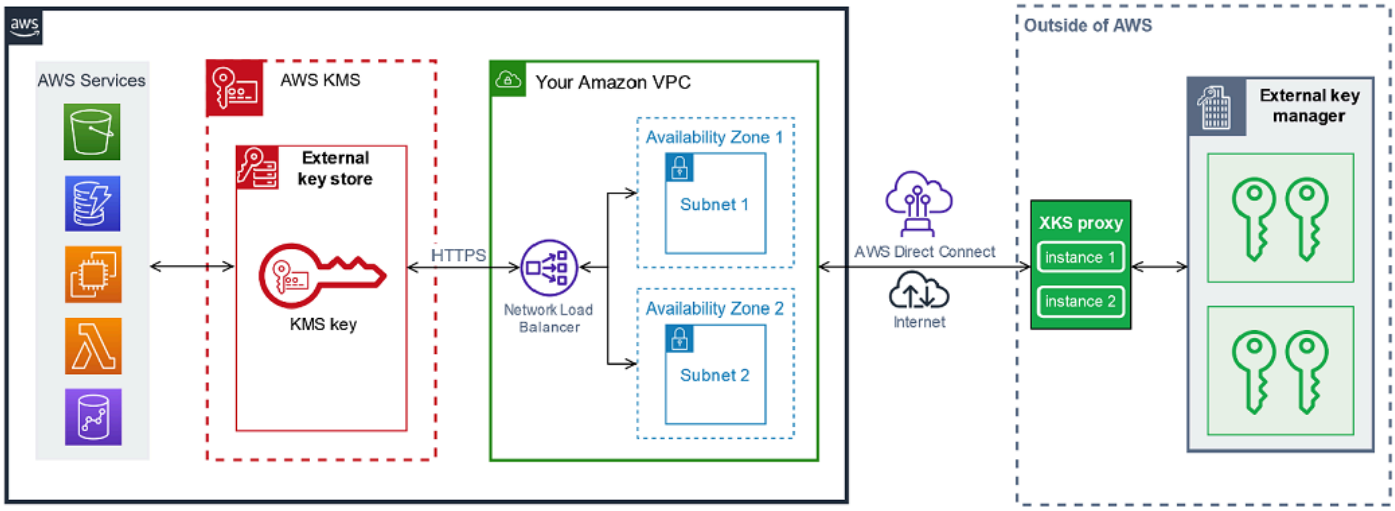
Anda dapat menemukan proxy toko kunci eksternal Anda di VPC Amazon Anda.

XKS proxy hosted in Amazon VPC



Atau, Anda dapat menemukan proxy penyimpanan kunci eksternal di luar AWS Cloud dan menggunakan layanan titik akhir VPC Amazon Anda hanya untuk komunikasi yang aman. AWS KMS

XKS proxy connected via Amazon VPC endpoint service



Anda juga dapat menghubungkan toko kunci eksternal ke layanan endpoint VPC Amazon yang dimiliki oleh orang lain. Akun AWS Keduanya Akun AWS membutuhkan izin yang diperlukan untuk memungkinkan komunikasi antara AWS KMS dan layanan titik akhir VPC

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.

Konfigurasikan 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
- Langkah 1: Buat VPC Amazon dan subnet

- Langkah 2: Buat grup target
- Langkah 3: Buat penyeimbang beban jaringan
- Langkah 4: Buat layanan titik akhir VPC
- Langkah 5: Verifikasi domain nama DNS pribadi Anda
- Langkah 6: Otorisasi AWS KMS untuk terhubung ke layanan titik akhir VPC

Persyaratan untuk konektivitas layanan titik akhir VPC

Jika Anda memilih konektivitas layanan titik akhir VPC untuk penyimpanan kunci eksternal Anda, sumber daya berikut diperlukan.

- 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 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 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.

Untuk semua persyaratan untuk penyimpanan kunci eksternal, lihat Merakit prasyarat.

Langkah 1: Buat VPC Amazon 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

Layanan endpoint Amazon VPC harus memiliki properti berikut untuk bekerja dengan toko kunci eksternal.

- Harus berada di Wilayah yang 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 digunakan 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 Connect VPC Anda ke jaringan lain dan opsi konektivitas Network-to-Amazon VPC. Untuk detail selengkapnya 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
IPv4 Blok CIDR	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 (AZs)	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.
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.

Langkah 2: Buat 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
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
IPv4 alamat	Alamat pribadi proxy toko kunci eksternal Anda
Port	443

Langkah 3: Buat 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.
Protokol	TCP
Port	443
Tindakan default: Teruskan ke	Pilih grup target untuk penyeimbang beban jaringan Anda.

Langkah 4: Buat 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 dapat Akun AWS sama dengan penyimpanan kunci eksternal Anda atau yang lain. Akun AWS

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
Jenis 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. Tidak memerlukan penerimaan. AWS KMS tidak 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 dalamnya Wilayah 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`

Bidang	Nilai
	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` Lihat perinciannya di Kesalahan konfigurasi umum.
Jenis alamat IP yang didukung	IPv4

Langkah 5: Verifikasi 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 PrivateLink mencari 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 Anda AWS 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.

Langkah 6: Otorisasi AWS KMS untuk terhubung ke layanan titik akhir VPC

Lihat prosedur berikut untuk mengelola izin layanan titik akhir Amazon VPC Anda. Setiap langkah tergantung pada konektivitas dan konfigurasi antara penyimpanan kunci eksternal Anda, layanan titik akhir VPC, dan. Akun AWS

Same Akun AWS

Ketika layanan titik akhir VPC Anda dimiliki Akun AWS sama dengan penyimpanan kunci eksternal Anda, 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 PrivateLink Panduan. Gunakan nilai yang diperlukan berikut.

Bidang	Nilai
AWS KMS ARN	`cks.kms.<region>.amazonaws.com` Sebagai contoh, `cks.kms.us-east-1.amazonaws.com` .

Cross Akun AWS

Ketika layanan titik akhir VPC Anda dimiliki oleh yang lain, Akun AWS Anda harus menambahkan keduanya AWS KMS dan akun Anda ke daftar Izinkan prinsipal. Ini memungkinkan AWS KMS

dan penyimpanan kunci eksternal Anda 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. Anda harus menyediakan Akun AWS ARN tempat penyimpanan kunci eksternal berada.

Ikuti petunjuk dalam topik Kelola izin di AWS PrivateLink Panduan. Gunakan nilai yang diperlukan berikut.

Bidang	Nilai
AWS KMS ARN	`cks.kms.<region>.amazonaws.com` Sebagai contoh, `cks.kms.us-east-1.amazonaws.com` .
Akun AWS ARN	`arn:aws:iam:: 111122223333 :role/role_name` Sebagai contoh, `arn:aws:iam:: 123456789012 :role/cks_role`.

Selanjutnya: Buat penyimpanan kunci eksternal

Buat penyimpanan kunci eksternal

Anda dapat membuat satu atau banyak penyimpanan kunci eksternal di masing-masing Akun AWS dan Wilayah. Setiap penyimpanan kunci eksternal harus dikaitkan dengan pengelola kunci eksternal di luar AWS, dan proxy penyimpanan kunci eksternal (proxy XKS) yang memediasi komunikasi antara AWS KMS dan manajer kunci eksternal Anda. Lihat perinciannya di Pilih opsi konektivitas proxy penyimpanan kunci eksternal. Sebelum Anda mulai, kon firmasikan bahwa Anda memerlukan penyimpanan kunci eksternal. Sebagian besar pelanggan dapat menggunakan kunci KMS yang didukung oleh bahan AWS KMS kunci.

> **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 penyimpanan kunci eksternal Anda, Anda perlu menyus un 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 tempat AWS KMS mengirimkan permintaan API ke proxy.

Pertimbangan-pertimbangan

- KMS tidak dapat berkomunikasi melalui IPv6 dengan penyimpanan kunci eksternal.
- Jika Anda menggunakan konektivitas titik akhir publik, pastikan itu AWS KMS dapat berkomunikasi dengan proxy Anda melalui internet menggunakan koneksi HTTPS. Ini termasuk mengonfigurasi TLS pada proxy penyimpanan kunci eksternal dan memastikan bahwa firewall apa pun antara AWS KMS dan proxy memungkinkan lalu lintas IPv4 ke dan dari port 443 pada proxy. Saat membuat penyimpanan kunci eksternal dengan konektivitas titik akhir publik, AWS KMS uji koneksi dengan mengirimkan permintaan status ke proxy penyimpanan kunci eksternal. Pengujian ini memverifikasi bahwa titik akhir dapat dijangkau dan bahwa proxy penyimpanan kunci eksternal Anda akan menerima permintaan yang ditandatangani dengan kredensia 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 KMS menguji konektivitas layanan titik akhir VPC saat Anda menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternalnya.)
- AWS KMS mencatat 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 eksternal Anda. Alarm ini mengingatkan Anda tentang tanda-tanda awal kinerja dan masalah operasional sebelum terjadi. Untuk petunjuk, lihat Pantau toko kunci eksternal.
- Toko kunci eksternal tunduk pada kuota sumber daya. Penggunaan kunci KMS di penyimpanan 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 bekerja. Misalnya, jika Anda membuat proxy penyimpanan kunci eksternal menggunakan AWS sumber daya, pastikan pengoperasian proxy tidak memerlukan ketersediaan kunci KMS di penyimpanan kunci eksternal yang diakses melalui proxy tersebut.

Semua penyimpanan kunci eksternal baru dibuat dalam keadaan terputus. Sebelum Anda dapat membuat kunci KMS penyimpanan kunci eksternal Anda, Anda harus menghubungkannya ke proxy penyimpanan kunci eksternalnya. Untuk mengubah properti penyimpanan kunci eksternal Anda, edit setelan penyimpanan kunci eksternal Anda.

Topik

- Memasang prasyarat
- Buat penyimpanan kunci eksternal baru

Memasang prasyarat

Sebelum 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 penyimpanan kunci eksternal. Selain komponen-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 JSON-based proxy yang menentukan jalur URI proxy dan kreden si 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 Anda.

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 pengelola kunci tunggal, tetapi sebaiknya setidaknya dua instance pengelola kunci terkait yang berbagi kunci kriptografi untuk redundansi. Penyimpanan kunci eksternal tidak memerlukan penggunaan eksklusif manajer kunci eksternal. Namun, pengelola kunci eksternal harus memiliki kapasitas untuk menangani frekuensi permintaan enkripsi dan dekripsi yang diharapkan 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 a Wilayah AWS sehingga waktu pulang-pergi jaringan (RTT) adalah 35 milidetik atau kurang.

Jika proxy penyimpanan kunci eksternal mengizinkannya, Anda dapat mengubah pengelola kunci eksternal yang Anda kaitkan dengan proxy penyimpanan kunci eksternal Anda, tetapi pengelola 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 GetHealthStatus respons dari proxy melaporkan bahwa semua instance pengelola kunci eksternal adalah`Unavailable`, semua upaya untuk membuat penyimpanan kunci eksternal gagal dengan `XksProxyUriUnreachableException`.

Proksi penyimpanan kunci eksternal

Anda harus menentukan proxy penyimpanan kunci eksternal (proxy XKS) yang sesuai dengan persyaratan desain dalam Spesifikasi API Proksi Penyimpanan K AWS KMS unci Eksternal. Anda dapat mengembangkan atau membeli proxy penyimpanan kunci eksternal, atau menggunakan proxy penyimpanan kunci eksternal yang disediakan oleh atau dibangun ke dalam pengelola kunci

eksternal Anda. AWS KMS merekomendasikan 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 a 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 URI dan jalur unik di dalam proxy penyimpanan kunci eksternal untuk permintaannya.

Jika Anda menggunakan konektivitas layanan titik akhir VPC, Anda dapat menemukan proxy penyimpanan kunci eksternal di Amazon VPC Anda, tetapi itu tidak diperlukan. Anda dapat menemukan proxy Anda di luar AWS, 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 kredentif otentikasi proxy penyimpanan kunci eksternal Anda (`XksProxyAuthenticationCredential`).

Anda harus membuat kreden si otentikasi (`XksProxyAuthenticationCredential`) untuk AWS KMS proxy penyimpanan kunci eksternal Anda. AWS KMS mengotentikasi ke proxy Anda dengan menandatangani permintaannya menggunakan proses Sig nature Version 4 (SIGv4) dengan kredentif 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 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.

Kredensia SIGv4 yang digunakan untuk AWS KMS menandatangani permintaan ke proxy penyimpanan kunci eksternal tidak terkait dengan kredenSIAL SIGv4 apa pun yang terkait dengan

prinsipal apa pun di akun AWS Identity and Access Management Anda. AWS Jangan menggunakan kembali kredenSIAL 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 KMS dapat 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 dikonfigurasi dan dipelihara, mungkin tidak memenuhi persyaratan keamanan untuk setiap instalasi. Jika Anda memilih opsi konektivitas layanan titik akhir Amazon VPC, Anda harus membuat dan memelihara komponen yang diperlukan, termasuk Amazon VPC dengan setidaknya dua subnet di dua Zona Ketersediaan 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 ketersediaan 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.

Untuk bantuan memutuskan opsi konektivitas proxy mana yang terbaik untuk penyimpanan kunci eksternal Anda, lihatPilih opsi konektivitas proxy penyimpanan kunci eksternal. Untuk bantuan membuat konektivitas layanan titik akhir VPC yang mengonfigurasi, lihatKonfigurasikan konektivitas layanan titik akhir VPC.

Titik akhir URI proxy

Untuk membuat penyimpanan kunci eksternal, Anda harus menentukan titik akhir (`XksProxyUriEndpoint`) yang AWS KMS digunakan untuk mengirim permintaan ke proxy penyimpanan kunci eksternal.

Protokolnya harus HTTPS. AWS KMS berkomunikasi melalui IPv4 pada 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.

- Konektivitas layanan titik akhir VPC — Tentukan `https://` diikuti oleh nama DNS pribadi dari layanan titik akhir VPC.

Sertifikat server TLS yang dikonfigurasi pada proxy penyimpanan kunci eksternal harus sesuai 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 Te percaya. 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 `myproxy-private.xks.example.com` atau`*.xks.example.com`.

Anda dapat mengubah titik akhir URI proxy, 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

- Gabungan nilai titik akhir URI proxy (`XksProxyUriEndpoint`) dan proxy URI path (`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 mereka 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 seperti penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC yang sama Wilayah AWS, bahkan jika penyimpanan kunci berada di tempat yang 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 Wil Akun AWS ayah dan.

Jalur URI proxy

Untuk membuat penyimpanan kunci eksternal, Anda harus menentukan jalur dasar di proxy penyimpanan kunci eksternal Anda ke API proxy yang diperlukan. Nilai harus dimulai dengan `/` dan harus diakhiri dengan`/kms/xks/v1` di mana `v1` mewakili versi AWS KMS API untuk proxy penyimpanan kunci eksternal. 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 KMS mengirimkan permintaan proxy ke alamat yang ditentukan oleh penggabungan titik akhir URI proxy dan jalur URI proxy. Misalnya, jika titik akhir URI proxy adalah `https://myproxy.xks.example.com` dan jalur URI proxy adalah`/kms/xks/v1`, AWS KMS mengirimkan permintaan API proksi kehttps://myproxy.xks.example.com/kms/xks/v1.

Anda dapat mengubah jalur URI proxy, 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

- Gabungan nilai titik akhir URI proxy (`XksProxyUriEndpoint`) dan proxy URI path (`XksProxyUriPath`) harus unik di Akun AWS dan Region.

Layanan titik akhir VPC

Menentukan nama layanan titik akhir 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 untuk penyimpanan kunci eksternal, lihatKonfigurasikan konektivitas layanan titik akhir VPC.

Layanan titik akhir VPC harus memiliki properti berikut:

- Layanan titik akhir VPC dapat berada di penyimpanan kunci eksternal yang sama atau berbeda Akun AWS .
  - Layanan titik akhir VPC harus berada di tempat yang Wilayah AWS sama dengan penyimpanan kunci eksternal.
  - Anda harus memberikan Akun AWS ID layanan titik akhir VPC jika berada di layanan yang berbeda Akun AWS.
- Itu harus memiliki penyeimbang beban jaringan (NLB) yang terhubung ke setidaknya dua subnet, masing-masing di Zona Ketersediaan yang berbeda.

- Daftar prinsipal yang diizinkan untuk layanan titik akhir VPC harus menyertakan prinsi AWS KMS pal layanan untuk Wilayah:`cks.kms.<region>.amazonaws.com`, seperti. `cks.kms.us-east-1.amazonaws.com`
  - Jika layanan titik akhir Amazon VPC Anda dimiliki oleh layanan Akun AWS lain selain pemilik penyimpanan kunci eksternal (XKS), Anda juga harus mengizinkan akses XKS ke layanan titik akhir VPC. Akun AWS Untuk melakukannya, izinkan daftar Akun AWS ID XKS sebagai prinsipal untuk layanan titik akhir Amazon VPC.
- Itu tidak harus 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 dari domain nama DNS pribadi harus`verified`.
- Sertifikat server TLS yang dikonfigurasi pada proxy penyimpanan kunci eksternal harus menentukan nama host DNS pribadi tempat titik akhir dapat dijangkau.

Persyaratan keunikan

- Penyimpanan kunci eksternal dengan konektivitas titik akhir VPC dapat berbagi`Amazon VPC`, tetapi setiap penyimpanan kunci eksternal harus memiliki layanan titik akhir VPC dan nama DNS pribadinya sendiri.

Berkas konfigurasi proxy

File konfigurasi proxy adalah JSON-based file opsional yang berisi nilai untuk jalur URI proxy dan properti kreden sia 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 menempelkan, 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 UpdateCustomKeyStore operasi CreateCustomKeyStore atau, tetapi Anda dapat menggunakan nilai dalam file konfigurasi proxy untuk memastikan bahwa nilai parameter Anda benar.

Buat penyimpanan kunci eksternal baru

Setelah Anda menyusun prasyarat yang diperlukan, Anda dapat membuat penyimpanan kunci eksternal baru di AWS KMS konsol atau dengan menggunakan CreateCustomKeyStore operasi.

Menggunakan AWS KMS konsol

Sebelum membuat penyimpanan 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 di Konsol Manajemen AWS, Anda dapat mengunggah file konfigurasi JSON-based proxy dengan nilai untuk jalur URI proxy dan kreden tif otentikasi proxy. Beberapa proxy menghasilkan file ini untuk Anda. Itu tidak diperlukan.

1. Masuk ke Konsol Manajemen AWS 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 Peny impanan kunci khusus, Peny impanan kunci eksternal.
4. Pilih Buat penyimpanan kunci eksternal.
5. Masukkan nama yang ramah untuk penyimpanan kunci eksternal. Nama harus unik di antara semua toko kunci eksternal di akun Anda.

   > **Important**
   >
   > Jangan memasukkan informasi rahasia atau sensitif dalam 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, lihatPilih opsi konektivitas proxy penyimpanan kunci eksternal.

   a. Pilih layanan titik akhir Cross-account VPC jika layanan titik akhir VPC Anda berada di tempat yang berbeda. Akun AWS Kemudian masukkan Akun AWS ID untuk pemilik titik akhir VPC di bidang ID akun pemilik layanan titik akhir VPC.

   b. Pilih atau masukkan nama layanan titik akhir VPC untuk penyimpanan kunci eksternal ini. Langkah ini hanya muncul jika 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. Lihat perinciannya di the section called “Memasang prasyarat”.

7. Pilih atau masukkan nama layanan titik akhir VPC untuk penyimpanan kunci eksternal ini. Langkah ini hanya muncul jika 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. Lihat perinciannya di the section called “Memasang prasyarat”.

8. Masukkan titik akhir URI proxy Anda. Protokolnya harus HTTPS. AWS KMS berkomunikasi melalui IPv4 pada 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 awalan jalur URI proxy dan kredentif 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-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 kredensia otentikasi proxy secara manual.

     a. Jika Anda tidak memiliki file konfigurasi proxy, Anda dapat memasukkan URI proxy secara manual. Konsol menyediakan nilaikms/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 URI Proxy. Jika tidak, biarkan bidang kosong.

     b. Jika Anda tidak memiliki file konfigurasi proxy, Anda dapat memasukkan kredensi otentikasi proxy secara manual. Diperlukan ID kunci akses dan kunci akses rahasia.

        - Di Proxy Credential: Access key ID, 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 menyetel atau mengubah kredensi otentikasi yang Anda tetapkan pada proxy penyimpanan kunci eksternal Anda. Itu hanya mengaitkan nilai-nilai ini dengan penyimpanan kunci eksternal Anda. Untuk informasi tentang pengaturan,

perubahan, dan kredensi otentikasi proxy yang berputar, lihat dokumentasi untuk proksi penyimpanan kunci eksternal atau perangkat lunak manajemen kunci.

Jika kredensi otentikasi proxy Anda berubah, edit setelan kredensia untuk penyimpanan kunci eksternal Anda.

10. Pilih Buat penyimpanan kunci eksternal.

Ketika prosedur berhasil, penyimpanan kunci eksternal baru muncul dalam daftar penyimpanan 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 penyimpanan kunci eksternal, Anda harus menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternalnya.

Menggunakan AWS KMS API

Anda dapat menggunakan CreateCustomKeyStore operasi untuk membuat penyimpanan kunci eksternal baru. Untuk bantuan menemukan nilai untuk parameter yang diperlukan, lihat dokumentasi untuk proxy penyimpanan kunci eksternal atau perangkat lunak manajemen kunci Anda.

> **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 penyimpanan kunci eksternal yang unik di akun.

> **Important**
>
> Jangan memasukkan informasi rahasia atau sensitif dalam 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 AWS KMS yang digunakan untuk berkomunikasi dengan proxy penyimpanan kunci eksternal Anda.
- `XksProxyUriPath`— Jalur dalam proxy ke API proxy.
- `XksProxyVpcEndpointServiceName`— Diperlukan hanya jika `XksProxyConnectivity` nilai Anda ada`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-contoh berikut menggunakan nilai fiktif. Sebelum menjalankan perintah, ganti dengan nilai yang valid untuk penyimpanan kunci eksternal Anda.

Buat penyimpanan 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 penyimpanan 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, hubun gkan ke proxy penyimpanan kunci eksternalnya.

Edit 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 JSON-based proxy yang menentukan jalur URI proxy dan 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	✓	

Properti penyimpanan kunci eksternal yang dapat diedit	Status koneksi apa pun	Memerlukan status Terputus
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 (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.		✓

Properti penyimpanan kunci eksternal yang dapat diedit	Status koneksi apa pun	Memerlukan status Terputus
Nama layanan titik akhir VPC proxy () XksProxyVpcEndpointServiceName (Bidang ini diperlukan untuk konektivitas layanan titik akhir VPC)		✓
Pemilik layanan titik akhir VPC () XksProxyVpcEndpointServiceOwner (Bidang ini diperlukan untuk konektivitas layanan titik akhir VPC)		✓

Edit properti toko kunci eksternal Anda

Anda dapat mengedit properti penyimpanan kunci eksternal Anda di AWS KMS konsol atau dengan menggunakan UpdateCustomKeyStoreoperasi.

Menggunakan AWS KMS konsol

Saat Anda mengedit penyimpanan kunci, Anda dapat mengubah salah satu 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 Konsol Manajemen AWS 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 toko kunci yang ingin Anda edit.
    - Jika perlu, lepaskan penyimpanan kunci eksternal dari proxy penyimpanan kunci eksternal. Dari menu Key Store Actions, pilih Disconnect.

5. Dari menu Key store actions, pilih Edit.
6. 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 kredensyal otentikasi proksi. Anda dapat menggunakan file konfigurasi proxy meskipun beberapa nilai yang ditentukan dalam file tidak berubah.
7. Pilih Perbarui toko kunci eksternal.
8. 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.
9. 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.

Menggunakan AWS KMS 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 ke AWS 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
```

Lihat 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 eksternal. 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. Pantau toko kunci eksternal

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 untuk Wilayah 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. Lihat perinciannya di 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 KMS menggunakan 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 dari. 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.

ID pemilik layanan titik akhir VPC

ID layanan endpoint Amazon VPC yang mendukung penyimpanan 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 properti penyimpanan kunci eksternal Anda

Anda dapat melihat penyimpanan kunci eksternal dan properti terkaitnya di AWS KMS konsol atau dengan menggunakan DescribeCustomKeyStoresoperasi.

Menggunakan AWS KMS konsol

Untuk melihat penyimpanan kunci eksternal di akun dan Wilayah tertentu, gunakan prosedur berikut.

1. Masuk ke Konsol Manajemen AWS 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.

Menggunakan AWS KMS 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"
      }
    }
  ]
}
```

Pantau toko kunci eksternal

AWS KMS mengumpulkan 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

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 KMS metrik. 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 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 KMS menyediakan beberapa grafik untuk memantau kesehatan toko kunci eksternal Anda di dalam AWS KMS konsol. Grafik ini secara otomatis dikonfigurasi dan berasal dari AWS KMS metrik.

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 KMS merekomendasikan 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 Dukungan Pusat 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 Buat alarm 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 pengelola kunci eksternal Anda.

AWS KMS mengharapkan 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 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.

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 Buat alarm 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 menunjukkan masalah dengan konfigurasi penyimpanan kunci eksternal Anda. Misalnya, lonjakan

`AuthenticationFailedExceptions` menunjukkan perbedaan antara kredensil otentikasi yang dikonfigurasi AWS KMS dan proxy penyimpanan kunci eksternal. Untuk melihat konfigurasi penyimpanan kunci eksternal Anda, lihatLihat toko kunci eksternal. Untuk mengedit pengaturan penyimpanan kunci eksternal Anda, lihatEdit 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 KMS operasi kriptografi mengembalikan `KMSInvalidStateException` untuk semua kegagalan yang terkait dengan konfigurasi eksternal atau status koneksi 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 ulang	**`AccessDeniedException`** Untuk bantuan penyelesaian masalah, lihat Masalah otorisasi proxy.	**`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Tidak dapat dicoba ulang	**`AuthenticationFailedException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan kredensi otentikasi.	**`XksProxyIncorrectAuthenticationCredentialException`** dalam menanggapi `CreateCustomKeyStore` dan `UpdateCustomKeyStore` operasi. **`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi.

Jenis kesalahan	Pengecualian ditampilkan dalam grafik	Pengecualian yang AWS KMS kembali kepada Anda
		`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.
Dicoba ulang	**`InternalException`** Proxy penyimpanan kunci eksternal menolak permintaan karena tidak dapat berkomunikasi dengan pengelola 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.

Jenis kesalahan	Pengecualian ditampilkan dalam grafik	Pengecualian yang AWS KMS kembali kepada Anda
Tidak dapat dicoba ulang	**`InvalidCiphertextException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan dekripsi.	**`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Tidak dapat dicoba ulang	**`InvalidKeyUsageException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan operasi kriptografi untuk kunci eksternal.	**`XksKeyInvalidConfigurationException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Tidak dapat dicoba ulang	**`InvalidStateException`** 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 ulang	**`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 ulang	**`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 ulang	**`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 ulang	**`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 Buat alarm 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. Lihat informasi yang lebih lengkap di Buat alarm untuk kedaluwarsa sertifikat.

Connect dan lepaskan 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 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.

> **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. Lihat perinciannya di Pantau toko kunci eksternal.
> Manajer kunci eksternal Anda mungkin menyediakan metode tambahan untuk menghentikan dan memulai kembali komunikasi antara penyimpanan kunci AWS KMS eksternal Anda 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

- Status koneksi
- Connect toko kunci eksternal
- Putuskan sambungan penyimpanan kunci eksternal

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 Lihat toko AWS CloudHSM kunci dan Lihat 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.

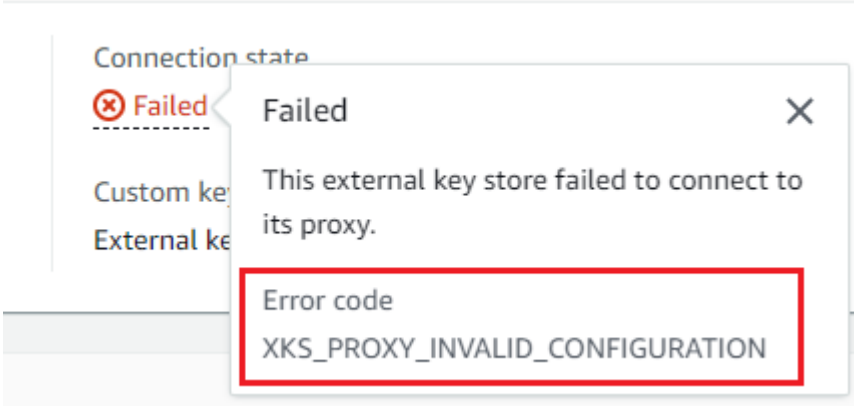
  Toko kunci pendukung untuk toko AWS CloudHSM kunci adalah AWS CloudHSM cluster terkaitnya. Penyimpanan kunci pendukung untuk penyimpanan kunci eksternal adalah proxy penyimpanan kunci eksternal dan manajer kunci eksternal yang didukungnya.

  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. Lihat perinciannya di Pantau 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.



Connect 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.

Connect dan sambungkan kembali ke toko kunci eksternal Anda

Anda dapat menghubungkan, atau menghubungkan kembali, penyimpanan kunci eksternal Anda di AWS KMS konsol atau dengan menggunakan ConnectCustomKeyStoreoperasi.

Menggunakan AWS KMS konsol

Anda dapat menggunakan AWS KMS konsol untuk menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternal.

1. Masuk ke Konsol Manajemen AWS 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.

Menggunakan AWS KMS API

Untuk menghubungkan toko kunci eksternal yang terputus, gunakan ConnectCustomKeyStoreoperasi.

Sebelum menghubungkan, status koneksi dari toko kunci eksternal harus`DISCONNECTED`. Jika keadaan koneksi saat ini`FAILED`, lepaskan penyimpanan kunci eksternal, lalu sambungkan.

Proses koneksi memakan waktu sekitar lima menit untuk diselesaikan. 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
```

`ConnectCustomKeyStore`Operasi 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. `ConnectionState`Nilai `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

Ketika Anda memutuskan penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC dari proxy penyimpanan kunci eksternal AWS 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 penyimpanan 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. Lihat perinciannya di Edit 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 sambungan penyimpanan kunci eksternal sebelum Anda mencoba menghubungkannya lagi.

Putuskan sambungan toko kunci eksternal Anda

Anda dapat memutuskan sambungan penyimpanan kunci eksternal Anda di AWS KMS konsol atau dengan menggunakan DisconnectCustomKeyStoreoperasi.

Menggunakan AWS KMS 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 Konsol Manajemen AWS 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.

Menggunakan AWS KMS 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"
      }
    }
  ]
}
```

Hapus penyimpanan kunci eksternal

Ketika Anda menghapus penyimpanan kunci eksternal, AWS KMS menghapus semua metadata tentang penyimpanan kunci eksternal dari AWS 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 KMS termasuk 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.

Anda dapat menghapus toko kunci eksternal Anda di AWS KMS konsol atau dengan menggunakan DeleteCustomKeyStoreoperasi.

Menggunakan AWS KMS konsol

Anda dapat menggunakan AWS KMS konsol untuk menghapus toko kunci eksternal.

1. Masuk ke Konsol Manajemen AWS 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.

Menggunakan AWS KMS 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:
```

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, tumpukan IP, 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 SDKs. Bahkan jika Anda menggunakan AWS KMS konsol, Anda mungkin menemukan referensi ini berguna. Misalnya, lihat daftar Kesalahan untuk `CreateCustomKeyStores` operasi.

Untuk mengoptimalkan kinerja proxy penyimpanan kunci eksternal Anda, AWS KMS mengembalikan pengecualian berdasarkan keandalan proxy Anda dalam periode agregasi tertentu selama 5 menit. Jika terjadi Kesalahan Server Internal 500, Layanan 503 Tidak Tersedia, atau batas waktu koneksi, proxy dengan keandalan tinggi mengembalikan `KMSInternalException` dan memicu percobaan ulang otomatis untuk memastikan bahwa permintaan akhirnya berhasil. Namun, proxy dengan keandalan rendah kembali`KMSInvalidStateException`. Untuk informasi selengkapnya, lihat Memantau penyimpanan kunci eksternal.

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. Pantau 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. Lihat perinciannya di Lihat 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, lihatEdit 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 eksternal

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 proxy APIs. 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, lihatEdit 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 dukungan CAs, 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 Lihat 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 Konfigurasikan konektivitas layanan titik akhir VPC dan Buat penyimpanan 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, lihatLangkah 5: Verifikasi 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.

- Pastikan bahwa setiap firewall antara AWS KMS dan proxy penyimpanan kunci eksternal memungkinkan lalu lintas ke dan dari port 443 pada proxy. AWS KMS berkomunikasi di port 443 over. IPv4 Nilai ini tidak dapat dikonfigurasi.
- 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, lihatLihat toko kunci eksternal. Untuk mengubah nilai titik akhir URI proxy, lihatEdit 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**
>
> Ketika status koneksi penyimpanan kunci kustom`FAILED`, Anda harus memutuskan penyimpanan kunci khusus sebelum mencoba menghubungkannya kembali. Anda tidak dapat menghubungkan penyimpanan kunci kustom dengan status koneksi `FAILED`.

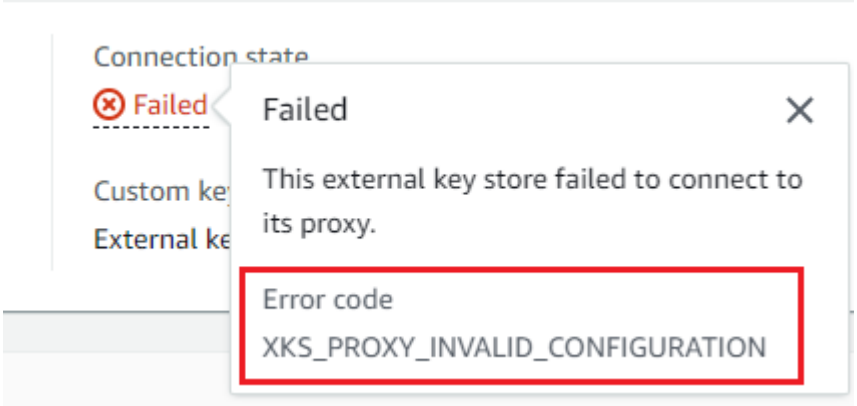
Untuk melihat status koneksi penyimpanan kunci eksternal:

- 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, lihatEdit 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 over. IPv4 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. 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, lihatNegara-negara kunci AWS KMS keys. 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 Identifikasi dan lihat 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 Aktifkan dan nonaktifkan 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 Jadwalkan 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 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 IDs 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.
- 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. Spesifikasi kunci dari kunci eksternal yang ditentukan adalah*<key-spec>* .

- 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 administrator proxy penyimpanan kunci eksternal Anda.

- AWS KMS operasi menghasilkan pengecualian ini ketika proxy mengembalikan respons tidak terdefinisi 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 proxy yang APIs 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.

Keamanan AWS Key Management Service

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk AWS Key Management Service (AWS KMS), lihat AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Selain konfigurasi dan penggunaan Anda AWS 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. Ini menunjukkan kepada Anda cara mengonfigurasi AWS KMS untuk memenuhi tujuan keamanan dan kepatuhan Anda.

Topik

- Perlindungan data di AWS Key Management Service
- Identitas dan manajemen akses untuk AWS Key Management Service
- Penebangan dan pemantauan di AWS Key Management Service
- Validasi kepatuhan untuk AWS Key Management Service
- Ketahanan di AWS Key Management Service
- Keamanan infrastruktur di AWS Key Management Service

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
- Enkripsi data
- Privasi lalu lintas antar jaringan

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 luar AWS KMS.

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 140-3 Security Level 3—modul keamanan perangkat keras yang divalidasi (). HSMs 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(HSMs Yang AWS KMS digunakan di Wilayah Tiongkok disertifikasi oleh OSCCA dan mematuhi semua peraturan Tiongkok terkait, tetapi tidak divalidasi berdasarkan Program Validasi Modul Kriptografi FIPS 140-3.)

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. HSMs 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.)

AWS menegaskan 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 AWS KMS FIPS 140-3 yang divalidasi HSMs 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 lengkap dan eksklusif atas kebijakan otorisasi yang mengontrol akses ke kunci. Untuk Kunci 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 KMS memberikan alternatif untuk bahan utama yang dihasilkan di AWS 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. Ini HSMs disertifikasi di FIPS 140-2 Security Level 3 atau 140-3 Security Level 3. Kunci KMS di toko kunci eksternal didukung oleh kunci di manajer kunci eksternal yang Anda kontrol dan kelola di luar AWS, 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 perjalanan AWS 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, bahan AWS KMS kunci impor plaintext tidak pernah meninggalkan yang tidak terenkripsi. HSMs Namun, pelanggan yang menyediakan bahan utama bertanggung jawab atas penggunaan yang aman, daya tahan, dan pemeliharaan material utama di luar AWS KMS.

Enkripsi data

Data AWS KMS terdiri dari AWS KMS keys dan bahan kunci enkripsi yang mereka wakili. Materi kunci ini ada dalam teks biasa hanya dalam modul keamanan AWS KMS perangkat keras (HSMs) 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 yang tidak terenkripsi. AWS KMS HSMs Itu tidak diekspor atau ditransmisikan dalam operasi AWS KMS API 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 saat diam
- Enkripsi saat bergerak

Enkripsi saat diam

AWS KMS menghasilkan materi utama untuk modul AWS KMS keys keamanan perangkat keras FIPS 140-3 Security Level 3 yang sesuai (). HSMs Satu-satunya pengecualian adalah Wilayah Tiongkok, di mana HSMs AWS KMS yang digunakan untuk menghasilkan kunci KMS mematuhi semua peraturan China terkait, tetapi tidak divalidasi berdasarkan Program Validasi Modul Kriptografi FIPS 140-3. 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 HSMs 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 saat bergerak

Materi kunci yang AWS KMS dihasilkan untuk kunci KMS tidak pernah diekspor atau ditransmisikan dalam operasi AWS KMS API. AWS KMS menggunakan 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 AWS KMS API 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 KMS membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3 di semua

wilayah. AWS KMS juga mendukung TLS pasca-kuantum hibrida untuk titik akhir AWS KMS layanan di semua wilayah, kecuali Wilayah Tiongkok. AWS KMS tidak 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 tervalidasi FIPS 140-3 saat 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) 140-3. Untuk daftar titik akhir AWS KMS FIPS, lihat AWS Key Management Service titik akhir dan kuota di. Referensi Umum AWS

Komunikasi antara host AWS KMS layanan dan HSMs dilindungi menggunakan Elliptic Curve Cryptography (ECC) dan Advanced Encryption Standard (AES) dalam skema enkripsi yang diautentikasi. Untuk detail selengkapnya, lihat Keamanan komunikasi internal di Detail AWS Key Management Service Kriptografi.

Privasi lalu lintas antar jaringan

AWS KMS mendukung satu Konsol Manajemen AWS 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 pribadi Anda ke AWS.

- Koneksi IPSec VPN melalui internet
- AWS Direct Connect, yang menghubungkan jaringan internal Anda ke Direct Connect lokasi 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 (HSMs) 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 mengirim lalu lintas melalui internet publik, gunakan titik akhir VPC, yang didukung oleh. AWS PrivateLink Untuk informasi selengkapnya, lihat Connect ke AWS KMS melalui titik akhir VPC.

AWS KMS juga mendukung opsi pertukaran kunci pasca-kuantum hibrida untuk protokol enkripsi jaringan Transport Layer Security (TLS). Anda dapat menggunakan opsi ini dengan TLS saat Anda terhubung ke titik akhir AWS KMS API.

Identitas dan manajemen akses 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 di AWS 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 Akses dan izin kunci 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.

Topik

- AWS kebijakan yang dikelola untuk AWS Key Management Service
- Menggunakan peran terkait layanan untuk AWS KMS

AWS kebijakan yang dikelola untuk AWS Key Management Service

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS kebijakan terkelola dirancang untuk memberikan izin untuk banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan ter AWS kelola mungkin tidak memberikan izin hak istimewa terkecil untuk kasus penggunaan spesifik Anda karena kebijakan tersebut tersedia untuk digunakan semua AWS pelanggan. Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan ter AWS kelola ketika yang baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

AWS kebijakan yang dikelola: AWSKeyManagementServicePowerUser

Anda dapat melampirkan kebijakan `AWSKeyManagementServicePowerUser` ke identitas IAM Anda.

Anda dapat menggunakan kebijakan ter `AWSKeyManagementServicePowerUser` kelola untuk memberikan izin pengguna yang kuat kepada prinsipal IAM di akun Anda. Pengguna canggih dapat membuat kunci KMS, menggunakan dan mengelola kunci KMS yang mereka buat, dan melihat semua kunci KMS dan identitas IAM. Kepala sekolah yang memiliki kebijakan ter `AWSKeyManagementServicePowerUser` kelola juga dapat memperoleh izin dari sumber lain, termasuk kebijakan utama, kebijakan IAM lainnya, dan hibah.

`AWSKeyManagementServicePowerUser`adalah kebijakan IAM yang AWS dikelola. Untuk informasi selengkapnya tentang AWS kebijakan terkelola, AWS lihat kebijakan ter kelola di Panduan Pengguna IAM.

> **Note**
>
> Izin dalam kebijakan ini yang khusus untuk kunci KMS, seperti `kms:TagResource` dan`kms:GetKeyRotationStatus`, hanya berlaku jika kebijakan kunci untuk kunci KMS tersebut secara eksplisit mengizinkan penggunaan 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 Res ources.
> Kebijakan ini memberikan izin pengguna yang kuat pada kunci KMS apa pun dengan kebijakan kunci yang mengizinkan operasi. 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 Cross-account penggunaan.

Untuk memungkinkan kepala sekolah melihat AWS KMS konsol tanpa kesalahan, kepala sekolah memerlukan tag: GetResources izin, yang tidak termasuk dalam `AWSKeyManagementServicePowerUser` kebijakan. Anda dapat mengizinkan izin ini dalam kebijakan IAM terpisah.

Kebijakan AWSKeyManagementServicePowerUser IAM terkelola mencakup izin berikut.

- Memungkinkan kepala sekolah untuk membuat kunci KMS. Karena proses ini mencakup pengaturan kebijakan kunci, pengguna tingkat lanjut dapat memberikan izin kepada diri mereka sendiri dan orang lain untuk menggunakan dan mengelola kunci KMS yang mereka buat.
- Memungkinkan kepala sekolah 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 kepala sekolah untuk mendapatkan informasi terperinci tentang semua kunci KMS, termasuk kunci ARN mereka, konfigurasi kriptografi, kebijakan kunci, alias, tag, dan status rotasi. ???
- Memungkinkan kepala sekolah untuk mencantumkan pengguna, grup, dan peran IAM.
- Kebijakan ini tidak mengizinkan prinsipal 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.

Untuk melihat izin kebijakan ini, lihat AWSKeyManagementServicePowerUser di Referensi Kebijakan AWS Terkelola.

AWS kebijakan yang dikelola: AWSServiceRoleForKeyManagementServiceCustomKeyStores

Anda tidak dapat melampirkan `AWSServiceRoleForKeyManagementServiceCustomKeyStores` ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memberikan AWS KMS izin untuk melihat cluster yang terkait dengan penyimpanan AWS CloudHSM kunci Anda dan membuat jaringan untuk mendukung koneksi antara penyimpanan kunci kustom Anda dan AWS CloudHSM klasternya AWS CloudHSM . Untuk informasi selengkapnya, lihat Otorisasi AWS KMS untuk mengelola AWS CloudHSM dan sumber daya Amazon EC2.

AWS kebijakan yang dikelola: AWSServiceRoleForKeyManagementServiceMultiRegionKeys

Anda tidak dapat melampirkan `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memberikan AWS KMS izin untuk menyinkronkan perubahan apa pun pada materi kunci kunci utama Multi-wilayah ke kunci replika. Untuk informasi selengkapnya, lihat Otorisasi AWS KMS untuk menyinkronkan kunci Multi-region.

AWS KMS update ke AWS kebijakan terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola AWS KMS sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan ke umpan RSS pada halaman AWS KMS Riwayat dokumen.

Perubahan	Deskripsi	Date
AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy – Pembaruan ke kebijakan yang sudah ada	AWS KMS menambahkan kolom pernyataan ID (`Sid`) ke kebijakan terkelola dalam versi kebijakan v2.	November 21, 2024
AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy – Pembaruan ke kebijakan yang sudah ada	AWS KMS menambahk an`ec2:Descr ibeVpcs` ,`ec2:Descr ibeNetworkAcls` , dan `ec2:DescribeNetwor kInterfaces` izin untuk memantau perubahan di VPC yang berisi AWS CloudHSM cluster Anda sehingga AWS KMS dapat memberikan pesan kesalahan yang jelas jika terjadi kegagalan.	10 November 2023
AWS KMS mulai melacak perubahan	AWS KMS mulai melacak perubahan untuk kebijakan yang AWS dikelolanya.	10 November 2023

Menggunakan peran terkait layanan untuk AWS KMS

AWS Key Management Service menggunakan AWS Identity and Access Management peran terkait layanan (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke. AWS KMS Peran terkait layanan ditentukan oleh AWS KMS dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Peran terkait layanan membuat pengaturan AWS KMS lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. AWS KMS mendefinisikan izin peran terkait layanan, 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 tertaut layanan hanya setelah menghapus sumber daya terkait terlebih dahulu. Ini melindungi AWS KMS sumber daya Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat Layanan AWS yang bisa digunakan dengan IAM dan carilah layanan yang memiliki opsi Ya di kolom Peran Terkait Layanan. Pilih Ya dengan sebuah tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Untuk melihat detail tentang pembaruan pada peran terkait layanan yang dibahas dalam topik ini, lihat. AWS KMS update ke AWS kebijakan terkelola

Topik

- Otorisasi AWS KMS untuk mengelola AWS CloudHSM dan sumber daya Amazon EC2
- Otorisasi AWS KMS untuk menyinkronkan kunci Multi-region

Otorisasi AWS KMS untuk mengelola AWS CloudHSM dan sumber daya 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.

Untuk melihat detail tentang pembaruan kebijakan AWSKeyManagementServiceCustomKeyStoresServiceRolePolicyterkelola, lihatAWS KMS update ke AWS kebijakan terkelola.

Topik

- Tentang peran AWS KMS terkait layanan
- Membuat peran terkait layanan
- Mengedit deskripsi peran tertaut layanan
- Menghapus peran tertaut layanan

Tentang peran AWS KMS terkait layanan

Peran terkait layanan adalah peran IAM yang memberikan satu izin AWS layanan untuk memanggil AWS layanan lain atas nama Anda. Ini dirancang untuk memudahkan Anda menggunakan fitur dari beberapa AWS layanan terintegrasi tanpa harus membuat dan memelihara kebijakan IAM yang kompleks. Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk AWS KMS.

Untuk penyimpanan AWS CloudHSM utama, AWS KMS buat peran AWSServiceRoleForKeyManagementServiceCustomKeyStoresterkait layanan dengan kebijakan AWSKeyManagementServiceCustomKeyStoresServiceRolePolicyterkelola. 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 jaringan ACLs untuk VPC yang berisi cluster AWS CloudHSM Anda sehingga dapat memberikan pesan kesalahan AWS KMS 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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudhsm:Describe*",
        "ec2:CreateNetworkInterface",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeSecurityGroups",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:DeleteSecurityGroup",
        "ec2:DescribeVpcs",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
```

```
    }
  ]
}
```

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. Itu tidak memberikan AWS KMS izin tambahan. Misalnya, AWS KMS tidak memiliki izin untuk membuat, mengelola, atau menghapus AWS CloudHSM cluster, HSMs, atau backup Anda.

