



Guia do desenvolvedor

AWS Key Management Service



AWS Key Management Service: Guia do desenvolvedor

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

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

AWS Key Management Service	1
Por que usar AWS KMS?	1
AWS KMS in Regiões da AWS	2
AWS KMS preços	2
AWS KMS contrato de nível de serviço	2
Acessando AWS Key Management Service	3
Console de gerenciamento da AWS	3
Permissões necessárias para usar o AWS KMS console	3
AWS Command Line Interface	4
AWS KMS REST API	4
AWS SDKs	4
Trabalhando com AWS SDKs	5
AWS Encryption SDK	6
AWS KMS consistência eventual	6
TLS pós-quântico híbrido	7
Sobre o TLS pós-quântico	8
Como usar	9
Configurar o TLS pós-quântico híbrido	9
Saiba mais	11
Conecte-se AWS KMS por meio de um VPC endpoint	12
Crie um VPC endpoint para AWS KMS	13
Conectar-se a um endpoint da VPC	14
Use VPC endpoints para controlar o acesso aos recursos AWS KMS	15
Registrar AWS KMS solicitações que usam um VPC endpoint	19
Endpoints de pilha dupla	20
Recursos não disponíveis acima IPv6	21
Conceitos	22
Introdução	22
Objetivos de projeto	23
AWS KMS keys	24
Chaves gerenciadas pelo cliente	28
Chaves gerenciadas pela AWS	28
Chaves pertencentes à AWS	29
AWS KMS key hierarquia	30

Identificadores-chave () KeyId	32
Chaves assimétricas	35
Chaves de HMAC	37
Chaves ML-DSA	39
Chaves de várias regiões	40
Material de chave importado	52
Chaves do KMS em um repositório de chaves do CloudHSM	59
Chaves do KMS em repositórios de chaves externos	62
AWS KMS fundamentos da criptografia	65
Entropia e geração de números aleatórios	65
Operações de chave simétrica (somente criptografia)	66
Operações de chave assimétrica (criptografia, assinatura digital e verificação de assinatura)	66
Funções de derivação de chave	67
AWS KMS uso interno de assinaturas digitais	67
Criptografia de envelope	67
Operações criptográficas	69
Algoritmos compatíveis	71
Algoritmos criptográficos	71
Acesso e permissões de chave do KMS	75
Políticas de chave do KMS	75
Concessões de chave do KMS	76
Políticas de chaves	77
Criação de uma política de chave	77
Política de chaves padrão	85
Visualizar uma política de chave	101
Alterar uma política de chave	104
Permissões para AWS serviços	107
Políticas do IAM	108
Permitir que diversas entidades principais do IAM acessem uma chave do KMS	110
Práticas recomendadas para políticas do IAM	111
Especificando chaves do KMS em instruções de políticas do IAM	113
Exemplos	117
Políticas de controle de recursos	123
Concessões	126
Conceitos sobre concessões	128

Práticas recomendadas	133
Controlar o acesso a concessões	134
Criar concessões	135
Visualizar concessões	144
Usar um token de concessão	145
Retirar e revogar concessões	147
Chaves de condição	148
AWS chaves de condição globais	149
AWS KMS chaves de condição	152
AWS KMS chaves de condição para plataformas certificadas	224
Permissões de privilégio mínimo	230
Implementação de permissões de privilégio mínimo	231
Controle de acesso por atributo (ABAC)	235
Chaves de condição ABAC para AWS KMS	236
Etiquetas ou aliases?	239
Solução de problemas do ABAC para AWS KMS	240
Regras de controle de acesso com base em função (RBAC)	245
Acesso entre contas	247
Etapa 1: Incluir uma declaração de política de chaves na conta local	249
Etapa 2: Adicionar políticas do IAM à conta externa	253
Permitindo o uso de chaves KMS externas com Serviços da AWS	255
Usar chaves do KMS em outras contas	255
Controlar o acesso a chaves de várias regiões	256
Noções básicas de autorização para chaves de várias regiões	257
Autorizar administradores e usuários de chave de várias regiões	258
Determinar o acesso	263
Examinar a política de chaves	263
Examinar políticas do IAM	267
Examinar concessões	269
Contexto de criptografia	270
Regras de contexto de criptografia	271
Contexto de criptografia em políticas	272
Contexto de criptografia em concessões	273
Registrar em log o contexto de criptografia	273
Armazenar o contexto de criptografia	274
Testar suas permissões	274

O que DryRun é	275
Especificando DryRun com a API	276
Solução de problemas de AWS KMS permissões	276
Exemplo 1: O acesso a uma chave KMS é negado ao usuário em seu Conta da AWS	279
Exemplo 2: O usuário assume a função com permissão para usar uma chave KMS em uma outra Conta da AWS	281
Glossário	285
Autenticação	285
Autorização	285
Autenticação com identidades	285
Gerenciar o acesso usando políticas	287
AWS KMS recursos	289
Criar uma chave do KMS	290
Permissões para criar chaves do KMS	292
Escolher qual tipo de chave do KMS criar	293
Criar uma chave do KMS de criptografia simétrica	296
Criar uma chave do KMS assimétrica	301
Criar uma chave do KMS HMAC	308
Criar chaves primárias de várias regiões	313
Criar chaves de réplica de várias regiões	318
Etapa 1: escolher regiões de réplica	319
Etapa 2: criar chaves de réplica	319
Criar uma chave do KMS com material de chave importado	325
Permissões para importar material de chave	327
Requisitos para material de chave importada	328
Etapa 1: criar um material AWS KMS key sem chave	331
Etapa 2: Fazer download da chave pública de empacotamento e do token de importação ...	334
Etapa 3: Criptografar o material de chave	344
Etapa 4: Importar o material de chave	354
Criar uma chave KMS em um armazenamento de AWS CloudHSM chaves	361
Criar uma nova chave do KMS em seu repositório de chaves do CloudHSM	362
Criar uma chave do KMS em repositórios de chaves externos	369
Requisitos para uma chave do KMS em um armazenamento de chaves externas	371
Criar uma nova chave do KMS em seu repositório de chaves externo	372
Identificar e visualizar chaves	381
Encontrar o ID e o ARN da chave	381

Acessar e listar detalhes de chaves do KMS	383
Identificar diferentes tipos de chaves	393
Identificar chaves do KMS assimétricas	393
Identificar chaves do KMS de HMAC	395
Identificar chaves do KMS de várias regiões	395
Identificar chaves do KMS com material de chave importado	396
Identificar chaves KMS em lojas de AWS CloudHSM chaves	397
Identificar chaves do KMS em repositórios de chaves externos	398
Personalizar a exibição do console	399
Classificar e filtrar suas chaves do KMS	399
Personalizar suas tabelas de chaves do KMS	402
Encontre chaves KMS e material de chaves em um armazenamento de AWS CloudHSM	
chaves	404
Encontre as chaves KMS em um armazenamento de AWS CloudHSM chaves	405
Encontre todas as chaves de uma loja de AWS CloudHSM chaves	407
Encontre a chave KMS para uma AWS CloudHSM chave	409
Encontre a AWS CloudHSM chave para uma chave KMS	414
Habilitar e desabilitar chaves	418
Alternar chaves	421
Por que alternar as chaves do KMS?	423
Funcionamento da alternância	424
Habilitar a alternância automática de chave	429
Desabilitar a alternância automática de chave	432
Executar alternância de chaves sob demanda	434
Como iniciar a alternância de chaves sob demanda (console)	435
Iniciando a rotação de chaves sob demanda (API)AWS KMS	436
Listar rodízios e materiais de chave	437
Listar rodízios e materiais de chave (console)	437
Listar rotações e principais materiais (AWS KMS API)	438
Alternar chaves manualmente	440
Alterar a chave primária em um conjunto de chaves de várias regiões	442
Atualizar a região primária	444
Excluir chaves	447
Sobre o período de espera	448
Considerações especiais	449
Controlar o acesso à exclusão de chaves	452

Permitir que administradores de chaves agendem e cancelem a exclusão de chaves	453
Agendar exclusão de chave	455
Cancelar a exclusão de chaves	457
Criar um alarme	458
Determinar a utilização anterior de uma chave do KMS	460
Examinar as permissões da chave do KMS para determinar o escopo da utilização em potencial	461
Examine AWS CloudTrail os registros para determinar o uso real	461
Excluir material de chave importado	464
Gerar chaves de dados	467
Criação de uma chave de dados	467
Funcionamento das operações criptográficas com chaves de dados	468
Criptografia de dados com uma chave de dados	468
Descriptografia dos dados com uma chave de dados	469
Como as chaves do KMS inutilizáveis afetam as chaves de dados	470
Gerar pares de chaves de dados	472
Criar um par de chaves de dados	472
Funcionamento das operações criptográficas com pares de chaves de dados	474
Criptografar dados com um par de chaves de dados	474
Descriptografar dados com um par de chaves de dados	474
Assinar mensagens com um par de chaves de dados	475
Verificar uma assinatura com um par de chaves de dados	476
Derivar um segredo compartilhado com pares de chaves de dados	477
Realizar operações offline com chaves públicas	478
Considerações especiais sobre o download de chaves públicas	479
Fazer download de chave pública	480
Exemplo de operações offline	482
Derivar segredos compartilhados offline	482
Verificação off-line com pares de chaves ML-DSA	483
Verificação off-line com pares de SM2 chaves (somente regiões da China)	486
Monitorar chaves	492
Ferramentas de monitoramento	493
Ferramentas automatizadas	493
Ferramentas manuais	494
Fazendo login com AWS CloudTrail	495
Localizando entradas de AWS KMS registro em CloudTrail	496

Excluindo AWS KMS eventos de uma trilha	497
Exemplos de entradas de AWS KMS registro	498
Monitore as teclas com CloudWatch	584
AWS KMS métricas e dimensões	584
Crie um CloudWatch alarme para expiração do material chave importado	594
Crie CloudWatch alarmes para armazenamentos externos de chaves	595
Monitore as chaves com a Amazon EventBridge	600
Alternância de CMKs do KMS	600
Expiração do material de chave importado do KMS	601
Exclusão da CMK do KMS	602
Aliases	603
Funcionamento dos aliases	604
Controlar o acesso a aliases	607
kms:CreateAlias	607
kms:ListAliases	608
kms:UpdateAlias	609
kms>DeleteAlias	611
Limitar permissões de alias	612
Criar alias	613
Encontrar o nome e o ARN do alias	615
Atualizar aliases	620
Excluir um alias	621
Usar aliases para controlar o acesso a chaves do KMS	623
kms:RequestAlias	624
kms:ResourceAliases	625
Saiba como usar aliases em suas aplicações	627
Encontre aliases em registros AWS CloudTrail	629
Tags	631
Controlar o acesso às tags	632
Permissões de etiquetas em políticas	633
Limitar permissões de etiquetas	635
Adicionar tags.	637
Adicionar etiquetas ao criar uma chave do KMS	637
Adicionar tags a chaves existentes do KMS	639
Editar tags	640
Remover marcações	641

Visualizar tags	642
Usar tags para controlar o acesso a chaves do KMS	644
Repositórios de chaves	649
AWS KMS armazenamento de chaves padrão	649
AWS KMS armazenamento de chaves padrão com material de chave importado	649
AWS KMS lojas de chaves personalizadas	651
AWS CloudHSM armazenamento de chaves	652
Armazenamento de chaves externas	652
AWS CloudHSM lojas principais	653
AWS CloudHSM conceitos-chave da loja	656
Controle o acesso à sua loja de AWS CloudHSM chaves	659
Crie um armazenamento de AWS CloudHSM chaves	661
Exibir uma loja de AWS CloudHSM chaves	668
Editar configurações de armazenamento de AWS CloudHSM chaves	671
Connect a um armazenamento de AWS CloudHSM chaves	675
Desconectar um armazenamento de AWS CloudHSM chaves	679
Excluir um armazenamento de AWS CloudHSM chaves	683
Solucionar problemas de um armazenamento de chaves personalizado	686
Armazenamentos de chaves externas	702
Conceitos de armazenamento de chaves externas	707
Como funcionam os armazenamentos de chaves externas	717
Controlar o acesso ao repositório de chaves externo	719
Escolher uma opção de conectividade de proxy	725
Criar um repositório de chaves externo	739
Editar propriedades do repositório de chaves externo	754
Visualizar repositórios de chaves externos	761
Monitorar repositórios de chaves externos	768
Conectar e desconectar repositórios de chaves externos	781
Excluir um repositório de chaves externo	792
Solução de problemas de armazenamentos de chaves externas	794
Segurança	822
Proteção de dados	823
Proteger material de chave	823
Criptografia de dados	825
Privacidade entre redes	826
Gerenciamento de identidade e acesso	827

AWS políticas gerenciadas	828
Perfis vinculados ao serviço	831
Registro em log e monitoramento	837
Validação de conformidade	839
Documentos de conformidade e segurança	839
Saiba mais	840
Resiliência	840
Isolamento regional	841
Design de vários locatários	841
Melhores práticas de resiliência em AWS KMS	842
Segurança da infraestrutura	842
Isolamento de hosts físicos	844
Cotas	845
Cotas de recurso	845
AWS KMS keys: 100,000	846
Aliases por chave do KMS: 50	846
Concessões por chave do KMS: 50.000	847
Cota de recursos de armazenamento de chaves personalizado: 10	847
Rotação sob demanda: 25	848
Cotas de solicitações	848
Solicite cotas para cada operação de AWS KMS API	849
Aplicar cotas de solicitações	856
Cotas compartilhadas para operações de criptografia	857
Solicitações de API dofeitas em seu nome	858
Solicitações entre contas	859
Cotas de solicitação de armazenamento de chaves personalizadas	859
Controle de utilização de solicitações do	861
Exemplos de código	863
Conceitos básicos	864
Olá AWS KMS	865
Conheça os conceitos básicos	869
Ações	943
Cenários	1114
Trabalhar com a criptografia de tabelas	1114
Suporte de atestado criptográfico em AWS KMS	1117
Como fazer chamadas atestadas para AWS KMS	1119

Monitorar solicitações atestadas	1119
Solicitações de monitoramento para Nitro enclaves	1120
Monitorar solicitações para o NitroTPM	1125
Serviços de criptografia AWS	1130
Amazon Elastic Block Store (Amazon EBS)	1131
Criptografia de Amazon EBS	1131
Usar chaves do KMS e chaves de dados	1132
Contexto de criptografia do Amazon EBS	1132
Detectar falhas do Amazon EBS	1133
Usando AWS CloudFormation para criar volumes criptografados do Amazon EBS	1134
Amazon EMR	1134
Criptografar dados no EMR File System (EMRFS)	1135
Criptografar dados nos volumes de armazenamento dos nós de cluster	1138
Contexto de criptografia	1139
banco de dados de origem	1140
Criptografia do Amazon Redshift	1140
Contexto de criptografia	1141
Referência	1142
Referência de estado de chave	1143
Estados de chave e tipos de chaves do KMS	1143
Tabela de estados de chave	1144
Referência de tipos de chaves	1152
Tabela de tipos de chave	1152
Tabela de recursos especiais	1160
Referência de especificação de chave	1170
Especificação da chave SYMMETRIC_DEFAULT	1171
Especificações da chave RSA	1173
Especificações da chave de curva elíptica	1177
Especificações de chave para chaves do KMS de HMAC	1180
Especificações de chave ML-DSA	1180
SM2 especificação chave (somente regiões da China)	1181
Referência de permissões	1182
Descrições das colunas	1232
AWS KMS operações internas	1234
Domínios e estado do domínio	1235
Segurança de comunicação interna	1239

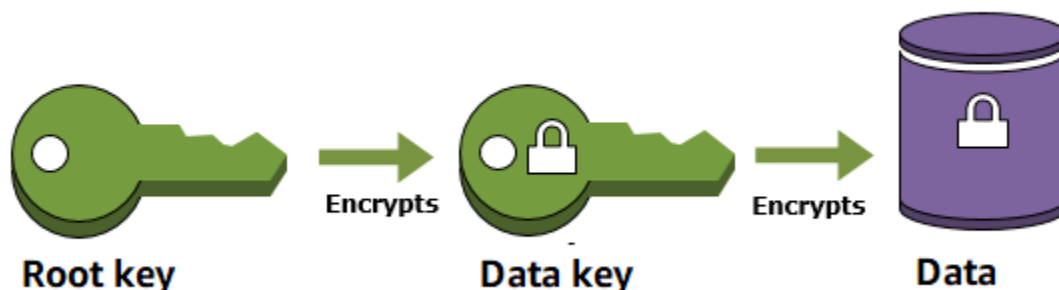
Processo de replicação para chaves em multirregiões	1242
Proteção de durabilidade	1243
Histórico do documento	1245
Atualizações recentes	1245
Atualizações anteriores	1252
.....	mcclvii

AWS Key Management Service

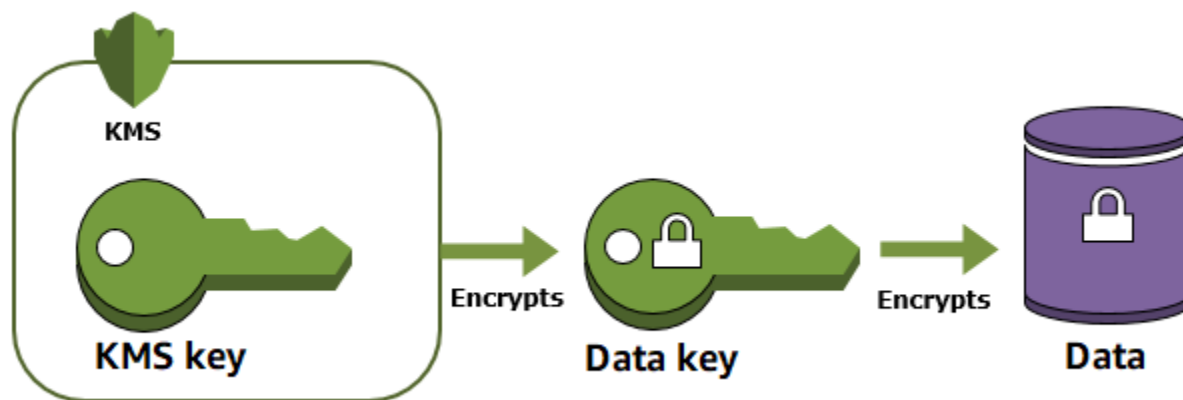
AWS Key Management Service (AWS KMS) é um serviço AWS gerenciado que facilita a criação e o controle das chaves usadas para criptografar e assinar seus dados. Os AWS KMS keys que você cria AWS KMS são protegidos por [módulos de segurança de hardware \(HSM\) validados pelo FIPS 140-3 Security Level 3](#). Eles nunca saem AWS KMS sem criptografia. Para usar ou gerenciar suas chaves KMS, você interage com AWS KMS.

Por que usar AWS KMS?

Ao criptografar dados, você precisa proteger a chave de criptografia. Se você criptografar sua chave, precisará proteger sua chave de criptografia. Eventualmente, você deverá proteger a chave de criptografia de nível mais alto (conhecida como chave-raiz) na hierarquia que protege seus dados. É aí que AWS KMS entra.



AWS KMS protege suas chaves raiz. As chaves KMS são criadas, gerenciadas, usadas e excluídas inteiramente nele AWS KMS. Elas nunca saem do serviço sem estarem criptografadas. Para usar ou gerenciar suas chaves KMS, você liga AWS KMS.



Além disso, você pode criar e gerenciar [políticas de chaves](#) em AWS KMS, garantindo que somente usuários confiáveis tenham acesso às chaves do KMS.

AWS KMS in Regiões da AWS

Os itens Regiões da AWS nos quais AWS KMS é suportado estão listados em [AWS Key Management Service Endpoints e Quotas](#). Se um AWS KMS recurso não for suportado em um Região da AWS que AWS KMS ofereça suporte, a diferença regional será descrita no tópico sobre o recurso.

AWS KMS preços

Como acontece com outros AWS produtos, o uso AWS KMS não exige contratos ou compras mínimas. Para obter mais informações sobre AWS KMS preços, consulte [AWS Key Management Service Preços](#).

AWS KMS contrato de nível de serviço

AWS Key Management Service é respaldado por um [contrato de nível de serviço](#) que define nossa política de disponibilidade de serviços.

Acessando AWS Key Management Service

Você pode trabalhar com AWS KMS as seguintes formas:

Tópicos

- [Console de gerenciamento da AWS](#)
- [AWS Command Line Interface](#)
- [AWS KMS REST API](#)
- [AWS SDKs](#)
- [Usando esse serviço com um AWS SDK](#)
- [AWS Encryption SDK](#)
- [AWS KMS consistência eventual](#)
- [Usando TLS pós-quântico híbrido com AWS KMS](#)
- [Conecte-se AWS KMS por meio de um VPC endpoint](#)
- [Suporte a endpoint de pilha dupla](#)

Console de gerenciamento da AWS

O console é uma interface de usuário baseada na web para gerenciamento AWS KMS e AWS recursos. Se você se inscreveu em uma Conta da AWS, você pode acessar o AWS KMS console fazendo login no Console de gerenciamento da AWS e escolhendo AWS KMS na página Console de gerenciamento da AWS inicial.

Permissões necessárias para usar o AWS KMS console

Para trabalhar com o AWS KMS console, os usuários devem ter um conjunto mínimo de permissões que lhes permita trabalhar com os AWS KMS recursos em sua Conta da AWS. Além dessas permissões do AWS KMS, os usuários precisam ter também permissões para listar usuários do IAM e perfis do IAM. Se você criar uma política do IAM que seja mais restritiva que as permissões mínimas necessárias, o console do AWS KMS não funcionará como pretendido para os usuários com essa política do IAM.

Para obter as permissões mínimas necessárias para conceder ao usuário acesso somente leitura ao console do AWS KMS, consulte [Permitir que um usuário visualize as chaves KMS no console AWS KMS](#).

Para permitir que os usuários trabalhem com o AWS KMS console para criar e gerenciar chaves KMS, anexe a política `AWSKeyManagementServicePowerUser` gerenciada ao usuário, conforme descrito em [AWS políticas gerenciadas para AWS Key Management Service](#).

Você não precisa permitir permissões mínimas do console para usuários que estão trabalhando com a AWS KMS API por meio do [AWS SDKs](#), [AWS Command Line Interface](#), ou [Ferramentas da AWS para PowerShell](#). No entanto, você precisa conceder permissão a esses usuários para usar a API. Para obter mais informações, consulte [Referência de permissões](#).

AWS Command Line Interface

Você pode usar as AWS CLI ferramentas para emitir comandos ou criar scripts na linha de comando do seu sistema para realizar AWS (inclusive AWS KMS) tarefas.

Para obter mais informações sobre como usar AWS KMS por meio do AWS CLI, consulte a [Referência de AWS CLI Comandos](#)

AWS KMS REST API

A arquitetura do AWS KMS foi projetada para ser neutra em termos de linguagem de programação. REST API é uma interface HTTP para AWS KMS. Com a REST API, você usa solicitações HTTP padrão para criar, buscar e excluir chaves.

Para obter mais informações sobre como usar o AWS KMS REST API, consulte a [Referência AWS Key Management Service da API](#)

AWS SDKs

AWS fornece SDKs (kits de desenvolvimento de software) que consistem em bibliotecas e exemplos de código para linguagens e plataformas de programação comuns (Java JavaScript, C, Python e assim por diante). Eles AWS SDKs fornecem uma maneira conveniente de criar acesso programático a AWS KMS e AWS KMS é um REST serviço. Você pode enviar solicitações AWS KMS usando as bibliotecas do AWS SDK, que agrupam o subjacente AWS KMS REST API e simplificam suas tarefas de programação. Para obter informações sobre o AWS SDKs, incluindo como baixá-los e instalá-los, consulte [Ferramentas para criar AWS](#).

[Exemplos de código para AWS KMS usar AWS SDKs](#) Fornece um bom ponto de partida para uso AWS KMS por meio do AWS SDKs.

Usando esse serviço com um AWS SDK

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que permitem que os desenvolvedores criem facilmente aplicações em seu idioma de preferência.

Documentação do SDK	Exemplos de código
AWS SDK para C++	AWS SDK para C++ exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK para Go	AWS SDK para Go exemplos de código
AWS SDK para Java	AWS SDK para Java exemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código
AWS SDK para .NET	AWS SDK para .NET exemplos de código
AWS SDK para PHP	AWS SDK para PHP exemplos de código
Ferramentas da AWS para PowerShell	Ferramentas da AWS para PowerShell exemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) exemplos de código
AWS SDK para Ruby	AWS SDK para Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP exemplos de código
AWS SDK para Swift	AWS SDK para Swift exemplos de código

Exemplo de disponibilidade

Não consegue encontrar o que precisa? Solicite um exemplo de código usando o link [Fornecer feedback](#) na parte inferior desta página.

AWS Encryption SDK

AWS Encryption SDK É uma ferramenta para implementar a criptografia do lado do cliente em seu aplicativo. Ele não fornece acesso total ao KMS, mas se integra ou pode ser usado como um SDK independente sem fazer referência às chaves do KMS. AWS KMS As bibliotecas estão disponíveis para Java JavaScript, C, Python e outras linguagens de programação.

Para obter mais informações, consulte o [Guia do desenvolvedor do AWS Encryption SDK](#).

AWS KMS key políticas e políticas do IAM

AWS KMS consistência eventual

A AWS KMS API segue um modelo de [consistência eventual](#) devido à natureza distribuída do sistema. Como resultado, as alterações nos AWS KMS recursos podem não ser imediatamente visíveis para os comandos subsequentes que você executar.

Quando você realiza chamadas de AWS KMS API, pode haver um breve atraso até que a alteração esteja disponível por toda parte AWS KMS. Normalmente, a alteração leva menos de alguns segundos para se propagar por todo o sistema, mas, em alguns casos, pode levar vários minutos. Você poderá receber erros inesperados, como `NotFoundException` ou `InvalidStateException`, durante esse período. Por exemplo, AWS KMS pode retornar um `NotFoundException` se você ligar `GetParametersForImport` imediatamente após a `ligaçãoCreateKey`.

Recomendamos que você configure uma estratégia de repetição em seus AWS KMS clientes para repetir automaticamente as operações após um breve período de espera. Para obter mais informações, consulte o [comportamento de repetição](#) no Guia de referência de ferramentas AWS SDKs e ferramentas.

Para chamadas de API relacionadas a concessões, é possível [usar um token de concessão](#) para evitar possíveis atrasos e usar as permissões em uma concessão imediatamente. Para obter informações, consulte [Eventual consistency \(for grants\)](#) (Consistência eventual (para concessões)).

Usando TLS pós-quântico híbrido com AWS KMS

AWS Key Management Service (AWS KMS) suporta uma opção híbrida de troca de chaves pós-quântica para o protocolo de criptografia de rede Transport Layer Security (TLS). Você pode usar essa opção TLS ao se conectar aos endpoints de API do AWS KMS. Esses recursos opcionais de troca de chaves pós-quânticas híbridas são, no mínimo, tão seguros quanto a criptografia TLS que usamos atualmente e provavelmente fornecerão benefícios adicionais de segurança em longo prazo. No entanto, eles afetam a latência e o throughput em comparação com os protocolos clássicos de troca de chaves em uso atualmente.

Os dados que você envia para AWS Key Management Service (AWS KMS) são protegidos em trânsito pela criptografia fornecida por uma conexão TLS (Transport Layer Security). Os pacotes de criptografia clássica aos quais o AWS KMS oferece suporte a sessões TLS tornam inviáveis os ataques de força bruta nos mecanismos de troca de chaves com a tecnologia atual. No entanto, se a computação quântica em grande escala se tornar prática no futuro, os pacotes de criptografia clássica usados nos mecanismos de troca de chaves TLS serão suscetíveis a esses ataques. Se você estiver desenvolvendo aplicativos que dependem da confidencialidade de longo prazo dos dados transmitidos por uma conexão TLS, considere um plano para migrar para a criptografia pós-quântica antes que computadores quânticos em grande escala estejam disponíveis para uso. AWS está trabalhando para se preparar para esse futuro, e queremos que você também esteja bem preparado.

Para proteger os dados criptografados hoje contra possíveis ataques futuros, AWS está participando com a comunidade criptográfica no desenvolvimento de algoritmos quânticos resistentes ou pós-quânticos. Implementamos suítes de cifras híbridas pós-quânticas de troca de chaves AWS KMS que combinam elementos clássicos e pós-quânticos para garantir que sua conexão TLS seja pelo menos tão forte quanto seria com suítes de criptografia clássicas.

[Esses pacotes de criptografia híbrida estão disponíveis para uso em suas cargas de trabalho de produção na maioria. Regiões da AWS](#) No entanto, como as características de desempenho e os requisitos de largura de banda dos pacotes de criptografia híbrida são diferentes dos mecanismos clássicos de troca de chaves, recomendamos que você [os teste em suas chamadas de AWS KMS API](#) sob condições diferentes.

Feedback

Como sempre, agradecemos seu feedback e sua participação em nossos repositórios de código aberto. Gostaríamos especialmente de saber como sua infraestrutura interage com essa nova variante do tráfego TLS.

- Para fornecer feedback sobre esse tópico, use o link Feedback no canto superior direito desta página.
- Estamos desenvolvendo esses pacotes de criptografia híbrida em código aberto no [s2n-tls](#) repositório em. GitHub Para fornecer feedback sobre a usabilidade dos pacotes de criptografia ou compartilhar novas condições de teste ou resultados, [crie uma ocorrência](#) no repositório s2n-tls.
- Estamos escrevendo exemplos de código para usar o TLS pós-quântico híbrido AWS KMS [aws-kms-pq-tls-example](#) GitHub no repositório. Para fazer perguntas ou compartilhar ideias sobre como configurar seu cliente HTTP ou AWS KMS cliente para usar os pacotes de criptografia híbridos, [crie um problema](#) no repositório. aws-kms-pq-tls-example

Suportado Regiões da AWS

O Post-quantum TLS for AWS KMS está disponível em todos os Regiões da AWS que AWS KMS suportam.

Para obter uma lista de AWS KMS endpoints para cada um Região da AWS, consulte [AWS Key Management Service endpoints e cotas](#) no. Referência geral da Amazon Web Services Para obter informações sobre endpoints do FIPS, consulte [Endpoints do FIPS](#) no Referência geral da Amazon Web Services.

Sobre a troca de chaves pós-quântica híbrida no TLS

AWS KMS suporta suítes de cifras híbridas pós-quânticas de troca de chaves. Você pode usar o AWS SDK for Java 2.x e o AWS Common Runtime em sistemas Linux para configurar um cliente HTTP que usa esses conjuntos de criptografia. Então, sempre que você se conecta a um AWS KMS endpoint com seu cliente HTTP, os conjuntos de criptografia híbrida são usados.

Este cliente HTTP usa [s2n-tls](#), que é uma implementação de código aberto do protocolo TLS. Os pacotes de codificação híbridos usados por s2n-tls são implementados apenas para troca de chaves, não para a criptografia de dados direta. Durante a troca de chaves, o cliente e o servidor calculam a chave que usarão para criptografar e descriptografar os dados na rede.

Os algoritmos que o s2n-tls usa são um híbrido que combina o [Elliptic Curve Diffie-Hellman](#) (ECDH), um algoritmo clássico de troca de chaves atualmente usado no TLS, com o [Module-Lattice-Based Key-Encapsulation Mechanism](#) (ML-KEM), um algoritmo de criptografia de chave pública e estabelecimento de chaves que o National Institute for Standards and Technology (NIST) [designou como seu primeiro algoritmo de acordo de chaves](#) pós-quântico padrão. Esse híbrido utiliza cada um dos algoritmos de maneira independente para gerar uma chave. Depois, ele combina as duas

chaves criptograficamente. Com s2n-tls, é possível [configurar um cliente HTTP](#) para ter o TLS pós-quântico como preferência, o que define ECDH com ML-KEM em primeiro lugar na lista de preferências. Algoritmos clássicos de troca de chaves são incluídos na lista de preferências para garantir a compatibilidade, mas eles estarão mais abaixo na ordem de preferência.

Usando TLS pós-quântico híbrido com AWS KMS

Você pode usar o TLS pós-quântico híbrido para suas chamadas para o AWS KMS. Ao configurar seu ambiente de teste de cliente HTTP, esteja ciente das seguintes informações:

Criptografia em trânsito

Os pacotes de criptografia híbrida no s2n-tls são usados somente para a criptografia em trânsito. Eles protegem seus dados enquanto eles viajam do seu cliente para o AWS KMS endpoint. O AWS KMS não usa esses conjuntos de cifras para criptografar dados em AWS KMS keys.

Em vez disso, ao fazer o AWS KMS criptografar seus dados com chaves KMS, ele usa criptografia simétrica com chaves de 256 bits e o algoritmo Advanced Encryption Standard in Galois Counter Mode (AES-GCM), que já é resistente ao quantum. Num futuro teórico, ataques de computação quântica em grande escala em textos cifrados criados sob chaves AES-GCM de 256 bits [reduzem a segurança efetiva da chave para 128 bits](#). Esse nível de segurança é suficiente para inviabilizar ataques de força bruta em textos AWS KMS cifrados.

Sistemas suportados

O uso dos pacotes de criptografia híbrida no s2n-tls tem suporte atualmente somente em sistemas Linux. Além disso, esses pacotes de criptografia são suportados somente nos SDKs AWS Common Runtime, como o AWS SDK for Java 2.x. Para ver um exemplo, consulte [Configurar o TLS pós-quântico híbrido](#).

AWS KMS Endpoints

O AWS KMS suporta TLS híbrido pós-quântico em todos os endpoints, incluindo endpoints validados pelo [FIPS 140-3](#).

Configurar o TLS pós-quântico híbrido

Neste procedimento, adicione uma dependência do Maven para o AWS Common Runtime HTTP Client. Em seguida, configure um cliente HTTP com preferência em TLS pós-quântico. Em seguida, crie um AWS KMS cliente que use o cliente HTTP.

Para ver exemplos funcionais completos de configuração e uso do TLS pós-quântico híbrido com o AWS KMS, consulte o repositório [aws-kms-pq-tls-example](#).

Note

O AWS Common Runtime HTTP Client, que estava disponível como uma prévia, tornou-se disponível ao público em fevereiro de 2023. Nesse lançamento, a classe `TlsCipherPreference` e o parâmetro de método `tlsCipherPreference()` foram substituídos pelo parâmetro de método `postQuantumTlsEnabled()`. Se você estava usando este exemplo durante a prévia, precisará atualizar seu código.

1. Adicione o cliente AWS Common Runtime às suas dependências do Maven. Recomendamos usar a versão mais recente disponível.

Por exemplo, essa declaração adiciona a versão 2.30.22 do cliente AWS Common Runtime às suas dependências do Maven.

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

2. Para habilitar os pacotes híbridos de criptografia pós-quântica, adicione-os AWS SDK for Java 2.x ao seu projeto e inicialize-o. Em seguida, habilite os pacotes de codificação pós-quântica híbrida em seu cliente HTTP, conforme mostrado no exemplo a seguir.

Esse código usa o parâmetro de método `postQuantumTlsEnabled()` para configurar um [cliente HTTP de runtime da AWS](#) que prefere o conjunto de cifras híbridas pós-quânticas recomendado, ECDH com ML-KEM. Em seguida, ele usa o cliente HTTP configurado para criar uma instância do cliente AWS KMS assíncrono, [KmsAsyncClient](#). Após a conclusão desse código, todas as solicitações de [API do AWS KMS](#) na instância `KmsAsyncClient` usam TLS pós-quântico híbrido.

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

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

3. Teste suas AWS KMS chamadas com o TLS híbrido pós-quântico.

Quando você chama operações de AWS KMS API no AWS KMS cliente configurado, suas chamadas são transmitidas para o AWS KMS endpoint usando TLS híbrido pós-quântico. Para testar sua configuração, chame uma AWS KMS API, como [ListKeys](#).

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

Testar sua configuração TLS pós-quântico híbrido

Considere executar os testes a seguir com pacotes de criptografia híbridos em seus aplicativos que chamam AWS KMS.

- Veja a `tlsDetails` seção na entrada de CloudTrail registro de uma chamada de AWS KMS API feita pelo seu aplicativo. O `keyExchange` campo deve mencionar um algoritmo híbrido, como `X25519MLKEM768`. Para ver um exemplo, consulte [Descriptografe com uma chave de criptografia simétrica padrão em uma conexão TLS pós-quântica](#).
- Execute benchmarks usando TLS pós-quântico híbrido. A troca de chaves híbridas aumenta o tamanho e o tempo de processamento de algumas mensagens no handshake TLS, mas o impacto geral no desempenho deve ser imperceptível na maioria dos casos.
- Tente conectar-se de diferentes locais. Dependendo do caminho de rede que sua solicitação segue, você pode descobrir que hosts intermediários, proxies ou firewalls legados com inspeção profunda de pacotes (DPI) bloqueiam a solicitação. Isso pode resultar do uso dos novos grupos de troca de chaves na [ClientHello](#) parte do handshake TLS ou de mensagens maiores de troca de chaves. Se você tiver problemas para resolver esses problemas, trabalhe com sua equipe de segurança ou administradores de TI para atualizar a configuração relevante e desbloquear os novos grupos de troca de chaves TLS.

Saiba mais sobre o TLS pós-quântico em AWS KMS

Para obter mais informações sobre como usar o TLS híbrido pós-quântico em AWS KMS, consulte os recursos a seguir.

- Para aprender sobre criptografia pós-quântica em AWS, incluindo links para postagens de blogs e trabalhos de pesquisa, consulte Criptografia [pós-quântica](#).
- Para obter informações sobre o s2n-tls, consulte [Apresentação do s2n-tls, uma nova implementação do TLS de código aberto](#) e [Uso do s2n-tls](#).
- Para obter informações sobre o cliente HTTP do AWS Common Runtime, consulte [Configurando o cliente HTTP AWS baseado em CRT no Guia](#) do AWS SDK for Java 2.x desenvolvedor.
- Para obter informações sobre o projeto de criptografia pós-quântica no Instituto Nacional de Padrões e Tecnologia (NIST — National Institute for Standards and Technology), consulte [Criptografia pós-quântica](#).
- Para obter informações sobre a padronização de criptografia pós-quântica do NIST, consulte [Post-Quantum Cryptography Standardization](#) (Padronização de criptografia pós-quântica).

Conecte-se AWS KMS por meio de um VPC endpoint

Você pode se conectar diretamente AWS KMS por meio de um endpoint de interface privada em sua nuvem privada virtual (VPC). Quando você usa uma interface VPC endpoint, a comunicação entre sua VPC AWS KMS é conduzida inteiramente dentro da rede. AWS

AWS KMS suporta endpoints da Amazon Virtual Private Cloud (Amazon VPC) alimentados por. [AWS PrivateLink](#) Cada VPC endpoint é representado por uma ou mais [interfaces de rede elástica](#) (ENIs) com endereços IP privados em suas sub-redes VPC.

A interface VPC endpoint conecta sua VPC diretamente AWS KMS sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão. AWS Direct Connect As instâncias em sua VPC não precisam de endereços IP públicos para se comunicar. AWS KMS

Regiões

AWS KMS oferece suporte a endpoints de VPC e políticas de endpoints de VPC em tudo o que é compatível. Regiões da AWS [AWS KMS](#)

Considerações sobre AWS KMS VPC endpoints

Antes de configurar uma interface para o VPC endpoint AWS KMS, consulte o tópico [Propriedades e limitações do endpoint de interface](#) no Guia.AWS PrivateLink

AWS KMS o suporte para um VPC endpoint inclui o seguinte.

- Você pode usar o endpoint da VPC para chamar todas as [operações de API do AWS KMS](#) em sua VPC.
- [Você pode criar uma interface VPC endpoint que se conecta a um endpoint AWS KMS regional ou a um endpoint FIPS.AWS KMS](#)
- Você pode usar AWS CloudTrail registros para auditar o uso das chaves KMS por meio do VPC endpoint. Para obter detalhes, consulte [Registrar AWS KMS solicitações que usam um VPC endpoint](#).

Tópicos

- [Crie um VPC endpoint para AWS KMS](#)
- [Conecte-se a um AWS KMS VPC endpoint](#)
- [Use VPC endpoints para controlar o acesso aos recursos AWS KMS](#)
- [Registrar AWS KMS solicitações que usam um VPC endpoint](#)

Crie um VPC endpoint para AWS KMS

Você pode criar um VPC endpoint AWS KMS usando o console da Amazon VPC ou a API da Amazon VPC. Siga os procedimentos para [Criar um VPC endpoint](#) usando um dos seguintes valores.

- Para criar um VPC endpoint para AWS KMS, use o seguinte nome de serviço:

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

Por exemplo, na Região Oeste dos EUA (Oregon) (us-west-2), o nome do serviço seria:

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

- Para criar um endpoint da VPC na que se conecta a um [endpoint FIPS do AWS KMS](#), use o seguinte nome de serviço:

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

Por exemplo, na Região Oeste dos EUA (Oregon) (us-west-2), o nome do serviço seria:

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

Para facilitar o uso do endpoint da VPC, é possível habilitar um [nome de DNS privado](#) para seu endpoint da VPC. Se você selecionar a opção Enable DNS Name (Habilitar nome DNS), o nome de host DNS padrão do AWS KMS será resolvido para seu endpoint da VPC. Por exemplo, `https://kms.us-west-2.amazonaws.com` resolveria para um endpoint da VPC conectado ao nome de serviço `com.amazonaws.us-west-2.kms`.

Essa opção facilita usar o endpoint da VPC. O AWS SDKs e AWS CLI usa o nome de host AWS KMS DNS padrão por padrão, então você não precisa especificar a URL do VPC endpoint em aplicativos e comandos.

Para mais informações, consulte [Acessar um serviço por meio de um endpoint de interface](#) no Guia do AWS PrivateLink .

Conecte-se a um AWS KMS VPC endpoint

Você pode se conectar AWS KMS por meio do VPC endpoint usando um AWS SDK, o ou. AWS CLI Ferramentas da AWS para PowerShell Para especificar o endpoint da VPC, use seu nome de DNS.

Por exemplo, este comando [list-keys](#) usa o parâmetro `endpoint-url` para especificar o endpoint da VPC. Para usar um comando como este, substitua o exemplo de ID de endpoint da VPC na sua conta.

```
$ aws kms list-keys --endpoint-url https://vpce-1234abcdef5678c90a-09p7654s-us-east-1a.ec2.us-east-1.vpce.amazonaws.com
```

Permissões obrigatórias

Para que uma AWS KMS solicitação que usa um VPC endpoint seja bem-sucedida, o principal exige permissões de duas fontes:

- Uma [política de chaves](#), uma [política do IAM](#) ou uma [concessão](#) deve conceder à entidade principal permissão para chamar a operação no recurso (chave do KMS ou alias).
- Uma política de endpoint da VPC deve dar permissão à entidade principal para usar o endpoint para fazer a solicitação.

Por exemplo, uma política de chaves pode dar à entidade principal permissão para chamar [Decrypt](#) em uma chave do KMS específica. No entanto, a política de endpoint da VPC pode não permitir que a entidade principal chame Decrypt nessa chave do KMS usando o endpoint.

Ou uma política de VPC endpoint pode permitir que um principal use o endpoint para chamar [DisableKey](#) determinadas chaves do KMS. Porém, se a entidade principal não tiver essas

permissões de uma política de chaves, política do IAM ou concessão, ocorrerá falha na solicitação.

É possível criar uma política de endpoint da VPC ao criar seu endpoint e alterar a política de endpoint da VPC a qualquer momento. Use o console de gerenciamento da VPC [CreateVpcEndpoint](#) ou [ModifyVpcEndpoint](#) as operações. Você também pode criar e alterar uma política de VPC endpoint [usando](#) um modelo. AWS CloudFormation Para obter ajuda sobre o uso do console de gerenciamento da VPC, consulte [Criar um endpoint de interface](#) e [Modificar um endpoint de interface](#) no Guia do AWS PrivateLink .

Nomes de host privados

Se os nomes de host privados tiverem sido ativados ao criar o endpoint da VPC, você não precisa especificar o URL do endpoint da VPC nos comandos de CLI ou na configuração da aplicação. O nome de host DNS padrão do AWS KMS será resolvido para o endpoint da VPC. O AWS CLI e SDKs usa esse nome de host por padrão, para que você possa começar a usar o VPC endpoint para se conectar a AWS KMS um endpoint regional sem alterar nada em seus scripts e aplicativos.

Para usar nomes de host privados, os atributos `enableDnsHostnames` e `enableDnsSupport` da sua VPC devem ser definidos como `true`. Para definir esses atributos, use a [ModifyVpcAttribute](#) operação. Para mais detalhes, consulte [Exibir e atualizar atributos DNS para sua VPC](#) no Guia do usuário do Amazon VPC.

Use VPC endpoints para controlar o acesso aos recursos AWS KMS

Você pode controlar o acesso a AWS KMS recursos e operações quando a solicitação vem da VPC ou usa um VPC endpoint. Para fazer isso, use uma das [chaves de condição global](#) a seguir em uma [política de chave](#) ou [política do IAM](#).

- Use a chave de condição `aws:sourceVpce` para conceder ou restringir o acesso com base no endpoint da VPC.
- Use a chave de condição `aws:sourceVpc` para conceder ou restringir o acesso a uma VPC que hospedar o endpoint privado.

Note

Tome cuidado ao criar políticas de chaves e políticas do IAM com base no seu endpoint da VPC. Se uma declaração de política exigir que as solicitações venham de um determinado VPC ou VPC endpoint, as solicitações de AWS serviços integrados que usam um AWS KMS recurso em seu nome podem falhar. Para obter ajuda, consulte [Usando condições de endpoint VPC em políticas com permissões AWS KMS](#).

Além disso, a chave de condição `aws:sourceIP` não será efetiva se a solicitação vier de um [endpoint da Amazon VPC](#). Para restringir solicitações a um endpoint da VPC, use as chaves de condições `aws:sourceVpce` ou `aws:sourceVpc`. Para obter mais informações, consulte [Gerenciamento de identidade e acesso para VPC endpoints e serviços de VPC endpoint](#) no Guia do AWS PrivateLink .

Você pode usar essas chaves de condição globais para controlar o acesso a AWS KMS keys (chaves KMS), aliases e operações como essas [CreateKey](#) que não dependem de nenhum recurso específico.

Por exemplo, o exemplo de política de chave a seguir permite que um usuário execute algumas operações de criptografia com uma chave do KMS somente quando a solicitação usa o endpoint da VPC especificado. Quando um usuário faz uma solicitação para AWS KMS, o ID do VPC endpoint na solicitação é comparado ao valor da chave de `aws:sourceVpce` condição na política. Se não coincidirem, a solicitação será negada.

Para usar uma política como essa, substitua o Conta da AWS ID do espaço reservado e o VPC IDs endpoint por valores válidos para sua conta.

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:sourceVpc": "vpce-1234abcdef5678c90a"
        }
      }
    }
  ]
}

```

Você também pode usar a chave de condição `aws:sourceVpc` para restringir o acesso às suas chaves do KMS com base na VPC no qual o endpoint da VPC reside.

O exemplo de política de chave a seguir permite comandos que gerenciam a chave do KMS somente quando eles são provenientes do `vpc-12345678`. Além disso, ele permite comandos que usam a chave do KMS para operações de criptografia somente quando eles são provenientes de `vpc-2b2b2b2b`. Você pode usar uma política como essa se uma aplicação é executada em uma VPC, mas você usa uma segunda VPC isolada para funções de gerenciamento.

Para usar uma política como essa, substitua o Conta da AWS ID do espaço reservado e o VPC IDs endpoint por valores válidos para sua conta.

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": "*"
  }
]
}

```

Registrar AWS KMS solicitações que usam um VPC endpoint

AWS CloudTrail registra todas as operações que usam o VPC endpoint. Quando uma solicitação AWS KMS usa um VPC endpoint, o ID do VPC endpoint aparece na entrada de [AWS CloudTrail registro que registra](#) a solicitação. Você pode usar o ID do endpoint para auditar o uso do seu AWS KMS VPC endpoint.

No entanto, seus CloudTrail registros não incluem operações solicitadas por diretores em outras contas nem solicitações de AWS KMS operações em chaves e aliases do KMS em outras contas. Além disso, para proteger sua VPC, as solicitações que são negadas por uma política de endpoint da VPC, mas teriam sido permitidas de outra forma, não são registradas no [AWS CloudTrail](#).

Por exemplo, este exemplo de entrada de log registra uma solicitação de [GenerateDataKey](#) que usou o VPC endpoint. O campo `vpcEndpointId` aparece no final da entrada de 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-afcabbff2f04c",
"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",
"vpceEndpointId": "vpce-1234abcd5678c90a"
}

```

Suporte a endpoint de pilha dupla

AWS KMS fornece um endpoint público de pilha dupla que oferece suporte a ambos e aos clientes. IPv4 IPv6 Um endpoint de pilha dupla permite que os clientes se comuniquem AWS KMS usando um ou mais endereços. IPv4 IPv6 Para obter mais informações sobre os AWS KMS endpoints, consulte [AWS Key Management Service endpoints e cotas](#).

O endpoint público de AWS KMS pilha dupla `https://kms.your-region.api.aws` oferece suporte tanto para os clientes quanto para os clientes. IPv4 IPv6 AWS KMS também pode ser acessado de forma privada por IPv4 IPv6 meio de sua nuvem privada virtual (VPC) usando. AWS PrivateLink Para obter mais informações sobre a criação de endpoints VPC de interface privada para AWS KMS, consulte. [Conecte-se AWS KMS por meio de um VPC endpoint](#)

Para obter mais informações sobre IPv6 endereçamento para você VPCs, consulte [Como a Amazon VPC funciona](#) no Guia do usuário da Amazon Virtual Private Cloud. Para obter mais informações sobre como configurar sua VPC para o modo de pilha dupla, consulte o [endereçamento IP para você VPCs e suas sub-redes no Guia do usuário da Amazon Virtual Private Cloud](#).

Recursos não disponíveis acima IPv6

AWS KMS não pode se comunicar IPv6 com armazenamentos de AWS CloudHSM chaves ou armazenamentos de chaves externos. Essa limitação não impede que você ligue AWS KMS APIs IPv6.

AWS KMS conceitos

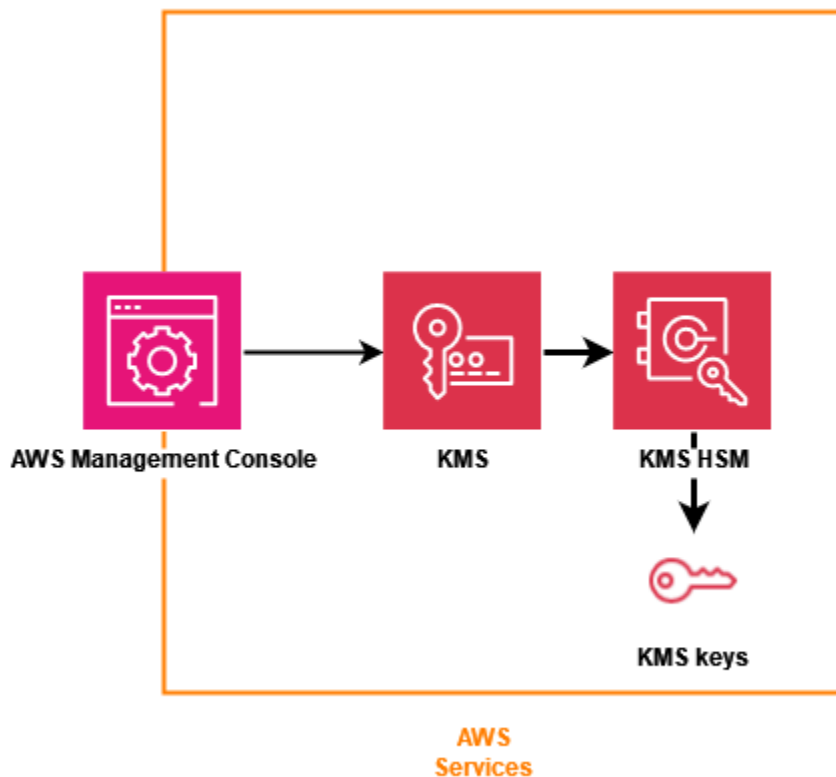
Aprenda os termos e conceitos básicos usados em AWS Key Management Service (AWS KMS) e como eles funcionam juntos para ajudar a proteger seus dados.

Introdução ao AWS KMS

AWS Key Management Service (AWS KMS) fornece uma interface web para gerar e gerenciar chaves criptográficas e opera como um provedor de serviços criptográficos para proteger dados. AWS KMS oferece serviços tradicionais de gerenciamento de chaves integrados aos AWS serviços para fornecer uma visão consistente das chaves dos clientes AWS, com gerenciamento e auditoria centralizados.

AWS KMS inclui uma interface web por meio da Console de gerenciamento da AWS interface de linha de comando e operações de RESTful API para solicitar operações criptográficas de uma frota distribuída de módulos de segurança de hardware validados pelo FIPS 140-3 (). HSMs O AWS KMS HSM é um dispositivo criptográfico de hardware autônomo com vários chips projetado para fornecer funções criptográficas dedicadas para atender aos requisitos de segurança e escalabilidade do. AWS KMS Você pode estabelecer sua própria hierarquia criptográfica baseada em HSM sob chaves gerenciadas como AWS KMS keys. Essas chaves são disponibilizadas somente no HSMs e somente na memória pelo tempo necessário para processar sua solicitação criptográfica. Você pode criar várias chaves KMS, cada uma representada por seu ID de chave. Somente nas funções e contas AWS do IAM administradas por cada cliente, as chaves KMS gerenciadas pelo cliente podem ser criadas, excluídas ou usadas para criptografar, descriptografar, assinar ou verificar dados. Você pode definir controles de acesso sobre quem pode gerenciar and/or o uso de chaves KMS criando uma política anexada à chave. Essas políticas permitem que você defina usos específicos de aplicação para suas chaves para cada operação de API.

Além disso, a maioria dos AWS serviços oferece suporte à criptografia de dados em repouso usando chaves KMS. Esse recurso permite que os clientes controlem como e quando AWS os serviços podem acessar dados criptografados, controlando como e quando as chaves KMS podem ser acessadas.



AWS KMS é um serviço em camadas que consiste em AWS KMS hosts voltados para a web e um nível de HSMs. O agrupamento desses hosts em camadas forma a AWS KMS pilha. Todas as solicitações AWS KMS devem ser feitas pelo protocolo Transport Layer Security (TLS) e terminar em um AWS KMS host. AWS KMS [os hosts só permitem TLS com um conjunto de cifras que fornece sigilo direto perfeito](#). AWS KMS autentica e autoriza suas solicitações usando os mesmos mecanismos de credenciais e políticas do AWS Identity and Access Management (IAM) que estão disponíveis para todas as outras AWS operações de API.

AWS KMS metas de design

AWS KMS foi projetado para atender aos seguintes requisitos.

Durabilidade

A durabilidade das chaves criptográficas foi projetada para ser igual à dos serviços de maior durabilidade em AWS. Uma única chave criptográfica pode criptografar grandes volumes dos seus dados acumulados ao longo de um longo período de tempo.

Confiável

O uso de chaves é protegido por políticas de controle de acesso que você define e gerencia. Não há mecanismo para exportar chaves KMS de texto simples. A confidencialidade das suas chaves criptográficas é crucial. Vários funcionários da Amazon com acesso específico por função aos controles de acesso baseados em quórum são obrigados a realizar ações administrativas no. HSMs

Baixa latência e alto throughput

AWS KMS fornece operações criptográficas em níveis de latência e taxa de transferência adequados para uso por outros serviços em. AWS

Regiões independentes

AWS fornece regiões independentes para clientes que precisam restringir o acesso aos dados em diferentes regiões. O uso da chave pode ser isolado dentro de um Região da AWS.

Fonte segura de números aleatórios

Como a criptografia forte depende da geração de números aleatórios verdadeiramente imprevisível, AWS KMS fornece uma fonte validada e de alta qualidade de números aleatórios.

Auditoria

AWS KMS registra o uso e o gerenciamento de chaves criptográficas em AWS CloudTrail registros. Você pode usar AWS CloudTrail registros para inspecionar o uso de suas chaves criptográficas, incluindo o uso de chaves por AWS serviços em seu nome.

Para atingir esses objetivos, o AWS KMS sistema inclui um conjunto de AWS KMS operadores e operadores de hospedagem de serviços (coletivamente, “operadores”) que administram “domínios”. Um domínio é um conjunto de AWS KMS servidores e operadores definidos regionalmente. HSMs Cada AWS KMS operador tem um token de hardware que contém um par de chaves pública e privada que é usado para autenticar suas ações. Eles HSMs têm um par adicional de chaves públicas e privadas para estabelecer chaves de criptografia que protejam a sincronização do estado do HSM.

AWS KMS keys

As chaves do KMS que você cria e gerencia para uso em suas próprias aplicações criptográficas são de um tipo conhecido como chaves gerenciadas pelo cliente. As chaves gerenciadas pelo cliente também podem ser usadas em conjunto com AWS serviços que usam chaves KMS para

criptografar os dados que o serviço armazena em seu nome. As chaves gerenciadas pelo cliente são recomendadas para clientes que desejam controle total sobre o ciclo de vida e o uso de suas chaves. Há um custo mensal para ter uma chave gerenciada pelo cliente em sua conta. Além disso, as solicitações de uso para and/or gerenciar a chave incorrem em um custo de uso. Para obter mais detalhes, consulte [Preços do AWS Key Management Service](#).

Há casos em que um cliente pode querer um AWS serviço para criptografar seus dados, mas não quer a sobrecarga de gerenciar chaves e não quer pagar por uma chave. Uma Chave gerenciada pela AWS é uma chave do KMS que existe na sua conta, mas que só pode ser usada sob determinadas circunstâncias. Especificamente, ela só pode ser usada no contexto do AWS serviço em que você está operando e só pode ser usada pelos diretores da conta em que a chave existe. Você não pode gerenciar nada sobre o ciclo de vida ou as permissões dessas chaves. Ao usar recursos de criptografia nos AWS serviços, você pode ver Chaves gerenciadas pela AWS; eles usam um alias no formato “aws<service code>”. Por exemplo, uma chave do aws/ebs só pode ser usada para criptografar volumes do EBS e somente para volumes usados por entidades principais do IAM na mesma conta da chave. Pense em um Chave gerenciada pela AWS que tenha um escopo reduzido para uso somente por usuários em sua conta para recursos em sua conta. Você não pode compartilhar recursos criptografados sob e Chave gerenciada pela AWS com outras contas. Embora a existência de um Chave gerenciada pela AWS seja gratuita em sua conta, você é cobrado por qualquer uso desse tipo de chave pelo AWS serviço atribuído à chave.

Chaves gerenciadas pela AWS são um tipo de chave herdado que não está mais sendo criado para novos AWS serviços a partir de 2021. Em vez disso, AWS serviços novos (e antigos) estão usando o que é conhecido como an Chave pertencente à AWS para criptografar os dados do cliente por padrão. An Chave pertencente à AWS é uma chave KMS que está em uma conta gerenciada pelo AWS serviço, para que os operadores do serviço tenham a capacidade de gerenciar seu ciclo de vida e permissões de uso. Ao usar Chaves pertencentes à AWS, AWS os serviços podem criptografar seus dados de forma transparente e permitir o fácil compartilhamento de dados entre contas ou regiões sem que você precise se preocupar com as principais permissões. Use Chaves pertencentes à AWS para encryption-by-default cargas de trabalho que fornecem proteção de dados mais fácil e automatizada. Como essas chaves pertencem e são gerenciadas por AWS, você não é cobrado pela existência ou uso delas, não pode alterar suas políticas, não pode auditar atividades nessas chaves e não pode excluí-las. Use chaves gerenciadas pelo cliente quando o controle for importante, mas use Chaves pertencentes à AWS quando a conveniência for mais importante.

Chaves gerenciadas pelo cliente	Chaves gerenciadas pela AWS	Chaves pertencentes à AWS

Política de chave	Controlada exclusivamente pelo cliente	Controlada pelo serviço; visível para o cliente	Controlado exclusivamente e visível apenas pelo AWS serviço que criptografa seus dados
Registro em log	CloudTrail rastreamento de clientes ou armazenamento de dados de eventos	CloudTrail rastreamento de clientes ou armazenamento de dados de eventos	Não visível para o cliente
Gerenciamento de ciclo de vida	O cliente gerencia a alternância, a exclusão e a localização regional	AWS KMS gerencia a rotação (anual), a exclusão e a localização regional	AWS service (Serviço da AWS) gerencia rotação, exclusão e localização regional
Preços	Tarifa mensal por existência das chaves (pro rata por hora). Também há cobrança pelo uso da chave	Sem taxa mensal; mas o chamador é cobrado pelo uso da API nessas chaves	Sem cobranças para o cliente

As chaves do KMS que você cria são [chaves gerenciadas pelo cliente](#). Os Serviços da AWS que usam chaves do KMS para criptografar seus recursos dos serviços geralmente criam chaves para você. As chaves KMS Serviços da AWS criadas em sua AWS conta são [Chaves gerenciadas pela AWS](#). As chaves KMS Serviços da AWS criadas em uma conta de serviço são [Chaves pertencentes à AWS](#).

Tipo de chave do KMS	Pode visualizar metadados da chave do KMS	Pode gerenciar a chave do KMS	Usado apenas para o meu Conta da AWS	Rotação automática	Preços
Chave gerenciada pelo cliente	Sim	Sim	Sim	Opcional.	Tarifa mensal (proporcional por hora)

Tipo de chave do KMS	Pode visualizar metadados da chave do KMS	Pode gerenciar a chave do KMS	Usado apenas para o meu Conta da AWS	Rotação automática	Preços
					Tarifa por uso
Chave gerenciada pela AWS	Sim	Não	Sim	Obrigatório. A cada ano (aproximadamente 365 dias).	Nenhuma tarifa mensal Taxa por uso (alguns Serviços da AWS pagam essa taxa para você)
Chave pertencente à AWS	Não	Não	Não	O AWS service (Serviço da AWS) gerencia a estratégia de rotação	Nenhuma tarifa

[AWS os serviços que se integram AWS KMS](#) diferem em seu suporte às chaves KMS. Alguns AWS serviços criptografam seus dados por padrão com um Chave pertencente à AWS ou um Chave gerenciada pela AWS. Alguns AWS serviços oferecem suporte a chaves gerenciadas pelo cliente. Outros AWS serviços oferecem suporte a todos os tipos de chaves KMS para permitir a facilidade Chave pertencente à AWS, a visibilidade de uma Chave gerenciada pela AWS chave ou o controle de uma chave gerenciada pelo cliente. Para obter informações detalhadas sobre as opções de criptografia que um AWS serviço oferece, consulte o tópico Criptografia em repouso no guia do usuário ou no guia do desenvolvedor do serviço.

Chaves gerenciadas pelo cliente

As chaves do KMS que você cria são chaves gerenciadas pelo cliente. As chaves gerenciadas pelo cliente são chaves KMS Conta da AWS que você cria, possui e gerencia. Você tem controle total sobre essas chaves do KMS, incluindo estabelecer e manter suas [políticas de chaves](#), [políticas do IAM e concessões](#), além de [habilitá-las e desabilitá-las](#), [alternar seu material criptográfico](#), [adicionar etiquetas](#), [criar aliases](#) que fazem referência às chaves do KMS e [programar a exclusão de chaves do KMS](#).

Chaves gerenciadas pelo cliente são exibidas na página Chaves gerenciadas pelo cliente do Console de gerenciamento da AWS para o AWS KMS. Para identificar definitivamente uma chave gerenciada pelo cliente, use a operação [DescribeKey](#). Para chaves gerenciadas pelo cliente, o valor do campo KeyManager da resposta DescribeKey é CUSTOMER.

Você pode usar suas chaves gerenciadas pelo cliente em operações de criptografia e auditar o uso em logs do AWS CloudTrail . Além disso, muitos [serviços da AWS que se integram com o AWS KMS](#) permitem que você especifique uma chave gerenciada pelo cliente para proteger os dados armazenados e gerenciados.

Chaves gerenciadas pelo cliente geram uma taxa mensal e uma taxa para uso que excede o nível gratuito. Eles são contabilizados de acordo com as AWS KMS [cotas](#) da sua conta. Para obter mais detalhes, consulte [AWS Key Management Service Definição de preço](#) e [Cotas](#).

Chaves gerenciadas pela AWS

Chaves gerenciadas pela AWS são chaves KMS em sua conta que são criadas, gerenciadas e usadas em seu nome por um [AWS serviço integrado a. AWS KMS](#)

Alguns AWS serviços permitem que você escolha uma chave gerenciada pelo cliente Chave gerenciada pela AWS ou uma chave gerenciada pelo cliente para proteger seus recursos nesse serviço. Em geral, a menos que você precise controlar a chave de criptografia que protege seus recursos, uma Chave gerenciada pela AWS é uma boa escolha. Você não precisa criar ou manter a chave ou sua política de chave, e nenhuma tarifa mensal é necessária para uma Chave gerenciada pela AWS.

Você tem permissão para [visualizá-los Chaves gerenciadas pela AWS](#) em sua conta, [visualizar suas principais políticas](#) e [auditar seu uso](#) nos AWS CloudTrail registros. No entanto, você não pode alterar nenhuma propriedade Chaves gerenciadas pela AWS, alterná-las, alterar suas políticas de

chaves ou programá-las para exclusão. E você não pode usar diretamente Chaves gerenciadas pela AWS em operações criptográficas; o serviço que as cria as usa em seu nome.

As [políticas de controle de recursos](#) de sua organização não se aplicam a Chaves gerenciadas pela AWS.

Chaves gerenciadas pela AWS aparecem na Chaves gerenciadas pela AWS página do Console de gerenciamento da AWS formulário AWS KMS. Você também pode se identificar Chaves gerenciadas pela AWS por seus aliases, que têm o formato `aws/service-name`, como `aws/redshift`. Para identificar definitivamente um Chaves gerenciadas pela AWS, use a [DescribeKey](#) operação. Para Chaves gerenciadas pela AWS, o valor do campo `KeyManager` da resposta `DescribeKey` é `AWS`.

Todos Chaves gerenciadas pela AWS são alternados automaticamente todos os anos. alternar Não é possível alterar essa programação de rotação.

Note

Em maio de 2022, AWS KMS alterou o cronograma de rotação Chaves gerenciadas pela AWS de três em três anos (aproximadamente 1.095 dias) para todos os anos (aproximadamente 365 dias).

Não há taxa mensal para Chaves gerenciadas pela AWS. Eles podem estar sujeitos a taxas de uso além do nível gratuito, mas alguns AWS serviços cobrem esses custos para você. Para obter detalhes, consulte o tópico Criptografia em repouso, no manual do usuário ou no guia do desenvolvedor do serviço. Para obter mais informações, consulte [Definição de preço do AWS Key Management Service](#).

Chaves gerenciadas pela AWS não conte com as cotas de recursos o número de chaves KMS em cada região da sua conta. No entanto, quando usadas em nome de uma entidade principal na sua conta, essas chaves do KMS contam para as cotas de solicitações. Para obter detalhes, consulte [Cotas](#).

Chaves pertencentes à AWS

Chaves pertencentes à AWS são uma coleção de chaves KMS que um AWS serviço possui e gerencia para uso em várias Contas da AWS. Embora não Chaves pertencentes à AWS estejam na sua Conta da AWS, um AWS serviço pode usar um Chave pertencente à AWS para proteger os recursos da sua conta.

Alguns AWS serviços permitem que você escolha uma chave gerenciada pelo cliente Chave pertencente à AWS ou uma chave gerenciada pelo cliente. Em geral, a menos que você precise auditar ou controlar a chave de criptografia que protege seus recursos, uma Chave pertencente à AWS é uma boa escolha. Chaves pertencentes à AWS são totalmente gratuitos (sem taxas mensais ou taxas de uso), não contam nas [AWS KMS cotas](#) da sua conta e são fáceis de usar. Você não precisa criar ou manter a chave ou sua política de chave.

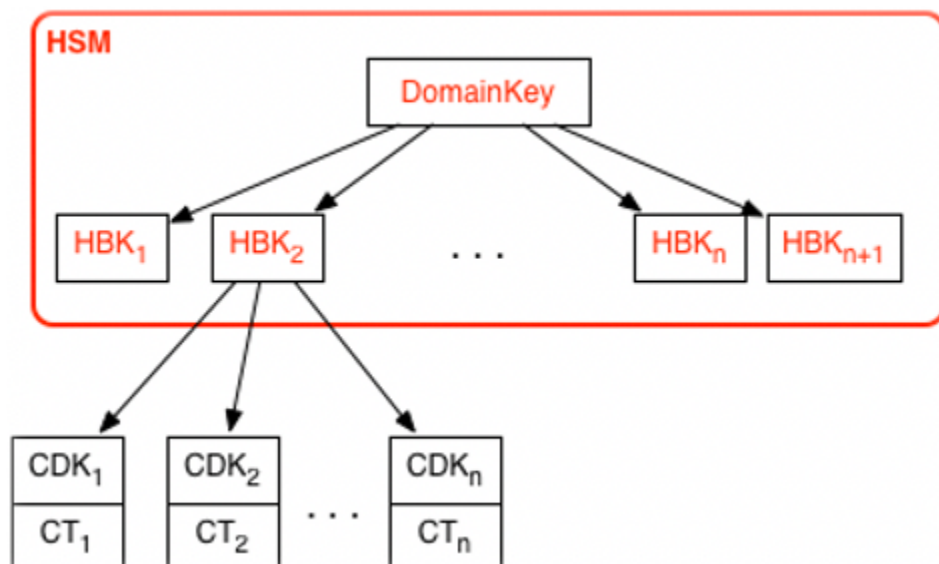
A rotação de Chaves pertencentes à AWS varia entre os serviços. Para obter informações sobre a rotação de um determinado serviço Chave pertencente à AWS, consulte o tópico Criptografia em repouso no guia do usuário ou no guia do desenvolvedor do serviço.

AWS KMS key hierarquia

Sua hierarquia de chaves começa com uma chave lógica de nível superior, uma. AWS KMS key Uma chave KMS representa um contêiner para o material de chave de nível superior e é definida exclusivamente dentro do namespace de serviço AWS com um Nome do Recurso Amazon (ARN). O ARN inclui um identificador de chave gerado exclusivamente, um ID de chave. Uma chave KMS é criada com base em uma solicitação iniciada pelo usuário por meio de. AWS KMS Após a recepção, AWS KMS solicita a criação de uma chave de apoio inicial do HSM (HBK) para ser colocada no contêiner da chave KMS. O HBK é gerado em um HSM no domínio e foi projetado para nunca ser exportado do HSM em texto simples. Em vez disso, o HBK é exportado criptografado em chaves de domínio gerenciadas por HSM. Esses exportados HBKs são chamados de tokens de chave exportados (EKTs).

O EKT é exportado para um armazenamento altamente durável e de baixa latência. Por exemplo, suponha que você receba um ARN para a chave KMS lógica. Para você, isso representa o topo de uma hierarquia de chaves, ou contexto criptográfico. Você pode criar várias chaves KMS em sua conta e definir políticas para suas chaves KMS como qualquer outro recurso AWS nomeado.

Dentro da hierarquia de uma chave KMS específica, o HBK pode ser considerado uma versão da chave KMS. Quando você deseja alternar a chave KMS AWS KMS, um novo HBK é criado e associado à chave KMS como o HBK ativo para a chave KMS. Os mais antigos HBKs são preservados e podem ser usados para descriptografar e verificar dados previamente protegidos. Mas somente a chave criptográfica ativa pode ser usada para proteger novas informações.



Você pode fazer solicitações AWS KMS para usar suas chaves KMS para proteger diretamente as informações ou solicitar chaves adicionais geradas pelo HSM que estejam protegidas por sua chave KMS. Essas chaves são chamadas de chaves de dados do cliente ou CDKs. CDKs podem ser retornados criptografados como texto cifrado (CT), em texto simples ou ambos. Todos os objetos criptografados em uma chave KMS (dados fornecidos pelo cliente ou chaves geradas pelo HSM) só podem ser descriptografados em um HSM por meio de uma chamada. AWS KMS

O texto cifrado retornado, ou a carga útil descriptografada, nunca é armazenado nele. AWS KMS As informações são retornadas a você através da sua conexão TLS com o AWS KMS. Isso também se aplica às chamadas feitas pelos AWS serviços em seu nome.

A hierarquia de chaves e as propriedades específicas de chave aparecem na tabela a seguir.

Chave	Description	Ciclo de vida
Chave de domínio	Uma chave AES-GCM de 256 bits somente na memória de um HSM usado para quebrar versões das chaves KMS, as chaves de reserva HSM.	Alternado diariamente ¹
Chave de reserva HSM	Uma chave simétrica de 256 bits ou chave privada RSA ou curva elíptica, usada para proteger dados e chaves do cliente e armazenada criptogra	Alternadas anualmente ² (configuração opcional)

Chave	Description	Ciclo de vida
	fada sob chaves de domínio. Uma ou mais chaves de apoio HSM formam a chave KMS, representada pelo keyID.	
Chave de criptografia derivada	Uma chave AES-GCM de 256 bits somente na memória de um HSM usado para criptografar dados e chaves do cliente. Derivado de um HBK para cada criptografia.	Usado uma vez por criptografia e regenerado ao descriptografar
Chave de dados do cliente	Chave simétrica ou assimétrica definida pelo usuário exportada do HSM em texto simples e texto cifrado. Criptografado com uma chave de reserva HSM e retornado aos usuários autorizados pelo canal TLS.	Alternância e uso controlados por aplicação

De tempos em tempos, é AWS KMS possível reduzir a rotação da chave de domínio para, no máximo, uma vez por semana, para contabilizar as tarefas de administração e configuração do domínio.

² Os padrões Chaves gerenciadas pela AWS criados e gerenciados por AWS KMS em seu nome são alternados automaticamente anualmente.

Identificadores-chave () KeyId

Identificadores de chave atuam como nomes para suas chaves do KMS. Eles ajudam a reconhecer suas chaves do KMS no console. Você poderá usá-las para indicar quais chaves do KMS deseja usar em operações de API do AWS KMS, políticas do IAM e concessões. Os valores do identificador de chave não estão completamente relacionados ao material de chaves associado à chave KMS.

AWS KMS define vários identificadores-chave. Quando você cria uma chave KMS, AWS KMS gera um ARN e ID da chave, que são propriedades da chave KMS. Quando você cria um [alias](#), AWS KMS gera um ARN de alias com base no nome do alias que você define. Você pode ver os identificadores de chave e alias na Console de gerenciamento da AWS e na AWS KMS API.

No AWS KMS console, você pode visualizar e filtrar as chaves KMS pelo ARN da chave, ID da chave ou nome do alias e classificar por ID da chave e nome do alias. Para obter ajuda para encontrar os identificadores de chave no console, consulte [the section called “Encontrar o ID e o ARN da chave”](#).

Na AWS KMS API, os parâmetros que você usa para identificar uma chave KMS são nomeados `KeyId` ou uma variação, como `TargetKeyId` ou `DestinationKeyId`. No entanto, os valores desses parâmetros não estão limitados à chave IDs. Alguns podem usar qualquer identificador de chave válido. Para obter informações sobre os valores de cada parâmetro, consulte a descrição do parâmetro na Referência da AWS Key Management Service API.

Note

Ao usar a AWS KMS API, tenha cuidado com o identificador de chave que você usa. Diferentes APIs exigem identificadores de chave diferentes. Em geral, use o identificador de chave mais completo e prático para a sua tarefa.

AWS KMS suporta os seguintes identificadores-chave.

ARN de chave

O ARN de chave é o Amazon Resource Name (ARN) de uma chave do KMS. É um identificador exclusivo e totalmente qualificado para a chave do KMS. O ARN da chave inclui a Conta da AWS região e o ID da chave. Para obter ajuda para encontrar o ARN de chave de uma chave do KMS, consulte [the section called “Encontrar o ID e o ARN da chave”](#).

O formato de um ARN de chave é o seguinte:

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

Veja a seguir um exemplo de ARN de chave para uma chave do KMS de região única.

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

O *key-id* elemento da chave ARNs das [chaves multirregionais](#) começa com o `mrk-` prefixo. Veja a seguir um exemplo de ARN de chave para uma chave do KMS de várias regiões.

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

ID da chave

O ID de chave identifica exclusivamente uma chave do KMS dentro de uma conta e uma região. Para obter ajuda para encontrar o ID de chave de uma chave do KMS, consulte [the section called “Encontrar o ID e o ARN da chave”](#).

Veja a seguir um exemplo de ID de chave para uma chave do KMS de região única.

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

A chave IDs das [chaves multirregionais](#) começa com o `mrk-` prefixo. Veja a seguir um exemplo de ID de chave para uma chave do KMS de várias regiões.

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

ARN do alias

O alias ARN é o Amazon Resource Name (ARN) de um alias. AWS KMS É um identificador exclusivo e totalmente qualificado para o alias e para a chave do KMS que ele representa. Um ARN de alias inclui Conta da AWS a Região e o nome do alias.

A qualquer momento, um ARN de alias identificará uma chave do KMS específica. No entanto, como é possível alterar a chaves do KMS associada ao alias, o ARN do alias pode identificar diferentes chaves do KMS em momentos diferentes. Para obter ajuda para encontrar o ARN de alias de uma chave do KMS, consulte [Encontrar o nome e o ARN do alias para uma chave do KMS](#).

O formato de um ARN de alias é o seguinte:

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

Veja a seguir o ARN de alias de um fictício ExampleAlias.

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

Nome do alias

O nome do alias é uma string com até 256 caracteres. Ele identifica exclusivamente uma chave do KMS associada em uma conta e uma região. Na AWS KMS API, os nomes de alias sempre começam com `alias/`. Para obter ajuda para encontrar o nome de alias de uma chave do KMS, consulte [Encontrar o nome e o ARN do alias para uma chave do KMS](#).

O formato de um nome de alias é o seguinte:

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

Por exemplo:

```
alias/ExampleAlias
```

O prefixo `aws/` de um nome de alias é reservado para [Chaves gerenciadas pela AWS](#). Não é possível criar um alias com esse prefixo. Por exemplo, o nome de alias do Chave gerenciada pela AWS Amazon Simple Storage Service (Amazon S3) é o seguinte.

```
alias/aws/s3
```

Chaves assimétricas em AWS KMS

Uma chave do KMS assimétrica representa um par de chave pública e chave privada matematicamente relacionadas. É possível dar a chave privada para qualquer pessoa, mesmo se ela não for confiável, mas a chave privada deve ser mantida em segredo.

Em uma chave KMS assimétrica, a chave privada é criada AWS KMS e nunca sai sem criptografia. AWS KMS Para usar a chave privada, você deve ligar AWS KMS. Você pode usar a chave pública interna AWS KMS chamando as operações AWS KMS da API. Ou você pode [baixar a chave pública](#) e usá-la fora do AWS KMS.

Se seu caso de uso exigir criptografia externa AWS por usuários que não podem ligar AWS KMS, as chaves KMS assimétricas são uma boa opção. No entanto, se você estiver criando uma chave KMS para criptografar os dados que você armazena ou gerencia em um AWS serviço, use uma chave KMS de criptografia simétrica. [AWS os serviços integrados AWS KMS](#) usam somente chaves KMS de criptografia simétrica para criptografar seus dados. Esses serviços não fornecem suporte para criptografia com chaves do KMS assimétricas.

Ao assinar mensagens maiores que 4 KB com AWS KMS, você deve fazer o hash da mensagem fora de AWS KMS antes de assinar. AWS KMS fornece três `MessageType` opções para lidar com a entrada de mensagens: `RAW` para mensagens de texto simples (onde AWS KMS executa o hash), `DIGEST` para mensagens pré-criptografadas (onde AWS KMS pula a etapa de hash) e, `EXTERNAL_MU` especificamente, para especificações de chave KMS do ML-DSA em que

a entrada é um valor μ representativo de 64 bytes. [Para mensagens grandes que excedam o limite de 4 KB, faça o hash da mensagem externamente e use MessageType : DIGEST \(ou MessageType : EXTERNAL_MU para chaves KMS ML-DSA\)](#) ao chamar as operações de Assinar e Verificar. [AWS KMS](#)

AWS KMS suporta vários tipos de chaves KMS assimétricas.

Chaves do KMS RSA

Uma chave KMS com um par de chaves RSA para criptografia e descryptografia ou assinatura e verificação (mas não ambas). AWS KMS suporta vários comprimentos de chave para diferentes requisitos de segurança.

Para obter detalhes técnicos sobre os algoritmos de criptografia e assinatura que oferecem AWS KMS suporte às chaves RSA KMS, consulte as especificações da chave [RSA](#).

Chaves do KMS de curva elíptica (ECC)

Uma chave KMS com um par de chaves de curva elíptica para assinatura e verificação ou derivação de segredos compartilhados (mas não ambos). AWS KMS suporta várias curvas comumente usadas.

Para obter detalhes técnicos sobre os algoritmos de assinatura que oferecem AWS KMS suporte às chaves ECC KMS, consulte Especificações da chave da curva [elíptica](#).

Chaves do KMS ML-DSA

Uma chave do KMS com um par de chaves ML-DSA para assinatura e verificação. O ML-DSA é um padrão de criptografia pós-quântica desenvolvido pelo US National Institute of Standards and Technology (NIST) para proteger contra as ameaças à segurança representadas pela computação quântica. O ML-DSA é o algoritmo de assinatura digital recomendado para organizações que estão fazendo a transição dos algoritmos de assinatura digital RSA ou Elliptic Curve para a criptografia segura pós-quântica.

AWS KMS suporta vários comprimentos de chave para diferentes requisitos de segurança. [Para obter detalhes técnicos sobre os algoritmos de assinatura que oferecem AWS KMS suporte às chaves KMS ML-DSA, consulte Especificação da chave ML-DSA.](#)

SM2 Chaves KMS (somente regiões da China)

Uma chave KMS com um par de SM2 chaves para criptografia e descryptografia, assinatura e verificação ou derivação de segredos compartilhados (você deve escolher um tipo). [Key usage](#)

Para obter detalhes técnicos sobre os algoritmos de criptografia e assinatura AWS KMS compatíveis com chaves SM2 KMS (somente regiões da China), consulte as [especificações da SM2 chave](#).

Para obter ajuda na escolha da sua configuração de chave assimétrica, consulte [Escolher qual tipo de chave do KMS criar](#).

Regiões

As chaves KMS assimétricas e os pares de chaves de dados assimétricos são aceitos em todos os suportes. Regiões da AWS AWS KMS

Saiba mais

- Para criar chaves do KMS assimétricas, consulte [Criar uma chave do KMS assimétrica](#).
- Para criar chaves do KMS assimétricas de várias regiões, consulte [Criar chaves primárias de várias regiões](#).
- Para saber como assinar mensagens e verificar assinaturas com chaves do KMS assimétricas, consulte [Assinatura digital com o novo recurso de chaves assimétricas do AWS KMS](#) no Blog de segurança da AWS .
- Para saber mais sobre considerações especiais sobre a exclusão de chaves do KMS assimétricas, consulte [Deleting asymmetric KMS keys](#).
- Para identificar e visualizar chaves do KMS assimétricas, consulte [Identificar chaves do KMS assimétricas](#).

Chaves HMAC em AWS KMS

As chaves KMS do Código de Autenticação de Mensagens Baseado em Hash (HMAC) são chaves simétricas que você usa para gerar e verificar dentro delas. HMACs AWS KMS O material exclusivo de chave associado a cada chave do KMS de HMAC fornece a chave secreta exigida pelos algoritmos de HMAC. Você pode usar uma chave do KMS de HMAC com as operações [GenerateMac](#) e [VerifyMac](#) para verificar a integridade e a autenticidade dos dados no AWS KMS.

Os algoritmos de HMAC combinam uma função de hash criptográfica e uma chave secreta compartilhada. Eles pegam uma mensagem e uma chave secreta, como o material da chave em uma chave do KMS de HMAC, e retornam um código ou etiqueta exclusiva e de tamanho fixo. Se até mesmo um caractere da mensagem mudar, ou se a chave secreta não for idêntica, a

etiqueta resultante será totalmente diferente. Ao exigir uma chave secreta, o HMAC também fornece autenticidade; é impossível gerar uma tag HMAC idêntica sem a chave secreta. HMACs às vezes são chamadas de assinaturas simétricas, porque funcionam como assinaturas digitais, mas usam uma única chave para assinatura e verificação.

[As chaves HMAC KMS e os algoritmos HMAC AWS KMS usados estão em conformidade com os padrões do setor definidos na RFC 2104.](#) A AWS KMS [GenerateMac](#) operação gera tags HMAC padrão. As chaves HMAC KMS são geradas em módulos de segurança de AWS KMS hardware certificados pelo [Programa de Validação de Módulos Criptográficos FIPS 140-3](#) (exceto nas regiões da China (Pequim) e China (Ningxia)) e nunca saem sem criptografia. AWS KMS Para usar uma chave do KMS de HMAC, é necessário chamar o AWS KMS.

Você pode usar chaves do KMS de HMAC para determinar a autenticidade de uma mensagem, como um JSON Web Token (JWT), informações tokenizadas de cartão de crédito ou uma senha enviada. Eles também podem ser usados como Funções de Derivação de Chave (KDFs) seguras, especialmente em aplicativos que exigem chaves determinísticas.

As chaves HMAC KMS oferecem uma vantagem em relação ao HMACs software aplicativo porque o material da chave é gerado e usado inteiramente internamente AWS KMS, sujeito aos controles de acesso que você define na chave.

Tip

As práticas recomendadas sugerem que você limite o tempo durante o qual qualquer mecanismo de assinatura, inclusive um HMAC, permanece vigente. Isso dissuade um ataque no qual o protagonista usa uma mensagem assinada para estabelecer a validade repetidamente ou muito depois que a mensagem é substituída. As etiquetas de HMAC não incluem um carimbo de data/hora, mas você pode incluir um carimbo de data/hora no token ou na mensagem para ajudar você a detectar quando está na hora de atualizar o HMAC.

Operações criptográficas compatíveis

As chaves do KMS de HMAC só são compatíveis com as operações criptográficas [GenerateMac](#) e [VerifyMac](#). Você não pode usar chaves do KMS de HMAC para criptografar dados ou assinar mensagens, nem usar nenhum outro tipo de chave do KMS em operações de HMAC. Ao usar a operação [GenerateMac](#), você fornece uma mensagem de até 4.096 bytes, uma chave do KMS de HMAC e o algoritmo de Message authentication code (MAC – Código de autenticação

de mensagem) compatível com a especificação de chave de HMAC, e GenerateMac calcula a etiqueta de HMAC. Para verificar uma etiqueta de HMAC, você deve fornecer a etiqueta de HMAC e a mesma mensagem, a chave do KMS de HMAC e o algoritmo de MAC que GenerateMac usou para calcular a etiqueta original de HMAC. A operação VerifyMac calcula a etiqueta de HMAC e verifica se ela é idêntica à etiqueta de HMAC fornecida. Se as etiquetas de HMAC recebidas e calculadas não forem idênticas, a verificação falhará.

As chaves do KMS de HMAC não são compatíveis com [alternância automática de chaves](#) e não é possível criar uma chave do KMS de HMAC em um [armazenamento personalizado de chaves](#).

Se você estiver criando uma chave KMS para criptografar dados em um AWS serviço, use uma chave de criptografia simétrica. Não é possível usar uma chave do KMS de HMAC.

Regiões

As chaves HMAC KMS são suportadas em todos os Regiões da AWS suportadas. AWS KMS

Saiba mais

- Para criar chaves do KMS de HMAC, consulte [Criar uma chave do KMS HMAC](#).
- Para criar chaves do KMS de HMAC de várias regiões, consulte [Chaves multirregionais em AWS KMS](#).
- Para examinar a diferença na política de chaves padrão que o AWS KMS console define para chaves HMAC KMS, consulte [the section called “Permite que usuários de chaves usem uma chave do KMS para operações de criptografia”](#)
- Para identificar e visualizar chaves do KMS de HMAC, consulte [Identificar chaves do KMS de HMAC](#).
- Para saber como usar HMACs para criar tokens web JSON, consulte [Como se proteger HMACs internamente AWS KMS](#) no Blog AWS de segurança.
- Ouça um podcast: [Apresentando HMACs o AWS Key Management Service](#) The Official AWS Podcast.

Chaves ML-DSA em AWS KMS

AWS Key Management Service (AWS KMS) suporta o algoritmo de assinatura digital Module-Lattice (ML-DSA) para assinaturas criptográficas pós-quânticas. Essa implementação segue o [padrão 204 do Federal Information Processing Standards \(FIPS\)](#) para ajudar a proteger contra futuras ameaças

de computação quântica. O AWS KMS cria e protege todas as chaves e operações de assinatura ML-DSA nos módulos de segurança de hardware validados pelo FIPS 140-3 Security Level 3. Para ajudar a equilibrar segurança com desempenho, o ML-DSA in AWS KMS oferece três níveis de segurança distintos por meio de diferentes especificações-chave, ML_DSA_44, ML_DSA_65 e ML_DSA_87.

AWS KMS suporta assinaturas de chave assimétrica para mensagens de até 4 KB usando o RAW tipo de mensagem. Para mensagens maiores, você deve calcular externamente o μ de representação de mensagem de 64 bytes usado na assinatura ML-DSA, conforme definido no NIST FIPS 204 seção 6.2. Use o tipo de EXTERNAL_MU mensagem na operação AWS KMS [Assinar](#) para especificar essa mensagem pré-processada de 64 bytes. As assinaturas produzidas pelo μ computado externamente são iguais às RAW ao usar a mesma mensagem e chave privada. Observe que essa assinatura é diferente da ML-DSA “pré-hash” ou HashML-DSA da seção 5.4 do NIST FIPS 204 .

Para obter mais informações sobre como usar ML-DSA e o tipo de mensagem EXTERNAL_MU, consulte [Especificações de chave ML-DSA](#).

Para obter um exemplo de uso de ML-DSA e do tipo de mensagem EXTERNAL_MU, consulte [Verificação off-line com pares de chaves ML-DSA](#).

Chaves multirregionais em AWS KMS

AWS KMS suporta chaves multirregionais, que são AWS KMS keys diferentes Regiões da AWS e podem ser usadas de forma intercambiável, como se você tivesse a mesma chave em várias regiões. Cada conjunto de chaves multirregionais relacionadas tem o mesmo material de chave e [ID de chave](#), portanto, você pode criptografar dados em uma Região da AWS e descriptografá-las em outra Região da AWS sem precisar recriptografar ou fazer uma chamada entre regiões. AWS KMS

Como todas as chaves KMS, as chaves multirregionais nunca saem AWS KMS sem criptografia. Você pode criar chaves de várias regiões simétricas ou assimétricas para criptografia ou assinatura, criar chaves de várias regiões de HMAC para gerar e verificar etiquetas de HMAC, e criar [chaves de várias regiões com material de chave importado](#) ou o material de chave que o AWS KMS gera. Você deve gerenciar cada chave de várias regiões de maneira independente, incluindo a criação de aliases e tags, a definição de suas principais políticas e concessões e sua habilitação e sua desabilitação seletivas. Você pode usar chaves de várias regiões em todas as operações de criptografia que podem ser feitas com chaves de região única.

As chaves de várias regiões são uma solução flexível e eficiente para vários cenários comuns de segurança de dados.

Recuperação de desastres

Em uma arquitetura de backup e recuperação, as chaves multirregionais permitem que você processe dados criptografados sem interrupção, mesmo no caso de uma Região da AWS interrupção. Os dados mantidos na região de backup podem ser descriptografados nessa região, enquanto os dados recém-criptografados na região de backup podem ser descriptografados na região principal quando esta for restaurada.

Gerenciamento de dados globais

As empresas que operam globalmente precisam de dados distribuídos globalmente que estejam disponíveis de forma consistente entre Regiões da AWS. Você pode criar chaves de várias regiões em todas as regiões nas quais seus dados residem e, em seguida, usar essas chaves como se fossem uma chave de região única, sem a latência de uma chamada entre regiões ou o custo de recriptografar os dados com uma chave diferente em cada região.

Aplicações de assinatura distribuídas

As aplicações que exigem recursos de assinatura entre regiões podem usar chaves de assinatura assimétrica de várias regiões para gerar assinaturas digitais idênticas de maneira consistente e repetida em diferentes Regiões da AWS.

Se você usa o encadeamento de certificados com um único repositório confiável global (para uma única autoridade de certificação (CA) raiz e um intermediário regional CAs assinado pela CA raiz, você não precisa de chaves multirregionais. No entanto, se seu sistema não oferecer suporte a intermediários CAs, como assinatura de aplicativos, você poderá usar chaves multirregionais para dar consistência às certificações regionais.

Aplicações ativas-ativas que abrangem várias regiões

Algumas workloads e aplicações podem abranger várias regiões em arquiteturas ativas-ativas. Para essas aplicações, chaves de várias regiões podem reduzir a complexidade, fornecendo o mesmo material de chave para operações simultâneas de criptografia e descriptografia em dados que podem estar se movendo pelos limites da região.

Você pode usar chaves de várias regiões com bibliotecas de criptografia do lado do cliente, como [AWS Encryption SDK](#), [AWS Database Encryption SDK](#) e a [criptografia do lado do cliente do Amazon S3](#).

[AWS os serviços que se integram AWS KMS](#) para criptografia em repouso ou assinaturas digitais atualmente tratam as chaves multirregionais como se fossem chaves de uma única região. Elas

podem reempacotar ou criptografar novamente os dados movidos entre Regiões. Por exemplo, a replicação entre regiões do Amazon S3 descriptografa e recriptografa os dados em uma chave do KMS na Região de destino, mesmo ao replicar objetos protegidos por uma chave de várias Regiões.

Chaves de várias Regiões não são globais. Você cria uma chave primária de várias Regiões e, em seguida, a replica em Regiões selecionadas em uma [partição da AWS](#). Em seguida, você gerencia a chave de várias Regiões em cada Região independentemente. AWS Nem AWS KMS nunca cria ou replica automaticamente chaves multirregionais em qualquer região em seu nome. [Chaves gerenciadas pela AWS](#), as chaves KMS que AWS os serviços criam em sua conta para você, são sempre chaves de região única.

Nas regiões China, você pode usar o atributo de chave multirregional para replicar chaves do KMS dentro da partição Regiões China (aws-cn). Por exemplo, você pode replicar uma chave da região China (Pequim) para China (Ningxia) ou vice-versa. Ao replicar uma chave de uma região da China para outra, você concorda em usar a AWS Key Management Service da região de destino e cumprir todos os termos de contrato aplicáveis à região de destino. Você não pode replicar uma chave das regiões de Pequim e Ningxia em uma AWS região fora da partição das regiões da China. Da mesma forma, você não pode replicar uma chave de uma região fora da partição Regiões China para as regiões Pequim e Ningxia.

Não é possível converter uma chave de Região única existente em uma chave de várias Regiões. Esse design garante que todos os dados protegidos com chaves de Região única existentes mantenham as mesmas propriedades de residência e soberania dos dados.

Para a maioria das necessidades de segurança de dados, o isolamento regional e a tolerância a falhas dos recursos regionais tornam as chaves padrão AWS KMS de região única a solução mais adequada. No entanto, quando você precisa criptografar ou assinar dados em aplicações no lado do cliente em várias Regiões, as chaves de várias Regiões podem ser a melhor solução.

Regiões

As chaves multirregionais são suportadas em todos Regiões da AWS os AWS KMS suportes.

Preços e cotas

Cada chave em um conjunto de chaves de várias Regiões relacionado conta como uma única chave do KMS para preços e cotas. As [cotas do AWS KMS](#) são calculadas separadamente para cada Região de uma conta. O uso e o gerenciamento das chaves de várias Regiões em cada Região contam para as cotas para essa Região.

Tipos de chave do KMS com suporte

É possível criar os seguintes tipos de chaves KMS multirregionais:

- Chaves do KMS de criptografia simétrica
- Chaves do KMS assimétricas
- Chaves do KMS de HMAC
- Chaves do KMS com material de chave importado

Não é possível criar chaves de várias Regiões em um armazenamento de chaves personalizado.

Saiba mais

- Para saber como controlar o acesso a chaves do KMS de várias regiões, consulte [Controlar o acesso a chaves de várias regiões](#).
- Para criar chaves do KMS primárias de várias regiões, consulte [Criar chaves primárias de várias regiões](#).
- Para criar chaves do KMS de réplica de várias regiões, consulte [Criar chaves de réplica de várias regiões](#).
- Para atualizar a região primária, consulte [Alterar a chave primária em um conjunto de chaves de várias regiões](#).
- Para identificar e visualizar chaves do KMS de várias regiões, consulte [Identificar chaves do KMS de HMAC](#).
- Para saber mais sobre considerações especiais sobre a exclusão de chaves do KMS de várias regiões, consulte [Deleting multi-Region keys](#).

Terminologia e conceitos

Os seguintes termos e conceitos são usados com chaves de várias Regiões.

Chave de várias Regiões

A chave de várias Regiões é uma chave de um conjunto de chaves do KMS com o mesmo ID de chave e material de chave (e outras [propriedades compartilhadas](#)) em diferentes Regiões da AWS. Cada chave de várias Regiões é uma chave do KMS totalmente funcional que pode ser usada de maneira independente de suas chaves de várias Regiões relacionadas. Como todas as

chaves multirregionais relacionadas têm o mesmo ID e material de chave, elas são interoperáveis, ou seja, qualquer chave multirregional relacionada em qualquer uma delas Região da AWS pode descriptografar texto cifrado criptografado por qualquer outra chave multirregional relacionada.

Você define a propriedade de várias Regiões de uma chave do KMS ao criá-la. Não é possível alterar a propriedade de várias regiões em uma chave existente. Não é possível converter uma chave de região única em chave de várias regiões nem converter uma chave de várias regiões em uma chave de região única. Para mover workloads existentes para cenários de várias regiões, é necessário recriptografar seus dados ou criar novas assinaturas com novas chaves de várias regiões.

[Uma chave multirregional pode ser simétrica ou assimétrica e pode usar material chave ou material AWS KMS chave importado.](#) Não é possível criar chaves de várias Regiões em um [armazenamento de chaves personalizado](#).

Em um conjunto de chaves de várias Regiões relacionadas, há exatamente uma [chave primária](#) em um determinado momento. Você pode criar [chaves de réplica](#) dessa chave primária em outras Regiões da AWS. Também pode [atualizar a região primária](#), o que transforma a chave primária em uma chave de réplica e transforma uma chave de réplica especificada na chave primária. No entanto, você pode manter somente uma chave primária ou chave de réplica em cada uma Região da AWS. Todas as Regiões devem estar na mesma [partição da AWS](#).

Você pode ter diversos conjuntos de chaves de várias regiões relacionadas nas mesmas Regiões da AWS ou em regiões diferentes. Embora chaves de várias Regiões relacionadas sejam interoperáveis, chaves de várias Regiões não relacionadas não são interoperáveis.

Chave primária

Uma chave primária multirregional é uma chave KMS que pode ser replicada em outras Regiões da AWS na mesma partição. Cada conjunto de chaves de várias Regiões tem somente uma chave primária.

Uma chave primária é diferente de uma chave de réplica nos seguintes aspectos:

- Somente uma chave primária pode ser [replicada](#).
- A chave primária é a fonte de [propriedades compartilhadas](#) de suas [chaves de réplica](#), incluindo o material da chave e o ID da chave.
- Você pode habilitar e desabilitar a [alternância automática de chaves](#) somente em uma chave primária.

- Você pode [programar a exclusão de uma chave primária](#) a qualquer momento. Mas não AWS KMS excluirá uma chave primária até que todas as suas chaves de réplica sejam excluídas.

Entretanto, as chaves primárias e de réplica não diferem em nenhuma propriedade criptográfica. É possível usar uma chave primária e suas chaves de réplica alternadamente.

Não é necessário replicar uma chave primária. Você pode usá-la como faria com qualquer chave do KMS e replicá-la se e quando ela for útil. No entanto, como chaves de várias Regiões têm propriedades de segurança diferentes das chaves de uma única Região, recomendamos criar uma chave de várias Regiões apenas quando você planeja replicá-la.

Chave de réplica

Uma chave de réplica de várias regiões é uma chave do KMS que tem o mesmo [ID de chave](#) e material de chave da sua [chave primária](#) e chaves de réplica relacionadas, mas existe em outra Região da AWS.

Uma chave de réplica é uma chave do KMS totalmente funcional com sua própria política de chave, concessões, alias, etiquetas e outras propriedades. Ela não é uma cópia ou ponteiro da chave primária ou de qualquer outra chave. Você pode usar uma chave de réplica mesmo quando sua chave primária e todas as chaves de réplica relacionadas estão desabilitadas. Também pode converter uma chave de réplica em uma chave primária, e vice-versa. Depois de criada, uma chave de réplica depende de sua chave primária somente para [alternância de chaves](#) e [atualização da Região primária](#).

Chaves primárias e de réplica não diferem em nenhuma propriedade criptográfica. É possível usar uma chave primária e suas chaves de réplica alternadamente. Os dados criptografados por uma chave primária ou uma chave de réplica podem ser descriptografados pela mesma chave ou por qualquer chave primária ou chave de réplica relacionada.

Replicar

Você pode replicar uma [chave primária](#) multirregional em outra Região da AWS na mesma partição. Ao fazer isso, AWS KMS cria uma [chave de réplica](#) multirregional na região especificada com o mesmo [ID de chave](#) e outras [propriedades compartilhadas](#) da chave primária. Para chaves KMS com AWS_KMS origem, transporta com AWS KMS segurança o material da chave através do limite da região e o associa à nova chave de réplica, tudo dentro. AWS KMS Para chaves KMS com EXTERNAL origem, você deve importar o mesmo material de chave que você importou para a chave de região primária para cada réplica de chave de região individualmente.

Propriedades compartilhadas

As propriedades compartilhadas são propriedades de uma chave primária multirregional que são compartilhadas com suas chaves de réplica. AWS KMS cria as chaves de réplica com os mesmos valores de propriedade compartilhada da chave primária. Em seguida, ele sincroniza periodicamente os valores das propriedades compartilhadas da chave primária com suas chaves de réplica. Não é possível definir essas propriedades em uma chave de réplica.

Veja a seguir as propriedades compartilhadas de chaves de várias Regiões.

- [ID da chave](#): (O elemento Region do elemento do é diferente do [ARN da chave](#).)
- [Material chave](#) — As chaves primária e de réplica em um conjunto de chaves multirregionais relacionadas compartilham o mesmo material chave. Para chaves multirregionais cujo material principal é gerado por AWS KMS (AWS_KMSorigem), transporta AWS KMS com segurança todos os materiais principais do primário para cada réplica quando a réplica é criada ou quando um novo material de chave é criado por meio de rotação automática ou sob demanda. Para chaves multirregionais com material de chave importado (EXTERNALorigem), AWS KMS sincroniza o identificador do material da chave primária, mas você deve importar o material da chave para cada chave de réplica de forma independente.
- [Origem do material de chave](#)
- [Especificação da chave](#) e algoritmos de criptografia
- [Uso da chave](#)
- [Alternância da chave automática](#):: você pode habilitar e desabilitar a alternância automática de chaves somente em uma chave primária. Novas chaves de réplica são criadas com todas as versões do material de chave compartilhado. Para obter detalhes, consulte [Rotating multi-Region keys](#).
- [Alternância sob demanda](#): você só pode realizar a alternância sob demanda na chave primária. Para chaves multirregionais cujo material de chave é gerado por AWS KMS (AWS_KMSorigem), AWS KMS cria chaves de réplica com todas as versões do material de chave compartilhado. Para chaves multirregionais com material de chave importado (EXTERNALorigem), AWS KMS propaga o ID do material-chave e a descrição do material-chave da chave primária para as chaves de réplica, mas não o material da chave. Você deve importar o material de chave correto para cada chave de réplica individualmente. Para obter detalhes, consulte [Rotating multi-Region keys](#).

Você também pode pensar nemas designações primárias e de réplicas de chaves de várias Regiões relacionadas como propriedades compartilhadas. Quando você [cria novas chaves de réplica](#) ou

[atualiza a chave primária](#), AWS KMS sincroniza a alteração com todas as chaves multirregionais relacionadas. Quando essas alterações estiverem concluídas, todas as chaves de várias Regiões relacionadas listarão suas chaves primárias e chaves de réplica com precisão.

[Todas as outras propriedades das chaves multirregionais são propriedades independentes, incluindo a descrição da chave, a política da chave, as concessões, os estados da chave ativada e desativada, os aliases e as tags](#). Você pode definir os mesmos valores para essas propriedades em todas as chaves de várias regiões relacionadas. Porém, se você alterar o valor de uma propriedade independente, o AWS KMS não a sincronizará.

Você pode rastrear a sincronização das propriedades compartilhadas das suas chaves de várias regiões. Em seu AWS CloudTrail registro, procure o [SynchronizeMultiRegionKey](#) evento.

Considerações de segurança de chaves de várias Regiões

Use uma chave AWS KMS multirregional somente quando precisar de uma. Chaves de várias Regiões fornecem uma solução flexível e escalável para workloads que movem dados criptografados entre Regiões da AWS ou que precisam de acesso entre Regiões. Considere usar uma chave de várias Regiões se precisar compartilhar, mover ou fazer backup de dados protegidos entre Regiões ou se precisar criar assinaturas digitais idênticas de aplicações que operam em Regiões diferentes.

No entanto, o processo de criar uma chave de várias Regiões move seu material de chave entre limites de Região da AWS dentro do AWS KMS. O texto criptografado gerado por uma chave de várias Regiões pode ser descriptografado por várias chaves relacionadas em diversas localizações geográficas. Existem também benefícios significativos para serviços e recursos regionalmente isolados. Cada Região da AWS é independente e isolada das outras Regiões. As regiões fornecem tolerância a falhas, estabilidade e resiliência e também podem reduzir a latência. Elas permitem criar recursos redundantes que permanecem disponíveis e não são afetados por uma interrupção em outra Região. Em AWS KMS, eles também garantem que cada texto cifrado possa ser decifrado por apenas uma chave.

As chaves de várias Regiões também levantam novas considerações de segurança:

- É mais complexo controlar o acesso e impor a política de segurança de dados com chaves de várias Regiões. Você precisa garantir que a política seja auditada de maneira consistente na chave entre várias regiões isoladas. Também precisa usar políticas para impor limites, em vez de depender de chaves separadas.

Por exemplo, você precisa definir condições de políticas nos dados para evitar que equipes de folha de pagamento em uma região possam ler dados de folha de pagamento de uma Região

diferente. Além disso, você deve usar o controle de acesso para evitar um cenário em que uma chave de várias Regiões em uma Região protege os dados de um locatário e uma chave de várias Regiões relacionada em outra Região protege os dados de um locatário diferente.

- A auditoria de chaves entre Regiões também é mais complexa. Com chaves de várias Regiões, você precisa examinar e reconciliar atividades de auditoria em várias Regiões para obter uma compreensão completa das principais atividades em dados protegidos.
- A conformidade com exigências de residência de dados pode ser mais complexa. Com Regiões isoladas, você garante a conformidade da residência de dados e a soberania dos dados. As chaves do KMS em uma determinada Região podem descriptografar dados sigilosos somente nessa Região. Os dados criptografados em uma Região permanecem protegidos e inacessíveis em qualquer outra Região.

Para verificar a residência e a soberania dos dados com chaves multirregionais, você precisa implementar políticas de acesso e compilar AWS CloudTrail eventos em várias regiões.

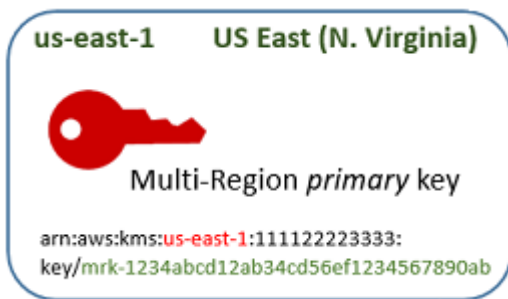
[Para facilitar o gerenciamento do controle de acesso em chaves multirregionais, a permissão para replicar uma chave multirregional \(kms: ReplicateKey\) é separada da permissão padrão para criar chaves \(kms:\). CreateKey](#) Além disso, AWS KMS oferece suporte a várias condições de política para chaves multirregionais `kms:MultiRegion`, inclusive, que permite ou nega permissão para criar, usar ou gerenciar chaves multirregionais e `kms:ReplicaRegion` que restringe as regiões nas quais uma chave multirregional pode ser replicada. Para obter detalhes, consulte [Controlar o acesso a chaves de várias regiões](#).

Como funcionam chaves de várias Regiões

Você começa criando uma [chave primária multirregional](#) simétrica ou assimétrica em uma Região da AWS que AWS KMS ofereça suporte, como Leste dos EUA (Norte da Virgínia). Você decide se uma chave é de Região única ou de várias Regiões apenas ao criá-la. Não será possível alterar essa propriedade mais tarde. Como em qualquer chave do KMS, você define uma política de chave para a chave de várias Região, além de poder criar concessões e adicionar aliases e tags para categorização e autorização. (Estas são [propriedades independentes](#) que não são compartilhadas ou sincronizadas com outras chaves.) Você pode usar sua chave primária de várias Regiões em operações de criptografia para criptografia ou assinatura.

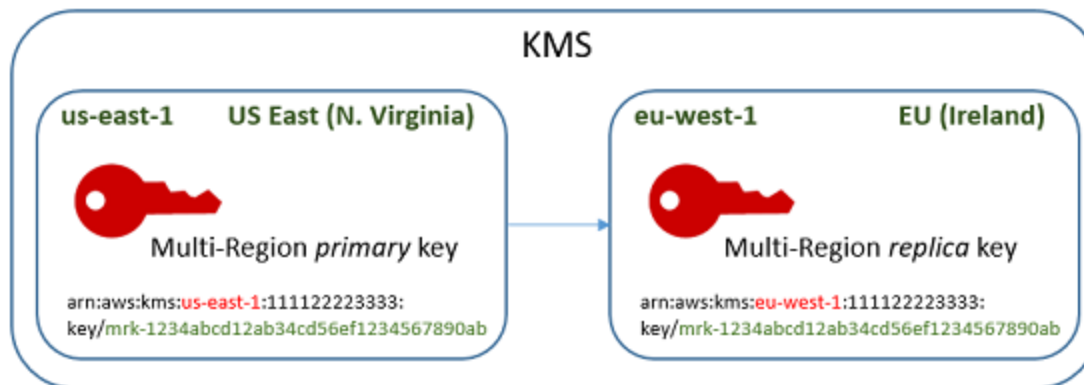
Você pode [criar uma chave primária multirregional](#) no AWS KMS console ou usando a [CreateKey](#) API com o `MultiRegion` parâmetro definido como `true`. Observe que chaves de várias Regiões têm

um ID de chave distintivo que começa com `mrk-`. Você pode usar o `mrk-` prefixo para identificar MRKs programaticamente.



Se você escolher, poderá [replicar](#) a chave primária multirregional em uma ou mais diferentes Regiões da AWS na mesma [AWS partição](#), como Europa (Irlanda). Ao fazer isso, AWS KMS cria uma [chave de réplica](#) na região especificada com o mesmo ID de chave e outras [propriedades compartilhadas](#) da chave primária. O resultado são duas chaves de várias Regiões relacionados, uma chave primária e uma chave de réplica, que podem ser usadas de maneira intercambiável.

Você pode [criar uma chave de réplica multirregional](#) no AWS KMS console ou usando a [ReplicateKey](#) API.



A [chave de réplica de várias Regiões](#) resultante é uma chave do KMS totalmente funcional com as mesmas [propriedades compartilhadas](#) que a chave primária. Em todos os outros aspectos, ela é uma chave do KMS independente com sua própria descrição, política de chave, concessões, aliases e etiquetas. Habilitar ou desabilitar uma chave de várias Regiões não tem efeito sobre chaves de várias Regiões relacionadas. Você pode usar as chaves primárias e de réplica independentemente em operações de criptografia ou coordenar seu uso. Por exemplo, você pode criptografar dados com a chave primária na região Leste dos EUA (Norte da Virgínia), mover os dados até a região Europa (Irlanda) e usar a chave de réplica para descriptografar esses dados.

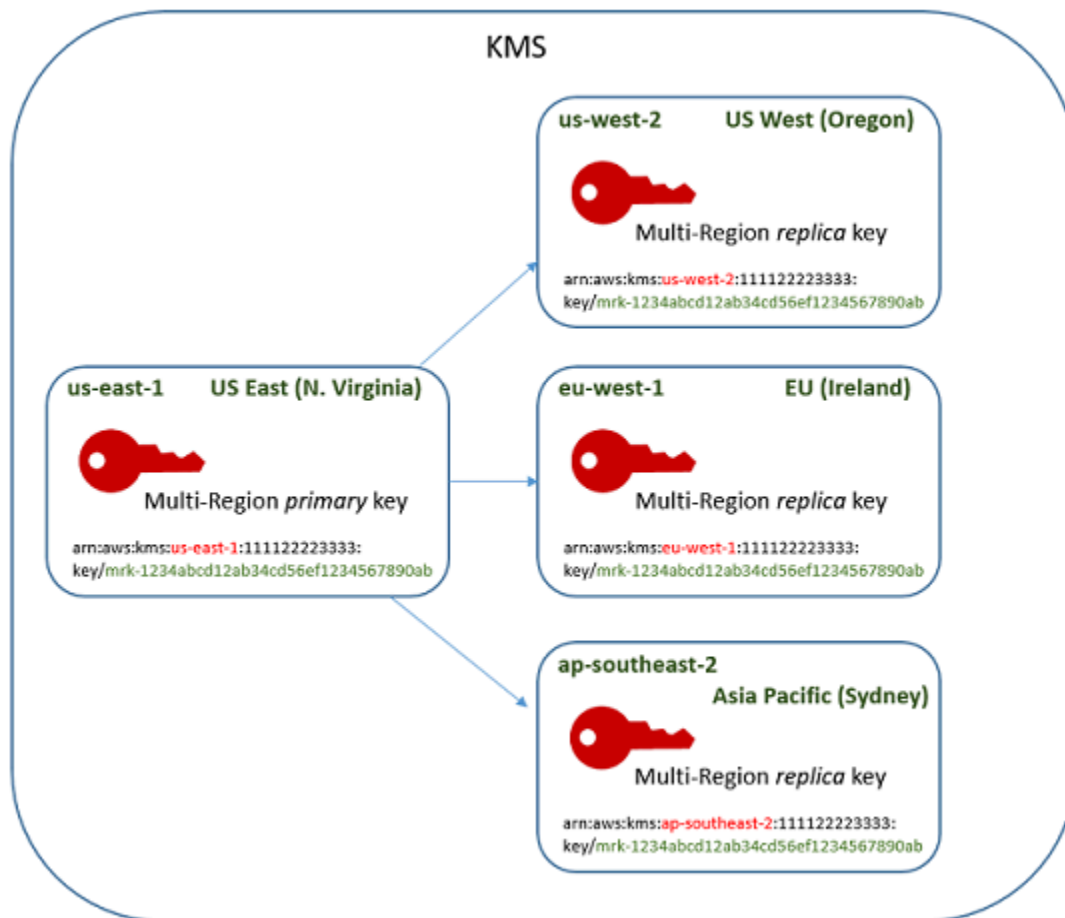
Chaves de várias Regiões relacionadas têm o mesmo ID de chave. Sua chave ARNs (Amazon Resource Names) difere somente no campo Região. Por exemplo, a chave primária multirregional

e as chaves de réplica podem ter o seguinte exemplo de chave. ARNs O ID da chave, o último elemento no ARN da chave, é idêntico. Ambas as chaves têm o ID de chave distinto de chaves de várias regiões, que começa com `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
```

Ter o mesmo ID de chave é um requisito para interoperabilidade. Ao criptografar, AWS KMS vincula a ID da chave KMS ao texto cifrado para que o texto cifrado possa ser descriptografado somente com essa chave KMS ou com uma chave KMS com a mesma ID de chave. Esse recurso também torna as chaves de várias Regiões relacionadas fáceis de reconhecer e facilita sua utilização intercambiável. Por exemplo, ao usá-las em uma aplicação, você pode fazer referência a chaves de várias Regiões relacionadas com base no ID de chave compartilhada. Em seguida, se necessário, especifique a Região ou o ARN para diferenciá-las.

Conforme suas necessidades de dados mudam, você pode replicar a chave primária para outra Regiões da AWS na mesma partição, como Oeste dos EUA (Oregon) e Ásia-Pacífico (Sydney). O resultado são quatro chaves multirregionais relacionadas com o mesmo material e chave IDs, conforme mostrado no diagrama a seguir. Você gerencia as chaves de maneira independente. Para chaves multirregionais com material de chave importado, você é responsável por importar o material de chaves para cada chave relacionada individualmente. Elas podem ser usadas de maneira independente ou coordenada. Por exemplo, é possível criptografar dados com a chave de réplica na região Ásia-Pacífico (Sydney), mover os dados para a região Oeste dos EUA (Oregon) e descriptografá-los com a chave de réplica na região Oeste dos EUA (Oregon).



Outras considerações para chaves de várias Regiões incluem as seguintes.

Sincronização de propriedades compartilhadas — [Se uma propriedade compartilhada das chaves multirregionais for alterada, AWS KMS sincronizará automaticamente a alteração da chave primária para todas as suas chaves de réplica.](#) Você não pode solicitar ou forçar uma sincronização de propriedades compartilhadas. AWS KMS detecta e sincroniza todas as alterações para você. No entanto, você pode auditar a sincronização usando o [SynchronizeMultiRegionKey](#) evento nos CloudTrail registros.

Por exemplo, se você habilitar a rotação automática de chaves em uma chave primária multirregional simétrica com AWS_KMS origem, AWS KMS copiará essa configuração para todas as chaves de réplica. Quando o material de chave é alternado, a alternância é sincronizada entre todas as chaves de várias Regiões relacionadas, para que elas continuem a ter o mesmo material de chave atual e também acesso a todas as versões mais antigas do material de chave. Se você criar uma nova chave de réplica, ela terá o mesmo material de chave atual de todas as chaves de várias Regiões relacionadas e terá acesso a todas as versões anteriores do material de chave. Para obter detalhes, consulte [Rotating multi-Region keys](#).

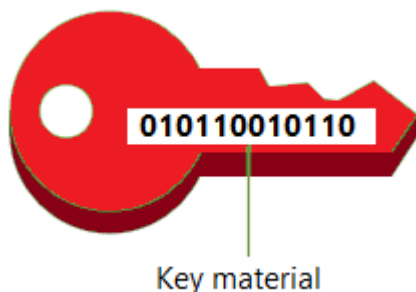
Alterar a chave primária: todo conjunto de chaves de várias Regiões deve ter exatamente uma chave primária. A [chave primária](#) é a única que pode ser replicada. Ela é também a fonte das propriedades compartilhadas das suas chaves de réplica. Porém, você pode transformar a chave primária em uma réplica e promover uma das chaves de réplica para chave primária. Você pode fazer isso para excluir uma chave primária de várias Regiões de uma determinada Região ou localizar a chave primária em uma Região mais próxima dos administradores do projeto. Para obter detalhes, consulte [Alterar a chave primária em um conjunto de chaves de várias regiões](#).

Excluindo chaves multirregionais — Como todas as chaves KMS, você deve programar a exclusão das chaves multirregionais antes de excluí-las. AWS KMS Enquanto a exclusão da chave estiver pendente, não será possível usá-la em operações de criptografia. No entanto, não AWS KMS excluirá uma chave primária multirregional até que todas as chaves de réplica sejam excluídas. Para obter mais detalhes, consulte [Deleting multi-Region keys](#).

Importação de material chave para AWS KMS chaves

Você pode criar uma AWS KMS keys (chave KMS) com o material de chave fornecido por você.

Uma chave do KMS é uma representação lógica de uma chave de dados. Os metadados de uma chave do KMS incluem o ID do material de chave usado para executar operações criptográficas. Quando você [cria uma chave KMS](#), por padrão, AWS KMS gera o material da chave para essa chave KMS. No entanto, você pode criar uma chave do KMS sem material de chave e importar o seu próprio material de chaves para essa chave do KMS, um recurso normalmente conhecido como "traga sua própria chave" (BYOK).



Note

AWS KMS não suporta a descryptografia de nenhum AWS KMS texto cifrado criptografado por uma chave KMS de criptografia simétrica externa AWS KMS, mesmo que o texto cifrado tenha sido criptografado em uma chave KMS com material de chave importado. AWS KMS

não publica o formato de texto cifrado exigido por essa tarefa e o formato pode ser alterado sem aviso prévio.

Ao usar material de chave importado, você permanece responsável pelo material de chaves, AWS KMS permitindo o uso de uma cópia dele. Você pode optar por fazer isso por um ou mais dos seguintes motivos:

- Para comprovar que o material de chaves foi gerado usando uma origem de entropia que atende aos seus requisitos.
- Para usar o material chave de sua própria infraestrutura com AWS serviços e AWS KMS para gerenciar o ciclo de vida desse material chave em seu interior. AWS
- Para usar chaves existentes e bem estabelecidas AWS KMS, como chaves para assinatura de código, assinatura de certificado PKI e aplicativos fixados em certificados
- Para definir um prazo de validade para o material de chave AWS e [excluí-lo manualmente](#), mas também para disponibilizá-lo novamente no futuro. Por outro lado, [programar a exclusão de chaves](#) exige um período de espera de 7 a 30 dias, após os quais você não pode recuperar a chave do KMS excluída.
- Possuir a cópia original do material principal e mantê-la do lado de fora AWS para maior durabilidade e recuperação de desastres durante todo o ciclo de vida do material principal.
- Para chaves assimétricas e chaves HMAC, a importação cria chaves compatíveis e interoperáveis que operam dentro e fora dela. AWS

Tipos de chave do KMS com suporte

AWS KMS suporta material de chave importado para os seguintes tipos de chaves KMS. Não é possível importar material de chave para chaves do KMS em [repositórios de chaves personalizados](#).

- [Chaves do KMS de criptografia simétrica](#)
- [Chaves do KMS assimétricas \(exceto chaves ML-DSA\)](#)
- [Chaves do KMS de HMAC](#)
- [Chaves multirregionais](#) de todos os tipos compatíveis.

Regiões

O material chave importado é suportado em todas as Regiões da AWS e os AWS KMS suportados.

Nas regiões da China, os requisitos de material de chave para chaves do KMS de criptografia simétrica são diferentes dos requisitos de outras regiões. Para obter detalhes, consulte [Etapa 3: Criptografar o material de chave](#).

Saiba mais

- Para criar chaves do KMS com material de chave importado, consulte [Criar uma chave do KMS com material de chave importado](#).
- Para criar um alarme que envie uma notificação quando o material de chave importado em uma chave do KMS estiver se aproximando da data de expiração, consulte [Crie um CloudWatch alarme para expiração do material chave importado](#).
- Para reimportar o material de chave para uma chave do KMS, consulte [Reimportar material de chave](#).
- Para importar novo material de chave para uma chave do KMS para rodízio sob demanda, consulte [Importar novo material de chave](#) e [Executar alternância de chaves sob demanda](#).
- Para identificar e visualizar chaves do KMS com material de chave importado, consulte [Identificar chaves do KMS com material de chave importado](#).
- Para conhecer considerações especiais sobre a exclusão de chaves do KMS com material de chave importado, consulte [Deleting KMS keys with imported key material](#).

Considerações especiais para material de chave importado

Antes de decidir importar material de chave para AWS KMS, você deve compreender as seguintes características do material de chave importado.

Você gera o material de chave

Você é responsável por gerar o material de chaves usando uma origem de aleatoriedade que atende aos seus requisitos de segurança.

Você é responsável pela disponibilidade e durabilidade

AWS KMS foi projetado para manter o material chave importado altamente disponível. Mas AWS KMS não mantém a durabilidade do material chave importado no mesmo nível do material chave AWS KMS gerado. Para obter detalhes, consulte [Proteger o material de chave importada](#).

Você pode excluir o material de chave.

Você pode [excluir material de chave importado](#) de uma chave do KMS, tornando imediatamente a chave do KMS inutilizável. Além disso, ao importar o material de chave para uma chave do

KMS, é possível determinar se a chave expira e [definir seu prazo de validade](#). Quando o prazo de validade chegar, AWS KMS [exclui o material da chave](#). Sem o material de chave, a chave do KMS não pode ser usada em nenhuma operação criptográfica. Para restaurar a chave, é necessário reimportar o mesmo material de chave para a chave.

Você não pode alterar o material da chave para chaves assimétricas e HMAC

Quando você importa o material de chave para uma chave do KMS, esta é associada permanentemente a esse material de chave. Você pode [reimportar o mesmo material de chaves](#), mas não pode importar outro material de chaves para essa chave do KMS. Além disso, não é possível [habilitar a alternância automática de chaves](#) de uma chave do KMS com material de chaves importado. No entanto, você pode [alternar manualmente uma chave do KMS](#) com material de chave importado.

Você pode executar a rotação sob demanda em chaves de criptografia simétricas

Chaves de criptografia simétricas com material de chave importado oferecem suporte à rotação sob demanda. Você pode [importar vários materiais de chave](#) para essas chaves e usar [rodízio sob demanda](#) para atualizar o material de chave atual. O material de chave atual é usado tanto para criptografia quanto para descryptografia, mas outros materiais de chave (não atuais) podem ser usados somente para decodificação.

Não é possível alterar a origem do material de chave

As chaves do KMS projetadas para material de chave importado têm um valor de [origem](#) de EXTERNAL que não pode ser alterado. Você não pode converter uma chave KMS de material de chave importado para usar material de chave de qualquer outra fonte, inclusive AWS KMS. Da mesma forma, você não pode converter uma chave KMS com material de AWS KMS em uma projetada para material de chave importado.

Não é possível exportar material de chave

Você não pode exportar nenhum material de chave que tenha importado. AWS KMS não é possível devolver o material da chave importada para você de nenhuma forma. Você deve manter uma cópia do material de chaves importado fora dele AWS, de preferência em um gerenciador de chaves, como um módulo de segurança de hardware (HSM), para poder reimportar o material de chaves se você excluí-lo ou se ele expirar.

É possível criar chaves de várias regiões com material de chave importado.

A chave de várias regiões com material de chave importado tem recursos das chaves do KMS com material de chave importado e pode interoperar entre as Regiões da AWS. Para criar uma chave de várias regiões com material de chave importado, você deve importar o mesmo material

de chave para a chave do KMS primária e para cada chave de réplica. Para obter mais detalhes sobre a importação de materiais essenciais para chaves multirregionais, consulte.

As chaves assimétricas e as chaves de HMAC são portáteis e interoperáveis

Você pode usar seu material de chave assimétrica e material de chave HMAC externamente AWS para interoperar com AWS KMS chaves com o mesmo material de chave importado.

Ao contrário do texto cifrado AWS KMS simétrico, que está inextricavelmente vinculado à chave KMS usada no algoritmo, AWS KMS usa formatos HMAC e assimétricos padrão para criptografia, assinatura e geração de MAC. Como resultado, as chaves são portáteis e compatíveis com cenários tradicionais de garantia de chaves.

Quando sua chave KMS tiver importado material de chave, você poderá usar o material de chave importado de fora AWS para realizar as seguintes operações.

- Chaves de HMAC — É possível verificar uma etiqueta de HMAC que foi gerada pela chave do KMS de HMAC com material de chave importado. Você também pode usar a chave HMAC KMS com o material de chave importado para verificar uma tag HMAC que foi gerada pelo material de chave externo. AWS
- Chaves de criptografia assimétricas — Você pode usar sua chave privada de criptografia assimétrica externa para AWS descriptografar um texto cifrado criptografado pela chave KMS com a chave pública correspondente. Você também pode usar sua chave KMS assimétrica para descriptografar um texto cifrado assimétrico que foi gerado fora do. AWS
- Chaves de assinatura assimétrica — Você pode usar sua chave KMS de assinatura assimétrica com material de chave importado para verificar as assinaturas digitais geradas por sua chave de assinatura privada fora da. AWS Você também pode usar sua chave de assinatura pública assimétrica no exterior AWS para verificar as assinaturas geradas pela sua chave KMS assimétrica.
- Chaves de acordo de chave assimétrica: é possível usar a chave do KMS de acordo de chave assimétrica com material de chave importado para obter segredos compartilhados com um par fora da AWS.

Se você importar o mesmo material de chave para chaves do KMS diferentes da mesma Região da AWS, essas chaves também serão interoperáveis. Para criar chaves KMS interoperáveis em diferentes Regiões da AWS, crie uma chave multirregional com material de chave importado.

Chaves privadas RSA

- AWS KMS exige que as chaves privadas RSA importadas tenham fatores primos em conformidade com o teste descrito no [FIPS 186-5](#), Seção A. 1.3. Outros software ou

dispositivos podem usar algoritmos diferentes para validar esses fatores primos de chaves privadas RSA. Em casos raros, chaves validadas usando outros algoritmos podem não ser aceitas pelo AWS KMS.

As chaves de criptografia simétricas não são portáteis nem interoperáveis

Os textos cifrados simétricos que AWS KMS produz não são portáteis nem interoperáveis. AWS KMS não publica o formato de texto cifrado simétrico exigido pela portabilidade, e o formato pode mudar sem aviso prévio.

- AWS KMS não é possível descriptografar textos cifrados simétricos que você criptografa fora AWS, mesmo se você usar material de chave que tenha importado.
- AWS KMS não suporta a descriptografia de nenhum texto cifrado AWS KMS simétrico fora do AWS KMS, mesmo que o texto cifrado tenha sido criptografado em uma chave KMS com material de chave importado.
- As chaves do KMS com o mesmo material de chave importado não são interoperáveis. O texto cifrado simétrico que AWS KMS gera um texto cifrado específico para cada chave KMS. Esse formato de texto cifrado garante que somente a chave do KMS que criptografou os dados poderá descriptografá-los.

Além disso, você não pode usar nenhuma AWS ferramenta, como a [criptografia do lado do cliente AWS Encryption SDK no Amazon S3](#), para descriptografar textos cifrados simétricos. AWS KMS

Como resultado, você não pode usar chaves com material de chave importado para apoiar acordos de custódia de chaves em que um terceiro autorizado com acesso condicional ao material de chaves possa decifrar determinados textos cifrados fora do. AWS KMS Para oferecer suporte à garantia de chave, use o [AWS Encryption SDK](#) para criptografar sua mensagem em uma chave que seja independente do AWS KMS.

Proteger o material de chave importada

O material de chave importada é protegido em trânsito e em repouso. Antes de importar o material da chave, você criptografa (ou “empacota”) o material da chave com a chave pública de um par de chaves RSA gerado nos módulos de segurança de AWS KMS hardware (HSMs) validados pelo Programa de Validação do Módulo Criptográfico [FIPS 140-3](#). É possível criptografar o material da chave diretamente com a chave pública de empacotamento ou criptografar o material da chave com uma chave simétrica AES e, em seguida, criptografar a chave simétrica AES com a chave pública RSA.

Após o recebimento, AWS KMS descriptografa o material da chave com a chave privada correspondente em um AWS KMS HSM e o recriptografa sob uma chave simétrica AES que existe somente na memória volátil do HSM. O material de chave nunca sai do HSM em texto sem formatação. Ele é descriptografado somente enquanto está em uso e somente dentro dele. AWS KMS HSMs

O uso da chave do KMS com material de chave importado é determinado exclusivamente pelas [políticas de controle de acesso](#) que você define na chave do KMS. Além disso, é possível usar [alias](#) e [tags](#) para identificar e [controlar o acesso](#) à chave do KMS. É possível [habilitar e desabilitar](#) a chave, [visualizar](#) e [monitorar](#) a chave usando serviços como o AWS CloudTrail.

No entanto, você mantém a única cópia à prova de falhas do seu material de chave. Em troca dessa medida extra de controle, você é responsável pela durabilidade e disponibilidade geral do material chave importado. AWS KMS foi projetado para manter o material chave importado altamente disponível. Mas AWS KMS não mantém a durabilidade do material chave importado no mesmo nível do material chave AWS KMS gerado.

Essa diferença em durabilidade é importante nos seguintes casos:

- Quando você [define um prazo de validade](#) para o material de chaves importado, AWS KMS exclui o material de chaves depois que ele expira. AWS KMS não exclui a chave KMS nem seus metadados. Você pode [criar um CloudWatch alarme da Amazon](#) que o notifique quando o material de chave importado estiver se aproximando da data de expiração.

Você não pode excluir o material de chave AWS KMS gerado para uma chave KMS e não pode definir que o material de AWS KMS chave expire.

- Quando você [exclui manualmente o material da chave importada](#), AWS KMS exclui o material da chave, mas não exclui a chave KMS ou seus metadados. Por outro lado, o [agendamento da exclusão da chave](#) requer um período de espera de 7 a 30 dias, após o qual exclui AWS KMS permanentemente a chave KMS, seus metadados e seu material de chaves.
- No caso improvável de certas falhas em toda a região que afetem AWS KMS (como perda total de energia), AWS KMS não é possível restaurar automaticamente o material de chave importado. No entanto, AWS KMS pode restaurar a chave KMS e seus metadados.

Você deve reter uma cópia do material de chave importado fora AWS de um sistema que você controla. Recomendamos armazenar uma cópia exportável do material de chave importado em um sistema de gerenciamento de chaves, como um HSM. Como prática recomendada, você deve armazenar uma referência ao ARN da chave do KMS e ao ID do material de chave gerado pelo

AWS KMS junto com a cópia exportável do material de chave. Se o material de chave importado for excluído ou expirar, a chave do KMS associada se tornará inutilizável até que você reimporte o mesmo material de chave. Se o material de chave importada for perdido permanentemente, todo texto cifrado criptografado sob a chave do KMS será irrecoverável.

Important

As chaves de criptografia simétricas podem ter vários materiais de chave associados a elas. Toda a chave KMS se torna inutilizável assim que você exclui qualquer um desses materiais de chave ou se algum desses materiais de chave expirar (a menos que o material de chave excluído ou expirado seja ou). `PENDING_ROTATION` `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` Você deve reimportar qualquer material de chave expirado ou excluído associado a essa chave antes que a chave se torne utilizável para operações criptográficas.

Chaves do KMS em um repositório de chaves do CloudHSM

Você pode criar, visualizar, gerenciar, usar e programar a exclusão do AWS KMS keys em um armazenamento de AWS CloudHSM chaves. Os procedimentos são muito semelhantes aos usados em outras chaves do KMS. A única diferença é que você especifica um armazenamento de AWS CloudHSM chaves ao criar a chave KMS. Em seguida, AWS KMS cria material de chave não extraível para a chave KMS no AWS CloudHSM cluster associado ao AWS CloudHSM armazenamento de chaves. Quando você usa uma chave KMS em um armazenamento de AWS CloudHSM chaves, as [operações criptográficas](#) são executadas HSMs no cluster.

Recursos compatíveis

Além dos procedimentos discutidos nesta seção, você pode fazer o seguinte com as chaves KMS em um AWS CloudHSM armazenamento de chaves:

- Usar políticas de chaves, políticas do IAM e concessões para [autorizar o acesso](#) às chaves do KMS.
- [Habilite e desabilite](#) as chaves do KMS.
- Atribua [etiquetas](#), crie [alias](#) e use o controle de acesso por atributo (ABAC) para autorizar o acesso às chaves do KMS.
- Use as chaves do KMS para executar as seguintes operações de criptografia:
 - [Encrypt](#)

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

As operações que geram pares de chaves de dados assimétricos, [GenerateDataKeyPair](#) e [GenerateDataKeyPairWithoutPlaintext](#), não são suportadas em armazenamentos de chaves personalizadas.

- Use as chaves do KMS com [serviços da AWS que se integram ao AWS KMS](#) e oferecem suporte a chaves gerenciadas pelo cliente.
- Acompanhe o uso de suas chaves KMS nos [AWS CloudTrail registros](#) e nas [ferramentas de CloudWatch monitoramento da Amazon](#).

Atributos não compatíveis

- AWS CloudHSM os armazenamentos de chaves oferecem suporte somente a chaves KMS de criptografia simétrica. Você não pode criar chaves HMAC KMS, chaves KMS assimétricas ou pares de chaves de dados assimétricos em um armazenamento de chaves. AWS CloudHSM
- Você não pode [importar material de chaves](#) para uma chave KMS em um armazenamento de AWS CloudHSM chaves. AWS KMS gera o material chave para a chave KMS no AWS CloudHSM cluster.
- Você não pode ativar ou desativar a [rotação automática](#) do material da chave para uma chave KMS em um armazenamento de AWS CloudHSM chaves.

Usando chaves KMS em um armazenamento de AWS CloudHSM chaves

Ao usar sua chave KMS em uma solicitação, identifique a chave KMS por seu ID ou alias; você não precisa especificar o armazenamento de AWS CloudHSM chaves ou o cluster. AWS CloudHSM A resposta inclui os mesmos campos que são retornados para qualquer chave do KMS de criptografia simétrica.

No entanto, quando você usa uma chave KMS em um armazenamento de AWS CloudHSM chaves, a operação criptográfica é executada inteiramente dentro do AWS CloudHSM cluster associado ao armazenamento de AWS CloudHSM chaves. A operação usa o material de chave no cluster associado à chave do KMS escolhida.

Para tornar isso possível, as seguintes condições são necessárias.

- O [estado de chave](#) da chave do KMS deve ser Enabled. Para encontrar o estado da chave, use o campo Status no [AWS KMS console](#) ou o KeyState campo na [DescribeKey](#) resposta.

- O armazenamento de AWS CloudHSM chaves deve estar conectado ao AWS CloudHSM cluster. Seu status no [AWS KMS console](#) ou ConnectionState na [DescribeCustomKeyStores](#) resposta deve ser CONNECTED.
- O AWS CloudHSM cluster associado ao armazenamento de chaves personalizadas deve conter pelo menos um HSM ativo. Para encontrar o número de ativos HSMs no cluster, use o [AWS KMS console](#), o AWS CloudHSM console ou a [DescribeClusters](#) operação.
- O AWS CloudHSM cluster deve conter o material chave da chave KMS. Se o material de chaves foi excluído do cluster ou um HSM foi criado de um backup que não incluía o material de chaves, haverá falha na operação de criptografia.

Se essas condições não forem atendidas, a operação criptográfica falhará e AWS KMS retornará uma `KMSInvalidStateException` exceção. Normalmente, você só precisa [reconectar o armazenamento de AWS CloudHSM chaves](#). Para obter ajuda adicional, consulte [Como corrigir uma chave do KMS com falha](#).

Ao usar as chaves KMS em um AWS CloudHSM armazenamento de chaves, esteja ciente de que as chaves KMS em cada AWS CloudHSM armazenamento de chaves compartilham uma [cota de solicitação de armazenamento de chaves personalizada](#) para operações criptográficas. Se você exceder a cota, AWS KMS retorna a `ThrottlingException`. Se o AWS CloudHSM cluster associado ao armazenamento de AWS CloudHSM chaves estiver processando vários comandos, incluindo aqueles não relacionados ao armazenamento de AWS CloudHSM chaves, você poderá obter um `ThrottlingException` com uma taxa ainda menor. Se você receber uma `ThrottlingException` para qualquer solicitação, diminua a sua taxa de solicitações e tente os comandos novamente. Para obter detalhes sobre a cota de solicitações do armazenamento de chaves personalizado, consulte [Cotas de solicitação de armazenamento de chaves personalizadas](#).

Saiba mais

- Para saber mais sobre as AWS CloudHSM principais lojas, consulte [AWS CloudHSM lojas principais](#).
- Para criar chaves KMS em um armazenamento de AWS CloudHSM chaves, consulte [Criar uma chave KMS em um armazenamento de AWS CloudHSM chaves](#).
- Para identificar e visualizar as chaves KMS em um armazenamento de AWS CloudHSM chaves, consulte [Identificar chaves KMS em lojas de AWS CloudHSM chaves](#).
- Para encontrar chaves KMS e material de chaves em um armazenamento de AWS CloudHSM chaves, consulte [Encontre chaves KMS e material de chaves em um armazenamento de AWS CloudHSM chaves](#).

- Para saber mais sobre considerações especiais para excluir chaves KMS em um armazenamento de chaves, consulte [Excluindo AWS CloudHSM chaves KMS de um armazenamento de chaves](#). AWS CloudHSM

Chaves do KMS em repositórios de chaves externos

Para criar, visualizar, gerenciar, usar e programar a exclusão de chaves do KMS em um armazenamento de chaves externas, use procedimentos muito semelhantes aos usados para outras chaves do KMS. No entanto, ao criar uma chave do KMS em um armazenamento de chaves externas, especifique um [armazenamento de chaves externas](#) e uma [chave externa](#). Quando você usa uma chave do KMS em um armazenamento de chaves externas, [as operações de criptografia e descriptografia](#) são executadas pelo gerenciador de chaves externas usando a chave externa especificada.

AWS KMS não pode criar, visualizar, atualizar ou excluir nenhuma chave criptográfica em seu gerenciador de chaves externo. AWS KMS nunca acessa diretamente seu gerenciador de chaves externo ou qualquer chave externa. Todas as solicitações de operações de criptografia são mediadas por seu [proxy de armazenamento de chaves externas](#). Para usar uma chave do KMS em um armazenamento de chaves externas, o armazenamento de chaves externas que hospeda a chave do KMS deve estar [conectado](#) ao proxy de armazenamento de chaves externas.

Recursos compatíveis

Além dos procedimentos discutidos nesta seção, você pode fazer o seguinte com chaves do KMS de um armazenamento de chaves externas:

- Use [políticas de chaves](#), [políticas do IAM](#) e [concessões](#) para autorizar o acesso às chaves do KMS.
- [Habilite e desabilite](#) as chaves do KMS. Essas ações não afetam a chave externa no gerenciador de chaves externas.
- Atribua [etiquetas](#), crie [alias](#)es e use o [controle de acesso por atributo](#) (ABAC) para autorizar o acesso às chaves do KMS.
- Use as chaves do KMS para executar as seguintes operações de criptografia:
 - [Encrypt](#)
 - [Decrypt](#)
 - [GenerateDataKey](#)
 - [GenerateDataKeyWithoutPlaintext](#)

- [ReEncrypt](#)

As operações que geram pares de chaves de dados assimétricos, [GenerateDataKeyPair](#) e [GenerateDataKeyPairWithoutPlaintext](#), não são suportadas em armazenamentos de chaves personalizadas.

- Usar as chaves do KMS com [Serviços da AWS que se integram ao AWS KMS](#) e oferecem suporte a [chaves gerenciadas pelo cliente](#).

Atributos não compatíveis

- Armazenamentos de chaves externas são compatíveis apenas com [chaves do KMS de criptografia simétrica](#). Você não pode criar chaves do KMS de HMAC ou assimétricas em um armazenamento de chaves externas.
- [GenerateDataKeyPair](#) e não [GenerateDataKeyPairWithoutPlaintext](#) são compatíveis com chaves KMS em um armazenamento de chaves externo.
- Você não pode usar um [AWS::KMS::Key CloudFormation modelo](#) para criar um armazenamento de chaves externo ou uma chave KMS em um armazenamento de chaves externo.
- As [chaves multirregionais](#) não são compatíveis com o armazenamento de chaves externas.
- As chaves do KMS com [material de chave importado](#) não são compatíveis com o armazenamento de chaves externas.
- A [alternância automática de chaves](#) não é compatível com chaves do KMS em armazenamentos de chaves externas.

Usar chaves do KMS em um repositório de chaves externo

Ao usar sua chave do KMS em uma solicitação, identifique a chave do KMS pelo [ID de chave](#), [ARN de chave](#), [alias ou ARN do alias](#). Não é necessário especificar o armazenamento de chaves externas. A resposta inclui os mesmos campos que são retornados para qualquer chave do KMS de criptografia simétrica. Porém, quando você usa uma chave do KMS em um armazenamento de chaves externas, as operações de criptografia e descryptografia são executadas pelo gerenciador de chaves externas usando a chave externa que está associada à chave do KMS.

[Para garantir que o texto cifrado criptografado por uma chave KMS em um armazenamento de chaves externo seja pelo menos tão seguro quanto qualquer texto cifrado criptografado por uma chave KMS padrão, use criptografia dupla. AWS KMS](#) Os dados são primeiro criptografados AWS KMS usando material AWS KMS chave. Em seguida, ele é criptografado pelo gerenciador de chaves externas usando a chave externa para a chave do KMS. Para descryptografar texto cifrado com criptografia dupla, o texto cifrado é primeiro descryptografado pelo gerenciador de chaves

externas usando a chave externa para a chave do KMS. Em seguida, ele é descriptografado AWS KMS usando o material de AWS KMS chave da chave KMS.

Para tornar isso possível, as seguintes condições são necessárias.

- O [estado de chave](#) da chave do KMS deve ser Enabled. Para encontrar o estado da chave, consulte o campo Status das chaves gerenciadas pelo cliente, o [AWS KMS console](#) ou o KeyState campo na [DescribeKey](#) resposta.
- O armazenamento de chaves externas que hospeda a chave do KMS deve estar conectado ao [proxy de armazenamento de chaves externas](#), ou seja, o [estado da conexão](#) do armazenamento de chaves externas deve ser CONNECTED.

Você pode ver o estado da conexão na página Armazenamentos externos de chaves no AWS KMS console ou na [DescribeCustomKeyStores](#) resposta. O estado de conexão do armazenamento de chaves externas também é exibido na página de detalhes da chave do KMS no console do AWS KMS. Na página de detalhes, escolha a guia Cryptographic configuration (Configuração criptográfica) e veja o campo Connection state (Estado da conexão) na seção Custom key store (Armazenamento de chaves personalizado).

Se o estado da conexão for DISCONNECTED, primeiro é necessário conectá-lo. Se o estado da conexão for FAILED, você deverá resolver o problema, desconectar o armazenamento de chaves externas e conectá-lo. Para instruções, consulte [Conectar e desconectar repositórios de chaves externos](#).

- O proxy de armazenamento de chaves externas deve ser capaz de encontrar a chave externa.
- A chave externa deve estar habilitada e deve executar criptografia e descriptografia.

O status da chave externa é independente e não é afetado por alterações no [estado da chave](#) da chave do KMS, inclusive habilitar e desabilitar a chave do KMS. Da mesma forma, desabilitar ou excluir a chave externa não alterará o estado da chave do KMS, mas as operações de criptografia que usam a chave do KMS associada falharão.

Se essas condições não forem atendidas, a operação criptográfica falhará e AWS KMS retornará uma `KMSInvalidStateException` exceção. Talvez seja necessário [reconectar o armazenamento de chaves externas](#) ou usar as ferramentas do gerenciador de chaves externas para reconfigurar ou reparar a chave externa. Para obter ajuda adicional, consulte [the section called “Solução de problemas de armazenamentos de chaves externas”](#).

Ao usar as chaves do KMS em um armazenamento de chaves externas, esteja ciente de que as chaves do KMS em cada armazenamento de chaves externas compartilha uma [cota de](#)

[solicitações de armazenamento de chaves personalizado](#) para operações de criptografia. Se você exceder a cota, AWS KMS retorna a `ThrottlingException`. Para obter detalhes sobre a cota de solicitações do armazenamento de chaves personalizado, consulte [Cotas de solicitação de armazenamento de chaves personalizadas](#).

Saiba mais

- Para saber mais sobre repositórios de chaves externos, consulte [Armazenamentos de chaves externas](#).
- Para saber mais sobre material de chave em repositórios de chaves externos, consulte [Chave externa](#).
- Para criar chaves do KMS em um repositório de chaves externo, consulte [Criar uma chave do KMS em repositórios de chaves externos](#).
- Para identificar e visualizar chaves do KMS em um repositório de chaves externo, consulte [Identificar chaves do KMS em repositórios de chaves externos](#).
- Para saber mais sobre considerações especiais sobre a exclusão de chaves do KMS em um repositório de chaves externo, consulte [Excluir chaves do KMS de um repositório de chaves externo](#).