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 Service Titik 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 terkait layanan

AWS KMS secara 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 KMS tidak 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

Otorisasi AWS KMS untuk menyinkronkan kunci Multi-region

Untuk mendukung kunci Multi-region, AWS KMS perlu izin untuk menyinkronkan properti bersama dari kunci utama Multi-region dengan kunci replika. Untuk mendapatkan izin ini, AWS KMS buat peran AWSServiceRoleForKeyManagementServiceMultiRegionKeysterkait layanan di Anda. Akun AWS Pengguna yang membuat kunci Multi-wilayah harus memiliki `iam:CreateServiceLinkedRole` izin yang memungkinkan mereka membuat peran terkait layanan.

Anda dapat melihat SynchronizeMultiRegionKey CloudTrail peristiwa yang merekam AWS KMS sinkronisasi properti bersama di AWS CloudTrail log Anda.

Untuk melihat detail tentang pembaruan kebijakan AWSKeyManagementServiceMultiRegionKeysServiceRolePolicyterkelola, lihatAWS KMS update ke AWS kebijakan terkelola.

Topik

- Tentang peran yang tertaut ke layanan untuk kunci multi-Wilayah
- Membuat peran terkait layanan
- Mengedit deskripsi peran tertaut layanan
- Menghapus peran tertaut layanan

Tentang peran yang tertaut ke layanan untuk kunci multi-Wilayah

Peran terkait layanan adalah peran IAM yang memberikan satu izin AWS layanan untuk memanggil AWS layanan lain atas nama Anda. Ini dirancang untuk memudahkan Anda menggunakan fitur dari beberapa AWS layanan terintegrasi tanpa harus membuat dan memelihara kebijakan IAM yang kompleks.

Untuk kunci Multi-wilayah, AWS KMS buat peran AWSServiceRoleForKeyManagementServiceMultiRegionKeysterkait layanan dengan kebijakan

terkelola. 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 AWS KMS perlu menyinkronkan properti bersama Multi-region. Itu tidak memberikan AWS KMS izin tambahan. Misalnya, AWS KMS tidak memiliki izin untuk membuat, mereplikasi, atau menghapus kunci KMS apa pun.

Untuk informasi selengkapnya tentang cara AWS layanan menggunakan peran terkait layanan, lihat Menggunakan Peran Tertaut Layanan di Panduan Pengguna IAM.

JSON

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

Membuat peran terkait layanan

AWS KMS secara 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 KMS tidak 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.

Penebangan dan pemantauan di AWS Key Management Service

Pemantauan adalah bagian penting untuk memahami ketersediaan, keadaan, dan penggunaan situs Anda AWS KMS keys AWS KMS. Pemantauan membantu menjaga keamanan, keandalan, ketersediaan, dan kinerja AWS solusi Anda. AWS menyediakan beberapa alat untuk memantau kunci KMS Anda.

AWS CloudTrail Log

Setiap panggilan ke operasi AWS KMS API ditangkap sebagai peristiwa dalam AWS CloudTrail log. Log ini merekam semua panggilan API dari AWS KMS konsol, dan panggilan yang dilakukan oleh AWS KMS dan AWS layanan lainnya. Panggilan API lintas akun, seperti panggilan untuk menggunakan kunci KMS secara berbeda Akun 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

Pantau, simpan, dan akses file log Anda dari AWS CloudTrail dan sumber lain. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

Untuk AWS 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 “Tombol monitor dengan CloudWatch”.

Amazon EventBridge

AWS KMS menghasilkan EventBridge peristiwa ketika kunci KMS Anda diputar atau dihapus atau materi kunci yang diimpor dalam kunci KMS Anda kedaluwarsa. Cari AWS KMS peristiwa (operasi API) dan arahkan ke satu atau beberapa fungsi atau aliran target untuk menangkap informasi status. Untuk informasi selengkapnya, lihat the section called “Tombol monitor dengan Amazon EventBridge” 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 CSPM

Anda dapat memantau AWS KMS penggunaan Anda untuk standar industri keamanan dan praktik terbaik kepatuhan menggunakan AWS Security Hub CSPM. Security Hub CSPM menggunakan kontrol keamanan untuk mengevaluasi konfigurasi sumber daya dan standar keamanan untuk membantu Anda mematuhi berbagai kerangka kerja kepatuhan. Untuk informasi selengkapnya, lihat AWS Key Management Service kontrol di Panduan AWS Security Hub Pengguna.

Validasi kepatuhan untuk AWS Key Management Service

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS Key Management Service sebagai bagian dari beberapa program AWS kepatuhan. Program ini mencakup SOC, PCI, FedRAMP, HIPAA, dan lainnya.

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 AWS Artifact.

Pelajari selengkapnya

Tanggung jawab kepatuhan Anda saat menggunakan AWS KMS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. Jika penggunaan AWS KMS Anda tunduk pada kepatuhan dengan standar yang dipublikasikan, AWS sediakan sumber daya untuk membantu:

- AWS Layanan dalam Lingkup oleh Program Kepatuhan - Halaman ini mencantumkan AWS layanan yang berada dalam lingkup program kepatuhan tertentu. Untuk informasi umum, lihat Program Kepatuhan AWS.

- Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan. AWS
- AWS Sumber Daya Kepatuhan — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Config AWS Layanan ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub CSPM— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub CSPM menggunakan kontrol keamanan untuk mengevaluasi AWS sumber daya Anda dan untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk mengetahui daftar layanan dan kontrol yang didukung, lihat Referensi kontrol CSPM Security Hub.

Ketahanan di AWS Key Management Service

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang secara otomatis melakukan failover di antara Zona Ketersediaan tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur biasa yang terdiri dari satu atau beberapa pusat data.

Selain infrastruktur AWS global, AWS KMS menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda. Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur AWS Global.

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 KMS dirancang untuk memastikan nol waktu henti yang direncanakan, dengan semua pembaruan perangkat lunak dan operasi penskalaan dilakukan dengan mulus dan tanpa terasa.

AWS KMS Service Level Agreement (SLA) mencakup komitmen layanan sebesar 99,999% untuk semua KMS. APIs 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 KMS Infrastruktur direplikasi di setidaknya tiga Availability Zones (AZs) di setiap Region. Untuk memastikan bahwa beberapa kegagalan host tidak mempengaruhi AWS KMS kinerja, AWS KMS dirancang untuk melayani lalu lintas pelanggan dari salah satu AZs di 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 (HSMs), 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 HSMs; itu tidak pernah ditulis ke penyimpanan persisten.

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 seperti AWS 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 di AWS Key Management Service

Sebagai layanan terkelola, AWS Key Management Service (AWS KMS) dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam Amazon Web Services: Tinjauan 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 KMS membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3 di semua wilayah. AWS KMS juga mendukung TLS pasca-kuantum hibrida untuk titik akhir AWS KMS layanan di semua wilayah, kecuali Wilayah Tiongkok. AWS KMS tidak 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. Lihat perinciannya di 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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"AWS":
    "arn:aws:iam::111122223333:role/KMSTestRole"},
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
      "NotIpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24",
          "203.0.113.0/24"
        ]
      }
    }
  }
}
```

Isolasi pada Host Fisik

Keamanan infrastruktur fisik yang AWS KMS digunakan tunduk pada kontrol yang dijelaskan di bagian Keamanan Fisik dan Lingkungan dari Amazon Web Services: Tinjauan 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 (HSMs) yang dirancang dengan kontrol khusus untuk menahan serangan fisik. Itu HSMs adalah perangkat fisik yang tidak memiliki lapisan virtualisasi, seperti hypervisor, yang berbagi perangkat fisik di antara beberapa penyewa logis. Bahan kunci untuk AWS KMS keys disimpan hanya dalam memori volatil pada HSMs, 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 rinci tentang pengoperasian AWS KMS HSMs, lihat Detail AWS Key Management Service Kriptografi.

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 untuk kuota sumber daya rotasi sesuai permintaan dan kuota permintaan toko AWS CloudHSM utama. 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 Dukungan Center dan buat kasus.

Topik

- Kuota sumber daya
- Kuota permintaan
- Permintaan pelambatan AWS KMS

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 untuk 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 Dukungan 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
Kuota sumber daya toko kunci kustom	10	Akun AWS dan Wilayah	Ya
On-demand rotasi	25	Kunci yang dikelola pelanggan	Tidak

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.

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.

On-demand rotasi: 25

Anda dapat melakukan rotasi kunci sesuai permintaan maksimum 25 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 Dukungan 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. Saat Anda melebihi kuota permintaan API AWS KMS , batasi permintaan.

Semua kuota AWS KMS permintaan dapat disesuaikan, kecuali untuk 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 Kuota Layanan, atau untuk mengubah kuota di Wilayah AWS mana Kuota Layanan tidak AWS KMS tersedia, silakan kunjungi AWS Dukungan Pusat dan buat kasus.

Jika Anda melebihi kuota permintaan untuk GenerateDataKey operasi, pertimbangkan untuk menggunakan fitur caching kunci data dari AWS Encryption SDK. Menggunakan kembali kunci data dapat mengurangi frekuensi permintaan Anda. AWS KMS

Selain permintaan kuota, AWS KMS menggunakan kuota sumber daya untuk memastikan kapasitas untuk semua pengguna. Lihat perinciannya di Kuota sumber daya.

Untuk melihat tren tarif permintaan Anda, gunakan konsol Kuota Layanan. 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 menggunakan Kuota Layanan dan Amazon CloudWatch di Blog AWS Keamanan.

Topik

- Minta kuota untuk masing-masing AWS KMS Operasi API
- Menerapkan kuota permintaan
- Kuota bersama untuk operasi kriptografis
- Permintaan API yang dibuat atas nama Anda
- Cross-account permintaan
- Kuota permintaan penyimpanan kunci khusus

Minta kuota untuk masing-masing AWS KMS Operasi API

Tabel ini mencantumkan kode ku ota Kuota Layanan dan nilai default untuk setiap kuota AWS KMS permintaan. Semua kuota AWS KMS permintaan dapat disesuaikan, kecuali untuk kuota permintaan penyimpanan 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`
• `Encrypt`
• `GenerateDataKey`
• `GenerateDataKeyWithoutPlaintext`
• `GenerateMac`
• `GenerateRandom`
• `ReEncrypt`
• `VerifyMac`
	Kuota bersama ini bervariasi dengan Wilayah AWS dan jenis kunci KMS yang digunakan dalam permintaan. Setiap kuota dihitung secara terpisah.
	
	• 10.000 (dibagikan)
	• 20.000 (dibagi) 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
	  • Eropa (Spanyol), eu-selatan-2
	• 100.000 (dibagikan) 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`	1.000 (bersama) untuk kunci RSA KMS

Nama kuota	Nilai default (permintaan per detik)
Berlaku untuk: • `Decrypt` • `Encrypt` • `ReEncrypt` • `Sign` • `Verify`	
`Cryptographic operations (ML-DSA) request rate` Berlaku untuk: • `Sign` • `Verify`	1.000 (bersama) untuk kunci ML-DSA KMS
`Cryptographic operations (ECC and SM2) request rate` Berlaku untuk: • `Decrypt`—hanya didukung untuk kunci KMS SM2 (hanya Wilayah China) • `DeriveSharedSecret` • `Encrypt`—hanya didukung untuk kunci KMS SM2 (hanya Wilayah China) • `ReEncrypt` —hanya didukung untuk kunci KMS SM2 (hanya Wilayah China) • `Sign` • `Verify`	1.000 (bersama) untuk kunci KMS kurva elips (ECC) dan SM2 (hanya Wilayah China)

Nama kuota	Nilai default (permintaan per detik)
`Custom key store request quotas` Berlaku untuk: • `Decrypt` • `DeriveSharedSecret` • `Encrypt` • `GenerateDataKey` • `GenerateDataKeyWithoutPlaintext` • `GenerateRandom` • `ReEncrypt`	Kuota permintaan penyimpanan kunci khusus dihitung secara terpisah untuk setiap penyimpanan kunci khusus • 1.800 (dibagikan) untuk setiap toko AWS CloudHSM kunci • 1.800 (bersama) untuk setiap penyimpanan kunci eksternal
`CancelKeyDeletion request rate`	5
`ConnectCustomKeyStore request rate`	5
`CreateAlias request rate`	5
`CreateCustomKeyStore request rate`	5
`CreateGrant request rate`	50
`CreateKey request rate`	5
`DeleteAlias request rate`	15
`DeleteCustomKeyStore request rate`	5
`DeleteImportedKeyMaterial request rate`	15
`DescribeCustomKeyStores request rate`	5
`DescribeKey request rate`	2000

Nama kuota	Nilai default (permintaan per detik)
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
GenerateDataKeyPair (ECC_NIST_P384) request rate Berlaku untuk: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (ECC_NIST_P521) request rate Berlaku untuk: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100

Nama kuota	Nilai default (permintaan per detik)
`GenerateDataKeyPair (ECC_SECG_P256K1) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100
`GenerateDataKeyPair (ECC_NIST_EDWARDS25519) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100

`GenerateDataKeyPair (RSA_2048) request rate`
 	
Berlaku untuk:
 	
• `GenerateDataKeyPair`
• `GenerateDataKeyPairWithoutPlaintext`
	Kuota ini bervariasi dengan Wilayah AWS.
	
	• 20 (semua Wilayah lainnya)
	• Batasnya adalah 1 di Wilayah berikut:
	  • Tiongkok (Beijing), cn-utara-1
	  • Tiongkok (Ningxia), cn-northwest-1

`GenerateDataKeyPair (RSA_3072) request rate`
 	
Berlaku untuk:
 	
• `GenerateDataKeyPair`
• `GenerateDataKeyPairWithoutPlaintext`
	Kuota ini bervariasi dengan Wilayah AWS.
	
	• 4 (semua wilayah lainnya)
	• Batasnya adalah 0,5 di Wilayah berikut:
	  • Tiongkok (Beijing), cn-utara-1
	  • Tiongkok (Ningxia), cn-northwest-1

Nama kuota	Nilai default (permintaan per detik)

`GenerateDataKeyPair (RSA_4096) request rate`
 	
Berlaku untuk:
• `GenerateDataKeyPair`
• `GenerateDataKeyPairWithoutPlaintext`
	Kuota ini bervariasi dengan Wilayah AWS.
	• 1 (semua wilayah lainnya)
	• Batasnya adalah 0,1 di Wilayah berikut:
	  • Tiongkok (Beijing), cn-utara-1
	  • Tiongkok (Ningxia), cn-northwest-1

`GenerateDataKeyPair (SM2 – China Regions only) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	25
`GetKeyLastUsage request rate`	5
`GetKeyPolicy request rate`	1000
`GetKeyRotationStatus request rate`	1000
`GetParametersForImport request rate`	1
`GetPublicKey request rate`	2000
`ImportKeyMaterial request rate`	15
`ListAliases request rate`	500
`ListGrants request rate`	100
`ListKeyPolicies request rate`	100
`ListKeys request rate`	500

Nama kuota	Nilai default (permintaan per detik)
`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`	50
`RevokeGrant request rate`	50
`RotateKeyOnDemand request rate`	5
`ScheduleKeyDeletion request rate`	15
`TagResource request rate`	10
`UntagResource request rate`	5
`UpdateAlias request rate`	5
`UpdateCustomKeyStore request rate`	5
`UpdateKeyDescription request rate`	5

Nama kuota	Nilai default (permintaan per detik)
`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 AWS tidak dihitung terhadap kuota permintaan untuk Anda Akun AWS, bahkan ketika kuota tersebut 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 titik akhir AWS Key Management Service dan kuota di. Referensi Umum AWS
- Throttling didasarkan pada semua permintaan pada kunci KMS dari semua jenis di Wilayah. Total ini mencakup permintaan dari semua kepala sekolah di Akun AWS, termasuk permintaan dari AWS layanan atas nama Anda.
- Setiap kuota permintaan secara independen. Misalnya, permintaan untuk CreateKey operasi tidak berpengaruh pada kuota permintaan untuk CreateAlias operasi tersebut. 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 En kripsi dan Dekripsi berbagi kuota permintaan, tetapi kuota itu tidak bergantung 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. Pengecualiannya adalah GenerateDataKeyPair dan 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 dalam 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 K MS simetris Wilayah AWS dengan kuota bersama 10.000 permintaan per detik. Ketika Anda membuat 7.000 GenerateDataKey permintaan per detik dan 2.000 https://docs.aws.amazon.com/kms/latest/APIReference/API_Decrypt.html permintaan Decrypt 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 dalam penyimpanan kunci khusus di hitung terhadap tingkat permintaan operasi Kriptografi (simetris) untuk akun dan kuota permintaan penyimpanan 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 1.000 operasi per detik, Anda dapat membuat 400 permintaan Enkripsi dan 200 permintaan Dekripsi dengan kunci RSA KMS yang dapat mengenkripsi dan mendekripsi, ditambah 250 permintaan S ign dan 150 Per mintaan 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 eliptik (ECC) dan kunci KMS asimetris SM. ???

  Misalnya, dengan kuota permintaan 1.000 operasi per detik, Anda dapat membuat 400 permintaan Menandatangani dan 200 permintaan Verifikasi dengan kunci KMS ECC yang

dapat menandatangani dan memverifikasi, ditambah 250 permintaan Tanda tangani dan 150 permintaan Verifikasi dengan kunci SM2 KMS yang dapat menandatangani dan memverifikasi.

- Kuota permintaan penyimpanan kunci khusus adalah kuota permintaan bersama untuk operasi kriptografi pada kunci KMS di penyimpanan kunci khusus. Kuota ini dihitung secara terpisah untuk setiap penyimpanan kunci khusus.

  Operasi kriptografi pada kunci KMS enkripsi simetris dalam penyimpanan kunci khusus di hitung terhadap tingkat permintaan operasi Kriptografi (simetris) untuk akun dan kuota permintaan penyimpanan 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, ditambah hingga 300 permintaan tambahan per detik dengan kunci KMS Anda. ECC-based

Permintaan API yang dibuat atas nama Anda

Anda dapat membuat permintaan API secara langsung atau dengan menggunakan AWS layanan terintegrasi yang membuat AWS KMS permintaan API atas nama Anda. Kuota berlaku untuk kedua jenis permintaan.

Misalnya, Anda dapat 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 membuat permintaan `GenerateDataKey` (untuk unggahan) atau `Decrypt` (untuk unduhan) atas nama Anda. AWS KMS Permintaan ini dihitung dalam kuota Anda, AWS KMS jadi batasi permintaan jika Anda melebihi total gabungan 5.500 (atau 10.000 atau 50.000 tergantung pada Anda Wilayah AWS) unggahan atau unduhan per detik dari objek S3 yang dienkripsi dengan. SSE-KMS

Cross-account permintaan

Ketika aplikasi dalam satu Akun AWS menggunakan kunci KMS yang dimiliki oleh akun lain, 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 penyimpanan kunci khusus

AWS KMS mempertahankan kuota permintaan untuk operasi kriptografi pada kunci KMS di penyimpanan kunci khusus. Kuota permintaan ini dihitung secara terpisah untuk setiap penyimpanan kunci khusus.

Kuota permintaan toko kunci khusus	Nilai default (permintaan per detik) untuk setiap penyimpanan kunci khusus	Dapat Disesuaikan
AWS CloudHSM kuota permintaan toko kunci	1800	Tidak
Kuota permintaan penyimpanan kunci eksternal	1800	Tidak

> **Note**
>
> AWS KMS kuota permintaan penyimpanan kunci kustom tidak muncul di konsol Kuota Layanan. Anda tidak dapat melihat atau mengelola kuota ini dengan menggunakan operasi API Kuota Layanan.
>
> 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` pada tingkat yang lebih rendah dari yang diharapkan. Jika ini terjadi, turunkan tingkat permintaan Anda ke AWS KMS, kurangi beban yang tidak terkait, atau gunakan AWS CloudHSM cluster khusus untuk penyimpanan AWS CloudHSM kunci Anda.
>
> AWS KMS melaporkan pelambatan permintaan penyimpanan kunci eksternal dalam `ExternalKeyStoreThrottle` CloudWatch metrik. Anda dapat menggunakan metrik ini untuk melihat pola pelambatan, membuat alarm, dan menyesuaikan kuota permintaan penyimpanan kunci eksternal Anda.

Permintaan untuk operasi kriptografi pada kunci KMS di penyimpanan kunci khusus diperhitungkan dalam dua kuota:

- Kuota tarif permintaan operasi kriptografi (simetris) (per akun)

Permintaan untuk operasi kriptografi pada kunci KMS di penyimpanan kunci khusus dihitung dalam `Cryptographic operations (symmetric) request rate` kuota untuk masing-masing Akun AWS dan Wilayah. Misalnya, di AS Timur (Virginia Utara) (us-east-1), masing-masing Akun AWS dapat memiliki hingga 100.000 permintaan per detik pada kunci KMS enkripsi simetris, termasuk permintaan yang menggunakan kunci KMS di penyimpanan kunci khusus.

- Kuota permintaan penyimpanan kunci khusus (per penyimpanan kunci khusus)

  Permintaan untuk operasi kriptografi pada kunci KMS di penyimpanan kunci khusus juga dihitung terhadap 1.800 operasi per detik. `Custom key store request quota` Kuota ini dihitung secara terpisah untuk setiap penyimpanan kunci khusus. Mereka mungkin menyertakan permintaan dari beberapa Akun AWS yang menggunakan kunci KMS di penyimpanan kunci khusus.

Misalnya, operasi Enkripsi pada kunci KMS di penyimpanan kunci khusus (salah satu jenis) di Wilayah AS Timur (Virginia Utara) (us-east-1) dihitung terhadap kuota `Cryptographic operations (symmetric) request rate` tingkat akun (100.000 permintaan per detik) untuk akun dan Wilayahnya, dan menuju `Custom key store request quota` (1.800 permintaan per detik) untuk penyimpanan kunci kustomnya. Namun, permintaan untuk operasi manajemen, seperti PutKeyPolicy, pada kunci KMS di penyimpanan 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.

- Tarif 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 diimplementasikan secara otomatis untuk kesalahan HTTP 400 di beberapa AWS SDKs.

- 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 bahkan jika 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 untuk kuota sumber daya rotasi sesuai permintaan dan kuota permintaan toko AWS CloudHSM utama. 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 Dukungan 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.

Contoh kode untuk AWS KMS menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan AWS KMS kit pengembangan AWS perangkat lunak (SDK).

Dasar-dasar adalah contoh kode yang menunjukkan kepada Anda bagaimana melakukan operasi penting dalam suatu layanan.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- Contoh dasar untuk AWS KMS menggunakan AWS SDKs
  - Halo AWS Key Management Service
  - Pelajari dasar-dasar AWS KMS dengan AWS SDK
  - Tindakan untuk AWS KMS menggunakan AWS SDKs
    - Gunakan CreateAlias dengan AWS SDK atau CLI
    - Gunakan CreateGrant dengan AWS SDK atau CLI
    - Gunakan CreateKey dengan AWS SDK atau CLI
    - Gunakan Decrypt dengan AWS SDK atau CLI
    - Gunakan DeleteAlias dengan AWS SDK atau CLI
    - Gunakan DescribeKey dengan AWS SDK atau CLI
    - Gunakan DisableKey dengan AWS SDK atau CLI
    - Gunakan EnableKey dengan AWS SDK atau CLI
    - Gunakan EnableKeyRotation dengan AWS SDK atau CLI
    - Gunakan Encrypt dengan AWS SDK atau CLI

    - Gunakan GenerateDataKey dengan AWS SDK atau CLI
    - Gunakan GenerateDataKeyWithoutPlaintext dengan AWS SDK atau CLI
    - Gunakan GenerateRandom dengan AWS SDK atau CLI
    - Gunakan GetKeyPolicy dengan AWS SDK atau CLI
    - Gunakan ListAliases dengan AWS SDK atau CLI
    - Gunakan ListGrants dengan AWS SDK atau CLI
    - Gunakan ListKeyPolicies dengan AWS SDK atau CLI
    - Gunakan ListKeys dengan AWS SDK atau CLI
    - Gunakan PutKeyPolicy dengan AWS SDK atau CLI
    - Gunakan ReEncrypt dengan AWS SDK atau CLI
    - Gunakan RetireGrant dengan AWS SDK atau CLI
    - Gunakan RevokeGrant dengan AWS SDK atau CLI
    - Gunakan ScheduleKeyDeletion dengan AWS SDK atau CLI
    - Gunakan Sign dengan AWS SDK atau CLI
    - Gunakan TagResource dengan AWS SDK atau CLI
    - Gunakan UpdateAlias dengan AWS SDK atau CLI
    - Gunakan Verify dengan AWS SDK atau CLI
- Skenario untuk AWS KMS menggunakan AWS SDKs
    - Bekerja dengan enkripsi tabel DynamoDB menggunakan v2 AWS Command Line Interface

Contoh dasar untuk AWS KMS menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar AWS Key Management Service dengan AWS SDKs.

Contoh

- Halo AWS Key Management Service
- Pelajari dasar-dasar AWS KMS dengan AWS SDK
- Tindakan untuk AWS KMS menggunakan AWS SDKs
    - Gunakan CreateAlias dengan AWS SDK atau CLI
    - Gunakan CreateGrant dengan AWS SDK atau CLI

- Gunakan CreateKey dengan AWS SDK atau CLI
- Gunakan Decrypt dengan AWS SDK atau CLI
- Gunakan DeleteAlias dengan AWS SDK atau CLI
- Gunakan DescribeKey dengan AWS SDK atau CLI
- Gunakan DisableKey dengan AWS SDK atau CLI
- Gunakan EnableKey dengan AWS SDK atau CLI
- Gunakan EnableKeyRotation dengan AWS SDK atau CLI
- Gunakan Encrypt dengan AWS SDK atau CLI
- Gunakan GenerateDataKey dengan AWS SDK atau CLI
- Gunakan GenerateDataKeyWithoutPlaintext dengan AWS SDK atau CLI
- Gunakan GenerateRandom dengan AWS SDK atau CLI
- Gunakan GetKeyPolicy dengan AWS SDK atau CLI
- Gunakan ListAliases dengan AWS SDK atau CLI
- Gunakan ListGrants dengan AWS SDK atau CLI
- Gunakan ListKeyPolicies dengan AWS SDK atau CLI
- Gunakan ListKeys dengan AWS SDK atau CLI
- Gunakan PutKeyPolicy dengan AWS SDK atau CLI
- Gunakan ReEncrypt dengan AWS SDK atau CLI
- Gunakan RetireGrant dengan AWS SDK atau CLI
- Gunakan RevokeGrant dengan AWS SDK atau CLI
- Gunakan ScheduleKeyDeletion dengan AWS SDK atau CLI
- Gunakan Sign dengan AWS SDK atau CLI
- Gunakan TagResource dengan AWS SDK atau CLI
- Gunakan UpdateAlias dengan AWS SDK atau CLI
- Gunakan Verify dengan AWS SDK atau CLI

Halo AWS Key Management Service

Contoh kode berikut menunjukkan cara untuk mulai menggunakan AWS Key Management Service.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /*
         * The `subscribe` method is required when using paginator methods in the
 AWS SDK
         * because paginator methods return an instance of a `ListKeysPublisher`,
 which is
```

```
         * based on a reactive stream. This allows asynchronous retrieval of
 paginated
         * results as they become available. By subscribing to the stream, we can
 process
         * each page of results as they are emitted.
         */
        ListKeysPublisher keysPublisher =
 kmsAsyncClient.listKeysPaginator(listKeysRequest);
        CompletableFuture<Void> future = keysPublisher
            .subscribe(r -> r.keys().forEach(key ->
                System.out.println("The key ARN is: " + key.keyArn() + ". The key
 Id is: " + key.keyId())))
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    System.err.println("Error occurred: " +
 exception.getMessage());
                } else {
                    System.out.println("Successfully listed all keys.");
                }
            });

        try {
            future.join();
        } catch (Exception e) {
            System.err.println("Failed to list keys: " + e.getMessage());
        }
    }
}
```

- Untuk detail API, lihat ListKeysdi Referensi AWS SDK for Java 2.x API.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
include "vendor/autoload.php";

use Aws\Kms\KmsClient;

echo "This file shows how to connect to the KmsClient, uses a paginator to get
 the keys for the account, and lists the KeyIds for up to 10 keys.\n";

$client = new KmsClient([]);

$pageLength = 10; // Change this value to change the number of records shown, or
 to break up the result into pages.

$keys = [];
$keysPaginator = $client->getPaginator("ListKeys", ['Limit' => $pageLength]);
foreach($keysPaginator as $page){
    foreach($page['Keys'] as $index => $key){
        echo "The $index index Key's ID is: {$key['KeyId']}\n";
    }
    echo "End of page one of results. Alter the \$pageLength variable to see more
 results.\n";
    break;
}
```

- Untuk detail API, lihat ListKeysdi Referensi AWS SDK untuk PHP API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kms->listkeys( ).
        MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.
      CATCH /aws1/cx_kmskmsinternalex.
```

```
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListKeysdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Pelajari dasar-dasar AWS KMS dengan AWS SDK

Contoh-contoh kode berikut menunjukkan cara:

- Buat kunci KMS.
- Buat daftar kunci KMS untuk akun Anda dan dapatkan detailnya.
- Aktifkan dan nonaktifkan tombol KMS.
- Hasilkan kunci data simetris yang dapat digunakan untuk enkripsi sisi klien.
- Hasilkan kunci asimetris yang digunakan untuk menandatangani data secara digital.
- Tombol tag.
- Hapus kunci KMS.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Menjalankan skenario di prompt perintah.

```
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

```
import software.amazon.awssdk.services.kms.model.AlreadyExistsException;
import software.amazon.awssdk.services.kms.model.DisabledException;
import software.amazon.awssdk.services.kms.model.EnableKeyRotationResponse;
import software.amazon.awssdk.services.kms.model.KmsException;
import software.amazon.awssdk.services.kms.model.NotFoundException;
import software.amazon.awssdk.services.kms.model.RevokeGrantResponse;
import java.util.List;
import java.util.Scanner;
import java.util.concurrent.CompletableFuture;
import java.util.concurrent.CompletionException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class KMSScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");
    private static String accountId = "";

    private static final Logger logger =
 LoggerFactory.getLogger(KMSScenario.class);

    static KMSActions kmsActions = new KMSActions();

    static Scanner scanner = new Scanner(System.in);

    static String aliasName = "alias/dev-encryption-key";

    public static void main(String[] args) {
        final String usage = """
            Usage: <granteePrincipal>

            Where:
               granteePrincipal - The principal (user, service account, or group)
 to whom the grant or permission is being given.
            """;
```

```
        if (args.length != 1) {
            logger.info(usage);
            return;
        }
        String granteePrincipal = args[0];
        String policyName = "default";

        accountId = kmsActions.getAccountId();
        String keyDesc = "Created by the AWS KMS API";

        logger.info(DASHES);
        logger.info("""
            Welcome to the AWS Key Management SDK Basics scenario.

            This program demonstrates how to interact with AWS Key Management
using the AWS SDK for Java (v2).
            The AWS Key Management Service (KMS) is a secure and highly available
service that allows you to create
            and manage AWS KMS keys and control their use across a wide range of
AWS services and applications.
            KMS provides a centralized and unified approach to managing
encryption keys, making it easier to meet your
            data protection and regulatory compliance requirements.

            This Basics scenario creates two key types:

            - A symmetric encryption key is used to encrypt and decrypt data.
            - An asymmetric key used to digitally sign data.

            Let's get started...
            """);
        waitForInputToContinue(scanner);

        try {
        // Run the methods that belong to this scenario.
        String targetKeyId = runScenario(granteePrincipal, keyDesc, policyName);
        requestDeleteResources(aliasName, targetKeyId);

        } catch (Throwable rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
```

```
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
        }
    }

    private static String runScenario(String granteePrincipal, String keyDesc,
String policyName) throws Throwable {
        logger.info(DASHES);
        logger.info("1. Create a symmetric KMS key\n");
        logger.info("First, the program will creates a symmetric KMS key that you
can used to encrypt and decrypt data.");
        waitForInputToContinue(scanner);
        String targetKeyId;
        try {
            CompletableFuture<String> futureKeyId =
kmsActions.createKeyAsync(keyDesc);
            targetKeyId = futureKeyId.join();
            logger.info("A symmetric key was successfully created " +
targetKeyId);

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("""
            2. Enable a KMS key

            By default, when the SDK creates an AWS key, it is enabled. The next
bit of code checks to
            determine if the key is enabled.
             """);
        waitForInputToContinue(scanner);
        boolean isEnabled;
        try {
```

```
            CompletableFuture<Boolean> futureIsKeyEnabled =
kmsActions.isKeyEnabledAsync(targetKeyId);
            isEnabled = futureIsKeyEnabled.join();
            logger.info("Is the key enabled? {}", isEnabled);

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            throw cause;
        }

        if (!isEnabled)
            try {
                CompletableFuture<Void> future =
kmsActions.enableKeyAsync(targetKeyId);
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof KmsException kmsEx) {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                } else {
                    logger.info("An unexpected error occurred: " +
rt.getMessage());
                }
                throw cause;
            }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("3. Encrypt data using the symmetric KMS key");
        String plaintext = "Hello, AWS KMS!";
        logger.info("""
            One of the main uses of symmetric keys is to encrypt and decrypt
data.
            Next, the code encrypts the string {} with the SYMMETRIC_DEFAULT
encryption algorithm.
            """, plaintext);
```

```
        waitForInputToContinue(scanner);
        SdkBytes encryptedData;
        try {
            CompletableFuture<SdkBytes> future =
kmsActions.encryptDataAsync(targetKeyId, plaintext);
            encryptedData = future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof DisabledException kmsDisabledEx) {
                logger.info("KMS error occurred due to a disabled
key: Error message: {}, Error code {}", kmsDisabledEx.getMessage(),
kmsDisabledEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("4. Create an alias");
        logger.info("""

            The alias name should be prefixed with 'alias/'.
            The default, 'alias/dev-encryption-key'.
             """);
        waitForInputToContinue(scanner);

        try {
            CompletableFuture<Void> future =
kmsActions.createCustomAliasAsync(targetKeyId, aliasName);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof AlreadyExistsException kmsExistsEx) {
                if (kmsExistsEx.getMessage().contains("already exists")) {
                    logger.info("The alias '" + aliasName + "' already exists.
Moving on...");
                }
            } else {
```

```
                logger.error("An unexpected error occurred: " + rt.getMessage(),
rt);
                deleteKey(targetKeyId);
                throw cause;
            }
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("5. List all of your aliases");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Object> future = kmsActions.listAllAliasesAsync();
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("6. Enable automatic rotation of the KMS key");
        logger.info("""

            By default, when the SDK enables automatic rotation of a KMS key,
            KMS rotates the key material of the KMS key one year (approximately
365 days) from the enable date and every year
            thereafter.
            """);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<EnableKeyRotationResponse> future =
kmsActions.enableKeyRotationAsync(targetKeyId);
            future.join();
```

```
        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("""
            7. Create a grant

            A grant is a policy instrument that allows Amazon Web Services
principals to use KMS keys.
            It also can allow them to view a KMS key (DescribeKey) and create and
manage grants.
            When authorizing access to a KMS key, grants are considered along
with key policies and IAM policies.
            """);

        waitForInputToContinue(scanner);
        String grantId = null;
        try {
            CompletableFuture<String> futureGrantId =
kmsActions.grantKeyAsync(targetKeyId, granteePrincipal);
            grantId = futureGrantId.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteKey(targetKeyId);
            throw cause;
        }
```

```
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("8. List grants for the KMS key");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Object> future =
kmsActions.displayGrantIdsAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("9. Revoke the grant");
        logger.info("""
            The revocation of a grant immediately removes the permissions and
access that the grant had provided.
            This means that any principal (user, role, or service) that was
granted access to perform specific
            KMS operations on a KMS key will no longer be able to perform those
operations.
            """);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<RevokeGrantResponse> future =
kmsActions.revokeKeyGrantAsync(targetKeyId, grantId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
```

```
                if (kmsEx.getMessage().contains("Grant does not exist")) {
                    logger.info("The grant ID '" + grantId + "' does not exist.
Moving on...");
                } else {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                    throw cause;
                }
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
                deleteAliasName(aliasName);
                deleteKey(targetKeyId);
                throw cause;
            }
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("10. Decrypt the data\n");
        logger.info("""
            Lets decrypt the data that was encrypted in an early step.
            The code uses the same key to decrypt the string that we encrypted
earlier in the program.
            """);
        waitForInputToContinue(scanner);
        String decryptedData = "";
        try {
            CompletableFuture<String> future =
kmsActions.decryptDataAsync(encryptedData, targetKeyId);
            decryptedData = future.join();
            logger.info("Decrypted data: " + decryptedData);

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
```

```
        logger.info("Decrypted text is: " + decryptedData);
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("11. Replace a key policy\n");
        logger.info("""
            A key policy is a resource policy for a KMS key. Key policies are the
primary way to control
            access to KMS keys. Every KMS key must have exactly one key policy.
The statements in the key policy
            determine who has permission to use the KMS key and how they can use
it.
            You can also use IAM policies and grants to control access to the KMS
key, but every KMS key
            must have a key policy.

            By default, when you create a key by using the SDK, a policy is
created that
            gives the AWS account that owns the KMS key full access to the KMS
key.

            Let's try to replace the automatically created policy with the
following policy.

                "Version":"2012-10-17",
                "Statement": [{
                "Effect": "Allow",
                "Principal": {"AWS": "arn:aws:iam::0000000000:root"},
                "Action": "kms:*",
                "Resource": "*"
                }]
            """);

        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Boolean> future =
kmsActions.replacePolicyAsync(targetKeyId, policyName, accountId);
            boolean success = future.join();
            if (success) {
                logger.info("Key policy replacement succeeded.");
            } else {
                logger.error("Key policy replacement failed.");
            }
```

```
        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("12. Get the key policy\n");
        logger.info("The next bit of code that runs gets the key policy to make
sure it exists.");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<String> future =
kmsActions.getKeyPolicyAsync(targetKeyId, policyName);
            String policy = future.join();
            if (!policy.isEmpty()) {
                logger.info("Retrieved policy: " + policy);
            }

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("13. Create an asymmetric KMS key and sign your data\n");
        logger.info("""
```

```
            Signing your data with an AWS key can provide several benefits that
make it an attractive option
            for your data signing needs. By using an AWS KMS key, you can
leverage the
            security controls and compliance features provided by AWS,
            which can help you meet various regulatory requirements and enhance
the overall security posture
            of your organization.
            """);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Boolean> future = kmsActions.signVerifyDataAsync();
            boolean success = future.join();
            if (success) {
                logger.info("Sign and verify data operation succeeded.");
            } else {
                logger.error("Sign and verify data operation failed.");
            }

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("14. Tag your symmetric KMS Key\n");
        logger.info("""
            By using tags, you can improve the overall management, security, and
governance of your
            KMS keys, making it easier to organize, track, and control access to
your encrypted data within
            your AWS environment
            """);
        waitForInputToContinue(scanner);
        try {
```

```
            CompletableFuture<Void> future =
 kmsActions.tagKMSKeyAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
 {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);
        return targetKeyId;
    }

    // Deletes KMS resources with user input.
    private static void requestDeleteResources(String aliasName, String
 targetKeyId) {
        logger.info(DASHES);
        logger.info("15. Schedule the deletion of the KMS key\n");
        logger.info("""
            By default, KMS applies a waiting period of 30 days,
            but you can specify a waiting period of 7-30 days. When this
 operation is successful,
            the key state of the KMS key changes to PendingDeletion and the key
 can't be used in any
            cryptographic operations. It remains in this state for the duration
 of the waiting period.

            Deleting a KMS key is a destructive and potentially dangerous
 operation. When a KMS key is deleted,
            all data that was encrypted under the KMS key is unrecoverable.
            """);
        logger.info("Would you like to delete the Key Management resources? (y/
n)");
        String delAns = scanner.nextLine().trim();
        if (delAns.equalsIgnoreCase("y")) {
            logger.info("You selected to delete the AWS KMS resources.");
            waitForInputToContinue(scanner);
```

```
            try {
                CompletableFuture<Void> future =
kmsActions.deleteSpecificAliasAsync(aliasName);
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof KmsException kmsEx) {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                } else {
                    logger.info("An unexpected error occurred: " +
rt.getMessage());
                }
            }
            waitForInputToContinue(scanner);
            try {
                CompletableFuture<Void> future =
kmsActions.disableKeyAsync(targetKeyId);
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof KmsException kmsEx) {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                } else {
                    logger.info("An unexpected error occurred: " +
rt.getMessage());
                }
            }

            try {
                CompletableFuture<Void> future =
kmsActions.deleteKeyAsync(targetKeyId);
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof KmsException kmsEx) {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                } else {
```

```
                    logger.info("An unexpected error occurred: " +
rt.getMessage());
                }
            }

        } else {
            logger.info("The Key Management resources will not be deleted");
        }

        logger.info(DASHES);
        logger.info("This concludes the AWS Key Management SDK scenario");
        logger.info(DASHES);
    }

    // This method is invoked from Exceptions to clean up the resources.
    private static void deleteKey(String targetKeyId) {
        try {
            CompletableFuture<Void> future =
kmsActions.disableKeyAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
        }

        try {
            CompletableFuture<Void> future =
kmsActions.deleteKeyAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
```

```
        }
    }

    // This method is invoked from Exceptions to clean up the resources.
    private static void deleteAliasName(String aliasName) {
        try {
            CompletableFuture<Void> future =
 kmsActions.deleteSpecificAliasAsync(aliasName);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
 {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
        }
    }

    private static void waitForInputToContinue(Scanner scanner) {
        while (true) {
            logger.info("");
            logger.info("Enter 'c' followed by <ENTER> to continue:");
            String input = scanner.nextLine();

            if (input.trim().equalsIgnoreCase("c")) {
                logger.info("Continuing with the program...");
                logger.info("");
                break;
            } else {
                // Handle invalid input.
                logger.info("Invalid input. Please try again.");
            }
        }
    }
}
```