AWS KMS fundamentos da criptografia

AWS KMS usa algoritmos criptográficos configuráveis para que o sistema possa migrar rapidamente de um algoritmo ou modo aprovado para outro. O conjunto padrão inicial de algoritmos criptográficos foi selecionado a partir de algoritmos do Padrão Federal de Processamento de Informações (aprovados pelo Federal Information Processing Standard – FIPS) para suas propriedades de segurança e performance.

Para obter mais informações sobre os algoritmos criptográficos preferenciais e aceitáveis, consulte [Algoritmos criptográficos com suporte](#).

Entropia e geração de números aleatórios

AWS KMS a geração de chaves é realizada no AWS KMS HSMs. Eles HSMs implementam um gerador híbrido de números aleatórios que usa o [NIST SP800-90A Deterministic Random Bit Generator \(DRBG\) CTR_DRBG using AES-256](#). Ele é preparado com um gerador de bits aleatórios não determinísticos com 384 bits de entropia e atualizado com entropia adicional para fornecer resistência de previsão em cada chamada de material criptográfico.

Operações de chave simétrica (somente criptografia)

Todos os comandos de criptografia de chave simétrica usados HSMs usam os [Advanced Encryption Standards \(AES\)](#), no [Galois Counter Mode \(GCM\)](#) usando chaves de 256 bits. As chamadas análogas para descriptografar usam a função inversa.

AES-GCM é um esquema de criptografia autenticado. Além de criptografar texto sem formatação para produzir texto cifrado, ele calcula uma tag de autenticação sobre o texto cifrado e quaisquer dados adicionais para os quais a autenticação é necessária (dados autenticados adicionalmente – AAD). A tag de autenticação ajuda a garantir que os dados são da fonte suposta e que o texto cifrado e os AAD não foram modificados.

Frequentemente, AWS omite a inclusão do AAD em nossas descrições, especialmente quando se refere à criptografia de chaves de dados. Está implícito pelo texto circundante nesses casos que a estrutura a ser criptografada é particionada entre o texto simples a ser criptografado e os AAD de texto simples a serem protegidos.

AWS KMS fornece uma opção para você importar o material chave para um, em AWS KMS key vez de AWS KMS depender da geração do material chave. Esse material de chave importado pode ser criptografado usando [RSAES-OAEP](#) para proteger a chave durante o transporte para o HSM. AWS KMS Os pares de chaves RSA são gerados em. AWS KMS HSMs O material da chave importada é descriptografado em um AWS KMS HSM e recriptografado no AES-GCM antes de ser armazenado pelo serviço.

Operações de chave assimétrica (criptografia, assinatura digital e verificação de assinatura)

AWS KMS suporta o uso de operações de chave assimétrica para operações de criptografia, assinatura digital e contrato de chaves. Operações de chave assimétrica contam com um par de chave pública e chave privada relacionadas matematicamente que pode ser usado para criptografia e descriptografia, assinatura e verificação de assinatura ou derivação de segredos compartilhados. A chave privada nunca sai AWS KMS sem criptografia. Você pode usar a chave pública interna AWS KMS chamando as operações da AWS KMS API ou fazer o download da chave pública e usá-la fora dela AWS KMS.

AWS KMS suporta as seguintes cifras assimétricas.

- RSA-OAEP (para criptografia) & RSA-PSS e RSA-PKCS- #1 -v1_5 (para assinatura e verificação): Suporta comprimentos de chave RSA (em bits): 2048, 3072 e 4096 para diferentes requisitos de segurança.
- Elliptic Curve (ECC): usada para assinar e verificar ou derivar segredos compartilhados, mas não para ambos. Suporta curvas ECC: NIST P256, P384, P521, SECP 256k1, Ed25519.
- ML-DSA: usado para assinatura e verificação. As especificações de chave ML-DSA compatíveis são: ML_DSA_44, ML_DSA_65 e ML_DSA_87.
- SM2 (Somente regiões da China) — Usado para criptografia e descryptografia, assinatura e verificação ou derivação de segredos compartilhados, mas você deve escolher um uso de chave. Suporta SM2 PKE para criptografia e SM2 DSA para assinatura.

Funções de derivação de chave

Uma função de derivação de chave é usada para derivar chaves adicionais de um segredo ou chave inicial. AWS KMS usa uma função de derivação de chave (KDF) para derivar chaves por chamada para cada criptografia sob um. AWS KMS key [Todas as operações do KDF usam o KDF no modo contador usando HMAC \[\] com \[\] FIPS197. SHA256 FIPS180](#) A chave derivada de 256 bits é usada com AES-GCM para criptografar ou descryptografar dados e chaves do cliente.

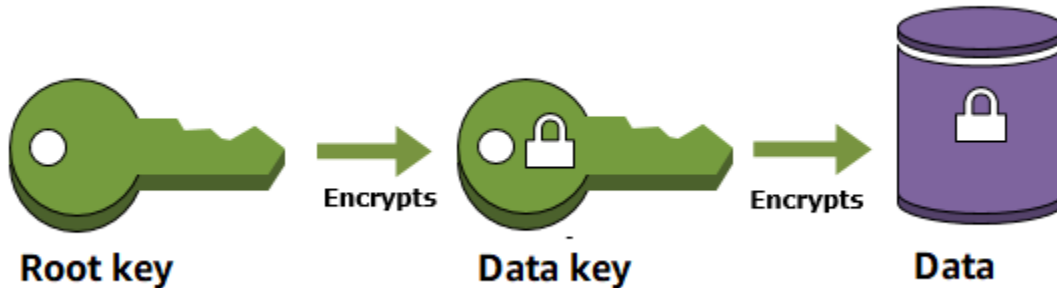
AWS KMS uso interno de assinaturas digitais

Assinaturas digitais também são usadas para autenticar comandos e comunicações entre entidades do AWS KMS . Todas as entidades de serviço têm um par de chaves de algoritmo de assinatura digital de curva elíptica (ECDSA). Elas executam o ECDSA conforme definido na [Uso de algoritmos de criptografia de curva elíptica \(ECC\) na sintaxe de mensagem criptográfica \(CMS\)](#) e X9.62-2005: Criptografia de chave pública para o setor de serviços financeiros: o algoritmo de assinatura digital de curva elíptica (ECDSA). As entidades usam o algoritmo de hash seguro definido nas [Publicações de Normas Federais de Processamento de Informações, FIPS PUB 180-4](#), conhecido como. SHA384 As chaves são geradas na curva secp384r1 (NIST-P384).

Criptografia de envelope

Quando você criptografa seus dados, os dados são protegidos, mas é necessário proteger a chave de criptografia. Uma estratégia é para criptografá-la. Criptografia de envelope é a prática de criptografar dados de texto simples com uma chave de dados e criptografar a chave de dados em outra chave.

Você pode até mesmo criptografar a chave de criptografia dos dados em outra chave de criptografia e criptografar essa chave de criptografia em outra chave de criptografia. Mas em algum momento deverá manter uma chave em texto simples para que possa descriptografar as chaves e seus dados. Essa chave de criptografia em texto simples de nível superior é chamada de chave raiz.



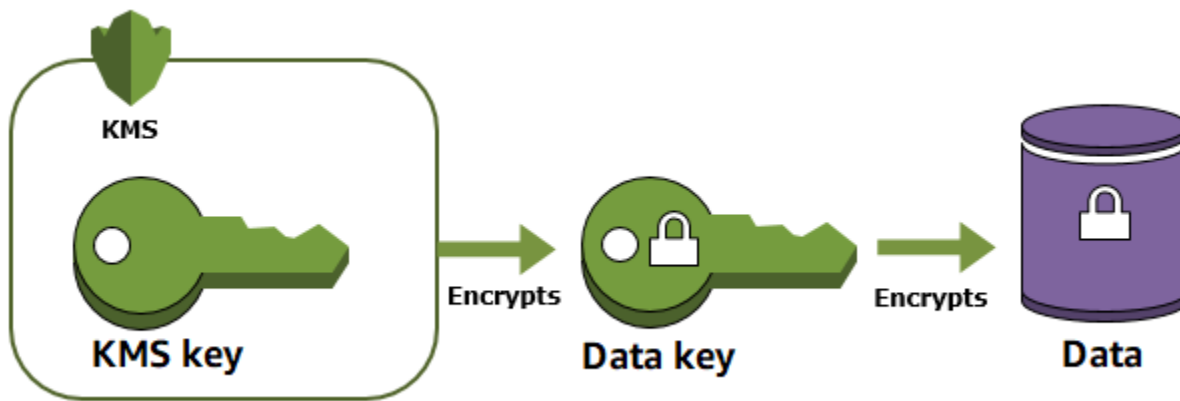
AWS KMS ajuda você a proteger suas chaves de criptografia armazenando-as e gerenciando-as com segurança. A chave raiz armazenada AWS KMS, conhecida como AWS KMS keys, nunca deixa os módulos de segurança de [hardware validados pelo AWS KMS FIPS 140-3 Security Level 3](#) sem criptografia. Para usar uma chave KMS, você deve ligar AWS KMS.

Uma construção básica usada em muitos sistemas criptográficos é a criptografia de envelope. A criptografia de envelope usa duas ou mais chaves criptográficas para proteger uma mensagem. Normalmente, uma chave é derivada a partir de uma chave estática de longo prazo k , e outra chave é uma chave por mensagem $msgKey$, que é gerada para criptografar a mensagem. O envelope é formado criptografando a mensagem: $\text{texto cifrado} = \text{Criptografia}(msgKey, \text{mensagem})$. Em seguida, a chave de mensagem é criptografada com a chave estática de longo prazo: $encKey = \text{Criptografia}(k, msgKey)$. Por fim, os dois valores ($encKey$, texto cifrado) são empacotados em uma única estrutura, ou mensagem criptografada de envelope.

O destinatário, com acesso a k , pode abrir a mensagem envolta primeiro descriptografando a chave criptografada e depois descriptografando a mensagem.

AWS KMS fornece a capacidade de gerenciar essas chaves estáticas de longo prazo e automatizar o processo de criptografia de envelopes de seus dados.

Além dos recursos de criptografia fornecidos no AWS KMS serviço, o [SDK de criptografia fornece bibliotecas de AWS criptografia](#) de envelopes do lado do cliente. Você pode usar essas bibliotecas para proteger os seus dados e as chaves de criptografia usadas para criptografar esses dados.



A criptografia de envelope oferece vários benefícios:

- Proteção de chaves de dados

Quando você criptografa uma chave de dados, não precisa se preocupar em armazenar a chave de dados criptografada porque a chave de dados é inerentemente protegida pela criptografia. A chave de dados criptografada pode ser armazenada com segurança junto com os dados criptografados.

- Criptografar os mesmos dados com várias chaves

As operações de criptografia podem ser demoradas, especialmente quando os dados que estão sendo criptografados são objetos grandes. Em vez de recriptografar dados brutos várias vezes com diferentes chaves, você pode recriptografar somente as chaves de dados que protegem os dados brutos.

- Combinação de pontos fortes de vários algoritmos

Em geral, os algoritmos de chave simétrica são mais rápidos e geram textos cifrados menores do que os algoritmos de chave pública. No entanto, os algoritmos de chave pública fornecem separação inerente de funções e gerenciamento de chaves mais fácil. A criptografia de envelope permite associar os pontos fortes de cada estratégia.

Operações criptográficas

Em AWS KMS, as operações criptográficas são operações de API que usam chaves KMS para proteger dados. Como as chaves KMS permanecem dentro AWS KMS, você deve ligar AWS KMS para usar uma chave KMS em uma operação criptográfica.

Para realizar operações criptográficas com chaves KMS, use o AWS SDKs, AWS Command Line Interface (AWS CLI) ou o. Ferramentas da AWS para PowerShell. Não é possível executar operações de criptografia no console do AWS KMS. Para obter exemplos de chamadas de operações de criptografia em várias linguagens de programação, consulte [Exemplos de código para AWS KMS usar AWS SDKs](#).

A tabela a seguir lista as operações AWS KMS criptográficas. Ela também mostra os requisitos de tipo de chave e [uso de chave](#) para chaves do KMS usadas na operação.

Operation	Tipo de chave	Uso da chave
Decrypt	Simétrico ou assimétrico	ENCRYPT_DECRYPT
DeriveSharedSecret	Assimétrica	KEY_AGREEMENT
Encrypt	Simétrico ou assimétrico	ENCRYPT_DECRYPT
GenerateDataKey	Simétrica	ENCRYPT_DECRYPT
GenerateDataKeyPair	Simétrica [1] Não é compatível com chaves do KMS em armazenamentos de chaves personalizados.	ENCRYPT_DECRYPT
GenerateDataKeyPairWithoutPlaintext	Simétrica [1] Não é compatível com chaves do KMS em armazenamentos de chaves personalizados.	ENCRYPT_DECRYPT
GenerateDataKeyWithoutPlaintext	Simétrica	ENCRYPT_DECRYPT

Operation	Tipo de chave	Uso da chave
GenerateMac	HMAC	GENERATE_VERIFY_MAC
GenerateRandom	N/D. Essa operação não usa uma chave do KMS.	N/D
ReEncrypt	Simétrico ou assimétrico	ENCRYPT_DECRYPT
Sign	Assimétrica	SIGN_VERIFY
Verificar	Assimétrica	SIGN_VERIFY
VerifyMac	HMAC	GENERATE_VERIFY_MAC

[1] Gera um par de chaves de dados assimétricas que é protegido por uma chave do KMS de criptografia simétrica.

Para obter informações sobre as permissões para operações de criptografia, consulte a [the section called “Referência de permissões”](#).

Para tornar AWS KMS responsivo e altamente funcional para todos os usuários, AWS KMS estabelece cotas no número de operações criptográficas chamadas em cada segundo. Para obter mais detalhes, consulte [the section called “Cotas compartilhadas para operações de criptografia”](#).

Algoritmos criptográficos com suporte

Algoritmos criptográficos

As tabelas a seguir resumem os algoritmos criptográficos, as cifras, os modos e os tamanhos de chave AWS implantados em seus serviços para proteger seus dados. Eles não devem ser considerados uma lista exaustiva de todas as opções de criptografia disponíveis na AWS. Os algoritmos se encaixam em duas categorias:

- Os algoritmos preferidos atendem aos padrões de AWS segurança e desempenho.

- Os algoritmos aceitáveis podem ser usados por compatibilidade em algumas aplicações, mas não são os preferenciais.

Criptografia assimétrica

A tabela a seguir lista os algoritmos assimétricos compatíveis para criptografia, acordo de chaves e assinaturas digitais.

Tipo	Algoritmo	Status
Criptografia	RSA-OAEP (módulos de 2048 ou 3072 bits)	Aceitável
Criptografia	HPKE (P-256 ou P-384, HKDF e AES-GCM)	Aceitável
Acordo de chave	ML-KEM-768 ou ML-KEM-1024	Preferencial (resistente a quântico)
Acordo de chave	ECDH(E) com P-384	Aceitável
Acordo de chave	ECDH(E) com P-256, P-521 ou X25519	Aceitável
Acordo de chave	ECDH(E) com brainpool P256r1, brainpoolP384r1 ou brainpoolP512r1	Aceitável
Assinaturas	ML-DSA-65 ou ML-DSA-87	Preferencial (resistente a quântico)
Assinaturas	SLH-DSA	Preferencial (assinatura com resistência quântica software/firmware)
Assinaturas	ECDSA com P-384	Aceitável
Assinaturas	ECDSA com P-256, P-521 ou Ed25519	Aceitável

Tipo	Algoritmo	Status
Assinaturas	RSA-2048 ou RSA-3072	Aceitável

Criptografia simétrica

A tabela a seguir lista algoritmos simétricos compatíveis para criptografia, criptografia autenticada e empacotamento de chaves.

Tipo	Algoritmo	Status
Criptografia autenticada	AES-GCM-256	Preferencial
Criptografia autenticada	AES-GCM-128	Aceitável
Criptografia autenticada	ChaCha20/Poliéster 1305	Aceitável
Modos de criptografia	AES-XTS-256 (para armazenamento em bloco)	Preferencial
Modos de criptografia	AES-CBC/CTR (modos não autenticados)	Aceitável
Chave de encapsulamento	AES-GCM-256	Preferencial
Chave de encapsulamento	AES-KW ou AES-KWP com chaves de 256 bits	Aceitável

Funções criptográficas

A tabela a seguir lista os algoritmos compatíveis para hash, derivação de chaves, autenticação de mensagens e hash de senhas.

Tipo	Algoritmo	Status
Hashing	SHA2-384	Preferencial
Hashing	SHA2-256	Aceitável

Tipo	Algoritmo	Status
Hashing	SHA3	Aceitável
Derivação de chaves	HKDF_Expand ou HKDF com -256 SHA2	Preferencial
Derivação de chaves	Modo de contador KDF com HMAC- -256 SHA2	Aceitável
Código de autenticação de mensagem	HMAC-384 SHA2	Preferencial
Código de autenticação de mensagem	HMAC-256 SHA2	Aceitável
Código de autenticação de mensagem	KMAC	Aceitável
Hash de senha	criptografar com SHA384	Preferencial
Hash de senha	PBKDF2	Aceitável

Para obter mais detalhes sobre algoritmos criptográficos implantados em AWS, consulte Algoritmos de [criptografia e. Serviços da AWS](#)

Acesso e permissões de chave do KMS

Para usar AWS KMS, você deve ter credenciais que AWS possam ser usadas para autenticar suas solicitações. As credenciais devem incluir permissões para acessar AWS recursos: AWS KMS keys e [alias](#)s. Nenhum AWS diretor tem permissões para uma chave KMS, a menos que essa permissão seja fornecida explicitamente e nunca negada. Não há permissão implícita ou automática para usar ou gerenciar uma chave do KMS.

Para controlar o acesso às suas chaves do KMS, você pode usar os seguintes mecanismos de política.

- [Política de chaves](#): cada chave do KMS tem uma política de chave. Esse é o mecanismo principal para controlar o acesso a uma chave do KMS. Você pode usar apenas a política de chaves para controlar o acesso, ou seja, o escopo completo de acesso à chave do KMS é definido em um único documento (a política de chaves). Para mais informações sobre como usar políticas de chaves, consulte [Políticas de chaves](#).
- [Políticas do IAM](#): você pode usar políticas do IAM combinadas com a política de chaves e concessões para controlar o acesso a uma chave do KMS. Com esse controle de acesso, você pode gerenciar todas as permissões para suas identidades do IAM no IAM. Para usar uma política do IAM para permitir o acesso a uma chave do KMS, a política de chaves deve explicitamente permitir isso. Para mais informações sobre como usar políticas do IAM, consulte [Políticas do IAM](#).
- [Concessões](#): você pode usar concessões combinadas com a política de chaves e políticas do IAM para permitir o acesso a uma chave do KMS. Controlar o acesso dessa maneira possibilita permitir o acesso à chave do KMS na política de chaves e permitir que as identidades deleguem o acesso para outras pessoas. Para obter mais informações sobre como usar concessões, consulte [Subsídios em AWS KMS](#).

Políticas de chave do KMS

A principal forma de gerenciar o acesso aos seus AWS KMS recursos é com políticas. Políticas são documentos que descrevem quais entidades principais podem acessar recursos específicos. As políticas anexadas a uma identidade do IAM são chamadas de políticas baseadas em identidade (ou políticas do IAM), e as políticas anexadas a outros tipos de recursos são chamadas de políticas de recursos. AWS KMS as políticas de recursos para chaves KMS são chamadas de políticas de chaves.

Todas as chaves do KMS têm uma política de chaves. Se você não fornecer um, AWS KMS cria um para você. A [política de chaves padrão](#) AWS KMS usada difere dependendo se você cria a chave no AWS KMS console ou usa a AWS KMS API. Recomendamos que você edite a política de chave padrão para se alinhar aos requisitos de [permissões de privilégios mínimos](#) da sua organização.

Você pode usar a política de chaves sozinha para controlar o acesso se a chave e o principal do IAM estiverem na mesma AWS conta, o que significa que o escopo completo do acesso à chave KMS é definido em um único documento (a política de chaves). No entanto, quando um chamador em uma conta precisar acessar uma chave em outra conta, você não poderá usar a política de chave sozinha para conceder acesso. No cenário entre contas, uma política do IAM deverá ser anexada ao usuário ou ao perfil do chamador permitindo explicitamente que o chamador faça a chamada de API.

Você também pode usar políticas do IAM juntamente com políticas de chave e concessões para controlar o acesso à uma chave do KMS. Para usar uma política do IAM para controlar o acesso a uma chave do KMS, a política de chave deve conceder permissão para a conta usar políticas do IAM. Você pode especificar uma [declaração de política de chave que habilite as políticas do IAM](#) ou pode [especificar as entidades principais permitidas](#) explicitamente na política de chave.

Ao escrever políticas, certifique-se de ter controles rígidos que restrinjam quem pode realizar as seguintes ações:

- Atualizar, criar e excluir políticas de chave do IAM e do KMS
- Anexar e desanexar políticas do IAM de usuários, perfis e grupos
- Anexar e desanexar políticas de chaves do KMS de suas chaves do KMS

Concessões de chave do KMS

Além do IAM e das principais políticas, AWS KMS apoia [subsídios](#). As concessões oferecem uma maneira flexível e poderosa de delegar permissões. Você pode usar concessões para emitir acesso por chave KMS com prazo determinado aos diretores do IAM em sua AWS conta ou em outras contas. AWS Recomendamos emitir acesso com limite de tempo se você não souber os nomes das entidades principais no momento em que as políticas foram criadas ou se as entidades principais que precisam acesso mudarem com frequência. A [entidade principal](#) receptora da concessão pode estar na mesma conta que a chave do KMS ou em outra conta. Se a entidade principal e a chave do KMS estiverem em contas diferentes, você deverá especificar uma política do IAM além da concessão. As concessões exigem gerenciamento adicional porque você precisa chamar uma API para criar a concessão e retirar ou revogar a concessão quando ela não for mais necessária.

Políticas-chave em AWS KMS

Uma política fundamental é uma política de recursos para um AWS KMS key. Políticas de chaves são a principal maneira de controlar o acesso a chaves do KMS. Cada chave do KMS deve ter exatamente uma política de chaves. As instruções na política de chaves determinam quem tem permissão para usar a chave do KMS e como eles podem usá-la. Você também pode usar [políticas do IAM](#) e [concessões](#) para controlar o acesso à chave do KMS, mas cada chave do KMS deve ter uma política de chaves.

Nenhum AWS diretor, incluindo o usuário raiz da conta ou o criador da chave, tem qualquer permissão para uma chave KMS, a menos que seja explicitamente permitida, e nunca negada, em uma política de chaves, política do IAM ou concessão.

A menos que a política de chaves permita isto explicitamente, você não pode usar políticas do IAM para autorizar o acesso a uma chave do KMS. Sem permissão da política de chaves, as políticas do IAM que autorizam permissões não têm efeito. (Você pode usar uma política do IAM para negar uma permissão para uma chave KMS sem permissão de uma política de chaves.) A política de chaves padrão habilita as políticas do IAM. Para habilitar políticas do IAM em sua política de chaves, adicione a declaração de política descrita em [Permite o acesso Conta da AWS e ativa as políticas do IAM](#).

Ao contrário das políticas do IAM, que são globais, as políticas de chaves são regionais. Uma política de chaves controla o acesso somente a uma chave do KMS na mesma região. Ela não tem efeito sobre as chaves do KMS em outras regiões.

Tópicos

- [Criação de uma política de chave](#)
- [Política de chaves padrão](#)
- [Visualizar uma política de chave](#)
- [Alterar uma política de chave](#)
- [Permissões para AWS serviços nas principais políticas](#)

Criação de uma política de chave

Você pode criar e gerenciar políticas de chaves no AWS KMS console ou usando operações de AWS KMS API [CreateKey](#), como [ReplicateKey](#), [PutKeyPolicy](#).

Quando você cria uma chave KMS no AWS KMS console, o console orienta você pelas etapas de criação de uma política de chaves com base na [política de chaves padrão do console](#). Quando você usa o CreateKey ou ReplicateKey APIs, se você não especificar uma política de chaves, eles aplicam a [política de chaves padrão para chaves criadas programaticamente](#). Ao usar a API PutKeyPolicy, você precisa especificar uma política de chaves.

Cada documento de política pode ter uma ou mais declarações de política. O exemplo a seguir mostra um documento válido de política de chaves com uma instrução de política.

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"
        }
      }
    }
  ]
}
```

Tópicos

- [Formato de política de chaves](#)
- [Elementos em uma política de chaves](#)
- [Exemplo de política de chave do](#)

Formato de política de chaves

Um documento de política de chaves deve estar de acordo com as seguintes regras:

- Até 32 kilobytes (32.768 bytes)
- O elemento `Sid` em uma instrução de política de chaves pode incluir espaços. (É proibido usar espaços no elemento `Sid` de um documento de política do IAM.)

Um documento de política de chaves pode incluir apenas os seguintes caracteres:

- Caracteres ASCII imprimíveis
- Caracteres imprimíveis no conjunto de caracteres Basic Latin e Latin-1 Supplement
- Os caracteres especiais de tabulação (`\u0009`), alimentação de linha (`\u000A`) e retorno de carro (`\u000D`)

Elementos em uma política de chaves

Um documento de política de chaves deve ter os elementos a seguir:

Versão

Especifica a versão do documento de política de chaves. Define a versão como 2012-10-17 (a versão mais recente).

Declaração

Inclui as instruções da política. Um documento de política de chaves deve ter pelo menos uma instrução.

Cada instrução de política de chaves pode consistir em até seis elementos. Os elementos `Effect`, `Principal`, `Action` e `Resource` são obrigatórios.

Sid

(Opcional) O identificador de instrução (`Sid`) é uma string arbitrária que você pode usar para descrever a instrução. O `Sid` em uma política de chaves pode incluir espaços. (Você não pode incluir espaços no elemento `Sid` de uma política do IAM.)

Efeito

(Obrigatório) Determina se as permissões devem ser permitidas ou negadas na instrução de política. Os valores válidos são Allow ou Deny. Se você não permitir explicitamente o acesso a uma chave do KMS, esse acesso será implicitamente negado. Também é possível negar explicitamente o acesso a uma chave do KMS. Você poderia fazer isso para garantir que um usuário não possa acessá-la, mesmo quando uma política diferente permite o acesso.

Entidade principal

(Obrigatório) A [entidade principal](#) é a identidade que recebe a especificação de permissões na instrução de política. Você pode especificar usuários do IAM Contas da AWS, funções do IAM e alguns AWS serviços como principais em uma política de chaves. [Grupos de usuários](#) do IAM não são uma entidade principal válida em nenhum tipo de política.

Um valor de asterisco, como "AWS": "*" representa todas as identidades da AWS em todas as contas.

Important

Não defina a entidade principal como um asterisco (*) em qualquer instrução de política de chave que permita permissões, a menos que você utilize [condições](#) para limitar a política de chave. Um asterisco dá a cada identidade em cada Conta da AWS permissão para usar a chave KMS, a menos que outra declaração de política a negue explicitamente. Usuários de outras Contas da AWS podem usar sua chave KMS sempre que tiverem permissões correspondentes em suas próprias contas.

Note

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Quando a entidade em uma declaração de política de chaves é uma [entidade principal da Conta da AWS](#) no formato `arn:aws:iam::111122223333:root`, a declaração da política não dá permissão a nenhuma entidade principal do IAM. Em vez disso, ele dá Conta da AWS

permissão para usar políticas do IAM para delegar as permissões especificadas na política de chaves. (Uma entidade principal no formato `arn:aws:iam::111122223333:root` não representa o [usuário raiz de conta da AWS](#), apesar do uso de “root” no identificador da conta. No entanto, a entidade principal da conta representa a conta e seus administradores, incluindo o usuário raiz da conta.)

Quando o principal é outro Conta da AWS ou seus diretores, as permissões são efetivas somente quando a conta é ativada na região com a chave e a política de chaves do KMS. Para obter informações sobre regiões não habilitadas por padrão (“regiões de adesão”), consulte [Gerenciar Regiões da AWS](#) em Referência geral da AWS.

Para permitir que outra Conta da AWS pessoa ou seus principais usem uma chave KMS, você deve fornecer permissão em uma política de chaves e em uma política do IAM na outra conta. Para obter detalhes, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

Ação

(Obrigatório) Especifica as operações de API que serão permitidas ou negadas. Por exemplo, a `kms:Encrypt` ação corresponde à operação AWS KMS [Criptografar](#). Você pode listar mais de uma ação em uma declaração de política. Para obter mais informações, consulte [Referência de permissões](#).

Note

Se o elemento `Action` necessário estiver ausente de uma declaração de política de chave, a declaração de política não terá efeito. Uma declaração de política de chaves sem um elemento `Action` não é aplicável a nenhuma chave do KMS.

Quando uma declaração de política chave não tem seu `Action` elemento, o AWS KMS console relata corretamente um erro, mas o [CreateKey](#) e [PutKeyPolicy](#) APIs bem-sucedido, mesmo que a declaração de política seja ineficaz.

Recurso

(Obrigatório) Em uma política de chaves, o valor do elemento Recurso é “*”, o que significa “esta chave do KMS”. O asterisco (“*”) identifica a chave do KMS à qual a política de chaves está associada.

 Note

Se o elemento Resource necessário estiver ausente de uma declaração de política de chave, a declaração de política não terá efeito. Uma declaração de política de chaves sem um elemento Resource não se aplica a nenhuma chave do KMS. Quando uma declaração de política chave não tem seu Resource elemento, o AWS KMS console relata corretamente um erro, mas o [CreateKey](#) e [PutKeyPolicy](#) APIs bem-sucedido, mesmo que a declaração de política seja ineficaz.

Condição

(Opcional) As condições especificam os requisitos que devem ser atendidos para que a política de chaves entre em vigor. Com condições, AWS pode avaliar o contexto de uma solicitação de API para determinar se a declaração de política se aplica ou não.

Para especificar condições, você usa chaves de condição predefinidas. AWS KMS suporta chaves de [condição AWS globais e chaves de AWS KMS condição](#). Para oferecer suporte ao controle de acesso baseado em atributos (ABAC), AWS KMS fornece chaves de condição que controlam o acesso a uma chave KMS com base em tags e aliases. Para obter detalhes, consulte [ABAC para AWS KMS](#).

O formato de uma condição é:

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

como:

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

Para obter mais informações sobre a sintaxe da AWS política, consulte [Referência de política AWS do IAM](#) no Guia do usuário do IAM.

Exemplo de política de chave do

O exemplo a seguir mostra uma política de chaves completa para uma chave do KMS de criptografia simétrica. Você pode usá-lo para referência ao ler sobre os principais conceitos de política neste

capítulo. Esta política de chaves combina as declarações da política de exemplo da seção anterior de [política de chaves padrão](#) em uma política de chaves única que faz o seguinte:

- Permite Conta da AWS, por exemplo, 111122223333, acesso total à chave KMS. Há permissão para que a conta e seus administradores, incluindo o usuário raiz da conta (para emergências), usem políticas do IAM na conta para conceder acesso à chave do KMS.
- Permite que o perfil do IAM ExampleAdminRole administre a chave do KMS.
- Permite que o perfil do IAM ExampleUserRole use a chave do 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"
        }
    }
}
]

```

```
}
```

Política de chaves padrão

Ao criar uma chave do KMS, é possível especificar a política de chaves da nova chave do KMS. Se você não fornecer um, AWS KMS cria um para você. A política de chaves padrão AWS KMS usada difere dependendo se você cria a chave no AWS KMS console ou usa a AWS KMS API.

Política de chaves padrão ao criar uma chave do KMS de forma programática

Quando você cria uma chave KMS programaticamente com a [AWS KMS API](#) (inclusive usando o [AWS SDKs](#), [AWS Command Line Interface](#) ou [Ferramentas da AWS para PowerShell](#)) e não especifica uma política de chaves, AWS KMS aplica uma política de chaves padrão muito simples. Essa política de chaves padrão tem uma declaração de política que dá ao Conta da AWS proprietário da chave KMS permissão para usar políticas do IAM para permitir acesso a todas as AWS KMS operações na chave KMS. Para obter mais informações sobre essa declaração de política, consulte [Permite o acesso Conta da AWS e ativa as políticas do IAM](#).

Política de chaves padrão quando você cria uma chave KMS com o Console de gerenciamento da AWS

Quando você [cria uma chave KMS com o Console de gerenciamento da AWS](#), a política de chaves começa com a declaração de política que [permite o acesso Conta da AWS e ativa as políticas do IAM](#). Em seguida, o console adiciona uma declaração de administradores de chaves, uma declaração de usuários-chave e (para a maioria dos tipos de chaves) uma declaração que [permite que os diretores usem a chave KMS com outros serviços. AWS](#) Você pode usar os recursos do AWS KMS console para especificar os usuários do IAM, as funções do IAM e Contas da AWS quem são os principais administradores e aqueles que são os principais usuários (ou ambos).

Permissões

- [Permite o acesso Conta da AWS e ativa as políticas do IAM](#)
- [Permite que administradores de chaves administrem a chave do KMS](#)
- [Permite que os usuários de chaves usem a chave do KMS](#)
 - [Permite que usuários de chaves usem uma chave do KMS para operações de criptografia](#)
 - [Permite que os principais usuários usem a chave KMS com serviços AWS](#)

Permite o acesso Conta da AWS e ativa as políticas do IAM

A declaração de política de chave padrão a seguir é fundamental.

- Ele dá ao Conta da AWS proprietário da chave KMS acesso total à chave KMS.

Diferentemente AWS de outras políticas de recursos, uma política de AWS KMS chaves não dá permissão automática à conta ou a nenhuma de suas identidades. Para dar permissão aos administradores de conta, a política de chaves deve incluir uma declaração explícita que forneça essa permissão, como esta.

- Ele permite que a conta use políticas do IAM para permitir acesso à chave do KMS, além da política de chaves.

Sem essa permissão, as políticas do IAM que permitem o acesso à chave são ineficazes, embora as políticas do IAM que negam acesso à chave ainda sejam efetivas.

- Ela reduz o risco de a chave se tornar não gerenciável, dando permissão de controle de acesso aos administradores da conta, incluindo o usuário raiz da conta, que não pode ser excluído.

A declaração de política de chaves a seguir é toda a política de chaves padrão para chaves do KMS criadas programaticamente. É a primeira declaração de política na política de chaves padrão para chaves KMS criadas no AWS KMS console.

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

Permite que políticas do IAM permitam o acesso à chave do KMS.

A declaração de política de chaves mostrada acima dá ao Conta da AWS proprietário da chave permissão para usar políticas do IAM, bem como políticas de chave, para permitir todas as ações (kms : *) na chave KMS.

A entidade principal nesta declaração de política de chave é a [entidade principal da conta](#), que é representada por um ARN neste formato: `arn:aws:iam::account-id:root`. O principal da conta representa a AWS conta e seus administradores.

Quando a entidade principal em uma instrução de política de chave é a entidade principal da conta, a instrução de política não concede permissões para o uso da chave do KMS a nenhuma entidade principal do IAM. Em vez disso, ela permite que a conta use políticas do IAM para delegar as permissões especificadas na política de chaves. Essa declaração de política de chaves padrão permite que a conta use políticas do IAM para delegar permissão para todas as ações (`kms : *`) na chave do KMS.

reduz o risco de a chave do KMS perder a capacidade de gerenciamento.

Ao contrário AWS de outras políticas de recursos, uma política de AWS KMS chaves não dá permissão automática à conta ou a qualquer um de seus diretores. Para dar permissão a qualquer entidade principal, incluindo a [entidade principal da conta](#), você deve usar uma declaração de política chave que forneça a permissão explicitamente. Você não precisa dar à entidade principal da conta, ou a qualquer entidade principal, acesso à chave do KMS. No entanto, dar acesso à entidade principal da conta ajuda você a evitar que a chave se torne não gerenciável.

Por exemplo, suponha que você crie uma política de chaves que dê acesso a apenas um usuário à chave do KMS. Se você excluir esse usuário, a chave se tornará não gerenciável e você deverá [contatar o suporte da AWS](#) para recuperar o acesso à chave do KMS.

A instrução da política de chave mostrada acima concede permissão à [entidade principal da conta](#) para controlar a chave. O principal da conta representa o Conta da AWS e seus administradores, incluindo o [usuário raiz da conta](#). O usuário raiz da conta é a única entidade principal que não pode ser excluída, a menos que você exclua a Conta da AWS. As práticas recomendadas do IAM desencorajam agir em nome do usuário raiz da conta, exceto em caso de emergência. No entanto, talvez seja necessário atuar como usuário raiz da conta se você excluir todos os outros usuários e funções com acesso à chave do KMS.

Permite que administradores de chaves administrem a chave do KMS

A política de chaves padrão criada pelo console permite que você escolha usuários e funções do IAM na conta e os torne administradores de chaves. Essa declaração é chamada de declaração de administradores de chaves. Os administradores de chaves têm permissões para gerenciar a chave do KMS, mas não têm permissões para usar essa chave em [operações de criptografia](#). Você pode

adicionar usuários e funções do IAM à lista de administradores de chaves ao criar a chave do KMS na visualização padrão ou na visualização de políticas.

 Warning

Como os administradores de chaves têm permissão para alterar a política de chaves e criar concessões, eles podem conceder a si mesmos e a outras pessoas AWS KMS permissões não especificadas nessa política.

As entidades principais que têm permissão para gerenciar etiquetas e aliases também podem controlar o acesso a uma chave do KMS. Para obter detalhes, consulte [ABAC para AWS KMS](#).

 Note

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

O exemplo a seguir mostra a declaração de administradores de chaves na visualização padrão do console do AWS KMS .

Key policy | Tags

Key policy Switch to policy view

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

Add Remove

< 1 >

☐	Name	Path	Type
☐	ExampleAdminRole	/	Role

Key deletion

☒ Allow key administrators to delete this key

A seguir é apresentado um exemplo que mostra a declaração de administradores de chaves na visualização padrão do console do AWS KMS . Essa instrução de administrador de chave destina-se a uma chave do KMS de criptografia simétrica de região única.

 Note

O AWS KMS console adiciona administradores de chaves à política de chaves sob o identificador "Allow access for Key Administrators" de instrução. A modificação desse identificador de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

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

A instrução padrão de administradores de chaves para a chave mais comum do KMS, uma chave do KMS de criptografia simétrica de região única, possibilita as permissões a seguir. Para obter informações detalhadas sobre cada permissão, consulte a [AWS KMS permissões](#).

Quando você usa o AWS KMS console para criar uma chave KMS, o console adiciona os usuários e as funções que você especifica ao `Principal` elemento na instrução dos administradores de chaves.

Muitas dessas permissões contêm o caractere curinga (*), que permite todas as permissões que começam com o verbo especificado. Como resultado, ao AWS KMS adicionar novas operações de API, os administradores de chaves podem usá-las automaticamente. Não é necessário atualizar suas políticas de chaves para incluir as novas operações. Se você preferir limitar seus administradores de chaves a um conjunto fixo de operações de API, poderá [alterar sua política de chaves](#).

kms:Create*

Permite [kms:CreateAlias](#) e [kms:CreateGrant](#). (A permissão `kms:CreateKey` é válida somente em uma política do IAM.)

kms:Describe*

Permite [kms:DescribeKey](#). A permissão `kms:DescribeKey` é necessária para visualizar a página de detalhes de chaves de uma chave do KMS no Console de gerenciamento da AWS.

kms:Enable*

Permite [kms:EnableKey](#). Para chaves do KMS de criptografia simétrica, ela também permite [kms:EnableKeyRotation](#).

kms:List*

Permite [kms:ListGrants](#), [kms:ListKeyPolicies](#) e [kms:ListResourceTags](#). (As permissões `kms:ListAliases` e `kms:ListKeys`, que são necessárias para exibir chaves do KMS no Console de gerenciamento da AWS, são válidas somente em políticas do IAM.)

kms:Put*

Permite [kms:PutKeyPolicy](#). Essa permissão permite que os administradores de chaves alterem a política de chaves dessa chave do KMS.

kms:Update*

Permite [kms:UpdateAlias](#) e [kms:UpdateKeyDescription](#). Para chaves de várias regiões, ele permite [kms:UpdatePrimaryRegion](#) nesta chave do KMS.

kms:Revoke*

Permite que o [kms:RevokeGrant](#), que permite que os administradores de chave [excluam uma concessão](#) mesmo que eles não sejam uma [entidade principal de retirada](#) na concessão.

kms:Disable*

Permite [kms:DisableKey](#). Para chaves do KMS de criptografia simétrica, ela também permite [kms:DisableKeyRotation](#).

kms:Get*

Permite [kms:GetKeyPolicy](#) e [kms:GetKeyRotationStatus](#). Para chaves do KMS com material de chave importado, ela permite [kms:GetParametersForImport](#). Para chaves do KMS assimétricas, ela permite [kms:GetPublicKey](#). A permissão `kms:GetKeyPolicy` é necessária para visualizar a política da chave de uma chave do KMS no Console de gerenciamento da AWS.

kms:Delete*

Permite [kms:DeleteAlias](#). Para chaves com material de chave importado, ela permite [kms:DeleteImportedKeyMaterial](#). A permissão `kms:Delete*` não permite que os administradores de chaves excluam a chave do KMS (`ScheduleKeyDeletion`).

kms:TagResource

Permite [kms:TagResource](#), que, por sua vez, permite que os administradores de chaves adicionem tags à chave do KMS. Como as tags também podem ser usadas para controlar o acesso à chave do KMS, essa permissão pode permitir que os administradores concedam ou neguem acesso à chave do KMS. Para obter detalhes, consulte [ABAC para AWS KMS](#).

kms:UntagResource

Permite [kms:UntagResource](#), que, por sua vez, permite que os administradores de chaves excluam tags da chave do KMS. Como as tags podem ser usadas para controlar o acesso à chave, essa permissão pode permitir que os administradores concedam ou neguem acesso à chave do KMS. Para obter detalhes, consulte [ABAC para AWS KMS](#).

kms:ScheduleKeyDeletion

Permite [kms:ScheduleKeyDeletion](#), que, por sua vez, permite que os administradores de chaves excluam [esta chave do KMS](#). Para excluir essa permissão, desmarque a opção Allow key administrators to delete this key(Permitir que os administradores de chaves excluam essa chave).

kms:CancelKeyDeletion

Permite [kms:CancelKeyDeletion](#), que, por sua vez, permite que os administradores de chaves [cancelem a exclusão desta chave do KMS](#). Para excluir essa permissão, desmarque a opção Allow key administrators to delete this key(Permitir que os administradores de chaves excluam essa chave).

kms:RotateKeyOnDemand

Permite [kms:RotateKeyOnDemand](#), que permite que os administradores de chaves [realizem a alternância sob demanda do material de chave nessa chave do KMS](#).

AWS KMS adiciona as seguintes permissões à declaração padrão dos administradores de chaves quando você cria chaves para fins especiais.

kms:ImportKeyMaterial

A permissão [kms:ImportKeyMaterial](#) permite que os administradores de chaves importem material de chave para a chave do KMS. Essa permissão está incluída na política de chaves somente quando você [cria uma chave do KMS sem material de chave](#).

kms:ReplicateKey

A [kms:ReplicateKey](#) permissão permite que os administradores de chaves [criem uma réplica de uma chave primária multirregional em uma região](#) diferente. AWS Essa permissão é incluída na política de chaves somente quando você cria uma chave primária ou de réplica de várias regiões.

kms:UpdatePrimaryRegion

A permissão [kms:UpdatePrimaryRegion](#) permite que os administradores de chaves [alterem uma chave de réplica de várias regiões para uma chave primária de várias regiões](#). Essa permissão é incluída na política de chaves somente quando você cria uma chave primária ou de réplica de várias regiões.

Permite que os usuários de chaves usem a chave do KMS

A política de chaves padrão que o console cria para chaves KMS permite que você escolha usuários do IAM e funções do IAM na conta, e externas Contas da AWS, e os torne usuários-chave.

O console adiciona duas instruções de política à política de chaves para usuários de chaves.

- [Use a chave do KMS diretamente](#) – A primeira instrução de política de chaves dá aos usuários de chave permissão para usar a chave do KMS diretamente para todas as [operações de criptografia](#) com suporte para esse tipo de chave do KMS.
- [Use a chave KMS com AWS serviços](#) — A segunda declaração de política dá permissão aos principais usuários para permitir que os AWS serviços integrados usem a chave KMS em seu nome AWS KMS para proteger recursos, como buckets do Amazon S3 e tabelas do Amazon DynamoDB.

Você pode adicionar usuários do IAM, funções do IAM e outros Contas da AWS à lista de usuários principais ao criar a chave do KMS. Você também pode editar a lista com a visualização padrão do console para políticas de chaves, conforme mostrado na imagem a seguir. A visualização padrão para políticas de chaves está na página de detalhes de chaves. Para obter mais informações sobre

como permitir que usuários de outras Contas da AWS possam usar a chave KMS, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

Note

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Key users

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

AddRemove

< 1 >

☐	Name	Path	Type
☐	ExampleRole	/	Role

Other AWS accounts

- arn:aws:iam::444455556666:root

As declarações de usuários de chaves padrão para uma chave simétrica de região única possibilita as permissões a seguir. Para obter informações detalhadas sobre cada permissão, consulte a [AWS KMS permissões](#).

Quando você usa o AWS KMS console para criar uma chave KMS, o console adiciona os usuários e as funções que você especifica ao Principal elemento em cada declaração de usuários da chave.

 Note

O AWS KMS console adiciona os principais usuários à política de chaves sob os identificadores de declaração "Allow use of the key" e "Allow attachment of persistent resources". A modificação desses identificadores de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

```
{
  "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}}
}
```


Permite que usuários de chaves usem uma chave do KMS para operações de criptografia

Os usuários de chaves têm permissão para usar a chave do KMS diretamente em todas as [operações de criptografia](#) com suporte na chave do KMS. Eles também podem usar a [DescribeKey](#) operação para obter informações detalhadas sobre a chave KMS no AWS KMS console ou usando as operações da AWS KMS API.

Por padrão, o AWS KMS console adiciona declarações de usuários-chave, como as dos exemplos a seguir, à política de chaves padrão. Como elas são compatíveis com diferentes operações de API, as ações nas instruções de política para chaves do KMS de criptografia simétrica, chaves do KMS de Hash-based message authentication code (HMAC – Código de autenticação de mensagem por hash), chaves do KMS assimétricas para criptografia de chave pública e chaves do KMS assimétricas para assinatura e verificação são ligeiramente diferentes.

Chaves do KMS de criptografia simétrica

O console adiciona a seguinte instrução à política de chaves para chaves do KMS de criptografia simétrica.

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

Chaves do KMS de HMAC

O console adiciona a seguinte instrução à política de chaves para chaves do KMS de HMAC.

```
{
  "Sid": "Allow use of the key",
```

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

Chaves do KMS assimétricas para criptografia de chave pública

O console adiciona a seguinte instrução à política de chaves para chaves do KMS assimétricas com um uso de chave de Encrypt and decrypt (Criptografar e descriptografar).

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

Chaves do KMS assimétricas para assinatura e verificação

O console adiciona a seguinte instrução à política de chaves para chaves do KMS assimétricas com um uso de chave de Sign and verify (Assinar e verificar).

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

Chaves assimétricas do KMS para derivar segredos compartilhados

O console adiciona a seguinte instrução à política de chave para chaves assimétricas do KMS com um uso de chave de Acordo de chave.

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

As ações nessas declarações concedem aos usuários de chaves as permissões a seguir.

[kms:Encrypt](#)

Permite que os usuários de chaves criptografem dados com essa chave do KMS.

[kms:Decrypt](#)

Permite que os usuários de chaves descriptografem dados com essa chave do KMS.

[kms:DeriveSharedSecret](#)

Permite que os usuários de chaves derivem segredos compartilhados com essa chave do KMS.

[kms:DescribeKey](#)

Permite que os usuários de chaves obtenham informações detalhadas sobre essa chave do KMS, incluindo seus identificadores, data de criação e estado de chave. Também permite que os principais usuários exibam detalhes sobre a chave KMS no AWS KMS console.

kms:GenerateDataKey*

Permite que os usuários de chaves solicitem uma chave de dados simétrica ou um par de chaves de dados assimétricas para operações criptográficas no lado do cliente. O console usa o caractere curinga * para representar a permissão para as seguintes operações de API: [GenerateDataKeyGenerateDataKeyWithoutPlaintext](#), [GenerateDataKeyPair](#), e. [GenerateDataKeyPairWithoutPlaintext](#) Essas permissões são válidas somente nas chaves do KMS simétricas que criptografam as chaves de dados.

[kms: GenerateMac](#)

Permite que os usuários de chaves usem uma chave do KMS de HMAC para gerar uma etiqueta de HMAC.

[kms: GetPublicKey](#)

Permite que os usuários de chaves baixem a chave pública da chave do KMS assimétrica. As partes com quem você compartilha essa chave pública podem criptografar dados fora do AWS KMS. No entanto, esses textos cifrados só podem ser descriptografados chamando a operação [Decrypt](#) no AWS KMS.

[kms: * ReEncrypt](#)

Permite que os usuários de chaves criptografem novamente os dados que foram originalmente criptografados com essa chave do KMS ou permite que eles usem essa chave do KMS para criptografar novamente dados já criptografados. A [ReEncrypt](#) operação requer acesso às chaves KMS de origem e destino. Para fazer isso, é possível conceder a permissão `kms:ReEncryptFrom` na chave do KMS de origem e a permissão `kms:ReEncryptTo` na chave do KMS de destino. No entanto, para simplificar, o console permite `kms:ReEncrypt*` (com o caractere curinga *) nas duas chaves do KMS.

[kms:Sign](#)

Permite que os usuários de chaves assinem mensagens com essa chave do KMS.

[kms:Verify](#)

Permite que os usuários de chaves verifiquem assinaturas com essa chave do KMS.

[kms: VerifyMac](#)

Permite que os usuários de chaves usem uma chave do KMS de HMAC para verificar uma etiqueta de HMAC.

Permite que os principais usuários usem a chave KMS com serviços AWS

A política de chaves padrão no console também dá aos principais usuários as permissões de que precisam para proteger seus dados em AWS serviços que usam concessões. AWS os serviços geralmente usam concessões para obter permissão específica e limitada para usar uma chave KMS.

Essa declaração de política de chaves permite que o usuário da chave crie, visualize e revogue concessões na chave KMS, mas somente quando a solicitação de operação de concessão vem de um [AWS serviço integrado](#) com o. AWS KMS A condição [kms: GrantIsFor AWSResource](#) policy não permite que o usuário chame essas operações de concessão diretamente. Quando o usuário da chave permite, um AWS serviço pode criar uma concessão em nome do usuário que permite que o serviço use a chave KMS para proteger os dados do usuário.

Os usuários de chaves precisam dessas permissões de concessão para usar a chave do KMS com os serviços integrados, mas essas permissões não são suficientes. Os usuários de chaves também precisam de permissão para usar os serviços integrados. Para obter detalhes sobre como dar aos usuários acesso a um AWS serviço que se integra ao AWS KMS, consulte a documentação do serviço integrado.

```
{
  "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}}
}
```

Por exemplo, os usuários de chaves podem usar essas permissões na chave do KMS das maneiras indicadas a seguir.

- Use essa chave do KMS com o Amazon Elastic Block Store (Amazon EBS) e com o Amazon Elastic Compute Cloud (Amazon EC2) para anexar um volume do EBS criptografado a uma instância do EC2. O usuário de chaves concede implicitamente permissão ao Amazon EC2 para usar a chave do KMS com o objetivo de associar o volume criptografado à instância. Para obter mais informações, consulte [Como o Amazon Elastic Block Store \(Amazon EBS\) usa AWS KMS](#).

- Use essa chave do KMS com o Amazon Redshift para executar um cluster criptografado. O usuário de chaves concede implicitamente permissão ao Amazon Redshift para usar a chave do KMS para executar o cluster criptografado e criar snapshots criptografados. Para obter mais informações, consulte [Como o Amazon Redshift usa AWS KMS](#).
- Use essa chave do KMS com outros [serviços da AWS integrados ao AWS KMS](#) que usem concessões para criar, gerenciar ou usar recursos criptografados com esses serviços.

Apolítica de chaves padrão permite que os usuários de chaves deleguem suas permissões de concessão para todos os serviços integrados que usam concessões. No entanto, você pode criar uma política de chaves personalizada que restrinja a permissão a AWS serviços específicos. Para obter mais informações, consulte a [kms:ViaService](#) chave de condição.

Visualizar uma política de chave

Você pode ver a política de chaves de uma [chave gerenciada pelo AWS KMS cliente](#) ou [Chave gerenciada pela AWS](#) em sua conta usando o AWS KMS console ou a [GetKeyPolicy](#) operação na AWS KMS API. Não é possível usar essas técnicas para visualizar a política de chaves de uma chave do KMS em outra Conta da AWS.

Para saber mais sobre as AWS KMS principais políticas, consulte [Políticas-chave em AWS KMS](#). Para saber como determinar quais usuários e funções têm acesso a uma chave do KMS, consulte [the section called “Determinar o acesso”](#).

Usando o AWS KMS console

Os usuários autorizados podem visualizar a política de chaves de uma [Chave gerenciada pela AWS](#) ou de uma [chave gerenciada pelo cliente](#) na guia Key policy (Política de chaves) do Console de gerenciamento da AWS.

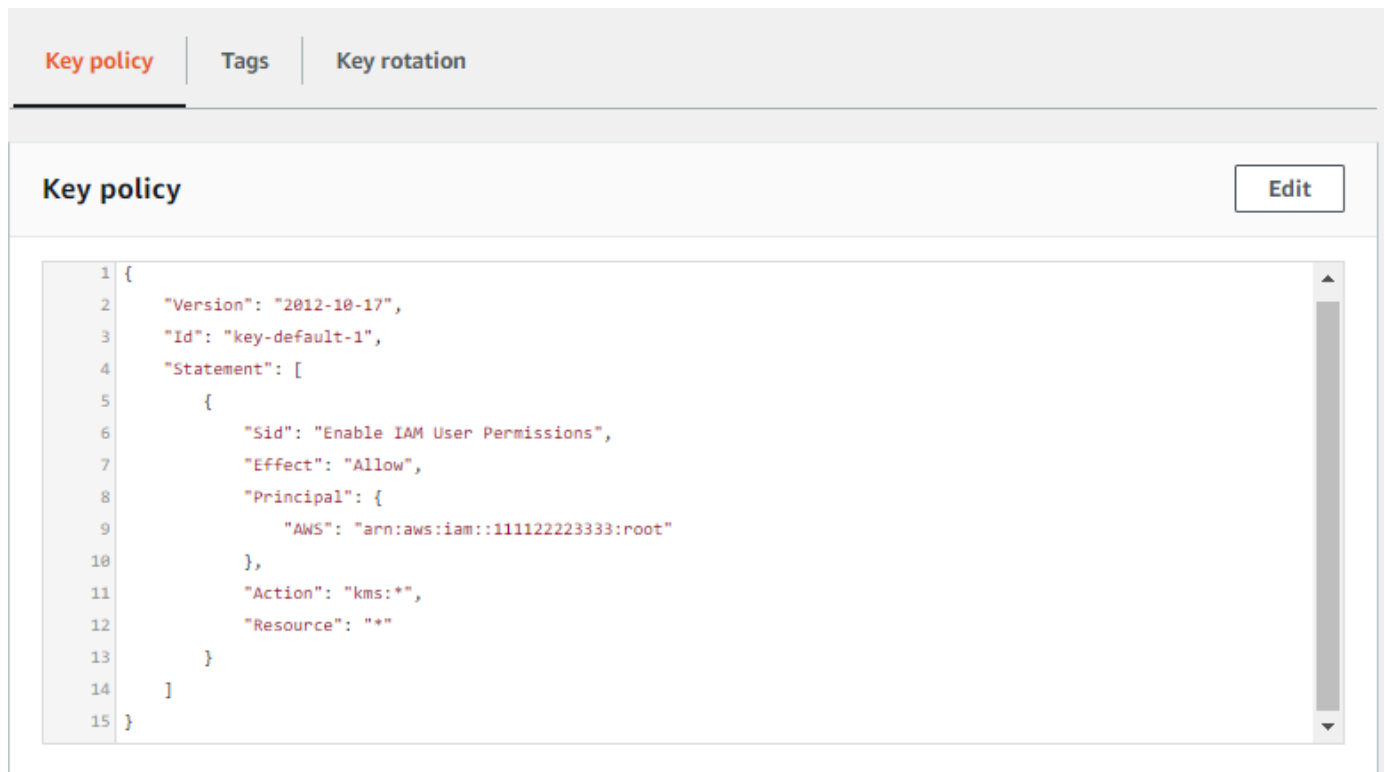
Para visualizar a política de chaves de uma chave KMS no Console de gerenciamento da AWS, você deve ter as permissões [kms: ListAliases](#), [kms: DescribeKey](#) e [kms: GetKeyPolicy](#).

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. Para ver as chaves em sua conta que AWS cria e gerencia para você, no painel de navegação, escolha chaves AWS gerenciadas. Para exibir as chaves em sua conta que você cria e gerencia, no painel de navegação, escolha Customer managed keys (Chaves gerenciadas de cliente).

4. Na lista de chaves do KMS, escolha o alias ou o ID de chave da chaves do KMS que você deseja examinar.
5. Selecione a guia Key policy (Política de chaves).

Na seção Key policy (Política de chaves), você pode ver o documento da política de chaves. Essa é a visualização da política. Nas instruções da política de chaves, é possível ver as entidades principais que receberam acesso à chave do KMS pela política de chaves e ver as ações que elas podem executar.

O exemplo a seguir mostra a visualização da [política de chaves padrão](#).



Ou, se você criou a chave KMS no Console de gerenciamento da AWS, você verá a visualização padrão com seções para administradores de chaves, exclusão de chaves e usuários de chaves. Para ver o documento da política de chaves, selecione Switch to policy view (Alternar para a visualização da políticas).

O exemplo a seguir mostra a visualização padrão da [política de chaves padrão](#).

Key policy

Tags

Key rotation

Key policy

Switch to policy view

Key administrators

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

Add

Remove

Q

< 1 >

	Name	Path	Type
Empty Resources			
No resources to display			

Key deletion

☐ Allow key administrators to delete this key

Key users

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

Add

Remove

Q

< 1 >

	Name	Path	Type
Empty Resources			
No resources to display			

Usando a AWS KMS API

Para obter a política de chaves de uma chave KMS em sua Conta da AWS, use a [GetKeyPolicy](#) operação na AWS KMS API. Não é possível usar essa operação para visualizar uma política de chaves em uma conta diferente.

O exemplo a seguir usa o [get-key-policy](#) comando no AWS Command Line Interface (AWS CLI), mas você pode usar qualquer AWS SDK para fazer essa solicitação.

Visualizar uma política de chave

103

Observe que o parâmetro `PolicyName` será exigido mesmo se `default` for seu único valor válido. Além disso, esse comando solicita a saída em texto, em vez de em JSON, para facilitar a visualização.

Antes de executar esse comando, substitua o ID de chave de exemplo por um válido da sua conta.

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

A resposta deve ser semelhante à seguinte, que retorna a [política de chaves padrão](#).

JSON

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

Alterar uma política de chave

Você pode alterar a política de chaves de uma chave KMS na sua Conta da AWS usando a Console de gerenciamento da AWS ou a [PutKeyPolicy](#) operação. Não é possível usar essas técnicas para alterar a política de chaves de uma chave do KMS em outra Conta da AWS.

Ao alterar a política de chaves, lembre-se das seguintes regras:

- É possível visualizar a política de chaves de uma [Chave gerenciada pela AWS](#) ou de uma [chave gerenciada pelo cliente](#), mas só é possível alterar a política de chaves de uma chave gerenciada pelo cliente. As políticas de Chaves gerenciadas pela AWS são criadas e gerenciadas pelo AWS

serviço que criou a chave KMS em sua conta. Não é possível visualizar ou alterar a política de chaves para uma [Chave pertencente à AWS](#).

- Você pode adicionar ou remover usuários do IAM, funções do IAM e Contas da AWS na política de chaves e alterar as ações que são permitidas ou negadas para esses diretores. Para mais informações sobre as maneiras de especificar principais e permissões em uma política de chaves, consulte [Políticas de chaves](#).
- Não é possível adicionar grupos do IAM a uma política de chaves, mas é possível adicionar diversos usuários do IAM e perfis do IAM. Para obter mais informações, consulte [Permitir que diversas entidades principais do IAM acessem uma chave do KMS](#).
- Se você adicionar externo Contas da AWS a uma política de chaves, também deverá usar políticas do IAM nas contas externas para dar permissões aos usuários, grupos ou funções do IAM nessas contas. Para obter mais informações, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).
- O documento de política de chaves resultante não pode exceder 32 KB (32.768 bytes).

Como alterar uma política de chaves

Você pode alterar uma política de chaves de três maneiras diferentes, conforme explicado nas seções a seguir.

Tópicos

- [Usando a visualização Console de gerenciamento da AWS padrão](#)
- [Usando a visualização Console de gerenciamento da AWS da política](#)
- [Usando a AWS KMS API](#)

Usando a visualização Console de gerenciamento da AWS padrão

Você pode usar o console para alterar uma política de chaves com uma interface gráfica chamada visualização padrão.

Se as seguintes etapas não correspondem à visualização no console, isso significa que essa política de chaves não foi criada pelo console. Ou significa que a política de chaves foi modificada e é incompatível com a visualização padrão do console. Nesse caso, siga as etapas em [Usando a visualização Console de gerenciamento da AWS da política](#) ou [Usando a AWS KMS API](#).

1. Visualize a política de chaves de uma chave gerenciada pelo cliente conforme descrito em [Usando o AWS KMS console](#). (Você não pode alterar as principais políticas do Chaves gerenciadas pela AWS.)
2. Defina o que alterar.
 - Para adicionar ou remover [administradores de chaves](#) e permitir ou impedir que eles [excluam a chave do KMS](#), use os controles na seção Key administrators (Administradores de chaves) da página. Os administradores de chaves gerenciam a chave do KMS, incluindo sua habilitação e desabilitação, a configuração da política de chaves e a [habilitação da alternância de chaves](#).
 - Para adicionar ou remover [usuários principais](#) e permitir ou Contas da AWS proibir o uso externo da chave KMS, use os controles na seção Usuários principais da página. Usuários de chaves podem usar a chave do KMS em [operações de criptografia](#), como criptografar, descriptografar, recriptografar e gerar chaves de dados.

Usando a visualização Console de gerenciamento da AWS da política

Você pode usar o console para alterar um documento de política de chaves com a visualização de política do console.

1. Visualize a política de chaves de uma chave gerenciada pelo cliente conforme descrito em [Usando o AWS KMS console](#). (Você não pode alterar as principais políticas do Chaves gerenciadas pela AWS.)
2. Na seção Política de chave, selecione Alternar para visualização de política.
3. Escolha Editar.
4. Defina o que alterar.
 - Para adicionar uma nova instrução, escolha Adicionar nova instrução. Então, você pode selecionar as ações, as entidades principais e as condições para a nova instrução de política de chave entre as opções listadas no painel do construtor de instruções ou inserir manualmente os elementos de instrução da política.
 - Para remover uma instrução da política de chave, selecione a instrução e escolha Remover. Leia a instrução de política selecionada e confirme se deseja removê-la. Se você decidir que não deseja continuar com a remoção da instrução, escolha Cancelar.

- Para editar uma instrução de política de chave existente, selecione a instrução. Em seguida, use o painel do construtor de instruções para escolher os elementos específicos que deseja modificar, ou edite manualmente a instrução.

5. Escolha Salvar alterações.

Usando a AWS KMS API

Você pode usar a [PutKeyPolicy](#) operação para alterar a política de chaves de uma chave KMS no seu Conta da AWS. Não é possível usar essa API em uma chave do KMS em outra Conta da AWS.

1. Use a [GetKeyPolicy](#) operação para obter o documento de política de chaves existente e, em seguida, salve o documento de política de chaves em um arquivo. Para obter o código de exemplo em várias linguagens de programação, consulte [Use GetKeyPolicy com um AWS SDK ou CLI](#).
2. Abra o documento de política de chaves no editor de texto de sua preferência, edite-o e salve o arquivo.
3. Use a [PutKeyPolicy](#) operação para aplicar o documento de política de chaves atualizado à chave KMS. Para obter o código de exemplo em várias linguagens de programação, consulte [Use PutKeyPolicy com um AWS SDK ou CLI](#).

Para ver um exemplo de cópia de uma política de chaves de uma chave KMS para outra, consulte o [GetKeyPolicy exemplo](#) na Referência de AWS CLI Comandos.

Permissões para AWS serviços nas principais políticas

Muitos AWS serviços são usados AWS KMS keys para proteger os recursos que gerenciam. Quando um serviço usa [Chaves pertencentes à AWS](#) ou [Chaves gerenciadas pela AWS](#), o serviço estabelece e mantém as políticas de chaves para essas chaves do KMS.

No entanto, ao usar uma [chave gerenciada pelo cliente](#) com um serviço da AWS, você é quem define e mantém a política de chaves. Essa política de chaves deve conceder ao serviço as permissões mínimas necessárias para proteger o recurso em seu nome. Recomendamos seguir o princípio de privilégio mínimo: conceda ao serviço apenas as permissões necessárias. Você pode fazer isso de forma eficaz aprendendo quais permissões o serviço precisa e usando [chaves de condição globais da AWS](#) e [chaves de condição do AWS KMS](#) para refinar as permissões.

Para encontrar as permissões que o serviço exige em uma chave gerenciada pelo cliente, consulte a documentação de criptografia do serviço. A lista a seguir inclui links para a documentação de alguns serviços:

- AWS CloudTrail permissões - [Configure as AWS KMS principais políticas para CloudTrail](#)
- Permissões do Amazon Elastic Block Store: [Guia do usuário do Amazon EC2](#)
- Permissões do AWS Lambda: [criptografia de dados em repouso para o Lambda](#)
- Permissões do Amazon Q: [criptografia de dados para o Amazon Q](#)
- Permissões do Amazon Relational Database Service: [gerenciamento de chave do AWS KMS](#)
- Permissões da AWS Secrets Manager: [autorização de uso da chave do KMS](#)
- Permissões do Amazon Simple Queue Service: [gerenciamento de chave do Amazon SQS](#)

Usando políticas do IAM com AWS KMS

Você pode usar as políticas do IAM, junto com [as principais políticas](#), [concessões](#) e [políticas de VPC endpoint](#), para controlar o acesso à sua entrada. AWS KMS keys AWS KMS

Note

Para usar uma política do IAM para controlar o acesso a uma chave do KMS, a política de chaves da chave do KMS deve conceder à conta permissão para usar políticas do IAM. Especificamente, a política de chaves deve incluir a [instrução de política que habilita políticas do IAM](#).

Esta seção explica como usar as políticas do IAM para controlar o acesso às AWS KMS operações. Para obter informações mais gerais sobre permissões do IAM, consulte o [Manual do usuário do IAM](#).

Todas chaves do KMS do KMS têm uma política de chaves. Políticas do IAM são opcionais. Para usar uma política do IAM para controlar o acesso a uma chave do KMS, a política de chaves da chave do KMS deve conceder à conta permissão para usar políticas do IAM. Especificamente, a política de chaves deve incluir a [instrução de política que habilita políticas do IAM](#).

As políticas do IAM podem controlar o acesso a qualquer AWS KMS operação. Diferentemente das políticas de chaves, as políticas do IAM podem controlar o acesso a várias chaves do KMS e

fornecer permissões para as operações de vários AWS serviços relacionados. Mas as políticas do IAM são particularmente úteis para controlar o acesso a operações [CreateKey](#), como aquelas que não podem ser controladas por uma política de chaves porque não envolvem nenhuma chave KMS específica.

Se você acessar AWS KMS por meio de um endpoint da Amazon Virtual Private Cloud (Amazon VPC), você também pode usar uma política de endpoint de VPC para limitar o acesso aos seus AWS KMS recursos ao usar o endpoint. Por exemplo, ao usar o VPC endpoint, você pode permitir que apenas os principais acessem suas Conta da AWS chaves gerenciadas pelo cliente. Para obter detalhes, consulte [Editar a política de endpoints da VPC](#).

Para ajuda sobre como escrever e formatar um documento de política JSON, consulte [a Referência a políticas JSON do IAM](#), no Manual do usuário do IAM.

Use as políticas do IAM das seguintes maneiras:

- Anexar uma política de permissões a uma função de federação ou a permissões entre contas – Você pode anexar uma política do IAM a uma função do IAM para permitir a federação de identidades, conceder permissões entre contas ou a aplicações em execução nas instâncias do EC2. Para mais informações sobre os vários casos de uso das funções do IAM, consulte [Funções do IAM](#) no Manual do usuário do IAM.
- Vincular uma política de permissões a um usuário ou a um grupo: é possível vincular uma política que permite que um usuário ou um grupo de usuários chame operações do AWS KMS. No entanto, as práticas recomendadas do IAM recomendam usar identidades com credenciais temporárias, como perfis do IAM, sempre que possível.

O exemplo a seguir mostra uma política do IAM com AWS KMS permissões. Essa política permite que as identidades do IAM às quais está associada obtenham todas as chaves do KMS e aliases.

JSON

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

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

Como todas as políticas do IAM, essa política não tem um elemento `Principal`. Quando você vincula uma política do IAM a uma identidade do IAM, essa identidade obtém as permissões especificadas na política.

Para ver uma tabela mostrando todas as ações da AWS KMS API e os recursos aos quais elas se aplicam, consulte [Referência de permissões](#) o.

Permitir que diversas entidades principais do IAM acessem uma chave do KMS

Grupos do IAM não são entidades principais válidas em uma política de chaves. Para permitir que diversos usuários e perfis acessem uma chave do KMS, siga um dos procedimentos a seguir:

- Use um perfil do IAM como entidade principal na política de chaves. Diversos usuários autorizados podem assumir o perfil, conforme necessário. Para obter detalhes, consulte [Perfis do IAM](#) no Guia do usuário do IAM.

Embora você possa listar diversos usuários do IAM em uma política de chaves, essa prática não é recomendada porque requer que você atualize a política de chaves sempre que a lista de usuários autorizados for alterada. Além disso, as práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

- Use uma política do IAM para conceder permissões a um grupo do IAM. Para fazer isso, certifique-se de que a política de chaves inclua a instrução que [possibilita que as políticas do IAM concedam acesso à chave do KMS](#), [crie uma política do IAM](#) que conceda acesso à chave do KMS e, em seguida, [vincule essa política a um grupo do IAM](#) que contenha os usuários do IAM autorizados. Com essa abordagem, você não precisará alterar nenhuma política quando a lista de usuários autorizados for alterada. Basta adicionar ou remover esses usuários do grupo do IAM apropriado. Para obter detalhes, consulte [Grupos de usuários do IAM](#) no Guia do usuário do IAM.

Para obter mais informações sobre como as AWS KMS principais políticas e as políticas do IAM funcionam juntas, consulte [Solução de problemas de AWS KMS permissões](#).

Práticas recomendadas para políticas do IAM

Proteger o acesso ao AWS KMS keys é fundamental para a segurança de todos os seus AWS recursos. Chaves do KMS são usadas para proteger muitos dos recursos mais confidenciais da sua Conta da AWS. Reserve algum tempo para elaborar as [políticas de chave](#), políticas do IAM, [concessões](#) e políticas de endpoint da VPC que controlam o acesso às suas chaves do KMS.

Em instruções de políticas do IAM que controlam o acesso a chaves do KMS, use o [princípio de menor privilégio](#). Conceda às entidades principais do IAM somente as permissões de que eles precisam somente nas chaves do KMS que elas devem usar ou gerenciar.

As práticas recomendadas a seguir se aplicam às políticas do IAM que controlam o acesso a AWS KMS chaves e aliases. Para obter uma orientação geral sobre as práticas recomendadas da política do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Usar políticas de chaves

Sempre que possível, conceda permissões em políticas de chaves que afetem uma chave do KMS, em vez de em uma política do IAM que possa ser aplicada a muitas chaves do KMS, incluindo aquelas em outras Contas da AWS. Isso é particularmente importante para permissões confidenciais, como [kms: PutKeyPolicy](#) e [kms:](#), ScheduleKeyDeletion mas também para operações criptográficas que determinam como seus dados são protegidos.

Limitar CreateKey a permissão

Dê permissão para criar chaves ([kms: CreateKey](#)) somente aos diretores que precisarem delas. Entidades principais que criam uma chave do KMS também definem sua política de chave, para que possam conceder a si mesmas e a outros permissão para usar e gerenciar as chaves do KMS criadas. Ao conceder essa permissão, considere limitá-la usando [condições de política](#). Por exemplo, você pode usar a KeySpec condição [kms:](#) para limitar a permissão às chaves KMS de criptografia simétrica.

Especificar chaves do KMS em uma política do IAM

Como prática recomendada, especifique o [ARN da chave](#) de cada chave do KMS à qual a permissão se aplica no elemento Resource da instrução de política. Esta prática restringe a permissão a chaves do KMS necessárias para a entidade principal. Por exemplo, esse elemento Resource lista apenas as chaves do KMS que a entidade principal precisa usar.

```
"Resource": [
```



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

Quando especificar chaves KMS for impraticável, use um Resource valor que limite o acesso às chaves KMS em uma região confiável, Conta da AWS como.

`arn:aws:kms:region:account:key/*` Ou limite o acesso às chaves KMS em todas as regiões (*) de uma rede confiável Conta da AWS, como `arn:aws:kms:*:account:key/*`.

Não é possível usar um [ID de chave](#), um [alias do nome](#) ou [alias do ARN](#) para representar uma chave do KMS no campo Resource de uma política do IAM. Se você especificar um ARN de alias, a política se aplicará ao alias, e não à chave do KMS. Para obter informações gerais sobre políticas do IAM para aliases, consulte [Controlar o acesso a aliases](#)

Evite "Resource": "*" em uma política do política do IAM

Use caracteres curinga (*) com critério. Em uma política de chaves, o caractere curinga no elemento Resource representa a chave do KMS à qual a política de chaves está associada. Mas em uma política do IAM, um caractere curinga sozinho no Resource elemento ("Resource": "*") aplica as permissões a todas as chaves do KMS em tudo o Contas da AWS que a conta do diretor tem permissão para usar. Isso pode incluir [chaves KMS em outras Contas da AWS](#), bem como chaves KMS na conta do diretor.

Por exemplo, para usar uma chave KMS em outra Conta da AWS, um principal precisa da permissão da política de chaves da chave KMS na conta externa e de uma política do IAM em sua própria conta. Suponha que uma conta arbitrária concedeu permissão à sua Conta da AWS [kms:Decrypt](#) em suas . Em caso afirmativo, uma política do IAM na sua conta que dê acesso uma função com permissão `kms:Decrypt` em todas as chaves do KMS ("Resource": "*") atenderá à parte do IAM do requisito. Como resultado, as entidades principais que podem assumir essa função agora podem descriptografar textos cifrados usando a chave do KMS na conta não confiável. As entradas de suas operações aparecem nos CloudTrail registros de ambas as contas.

Especificamente, evite usar "Resource": "*" em uma declaração de política que permita as operações de API a seguir. Essas operações podem ser chamadas em chaves KMS em outras Contas da AWS.

- [DescribeKey](#)
- [GetKeyRotationStatus](#)

- [Operações criptográficas \(criptografar, descriptografar,,,,, GenerateDataKey,, GenerateDataKeyPair, GenerateDataKeyWithoutPlaintextassinar GenerateDataKeyPairWithoutPlaintextGetPublicKey, ReEncryptverificar\)](#)
- [CreateGrant](#), [ListGrants](#), [ListRetirableGrants](#), [RetireGrant](#), [RevokeGrant](#)

Quando usar "Resource": "*"

Em uma política do IAM, use um caractere curinga no elemento Resource somente para permissões que o exijam. Somente as permissões a seguir exigem o elemento "Resource":
" * ".

- [kms: CreateKey](#)
- [kms: GenerateRandom](#)
- [kms: ListAliases](#)
- [kms: ListKeys](#)
- Permissões para armazenamentos de chaves personalizadas, como [kms: CreateCustomKeyStore](#) e [kms: ConnectCustomKeyStore](#)

Note

As permissões para operações de alias ([kms: CreateAlias](#), [kms: UpdateAlias](#), [kms: DeleteAlias](#)) devem ser anexadas ao alias e à chave KMS. É possível usar "Resource": " * " em uma política do IAM para representar os aliases e as chaves do KMS, ou especificar os aliases e as chaves do KMS no elemento Resource. Para obter exemplos, consulte [Controlar o acesso a aliases](#).

Os exemplos deste tópico fornecem mais informações e orientações para criar políticas do IAM para chaves do KMS. Para ver as melhores práticas do IAM para todos os AWS recursos, consulte [as melhores práticas de segurança no IAM](#) no Guia do usuário do IAM.

Especificando chaves do KMS em instruções de políticas do IAM

É possível usar uma política do IAM para permitir que uma entidade principal use ou gerencie chaves do KMS. Chaves do KMS são especificadas no elemento Resource da instrução de política.

- Para especificar uma chave do KMS em uma instrução de política do IAM, use seu [ARN de chave](#). Não é possível usar um [ID de chave](#), um [nome de alias](#) nem um [ARN de alias](#) para identificar uma chave do KMS em uma instrução de política do IAM.

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

Para controlar o acesso a uma chave KMS com base em seus aliases, use as chaves de condição [kms: RequestAlias](#) ou [kms: ResourceAliases](#). Para obter detalhes, consulte [ABAC para AWS KMS](#).

Use um alias ARN como recurso somente em uma declaração de política que controle o acesso às operações de alias, [CreateAlias](#) como [UpdateAlias](#), ou. [DeleteAlias](#). Para obter detalhes, consulte [Controlar o acesso a aliases](#).

- Para especificar várias chaves do KMS na conta e região, use caracteres curinga (*) nas posições de Região ou ID de recurso do ARN de chave.

Por exemplo, para especificar todas as chaves do KMS na região Oeste dos EUA (Oregon) de uma conta, use "Resource": "arn:aws:kms:us-west-2:111122223333:key/*". Para especificar todas as chaves do KMS em todas as regiões da conta, use "Resource": "arn:aws:kms*:111122223333:key/*".

- Para representar todas as chaves do KMS, use um caractere curinga sozinho ("*"). Use esse formato para operações que não usam nenhuma chave KMS específica, a saber [CreateKeyGenerateRandom](#), [ListAliases](#), e. [ListKeys](#)

Ao escrever suas instruções de política, é uma [prática recomendada](#) limitar as chaves do KMS àquelas que as entidades principais precisam usar, em vez de conceder acesso a todas as chaves do KMS.

Por exemplo, a declaração de política do IAM a seguir permite que o diretor chame as operações [DescribeKeyGenerateDataKey](#), [Decrypt](#) somente nas chaves KMS listadas no Resource elemento da declaração de política. Especificar chaves do KMS por ARN de chave, que é uma prática recomendada, garante que as permissões sejam limitadas apenas às chaves do KMS especificadas.

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"
    ]
}

```

Para aplicar a permissão a todas as chaves KMS em um determinado local confiável Conta da AWS, você pode usar caracteres curinga (*) na região e nas posições de ID da chave. Por exemplo, a instrução de política a seguir permite que a entidade principal chame as operações especificadas em quaisquer chaves do KMS em duas contas confiáveis demonstrativas.

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/*"
    ]
  }
}

```

Também é possível usar um caractere curinga ("*") sozinho no elemento `Resource`. Como ela permite o acesso a todas as chaves do KMS que a conta tem permissão para usar, é recomendável principalmente para operações que não envolvam uma chave do KMS específica e para instruções `Deny`. Você também pode usá-lo em declarações de política que permitam apenas operações somente leitura menos confidenciais. Para determinar se uma AWS KMS operação envolve uma chave KMS específica, procure o valor da chave KMS na coluna Recursos da tabela em [the section called “Referência de permissões”](#)

Por exemplo, a instrução de política a seguir usa um efeito `Deny` para proibir as entidades principais de usar as operações especificadas em qualquer chave do KMS. Ela usa um caractere curinga no elemento `Resource` para representar todas as chaves do KMS.

JSON

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

A instrução de política a seguir usa um caractere curinga isoladamente para representar todas as chaves do KMS. Porém, ela permite apenas operações menos confidenciais somente leitura e operações que não se aplicam a uma chave do KMS específica.

JSON

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

```
    "kms:CreateKey",
    "kms:ListKeys",
    "kms:ListAliases",
    "kms:ListResourceTags"
  ],
  "Resource": "*"
}
```

Exemplos de política do IAM

Nesta seção, você pode encontrar exemplos de políticas do IAM que concedem permissões para várias ações do AWS KMS .

Important

Algumas permissões das políticas a seguir são concedidas somente quando a política de chaves da chave do KMS também as concede. Para obter mais informações, consulte [Referência de permissões](#).

Para ajuda sobre como escrever e formatar um documento de política JSON, consulte [a Referência a políticas JSON do IAM](#), no Manual do usuário do IAM.

Exemplos

- [Permitir que um usuário visualize as chaves KMS no console AWS KMS](#)
- [Permitir que um usuário crie chaves do KMS](#)
- [Permitir que um usuário criptografe e descriptografe com qualquer chave KMS em uma determinada Conta da AWS](#)
- [Permitir que um usuário criptografe e descriptografe com qualquer chave KMS em uma região específica Conta da AWS](#)
- [Permitir que um usuário criptografe e descriptografe com chaves do KMS específicas](#)
- [Impedir que um usuário desabilite ou exclua qualquer chave do KMS](#)

Permitir que um usuário visualize as chaves KMS no console AWS KMS

A política do IAM a seguir permite que os usuários tenham acesso somente de leitura ao AWS KMS console. Os usuários com essas permissões podem visualizar todas as chaves KMS em suas Conta da AWS, mas não podem criar ou alterar nenhuma chave KMS.

[Para visualizar as chaves KMS nas páginas de chaves gerenciadas pelo cliente Chaves gerenciadas pela AWSe pelas páginas de chaves gerenciadas pelo clienteListKeys, os diretores exigem as GetResources permissões kms: ListAliases, kms: e tag:, mesmo que as chaves não tenham tags ou aliases.](#) As permissões restantes, especialmente [kms: DescribeKey](#), são necessárias para visualizar colunas e dados opcionais da tabela de chaves KMS nas páginas de detalhes da chave KMS. As ListRoles permissões [iam: ListUsers](#) e [iam:](#) são necessárias para exibir a política de chaves na visualização padrão sem erros. Para visualizar dados na página Armazenamentos de chaves personalizadas e detalhes sobre chaves KMS em armazenamentos de chaves personalizadas, os diretores também precisam da permissão [kms: DescribeCustomKeyStores](#).

Se você limitar o acesso do console de um usuário a chaves do KMS específicas, o console exibirá um erro para cada chave do KMS que não estiver visível.

Essa política inclui duas declarações de política. O elemento Resource na primeira instrução de política habilita as permissões especificadas em todas as chaves do KMS em todas as regiões da Conta da AWS de exemplo. Os visualizadores do console não precisam de acesso adicional porque o console do AWS KMS exibe somente chaves do KMS na conta da entidade principal. Isso é verdade mesmo que eles tenham permissão para visualizar as chaves KMS em outras Contas da AWS. As permissões restantes AWS KMS e do IAM exigem um "Resource": "*" elemento porque não se aplicam a nenhuma chave KMS específica.

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": "*"
}
]
}

```

Permitir que um usuário crie chaves do KMS

A seguinte política do IAM permite que um usuário crie todos os tipos de chaves do KMS. O valor do Resource elemento é * porque a CreateKey operação não usa nenhum AWS KMS recurso específico (chaves ou aliases do KMS).

Para restringir o usuário a tipos específicos de chaves KMS, use as chaves de [condição kms:KeySpec](#), [kms: KeyUsage](#) e [kms: KeyOrigin](#).

JSON

```

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

```


As entidades principais que criam chaves podem precisar de algumas permissões relacionadas.

- **kms: PutKeyPolicy** — Os diretores que têm `kms:CreateKey` permissão podem definir a política inicial de chaves para a chave KMS. No entanto, o `CreateKey` chamador deve ter a `PutKeyPolicy` permissão [kms:](#), que permite alterar a política de chaves do KMS, ou deve especificar o `BypassPolicyLockoutSafetyCheck` parâmetro de `CreateKey`, o que não é recomendado. O autor da chamada `CreateKey` pode obter a permissão `kms:PutKeyPolicy` para a chave do KMS a partir de uma política do IAM ou incluir essa permissão na política de chaves da chave do KMS que ele está criando.
- **kms: TagResource** — Para adicionar tags à chave KMS durante a `CreateKey` operação, o `CreateKey` chamador deve ter a `TagResource` permissão [kms:](#) em uma política do IAM. Não é suficiente incluir essa permissão na política de chaves da nova chave do KMS. No entanto, se o autor da chamada `CreateKey` incluir `kms:TagResource` na política de chaves inicial, ele poderá adicionar etiquetas a uma chamada separada após a criação da chave do KMS.
- **kms: CreateAlias** — Os diretores que criam uma chave KMS no AWS KMS console devem ter a `CreateAlias` permissão [kms:](#) na chave KMS e no alias. (O console faz duas chamadas; uma para `CreateKey` e outra para `CreateAlias`). É necessário fornecer a permissão de alias em uma política do IAM. É possível fornecer a permissão da chave do KMS em uma política de chave ou política do IAM. Para obter detalhes, consulte [Controlar o acesso a aliases](#).

Além disso `kms:CreateKey`, a política do IAM a seguir fornece `kms:TagResource` permissão para todas as chaves KMS na Conta da AWS e `kms:CreateAlias` permissão para todos os aliases da conta. Ela também inclui algumas permissões úteis somente leitura que podem ser fornecidas somente em uma política do IAM.

Essa política do IAM não inclui a permissão `kms:PutKeyPolicy` nem outras permissões que possam ser definidas em uma política de chaves. Uma [prática recomendada](#) é definir essas permissões na política de chaves em que elas se aplicam exclusivamente a uma chave do 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": "*"
  }
]
```

Permitir que um usuário criptografe e descriptografe com qualquer chave KMS em uma determinada Conta da AWS

A política do IAM a seguir permite que um usuário criptografe e descriptografe dados com qualquer chave KMS em 111122223333. Conta da AWS

JSON

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

```
}
```

Permitir que um usuário criptografe e descriptografe com qualquer chave KMS em uma região específica Conta da AWS

A política do IAM a seguir permite que um usuário criptografe e descriptografe dados com qualquer chave KMS Conta da AWS 111122223333 na região Oeste dos EUA (Oregon).

JSON

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

Permitir que um usuário criptografe e descriptografe com chaves do KMS específicas

A política do IAM a seguir permite que um usuário criptografe e descriptografe dados com as duas chaves do KMS especificadas no elemento Resource. Ao especificar uma chave do KMS em uma instrução de política do IAM, você deve usar o [ARN de chave](#) dessa chave do 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'"
  ]
}

```

Impedir que um usuário desabilite ou exclua qualquer chave do KMS

A política do IAM a seguir impede que um usuário desabilite ou exclua chaves do KMS, mesmo quando outra política do IAM ou uma política de chaves concede essas permissões. Uma política que nega explicitamente permissões substitui todas as outras políticas, mesmo aquelas que concedem explicitamente as mesmas permissões. Para obter mais informações, consulte [Solução de problemas de AWS KMS permissões](#).

JSON

```

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

```

Políticas de controle de recursos em AWS KMS

As políticas de controle de recursos (RCPs) são um tipo de política organizacional que você pode usar para impor controles preventivos sobre AWS os recursos em sua organização. RCPs ajudam você a restringir centralmente o acesso externo aos seus AWS recursos em grande escala.

RCPs complementam as políticas de controle de serviços (SCPs). Embora SCPs possa ser usado para definir centralmente as permissões máximas sobre as funções e usuários do IAM em sua organização, RCPs pode ser usado para definir centralmente o máximo de permissões em AWS recursos em sua organização.

Você pode usar RCPs para gerenciar as permissões das chaves KMS gerenciadas pelo cliente em sua organização. RCPs sozinhos, não são suficientes para conceder permissões às chaves gerenciadas pelo cliente. Nenhuma permissão é concedida por uma RCP. Uma RCP define uma barreira de proteção de permissões, ou define limites, nas ações que as identidades podem realizar em recursos nas contas afetadas. O administrador ainda deve anexar políticas baseadas em identidade a usuários ou perfis do IAM, ou políticas de chave para conceder permissões de fato.

 Note

As políticas de controle de recursos de sua organização não se aplicam a [Chaves gerenciadas pela AWS](#).

Chaves gerenciadas pela AWS são criados, gerenciados e usados em seu nome por um AWS serviço, você não pode alterar ou gerenciar suas permissões.

Saiba mais

- Para obter mais informações gerais sobre RCPs, consulte [Políticas de controle de recursos](#) no Guia AWS Organizations do usuário.
- Para obter detalhes sobre como definir RCPs, incluindo exemplos, consulte a [sintaxe RCP](#) no Guia do AWS Organizations Usuário.

O exemplo a seguir demonstra como usar uma RCP para impedir que entidades principais externas acessem chaves gerenciadas pelo cliente em sua organização. Essa política é somente um exemplo. Você deverá adaptá-la para atender às suas necessidades específicas de negócios e segurança. Por exemplo, talvez seja necessário personalizar sua política para permitir o acesso de seus parceiros de negócios. Para obter mais detalhes, consulte o [repositório de exemplos de política de perímetro de dados](#).

 Note

A permissão `kms:RetireGrant` não tem efeito em uma RCP, mesmo que o elemento Action especifique um asterisco (*) como curinga.

Para obter mais informações sobre como a permissão `kms:RetireGrant` é determinada, consulte [Retirar e revogar concessões](#).

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"
        }
      }
    }
  ]
}
```

O exemplo de RCP a seguir exige que os diretores de AWS serviço só possam acessar suas chaves KMS gerenciadas pelo cliente quando a solicitação for originada de sua organização. Esta política aplica o controle somente às solicitações que têm `aws:SourceAccount` presente. Isso garante que as integrações de serviços que não exigem o uso de `aws:SourceAccount` não sejam afetadas. Se `aws:SourceAccount` estiver presente no contexto da solicitação, a condição `Null` será avaliada como `true`, fazendo com que a chave `aws:SourceOrgID` seja aplicada.

Para obter mais informações sobre o problema de “confused deputy”, consulte [O problema de “confused deputy”](#) no Guia do usuário do 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"
        }
      }
    }
  ]
}
```

Subsídios em AWS KMS

Uma concessão é um instrumento de política que permite que as [entidades principais da AWS](#) usem chaves do KMS em operações de criptografia. Ela também pode permitir que essas entidades visualizem uma chave do KMS (`DescribeKey`) e criem e gerenciem concessões. Ao autorizar o acesso a uma chave do KMS, concessões são consideradas junto com [políticas de chave](#) e [políticas do IAM](#). Concessões geralmente são usadas para permissões temporárias, pois você pode criar uma, usar suas permissões e excluí-la sem alterar suas principais políticas ou políticas do IAM.

As concessões são comumente usadas por AWS serviços que se integram AWS KMS para criptografar seus dados em repouso. O serviço cria uma concessão em nome de um usuário na conta, usa suas permissões e desativa a concessão assim que sua tarefa é concluída. Para obter

detalhes sobre como AWS os serviços usam concessões, consulte o tópico [Criptografia em repouso](#) no guia do usuário ou no guia do desenvolvedor do serviço.

Concessões são um mecanismo de controle de acesso muito flexível e útil. Quando você cria uma concessão para uma chave do KMS, essa concessão permite que a entidade principal autorizada chame as operações de concessão especificadas na chave do KMS, desde que todas as condições especificadas na concessão sejam atendidas.

- Cada concessão permite o acesso a exatamente uma chave do KMS. Você pode criar uma concessão para uma chave do KMS em uma Conta da AWS diferente.
- Uma concessão pode permitir acesso a uma chave do KMS, mas não pode negar acesso.
- Cada concessão tem uma [entidade principal autorizada](#). O principal beneficiário pode representar uma ou mais identidades na Conta da AWS mesma chave KMS ou em uma conta diferente.
- Uma concessão só pode permitir [operações de concessão](#). Operações de concessão devem ter suporte pela chave do KMS na concessão. Se você especificar uma operação sem suporte, a [CreateGrant](#) solicitação falhará com uma `ValidationError` exceção.
- A entidade principal autorizada pode usar as permissões autorizadas pela concessão sem especificar a concessão, exatamente como fariam se as permissões fossem provenientes de uma política de chave ou de uma política do IAM. No entanto, como a AWS KMS API segue um modelo de [consistência eventual](#), quando você cria, retira ou revoga uma concessão, pode haver um breve atraso antes que a alteração esteja disponível por completo. AWS KMS Para usar as permissões em uma concessão imediatamente, [use um token de concessão](#).
- Uma entidade principal autorizada pode excluir a concessão ([retirar](#) ou [revogar](#) a concessão). A exclusão de uma concessão elimina todas as permissões permitidas por ela. Você não precisa descobrir quais políticas adicionar ou remover para desfazer a concessão.
- AWS KMS limita o número de concessões em cada chave KMS. Para obter detalhes, consulte [Concessões por chave do KMS: 50.000](#).

Tenha cautela ao criar concessões e ao dar permissão a outras pessoas para criar concessões. A permissão para criar concessões tem implicações de segurança, assim como permitir que o [kms:PutKeyPolicy](#) defina políticas.

- Usuários com permissão para criar concessões para uma chave KMS (`kms:CreateGrant`) podem usar uma concessão para permitir que usuários e funções, incluindo AWS serviços, usem a chave KMS. Os diretores podem ser identidades próprias Conta da AWS ou identidades em uma conta ou organização diferente.

- Os subsídios podem permitir somente um subconjunto de AWS KMS operações. É possível usar concessões para permitir que as entidades principais exibam a chave do KMS, usá-la em operações de criptografia e criar e retirar concessões. Para obter mais detalhes, consulte [Operações de concessão](#). Você também pode usar [restrições de concessão](#) para limitar as permissões em uma concessão para uma chave de criptografia simétrica.
- As entidades principais podem obter permissão para criar concessões a partir de uma política de chave ou de uma política do IAM. Entidades principais que recebem a permissão `kms:CreateGrant` de uma política podem criar concessões para qualquer [operação de concessão](#) na chave do KMS. Essas entidades principais não precisam ter a permissão que estão concedendo na chave. Quando você concede a permissão `kms:CreateGrant` em uma política, pode usar [condições de políticas](#) para limitar essa permissão.
- As entidades principais também podem obter permissão para criar concessões de uma concessão. Mesmo que tenham outras permissões de uma política, essas entidades principais só podem delegar as permissões que foram concedidas a elas. Para obter detalhes, consulte [Concedendo permissão CreateGrant](#).

Conceitos sobre concessões

Para usar concessões de maneira eficaz, você precisa entender os termos e conceitos que são usados pela AWS KMS .

Restrições de concessão

Uma condição que limita as permissões na concessão. Atualmente, AWS KMS suporta restrições de concessão com base no [contexto de criptografia](#) na solicitação de uma operação criptográfica. Para obter detalhes, consulte [Usar restrições de concessão](#).

ID de concessão

O identificador exclusivo de uma concessão para uma chave do KMS. Você pode usar uma ID de concessão, junto com um [identificador de chave](#), para identificar uma concessão em uma [RevokeGrants](#)solicitação [RetireGrantor](#).

Operações de concessão

As AWS KMS operações que você pode permitir em uma concessão. Se você especificar outras operações, a [CreateGrants](#)solicitação falhará com uma `ValidationError` exceção. Estas são também as operações que aceitam um [token de concessão](#). Para obter informações detalhadas sobre essas permissões, consulte a [AWS KMS permissões](#).

Essas operações de concessão realmente representam permissão para usar a operação. Portanto, para a operação `ReEncrypt`, você pode especificar `ReEncryptFrom`, `ReEncryptTo` ou ambos `ReEncrypt*`.

As operações de concessão são:

- Operações criptográficas
 - [Decrypt](#)
 - [DeriveSharedSecret](#)
 - [Encrypt](#)
 - [GenerateDataKey](#)
 - [GenerateDataKeyPair](#)
 - [GenerateDataKeyPairWithoutPlaintext](#)
 - [GenerateDataKeyWithoutPlaintext](#)
 - [GenerateMac](#)
 - [ReEncryptFrom](#)
 - [ReEncryptTo](#)
 - [Sign](#)
 - [Verificar](#)
 - [VerifyMac](#)
- Outras operações
 - [CreateGrant](#)
 - [DescribeKey](#)
 - [GetPublicKey](#)
 - [RetireGrant](#)

As operações de concessão permitidas devem ter suporte pela chave do KMS na concessão. Se você especificar uma operação sem suporte, a [CreateGrant](#) solicitação falhará com uma `ValidationError` exceção. Por exemplo, as concessões para chaves do KMS de criptografia simétrica não podem permitir as operações [Sign](#), [Verify](#), [GenerateMac](#) ou [VerifyMac](#). As concessões para chaves assimétricas do KMS não podem permitir nenhuma operação que gere chaves de dados ou pares de chaves de dados.

Token de concessão

A AWS KMS API segue um modelo de [consistência eventual](#). Quando você cria uma concessão, pode haver um breve atraso antes que a alteração esteja disponível em todo o AWS KMS. Normalmente, a alteração leva menos de alguns segundos para se propagar por todo o sistema, mas, em alguns casos, pode levar vários minutos. Se você tentar usar uma concessão antes de ela se propagar totalmente pelo sistema, poderá obter um erro de acesso negado. Um token de concessão permite que você faça referência à concessão e use as permissões de concessão imediatamente.

Um token de concessão é uma string única, não secreta, de comprimento variável e codificada em base64 que representa uma concessão. Você pode usar o token de concessão para identificar a concessão em qualquer [operação de concessão](#). No entanto, como o valor do token é um resumo de hash, ele não revela detalhes sobre a concessão.

Um token de concessão é projetado para ser usado somente até que a concessão seja totalmente propagada em todo o AWS KMS. Depois disso, a [entidade principal receptora da concessão](#) pode usar a permissão nessa concessão sem fornecer um token de concessão ou qualquer outra evidência da concessão. Você pode usar um token de concessão a qualquer momento, mas quando a concessão estiver consistente, AWS KMS use a concessão para determinar as permissões, não o token de concessão.

Por exemplo, o comando a seguir chama a [GenerateDataKey](#) operação. Ele usa um token de concessão para representar a concessão que dá ao autor da chamada (a entidade principal receptora da concessão) permissão para chamar GenerateDataKey na chave do KMS especificada.

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

Também é possível usar um token de concessão para identificar uma concessão em operações que gerenciam concessões. Por exemplo, o [diretor que está se aposentando](#) pode usar um token de concessão em uma chamada para a [RetireGrant](#) operação.

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

`CreateGrant` é a única operação que retorna um token de concessão. Você não pode obter um token de concessão de nenhuma outra AWS KMS operação ou do [evento de CloudTrail log](#) da `CreateGrant` operação. As [ListRetirableGrants](#) operações [ListGrant](#)se retornam o [ID de concessão](#), mas não um token de concessão.

Para obter detalhes, consulte [Usar um token de concessão](#).

Entidade principal receptora da concessão

As identidades que obtêm as permissões especificadas na concessão. Cada concessão tem uma entidade principal autorizada, mas a entidade principal autorizada pode representar várias identidades.

O principal beneficiário pode ser qualquer AWS principal, incluindo um Conta da AWS (root), um [usuário do IAM](#), uma função do [IAM](#), uma função ou usuário [federado ou um usuário com função](#) assumida. A entidade principal receptora da concessão pode estar na mesma conta da que a chave do KMS ou uma conta diferente. No entanto, a entidade principal receptora da concessão não pode ser uma [entidade principal do serviço](#), um [grupo do IAM](#) ou um [AWS organização](#).

Note

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Retira (uma concessão)

Encerra uma concessão. Você retira uma concessão ao terminar de usar as permissões.

A revogação e a retirada de uma concessão excluem essa concessão. No entanto, o processo é feito por uma entidade principal especificada na concessão. A revogação geralmente é feita por um administrador de chave. Para obter detalhes, consulte [Retirar e revogar concessões](#).

Retirada da entidade principal

Uma entidade principal que pode [retirar uma concessão](#). Você pode especificar uma entidade de retirada em uma concessão, mas ela não é necessária. O diretor que está se aposentando pode ser qualquer AWS principal, incluindo usuários do IAM Contas da AWS, funções do IAM, usuários

federados e usuários com funções assumidas. A entidade principal de retirada pode estar na mesma conta da que a chave do KMS ou uma conta diferente.

 Note

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Além de se aposentar do diretor especificado no subsídio, um subsídio pode ser aposentado pela pessoa Conta da AWS em que o subsídio foi criado. Se a concessão permitir a operação `RetireGrant`, a [entidade principal receptora da concessão](#) poderá retirar a concessão. Além disso, o Conta da AWS ou Conta da AWS aquele que é o diretor que está se aposentando pode delegar a permissão para retirar uma bolsa a um diretor do IAM no mesmo. Conta da AWS Para obter detalhes, consulte [Retirar e revogar concessões](#).

Revocar (uma concessão)

Encerra uma concessão. Você revoga uma concessão para negar ativamente as permissões que a concessão permite.

A revogação e a retirada de uma concessão excluem essa concessão. No entanto, o processo é feito por uma entidade principal especificada na concessão. A revogação geralmente é feita por um administrador de chave. Para obter detalhes, consulte [Retirar e revogar concessões](#).

Consistência eventual (para concessões)

A AWS KMS API segue um modelo de [consistência eventual](#). Quando você cria, retira ou revoga uma concessão, pode haver um breve atraso antes que a alteração esteja disponível em todo o AWS KMS. Normalmente, a alteração leva menos de alguns segundos para se propagar por todo o sistema, mas, em alguns casos, pode levar vários minutos.

Você pode ter conhecimento desse breve atraso se receber erros inesperados. Por exemplo, se você tentar gerenciar uma nova concessão ou usar as permissões em uma nova concessão antes que a concessão seja totalmente conhecida AWS KMS, você pode receber um erro de acesso negado. Se você retirar ou revogar uma concessão, a entidade principal receptora da concessão ainda poderá usar suas permissões por um breve período até que a concessão seja totalmente excluída. A estratégia típica é repetir a solicitação, e algumas AWS SDKs incluem lógica automática de recuo e repetição.

AWS KMS tem recursos para mitigar esse breve atraso.

- Para usar as permissões em uma nova concessão imediatamente, use um [token de concessão](#). Você pode usar um token de concessão para fazer referência a uma concessão em qualquer [operação de concessão](#). Para instruções, consulte [Usar um token de concessão](#).
- A [CreateGrant](#) operação tem um Name parâmetro que impede que operações de repetição criem concessões duplicadas.

Note

Os tokens de concessão substituem a validade da concessão até que todos os endpoints do serviço sejam atualizados com o novo estado de concessão. Na maioria dos casos, a consistência eventual será alcançada em cinco minutos.

Para obter informações, consulte [consistência eventual do AWS KMS](#).

Melhores práticas para AWS KMS subsídios

AWS KMS recomenda as seguintes melhores práticas ao criar, usar e gerenciar subsídios.

- Limite as permissões na concessão àquelas que são exigidas pela entidade principal receptora da concessão. Use o princípio de [acesso com menos privilégio](#).
- Use uma entidade principal receptora da concessão específica, como uma função do IAM, e dê a ela permissão para usar somente as operações de API necessárias.
- Use [restrições de concessão](#) do contexto de criptografia para garantir que os autores de chamadas estejam usando a chave do KMS para o propósito pretendido. Para obter detalhes sobre como usar o contexto de criptografia em uma solicitação para proteger seus dados, consulte [Como proteger a integridade de seus dados criptografados usando AWS Key Management Service e EncryptionContext](#) no blog AWS de segurança.

Tip

Use a restrição de [EncryptionContextEqual](#) concessão sempre que possível. A restrição de [EncryptionContextSubset](#) concessão é mais difícil de usar corretamente. Se precisar usá-la, leia atentamente a documentação e teste a restrição de concessão para se certificar de que ela funciona como pretendido.

- Exclua concessões duplicadas. Concessões duplicadas têm o mesmo ARN de chave, ações de API, entidade principal receptora da concessão, contexto de criptografia e nome. Se você retirar ou revogar a concessão original, mas deixar as duplicatas, as concessões duplicadas restantes constituirão escalonamentos não intencionais de privilégio. Para evitar duplicar concessões ao tentar novamente uma solicitação `CreateGrant`, use o [parâmetro Name](#). Para detectar concessões duplicadas, use a [ListGrants](#) operação. Se você criar acidentalmente uma concessão duplicada, retire ou revogue-a assim que possível.

Note

Concessões para [chaves gerenciadas da AWS](#) podem parecer duplicadas, mas têm diferentes entidades principais receptoras de concessão.

O campo `GranteePrincipal` na resposta `ListGrants` geralmente contém o principal favorecido da concessão. No entanto, quando o principal beneficiário da concessão é um AWS serviço, o `GranteePrincipal` campo contém o [principal do serviço](#), que pode representar vários diretores beneficiários diferentes.

- Lembre-se de que as concessões não expiram automaticamente. [Reitere ou revogue a concessão](#) assim que a permissão não for mais necessária. Concessões que não forem excluídas podem criar riscos de segurança para recursos criptografados.

Controlar o acesso a concessões

Você pode controlar o acesso às operações que criam e gerenciam concessões em políticas de chaves, políticas do IAM e concessões. As entidades principais que recebem a permissão `CreateGrant` de uma concessão tem [permissões de concessão mais limitadas](#).

Operação de API	Política de chaves ou política do IAM	Concessão
<code>CreateGrant</code>	✓	✓
<code>ListGrants</code>	✓	-
<code>ListRetirableGrants</code>	✓	-

Operação de API	Política de chaves ou política do IAM	Concessão
Retirar concessões	(Limitado. Consulte Retirar e revogar concessões)	✓
RevokeGrant	✓	-

Ao usar uma política de chaves ou uma política do IAM para controlar o acesso às operações que criam e gerenciam concessões, você pode usar uma ou mais das seguintes condições de política para limitar a permissão. AWS KMS suporta todas as seguintes chaves de condição relacionadas à concessão. Para obter informações e exemplos detalhados, consulte [AWS KMS chaves de condição](#).

[kms: GrantConstraintType](#)

Permite que as entidades principais criem uma concessão somente quando esta inclui a [restrição de concessão](#) especificada.

[kms: GrantsFor AWSResource](#)

Permite que os diretores CreateGrant liguem ou RevokeGrant somente quando [um AWS serviço integrado AWS KMS](#) envia a solicitação em nome do diretor. ListGrants

[kms: GrantOperations](#)

Permite que as entidades principais criem uma concessão, mas limita a concessão às operações especificadas.

[kms: GranteePrincipal](#)

Permite que as entidades principais criem uma concessão somente para a [entidade principal receptora da concessão](#).

[kms: RetiringPrincipal](#)

Permite que as entidades principais criem uma concessão somente quando esta especifica um [entidade principal de retirada](#).

Criar concessões

Antes de criar uma concessão, saiba mais sobre as opções para personalizar sua concessão. Você pode usar restrições de concessão para limitar as permissões na concessão. Além disso, saiba mais

sobre como conceder a permissão `CreateGrant`. As entidades principais que obtêm permissão para criar concessões a partir de uma concessão são limitados em termos das concessões que elas podem criar.

Tópicos

- [Criar uma concessão](#)
- [Concedendo permissão `CreateGrant`](#)

Criar uma concessão

Para criar uma concessão, chame a [CreateGrant](#) operação. Especifique uma chave do KMS, uma [entidade principal receptora da concessão](#) e uma lista de [operações de concessão](#). Você também pode designar uma [entidade principal de retirada](#) opcional. Para personalizar a concessão, use parâmetros Constraints opcionais para definir as [restrições de concessão](#)

Quando você cria, retira ou revoga uma concessão, pode haver um breve atraso, geralmente menos de cinco minutos, antes que a alteração esteja disponível em todo o AWS KMS. Para obter informações, consulte [Eventual consistency \(for grants\)](#) (Consistência eventual (para concessões)).

Por exemplo, o comando `CreateGrant` a seguir cria uma concessão que permite que usuários que estão autorizados a assumir o perfil `keyUserRole` chamem a operação [Decrypt](#) na [chave do KMS simétrica](#) especificada. A concessão usa o parâmetro `RetiringPrincipal` para designar um principal que pode desativar a concessão. Ele também inclui uma restrição de concessão que concede a permissão somente quando o [contexto de criptografia](#) na solicitação inclui `"Department": "IT"`.

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

Se o seu código tentar novamente a operação `CreateGrant` ou usar um [SDK da AWS que repete solicitações automaticamente](#), use o parâmetro [Name](#) opcional para impedir a criação de concessões duplicadas. Se AWS KMS receber uma `CreateGrant` solicitação de concessão com as mesmas propriedades de uma concessão existente, incluindo o nome, ela reconhecerá a solicitação como

uma nova tentativa e não criará uma nova concessão. Não é possível usar o valor Name para identificar a concessão em qualquer operação do AWS KMS .

 Important

Não inclua informações confidenciais ou sigilosas no nome da concessão. Ele pode aparecer em texto simples em CloudTrail registros e outras saídas.

```
$ 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}
```

Para exemplos de código que demonstram como criar concessões em várias linguagens de programação, consulte [Use CreateGrant com um AWS SDK ou CLI](#).

Usar restrições de concessão

[Restrições de concessão](#) definem condições para as permissões que a concessão fornece à entidade principal receptora da concessão. Restrições de concessão ocupam o lugar de [chaves de condição](#) em uma [política de chaves](#) ou [política do IAM](#). Cada valor de restrição de concessão pode incluir até oito pares de contexto de criptografia. O valor do contexto de criptografia em cada restrição de concessão não pode exceder 384 caracteres.

 Important

Não inclua informações confidenciais ou sigilosas nesse campo. Esse campo pode ser exibido em texto simples em CloudTrail registros e outras saídas.

AWS KMS suporta duas restrições de concessão `EncryptionContextEquals` e `EncryptionContextSubset` ambas estabelecem requisitos para o [contexto de criptografia](#) em uma solicitação de uma operação criptográfica.

As restrições de concessão de contexto de criptografia foram projetadas para serem usadas com [operações de concessão](#) que têm um parâmetro de contexto de criptografia.

- As restrições de contexto de criptografia são válidas somente em uma concessão para uma chave do KMS de criptografia simétrica. Operações criptográficas com outras chaves do KMS não são compatíveis com um contexto de criptografia.
- A restrição do contexto de criptografia é ignorada para operações `DescribeKey` e `RetireGrant`. `DescribeKey` e `RetireGrant` não têm um parâmetro de contexto de criptografia, mas você pode incluir essas operações em uma concessão que tenha uma restrição de contexto de criptografia.
- É possível usar uma restrição de contexto de criptografia em uma concessão para a operação `CreateGrant`. A restrição de contexto de criptografia requer que todas as concessões criadas com a permissão `CreateGrant` tenham uma restrição de contexto de criptografia igualmente rigorosa ou ainda mais rigorosa.

AWS KMS suporta as seguintes restrições de concessão de contexto de criptografia.

EncryptionContextEquals

Use `EncryptionContextEquals` para especificar o contexto de criptografia exato para solicitações permitidas.

`EncryptionContextEquals` requer que os pares de contexto de criptografia na solicitação sejam uma correspondência exata, com distinção entre maiúsculas e minúsculas, para os pares de contexto de criptografia na restrição de concessão. Os pares podem aparecer em qualquer ordem, mas as chaves e os valores em cada par não podem variar.

Por exemplo, se a restrição de concessão `EncryptionContextEquals` exigir o par de contexto de criptografia `"Department": "IT"`, a concessão permitirá solicitações do tipo especificado apenas quando o contexto de criptografia na solicitação for exatamente `"Department": "IT"`.

EncryptionContextSubset

Use `EncryptionContextSubset` para exigir que as solicitações incluam pares de contexto de criptografia específicos.

`EncryptionContextSubset` requer que a solicitação inclua todos os pares de contexto de criptografia na restrição de concessão (uma correspondência exata com distinção entre maiúsculas e minúsculas), mas a solicitação também pode conter pares de contexto de criptografia adicionais. Os pares podem aparecer em qualquer ordem, mas as chaves e os valores em cada par não podem variar.

Por exemplo, se a restrição de concessão `EncryptionContextSubset` exigir o par de contexto de criptografia `Department=IT`, a concessão permitirá solicitações do tipo especificado quando o contexto de criptografia na solicitação for `"Department": "IT"` ou incluir `"Department": "IT"` juntamente com outros pares de contexto de criptografia, como `"Department": "IT", "Purpose": "Test"`.

Para especificar uma restrição de contexto de criptografia em uma concessão para uma chave KMS de criptografia simétrica, use o `Constraints` parâmetro na operação. [CreateGrant](#) A concessão que esse comando cria concede aos usuários que estão autorizados a assumir a permissão do perfil `keyUserRole` para chamar a operação [Decrypt](#). Porém, essa permissão entrará em vigor somente quando o contexto de criptografia na solicitação `Decrypt` for um par de contexto de criptografia `"Department": "IT"`.

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

A concessão resultante é semelhante à seguinte. Observe que a permissão concedida ao perfil `keyUserRole` entrará em vigor somente quando a solicitação `Decrypt` usar o mesmo par de contexto de criptografia especificado na restrição de concessão. Para encontrar as concessões em uma chave KMS, use a [ListGrants](#) operação.

```
$ 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"
}
]
}

```

Para atender à restrição de concessão `EncryptionContextEquals`, o contexto de criptografia na solicitação para a operação `Decrypt` deve ser um par `"Department": "IT"`. Uma solicitação como a seguinte da entidade principal receptora da concessão atenderia à restrição de concessão `EncryptionContextEquals`.

```

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

```

Quando a restrição de concessão é `EncryptionContextSubset`, os pares de contexto de criptografia na solicitação devem incluir pares de contexto de criptografia na restrição de concessão, mas a solicitação também pode incluir outros pares de contexto de criptografia. A restrição de concessão a seguir exige que um dos pares de contexto de criptografia na solicitação seja `"Department": "IT"`.

```

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

```

A solicitação a seguir da entidade principal receptora da concessão atenderia a ambas as restrições de concessão `EncryptionContextEqual` e `EncryptionContextSubset` neste exemplo.

```

$ 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
```

No entanto, uma solicitação como o seguinte da entidade principal receptora da concessão atenderia a restrição de concessão `EncryptionContextSubset`, mas falharia na restrição de concessão `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 os serviços geralmente usam restrições de contexto de criptografia nas concessões que lhes dão permissão para usar chaves KMS em seu. Conta da AWS Por exemplo, o Amazon DynamoDB usa uma concessão, como a seguinte, para obter permissão de uso da [Chave gerenciada pela AWS](#) para o DynamoDB na sua conta. A restrição de concessão `EncryptionContextSubset` nessa concessão colocará em vigor as permissões na concessão somente quando o contexto de criptografia na solicitação incluir os pares `"subscriberID": "111122223333"` e `"tableName": "Services"`. Essa restrição de concessão significa que a concessão permite que o DynamoDB use a chave do KMS especificada apenas para uma tabela específica na sua Conta da AWS.

Para obter essa saída, execute a [ListGrants](#) operação no Chave gerenciada pela AWS for DynamoDB em sua conta.

```
$ 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"
}
]
}

```

Concedendo permissão CreateGrant

Uma concessão pode incluir permissão para chamar a operação `CreateGrant`. Porém, quando uma [entidade principal receptora da concessão](#) obtém permissão para chamar `CreateGrant` de uma concessão, em vez de uma política, essa permissão é limitada.

- A entidade principal receptora da concessão só pode criar concessões que permitam algumas ou todas as operações na concessão pai.
- As [restrições de concessão](#) nas concessões que elas criam devem ser pelo menos tão rigorosas quanto as contidos na concessão pai.

Essas limitações não se aplicam a entidades principais que obtêm a permissão `CreateGrant` de uma política, embora suas permissões possam ser limitadas por [condições de política](#).

Por exemplo, considere uma concessão que permite que o principal favorecido chame as operações `GenerateDataKey`, `Decrypt` e `CreateGrant`. Chamamos uma concessão que aceita a permissão `CreateGrant` para uma concessão pai.

```

# 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"
        }
    },
}
]
}

```

A entidade principal receptora da concessão, `exampleUser`, pode usar essa permissão para criar uma concessão que inclui qualquer subconjunto das operações especificadas na concessão original, como `CreateGrant` e `Decrypt`. A concessão filho não pode incluir outras operações, como `ScheduleKeyDeletion` ou `ReEncrypt`.

Além disso, as [restrições de concessão](#) nas concessões filho devem ser igualmente ou mais restritivas que as da concessão pai. Por exemplo, a concessão filho pode adicionar pares a uma restrição `EncryptionContextSubset` na concessão pai, mas não pode removê-los. A concessão filho pode alterar uma restrição `EncryptionContextSubset` para uma restrição `EncryptionContextEquals`, mas não o contrário.

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Por exemplo, a entidade principal receptora da concessão pode usar a permissão `CreateGrant` que obteve da concessão pai para criar a seguinte concessão filho. As operações na concessão filho são um subconjunto das operações na concessão pai, e as restrições de concessão são mais restritivas.


```
# 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"
        }
      },
    }
  ]
}
```

A entidade principal receptora da concessão na concessão filho, `anotherUser`, pode usar a permissão `CreateGrant` para criar concessões. No entanto, as concessões criadas por `anotherUser` devem incluir as operações em sua concessão pai ou um subconjunto, e as restrições de concessão devem ser iguais ou mais rigorosas.

Visualizar concessões

Para visualizar a concessão, use a [ListGrants](#) operação. Você deve especificar a chave do KMS à qual as concessões se aplicam. Você também pode filtrar a lista de concessões por ID de concessão ou entidade principal receptora da concessão. Para obter mais exemplos, consulte [Use ListGrants com um AWS SDK ou CLI](#).

Para ver todas as concessões na Conta da AWS região com um determinado [diretor aposentado](#), use [ListRetirableGrants](#). As respostas incluem detalhes sobre cada concessão.

 Note

O campo `GranteePrincipal` na resposta `ListGrants` geralmente contém o principal favorecido da concessão. No entanto, quando o principal beneficiário da concessão é um AWS serviço, o `GranteePrincipal` campo contém o [principal do serviço](#), que pode representar vários diretores beneficiários diferentes.

Por exemplo, o comando a seguir lista todas as concessões para uma chave do 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"
      ]
    }
  ]
}
```

Usar um token de concessão

A AWS KMS API segue um modelo de [consistência eventual](#). Ao criar uma concessão, esta pode não entrar em vigor imediatamente. Pode haver um breve atraso antes que a alteração esteja disponível em todo o AWS KMS. Normalmente, a alteração leva menos de alguns segundos para se propagar por todo o sistema, mas, em alguns casos, pode levar vários minutos. Uma vez que

a alteração tenha se propagado em todo o sistema, a entidade principal receptora da concessão poderá usar as permissões na concessão sem especificar o token de concessão ou qualquer evidência da concessão. No entanto, se uma concessão for tão nova que ainda não seja conhecida por todos AWS KMS, a solicitação poderá falhar com um `AccessDeniedException` erro.

Para usar as permissões em uma nova concessão imediatamente, use o [token de concessão](#) para a concessão. Salve o token de concessão que a [CreateGrant](#) operação retorna. Em seguida, envie o token de concessão na solicitação da AWS KMS operação. Você pode enviar um token de concessão para qualquer [operação de AWS KMS concessão](#) e pode enviar vários tokens de concessão na mesma solicitação.

O exemplo a seguir usa a `CreateGrant` operação para criar uma concessão que permite as operações [GenerateDataKey](#) [Decrypt](#). Ele salva o token de concessão retornado por `CreateGrant` na variável `token`. Em seguida, em uma chamada para `GenerateDataKey`, ele usa o token de concessão na variável `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
```

As entidades principais com permissão também podem usar um token de concessão para desativar uma nova concessão antes mesmo que ela fique disponível por meio do AWS KMS. (A operação `RevokeGrant` não aceita um token de concessão.) Para obter detalhes, consulte [Retirar e revogar concessões](#).

```
# Retire the grant
$ aws kms retire-grant --grant-token $token
```

Retirar e revogar concessões

Para excluir uma concessão, retire-a ou revogue-a.

As [RevokeGrant](#) operações [RetireGrante](#) são muito semelhantes entre si. Ambas excluem uma concessão, o que elimina as permissões por ela permitidas. A principal diferença entre elas é como elas são autorizadas.

RevokeGrant

Como a maioria das AWS KMS operações, o acesso à RevokeGrant operação é controlado por meio de [políticas-chave e políticas do IAM](#). A [RevokeGrant](#) API pode ser chamada por qualquer diretor com `kms:RevokeGrant` permissão. Essa permissão está incluída nas permissões padrão fornecidas aos administradores de chaves. Normalmente, os administradores revogam uma concessão para negar permissões que são permitidas pela concessão.

RetireGrant

A concessão determina quem pode retirá-la. Esse design permite que você controle o ciclo de vida de uma concessão sem alterar políticas de chaves ou políticas do IAM. Normalmente, você retira uma concessão ao terminar de usar suas permissões.

Uma concessão pode ser retirada por uma [entidade principal de retirada](#) especificada nessa concessão. A [entidade principal receptora da concessão](#) também pode retirar a concessão, mas somente se ela também for uma entidade principal de retirada ou se a concessão incluir a operação `RetireGrant`. Como backup, aquele Conta da AWS em que o subsídio foi criado pode retirar o subsídio.

Existe uma permissão `kms:RetireGrant` que pode ser usada em políticas do IAM, mas sua utilidade é limitada. Entidades principais especificadas na concessão podem retirar uma concessão sem a permissão `kms:RetireGrant`. A permissão `kms:RetireGrant` por si só não permite que as entidades principais retirem uma concessão. A permissão de `kms:RetireGrant` não tem efeito em uma [política de chave](#) ou [política de controle de recurso](#).

- Para negar a permissão de retirar uma concessão, você pode usar uma ação `Deny` com a permissão de `kms:RetireGrant` em suas políticas do IAM.
- O Conta da AWS proprietário da chave KMS pode delegar a `kms:RetireGrant` permissão a um diretor do IAM na conta.

- Se o diretor aposentado for diferente Conta da AWS, os administradores da outra conta podem usar `kms:RetireGrant` para delegar permissão para retirar a concessão a um diretor do IAM nessa conta.

A AWS KMS API segue um modelo de [consistência eventual](#). Quando você cria, retira ou revoga uma concessão, pode haver um breve atraso antes que a alteração esteja disponível em todo o AWS KMS. Normalmente, a alteração leva menos de alguns segundos para se propagar por todo o sistema, mas, em alguns casos, pode levar vários minutos. Se você precisar excluir uma nova concessão imediatamente, antes que ela esteja disponível por toda parte AWS KMS, [use um token de concessão](#) para retirar a concessão. Não é possível usar um token de concessão para revogar uma concessão.

Chaves de condição para AWS KMS

Você pode especificar condições nas [principais políticas e políticas do IAM](#) que controlam o acesso aos AWS KMS recursos. A declaração de política é efetiva apenas quando as condições forem verdadeiras. Por exemplo, talvez você deseje que uma declaração de política só entre em vigor após uma data específica. Ou, você pode querer que uma declaração de política controle o acesso apenas quando um valor específico existe em uma solicitação de API.

Para especificar condições, use chaves de condição no [elemento Condition](#) de uma instrução de política com [operadores de condição do IAM](#). Algumas chaves de condição geralmente se aplicam a AWS; outras são específicas AWS KMS a.

Os valores da chave de condição devem seguir as regras de caracteres e codificação das AWS KMS principais políticas e políticas do IAM. Para obter detalhes sobre as regras de documento de política de chaves, consulte [Formato de política de chaves](#). Para obter detalhes sobre as regras de documento de política do IAM, consulte [Requisitos de nome do IAM](#) no Guia do usuário do IAM.

Tópicos

- [AWS chaves de condição globais](#)
- [AWS KMS chaves de condição](#)
- [AWS KMS chaves de condição para plataformas certificadas](#)

AWS chaves de condição globais

AWS define [chaves de condição globais](#), um conjunto de chaves de condições de política para todos os AWS serviços que usam o IAM para controle de acesso. AWS KMS suporta todas as chaves de condição globais. Você pode usá-las nas AWS KMS principais políticas e nas políticas do IAM.

Por exemplo, você pode usar a chave de condição [aws: PrincipalArn](#) global para permitir o acesso a uma AWS KMS key (chave KMS) somente quando o principal na solicitação for representado pelo Amazon Resource Name (ARN) no valor da chave de condição. Para oferecer suporte ao [controle de acesso baseado em atributos](#) (ABAC) em AWS KMS, você pode usar a chave de condição global [aws:ResourceTag/tag-key](#) em uma política do IAM para permitir o acesso às chaves do KMS com uma tag específica.

Para ajudar a evitar que um AWS serviço seja usado como substituto confuso em uma política em que o diretor é o principal do [AWS serviço](#), você pode usar as chaves de condição [aws:SourceArn](#) ou as chaves de condição [aws:SourceAccount](#) globais. Para obter detalhes, consulte [Uso de chaves de condição aws:SourceArn ou aws:SourceAccount](#).

Para obter informações sobre chaves de condição AWS globais, incluindo os tipos de solicitações nas quais elas estão disponíveis, consulte [Chaves de contexto de condição AWS global](#) no Guia do usuário do IAM. Para exemplos de como usar chaves de condição globais em políticas do IAM, consulte [Controlar o acesso a solicitações](#) e [Controlar chaves de etiquetas](#), no Manual do usuário do IAM.

Os tópicos a seguir fornecem orientações especiais para o uso de chaves de condição com base em endereços IP e endpoint da VPCs.

Tópicos

- [Usando a condição de endereço IP em políticas com AWS KMS permissões](#)
- [Usando condições de endpoint VPC em políticas com permissões AWS KMS](#)
- [Usando IPv6 endereços no IAM e nas AWS KMS principais políticas](#)

Usando a condição de endereço IP em políticas com AWS KMS permissões

Você pode usar AWS KMS para proteger seus dados em um [AWS serviço integrado](#). Mas tenha cuidado ao especificar os [operadores de condição de endereço IP](#) ou a chave de `aws:SourceIp` condição na mesma declaração de política que permite ou nega acesso a. AWS KMS Por exemplo,

a política em [AWS: Nega acesso a AWS Com base no IP de origem](#) restringe AWS as ações às solicitações do intervalo de IP especificado.

Considere este cenário:

1. Você anexa uma política como a mostrada em [AWS: Nega acesso AWS com base no IP de origem a](#) uma identidade do IAM. Defina o valor da `aws:SourceIp` chave de condição para o intervalo de endereços IP para a empresa do usuário. Essa identidade do IAM tem outras políticas vinculadas que permitem que ela use o Amazon EBS, o Amazon EC2 e o AWS KMS.
2. A identidade tenta vincular um volume do EBS criptografado a uma instância do EC2. Esta ação falha com um erro de autorização, embora o usuário tenha permissão para usar todos os serviços relevantes.

A etapa 2 falha porque a solicitação AWS KMS para descriptografar a chave de dados criptografada do volume vem de um endereço IP associado à infraestrutura do Amazon EC2. Para que a solicitação seja bem-sucedida, ela deve ter o endereço IP do usuário de origem. Como a política na etapa 1 nega explicitamente todas as solicitações de endereços IP que não sejam aqueles especificados, o Amazon EC2 não recebe permissão para descriptografar a chave de dados criptografada do volume do EBS.

Além disso, a chave de condição `aws:SourceIP` não será efetiva se a solicitação vier de um [endpoint da Amazon VPC](#). Para restringir solicitações a um endpoint da VPC, incluindo um [endpoint da VPC do AWS KMS](#), use as chaves de condições `aws:SourceVpce` ou `aws:SourceVpc`. Para obter mais informações, consulte [endpoint da VPCs – controle do uso de endpoints](#) no Manual do usuário do Amazon VPC.

Usando condições de endpoint VPC em políticas com permissões AWS KMS

[AWS KMS suporta endpoints da Amazon Virtual Private Cloud \(Amazon VPC\)](#) que são alimentados por [AWS PrivateLink](#). Você pode usar as seguintes chaves de [condição globais nas políticas de chaves](#) e nas políticas do IAM para controlar o acesso aos AWS KMS recursos quando a solicitação vem de uma VPC ou usa um VPC endpoint. Para obter detalhes, consulte [Use VPC endpoints para controlar o acesso aos recursos AWS KMS](#).

- `aws:SourceVpc` limita o acesso a solicitações da VPC especificado.
- `aws:SourceVpce` limita o acesso a solicitações do endpoint da VPC especificado.

Se você usar essas chaves de condição para controlar o acesso às chaves KMS, poderá inadvertidamente negar o acesso aos AWS serviços usados AWS KMS em seu nome.

Tome cuidado para evitar uma situação como o exemplo de [chaves de condições de endereço IP](#). Se você restringir as solicitações de uma chave KMS a uma VPC ou VPC endpoint, as chamadas AWS KMS de um serviço integrado, como Amazon S3 ou Amazon EBS, podem falhar. Isso pode acontecer mesmo se a solicitação de origem for originada, em última análise, na VPC ou no endpoint da VPC.

Usando IPv6 endereços no IAM e nas AWS KMS principais políticas

Antes de tentar acessar IPv6, certifique-se AWS KMS de que todas as chaves e políticas do IAM que contenham restrições de endereço IP sejam atualizadas para incluir intervalos de IPv6 endereços. Políticas baseadas em IP que não são atualizadas para lidar com IPv6 endereços podem fazer com que os clientes percam ou obtenham acesso incorretamente quando começam a usar IPv6. Para obter orientação geral sobre controles de acesso do KMS, consulte [Acesso e permissões de chave do KMS](#). Para saber mais sobre o KMS e compatibilidade com pilha dupla, consulte [Suporte a endpoint de pilha dupla](#).

Important

Essas instruções não permitem nenhuma ação. Use essas instruções em combinação com outras instruções que permitem ações específicas.

A declaração a seguir nega explicitamente o acesso a todas as permissões do KMS para solicitações originadas do 192.0.2.* intervalo de endereços. IPv4 As permissões do KMS não são explicitamente negadas aos endereços IP fora desse intervalo. Como todos os IPv6 endereços estão fora do intervalo negado, essa declaração não nega explicitamente as permissões do KMS para nenhum IPv6 endereço.

```
{
  "Sid": "DenyKMSPermissions",
  "Effect": "Deny",
  "Action": [
    "kms:*"
  ],
  "Resource": "*",
  "Condition": {
    "NotIpAddress": {
```



```

        "aws:SourceIp": [
            "192.0.2.0/24"
        ]
    }
}

```

Você pode modificar o Condition elemento para negar os intervalos de endereço IPv4 IPv6 (192.0.2.0/24 2001:db8:1234::/32) e (), conforme mostrado no exemplo a seguir.

```

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

```

AWS KMS chaves de condição

AWS KMS fornece um conjunto de chaves de condição que você pode usar em políticas de chaves e políticas do IAM. Essas chaves de condição são específicas para AWS KMS. Por exemplo, é possível usar a chave de condição `kms:EncryptionContext:context-key` para exigir um [contexto de criptografia](#) específico ao controlar o acesso a uma chave do KMS de criptografia simétrica.

Condições para uma solicitação de operação de API

Muitas chaves de AWS KMS condição controlam o acesso a uma chave KMS com base no valor de um parâmetro na solicitação de uma AWS KMS operação. Por exemplo, você pode usar a chave de KeySpec condição [kms:](#) em uma política do IAM para permitir o uso da [CreateKey](#) operação somente quando o valor do KeySpec parâmetro na CreateKey solicitação for `forRSA_4096`.

Esse tipo de condição funciona mesmo quando o parâmetro não aparece na solicitação, como quando você usa o valor padrão do parâmetro. Por exemplo, é possível usar a chave de condição [kms:KeySpec](#) para permitir que os usuários usem a operação `CreateKey` somente quando o valor do parâmetro `KeySpec` é `SYMMETRIC_DEFAULT`, que é o valor padrão. Essa condição permite solicitações que têm o parâmetro `KeySpec` com o valor `SYMMETRIC_DEFAULT` e solicitações que não têm parâmetro `KeySpec`.

Condições para chaves do KMS usadas em operações de API

Algumas chaves de AWS KMS condição podem controlar o acesso às operações com base em uma propriedade da chave KMS usada na operação. Por exemplo, você pode usar a `KeyOrigin` condição [kms:](#) para permitir que os diretores [GenerateDataKey](#) chamem uma chave KMS somente quando a `Origin` chave KMS for `AWS_KMS`. Para descobrir se uma chave de condição pode ser usada dessa maneira, consulte a descrição da chave de condição.

A operação deve ser uma operação de recurso de chave do KMS, ou seja, uma operação autorizada para uma chave do KMS específica. Para identificar as operações de recursos de chaves do KMS, na [Tabela de ações e recursos](#), procure um valor de KMS `key` na coluna `Resources` para a operação. Se você usar esse tipo de chave de condição com uma operação não autorizada para um determinado recurso de chave KMS, por exemplo [ListKeys](#), a permissão não será efetiva porque a condição nunca poderá ser satisfeita. Não há recursos da chave do KMS envolvidos na autorização da operação `ListKeys` e nenhuma propriedade `KeySpec`.

Os tópicos a seguir descrevem cada chave de AWS KMS condição e incluem exemplos de declarações de política que demonstram a sintaxe da política.

Usar operadores de conjuntos com chaves de condição

Quando uma condição de política compara dois conjuntos de valores, como o conjunto de tags em uma solicitação e o conjunto de tags em uma política, você precisa saber AWS como comparar os conjuntos. O IAM define dois operadores de conjunto, `ForAnyValue` e `ForAllValues`, para essa finalidade. Use operadores de conjunto somente com chaves de condição de vários valores, que exigem esses operadores. Não use operadores de conjuntos com chaves de condição de valor único. Como sempre, teste suas instruções de política completamente antes de usá-las em um ambiente de produção.

Chaves de condição são de valor único ou de vários valores. Para determinar se uma chave de AWS KMS condição tem valor único ou de vários valores, consulte a coluna `Tipo de valor` na descrição da chave de condição.

- Chaves de condição de valor único têm no máximo um valor no contexto de autorização (a solicitação ou o recurso). Por exemplo, como cada chamada de API pode ser originada de apenas uma Conta da AWS, [kms: CallerAccount](#) é uma chave de condição de valor único. Não use um operador de conjunto com uma chave de condição de valor único.
- Chaves de condição de vários valores têm diversos valores no contexto de autorização (a solicitação ou o recurso). Por exemplo, como cada chave KMS pode ter vários aliases, [kms: ResourceAliases](#) pode ter vários valores. Chaves de condição de vários valores exigem um operador de conjunto.

Observe que a diferença entre chaves de condição de valor único e de vários valores depende do número de valores no contexto de autorização, e não do número de valores na condição da política.

Warning

O uso de um operador de conjunto com uma chave de condição de valor único pode criar uma instrução de política excessivamente permissiva (ou excessivamente restritiva). Use operadores de conjunto somente com chaves de condição de vários valores.

Se você criar ou atualizar uma política que inclua um operador de `ForAllValues` conjunto com `kms:EncryptionContext`: chave de contexto ou chaves de `aws:RequestTag/tag-key` condição, AWS KMS retornará a seguinte mensagem de erro:

```
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.
```

Para obter informações detalhadas sobre os operadores de conjunto `ForAnyValue` e `ForAllValues`, consulte [Usar várias chaves e valores](#), no Manual do usuário do IAM. Para obter informações sobre o risco de usar o operador `ForAllValues` set com uma condição de valor único, consulte [Aviso de segurança — ForAllValues com chave de valor único](#) no Guia do usuário do IAM.

Tópicos

- [kms:BypassPolicyLockoutSafetyCheck](#)
- [kms:CallerAccount](#)
- [kms: CustomerMasterKeySpec](#) (obsoleto)
- [kms: CustomerMasterKeyUsage](#) (obsoleto)

- [kms:DataKeyPairSpec](#)
- [kms:EncryptionAlgorithm](#)
- [kms:EncryptionContext: chave de contexto](#)
- [kms:EncryptionContextKeys](#)
- [kms:ExpirationModel](#)
- [kms:GrantConstraintType](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GrantOperations](#)
- [kms:GranteePrincipal](#)
- [kms:KeyAgreementAlgorithm](#)
- [kms:KeyOrigin](#)
- [kms:KeySpec](#)
- [kms:KeyUsage](#)
- [kms:MacAlgorithm](#)
- [kms:MessageType](#)
- [kms:MultiRegion](#)
- [kms:MultiRegionKeyType](#)
- [kms:PrimaryRegion](#)
- [kms:ReEncryptOnSameKey](#)
- [kms:RequestAlias](#)
- [kms:ResourceAliases](#)
- [kms:ReplicaRegion](#)
- [kms:RetiringPrincipal](#)
- [kms:RotationPeriodInDays](#)
- [kms:ScheduleKeyDeletionPendingWindowInDays](#)
- [kms:SigningAlgorithm](#)
- [kms:ValidTo](#)
- [kms:ViaService](#)
- [kms:WrappingAlgorithm](#)
- [kms:WrappingKeySpec](#)

kms:BypassPolicyLockoutSafetyCheck

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:BypassPolicyLockoutSafetyCheck	Booleano	Valor único	CreateKey PutKeyPolicy	Somente políticas do IAM Políticas de chaves e políticas do IAM

A chave de kms:BypassPolicyLockoutSafetyCheck condição controla o acesso às [PutKeyPolicy](#) operações [CreateKey](#) com base no valor do BypassPolicyLockoutSafetyCheck parâmetro na solicitação.

A instrução de política do IAM de exemplo a seguir impede que os usuários ignorem a verificação de segurança de bloqueio da política ao negar a eles a permissão para criar chaves do KMS quando o valor do parâmetro BypassPolicyLockoutSafetyCheck na solicitação CreateKey é true.

```
{
  "Effect": "Deny",
  "Action": [
    "kms:CreateKey",
    "kms:PutKeyPolicy"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

Você também pode usar a chave de condição kms:BypassPolicyLockoutSafetyCheck em uma política do IAM ou política de chaves para controlar o acesso à operação PutKeyPolicy. A declaração de política de exemplo a seguir de uma política de chaves impede que os usuários

ignorem a verificação de segurança de bloqueio da política ao alterar a política de uma chave do KMS.

Em vez de usar Deny de forma explícita, esta declaração de política usa Allow com o [operador de condição nula](#) para permitir o acesso somente quando a solicitação não inclui o parâmetro `BypassPolicyLockoutSafetyCheck`. Quando o parâmetro não for usado, o valor padrão será `false`. Essa instrução de política um pouco mais fraca pode ser substituída no caso raro em que uma derivação é necessária.

```
{
  "Effect": "Allow",
  "Action": "kms:PutKeyPolicy",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

Consulte também

- [kms:KeySpec](#)
- [kms:KeyOrigin](#)
- [kms:KeyUsage](#)

kms:CallerAccount

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:CallerAccount	String	Valor único	Operações de recursos de chaves do KMS Operações do armazenam	Políticas de chaves e políticas do IAM

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política

ento de chaves
personalizado

É possível usar essa chave de condição para permitir ou negar acesso a todas as identidades (usuários e perfis) em uma Conta da AWS. Em políticas de chaves, você pode usar o elemento `Principal` para especificar as identidades às quais a declaração de política se aplica. A sintaxe do elemento `Principal` não fornece uma forma de especificar todas as identidades em uma Conta da AWS. Mas você pode obter esse efeito combinando essa chave de condição com um `Principal` elemento que especifica todas as AWS identidades.

Você pode usá-lo para controlar o acesso a qualquer operação de recurso de chave KMS, ou seja, qualquer AWS KMS operação que use uma chave KMS específica. Para identificar as operações de recursos de chaves do KMS, na [Tabela de ações e recursos](#), procure um valor de `KMS key` na coluna `Resources` para a operação. Ele também é válido para operações que gerenciam [armazenamentos de chaves personalizados](#).

Por exemplo, a instrução de política de chaves a seguir demonstra como usar a chave de condição `kms:CallerAccount`. Esta declaração de política está na política principal do Chave gerenciada pela AWS Amazon EBS. Ele combina um `Principal` elemento que especifica todas as AWS identidades com a chave de `kms:CallerAccount` condição para permitir efetivamente o acesso a todas as identidades em 111122223333. Conta da AWS Ele contém uma chave de AWS KMS condição adicional (`kms:ViaService`) para limitar ainda mais as permissões, permitindo somente solicitações provenientes do Amazon EBS. Para obter mais informações, consulte [kms:ViaService](#).

```
{
  "Sid": "Allow access through EBS for all principals in the account that are
authorized to use EBS",
  "Effect": "Allow",
  "Principal": {"AWS": "*"},
  "Condition": {
    "StringEquals": {
      "kms:CallerAccount": "111122223333",
      "kms:ViaService": "ec2.us-west-2.amazonaws.com"
    }
  }
},
```

```

"Action": [
  "kms:Encrypt",
  "kms:Decrypt",
  "kms:ReEncrypt*",
  "kms:GenerateDataKey*",
  "kms:CreateGrant",
  "kms:DescribeKey"
],
"Resource": "*"
}

```

kms: CustomerMasterKeySpec (obsoleto)

A chave de condição `kms:CustomerMasterKeySpec` está defasada. Em vez disso, use a chave de `KeySpec` condição [kms:](#).

As chaves de condição `kms:CustomerMasterKeySpec` e `kms:KeySpec` funcionam da mesma forma. Apenas os nomes são diferentes. Recomendamos usar o `kms:KeySpec`. No entanto, para evitar alterações significativas, AWS KMS suporta as duas teclas de condição.

kms: CustomerMasterKeyUsage (obsoleto)

A chave de condição `kms:CustomerMasterKeyUsage` está defasada. Em vez disso, use a chave de `KeyUsage` condição [kms:](#).

As chaves de condição `kms:CustomerMasterKeyUsage` e `kms:KeyUsage` funcionam da mesma forma. Apenas os nomes são diferentes. Recomendamos usar o `kms:KeyUsage`. No entanto, para evitar alterações significativas, AWS KMS suporta as duas teclas de condição.

kms:DataKeyPairSpec

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
<code>kms:DataKeyPairSpec</code>	String	Valor único	GeneratedDataKeyPair GeneratedDataKeyPair	Políticas de chaves e políticas do IAM

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
			GenerateDataKeyPairWithoutPlaintext	

Você pode usar essa chave de condição para controlar o acesso às operações [GenerateDataKeyPairWithoutPlaintext](#) com base no valor do `KeyPairSpec` parâmetro na solicitação. Por exemplo, é possível permitir que os usuários gerem somente tipos de pares de chaves de dados específicos.

O exemplo de instrução de política de chaves a seguir usa a chave de condição `kms:DataKeyPairSpec` para permitir que os usuários usem a chave do KMS para gerar somente pares de chaves de dados RSA.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:DataKeyPairSpec": "RSA*"
    }
  }
}
```

Consulte também

- [kms:KeySpec](#)
- [the section called “kms:EncryptionAlgorithm”](#)
- [the section called “kms:EncryptionContext: chave de contexto”](#)
- [the section called “kms:EncryptionContextKeys”](#)

kms:EncryptionAlgorithm

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Encry ptionAlgo rithm	String	Valor único	Decrypt Encrypt GenerateDataKey GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext GenerateDataKeyWithoutPlaintext ReEncrypt	Políticas de chaves e políticas do IAM

É possível usar a chave de condição `kms:EncryptionAlgorithm` para controlar o acesso às operações de criptografia com base no algoritmo de criptografia usado na operação. Para as [ReEncrypt](#) operações [Encrypt](#), [Decrypt](#) e, ele controla o acesso com base no valor do [EncryptionAlgorithm](#) parâmetro na solicitação. Para operações que geram chaves de dados e pares de chaves de dados, ela controla o acesso com base no algoritmo de criptografia que é usado para criptografar a chave de dados.

Essa chave de condição não tem efeito nas operações realizadas fora da AWS KMS, como a criptografia com a chave pública em um par de chaves KMS assimétrico fora da AWS KMS

`EncryptionAlgorithm` parâmetro em uma solicitação

Para permitir que os usuários usem apenas um algoritmo de criptografia específico com uma chave do KMS, use uma instrução de política com um efeito Deny e um operador de condição `StringNotEquals`. Por exemplo, o seguinte exemplo de instrução de política de chaves proíbe que as entidades principais capazes de assumir a função `ExampleRole` usem essa chave do KMS nas operações de criptografia especificadas, a menos que o algoritmo de criptografia na solicitação seja `RSAES_OAEP_SHA_256`, um algoritmo criptográfico assimétrico usado em chaves RSA do KMS.

Diferente de uma instrução de política que permite que um usuário utilize um algoritmo de criptografia específico, uma instrução de política com um duplo negativo como essa evita que outras políticas e concessões da chave do KMS permitam que essa função use outros algoritmos de criptografia. O Deny nessa instrução de política de chaves tem precedência sobre qualquer política de chaves ou política do IAM com um efeito Allow e tem precedência sobre todas as concessões dessa chave do KMS e suas entidades principais.

```
{
  "Sid": "Allow only one encryption algorithm with this asymmetric KMS key",
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*"
  ],
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:EncryptionAlgorithm": "RSAES_OAEP_SHA_256"
    }
  }
}
```

Algoritmo de criptografia usado para a operação

Você também pode usar a chave de condição `kms:EncryptionAlgorithm` para controlar o acesso às operações com base no algoritmo de criptografia utilizado na operação, mesmo quando esse algoritmo não está especificado na solicitação. Isso permite exigir ou proibir o algoritmo `SYMMETRIC_DEFAULT`, que pode não ser especificado em uma solicitação por ser o valor padrão.

Também é possível usar a chave de condição `kms:EncryptionAlgorithm` para controlar o acesso às operações que geram chaves de dados e pares de chaves de dados. Essas operações só usam chaves do KMS de criptografia simétrica e o algoritmo `SYMMETRIC_DEFAULT`.

Por exemplo, essa política do IAM limita seus principais à criptografia simétrica. Ela nega o acesso a qualquer chave do KMS na conta de exemplo para operações de criptografia, a menos que o algoritmo de criptografia especificado na solicitação ou usado na operação seja `SYMMETRIC_DEFAULT`. Incluindo `GenerateDataKey*` [GenerateDataKeyPair](#) [GenerateDataKeyPairWithoutPlaintext](#) e [GenerateDataKeyWithoutPlaintext](#) às permissões. A condição não tem efeito nessas operações porque elas sempre utilizam um algoritmo de criptografia simétrica.

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

Consulte também

- [the section called “kms:MacAlgorithm”](#)
- [kms:SigningAlgorithm](#)

kms:EncryptionContext: chave de contexto

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Encryp tionCont ext: <i>context- key</i>	String	Valor único	CreateGrant Encrypt Decrypt GenerateDataKey GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext GenerateDataKeyWithoutPlaintext ReEncrypt RetireGrant	Políticas de chaves e políticas do IAM

É possível usar a chave de condição `kms:EncryptionContext:context-key` para controlar o acesso a uma [chave do KMS de criptografia simétrica](#) com base no [contexto de criptografia](#) em uma solicitação de uma [operação criptográfica](#). Use essa chave de condição para avaliar a chave e o valor no par de contexto de criptografia. Para avaliar somente as chaves do contexto de criptografia ou exigir um contexto de criptografia, independentemente das chaves ou dos valores, use a chave de EncryptionContextKeys condição [kms:](#).