Tentukan kelas yang membungkus tindakan KMS.

```
public class KMSActions {
```

```
   private static final Logger logger =
LoggerFactory.getLogger(KMSActions.class);
   private static KmsAsyncClient kmsAsyncClient;

   /**
    * Retrieves an asynchronous AWS Key Management Service (KMS) client.
    * <p>
    * This method creates and returns a singleton instance of the KMS async
client, with the following configurations:
    * <ul>
    *   <li>Max concurrency: 100</li>
    *   <li>Connection timeout: 60 seconds</li>
    *   <li>Read timeout: 60 seconds</li>
    *   <li>Write timeout: 60 seconds</li>
    *   <li>API call timeout: 2 minutes</li>
    *   <li>API call attempt timeout: 90 seconds</li>
    *   <li>Retry policy: up to 3 retries</li>
    *   <li>Credentials provider: environment variable credentials provider</li>
    * </ul>
    * <p>
    * If the client instance has already been created, it is returned instead of
creating a new one.
    *
    * @return the KMS async client instance
    */
   private static KmsAsyncClient getAsyncClient() {
       if (kmsAsyncClient == null) {
           SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
               .maxConcurrency(100)
               .connectionTimeout(Duration.ofSeconds(60))
               .readTimeout(Duration.ofSeconds(60))
               .writeTimeout(Duration.ofSeconds(60))
               .build();

           ClientOverrideConfiguration overrideConfig =
ClientOverrideConfiguration.builder()
               .apiCallTimeout(Duration.ofMinutes(2))
               .apiCallAttemptTimeout(Duration.ofSeconds(90))
               .retryPolicy(RetryPolicy.builder()
                   .numRetries(3)
                   .build())
               .build();

           kmsAsyncClient = KmsAsyncClient.builder()
```

```
                .httpClient(httpClient)
                .overrideConfiguration(overrideConfig)
                .build();
        }
        return kmsAsyncClient;
    }

    /**
     * Creates a new symmetric encryption key asynchronously.
     *
     * @param keyDesc the description of the key to be created
     * @return a {@link CompletableFuture} that completes with the ID of the
newly created key
     * @throws RuntimeException if an error occurs while creating the key
     */
    public CompletableFuture<String> createKeyAsync(String keyDesc) {
        CreateKeyRequest keyRequest = CreateKeyRequest.builder()
            .description(keyDesc)
            .keySpec(KeySpec.SYMMETRIC_DEFAULT)
            .keyUsage(KeyUsageType.ENCRYPT_DECRYPT)
            .build();

        return getAsyncClient().createKey(keyRequest)
            .thenApply(resp -> resp.keyMetadata().keyId())
            .exceptionally(ex -> {
                throw new RuntimeException("An error occurred while creating the
key: " + ex.getMessage(), ex);
            });
    }

    /**
     * Asynchronously checks if a specified key is enabled.
     *
     * @param keyId the ID of the key to check
     * @return a {@link CompletableFuture} that, when completed, indicates
whether the key is enabled or not
     *
     * @throws RuntimeException if an exception occurs while checking the key
state
     */
    public CompletableFuture<Boolean> isKeyEnabledAsync(String keyId) {
        DescribeKeyRequest keyRequest = DescribeKeyRequest.builder()
            .keyId(keyId)
            .build();
```

```
        CompletableFuture<DescribeKeyResponse> responseFuture =
getAsyncClient().describeKey(keyRequest);
        return responseFuture.whenComplete((resp, ex) -> {
            if (resp != null) {
                KeyState keyState = resp.keyMetadata().keyState();
                if (keyState == KeyState.ENABLED) {
                    logger.info("The key is enabled.");
                } else {
                    logger.info("The key is not enabled. Key state: {}",
keyState);
                }
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(resp -> resp.keyMetadata().keyState() == KeyState.ENABLED);
    }

    /**
     * Asynchronously enables the specified key.
     *
     * @param keyId the ID of the key to enable
     * @return a {@link CompletableFuture} that completes when the key has been
enabled
     */
    public CompletableFuture<Void> enableKeyAsync(String keyId) {
        EnableKeyRequest enableKeyRequest = EnableKeyRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<EnableKeyResponse> responseFuture =
getAsyncClient().enableKey(enableKeyRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Key with ID [{}] has been enabled.", keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while enabling
key: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while enabling key: " + exception.getMessage(), exception);
                }
            }
```

```
        });

        return responseFuture.thenApply(response -> null);
    }

    /**
     * Encrypts the given text asynchronously using the specified KMS client and
key ID.
     *
     * @param keyId the ID of the KMS key to use for encryption
     * @param text the text to encrypt
     * @return a CompletableFuture that completes with the encrypted data as an
SdkBytes object
     */
    public CompletableFuture<SdkBytes> encryptDataAsync(String keyId, String
text) {
        SdkBytes myBytes = SdkBytes.fromUtf8String(text);
        EncryptRequest encryptRequest = EncryptRequest.builder()
            .keyId(keyId)
            .plaintext(myBytes)
            .build();

        CompletableFuture<EncryptResponse> responseFuture =
getAsyncClient().encrypt(encryptRequest).toCompletableFuture();
        return responseFuture.whenComplete((response, ex) -> {
            if (response != null) {
                String algorithm = response.encryptionAlgorithm().toString();
                logger.info("The string was encrypted with algorithm {}.",
algorithm);
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(EncryptResponse::ciphertextBlob);
    }

    /**
     * Creates a custom alias for the specified target key asynchronously.
     *
     * @param targetKeyId the ID of the target key for the alias
     * @param aliasName   the name of the alias to create
     * @return a {@link CompletableFuture} that completes when the alias creation
operation is finished
     */
```

```
   public CompletableFuture<Void> createCustomAliasAsync(String targetKeyId,
String aliasName) {
       CreateAliasRequest aliasRequest = CreateAliasRequest.builder()
           .aliasName(aliasName)
           .targetKeyId(targetKeyId)
           .build();

       CompletableFuture<CreateAliasResponse> responseFuture =
getAsyncClient().createAlias(aliasRequest);
       responseFuture.whenComplete((response, exception) -> {
           if (exception == null) {
               logger.info("{} was successfully created.", aliasName);
           } else {
               if (exception instanceof ResourceExistsException) {
                   logger.info("Alias [{}] already exists. Moving on...",
aliasName);
               } else if (exception instanceof KmsException kmsEx) {
                   throw new RuntimeException("KMS error occurred while creating
alias: " + kmsEx.getMessage(), kmsEx);
               } else {
                   throw new RuntimeException("An unexpected error occurred
while creating alias: " + exception.getMessage(), exception);
               }
           }
       });

       return responseFuture.thenApply(response -> null);
   }

   /**
    * Asynchronously lists all the aliases in the current AWS account.
    *
    * @return a {@link CompletableFuture} that completes when the list of
aliases has been processed
    */
   public CompletableFuture<Object> listAllAliasesAsync() {
       ListAliasesRequest aliasesRequest = ListAliasesRequest.builder()
           .limit(15)
           .build();

       ListAliasesPublisher paginator =
getAsyncClient().listAliasesPaginator(aliasesRequest);
       return paginator.subscribe(response -> {
               response.aliases().forEach(alias ->
```

```
                    logger.info("The alias name is: " + alias.aliasName())
                );
            })
            .thenApply(v -> null)
            .exceptionally(ex -> {
                if (ex.getCause() instanceof KmsException) {
                    KmsException e = (KmsException) ex.getCause();
                    throw new RuntimeException("A KMS exception occurred: " +
e.getMessage());
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage());
                }
            });
    }

    /**
     * Enables key rotation asynchronously for the specified key ID.
     *
     * @param keyId the ID of the key for which to enable key rotation
     * @return a CompletableFuture that represents the asynchronous operation of
enabling key rotation
     * @throws RuntimeException if there was an error enabling key rotation,
either due to a KMS exception or an unexpected error
     */
    public CompletableFuture<EnableKeyRotationResponse>
enableKeyRotationAsync(String keyId) {
        EnableKeyRotationRequest enableKeyRotationRequest =
EnableKeyRotationRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<EnableKeyRotationResponse> responseFuture =
getAsyncClient().enableKeyRotation(enableKeyRotationRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Key rotation has been enabled for key with id [{}]",
keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("Failed to enable key rotation: "
+ kmsEx.getMessage(), kmsEx);
                } else {
```

```
                    throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
                }
            }
        });

        return responseFuture;
    }

    /**
     * Grants permissions to a specified principal on a customer master key (CMK)
asynchronously.
     *
     * @param keyId             The unique identifier for the customer master key
(CMK) that the grant applies to.
     * @param granteePrincipal  The principal that is given permission to perform
the operations that the grant permits on the CMK.
     * @return A {@link CompletableFuture} that, when completed, contains the ID
of the created grant.
     * @throws RuntimeException If an error occurs during the grant creation
process.
     */
    public CompletableFuture<String> grantKeyAsync(String keyId, String
granteePrincipal) {
        List<GrantOperation> grantPermissions = List.of(
            GrantOperation.ENCRYPT,
            GrantOperation.DECRYPT,
            GrantOperation.DESCRIBE_KEY
        );

        CreateGrantRequest grantRequest = CreateGrantRequest.builder()
            .keyId(keyId)
            .name("grant1")
            .granteePrincipal(granteePrincipal)
            .operations(grantPermissions)
            .build();

        CompletableFuture<CreateGrantResponse> responseFuture =
getAsyncClient().createGrant(grantRequest);
        responseFuture.whenComplete((response, ex) -> {
            if (ex == null) {
                logger.info("Grant created successfully with ID: " +
response.grantId());
            } else {
```

```
                if (ex instanceof KmsException kmsEx) {
                    throw new RuntimeException("Failed to create grant: " +
kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage(), ex);
                }
            }
        });

        return responseFuture.thenApply(CreateGrantResponse::grantId);
    }

    /**
     * Asynchronously displays the grant IDs for the specified key ID.
     *
     * @param keyId the ID of the AWS KMS key for which to list the grants
     * @return a {@link CompletableFuture} that, when completed, will be null
if the operation succeeded, or will throw a {@link RuntimeException} if the
operation failed
     * @throws RuntimeException if there was an error listing the grants, either
due to an {@link KmsException} or an unexpected error
     */
    public CompletableFuture<Object> displayGrantIdsAsync(String keyId) {
        ListGrantsRequest grantsRequest = ListGrantsRequest.builder()
            .keyId(keyId)
            .limit(15)
            .build();

        ListGrantsPublisher paginator =
getAsyncClient().listGrantsPaginator(grantsRequest);
        return paginator.subscribe(response -> {
                response.grants().forEach(grant -> {
                    logger.info("The grant Id is: " + grant.grantId());
                });
            })
            .thenApply(v -> null)
            .exceptionally(ex -> {
                Throwable cause = ex.getCause();
                if (cause instanceof KmsException) {
                    throw new RuntimeException("Failed to list grants: " +
cause.getMessage(), cause);
                } else {
```

```
                    throw new RuntimeException("An unexpected error occurred: " +
cause.getMessage(), cause);
                }
            });
    }

    /**
     * Revokes a grant for the specified AWS KMS key asynchronously.
     *
     * @param keyId   The ID or key ARN of the AWS KMS key.
     * @param grantId The identifier of the grant to be revoked.
     * @return A {@link CompletableFuture} representing the asynchronous
operation of revoking the grant.
     *         The {@link CompletableFuture} will complete with a {@link
RevokeGrantResponse} object
     *         if the operation is successful, or with a {@code null} value if an
error occurs.
     */
    public CompletableFuture<RevokeGrantResponse> revokeKeyGrantAsync(String
keyId, String grantId) {
        RevokeGrantRequest grantRequest = RevokeGrantRequest.builder()
            .keyId(keyId)
            .grantId(grantId)
            .build();

        CompletableFuture<RevokeGrantResponse> responseFuture =
getAsyncClient().revokeGrant(grantRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Grant ID: [" + grantId + "] was successfully
revoked!");
            } else {
                if (exception instanceof KmsException kmsEx) {
                    if (kmsEx.getMessage().contains("Grant does not exist")) {
                        logger.info("The grant ID '" + grantId + "' does not
exist. Moving on...");
                    } else {
                        throw new RuntimeException("KMS error occurred: " +
kmsEx.getMessage(), kmsEx);
                    }
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
                }
```

```
            }
        });

        return responseFuture;
    }


    /**
     * Asynchronously decrypts the given encrypted data using the specified key
ID.
     *
     * @param encryptedData The encrypted data to be decrypted.
     * @param keyId The ID of the key to be used for decryption.
     * @return A CompletableFuture that, when completed, will contain the
decrypted data as a String.
     *         If an error occurs during the decryption process, the
CompletableFuture will complete
     *         exceptionally with the error, and the method will return an empty
String.
     */
    public CompletableFuture<String> decryptDataAsync(SdkBytes encryptedData,
String keyId) {
        DecryptRequest decryptRequest = DecryptRequest.builder()
            .ciphertextBlob(encryptedData)
            .keyId(keyId)
            .build();

        CompletableFuture<DecryptResponse> responseFuture =
getAsyncClient().decrypt(decryptRequest);
        responseFuture.whenComplete((decryptResponse, exception) -> {
            if (exception == null) {
                logger.info("Data decrypted successfully for key ID: " + keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while
decrypting data: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while decrypting data: " + exception.getMessage(), exception);
                }
            }
        });
```

```
        return responseFuture.thenApply(decryptResponse ->
decryptResponse.plaintext().asString(StandardCharsets.UTF_8));
    }

    /**
     * Asynchronously replaces the policy for the specified KMS key.
     *
     * @param keyId       the ID of the KMS key to replace the policy for
     * @param policyName  the name of the policy to be replaced
     * @param accountId   the AWS account ID to be used in the policy
     * @return a {@link CompletableFuture} that completes with a boolean
indicating
     *         whether the policy replacement was successful or not
     */
    public CompletableFuture<Boolean> replacePolicyAsync(String keyId, String
policyName, String accountId) {
        String policy = """
    {
      "Version":"2012-10-17",
      "Statement": [{
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::%s:root"},
        "Action": "kms:*",
        "Resource": "*"
      }]
    }
    """.formatted(accountId);

        PutKeyPolicyRequest keyPolicyRequest = PutKeyPolicyRequest.builder()
            .keyId(keyId)
            .policyName(policyName)
            .policy(policy)
            .build();

        // First, get the current policy to check if it exists
        return getAsyncClient().getKeyPolicy(r ->
r.keyId(keyId).policyName(policyName))
            .thenCompose(response -> {
                logger.info("Current policy exists. Replacing it...");
                return getAsyncClient().putKeyPolicy(keyPolicyRequest);
            })
            .thenApply(putPolicyResponse -> {
                logger.info("The key policy has been replaced.");
                return true;
```

```
            })
            .exceptionally(throwable -> {
                if (throwable.getCause() instanceof LimitExceededException) {
                    logger.error("Cannot replace policy, as only one policy is
allowed per key.");
                    return false;
                }
                throw new RuntimeException("Error replacing policy", throwable);
            });
    }


    /**
     * Asynchronously retrieves the key policy for the specified key ID and
policy name.
     *
     * @param keyId       the ID of the AWS KMS key for which to retrieve the
policy
     * @param policyName the name of the key policy to retrieve
     * @return a {@link CompletableFuture} that, when completed, contains the key
policy as a {@link String}
     */
    public CompletableFuture<String> getKeyPolicyAsync(String keyId, String
policyName) {
        GetKeyPolicyRequest policyRequest = GetKeyPolicyRequest.builder()
            .keyId(keyId)
            .policyName(policyName)
            .build();

        return getAsyncClient().getKeyPolicy(policyRequest)
            .thenApply(response -> {
                String policy = response.policy();
                logger.info("The response is: " + policy);
                return policy;
            })
            .exceptionally(ex -> {
                throw new RuntimeException("Failed to get key policy", ex);
            });
    }

    /**
     * Asynchronously signs and verifies data using AWS KMS.
     *
     * <p>The method performs the following steps:
```

```
    * <ol>
    *     <li>Creates an AWS KMS key with the specified key spec, key usage, and
origin.</li>
    *     <li>Signs the provided message using the created KMS key and the
RSASSA-PSS-SHA-256 algorithm.</li>
    *     <li>Verifies the signature of the message using the created KMS key
and the RSASSA-PSS-SHA-256 algorithm.</li>
    * </ol>
    *
    * @return a {@link CompletableFuture} that completes with the result of the
signature verification,
    *         {@code true} if the signature is valid, {@code false} otherwise.
    * @throws KmsException if any error occurs during the KMS operations.
    * @throws RuntimeException if an unexpected error occurs.
    */
   public CompletableFuture<Boolean> signVerifyDataAsync() {
       String signMessage = "Here is the message that will be digitally signed";

       // Create an AWS KMS key used to digitally sign data.
       CreateKeyRequest createKeyRequest = CreateKeyRequest.builder()
           .keySpec(KeySpec.RSA_2048)
           .keyUsage(KeyUsageType.SIGN_VERIFY)
           .origin(OriginType.AWS_KMS)
           .build();

       return getAsyncClient().createKey(createKeyRequest)
           .thenCompose(createKeyResponse -> {
               String keyId = createKeyResponse.keyMetadata().keyId();

               SdkBytes messageBytes = SdkBytes.fromString(signMessage,
Charset.defaultCharset());
               SignRequest signRequest = SignRequest.builder()
                   .keyId(keyId)
                   .message(messageBytes)
                   .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                   .build();

               return getAsyncClient().sign(signRequest)
                   .thenCompose(signResponse -> {
                       byte[] signedBytes =
signResponse.signature().asByteArray();

                       VerifyRequest verifyRequest = VerifyRequest.builder()
                           .keyId(keyId)
```

```
.message(SdkBytes.fromByteArray(signMessage.getBytes(Charset.defaultCharset())))

.signature(SdkBytes.fromByteBuffer(ByteBuffer.wrap(signedBytes)))

.signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                            .build();

                        return getAsyncClient().verify(verifyRequest)
                            .thenApply(verifyResponse -> {
                                return (boolean) verifyResponse.signatureValid();
                            });
                    });
            })
            .exceptionally(throwable -> {
               throw new RuntimeException("Failed to sign or verify data",
throwable);
            });
   }

   /**
    * Asynchronously tags a KMS key with a specific tag.
    *
    * @param keyId the ID of the KMS key to be tagged
    * @return a {@link CompletableFuture} that completes when the tagging
operation is finished
    */
   public CompletableFuture<Void> tagKMSKeyAsync(String keyId) {
       Tag tag = Tag.builder()
           .tagKey("Environment")
           .tagValue("Production")
           .build();

       TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
           .keyId(keyId)
           .tags(tag)
           .build();

       return getAsyncClient().tagResource(tagResourceRequest)
           .thenRun(() -> {
               logger.info("{} key was tagged", keyId);
           })
           .exceptionally(throwable -> {
```

```
                throw new RuntimeException("Failed to tag the KMS key",
throwable);
            });
    }

    /**
     * Deletes a specific KMS alias asynchronously.
     *
     * @param aliasName the name of the alias to be deleted
     * @return a {@link CompletableFuture} representing the asynchronous
operation of deleting the specified alias
     */
    public CompletableFuture<Void> deleteSpecificAliasAsync(String aliasName) {
        DeleteAliasRequest deleteAliasRequest = DeleteAliasRequest.builder()
            .aliasName(aliasName)
            .build();

        return getAsyncClient().deleteAlias(deleteAliasRequest)
            .thenRun(() -> {
                logger.info("Alias {} has been deleted successfully", aliasName);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to delete alias: " +
aliasName, throwable);
            });
    }

    /**
     * Asynchronously disables the specified AWS Key Management Service (KMS)
key.
     *
     * @param keyId the ID or Amazon Resource Name (ARN) of the KMS key to be
disabled
     * @return a CompletableFuture that, when completed, indicates that the key
has been disabled successfully
     */
    public CompletableFuture<Void> disableKeyAsync(String keyId) {
        DisableKeyRequest keyRequest = DisableKeyRequest.builder()
            .keyId(keyId)
            .build();

        return getAsyncClient().disableKey(keyRequest)
            .thenRun(() -> {
                logger.info("Key {} has been disabled successfully",keyId);
```

```
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to disable key: " + keyId,
throwable);
            });
    }

    /**
     * Deletes a KMS key asynchronously.
     *
     * <p><strong>Warning:</strong> Deleting a KMS key is a destructive and
potentially dangerous operation.
     * When a KMS key is deleted, all data that was encrypted under the KMS key
becomes unrecoverable.
     * This means that any files, databases, or other data that were encrypted
using the deleted KMS key
     * will become permanently inaccessible. Exercise extreme caution when
deleting KMS keys.</p>
     *
     * @param keyId the ID of the KMS key to delete
     * @return a {@link CompletableFuture} that completes when the key deletion
is scheduled
     */
    public CompletableFuture<Void> deleteKeyAsync(String keyId) {
        ScheduleKeyDeletionRequest deletionRequest =
ScheduleKeyDeletionRequest.builder()
            .keyId(keyId)
            .pendingWindowInDays(7)
            .build();

        return getAsyncClient().scheduleKeyDeletion(deletionRequest)
            .thenRun(() -> {
                logger.info("Key {} will be deleted in 7 days", keyId);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to schedule key deletion for
key ID: " + keyId, throwable);
            });
    }


    public String getAccountId(){
        try (StsClient stsClient = StsClient.create()){
```

```
            GetCallerIdentityResponse callerIdentity =
 stsClient.getCallerIdentity();
            return callerIdentity.account();
        }
    }
}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Java 2.x .
  - CreateAlias
  - CreateGrant
  - CreateKey
  - Dekripsi
  - DescribeKey
  - DisableKey
  - EnableKey
  - Enkripsi
  - GetKeyPolicy
  - ListAliases
  - ListGrants
  - ListKeys
  - RevokeGrant
  - ScheduleKeyDeletion
  - Tanda
  - TagResource

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkap dan pelajari cara menyiapkan dan menjalankan di Repositori Contoh Kode AWS.

```
        echo "\n";
        echo "-------------------------------------\n";
        echo <<<WELCOME
Welcome to the AWS Key Management Service SDK Basics scenario.

This program demonstrates how to interact with AWS Key Management Service using
 the AWS SDK for PHP (v3).
The AWS Key Management Service (KMS) is a secure and highly available service
 that allows you to create
and manage AWS KMS keys and control their use across a wide range of AWS services
 and applications.
KMS provides a centralized and unified approach to managing encryption keys,
 making it easier to meet your
data protection and regulatory compliance requirements.

This KMS Basics scenario creates two key types:
- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...\n
WELCOME;
        echo "-------------------------------------\n";
        $this->pressEnter();

        $this->kmsClient = new KmsClient([]);
        // Initialize the KmsService class with the client. This allows you to
 override any defaults in the client before giving it to the service class.
        $this->kmsService = new KmsService($this->kmsClient);

        // 1. Create a symmetric KMS key.
        echo "\n";
        echo "1. Create a symmetric KMS key.\n";
        echo "First, we will create a symmetric KMS key that is used to encrypt
 and decrypt data by invoking createKey().\n";
        $this->pressEnter();

        $key = $this->kmsService->createKey();
        $this->resources['symmetricKey'] = $key['KeyId'];
        echo "Created a customer key with ARN {$key['Arn']}.\n";
        $this->pressEnter();

        // 2. Enable a KMS key.
```

```
        echo "\n";
        echo "2. Enable a KMS key.\n";
        echo "By default when you create an AWS key, it is enabled. The code
 checks to
determine if the key is enabled. If it is not enabled, the code enables it.\n";
        $this->pressEnter();

        $keyInfo = $this->kmsService->describeKey($key['KeyId']);
        if(!$keyInfo['Enabled']){
            echo "The key was not enabled, so we will enable it.\n";
            $this->pressEnter();
            $this->kmsService->enableKey($key['KeyId']);
            echo "The key was successfully enabled.\n";
        }else{
            echo "The key was already enabled, so there was no need to enable it.
\n";
        }
        $this->pressEnter();

        // 3. Encrypt data using the symmetric KMS key.
        echo "\n";
        echo "3. Encrypt data using the symmetric KMS key.\n";
        echo "One of the main uses of symmetric keys is to encrypt and decrypt
 data.\n";
        echo "Next, we'll encrypt the string 'Hello, AWS KMS!' with the
 SYMMETRIC_DEFAULT encryption algorithm.\n";
        $this->pressEnter();
        $text = "Hello, AWS KMS!";
        $encryption = $this->kmsService->encrypt($key['KeyId'], $text);
        echo "The plaintext data was successfully encrypted with the algorithm:
 {$encryption['EncryptionAlgorithm']}.\n";
        $this->pressEnter();

        // 4. Create an alias.
        echo "\n";
        echo "4. Create an alias.\n";
        $aliasInput = testable_readline("Please enter an alias prefixed with
 \"alias/\" or press enter to use a default value: ");
        if($aliasInput == ""){
            $aliasInput = "alias/dev-encryption-key";
        }
        $this->kmsService->createAlias($key['KeyId'], $aliasInput);
        $this->resources['alias'] = $aliasInput;
        echo "The alias \"$aliasInput\" was successfully created.\n";
```

```
        $this->pressEnter();

        // 5. List all of your aliases.
        $aliasPageSize = 10;
        echo "\n";
        echo "5. List all of your aliases, up to $aliasPageSize.\n";
        $this->pressEnter();
        $aliasPaginator = $this->kmsService->listAliases();
        foreach($aliasPaginator as $pages){
            foreach($pages['Aliases'] as $alias){
                echo $alias['AliasName'] . "\n";
            }
            break;
        }
        $this->pressEnter();

        // 6. Enable automatic rotation of the KMS key.
        echo "\n";
        echo "6. Enable automatic rotation of the KMS key.\n";
        echo "By default, when the SDK enables automatic rotation of a KMS key,
KMS rotates the key material of the KMS key one year (approximately 365 days)
 from the enable date and every year
thereafter.";
        $this->pressEnter();
        $this->kmsService->enableKeyRotation($key['KeyId']);
        echo "The key's rotation was successfully set for key:
 {$key['KeyId']}\n";
        $this->pressEnter();

        // 7. Create a grant.
        echo "7. Create a grant.\n";
        echo "\n";
        echo "A grant is a policy instrument that allows Amazon Web Services
 principals to use KMS keys.
It also can allow them to view a KMS key (DescribeKey) and create and manage
 grants.
When authorizing access to a KMS key, grants are considered along with key
 policies and IAM policies.\n";
        $granteeARN = testable_readline("Please enter the Amazon Resource Name
 (ARN) of an Amazon Web Services principal. Valid principals include Amazon
 Web Services accounts, IAM users, IAM roles, federated users, and assumed
 role users. For help with the ARN syntax for a principal, see IAM ARNs in the
 Identity and Access Management User Guide. \nTo skip this step, press enter
 without any other values: ");
```

```
        if($granteeARN){
            $operations = [
                "ENCRYPT",
                "DECRYPT",
                "DESCRIBE_KEY",
            ];
            $grant = $this->kmsService->createGrant($key['KeyId'], $granteeARN,
$operations);
            echo "The grant Id is: {$grant['GrantId']}\n";
        }else{
            echo "Steps 7, 8, and 9 will be skipped.\n";
        }
        $this->pressEnter();

        // 8. List grants for the KMS key.
        if($granteeARN){
            echo "8. List grants for the KMS key.\n\n";
            $grantsPaginator = $this->kmsService->listGrants($key['KeyId']);
            foreach($grantsPaginator as $page){
                foreach($page['Grants'] as $grant){
                    echo $grant['GrantId'] . "\n";
                }
            }
        }else{
            echo "Skipping step 8...\n";
        }
        $this->pressEnter();

        // 9. Revoke the grant.
        if($granteeARN) {
            echo "\n";
            echo "9. Revoke the grant.\n";
            $this->pressEnter();
            $this->kmsService->revokeGrant($grant['GrantId'], $keyInfo['KeyId']);
            echo "{$grant['GrantId']} was successfully revoked!\n";
        }else{
            echo "Skipping step 9...\n";
        }
        $this->pressEnter();

        // 10. Decrypt the data.
        echo "\n";
        echo "10. Decrypt the data.\n";
        echo "Let's decrypt the data that was encrypted before.\n";
```

```
        echo "We'll use the same key to decrypt the string that we encrypted
 earlier in the program.\n";
        $this->pressEnter();
        $decryption = $this->kmsService->decrypt($keyInfo['KeyId'],
 $encryption['CiphertextBlob'], $encryption['EncryptionAlgorithm']);
        echo "The decrypted text is: {$decryption['Plaintext']}\n";
        $this->pressEnter();

        // 11. Replace a Key Policy.
        echo "\n";
        echo "11. Replace a Key Policy.\n";
        echo "A key policy is a resource policy for a KMS key. Key policies are
 the primary way to control access to KMS keys.\n";
        echo "Every KMS key must have exactly one key policy. The statements in
 the key policy determine who has permission to use the KMS key and how they can
 use it.\n";
        echo " You can also use IAM policies and grants to control access to the
 KMS key, but every KMS key must have a key policy.\n";
        echo "We will replace the key's policy with a new one:\n";
        $stsClient = new StsClient([]);
        $result = $stsClient->getCallerIdentity();
        $accountId = $result['Account'];
        $keyPolicy = <<< KEYPOLICY
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::$accountId:root"},
        "Action": "kms:*",
        "Resource": "*"
    }]
}
KEYPOLICY;
        echo $keyPolicy;
        $this->pressEnter();
        $this->kmsService->putKeyPolicy($keyInfo['KeyId'], $keyPolicy);
        echo "The Key Policy was successfully replaced!\n";
        $this->pressEnter();

        // 12. Retrieve the key policy.
        echo "\n";
        echo "12. Retrieve the key policy.\n";
        echo "Let's get some information about the new policy and print it to the
 screen.\n";
```

```
        $this->pressEnter();
        $policyInfo = $this->kmsService->getKeyPolicy($keyInfo['KeyId']);
        echo "We got the info! Here is the policy: \n";
        echo $policyInfo['Policy'] . "\n";
        $this->pressEnter();

        // 13. Create an asymmetric KMS key and sign data.
        echo "\n";
        echo "13. Create an asymmetric KMS key and sign data.\n";
        echo "Signing your data with an AWS key can provide several benefits that
make it an attractive option for your data signing needs.\n";
        echo "By using an AWS KMS key, you can leverage the security controls and
compliance features provided by AWS, which can help you meet various regulatory
requirements and enhance the overall security posture of your organization.\n";
        echo "First we'll create the asymmetric key.\n";
        $this->pressEnter();
        $keySpec = "RSA_2048";
        $keyUsage = "SIGN_VERIFY";
        $asymmetricKey = $this->kmsService->createKey($keySpec, $keyUsage);
        $this->resources['asymmetricKey'] = $asymmetricKey['KeyId'];
        echo "Created the key with ID: {$asymmetricKey['KeyId']}\n";
        echo "Next, we'll sign the data.\n";
        $this->pressEnter();
        $algorithm = "RSASSA_PSS_SHA_256";
        $sign = $this->kmsService->sign($asymmetricKey['KeyId'], $text,
$algorithm);
        $verify = $this->kmsService->verify($asymmetricKey['KeyId'], $text,
$sign['Signature'], $algorithm);
        echo "Signature verification result: {$sign['signature']}\n";
        $this->pressEnter();

        // 14. Tag the symmetric KMS key.
        echo "\n";
        echo "14. Tag the symmetric KMS key.\n";
        echo "By using tags, you can improve the overall management, security,
and governance of your KMS keys, making it easier to organize, track, and
control access to your encrypted data within your AWS environment.\n";
        echo "Let's tag our symmetric key as Environment->Production\n";
        $this->pressEnter();
        $this->kmsService->tagResource($key['KeyId'], [
            [
                'TagKey' => "Environment",
                'TagValue' => "Production",
            ],
```

```
        ]);
        echo "The key was successfully tagged!\n";
        $this->pressEnter();

        // 15. Schedule the deletion of the KMS key
        echo "\n";
        echo "15. Schedule the deletion of the KMS key.\n";
        echo "By default, KMS applies a waiting period of 30 days, but you can
 specify a waiting period of 7-30 days.\n";
        echo "When this operation is successful, the key state of the KMS key
 changes to PendingDeletion and the key can't be used in any cryptographic
 operations.\n";
        echo "It remains in this state for the duration of the waiting period.\n
\n";

        echo "Deleting a KMS key is a destructive and potentially dangerous
 operation. When a KMS key is deleted, all data that was encrypted under the KMS
 key is unrecoverable.\n\n";

        $cleanUp = testable_readline("Would you like to delete the resources
 created during this scenario, including the keys? (y/n): ");
        if($cleanUp == "Y" || $cleanUp == "y"){
            $this->cleanUp();
        }

        echo
 "--------------------------------------------------------------------------------
\n";
        echo "This concludes the AWS Key Management SDK Basics scenario\n";
        echo
 "--------------------------------------------------------------------------------
\n";



namespace Kms;

use Aws\Kms\Exception\KmsException;
use Aws\Kms\KmsClient;
use Aws\Result;
use Aws\ResultPaginator;
use AwsUtilities\AWSServiceClass;

class KmsService extends AWSServiceClass
```

```
{

    protected KmsClient $client;
    protected bool $verbose;

    /***
     * @param KmsClient|null $client
     * @param bool $verbose
     */
    public function __construct(KmsClient $client = null, bool $verbose = false)
    {
        $this->verbose = $verbose;
        if($client){
            $this->client = $client;
            return;
        }
        $this->client = new KmsClient([]);
    }


    /***
     * @param string $keySpec
     * @param string $keyUsage
     * @param string $description
     * @return array
     */
    public function createKey(string $keySpec = "", string $keyUsage = "", string
 $description = "Created by the SDK for PHP")
    {
        $parameters = ['Description' => $description];
        if($keySpec && $keyUsage){
            $parameters['KeySpec'] = $keySpec;
            $parameters['KeyUsage'] = $keyUsage;
        }
        try {
            $result = $this->client->createKey($parameters);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            // Check for error specific to createKey operations
            if ($caught->getAwsErrorMessage() == "LimitExceededException"){
                echo "The request was rejected because a quota was exceeded. For
 more information, see Quotas in the Key Management Service Developer Guide.";
            }
            throw $caught;
```

```
        }
    }



    /***
     * @param string $keyId
     * @param string $ciphertext
     * @param string $algorithm
     * @return Result
     */
    public function decrypt(string $keyId, string $ciphertext, string $algorithm
 = "SYMMETRIC_DEFAULT")
    {
        try{
            return $this->client->decrypt([
                'CiphertextBlob' => $ciphertext,
                'EncryptionAlgorithm' => $algorithm,
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem decrypting the data: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $text
     * @return Result
     */
    public function encrypt(string $keyId, string $text)
    {
        try {
            return $this->client->encrypt([
                'KeyId' => $keyId,
                'Plaintext' => $text,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "DisabledException"){
```

```
                echo "The request was rejected because the specified KMS key is
not enabled.\n";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param int $limit
     * @return ResultPaginator
     */
    public function listAliases(string $keyId = "", int $limit = 0)
    {
        $args = [];
        if($keyId){
            $args['KeyId'] = $keyId;
        }
        if($limit){
            $args['Limit'] = $limit;
        }
        try{
            return $this->client->getPaginator("ListAliases", $args);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidMarkerException"){
                echo "The request was rejected because the marker that specifies
where pagination should next begin is not valid.\n";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $alias
     * @return void
     */
    public function createAlias(string $keyId, string $alias)
    {
        try{
```

```
            $this->client->createAlias([
                'TargetKeyId' => $keyId,
                'AliasName' => $alias,
            ]);
        }catch (KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidAliasNameException"){
                echo "The request was rejected because the specified alias name
is not valid.";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $granteePrincipal
     * @param array $operations
     * @param array $grantTokens
     * @return Result
     */
    public function createGrant(string $keyId, string $granteePrincipal, array
$operations, array $grantTokens = [])
    {
        $args = [
            'KeyId' => $keyId,
            'GranteePrincipal' => $granteePrincipal,
            'Operations' => $operations,
        ];
        if($grantTokens){
            $args['GrantTokens'] = $grantTokens;
        }
        try{
            return $this->client->createGrant($args);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidGrantTokenException"){
                echo "The request was rejected because the specified grant token
is not valid.\n";
            }
            throw $caught;
        }
    }
```

```
    /***
     * @param string $keyId
     * @return array
     */
    public function describeKey(string $keyId)
    {
        try {
            $result = $this->client->describeKey([
                "KeyId" => $keyId,
            ]);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
 resource could not be found.\n";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @return void
     */
    public function disableKey(string $keyId)
    {
        try {
            $this->client->disableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem disabling the key: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
```

```
     * @param string $keyId
     * @return void
     */
    public function enableKey(string $keyId)
    {
        try {
            $this->client->enableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
 resource could not be found.\n";
            }
            throw $caught;
        }
    }



    /***
     * @return array
     */
    public function listKeys()
    {
        try {
            $contents = [];
            $paginator = $this->client->getPaginator("ListKeys");
            foreach($paginator as $result){
                foreach ($result['Content'] as $object) {
                    $contents[] = $object;
                }
            }
            return $contents;
        }catch(KmsException $caught){
            echo "There was a problem listing the keys: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
```

```
     * @param string $keyId
     * @return Result
     */
    public function listGrants(string $keyId)
    {
        try{
            return $this->client->listGrants([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "    The request was rejected because the specified entity
 or resource could not be found.\n";
            }
            throw $caught;
        }
    }


    /***
     * @param string $keyId
     * @return Result
     */
    public function getKeyPolicy(string $keyId)
    {
        try {
            return $this->client->getKeyPolicy([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem getting the key policy: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }


    /***
     * @param string $grantId
     * @param string $keyId
     * @return void
     */
    public function revokeGrant(string $grantId, string $keyId)
    {
```

```
        try{
            $this->client->revokeGrant([
                'GrantId' => $grantId,
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem with revoking the grant: {$caught-
>getAwsErrorMessage()}.\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param int $pendingWindowInDays
     * @return void
     */
    public function scheduleKeyDeletion(string $keyId, int $pendingWindowInDays =
 7)
    {
        try {
            $this->client->scheduleKeyDeletion([
                'KeyId' => $keyId,
                'PendingWindowInDays' => $pendingWindowInDays,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem scheduling the key deletion: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param array $tags
     * @return void
     */
    public function tagResource(string $keyId, array $tags)
    {
        try {
```

```
            $this->client->tagResource([
                'KeyId' => $keyId,
                'Tags' => $tags,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem applying the tag(s): {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $message
     * @param string $algorithm
     * @return Result
     */
    public function sign(string $keyId, string $message, string $algorithm)
    {
        try {
            return $this->client->sign([
                'KeyId' => $keyId,
                'Message' => $message,
                'SigningAlgorithm' => $algorithm,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem signing the data: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param int $rotationPeriodInDays
     * @return void
     */
    public function enableKeyRotation(string $keyId, int $rotationPeriodInDays =
 365)
    {
```

```
        try{
            $this->client->enableKeyRotation([
                'KeyId' => $keyId,
                'RotationPeriodInDays' => $rotationPeriodInDays,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
 resource could not be found.\n";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $policy
     * @return void
     */
    public function putKeyPolicy(string $keyId, string $policy)
    {
        try {
            $this->client->putKeyPolicy([
                'KeyId' => $keyId,
                'Policy' => $policy,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem replacing the key policy: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $aliasName
     * @return void
     */
    public function deleteAlias(string $aliasName)
    {
        try {
```

```
            $this->client->deleteAlias([
                'AliasName' => $aliasName,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem deleting the alias: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $message
     * @param string $signature
     * @param string $signingAlgorithm
     * @return bool
     */
    public function verify(string $keyId, string $message, string $signature,
 string $signingAlgorithm)
    {
        try {
            $result = $this->client->verify([
                'KeyId' => $keyId,
                'Message' => $message,
                'Signature' => $signature,
                'SigningAlgorithm' => $signingAlgorithm,
            ]);
            return $result['SignatureValid'];
        }catch(KmsException $caught){
            echo "There was a problem verifying the signature: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }


}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK untuk PHP .

- CreateAlias
- CreateGrant
- CreateKey
- Dekripsi
- DescribeKey
- DisableKey
- EnableKey
- Enkripsi
- GetKeyPolicy
- ListAliases
- ListGrants
- ListKeys
- RevokeGrant
- ScheduleKeyDeletion
- Tanda
- TagResource

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KMSScenario:
    """Runs an interactive scenario that shows how to get started with KMS."""

    def __init__(
        self,
        key_manager: KeyManager,
        key_encryption: KeyEncrypt,
        alias_manager: AliasManager,
```

```
        grant_manager: GrantManager,
        key_policy: KeyPolicy,
    ):
        self.key_manager = key_manager
        self.key_encryption = key_encryption
        self.alias_manager = alias_manager
        self.grant_manager = grant_manager
        self.key_policy = key_policy
        self.key_id = ""
        self.alias_name = ""
        self.asymmetric_key_id = ""

    def kms_scenario(self):
        key_description = "Created by the AWS KMS API"

        print(DASHES)
        print(
            """
Welcome to the AWS Key Management SDK Basics scenario.

This program demonstrates how to interact with AWS Key Management using the AWS
 SDK for Python (Boto3).
The AWS Key Management Service (KMS) is a secure and highly available service
 that allows you to create
and manage AWS KMS keys and control their use across a wide range of AWS services
 and applications.
KMS provides a centralized and unified approach to managing encryption keys,
 making it easier to meet your
data protection and regulatory compliance requirements.

This Basics scenario creates two key types:

- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...
        """
        )
        q.ask("Press Enter to continue...")

        print(DASHES)
        print(f"1. Create a symmetric KMS key\n")
        print(
```

```
            f"First, the program will creates a symmetric KMS key that you can
 used to encrypt and decrypt data."
        )
        q.ask("Press Enter to continue...")
        self.key_id = self.key_manager.create_key(key_description)["KeyId"]
        print(f"A symmetric key was successfully created {self.key_id}.")
        q.ask("Press Enter to continue...")
        print(DASHES)
        print(
            """
2. Enable a KMS key

By default, when the SDK creates an AWS key, it is enabled. The next bit of code
 checks to
determine if the key is enabled.
        """
        )
        q.ask("Press Enter to continue...")
        is_enabled = self.is_key_enabled(self.key_id)
        print(f"Is the key enabled? {is_enabled}")
        if not is_enabled:
            self.key_manager.enable_key(self.key_id)
        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"3. Encrypt data using the symmetric KMS key")
        plain_text = "Hello, AWS KMS!"
        print(
            f"""
One of the main uses of symmetric keys is to encrypt and decrypt data.
Next, the code encrypts the string "{plain_text}" with the SYMMETRIC_DEFAULT
 encryption algorithm.
        """
        )
        q.ask("Press Enter to continue...")
        encrypted_text = self.key_encryption.encrypt(self.key_id, plain_text)
        print(DASHES)
        print(f"4. Create an alias")
        print(
            """
Now, the program will create an alias for the KMS key. An alias is a friendly
 name that you
can associate with a KMS key. The alias name should be prefixed with 'alias/'.
        """
        )
```

```
        alias_name = q.ask("Enter an alias name: ", q.non_empty)
        self.alias_manager.create_alias(self.key_id, alias_name)
        print(f"{alias_name} was successfully created.")
        self.alias_name = alias_name
        print(DASHES)
        print(f"5. List all of your aliases")
        q.ask("Press Enter to continue...")
        self.alias_manager.list_aliases(10)
        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"6. Enable automatic rotation of the KMS key")
        print(
            """

By default, when the SDK enables automatic rotation of a KMS key,
KMS rotates the key material of the KMS key one year (approximately 365 days)
 from the enable date and every year
thereafter.
        """
        )
        q.ask("Press Enter to continue...")
        self.key_manager.enable_key_rotation(self.key_id)
        print(DASHES)
        print(f"Key rotation has been enabled for key with id {self.key_id}")
        print(
            """
7. Create a grant

A grant is a policy instrument that allows Amazon Web Services principals to use
 KMS keys.
It also can allow them to view a KMS key (DescribeKey) and create and manage
 grants.
When authorizing access to a KMS key, grants are considered along with key
 policies and IAM policies.
        """
        )
        print(
            """
To create a grant you must specify a account_id. To specify the grantee
 account_id, use the Amazon Resource Name (ARN)
of an AWS account_id. Valid principals include AWS accounts, IAM users, IAM
 roles, federated users,
and assumed role users.
        """
```

```
        )
        account_id = q.ask(
            "Enter an account_id, or press enter to skip creating a grant... "
        )
        grant = None
        if account_id != "":
            grant = self.grant_manager.create_grant(
                self.key_id,
                account_id,
                [
                    "Encrypt",
                    "Decrypt",
                    "DescribeKey",
                ],
            )
            print(f"Grant created successfully with ID: {grant['GrantId']}")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(DASHES)
        print(f"8. List grants for the KMS key")
        q.ask("Press Enter to continue...")
        self.grant_manager.list_grants(self.key_id)
        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"9. Revoke the grant")
        print(
            """
The revocation of a grant immediately removes the permissions and access that the
 grant had provided.
This means that any account_id (user, role, or service) that was granted access
 to perform specific
KMS operations on a KMS key will no longer be able to perform those operations.
        """
        )
        q.ask("Press Enter to continue...")

        if grant is not None:
            self.grant_manager.revoke_grant(self.key_id, grant["GrantId"])
            print(f"Grant ID: {grant['GrantId']} was successfully revoked!")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"10. Decrypt the data\n")
```

```
        print(
            """
Lets decrypt the data that was encrypted in an early step.
The code uses the same key to decrypt the string that we encrypted earlier in the
 program.
        """
        )
        q.ask("Press Enter to continue...")
        decrypted_data = self.key_encryption.decrypt(self.key_id, encrypted_text)
        print(f"Data decrypted successfully for key ID: {self.key_id}")
        print(f"Decrypted data: {decrypted_data}")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"11. Replace a key policy\n")
        print(
            """
A key policy is a resource policy for a KMS key. Key policies are the primary way
 to control
access to KMS keys. Every KMS key must have exactly one key policy. The
 statements in the key policy
determine who has permission to use the KMS key and how they can use it.
You can also use IAM policies and grants to control access to the KMS key, but
 every KMS key
must have a key policy.

By default, when you create a key by using the SDK, a policy is created that
gives the AWS account that owns the KMS key full access to the KMS key.

Let's try to replace the automatically created policy with the following policy.
{
"Version":"2012-10-17",
"Statement": [{
"Effect": "Allow",
"Principal": {"AWS": "arn:aws:iam::0000000000:root"},
"Action": "kms:*",
"Resource": "*"
}]
}
        """
        )
        account_id = q.ask("Enter your account ID or press enter to skip: ")
        if account_id != "":
            policy = {
```

```
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {"AWS": f"arn:aws:iam::{account_id}:root"},
                        "Action": "kms:*",
                        "Resource": "*",
                    }
                ],
            }

            self.key_policy.set_new_policy(self.key_id, policy)
            print("Key policy replacement succeeded.")
            q.ask("Press Enter to continue...")
        else:
            print("Skipping replacing the key policy.")

        print(DASHES)
        print(f"12. Get the key policy\n")
        print(
            f"The next bit of code that runs gets the key policy to make sure it
exists."
        )
        q.ask("Press Enter to continue...")
        policy = self.key_policy.get_policy(self.key_id)
        print(f"The key policy is: {policy}")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"13. Create an asymmetric KMS key and sign your data\n")
        print(
            """
        Signing your data with an AWS key can provide several benefits that make
it an attractive option
        for your data signing needs. By using an AWS KMS key, you can leverage
the
        security controls and compliance features provided by AWS,
        which can help you meet various regulatory requirements and enhance the
overall security posture
        of your organization.
        """
        )
        q.ask("Press Enter to continue...")
        print(f"Sign and verify data operation succeeded.")
```

```
        self.asymmetric_key_id = self.key_manager.create_asymmetric_key()
        message = "Here is the message that will be digitally signed"
        signature = self.key_encryption.sign(self.asymmetric_key_id, message)
        if self.key_encryption.verify(self.asymmetric_key_id, message,
signature):
            print("Signature verification succeeded.")
        else:
            print("Signature verification failed.")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"14. Tag your symmetric KMS Key\n")
        print(
            """
        By using tags, you can improve the overall management, security, and
governance of your
        KMS keys, making it easier to organize, track, and control access to your
encrypted data within
        your AWS environment
        """
        )
        q.ask("Press Enter to continue...")
        self.key_manager.tag_resource(self.key_id, "Environment", "Production")
        self.clean_up()

    def is_key_enabled(self, key_id: str) -> bool:
        """
        Check if the key is enabled or not.

        :param key_id: The key to check.
        :return: True if the key is enabled, otherwise False.
        """
        response = self.key_manager.describe_key(key_id)
        return response["Enabled"] is True

    def clean_up(self):
        """
        Delete resources created by this scenario.
        """
        if self.alias_name != "":
            print(f"Deleting the alias {self.alias_name}.")
            self.alias_manager.delete_alias(self.alias_name)
        window = 7  # The window in days for a scheduled deletion.
        if self.key_id != "":
```

```
            print(
                """
Warning:
Deleting a KMS key is a destructive and potentially dangerous operation. When a
 KMS key is deleted,
all data that was encrypted under the KMS key is unrecoverable.
                """
            )
            if q.ask(
                f"Do you want to delete the key with ID {self.key_id} (y/n)?",
                q.is_yesno,
            ):
                print(
                    f"The key {self.key_id} will be deleted with a window of
 {window} days. You can cancel the deletion before"
                )
                print("the window expires.")
                self.key_manager.delete_key(self.key_id, window)
                self.key_id = ""

        if self.asymmetric_key_id != "":
            if q.ask(
                f"Do you want to delete the asymmetric key with ID
 {self.asymmetric_key_id} (y/n)?",
                q.is_yesno,
            ):
                print(
                    f"The key {self.asymmetric_key_id} will be deleted with a
 window of {window} days. You can cancel the deletion before"
                )
                print("the window expires.")
                self.key_manager.delete_key(self.asymmetric_key_id, window)
                self.asymmetric_key_id = ""


if __name__ == "__main__":
    kms_scenario = None
    try:
        kms_client = boto3.client("kms")
        a_key_manager = KeyManager(kms_client)
        a_key_encrypt = KeyEncrypt(kms_client)
        an_alias_manager = AliasManager(kms_client)
        a_grant_manager = GrantManager(kms_client)
        a_key_policy = KeyPolicy(kms_client)
```

```
        kms_scenario = KMSScenario(
            key_manager=a_key_manager,
            key_encryption=a_key_encrypt,
            alias_manager=an_alias_manager,
            grant_manager=a_grant_manager,
            key_policy=a_key_policy,
        )
        kms_scenario.kms_scenario()
    except Exception:
        logging.exception("Something went wrong with the demo!")
        if kms_scenario is not None:
            kms_scenario.clean_up()
```

Kelas pembungkus dan metode untuk manajemen kunci KMS.

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_key(self, key_description: str) -> dict[str, any]:
        """
        Creates a key with a user-provided description.

        :param key_description: A description for the key.
        :return: The key ID.
        """
        try:
```

```
            key = self.kms_client.create_key(Description=key_description)
["KeyMetadata"]
            self.created_keys.append(key)
            return key
        except ClientError as err:
            logging.error(
                "Couldn't create your key. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def describe_key(self, key_id: str) -> dict[str, any]:
        """
        Describes a key.

        :param key_id: The ARN or ID of the key to describe.
        :return: Information about the key.
        """

        try:
            key = self.kms_client.describe_key(KeyId=key_id)["KeyMetadata"]
            return key
        except ClientError as err:
            logging.error(
                "Couldn't get key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise


    def enable_key_rotation(self, key_id: str) -> None:
        """
        Enables rotation for a key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        """
        try:
            self.kms_client.enable_key_rotation(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't enable rotation for key '%s'. Here's why: %s",
                key_id,
```

```
                err.response["Error"]["Message"],
            )
            raise


    def create_asymmetric_key(self) -> str:
        """
        Creates an asymmetric key in AWS KMS for signing messages.

        :return: The ID of the created key.
        """
        try:
            key = self.kms_client.create_key(
                KeySpec="RSA_2048", KeyUsage="SIGN_VERIFY", Origin="AWS_KMS"
            )["KeyMetadata"]
            self.created_keys.append(key)
            return key["KeyId"]
        except ClientError as err:
            logger.error(
                "Couldn't create your key. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def tag_resource(self, key_id: str, tag_key: str, tag_value: str) -> None:
        """
        Add or edit tags on a customer managed key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        :param tag_key: Key for the tag.
        :param tag_value: Value for the tag.
        """
        try:
            self.kms_client.tag_resource(
                KeyId=key_id, Tags=[{"TagKey": tag_key, "TagValue": tag_value}]
            )
        except ClientError as err:
            logging.error(
                "Couldn't add a tag for the key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

```
    def delete_key(self, key_id: str, window: int) -> None:
        """
        Deletes a list of keys.

        Warning:
        Deleting a KMS key is a destructive and potentially dangerous operation.
 When a KMS key is deleted,
        all data that was encrypted under the KMS key is unrecoverable.

        :param key_id: The ARN or ID of the key to delete.
        :param window: The waiting period, in days, before the KMS key is
 deleted.
        """

        try:
            self.kms_client.schedule_key_deletion(
                KeyId=key_id, PendingWindowInDays=window
            )
        except ClientError as err:
            logging.error(
                "Couldn't delete key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

Kelas pembungkus dan metode untuk alias kunci KMS.

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.
```

```
        :return: An instance of AliasManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_alias(self, key_id: str, alias: str) -> None:
        """
        Creates an alias for the specified key.

        :param key_id: The ARN or ID of a key to give an alias.
        :param alias: The alias to assign to the key.
        """
        try:
            self.kms_client.create_alias(AliasName=alias, TargetKeyId=key_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "AlreadyExistsException":
                logger.error(
                    "Could not create the alias %s because it already exists.",
key_id
                )
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
                raise


    def list_aliases(self, page_size: int) -> None:
        """
        Lists aliases for the current account.
        :param page_size: The number of aliases to list per page.
        """
        try:
            alias_paginator = self.kms_client.get_paginator("list_aliases")
            for alias_page in alias_paginator.paginate(
                PaginationConfig={"PageSize": page_size}
            ):
                print(f"Here are {page_size} aliases:")
                pprint(alias_page["Aliases"])
                if alias_page["Truncated"]:
                    answer = input(
```

```
                        f"Do you want to see the next {page_size} aliases (y/n)?
 "
                    )
                    if answer.lower() != "y":
                        break
                else:
                    print("That's all your aliases!")
        except ClientError as err:
            logging.error(
                "Couldn't list your aliases. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def delete_alias(self, alias: str) -> None:
        """
        Deletes an alias.

        :param alias: The alias to delete.
        """
        try:
            self.kms_client.delete_alias(AliasName=alias)
        except ClientError as err:
            logger.error(
                "Couldn't delete alias %s. Here's why: %s",
                alias,
                err.response["Error"]["Message"],
            )
            raise
```

Kelas pembungkus dan metode untuk enkripsi kunci KMS.

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
```

```
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def encrypt(self, key_id: str, text: str) -> bytes:
        """
        Encrypts text by using the specified key.

        :param key_id: The ARN or ID of the key to use for encryption.
        :param text: The text to encrypt.
        :return: The encrypted version of the text.
        """
        try:
            response = self.kms_client.encrypt(KeyId=key_id,
Plaintext=text.encode())
            print(
                f"The string was encrypted with algorithm
{response['EncryptionAlgorithm']}"
            )
            return response["CiphertextBlob"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DisabledException":
                logger.error(
                    "Could not encrypt because the key %s is disabled.", key_id
                )
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            raise


    def decrypt(self, key_id: str, cipher_text: bytes) -> str:
        """
        Decrypts text previously encrypted with a key.

        :param key_id: The ARN or ID of the key used to decrypt the data.
        :param cipher_text: The encrypted text to decrypt.
```

```
        :return: The decrypted text.
        """
        try:
            return self.kms_client.decrypt(KeyId=key_id,
CiphertextBlob=cipher_text)[
                "Plaintext"
            ].decode()
        except ClientError as err:
            logger.error(
                "Couldn't decrypt your ciphertext. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def sign(self, key_id: str, message: str) -> str:
        """
        Signs a message with a key.

        :param key_id: The ARN or ID of the key to use for signing.
        :param message: The message to sign.
        :return: The signature of the message.
        """
        try:
            return self.kms_client.sign(
                KeyId=key_id,
                Message=message.encode(),
                SigningAlgorithm="RSASSA_PSS_SHA_256",
            )["Signature"]
        except ClientError as err:
            logger.error(
                "Couldn't sign your message. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def verify(self, key_id: str, message: str, signature: str) -> bool:
        """
        Verifies a signature against a message.

        :param key_id: The ARN or ID of the key used to sign the message.
        :param message: The message to verify.
        :param signature: The signature to verify.
```

```
        :return: True when the signature matches the message, otherwise False.
        """
        try:
            response = self.kms_client.verify(
                KeyId=key_id,
                Message=message.encode(),
                Signature=signature,
                SigningAlgorithm="RSASSA_PSS_SHA_256",
            )
            valid = response["SignatureValid"]
            print(f"The signature is {'valid' if valid else 'invalid'}.")
            return valid
        except ClientError as err:
            if err.response["Error"]["Code"] == "SignatureDoesNotMatchException":
                print("The signature is not valid.")
            else:
                logger.error(
                    "Couldn't verify your signature. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            raise
```

Kelas pembungkus dan metode untuk hibah kunci KMS.

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_grant(
```

```
        self, key_id: str, principal: str, operations: [str]
    ) -> dict[str, str]:
        """
        Creates a grant for a key that lets a principal generate a symmetric data
        encryption key.

        :param key_id: The ARN or ID of the key.
        :param principal: The principal to grant permission to.
        :param operations: The operations to grant permission for.
        :return: The grant that is created.
        """
        try:
            return self.kms_client.create_grant(
                KeyId=key_id,
                GranteePrincipal=principal,
                Operations=operations,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create a grant on key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise


    def list_grants(self, key_id):
        """
        Lists grants for a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The grants for the key.
        """
        try:
            paginator = self.kms_client.get_paginator("list_grants")
            grants = []
            page_iterator = paginator.paginate(KeyId=key_id)
            for page in page_iterator:
                grants.extend(page["Grants"])

            print(f"Grants for key {key_id}:")
            pprint(grants)
            return grants
        except ClientError as err:
```

```
            logger.error(
                "Couldn't list grants for key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise


    def revoke_grant(self, key_id: str, grant_id: str) -> None:
        """
        Revokes a grant so that it can no longer be used.

        :param key_id: The ARN or ID of the key associated with the grant.
        :param grant_id: The ID of the grant to revoke.
        """
        try:
            self.kms_client.revoke_grant(KeyId=key_id, GrantId=grant_id)
        except ClientError as err:
            logger.error(
                "Couldn't revoke grant %s. Here's why: %s",
                grant_id,
                err.response["Error"]["Message"],
            )
            raise
```

Kelas pembungkus dan metode untuk kebijakan kunci KMS.

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
```

```
        return cls(kms_client)


    def set_new_policy(self, key_id: str, policy: dict[str, any]) -> None:
        """
        Sets the policy of a key. Setting a policy entirely overwrites the
existing
        policy, so care is taken to add a statement to the existing list of
statements
        rather than simply writing a new policy.

        :param key_id: The ARN or ID of the key to set the policy to.
        :param policy: A new key policy. The key policy must allow the calling
principal to make a subsequent
                       PutKeyPolicy request on the KMS key. This reduces the risk
that the KMS key becomes unmanageable
        """

        try:
            self.kms_client.put_key_policy(KeyId=key_id,
Policy=json.dumps(policy))
        except ClientError as err:
            logger.error(
                "Couldn't set policy for key %s. Here's why %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise



    def get_policy(self, key_id: str) -> dict[str, str]:
        """
        Gets the policy of a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The key policy as a dict.
        """
        if key_id != "":
            try:
                response = self.kms_client.get_key_policy(
                    KeyId=key_id,
                )
                policy = json.loads(response["Policy"])
```

```
            except ClientError as err:
                logger.error(
                    "Couldn't get policy for key %s. Here's why: %s",
                    key_id,
                    err.response["Error"]["Message"],
                )
                raise
            else:
                pprint(policy)
                return policy
        else:
            print("Skipping get policy demo.")
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK untuk Python (Boto3).
  - CreateAlias
  - CreateGrant
  - CreateKey
  - Dekripsi
  - DescribeKey
  - DisableKey
  - EnableKey
  - Enkripsi
  - GetKeyPolicy
  - ListAliases
  - ListGrants
  - ListKeys
  - RevokeGrant
  - ScheduleKeyDeletion
  - Tanda
  - TagResource

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Tindakan untuk AWS KMS menggunakan AWS SDKs

Contoh kode berikut menunjukkan bagaimana melakukan AWS KMS tindakan individu dengan AWS SDKs. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Kutipan ini memanggil AWS KMS API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Anda dapat melihat tindakan dalam konteks diSkenario untuk AWS KMS menggunakan AWS SDKs.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi AWS Key Management Service API.

Contoh

- Gunakan CreateAlias dengan AWS SDK atau CLI
- Gunakan CreateGrant dengan AWS SDK atau CLI
- Gunakan CreateKey dengan AWS SDK atau CLI
- Gunakan Decrypt dengan AWS SDK atau CLI
- Gunakan DeleteAlias dengan AWS SDK atau CLI
- Gunakan DescribeKey dengan AWS SDK atau CLI
- Gunakan DisableKey dengan AWS SDK atau CLI
- Gunakan EnableKey dengan AWS SDK atau CLI
- Gunakan EnableKeyRotation dengan AWS SDK atau CLI
- Gunakan Encrypt dengan AWS SDK atau CLI
- Gunakan GenerateDataKey dengan AWS SDK atau CLI
- Gunakan GenerateDataKeyWithoutPlaintext dengan AWS SDK atau CLI
- Gunakan GenerateRandom dengan AWS SDK atau CLI
- Gunakan GetKeyPolicy dengan AWS SDK atau CLI
- Gunakan ListAliases dengan AWS SDK atau CLI

- Gunakan ListGrants dengan AWS SDK atau CLI
- Gunakan ListKeyPolicies dengan AWS SDK atau CLI
- Gunakan ListKeys dengan AWS SDK atau CLI
- Gunakan PutKeyPolicy dengan AWS SDK atau CLI
- Gunakan ReEncrypt dengan AWS SDK atau CLI
- Gunakan RetireGrant dengan AWS SDK atau CLI
- Gunakan RevokeGrant dengan AWS SDK atau CLI
- Gunakan ScheduleKeyDeletion dengan AWS SDK atau CLI
- Gunakan Sign dengan AWS SDK atau CLI
- Gunakan TagResource dengan AWS SDK atau CLI
- Gunakan UpdateAlias dengan AWS SDK atau CLI
- Gunakan Verify dengan AWS SDK atau CLI

Gunakan **`CreateAlias`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateAlias`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
using System;
using System.Threading.Tasks;
```

```
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Creates an alias for an AWS Key Management Service (AWS KMS) key.
/// </summary>
public class CreateAlias
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The alias name must start with alias/ and can be
        // up to 256 alphanumeric characters long.
        var aliasName = "alias/ExampleAlias";

        // The value supplied as the TargetKeyId can be either
        // the key ID or key Amazon Resource Name (ARN) of the
        // AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

        var request = new CreateAliasRequest
        {
            AliasName = aliasName,
            TargetKeyId = keyId,
        };

        var response = await client.CreateAliasAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"Alias, {aliasName}, successfully created.");
        }
        else
        {
            Console.WriteLine($"Could not create alias.");
        }
    }
}
```

- Untuk detail API, lihat CreateAliasdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk membuat alias untuk kunci KMS

`create-alias`Perintah berikut membuat alias bernama `example-alias` untuk kunci KMS diidentifikasi oleh ID kunci. `1234abcd-12ab-34cd-56ef-1234567890ab`

Nama alias harus dimulai dengan`alias/`. Jangan gunakan nama alias yang dimulai dengan`alias/aws`; ini dicadangkan untuk digunakan oleh AWS.

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

Perintah ini tidak mengembalikan output apa pun. Untuk melihat alias baru, gunakan `list-aliases` perintah.

Untuk informasi selengkapnya, lihat Menggunakan alias di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat CreateAliasdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /**
     * Creates a custom alias for the specified target key asynchronously.
     *
     * @param targetKeyId the ID of the target key for the alias
     * @param aliasName   the name of the alias to create
     * @return a {@link CompletableFuture} that completes when the alias creation
operation is finished
```

```
     */
    public CompletableFuture<Void> createCustomAliasAsync(String targetKeyId,
String aliasName) {
        CreateAliasRequest aliasRequest = CreateAliasRequest.builder()
            .aliasName(aliasName)
            .targetKeyId(targetKeyId)
            .build();

        CompletableFuture<CreateAliasResponse> responseFuture =
getAsyncClient().createAlias(aliasRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("{} was successfully created.", aliasName);
            } else {
                if (exception instanceof ResourceExistsException) {
                    logger.info("Alias [{}] already exists. Moving on...",
aliasName);
                } else if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while creating
alias: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while creating alias: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(response -> null);
    }
```

- Untuk detail API, lihat CreateAliasdi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun createCustomAlias(
    targetKeyIdVal: String?,
    aliasNameVal: String?,
) {
    val request =
        CreateAliasRequest {
            aliasName = aliasNameVal
            targetKeyId = targetKeyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        kmsClient.createAlias(request)
        println("$aliasNameVal was successfully created")
    }
}
```

- Untuk detail API, lihat CreateAliasdi AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @param string $alias
     * @return void
     */
    public function createAlias(string $keyId, string $alias)
    {
        try{
            $this->client->createAlias([
                'TargetKeyId' => $keyId,
                'AliasName' => $alias,
```

```
            ]);
        }catch (KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidAliasNameException"){
                echo "The request was rejected because the specified alias name
 is not valid.";
            }
            throw $caught;
        }
    }
```

- Untuk detail API, lihat CreateAliasdi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_alias(self, key_id: str, alias: str) -> None:
```

```
        """
        Creates an alias for the specified key.

        :param key_id: The ARN or ID of a key to give an alias.
        :param alias: The alias to assign to the key.
        """
        try:
            self.kms_client.create_alias(AliasName=alias, TargetKeyId=key_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "AlreadyExistsException":
                logger.error(
                    "Could not create the alias %s because it already exists.",
key_id
                )
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
                raise
```

- Untuk detail API, lihat CreateAliasdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_alias_name = 'alias/my-key-alias'
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->createalias(
          iv_aliasname = iv_alias_name
          iv_targetkeyid = iv_key_id
```

```
        ).
        MESSAGE 'Alias created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsalreadyexistsex.
        MESSAGE 'Alias already exists.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvalidaliasnameex.
        MESSAGE 'Invalid alias name.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateAliasdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`CreateGrant`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateGrant`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        public static async Task Main()
        {
```

```
            var client = new AmazonKeyManagementServiceClient();

            // The identity that is given permission to perform the operations
            // specified in the grant.
            var grantee = "arn:aws:iam::111122223333:role/ExampleRole";

            // The identifier of the AWS KMS key to which the grant applies. You
            // can use the key ID or the Amazon Resource Name (ARN) of the KMS
key.
            var keyId = "7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";

            var request = new CreateGrantRequest
            {
                GranteePrincipal = grantee,
                KeyId = keyId,

                // A list of operations that the grant allows.
                Operations = new List<string>
                {
                    "Encrypt",
                    "Decrypt",
                },
            };

            var response = await client.CreateGrantAsync(request);

            string grantId = response.GrantId; // The unique identifier of the
grant.
            string grantToken = response.GrantToken; // The grant token.

            Console.WriteLine($"Id: {grantId}, Token: {grantToken}");
        }
    }
```

- Untuk detail API, lihat CreateGrantdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk membuat hibah

`create-grant`Contoh berikut membuat hibah yang memungkinkan `exampleUser` pengguna untuk menggunakan `decrypt` perintah pada kunci KMS `1234abcd-12ab-34cd-56ef-1234567890ab` contoh. Kepala sekolah yang pensiun adalah `adminRole` perannya. Hibah menggunakan batasan `EncryptionContextSubset` hibah untuk mengizinkan izin ini hanya jika konteks enkripsi dalam `decrypt` permintaan menyertakan pasangan nilai `"Department": "IT"` kunci.

```
aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::123456789012:user/exampleUser \
    --operations Decrypt \
    --constraints EncryptionContextSubset={Department=IT} \
    --retiring-principal arn:aws:iam::123456789012:role/adminRole
```

Output:

```
{
    "GrantId":
 "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
    "GrantToken": "<grant token here>"
}
```

Untuk melihat informasi terperinci tentang hibah, gunakan `list-grants` perintah.

Untuk informasi selengkapnya, lihat Hibah di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Utama.

- Untuk detail API, lihat CreateGrantdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /**
```

```
    * Grants permissions to a specified principal on a customer master key (CMK)
asynchronously.
    *
    * @param keyId             The unique identifier for the customer master key
(CMK) that the grant applies to.
    * @param granteePrincipal  The principal that is given permission to perform
the operations that the grant permits on the CMK.
    * @return A {@link CompletableFuture} that, when completed, contains the ID
of the created grant.
    * @throws RuntimeException If an error occurs during the grant creation
process.
    */
   public CompletableFuture<String> grantKeyAsync(String keyId, String
granteePrincipal) {
       List<GrantOperation> grantPermissions = List.of(
           GrantOperation.ENCRYPT,
           GrantOperation.DECRYPT,
           GrantOperation.DESCRIBE_KEY
       );

       CreateGrantRequest grantRequest = CreateGrantRequest.builder()
           .keyId(keyId)
           .name("grant1")
           .granteePrincipal(granteePrincipal)
           .operations(grantPermissions)
           .build();

       CompletableFuture<CreateGrantResponse> responseFuture =
getAsyncClient().createGrant(grantRequest);
       responseFuture.whenComplete((response, ex) -> {
           if (ex == null) {
               logger.info("Grant created successfully with ID: " +
response.grantId());
           } else {
               if (ex instanceof KmsException kmsEx) {
                   throw new RuntimeException("Failed to create grant: " +
kmsEx.getMessage(), kmsEx);
               } else {
                   throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage(), ex);
               }
           }
       });
```

```
        return responseFuture.thenApply(CreateGrantResponse::grantId);
    }
```

- Untuk detail API, lihat CreateGrantdi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun createNewGrant(
    keyIdVal: String?,
    granteePrincipalVal: String?,
    operation: String,
): String? {
    val operationOb = GrantOperation.fromValue(operation)
    val grantOperationList = ArrayList<GrantOperation>()
    grantOperationList.add(operationOb)

    val request =
        CreateGrantRequest {
            keyId = keyIdVal
            granteePrincipal = granteePrincipalVal
            operations = grantOperationList
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.createGrant(request)
        return response.grantId
    }
}
```

- Untuk detail API, lihat CreateGrantdi AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /***
    * @param string $keyId
    * @param string $granteePrincipal
    * @param array $operations
    * @param array $grantTokens
    * @return Result
    */
   public function createGrant(string $keyId, string $granteePrincipal, array
$operations, array $grantTokens = [])
   {
       $args = [
           'KeyId' => $keyId,
           'GranteePrincipal' => $granteePrincipal,
           'Operations' => $operations,
       ];
       if($grantTokens){
           $args['GrantTokens'] = $grantTokens;
       }
       try{
           return $this->client->createGrant($args);
       }catch(KmsException $caught){
           if($caught->getAwsErrorMessage() == "InvalidGrantTokenException"){
               echo "The request was rejected because the specified grant token
is not valid.\n";
           }
           throw $caught;
       }
   }
```

- Untuk detail API, lihat CreateGrantdi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_grant(
        self, key_id: str, principal: str, operations: [str]
    ) -> dict[str, str]:
        """
        Creates a grant for a key that lets a principal generate a symmetric data
        encryption key.

        :param key_id: The ARN or ID of the key.
        :param principal: The principal to grant permission to.
        :param operations: The operations to grant permission for.
        :return: The grant that is created.
        """
        try:
            return self.kms_client.create_grant(
```

```
                KeyId=key_id,
                GranteePrincipal=principal,
                Operations=operations,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create a grant on key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat CreateGrantdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_grantee_principal = 'arn:aws:iam::123456789012:role/my-role'
        " it_operations contains 'Encrypt', 'Decrypt', 'GenerateDataKey'
        oo_result = lo_kms->creategrant(
          iv_keyid = iv_key_id
          iv_granteeprincipal = iv_grantee_principal
          it_operations = it_operations
        ).
        MESSAGE 'Grant created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
```

```
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateGrantdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`CreateKey`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateKey`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya
- Bekerja dengan enkripsi tabel

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Shows how to create a new AWS Key Management Service (AWS KMS)
/// key.
/// </summary>
public class CreateKey
```

```
    {
        public static async Task Main()
        {
            // Note that if you need to create a Key in an AWS Region
            // other than the Region defined for the default user, you need to
            // pass the Region to the client constructor.
            var client = new AmazonKeyManagementServiceClient();

            // The call to CreateKeyAsync will create a symmetrical AWS KMS
            // key. For more information about symmetrical and asymmetrical
            // keys, see:
            //
            // https://docs.aws.amazon.com/kms/latest/developerguide/symm-asymm-
choose.html
            var response = await client.CreateKeyAsync(new CreateKeyRequest());

            // The KeyMetadata object contains information about the new AWS KMS
 key.
            KeyMetadata keyMetadata = response.KeyMetadata;

            if (keyMetadata is not null)
            {
                Console.WriteLine($"KMS Key: {keyMetadata.KeyId} was successfully
 created.");
            }
            else
            {
                Console.WriteLine("Could not create KMS Key.");
            }
        }
    }
```

- Untuk detail API, lihat CreateKeydi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Contoh 1: Untuk membuat kunci KMS yang dikelola pelanggan di AWS KMS

`create-key`Contoh berikut membuat kunci KMS enkripsi simetris.

Untuk membuat kunci KMS dasar, kunci enkripsi simetris, Anda tidak perlu menentukan parameter apa pun. Nilai default untuk parameter tersebut membuat kunci enkripsi simetris.

Karena perintah ini tidak menentukan kebijakan kunci, kunci KMS mendapatkan kebijakan kunci default untuk kunci KMS yang dibuat secara terprogram. Untuk melihat kebijakan kunci, gunakan `get-key-policy` perintah. Untuk mengubah kebijakan kunci, gunakan `put-key-policy` perintah.

```
aws kms create-key
```

`create-key`Perintah mengembalikan metadata kunci, termasuk ID kunci dan ARN dari kunci KMS baru. Anda dapat menggunakan nilai-nilai ini untuk mengidentifikasi kunci KMS dalam operasi AWS KMS lainnya. Outputnya tidak termasuk tag. Untuk melihat tag untuk kunci KMS, gunakan tombol. `list-resource-tags command`

Output:

```
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2017-07-05T14:04:55-07:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_KMS"
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Catatan: `create-key` Perintah tidak memungkinkan Anda menentukan alias, Untuk membuat alias untuk kunci KMS baru, gunakan perintah. `create-alias`

Untuk informasi selengkapnya, lihat Membuat AWS kunci di Panduan Pengembang Layanan Manajemen Kunci.

Contoh 2: Untuk membuat kunci KMS RSA asimetris untuk enkripsi dan dekripsi

`create-key`Contoh berikut membuat kunci KMS yang berisi asimetris RSA key pair untuk enkripsi dan dekripsi. Spesifikasi kunci dan penggunaan kunci tidak dapat diubah setelah kunci dibuat. :

```
aws kms create-key \
   --key-spec RSA_4096 \
   --key-usage ENCRYPT_DECRYPT
```

Output:

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2021-04-05T14:04:55-07:00",
        "CustomerMasterKeySpec": "RSA_4096",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "RSAES_OAEP_SHA_1",
            "RSAES_OAEP_SHA_256"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "RSA_4096",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_KMS"
    }
}
```

Untuk informasi selengkapnya, lihat Kunci asimetris di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

Contoh 3: Untuk membuat kunci KMS kurva elips asimetris untuk penandatanganan dan verifikasi

Untuk membuat kunci KMS asimetris yang berisi key pair asymmetric elliptic curve (ECC) untuk penandatanganan dan verifikasi. `--key-usage`Parameter diperlukan meskipun `SIGN_VERIFY` merupakan satu-satunya nilai yang valid untuk kunci ECC KMS. Spesifikasi kunci dan penggunaan kunci tidak dapat diubah setelah kunci dibuat. :

```
aws kms create-key \
    --key-spec ECC_NIST_P521 \
    --key-usage SIGN_VERIFY
```

Output:

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "ECC_NIST_P521",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "MultiRegion": false,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ]
    }
}
```

Untuk informasi selengkapnya, lihat Kunci asimetris di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

Contoh 4: Untuk membuat kunci KMS ML-DSA asimetris untuk penandatanganan dan verifikasi

Contoh ini membuat kunci modul-kisi digital signature algorithm (ML-DSA) untuk penandatanganan dan verifikasi. Parameter penggunaan kunci diperlukan meskipun `SIGN_VERIFY` merupakan satu-satunya nilai yang valid untuk kunci ML-DSA.

```
aws kms create-key \
    --key-spec ML_DSA_65 \
    --key-usage SIGN_VERIFY
```

Output:

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "ML_DSA_65",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "MultiRegion": false,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ML_DSA_SHAKE_256"
        ]
    }
}
```

Untuk informasi selengkapnya, lihat Kunci asimetris di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

Contoh 5: Untuk membuat kunci HMAC KMS

`create-key`Contoh berikut membuat kunci KMS HMAC 384-bit. `GENERATE_VERIFY_MAC`Nilai untuk `--key-usage` parameter diperlukan meskipun itu satu-satunya nilai yang valid untuk kunci HMAC KMS.

```
aws kms create-key \
    --key-spec HMAC_384 \
    --key-usage GENERATE_VERIFY_MAC
```

Output:

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2022-04-05T14:04:55-07:00",
        "CustomerMasterKeySpec": "HMAC_384",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "HMAC_384",
        "KeyState": "Enabled",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "MacAlgorithms": [
            "HMAC_SHA_384"
        ],
        "MultiRegion": false,
        "Origin": "AWS_KMS"
    }
}
```

Untuk informasi selengkapnya, lihat kunci HMAC di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

Contoh 6: Untuk membuat kunci KMS primer Multi-wilayah

`create-key`Contoh berikut membuat kunci enkripsi simetris primer Multi-region. Karena nilai default untuk semua parameter membuat kunci enkripsi simetris, hanya `--multi-region` parameter yang diperlukan untuk kunci KMS ini. Di AWS CLI, untuk menunjukkan bahwa parameter Boolean benar, cukup tentukan nama parameternya.

```
aws kms create-key \
    --multi-region
```

Output:

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2021-09-02T016:15:21-09:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
          "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "mrk-1234abcd12ab34cd56ef12345678990ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
                "Region": "us-west-2"
            },
            "ReplicaKeys": []
        },
        "Origin": "AWS_KMS"
    }
}
```

Untuk informasi selengkapnya, lihat Kunci asimetris di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

Contoh 7: Untuk membuat kunci KMS untuk bahan kunci yang diimpor

`create-key`Contoh berikut membuat membuat kunci KMS tanpa bahan kunci. Ketika operasi selesai, Anda dapat mengimpor materi kunci Anda sendiri ke dalam kunci KMS. Untuk membuat kunci KMS ini, atur `--origin` parameter ke`EXTERNAL`.

```
aws kms create-key \
    --origin EXTERNAL
```

Output:

```
{
     "KeyMetadata": {
         "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "AWSAccountId": "111122223333",
         "CreationDate": "2019-12-02T07:48:55-07:00",
         "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
         "Description": "",
         "Enabled": false,
         "EncryptionAlgorithms": [
             "SYMMETRIC_DEFAULT"
         ],
         "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
         "KeyManager": "CUSTOMER",
         "KeySpec": "SYMMETRIC_DEFAULT",
         "KeyState": "PendingImport",
         "KeyUsage": "ENCRYPT_DECRYPT",
         "MultiRegion": false,
         "Origin": "EXTERNAL"
     }
 }
```

Untuk informasi selengkapnya, lihat Mengimpor materi kunci dalam kunci AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

Contoh 6: Untuk membuat kunci KMS di toko kunci AWS CloudHSM

`create-key`Contoh berikut membuat membuat kunci KMS di toko kunci AWS CloudHSM tertentu. Operasi membuat kunci KMS dan metadatanya di AWS KMS dan membuat materi kunci di klaster CloudHSM yang terkait dengan penyimpanan kunci AWS kustom. Parameter `--custom-key-store-id` dan `--origin` diperlukan.

```
aws kms create-key \
    --origin AWS_CLOUDHSM \
    --custom-key-store-id cks-1234567890abcdef0
```

Output:

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CloudHsmClusterId": "cluster-1a23b4cdefg",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "CustomKeyStoreId": "cks-1234567890abcdef0",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_CLOUDHSM"
    }
}
```

Untuk informasi selengkapnya, lihat AWS Penyimpanan kunci CloudHSM di Panduan AWS Pengembang Layanan Manajemen Kunci.

Contoh 8: Untuk membuat kunci KMS di toko kunci eksternal

`create-key`Contoh berikut membuat membuat kunci KMS di toko kunci eksternal yang ditentukan. `--xks-key-id`Parameter `--custom-key-store-id--origin`,, dan diperlukan dalam perintah ini.

`--xks-key-id`Parameter menentukan ID kunci enkripsi simetris yang ada di pengelola kunci eksternal Anda. Kunci ini berfungsi sebagai bahan kunci eksternal untuk kunci KMS.Nilai `--origin` parameter `EXTERNAL_KEY_STORE` `custom-key-store-id` harus.Parameter harus mengidentifikasi penyimpanan kunci eksternal yang terhubung ke proxy penyimpanan kunci eksternal.

```
aws kms create-key \
```

```
    --origin EXTERNAL_KEY_STORE \
    --custom-key-store-id cks-9876543210fedcba9 \
    --xks-key-id bb8562717f809024
```

Output:

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2022-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "CustomKeyStoreId": "cks-9876543210fedcba9",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "EXTERNAL_KEY_STORE",
        "XksKeyConfiguration": {
            "Id": "bb8562717f809024"
        }
    }
}
```

Untuk informasi selengkapnya, lihat Penyimpanan kunci eksternal di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat CreateKeydi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /**
    * Creates a new symmetric encryption key asynchronously.
    *
    * @param keyDesc the description of the key to be created
    * @return a {@link CompletableFuture} that completes with the ID of the
newly created key
    * @throws RuntimeException if an error occurs while creating the key
    */
   public CompletableFuture<String> createKeyAsync(String keyDesc) {
       CreateKeyRequest keyRequest = CreateKeyRequest.builder()
           .description(keyDesc)
           .keySpec(KeySpec.SYMMETRIC_DEFAULT)
           .keyUsage(KeyUsageType.ENCRYPT_DECRYPT)
           .build();

       return getAsyncClient().createKey(keyRequest)
           .thenApply(resp -> resp.keyMetadata().keyId())
           .exceptionally(ex -> {
               throw new RuntimeException("An error occurred while creating the
key: " + ex.getMessage(), ex);
           });
   }
```

- Untuk detail API, lihat CreateKeydi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun createKey(keyDesc: String?): String? {
    val request =
        CreateKeyRequest {
            description = keyDesc
            customerMasterKeySpec = CustomerMasterKeySpec.SymmetricDefault
            keyUsage = KeyUsageType.fromValue("ENCRYPT_DECRYPT")
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val result = kmsClient.createKey(request)
        println("Created a customer key with id " + result.keyMetadata?.arn)
        return result.keyMetadata?.keyId
    }
}
```

- Untuk detail API, lihat CreateKeydi AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
```

```
     * @param string $keySpec
     * @param string $keyUsage
     * @param string $description
     * @return array
     */
    public function createKey(string $keySpec = "", string $keyUsage = "", string
$description = "Created by the SDK for PHP")
    {
        $parameters = ['Description' => $description];
        if($keySpec && $keyUsage){
            $parameters['KeySpec'] = $keySpec;
            $parameters['KeyUsage'] = $keyUsage;
        }
        try {
            $result = $this->client->createKey($parameters);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            // Check for error specific to createKey operations
            if ($caught->getAwsErrorMessage() == "LimitExceededException"){
                echo "The request was rejected because a quota was exceeded. For
more information, see Quotas in the Key Management Service Developer Guide.";
            }
            throw $caught;
        }
    }
```

- Untuk detail API, lihat CreateKeydi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyManager:
    def __init__(self, kms_client):
```

```
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_key(self, key_description: str) -> dict[str, any]:
        """
        Creates a key with a user-provided description.

        :param key_description: A description for the key.
        :return: The key ID.
        """
        try:
            key = self.kms_client.create_key(Description=key_description)
["KeyMetadata"]
            self.created_keys.append(key)
            return key
        except ClientError as err:
            logging.error(
                "Couldn't create your key. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat CreateKeydi AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Create a AWS KMS key.
# As long we are only encrypting small amounts of data (4 KiB or less) directly,
# a KMS key is fine for our purposes.
# For larger amounts of data,
# use the KMS key to encrypt a data encryption key (DEK).

client = Aws::KMS::Client.new

resp = client.create_key({
                           tags: [
                             {
                               tag_key: 'CreatedBy',
                               tag_value: 'ExampleUser'
                             }
                           ]
                         })

puts resp.key_metadata.key_id
```

- Untuk detail API, lihat CreateKeydi Referensi AWS SDK untuk Ruby API.

Rust

SDK for Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn make_key(client: &Client) -> Result<(), Error> {
    let resp = client.create_key().send().await?;

    let id = resp.key_metadata.as_ref().unwrap().key_id();

    println!("Key: {}", id);

    Ok(())
}
```

- Untuk detail API, lihat CreateKeyreferensi AWS SDK for Rust API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_description = 'Created by the AWS SDK for SAP ABAP'
        oo_result = lo_kms->createkey( iv_description = iv_description ).
        MESSAGE 'KMS key created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
      CATCH /aws1/cx_kmslimitexceededex.
```

```
        MESSAGE 'Limit exceeded for KMS resources.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateKeydi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`Decrypt`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`Decrypt`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

CLI

AWS CLI

Contoh 1: Untuk mendekripsi pesan terenkripsi dengan kunci KMS simetris (Linux dan macOS)

Contoh `decrypt` perintah berikut menunjukkan cara yang disarankan untuk mendekripsi data dengan CLI. AWS Versi ini menunjukkan cara mendekripsi data di bawah kunci KMS simetris.