 Note

Os valores das chaves de condição devem estar em conformidade com as regras de caracteres para políticas de chaves e políticas do IAM. Alguns caracteres que são válidos em um contexto de criptografia não são válidos em políticas. Talvez você não consiga usar essa chave de condição para expressar todos os valores válidos de contexto de criptografia. Para obter detalhes sobre as regras de documento de política de chaves, consulte [Formato de política de chaves](#). Para obter detalhes sobre as regras de documento de política do IAM, consulte [Requisitos de nome do IAM](#) no Guia do usuário do IAM.

Não é possível especificar um contexto de criptografia em uma operação criptográfica com uma [chave do KMS assimétrica](#) ou uma [chave do KMS de Hash-based message authentication code \(HMAC – Código de autenticação de mensagem por hash\)](#). Algoritmos assimétricos e algoritmos de Message authentication code (MAC – Código de autenticação de mensagem) não são compatíveis com um contexto de criptografia.

Para usar a chave de condição `kms:EncryptionContext:` chave de contexto, substitua o *context-key* espaço reservado pela chave de contexto de criptografia. Substitua o espaço reservado *context-value* pelo valor de contexto de criptografia.

```
"kms:EncryptionContext:context-key": "context-value"
```

Por exemplo, a seguinte chave de condição especifica um contexto de criptografia no qual a chave é `AppName` e o valor é `ExampleApp` (`AppName = ExampleApp`).

```
"kms:EncryptionContext:AppName": "ExampleApp"
```

Esta é uma [chave de condição de valor único](#). A chave na chave de condição especifica uma chave de contexto de criptografia específica (`context-key`). Embora seja possível incluir vários pares de contexto de criptografia em cada solicitação de API, o par de contexto de criptografia com a `context-key` especificada pode ter somente um valor. Por exemplo, a chave de condição `kms:EncryptionContext:Department` aplica-se somente a pares de contexto de criptografia com uma chave `Department`, e qualquer par de contexto de criptografia com a chave `Department` pode ter somente um valor.

Não use um operador de conjunto com a chave de condição `kms:EncryptionContext:context-key`. Se você criar uma instrução de política com uma ação

Allow, a chave de condição `kms:EncryptionContext:context-key` e o operador de conjunto `ForAllValues`, a condição permitirá solicitações sem contexto de criptografia e solicitações com pares de contexto de criptografia que não estão especificados na condição de política.

 Warning

Não use um operador de conjunto `ForAnyValue` ou `ForAllValues` com essa chave de condição de valor único. Esses operadores de conjunto podem criar uma condição de política que não requer valores que você pretende exigir e permite valores que você pretende proibir. Se você criar ou atualizar uma política que inclua um operador de `ForAllValues` conjunto com a chave de contexto `kms:EncryptionContext:`, AWS KMS retornará a seguinte mensagem de erro:

```
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.
```

Para exigir um par de contexto de criptografia específico, use a chave de condição `kms:EncryptionContext:context-key` com o operador `StringEquals`.

A seguinte instrução de política de chaves de exemplo permite que as entidades principais que podem assumir a função usem a chave do KMS em uma solicitação `GenerateDataKey` apenas quando o contexto de criptografia nessa solicitação inclui o par `AppName:ExampleApp`. Outros pares de contexto de criptografia são permitidos.

O nome da chave não é sensível a maiúsculas e minúsculas. A diferenciação de maiúsculas e minúsculas do valor é determinada pelo operador de condição, como `StringEquals`. Para obter detalhes, consulte [Diferenciação de letras maiúsculas e minúsculas da condição de contexto de criptografia](#).

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
```

```
"StringEquals": {  
  "kms:EncryptionContext:AppName": "ExampleApp"  
}  
}
```

Para exigir um par de contexto de criptografia e proibir todos os outros pares de contexto de criptografia, use ambos `kms:EncryptionContext`: chave de contexto e [kms:EncryptionContextKeys](#) na declaração de política. A instrução de política de exemplo a seguir usa a condição `kms:EncryptionContext:AppName` para exigir o par de contextos de criptografia `AppName=ExampleApp` na solicitação. Ela também usa uma chave de condição `kms:EncryptionContextKeys` com o operador de conjunto `ForAllValues` para permitir apenas a chave de contexto de criptografia `AppName`.

O operador de conjunto `ForAllValues` limita as chaves de contexto de criptografia na solicitação para `AppName`. Se a condição `kms:EncryptionContextKeys` com o operador de conjunto `ForAllValues` tivesse sido usada sozinha em uma instrução de política, esse operador de conjunto permitiria solicitações sem contexto de criptografia. No entanto, se a solicitação não tivesse um contexto de criptografia, a condição `kms:EncryptionContext:AppName` falharia. Para detalhes sobre o operador de conjunto `ForAllValues`, consulte [Usar várias chaves e valores](#), no Manual do usuário do 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"  
      ]  
    }  
  }  
}
```


Também é possível usar essa chave de condição para negar acesso a uma chave do KMS para uma operação específica. O exemplo a seguir de instrução de política de chaves usa um efeito Deny para proibir a entidade principal de usar a chave do KMS quando o contexto de criptografia na solicitação inclui um par de contextos de criptografia Stage=Restricted. Essa condição permite uma solicitação com outros pares de contextos de criptografia, incluindo pares de contextos de criptografia com a chave Stage e outros valores, como Stage=Test.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": "Restricted"
    }
  }
}
```

Usar vários pares de contextos de criptografia

Você pode exigir ou proibir vários pares de contextos de criptografia. Também pode exigir um dos vários pares de contextos de criptografia. Para obter detalhes sobre a lógica usada para interpretar essas condições, consulte [Criar uma condição com várias chaves ou valores](#), no Manual do usuário do IAM.

Note

As versões anteriores deste tópico exibiam declarações de política que usavam os operadores ForAnyValue e ForAllValues definiam com a chave de condição kms:EncryptionContext: chave de contexto. Usar um operador de conjunto com uma [chave de condição de valor único](#) pode resultar em políticas que permitem solicitações sem contexto de criptografia e pares de contextos de criptografia não especificados. Por exemplo, uma condição de política com o efeito Allow, o operador de conjunto ForAllValues e a chave de condição "kms:EncryptionContext:Department": "IT" não limita o contexto de criptografia ao par "Department=IT". Ele permite solicitações sem contexto de criptografia e solicitações com pares de contextos de criptografia não especificados, como Stage=Restricted.

Revise suas políticas e elimine o operador definido de qualquer condição com `kms:EncryptionContext: context-key`. Tentativas de criar ou atualizar uma política com esse formato falham com uma exceção `OverlyPermissiveCondition`. Para resolver o erro, exclua o operador de conjunto.

Para exigir vários pares de contextos de criptografia, liste esses pares na mesma condição. A seguinte instrução de política de chaves de exemplo requer dois pares de contextos de criptografia: `Department=IT` e `Project=Alpha`. Como as condições têm chaves diferentes (`kms:EncryptionContext:Department` e `kms:EncryptionContext:Project`), elas são implicitamente conectadas por um operador AND. Outros pares de contextos de criptografia são permitidos, mas não são necessários.

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

Para exigir um par de contextos de criptografia OU outro par, coloque cada chave de condição em uma instrução de política separada. O exemplo a seguir de política de chaves requer pares `Department=IT` ou `Project=Alpha`, ou ambos. Outros pares de contextos de criptografia são permitidos, mas não são necessários.

```
{
  "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"
    }
  }
}

```

Para exigir pares de criptografia específicos e excluir todos os outros pares de contexto de criptografia, use `kms:EncryptionContext`: chave de contexto e [kms:EncryptionContextKeys](#) na declaração de política. A seguinte declaração de política de chaves usa a condição `kms:EncryptionContext`: `context-key` para exigir um contexto de criptografia com pares `Department=IT` e `Project=Alpha`. Ela usa uma chave de condição `kms:EncryptionContextKeys` com o operador de conjunto `ForAllValues` para permitir apenas as chaves de contexto de criptografia `Department` e `Project`.

O operador de conjunto `ForAllValues` limita as chaves de contexto de criptografia na solicitação para `Department` e `Project`. Se fosse usado sozinho em uma condição, esse operador de conjunto permitiria solicitações sem contexto de criptografia, mas nessa configuração, a chave de contexto `kms:EncryptionContext`: nessa condição falharia.

```

{
  "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"
    ]
  }
}
```

Você também pode proibir vários pares de contexto de criptografia. O seguinte exemplo de instrução de política de chaves usa um efeito Deny para proibir a entidade principal de usar as chaves do KMS se o contexto de criptografia na solicitação incluir um par Stage=Restricted ou Stage=Production.

Vários valores (Restricted e Production) para a mesma chave (kms:EncryptionContext:Stage) são implicitamente conectados por OR. Para obter mais detalhes, consulte [Lógica de avaliação para condições com várias chaves ou valores](#), no Manual do usuário do IAM.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": [
        "Restricted",
        "Production"
      ]
    }
  }
}
```

Diferenciação de letras maiúsculas e minúsculas da condição de contexto de criptografia

O contexto de criptografia especificado em uma operação de criptografia deve ser uma correspondência exata, que diferencia maiúsculas de minúsculas para o contexto de criptografia especificado na operação de criptografia. Somente a ordem dos pares em um contexto de criptografia com vários pares pode variar.

No entanto, em condições de política, a chave de condição não diferencia maiúsculas de minúsculas. A diferenciação de maiúsculas e minúsculas do valor da condição é determinada pelo [operador de condição de política](#) que você usa, como `StringEquals` ou `StringEqualsIgnoreCase`.

Dessa forma, a chave de condição, que consiste do prefixo `kms:EncryptionContext:` e da substituição *context-key*, não diferencia maiúsculas de minúsculas. Uma política que usa essa condição não verifica a capitalização dos elementos da chave de condição. A diferenciação de maiúsculas e minúsculas do valor, ou seja, a substituição *context-value*, é determinada pelo operador de condição de política.

Por exemplo, a declaração de política a seguir permite a operação quando o contexto de criptografia inclui uma chave `Appname`, independentemente da sua capitalização. A condição `StringEquals` exige que `ExampleApp` seja capitalizado da maneira como foi especificado.

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

Para exigir uma chave de contexto de criptografia com distinção entre maiúsculas e minúsculas, use a condição [kms: EncryptionContextKeys policy com um operador](#) de condição que diferencia maiúsculas de minúsculas, como `StringEqualsIgnoreCase`. Nesta condição de política, como a chave de contexto de criptografia é o valor da condição de política, o recurso é determinado pelo operador da condição.

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

Para exigir uma avaliação com distinção entre maiúsculas e minúsculas da chave de contexto e do valor da criptografia, use as condições de política de chave de contexto *kms:EncryptionContextKeys* e *kms:EncryptionContext*: juntas na mesma declaração de política. O operador de condição sensível a maiúsculas e minúsculas (como *StringEquals*) sempre aplica-se ao valor da condição. A chave de contexto de criptografia (como *AppName*) é o valor da condição *kms:EncryptionContextKeys*. O valor do contexto de criptografia (como *ExampleApp*) é o valor da condição *kms:EncryptionContext*: *context-key*.

Por exemplo, na instrução de política de chaves a seguir, como o operador *StringEquals* diferencia maiúsculas de minúsculas, a chave de contexto de criptografia e o valor de contexto de criptografia diferenciam maiúsculas de minúsculas.

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

```
}
```

Usar variáveis em uma condição de contexto de criptografia

A chave e o valor em um par de contexto de criptografia devem ser strings literais simples. Não podem ser números inteiros, objetos, ou qualquer tipo que não esteja totalmente resolvido. Se você usar um tipo diferente, como um número inteiro ou flutuante, AWS KMS interprete-o como uma string literal.

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

No entanto, o valor da chave de condição `kms:EncryptionContext:context-key` pode ser uma [variável de política do IAM](#). Essas variáveis de política são resolvidas em runtime com base nos valores da solicitação. Por exemplo, o `aws:CurrentTime` é resolvido como a hora da solicitação e `aws:username` é resolvido como o nome amigável do chamador.

É possível usar essas variáveis de política para criar uma declaração de política com uma condição que requer informações muito específicas em um contexto de criptografia, como o nome de usuário do chamador. Como contém uma variável, é possível usar a mesma declaração de política para todos os usuários que podem assumir a função. Não é necessário escrever uma declaração de política diferente para cada usuário.

Considere uma situação em que você deseja que todos os usuários que podem assumir uma função usem a mesma chave do KMS para criptografar e descriptografar seus dados. No entanto, você deseja permitir que eles descriptografem somente os dados que criptografaram. Comece exigindo que cada solicitação AWS KMS inclua um contexto de criptografia em que a chave esteja `user` e o valor seja o nome de AWS usuário do chamador, como o seguinte.

```
"encryptionContext": {  
  "user": "bob"  
}
```

Para impor esse requisito, é possível usar uma declaração de política como a do exemplo a seguir. Essa instrução de política concede à função `TestTeam` permissão para criptografar e descriptografar dados com a chave do KMS. No entanto, a permissão será válida somente quando o contexto de criptografia na solicitação incluir um par `"user": "<username>"`. Para representar o nome do usuário, a condição usa a variável de política [aws:username](#).

Quando a solicitação for avaliada, o nome de usuário do chamador substituirá a variável na condição. Dessa forma, a condição requer um contexto de criptografia de "user": "bob" para "bob" e "user": "alice" para "alice".

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:user": "${aws:username}"
    }
  }
}
```

É possível usar uma variável de política do IAM somente no valor do par de chaves da condição `kms:EncryptionContext:context-key`. Não é possível usar uma variável na chave.

Também não é possível usar [chaves de contexto específicas do provedor](#) em variáveis. Essas chaves de contexto identificam de forma exclusiva os usuários que se conectaram AWS usando a federação de identidade da web.

Como todas as variáveis, essas variáveis podem ser usadas apenas na condição de política `kms:EncryptionContext:context-key`, não no contexto de criptografia real. E elas podem ser usadas apenas no valor da condição, não na chave.

Por exemplo, a declaração de política de chaves a seguir é semelhante à anterior. No entanto, a condição requer um contexto de criptografia em que a chave é sub e o valor identifica exclusivamente um usuário conectado a um grupo de usuários do Amazon Cognito. Para obter detalhes sobre como identificar usuários e funções no Amazon Cognito, consulte [Funções do IAM](#), no [Guia do desenvolvedor do 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}"
    }
  }
}
```

Consulte também

- [the section called “kms:EncryptionContextKeys”](#)
- [the section called “kms:GrantConstraintType”](#)

kms:EncryptionContextKeys

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Encry ptionCont extKeys	String (lista)	Vários valores	CreateGrant Decrypt Encrypt Generated ataKey Generated ataKeyPair Generated ataKeyPai	Políticas de chaves e políticas do IAM

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
			rWithoutP laintext Generated ataKeyWit houtPlain text ReEncrypt RetireGrant	

É possível usar a chave de condição `kms:EncryptionContextKeys` para controlar o acesso a uma [chave do KMS de criptografia simétrica](#) com base no [contexto de criptografia](#) em uma solicitação de uma operação criptográfica. Use essa chave de condição para avaliar apenas a chave em cada par de contexto de criptografia. Para avaliar tanto a chave quanto o valor no contexto de criptografia, use a chave de condição `kms:EncryptionContext:context-key`.

Não é possível especificar um contexto de criptografia em uma operação criptográfica com uma [chave do KMS assimétrica](#) ou uma [chave do KMS de Hash-based message authentication code \(HMAC – Código de autenticação de mensagem por hash\)](#). Algoritmos assimétricos e algoritmos de Message authentication code (MAC – Código de autenticação de mensagem) não são compatíveis com um contexto de criptografia.

Note

Os valores da chave de condição, incluindo uma chave de contexto de criptografia, devem estar em conformidade com as regras de caracteres e codificação das políticas de AWS KMS chaves. Talvez você não consiga usar essa chave de condição para expressar todas as chaves válidas de contexto de criptografia. Para obter detalhes sobre as regras de documento de política de chaves, consulte [Formato de política de chaves](#). Para obter detalhes sobre as regras de documento de política do IAM, consulte [Requisitos de nome do IAM](#) no Guia do usuário do IAM.

Esta é uma [chave de condição de vários valores](#). Você pode especificar vários pares de contexto de criptografia em cada solicitação de API. `kms:EncryptionContextKeys` compara as chaves de contexto de criptografia na solicitação com o conjunto de chaves de contexto de criptografia na política. Para determinar como esses conjuntos são comparados, você deve fornecer um operador de conjunto `ForAnyValue` ou `ForAllValues` na condição de política. Para detalhes sobre os operadores de conjunto, consulte [Usar várias chaves e valores](#), no Manual do usuário do IAM.

- `ForAnyValue`: pelo menos uma chave de contexto de criptografia na solicitação deve corresponder a uma chave de contexto de criptografia na condição da política. Outras chaves de contexto de criptografia são permitidas. Se a solicitação não tiver um contexto de criptografia, a condição não será atendida.
- `ForAllValues`: cada chave de contexto de criptografia na solicitação deve corresponder a uma chave de contexto de criptografia na condição da política. Esse operador de conjunto limita as chaves de contexto de criptografia àquelas na condição de política. Ele não requer uma chave de contexto de criptografia, mas proíbe chaves de contexto de criptografia não especificadas.

O exemplo a seguir de instrução de política de chaves usa a chave de condição `kms:EncryptionContextKeys` com o operador de conjunto `ForAnyValue`. Essa instrução de política permite o uso de uma chave do KMS para as operações especificadas, mas somente quando pelo menos um dos pares de contextos de criptografia na solicitação inclui a chave `AppName`, independentemente do seu valor.

Por exemplo, essa instrução de política de chaves permite uma solicitação `GenerateDataKey` com dois pares de contextos de criptografia, `AppName=Helper` e `Project=Alpha`, pois o primeiro par de contextos de criptografia atende à condição. Uma solicitação com apenas `Project=Alpha` ou sem um contexto de criptografia falharia.

Como a operação de [StringEquals](#) condição diferencia maiúsculas de minúsculas, essa declaração de política exige a ortografia e as maiúsculas e minúsculas da chave de contexto de criptografia. No entanto, você pode usar um operador de condição que ignora a capitalização da chave, como `StringEqualsIgnoreCase`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
```

```
"kms:Encrypt",
"kms:GenerateDataKey*"
],
"Resource": "*",
"Condition": {
  "ForAnyValue:StringEquals": {
    "kms:EncryptionContextKeys": "AppName"
  }
}
}
```

Você também pode usar a chave de condição `kms:EncryptionContextKeys` para exigir um contexto de criptografia (qualquer um) em operações de criptografia que usam a chave do KMS.

A instrução de política de exemplo a seguir usa a chave de condição `kms:EncryptionContextKeys` com o [Operador de condição nula](#) para permitir o acesso a uma chave do KMS apenas quando o contexto de criptografia na solicitação da API não é nulo. Essa condição não confere as chaves ou os valores do contexto de criptografia. Ela só verifica se o contexto de criptografia existe.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContextKeys": false
    }
  }
}
```

Consulte também

- [kms:EncryptionContext: chave de contexto](#)
- [kms:GrantConstraintType](#)

kms:ExpirationModel

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:ExpirationModel	String	Valor único	ImportKeyMaterial	Políticas de chaves e políticas do IAM

A chave de kms:ExpirationModel condição controla o acesso à [ImportKeyMaterial](#) operação com base no valor do [ExpirationModel](#) parâmetro na solicitação.

ExpirationModel é um parâmetro opcional que determina se o material de chave importada expira. Os valores válidos são KEY_MATERIAL_EXPIRES e KEY_MATERIAL_DOES_NOT_EXPIRE. KEY_MATERIAL_EXPIRES é o valor padrão.

A data e a hora de expiração são determinadas pelo valor do [ValidTo](#) parâmetro. O parâmetro ValidTo só não será necessário se o valor do parâmetro ExpirationModel for KEY_MATERIAL_DOES_NOT_EXPIRE. Você também pode usar a chave de ValidTo condição [kms:](#) para exigir uma data de expiração específica como condição de acesso.

A instrução de política exemplificada a seguir usa a chave de condição kms:ExpirationModel para permitir que os usuários importem o material de chave para uma chave do KMS somente quando a solicitação incluir o parâmetro ExpirationModel e seu valor for 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"
    }
  }
}
```

```
}

```

Também é possível usar a chave de condição `kms:ExpirationModel` para permitir que os usuários importem o material de chave somente quando este expirar. A instrução de política de chave exemplificada a seguir usa a chave de condição `kms:ExpirationModel` com o [operador de condição Null](#) para permitir que os usuários importem o material de chave somente quando a solicitação não tiver um parâmetro `ExpirationModel`. O valor padrão para `ExpirationModel` é `KEY_MATERIAL_EXPIRES`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:ExpirationModel": true
    }
  }
}
```

Consulte também

- [kms:ValidTo](#)
- [kms:WrappingAlgorithm](#)
- [kms:WrappingKeySpec](#)

kms:GrantConstraintType

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Grant Constrain tType	String	Valor único	CreateGrant RetireGrant	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para controlar o acesso à [CreateGrant](#) operação com base no tipo de [restrição de concessão](#) na solicitação.

Ao criar uma concessão, você pode especificar uma restrição de concessão para permitir as operações permitidas pela concessão apenas quando um [contexto de criptografia](#) específico existir. A restrição de concessão pode ser de dois tipos: `EncryptionContextEquals` ou `EncryptionContextSubset`. Você pode usar esta chave de condição para verificar se a solicitação contém um tipo ou outro.

 Important

Não inclua informações confidenciais ou sigilosas nesse campo. Esse campo pode ser exibido em texto simples em CloudTrail registros e outras saídas.

A instrução de política de chaves exemplificada a seguir usa a chave de condição `kms:GrantConstraintType` para permitir que os usuários criem concessões somente quando a solicitação incluir uma restrição de concessão `EncryptionContextEquals`. O exemplo mostra uma declaração de política em uma política de chaves.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GrantConstraintType": "EncryptionContextEquals"
    }
  }
}
```

Consulte também

- [kms:EncryptionContext: chave de contexto](#)
- [kms:EncryptionContextKeys](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GrantOperations](#)

- [kms:GranteePrincipal](#)
- [kms:RetiringPrincipal](#)

kms:GrantIsForAWSResource

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:GrantIsForAWSResource	Booleano	Valor único	CreateGrant ListGrants RevokeGrant	Políticas de chaves e políticas do IAM

Permite ou nega permissão para as [RevokeGrant](#) operações [CreateGrantListGrants](#), ou somente quando um [AWS serviço integrado AWS KMS](#) chama a operação em nome do usuário. Essa condição de política não permite que o usuário chame essas operações de concessão diretamente.

A declaração de política de chaves demonstrativa a seguir usa a chave de condição kms:GrantIsForAWSResource. Ele permite que AWS serviços integrados AWS KMS, como o Amazon EBS, criem concessões nessa chave KMS em nome do principal especificado.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
}
```

Consulte também

- [kms:GrantConstraintType](#)
- [kms:GrantOperations](#)
- [kms:GranteePrincipal](#)
- [kms:RetiringPrincipal](#)

kms:GrantOperations

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Grant Operations	String	Vários valores	CreateGrant	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para controlar o acesso à [CreateGrant](#) operação com base nas [operações de concessão](#) na solicitação. Por exemplo, é possível permitir que os usuários criem concessões que delegam permissão para criptografar, mas não para descriptografar. Para obter mais informações sobre concessões, consulte [Usar concessões](#).

Esta é uma [chave de condição de vários valores](#). O kms:GrantOperations compara o conjunto de operações de concessão na solicitação CreateGrant para o conjunto de operações de concessão na política. Para determinar como esses conjuntos são comparados, você deve fornecer um operador de conjunto ForAnyValue ou ForAllValues na condição de política. Para detalhes sobre os operadores de conjunto, consulte [Usar várias chaves e valores](#), no Manual do usuário do IAM.

- ForAnyValue: pelo menos uma operação de concessão na solicitação deve corresponder a uma das operações de concessão na condição da política. Outras operações de concessão são permitidas.
- ForAllValues: cada operação de concessão na solicitação deve corresponder a uma operação de concessão na condição da política. Este operador de conjunto limita as operações de concessão àquelas especificadas na condição da política. Ele não requer uma operação de concessão, mas proíbe operações de concessão não especificadas.

ForAllValues também retorna verdadeiro quando não há operações de concessão na solicitação, mas CreateGrant não permite isso. Se o parâmetro Operations ausente ou tiver um valor nulo, o parâmetro CreateGrant falhará na solicitação.

A instrução de política de chaves exemplificada a seguir usa a chave de condição kms:GrantOperations para criar concessões somente quando as operações de concessões são Encrypt e/ou ReEncryptTo. Se a concessão incluir quaisquer outras operações, a solicitação CreateGrant falhará.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Encrypt",
        "ReEncryptTo"
      ]
    }
  }
}
```

Se você alterar o operador de conjunto na condição de política para ForAnyValue, a instrução de política exigirá que pelo menos uma das operações de concessão na concessão seja Encrypt ou ReEncryptTo, mas permitirá outras operações de concessão, como Decrypt ou ReEncryptFrom.

Consulte também

- [kms:GrantConstraintType](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GranteePrincipal](#)
- [kms:RetiringPrincipal](#)

kms:GranteePrincipal

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:GranteePrincipal	String	Valor único	CreateGrant	Políticas do IAM e de chaves

Você pode usar essa chave de condição para controlar o acesso à [CreateGrant](#) operação com base no valor do [GranteePrincipal](#) parâmetro na solicitação. Por exemplo, é possível criar concessões para usar uma chave do KMS somente quando a entidade principal favorecida na solicitação CreateGrant corresponder à entidade principal especificada na instrução de condição.

Para especificar a entidade principal beneficiada, use o nome do recurso da Amazon (ARN) de uma entidade principal da AWS. Os principais válidos incluem usuários do IAM Contas da AWS, funções do IAM, usuários federados e usuários com funções assumidas. Para obter ajuda com a sintaxe do ARN para um diretor, consulte [IAM ARNs no Guia do](#) usuário do IAM.

A instrução de política de chaves exemplificada a seguir usa a chave de condição kms:GranteePrincipal para criar concessões para uma chave do KMS somente quando a entidade principal favorecida na concessão for 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"
    }
  }
}
```

Consulte também

- [kms:GrantConstraintType](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GrantOperations](#)
- [kms:RetiringPrincipal](#)

kms:KeyAgreementAlgorithm

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:KeyAgreementAlgorithm	String	Valor único	DeriveSharedSecret	Políticas de chaves e políticas do IAM

Você pode usar a chave de kms:KeyAgreementAlgorithm condição para controlar o acesso à [DeriveSharedSecret](#) operação com base no valor do KeyAgreementAlgorithm parâmetro na solicitação. O único valor válido para KeyAgreementAlgorithm é ECDH.

Por exemplo, a declaração de política chave a seguir usa a chave de kms:KeyAgreementAlgorithm condição para negar todo o acesso, a DeriveSharedSecret menos que KeyAgreementAlgorithm seja ECDH.

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

Consulte também

- [the section called “kms:KeyUsage”](#)

kms:KeyOrigin

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:KeyOrigin	String	Valor único	CreateKey Operações de recursos de chaves do KMS	Políticas do IAM Políticas de chaves e políticas do IAM

A chave de condição `kms:KeyOrigin` controla o acesso às operações com base no valor da propriedade `Origin` da chave do KMS que é criada pela operação ou usada nela. Ela funciona como uma condição de recurso ou de solicitação.

Você pode usar essa chave de condição para controlar o acesso à [CreateKey](#) operação com base no valor do parâmetro [Origin](#) na solicitação. Os valores válidos para `Origin` são `AWS_KMS`, `AWS_CLOUDHSM` e `EXTERNAL`.

Por exemplo, você pode criar uma chave KMS somente quando o material da chave é gerado em AWS KMS (`AWS_KMS`), somente quando o material da chave é gerado em um AWS CloudHSM cluster associado a um [armazenamento de chaves personalizado](#) (`AWS_CLOUDHSM`) ou somente quando o [material da chave é importado](#) de uma fonte externa (`EXTERNAL`).

O exemplo de declaração de política de chaves a seguir usa a chave de `kms:KeyOrigin` condição para criar uma chave KMS somente quando AWS KMS cria o material da chave.

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"
      }
    }
  }
]
}

```

Também é possível usar a chave de condição `kms:KeyOrigin` para controlar o acesso a operações que usam ou gerenciam uma chave do KMS com base na propriedade `Origin` da chave do KMS usada para a operação. A operação deve ser uma operação de recurso de chave do KMS, ou seja, uma operação autorizada para uma chave do KMS específica. Para identificar as operações de

recursos de chaves do KMS, na [Tabela de ações e recursos](#), procure um valor de KMS key na coluna Resources para a operação.

Por exemplo, a política do IAM a seguir permite que as entidades principais realizem as operações de recursos de chave do KMS especificadas, mas somente com as chaves do KMS na conta que foram criadas em um armazenamento de chaves personalizado.

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

Consulte também

- [kms:BypassPolicyLockoutSafetyCheck](#)
- [kms:KeySpec](#)
- [kms:KeyUsage](#)

kms:KeySpec

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:KeySpec	String	Valor único	CreateKey	Políticas do IAM

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
			Operações de recursos de chaves do KMS	Políticas de chaves e políticas do IAM

A chave de condição `kms:KeySpec` controla o acesso às operações com base no valor da propriedade `KeySpec` da chave do KMS que é criada pela operação ou usada nela.

Você pode usar essa chave de condição em uma política do IAM para controlar o acesso à [CreateKey](#) operação com base no valor do [KeySpec](#) parâmetro em uma `CreateKey` solicitação. Por exemplo, é possível usar essa condição para permitir que os usuários criem somente chaves do KMS de criptografia simétrica ou somente chaves do KMS de HMAC.

O seguinte exemplo de instrução de política do IAM usa a chave de condição `kms:KeySpec` para permitir que as entidades principais criem apenas chaves do KMS assimétricas RSA. A permissão só é válida quando o `KeySpec` na solicitação começa com `RSA_`.

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

Também é possível usar a chave de condição `kms:KeySpec` para controlar o acesso a operações que usam ou gerenciam uma chave do KMS com base na propriedade `KeySpec` da chave do KMS usada para a operação. A operação deve ser uma operação de recurso de chave do KMS, ou seja, uma operação autorizada para uma chave do KMS específica. Para identificar as operações de recursos de chaves do KMS, na [Tabela de ações e recursos](#), procure um valor de KMS key na coluna `Resources` para a operação.

Por exemplo, a seguinte política do IAM permite que as entidades principais executem as operações especificadas de recursos de chave do KMS, mas somente com chaves do KMS de criptografia simétrica na conta.

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

Consulte também

- [kms:BypassPolicyLockoutSafetyCheck](#)
- [kms: CustomerMasterKeySpec \(obsoleto\)](#)
- [kms:DataKeyPairSpec](#)
- [kms:KeyOrigin](#)
- [kms:KeyUsage](#)

kms:KeyUsage

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:KeyUsage	String	Valor único	CreateKey	Políticas do IAM
				Políticas de chaves e políticas do IAM

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
			Operações de recursos de chaves do KMS	

A chave de condição `kms:KeyUsage` controla o acesso às operações com base no valor da propriedade `KeyUsage` da chave do KMS que é criada pela operação ou usada nela.

Você pode usar essa chave de condição para controlar o acesso à [CreateKey](#) operação com base no valor do [KeyUsage](#) parâmetro na solicitação. Os valores válidos para `KeyUsage` são `ENCRYPT_DECRYPT`, `SIGN_VERIFY`, `GENERATE_VERIFY_MAC` e `KEY_AGREEMENT`.

Por exemplo, será possível criar uma chave do KMS somente quando `KeyUsage` for `ENCRYPT_DECRYPT` ou negar a permissão de um usuário quando `KeyUsage` for `SIGN_VERIFY`.

A instrução de política do IAM exemplificada a seguir usa a chave de condição `kms:KeyUsage` para criar uma chave do KMS somente quando `KeyUsage` for `ENCRYPT_DECRYPT`.

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

Também é possível usar a chave de condição `kms:KeyUsage` para controlar o acesso a operações que usam ou gerenciam uma chave do KMS com base na propriedade `KeyUsage` da chave do KMS na operação. A operação deve ser uma operação de recurso de chave do KMS, ou seja, uma operação autorizada para uma chave do KMS específica. Para identificar as operações de recursos de chaves do KMS, na [Tabela de ações e recursos](#), procure um valor de KMS key na coluna `Resources` para a operação.

Por exemplo, a política do IAM a seguir permite que as entidades principais realizem as operações de recursos de chaves do KMS especificadas, mas somente com as chaves do KMS na conta que são usadas para assinatura e verificação.

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

Consulte também

- [kms:BypassPolicyLockoutSafetyCheck](#)
- [kms: CustomerMasterKeyUsage \(obsoleto\)](#)
- [kms:KeyOrigin](#)
- [kms:KeySpec](#)

kms:MacAlgorithm

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:MacAl gorithm	String	Valor único	GenerateMac VerifyMac	Políticas de chaves e políticas do IAM

Você pode usar a chave de `kms:MacAlgorithm` condição para controlar o acesso às [VerifyMac](#) operações [GenerateMac](#) com base no valor do `MacAlgorithm` parâmetro na solicitação.

O seguinte exemplo de política de chaves permite que usuários capazes de assumir a função `testers` só usem a chave do KMS de HMAC para gerar e verificar etiquetas HMAC quando o algoritmo de MAC na solicitação for `HMAC_SHA_384` ou `HMAC_SHA_512`. Essa política usa duas instruções distintas de política, cada uma com sua própria condição. Se você especificar mais de um algoritmo de MAC em uma única instrução de condição, a condição exigirá ambos os algoritmos em vez de um ou outro.

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"
    }
  }
}
```

Consulte também

- [the section called “kms:EncryptionAlgorithm”](#)
- [kms:SigningAlgorithm](#)

kms:MessageType

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Messa geType	String	Valor único	Sign Verify	Políticas de chaves e políticas do IAM

A chave de condição `kms:MessageType` controla o acesso às operações [Sign](#) e [Verify](#) com base no valor do parâmetro `MessageType` na solicitação. Os valores válidos para `MessageType` são `RAW` e `DIGEST`.

Por exemplo, a instrução de política de chaves a seguir usa a chave de condição `kms:MessageType` para usar uma chave do KMS assimétrica para assinar uma mensagem, mas não um resumo da mensagem.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:Sign",
  "Resource": "*",
}
```

```

"Condition": {
  "StringEquals": {
    "kms:MessageType": "RAW"
  }
}
}

```

Consulte também

- [the section called “kms:SigningAlgorithm”](#)

kms:MultiRegion

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:MultiRegion	Booleano	Valor único	CreateKey Operações de recursos de chaves do KMS	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para permitir operações somente em chaves de região única ou somente em [chaves de várias regiões](#). A chave de kms:MultiRegion condição controla o acesso às AWS KMS operações nas chaves KMS e à [CreateKey](#) operação com base no valor da MultiRegion propriedade da chave KMS. Os valores válidos são true (várias regiões) e false (região única). Todas as chaves do KMS têm uma propriedade MultiRegion.

Por exemplo, a seguinte instrução de política do IAM usa a chave de condição kms:MultiRegion para permitir que as entidades principais criem apenas chaves de região única.

```

{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:MultiRegion": false
    }
  }
}

```

```

    }
  }
}
```

kms:MultiRegionKeyType

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:MultiRegionKeyType	String	Valor único	CreateKey Operações de recursos de chaves do KMS	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para permitir operações somente em [chaves primárias de várias regiões](#) ou somente em [chaves de réplica de várias regiões](#). A chave de kms:MultiRegionKeyType condição controla o acesso às AWS KMS operações nas chaves KMS e a [CreateKey](#) operação com base na MultiRegionKeyType propriedade da chave KMS. Os valores válidos são PRIMARY e REPLICA. Somente chaves de várias regiões têm uma propriedade MultiRegionKeyType.

Normalmente, você pode usar a chave de condição kms:MultiRegionKeyType em uma política do IAM para controlar o acesso a várias chaves do KMS. No entanto, como uma determinada chave de várias regiões pode mudar para primária ou réplica, convém usar essa condição em uma política de chaves para permitir uma operação somente quando a chave de várias regiões específica for uma chave primária ou de réplica.

Por exemplo, a instrução de política do IAM a seguir usa a chave de condição kms:MultiRegionKeyType para permitir que as entidades principais programem e cancelem a exclusão de chaves somente em chaves de réplica de várias regiões na Conta da AWS especificada.

```

{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ]
}
```

```

],
"Resource": "arn:aws:kms:*:111122223333:key/*",
"Condition": {
  "StringEquals": {
    "kms:MultiRegionKeyType": "REPLICA"
  }
}
}
}

```

Para permitir ou negar acesso a todas as chaves de várias regiões, você pode usar ambos os valores ou um valor nulo com `kms:MultiRegionKeyType`. No entanto, a chave de MultiRegion condição [kms:](#) é recomendada para essa finalidade.

kms:PrimaryRegion

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
<code>kms:PrimaryRegion</code>	String (lista)	Valor único	<code>UpdatePrimaryRegion</code>	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para limitar as regiões de destino em uma [UpdatePrimaryRegion](#) operação. São elas Regiões da AWS que podem hospedar suas chaves primárias multirregionais.

A chave de `kms:PrimaryRegion` condição controla o acesso à [UpdatePrimaryRegion](#) operação com base no valor do `PrimaryRegion` parâmetro. O `PrimaryRegion` parâmetro especifica a [chave Região da AWS de réplica multirregional](#) que está sendo promovida para primária. O valor da condição é um ou mais Região da AWS nomes, como `us-east-1` ou `ap-southeast-2`, ou padrões de nome de região, como `eu-*`

Por exemplo, a instrução de política de chaves a seguir usa a chave de condição `kms:PrimaryRegion` para permitir que as entidades principais atualizem a região primária de uma chave de várias regiões para uma das quatro regiões especificadas.

```
{
```



```
"Effect": "Allow",
"Action": "kms:UpdatePrimaryRegion",
"Principal": {
  "AWS": "arn:aws:iam::111122223333:role/Developer"
},
"Resource": "*",
"Condition": {
  "StringEquals": {
    "kms:PrimaryRegion": [
      "us-east-1",
      "us-west-2",
      "eu-west-3",
      "ap-southeast-2"
    ]
  }
}
```

kms:ReEncryptOnSameKey

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:ReEncryptOnSameKey	Booleano	Valor único	ReEncrypt	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para controlar o acesso à [ReEncrypt](#) operação com base no fato de a solicitação especificar uma chave KMS de destino que seja a mesma usada para a criptografia original.

Por exemplo, a instrução de política de chaves a seguir usa a chave de condição `kms:ReEncryptOnSameKey` para criptografar novamente somente quando a chave do KMS de destino é semelhante à usada para a criptografia original.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
```

```

    },
    "Action": "kms:ReEncrypt*",
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:ReEncryptOnSameKey": true
        }
    }
}

```

kms:RequestAlias

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Reque stAlias	String (lista)	Valor único	Operações criptográficas DescribeKey GetPublicKey	Políticas de chaves e políticas do IAM

É possível usar essa chave de condição para permitir uma operação somente quando a solicitação usa um alias específico para identificar a chave do KMS. A chave de condição `kms:RequestAlias` controla o acesso a uma chave do KMS usada em uma operação criptográfica, `GetPublicKey` ou `DescribeKey` com base no [alias](#) que identifica essa chave do KMS na solicitação. (Essa condição de política não tem efeito na [GenerateRandom](#) operação porque a operação não usa uma chave ou alias KMS.)

Essa condição oferece suporte [ao controle de acesso baseado em atributos](#) (ABAC) em AWS KMS, que permite controlar o acesso às chaves KMS com base nas tags e aliases de uma chave KMS. Você pode usar tags e aliases para conceder ou negar acesso a uma chave do KMS sem alterar políticas ou permissões. Para obter detalhes, consulte [ABAC para AWS KMS](#).

Para especificar o alias nessa condição de política, use um [nome de alias](#), como `alias/project-alpha`, ou um padrão de nome de alias, como `alias/*test*`. Não é possível especificar um [ARN de alias](#) no valor dessa chave de condição.

Para satisfazer essa condição, o valor do parâmetro `KeyId` na solicitação deve ser um nome de alias correspondente ou um ARN de alias. Se a solicitação usar um [identificador de chave](#) diferente, ela não atenderá à condição, mesmo que identifique a mesma chave do KMS.

Por exemplo, a declaração de política chave a seguir permite que o diretor chame a [GenerateDataKey](#) operação na chave KMS. No entanto, isso é permitido somente quando o valor do parâmetro `KeyId` na solicitação é `alias/finance-key` ou um ARN de alias com esse nome de alias, como `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"
    }
  }
}
```

Você não pode usar essa chave de condição para controlar o acesso às operações de alias, como [CreateAlias](#) ou [DeleteAlias](#). Para obter informações sobre como controlar o acesso a todas as operações de alias, consulte [Controlar o acesso a aliases](#).

kms:ResourceAliases

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
<code>kms:ResourceAliases</code>	String (lista)	Vários valores	Operações de recursos de chaves do KMS	Somente políticas do IAM

Use essa chave de condição para controlar o acesso a uma chave do KMS com base nos [alias](#) que estão associados à chave do KMS. A operação deve ser uma operação de recurso de chave do KMS, ou seja, uma operação autorizada para uma chave do KMS específica. Para identificar as operações de recursos de chaves do KMS, na [Tabela de ações e recursos](#), procure um valor de KMS key na coluna Resources para a operação.

Essa condição oferece suporte ao controle de acesso baseado em atributos (ABAC) no AWS KMS. Com o ABAC, você pode controlar o acesso a chaves do KMS com base nas etiquetas atribuídas a uma chave do KMS e nos aliases associados a uma chave do KMS. Você pode usar tags e aliases para conceder ou negar acesso a uma chave do KMS sem alterar políticas ou permissões. Para obter detalhes, consulte [ABAC para AWS KMS](#).

Um alias deve ser exclusivo em uma região Conta da AWS e, mas essa condição permite controlar o acesso a várias chaves KMS na mesma região (usando o operador de StringLike comparação) ou a várias chaves KMS em diferentes contas Regiões da AWS .

 Note

A ResourceAliases condição [kms:](#) é efetiva somente quando a chave KMS está em conformidade com os [aliases por](#) cota de chave KMS. Se uma chave do KMS exceder essa cota, as entidades principais autorizadas a usar essa chave pela condição `kms:ResourceAliases` terão acesso negado a ela.

Para especificar o alias nessa condição de política, use um [nome de alias](#), como `alias/project-alpha`, ou um padrão de nome de alias, como `alias/*test*`. Não é possível especificar um [ARN de alias](#) no valor dessa chave de condição. Para atender à condição, a chave do KMS usada na operação deve ter o alias especificado. Não importa se ou como a chave do KMS é identificada na solicitação para a operação.

Esta é uma chave de condição de vários valores que compara o conjunto de aliases associados a uma chave do KMS com o conjunto de aliases na política. Para determinar como esses conjuntos são comparados, você deve fornecer um operador de conjunto `ForAnyValue` ou `ForAllValues` na condição de política. Para detalhes sobre os operadores de conjunto, consulte [Usar várias chaves e valores](#), no Manual do usuário do IAM.

- `ForAnyValue`: pelo menos um alias associado à chave KMS deve corresponder a um alias na condição da política. Outros aliases são permitidos. Se a chave do KMS não tiver alias, a condição não será atendida.

- **ForAllValues**: cada alias associado à chave KMS deve corresponder a um alias na política. Esse operador de conjunto limita os aliases associados à chave do KMS àqueles na condição da política. Ele não requer aliases, mas proíbe aliases não especificados.

Por exemplo, a declaração de política do IAM a seguir permite que o diretor chame a [GenerateDataKey](#) operação em qualquer chave KMS especificada Conta da AWS associada ao `finance-key` alias. (As políticas de chaves das chaves do KMS afetadas também devem permitir que a conta da entidade principal as use para essa operação.) Para indicar que a condição é atendida quando um dos muitos aliases que podem estar associados à chave do KMS é `alias/finance-key`, a condição usa o operador de conjunto `ForAnyValue`.

Como a condição `kms:ResourceAliases` é baseada no recurso, não na solicitação, uma chamada para `GenerateDataKey` é bem-sucedida para qualquer chave do KMS associada ao alias `finance-key`, mesmo que a solicitação use um [ID de chave](#) ou um [ARN de chave](#) para identificar a chave do 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"
    }
  }
}
```

O seguinte exemplo de instrução de política do IAM permite que a entidade principal habilite e desabilite chaves do KMS, mas somente quando todos os aliases das chaves do KMS incluem `Test`. Essa instrução de política usa duas condições. A condição com o operador de conjunto `ForAllValues` requer que todos os aliases associados à chave do KMS incluam `Test`. A condição com o operador de conjunto `ForAnyValue` requer que a chave do KMS tenha pelo menos um alias com `Test`. Sem a condição `ForAnyValue`, essa instrução de política teria permitido que a entidade principal usasse chaves do KMS sem aliases.

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": [
    "kms:EnableKey",
    "kms:DisableKey"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "ForAllValues:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    },
    "ForAnyValue:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    }
  }
}
```

kms:ReplicaRegion

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Repli caRegion	String (lista)	Valor único	Replicate Key	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para limitar a Regiões da AWS possibilidade de um principal replicar uma chave [multirregional](#). A chave de kms:ReplicaRegion condição controla o acesso à [ReplicateKey](#) operação com base no valor do [ReplicaRegion](#) parâmetro na solicitação. Esse parâmetro especifica a propriedade Região da AWS da nova [chave de réplica](#).

O valor da condição é um ou mais Região da AWS nomes, como us-east-1 ouap-southeast-2, ou padrões de nomes, comoeu-*. Para obter uma lista dos nomes desses Regiões da AWS AWS

KMS suportes, consulte [AWS Key Management Service endpoints e cotas](#) no. Referência geral da AWS

Por exemplo, a declaração de política de chaves a seguir usa a chave de kms:ReplicaRegion condição para permitir que os diretores chamem a [ReplicateKey](#) operação somente quando o valor do ReplicaRegion parâmetro for uma das regiões especificadas.

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

Essa chave de condição controla o acesso somente à [ReplicateKey](#) operação. Para controlar o acesso à [UpdatePrimaryRegion](#) operação, use a chave de PrimaryRegion condição [kms:](#).

kms:RetiringPrincipal

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Retir ingPrinci pal	String (lista)	Valor único	CreateGrant	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para controlar o acesso à [CreateGrant](#) operação com base no valor do [RetiringPrincipal](#) parâmetro na solicitação. Por exemplo, é possível criar concessões

para usar uma chave do KMS somente quando `RetiringPrincipal` na solicitação `CreateGrant` corresponder a `RetiringPrincipal` na instrução da condição.

Para especificar a entidade principal em processo de desativação, use o ARN de uma entidade principal da AWS. Os principais válidos incluem usuários do IAM, Contas da AWS, funções do IAM, usuários federados e usuários com funções assumidas. Para obter ajuda com a sintaxe do ARN para um diretor, consulte [IAM ARNs no Guia do](#) usuário do IAM.

O exemplo de declaração de política de chave a seguir permite que um usuário crie concessões para a chave do KMS. A chave de condição `kms:RetiringPrincipal` restringe a permissão a solicitações `CreateGrant` nas quais a entidade principal em processo de desativação na concessão é `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"
    }
  }
}
```

Consulte também

- [kms:GrantConstraintType](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GrantOperations](#)
- [kms:GranteePrincipal](#)

kms:RotationPeriodInDays

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:RotationPeriodInDays	Numérico	Valor único	EnableKeyRotation	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para limitar os valores que os diretores podem especificar no `RotationPeriodInDays` parâmetro de uma [EnableKeyRotation](#) solicitação.

`RotationPeriodInDays` Especifica o número de dias entre cada data de rotação automática da chave. AWS KMS permite especificar um período de rotação entre 90 e 2560 dias, mas você pode usar a chave de `kms:RotationPeriodInDays` condição para restringir ainda mais o período de rotação, como impor um período mínimo de rotação dentro do intervalo válido.

Por exemplo, a declaração de política de chave a seguir usa a chave de condição `kms:RotationPeriodInDays` para impedir que entidades principais habilitem a alternância de chave se o período de alternância for menor ou igual a 180 dias.

```
{
  "Effect": "Deny",
  "Action": "kms:EnableKeyRotation",
  "Principal": "*",
  "Resource": "*",
  "Condition": {
    "NumericLessThanEquals": {
      "kms:RotationPeriodInDays": "180"
    }
  }
}
```

kms:ScheduleKeyDeletionPendingWindowInDays

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:ScheduleKeyDeletionPendingWindowInDays	Numérico	Valor único	ScheduleKeyDeletion	Políticas de chaves e políticas do IAM

Você pode usar essa chave de condição para limitar os valores que os diretores podem especificar no PendingWindowInDays parâmetro de uma [ScheduleKeyDeletion](#) solicitação.

PendingWindowInDaysEspecifica o número de dias que AWS KMS serão esperados antes de excluir uma chave. AWS KMS permite especificar um período de espera entre 7 e 30 dias, mas você pode usar a chave de kms:ScheduleKeyDeletionPendingWindowInDays condição para restringir ainda mais o período de espera, como impor um período mínimo de espera dentro do intervalo válido.

Por exemplo, a declaração de política de chave a seguir usa a chave de condição kms:ScheduleKeyDeletionPendingWindowInDays para impedir que a entidade principal programe a exclusão da chave se o período de espera for menor ou igual a 21 dias.

```
{
  "Effect": "Deny",
  "Action": "kms:ScheduleKeyDeletion",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
    "NumericLessThanEquals" : {
      "kms:ScheduleKeyDeletionPendingWindowInDays" : "21"
    }
  }
}
```

kms:SigningAlgorithm

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:SigningAlgorithm	String	Valor único	Sign Verify	Políticas de chaves e políticas do IAM

Você pode usar a chave de kms:SigningAlgorithm condição para controlar o acesso às operações de [Assinar](#) e [Verificar](#) com base no valor do [SigningAlgorithm](#) parâmetro na solicitação. Essa chave de condição não tem efeito nas operações realizadas fora da AWS KMS, como a verificação de assinaturas com a chave pública em um par de chaves KMS assimétrico fora da AWS KMS.

O exemplo de política de chaves a seguir permite que os usuários que podem assumir a função testers usem a chave do KMS para assinar mensagens somente quando o algoritmo de assinatura usado para a solicitação é um algoritmo RSASSA_PSS, como RSASSA_PSS_SHA512.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/testers"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:SigningAlgorithm": "RSASSA_PSS*"
    }
  }
}
```

Consulte também

- [kms:EncryptionAlgorithm](#)
- [the section called “kms:MacAlgorithm”](#)

- [the section called “kms:MessageType”](#)

kms:ValidTo

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:ValidTo	Timestamp	Valor único	ImportKey Material	Políticas de chaves e políticas do IAM

A chave de kms:ValidTo condição controla o acesso à [ImportKeyMaterial](#) operação com base no valor do [ValidTo](#) parâmetro na solicitação, que determina quando o material da chave importada expira. O valor é expresso no [horário do Unix](#).

Por padrão, o parâmetro ValidTo é obrigatório em uma solicitação ImportKeyMaterial. No entanto, se o valor do [ExpirationModel](#) parâmetro for KEY_MATERIAL_DOES_NOT_EXPIRE, o ValidTo parâmetro é inválido. Você também pode usar a chave de ExpirationModel condição [kms:](#) para exigir o ExpirationModel parâmetro ou um valor de parâmetro específico.

A seguinte instrução de política de exemplo permite a um usuário importar material-chave em uma chave do KMS. A chave de condição kms:ValidTo limita a permissão para solicitações ImportKeyMaterial em que o valor ValidTo é menor que ou igual a 1546257599.0 (31 de dezembro de 2018 23:59:59).

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "NumericLessThanEquals": {
      "kms:ValidTo": "1546257599.0"
    }
  }
}
```

}

Consulte também

- [kms:ExpirationModel](#)
- [kms:WrappingAlgorithm](#)
- [kms:WrappingKeySpec](#)

kms:ViaService

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:ViaService	String	Valor único	Operações de recursos de chaves do KMS	Políticas de chaves e políticas do IAM

A chave de kms:ViaService condição limita o uso de uma chave KMS às solicitações Serviços da AWS especificadas. Essa chave de condição se aplica somente às [sessões de acesso direto](#). Você pode especificar um ou mais serviços em cada chave de condição kms:ViaService. A operação deve ser uma operação de recurso de chave do KMS, ou seja, uma operação autorizada para uma chave do KMS específica. Para identificar as operações de recursos de chaves do KMS, na [Tabela de ações e recursos](#), procure um valor de KMS key na coluna Resources para a operação.

Por exemplo, a seguinte instrução de política de chaves usa a chave de condição kms:ViaService para permitir que uma [chave gerenciada pelo cliente](#) seja usada para as ações especificadas somente quando a solicitação vem do Amazon EC2 ou do Amazon RDS na região Oeste dos EUA (Oregon) em nome de 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"
      ]
    }
  }
}

```

Também é possível usar uma chave de condição `kms:ViaService` para negar permissão para usar uma chave do KMS quando a solicitação é proveniente de serviços específicos. Por exemplo, a instrução de política a seguir usa uma chave de condição `kms:ViaService` para impedir que uma chave gerenciada pelo cliente seja usada para operações `Encrypt` quando a solicitação é proveniente do AWS Lambda em nome de `ExampleRole`.

```

{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "lambda.us-west-2.amazonaws.com"
      ]
    }
  }
}

```

 Important

Quando você usa a chave de condição `kms:ViaService`, o serviço faz a solicitação em nome de uma entidade principal na conta da Conta da AWS. Essas entidades principais deve ter as seguintes permissões:

- Permissão para usar a chave do KMS. A entidade principal precisa conceder essas permissões ao serviço integrado, de forma que o serviço possa usar a chave gerenciada pelo cliente em nome da entidade principal. Para obter mais informações, consulte [Usando AWS KMS criptografia com AWS serviços](#).
- Permissão para usar o serviço integrado. Para obter detalhes sobre como dar aos usuários acesso a um AWS serviço que se integra ao AWS KMS, consulte a documentação do serviço integrado.

Todas as [Chaves gerenciadas pela AWS](#) usam uma chave de condição `kms:ViaService` no seu documento de política de chaves. Essa condição permite que a chave do KMS seja usada apenas para solicitações provenientes do serviço que criou a chave do KMS. Para ver a política principal de um Chave gerenciada pela AWS, use a [GetKeyPolicy](#) operação.

A chave de condição `kms:ViaService` é válida no IAM e em declarações de políticas de chaves. Os serviços que você especificar devem ser [integrados ao AWS KMS](#) e compatíveis com a chave de condição `kms:ViaService`.

Serviços que oferecem suporte à chave de condição **`kms:ViaService`**

A tabela a seguir lista AWS os serviços que estão integrados AWS KMS e oferecem suporte ao uso da chave de `kms:ViaService` condição nas chaves gerenciadas pelo cliente. Os serviços desta tabela podem não estar disponíveis em todas as regiões. Use o `.amazonaws.` com sufixo do AWS KMS `ViaService` nome em todas as AWS partições.

 Note

Talvez seja necessário rolar horizontalmente ou verticalmente para ver todos os dados nessa tabela.

Nome do serviço	AWS KMS ViaService nome
Operações de IA da Amazon	aiops. <i>AWS_region</i> .amazonaws.com
AWS App Runner	apprunner. <i>AWS_region</i> .amazonaws.com
AWS AppFabric	appfabric. <i>AWS_region</i> .amazonaws.com
Amazon AppFlow	appflow. <i>AWS_region</i> .amazonaws.com
AWS Application Migration Service	mgn. <i>AWS_region</i> .amazonaws.com
Amazon Athena	athena. <i>AWS_region</i> .amazonaws.com
AWS Audit Manager	auditmanager. <i>AWS_region</i> .amazonaws.com
Amazon Aurora	rds. <i>AWS_region</i> .amazonaws.com
AWS Backup	backup. <i>AWS_region</i> .amazonaws.com
AWS Backup Gateway	backup-gateway. <i>AWS_region</i> .amazonaws.com
Cópia de modelo do Amazon Bedrock	bedrock. <i>AWS_region</i> .amazonaws.com
SDK do Amazon Chime	chimevoiceconnector. <i>AWS_region</i> .amazonaws.com
AWS Clean Rooms ML	cleanrooms-ml. <i>AWS_region</i> .amazonaws.com
AWS CodeArtifact	codeartifact. <i>AWS_region</i> .amazonaws.com
CodeGuru Revisor da Amazon	codeguru-reviewer. <i>AWS_region</i> .amazonaws.com

Nome do serviço	AWS KMS ViaService nome
Amazon Comprehend	comprehend. <i>AWS_region</i> .amazonaws.com
Amazon Connect	connect. <i>AWS_region</i> .amazonaws.com
Amazon Connect Customer Profiles	profile. <i>AWS_region</i> .amazonaws.com
Amazon Q no Connect	wisdom. <i>AWS_region</i> .amazonaws.com
AWS Database Migration Service (AWS DMS)	dms. <i>AWS_region</i> .amazonaws.com
AWS DeepRacer	deepracer. <i>AWS_region</i> .amazonaws.com
AWS Directory Service	directoryservice. <i>AWS_region</i> .amazonaws.com
Amazon DocumentDB	docdb-elastic. <i>AWS_region</i> .amazonaws.com
Amazon DynamoDB	dynamodb. <i>AWS_region</i> .amazonaws.com
Amazon EC2 Systems Manager (SSM)	ssm. <i>AWS_region</i> .amazonaws.com
Amazon Elastic Block Store (Amazon EBS)	ec2. <i>AWS_region</i> .amazonaws.com (somente EBS)
Amazon Elastic Container Registry (Amazon ECR)	ecr. <i>AWS_region</i> .amazonaws.com
Amazon Elastic File System (Amazon EFS)	elasticfilesystem. <i>AWS_region</i> .amazonaws.com

Nome do serviço	AWS KMS ViaService nome
Amazon ElastiCache	Inclua os dois ViaService nomes no valor da chave de condição: • elasticache. <i>AWS_region</i> .amazonaws.com • dax.<i>AWS_region</i> .amazonaws.com
AWS Elemental MediaTailor	mediatailor. <i>AWS_region</i> .amazonaws.com
AWS Resolução de entidades	entityresolution. <i>AWS_region</i> .amazonaws.com
Amazon EventBridge	events. <i>AWS_region</i> .amazonaws.com
Amazon FinSpace	finspace. <i>AWS_region</i> .amazonaws.com
Amazon Forecast	forecast. <i>AWS_region</i> .amazonaws.com
Amazon FSx	fsx. <i>AWS_region</i> .amazonaws.com
AWS Glue	glue. <i>AWS_region</i> .amazonaws.com
AWS Ground Station	groundstation. <i>AWS_region</i> .amazonaws.com
Amazon GuardDuty	malware-protection. <i>AWS_region</i> .amazonaws.com
AWS HealthLake	healthlake. <i>AWS_region</i> .amazonaws.com
AWS IoT SiteWise	iotsitewise. <i>AWS_region</i> .amazonaws.com
Amazon Kendra	kendra. <i>AWS_region</i> .amazonaws.com

Nome do serviço	AWS KMS ViaService nome
Amazon Keyspaces (para Apache Cassandra)	cassandra. <i>AWS_region</i> .amazonaws.com
Amazon Kinesis	kinesis. <i>AWS_region</i> .amazonaws.com
Amazon Data Firehose	firehose. <i>AWS_region</i> .amazonaws.com
Amazon Kinesis Video Streams	kinesisvideo. <i>AWS_region</i> .amazonaws.com
AWS Lambda	lambda. <i>AWS_region</i> .amazonaws.com
Amazon Lex	lex. <i>AWS_region</i> .amazonaws.com
AWS License Manager	license-manager. <i>AWS_region</i> .amazonaws.com
Amazon Location Service	geo. <i>AWS_region</i> .amazonaws.com
Amazon Lookout for Equipment	lookoutequipment. <i>AWS_region</i> .amazonaws.com
Amazon Lookout for Metrics	lookoutmetrics. <i>AWS_region</i> .amazonaws.com
Amazon Lookout for Vision	lookoutvision. <i>AWS_region</i> .amazonaws.com
Amazon Macie	macie. <i>AWS_region</i> .amazonaws.com
AWS Mainframe Modernization	m2. <i>AWS_region</i> .amazonaws.com
AWS Mainframe Modernization Teste de aplicativos	apptest. <i>AWS_region</i> .amazonaws.com
Amazon Managed Blockchain	managedblockchain. <i>AWS_region</i> .amazonaws.com

Nome do serviço	AWS KMS ViaService nome
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	kafka. <i>AWS_region</i> .amazonaws.com
Amazon Managed Workflows for Apache Airflow (MWAA)	airflow. <i>AWS_region</i> .amazonaws.com
Amazon MemoryDB	memorydb. <i>AWS_region</i> .amazonaws.com
Amazon Monitron	monitron. <i>AWS_region</i> .amazonaws.com
Amazon MQ	mq. <i>AWS_region</i> .amazonaws.com
Amazon Neptune	rds. <i>AWS_region</i> .amazonaws.com
Amazon Nimble Studio	nimble. <i>AWS_region</i> .amazonaws.com
AWS HealthOmics	omics. <i>AWS_region</i> .amazonaws.com
OpenSearch Serviço Amazon	es. <i>AWS_region</i> .amazonaws.com , aoss. <i>AWS_region</i> .amazonaws.com
Pacotes OpenSearch personalizados da Amazon	custom-packages. <i>AWS_region</i> .amazonaws.com
AWS Proton	proton. <i>AWS_region</i> .amazonaws.com
Amazon Quantum Ledger Database (Amazon QLDB)	qldb. <i>AWS_region</i> .amazonaws.com
Insights de Performance do Amazon RDS	rds. <i>AWS_region</i> .amazonaws.com
banco de dados de origem	redshift. <i>AWS_region</i> .amazonaws.com
Editor de consultas do Amazon Redshift V2	sqlworkbench. <i>AWS_region</i> .amazonaws.com

Nome do serviço	AWS KMS ViaService nome
Amazon Redshift Serverless	redshift-serverless. <i>AWS_region</i> .amazonaws.com
Amazon Rekognition	rekognition. <i>AWS_region</i> .amazonaws.com
Amazon Relational Database Service (Amazon RDS)	rds. <i>AWS_region</i> .amazonaws.com
Armazenamento de dados replicado da Amazon	ards. <i>AWS_region</i> .amazonaws.com
SageMaker IA da Amazon	sagemaker. <i>AWS_region</i> .amazonaws.com
AWS Secrets Manager	secretsmanager. <i>AWS_region</i> .amazonaws.com
Amazon Security Lake	securitylake. <i>AWS_region</i> .amazonaws.com
Amazon Simple Email Service (Amazon SES)	ses. <i>AWS_region</i> .amazonaws.com
Amazon Simple Notification Service (Amazon SNS)	sns. <i>AWS_region</i> .amazonaws.com
Amazon Simple Queue Service (Amazon SQS)	sqs. <i>AWS_region</i> .amazonaws.com
Amazon Simple Storage Service (Amazon S3)	s3. <i>AWS_region</i> .amazonaws.com
Tabelas do Amazon S3	s3tables. <i>AWS_region</i> .amazonaws.com
AWS Snowball Edge	importexport. <i>AWS_region</i> .amazonaws.com
AWS Step Functions	states. <i>AWS_region</i> .amazonaws.com

Nome do serviço	AWS KMS ViaService nome
AWS Storage Gateway	storagegateway. <i>AWS_region</i> <i>n</i> .amazonaws.com
AWS Systems Manager Incident Manager	ssm-incidents. <i>AWS_region</i> .amazonaws.com
AWS Systems Manager Incident Manager Contatos	ssm-contacts. <i>AWS_region</i> .amazonaws.com
Amazon Timestream	timestream. <i>AWS_region</i> .amazonaws.com
Amazon Translate	translate. <i>AWS_region</i> .amazonaws.com
Acesso Verificado pela AWS	verified-access. <i>AWS_region</i> <i>n</i> .amazonaws.com
Amazon WorkMail	workmail. <i>AWS_region</i> .amazonaws.com
Amazon WorkSpaces	workspaces. <i>AWS_region</i> .amazonaws.com
Amazon WorkSpaces Thin Client	thinclient. <i>AWS_region</i> .amazonaws.com
Amazon WorkSpaces Web	workspaces-web. <i>AWS_region</i> <i>n</i> .amazonaws.com
AWS X-Ray	xray. <i>AWS_region</i> .amazonaws.com

kms:WrappingAlgorithm

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:WrappingAlgorithm	String	Valor único	GetParametersForImport	Políticas de chaves e políticas do IAM

Essa chave de condição controla o acesso à [GetParametersForImport](#) operação com base no valor do [WrappingAlgorithm](#) parâmetro na solicitação. Você pode usar essa condição para exigir que os principais usem determinado algoritmo para criptografar material de chaves durante o processo de importação. As solicitações para a chave pública e o token de importação exigidos falham quando especificam um algoritmo de empacotamento diferente.

A instrução de política de exemplo a seguir usa a chave de condição kms:WrappingAlgorithm para dar ao usuário de exemplo permissão para chamar a operação GetParametersForImport, mas o impede de usar o algoritmo de empacotamento RSAES_OAEP_SHA_1. Quando o WrappingAlgorithm na solicitação GetParametersForImport é RSAES_OAEP_SHA_1, a operação falha.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:WrappingAlgorithm": "RSAES_OAEP_SHA_1"
    }
  }
}
```

Consulte também

- [kms:ExpirationModel](#)

- [kms:ValidTo](#)
- [kms:WrappingKeySpec](#)

kms:WrappingKeySpec

AWS KMS chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:WrappingKeySpec	String	Valor único	GetParametersForImport	Políticas de chaves e políticas do IAM

Essa chave de condição controla o acesso à [GetParametersForImport](#) operação com base no valor do [WrappingKeySpec](#) parâmetro na solicitação. Você pode usar essa condição para exigir que as entidades principais usem determinado tipo de chave pública durante o processo de importação. Se a solicitação especifica um tipo de chave diferente, ela falha.

Como o único valor válido para o valor do parâmetro WrappingKeySpec é RSA_2048, impedir que os usuários usem esse valor efetivamente os impede de usar a operação GetParametersForImport.

A declaração de política de exemplo a seguir usa a chave de condição kms:WrappingAlgorithm para exigir que WrappingKeySpec na solicitação seja RSA_4096.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:WrappingKeySpec": "RSA_4096"
    }
  }
}
```


Consulte também

- [kms:ExpirationModel](#)
- [kms:ValidTo](#)
- [kms:WrappingAlgorithm](#)

AWS KMS chaves de condição para plataformas certificadas

AWS KMS fornece chaves de condição para apoiar o atestado criptográfico para [AWS Nitro Enclaves e NitroTPM](#). AWS O Nitro Enclaves é um recurso do Amazon EC2 que permite criar ambientes computacionais isolados chamados [enclaves](#) para proteger e processar dados altamente confidenciais. O NitroTPM estende uma funcionalidade de atestação semelhante às instâncias do EC2.

Quando você chama as operações [Decrypt](#), [DeriveSharedSecret](#), [GenerateDataKeyPair](#), ou [GenerateRandom](#) API com um documento de atestado assinado, elas APIs criptografam o texto sem formatação na resposta sob a chave pública do documento de atestado e retornam texto cifrado em vez de texto sem formatação. Esse texto cifrado pode ser descriptografado apenas usando a chave privada no enclave. Para obter mais informações, consulte [Suporte de atestado criptográfico em AWS KMS](#).

Note

Se você não fornecer uma política de chaves ao criar uma AWS KMS chave, AWS crie uma para você. Essa [política de chaves padrão](#) concede às Contas da AWS proprietárias da chave do KMS acesso total à chave e permite que a conta use políticas do IAM para permitir acesso à chave. Essa política permite todas as ações, como [Decrypt](#). A AWS recomenda aplicar a entidade principal do [Permissões de privilégio mínimo](#) às suas políticas de chave do KMS. Você também pode restringir o acesso [modificando a ação da política de chave do KMS](#) de `kms:*` para `NotAction:kms:Decrypt`.

As seguintes chaves de condição permitem limitar as permissões para essas operações com base no conteúdo do documento de atestado assinado. Antes de permitir uma operação, AWS KMS compara o documento de atestado com os valores nessas AWS KMS chaves de condição.

Chaves de condição para o Nitro Enclaves

As seguintes chaves de condição são específicas da atestação do Nitro Enclaves:

kmRecipientAttestation: 384 ImageSha

AWS KMS Chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:RecipientAttestation:ImageSha384	String	Valor único	Decrypt DeriveSharedSecret GenerateDataKey GenerateDataKeyPair GenerateRandom	Políticas de chaves e políticas do IAM

A chave de condição kms:RecipientAttestation:ImageSha384 controla o acesso a Decrypt, DeriveSharedSecret, GenerateDataKey, GenerateDataKeyPair e GenerateRandom com uma chave do KMS quando o resumo da imagem do documento de atestado assinado na solicitação corresponde ao valor na chave de condição. O ImageSha384 valor corresponde ao PCR0 do documento de atestação. Essa chave de condição só é efetiva quando o Recipient parâmetro na solicitação especifica um documento de atestação assinado para um AWS enclave Nitro.

Esse valor também está incluído em [CloudTrail eventos](#) AWS KMS para solicitações de enclaves Nitro.

Por exemplo, a declaração de política chave a seguir permite que a data-processing função use a chave KMS para [descriptografar](#), [DeriveSharedSecret](#), [GenerateDataKey](#), [GenerateDataKeyPair](#) e [GenerateRandom](#). A chave de kms:RecipientAttestation:ImageSha384 condição permite as operações somente quando o valor do resumo da imagem (PCR0) do documento de atestado na solicitação corresponde ao valor do resumo da imagem na condição. Essa chave de

condição só é efetiva quando o Recipient parâmetro na solicitação especifica um documento de atestação assinado para um AWS enclave Nitro.

Se a solicitação não incluir um documento de atestação válido de um enclave AWS Nitro, a permissão será negada porque essa condição não foi satisfeita.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:DeriveSharedSecret",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
"9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

kms:: PCR RecipientAttestation <PCR_ID>

AWS KMS Chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Recip ientAttes tation:PC R<PCR_ID>	String	Valor único	Decrypt DeriveSha redSecret Generated ataKey	Políticas de chaves e políticas do IAM

AWS KMS Chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
			GeneratedDataKeyPair	
			GenerateRandom	

A chave de `kms:RecipientAttestation:PCR<PCR_ID>` condição controla o acesso a `DecryptDeriveSharedSecret`, `GenerateDataKey`, `GenerateDataKeyPair`, e `GenerateRandom` com uma chave KMS somente quando a configuração da plataforma registra (PCRs) do documento de atestação assinado na solicitação coincidir com a PCRs chave de condição. Essa chave de condição só é efetiva quando o `Recipient` parâmetro na solicitação especifica um documento de atestação assinado de um enclave Nitro. AWS

Esse valor também está incluído em [CloudTrail eventos](#) que representam solicitações AWS KMS para enclaves Nitro.

Para especificar um valor de PCR, use o seguinte formato. Concatene o ID do PCR com o nome da chave da condição. Você pode especificar um ID de PCR que identifique uma das [seis medidas de enclave](#) ou um ID de PCR personalizado definido para um caso de uso específico. O valor do PCR deve ser uma string hexadecimal minúscula de até 96 bytes.

```
"kms:RecipientAttestation:PCR<PCR_ID>": "<PCR_value>"
```

Por exemplo, a chave de condição a seguir especifica um valor específico para PCR1, que corresponde ao hash do kernel usado para o enclave e o processo de bootstrap.

```
kms:RecipientAttestation:PCR1:
  "abc1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcde
```

Por exemplo, a instrução de política de chave a seguir permite que a função `data-processing` use a chave do KMS para a operação [Decrypt](#).

A chave de `kms:RecipientAttestation:PCR` condição nessa declaração permite a operação somente quando o PCR1 valor no documento de atestado assinado na solicitação

corresponde ao kms:RecipientAttestation:PCR1 valor na condição. Use a política aStringEqualsIgnoreCase para exigir uma comparação sem distinção entre maiúsculas e minúsculas dos valores de PCR.

Se a solicitação não incluir um documento de atestado, a permissão será negada porque essa condição não foi atendida.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:PCR1":
      "abc1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af7935581474844767453082c6f1"
    }
  }
}
```

Chaves de condição para o NitroTPM

As seguintes chaves de condição são específicas da atestação do NitroTPM:

kms:: NitroTPMPCR RecipientAttestation <PCR_ID>

AWS KMS Chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
kms:Recip ientAttes tation:Ni troTPMPCR <PCR_ID>	String	Valor único	Decrypt DeriveSha redSecret Generated ataKey	Políticas de chaves e políticas do IAM

AWS KMS Chaves de condição	Tipo de condição	Tipo de valor	Operações de API	Tipo de política
			Generated ataKeyPair	
			GenerateR andom	

A chave de `kms:RecipientAttestation:NitroTPMPCR<PCR_ID>` condição controla o acesso a `DecryptDeriveSharedSecret`, `GenerateDataKey`, `GenerateDataKeyPair`, e `GenerateRandom` com uma chave KMS somente quando a configuração da plataforma registra (PCRs) do documento de atestação assinado na solicitação coincidir com a PCRs chave de condição. Essa chave de condição tem efeito somente quando o parâmetro `Recipient` da solicitação especifica um atestado assinado do NitroTPM.

Esse valor também está incluído em [CloudTrail eventos](#) que representam solicitações de NitroTPM.

AWS KMS

Para especificar um valor de PCR, use o seguinte formato. Concatene o ID do PCR com o nome da chave da condição. O valor do PCR deve ser uma string hexadecimal minúscula de até 96 bytes.

```
"kms:RecipientAttestation:NitroTPMPCR<PCR_ID>": "<PCR_value>"
```

Por exemplo, a chave de condição a seguir especifica um valor específico para PCR4:

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

Por exemplo, a instrução de política de chave a seguir permite que a função `data-processing` use a chave do KMS para a operação [Decrypt](#).

A chave de `kms:RecipientAttestation:NitroTPMPCR` condição nessa declaração permite a operação somente quando o PCR4 valor no documento de atestado assinado na solicitação corresponde ao `kms:RecipientAttestation:NitroTPMPCR4` valor na condição. Use a política `aStringEqualsIgnoreCase` para exigir uma comparação sem distinção entre maiúsculas e minúsculas dos valores de PCR.

Se a solicitação não incluir um documento de atestado, a permissão será negada porque essa condição não foi atendida.

```
{
  "Sid" : "Enable NitroTPM data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:NitroTPMPCR4":
      "abc1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af7935581474844767453082c6f1"
    }
  }
}
```

Permissões de privilégio mínimo

Como suas chaves do KMS protegem informações confidenciais, recomendamos seguir o princípio do acesso com privilégio mínimo. Delegue as permissões mínimas necessárias para executar uma tarefa ao definir as políticas de chave. Só permita todas as ações (`kms : *`) em uma política de chave do KMS se planejar restringir ainda mais as permissões com políticas adicionais do IAM. Se planejar gerenciar permissões com políticas do IAM, limite quem tem a capacidade de criar e anexar políticas do IAM a entidades principais do IAM e [monitore as mudanças nas políticas](#).

Se você permitir todas as ações (`kms : *`) na política de chaves e na política do IAM, a entidade principal terá permissões administrativas e de uso para a chave do KMS. Como prática recomendada de segurança, recomendamos delegar essas permissões somente a entidades principais específicas. Você pode fazer isso nomeando explicitamente a entidade principal na política de chave ou limitando as entidades principais às quais a política do IAM está anexada. Você também pode usar [chaves de condição](#) para restringir as permissões. Por exemplo, você pode usar [aws:PrincipalTag](#) para permitir todas as ações se a entidade principal que está fazendo a chamada de API tiver a tag especificada na regra de condição.

Para ajudar a entender como as declarações de política são avaliadas em AWS, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM. Antes de criar políticas, recomendamos revisar

esse tópico para reduzir a chance de sua política ter efeitos indesejados, como fornecer acesso a entidades principais que não deveriam ter acesso.

 Tip

Ao testar uma aplicação em um ambiente que não seja de produção, use o [IAM Access Analyzer](#) para ajudar você a aplicar privilégios mínimos às suas políticas do IAM.

Se você usa usuários do IAM em vez de funções do IAM, é altamente recomendável ativar a [autenticação AWS multifator](#) (MFA) para mitigar a vulnerabilidade das credenciais de longo prazo. Você pode usar o MFA para:

- Exija que os usuários validem suas credenciais com a MFA antes de realizar ações privilegiadas, como agendar a exclusão da chave.
- Divida a propriedade de uma conta de administrador, senha e dispositivo de MFA entre indivíduos para implementar a autorização dividida.

Saiba mais

- [AWS políticas gerenciadas para funções de trabalho](#)
- [Técnicas para criar políticas do IAM de privilégio mínimo](#)

Implementação de permissões de privilégio mínimo

Ao conceder permissão a um AWS serviço para usar uma chave KMS, certifique-se de que a permissão seja válida somente para os recursos que o serviço deve acessar em seu nome. Essa estratégia de privilégios mínimos ajuda a evitar o uso não autorizado de uma chave KMS quando as solicitações são passadas entre serviços. AWS

Para implementar uma estratégia de privilégios mínimos, recomendamos o uso de chaves de condição de contexto de AWS KMS criptografia e do ARN de origem global ou das chaves de condição da conta de origem.

Uso de chaves de condição de contexto de criptografia

A maneira mais eficaz de implementar permissões com menos privilégios ao usar AWS KMS recursos é incluir as chaves de [kms:EncryptionContextKeys](#) condição [kms:EncryptionContext:context-](#)

[key](#)ou na política que permite que os diretores chamem operações AWS KMS criptográficas. Essas chaves de condição são particularmente eficazes porque associam a permissão ao [contexto de criptografia](#) que está vinculado ao texto cifrado quando o recurso é criptografado.

[Use chaves de condições de contexto de criptografia somente quando a ação na declaração de política for CreateGrantuma operação criptográfica AWS KMS simétrica que usa um EncryptionContext parâmetro, como as operações como GenerateDataKeyou Decrypt.](#) (Para ver uma lista das operações válidas, consulte [kms:EncryptionContext:context-key](#) ou [kms:EncryptionContextKeys](#). Se você usar essas chaves de condição para permitir outras operações, como, por exemplo [DescribeKey](#), a permissão será negada.

Defina o valor para o contexto de criptografia usado pelo serviço ao criptografar o recurso. Essas informações geralmente estão disponíveis no capítulo Segurança da documentação do serviço. Por exemplo, o [contexto de criptografia do AWS Proton](#) identifica o recurso AWS Proton e seu modelo associado. O [contexto de criptografia do AWS Secrets Manager](#) identifica o segredo e sua versão. O [contexto de criptografia para Amazon Location](#) identifica o rastreador ou a coleção.

O exemplo de declaração de política de chave a seguir permite que o Amazon Location Service crie concessões em nome de usuários autorizados. Essa declaração de política limita a permissão usando as chaves [kms: ViaService](#), [kms: CallerAccount](#) e de `kms:EncryptionContext:context-key` condição para vincular a permissão a um recurso específico do rastreador.

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

Uso de chaves de condição `aws:SourceArn` ou `aws:SourceAccount`

Quando a entidade principal de uma instrução de política de chave é uma [entidade principal de serviço da AWS](#), recomendamos usar as chaves de condição globais `aws:SourceArn` ou `aws:SourceAccount`, além da chave de condição `kms:EncryptionContext:context-key`. O ARN e os valores da conta são incluídos no contexto de autorização somente quando uma solicitação AWS KMS vem de outro AWS serviço. Essa combinação de condições implementa permissões de privilégio mínimo e evita um possível [cenário de auxiliar confuso](#). Os diretores de serviços normalmente não são usados como diretores em uma política chave, mas alguns AWS serviços, como o AWS CloudTrail, exigem isso.

Para usar as chaves de condição globais `aws:SourceArn` ou `aws:SourceAccount`, defina o valor como o nome do recurso da Amazon (ARN) ou a conta do recurso que está sendo criptografado. Por exemplo, em uma declaração de política de chave que concede ao AWS CloudTrail permissão para criptografar uma trilha, defina o valor de `aws:SourceArn` como o ARN da trilha. Sempre que possível, use `aws:SourceArn`, que é mais específico. Defina o valor como o ARN ou um padrão de ARN com caracteres curinga. Se você não conhece o ARN do recurso, use `aws:SourceAccount` em vez disso.

Note

Se um ARN de recurso incluir caracteres que não são permitidos em uma política de AWS KMS chave, você não pode usar esse ARN de recurso no valor da chave de condição. `aws:SourceArn` Em vez disso, use a chave de condição `aws:SourceAccount`. Para obter detalhes sobre as regras de documento de política de chaves, consulte [Formato de política de chaves](#).

No exemplo de política de chaves a seguir, a entidade principal que obtém as permissões é a entidade principal do serviço AWS CloudTrail, `cloudtrail.amazonaws.com`. Para implementar o privilégio mínimo, essa política usa as chaves de condição `aws:SourceArn` e `kms:EncryptionContext:context-key`. A declaração de política permite CloudTrail usar a chave KMS para [gerar a chave de dados](#) que ela usa para criptografar uma trilha. As condições `aws:SourceArn` e `kms:EncryptionContext:context-key` são avaliadas de forma independente. Qualquer solicitação para usar a chave do KMS para a operação especificada deve atender às duas condições.

Para restringir a permissão do serviço à trilha `finance` na conta de exemplo (111122223333) e na região `us-west-2`, esta declaração de política define a chave de condição `aws:SourceArn` como o ARN de uma trilha específica. A declaração de condição usa o [ArnEquals](#) operador para garantir que cada elemento no ARN seja avaliado de forma independente durante a correspondência. O exemplo também usa a chave de condição `kms:EncryptionContext:context-key` para limitar a permissão a trilhas em uma determinada conta e região.

Antes de usar essa política de chaves, substitua os valores de ID da conta, região e nome da trilha de exemplo por valores válidos da sua conta.

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 para AWS KMS

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. AWS KMS oferece suporte ao ABAC, permitindo que você controle o acesso às chaves gerenciadas pelo cliente com base nas tags e aliases associados às chaves KMS. As chaves de condição de tag e alias que permitem a entrada do ABAC AWS KMS fornecem uma maneira poderosa e flexível de autorizar os diretores a usar as chaves do KMS sem editar políticas ou gerenciar concessões. Porém, você deve usar esse recurso com cuidado, para que as entidades principais não sejam tenham o acesso inadvertidamente permitido ou negado.

Se você usar ABAC, esteja ciente de que a permissão para gerenciar etiquetas e aliases agora é uma permissão de controle de acesso. Certifique-se de conhecer as etiquetas e os aliases existentes em todas as chaves do KMS antes de implantar uma política que depende de etiquetas ou aliases. Tome precauções razoáveis ao adicionar, excluir e atualizar aliases e ao marcar e desmarcar chaves. Conceda permissões para gerenciar etiquetas e aliases apenas às entidades principais que precisam deles e limite as etiquetas e os aliases que eles podem gerenciar.

Observações

Ao usar o ABAC para AWS KMS, tenha cuidado ao dar permissão aos diretores para gerenciar tags e aliases. A alteração de uma etiqueta ou de um alias pode permitir ou negar permissão para uma chave do KMS. Os administradores de chaves que não tiverem permissão para alterar políticas de chaves ou criar concessões poderão controlar o acesso às chaves do KMS se tiverem permissão para gerenciar etiquetas ou aliases.

Pode levar até cinco minutos para que alterações de etiqueta e alias afetem a autorização de chaves do KMS. Alterações recentes podem estar visíveis em operações de API antes de afetarem a autorização.

Para controlar o acesso a uma chave do KMS com base em seu alias, você deve usar uma chave de condição. Você não pode usar um alias para representar uma chave do KMS no elemento `Resource` de uma instrução de política. Quando um alias é exibido no elemento `Resource`, a instrução de política aplica-se a esse alias, e não à chave do KMS associada.

Saiba mais

- Para obter detalhes sobre o AWS KMS suporte ao ABAC, incluindo exemplos, consulte [Usar aliases para controlar o acesso a chaves do KMS](#) e [Usar tags para controlar o acesso a chaves do KMS](#).

- Para obter mais informações gerais sobre o uso de tags para controlar o acesso aos AWS recursos, consulte [Para AWS que serve o ABAC?](#) e [controlar o acesso aos AWS recursos usando tags de recursos](#) no Guia do usuário do IAM.

Chaves de condição ABAC para AWS KMS

Para autorizar o acesso a chaves do KMS com base em suas etiquetas e aliases, use as seguintes chaves de condição em uma política de chaves ou política do IAM.

Chave de condição para ABAC	Description	Tipo de política	AWS KMS operações
leis: ResourceTag	A etiqueta (chave e valor) na chave do KMS corresponde à etiqueta (chave e valor) ou ao padrão de etiqueta na política	Somente política do IAM	Operações de recursos de chaves do KMS ²
aws:RequestTag//tecla de tag	A etiqueta (chave e valor) na solicitação corresponde à etiqueta (chave e valor) ou ao padrão de etiqueta na política	Políticas de chaves e políticas do IAM ¹	TagResource , UntagResource
leis: TagKeys	Chaves de etiqueta na solicitação correspondem a chaves de etiqueta na política	Políticas de chaves e políticas do IAM ¹	TagResource , UntagResource
kms: ResourceAliases	Aliases associados à chave do KMS correspondem a aliases ou padrões de alias na política	Somente política do IAM	Operações de recursos de chaves do KMS ²

Chave de condição para ABAC	Description	Tipo de política	AWS KMS operações
kms: RequestAlias	O alias que represent a a chave do KMS na solicitação correspon de ao alias ou aos padrões de alias na política.	Políticas de chaves e políticas do IAM ¹	Operações criptográficas , DescribeKey , GetPublicKey

¹Qualquer chave de condição que possa ser usada em uma política de chaves também pode ser usada em uma política do IAM, mas somente se [a política de chaves a permitir](#).

²A operação de recurso de chave do KMS é uma operação autorizada para uma chave do KMS específica. Para identificar as operações de recursos de chaves do KMS, no [tabela de permissões do AWS KMS](#), procure um valor de chave do KMS na coluna Resources para a operação.

Por exemplo, é possível usar essas chaves de condição para criar as seguintes políticas.

- Uma política do IAM com `kms:ResourceAliases` que concede permissão para usar chaves do KMS com um alias específico ou um padrão de alias. Isso é um pouco diferente das políticas que dependem de tags: embora você possa usar padrões de aliases em uma política, cada alias deve ser exclusivo em uma região Conta da AWS e. Isso permite que você aplique uma política a um conjunto selecionado de chaves KMS sem listar a chave ARNs das chaves KMS na declaração de política. Para adicionar ou remover chaves do KMS do conjunto, altere o alias da chave do KMS.
- Uma política de chaves com `kms:RequestAlias` que permite que as entidades principais usem uma chave do KMS em uma operação `Encrypt`, mas somente quando a solicitação `Encrypt` usa esse alias para identificar a chave do KMS.
- Uma política do IAM com `aws:ResourceTag/tag-key` que nega permissão para usar chaves do KMS com uma determinada chave de etiqueta e um valor de etiqueta. Isso permite aplicar uma política a um conjunto selecionado de chaves KMS sem listar a chave ARNs das chaves KMS na declaração de política. Para adicionar ou remover chaves do KMS do conjunto, marque ou desmarque a chave do KMS.
- Uma política do IAM com `aws:RequestTag/tag-key` que permite que as entidades principais excluam somente etiquetas `"Purpose"="Test"` de chaves do KMS.

- Uma política do IAM com `aws:TagKeys` que nega permissão para marcar ou desmarcar uma chave do KMS com uma chave de etiqueta `Restricted`.

O ABAC torna o gerenciamento de acesso flexível e escalável. Por exemplo, você pode usar a chave de condição `aws:ResourceTag/tag-key` para criar uma política do IAM que permite que as entidades principais usem uma chave do KMS para operações especificadas somente quando essa chave tem uma etiqueta `Purpose=Test`. A política aplica-se a todas as chaves do KMS em todas as regiões da Conta da AWS.

Quando anexada a um usuário ou função, a seguinte política do IAM permite que as entidades principais usem todas as chaves do KMS existentes com uma etiqueta `Purpose=Test` para as operações especificadas. Para fornecer esse acesso a chaves do KMS novas ou existentes, não é necessário alterar a política. Basta anexar a etiqueta `Purpose=Test` às chaves do KMS. Da mesma forma, para remover esse acesso das chaves do KMS com uma etiqueta `Purpose=Test`, edite ou exclua essa etiqueta.

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"
        }
      }
    }
  ]
}
```

No entanto, se você usar esse recurso, tenha cautela ao gerenciar etiquetas e aliases. Operações de adicionar, alterar ou excluir uma marca ou alias podem inadvertidamente permitir ou negar acesso a uma chave do KMS. Os administradores de chaves que não tiverem permissão para alterar políticas de chaves ou criar concessões poderão controlar o acesso às chaves do KMS se tiverem permissão para gerenciar etiquetas e aliases. Para atenuar esse risco, considere [limitar permissões para gerenciar etiquetas e aliases](#). Por exemplo, é possível permitir que apenas as entidades principais selecionadas gerenciem etiquetas do Purpose=Test. Para obter mais detalhes, consulte [Usar aliases para controlar o acesso a chaves do KMS](#) e [Usar tags para controlar o acesso a chaves do KMS](#).

Etiquetas ou aliases?

AWS KMS suporta ABAC com tags e aliases. Ambas as opções proporcionam uma estratégia de controle de acesso flexível e escalável, mas são ligeiramente diferentes uma da outra.

Você pode decidir usar tags ou usar aliases com base em seus padrões de AWS uso específicos. Por exemplo, se você já concedeu permissões de marcação para a maioria dos administradores, talvez seja mais fácil controlar uma estratégia de autorização baseada em aliases. Ou, se você estiver se aproximando da cota de [aliases por chave do KMS](#), talvez prefira uma estratégia de autorização baseada em etiquetas.

Os benefícios a seguir são de interesse geral.

Benefícios do controle de acesso baseado em tags

- O mesmo mecanismo de autorização para diferentes tipos de AWS recursos.

É possível usar a mesma etiqueta ou chave de etiqueta para controlar o acesso a vários tipos de recursos, como cluster do Amazon Relational Database Service (Amazon RDS), volume do Amazon Elastic Block Store (Amazon EBS) e chave do KMS. Esse recurso permite diversos modelos de autorização diferentes que são mais flexíveis que o controle de acesso baseado em função tradicional.

- Autorize o acesso a um grupo de chaves do KMS.

É possível usar etiquetas para gerenciar o acesso a um grupo de chaves do KMS na mesma região e Conta da AWS. Atribua a mesma etiqueta ou chave de etiqueta às chaves do KMS que você escolher. Em seguida, crie uma declaração easy-to-maintain de política simples baseada na tag ou na chave da tag. Para adicionar ou remover uma chave do KMS do seu grupo de autorização, adicione ou remova a etiqueta: não é necessário editar a política.

Benefícios do controle de acesso baseado em alias

- Autorize o acesso a operações de criptografia com base em aliases.

A maioria das condições de política baseadas em solicitações para atributos, incluindo [aws:RequestTag/tag-key](#), afetam somente as operações que adicionam, editam ou excluem o atributo. Mas a chave [kms: RequestAlias](#) condition controla o acesso às operações criptográficas com base no alias usado para identificar a chave KMS na solicitação. Por exemplo, você pode conceder uma permissão à entidade principal para usar uma chave do KMS em uma operação Encrypt, mas somente quando o valor do parâmetro KeyIdéalias/restricted-key-1. Para atender a essa condição, é necessário o seguinte:

- A chave do KMS deve estar associada a esse alias.
- A solicitação deve usar o alias para identificar a chave do KMS.
- A entidade principal deve ter permissão para usar a chave do KMS sujeita à condição `kms:RequestAlias`.

Isso é particularmente útil se seus aplicativos geralmente usam nomes de alias ou alias ARNs para se referir às chaves KMS.

- Forneça permissões muito limitadas.

Um alias deve ser exclusivo em uma região Conta da AWS e. Como resultado, o ato de conceder acesso para as entidades principais a uma chave do KMS baseada em um alias pode ser muito mais restritivo do que o conceder acesso para eles com base em uma etiqueta. Diferentemente de aliases, etiquetas podem ser atribuídas a várias chaves do KMS na mesma conta e região. Se você escolher, é possível usar um padrão de alias, como `alias/test*`, a fim de conceder acesso para as entidades principais a um grupo de chaves do KMS na mesma conta e região. Porém, permitir ou negar o acesso a um alias específico possibilita um controle muito rigoroso sobre as chaves do KMS.

Solução de problemas do ABAC para AWS KMS

Controlar o acesso a chaves do KMS com base em etiquetas e aliases é uma estratégia conveniente e poderosa. No entanto, ela está propensa a alguns erros previsíveis que convém evitar.

Acesso alterado devido a mudanças de etiqueta

Se uma etiqueta for excluída ou seu valor for alterado, as entidades principais que tiverem acesso a uma chave do KMS com base apenas nessa etiqueta terão acesso negado à chave do KMS. Isso

também pode acontecer quando uma etiqueta incluída em uma instrução de política de negação é adicionada a uma chave do KMS. Adicionar uma etiqueta relacionada a uma política a uma chave do KMS pode permitir acesso a entidades principais que não devem ter esse acesso a uma chave do KMS.

Por exemplo, suponha que uma entidade principal tenha acesso a uma chave do KMS com base na etiqueta `Project=Alpha`, como a permissão fornecida pelo seguinte exemplo de instrução de política do IAM.

JSON

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

Se a etiqueta for excluída dessa chave do KMS ou se o valor da etiqueta for alterado, a entidade principal não terá mais permissão para usar a chave do KMS nas operações especificadas. Isso pode ficar evidente quando o diretor tenta ler ou gravar dados em um AWS serviço que usa uma chave gerenciada pelo cliente. Para rastrear a alteração da tag, revise seus CloudTrail registros [TagResource](#) ou [UntagResource](#) entradas.

Para restaurar o acesso sem atualizar a política, altere as etiquetas na chave do KMS. Essa ação tem impacto mínimo diferente de um breve período em que está entrando em vigor no AWS KMS. Para evitar um erro como esse, conceda permissões de marcação e desmarcação apenas a

entidades principais que precisem delas [elimite as permissões de marcação](#) a etiquetas que eles precisam gerenciar. Antes de alterar uma etiqueta, pesquise políticas para detectar o acesso que depende dessa etiqueta e obtenha chaves do KMS em todas as regiões que tenham a etiqueta. Você pode considerar criar um CloudWatch alarme da Amazon quando determinadas tags forem alteradas.

Alteração do acesso devido a mudanças de alias

Se um alias for excluído ou associado a uma chave do KMS diferente, as entidades principais que tiverem acesso a essa chave do KMS com base apenas nesse alias terão acesso negado a ela. Isso também pode acontecer quando um alias associado a uma chave do KMS é incluído em uma instrução de política de negação. Adicionar um alias relacionado a uma política a uma chave do KMS também pode permitir acesso a entidades principais que não devem ter esse acesso a uma chave do KMS.

Por exemplo, a seguinte declaração de política do IAM usa a chave de ResourceAliases condição [kms:](#) para permitir o acesso às chaves KMS em diferentes regiões da conta com qualquer um dos aliases especificados.

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"
          ]
        }
      }
    }
  ]
}
```

```
}  
}  
]  
}
```

Para rastrear a alteração do alias, revise seus CloudTrail registros de [CreateAliasUpdateAlias](#), e [DeleteAlias](#) entradas.

Para restaurar o acesso sem atualizar a política, modifique o alias associado à chave do KMS. Como cada alias pode ser associado a apenas uma chave do KMS em uma conta e região, o gerenciamento de aliases é um pouco mais difícil que o de etiquetas. Restaurar o acesso a algumas das entidades principais em uma chave do KMS pode negar acesso para as mesmas ou outras entidades principais a uma chave do KMS diferente.

Para evitar esse erro, conceda permissões de gerenciamento de alias somente às entidades principais que precisam dele e [limite suas permissões de gerenciamento de alias](#) para aliases que elas precisam gerenciar. Antes de atualizar ou excluir um alias, pesquise políticas para detectar o acesso que depende do alias e localize chaves do KMS em todas as regiões associadas a ele.

Acesso negado devido à cota de aliases

Os usuários autorizados a usar uma chave KMS por meio de uma ResourceAliases condição [kms:](#) receberão uma `AccessDenied` exceção se a chave KMS exceder os [aliases padrão por cota de chave KMS](#) para essa conta e região.

Para restaurar o acesso, exclua aliases associados à chave do KMS, para que esta esteja em conformidade com a cota. Outra opção é usar um mecanismo alternativo para conceder aos usuários acesso à chave do KMS.

Alteração de autorização atrasadas

Alterações feitas em etiquetas e aliases podem levar até cinco minutos para afetar a autorização de chaves do KMS. Como resultado, uma alteração de etiqueta ou alias pode ser refletida nas respostas das operações da API antes que estas afetem a autorização. É provável que esse atraso seja maior do que o breve atraso eventual de consistência que afeta a maioria das AWS KMS operações.

Por exemplo, você pode ter uma política do IAM que permita que certas entidades principais utilizem qualquer chave do KMS com uma etiqueta `"Purpose"="Test"`. Em seguida, você adiciona a etiqueta `"Purpose"="Test"` a uma chave do KMS. Embora a [TagResource](#) operação seja

concluída e a [ListResourceTags](#) resposta confirme que a tag está atribuída à chave KMS, os diretores podem não ter acesso à chave KMS por até cinco minutos.

Para evitar erros, crie esse atraso esperado no seu código.

Solicitações com falha devido a atualizações de alias

Quando você atualiza um alias, associa um alias existente a uma chave do KMS diferente.

A [descriptografia](#) e as [ReEncrypt](#) solicitações que especificam o [nome do alias](#) ou o [ARN](#) do alias podem falhar porque o alias agora está associado a uma chave KMS que não criptografou o texto cifrado. Essa situação geralmente retorna uma `IncorrectKeyException` ou `NotFoundException`. Ou, se a solicitação não tiver o parâmetro `KeyId` ou `DestinationKeyId`, a operação pode falhar com uma exceção `AccessDenied` porque o autor da chamada não tem mais acesso à chave do KMS que criptografou o texto cifrado.

Você pode rastrear a alteração examinando CloudTrail os registros de [CreateAliasUpdateAlias](#), e as entradas de [DeleteAlias](#) registro. Você também pode usar o valor do `LastUpdatedDate` campo na [ListAliases](#) resposta para detectar uma alteração.

Por exemplo, o [ListAliases](#) exemplo de resposta a seguir mostra que o `ProjectAlpha_Test` alias na `kms:ResourceAliases` condição foi atualizado. Como resultado, as entidades principais que têm acesso com base nesse alias perdem acesso à chave do KMS anteriormente associada. Em vez disso, elas têm acesso à chave do KMS recém-associada.

```
$ 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  
    }  
]  
}
```

A solução para essa alteração não é simples. É possível atualizar o alias novamente para associá-lo à chave do KMS original. No entanto, antes de fazer isso, você precisa considerar o efeito dessa alteração na chave do KMS atualmente associada. Se as entidades principais usaram a última chave do KMS em operações de criptografia, talvez elas precisem de acesso contínuo a ela. Nesse caso, convém atualizar a política para garantir que as entidades principais tenham permissão para usar as duas chaves do KMS.

Para evitar um erro como esse, antes de atualizar um alias, pesquise políticas para detectar o acesso que depende do alias. Em seguida, obtenha chaves do KMS em todas as regiões associadas a esse alias. Conceda permissões de gerenciamento de alias somente às entidades principais que precisam dele e [limite suas permissões de gerenciamento de alias](#) para aliases que elas precisam gerenciar.

RBAC para AWS KMS

O controle de acesso baseado em funções (RBAC) é uma estratégia de autorização que fornece aos usuários somente as permissões necessárias para realizar suas tarefas e nada mais. O AWS KMS viabiliza o RBAC permitindo que você controle o acesso às suas chaves especificando permissões granulares sobre o uso da chave nas [políticas de chave](#). As políticas de chave especificam um recurso, ação, efeito, entidade principal e condições opcionais para conceder acesso às chaves.

Para implementar o RBAC em AWS KMS, recomendamos separar as permissões dos principais usuários e administradores de chaves.

Key users

O exemplo de política de chave a seguir permite que o perfil do IAM `ExampleUserRole` use a chave do 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": "*"
}
```

Seus usuários de chave podem precisar de menos permissões do que o usuário neste exemplo. Atribua apenas as permissões necessárias para o usuário. Use as seguintes perguntas para ajudar a refinar ainda mais as permissões.

- Quais entidades principais do IAM (perfis ou usuários) precisam acessar a chave?
- Quais ações cada entidade principal precisa realizar com a chave? Por exemplo, a entidade principal precisa somente das permissões de Criptografar e Assinar?
- O usuário é humano ou um AWS serviço? Se for um AWS serviço, você pode usar a [chave de condição](#) para restringir o uso da chave a um AWS serviço específico.

Key administrators

O exemplo de política de chave a seguir permite que o perfil do IAM `ExampleAdminRole` administre a chave do 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": "*"
}
```

Seus administradores de chave podem precisar de menos permissões do que o administrador neste exemplo. Atribua somente as permissões de que seus administradores de chave precisam.

Dê aos usuários somente as permissões necessárias para cumprir suas funções. As permissões de um usuário podem variar dependendo se a chave é usada em ambientes de teste ou de produção. Se você usar permissões menos restritivas em determinados ambientes que não sejam de produção, implemente um processo para testar as políticas antes que elas sejam liberadas para produção.

Saiba mais

- [Identidades do IAM \(usuários, grupos de usuários e perfis\)](#)
- [Tipos de controle de acesso](#)

Permitir que usuários de outras contas usem uma chave do KMS

Você pode permitir que usuários ou funções em outras Conta da AWS usem uma chave KMS em sua conta. O acesso entre contas requer permissão na política de chaves da chave do KMS e em uma política do IAM na conta do usuário externo.

A permissão entre contas é efetiva apenas nas seguintes operações:

- [Operações criptográficas](#)
- [CreateGrant](#)
- [DescribeKey](#)
- [GetKeyRotationStatus](#)
- [GetPublicKey](#)

- [ListGrants](#)
- [RetireGrant](#)
- [RevokeGrant](#)

Se você conceder a um usuário em outra conta permissão para outras operações, essas permissões não terão efeito. Por exemplo, se você der a um principal em uma conta diferente [kms: ListKeys](#) permissão em uma política do IAM ou [kms: ScheduleKeyDeletion](#) permissão em uma chave KMS em uma política de chaves, as tentativas do usuário de chamar essas operações em seus recursos ainda falharão.

Para obter detalhes sobre o uso de chaves KMS em diferentes contas para AWS KMS operações, consulte a coluna Uso entre contas no e. [AWS KMS permissões Usar chaves do KMS em outras contas](#) Existe também uma seção Uso entre contas em cada descrição da API na [Referência de APIs do AWS Key Management Service](#).

 Warning

Tenha cautela ao conceder permissões para as entidades principais usarem suas chaves do KMS. Sempre que possível, siga o princípio de menor privilégio. Conceda aos usuários acesso apenas às chaves do KMS necessárias para a conclusão de suas operações.

Além disso, tenha cautela ao usar qualquer chave do KMS desconhecida, especialmente uma chave do KMS em outra conta. Usuários mal-intencionados podem conceder a você permissões para usar a chave do KMS deles para obter informações sobre você ou sua conta.

Para obter informações sobre como usar políticas para proteger os recursos em sua conta, consulte [Práticas recomendadas para políticas do IAM](#).

Para conceder permissão para usuários e funções usarem uma chave do KMS em outra conta, você deve usar dois tipos diferentes de políticas:

- A política de chaves da chave do KMS deve conceder à conta externa (ou aos usuários e às funções na conta externa) permissão para usar a chave do KMS. A política de chaves está na conta proprietária da chave do KMS.
- Políticas do IAM na conta externa devem delegar as permissões de políticas de chaves para seus usuários e funções. Essas políticas são definidas na conta externa e concedem permissões a usuários e funções nessa conta.

A política de chaves determina quem pode ter acesso à chave do KMS. A política do IAM determina quem tem acesso à chave do KMS. Sozinhas, nem a política de chaves, nem a política do IAM são suficientes: você deve alterar as duas.

Para editar a política de chaves, você pode usar a [Exibição de Política](#) nas [PutKeyPolicy](#) operações Console de gerenciamento da AWS [CreateKey](#) ou usar as.

Para obter ajuda sobre como editar políticas do IAM, consulte [Usando políticas do IAM com AWS KMS](#).

Para obter um exemplo que mostra como a política de chaves e as políticas do IAM funcionam em conjunto para permitir o uso de uma chave do KMS em uma conta diferente, consulte [Exemplo 2: O usuário assume a função com permissão para usar uma chave KMS em uma outra Conta da AWS](#).

[Você pode ver as AWS KMS operações resultantes entre contas na chave KMS em seus AWS CloudTrail registros](#). Operações que usam chaves do KMS em outras contas são registradas na conta do autor da chamada e na conta do proprietário da chave do KMS.

Tópicos

- [Etapa 1: Incluir uma declaração de política de chaves na conta local](#)
- [Etapa 2: Adicionar políticas do IAM à conta externa](#)
- [Permitindo o uso de chaves KMS externas com Serviços da AWS](#)
- [Usar chaves do KMS em outras contas](#)

Note

Os exemplos neste tópico mostram como usar uma política de chaves e uma política do IAM juntas para fornecer e limitar o acesso a uma chave do KMS. Esses exemplos genéricos não se destinam a representar as permissões que qualquer um em particular AWS service (Serviço da AWS) exige em uma chave KMS. Para obter informações sobre as permissões que um AWS service (Serviço da AWS) homem exige, consulte o tópico de criptografia na documentação do serviço.

Etapa 1: Incluir uma declaração de política de chaves na conta local

A política de chaves de uma chave do KMS é o determinante principal de quem pode acessar a chave do KMS e quais operações podem ser realizadas. A política de chaves está sempre na conta

proprietária da chave do KMS. Ao contrário de políticas do IAM, políticas de chaves não especificam um recurso. O recurso é a chave do KMS associada à política de chaves. Ao fornecer permissões entre contas, a política de chaves para a chave do KMS deve conceder à conta externa (ou aos usuários e perfis na conta externa) permissão para usar a chave do KMS.

Para conceder permissão a uma conta externa para usar a chave do KMS, adicione uma instrução à política de chaves que especifique a conta externa. No elemento `Principal` da política de chaves, insira o Amazon Resource Name (ARN) da conta externa.

Quando você especifica uma conta externa em uma política de chaves, os administradores do IAM na conta externa podem usar políticas do IAM para delegar essas permissões a todos os usuários e funções na conta externa. Eles também podem decidir quais ações especificadas na política de chaves os usuários e as funções podem executar.

As permissões concedidas à conta externa e suas entidades principais serão efetivas somente se a conta externa estiver habilitada na região que hospeda a chave do KMS e sua política de chaves. Para obter informações sobre regiões não habilitadas por padrão (“regiões de adesão”), consulte [Gerenciar Regiões da AWS](#) em Referência geral da AWS.

Por exemplo, suponha que você queira permitir que a conta 444455556666 use uma chave do KMS de criptografia simétrica na conta 111122223333. Para isso, adicione uma instrução de política, como a instrução no exemplo a seguir, à política de chaves na conta 111122223333. Essa instrução de política permite que a conta externa (444455556666) use a chave do KMS em operações criptográficas para chaves do KMS de criptografia simétrica.

Note

O exemplo a seguir representa uma amostra de política de chaves para o compartilhamento de uma chave do KMS com outra conta. Substitua os valores de exemplo `Sid`, `Principal` e `Action` por valores válidos para o uso pretendido de sua chave do KMS.

```
{
  "Sid": "Allow an external account to use this KMS key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::444455556666:root"
    ]
  },
}
```

```
"Action": [  
    "kms:Encrypt",  
    "kms:Decrypt",  
    "kms:ReEncrypt*",  
    "kms:GenerateDataKey*",  
    "kms:DescribeKey"  
],  
"Resource": "*" }  
}
```

Em vez de conceder permissão à conta externa, você pode especificar usuários e funções externos específicos na política de chaves. No entanto, esses usuários e funções não poderão usar a chave do KMS até que os administradores do IAM na conta externa associem as políticas do IAM adequadas às suas identidades. As políticas do IAM podem conceder permissão a todos ou a um subconjunto de usuários e funções externos especificados na política de chaves. E podem permitir todas ou um subconjunto das ações especificadas na política de chaves.

Especificar identidades em uma política de chaves restringe as permissões que os administradores do IAM na conta externa podem fornecer. No entanto, isso torna o gerenciamento de políticas com duas contas mais complexo. Por exemplo, suponha que você precise adicionar um usuário ou uma função. É necessário adicionar essa identidade à política de chaves na conta que possui a chave do KMS e criar políticas do IAM na conta da identidade.

Para especificar determinados usuários ou funções externos em uma política de chaves, no elemento `Principal`, insira o Amazon Resource Name (ARN) de um usuário ou função na conta externa.

Por exemplo, a instrução de política de chaves exemplificada a seguir permite que `ExampleRole` na conta 444455556666 use uma chave do KMS na conta 111122223333. Essa instrução de política de chaves permite que a conta externa (444455556666) use a chave do KMS em operações criptográficas para chaves do KMS de criptografia simétrica.

Note

O exemplo a seguir representa uma amostra de política de chaves para o compartilhamento de uma chave do KMS com outra conta. Substitua os valores de exemplo `Sid`, `Principal` e `Action` por valores válidos para o uso pretendido de sua chave do KMS.

```
{
```

```
"Sid": "Allow an external account to use this KMS key",
"Effect": "Allow",
"Principal": {
  "AWS": "arn:aws:iam::444455556666:role/ExampleRole"
},
"Action": [
  "kms:Encrypt",
  "kms:Decrypt",
  "kms:ReEncrypt*",
  "kms:GenerateDataKey*",
  "kms:DescribeKey"
],
"Resource": "*"
}
```

Note

Não defina a entidade principal como um asterisco (*) em qualquer instrução de política de chave que permita permissões, a menos que você utilize [condições](#) para limitar a política de chave. Um asterisco dá a cada identidade em cada Conta da AWS permissão para usar a chave KMS, a menos que outra declaração de política a negue explicitamente. Usuários de outros usuários Contas da AWS podem usar sua chave KMS sempre que tiverem permissões correspondentes em suas próprias contas.

Você também precisa decidir quais permissões deseja conceder à conta externa. Por exemplo, você pode conceder aos usuários permissão para descriptografar, mas não para criptografar, ou permissão para visualizar a chave do KMS, mas não para usá-la. Para obter uma lista de permissões em chaves do KMS, consulte [AWS KMS permissões](#).

Definir a política de chave ao criar uma chave do KMS

Ao usar a [CreateKey](#) operação para criar uma chave KMS, você pode usar seu Policy parâmetro para especificar uma política de chaves que conceda a uma conta externa, ou usuários e funções externos, permissão para usar a chave KMS.

Ao criar uma chave KMS no Console de gerenciamento da AWS, você também cria sua política de chaves. Ao selecionar identidades nas seções Key Administrators (Administradores de chaves) e Key Users (Usuários de chaves), o AWS KMS adiciona instruções de política para

essas identidades à política de chaves da chave do KMS. A seção Key Users (Usuários de chaves) também permite adicionar contas externas como usuários de chaves.

Quando você insere o ID da conta de uma conta externa, AWS KMS adiciona duas declarações à política principal. Essa ação afeta somente a política de chaves. Os usuários e funções na conta externa não poderão usar a chave do KMS até que você anexe as políticas do IAM para conceder a eles algumas dessas permissões ou todas elas.

A primeira instrução de política concede à conta externa permissão para usar a chave do KMS em operações de criptografia. A segunda declaração de política fundamental permite que a conta externa crie, visualize e revogue concessões na chave KMS, mas somente quando a solicitação vem de um [AWS serviço integrado](#) ao. AWS KMS Essas permissões permitem que outros AWS serviços que criptografam dados do usuário usem a chave KMS. Essas permissões são desenvolvidas para chaves do KMS que criptografam dados do usuário em serviços da AWS .

Etapa 2: Adicionar políticas do IAM à conta externa

A política de chaves na conta proprietária da chave do KMS define o intervalo válido de permissões. No entanto, os usuários e as funções na conta externa não poderão usar a chave do KMS até que você anexe as políticas do IAM que deleguem essas permissões ou use concessões para gerenciar o acesso à chave do KMS. As políticas do IAM são definidas na conta externa.

Se a política de chaves conceder permissão à conta externa, você poderá anexar as políticas do IAM a qualquer usuário ou função na conta. No entanto, se a política de chaves conceder permissão a usuários ou funções especificados, a política do IAM só poderá conceder essas permissões a todos ou a um subconjunto de usuários e funções especificados. Se uma política do IAM conceder acesso à chave do KMS para outros usuários ou funções externos, este não terá efeito.

A política de chaves também limita as ações na política do IAM. A política do IAM pode delegar todas as ações ou um subconjunto das ações especificadas na política de chaves. Se a política do IAM listar ações que não estão especificadas na política de chaves, essas permissões não terão efeito.

O seguinte exemplo de política do IAM permite que a entidade principal use a chave do KMS na conta 111122223333 para operações de criptografia. Para conceder essa permissão a usuários e funções na conta 444455556666, [associe a política](#) aos usuários ou às funções na conta 444455556666.

 Note

O exemplo a seguir representa uma amostra de política do IAM para o compartilhamento de uma chave do KMS com outra conta. Substitua os valores de exemplo Sid, Resource e Action por valores válidos para o uso pretendido de sua chave do KMS.

```
{
  "Sid": "AllowUseOfKeyInAccount111122223333",
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Observe os seguintes detalhes sobre essa política:

- Ao contrário de políticas de chaves, as instruções de políticas do IAM não contêm o elemento Principal. Nas políticas do IAM, a identidade principal é aquela à qual a política está anexada.
- O elemento Resource na política do IAM identifica a chave do KMS que a entidade principal pode usar. Para especificar uma chave do KMS, adicione seu [ARN de chave](#) ao elemento Resource.
- É possível especificar mais de uma chave do KMS no elemento Resource. No entanto, se você não especificar chaves do KMS específicas no elemento Resource, poderá conceder acesso acidentalmente a mais chaves do KMS do que o desejado.
- Para permitir que o usuário externo use a chave do KMS com serviços da [AWS que se integram ao AWS KMS](#), pode ser necessário adicionar permissões à política de chaves ou à política do IAM. Para obter detalhes, consulte [Permitindo o uso de chaves KMS externas com Serviços da AWS](#).

Para mais informações sobre como trabalhar com políticas do IAM, consulte [Políticas do IAM](#).

Permitindo o uso de chaves KMS externas com Serviços da AWS

Você pode conceder a um usuário em outra conta permissão para usar sua chave do KMS com um serviço integrado ao AWS KMS. Por exemplo, um usuário em uma conta externa pode usar sua chave do KMS para criptografar os objetos em um bucket do Amazon S3 ou para criptografar os segredos que armazena no AWS Secrets Manager.

A política de chaves deve conceder ao usuário externo ou à conta do usuário externo a devida permissão para usar a chave do KMS. Além disso, você precisa vincular políticas do IAM à identidade que concede permissões ao usuário para usar o AWS service (Serviço da AWS). O serviço também pode exigir que os usuários tenham permissões adicionais na política de chaves ou na política do IAM. Para obter uma lista das permissões AWS service (Serviço da AWS) exigidas em uma chave gerenciada pelo cliente, consulte o tópico *Proteção de dados* no capítulo *Segurança* do guia do usuário ou do guia do desenvolvedor do serviço.

Usar chaves do KMS em outras contas

Se você tiver permissão para usar uma chave KMS em outra Conta da AWS, poderá usar a chave KMS em Console de gerenciamento da AWS, AWS SDKs AWS CLI, e Ferramentas da AWS para PowerShell

Para identificar uma chave do KMS em uma conta diferente em um comando shell ou solicitação de API, use os seguintes [identificadores de chave](#).

- Para [operações criptográficas](#), e [DescribeKeyGetPublicKey](#), use o ARN da [chave ou o alias ARN](#) da chave KMS.
- Para [CreateGrant](#), [GetKeyRotationStatus](#), [ListGrants](#), e [RevokeGrant](#), use a chave ARN da chave KMS.

Se você inserir somente um ID de chave ou nome de alias, AWS presume que a chave KMS esteja na sua conta.

O AWS KMS console não exibe chaves KMS em outras contas, mesmo que você tenha permissão para usá-las. Além disso, as listas de chaves do KMS exibidas nos consoles de outros serviços da AWS não incluem chaves do KMS em outras contas.

Para especificar uma chave KMS em uma conta diferente no console de um AWS serviço, você deve inserir o ARN da chave ou o alias ARN da chave KMS. O identificador de chave necessário varia de

acordo com o serviço e pode diferir entre o console de serviço e suas operações de API. Para obter detalhes, consulte a documentação do serviço.

Controlar o acesso a chaves de várias regiões

Você pode usar chaves de várias regiões em cenários de conformidade, recuperação de desastres e backup que seriam mais complexos com chaves de uma única região. No entanto, como as propriedades de segurança das chaves de várias regiões são significativamente diferentes das de chaves de uma única região, convém ter cuidado ao autorizar a criação, o gerenciamento e o uso de chaves de várias regiões.

Note

Instruções de políticas do IAM existentes com caracteres curinga no campo Resource agora aplicam-se a chaves de região única e de várias regiões. Para restringi-las a chaves KMS de região única ou chaves de várias regiões, use a chave de condição [kms:MultiRegion](#).

Use suas ferramentas de autorização para impedir a criação e o uso de chaves de várias regiões em qualquer cenário em que uma única região seja suficiente. Permita que os diretores repliquem uma chave multirregional somente no Regiões da AWS que for necessário. Dê permissão para chaves de várias regiões apenas às entidades principais que precisam delas e apenas para tarefas necessárias.

Você pode usar políticas de chaves, políticas de IAM e concessões para permitir que os diretores do IAM gerenciem e usem chaves multirregionais em seu. Conta da AWS Cada chave de várias regiões é um recurso independente com um ARN de chave exclusivo e uma política de chaves. Você precisa estabelecer e manter uma política de chaves para cada chave e certificar-se de que as políticas do IAM novas e existentes implementem sua estratégia de autorização.

Para oferecer suporte a chaves multirregionais, AWS KMS usa uma função vinculada ao serviço IAM. Essa função dá ao AWS KMS as permissões necessárias para sincronizar [propriedades compartilhadas](#). Para obter mais informações, consulte [Autorizando a sincronização AWS KMS de chaves multirregionais](#).

Tópicos

- [Noções básicas de autorização para chaves de várias regiões](#)
- [Autorizar administradores e usuários de chave de várias regiões](#)

Noções básicas de autorização para chaves de várias regiões

Ao projetar políticas de chaves e políticas do IAM para chaves de várias regiões, considere os seguintes princípios.

- Política de chaves — Cada chave de várias regiões é um recurso de chave do KMS independente com sua própria [política de chaves](#). Você pode aplicar a mesma política de chaves ou uma política de chaves diferente a cada chave no conjunto de chaves de várias regiões relacionadas. As políticas de chaves não são [propriedades compartilhadas](#) de chaves multirregionais. AWS KMS não copia nem sincroniza políticas de chaves entre chaves multirregionais relacionadas.

Quando você cria uma chave de réplica no AWS KMS console, o console exibe a política de chaves atual da chave primária como uma conveniência. É possível usar essa política de chaves, editá-la ou excluí-la e substituí-la. Mas mesmo que você aceite a política de chave primária inalterada, AWS KMS não sincroniza as políticas. Por exemplo, se você alterar a política de chaves da chave primária, a política de chaves da chave de réplica permanecerá igual.

- Política de chaves padrão — Quando você cria chaves multirregionais usando as `ReplicateKey` operações [CreateKey](#), a [política de chaves padrão](#) é aplicada, a menos que você especifique uma política de chaves na solicitação. Essa é a mesma política de chave padrão aplicada a chaves de região única.
- Políticas do IAM — Assim como acontece com todas as chaves do KMS, você pode usar políticas do IAM para controlar o acesso a chaves de várias regiões somente quando a [política de chaves permite](#). [As políticas do IAM](#) se aplicam a todas as Regiões da AWS por padrão. No entanto, você pode usar chaves de condição, como [aws: RequestedRegion](#), para limitar as permissões a uma região específica.

Para criar chaves primárias e de réplica, as entidades principais devem ter a permissão `kms:CreateKey` em uma política do IAM aplicável à região na qual a chave é criada.

- Subsídios — os AWS KMS [subsídios](#) são regionais. Cada concessão dá permissões para uma chave do KMS. É possível usar concessões para dar permissões para uma chave primária ou chave de réplica de várias regiões. Porém, você não pode usar uma única concessão para dar permissões para várias chaves do KMS, mesmo que elas sejam chaves de várias regiões relacionadas.
- ARN de chave — Cada chave de várias regiões tem um [ARN de chave](#). A chave ARNs das chaves multirregionais relacionadas tem a mesma partição, conta e ID de chave, mas regiões diferentes.

Para aplicar uma instrução de política do IAM a uma chave de várias regiões específica, use seu ARN de chave ou um padrão de ARN de chave que inclua a região. Para aplicar uma instrução de política do IAM a todas as chaves de várias regiões relacionadas, use um caractere curinga (*) no elemento Region do ARN, como mostra o exemplo abaixo.

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

Para aplicar uma declaração de política a todas as chaves multirregionais em sua Conta da AWS, você pode usar a condição [kms: MultiRegion](#) policy ou um padrão de ID de chave que inclua o prefixo distintomrk-.

- Função vinculada ao serviço — [Os diretores que criam chaves primárias multirregionais devem ter a permissão iam: CreateServiceLinkedRole](#)

[Para sincronizar as propriedades compartilhadas das chaves multirregionais relacionadas, AWS KMS assume uma função vinculada ao serviço do IAM.](#) AWS KMS cria a função vinculada ao serviço Conta da AWS sempre que você cria uma chave primária multirregional. (Se a função existir, o AWS KMS a recriará, o que não causa nenhum problema.) A função é válida em todas as regiões. AWS KMS [Para permitir a criação \(ou recriação\) da função vinculada ao serviço, os diretores que criam chaves primárias multirregionais devem ter a permissão iam: CreateServiceLinkedRole](#)

Autorizar administradores e usuários de chave de várias regiões

As entidades principais que criam e gerenciam chaves de várias regiões precisam das seguintes permissões nas regiões primária e de réplica:

- kms:CreateKey
- kms:ReplicateKey

- kms:UpdatePrimaryRegion
- iam:CreateServiceLinkedRole

Criar uma chave primária

Para [criar uma chave primária multirregional](#), o principal precisa das CreateServiceLinkedRole permissões [kms: CreateKey](#) e [iam:](#) em uma política do IAM que seja efetiva na região da chave primária. As entidades principais que têm essas permissões podem criar chaves de região única e de várias regiões, a menos que você restrinja suas permissões.

A iam:CreateServiceLinkedRole permissão permite criar AWS KMS a [AWSServiceRoleForKeyManagementServiceMultiRegionKeysfunção](#) para sincronizar as [propriedades compartilhadas das chaves](#) multirregionais relacionadas.

Por exemplo, essa política do IAM permite que uma entidade principal crie chaves multirregionais, anexe políticas para essas chaves e perfis vinculados ao serviço para as chaves multirregionais.

JSON

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

Para permitir ou negar a permissão para criar chaves primárias multirregionais, use a chave de MultiRegion condição [kms:](#). Os valores válidos são true (chave de várias regiões) ou false (chave de região única). Por exemplo, a instrução de política do IAM a seguir usa uma ação Deny com a chave de condição kms:MultiRegion para impedir que as entidades principais criem chaves de várias regiões.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Action": "kms:CreateKey",
    "Effect": "Deny",
    "Resource": "*",
    "Condition": {
      "Bool": {
        "kms:MultiRegion": true
      }
    }
  }
}
```

Replicação de chaves

Para [criar uma chave de réplica de várias regiões](#), a entidade principal precisará das seguintes permissões:

- [kms: ReplicateKey](#) permissão na política de chaves da chave primária.
- [kms: CreateKey](#) permissão em uma política do IAM que é efetiva na região da chave de réplica.

Tenha cuidado ao conceder essas permissões. Elas permitem que as entidades principais criem chaves do KMS e as políticas de chaves que autorizam seu uso. A permissão `kms:ReplicateKey` também autoriza a transferência de material chave através dos limites da região no AWS KMS.

Para restringir o modo Regiões da AWS em que uma chave multirregional pode ser replicada, use a chave de condição [kms: ReplicaRegion](#). Ela limita apenas a permissão `kms:ReplicateKey`. De outra forma, ela não terá efeito. Por exemplo, a política de chaves a seguir permite que a entidade principal replique essa chave primária, mas somente nas regiões especificadas.

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

Atualizar a região primária

As entidades principais autorizadas podem converter uma chave de réplica em uma chave primária, o que transforma a chave primária anterior em uma réplica. Essa ação é conhecida como [atualização da região primária](#). Para atualizar a região principal, o principal precisa de [kms:UpdatePrimaryRegion](#) permissão em ambas as regiões. Você pode fornecer essas permissões em uma política de chaves ou política do IAM.

- kms:UpdatePrimaryRegion na chave primária. Essa permissão deve ser efetiva na região da chave primária.
- kms:UpdatePrimaryRegion na chave de réplica. Essa permissão deve ser efetiva na região da chave de réplica.

Por exemplo, a política de chaves a seguir dá aos usuários que podem assumir a função Administrador permissão para atualizar a região primária da chave do KMS. Essa chave do KMS pode ser a chave primária ou uma chave de réplica nessa operação.

```
{
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:UpdatePrimaryRegion"
}
```

Para restringir o Regiões da AWS que pode hospedar uma chave primária, use a chave de PrimaryRegion condição [kms:](#). Por exemplo, a declaração de política do IAM a seguir permite que

os diretores atualizem a região primária das chaves multirregionais no Conta da AWS, mas somente quando a nova região primária for uma das regiões especificadas.

```
{
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Resource": {
    "arn:aws:kms*:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": {
      "kms:PrimaryRegion": [
        "us-west-2",
        "sa-east-1",
        "ap-southeast-1"
      ]
    }
  }
}
```

Usar e gerenciar chaves de várias regiões

Por padrão, as entidades principais que têm permissão para usar e gerenciar chaves do KMS em uma Conta da AWS e região também têm permissão para usar e gerenciar chaves de várias regiões. No entanto, você pode usar a chave de MultiRegion condição [kms:](#) para permitir somente chaves de região única ou somente chaves de várias regiões. Ou use a chave de MultiRegionKeyType condição [kms:](#) para permitir somente chaves primárias multirregionais ou somente chaves de réplica. Ambas as chaves de condição controlam o acesso à [CreateKey](#) operação e a qualquer operação que use uma chave KMS existente, como [Criptografar ou. EnableKey](#)

A instrução de política do IAM a seguir usa a chave de condição `kms:MultiRegion` para impedir que as entidades principais usem ou gerenciem qualquer chave de várias regiões.

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

Esse exemplo de instrução de política do IAM usa a condição `kms:MultiRegionKeyType` para permitir que as entidades principais programem e cancelem a exclusão de chaves, mas somente em chaves de réplica de várias regiões.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": {
    "arn:aws:kms:us-west-2:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": "kms:MultiRegionKeyType": "REPLICA"
  }
}
```

Determinando o acesso ao AWS KMS keys

Para determinar a extensão total de quem ou o que atualmente tem acesso a uma AWS KMS key, você deve examinar a política principal da chave KMS, todas as [concessões](#) que se aplicam à chave KMS e, potencialmente, todas as políticas AWS Identity and Access Management (IAM). Você pode fazer isso para determinar o escopo do uso potencial de uma chave do KMS ou para ajudar a cumprir os requisitos de conformidade ou de auditoria. Os tópicos a seguir podem ajudar você a gerar uma lista completa das entidades principais da AWS (identidades) que no momento têm acesso a uma chave do KMS.

Tópicos

- [Examinar a política de chaves](#)
- [Examinar políticas do IAM](#)
- [Examinar concessões](#)

Examinar a política de chaves

[Políticas de chaves](#) são a principal maneira de controlar o acesso a chaves do KMS. Cada chave do KMS tem exatamente uma política de chaves.

Quando uma política de chaves consiste em ou inclui a [política de chaves padrão](#), ela permite que os administradores do IAM na conta usem políticas do IAM para controlar o acesso à chave do KMS. Além disso, se a política de chaves conceder a [outra Conta da AWS](#) permissão para usar a chave do KMS, os administradores do IAM na conta externa poderão usar políticas do IAM para delegar essas permissões. Para determinar a lista completa de entidades principais que podem acessar a chave do KMS, [examine as políticas do IAM](#).

Para ver a política de chaves de uma [chave gerenciada pelo AWS KMS cliente](#) ou [Chave gerenciada pela AWS](#) em sua conta, use a operação Console de gerenciamento da AWS ou a [GetKeyPolicy](#) operação na AWS KMS API. Para visualizar a política de chaves, é necessário ter permissões `kms:GetKeyPolicy` para a chave do KMS. Para obter instruções sobre como visualizar a política de chaves de uma chave do KMS, consulte [the section called “Visualizar uma política de chave”](#).

Examine o documento de política de chaves e anote todas as principais especificadas em cada elemento `Principal` da declaração de política. Em uma declaração de política com `Allow` efeito, os usuários do IAM, as funções do IAM e Contas da AWS o `Principal` elemento têm acesso a essa chave do KMS.

Note

Não defina a entidade principal como um asterisco (*) em qualquer instrução de política de chave que permita permissões, a menos que você utilize [condições](#) para limitar a política de chave. Um asterisco dá a cada identidade em cada Conta da AWS permissão para usar a chave KMS, a menos que outra declaração de política a negue explicitamente. Usuários de outros usuários Contas da AWS podem usar sua chave KMS sempre que tiverem permissões correspondentes em suas próprias contas.

Os exemplos a seguir usam as declarações de política encontradas na [política de chaves padrão](#) para demonstrar como fazer isso.

Example Declaração de política 1

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

```
"Resource": "*"
}
```

Na declaração de política 1, `arn:aws:iam::111122223333:root` é um [principal de AWS conta](#) que se refere ao Conta da AWS 111122223333. (Não corresponde ao usuário raiz da conta.) Por padrão, uma declaração de política como essa é incluída no documento de política de chaves quando você cria uma nova chave KMS com o Console de gerenciamento da AWS, ou cria uma nova chave KMS programaticamente, mas não fornece uma política de chaves.

Um documento de política de chaves com uma declaração que permite o acesso às [políticas de Conta da AWS habilitação do IAM na conta para permitir o acesso à chave KMS](#). Isso significa que os usuários e perfis na conta podem ter acesso à chave do KMS, mesmo se não estiverem explicitamente listados como entidades principais no documento de política de chaves. [Examine todas as políticas do IAM](#) em todas as Contas da AWS listadas como principais para determinar se elas permitem acesso a essa chave do KMS.

Example Declaração de política 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": "*"
}
```

Na declaração de política 2, `arn:aws:iam::111122223333:role/KMSKeyAdmins` refere-se à função do IAM chamada KMSKey Admins em Conta da AWS 111122223333. Os usuários

autorizados a assumir esse perfil podem executar as ações listadas na instrução de política, que são as ações administrativas para gerenciar uma chave do KMS.

Example Declaração de política 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": "*"
}
```

Na declaração de política 3, `arn:aws:iam::111122223333:role/EncryptionApp` refere-se à função do IAM nomeada `EncryptionApp` em Conta da AWS 111122223333. As entidades principais autorizadas a assumir esse perfil podem executar as ações listadas na instrução de política, que incluem as [operações criptográficas](#) para uma chave do KMS de criptografia simétrica.

Example Declaração de política 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}}
}
```

Na declaração de política 4, `arn:aws:iam::111122223333:role/EncryptionApp` refere-se à função do IAM nomeada `EncryptionApp` em Conta da AWS 111122223333. As entidades principais autorizadas assumem esse perfil e podem executar as ações listadas na instrução de política. Essas

ações, quando combinadas com as ações permitidas na Instrução de política de exemplo 3, são aquelas necessárias para delegar o uso da chave do KMS à maioria dos [serviços da AWS integrados ao AWS KMS](#), especificamente os serviços que usam [concessões](#). O `GrantIsFor AWSResource` valor `kms:` no `Condition` elemento garante que a delegação seja permitida somente quando o delegado é um AWS serviço que se integra AWS KMS e usa concessões para autorização.

Para conhecer todas as diferentes maneiras de especificar uma entidade principal em um documento de política de chaves, consulte [Especificar uma entidade principal](#), no Manual do usuário do IAM.

Para saber mais sobre as AWS KMS principais políticas, consulte [Políticas-chave em AWS KMS](#).

Examinar políticas do IAM

Além da política de chaves e concessões, também é possível usar as [políticas do IAM](#) para permitir o acesso a uma chave do KMS. Para obter mais informações sobre como as políticas do IAM e as políticas de chaves funcionam juntas, consulte [Solução de problemas de AWS KMS permissões](#).

Para determinar quais entidades principais têm acesso no momento a uma chave do KMS por meio de políticas do IAM, você pode usar a ferramenta baseada em navegador [Simulador de políticas do IAM](#), ou pode fazer solicitações para a API do IAM.

Maneiras de examinar políticas do IAM

- [Examinar políticas do IAM com o Simulador de políticas do IAM](#)
- [Examinar políticas do IAM com a API do IAM](#)

Examinar políticas do IAM com o Simulador de políticas do IAM

O Simulador de políticas do IAM pode ajudar a saber quais entidades principais têm acesso a uma chave do KMS por meio de uma política do IAM.

Para usar o simulador de políticas do IAM para determinar o acesso a uma chave do KMS

1. Faça login no Console de gerenciamento da AWS e abra o IAM Policy Simulator em <https://policysim.aws.amazon.com/>.
2. No painel Users, Groups, and Roles (Usuários, grupos e funções), escolha o usuário, o grupo ou a função cujas políticas você deseja simular.
3. (Opcional) Desmarque a caixa de seleção ao lado de qualquer política que você deseja omitir da simulação. Para simular todas as políticas, deixe todas as políticas selecionadas.

4. No painel Policy Simulator (Simulador de políticas), faça o seguinte:
 - a. Para Select service (Selecionar serviço), selecione Key Management Service.
 - b. Para simular AWS KMS ações específicas, em Selecionar ações, escolha as ações a serem simuladas. Para simular todas as AWS KMS ações, escolha Selecionar tudo.
5. (Opcional) O Simulador de políticas simula o acesso a todas as chaves do KMS por padrão. Para simular o acesso a uma determinada chave do KMS, escolha Simulation Settings (Configurações de simulação) e digite o nome do recurso da Amazon (ARN) da chave do KMS a ser simulada.
6. Selecione Run Simulation (Executar simulação).

Você pode visualizar os resultados da simulação na seção Results (Resultados). Repita as etapas 2 a 6 para cada usuário, grupo e perfil na Conta da AWS.

Examinar políticas do IAM com a API do IAM

Você pode usar a API do IAM para examinar políticas do IAM programaticamente. As etapas a seguir fornecem uma visão geral de como fazer isso:

1. Para cada um Conta da AWS listado como principal na política de chaves (ou seja, cada [principal de AWS conta](#) especificado neste formato: "Principal": {"AWS": "arn:aws:iam::111122223333:root"}), use as [ListRoles](#) operações [ListUsers](#) na API do IAM para obter todos os usuários e funções na conta.
2. Para cada usuário e função na lista, use a [SimulatePrincipalPolicy](#) operação na API IAM, transmitindo os seguintes parâmetros:
 - Para PolicySourceArn, especifique o Amazon Resource Name (ARN) de um usuário ou função da sua lista. É possível especificar somente um PolicySourceArn para cada solicitação SimulatePrincipalPolicy, portanto, você deve chamar essa operação diversas vezes, uma vez para cada perfil e usuário em sua lista.
 - Para a ActionNames lista, especifique cada ação AWS KMS da API a ser simulada. Para simular todas as ações AWS KMS da API, use kms : *. Para testar ações de AWS KMS API individuais, preceda cada ação de API com "kms : ", por exemplo, "kms : ListKeys". Para obter uma lista completa das ações de API do AWS KMS , consulte [Actions](#) (Ações) na Referência de API do AWS Key Management Service .
 - (Opcional) Para determinar se os usuários ou funções têm acesso a chaves KMS específicas, use o ResourceArns parâmetro para especificar uma lista dos nomes de recursos da Amazon

(ARNs) das chaves do KMS. Para determinar se os usuários ou perfis têm acesso a qualquer chave do KMS, omita o parâmetro `ResourceArns`.

O IAM responde a cada solicitação `SimulatePrincipalPolicy` com uma decisão de avaliação: `allowed`, `explicitDeny` ou `implicitDeny`. Para cada resposta que contém uma decisão de avaliação `deallowed`, a resposta inclui o nome da operação de AWS KMS API específica que é permitida. Ela também inclui o ARN da chave do KMS que foi usada na avaliação, se houver.

Examinar concessões

As concessões são mecanismos avançados para especificar permissões que você ou um AWS serviço integrado AWS KMS podem usar para especificar como e quando uma chave KMS pode ser usada. Concessões são associadas a uma chave do KMS, e cada concessão contém a entidade principal que recebe permissão para usar a chave do KMS e uma lista de operações que são permitidas. As concessões são uma alternativa para a política de chaves e são úteis para casos de uso específicos. Para obter mais informações, consulte [Subsídios em AWS KMS](#).

Para obter uma lista de concessões para uma chave KMS, use a AWS KMS [ListGrants](#) operação. É possível examinar as concessões de uma chave do KMS para determinar quem ou o quê tem acesso no momento para usar a chave do KMS por meio dessas concessões. Por exemplo, a seguir há uma representação JSON de uma concessão que foi obtida do comando [list-grants](#) na 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"}}
}]}
```

Para saber quem ou o quê tem acesso para usar a chave do KMS, procure o elemento `"GranteePrincipal"`. No exemplo anterior, o principal favorecido é um usuário de função assumida associado à instância do EC2 `i-5d476fab`. A infraestrutura do EC2 usa essa função

para anexar o volume do EBS criptografado vol-5cccfb4e à instância. Nesse caso, a função da infraestrutura do EC2 tem permissão para usar a chave do KMS porque você já criou um volume do EBS criptografado protegido por essa chave do KMS. Anexe o volume a uma instância do EC2.

A seguir há outro exemplo de representação JSON de uma concessão que foi obtida do comando [list-grants](#) na AWS CLI. No exemplo a seguir, o principal beneficiário é outro. Conta da 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
}]}
```

Contexto de criptografia

Note

Não é possível especificar um contexto de criptografia em uma operação criptográfica com uma [chave do KMS assimétrica](#) ou uma [chave do KMS de Hash-based message authentication code \(HMAC – Código de autenticação de mensagem por hash\)](#). Algoritmos assimétricos e algoritmos de Message authentication code (MAC – Código de autenticação de mensagem) não são compatíveis com um contexto de criptografia.

Todas as [operações AWS KMS criptográficas](#) com [chaves KMS de criptografia simétrica](#) aceitam um contexto de criptografia, um conjunto opcional de pares de chave-valor não secretos que podem conter informações contextuais adicionais sobre os dados. Você pode inserir o contexto de criptografia nas Encrypt operações AWS KMS para aprimorar a autorização e a auditabilidade de suas chamadas de descriptografia da AWS KMS API. AWS KMS usa o contexto de criptografia como dados autenticados adicionais (AAD) para oferecer suporte à criptografia autenticada. O contexto de criptografia é associado de maneira criptográfica ao texto cifrado, de modo que o mesmo contexto de criptografia é necessário para descriptografar os dados.

O contexto de criptografia não é secreto nem criptografado. Ele é exibido em texto simples em [Logs do AWS CloudTrail](#), para você possa usá-lo para identificar e categorizar suas operações de criptografia. Seu contexto de criptografia não deve incluir informações confidenciais. Recomendamos que o seu contexto de criptografia descreva os dados que estão sendo criptografados ou descriptografados. Por exemplo, quando você criptografar um arquivo, poderá usar parte do caminho do arquivo como contexto de criptografia.

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

Por exemplo, ao criptografar volumes e snapshots criados com a operação [Amazon Elastic Block Store](#) (Amazon EBS) [CreateSnapshot](#), o Amazon EBS usa o ID do volume como valor do contexto de criptografia.

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

Você também pode usar o contexto de criptografia para refinar ou limitar o acesso à AWS KMS keys sua conta. Você pode usar o contexto de criptografia [como uma restrição em concessões](#) e como uma [condição nas declarações de política](#). As chaves de contexto de criptografia e seus valores podem ser strings arbitrárias com aws. Esses valores contrastam com as [tags geradas pela AWS](#), como [aws:cloudformation:stack-name](#). Para obter mais informações, consulte [kms:EncryptionContext: chave de contexto](#).

Para saber como usar o contexto de criptografia para proteger a integridade dos dados criptografados, consulte [a postagem Como proteger a integridade de seus dados criptografados usando AWS Key Management Service e EncryptionContext](#) no blog AWS de segurança.

Regras de contexto de criptografia

AWS KMS impõe as seguintes regras para chaves e valores de contexto de criptografia.

- A chave e o valor em um par de contexto de criptografia devem ser strings literais simples. Se você usar um tipo diferente, como um inteiro ou flutuante, o AWS KMS o interpretará como uma string.
- As chaves e valores em um contexto de criptografia podem incluir caracteres Unicode. Se um contexto de criptografia incluir caracteres que não são permitidos em políticas de chaves ou

políticas do IAM, você não poderá especificar o contexto de criptografia em chaves de condição de política, como [kms:EncryptionContext:context-key](#) e [kms:EncryptionContextKeys](#). Para obter detalhes sobre as regras de documento de política de chaves, consulte [Formato de política de chaves](#). Para obter detalhes sobre as regras de documento de política do IAM, consulte [Requisitos de nome do IAM](#) no Guia do usuário do IAM.

Contexto de criptografia em políticas

O contexto de criptografia é usado principalmente para verificar a integridade e a autenticidade. No entanto, também é possível usar o contexto de criptografia para controlar o acesso a AWS KMS keys de criptografia simétrica em políticas de chave e em políticas do IAM.

As chaves de EncryptionContextKeys condição [kmsEncryptionContext::](#) e [kms:](#) permitem (ou negam) uma permissão somente quando a solicitação inclui chaves de contexto de criptografia específicas ou pares de valores-chave.

Por exemplo, a instrução de política de chave a seguir permite que a função RoleForExampleApp use a chave do KMS em operações Decrypt. Ela usa a chave da condição kms:EncryptionContext:context-key para conceder essa permissão somente quando o contexto de criptografia na solicitação inclui um par de contexto de criptografia AppName:ExampleApp.

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

Para obter mais informações sobre essas chaves de condição de contexto de criptografia, consulte [Chaves de condição para AWS KMS](#).

Contexto de criptografia em concessões

Ao [criar uma concessão](#), você pode incluir [restrições de concessão](#) que estabeleçam condições para as permissões de concessão. AWS KMS suporta duas restrições de concessão `EncryptionContextEquals` e `EncryptionContextSubset` ambas envolvem o [contexto de criptografia](#) em uma solicitação de uma operação criptográfica. Quando você usa essas restrições de concessão, as permissões na concessão são efetivas apenas quando o contexto de criptografia na solicitação para a operação criptográfica atende aos requisitos das restrições de concessão.

Por exemplo, você pode adicionar uma restrição de `EncryptionContextEquals` concessão a uma concessão que permita a [GenerateDataKey](#) operação. Com essa restrição, a concessão permite a operação apenas quando o contexto de criptografia na solicitação é uma correspondência com maiúsculas e minúsculas ao contexto de criptografia na restrição de concessão.

```
$ 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}
```

Uma solicitação como a seguinte da entidade principal receptora da concessão atenderia à restrição `EncryptionContextEquals`.

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

Para obter detalhes sobre as restrições de concessão, consulte [Usar restrições de concessão](#). Para obter informações detalhadas sobre concessões, consulte [the section called “Concessões”](#).

Registrar em log o contexto de criptografia

AWS KMS usa AWS CloudTrail para registrar o contexto de criptografia para que você possa determinar quais chaves e dados do KMS foram acessados. A entrada de log mostra exatamente quais chaves do KMS foram usada para criptografar ou descriptografar dados específicos referenciados pelo contexto de criptografia na entrada de log.

 Important

Como o contexto de criptografia é registrado em log, ele não deve conter informações confidenciais.

Armazenar o contexto de criptografia

Para simplificar o uso de qualquer contexto de criptografia quando você chama as operações [Decrypt](#) ou [ReEncrypt](#), é possível armazenar o contexto de criptografia com os dados criptografados. Recomendamos que você armazene apenas o suficiente do contexto de criptografia para ajudar a criar o contexto de criptografia por completo quando você precisar dele para criptografia ou descriptografia.

Por exemplo, se o contexto de criptografia é o caminho para um arquivo, armazene apenas parte desse caminho com o conteúdo do arquivo criptografado. Quando você precisar do contexto de criptografia completo, reconstrua-o do fragmento armazenado. Se alguém tentar violar o arquivo, como renomear ou mover para um local diferente, o valor do contexto de criptografia será alterado e a solicitação de descriptografia falhará.

Testar suas permissões

Para usar AWS KMS, você precisa ter credenciais que AWS possam ser usadas para autenticar suas solicitações de API. As credenciais devem incluir permissões para acessar chaves KMS e aliases. As permissões são determinadas pelas política de chave, políticas do IAM, concessões e controles de acesso entre contas. Além de controlar o acesso às chaves KMS, é possível controlar o acesso ao seu CloudHSM e aos seus repositórios de chaves personalizados.

É possível especificar o parâmetro da API `DryRun` para verificar se você tem as permissões necessárias para usar as chaves do AWS KMS. Você também pode usar `DryRun` para verificar se os parâmetros de solicitação em uma chamada de AWS KMS API estão especificados corretamente.

Tópicos

- [Qual é o DryRun parâmetro?](#)
- [Especificando DryRun com a API](#)

Qual é o DryRun parâmetro?

DryRun é um parâmetro de API opcional que você especifica para verificar se as chamadas da API do AWS KMS serão bem-sucedidas. Use DryRun para testar sua chamada de API, antes de realmente fazer a chamada para o AWS KMS. É possível verificar o seguinte:

- Que você tem as permissões necessárias para usar as chaves do AWS KMS .
- Que você especificou os parâmetros na chamada corretamente.

AWS KMS suporta o uso do DryRun parâmetro em determinadas ações da API:

- [CreateGrant](#)
- [Decrypt](#)
- [DeriveSharedSecret](#)
- [Encrypt](#)
- [GenerateDataKey](#)
- [GenerateDataKeyPair](#)
- [GenerateDataKeyPairWithoutPlaintext](#)
- [GenerateDataKeyWithoutPlaintext](#)
- [GenerateMac](#)
- [ReEncrypt](#)
- [RetireGrant](#)
- [RevokeGrant](#)
- [Sign](#)
- [Verificar](#)
- [VerifyMac](#)

O uso do parâmetro DryRun incorrerá em cobranças e será cobrado como uma solicitação de API padrão. Para obter mais informações sobre AWS KMS preços, consulte [AWS Key Management Service Preços](#).