Berikan ciphertext dalam file.Dalam nilai `--ciphertext-blob` parameter, gunakan `fileb://` awalan, yang memberitahu CLI untuk membaca data dari file biner. Jika file tidak ada di direktori saat ini, ketik path lengkap ke file. Untuk informasi lebih lanjut tentang membaca nilai parameter AWS CLI dari file, lihat Memuat parameter AWS CLI dari file < https://docs.aws.amazon.com/cli/ latest/userguide/cli - usage-parameters-file .html> di Panduan Pengguna Antarmuka Baris AWS Perintah dan Praktik Terbaik untuk Parameter File Lokal< https://aws.amazon.com/blogs/ pengembang/ best-practices-for-local -file-parameters/ > di Blog Alat Baris Perintah. Tentukan kunci KMS untuk mendekripsi cipherText.Parameter tidak diperlukan saat AWS mendekripsi dengan kunci KMS simetris. `--key-id` AWS KMS bisa mendapatkan ID kunci dari kunci KMS yang digunakan untuk mengenkripsi data dari

metadata dalam ciphertext. Tetapi selalu merupakan praktik terbaik untuk menentukan kunci KMS yang Anda gunakan. Praktek ini memastikan bahwa Anda menggunakan kunci KMS yang Anda inginkan, dan mencegah Anda dari secara tidak sengaja mendekripsi ciphertext menggunakan kunci KMS yang tidak Anda percayai. Minta output plaintext sebagai nilai teks.Parameter `--query` memberitahu CLI untuk mendapatkan hanya nilai bidang dari output. `Plaintext` `--output`Parameter mengembalikan output sebagai text.base64-decode plaintext dan menyimpannya dalam file.Contoh berikut pipa (|) nilai `Plaintext` parameter ke utilitas Base64, yang menerjemahkan itu. Kemudian, itu mengalihkan (>) output yang diterjemahkan ke file. `ExamplePlaintext`

Sebelum menjalankan perintah ini, ganti ID kunci contoh dengan ID kunci yang valid dari AWS akun Anda.

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text \
    --query Plaintext | base64 \
    --decode > ExamplePlaintextFile
```

Perintah ini tidak menghasilkan output. Output dari `decrypt` perintah adalah base64-decoded dan disimpan dalam file.

Untuk informasi selengkapnya, lihat Mendekripsi di Referensi API Layanan Manajemen AWS Kunci.

Contoh 2: Untuk mendekripsi pesan terenkripsi dengan tombol KMS simetris (prompt perintah Windows)

Contoh berikut adalah sama dengan yang sebelumnya kecuali bahwa ia menggunakan `certutil` utilitas untuk Base64-decode data plaintext. Prosedur ini membutuhkan dua perintah, seperti yang ditunjukkan pada contoh berikut.

Sebelum menjalankan perintah ini, ganti ID kunci contoh dengan ID kunci yang valid dari AWS akun Anda.

```
aws kms decrypt ^
    --ciphertext-blob fileb://ExampleEncryptedFile ^
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
    --output text ^
```

```
    --query Plaintext > ExamplePlaintextFile.base64
```

Jalankan perintah `certutil`.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Output:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Untuk informasi selengkapnya, lihat Mendekripsi di Referensi API Layanan Manajemen AWS Kunci.

Contoh 3: Untuk mendekripsi pesan terenkripsi dengan kunci KMS asimetris (Linux dan macOS)

Contoh `decrypt` perintah berikut menunjukkan cara mendekripsi data yang dienkripsi di bawah kunci KMS asimetris RSA.

Saat menggunakan kunci KMS asimetris, `encryption-algorithm` parameter, yang menentukan algoritma yang digunakan untuk mengenkripsi plaintext, diperlukan.

Sebelum menjalankan perintah ini, ganti ID kunci contoh dengan ID kunci yang valid dari AWS akun Anda.

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --encryption-algorithm RSAES_OAEP_SHA_256 \
    --output text \
    --query Plaintext | base64 \
    --decode > ExamplePlaintextFile
```

Perintah ini tidak menghasilkan output. Output dari `decrypt` perintah adalah base64-decoded dan disimpan dalam file.

Untuk informasi selengkapnya, lihat Kunci asimetris di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat Mendekripsi di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /**
    * Asynchronously decrypts the given encrypted data using the specified key
ID.
    *
    * @param encryptedData The encrypted data to be decrypted.
    * @param keyId The ID of the key to be used for decryption.
    * @return A CompletableFuture that, when completed, will contain the
decrypted data as a String.
    *         If an error occurs during the decryption process, the
CompletableFuture will complete
    *         exceptionally with the error, and the method will return an empty
String.
    */
   public CompletableFuture<String> decryptDataAsync(SdkBytes encryptedData,
String keyId) {
       DecryptRequest decryptRequest = DecryptRequest.builder()
           .ciphertextBlob(encryptedData)
           .keyId(keyId)
           .build();

       CompletableFuture<DecryptResponse> responseFuture =
getAsyncClient().decrypt(decryptRequest);
       responseFuture.whenComplete((decryptResponse, exception) -> {
           if (exception == null) {
               logger.info("Data decrypted successfully for key ID: " + keyId);
           } else {
               if (exception instanceof KmsException kmsEx) {
                   throw new RuntimeException("KMS error occurred while
decrypting data: " + kmsEx.getMessage(), kmsEx);
               } else {
```

```
                    throw new RuntimeException("An unexpected error occurred
 while decrypting data: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(decryptResponse ->
 decryptResponse.plaintext().asString(StandardCharsets.UTF_8));
    }
```

- Untuk detail API, lihat Mendekripsi di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun encryptData(keyIdValue: String): ByteArray? {
    val text = "This is the text to encrypt by using the AWS KMS Service"
    val myBytes: ByteArray = text.toByteArray()

    val encryptRequest =
        EncryptRequest {
            keyId = keyIdValue
            plaintext = myBytes
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.encrypt(encryptRequest)
        val algorithm: String = response.encryptionAlgorithm.toString()
        println("The encryption algorithm is $algorithm")

        // Return the encrypted data.
        return response.ciphertextBlob
    }
}
```

```
suspend fun decryptData(
    encryptedDataVal: ByteArray?,
    keyIdVal: String?,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }
    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
        val myVal = decryptResponse.plaintext

        // Print the decrypted data.
        print(myVal)
    }
}
```

- Untuk detail API, lihat Mendekripsi di AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @param string $ciphertext
     * @param string $algorithm
     * @return Result
     */
    public function decrypt(string $keyId, string $ciphertext, string $algorithm
 = "SYMMETRIC_DEFAULT")
    {
```

```
        try{
            return $this->client->decrypt([
                'CiphertextBlob' => $ciphertext,
                'EncryptionAlgorithm' => $algorithm,
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem decrypting the data: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- Untuk detail API, lihat Mendekripsi di Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)
```

```
    def decrypt(self, key_id: str, cipher_text: bytes) -> str:
        """
        Decrypts text previously encrypted with a key.

        :param key_id: The ARN or ID of the key used to decrypt the data.
        :param cipher_text: The encrypted text to decrypt.
        :return: The decrypted text.
        """
        try:
            return self.kms_client.decrypt(KeyId=key_id,
CiphertextBlob=cipher_text)[
                "Plaintext"
            ].decode()
        except ClientError as err:
            logger.error(
                "Couldn't decrypt your ciphertext. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat Mendekripsi dalam AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Decrypted blob

blob =
 '01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593
```

```
blob_packed = [blob].pack('H*')

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.decrypt({
                        ciphertext_blob: blob_packed
                      })

puts 'Raw text: '
puts resp.plaintext
```

- Untuk detail API, lihat Mendekripsi di Referensi AWS SDK untuk Ruby API.

Rust

SDK for Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn decrypt_key(client: &Client, key: &str, filename: &str) -> Result<(),
 Error> {
    // Open input text file and get contents as a string
    // input is a base-64 encoded string, so decode it:
    let data = fs::read_to_string(filename)
        .map(|input| {
            base64::decode(input).expect("Input file does not contain valid base
 64 characters.")
        })
        .map(Blob::new);

    let resp = client
        .decrypt()
        .key_id(key)
        .ciphertext_blob(data.unwrap())
        .send()
        .await?;
```

```
    let inner = resp.plaintext.unwrap();
    let bytes = inner.as_ref();

    let s = String::from_utf8(bytes.to_vec()).expect("Could not convert to
 UTF-8");

    println!();
    println!("Decoded string:");
    println!("{}", s);

    Ok(())
}
```

- Untuk detail API, lihat Mendekripsi di AWS SDK untuk referensi API Rust.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_ciphertext_blob contains the encrypted data
        oo_result = lo_kms->decrypt(
          iv_keyid = iv_key_id
          iv_ciphertextblob = iv_ciphertext_blob
        ).
        MESSAGE 'Text decrypted successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsincorrectkeyex.
        MESSAGE 'Incorrect key for decryption.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
```

```
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Mendekripsi di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DeleteAlias`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteAlias`.

CLI

AWS CLI

Untuk menghapus alias AWS KMS

`delete-alias`Contoh berikut menghapus `alias/example-alias` alias. Nama alias harus dimulai dengan alias/.

```
aws kms delete-alias \
    --alias-name alias/example-alias
```

Perintah ini tidak menghasilkan output. Untuk menemukan alias, gunakan `list-aliases` perintah.

Untuk informasi selengkapnya, lihat Menghapus alias di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat DeleteAliasdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /**
     * Deletes a specific KMS alias asynchronously.
     *
     * @param aliasName the name of the alias to be deleted
     * @return a {@link CompletableFuture} representing the asynchronous
operation of deleting the specified alias
     */
    public CompletableFuture<Void> deleteSpecificAliasAsync(String aliasName) {
        DeleteAliasRequest deleteAliasRequest = DeleteAliasRequest.builder()
            .aliasName(aliasName)
            .build();

        return getAsyncClient().deleteAlias(deleteAliasRequest)
            .thenRun(() -> {
                logger.info("Alias {} has been deleted successfully", aliasName);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to delete alias: " +
aliasName, throwable);
            });
    }
```

- Untuk detail API, lihat DeleteAliasdi Referensi AWS SDK for Java 2.x API.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $aliasName
     * @return void
     */
    public function deleteAlias(string $aliasName)
    {
        try {
            $this->client->deleteAlias([
                'AliasName' => $aliasName,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem deleting the alias: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- Untuk detail API, lihat DeleteAliasdi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def delete_alias(self, alias: str) -> None:
        """
        Deletes an alias.

        :param alias: The alias to delete.
        """
        try:
            self.kms_client.delete_alias(AliasName=alias)
        except ClientError as err:
            logger.error(
                "Couldn't delete alias %s. Here's why: %s",
                alias,
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat DeleteAliasdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_alias_name = 'alias/my-key-alias'
    lo_kms->deletealias( iv_aliasname = iv_alias_name ).
    MESSAGE 'Alias deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Alias not found.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteAliasdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DescribeKey`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DescribeKey`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Retrieve information about an AWS Key Management Service (AWS KMS) key.
/// You can supply either the key Id or the key Amazon Resource Name (ARN)
/// to the DescribeKeyRequest KeyId property.
/// </summary>
public class DescribeKey
{
    public static async Task Main()
    {
        var keyId = "7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
        var request = new DescribeKeyRequest
        {
            KeyId = keyId,
        };

        var client = new AmazonKeyManagementServiceClient();

        var response = await client.DescribeKeyAsync(request);
        var metadata = response.KeyMetadata;

        Console.WriteLine($"{metadata.KeyId} created on:
{metadata.CreationDate}");
        Console.WriteLine($"State: {metadata.KeyState}");
        Console.WriteLine($"{metadata.Description}");
    }
}
```

- Untuk detail API, lihat DescribeKeydi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Contoh 1: Untuk menemukan informasi rinci tentang kunci KMS

`describe-key`Contoh berikut mendapatkan informasi rinci tentang kunci AWS terkelola untuk Amazon S3 di akun contoh dan Wilayah. Anda dapat menggunakan perintah ini untuk menemukan detail tentang kunci AWS terkelola dan kunci yang dikelola pelanggan.

Untuk menentukan kunci KMS, gunakan `key-id` parameter. Contoh ini menggunakan nilai nama alias, tetapi Anda dapat menggunakan ID kunci, ARN kunci, nama alias, atau alias ARN dalam perintah ini.

```
aws kms describe-key \
    --key-id alias/aws/s3
```

Output:

```
{
    "KeyMetadata": {
        "AWSAccountId": "846764612917",
        "KeyId": "b8a9477d-836c-491f-857e-07937918959b",
        "Arn": "arn:aws:kms:us-west-2:846764612917:key/
b8a9477d-836c-491f-857e-07937918959b",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CreationDate": 2017-06-30T21:44:32.140000+00:00,
        "Enabled": true,
        "Description": "Default KMS key that protects my S3 objects when no other
 key is defined",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "AWS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
```

```
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Untuk informasi selengkapnya, lihat Melihat AWS kunci di Panduan Pengembang Layanan Manajemen Kunci.

Contoh 2: Untuk mendapatkan detail tentang kunci KMS asimetris RSA

`describe-key`Contoh berikut mendapatkan informasi rinci tentang kunci KMS RSA asimetris yang digunakan untuk penandatanganan dan verifikasi.

```
aws kms describe-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2019-12-02T19:47:14.861000+00:00",
        "CustomerMasterKeySpec": "RSA_2048",
        "Enabled": false,
        "Description": "",
        "KeyState": "Disabled",
        "Origin": "AWS_KMS",
        "MultiRegion": false,
        "KeyManager": "CUSTOMER",
        "KeySpec": "RSA_2048",
        "KeyUsage": "SIGN_VERIFY",
        "SigningAlgorithms": [
            "RSASSA_PKCS1_V1_5_SHA_256",
            "RSASSA_PKCS1_V1_5_SHA_384",
            "RSASSA_PKCS1_V1_5_SHA_512",
            "RSASSA_PSS_SHA_256",
            "RSASSA_PSS_SHA_384",
            "RSASSA_PSS_SHA_512"
        ]
```

```
    }
}
```

Contoh 3: Untuk mendapatkan detail tentang kunci replika Multi-wilayah

`describe-key`Contoh berikut mendapatkan metadata untuk kunci replika Multi-wilayah. Kunci Multi-region ini adalah kunci enkripsi simetris. Output dari `describe-key` perintah untuk setiap kunci Multi-region mengembalikan informasi tentang kunci utama dan semua replika nya.

```
aws kms describe-key \
    --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Output:

```
{
    "KeyMetadata": {
        "MultiRegion": true,
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": "2021-06-28T21:09:16.114000+00:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "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"
                }
            ]
        }
    }
}
```

Contoh 4: Untuk mendapatkan detail tentang kunci HMAC KMS

`describe-key`Contoh berikut mendapatkan informasi rinci tentang kunci HMAC KMS.

```
aws kms describe-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{
    "KeyMetadata": {
        "AWSAccountId": "123456789012",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Arn": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2022-04-03T22:23:10.194000+00:00",
        "Enabled": true,
        "Description": "Test key",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
```

```
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "HMAC_256",
        "MacAlgorithms": [
            "HMAC_SHA_256"
        ],
        "MultiRegion": false
    }
}
```

- Untuk detail API, lihat DescribeKeydi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /**
     * Asynchronously checks if a specified key is enabled.
     *
     * @param keyId the ID of the key to check
     * @return a {@link CompletableFuture} that, when completed, indicates
whether the key is enabled or not
     *
     * @throws RuntimeException if an exception occurs while checking the key
state
     */
    public CompletableFuture<Boolean> isKeyEnabledAsync(String keyId) {
        DescribeKeyRequest keyRequest = DescribeKeyRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<DescribeKeyResponse> responseFuture =
getAsyncClient().describeKey(keyRequest);
        return responseFuture.whenComplete((resp, ex) -> {
            if (resp != null) {
                KeyState keyState = resp.keyMetadata().keyState();
                if (keyState == KeyState.ENABLED) {
```

```
                    logger.info("The key is enabled.");
                } else {
                    logger.info("The key is not enabled. Key state: {}",
 keyState);
                }
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(resp -> resp.keyMetadata().keyState() == KeyState.ENABLED);
    }
```

- Untuk detail API, lihat DescribeKeydi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun describeSpecifcKey(keyIdVal: String?) {
    val request =
        DescribeKeyRequest {
            keyId = keyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.describeKey(request)
        println("The key description is ${response.keyMetadata?.description}")
        println("The key ARN is ${response.keyMetadata?.arn}")
    }
}
```

- Untuk detail API, lihat DescribeKeydi AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @return array
     */
    public function describeKey(string $keyId)
    {
        try {
            $result = $this->client->describeKey([
                "KeyId" => $keyId,
            ]);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
resource could not be found.\n";
            }
            throw $caught;
        }
    }
```

- Untuk detail API, lihat DescribeKeydi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def describe_key(self, key_id: str) -> dict[str, any]:
        """
        Describes a key.

        :param key_id: The ARN or ID of the key to describe.
        :return: Information about the key.
        """

        try:
            key = self.kms_client.describe_key(KeyId=key_id)["KeyMetadata"]
            return key
        except ClientError as err:
            logging.error(
                "Couldn't get key '%s'. Here's why: %s",
                key_id,
```

```
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat DescribeKeydi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->describekey( iv_keyid = iv_key_id ).
        DATA(lo_key) = oo_result->get_keymetadata( ).
        MESSAGE 'Retrieved key information successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DescribeKeydi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DisableKey`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DisableKey`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Disable an AWS Key Management Service (AWS KMS) key and then retrieve
/// the key's status to show that it has been disabled.
/// </summary>
public class DisableKey
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The identifier of the AWS KMS key to disable. You can use the
        // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

        var request = new DisableKeyRequest
        {
            KeyId = keyId,
        };

        var response = await client.DisableKeyAsync(request);
```

```
            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                // Retrieve information about the key to show that it has now
                // been disabled.
                var describeResponse = await client.DescribeKeyAsync(new
DescribeKeyRequest
                {
                    KeyId = keyId,
                });
                Console.WriteLine($"{describeResponse.KeyMetadata.KeyId} - state:
{describeResponse.KeyMetadata.KeyState}");
            }
        }
    }
```

- Untuk detail API, lihat DisableKeydi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk menonaktifkan sementara kunci KMS

`disable-key`Perintah berikut menonaktifkan kunci KMS yang dikelola pelanggan. Untuk mengaktifkan kembali kunci KMS, gunakan perintah. `enable-key`

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

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat Mengaktifkan dan Menonaktifkan Kunci di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat DisableKeydi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /**
    * Asynchronously disables the specified AWS Key Management Service (KMS)
key.
    *
    * @param keyId the ID or Amazon Resource Name (ARN) of the KMS key to be
disabled
    * @return a CompletableFuture that, when completed, indicates that the key
has been disabled successfully
    */
   public CompletableFuture<Void> disableKeyAsync(String keyId) {
       DisableKeyRequest keyRequest = DisableKeyRequest.builder()
           .keyId(keyId)
           .build();

       return getAsyncClient().disableKey(keyRequest)
           .thenRun(() -> {
               logger.info("Key {} has been disabled successfully",keyId);
           })
           .exceptionally(throwable -> {
               throw new RuntimeException("Failed to disable key: " + keyId,
throwable);
           });
   }
```

- Untuk detail API, lihat DisableKeydi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun disableKey(keyIdVal: String?) {
    val request =
        DisableKeyRequest {
            keyId = keyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        kmsClient.disableKey(request)
        println("$keyIdVal was successfully disabled")
    }
}
```

- Untuk detail API, lihat DisableKeydi AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @return void
     */
```

```
    public function disableKey(string $keyId)
    {
        try {
            $this->client->disableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem disabling the key: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- Untuk detail API, lihat DisableKeydi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)
```

```
    def disable_key(self, key_id: str) -> None:
        try:
            self.kms_client.disable_key(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't disable key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat DisableKeydi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->disablekey( iv_keyid = iv_key_id ).
        MESSAGE 'KMS key disabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DisableKeydi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`EnableKey`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`EnableKey`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Enable an AWS Key Management Service (AWS KMS) key.
/// </summary>
public class EnableKey
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The identifier of the AWS KMS key to enable. You can use the
        // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";
```

```
            var request = new EnableKeyRequest
            {
                KeyId = keyId,
            };

            var response = await client.EnableKeyAsync(request);
            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                // Retrieve information about the key to show that it has now
                // been enabled.
                var describeResponse = await client.DescribeKeyAsync(new
DescribeKeyRequest
                {
                    KeyId = keyId,
                });
                Console.WriteLine($"{describeResponse.KeyMetadata.KeyId} - state:
{describeResponse.KeyMetadata.KeyState}");
            }
        }
    }
```

- Untuk detail API, lihat EnableKeydi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk mengaktifkan kunci KMS

`enable-key`Contoh berikut memungkinkan kunci yang dikelola pelanggan. Anda dapat menggunakan perintah seperti ini untuk mengaktifkan kunci KMS yang Anda nonaktifkan sementara dengan menggunakan `disable-key` perintah. Anda juga dapat menggunakannya untuk mengaktifkan kunci KMS yang dinonaktifkan karena dijadwalkan untuk dihapus dan penghapusan dibatalkan.

Untuk menentukan kunci KMS, gunakan `key-id` parameter. Contoh ini menggunakan nilai ID kunci, tetapi Anda dapat menggunakan ID kunci atau nilai ARN kunci dalam perintah ini.

Sebelum menjalankan perintah ini, ganti ID kunci contoh dengan yang valid.

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

Perintah ini tidak menghasilkan output. Untuk memverifikasi bahwa kunci KMS diaktifkan, gunakan `describe-key` perintah. Lihat nilai `KeyState` dan `Enabled` bidang dalam `describe-key` output.

Untuk informasi selengkapnya, lihat Mengaktifkan dan Menonaktifkan Kunci di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat EnableKeydi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /**
     * Asynchronously enables the specified key.
     *
     * @param keyId the ID of the key to enable
     * @return a {@link CompletableFuture} that completes when the key has been
enabled
     */
    public CompletableFuture<Void> enableKeyAsync(String keyId) {
        EnableKeyRequest enableKeyRequest = EnableKeyRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<EnableKeyResponse> responseFuture =
getAsyncClient().enableKey(enableKeyRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Key with ID [{}] has been enabled.", keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
```

```
                    throw new RuntimeException("KMS error occurred while enabling
 key: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
 while enabling key: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(response -> null);
    }
```

- Untuk detail API, lihat EnableKeydi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun enableKey(keyIdVal: String?) {
    val request =
        EnableKeyRequest {
            keyId = keyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        kmsClient.enableKey(request)
        println("$keyIdVal was successfully enabled.")
    }
}
```

- Untuk detail API, lihat EnableKeydi AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @return void
     */
    public function enableKey(string $keyId)
    {
        try {
            $this->client->enableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
resource could not be found.\n";
            }
            throw $caught;
        }
    }
```

- Untuk detail API, lihat EnableKeydi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def enable_key(self, key_id: str) -> None:
        """
        Enables a key. Gets the key state after each state change.

        :param key_id: The ARN or ID of the key to enable.
        """
        try:
            self.kms_client.enable_key(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't enable key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat EnableKeydi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->enablekey( iv_keyid = iv_key_id ).
        MESSAGE 'KMS key enabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat EnableKeydi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`EnableKeyRotation`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`EnableKeyRotation`.

CLI

AWS CLI

Untuk mengaktifkan rotasi otomatis tombol KMS

`enable-key-rotation`Contoh berikut memungkinkan rotasi otomatis kunci KMS yang dikelola pelanggan dengan periode rotasi 180 hari. Kunci KMS akan diputar satu tahun (perkiraan 365 hari) dari tanggal perintah ini selesai dan setiap tahun setelahnya.

`--key-id`Parameter mengidentifikasi kunci KMS. Contoh ini menggunakan nilai ARN kunci, tetapi Anda dapat menggunakan ID kunci atau ARN dari kunci `--rotation-period-in-days` KMS.Parameter menentukan jumlah hari antara setiap tanggal rotasi. Tentukan nilai antara 90 dan 2560 hari. Jika tidak ada nilai yang ditentukan, nilai defaultnya adalah 365 hari.

```
aws kms enable-key-rotation \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --rotation-period-in-days 180
```

Perintah ini tidak menghasilkan output. Untuk memverifikasi bahwa kunci KMS diaktifkan, gunakan `get-key-rotation-status` perintah.

Untuk informasi selengkapnya, lihat Memutar kunci di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat EnableKeyRotationdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
```

```
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def enable_key_rotation(self, key_id: str) -> None:
        """
        Enables rotation for a key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        """
        try:
            self.kms_client.enable_key_rotation(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't enable rotation for key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat EnableKeyRotationdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->enablekeyrotation( iv_keyid = iv_key_id ).
        MESSAGE 'Key rotation enabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsunsupportedopex.
        MESSAGE 'Operation not supported for this key.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat EnableKeyRotationdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`Encrypt`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`Encrypt`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

CLI

AWS CLI

Contoh 1: Untuk mengenkripsi isi file di Linux atau macOS

`encrypt`Perintah berikut menunjukkan cara yang disarankan untuk mengenkripsi data dengan CLI AWS .

```
aws kms encrypt \
```

```
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

Perintah tersebut melakukan beberapa hal:

Menggunakan `--plaintext` parameter untuk menunjukkan data untuk mengenkripsi. Nilai parameter ini harus dikodekan Base64-. Nilai `plaintext` parameter harus dikodekan base64, atau Anda harus menggunakan awalan`fileb://`, yang memberitahu AWS CLI untuk membaca data biner dari file.Jika file tidak ada di direktori saat ini, ketik path lengkap ke file. Misalnya: `fileb:///var/tmp/ExamplePlaintextFile` atau `fileb://C:\Temp\ExamplePlaintextFile`. Untuk informasi lebih lanjut tentang membaca nilai parameter AWS CLI dari file, lihat Memuat Parameter dari File di Panduan Pengguna Antarmuka Baris AWS Perintah dan Praktik Terbaik untuk Parameter File Lokal di Blog Alat Baris AWS Perintah.Menggunakan `--output` dan `--query` parameter untuk mengontrol output perintah.Parameter ini mengekstrak data terenkripsi, yang disebut ciphertext, dari output perintah.Untuk informasi lebih lanjut tentang mengontrol output, lihat Mengontrol Perintah Output dalam Panduan Pengguna Antarmuka Baris AWS Perintah.Menggunakan `base64` utilitas untuk memecahkan kode output yang diekstraksi ke dalam data biner.Ciphertext yang dikembalikan oleh perintah yang berhasil adalah teks yang dikodekan base64. `encrypt` Anda harus memecahkan kode teks ini sebelum Anda dapat menggunakan AWS CLI untuk mendekripsi itu.Menyimpan ciphertext biner ke file.Bagian terakhir dari perintah > `ExampleEncryptedFile` () menyimpan ciphertext biner ke file untuk membuat dekripsi lebih mudah. Untuk contoh perintah yang menggunakan AWS CLI untuk mendekripsi data, lihat contoh dekripsi.

Contoh 2: Menggunakan AWS CLI untuk mengenkripsi data pada Windows

Contoh ini sama dengan yang sebelumnya, kecuali bahwa ia menggunakan `certutil` alat sebagai gantinya`base64`. Prosedur ini membutuhkan dua perintah, seperti yang ditunjukkan pada contoh berikut.

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob > C:\Temp\ExampleEncryptedFile.base64
```

```
certutil -decode C:\Temp\ExampleEncryptedFile.base64 C:\Temp\ExampleEncryptedFile
```

Contoh 3: Mengenkripsi dengan tombol KMS asimetris

`encrypt`Perintah berikut menunjukkan cara mengenkripsi plaintext dengan kunci KMS asimetris. parameter `--encryption-algorithm` diperlukan. Seperti pada semua perintah `encrypt` CLI, `plaintext` parameter harus dikodekan base64, atau Anda harus menggunakan `fileb://` awalan, yang memberi tahu AWS CLI untuk membaca data biner dari file.

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encryption-algorithm RSAES_OAEP_SHA_256 \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

Perintah ini tidak menghasilkan output.

- Untuk detail API, lihat Enkripsi di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /**
     * Encrypts the given text asynchronously using the specified KMS client and
key ID.
     *
     * @param keyId the ID of the KMS key to use for encryption
     * @param text the text to encrypt
     * @return a CompletableFuture that completes with the encrypted data as an
SdkBytes object
```

```
     */
    public CompletableFuture<SdkBytes> encryptDataAsync(String keyId, String
text) {
        SdkBytes myBytes = SdkBytes.fromUtf8String(text);
        EncryptRequest encryptRequest = EncryptRequest.builder()
            .keyId(keyId)
            .plaintext(myBytes)
            .build();

        CompletableFuture<EncryptResponse> responseFuture =
getAsyncClient().encrypt(encryptRequest).toCompletableFuture();
        return responseFuture.whenComplete((response, ex) -> {
            if (response != null) {
                String algorithm = response.encryptionAlgorithm().toString();
                logger.info("The string was encrypted with algorithm {}.",
algorithm);
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(EncryptResponse::ciphertextBlob);
    }
```

- Untuk detail API, lihat Enkripsi di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun encryptData(keyIdValue: String): ByteArray? {
    val text = "This is the text to encrypt by using the AWS KMS Service"
    val myBytes: ByteArray = text.toByteArray()

    val encryptRequest =
        EncryptRequest {
            keyId = keyIdValue
```

```
            plaintext = myBytes
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.encrypt(encryptRequest)
        val algorithm: String = response.encryptionAlgorithm.toString()
        println("The encryption algorithm is $algorithm")

        // Return the encrypted data.
        return response.ciphertextBlob
    }
}

suspend fun decryptData(
    encryptedDataVal: ByteArray?,
    keyIdVal: String?,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }
    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
        val myVal = decryptResponse.plaintext

        // Print the decrypted data.
        print(myVal)
    }
}
```

- Untuk detail API, lihat Enkripsi di AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @param string $text
     * @return Result
     */
    public function encrypt(string $keyId, string $text)
    {
        try {
            return $this->client->encrypt([
                'KeyId' => $keyId,
                'Plaintext' => $text,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "DisabledException"){
                echo "The request was rejected because the specified KMS key is
 not enabled.\n";
            }
            throw $caught;
        }
    }
```

- Untuk detail API, lihat Enkripsi di Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
```

```
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def encrypt(self, key_id: str, text: str) -> bytes:
        """
        Encrypts text by using the specified key.

        :param key_id: The ARN or ID of the key to use for encryption.
        :param text: The text to encrypt.
        :return: The encrypted version of the text.
        """
        try:
            response = self.kms_client.encrypt(KeyId=key_id,
Plaintext=text.encode())
            print(
                f"The string was encrypted with algorithm
{response['EncryptionAlgorithm']}"
            )
            return response["CiphertextBlob"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DisabledException":
                logger.error(
                    "Could not encrypt because the key %s is disabled.", key_id
                )
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            raise
```

- Untuk detail API, lihat Mengenkripsi di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# ARN of the AWS KMS key.
#
# Replace the fictitious key ARN with a valid key ID

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

text = '1234567890'

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.encrypt({
                        key_id: keyId,
                        plaintext: text
                      })

# Display a readable version of the resulting encrypted blob.
puts 'Blob:'
puts resp.ciphertext_blob.unpack('H*')
```

- Untuk detail API, lihat Enkripsi di Referensi AWS SDK untuk Ruby API.

Rust

SDK for Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn encrypt_string(
    verbose: bool,
    client: &Client,
    text: &str,
    key: &str,
    out_file: &str,
) -> Result<(), Error> {
    let blob = Blob::new(text.as_bytes());

    let resp = client.encrypt().key_id(key).plaintext(blob).send().await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    let mut ofile = File::create(out_file).expect("unable to create file");
    ofile.write_all(s.as_bytes()).expect("unable to write");

    if verbose {
        println!("Wrote the following to {:?}", out_file);
        println!("{}", s);
    }

    Ok(())
}
```

- Untuk detail API, lihat Enkripsi di AWS SDK untuk referensi API Rust.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_plaintext contains the data to encrypt
        oo_result = lo_kms->encrypt(
          iv_keyid = iv_key_id
          iv_plaintext = iv_plaintext
        ).
        MESSAGE 'Text encrypted successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Encrypt in AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`GenerateDataKey`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GenerateDataKey`.

CLI

AWS CLI

Contoh 1: Untuk menghasilkan kunci data simetris 256-bit

`generate-data-key`Contoh berikut meminta kunci data simetris 256-bit untuk digunakan di luar. AWS Perintah mengembalikan kunci data plaintext untuk segera digunakan dan dihapus, dan salinan kunci data yang dienkripsi di bawah kunci KMS yang ditentukan. Anda dapat dengan aman menyimpan kunci data terenkripsi dengan data terenkripsi.

Untuk meminta kunci data 256-bit, gunakan `key-spec` parameter dengan nilai. `AES_256` Untuk meminta kunci data 128-bit, gunakan `key-spec` parameter dengan nilai. `AES_128` Untuk semua panjang kunci data lainnya, gunakan `number-of-bytes` parameter.

Kunci KMS yang Anda tentukan harus berupa kunci KMS enkripsi simetris, yaitu kunci KMS dengan nilai spesifikasi kunci SYMMETRIC_DEFAULT.

```
aws kms generate-data-key \
    --key-id alias/ExampleAlias \
    --key-spec AES_256
```

Output:

```
{
    "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
+YdhV8MrkBQPeac0ReRVNDt9qleAt+SHgIRF8P0H+7U="
}
```

`Plaintext`(Kunci data teks biasa) dan `CiphertextBlob` (kunci data terenkripsi) dikembalikan dalam format yang disandikan base64.

Untuk informasi selengkapnya, lihat Kunci data di Panduan Pengembang Layanan Manajemen AWS Kunci. Contoh 2: Untuk menghasilkan kunci data simetris 512-bit

`generate-data-key`Contoh berikut meminta kunci data simetris 512-bit untuk enkripsi dan dekripsi. Perintah mengembalikan kunci data plaintext untuk segera digunakan dan dihapus, dan salinan kunci data yang dienkripsi di bawah kunci KMS yang ditentukan. Anda dapat dengan aman menyimpan kunci data terenkripsi dengan data terenkripsi.

Untuk meminta panjang kunci selain 128 atau 256 bit, gunakan `number-of-bytes` parameter. Untuk meminta kunci data 512-bit, contoh berikut menggunakan `number-of-bytes` parameter dengan nilai 64 (byte).

Kunci KMS yang Anda tentukan harus berupa kunci KMS enkripsi simetris, yaitu kunci KMS dengan nilai spesifikasi kunci SYMMETRIC_DEFAULT.

CATATAN: Nilai-nilai dalam output dari contoh ini terpotong untuk tampilan.

```
aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --number-of-bytes 64
```

Output:

```
{
    "CiphertextBlob": "AQIBAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y+hAEnX/
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSIb3DQEHBqCCAlUwggJRAgEAMIICSgYJKoZ...",
    "Plaintext": "ty8Lr0Bk6OF07M2BWt6qbFdNB
+G00ZLtf5MSEb4al3R2UKWGOp06njAwy2n72VRm2m7z/
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm4OovEXiVfBveyX3DQxDzRSwbKDPk/...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

`Plaintext`(Kunci data teks biasa) dan `CiphertextBlob` (kunci data terenkripsi) dikembalikan dalam format yang disandikan base64.

Untuk informasi selengkapnya, lihat Kunci data di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat GenerateDataKeydi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def generate_data_key(self, key_id):
        """
        Generates a symmetric data key that can be used for client-side
 encryption.
        """
        answer = input(
            f"Do you want to generate a symmetric data key from key {key_id} (y/
n)? "
        )
        if answer.lower() == "y":
            try:
                data_key = self.kms_client.generate_data_key(
                    KeyId=key_id, KeySpec="AES_256"
                )
            except ClientError as err:
```

```
                logger.error(
                    "Couldn't generate a data key for key %s. Here's why: %s",
                    key_id,
                    err.response["Error"]["Message"],
                )
            else:
                pprint(data_key)
```

- Untuk detail API, lihat GenerateDataKeydi AWS SDK for Python (Boto3) Referensi API.

Rust

SDK for Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn make_key(client: &Client, key: &str) -> Result<(), Error> {
    let resp = client
        .generate_data_key()
        .key_id(key)
        .key_spec(DataKeySpec::Aes256)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
}
```

- Untuk detail API, lihat GenerateDataKeyreferensi AWS SDK for Rust API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_keyspec = 'AES_256'
        oo_result = lo_kms->generatedatakey(
          iv_keyid = iv_key_id
          iv_keyspec = 'AES_256'
        ).
        MESSAGE 'Data key generated successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GenerateDataKeydi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`GenerateDataKeyWithoutPlaintext`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GenerateDataKeyWithoutPlaintext`.

CLI

AWS CLI

Untuk menghasilkan kunci data simetris 256-bit tanpa kunci plaintext

`generate-data-key-without-plaintext`Contoh berikut meminta salinan terenkripsi dari kunci data simetris 256-bit untuk digunakan di luar. AWS Anda dapat memanggil AWS KMS untuk mendekripsi kunci data saat Anda siap menggunakannya.

Untuk meminta kunci data 256-bit, gunakan `key-spec` parameter dengan nilai. `AES_256` Untuk meminta kunci data 128-bit, gunakan `key-spec` parameter dengan nilai. `AES_128` Untuk semua panjang kunci data lainnya, gunakan `number-of-bytes` parameter.

Kunci KMS yang Anda tentukan harus berupa kunci KMS enkripsi simetris, yaitu kunci KMS dengan nilai spesifikasi kunci SYMMETRIC_DEFAULT.

```
aws kms generate-data-key-without-plaintext \
    --key-id "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --key-spec AES_256
```

Output:

```
{
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

`CiphertextBlob`(Kunci data terenkripsi) dikembalikan dalam format yang dikodekan base64.

Untuk informasi selengkapnya, lihat Kunci data di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat GenerateDataKeyWithoutPlaintextdi Referensi AWS CLI Perintah.

Rust

SDK for Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn make_key(client: &Client, key: &str) -> Result<(), Error> {
    let resp = client
        .generate_data_key_without_plaintext()
        .key_id(key)
        .key_spec(DataKeySpec::Aes256)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
}
```

- Untuk detail API, lihat GenerateDataKeyWithoutPlaintextreferensi AWS SDK for Rust API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`GenerateRandom`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GenerateRandom`.

CLI

AWS CLI

Contoh 1: Untuk menghasilkan string byte acak 256-bit (Linux atau macOS)

`generate-random`Contoh berikut menghasilkan 256-bit (32-byte), string byte acak yang dikodekan base64. Contoh menerjemahkan string byte dan menyimpannya dalam file acak.

Ketika Anda menjalankan perintah ini, Anda harus menggunakan `number-of-bytes` parameter untuk menentukan panjang nilai acak dalam byte.

Anda tidak menentukan kunci KMS ketika Anda menjalankan perintah ini. String byte acak tidak terkait dengan kunci KMS apa pun.

Secara default, AWS KMS menghasilkan nomor acak. Namun, jika Anda menentukan penyimpanan kunci khusus, string byte acak dihasilkan di klaster AWS CloudHSM yang terkait dengan penyimpanan kunci kustom.

Contoh ini menggunakan parameter dan nilai berikut:

Ini menggunakan `--number-of-bytes` parameter yang diperlukan dengan nilai `32` untuk meminta string 32-byte (256-bit). Ini menggunakan `--output` parameter dengan nilai untuk `text` mengarahkan AWS CLI untuk mengembalikan output sebagai teks, bukan JSON.Ini menggunakan untuk mengekstrak nilai `Plaintext` properti dari response.It pipa (|) output dari perintah `--query parameter` ke `base64` utilitas, yang menerjemahkan output yang diekstraksi.Ini menggunakan operator pengalihan (**>**) untuk menyimpan string byte yang diterjemahkan ke file. Ini menggunakan operator pengalihan (**>**) `ExampleRandom` untuk menyimpan ciphertext biner ke file.

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
    --query Plaintext | base64 --decode > ExampleRandom
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat GenerateRandomdi Referensi API Layanan Manajemen AWS Kunci.

Contoh 2: Untuk menghasilkan nomor acak 256-bit (Windows Command Prompt)

Contoh berikut menggunakan `generate-random` perintah untuk menghasilkan string byte acak 256-bit (32-byte) yang dikodekan base64. Contoh menerjemahkan string byte dan menyimpannya dalam file acak. Contoh ini sama dengan contoh sebelumnya, kecuali bahwa ia menggunakan `certutil` utilitas di Windows untuk base64-decode string byte acak sebelum menyimpannya dalam file.

Pertama, buat string byte acak yang dikodekan base64 dan simpan dalam file sementara,. `ExampleRandom.base64`

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
    --query Plaintext > ExampleRandom.base64
```

Karena output dari `generate-random` perintah disimpan dalam file, contoh ini tidak menghasilkan output.

Sekarang gunakan `certutil -decode` perintah untuk memecahkan kode string byte yang dikodekan base64 dalam file. `ExampleRandom.base64` Kemudian, ia menyimpan string byte yang diterjemahkan dalam file. `ExampleRandom`

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

Output:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Untuk informasi selengkapnya, lihat GenerateRandomdi Referensi API Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat GenerateRandomdi Referensi AWS CLI Perintah.

Rust

SDK for Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn make_string(client: &Client, length: i32) -> Result<(), Error> {
    let resp = client
        .generate_random()
        .number_of_bytes(length)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.plaintext.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
}
```

- Untuk detail API, lihat GenerateRandomreferensi AWS SDK for Rust API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`GetKeyPolicy`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetKeyPolicy`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

CLI

AWS CLI

Untuk menyalin kebijakan kunci dari satu kunci KMS ke kunci KMS lainnya

`get-key-policy`Contoh berikut mendapatkan kebijakan kunci dari satu kunci KMS dan menyimpannya dalam file teks. Kemudian, itu menggantikan kebijakan kunci KMS yang berbeda menggunakan file teks sebagai input kebijakan.

Karena `--policy` parameter `put-key-policy` membutuhkan string, Anda harus menggunakan `--output text` opsi untuk mengembalikan output sebagai string teks bukan JSON.

```
aws kms get-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --query Policy \
    --output text > policy.txt

aws kms put-key-policy \
    --policy-name default \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --policy file://policy.txt
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat PutKeyPolicydi Referensi API AWS KMS.