Todas as solicitações de API que usam o parâmetro DryRun se aplicam à cota de solicitações da API e podem resultar em uma exceção de controle de utilização se você exceder a cota de

solicitação da API. Por exemplo, chamar [Decrypt](#) com DryRun ou sem DryRun conta na mesma cota de operações criptográficas. Para saber mais, consulte [Solicitações de AWS KMS limitação](#).

Cada chamada para uma operação de AWS KMS API é capturada como um evento e registrada em um AWS CloudTrail registro. A saída de qualquer operação que especifique o DryRun parâmetro aparece no seu CloudTrail registro. Para obter mais informações, consulte [Registrando chamadas de AWS KMS API com AWS CloudTrail](#).

Especificando DryRun com a API

Para usar DryRun, especifique o `--dry-run` parâmetro em AWS CLI comandos e chamadas de AWS KMS API compatíveis com o parâmetro. Ao fazer isso, AWS KMS verificará se sua chamada será bem-sucedida. AWS KMS as chamadas usadas sempre falharão e DryRun retornarão uma mensagem com informações sobre o motivo pelo qual a chamada falhou. A mensagem pode incluir as seguintes exceções:

- `DryRunOperationException` - A solicitação seria bem-sucedida se DryRun não estivesse especificada.
- `ValidationException` - A solicitação falhou devido à especificação de um parâmetro de API incorreto.
- `AccessDeniedException` - Você não tem permissões para realizar a ação de API especificada no recurso KMS.

Por exemplo, o comando a seguir usa a [CreateGrant](#) operação e cria uma concessão que permite que os usuários autorizados a assumir a `keyUserRole` função chamem a operação [Decrypt](#) em uma chave KMS [simétrica](#) especificada. O parâmetro DryRun está especificado.

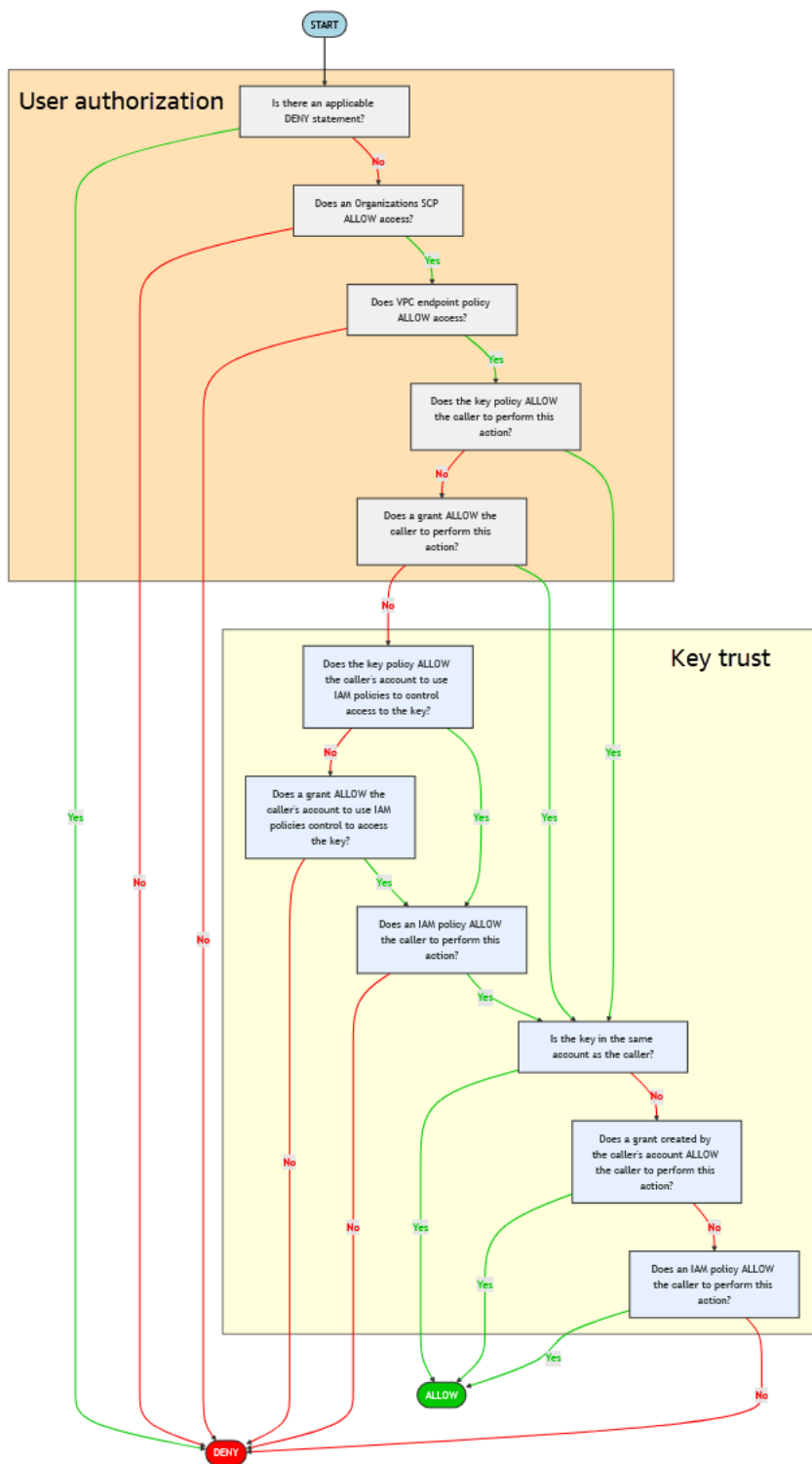
```
$ aws kms create-grant \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
  --operations Decrypt \
  --dry-run
```

Solução de problemas de AWS KMS permissões

Ao autorizar o acesso a uma chave KMS, AWS KMS avalia o seguinte:

- A [política de chaves](#) associada à chave do KMS. A política de chaves é sempre definida na Conta da AWS região proprietária da chave KMS.
- Todas as [políticas do IAM](#) que são vinculadas ao usuário ou ao perfil que realiza a solicitação. As políticas do IAM do que controlam o uso de uma chave do KMS pela entidade principal sempre são definidas na Conta da AWS da entidade principal.
- Todas as [concessões](#) que se aplicam à chave do KMS.
- Outros tipos de políticas que podem se aplicar à solicitação para usar a chave do KMS, como [políticas de controle de serviço do AWS Organizations](#) e [políticas de endpoint da VPC](#). Essas políticas são opcionais e permitem todas as ações por padrão, mas é possível usá-las para restringir permissões concedidas de outra forma às entidades principais.

AWS KMS avalia esses mecanismos de política em conjunto para determinar se o acesso à chave KMS é permitido ou negado. Para fazer isso, AWS KMS usa um processo semelhante ao descrito no fluxograma a seguir. O fluxograma a seguir fornece uma representação visual do processo de avaliação de política.



Esse fluxograma está dividido em duas partes. As partes parecem ser sequenciais, mas geralmente são avaliadas ao mesmo tempo.

- A autorização de uso determina se você tem permissão para usar uma chave do KMS com base em sua política de chaves, políticas do IAM, concessões e outras políticas aplicáveis.
- A confiança de chave determina se você deve confiar em uma chave do KMS que tem autorização para usar. Em geral, você confia nos recursos do seu Conta da AWS. Mas você também pode se sentir confiante em usar as chaves KMS de outra forma Conta da AWS se uma concessão ou política do IAM em sua conta permitir que você use a chave KMS.

Use esse fluxograma para descobrir por que um autor da chamada recebeu ou não permissão para usar uma chave do KMS. Ou use-o para avaliar suas políticas e concessões. Por exemplo, o fluxograma mostra que um autor de chamada pode ter acesso negado por uma instrução DENY explícita ou pela ausência de uma instrução ALLOW explícita na política de chaves, política do IAM ou concessão.

O fluxograma consegue explicar alguns cenários comuns de permissão.

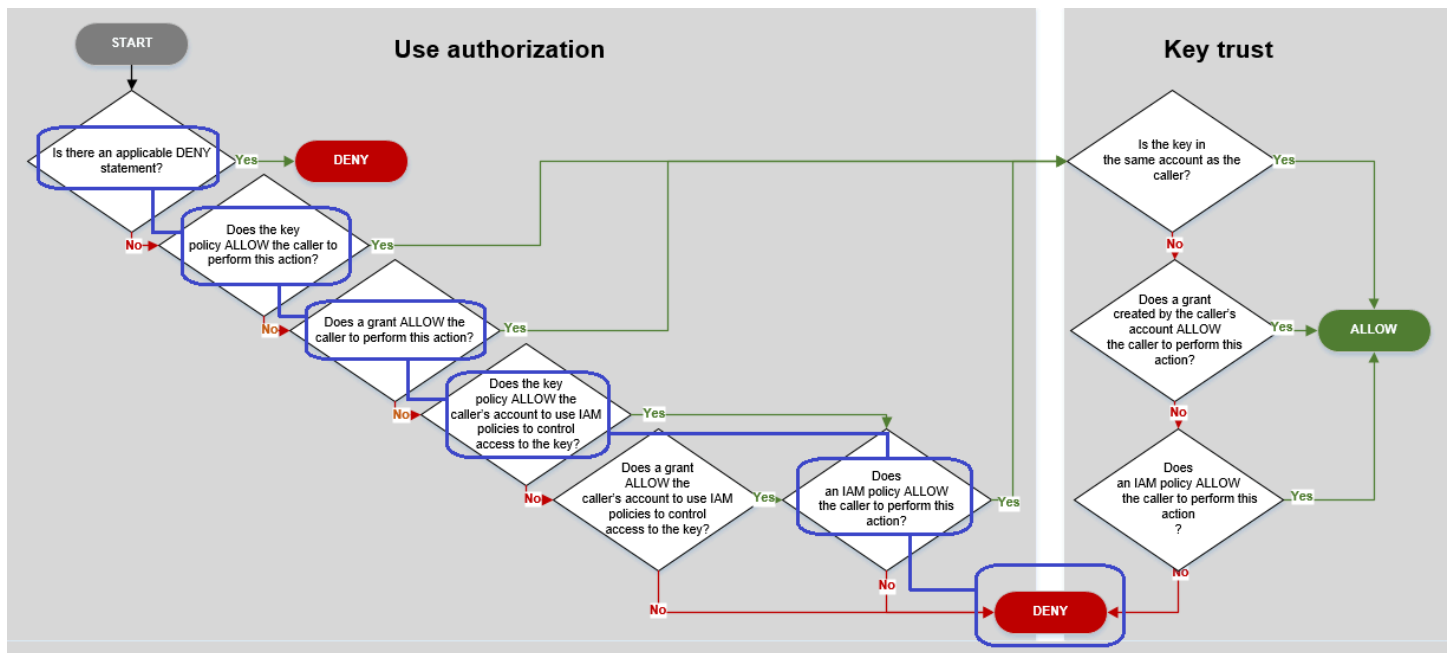
Exemplos de permissão

- [Exemplo 1: O acesso a uma chave KMS é negado ao usuário em seu Conta da AWS](#)
- [Exemplo 2: O usuário assume a função com permissão para usar uma chave KMS em uma outra Conta da AWS](#)

Exemplo 1: O acesso a uma chave KMS é negado ao usuário em seu Conta da AWS

Alice é usuária do IAM no Conta da AWS 111122223333. Ela teve o acesso negado a uma chave do KMS na mesma Conta da AWS. Por que Alice não pode usar a chave do KMS?

Nesse caso, Alice teve o acesso negado à chave do KMS porque não há uma política de chaves, política do IAM ou concessão que dê a ela as permissões necessárias. A política de chaves da chave KMS permite o uso de políticas do IAM Conta da AWS para controlar o acesso à chave do KMS, mas nenhuma política do IAM dá permissão a Alice para usar a chave do KMS.



Considere as políticas relevantes para este exemplo.

- A chave do KMS que Alice deseja usar tem a [política de chaves padrão](#). Essa política [permite que a Conta da AWS](#) que possui a chave do KMS use as políticas do IAM para controlar o acesso à chave do KMS. Esta política de chaves atende à condiçãoA política de chaves PERMITE que a conta dos chamadores usem políticas do IAM para controlar o acesso à chave? no fluxograma.

JSON

```

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

- No entanto, nenhuma política de chaves, política do IAM ou concessão permite que Alice use a chave do KMS. Portanto, é negada à Alice a permissão para usar a chave do KMS.

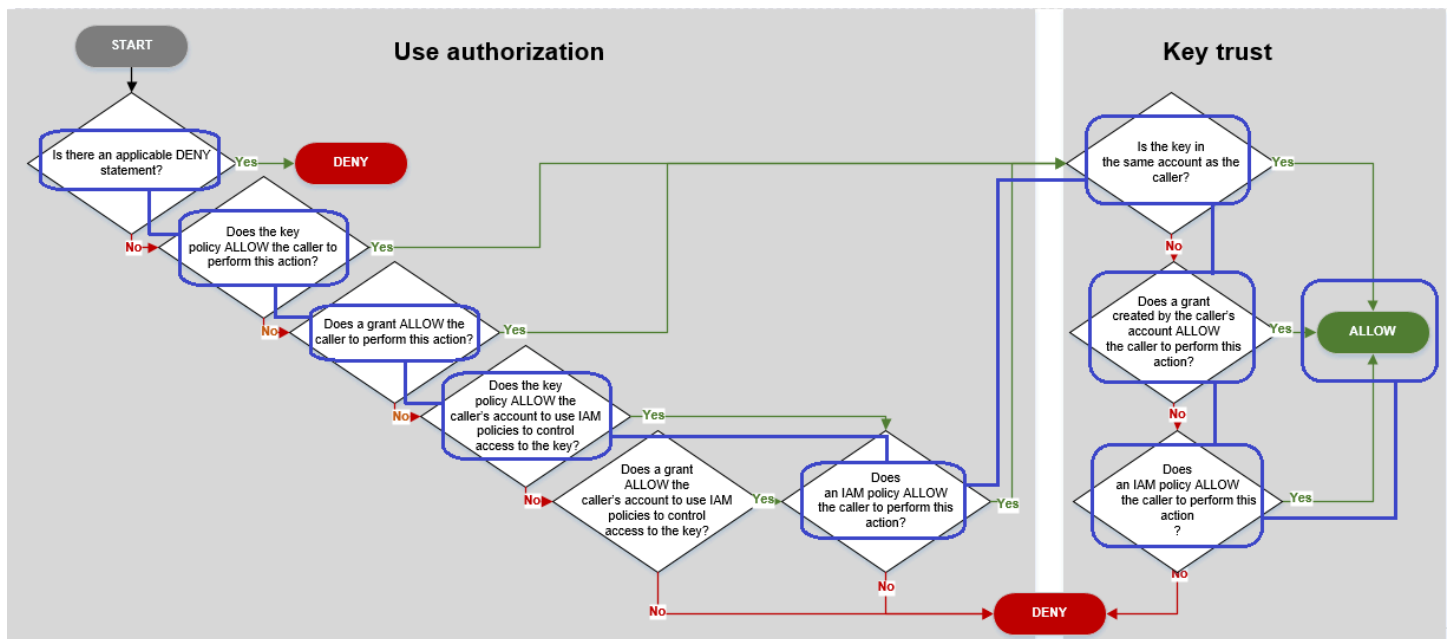
Exemplo 2: O usuário assume a função com permissão para usar uma chave KMS em uma outra Conta da AWS

Bob é um usuário na conta 1 (111122223333). Ele tem permissão para usar uma chave do KMS na conta 2 (444455556666) em [operações de criptografia](#). Como isso é possível?

Tip

Ao avaliar permissões entre contas, lembre-se de que a política de chaves é especificada na conta da chave do KMS. A política do IAM é especificada na conta do autor da chamada, mesmo quando ele está em uma conta diferente. Para obter detalhes sobre como fornecer acesso entre contas a chaves do KMS, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

- A política de chaves da chave do KMS permite que a conta 2 use políticas do IAM para controlar o acesso à chave do KMS.
- A política de chaves da chave do KMS na conta 2 permite que a conta 1 use a chave do KMS em operações de criptografia. No entanto, a conta 1 deve usar políticas do IAM para conceder às suas entidades principais acesso à chave do KMS.
- Uma política do IAM na conta 1 permite que a função Engineering use a chave do KMS na conta 2 para operações de criptografia.
- Bob, um usuário na conta 1, tem permissão para assumir a função Engineering.
- Bob pode confiar nessa chave do KMS, pois, mesmo que ela não esteja na conta dele, uma política do IAM na conta dele concede uma permissão explícita para usar essa chave do KMS.



Considere as políticas que permitem que Bob, um usuário na conta 1, use a chave do KMS na conta 2.

- A política de chaves da chave do KMS permite que a conta 2 (444455556666, a conta que possui a chave do KMS) use políticas do IAM para controlar o acesso à chave do KMS. Essa política de chaves também permite que a conta 1 (111122223333) use a chave do KMS em operações de criptografia (especificadas no elemento `Action` da instrução de política). No entanto, ninguém na conta 1 pode usar a chave do KMS na conta 2 até que conta 1 defina políticas do IAM que concedam às entidades principais acesso à chave do KMS.

No fluxograma, essa política de chaves na conta 2 atende à condição A política de chaves PERMITE que a conta do autor da chamada use políticas do IAM para controlar o acesso à chave?.

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": "*"
  }
]
}

```

- Uma política do IAM no chamador Conta da AWS (conta 1, 111122223333) dá ao principal permissão para realizar operações criptográficas usando a chave KMS na conta 2 (444455556666). O elemento Action concede à entidade principal as mesmas permissões que a política de chaves na conta 2 concedeu à conta 1. Para dar essas permissões à função Engineering na conta 1, [essa política em linha está incorporada](#) na função Engineering.

Políticas do IAM entre contas como essa são efetivas somente quando a política de chaves para a chave do KMS na conta 2 concede à conta 1 permissão para usar a chave do KMS. Além disso, a conta 1 só pode conceder aos seus principais permissão para executar as ações que a política de chaves atribuiu à conta.

No fluxograma, isso atende à condição Uma política do IAM permite que o autor da chamada realize essa ação?.

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"
        ]
      }
    ]
  }
}

```

- O último elemento exigido é a definição da função Engineering na conta 1. O AssumeRolePolicyDocument na função permite que Bob assuma a função Engineering.

```

{
  "Role": {
    "Arn": "arn:aws:iam::111122223333:role/Engineering",
    "CreateDate": "2019-05-16T00:09:25Z",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": {
        "Principal": {
          "AWS": "arn:aws:iam::111122223333:user/bob"
        },
        "Effect": "Allow",
        "Action": "sts:AssumeRole"
      }
    },
    "Path": "/",
    "RoleName": "Engineering",
    "RoleId": "AR0A4KJY2TU23Y7NK62MV"
  }
}

```

AWS KMS glossário de controle de acesso

O tópico a seguir descreve termos e conceitos importantes no controle de AWS KMS acesso.

Autenticação

A autenticação é o processo de verificação da identidade. Para enviar uma solicitação para AWS KMS, você deve fazer login AWS usando suas AWS credenciais.

Autorização

A autorização fornece a permissão para enviar solicitações para criar, gerenciar ou usar AWS KMS recursos. Por exemplo, você deve ter autorização para usar uma chave do KMS em uma operação criptográfica.

Para controlar o acesso aos seus AWS KMS recursos, use [políticas de chave](#), [políticas de IAM](#) e [concessões](#). Cada chave do KMS deve ter uma política de chaves. Se a política de chaves permitir, também é possível usar concessões e políticas do IAM para conceder às entidades principais acesso à chave do KMS. Para detalhar a autorização, você pode usar [chaves de condição](#) que permitem ou negam acesso somente quando uma solicitação ou recurso atende às condições especificadas. Também é possível permitir o acesso às entidades principais em que confia em [outras Contas da AWS](#).

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte [Como fazer login em sua Conta da AWS](#) no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte [AWS Signature Version 4 para solicitações de API](#) no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte [O que é o IAM Identity Center?](#) no Guia do usuário do Centro de Identidade do AWS IAM .

Usuários e grupos do IAM

Um [usuário do IAM](#) é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte [Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias](#) no Guia do usuário do IAM.

Um [grupo do IAM](#) especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte [Casos de uso de usuários do IAM](#) no Guia do usuário do IAM.

Perfis do IAM

Uma [perfil do IAM](#) é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função [mudando de um usuário para uma função do IAM \(console\)](#) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte [Métodos para assumir um perfil](#) no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

Políticas baseadas em recursos

Uma [política de AWS KMS chaves](#) é uma política baseada em recursos que controla o acesso a uma chave KMS. Cada chave do KMS deve ter uma política de chaves. É possível usar outro mecanismo de autorização para permitir o acesso à chave do KMS, mas somente se a política de chaves permitir. (É possível usar uma política do IAM para negar o acesso para uma chave do KMS, mesmo que a política de chaves não permita isso explicitamente.)

Políticas baseadas em recursos são documentos de política JSON que você vincula a um recurso, como uma chave do KMS, para controlar o acesso ao recurso específico. A política baseada em recursos define as ações que uma entidade principal especificada pode executar nesse recurso e sob quais condições. Você não especifica o recurso em uma política baseada em recursos, mas deve especificar um principal, como contas, usuários, funções, usuários federados ou. Serviços da AWS As políticas baseadas em recursos são políticas em linha localizadas no serviço que gerencia o recurso. Você não pode usar políticas AWS gerenciadas do IAM, como a [política `AWSKeyManagementServicePowerUser` gerenciada, em uma política](#) baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte [Políticas de controle de serviço](#) no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM.

AWS KMS recursos

Em AWS KMS, o recurso principal é um AWS KMS key. AWS KMS também oferece suporte a um [alias](#), um recurso independente que fornece um nome amigável para uma chave KMS. Algumas AWS KMS operações permitem que você use um alias para identificar uma chave KMS.

Cada instância de uma chave do KMS ou um alias tem um [Amazon Resource Name](#) (ARN) exclusivo com um formato padrão. Em AWS KMS recursos, o nome do AWS serviço é kms.

- AWS KMS key

Formato ARN:

arn:AWS partition name:AWS service name:Região da AWS:Conta da AWS ID:key/key ID

Exemplo de ARN:

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

- Alias

Formato ARN:

arn:AWS partition name:AWS service name:Região da AWS:Conta da AWS ID:alias/alias name

Exemplo de ARN:

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

AWS KMS fornece um conjunto de operações de API para trabalhar com seus AWS KMS recursos. Para obter mais informações sobre a identificação de chaves KMS nas operações Console de gerenciamento da AWS e AWS KMS da API, consulte [Identificadores-chave \(\) KeyId](#). Para ver uma lista de AWS KMS operações, consulte a [Referência AWS Key Management Service da API](#).

Criar uma chave do KMS

Você pode criar AWS KMS keys no Console de gerenciamento da AWS ou usando a [CreateKey](#) operação ou o [AWS::KMS::Key AWS CloudFormation recurso](#). Durante esse processo, você define a política de chave para a chave do KMS, que pode ser alterada a qualquer momento. Você também seleciona os seguintes valores que definem o tipo de chave do KMS que você cria. Não é possível alterar essas propriedades depois que a chave do KMS é criada.

Tipo de chave KMS

O tipo de chave é uma propriedade que determina qual tipo de chave criptográfica é criada. AWS KMS oferece três tipos principais para proteger os dados:

- Chaves simétricas Advanced Encryption Standard (AES)

Chaves de 256 bits que são usadas no modo Galois Counter Mode (GCM) do AES para fornecer dados autenticados com menos de 4 KB encryption/decryption de tamanho. Esse é o tipo mais comum de chave e é usado para proteger outras chaves de criptografia de dados usadas em seus aplicativos e, com Serviços da AWS isso, criptografar seus dados em seu nome.

- RSA, curva elíptica ou (somente regiões SM2 da China) chaves assimétricas

Essas chaves estão disponíveis em vários tamanhos e são compatíveis com vários algoritmos. É possível usá-las para operações de criptografia e descryptografia, assinatura e verificação ou derivação de segredos compartilhados, dependendo da opção de algoritmo.

- Chaves simétricas para realizar operações de códigos de autenticação de mensagem por hash (HMAC)

Essas chaves são chaves de 256 bits usadas para operações de assinatura e verificação.

Não é possível exportar as chaves do KMS do serviço em texto simples. Eles são gerados e só podem ser usados nos módulos de segurança de hardware (HSMs) usados pelo serviço. Essa é a propriedade de segurança fundamental de AWS KMS garantir que as chaves não sejam comprometidas.

Uso da chave

O uso da chave é uma propriedade que determina as operações de criptografia compatíveis com a chave. As chaves do KMS podem ter um uso de chave de ENCRYPT_DECRYPT, SIGN_VERIFY,

GENERATE_VERIFY_MAC ou KEY_AGREEMENT. Cada chave do KMS pode ter apenas um uso de chave. Isso segue as melhores práticas de uso de chaves, de acordo com [Special Publication 800-57 Recommendations for Key Management do National Institute of Standards and Technology \(NIST\)](#), seção 5.2, Key usage. O uso de uma chave do KMS para mais de um tipo de operação torna o produto de ambas as operações mais vulnerável a ataques.

Especificação da chave

A especificação da chave é uma propriedade que representa a configuração criptográfica de uma chave. O significado da especificação de chave difere com o tipo de chave.

Para chaves do KMS, especificação da chave determina se a chave do KMS é simétrica ou assimétrica. Ela também determina o tipo do material de chave e os algoritmos compatíveis.

A especificação de chaves padrão, [SYMMETRIC_DEFAULT](#), representa uma chave de criptografia simétrica de 256 bits. Para obter uma descrição detalhada de todas as especificações de chave compatíveis, consulte [Referência de especificação de chave](#).

Origem do material de chave

A origem do material de chave é uma propriedade da chave do KMS que identifica a origem do material de chave na chave do KMS. Você escolhe a origem do material de chave ao criar a chave do KMS e não é possível alterá-la. A origem do material de chave afeta as características de segurança, durabilidade, disponibilidade, latência e throughput da chave do KMS.

Cada chave do KMS inclui uma referência ao material de chave em seus metadados. A origem do material de chave de chaves do KMS de criptografia simétrica pode variar. Você pode usar material de chave gerado pelo AWS KMS, material chave gerado em um [repositório de chaves personalizado](#) ou [importar seu próprio material de chave](#).

Por padrão, cada chave do KMS tem material de chave exclusivo. No entanto, você pode criar um conjunto de [chaves de várias regiões](#) com o mesmo material de chave.

As chaves KMS podem ter um dos seguintes valores de origem do material principal: AWS_KMS, EXTERNAL ([material de chave importado](#)), AWS_CLOUDHSM ([chave KMS em um armazenamento de chaves](#)) ou EXTERNAL_KEY_STORE ([AWS CloudHSM chave KMS em um armazenamento de chaves externo](#)).

Tópicos

- [Permissões para criar chaves do KMS](#)

- [Escolher qual tipo de chave do KMS criar](#)
- [Criar uma chave do KMS de criptografia simétrica](#)
- [Criar uma chave do KMS assimétrica](#)
- [Criar uma chave do KMS HMAC](#)
- [Criar chaves primárias de várias regiões](#)
- [Criar chaves de réplica de várias regiões](#)
- [Criar uma chave do KMS com material de chave importado](#)
- [Criar uma chave KMS em um armazenamento de AWS CloudHSM chaves](#)
- [Criar uma chave do KMS em repositórios de chaves externos](#)

Permissões para criar chaves do KMS

Para criar uma chave KMS no console ou usando o APIs, você deve ter a seguinte permissão em uma política do IAM. Sempre que possível, use [chaves de condição](#) para limitar as permissões. Por exemplo, você pode usar a chave de KeySpec condição [kms:](#) em uma política do IAM para permitir que os diretores criem somente chaves de criptografia simétricas.

Para obter um exemplo de uma política do IAM para entidades principais que criam chaves, consulte [Permitir que um usuário crie chaves do KMS](#).

Note

Tenha cuidado ao conceder permissão a entidades principais para gerenciar etiquetas e aliases. Alterar uma etiqueta ou um alias pode conceder ou negar uma permissão à chave gerenciada pelo cliente. Para obter detalhes, consulte [ABAC para AWS KMS](#).

- [kms: CreateKey](#) é obrigatório.
- [kms: CreateAlias](#) é necessário para criar uma chave KMS no console em que um alias é necessário para cada nova chave KMS.
- [kms: TagResource](#) é necessário adicionar tags ao criar a chave KMS.
- [iam: CreateServiceLinkedRole](#) é necessário para criar chaves primárias multirregionais. Para obter detalhes, consulte [Controlar o acesso a chaves de várias regiões](#).

A PutKeyPolicy permissão [kms:](#) não é necessária para criar a chave KMS. A permissão `kms:CreateKey` inclui permissão para definir a política de chaves inicial. Porém, você deve adicionar essa permissão à política de chaves ao criar a chave do KMS para garantir que seja possível controlar o acesso à chave do KMS. A alternativa é usar o [BypassLockoutSafetyCheck](#) parâmetro, o que não é recomendado.

As chaves KMS pertencem à AWS conta na qual foram criadas. O usuário do IAM que cria uma chave do KMS não é considerado o proprietário da chave e ele não recebe automaticamente permissão para usar ou gerenciar a chave do KMS que criou. Como qualquer outra entidade principal, o criador da chave precisa obter permissão por meio de uma política de chaves, política do IAM ou concessão. No entanto, as entidades principais que têm a permissão `kms:CreateKey` podem definir a política de chave inicial e conceder a si mesmas permissão para usar ou gerenciar a chave.

Escolher qual tipo de chave do KMS criar

O tipo de chave do KMS criado depende em grande parte de como você planeja usar a chave do KMS e dos seus requisitos de segurança e de autorização. O tipo de chave e o uso de uma chave do KMS determinam quais operações criptográficas a chave poderá realizar. Cada chave do KMS pode ter apenas um uso. O uso de uma chave do KMS para mais de um tipo de operação torna o produto de todas as operações mais vulnerável a ataques.

Para permitir que os diretores criem chaves KMS somente para um determinado uso de chave, use a chave de condição [kms: KeyUsage](#). Também é possível usar a chave de condição `kms:KeyUsage` para permitir que as entidades principais chamem operações de API para uma chave do KMS baseada em seu uso de chave. Por exemplo, apenas será possível conceder permissão para desabilitar uma chave do KMS se o seu uso de chave for `SIGN_VERIFY`.

Use as orientações a seguir para determinar o tipo de chave do KMS necessário com base no seu caso de uso.

Criptografar e descriptografar dados

Use uma [chave do KMS simétrica](#) para a maioria dos casos de uso que exigem criptografia e descriptografia dos dados. O algoritmo de criptografia simétrica usado pelo AWS KMS é rápido, eficiente e garante a confidencialidade e a autenticidade dos dados. Ele oferece suporte à criptografia autenticada com dados adicionais autenticados (ADD), definidos como um [contexto de criptografia](#). Esse tipo de chave KMS exige que o remetente e o destinatário dos dados criptografados tenham AWS credenciais válidas para ligar. AWS KMS

Se seu caso de uso exigir criptografia externa AWS por usuários que não podem ligar AWS KMS, [as chaves KMS assimétricas](#) são uma boa opção. Você pode distribuir a parte pública da chave do KMS assimétrica para permitir que esses usuários criptografem dados. Além disso, as aplicações que precisam descriptografar esses dados podem usar a chave privada da chave do KMS assimétrica no AWS KMS.

Assinar mensagens e verificar assinaturas

Para assinar mensagens e verificar assinaturas, é necessário usar uma [chave do KMS assimétrica](#). Você pode usar uma chave KMS com uma [especificação de chave](#) que represente um par de chaves RSA, um par de chaves de curva elíptica (ECC), um par de chaves ML-DSA ou um par de chaves (somente regiões da China). SM2 A especificação de chave escolhida é determinada pelo algoritmo de assinatura que você deseja usar. Os algoritmos de assinatura ECDSA compatíveis com os pares de chaves ECC são recomendados em vez dos algoritmos de assinatura RSA. Use um par de chaves ML-DSA ao migrar de chaves RSA ou ECC para chaves pós-quânticas. No entanto, pode ser necessário usar uma especificação de chave e um algoritmo de assinatura específicos para oferecer suporte aos usuários que verificam assinaturas de forma externa à AWS.

Criptografar com pares de chaves assimétricas

[Para criptografar dados com um par de chaves assimétrico, você deve usar uma chave KMS assimétrica com uma especificação de chave RSA ou uma SM2 especificação de chave \(somente regiões da China\)](#). Para criptografar dados no AWS KMS com a chave pública de um par de chaves do KMS, use a operação [Encrypt](#) (Criptografar). Você também pode [baixar a chave pública](#) e compartilhá-la com as partes que precisam criptografar dados fora dela AWS KMS.

Ao baixar a chave pública de uma chave do KMS assimétrica, é possível usá-la fora do AWS KMS. Mas ele não está mais sujeito aos controles de segurança que protegem a chave KMS. AWS KMS Por exemplo, você não pode usar políticas ou concessões de AWS KMS chaves para controlar o uso da chave pública. Você também não pode controlar se a chave é usada somente para criptografia e decodificação usando os algoritmos de criptografia compatíveis. AWS KMS Para obter mais detalhes, consulte [Considerações especiais sobre o download de chaves públicas](#).

[Para descriptografar dados que foram criptografados com a chave pública externa AWS KMS, chame a operação Decrypt](#). A operação Decrypt vai falhar se os dados tiverem sido criptografados com uma chave pública de uma chave do KMS com um uso de chave de SIGN_VERIFY. Também falhará se for criptografado usando um algoritmo que não AWS KMS suporta a especificação de chave que você selecionou. Para obter mais informações sobre

especificações de chaves e algoritmos compatíveis, consulte [Referência de especificação de chave](#).

Para evitar esses erros, qualquer pessoa que use uma chave pública fora da AWS KMS deve armazenar a configuração da chave. O AWS KMS console e a [GetPublicKey](#) resposta fornecem as informações que você deve incluir ao compartilhar a chave pública.

Derivar segredos compartilhados

Para derivar segredos compartilhados, use uma chave KMS com [curva elíptica padrão NIST ou material](#) de chave [SM2](#) (somente para regiões da China). AWS KMS usa o [cofator de criptografia de curva elíptica Diffie-Hellman Primitive](#) (ECDH) para estabelecer um acordo de chave entre dois pares, derivando um segredo compartilhado de seus pares de chaves público-privadas de curva elíptica. Você pode usar o segredo compartilhado bruto que a [DeriveSharedSecret](#) operação retorna para derivar uma chave simétrica que pode criptografar e descriptografar dados enviados entre duas partes ou gerar e verificar. HMACs AWS KMS recomenda que você siga as [recomendações do NIST para derivação de chaves](#) ao usar o segredo compartilhado bruto para derivar uma chave simétrica.

Gerar e verificar códigos de HMAC

Para gerar e verificar códigos de autenticação de mensagem por hash, use uma chave do KMS de HMAC. Quando você cria uma chave HMAC AWS KMS, AWS KMS cria e protege seu material de chave e garante que você use os algoritmos MAC corretos para sua chave. Também é possível usar os códigos de HMAC como números pseudo-aleatórios e para assinatura simétrica e tokenização em determinados cenários.

As chaves do KMS de HMAC são chaves simétricas. Ao criar uma chave do KMS de HMAC no console do AWS KMS, escolha o tipo de chave `Symmetric`.

Use com AWS serviços

Para criar uma chave KMS para uso com um [AWS serviço integrado AWS KMS](#), consulte a documentação do serviço. AWS serviços que criptografam seus dados exigem uma chave [KMS de criptografia simétrica](#).

Além dessas considerações, as operações criptográficas em chaves do KMS com diferentes especificações de chave têm preços e cotas de solicitação diferentes. Para obter mais informações sobre a definição de preço do AWS KMS, consulte [Definição de preço do AWS Key Management Service](#). Para obter mais informações sobre cotas de solicitações, consulte [Cotas de solicitações](#).

Criar uma chave do KMS de criptografia simétrica

Este tópico explica como criar a chave KMS básica, uma chave KMS de [criptografia simétrica](#) para uma única região com material de chaves de. AWS KMS Você pode usar essa chave do KMS para proteger seus recursos em um AWS service (Serviço da AWS).

[Você pode criar chaves KMS de criptografia simétrica no AWS KMS console, usando a CreateKeyAPI ou usando o AWS::KMS::Key CloudFormation modelo.](#)

A especificação de chave padrão, [SYMMETRIC_DEFAULT](#), é a especificação de chave para chaves do KMS de criptografia simétrica. Quando você seleciona o tipo de chave simétrica e o uso da chave de criptografia e descriptografia no AWS KMS console, ele seleciona a especificação da chave. SYMMETRIC_DEFAULT Na [CreateKey](#) operação, se você não especificar um KeySpec valor, SYMMETRIC_DEFAULT será selecionado. Se você não tiver um motivo para usar uma especificação de chave diferente, SYMMETRIC_DEFAULT é uma boa escolha.

Para obter informações sobre cotas que se aplicam a chaves do KMS, consulte [Cotas](#).

Usando o AWS KMS console

Você pode usar o Console de gerenciamento da AWS para criar AWS KMS keys (chaves KMS).

Important

Não inclua informações confidenciais ou sigilosas no alias, na descrição ou nas tags. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Escolha Criar chave.
5. Para criar uma chave do KMS de criptografia simétrica, em Key type (Tipo de chave), selecione Symmetric (Simétrica).
6. Em Uso da chave, a opção Criptografar e descriptografar é selecionada para você.

7. Escolha Próximo.
8. Digite um alias para a chave do KMS. O nome do alias não pode começar com **aws/**. O **aws/** prefixo é reservado pela Amazon Web Services para representar Chaves gerenciadas pela AWS em sua conta.

 Note

Adicionar, excluir ou atualizar um alias pode conceder ou negar uma permissão à chave do KMS. Para obter mais detalhes, consulte [ABAC para AWS KMS](#) e [Usar aliases para controlar o acesso a chaves do KMS](#).

Um alias é um nome de exibição que identifica a chave do KMS. Recomendamos que você escolha um alias que indique o tipo de dados que pretende proteger ou a aplicação a ser usada com a chave do KMS.

Aliases são necessários ao criar uma chave do KMS no Console de gerenciamento da AWS. Eles são opcionais quando você usa a [CreateKey](#) operação.

9. (Opcional) Digite uma descrição para a chave do KMS.

Você pode adicionar uma descrição agora ou atualizá-la a qualquer momento, a não ser que o [estado da chave](#) seja Pending Deletion ou Pending Replica Deletion. Para adicionar, alterar ou excluir a descrição de uma chave gerenciada pelo cliente existente, edite a descrição na página de detalhes da chave KMS na Console de gerenciamento da AWS ou use a [UpdateKeyDescription](#) operação.

10. (Opcional) Digite uma chave de tag e um valor de tag opcional. Para adicionar mais de uma etiqueta à chave do KMS, selecione Adicionar tag.

 Note

Marcar ou desmarcar uma chave do KMS pode conceder ou negar uma permissão a essa chave do KMS. Para obter mais detalhes, consulte [ABAC para AWS KMS](#) e [Usar tags para controlar o acesso a chaves do KMS](#).

Quando você adiciona tags aos seus AWS recursos, AWS gera um relatório de alocação de custos com uso e custos agregados por tags. Etiquetas também podem ser utilizadas para

controlar o acesso a uma chave do KMS. Para informações sobre marcação de chaves do KMS, consulte [Etiquetas em AWS KMS](#) e [ABAC para AWS KMS](#).

11. Escolha Próximo.
12. Selecione os usuários e as funções do IAM que podem administrar a chave do KMS.

Observações

Essa política de chaves dá o controle Conta da AWS total dessa chave KMS. Ela permite que os administradores de conta usem políticas do IAM para conceder a outras entidades principais a permissão para gerenciar a chave do KMS. Para obter detalhes, consulte [the section called “Política de chaves padrão”](#).

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

O AWS KMS console adiciona administradores de chaves à política de chaves sob o identificador "Allow access for Key Administrators" de instrução. A modificação desse identificador de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

13. (Opcional) Para evitar que os usuários e funções do IAM selecionados excluam essa chave do KMS, na seção Exclusão de chaves na parte inferior da página, desmarque a caixa de seleção Permitir que os administradores de chaves excluam essa chave.
14. Escolha Próximo.
15. Selecione os usuários e as funções do IAM que podem usar a chave em [operações de criptografia](#).

Observações

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

O AWS KMS console adiciona os principais usuários à política de chaves sob os identificadores de declaração "Allow use of the key" e "Allow attachment

of persistent resources" A modificação desses identificadores de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

16. (Opcional) Você pode permitir que outras Contas da AWS pessoas usem essa chave KMS para operações criptográficas. Para fazer isso, na Contas da AWS seção Outros na parte inferior da página, escolha Adicionar outro Conta da AWS e insira o número de Conta da AWS identificação de uma conta externa. Para adicionar várias contas externas, repita essa etapa.

 Note

Para permitir que as entidades principais de contas externas usem a chave do KMS, os administradores da conta externa devem criar políticas do IAM que forneçam essas permissões. Para obter mais informações, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

17. Escolha Próximo.
18. Consulte as instruções da política de chave para a chave. Para fazer alterações na política de chave, selecione Editar.
19. Escolha Próximo.
20. Revise as configurações que você escolheu. Ainda é possível voltar e alterar todas as configurações.
21. Selecione Finish (Concluir) para criar a chave do KMS.

Usando a AWS KMS API

Você pode usar a [CreateKey](#) operação para criar AWS KMS keys de todos os tipos. Estes exemplos usam a [AWS Command Line Interface \(AWS CLI\)](#). Para obter exemplos em várias linguagens de programação, consulte [Use CreateKey com um AWS SDK ou CLI](#).

 Important

Não inclua informações confidenciais ou sigilosas nos campos Description ou Tags. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

A operação a seguir cria uma chave de criptografia simétrica em uma única região com o suporte de material de chave gerado pelo AWS KMS. Essa operação não tem os parâmetros obrigatórios.

No entanto, você também pode usar o parâmetro `Policy` para especificar uma política de chaves. Você pode alterar a política de chaves ([PutKeyPolicy](#)) e adicionar elementos opcionais, como uma [descrição](#) e [tags](#), a qualquer momento. Você também pode criar [chaves assimétricas](#), [chaves de várias Regiões](#), chaves com [material de chave importado](#), e chaves em [armazenamentos de chaves personalizados](#). Para criar chaves de dados para criptografia do lado do cliente, use a [GenerateDataKey](#) operação.

A `CreateKey` operação não permite especificar um alias, mas você pode usar a [CreateAlias](#) operação para criar um alias para sua nova chave KMS.

Veja a seguir um exemplo de uma chamada para a operação `CreateKey` sem parâmetros. Esse comando usa todos os valores padrão. Ele cria uma chave do KMS de criptografia simétrica com o material de chave gerado pelo AWS KMS.

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

Se você não especificar uma política de chaves para sua nova chave do KMS, a [política de chave padrão](#) aplicada por `CreateKey` será diferente da política de chaves padrão que o console aplica quando você o usa para criar uma nova chave do KMS.

Por exemplo, essa chamada para a [GetKeyPolicy](#) operação retorna a política de chaves que `CreateKey` se aplica. Ele dá Conta da AWS acesso à chave KMS e permite criar políticas AWS Identity and Access Management (IAM) para a chave KMS. Para obter informações detalhadas sobre as políticas do IAM e as políticas de chaves para chaves do KMS, consulte [Acesso e permissões de chave do 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" : "*"
  } ]
}
```

Criar uma chave do KMS assimétrica

[Você pode criar chaves KMS assimétricas no AWS KMS console, usando a CreateKeyAPI ou usando o modelo. AWS::KMS::Key CloudFormation](#) Uma chave do KMS assimétrica representa um par de chaves pública e privada que pode ser usado para criptografia, assinatura ou derivação de segredos compartilhados. A chave privada permanece dentro AWS KMS. Para baixar a chave pública para uso fora do AWS KMS, consulte [Fazer download de chave pública](#).

Ao criar uma chave do KMS assimétrica, é necessário selecionar sua especificação de chave. Geralmente a especificação de chave escolhida é determinada por requisitos regulatórios, de segurança ou de negócios. Ela também pode ser influenciada pelo tamanho das mensagens que você precisa criptografar ou assinar. Em geral, chaves de criptografia maiores são mais resistentes

a ataques de força bruta. Para obter uma descrição detalhada de todas as especificações de chave compatíveis, consulte [Referência de especificação de chave](#).

AWS serviços que se integram com AWS KMS não oferecem suporte a chaves KMS assimétricas. Se você quiser criar uma chave KMS que criptografe os dados que você armazena ou gerencia em um AWS serviço, [crie uma chave KMS de criptografia simétrica](#).

Para obter informações sobre as permissões necessárias para criar chaves do KMS, consulte [Permissões para criar chaves do KMS](#).

Usando o AWS KMS console

Você pode usar o Console de gerenciamento da AWS para criar assimetrias AWS KMS keys (chaves KMS). Cada chave do KMS assimétrica representa um par de chaves pública e privada.

Important

Não inclua informações confidenciais ou sigilosas no alias, na descrição ou nas tags. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Escolha Criar chave.
5. Para criar uma chave do KMS assimétrica, em Key type (Tipo de chave), selecione Asymmetric (Assimétrica).
6. Para criar uma chave do KMS assimétrica para assinar mensagens e verificar assinaturas, em Key usage (Uso de chaves), selecione Encrypt and decrypt (Criptografar e descriptografar).

Para criar uma chave do KMS assimétrica para assinar mensagens e verificar assinaturas, em Uso de chave, selecione Assinar e verificar.

Para criar uma chave do KMS assimétrica para derivar segredos compartilhados, em Uso da chave, escolha Acordo de chave.

Para ajudar a escolher um valor de uso de chave, consulte [Escolher qual tipo de chave do KMS criar](#).

7. Escolha uma especificação (Especificação da chave) para a chave do KMS assimétrica.
8. Escolha Próximo.
9. Digite um [alias](#) para a chave do KMS. O nome do alias não pode começar com **aws/**. O prefixo **aws/** é reservado pela Amazon Web Services para representar as Chaves gerenciadas pela AWS na sua conta.

Um alias é um nome amigável que você pode usar para identificar a chave KMS no console e em alguns AWS KMS APIs. Recomendamos que você escolha um alias que indique o tipo de dados que pretende proteger ou a aplicação a ser usada com a chave do KMS.

Aliases são necessários ao criar uma chave do KMS no Console de gerenciamento da AWS. Você não pode especificar um alias ao usar a [CreateKey](#) operação, mas pode usar o console ou a [CreateAlias](#) operação para criar um alias para uma chave KMS existente. Para obter detalhes, consulte [Aliases em AWS KMS](#).

10. (Opcional) Digite uma descrição para a chave do KMS.

Insira uma descrição que explique o tipo de dados que você planeja proteger ou a aplicação que planeja usar com a chave do KMS.

Você pode adicionar uma descrição agora ou atualizá-la a qualquer momento, a não ser que o [estado da chave](#) seja Pending Deletion ou Pending Replica Deletion. Para adicionar, alterar ou excluir a descrição de uma chave gerenciada pelo cliente existente, edite a descrição na página de detalhes da chave KMS na Console de gerenciamento da AWS ou use a [UpdateKeyDescription](#) operação.

11. (Opcional) Digite uma chave de tag e um valor de tag opcional. Para adicionar mais de uma etiqueta à chave do KMS, selecione Adicionar tag.

Quando você adiciona tags aos seus AWS recursos, AWS gera um relatório de alocação de custos com uso e custos agregados por tags. Etiquetas também podem ser utilizadas para controlar o acesso a uma chave do KMS. Para informações sobre marcação de chaves do KMS, consulte [Etiquetas em AWS KMS](#) e [ABAC para AWS KMS](#).

12. Escolha Próximo.
13. Selecione os usuários e as funções do IAM que podem administrar a chave do KMS.

Observações

Essa política de chaves dá o controle Conta da AWS total dessa chave KMS. Ela permite que os administradores de conta usem políticas do IAM para conceder a outras entidades principais a permissão para gerenciar a chave do KMS. Para obter detalhes, consulte [the section called “Política de chaves padrão”](#).

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

O AWS KMS console adiciona administradores de chaves à política de chaves sob o identificador "Allow access for Key Administrators" de instrução. A modificação desse identificador de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

14. (Opcional) Para evitar que os usuários e funções do IAM selecionados excluam essa chave do KMS, na seção Exclusão de chaves na parte inferior da página, desmarque a caixa de seleção Permitir que os administradores de chaves excluam essa chave.
15. Escolha Próximo.
16. Selecione os usuários e as funções do IAM que podem usar a chave do KMS para [operações de criptografia](#).

Observações

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

O AWS KMS console adiciona os principais usuários à política de chaves sob os identificadores de declaração "Allow use of the key" e "Allow attachment of persistent resources". A modificação desses identificadores de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

17. (Opcional) Você pode permitir que outras Contas da AWS pessoas usem essa chave KMS para operações criptográficas. Para fazer isso, na seção Outras Contas da AWS, no fim da página,

escolha Adicionar outra Conta da AWS e insira o número de identificação de Conta da AWS de uma conta externa. Para adicionar várias contas externas, repita essa etapa.

 Note

Para permitir que as entidades principais de contas externas usem a chave do KMS, os administradores da conta externa devem criar políticas do IAM que forneçam essas permissões. Para obter mais informações, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

18. Escolha Próximo.
19. Consulte as instruções da política de chave para a chave. Para fazer alterações na política de chave, selecione Editar.
20. Escolha Próximo.
21. Revise as configurações que você escolheu. Ainda é possível voltar e alterar todas as configurações.
22. Selecione Finish (Concluir) para criar a chave do KMS.

Usando a AWS KMS API

Você pode usar a [CreateKey](#) operação para criar uma assimétrica AWS KMS key. Estes exemplos usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Ao criar uma chave do KMS assimétrica, você deve especificar o parâmetro KeySpec, que determina o tipo de chaves que criado. Além disso, é necessário especificar um valor KeyUsage de ENCRYPT_DECRYPT, SIGN_VERIFY ou KEY_AGREEMENT. Não é possível alterar essas propriedades depois que a chave do KMS é criada.

A [CreateKey](#) operação não permite especificar um alias, mas você pode usar a [CreateAlias](#) operação para criar um alias para sua nova chave KMS.

 Important

Não inclua informações confidenciais ou sigilosas nos campos Description ou Tags. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

Criar um par de chaves do KMS assimétricas para criptografia pública

O exemplo a seguir usa a operação `CreateKey` para criar uma chave do KMS assimétrica de chaves RSA de 4.096 bits projetada para criptografia de chave pública.

```
$ 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
  }
}
```

Criar um par de chaves do KMS assimétricas para assinatura e verificação

O exemplo de comando a seguir cria uma chave do KMS assimétrica que representa um par de chaves ECC usado para assinatura e verificação. Não é possível criar um par de chaves de curva elíptica para criptografia e descriptografia.

```
$ 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"
}
}

```

Criar um par de chaves do KMS assimétricas para derivar segredos compartilhados

O exemplo de comando a seguir cria uma chave do KMS assimétrica que representa um par de chaves ECDH usado para derivar segredos compartilhados. Não é possível criar um par de chaves de curva elíptica para criptografia e descryptografia.

```

$ 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
  }
}

```

Criar uma chave do KMS HMAC

[Você pode criar chaves HMAC KMS no AWS KMS console, usando a CreateKeyAPI ou usando o AWS::KMS::Key CloudFormation modelo.](#)

Ao criar uma chave HMAC KMS, você deve selecionar uma especificação de chave. AWS KMS suporta várias [especificações de chave para chaves HMAC KMS](#). A especificação de chave que você escolhe pode ser determinada por requisitos regulatórios, de segurança ou de negócios. Em geral, chaves maiores são mais resistentes a ataques de força bruta.

Para obter informações sobre as permissões necessárias para criar chaves do KMS, consulte [Permissões para criar chaves do KMS](#).

Usando o AWS KMS console

Você pode usar o Console de gerenciamento da AWS para criar chaves HMAC KMS. As chaves do KMS de HMAC são chaves simétricas com um uso de chave para gerar e verificar Message authentication code (MAC – Código de autenticação de mensagem). Você também pode criar chaves de HMAC de várias regiões.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Escolha Criar chave.
5. Para Key type (Tipo de chave), escolha Symmetric (Simétrica).

As chaves do KMS de HMAC são simétricas. Você usa a mesma chave para gerar e verificar etiquetas de HMAC.

6. Em Key usage (Uso de chave), escolha Generate and verify MAC (Gerar e verificar MAC).

Gerar e verificar MAC é o único uso de chave válido para chaves do KMS de HMAC.

Note

Key usage (Uso de chave) é exibido para chaves simétricas somente quando as chaves do KMS de HMAC são compatíveis com a região selecionada.

7. Escolha uma especificação (Key spec [Especificação de chave]) para sua chave do KMS de HMAC.

A especificação de chave que você escolhe pode ser determinada por requisitos regulatórios, de segurança ou de negócios. Em geral, chaves mais longas são mais seguras.

8. Para criar uma chave de HMAC primária de [várias regiões](#), em Advanced options (Opções avançadas), escolha Multi-Region key (Chave de várias regiões). As [propriedades compartilhadas](#) que você define para essa chave do KMS, como o tipo de chave e o uso de chave, serão compartilhados com suas réplicas de chave.

Não é possível aplicar esse procedimento para criar uma réplica de chave. Para criar uma réplica de chave de HMAC de várias regiões, siga as [instruções para criar uma réplica de chave](#).

9. Escolha Próximo.
10. Insira um [alias](#) para a chave do KMS. O nome do alias não pode começar com **aws/**. O prefixo **aws/** é reservado pela Amazon Web Services para representar as Chaves gerenciadas pela AWS na sua conta.

Recomendamos que você use um alias que identifique a chave do KMS como uma chave de HMAC, p. ex., HMAC/test-key. Isso facilitará a identificação das chaves HMAC no AWS KMS console, onde você pode classificar e filtrar as chaves por tags e aliases, mas não por especificação ou uso da chave.

Aliases são necessários ao criar uma chave do KMS no Console de gerenciamento da AWS. Você não pode especificar um alias ao usar a [CreateKey](#) operação, mas pode usar o console ou a [CreateAlias](#) operação para criar um alias para uma chave KMS existente. Para obter detalhes, consulte [Aliases em AWS KMS](#).

11. (Opcional) Insira uma descrição para a chave do KMS.

Insira uma descrição que explique o tipo de dados que você planeja proteger ou a aplicação que planeja usar com a chave do KMS.

Você pode adicionar uma descrição agora ou atualizá-la a qualquer momento, a não ser que o [estado da chave](#) seja Pending Deletion ou Pending Replica Deletion. Para adicionar, alterar ou excluir a descrição de uma chave gerenciada pelo cliente existente, edite a descrição na página de detalhes da chave KMS Console de gerenciamento da AWS em Console de gerenciamento da AWS ou use a [UpdateKeyDescription](#) operação.

12. (Opcional) Insira uma chave de etiqueta e um valor opcional de etiqueta. Para adicionar mais de uma etiqueta à chave do KMS, selecione Adicionar tag.

Considere a possibilidade de adicionar uma etiqueta que identifique a chave como uma chave de HMAC, p. ex., Type=HMAC. Isso facilitará a identificação das chaves HMAC no AWS KMS console, onde você pode classificar e filtrar as chaves por tags e aliases, mas não por especificação ou uso da chave.

Quando você adiciona tags aos seus AWS recursos, AWS gera um relatório de alocação de custos com uso e custos agregados por tags. Etiquetas também podem ser utilizadas para controlar o acesso a uma chave do KMS. Para informações sobre marcação de chaves do KMS, consulte [Etiquetas em AWS KMS](#) e [ABAC para AWS KMS](#).

13. Escolha Próximo.
14. Selecione os usuários e as funções do IAM que podem administrar a chave do KMS.

Observações

Essa política de chaves dá o controle Conta da AWS total dessa chave KMS. Ela permite que os administradores de conta usem políticas do IAM para conceder a outras entidades principais a permissão para gerenciar a chave do KMS. Para obter detalhes, consulte [the section called “Política de chaves padrão”](#).

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

O AWS KMS console adiciona administradores de chaves à política de chaves sob o identificador "Allow access for Key Administrators" de instrução. A modificação desse identificador de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

15. (Opcional) Para evitar que os usuários e funções do IAM selecionados excluam essa chave do KMS, na seção Exclusão de chaves na parte inferior da página, desmarque a caixa de seleção Permitir que os administradores de chaves excluam essa chave.
16. Escolha Próximo.
17. Selecione os usuários e as funções do IAM que podem usar a chave do KMS para [operações de criptografia](#).

Observações

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

O AWS KMS console adiciona os principais usuários à política de chaves sob os identificadores de declaração "Allow use of the key" e "Allow attachment of persistent resources". A modificação desses identificadores de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

18. (Opcional) Você pode permitir que outras Contas da AWS pessoas usem essa chave KMS para operações criptográficas. Para fazer isso, na seção Outras Contas da AWS, no fim da página, escolha Adicionar outra Conta da AWS e insira o número de identificação de Conta da AWS de uma conta externa. Para adicionar várias contas externas, repita essa etapa.

Note

Para permitir que as entidades principais de contas externas usem a chave do KMS, os administradores da conta externa devem criar políticas do IAM que forneçam essas permissões. Para obter mais informações, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

19. Escolha Próximo.
20. Consulte as instruções da política de chave para a chave. Para fazer alterações na política de chave, selecione Editar.
21. Escolha Próximo.
22. Revise as configurações que você escolheu. Ainda é possível voltar e alterar todas as configurações.
23. Escolha Finish (Concluir) para criar a chave do KMS de HMAC.

Usando a AWS KMS API

Você pode usar a [CreateKey](#) operação para criar uma chave HMAC KMS. Estes exemplos usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Ao criar uma chave do KMS de HMAC, você deve especificar o parâmetro `KeySpec`, que determina o tipo de chave do KMS. Além disso, você deve especificar um valor de `GENERATE_VERIFY_MAC` para `KeyUsage`, mesmo que seja o único valor válido de uso de chave para chaves de HMAC. Para criar uma chave do KMS de HMAC de [várias regiões](#), adicione o parâmetro `MultiRegion` com um valor de `true`. Não é possível alterar essas propriedades depois que a chave do KMS é criada.

A `CreateKey` operação não permite especificar um alias, mas você pode usar a [CreateAlias](#) operação para criar um alias para sua nova chave KMS. Recomendamos que você use um alias que identifique a chave do KMS como uma chave de HMAC, p. ex., `HMAC/test-key`. Isso facilitará a identificação das chaves HMAC no AWS KMS console, onde você pode classificar e filtrar as chaves por alias, mas não por especificação ou uso da chave.

Se você tentar criar uma chave HMAC KMS em uma Região da AWS na qual as chaves HMAC não sejam suportadas, a `CreateKey` operação retornará um `UnsupportedOperationException`

O exemplo a seguir usa a operação `CreateKey` para criar uma chave do KMS de HMAC de 512 bits.

```
$ 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
  }
}
```

Criar chaves primárias de várias regiões

Você pode criar uma [chave primária multirregional](#) no AWS KMS console ou usando a AWS KMS API. Você pode criar a chave primária em qualquer Região da AWS lugar que AWS KMS ofereça suporte a chaves multirregionais.

Para criar uma chave primária multirregional, o principal precisa das [mesmas permissões necessárias](#) para criar qualquer chave KMS, incluindo a CreateKey permissão [kms](#): em uma política do IAM. O diretor também precisa do [objetivo: CreateServiceLinkedRole](#) permissão. Você pode usar a chave de MultiRegionKeyType condição [kms](#): para permitir ou negar permissão para criar chaves primárias multirregionais.

Note

Ao criar sua chave primária multirregional, considere cuidadosamente os usuários e os perfis do IAM que você seleciona para administrar e usar a chave. As políticas do IAM podem conceder permissão para gerenciar a chave do KMS a outros usuários e funções do IAM. As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Usando o AWS KMS console

Para criar uma chave primária multirregional no AWS KMS console, use o mesmo processo que você usaria para criar qualquer chave KMS. Selecione uma chave de várias regiões em Advanced options (Opções avançadas). Para obter instruções completas, consulte [Criar uma chave do KMS](#).

Important

Não inclua informações confidenciais ou sigilosas no alias, na descrição ou nas tags. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.

3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Escolha Create key (Criar chave).
5. Selecione um tipo de chave [simétrica ou assimétrica](#). As chaves simétricas são o padrão.

Você pode criar chaves simétricas e assimétricas de várias regiões, inclusive chaves do KMS de HMAC de várias regiões, que são simétricas.

6. Selecione o uso da chave. Encrypt and decrypt (Criptografar e descriptografar) é o padrão.

Para obter ajuda, consulte [Criar uma chave do KMS](#), [the section called “Criar uma chave do KMS assimétrica”](#) ou [the section called “Criar uma chave do KMS HMAC”](#).

7. Expanda Advanced options (Opções avançadas).
8. Em Origem do material da chave, para AWS KMS gerar o material de chave que suas chaves primária e de réplica compartilharão, escolha KMS. Se você estiver [importando material de chave](#) para chaves primárias e de réplica, escolha External (Import key material) (Externo [Importar material de chave]).
9. Em Regionalidade, escolha Chave de várias regiões.

Não será possível alterar essa configuração após a criação da chave do KMS.

10. Digite um [alias](#) para a chave primária.

Aliases não são uma propriedade compartilhada de chaves de várias regiões. Você pode dar à sua chave primária multirregional e suas réplicas o mesmo alias ou aliases diferentes. AWS KMS não sincroniza os aliases das chaves multirregionais.

 Note

Adicionar, excluir ou atualizar um alias pode conceder ou negar uma permissão à chave do KMS. Para obter mais detalhes, consulte [ABAC para AWS KMS](#) e [Usar aliases para controlar o acesso a chaves do KMS](#).

11. (Opcional) Digite uma descrição da chave primária.

Descrições não são uma propriedade compartilhada de chaves de várias regiões. Você pode dar à sua chave primária multirregional e suas réplicas a mesma descrição ou descrições diferentes. AWS KMS não sincroniza as descrições das chaves de várias regiões.

12. (Opcional) Digite uma chave de tag e um valor de tag opcional. Para atribuir mais de uma etiqueta à chave primária, selecione Add tag (Adicionar etiqueta).

Etiquetas não são uma propriedade compartilhada de chaves de várias regiões. Você pode dar à chave primária de várias regiões e suas réplicas as mesmas etiquetas ou etiquetas diferentes. AWS KMS não sincroniza as etiquetas de chaves de várias regiões. É possível alterar as etiquetas em chaves do KMS a qualquer momento.

Note

Marcar ou desmarcar uma chave do KMS pode conceder ou negar uma permissão a essa chave do KMS. Para obter mais detalhes, consulte [ABAC para AWS KMS](#) e [Usar tags para controlar o acesso a chaves do KMS](#).

13. Selecione os usuários e as funções do IAM que podem administrar a chave primária.

Observações

- Essa etapa inicia o processo de criação de uma [política de chaves](#) para a chave primária. Políticas de chaves não são uma propriedade compartilhada de chaves de várias regiões. Você pode dar à sua chave primária multirregional e suas réplicas a mesma política de chaves ou políticas de chave diferentes. AWS KMS não sincroniza as principais políticas das chaves multirregionais. Você pode alterar a política de chaves de uma chave do KMS a qualquer momento.
- Ao criar uma chave primária multirregional, considere usar a [política de chave padrão](#) gerada pelo console. Se você modificar essa política, o console não fornecerá as etapas para selecionar os administradores e usuários das chaves ao criar chaves-réplicas, nem adicionará as instruções de política correspondentes. Assim sendo, você precisará adicioná-los manualmente.
- O AWS KMS console adiciona administradores de chaves à política de chaves sob o identificador "Allow access for Key Administrators" de instrução. A modificação desse identificador de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

14. (Opcional) Para evitar que os usuários e funções do IAM selecionados excluam essa chave do KMS, na seção Exclusão de chaves na parte inferior da página, desmarque a caixa de seleção Permitir que os administradores de chaves excluam essa chave.

15. Escolha Próximo.

16. Selecione os usuários e as funções do IAM que podem usar a chave do KMS para [operações de criptografia](#).

 Observações

O AWS KMS console adiciona os principais usuários à política de chaves sob os identificadores de declaração "Allow use of the key" e "Allow attachment of persistent resources". A modificação desses identificadores de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

17. (Opcional) Você pode permitir que outras Contas da AWS usem essa chave KMS para operações criptográficas. Para fazer isso, na seção Outras Contas da AWS, no fim da página, escolha Adicionar outra Conta da AWS e insira o número de identificação de Conta da AWS de uma conta externa. Para adicionar várias contas externas, repita essa etapa.

 Note

Para permitir que as entidades principais de contas externas usem a chave do KMS, os administradores da conta externa devem criar políticas do IAM que forneçam essas permissões. Para obter mais informações, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

18. Escolha Próximo.
19. Consulte as instruções da política de chave para a chave. Para fazer alterações na política de chave, selecione Editar.
20. Escolha Próximo.
21. Revise as configurações que você escolheu. Ainda é possível voltar e alterar todas as configurações.
22. Escolha Finalizar para criar a chave primária multirregional.

Usando a AWS KMS API

Para criar uma chave primária multirregional, use a [CreateKey](#) operação. Use também o parâmetro `MultiRegion` com um valor de `True`.

Por exemplo, o comando a seguir cria uma chave primária multirregional na chave do chamador (Região da AWS us-east-1). Ele aceita valores padrão para todas as outras propriedades, incluindo

a política de chaves. Os valores padrão para chaves primárias de várias regiões são os mesmos que os valores padrão para todas as outras chaves do KMS, incluindo a [política de chaves padrão](#). Este procedimento cria uma chave de criptografia simétrica, a chave padrão do KMS.

A resposta inclui o elemento `MultiRegion` e o elemento `MultiRegionConfiguration` com subelementos típicos e valores para uma chave primária de várias regiões sem chaves de réplica. O [ID de chave](#) de uma chave de várias regiões sempre começa com `mrk-`.

 Important

Não inclua informações confidenciais ou sigilosas nos campos `Description` ou `Tags`. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

```
$ 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": [ ]
    }
  }
}
```

```
}  
  }  
}
```

Criar chaves de réplica de várias regiões

[Você pode criar uma chave de réplica multirregional no AWS KMS console, usando a `ReplicateKey` operação ou usando um `AWS::KMS::ReplicaKey` CloudFormation modelo.](#) Você não pode usar a `CreateKey` operação para criar uma chave de réplica.

Você pode usar esses procedimentos para replicar qualquer chave primária de várias regiões, inclusive uma [chave do KMS de criptografia simétrica](#), uma [chave do KMS de criptografia assimétrica](#) ou uma [chave do KMS de Hash-based message authentication code \(HMAC – Código de autenticação de mensagem por hash\)](#).

Quando essa operação for concluída, a nova chave de réplica terá um [estado de chave](#) transitório de `Creating`. Esse estado de chave mudará para `Enabled` (ou `PendingImport` se você criar uma chave de várias regiões com [material de chave importado](#)) após alguns segundos quando o processo de criação da nova chave de réplica estiver concluído. Enquanto o estado da chave for `Creating`, você poderá gerenciar a chave, mas ainda não poderá usá-la em operações de criptografia. Se você estiver criando e usando a chave de réplica programaticamente, tente novamente `KMSInvalidStateException` ou ligue [DescribeKey](#) para verificar seu `KeyState` valor antes de usá-la.

Se você excluir uma chave de réplica por engano, poderá usar esse procedimento para recriá-la. Se você replicar a mesma chave primária na mesma região, a nova chave de réplica criada terá as mesmas [propriedades compartilhadas](#) que a chave de réplica original.

Important

Não inclua informações confidenciais ou sigilosas no alias, na descrição ou nas tags. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

Para usar um AWS CloudFormation modelo para criar uma chave de réplica, consulte [AWS::KMS::ReplicaKey](#) Guia do AWS CloudFormation usuário.

Etapa 1: escolher regiões de réplica

Normalmente, você escolhe replicar uma chave multirregional em uma Região da AWS com base em seu modelo de negócios e requisitos regulatórios. Por exemplo, você pode replicar uma chave em regiões nas quais você mantém seus recursos. Ou, para atender a um requisito de recuperação de desastres, você pode replicar uma chave em regiões geograficamente distantes.

A seguir estão os AWS KMS requisitos para regiões de réplica. Se a região escolhida não atender a esses requisitos, as tentativas de replicar uma chave falharão.

- Uma chave de várias regiões relacionada por região — Não é possível criar uma chave de réplica na mesma região que a chave primária ou na mesma região que outra réplica da chave primária.

Se você tentar replicar uma chave primária em uma região que já tenha uma réplica dessa chave primária, a tentativa falhará. Se a chave de réplica atual na região estiver no [estado de chave PendingDeletion](#), você poderá [cancelar a exclusão da chave de réplica](#) ou aguardar até que a chave de réplica seja excluída.

- Várias chaves de várias regiões não relacionadas na mesma região — É possível ter várias chaves de várias regiões não relacionadas na mesma região. Por exemplo, você pode ter duas chaves primárias de várias regiões na região us-east-1. Cada uma das chaves primárias pode ter uma chave de réplica na região us-west-2.
- Regiões na mesma partição — A região da chave de réplica deve estar na mesma [partição da AWS](#) que a região da chave primária.
- A região deve estar habilitada — Se uma região estiver [desabilitada por padrão](#), você não poderá criar nenhum recurso nessa região até que ela esteja habilitada para a sua Conta da AWS.

Etapa 2: criar chaves de réplica

Note

Ao criar chaves-réplicas, considere cuidadosamente os usuários e os perfis do IAM que você seleciona para administrar e usar a chave-réplica. As políticas do IAM podem conceder permissão para gerenciar a chave do KMS a outros usuários e funções do IAM. As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais

temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Usando o AWS KMS console

No AWS KMS console, você pode criar uma ou várias réplicas de uma chave primária multirregional na mesma operação.

Esse procedimento é semelhante à criação de uma chave do KMS de região única padrão no console. No entanto, como uma chave de réplica é baseada na chave primária, você não seleciona valores para [propriedades compartilhadas](#), como a especificação da chave (simétrica ou assimétrica), o uso da chave ou a origem da chave.

Você especifica propriedades que não são compartilhadas, incluindo alias, etiquetas, descrição e política de chaves. Por conveniência, o console mostra os valores de propriedade atuais da chave primária, mas é possível alterá-los. Mesmo que você mantenha os valores da chave primária, AWS KMS não os mantém sincronizados.

Important

Não inclua informações confidenciais ou sigilosas no alias, na descrição ou nas tags. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Customer managed keys (Chaves gerenciadas de cliente).
4. Selecione o ID de chave ou alias de uma [chave primária de várias regiões](#). Isso abre a página de detalhes da chave do KMS.

Para identificar uma chave primária de várias regiões, use o ícone de ferramenta no canto superior direito para adicionar a coluna Regionality (Regionalidade) à tabela.

5. Escolha a guia Regionality (Regionalidade).
6. Na seção Related multi-Region keys (Chaves de várias regiões relacionadas), selecione Create new replica keys (Criar novas chaves de réplica).

A seção Related multi-Region keys (Chaves de várias regiões relacionadas) mostra a região da chave primária e suas chaves de réplica. Você pode usar essa tela para ajudar a escolher a região da sua nova chave de réplica.

7. Escolha uma ou mais Regiões da AWS. Esse procedimento cria uma chave de réplica em cada uma das regiões selecionadas.

O menu inclui somente regiões na mesma AWS partição da chave primária. As regiões que já têm uma chave de várias regiões relacionada são exibidas, mas podem ser selecionadas. Talvez você não tenha permissão para replicar uma chave em todas as regiões no menu.

Quando terminar de escolher regiões, feche o menu. As regiões escolhidas são exibidas. Para cancelar a replicação em uma região, escolha a opção X ao lado do nome da região.

8. Digite um [alias](#) para a chave de réplica.

O console exibe um dos aliases atuais da chave primária, mas você pode alterá-lo. Você pode dar à chave primária de várias regiões e suas réplicas o mesmo alias ou alias diferentes. Os aliases não são uma [propriedade compartilhada das chaves](#) multirregionais. AWS KMS não sincroniza os aliases das chaves multirregionais.

Adicionar, excluir ou atualizar um alias pode conceder ou negar uma permissão à chave do KMS. Para obter mais detalhes, consulte [ABAC para AWS KMS](#) e [Usar aliases para controlar o acesso a chaves do KMS](#).

9. (Opcional) Digite uma descrição da chave de réplica.

O console exibe a descrição atual da chave primária, mas é possível alterá-la. Descrições não são uma propriedade compartilhada de chaves de várias regiões. Você pode dar à sua chave primária multirregional e suas réplicas a mesma descrição ou descrições diferentes. AWS KMS não sincroniza as descrições das chaves de várias regiões.

10. (Opcional) Digite uma chave de etiqueta e um valor de etiqueta opcional. Para atribuir mais de uma etiqueta à chave de réplica, escolha Add tag (Adicionar etiqueta).

O console mostra as etiquetas atualmente anexadas à chave primária, mas é possível alterá-las. Tags não são uma propriedade compartilhada de chaves de várias regiões. Você pode atribuir à sua chave primária multirregional e suas réplicas as mesmas tags ou tags diferentes. AWS KMS não sincroniza as tags das chaves multirregionais.

Marcar ou desmarcar uma chave do KMS pode conceder ou negar uma permissão a essa chave do KMS. Para obter mais detalhes, consulte [ABAC para AWS KMS](#) e [Usar tags para controlar o acesso a chaves do KMS](#).

11. Selecione os usuários e as funções do IAM que podem administrar a chave de réplica.

Observações

- Se você modificou a política de chave padrão ao criar a chave primária multirregional, o console não solicitará que selecione os administradores ou usuários de chave (etapas 11 a 15) durante a criação da chave-réplica. Nesse caso, você precisará adicionar manualmente as permissões necessárias para administradores e usuários de chave à política de chave selecionando Editar na etapa Editar política de chave (etapa 17).
- Essa etapa inicia o processo de criação de uma [política de chaves](#) para a chave de réplica. O console exibe a política de chaves atual da chave primária, mas é possível alterá-la. Políticas de chaves não são uma propriedade compartilhada de chaves de várias regiões. Você pode fornecer à chave primária de várias regiões e suas réplicas a mesma política de chaves ou políticas de chave diferentes. O AWS KMS não sincroniza políticas de chaves. Você pode alterar a política de chaves de qualquer chave do KMS a qualquer momento.
- O AWS KMS console adiciona administradores de chaves à política de chaves sob o identificador "Allow access for Key Administrators" de instrução. A modificação desse identificador de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

12. (Opcional) Para evitar que os usuários e funções do IAM selecionados excluam essa chave do KMS, na seção Exclusão de chaves na parte inferior da página, desmarque a caixa de seleção Permitir que os administradores de chaves excluam essa chave.
13. Escolha Próximo.
14. Selecione os usuários e as funções do IAM que podem usar a chave do KMS para [operações de criptografia](#).

 Observação

O AWS KMS console adiciona os principais usuários à política de chaves sob os identificadores de declaração "Allow use of the key" e "Allow attachment of persistent resources". A modificação desses identificadores de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

15. (Opcional) Você pode permitir que outras Contas da AWS pessoas usem essa chave KMS para operações criptográficas. Para fazer isso, na seção Outras Contas da AWS, no fim da página, escolha Adicionar outra Conta da AWS e insira o número de identificação de Conta da AWS de uma conta externa. Para adicionar várias contas externas, repita essa etapa.

 Note

Para permitir que as entidades principais de contas externas usem a chave do KMS, os administradores da conta externa devem criar políticas do IAM que forneçam essas permissões. Para obter mais informações, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

16. Escolha Próximo.
17. Consulte as instruções da política de chave para a chave. Para fazer alterações na política de chave, selecione Editar.
18. Escolha Próximo.
19. Revise as configurações que você escolheu. Ainda é possível voltar e alterar todas as configurações.
20. Escolha Finalizar para criar a chave- réplica multirregional.

Usando a AWS KMS API

Para criar uma chave de réplica multirregional, use a [ReplicateKey](#) operação. Você não pode usar a [CreateKey](#) operação para criar uma chave de réplica. Essa operação cria uma chave de réplica por vez. A região que você especificar deve estar em conformidade com os [Requisitos de região](#) para chaves de réplica.

Ao usar a operação `ReplicateKey`, não especifique valores para [propriedades compartilhadas](#) de chaves de várias regiões. Valores de propriedades compartilhadas são copiados da chave

primária e mantidos sincronizados. No entanto, é possível especificar valores para propriedades não compartilhadas. Caso contrário, AWS KMS aplica os valores padrão padrão para as chaves KMS, não os valores da chave primária.

Note

Se você não especificar valores para os Tags parâmetros `Description`, ou `KeyPolicy`, AWS KMS cria a chave de réplica com uma descrição de string vazia, a [política de chaves padrão](#) e sem tags.

Não inclua informações confidenciais ou sigilosas nos campos `Description` ou `Tags`.

Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

Por exemplo, o comando a seguir cria uma chave de várias regiões na região Ásia-Pacífico (Sydney) (ap-southeast-2). Essa chave de réplica é modelada na chave primária na região Leste dos EUA (Norte da Virgínia) (us-east-1), identificada pelo valor do parâmetro `KeyId`. Esse exemplo aceita valores padrão para todas as outras propriedades, incluindo a política de chaves.

A resposta descreve a nova chave de réplica. Ela inclui campos para propriedades compartilhadas, como `KeyId`, `KeySpec`, `KeyUsage` e a origem do material de chave (`Origin`). Ela também inclui propriedades que são independentes da chave primária, como `Description`, política de chaves (`ReplicaKeyPolicy`) e etiquetas (`ReplicaTags`).

A resposta também inclui o ARN de chave e a região da chave primária e todas as suas chaves de réplica, incluindo aquela que acabou de ser criada na região ap-southeast-2. Neste exemplo, o elemento `ReplicaKey` mostra que essa chave primária já foi replicada na região Europa (Irlanda) (eu-west-1).

```
$ aws kms replicate-key \
  --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
  --replica-region ap-southeast-2
{
  "ReplicaKeyMetadata": {
    "MultiRegion": true,
    "MultiRegionConfiguration": {
      "MultiRegionKeyType": "REPLICA",
      "PrimaryKey": {
        "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
```

```

        "Region": "us-east-1"
    },
    "ReplicaKeys": [
        {
            "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab",
            "Region": "ap-southeast-2"
        },
        {
            "Arn": "arn:aws:kms:eu-west-1:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab",
            "Region": "eu-west-1"
        }
    ]
},
"AWSAccountId": "111122223333",
"Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab",
"CreationDate": 1607472987.918,
"Description": "",
"Enabled": true,
"KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
"KeyManager": "CUSTOMER",
"KeySpec": "SYMMETRIC_DEFAULT",
"KeyState": "Enabled",
"KeyUsage": "ENCRYPT_DECRYPT",
"Origin": "AWS_KMS",
"CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
"EncryptionAlgorithms": [
    "SYMMETRIC_DEFAULT"
]
},
"ReplicaKeyPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-default-1\",...,
  \"ReplicaTags\": []
}

```

Criar uma chave do KMS com material de chave importado

O material de chaves importadas permite que você proteja seus AWS recursos com as chaves criptográficas que você gera. A visão geral a seguir explica o processo para importar seu material de

chaves para o AWS KMS. Para obter mais detalhes sobre cada etapa no processo, consulte o tópico correspondente.

1. [Crie uma chave do KMS sem material de chave](#) - A origem deve ser EXTERNAL. A origem da chave EXTERNAL indica que a chave foi projetada para material de chave importado e AWS KMS impede a geração de material de chave para a chave KMS. Em uma etapa posterior, você importará seu próprio material de chave para essa chave do KMS.

O material da chave que você importa deve ser compatível com a especificação da AWS KMS chave associada. Para obter mais informações sobre compatibilidade, consulte [the section called “Requisitos para material de chave importada”](#).

2. [Baixar a chave pública de empacotamento e o token de importação](#) – depois de concluir a etapa 1, baixe uma chave pública de empacotamento e um token de importação. Esses itens protegem seu material principal enquanto ele é importado para AWS KMS o.

Nesta etapa, você escolhe o tipo (“especificação de chave”) para a chave de empacotamento RSA e o algoritmo de empacotamento que você usará para criptografar os dados em trânsito para o AWS KMS. É possível escolher uma especificação de chave de empacotamento e um algoritmo de chave de empacotamento diferentes sempre que importar ou reimportar o mesmo material de chave.

3. [Criptografar o material de chaves](#) – use a chave pública de empacotamento baixada na etapa 2 para criptografar o material de chaves que você criou no seu próprio sistema.
4. [Importar o material de chaves](#) – carregue o material de chave criptografado que você criou na etapa 3 e o token de importação que você baixou na etapa 2.

Nessa etapa, é possível [definir um prazo de validade opcional](#). Quando o material da chave importada expira, ele é AWS KMS excluído e a chave KMS fica inutilizável. Para continuar usando a chave do KMS, você deve reimportar o mesmo material de chave.

Quando a operação de importação é concluída com êxito, o estado da chave KMS muda de PendingImport para Enabled. Agora, você pode usar suas chaves do KMS em operações de criptografia.

AWS KMS registra uma entrada em seu AWS CloudTrail registro quando você [cria a chave KMS](#), [baixa a chave pública de empacotamento e o token de importação](#) e [importa o material da chave](#).

AWS KMS também registra uma entrada quando você exclui material de chave importado ou quando AWS KMS [exclui material de chave expirado](#).

Permissões para importar material de chave

Para criar e gerenciar chaves do KMS com material de chave importado, o usuário precisa de permissão para as operações nesse processo. Você pode fornecer as permissões `kms:GetParametersForImport`, `kms:ImportKeyMaterial` e `kms:DeleteImportedKeyMaterial` na política de chaves ao criar a chave do KMS. No AWS KMS console, essas permissões são adicionadas automaticamente para administradores de chaves quando você cria uma chave com uma origem de material de chave externa.

Para criar chaves do KMS com material de chave importado, a entidade principal precisa das permissões a seguir.

- [kms: CreateKey](#) (política do IAM)
 - Para limitar essa permissão às chaves KMS com material de chave importado, use a condição [kms: KeyOrigin](#) policy com um valor de. EXTERNAL

```
{
  "Sid": "CreateKMSKeysWithoutKeyMaterial",
  "Effect": "Allow",
  "Resource": "*",
  "Action": "kms:CreateKey",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL"
    }
  }
}
```

- [kms: GetParametersForImport](#) (Política de chave ou política do IAM)
 - Para limitar essa permissão a solicitações que usam um algoritmo de empacotamento específico e uma especificação de chave de encapsulamento, use as condições de política [kms: WrappingAlgorithm](#) e [kms: WrappingKeySpec](#)
- [kms: ImportKeyMaterial](#) (Política de chave ou política do IAM)
 - Para permitir ou proibir o material chave que expira e controlar a data de expiração, use as condições da política [kms: ExpirationModel](#) e [kms: ValidTo](#)

Para reimportar o material de chave importado, o diretor precisa das permissões [kms: GetParametersForImport](#) e [kms: ImportKeyMaterial](#)

Para excluir o material de chave importado, o diretor precisa de [kms: DeleteImportedKeyMaterial](#) permission.

Por exemplo, para permitir que o exemplo KMSAdminRole gerencie todos os aspectos de uma chave do KMS com material de chave importado, inclua uma declaração de política de chaves como a seguinte na política de chaves da chave do 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"
  ]
}
```

Requisitos para material de chave importada

O material da chave que você importa deve ser compatível com a [especificação de chave](#) da chave do KMS correspondente. Para pares de chaves assimétricas, importe somente a chave privada do par. AWS KMS deriva a chave pública da chave privada.

AWS KMS suporta as seguintes especificações principais para chaves KMS com material de chave importado.

- Chaves de criptografia simétrica
 - Especificação de chave:
 - SYMMETRIC_DEFAULT.
 - Requisitos:
 - 256 bits (32 bytes) de dados binários.
 - Nas regiões da China, elas devem ter 128 bits (16 bytes) de dados binários.
- Chaves de HMAC
 - Especificações de chave:
 - HMAC_224

- HMAC_256
- HMAC_384
- HMAC_512
- Requisitos:
 - O material da chave de HMAC deve estar em conformidade com a [RFC 2104](#).
 - O tamanho da chave deve corresponder ao especificado pela especificação de chave. O tamanho máximo da chave é de 1024 bits.
 - Se o material da chave exceder 1024 bits, você poderá fazer o hash do material da chave e importar a saída do hash. O algoritmo de hash deve corresponder à especificação de chave do KMS HMAC que você está criando.
- Exemplo:
 - Para importar 2048 bits de material de chave em uma chave HMAC_256, primeiro calcule o hash SHA-256 do material de chave de 2048 bits e, em seguida, importe a saída de hash de 256 bits resultante para a chave do KMS.
- Comprimentos de chave válidos:
 - HMAC_224: 224 a 1024 bits
 - HMAC_256: 256 a 1024 bits
 - HMAC_384: 384 a 1024 bits
 - HMAC_512: 512 a 1024 bits
- Chave privada assimétrica RSA
 - Especificações de chave:
 - RSA_2048
 - RSA_3072
 - RSA_4096
 - Requisitos:
 - [A chave privada assimétrica RSA que você importa deve fazer parte de um par de chaves que esteja em conformidade com a RFC 3447.](#)
 - Módulo: 2048 bits, 3072 bits ou 4096 bits
 - Número de primes: 2 (chaves RSA com vários primes não são compatíveis)
 - O material de chave assimétrica deve ser codificado em BER ou em DER no formato Public-Key Cryptography Standards (PKCS) n.º 8, compatível com a [RFC 5208](#).

- Chave privada assimétrica de curva elíptica
 - Especificações de chave:
 - ECC_NIST_P256 (secp256r1)
 - ECC_NIST_P384 (secp384r1)
 - ECC_NIST_P521 (secp521r1)
 - ECC_SECG_P256K1 (secp256k1)
 - EDWARDS25519 ECC_NIST_ (ed25519)
 - Requisitos:
 - A chave privada assimétrica ECC que você importa deve fazer parte de um par de chaves que esteja em conformidade com a [RFC 5915](#).
 - Curva: NIST P-256, NIST P-384, NIST P-521, SECP256K1, NIST Ed25519.
 - Parâmetros: somente curvas nomeadas (chaves ECC com parâmetros explícitos são rejeitadas).
 - Coordenadas de ponto público: podem ser compactadas, não compactadas ou projetivas.
 - O material de chave assimétrica deve ser codificado em BER ou em DER no formato Public-Key Cryptography Standards (PKCS) n.º 8, compatível com a [RFC 5208](#).
- Chave ML-DSA
 - Especificações de chave:
 - ML_DSA_44
 - ML_DSA_65
 - ML_DSA_87

 Important

A importação de chaves ML-DSA não é compatível.

- SM2 chave privada assimétrica (somente regiões da China)
 - Requisitos:
 - A chave privada SM2 assimétrica que você importa deve fazer parte de um par de chaves que esteja em conformidade com 0003. GM/T
 - Curva: SM2.
 - Parâmetros: somente curva nomeada (SM2 chaves com parâmetros explícitos são rejeitadas).
 - Coordenadas de ponto público: podem ser compactadas, não compactadas ou projetivas.

- O material de chave assimétrica deve ser codificado em BER ou em DER no formato Public-Key Cryptography Standards (PKCS) n.º 8, compatível com a [RFC 5208](#).

Etapa 1: criar um material AWS KMS key sem chave

Por padrão, AWS KMS cria material de chave para você quando você cria uma chave KMS. Se preferir importar seu próprio material de chave, comece criando uma chave do KMS sem material de chave. Em seguida, importe o material de chave. Para criar uma chave KMS sem material de chave, use o AWS KMS console ou a [CreateKey](#) operação.

Para criar uma chave sem material de chave, especifique uma [origem](#) como EXTERNAL. A propriedade de origem de uma chave do KMS é imutável. Depois de criá-la, você não poderá converter uma chave KMS projetada para material de chave importado em uma chave KMS com material de chave de AWS KMS ou de qualquer outra fonte.

O [key state](#) (estado de chave) de uma chave do KMS com uma origem EXTERNAL e sem material de chave é PendingImport. Uma chave do KMS pode permanecer no estado PendingImport indefinidamente. No entanto, não é possível usar uma chave do KMS no estado PendingImport em operações criptográficas. Quando o material da chave é importado, o estado da chave do KMS muda para Enabled, e você pode usar a chave do KMS em operações de criptografia.

AWS KMS registra um evento em seu AWS CloudTrail registro quando você [cria a chave KMS](#), [baixa a chave pública e o token de importação e importa o material da chave](#). AWS KMS também registra um CloudTrail evento quando você [exclui material de chave importado](#) ou quando AWS KMS [exclui material de chave expirado](#).

Tópicos

- [Criar uma chave do KMS sem material de chave \(console\)](#)
- [Criação de uma chave KMS sem material de chave \(AWS KMS API\)](#)

Criar uma chave do KMS sem material de chave (console)

Você somente precisa criar uma chave do KMS para o material de chave importado uma vez. É possível importar/reimportar o mesmo material de chaves para as chaves do KMS quantas vezes desejar, mas não é possível importar outro material de chaves para uma chave do KMS. Para obter detalhes, consulte [Etapa 2: Fazer download da chave pública de empacotamento e do token de importação](#).

Para encontrar chaves do KMS existentes com material de chave importado em sua tabela Customer managed keys (Chaves gerenciadas pelo cliente), use o ícone de engrenagem no canto superior direito para mostrar a coluna Origin (Origem) na lista de chaves do KMS. As chaves importadas têm um valor Origin (Origem) External (Externo) (Importar material de chave).

Para criar uma chave do KMS com material de chave importado, comece seguindo as [instruções para criar uma chave do KMS do seu tipo de chave preferido](#), com a exceção a seguir.

Depois de escolher o uso de chave, faça o seguinte:

1. Expanda Advanced options (Opções avançadas).
2. Em Key material origin (Origem do material de chave), escolha External (Import key material) (Externo [material de chave importado]).
3. Marque a caixa de seleção ao lado de I understand the security, availability, and durability implications of using an imported key (Entendo as implicações de segurança, disponibilidade e durabilidade do uso de uma chave importada) para indicar que você compreende as implicações do uso de material de chaves importadas. Para ler sobre essas implicações, consulte [Proteger o material de chave importada](#).
4. Opcional: para criar uma [chave do KMS de várias regiões](#) com material de chave importado, em Regionalidade, selecione Chave de várias regiões.
5. Retorne às instruções básicas. As etapas restantes do procedimento básico são as mesmas para todas as chaves do KMS desse tipo.

Ao escolher Concluir, você criou uma chave do KMS sem material de chave e um status ([estado da chave](#)) de importação pendente.

No entanto, em vez de retornar à tabela de chaves gerenciadas pelo cliente, o console exibirá uma página na qual é possível baixar a chave pública e o token de importação necessários para importar o material da chave. É possível continuar com a etapa de download agora ou escolher Cancelar para parar nesse momento. É possível retornar a essa etapa de download a qualquer momento.

Próximo: [Etapa 2: Fazer download da chave pública de empacotamento e do token de importação](#).

Criação de uma chave KMS sem material de chave (AWS KMS API)

Para usar a [AWS KMS API](#) para criar uma chave KMS de criptografia simétrica sem material de chave, envie uma [CreateKey](#) solicitação com o Origin parâmetro definido como. EXTERNAL O exemplo a seguir mostra como fazer isso com a [AWS Command Line Interface \(AWS CLI\)](#).

```
$ aws kms create-key --origin EXTERNAL
```

Quando o comando é bem-sucedido, a saída é semelhante à seguinte. A AWS KMS chave `Origin` é `EXTERNAL` e `KeyState` é `PendingImport`.

Tip

Se o comando não for bem-sucedido, você poderá ver uma `KMSInvalidStateException` ou `NotFoundException`. Você poderá repetir a solicitação.

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

Copie o valor `KeyId` da saída do seu comando para usar em etapas posteriores e prossiga para [Etapa 2: Fazer download da chave pública de empacotamento e do token de importação](#).

Note

Esse comando cria uma chave do KMS de criptografia simétrica com uma `KeySpec` de `SYMMETRIC_DEFAULT` e `KeyUsage` de `ENCRYPT_DECRYPT`. É possível usar os parâmetros

opcionais --key-spec e --key-usage para criar uma chave assimétrica ou HMAC KMS. Para obter mais informações, consulte a operação [CreateKey](#).

Etapa 2: Fazer download da chave pública de empacotamento e do token de importação

Depois de [criar um material AWS KMS key sem chave](#), baixe uma chave pública de empacotamento e um token de importação para essa chave KMS usando o AWS KMS console ou a [GetParametersForImport](#) API. A chave pública de empacotamento e o token de importação são um conjunto indivisível que deve ser usado em conjunto.

Você usará a chave pública de empacotamento para [criptografar seu material de chave](#) para transporte. [Antes de baixar um par de chaves de agrupamento RSA, você seleciona o comprimento \(especificação de chave\) do par de chaves de encapsulamento RSA e o algoritmo de empacotamento que você usará para criptografar seu material de chave importado para transporte na etapa 3.](#) AWS KMS também suporta a especificação da chave de SM2 embalagem (somente regiões da China).

Cada conjunto de chave pública de empacotamento e de token de importação é válido por 24 horas. Se você não usá-los para importar material de chaves em até 24 horas após o download, será necessário fazer download de um novo conjunto. É possível fazer download de uma nova chave pública de empacotamento e de conjuntos de tokens de empacotamento a qualquer momento. Isso permitirá que você altere o comprimento da chave de empacotamento RSA (“especificação da chave”) ou substitua um conjunto perdido.

Também é possível baixar uma chave pública de empacotamento e o conjunto de tokens de importação para [reimportar o mesmo material de chave](#) para uma chave do KMS. É possível fazer isso para definir ou alterar o tempo de expiração do material de chaves ou para restaurar o material de chaves expirado ou excluído. Você deve baixar e recriptografar seu material de chave toda vez que importá-lo para o AWS KMS

Uso da chave pública de empacotamento

O download inclui uma chave pública exclusiva para você Conta da AWS, também chamada de chave pública de empacotamento.

Antes de importar o material da chave, você criptografa o material da chave com a chave de empacotamento pública e, em seguida, carrega o material da chave criptografada para AWS

KMS Ao AWS KMS receber seu material de chave criptografada, ele descriptografa o material de chave com a chave privada correspondente e, em seguida, recriptografa o material de chave sob uma chave simétrica AES, tudo dentro de um módulo de segurança de AWS KMS hardware (HSM).

Uso do token de importação

O download inclui um token de importação que contém metadados para garantir que o seu material de chaves seja importado corretamente. Ao carregar seu material de chave criptografada para AWS KMS, você deve carregar o mesmo token de importação que você baixou nesta etapa.

Selecionar uma especificação de chave pública de empacotamento

Para proteger seu material de chaves durante a importação, você o criptografa usando uma chave pública de agrupamento da qual você baixa e um AWS KMS algoritmo de [agrupamento](#) compatível. Selecione uma especificação de chave antes de fazer download da chave pública de empacotamento e do token de importação. Todos os pares de chaves de empacotamento são gerados em módulos de segurança de AWS KMS hardware (HSMs). A chave privada nunca sai do HSM em texto simples.

Especificações da chave de empacotamento RSA

A especificação de chave da chave pública de empacotamento determina o comprimento das chaves no par de chaves RSA que protege o material de chave durante o transporte para o AWS KMS. Em geral, recomendamos o uso da chave pública de empacotamento mais longa possível. Oferecemos várias especificações de chave pública de embalagem para oferecer suporte a uma variedade de gerenciadores HSMs de chaves.

AWS KMS suporta as seguintes especificações principais para as chaves de empacotamento RSA usadas para importar material de chaves de todos os tipos, exceto conforme indicado.

- RSA_4096 (recomendado)
- RSA_3072
- RSA_2048

Note

A combinação a seguir NÃO é compatível: material de chave ECC_NIST_P521, especificação de chave de empacotamento pública RSA_2048 e um algoritmo de empacotamento RSAES_OAEP_SHA_*.

Não é possível empacotar diretamente o material de chave ECC_NIST_P521 com uma chave de empacotamento pública RSA_2048. Use uma chave de empacotamento maior ou um algoritmo de empacotamento RSA_AES_KEY_WRAP_SHA_*.

SM2 especificação da chave de embalagem (somente regiões da China)

AWS KMS suporta a seguinte especificação de chave para as chaves de SM2 empacotamento usadas para importar material de chave assimétrico.

- SM2

Selecionar um algoritmo de empacotamento

Para proteger o material de chaves durante a importação, criptografe-o usando a chave pública de empacotamento baixada e um algoritmo de empacotamento.

AWS KMS suporta vários algoritmos de empacotamento RSA padrão e um algoritmo de empacotamento híbrido de duas etapas. Em geral, recomendamos usar o algoritmo de empacotamento mais seguro que seja compatível com o material da chave importada e com a [especificação da chave de empacotamento](#). Normalmente, você escolhe um algoritmo que é compatível com o módulo de segurança de hardware (HSM) ou com o sistema de gerenciamento de chaves que protege o material de chaves.

A tabela a seguir mostra os algoritmos de empacotamento compatíveis com cada tipo de material de chave e chave do KMS. Os algoritmos estão listados em ordem de preferência.

Material de chave	Algoritmo e especificação de empacotamento compatíveis
Chave de criptografia simétrica	Algoritmos de empacotamento:
Chave AES de 256 bits	RSAES_OAEP_SHA_256
SM4 Chave de 128 bits (somente regiões da China)	RSAES_OAEP_SHA_1
	Algoritmos de empacotamento obsoletos:
	RAIS_ _V1 PKCS1

Material de chave	Algoritmo e especificação de empacotamento compatíveis
	 Note Em 10 de outubro de 2023, AWS KMS não oferece suporte ao algoritmo de empacotamento RSAES_PKCS1_V1_5. Especificações da chave de empacotamento: RSA_2048 RSA_3072 RSA_4096
Chave privada RSA assimétrica	Algoritmos de empacotamento: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 SM2PKE (somente regiões da China) Especificações da chave de empacotamento: RSA_2048 RSA_3072 RSA_4096 SM2 (Somente regiões da China)

Material de chave	Algoritmo e especificação de empacotamento compatíveis
Chave privada assimétrica de curva elíptica (ECC) Não é possível usar os algoritmos de empacotamento RSAES_OAEP_SHA_* com a especificação da chave de empacotamento RSA_2048 para empacotar o material da chave ECC_NIST_P521.	Algoritmos de empacotamento: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (somente regiões da China) Especificações da chave de empacotamento: RSA_2048 RSA_3072 RSA_4096 SM2 (Somente regiões da China)
Chave SM2 privada assimétrica (somente regiões da China)	Algoritmos de empacotamento: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (somente regiões da China) Especificações da chave de empacotamento: RSA_2048 RSA_3072 RSA_4096 SM2 (Somente regiões da China)

Material de chave	Algoritmo e especificação de empacotamento compatíveis
Chave de HMAC	Algoritmos de empacotamento: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Especificações da chave de empacotamento: RSA_2048 RSA_3072 RSA_4096

 Note

Os algoritmos de empacotamento RSA_AES_KEY_WRAP_SHA_256 e RSA_AES_KEY_WRAP_SHA_1 não são compatíveis em regiões da China.

- RSA_AES_KEY_WRAP_SHA_256 - Um algoritmo de empacotamento híbrido de duas etapas que combina a criptografia do seu material de chave com uma chave simétrica AES que você gera e, em seguida, criptografa a chave simétrica AES com a chave de empacotamento pública RSA baixada e o algoritmo de empacotamento RSAES_OAEP_SHA_256.

É necessário um algoritmo de empacotamento RSA_AES_KEY_WRAP_SHA_* para empacotar material de chave privada RSA, exceto em regiões da China, nas quais é necessário usar o algoritmo de empacotamento SM2PKE.

- RSA_AES_KEY_WRAP_SHA_1 - Um algoritmo de empacotamento híbrido de duas etapas que combina a criptografia do seu material de chave com uma chave simétrica AES que você gera e, em seguida, criptografa a chave simétrica AES com a chave pública de empacotamento RSA baixada e o algoritmo de empacotamento RSAES_OAEP_SHA_1.

É necessário um algoritmo de empacotamento RSA_AES_KEY_WRAP_SHA_* para empacotar material de chave privada RSA, exceto em regiões da China, nas quais é necessário usar o algoritmo de empacotamento SM2PKE.

- **RSAES_OAEP_SHA_256** — O algoritmo de criptografia RSA com Preenchimento da criptografia assimétrica ideal (OAEP) e função de hash SHA-256.
- **RSAES_OAEP_SHA_1** — O algoritmo de criptografia RSA com Preenchimento da criptografia assimétrica ideal (OAEP) e função de hash SHA-1.
- **RSAES_PKCS1_V1_5**(Obsoleto; em 10 de outubro de 2023, AWS KMS não oferece suporte ao algoritmo de encapsulamento RSAES_PKCS1_V1_5) — O algoritmo de criptografia RSA com o formato de preenchimento definido no PKCS #1 Versão 1.5.
- **SM2PKE**(Somente regiões da China) — Um algoritmo de criptografia baseado em curva elíptica definido pela OSCCA em 0003.4-2012. GM/T

Tópicos

- [Fazer download da chave pública de empacotamento e do token de importação \(console\)](#)
- [Baixando a chave pública de empacotamento e o token de importação \(AWS KMS API\)](#)

Fazer download da chave pública de empacotamento e do token de importação (console)

Você pode usar o AWS KMS console para baixar a chave pública de empacotamento e o token de importação.

1. Se você acabou de concluir as etapas para [criar uma chave do KMS sem material de chave](#) e está na página Download wrapping key and import token (Baixar chave de empacotamento e token de importação), vá para [Step 10](#).
2. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
3. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
4. No painel de navegação, escolha Chaves gerenciadas pelo cliente.

Tip

É possível importar material de chave somente para uma chave do KMS simétrica com uma Origin (Origem) de External (Import key material) (Externa [Importar material de chave]). Isso indica que a chave do KMS foi criada sem material de chave. Para adicionar a coluna Origin (Origem) à tabela, no canto superior direito da página, selecione o ícone de configurações



Ative Origin (Origem) e escolha Confirm (Confirmar).

5. Escolha o alias ou o ID da chave do KMS com importação pendente.
6. Expanda a seção Cryptographic configuration (Configuração criptográfica) e visualize seus valores. As guias estão abaixo da seção General configuration (Configuração geral).

É possível importar material de chave somente para uma chave do KMS com uma Origin (Origem) de External (Import key material) (Externa [Importar material de chave]). Para obter informações sobre como criar chaves do KMS com material de chave importado, consulte [Importação de material chave para AWS KMS chaves](#).

7. Escolha a guia apropriada com base no seu tipo de chave.
 - Para chaves assimétricas e HMAC, escolha a guia Material da chave.
 - Para chaves de criptografia simétricas, escolha a guia Material da chave e rotações.
8. Escolha a ação de importação.
 - Para chaves assimétricas e HMAC, escolha Importar material de chave.
 - Para chaves de criptografia simétricas, escolha uma das seguintes opções:
 - Importar material de chave inicial (se nenhum material de chave tiver sido importado ainda)
 - Importar novo material chave (para adicionar novo material para rotação)
 - Reimportar material chave (disponível no menu Ações na tabela de materiais principais)

Note

Para chaves multirregionais, você deve primeiro importar o novo material de chave para a chave de região primária. Em seguida, importe o mesmo material de chave em cada chave de região de réplica.

Para chaves multirregionais primárias, a tabela de materiais principais inclui uma coluna de estado de importação da réplica que exibe o status da importação em todas as regiões da réplica (por exemplo, “0 de 3 importados”). Escolha o valor do estado de importação da réplica para abrir um modal que mostre o status de importação de cada região da réplica. O modal fornece links de importação de material de chave para regiões de réplica onde o novo material de chave não foi importado.

9. Em Selecionar especificação da chave de empacotamento, escolha a configuração da sua chave do KMS. Após a criação dessa chave, não será possível alterar a especificação da chave.
10. Para Select wrapping algorithm, escolha a opção que você usará para criptografar o material de chaves. Para obter mais informações sobre as opções, consulte [Selecionar um algoritmo de empacotamento](#).
11. Escolha Download wrapping key and import token (fazer download da chave de empacotamento e token de importação) e salve o arquivo.

Se houver a opção Next (Próximo), para continuar o processo agora, selecione Next (Próximo). Para continuar mais tarde, selecione Cancel (Cancelar).

12. Descompacte o arquivo .zip que você salvou na etapa anterior (Import_Parameters_<key_id>_<timestamp>).

A pasta contém os seguintes arquivos:

- Uma chave pública de empacotamento em um arquivo chamado WrappingPublicKey.bin.
- Um token de importação em um arquivo chamado ImportToken.bin.
- Um arquivo de texto chamado README.txt. Este arquivo contém informações sobre a chave pública de empacotamento, o algoritmo de encapsulamento a ser usado para criptografar o material de chaves e a data e hora em que a chave pública de empacotamento e o token de importação expiram.

13. Para continuar o processo, consulte [criptografe o material de chaves](#).

Baixando a chave pública de empacotamento e o token de importação (AWS KMS API)

Para baixar a chave pública e o token de importação, use a [GetParametersForImport](#) API.

Especifique a chave do KMS que será associada ao material de chave importado. Essa chave do KMS deve ter um valor de [Origem](#) de EXTERNAL.

Note

Não é possível importar material de chave para chaves do KMS ML-DSA.

Este exemplo especifica o algoritmo de empacotamento RSA_AES_KEY_WRAP_SHA_256, a especificação de chave pública de empacotamento RSA_3072 e um exemplo de ID de chave.

Substitua esses valores de exemplo por valores válidos para download. Para o ID de chave, é possível usar o [ID da chave](#) ou o [ARN da chave](#), mas não um [alias do nome](#) ou [alias do ARN](#) nessa operação.

```
$ 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
```

Quando o comando é bem-sucedido, a saída é semelhante à seguinte:

```
{  
  "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)"  
}
```

Para preparar os dados para a próxima etapa, o base64 decodifica a chave pública e o token de importação e salva os valores decodificados nos arquivos.

Para o base64 decodificar a chave pública e o token de importação:

1. Copie a chave pública codificada em base64 (representada **public key (base64 encoded)** na saída do exemplo), cole-a em um novo arquivo e salve o arquivo. Dê um nome descritivo ao arquivo, como `PublicKey.b64`.
2. Use o [OpenSSL](#) para decodificar o conteúdo do arquivo base64 e salve os dados decodificados em um novo arquivo. O exemplo a seguir decodifica os dados no arquivo que você salvou na etapa anterior (`PublicKey.b64`) e salva a saída em um novo arquivo chamado `WrappingPublicKey.bin`.

```
$ openssl enc -d -base64 -A -in PublicKey.b64 -out WrappingPublicKey.bin
```

3. Copie o token de importação codificado em base64 (representado **import token (base64 encoded)** na saída do exemplo), cole-o em um novo arquivo e salve o arquivo. Dê ao arquivo um nome descritivo, por exemplo `importtoken.b64`.
4. Use o [OpenSSL](#) para decodificar o conteúdo do arquivo base64 e salve os dados decodificados em um novo arquivo. O exemplo a seguir decodifica os dados no arquivo que você salvou

na etapa anterior (`ImportToken.b64`) e salva a saída em um novo arquivo chamado `ImportToken.bin`.

```
$ openssl enc -d -base64 -A -in importtoken.b64 -out ImportToken.bin
```

Vá para [Etapa 3: Criptografar o material de chave](#).

Etapa 3: Criptografar o material de chave

Depois de [fazer download da chave pública e importar o token](#), criptografe seu material de chaves usando a chave pública que você baixou e o algoritmo de empacotamento que você especificou. Se precisar substituir a chave pública ou o token de importação, ou alterar o algoritmo de empacotamento, você deverá baixar uma nova chave pública e importar o token. Para obter informações sobre as chaves públicas e os algoritmos de empacotamento que AWS KMS oferecem suporte, consulte [Selecionar uma especificação de chave pública de empacotamento](#) e [Selecionar um algoritmo de empacotamento](#).

O material de chaves deve estar em formato binário. Para obter informações detalhadas, consulte [Requisitos para material de chave importada](#).

Note

Para pares de chaves assimétricas, criptografe e importe somente a chave privada. AWS KMS deriva a chave pública da chave privada.

A combinação a seguir NÃO é compatível: material de chave `ECC_NIST_P521`, especificação de chave de empacotamento pública `RSA_2048` e um algoritmo de empacotamento `RSAES_OAEP_SHA_*`.

Não é possível empacotar diretamente o material de chave `ECC_NIST_P521` com uma chave de empacotamento pública `RSA_2048`. Use uma chave de empacotamento maior ou um algoritmo de empacotamento `RSA_AES_KEY_WRAP_SHA_*`.

Os algoritmos de empacotamento `RSA_AES_KEY_WRAP_SHA_256` e `RSA_AES_KEY_WRAP_SHA_1` não são compatíveis em regiões da China.

Geralmente, você criptografa o material de chaves ao exportá-lo do HSM (módulo de segurança de hardware) ou do sistema de gerenciamento de chaves. Para obter informações sobre como exportar material de chaves em formato binário, consulte a documentação do HSM ou do sistema

de gerenciamento de chaves. Você também pode consultar a seção a seguir, que fornece uma demonstração da prova de conceito usando OpenSSL.

Ao criptografar o material de chaves, use o mesmo algoritmo de empacotamento que você especificou quando [fez download da chave pública e do token de importação](#). Para encontrar o algoritmo de empacotamento que você especificou, consulte o evento de CloudTrail log da [GetParametersForImport](#) solicitação associada.

Gerar material de chave para testes

Os comandos do OpenSSL a seguir geram material de chave de cada tipo compatível para testes. Esses exemplos são fornecidos somente para testes e proof-of-concept demonstrações. Para sistemas de produção, use um método mais seguro para gerar o material de chaves, como um módulo de segurança de hardware ou um sistema de gerenciamento de chaves.

Para converter as chaves privadas de pares de chaves assimétricas em formato codificado em DER, canalize o comando de geração de material de chave para o comando `openssl pkcs8` a seguir. O parâmetro `topk8` direciona o OpenSSL a usar uma chave privada como entrada e retornar uma chave formatada PKCS#8. (O comportamento padrão é o oposto.)

```
openssl pkcs8 -topk8 -outform der -nocrypt
```

Os comandos a seguir geram material de chave de teste para cada tipo de chave compatível.

- Chave de criptografia simétrica (32 bytes)

Esse comando gera uma chave simétrica de 256 bits (sequência aleatória de 32 bytes) e a salva no arquivo `PlaintextKeyMaterial.bin`. Não é necessário codificar esse material de chaves.

```
openssl rand -out PlaintextKeyMaterial.bin 32
```

Somente nas regiões da China, você deve gerar uma chave simétrica de 128 bits (sequência aleatória de 16 bytes).

```
openssl rand -out PlaintextKeyMaterial.bin 16
```

- Chaves de HMAC

Esse comando gera uma sequência de bytes aleatória do tamanho especificado. Não é necessário codificar esse material de chaves.

O comprimento da chave de HMAC deve corresponder ao comprimento definido pela especificação da chave do KMS. Por exemplo, se a chave do KMS for HMAC_384, você deverá importar uma chave de 384 bits (48 bytes).

```
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
```

- Chaves privadas 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
```

- Chaves privadas 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 chaves privadas (somente regiões da China)

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:sm2 | openssl pkcs8 -topk8 -
outform der -nocrypt > SM2_PrivateKey.der
```

Exemplos de material de chave de criptografia com OpenSSL

Os exemplos a seguir mostram como usar o [OpenSSL](#) para criptografar seu material de chaves com a chave pública que você baixou. Para criptografar seu material de chave usando uma chave SM2 pública (somente para regiões da China), use a [SM2OfflineOperationHelper](#) classe. Para obter mais informações sobre os tipos de material de chave compatíveis com cada algoritmo de empacotamento, consulte [the section called “Selecionar um algoritmo de empacotamento”](#).

Important

Esses exemplos são apenas uma demonstração da prova de conceito. Para sistemas de produção, use um método mais seguro (tal como um HSM ou sistema de gerenciamento de chaves comercial) para gerar e armazenar seu material de chaves.

A combinação a seguir NÃO é compatível: material de chave ECC_NIST_P521, especificação de chave de empacotamento pública RSA_2048 e um algoritmo de empacotamento RSAES_OAEP_SHA_*.

Não é possível empacotar diretamente o material de chave ECC_NIST_P521 com uma chave de empacotamento pública RSA_2048. Use uma chave de empacotamento maior ou um algoritmo de empacotamento RSA_AES_KEY_WRAP_SHA_*.

RSAES_OAEP_SHA_1

AWS KMS suporta o RSAES_OAEP_SHA_1 para chaves de criptografia simétricas (SYMMETRIC_DEFAULT), chaves privadas de curva elíptica (ECC), chaves privadas e chaves HMAC. SM2

O RSAES_OAEP_SHA_1 não é compatível com chaves privadas de RSA. Além disso, não é possível usar uma chave pública de empacotamento RSA_2048 com nenhum algoritmo de empacotamento RSAES_OAEP_SHA_* para empacotar uma chave privada ECC_NIST_P521 (secp521r1). Você deve usar uma chave pública de empacotamento maior ou um algoritmo de empacotamento RSA_AES_KEY_WRAP.

O exemplo a seguir criptografa seu material de chaves com a [chave pública que você baixou](#) e com o algoritmo de encapsulamento RSAES_OAEP_SHA_1 e o salva no arquivo `EncryptedKeyMaterial.bin`.

Neste exemplo:

- *WrappingPublicKey.bin* é o arquivo que contém a chave pública de empacotamento baixada.
- *PlaintextKeyMaterial.bin* é o arquivo que contém o material de chave que você está criptografando, como PlaintextKeyMaterial.bin, HMAC_384_PlaintextKey.bin ou 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 suporta o RSAES_OAEP_SHA_256 para chaves de criptografia simétricas (SYMMETRIC_DEFAULT), chaves privadas de curva elíptica (ECC), chaves privadas e chaves HMAC. SM2

O RSAES_OAEP_SHA_256 não é compatível com chaves privadas de RSA. Além disso, não é possível usar uma chave pública de empacotamento RSA_2048 com nenhum algoritmo de empacotamento RSAES_OAEP_SHA_* para empacotar uma chave privada ECC_NIST_P521 (secp521r1). Você deve usar uma chave pública maior ou um algoritmo de empacotamento RSA_AES_KEY_WRAP.

O exemplo a seguir criptografa o material de chaves com a [chave pública que você baixou](#) e com o algoritmo de empacotamento RSAES_OAEP_SHA_256 e o salva no arquivo EncryptedKeyMaterial.bin.

Neste exemplo:

- *WrappingPublicKey.bin* é o arquivo que contém a chave de empacotamento pública baixada. Se você fez download da chave pública do console, esse arquivo será chamado wrappingKey_KMS_key_key_ID_timestamp (por exemplo, wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909).

- *PlaintextKeyMaterial.bin* é o arquivo que contém o material de chave que você está criptografando, como `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin` ou `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

O algoritmo de empacotamento `RSA_AES_KEY_WRAP_SHA_1` envolve duas operações de criptografia.

1. Criptografe o material de chaves com uma chave simétrica AES que você gera e um algoritmo de criptografia simétrica AES.
2. Criptografe a chave simétrica AES que você usou com a chave pública que você baixou e o algoritmo de empacotamento `RSAES_OAEP_SHA_1`.

O algoritmo de empacotamento `RSA_AES_KEY_WRAP_SHA_1` requer OpenSSL versão 3.x ou superior.

1. Gerar uma chave de criptografia simétrica AES de 256 bits

Esse comando gera uma chave de criptografia simétrica AES que consiste em 256 bits aleatórios e a salva no arquivo `aes-key.bin`

```
# Generate a 32-byte AES symmetric encryption key  
$ openssl rand -out aes-key.bin 32
```

2. Criptografar o material de chave com a chave de criptografia simétrica AES

Esse comando criptografa o material da chave com a chave de criptografia simétrica AES e salva o material da chave criptografada no arquivo `key-material-wrapped.bin`.

Neste comando de exemplo:

- *PlaintextKeyMaterial.bin* é o arquivo que contém o material de chave que você está importando, como `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der` ou `ECC_NIST_P521_PrivateKey.der`.
- *aes-key.bin* é o arquivo que contém a chave de criptografia simétrica AES de 256 bits que você gerou no comando anterior.

```
# 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. Criptografar a chave de criptografia simétrica AES com a chave pública

Esse comando criptografa sua chave de criptografia simétrica AES com a chave pública que você baixou e o algoritmo de empacotamento `RSAES_OAEP_SHA_1`, a codifica em DER e a salva no arquivo `aes-key-wrapped.bin`.

Neste comando de exemplo:

- *WrappingPublicKey.bin* é o arquivo que contém a chave de empacotamento pública baixada. Se você fez download da chave pública do console, esse arquivo será chamado `wrappingKey_KMS key_key_ID_timestamp` (por exemplo, `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`).
- *aes-key.bin* é o arquivo que contém a chave de criptografia simétrica AES de 256 bits que você gerou no primeiro comando dessa sequência de exemplos.

```
# 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. Gerar o arquivo a ser importado

Concatene o arquivo com o material da chave criptografada e o arquivo com a chave AES criptografada. Salve-os no arquivo `EncryptedKeyMaterial.bin`, que é o arquivo que você importará no [Etapa 4: Importar o material de chave](#).

Neste comando de exemplo:

- *key-material-wrapped.bin* é o arquivo que contém o material de chaves criptografado.
- *aes-key-wrapped.bin* é o arquivo que contém a chave de criptografia AES criptografada.

```
# 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

O algoritmo de empacotamento `RSA_AES_KEY_WRAP_SHA_256` envolve duas operações de criptografia.

1. Criptografe o material de chaves com uma chave simétrica AES que você gera e um algoritmo de criptografia simétrica AES.
2. Criptografe a chave simétrica AES que você usou com a chave pública que você baixou e o algoritmo de empacotamento `RSAES_OAEP_SHA_256`.

O algoritmo de empacotamento `RSAES_OAEP_SHA_256` requer OpenSSL versão 3.x ou superior.

1. Gerar uma chave de criptografia simétrica AES de 256 bits

Esse comando gera uma chave de criptografia simétrica AES que consiste em 256 bits aleatórios e a salva no arquivo `aes-key.bin`

```
# Generate a 32-byte AES symmetric encryption key
$ openssl rand -out aes-key.bin 32
```

2. Criptografar o material de chave com a chave de criptografia simétrica AES

Esse comando criptografa o material da chave com a chave de criptografia simétrica AES e salva o material da chave criptografada no arquivo `key-material-wrapped.bin`.

Neste comando de exemplo:

- *PlaintextKeyMaterial.bin* é o arquivo que contém o material de chave que você está importando, como `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der` ou `ECC_NIST_P521_PrivateKey.der`.
- *aes-key.bin* é o arquivo que contém a chave de criptografia simétrica AES de 256 bits que você gerou no comando anterior.

```
# 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. Criptografar a chave de criptografia simétrica AES com a chave pública

Esse comando criptografa a chave de criptografia simétrica AES com a chave pública que você baixou e o algoritmo de empacotamento `RSAES_OAEP_SHA_256`, a codifica em DER e a salva no arquivo `aes-key-wrapped.bin`.

Neste comando de exemplo:

- *WrappingPublicKey.bin* é o arquivo que contém a chave de empacotamento pública baixada. Se você fez download da chave pública do console, esse arquivo

será chamado `wrappingKey_KMS key_key_ID_timestamp` (por exemplo, `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`

- `aes-key.bin` é o arquivo que contém a chave de criptografia simétrica AES de 256 bits que você gerou no primeiro comando dessa sequência de exemplos.

```
# 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. Gerar o arquivo a ser importado

Concatene o arquivo com o material da chave criptografada e o arquivo com a chave AES criptografada. Salve-os no arquivo `EncryptedKeyMaterial.bin`, que é o arquivo que você importará no [Etapa 4: Importar o material de chave](#).

Neste comando de exemplo:

- `key-material-wrapped.bin` é o arquivo que contém o material de chaves criptografado.
- `aes-key-wrapped.bin` é o arquivo que contém a chave de criptografia AES criptografada.

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

Vá para [Etapa 4: Importar o material de chave](#).

Etapa 4: Importar o material de chave

Depois de [criptografar o material de chave](#), você pode importá-lo para uso com uma AWS KMS key. Para importar o material de chaves, você faz upload do material de chaves criptografado de [Etapa 3: Criptografar o material de chave](#) e o token de importação que você baixou em [Etapa 2: Fazer download da chave pública de empacotamento e do token de importação](#). É necessário importar o material de chaves para a mesma chave do KMS que especificou quando você [baixou a chave pública e o token de importação](#). Quando o material da chaves é importado com êxito, o [estado da chave](#) da chave do KMS muda para Enabled, e você pode usar a chave do KMS em operações de criptografia.

Ao importar material de chaves, é possível [definir uma hora de expiração opcional](#) para ele. Quando o material de chave perde a validade, o AWS KMS exclui o material de chave e a chave KMS se torna inutilizável. Depois de importar o material de chaves, você não pode definir, alterar ou cancelar a data de expiração da importação atual. Para alterar esses valores, você deve [reimportar](#) o mesmo material de chave.

Em todas as chaves do KMS com a origem EXTERNAL, o primeiro material de chave importado se torna atual e permanentemente associado a ela. Chaves de criptografia simétricas com suporte de EXTERNAL origem para rotação sob demanda. Você pode associar vários materiais de chave a chaves importadas compatíveis com rodízio sob demanda. O processo de importação de novo material de chave é diferente para chaves de região única e de várias regiões, conforme descrito na seção [Importar novo](#) material de chave. Você deve definir o `importType` parâmetro como `NEW_KEY_MATERIAL` com a [ImportKeyMaterial](#) ação para associar o novo material de chave a uma chave KMS. O valor padrão do parâmetro opcional `ImportType` é `EXISTING_KEY_MATERIAL`. Quando você omite o parâmetro `ImportType` ou o especifica como `EXISTING_KEY_MATERIAL`, é necessário importar um material de chave previamente associado à chave do KMS.

Para chaves assimétricas ou HMAC KMS com EXTERNAL origem, somente um material de chave pode ser associado à chave. AWS KMS rejeitará solicitações de [ImportKeyMaterial](#) API com o `ImportType` parâmetro.

Quando todos os materiais de chave permanentemente associados a uma chave do KMS são importados, a chave do KMS fica disponível para uso em operações criptográficas. Se algum desses materiais de chave for excluído ou deixado expirar, o estado da chave do KMS mudará para `PendingImport` e a chave ficará inutilizável para operações criptográficas.

Para importar material chave, você pode usar o [AWS KMS console](#) ou a [ImportKeyMaterial](#) API. Você pode usar a API diretamente fazendo solicitações HTTP ou usando um [AWS SDKs](#), [AWS Command Line Interface](#) ou [Ferramentas da AWS para PowerShell](#).

Quando você importa o material da chave, uma [ImportKeyMaterial](#) entrada é adicionada ao seu AWS CloudTrail registro para registrar a `ImportKeyMaterial` operação. A CloudTrail entrada é a mesma se você usa o AWS KMS console ou a AWS KMS API.

Definir um prazo de validade (opcional)

Ao importar o material de chave para a chave do KMS, você pode definir uma data e hora de validade opcionais para o material de chave de até 365 dias a partir da data de importação. Quando o material de chave importado expira, ele é AWS KMS excluído. Essa ação altera [estado de chave](#) da chave do KMS para `PendingImport`, impedindo que ela seja usada em operações de criptografia. Para usar a chave do KMS, você deve [reimportar uma cópia do material de chave original](#).

Garantir que o material de chave importado expire com frequência pode ajudar a atender aos requisitos regulatórios, mas aumenta o risco dos dados criptografados sob a chave do KMS. Até que você reimporte uma cópia do material da chave original, uma chave do KMS com material de chave expirado ficará inutilizável, e todos os dados criptografados sob a chave do KMS ficarão inacessíveis. Se você não reimportar o material da chave por qualquer motivo, inclusive por perda da cópia do material da chave original, a chave do KMS ficará permanentemente inutilizável, e os dados criptografados sob a chave do KMS ficarão irre recuperáveis.

Para reduzir esse risco, certifique-se de que sua cópia do material de chave importado esteja acessível e crie um sistema para excluir e reimportar o material de chaves antes que ele expire e interrompa sua carga de trabalho. AWS Recomendamos [definir um alarme](#) para a expiração do material de chave importado, o que lhe dará tempo suficiente para reimportar o material da chave antes que ele expire. Você também pode usar seus CloudTrail registros para auditar operações que [importam \(e reimportam\) material de chave e excluem material de chave importado, e a AWS KMS operação para excluir material de chave expirado](#).

AWS KMS não pode restaurar, recuperar ou reproduzir o material de chave excluído. Em vez de definir um prazo de validade, você pode [excluir](#) e [reimportar](#) o material de chave importado de maneira programática periodicamente, mas os requisitos para reter uma cópia do material da chave original são os mesmos.

Ao importar o material de chave, você determina se o material de chave importado expirará e quando isso ocorrerá. Mas é possível ativar e desativar a expiração ou definir um novo prazo de validade

reimportando o material de chave. Use o `ExpirationModel` parâmetro de [ImportKeyMaterial](#) para ativar (`KEY_MATERIAL_EXPIRES`) e desativar a expiração (`KEY_MATERIAL_DOES_NOT_EXPIRE`) e o `ValidTo` parâmetro para definir o tempo de expiração. O tempo máximo é de 365 dias a partir da importação de dados. Não há mínimo, mas o tempo deve estar no futuro.

Definir descrição de material de chave

As chaves de criptografia simétricas com `EXTERNAL` origem podem ter vários materiais de chave associados a elas. Você pode especificar uma descrição opcional de material de chave ao importar o material de chave para essas chaves. A descrição pode ser usada para rastrear onde o material de chave correspondente é mantido a longo prazo no AWS KMS exterior.

Para chaves multirregionais, você pode definir ou alterar a descrição do material da chave somente na chave de região primária. AWS KMS propaga automaticamente a descrição do material chave para réplicas de chaves de região.

Importar novo material de chave

Primeiro você precisará importar um novo material de chave, não associado anteriormente à chave para realizar um rodízio sob demanda de uma chave do KMS de criptografia simétrica com material de chave importado.

- Chaves de região única
 - Use a [ImportKeyMaterial](#) operação com o `ImportType` parâmetro definido `NEW_KEY_MATERIAL` para realizar essa tarefa. Esse material de chave não é associado permanentemente à chave até que você execute a [RotateKeyOnDemand](#) operação ou gire a chave no Console de gerenciamento da AWS. Até lá, esse material de chave fica no estado `PENDING_ROTATION`. Uma chave do KMS nunca pode ter mais que um material de chave no estado `PENDING_ROTATION`. Um material de chave em `PENDING_ROTATION` estado pode ser excluído sem afetar a usabilidade da chave em operações criptográficas.
- Chaves de várias regiões
 - Para importar material chave em uma chave multirregional, você deve primeiro importar o novo material chave para a chave de região primária. Você não pode importar diretamente novos materiais de chave para réplicas de chaves de região. Depois de importar o novo material de chave para a chave de região primária, você pode importar os mesmos materiais de chave para as réplicas de chaves de região.
 - Use a [ImportKeyMaterial](#) operação com o `ImportType` parâmetro definido como `NEW_KEY_MATERIAL` para a chave de região primária para realizar essa tarefa. Para a chave

de região da réplica, use o **EXISTING_KEY_MATERIAL** parâmetro `ImportType` para a `ImportKeyMaterial` operação.

- O material chave para chaves multirregionais de criptografia simétrica deve ser importado para todas as chaves de região de réplica e chaves de região primárias antes que o estado do material da chave mude para estado. `PENDING_ROTATION` Até lá, o estado do novo material chave é `PENDING_MULTI_REGION_IMPORT_AND_ROTATION`. Uma chave KMS pode ter no máximo um material de chave `PENDING_ROTATION` ou `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` estado a qualquer momento (veja a `KeyMaterialState` descrição em [RotationsListEntry](#)). Um material de chave em `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` ou `PENDING_ROTATION` estado não está associado permanentemente à chave e pode ser excluído sem afetar a usabilidade da chave em operações criptográficas.

Reimportar material de chave

Se você gerencia uma chave do KMS com material de chaves importado, talvez seja necessário importar novamente o material de chave. Você também reimportar o material de chave para substituir material de chave prestes a expirar ou excluído ou para alterar o modelo de expiração ou a data de validade do material de chave.

Você pode reimportar material da chave a qualquer momento, em qualquer programação que atenda aos seus requisitos de segurança. Não é necessário esperar até que o material da chave esteja expirando ou prestes a expirar.

Os procedimentos para reimportar material de chave são iguais ao procedimento que você usou para importar o material de chave a primeira vez, com as exceções a seguir.

- Utilize uma chave do KMS existente em vez de criar uma nova. Você pode ignorar a [Etapa 1](#) do procedimento de importação.
- Ao reimportar o material de chave, você pode alterar o modelo de expiração e data de expiração. Para chaves de criptografia simétricas, você também pode alterar a descrição do material da chave.

Para chaves multirregionais, você pode definir ou alterar a descrição do material da chave somente na chave de região primária. AWS KMS propaga automaticamente a descrição do material chave para réplicas de chaves de região.

Cada vez que você importa o material de chave para uma chave do KMS, é necessário [baixar e usar uma nova chave de empacotamento e token de importação](#) da chave do KMS. O procedimento de empacotamento não afeta o conteúdo do material de chave, portanto você pode usar diferentes chaves públicas de empacotamento e diferentes algoritmos de empacotamento para importar o mesmo material de chave.

Importar o material de chave (console)

Você pode usar o Console de gerenciamento da AWS para importar material chave.

1. Se você estiver na página Upload your wrapped key material (Fazer upload de chave empacotada), vá para [Step 10](#).
2. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
3. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
4. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
5. Escolha o alias ou o ID da chave do KMS para a qual você baixou a chave pública e o token de importação.
6. Expanda a seção Cryptographic configuration (Configuração criptográfica) e visualize seus valores. As guias estão na página de detalhes de uma chave do KMS, abaixo da seção General configuration (Configuração geral).

É possível importar material de chaves somente para uma chave do KMS com uma Origin (Origem) de External (Import key material) (Externa [Importar material de chave]). Para obter informações sobre como criar chaves do KMS com material de chave importado, consulte [Importação de material chave para AWS KMS chaves](#).

7. Escolha a guia apropriada com base no seu tipo de chave.
 - Para chaves assimétricas e HMAC, escolha a guia Material da chave.
 - Para chaves de criptografia simétricas, escolha a guia Material da chave e rotações.
8. Escolha a ação de importação.
 - Para chaves assimétricas e HMAC, escolha Importar material de chave.
 - Para chaves de criptografia simétricas, escolha uma das seguintes opções:
 - Importar material de chave inicial (se nenhum material de chave tiver sido importado ainda)
 - Importar novo material chave (para adicionar novo material para rotação)

- Reimportar material chave (disponível no menu Ações na tabela de materiais principais)

 Note

Para chaves multirregionais, você deve primeiro importar o novo material de chave para a chave de região primária. Em seguida, importe o mesmo material de chave em cada chave de região de réplica.

Para chaves multirregionais primárias, a tabela de materiais principais inclui uma coluna de estado de importação da réplica que exibe o status da importação em todas as regiões da réplica (por exemplo, “0 de 3 importados”). Escolha o valor do estado de importação da réplica para abrir um modal que mostre o status de importação de cada região da réplica. O modal fornece links de importação de material de chave para regiões de réplica onde o novo material de chave não foi importado.

9. Se você baixou o material da chave, o token de importação e criptografou o material da chave, escolha Next (Avançar).

 Note

Para chaves multirregionais, você deve primeiro importar o novo material de chave para a chave de região primária. Em seguida, você pode importar o mesmo material de chave para as chaves de região da réplica.

10. Na seção Encrypted key material and import token (Material de chave criptografada e token de importação), faça o seguinte:
 - a. Em Wrapped key material (material de chave empacotada), escolha Choose file (Escolher arquivo). Faça upload do arquivo que contém o material de chaves encapsulado (criptografado).
 - b. Em Import token (Importar token), escolha Choose file (Escolher arquivo). Faça upload do arquivo que contém o token de importação [obtido por download](#).
11. Na seção Expiration option (Opção de expiração), determine se o material de chave expira. Para definir uma data e hora de expiração, escolha Key material expires (O material de chaves expira), e use o calendário para selecionar uma data e hora. Você pode especificar uma data de até 365 dias da data e hora atuais.

12. Para chaves de criptografia simétrica, você tem a opção de especificar uma descrição para o material da chave que está sendo importado.
13. Escolha Importar material de chave.

Importar material chave (AWS KMS API)

Para importar material chave, use a [ImportKeyMaterial](#) operação. O exemplo a seguir usa a [AWS CLI](#), mas você pode usar qualquer linguagem de programação compatível.

Para usar este exemplo:

1. Substitua *1234abcd-12ab-34cd-56ef-1234567890ab* pelo ID da chave do KMS que você especificou quando baixou a chave pública e o token de importação. Para identificar a chave do KMS, use seu [ID de chave](#) ou [ARN de chave](#). Você não pode usar um [nome de alias](#) ou [ARN de alias](#) para esta operação.
2. Substitua *EncryptedKeyMaterial.bin* pelo nome do arquivo que contém o material de chaves criptografado.
3. Substitua *ImportToken.bin* pelo nome do arquivo que contém o token de importação.
4. Se você quiser que o material da chave importada expire, defina o valor do parâmetro `expiration-model` para seu valor padrão, `KEY_MATERIAL_EXPIRES`, ou omita o parâmetro `expiration-model`. Em seguida, substitua o valor do parâmetro `valid-to` com a data e a hora em que você deseja que o material da chave expire. A data e a hora podem ser de até 365 dias a partir do momento da solicitação.

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

Se você não quiser que o material da chave importada expire, defina o valor do parâmetro `expiration-model` como `KEY_MATERIAL_DOES_NOT_EXPIRE` e omita o parâmetro `valid-to` do comando.

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
  --import-token fileb://ImportToken.bin \
```

```
--expiration-model KEY_MATERIAL_DOES_NOT_EXPIRE
```

5. Se você quiser importar novo material de chave, não associado anteriormente à chave do KMS, defina o parâmetro `ImportType` como `NEW_KEY_MATERIAL`. Essa opção só pode ser usada com chaves de criptografia simétricas. Para essas chaves, você também pode usar o parâmetro opcional `KeyMaterialDescription` para definir uma descrição para o material de chave importado no seguinte exemplo de linha de comando:

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

6. Para chaves multirregionais, você pode definir ou alterar a descrição do material da chave somente na chave de região primária. AWS KMS propaga automaticamente a descrição do material chave para réplicas de chaves de região.

Tip

Se o comando não for bem-sucedido, você poderá ver uma `KMSInvalidStateException` ou `NotFoundException`. Você poderá repetir a solicitação.

Criar uma chave KMS em um armazenamento de AWS CloudHSM chaves

Depois de criar um armazenamento de AWS CloudHSM chaves, você pode criar AWS KMS keys em seu armazenamento de chaves. Elas devem ser [chaves KMS de criptografia simétrica](#) com material de chave gerado AWS KMS. Não é possível criar [chaves do KMS assimétricas](#), [chaves do KMS de Hash-based message authentication code \(HMAC – Código de autenticação de mensagem por hash\)](#) ou chaves do KMS com [material de chave importado](#) em um armazenamento personalizado de chave. Além disso, não é possível usar chaves do KMS de criptografia simétrica em um armazenamento personalizado de chaves para gerar pares de chaves assimétricos de dados. O KMS não pode se comunicar IPv6 com as AWS CloudHSM principais lojas.

Para criar uma chave KMS em um armazenamento de AWS CloudHSM chaves, o armazenamento de AWS CloudHSM chaves deve estar [conectado ao AWS CloudHSM cluster associado](#) e o cluster deve conter pelo menos dois ativos HSMs em diferentes zonas de disponibilidade. Para encontrar o estado da conexão e o número de HSMs, consulte a [página de armazenamentos de AWS CloudHSM chaves](#) no Console de gerenciamento da AWS. Ao usar as operações da API, use a [DescribeCustomKeyStores](#) operação para verificar se o armazenamento de AWS CloudHSM chaves está conectado. Para verificar o número de ativos HSMs no cluster e suas zonas de disponibilidade, use a AWS CloudHSM [DescribeClusters](#) operação.

Quando você cria uma chave KMS em seu armazenamento de AWS CloudHSM chaves, AWS KMS cria a chave KMS em. AWS KMS Porém, ele cria o material chave para a chave KMS no AWS CloudHSM cluster associado. Especificamente AWS KMS , entra no cluster como a [kmsuserUC que você criou](#). Em seguida, ele cria uma chave simétrica persistente e não extraível de AES (Advanced Encryption Standard) de 256 bits no cluster. O AWS KMS define o valor do [atributo de rótulo de chave](#), que é visível apenas no cluster, como o nome do recurso da Amazon (ARN) da chave do KMS.

Quando o comando é bem-sucedido, o [estado de chave](#) da nova chave do KMS é Enabled e sua origem é AWS_CLOUDHSM. Não é possível alterar a origem de uma chave do KMS após a sua criação. Ao visualizar uma chave KMS em um armazenamento de AWS CloudHSM chaves no AWS KMS console ou usando a [DescribeKey](#) operação, você pode ver propriedades típicas, como ID da chave, estado da chave e data de criação. No entanto, você também pode ver o ID do armazenamento de chaves personalizado e (opcionalmente) o ID do cluster do AWS CloudHSM .

Se sua tentativa de criar uma chave KMS no armazenamento de AWS CloudHSM chaves falhar, use a mensagem de erro para ajudá-lo a determinar a causa. Isso pode indicar que o armazenamento de AWS CloudHSM chaves não está conectado (CustomKeyStoreInvalidStateException) ou HSMs que o AWS CloudHSM cluster associado não tem os dois ativos necessários para essa operação (CloudHsmClusterInvalidConfigurationException). Para obter ajuda, consulte [Solucionar problemas de um armazenamento de chaves personalizado](#).

Para obter um exemplo do AWS CloudTrail registro da operação que cria uma chave KMS em um armazenamento de AWS CloudHSM chaves, consulte [CreateKey](#).

Criar uma nova chave do KMS em seu repositório de chaves do CloudHSM

Você pode criar uma chave KMS de criptografia simétrica no seu armazenamento de AWS CloudHSM chaves no AWS KMS console ou usando a [CreateKey](#) operação.

Usando o AWS KMS console

Use o procedimento a seguir para criar uma chave KMS de criptografia simétrica em um armazenamento de AWS CloudHSM chaves.

Note

Não inclua informações confidenciais ou sigilosas no alias, na descrição ou nas tags. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Escolha Criar chave.
5. Selecione Symmetric (Simétrica).
6. Em Key usage (Uso da chave), a opção Encrypt and decrypt (Criptografar e descriptografar) é selecionada para você. Não altere essa opção.
7. Escolha Advanced options (Opções avançadas).
8. Em Key material origin (Origem do material de chave), escolha AWS CloudHSM key store (Armazenamento de chaves do).

Você não pode criar uma chave multirregional em um armazenamento de AWS CloudHSM chaves.

9. Escolha Próximo.
10. Selecione um repositório de AWS CloudHSM chaves para sua nova chave KMS. Para criar um novo armazenamento de AWS CloudHSM chaves, escolha Criar armazenamento de chaves personalizado.

O armazenamento de AWS CloudHSM chaves selecionado deve ter o status Conectado. Seu AWS CloudHSM cluster associado deve estar ativo e conter pelo menos dois ativos HSMs em diferentes zonas de disponibilidade.

Para obter ajuda com a conexão de um armazenamento de AWS CloudHSM chaves, consulte [Desconectar um armazenamento de AWS CloudHSM chaves](#). Para obter ajuda com a adição HSMs, consulte [Adicionar um HSM](#) no Guia do AWS CloudHSM usuário.

11. Escolha Próximo.
12. Digite um alias e uma descrição opcional para a chave do KMS.
13. (Opcional). Na página Add Tags (Adicionar etiquetas), adicione etiquetas que identificam ou categorizam a chave do KMS.

Quando você adiciona tags aos seus AWS recursos, AWS gera um relatório de alocação de custos com uso e custos agregados por tags. Etiquetas também podem ser utilizadas para controlar o acesso a uma chave do KMS. Para informações sobre marcação de chaves do KMS, consulte [Etiquetas em AWS KMS](#) e [ABAC para AWS KMS](#).

14. Escolha Próximo.
15. Na seção Key Administrators (Administradores de chaves), selecione os usuários e as funções do IAM que podem gerenciar a chave do KMS. Para obter mais informações, consulte [Permite que administradores de chaves administrem a chave do KMS](#).

Observações

As políticas do IAM podem conceder permissão para usar a chave do KMS a outros usuários e funções do IAM.

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

O AWS KMS console adiciona administradores de chaves à política de chaves sob o identificador "Allow access for Key Administrators" de instrução. A modificação desse identificador de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

16. (Opcional) Para evitar que esses administradores de chaves excluam essa chave do KMS, desmarque a caixa na parte inferior da página para Permitir que os administradores de chaves excluam essa chave.
17. Escolha Próximo.
18. Na seção Esta conta, selecione os usuários e funções do IAM Conta da AWS que podem usar a chave KMS em operações [criptográficas](#). Para obter mais informações, consulte [Permite que os usuários de chaves usem a chave do KMS](#).

 Observações

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

O AWS KMS console adiciona os principais usuários à política de chaves sob os identificadores de declaração "Allow use of the key" e "Allow attachment of persistent resources". A modificação desses identificadores de instrução pode afetar a forma como o console exibe as atualizações feitas na instrução.

19. (Opcional) Você pode permitir que outras Contas da AWS usem essa chave KMS para operações criptográficas. Para fazer isso, na Contas da AWS seção Outros na parte inferior da página, escolha Adicionar outra Conta da AWS e insira o ID da Conta da AWS de uma conta externa. Para adicionar várias contas externas, repita essa etapa.

 Note

Os administradores de outras Contas da AWS devem permitir o acesso à chave KMS criando políticas do IAM para seus usuários. Para obter mais informações, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

20. Escolha Próximo.
21. Consulte as instruções da política de chave para a chave. Para fazer alterações na política de chave, selecione Editar.
22. Escolha Próximo.
23. Revise as configurações que você escolheu. Ainda é possível voltar e alterar todas as configurações.
24. Quando terminar, escolha Finish (Terminar) para criar a chave.

Quando o procedimento for bem-sucedido, a tela mostrará a nova chave KMS no armazenamento de AWS CloudHSM chaves que você escolheu. Quando você escolhe o nome ou alias da nova chave KMS, a guia Configuração criptográfica em sua página de detalhes exibe a origem da chave KMS (AWS CloudHSM), o nome, o ID e o tipo do armazenamento de chaves personalizadas e o ID do

cluster. AWS CloudHSM Se houver falha no procedimento, uma mensagem descrevendo a falha será exibida.

 Tip

Para facilitar a identificação de chaves do KMS em um armazenamento de chaves personalizado, na página Customer managed keys (Chaves gerenciadas pelo cliente), adicione a coluna Custom key store ID (ID de armazenamento de chaves personalizado) à exibição. Clique no ícone de engrenagem no canto superior direito e selecione Custom key store ID (ID de armazenamento de chaves personalizado). Para obter detalhes, consulte [Personalizar a exibição do console](#).

Usando a AWS KMS API

Para criar uma nova AWS KMS key (chave KMS) em seu armazenamento de AWS CloudHSM chaves, use a [CreateKey](#) operação. Use o parâmetro CustomKeyId para identificar o armazenamento de chaves personalizado e especificar um valor de Origin da AWS_CLOUDHSM.

Você também pode usar o parâmetro Policy para especificar uma política de chaves. Você pode alterar a política de chaves ([PutKeyPolicy](#)) e adicionar elementos opcionais, como uma [descrição](#) e [tags](#), a qualquer momento.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

O exemplo a seguir começa com uma chamada para a [DescribeCustomKeyStores](#) operação para verificar se o armazenamento de AWS CloudHSM chaves está conectado ao AWS CloudHSM cluster associado. Por padrão, essa operação retorna todos os armazenamentos de chaves personalizados em sua conta e região. Para descrever somente um determinado armazenamento de AWS CloudHSM chaves, use seu CustomKeyName parâmetro CustomKeyId or (mas não ambos).

Antes de executar um comando como esse, substitua o ID de exemplo do armazenamento de chaves personalizado por um ID válido.

Note

Não inclua informações confidenciais ou sigilosas nos campos `Description` ou `Tags`. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

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

O próximo comando de exemplo usa a [DescribeClusters](#) operação para verificar se o AWS CloudHSM cluster associado ao `ExampleKeyStore` (`cluster-1a23b4cdefg`) tem pelo menos dois ativos. HSMs Se o cluster tiver menos de dois HSMs, a `CreateKey` operação falhará.

```
$ 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\n-----END CERTIFICATE-----\n"
      },
      "Hsms": [
```



```

    {
      "AvailabilityZone": "us-west-2a",
      "EniIp": "10.0.1.11",
      "ClusterId": "cluster-1a23b4cdefg",
      "EniId": "eni-ea8647e1",
      "StateMessage": "HSM created.",
      "SubnetId": "subnet-a6b10bd1",
      "HsmId": "hsm-abcdefghijkl",
      "State": "ACTIVE"
    },
    {
      "AvailabilityZone": "us-west-2b",
      "EniIp": "10.0.0.2",
      "ClusterId": "cluster-1a23b4cdefg",
      "EniId": "eni-ea8647e1",
      "StateMessage": "HSM created.",
      "SubnetId": "subnet-b6b10bd2",
      "HsmId": "hsm-zyxwvutsrq",
      "State": "ACTIVE"
    },
  ],
  "State": "ACTIVE"
}
]
}

```

Este exemplo de comando usa a [CreateKey](#) operação para criar uma chave KMS em um armazenamento de AWS CloudHSM chaves. Para criar uma chave KMS em um armazenamento de AWS CloudHSM chaves, você deve fornecer o ID de armazenamento de chaves personalizado do armazenamento de AWS CloudHSM chaves e especificar um `Origin` valor de `AWS_CLOUDHSM`.

A resposta inclui o armazenamento IDs de chaves personalizadas e o AWS CloudHSM cluster.

Antes de executar um comando como esse, substitua o ID de exemplo do armazenamento de chaves personalizado por um ID válido.