- Untuk detail API, lihat GetKeyPolicydi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def get_policy(self, key_id: str) -> dict[str, str]:
        """
        Gets the policy of a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The key policy as a dict.
        """
        if key_id != "":
            try:
                response = self.kms_client.get_key_policy(
                    KeyId=key_id,
                )
                policy = json.loads(response["Policy"])
            except ClientError as err:
                logger.error(
                    "Couldn't get policy for key %s. Here's why: %s",
```

```
                    key_id,
                    err.response["Error"]["Message"],
                )
                raise
            else:
                pprint(policy)
                return policy
        else:
            print("Skipping get policy demo.")
```

- Untuk detail API, lihat GetKeyPolicydi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->getkeypolicy(
          iv_keyid = iv_key_id
          iv_policyname = 'default'
        ).
        MESSAGE 'Retrieved key policy successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GetKeyPolicydi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `ListAliases` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListAliases`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.KeyManagementService;
    using Amazon.KeyManagementService.Model;

    /// <summary>
    /// List the AWS Key Management Service (AWS KMS) aliases that have been
defined for
    /// the keys in the same AWS Region as the default user. If you want to list
    /// the aliases in a different Region, pass the Region to the client
    /// constructor.
    /// </summary>
    public class ListAliases
    {
        public static async Task Main()
        {
            var client = new AmazonKeyManagementServiceClient();
            var request = new ListAliasesRequest();
```

```
            var response = new ListAliasesResponse();

            do
            {
                response = await client.ListAliasesAsync(request);

                response.Aliases.ForEach(alias =>
                {
                    Console.WriteLine($"Created: {alias.CreationDate} Last
Update: {alias.LastUpdatedDate} Name: {alias.AliasName}");
                });

                request.Marker = response.NextMarker;
            }
            while (response.Truncated);
        }
    }
```

- Untuk detail API, lihat ListAliasesdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Contoh 1: Untuk mencantumkan semua alias di AWS akun dan Wilayah

Contoh berikut menggunakan `list-aliases` perintah untuk daftar semua alias di Wilayah default AWS akun. Outputnya mencakup alias yang terkait dengan kunci KMS AWS terkelola dan kunci KMS yang dikelola pelanggan.

```
aws kms list-aliases
```

Output:

```
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/testKey",
            "AliasName": "alias/testKey",
```

```
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/FinanceDept",
            "AliasName": "alias/FinanceDept",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "AliasName": "alias/aws/dynamodb",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "AliasName": "alias/aws/ebs",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef"
        },
        ...
    ]
}
```

Contoh 2: Untuk daftar semua alias untuk kunci KMS tertentu

Contoh berikut menggunakan `list-aliases` perintah dan `key-id` parameternya untuk daftar semua alias yang terkait dengan kunci KMS tertentu.

Setiap alias dikaitkan dengan hanya satu kunci KMS, tetapi kunci KMS dapat memiliki beberapa alias. Perintah ini sangat berguna karena konsol AWS KMS hanya mencantumkan satu alias untuk setiap tombol KMS. Untuk menemukan semua alias untuk kunci KMS, Anda harus menggunakan perintah. `list-aliases`

Contoh ini menggunakan ID kunci kunci KMS untuk `--key-id` parameter, tetapi Anda dapat menggunakan ID kunci, kunci ARN, nama alias, atau alias ARN dalam perintah ini.

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

Output:

```
{
    "Aliases": [
        {
```

```
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/oregon-test-
key",
            "AliasName": "alias/oregon-test-key"
        },
        {
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project121-
test",
            "AliasName": "alias/project121-test"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Bekerja dengan Alias di Panduan Pengembang Layanan Manajemen AWS Utama.

- Untuk detail API, lihat ListAliasesdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /**
     * Asynchronously lists all the aliases in the current AWS account.
     *
     * @return a {@link CompletableFuture} that completes when the list of
 aliases has been processed
     */
    public CompletableFuture<Object> listAllAliasesAsync() {
        ListAliasesRequest aliasesRequest = ListAliasesRequest.builder()
            .limit(15)
            .build();

        ListAliasesPublisher paginator =
getAsyncClient().listAliasesPaginator(aliasesRequest);
```

```
        return paginator.subscribe(response -> {
                response.aliases().forEach(alias ->
                    logger.info("The alias name is: " + alias.aliasName())
                );
            })
            .thenApply(v -> null)
            .exceptionally(ex -> {
                if (ex.getCause() instanceof KmsException) {
                    KmsException e = (KmsException) ex.getCause();
                    throw new RuntimeException("A KMS exception occurred: " +
 e.getMessage());
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
 ex.getMessage());
                }
            });
    }
```

- Untuk detail API, lihat ListAliasesdi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun listAllAliases() {
    val request =
        ListAliasesRequest {
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listAliases(request)
        response.aliases?.forEach { alias ->
            println("The alias name is ${alias.aliasName}")
        }
```

```
    }
}
```

- Untuk detail API, lihat ListAliasesdi AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @param int $limit
     * @return ResultPaginator
     */
    public function listAliases(string $keyId = "", int $limit = 0)
    {
        $args = [];
        if($keyId){
            $args['KeyId'] = $keyId;
        }
        if($limit){
            $args['Limit'] = $limit;
        }
        try{
            return $this->client->getPaginator("ListAliases", $args);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidMarkerException"){
                echo "The request was rejected because the marker that specifies
where pagination should next begin is not valid.\n";
            }
            throw $caught;
        }
    }
```

- Untuk detail API, lihat ListAliasesdi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def list_aliases(self, page_size: int) -> None:
        """
        Lists aliases for the current account.
        :param page_size: The number of aliases to list per page.
        """
        try:
            alias_paginator = self.kms_client.get_paginator("list_aliases")
            for alias_page in alias_paginator.paginate(
                PaginationConfig={"PageSize": page_size}
            ):
                print(f"Here are {page_size} aliases:")
```

```
                pprint(alias_page["Aliases"])
                if alias_page["Truncated"]:
                    answer = input(
                        f"Do you want to see the next {page_size} aliases (y/n)?
"
                    )
                    if answer.lower() != "y":
                        break
                else:
                    print("That's all your aliases!")
        except ClientError as err:
            logging.error(
                "Couldn't list your aliases. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat ListAliasesdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      oo_result = lo_kms->listaliases( ).
      MESSAGE 'Retrieved KMS aliases list.' TYPE 'I'.
    CATCH /aws1/cx_kmskmsinternalex.
      MESSAGE 'An internal error occurred.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat ListAliasesdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListGrants`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListGrants`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   using System;
   using System.Threading.Tasks;
   using Amazon.KeyManagementService;
   using Amazon.KeyManagementService.Model;

   /// <summary>
   /// List the AWS Key Management Service (AWS KMS) grants that are associated
with
   /// a specific key.
   /// </summary>
   public class ListGrants
   {
       public static async Task Main()
       {
           // The identifier of the AWS KMS key to disable. You can use the
           // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
           var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";
           var client = new AmazonKeyManagementServiceClient();
```

```
            var request = new ListGrantsRequest
            {
                KeyId = keyId,
            };

            var response = new ListGrantsResponse();

            do
            {
                response = await client.ListGrantsAsync(request);

                response.Grants.ForEach(grant =>
                {
                    Console.WriteLine($"{grant.GrantId}");
                });

                request.Marker = response.NextMarker;
            }
            while (response.Truncated);
        }
    }
```

- Untuk detail API, lihat ListGrantsdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk melihat hibah pada kunci AWS KMS

`list-grants`Contoh berikut menampilkan semua hibah pada kunci KMS AWS terkelola yang ditentukan untuk Amazon DynamoDB di akun Anda. Hibah ini memungkinkan DynamoDB untuk menggunakan kunci KMS atas nama Anda untuk mengenkripsi tabel DynamoDB sebelum menulisnya ke disk. Anda dapat menggunakan perintah seperti ini untuk melihat hibah pada kunci KMS yang AWS dikelola dan kunci KMS yang dikelola pelanggan di AWS akun dan Wilayah.

Perintah ini menggunakan `key-id` parameter dengan ID kunci untuk mengidentifikasi kunci KMS. Anda dapat menggunakan ID kunci atau kunci ARN untuk mengidentifikasi kunci KMS.

Untuk mendapatkan ID kunci atau kunci ARN dari kunci KMS yang AWS dikelola, gunakan perintah or. `list-keys` `list-aliases`

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

Output menunjukkan bahwa hibah memberikan izin Amazon DynamoDB untuk menggunakan kunci KMS untuk operasi kriptografi, dan memberikan izin untuk melihat detail tentang kunci KMS () dan untuk menghentikan hibah `DescribeKey` (). `RetireGrant` Batasan `EncryptionContextSubset` membatasi izin ini untuk permintaan yang menyertakan pasangan konteks enkripsi yang ditentukan. Akibatnya, izin dalam hibah hanya efektif pada akun tertentu dan tabel DynamoDB.

```
{
    "Grants": [
        {
            "Constraints": {
                "EncryptionContextSubset": {
                    "aws:dynamodb:subscriberId": "123456789012",
                    "aws:dynamodb:tableName": "Services"
                }
            },
            "IssuingAccount": "arn:aws:iam::123456789012:root",
            "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
            "Operations": [
                "Decrypt",
                "Encrypt",
                "GenerateDataKey",
                "ReEncryptFrom",
                "ReEncryptTo",
                "RetireGrant",
                "DescribeKey"
            ],
            "GrantId":
 "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59",
            "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
            "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
            "CreationDate": "2021-05-13T18:32:45.144000+00:00"
        }
    ]
```

```
}
```

Untuk informasi selengkapnya, lihat Hibah di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Utama.

- Untuk detail API, lihat ListGrantsdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /**
    * Asynchronously displays the grant IDs for the specified key ID.
    *
    * @param keyId the ID of the AWS KMS key for which to list the grants
    * @return a {@link CompletableFuture} that, when completed, will be null
if the operation succeeded, or will throw a {@link RuntimeException} if the
operation failed
    * @throws RuntimeException if there was an error listing the grants, either
due to an {@link KmsException} or an unexpected error
    */
   public CompletableFuture<Object> displayGrantIdsAsync(String keyId) {
       ListGrantsRequest grantsRequest = ListGrantsRequest.builder()
           .keyId(keyId)
           .limit(15)
           .build();

       ListGrantsPublisher paginator =
getAsyncClient().listGrantsPaginator(grantsRequest);
       return paginator.subscribe(response -> {
               response.grants().forEach(grant -> {
                   logger.info("The grant Id is: " + grant.grantId());
               });
           })
           .thenApply(v -> null)
           .exceptionally(ex -> {
```

```
                Throwable cause = ex.getCause();
                if (cause instanceof KmsException) {
                    throw new RuntimeException("Failed to list grants: " +
 cause.getMessage(), cause);
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
 cause.getMessage(), cause);
                }
            });
    }
```

- Untuk detail API, lihat ListGrantsdi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun displayGrantIds(keyIdVal: String?) {
    val request =
        ListGrantsRequest {
            keyId = keyIdVal
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listGrants(request)
        response.grants?.forEach { grant ->
            println("The grant Id is ${grant.grantId}")
        }
    }
}
```

- Untuk detail API, lihat ListGrantsdi AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @return Result
     */
    public function listGrants(string $keyId)
    {
        try{
            return $this->client->listGrants([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "    The request was rejected because the specified entity
or resource could not be found.\n";
            }
            throw $caught;
        }
    }
```

- Untuk detail API, lihat ListGrantsdi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def list_grants(self, key_id):
        """
        Lists grants for a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The grants for the key.
        """
        try:
            paginator = self.kms_client.get_paginator("list_grants")
            grants = []
            page_iterator = paginator.paginate(KeyId=key_id)
            for page in page_iterator:
                grants.extend(page["Grants"])

            print(f"Grants for key {key_id}:")
            pprint(grants)
```

```
            return grants
        except ClientError as err:
            logger.error(
                "Couldn't list grants for key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat ListGrantsdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->listgrants( iv_keyid = iv_key_id ).
        MESSAGE 'Retrieved grants list.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListGrantsdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListKeyPolicies`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListKeyPolicies`.

CLI

AWS CLI

Untuk mendapatkan nama kebijakan kunci untuk kunci KMS

`list-key-policies`Contoh berikut mendapatkan nama-nama kebijakan utama untuk kunci yang dikelola pelanggan di akun contoh dan Wilayah. Anda dapat menggunakan perintah ini untuk menemukan nama kebijakan kunci untuk kunci AWS terkelola dan kunci yang dikelola pelanggan.

Karena satu-satunya nama kebijakan kunci yang valid adalah`default`, perintah ini tidak berguna.

Untuk menentukan kunci KMS, gunakan `key-id` parameter. Contoh ini menggunakan nilai ID kunci, tetapi Anda dapat menggunakan ID kunci atau kunci ARN dalam perintah ini.

```
aws kms list-key-policies \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{
    "PolicyNames": [
    "default"
    ]
}
```

Untuk informasi selengkapnya tentang kebijakan kunci AWS KMS, lihat Menggunakan Kebijakan Utama di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat ListKeyPoliciesdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /**
    * Asynchronously retrieves the key policy for the specified key ID and
policy name.
    *
    * @param keyId       the ID of the AWS KMS key for which to retrieve the
policy
    * @param policyName the name of the key policy to retrieve
    * @return a {@link CompletableFuture} that, when completed, contains the key
policy as a {@link String}
    */
   public CompletableFuture<String> getKeyPolicyAsync(String keyId, String
policyName) {
       GetKeyPolicyRequest policyRequest = GetKeyPolicyRequest.builder()
           .keyId(keyId)
           .policyName(policyName)
           .build();

       return getAsyncClient().getKeyPolicy(policyRequest)
           .thenApply(response -> {
               String policy = response.policy();
               logger.info("The response is: " + policy);
               return policy;
           })
           .exceptionally(ex -> {
               throw new RuntimeException("Failed to get key policy", ex);
           });
   }
```

- Untuk detail API, lihat ListKeyPoliciesdi Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def list_policies(self, key_id):
        """
        Lists the names of the policies for a key.

        :param key_id: The ARN or ID of the key to query.
        """
        try:
            policy_names = self.kms_client.list_key_policies(KeyId=key_id)[
                "PolicyNames"
            ]
        except ClientError as err:
            logging.error(
                "Couldn't list your policies. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

```
        else:
            print(f"The policies for key {key_id} are:")
            pprint(policy_names)
```

- Untuk detail API, lihat ListKeyPoliciesdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->listkeypolicies( iv_keyid = iv_key_id ).
        MESSAGE 'Retrieved key policies list.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListKeyPoliciesdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListKeys`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListKeys`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.KeyManagementService;
    using Amazon.KeyManagementService.Model;

    /// <summary>
    /// List the AWS Key Managements Service (AWS KMS) keys for the AWS Region
    /// of the default user. To list keys in another AWS Region, supply the
Region
    /// as a parameter to the client constructor.
    /// </summary>
    public class ListKeys
    {
        public static async Task Main()
        {
            var client = new AmazonKeyManagementServiceClient();
            var request = new ListKeysRequest();
            var response = new ListKeysResponse();

            do
            {
                response = await client.ListKeysAsync(request);

                response.Keys.ForEach(key =>
                {
                    Console.WriteLine($"ID: {key.KeyId}, {key.KeyArn}");
```

```
                });

                // Set the Marker property when response.Truncated is true
                // in order to get the next keys.
                request.Marker = response.NextMarker;
            }
            while (response.Truncated);
        }
    }
```

- Untuk detail API, lihat ListKeysdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk mendapatkan kunci KMS di akun dan Wilayah

`list-keys`Contoh berikut mendapatkan kunci KMS di akun dan Wilayah. Perintah ini mengembalikan kunci AWS terkelola dan kunci yang dikelola pelanggan.

```
aws kms list-keys
```

Output:

```
{
    "Keys": [
        {
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
            "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
        },
        {
            "KeyArn": "arn:aws:kms:us-
east-2:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
```

```
            "KeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Melihat AWS Kunci di Panduan Pengembang Layanan Manajemen Kunci.

- Untuk detail API, lihat ListKeysdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
```

```
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /*
         * The `subscribe` method is required when using paginator methods in the
 AWS SDK
         * because paginator methods return an instance of a `ListKeysPublisher`,
 which is
         * based on a reactive stream. This allows asynchronous retrieval of
 paginated
         * results as they become available. By subscribing to the stream, we can
 process
         * each page of results as they are emitted.
         */
        ListKeysPublisher keysPublisher =
 kmsAsyncClient.listKeysPaginator(listKeysRequest);
        CompletableFuture<Void> future = keysPublisher
            .subscribe(r -> r.keys().forEach(key ->
                System.out.println("The key ARN is: " + key.keyArn() + ". The key
 Id is: " + key.keyId())))
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    System.err.println("Error occurred: " +
 exception.getMessage());
                } else {
                    System.out.println("Successfully listed all keys.");
                }
            });

        try {
            future.join();
        } catch (Exception e) {
            System.err.println("Failed to list keys: " + e.getMessage());
        }
    }
}
```

- Untuk detail API, lihat ListKeysdi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun listAllKeys() {
    val request =
        ListKeysRequest {
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listKeys(request)
        response.keys?.forEach { key ->
            println("The key ARN is ${key.keyArn}")
            println("The key Id is ${key.keyId}")
        }
    }
}
```

- Untuk detail API, lihat ListKeysdi AWS SDK untuk referensi API Kotlin.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
```

```
     * @return array
     */
    public function listKeys()
    {
        try {
            $contents = [];
            $paginator = $this->client->getPaginator("ListKeys");
            foreach($paginator as $result){
                foreach ($result['Content'] as $object) {
                    $contents[] = $object;
                }
            }
            return $contents;
        }catch(KmsException $caught){
            echo "There was a problem listing the keys: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- Untuk detail API, lihat ListKeysdi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
```

```
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def list_keys(self):
        """
        Lists the keys for the current account by using a paginator.
        """
        try:
            page_size = 10
            print("\nLet's list your keys.")
            key_paginator = self.kms_client.get_paginator("list_keys")
            for key_page in key_paginator.paginate(PaginationConfig={"PageSize":
10}):
                print(f"Here are {len(key_page['Keys'])} keys:")
                pprint(key_page["Keys"])
                if key_page["Truncated"]:
                    answer = input(
                        f"Do you want to see the next {page_size} keys (y/n)? "
                    )
                    if answer.lower() != "y":
                        break
                else:
                    print("That's all your keys!")
        except ClientError as err:
            logging.error(
                "Couldn't list your keys. Here's why: %s",
                err.response["Error"]["Message"],
            )
```

- Untuk detail API, lihat ListKeysdi AWS SDK for Python (Boto3) Referensi API.

Rust

SDK for Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn show_keys(client: &Client) -> Result<(), Error> {
    let resp = client.list_keys().send().await?;

    let keys = resp.keys.unwrap_or_default();

    let len = keys.len();

    for key in keys {
        println!("Key ARN: {}", key.key_arn.as_deref().unwrap_or_default());
    }

    println!();
    println!("Found {} keys", len);

    Ok(())
}
```

- Untuk detail API, lihat ListKeysreferensi AWS SDK for Rust API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kms->listkeys( ).
        MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListKeysdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`PutKeyPolicy`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`PutKeyPolicy`.

CLI

AWS CLI

Untuk mengubah kebijakan kunci untuk kunci KMS

`put-key-policy`Contoh berikut mengubah kebijakan kunci untuk kunci yang dikelola pelanggan.

Untuk memulai, buat kebijakan kunci dan simpan dalam file JSON lokal. Dalam contoh ini, file tersebut adalah`key_policy.json`. Anda juga dapat menentukan kebijakan kunci sebagai nilai string `policy` parameter.

Pernyataan pertama dalam kebijakan kunci ini memberikan izin AWS akun untuk menggunakan kebijakan IAM untuk mengontrol akses ke kunci KMS. Pernyataan kedua memberikan izin `test-user` pengguna untuk menjalankan `list-keys` perintah `describe-key` dan pada kunci KMS.

Isi dari `key_policy.json`:

```
{
    "Version":"2012-10-17",
    "Id" : "key-default-1",
```

```
    "Statement" : [
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
            "Resource" : "*"
        },
        {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [
                "kms:DescribeKey",
                "kms:ListKeys"
            ],
            "Resource" : "*"
        }
    ]
}
```

Untuk mengidentifikasi kunci KMS, contoh ini menggunakan ID kunci, tetapi Anda juga dapat menggunakan kunci ARN. Untuk menentukan kebijakan kunci, perintah menggunakan `policy` parameter. Untuk menunjukkan bahwa kebijakan tersebut ada dalam file, kebijakan tersebut menggunakan `file://` awalan yang diperlukan. Awalan ini diperlukan untuk mengidentifikasi file pada semua sistem operasi yang didukung. Akhirnya, perintah menggunakan `policy-name` parameter dengan nilai`default`. Jika tidak ada nama kebijakan yang ditentukan, nilai defaultnya adalah`default`. Satu-satunya nilai yang valid adalah `default`.

```
aws kms put-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --policy file://key_policy.json
```

Perintah ini tidak menghasilkan output apa pun. Untuk memverifikasi bahwa perintah itu efektif, gunakan `get-key-policy` perintah. Contoh perintah berikut mendapatkan kebijakan kunci

untuk kunci KMS yang sama. `output`Parameter dengan nilai `text` mengembalikan format teks yang mudah dibaca.

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

Output:

```
{
    "Version":"2012-10-17",
    "Id" : "key-default-1",
    "Statement" : [
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
            "Resource" : "*"
            },
            {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [ "kms:Describe", "kms:List" ],
            "Resource" : "*"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Mengubah Kebijakan Utama di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat PutKeyPolicydi Referensi AWS CLI Perintah.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @param string $policy
     * @return void
     */
    public function putKeyPolicy(string $keyId, string $policy)
    {
        try {
            $this->client->putKeyPolicy([
                'KeyId' => $keyId,
                'Policy' => $policy,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem replacing the key policy: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- Untuk detail API, lihat PutKeyPolicydi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def set_policy(self, key_id: str, policy: dict[str, any]) -> None:
        """
        Sets the policy of a key. Setting a policy entirely overwrites the
 existing
        policy, so care is taken to add a statement to the existing list of
 statements
        rather than simply writing a new policy.

        :param key_id: The ARN or ID of the key to set the policy to.
        :param policy: The existing policy of the key.
        :return: None
        """
        principal = input(
            "Enter the ARN of an IAM role to set as the principal on the policy:
 "
        )
```

```
        if key_id != "" and principal != "":
            # The updated policy replaces the existing policy. Add a new
statement to
            # the list along with the original policy statements.
            policy["Statement"].append(
                {
                    "Sid": "Allow access for ExampleRole",
                    "Effect": "Allow",
                    "Principal": {"AWS": principal},
                    "Action": [
                        "kms:Encrypt",
                        "kms:GenerateDataKey*",
                        "kms:Decrypt",
                        "kms:DescribeKey",
                        "kms:ReEncrypt*",
                    ],
                    "Resource": "*",
                }
            )
            try:
                self.kms_client.put_key_policy(KeyId=key_id,
Policy=json.dumps(policy))
            except ClientError as err:
                logger.error(
                    "Couldn't set policy for key %s. Here's why %s",
                    key_id,
                    err.response["Error"]["Message"],
                )
                raise
            else:
                print(f"Set policy for key {key_id}.")
        else:
            print("Skipping set policy demo.")
```

- Untuk detail API, lihat PutKeyPolicydi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_policy = '{"Version":"2012-10-17",        "Statement": [...]}'
        lo_kms->putkeypolicy(
          iv_keyid = iv_key_id
          iv_policyname = 'default'
          iv_policy = iv_policy
        ).
        MESSAGE 'Key policy updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsmalformedplydocex.
        MESSAGE 'Malformed policy document.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat PutKeyPolicydi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ReEncrypt`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ReEncrypt`.

CLI

AWS CLI

Contoh 1: Untuk mengenkripsi ulang pesan terenkripsi di bawah kunci KMS simetris yang berbeda (Linux dan macOS).

Contoh `re-encrypt` perintah berikut menunjukkan cara yang disarankan untuk mengenkripsi ulang data dengan CLI. AWS

Berikan ciphertext dalam file.Dalam nilai `--ciphertext-blob` parameter, gunakan `fileb://` awalan, yang memberitahu CLI untuk membaca data dari file biner. Jika file tidak ada di direktori saat ini, ketik path lengkap ke file. Untuk informasi lebih lanjut tentang membaca nilai parameter AWS CLI dari file, lihat Memuat parameter AWS CLI dari file di Panduan Pengguna Antarmuka Baris AWS Perintah dan Praktik Terbaik untuk Parameter File Lokal di Blog Alat Baris AWS Perintah. Tentukan kunci KMS sumber, yang mendekripsi ciphertext. Parameter tidak diperlukan saat mendekripsi dengan kunci KMS enkripsi `--source-key-id` simetris. AWS KMS bisa 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 Anda inginkan, dan mencegah Anda mendekripsi ciphertext secara tidak sengaja menggunakan kunci KMS yang tidak Anda percayai. Tentukan kunci KMS tujuan, yang mengenkripsi ulang data.Parameter selalu diperlukan. `--destination-key-id` Contoh ini menggunakan ARN kunci, tetapi Anda dapat menggunakan pengidentifikasi kunci yang valid.Minta output teks biasa sebagai nilai teks.Parameter `--query` memberitahu CLI untuk mendapatkan hanya nilai bidang dari output. `Plaintext` `--output`Parameter mengembalikan output sebagai text.base64-decode plaintext dan menyimpannya dalam file.Contoh berikut pipa (|) nilai `Plaintext` parameter ke utilitas Base64, yang menerjemahkan itu. Kemudian, itu mengalihkan (>) output yang diterjemahkan ke file. `ExamplePlaintext`

Sebelum menjalankan perintah ini, ganti kunci contoh IDs dengan pengidentifikasi kunci yang valid dari AWS akun Anda.

```
aws kms re-encrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --query CiphertextBlob \
```

```
    --output text | base64 --decode > ExampleReEncryptedFile
```

Perintah ini tidak menghasilkan output. Output dari `re-encrypt` perintah adalah base64-decoded dan disimpan dalam file.

Untuk informasi selengkapnya, lihat ReEncryptdi Referensi API Layanan Manajemen AWS Kunci.

Contoh 2: Untuk mengenkripsi ulang pesan terenkripsi di bawah tombol KMS simetris yang berbeda (prompt perintah Windows).

Contoh `re-encrypt` perintah berikut adalah sama dengan yang sebelumnya kecuali bahwa ia menggunakan `certutil` utilitas untuk Base64-decode data plaintext. Prosedur ini membutuhkan dua perintah, seperti yang ditunjukkan pada contoh berikut.

Sebelum menjalankan perintah ini, ganti ID kunci contoh dengan ID kunci yang valid dari AWS akun Anda.

```
aws kms re-encrypt ^
    --ciphertext-blob fileb://ExampleEncryptedFile ^
    --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
    --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 ^
    --query CiphertextBlob ^
    --output text > ExampleReEncryptedFile.base64
```

Kemudian gunakan `certutil` utilitas

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Output:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Untuk informasi selengkapnya, lihat ReEncryptdi Referensi API Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat ReEncryptdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def re_encrypt(self, source_key_id, cipher_text):
        """
        Takes ciphertext previously encrypted with one key and reencrypt it by
 using
        another key.

        :param source_key_id: The ARN or ID of the original key used to encrypt
 the
                              ciphertext.
        :param cipher_text: The encrypted ciphertext.
        :return: The ciphertext encrypted by the second key.
        """
        destination_key_id = input(
            f"Your ciphertext is currently encrypted with key {source_key_id}. "
            f"Enter another key ID or ARN to reencrypt it: "
        )
```

```
        if destination_key_id != "":
            try:
                cipher_text = self.kms_client.re_encrypt(
                    SourceKeyId=source_key_id,
                    DestinationKeyId=destination_key_id,
                    CiphertextBlob=cipher_text,
                )["CiphertextBlob"]
            except ClientError as err:
                logger.error(
                    "Couldn't reencrypt your ciphertext. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            else:
                print(f"Reencrypted your ciphertext as: {cipher_text}")
                return cipher_text
        else:
            print("Skipping reencryption demo.")
```

- Untuk detail API, lihat ReEncryptdi AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Human-readable version of the ciphertext of the data to reencrypt.

blob =
 '01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593
sourceCiphertextBlob = [blob].pack('H*')

# Replace the fictitious key ARN with a valid key ID
```

```
destinationKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.re_encrypt({
                           ciphertext_blob: sourceCiphertextBlob,
                           destination_key_id: destinationKeyId
                         })

# Display a readable version of the resulting re-encrypted blob.
puts 'Blob:'
puts resp.ciphertext_blob.unpack('H*')
```

- Untuk detail API, lihat ReEncryptdi Referensi AWS SDK untuk Ruby API.

Rust

SDK for Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn reencrypt_string(
    verbose: bool,
    client: &Client,
    input_file: &str,
    output_file: &str,
    first_key: &str,
    new_key: &str,
) -> Result<(), Error> {
    // Get blob from input file
    // Open input text file and get contents as a string
    // input is a base-64 encoded string, so decode it:
    let data = fs::read_to_string(input_file)
        .map(|input_file| base64::decode(input_file).expect("invalid base 64"))
        .map(Blob::new);
```

```
    let resp = client
        .re_encrypt()
        .ciphertext_blob(data.unwrap())
        .source_key_id(first_key)
        .destination_key_id(new_key)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);
    let o = &output_file;

    let mut ofile = File::create(o).expect("unable to create file");
    ofile.write_all(s.as_bytes()).expect("unable to write");

    if verbose {
        println!("Wrote the following to {}:", output_file);
        println!("{}", s);
    } else {
        println!("Wrote base64-encoded output to {}", output_file);
    }

    Ok(())
}
```

- Untuk detail API, lihat ReEncryptreferensi AWS SDK for Rust API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_source_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_destination_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'
        " iv_ciphertext_blob contains the encrypted data
        oo_result = lo_kms->reencrypt(
          iv_sourcekeyid = iv_source_key_id
          iv_destinationkeyid = iv_destination_key_id
          iv_ciphertextblob = iv_ciphertext_blob
        ).
        MESSAGE 'Ciphertext reencrypted successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsincorrectkeyex.
        MESSAGE 'Incorrect source key for decryption.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ReEncryptdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`RetireGrant`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`RetireGrant`.

CLI

AWS CLI

Untuk pensiun hibah pada kunci master pelanggan

`retire-grant`Contoh berikut menghapus hibah dari kunci KMS.

Contoh perintah berikut menentukan `grant-id` dan `key-id` parameter. Nilai `key-id` parameter harus menjadi kunci ARN dari kunci KMS.

```
aws kms retire-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Perintah ini tidak menghasilkan output. Untuk mengonfirmasi bahwa hibah telah pensiun, gunakan `list-grants` perintah.

Untuk informasi selengkapnya, lihat Pensiun dan pencabutan hibah di Panduan Pengembang Layanan Manajemen AWS Utama.

- Untuk detail API, lihat RetireGrantdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)
```

```
    def retire_grant(self, grant):
        """
        Retires a grant so that it can no longer be used.

        :param grant: The grant to retire.
        """
        try:
            self.kms_client.retire_grant(GrantToken=grant["GrantToken"])
        except ClientError as err:
            logger.error(
                "Couldn't retire grant %s. Here's why: %s",
                grant["GrantId"],
                err.response["Error"]["Message"],
            )
        else:
            print(f"Grant {grant['GrantId']} retired.")
```

- Untuk detail API, lihat RetireGrantdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_grant_token = 'AQpAM2RhZ...'
        lo_kms->retiregrant( iv_granttoken = iv_grant_token ).
        MESSAGE 'Grant retired successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Grant not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvgranttokenex.
        MESSAGE 'Invalid grant token.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
```

```
    ENDTRY.
```

- Untuk detail API, lihat RetireGrantdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`RevokeGrant`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`RevokeGrant`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

CLI

AWS CLI

Untuk mencabut hibah pada kunci master pelanggan

`revoke-grant`Contoh berikut menghapus hibah dari kunci KMS. Contoh perintah berikut menentukan `grant-id` dan `key-id` parameter. Nilai `key-id` parameter dapat berupa ID kunci atau kunci ARN dari kunci KMS.

```
aws kms revoke-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Perintah ini tidak menghasilkan output. Untuk mengonfirmasi bahwa hibah telah dicabut, gunakan `list-grants` perintah.

Untuk informasi selengkapnya, lihat Pensiun dan pencabutan hibah di Panduan Pengembang Layanan Manajemen AWS Utama.

- Untuk detail API, lihat RevokeGrantdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /**
    * Revokes a grant for the specified AWS KMS key asynchronously.
    *
    * @param keyId   The ID or key ARN of the AWS KMS key.
    * @param grantId The identifier of the grant to be revoked.
    * @return A {@link CompletableFuture} representing the asynchronous
operation of revoking the grant.
    *         The {@link CompletableFuture} will complete with a {@link
RevokeGrantResponse} object
    *         if the operation is successful, or with a {@code null} value if an
error occurs.
    */
   public CompletableFuture<RevokeGrantResponse> revokeKeyGrantAsync(String
keyId, String grantId) {
       RevokeGrantRequest grantRequest = RevokeGrantRequest.builder()
           .keyId(keyId)
           .grantId(grantId)
           .build();

       CompletableFuture<RevokeGrantResponse> responseFuture =
getAsyncClient().revokeGrant(grantRequest);
       responseFuture.whenComplete((response, exception) -> {
           if (exception == null) {
               logger.info("Grant ID: [" + grantId + "] was successfully
revoked!");
           } else {
               if (exception instanceof KmsException kmsEx) {
                   if (kmsEx.getMessage().contains("Grant does not exist")) {
                       logger.info("The grant ID '" + grantId + "' does not
exist. Moving on...");
                   } else {
```

```
                        throw new RuntimeException("KMS error occurred: " +
 kmsEx.getMessage(), kmsEx);
                    }
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
 exception.getMessage(), exception);
                }
            }
        });

        return responseFuture;
    }
```

- Untuk detail API, lihat RevokeGrantdi Referensi AWS SDK for Java 2.x API.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $grantId
     * @param string $keyId
     * @return void
     */
    public function revokeGrant(string $grantId, string $keyId)
    {
        try{
            $this->client->revokeGrant([
                'GrantId' => $grantId,
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem with revoking the grant: {$caught-
>getAwsErrorMessage()}.\n";
```

```
            throw $caught;
        }
    }
```

- Untuk detail API, lihat RevokeGrantdi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def revoke_grant(self, key_id: str, grant_id: str) -> None:
        """
        Revokes a grant so that it can no longer be used.

        :param key_id: The ARN or ID of the key associated with the grant.
        :param grant_id: The ID of the grant to revoke.
        """
        try:
```

```
            self.kms_client.revoke_grant(KeyId=key_id, GrantId=grant_id)
        except ClientError as err:
            logger.error(
                "Couldn't revoke grant %s. Here's why: %s",
                grant_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat RevokeGrantdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_grant_id = '1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p'
        lo_kms->revokegrant(
          iv_keyid = iv_key_id
          iv_grantid = iv_grant_id
        ).
        MESSAGE 'Grant revoked successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Grant or key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvalidgrantidex.
        MESSAGE 'Invalid grant ID.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat RevokeGrantdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ScheduleKeyDeletion`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ScheduleKeyDeletion`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

CLI

AWS CLI

Untuk menjadwalkan penghapusan kunci KMS yang dikelola pelanggan.

`schedule-key-deletion`Contoh berikut menjadwalkan kunci KMS terkelola pelanggan yang ditentukan untuk dihapus dalam 15 hari.

`--key-id`Parameter mengidentifikasi kunci KMS. Contoh ini menggunakan nilai ARN kunci, tetapi Anda dapat menggunakan ID kunci atau ARN dari kunci `--pending-window-in-days` KMS.Parameter menentukan panjang masa tunggu 7-30 hari. Secara default, masa tunggu adalah 30 hari. Contoh ini menentukan nilai 15, yang memberitahu AWS untuk secara permanen menghapus kunci KMS 15 hari setelah perintah selesai.

```
aws kms schedule-key-deletion \
    --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --pending-window-in-days 15
```

Respons termasuk ARN kunci, status kunci, periode tunggu (`PendingWindowInDays`), dan tanggal penghapusan dalam waktu Unix. Untuk melihat tanggal penghapusan dalam waktu setempat, gunakan konsol AWS KMS. Kunci KMS dalam status `PendingDeletion` kunci tidak dapat digunakan dalam operasi kriptografi.

```
{
    "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "DeletionDate": "2022-06-18T23:43:51.272000+00:00",
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 15
}
```

Untuk informasi selengkapnya, lihat Menghapus kunci di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat ScheduleKeyDeletiondi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /**
    * Deletes a KMS key asynchronously.
    *
    * <p><strong>Warning:</strong> Deleting a KMS key is a destructive and
potentially dangerous operation.
    * When a KMS key is deleted, all data that was encrypted under the KMS key
becomes unrecoverable.
    * This means that any files, databases, or other data that were encrypted
using the deleted KMS key
    * will become permanently inaccessible. Exercise extreme caution when
deleting KMS keys.</p>
    *
    * @param keyId the ID of the KMS key to delete
    * @return a {@link CompletableFuture} that completes when the key deletion
is scheduled
    */
   public CompletableFuture<Void> deleteKeyAsync(String keyId) {
```

```
        ScheduleKeyDeletionRequest deletionRequest =
ScheduleKeyDeletionRequest.builder()
            .keyId(keyId)
            .pendingWindowInDays(7)
            .build();

        return getAsyncClient().scheduleKeyDeletion(deletionRequest)
            .thenRun(() -> {
                logger.info("Key {} will be deleted in 7 days", keyId);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to schedule key deletion for
key ID: " + keyId, throwable);
            });
    }
```

- Untuk detail API, lihat ScheduleKeyDeletiondi Referensi AWS SDK for Java 2.x API.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @param int $pendingWindowInDays
     * @return void
     */
    public function scheduleKeyDeletion(string $keyId, int $pendingWindowInDays =
7)
    {
        try {
            $this->client->scheduleKeyDeletion([
                'KeyId' => $keyId,
                'PendingWindowInDays' => $pendingWindowInDays,
```

```
            ]);
        }catch(KmsException $caught){
            echo "There was a problem scheduling the key deletion: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- Untuk detail API, lihat ScheduleKeyDeletiondi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def delete_key(self, key_id: str, window: int) -> None:
        """
        Deletes a list of keys.
```

```
        Warning:
        Deleting a KMS key is a destructive and potentially dangerous operation.
 When a KMS key is deleted,
        all data that was encrypted under the KMS key is unrecoverable.

        :param key_id: The ARN or ID of the key to delete.
        :param window: The waiting period, in days, before the KMS key is
 deleted.
        """

        try:
            self.kms_client.schedule_key_deletion(
                KeyId=key_id, PendingWindowInDays=window
            )
        except ClientError as err:
            logging.error(
                "Couldn't delete key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat ScheduleKeyDeletiondi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_pending_window_days = 7
        oo_result = lo_kms->schedulekeydeletion(
```

```
          iv_keyid = iv_key_id
          iv_pendingwindowindays = iv_pending_window_days
        ).
        MESSAGE 'Key scheduled for deletion.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ScheduleKeyDeletiondi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`Sign`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`Sign`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

CLI

AWS CLI

Contoh 1: Untuk menghasilkan tanda tangan digital untuk pesan

`sign`Contoh berikut menghasilkan tanda tangan kriptografi untuk pesan singkat. Output dari perintah mencakup `Signature` bidang yang dikodekan basis-64 yang dapat Anda verifikasi dengan menggunakan perintah. `verify`

Anda harus menentukan pesan yang akan ditandatangani dan algoritma penandatanganan yang didukung oleh kunci KMS asimetris Anda. Untuk mendapatkan algoritma penandatanganan untuk kunci KMS Anda, gunakan perintah. `describe-key`

Di AWS CLI v2, nilai `message` parameter harus dikodekan Base64. Atau, Anda dapat menyimpan pesan dalam file dan menggunakan `fileb://` awalan, yang memberitahu AWS CLI untuk membaca data biner dari file.

Sebelum menjalankan perintah ini, ganti ID kunci contoh dengan ID kunci yang valid dari AWS akun Anda. ID kunci harus mewakili kunci KMS asimetris dengan penggunaan kunci SIGN_VERIFY.

```
msg=(echo 'Hello World' | base64)

aws kms sign \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://UnsignedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256
```

Output:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "Signature": "ABCDEFhpyVYyTxbafE74ccSvEJLJr3zuoV1Hfymz4qv+/
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Untuk informasi selengkapnya tentang penggunaan kunci KMS asimetris di AWS KMS, lihat Kunci asimetris di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

Contoh 2: Untuk menyimpan tanda tangan digital dalam file (Linux dan macOS)

`sign`Contoh berikut menghasilkan tanda tangan kriptografi untuk pesan singkat yang disimpan dalam file lokal. Perintah juga mendapatkan `Signature` properti dari respons, Base64-mendekodekannya dan menyimpannya dalam file. ExampleSignature Anda dapat menggunakan file tanda tangan dalam `verify` perintah yang memverifikasi tanda tangan.

`sign`Perintah ini memerlukan pesan yang disandikan Base64 dan algoritma penandatanganan yang didukung oleh kunci KMS asimetris Anda. Untuk mendapatkan algoritma penandatanganan yang didukung oleh kunci KMS Anda, gunakan perintah. `describe-key`

Sebelum menjalankan perintah ini, ganti ID kunci contoh dengan ID kunci yang valid dari AWS akun Anda. ID kunci harus mewakili kunci KMS asimetris dengan penggunaan kunci SIGN_VERIFY.

```
echo 'hello world' | base64 > EncodedMessage

aws kms sign \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
    --output text \
    --query Signature | base64 --decode > ExampleSignature
```

Perintah ini tidak menghasilkan output. Contoh ini mengekstrak `Signature` properti output dan menyimpannya dalam file.

Untuk informasi selengkapnya tentang penggunaan kunci KMS asimetris di AWS KMS, lihat Kunci asimetris di AWS KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat Referensi AWS CLI Perintah Masuk.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /**
     * Asynchronously signs and verifies data using AWS KMS.
     *
     * <p>The method performs the following steps:
     * <ol>
     *     <li>Creates an AWS KMS key with the specified key spec, key usage, and
origin.</li>
     *     <li>Signs the provided message using the created KMS key and the
RSASSA-PSS-SHA-256 algorithm.</li>
```

```
    *     <li>Verifies the signature of the message using the created KMS key
and the RSASSA-PSS-SHA-256 algorithm.</li>
    * </ol>
    *
    * @return a {@link CompletableFuture} that completes with the result of the
signature verification,
    *         {@code true} if the signature is valid, {@code false} otherwise.
    * @throws KmsException if any error occurs during the KMS operations.
    * @throws RuntimeException if an unexpected error occurs.
    */
   public CompletableFuture<Boolean> signVerifyDataAsync() {
       String signMessage = "Here is the message that will be digitally signed";

       // Create an AWS KMS key used to digitally sign data.
       CreateKeyRequest createKeyRequest = CreateKeyRequest.builder()
           .keySpec(KeySpec.RSA_2048)
           .keyUsage(KeyUsageType.SIGN_VERIFY)
           .origin(OriginType.AWS_KMS)
           .build();

       return getAsyncClient().createKey(createKeyRequest)
           .thenCompose(createKeyResponse -> {
               String keyId = createKeyResponse.keyMetadata().keyId();

               SdkBytes messageBytes = SdkBytes.fromString(signMessage,
Charset.defaultCharset());
               SignRequest signRequest = SignRequest.builder()
                   .keyId(keyId)
                   .message(messageBytes)
                   .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                   .build();

               return getAsyncClient().sign(signRequest)
                   .thenCompose(signResponse -> {
                       byte[] signedBytes =
signResponse.signature().asByteArray();

                       VerifyRequest verifyRequest = VerifyRequest.builder()
                           .keyId(keyId)

.message(SdkBytes.fromByteArray(signMessage.getBytes(Charset.defaultCharset())))

.signature(SdkBytes.fromByteBuffer(ByteBuffer.wrap(signedBytes)))
```

```
.signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                            .build();

                        return getAsyncClient().verify(verifyRequest)
                            .thenApply(verifyResponse -> {
                                return (boolean) verifyResponse.signatureValid();
                            });
                    });
            })
            .exceptionally(throwable -> {
               throw new RuntimeException("Failed to sign or verify data",
throwable);
            });
    }
```

- Untuk detail API, lihat Referensi AWS SDK for Java 2.x API Masuk.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @param string $message
     * @param string $algorithm
     * @return Result
     */
    public function sign(string $keyId, string $message, string $algorithm)
    {
        try {
            return $this->client->sign([
                'KeyId' => $keyId,
                'Message' => $message,
```

```
                'SigningAlgorithm' => $algorithm,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem signing the data: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- Untuk detail API, lihat Referensi AWS SDK untuk PHP API Masuk.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def sign(self, key_id: str, message: str) -> str:
        """
        Signs a message with a key.
```

```
        :param key_id: The ARN or ID of the key to use for signing.
        :param message: The message to sign.
        :return: The signature of the message.
        """
        try:
            return self.kms_client.sign(
                KeyId=key_id,
                Message=message.encode(),
                SigningAlgorithm="RSASSA_PSS_SHA_256",
            )["Signature"]
        except ClientError as err:
            logger.error(
                "Couldn't sign your message. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat Masuk AWS SDK untuk Referensi API Python (Boto3).