```

$ 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"
  ]
}
```

Criar uma chave do KMS em repositórios de chaves externos

Depois de [criar](#) e [conectar](#) seu armazenamento de chaves externo, você pode criar AWS KMS keys em seu armazenamento de chaves. Devem ser [chaves do KMS de criptografia simétrica](#) com um valor de origem de armazenamento de chaves externas (EXTERNAL_KEY_STORE). Não é possível criar [chaves do KMS assimétricas](#), [chaves do KMS de Hash-based message authentication code \(HMAC – Código de autenticação de mensagem por hash\)](#) ou chaves do KMS com [material de chave importado](#) em um armazenamento personalizado de chave. Além disso, não é possível usar chaves do KMS de criptografia simétrica em um armazenamento personalizado de chaves para gerar pares de chaves assimétricos de dados.

Uma chave do KMS em um armazenamento de chaves externas pode ter menor latência, durabilidade e disponibilidade do que uma chave do KMS padrão, pois depende de componentes localizados fora da AWS. Antes de criar ou usar uma chave do KMS em um armazenamento de chaves externas, verifique se você precisa de uma chave com propriedades de armazenamento de chaves externas.

 Note

Alguns gerenciadores de chaves externas fornecem um método mais simples de criar chaves do KMS em um armazenamento de chaves externas. Para obter detalhes, consulte a documentação do gerenciador de chaves externas.

Para criar uma chave do KMS no armazenamento de chaves externas, especifique:

- O ID do armazenamento de chaves externas.
- A [origem do material de chave](#) do armazenamento de chaves externas (EXTERNAL_KEY_STORE).
- O ID de uma [chave externa](#) existente no [gerenciador de chaves externas](#) associado a seu armazenamento de chaves externas. Essa chave externa serve como material de chave para a chave do KMS. Você pode alterar o ID da chave externa após criar a chave do KMS.

AWS KMS fornece o ID da chave externa ao proxy externo do armazenamento de chaves nas solicitações de operações de criptografia e descriptografia. AWS KMS não pode acessar diretamente seu gerenciador de chaves externo ou qualquer uma de suas chaves criptográficas.

Além da chave externa, uma chave KMS em um armazenamento de chaves externo também tem material de AWS KMS chave. Todos os dados criptografados sob a chave KMS são primeiro criptografados AWS KMS usando o material de chave da AWS KMS chave e, em seguida, pelo seu gerenciador de chaves externo usando sua chave externa. Esse processo de [criptografia dupla](#) garante que o texto cifrado protegido pela chave do KMS em um armazenamento de chaves externas seja pelo menos tão forte quanto o texto cifrado protegido somente pelo AWS KMS. Para obter detalhes, consulte [Como funcionam os armazenamentos de chaves externas](#).

Quando a operação CreateKey é bem-sucedida, o [estado da chave](#) da nova chave do KMS é Enabled. Quando você [visualiza uma chave do KMS em um repositório de chaves externo](#), pode ver as propriedades comuns, como ID da chave, [especificação da chave](#), [uso da chave](#), [estado da chave](#) e a data da criação. Mas você também pode ver o ID e o [estado da conexão](#) do armazenamento de chaves externas e o ID da chave externa.

Se ocorrer uma falha na tentativa de criar uma chave do KMS no seu armazenamento de chaves externas, use a mensagem de erro para identificar a causa. Isso pode indicar que o armazenamento de chaves externas não está conectado (CustomKeyStoreInvalidStateException), que o proxy de armazenamento de chaves externas não consegue encontrar uma chave externa

com o ID da chave externa especificada (`XksKeyNotFoundException`) ou que a chave externa já está associada a uma chave do KMS no mesmo armazenamento de chaves externas `XksKeyAlreadyInUseException`.

Para obter um exemplo do AWS CloudTrail registro da operação que cria uma chave KMS em um armazenamento de chaves externo, consulte [CreateKey](#).

Tópicos

- [Requisitos para uma chave do KMS em um armazenamento de chaves externas](#)
- [Criar uma nova chave do KMS em seu repositório de chaves externo](#)

Requisitos para uma chave do KMS em um armazenamento de chaves externas

Para criar uma chave do KMS em um armazenamento de chaves externas, as seguintes propriedades são obrigatórias no armazenamento de chaves externas, na chave do KMS e na chave externa que serve como material de chave de criptografia externa para a chave do KMS.

Requisitos de armazenamento de chaves externas

- Deve estar conectado ao proxy de armazenamento de chaves externas.

Para visualizar o [estado da conexão](#) do armazenamento de chaves externas, consulte [Visualizar repositórios de chaves externos](#). Para conectar o armazenamento de chaves externas, consulte [Conectar e desconectar repositórios de chaves externos](#).

Requisitos de chaves do KMS

Você não pode alterar essas propriedades após criar a chave do KMS.

- Especificação da chave: `SYMMETRIC_DEFAULT`
- Uso da chave: `ENCRYPT_DECRYPT`
- Origem do material de chave: `EXTERNAL_KEY_STORE`
- Várias regiões: `FALSE`

Requisitos de chaves externas

- Chave de criptografia AES de 256 bits (256 bits aleatórios). O KeySpec da chave externa deve ser AES_256.
- Habilitadas e disponíveis para uso. O Status da chave externa deve ser ENABLED.
- Configurada para criptografia e descriptografia. O KeyUsage da chave externa deve incluir ENCRYPT e DECRYPT.
- Usado somente com essa chave do KMS. Cada KMS key em um armazenamento de chaves externas deve estar associada a uma chave externa diferente.

AWS KMS também recomenda que a chave externa seja usada exclusivamente para o armazenamento externo de chaves. Essa restrição facilita identificar e resolver problemas da chave.

- Acessível pelo [proxy de armazenamento de chaves externas](#) para o armazenamento de chaves externas.

Se o proxy de armazenamento de chaves externas não conseguir encontrar a chave usando o ID de chave externa especificado, a operação CreateKey falhará.

- Pode lidar com o tráfego previsto que seu uso Serviços da AWS gera. AWS KMS recomenda que as chaves externas estejam preparadas para lidar com até 1800 solicitações por segundo.

Criar uma nova chave do KMS em seu repositório de chaves externo

Você pode criar uma nova chave KMS em seu armazenamento externo de chaves no AWS KMS console ou usando a [CreateKey](#) operação.

Usando o AWS KMS console

Há duas maneiras de criar uma chave do KMS em um armazenamento de chaves externas.

- Método 1 (recomendado): escolha um armazenamento de chaves externas e crie uma chave do KMS nesse armazenamento de chaves externo.
- Método 2: crie uma chave do KMS e indique que ela está em um armazenamento de chaves externo.

Se você usar o Método 1, em que escolhe seu armazenamento de chaves externo antes de criar sua chave, AWS KMS escolhe todas as propriedades de chave KMS necessárias para você e preenche a ID do seu armazenamento de chaves externo. Esse método evita erros que você possa cometer ao criar sua chave do KMS.

 Note

Não inclua informações confidenciais ou sigilosas no alias, na descrição ou nas tags. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

Método 1 (recomendado): comece com o armazenamento de chaves externo

Para usar esse método, escolha seu armazenamento de chaves externas e crie uma chave do KMS. O AWS KMS console escolhe todas as propriedades necessárias para você e preenche o ID do seu armazenamento de chaves externo. Esse método evita muitos erros que você possa cometer ao criar sua chave do KMS.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Armazenamentos de chaves personalizados), External key stores (Armazenamentos de chaves externas).
4. Escolha o nome do armazenamento de chaves externas.
5. No canto superior direito, escolha Create a KMS key in this key store (Criar uma chave do KMS neste armazenamento de chaves).

Se o armazenamento de chaves externas não estiver conectado, você verá um aviso para conectá-lo. Se a tentativa de conexão falhar, será necessário resolver o problema e conectar o armazenamento de chaves externas antes de criar uma nova chave do KMS nele.

Se o armazenamento de chaves externas estiver conectado, você será redirecionado para a página Customer managed keys (Chaves gerenciadas pelo cliente) para criar uma chave. Os valores de Key configuration (Configuração de chave) obrigatórios já foram escolhidos para você. Além disso, o ID de armazenamento de chaves personalizado do armazenamento de chaves externas está preenchido, mas é possível alterá-lo.

6. Insira o ID da [chave externa](#) no [gerenciador de chaves externas](#). Essa chave externa deve [atender aos requisitos](#) para uso com uma chave do KMS. Você não pode alterar o valor depois que a chave é criada.

Se a chave externa tiver várias IDs, insira a ID da chave que o proxy do armazenamento de chaves externo usa para identificar a chave externa.

7. Confirme que você pretende criar uma chave do KMS no armazenamento de chaves externas especificado.
8. Escolha Próximo.

O restante desse procedimento é o mesmo que a [criação de uma chave do KMS padrão](#).

9. Digite um alias (obrigatório) e uma descrição (opcional) para a chave do KMS.
10. (Opcional). Na página Add Tags (Adicionar etiquetas), adicione etiquetas que identificam ou categorizam a chave do KMS.

Quando você adiciona tags aos seus AWS recursos, AWS gera um relatório de alocação de custos com uso e custos agregados por tags. Etiquetas também podem ser utilizadas para controlar o acesso a uma chave do KMS. Para informações sobre marcação de chaves do KMS, consulte [Etiquetas em AWS KMS](#) e [ABAC para AWS KMS](#).

11. Escolha Próximo.
12. Na seção Key Administrators (Administradores de chaves), selecione os usuários e as funções do IAM que podem gerenciar a chave do KMS. Para obter mais informações, consulte [Permite que administradores de chaves administrem a chave do KMS](#).

 Note

As políticas do IAM podem conceder permissão para usar a chave do KMS a outros usuários e funções do IAM.

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

13. (Opcional) Para evitar que esses administradores de chaves excluam essa chave do KMS, desmarque a caixa de seleção Allow key administrators to delete this key (Permitir que os administradores de chaves excluam essa chave).

Excluir uma chave do KMS é uma operação destrutiva e irreversível que pode tornar o texto cifrado irrecuperável. Você não pode recriar uma chave do KMS simétrica em um armazenamento de chaves externas, mesmo que você tenha o material de chave externa. No entanto, a exclusão de uma chave do KMS não afeta a chave externa associada. Para obter informações sobre como excluir uma chave do KMS de um repositório de chaves externo, consulte [Considerações especiais para a exclusão de chaves](#).

14. Escolha Próximo.
15. Na seção Esta conta, selecione os usuários e funções do IAM Conta da AWS que podem usar a chave KMS em operações [criptográficas](#). Para obter mais informações, consulte [Permite que os usuários de chaves usem a chave do KMS](#).

 Note

As políticas do IAM podem conceder permissão para usar a chave do KMS a outros usuários e funções do IAM.

As práticas recomendadas do IAM não encorajam o uso de usuários do IAM com credenciais de longo prazo. Sempre que possível, use os perfis do IAM, por fornecerem credenciais temporárias. Para obter detalhes, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

16. (Opcional) Você pode permitir que outras Contas da AWS pessoas usem essa chave KMS para operações criptográficas. Para fazer isso, na Contas da AWS seção Outros na parte inferior da página, escolha Adicionar outro Conta da AWS e insira o Conta da AWS ID de uma conta externa. Para adicionar várias contas externas, repita essa etapa.

 Note

Os administradores do outro também Contas da AWS devem permitir o acesso à chave KMS criando políticas do IAM para seus usuários. Para obter mais informações, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

17. Escolha Próximo.
18. Revise as configurações que você escolheu. Ainda é possível voltar e alterar todas as configurações.
19. Quando terminar, escolha Finish (Terminar) para criar a chave.

Método 2: comece com chaves gerenciadas pelo cliente

Esse procedimento é igual ao procedimento para criar uma chave de criptografia simétrica com material de AWS KMS chave. Mas, nesse procedimento, você especificará o ID de armazenamento de chaves personalizado do armazenamento de chaves externas e o ID da chave externa. Também é necessário especificar os [valores de propriedade obrigatórios](#) para uma chave do KMS em um armazenamento de chaves externas, como a especificação da chave e o uso da chave.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Escolha Criar chave.
5. Selecione Symmetric (Simétrica).
6. Em Key usage (Uso da chave), a opção Encrypt and decrypt (Criptografar e descriptografar) é selecionada para você. Não altere essa opção.
7. Escolha Advanced options (Opções avançadas).
8. Para Key material origin (Origem do material de chave), escolha External key store (Armazenamento de chaves externas).
9. Confirme que você pretende criar uma chave do KMS no armazenamento de chaves externas especificado.
10. Escolha Próximo.
11. Escolha a linha que representa o armazenamento de chaves externas para a nova chave do KMS.

Você pode escolher um armazenamento de chaves externas desconectado. Para conectar um armazenamento de chaves que esteja desconectado, escolha o nome do armazenamento de chaves e, em Key store actions (Ações do armazenamento de chaves), escolha Connect (Conectar). Para obter detalhes, consulte [Usando o AWS KMS console](#).

12. Insira o ID da [chave externa](#) no [gerenciador de chaves externas](#). Essa chave externa deve [atender aos requisitos](#) para uso com uma chave do KMS. Você não pode alterar o valor depois que a chave é criada.

Se a chave externa tiver várias IDs, insira a ID da chave que o proxy do armazenamento de chaves externo usa para identificar a chave externa.

13. Escolha Próximo.

O restante desse procedimento é o mesmo que a [criação de uma chave do KMS padrão](#).

14. Digite um alias e uma descrição opcional para a chave do KMS.
15. (Opcional). Na página Add Tags (Adicionar etiquetas), adicione etiquetas que identificam ou categorizam a chave do KMS.

Quando você adiciona tags aos seus AWS recursos, AWS gera um relatório de alocação de custos com uso e custos agregados por tags. Etiquetas também podem ser utilizadas para controlar o acesso a uma chave do KMS. Para informações sobre marcação de chaves do KMS, consulte [Etiquetas em AWS KMS](#) e [ABAC para AWS KMS](#).

16. Escolha Próximo.

17. Na seção Key Administrators (Administradores de chaves), selecione os usuários e as funções do IAM que podem gerenciar a chave do KMS. Para obter mais informações, consulte [Permite que administradores de chaves administrem a chave do KMS](#).

 Note

As políticas do IAM podem conceder permissão para usar a chave do KMS a outros usuários e funções do IAM.

18. (Opcional) Para evitar que esses administradores de chaves excluam essa chave do KMS, desmarque a caixa de seleção Allow key administrators to delete this key (Permitir que os administradores de chaves excluam essa chave).

Excluir uma chave do KMS é uma operação destrutiva e irreversível que pode tornar o texto cifrado irrecuperável. Você não pode recriar uma chave do KMS simétrica em um armazenamento de chaves externas, mesmo que você tenha o material de chave externa. No entanto, a exclusão de uma chave do KMS não afeta a chave externa associada. Para obter informações sobre como excluir uma chave do KMS de um armazenamento de chaves externas, consulte [Excluir um AWS KMS key](#).

19. Escolha Próximo.

20. Na seção Esta conta, selecione os usuários e funções do IAM Conta da AWS que podem usar a chave KMS em operações [criptográficas](#). Para obter mais informações, consulte [Permite que os usuários de chaves usem a chave do KMS](#).

 Note

As políticas do IAM podem conceder permissão para usar a chave do KMS a outros usuários e funções do IAM.

21. (Opcional) Você pode permitir que outras Contas da AWS pessoas usem essa chave KMS para operações criptográficas. Para fazer isso, na Contas da AWS seção Outros na parte inferior

da página, escolha Adicionar outro Conta da AWS e insira o Conta da AWS ID de uma conta externa. Para adicionar várias contas externas, repita essa etapa.

 Note

Os administradores do outro também Contas da AWS devem permitir o acesso à chave KMS criando políticas do IAM para seus usuários. Para obter mais informações, consulte [Permitir que usuários de outras contas usem uma chave do KMS](#).

22. Escolha Próximo.
23. Revise as configurações que você escolheu. Ainda é possível voltar e alterar todas as configurações.
24. Quando terminar, escolha Finish (Terminar) para criar a chave.

Se o procedimento for bem-sucedido, a tela exibirá a nova chave do KMS no armazenamento de chaves externas que você escolheu. Ao escolher o nome ou alias para a nova chave do KMS, a guia Cryptographic configuration (Configuração criptográfica) da página de detalhes exibe a origem da chave do KMS (External key store [Armazenamento de chaves externas]), o nome, o ID e o tipo de armazenamento de chaves personalizado, o uso da chave e o status da chave externa. Se houver falha no procedimento, uma mensagem descrevendo a falha será exibida. Em , consulte [Solução de problemas de armazenamentos de chaves externas](#).

 Tip

Para facilitar a identificação de chaves do KMS em um armazenamento de chaves personalizado, na página Customer managed keys (Chaves gerenciadas pelo cliente), adicione a coluna Origin (Origem) e Custom key store ID (ID de armazenamento de chaves personalizado) à exibição. Para alterar os campos da tabela, escolha o ícone de engrenagem no canto superior direito da página. Para obter detalhes, consulte [Personalizar a exibição do console](#).

Usando a AWS KMS API

Para criar uma nova chave KMS em um armazenamento de chaves externo, use a [CreateKey](#) operação. Os seguintes parâmetros são obrigatórios:

- O valor de Origin deve ser EXTERNAL_KEY_STORE.

- O parâmetro `CustomKeyStoreId` identifica o armazenamento de chaves externas. O [ConnectionState](#) do armazenamento de chaves externas especificado deve ser `CONNECTED`. Para encontrar `CustomKeyStoreId` e `ConnectionState`, use a operação `DescribeCustomKeyStores`.
- O parâmetro `XksKeyId` identifica a chave externa. Essa chave externa deve [atender aos requisitos](#) para associação com uma chave do KMS.

Você também pode usar qualquer um dos parâmetros opcionais da operação `CreateKey`, como usar os parâmetros de `Policy` ou de [Tags](#) (Etiquetas).

 Note

Não inclua informações confidenciais ou sigilosas nos campos `Description` ou `Tags`. Esses campos podem aparecer em texto simples em CloudTrail registros e outras saídas.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Este exemplo de comando usa a [CreateKey](#) operação para criar uma chave KMS em um armazenamento de chaves externo. A resposta inclui as propriedades das chaves do KMS, o ID do armazenamento de chaves externas e o ID, uso e status da chave externa.

Antes de executar um comando como esse, substitua o ID de exemplo do armazenamento de chaves personalizado por um ID válido.

```
$ 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"  
  }  
}  
}
```

Identificar e visualizar chaves

Você pode usar [Console de gerenciamento da AWS](#) ou a [API AWS Key Management Service \(AWS KMS\)](#) para visualizar AWS KMS keys em cada conta e região, incluindo as chaves KMS que você gerencia e as chaves KMS que são gerenciadas por AWS.

Tópicos

- [Encontrar o ID e o ARN da chave](#)
- [Acessar e listar detalhes de chaves do KMS](#)
- [Identificar diferentes tipos de chaves](#)
- [Personalizar a exibição do console](#)
- [Encontre chaves KMS e material de chaves em um armazenamento de AWS CloudHSM chaves](#)

Encontrar o ID e o ARN da chave

Para identificar um AWS KMS key, você pode usar o [ID da chave](#) ou o Amazon Resource Name ([ARN da chave](#)). Em [operações de criptografia](#), também é possível usar o [nome do alias](#) ou o [ARN do alias](#).

Você pode usar o [AWS KMS console](#) ou a [ListKeys](#) operação para identificar o ID e o ARN da chave de cada chave KMS em sua conta e região.

Para obter informações detalhadas sobre os identificadores de chave KMS compatíveis com AWS KMS, consulte [Identificadores-chave \(\) KeyId](#). Para obter ajuda para encontrar um nome de alias e um ARN de alias, consulte [Encontrar o nome e o ARN do alias para uma chave do KMS](#).

Usando o AWS KMS console

1. Abra o AWS KMS console em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. Para exibir as chaves em sua conta que você cria e gerencia, no painel de navegação, escolha Customer managed keys (Chaves gerenciadas de cliente). Para ver as chaves em sua conta que AWS cria e gerencia para você, no painel de navegação, escolha chaves AWS gerenciadas.
4. Para encontrar o [ID de chave](#) de uma chave do KMS, consulte a linha que começa com o alias da chave do KMS.

A coluna Key ID (ID de chave) aparece na tabela por padrão. Se a coluna Key ID (ID de chave) não for exibida na tabela, use o procedimento descrito em [the section called “Personalizar a exibição do console”](#) para restaurá-la. Também é possível visualizar o ID de chave de uma chave do KMS em sua página de detalhes.

Customer managed keys					Key actions ▼	Create key
					< 1 >	⚙
☐	Aliases ▲	Key ID ▼	Status	Creation date		
☐	key-test	1234abcd-12ab-34cd-56ef-1234567890ab	Enabled	Oct 19, 2018 12:43 PDT		

- Para encontrar o Amazon Resource Name (ARN) da chave do KMS, escolha o alias ou o ID da chave. O [ARN da chave](#) é exibido na seção General Configuration (Configuração geral).

General configuration		
Aliases key-test	Status Enabled	ARN arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
Description -	Creation date Nov 06, 2018 15:11 PST	

Usando a AWS KMS API

Para encontrar o [ID da chave](#) e o [ARN](#) da chave de um AWS KMS key, use a [ListKeys](#) operação.

A operação `ListKeys` retorna o ID de chave e o nome do recurso da Amazon (ARN) de todas as chaves do KMS na conta e na região do chamador.

Por exemplo, essa chamada para a operação `ListKeys` retorna o ID e o ARN de cada chave do KMS nessa conta fictícia. Para obter exemplos em várias linguagens de programação, consulte [Use ListKeys com um AWS SDK ou 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"
  }
]
```

Acessar e listar detalhes de chaves do KMS

Você pode usar o [AWS KMS console](#) ou a [DescribeKey](#) operação para acessar e listar informações detalhadas sobre as chaves KMS na conta e na região.

Os procedimentos a seguir demonstram como acessar os detalhes da chave do KMS, como ID da chave, especificação da chave, uso da chave e muito mais.

Usando o AWS KMS console

A página de detalhes de cada chave do KMS mostra as propriedades da chave do KMS. Ela difere ligeiramente de acordo com os diversos tipos de chaves do KMS.

Para exibir informações detalhadas sobre uma chave KMS, na página Chaves gerenciadas pela AWS ou Chaves gerenciadas pelo cliente, escolha o alias ou ID da chave KMS.

A página de detalhes de uma chave do KMS contém uma seção General configuration (Configuração Geral), que mostra as propriedades básicas da chave do KMS. Também inclui guias nas quais você pode visualizar e editar as propriedades de chave do KMS, como Política de chave, Configuração criptográfica, Tags, Material de chave e rodízios (para chaves do KMS compatíveis com rodízio automático ou sob demanda), Regionalidade (para chaves multirregionais) e Chave pública (para chaves do KMS assimétricas).

Note

O AWS KMS console exibe as chaves KMS que você tem [permissão para visualizar](#) em sua conta e região. As chaves KMS em outras Contas da AWS não aparecem no console,

mesmo que você tenha permissão para visualizá-las, gerenciá-las e usá-las. Para visualizar as chaves KMS em outras contas, use a [DescribeKey](#) operação.

Para navegar até a página de detalhes de uma chave do KMS.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. Para exibir as chaves em sua conta que você cria e gerencia, no painel de navegação, escolha Customer managed keys (Chaves gerenciadas de cliente). Para ver as chaves em sua conta que AWS cria e gerencia para você, no painel de navegação, escolha chaves AWS gerenciadas.
4. Para abrir a página de detalhes da chave, na tabela de chaves, escolha o ID ou alias da chave do KMS.

Se a chave do KMS incluir vários aliases, um resumo de aliases (+mais n) será exibido do lado do nome de um dos aliases. Escolher resumo de aliases leva você diretamente à guia Aliases na página de detalhes da chave.

KMS

>

Customer managed keys

>

Key ID: 6f84178a-007d-4501-8229-abc6af9e609c

6f84178a-007d-4501-8229-abc6af9e609c

Key actions

Edit

General configuration

Alias

kms-test

ARN

arn:aws:kms:us-east-1:849743133991:key/6f84178a-007d-4501-8229-abc6af9e609c

Current key material ID

f2e38520f5b8424d0ccd2dc64530a25e830926738224b0cb97d8e62302549f81

Status

Enabled

Description

-

Creation date

May 26, 2025 01:49 PDT

Regionality

Single Region

Key policy

Cryptographic configuration

Tags

Key material and rotations

Aliases

Cryptographic configuration

Key Type

Symmetric

Origin

AWS KMS

Key Spec

SYMMETRIC_DEFAULT

Key Usage

Encrypt and decrypt

A lista a seguir descreve os campos na exibição detalhada, incluindo o campo nas guias. Alguns desses campos também estão disponíveis como colunas na exibição de tabela.

Aliases

Local: guia Aliases

Um nome amigável para a chave do KMS. Você pode usar um alias para identificar a chave KMS no console e em alguns. AWS KMS APIs Para obter detalhes, consulte [Aliases em AWS KMS](#).

A guia Aliases exibe todos os aliases associados à chave KMS na região e. Conta da AWS

ARN

Local: seção General configuration (Configuração geral)

O nome do recurso da Amazon (ARN) da chave do KMS. Esse valor identifica exclusivamente a chave do KMS. É possível usá-lo para identificar a chave do KMS nas operações de API do AWS KMS .

Estado da conexão

Indica se um [armazenamento de chaves personalizado](#) está conectado a seu armazenamento de chaves de reserva. Esse campo é exibido somente quando a chave do KMS é criada em um armazenamento de chaves personalizado.

Para obter informações sobre os valores nesse campo, consulte [ConnectionState](#) na Referência da AWS KMS API.

Data de Criação

Local: seção General configuration (Configuração geral)

A data e a hora em que a chave do KMS foi criada. Esse valor é exibido na hora local do dispositivo. O fuso horário não depende da região.

Diferente de Expiration (Validade), a criação se refere somente à chave do KMS, e não ao seu material de chave.

ID do cluster do CloudHSM

Local: guia Cryptographic configuration (Configuração criptográfica)

O ID do AWS CloudHSM cluster que contém o material de chave para a chave KMS. Esse campo é exibido somente quando a chave do KMS é criada em um [armazenamento de chaves personalizado](#).

Se você escolher o ID do cluster do CloudHSM, ele abrirá a página Clusters no console. AWS CloudHSM

Material de chave atual

Local: seção General configuration (Configuração geral)

Chaves de criptografia simétrica com a origem AWS_KMS são compatíveis com rodízio automático e sob demanda. Chaves de criptografia simétricas com suporte de EXTERNAL origem para rotação sob demanda. Essas chaves podem ter vários materiais de chave associados a elas. O material de chave do rodízio mais recente pode ser usado tanto para criptografia quanto para descriptografia. Esse material chave é identificado como o material chave atual. Outros materiais de chave podem ser usados somente para descriptografia. O rodízio automático ou sob demanda de uma chave do KMS altera seu material de chave atual.

ID do armazenamento de chaves personalizado

Local: guia Cryptographic configuration (Configuração criptográfica)

O ID do [armazenamento de chaves personalizado](#) que contém a chave do KMS. Esse campo é exibido somente quando a chave do KMS é criada em um armazenamento de chaves personalizado.

Se você escolher o ID de armazenamento de chaves personalizadas, ele abrirá a página Armazenamentos de chaves personalizadas no AWS KMS console.

Nome do armazenamento de chaves personalizado

Local: guia Cryptographic configuration (Configuração criptográfica)

O nome do [armazenamento de chaves personalizado](#) que contém a chave do KMS. Esse campo é exibido somente quando a chave do KMS é criada em um armazenamento de chaves personalizado.

Tipo do armazenamento de chaves personalizado

Local: guia Cryptographic configuration (Configuração criptográfica)

Indica se o armazenamento de chaves personalizado é um [armazenamento de chaves do AWS CloudHSM](#) ou um [armazenamento de chaves externas](#). Esse campo é exibido somente quando a chave do KMS é criada em um [armazenamento de chaves personalizado](#).

Description

Local: seção General configuration (Configuração geral)

Uma descrição breve e opcional da chave do KMS que é possível escrever e editar. Para adicionar ou atualizar a descrição de uma chave gerenciada pelo cliente, acima de General Configuration (Configuração geral), selecione Edit (Editar).

Algoritmos de criptografia

Local: guia Cryptographic configuration (Configuração criptográfica)

Lista os algoritmos de criptografia que podem ser usados com a chave do KMS no AWS KMS. Esse campo é exibido somente quando o Key type (Tipo de chave) é Asymmetric (Assimétrico) e o Key usage (Uso da chave) é Encrypt and decrypt (Criptografar e descriptografar). Para obter informações sobre os algoritmos de criptografia que oferecem AWS KMS suporte, consulte [Especificação da chave SYMMETRIC_DEFAULT](#) [Especificações de chave RSA para criptografia e descriptografia](#) e.

Data de expiração

Local: guia Key material (Material da chave)

A data e a hora quando o material de chave da chave do KMS expira. Esse campo é exibido somente para chaves do KMS com [material de chave importado](#), ou seja, quando a Origem é Externa e a chave do KMS tem material de chave que expira. As chaves de criptografia simétricas podem ter vários materiais de chave associados a elas. Para essas chaves, esse campo indica a primeira data e hora de expiração de um dos materiais de chave associados.

ID da chave externa

Local: guia Cryptographic configuration (Configuração criptográfica)

O ID da [chave externa](#) associada a uma chave do KMS que está em um [armazenamento de chaves externas](#). Esse campo é exibido somente para chaves do KMS em um armazenamento de chaves externas.

Status da chave externa

Local: guia Cryptographic configuration (Configuração criptográfica)

O status mais recente que o [proxy do armazenamento de chaves externas](#) relatou para a [chave externa](#) associada à chave do KMS. Esse campo é exibido somente para chaves do KMS em um armazenamento de chaves externas.

Uso de chaves externas

Local: guia Cryptographic configuration (Configuração criptográfica)

As operações de criptografia que estão habilitadas na [chave externa](#) associada à chave do KMS. Esse campo é exibido somente para chaves do KMS em um armazenamento de chaves externas.

Política de chave

Local: guia Key policy (Política de chaves)

Controla o acesso à chave do KMS juntamente com [políticas do IAM](#) e [concessões](#). Cada chave do KMS tem uma política de chaves. Ela é o único elemento de autorização obrigatório. Para alterar a política de chaves de uma chave gerenciada pelo cliente, na guia Key policy (Política de chaves), selecione Edit (Editar). Para obter detalhes, consulte [the section called “Políticas de chaves”](#).

Material de chave e rodízios

Onde: guia Material de chave e rodízios

Essa guia é exibida somente para chaves de criptografia simétrica com a origem AWS_KMS (compatíveis com rodízio automático e sob demanda), bem como para chaves de criptografia simétrica de região única com a origem EXTERNAL (compatíveis com rodízio sob demanda).

A guia tem três painéis:

Rodízio automático: habilita e desabilita o [rodízio automático](#) do material de chave em uma [chave do KMS gerenciada pelo cliente](#). Para alterar o status de rodízio de uma [chave gerenciada pelo cliente](#), use a caixa de seleção. Não é possível habilitar ou desabilitar a alternância do material da chave em uma [Chave gerenciada pela AWS](#). As Chaves gerenciadas pela AWS são alternadas automaticamente a cada ano.

Rodízio sob demanda: inicie um [rodízio sob demanda](#) do material de chave em uma [chave gerenciada pelo cliente](#). Para chaves importadas, já deve haver um material de chave importado em estado PENDING_ROTATION para que a opção Fazer rodízio agora esteja disponível.

Materiais de chave: lista todos os materiais de chave associados à chave do KMS. Cada material de chave tem um identificador exclusivo e sua linha exibe informações adicionais sobre o material de chave, como a data de rodízio em que o material de chave ficou disponível para uso no KMS. Para chaves importadas, cada linha também tem um menu Ações que pode ser usado para excluir um material de chave específico ou reimportá-lo para a chave do KMS.

Especificação da chave

Local: guia Cryptographic configuration (Configuração criptográfica)

O tipo de material de chave na chave KMS. AWS KMS suporta chaves KMS de criptografia simétrica (SYMMETRIC_DEFAULT), chaves HMAC KMS de diferentes comprimentos, chaves KMS para chaves RSA de diferentes comprimentos e chaves de curva elíptica com curvas diferentes. Para obter detalhes, consulte [Key spec](#).

Tipo de chave

Local: guia Cryptographic configuration (Configuração criptográfica)

Indica se a chave do KMS é Simétrica ou Assimétrica.

Uso da chave

Local: guia Cryptographic configuration (Configuração criptográfica)

Indica se uma chave do KMS pode ser usada para criptografar e descriptografar, assinar e verificar ou gerar e verificar um MAC. Para obter detalhes, consulte [Key usage](#).

Origem

Local: guia Cryptographic configuration (Configuração criptográfica)

A origem do material de chave da chave do KMS. Os valores válidos são:

- AWS KMS para material de chaves gerado pelo AWS KMS
- AWS CloudHSM para chaves do KMS no [armazenamento de chaves do AWS CloudHSM](#)
- Externa para [material de chave importado](#) (BYOK)
- Armazenamento de chaves externas para chaves do KMS em um [armazenamento de chaves externas](#)

Algoritmos de Message authentication code (MAC – Código de autenticação de mensagem)

Local: guia Cryptographic configuration (Configuração criptográfica)

Lista os algoritmos de Message authentication code (MAC – Código de autenticação de mensagem) que podem ser usados com uma chave do KMS de HMAC no AWS KMS. Esse campo só aparece quando a Key spec (Especificação da chave) é uma especificação de chave de HMAC (HMAC_*). Para obter informações sobre os algoritmos de MAC compatíveis com o AWS KMS, consulte [Especificações de chave para chaves do KMS de HMAC](#).

Chave primária

Local: guia Regionality (Regionalidade)

Indica que essa chave do KMS é uma [chave primária de várias regiões](#). Os usuários autorizados podem usar essa seção para [transformar a chave primária](#) em uma chave de várias Regiões relacionada diferente. Esse campo só aparece quando a chave do KMS é uma chave primária de várias regiões.

Chave pública

Local: guia Public key (Chave pública)

Exibe a chave pública de uma chave do KMS assimétrica. Os usuários autorizados podem usar essa guia para [copiar e fazer download da chave pública](#).

Regionalidade

Local: seção General configuration (Configuração geral) e guias Regionality (Regionalidade)

Indica se uma chave do KMS é de região única, uma [chave primária de várias regiões](#) ou uma [chave de réplica de várias regiões](#). Esse campo só aparece quando a chave do KMS é uma chave de várias regiões.

Chaves de várias regiões relacionadas

Local: guia Regionality (Regionalidade)

Exibe todas as [chaves primárias e chaves de réplica de várias regiões](#) relacionadas, exceto a chave do KMS atual. Esse campo só aparece quando a chave do KMS é uma chave de várias regiões.

Na seção Related multi-Region keys (Chaves de várias regiões relacionadas) de uma chave primária, os usuários autorizados podem [criar novas chaves de réplica](#).

Chave de réplica

Local: guia Regionality (Regionalidade)

Indica que essa chave do KMS é uma [chave de réplica de várias regiões](#). Esse campo só aparece quando a chave do KMS é uma chave de réplica de várias regiões.

Algoritmos de assinatura

Local: guia Cryptographic configuration (Configuração criptográfica)

Lista os algoritmos de assinatura que podem ser usados com a chave do KMS no AWS KMS. Esse campo é exibido somente quando o Key type (Tipo de chave) é Asymmetric (Assimétrico) e o Key usage (Uso da chave) é Sign and verify (Assinar e verificar). Para obter informações sobre

os algoritmos de assinatura que oferecem AWS KMS suporte, consulte [Especificações de chave RSA para assinatura e verificação](#) [Especificações da chave de curva elíptica](#) e.

Status

Local: seção General configuration (Configuração geral)

O estado da chave do KMS. É possível usar a chave do KMS em [operações criptográficas](#) somente quando o status é Enabled (Habilitado). Para obter uma descrição detalhada do status de cada chave do KMS e seu efeito nas operações que podem ser executadas na chave do KMS, consulte [Principais estados das AWS KMS chaves](#).

Tags

Local: guia Tags (Etiquetas)

Pares de chave-valor opcionais que descrevem a chave do KMS. Para adicionar ou alterar as etiquetas de uma chave do KMS, na guia Tags (Etiquetas), selecione Edit (Editar).

Quando você adiciona tags aos seus AWS recursos, AWS gera um relatório de alocação de custos com uso e custos agregados por tags. Etiquetas também podem ser utilizadas para controlar o acesso a uma chave do KMS. Para informações sobre marcação de chaves do KMS, consulte [Etiquetas em AWS KMS](#) e [ABAC para AWS KMS](#).

Usando a AWS KMS API

A [DescribeKey](#) operação retorna detalhes sobre a chave KMS especificada. Para identificar a chave do KMS, use seu [ID de chave](#), [ARN de chave](#), [nome de alias](#) ou [ARN de alias](#).

Ao contrário da [ListKeys](#) operação, que exibe somente as chaves KMS na conta e na região do chamador, os usuários autorizados podem usar a DescribeKey operação para obter detalhes sobre as chaves KMS em outras contas.

Note

A resposta de DescribeKey inclui os membros KeySpec e CustomerMasterKeySpec com os mesmos valores. O membro CustomerMasterKeySpec é defasado.

Por exemplo, essa chamada para DescribeKey retorna informações sobre uma chave do KMS de criptografia simétrica. Os campos na resposta variam de acordo com a [especificação da AWS KMS](#)

[key](#), o [estado da chave](#) e a [origem do material de chave](#). Para obter exemplos em várias linguagens de programação, consulte [Use DescribeKey com um AWS SDK ou 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"
  }
}
```

Esse exemplo chama a operação `DescribeKey` em uma chave do KMS assimétrica usada para assinatura e verificação. A resposta inclui os algoritmos de assinatura compatíveis com o AWS KMS para essa chave do KMS.

```
$ aws kms describe-key --key-id 0987dcba-09fe-87dc-65ba-ab0987654321

{
  "KeyMetadata": {
    "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "Origin": "AWS_KMS",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
    "KeyState": "Enabled",
  }
}
```

```
{
  "KeyUsage": "SIGN_VERIFY",
  "CreationDate": 1569973196.214,
  "Description": "",
  "KeySpec": "ECC_NIST_P521",
  "CustomerMasterKeySpec": "ECC_NIST_P521",
  "AWSAccountId": "111122223333",
  "Enabled": true,
  "MultiRegion": false,
  "KeyManager": "CUSTOMER",
  "SigningAlgorithms": [
    "ECDSA_SHA_512"
  ]
}
```

Identificar diferentes tipos de chaves

Os tópicos a seguir explicam como identificar diferentes tipos de chaves no AWS KMS console e [DescribeKey](#) nas respostas.

Para obter ajuda na navegação até a guia Configuração criptográfica na página de detalhes de uma chave do KMS, consulte [the section called “Acessar e listar detalhes de chaves do KMS”](#).

Tópicos

- [Identificar chaves do KMS assimétricas](#)
- [Identificar chaves do KMS de HMAC](#)
- [Identificar chaves do KMS de várias regiões](#)
- [Identificar chaves do KMS com material de chave importado](#)
- [Identificar chaves KMS em lojas de AWS CloudHSM chaves](#)
- [Identificar chaves do KMS em repositórios de chaves externos](#)

Identificar chaves do KMS assimétricas

No AWS KMS console

A coluna Tipo de chave da tabela Chaves gerenciadas pelo cliente mostra se cada chave do KMS é simétrica ou assimétrica. Você pode filtrar a tabela pelo valor de Tipo de chave para exibir

somente chaves do KMS assimétricas. Para obter mais informações, consulte [the section called “Classificar e filtrar suas chaves do KMS”](#).

A guia Configuração criptográfica na página de detalhes de uma chave do KMS exibe o Tipo de chave, que indica se a chave é simétrica ou assimétrica. Ela também exibe o Uso da chave, que indica se sua chave do KMS assimétrica é usada para criptografia e descriptografia, assinatura e verificação ou derivação de segredos compartilhados.

Nas DescribeKey respostas

Quando você chama a operação `DescribeKey` em uma chave do KMS assimétrica, a resposta inclui os valores `KeySpec` e `KeyUsage`, que podem ser usados para determinar se uma chave do KMS é simétrica ou assimétrica.

Se o valor de `KeySpec` for `SYMMETRIC_DEFAULT`, a chave será uma chave do KMS de criptografia simétrica. Para obter detalhes sobre as especificações da chave assimétrica, consulte [Referência de especificação de chave](#).

Se o valor de `KeyUsage` for `SIGN_VERIFY` ou `KEY_AGREEMENT`, a chave será uma chave do KMS assimétrica.

A operação `DescribeKey` também retorna os detalhes a seguir para chaves do KMS assimétricas.

- Para chaves do KMS assimétricas com um valor `KeyUsage` de `ENCRYPT_DECRYPT`, a operação retorna `EncryptionAlgorithms`, que lista os algoritmos de criptografia válidos para a chave.
- Para chaves do KMS assimétricas com um valor `KeyUsage` de `SIGN_VERIFY`, a operação retorna `SigningAlgorithms`, que lista os algoritmos de assinatura válidos para a chave.
- Para chaves do KMS assimétricas com um valor `KeyUsage` de `KEY_AGREEMENT`, a operação retorna `KeyAgreementAlgorithms`, que lista os algoritmos de acordo de chave válidos para a chave.

Para obter mais informações sobre chaves do KMS assimétricas, consulte [the section called “Chaves assimétricas”](#).

Identificar chaves do KMS de HMAC

No AWS KMS console

As chaves do KMS de HMAC estão incluídas na tabela de Chaves gerenciadas pelo cliente, mas você não pode classificar ou filtrar essa tabela com base na especificação da chave ou nos valores de uso da chave que identificam as chaves de HMAC. Para facilitar a localização de suas chaves de HMAC, atribua um alias ou etiqueta distinta a elas. Em seguida, você pode classificar ou filtrar com base no alias ou etiqueta.

A guia Configuração criptográfica na página de detalhes de uma chave do KMS exibe o Tipo de chave, que indica se a chave é simétrica ou assimétrica. As chaves do KMS de HMAC são simétricas. A guia Configuração criptográfica também exibe o Uso da chave. O valor de uso de chave para chaves do KMS de HMAC sempre será Gerar e verificar MAC.

Nas DescribeKey respostas

Quando você chama a operação `DescribeKey` em uma chave do KMS de HMAC, a resposta inclui os valores `KeySpec` e `KeyUsage`. Para chaves do KMS de HMAC, o valor de uso da chave sempre será `GENERATE_VERIFY_MAC` e o valor da especificação da chave sempre começará com `HMAC_`.

Para obter mais informações sobre as chaves do KMS de HMAC, consulte [the section called “Chaves de HMAC”](#).

Identificar chaves do KMS de várias regiões

No AWS KMS console

A tabela Chaves gerenciadas pelo cliente mostra apenas as chaves do KMS na região selecionada. É possível visualizar chaves primárias e de réplica de várias regiões na região selecionada. Para alterar a AWS região, use o seletor de região no canto superior direito do console.

Para facilitar a identificação de chaves de várias regiões na tabela Chaves gerenciadas pelo cliente, adicione a coluna Regionalidade à sua tabela. Para obter ajuda, consulte [the section called “Personalizar suas tabelas de chaves do KMS”](#).

A página de detalhes das chaves do KMS de várias regiões inclui uma guia Regionalidade. A guia Regionality (Regionalidade) de uma chave primária inclui os botões Change primary Region

(Alterar região primária) e Create new replica keys (Criar novas chaves de réplica). (A guia Regionality (Regionalidade) de uma chave de réplica não tem botão.) A seção Related multi-Region keys (Chaves de várias regiões relacionadas) lista todas as chaves de várias regiões relacionadas à atual. Se a chave atual for uma chave de réplica, essa lista incluirá a chave primária.

Se você escolher uma chave multirregional relacionada na tabela Chaves multirregionais relacionadas, o AWS KMS console mudará para a região da chave selecionada e abrirá a página de detalhes da chave. Por exemplo, se você escolher a chave de réplica na sa-east-1 região na seção de exemplo de chaves multirregionais relacionadas abaixo, o AWS KMS console mudará para a sa-east-1 região para exibir a página de detalhes dessa chave de réplica. Você pode fazer isso para exibir o alias ou a política de chave da chave de réplica. Para alterar novamente a região, use o seletor de região no canto superior direito da página.

Nas DescribeKey respostas

Por padrão, as operações de AWS KMS API são regionais e retornam somente os recursos na região atual ou especificada. Porém, quando você chama a DescribeKey operação em uma chave KMS multirregional, a resposta inclui todas as chaves multirregionais relacionadas em outras AWS regiões no elemento. MultiRegionConfiguration

Para obter mais informações sobre as chaves do KMS de várias regiões, consulte [the section called “Chaves de várias regiões”](#).

Identificar chaves do KMS com material de chave importado

No AWS KMS console

Para facilitar a identificação de chaves do KMS com material de chave importado na tabela Chaves gerenciadas pelo cliente, adicione a coluna Origem à sua tabela. A coluna Origin (Origem) facilita a identificação das chaves do KMS com um valor de propriedade de origem External (Import Key material) (Externa [Importar material-chave]). Para obter ajuda, consulte [the section called “Personalizar suas tabelas de chaves do KMS”](#).

A guia Configuração criptográfica na página de detalhes de uma chave do KMS exibe a Origem, que identifica a origem do material da chave do KMS. Para chaves do KMS com material de chave importado, o valor de origem sempre será Externo (importar material de chave). A página de detalhes também inclui uma guia Material de chave que disponibiliza informações detalhadas sobre o material de chave importado. As chaves de criptografia simétricas com EXTERNAL origem

oferecem suporte a rotações sob demanda e podem ter vários materiais de chave associados a elas. Para essas chaves, a guia tem o rótulo Material da chave e rodízios.

Nas DescribeKey respostas

Quando você chama a operação `DescribeKey` em uma chave do KMS com material de chave importado, a resposta inclui os valores `Origin`, `ExpirationModel` e `ValidTo`. Para chaves do KMS com material de chave importado, o valor de origem sempre será `EXTERNAL`. O valor `ExpirationModel` indica se o material de chave está ou não configurado para expirar, e o valor `ValidTo` indica quando o material de chave expirará. Quando vários materiais de chave são associados a uma chave, o valor `ValidTo` indica a primeira data de expiração de validade de todos os materiais de chave (exceto o que está com rodízio pendente) e `ExpirationModel` é definido como `DOES_NOT_EXPIRE` somente se nenhum desses materiais de chave estiver definido para expirar. Para obter mais informações, consulte [Definir um prazo de validade \(opcional\)](#).

Para obter mais informações sobre chaves do KMS com material de chave importado, consulte [the section called “Material de chave importado”](#).

Identificar chaves KMS em lojas de AWS CloudHSM chaves

No AWS KMS console

Para facilitar a identificação de chaves KMS em AWS CloudHSM armazenamentos de chaves na tabela de chaves gerenciadas pelo cliente, adicione a coluna Origem à sua tabela. A coluna Origem facilita a identificação das chaves do KMS com um valor de propriedade de origem AWS CloudHSM. Para obter ajuda, consulte [the section called “Personalizar suas tabelas de chaves do KMS”](#).

A guia Configuração criptográfica na página de detalhes de uma chave do KMS exibe a Origem, que identifica a origem do material da chave do KMS. Para chaves KMS em armazenamentos de AWS CloudHSM chaves, o valor de origem é sempre AWS CloudHSM.

Para uma chave KMS em um armazenamento de AWS CloudHSM chaves, a guia Configuração criptográfica inclui uma seção adicional, Armazenamento de chaves personalizadas, que fornece informações sobre o armazenamento de chaves e o AWS CloudHSM cluster associado à AWS CloudHSM chave KMS.

Nas DescribeKey respostas

Quando você chama a operação `DescribeKey` em uma chave do KMS em um repositório de chaves do AWS CloudHSM, a resposta inclui `Origin`, que identifica a origem do material de chave. Para chaves KMS em um armazenamento de AWS CloudHSM chaves, o valor de origem é sempre `AWS_CLOUDHSM`. A operação também retorna os seguintes campos especiais para chaves KMS nos AWS CloudHSM armazenamentos de chaves:

- `CloudHsmClusterId`
- `CustomKeyStoreId`

Para obter mais informações sobre as AWS CloudHSM principais lojas, consulte [the section called “AWS CloudHSM lojas principais”](#).

Identificar chaves do KMS em repositórios de chaves externos

No AWS KMS console

Para facilitar a identificação de chaves do KMS em repositórios de chaves externos na tabela Chaves gerenciadas pelo cliente, adicione a coluna Origem à sua tabela. A coluna Origem facilita a identificação das chaves do KMS com um valor de propriedade de origem Repositório de chaves externo. Para obter ajuda, consulte [the section called “Personalizar suas tabelas de chaves do KMS”](#).

A guia Configuração criptográfica na página de detalhes de uma chave do KMS exibe a Origem, que identifica a origem do material da chave do KMS. Para chaves do KMS em repositórios de chaves externos, o valor de origem sempre será Repositório de chaves externo.

Para uma chave do KMS em um repositório de chaves externo, a guia Configuração criptográfica inclui mais duas seções, Repositório de chaves personalizado e Chave externa. A tabela Repositório de chaves personalizado fornece informações sobre o repositório de chaves externo associado à chave do KMS. A tabela Chave externa é exibida no console do AWS KMS somente para chaves do KMS em repositórios de chaves externos. Ela fornece informações sobre a chave externa associada à chave do KMS. A [chave externa](#) é uma chave criptográfica externa AWS que serve como material de chave para a chave KMS no armazenamento de chaves externo. Quando você criptografa ou descriptografa com a chave do KMS, a operação é executada pelo [gerenciador de chaves externas](#) usando a chave externa especificada.

Os valores a seguir são exibidos na seção External key (Chave externa).

ID da chave externa

O identificador da chave externa no gerenciador de chaves externas. Este é o valor que o proxy de armazenamento de chaves externas usa para identificar a chave externa. Você especifica o ID da chave externa ao criar a chave do KMS e não pode alterá-la. Se o valor do ID da chave externa que você usou para criar a chave do KMS for alterado ou se tornar inválido, você deverá [agendar a exclusão da chave do KMS](#) e [criar uma nova chave do KMS](#) com o valor correto do ID da chave externa.

Nas DescribeKey respostas

Quando você chama a operação `DescribeKey` em uma chave do KMS em um repositório de chaves externo, a resposta inclui `Origin`, que identifica a origem do material de chave. Para chaves KMS em um armazenamento de AWS CloudHSM chaves, o valor de origem é sempre `EXTERNAL_KEY_STORE`. A operação também retorna o elemento `CustomKeyId`, que identifica o repositório de chaves externo associado às chaves do KMS.

Para mais informações sobre repositórios de chaves externos, consulte [the section called “Armazenamentos de chaves externas”](#).

Personalizar a exibição do console

Você pode personalizar a visualização do AWS KMS console para facilitar a localização de suas chaves KMS. Personalize as tabelas que aparecem nas páginas Chaves gerenciadas pela AWS e Chaves gerenciadas pelo cliente para mostrar as informações de que você mais precisa, ou classifique e filtre as chaves do KMS retornadas nas tabelas.

Tópicos

- [Classificar e filtrar suas chaves do KMS](#)
- [Personalizar suas tabelas de chaves do KMS](#)

Classificar e filtrar suas chaves do KMS

Para facilitar a localização das chaves do KMS no console, é possível classificar e filtrar as tabelas de chaves.

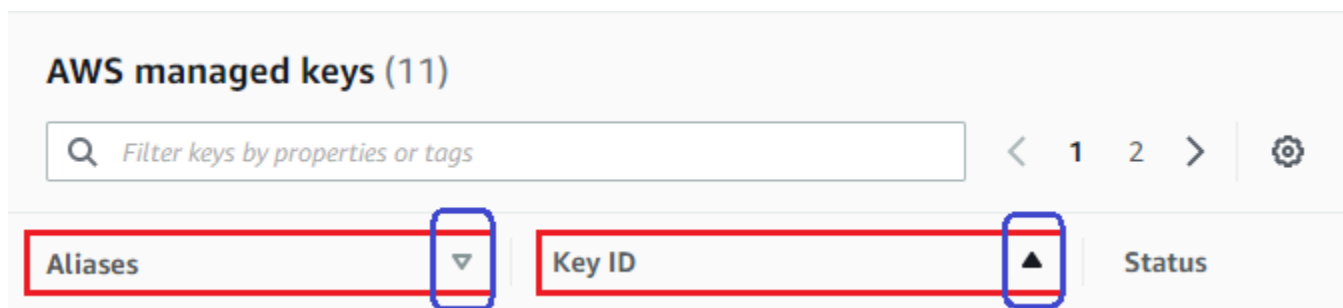
Classificar

É possível classificar chaves do KMS gerenciadas pelo cliente em ordem ascendente ou descendente pelos valores da coluna. Esse recurso classifica todas as chaves do KMS da tabela, mesmo se elas não forem exibidas na página atual da tabela.

As colunas classificáveis são indicadas por uma seta ao lado do nome da coluna. Na página Chaves gerenciadas pela AWS, você pode classificar por Aliases ou ID da chave. Na página Customer managed keys (Chaves gerenciadas pelo cliente), é possível classificar por Aliases, Key ID (ID de chave) ou Key type (Tipo de chave).

Para classificar em ordem ascendente, selecione o cabeçalho da coluna até que a seta aponte para cima. Para classificar em ordem descendente, selecione o cabeçalho da coluna até que a seta aponte para baixo. É possível classificar apenas por uma coluna de cada vez.

Por exemplo, é possível classificar chaves do KMS em ordem crescente por ID de chave, em vez de aliases, que é o padrão.



Ao classificar as chaves do KMS na página Customer managed keys (Chaves gerenciadas pelo cliente) em ordem crescente por Key type (Tipo de chave), todas as chaves assimétricas são exibidas antes de todas as chaves simétricas.

Filtro

É possível filtrar todas as chaves do KMS com base em seus valores de propriedade ou etiquetas. O filtro aplica-se a todas as chaves do KMS da tabela, mesmo se elas não forem exibidas na página atual da tabela. O filtro não diferencia letras maiúsculas de minúsculas.

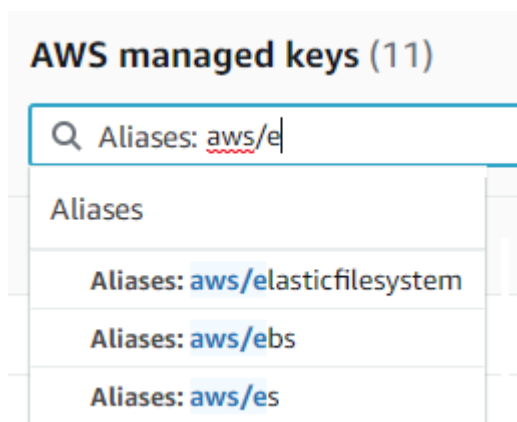
As propriedades filtráveis são listadas na caixa de filtro. Na página Chaves gerenciadas pela AWS, é possível filtrar por alias e ID da chave. Na página Customer managed keys (Chaves gerenciadas pelo cliente), é possível filtrar por alias, ID de chave e tipo de chave, bem como por etiquetas.

- Na página Chaves gerenciadas pela AWS, é possível filtrar por alias e ID da chave.

- Na página Customer managed keys (Chaves gerenciadas pelo cliente), é possível filtrar por etiquetas ou por alias, ID de chave, tipo de chave e regionalidade.

Para filtrar com base em um valor de propriedade, selecione o filtro, o nome da propriedade e uma opção na lista de valores de propriedade reais. Para filtrar com base em uma etiqueta, escolha a chave de etiqueta e depois selecione uma opção na lista de valores de etiquetas reais. Depois de escolher uma chave de etiqueta ou propriedade, também é possível digitar todo o valor da propriedade ou da etiqueta, ou parte do valor. Será exibida uma previsualização dos resultados antes de você fazer sua escolha.

Por exemplo, para exibir chaves do KMS com um nome de alias que contenha aws/e, escolha a caixa de filtro, escolha Alias, digite aws/e e pressione Enter ou Return para adicionar o filtro.



Filtros sugeridos para a tabela de chaves do KMS

Filtro para chaves do KMS assimétricas

Para exibir somente chaves do KMS assimétricas na página Customer managed keys (Chaves gerenciadas pelo cliente), clique na caixa de filtro, escolha Key type (Tipo de chave) e escolha Key type: Asymmetric (Tipo de chave: assimétrica). A opção Asymmetric (Assimétrica) é exibida somente quando você tem chaves do KMS assimétricas na tabela.

Filtro para chaves de várias regiões

Para exibir apenas as chaves de várias Regiões, na página Customer managed keys (Chaves gerenciadas pelo cliente), selecione a caixa de filtro, escolha Regionality (Regionalidade) e depois Regionality: Multi-Region (Regionalidade: várias regiões). A opção Multi-Region (Várias regiões) será exibida apenas quando você tiver chaves de várias Regiões na tabela.

Filtro para tags

Para exibir apenas chaves do KMS com uma etiqueta específica, selecione a caixa de filtro, a chave de etiqueta e entre os valores de etiquetas reais. Também é possível digitar todo o valor da etiqueta ou parte dele.

A tabela resultante mostra todas as chaves do KMS com a etiqueta selecionada. Porém, essa tabela não mostra a etiqueta. Para ver a etiqueta, selecione o alias ou o ID da chave da KMS e, na página de detalhes, escolha a guia Tags (Etiquetas). As guias aparecem sob a seção General configuration (Configuração geral).

Esse filtro exige a chave e o valor da etiqueta. Ele não localizará chaves do KMS apenas com a chave da etiqueta ou apenas com seu valor. Para filtrar as tags por toda ou parte da chave ou valor da tag, use a [ListResourceTags](#) operação para obter chaves KMS marcadas e, em seguida, use os recursos de filtragem da sua linguagem de programação.

Filtrar por texto

Para procurar texto, na caixa de filtragem, digite todo o alias, ou parte dele, o ID de chave, o tipo de chave ou a chave da etiqueta. (Após a seleção da chave da etiqueta, você poderá procurar um valor de etiqueta). Será exibida uma previsualização dos resultados antes de você fazer sua escolha.

Por exemplo, para mostrar chaves do KMS com test em suas chaves de etiqueta ou propriedades filtráveis, digite test na caixa de filtragem. A previsualização exibe as chaves do KMS que o filtro selecionará. Nesse caso, test aparece somente na propriedade Alias.

Personalizar suas tabelas de chaves do KMS

Você pode personalizar as tabelas que aparecem nas páginas de chaves gerenciadas pelo cliente Chaves gerenciadas pela AWS e nas páginas de chaves gerenciadas pelo cliente Console de gerenciamento da AWS para atender às suas necessidades. Você pode escolher as colunas da tabela, o número de AWS KMS keys em cada página (Tamanho da página) e a quebra de texto. A configuração escolhida será salva quando você confirmá-la e será reaplicada sempre que você abrir as páginas.

Como personalizar suas tabelas de chaves do KMS

1. Na página de Chaves gerenciadas pela AWS ou de chaves gerenciadas pelo cliente, escolha o ícone de configurações



no canto superior direito da página.

2. Na página Preferences (Preferências), escolha as configurações de preferência e selecione Confirm (Confirmar).

Considere o uso da configuração Page size (Tamanho da página) para aumentar o número de chaves do KMS exibidas em cada página, principalmente se você costuma usar um dispositivo que tem a rolagem fácil.

As colunas de dados exibidas podem variar dependendo da tabela, da função do trabalho e dos tipos de chaves do KMS na conta e na região. A tabela a seguir oferece algumas configurações sugeridas. Para obter descrições das colunas, consulte [Usando o AWS KMS console](#).

Configurações sugeridas para a tabela de chaves do KMS

Você pode personalizar as colunas que aparecem na tabela de chaves do KMS para exibir as informações necessárias sobre as suas chaves do KMS.

Chaves gerenciadas pela AWS

Por padrão, a tabela Chave gerenciada pela AWS exibe as colunas Aliases, ID da chave e Status. Essas colunas são ideais para a maioria dos casos de uso.

Chaves do KMS de criptografia simétrica

Se você usar somente chaves do KMS de criptografia simétrica com o material de chave gerado pelo AWS KMS, provavelmente as colunas Aliases, ID da chave, Status e Data de criação serão as mais úteis.

Chaves do KMS assimétricas

Se você usar chaves do KMS assimétricas, além das colunas Aliases, ID da chave e Status, considere adicionar as colunas Tipo de chave, Especificação da chave e Uso da chave. Essas colunas mostrarão se uma chave do KMS é simétrica ou assimétrica, o tipo do material de chave e se a chave do KMS pode ser usada para criptografia ou para assinatura.

Chaves do KMS de HMAC

Se você usar chaves do KMS HMAC, além das colunas Aliases, ID da chave e Status, avalie a possibilidade de adicionar as colunas Especificação de chave e Uso da chave. Essas colunas

mostrarão se uma chave do KMS é uma chave de HMAC. Como você não pode classificar as chaves KMS por especificação ou uso da chave, use aliases e tags para identificar suas chaves HMAC e, em seguida, use os [recursos de filtro](#) do AWS KMS console para filtrar por aliases ou tags.

Material de chave importado

Se você tiver chaves do KMS com [material de chave importado](#), considere adicionar as colunas Origin (Origem) e Expiration date (Data de validade). Essas colunas mostrarão se o material da chave em uma chave KMS é importado ou gerado por AWS KMS e quando o material da chave expira, se for o caso. O campo Creation date (Data de criação) exibe a data em que a chave do KMS foi criada (sem o material de chave). Ele não reflete nenhuma característica do material de chave.

Chaves em armazenamentos de chaves personalizados

Se você tiver chaves do KMS em [armazenamentos de chaves personalizados](#), considere adicionar as colunas Origin (Origem) e Custom key store ID (ID do armazenamento de chaves personalizado). Essas colunas informam se a chave do KMS está em um armazenamento de chaves personalizado, exibem o tipo do armazenamento de chaves personalizado e identificam o armazenamento de chaves personalizado.

Chaves de várias regiões

Se você tiver [chaves de várias regiões](#), considere adicionar a coluna Regionality (Regionalidade). Ela mostra se uma chave do KMS é de região única, uma [chave primária de várias regiões](#) ou uma [chave de réplica de várias regiões](#).

Encontre chaves KMS e material de chaves em um armazenamento de AWS CloudHSM chaves

Se você gerencia um armazenamento de AWS CloudHSM chaves, talvez seja necessário identificar as chaves KMS em cada armazenamento de AWS CloudHSM chaves. Por exemplo, pode ser necessário executar uma das seguintes tarefas.

- Rastreie as chaves KMS no armazenamento de AWS CloudHSM chaves nos AWS CloudTrail registros.
- Preveja o efeito da desconexão de um armazenamento de chaves nas AWS CloudHSM chaves KMS.

- Agende a exclusão das chaves KMS antes de excluir um armazenamento de AWS CloudHSM chaves.

Além disso, talvez você queira identificar as chaves em seu AWS CloudHSM cluster que servem como material chave para suas chaves KMS. Embora AWS KMS gerencie as chaves do KMS e o material de chaves, você ainda mantém o controle e a responsabilidade pelo gerenciamento do seu AWS CloudHSM cluster, bem como pelos backups HSMs e pelas chaves no HSMs. Talvez seja necessário identificar as chaves para auditar o material da chave, protegê-lo contra exclusão acidental ou excluí-lo dos backups do HSMs cluster após excluir a chave KMS.

Todo o material de chaves para as chaves KMS em seu armazenamento de AWS CloudHSM chaves é de propriedade do [usuário kmsuser criptográfico](#) (CU). AWS KMS define o atributo do rótulo da chave, que só pode ser visualizado em AWS CloudHSM, como o Amazon Resource Name (ARN) da chave KMS.

Para encontrar as chaves do KMS e o material de chave, use uma das técnicas a seguir.

- [Encontre as chaves KMS em um armazenamento de AWS CloudHSM chaves](#)— Como identificar as chaves KMS em um ou em todos os seus armazenamentos de AWS CloudHSM chaves.
- [Encontre todas as chaves de uma loja de AWS CloudHSM chaves](#): como encontrar todas as chaves no cluster que funcionam como material de chave para as chaves do KMS no armazenamento de chaves do AWS CloudHSM .
- [Encontre a AWS CloudHSM chave para uma chave KMS](#)— Como encontrar a chave em seu cluster que serve como material chave para uma chave KMS específica em seu armazenamento de AWS CloudHSM chaves.
- [Encontre a chave KMS para uma AWS CloudHSM chave](#): como encontrar a chave do KMS para uma determinada chave no seu cluster.

Encontre as chaves KMS em um armazenamento de AWS CloudHSM chaves

Se você gerencia um armazenamento de AWS CloudHSM chaves, talvez seja necessário identificar as chaves KMS em cada armazenamento de AWS CloudHSM chaves. Você pode usar essas informações para rastrear as operações da chave KMS em AWS CloudTrail registros, prever o efeito da desconexão de um armazenamento de chaves personalizado nas chaves KMS ou programar a exclusão das chaves KMS antes de excluir um armazenamento de chaves. AWS CloudHSM

Para encontrar as chaves KMS em um armazenamento de AWS CloudHSM chaves (console)

Para encontrar as chaves KMS em um repositório de AWS CloudHSM chaves específico, na página Chaves gerenciadas pelo cliente, visualize os valores nos campos Nome do armazenamento de chaves personalizadas ou ID do armazenamento de chaves personalizadas. Para identificar chaves KMS em qualquer armazenamento de AWS CloudHSM chaves, procure chaves KMS com um valor de origem de. AWS CloudHSM Para adicionar colunas opcionais à exibição, selecione o ícone de engrenagem no canto superior direito da página.

Para encontrar as chaves KMS em um armazenamento de AWS CloudHSM chaves (API)

Para encontrar as chaves KMS em um armazenamento de AWS CloudHSM chaves, use as [DescribeKey](#) operações [ListKeyse](#) e depois filtre por CustomKeyId valor. Antes de executar os exemplos a seguir, substitua os valores fictícios de ID do repositório de chaves personalizado por um valor válido.

Bash

Para encontrar chaves KMS em um determinado repositório de AWS CloudHSM chaves, obtenha todas as suas chaves KMS na conta e na região. Filtre pelo ID do armazenamento de chaves personalizado.

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;  
do aws kms describe-key --key-id $key |  
grep '"CustomKeyId": "cks-1234567890abcdef0"' --context 100; done
```

Para obter chaves KMS em qualquer armazenamento de AWS CloudHSM chaves na conta e na região, pesquise CustomKeyType com um valor deAWS_CloudHSM.

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;  
do aws kms describe-key --key-id $key |  
grep '"CustomKeyType": "AWS_CloudHSM"' --context 100; done
```

PowerShell

Para encontrar chaves KMS em um determinado armazenamento de AWS CloudHSM chaves, use os KmsKey cmdlets [Get-KmsKeyList](#) e [Get-](#) para obter todas as suas chaves KMS na conta e na região. Filtre pelo ID do armazenamento de chaves personalizado.

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq  
'cks-1234567890abcdef0'
```

Para obter as chaves KMS em qualquer armazenamento de AWS CloudHSM chaves na conta e na região, filtre pelo CustomKeyStoreType valor deAWS_CLOUDHSM.

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreType -eq 'AWS_CLOUDHSM'
```

Encontre todas as chaves de uma loja de AWS CloudHSM chaves

Você pode identificar as chaves em seu AWS CloudHSM cluster que servem como material chave para seu armazenamento de AWS CloudHSM chaves. Para fazer isso, use o comando [key list](#) na CLI do CloudHSM.

Você também pode usar o comando key list para encontrar a chave AWS KMS para uma AWS CloudHSM chave. Quando AWS KMS cria o material de chave para uma chave KMS em seu AWS CloudHSM cluster, ele grava o Amazon Resource Name (ARN) da chave KMS no rótulo da chave. O comando key list retorna key-reference e label.

Observações

Os procedimentos a seguir usam a ferramenta de linha de comando do AWS CloudHSM Client SDK 5, a CLI do [CloudHSM](#). A CLI do CloudHSM substitui key-handle por key-reference.

Em 1º de janeiro de 2025, AWS CloudHSM encerrará o suporte às ferramentas de linha de comando do Client SDK 3, ao CloudHSM Management Utility (CMU) e ao Key Management Utility (KMU). Para obter mais informações sobre as diferenças entre as ferramentas de linha de comando do Client SDK 3 e a ferramenta de linha de comando do Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) no Guia do usuário do AWS CloudHSM .

Para executar esse procedimento, você precisa desconectar temporariamente o armazenamento de AWS CloudHSM chaves para poder fazer login como kmsuser UC.

1. Desconecte o armazenamento de AWS CloudHSM chaves, se ele ainda não estiver desconectado, faça login com `comokmsuser`, conforme explicado em. [Como desconectar e fazer login](#)

 Note

Enquanto um armazenamento de chaves personalizado estiver desconectado, todas as tentativas de criar chaves do KMS nele ou de usar chaves do KMS existentes em operações de criptografia falharão. Essa ação pode impedir que os usuários armazenem e acessem dados sigilosos.

2. Use o comando [key list](#) na CLI do CloudHSM para encontrar todas as chaves do usuário atual presente em seu cluster. AWS CloudHSM

Por padrão, somente 10 chaves do usuário atualmente conectado são exibidas, e somente o `key-reference` e `label` são exibidos como saída. Para mais opções, consulte [key list](#) no Guia do usuário do AWS CloudHSM .

```
aws-cloudhsm > key list
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000000000123",
        "attributes": {
          "label": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
      },
      {
        "key-reference": "0x00000000000000456",
        "attributes": {
          "label": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
        }
      },
      ...8 keys later...
    ],
    "total_key_count": 56,
    "returned_key_count": 10,
```

```
"next_token": "10"
}
```

3. Saia e reconecte o armazenamento de AWS CloudHSM chaves conforme descrito em [Como fazer logout e se conectar novamente](#).

Encontre a chave KMS para uma AWS CloudHSM chave

Se você souber a referência de chave ou o ID de uma chave que `kmsuser` possui no cluster, você pode usar esse valor para identificar a chave KMS associada em seu armazenamento de AWS CloudHSM chaves.

Quando AWS KMS cria o material de chave para uma chave KMS em seu AWS CloudHSM cluster, ele grava o Amazon Resource Name (ARN) da chave KMS no rótulo da chave. A menos que tenha alterado o valor do rótulo, você pode usar o comando [key list](#) na CLI do CloudHSM para identificar a chave do KMS associada à chave do AWS CloudHSM.

Observações

Os procedimentos a seguir usam a ferramenta de linha de comando do AWS CloudHSM Client SDK 5, a CLI do [CloudHSM](#). A CLI do CloudHSM substitui `key-handle` por `key-reference`.

Em 1º de janeiro de 2025, AWS CloudHSM encerrará o suporte às ferramentas de linha de comando do Client SDK 3, ao CloudHSM Management Utility (CMU) e ao Key Management Utility (KMU). Para obter mais informações sobre as diferenças entre as ferramentas de linha de comando do Client SDK 3 e a ferramenta de linha de comando do Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) no Guia do usuário do AWS CloudHSM.

Para executar esses procedimentos, você precisa desconectar temporariamente o armazenamento de AWS CloudHSM chaves para poder fazer login como `kmsuser` UC.

Note

Enquanto um armazenamento de chaves personalizado estiver desconectado, todas as tentativas de criar chaves do KMS nele ou de usar chaves do KMS existentes em operações

de criptografia falharão. Essa ação pode impedir que os usuários armazenem e acessem dados sigilosos.

Tópicos

- [Identificar a chave do KMS associada a uma referência de chave](#)
- [Identificar a chave do KMS associada a um ID de chave de reserva](#)

Identificar a chave do KMS associada a uma referência de chave

Os procedimentos a seguir demonstram como usar o comando [key list](#) na CLI do CloudHSM com o filtro de atributos `key-reference` para encontrar a chave no seu cluster que atua como material de chave para uma chave do KMS específica no seu repositório de chaves do AWS CloudHSM .

1. Desconecte o armazenamento de AWS CloudHSM chaves, se ele ainda não estiver desconectado, faça login com `kmsuser`, conforme explicado em. [Como desconectar e fazer login](#)
2. Use o comando [key list](#) na CLI do CloudHSM para filtrar pelo atributo `key-reference`. Especifique o argumento `verbose` para incluir todos os atributos e as informações de chave da chave correspondente. Se você não especificar o argumento `verbose`, a operação `key list` retornará somente a referência de chave e o atributo de rótulo da chave correspondente.

Antes de executar esse comando, substitua o exemplo de `key-reference` por um elemento válido da sua conta.

```
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. Saia e reconecte o armazenamento de AWS CloudHSM chaves conforme descrito em [Como fazer logout e se conectar novamente](#).

Identificar a chave do KMS associada a um ID de chave de reserva

Todas as entradas de CloudTrail registro para operações criptográficas com uma chave KMS em um armazenamento de AWS CloudHSM chaves incluem um `additionalEventData` campo com `e.customKeyStoreId` `backingKeyId`. O valor retornado no campo `backingKeyId` está correlacionado ao atributo `id` da chave do CloudHSM. Você pode filtrar a operação [key list](#) pelo atributo `id` para identificar a chave do KMS associada a uma `backingKeyId` específica.

1. Desconecte o armazenamento de AWS CloudHSM chaves, se ele ainda não estiver desconectado, faça login como `okmsuser`, conforme explicado em. [Como desconectar e fazer login](#)
2. Use o comando [key list](#) na CLI do CloudHSM com o filtro de atributos para encontrar a chave no seu cluster que atua como material de chave para uma chave do KMS específica no seu repositório de chaves do AWS CloudHSM .

O exemplo a seguir demonstra como filtrar pelo `id` atributo. O AWS CloudHSM reconhece o valor de `id` como um valor hexadecimal. Para filtrar a operação da lista de chaves pelo `id` atributo, você deve primeiro converter o `backingKeyId` valor identificado na entrada do CloudTrail registro em um formato que AWS CloudHSM reconheça.

- a. Use o seguinte comando do Linux para converter `backingKeyId` em uma representação hexadecimal.

```
echo backingKeyId | tr -d '\n' | xxd -p
```

O exemplo a seguir demonstra como converter a matriz de bytes `backingKeyId` em uma representação hexadecimal.

```
echo 5890723622dc15f699aa9ab2387a9f744b2b884c18b2186ee8ada4f556a2eb9d | tr -d '\n' | xxd -p
353839303732333632326463313566363939616139616232333837613966373434623262383834633138623
```

- b. Prefixe a representação hexadecimal de `backingKeyId` com `0x`.

```
0x353839303732333632326463313566363939616139616232333837613966373434623262383834633138623
```

- c. Use o valor `backingKeyId` convertido para filtrar pelo atributo `id`. Especifique o argumento `verbose` para incluir todos os atributos e as informações de chave da chave

correspondente. Se você não especificar o argumento `verbose`, a operação `key list` retornará somente a referência de chave e o atributo de rótulo da chave correspondente.

```
aws-cloudhsm > key list --filter
attr.id="0x353839303732333632326463313566363939616139616232333837613966373434623262383
--verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000000120034",
        "key-info": {
          "key-owners": [
            {
              "username": "kmsuser",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "id":
"0x35383930373233363232646331356636393961613961623233383761396637343462326238383463313
          "check-value": "0x29bbd1",
          "class": "my_test_key",
          "encrypt": true,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": false,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": false,
```

```
        "trusted": false,
        "unwrap": true,
        "verify": false,
        "wrap": true,
        "wrap-with-trusted": false,
        "key-length-bytes": 32
    }
},
"total_key_count": 1,
"returned_key_count": 1
}
```

3. Saia e reconecte o armazenamento de AWS CloudHSM chaves conforme descrito em [Como fazer logout e se conectar novamente](#).

Encontre a AWS CloudHSM chave para uma chave KMS

Você pode usar o ID da chave KMS de uma chave KMS em um armazenamento de AWS CloudHSM chaves para identificar a chave em seu AWS CloudHSM cluster que serve como material de chave.

Quando AWS KMS cria o material de chave para uma chave KMS em seu AWS CloudHSM cluster, ele grava o Amazon Resource Name (ARN) da chave KMS no rótulo da chave. A menos que tenha alterado o valor do rótulo, você pode usar o comando [key list](#) na CLI do CloudHSM para encontrar o recurso de chave e o id do material de chave da chave do KMS.

Todas as entradas de CloudTrail registro para operação criptográfica com uma chave KMS em um armazenamento de AWS CloudHSM chaves incluem um `additionalEventData` campo com `e.customKeyId` e `backingKeyId`. O valor retornado no `backingKeyId` campo é o atributo `id` AWS CloudHSM chave. Você pode filtrar a operação da AWS CloudHSM CLI da lista de chaves pelo ARN da chave KMS para identificar o atributo da chave do CloudHSM associado a uma chave KMS específica. `id`

Para executar esse procedimento, você precisa desconectar temporariamente o armazenamento de AWS CloudHSM chaves para poder fazer login como `kmsuser` UC.

Observações

Os procedimentos a seguir usam a ferramenta de linha de comando do AWS CloudHSM Client SDK 5, a CLI do [CloudHSM](#). A CLI do CloudHSM substitui `key-handle` por `key-reference`.

Em 1º de janeiro de 2025, AWS CloudHSM encerrará o suporte às ferramentas de linha de comando do Client SDK 3, ao CloudHSM Management Utility (CMU) e ao Key Management Utility (KMU). Para obter mais informações sobre as diferenças entre as ferramentas de linha de comando do Client SDK 3 e a ferramenta de linha de comando do Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) no Guia do usuário do AWS CloudHSM.

1. Desconecte o armazenamento de AWS CloudHSM chaves, se ele ainda não estiver desconectado, faça login com `okmsuser`, conforme explicado em. [Como desconectar e fazer login](#)

Note

Enquanto um armazenamento de chaves personalizado estiver desconectado, todas as tentativas de criar chaves do KMS nele ou de usar chaves do KMS existentes em operações de criptografia falharão. Essa ação pode impedir que os usuários armazenem e acessem dados sigilosos.

2. Use o comando [key list](#) na CLI do CloudHSM e `label` filtre por para encontrar a chave KMS de uma chave específica em seu cluster. AWS CloudHSM Especifique o argumento `verbose` para incluir todos os atributos e as informações de chave da chave correspondente. Se você não especificar o argumento `verbose`, a operação `key list` retornará somente os atributos de referência e rótulo de chave da chave correspondente.

O exemplo a seguir demonstra como filtrar pelo atributo `label` que armazena o ARN da chave do KMS. Antes de executar esse comando, substitua o ARN de chave do KSM de exemplo por um valor válido da sua conta.

```
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": "0x00000000000120034",
      "key-info": {
        "key-owners": [
          {
            "username": "kmsuser",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "id": "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. Saia e reconecte o armazenamento de AWS CloudHSM chaves conforme descrito em [Como fazer logout e se conectar novamente](#).

Habilitar e desabilitar chaves

É possível desabilitar e reabilitar chaves gerenciadas pelo cliente. Ao criar uma chave do KMS, ela é habilitada por padrão. Se você desabilitar uma chave do KMS, ela não poderá ser usada em nenhuma [operação criptográfica](#) até que seja reabilitada.

Por ser temporário e facilmente desfeito, desabilitar uma chave do KMS é uma alternativa segura a excluir uma chave do KMS, que é uma ação destrutiva e irreversível. Se você estiver pensando em excluir uma chave KMS, desative-a primeiro e defina um [CloudWatch alarme](#) ou mecanismo semelhante para garantir que você nunca precise usar a chave para descriptografar dados criptografados.

Quando você desabilita uma chave do KMS, ela se torna inutilizável imediatamente (sujeita a consistência posterior). Porém, os recursos criptografados com [chaves de dados](#) protegidas pela chave do KMS não são afetados até que a chave do KMS seja usada novamente, por exemplo, para descriptografar a chave de dados. Esse problema afeta Serviços da AWS muitos dos quais usam chaves de dados para proteger seus recursos. Para obter detalhes, consulte [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#).

Você não pode ativar ou desativar [Chaves gerenciadas pela AWS](#) ou [Chaves pertencentes à AWS](#). Chaves gerenciadas pela AWS estão permanentemente habilitadas para uso por [serviços que usam AWS KMS](#). Chaves pertencentes à AWS são gerenciados exclusivamente pelo serviço que os possui.

Note

AWS KMS não altera o material da chave das chaves gerenciadas pelo cliente enquanto elas estão desativadas. Para obter mais informações, consulte [Funcionamento da alternância](#).

Usando o AWS KMS console

Você pode usar o AWS KMS console para ativar e desativar [as chaves gerenciadas pelo cliente](#).

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.

2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Marque a caixa de seleção das chaves do KMS que deseja habilitar ou desabilitar.
5. Para habilitar uma chave do KMS, selecione Key actions (Ações de chaves), Enable (Habilitar). Para desativar chave do KMS, escolha Key actions (Ações de chaves), Disable (Desabilitar).

Usando a AWS KMS API

A [EnableKey](#) operação ativa um desativado AWS KMS key. Estes exemplos usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível. O parâmetro `key-id` é obrigatório.

Esta operação não apresenta nenhuma saída. Para ver o status da chave, use a [DescribeKey](#) operação.

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

A [DisableKey](#) operação desativa uma chave KMS ativada. O parâmetro `key-id` é obrigatório.

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

Esta operação não apresenta nenhuma saída. Para ver o status da chave, use a [DescribeKey](#) operação e veja o `Enabled` campo.

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyMetadata": {
    "Origin": "AWS_KMS",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Description": "",
    "KeyManager": "CUSTOMER",
    "MultiRegion": false,
    "Enabled": false,
    "KeyState": "Disabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "CreationDate": 1502910355.475,
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333"
```

```
    "KeySpec": "SYMMETRIC_DEFAULT",  
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",  
    "EncryptionAlgorithms": [  
        "SYMMETRIC_DEFAULT"  
    ]  
}  
}
```

Girar AWS KMS keys

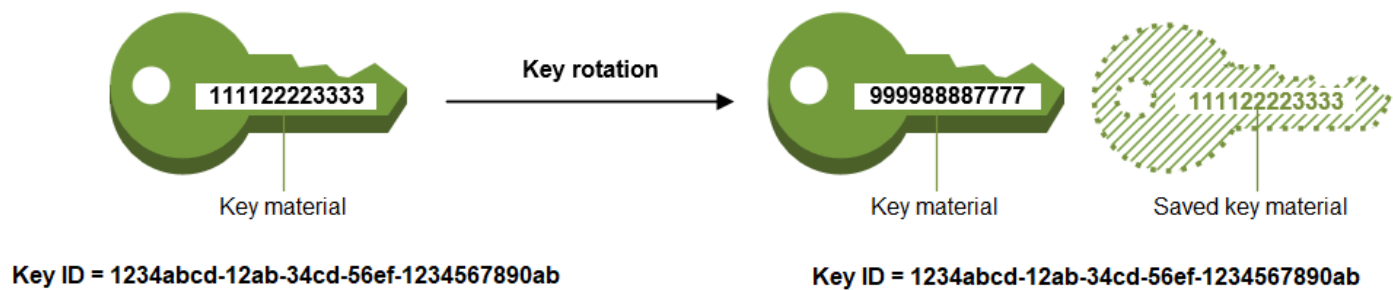
Para criar um novo material criptográfico para suas [chaves gerenciadas pelo cliente](#), você pode criar novas chaves do KMS e, em seguida, alterar suas aplicações ou seus aliases para usar essas novas chaves do KMS. Como alternativa, você pode alternar o material de chave associado a uma chave existente do KMS habilitando a alternância automática da chave ou executando a alternância sob demanda.

Por padrão, quando você ativa a rotação automática de chaves para uma chave KMS, AWS KMS gera novo material criptográfico para a chave KMS todos os anos. Você também pode especificar um personalizado [rotation-period](#) para definir o número de dias após ativar a rotação automática de chaves que AWS KMS girará o material da chave e o número de dias entre cada rotação automática a partir de então. Se precisar iniciar imediatamente a alternância do material de chave, você poderá realizar a alternância sob demanda, independentemente de a alternância automática de chaves estar ou não habilitada. As alternâncias sob demanda não alteram os cronogramas de alternância automática existentes.

Você pode [acompanhar a rotação](#) do material de chaves para suas chaves KMS na Amazon CloudWatch e no AWS Key Management Service console. AWS CloudTrail Você também pode usar a [GetKeyRotationStatus](#) operação para verificar se a rotação automática está habilitada para uma chave KMS e identificar quaisquer rotações sob demanda em andamento. Você pode usar a [ListKeyRotations](#) operação para visualizar os detalhes das rotações concluídas.

O rodízio de chaves troca somente o material de chave, que é o segredo criptográfico usado em operações de criptografia. Quando você usa a chave KMS rotacionada para descriptografar texto cifrado, AWS KMS usa o material de chave que foi usado para criptografá-la. Não é possível selecionar um material de chave específico para operações de descriptografia. O AWS KMS escolhe automaticamente a versão correta do material de chave. Como a decodificação é AWS KMS transparente com o material de chave apropriado, você pode usar com segurança uma chave KMS girada em aplicativos e sem alterações no código. Serviços da AWS

A chave do KMS é o mesmo recurso lógico, independentemente da ocorrência de alterações ou do número de vezes que estas tenham sido feitas no material de chave. As propriedades da chave do KMS não são alteradas, conforme mostrado na imagem a seguir.



Você pode optar por criar uma nova chave do KMS e usá-la em vez da chave do KMS original. Isso tem o mesmo efeito que alternar o material da chave em uma chave do KMS existente e, portanto, normalmente é considerado como [alternar manualmente a chave](#). [A rotação manual é uma boa opção quando você deseja girar chaves KMS que não estão qualificadas para rotação automática ou sob demanda, incluindo chaves KMS assimétricas, chaves HMAC KMS e chaves KMS em lojas de chaves personalizadas.](#)

Note

O rodízio de chaves não afeta os dados protegidos pela chave do KMS. Ele não faz rodízio das chaves de dados geradas pela chave do KMS nem criptografa novamente nenhum dado protegido pela chave do KMS. O rodízio de chaves não mitiga o efeito de uma chave de dados comprometida.

Alternância de chaves e definição de preço

AWS KMS cobra uma taxa mensal pela primeira e segunda rotação do material de chave mantido para sua chave KMS. Esse aumento de preço é limitado à segunda alternância, e qualquer alternância subsequente não será cobrada. Para obter mais informações, consulte [Definição de preço do AWS Key Management Service](#).

Note

É possível usar o [AWS Cost Explorer Service](#) para ver um detalhamento das cobranças de armazenamento de chaves. Por exemplo, é possível filtrar a visualização para ver o total de cobranças de chaves cobradas como chaves KMS atuais e alternadas especificando \$REGION-KMS-Keys para o Tipo de Uso e agrupando os dados por Operação de API.

Talvez você ainda veja instâncias da operação da API Unknown legada para datas históricas.

Alternância de chaves e cotas

Cada chave do KMS conta como uma chave ao calcular cotas de recursos de chaves, independentemente do número de versões de material de chave alternadas.

Para obter informações detalhadas sobre chaves de backup e alternância, consulte o [Detalhes criptográficos do AWS Key Management Service](#).

Tópicos

- [Por que alternar as chaves do KMS?](#)
- [Funcionamento da alternância](#)
- [Habilitar a alternância automática de chave](#)
- [Desabilitar a alternância automática de chave](#)
- [Executar alternância de chaves sob demanda](#)
- [Listar rodízios e materiais de chave](#)
- [Alternar chaves manualmente](#)
- [Alterar a chave primária em um conjunto de chaves de várias regiões](#)

Por que alternar as chaves do KMS?

As melhores práticas criptográficas desencorajam a reutilização extensiva de chaves que criptografam dados diretamente, como as chaves de [dados](#) geradas. AWS KMS Quando as chaves de dados de 256 bits criptografam milhões de mensagens, elas podem se esgotar e começar a produzir texto cifrado com padrões sutis que atores inteligentes podem explorar para descobrir os bits da chave. É melhor usar as chaves de dados uma vez ou somente algumas vezes para reduzir esse esgotamento de chaves.

No entanto, as chaves do KMS são mais frequentemente usadas como chaves de empacotamento, também conhecidas como chaves de criptografia de chaves. Em vez de criptografar dados, as chaves de empacotamento criptografam as chaves de dados que criptografam seus dados. Dessa forma, elas são usadas com muito menos frequência do que as chaves de dados e quase nunca são reutilizadas o suficiente para correr o risco de esgotamento das chaves.

Apesar desse risco de exaustão muito baixo, talvez seja necessário alternar suas chaves do KMS devido a regras comerciais ou contratuais, ou regulamentações governamentais. Quando for obrigado a fazer o rodízio de chaves do KMS, recomendamos que você use o rodízio de chaves automático, sempre que possível, use o rodízio de chaves manual quando o rodízio de chaves automático não for compatível e o rodízio de chaves manual quando nem o rodízio automático nem o rodízio sob demanda forem compatíveis.

Você poderá considerar realizar alternâncias sob demanda para demonstrar os principais recursos de alternância de material de chave ou para validar scripts de automação. Recomendamos usar rodízios sob demanda para rodízios não planejados e usar o rodízio de chaves automático com um [período de rodízio](#) personalizado sempre que possível.

Funcionamento da alternância

AWS KMS a rotação da chave foi projetada para ser transparente e fácil de usar. AWS KMS suporta rotação opcional de chaves automática e sob demanda somente para [chaves gerenciadas pelo cliente](#).

Alternância automática de chaves

AWS KMS gira a chave KMS automaticamente na próxima data de rotação definida pelo seu período de rotação. Você não precisa lembrar nem programar a atualização.

A rotação automática de chaves é suportada somente em chaves KMS de criptografia simétrica com material de chave que AWS KMS gera (AWS_KMSorigem).

A rotação automática é opcional para chaves KMS gerenciadas pelo cliente. AWS KMS sempre alterna o material da chave para as chaves AWS gerenciadas do KMS todos os anos. A rotação das chaves KMS de AWS propriedade é gerenciada pelo AWS service (Serviço da AWS) proprietário da chave.

Alternância sob demanda

Inicie imediatamente a alternância do material de chave associado à sua chave do KMS, independentemente de a alternância automática de chaves estar ou não habilitada.

A rotação de chaves sob demanda é suportada em chaves KMS de criptografia simétrica com material de chave que AWS KMS gera (AWS_KMSorigem) e chaves KMS de criptografia simétrica com material de chave importado (origem). EXTERNAL

Rodízio manual

Nem o rodízio automático nem o rodízio sob demanda são compatíveis com os seguintes tipos de chave do KMS, mas você pode [fazer o rodízio dessas chaves do KMS manualmente](#).

- [Chaves do KMS assimétricas](#)
- [Chaves do KMS de HMAC](#)
- Chaves do KMS em [armazenamentos de chaves personalizados](#)

Gerenciamento do material de chave

AWS KMS retém todo o material de chaves de uma chave KMS com AWS_KMS origem, mesmo se a rotação de chaves estiver desativada. AWS KMS exclui o material da chave somente quando você exclui a chave KMS.

Você gerencia os materiais de chave para chaves de criptografia simétrica com a origem EXTERNAL. Você pode excluir qualquer material chave usando a [DeleteImportedKeyMaterial](#) operação ou definir um prazo de validade ao importar o material. A chave do KMS se torna inutilizável no momento que algum de seus materiais expira ou é excluído.

Uso do material de chave

Quando você usa uma chave KMS rotacionada para criptografar dados, AWS KMS usa o material de chave atual. Quando você usa a chave do KMS alternada para descriptografar texto cifrado, o AWS KMS usa a mesma versão do material de chave que foi usado para criptografá-lo. Você não pode selecionar uma versão específica do material chave para operações de descriptografia, escolhe AWS KMS automaticamente a versão correta.

Período de alternância

O período de rotação define o número de dias após você ativar a rotação automática de chaves que AWS KMS girará seu material de chaves e o número de dias entre cada rotação automática de chaves a partir de então. Se você não especificar um valor para `RotationPeriodInDays` ao habilitar a alternância automática de chaves, o valor padrão será de 365 dias.

Você pode usar a chave de `RotationPeriodInDays` condição [kms:](#) para restringir ainda mais os valores que os principais podem especificar no parâmetro. `RotationPeriodInDays`

Data de rotação

A data do rodízio corresponde a data em que o material de chave atual de uma chave do KMS foi atualizado como resultado do rodízio automático (programado) ou de um rodízio de chaves sob demanda.

Data de alternância

AWS KMS gira automaticamente a chave KMS na data de rotação definida pelo seu período de rotação. O período de alternância padrão é de 365 dias.

Chaves gerenciadas pelo cliente

Como a alternância automática de chaves é opcional nas [chaves gerenciadas pelo cliente](#) e pode ser ativada e desativada a qualquer momento, a data de alternância dependerá da data em que a alternância foi ativada mais recentemente. A data pode mudar se você modificar o período de alternância para uma chave na qual você habilitou anteriormente a alternância automática de chaves. A data de alternância pode mudar várias vezes ao longo da vida útil da chave.

Por exemplo, se você criar uma chave gerenciada pelo cliente em 1.º de janeiro de 2022 e habilitar a alternância automática de chaves com o período padrão de alternância de 365 dias em 15 de março de 2022, o AWS KMS alternará o material da chave em 15 de março de 2023, 15 de março de 2024 e a cada 365 dias desse ponto em diante.

Os exemplos a seguir pressupõem que a alternância automática de chaves foi habilitada com o período de alternância padrão de 365 dias. Esses exemplos apresentam casos especiais que podem afetar o período de alternância de uma chave.

- Desabilitar a alternância de chaves — Se você [desabilitar a alternância de chaves automática](#) a qualquer momento, a chave do KMS continuará usando a versão do material de chaves que estava usando quando a alternância foi desabilitada. Se você ativar a rotação automática da chave novamente, AWS KMS gira o material da chave com base na nova data de ativação da rotação.
- Chaves KMS desativadas — Quando uma chave KMS está desativada, AWS KMS ela não é girada. No entanto, o status da alternância de chaves não muda, e você não pode alterá-lo enquanto a chave do KMS está desabilitada. Quando a chave KMS é reativada, se o material da chave tiver passado da última data de rotação programada, ele será AWS KMS rotacionado imediatamente. Se o material da chave não tiver perdido a última data de rotação programada, AWS KMS retoma a programação de rotação da chave original.

- Chaves KMS com exclusão pendente — Enquanto uma chave KMS está pendente de exclusão, AWS KMS não a gira. O status da alternância de chaves é definido como `false` e você não pode alterá-lo enquanto a exclusão estiver pendente. Se a exclusão for cancelada, o status da alternância de chaves anterior será restaurado. Se o material principal tiver passado da última data de rotação programada, ele será AWS KMS rotacionado imediatamente. Se o material da chave não tiver perdido a última data de rotação programada, AWS KMS retoma a programação de rotação da chave original.

Chaves gerenciadas pela AWS

AWS KMS gira automaticamente a Chaves gerenciadas pela AWS cada ano (aproximadamente 365 dias). Não é possível habilitar ou desabilitar a alternância de chaves para [Chaves gerenciadas pela AWS](#).

O material principal de um Chave gerenciada pela AWS é rotacionado primeiro um ano após a data de criação e, a partir de então, todos os anos (aproximadamente 365 dias a partir da última rotação).

Note

Em maio de 2022, AWS KMS alterou o cronograma de rotação Chaves gerenciadas pela AWS de três em três anos (aproximadamente 1.095 dias) para todos os anos (aproximadamente 365 dias).

Chaves pertencentes à AWS

Não é possível habilitar ou desabilitar a alternância de chaves para Chaves pertencentes à AWS. A estratégia de [rotação de chaves](#) para um Chave pertencente à AWS é determinada pelo AWS serviço que cria e gerencia a chave. Para obter detalhes, consulte o tópico Criptografia em repouso, no manual do usuário ou no guia do desenvolvedor do serviço.

Alternância de chaves de várias regiões

O comportamento de rotação difere dependendo se o material chave é gerado por AWS KMS (AWS_KMSorigem) ou importado (EXTERNALorigem).

Chaves multirregionais com origem **AWS_KMS**

É possível habilitar e desabilitar o rodízio automático e realizar o rodízio sob demanda do material de chave em [chaves multirregionais](#) de criptografia simétrica com a origem AWS_KMS. A alternância de chaves é uma [propriedade compartilhada](#) de chaves de várias regiões.

Você habilita e desabilita a alternância automática de chaves somente na chave primária. Você inicia a alternância sob demanda apenas na chave primária.

- Quando AWS KMS sincroniza as chaves multirregionais, ele copia a configuração da propriedade de rotação da chave primária para todas as chaves de réplica relacionadas.
- Quando AWS KMS gira o material da chave, ele cria um novo material de chave para a chave primária e, em seguida, copia o novo material de chave além dos limites da região para todas as chaves de réplica relacionadas. O material da chave nunca sai AWS KMS sem criptografia. Essa etapa é cuidadosamente controlada para garantir que o material de chave seja totalmente sincronizado antes que qualquer chave seja usada em uma operação criptográfica.
- AWS KMS não criptografa nenhum dado com o novo material de chave até que esse material esteja disponível na chave primária e em cada uma de suas chaves de réplica.
- Quando você replica uma chave primária que foi alternada, a nova chave de réplica tem o material de chave atual e todas as versões anteriores do material de chave para suas chaves de várias regiões relacionadas.

Esse padrão garante que as chaves de várias regiões relacionadas sejam totalmente interoperáveis. Qualquer chave de várias regiões pode descriptografar qualquer texto cifrado por uma chave de várias regiões relacionada, mesmo que esse texto cifrado tenha sido criptografado antes de a chave ser criada.

Chaves multirregionais com origem **EXTERNAL**

Você pode realizar a rotação sob demanda do material da chave em [chaves multirregionais](#) de criptografia simétrica com origem. EXTERNAL A alternância de chaves é uma [propriedade compartilhada](#) de chaves de várias regiões.

Você inicia a rotação sob demanda somente na chave primária depois de importar o novo material de chave para a chave primária e cada chave de réplica.

- Ao importar um novo material de chave para a chave primária, AWS KMS copia o identificador do material-chave e a descrição do material-chave da chave primária para todas as chaves de réplica relacionadas. Ele não copia o material chave.
- Você deve importar o mesmo material de chave em cada chave de réplica individualmente. Depois que o material da chave for importado para todas as chaves multirregionais relacionadas, você poderá iniciar a rotação sob demanda na chave primária. Isso garante que AWS KMS não criptografe nenhum dado com o novo material de chave até que esse material esteja disponível na chave primária e em cada uma de suas chaves de réplica.

- Cada material de chave na chave primária ou em qualquer chave de réplica pode expirar ou ser excluído independentemente de outro material de chave na mesma chave ou em qualquer outra chave multirregional relacionada.

AWS serviços

Você pode ativar a rotação automática de chaves nas [chaves gerenciadas pelo cliente](#) que você usa para criptografia do lado do servidor nos serviços. AWS A rotação anual é transparente e compatível com AWS os serviços.

Monitoramento da alternância de chaves

Quando AWS KMS gira o material da chave para uma [chave gerenciada Chave gerenciada pela AWS](#)[Sou gerenciada pelo cliente](#), ele grava um KMS CMK Rotation evento na Amazon EventBridge e um [RotateKey evento](#) em seu AWS CloudTrail registro. É possível usar esses registros para verificar se a chave do KMS foi alternada.

Você pode usar o AWS Key Management Service console para visualizar o número de rotações sob demanda restantes e uma lista de todas as rotações de material-chave concluídas para uma chave KMS.

Você pode usar a [ListKeyRotations](#) operação para visualizar os detalhes das rotações concluídas.

Consistência eventual

A alternância de chaves está sujeita aos mesmos efeitos de consistência eventual de outras operações de gerenciamento do AWS KMS . Pode haver um pequeno atraso antes que o novo material de chave esteja disponível no AWS KMS. Porém, o material de chave alternante não causa interrupção ou atraso em operações de criptografia. O material de chave atual será usado em operações de criptografia até que o novo material de chave esteja disponível no AWS KMS. Quando o material-chave de uma chave multirregional é rotacionado automaticamente, AWS KMS usa o material de chave atual até que o novo material de chave esteja disponível em todas as regiões com uma chave multirregional relacionada.

Habilitar a alternância automática de chave

Por padrão, quando você ativa a rotação automática de chaves para uma chave KMS, AWS KMS gera novo material criptográfico para a chave KMS todos os anos. Você também pode especificar um personalizado [rotation-period](#) para definir o número de dias após ativar a rotação automática de chaves que AWS KMS girará o material da chave e o número de dias entre cada rotação automática a partir de então.

A alternância de chaves automática tem os seguintes benefícios:

- As propriedades da chave do KMS, incluindo seu [ID de chave](#), [ARN de chave](#), região, políticas e permissões, não são alteradas quando há alternância na chave.
- Não é necessário alterar aplicações ou aliases que fazem referência ao ID de chave ou ARN de chave da chave do KMS.
- A alternância do material de chave não afeta o uso da chave do KMS em nenhum AWS service (Serviço da AWS).
- Depois de ativar a rotação da chave, AWS KMS gira a chave KMS automaticamente na próxima data de rotação definida pelo seu período de rotação. Você não precisa lembrar nem programar a atualização.

Você pode ativar a rotação automática da chave no AWS KMS console ou usando a [EnableKeyRotation](#) operação. Para habilitar a alternância automática de chave, você precisará de permissões `kms:EnableKeyRotation`. Para obter mais informações sobre AWS KMS permissões, consulte [Referência de permissões](#).

Usando o AWS KMS console

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. (Não é possível habilitar ou desabilitar a alternância de Chaves gerenciadas pela AWS. Elas são alternadas automaticamente a cada ano.)
4. Escolha o alias ou ID de chave de uma chave do KMS.
5. Selecione a guia Key rotation (Rotação de chaves).

A guia Rotação de chaves aparece somente na página de detalhes das chaves KMS de criptografia simétrica com o material de chave AWS KMS gerado (a origem é `AWS_KMS`), incluindo chaves KMS de criptografia simétrica [multirregional](#).

Não é possível alternar automaticamente chaves do KMS assimétricas, chaves do KMS de HMAC, chaves do KMS com [material de chave importado](#) nem chaves do KMS em [armazenamentos personalizados de chave](#). No entanto, é possível [alterná-las manualmente](#).

6. Na seção Alternância automática de chaves, escolha Editar.

7. Em Alternância de chaves, selecione Habilitar.

Note

Se uma chave KMS estiver desativada ou pendente de exclusão, AWS KMS ela não rotacionará o material da chave e você não poderá atualizar o status da rotação automática da chave ou o período de rotação. Para atualizar a configuração de alternância automática de chaves, habilite a chave do KMS ou cancele a exclusão. Para obter mais detalhes, consulte [Funcionamento da alternância](#) e [Principais estados das AWS KMS chaves](#).

8. (Opcional) Digite um período de alternância entre 90 e 2.560 dias. O valor padrão é de 365 dias. Se você não especificar um período de rotação personalizado, AWS KMS alternará o material da chave todos os anos.

Você pode usar a chave de RotationPeriodInDays condição [kms:](#) para limitar os valores que os diretores podem especificar para o período de rotação.

9. Escolha Salvar.

Usando a AWS KMS API

É possível usar a [API do AWS Key Management Service \(AWS KMS\)](#) para habilitar a alternância automática de chaves e visualizar o status atual da alternância de qualquer chave gerenciada pelo cliente. Estes exemplos usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

A [EnableKeyRotation](#) operação permite a rotação automática de chaves para a chave KMS especificada. Para identificar a chave do KMS nessa operação, use seu [ID de chave](#) ou [ARN de chave](#). Por padrão, a mudança de chaves está desabilitada nas chaves gerenciadas de cliente.

Você pode usar a chave de [kms:RotationPeriodInDays](#) condição para limitar os valores que os diretores podem especificar para o RotationPeriodInDays parâmetro de uma EnableKeyRotation solicitação.

O exemplo a seguir permite a rotação de chaves com um período de rotação de 180 dias na chave KMS de criptografia simétrica especificada e usa a [GetKeyRotationStatus](#) operação para ver o resultado.

```
$ 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"  
}
```

Desabilitar a alternância automática de chave

Após habilitar a alternância automática de chaves em uma chave gerenciada pelo cliente, você poderá optar por desabilitá-la a qualquer momento.

Se você desabilitar a alternância automática de chaves, a chave do KMS continuará usando a versão do material de chave que estava usando quando a alternância foi desabilitada. Se você ativar a rotação automática da chave novamente, AWS KMS gira o material da chave com base na nova data de ativação da rotação.

Desabilitar a alternância automática não afeta sua capacidade de [realizar alternâncias sob demanda](#), nem cancela nenhuma alternância sob demanda que esteja em andamento.

Você pode desativar a rotação automática de chaves no AWS KMS console ou usando a [DisableKeyRotation](#) operação. Para desabilitar a alternância automática de chave, você precisará de permissões `kms:DisableKeyRotation`. Para obter mais informações sobre AWS KMS permissões, consulte [Referência de permissões](#) o.

Usando o AWS KMS console

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. (Não é possível habilitar ou desabilitar a alternância de Chaves gerenciadas pela AWS. Elas são alternadas automaticamente a cada ano.)
4. Escolha o alias ou ID de chave de uma chave do KMS.
5. Selecione a guia Key rotation (Rotação de chaves).

A guia Rotação de chaves aparece somente na página de detalhes das chaves KMS de criptografia simétrica com o material de chave AWS KMS gerado (a origem é AWS_KMS), incluindo chaves KMS de criptografia simétrica [multirregional](#).

Não é possível alternar automaticamente chaves do KMS assimétricas, chaves do KMS de HMAC, chaves do KMS com [material de chave importado](#) nem chaves do KMS em [armazenamentos personalizados de chave](#). No entanto, é possível [alterná-las manualmente](#).

6. Na seção Alternância automática de chaves, escolha Editar.
7. Em Alternância de chaves, selecione Desabilitar.

Note

Se uma chave KMS estiver desativada ou pendente de exclusão, AWS KMS ela não rotacionará o material da chave e você não poderá atualizar o status da rotação automática da chave ou o período de rotação. Para atualizar a configuração de alternância automática de chaves, habilite a chave do KMS ou cancele a exclusão. Para obter mais detalhes, consulte [Funcionamento da alternância](#) e [Principais estados das AWS KMS chaves](#).

8. Escolha Salvar.

Usando a AWS KMS API

É possível usar a [API do AWS Key Management Service \(AWS KMS\)](#) para desabilitar a alternância automática de chaves e visualizar o status atual da alternância de qualquer chave gerenciada pelo cliente. Este exemplo usa a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

A [DisableKeyRotation](#) operação desativa a rotação automática da chave. Para identificar a chave do KMS nessa operação, use seu [ID de chave](#) ou [ARN de chave](#). Por padrão, a mudança de chaves está desabilitada nas chaves gerenciadas de cliente.

O exemplo a seguir desativa a rotação automática de chaves na chave KMS de criptografia simétrica especificada e usa a [GetKeyRotationStatus](#) operação para ver o resultado.

```
$ 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
}
```

Executar alternância de chaves sob demanda

Você pode realizar a alternância sob demanda do material de chave nas chaves do KMS gerenciadas pelo cliente, independentemente de a alternância automática de chaves estar ou não habilitada. Desativar a rotação automática ([DisableKeyRotation](#)) não afeta sua capacidade de realizar rotações sob demanda, nem cancela nenhuma rotação sob demanda em andamento. As alternâncias sob demanda não alteram os cronogramas de alternância automática existentes. Por exemplo, considere uma chave do KMS que tenha a alternância automática de chaves habilitada com um período de alternância de 730 dias. Se a chave estiver programada para alternar automaticamente em 14 de abril de 2024 e você realizar uma alternância sob demanda em 10 de abril de 2024, a chave será alternada automaticamente, conforme programado, em 14 de abril de 2024 e a cada 730 dias depois disso.

Você pode realizar a rotação de chaves sob demanda no máximo 25 vezes por chave KMS. Você pode usar o AWS KMS console para visualizar o número de rotações sob demanda restantes disponíveis para uma chave KMS.

A alternância de chaves sob demanda só é compatível com [chaves do KMS de criptografia simétrica](#). [Você não pode realizar a rotação sob demanda de chaves KMS assimétricas, chaves HMAC KMS ou chaves KMS em um armazenamento de chaves personalizado](#). Para realizar a alternância sob demanda de um conjunto de [chaves de várias regiões](#) relacionadas, invoque a alternância sob demanda na chave primária.

Usuários autorizados `kms:RotateKeyOnDemand` e com `kms:GetKeyRotationStatus` permissões podem usar o AWS KMS console e a AWS KMS API para iniciar a rotação de chaves sob demanda e visualizar o status da rotação de chaves. Use [ListKeyRotations](#) para visualizar as rotações concluídas de uma chave KMS.

Tópicos

- [Como iniciar a alternância de chaves sob demanda \(console\)](#)
- [Iniciando a rotação de chaves sob demanda \(API\)](#)[AWS KMS](#)

Como iniciar a alternância de chaves sob demanda (console)

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. (Não é possível realizar a alternância sob demanda de Chaves gerenciadas pela AWS. Elas são alternadas automaticamente a cada ano.)
4. Escolha o alias ou ID de chave de uma chave do KMS.
5. Escolha a guia Material de chave e rodízios.

A guia Material de chave e rodízios aparece somente na página de detalhes das chaves do KMS de criptografia simétrica compatíveis com rodízio automático ou sob demanda. Isso inclui chaves KMS com material de chave AWS KMS gerado (AWS_KMSorigem) e chaves KMS com material de chave importado (origem EXTERNA)

[Você não pode realizar a rotação sob demanda de chaves KMS assimétricas, chaves HMAC KMS ou chaves KMS em armazenamentos de chaves personalizadas.](#) No entanto, é possível [alterná-las manualmente.](#)

6. Escolha Fazer rodízio agora. Para chaves de criptografia simétricas com material de chave importado, a opção Girar agora está disponível somente se você tiver [importado um novo material de chave](#) anteriormente e estiver no estado de rotação pendente.

Note

Para chaves multirregionais, somente a chave de região primária pode ser girada.

7. Leia e considere o aviso e as informações sobre o número restante de alternâncias sob demanda para a chave. Você também verá informações como ID, descrição e prazo de validade do material de chave que serão atualizadas após o rodízio. Se você decidir que não deseja continuar com a alternância sob demanda, escolha Cancelar.
8. Escolha Alternar chave para confirmar a alternância sob demanda.

Note

A rotação sob demanda está sujeita aos mesmos efeitos de consistência eventuais de outras operações AWS KMS de gerenciamento. Pode haver um pequeno atraso

antes que o novo material de chave esteja disponível no AWS KMS. O banner na parte superior do console notificará você quando a alternância sob demanda estiver concluída.

Iniciando a rotação de chaves sob demanda (API)AWS KMS

É possível usar a [API do AWS Key Management Service \(AWS KMS\)](#) para iniciar a alternância de chaves sob demanda e visualizar o status atual da alternância de qualquer chave gerenciada pelo cliente. Este exemplo usa a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

A [RotateKeyOnDemand](#) operação inicia imediatamente a rotação de chaves sob demanda para a chave KMS especificada. Para identificar a chave do KMS nessas operações, use seu [ID de chave](#) ou [ARN de chave](#).

O exemplo a seguir inicia a rotação de chaves sob demanda na chave KMS de criptografia simétrica especificada e usa a [GetKeyRotationStatus](#) operação para verificar se a rotação sob demanda está em andamento. O `OnDemandRotationStartDate` na resposta `kms:GetKeyRotationStatus` identifica a data e a hora em que uma alternância sob demanda em andamento foi iniciada. Neste exemplo, a chave do KMS também tem rodízio automático habilitado com um período de 365 dias.

```
$ aws kms rotate-key-on-demand --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyRotationEnabled": true,
  "NextRotationDate": "2024-03-14T18:14:33.587000+00:00",
  "OnDemandRotationStartDate": "2024-02-24T18:44:48.587000+00:00"
  "RotationPeriodInDays": 365
}
```

Se a chave do KMS não for compatível com rodízio automático ou não tiver rodízio automático habilitado, a resposta `kms:GetKeyRotationStatus` terá menos campos, como mostrado no seguinte exemplo:

```
$ 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"
}
```

Listar rodízios e materiais de chave

As chaves do KMS compatíveis com o rodízio automático ou sob demanda podem ter vários materiais de chave associados a elas. Essas chaves têm um material de chave inicial e um material de chave adicional para cada rodízio automático ou sob demanda.

Usuários autorizados com `kms:ListKeyRotations` permissão podem usar o AWS KMS console e a [ListKeyRotations](#) API para listar todos os principais materiais associados a uma chave KMS, incluindo aqueles de rotações automáticas e sob demanda concluídas.

Tópicos

- [Listar rodízios e materiais de chave \(console\)](#)
- [Listar rotações e principais materiais \(AWS KMS API\)](#)

Listar rodízios e materiais de chave (console)

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Escolha o alias ou ID de chave de uma chave do KMS.
5. Escolha a guia Material de chave e rodízios.
 - A guia Material de chave e rodízios aparece somente na página de detalhes das chaves do KMS de criptografia simétrica compatíveis com rodízio automático ou sob demanda. Isso inclui

chaves KMS com material de chave AWS KMS gerado (AWS_KMSorigem) e chaves KMS com material de chave importado (EXTERNALorigem).

- A tabela Materiais de chave na guia Material de chave e rodízios lista todos os materiais de chave associados à chave do KMS. Para cada material de chave, a entrada correspondente exibe seu identificador exclusivo atribuído pelo AWS KMS, a data de rodízio e o estado do material de chave. A data de rodízio identifica quando o material de chave se tornou atual após um rodízio de chaves automático ou sob demanda. Não há data de rodízio associada ao primeiro material de chave ou ao material de chave com Pending rotation. O estado do material chave determina como AWS KMS usa o material chave. Material de chave atual é usado tanto para criptografia quanto para descriptografia. Material de chave não atual é usado somente para descriptografia. Um estado Pending rotation do material de chave indica que o material de chave está preparado para rodízio. Esse material de chave não é usado para nenhuma operação criptográfica até que um rodízio de chaves sob demanda o torne o material de chave atual. As informações adicionais exibidas para o material de chave dependem do tipo de chave do KMS.
- Para chaves do KMS de criptografia simétrica com a origem AWS_KMS, cada linha também exibe o tipo de rodízio: On-demand ou Automatic.
- As chaves KMS de criptografia simétrica com material de chave importado (EXTERNALorigem) suportam apenas On-demand rotação, portanto, não há coluna do tipo rotação. Em vez disso, cada linha exibe um estado de importação, uma descrição especificada pelo usuário, informações de expiração e um menu Ações. O estado de importação é Importado, indicando que o material da chave está disponível no interior, AWS KMS ou Importação pendente, indicando que o material da chave não está disponível no interior AWS KMS. O menu Ações pode ser usado para excluir material de chave importado ou reimportar material de chave. A ação Excluir material de chave estará desabilitada se o estado de importação do material de chave for Importação pendente. A ação Reimportar material de chave está sempre disponível. Você não precisa esperar a expiração ou exclusão de um material de chave para importá-lo novamente.

Listar rotações e principais materiais (AWS KMS API)

É possível usar a [API do AWS Key Management Service \(AWS KMS\)](#) para iniciar o rodízio de chaves sob demanda e visualizar o status atual do rodízio de qualquer chave gerenciada pelo cliente. Este exemplo usa a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

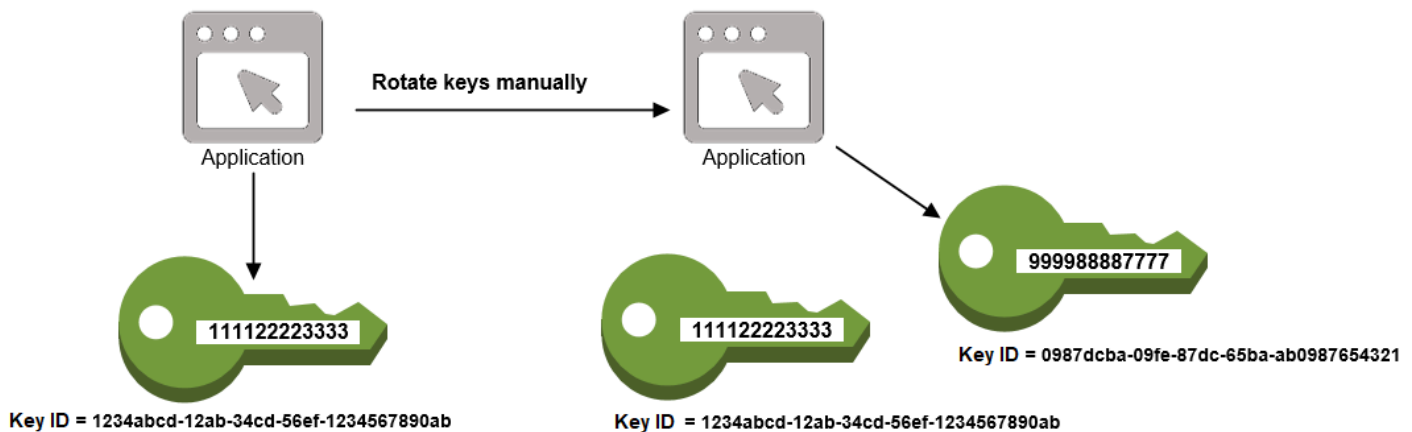
A [ListKeyRotations](#) operação lista todas as rotações e os materiais principais da chave KMS especificada. Para identificar a chave do KMS nessas operações, use seu [ID de chave](#) ou [ARN de chave](#).

Essa operação é compatível com um parâmetro opcional `IncludeKeyMaterial`. O valor padrão desse parâmetro é `ROTATIONS_ONLY`. Se você omitir esse parâmetro, AWS KMS retornará informações sobre os materiais principais criados pela rotação automática ou sob demanda da chave. Quando você especifica um valor de `ALL_KEY_MATERIAL`, o AWS KMS adiciona à resposta o primeiro material de chave e qualquer material de chave importado com rodízio pendente. Esse parâmetro pode ser usado somente com chaves do KMS compatíveis com rodízio de chaves automático ou sob demanda.

```
$ aws kms list-key-rotations --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --include-key-material ALL_KEY_MATERIAL
{
  "Rotations": [
    {
      "KeyId": 1234abcd-12ab-34cd-56ef-1234567890ab,
      "KeyMaterialId":
123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0,
      "KeyMaterialDescription": "KeyMaterialA",
      "ImportState": "PENDING_IMPORT",
      "KeyMaterialState": "NON_CURRENT"
    },
    {
      "KeyId": 1234abcd-12ab-34cd-56ef-1234567890ab,
      "KeyMaterialId":
96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068,
      "ImportState": "IMPORTED",
      "KeyMaterialState": "CURRENT",
      "ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE",
      "RotationDate": "2025-05-01T15:50:51.045000-07:00",
      "RotationType": "ON_DEMAND"
    }
  ],
  "Truncated": false
}
```


Alternar chaves manualmente

Talvez você prefira criar e usar uma nova chave do KMS em vez da chave atual do KMS a habilitar o rodízio de chaves automático. Quando a nova chave do KMS tem material criptográfico diferente daquele da chave do KMS atual, usar a nova chave do KMS tem o mesmo efeito de alterar o material de chave em uma chave do KMS existente. O processo de substituir uma chave do KMS por outra é conhecido como alternância manual de chaves.



A rotação manual é uma boa opção quando você deseja girar chaves KMS que não estão qualificadas para rotação automática ou sob demanda, como chaves KMS assimétricas, chaves HMAC KMS e chaves KMS em lojas de chaves personalizadas.

Note

Ao começar a usar a nova chave KMS, certifique-se de manter a chave KMS original ativada para que ela AWS KMS possa descriptografar os dados criptografados pela chave KMS original.

Ao alternar as chaves do KMS manualmente, você precisa atualizar as referências ao ARN ou ao ID da chave do KMS nas suas aplicações. [Aliases](#), que associam um nome amigável a uma chave do KMS, tornam esse processo mais fácil. Use um alias para fazer referência a uma chave do KMS em suas aplicações. Quando quiser alterar a chave do KMS utilizada pela aplicação, em vez de editar o código da aplicação, altere a chave do KMS de destino do alias. Para obter detalhes, consulte [Saiba como usar aliases em suas aplicações](#).

Note

Os aliases que apontam para a versão mais recente de uma chave KMS girada manualmente são uma boa solução para [DescribeKey](#) operações criptográficas como [Criptografar](#) [DeriveSharedSecret](#),,,, [Assinar](#) [GetPublicKey](#) [GenerateDataKey](#) [GenerateDataKeyPair](#) [Verificar](#). [GenerateMac](#) [VerifyMac](#) Aliases não são permitidos em operações que gerenciam chaves KMS, como [DisableKey](#) ou [ScheduleKeyDeletion](#).

Ao chamar a operação [Decrypt](#) em chaves KMS de criptografia simétrica giradas manualmente, omita o parâmetro do comando. KeyId AWS KMS usa automaticamente a chave KMS que criptografou o texto cifrado.

O KeyId parâmetro é obrigatório ao chamar [Decrypt](#) ou [verificar](#) com uma chave KMS assimétrica ou ao chamar [VerifyMac](#) com uma chave HMAC KMS. Essas solicitações falham quando o valor do parâmetro KeyId é um alias que não aponta mais para a chave do KMS que executou a operação criptográfica, como quando uma chave é alternada manualmente. Para evitar esse erro, você deve rastrear e especificar a chave KMS correta para cada operação.

Para alterar a chave KMS de destino de um alias, use a [UpdateAlias](#) operação na AWS KMS API. Por exemplo, este comando atualiza o alias `alias/TestKey` para apontar para uma nova chave do KMS. Como a operação não retorna nenhuma saída, o exemplo usa a [ListAliases](#) operação para mostrar que o alias agora está associado a uma chave KMS diferente e o `LastUpdatedDate` campo está atualizado. Os `ListAliases` comandos usam o [query parâmetro](#) no AWS CLI para obter somente o `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
    },
  ]
}
```

Alterar a chave primária em um conjunto de chaves de várias regiões

Todos os conjuntos de chaves de várias regiões relacionadas devem ter uma chave primária. Porém, você pode alterar a chave primária. Esta ação, conhecida como atualizar a região primária, converte a chave primária atual em uma chave de réplica e converte uma das chaves de réplica relacionadas na chave primária. Você pode fazer isso quando precisa excluir a chave primária atual enquanto mantém as chaves de réplica ou para localizar a chave primária na mesma região que seus administradores de chave.

Você pode selecionar qualquer chave de réplica relacionada para ser a nova chave primária. Tanto a chave primária quanto a chave de réplica devem estar no [estado de chave](#) Enabled quando a operação começar.

O estado de chave **Updating**

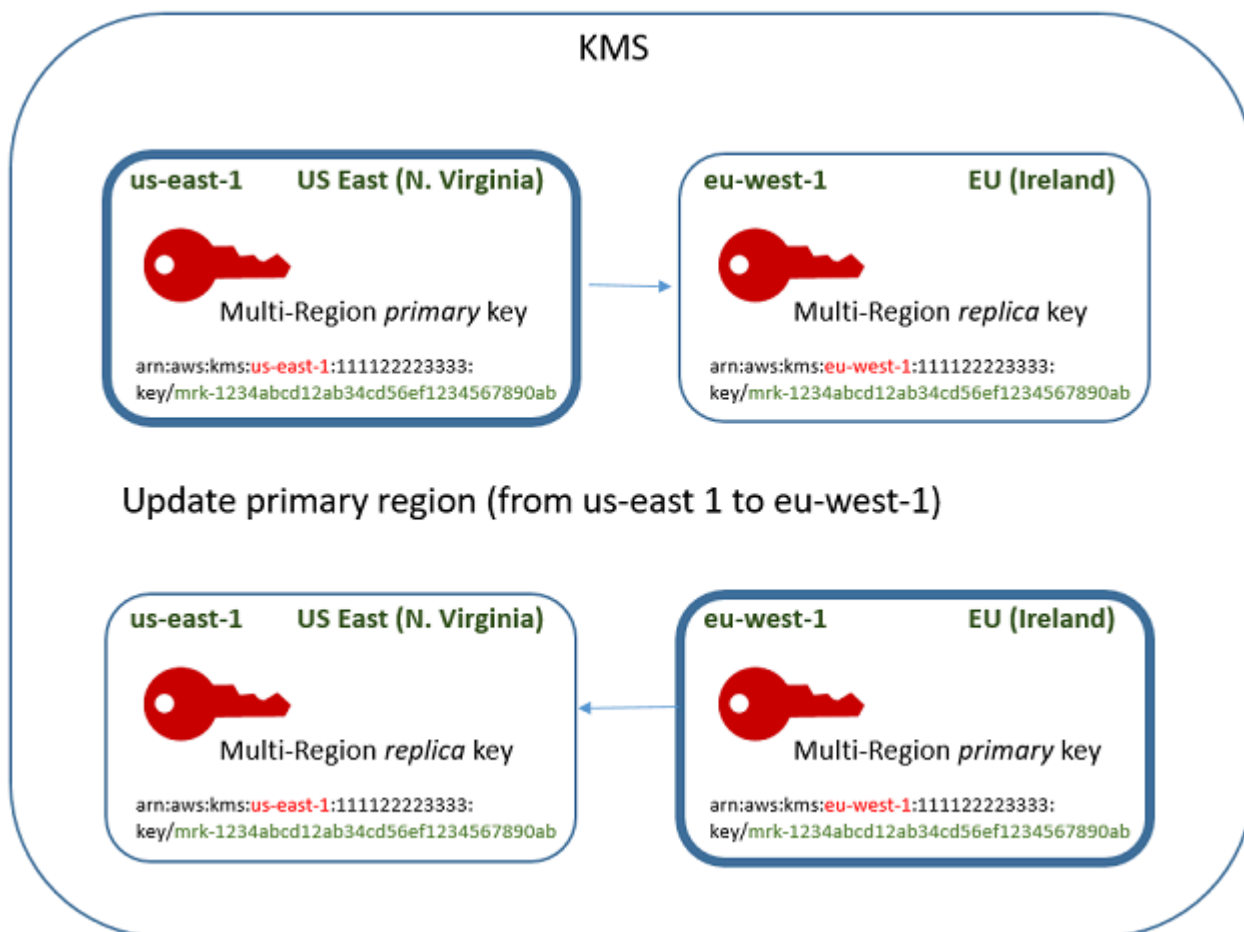
Mesmo após a conclusão da operação UpdatePrimaryRegion, o processo de atualização da região primária pode permanecer em andamento por mais alguns segundos. Durante esse tempo, as chaves primárias antigas e novas têm um estado de chave transitório [Updating](#) (Atualizando). Enquanto o estado de chave é Updating, você pode usar as chaves em operações de criptografia, mas não pode replicar a nova chave primária ou realizar certas operações de gerenciamento, como habilitar ou desabilitar essas chaves. Operações como a

[DescribeKey](#) podem exibir as chaves primárias antigas e novas como réplicas. O estado de chave Enabled será restaurado quando a atualização estiver concluída.

Para obter informações sobre o efeito do estado de chave Updating, consulte [Principais estados das AWS KMS chaves](#).

Como funciona

Suponha que você tenha uma chave primária no Leste dos EUA (Norte da Virgínia) (us-east-1) e uma chave de réplica na Europa (Irlanda) (eu-west-1). É possível usar o recurso de atualização para transformar a chave primária na região Leste dos EUA (Norte da Virgínia) (us-east-1) em uma chave de réplica e transformar a chave de réplica na Europa (Irlanda) (eu-west-1) na chave primária.



Quando o processo de atualização for concluído, a chave de várias regiões na região Europa (Irlanda) (eu-west-1) é uma chave primária de várias regiões, enquanto a chave na região Leste dos EUA (Norte da Virgínia) (us-east-1) é sua chave de réplica. Se houver outras chaves de réplica relacionadas, elas se tornarão réplicas da nova chave primária. Na próxima vez que

AWS KMS sincronizar as propriedades compartilhadas das chaves multirregionais, ele obterá as [propriedades compartilhadas](#) da nova chave primária e as copiará para suas chaves de réplica, incluindo a chave primária anterior.

A operação de atualização não surte efeito no [ARN de chave](#) de chaves de várias regiões. Ela também não surte efeito em propriedades compartilhadas, como material de chave, ou em propriedades independentes, como política de chaves. Porém, talvez você queira [Atualizar a política de chaves](#) da nova chave primária. Por exemplo, talvez você queira adicionar [kms: ReplicateKey](#) permission for trust principals à nova chave primária e removê-la da nova chave de réplica.

Atualizar a região primária

É possível converter uma chave de réplica em uma chave primária, o que transforma a chave primária anterior em uma réplica. Para atualizar a região principal, você precisa da `UpdatePrimaryRegion` permissão [kms:](#) nas duas regiões.

Você pode atualizar a região primária no AWS KMS console ou usando a [UpdatePrimaryRegion](#) operação.

Usando o AWS KMS console

Você pode atualizar a chave primária no AWS KMS console. Comece na página de detalhes de chaves da chave primária atual.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Customer managed keys (Chaves gerenciadas de cliente).
4. Selecione o ID de chave ou alias da [chave primária de várias regiões](#). Isso abre a página de detalhes da chave primária.

Para identificar uma chave primária de várias regiões, use o ícone de ferramenta no canto superior direito para adicionar a coluna Regionality (Regionalidade) à tabela.

5. Escolha a guia Regionality (Regionalidade).
6. Na seção Primary key (Chave primária), escolha Change primary Region (Alterar região primária).

7. Escolha a região da nova chave primária. Você só pode escolher uma região no menu.

O menu Change primary Regions (Alterar regiões primárias) inclui apenas regiões que têm uma chave de várias regiões relacionada. Você pode não ter [permissão para atualizar a região primária](#) em todas as regiões no menu.

8. Escolha Alterar região primária.

Usando a AWS KMS API

Para alterar a chave primária em um conjunto de chaves multirregionais relacionadas, use a [UpdatePrimaryRegion](#) operação.

Usar o parâmetro KeyId para identificar a chave primária atual. Use o PrimaryRegion parâmetro para indicar a Região da AWS da nova chave primária. Se a chave primária ainda não tiver uma réplica na nova região primária, a operação falhará.

O exemplo a seguir altera a chave primária da chave de várias regiões na região us-west-2 para sua réplica na região eu-west-1. O parâmetro KeyId identifica a chave primária atual na região us-west-2. O PrimaryRegion parâmetro especifica a Região da AWS da nova chave primária, 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
```

Quando bem-sucedida, essa operação não retorna saída; apenas o código de status HTTP. Para ver o efeito, chame a [DescribeKey](#) operação em qualquer uma das teclas multirregionais. Talvez você queira esperar até que o estado da chave volte a ser Enabled. Enquanto estado da chave é [Updating](#) (Atualizando), os valores da chave ainda podem estar em fluxo.

Por exemplo, a seguinte chamada DescribeKey obtém os detalhes sobre a chave de várias regiões na região eu-west-1. A saída mostra que a chave de várias regiões na região eu-west-1 é agora a chave primária. A chave de várias regiões relacionada (mesma ID de chave) na região us-west-2 agora é uma chave de réplica.

```
$ 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"
        }
      ]
    }
  }
}

```

Excluir um AWS KMS key

Excluir um AWS KMS key é destrutivo e potencialmente perigoso. Essa ação exclui o material de chave e todos os metadados associados à chave do KMS e é irreversível. Depois que uma chave do KMS é excluída, não é mais possível descriptografar os dados que foram criptografados com ela, o que significa que os dados são irrecuperáveis. (As únicas exceções são chaves de [réplica multirregionais e chaves](#) assimétricas e as chaves do KMS de HMAC com material de chave importado.) Esse risco é significativo para [chaves KMS assimétricas usadas para criptografia](#), nas quais, sem aviso ou erro, os usuários podem continuar gerando textos cifrados com a chave pública que não podem ser descriptografados após a exclusão da chave privada. AWS KMS

Só exclua uma chave do KMS quando você tiver certeza de que não vai mais precisar dela. Caso não tenha certeza, [desabilite a chave do KMS](#) em vez de excluí-la. Você poderá reabilitar a chave do KMS desabilitada e [cancelar a exclusão agendada](#) de uma chave do KMS, mas não poderá recuperar a chave do KMS excluída.

Apenas é possível agendar a exclusão de uma chave gerenciada pelo cliente. Você não pode excluir Chaves gerenciadas pela AWS ou Chaves pertencentes à AWS.

Antes de excluir uma chave KMS, talvez você queira saber quantos textos cifrados foram criptografados sob essa chave KMS. AWS KMS não armazena essas informações e não armazena nenhum dos textos cifrados. Para obter essas informações, você deve determinar o uso anterior de uma chave do KMS. Para obter ajuda, acesse [Determinar a utilização anterior de uma chave do KMS](#).

AWS KMS nunca exclui suas chaves KMS, a menos que você as agende explicitamente para exclusão e o período de espera obrigatório expire.

No entanto, você pode optar por excluir uma chave do KMS devido a um ou mais dos seguintes motivos:

- Para concluir o ciclo de vida das chaves do KMS que não são mais necessárias
- Para evitar a sobrecarga de gerenciamento e os [custos](#) associados à manutenção de chaves do KMS não usadas
- Para reduzir o número de chaves do KMS que contam para a sua [cota de recursos de chaves do KMS](#)

 Note

Se você [fechar o seu Conta da AWS](#), suas chaves KMS ficarão inacessíveis e você não será mais cobrado por elas.

AWS KMS registra uma entrada no seu AWS CloudTrail registro quando você [agenda a exclusão](#) da chave KMS e quando a chave [KMS é realmente excluída](#).

Sobre o período de espera

Como é destrutivo e potencialmente perigoso excluir uma chave KMS, é AWS KMS necessário definir um período de espera de 7 a 30 dias. O período de espera padrão é de 30 dias.

No entanto, o período de espera real pode ser até 24 horas mais longo do que o programado. Para obter a data e a hora reais em que a chave KMS será excluída, use a [DescribeKey](#) operação. Ou, no console do AWS KMS, na [página de detalhes](#) da chave do KMS, na seção General configuration (Configuração geral), consulte a seção Scheduled deletion date (Data de exclusão programada). Certifique-se de anotar o fuso horário.

Durante o período de espera, o status da chave do KMS e o status da chave é Pending deletion (Exclusão pendente).

- Uma chave do KMS com exclusão pendente não pode ser usada em nenhuma [operação criptográfica](#).
- AWS KMS não [gira o material das chaves](#) KMS que estão pendentes de exclusão.

Após o término do período de espera, AWS KMS exclui a chave KMS, seus aliases e todos os metadados relacionados. AWS KMS

Programar a exclusão de uma chave do KMS pode não afetar imediatamente as chaves de dados criptografadas pela chave do KMS. Para obter detalhes, consulte [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#).

Use o período de espera para garantir que não vai precisar da chave do KMS agora nem no futuro. Você pode [configurar um CloudWatch alarme da Amazon](#) para avisá-lo se uma pessoa ou aplicativo tentar usar a chave KMS durante o período de espera. Para recuperar a chave do KMS, basta

cancelar a exclusão de chaves antes do término do período de espera. Após o término do período de espera, você não pode cancelar a exclusão da chave e AWS KMS exclui a chave KMS.

Considerações especiais

Antes de programar suas chaves para exclusão, revise as seguintes considerações especiais sobre a exclusão de chaves do KMS para fins especiais.

Excluir chaves do KMS assimétricas

Usuários [autorizados](#) podem excluir chaves do KMS simétricas ou assimétricas. O procedimento para programar a exclusão dessas chaves do KMS é o mesmo para os dois tipos de chaves. No entanto, como a [chave pública de uma chave KMS assimétrica pode ser baixada](#) e usada fora dela AWS KMS, a operação representa riscos adicionais significativos, especialmente para chaves KMS assimétricas usadas para criptografia (o uso da chave é). ENCRYPT_DECRYPT

- Ao programar a exclusão de uma chave do KMS, o estado dessa chave é alterado para Pending deletion (Exclusão pendente), e a chave do KMS não pode ser usada em [operações de criptografia](#). No entanto, o agendamento da exclusão não tem efeito nas chaves públicas fora do. AWS KMS Os usuários que têm a chave pública podem continuar a usá-la para criptografar mensagens. Eles não recebem nenhuma notificação sobre a alteração do estado da chave. A menos que a exclusão seja cancelada, o texto cifrado criado com a chave pública não pode ser descriptografado.
- Alarmes, logs e outras estratégias que detectam a tentativa de uso da chave do KMS com exclusão pendente não podem detectar o uso da chave pública fora do AWS KMS.
- Quando a chave KMS é excluída, todas as AWS KMS ações envolvendo essa chave KMS falham. No entanto, os usuários que têm a chave pública podem continuar a usá-la para criptografar mensagens. Esses textos cifrados não podem ser descriptografados.

Se você precisar excluir uma chave KMS assimétrica com um uso de chave de ENCRYPT_DECRYPT, use suas entradas de CloudTrail registro para determinar se a chave pública foi baixada e compartilhada. Se for o caso, verifique se a chave pública não está sendo usada fora do AWS KMS. Depois, considere [desabilitar a chave do KMS](#) em vez de excluí-la.

O risco de excluir uma chave do KMS assimétrica é reduzido para chaves do KMS assimétricas com material de chave importado. Para obter detalhes, consulte [Deleting KMS keys with imported key material](#).

Excluir chaves de várias regiões

Para excluir uma chave primária, você deve programar a exclusão de todas as chaves de réplica e, em seguida, aguardar até que as chaves de réplica sejam excluídas. O período de espera necessário para excluir uma chave primária começa quando a última de suas chaves de réplica é excluída. Se você precisar excluir uma chave primária de uma determinada região sem excluir suas chaves de réplica, transforme a chave primária em uma chave de réplica [atualizando a região primária](#).

É possível excluir uma chave de réplica a qualquer momento. Ele não depende do estado da chave de qualquer outra chave do KMS. Se você excluir uma chave de réplica por engano, poderá recriá-la replicando a mesma chave primária na mesma região. A nova chave de réplica que você criar terá as mesmas [propriedades compartilhadas](#) do que a chave de réplica original.

Excluir chaves do KMS com material de chave importado

A exclusão do material de chave de uma chave do KMS com material de chave importado é temporária e reversível. Para restaurar a chave, reimporte o material da chave.

Entretanto, a exclusão de uma chave do KMS é irreversível. Se você [agendar a exclusão da chave](#) e o período de espera necessário expirar, excluirá AWS KMS permanente e irreversivelmente a chave KMS, seu material de chave e todos os metadados associados à chave KMS.

No entanto, o risco e a consequência da exclusão de uma chave do KMS com material de chave importado dependem do tipo (“especificação de chave”) da chave do KMS.

- **Chaves de criptografia simétricas** — Se você excluir uma chave do KMS de criptografia simétrica, todos os textos cifrados restantes criptografados por essa chave serão irrecuperáveis. Não é possível criar uma nova chave do KMS de criptografia simétrica que possa descriptografar os textos cifrados de uma chave de criptografia simétrica excluída com chave do KMS, mesmo que você tenha o mesmo material de chave. Os metadados exclusivos de cada chave do KMS são vinculados criptograficamente a cada texto cifrado simétrico. Esse recurso de segurança garante que somente a chave do KMS que criptografou o texto cifrado simétrico poderá descriptografá-lo, mas impede que você recrie uma chave do KMS equivalente.
- **Chaves assimétricas e HMAC** — Se você tiver o material da chave original, poderá criar uma nova chave KMS com as mesmas propriedades criptográficas de uma chave KMS assimétrica ou HMAC que foi excluída. AWS KMS gera cifrotextos e assinaturas RSA padrão, assinaturas ECC e tags HMAC, que não incluem nenhum recurso de segurança exclusivo. Além disso, é

possível usar uma chave do HMAC ou a chave privada de um par de chaves assimétricas fora da AWS.

Uma nova chave do KMS criada com o mesmo material de chave assimétrica ou HMAC terá um identificador de chave diferente. Você precisará criar uma nova política de chaves, recriar todos os aliases e atualizar as políticas e concessões existentes do IAM para se referir à nova chave.

Excluindo chaves KMS de um repositório de AWS CloudHSM chaves

Quando você agenda a exclusão de uma chave KMS de um armazenamento de AWS CloudHSM chaves, o [estado da chave](#) muda para Exclusão pendente. A chave do KMS permanecerá no estado Pending deletion (Exclusão pendente) durante todo o período de espera, mesmo que ela fique indisponível porque você [desconectou o armazenamento de chaves personalizado](#). Isso permite que você cancele a exclusão da chave do KMS a qualquer momento durante o período de espera.

Quando o período de espera expirar, AWS KMS exclui a chave KMS de. AWS KMS Em seguida, AWS KMS faça o possível para excluir o material chave do AWS CloudHSM cluster associado. Se o AWS KMS não puder excluir o material de chaves, como, por exemplo, quando o armazenamento de chaves é desconectado do AWS KMS, pode ser necessário [excluir manualmente o material de chaves órfãs](#) do cluster.

AWS KMS não exclui o material chave dos backups do cluster. Mesmo que você exclua a chave KMS AWS KMS e exclua o material de chaves do seu AWS CloudHSM cluster, os clusters criados a partir de backups podem conter o material de chave excluído. Para excluir permanentemente o material da chave, use a [DescribeKey](#) operação para identificar a data de criação da chave KMS. Em seguida, [exclua todos os backups do cluster](#) que podem conter o material de chaves.

Quando você agenda a exclusão de uma chave KMS de um armazenamento de chaves, a AWS CloudHSM chave KMS se torna inutilizável imediatamente (sujeita a uma eventual consistência). Contudo, os recursos criptografados com [chaves de dados](#) protegidas pela chave do KMS não serão afetados até que a chave do KMS seja usada novamente, por exemplo, para descriptografar a chave de dados. Esse problema afeta Serviços da AWS muitos dos quais usam chaves de dados para proteger seus recursos. Para obter detalhes, consulte [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#).

Excluir chaves do KMS de um repositório de chaves externo

Excluir uma chave do KMS de um armazenamento de chaves externas não afetará a [chave externa](#) que serviu como material de chave.

Se você agendar a exclusão de uma chave do KMS de um armazenamento de chaves externas, o [estado da chave](#) será alterado para Pending deletion (Exclusão pendente). A chave do KMS permanecerá no estado Pending deletion (Exclusão pendente) durante todo o período de espera, mesmo que ela fique indisponível porque você [desconectou o armazenamento de chaves externas](#). Isso permite que você cancele a exclusão da chave do KMS a qualquer momento durante o período de espera. Quando o período de espera expirar, AWS KMS exclui a chave KMS de. AWS KMS

Quando você agenda a exclusão de uma chave do KMS de um armazenamento de chaves externas, a chave do KMS torna-se inutilizável imediatamente (sujeita a consistência posterior). Contudo, os recursos criptografados com [chaves de dados](#) protegidas pela chave do KMS não serão afetados até que a chave do KMS seja usada novamente, por exemplo, para descriptografar a chave de dados. Esse problema afeta os Serviços da AWS, pois muitos deles usam chaves de dados para proteger recursos. Para obter detalhes, consulte [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#).

Controlar o acesso à exclusão de chaves

Se você usa políticas do IAM para permitir AWS KMS permissões, as identidades do IAM que têm acesso de AWS administrador ("Action": "*") ou acesso AWS KMS total ("Action": "kms:*") já podem agendar e cancelar a exclusão das chaves do KMS. Para permitir que os administradores de chaves programem e cancelem a exclusão de chaves na política de chaves, use o AWS KMS console ou a AWS KMS API.

Normalmente, apenas os administradores de chaves têm permissão para programar ou cancelar a exclusão da chave. Porém, você pode conceder essas permissões a outras identidades do IAM adicionando a permissãokms:ScheduleKeyDeletion e kms:CancelKeyDeletion à política de chaves ou a uma política do IAM. Você também pode usar a chave de [kms:ScheduleKeyDeletionPendingWindowInDays](#) condição para restringir ainda mais os valores que os principais podem especificar no PendingWindowInDays parâmetro de uma [ScheduleKeyDeletionsolicitação](#).

Permitir que administradores de chaves agendem e cancelem a exclusão de chaves

Usando o AWS KMS console

Para conceder aos administradores de chaves permissão para agendar e cancelar a exclusão de chaves.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Escolha o alias ou o ID de chave da chave do KMS cujas permissões você quer alterar.
5. Selecione a guia Key policy (Política de chaves).
6. A próxima etapa difere a visualização padrão da visualização de política de sua política de chaves. A visualização padrão estará disponível somente se você estiver usando a política de chaves padrão do console. Senão, apenas a visualização da política estará disponível.

Quando a visualização padrão está disponível, o botão Switch to policy view (Alternar para visualização de política) ou Switch to default view (Alternar para visualização padrão) é exibido na guia Key policy (Política de chaves).

- Na visualização padrão:
 - Em Key deletion (Exclusão de chaves), escolha Allow key administrators to delete this key (Permitir que administradores de chaves excluam esta chave).
- Na visualização de política:
 - a. Escolha Editar.
 - b. Na declaração de política para administradores de chave, adicione as permissões `kms:ScheduleKeyDeletion` e `kms:CancelKeyDeletion` ao elemento Action.

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
```

```

    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}

```

- c. Escolha Salvar alterações.

Usando a AWS KMS API

Você pode usar o AWS Command Line Interface para adicionar permissões para agendar e cancelar a exclusão da chave.

Para adicionar permissão para programar e cancelar a exclusão de chaves

1. Use o comando [aws kms get-key-policy](#) para recuperar a política de chaves existente e, em seguida, salve o documento de política em um arquivo.
2. Abra o documento de política no editor de texto de sua preferência. Na declaração de política para administradores de chave, adicione as permissões `kms:ScheduleKeyDeletion` e `kms:CancelKeyDeletion`. O exemplo a seguir mostra uma declaração de política com essas duas permissões:

```

{
  "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. Use o comando [aws kms put-key-policy](#) para aplicar a política de chaves à chave do KMS.

Agendar exclusão de chave

Os procedimentos a seguir descrevem como programar a exclusão da chave e cancelar a exclusão da chave AWS KMS keys (chaves KMS) AWS KMS usando a Console de gerenciamento da AWS e a API. AWS KMS

Warning

Excluir uma chave do KMS é um processo destrutivo e potencialmente perigoso. Prosiga somente quando você tiver certeza de que não vai precisar mais usar a chave do KMS futuramente. Caso não tenha certeza, [desabilite a chave do KMS](#) em vez de excluí-la.

Para excluir uma chave do KMS, é preciso ter a respectiva permissão. Para obter informações sobre como conceder essas permissões aos administradores de chaves, consulte [Controlar o acesso à exclusão de chaves](#). Também é possível usar a chave de condição [kms:ScheduleKeyDeletionPendingWindowInDays](#) para restringir ainda mais o período de espera, como impor um período mínimo de espera.

AWS KMS registra uma entrada no seu AWS CloudTrail registro quando você [agenda a exclusão](#) da chave KMS e quando a chave [KMS é realmente excluída](#).

Usando o AWS KMS console

No Console de gerenciamento da AWS, você pode programar e cancelar a exclusão de várias chaves KMS ao mesmo tempo.

Para programar a exclusão de chaves

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.

Não é possível agendar a exclusão de [Chaves gerenciadas pela AWS](#) ou [Chaves pertencentes à AWS](#).

4. Escolha a caixa de seleção ao lado da chave do KMS que você deseja excluir.
5. Escolha Key actions (Ações de chave), Schedule key deletion (Programar exclusão da chave).
6. Leia e considere o aviso e as informações sobre o cancelamento e a exclusão durante o período de espera. Se você decidir cancelar a exclusão, no final da página, selecione Cancel (Cancelar).
7. Para Waiting period (in days) (Período de espera (em dias)), digite um número de dias entre 7 e 30.
8. Revise as chaves do KMS que você está excluindo.
9. Marque a caixa de seleção ao lado de Confirmar que você deseja agendar essa chave para exclusão em **<number of days>** dias. .
10. Escolha Schedule deletion.

O status da chave muda para Pending deletion (Exclusão pendente).

Usando a AWS KMS API

Use o comando [aws kms schedule-key-deletion](#) para agendar a exclusão de chaves de uma [chave gerenciada pelo cliente](#), conforme mostrado no exemplo a seguir.