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
        " iv_message contains the message to sign
        " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
        oo_result = lo_kms->sign(
          iv_keyid = iv_key_id
          iv_message = iv_message
          iv_signingalgorithm = iv_signing_algorithm
        ).
```

```
        MESSAGE 'Message signed successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvalidkeyusageex.
        MESSAGE 'Key cannot be used for signing.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Sign in AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`TagResource`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`TagResource`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Pelajari dasar-dasarnya

CLI

AWS CLI

Untuk menambahkan tag ke kunci KMS

`tag-resource`Contoh berikut menambahkan `"Purpose":"Test"` dan `"Dept":"IT"` tag ke kunci KMS yang dikelola pelanggan. Anda dapat menggunakan tag seperti ini untuk memberi label kunci KMS dan membuat kategori kunci KMS untuk izin dan audit.

Untuk menentukan kunci KMS, gunakan `key-id` parameter. Contoh ini menggunakan nilai ID kunci, tetapi Anda dapat menggunakan ID kunci atau kunci ARN dalam perintah ini.

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

```
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

Perintah ini tidak menghasilkan output. Untuk melihat tag pada tombol AWS KMS KMS, gunakan perintah. `list-resource-tags`

Untuk informasi selengkapnya tentang penggunaan tag di AWS KMS, lihat Menandai kunci di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat TagResourcedi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /**
    * Asynchronously tags a KMS key with a specific tag.
    *
    * @param keyId the ID of the KMS key to be tagged
    * @return a {@link CompletableFuture} that completes when the tagging
operation is finished
    */
   public CompletableFuture<Void> tagKMSKeyAsync(String keyId) {
       Tag tag = Tag.builder()
           .tagKey("Environment")
           .tagValue("Production")
           .build();

       TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
           .keyId(keyId)
           .tags(tag)
           .build();

       return getAsyncClient().tagResource(tagResourceRequest)
           .thenRun(() -> {
               logger.info("{} key was tagged", keyId);
```

```
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to tag the KMS key",
throwable);
            });
    }
```

- Untuk detail API, lihat TagResourcedi Referensi AWS SDK for Java 2.x API.

PHP

SDK untuk PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /***
     * @param string $keyId
     * @param array $tags
     * @return void
     */
    public function tagResource(string $keyId, array $tags)
    {
        try {
            $this->client->tagResource([
                'KeyId' => $keyId,
                'Tags' => $tags,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem applying the tag(s): {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- Untuk detail API, lihat TagResourcedi Referensi AWS SDK untuk PHP API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def tag_resource(self, key_id: str, tag_key: str, tag_value: str) -> None:
        """
        Add or edit tags on a customer managed key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        :param tag_key: Key for the tag.
        :param tag_value: Value for the tag.
        """
        try:
            self.kms_client.tag_resource(
                KeyId=key_id, Tags=[{"TagKey": tag_key, "TagValue": tag_value}]
            )
        except ClientError as err:
```

```
            logging.error(
                "Couldn't add a tag for the key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat TagResourcedi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    DATA lt_tags TYPE /aws1/cl_kmstag=>tt_taglist.

    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_tag_key = 'Environment'
        " iv_tag_value = 'Production'
        APPEND NEW /aws1/cl_kmstag(
          iv_tagkey = iv_tag_key
          iv_tagvalue = iv_tag_value
        ) TO lt_tags.

        lo_kms->tagresource(
          iv_keyid = iv_key_id
          it_tags = lt_tags
        ).
        MESSAGE 'Tag added to KMS key successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmstagexception.
        MESSAGE 'Invalid tag format.' TYPE 'E'.
```

```
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat TagResourcedi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`UpdateAlias`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`UpdateAlias`.

CLI

AWS CLI

Untuk mengaitkan alias dengan kunci KMS yang berbeda

`update-alias`Contoh berikut mengaitkan alias `alias/test-key` dengan kunci KMS yang berbeda.

`--alias-name`Parameter menentukan alias. Nilai nama alias harus dimulai dengan`alias/`. `--target-key-id` Parameter menentukan kunci KMS untuk mengaitkan dengan alias. Anda tidak perlu menentukan kunci KMS saat ini untuk alias.

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

Perintah ini tidak menghasilkan output. Untuk menemukan alias, gunakan `list-aliases` perintah.

Untuk informasi selengkapnya, lihat Memperbarui alias di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat UpdateAliasdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def update_alias(self, alias, current_key_id):
        """
        Updates an alias by assigning it to another key.

        :param alias: The alias to reassign.
        :param current_key_id: The ARN or ID of the key currently associated with
 the alias.
        """
        new_key_id = input(
            f"Alias {alias} is currently associated with {current_key_id}. "
            f"Enter another key ID or ARN that you want to associate with
 {alias}: "
        )
        if new_key_id != "":
            try:
```

```
                self.kms_client.update_alias(AliasName=alias,
 TargetKeyId=new_key_id)
            except ClientError as err:
                logger.error(
                    "Couldn't associate alias %s with key %s. Here's why: %s",
                    alias,
                    new_key_id,
                    err.response["Error"]["Message"],
                )
            else:
                print(f"Alias {alias} is now associated with key {new_key_id}.")
        else:
            print("Skipping alias update.")
```

- Untuk detail API, lihat UpdateAliasdi AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_alias_name = 'alias/my-key-alias'
        " iv_target_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'
        lo_kms->updatealias(
          iv_aliasname = iv_alias_name
          iv_targetkeyid = iv_target_key_id
        ).
        MESSAGE 'Alias updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Alias or key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat UpdateAliasdi AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`Verify`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`Verify`.

CLI

AWS CLI

Untuk memverifikasi tanda tangan digital

`verify`Perintah berikut memverifikasi tanda tangan kriptografi untuk pesan singkat yang disandikan Base64. ID kunci, pesan, jenis pesan, dan algoritma penandatanganan harus sama dengan yang digunakan untuk menandatangani pesan.

Di AWS CLI v2, nilai `message` parameter harus dikodekan Base64. Atau, Anda dapat menyimpan pesan dalam file dan menggunakan `fileb://` awalan, yang memberitahu AWS CLI untuk membaca data biner dari file.

Tanda tangan yang Anda tentukan tidak dapat dikodekan base64. Untuk bantuan decoding tanda tangan yang dikembalikan `sign` perintah, lihat contoh `sign` perintah.

Output dari perintah termasuk `SignatureValid` bidang Boolean yang menunjukkan bahwa tanda tangan telah diverifikasi. Jika validasi tanda tangan gagal, `verify` perintah juga gagal.

Sebelum menjalankan perintah ini, ganti ID kunci contoh dengan ID kunci yang valid dari AWS akun Anda.

```
aws kms verify \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
    --signature fileb://ExampleSignature
```

Output:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "SignatureValid": true,
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Untuk informasi selengkapnya tentang penggunaan kunci KMS asimetris di AWS KMS, lihat Menggunakan kunci asimetris di Panduan Pengembang Layanan Manajemen AWS Kunci.

- Untuk detail API, lihat Verifikasi di Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def verify(self, key_id: str, message: str, signature: str) -> bool:
        """
```

```
        Verifies a signature against a message.

        :param key_id: The ARN or ID of the key used to sign the message.
        :param message: The message to verify.
        :param signature: The signature to verify.
        :return: True when the signature matches the message, otherwise False.
        """
        try:
            response = self.kms_client.verify(
                KeyId=key_id,
                Message=message.encode(),
                Signature=signature,
                SigningAlgorithm="RSASSA_PSS_SHA_256",
            )
            valid = response["SignatureValid"]
            print(f"The signature is {'valid' if valid else 'invalid'}.")
            return valid
        except ClientError as err:
            if err.response["Error"]["Code"] == "SignatureDoesNotMatchException":
                print("The signature is not valid.")
            else:
                logger.error(
                    "Couldn't verify your signature. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            raise
```

- Untuk detail API, lihat Verifikasi di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
```

```
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
        " iv_message contains the original message
        " iv_signature contains the signature to verify
        " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
        oo_result = lo_kms->verify(
          iv_keyid = iv_key_id
          iv_message = iv_message
          iv_signature = iv_signature
          iv_signingalgorithm = iv_signing_algorithm
        ).
        DATA(lv_valid) = oo_result->get_signaturevalid( ).
        IF lv_valid = abap_true.
          MESSAGE 'Signature is valid.' TYPE 'I'.
        ELSE.
          MESSAGE 'Signature is invalid.' TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinvalidsigex.
        MESSAGE 'Invalid signature.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Verifikasi di AWS SDK untuk referensi API SAP ABAP.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Skenario untuk AWS KMS menggunakan AWS SDKs

Contoh kode berikut menunjukkan kepada Anda bagaimana menerapkan skenario umum AWS KMS dengan AWS SDKs. Skenario ini menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi di dalam AWS KMS atau dikombinasikan dengan yang lain Layanan AWS. Setiap skenario menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode.

Skenario menargetkan tingkat pengalaman menengah untuk membantu Anda memahami tindakan layanan dalam konteks.

Contoh

- Bekerja dengan enkripsi tabel DynamoDB menggunakan v2 AWS Command Line Interface

Bekerja dengan enkripsi tabel DynamoDB menggunakan v2 AWS Command Line Interface

Contoh kode berikut menunjukkan cara mengelola opsi enkripsi untuk tabel DynamoDB.

- Buat tabel dengan enkripsi default.
- Buat tabel dengan CMK yang dikelola pelanggan.
- Perbarui pengaturan enkripsi tabel.
- Jelaskan enkripsi tabel.

Bash

AWS CLI dengan skrip Bash

Buat tabel dengan enkripsi default.

```
# Create a table with default encryption (AWS owned key)
aws dynamodb create-table \
    --table-name CustomerData \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS
```

Buat tabel dengan CMK yang dikelola pelanggan.

```
# Step 1: Create a customer managed key in KMS
aws kms create-key \
    --description "Key for DynamoDB table encryption" \
```

```
    --key-usage ENCRYPT_DECRYPT \
    --customer-master-key-spec SYMMETRIC_DEFAULT

# Store the key ID for later use
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for
 DynamoDB')].KeyId" --output text)

# Step 2: Create a table with the customer managed key
aws dynamodb create-table \
    --table-name SensitiveData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

Perbarui enkripsi tabel.

```
# Update a table to use a different KMS key
aws dynamodb update-table \
    --table-name CustomerData \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

Jelaskan enkripsi tabel.

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
    --table-name CustomerData \
    --query "Table.SSEDescription"
```

- Untuk detail API, lihat topik berikut di Referensi Perintah AWS CLI .
  - CreateKey
  - CreateTable
  - DescribeTable
  - UpdateTable

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Dukungan pengesahan kriptografi di AWS KMS

AWS KMS mendukung pengesahan kriptografi untuk AWS Nitro Enclave dan NitroTPM.AWS Aplikasi yang mendukung metode pengesahan ini memanggil operasi AWS KMS kriptografi berikut dengan dokumen pengesahan yang ditandatangani. AWS KMS memverifikasi bahwa dokumen pengesahan berasal dari sumber yang valid (baik enklave Nitro atau NitroTPM). 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 instans enclave atau EC2.

- Dekripsi
- DeriveSharedSecret
- GenerateDataKey
- GenerateDataKeyPair
- GenerateRandom

Tabel berikut menunjukkan bagaimana respons terhadap permintaan yang dibuktikan berbeda dari respons standar untuk setiap operasi API.

AWS KMS operasi	Respon standar	Tanggapan untuk permintaan yang dibuktikan
`Decrypt`	Mengembalikan data plaintext	Mengembalikan data plaintext yang dienkripsi oleh kunci publik dari dokumen pengesahan
`DeriveSharedSecret`	Mengembalikan rahasia bersama mentah	Mengembalikan rahasia bersama mentah yang dienkripsi oleh kunci publik dari dokumen pengesahan
`GenerateDataKey`	Mengembalikan salinan plaintext dari kunci data	Mengembalikan salinan kunci data yang dienkripsi oleh kunci publik dari dokumen pengesahan

AWS KMS operasi	Respon standar	Tanggapan untuk permintaan yang dibuktikan
	(Juga mengembalikan salinan kunci data yang dienkripsi oleh kunci KMS)	(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 kunci publik dari dokumen pengesahan

AWS KMS mendukung kunci kondisi kebijakan yang dapat Anda gunakan untuk mengizinkan atau menolak operasi yang dibuktikan dengan AWS KMS kunci berdasarkan konten dokumen pengesahan. Anda juga dapat memantau permintaan yang dibuktikan ke AWS KMS dalam AWS CloudTrail log Anda.

Pelajari selengkapnya

- Pengesahan kriptografi
- AWS KMS kunci kondisi untuk platform yang dibuktikan
- Cara membuat panggilan yang dibuktikan ke AWS KMS
- Memantau permintaan yang dibuktikan

Cara membuat panggilan yang dibuktikan ke AWS KMS

Untuk membuat panggilan yang dibuktikan AWS KMS, gunakan `Recipient` parameter dalam permintaan untuk menyediakan dokumen pengesahan yang ditandatangani dan algoritma enkripsi untuk digunakan dengan kunci publik dalam dokumen pengesahan. 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 AWS Nitro Enclave atau NitroTPM. AWS AWS KMS bergantung pada tanda tangan digital untuk dokumen pengesahan untuk membuktikan bahwa kunci publik dalam permintaan berasal dari sumber yang valid. Anda tidak dapat menyediakan sertifikat Anda sendiri untuk menandatangani dokumen pengesahan secara digital.

AWS Nitro Enclave SDK, yang didukung hanya dalam enklave Nitro, secara otomatis menambahkan `Recipient` parameter dan nilainya ke setiap permintaan. AWS KMS

Untuk membuat permintaan yang dibuktikan di AWS SDKs, Anda harus menentukan `Recipient` parameter dan nilainya. Dokumen pengesahan dapat diambil dari NITROTPM menggunakan nitro-tpm-attest utilitas atau dari Nitro Secure Module (NSM) menggunakan NSM API.

AWS KMS mendukung kunci kondisi kebijakan yang dapat Anda gunakan untuk mengizinkan atau menolak operasi yang dibuktikan dengan AWS KMS kunci berdasarkan konten dokumen pengesahan. Anda juga dapat memantau permintaan yang dibuktikan ke AWS KMS dalam AWS CloudTrail log Anda.

Untuk informasi terperinci tentang `Recipient` parameter dan bidang `CiphertextForRecipient` respons AWS, lihat Dekripsi,,, DeriveSharedSecret, dan GenerateRandomtopik di Referensi AWS Key Management Service API GenerateDataKeyGenerateDataKeyPair, AWS Nitro Enclave SDK, atau SDK apa pun. AWS Untuk informasi tentang pengaturan data dan kunci data Anda untuk enkripsi, lihat Menggunakan pengesahan kriptografi dengan. AWS KMS

Memantau permintaan yang dibuktikan

Anda dapat menggunakan AWS CloudTrail log Anda untuk memantau Dekripsi,,, DeriveSharedSecret, GenerateDataKeyGenerateDataKeyPair, dan GenerateRandomoperasi yang menggunakan pengesahan. Dalam entri log ini, `additionalEventData` bidang memiliki `recipient` bidang dengan informasi dari dokumen pengesahan dalam permintaan. Bidang ini

disertakan hanya jika `Recipient` parameter dalam permintaan menentukan dokumen pengesahan yang ditandatangani.

Informasi spesifik yang disertakan dalam CloudTrail log tergantung pada metode pengesahan yang digunakan.

Memantau permintaan untuk kantong Nitro

Untuk pengesahan enclave Nitro, CloudTrail log menyertakan ID modul (`attestationDocumentModuleId`), image digest (), dan register konfigurasi platform (`attestationDocumentEnclaveImageDigest`) dari dokumen pengesahan. PCRs

ID modul adalah ID enclave dari enclave Nitro. Image digest adalah SHA384 hash dari gambar enclave. Anda dapat menggunakan intisari gambar dan nilai PCR dalam kondisi untuk kebijakan utama dan kebijakan IAM. Untuk informasi tentang PCRs, lihat Tempat mendapatkan pengukuran enclave di Panduan Pengguna AWS Nitro 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"
}
```

Memantau permintaan untuk NitroTPM

Untuk pengesahan NitroTPM, CloudTrail log menyertakan ID modul (`attestationDocumentModuleId`) dan register konfigurasi platform (PCRs) dari dokumen pengesahan.

ID modul adalah ID dari instans EC2 dengan NitroTPM dengan pengidentifikasi TPM. Anda dapat menggunakan nilai PCR dalam kondisi untuk kebijakan utama dan kebijakan IAM.

Bagian ini menunjukkan contoh entri CloudTrail log untuk setiap permintaan NitRotPM yang didukung. AWS KMS

Dekripsi (untuk NitroTPM)

Contoh berikut menunjukkan entri AWS CloudTrail log operasi Dekripsi untuk NitRotPM.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T22:58:24Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
```

```
            "attestationDocumentModuleId": "i-123456789abcde123-tpm0000000000000000",
            "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
            "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
            "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
            "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
            "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
        }
    },
    "requestID": "b4a65126-30d5-4b28-98b9-9153da559963",
    "eventID": "e5a2f202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey (untuk NitroTPM)

Contoh berikut menunjukkan entri AWS CloudTrail log GenerateDataKeyoperasi untuk NitroTPM.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
```

```
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "numberOfBytes": 32
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
           "attestationDocumentModuleId": "i-123456789abcde123-tpm0000000000000000",
           "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
           "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
           "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
           "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
           "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
           "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
        }
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyPair (untuk NitroTPM)

Contoh berikut menunjukkan entri AWS CloudTrail log GenerateDataKeyPairoperasi untuk NitroTPM.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
```

```
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
           "attestationDocumentModuleId": "i-123456789abcde123-tpm0000000000000000",
           "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
           "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
           "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
           "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
           "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
           "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
        }
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateRandom (untuk NitroTPM)

Contoh berikut menunjukkan entri AWS CloudTrail log GenerateRandomoperasi untuk NitroTPM.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateRandom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
           "attestationDocumentModuleId": "i-123456789abcde123-tpm0000000000000000",
           "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
           "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
           "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
           "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
           "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
           "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
        }
    },
    "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Penggunaan AWS KMS enkripsi dengan AWS layanan

Dengan AWS Key Management Service, Anda dapat memberikan kunci enkripsi untuk melindungi data di AWS layanan lain. Menggunakan AWS KMS enkripsi dengan AWS layanan mengacu pada proses integrasi AWS KMS dengan AWS layanan lain untuk mengenkripsi dan mendekripsi data saat istirahat atau dalam perjalanan. Pengembang, administrator sistem, dan profesional keamanan mungkin tertarik dengan topik ini untuk mengamankan data sensitif yang disimpan atau dikirimkan melalui AWS layanan, memenuhi persyaratan kepatuhan peraturan, atau menerapkan praktik terbaik enkripsi. Kasus penggunaan umum termasuk mengenkripsi volume Amazon EBS, bucket Amazon S3, dan database Amazon RDS. Bagian berikut akan mencakup langkah-langkah untuk mengonfigurasi dan mengelola kunci AWS KMS enkripsi untuk AWS layanan tertentu, memastikan kerahasiaan dan integritas data di seluruh AWS lingkungan Anda. Untuk daftar lengkap layanan yang terintegrasi dengannya AWS KMS, lihat Integrasi AWS Layanan.AWS

Topik berikut membahas secara rinci bagaimana layanan tertentu digunakan AWS 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**
>
> AWS layanan 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. Identifikasi berbagai jenis kunci

Topik

- Bagaimana Amazon Elastic Block Store (Amazon EBS) menggunakan AWS KMS
- Bagaimana Amazon EMR menggunakan AWS KMS
- Bagaimana Amazon Redshift menggunakan AWS KMS

Bagaimana Amazon Elastic Block Store (Amazon EBS) menggunakan AWS KMS

Topik ini membahas secara rinci bagaimana Amazon Elastic Block Store (Amazon EBS) digunakan AWS KMS untuk mengenkripsi volume dan snapshot. Untuk petunjuk dasar tentang mengenkripsi volume Amazon EBS, lihat Enkripsi Amazon EBS.

Topik

- Enkripsi EBS Amazon
- Menggunakan kunci KMS dan kunci data
- Konteks enkripsi Amazon EBS
- Mendeteksi kegagalan Amazon EBS
- Penggunaan AWS CloudFormation untuk membuat volume Amazon EBS terenkripsi

Enkripsi EBS Amazon

Saat Anda melampirkan volume Amazon EBS terenkripsi ke jenis instans Amazon Elastic Compute Cloud (Amazon EC2) yang didukung, data yang disimpan saat diam 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 Amazon EBS di Panduan Pengguna Amazon EBS.

Menggunakan kunci KMS dan kunci data

Saat Anda membuat volume Amazon EBS terenkripsi, Anda menentukan. AWS KMS key Secara default, Amazon EBS menggunakan Kunci yang dikelola AWS untuk 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 kepada Amazon EBS untuk menggunakan kunci KMS atas nama Anda. Untuk informasi selengkapnya, lihat Cara kerja enkripsi Amazon EBS di Panduan Pengguna Amazon EBS.

> **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. Identifikasi berbagai jenis kunci

Untuk setiap volume, Amazon EBS meminta AWS KMS untuk membuat kunci data unik yang dienkripsi di bawah kunci KMS yang Anda tentukan. Amazon EBS menyimpan kunci data terenkripsi dengan volume. Kemudian, saat Anda melampirkan volume ke instans Amazon EC2, Amazon EBS memanggil AWS KMS untuk mendekripsi kunci data. Amazon EBS menggunakan kunci data plaintext dalam memori hypervisor untuk mengenkripsi semua disk I/O ke volume. Untuk detailnya, lihat Cara kerja enkripsi EBS di Panduan Pengguna Amazon EC2 atau Panduan Pengguna Amazon EC2.

Konteks enkripsi Amazon EBS

Dalam permintaan GenerateDataKeyWithoutPlaintext dan 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. Saat Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS secara kriptografis mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda harus meneruskan konteks enkripsi yang sama.

Untuk semua volume dan snapshot terenkripsi yang dibuat dengan CreateSnapshot operasi Amazon EBS, Amazon EBS menggunakan ID volume sebagai nilai konteks enkripsi. Di `requestParameters` bidang entri CloudTrail log, konteks enkripsi terlihat mirip dengan berikut ini:

```
"encryptionContext": {
  "aws:ebs:id": "vol-0cfb133e847d28be9"
}
```