Você não pode programar a exclusão de um Chave gerenciada pela AWS ou Chave pertencente à AWS.

```
$ aws kms schedule-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --  
pending-window-in-days 10
```

Quando usado com sucesso, ele AWS CLI retorna uma saída como a mostrada no exemplo a seguir:

```
{
```

```
"KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
"DeletionDate": 1598304792.0,
"KeyState": "PendingDeletion",
"PendingWindowInDays": 10
}
```

Cancelar a exclusão de chaves

Após [agendar a exclusão de uma chave do KMS](#), você pode cancelar a exclusão da chave enquanto ela ainda estiver no estado de [exclusão pendente](#). Você pode cancelar a exclusão da chave no AWS KMS console ou usando a [CancelKeyDeletion](#) operação. Depois de cancelar a exclusão pendente de uma chave do KMS, o estado da chave do KMS será Disabled. Para mais informações sobre como habilitar a chave do KMS, consulte [Habilitar e desabilitar chaves](#).

Usando o AWS KMS console

Para cancelar a exclusão de chaves

1. Abra o AWS KMS console em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Marque a caixa de seleção ao lado da chave do KMS que você deseja recuperar.
5. Escolha Key actions (Ações de chave), Cancel key deletion (Cancelar exclusão da chave).

O status da chave muda de Pending deletion (Exclusão pendente) para Disabled (Desabilitada). Para usar a chave, você deve [habilitá-la](#).

Usando a AWS KMS API

Use o [aws kms cancel-key-deletion](#) comando para cancelar a exclusão da chave do AWS CLI, conforme mostrado no exemplo a seguir.

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

Quando usado com sucesso, ele AWS CLI retorna uma saída como a mostrada no exemplo a seguir:

```
{
```

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

O status da chave do KMS muda de Pending Deletion (Exclusão pendente) para Disabled (Desabilitada). Para usar a chave, você deve [habilitá-la](#).

Criar um alarme que detecte o uso de uma chave do KMS com exclusão pendente

Você pode combinar os recursos do AWS CloudTrail Amazon CloudWatch Logs e do Amazon Simple Notification Service (Amazon SNS) para criar um alarme da CloudWatch Amazon que notifica você quando alguém em sua conta tenta usar uma chave KMS que está pendente de exclusão. Se você receber essa notificação, convém cancelar a exclusão da chave do KMS e reconsiderar sua decisão de excluí-la.

Os procedimentos a seguir criam um alarme que notifica você sempre que a mensagem de erro `Key ARN is pending deletion` for gravada em seus arquivos de CloudTrail log. Essa mensagem de erro indica que uma pessoa ou aplicação tentou usar a chave do KMS em uma [operação criptográfica](#). Como a notificação está vinculada à mensagem de erro, ela não é acionada quando você usa operações da API que são permitidas em chaves do KMS com exclusão pendente, como `ListKeys`, `CancelKeyDeletion` e `PutKeyPolicy`. Para ver uma lista das operações de AWS KMS API que retornam essa mensagem de erro, consulte [Principais estados das AWS KMS chaves](#).

O e-mail de notificação recebido não indicará a chave do KMS ou a operação criptográfica. É possível encontrar essas informações em [seu log do CloudTrail](#). Em vez disso, o e-mail informa que o estado do alarme mudou de OK para Alarme. Para obter mais informações sobre CloudWatch alarmes e mudanças de estado, consulte [Usando CloudWatch alarmes da Amazon no Guia CloudWatch](#) do usuário da Amazon.

Warning

Esse CloudWatch alarme da Amazon não pode detectar o uso da chave pública de uma chave KMS assimétrica fora do. AWS KMS Para obter detalhes sobre os riscos especiais de exclusão de chaves do KMS assimétricas usadas para a criptografia de chave pública,

incluindo a criação de textos cifrados que não podem ser descriptografados, consulte [Deleting asymmetric KMS keys](#).

Neste procedimento, você cria um filtro métrico de grupo de CloudWatch registros que localiza instâncias da exceção de exclusão pendente. Em seguida, você cria um CloudWatch alarme com base na métrica do grupo de registros. Para obter informações sobre filtros métricos de grupos de registros, consulte [Criação de métricas a partir de eventos de log usando filtros](#) no Guia do usuário do Amazon CloudWatch Logs.

1. Crie um filtro CloudWatch métrico que analise os CloudTrail registros.

Siga as instruções em [Criar um filtro de métrica para um grupo de logs](#) usando os seguintes valores obrigatórios. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Padrão de filtro	<code>{ \$.eventSource = kms* && \$.errorMessage = "* is pending deletion." }</code>
Valor da métrica	1

2. Crie um CloudWatch alarme com base no filtro métrico que você criou na Etapa 1.

Siga as instruções em [Criar um CloudWatch alarme com base em um filtro métrico de grupo de registros](#) usando os seguintes valores obrigatórios. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Filtro de métrica	O nome do filtro de métrica que você criou na Etapa 1.
Tipo de limite	Estático
Condições	Sempre que <i>metric-name</i> é maior/igual a 1

Campo	Valor
Pontos de dados para o alarme	1 de 1
Tratamento de dados ausentes	Treat missing data as good (not breaching threshold) (Tratar dados ausentes como bons [sem violar o limite])

Depois de concluir esse procedimento, você receberá uma notificação sempre que o novo CloudWatch alarme entrar no ALARM estado. Se você receber uma notificação desse alarme, talvez isso signifique que uma chave do KMS programada para exclusão ainda seja necessária para criptografar ou descriptografar dados. Nesse caso, [cancele a exclusão da chave do KMS](#) e reconsidere sua decisão de excluí-la.

Determinar a utilização anterior de uma chave do KMS

Antes de excluir uma chave KMS, talvez você queira saber quantos textos cifrados foram criptografados sob essa chave. AWS KMS não armazena essas informações e não armazena nenhum dos textos cifrados. Saber como uma chave do KMS foi utilizada no passado pode ajudar você a decidir se ela será ou não necessária no futuro. Este tópico sugere várias estratégias que podem ajudar você a determinar o uso anterior de uma chave do KMS.

Warning

Essas estratégias para determinar o uso passado e real são eficazes somente para AWS usuários e AWS KMS operações. Elas não podem detectar o uso da chave pública de uma chave do KMS assimétrica fora do AWS KMS. Para obter detalhes sobre os riscos especiais de exclusão de chaves do KMS assimétricas usadas para a criptografia de chave pública, incluindo a criação de textos cifrados que não podem ser descriptografados, consulte [Deleting asymmetric KMS keys](#).

Tópicos

- [Examinar as permissões da chave do KMS para determinar o escopo da utilização em potencial](#)

- [Examine AWS CloudTrail os registros para determinar o uso real](#)

Examinar as permissões da chave do KMS para determinar o escopo da utilização em potencial

Determinar quem ou o quê tem acesso no momento a uma chave do KMS pode ajudar a determinar o quão amplamente ela foi usada e se ainda é necessária. Para saber como determinar quem ou o quê tem acesso no momento a uma chave do KMS, acesse [Determinando o acesso ao AWS KMS keys](#).

Examine AWS CloudTrail os registros para determinar o uso real

Talvez seja possível usar o histórico de uso de uma chave do KMS para ajudar a determinar se você tem textos cifrados em uma chave do KMS específica.

Todas as atividades AWS KMS da API são registradas em arquivos de AWS CloudTrail log. Se você [criou uma CloudTrail trilha](#) na região em que sua chave KMS está localizada, você pode examinar seus arquivos de CloudTrail log para ver um histórico de todas as atividades de AWS KMS API de uma chave KMS específica. Se você não tiver uma trilha, ainda poderá ver eventos recentes no [Histórico de eventos do CloudTrail](#). Para obter detalhes sobre como AWS KMS usar CloudTrail, consulte [Registrando chamadas de AWS KMS API com AWS CloudTrail](#).

Os exemplos a seguir mostram entradas de CloudTrail registro que são geradas quando uma chave KMS é usada para proteger um objeto armazenado no Amazon Simple Storage Service (Amazon S3). Neste exemplo, o objeto é carregado no Amazon S3 usando [Proteção de dados usando criptografia no lado do servidor com chaves do KMS \(SSE-KMS\)](#). Ao carregar um objeto no Amazon S3 com SSE-KMS, você especifica a chave do KMS a ser usada para proteger esse objeto. O Amazon S3 usa a AWS KMS [GenerateDataKey](#) operação para solicitar uma chave de dados exclusiva para o objeto, e esse evento de solicitação é registrado CloudTrail com uma entrada semelhante à seguinte:

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AR0ACKCEVSQ6C2EXAMPLE:example-user",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admins/example-user",
    "accountId": "111122223333",
```

```

    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-09-10T23:12:48Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "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"
}

```

Posteriormente, quando você baixa esse objeto do Amazon S3, o Amazon S3 envia Decrypt uma solicitação AWS KMS para descriptografar a chave de dados do objeto usando a chave KMS

especificada. Ao fazer isso, seus arquivos de CloudTrail log incluem uma entrada semelhante à seguinte:

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE: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"  
}
```

Todas as atividades AWS KMS da API são registradas por CloudTrail. Ao avaliar essas entradas de log, talvez você possa determinar a utilização anterior de uma chave do KMS específica, e isso pode ajudar a determinar se você deseja ou não excluí-la.

Para ver mais exemplos de como a atividade da AWS KMS API aparece em seus arquivos de CloudTrail log, acesse [Registrando chamadas de AWS KMS API com AWS CloudTrail](#). Para obter mais informações sobre, CloudTrail acesse o [Guia AWS CloudTrail do usuário](#).

Excluir material de chave importado

Você pode excluir o material de chave importado de uma chave do KMS a qualquer momento. Além disso, quando o material de chave importado com uma data de validade expirar, AWS KMS exclui o material de chave. Em ambos os casos, quando o material da chave é excluído, o [estado da chave](#) do KMS muda para Importação pendente, e a chave do KMS não pode ser usada em nenhuma operação criptográfica.

As chaves de criptografia simétricas podem ter vários materiais de chave associados a elas. Para essas chaves, o KMS atribui um identificador exclusivo a cada material de chave. Você pode usar a [ListKeyRotations](#) API para visualizar esses identificadores de material-chave e o estado correspondente do material-chave (consulte [RotationsListEntry](#)). Um estado do material chave PENDING_ROTATION ou PENDING_MULTI_REGION_IMPORT_AND_ROTATION indica que o material chave não está associado permanentemente à chave KMS. A exclusão ou expiração de qualquer material de chave associado permanentemente altera o estado da chave para Importação pendente. Você pode excluir um material de chave específico especificando seu identificador usando o `key-material-id` parâmetro na [DeleteImportedKeyMaterial](#) API.

Considerações sobre chaves multirregionais

- Ao excluir o material principal de uma chave de região primária que está em PENDING_ROTATION ou PENDING_MULTI_REGION_IMPORT_AND_ROTATION estado, você também excluirá os materiais principais das réplicas das chaves de região.
- Se você excluir o material da chave em uma chave de região primária ou de réplica, somente essa chave específica será afetada e outras chaves multirregionais relacionadas permanecerão inalteradas. Qualquer chave de região primária ou de réplica que tenha todos os seus materiais de chave permanentemente associados continua sendo utilizável em operações criptográficas.

 Warning

O parâmetro `key-material-id` é opcional e, se não for especificado, o AWS KMS excluirá o material de chave atual.

Além de desativar a chave do KMS e retirar as permissões, a exclusão do material da chave pode ser usada como uma estratégia para interromper o uso da chave do KMS de forma rápida, mas temporária. Por outro lado, programar a exclusão de uma chave do KMS com material de chave importado também interrompe rapidamente o uso da chave do KMS. Porém, se a exclusão não for cancelada durante o período de espera, a chave do KMS, os materiais de chave associados e todos os metadados da chave serão excluídos permanentemente. Para obter detalhes, consulte [Deleting KMS keys with imported key material](#).

Para excluir o material chave, você pode usar o AWS KMS console ou a operação [DeleteImportedKeyMaterial](#) da API. AWS KMS registra uma entrada em seu AWS CloudTrail registro quando você [exclui material de chave importado](#) e quando [AWS KMS exclui material de chave expirado](#).

Como a exclusão do material essencial afeta os serviços AWS

Quando você exclui qualquer material da chave, a chave do KMS torna-se inutilizável imediatamente (sujeita a consistência posterior). Porém, os recursos criptografados com [chaves de dados](#) protegidas pela chave do KMS não serão afetados até que a chave do KMS seja usada novamente, por exemplo, para descriptografar a chave de dados. Esse problema afeta Serviços da AWS muitos dos quais usam chaves de dados para proteger seus recursos. Para obter detalhes, consulte [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#).

Usando o AWS KMS console

Você pode usar o AWS KMS console para excluir o material principal.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Execute um destes procedimentos:

- Marque a caixa de seleção de uma chave do KMS com material de chave importado. Escolha Key actions (Ações de chave), Delete key material (Excluir material de chaves). Para chaves de criptografia simétrica que têm vários materiais de chave associados a elas, isso excluirá o material de chave atual.
 - Para chaves KMS de criptografia simétrica com material de chave importado, escolha o alias ou ID da chave de uma chave KMS. Escolha a guia Material de chave e rodízios. A tabela de materiais de chave listará todos os materiais de chave associados à chave. Escolha Excluir material de chave no menu Ações na linha correspondente ao material de chave que você deseja excluir.
5. Confirme que você deseja excluir o material de chaves e selecione Delete key material (Excluir material de chaves). O status da chave do KMS, que corresponde ao seu [estado de chave](#), muda para Pending import (Importação pendente). Se o material de chave excluído estiver no estado PENDING_ROTATION, não haverá alteração no status da chave do KMS.

Usando a AWS KMS API

Para usar a [AWS KMS API](#) para excluir material chave, envie uma [DeleteImportedKeyMaterial](#) solicitação. O exemplo a seguir mostra como fazer isso com a [AWS CLI](#).

Substitua **1234abcd-12ab-34cd-56ef-1234567890ab** pelo ID da chave do KMS cujo material de chave você quer excluir. É possível usar o ID de chave ou o ARN da chave do KMS, mas não é possível usar um alias para essa operação. O comando a seguir exclui o material da chave atual, que talvez seja o único material da chave associado à chave.

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

Para excluir um determinado material de chave, especifique o material de chave identificado usando o parâmetro key-material-id. Substitua **123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0** pelo identificador da chave cujo material de chave você deseja excluir.

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --key-material-id 123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0
```

Gerar chaves de dados

Chaves de dados são chaves simétricas que você pode usar para criptografar dados, inclusive grandes quantidades de dados e outras chaves de criptografia dos dados. Ao contrário de chaves do KMS assimétricas, que não podem ser baixadas, as chaves de dados são devolvidas para você para uso fora do AWS KMS.

Quando AWS KMS gera chaves de dados, ele retorna uma chave de dados em texto simples para uso imediato (opcional) e uma cópia criptografada da chave de dados que você pode armazenar com segurança com os dados. Quando estiver pronto para descriptografar os dados, primeiro solicite a descriptografia da chave AWS KMS de dados criptografada.

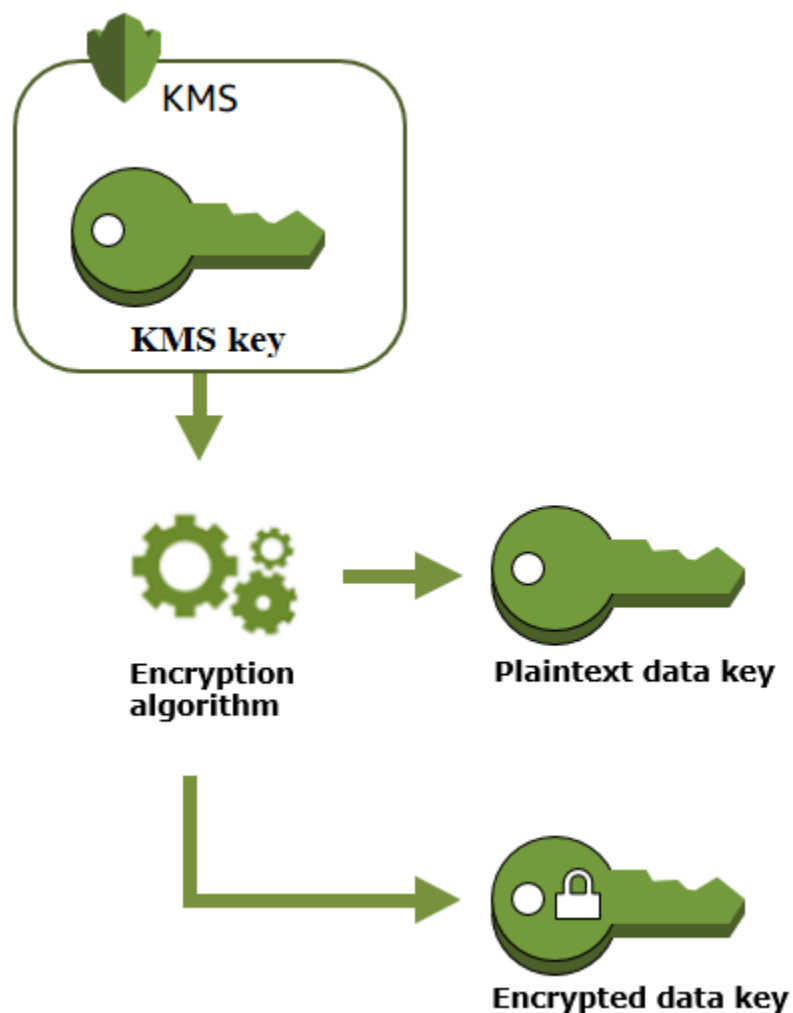
AWS KMS gera, criptografa e descriptografa chaves de dados. No entanto, AWS KMS não armazena, gerencia nem rastreia suas chaves de dados nem executa operações criptográficas com chaves de dados. Você deve usar e gerenciar chaves de dados fora do AWS KMS. Para obter ajuda para usar as chaves de dados de forma segura, consulte a [AWS Encryption SDK](#).

Tópicos

- [Criação de uma chave de dados](#)
- [Funcionamento das operações criptográficas com chaves de dados](#)
- [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#)

Criação de uma chave de dados

Para criar uma chave de dados, chame a [GenerateDataKey](#) operação. AWS KMS gera a chave de dados. Em seguida, ele criptografa uma cópia da chave de dados em uma [chave do KMS de criptografia simétrica](#) especificada por você. A operação retorna uma cópia em texto simples da chave de dados e a cópia da chave de dados criptografada de acordo com a chave do KMS. A imagem a seguir mostra essa operação.



AWS KMS também suporta a [GenerateDataKeyWithoutPlaintext](#) operação, que retorna somente uma chave de dados criptografada. Quando precisar usar a chave de dados, peça AWS KMS para [descriptografá-la](#).

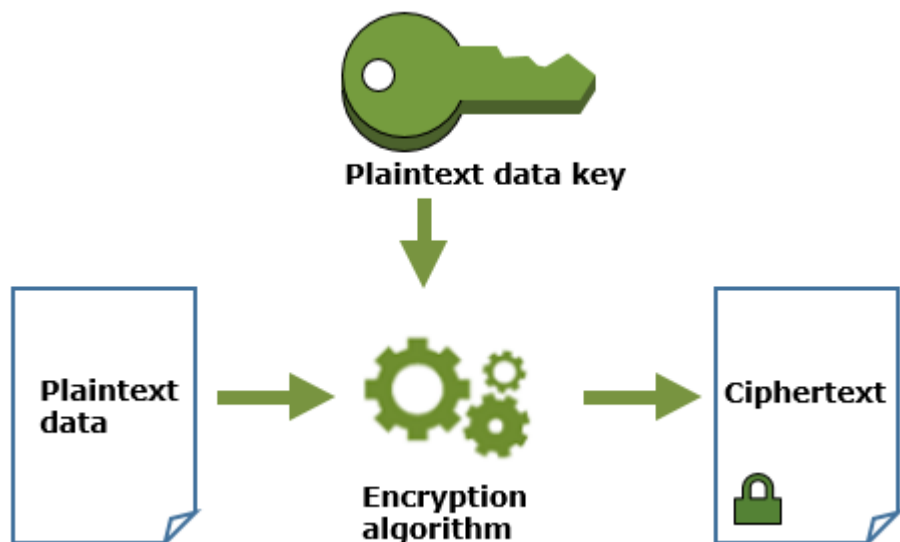
Funcionamento das operações criptográficas com chaves de dados

Os tópicos a seguir explicam como as chaves de dados geradas por uma [GenerateDataKeyWithoutPlaintext](#) operação [GenerateDataKey](#) ou funcionam.

Criptografia de dados com uma chave de dados

AWS KMS não é possível usar uma chave de dados para criptografar dados. Mas você pode usar a chave de dados fora da AWS KMS, por exemplo, usando o OpenSSL ou uma biblioteca criptográfica como o [AWS Encryption SDK](#)

Depois de usar a chave de dados de texto simples para criptografar dados e removê-la da memória assim que possível. Você pode armazenar seguramente as chaves de dados criptografadas com os dados criptografados para que estejam disponíveis para descriptografar os dados.

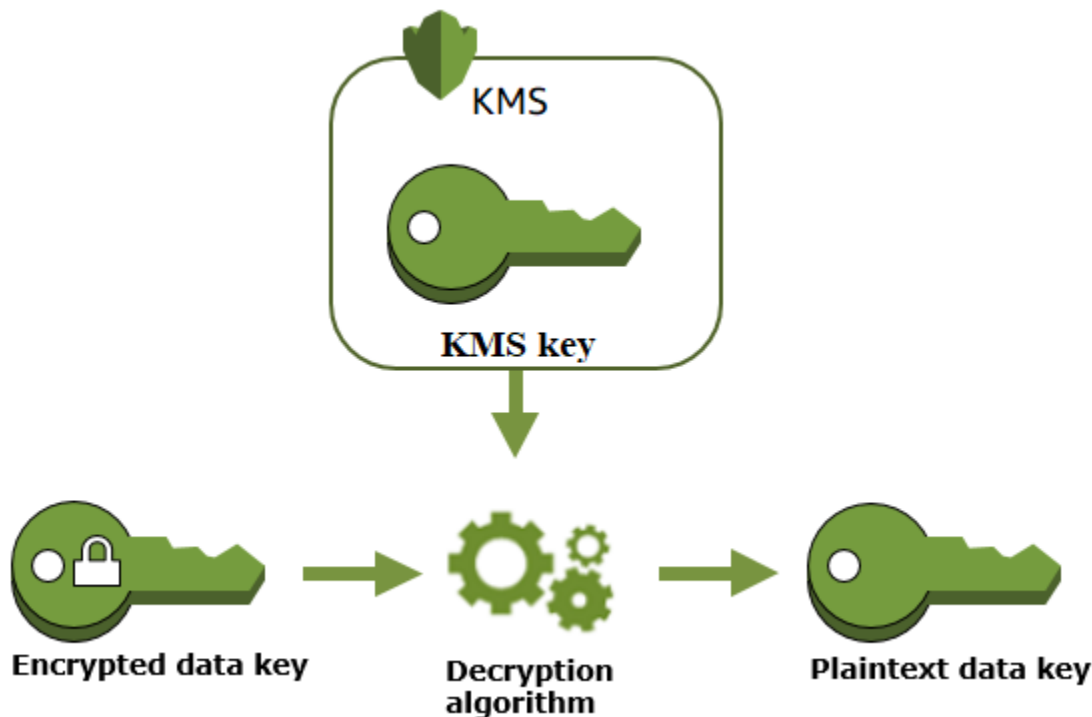


Descriptografia dos dados com uma chave de dados

[Para descriptografar seus dados, passe a chave de dados criptografada para a operação Decrypt.](#)

AWS KMS usa sua chave KMS para descriptografar a chave de dados e, em seguida, retorna a chave de dados em texto simples. Use a chave de dados de texto simples para descriptografar seus dados e removê-la da memória assim que possível.

O diagrama a seguir mostra como usar a operação `Decrypt` para descriptografar uma chave de dados criptografada.



Como as chaves do KMS inutilizáveis afetam as chaves de dados

Quando uma chave do KMS torna-se inutilizável, o efeito é quase imediato (sujeito a consistência posterior). O [estado de chave](#) da chave do KMS é alterado para refletir sua nova condição, e todas as solicitações para usar a chave do KMS em [operações de criptografia](#) falham.

No entanto, o efeito nas chaves de dados criptografadas pela chave do KMS e nos dados criptografados pela chave de dados é adiado até que a chave do KMS seja usada novamente, por exemplo, para descriptografar a chave de dados.

As chaves do KMS podem se tornar inutilizáveis por vários motivos, inclusive pelas ações a seguir que você pode executar.

- [Desabilitar a chave do KMS](#)
- [Agendar a exclusão da chave do KMS](#)
- [Excluir o material de chaves](#) de uma chave do KMS com material de chaves importado ou permitir que o material de chaves importado expire. Se uma chave do KMS com a origem EXTERNAL tiver vários materiais de chave associados, a exclusão ou expiração de qualquer material de chave fará com que a chave se torne inutilizável.
- [Desconectar o armazenamento de AWS CloudHSM chaves](#) que hospeda a chave KMS ou [excluir a chave do AWS CloudHSM cluster](#) que serve como material de chave para a chave KMS.

- [Desconectar o armazenamento de chaves externas](#) que hospeda a chave do KMS ou qualquer outra ação que interfira nas solicitações de criptografia e descriptografia do proxy de armazenamento de chaves externas, inclusive excluir a chave externa do gerenciador de chaves externas.

Esse efeito é particularmente importante para muitos Serviços da AWS que usam chaves de dados para proteger os recursos que o serviço gerencia. O exemplo a seguir usa o Amazon Elastic Block Store (Amazon EBS) e o Amazon Elastic Compute Cloud EC2 (Amazon). Diferentes Serviços da AWS usam chaves de dados de maneiras diferentes. Para obter detalhes, consulte a seção de proteção de dados do capítulo de segurança do AWS service (Serviço da AWS).

Por exemplo, considere este cenário:

1. [Crie um volume do EBS criptografado](#) e especifique uma chave do KMS para protegê-lo. O Amazon EBS solicita que o AWS KMS use a chave do KMS para [gerar uma chave de dados criptografada](#) para o volume. O Amazon EBS armazena a chave de dados criptografada com os metadados do volume.
2. Quando você anexa o volume do EBS a uma EC2 instância, a Amazon EC2 usa sua chave KMS para descriptografar a chave de dados criptografada do volume do EBS. A Amazon EC2 usa a chave de dados no hardware Nitro, que é responsável por criptografar todo o disco no volume I/O do EBS. A chave de dados persiste no hardware Nitro enquanto o volume do EBS está conectado à instância. EC2
3. Você executa uma ação que torna a chave do KMS inutilizável. Isso não tem efeito imediato na EC2 instância ou no volume do EBS. A Amazon EC2 usa a chave de dados, não a chave KMS, para criptografar todo o disco I/O enquanto o volume está conectado à instância.
4. No entanto, quando o volume criptografado do EBS é separado da EC2 instância, o Amazon EBS remove a chave de dados do hardware Nitro. Na próxima vez que o volume criptografado do EBS for anexado a uma EC2 instância, o anexo falhará, porque o Amazon EBS não pode usar a chave KMS para descriptografar a chave de dados criptografada do volume. Para usar o volume do EBS novamente, é necessário tornar a chave do KMS utilizável novamente.

Gerar pares de chaves de dados

Uma chave do KMS assimétrica representa um par de chaves de dados. Pares de chaves de dados são chaves de dados assimétricas que consistem em uma chave pública e uma chave privada matematicamente relacionadas. Eles são projetados para uso em criptografia e decodificação do lado do cliente, assinatura e verificação externas ou para estabelecer um segredo compartilhado entre dois pares. AWS KMS

Ao contrário dos pares de chaves de dados que ferramentas como o OpenSSL geram AWS KMS , protege a chave privada em cada par de chaves de dados sob uma chave KMS de criptografia simétrica especificada por você. AWS KMS No entanto, AWS KMS não armazena, gerencia nem rastreia seus pares de chaves de dados nem executa operações criptográficas com pares de chaves de dados. É necessário usar e gerenciar pares de chaves fora do AWS KMS.

Tópicos

- [Criar um par de chaves de dados](#)
- [Funcionamento das operações criptográficas com pares de chaves de dados](#)

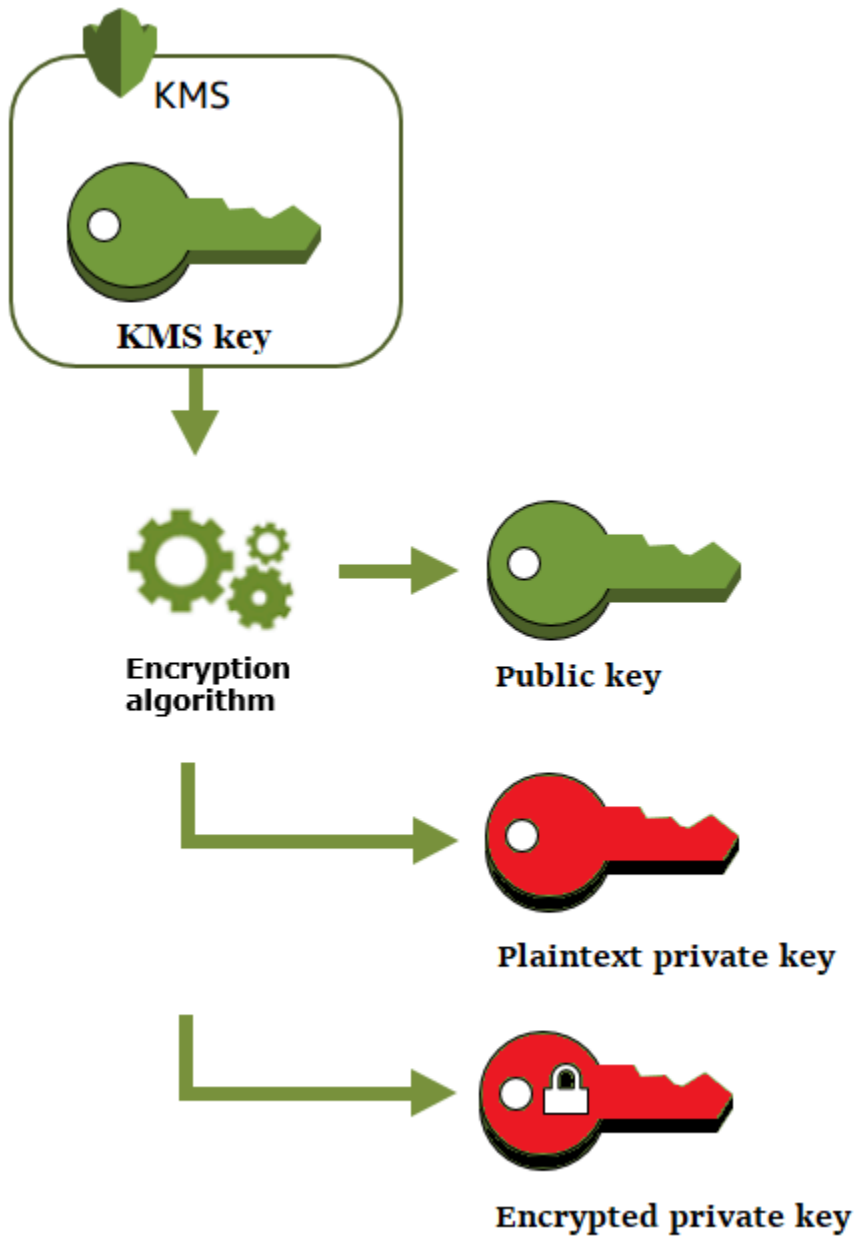
Criar um par de chaves de dados

Para criar um par de chaves de dados, chame as [GenerateDataKeyPairWithoutPlaintext](#) operações [GenerateDataKeyPair](#) ou. Especifique a [chave do KMS de criptografia simétrica](#) que você deseja usar para criptografar a chave privada.

`GenerateDataKeyPair` retorna uma chave pública de texto não criptografado, uma chave privada de texto não criptografado e uma chave privada criptografada. use essa operação quando precisar imediatamente de uma chave privada de texto não criptografado, como para gerar uma assinatura digital.

`GenerateDataKeyPairWithoutPlaintext` retorna uma chave pública de texto não criptografado e uma chave privada criptografada, mas não uma chave privada de texto não criptografado. Use essa operação quando não precisar imediatamente de uma chave privada de texto não criptografado, como quando você está criptografando uma chave pública. Posteriormente, quando precisar de uma chave privada de texto não criptografado para descriptografar os dados, poderá chamar a operação [Decrypt](#).

A imagem a seguir mostra a operação `GenerateDataKeyPair`. A operação `GenerateDataKeyPairWithoutPlaintext` omite a chave privada de texto não criptografado.



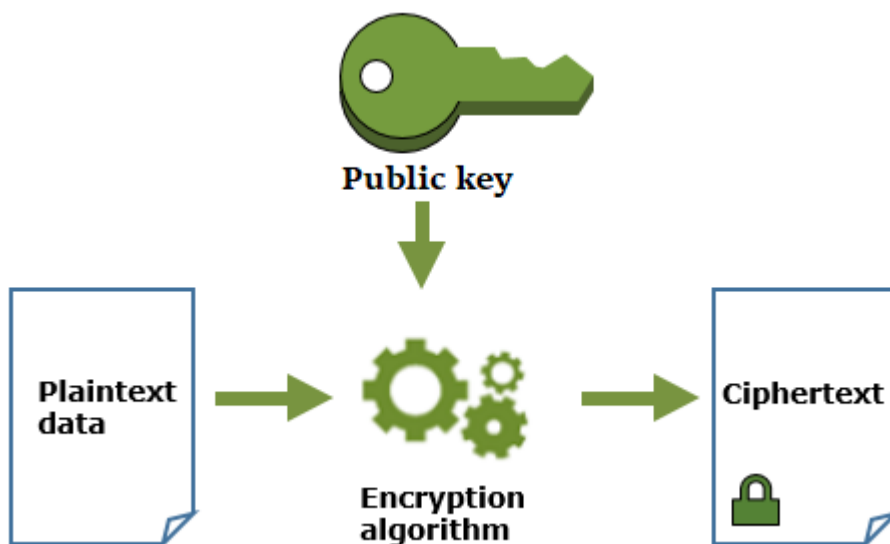
Funcionamento das operações criptográficas com pares de chaves de dados

Os tópicos a seguir explicam quais operações criptográficas você pode realizar com pares de chaves de dados gerados por uma [GenerateDataKeyPairWithoutPlaintext](#) operação [GenerateDataKeyPair](#) e como elas funcionam.

Criptografar dados com um par de chaves de dados

Ao criptografar com um par de chaves de dados, você usa a chave pública do par para criptografar os dados e a chave privada do mesmo par para descriptografar os dados. Normalmente, pares de chaves de dados são usados quando muitos usuários precisam criptografar dados que só o usuário que tem a chave privada pode descriptografar.

As partes com a chave pública usam essa chave para criptografar dados, conforme mostrado no diagrama a seguir.

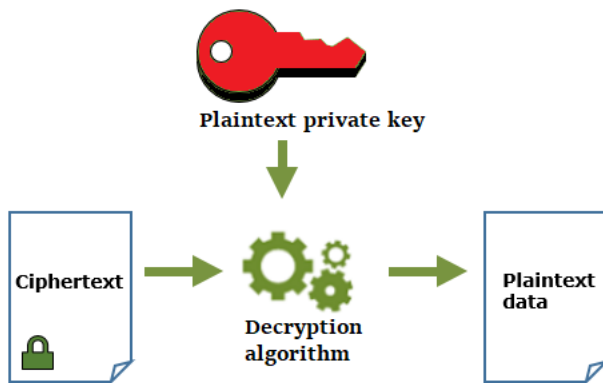


Descriptografar dados com um par de chaves de dados

Para descriptografar seus dados, use a chave privada no par de chaves de dados. Para que a operação seja bem-sucedida, as chaves pública e privada devem ser do mesmo par de chaves de dados e você deve usar o mesmo algoritmo de criptografia.

Para descriptografar a chave privada criptografada, transmita-a para a operação [Decrypt](#). Use a chave privada de texto não criptografado para descriptografar os dados. Depois, remova a chave privada de texto não criptografado da memória assim que possível.

O diagrama a seguir mostra como usar a chave privada em um par de chaves de dados para descriptografar texto cifrado.



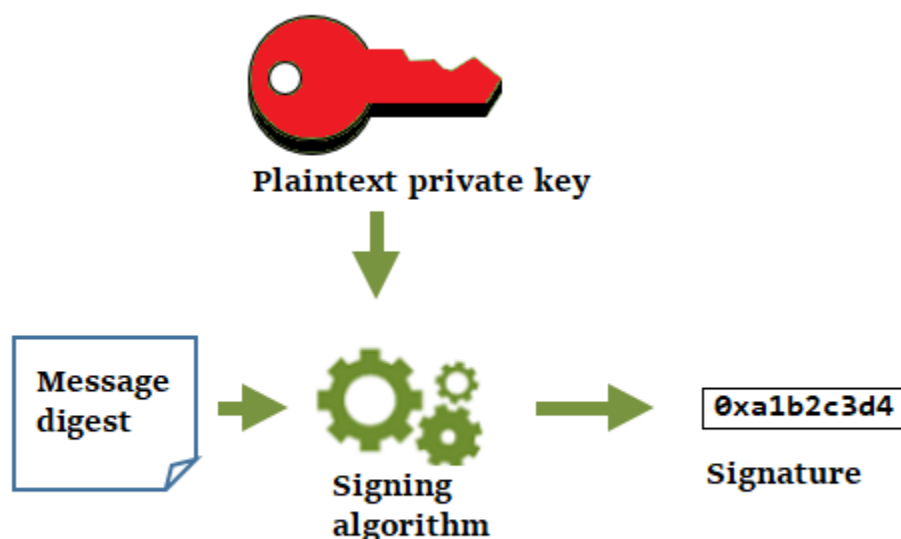
Assinar mensagens com um par de chaves de dados

Para gerar uma assinatura criptográfica para uma mensagem, use a chave privada no par de chaves de dados. Qualquer pessoa com a chave pública pode usá-la para verificar se a mensagem foi assinada com a chave privada e se ela não foi alterada desde que foi assinada.

Se você criptografar sua chave privada, passe a chave privada criptografada para a operação [Decrypt](#). AWS KMS usa sua chave KMS para descriptografar a chave de dados e, em seguida, retorna a chave privada em texto simples. Use a chave privada de texto não criptografado para gerar a assinatura. Depois, remova a chave privada de texto não criptografado da memória assim que possível.

Para assinar uma mensagem, crie um resumo de mensagens usando uma função de hash criptográfica, como o comando [dgst](#) no OpenSSL. Depois, transmita a chave privada de texto não criptografado ao algoritmo de assinatura. O resultado é uma assinatura que representa os conteúdos da mensagem. (Talvez seja possível assinar mensagens mais curtas sem antes criar um resumo. O tamanho máximo da mensagem varia de acordo com a ferramenta de assinatura que você usa.)

O diagrama a seguir mostra como usar a chave privada em um par de chaves para assinar a mensagem.

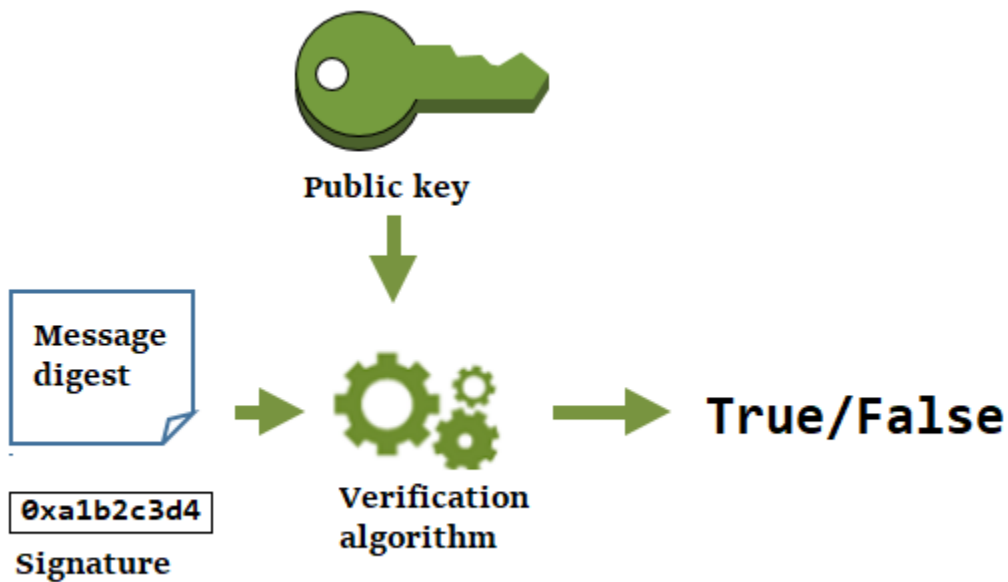


Verificar uma assinatura com um par de chaves de dados

Qualquer pessoa que tenha a chave pública no par de chaves de dados pode usá-la para verificar a assinatura gerada com sua chave privada. A verificação confirma que um usuário autorizado assinou a mensagem com a chave privada e o algoritmo de assinatura especificados, e que a mensagem não foi alterada desde que foi assinada.

Para ser bem-sucedida, a parte que está verificando a assinatura deve gerar o mesmo tipo de resumo, usar o mesmo algoritmo e usar a chave pública que corresponde à chave privada usada para assinar a mensagem.

O diagrama a seguir mostra como usar a chave pública em um par de chaves para verificar a assinatura de uma mensagem.



Derivar um segredo compartilhado com pares de chaves de dados

O acordo de chaves permite que dois pares, cada um com um par de chaves pública/privada de curva elíptica, estabeleçam um segredo compartilhado por um canal inseguro. Para [derivar um segredo compartilhado](#), os dois pares devem trocar suas chaves públicas pelo canal de comunicação inseguro (como a Internet). Em seguida, cada parte usa sua chave privada e a chave pública do seu par para calcular o mesmo segredo compartilhado usando um algoritmo de acordo de chave. Você pode usar o valor do segredo compartilhado para derivar uma chave simétrica que pode criptografar e descriptografar dados enviados entre os dois pares ou que podem ser gerados e verificados.

HMACs

Note

AWS KMS é altamente recomendável verificar se a chave pública que você recebe veio da parte esperada antes de usá-la para derivar um segredo compartilhado.

Realizar operações offline com chaves públicas

Em uma chave KMS assimétrica, a chave privada é criada AWS KMS e nunca sai sem criptografia. AWS KMS Para usar a chave privada, você deve ligar AWS KMS. Você pode usar a chave pública interna AWS KMS chamando as operações AWS KMS da API. Ou você pode [baixar a chave pública](#) e compartilhá-la para uso fora do AWS KMS.

Você pode compartilhar uma chave pública para permitir que outras pessoas criptografem dados além dos AWS KMS que você pode descriptografar somente com sua chave privada. Ou para permitir que outras pessoas verifiquem uma assinatura digital fora do AWS KMS que você gerou com a chave privada. Ou ainda para compartilhar sua chave pública com um par para obter um segredo compartilhado.

Ao usar a chave pública em sua chave KMS assimétrica AWS KMS, você se beneficia da autenticação, autorização e registro que fazem parte de cada operação. AWS KMS Você também reduz o risco de criptografar dados que não podem ser descriptografados. Esses recursos não são eficazes fora do AWS KMS. Para obter detalhes, consulte [Considerações especiais sobre o download de chaves públicas](#).

Tip

Está procurando por chaves de dados ou chaves SSH? Este tópico explica como gerenciar chaves assimétricas no AWS Key Management Service, onde a chave privada não é exportável. Para pares de chaves de dados exportáveis em que a chave privada é protegida por uma chave KMS de criptografia simétrica, consulte. [GenerateDataKeyPair](#) Para obter ajuda com o download da chave pública associada a uma instância do Amazon EC2, consulte Recuperação da chave pública no [Guia do usuário do Amazon EC2](#) e no [Guia do usuário do Amazon EC2](#).

Tópicos

- [Considerações especiais sobre o download de chaves públicas](#)
- [Fazer download de chave pública](#)
- [Exemplo de operações offline](#)

Considerações especiais sobre o download de chaves públicas

Para proteger suas chaves KMS, AWS KMS fornece controles de acesso, criptografia autenticada e registros detalhados de cada operação. AWS KMS também permite impedir o uso de chaves KMS, temporária ou permanentemente. Por fim, AWS KMS as operações são projetadas para minimizar o risco de criptografar dados que não podem ser descriptografados. Esses recursos não estão disponíveis quando você usa chaves públicas baixadas fora do AWS KMS.

Autorização

[Políticas-chave e políticas do IAM](#) que controlam o acesso à chave KMS interna AWS KMS não afetam as operações realizadas fora da AWS. Qualquer usuário que possa obter a chave pública pode usá-la externamente, AWS KMS mesmo que não tenha permissão para criptografar dados ou verificar assinaturas com a chave KMS.

Restrições de uso da chave

As principais restrições de uso não são efetivas fora do AWS KMS. Se você chamar a operação [Criptografar](#) com uma chave KMS que tenha um KeyUsage of SIGN_VERIFY, a AWS KMS operação falhará. Mas se você criptografar dados externos AWS KMS com uma chave pública de uma chave KMS com um KeyUsage de SIGN_VERIFY ou KEY_AGREEMENT, os dados não poderão ser descriptografados.

Restrições de algoritmo

Restrições nos algoritmos de criptografia e assinatura AWS KMS compatíveis não são efetivas fora do AWS KMS. Se você criptografar dados com a chave pública de uma chave KMS externa e usar um algoritmo de AWS KMS criptografia incompatível, os dados AWS KMS não poderão ser descriptografados.

Desabilitar e excluir chaves do KMS

As ações que você pode tomar para impedir o uso da chave KMS em uma operação criptográfica interna AWS KMS não impedem que ninguém use a chave pública fora dela. AWS KMS Por exemplo, desabilitar uma chave do KMS, programar a exclusão de uma chave do KMS, excluir uma chave do KMS ou excluir o material de chave de uma chave do KMS não têm efeito em uma chave pública fora do AWS KMS. Se você excluir uma chave KMS assimétrica ou excluir ou perder seu material de chaves, os dados que você criptografa com uma chave pública externa não poderão ser recuperados. AWS KMS

Registro em log

AWS CloudTrail registros que registram todas as AWS KMS operações, incluindo solicitação, resposta, data, hora e usuário autorizado, não registram o uso da chave pública fora do AWS KMS.

Verificação off-line com pares de SM2 chaves (somente regiões da China)

Para verificar uma assinatura externa AWS KMS com uma chave SM2 pública, você deve especificar a ID distintiva. Por padrão, AWS KMS usa 1234567812345678 como ID distintiva. Para obter mais informações, consulte [Verificação off-line com pares de SM2 chaves \(somente regiões da China\)](#).

Fazer download de chave pública

Você pode baixar a chave pública de um par de chaves KMS assimétrico no AWS KMS console ou usando a operação. [GetPublicKey](#) Para baixar a chave pública, é necessário ter a permissão `kms:GetPublicKey` na chave do KMS assimétrica.

[A chave pública que AWS KMS retorna é uma chave pública X.509 codificada em DER, também conhecida como SubjectPublicKeyInfo \(SPKI\), conforme definido na RFC 5280.](#) Quando você usa a API HTTP ou a AWS CLI, o valor é codificado em Base64. Caso contrário, ele não é codificado em base64.

Para baixar a chave pública de um par de chave do KMS assimétrica, é necessário ter a permissão `kms:GetPublicKey`. Para obter mais informações sobre AWS KMS permissões, consulte [Referência de permissões](#) o.

Usando o AWS KMS console

Você pode usar o Console de gerenciamento da AWS para visualizar, copiar e baixar a chave pública de uma chave KMS assimétrica em seu. Conta da AWS Para baixar a chave pública de uma chave KMS assimétrica em diferente Conta da AWS, use a API. AWS KMS

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente.
4. Escolha o alias ou o ID de chave de uma chave do KMS assimétrica.

5. Selecione a guia Cryptographic configuration (Configuração criptográfica). Registre os valores dos campos Especificação de chave, Uso de chave e Algoritmos de criptografia ou Algoritmos de assinatura. Você precisará usar esses valores para usar a chave pública fora do AWS KMS. Compartilhe essas informações ao compartilhar a chave pública.
6. Selecione a guia Public key (Chave pública).
7. Para copiar a chave pública para a área de transferência, selecione Copy (Copiar). Para fazer download da chave pública em um arquivo, selecione Download (Fazer download).

Usando a AWS KMS API

A [GetPublicKey](#) operação retorna a chave pública em uma chave KMS assimétrica. Ele também retorna informações críticas das quais você precisa usar corretamente a chave pública fora dela AWS KMS, incluindo o uso da chave e os algoritmos de criptografia. Salve esses valores e compartilhe-os sempre que compartilhar a chave pública.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Para especificar uma chave do KMS, use seu [ID de chave](#), [ARN de chave](#), [nome de alias](#) ou [ARN de alias](#). Ao usar um nome de alias, use alias/ como prefixo dele. Para especificar uma chave KMS em outra Conta da AWS, você deve usar o ARN da chave ou o alias ARN.

Antes de executar esse comando, substitua o nome de alias de exemplo por um identificador válido para a chave do KMS. Para executar esse comando, é necessário ter as permissões `kms:GetPublicKey` na chave do 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..."
}
```

Exemplo de operações offline

Após [baixar a chave pública](#) do seu par de chaves do KMS assimétricas, você poderá compartilhá-lo com outras pessoas e usá-lo para realizar operações offline.

AWS CloudTrail registros que registram todas as AWS KMS operações, incluindo solicitação, resposta, data, hora e usuário autorizado, não registram o uso da chave pública fora do AWS KMS.

Este tópico fornece exemplos de operações off-line e detalhes que as ferramentas AWS KMS fornecem para facilitar as operações off-line.

Tópicos

- [Derivar segredos compartilhados offline](#)
- [Verificação off-line com pares de chaves ML-DSA](#)
- [Verificação off-line com pares de SM2 chaves \(somente regiões da China\)](#)

Derivar segredos compartilhados offline

Você pode [baixar a chave pública](#) do seu par de chaves ECC para uso em operações offline, ou seja, operações fora do AWS KMS.

O seguinte passo a passo do [OpenSSL](#) demonstra um método de derivar um segredo compartilhado fora do uso da chave pública de um par AWS KMS de chaves ECC KMS e de uma chave privada criada com o OpenSSL.

1. Crie um par de chaves ECC no OpenSSL e prepare-o para uso com o. 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. Crie um par de chaves de acordo de chave ECC AWS KMS e prepare-o para uso com o 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. Derive o segredo compartilhado no OpenSSL usando a chave privada no OpenSSL e a chave pública do KMS.

```
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}
```

Verificação off-line com pares de chaves ML-DSA

AWS KMS suporta uma variante protegida da assinatura ML-DSA, conforme descrito na seção 3.4 do [padrão Federal Information Processing Standards \(FIPS\) 204](#) para mensagens de até 4 KB bytes.

Para assinar mensagens maiores que 4 KB, você executa a etapa de pré-processamento de mensagens fora do AWS KMS. Essa etapa de hash cria um μ representativo de mensagem de 64 bytes, conforme definido no NIST FIPS 204, seção 6.2.

AWS KMS tem um tipo de mensagem chamado `EXTERNAL_MU` para mensagens maiores que 4 KB. Quando você usa isso em vez do tipo de `RAW` mensagem, AWS KMS:

- Presume que você já tenha executado a etapa de hash
- Ignora seu próprio processo interno de hash
- Funciona com mensagens de qualquer tamanho

Quando você verifica uma mensagem, o método usado depende da restrição de tamanho da biblioteca ou sistema externo e ser ou não compatível com o μ representativo de mensagem de 64 bytes:

- Se a mensagem for menor que a restrição de tamanho, use o tipo de mensagem `RAW`.
- Se a mensagem for maior que a restrição de tamanho, use o μ representativo no sistema externo.

As seções a seguir demonstram como assinar AWS KMS e verificar mensagens usando o OpenSSL. Fornecemos exemplos para mensagens abaixo e acima do limite de tamanho de mensagem de 4 KB imposto pelo AWS KMS. O OpenSSL não impõe um limite no tamanho da mensagem para verificação.

Para ambos os exemplos, primeiro obtenha a chave pública de AWS KMS. Use o seguinte comando AWS CLI :

```
aws kms get-public-key \
  --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
  --output text \
  --query PublicKey | base64 --decode > public_key.der
```

Tamanho de mensagem menor que 4 KB

Para mensagens com menos de 4 KB, use o tipo de `RAW` mensagem com AWS KMS. Embora você possa usar `EXTERNAL_MU`, isso não é necessário para mensagens dentro do limite de tamanho.

Use o AWS CLI comando a seguir para assinar a mensagem:

```
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
```

Para verificar essa mensagem usando o OpenSSL, use o seguinte comando:

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature  
ExampleSignature.bin
```

Tamanho de mensagem maior que 4 KB

Para assinar mensagens maiores que 4 KB, use o tipo de mensagem `EXTERNAL_MU`. Quando usa `EXTERNAL_MU`, você faz externamente o pré-hash da mensagem para um μ representativo de 64 bytes conforme definido no NIST FIPS 204, seção 6.2 e a encaminha para as operações de assinatura ou verificação. Observe que isso é diferente de "MLDSA pré-hash" ou HashML-DSA definido no NIST FIPS 204, seção 5.4.

1. Primeiro, crie um prefixo de mensagem. O prefixo contém um separador de domínio, o tamanho de qualquer contexto e o contexto. O padrão para o separador de domínio e o comprimento de contexto é zero.
2. Anexe o prefixo à mensagem.
3. Use SHAKE256 para fazer o hash da chave pública e anexá-la ao resultado da etapa 2.
4. Por fim, faça um hash no resultado da etapa 3 para gerar o `EXTERNAL_MU` de 64 bytes.

O exemplo a seguir usa o OpenSSL 3.5 para criar o: `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
```

Depois de criar o `mu.bin` arquivo, chame a AWS KMS API com o seguinte comando para assinar a mensagem:

```
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
```

A assinatura resultante é igual à assinatura RAW da mensagem original. Você pode usar o mesmo comando do OpenSSL 3.5 para verificar a mensagem:

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature  
ExampleSignature.bin
```

Verificação off-line com pares de SM2 chaves (somente regiões da China)

Para verificar uma assinatura externa AWS KMS com uma chave SM2 pública, você deve especificar a ID distintiva. Quando você passa uma mensagem bruta, [MessageType:RAW](#), para a API [Sign](#), AWS KMS usa o ID distintivo padrão1234567812345678, definido pela OSCCA em 0009-2012. GM/T Você não pode especificar seu próprio ID de distinção no AWS KMS.

No entanto, se você estiver gerando um resumo da mensagem fora do AWS, poderá especificar seu próprio ID distintivo e, em seguida, passar o resumo da mensagem, [MessageType:DIGEST](#), para AWS KMS assinar. Para isso, altere o valor de DEFAULT_DISTINGUISHING_ID na classe SM2OfflineOperationHelper. O ID distintivo especificado pode ser qualquer string de até 8.192 caracteres. Depois de AWS KMS assinar o resumo da mensagem, você precisa do resumo da mensagem ou da mensagem e do ID distintivo usado para computar o resumo para verificá-lo offline.

Important

O código de referência SM2OfflineOperationHelper foi projetado para ser compatível com o [Bouncy Castle](#) versão 1.68. Para obter ajuda com outras versões, entre em contato com [bouncycastle.org](https://www.bouncycastle.org).

Classe SM2OfflineOperationHelper

Para ajudá-lo com operações off-line com SM2 chaves, a SM2OfflineOperationHelper classe para Java tem métodos que executam as tarefas para você. Essa classe auxiliar pode ser usada como modelo para outros provedores criptográficos.

No interior AWS KMS, as conversões de texto cifrado bruto e os cálculos de resumo de mensagens do SM2 DSA ocorrem automaticamente. Nem todos os provedores de criptografia SM2 implementam da mesma forma. Algumas bibliotecas, como as versões 1.1.1 e posteriores do [OpenSSL](#), executam essas ações automaticamente. AWS KMS confirmou esse comportamento em testes com o OpenSSL versão 3.0. Use a seguinte classe `SM2OfflineOperationHelper` com bibliotecas, como a [Bouncy Castle](#), que exigem que você faça essas conversões e cálculos manualmente.

A classe `SM2OfflineOperationHelper` fornece métodos para as seguintes operações:

- Cálculo de resumo de mensagem

Para gerar um resumo de mensagens offline que você possa usar para verificação off-line ou que você possa passar AWS KMS para assinar, use o `calculateSM2Digest` método. O `calculateSM2Digest` método gera um resumo da mensagem com o algoritmo de SM3 hash. A [GetPublicKey](#) API retorna sua chave pública em formato binário. Você deve analisar a chave binária em um Java `PublicKey`. Forneça a chave pública analisada com a mensagem. O método combina automaticamente sua mensagem com o ID distintivo padrão, 1234567812345678, mas você pode definir seu próprio ID distintivo alterando o valor de `DEFAULT_DISTINGUISHING_ID`.

- Verificar

Para verificar uma assinatura offline, use o método `offlineSM2DSAVerify`. O método `offlineSM2DSAVerify` usa o resumo de mensagem calculado com base no ID distintivo especificado e a mensagem original fornecida para verificar a assinatura digital. A [GetPublicKey](#) API retorna sua chave pública em formato binário. Você deve analisar a chave binária em um Java `PublicKey`. Forneça a chave pública analisada com a mensagem original e a assinatura que você deseja verificar. Para obter mais detalhes, consulte [Verificação off-line com pares de SM2 chaves](#).

- Encrypt

Para criptografar texto sem formatação offline, use o método `offlineSM2PKEEncrypt`. Esse método garante que o texto cifrado esteja em um formato que AWS KMS possa ser descifrado. O `offlineSM2PKEEncrypt` método criptografa o texto sem formatação e, em seguida, converte o texto cifrado bruto produzido pelo SM2 PKE no formato ASN.1. A [GetPublicKey](#) API retorna sua chave pública em formato binário. Você deve analisar a chave binária em um Java `PublicKey`. Forneça a chave pública analisada com o texto simples que você deseja criptografar.

Se não tiver certeza de que a conversão é necessário, use a seguinte operação OpenSSL para testar o formato do texto cifrado. Se a operação falhar, significa que você precisa converter o texto cifrado no formato ASN.1.

```
openssl asn1parse -inform DER -in ciphertext.der
```

Por padrão, a `SM2OfflineOperationHelper` classe usa o ID distintivo padrão, `1234567812345678`, ao gerar resumos de mensagens para operações de SM2 DSA.

```
package com.amazon.kms.utils;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import java.io.IOException;
import java.math.BigInteger;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.PrivateKey;
import java.security.PublicKey;

import org.bouncycastle.crypto.CryptoException;
import org.bouncycastle.jce.interfaces.ECPublicKey;

import java.util.Arrays;

import org.bouncycastle.asn1.ASN1EncodableVector;
import org.bouncycastle.asn1.ASN1Integer;
import org.bouncycastle.asn1.DEROctetString;
import org.bouncycastle.asn1.DERSequence;
import org.bouncycastle.asn1.gm.GMNamedCurves;
import org.bouncycastle.asn1.x9.X9ECParameters;
import org.bouncycastle.crypto.CipherParameters;
import org.bouncycastle.crypto.params.ParametersWithID;
import org.bouncycastle.crypto.params.ParametersWithRandom;
import org.bouncycastle.crypto.signers.SM2Signer;
```

```

import org.bouncycastle.jcajce.provider.asymmetric.util.ECUtil;

public class SM2OfflineOperationHelper {
    // You can change the DEFAULT_DISTINGUISHING_ID value to set your own
    distinguishing ID,
    // the DEFAULT_DISTINGUISHING_ID can be any string up to 8,192 characters long.
    private static final byte[] DEFAULT_DISTINGUISHING_ID =
"1234567812345678".getBytes(StandardCharsets.UTF_8);
    private static final X9ECParameters SM2_X9EC_PARAMETERS =
GMNamedCurves.getByName("sm2p256v1");

    // ***calculateSM2Digest***
    // Calculate message digest
    public static byte[] calculateSM2Digest(final PublicKey publicKey, final byte[]
message) throws
        NoSuchProviderException, NoSuchAlgorithmException {
        final ECPublicKey ecPublicKey = (ECPublicKey) publicKey;

        // Generate SM3 hash of default distinguishing ID, 1234567812345678
        final int entlenA = DEFAULT_DISTINGUISHING_ID.length * 8;
        final byte [] entla = new byte[] { (byte) (entlenA & 0xFF00), (byte) (entlenA &
0x00FF) };
        final byte [] a = SM2_X9EC_PARAMETERS.getCurve().getA().getEncoded();
        final byte [] b = SM2_X9EC_PARAMETERS.getCurve().getB().getEncoded();
        final byte [] xg = SM2_X9EC_PARAMETERS.getG().getXCoord().getEncoded();
        final byte [] yg = SM2_X9EC_PARAMETERS.getG().getYCoord().getEncoded();
        final byte[] xa = ecPublicKey.getQ().getXCoord().getEncoded();
        final byte[] ya = ecPublicKey.getQ().getYCoord().getEncoded();
        final byte[] za = MessageDigest.getInstance("SM3", "BC")
            .digest(ByteBuffer.allocate(entla.length +
DEFAULT_DISTINGUISHING_ID.length + a.length + b.length + xg.length + yg.length +
xa.length +
ya.length).put(entla).put(DEFAULT_DISTINGUISHING_ID).put(a).put(b).put(xg).put(yg).put(xa).put
            .array());

        // Combine hashed distinguishing ID with original message to generate final
        digest
        return MessageDigest.getInstance("SM3", "BC")
            .digest(ByteBuffer.allocate(za.length +
message.length).put(za).put(message)
            .array());
    }

    // ***offlineSM2DSAVerify***

```

```

// Verify digital signature with SM2 public key
public static boolean offlineSM2DSAVerify(final PublicKey publicKey, final byte []
message,
    final byte [] signature) throws InvalidKeyException {
    final SM2Signer signer = new SM2Signer();
    CipherParameters cipherParameters =
ECUtil.generatePublicKeyParameter(publicKey);
    cipherParameters = new ParametersWithID(cipherParameters,
DEFAULT_DISTINGUISHING_ID);
    signer.init(false, cipherParameters);
    signer.update(message, 0, message.length);
    return signer.verifySignature(signature);
}

// ***offlineSM2PKEEncrypt***
// Encrypt data with SM2 public key
public static byte[] offlineSM2PKEEncrypt(final PublicKey publicKey, final byte []
plaintext) throws
    NoSuchPaddingException, NoSuchAlgorithmException, NoSuchProviderException,
InvalidKeyException,
    BadPaddingException, IllegalBlockSizeException, IOException {
    final Cipher sm2Cipher = Cipher.getInstance("SM2", "BC");
    sm2Cipher.init(Cipher.ENCRYPT_MODE, publicKey);

    // By default, Bouncy Castle returns raw ciphertext in the c1c2c3 format
    final byte [] cipherText = sm2Cipher.doFinal(plaintext);

    // Convert the raw ciphertext to the ASN.1 format before passing it to AWS KMS
    final ASN1EncodableVector asn1EncodableVector = new ASN1EncodableVector();
    final int coordinateLength = (SM2_X9EC_PARAMETERS.getCurve().getFieldSize() +
7) / 8 * 2 + 1;
    final int sm3HashLength = 32;
    final int xCoordinateInCipherText = 33;
    final int yCoordinateInCipherText = 65;
    byte[] coords = new byte[coordinateLength];
    byte[] sm3Hash = new byte[sm3HashLength];
    byte[] remainingCipherText = new byte[cipherText.length - coordinateLength -
sm3HashLength];

    // Split components out of the ciphertext
    System.arraycopy(cipherText, 0, coords, 0, coordinateLength);
    System.arraycopy(cipherText, cipherText.length - sm3HashLength, sm3Hash, 0,
sm3HashLength);

```

```
        System.arraycopy(cipherText, coordinateLength, remainingCipherText,
0,cipherText.length - coordinateLength - sm3HashLength);

        // Build standard SM2PKE ASN.1 ciphertext vector
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
Arrays.copyOfRange(coords, 1, xCoordinateInCipherText))));
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
Arrays.copyOfRange(coords, xCoordinateInCipherText, yCoordinateInCipherText))));
        asn1EncodableVector.add(new DEROctetString(sm3Hash));
        asn1EncodableVector.add(new DEROctetString(remainingCipherText));

        return new DERSequence(asn1EncodableVector).getEncoded("DER");
    }
}
```

Monitorar AWS KMS keys

O monitoramento é uma parte importante para entender a disponibilidade, o estado e o uso de suas AWS KMS keys informações AWS KMS e manter a confiabilidade, a disponibilidade e o desempenho de suas AWS soluções. Coletar dados de monitoramento de todas as partes de sua solução da AWS irá ajudá-lo a depurar uma falha de vários pontos, caso ocorra. Porém, para começar a monitorar suas chaves do KMS, é necessário criar um plano de monitoramento que inclua respostas às seguintes perguntas:

- Quais são seus objetivos de monitoramento?
- Quais recursos você vai monitorar?
- Com que frequência você vai monitorar esses recursos?
- Quais [ferramentas de monitoramento](#) você usará?
- Quem realizará o monitoramento das tarefas?
- Quem deve ser notificado quando algo acontece?

A próxima etapa é monitorar as chaves do KMS ao longo do tempo para estabelecer uma linha de base para a utilização normal do AWS KMS e as expectativas no ambiente. Ao monitorar as suas chaves do KMS, armazene dados de monitoramento históricos para compará-los com os dados atuais, identificar padrões normais e anomalias e elaborar métodos para resolver problemas.

Por exemplo, você pode monitorar a atividade e os eventos AWS KMS da API que afetam suas chaves KMS. Quando os dados ficam acima ou abaixo das normas estabelecidas, pode ser necessário investigar ou executar a ação corretiva.

Para estabelecer uma linha de base para padrões normais, monitore os seguintes itens:

- AWS KMS Atividade de API para operações de plano de dados. [Essas são operações criptográficas que usam uma chave KMS, como Decrypt, Encrypt, e. ReEncryptGenerateDataKey](#)
- AWS KMS Atividade de API para operações de plano de controle que são importantes para você. Essas operações gerenciam uma chave KMS, e talvez você queira monitorar aquelas que alteram a disponibilidade de uma chave KMS (como [ScheduleKeyDeletion](#),,, [CancelKeyDeletionDisableKeyEnableKeyImportKeyMaterial](#), e [DeleteImportedKeyMaterial](#)) ou alteram o controle de acesso de uma chave KMS (como [PutKeyPolicy](#) e [RevokeGrant](#)

- Outras AWS KMS métricas (como o tempo restante até que seu [material de chave importado](#) expire) e eventos (como a expiração do material de chave importado ou a exclusão ou rotação de chaves de uma chave KMS).

Ferramentas de monitoramento

AWS fornece várias ferramentas que você pode usar para monitorar suas chaves KMS. É possível configurar algumas dessas ferramentas para fazer o monitoramento em seu lugar, e, ao mesmo tempo, algumas das ferramentas exigem intervenção manual. Recomendamos que as tarefas de monitoramento sejam automatizadas ao máximo possível.

Ferramentas de monitoramento automatizadas

Use as seguintes ferramentas de monitoramento automatizadas para observar suas chaves do KMS e gerar relatórios quando algo for alterado.

- AWS CloudTrail Monitoramento de registros — compartilhe arquivos de log entre contas, monitore arquivos de CloudTrail log em tempo real enviando-os para o CloudWatch Logs, grave aplicativos de processamento de log com a [Biblioteca de CloudTrail Processamento](#) e valide se seus arquivos de log não foram alterados após a entrega. CloudTrail Para obter mais informações, consulte [Trabalhando com arquivos de CloudTrail log](#) no Guia AWS CloudTrail do usuário.
- Amazon CloudWatch Alarms — Observe uma única métrica durante um período de tempo especificado por você e execute uma ou mais ações com base no valor da métrica em relação a um determinado limite em vários períodos. A ação é uma notificação enviada para um tópico do Amazon Simple Notification Service (Amazon SNS) ou para uma política do Amazon Auto EC2 Scaling. CloudWatch os alarmes não invocam ações simplesmente porque estão em um determinado estado; o estado deve ter sido alterado e mantido por um determinado número de períodos. Para obter mais informações, consulte [Monitore as chaves KMS com a Amazon CloudWatch](#).
- Amazon EventBridge — Combine eventos e encaminhe-os para uma ou mais funções ou fluxos de destino para capturar informações sobre o estado e, se necessário, fazer alterações ou tomar medidas corretivas. Para obter mais informações, consulte [Monitore as chaves KMS com a Amazon EventBridge](#) o [Guia do EventBridge usuário da Amazon](#).
- Amazon CloudWatch Logs — Monitore, armazene e acesse seus arquivos de log de AWS CloudTrail ou de outras fontes. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).

Ferramentas de monitoramento manual

Outra parte importante do monitoramento das chaves KMS envolve o monitoramento manual dos itens que os CloudWatch alarmes e eventos não cobrem. Os AWS painéis AWS KMS CloudWatch, AWS Trusted Advisor,, e outros fornecem uma at-a-glance visão do estado do seu AWS ambiente.

É possível [personalizar](#) as páginas Chaves gerenciadas pela AWS e Chaves gerenciadas pelo cliente do [console do AWS KMS](#) para exibir as seguintes informações sobre cada chave do KMS:

- ID da chave
- Status
- Data de Criação
- Data de validade (para chaves do KMS com [material de chave importado](#))
- Origem
- ID de armazenamento de chave personalizado (para chaves do KMS em [armazenamentos de chaves personalizados](#))

O [painel do console do CloudWatch](#) mostra o seguinte:

- Alertas e status atual
- Gráficos de alertas e recursos
- Estado de integridade do serviço

Além disso, você pode usar CloudWatch para fazer o seguinte:

- Crie [painéis personalizados](#) para monitorar os serviços com os quais você se preocupa.
- Colocar em gráfico dados de métrica para solucionar problemas e descobrir tendências
- Pesquise e navegue em todas as suas métricas AWS de recursos
- Criar e editar alertas para ser notificado sobre problemas

AWS Trusted Advisor pode ajudá-lo a monitorar seus AWS recursos para melhorar o desempenho, a confiabilidade, a segurança e a economia. Quatro Trusted Advisor cheques estão disponíveis para todos os usuários; mais de 50 cheques estão disponíveis para usuários com um plano de suporte Business ou Enterprise. Para obter mais informações, consulte [AWS Trusted Advisor](#).

Registrando chamadas de AWS KMS API com AWS CloudTrail

AWS KMS é integrado com [AWS CloudTrail](#), um serviço que registra todas as chamadas feitas AWS KMS por usuários, funções e outros AWS serviços. CloudTrail captura todas as chamadas de API para AWS KMS como eventos, incluindo chamadas do AWS KMS console AWS KMS APIs, CloudFormation modelos, do AWS Command Line Interface (AWS CLI) e. Ferramentas da AWS para PowerShell

CloudTrail [registra todas as AWS KMS operações, incluindo operações somente para leitura, como ListAliases e GetKeyRotationStatus, operações que gerenciam chaves KMS, como e, CreateKey e operações criptográficas PutKeyPolicy, como e Decrypt. GenerateDataKey](#) Ele também registra operações internas que AWS KMS chamam por você [DeleteExpiredKeyMaterial](#), como [DeleteKeySynchronizeMultiRegionKey](#), [RotateKey](#).

CloudTrail registra todas as operações bem-sucedidas e, em alguns cenários, tentativas de chamadas que falharam, como quando o chamador não tem acesso a um recurso. As [operações entre contas em chaves do KMS](#) são registradas em log na conta do autor da chamada e na conta do proprietário da chave do KMS. No entanto, as AWS KMS solicitações entre contas que são rejeitadas porque o acesso foi negado são registradas somente na conta do chamador.

Por motivos de segurança, alguns campos são omitidos das entradas de AWS KMS registro, como o Plaintext parâmetro de uma solicitação [Encrypt](#) e a resposta [GetKeyPolicy](#) ou qualquer operação criptográfica. Para facilitar a pesquisa de entradas de CloudTrail registro para chaves KMS específicas, AWS KMS adiciona o [ARN da chave](#) KMS afetada ao responseElements campo nas entradas de registro de AWS KMS algumas operações de gerenciamento de chaves, mesmo quando a operação da API não retorna o ARN da chave.

Embora, por padrão, todas AWS KMS as ações sejam registradas como CloudTrail eventos, você pode excluir AWS KMS ações de uma CloudTrail trilha. Para obter detalhes, consulte [Excluindo AWS KMS eventos de uma trilha](#).

Saiba mais:

- Para obter exemplos de CloudTrail registros de AWS KMS operações para plataformas certificadas, consulte [Monitorar solicitações atestadas](#).

Tópicos

- [Localizando entradas de AWS KMS registro em CloudTrail](#)

- [Excluindo AWS KMS eventos de uma trilha](#)
- [Exemplos de entradas de AWS KMS registro](#)

Localizando entradas de AWS KMS registro em CloudTrail

Para pesquisar entradas de CloudTrail registro, use o [CloudTrail console](#) ou a [CloudTrail LookupEvents](#) operação. CloudTrail suporta vários [valores de atributos](#) para filtrar sua pesquisa, incluindo nome do evento, nome do usuário e fonte do evento.

Para ajudá-lo a pesquisar entradas de AWS KMS registro em CloudTrail, AWS KMS preenche os seguintes campos de entrada de CloudTrail registro.

Note

A partir de dezembro de 2022, AWS KMS preenche os atributos Tipo de recurso e Nome do recurso em todas as operações de gerenciamento que alteram uma chave KMS específica. Esses valores de atributos podem ser nulos em CloudTrail entradas mais antigas para as seguintes operações: [CreateAlias](#), [CreateGrant](#), [DeleteAlias](#), [DeleteImportedKeyMaterial](#), [ImportKeyMaterial](#), [ReplicateKey](#), [RetireGrant](#), [RevokeGrant](#), [UpdateAlias](#), e. [UpdatePrimaryRegion](#)

Atributo	Valor	Entradas de log
Origem do evento (EventSource)	kms . amazonaws . com	Todas as operações.
Tipo de recurso (ResourceType)	AWS : : KMS : : Key	Operações de gerenciamento que alteram uma chave do KMS específica, como <code>CreateKey</code> e <code>EnableKey</code> , mas não <code>ListKeys</code> .
Nome do recurso (ResourceName)	ARN da chave (ou ID da chave e ARN da chave)	Operações de gerenciamento que alteram uma chave do KMS específica, como

Atributo	Valor	Entradas de log
		CreateKey e EnableKey , mas não ListKeys.

Para ajudá-lo a encontrar entradas de registro para operações de gerenciamento em chaves KMS específicas, AWS KMS registra o ARN da chave KMS afetada no elemento `responseElements.keyId` da entrada de registro, mesmo quando a operação da API não AWS KMS retorna o ARN da chave.

Por exemplo, uma chamada bem-sucedida para a [DisableKey](#) operação não retorna nenhum valor na resposta, mas em vez de um valor nulo, o `responseElements.keyId` valor na [entrada do DisableKey registro](#) inclui o ARN da chave KMS desativada.

Esse recurso foi adicionado em dezembro de 2022 e afeta as seguintes entradas de CloudTrail registro: [CreateAliasCreateGrantDeleteAlias](#), [DeleteKey](#), [DisableKey](#), [EnableKey](#), [EnableKeyRotation](#), [ImportKeyMaterial](#), [RotateKey](#), [SynchronizeMultiRegionKey](#), [TagResource](#), [UntagResource](#), [UpdateAlias](#), [UpdatePrimaryRegion](#).

Excluindo AWS KMS eventos de uma trilha

Para fornecer um registro do uso e gerenciamento de seus AWS KMS recursos, a maioria dos AWS KMS usuários confia nos eventos em uma CloudTrail trilha. A trilha pode ser uma fonte valiosa de dados para auditar eventos críticos, como criar, desativar e excluir AWS KMS keys, alterar a política de chaves e o uso de suas chaves KMS por AWS serviços em seu nome. Em alguns casos, os metadados em uma entrada de CloudTrail registro, como o [contexto de criptografia](#) em uma operação de criptografia, podem ajudá-lo a evitar ou resolver erros.

No entanto, como AWS KMS pode gerar um grande número de eventos, AWS CloudTrail permite excluir AWS KMS eventos de uma trilha. Essa configuração por trilha exclui todos os AWS KMS eventos; você não pode excluir eventos específicos AWS KMS .

Warning

A exclusão de AWS KMS eventos de um CloudTrail registro pode obscurecer ações que usam suas chaves KMS. Tenha cuidado ao conceder aos principais a permissão `cloudtrail:PutEventSelectors` que é necessária para executar essa operação.

Para excluir AWS KMS eventos de uma trilha:

- No CloudTrail console, use a configuração Registrar eventos do Serviço de Gerenciamento de Chaves ao [criar uma trilha](#) ou [atualizar uma trilha](#). Para obter instruções, consulte [Registrar eventos de gerenciamento com o Console de gerenciamento da AWS](#) no Guia AWS CloudTrail do usuário.
- Na CloudTrail API, use a [PutEventSelectors](#) operação. Adicione o atributo `ExcludeManagementEventSources` aos seletores de eventos com um valor de `kms.amazonaws.com`. Por exemplo, consulte [Exemplo: uma trilha que não registra AWS Key Management Service eventos](#) no Guia do AWS CloudTrail usuário.

É possível desativar essa exclusão a qualquer momento alterando a configuração do console ou os seletores de eventos de uma trilha. A trilha então começará a registrar AWS KMS os eventos. No entanto, ele não pode recuperar AWS KMS eventos que ocorreram enquanto a exclusão era efetiva.

Quando você exclui AWS KMS eventos usando o console ou a API, a operação de CloudTrail `PutEventSelectors` API resultante também é registrada nos seus CloudTrail registros. Se AWS KMS os eventos não aparecerem nos seus CloudTrail registros, procure um `PutEventSelectors` evento com o `ExcludeManagementEventSources` atributo definido como `kms.amazonaws.com`.

Exemplos de entradas de AWS KMS registro

AWS KMS grava entradas no seu CloudTrail registro quando você chama uma AWS KMS operação e quando um AWS serviço chama uma operação em seu nome. AWS KMS também grava uma entrada quando chama uma operação para você. Por exemplo, ele grava uma entrada quando [exclui uma chave do KMS](#) que você agendou para exclusão.

Os tópicos a seguir mostram exemplos de entradas de CloudTrail registro para AWS KMS operações.

Para obter exemplos de entradas de CloudTrail registro de solicitações AWS KMS de plataformas certificadas, consulte [Monitorar solicitações atestadas](#).

Tópicos

- [CancelKeyDeletion](#)
- [ConnectCustomKeyStore](#)
- [CreateAlias](#)
- [CreateCustomKeyStore](#)

- [CreateGrant](#)
- [CreateKey](#)
- [Decrypt](#)
- [DeleteAlias](#)
- [DeleteCustomKeyStore](#)
- [DeleteExpiredKeyMaterial](#)
- [DeleteImportedKeyMaterial](#)
- [DeleteKey](#)
- [DescribeCustomKeyStores](#)
- [DescribeKey](#)
- [DisableKey](#)
- [DisableKeyRotation](#)
- [DisconnectCustomKeyStore](#)
- [EnableKey](#)
- [EnableKeyRotation](#)
- [Encrypt](#)
- [GenerateDataKey](#)
- [GenerateDataKeyPair](#)
- [GenerateDataKeyPairWithoutPlaintext](#)
- [GenerateDataKeyWithoutPlaintext](#)
- [GenerateMac](#)
- [GenerateRandom](#)
- [GetKeyPolicy](#)
- [GetKeyRotationStatus](#)
- [GetParametersForImport](#)
- [ImportKeyMaterial](#)
- [ListAliases](#)
- [ListGrants](#)
- [ListKeyRotations](#)
- [PutKeyPolicy](#)

- [ReEncrypt](#)
- [ReplicateKey](#)
- [RetireGrant](#)
- [RevokeGrant](#)
- [RotateKey](#)
- [RotateKeyOnDemand](#)
- [ScheduleKeyDeletion](#)
- [Sign](#)
- [SynchronizeMultiRegionKey](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateAlias](#)
- [UpdateCustomKeyStore](#)
- [UpdateKeyDescription](#)
- [UpdatePrimaryRegion](#)
- [VerifyMac](#)
- [Verificar](#)
- [Amazon EC2 , exemplo um](#)
- [Amazon EC2 , exemplo dois](#)

CancelKeyDeletion

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [CancelKeyDeletion](#) operação. Para obter informações sobre a exclusão AWS KMS keys, consulte [Excluir um 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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [ConnectCustomKeyStore](#) operação. Para obter informações sobre conexão de um armazenamento de chaves personalizado, consulte [Desconectar um armazenamento de AWS CloudHSM chaves](#).

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [CreateAlias](#) operação. O elemento `resources` inclui campos para o alias e recursos de chaves do KMS. Para obter informações sobre a criação de aliases em AWS KMS, consulte [Criar alias](#).

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada pela chamada da [CreateCustomKeyStore](#) operação em um armazenamento de AWS CloudHSM chaves. Para

obter informações sobre a criação de armazenamentos de chaves personalizados, consulte [Crie um armazenamento de AWS CloudHSM chaves](#).

```
{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [CreateGrant](#) operação. Para obter informações sobre a criação de subsídios em AWS KMS, consulte [Subsídios em AWS KMS](#).

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:53:12Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "constraints": {
      "encryptionContextSubset": {
        "ContextKey1": "Value1"
      }
    },
    "operations": ["Encrypt", "RetireGrant"],
    "granteePrincipal": "EX_PRINCIPAL_ID"
  },
  "responseElements": {
    "grantId": "f020fe75197b93991dc8491d6f19dd3cebb24ee62277a05914386724f3d48758",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "f3c08808-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "5d529779-2d27-42b5-92da-91aaea1fc4b5",
  "readOnly": false,
  "resources": [{
    "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

CreateKey

Esses exemplos mostram entradas de AWS CloudTrail registro da [CreateKey](#) operação.

Uma entrada de CreateKey registro pode resultar de uma CreateKey solicitação ou da CreateKey operação de uma [ReplicateKey](#) solicitação.

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma [CreateKey](#) operação que cria uma chave [KMS de criptografia simétrica](#). Para obter mais informações sobre como criar chaves do KMS, consulte [Criar uma chave do 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"
}

```

O exemplo a seguir mostra o CloudTrail log de uma CreateKey operação que cria uma chave KMS de criptografia simétrica em um armazenamento de [AWS CloudHSM chaves](#).

```

{
    "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",
        "customKeyId": "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",
            "customKeyId": "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"
  }
}

```

O exemplo a seguir mostra o CloudTrail log de uma CreateKey operação que cria uma chave KMS de criptografia simétrica em um armazenamento de [chaves externo](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-09-07T22:37:45Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "tags": [],
    "keyUsage": "ENCRYPT_DECRYPT",
    "description": "",

```

```

    "origin": "EXTERNAL_KEY_STORE",
    "multiRegion": false,
    "keySpec": "SYMMETRIC_DEFAULT",
    "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "bypassPolicyLockoutSafetyCheck": false,
    "customKeyId": "cks-1234567890abcdef0",
    "xksKeyId": "bb8562717f809024"
  },
  "responseElements": {
    "keyMetadata": {
      "awsAccountId": "111122223333",
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "creationDate": "Dec 7, 2022, 10:37:45 PM",
      "enabled": true,
      "description": "",
      "keyUsage": "ENCRYPT_DECRYPT",
      "keyState": "Enabled",
      "origin": "EXTERNAL_KEY_STORE",
      "customKeyId": "cks-1234567890abcdef0",
      "keyManager": "CUSTOMER",
      "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "keySpec": "SYMMETRIC_DEFAULT",
      "encryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
      ],
      "multiRegion": false,
      "xksKeyConfiguration": {
        "id": "bb8562717f809024"
      }
    }
  },
  "requestID": "ba197c82-3ac7-487a-8ff4-7736bbbeb1316",
  "eventID": "838ad5f4-5fdd-4044-afd7-4dbd88c6af56",
  "readOnly": false,
  "resources": [
    {
      "accountId": "227179770375",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-east-1:227179770375:key/39c5eb22-
f37c-4956-92ca-89e8f8b57ab2"
    }
  ],

```

```
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}
```

Decrypt

Esses exemplos mostram entradas de AWS CloudTrail registro para a operação [Decrypt](#).

A entrada de CloudTrail registro de uma Decrypt operação sempre inclui o `encryptionAlgorithm` in the, `requestParameters` mesmo que o algoritmo de criptografia não tenha sido especificado na solicitação. O texto cifrado na solicitação e o texto sem formatação na resposta são omitidos.

Tópicos

- [Descriptografar com uma chave de criptografia simétrica padrão](#)
- [Descriptografar falhas com uma chave de criptografia simétrica padrão](#)
- [Descriptografar com uma chave KMS em um armazenamento de chaves AWS CloudHSM](#)
- [Descriptografar com uma chave do KMS em um armazenamento de chaves externas](#)
- [Descriptografar falhas com uma chave do KMS em um armazenamento de chaves externas](#)
- [Descriptografe com uma chave de criptografia simétrica padrão em uma conexão TLS pós-quântica](#)

Descriptografar com uma chave de criptografia simétrica padrão

Veja a seguir um exemplo de entrada de CloudTrail registro para uma Decrypt operação com uma chave de criptografia simétrica padrão.

```
{
  "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"
    }
  }
}

```

Descriptografar falhas com uma chave de criptografia simétrica padrão

O exemplo de entrada de CloudTrail registro a seguir registra uma Decrypt operação com falha com uma chave KMS de criptografia simétrica padrão. A exceção (errorCode) e a mensagem de erro (errorMessage) foram incluídas para ajudar a resolver o erro.

Nesse caso, a chave do KMS de criptografia simétrica especificada na solicitação Decrypt não era a chave do KMS de criptografia simétrica usada para criptografar os dados.

```
{
  "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"
}

```

Descriptografar com uma chave KMS em um armazenamento de chaves AWS CloudHSM

O exemplo de entrada de CloudTrail registro a seguir Decrypt registra uma operação com uma chave KMS em um [armazenamento de AWS CloudHSM chaves](#). Todas as entradas de log para operações criptográficas com uma chave do KMS em um repositório de chaves personalizado incluem um campo `additionalEventData` com `customKeyStoreId` e `backingKeyId`. O valor retornado no campo `backingKeyId` é o atributo `id` da chave do CloudHSM. `additionalEventData` não é especificado na solicitação.

```

{
  "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"
  }
}

```

Descriptografar com uma chave do KMS em um armazenamento de chaves externas

O exemplo de entrada de CloudTrail registro a seguir Decrypt registra uma operação com uma chave KMS em um [armazenamento de chaves externo](#). Além de customKeyStoreId, o campo additionalEventData inclui o [ID da chave externa](#) (XksKeyId). additionalEventData não é especificado na solicitação.

```

{
  "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"
  }

```

Descriptografar falhas com uma chave do KMS em um armazenamento de chaves externas

O exemplo de entrada de CloudTrail registro a seguir registra uma falha na solicitação de uma Decrypt operação com uma chave KMS em um [armazenamento de chaves externo](#). CloudWatch registra solicitações que falham, além de solicitações bem-sucedidas. Ao registrar uma falha, a entrada do CloudTrail registro inclui a exceção (ErrorCode) e a mensagem de erro que a acompanha (ErrorMessage).

Se a solicitação com falha atingir seu proxy de armazenamento de chaves externas, como neste exemplo, você poderá usar o valor requestId para associar a solicitação com falha a uma

solicitação correspondente que seu proxy de armazenamento de chaves externas registra, se o proxy os fornecer.

Para obter ajuda com solicitações Decrypt em armazenamentos de chaves externas, consulte [Erros decriptografia](#).

```
{
  "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"
  }
}

```

Descriptografe com uma chave de criptografia simétrica padrão em uma conexão TLS pós-quântica

Veja a seguir um exemplo de entrada de CloudTrail registro para uma Decrypt operação com uma chave de criptografia simétrica padrão em uma conexão TLS pós-quântica. O campo KeyExchange na tlsDetails seção X25519MLKEM768 menciona. [Esse é um algoritmo híbrido que combina a curva elíptica Diffie-Hellman \(ECDH\) com a curva 25519, um algoritmo clássico de troca de chaves usado atualmente no TLS, com o mecanismo de encapsulamento de chaves baseado em rede modular \(\), um algoritmo de criptografia de chave pública e estabelecimento de chaves que o Instituto Nacional de Padrões e Tecnologia \(NISTML-KEM\) designou como seu primeiro algoritmo padrão de acordo de chave pós-quântico.](#)

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [DeleteAlias](#) operação. Para obter informações sobre como excluir aliases, consulte [Excluir um alias](#).

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

```
{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [DeleteCustomKeyStore](#) operação. Para obter informações sobre a criação de armazenamentos de chaves personalizados, consulte [Excluir um armazenamento de AWS CloudHSM chaves](#).

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-21T20:17:32Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DeleteCustomKeyStore",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "customKeyId": "cks-1234567890abcdef0"
  },
  "responseElements": null,
  "additionalEventData": {
    "customKeyName": "ExampleKeyStore",
    "clusterId": "cluster-1a23b4cdefg"
  },
  "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
  "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333"
}
```

```
}
```

DeleteExpiredKeyMaterial

Ao importar material de chave para uma AWS KMS key (chave KMS), você pode definir uma data e hora de expiração para esse material de chave. AWS KMS registra uma entrada em seu CloudTrail registro quando você [importa o material da chave](#) (com as configurações de expiração) e quando AWS KMS exclui o material da chave expirada. Para obter informações sobre como criar uma chave do KMS com o material de chave importado, consulte [Importação de material chave para AWS KMS chaves](#).

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao AWS KMS excluir o material da chave expirada.

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2025-05-22T19:55:11Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DeleteExpiredKeyMaterial",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "cfa932fd-0d3a-4a76-a8b8-616863a2b547",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

```

    "keyMaterialId":
    "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  },
  "eventCategory": "Management"
}

```

DeleteImportedKeyMaterial

Se você importar material de chave em uma chave KMS, poderá excluir o material de chave importado a qualquer momento usando a [DeleteImportedKeyMaterial](#) operação. Quando você exclui o material de chave importado de uma chave do KMS, o estado da chave da chave do KMS é alterado para PendingImport, e a chave do KMS não pode ser usada em operações de criptografia. Para obter detalhes, consulte [Excluir material de chave importado](#).

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada para a DeleteImportedKeyMaterial operação.

```

{
  "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

Esses exemplos mostram a entrada de AWS CloudTrail registro que é gerada quando uma chave KMS é excluída. Para excluir uma chave KMS, você usa a [ScheduleKeyDeletion](#) operação. Depois que o período de espera especificado expirar, AWS KMS exclui a chave KMS e registra uma entrada como a seguinte em seu registro para CloudTrail registrar esse evento.

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

Para obter um exemplo da entrada de CloudTrail registro da `ScheduleKeyDeletion` operação, consulte [ScheduleKeyDeletion](#). Para informações sobre como excluir chaves do KMS, consulte [Excluir um AWS KMS key](#).

O exemplo de entrada de CloudTrail registro a seguir registra uma `DeleteKey` operação de uma chave KMS com material de chave inserido. 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"
}
```

A entrada de CloudTrail registro a seguir registra uma DeleteKey operação de uma chave KMS em um [armazenamento de chaves AWS CloudHSM personalizado](#).

```
{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [DescribeCustomKeyStores](#) operação. Para obter informações sobre visualização de armazenamentos de chaves personalizados, consulte [Exibir uma loja de AWS CloudHSM chaves](#).

```

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",

```

```

    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-21T20:17:32Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeCustomKeyStores",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "customKeyId": "cks-1234567890abcdef0"
  },
  "responseElements": null,
  "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
  "eventID": "2ea1735f-628d-43e3-b2ee-486d02913a78",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333"
}

```

DescribeKey

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [DescribeKey](#) operação. AWS KMS registra uma entrada como a seguinte quando você chama a DescribeKey operação ou [visualiza as chaves KMS](#) no AWS KMS console. Essa chamada é o resultado da visualização de uma chave no console AWS KMS de gerenciamento.

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [DisableKey](#) operação. Para obter informações sobre como ativar e desativar o AWS KMS keys in AWS KMS, consulte [Habilitar e desabilitar chaves](#).

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

```

{
    "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [DisableKeyRotation](#) operação. Para obter informações sobre alternância automática de chaves, consulte [Girar 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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [DisconnectCustomKeyStore](#) operação. Para obter informações sobre desconexão de um armazenamento de chaves personalizado, consulte [Desconectar um armazenamento de AWS CloudHSM chaves](#).

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [EnableKey](#) operação. Para obter informações sobre como ativar e desativar o AWS KMS keys in AWS KMS, consulte.. [Habilitar e desabilitar chaves](#)

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

```

{
    "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma chamada para a [EnableKeyRotation](#) operação. Para obter um exemplo da entrada de CloudTrail registro que é gravada quando a chave é girada, consulte [RotateKey](#). Para obter informações sobre a alternância de AWS KMS keys, consulte [Girar AWS KMS keys](#).

Note

[rotation-period](#) é um parâmetro opcional de solicitação. Se você não especificar um período de alternância ao habilitar a alternância automática de chaves, o valor padrão será de 365 dias.

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

```

{
    "eventVersion": "1.05",

```

```

    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::111122223333:user/Alice",
      "accountId": "111122223333",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2020-07-25T23:41:56Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "EnableKeyRotation",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "rotationPeriodInDays": 180
    },
    "responseElements": {
      "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "81f5b794-452b-4d6a-932b-68c188165273",
    "eventID": "fefc43a7-8e06-419f-bcab-b3bf18d6a401",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
}

```

Encrypt

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a operação [Criptografar](#).

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [GenerateDataKey](#) operação.

```
{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [GenerateDataKeyPair](#) operação. Este exemplo registra uma operação que gera um par de chaves RSA criptografado com uma AWS KMS key de criptografia simétrica.

```

{
    "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [GenerateDataKeyPairWithoutPlaintext](#) operação. Este exemplo registra uma operação que gera um par de chaves RSA criptografado com uma AWS KMS key de criptografia simétrica.

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [GenerateDataKeyWithoutPlaintext](#) operação.

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [GenerateMac](#) operação.

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [GenerateRandom](#) operação. Como essa operação não usa um AWS KMS key, o `resources` campo está vazio.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateRandom",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
  "readOnly": true,
  "resources": [],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

GetKeyPolicy

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [GetKeyPolicy](#) operação. Para obter informações sobre como visualizar a política de chaves de uma chave do KMS, consulte [Visualizar uma política de chave](#).

```
{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [GetKeyRotationStatus](#) operação. Para obter informações sobre alternância automática e sob demanda de material de chave para uma chave do KMS, consulte [Girar 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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada quando você usa a [GetParametersForImport](#) operação. Essa operação retorna a chave pública e o token de importação usados ao importar material de chave em uma chave do KMS. A mesma CloudTrail entrada é registrada quando você usa a `GetParametersForImport` operação ou usa o AWS KMS console para [baixar a chave pública e o token de importação](#).

```
{
```



```

    "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada quando você usa a [ImportKeyMaterial](#) operação. A mesma CloudTrail entrada é registrada quando você usa a ImportKeyMaterial operação ou usa o AWS KMS console para [importar material chave](#) para um AWS KMS key.

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

```
{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [ListAliases](#) operação. Como essa operação não usa nenhum alias específico ou AWS KMS key, o `resources` campo está vazio. Para obter informações sobre a visualização de aliases em AWS KMS, consulte [Encontrar o nome e o ARN do alias para uma chave do 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":
"eyJiIjoieWxwYXNvZTU0Y2MxOTMtYTMwNC00YzEwLTliZWItYTJjZjA3NjA2OTJhIiwieYSiOi6ImFsaWFzL2U1NGNmTkZL",
  },
  "responseElements": null,
  "requestID": "bfe6c190-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "a27dda7b-76f1-4ac3-8b40-42dfba77bcd6",

```

```
"readOnly": true,
"resources": [],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}
```

ListGrants

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [ListGrant](#) operação. Para obter informações sobre subsídios em AWS KMS, consulte [Subsídios em 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":
      "eyJncmFudElkIjoimWY4M2U2ZmM0YTY2NDgxYjQ2Yzc4MTdhM2Y4YmQwMDFKZDNiYmQ1MGVlYTMyY2RmOWFiNWY1Nzcz1N\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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [ListKeyRotations](#) operação. Para obter informações sobre alternância automática e sob demanda de material de chave para uma chave do KMS, consulte [Girar 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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [PutKeyPolicy](#) operação. Para obter informações sobre como atualizar uma política de chaves, consulte [Alterar uma política de chave](#).

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [ReEncrypt](#) operação. O `resources` campo nessa entrada de registro especifica duas AWS KMS keys, a chave KMS de origem e a chave KMS de destino, nessa ordem.

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [ReplicateKey](#) operação. Uma ReplicateKey solicitação resulta em uma ReplicateKey operação e em uma [CreateKey](#) operação.

Para informações sobre como replicar chaves de várias regiões, consulte [Criar chaves de réplica de várias regiões](#).

```
{
  "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    \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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [RetireGrant](#) operação. Para obter informações sobre a retirada de concessões, consulte [Retirar e revogar concessões](#).

```
{  
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [RevokeGrant](#) operação. Para obter informações sobre a revogação de concessões, consulte [Retirar e revogar concessões](#).

```
{  
  "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

Esses exemplos mostram as entradas de AWS CloudTrail registro das operações que giram. AWS KMS keys Para informações sobre como alternar chaves do KMS, consulte [Girar AWS KMS keys](#).

O exemplo a seguir mostra uma entrada de CloudTrail registro para a operação que gira uma chave KMS de criptografia simétrica na qual a rotação automática de chaves está ativada. Para obter informações sobre como habilitar a alternância automática, consulte [Girar AWS KMS keys](#).

Para obter um exemplo da entrada de CloudTrail registro que registra a EnableKeyRotation operação, consulte [EnableKeyRotation](#).

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2025-05-20T20:44:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "RotateKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsServiceEvent",
```

```

    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
      "rotationType": "AUTOMATIC",
      "keyOrigin": "AWS_KMS",
      "previousKeyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
      "currentKeyMaterialId":
"96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "eventCategory": "Management"
  }
}

```

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma rotação sob demanda iniciada pela [RotateKeyOnDemand](#) operação. Para obter informações sobre o rodízio de chaves do KMS de criptografia simétrica sob demanda, consulte [Executar alternância de chaves sob demanda](#).

Para obter um exemplo da entrada de CloudTrail registro que registra a RotateKeyOnDemand operação, consulte [RotateKeyOnDemand](#).

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2025-05-20T20:44:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "RotateKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ]
}

```

```

    }
  ],
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "rotationType": "ON_DEMAND",
    "keyOrigin": "EXTERNAL",
    "previousKeyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
    "currentKeyMaterialId":
"96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "eventCategory": "Management"
}

```

RotateKeyOnDemand

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [RotateKeyOnDemand](#) operação. Para obter um exemplo da entrada de CloudTrail registro que é gravada quando a chave é girada, consulte [RotateKey](#). Para obter mais informações sobre alternância sob demanda de material de chave para uma chave do KMS, consulte [Executar alternância de chaves sob demanda](#).

```

{
  "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

Esses exemplos mostram entradas de AWS CloudTrail registro da [ScheduleKeyDeletion](#) operação.

Para obter um exemplo da entrada de CloudTrail registro que é gravada quando a chave é excluída, consulte [DeleteKey](#). Para obter informações sobre como excluir AWS KMS keys, consulte [Excluir um AWS KMS key](#).

O exemplo a seguir registra uma solicitação ScheduleKeyDeletion para uma chave do KMS de região única.

```

{
  "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"
}

```

O exemplo a seguir registra uma solicitação `ScheduleKeyDeletion` para uma chave do KMS de várias regiões.

Como AWS KMS não excluirá uma chave multirregional até que todas as chaves de réplica sejam excluídas, no `responseElements` campo, o `keyState` é `PendingReplicaDeletion` e o `deletionDate` campo é omitido.

```
{
```

```

"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"
}

```

O exemplo a seguir registra uma `ScheduleKeyDeletion` solicitação de uma chave KMS em um [armazenamento de chaves AWS CloudHSM personalizadas](#).

```
{
  "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

Esses exemplos mostram entradas de AWS CloudTrail registro para a operação [Sign](#).

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma operação de [assinatura](#) que usa uma chave RSA KMS assimétrica para gerar uma assinatura digital para um arquivo.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-03-07T22:36:44Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Sign",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "messageType": "RAW",
    "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
  },
  "responseElements": null,
  "requestID": "8d0b35e0-46cf-48b9-be99-bf2ebc9ab9fb",
  "eventID": "107b3cac-b125-4556-9702-12a2b9afc7f7",
  "readOnly": true,
  "resources": [

```

```
{
  "accountId": "111122223333",
  "type": "AWS::KMS::Key",
  "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
},
{
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```

SynchronizeMultiRegionKey

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao AWS KMS sincronizar uma chave [multirregional](#). A sincronização envolve chamadas entre regiões para copiar as [propriedades compartilhadas](#) de uma chave primária multirregional para suas chaves de réplica. AWS KMS sincroniza chaves multirregionais periodicamente para garantir que todas as chaves multirregionais relacionadas tenham o mesmo material de chave.

O `resources` elemento da entrada de CloudTrail registro inclui o ARN da chave primária multirregional, incluindo sua. Região da AWS As chaves de réplica de várias regiões relacionadas e suas regiões não estão listadas nessa entrada de log.

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

```
{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma chamada para a [TagResource](#) operação para adicionar uma tag com uma chave de tag de Department e um valor de tag de IT.

Para obter um exemplo de uma entrada de UntagResource CloudTrail registro que é gravada quando a chave é girada, consulte [UntagResource](#). Para obter informações sobre marcação AWS KMS keys, consulte [Etiquetas em 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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma chamada para a [UntagResource](#) operação para excluir uma tag com uma chave de tag de Dept.

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

Para obter um exemplo de uma entrada de TagResource CloudTrail registro, consulte [TagResource](#). Para informações sobre marcação de AWS KMS keys, consulte [Etiquetas em 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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [UpdateAlias](#) operação. O elemento `resources` inclui campos para o alias e recursos de chaves do KMS. Para obter informações sobre a criação de aliases em AWS KMS, consulte [Criar alias](#).

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

```
{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [UpdateCustomKeyStore](#) operação para atualizar o ID do cluster para um armazenamento de chaves personalizado. Para obter informações sobre edição de armazenamentos de chaves personalizados, consulte [Editar configurações de armazenamento de AWS CloudHSM chaves](#).

```
{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro gerada ao chamar a [UpdateKeyDescription](#) operação.

```

{
  "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

O exemplo a seguir mostra as entradas de AWS CloudTrail registro que são geradas ao chamar a [UpdatePrimaryRegion](#) operação em uma [chave multirregional](#).

A UpdatePrimaryRegion operação grava duas entradas de CloudTrail registro: uma na região com a chave primária multirregional que é convertida em uma chave de réplica e outra na região com uma chave de réplica multirregional que é convertida em uma chave primária.

CloudTrail as entradas de registro dessa operação registradas em ou após dezembro de 2022 incluem o ARN da chave KMS afetada no `responseElements.keyId` valor, mesmo que essa operação não retorne o ARN da chave.

O exemplo a seguir mostra uma entrada de CloudTrail registro na região UpdatePrimaryRegion em que a chave multirregional mudou de uma chave primária para uma chave de réplica (us-west-2). O campo `primaryRegion` mostra a região que agora hospeda a chave primária (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"
}

```

O exemplo a seguir representa a entrada de CloudTrail registro da região UpdatePrimaryRegion em que a chave multirregional mudou de uma chave de réplica para uma chave primária (ap-northeast-1). Essa entrada de log não identifica a região primária anterior.

```

{
  "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

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro para a [VerifyMac](#) operação.

```

{
    "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"
}

```

Verificar

Esses exemplos mostram entradas de AWS CloudTrail registro para a operação [Verify](#).

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma operação de [verificação](#) que usa uma chave RSA KMS assimétrica para verificar uma assinatura 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 , exemplo um

O exemplo a seguir registra um diretor do IAM criando um volume criptografado usando a chave de volume padrão no console EC2 de gerenciamento da Amazon.

O exemplo a seguir mostra uma entrada de CloudTrail registro na qual a usuária Alice cria um volume criptografado com uma chave de volume padrão no console EC2 de gerenciamento da Amazon. O EC2 registro do arquivo de log inclui um `volumeId` campo com um valor de `vol-13439757`. O AWS KMS registro contém um `encryptionContext` campo com um valor de `aws:ebs:id: "vol-13439757"`. Da mesma forma, o `principalId` e o `accountId` entre os dois registros coincidem. Os registros refletem o fato de que a criação de um volume criptografado gera uma chave de dados que é usada para criptografar o conteúdo do 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 , exemplo dois

No exemplo a seguir, um diretor do IAM executando uma EC2 instância da Amazon cria e monta um volume de dados criptografado sob uma chave KMS. Essa ação gera vários registros de CloudTrail log. Para obter mais informações sobre o Amazon EBS e criptografia, consulte [Requirements for Amazon EBS encryption](#).

Quando o volume é criado, a Amazon EC2, agindo em nome do cliente, obtém uma chave de dados criptografada de AWS KMS (GenerateDataKeyWithoutPlaintext). Depois, ele cria uma concessão (CreateGrant) que permite descriptografar a chave de dados. Quando o volume é montado, a Amazon EC2 liga AWS KMS para descriptografar a chave de dados (). Decrypt

A EC2 instância `instanceId` da Amazon, "i-81e2f56c", aparece no `RunInstances` evento. O mesmo ID de instância qualifica o `granteePrincipal` da concessão que é criada

("111122223333:aws:ec2-infrastructure:i-81e2f56c") e a função assumida que é o principal na chamada de Decrypt("arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/i-81e2f56c").

O [ARN da chave](#) KMS que protege o volume de dados,arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab, aparece em todas as três AWS KMS chamadas (CreateGrantGenerateDataKeyWithoutPlaintext, e). Decrypt

```
{
  "Records": [
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
      },
      "eventTime": "2014-11-05T21:35:27Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "RunInstances",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "instancesSet": {
          "items": [
            {
              "imageId": "ami-b66ed3de",
              "minCount": 1,
              "maxCount": 1
            }
          ]
        },
        "groupSet": {
          "items": [
            {
              "groupId": "sg-98b6e0f2"
            }
          ]
        }
      }
    },
  ],
}
```

```
"instanceType": "m3.medium",
"blockDeviceMapping": {
  "items": [
    {
      "deviceName": "/dev/xvda",
      "ebs": {
        "volumeSize": 8,
        "deleteOnTermination": true,
        "volumeType": "gp2"
      }
    },
    {
      "deviceName": "/dev/sdb",
      "ebs": {
        "volumeSize": 8,
        "deleteOnTermination": false,
        "volumeType": "gp2",
        "encrypted": true
      }
    }
  ]
},
"monitoring": {
  "enabled": false
},
"disableApiTermination": false,
"instanceInitiatedShutdownBehavior": "stop",
"clientToken": "XdKUT141516171819",
"ebsOptimized": false
},
"responseElements": {
  "reservationId": "r-5ebc9f74",
  "ownerId": "111122223333",
  "groupSet": {
    "items": [
      {
        "groupId": "sg-98b6e0f2",
        "groupName": "launch-wizard-2"
      }
    ]
  }
},
"instancesSet": {
  "items": [
    {
```

```
"instanceId": "i-81e2f56c",
"imageId": "ami-b66ed3de",
"instanceState": {
  "code": 0,
  "name": "pending"
},
"amiLaunchIndex": 0,
"productCodes": {

},
"instanceType": "m3.medium",
"launchTime": 1415223328000,
"placement": {
  "availabilityZone": "us-east-1a",
  "tenancy": "default"
},
"monitoring": {
  "state": "disabled"
},
"stateReason": {
  "code": "pending",
  "message": "pending"
},
"architecture": "x86_64",
"rootDeviceType": "ebs",
"rootDeviceName": "/dev/xvda",
"blockDeviceMapping": {

},
"virtualizationType": "hvm",
"hypervisor": "xen",
"clientToken": "XdKUT1415223327917",
"groupSet": {
  "items": [
    {
      "groupId": "sg-98b6e0f2",
      "groupName": "launch-wizard-2"
    }
  ]
},
"networkInterfaceSet": {

},
"ebsOptimized": false
```

```

    }
  ]
}
},
"requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
"eventID": "cd75a605-2fee-4fda-b847-9c3d330ebaae",
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
},
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-05T21:35:35Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "constraints": {
      "encryptionContextSubset": {
        "aws:ebs:id": "vol-f67bafb2"
      }
    },
    "granteePrincipal": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": {
    "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
  },
  "requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
  "eventID": "c1ad79e3-0d3f-402a-b119-d5c31d7c6a6c",
  "readOnly": false,
  "resources": [
    {

```

```

      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
},
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-05T21:35:32Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyWithoutPlaintext",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "encryptionContext": {
      "aws:ebs:id": "vol-f67bafb2"
    }
  },
  "numberOfBytes": 64,
  "keyId": "alias/aws/ebs"
},
  "responseElements": null,
  "requestID": "create-111122223333-758247346-1415223332",
  "eventID": "ac3cab10-ce93-4953-9d62-0b6e5cba651d",
  "readOnly": true,
  "resources": [
    {
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

```

    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
        "arn": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-81e2f56c",
        "accountId": "111122223333",
        "accessKeyId": "",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2014-11-05T21:35:38Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "111122223333:aws:ec2-infrastructure",
            "arn": "arn:aws:iam::111122223333:role/aws:ec2-infrastructure",
            "accountId": "111122223333",
            "userName": "aws:ec2-infrastructure"
          }
        }
      },
      "eventTime": "2014-11-05T21:35:47Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "Decrypt",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-f67bafb2"
        }
      },
      "responseElements": null,
      "requestID": "b4b27883-6533-11e4-b4d9-751f1761e9e5",
      "eventID": "edb65380-0a3e-4123-bbc8-3d1b7cff49b0",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ]
    }
  ]
}

```



```
    ],  
    "eventType": "AwsApiCall",  
    "recipientAccountId": "111122223333"  
  }  
]  
}
```

Monitore as chaves KMS com a Amazon CloudWatch

Você pode monitorar seu AWS KMS keys uso [da Amazon CloudWatch](#), um AWS serviço que coleta e processa dados brutos AWS KMS em métricas legíveis e quase em tempo real. Esses dados são registrados por um período de duas semanas, para que você possa acessar informações históricas e obter um melhor entendimento do uso das suas chaves do KMS e das suas alterações ao longo do tempo.

Você pode usar CloudWatch a Amazon para alertá-lo sobre eventos importantes, como os seguintes.

- O material da chave importada em uma chave do KMS está próximo da data de validade.
- Uma chave do KMS com exclusão pendente ainda está sendo usada.
- O material de chave em uma chave do KMS foi alternado automaticamente.
- Uma chave do KMS foi excluída.

Você também pode criar um CloudWatch alarme [da Amazon](#) que o alerta quando sua taxa de solicitação atingir uma determinada porcentagem do valor da cota. Para obter detalhes, consulte [Gerenciar suas taxas de solicitação de AWS KMS API usando Service Quotas e Amazon CloudWatch](#) no Blog de AWS Segurança.

AWS KMS métricas e dimensões

AWS KMS predefine CloudWatch as métricas da Amazon para facilitar o monitoramento de dados críticos e a criação de alarmes. Você pode visualizar as AWS KMS métricas usando a Console de gerenciamento da AWS e a CloudWatch API da Amazon.

Esta seção lista cada AWS KMS métrica e as dimensões de cada métrica e fornece algumas orientações básicas para criar CloudWatch alarmes com base nessas métricas e dimensões.

Note

Nome do grupo de dimensões:

Para visualizar uma métrica no CloudWatch console da Amazon, na seção Métricas, selecione o nome do grupo de dimensões. Em seguida, é possível filtrar pelo nome da métrica. O tópico inclui o nome da métrica e o nome do grupo de dimensões para cada métrica do AWS KMS .

Você pode visualizar AWS KMS métricas usando a Console de gerenciamento da AWS e a CloudWatch API da Amazon. Para obter mais informações, consulte [Visualizar métricas disponíveis](#) no Guia do CloudWatch usuário da Amazon.

Tópicos

- [SuccessfulRequest](#)
- [SecondsUntilKeyMaterialExpiration](#)
- [Nuvem HSMKey StoreThrottle](#)
- [ExternalKeyStoreThrottle](#)
- [XksProxyCertificateDaysToExpire](#)
- [XksProxyCredentialAge](#)
- [XksProxyErrors](#)
- [XksExternalKeyManagerStates](#)
- [XksProxyLatency](#)

SuccessfulRequest

O número de solicitações de operações criptográficas bem-sucedidas em uma determinada chave do KMS. Ao usar a `SuccessfulRequest` métrica, você pode aplicar a filtragem em nível de chave ao uso AWS KMS da API em. CloudWatch A estatística Sum para essa métrica define o número total de solicitações bem-sucedidas durante o período.

Use essa métrica para identificar quais chaves do KMS consomem a maior parte da sua cota de solicitações ou contribuem mais para os custos de API. Você também pode criar um CloudWatch alarme com base na `SuccessfulRequest` métrica para notificá-lo sobre padrões anômalos de uso AWS KMS da API. Esses alertas podem ajudar a identificar fluxos de trabalho ineficientes que podem exceder inadvertidamente suas cotas de solicitações ou incorrer em custos inesperados.

Dimensões para **SuccessfulRequest**

Dimensão	Description
KeyArn	Valor para cada chave do KMS.
Operation	Valor para cada operação de AWS KMS API. Essa métrica se aplica somente a operações criptográficas.

Para [ReEncrypt](#) operações, a `SuccessfulRequest` métrica inclui dimensões para as chaves KMS de origem e de destino.

Dimensão	Description
SourceKeyArn	Valor da chave do KMS que descriptografou o texto cifrado.
DestinationKeyArn	Valor da chave do KMS que recriptografou os dados.
Operation	Valor para cada operação de AWS KMS API, nesse caso, <code>ReEncrypt</code> .

SecondsUntilKeyMaterialExpiration

O número de segundos restantes até a primeira data de expiração de [material de chave importado](#) em uma chave do KMS. Essa métrica só é válida para chaves do KMS com material de chave importado (uma [origem de material de chave](#) de `EXTERNAL`) e data de validade.

Use essa métrica para acompanhar o tempo restante até a primeira data de expiração de material de chave importada. Quando esse tempo for menos que um limite que você define, reimporte o material de chave com uma nova data de expiração para manter uma chave do KMS utilizável. A métrica `SecondsUntilKeyMaterialExpiration` é específica para uma chave do KMS. Você não pode usar essa métrica para monitorar várias chaves do KMS ou chaves do KMS que você possa vir a criar futuramente. Para obter ajuda na criação de um CloudWatch alarme para monitorar essa métrica, consulte [Crie um CloudWatch alarme para expiração do material chave importado](#).

A estatística mais útil para essa métrica é `Minimum`, que informa o menor tempo restante para todos os pontos de dados no período estatístico especificado. A única unidade válida para essa métrica é `Seconds`.

Nome do grupo de dimensões: métricas por chave

Dimensões do **SecondsUntilKeyMaterialExpiration**

Dimensão	Descrição; relacionado a AWS
KeyId	Valor para cada chave do KMS.

Ao [programar a exclusão](#) de uma chave do KMS, o AWS KMS impõe um período de espera antes de excluir a chave do KMS. É possível usar o período de espera para garantir que não precisa mais da chave do KMS agora ou no futuro. Você também pode configurar um CloudWatch alarme para avisá-lo se uma pessoa ou aplicativo tentar usar a chave KMS em uma [operação criptográfica](#) durante o período de espera. Se você receber uma notificação de um alarme desse tipo, poderá cancelar a exclusão da chave do KMS.

Para instruções, consulte [Criar um alarme que detecte o uso de uma chave do KMS com exclusão pendente](#).

Nuvem HSMKey StoreThrottle

O número de solicitações de operações criptográficas em chaves KMS em cada AWS CloudHSM armazenamento de chaves que acelera AWS KMS (responde com a). ThrottlingException Essa métrica se aplica somente às AWS CloudHSM principais lojas.

A CloudHSMKeyStoreThrottle métrica se aplica somente às chaves KMS em um armazenamento de AWS CloudHSM chaves e somente às solicitações de operações [criptográficas](#). AWS KMS [limita essas solicitações quando a taxa de solicitações](#) excede a [cota de solicitações de armazenamento de chaves personalizadas](#) para seu AWS CloudHSM armazenamento de chaves. Essa métrica também inclui a limitação pelo AWS CloudHSM cluster.

Nome do grupo de dimensões: métrica de controle de utilização de armazenamento de chaves

Dimensão	Description
CustomKeyStoreId	Valor para cada armazenamento de AWS CloudHSM chaves.

Dimensão	Description
KmsOperation	Valor para cada operação de AWS KMS API. Essa métrica se aplica somente às operações de criptografia em chaves do KMS em um repositório de chaves do AWS CloudHSM .
KeySpec	Valor para cada tipo de chave do KMS. A única especificação de chave compatível com chaves KMS em um armazenamento de chaves é AWS CloudHSM SYMMETRIC_DEFAULT.

ExternalKeyStoreThrottle

O número de solicitações de operações criptográficas em chaves KMS em cada armazenamento de chaves externo que acelera AWS KMS (responde com a). `ThrottlingException` Essa métrica se aplica apenas aos [armazenamentos de chaves externas](#).

A `ExternalKeyStoreThrottle` métrica se aplica somente às chaves KMS em um armazenamento de chaves externo e somente às solicitações de operações [criptográficas](#). AWS KMS [limita essas solicitações](#) quando a taxa de solicitação excede a [cota de solicitações de armazenamento de chaves personalizadas](#) para seu armazenamento de chaves externo. Essa métrica não inclui o controle de utilização por seu proxy de armazenamento de chaves externas ou gerenciador de chaves externas.

Use essa métrica para revisar e ajustar o valor da cota de solicitações ao repositório de chaves personalizado. Se essa métrica indicar que AWS KMS está frequentemente limitando suas solicitações dessas chaves KMS, considere solicitar um aumento no valor da cota de solicitação do armazenamento de chaves personalizadas. Para obter ajuda, consulte [Requesting a quota increase](#) (Solicitar um aumento de cota) no Guia do usuário do Service Quotas.

Se você estiver recebendo erros `KMSInvalidStateException` com muita frequência com uma mensagem que explica que a solicitação foi rejeitada “due to a very high request rate” (devido a uma taxa de solicitação muito alta) ou que a solicitação foi rejeitada “because the external key store proxy did not respond in time” (porque o proxy de armazenamento de chaves externas não respondeu a tempo), isso pode indicar que o gerenciador de chaves externas ou o proxy de armazenamento de chaves externas não consegue acompanhar a taxa de solicitação atual. Se possível, reduza a taxa de solicitação. Considere também solicitar uma redução no valor da cota de solicitação de armazenamento de chaves personalizado. Diminuir esse valor da cota pode aumentar a limitação (e o valor da `ExternalKeyStoreThrottle` métrica), mas indica que AWS KMS está rejeitando

solicitações em excesso rapidamente antes de serem enviadas ao proxy externo do armazenamento de chaves ou ao gerenciador de chaves externo. Para solicitar uma redução de cota, acesse o [AWS Support Center](#) e crie um caso.

Nome do grupo de dimensões: métrica de controle de utilização de armazenamento de chaves

Dimensão	Description
CustomKeyStoreId	Valor para cada armazenamento de chaves externas.
KmsOperation	Valor para cada operação de AWS KMS API. Essa métrica se aplica somente às operações de criptografia em chaves do KMS em um repositório de chaves externo.
KeySpec	Valor para cada tipo de chave do KMS. A única especificação de chave compatível com chaves do KMS em um armazenamento de chaves externas é SYMMETRIC_DEFAULT.

XksProxyCertificateDaysToExpire

O número de dias até o certificado TLS do endpoint do [proxy de armazenamento de chaves externas](#) (XksProxyUriEndpoint) expirar. Essa métrica se aplica apenas aos [armazenamentos de chaves externas](#).

Use essa métrica para criar um CloudWatch alarme que o notifique sobre a próxima expiração do seu certificado TLS. Quando o certificado expira, AWS KMS não é possível se comunicar com o proxy externo do armazenamento de chaves. Todos os dados protegidos por chaves do KMS em seu armazenamento de chaves externas ficarão inacessíveis até você renovar o certificado.

Um alarme do certificado evita que a expiração do certificado impeça você de acessar os recursos criptografados. Defina o alarme para dar tempo para a sua organização renovar o certificado antes que ele expire.

Nome do grupo de dimensões: métricas de certificado do proxy XKS

Dimensão	Description
CustomKey StoreId	Valor para cada armazenamento de chaves externas.
Certifica teName	Nome (CN) da entidade no certificado TLS.

Você pode criar CloudWatch alarmes com base nas métricas para armazenamentos de chaves externos e chaves KMS em armazenamentos de chaves externos. Para instruções, consulte [Monitorar repositórios de chaves externas](#).

XksProxyCredentialAge

O número de dias desde que a [credencial de autenticação do proxy](#) (XksProxyAuthenticationCredential) atual do armazenamento de chaves externas foi associada ao armazenamento de chaves externas. Essa contagem começa quando você insere a credencial de autenticação como parte da criação ou atualização do armazenamento de chaves externas. Essa métrica se aplica apenas aos [armazenamentos de chaves externas](#).

Esse valor foi criado para lembrar você sobre a idade da credencial de autenticação. No entanto, como começamos a contagem quando você associa a credencial ao armazenamento de chaves externas, não quando você cria sua credencial de autenticação em seu proxy de armazenamento de chaves externas, isso pode não ser um indicador preciso da idade da credencial no proxy.

Use essa métrica para criar um CloudWatch alarme que lembre você de alternar sua credencial de autenticação de proxy do armazenamento de chaves externo.

Nome do grupo de dimensões: métricas por armazenamento de chaves

Dimensão	Description
CustomKey StoreId	Valor para cada armazenamento de chaves externas.

Você pode criar CloudWatch alarmes com base nas métricas para armazenamentos de chaves externos e chaves KMS em armazenamentos de chaves externos. Para instruções, consulte [Monitorar repositórios de chaves externos](#).

XksProxyErrors

O número de exceções relacionadas às AWS KMS solicitações ao [proxy externo do armazenamento de chaves](#). Essa contagem inclui exceções às quais o proxy externo do armazenamento de chaves retorna AWS KMS e erros de tempo limite que ocorrem quando o proxy do armazenamento de chaves externo não responde AWS KMS dentro do intervalo de tempo limite de 250 milissegundos. Essa métrica se aplica apenas aos [armazenamentos de chaves externas](#).

Use essa métrica para rastrear a taxa de erro das chaves do KMS no armazenamento de chaves externas. Ela revela os erros mais frequentes para que você possa priorizar seus esforços de engenharia. Por exemplo, chaves do KMS que estão gerando altas taxas de erros sem nova tentativa podem indicar um problema com a configuração do armazenamento de chaves externas. Para visualizar a configuração do armazenamento de chaves externas, consulte [Visualizar repositórios de chaves externos](#). Para editar suas configurações de armazenamento de chaves externas, consulte [Editar propriedades do repositório de chaves externo](#).

Nome do grupo de dimensões: métricas de erro do proxy XKS

Dimensão	Description
CustomKeyStoreId	Valor para cada armazenamento de chaves externas.
KmsOperation	Valor de cada operação de AWS KMS API que gerou uma solicitação ao proxy XKS.
XksOperation	Valor para cada operação de API do proxy de armazenamento de chaves externas .
KeySpec	Valor para cada tipo de chave do KMS. A única especificação de chave compatível com chaves do KMS em um armazenamento de chaves externas é SYMMETRIC_DEFAULT.
ErrorType	Valores: Erros com nova tentativa: provavelmente são transitórios, como erros de redes.

Dimensão	Description
	- Erros sem nova tentativa: provavelmente indicam um problema com a configuração do armazenamento de chaves personalizado ou com os componentes externos. - N/A: solicitação bem-sucedida; sem erros
Exception Name	Valores: - Nome da exceção - Nenhum: solicitação bem-sucedida; sem erros

Você pode criar CloudWatch alarmes com base nas métricas para armazenamentos de chaves externos e chaves KMS em armazenamentos de chaves externos. Para instruções, consulte [Monitorar repositórios de chaves externos](#).

XksExternalKeyManagerStates

Uma contagem do número de [instâncias do gerenciador de chaves externas](#) em cada um dos seguintes estados de integridade: `Active`, `Degraded` e `Unavailable`. As informações dessa métrica vêm do proxy de armazenamento de chaves externas associado a cada armazenamento de chaves externas. Essa métrica se aplica apenas aos [armazenamentos de chaves externas](#).

Veja a seguir os estados de integridade das instâncias externas do gerenciador de chaves associadas a um armazenamento de chaves externas. Cada proxy de armazenamento de chaves externas pode usar indicadores diferentes para medir o estado de integridade do gerenciador de chaves externas. Para obter detalhes, consulte a documentação do proxy de armazenamento de chaves externas.

- `Active`: o gerenciador de chaves externas está íntegro.
- `Degraded`: o gerenciador de chaves externas não está íntegro, mas ainda pode atender ao tráfego
- `Unavailable`: o gerenciador de chaves externas não pode atender ao tráfego.

Use essa métrica para criar um CloudWatch alarme que alerta você sobre instâncias do gerenciador de chaves externo degradadas e indisponíveis. Para determinar quais instâncias externas do

gerenciador de chaves estão em cada estado, consulte seus logs de proxy do armazenamento de chaves externas.

Nome do grupo de dimensões: métrica do gerenciados de chaves externas XKS

Dimensão	Description
CustomKeyStoreId	Valor para cada armazenamento de chaves externas.
XksExternalKeyManagerState	Valor para cada estado de integridade.

Você pode criar CloudWatch alarmes com base nas métricas para armazenamentos de chaves externos e chaves KMS em armazenamentos de chaves externos. Para instruções, consulte [Monitorar repositórios de chaves externos](#).

XksProxyLatency

O número de milissegundos necessários para que um proxy de armazenamento de chaves externas responda a uma solicitação do AWS KMS . Se a solicitação atingiu o tempo limite, o valor registrado é o limite de tempo limite de 250 milissegundos. Essa métrica se aplica apenas aos [armazenamentos de chaves externas](#).

Use essas métrica para avaliar a performance do proxy de armazenamento de chaves externas e do gerenciador de chaves externas. Por exemplo, se o proxy estiver frequentemente atingindo o tempo limite nas operações de criptografia e descriptografia, consulte o administrador do proxy externo.

Respostas lentas também podem indicar que seu gerenciador de chaves externo não consegue lidar com o tráfego de solicitações atual. AWS KMS recomenda que seu gerenciador de chaves externo seja capaz de lidar com até 1800 solicitações de operações criptográficas por segundo. Se o gerenciador de chaves externas não conseguir lidar com a taxa de 1800 solicitações por segundo, considere [solicitar uma redução na cota de solicitações de chaves do KMS em um armazenamento de chaves personalizado](#). As solicitações de operações de criptografia que usam as chaves do KMS em seu armazenamento de chaves externas se anteciparão à falha com uma [exceção de controle de utilização](#), em vez de serem processadas e posteriormente rejeitadas pelo proxy de armazenamento de chaves externas ou pelo gerenciador de chaves externas.

Nome do grupo de dimensões: métricas de latência do proxy XKS

Dimensão	Description
CustomKeyStoreId	Valor para cada armazenamento de chaves externas.
KmsOperation	Valor de cada operação de AWS KMS API que gerou uma solicitação ao proxy XKS.
XksOperation	Valor para cada operação de API do proxy de armazenamento de chaves externas .
KeySpec	Valor para cada tipo de chave do KMS. A única especificação de chave compatível com chaves do KMS em um armazenamento de chaves externas é SYMMETRIC_DEFAULT.

Você pode criar CloudWatch alarmes com base nas métricas para armazenamentos de chaves externos e chaves KMS em armazenamentos de chaves externos. Para instruções, consulte [Monitorar repositórios de chaves externas](#).

Crie um CloudWatch alarme para expiração do material chave importado

Você pode criar um CloudWatch alarme que o notifique quando o material de chave importado em uma chave KMS estiver se aproximando do prazo de expiração. Por exemplo, o alarme pode notificá-lo quando faltarem menos de 30 dias para a expiração.

Ao [importar o material de chave para uma chave do KMS](#), é possível especificar opcionalmente uma data e hora quando o material de chave expira. Quando o material da chave expira, ele é AWS KMS excluído e a chave KMS fica inutilizável. Para usar a chave do KMS novamente, você deve [reimportar o material de chave](#). No entanto, se você reimportar o material de chave antes que ele expire, você poderá evitar interromper processos que usem essa chave do KMS.

Esse alarme usa a [SecondsUntilKeyMaterialExpires métrica](#) que AWS KMS publica CloudWatch para chaves KMS com material de chave importado que expira. Cada alarme usa essa métrica para monitorar o material de chave importado para uma chave específica do KMS. Você não pode criar um alarme único para todas as chaves do KMS com material de chave expirado ou um alarme para chaves do KMS que você possa vir a criar futuramente.

Requisitos

Os seguintes recursos são necessários para um CloudWatch alarme que monitora a expiração do material de chave importado.

- Uma chave do KMS com material de chave importado que expira.
- Um tópico do Amazon SNS. Para obter detalhes, consulte o [tópico Criação de um Amazon SNS no Guia CloudWatch](#) do usuário da Amazon.

Criar o alarme

Siga as instruções em [Criar um CloudWatch alarme com base em um limite estático](#) usando os seguintes valores obrigatórios. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Selecionar métrica	Escolha KMS e, em seguida, selecione Per-Key Metrics (Métricas por chave). Selecione a linha com a chave do KMS e a métrica <code>SecondsUntilKeyMaterialExpires</code> . Depois, escolha Select metric (Selecionar métrica). A lista Metrics (Métricas) exibe a métrica <code>SecondsUntilKeyMaterialExpires</code> apenas para chaves do KMS com material de chave importado que expira. Se você não tiver chaves do KMS com essas propriedades na conta e na região, essa lista estará vazia.
Estatística	Mínimo
Período	1 minuto
Tipo de limite	Estático
Sempre que...	Sempre que <i><code>metric-name</code></i> é maior que 1

Crie CloudWatch alarmes para armazenamentos externos de chaves

Você pode criar CloudWatch alarmes da Amazon com base em métricas externas do armazenamento de chaves para notificá-lo quando um valor métrico exceder um limite especificado

por você. O alarme pode enviar a mensagem para um [tópico do Amazon Simple Notification Service \(Amazon SNS\)](#) ou uma [política do Amazon EC2 Auto Scaling](#). Para obter informações detalhadas sobre CloudWatch alarmes, consulte Como [usar CloudWatch alarmes da Amazon no Guia CloudWatch](#) do usuário da Amazon.

Antes de criar um CloudWatch alarme da Amazon, você precisa de um tópico do Amazon SNS. Para obter detalhes, consulte o [tópico Criação de um Amazon SNS no Guia CloudWatch](#) do usuário da Amazon.

Tópicos

- [Criar um alarme para expiração de certificado](#)
- [Criar um alarme para tempo limite de resposta](#)
- [Criar um alarme para erros passíveis de nova tentativa](#)
- [Criar um alarme para erros não passíveis de nova tentativa](#)

Criar um alarme para expiração de certificado

Esse alarme usa a [XksProxyCertificateDaysToExpire](#) métrica AWS KMS publicada para registrar CloudWatch a expiração prevista do certificado TLS associado ao seu endpoint proxy externo de armazenamento de chaves. Você pode criar um alarme único para todos os armazenamentos de chaves externas de sua conta ou um alarme para armazenamentos de chaves externas que você possa criar futuramente.

Recomendamos configurar o alarme para alertar você dez dias antes do prazo de validade do certificado, mas você deve definir o limite que melhor atenda às suas necessidades.

Criar o alarme

Siga as instruções em [Criar um CloudWatch alarme com base em um limite estático](#) usando os seguintes valores obrigatórios. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Selecionar métrica	Escolha KMS e escolha XKS Proxy Certificate Metrics (Métricas de certificado do proxy XKS).

Campo	Valor
	Marque a caixa de seleção ao lado da <code>XksProxyCertificateName</code> que você deseja monitorar. Depois, escolha <code>Select metric</code> (Selecionar métrica).
Estatística	Mínimo
Período	5 minutos
Tipo de limite	Estático
Sempre que...	Sempre que <code>XksProxyCertificateDaysToExpire</code> for Lower do que 10.

Criar um alarme para tempo limite de resposta

Esse alarme usa a [XksProxyLatency](#) métrica que AWS KMS publica CloudWatch para registrar o número de milissegundos necessários para que um proxy externo de armazenamento de chaves responda a uma AWS KMS solicitação. Você pode criar um alarme único para todos os armazenamentos de chaves externas de sua conta ou um alarme para armazenamentos de chaves externas que você possa criar futuramente.

AWS KMS espera que o proxy externo do armazenamento de chaves responda a cada solicitação em 250 milissegundos. Recomendamos configurar um alarme para alertar você quando o proxy de armazenamento de chaves externas levar mais de 200 milissegundos para responder, mas você deve definir o limite que melhor atenda às suas necessidades.

Criar o alarme

Siga as instruções em [Criar um CloudWatch alarme com base em um limite estático](#) usando os seguintes valores obrigatórios. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Selecionar métrica	Escolha KMS e escolha XKS Proxy Latency Metrics (Métricas de latência do proxy XKS).

Campo	Valor
	Marque a caixa de seleção ao lado da <code>KmsOperation</code> que você deseja monitorar. Depois, escolha <code>Select metric</code> (Selecionar métrica).
Estatística	Média
Período	5 minutos
Tipo de limite	Estático
Sempre que...	Sempre que <code>XksProxyLatencyfor Greater</code> do que <code>200</code> .

Criar um alarme para erros passíveis de nova tentativa

Esse alarme usa a [XksProxyErrors](#) métrica que AWS KMS publica CloudWatch para registrar o número de exceções relacionadas às AWS KMS solicitações ao proxy externo do armazenamento de chaves. Você pode criar um alarme único para todos os armazenamentos de chaves externas de sua conta ou um alarme para armazenamentos de chaves externas que você possa criar futuramente.

Erros com nova tentativa diminuirão sua porcentagem de confiabilidade e podem indicar erros de redes. Recomendamos configurar um alarme para alertar você quando mais de cinco erros com nova tentativa forem registrados em um período de um minuto, mas você deve definir o limite que atenda melhor às suas necessidades.

Siga as instruções em [Criar um CloudWatch alarme com base em um limite estático](#) usando os seguintes valores obrigatórios. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Selecionar métrica	Escolha a guia <code>Queries</code> (Consultas). Escolha <code>AWS/KMS</code> para <code>Namespace</code>. Insira <code>SUM(XksProxyErrors)</code> em <code>Metric name</code> (Nome da métrica). Insira <code>ErrorType = Retryable</code> em <code>Filter by</code> (Filtrar por).

Campo	Valor
	Escolha Executar. Depois, escolha Select metric (Selecionar métrica).
Rótulo	<i>Retryable errors</i>
Período	1 minuto
Tipo de limite	Estático
Sempre que...	Sempre que q1 for Greater que 5.

Criar um alarme para erros não passíveis de nova tentativa

Esse alarme usa a [XksProxyErrors](#) métrica que AWS KMS publica CloudWatch para registrar o número de exceções relacionadas às AWS KMS solicitações ao proxy externo do armazenamento de chaves. Você pode criar um alarme único para todos os armazenamentos de chaves externas de sua conta ou um alarme para armazenamentos de chaves externas que você possa criar futuramente.

Erros sem nova tentativa podem indicar um problema com a configuração do armazenamento de chaves externas. Recomendamos configurar um alarme para alertar você quando mais de cinco erros sem nova tentativa forem registrados em um período de um minuto, mas você deve definir o limite que atenda melhor a suas necessidades.

Siga as instruções em [Criar um CloudWatch alarme com base em um limite estático](#) usando os seguintes valores obrigatórios. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Selecionar métrica	Escolha a guia Queries (Consultas). Escolha AWS/KMS para Namespace. Insira SUM(XksProxyErrors) em Metric name (Nome da métrica). Insira ErrorType = Non-retryable em Filter by (Filtrar por). Escolha Executar. Depois, escolha Select metric (Selecionar métrica).

Campo	Valor
Rótulo	<i>Non-retryable errors</i>
Período	1 minuto
Tipo de limite	Estático
Sempre que...	Sempre que q1 for Greater que 5.

Monitore as chaves KMS com a Amazon EventBridge

Você pode usar a Amazon EventBridge (antiga Amazon CloudWatch Events) para alertá-lo sobre os seguintes eventos importantes no ciclo de vida de suas chaves KMS.

- Foi feito o rodízio automático ou sob demanda do material de chave em uma chave do KMS.
- O material de chave importado em uma chave do KMS expirada.
- Uma chave do KMS que havia sido agendada para exclusão foi excluída.

AWS KMS se integra à Amazon EventBridge para notificá-lo sobre eventos importantes que afetam suas chaves KMS. Cada evento é representado em [JSON \(JavaScriptObject Notation\)](#) e inclui o nome do evento, a data e a hora em que o evento ocorreu e os afetados. Você pode coletar esses eventos e estabelecer regras que os encaminhem para um ou mais destinos, como AWS Lambda funções, tópicos do Amazon SNS, filas do Amazon SQS, streams no Amazon Kinesis Data Streams ou destinos integrados.

Para obter mais informações sobre o uso EventBridge com outros tipos de eventos, incluindo aqueles emitidos AWS CloudTrail quando ele registra uma solicitação de read/write API, consulte o [Guia do EventBridge usuário da Amazon](#).

Os tópicos a seguir descrevem os EventBridge eventos AWS KMS gerados.

Alternância de CMKs do KMS

AWS KMS suporta a [rotação automática e sob demanda](#) do material da chave em chaves KMS de criptografia simétrica.

Sempre que AWS KMS gira o material da chave, ele envia um KMS CMK Rotation evento para EventBridge. AWS KMS gera esse evento com base no melhor esforço.

A seguir há um exemplo deste evento.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "KMS CMK Rotation",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "key-origin": "AWS_KMS",
    "rotation-type": "ON_DEMAND",
    "previous-key-material-id":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
    "current-key-material-id":
"96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068"
  }
}
```

Expiração do material de chave importado do KMS

Ao [importar o material de chave em uma chave do KMS](#), é possível especificar opcionalmente em que hora o material de chave expira. Quando o material chave expira, AWS KMS exclui o material chave e envia um KMS Imported Key Material Expiration evento correspondente para EventBridge. AWS KMS gera esse evento com base no melhor esforço.

A seguir há um exemplo deste evento.

```
{
  "version": "0",
  "id": "9da9af57-9253-4406-87cb-7cc400e43465",
  "detail-type": "KMS Imported Key Material Expiration",
  "source": "aws.kms",
  "account": "111122223333",
```

```
{
  "time": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "key-material-id":
      "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  }
}
```

Exclusão da CMK do KMS

Ao [programar a exclusão](#) de uma chave do KMS, o AWS KMS impõe um período de espera antes de excluir a chave do KMS. Após o término do período de espera, AWS KMS exclui a chave KMS e envia um KMS CMK Deletion evento para EventBridge AWS KMS garante esse EventBridge evento. Devido a novas tentativas, ele pode gerar vários eventos dentro de alguns segundos que excluem a mesma chave do KMS.

A seguir há um exemplo deste evento.

```
{
  "version": "0",
  "id": "e9ce3425-7d22-412a-a699-e7a5fc3fbc9a",
  "detail-type": "KMS CMK Deletion",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Aliases em AWS KMS

Um alias é um nome amigável para uma AWS KMS key. Por exemplo, um alias permite fazer referência a uma chave do KMS como test-key em vez de 1234abcd-12ab-34cd-56ef-1234567890ab.

[Você pode usar um alias para identificar uma chave KMS no AWS KMS console, na DescribeKey operação e em operações criptográficas, como Criptografar e. GenerateDataKey](#) Aliases também facilitam o reconhecimento de uma [Chave gerenciada pela AWS](#). Aliases para estas chaves do KMS sempre têm o formato: aws/<service-name>. Por exemplo, o alias Chave gerenciada pela AWS para o Amazon DynamoDB é. aws/dynamodb É possível estabelecer padrões de alias semelhantes para seus projetos, como introduzir o nome de um projeto ou categoria em seus aliases.

Também é possível permitir e negar acesso a chaves do KMS com base em aliases sem precisar editar políticas ou gerenciar concessões. Esse recurso faz parte do AWS KMS suporte ao [controle de acesso baseado em atributos](#) (ABAC). Para obter detalhes, consulte [Usar aliases para controlar o acesso a chaves do KMS](#).

Grande parte do poder dos aliases vem da sua capacidade de alterar a chave do KMS associada a um alias a qualquer momento. Aliases podem tornar seu código mais fácil de escrever e manter. Por exemplo, suponha que você use um alias para fazer referência a uma determinada chave do KMS e queira alterar essa chave do KMS. Nesse caso, basta associar o alias a outra chave do KMS. Você não precisa mudar o código.

Aliases também facilitam a reutilização do mesmo código em Regiões da AWS diferentes. Crie aliases com o mesmo nome em várias regiões e associe cada alias a uma chave do KMS em sua respectiva região. Quando o código é executado em cada região, o alias faz referência à chave do KMS associada nessa região. Para ver um exemplo, consulte [Saiba como usar aliases em suas aplicações](#).

[Você pode criar um alias para uma chave KMS no AWS KMS console, usando a CreateAliasAPI ou usando o AWS::KMS::Alias CloudFormation modelo.](#)

A AWS KMS API fornece controle total dos aliases em cada conta e região. A API inclui operações para criar um alias ([CreateAlias](#)), visualizar nomes de alias e alias ARNs ([ListAliases](#)), alterar a chave KMS associada a um alias () e excluir um alias ([UpdateAlias](#)). [DeleteAlias](#)

Funcionamento dos aliases

Saiba como os aliases funcionam no AWS KMS.

Um alias é um recurso independente AWS

Um alias não é uma propriedade de uma chave do KMS. As ações executadas no alias não afetam a chave do KMS associada. É possível criar um alias para uma chave do KMS e depois atualizá-lo para associá-lo a outra chave do KMS. Você pode até mesmo excluir o alias sem nenhum efeito sobre a chave do KMS associada. No entanto, se você excluir uma chave do KMS, todos os aliases associados a ela serão excluídos.

Se você especificar um alias como o recurso em uma política do IAM, esta fará referência ao alias, e não à chave do KMS associada.

Cada alias tem dois formatos

Ao criar um alias, você especifica o nome do alias. AWS KMS cria o alias ARN para você.

- Um [ARN de alias](#) é um Amazon Resource Name (ARN) que identifica exclusivamente o alias.

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

- Um [nome de alias](#) que é único na conta e na região. Na AWS KMS API, o nome do alias é sempre prefixado por. `alias/` Esse prefixo é omitido no console. AWS KMS

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

Os aliases não são secretos

Os aliases podem ser exibidos em texto simples em CloudTrail registros e outras saídas. Não inclua informações confidenciais ou sigilosas nos nomes dos aliases.

Cada alias é associado a uma chave do KMS por vez

O alias e sua chave do KMS devem estar na mesma conta e região.

Você pode associar um alias a qualquer [chave gerenciada pelo cliente](#) na mesma Conta da AWS região. No entanto, você não tem permissão para associar um alias a uma [Chave gerenciada pela AWS](#).

Por exemplo, essa [ListAliases](#) saída mostra que o test-key alias está associado a exatamente uma chave KMS de destino, que é representada pela TargetKeyId propriedade.

```
{
  "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
}
```

Vários aliases podem ser associados à mesma chave do KMS

Por exemplo, você pode associar os aliases test-key e project-key à mesma chave do KMS.

```
{
  "AliasName": "alias/test-key",
  "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
  "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "CreationDate": 1593622000.191,
  "LastUpdatedDate": 1593622000.191
},
{
  "AliasName": "alias/project-key",
  "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project-key",
  "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "CreationDate": 1516435200.399,
  "LastUpdatedDate": 1516435200.399
}
```

O alias deve ser exclusivo em uma conta e região.

Por exemplo, é possível ter apenas um alias test-key em cada conta e região. Os aliases diferenciam maiúsculas de minúsculas, mas os aliases que diferem apenas no tamanho das letras são muito propensos a erros. Não é possível alterar um nome de alias. No entanto, você pode excluir o alias e criar um novo com o nome desejado.

É possível criar um alias com o mesmo nome em regiões diferentes

Por exemplo, é possível ter um alias finance-key na região Leste dos EUA (Norte da Virgínia) e um alias finance-key na região Europa (Frankfurt). Cada alias seria associado a uma chave

do KMS em sua região. Se o seu código se referir a um nome de alias como `alias/finance-key`, você poderá executá-lo em várias regiões. Em cada região, ele usa uma chave do KMS diferente. Para obter detalhes, consulte [Saiba como usar aliases em suas aplicações](#).

É possível alterar a chave do KMS associada a um alias

Você pode usar a [UpdateAlias](#) operação para associar um alias a uma chave KMS diferente. Por exemplo, se o alias `finance-key` estiver associado à chave do KMS `1234abcd-12ab-34cd-56ef-1234567890ab`, você poderá atualizá-lo para associá-lo à chave do KMS `0987dcba-09fe-87dc-65ba-ab0987654321`.

No entanto, a chave do KMS atual e nova devem ser do mesmo tipo (ambas devem ser simétricas, assimétricas ou HMAC) e devem ter o mesmo [uso de chave](#) (ENCRYPT_DECRYPT ou SIGN_VERIFY or GENERATE_VERIFY_MAC). Essa restrição impede erros no código que usa aliases. Se for necessário associar um alias a um tipo diferente de chave e você tiver atenuado os riscos, poderá excluir e recriar o alias.

Algumas chaves do KMS não têm aliases

Ao criar uma chave KMS no AWS KMS console, você deve dar a ela um novo alias. Mas um alias não é necessário quando você usa a [CreateKey](#) operação para criar uma chave KMS. Além disso, você pode usar a [UpdateAlias](#) operação para alterar a chave KMS associada a um alias e a [DeleteAlias](#) operação para excluir um alias. Como resultado, algumas chaves do KMS podem ter vários aliases, e outras podem não ter nenhum.

AWS cria aliases em sua conta

AWS cria aliases em sua conta para [Chaves gerenciadas pela AWS](#). Esses aliases têm nomes no formato `alias/aws/<service-name>`, como `alias/aws/s3`.

Alguns AWS aliases não têm chave KMS. Esses aliases predefinidos geralmente são associados a um Chave gerenciada pela AWS quando você começa a usar o serviço.

Usar aliases para identificar chaves do KMS

Você pode usar um [nome de alias](#) ou [ARN de alias](#) para identificar uma chave KMS em operações [criptográficas](#), e. [DescribeKeyGetPublicKey](#) (Se a [chave do KMS estiver em uma Conta da AWS diferente](#), você deverá usar seu [ARN de chave](#) ou ARN de alias.) Aliases não são identificadores válidos para chaves do KMS em outras operações do AWS KMS. Para obter informações sobre os [identificadores de chave](#) válidos para cada operação de AWS KMS API, consulte as descrições dos KeyId parâmetros na Referência da AWS Key Management Service API.

Não é possível usar um nome ou um ARN de alias para [identificar uma chave do KMS em uma política do IAM](#). Para controlar o acesso a uma chave KMS com base em seus aliases, use as chaves de condição [kms: RequestAlias](#) ou [kms: ResourceAliases](#). Para obter detalhes, consulte [ABAC para AWS KMS](#).

Controlar o acesso a aliases

Ao criar ou alterar um alias, você afeta o alias e sua chave do KMS associada. Portanto, as entidades principais que gerenciam aliases devem ter permissão para chamar a operação no alias e em todas as chaves do KMS afetadas. É possível fornecer essas permissões usando [políticas de chave](#), [políticas do IAM](#) e [concessões](#).

Note

Tenha cuidado ao conceder permissão a entidades principais para gerenciar etiquetas e aliases. Alterar uma etiqueta ou um alias pode conceder ou negar uma permissão à chave gerenciada pelo cliente. Para obter mais detalhes, consulte [ABAC para AWS KMS](#) e [Usar aliases para controlar o acesso a chaves do KMS](#).

Para obter informações sobre como controlar o acesso a todas AWS KMS as operações, consulte [Referência de permissões](#).

As permissões para criar e gerenciar aliases funcionam da forma a seguir.

kms:CreateAlias

Para criar um alias, a entidade principal precisa das seguintes permissões para o alias e a chave do KMS associada.

- `kms:CreateAlias` para o alias. Forneça essa permissão em uma política do IAM associada à entidade principal que tem permissão para criar o alias.

A instrução de política de exemplo a seguir especifica o alias em um elemento `Resource`. Mas você pode listar vários alias ARNs ou especificar um padrão de alias, como “teste*”. Porém, é possível especificar um valor de `Resource` de “*” para permitir que a entidade principal crie um alias na conta e na região. A permissão para criar um alias também pode ser incluída em uma permissão `kms:Create*` para todos os recursos em uma conta e região.


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

- `kms:CreateAlias` para a chave do KMS. Essa permissão deve ser fornecida em uma política de chave ou em uma política do IAM delegada a partir da política de chaves.

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

Você pode usar chaves de condição para limitar as chaves do KMS que podem ser associadas a um alias. Por exemplo, você pode usar a chave de KeySpec condição [kms:](#) para permitir que o principal crie aliases somente em chaves KMS assimétricas. Para obter uma lista completa de chaves de condições que podem ser usadas para limitar a permissão `kms:CreateAlias` em recursos de chaves do KMS, consulte [AWS KMS permissões](#).

kms:ListAliases

Para listar aliases na conta e na região, a entidade principal deve ter a permissão `kms:ListAliases` em uma política do IAM. Como essa política não está relacionada a uma chave do KMS ou recurso de alias específico, o valor do elemento de recurso na política [deve ser "*"](#).

Por exemplo, a instrução de política do IAM a seguir dá à entidade principal permissão para listar todas as chaves do KMS e aliases na conta e na região.

JSON

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

kms:UpdateAlias

Para alterar a chave do KMS associada a um alias, a entidade principal precisa de três elementos de permissão: um para o alias, um para a chave do KMS atual e outro para a nova chave do KMS.

Por exemplo, suponha que você queira alterar o alias `test-key` da chave do KMS com ID de chave `1234abcd-12ab-34cd-56ef-1234567890ab` para a chave do KMS com ID de chave `0987dcba-09fe-87dc-65ba-ab0987654321`. Nesse caso, inclua declarações de política semelhantes aos exemplos desta seção.

- `kms:UpdateAlias` para o alias. Forneça essa permissão em uma política do IAM associada à entidade principal. A política do IAM a seguir define um alias específico. Mas você pode listar vários alias ARNs ou especificar um padrão de alias, como `"test*"`. Porém, é possível especificar um valor de `Resource` de `"*"` para permitir que a entidade principal atualize um alias na conta e na região.

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

```
}
```

- `kms:UpdateAlias` para a chave do KMS que está atualmente associada ao alias. Essa permissão deve ser fornecida em uma política de chave ou em uma política do IAM delegada a partir da política de chaves.

```
{
  "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` para a chave do KMS que a operação associa ao alias. Essa permissão deve ser fornecida em uma política de chave ou em uma política do IAM delegada a partir da política de chaves.

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

Você pode usar chaves de condição para limitar uma ou ambas as chaves do KMS em uma operação `UpdateAlias`. Por exemplo, você pode usar uma chave de `ResourceAliases` condição [kms:](#) para permitir que o principal atualize os aliases somente quando a chave KMS de destino já tiver um alias específico. Para obter uma lista completa de chaves de condições que podem ser usadas para limitar a permissão `kms:UpdateAlias` em um recurso de chave do KMS, consulte [AWS KMS permissões](#).

kms:DeleteAlias

Para excluir um alias, a entidade principal precisa de permissão para o alias e a chave do KMS associada.

Como sempre, tenha cuidado ao conceder permissão às entidades principais para excluir um recurso. No entanto, a exclusão de um alias não afeta a chave do KMS associada. Embora isso possa causar uma falha em uma aplicação dependente do alias, se você excluir um alias por engano, poderá recriá-lo.

- `kms:DeleteAlias` para o alias. Forneça essa permissão em uma política do IAM associada à entidade principal que tem permissão para excluir o alias.

A declaração de política demonstrativa a seguir especifica o alias em um elemento `Resource`. Mas você pode listar vários alias ARNs ou especificar um padrão de alias, como `"test*"`. Você também pode especificar um `Resource` valor de `"*"` para permitir que o principal exclua qualquer alias na conta e na região.

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

- `kms:DeleteAlias` para a chave do KMS associada. Essa permissão deve ser fornecida em uma política de chave ou em uma política do IAM delegada a partir da política de chaves.

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

Limitar permissões de alias

É possível usar chaves de condição para limitar permissões de alias quando o recurso é uma chave do KMS. Por exemplo, a política do IAM a seguir permite as operações de alias em chaves do KMS em uma determinada conta e região. No entanto, ele usa a chave de KeyOrigin condição [kms:](#) para limitar ainda mais as permissões às chaves KMS com material de chave de. AWS KMS

Para obter uma lista completa de chaves de condições que podem ser usadas para limitar a permissão de alias em um recurso de chave do KMS, consulte [AWS KMS permissões](#).

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

Você não pode usar chaves de condição em uma instrução de política na qual o recurso é um alias. Para limitar os aliases que uma entidade principal pode gerenciar, use o valor da propriedade Resource da instrução de política do IAM que controla o acesso ao alias. Por exemplo, as declarações de política a seguir permitem que o diretor crie, atualize ou exclua qualquer alias na região Conta da AWS e, a menos que o alias comece com. 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*"
}
```

Criar alias

Você pode criar aliases no AWS KMS console ou usando operações de AWS KMS API.

O alias deve ser uma string contendo de 1 até 256 caracteres. Só pode conter caracteres alfanuméricos, barras (/), sublinhados (_) e traços (-). O nome do alias de uma [chave gerenciada pelo cliente](#) não pode começar com `alias/aws/`. O prefixo `alias/aws/` está reservado para a [Chave gerenciada pela AWS](#).

É possível criar um alias para uma nova chave do KMS ou para uma chave do KMS existente. Você pode adicionar um alias para que uma determinada chave do KMS seja usada em um projeto ou uma aplicação.

Você também pode usar um AWS CloudFormation modelo para criar um alias para uma chave KMS. Consulte mais informações em [AWS::KMS::Alias](#) no Guia de Usuário AWS CloudFormation .

Usando o AWS KMS console

Ao [criar uma chave KMS](#) no AWS KMS console, você deve criar um alias para a nova chave KMS. Para criar um alias para uma chave do KMS existente, use a guia Aliases na página de detalhes da chave do KMS.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. Você não pode gerenciar aliases para Chaves gerenciadas pela AWS ou Chaves pertencentes à AWS.
4. Na tabela, escolha o ID de chave ou alias da chave do KMS. Em seguida, na página de detalhes da chave do KMS, escolha a guia Aliases.

Se uma chave do KMS tiver vários aliases, a coluna Aliases na tabela exibe um alias e um resumo de alias, como (Mais n). Escolher o resumo de aliases leva diretamente à guia Aliases na página de detalhes da chave do KMS.

5. Na guia Aliases, escolha Create alias (Criar alias). Insira um nome de alias e escolha Create alias (Criar alias).

Important

Não inclua informações confidenciais ou sigilosas nesse campo. Esse campo pode ser exibido em texto simples em CloudTrail registros e outras saídas.

Note

Não adicione o prefixo alias/. O console adiciona isso para você automaticamente. Se você inserir alias/ExampleAlias, o nome do alias real será alias/alias/ExampleAlias.

Usando a AWS KMS API

Para criar um alias, use a [CreateAlias](#) operação. Diferentemente do processo de criação de chaves KMS no console, a [CreateKey](#) operação não cria um alias para uma nova chave KMS.

Important

Não inclua informações confidenciais ou sigilosas nesse campo. Esse campo pode ser exibido em texto simples em CloudTrail registros e outras saídas.

Você pode usar a operação `CreateAlias` para criar um alias para uma nova chave do KMS sem alias. Você também pode usar a operação `CreateAlias` para adicionar um alias a qualquer chave do KMS existente ou para recriar um alias excluído acidentalmente.

Nas operações da AWS KMS API, o nome do alias deve começar com `alias/` seguido por um nome, como `alias/ExampleAlias`. O alias deve ser exclusivo na conta e na região da . Para encontrar os nomes de alias que já estão em uso, use a [ListAliases](#) operação. O nome do alias diferencia maiúsculas de minúsculas.

O `TargetKeyId` pode ser qualquer [chave gerenciada pelo cliente](#) na mesma Região da AWS. Para identificar a chave do KMS, use seu [ID de chave](#) ou [ARN de chave](#). Não é possível usar outro alias.

O exemplo a seguir cria o alias `example-key` e o associa à chave do KMS especificada. Esses exemplos usam o AWS Command Line Interface (AWS CLI). Para obter exemplos em várias linguagens de programação, consulte [Use CreateAlias com um AWS SDK ou CLI](#).

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

`CreateAlias` não retorna nenhuma saída. Para ver o novo alias, use a operação `ListAliases`. Para obter detalhes, consulte [Usando a AWS KMS API](#).

Encontrar o nome e o ARN do alias para uma chave do KMS

Os aliases facilitam o reconhecimento das chaves KMS no AWS KMS console. Você pode visualizar os aliases de uma chave KMS no AWS KMS console ou usando a [ListAliases](#) operação. A [DescribeKey](#) operação, que retorna as propriedades de uma chave KMS, não inclui aliases.

Os procedimentos a seguir demonstram como visualizar e identificar os aliases associados a uma chave KMS usando o AWS KMS console e AWS KMS a API. Os exemplos de AWS KMS API usam o [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Usando o AWS KMS console

O AWS KMS console exibe os aliases associados à chave KMS.

1. Abra o AWS KMS console em <https://console.aws.amazon.com/kms>.

2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. Para exibir as chaves em sua conta que você cria e gerencia, no painel de navegação, escolha Customer managed keys (Chaves gerenciadas de cliente). Para ver as chaves em sua conta que AWS cria e gerencia para você, no painel de navegação, escolha chaves AWS gerenciadas.
4. A coluna Aliases exibe o alias de cada chave do KMS. Se uma chave do KMS não tiver um alias, um traço (-) será exibido na coluna Aliases.

Se uma chave do KMS tiver vários aliases, a coluna Aliases também terá um resumo dos aliases, como (mais n). Por exemplo, a seguinte chave do KMS tem dois aliases, um dos quais é key-test.

Para encontrar o nome e o ARN de todos os aliases para a chave do KMS, use a guia Aliases.

- Para acessar diretamente a guia Aliases, na coluna Aliases, escolha o resumo do alias (Mais n). Um resumo de aliases será exibido somente se a chave do KMS tiver mais de um alias.
- Outra opção é escolher o alias ou o ID da chave do KMS (que abre a página de detalhes da chave do KMS) e escolher a guia Aliases. As guias estão na seção General configuration (Configuração geral).

Customer managed keys (16)				Key actions ▼	Create key
Filter keys by aliases, key ID, or key type				< 1 2 >	⚙
☐	Aliases ▼	Key ID ▼	Status		
☐	key-test (+1 more)	1234abcd-12ab-34cd-56ef-1234567890ab	Enabled		
☐	-	1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d	Enabled		

5. A guia Aliases exibe o nome do alias e o ARN do alias de todos os aliases de uma chave do KMS. Também é possível criar e excluir aliases para a chave do KMS nessa guia.

Key policy	Cryptographic configuration	Key material	Tags	Public key	Aliases
Aliases Info					Delete Create new alias
Filter by Alias name					< 1 >
☐	Alias name	Alias ARN			
☐	key-test	 arn:aws:kms:us-east-1:111122223333:alias/key-test			
☐	project-key	 arn:aws:kms:us-east-1:111122223333:alias/project-key			

Chaves gerenciadas pela AWS

Você pode usar o alias para reconhecer um Chave gerenciada pela AWS, conforme mostrado nesta Chaves gerenciadas pela AWS página de exemplo. Os aliases de Chaves gerenciadas pela AWS sempre têm o formato: `aws/<service-name>`. Por exemplo, o alias Chave gerenciada pela AWS para o Amazon DynamoDB é. `aws/dynamodb`

AWS managed keys (9)	
Filter keys by alias or key ID	
Alias	▲
aws/dynamodb	
aws/ebs	
aws/lightsail	
aws/rds	
aws/s3	
aws/secretsmanager	
aws/ssm	
aws/workmail	
aws/xray	

Usando a AWS KMS API

A [ListAliases](#) operação retorna o nome do alias e o ARN do alias da conta e da região. A saída inclui aliases para Chaves gerenciadas pela AWS e para chaves gerenciadas pelo cliente. Os aliases para Chaves gerenciadas pela AWS têm o formato `aws/<service-name>`, como `aws/dynamodb`.

A resposta também pode incluir aliases sem campo `TargetKeyId`. Esses são aliases predefinidos que AWS foram criados, mas ainda não foram associados a uma chave 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
  }
]
}

```

Para obter todos os aliases associados a uma determinada chave do KMS, use o parâmetro opcional `KeyId` da operação `ListAliases`. O parâmetro `KeyId` usa o [ID de chave](#) ou o [ARN de chave](#) da chave do KMS.

Esse exemplo obtém todos os aliases associados à chave do KMS `0987dcba-09fe-87dc-65ba-ab0987654321`.

```

$ aws kms list-aliases --key-id 0987dcba-09fe-87dc-65ba-ab0987654321
{
  "Aliases": [
    {
      "AliasName": "alias/access-key",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "CreationDate": "2018-01-20T15:23:10.194000-07:00",
      "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
    },
    {
      "AliasName": "alias/finance-project",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "CreationDate": 1604958290.014,
      "LastUpdatedDate": 1604958290.014
    }
  ]
}

```

O parâmetro `KeyId` não usa caracteres curinga, mas você pode usar os recursos da linguagem de programação para filtrar a resposta.

Por exemplo, o AWS CLI comando a seguir obtém somente os aliases de Chaves gerenciadas pela AWS.

```

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

```

O comando a seguir obtém apenas o alias `access-key`. O nome do alias diferencia maiúsculas de minúsculas.

```
$ 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"
  }
]
```

Atualizar aliases

Como um alias é um recurso independente, é possível alterar a chave do KMS associada a ele. Por exemplo, se o `test-key` alias estiver associado a uma chave KMS, você poderá usar a [UpdateAlias](#) operação para associá-lo a uma chave KMS diferente. Esta é uma das várias maneiras de [alternar manualmente uma chave do KMS](#) sem alterar seu material de chave. Também é possível atualizar uma chave do KMS para que uma aplicação que estava usando uma determinada chave do KMS para novos recursos use agora uma diferente.

Você não pode atualizar um alias no AWS KMS console. Além disso, você não pode usar `UpdateAlias` (nem qualquer outra operação) para alterar um nome de alias. Para alterar um nome de alias, exclua o alias atual e crie um novo para a chave do KMS.

Quando você atualiza um alias, a chave atual do KMS e a nova chave do KMS devem ser do mesmo tipo (ambas simétricas, assimétricas ou HMAC). Elas também devem ter o mesmo uso de chave (`ENCRYPT_DECRYPT` ou `SIGN_VERIFY` ou `GENERATE_VERIFY_MAC`). Essa restrição previne erros criptográficos no código que usa aliases.

O exemplo a seguir começa usando a [ListAliases](#) operação para mostrar que o `test-key` alias está atualmente associado à chave 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
  }
]
```

Em seguida, ele usará a operação `UpdateAlias` para alterar a chave do KMS associada ao alias `test-key` para a chave do KMS `0987dcba-09fe-87dc-65ba-ab0987654321`. Não é necessário especificar a chave do KMS atualmente associada, apenas a nova chave do KMS ("de destino"). O nome do alias diferencia maiúsculas de minúsculas.

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

Para verificar se o alias agora está associado à chave do KMS de destino, use a operação `ListAliases` novamente. Esse AWS CLI comando usa o `--query` parâmetro para obter somente o `test-key` alias. Os campos `TargetKeyId` e `LastUpdatedDate` são atualizados.

```
$ 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
  }
]
```

Excluir um alias

Você pode excluir um alias no AWS KMS console ou usando a [DeleteAlias](#) operação. Antes de excluir um alias, verifique se ele não está em uso. Embora a exclusão de um alias não afete a chave do KMS associada, ela pode criar problemas para qualquer aplicação que use esse alias. Se você excluir um alias por engano, poderá criar um novo alias com o mesmo nome e associá-lo à mesma chave do KMS ou a outra chave do KMS.

Se uma chave do KMS for excluída, todos os aliases associados a ela serão excluídos também.

Usando o AWS KMS console

Para excluir um alias no AWS KMS console, use a guia Aliases na página de detalhes da chave KMS. É possível excluir vários aliases de uma chave do KMS ao mesmo tempo.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. Você não pode gerenciar aliases para Chaves gerenciadas pela AWS ou Chaves pertencentes à AWS.
4. Na tabela, escolha o ID de chave ou alias da chave do KMS. Em seguida, na página de detalhes da chave do KMS, escolha a guia Aliases.

Se uma chave do KMS tiver vários aliases, a coluna Aliases na tabela exibe um alias e um resumo de alias, como (Mais n). Escolher o resumo de aliases leva diretamente à guia Aliases na página de detalhes da chave do KMS.

5. Na guia Aliases, marque a caixa de seleção ao lado dos aliases que deseja excluir. Escolha Excluir.

Usando a AWS KMS API

Para excluir um alias, use a [DeleteAlias](#) operação. Essa operação exclui um alias por vez. O nome do alias faz distinção entre maiúsculas e minúsculas e deve ser precedido pelo prefixo `alias/`.

Por exemplo, o comando a seguir exclui o alias chamado `test-key`. Esse comando não retorna nenhuma saída.

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

Para verificar se o alias foi excluído, use a [ListAliases](#) operação. O comando a seguir usa o `--query` parâmetro no AWS CLI para obter somente o `test-key` alias. Os colchetes vazios na resposta indicam que a resposta `ListAliases` não incluiu um alias `test-key`. Para eliminar os colchetes, use o parâmetro `--output text` e o valor.

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

Usar aliases para controlar o acesso a chaves do KMS

Você pode controlar o acesso às chaves do KMS de acordo com os aliases associados à chave do KMS. Para fazer isso, use as teclas de ResourceAliases condição [kms: RequestAlias](#) e [kms:](#). Esse recurso faz parte do AWS KMS suporte ao [controle de acesso baseado em atributos](#) (ABAC).

A chave de condição `kms:RequestAlias` permite ou nega acesso a uma chave do KMS de acordo com o alias em uma solicitação. A chave de condição `kms:ResourceAliases` permite ou nega acesso a uma chave do KMS de acordo com os aliases associados à chave do KMS.

Esses recursos não permitem identificar uma chave do KMS usando um alias no elemento `resource` de uma instrução de política. Quando um alias é o valor de um elemento `resource`, a política aplica-se ao recurso de alias, e não a qualquer chave do KMS que possa estar associada a ela.

Note

Pode levar até cinco minutos para que alterações de etiqueta e alias afetem a autorização de chaves do KMS. Alterações recentes podem estar visíveis em operações de API antes de afetarem a autorização.

Ao usar aliases para controlar o acesso a chaves do KMS, considere o seguinte:

- Use aliases para reforçar as práticas recomendadas de [acesso com privilégio mínimo](#). Conceda às entidades principais do IAM somente as permissões de que eles precisam para as chaves do KMS que elas devem usar ou gerenciar. Por exemplo, use aliases para identificar as chaves do KMS usadas para um projeto. Em seguida, conceda permissão à equipe do projeto para usar apenas chaves do KMS com os aliases do projeto.
- Tenha cautela ao conceder às entidades principais as permissões `kms:CreateAlias`, `kms:UpdateAlias` ou `kms>DeleteAlias`, que permitem adicionar, editar e excluir aliases. Quando você usa aliases para controlar o acesso a chaves do KMS, a alteração de um alias pode conceder às entidades principais permissão para usar chaves do KMS para as quais, de outra forma, eles não teriam permissão para usar. Ele também pode negar acesso a chaves do KMS que outras entidades principais exigem para realizar seus trabalhos.
- Revise os diretores Conta da AWS que atualmente têm permissão para gerenciar aliases e ajustar as permissões, se necessário. Os administradores de chaves que não têm permissão para alterar

políticas de chave ou criar concessões podem controlar o acesso a chaves do KMS quando têm a devida permissão para gerenciar aliases.

Por exemplo, a [política de chaves padrão para administradores de chaves](#) do console inclui as permissões `kms:CreateAlias`, `kms>DeleteAlias` e `kms:UpdateAlias`. As políticas do IAM podem dar permissões de alias para todas as chaves do KMS na sua Conta da AWS. Por exemplo, a política [AWSKeyManagementServicePowerUser](#) gerenciada permite que os diretores criem, excluam e listem aliases para todas as chaves do KMS, mas não as atualizem.

- Antes de definir uma política que dependa de um alias, revise os aliases nas chaves KMS em seu. Conta da AWS. Assegure-se de que sua política se aplique somente aos aliases que você pretende incluir. Use [CloudTrail registros](#) e [CloudWatch alarmes](#) para alertá-lo sobre alterações de alias que podem afetar o acesso às suas chaves KMS. Além disso, a [ListAliases](#) resposta inclui a data de criação e a data da última atualização de cada alias.
- As condições de política de alias usam correspondência de padrões. Elas não estão vinculadas a uma instância específica de um alias. Uma política que usa chaves de condição com base em alias afeta todos os aliases novos e existentes que correspondem ao padrão. Se você excluir e recriar um alias que corresponde a uma condição de política, esta última se aplicará ao novo alias e também ao antigo.

A chave de condição `kms:RequestAlias` depende do alias especificado explicitamente em uma solicitação de operação. A chave de condição `kms:ResourceAliases` depende dos aliases associados a uma chave do KMS, mesmo que eles não apareçam na solicitação.

kms:RequestAlias

Permita ou negue o acesso a uma chave do KMS de acordo com o alias que identifica essa chave em uma solicitação. Você pode usar a chave de RequestAlias condição [kms:](#) em uma política de [chaves ou política](#) do IAM. Ela se aplica às operações que usam um alias para identificar uma chave KMS em uma solicitação, ou seja, [operações criptográficas](#), e. [DescribeKeyGetPublicKey](#) Não é válido para operações de alias, como [CreateAlias](#) ou [DeleteAlias](#).

Na chave de condição, especifique um [nome de alias](#) ou um padrão de nome de alias. Você não pode especificar um [ARN de alias](#).

Por exemplo, a seguinte instrução de política de chaves permite que as entidades principais usem as operações especificadas na chave do KMS. A permissão será efetiva apenas quando a solicitação usar um alias que inclui `alpha` para identificar a chave do 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*"
    }
  }
}
```

O seguinte exemplo de solicitação de uma entidade principal autorizada atenderia à condição. No entanto, uma solicitação que usasse um [ID de chave](#), um [ARN de chave](#) ou um alias diferente não atenderia à condição, mesmo que esses valores identificassem a mesma chave do KMS.

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

kms:ResourceAliases

Permita ou negue acesso a uma chave do KMS de acordo com os aliases associados à chave do KMS, mesmo que o alias não seja usado em uma solicitação. A chave de ResourceAliases condição [kms:](#) permite especificar um alias ou padrão de alias, como `alias/test*`, para que você possa usá-la em uma política do IAM para controlar o acesso a várias chaves do KMS na mesma região. É válido para qualquer AWS KMS operação que use uma chave KMS.

Por exemplo, a política do IAM a seguir permite que os diretores chamem as operações especificadas nas chaves KMS em duas. Contas da AWS No entanto, a permissão apenas é aplicada a chaves do KMS associadas a aliases que começam com `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*"
        }
      }
    }
  ]
}
```

A condição `kms:ResourceAliases` é uma condição do recurso, e não a solicitação. Dessa forma, uma solicitação que não especifique o alias ainda pode atender à condição.

O seguinte exemplo de solicitação, que especifica um alias correspondente, atende à condição.

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

No entanto, o seguinte exemplo de solicitação também atende à condição, desde que a chave do KMS especificada tenha um alias que comece com `restricted`, mesmo que este não seja utilizado na solicitação.

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

Saiba como usar aliases em suas aplicações

É possível usar um alias para representar uma chave do KMS no código da sua aplicação. O `KeyId` parâmetro em [operações AWS KMS criptográficas](#), [DescribeKey](#), e [GetPublicKey](#) aceita um nome de alias ou ARN de alias.

Por exemplo, o comando `GenerateDataKey` a seguir usa um nome de alias (`alias/finance`) para identificar uma chave do KMS. O nome do alias é o valor do parâmetro `KeyId`.

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

Se a chave KMS estiver em um local diferente Conta da AWS, você deverá usar um ARN de chave ou um alias ARN nessas operações. Ao usar um ARN de alias, não se esqueça de que o alias de uma chave do KMS é definido na conta que tem a chave do KMS e que ele pode ser diferente em cada região. Para obter ajuda sobre como encontrar o ARN do alias, consulte [Encontrar o nome e o ARN do alias para uma chave do KMS](#).

Por exemplo, o seguinte comando `GenerateDataKey` usa uma chave do KMS que não está na conta do autor da chamada. O alias `ExampleAlias` está associado à chave do KMS na conta e região especificadas.

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

Um dos usos mais poderosos dos aliases é em aplicações executadas em várias Regiões da AWS. Por exemplo, você pode ter uma aplicação global que usa uma [chave do KMS assimétrica RSA](#) para assinatura e verificação.

- Na região Oeste dos EUA (Oregon) (`us-west-2`), você deseja usar `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
- Na região Europa (Frankfurt) (`eu-central-1`), você deseja usar `arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321`.
- Na região Ásia-Pacífico (Singapura) (`ap-southeast-1`), você deseja usar `arn:aws:kms:ap-southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d`.

É possível criar uma versão diferente da sua aplicação em cada região ou usar um dicionário ou instrução de alternância para selecionar a chave do KMS correta para cada região. Mas é muito

mais fácil criar um alias com o mesmo nome em cada região. Lembre-se de que o nome do alias diferencia maiúsculas de minúsculas.

```
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
```

Depois, use o alias em seu código. Quando seu código for executado em cada região, o alias fará referência à sua chave do KMS associada nessa região. Por exemplo, esse código chama a operação [Sign](#) com um nome de alias.

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

No entanto, existe o risco de que o alias seja excluído ou atualizado para ser associado a outra chave do KMS. Nesse caso, as tentativas da aplicação de verificar assinaturas usando o nome do alias falharão e talvez seja necessário recriar ou atualizar o alias.

Para atenuar esse risco, tenha cuidado ao conceder permissão às entidades principais para gerenciar os aliases usados em sua aplicação. Para obter detalhes, consulte [Controlar o acesso a aliases](#).

Existem várias outras soluções para aplicações que criptografam dados em várias regiões da Regiões da AWS, incluindo [AWS Encryption SDK](#).

Encontre aliases em registros AWS CloudTrail

Você pode usar um alias para representar uma AWS KMS key em uma operação de AWS KMS API. Ao fazer isso, o alias e o ARN da chave KMS são registrados na entrada de registro AWS CloudTrail do evento. O alias é exibido no campo `requestParameters`. O ARN de chave é exibido no campo `resources`. Isso é verdade mesmo quando um AWS serviço usa um Chave gerenciada pela AWS em sua conta.

Por exemplo, a [GenerateDataKey](#) solicitação a seguir usa o `project-key` alias para representar uma chave KMS.

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

Quando essa solicitação é registrada no CloudTrail registro, a entrada do registro inclui o alias e o ARN da chave KMS real que foi usada.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "ABCDE",
    "arn": "arn:aws:iam::111122223333:role/ProjectDev",
    "accountId": "111122223333",
    "accessKeyId": "FFHIJ",
    "userName": "example-dev"
  },
  "eventTime": "2020-06-29T23:36:41Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "205.205.123.000",
  "userAgent": "aws-cli/1.18.89 Python/3.6.10
Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 boto3/1.17.12",
  "requestParameters": {
    "keyId": "alias/project-key",
    "keySpec": "AES_256"
  },
  "responseElements": null,
  "requestID": "d93f57f5-d4c5-4bab-8139-5a1f7824a363",
  "eventID": "d63001e2-dbc6-4aae-90cb-e5370aca7125",
  "readOnly": true,
```

```
    "resources": [  
      {  
        "accountId": "111122223333",  
        "type": "AWS::KMS::Key",  
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"  
      },  
      {  
        "eventType": "AwsApiCall",  
        "recipientAccountId": "111122223333"  
      }  
    ]  
  }
```

Para obter detalhes sobre AWS KMS as operações de registro em CloudTrail registros, consulte [Registrando chamadas de AWS KMS API com AWS CloudTrail](#).

Etiquetas em AWS KMS

Uma tag é um rótulo de metadados opcional que você pode atribuir (ou AWS atribuir) a um AWS recurso. Cada tag consiste em uma chave de tag e um valor de tag, sendo ambos strings que diferenciam maiúsculas de minúsculas. O valor da tag pode ser uma string vazia (nula). Cada tag em um recurso precisa ter uma chave de tag diferente, mas você pode adicionar a mesma tag a vários AWS recursos. Cada recurso pode ter até 50 tags criadas pelo usuário.

Não inclua informações confidenciais ou sigilosas na chave ou no valor da tag. As tags podem ser acessadas por muitos Serviços da AWS, incluindo o faturamento.

[Em AWS KMS, você pode adicionar tags a uma chave gerenciada pelo cliente ao criar a chave KMS e marcar ou desmarcar as chaves KMS existentes, a menos que estejam pendentes de exclusão.](#) Você não pode marcar aliases Chaves gerenciadas pela AWS, Chaves pertencentes à AWS armazenamentos de chaves personalizadas ou chaves KMS em outros. Contas da AWS Etiquetas são opcionais, mas podem ser bastante úteis.

Por exemplo, você pode adicionar uma etiqueta "Project"="Alpha" a todas as chaves do KMS e buckets do Amazon S3 usados para o projeto Alpha.

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

Para obter informações gerais sobre tags, incluindo o formato e a sintaxe, consulte [AWS Recursos de marcação](#) no. Referência geral da Amazon Web Services

As tags ajudam a:

- Identifique e organize seus AWS recursos. Muitos AWS serviços oferecem suporte à marcação, então você pode atribuir a mesma tag a recursos de serviços diferentes para indicar que os recursos estão relacionados. Por exemplo, você pode atribuir a mesma tag a uma chave KMS e a um volume ou segredo do Amazon Elastic Block Store (Amazon EBS). AWS Secrets Manager Você também pode usar etiquetas para identificar as chaves do KMS para automação.
- Acompanhe seus AWS custos. Quando você adiciona tags aos seus AWS recursos, AWS gera um relatório de alocação de custos com uso e custos agregados por tags. Você pode usar esse recurso para controlar AWS KMS os custos de um projeto, aplicativo ou centro de custos.

Para obter mais informações sobre como usar etiquetas para alocação de custos, consulte [Usar etiquetas de alocação de custos](#) no Manual do usuário do AWS Billing . Para obter informações

sobre as regras para chaves e valores de etiquetas, consulte [Restrições de etiquetas definidas pelo usuário](#), no Manual do usuário do AWS Billing .

- Controle o acesso aos seus AWS recursos. Permitir e negar acesso às chaves KMS com base em suas tags faz parte do AWS KMS suporte ao [controle de acesso baseado em atributos \(ABAC\)](#). Para obter informações sobre como controlar o acesso AWS KMS keys com base em suas tags, consulte [Usar tags para controlar o acesso a chaves do KMS](#). Para obter mais informações gerais sobre o uso de tags para controlar o acesso aos AWS recursos, consulte Como [controlar o acesso aos AWS recursos usando tags](#) de recursos no Guia do usuário do IAM.

AWS KMS grava uma entrada no seu AWS CloudTrail registro quando você usa as [ListResourceTags](#) operações [TagResource](#) [UntagResource](#), ou.

Tópicos

- [Controlar o acesso às tags](#)
- [Adicionar tags a uma chave do KMS](#)
- [Editar tags associadas a uma chave do KMS](#)
- [Remover tags associadas a uma chave do KMS](#)
- [Visualizar tags associadas a uma chave do KMS](#)
- [Usar tags para controlar o acesso a chaves do KMS](#)

Controlar o acesso às tags

Para adicionar, visualizar e excluir tags, seja no AWS KMS console ou usando a API, os diretores precisam de permissões de marcação. Você pode fornecer essas permissões em [políticas de chaves](#). Também é possível fornecê-las em políticas do IAM (incluindo [políticas de endpoint da VPC](#)), mas somente se [permitido pela política de chaves](#). A política [AWSKeyManagementServicePowerUser](#) gerenciada permite que os diretores marquem, desmarquem e listem as tags em todas as chaves KMS que a conta pode acessar.

Você também pode limitar essas permissões usando chaves de condição AWS globais para tags. Em AWS KMS, essas condições podem controlar o acesso às operações de marcação, como [TagResource](#) e [UntagResource](#).

Note

Tenha cuidado ao conceder permissão a entidades principais para gerenciar etiquetas e aliases. Alterar uma etiqueta ou um alias pode conceder ou negar uma permissão à chave gerenciada pelo cliente. Para obter mais detalhes, consulte [ABAC para AWS KMS](#) e [Usar tags para controlar o acesso a chaves do KMS](#).

Para mais informações e exemplos de políticas, consulte [Controlar o acesso baseado em chaves de etiqueta](#), no Guia do Usuário do IAM.

As permissões para criar e gerenciar aliases funcionam como a seguir.

kms:TagResource

Permite que as entidades principais adicionem ou editem etiquetas. Para adicionar etiquetas ao criar uma chave do KMS, a entidade principal deve ter permissão em uma política do IAM que não esteja restrita a chaves do KMS específicas.

kms:ListResourceTags

Permite que as entidades principais visualizem etiquetas em chaves do KMS.

kms:UntagResource

Permite que as entidades principais excluam etiquetas de chaves do KMS.

Permissões de etiquetas em políticas

Você pode fornecer permissões de marcação em uma política de chaves ou política do IAM. O seguinte exemplo de política de chaves concede permissão de marcação a usuários selecionados na chave do KMS. Ele concede a todos os usuários que podem assumir os exemplos de funções Administrador ou Desenvolvedor permissão para visualizar etiquetas.

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": "*"
}
]
}

```

Para conceder permissão de marcação de entidades principais em várias chaves do KMS, é possível usar uma política do IAM. Para que essa política seja eficiente, a política de chaves de cada chave do KMS deve permitir que a conta utilize políticas do IAM para controlar o acesso à chave do KMS.

Por exemplo, a seguinte política do IAM permite que as entidades principais criem chaves do KMS. Ela também permite que eles criem e gerenciem etiquetas em todas as chaves do KMS na conta especificada. Essa combinação permite que os diretores usem o parâmetro [Tags](#) da [CreateKey](#) operação para adicionar tags a uma chave KMS enquanto a criam.

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/*"
    }
  ]
}
```

Limitar permissões de etiquetas

É possível limitar permissões de marcação usando [condições de política](#). As seguintes condições de política podem ser aplicadas às permissões `kms:TagResource` e `kms:UntagResource`. Por exemplo, você pode usar a condição `aws:RequestTag/tag-key` para permitir que uma entidade principal adicione apenas etiquetas específicas, ou pode impedir que uma entidade principal adicione etiquetas com chaves de etiqueta específicas. Você também pode usar a condição `kms:KeyOrigin` para impedir que as entidades principais marquem ou desmarquem chaves do KMS com [material de chave importado](#).

- [leis: RequestTag](#)
- [aws:ResourceTag/tag-key](#) (somente políticas do IAM)
- [leis: TagKeys](#)
- [kms: CallerAccount](#)

- [kms: KeySpec](#)
- [kms: KeyUsage](#)
- [kms: KeyOrigin](#)
- [kms: ViaService](#)

Como prática recomendada ao usar etiquetas para controlar o acesso a chaves do KMS, use a chave de condição `aws:RequestTag/tag-key` ou `aws:TagKeys` para determinar quais etiquetas (ou chaves de etiqueta) são permitidas.

Por exemplo, a política do IAM a seguir é semelhante à anterior. No entanto, essa política permite que as entidades principais criem etiquetas (`TagResource`) e excluam etiquetas `UntagResource` somente para etiquetas com um chave de etiqueta `Project`.

Como `TagResource` as `UntagResource` solicitações podem incluir várias tags, você deve especificar um operador `ForAllValues` ou `ForAnyValue` definir com a `TagKeys` condição [aws:](#). O operador `ForAnyValue` exige que pelo menos uma das chaves de etiqueta na solicitação corresponda a uma das chaves de etiqueta na política. O operador `ForAllValues` requer que todas as chaves de etiqueta na solicitação correspondam a uma das chaves de etiqueta na política. O `ForAllValues` operador também retorna `true` se não houver tags na solicitação, mas `TagResource` `UntagResource` falhará quando nenhuma tag for especificada. Para detalhes sobre os operadores de conjunto, consulte [Usar várias chaves e valores](#), no Manual do usuário do 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"}
      }
    }
  ]
}
```

Adicionar tags a uma chave do KMS

As tags ajudam a identificar e organizar seus AWS recursos. É possível adicionar tags a uma chave gerenciada pelo cliente ao [criar a chave do KMS](#) ou adicionar tags a chaves existentes do KMS. Você não pode marcar Chaves gerenciadas pela AWS.

Os procedimentos a seguir demonstram como adicionar tags às chaves gerenciadas pelo cliente usando o AWS KMS console e a AWS KMS API. Os exemplos de AWS KMS API usam o [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Tópicos

- [Adicionar etiquetas ao criar uma chave do KMS](#)
- [Adicionar tags a chaves existentes do KMS](#)

Adicionar etiquetas ao criar uma chave do KMS

Você pode adicionar tags a uma chave KMS ao criar a chave usando o AWS KMS console ou a [CreateKey](#) operação. Para adicionar tags ao criar uma chave do KMS, é necessário ter uma permissão `kms:TagResource` em uma política do IAM, além das permissões necessárias para criar chaves do KMS. Essa permissão deve abranger pelo menos todas as chaves do KMS na conta e na região. Para obter detalhes, consulte [Controlar o acesso às tags](#).

Usando o AWS KMS console

Para adicionar tags ao criar uma chave do KMS no console, é necessário ter as permissões necessárias para visualizar chaves do KMS no console, além das permissões necessárias para marcar com tag e criar chaves do KMS. Essa permissão deve abranger pelo menos todas as chaves do KMS na conta e na região.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. (Você não pode gerenciar as etiquetas de uma Chave gerenciada pela AWS)
4. Escolha o tipo de chave e selecione Next (Avançar).
5. Insira um alias e uma descrição opcional.
6. Insira uma chave de etiqueta e, opcionalmente, um valor de etiqueta. Para adicionar mais etiquetas, selecione Add new tag (Adicionar nova etiqueta). Para remover uma etiqueta, escolha Remove tag (Remover etiqueta). Quando terminar de marcar sua nova chave do KMS, selecione Next (Avançar).
7. Finalize a criação da sua chave do KMS.

Usando a AWS KMS API

Para especificar tags ao criar chaves usando a [CreateKey](#) operação, use o Tags parâmetro da operação.

O valor do parâmetro Tags de CreateKey é uma coleção de pares de chave de etiqueta e valor de etiqueta que faz distinção entre maiúsculas e minúsculas. Cada etiqueta em uma chave do KMS deve ter um nome de etiqueta diferente. O valor da tag pode ser uma string nula ou vazia.

Por exemplo, o AWS CLI comando a seguir cria uma chave KMS de criptografia simétrica com uma Project:Alpha tag. Ao especificar mais de um par de chave-valor, use um espaço para separar cada par.

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

Quando esse comando é bem-sucedido, ele retorna um objeto `KeyMetadata` com informações sobre a nova chave do KMS. No entanto, o `KeyMetadata` não inclui tags. Para obter as tags, use a [ListResourceTags](#) operação.

Adicionar tags a chaves existentes do KMS

Você pode adicionar tags às suas chaves KMS gerenciadas pelo cliente existentes no AWS KMS console ou usando a [TagResource](#) operação. Para adicionar tags, é necessário ter permissão para marcação de tags na chave do KMS. É possível obter essa permissão da política de chaves da chave do KMS ou, se a política de chaves permitir, de uma política do IAM que inclua essa chave do KMS.

Usando o AWS KMS console

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. (Você não pode gerenciar as etiquetas de uma Chave gerenciada pela AWS)
4. É possível usar o filtro de tabela para exibir somente chaves do KMS com etiquetas específicas. Para obter detalhes, consulte [Visualizar tags usando o console do AWS KMS](#).
5. Marque a caixa de seleção ao lado do alias de uma chave do KMS.
6. Selecione Key actions (Ações de chave), Add or edit tags (Adicionar ou editar tags).
7. Na página de detalhes da chave do KMS, selecione a guia Tags (Etiquetas).
 - Para criar sua primeira etiqueta, selecione Create tag (Criar etiqueta), digite um nome e um valor de etiqueta e selecione Save (Salvar).

Se você deixar o valor da tag em branco, o valor da tag real será uma string nula ou vazia.
 - Para adicionar uma etiqueta, selecione Edit (Editar), Add tag (Adicionar etiqueta), digite um nome de etiqueta e um valor de etiqueta e selecione Save (Salvar).
8. Para salvar suas alterações, escolha Salvar alterações.

Usando a AWS KMS API

A [TagResource](#) operação adiciona uma ou mais tags a uma chave KMS. Não é possível usar essa operação para adicionar tags em uma Conta da AWS diferente. Você também pode usar

a `TagResource` operação para editar tags existentes. Para obter mais informações, consulte [the section called “Editar tags”](#).

Para adicionar uma tag, especifique uma nova chave e um valor de tag. Cada etiqueta em uma chave do KMS deve ter um nome de etiqueta diferente. O valor da tag pode ser uma string nula ou vazia.

Por exemplo, o comando a seguir adiciona as etiquetas **Purpose** e **Department** a uma chave do KMS demonstrativa.

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

Quando esse comando for executado com êxito, ele não retornará nenhuma saída. Para visualizar as tags em uma chave KMS, use a [ListResourceTags](#) operação.

Editar tags associadas a uma chave do KMS

As tags ajudam a identificar e organizar seus AWS recursos. Você pode editar as tags associadas às chaves KMS gerenciadas pelo cliente no AWS KMS console ou usando a [TagResource](#) operação. Você não pode editar as tags de um Chave gerenciada pela AWS.

Os procedimentos a seguir demonstram como editar as tags associadas a uma chave do KMS. Os exemplos de API do AWS KMS usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Usando o AWS KMS console

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. (Você não pode editar as tags de um Chave gerenciada pela AWS)
4. É possível usar o filtro de tabela para exibir somente chaves do KMS com etiquetas específicas. Para obter detalhes, consulte [Visualizar tags usando o console do AWS KMS](#).
5. Marque a caixa de seleção ao lado do alias de uma chave do KMS.
6. Selecione Key actions (Ações de chave), Add or edit tags (Adicionar ou editar tags).

7. Na página de detalhes da chave do KMS, selecione a guia Tags (Etiquetas).
 - Para alterar o nome ou o valor de uma tag, selecione Edit (Editar), faça as alterações e selecione Save (Salvar).
8. Para salvar suas alterações, escolha Salvar alterações.

Usando a AWS KMS API

A [TagResource](#) operação adiciona uma ou mais tags a uma chave gerenciada pelo cliente;. No entanto, você também pode usar TagResource para alterar o valor de uma tag existente. Não é possível usar essa operação para adicionar ou editar etiquetas em uma Conta da AWS diferente.

Para editar uma tag, especifique uma chave de tag existente e um novo valor de tag. Cada etiqueta em uma chave do KMS deve ter um nome de etiqueta diferente. O valor da tag pode ser uma string nula ou vazia.

Por exemplo, esse comando altera o valor da tag Purpose de Pretest para Test.

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

Remover tags associadas a uma chave do KMS

As tags ajudam a identificar e organizar seus AWS recursos. Você pode remover as tags associadas às chaves KMS gerenciadas pelo cliente no AWS KMS console ou usando a [UntagResource](#) operação. Você não pode editar ou remover as tags de um Chave gerenciada pela AWS.

Os procedimentos a seguir demonstram como remover tags de uma chave do KMS. Os exemplos de API do AWS KMS usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Usando o AWS KMS console

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.

3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. (Você não pode gerenciar as etiquetas de uma Chave gerenciada pela AWS)
4. É possível usar o filtro de tabela para exibir somente chaves do KMS com etiquetas específicas. Para obter detalhes, consulte [Visualizar tags usando o console do AWS KMS](#).
5. Marque a caixa de seleção ao lado do alias de uma chave do KMS.
6. Selecione Key actions (Ações de chave), Add or edit tags (Adicionar ou editar tags).
7. Na página de detalhes da chave do KMS, selecione a guia Tags (Etiquetas).
 - Para excluir uma tag, selecione Edit (Editar). Na linha da tag, selecione Remove (Remover) e Save (Salvar).
8. Para salvar suas alterações, escolha Salvar alterações.

Usando a AWS KMS API

A [UntagResource](#) operação exclui tags de uma chave KMS. Para identificar as etiquetas a serem excluídas, especifique as chaves de etiqueta. Essa operação não pode ser usada para excluir etiquetas de chaves do KMS em uma Conta da AWS diferente.

Quando é bem-sucedida, a operação UntagResource não retorna nenhuma saída. Além disso, se a chave de etiqueta especificada não for encontrada na chave do KMS, ela não lançará uma exceção nem retornará uma resposta. Para confirmar se a operação funcionou, use a [ListResourceTags](#) operação.

Por exemplo, esse comando exclui a etiqueta **Purpose** e seu valor da chave do KMS especificada.

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

Visualizar tags associadas a uma chave do KMS

As tags ajudam a identificar e organizar seus AWS recursos. Você pode visualizar as tags associadas às chaves KMS gerenciadas pelo cliente no AWS KMS console ou usando a [ListResourceTags](#) operação.

Os procedimentos a seguir demonstram como encontrar as tags associadas a uma chave específica do KMS. Os exemplos de AWS KMS API usam o [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Usando o AWS KMS console

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, escolha Chaves gerenciadas pelo cliente. (Você não pode gerenciar as etiquetas de uma Chave gerenciada pela AWS)
4. É possível usar o filtro de tabela para exibir somente chaves do KMS com etiquetas específicas.

Para exibir apenas chaves do KMS com uma etiqueta específica, selecione a caixa de filtro, a chave de etiqueta e entre os valores de etiquetas reais. Também é possível digitar todo o valor da etiqueta ou parte dele.

A tabela resultante mostra todas as chaves do KMS com a etiqueta selecionada. Porém, essa tabela não mostra a etiqueta. Para ver a etiqueta, selecione o alias ou o ID da chave da KMS e, na página de detalhes, escolha a guia Tags (Etiquetas). As guias aparecem sob a seção General configuration (Configuração geral).

Esse filtro exige a chave e o valor da etiqueta. Ele não localizará chaves do KMS apenas com a chave da etiqueta ou apenas com seu valor. Para filtrar as tags por toda ou parte da chave ou valor da tag, use a [ListResourceTags](#) operação para obter chaves KMS marcadas e, em seguida, use os recursos de filtragem da sua linguagem de programação.

5. Marque a caixa de seleção ao lado do alias de uma chave do KMS.
6. Selecione Key actions (Ações de chave), Add or edit tags (Adicionar ou editar tags).
7. Na página de detalhes da chave do KMS, selecione a guia Tags (Etiquetas).

Usando a AWS KMS API

A [ListResourceTags](#) operação obtém as tags de uma chave KMS. O parâmetro KeyId é obrigatório. Essa operação não pode ser usada para visualizar as etiquetas nas chaves do KMS em uma Conta da AWS diferente.

Por exemplo, o comando a seguir obtém as etiquetas para uma chave do KMS demonstrativa.

```
$ 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"  
  }  
]  
}
```

Usar tags para controlar o acesso a chaves do KMS

Você pode controlar o acesso AWS KMS keys com base nas tags na chave KMS. Por exemplo, você pode escrever uma política do IAM que permite que as entidades principais habilitem e desabilitem somente as chaves do KMS que possuem uma etiqueta específica. Ou você pode usar uma política do IAM para impedir que as entidades principais usem chaves do KMS em operações de criptografia, a menos que a chave do KMS tenha uma etiqueta específica.

Esse recurso faz parte do AWS KMS suporte ao [controle de acesso baseado em atributos](#) (ABAC). Para obter informações sobre o uso de tags para controlar o acesso aos AWS recursos, consulte [Para AWS que serve o ABAC?](#) e [controlar o acesso aos AWS recursos usando tags de recursos](#) no Guia do usuário do IAM. Para obter ajuda para resolver problemas de acesso relacionados ao ABAC, consulte [Solução de problemas do ABAC para AWS KMS](#).

Note

Pode levar até cinco minutos para que alterações de etiqueta e alias afetem a autorização de chaves do KMS. Alterações recentes podem estar visíveis em operações de API antes de afetarem a autorização.

AWS KMS suporta a chave de [contexto de condição global aws:ResourceTag/tag-key](#), que permite controlar o acesso às chaves KMS com base nas tags da chave KMS. Como várias chaves do KMS podem ter a mesma etiqueta, esse recurso permite que você aplique a permissão a um conjunto

selecionado de chaves do KMS. Também é possível alterar facilmente as chaves do KMS no conjunto alterando suas etiquetas.

Em AWS KMS, a chave de `aws:ResourceTag/tag-key` condição é suportada somente nas políticas do IAM. Ela não é suportada em políticas de chaves, que se aplicam somente a uma chave KMS, ou em operações que não usam uma chave KMS específica, como as operações [ListKeys](#) ou [ListAliases](#).

Controlar o acesso com etiquetas é uma maneira simples, escalável e flexível de gerenciar permissões. No entanto, se isso não for projetado e gerenciado corretamente, poderá permitir ou negar acesso às chaves do KMS inadvertidamente. Se estiver usando etiquetas para controlar o acesso, considere as seguintes práticas.

- Use tags para reforçar a prática recomendada do [acesso com privilégio mínimo](#). Conceda às entidades principais do IAM somente as permissões de que eles precisam nas chaves do KMS que elas devem usar ou gerenciar. Por exemplo, use etiquetas para rotular as chaves do KMS usadas para um projeto. Em seguida, dê permissão à equipe do projeto para usar somente chaves do KMS com a etiqueta do projeto.
- Tenha cuidado ao conceder às entidades principais as permissões `kms:TagResource` e `kms:UntagResource`, com as quais elas podem adicionar, editar e excluir etiquetas. Quando você usa etiquetas para controlar o acesso a chaves do KMS, a alteração de uma etiqueta pode dar permissão às entidades principais para usar chaves do KMS que, de outra forma, elas não teriam permissão de usar. Ele também pode negar acesso a chaves do KMS que outras entidades principais exigem para realizar seus trabalhos. Os administradores de chaves que não tiverem permissão para alterar políticas de chaves ou criar concessões poderão controlar o acesso às chaves do KMS se tiverem permissão para gerenciar etiquetas.

Sempre que possível, use uma condição de política, como `aws:RequestTag/tag-key` ou `aws:TagKeys`, para [limitar as permissões de marcação de uma entidade principal](#) para etiquetas ou padrões de etiquetas específicos em chaves do KMS específicas.

- Revise os diretores Conta da AWS que atualmente têm permissões de marcação e desmarcação e ajuste-os, se necessário. Por exemplo, a [política de chaves padrão para administradores de chaves](#) do console inclui as permissões `kms:TagResource` e `kms:UntagResource` nessa chave do KMS. As políticas do IAM podem conceder permissões de marcação ou desmarcação em todas as chaves do KMS. Por exemplo, a política [AWSKeyManagementServicePowerUser](#) gerenciada permite que os diretores marquem, desmarquem e listem as tags em todas as chaves do KMS.

- Antes de definir uma política que dependa de uma tag, revise as tags nas chaves KMS do seu Conta da AWS. Certifique-se de que sua política se aplica somente às etiquetas que você pretende incluir. Use [CloudTrail registros](#) e [CloudWatch alarmes](#) para alertá-lo sobre alterações nas tags que possam afetar o acesso às suas chaves KMS.
- As condições de políticas baseadas em etiquetas usam correspondência de padrões. Elas não estão vinculadas a uma instância específica de uma etiqueta. Uma política que usa chaves de condição baseadas em etiquetas afeta todas as etiquetas novas e existentes que correspondem ao padrão. Se você excluir e recriar uma etiqueta que corresponde a uma condição de política, a condição se aplicará à nova etiqueta, assim como à antiga.

Por exemplo, considere a seguinte política do IAM. Ele permite que os diretores liguem para as operações [GenerateDataKeyWithoutPlaintext](#) e [descriptografem](#) somente em chaves KMS em sua conta que estejam na região Ásia-Pacífico (Cingapura) e tenham uma tag. "Project"="Alpha" Você pode anexar essa política a funções no exemplo do projeto Alpha.

JSON

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

O exemplo de política do IAM a seguir permite que as entidades principais usem qualquer chave do KMS na conta para determinadas operações de criptografia. Porém, ela proíbe as entidades principais de usar estas operações criptográficas em chaves do KMS com uma tag "Type"="Reserved" ou sem etiqueta "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"
    }
  }
}
```

Repositórios de chaves

Um repositório de chaves é um local seguro para armazenar e usar chaves de criptografia. O armazenamento de chaves padrão AWS KMS também oferece suporte a métodos para gerar e gerenciar as chaves que ele armazena. Por padrão, o material da chave criptográfica AWS KMS keys que você cria AWS KMS é gerado e protegido por módulos de segurança de hardware (HSMs) que são o Programa de Validação do Módulo [Criptográfico FIPS 140-3](#). O material chave para suas chaves KMS nunca sai HSMs sem criptografia.

AWS KMS oferece suporte a vários tipos de armazenamentos de chaves para proteger seu material de chaves ao usá-lo AWS KMS para criar e gerenciar suas chaves de criptografia. Todas as opções de armazenamento de chaves fornecidas pela AWS KMS são continuamente validadas de acordo com o FIPS 140-3 no nível de segurança 3 e foram projetadas para impedir que qualquer pessoa, incluindo AWS operadores, acesse suas chaves de texto simples ou as use sem sua permissão.

AWS KMS armazenamento de chaves padrão

Por padrão, uma chave KMS é criada usando o AWS KMS HSM padrão. Esse tipo de HSM pode ser considerado uma frota multilocatária HSMs que permite o armazenamento de chaves mais escalável, de menor custo e mais fácil de gerenciar do seu ponto de vista. Se você estiver criando uma chave KMS para uso em uma ou mais Serviços da AWS, para que o serviço possa criptografar seus dados em seu nome, você criará uma chave simétrica. Se você estiver usando uma chave do KMS para o design de sua própria aplicação, poderá optar por criar uma chave de criptografia simétrica, uma chave assimétrica ou uma chave de HMAC.

Na opção de armazenamento de chaves padrão, AWS KMS cria sua chave e a criptografa com as chaves que o serviço gerencia internamente. Em seguida, várias cópias de versões criptografadas de suas chaves serão armazenadas em sistemas projetados visando durabilidade. Gerar e proteger seu material de chaves no tipo de armazenamento de chaves padrão permite que você aproveite ao máximo a escalabilidade, a disponibilidade e a durabilidade AWS KMS com a menor carga operacional e o menor custo dos armazenamentos de AWS chaves.

AWS KMS armazenamento de chaves padrão com material de chave importado

Em vez de solicitar AWS KMS a geração e o armazenamento das únicas cópias de uma determinada chave, você pode optar por importar o material da chave AWS KMS, permitindo gerar sua própria

chave de criptografia simétrica de 256 bits, chave RSA ou curva elíptica (ECC) ou chave de Código de Autenticação de Mensagens Baseadas em Hash (HMAC) e aplicá-la a um identificador de chave KMS (KeyID). Às vezes, isso é chamado de traga sua própria chave (BYOK). O material de chaves importado do seu sistema de gerenciamento de chaves local deve ser protegido usando uma chave pública emitida por AWS KMS, um algoritmo de empacotamento criptográfico compatível e um token de importação baseado em tempo fornecido pela AWS KMS. Esse processo verifica se sua chave criptografada e importada só pode ser descriptografada por um HSM do AWS KMS após ter deixado seu ambiente.

O material de chaves importado pode ser útil se você tiver requisitos específicos no sistema que gera chaves ou se quiser uma cópia externa da chave AWS como backup. Observe que você será responsável pela disponibilidade e durabilidade gerais do material de chave importado. Embora AWS KMS tenha uma cópia de sua chave importada e permaneça altamente disponível enquanto você precisar dela, as chaves importadas oferecem uma API especial para exclusão — `DeleteImportedKeyMaterial`. Essa API excluirá imediatamente todas as cópias do material de chave importado que AWS KMS possui, sem AWS a opção de recuperar a chave. Além disso, você pode definir um prazo de validade em uma chave importada, após o qual a chave ficará inutilizável. Para tornar a chave útil novamente AWS KMS, você precisará reimportar o material da chave e atribuí-lo ao mesmo KeyID. Essa ação de exclusão para chaves importadas é diferente das chaves padrão que são AWS KMS geradas e armazenadas para você em seu nome. No caso padrão, o processo de exclusão de chaves tem um período de espera obrigatório durante o qual o uso de uma chave programada para exclusão é bloqueado pela primeira vez. Essa ação permite que você veja erros de acesso negado nos registros de qualquer aplicativo ou AWS serviço que possa precisar dessa chave para acessar os dados. Se você receber essas solicitações de acesso, poderá optar por cancelar a exclusão programada e reativar a chave. Somente após um período de espera configurável (entre 7 e 30 dias), o KMS excluirá o material de chave, o keyID e todos os metadados associados à chave. Para obter mais informações sobre disponibilidade e durabilidade, consulte [Protecting imported key material](#) no AWS KMS Developer Guide.

É necessário ter ciência de algumas limitações adicionais com o material de chave importado. Como o AWS KMS não é capaz de gerar novo material de chave, não há como configurar a alternância automática de chaves importadas. Você precisará criar uma nova chave do KMS com uma nova keyID e, em seguida, importar o novo material de chave para obter uma alternância efetiva. Além disso, textos cifrados criados AWS KMS sob uma chave simétrica importada não podem ser facilmente descriptografados usando sua cópia local da chave externa. AWS Isso ocorre porque o formato de criptografia autenticado usado pelo AWS KMS acrescenta metadados adicionais ao texto cifrado para fornecer garantias durante a operação de descriptografia de que o texto cifrado

foi criado pela chave KMS esperada em uma operação de criptografia anterior. A maioria dos sistemas criptográficos externos não saberá como analisar esses metadados para obter acesso ao texto cifrado bruto e usar sua cópia de uma chave simétrica. Textos cifrados criados sob chaves assimétricas importadas (por exemplo, RSA ou ECC) podem ser usados fora da AWS KMS parte correspondente (pública ou privada) da chave porque não há metadados adicionais adicionados ao texto cifrado. AWS KMS

AWS KMS lojas de chaves personalizadas

No entanto, se você precisar de ainda mais controle do HSMs, poderá criar um armazenamento de chaves personalizado.

Um armazenamento de chaves personalizado é um armazenamento de chaves interno AWS KMS que é apoiado por um gerenciador de chaves externo AWS KMS, que você possui e gerencia. Os armazenamentos de chaves personalizados combinam a interface de gerenciamento de chaves conveniente e abrangente AWS KMS com a capacidade de possuir e controlar o material chave e as operações criptográficas. Ao usar uma chave do KMS em um armazenamento de chaves personalizado, as operações de criptografia são executadas pelo gerenciador de chaves usando chaves de criptografia. Como resultado, você assume mais responsabilidade pela disponibilidade e durabilidade das chaves criptográficas e pela operação do HSMs.

Possuir sua propriedade HSMs pode ser útil para ajudar a atender a certos requisitos regulatórios que ainda não permitem que serviços web multilocatários, como o armazenamento de chaves KMS padrão, mantenham suas chaves criptográficas. Os armazenamentos de chaves personalizados não são mais seguros do que os armazenamentos de chaves KMS AWS gerenciados pelo HSMs usuário, mas têm implicações de gerenciamento e custo diferentes (e maiores). Como resultado, você assume mais responsabilidade pela disponibilidade e durabilidade das chaves criptográficas e pela operação do HSMs. Independentemente de você usar o armazenamento de chaves padrão com AWS KMS HSMs ou um armazenamento de chaves personalizado, o serviço foi projetado para que ninguém, incluindo AWS funcionários, possa recuperar suas chaves de texto simples ou usá-las sem sua permissão. AWS KMS suporta dois tipos de armazenamentos de chaves personalizados, armazenamentos de AWS CloudHSM chaves e armazenamentos de chaves externos.

Recursos sem suporte

AWS KMS não oferece suporte aos seguintes recursos em armazenamentos de chaves personalizadas.

- [Chaves do KMS assimétricas](#)

- [Chaves do KMS de HMAC](#)
- [Chaves do KMS com material de chave importado](#)
- [Alternância automática de chaves](#)
- [Chaves de várias regiões](#)

AWS CloudHSM armazenamento de chaves

Você pode criar uma chave KMS em um armazenamento de [AWS CloudHSM](#) chaves, onde as chaves do usuário raiz são geradas, armazenadas e usadas em um AWS CloudHSM cluster que você possui e gerencia. As solicitações AWS KMS para usar uma chave para alguma operação criptográfica são encaminhadas ao seu AWS CloudHSM cluster para realizar a operação. Embora um AWS CloudHSM cluster seja hospedado por AWS, ele é uma solução de inquilino único gerenciada e operada diretamente por você. Você possui grande parte da disponibilidade e do desempenho das chaves KMS em um AWS CloudHSM cluster. Para ver se um armazenamento de chaves AWS CloudHSM personalizadas é adequado às suas necessidades, leia [Os armazenamentos de chaves AWS KMS personalizadas são adequados para você?](#) no blog AWS de segurança.

Armazenamento de chaves externas

Você pode configurar AWS KMS para usar um Armazenamento de Chaves Externo (XKS), onde as chaves do usuário raiz são geradas, armazenadas e usadas em um sistema de gerenciamento de chaves fora do Nuvem AWS. As solicitações para o AWS KMS usar uma chave para alguma operação criptográfica são encaminhadas ao seu sistema hospedado externamente realizar a operação. Mais especificamente, as solicitações são encaminhadas para um XKS Proxy em sua rede, que então encaminha a solicitação para qualquer sistema criptográfico que você usa. O XKS Proxy tem uma especificação de código aberto com a qual qualquer pessoa pode se integrar. Muitos fornecedores comerciais de gerenciamento de chaves são compatíveis com a especificação XKS Proxy. Como um repositório de chaves externo é hospedado por você ou por terceiros, você é o proprietário de toda a disponibilidade, durabilidade e desempenho das chaves no sistema. Para ver se um armazenamento externo de chaves é adequado às suas necessidades, leia [Anunciando o armazenamento AWS KMS externo de chaves \(XKS\) no blog](#) de AWS notícias.

AWS CloudHSM lojas principais

Um armazenamento de AWS CloudHSM chaves é um [armazenamento de chaves personalizado](#) apoiado por um [AWS CloudHSM cluster](#). Quando você cria um AWS KMS key em um armazenamento de chaves personalizado, AWS KMS gera e armazena material de chave não extraível para a chave KMS em um AWS CloudHSM cluster que você possui e gerencia. Quando você usa uma chave KMS em um armazenamento de chaves personalizado, as [operações criptográficas](#) são executadas HSMs no cluster. Esse recurso combina a conveniência e a ampla integração do AWS KMS com o controle adicional de um AWS CloudHSM cluster em seu Conta da AWS.

AWS KMS fornece suporte completo de console e API para criar, usar e gerenciar seus armazenamentos de chaves personalizadas. Use as chaves do KMS no seu armazenamento de chaves personalizado da mesma maneira que você usa qualquer chave do KMS. Por exemplo, você pode usar as chaves do KMS para gerar chaves de dados e criptografar dados. Você também pode usar as chaves KMS em seu armazenamento de chaves personalizadas com AWS serviços que oferecem suporte a chaves gerenciadas pelo cliente.

Eu preciso de um armazenamento de chaves personalizado?

Para a maioria dos usuários, o armazenamento de AWS KMS chaves padrão, protegido por [módulos criptográficos validados pelo FIPS 140-3](#), atende aos requisitos de segurança. Não é necessário adicionar uma camada extra de responsabilidade de manutenção nem uma dependência em um serviço adicional.

No entanto, você pode considerar a criação de um armazenamento de chaves personalizado se a sua organização tiver um dos seguintes requisitos:

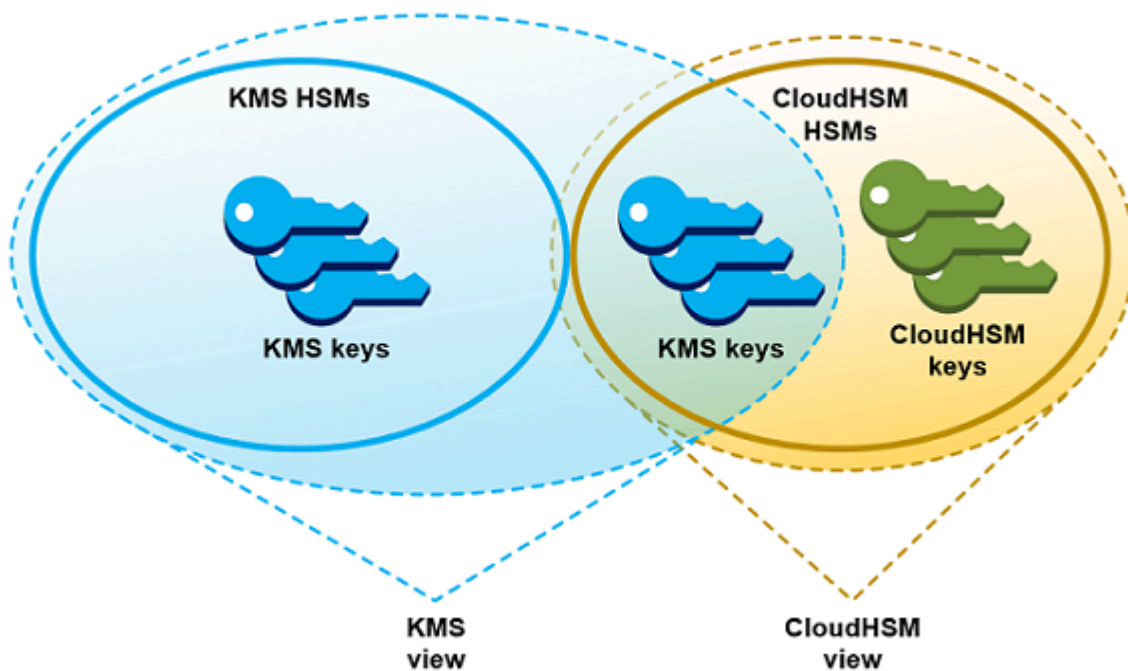
- Algumas chaves precisam ser explicitamente protegidas em um único HSM locatário ou em um HSM sobre o qual você tem controle direto.
- Você precisa da capacidade de remover imediatamente o material chave do AWS KMS.
- Você precisa ser capaz de auditar todo o uso de suas chaves, independentemente de AWS KMS ou AWS CloudTrail.

Como funcionam os armazenamentos de chaves personalizados?

Cada armazenamento de chaves personalizadas está associado a um AWS CloudHSM cluster no seu Conta da AWS. Quando você conecta o armazenamento de chaves personalizadas ao cluster,

AWS KMS cria a infraestrutura de rede para suportar a conexão. Em seguida, ele faz login no AWS CloudHSM cliente-chave no cluster usando as credenciais de um [usuário criptográfico dedicado](#) no cluster.

Você cria e gerencia seus armazenamentos de chaves personalizadas AWS KMS e cria e gerencia seus clusters de HSM em AWS CloudHSM. Ao criar AWS KMS keys em um armazenamento de chaves AWS KMS personalizado, você visualiza e gerencia as chaves KMS em AWS KMS. No entanto, você também pode visualizar e gerenciar seu material de chaves no AWS CloudHSM, do mesmo modo que para outras chaves no cluster.



Você pode [criar chaves KMS de criptografia simétrica com o material de chaves](#) gerado AWS KMS em seu armazenamento de chaves personalizadas. Em seguida, use as mesmas técnicas para visualizar e gerenciar as chaves KMS em seu armazenamento de chaves personalizadas que você usa para as chaves KMS no AWS KMS armazenamento de chaves. É possível controlar o acesso com políticas de chaves e do IAM, criar etiquetas e aliases, habilitar e desabilitar as chaves do KMS e programar a exclusão de chaves. Você pode usar as chaves KMS para [operações criptográficas](#) e usá-las com AWS serviços que se integram com o. AWS KMS

Além disso, você tem controle total sobre o AWS CloudHSM cluster, incluindo a criação, a exclusão HSMs e o gerenciamento de backups. Você pode usar o AWS CloudHSM cliente e as bibliotecas de software compatíveis para visualizar, auditar e gerenciar o material de chaves de suas chaves KMS. Embora o armazenamento de chaves personalizadas esteja desconectado, AWS KMS

não é possível acessá-lo e os usuários não podem usar as chaves KMS no armazenamento de chaves personalizadas para operações criptográficas. Essa camada adicional de controle torna os armazenamentos de chaves personalizados uma excelente solução para as organizações que necessitam dela.

Por onde começar?

Para criar e gerenciar um armazenamento de AWS CloudHSM chaves, você usa recursos de AWS KMS AWS CloudHSM e.

1. Comece em AWS CloudHSM. [Crie um cluster do AWS CloudHSM ativo](#) ou selecione um cluster existente. O cluster deve ter pelo menos dois ativos HSMs em diferentes zonas de disponibilidade. Crie uma [conta de usuário de criptografia \(CU\) dedicado](#) nesse cluster para o AWS KMS.
2. Em AWS KMS, [crie um armazenamento de chaves personalizado](#) associado ao AWS CloudHSM cluster selecionado. AWS KMS fornece uma interface de gerenciamento completa que permite criar, visualizar, editar e excluir seus repositórios de chaves personalizados.
3. Quando você estiver pronto para usar seu armazenamento de chaves personalizadas, [conecte-o ao AWS CloudHSM cluster associado](#). AWS KMS cria a infraestrutura de rede necessária para suportar a conexão. Ele faz login no cluster usando as credenciais da conta de usuário de criptografia dedicado para que possa gerar e gerenciar o material de chaves no cluster.
4. Agora, você pode [criar chaves do KMS de criptografia simétrica em seu armazenamento personalizado de chaves](#). Basta especificar esse armazenamento de chaves personalizado ao criar a chave do KMS.

Se você ficar preso em alguma etapa, é possível encontrar ajuda no tópico [Solucionar problemas de um armazenamento de chaves personalizado](#). Se a sua pergunta não for respondida, use o link de comentários na parte inferior de cada página deste guia ou publique uma pergunta no [fórum de discussão do AWS Key Management Service](#).

Cotas

AWS KMS permite até [10 armazenamentos de chaves personalizadas](#) em cada Conta da AWS região, incluindo armazenamentos de [AWS CloudHSM chaves e armazenamentos de chaves externos](#), independentemente do estado da conexão. Além disso, há cotas de AWS KMS solicitação sobre o [uso de chaves KMS em um AWS CloudHSM armazenamento de chaves](#).

Preços

Para obter informações sobre o custo dos armazenamentos de chaves AWS KMS personalizadas e das chaves gerenciadas pelo cliente em um repositório de chaves personalizadas, consulte [AWS Key Management Service preços](#). Para obter informações sobre o custo dos AWS CloudHSM clusters e HSMs, consulte [AWS CloudHSM Preços](#).

Regiões

AWS KMS oferece suporte às AWS CloudHSM principais lojas em todos os Regiões da AWS lugares onde AWS KMS há suporte, exceto na Ásia-Pacífico (Melbourne), China (Pequim), China (Ningxia) e Europa (Espanha).

Recursos sem suporte

AWS KMS não oferece suporte aos seguintes recursos em armazenamentos de chaves personalizadas.

- [Chaves do KMS assimétricas](#)
- [Chaves do KMS de HMAC](#)
- [Chaves do KMS com material de chave importado](#)
- [Alternância automática de chaves](#)
- [Chaves de várias regiões](#)

AWS CloudHSM conceitos-chave da loja

Este tópico explica alguns dos termos e conceitos usados nas AWS CloudHSM principais lojas.

AWS CloudHSM armazenamento de chaves

Um armazenamento de AWS CloudHSM chaves é um [armazenamento de chaves personalizado](#) associado a um AWS CloudHSM cluster que você possui e gerencia. AWS CloudHSM os clusters são apoiados por módulos de segurança de hardware (HSMs) certificados em [FIPS 140-2 ou FIPS 140-3 Nível 3](#).

Quando você cria uma chave KMS em seu armazenamento de chaves, AWS KMS gera uma AWS CloudHSM chave simétrica Advanced Encryption Standard (AES) persistente e não exportável de 256 bits no cluster associado. AWS CloudHSM Esse material chave nunca deixa você HSMs sem criptografia. Quando você usa uma chave KMS em um armazenamento de AWS CloudHSM chaves, as operações criptográficas são executadas HSMs no cluster.

AWS CloudHSM os armazenamentos de chaves combinam a interface conveniente e abrangente de gerenciamento de chaves AWS KMS com os controles adicionais fornecidos por um AWS CloudHSM cluster em seu Conta da AWS. Esse recurso integrado permite que você crie, gerencie e use chaves KMS AWS KMS enquanto mantém controle total sobre quem armazena seu material de chaves, incluindo o gerenciamento de clusters e backups. HSMs HSMs Você pode usar o AWS KMS console e APIs gerenciar o armazenamento de AWS CloudHSM chaves e suas chaves KMS. Você também pode usar o AWS CloudHSM console APIs, o software cliente e as bibliotecas de software associadas para gerenciar o cluster associado.

Você pode [visualizar e gerenciar](#) seu armazenamento de AWS CloudHSM chaves, [editar suas propriedades](#) e [conectá-lo](#) e [desconectá-lo](#) do AWS CloudHSM cluster associado. Se precisar [excluir um armazenamento de AWS CloudHSM chaves](#), primeiro exclua as chaves KMS no armazenamento de chaves agendando sua exclusão e aguardando até que o período de carência expire. AWS CloudHSM A exclusão do armazenamento de AWS CloudHSM chaves remove o recurso AWS KMS, mas isso não afeta seu AWS CloudHSM cluster.

AWS CloudHSM agrupamento

Cada armazenamento de AWS CloudHSM chaves está associado a um AWS CloudHSM cluster. Quando você cria um AWS KMS key em seu armazenamento de AWS CloudHSM chaves, AWS KMS cria seu material de chaves no cluster associado. Quando você usa uma chave do KMS no armazenamento de chaves do AWS CloudHSM , a operação de criptografia é realizada no cluster associado.

Cada AWS CloudHSM cluster pode ser associado a apenas um armazenamento de AWS CloudHSM chaves. O cluster que você escolher não pode ser associado a outro armazenamento de AWS CloudHSM chaves nem compartilhar um histórico de backup com um cluster associado a outro armazenamento de AWS CloudHSM chaves. O cluster deve estar inicializado e ativo, e deve estar na mesma Conta da AWS região do armazenamento de AWS CloudHSM chaves. Você pode criar um novo cluster ou usar um existente. AWS KMS não precisa do uso exclusivo do cluster. Para criar chaves KMS no armazenamento de AWS CloudHSM chaves, seu cluster associado deve conter pelo menos duas ativas HSMs. Todas as outras operações exigem apenas um HSM.

Você especifica o AWS CloudHSM cluster ao criar o armazenamento de AWS CloudHSM chaves e não pode alterá-lo. No entanto, você pode substituir qualquer cluster que compartilhe um histórico de backup pelo cluster original. Isso permite a você excluir o cluster, se necessário, e substituí-lo por um cluster criado a partir de um de seus backups. Você mantém o controle total do AWS CloudHSM cluster associado para poder gerenciar usuários e chaves, criar e excluir HSMs, usar e gerenciar backups.

Quando estiver pronto para usar seu armazenamento de AWS CloudHSM chaves, conecte-o ao AWS CloudHSM cluster associado. Você pode conectar e desconectar o armazenamento de chaves personalizado a qualquer momento. Se o armazenamento de chaves personalizado estiver conectado, você poderá criar e usar suas chaves do KMS. Quando ele é desconectado, você pode visualizar e gerenciar o armazenamento de AWS CloudHSM chaves e suas chaves KMS. Mas você não pode criar novas chaves KMS nem usar as chaves KMS no armazenamento de chaves para AWS CloudHSM operações criptográficas.

Usuário de criptografia **kmsuser**

Para criar e gerenciar o material de chaves no AWS CloudHSM cluster associado em seu nome, AWS KMS use um [usuário AWS CloudHSM criptográfico](#) (UC) dedicado no cluster chamado `kmsuser`. A `kmsuser` UC é uma conta de UC padrão que é automaticamente sincronizada com tudo HSMs no cluster e salva nos backups do cluster.

Antes de criar seu armazenamento de AWS CloudHSM chaves, você [cria uma conta de kmsuser UC](#) em seu AWS CloudHSM cluster usando o comando [user create](#) na CLI do CloudHSM. Em seguida, ao [criar o armazenamento de AWS CloudHSM chaves](#), você fornece a senha da `kmsuser` conta para AWS KMS. Quando você [conecta o armazenamento de chaves personalizadas](#), AWS KMS efetua login no cluster como a `kmsuser` UC e altera sua senha. AWS KMS criptografa sua `kmsuser` senha antes de armazená-la com segurança. Quando a senha é alternada, a nova senha é criptografada e armazenada da mesma maneira.

AWS KMS permanece conectado `kmsuser` enquanto o armazenamento de AWS CloudHSM chaves estiver conectado. Você não deve usar essa conta CU para outros fins. No entanto, você mantém o controle final do CU da conta `kmsuser`. A qualquer momento, você pode [encontrar as chaves](#) que pertencem ao `kmsuser`. Se necessário, você pode [desconectar o repositório de chaves personalizado](#), alterar a senha do `kmsuser`, [fazer login no cluster como kmsuser](#) e exibir e gerenciar as chaves pertencentes ao `kmsuser`.

Para obter instruções sobre como criar sua conta CU `kmsuser`, consulte [Criar o Usuário de criptografia kmsuser](#).

Chaves KMS em um armazenamento de AWS CloudHSM chaves

Você pode usar a AWS KMS API AWS KMS ou para criar um AWS KMS keys em um armazenamento de AWS CloudHSM chaves. Use a mesma técnica que você usaria em qualquer chave do KMS. A única diferença é que você deve identificar o armazenamento de AWS CloudHSM chaves e especificar que a origem do material da chave é o AWS CloudHSM cluster.

Quando você [cria uma chave KMS em um armazenamento de chaves](#), AWS KMS cria a [AWS CloudHSM chave](#) KMS AWS KMS e gera um material de chave simétrica Advanced Encryption Standard (AES) persistente e não exportável de 256 bits em seu cluster associado. Quando você usa a AWS KMS chave em uma operação criptográfica, a operação é executada no AWS CloudHSM cluster usando a chave AES baseada em cluster. Embora AWS CloudHSM ofereça suporte a chaves simétricas e assimétricas de diferentes tipos, os armazenamentos de chaves oferecem suporte somente a AWS CloudHSM chaves de criptografia simétricas AES.

Você pode visualizar as chaves KMS em um armazenamento de AWS CloudHSM chaves no AWS KMS console e usar as opções do console para exibir a ID personalizada do armazenamento de chaves. Você também pode usar a [DescribeKey](#) operação para encontrar o ID do armazenamento de AWS CloudHSM chaves e o ID AWS CloudHSM do cluster.

As chaves KMS em um armazenamento de chaves funcionam exatamente como qualquer AWS CloudHSM chave KMS em. AWS KMS Os usuários autorizados precisam das mesmas permissões para usar e gerenciar as chaves do KMS. Você usa os mesmos procedimentos de console e operações de API para visualizar e gerenciar as chaves KMS em um armazenamento de AWS CloudHSM chaves. Isso inclui a habilitar e desabilitar chaves do KMS, criar e usar etiquetas e aliases e definir e alterar políticas de chaves e do IAM. Você pode usar as chaves KMS em um AWS CloudHSM armazenamento de chaves para operações criptográficas e usá-las com [AWS serviços integrados](#) que oferecem suporte ao uso de chaves gerenciadas pelo cliente. No entanto, você não pode ativar a [rotação automática](#) de chaves ou [importar material de chaves para uma chave](#) KMS em um armazenamento de chaves. AWS CloudHSM

Você também usa o mesmo processo para [agendar a exclusão](#) de uma chave KMS em um armazenamento de AWS CloudHSM chaves. Depois que o período de espera expirar, AWS KMS exclui a chave KMS do KMS. Em seguida, ele se esforça ao máximo para excluir o material da chave KMS do AWS CloudHSM cluster associado. No entanto, pode ser necessário [excluir manualmente o material de chaves órfãos](#) do cluster e de seus backups.

Controle o acesso à sua loja de AWS CloudHSM chaves

Você usa políticas do IAM para controlar o acesso ao seu armazenamento de AWS CloudHSM chaves e ao seu AWS CloudHSM cluster. Você pode usar políticas de chaves, políticas de IAM e concessões para controlar o acesso ao AWS KMS keys em seu armazenamento de AWS CloudHSM chaves. Recomendamos que você forneça aos usuários, grupos e funções apenas as permissões que precisam para as tarefas que possivelmente executarão.

Para oferecer suporte aos seus AWS CloudHSM principais armazenamentos, AWS KMS precisa de permissão para obter informações sobre seus AWS CloudHSM clusters. Ele também precisa de permissão para criar a infraestrutura de rede que conecta seu armazenamento de AWS CloudHSM chaves ao AWS CloudHSM cluster. Para obter essas permissões, AWS KMS crie a função `AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculada ao serviço em seu. Conta da AWS Para obter mais informações, consulte [Autorização AWS KMS para gerenciar recursos AWS CloudHSM do Amazon EC2](#).

Ao projetar seu armazenamento de AWS CloudHSM chaves, certifique-se de que os diretores que o usam e gerenciam tenham apenas as permissões necessárias. A lista a seguir descreve as permissões mínimas necessárias para os AWS CloudHSM principais gerentes e usuários da loja.

- Os diretores que criam e gerenciam seu armazenamento de AWS CloudHSM chaves precisam da seguinte permissão para usar as operações da API do armazenamento de AWS CloudHSM chaves.
 - `cloudhsm:DescribeClusters`
 - `kms:CreateCustomKeyStore`
 - `kms:ConnectCustomKeyStore`
 - `kms>DeleteCustomKeyStore`
 - `kms:DescribeCustomKeyStores`
 - `kms:DisconnectCustomKeyStore`
 - `kms:UpdateCustomKeyStore`
 - `iam:CreateServiceLinkedRole`
- Os diretores que criam e gerenciam o AWS CloudHSM cluster associado ao seu armazenamento de AWS CloudHSM chaves precisam de permissão para criar e inicializar um AWS CloudHSM cluster. Isso inclui permissão para criar ou usar uma Amazon Virtual Private Cloud (VPC), criar sub-redes e criar uma instância da Amazon. EC2 Talvez eles também precisem criar HSMs, excluir e gerenciar backups. Para obter as listas das permissões necessárias, consulte [Identity and access management for AWS CloudHSM](#) (Gerenciamento de identidade e acesso para o) no Guia do usuário do AWS CloudHSM .
- Os diretores que criam e gerenciam AWS KMS keys em seu armazenamento de AWS CloudHSM chaves exigem [as mesmas permissões](#) que aqueles que criam e gerenciam qualquer chave KMS em. AWS KMS A [política de chaves padrão](#) para uma chave KMS em um armazenamento de AWS CloudHSM chaves é idêntica à política de chaves padrão para chaves KMS em. AWS KMS O [controle de acesso baseado em atributos](#) (ABAC), que usa tags e aliases para controlar o

acesso às chaves KMS, também é eficaz nas chaves KMS nos armazenamentos de chaves. AWS CloudHSM

- [Os diretores que usam as chaves KMS em seu AWS CloudHSM armazenamento de chaves para operações criptográficas precisam de permissão para realizar a operação criptográfica com a chave KMS, como kms:Decrypt.](#) Você pode fornecer essas permissões em uma política de chaves, política do IAM. Porém, eles não precisam de nenhuma permissão adicional para usar uma chave KMS em um armazenamento de AWS CloudHSM chaves.

Crie um armazenamento de AWS CloudHSM chaves

Você pode criar um ou vários armazenamentos de AWS CloudHSM chaves em sua conta. Cada armazenamento de AWS CloudHSM chaves está associado a um AWS CloudHSM cluster na mesma Conta da AWS região. Antes de criar o armazenamento de chaves do AWS CloudHSM , você precisa [organizar os pré-requisitos](#). Então, antes de poder usar seu armazenamento de AWS CloudHSM chaves, você deve [conectá-lo](#) ao AWS CloudHSM cluster.

Observações

O KMS não pode se comunicar IPv6 com as AWS CloudHSM principais lojas.

Se você tentar criar um armazenamento de AWS CloudHSM chaves com todos os mesmos valores de propriedade de um armazenamento de AWS CloudHSM chaves desconectado existente, AWS KMS não criará um novo armazenamento de AWS CloudHSM chaves e não gerará uma exceção nem exibirá um erro. Em vez disso, AWS KMS reconhece a duplicata como a provável consequência de uma nova tentativa e retorna a ID do armazenamento de chaves existente AWS CloudHSM .

Você não precisa conectar seu armazenamento de AWS CloudHSM chaves imediatamente. Você pode deixá-lo em um estado desconectado até estar pronto para usá-lo. No entanto, para verificar se ele está configurado corretamente, convém [conectá-lo](#), [visualizar o estado da conexão](#) e [desconectá-lo](#).

Tópicos

- [Organizar os pré-requisitos](#)
- [Criar um novo repositório de AWS CloudHSM chaves](#)

Organizar os pré-requisitos

Cada armazenamento de AWS CloudHSM chaves é apoiado por um AWS CloudHSM cluster. Para criar um armazenamento de AWS CloudHSM chaves, você deve especificar um AWS CloudHSM cluster ativo que ainda não esteja associado a outro armazenamento de chaves. Você também precisa criar um usuário criptográfico (UC) dedicado no cluster HSMs que AWS KMS possa ser usado para criar e gerenciar chaves em seu nome.

Antes de criar um armazenamento de AWS CloudHSM chaves, faça o seguinte:

Selecione um AWS CloudHSM cluster

Cada armazenamento de AWS CloudHSM chaves está [associado a exatamente um AWS CloudHSM cluster](#). Quando você cria AWS KMS keys em seu armazenamento de AWS CloudHSM chaves, AWS KMS cria os metadados da chave KMS, como um ID e um nome de recurso da Amazon (ARN) em. AWS KMS Em seguida, ele cria o HSMs material chave no cluster associado. Você pode [criar um novo AWS CloudHSM](#) cluster ou usar um existente. AWS KMS não exige acesso exclusivo ao cluster.

O AWS CloudHSM cluster que você seleciona está permanentemente associado ao armazenamento de AWS CloudHSM chaves. Depois de criar o armazenamento de AWS CloudHSM chaves, você pode [alterar a ID](#) do cluster associado, mas o cluster especificado deve compartilhar um histórico de backup com o cluster original. Para usar um cluster não relacionado, você precisa criar um novo armazenamento de AWS CloudHSM chaves.

O AWS CloudHSM cluster selecionado deve ter as seguintes características:

- O cluster deve estar ativo.

Você deve criar o cluster, inicializá-lo, instalar o software AWS CloudHSM cliente para sua plataforma e, em seguida, ativar o cluster. Para obter instruções detalhadas, consulte [Conceitos básicos do AWS CloudHSM](#) no Guia do usuário do AWS CloudHSM .

- O TLS mútuo (mTLS) não está habilitado.

O KMS não é compatível com mTLS para clusters. Essa configuração não deve ser habilitada.

- O cluster deve estar na mesma conta e região do armazenamento de AWS CloudHSM chaves. Você não pode associar um armazenamento de AWS CloudHSM chaves em uma região a um cluster em uma região diferente. Para criar uma infraestrutura chave em várias regiões, você deve criar armazenamentos e clusters de AWS CloudHSM chaves em cada região.

- Não é possível associar o cluster a outro armazenamento personalizado de chaves na mesma conta e região. Cada armazenamento de AWS CloudHSM chaves na conta e na região deve estar associado a um AWS CloudHSM cluster diferente. Você não pode especificar um cluster que já esteja associado a um armazenamento de chaves personalizado, tampouco um cluster que compartilhe um histórico de backup com um cluster associado. Os clusters que compartilham um histórico de backup têm o mesmo certificado do cluster. Para visualizar o certificado de cluster de um cluster, use o AWS CloudHSM console ou a [DescribeClusters](#) operação.

Se você [fizer backup de um cluster do AWS CloudHSM em uma região diferente](#), ele será considerado um cluster diferente e você poderá associar o backup a um armazenamento personalizado de chaves na região dele. No entanto, as chaves KMS nos dois armazenamentos de chaves personalizadas não são interoperáveis, mesmo que tenham a mesma chave de apoio. AWS KMS vincula os metadados ao texto cifrado para que eles possam ser descryptografados somente pela chave KMS que os criptografou.

- O cluster deve ser configurado com [sub-redes privadas](#) em pelo menos duas zonas de disponibilidade na região. Como não AWS CloudHSM é compatível com todas as zonas de disponibilidade, recomendamos que você crie sub-redes privadas em todas as zonas de disponibilidade da região. Você não pode reconfigurar as sub-redes para um cluster existente, mas pode [criar um cluster a partir de um backup](#) com diferentes sub-redes na configuração do cluster.

 Important

Depois de criar seu armazenamento de AWS CloudHSM chaves, não exclua nenhuma das sub-redes privadas configuradas para seu AWS CloudHSM cluster. Se AWS KMS não conseguir encontrar todas as sub-redes na configuração do cluster, as tentativas de [conexão com o armazenamento de chaves personalizadas](#) falharão com um estado de erro de SUBNET_NOT_FOUND conexão. Para obter detalhes, consulte [Como corrigir uma falha de conexão](#).

- O [grupo de segurança do cluster](#) (cloudhsm-cluster-*<cluster-id>*-sg) deve incluir regras de entrada e regras de saída que permitam a passagem do tráfego TCP nas portas IPv4 2223-2225. O Source (Origem) nas regras de entrada e o Destination (Destino) nas regras de saída deve corresponder ao ID do grupo de segurança. Essas regras são definidas por padrão quando você cria o cluster. Não as exclua nem as altere.

- O cluster deve conter pelo menos dois ativos HSMs em diferentes zonas de disponibilidade. Para verificar o número de HSMs, use o AWS CloudHSM console ou a [DescribeClusters](#) operação. Se necessário, você pode [adicionar um HSM](#).

Localizar o certificado da âncora de confiança

Ao criar um armazenamento de chaves personalizado, você deve carregar o certificado de âncora confiável para o AWS CloudHSM cluster. AWS KMS precisa do certificado de âncora confiável para conectar o armazenamento de AWS CloudHSM chaves ao AWS CloudHSM cluster associado.

Cada AWS CloudHSM cluster ativo tem um certificado de âncora confiável. Ao [inicializar o cluster](#), você gera esse certificado, salve-o no `customerCA.crt` arquivo e copie-o em hosts que se conectam ao cluster.

Crie o usuário `kmsuser` criptográfico para AWS KMS

Para administrar seu armazenamento de AWS CloudHSM chaves, faça AWS KMS login na conta de [usuário `kmsuser` criptográfico](#) (CU) no cluster selecionado. Antes de criar seu armazenamento de AWS CloudHSM chaves, você deve criar a `kmsuser` UC. Então, ao criar seu armazenamento de AWS CloudHSM chaves, você fornece a senha `kmsuser` para AWS KMS. Sempre que você conectar o armazenamento de AWS CloudHSM chaves ao AWS CloudHSM cluster associado, AWS KMS efetua login como `kmsuser` e alterna a senha `kmsuser`.

Important

Não especifique a opção 2FA ao criar o CU do `kmsuser`. Se você fizer isso, AWS KMS não poderá fazer login e seu armazenamento de AWS CloudHSM chaves não poderá ser conectado a esse AWS CloudHSM cluster. Ao especificar o código de autenticação de dois fatores, você não pode desfazê-lo. Em vez disso, você deve excluir o CU e recriá-lo.

Observações

Os procedimentos a seguir usam a ferramenta de linha de comando do AWS CloudHSM Client SDK 5, a CLI do [CloudHSM](#). A CLI do CloudHSM substitui `key-handle` por `key-reference`.

Em 1º de janeiro de 2025, AWS CloudHSM encerrará o suporte às ferramentas de linha de comando do Client SDK 3, ao CloudHSM Management Utility (CMU) e ao Key Management Utility (KMU). Para obter mais informações sobre as diferenças entre as

ferramentas de linha de comando do Client SDK 3 e a ferramenta de linha de comando do Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) no Guia do usuário do AWS CloudHSM .

1. Siga os procedimentos de conceitos básicos conforme descritos no tópico [Getting started with CloudHSM Command Line Interface \(CLI\)](#) do Guia do usuário do AWS CloudHSM .
2. Use o comando [user create](#) para criar um usuário de criptografia chamado kmsuser.

A senha deve conter de 7 a 32 caracteres alfanuméricos. Ela diferencia maiúsculas de minúsculas e não pode conter caracteres especiais.

O exemplo de comando a seguir cria um usuário de criptografia kmsuser.

```
aws-cloudhsm > user create --username kmsuser --role crypto-user
Enter password:
Confirm password:
{
  "error_code": 0,
  "data": {
    "username": "kmsuser",
    "role": "crypto-user"
  }
}
```

Criar um novo repositório de AWS CloudHSM chaves

Depois de [montar os pré-requisitos](#), você pode criar um novo armazenamento de AWS CloudHSM chaves no AWS KMS console ou usando a operação. [CreateCustomKeyStore](#)

Usando o AWS KMS console

Ao criar um armazenamento de AWS CloudHSM chaves no Console de gerenciamento da AWS, você pode adicionar e criar os [pré-requisitos](#) como parte do seu fluxo de trabalho. No entanto, o processo é mais rápido quando eles são organizados com antecedência.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.

3. No painel de navegação, selecione Custom key stores (Repositórios de chaves personalizados), AWS CloudHSM key stores (Repositórios de chaves do).
4. Selecione Create key store (Criar armazenamento de chaves).
5. Digite um nome amigável para o armazenamento de chaves personalizado. O nome deve ser exclusivo entre todos os outros armazenamentos de chaves personalizados de sua conta.

 Important

Não inclua informações confidenciais ou sigilosas nesse campo. Esse campo pode ser exibido em texto simples em CloudTrail registros e outras saídas.

6. Selecione [um AWS CloudHSM cluster](#) para o armazenamento de AWS CloudHSM chaves. Ou, para criar um novo AWS CloudHSM cluster, escolha o link Criar um AWS CloudHSM cluster.

O menu exibe os AWS CloudHSM clusters em sua conta e região que ainda não estão associados a um armazenamento de AWS CloudHSM chaves. O cluster deve [cumprir os requisitos](#) para associação com um armazenamento de chaves personalizado.

7. Escolha Escolher arquivo e, em seguida, carregue o certificado de âncora confiável para o AWS CloudHSM cluster que você escolheu. Esse é o arquivo `customerCA.crt` que você criou ao [inicializar o cluster](#).
8. Insira a senha [do usuário de criptografia kmsuser](#) (CU) que você criou no cluster selecionado.
9. Escolha Criar.

Quando o procedimento for bem-sucedido, o novo armazenamento de AWS CloudHSM chaves aparecerá na lista de armazenamentos de AWS CloudHSM chaves na conta e na região. Se ele for malsucedido, será exibida uma mensagem de erro descrevendo o problema e fornecendo ajuda para corrigi-lo. Se precisar de ajuda adicional, consulte [Solucionar problemas de um armazenamento de chaves personalizado](#).

Se você tentar criar um armazenamento de AWS CloudHSM chaves com todos os mesmos valores de propriedade de um armazenamento de AWS CloudHSM chaves desconectado existente, AWS KMS não criará um novo armazenamento de AWS CloudHSM chaves e não gerará uma exceção nem exibirá um erro. Em vez disso, AWS KMS reconhece a duplicata como a provável consequência de uma nova tentativa e retorna a ID do armazenamento de chaves existente AWS CloudHSM .

Próximo: Os novos armazenamentos de AWS CloudHSM chaves não são conectados automaticamente. Antes de criar AWS KMS keys no armazenamento de AWS CloudHSM chaves,

you must [connect the storage of custom keys](#) to the AWS CloudHSM cluster associated.

Using the AWS KMS API

You can use the [CreateCustomKeyStore](#) operation to create a new AWS CloudHSM keys storage associated to an AWS CloudHSM cluster in the account and the region. These examples use the AWS Command Line Interface (AWS CLI), but you can use any compatible programming language.

The `CreateCustomKeyStore` operation requires the following parameter values.

- `CustomKeyStoreName` — A friendly name for the storage of custom keys that is exclusive in the account.

Important

Do not include confidential or sensitive information in this field. This field can be displayed in plain text in CloudTrail logs and other outputs.

- `CloudHsmClusterId` — The ID of the cluster of an AWS CloudHSM cluster that [meets the requirements](#) of an AWS CloudHSM keys storage.
- `KeyStorePassword` — The password of the `kmsuser` UC in the specified cluster.
- `TrustAnchorCertificate` — The content of the `customerCA.crt` file that you created when [initializing the cluster](#).

The example that follows uses a fictitious cluster ID. Before executing the command, replace it with a valid cluster ID.

```
$ aws kms create-custom-key-store
  --custom-key-store-name ExampleCloudHSMKeyStore \
  --cloud-hsm-cluster-id cluster-1a23b4cdefg \
  --key-store-password kmsPswd \
  --trust-anchor-certificate <certificate-goes-here>
```

If you are using the AWS CLI, you can specify the path to the certificate file instead of its content. In the example that follows, the `customerCA.crt` file is in the root directory.

```
$ 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
```

Quando a operação é bem-sucedida, `CreateCustomKeyStore` retorna o ID do armazenamento de chaves personalizado, conforme exibido na resposta de exemplo a seguir.

```
{  
  "CustomKeyStoreId": cks-1234567890abcdef0  
}
```

Se a operação falhar, corrija o erro indicado pela exceção e tente novamente. Para obter ajuda adicional, consulte [Solucionar problemas de um armazenamento de chaves personalizado](#).

Se você tentar criar um armazenamento de AWS CloudHSM chaves com todos os mesmos valores de propriedade de um armazenamento de AWS CloudHSM chaves desconectado existente, AWS KMS não criará um novo armazenamento de AWS CloudHSM chaves e não gerará uma exceção nem exibirá um erro. Em vez disso, AWS KMS reconhece a duplicata como a provável consequência de uma nova tentativa e retorna a ID do armazenamento de chaves existente AWS CloudHSM .

Próximo: Para usar o armazenamento de AWS CloudHSM chaves, [conecte-o ao AWS CloudHSM cluster](#).

Exibir uma loja de AWS CloudHSM chaves

Você pode ver os AWS CloudHSM principais armazenamentos em cada conta e região usando o AWS KMS console ou a [DescribeCustomKeyStores](#) operação.

Usando o AWS KMS console

Ao visualizar os armazenamentos de AWS CloudHSM chaves no Console de gerenciamento da AWS, você pode ver o seguinte:

- O nome e ID do armazenamento de chaves personalizado
- O ID do AWS CloudHSM cluster associado
- O número de HSMs no cluster
- O estado da conexão atual

Um valor de estado de conexão (Status) de Desconectado indica que o armazenamento de chaves personalizadas é novo e nunca foi conectado, ou foi intencionalmente [desconectado de](#) seu cluster. AWS CloudHSM No entanto, se suas tentativas de usar uma chave KMS em um armazenamento de chaves personalizadas conectado falharem, isso pode indicar um problema com o armazenamento de chaves personalizadas ou seu AWS CloudHSM cluster. Para obter ajuda, consulte [Como corrigir uma chave do KMS com falha](#).

Para visualizar os armazenamentos de AWS CloudHSM chaves em uma determinada conta e região, use o procedimento a seguir.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Repositórios de chaves personalizados), AWS CloudHSM key stores (Repositórios de chaves do).

Para personalizar a exibição, clique no ícone de engrenagem que aparece abaixo do botão Create key store (Criar armazenamento de chaves).

Usando a AWS KMS API

Para visualizar suas lojas de AWS CloudHSM chaves, use a [DescribeCustomKeyStores](#) operação. Por padrão, essa operação retorna todos os armazenamentos de chaves personalizados na conta e região. No entanto, você pode usar o parâmetro CustomKeyId ou CustomKeyName (mas não ambos) para limitar o resultado para um determinado armazenamento de chaves personalizado. Para armazenamentos de AWS CloudHSM chaves, a saída consiste no ID e nome do armazenamento de chaves personalizadas, no tipo de armazenamento de chaves personalizadas, na ID do AWS CloudHSM cluster associado e no estado da conexão. Se o estado da conexão indica um erro, o resultado também inclui um código de erro que descreve o motivo do erro.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Por exemplo, o comando a seguir retorna todos os armazenamentos de chaves personalizados na conta e região. Você pode usar os parâmetros Marker e Limit para percorrer os armazenamentos de chaves personalizados do resultado.

```
$ aws kms describe-custom-key-stores
```

O comando de exemplo a seguir usa o parâmetro `CustomKeyStoreName` para obter apenas o armazenamento de chaves personalizado com o nome amigável `ExampleCloudHSMKeyStore`. É possível usar o parâmetro `CustomKeyStoreName` ou `CustomKeyStoreId` (mas não ambos) em cada comando.

O exemplo de saída a seguir representa um armazenamento de AWS CloudHSM chaves conectado ao seu AWS CloudHSM cluster.

 Note

O `CustomKeyStoreType` campo foi adicionado à `DescribeCustomKeyStores` resposta para distinguir os armazenamentos de AWS CloudHSM chaves dos armazenamentos de chaves externos.

```
$ 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>"
    }
  ]
}
```

Um `ConnectionState` de `Disconnected` indica que um armazenamento de chaves personalizadas nunca foi conectado ou foi intencionalmente [desconectado de seu AWS CloudHSM cluster](#). No entanto, se as tentativas de usar uma chave KMS em um armazenamento de AWS CloudHSM chaves conectado falharem, isso pode indicar um problema com o armazenamento de AWS CloudHSM chaves ou seu AWS CloudHSM cluster. Para obter ajuda, consulte [Como corrigir uma chave do KMS com falha](#).

Quando o `ConnectionState` do armazenamento de chaves personalizado é `FAILED`, a resposta `DescribeCustomKeyStores` inclui um elemento `ConnectionErrorCode` que explica o motivo desse erro.

Por exemplo, no resultado a seguir, o valor `INVALID_CREDENTIALS` indica que a conexão do armazenamento de chaves personalizado falhou porque a senha [kmsuser é inválido](#). Para obter ajuda com relação a esta e outras falhas de erro de conexão, consulte [Solucionar problemas de um armazenamento de chaves personalizado](#).

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "INVALID_CREDENTIALS",
      "ConnectionState": "FAILED",
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleCloudHSMKeyStore",
      "CustomKeyType": "AWS_CLOUDHSM",
      "CreationDate": "1.499288695918E9",
      "TrustAnchorCertificate": "<certificate appears here>"
    }
  ]
}
```

Saiba mais:

- [Visualizar repositórios de chaves externos](#)
- [Identificar chaves KMS em lojas de AWS CloudHSM chaves](#)
- [Registrando chamadas de AWS KMS API com AWS CloudTrail](#)

Editar configurações de armazenamento de AWS CloudHSM chaves

Você pode alterar as configurações de um armazenamento de AWS CloudHSM chaves existente. O armazenamento de chaves personalizadas deve estar desconectado de seu AWS CloudHSM cluster.

Para editar as configurações do armazenamento de AWS CloudHSM chaves:

1. [Desconectar o armazenamento de chaves personalizado](#) do cluster do AWS CloudHSM .
[Enquanto o armazenamento de chaves personalizadas estiver desconectado, você não pode criar AWS KMS keys \(chaves KMS\) no armazenamento de chaves personalizadas e não pode usar as chaves KMS que ele contém para operações criptográficas.](#)
2. Edite uma ou mais das configurações do armazenamento de AWS CloudHSM chaves.

Você pode editar as seguintes configurações em um armazenamento de chaves personalizado:

O nome amigável do armazenamento de chaves personalizado.

Inserir um novo nome amigável. O novo nome deve ser exclusivo entre todos os armazenamentos de chaves personalizadas em sua Conta da AWS.

 Important

Não inclua informações confidenciais ou sigilosas nesse campo. Esse campo pode ser exibido em texto simples em CloudTrail registros e outras saídas.

O ID do AWS CloudHSM cluster associado.

Edite esse valor para substituir um AWS CloudHSM cluster relacionado pelo original. Você pode usar esse recurso para reparar um armazenamento de chaves personalizado se o AWS CloudHSM cluster for corrompido ou excluído.

Especifique um AWS CloudHSM cluster que compartilhe um histórico de backup com o cluster original e [atenda aos requisitos](#) de associação com um armazenamento de chaves personalizado, incluindo dois ativos HSMS em diferentes zonas de disponibilidade. Os clusters que compartilham um histórico de backup têm o mesmo certificado do cluster. Para visualizar o certificado de cluster de um cluster, use a [DescribeClusters](#) operação. Você não pode usar o recurso de edição para associar o armazenamento de chaves personalizado a um cluster do AWS CloudHSM não relacionado.

A senha atual do [kmsuser usuário de criptografia](#) (CU).

Informa AWS KMS a senha atual da kmsuser UC no AWS CloudHSM cluster. Essa ação não altera a senha da kmsuser UC no AWS CloudHSM cluster.

Se você alterar a senha da kmsuser UC no AWS CloudHSM cluster, use esse recurso para informar AWS KMS a nova kmsuser senha. Caso contrário, o AWS KMS não poderá fazer login no cluster, e todas as tentativas de conectar o armazenamento de chaves personalizado ao cluster falham.

3. [Reconectar o armazenamento de chaves personalizado](#) ao cluster do AWS CloudHSM .

Editar as configurações de repositório de chaves

Você pode editar suas configurações de armazenamento de AWS CloudHSM chaves no AWS KMS console ou usando a [UpdateCustomKeyStore](#) operação.

Usando o AWS KMS console

Ao editar um armazenamento de AWS CloudHSM chaves, você pode alterar qualquer um dos valores configuráveis.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Repositórios de chaves personalizados), AWS CloudHSM key stores (Repositórios de chaves do).
4. Escolha a linha do armazenamento de AWS CloudHSM chaves que você deseja editar.

Se o valor na coluna Connection state (Estado da conexão) não for Disconnected (Desconectado), você deverá desconectar o armazenamento de chaves personalizado para editá-lo. (No menu Key store actions (Ações do armazenamento de chaves), escolha Disconnect [Desconectar].)

Enquanto um armazenamento de AWS CloudHSM chaves está desconectado, você pode gerenciar o armazenamento de AWS CloudHSM chaves e suas chaves KMS, mas não pode criar ou usar chaves KMS no AWS CloudHSM armazenamento de chaves.

5. No menu Key store actions (Ações do armazenamento de chaves), escolha Edit (Editar).
6. Faça uma ou mais das ações a seguir.
 - Digite um novo nome amigável para o armazenamento de chaves personalizado.
 - Digite o ID do cluster de um AWS CloudHSM cluster relacionado.
 - Digite a senha atual do usuário kmsuser criptográfico no AWS CloudHSM cluster associado.
7. Escolha Salvar.

Se o procedimento for bem-sucedido, uma mensagem descreverá as configurações que você editou. Se for malsucedido, será exibida uma mensagem de erro descrevendo o problema e fornecendo ajuda para corrigi-lo. Se precisar de ajuda adicional, consulte [Solucionar problemas de um armazenamento de chaves personalizado](#).

8. [Reconecte o armazenamento de chaves personalizado](#).

Para usar o armazenamento de AWS CloudHSM chaves, você deve reconectá-lo após a edição. Você pode deixar o armazenamento de chaves do AWS CloudHSM desconectado. [Mas enquanto estiver desconectado, você não poderá criar chaves KMS no armazenamento de chaves nem usar as AWS CloudHSM chaves KMS no armazenamento de chaves em operações AWS CloudHSM criptográficas.](#)

Usando a AWS KMS API

Para alterar as propriedades de um armazenamento de AWS CloudHSM chaves, use a [UpdateCustomKeyStore](#) operação. Você pode alterar várias propriedades de um armazenamento de chaves personalizado no mesmo comando. Se a operação for bem-sucedida, AWS KMS retornará uma resposta HTTP 200 e um objeto JSON sem propriedades. Para verificar se as alterações são efetivas, use a [DescribeCustomKeyStores](#) operação.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Comece usando [DisconnectCustomKeyStore](#) para [desconectar o armazenamento de chaves personalizadas](#) de seu AWS CloudHSM cluster. Substitua o exemplo de ID de armazenamento de chaves personalizado, cks-1234567890abcdef0, por um ID real.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

O primeiro exemplo usa [UpdateCustomKeyStore](#) para alterar o nome amigável do armazenamento de AWS CloudHSM chaves para DevelopmentKeys. O comando usa o CustomKeyId parâmetro para identificar o armazenamento de AWS CloudHSM chaves e CustomKeyName para especificar o novo nome para o armazenamento de chaves personalizado.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-custom-key-store-name DevelopmentKeys
```

O exemplo a seguir altera o cluster associado a um armazenamento de AWS CloudHSM chaves para outro backup do mesmo cluster. O comando usa o CustomKeyId parâmetro para identificar o armazenamento de AWS CloudHSM chaves e o CloudHsmClusterId parâmetro para especificar o novo ID do cluster.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

O exemplo a seguir indica AWS KMS que a kmsuser senha atual é `ExamplePassword`. O comando usa o `CustomKeyStoreId` parâmetro para identificar o armazenamento de AWS CloudHSM chaves e o `KeyStorePassword` parâmetro para especificar a senha atual.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-store-password ExamplePassword
```

O comando final reconecta o armazenamento de AWS CloudHSM chaves ao AWS CloudHSM cluster. É possível deixar o armazenamento de chaves personalizado no estado desconectado, mas ele precisa estar conectado para que seja possível criar chaves do KMS ou usar as chaves do KMS existentes para [operações de criptografia](#). Substitua o ID de exemplo do armazenamento de chaves personalizado por um ID real.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Connect a um armazenamento de AWS CloudHSM chaves

Os novos armazenamentos de AWS CloudHSM chaves não estão conectados. Antes de criar e usar AWS KMS keys em seu armazenamento de AWS CloudHSM chaves, você precisa conectá-lo ao AWS CloudHSM cluster associado. Você pode conectar e desconectar seu armazenamento de AWS CloudHSM chaves a qualquer momento e [visualizar seu estado de conexão](#).

Você não precisa conectar seu armazenamento de AWS CloudHSM chaves. Você pode deixar um armazenamento de AWS CloudHSM chaves em um estado desconectado indefinidamente e conectá-lo somente quando precisar usá-lo. No entanto, você pode desejar testar a conexão periodicamente para verificar se as configurações estão corretas e se ele pode ser conectado.

Note

AWS CloudHSM os armazenamentos de chaves têm um estado de DISCONNECTED conexão somente quando o armazenamento de chaves nunca foi conectado ou quando você o desconecta explicitamente. Se o estado da conexão do seu armazenamento de AWS CloudHSM chaves for, CONNECTED mas você estiver tendo problemas para usá-lo, verifique se o AWS CloudHSM cluster associado está ativo e contém pelo menos um ativo HSMs. Para obter ajuda com falhas de conexão, consulte [the section called “Solucionar problemas de um armazenamento de chaves personalizado”](#).

Quando você conecta um armazenamento de AWS CloudHSM chaves, AWS KMS encontra o AWS CloudHSM cluster associado, se conecta a ele, faz login no AWS CloudHSM cliente como [usuário kmsuser criptográfico \(UC\)](#) e, em seguida, alterna a kmsuser senha. AWS KMS permanece conectado ao AWS CloudHSM cliente enquanto o armazenamento de AWS CloudHSM chaves estiver conectado.

Para estabelecer a conexão, AWS KMS cria um [grupo de segurança](#) nomeado kms-*<custom key store ID>* na nuvem privada virtual (VPC) do cluster. O grupo de segurança tem uma única regra que permite o tráfego de entrada do grupo de segurança do cluster. AWS KMS também cria uma [interface de rede elástica](#) (ENI) em cada zona de disponibilidade da sub-rede privada do cluster. AWS KMS adiciona o ENIs ao grupo kms-*<cluster ID>* de segurança e ao grupo de segurança do cluster. A descrição de cada ENI é KMS managed ENI for cluster *<cluster-ID>*.

O processo de conexão pode demorar um período prolongado para ser concluído; até 20 minutos.

Antes de conectar o armazenamento de AWS CloudHSM chaves, verifique se ele atende aos requisitos.

- Seu AWS CloudHSM cluster associado deve conter pelo menos um HSM ativo. Para encontrar o número de HSMs no cluster, visualize o cluster no AWS CloudHSM console ou use a [DescribeClusters](#) operação. Se necessário, você pode [adicionar um HSM](#).
- O cluster deve ter uma conta de [usuário kmsuser criptográfico](#) (UC), mas essa UC não pode ser conectada ao cluster quando você conecta o AWS CloudHSM armazenamento de chaves. Para obter ajuda com o logout, consulte [Como fazer logout e se conectar novamente](#).
- O estado da conexão do armazenamento de AWS CloudHSM chaves não pode ser DISCONNECTING ou FAILED. Para ver o estado da conexão, use o AWS KMS console ou a [DescribeCustomKeyStores](#) resposta. Se o estado da conexão for FAILED, desconecte o armazenamento de chaves personalizado, corrija o problema e conecte-o.

Para obter ajuda com falhas de conexão, consulte [Como corrigir uma falha de conexão](#).

Quando seu armazenamento de AWS CloudHSM chaves está conectado, você pode [criar chaves KMS nele](#) e usar chaves KMS existentes em operações [criptográficas](#).

Conecte-se e reconecte-se ao seu armazenamento de AWS CloudHSM chaves

Você pode conectar ou reconectar seu armazenamento de AWS CloudHSM chaves no AWS KMS console ou usando a [ConnectCustomKeyStore](#) operação.

Usando o AWS KMS console

Para conectar um armazenamento de AWS CloudHSM chaves no Console de gerenciamento da AWS, comece selecionando o armazenamento de AWS CloudHSM chaves na página Armazenamentos de chaves personalizadas. O processo de conexão pode levar até 20 minutos.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Repositórios de chaves personalizados), AWS CloudHSM key stores (Repositórios de chaves do).
4. Escolha a linha do armazenamento de AWS CloudHSM chaves que você deseja conectar.

Se o estado de conexão do armazenamento de AWS CloudHSM chaves for Falha, você deverá [desconectar o armazenamento de chaves personalizado](#) antes de conectá-lo.

5. No menu Key store actions (Ações do armazenamento de chaves), escolha Connect (Conectar).

AWS KMS inicia o processo de conexão do seu armazenamento de chaves personalizadas. Ele localiza o cluster do AWS CloudHSM associado, cria a infraestrutura de rede necessária, se conecta, faz login no cluster do AWS CloudHSM como CU do kmsuser e muda a senha do kmsuser. Quando a operação é concluída, o estado de conexão é alterado para Connected (Conectado).

Se houver falha na operação, uma mensagem descrevendo o motivo da falha será exibida. Antes de tentar se conectar novamente, [veja o estado da conexão](#) do seu armazenamento de AWS CloudHSM chaves. Se for Failed (Falha), você deve [desconectar o armazenamento de chaves personalizado](#) antes de conectá-lo novamente. Se precisar de ajuda, consulte [Solucionar problemas de um armazenamento de chaves personalizado](#).

Próximo: [the section called “Criar uma chave KMS em um armazenamento de AWS CloudHSM chaves”](#).

Usando a AWS KMS API

Para conectar um armazenamento de AWS CloudHSM chaves desconectado, use a [ConnectCustomKeyStore](#) operação. O AWS CloudHSM cluster associado deve conter pelo menos um HSM ativo e o estado da conexão não pode ser FAILED.

O processo de conexão demora um período prolongado para ser concluído; até 20 minutos. A menos que se antecipe à falha, a operação retornará uma resposta HTTP 200 e um objeto JSON

sem propriedades. No entanto, essa resposta inicial não indica que a conexão foi bem-sucedida. Para determinar o estado da conexão do armazenamento de chaves personalizadas, veja a [DescribeCustomKeyStores](#) resposta.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Para identificar o armazenamento de AWS CloudHSM chaves, use seu ID de armazenamento de chaves personalizado. Você pode encontrar o ID na página Armazenamentos de chaves personalizadas no console ou usando a [DescribeCustomKeyStores](#) operação sem parâmetros. Antes de executar esse exemplo, substitua o ID de exemplo por um válido.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Para verificar se o armazenamento de AWS CloudHSM chaves está conectado, use a [DescribeCustomKeyStores](#) operação. Por padrão, essa operação retorna todos os armazenamentos de chaves personalizados em sua conta e região. No entanto, você pode usar o parâmetro CustomKeyId ou CustomKeyName (mas não ambos) para limitar a resposta para determinados armazenamentos de chaves personalizados. O valor ConnectionState CONNECTED indica que o armazenamento de chaves personalizado está conectado ao cluster do AWS CloudHSM .

 Note

O CustomKeyStoreType campo foi adicionado à DescribeCustomKeyStores resposta para distinguir os armazenamentos de AWS CloudHSM chaves dos armazenamentos de chaves externos.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleCloudHSMKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
    }
  ]
}
```



```
],  
}
```

Se ocorrer falha no valor `ConnectionState`, o elemento `ConnectionErrorCode` indica o motivo da falha. Nesse caso, não AWS KMS foi possível encontrar um AWS CloudHSM cluster na sua conta com o ID do cluster `cluster-1a23b4cdefg`. Se você excluiu o cluster, você pode [restaurá-lo a partir de um backup](#) do cluster original e [editar o ID do cluster](#) para o armazenamento de chaves personalizado. Para obter ajuda para responder a um código de erro de conexão, consulte [Como corrigir uma falha de conexão](#).

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

Desconectar um armazenamento de AWS CloudHSM chaves

Quando você desconecta um armazenamento de AWS CloudHSM chaves, AWS KMS sai do AWS CloudHSM cliente, se desconecta do AWS CloudHSM cluster associado e remove a infraestrutura de rede criada para suportar a conexão.

Enquanto um armazenamento de AWS CloudHSM chaves está desconectado, você pode gerenciar o armazenamento de AWS CloudHSM chaves e suas chaves KMS, mas não pode criar ou usar chaves KMS no AWS CloudHSM armazenamento de chaves. O estado da conexão do armazenamento de chaves é `DISCONNECTED`, e o [estado da chave](#) das chaves do KMS no armazenamento de chaves personalizado é `Unavailable`, a menos que elas estejam com `PendingDeletion`. Você pode reconectar o armazenamento de AWS CloudHSM chaves a qualquer momento.

 Note

AWS CloudHSM os armazenamentos de chaves têm um estado de DISCONNECTED conexão somente quando o armazenamento de chaves nunca foi conectado ou quando você o desconecta explicitamente. Se o estado da conexão do seu armazenamento de AWS CloudHSM chaves for, CONNECTED mas você estiver tendo problemas para usá-lo, verifique se o AWS CloudHSM cluster associado está ativo e contém pelo menos um ativo HSMs. Para obter ajuda com falhas de conexão, consulte [the section called “Solucionar problemas de um armazenamento de chaves personalizado”](#).

Quando você desconecta um armazenamento de chaves personalizado, as chaves do KMS do armazenamento de chaves tornam-se inutilizáveis imediatamente (sujeitas a consistência posterior). Porém, os recursos criptografados com [chaves de dados](#) protegidas pela chave do KMS não serão afetados até que a chave do KMS seja usada novamente, por exemplo, para descriptografar a chave de dados. Esse problema afeta os Serviços da AWS, pois muitos deles usam chaves de dados para proteger recursos. Para obter detalhes, consulte [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#).

 Note

Enquanto um armazenamento de chaves personalizado estiver desconectado, todas as tentativas de criar chaves do KMS nele ou de usar chaves do KMS existentes em operações de criptografia falharão. Essa ação pode impedir que os usuários armazenem e acessem dados sigilosos.

Para avaliar melhor o efeito de desconectar o armazenamento de chaves personalizado, [identifique as chaves do KMS](#) no armazenamento de chaves personalizado e [determine seu uso anterior](#).

Você pode desconectar um armazenamento de AWS CloudHSM chaves por motivos como os seguintes:

- Para mudar a senha do **kmsuser**. O AWS KMS altera a senha kmsuser toda vez que se conecta ao cluster do AWS CloudHSM . Para forçar uma mudança de senha, basta desconectar e reconectar.
- Para auditar o material chave das chaves KMS no AWS CloudHSM cluster. Quando você desconecta o armazenamento de chaves personalizadas, AWS KMS sai da conta de [usuário](#)

[kmsuser](#) criptográfica no AWS CloudHSM cliente. Isso permite fazer login no cluster como o CU `kmsuser` e auditar e gerenciar o material de chave para a chave do KMS.

- Para desabilitar imediatamente todas as chaves do KMS no armazenamento de chaves do AWS CloudHSM. Você pode [desativar e reativar as chaves KMS](#) em um armazenamento de AWS CloudHSM chaves usando a operação Console de gerenciamento da AWS ou. [DisableKey](#) Essas operações são concluídas rapidamente, mas atuam em uma chave do KMS de cada vez. Desconectar o armazenamento de AWS CloudHSM chaves altera imediatamente o estado da chave de todas as chaves KMS no armazenamento de AWS CloudHSM chaves para `Unavailable`, o que impede que elas sejam usadas em qualquer operação criptográfica.
- Para reparar uma falha na tentativa de conexão. Se uma tentativa de conectar um armazenamento de AWS CloudHSM chaves falhar (o estado de conexão do armazenamento de chaves personalizado é `FAILED`), você deve desconectar o armazenamento de AWS CloudHSM chaves antes de tentar conectá-lo novamente.

Desconecte seu armazenamento de AWS CloudHSM chaves

Você pode desconectar seu armazenamento de AWS CloudHSM chaves no AWS KMS console ou usando a [DisconnectCustomKeyStore](#) operação.

Desconecte usando o console AWS KMS

Para desconectar um armazenamento de AWS CloudHSM chaves conectado no AWS KMS console, comece escolhendo o armazenamento de AWS CloudHSM chaves na página Armazenamentos de chaves personalizadas.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Repositórios de chaves personalizados), AWS CloudHSM key stores (Repositórios de chaves do).
4. Escolha a linha do armazenamento de chaves externas que deseja desconectar.
5. No menu Key store actions (Ações do armazenamento de chaves), escolha Disconnect (Desconectar).

Quando a operação é concluída, o estado de conexão é alterado de Disconnecting (Desconectando) para Disconnected (Desconectado). Se ocorrer falha na operação, será exibida uma mensagem

de erro descrevendo o problema e fornecendo ajuda para corrigi-lo. Se precisar de ajuda adicional, consulte [Solucionar problemas de um armazenamento de chaves personalizado](#).

Desconecte-se usando a API AWS KMS

Para desconectar um armazenamento de AWS CloudHSM chaves conectado, use a [DisconnectCustomKeyStore](#) operação. Se a operação for bem-sucedida, AWS KMS retornará uma resposta HTTP 200 e um objeto JSON sem propriedades.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Este exemplo desconecta um armazenamento de AWS CloudHSM chaves. Antes de executar esse exemplo, substitua o ID de exemplo por um válido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Para verificar se o armazenamento de AWS CloudHSM chaves está desconectado, use a [DescribeCustomKeyStores](#) operação. Por padrão, essa operação retorna todos os armazenamentos de chaves personalizados em sua conta e região. No entanto, você pode usar o parâmetro CustomKeyId e CustomKeyName (mas não ambos) para limitar a resposta para determinados armazenamentos de chaves personalizados. O ConnectionState valor de DISCONNECTED indica que esse exemplo de armazenamento de AWS CloudHSM chaves não está conectado ao AWS CloudHSM cluster.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "1.499288695918E9",
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleKeyStore",
      "CustomKeyType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>"
    }
  ],
}
```

Excluir um armazenamento de AWS CloudHSM chaves

Quando você exclui um armazenamento de AWS CloudHSM chaves, AWS KMS exclui todos os metadados sobre o armazenamento de AWS CloudHSM chaves do KMS, incluindo informações sobre sua associação com um cluster. AWS CloudHSM Essa operação não afeta o AWS CloudHSM cluster HSMs, nem seus usuários. Você pode criar um novo armazenamento de AWS CloudHSM chaves associado ao mesmo AWS CloudHSM cluster, mas não pode desfazer a operação de exclusão.

Você só pode excluir um armazenamento de AWS CloudHSM chaves que esteja desconectado de seu AWS CloudHSM cluster e não contenha nenhum AWS KMS keys. Antes de excluir um armazenamento de chaves personalizado, faça o seguinte.

- Verifique se você ainda precisará usar qualquer uma das chaves do KMS no armazenamento de chaves para [operações de criptografia](#). Em seguida, [programe a exclusão](#) de todas as chaves do KMS do armazenamento de chaves. Para obter ajuda para encontrar as chaves KMS em um armazenamento de AWS CloudHSM chaves, consulte [Encontre as chaves KMS em um armazenamento de AWS CloudHSM chaves](#).
- Confirme se todas as chaves do KMS foram excluídas. Para visualizar as chaves KMS em um armazenamento de AWS CloudHSM chaves, consulte [the section called “Identificar chaves KMS em lojas de AWS CloudHSM chaves”](#).
- [Desconecte o armazenamento de AWS CloudHSM chaves](#) de seu AWS CloudHSM cluster.

Em vez de excluir o armazenamento de AWS CloudHSM chaves, considere [desconectá-lo](#) do cluster associado AWS CloudHSM . Enquanto um armazenamento de AWS CloudHSM chaves está desconectado, você pode gerenciar o armazenamento de AWS CloudHSM chaves e seu AWS KMS keys. Mas você não pode criar ou usar chaves KMS no armazenamento de AWS CloudHSM chaves. Você pode reconectar o armazenamento de AWS CloudHSM chaves a qualquer momento.

Exclua seu armazenamento de AWS CloudHSM chaves

Você pode excluir seu armazenamento de AWS CloudHSM chaves no AWS KMS console ou usando a [DeleteCustomKeyStore](#) operação.

Usando o AWS KMS console

Para excluir um armazenamento de AWS CloudHSM chaves no Console de gerenciamento da AWS, comece selecionando o armazenamento de AWS CloudHSM chaves na página Armazenamentos de chaves personalizadas.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Repositórios de chaves personalizados), AWS CloudHSM key stores (Repositórios de chaves do).
4. Encontre a linha que representa o armazenamento de AWS CloudHSM chaves que você deseja excluir. Se o estado de conexão do armazenamento de AWS CloudHSM chaves não for Desconectado, você deverá [desconectar o armazenamento de AWS CloudHSM chaves](#) antes de excluí-lo.
5. No menu Key store actions (Ações de armazenamento de chaves), escolha Delete (Excluir).

Quando a operação for concluída, uma mensagem de sucesso será exibida e o armazenamento de AWS CloudHSM chaves não aparecerá mais na lista de armazenamentos de chaves. Se a operação não for bem-sucedida, será exibida uma mensagem de erro descrevendo o problema e fornecendo ajuda para corrigi-lo. Se precisar de ajuda adicional, consulte [Solucionar problemas de um armazenamento de chaves personalizado](#).

Usando a AWS KMS API

Para excluir um armazenamento de AWS CloudHSM chaves, use a [DeleteCustomKeyStore](#) operação. Se a operação for bem-sucedida, AWS KMS retornará uma resposta HTTP 200 e um objeto JSON sem propriedades.

Para começar, verifique se o armazenamento de AWS CloudHSM chaves não contém nenhuma AWS KMS keys. Não é possível excluir um armazenamento de chaves personalizado que contém chaves do KMS. O primeiro exemplo de comando usa [ListKeyse DescribeKey](#) para pesquisar AWS KMS keys no armazenamento de AWS CloudHSM chaves com o exemplo de ID de armazenamento de chaves `cks-1234567890abcdef0` personalizado. Nesse caso, o comando não retorna chaves do KMS. Se isso acontecer, use a [ScheduleKeyDeletion](#) operação para programar a exclusão de cada uma das chaves do KMS.

Bash

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;  
do aws kms describe-key --key-id $key |  
grep '"CustomKeyIdStoreId": "cks-1234567890abcdef0"' --context 100; done
```

PowerShell

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyIdStoreId -eq  
'cks-1234567890abcdef0'
```

Em seguida, desconecte o armazenamento de AWS CloudHSM chaves. Este exemplo de comando usa a [DisconnectCustomKeyStore](#) operação para desconectar um armazenamento de AWS CloudHSM chaves de seu AWS CloudHSM cluster. Antes de executar um comando como esse, substitua o ID de exemplo do armazenamento de chaves personalizado por um válido.

Bash

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Disconnect-KMSCustomKeyStore -CustomKeyIdStoreId cks-1234567890abcdef0
```

Depois que o armazenamento de chaves personalizadas for desconectado, você poderá usar a [DeleteCustomKeyStore](#) operação para excluí-lo.

Bash

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Remove-KMSCustomKeyStore -CustomKeyIdStoreId cks-1234567890abcdef0
```

Solucionar problemas de um armazenamento de chaves personalizado

AWS CloudHSM as principais lojas são projetadas para estarem disponíveis e serem resilientes. No entanto, existem algumas condições de erro que talvez você precise reparar para manter seu armazenamento de AWS CloudHSM chaves operacional.

Tópicos

- [Como corrigir chaves do KMS indisponíveis](#)
- [Como corrigir uma chave do KMS com falha](#)
- [Como corrigir uma falha de conexão](#)
- [Como responder a uma falha de operação criptográfica](#)
- [Como corrigir credenciais kmsuser inválidas](#)
- [Como excluir material de chave órfã](#)
- [Como recuperar materiais de chave excluídos de uma chave do KMS](#)
- [Como fazer login como kmsuser](#)

Como corrigir chaves do KMS indisponíveis

Normalmente, o [estado AWS CloudHSM chave](#) AWS KMS keys de um armazenamento de chaves é `Enabled`. Como todas as chaves KMS, o estado da chave muda quando você desativa as chaves KMS em um armazenamento de AWS CloudHSM chaves ou as agenda para exclusão. No entanto, ao contrário de outras chaves do KMS, as chaves do KMS em um armazenamento de chaves personalizado também podem ter um [estado de chave](#) `Unavailable`.

Um estado de chave `Unavailable` indica que a chave do KMS está em um armazenamento de chaves personalizado que foi [intencionalmente desconectado](#) e que as tentativas de reconectá-lo, se houver, não foram bem-sucedidas. Enquanto uma chave do KMS está indisponível, é possível visualizá-la e gerenciá-la, mas não é possível usá-la para [operações de criptografia](#).

Para encontrar o estado de chave de uma chave do KMS, na página Customer managed keys (Chaves gerenciadas pelo cliente), visualize o campo Status da chave do KMS. Ou use a [DescribeKey](#) operação e visualize o `KeyState` elemento na resposta. Para obter detalhes, consulte [Identificar e visualizar chaves](#).

As chaves do KMS em um armazenamento de chaves personalizado desconectado terão o estado de chave `Unavailable` ou `PendingDeletion`. As chaves do KMS agendadas para exclusão

de um armazenamento de chaves personalizado têm um estado de chave Pending Deletion, mesmo quando o armazenamento de chaves personalizado está desconectado. Isso permite que você cancele a programação da exclusão de chaves sem reconectar o armazenamento de chaves personalizado.

Para corrigir uma chave do KMS indisponível, [reconecte o armazenamento de chaves personalizado](#). Após a reconexão do armazenamento de chaves personalizado, o estado de chave das chaves do KMS nesse armazenamento é automaticamente restaurado ao estado anterior, como Enabled ou Disabled. Chaves do KMS com exclusão pendente permanecem no estado PendingDeletion. No entanto, enquanto o problema persistir, [habilitar ou desabilitar uma chave do KMS](#) indisponível não altera seu estado de chave. A ação habilitar ou desabilitar entra em vigor somente quando a chave é disponibilizada.

Para obter ajuda com conexões com falha, consulte [Como corrigir uma falha de conexão](#).

Como corrigir uma chave do KMS com falha

Problemas com a criação e o uso de chaves KMS em AWS CloudHSM armazenamentos de chaves podem ser causados por um problema com seu armazenamento de AWS CloudHSM chaves, seu AWS CloudHSM cluster associado, a chave KMS ou seu material de chaves.

Quando um armazenamento de AWS CloudHSM chaves é desconectado de seu AWS CloudHSM cluster, o estado da chave das chaves KMS no armazenamento de chaves personalizadas é. Unavailable Todas as solicitações para criar chaves KMS em um armazenamento de AWS CloudHSM chaves desconectado retornam uma CustomKeyStoreInvalidStateException exceção. Todas as solicitações para criptografar, descriptografar e criptografar novamente ou gerar chaves de dados retornam uma exceção KMSInvalidStateException. Para corrigir o problema, [reconecte o armazenamento de AWS CloudHSM chaves](#).

No entanto, suas tentativas de usar uma chave KMS em um armazenamento de AWS CloudHSM chaves para [operações criptográficas](#) podem falhar mesmo quando o estado da chave é Enabled e o estado da conexão do armazenamento de AWS CloudHSM chaves é. Connected Isso pode ser causado por uma das seguintes condições.

- O material de chave para a chave do KMS pode ter sido excluído do cluster do AWS CloudHSM associado. Para investigar, [localize o id de chave](#) do material de chave para uma chave do KMS e, se necessário, tente [recuperar o material de chave](#).
- Todos HSMs foram excluídos do AWS CloudHSM cluster associado ao armazenamento de AWS CloudHSM chaves. Para usar uma chave KMS em um armazenamento de AWS CloudHSM

chaves em uma operação criptográfica, seu AWS CloudHSM cluster deve conter pelo menos um HSM ativo. Para verificar o número e o estado de HSMs em um AWS CloudHSM cluster, [use o AWS CloudHSM console](#) ou a [DescribeClusters](#) operação. Para adicionar um HSM ao cluster, use o AWS CloudHSM console ou a [CreateHsm](#) operação.

- O AWS CloudHSM cluster associado ao armazenamento de AWS CloudHSM chaves foi excluído. Para corrigir o problema, [crie um cluster a partir de um backup](#) relacionado ao cluster original, como um backup do cluster original, ou um backup que foi usado para criar o cluster original. [Edite o ID do cluster](#) nas configurações do armazenamento de chaves personalizado. Para instruções, consulte [Como recuperar materiais de chave excluídos de uma chave do KMS](#).
- O AWS CloudHSM cluster associado ao armazenamento de chaves personalizadas não tinha nenhuma sessão PKCS #11 disponível. Isso geralmente ocorre durante períodos de alto tráfego de intermitência quando sessões adicionais são necessárias para atender ao tráfego. Para responder a um `KMSInternalException` com uma mensagem de erro sobre as sessões do PKCS #11, volte e repita a solicitação.

Como corrigir uma falha de conexão

Se você tentar [conectar um armazenamento de AWS CloudHSM chaves](#) ao AWS CloudHSM cluster, mas a operação falhar, o estado da conexão do armazenamento de AWS CloudHSM chaves mudará para `FAILED`. Para encontrar o estado da conexão de um armazenamento de AWS CloudHSM chaves, use o AWS KMS console ou a [DescribeCustomKeyStores](#) operação.

Como alternativa, algumas tentativas de conexão falham rapidamente devido a erros de configuração de cluster facilmente detectados. Nesse caso, o estado da conexão ainda é `DISCONNECTED`. Essas falhas retornar uma mensagem de erro ou [exceção](#) que explica por que a tentativa falhou. Analise a descrição da exceção e [os requisitos do cluster](#), corrija o problema, [atualize o armazenamento de AWS CloudHSM chaves](#), se necessário, e tente se conectar novamente.

Quando o estado da conexão estiver `FAILED`, execute a [DescribeCustomKeyStores](#) operação e veja o `ConnectionErrorCode` elemento na resposta.

Note

Quando o estado de conexão de um armazenamento de AWS CloudHSM chaves é `FAILED`, você deve [desconectar o armazenamento de AWS CloudHSM chaves](#) antes de tentar reconectá-lo. Você não pode conectar um armazenamento de AWS CloudHSM chaves com um estado de `FAILED` conexão.

- **CLUSTER_NOT_FOUND** indica que AWS KMS não é possível encontrar um AWS CloudHSM cluster com o ID de cluster especificado. Isso pode ocorrer quando o ID de cluster errado é fornecido para uma operação de API ou o cluster é excluído e não substituído. Para corrigir esse erro, verifique o ID do cluster, por exemplo, usando o AWS CloudHSM console ou a [DescribeClusters](#) operação. Se o cluster foi excluído, [crie um cluster a partir de um backup recente](#) do original. Em seguida, [desconecte o armazenamento de AWS CloudHSM chaves](#), [edite a AWS CloudHSM configuração de ID do cluster de armazenamento](#) de chaves e [reconecte o armazenamento de AWS CloudHSM chaves](#) ao cluster.
- **INSUFFICIENT_CLOUDHSM_HSMS** indica que o AWS CloudHSM cluster associado não contém nenhum HSMs. Para se conectar, o cluster deve ter pelo menos um HSM. Para encontrar o número de HSMs no cluster, use a [DescribeClusters](#) operação. Para corrigir esse erro, [adicione pelo menos um HSM](#) ao cluster. Se você adicionar várias HSMs, é melhor criá-las em diferentes zonas de disponibilidade.
- **INSUFFICIENT_FREE_ADDRESSES_IN_SUBNET** indica que não AWS KMS foi possível conectar o armazenamento de AWS CloudHSM chaves ao AWS CloudHSM cluster porque pelo menos uma [sub-rede privada associada ao cluster](#) não tem nenhum endereço IP disponível. Uma conexão de armazenamento de AWS CloudHSM chaves requer um endereço IP gratuito em cada uma das sub-redes privadas associadas, embora duas sejam preferíveis.

[Não é possível adicionar endereços IP](#) (blocos CIDR) a uma sub-rede existente. Se possível, mova ou exclua outros recursos que estejam usando os endereços IP na sub-rede, como instâncias do EC2 não utilizadas ou interfaces elásticas de rede. Caso contrário, você pode [criar um cluster a partir de um backup recente](#) do AWS CloudHSM cluster com sub-redes privadas novas ou existentes que tenham [mais espaço de endereço livre](#). Em seguida, para associar o novo cluster ao seu armazenamento de AWS CloudHSM chaves, [desconecte o armazenamento de chaves personalizado](#), [altere o ID do cluster](#) do armazenamento de AWS CloudHSM chaves para o ID do novo cluster e tente se conectar novamente.

 Tip

Para evitar a [redefinição da kmsuser senha](#), use o backup mais recente do AWS CloudHSM cluster.

- **INTERNAL_ERROR** indica que não AWS KMS foi possível concluir a solicitação devido a um erro interno. Repetir a solicitação. Para `ConnectCustomKeyStore` solicitações, desconecte o armazenamento de AWS CloudHSM chaves antes de tentar se conectar novamente.

- **INVALID_CREDENTIALS** indica que AWS KMS não é possível fazer login no AWS CloudHSM cluster associado porque ele não tem a senha correta da `kmsuser` conta. Para obter ajuda com relação a esse erro, consulte [Como corrigir credenciais kmsuser inválidas](#).
- **NETWORK_ERRORS** geralmente indica problemas de rede temporários. [Desconecte o armazenamento de AWS CloudHSM chaves](#), aguarde alguns minutos e tente se conectar novamente.
- **SUBNET_NOT_FOUND** indica que pelo menos uma sub-rede na configuração do AWS CloudHSM cluster foi excluída. Se AWS KMS não conseguir encontrar todas as sub-redes na configuração do cluster, as tentativas de conectar o armazenamento de AWS CloudHSM chaves ao AWS CloudHSM cluster falharão.

Para corrigir esse erro, [crie um cluster a partir de um backup recente](#) do mesmo AWS CloudHSM cluster. (Esse processo cria uma configuração de cluster com uma VPC e sub-redes privadas.) Verifique se o novo cluster atende aos [requisitos para um armazenamento de chaves personalizado](#) e anote o novo ID do cluster. Em seguida, para associar o novo cluster ao seu armazenamento de AWS CloudHSM chaves, [desconecte o armazenamento de chaves personalizado](#), [altere o ID do cluster](#) do armazenamento de AWS CloudHSM chaves para o ID do novo cluster e tente se conectar novamente.

 Tip

Para evitar a [redefinição da kmsuser senha](#), use o backup mais recente do AWS CloudHSM cluster.

- **USER_LOCKED_OUT** indica que a [conta do usuário de criptografia \(CU\) kmsuser](#) está bloqueada do cluster do AWS CloudHSM associado devido ao excesso de tentativas de senha com falha. Para obter ajuda com relação a esse erro, consulte [Como corrigir credenciais kmsuser inválidas](#).

Para corrigir esse erro, [desconecte o armazenamento de AWS CloudHSM chaves e use o comando `user change-password`](#) na CLI do CloudHSM para alterar a senha da conta. `kmsuser` [Edite as configurações de senha kmsuser](#) do armazenamento de chaves personalizado e tente se conectar novamente. Para obter ajuda, use o procedimento descrito no tópico [Como corrigir credenciais kmsuser inválidas](#).

- **USER_LOGGED_IN** indica que a conta da `kmsuser` UC está conectada ao cluster associado AWS CloudHSM. Isso AWS KMS impede a rotação da senha da `kmsuser` conta e o login no cluster. Para corrigir esse erro, registre o CU `kmsuser` fora do cluster. Se você alterou a `kmsuser` senha para fazer login no cluster, também deverá atualizar o valor da senha do armazenamento de

chaves para o armazenamento de AWS CloudHSM chaves. Para obter ajuda, consulte [Como fazer logout e se conectar novamente](#).

- `USER_NOT_FOUND` indica que AWS KMS não é possível encontrar uma conta de `kmsuser` UC no AWS CloudHSM cluster associado. Para corrigir esse erro, [crie uma conta de `kmsuser` UC](#) no cluster e, em seguida, [atualize o valor da senha do armazenamento de chaves](#) para o armazenamento de AWS CloudHSM chaves. Para obter ajuda, consulte [Como corrigir credenciais `kmsuser` inválidas](#).

Como responder a uma falha de operação criptográfica

Uma operação criptográfica que usa uma chave do KMS em um armazenamento de chaves personalizado pode falhar com um `KMSInvalidStateException`. As mensagens de erro a seguir podem acompanhar o `KMSInvalidStateException`.

O KMS não pode se comunicar com seu cluster CloudHSM. Pode ser um problema de rede transitório. Se você ver esse erro repetidamente, verifique se as regras de rede ACLs e de grupo de segurança para a VPC do seu AWS CloudHSM cluster estão corretas.

- Embora este seja um erro HTTPS 400, ele pode ser resultante de problemas de rede transitórios. Para responder, comece tentando novamente a solicitação. No entanto, se o erro persistir, examine a configuração dos seus componentes de rede. O erro é causado provavelmente pela configuração incorreta de um componente de rede, como uma regra de firewall ou uma regra de grupo de segurança de VPC que está bloqueando o tráfego de saída. Por exemplo, o KMS não pode se comunicar com AWS CloudHSM clusters over IPv6. Para obter detalhes sobre pré-requisitos, consulte [Crie um armazenamento de AWS CloudHSM chaves](#).

O KMS não pode se comunicar com seu AWS CloudHSM cluster porque o `kmsuser` está bloqueado. Se você ver esse erro repetidamente, desconecte o armazenamento de AWS CloudHSM chaves e redefina a senha da conta `kmsuser`. Atualize a senha do `kmsuser` para o armazenamento de chaves personalizado e tente fazer a solicitação novamente.

- Essa mensagem de erro indica que a [conta do usuário de criptografia \(CU\) `kmsuser`](#) está bloqueada do cluster do AWS CloudHSM correspondente devido ao excesso de tentativas de

senha com falha. Para obter ajuda com relação a esse erro, consulte [Como desconectar e fazer login](#).

Como corrigir credenciais **kmsuser** inválidas

Quando você [conecta um armazenamento de AWS CloudHSM chaves](#), AWS KMS efetua login no AWS CloudHSM cluster associado como [usuário kmsuser criptográfico](#) (UC). Ele permanece conectado até que o armazenamento de AWS CloudHSM chaves seja desconectado. A resposta [DescribeCustomKeyStores](#) mostra um ConnectionState com valor FAILED e ConnectionErrorCode de INVALID_CREDENTIALS, conforme mostrado no exemplo a seguir.

Se você desconectar o armazenamento de AWS CloudHSM chaves e alterar a kmsuser senha, AWS KMS não poderá fazer login no AWS CloudHSM cluster com as credenciais da conta da kmsuser UC. Como resultado, todas as tentativas de conectar o armazenamento de AWS CloudHSM chaves falham. A resposta DescribeCustomKeyStores mostra um ConnectionState com valor FAILED e ConnectionErrorCode de INVALID_CREDENTIALS, conforme mostrado no exemplo a seguir.

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

Além disso, após cinco tentativas malsucedidas de efetuar login no cluster com uma senha incorreta, o AWS CloudHSM bloqueia a conta do usuário. Para efetuar login no cluster, você deve alterar a senha da conta.

Se AWS KMS receber uma resposta de bloqueio ao tentar fazer login no cluster como a kmsuser UC, a solicitação para conectar o armazenamento de AWS CloudHSM chaves falhará. A [DescribeCustomKeyStores](#) resposta inclui um ConnectionState de FAILED e um ConnectionErrorCode valor de USER_LOCKED_OUT, conforme mostrado no exemplo a seguir.

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

Para corrigir qualquer uma dessas condições, use o procedimento a seguir.

1. [Desconecte o armazenamento de AWS CloudHSM chaves.](#)
2. Execute a [DescribeCustomKeyStores](#) operação e visualize o valor do ConnectionErrorCode elemento na resposta.
 - Se o valor ConnectionErrorCode for INVALID_CREDENTIALS, determine a senha atual para a conta kmsuser. Se necessário, use o comando [user change-password](#) na CLI do CloudHSM para definir a senha com um valor conhecido.
 - Se o valor de ConnectionErrorCode for USER_LOCKED_OUT, você deverá usar o comando [user change-password](#) na CLI do CloudHSM para alterar a senha de kmsuser.
3. [Edite a configuração de senha kmsuser](#) para corresponder à senha kmsuser atual no cluster. Essa ação informa ao AWS KMS a senha a ser usada para fazer login no cluster. Ela não altera a senha kmsuser no cluster.
4. [Conectar o armazenamento de chaves personalizado](#)

Como excluir material de chave órfã

Depois de programar a exclusão de uma chave KMS de um armazenamento de AWS CloudHSM chaves, talvez seja necessário excluir manualmente o material de chave correspondente do cluster associado. AWS CloudHSM

Quando você cria uma chave KMS em um armazenamento de AWS CloudHSM chaves, AWS KMS cria os metadados da chave KMS AWS KMS e gera o material da chave no cluster associado. AWS CloudHSM Quando você agenda a exclusão de uma chave KMS em um armazenamento de

chaves, após o período de espera, AWS KMS exclui os metadados da AWS CloudHSM chave KMS. Em seguida, AWS KMS faça o possível para excluir o material de chave correspondente do AWS CloudHSM cluster. A tentativa pode falhar se AWS KMS não conseguir acessar o cluster, como quando ele é desconectado do armazenamento de AWS CloudHSM chaves ou quando a kmsuser senha é alterada. AWS KMS não tenta excluir o material chave dos backups do cluster.

AWS KMS relata os resultados de sua tentativa de excluir o material chave do cluster na entrada de DeleteKey eventos de seus AWS CloudTrail registros. Ela aparece no elemento backingKeysDeletionStatus do elemento additionalEventData, conforme mostrado na entrada de exemplo a seguir. A entrada também inclui o ARN da chave KMS, AWS CloudHSM o ID do cluster e o ID backing-key-id () do material da chave.