Untuk snapshot terenkripsi yang dibuat dengan CopySnapshot operasi Amazon EC2, Amazon EBS menggunakan ID snapshot sebagai nilai konteks enkripsi. Di `requestParameters` bidang entri CloudTrail log, konteks enkripsi terlihat mirip dengan berikut ini:

```
"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 kun cinya tidak—pembuatan volume atau `Enabled` lampiran volume gagal.

Dalam hal ini, Amazon EBS mengirimkan acara ke Amazon EventBridge (sebelumnya Ev CloudWatch ents) untuk memberi tahu Anda tentang kegagalan tersebut. Di EventBridge, Anda dapat menetapkan aturan yang memicu tindakan otomatis sebagai respons terhadap peristiwa ini. Untuk informasi selengkapnya, lihat CloudWatch Acara Amazon untuk Amazon EBS di Panduan Pengguna Amazon EBS, 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 di Konsol Manajemen AWS). Kemudian, lihat informasi di salah satu tautan berikut:

- Jika status kunci kunci KMS dinonaktifkan, aktifkan.

- Jika status kunci kunci KMS tertunda impor, impor materi kunci.
- Jika status kunci kunci KMS sedang menunggu penghapusan, batalkan penghapusan kunci.

Penggunaan AWS CloudFormation untuk membuat volume Amazon EBS terenkripsi

Anda dapat menggunakan AWS CloudFormation untuk membuat volume Amazon EBS terenkripsi. Untuk informasi selengkapnya, lihat AWS::EC2::Volume pada Panduan Pengguna AWS CloudFormation .

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 file AWS 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. Identifikasi berbagai jenis kunci

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 ke AWS 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 unik (kunci data) dan 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 dengan permintaan Dekripsi. AWS KMS Permintaan mencakup konteks enkripsi.
3. AWS KMS mendekripsi 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 ke AWS 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 unik (kunci data) dan kemudian mengirimkan dua salinan kunci data ini ke cluster. 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. Cluster mengekstrak kunci data terenkripsi dari objek terenkripsi, dan kemudian mengirimkan kunci data terenkripsi dengan permintaan Dekripsi. AWS KMS Permintaan mencakup konteks enkripsi.
4. AWS KMS mendekripsi 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 ke AWS 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 unik (kunci data) dan kemudian mengirimkan dua salinan kunci data ini ke node. 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 node reboot, node reboot mengirimkan kunci data terenkripsi dengan permintaan Dekripsi. AWS KMS
5. AWS KMS mendekripsi 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 otentikasi tambahan yang AWS KMS digunakan 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 Amazon Redshift menggunakan AWS KMS

Topik ini membahas bagaimana Amazon Redshift AWS KMS menggunakan 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 cluster. 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. Identifikasi berbagai jenis kunci

Konteks enkripsi

Setiap layanan yang terintegrasi dengan AWS KMS menentukan konteks enkripsi saat meminta kunci data, mengenkripsi, dan mendekripsi. Konteks enkripsi adalah data otentikasi tambahan (AAD) yang AWS KMS digunakan 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.

AWS KMS Referensi

Bahan 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 KMS harga. Daftar dan menjelaskan harga kunci KMS.

  Untuk menemukan: Berapa biaya untuk menggunakan kunci asimetris saya?

- AWS KMS permintaan 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 dalam setiap detik? Berapa banyak permintaan Dekripsi yang dapat saya jalankan pada kunci KMS di toko kunci khusus saya?

- AWS KMS kuota 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?

- AWS layanan terintegrasi dengan AWS KMS. Daftar AWS layanan yang menggunakan kunci KMS untuk melindungi sumber daya yang mereka buat, simpan, dan kelola.

  Untuk menemukan: Apakah Connect Customer menggunakan kunci KMS untuk melindungi sumber daya Connect saya?

Negara-negara kunci AWS KMS keys

AWS KMS key Selalu memiliki status kunci. Operasi pada kunci KMS dan lingkungannya dapat mengubah status kunci. Status kunci dapat berubah baik secara sementara, atau sampai operasi lain mengubah status kuncinya. Operasi ini dilakukan secara asinkron, atau dengan panggilan API.

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. Ketika materi kunci apa pun untuk kunci yang diimpor dihapus atau kedaluwarsa, maka status berubah dari `Enabled` ke`PendingImport`.
- `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`.
  - Multi-Region kunci berada dalam status `Updating` kunci sementara sementara 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`. Lihat perinciannya di Deleting multi-Region keys.

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 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]		N/A	 [14]	 [15]

API	Diaktifkan	Nonaktif	Penghapusan tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
DeriveSharedSecret	✓	⊗ [1]	⊗ [2] atau [3]	⊗ [5]	N/A	⊗ [14]	✓
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]

API	Diaktifkan	Nonaktif	Penghapusan tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
Enkripsi	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [11]	✗ [14]	✓
GenerateDataKey	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [11]	✗ [14]	✓
GenerateDataKeyPair	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [7]	✗ [14]	✓
GenerateDataKeyPairWithoutPlaintext	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [7]	✗ [14]	✓
GenerateDataKeyWithoutPlaintext	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [11]	✗ [14]	✓

API	Diaktifkan	Nonaktif	Penghapusan tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
GenerateMac	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	N/A	✗ [14]	✓
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓
GetKeyRotationStatus	? [7]	? [7]	? [7]	✗ [6]	✗ [7]	? [7]	? [7]
GetParametersForImport	? [9]	? [9]	✗ [8] atau [9]	✓	✗ [9]	✗ [14]	✗ [15]
GetPublicKey	✓	✓	✗ [2] atau [3]	✓	N/A	✗ [14]	✓
ImportKeyMaterial	? [9]	? [9]	✗ [9]	✓	✗ [9]	✗ [14]	✓

API	Diaktifkan	Nonaktif	Penghapusan tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
ListAliases	✓	✓	✓	✓	✓	✓	✓
ListGrants	✓	✓	✓	✓	✓	✓	✓
ListKeyPolicies	✓	✓	✓	✓	✓	✓	✓
ListKeyRotations	? [7]	? [7]	? [7]	✓	⊗ [7]	? [7]	? [7]
ListResourceTags	✓	✓	✓	✓	✓	✓	✓
PutKeyPolicy	✓	✓	✓	✓	✓	✓	✓
ReEncrypt	✓	⊗ [1]	⊗ [2] atau [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓

API	Diaktifkan	Nonaktif	Penghapusan tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
ReplicateKey	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	N/A	✗ [14]	✗ [15]
RetireGrant	✓	✓	✓	✓	✓	✓	✓
RevokeGrant	✓	✓	✓	✓	✓	✓	✓
RotateKeyOnDemand	? [7]	✗ [1] atau [7]	✗ [3] atau [7]	✗ [5]	✗ [7]	✗ [14]	? [7]
ScheduleKeyDeletion	✓	✓	✗ [3]	✓	✓	✓	✗ [15]
Sign	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	N/A	✗ [14]	✓

API	Diaktifkan	Nonaktif	Penghapusan tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
TagResource	✓	✓	⊗ [3]	✓	✓	✓	✓
UntagResource	✓	✓	⊗ [3]	✓	✓	✓	✓
UpdateAlias	✓	✓	? [10]	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	⊗ [3]	✓	✓	✓	✓
UpdatePrimaryRegion	✓	⊗ [1]	⊗ [2] atau [3]	⊗ [5]	N/A	⊗ [14]	✓
Verifikasi	✓	⊗ [1]	⊗ [2] atau [3]	⊗ [5]	N/A	⊗ [14]	✓

API	Diaktifkan	Nonaktif	Penghapusan tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
VerifyMac	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	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 because no key material has ever been imported or one of the imported key materials is deleted or expired.`
- [6] `UnsupportedOperationException: <key ARN> origin is EXTERNAL which is not valid for this operation.`
- [7] Jika kunci KMS ada di toko kunci khusus:`UnsupportedOperationException`.
- [8] Jika kunci KMS telah mengimpor bahan kunci: `KMSInvalidStateException`
- [9] Jika kunci KMS tidak dapat mengimpor bahan kunci:`UnsupportedOperationException`.

- [10] Jika kunci KMS sumber sedang menunggu 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`.

Referensi tipe kunci

AWS KMS mendukung 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.

AWS KMS Operasi API	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYP DECRYPT	Tombol KMS asimetris (SIGN_VERIFY)	Tombol KMS asimetris (KEY_AGREEMENT)
CancelKeyDeletion	Ya	Ya	Ya	Ya	Ya
CreateAlias	Ya	Ya	Ya	Ya	Ya
CreateGrant	Ya	Ya	Ya	Ya	Ya
CreateKey	Ya	Ya	Ya	Ya	Ya
Dekripsi	Ya	Tidak	Ya	Tidak	Tidak
DeleteAlias	Ya	Ya	Ya	Ya	Ya
DeleteImportedKeyMaterial Hanya berlaku pada kunci KMS dengan bahan kunci impor (`Origin` is `EXTERNAL`)	Ya	Ya	Ya	Ya	Ya
DeriveSharedSecret	Tidak	Tidak	Tidak	Tidak	Ya
DescribeKey	Ya	Ya	Ya	Ya	Ya
DisableKey	Ya	Ya	Ya	Ya	Ya
DisableKeyRotation	Ya Hanya berlaku pada kunci	Tidak	Tidak	Tidak	Tidak

AWS KMS Operasi API	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYP DECRYPT	Tombol KMS asimetris (SIGN_VERIFY)	Tombol KMS asimetris (KEY_AGREEMENT)
	KMS dengan materi AWS KMS kunci (`Originis`				
EnableKey	Ya	Ya	Ya	Ya	Ya
EnableKeyRotation	Ya Hanya berlaku pada kunci KMS dengan materi AWS KMS kunci (`Originis`	Tidak	Tidak	Tidak	Tidak
Enkripsi	Ya	Tidak	Ya	Tidak	Tidak
GenerateDataKey	Ya	Tidak	Tidak	Tidak	Tidak

AWS KMS Operasi API	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYP DECRYPT	Tombol KMS asimetris (SIGN_VERIFY)	Tombol KMS asimetris (KEY_AGREEMENT)
GenerateDataKeyPair Menghasilkan data key pair asimetris yang dilindungi oleh kunci KMS enkripsi simetris.	Ya Tidak berlaku pada kunci KMS di toko kunci khusus.	Tidak	Tidak	Tidak	Tidak
GenerateDataKeyPairWithoutPlaintext Menghasilkan data key pair asimetris yang dilindungi oleh kunci KMS enkripsi simetris.	Ya Tidak berlaku pada kunci KMS di toko kunci khusus.	Tidak	Tidak	Tidak	Tidak
GenerateDataKeyWithoutPlaintext	Ya	Tidak	Tidak	Tidak	Tidak
GenerateMac	Tidak	Ya	Tidak	Tidak	Tidak
GetKeyPolicy	Ya	Ya	Ya	Ya	Ya

AWS KMS Operasi API	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYP DECRYPT	Tombol KMS asimetris (SIGN_VERIFY)	Tombol KMS asimetris (KEY_AGREEMENT)
GetKeyRotationStatus	Ya	Ya (KeyRotaonEnabl akan selalu berupa false.)	Ya (KeyRotaonEnabl akan selalu berupa false.)	Ya (KeyRotationEnabled akan selalu berupa false.)	Ya (KeyRotationEnabled akan selalu berupa false.)
GetParametersForImport Hanya berlaku pada kunci KMS dengan bahan kunci impor (OriginisEXTERNAL	Ya	Ya	Ya	Ya	Ya
GetPublicKey	Tidak	Tidak	Ya	Ya	Ya
ImportKeyMaterial Hanya berlaku pada kunci KMS dengan bahan kunci impor (OriginisEXTERNAL	Ya	Ya	Ya	Ya	Ya
ListAliases	Ya	Ya	Ya	Ya	Ya
ListGrants	Ya	Ya	Ya	Ya	Ya
ListKeyPolicies	Ya	Ya	Ya	Ya	Ya

AWS KMS Operasi API	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYP DECRYPT	Tombol KMS asimetris (SIGN_VERIFY)	Tombol KMS asimetris (KEY_AGREEMENT)
ListKeyRotations	Ya	Ya (`Rotatic` Bidang akan selalu nol atau kosong.)	Ya (`Rotatic` Bidang akan selalu nol atau kosong.)	Ya (`Rotations` Bidang akan selalu nol atau kosong.)	Ya (`Rotations` Bidang akan selalu nol atau kosong.)
ListResourceTags	Ya	Ya	Ya	Ya	Ya
ListRetirableGrants	Ya	Ya	Ya	Ya	Ya
PutKeyPolicy	Ya	Ya	Ya	Ya	Ya
ReEncrypt	Ya	Tidak	Ya	Tidak	Tidak
ReplicateKey - Hanya berlaku pada kunci Multi-wilayah	Ya	Ya	Ya	Ya	Ya
RetireGrant	Ya	Ya	Ya	Ya	Ya
RevokeGrant	Ya	Ya	Ya	Ya	Ya

AWS KMS Operasi API	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYP DECRYPT	Tombol KMS asimetris (SIGN_VERIFY)	Tombol KMS asimetris (KEY_AGREEMENT)
RotateKeyOnDemand	Ya Hanya berlaku pada kunci KMS enkripsi simetris yang dikelola pelanggar dengan atau asal. AWS_KMS EXTERNAI	Tidak	Tidak	Tidak	Tidak
ScheduleKeyDeletion	Ya	Ya	Ya	Ya	Ya
Tanda	Tidak	Tidak	Tidak	Ya	Tidak
TagResource	Ya	Ya	Ya	Ya	Ya
UntagResource	Ya	Ya	Ya	Ya	Ya

AWS KMS Operasi API	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYP DECRYPT	Tombol KMS asimetris (SIGN_VERIFY)	Tombol KMS asimetris (KEY_AGREEMENT)
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.	Ya	Ya	Ya	Ya	Ya
UpdateKey Description	Ya	Ya	Ya	Ya	Ya
UpdateReplicaRegion - Hanya berlaku pada kunci Multi-wilayah	Ya	Ya	Ya	Ya	Ya
Verifikasi	Tidak	Tidak	Tidak	Ya	Tidak
VerifyMac	Tidak	Ya	Tidak	Tidak	Tidak

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.
  - Rotasi kunci sesuai permintaan (`RotateKeyOnDemand`) didukung untuk kunci KMS enkripsi simetris dengan bahan kunci yang diimpor.
- 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.

AWS KMS Operasi API	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
CancelKeyDeletion	✓	✓	✓

AWS KMS Operasi API	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
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	✓ Valid hanya `KeyUsage` jika `ENCRYPT_DECRYPT`	✓	✓
DeleteAlias	✓	✓	✓
DeleteImportedKeyMaterial	✓ Hanya berlaku untuk kunci dengan bahan kunci impor (`OriginisEXTERN`	✓	⊗

AWS KMS Operasi API	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
DeriveSharedSecret	✓ Hanya berlaku bila KeyUsage adaKEY_AGREEMENT)	✓ Hanya berlaku bila KeyUsage adaKEY_AGREEMENT)	⊗
DescribeKey	✓	✓	✓
DisableKey	✓	✓	✓
DisableKeyRotation	✓ Hanya berlaku pada kunci enkripsi simetris dengan material AWS KMS kunci (OriginisAWS_KM	⊗	⊗
EnableKey	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	✓

AWS KMS Operasi API	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
EnableKeyRotation	✓ Hanya berlaku pada kunci enkripsi simetris dengan material AWS KMS kunci (`OriginisAWS_KM`	⊗	⊗
Enkripsi	✓ Valid hanya `KeyUsage` jika `ENCRYPT_DECRYPT`	✓	✓
GenerateDataKey	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	✓
GenerateDataKeyPair	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	⊗

AWS KMS Operasi API	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
GenerateDataKeyPairWithoutPlaintext	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	⊗
GenerateDataKeyWithoutPlaintext	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	✓
GenerateMac Hanya berlaku pada kunci HMAC KMS	✓	✓	⊗
GetKeyPolicy	✓	✓	✓
GetKeyRotationStatus	✓	✓ (`KeyRotationEnabled` akan selalu berupa `false`.)	⊗

AWS KMS Operasi API	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
GetParametersForImport	✓ Hanya berlaku untuk kunci dengan bahan kunci impor (`OriginisEXTERN`	✓	⊗
GetPublicKey Hanya berlaku untuk kunci KMS asimetris.	✓	✓	⊗
ImportKeyMaterial	✓ Hanya berlaku untuk kunci dengan bahan kunci impor (`OriginisEXTERN`	✓	⊗
ListAliases	✓	✓	✓
ListGrants	✓	✓	✓
ListKeyPolicies	✓	✓	✓

AWS KMS Operasi API	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
ListKeyRotations	✓ Hanya berlaku pada kunci enkripsi simetris dengan `AWS_KMS` atau `EXTERNAL` asal.	✓ Hanya berlaku pada kunci enkripsi simetris.	✓
ListResourceTags	✓	✓	✓
ListRetirableGrants	✓	✓	✓
PutKeyPolicy	✓	✓	✓
ReEncrypt	✓ Valid hanya `KeyUsage` jika `ENCRYPT_DECRYPT`	✓	✓
ReplicateKey	✓ Hanya berlaku pada kunci utama Multi-wilayah.	✓ Hanya berlaku pada kunci utama Multi-wilayah.	⊗
RetireGrant	✓	✓	✓

AWS KMS Operasi API	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
RevokeGrant	✓	✓	✓
RotateKeyOnDemand	✓ Hanya berlaku pada kunci enkripsi simetris dengan `AWS_KMS` atau `EXTERNAL` asal.	✓ Hanya berlaku pada kunci enkripsi simetris.	⊗
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.	✓	✓	✓
UpdateKeyDescription	✓	✓	✓

AWS KMS Operasi API	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
UpdateReplicaRegion	✓	✓ Hanya berlaku pada kunci Multi-wilayah.	⊗
Verifikasi Hanya berlaku bila `KeyUsage` ada`SIGN_VERIFY` .	✓	✓	⊗
VerifyMac Hanya berlaku pada kunci HMAC KMS	✓	✓	⊗

Referensi spesifikasi kunci

Saat Anda membuat kunci KMS asimetris atau kunci HMAC KMS, Anda memilih spesifikasi kuncinya. Spesifikasi kunci, yang merupakan properti dari setiap AWS 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 Service Harga. Untuk informasi selengkapnya tentang kuota permintaan, lihat Kuota permintaan.

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`

AWS KMS mendukung spesifikasi kunci berikut untuk kunci KMS:

Spesifikasi kunci enkripsi simetris (default)

- SYMMETRIC_DEFAULT

Spesifikasi kunci RSA (enkripsi dan dekripsi -atau- penandatanganan dan verifikasi)

- RSA_2048
- RSA_3072
- RSA_4096

Spesifikasi kunci kurva eliptik

- Pasangan kunci kurva elips standar Nist asimetris (penandatanganan dan verifikasi -atau- memperoleh rahasia bersama)
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
  - ECC_NIST_ EDWARDS25519 (ed25519) - hanya penandatanganan dan verifikasi
    - Catatan: Untuk kunci ECC_NIST_ EDWARDS25519 KMS, algoritme penandatanganan _SHA_512 membutuhkan, sedangkan ED25519 _PH_SHA_512 membutuhkan. `MessageType:RAW` `ED25519MessageType:DIGEST` Jenis pesan ini tidak dapat digunakan secara bergantian.
- Pasangan kunci kurva elips asimetri lain (penandatanganan dan verifikasi)
  - ECC_SECG_P256K1 (secp256k1), biasa digunakan untuk mata uang kripto.

SM2 spesifikasi kunci (enkripsi dan dekripsi -atau- penandatanganan dan verifikasi -atau- memperoleh rahasia bersama)

- SM2 (Hanya Wilayah Tiongkok)

Spesifikasi kunci HMAC

- HMAC_224
- HMAC_256
- HMAC_384
- HMAC_512

Spesifikasi kunci ML-DSA

- ML_DSA_44
- ML_DSA_65
- ML_DSA_87

Spesifikasi kunci SYMMETRIC_DEFAULT

Spesifikasi kunci default, SYMMETRIC_DEFAULT, adalah spesifikasi kunci untuk kunci KMS enkripsi simetris. Ketika Anda memilih jenis kunci simetris dan Enkripsi dan dekripsi penggunaan kunci di AWS KMS konsol, itu 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 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.

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 membuat serangan brute force pada AWS KMS ciphertext tidak layak dilakukan.

Satu-satunya pengecualian di Wilayah Tiongkok, di mana SYMMETRIC_DEFAULT mewakili kunci simetris 128-bit yang menggunakan enkripsi. SM4 Anda hanya dapat membuat SM4 kunci 128-bit di Wilayah Tiongkok. Anda tidak dapat membuat kunci KMS AES-GCM 256-bit di Wilayah Tiongkok.

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. AWS layanan 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.

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.

Spesifikasi kunci RSA

Saat Anda menggunakan spesifikasi kunci RSA, AWS KMS buat kunci KMS asimetris dengan key pair RSA. Kunci pribadi tidak pernah meninggalkan yang tidak AWS KMS terenkripsi. Anda dapat menggunakan kunci publik di dalam AWS KMS, atau mengunduh kunci publik untuk digunakan di luar AWS KMS.

> **Warning**
>
> Saat Anda mengenkripsi data di luar AWS KMS, pastikan Anda dapat mendekripsi ciphertext Anda. Jika Anda menggunakan kunci publik dari kunci KMS yang telah dihapus AWS 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.

Di AWS 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 Key usage, ditentukan secara terpisah dari spesifikasi kunci, tetapi Anda harus membuat keputusan itu 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 AWS KMS harga, lihat Harga Layanan Manajemen AWS Utama. 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 KMS kemudian 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 KMS mendukung 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. Dalam AWS KMS, algoritma RSAES_OAEP selalu menggunakan fungsi hash yang sama untuk tujuan hashing dan untuk fungsi pembuatan topeng (). 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 dalam fungsi pembuatan MGF1 topeng bersama dengan label kosong.
RSAES_OAEP_SHA_256	PKCS #1, Bagian 7.1. Enkripsi RSA dengan OAEP Padding menggunakan SHA-256 untuk hash dan dalam fungsi pembuatan MGF1 topeng bersama dengan 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 KMS juga 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 plaintext maksimum dalam byte untuk algoritma 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 KMS kemudian 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 KMS mendukung 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- PKCS1 -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 intisari pesan dan fungsi pembuatan topeng bersama dengan garam 256-bit MGF1

Algoritme penandatanganan	Deskripsi algoritme
RSASSA_PSS_SHA_384	PKCS #1 v2.2, Bagian 8.1, tanda tangan RSA dengan padding PSS menggunakan SHA-384 untuk intisari pesan dan fungsi pembuatan topeng bersama dengan garam 384-bit MGF1
RSASSA_PSS_SHA_512	PKCS #1 v2.2, Bagian 8.1, tanda tangan RSA dengan padding PSS menggunakan SHA-512 untuk intisari pesan dan fungsi pembuatan topeng bersama dengan garam 512-bit MGF1
RSASSA_ _V1_5_SHA_256 PKCS1	PKCS #1 v2.2, Bagian 8.2, tanda tangan RSA dengan PKCS #1v1.5 Padding dan SHA-256
RSASSA_ _V1_5_SHA_384 PKCS1	PKCS #1 v2.2, Bagian 8.2, tanda tangan RSA dengan PKCS #1v1.5 Padding dan SHA-384
RSASSA_ _V1_5_SHA_512 PKCS1	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 KMS juga 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 kurva eliptik (ECC), AWS KMS buat kunci KMS asimetris dengan key pair ECC untuk penandatanganan dan verifikasi atau menurunkan rahasia bersama (tetapi tidak keduanya). Kunci pribadi yang menghasilkan tanda tangan atau memperoleh rahasia bersama tidak pernah meninggalkan yang tidak terenkripsi. AWS KMS Anda dapat menggunakan

kunci publik untuk memverifikasi tanda tangan di dalamnya AWS KMS, atau mengunduh kunci publik untuk digunakan di luar. AWS KMS

Saat Anda menggunakan spesifikasi kunci Edwards Curve, AWS KMS buat kunci KMS asimetris dengan key pair Ed25519 untuk penandatanganan dan verifikasi. Kunci pribadi yang menghasilkan tanda tangan tidak pernah meninggalkan AWS KMS yang tidak terenkripsi. Anda dapat menggunakan kunci publik untuk memverifikasi tanda tangan di dalamnya AWS KMS, atau mengunduh kunci publik untuk digunakan di luar. AWS KMS

AWS KMS mendukung spesifikasi kunci ECC berikut untuk kunci KMS asimetris.

- Pasangan kunci kurva elips standar Nist asimetris (penandatanganan dan verifikasi -atau- memperoleh rahasia bersama)
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
  - ECC_NIST_ EDWARDS25519 (ed25519) - hanya penandatanganan dan verifikasi
    - Catatan: Untuk kunci ECC_NIST_ EDWARDS25519 KMS, algoritme penandatanganan _SHA_512 membutuhkan, sedangkan ED25519 _PH_SHA_512 membutuhkan. `MessageType:RAW` ED25519`MessageType:DIGEST` Jenis pesan ini tidak dapat digunakan secara bergantian.
- 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 mendapatkan rahasia bersama, gunakan salah satu spesifikasi kunci kurva elips standar NIST (kecuali ECC_SECG_P256K1 dan ECC_NIST_). EDWARDS25519 Satu-satunya algoritma perjanjian kunci yang didukung untuk memperoleh rahasia bersama adalah Elliptic Curve Cryptography Cofactor Diffie-Hellman Primitive (ECDH). Untuk contoh cara mendapatkan rahasia bersama secara offline, lihatthe section called “Memperoleh rahasia bersama secara offline”.

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.

Tabel berikut menunjukkan algoritma penandatanganan yang AWS KMS mendukung 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, Section 6.4, ECDSA signature using the curve specified by the key and SHA-256 untuk intisari pesan.
ECC_NIST_P384	ECDSA_SHA_384	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-384 untuk intisari pesan.
ECC_NIST_P521	ECDSA_SHA_512	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-512 untuk intisari pesan.
ECC_SECG_P256K1	ECDSA_SHA_256	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-256 untuk intisari pesan.
ECC_NIST_ EDWARDS25519	ED25519_SHA_512	NIST FIPS 186-5, Section 7.6, EdDSA signature using the curve specified by the key and SHA-512 untuk intisari pesan. KMS membutuhkan

Spesifikasi kunci	Algoritme penandatanganan	Deskripsi algoritme
		`MessageType:RAW` dengan algoritma ini.
ECC_NIST_ EDWARDS25519	ED25519_PH_SHA_512	NIST FIPS 186-5, Section 7.8, HashEdDSA signature using the curve specified by the key and SHA-512 untuk intisari pesan. KMS membutuhkan `MessageType:DIGEST` _dengan algoritma ini. Saat Anda menentukan algoritma penandatanganan ED25519 _PH_SHA_5 12 dengan`MessageTy pe:DIGEST` , AWS KMS masih melakukan prehash SHA-512 yang dijelaskan dalam Langkah 1 dari Bagian 7.8.1. Ini berarti input di-hash dua kali: sekali oleh Anda dan sekali oleh KMS.

Spesifikasi utama untuk kunci HMAC KMS

AWS KMS mendukung 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

Spesifikasi kunci HMAC	Algoritma MAC
HMAC_384	HMAC_SHA_384
HMAC_512	HMAC_SHA_512

Spesifikasi kunci ML-DSA

Kunci ML-DSA adalah kunci kriptografi yang digunakan dalam Module-Lattice-Based Digital Signature Algorithm (ML-DSA), yang dirancang untuk kriptografi pasca-kuantum. Algoritma ini merupakan bagian dari upaya standardisasi NIST (National Institute of Standards and Technology), yang secara khusus diuraikan dalam Federal Information Processing Standards (FIPS) 204.

Kunci ML-DSA digunakan dalam sistem key pair publik-pribadi. Kunci pribadi digunakan untuk menandatangani data, sedangkan kunci publik digunakan untuk memverifikasi tanda tangan. Sistem ini memastikan keaslian, integritas, dan non-penolakan pesan digital atau dokumen, bahkan dalam menghadapi potensi ancaman komputer kuantum.

Saat Anda membuat kunci dengan spesifikasi kunci ML-DSA, AWS KMS buat kunci KMS asimetris dengan key pair ML-DSA untuk penandatanganan dan verifikasi. Kunci pribadi yang menghasilkan tanda tangan tidak pernah meninggalkan AWS KMS yang tidak terenkripsi. Anda dapat menggunakan kunci publik untuk memverifikasi tanda tangan di dalamnya AWS KMS, atau mengunduh kunci publik untuk digunakan di luar. AWS KMS

AWS KMS mendukung spesifikasi kunci ML-DSA berikut untuk kunci KMS asimetris:

- ML_DSA_44
- ML_DSA_65
- ML_DSA_87

AWS KMS mendukung algoritma penandatanganan ML_DSA_SHAKE_256 untuk semua spesifikasi kunci ML-DSA.

SM2 spesifikasi kunci (hanya Wilayah Tiongkok)

Spesifikasi utama adalah spesifikasi SM2 kunci kurva elips yang ditentukan dalam GM/T serangkaian spesifikasi yang diterbitkan oleh Kantor Administrasi Kriptografi Komersil Negara China (OSCCA). Spesifikasi SM2 kunci hanya tersedia di Wilayah Tiongkok. Saat Anda menggunakan spesifikasi

SM2 kunci, AWS KMS buat kunci KMS asimetris dengan key pair SM2 . Anda dapat menggunakan SM2 key pair di dalam AWS KMS, atau mengunduh kunci publik untuk digunakan di luar AWS KMS. Untuk informasi selengkapnya, lihat the section called “Verifikasi offline dengan pasangan kunci SM2 (hanya Wilayah China)”.

Setiap kunci KMS hanya dapat memiliki satuKey usage. Anda dapat menggunakan kunci SM2 KMS untuk penandatanganan dan verifikasi, enkripsi dan dekripsi, atau memperoleh rahasia bersama. Anda harus menentukan penggunaan kunci saat Anda membuat kunci KMS, dan Anda tidak dapat mengubahnya setelah kunci dibuat.

Jika Anda membuat kunci KMS asimetris untuk mendapatkan rahasia bersama, gunakan spesifikasi kunci. SM2 Satu-satunya algoritma perjanjian kunci yang didukung untuk memperoleh rahasia bersama adalah Elliptic Curve Cryptography Cofactor Diffie-Hellman Primitive (ECDH).

AWS KMS mendukung SM2 enkripsi dan algoritma penandatanganan berikut:

- SM2Algoritma enkripsi PKE

  SM2PKE adalah algoritma enkripsi berbasis kurva elips yang didefinisikan oleh OSCCA pada 0003.4-2012. GM/T
- SM2Algoritma penandatanganan DSA

  SM2DSA adalah algoritma penandatanganan berbasis kurva elips yang didefinisikan oleh OSCCA pada 0003.2-2012. GM/T SM2DSA memerlukan ID pembeda yang di-hash dengan algoritma SM3 hashing dan kemudian dikombinasikan dengan pesan, atau intisari pesan, yang Anda kirimkan. AWS KMS Nilai gabungan ini kemudian di-hash dan ditandatangani oleh. AWS KMS

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 memperbaiki setiap izin.

Untuk informasi lebih lanjut tentang AWS KMS operasi mana yang valid untuk kunci KMS enkripsi simetris, kunci KMS asimetris, dan kunci KMS HMAC, lihat. Referensi tipe kunci

> **Note**
>
> Anda mungkin harus menggulir horizontal atau vertikal untuk melihat semua data di dalam tabel.

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	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 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	Tipe kebijakan	Cross-account menggunakan	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) Lihat perinciannya di Mengontrol akses ke alias.	Kebijakan IAM (untuk alias) Kebijakan kunci (untuk kunci KMS)	Tidak	Alias Kunci KMS	Tidak ada (ketika mengontrol akses ke alias) 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	Tipe kebijakan	Cross-account menggunakan	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: GrantConstraintSourceArn km: GrantConstraintType km: GranteePrincipal km: GranteeServicePrincipal km: GrantIsForAWSResource km: GrantOperations km: RetiringPrincipal km: RetiringServicePrincipal Ketentuan untuk operasi kunci KMS: km: CallerAccount

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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) Lihat perinciannya di 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	Tipe kebijakan	Cross-account menggunakan	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

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
DeriveSharedSecret `kms:DeriveSharedSecret`	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: KeyAgreementAlgorithm
DescribeCustomKeyStores `kms:DescribeCustomKeyStores`	Kebijakan IAM	Tidak	*	km: CallerAccount

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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 Kondisi penggunaan terakhir utama: km: TrailingDaysWithoutKeyUsage

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
EnableKeyRotation `kms:EnableKeyRotation`	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 rotasi kunci otomatis: km: RotationPeriodInDays

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ViaService

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GenerateDataKey `kms:GenerateDataKey`	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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ViaService

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GenerateDataKeyPair `kms:GenerateDataKeyPair`	Kebijakan kunci	Ya	Kunci KMS 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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GenerateDataKeyPairWithoutPlaintext `kms:GenerateDataKeyPairWithoutPlaintext`	Kebijakan kunci	Ya	Kunci KMS 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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GenerateDataKeyWithoutPlaintext `kms:GenerateDataKeyWithoutPlaintext`	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GetKeyLastUsage `kms:GetKeyLastUsage`	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GetKeyRotationStatus `kms:GetKeyRotationStatus`	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

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GetPublicKey `kms:GetPublicKey`	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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: GrantIsFor AWSResource

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
ListKeyRotations `kms:ListKeyRotations`	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
ListKeys `kms:ListKeys`	Kebijakan IAM	Tidak	*	Tidak ada

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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	Kondisi konteks enkripsi: kms:EncryptionContext: kunci-konteks km: EncryptionContextKeys Ketentuan hibah: km: GrantConstraintType Ketentuan untuk operasi kunci KMS: km: CallerAccount km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
RotateKeyOnDemand `kms:RotateKeyOnDemand`	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	Tipe kebijakan	Cross-account menggunakan	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 Kondisi penggunaan terakhir utama: km: TrailingDaysWithoutKeyUsage

Tindakan dan izin	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
Tanda `kms:Sign`	Kebijakan kunci	Ya	Kunci KMS	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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 Lihat perinciannya di 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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	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	Tipe kebijakan	Cross-account menggunakan	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
Verifikasi `kms:Verify`	Kebijakan kunci	Ya	Kunci KMS	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	Tipe kebijakan	Cross-account menggunakan	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.
- Cross-account penggunaan 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 “Temukan ID kunci dan kunci ARN”.

  `arn:`*`AWS_partition_name`*`:kms:`*`AWS_Region`*`:`*`AWS_account_ID`*`:key/`*`key_ID`*

  Contoh:

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

Alias

Ketika sumber daya adalah alias, gunakan ARN alias. Untuk bantuan, lihat the section called "Temukan nama alias dan alias ARN".

`arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:alias/alias_name`

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 `kms:CreateKey` atau `kms:ListKeys` mengizinkan, Anda harus mengatur `Resource` elemen ke`*`.

- 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.

AWS KMS operasi internal

AWS Key Management Service (AWS KMS) menyediakan kunci kriptografi dan operasi yang diamankan oleh FIPS 140-3 Security Level 3 divalidasi modul keamanan perangkat keras (HSM) yang diskalakan untuk cloud. AWS KMS kunci dan fungsionalitas digunakan oleh beberapa layanan AWS cloud, dan Anda dapat menggunakannya untuk melindungi data dalam aplikasi Anda. Panduan teknis ini memberikan rincian tentang operasi kriptografi yang dijalankan di dalamnya AWS saat Anda menggunakannya AWS KMS.

AWS KMS internal diperlukan untuk skala dan keamanan HSMs untuk layanan manajemen kunci yang didistribusikan secara global.

Topik

- Domain dan status domain
- Keamanan komunikasi internal
- Proses replikasi untuk kunci Multi-wilayah
- Perlindungan daya tahan

Domain dan status domain

Kumpulan kooperatif AWS KMS entitas internal tepercaya dalam suatu domain Wilayah AWS disebut sebagai domain. Domain mencakup seperangkat entitas tepercaya, seperangkat aturan, dan seperangkat kunci rahasia, yang disebut kunci domain. Kunci domain dibagi di antara HSMs yang merupakan anggota domain. Status domain terdiri atas bidang berikut.

Nama

Nama domain untuk mengidentifikasi domain ini.

Anggota

Daftar HSMs yang merupakan anggota domain, termasuk kunci penandatanganan publik dan kunci perjanjian publik mereka.

Operator

Daftar entitas, kunci penandatanganan publik, dan peran (AWS KMS operator atau host layanan) yang mewakili operator layanan ini.

Aturan

Daftar aturan kuorum untuk setiap perintah yang harus dipenuhi untuk menjalankan perintah pada HSM.

Kunci domain

Daftar kunci domain (kunci simetris) saat ini digunakan dalam domain.

Status domain penuh tersedia hanya pada HSM. Status domain disinkronkan antara anggota domain HSM sebagai token domain yang diekspor.

Kunci domain

Semua HSMs dalam domain berbagi satu set kunci domain, $\{DK_r\}$. Kunci ini dibagi melalui rutin ekspor status domain. Status domain yang diekspor dapat diimpor ke setiap HSM yang merupakan anggota domain.

Set kunci domain, $\{DK_r\}$, selalu menyertakan satu kunci domain aktif, dan beberapa kunci domain yang dinonaktifkan. Kunci domain diputar setiap hari untuk memastikan bahwa AWS sesuai dengan Rekomendasi untuk Manajemen Kunci - Bagian 1. Selama rotasi kunci domain, semua kunci KMS yang ada yang dienkripsi di bawah kunci domain keluar dienkripsi ulang di bawah kunci domain aktif yang baru. Kunci domain aktif digunakan untuk mengenkripsi yang baru EKTs. Kunci domain kedaluwarsa hanya dapat digunakan untuk mendekripsi yang sebelumnya dienkripsi EKTs selama beberapa hari yang setara dengan jumlah kunci domain yang baru saja diputar.

Token domain yang diekspor

Ada kebutuhan rutin untuk menyinkronkan status di antara peserta domain. Hal ini dilakukan melalui pengeksporan status domain setiap kali perubahan dibuat ke domain. Status domain diekspor sebagai token domain yang diekspor.

Nama

Nama domain untuk mengidentifikasi domain ini.

Anggota

Daftar HSMs yang merupakan anggota domain, termasuk kunci publik penandatanganan dan perjanjian mereka.

Operator

Daftar entitas, kunci penandatanganan publik, dan peran yang mewakili operator layanan ini.

Aturan

Daftar aturan kuorum untuk setiap perintah yang harus dipenuhi untuk menjalankan perintah pada anggota domain HSM.

Kunci domain terenkripsi

Kunci domain terenkripsi envelope. Kunci domain dienkripsi oleh anggota penandatanganan untuk tiap-tiap anggota yang tercantum di atas, disampul ke kunci perjanjian publik mereka.

Tanda tangan

Tanda tangan pada status domain yang dihasilkan oleh HSM, tentu anggota domain yang diekspor status domain.

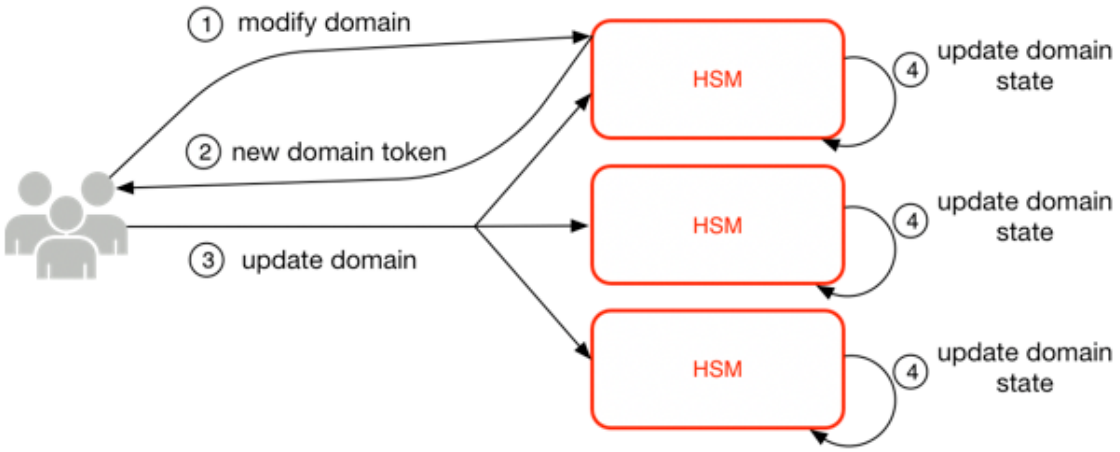
Token domain yang diekspor membentuk sumber kepercayaan mendasar untuk entitas yang beroperasi di dalam domain.

Mengelola status domain

Status domain dikelola melalui perintah terautentikasi kuorum. Perubahan ini termasuk modifikasi daftar peserta tepercaya di domain, modifikasi aturan kuorum untuk menjalankan perintah HSM, dan secara berkala memutar kunci domain. Perintah ini diautentikasi berdasarkan per-perintah sebagai lawan dari operasi sesi yang diautentikasi, seperti yang ditunjukkan pada gambar berikut.

Dalam keadaan inisialisasi dan operasionalnya, HSM berisi satu set kunci identitas asimetris yang dihasilkan sendiri, key pair penandatanganan, dan key pair key-establishment. Melalui proses manual, AWS KMS operator dapat membuat domain awal yang akan dibuat pada HSM pertama di suatu Wilayah. Domain awal ini terdiri atas status domain penuh seperti yang didefinisikan sebelumnya dalam topik ini. Hal ini diinstal melalui perintah penggabungan untuk tiap-tiap anggota HSM yang ditetapkan dalam domain.

Setelah HSM bergabung dengan domain awal, ia terikat dengan aturan yang didefinisikan dalam domain itu. Aturan ini mengatur perintah yang menggunakan kunci kriptografi pelanggan atau membuat perubahan pada status host atau domain. Operasi API sesi terautentikasi yang menggunakan kunci kriptografi Anda telah didefinisikan sebelumnya.



Gambar sebelumnya menggambarkan bagaimana status domain akan dimodifikasi. Prosesnya terdiri atas empat langkah:

1. Perintah berbasis kuorum dikirim ke HSM untuk memodifikasi domain.
2. Status domain baru dihasilkan dan diekspor sebagai token domain baru yang diekspor. Status pada HSM tidak dimodifikasi, yang berarti bahwa perubahan tidak diberlakukan pada HSM.
3. Perintah kedua dikirim ke masing-masing token domain yang baru diekspor untuk memperbarui status domain mereka dengan token domain baru. HSMs
4. Yang HSMs tercantum dalam token domain baru yang diekspor dapat mengautentikasi perintah dan token domain. Mereka juga dapat membongkar kunci domain untuk memperbarui status domain pada semua HSMs di domain.

HSMs Jangan berkomunikasi secara langsung satu sama lain. Sebaliknya, kuorum operator meminta perubahan ke status domain yang menghasilkan token domain baru yang diekspor. Anggota host layanan domain yang digunakan untuk mendistribusikan status domain baru untuk setiap HSM di domain.

Meninggalkan dan bergabung dari domain dilakukan melalui fungsi manajemen HSM. Modifikasi status domain dilakukan melalui fungsi manajemen domain.

Meninggalkan domain

Penyebab HSM meninggalkan domain, menghapus semua sisa dan kunci dari domain tersebut dari memori.

Bergabung ke domain

Penyebab HSM bergabung dengan domain baru atau memperbarui status domain saat ini ke status domain baru. Domain yang ada digunakan sebagai sumber set awal aturan untuk mengautentikasi pesan ini.

Membuat domain

Penyebab domain baru yang akan dibuat pada HSM. Mengembalikan token domain pertama yang dapat didistribusikan ke HSMs anggota domain.

Memodifikasi operator

Menambahkan atau menghapus operator dari daftar operator resmi dan peran mereka dalam domain.

Memodifikasi anggota

Menambahkan atau menghapus HSM dari daftar yang diotorisasi HSMs di domain.

Memodifikasi aturan

Memodifikasi set aturan kuorum yang diperlukan untuk menjalankan perintah pada HSM.

Merotasi kunci domain

Penyebab kunci domain baru yang akan dibuat dan ditandai sebagai kunci domain aktif. Ini memindahkan kunci aktif yang ada ke kunci yang dinonaktifkan dan menghapus kunci yang dinonaktifkan paling lama dari status domain.

Keamanan komunikasi internal

Perintah antara host layanan atau AWS KMS operator dan diamankan melalui dua mekanisme HSMs yang digambarkan dalamSesi yang diautentikasi: metode permintaan yang ditandatangani kuorum dan sesi yang diautentikasi menggunakan protokol host layanan HSM.

Perintah yang ditandatangani kuorum dirancang sedemikian rupa sehingga tidak ada operator tunggal yang dapat memodifikasi perlindungan keamanan penting yang disediakan. HSMs Perintah yang berjalan melalui sesi yang diautentikasi membantu memastikan bahwa hanya operator layanan resmi yang dapat melakukan operasi yang melibatkan kunci KMS. Semua informasi rahasia yang terikat pelanggan diamankan di seluruh infrastruktur. AWS

Pembentukan kunci

Untuk mengamankan komunikasi internal, AWS KMS gunakan dua metode pembentukan kunci yang berbeda. Yang pertama didefinisikan sebagai C (1, 2, ECC DH) dalam Rekomendasi untuk Skema Pendirian Kunci Berpasangan Menggunakan Kriptografi Logaritma Diskrit. Skema ini memiliki inisiator dengan kunci penandatanganan statis. Inisiator menghasilkan dan menandatangani kunci kurva eliptik efemeral Diffie-Hellman (ECDH), yang ditujukan untuk penerima dengan kunci perjanjian ECDH statis. Metode ini menggunakan satu kunci efemeral dan dua kunci statis menggunakan ECDH. Itu adalah derivasi dari label C (1, 2, ECC DH). Metode ini kadang disebut ECDH pas tunggal.

Metode pembentukan kunci yang kedua adalah C (2, 2, ECC, DH). Dalam skema ini, kedua belah pihak memiliki kunci penandatanganan statis, dan mereka menghasilkan, menandatangani, dan saling menukarkan kunci ECDH efemeral. Metode ini menggunakan dua kunci statis dan dua kunci efemeral, masing-masing menggunakan ECDH. Itu adalah derivasi dari label C (2, 2, ECC, DH).

Metode ini kadang-kadang disebut ECDH efemeral atau ECDHE. Semua kunci ECDH dihasilkan pada secp384r1 kurva (NIST-P384).

Batas keamanan HSM

Batas keamanan batin AWS KMS adalah HSM. HSM memiliki antarmuka berpemilik dan antarmuka fisik aktif lainnya tidak berada dalam status operasionalnya. HSM operasional disediakan selama inisialisasi dengan kunci kriptografi yang diperlukan untuk menetapkan perannya dalam domain. Bahan kriptografi sensitif dari HSM hanya akan disimpan di dalam memori volatil dan terhapus saat HSM meninggalkan status operasional, termasuk shutdown atau reset yang disengaja maupun yang tidak.

Operasi HSM API diautentikasi baik oleh perintah individu maupun melalui sesi rahasia yang saling diautentikasi yang ditetapkan oleh oleh host layanan.

Perintah yang ditandatangani kuorum

Perintah yang ditandatangani kuorum dikeluarkan oleh operator untuk. HSMs Bagian ini menjelaskan bagaimana perintah berbasis kuorum dibuat, ditandatangani, dan diautentikasi. Aturan-aturan ini cukup sederhana. Sebagai contoh, perintah Foo membutuhkan dua anggota dari peran Bar untuk diautentikasi. Ada tiga langkah dalam pembuatan dan verifikasi perintah berbasis kuorum. Langkah yang pertama adalah pembuatan perintah awal; yang kedua adalah pengiriman ke operator tambahan untuk ditandatangani; dan yang ketiga adalah verifikasi dan eksekusi.

Untuk tujuan memperkenalkan konsep, asumsikan bahwa ada satu set otentik kunci publik operator dan peran $\{QOS_s\}$, dan satu set kuorum-aturan QR = $\{Command_i, Rule_{\{i, t\}}\}$ di mana setiap Aturan adalah seperangkat peran dan jumlah minimum N {Role, N}. $_{t\ t}$ Agar perintah memenuhi aturan kuorum, set data perintah harus ditandatangani oleh satu set operator yang terdaftar di $\{QOS_s\}$ sehingga mereka memenuhi salah satu aturan yang tercantum untuk perintah itu. Seperti disebutkan sebelumnya, seperangkat aturan kuorum dan operator disimpan dalam status domain dan token domain yang diekspor.

Dalam praktiknya, penandatangan awal menandatangani perintah Sig_1 = Sign(dO_{p1}, Perintah) . Operator kedua juga menandatangani perintah Sig_2 = Sign(dO_{p2}, Perintah). Pesan yang ditandatangani dua kali dikirim ke HSM untuk dilaksanakan. HSM melakukan hal berikut:

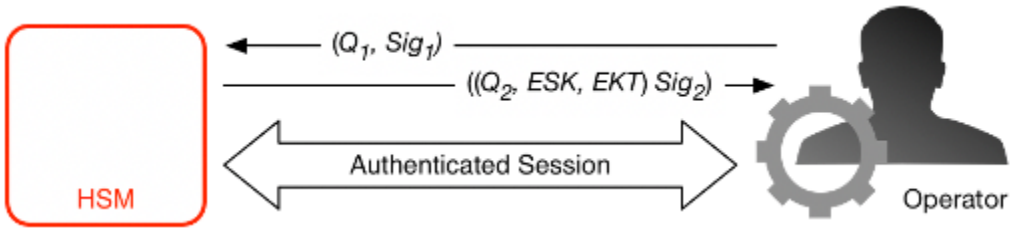
1. Untuk setiap tanda tangan, ia mengekstrak kunci publik penandatangan dari status domain dan memverifikasi tanda tangan pada perintah.
2. Ini memverifikasi bahwa himpunan penandatangan memenuhi aturan untuk perintah.

Sesi yang diautentikasi

Operasi kunci Anda berjalan antara host yang menghadap secara eksternal dan AWS KMS host. HSMs Perintah ini berkaitan dengan penciptaan dan penggunaan kunci kriptografi dan pembuatan nomor acak yang aman. Perintah berjalan melalui saluran yang diautentikasi sesi antara host layanan dan. HSMs Selain kebutuhan akan autentikasi, sesi ini membutuhkan kerahasiaan. Perintah yang berjalan di sesi ini mencakup kembalinya kunci data cleartext dan pesan terdekripsi yang ditujukan untuk Anda. Untuk memastikan bahwa sesi ini tidak dapat ditumbangkan melalui man-in-the-middle serangan, sesi diautentikasi.

Protokol ini melakukan perjanjian kunci ECDHE yang saling diautentikasi antara HSM dan host layanan. Pertukaran diprakarsai oleh host layanan dan diselesaikan oleh HSM. HSM juga mengembalikan kunci sesi (SK) yang dienkripsi oleh kunci negosiasi dan token kunci yang diekspor yang berisi kunci sesi. Token kunci yang diekspor berisi masa berlaku, sehingga host layanan harus menegosiasikan ulang kunci sesi setelah masa berlaku token tersebut habis.

Host layanan adalah anggota domain dan memiliki key pair penandatanganan identitas ($DHOS_i$, $QHOS_i$) dan salinan otentik dari 'kunci publik identitas. HSMs Ia menggunakan set kunci penandatanganan identitas untuk secara aman menegosiasikan kunci sesi yang dapat digunakan antara host layanan dan setiap HSM di domain. Token kunci yang diekspor memiliki masa berlaku yang terkait dengannya, dan kunci baru harus dinegosiasikan setelah masa berlaku token habis.



Proses dimulai dengan pengenalan host layanan yang memerlukan kunci sesi untuk mengirimkan dan menerima komunikasi sensitif yang mengalir antara host ini dan anggota HSM domain.

1. Host layanan menghasilkan pasangan kunci efemeral ECDH (d_1, Q_1) dan menandatanganinya dengan kunci identitasnya Sig1 = Sign(dOS,Q_1).
2. HSM memverifikasi tanda tangan pada kunci publik yang diterima menggunakan token domain saat ini dan membuat pasangan kunci efemeral ECDH (d_2, Q_2). Kemudian melengkapi ECDH-key-exchange sesuai dengan Rekomendasi untuk Skema Pendirian Kunci Berpasangan Menggunakan Kriptografi Logaritma Diskrit untuk membentuk kunci AES-GCM 256-bit yang dinegosiasikan. HSM membuat kunci sesi AES-GCM 256-bit yang baru. Ia mengenkripsi kunci sesi dengan kunci negosiasi untuk membentuk kunci sesi terenkripsi (ESK). Hal ini juga mengenkripsi kunci sesi di

bawah kunci domain sebagai token kunci yang diekspor EKT. Akhirnya, ia menandatangani nilai kembali dengan pasangan kunci identitasnya Sig_2 = Sign(dHSK, (Q_2, ESK, EKT)).

3. Host layanan memverifikasi tanda tangan pada kunci yang diterima menggunakan token domain saat ini. Host layanan kemudian menyelesaikan pertukaran kunci ECDH sesuai dengan Rekomendasi untuk Skema Pendirian Kunci Berpasangan Menggunakan Kriptografi Logaritma Diskrit. Berikutnya ia mendekripsi ESK untuk mendapatkan kunci sesi SK.

Selama masa berlaku di EKT, host layanan dapat menggunakan kunci sesi negosiasi SK untuk mengirim perintah terenkripsi envelope ke HSM. Setiap service-host-initiated perintah atas sesi yang diautentikasi ini mencakup EKT. HSM merespons menggunakan kunci sesi SK ternegosiasi yang sama.

Proses replikasi untuk kunci Multi-wilayah

AWS KMS menggunakan mekanisme replikasi lintas wilayah untuk menyalin materi kunci dalam kunci KMS dari HSM dalam satu Wilayah AWS ke HSM yang berbeda. Wilayah AWS Agar mekanisme ini berfungsi, kunci KMS yang sedang direplikasi harus berupa kunci Multi-wilayah dengan asal. `AWS_KMS` Untuk kunci Multi-wilayah dengan bahan kunci impor (`EXTERNAL`asal), Anda bertanggung jawab untuk mereplikasi materi kunci dengan mengimpor bahan kunci yang sama ke setiap replika satu per satu.

Saat mereplikasi kunci KMS dari satu Wilayah ke Wilayah lainnya, HSMs di Wilayah tidak dapat berkomunikasi secara langsung, karena mereka berada di jaringan yang terisolasi. Sebagai gantinya, pesan yang dipertukarkan selama replikasi Lintas wilayah dikirimkan oleh layanan proxy. Selama replikasi lintas wilayah, setiap pesan yang dihasilkan oleh AWS KMS HSM ditandatangani secara kriptografis menggunakan kunci penandatanganan replikasi. Kunci penandatanganan replikasi (RSKs) adalah kunci ECDSA pada kurva NIST P-384. Setiap Wilayah memiliki setidaknya satu RSK, dan komponen publik dari setiap RSK dibagi dengan setiap Wilayah lain dalam partisi yang sama. AWS

Proses replikasi lintas wilayah untuk menyalin materi kunci dari Wilayah A ke Wilayah B berfungsi sebagai berikut:

1. HSM di Wilayah B menghasilkan kunci ECDH fana pada kurva NIST P-384, Kunci Perjanjian Replikasi B (RAKB). Komponen publik RAKB dikirim ke HSM di Wilayah A oleh layanan proxy.
2. HSM di Wilayah A menerima komponen publik RAKB dan kemudian menghasilkan kunci ECDH fana lainnya pada kurva NIST P-384, Kunci Perjanjian Replikasi A (RAKA). HSM menjalankan

skema pembentukan kunci ECDH pada RAKA dan komponen publik RAKB, dan memperoleh kunci simetris dari output, Replication Wrapping Key (RWK). RWK digunakan untuk mengenkripsi materi kunci dari kunci KMS Multi-wilayah yang sedang direplikasi.

3. Komponen publik RAKA dan materi kunci yang dienkripsi dengan RWK dikirim ke HSM di Wilayah B melalui layanan proxy.
4. HSM di Wilayah B menerima komponen publik RAKA dan materi kunci yang dienkripsi menggunakan RWK. HSM berasal dari RWK dengan menjalankan skema pembentukan kunci ECDH pada RAKB dan komponen publik RAKA.
5. HSM di Wilayah B menggunakan RWK untuk mendekripsi materi kunci dari Wilayah A.

Perlindungan daya tahan

Daya tahan layanan tambahan untuk kunci yang dihasilkan oleh layanan disediakan oleh penggunaan offline HSMs, beberapa penyimpanan nonvolatile dari token domain yang diekspor, dan penyimpanan kunci KMS terenkripsi yang berlebihan. Offline HSMs adalah anggota domain yang ada. Dengan pengecualian tidak online dan berpartisipasi dalam operasi domain reguler, offline HSMs muncul secara identik dalam status domain sebagai anggota HSM yang ada.

Desain daya tahan dimaksudkan untuk melindungi semua kunci KMS di suatu Wilayah harus AWS mengalami kerugian skala besar baik online HSMs atau set kunci KMS yang disimpan dalam sistem penyimpanan utama kami. AWS KMS keys dengan bahan kunci impor tidak termasuk dalam perlindungan daya tahan yang diberikan kunci KMS lainnya. Jika terjadi kegagalan di seluruh wilayah AWS KMS, bahan kunci yang diimpor mungkin perlu diimpor kembali ke kunci KMS.

Offline HSMs, dan kredensil untuk mengaksesnya, disimpan di brankas di dalam ruang aman yang dipantau di beberapa lokasi geografis independen. Setiap brankas membutuhkan setidaknya satu petugas AWS keamanan dan satu AWS KMS operator, dari dua tim independen di AWS, untuk mendapatkan bahan-bahan ini. Penggunaan bahan-bahan ini diatur oleh kebijakan internal yang mengharuskan kuorum AWS KMS operator hadir.

Riwayat dokumen

Topik ini menjelaskan pembaruan yang signifikan untuk Panduan Pengembang AWS Key Management Service .

Topik

- Pembaruan terkini
- Pembaruan lebih awal

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 dokumentasi	Memperbarui halaman algoritma kriptografi yang didukung untuk secara akurat mencantumkan algoritma kriptografi yang mendukung. AWS KMS	September 29, 2026
Hibah utama layanan	AWS KMS sekarang mendukung pembuatan hibah untuk prinsipal AWS layanan. Anda dapat menggunakan `RetiringServicePrincipal` parameter baru `GranteeServicePrincipal` dan	Maret 6, 2026

	dalam `CreateGrant` operasi untuk menentuka n kepala layanan sebagai penerima hibah atau kepala pensiun hibah. Rilis ini juga memperkenalkan batasan `SourceArn` hibah, yang membatasi izin hibah untuk permintaan yang dibuat atas nama sumber daya tertentu. AWS Tiga tombol kondisi baru tersedia untuk `CreateGrant` permintaa n:`kms:GranteeService Principal` ,`kms:Retir ingServicePrincipal` , dan`kms:GrantConstrain tSourceArn` . Untuk informasi lebih lanjut, lihat Hibah di AWS KMS.	
Pembaruan fitur	AWS KMS mendukung rotasi on-demand enkripsi simetris, kunci multi-wilayah dengan bahan kunci yang diimpor.	November 26, 2025
Pembaruan fitur	AWS KMS mendukung `keyExchange` bidang baru di `tlsDetails` bagian CloudTrail acara.	November 24, 2025
Pembaruan fitur	AWS KMS mendukung `KeySpec` ECC_NIST_ EDWARDS25519 baru.	November 7, 2025
Dual-stack dukungan	AWS KMS mendukung dual-stack.	Juni 18, 2025

Pembaruan fitur	Menambahkan dukungan untuk Module-Lattice Digital Signature Algorithm (ML-DSA) tanda tangan kriptografi pasca-kuantum.	Juni 13, 2025
Rotasi kunci yang diimpor	Anda dapat melakukan rotasi sesuai permintaan kunci KMS enkripsi simetris dengan bahan kunci impor (asal). `EXTERNAL`	Juni 5, 2025
Pembaruan fitur	Menambahkan dukungan untuk kunci KMS Multi-wilayah di Wilayah Tiongkok.	November 21, 2024
AWS pembaruan kebijakan terkelola	Memperbarui peran AWSKeyManagementServiceMultiRegionKeysServiceRolePolicyterkait layanan dengan menambahkan ID pernyataan (`Sid`) ke kebijakan terkelola dengan kebijakan versi v2.	November 21, 2024
Perubahan kuota	Meningkatkan tingkat permintaan default untuk ImportKeyMaterialdan DeleteImportedKeyMaterialpermintaan.	Juli 23, 2024
Perubahan kuota	Meningkatkan tingkat permintaan operasi kriptografi default untuk kunci KMS enkripsi simetris, kunci KMS RSA, dan kunci ECC dan SM2 KMS.	Juli 8, 2024

Fitur baru	Menambahkan `KeyUsage` tipe baru `KEY_AGREEMENT` untuk kunci KMS kurva NIST-recommended elips (ECC) dan SM2 (hanya Wilayah Tiongkok) dan menambahkan dukungan untuk memperoleh rahasia bersama.	Juni 13, 2024
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 untuk `AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy` yang memungkinkan AWS KMS untuk memantau perubahan di VPC yang berisi cluster AWS CloudHSM Anda AWS KMS sehingga dapat memberikan pesan kesalahan yang jelas jika terjadi kegagalan.	10 November 2023
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	Maret 10, 2023
Pembaruan fitur	Algoritma `RSAES_PKCS1_V1_5` pembungkus tidak digunakan lagi. AWS KMS akan mengakhiri semua dukungan untuk `RSAES_PKCS1_V1_5` pada 1 Oktober 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 lebih awal

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.

Ubah	Deskripsi	Tanggal
Konten baru	Penambahan dokumentasi tentang Tag di AWS KMS.	Februari 15, 2017
Konten baru	Penambahan dokumentasi tentang Monitor AWS KMS keys dan Pantau tombol KMS dengan Amazon CloudWatch.	Agustus 31, 2016
Konten baru	Penambahan dokumentasi tentang Materi kunci yang diimpor.	Agustus 11, 2016
Konten baru	Ditambahkan dokumentasi berikut:Kebijakan IAM,Referensi izin, danKunci syarat.	Juli 5, 2016
Pembaruan	Bagian yang diperbarui dari dokumentasi di bab Akses dan izin kunci KMS.	Juli 5, 2016

Ubah	Deskripsi	Tanggal
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 utama untuk meningkatkan kejelasan dan akurasi.	Februari 17, 2016
Perbarui	Memperbarui halaman topik kunci Mengelola KMS untuk meningkatkan kejelasan.	Januari 5, 2016
Konten baru	Ditambahkan dokumentasi tentang CloudTrail.	November 18, 2015
Konten baru	Penambahan instruksi untuk Mengubah kebijakan utama.	November 18, 2015

Ubah	Deskripsi	Tanggal
Perbarui	Memperbarui dokumentasi tentang Bagaimana Amazon Relational Database Service AWS KMS menggunakan.	November 18, 2015
Konten baru	Ditambahkan dokumentasi tentang Amazon WorkSpaces.	November 6, 2015
Pembaruan	Pembaruan halaman Kebijakan utama di AWS KMS untuk meningkatkan kejelasan .	Oktober 22, 2015
Konten baru	Penambahan dokumenta si tentang Hapus sebuah AWS KMS key, termasuk dokumentasi pendukung tentang Membuat alarm dan Menentukan penggunaan kunci KMS sebelumnya.	Oktober 15, 2015
Konten baru	Penambahan dokumentasi tentang Menentukan akses ke AWS KMS keys.	Oktober 15, 2015
Konten baru	Penambahan dokumentasi tentang Negara-negara kunci AWS KMS keys.	Oktober 15, 2015
Konten baru	Ditambahkan dokumentasi tentang Amazon Simple Email Service.	Oktober 1, 2015
Pembaruan	Pembaruan halaman Kuota untuk menjelaskan kuota permintaan baru.	Agustus 31, 2015

Ubah	Deskripsi	Tanggal
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 menunjukkan penggunaan operasi `UpdateAlias` .	1 Juni 2015
Perbarui	AWS Key Management Service Memindahkan tabel daerah ke Referensi Umum AWS.	29 Mei 2015
Konten baru	Penambahan dokumentasi tentang Bagaimana Amazon EMR menggunakan AWS KMS.	Januari 28, 2015
Konten baru	Ditambahkan dokumentasi tentang Amazon WorkMail.	Januari 28, 2015
Konten baru	Menambahkan dokumentasi tentang Bagaimana Amazon Relational Database Service AWS KMS menggunakan.	Januari 6, 2015
Konten baru	Ditambahkan dokumentasi tentang Amazon Elastic Transcoder.	November 24, 2014

Ubah	Deskripsi	Tanggal
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.