```
{
  "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"
}
```

Observações

Os procedimentos a seguir usam a ferramenta de linha de comando do AWS CloudHSM Client SDK 5, a CLI do [CloudHSM](#). A CLI do CloudHSM substitui `key-handle` por `key-reference`.

Em 1º de janeiro de 2025, AWS CloudHSM encerrará o suporte às ferramentas de linha de comando do Client SDK 3, ao CloudHSM Management Utility (CMU) e ao Key Management Utility (KMU). Para obter mais informações sobre as diferenças entre as ferramentas de linha de comando do Client SDK 3 e a ferramenta de linha de comando do Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) no Guia do usuário do AWS CloudHSM .

Os procedimentos a seguir demonstram como excluir o material de chave órfã do cluster associado AWS CloudHSM .

1. Desconecte o armazenamento de AWS CloudHSM chaves, se ele ainda não estiver desconectado, faça [login](#), conforme explicado em. [Como desconectar e fazer login](#)

Note

Enquanto um armazenamento de chaves personalizado estiver desconectado, todas as tentativas de criar chaves do KMS nele ou de usar chaves do KMS existentes em operações de criptografia falharão. Essa ação pode impedir que os usuários armazenem e acessem dados sigilosos.

2. Use o comando [key delete](#) na CLI do CloudHSM para excluir a chave HSMs do no cluster.

Todas as entradas de CloudTrail registro para operação criptográfica com uma chave KMS em um armazenamento de AWS CloudHSM chaves incluem um `additionalEventData` campo com e. `customKeyStoreId` `backingKey` O valor retornado no campo `backingKeyId` é o atributo `id` da chave do CloudHSM. Recomendamos filtrar a operação de exclusão de chaves `id` para excluir o material de chave órfã que você identificou em seus registros. CloudTrail

AWS CloudHSM reconhece o `backingKeyId` valor como um valor hexadecimal. Para filtrar por `id`, você deve prefixar `backingKeyId` com `0x`. Por exemplo, se o `backingKeyId` em seu CloudTrail registro for `1a2b3c45678abcdef`, você filtraria por `0x1a2b3c45678abcdef`.

O exemplo a seguir exclui uma chave do HSMs em seu cluster. O `backing-key-id` está listado na entrada do CloudTrail registro. Antes de executar esse comando, substitua o exemplo de `backing-key-id` por um elemento válido da sua conta.

```
aws-cloudhsm key delete --filter attr.id="0x<backing-key-id>"
{
  "error_code": 0,
  "data": {
    "message": "Key deleted successfully"
  }
}
```

3. Saia e reconecte o armazenamento de AWS CloudHSM chaves conforme descrito em [Como fazer logout e se conectar novamente](#).

Como recuperar materiais de chave excluídos de uma chave do KMS

Se o material da chave de um AWS KMS key for excluído, a chave KMS ficará inutilizável e todo o texto cifrado que foi criptografado sob a chave KMS não poderá ser descriptografado. Isso pode acontecer se o material de chave de uma chave KMS em um armazenamento de AWS CloudHSM chaves for excluído do AWS CloudHSM cluster associado. No entanto, há a possibilidade de recuperação do material de chaves.

Quando você cria uma AWS KMS key (chave KMS) em um armazenamento de AWS CloudHSM chaves, AWS KMS efetua login no AWS CloudHSM cluster associado e cria o material de chave para a chave KMS. Ele também altera a senha para um valor que somente ele conhece e permanece conectado enquanto o armazenamento de AWS CloudHSM chaves estiver conectado. Como

somente o proprietário da chave, ou seja, a UC que criou a chave, pode excluir a chave, é improvável que a chave seja excluída HSMs acidentalmente.

No entanto, se o material de chave de uma chave KMS for excluído do HSMs em um cluster, o estado da chave KMS eventualmente mudará para. `UNAVAILABLE` Se você tentar usar a chave do KMS para uma operação de criptografia, ocorrerá falha na operação com uma exceção `KMSInvalidStateException`. E, o que é mais importante, os dados que tiverem sido criptografados com a chave do KMS não poderão ser descriptografados.

Em determinadas circunstâncias, é possível recuperar o material de chaves excluído, [criando um cluster a partir de um backup](#) que contenha o material de chaves. Essa estratégia funciona somente quando pelo menos um backup foi criado enquanto a chave existia e antes de ter sido excluída.

Use o processo a seguir para recuperar o material de chaves.

1. Encontre um backup de cluster que contém o material de chaves. O backup também deve conter todos os usuários e chaves de que você precisa para oferecer suporte ao cluster e seus dados criptografados.

Use a [DescribeBackups](#) operação para listar os backups de um cluster. Use o timestamp do backup para ajudá-lo a selecionar um backup. Para limitar a saída ao cluster associado ao armazenamento de AWS CloudHSM chaves, use o `Filters` parâmetro, conforme mostrado no exemplo a seguir.

```
$ aws cloudhsmv2 describe-backups --filters clusterIds=<cluster ID>
{
  "Backups": [
    {
      "ClusterId": "cluster-1a23b4cdefg",
      "BackupId": "backup-9g87f6edcba",
      "CreateTimestamp": 1536667238.328,
      "BackupState": "READY"
    },
    ...
  ]
}
```

2. [Crie um cluster a partir de um backup selecionado](#). Verifique se o backup contém a chave excluída e outros usuários e chaves que o cluster requer.
3. [Desconecte o armazenamento de AWS CloudHSM chaves](#) para poder editar suas propriedades.

4. [Edite o ID do cluster](#) do armazenamento de AWS CloudHSM chaves. Insira o ID do cluster que você criou a partir do backup. Como o cluster compartilha um histórico de backup com o cluster original, o novo ID do cluster deve ser válido.
5. [Reconecte o armazenamento de AWS CloudHSM chaves](#).

Como fazer login como **kmsuser**

Para criar e gerenciar o material de chaves no AWS CloudHSM cluster para seu armazenamento de AWS CloudHSM chaves, AWS KMS use a [conta de usuário kmsuser criptográfico \(CU\)](#). Você [cria a conta da kmsuser UC](#) em seu cluster e fornece sua senha AWS KMS ao criar seu armazenamento de AWS CloudHSM chaves.

Em geral, AWS KMS gerencia a kmsuser conta. No entanto, para algumas tarefas, você precisa desconectar o armazenamento de AWS CloudHSM chaves, fazer login no cluster como kmsuser CU e usar a interface de [linha de comando \(CLI\) do CloudHSM](#).

Note

Enquanto um armazenamento de chaves personalizado estiver desconectado, todas as tentativas de criar chaves do KMS nele ou de usar chaves do KMS existentes em operações de criptografia falharão. Essa ação pode impedir que os usuários armazenem e acessem dados sigilosos.

Este tópico explica como [desconectar seu armazenamento de AWS CloudHSM chaves e fazer login como kmsuser](#), executar a ferramenta de linha de AWS CloudHSM comando, [sair e reconectar seu armazenamento de AWS CloudHSM chaves](#).

Tópicos

- [Como desconectar e fazer login](#)
- [Como fazer logout e se conectar novamente](#)

Como desconectar e fazer login

Use o procedimento a seguir sempre que precisar fazer login em um cluster associado como o usuário de criptografia kmsuser.

Observações

Os procedimentos a seguir usam a ferramenta de linha de comando do AWS CloudHSM Client SDK 5, a CLI do [CloudHSM](#). A CLI do CloudHSM substitui `key-handle` por `key-reference`.

Em 1º de janeiro de 2025, AWS CloudHSM encerrará o suporte às ferramentas de linha de comando do Client SDK 3, ao CloudHSM Management Utility (CMU) e ao Key Management Utility (KMU). Para obter mais informações sobre as diferenças entre as ferramentas de linha de comando do Client SDK 3 e a ferramenta de linha de comando do Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) no Guia do usuário do AWS CloudHSM.

1. Desconecte o armazenamento de AWS CloudHSM chaves, se ele ainda não estiver desconectado. Você pode usar o AWS KMS console ou a AWS KMS API.

Enquanto sua AWS CloudHSM chave está conectada, AWS KMS está logado como o `kmsuser`. Isso evita que você faça login como `kmsuser` ou altere a senha `kmsuser`.

Por exemplo, esse comando é usado [DisconnectCustomKeyStore](#) para desconectar um exemplo de armazenamento de chaves. Substitua o exemplo de ID de armazenamento de AWS CloudHSM chaves por um válido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

2. Use o comando `login` e faça login como administrador. Siga os procedimentos descritos na seção [Using CloudHSM CLI](#) do Guia do usuário do AWS CloudHSM.

```
aws-cloudhsm > login --username admin --role admin
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

3. Use o comando [user change-password](#) na CLI do CloudHSM para alterar a senha da `kmsuser` conta para uma que você conheça. (AWS KMS gira a senha quando você conecta seu

armazenamento de AWS CloudHSM chaves.) A senha deve conter de 7 a 32 caracteres alfanuméricos. Ela diferencia maiúsculas de minúsculas e não pode conter caracteres especiais.

4. Faça login como `kmsuser` usando a senha que você definiu. Para obter instruções detalhadas, consulte a seção [Using CloudHSM CLI](#) do Guia do usuário do AWS CloudHSM .

```
aws-cloudhsm > login --username kmsuser --role crypto-user
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "kmsuser",
    "role": "crypto-user"
  }
}
```

Como fazer logout e se conectar novamente

Siga o procedimento abaixo sempre que precisar fazer login como o usuário criptográfico `kmsuser` e reconectar seu repositório de chaves.

Observações

Os procedimentos a seguir usam a ferramenta de linha de comando do AWS CloudHSM Client SDK 5, a CLI do [CloudHSM](#). A CLI do CloudHSM substitui `key-handle` por `key-reference`.

Em 1º de janeiro de 2025, AWS CloudHSM encerrará o suporte às ferramentas de linha de comando do Client SDK 3, ao CloudHSM Management Utility (CMU) e ao Key Management Utility (KMU). Para obter mais informações sobre as diferenças entre as ferramentas de linha de comando do Client SDK 3 e a ferramenta de linha de comando do Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) no Guia do usuário do AWS CloudHSM .

1. Execute a tarefa e, em seguida, use o comando [logout](#) na CLI do CloudHSM para sair. Se você não se desconectar, as tentativas de reconectar seu armazenamento de AWS CloudHSM chaves falharão.

```
aws-cloudhsm logout
```

```
{
  "error_code": 0,
  "data": "Logout successful"
}
```

2. [Edite a configuração de senha kmsuser](#) para o armazenamento de chaves personalizado.

Isso informa AWS KMS a senha atual do cluster. `kmsuser`. Se você omitir essa etapa, não AWS KMS conseguirá fazer login no cluster como `kmsuser`, e todas as tentativas de reconectar seu armazenamento de chaves personalizadas falharão. Você pode usar o AWS KMS console ou o `KeyStorePassword` parâmetro da [UpdateCustomKeyStore](#) operação.

Por exemplo, esse comando informa AWS KMS que a senha atual é `tempPassword`. Substitua a senha de exemplo pela real.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --
key-store-password tempPassword
```

3. Reconecte o armazenamento de AWS KMS chaves ao AWS CloudHSM cluster. Substitua o exemplo de ID de armazenamento de AWS CloudHSM chaves por um válido. Durante o processo de conexão, AWS KMS altera a `kmsuser` senha para um valor que somente ela conhece.

A [ConnectCustomKeyStore](#) operação retorna rapidamente, mas o processo de conexão pode levar um longo período de tempo. A resposta inicial não indica o sucesso do processo de conexão.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

4. Use a [DescribeCustomKeyStores](#) operação para verificar se o armazenamento de AWS CloudHSM chaves está conectado. Substitua o exemplo de ID de armazenamento de AWS CloudHSM chaves por um válido.

Neste exemplo, o campo de estado da conexão mostra que o armazenamento de AWS CloudHSM chaves agora está conectado.

```
$ 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"
],
}
```

Armazenamentos de chaves externas

Os armazenamentos externos de chaves permitem que você proteja seus AWS recursos usando chaves criptográficas externas. AWS Esse recurso avançado foi criado para workloads regulamentadas que você precisa proteger com chaves de criptografia armazenadas em um sistema de gerenciamento de chaves externas que você controla. Os armazenamentos externos de chaves apoiam o [compromisso de soberania AWS digital](#) de dar a você controle soberano sobre seus dados AWS, incluindo a capacidade de criptografar com material chave que você possui e controla externamente. AWS

Um armazenamento de chaves externo é um [armazenamento de chaves personalizado](#) apoiado por um gerenciador de chaves externo que você possui e gerencia fora dele AWS. Seu gerenciador de chaves externo pode ser um módulo de segurança de hardware físico ou virtual (HSMs) ou qualquer sistema baseado em hardware ou software capaz de gerar e usar chaves criptográficas. As operações de criptografia e descriptografia que usam uma chave KMS em um armazenamento de chaves externo são realizadas pelo seu gerenciador de chaves externo usando seu material de chave criptográfica, um recurso conhecido como mantenha suas próprias chaves (HYOKs).

AWS KMS nunca interage diretamente com seu gerenciador de chaves externo e não pode criar, visualizar, gerenciar ou excluir suas chaves. Em vez disso, AWS KMS interage somente com o software [proxy externo de armazenamento de chaves](#) (proxy XKS) fornecido por você. Seu proxy externo de armazenamento de chaves media toda a comunicação entre AWS KMS e seu gerenciador de chaves externo. Ele transmite todas as solicitações AWS KMS para o seu gerenciador de chaves externo e transmite as respostas do seu gerenciador de chaves externo de volta para o AWS KMS. O proxy externo do armazenamento de chaves também traduz solicitações genéricas AWS KMS em um formato específico do fornecedor que seu gerente de chaves externo possa entender, permitindo que você use armazenamentos de chaves externos com gerenciadores de chaves de vários fornecedores.

Você pode usar chaves do KMS em um armazenamento de chaves externas para criptografia do lado do cliente, inclusive com o [AWS Encryption SDK](#). Mas os armazenamentos externos de chaves são um recurso importante para a criptografia do lado do servidor, permitindo que você proteja seus AWS recursos de forma múltipla Serviços da AWS com suas chaves criptográficas externas. AWS Serviços da AWS que oferecem suporte a [chaves gerenciadas pelo cliente](#) para criptografia simétrica também oferecem suporte a chaves KMS em um armazenamento de chaves externo. Para obter detalhes sobre o suporte para serviços, consulte [AWS Service Integration](#) (Integração de produtos da).

Os armazenamentos externos de chaves permitem que você use AWS KMS para cargas de trabalho regulamentadas, nas quais as chaves de criptografia devem ser armazenadas e usadas fora do AWS. Mas eles são um grande desvio do modelo de responsabilidade compartilhada padrão e exigem o aumento dos encargos operacionais. O maior risco de disponibilidade e latência excederá, para a maioria dos clientes, os benefícios de segurança percebidos pelos armazenamentos de chaves externas.

Os armazenamentos de chaves externas permitem controlar a raiz de confiança. Os dados criptografados sob chaves do KMS em seu armazenamento de chaves externas só podem ser descriptografados usando o gerenciador de chaves externas que você controla. Se você revogar temporariamente o acesso ao seu gerenciador de chaves externo, por exemplo, desconectando o armazenamento de chaves externo ou desconectando o gerenciador de chaves externo do proxy externo do armazenamento de chaves, AWS perderá todo o acesso às suas chaves criptográficas até que você as restaure. Durante esse intervalo, o texto cifrado criptografado sob suas chaves do KMS não poderá ser descriptografado. Se você revogar permanentemente o acesso ao seu gerenciador de chaves externas, todo o texto cifrado criptografado sob uma chave do KMS em seu armazenamento de chaves externas vai se tornar irrecuperável. As únicas exceções são AWS serviços que armazenam brevemente em cache as [chaves de dados](#) protegidas por suas chaves KMS. Essas chaves de dados continuarão funcionando até você desativar o recurso ou o cache expirar. Para obter detalhes, consulte [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#).

Os armazenamentos externos de chaves desbloqueiam os poucos casos de uso de cargas de trabalho regulamentadas em que as chaves de criptografia devem permanecer exclusivamente sob seu controle e inacessíveis. AWS Porém, esta é uma grande mudança na forma como você opera a infraestrutura baseada em nuvem e uma alteração significativa no modelo de responsabilidade compartilhada. Para a maioria das workloads, o aumento da carga operacional e os maiores riscos de disponibilidade e performance excederão os benefícios de segurança percebidos pelos armazenamentos de chaves externas.

Preciso de armazenamento de chaves externas?

Para a maioria dos usuários, o armazenamento de AWS KMS chaves padrão, protegido pelos [módulos de segurança de hardware validados pelo FIPS 140-3 Security Level 3](#), atende aos requisitos regulatórios, de controle e de segurança. Usuários do armazenamento de chaves externas incorrem em custos consideráveis, sobrecarga de manutenção e solução de problemas, além de riscos de latência, disponibilidade e confiabilidade.

Ao considerar um armazenamento de chaves externo, dedique algum tempo para entender as alternativas, incluindo um [armazenamento de AWS CloudHSM chaves](#) apoiado por um AWS CloudHSM cluster que você possui e gerencia, e chaves KMS com [material de chave importado](#) que você gera por conta própria HSMs e pode ser excluído das chaves KMS sob demanda. Especificamente, importar material de chave com um intervalo de expiração muito curto pode fornecer um nível similar de controle sem riscos à performance ou à disponibilidade.

Um armazenamento de chaves externas pode ser a solução certa para sua organização, caso cumpra os seguintes requisitos:

- Você precisa usar chaves criptográficas em seu gerenciador de chaves local ou em um gerenciador de chaves fora do seu AWS controle.
- É necessário demonstrar que suas chaves de criptografia são retidas somente sob seu controle fora da nuvem.
- É necessário criptografar e descriptografar usando chaves de criptografia com autorização independente.
- O material de chaves deve estar sujeito a um caminho de auditoria secundário e independente.

Se escolher um armazenamento de chaves externas, limite o uso a workloads que necessitam de proteção com chaves de criptografia fora da AWS.

Modelo de responsabilidade compartilhada

As chaves KMS padrão usam material de chaves gerado e usado para HSMs que seja AWS KMS proprietário e gerenciado. Você estabelece as políticas de controle de acesso em suas chaves KMS e configura Serviços da AWS que usem chaves KMS para proteger seus recursos. AWS KMS

assume a responsabilidade pela segurança, disponibilidade, latência e durabilidade do material de chaves em suas chaves KMS.

As chaves do KMS em armazenamentos de chaves externas dependem do material e das operações do gerenciador de chaves externas. Assim, o equilíbrio de responsabilidades desloca-se em direção a você. Você é responsável pela segurança, confiabilidade, durabilidade e desempenho das chaves criptográficas em seu gerenciador de chaves externo. AWS KMS é responsável por responder prontamente às solicitações e se comunicar com seu proxy externo de armazenamento de chaves e por manter nossos padrões de segurança. [Para garantir que cada chave externa armazene texto cifrado pelo menos tão forte quanto o texto AWS KMS cifrado padrão, AWS KMS primeiro criptografa todo o texto sem formatação com material de chave específico da sua AWS KMS chave KMS e, em seguida, o envia ao seu gerenciador de chaves externo para criptografia com sua chave externa, um procedimento conhecido como criptografia dupla.](#) Como resultado, nem o AWS KMS nem o proprietário do material de chave externa podem descriptografar somente o texto cifrado com criptografia dupla.

Você é responsável por manter um gerenciador de chaves externas que atenda aos seus padrões regulatórios e de performance, por fornecer e manter um proxy de armazenamento de chaves externas que esteja em conformidade com a [AWS KMS External Key Store Proxy API Specification](#) (Especificação da API do proxy de armazenamento de chaves externas do) e por garantir a disponibilidade e durabilidade do material de chave. Você também deve criar, configurar e manter um armazenamento de chaves externas. Quando surgem erros causados por componentes que você mantém, você deve estar preparado para identificar e resolver os erros para que os AWS serviços possam acessar seus recursos sem interrupções indevidas. AWS KMS fornece [orientação de solução](#) de problemas para ajudá-lo a determinar a causa dos problemas e as resoluções mais prováveis.

Analise as [CloudWatch métricas e dimensões da Amazon](#) que AWS KMS registram para lojas de chaves externas. AWS KMS recomenda fortemente que você crie CloudWatch alarmes para monitorar seu armazenamento externo de chaves para que você possa detectar os primeiros sinais de problemas operacionais e de desempenho antes que eles ocorram.

O que está mudando?

Armazenamentos de chaves externas são compatíveis apenas com chaves do KMS de criptografia simétrica. Dentro AWS KMS, você usa e gerencia as chaves KMS em um armazenamento de chaves externo da mesma forma que gerencia outras [chaves gerenciadas pelo cliente](#), incluindo a [definição de políticas de controle de acesso](#) e o [monitoramento do uso da chave](#). Você usa o mesmo APIs com os mesmos parâmetros para solicitar uma operação criptográfica com uma chave KMS em

um armazenamento de chaves externo que você usa para qualquer chave KMS. O preço também é o mesmo das chaves do KMS padrão. Para obter mais detalhes, consulte [Chaves do KMS em repositórios de chaves externos](#) e [Preços do AWS Key Management Service](#).

Porém, com os armazenamentos de chaves externas, os seguintes princípios mudam:

- Você é responsável pela disponibilidade, durabilidade e latência das operações de chaves.
- Você é responsável por todos os custos de desenvolvimento, compra, operação e licenciamento de seu sistema de gerenciamento de chaves externas.
- Você pode implementar a [autorização independente](#) de todas as solicitações AWS KMS de seu proxy externo de armazenamento de chaves.
- Você pode monitorar, auditar e registrar todas as operações do proxy externo do armazenamento de chaves e todas as operações do gerenciador de chaves externo relacionadas às AWS KMS solicitações.

Por onde começar?

Para criar e gerenciar um armazenamento de chaves externas, é necessário [escolher sua opção de conectividade de proxy de armazenamento de chaves externas](#), [organizar os pré-requisitos](#) e [criar e configurar o armazenamento de chaves externas](#).

Cotas

AWS KMS permite até [10 armazenamentos de chaves personalizadas](#) em cada Conta da AWS região, incluindo armazenamentos de [AWS CloudHSM chaves e armazenamentos de chaves externos](#), independentemente do estado da conexão. Além disso, há cotas de solicitação do AWS KMS para o [uso de chaves do KMS em um armazenamento de chaves externas](#).

Se você escolher a [conectividade de proxy VPC](#) para seu proxy externo de armazenamento de chaves, também poderá haver cotas nos componentes necessários VPCs, como sub-redes e balanceadores de carga de rede. Para obter mais informações sobre essas cotas, use o [console do Service Quotas](#).

Regiões

Para minimizar a latência da rede, crie seus componentes de armazenamento de chaves externas na Região da AWS mais próxima do [gerenciador de chaves externas](#). Se possível, escolha uma região com um tempo de resposta (RTT) de 35 milissegundos ou menos.

Os armazenamentos de chaves externos são suportados Regiões da AWS em todas as regiões AWS KMS com suporte, exceto na China (Pequim) e na China (Ningxia).

Recursos sem suporte

AWS KMS não oferece suporte aos seguintes recursos em armazenamentos de chaves personalizadas.

- [Chaves do KMS assimétricas](#)
- [Chaves do KMS de HMAC](#)
- [Chaves do KMS com material de chave importado](#)
- [Alternância automática de chaves](#)
- [Chaves de várias regiões](#)

Saiba mais:

- [Announcing AWS KMS External Key Store](#) (Anúncio do armazenamento de chaves externas do) no blog de notícias da AWS .

Conceitos de armazenamento de chaves externas

Aprenda os termos e conceitos básicos usados em repositórios de chaves externos.

Armazenamento de chaves externas

Um armazenamento de chaves externo é um [armazenamento de chaves AWS KMS personalizado](#) apoiado por um gerenciador de chaves externo fora do AWS que você possui e gerencia. Cada chave do KMS em um armazenamento de chaves externas está associada a uma [chave externa](#) de seu gerenciador de chaves externas. Quando você usa uma chave do KMS em um armazenamento de chaves externas para criptografia ou descriptografia, a operação é executada pelo gerenciador de chaves externas usando sua chave externa, um mecanismo conhecido como mantenha sua própria chave (HYOK). Esse recurso foi desenvolvido para organizações que precisam manter chaves de criptografia em seu próprio gerenciador de chaves externas.

Os armazenamentos externos de chaves garantem que as chaves criptográficas e as operações que protegem seus AWS recursos permaneçam sob seu controle no gerenciador de chaves externo. AWS KMS envia solicitações ao seu gerenciador de chaves externo para criptografar e

descriptografar dados, mas AWS KMS não pode criar, excluir ou gerenciar nenhuma chave externa. Todas as solicitações do AWS KMS seu gerenciador de chaves externo são mediadas por um componente de software [proxy de armazenamento de chaves externo](#) que você fornece, possui e gerencia.

AWS os serviços que oferecem suporte a [chaves gerenciadas pelo AWS KMS cliente](#) podem usar as chaves KMS em seu armazenamento de chaves externo para proteger seus dados. Como resultado, seus dados são, em última análise, protegidos por suas chaves usando suas operações de criptografia no gerenciador de chaves externas.

As chaves do KMS em um armazenamento de chaves externas têm modelos de confiança, [acordos de responsabilidade compartilhada](#) e expectativas de performance consideravelmente diferentes das chaves do KMS padrão. Com armazenamentos de chaves externas, você é responsável pela segurança e integridade do material de chave e pelas operações de criptografia. A disponibilidade e a latência das chaves do KMS em um armazenamento de chaves externas são afetadas pelo hardware, software, componentes de redes e pela distância entre o AWS KMS e o gerenciador de chaves externas. Também é provável que você incorra em custos adicionais para seu gerenciador de chaves externo e para a infraestrutura de rede e balanceamento de carga com a qual seu gerenciador de chaves externo se comunique. AWS KMS

Você pode usar seu armazenamento de chaves externas como parte de sua estratégia geral de proteção de dados. Para cada AWS recurso que você protege, você pode decidir qual requer uma chave KMS em um armazenamento de chaves externo e qual pode ser protegido por uma chave KMS padrão. Isso lhe dá a flexibilidade para escolher chaves do KMS para classificações de dados, aplicações ou projetos específicos.

Gerenciador de chaves externas

Um gerenciador de chaves externas é um componente fora da AWS que pode gerar chaves simétricas AES de 256 bits e realizar criptografia e descriptografia simétricas. O gerenciador de chaves externas para um armazenamento de chaves externas pode ser um módulo de segurança de hardware (HSM) físico, um HSM virtual ou um gerenciador de chaves de software com ou sem um componente HSM. Ele pode estar localizado em qualquer lugar externo AWS, inclusive em suas instalações, em um data center local ou remoto ou em qualquer nuvem. O armazenamento de chaves externas pode basear-se em um único gerenciador de chaves externas ou em várias instâncias de gerenciador de chaves relacionadas que compartilham chaves de criptografia, como um cluster do HSM. Os armazenamentos de chaves externas são criados para oferecer suporte a uma variedade de gerenciadores externos de diferentes fornecedores. Para obter detalhes

sobre como se conectar ao seu gerenciador de chaves externo, consulte [Escolher uma opção de conectividade de proxy de um repositório de chaves externo](#).

Chave externa

Cada chave do KMS em um armazenamento de chaves externas está associada a uma chave de criptografia do [gerenciador de chaves externas](#) conhecida como chave externa. Ao criptografar ou descriptografar com uma chave do KMS no armazenamento de chaves externas, a operação de criptografia é executada pelo [gerenciador de chaves externas](#) usando suas chaves externas.

Warning

A chave externa é essencial para a operação da chave do KMS. Se a chave externa for perdida ou excluída, o texto cifrado criptografado sob a chave do KMS associada vai se tornar irrecuperável.

Para armazenamentos de chaves externas, a chave externa deve ser uma chave AES de 256 bits que esteja habilitada e possa executar criptografia e descriptografia. Para obter requisitos detalhados de chaves externas, consulte [Requisitos para uma chave do KMS em um armazenamento de chaves externas](#).

AWS KMS não pode criar, excluir ou gerenciar nenhuma chave externa. O material de chave de criptografia nunca sai do gerenciador de chaves externas. Ao criar uma chave do KMS em um armazenamento de chaves externas, você fornece o ID de uma chave externa (XksKeyId). Você não pode alterar o ID da chave externa associada a uma chave do KMS, embora seu gerenciador de chaves externas possa alternar o material de chave associado ao ID da chave externa.

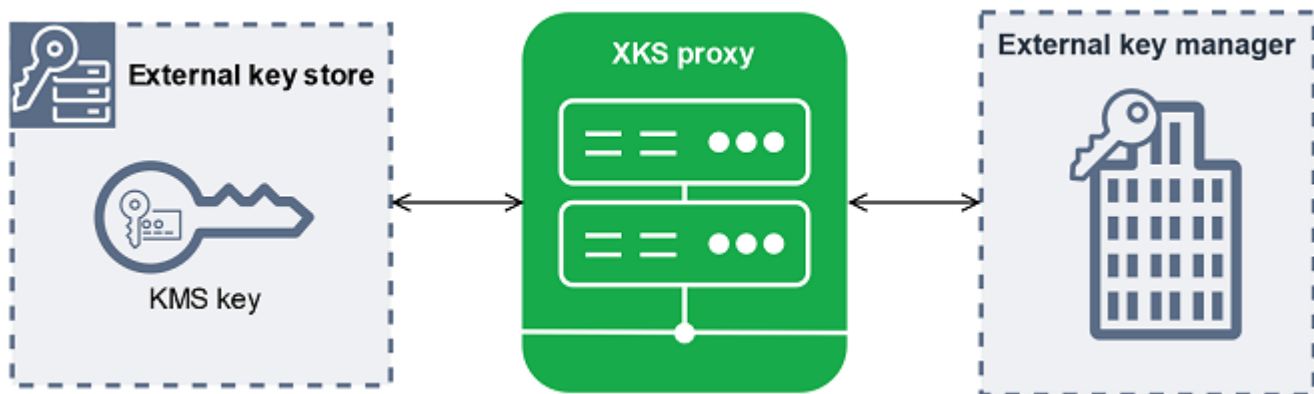
Além da chave externa, uma chave do KMS em um armazenamento de chaves externas também tem material de chave do AWS KMS. Os dados protegidos pela chave KMS são criptografados primeiro AWS KMS usando o material de AWS KMS chaves e, em seguida, pelo gerenciador de chaves externo usando sua chave externa. Esse processo de [criptografia dupla](#) garante que o texto cifrado protegido pela chave do KMS seja sempre pelo menos tão forte quanto o texto cifrado protegido somente pelo AWS KMS.

Muitas chaves de criptografia têm diferentes tipos de identificadores. Ao criar uma chave do KMS em um armazenamento de chaves externas, forneça o ID da chave externa que o [proxy de armazenamento de chaves externas](#) usa para se referir à chave externa. Se você usar o identificador

errado, haverá falha na tentativa de criar uma chave do KMS em seu armazenamento de chaves externas.

Proxy de armazenamento de chaves externas

O proxy externo de armazenamento de chaves (“proxy XKS”) é um aplicativo de software de propriedade e gerenciamento do cliente que medeia toda a comunicação entre AWS KMS e seu gerenciador de chaves externo. Ele também traduz AWS KMS solicitações genéricas em um formato que seu gerente de chaves externo específico do fornecedor entenda. É necessário um proxy de armazenamento de chaves externas para um armazenamento de chaves externas. Cada armazenamento de chaves externas é associado a um proxy de armazenamento de chaves externas.



AWS KMS não pode criar, excluir ou gerenciar nenhuma chave externa. O material de chave de criptografia nunca sai do gerenciador de chaves externas. Toda a comunicação entre AWS KMS e seu gerenciador de chaves externo é mediada pelo proxy externo do armazenamento de chaves. AWS KMS envia solicitações para o proxy externo do armazenamento de chaves e recebe respostas do proxy externo do armazenamento de chaves. O proxy externo do armazenamento de chaves é responsável por transmitir solicitações de AWS KMS seu gerenciador de chaves externo e transmitir respostas de seu gerenciador de chaves externo de volta para AWS KMS.

Você possui e gerencia o proxy de armazenamento de chaves externas para seu armazenamento de chaves externas e é responsável por sua manutenção e operação. Você pode desenvolver seu proxy de armazenamento de chaves externo com base na [especificação da API de proxy de armazenamento de chaves externo](#) de código aberto que AWS KMS publica ou compra um aplicativo proxy de um fornecedor. Seu proxy de armazenamento de chaves externas pode estar incluído em seu gerenciador de chaves externas. Para oferecer suporte ao desenvolvimento de proxy, AWS KMS também fornece um exemplo de proxy externo de armazenamento de chaves ([aws-kms-xks-proxy](#)).

e um cliente de teste ([xks-kms-xksproxy-test-client](#)) que verifica se o proxy do armazenamento de chaves externo está em conformidade com a especificação.

Para se autenticar AWS KMS, o proxy usa certificados TLS do lado do servidor. Para se autenticar em seu proxy, AWS KMS assina todas as solicitações em seu proxy externo de armazenamento de chaves com uma credencial de autenticação de [proxy](#) SigV4. Opcionalmente, seu proxy pode habilitar o TLS mútuo (mTLS) para garantir que ele só aceita solicitações de AWS KMS

O proxy externo de armazenamento de chaves deve oferecer suporte a HTTP/1.1 ou posterior e TLS 1.2 ou posterior com pelo menos um dos seguintes conjuntos de criptografia:

- TLS_AES_256_GCM_ (TLS 1.3SHA384)
- TLS_ _ CHACHA20 POLY1305 _ SHA256 (TLS 1.3)

 Note

O AWS GovCloud (US) Region não suporta TLS_ _ CHACHA20 _POLY1305. SHA256

- TLS_ECDHE_RSA_COM_AES_256_GCM_ (TLS 1.2SHA384)
- SHA384 TLS_ECDHE_ECDSA_COM_AES_256_GCM_ (TLS 1.2)

Para criar e usar as chaves do KMS em seu armazenamento de chaves externas, você deve primeiro [conectar o armazenamento de chaves externas](#) ao proxy de armazenamento de chaves externas. Você também pode desconectar o armazenamento de chaves externas do proxy sob demanda. Quando você faz isso, todas as chaves do KMS no armazenamento de chaves externas ficam [indisponíveis](#) e não podem ser usadas em nenhuma operação de criptografia.

Conectividade de proxy de armazenamento de chaves externas

A conectividade de proxy de armazenamento de chaves externo (“conectividade de proxy XKS”) descreve o método AWS KMS usado para se comunicar com seu proxy de armazenamento de chaves externo.

Você especifica sua opção de conectividade de proxy ao criar o armazenamento de chaves externas, e ela se torna uma propriedade do armazenamento de chaves externas. Você pode alterar sua opção de conectividade de proxy atualizando a propriedade do armazenamento de chaves personalizado, mas você deve ter certeza de que o proxy de armazenamento de chaves externas ainda pode acessar as mesmas chaves externas.

AWS KMS suporta as seguintes opções de conectividade.

- [Conectividade de endpoint público](#) — AWS KMS envia solicitações para seu proxy externo de armazenamento de chaves pela Internet para um endpoint público que você controla. Essa opção é simples de criar e manter, mas talvez não atenda aos requisitos de segurança de todas as instalações.
- [Conectividade do serviço de endpoint VPC](#) — AWS KMS envia solicitações para um serviço de endpoint da Amazon Virtual Private Cloud (Amazon VPC) que você cria e mantém. Você pode hospedar seu proxy externo de armazenamento de chaves dentro de sua Amazon VPC ou hospedar seu proxy externo de armazenamento de chaves fora AWS e usar a Amazon VPC somente para comunicação. Você também pode conectar o repositório de chaves externo a um serviço de endpoint da Amazon VPC pertencente a outra Conta da AWS.

Para obter detalhes sobre as opções de conectividade de proxy de armazenamento de chaves externas, consulte [Escolher uma opção de conectividade de proxy de um repositório de chaves externo](#).

Credencial de autenticação de proxy de armazenamento de chaves externas

Para se autenticar no proxy externo do armazenamento de chaves, AWS KMS assine todas as solicitações no proxy externo do armazenamento de chaves com uma credencial de autenticação [Signature V4 \(SigV4\)](#). Você estabelece e mantém a credencial de autenticação em seu proxy e, em seguida, fornece essa credencial AWS KMS ao criar seu armazenamento externo.

Note

A credencial SigV4 AWS KMS usada para assinar solicitações ao proxy XKS não está relacionada a nenhuma credencial SigV4 associada aos diretores em seu. AWS Identity and Access Management Contas da AWS Não reutilize nenhuma credencial SigV4 do IAM para seu proxy de armazenamento de chaves externas.

Toda credencial de autenticação do proxy tem duas partes. É necessário fornecer as duas partes ao criar um armazenamento de chaves externas ou atualizar a credencial de autenticação do armazenamento de chaves externas.

- ID da chave de acesso: identifica a chave de acesso secreta. Você pode fornecer esse ID em texto não criptografado.

- Chave de acesso secreta: a parte secreta da credencial. AWS KMS criptografa a chave de acesso secreta na credencial antes de armazená-la.

Você pode [editar a configuração da credencial](#) a qualquer momento, como ao inserir valores incorretos, ao alterar a credencial no proxy ou quando seu proxy altera a credencial. Para obter detalhes técnicos sobre a AWS KMS autenticação no proxy externo do armazenamento de chaves, consulte [Autenticação](#) na Especificação da API AWS KMS External Key Store Proxy.

Para permitir que você altere sua credencial sem interromper o Serviços da AWS uso de chaves KMS em seu armazenamento de chaves externo, recomendamos que o proxy do armazenamento de chaves externo suporte pelo menos duas credenciais de autenticação válidas para AWS KMS. Isso garante que sua credencial anterior continue funcionando enquanto você fornece sua nova credencial para o AWS KMS.

Para ajudá-lo a rastrear a idade da sua credencial de autenticação de proxy, AWS KMS define uma CloudWatch métrica da Amazon, [XksProxyCredentialAge](#). Você pode usar essa métrica para criar um CloudWatch alarme que o notifique quando a idade da sua credencial atingir um limite estabelecido por você.

Para fornecer uma garantia adicional de que seu proxy de armazenamento de chaves externas responde somente ao AWS KMS, alguns proxies de chaves externas oferecem suporte à Transport Layer Security mútua (mTLS). Para obter detalhes, consulte [Autenticação mTLS \(opcional\)](#).

Proxy APIs

Para oferecer suporte a um armazenamento de chaves AWS KMS externo, um [proxy externo de armazenamento de chaves](#) deve implementar o proxy necessário APIs conforme descrito na [Especificação da API AWS KMS External Key Store Proxy](#). Essas solicitações de API de proxy são as únicas solicitações AWS KMS enviadas ao proxy. Embora você nunca envie essas solicitações diretamente, conhecê-las pode ajudar a corrigir problemas que possam surgir no armazenamento de chaves externas ou no proxy. Por exemplo, AWS KMS inclui informações sobre a latência e as taxas de sucesso dessas chamadas de API em suas [CloudWatch métricas da Amazon](#) para lojas de chaves externas. Para obter detalhes, consulte [Monitorar repositórios de chaves externos](#).

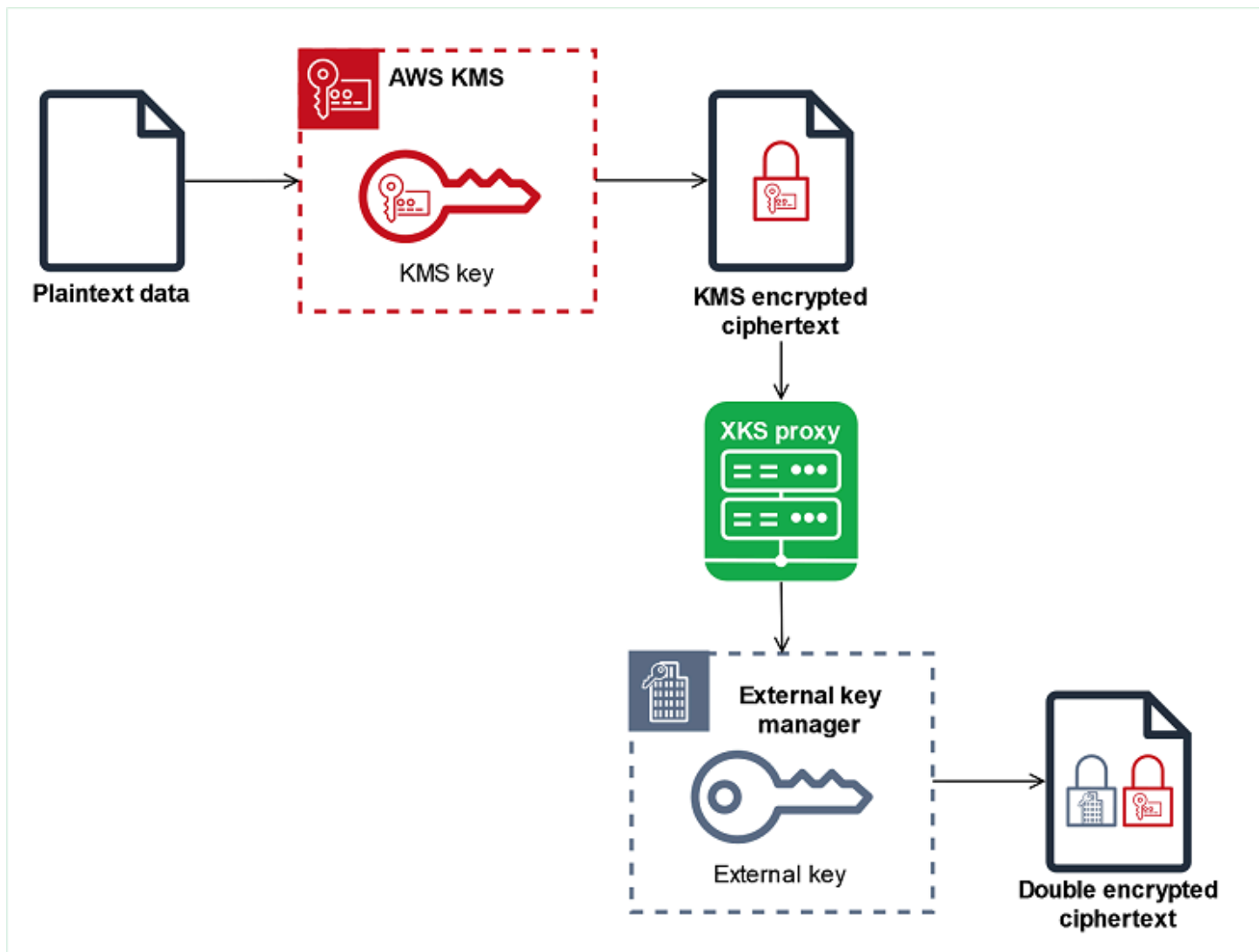
A tabela a seguir lista e descreve cada proxy APIs. Também inclui as AWS KMS operações que acionam uma chamada para a API proxy e quaisquer exceções de AWS KMS operação relacionadas à API proxy.

API de proxy	Description	AWS KMS Operações relacionadas
Decrypt	AWS KMS envia o texto cifrado a ser descriptografado e o ID da chave externa a ser usada . O algoritmo de criptografia obrigatório é AES_GCM.	Descriptografar , ReEncrypt
Encrypt	AWS KMS envia dados para serem criptografados e o ID da chave externa a ser usada. O algoritmo de criptografia obrigatório é AES_GCM.	Criptografar , GenerateDataKey , GenerateDataKeyWithoutPlainText , ReEncrypt
GetHealthStatus	AWS KMS solicita informações sobre o status do proxy e do seu gerenciador de chaves externo. Cada gerenciador de chaves externas pode ter um dos status a seguir. • Active: íntegro; pode atender ao tráfego • Degraded: não íntegro, mas pode atender ao tráfego • Unavailable : não íntegro; não pode atender ao tráfego	CreateCustomKeyStore(para conectividade de endpoint público), ConnectCustomKeyStore(para conectividade de serviço de endpoint VPC) Se todas as instâncias do gerenciador de chaves externas tiverem o status Unavailable , as tentativas de criar ou conectar o armazenamento de chaves falharão com XksProxyUriUnreachableException .
GetKeyMetadata	AWS KMS solicita informações sobre a chave externa associada a uma chave KMS em seu armazenamento de chaves externo. A resposta inclui a especificação da chave (AES_256), o uso da chave ([ENCRYPT, DECRYPT]) e se a chave externa está ENABLED ou DISABLED.	CreateKey Se a especificação da chave não for AES_256, se o uso da chave não for [ENCRYPT, DECRYPT] ou se o status for DISABLED, a operação CreateKey falhará com XksKeyInvalidConfigurationException .

Criptografia dupla

Os dados criptografados por uma chave do KMS em um armazenamento de chaves externas são criptografados duas vezes. Primeiro, AWS KMS criptografa os dados com material de AWS KMS chave específico da chave KMS. Em seguida, o texto cifrado criptografado pelo AWS KMS é criptografado pelo [gerenciador de chaves externas](#) usando sua [chave externa](#). O processo é conhecido como criptografia dupla.

A criptografia dupla garante que os dados criptografados por uma chave do KMS em um armazenamento de chaves externas sejam pelo menos tão fortes quanto o texto cifrado criptografado por uma chave do KMS padrão. Ele também protege o texto simples em trânsito do proxy externo do AWS KMS armazenamento de chaves. Com a criptografia dupla, você mantém o controle total de seus textos cifrados. Se você revogar permanentemente o acesso da AWS à sua chave externa por meio do proxy externo, todo texto cifrado da AWS restante será efetivamente destruído por criptografia.



Para habilitar a criptografia dupla, cada chave do KMS em um armazenamento de chaves externas tem duas chaves de reserva de criptografia:

- Um material de AWS KMS chave exclusivo da chave KMS. Esse material de chave é gerado e usado somente em módulos de segurança de hardware com certificação AWS KMS [FIPS 140-3 Security Level 3](#) (). HSMs
- Uma [chave externa](#) em seu gerenciador de chaves externas.

A criptografia dupla tem os seguintes efeitos:

- AWS KMS não pode descriptografar nenhum texto cifrado criptografado por uma chave KMS em um armazenamento de chaves externo sem acessar suas chaves externas por meio do proxy externo do armazenamento de chaves.

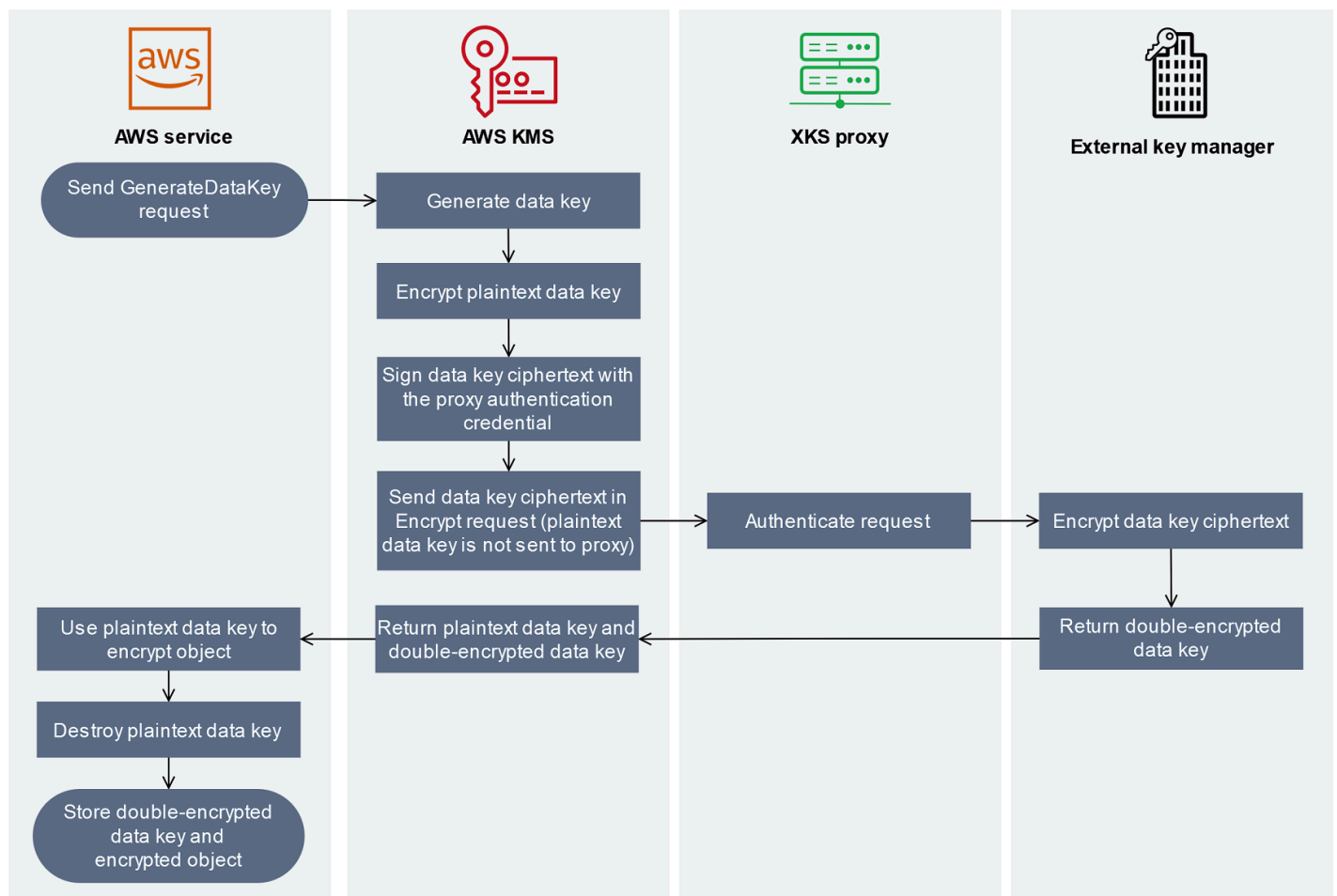
- Você não pode descriptografar nenhum texto cifrado criptografado por uma chave KMS em um armazenamento externo de chaves externo AWS, mesmo que você tenha o material de chave externo.
- Você não pode recriar uma chave do KMS que foi excluída de um armazenamento de chaves externas, mesmo que você tenha o material de chave externa. Cada chave do KMS tem metadados exclusivos que ela inclui no texto cifrado simétrico. Uma nova chave do KMS não seria capaz de descriptografar o texto cifrado criptografado pela chave original, ainda que usasse o mesmo material de chave externa.

Para obter um exemplo de criptografia dupla na prática, consulte [Como funcionam os armazenamentos de chaves externas](#).

Como funcionam os armazenamentos de chaves externas

O [repositório de chaves externo](#), o [proxy do repositório de chaves externo](#) e o [gerenciador de chaves externo](#) trabalham juntos para proteger seus recursos da AWS. O procedimento a seguir descreve o fluxo de trabalho de criptografia de um AWS service (Serviço da AWS) comum que criptografa cada objeto em uma chave de dados exclusiva protegida por uma chave do KMS. Nesse caso, você escolheu uma chave do KMS em um armazenamento de chaves externas para proteger o objeto. O exemplo mostra como AWS KMS usa [criptografia dupla](#) para proteger a chave de dados em trânsito e garantir que o texto cifrado gerado por uma chave KMS em um armazenamento de chaves externo seja sempre pelo menos tão forte quanto o texto cifrado criptografado por uma chave KMS simétrica padrão com material de chave inserido. AWS KMS

Os métodos de criptografia usados por cada unidade real AWS service (Serviço da AWS) que se integra AWS KMS variam. Para obter detalhes, consulte o tópico “Proteção dos dados” no capítulo Segurança da documentação do AWS service (Serviço da AWS).



1. Você adiciona um novo objeto ao seu AWS service (Serviço da AWS) recurso. Para criptografar o objeto, AWS service (Serviço da AWS) ele envia uma [GenerateDataKey](#) solicitação para AWS KMS usar uma chave KMS em seu armazenamento de chaves externo.
2. AWS KMS gera uma chave de [dados simétrica de 256 bits e se prepara para enviar uma cópia da chave](#) de dados em texto simples para o gerenciador de chaves externo por meio do proxy externo do armazenamento de chaves. AWS KMS inicia o processo de [criptografia dupla](#) criptografando a chave de dados em texto simples com o [material de AWS KMS chave](#) associado à chave KMS no armazenamento de chaves externo.
3. AWS KMS envia uma solicitação [de criptografia](#) para o proxy externo do armazenamento de chaves associado ao armazenamento de chaves externo. A solicitação inclui o texto cifrado da chave de dados a ser criptografado e a ID da [chave externa](#) associada à chave KMS. AWS KMS assina a solicitação usando a [credencial de autenticação de proxy](#) para seu proxy externo de armazenamento de chaves.

A cópia em texto simples da chave de dados não é enviada para o proxy de armazenamento de chaves externas.

4. O proxy de armazenamento de chaves externas autentica a solicitação e passa a solicitação de criptografia para o gerenciador de chaves externas.

Alguns proxies de armazenamento de chaves externas também implementam uma [política de autorização](#) opcional que permite que somente entidades principais selecionadas realizem operações sob condições específicas.

5. O gerenciador de chaves externas criptografa o texto cifrado da chave de dados usando a chave externa especificada. O gerenciador de chaves externas retorna a chave de dados com criptografia dupla para o proxy de armazenamento de chaves externas, que a retorna para o AWS KMS.
6. AWS KMS retorna a chave de dados em texto simples e a cópia criptografada dupla dessa chave de dados para o AWS service (Serviço da AWS)
7. O AWS service (Serviço da AWS) usa a chave de dados de texto simples para criptografar o objeto de recurso, destrói a chave de dados de texto sem formatação e armazena a chave de dados criptografada com o objeto criptografado.

Alguns Serviços da AWS podem armazenar em cache a chave de dados de texto simples para usar em vários objetos ou reutilizá-la enquanto o recurso estiver em uso. Para obter detalhes, consulte [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#).

Para descriptografar o objeto criptografado, é AWS service (Serviço da AWS) necessário enviar a chave de dados criptografada de volta AWS KMS em uma solicitação de descriptografia.

Para descriptografar a chave de dados criptografada, AWS KMS deve enviar a chave de dados criptografada de volta ao proxy externo do armazenamento de chaves com o ID da chave externa. Se a solicitação de descriptografia para o proxy externo do armazenamento de chaves falhar por algum motivo, não será AWS KMS possível descriptografar a chave de dados criptografada e não será possível descriptografar o AWS service (Serviço da AWS) objeto criptografado.

Controlar o acesso ao repositório de chaves externo

Todos os recursos de controle de AWS KMS acesso — [políticas principais](#), [políticas de IAM](#) e [concessões](#) — que você usa com chaves KMS padrão funcionam da mesma forma para chaves KMS em um armazenamento de chaves externo. Você pode usar políticas do IAM para controlar o acesso às operações de API que criam e gerenciam armazenamentos de chaves externas. Você

usa políticas do IAM e políticas de chaves para controlar o acesso ao AWS KMS keys em seu armazenamento de chaves externo. Você também pode usar [políticas de controle de serviço](#) para sua AWS organização e [políticas de VPC endpoint](#) para controlar o acesso às chaves KMS em seu armazenamento de chaves externo.

Recomendamos que você forneça aos usuários e perfis apenas as permissões necessárias para as tarefas que possivelmente executarão.

Tópicos

- [Autorizar gerenciadores de armazenamento de chaves externas](#)
- [Autorizar usuários de chaves do KMS em armazenamentos de chaves externas](#)
- [Autorizando AWS KMS a comunicação com seu proxy externo de armazenamento de chaves](#)
- [Autorização de proxy de armazenamento de chaves externas \(opcional\)](#)
- [Autenticação mTLS \(opcional\)](#)

Autorizar gerenciadores de armazenamento de chaves externas

As entidades principais que criam e gerenciam um armazenamento de chaves externas precisam de permissões para as operações de armazenamento de chaves personalizado. A lista a seguir descreve as permissões mínimas necessárias para os gerenciadores de armazenamento de chaves externas. Como um armazenamento de chaves personalizado não é um AWS recurso, você não pode fornecer permissão a um armazenamento de chaves externo para responsáveis em outros Contas da AWS.

- `kms:CreateCustomKeyStore`
- `kms:DescribeCustomKeyStores`
- `kms:ConnectCustomKeyStore`
- `kms:DisconnectCustomKeyStore`
- `kms:UpdateCustomKeyStore`
- `kms>DeleteCustomKeyStore`

Para criar um armazenamento de chaves externo com a [conectividade do serviço de endpoint Amazon VPC e o serviço](#) de endpoint VPC pertencer a outro Conta da AWS, você também precisará da seguinte permissão:

- `ec2:DescribeVPCEndpointServices`

As entidades principais que criam um armazenamento de chaves externas precisam de permissão para criar e configurar os componentes do armazenamento de chaves externas. Elas podem criar armazenamentos de chaves externas somente em suas próprias contas. Para criar um armazenamento de chaves externas com [conectividade de serviço de endpoint da VPC](#), as entidades principais devem ter permissão para criar os seguintes componentes:

- Uma Amazon VPC
- Sub-redes públicas e privadas
- Um balanceador de carga de rede e grupo de destino
- Um serviço de endpoint da Amazon VPC

Para obter detalhes, consulte [Identity and Access Management para a Amazon VPC](#), [Gerenciamento de identidade e acesso para endpoints da VPC e serviços de endpoint da VPC](#) e [Elastic Load Balancing API permissions](#) (Permissões de API do Elastic Load Balancing).

Autorizar usuários de chaves do KMS em armazenamentos de chaves externas

Os diretores que criam e gerenciam AWS KMS keys em seu armazenamento de chaves externo exigem [as mesmas permissões](#) que aqueles que criam e gerenciam qualquer chave KMS em. AWS KMS A [política de chaves padrão](#) para a chave do KMS em um armazenamento de chaves externas é idêntica à política de chaves padrão para chaves do KMS no AWS KMS. O [controle de acesso por atributo](#) (ABAC), que usa etiquetas e aliases para controlar o acesso a chaves do KMS, também é eficaz em chaves do KMS em armazenamentos de chaves externas.

As entidades principais que usam as chaves do KMS no seu armazenamento de chaves personalizado para [operações de criptografia](#) precisam de permissão para executar a operação criptográfica com a chaves do KMS, como [kms:Decrypt](#). Você pode fornecer essas permissões em uma política do IAM ou em uma política de chaves. No entanto, elas não precisam de permissões adicionais para usar uma chave do KMS em um armazenamento de chaves personalizado.

Para definir uma permissão que se aplique somente às chaves do KMS em um armazenamento de chaves externas, use a condição de política [kms:KeyOrigin](#) com um valor de `EXTERNAL_KEY_STORE`. Você pode usar essa condição para limitar a `CreateKey` permissão [kms:](#) ou qualquer permissão específica de um recurso de chave KMS. Por exemplo, a política do IAM a seguir permite que a identidade à qual está anexada chame as operações especificadas em todas as chaves do KMS da conta, desde que as chaves do KMS estejam em um armazenamento de chaves externas. Observe que você pode limitar a permissão às chaves KMS em um armazenamento de

chaves externo e às chaves KMS em um Conta da AWS, mas não a qualquer armazenamento de chaves externo específico na conta.

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

Autorizando AWS KMS a comunicação com seu proxy externo de armazenamento de chaves

AWS KMS se comunica com seu gerenciador de chaves externo somente por meio do [proxy externo de armazenamento de chaves](#) que você fornece. AWS KMS autentica seu proxy assinando suas solicitações usando o [processo Signature Version 4 \(SigV4\)](#) com a [credencial de autenticação de proxy do armazenamento de chaves externo](#) que você especificar. Se você estiver usando [conectividade de endpoint público](#) para seu proxy externo de armazenamento de chaves, AWS KMS não precisará de nenhuma permissão adicional.

No entanto, se você estiver usando a [conectividade do serviço de endpoint VPC](#), deverá dar AWS KMS permissão para criar um endpoint de interface para seu serviço de endpoint Amazon VPC. Essa permissão é necessária independentemente de o proxy externo do armazenamento de chaves estar na sua VPC ou se o proxy externo do armazenamento de chaves está localizado em outro lugar, mas usa o serviço de endpoint da VPC para se comunicar. AWS KMS

Para permitir AWS KMS a criação de um endpoint de interface, use o console [Amazon VPC](#) ou [ModifyVpcEndpointServicePermissions](#) a operação. Conceda permissões para a seguinte entidade principal: `cks.kms.<region>.amazonaws.com`.

Se o seu serviço de endpoint da Amazon VPC for de propriedade de Conta da AWS outra pessoa que não seja a Conta da AWS proprietária do armazenamento de chaves externo (XKS), você também precisará permitir que o XKS acesse o serviço de endpoint da VPC. Para fazer isso, [liste o Conta da AWS ID XKS como principal para o serviço de endpoint](#) da Amazon VPC.

Same Conta da AWS

Quando seu serviço de VPC endpoint pertence ao mesmo Conta da AWS que seu armazenamento de chaves externo, você deve adicioná-lo AWS KMS à lista Permitir diretores do seu serviço de VPC endpoint.

O exemplo a seguir usa o [modify-vpc-endpoint-service-permissions](#) AWS CLI comando AWS KMS para permitir a conexão com o serviço de endpoint de VPC especificado na região Oeste dos EUA (Oregon) (us-west-2). Antes de usar esse comando, substitua o ID do serviço Amazon VPC por valores válidos para sua configuração. Região da AWS

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com"]'
```

Para remover essa permissão, use o [console da Amazon VPC](#) ou o [ModifyVpcEndpointServicePermissions](#) com o RemoveAllowedPrincipals parâmetro.

Cross Conta da AWS

Quando seu serviço de VPC endpoint pertence um ao outro Conta da AWS, você deve adicionar ambos AWS KMS e seu armazenamento de chaves externo à lista Permitir principais do seu serviço de VPC endpoint.

O exemplo a seguir usa o [modify-vpc-endpoint-service-permissions](#) AWS CLI comando para permitir que ambos AWS KMS e seu armazenamento de chaves externo (XKS) se conectem ao serviço de VPC endpoint especificado na região Oeste dos EUA (Oregon) (us-west-2). Antes de usar esse comando, substitua o ID do serviço Amazon VPC, a Região da AWS e o ARN da entidade principal do IAM por valores válidos para sua configuração. O diretor do IAM deve ser substituído por um principal do proprietário Conta da AWS do XKS.

Neste exemplo, `arn:aws:iam::123456789012:role/cks_role` é a entidade principal do IAM na conta do proprietário do XKS, que será usada para criar, atualizar ou conectar o XKS ao serviço de endpoint da VPC. Se você quiser permitir que todas as entidades principais na

conta de proprietário do XKS acessem seu serviço de endpoint da VPC, você poderá especificar `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"]'
```

Para remover essa permissão, use o [console da Amazon VPC](#) ou o [ModifyVpcEndpointServicePermissions](#) com o `RemoveAllowedPrincipals` parâmetro.

Autorização de proxy de armazenamento de chaves externas (opcional)

Alguns proxies de armazenamento de chaves externas implementam requisitos de autorização para o uso de suas chaves externas. O proxy de armazenamento de chaves externas é permitido, mas não obrigatório, para criar e implementar um esquema de autorização que permita que usuários específicos solicitem operações específicas somente sob certas condições. Por exemplo, o proxy pode ser configurado para dar ao usuário A permissão para criptografar com uma chave externa específica, mas não para descriptografar com ela.

A autorização de proxy é independente da [autenticação de proxy baseada em SIGV4](#) que é AWS KMS necessária para todos os proxies de armazenamento de chaves externo. Também é independente das políticas de chave, políticas do IAM e concessões que autorizam o acesso a operações que afetam o armazenamento de chaves externas ou suas chaves do KMS.

Para habilitar a autorização pelo proxy externo do armazenamento de chaves, AWS KMS inclua metadados em cada [solicitação da API do proxy](#), incluindo o chamador, a chave KMS, a AWS KMS operação, a AWS service (Serviço da AWS) (se houver). Veja a seguir os metadados da solicitação para a versão 1 (v1) da API de proxy de chave externa.

```
"requestMetadata": {
  "awsPrincipalArn": string,
  "awsSourceVpc": string, // optional
  "awsSourceVpce": string, // optional
  "kmsKeyArn": string,
  "kmsOperation": string,
  "kmsRequestId": string,
  "kmsViaService": string // optional
}
```

Por exemplo, você pode configurar seu proxy para permitir solicitações de um determinado principal (`awsPrincipalArn`), mas somente quando a solicitação for feita em nome do principal por um determinado AWS service (Serviço da AWS) (`kmsViaService`).

Se a autorização do proxy falhar, a AWS KMS operação relacionada falhará com uma mensagem que explica o erro. Para obter mais detalhes, consulte [Problemas de autorização de proxy](#).

Autenticação mTLS (opcional)

Para permitir que seu proxy de armazenamento de chaves externo autentique solicitações AWS KMS, AWS KMS assine todas as solicitações em seu proxy de armazenamento de chaves externo com a [credencial de autenticação](#) de proxy Signature V4 (SigV4) para seu armazenamento de chaves externo.

Para fornecer garantia adicional de que seu proxy externo de armazenamento de chaves responda somente às AWS KMS solicitações, alguns proxies de chave externa oferecem suporte ao Transport Layer Security (mTLS) mútuo, no qual ambas as partes de uma transação usam certificados para se autenticarem mutuamente. O mTLS adiciona a autenticação do lado do cliente — em que o servidor proxy externo do armazenamento de chaves autentica o AWS KMS cliente — à autenticação do lado do servidor fornecida pelo TLS padrão. Caso a credencial de autenticação de proxy esteja comprometida, o que raramente acontece, o mTLS impede que terceiros façam solicitações de API bem-sucedidas ao proxy do armazenamento de chaves externas.

Para implementar o mTLS, configure seu proxy de armazenamento de chaves externas para aceitar somente certificados TLS do lado do cliente com as seguintes propriedades:

- O nome comum da entidade no certificado TLS deve ser `cks.kms.<Region>.amazonaws.com`, por exemplo, `cks.kms.eu-west-3.amazonaws.com`.
- O certificado deve estar vinculado a uma autoridade de certificação associada ao [Amazon Trust Services](#).

Escolher uma opção de conectividade de proxy de um repositório de chaves externo

Antes de criar seu armazenamento de chaves externo, escolha a opção de conectividade que determina como AWS KMS se comunica com os componentes do armazenamento de chaves externo. A opção de conectividade escolhida determinará o restante do processo de planejamento.

Se você estiver criando um armazenamento de chaves externo, precisará determinar como AWS KMS se comunica com seu [proxy de armazenamento de chaves externo](#). Essa escolha determinará quais componentes você precisa e como você os configura. AWS KMS suporta as seguintes opções de conectividade.

- [Conectividade de endpoints públicos](#)
- [Conectividade do serviço de endpoint da VPC](#)

Escolha a opção que atenda aos seus objetivos de performance e segurança.

Antes de começar, [confirme se você precisa de um armazenamento de chaves externas](#). A maioria dos clientes pode usar chaves KMS apoiadas por material de AWS KMS chaves.

Considerações

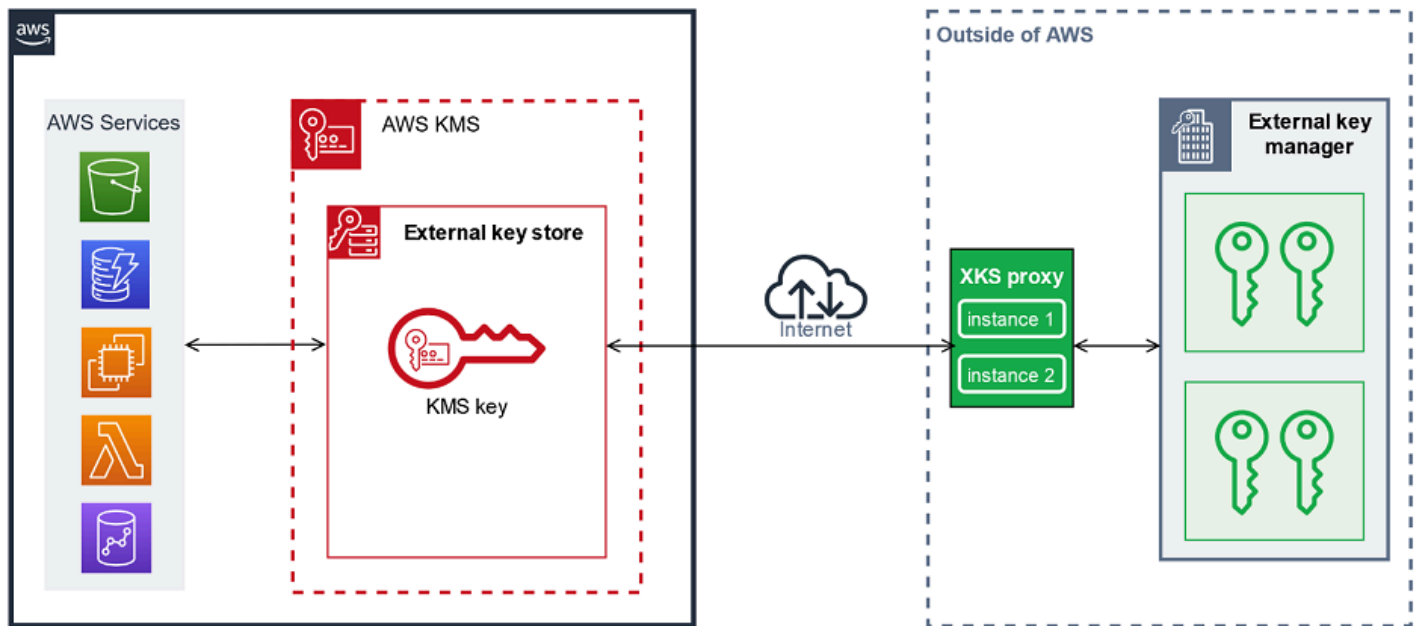
- Se seu proxy de armazenamento de chaves externas estiver incorporado ao gerenciador de chaves externas, a conectividade poderá ser predeterminada. Para obter orientações, consulte a documentação do gerenciador de chaves externas ou do proxy de armazenamento de chaves externas.
- Você pode [alterar a opção de conectividade de proxy de armazenamento de chaves externas](#) mesmo em um armazenamento de chaves externas operacional. Contudo, o processo deve ser planejado e executado com atenção para minimizar interrupções, evitar erros e garantir acesso contínuo às chaves de criptografia que criptografam seus dados.

Conectividade de endpoints públicos

AWS KMS se conecta ao proxy externo do armazenamento de chaves (proxy XKS) pela Internet usando um endpoint público.

Essa opção de conectividade é mais fácil de configurar e de manter e se alinha bem com alguns modelos de gerenciamento de chaves. No entanto, pode não atender aos requisitos de segurança de algumas organizações.

XKS proxy connected by a public endpoint



Requisitos

Se você escolher a conectividade de endpoint público, será necessário realizar as ações a seguir.

- Seu proxy de armazenamento de chaves externas deve estar acessível em um endpoint roteável publicamente.
- Você pode usar o mesmo endpoint público para vários armazenamentos de chaves externas, desde que eles usem valores de [caminho do URI do proxy](#) diferentes.
- Você não pode usar o mesmo endpoint para um armazenamento de chaves externo com conectividade de endpoint público e qualquer armazenamento de chaves externo com conectividade de serviços de endpoint VPC no mesmo Região da AWS, mesmo que os armazenamentos de chaves estejam em locais diferentes. Contas da AWS
- É necessário obter um certificado TLS emitido por uma autoridade de certificação pública com suporte para armazenamentos de chaves externas. Para obter uma lista, consulte [Trusted Certificate Authorities](#) (Autoridades de certificação confiáveis).

O nome comum (CN) da entidade no certificado TLS deve corresponder ao nome do domínio no [endpoint do URI do proxy](#) do armazenamento de chaves externas. Por exemplo, se o endpoint público for `https://myproxy.xks.example.com`, o TLS, o CN no certificado TLS deverá ser `myproxy.xks.example.com` ou `*.xks.example.com`.

- Certifique-se de que qualquer firewall entre AWS KMS e o proxy externo do armazenamento de chaves permita o tráfego de e para a porta 443 no proxy. AWS KMS se comunica na porta 443. IPv4 Esse valor não é configurável.

Para todos os requisitos de um armazenamento de chaves externas, consulte [Organizar os pré-requisitos](#).

Conectividade do serviço de endpoint da VPC

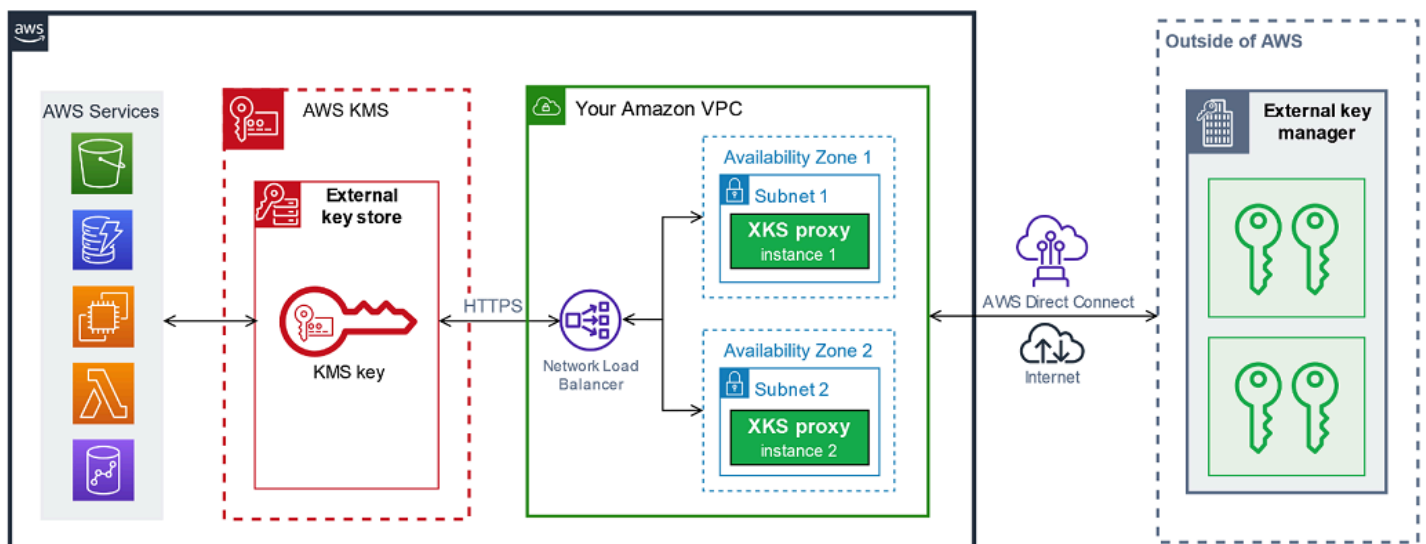
AWS KMS se conecta ao proxy externo do armazenamento de chaves (proxy XKS) criando um endpoint de interface para um serviço de endpoint da Amazon VPC que você cria e configura. Você é responsável por [criar o serviço de endpoint da VPC](#) e por conectar sua VPC ao gerenciador de chaves externas.

Seu serviço de endpoint pode usar qualquer uma das opções de [network-to-AWS VPC compatíveis](#) para comunicações, inclusive. [AWS Direct Connect](#)

Essa opção de conectividade é mais complicada de configurar e manter. Mas ele usa AWS PrivateLink, o que permite AWS KMS conectar-se de forma privada à sua Amazon VPC e ao seu proxy externo de armazenamento de chaves sem usar a Internet pública.

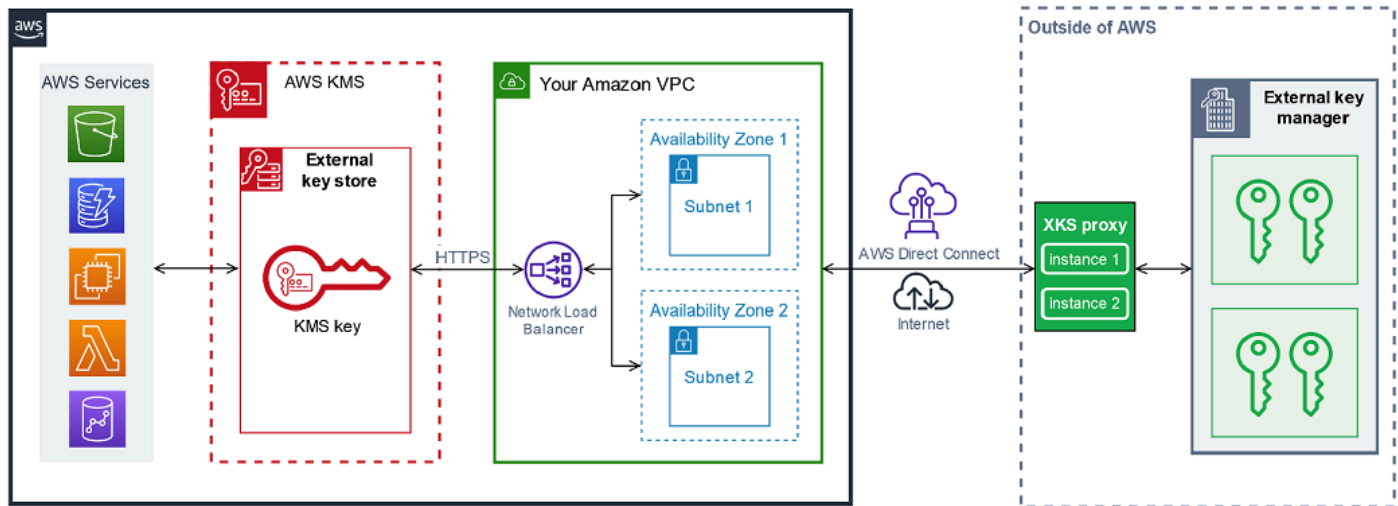
Você pode localizar o proxy de armazenamento de chaves externas na Amazon VPC.

XKS proxy hosted in Amazon VPC



Como alternativa, você pode localizar seu proxy externo de armazenamento de chaves fora Nuvem AWS e usar seu serviço de endpoint Amazon VPC somente para comunicação segura com. AWS KMS

XKS proxy connected via Amazon VPC endpoint service



Você também pode conectar um armazenamento de chaves externo a um serviço de endpoint da Amazon VPC de propriedade de outra pessoa. Conta da AWS Ambos Contas da AWS precisam das [permissões necessárias](#) para permitir a comunicação entre AWS KMS e o serviço de endpoint da VPC

Saiba mais:

- Analise o processo de criação de um armazenamento de chaves externas, incluindo a [montagem dos pré-requisitos](#). Isso ajudará a garantir que você tenha todos os componentes obrigatórios para criar seu armazenamento de chaves externas.
- Saiba como [controlar o acesso ao armazenamento de chaves externas](#), inclusive as permissões que os administradores e usuários do armazenamento de chaves externas exigem.
- Saiba mais sobre as [CloudWatch métricas e dimensões da Amazon](#) que AWS KMS registram para lojas de chaves externas. É altamente recomendável criar alarmes para monitorar o armazenamento de chaves externas para poder detectar os primeiros sinais de problemas operacionais e de performance.

Configurar a conectividade do serviço de endpoint da VPC

Use as orientações desta seção para criar e configurar os AWS recursos e os componentes relacionados que são necessários para um armazenamento de chaves externo que usa

conectividade de serviço de [endpoint VPC](#). Os recursos listados para essa opção de conectividade são um complemento aos [recursos obrigatórios para todos os armazenamentos de chaves externas](#). Depois de criar e configurar os recursos obrigatórios, você pode [criar seu armazenamento de chaves externas](#).

Você pode localizar seu proxy externo de armazenamento de chaves em sua Amazon VPC ou localizar o proxy fora dela AWS e usar seu serviço de endpoint de VPC para comunicação.

Antes de começar, [confirme se você precisa de um armazenamento de chaves externas](#). A maioria dos clientes pode usar chaves KMS apoiadas por material de AWS KMS chaves.

Note

Alguns dos elementos obrigatórios para a conectividade do serviço de endpoint da VPC podem estar incluídos no gerenciador de chaves externas. Além disso, seu software pode ter outros requisitos de configuração. Antes de criar e configurar os AWS recursos nesta seção, consulte a documentação do proxy e do gerenciador de chaves.

Tópicos

- [Requisitos para conectividade do serviço de endpoint da VPC](#)
- [Etapa 1: criar uma Amazon VPC e sub-redes](#)
- [Etapa 2: criar um grupo de destino](#)
- [Etapa 3: criar um Network Load Balancer](#)
- [Etapa 4: criar um serviço de endpoint da VPC](#)
- [Etapa 5: verificar o domínio do seu nome de DNS privado](#)
- [Etapa 6: Autorizar AWS KMS a conexão com o serviço de endpoint da VPC](#)

Requisitos para conectividade do serviço de endpoint da VPC

Se você escolher a conectividade do serviço de endpoint da VPC para seu armazenamento de chaves externas, serão necessários os recursos a seguir.

- Uma Amazon VPC conectada ao gerenciador de chaves externas. Deve ter pelo menos duas [sub-redes](#) privadas em duas zonas de disponibilidade diferentes.

Você pode usar uma Amazon VPC existente para seu armazenamento de chaves externas, desde que ela [atenda aos requisitos](#) de uso do armazenamento de chaves externas. Vários armazenamentos de chaves externas podem compartilhar uma Amazon VPC, mas cada armazenamento de chaves externas deve ter seu próprio serviço de endpoint da VPC e nome DNS privado.

- Um [serviço de endpoint da Amazon VPC desenvolvido pelo AWS PrivateLink](#) com um [balanceador de carga de rede](#) e um [grupo de destino](#).

O serviço de endpoint não pode exigir aceitação. Além disso, você deve adicionar o AWS KMS como entidade principal autorizada. Isso permite AWS KMS criar endpoints de interface para que ele possa se comunicar com seu proxy externo de armazenamento de chaves.

- Um nome DNS privado para o serviço de endpoint da VPC exclusivo em sua Região da AWS.

O nome DNS privado deve ser o subdomínio de um domínio público de nível superior. Por exemplo, se o nome DNS privado for `myproxy-private.xks.example.com`, ele deverá ser o subdomínio de um domínio público, como `xks.example.com` ou `example.com`.

É necessário [verificar a propriedade](#) do domínio DNS para o nome DNS privado.

- Um certificado TLS emitido por uma [autoridade de certificação pública](#) com suporte para o proxy de armazenamento de chaves externas.

O nome comum (CN) da entidade no certificado TLS deve corresponder ao nome DNS privado. Por exemplo, se o nome DNS privado for `myproxy-private.xks.example.com`, o CN no certificado TLS deverá ser `myproxy-private.xks.example.com` ou `*.xks.example.com`.

- Para minimizar a latência da rede, crie seus AWS componentes no [suporte Região da AWS](#) mais próximo do seu [gerenciador de chaves externo](#). Se possível, escolha uma região com um tempo de resposta (RTT) de 35 milissegundos ou menos.

Para todos os requisitos de um armazenamento de chaves externas, consulte [Organizar os pré-requisitos](#).

Etapa 1: criar uma Amazon VPC e sub-redes

A conectividade do serviço de endpoint da VPC exige uma Amazon VPC conectada ao gerenciador de chaves externas com pelo menos duas sub-redes privadas. Você pode criar uma Amazon VPC ou usar uma Amazon VPC existente que atenda aos requisitos para armazenamentos de chaves

externas. Para obter ajuda sobre como criar uma nova Amazon VPC, consulte [Criar uma VPC](#) no Guia do usuário da Amazon Virtual Private Cloud.

Requisitos para sua Amazon VPC

O serviço de endpoint VPC da Amazon deve ter as propriedades a seguir para funcionar com repositórios de chaves externos.

- Deve estar na mesma [região compatível](#) que o repositório de chaves externo.
- Requer pelo menos duas sub-redes privadas, cada uma em uma zona de disponibilidade diferente.
- O intervalo de endereços IP privados de sua Amazon VPC não deve se sobrepor ao intervalo de endereços IP privados do data center que hospeda seu [gerenciador de chaves externas](#).
- Todos os componentes devem ser usados IPv4.

Você tem muitas opções para conectar a Amazon VPC ao seu proxy de armazenamento de chaves externas. Escolha uma opção que atenda a suas necessidades de performance e segurança. Para ver uma lista, consulte [Conectar sua VPC a outras redes](#) e opções de conectividade de [Network-to-Amazon VPC](#). Para obter mais detalhes, consulte [Direct Connect](#) e o [Guia do usuário do AWS Site-to-Site VPN](#).

Criar uma Amazon VPC para seu armazenamento de chaves externas

Use as instruções a seguir para criar a Amazon VPC para o armazenamento de chaves externas. Uma Amazon VPC será necessária somente se você escolher a opção de [conectividade do serviço de endpoint da VPC](#). Você pode criar uma Amazon VPC existente que atenda aos requisitos de um armazenamento de chaves externas.

Siga as instruções no tópico [Criar uma VPC, sub-redes e outros recursos de VPC](#) usando os valores obrigatórios abaixo. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
IPv4 Bloco CIDR	Insira os endereços IP da VPC. O intervalo de endereços IP privados de sua Amazon VPC não deve se sobrepor ao intervalo de endereços IP privados do data center que hospeda seu gerenciador de chaves externas .
Número de zonas de	2 ou mais

Campo	Valor
disponibilidade (AZs)	
Número de sub-redes públicas	Nenhum é obrigatório (0)
Número de sub-redes privadas	Um para cada AZ
Gateways NAT	Nenhum é obrigatório.
Endpoints da VPC	Nenhum é obrigatório.
Enable DNS hostnames	Sim
Habilitar a resolução de DNS	Sim

Não se esqueça de testar a comunicação da VPC. Por exemplo, se o proxy de armazenamento de chaves externas não estiver localizado na Amazon VPC, crie uma instância do Amazon EC2 em sua Amazon VPC e verifique se a Amazon VPC pode se comunicar com seu proxy de armazenamento de chaves externas.

Conectar a VPC ao gerenciador de chaves externas

Conecte a VPC ao data center em que o gerenciador de chaves externas está hospedado usando uma das [opções de conectividade de rede](#) compatíveis com a Amazon VPC. Certifique-se de que a instância do Amazon EC2 na VPC (ou o proxy de repositório de chaves externo, se estiver na VPC) possa se comunicar com o data center e o gerenciador de chaves externas.

Etapa 2: criar um grupo de destino

Antes de criar o serviço de endpoint da VPC obrigatório, crie seus componentes obrigatórios, um balanceador de carga de rede (NLB) e um grupo de destino. O balanceador de carga de rede (NLB) distribui solicitações entre vários destinos íntegros, e qualquer um deles pode atender à

solicitação. Nesta etapa, você cria um grupo de destino com pelo menos dois hosts para seu proxy de armazenamento de chaves externas e registra seus endereços IP com o grupo de destino.

Siga as instruções no tópico [Configure a target group](#) (Configurar um grupo de destino) usando os valores obrigatórios a seguir. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Target type	Endereços IP
Protocolo	TCP
Porta	443
Tipo de endereço IP	IPv4
VPC	Escolha a VPC em que você criará o serviço de endpoint da VPC para seu armazenamento de chaves externas.
Protocolo e caminho de verificação de integridade	O protocolo e o caminho de verificação de integridade serão diferentes da configuração de proxy de armazenamento de chaves externas. Consulte a documentação do gerenciador de chaves externas ou do proxy de armazenamento de chaves externas. Para obter informações gerais sobre como configurar verificações de integridade para grupos de destino, consulte Health checks for your target groups (Verificações de integridade de grupos de destino) no Guia do usuário do Elastic Load Balancing para Network Load Balancers.
Rede	Outro endereço IP privado
IPv4 endereço	Os endereços privados do proxy de armazenamento de chaves externas
Portas	443

Etapa 3: criar um Network Load Balancer

O balanceador de carga de rede distribui o tráfego da rede, incluindo solicitações do AWS KMS para seu proxy de armazenamento de chaves externas, aos destinos configurados.

Siga as instruções no tópico [Configure a load balancer and a listener](#) (Configurar um balanceador de carga e um receptor) para configurar e adicionar um receptor e criar um balanceador de carga usando os valores obrigatórios a seguir. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Esquema	Interno
Tipo de endereço IP	IPv4
Mapeamento de rede	Escolha a VPC em que você criará o serviço de endpoint da VPC para seu armazenamento de chaves externas.
Mapeamento	Escolha as duas zonas de disponibilidade (pelo menos duas) que você configurou para as sub-redes da VPC. Verifique os nomes das sub-redes e o endereço IP privado.
Protocolo	TCP
Porta	443
Ação padrão: encaminhar para	Escolha o grupo de destino para seu balanceador de carga de rede.

Etapa 4: criar um serviço de endpoint da VPC

Normalmente, você cria um endpoint para um serviço. No entanto, ao criar um serviço de VPC endpoint, você é o provedor e AWS KMS cria um endpoint para seu serviço. Para um armazenamento de chaves externas, crie um serviço de endpoint da VPC com o balanceador de carga de rede criado na etapa anterior. O serviço de VPC endpoint pode estar no mesmo que seu armazenamento Conta da AWS de chaves externo ou em outro. Conta da AWS

Vários armazenamentos de chaves externas podem compartilhar uma Amazon VPC, mas cada armazenamento de chaves externas deve ter seu próprio serviço de endpoint da VPC e nome DNS privado.

Siga as instruções no tópico [Create an endpoint service](#) (Criar um serviço de endpoint) para criar seu serviço de endpoint da VPC com os valores obrigatórios a seguir. Para outros campos, aceite os valores padrão e forneça os nomes conforme solicitado.

Campo	Valor
Tipo de balanceador de carga	Rede
Balanceadores de carga disponíveis	Escolha o balanceador de carga de rede criado na etapa anterior. Se o novo balanceador de carga não for exibido na lista, verifique se o estado está ativo. Pode demorar alguns minutos para que o estado do balanceador de carga seja alterado de em provisionamento para ativo.
Aceitação obrigatória	Falso. Desmarque a caixa de seleção. Não exija aceitação. AWS KMS não é possível se conectar ao serviço de endpoint da VPC sem uma aceitação manual. Se a aceitação for obrigatória, as tentativas de criar o armazenamento de chaves externas falharão com uma exceção <code>XksProxyInvalidConfigurationException</code> .
Habilitar nome DNS privado	Associar um nome DNS privado ao serviço
Nome DNS privado	Insira um nome DNS privado que seja exclusivo na Região da AWS. O nome DNS privado deve ser o subdomínio de um domínio público de nível superior. Por exemplo, se o nome DNS privado for <code>myproxy-private.xks.example.com</code> , ele deverá ser o subdomínio de um domínio público, como <code>xks.example.com</code> ou <code>example.com</code> . Esse nome DNS privado deve corresponder ao nome comum (CN) da entidade no certificado TLS configurado em seu proxy de armazenamento de chaves externas. Por exemplo, se o nome DNS privado for <code>myproxy-</code>

Campo	Valor
	<code>private.xks.example.com</code> , o CN no certificado TLS deverá ser <code>myproxy-private.xks.example.com</code> ou <code>*.xks.example.com</code> . Se o certificado e o nome DNS privado não coincidirem, as tentativas de conectar um armazenamento de chaves externas ao proxy de armazenamento de chaves externas falharão com um código de erro de conexão de <code>XKS_PROXY_INVALID_TLS_CONFIGURATION</code> . Para obter detalhes, consulte Erros gerais de configuração.
Tipos de endereço IP compatíveis	IPv4

Etapa 5: verificar o domínio do seu nome de DNS privado

Quando você cria o serviço de endpoint da VPC, seu status de verificação de domínio é `pendingVerification`. Antes de usar o serviço de endpoint da VPC para criar um armazenamento de chaves externas, esse status deve ser `verified`. Para verificar se você é o proprietário do domínio associado ao nome DNS privado, é necessário criar um registro TXT em um servidor DNS público.

Por exemplo, se o nome DNS privado do seu serviço de VPC endpoint `myproxy-private.xks.example.com` for, você deverá criar um registro TXT em um domínio público, `xks.example.com` como `example.com` ou, o que for público. AWS PrivateLink procura o registro TXT primeiro `xks.example.com` e depois ligado. `example.com`

Tip

Depois que você adicionar um registro TXT, poderá levar alguns minutos para que o valor do status da verificação do domínio seja alterado de `pendingVerification` para `verify`.

Para começar, encontre o status de verificação do domínio usando um dos métodos a seguir. Os valores válidos são `verified`, `pendingVerification` e `failed`.

- No [console da Amazon VPC](#), escolha Endpoint services (Serviços de endpoint) e escolha o serviço de endpoint. No painel de detalhes, consulte Domain verification status (Status da verificação do domínio).
- Use a operação [DescribeVpcEndpointServiceConfigurations](#). O valor de State está no campo `ServiceConfigurations.PrivateDnsNameConfiguration.State`.

Se o status da verificação não for `verified`, siga as instruções no tópico [Domain ownership verification](#) (Verificação de propriedade do domínio) para adicionar um registro TXT ao servidor DNS do domínio e verificar se o registro TXT foi publicado. Em seguida, verifique o status de verificação novamente.

Não é necessário criar um registro A para o nome de domínio DNS privado. Ao AWS KMS criar um endpoint de interface para seu serviço de endpoint de VPC AWS PrivateLink, cria automaticamente uma zona hospedada com o registro A necessário para o nome de domínio privado na VPC. AWS KMS Para armazenamentos de chaves externas com conectividade de serviço de endpoint da VPC, isso acontece quando você [conecta seu armazenamento de chaves externas](#) ao proxy de armazenamento de chaves externas.

Etapa 6: Autorizar AWS KMS a conexão com o serviço de endpoint da VPC

Veja os procedimentos a seguir para gerenciar as permissões de serviço de endpoint da Amazon VPC. Cada etapa depende da conectividade e da configuração entre o repositório de chaves externo, o serviço de endpoint da VPC e a Conta da AWS.

Same Conta da AWS

Quando seu serviço de VPC endpoint pertence ao mesmo Conta da AWS que seu armazenamento de chaves externo, você deve adicioná-lo AWS KMS à lista Permitir diretores do seu serviço de VPC endpoint. Isso permite AWS KMS criar endpoints de interface para seu serviço de endpoint VPC. Se não AWS KMS for um principal permitido, as tentativas de criar um armazenamento de chaves externo falharão, com uma `XksProxyVpcEndpointServiceNotFoundException` exceção.

Siga as instruções no tópico [Manage permissions](#) (Gerenciar permissões) no Guia do AWS PrivateLink. Use o valor obrigatório a seguir.

Campo	Valor
AWS KMS ARN	<code>cks.kms.<region>.amazonaws.com</code> Por exemplo, <code>cks.kms.us-east-1.amazonaws.com</code> .

Cross Conta da AWS

Quando seu serviço de VPC endpoint é de propriedade de outra pessoa, Conta da AWS você deve adicionar ambos AWS KMS e sua conta à lista de diretores permitidos. Isso permite que AWS KMS seu armazenamento de chaves externo crie endpoints de interface para seu serviço de endpoint de VPC. Se o AWS KMS não for uma entidade principal autorizada, as tentativas de criar um armazenamento de chaves externas falharão, com a exceção `XksProxyVpcEndpointServiceNotFoundException`. Você precisará fornecer o Conta da AWS ARN em que o armazenamento externo de chaves reside.

Siga as instruções no tópico [Manage permissions](#) (Gerenciar permissões) no Guia do AWS PrivateLink . Use o valor obrigatório a seguir.

Campo	Valor
AWS KMS ARN	<code>cks.kms.<region>.amazonaws.com</code> Por exemplo, <code>cks.kms.us-east-1.amazonaws.com</code> .
Conta da AWS ARN	<code>arn:aws:iam:: 111122223333 :role/role_name</code> Por exemplo, <code>arn:aws:iam:: 123456789012 :role/cks_role</code> .

Próximo: [Criar um repositório de chaves externo](#)

Criar um repositório de chaves externo

Você pode criar um ou vários armazenamentos de chaves externos em cada Conta da AWS região. Cada armazenamento de chaves externo deve estar associado a um gerenciador de chaves externo e a um proxy externo de AWS armazenamento de chaves (proxy XKS) que medeia a comunicação entre AWS KMS e seu gerenciador de chaves externo. Para obter detalhes, consulte [Escolher uma](#)

[opção de conectividade de proxy de um repositório de chaves externo](#). Antes de começar, [confirme se você precisa de um armazenamento de chaves externas](#). A maioria dos clientes pode usar chaves KMS apoiadas por material de AWS KMS chaves.

 Tip

Alguns gerenciadores de chaves externas fornecem um método mais simples de criar um armazenamento de chaves externas. Para obter detalhes, consulte a documentação do gerenciador de chaves externas.

Antes de criar o armazenamento de chaves externas, você precisa [organizar os pré-requisitos](#). Durante o processo de criação, especifique as propriedades do armazenamento de chaves externas. Mais importante ainda, você indica se seu armazenamento de chaves externo AWS KMS usa um [endpoint público](#) ou um serviço de [VPC endpoint](#) para se conectar ao proxy externo do armazenamento de chaves. Você também especifica os detalhes da conexão, incluindo o endpoint de URI do proxy e o caminho dentro desse endpoint de proxy onde AWS KMS envia solicitações de API para o proxy.

Considerações

- O KMS não pode se comunicar IPv6 com armazenamentos de chaves externos.
- Se você usa conectividade de endpoint público, certifique-se de que ele AWS KMS possa se comunicar com seu proxy pela Internet usando uma conexão HTTPS. Isso inclui configurar o TLS no proxy externo do armazenamento de chaves e garantir que qualquer firewall entre AWS KMS e o proxy permita o IPv4 tráfego de e para a porta 443 no proxy. Ao criar um armazenamento de chaves externo com conectividade de endpoint público, AWS KMS testa a conexão enviando uma solicitação de status para o proxy externo do armazenamento de chaves. Esse teste verifica se o endpoint está acessível e se o proxy de armazenamento de chaves externas aceitará uma solicitação assinada com sua [credencial de autenticação de proxy de armazenamento de chaves externas](#). Se essa solicitação de teste falhar, a operação para criar o armazenamento de chaves externas falhará.
- Se você usa a conectividade do serviço de endpoint da VPC, verifique se o balanceador de carga de rede, o nome DNS privado e o serviço de endpoint da VPC estão configurados corretamente e em operação. Se o proxy externo do armazenamento de chaves não estiver na VPC, você precisará garantir que o serviço de endpoint da VPC possa se comunicar com o proxy externo do armazenamento de chaves. (AWS KMS testa a conectividade do serviço de endpoint VPC

quando você [conecta o armazenamento de chaves externo ao proxy externo do armazenamento de chaves](#).)

- AWS KMS registra [CloudWatch métricas e dimensões da Amazon](#), especialmente para lojas de chaves externas. Gráficos de monitoramento baseados em algumas dessas métricas aparecem no AWS KMS console para cada armazenamento externo de chaves. É altamente recomendável usar essas métricas para criar alarmes que monitorem o repositório de chaves externo. Esses alarmes alertam sobre os primeiros sinais de problemas operacionais e de performance antes que eles ocorram. Para instruções, consulte [Monitorar repositórios de chaves externos](#).
- Os armazenamentos de chaves externas estão sujeitos a [cotas de recursos](#). O uso de chaves do KMS em um armazenamento de chaves externas está sujeito às [cotas de solicitação](#). Analise essas cotas antes de projetar sua implementação de armazenamento de chaves externas.

Note

Revise sua configuração para ver se há dependências circulares que possam impedi-lo de funcionar.

Por exemplo, se você criar seu proxy de armazenamento de chaves externo usando AWS recursos, certifique-se de que a operação do proxy não exija a disponibilidade de uma chave KMS em um armazenamento de chaves externo que seja acessado por meio desse proxy.

Todos os novos armazenamentos de chaves externas são criados no estado desconectado. Antes de criar chaves do KMS em seu armazenamento de chaves externas, é necessário [conectá-lo](#) ao proxy de armazenamento de chaves externas. Para alterar as propriedades do armazenamento de chaves externas, [edite as configurações de armazenamento de chaves externas](#).

Tópicos

- [Organizar os pré-requisitos](#)
- [Criar um novo repositório de chaves externo](#)

Organizar os pré-requisitos

Antes de criar um armazenamento de chaves externo, você precisa montar os componentes necessários, incluindo o [gerenciador de chaves externo](#) que você usará para dar suporte ao armazenamento de chaves externo e o [proxy externo do armazenamento de chaves](#) que traduz as AWS KMS solicitações em um formato que seu gerenciador de chaves externo possa entender.

Os componentes a seguir são obrigatórios para todos os armazenamentos de chaves externas. Além desses componentes, é necessário fornecer os componentes que ofereçam suporte à [opção de conectividade do proxy de armazenamento de chaves externas](#) que você escolher.

 Tip

O gerenciador de chaves externas pode incluir alguns desses componentes ou eles poderão ser configurados para você. Para obter detalhes, consulte a documentação do gerenciador de chaves externas.

Se você estiver criando seu armazenamento externo de chaves no AWS KMS console, você tem a opção de carregar um [arquivo de configuração de proxy](#) baseado em JSON que especifica o [caminho do URI do proxy e a credencial](#) de autenticação do [proxy](#). Alguns proxies de armazenamento de chaves externas geram esse arquivo para você. Para obter detalhes, consulte a documentação do proxy de armazenamento de chaves externas ou do gerenciador de chaves externas.

Gerenciador de chaves externas

Cada armazenamento de chaves externas requer pelo menos uma instância do [gerenciador de chaves externas](#). Pode ser um módulo de segurança de hardware (HSM) físico ou virtual ou um software de gerenciamento de chaves.

Você pode usar um único gerenciador de chaves, mas recomendamos pelo menos duas instâncias de gerenciador de chaves relacionadas que compartilhem chaves de criptografia para redundância. O armazenamento de chaves externas não exige o uso exclusivo do gerenciador de chaves externas. No entanto, o gerenciador de chaves externo deve ter a capacidade de lidar com a frequência esperada de solicitações de criptografia e descriptografia dos AWS serviços que usam chaves KMS no armazenamento de chaves externo para proteger seus recursos. O gerenciador de chaves externas deve ser configurado para lidar com até 1.800 solicitações por segundo e responder dentro do tempo limite de 250 milissegundos para cada solicitação. Recomendamos que você localize o gerenciador de chaves externo próximo a um para Região da AWS que o tempo de ida e volta (RTT) da rede seja de 35 milissegundos ou menos.

Se o proxy de armazenamento de chaves externas permitir, será possível alterar o gerenciador de chaves externas que você associa ao proxy de armazenamento de chaves externas, mas o novo gerenciador de chaves externas deve ser um backup ou snapshot com o mesmo material de chave. Se a chave externa que você associa a uma chave KMS não estiver mais disponível para seu proxy

externo de armazenamento de chaves, AWS KMS não será possível descriptografar o texto cifrado criptografado com a chave KMS.

O gerenciador de chaves externas deve estar acessível ao proxy de armazenamento de chaves externas. Se a [GetHealthStatus](#) resposta do proxy informar que todas as instâncias externas do gerenciador de chaves são `Unavailable`, todas as tentativas de criar um armazenamento de chaves externo falharão com um [XksProxyUriUnreachableException](#).

Proxy de armazenamento de chaves externas

É necessário especificar um [proxy de armazenamento de chaves externas](#) (proxy XKS) que esteja em conformidade com os requisitos de design em [AWS KMS External Key Store Proxy API Specification](#) (Especificação da API de proxy de armazenamento de chaves externas do). Você pode desenvolver ou comprar um proxy externo de armazenamento de chaves ou usar um proxy externo de armazenamento de chaves fornecido ou incorporado ao seu gerenciador de chaves externo. AWS KMS recomenda que seu proxy externo de armazenamento de chaves seja configurado para lidar com até 1800 solicitações por segundo e responder dentro do tempo limite de 250 milissegundos para cada solicitação. Recomendamos que você localize o gerenciador de chaves externo próximo a um para Região da AWS que o tempo de ida e volta (RTT) da rede seja de 35 milissegundos ou menos.

Você pode usar um proxy de armazenamento de chaves externas para mais de um armazenamento de chaves externas, mas cada armazenamento de chaves externas deve ter um endpoint de URI e um caminho exclusivo dentro do proxy de armazenamento de chaves externas para as solicitações.

Se você estiver usando a conectividade do serviço de endpoint da VPC, poderá localizar o proxy de armazenamento de chaves externas na Amazon VPC, mas isso não é necessário. Você pode localizar seu proxy fora dele AWS, como em seu data center privado, e usar o serviço de VPC endpoint somente para se comunicar com o proxy.

Credencial de autenticação de proxy

Para criar um armazenamento de chaves externas, é necessário especificar sua credencial de autenticação de proxy de armazenamento de chaves externas (`XksProxyAuthenticationCredential`).

Você deve estabelecer uma [credencial de autenticação](#) (`XksProxyAuthenticationCredential`) AWS KMS no proxy externo do armazenamento de chaves. AWS KMS autentica seu proxy assinando suas solicitações usando o [processo Signature Version 4 \(SigV4\)](#) com a credencial de

autenticação de proxy do armazenamento de chaves externo. Você especifica a credencial de autenticação ao criar seu armazenamento de chaves externas e [pode alterá-la](#) a qualquer momento. Se o proxy alternar sua credencial, certifique-se de atualizar os valores da credencial para seu armazenamento de chaves externas.

A credencial de autenticação do proxy tem duas partes. É necessário fornecer as duas partes para o armazenamento de chaves externas.

- ID da chave de acesso: identifica a chave de acesso secreta. Você pode fornecer o ID em texto não criptografado.
- Chave de acesso secreta: a parte secreta da credencial. AWS KMS criptografa a chave de acesso secreta na credencial antes de armazená-la.

A credencial SigV4 AWS KMS usada para assinar solicitações ao proxy externo do armazenamento de chaves não está relacionada a nenhuma credencial SigV4 associada a nenhum principal em suas contas. AWS Identity and Access Management AWS Não reutilize nenhuma credencial SigV4 do IAM para seu proxy de armazenamento de chaves externas.

Conectividade do proxy

Para criar um armazenamento de chaves externas, é necessário especificar sua opção de conectividade do proxy de armazenamento de chaves externas (XksProxyConnectivity).

AWS KMS pode se comunicar com seu proxy externo de armazenamento de chaves usando um [endpoint público ou um serviço de endpoint da Amazon Virtual Private Cloud \(Amazon VPC\)](#). Embora seja mais simples de configurar e manter, um endpoint público pode não atender aos requisitos de segurança de todas as instalações. Se você escolher a opção de conectividade do serviço de endpoint da Amazon VPC, deverá criar e manter os componentes obrigatórios, inclusive uma Amazon VPC com pelo menos duas sub-redes em duas zonas de disponibilidade diferentes, um serviço de endpoint da VPC com um balanceador de carga de rede e grupo de destino e um nome DNS privado para o serviço de endpoint da VPC.

Você pode [alterar a opção de conectividade de proxy](#) para seu armazenamento de chaves externas. No entanto, você deve garantir a disponibilidade contínua do material de chave associado às chaves do KMS em seu armazenamento de chaves externas. Caso contrário, AWS KMS não será possível descriptografar nenhum texto cifrado criptografado com essas chaves KMS.

Para obter ajuda para decidir qual opção de conectividade de proxy é melhor para seu armazenamento de chaves externas, consulte [Escolher uma opção de conectividade de proxy de](#)

[um repositório de chaves externo](#). Para obter ajuda para criar uma configuração da conectividade do serviço de endpoint da VPC, consulte [Configurar a conectividade do serviço de endpoint da VPC](#).

Endpoint de URI do proxy

Para criar um armazenamento de chaves externo, você deve especificar o endpoint (XksProxyUriEndpoint) AWS KMS usado para enviar solicitações ao proxy externo do armazenamento de chaves.

O protocolo deve ser HTTPS. AWS KMS se comunica pela IPv4 porta 443. Não especifique a porta no valor do endpoint do URI do proxy.

- [Conectividade de endpoint público](#): especifique o endpoint disponível publicamente para seu proxy de armazenamento de chaves externas. Esse endpoint deve estar acessível antes de você criar o armazenamento de chaves externas.
- [Conectividade do serviço de endpoint da VPC](#): especifique `https://` seguido pelo nome DNS privado do serviço de endpoint da VPC.

O certificado do servidor TLS configurado no proxy do armazenamento de chaves externas deve corresponder ao nome do domínio no endpoint de URI do proxy de armazenamento de chaves externas e ser emitido por uma autoridade de certificação com suporte para armazenamentos de chaves externas. Para obter uma lista, consulte [Trusted Certificate Authorities](#) (Autoridades de certificação confiáveis). A autoridade de certificação exigirá prova de propriedade do domínio antes de emitir o certificado TLS.

O nome comum (CN) da entidade no certificado TLS deve corresponder ao nome DNS privado. Por exemplo, se o nome DNS privado for `myproxy-private.xks.example.com`, o CN no certificado TLS deverá ser `myproxy-private.xks.example.com` ou `*.xks.example.com`.

Você pode [alterar seu endpoint do URI do proxy](#), mas certifique-se de que o proxy do armazenamento de chaves externas tenha acesso ao material de chave associado às chaves do KMS de seu armazenamento de chaves externas. Caso contrário, AWS KMS não será possível descriptografar nenhum texto cifrado criptografado com essas chaves KMS.

Requisitos de exclusividade

- O valor combinado do endpoint (XksProxyUriEndpoint) do URI do proxy e do caminho do URI do proxy (XksProxyUriPath) deve ser exclusivo na região e Conta da AWS .

- Armazenamentos de chaves externas com conectividade de endpoint público podem compartilhar o mesmo endpoint do URI do proxy, desde que tenham valores do caminho do URI do proxy diferentes.
- Um armazenamento de chaves externo com conectividade de endpoint público não pode usar o mesmo valor de endpoint de URI de proxy que qualquer armazenamento de chaves externo com conectividade de serviços de endpoint VPC no mesmo Região da AWS, mesmo que os armazenamentos de chaves estejam em diferentes. Contas da AWS
- Cada armazenamento de chaves externas com conectividade de endpoint da VPC deve ter seu próprio nome DNS privado. O endpoint do URI do proxy (nome DNS privado) deve ser exclusivo na região Conta da AWS e.

Caminho do URI do proxy

Para criar um armazenamento de chaves externo, você deve especificar o caminho base em seu proxy de armazenamento de chaves externo para o [proxy necessário APIs](#). O valor deve começar com / e terminar com /kms/xks/v1, onde v1 representa a versão da AWS KMS API para o proxy externo do armazenamento de chaves. Esse caminho pode incluir um prefixo opcional entre os elementos obrigatórios, como /example-prefix/kms/xks/v1. Para encontrar esse valor, consulte a documentação do proxy de armazenamento de chaves externas.

AWS KMS envia solicitações de proxy para o endereço especificado pela concatenação do endpoint do URI do proxy e do caminho do URI do proxy. Por exemplo, se o endpoint do URI do proxy for `https://myproxy.xks.example.com` e o caminho do URI do proxy for /kms/xks/v1, AWS KMS envia suas solicitações de API do proxy para `https://myproxy.xks.example.com/kms/xks/v1`.

Você pode [alterar o caminho do URI do proxy](#), mas certifique-se de que o proxy do armazenamento de chaves externas tenha acesso ao material de chave associado às chaves do KMS de seu armazenamento de chaves externas. Caso contrário, AWS KMS não será possível descriptografar nenhum texto cifrado criptografado com essas chaves KMS.

Requisitos de exclusividade

- O valor combinado do endpoint (XksProxyUriEndpoint) do URI do proxy e do caminho do URI do proxy (XksProxyUriPath) deve ser exclusivo na região e Conta da AWS .

Serviço de VPC endpoint

Especifica o nome do serviço de endpoint da Amazon VPC usado para se comunicar com o proxy de armazenamento de chaves externas. Esse componente é obrigatório somente para armazenamentos de chaves externas que usam conectividade de serviços de endpoint da VPC. Para obter ajuda para instalar e configurar seu serviço de endpoint da VPC para um armazenamento de chaves externas, consulte [Configurar a conectividade do serviço de endpoint da VPC](#).

O serviço de endpoint da VPC deve ter as seguintes propriedades:

- O serviço de endpoint da VPC pode residir na mesma Conta da AWS que o repositório de chaves externo ou em outra.
- O serviço de endpoint da VPC deve residir no mesmo armazenamento Região da AWS de chaves externo.
- Você precisará fornecer o Conta da AWS ID do serviço de endpoint da VPC se ele residir em um local diferente. Conta da AWS
- Deve ter um balanceador de carga de rede (NLB) conectado a pelo menos duas sub-redes, cada uma em uma zona de disponibilidade diferente.
- A lista de diretores de permissão para o serviço de endpoint da VPC deve incluir AWS KMS o principal de serviço da Região `cks.kms.<region>.amazonaws.com`; como `cks.kms.us-east-1.amazonaws.com`
- Se o seu serviço de endpoint da Amazon VPC for de propriedade de Conta da AWS outra pessoa que não seja a Conta da AWS proprietária do armazenamento externo de chaves (XKS), você também precisará permitir que o XKS acesse o serviço de endpoint da VPC. Para fazer isso, [liste o Conta da AWS ID XKS como principal para o serviço de endpoint](#) da Amazon VPC.
- Não deve exigir a aceitação de solicitações de conexão.
- Deve ter um nome DNS privado dentro de um domínio público de nível superior. Por exemplo, você pode ter o nome DNS privado `myproxy-private.xks.example.com` no domínio público `xks.example.com`.

O nome DNS privado de um armazenamento de chaves externas com conectividade do serviço de endpoint da VPC deve ser exclusivo na Região da AWS.

- O [status da verificação do domínio](#) do nome DNS privado deverá ser `verified`.
- O certificado do servidor TLS configurado no proxy do armazenamento de chaves externas deve especificar o nome de host DNS privado no qual o endpoint pode ser acessado.

Requisitos de exclusividade

- Armazenamentos de chaves externas com conectividade de endpoint da VPC podem compartilhar uma Amazon VPC, mas cada armazenamento de chaves externas deve ter seu próprio serviço de endpoint da VPC e nome DNS privado.

Arquivo de configuração do proxy

Um arquivo de configuração de proxy é um arquivo opcional baseado em JSON que contém valores para o [caminho do URI do proxy](#) e as [propriedades da credencial de autenticação do proxy](#) do armazenamento de chaves externas. Ao criar ou [editar um armazenamento de chaves externas](#) no console do AWS KMS, é possível carregar um arquivo de configuração de proxy para fornecer valores de configuração para o armazenamento de chaves externas. Usar esse arquivo evita erros de digitação e colagem e garante que os valores do armazenamento de chaves externas correspondam aos valores do proxy de armazenamento de chaves externas.

Os arquivos de configuração do proxy são gerados pelo proxy de armazenamento de chaves externas. Para saber se o proxy de armazenamento de chaves externas oferece um arquivo de configuração de proxy, consulte a documentação do proxy de armazenamento de chaves externas.

Veja a seguir o exemplo de um arquivo de configuração de proxy bem formado com valores fictícios.

```
{
  "XksProxyUriPath": "/example-prefix/kms/xks/v1",
  "XksProxyAuthenticationCredential": {
    "AccessKeyId": "ABCDE12345670EXAMPLE",
    "RawSecretAccessKey": "0000EXAMPLEFA5FT0mCc3DrGUe2sti527BitkQ0Zr9M09+vE="
  }
}
```

Você pode carregar um arquivo de configuração de proxy somente ao criar ou editar um armazenamento de chaves externo no AWS KMS console. Você não pode usá-lo com as [UpdateCustomKeyStore](#) operações [CreateCustomKeyStore](#) ou, mas pode usar os valores no arquivo de configuração do proxy para garantir que os valores dos parâmetros estejam corretos.

Criar um novo repositório de chaves externo

Depois de reunir os pré-requisitos necessários, você pode criar um novo armazenamento externo de chaves no AWS KMS console ou usando a operação. [CreateCustomKeyStore](#)

Usando o AWS KMS console

Antes de criar um repositório de chaves externo, [escolha o tipo de conectividade de proxy](#) e verifique se você criou e configurou todos os [componentes necessários](#). Se precisar de ajuda para encontrar algum dos valores obrigatórios, consulte a documentação do proxy de armazenamento de chaves externas ou do software de gerenciamento de chaves.

Note

Ao criar um armazenamento de chave externo no Console de gerenciamento da AWS, você pode carregar um arquivo de configuração de proxy baseado em JSON com valores para o [caminho do URI do proxy](#) e a credencial de [autenticação do proxy](#). Alguns proxies geram esse arquivo para você. Não é obrigatório.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Armazenamentos de chaves personalizados), External key stores (Armazenamentos de chaves externas).
4. Escolha Create external key store (Criar armazenamento de chaves externas).
5. Digite um nome amigável para o armazenamento de chaves externas. O nome deve ser exclusivo entre todos os outros armazenamentos de chaves externas da conta.

Important

Não inclua informações confidenciais ou sigilosas nesse campo. Esse campo pode ser exibido em texto simples em CloudTrail registros e outras saídas.

6. Escolha o tipo de [conectividade de proxy](#).

A escolha de conectividade de proxy determina os [componentes obrigatórios](#) para o proxy de armazenamento de chaves externas. Para obter ajuda para fazer essa escolha, consulte [Escolher uma opção de conectividade de proxy de um repositório de chaves externo](#).

- a. Selecione Serviço de endpoint da VPC entre contas se seu serviço de endpoint da VPC residir em uma Conta da AWS diferente. Em seguida, insira o Conta da AWS ID do

proprietário do VPC endpoint no campo ID da conta do proprietário do serviço do VPC endpoint.

- b. Escolha ou insira o nome do [serviço de endpoint da VPC](#) para esse armazenamento de chaves externas. Essa etapa é exibida somente quando o tipo de conectividade de proxy do repositório de chaves externo de endpoint é serviço de endpoint da VPC.

O serviço de endpoint VPC e seu VPCs devem atender aos requisitos de um armazenamento de chaves externo. Para obter detalhes, consulte [the section called “Organizar os pré-requisitos”](#).

7. Escolha ou insira o nome do [serviço de endpoint da VPC](#) para esse armazenamento de chaves externas. Essa etapa é exibida somente quando o tipo de conectividade de proxy de armazenamento de chaves externas é o serviço de endpoint da VPC.

O serviço de endpoint VPC e seu VPCs devem atender aos requisitos de um armazenamento de chaves externo. Para obter detalhes, consulte [the section called “Organizar os pré-requisitos”](#).

8. Insira o [endpoint do URI do proxy](#). O protocolo deve ser HTTPS. AWS KMS se comunica pela IPv4 porta 443. Não especifique a porta no valor do endpoint do URI do proxy.

Se AWS KMS reconhecer o serviço de VPC endpoint que você especificou na etapa anterior, ele preencherá esse campo para você.

Para obter conectividade de endpoint público, insira um URI de endpoint disponível publicamente. Para obter conectividade do endpoint da VPC: insira `https://` seguido pelo nome DNS privado do serviço de endpoint da VPC.

9. Para inserir os valores do prefixo do [caminho do URI do proxy](#) e da [credencial de autenticação do proxy](#), carregue um arquivo de configuração do proxy ou insira os valores manualmente.
 - Se você tiver um [arquivo de configuração de proxy](#) opcional que contenha valores para o [caminho do URI do proxy](#) e a [credencial de autenticação do proxy](#), escolha Upload configuration file (Carregar arquivo de configuração). Siga as etapas para carregar o arquivo.

Quando o arquivo for carregado, o console exibirá os valores do arquivo em campos editáveis. Você pode alterar os valores agora ou [editar esses valores](#) depois que o armazenamento de chaves externas for criado.

Para exibir o valor da chave de acesso secreta, escolha Show secret access key (Exibir chave de acesso secreta).

- Caso não tenha um arquivo de configuração de proxy, você poderá inserir o caminho do URI do proxy e os valores da credencial de autenticação do proxy manualmente.
 - a. Caso não tenha um arquivo de configuração de proxy, insira o URI do proxy manualmente. O console fornece o valor/kms/xks/v1 necessário.

Se o [caminho do URI do proxy](#) incluir um prefixo opcional, como `example-prefix` em `/example-prefix/kms/xks/v1`, insira o prefixo no campo Proxy URI path prefix (Prefixo do caminho do URI do proxy). Senão, deixe o campo vazio.

- b. Caso não tenha um arquivo de configuração de proxy, insira a [credencial de autenticação do proxy](#) manualmente. Tanto o ID da chave de acesso como a chave de acesso secreta são obrigatórios.
 - Em Proxy credential: Access key ID (Credencial do proxy: ID da chave de acesso), insira o ID da chave de acesso da credencial de autenticação do proxy. O ID da chave de acesso identifica a chave de acesso secreta.
 - Em Proxy credential: Secret acces key (Credencial do proxy: chave de acesso secreta), insira a chave de acesso secreta da credencial de autenticação do proxy.

Para exibir o valor da chave de acesso secreta, escolha Show secret access key (Exibir chave de acesso secreta).

Esse procedimento não define nem altera a credencial de autenticação que você estabeleceu no proxy de armazenamento de chaves externas. Ele apenas associa esses valores ao armazenamento de chaves externas. Para obter informações sobre como configurar, alterar e alternar sua credencial de autenticação de proxy, consulte a documentação do proxy de armazenamento de chaves externas ou do software de gerenciamento de chaves.

Se a credencial de autenticação de proxy for alterada, [edite a configuração de credencial](#) para seu armazenamento de chaves externas.

10. Escolha Create external key store (Criar armazenamento de chaves externas).

Se o procedimento for bem-sucedido, o novo armazenamento de chaves externas será exibido na lista de armazenamentos de chaves externas na conta e região. Se ele for malsucedido, será exibida uma mensagem de erro descrevendo o problema e fornecendo ajuda para corrigi-lo. Se precisar de ajuda adicional, consulte [CreateKey erros na chave externa](#).

Próximo: os novos armazenamentos de chaves externas não são conectados automaticamente. Antes de criar AWS KMS keys em seu armazenamento de chaves externo, você deve [conectar o armazenamento de chaves externo](#) ao proxy de armazenamento de chaves externo.

Usando a AWS KMS API

Você pode usar a [CreateCustomKeyStore](#) operação para criar um novo armazenamento de chaves externo. Para obter ajuda para encontrar os valores dos parâmetros obrigatórios, consulte a documentação do proxy de armazenamento de chaves externas ou do software de gerenciamento de chaves.

Tip

Você não pode carregar um [arquivo de configuração de proxy](#) ao usar a operação `CreateCustomKeyStore`. Porém, você pode usar os valores no arquivo de configuração do proxy para garantir que os valores dos parâmetros estejam corretos.

Para criar um armazenamento de chaves externas, a operação `CreateCustomKeyStore` exige os valores de parâmetros a seguir.

- `CustomKeyStoreName`: um nome amigável para o armazenamento de chaves externas que é exclusivo na conta.

Important

Não inclua informações confidenciais ou sigilosas nesse campo. Esse campo pode ser exibido em texto simples em CloudTrail registros e outras saídas.

- `CustomKeyStoreType`: especifique `EXTERNAL_KEY_STORE`.
- [XksProxyConnectivity](#): especifique `PUBLIC_ENDPOINT` ou `VPC_ENDPOINT_SERVICE`.
- [XksProxyAuthenticationCredential](#): especifique o ID da chave de acesso e a chave de acesso secreta.
- [XksProxyUriEndpoint](#): o endpoint que o AWS KMS usa para se comunicar com o proxy de armazenamento de chaves externas.
- [XksProxyUriPath](#)— O caminho dentro do proxy até o proxy APIs.
- [XksProxyVpcEndpointServiceName](#): exigido somente quando o valor de `XksProxyConnectivity` é `VPC_ENDPOINT_SERVICE`.

 Note

Se você usa a AWS CLI versão 1.0, execute o comando a seguir antes de especificar um parâmetro com um valor HTTP ou HTTPS, como o `XksProxyUriEndpoint` parâmetro.

```
aws configure set cli_follow_urlparam false
```

Caso contrário, a AWS CLI versão 1.0 substitui o valor do parâmetro pelo conteúdo encontrado nesse endereço URI, causando o seguinte erro:

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
https:// : received non 200 status code of 404
```

Os exemplos a seguir usam valores fictícios. Antes de executar o comando, substitua-os por valores válidos para seu armazenamento de chaves externas.

Crie um armazenamento de chaves externas com conectividade pública de endpoints.

```
$ 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>
```

Crie um armazenamento de chaves externas com conectividade de serviço de endpoint da 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>
```

Quando a operação é bem-sucedida, `CreateCustomKeyStore` retorna o ID do armazenamento de chaves personalizado, conforme exibido na resposta de exemplo a seguir.

```
{
  "CustomKeyId": cks-1234567890abcdef0
}
```

Se a operação falhar, corrija o erro indicado pela exceção e tente novamente. Para obter ajuda adicional, consulte [Solução de problemas de armazenamentos de chaves externas](#).

Próximo: para usar o armazenamento de chaves externas, [conecte-o ao proxy do armazenamento de chaves externas](#).

Editar propriedades do repositório de chaves externo

Você pode editar as propriedades selecionadas de um armazenamento de chaves externas existente.

Você pode editar algumas propriedades enquanto o armazenamento de chaves externas está conectado ou desconectado. Para outras propriedades, é necessário primeiro [desconectar o armazenamento de chaves externas](#) do proxy de armazenamento de chaves externas. O [estado da conexão](#) do armazenamento de chaves externas deve ser DISCONNECTED. Enquanto seu armazenamento de chaves externas está desconectado, é possível gerenciar o armazenamento de chaves e suas chaves do KMS, mas não é possível criar nem usar chaves do KMS no armazenamento de chaves externas. Para encontrar o [estado da conexão](#) do seu armazenamento de chaves externo, use a [DescribeCustomKeyStores](#) operação ou consulte a seção Configuração geral na página de detalhes do armazenamento de chaves externo.

Antes de atualizar as propriedades do seu armazenamento de chaves externo, AWS KMS envia uma [GetHealthStatus](#) solicitação ao proxy do armazenamento de chaves externo usando os novos valores. Se a solicitação for bem-sucedida, isso indicará que você pode se conectar e se autenticar em um proxy de armazenamento de chaves externas com os valores de propriedade atualizados. Em caso de falha na solicitação, a operação de edição falhará com uma exceção que identifica o erro.

Quando a operação de edição for concluída, os valores de propriedade atualizados do seu armazenamento de chaves externo aparecerão no AWS KMS console e na `DescribeCustomKeyStores` resposta. No entanto, pode levar até cinco minutos para as alterações entrarem em vigor totalmente.

Se você editar seu armazenamento externo de chaves no AWS KMS console, terá a opção de carregar um [arquivo de configuração de proxy](#) baseado em JSON que especifica o [caminho do URI do proxy e a credencial](#) de autenticação do [proxy](#). Alguns proxies de armazenamento de chaves externas geram esse arquivo para você. Para obter detalhes, consulte a documentação do proxy de armazenamento de chaves externas ou do gerenciador de chaves externas.

Warning

Os valores de propriedade atualizados devem conectar seu armazenamento de chaves externas a um proxy para o mesmo gerenciador de chaves externas dos valores anteriores ou para um backup ou snapshot do gerenciador de chaves externas com as mesmas chaves de criptografia. Se o armazenamento de chaves externas perder o acesso às chaves externas associadas às chaves do KMS permanentemente, o texto cifrado criptografado sob essas chaves externas será irrecoverável. Em particular, alterar a conectividade proxy de um armazenamento de chaves externo pode AWS KMS impedir o acesso às chaves externas.

Tip

Alguns gerenciadores de chaves externas fornecem um método mais simples de editar propriedades de armazenamento de chaves externas. Para obter detalhes, consulte a documentação do gerenciador de chaves externas.

Você pode alterar as seguintes propriedades de um armazenamento de chaves externas.

Propriedades editáveis do armazenamento de chaves externas	Qualquer estado da conexão	Exigir estado Disconnected (Desconectado)
Nome do armazenamento de chaves personalizado		
Um nome amigável obrigatório para um armazenamento de chaves personalizado.		

Propriedades editáveis do armazenamento de chaves externas	Qualquer estado da conexão	Exigir estado Disconnected (Desconectado)
 Important Não inclua informações confidenciais ou sigilosas nesse campo. Esse campo pode ser exibido em texto simples em CloudTrail registros e outras saídas.		
<u>Credencial de autenticação de proxy</u> () XksProxyAuthenticationCredential (É necessário especificar o ID da chave de acesso e a chave de acesso secreta, mesmo que você esteja alterando apenas um elemento.)	✓	
<u>Caminho do URI do proxy</u> (XksProxyUriPath)	✓	
<u>Conectividade de proxy</u> (XksProxyConnectivity) (Também é necessário atualizar o endpoint do URI do proxy. Se estiver alterando para a conectividade do serviço de endpoint da VPC, você deverá especificar um nome de serviço de endpoint da VPC do proxy.)		✓
<u>Ponto final do URI do proxy</u> () XksProxyUriEndpoint Se você alterar o URI do endpoint do proxy, talvez também seja necessário alterar o certificado TLS.		✓
Nome do <u>serviço de endpoint VPC proxy</u> () XksProxyVpcEndpointServiceName (Este campo é obrigatório para a conectividade do serviço de endpoint da VPC)		✓

Propriedades editáveis do armazenamento de chaves externas	Qualquer estado da conexão	Exigir estado Disconnected (Desconectado)
Proprietário do serviço de endpoint VPC () XksProxyVpcEndpointServiceOwner (Este campo é obrigatório para a conectividade do serviço de endpoint da VPC)		

Editar as propriedades do seu repositório de chaves externo

Você pode editar as propriedades do seu armazenamento de chaves externo no AWS KMS console ou usando a [UpdateCustomKeyStore](#) operação.

Usando o AWS KMS console

Ao editar um armazenamento de chaves, você pode alterar qualquer um dos valores editáveis. Algumas alterações exigem que o armazenamento de chaves externas seja desconectado do proxy de armazenamento de chaves externas.

Se você estiver editando o caminho do URI do proxy ou a credencial de autenticação do proxy, poderá inserir os novos valores ou carregar um [arquivo de configuração de proxy](#) de armazenamento de chaves externas que contenha os novos valores.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Armazenamentos de chaves personalizados), External key stores (Armazenamentos de chaves externas).
4. Escolha o repositório de chaves que você deseja editar.
 - Se necessário, desconecte o armazenamento de chaves externas do proxy de armazenamento de chaves externas. No menu Key store actions (Ações do armazenamento de chaves), escolha Disconnect (Desconectar).
5. No menu Key store actions (Ações do armazenamento de chaves), escolha Edit (Editar).

6. Alterar uma ou mais das propriedades editáveis de armazenamento de chaves externas. Você também pode carregar um [arquivo de configuração de proxy](#) de armazenamento de chaves externas com valores para o caminho do URI do proxy e a credencial de autenticação do proxy. Você pode usar um arquivo de configuração de proxy mesmo que alguns valores especificados no arquivo não tenham sido alterados.
7. Escolha Update external key store (Atualizar armazenamento de chaves externas).
8. Revise o aviso e, se decidir continuar, confirme o aviso e escolha Update external key store (Atualizar armazenamento de chaves externas).

Se o procedimento for bem-sucedido, uma mensagem descreverá as propriedades que você editou. Se for malsucedido, será exibida uma mensagem de erro descrevendo o problema e fornecendo ajuda para corrigi-lo.

9. Se necessário, reconecte o armazenamento de chaves externas. No menu Key store actions (Ações do armazenamento de chaves), escolha Connect (Conectar).

Você pode deixar o armazenamento de chaves externas desconectado. Porém, enquanto estiver desconectado, não será possível criar as chaves do KMS no armazenamento de chaves externas nem usá-las no armazenamento de chaves externas em [operações de criptografia](#).

Usando a AWS KMS API

Para alterar as propriedades de um armazenamento de chaves externo, use a [UpdateCustomKeyStore](#) operação. Você pode alterar várias propriedades de um armazenamento de chaves externas na mesma operação. Se a operação for bem-sucedida, AWS KMS retornará uma resposta HTTP 200 e um objeto JSON sem propriedades.

Use o parâmetro CustomKeyId para identificar o armazenamento de chaves externas. Utilize os outros parâmetros para alterar as propriedades. Você não pode usar um [arquivo de configuração de proxy](#) com a operação UpdateCustomKeyStore. O arquivo de configuração do proxy é suportado somente pelo AWS KMS console. Contudo, é possível usar o arquivo de configuração do proxy para ajudar você a determinar os valores corretos dos parâmetros para o proxy de armazenamento de chaves externas.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Antes de começar, [se necessário, desconecte o armazenamento de chaves externas](#) do proxy de armazenamento de chaves externas. Após a atualização, se necessário, você pode [reconectar o](#)

[armazenamento de chaves externas](#) ao proxy do armazenamento de chaves externas. Você pode deixar o armazenamento de chaves externas no estado desconectado, mas é necessário reconectá-lo para que seja possível criar novas chaves do KMS no armazenamento de chaves ou usar as chaves do KMS existentes no armazenamento de chaves para operações de criptografia.

Note

Se você usa a AWS CLI versão 1.0, execute o comando a seguir antes de especificar um parâmetro com um valor HTTP ou HTTPS, como o `XksProxyUriEndpoint` parâmetro.

```
aws configure set cli_follow_urlparam false
```

Caso contrário, a AWS CLI versão 1.0 substitui o valor do parâmetro pelo conteúdo encontrado nesse endereço URI, causando o seguinte erro:

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve  
https:// : received non 200 status code of 404
```

Alterar o nome do armazenamento de chaves externas

O primeiro exemplo usa a [UpdateCustomKeyStore](#) operação para alterar o nome amigável do armazenamento externo de chaves para `XksKeyStore`. O comando usa o parâmetro `CustomKeyStoreId` para identificar o armazenamento de chaves personalizado e `CustomKeyStoreName` para especificar o novo nome para o armazenamento de chaves personalizado. Substitua todos os valores de exemplo por valores reais para o armazenamento de chaves externas.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-  
custom-key-store-name XksKeyStore
```

Alterar a credencial de autenticação de proxy

O exemplo a seguir atualiza a credencial de autenticação de proxy que o AWS KMS usa para se autenticar no proxy de armazenamento de chaves externas. Você pode usar um comando como este para atualizar a credencial, caso ela seja alternada em seu proxy.

Atualize primeiro a credencial no proxy de armazenamento de chaves externas. Em seguida, use esse recurso para relatar a alteração no AWS KMS. (Seu proxy oferecerá suporte brevemente à credencial antiga e à nova, para que você tenha tempo de atualizar sua credencial.) AWS KMS

É necessário sempre especificar o ID da chave de acesso e a chave de acesso secreta na credencial, mesmo que apenas um valor seja alterado.

Os dois primeiros comandos definem as variáveis para manter os valores das credenciais. As operações UpdateCustomKeyStore usam o parâmetro CustomKeyId para identificar o armazenamento de chaves externas. Utiliza o parâmetro XksProxyAuthenticationCredential com os campos AccessKeyId e RawSecretAccessKey para especificar a nova credencial. Substitua todos os valores de exemplo por valores reais para o armazenamento de chaves externas.

```
$ 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
```

Alterar o caminho do URI do proxy

O exemplo a seguir atualiza o caminho do URI do proxy (XksProxyUriPath). A combinação do endpoint do URI do proxy e do caminho do URI do proxy deve ser exclusiva na região Conta da AWS e. Substitua todos os valores de exemplo por valores reais para o armazenamento de chaves externas.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
  --xks-proxy-uri-path /kms/xks/v1
```

Alterar a conectividade do serviço de endpoint da VPC

O exemplo a seguir usa a [UpdateCustomKeyStore](#) operação para alterar o tipo de conectividade proxy do armazenamento de chaves externo para VPC_ENDPOINT_SERVICE. Para fazer essa alteração, é necessário especificar os valores obrigatórios para a conectividade do serviço de endpoint da VPC, inclusive o nome do serviço de endpoint da VPC (XksProxyVpcEndpointServiceName) e um valor de endpoint do URI do proxy (XksProxyUriEndpoint) que inclua o nome DNS privado do serviço de endpoint da VPC. Substitua todos os valores de exemplo por valores reais para o armazenamento de chaves externas.

```
$ 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
```

Alteração na conectividade de endpoints públicos

O exemplo a seguir altera o tipo de conectividade de proxy de armazenamento de chaves externas para PUBLIC_ENDPOINT. Ao fazer essa alteração, é necessário atualizar o valor do endpoint do URI do proxy (XksProxyUriEndpoint). Substitua todos os valores de exemplo por valores reais para o armazenamento de chaves externas.

Note

A conectividade de endpoint da VPC oferece maior segurança do que a conectividade de endpoint público. Antes de mudar para a conectividade de endpoint público, considere outras opções, como localizar seu proxy de armazenamento de chaves externas on-premises e usar a VPC somente para comunicação.

```
$ 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
```

Visualizar repositórios de chaves externos

Você pode visualizar os armazenamentos externos de chaves em cada conta e região usando o AWS KMS console ou usando a [DescribeCustomKeyStores](#) operação.

Ao visualizar um armazenamento de chaves externas, você pode ver o seguinte:

- Informações básicas sobre o armazenamento de chaves, inclusive seu nome amigável, ID, tipo de armazenamento de chaves e data de criação.
- Informações de configuração para o [proxy de armazenamento de chaves externa](#), inclusive o [tipo de conectividade](#), o [endpoint](#) e o [caminho do URI do proxy](#) e o [ID da chave de acesso](#) da [credencial de autenticação de proxy](#) atual.

- Se o proxy de armazenamento de chaves externas usar a [conectividade do serviço de endpoint da VPC](#), o console exibirá o nome do serviço de endpoint da VPC.
- O [estado da conexão](#) atual.

Note

Um valor do estado da conexão Disconnected (Desconectado) indica que o armazenamento de chaves externas nunca foi conectado ou que foi desconectado do proxy do armazenamento de chaves externas intencionalmente. No entanto, se as tentativas de usar uma chave do KMS em um armazenamento de chaves externas conectado falharem, pode ser indício de um problema com o armazenamento de chaves externas ou o proxy. Para obter ajuda, consulte [Erros de conexão do armazenamento de chaves externas](#).

- Uma seção de [monitoramento](#) com gráficos das [CloudWatch métricas da Amazon](#) projetada para ajudá-lo a detectar e resolver problemas com seu armazenamento externo de chaves. Para obter ajuda na interpretação dos gráficos, no uso deles no planejamento e na solução de problemas e na criação de CloudWatch alarmes com base nas métricas dos gráficos, consulte [Monitorar repositórios de chaves externos](#)

Propriedades do armazenamento de chaves externas

As seguintes propriedades de um armazenamento de chaves externo são visíveis no AWS KMS console e na [DescribeCustomKeyStores](#) resposta.

Propriedades do armazenamento de chaves personalizado

Os valores a seguir aparecem na seção Configuração geral da página de detalhes de cada armazenamento de chaves personalizadas. Essas propriedades se aplicam a todos os armazenamentos de chaves personalizadas, incluindo armazenamentos de chaves e armazenamentos de AWS CloudHSM chaves externos.

ID do armazenamento de chaves personalizado

Uma ID exclusiva que é AWS KMS atribuída ao armazenamento de chaves personalizadas.

Nome do armazenamento de chaves personalizado

Um nome amigável que você atribui ao armazenamento de chaves personalizado ao criá-lo. Você pode alterar esse valor a qualquer momento.

Tipo do armazenamento de chaves personalizado

O tipo do armazenamento de chaves personalizado. Os valores válidos são AWS CloudHSM (AWS_CLOUDHSM) ou External key store (Armazenamento de chaves externas) (EXTERNAL_KEY_STORE). Você não pode alterar o tipo depois de criar o armazenamento de chaves personalizado.

Data de Criação

A data em que o armazenamento de chaves personalizado foi criado. Essa data é exibida na hora local da Região da AWS.

Estado da conexão

Indica se o armazenamento de chaves personalizado está conectado a seu armazenamento de chaves de reserva. O estado da conexão será DISCONNECTED somente se o armazenamento de chaves personalizado nunca tiver sido conectado ao armazenamento de chaves de reserva ou se tiver sido desconectado intencionalmente. Para obter detalhes, consulte [the section called “Estado da conexão”](#).

Propriedades de configuração do armazenamento de chaves externas

Os valores a seguir aparecem na seção Configuração de proxy do armazenamento de chaves externo da página de detalhes de cada armazenamento de chaves externo e no XksProxyConfiguration elemento da [DescribeCustomKeyStores](#) resposta. Para obter uma descrição detalhada de cada campo, inclusive requisitos de exclusividade e ajuda para determinar o valor correto para cada campo, consulte [the section called “Organizar os pré-requisitos”](#) no tópico Criar um armazenamento de chaves externas.

Conectividade do proxy

Indica se o armazenamento de chaves externas usa [conectividade de endpoint público](#) ou [conectividade de serviço de endpoint da VPC](#).

Endpoint de URI do proxy

O endpoint AWS KMS usado para se conectar ao [proxy externo do armazenamento de chaves](#).

Caminho do URI do proxy

O caminho do endpoint do URI do proxy para o qual AWS KMS envia [solicitações da API do proxy](#).

Credencial de proxy: ID da chave de acesso

Parte da [credencial de autenticação de proxy](#) que você estabelece em seu proxy de armazenamento de chaves externas. O ID da chave de acesso identifica a chave de acesso secreta na credencial.

AWS KMS usa o processo de assinatura SigV4 e a credencial de autenticação do proxy para assinar suas solicitações no proxy externo do armazenamento de chaves. A credencial na assinatura permite que o proxy externo do armazenamento de chaves autentique solicitações em seu nome de. AWS KMS

Nome do serviço de endpoint da VPC

O nome do serviço de endpoint da Amazon VPC que oferece suporte ao armazenamento de chaves externas. Esse valor é exibido somente quando o armazenamento de chaves externas usa a [conectividade de serviço de endpoint da VPC](#). Você pode localizar o proxy de armazenamento de chaves externas na VPC ou usar o serviço de endpoint da VPC para se comunicar de forma segura com o proxy de armazenamento de chaves externas.

ID do proprietário do serviço de endpoint da VPC

O ID do serviço de endpoint da Amazon VPC compatível com o repositório de chaves externo. Esse valor é exibido somente quando o armazenamento de chaves externas usa a [conectividade de serviço de endpoint da VPC](#). Você pode localizar o proxy de armazenamento de chaves externas na VPC ou usar o serviço de endpoint da VPC para se comunicar de forma segura com o proxy de armazenamento de chaves externas.

Visualizar as propriedades do seu repositório de chaves externo

Você pode visualizar seu armazenamento externo de chaves e suas propriedades associadas no AWS KMS console ou usando a [DescribeCustomKeyStores](#) operação.

Usando o AWS KMS console

Para visualizar armazenamentos de chaves externas em uma determinada conta e região, use o procedimento a seguir.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.

3. No painel de navegação, selecione Custom key stores (Armazenamentos de chaves personalizados), External key stores (Armazenamentos de chaves externas).
4. Para exibir informações detalhadas sobre um armazenamento de chaves externas, escolha o nome do armazenamento de chaves.

Usando a AWS KMS API

Para visualizar seus armazenamentos externos de chaves, use a [DescribeCustomKeyStores](#) operação. Por padrão, essa operação retorna todos os armazenamentos de chaves personalizados na conta e região. No entanto, você pode usar o parâmetro CustomKeyId ou CustomKeyName (mas não ambos) para limitar o resultado para um determinado armazenamento de chaves personalizado.

Para armazenamentos de chaves personalizados, a saída consiste no ID, nome e tipo do armazenamento de chaves personalizado e no [estado da conexão](#) do armazenamento de chaves. Se o estado da conexão é FAILED, o resultado também inclui um ConnectionErrorCode que descreve o motivo do erro. Para obter ajuda para interpretar ConnectionErrorCode para um armazenamento de chaves externas, consulte [Códigos de erro de conexão para armazenamentos de chaves externas](#).

Para armazenamentos de chaves externas, a saída também inclui o elemento XksProxyConfiguration. Esse elemento inclui o [tipo de conectividade](#), o [endpoint do URI do proxy](#), o [caminho do URI do proxy](#) e o ID da chave de acesso da [credencial de autenticação do proxy](#).

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Por exemplo, o comando a seguir retorna todos os armazenamentos de chaves personalizados na conta e região. Você pode usar os parâmetros Marker e Limit para percorrer os armazenamentos de chaves personalizados do resultado.

```
$ aws kms describe-custom-key-stores
```

O comando a seguir usa o parâmetro CustomKeyName para obter apenas o armazenamento de chaves externas de exemplo com o nome amigável ExampleXksPublic. Este exemplo de armazenamento de chaves usa conectividade pública de endpoint. Está conectado ao proxy de armazenamento de chaves externas.

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

O comando a seguir obtém um exemplo de armazenamento de chaves externas com conectividade de serviço de endpoint da VPC. Neste exemplo, o armazenamento de chaves externas está conectado ao proxy de armazenamento de chaves externas.

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

O [ConnectionState](#) `Disconnected` indica que um armazenamento de chaves externas nunca foi conectado ou foi intencionalmente desconectado do proxy de armazenamento de chaves externas. No entanto, se as tentativas de usar uma chave do KMS em um armazenamento de chaves externas conectado falharem, pode ser indício de um problema com o proxy de armazenamento de chaves externas ou outros componentes externos.

Quando o `ConnectionState` do armazenamento de chaves externas `FAILED`, a resposta `DescribeCustomKeyStores` inclui um elemento `ConnectionErrorCode` que explica o motivo desse erro.

Por exemplo, na saída a seguir, o `XKS_PROXY_TIMED_OUT` valor indica que AWS KMS pode se conectar ao proxy externo do armazenamento de chaves, mas a conexão falhou porque o proxy do armazenamento de chaves externo não respondeu AWS KMS no tempo alocado. Caso veja esse código de erro de conexão repetidamente, notifique seu fornecedor de proxy de armazenamento de chaves externas. Para obter ajuda com relação a esta e outras falhas de erro de conexão, consulte [Solução de problemas de armazenamentos de chaves externas](#).

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


Monitorar repositórios de chaves externos

AWS KMS coleta métricas para cada interação com um armazenamento de chaves externo e as publica em sua CloudWatch conta. Essas métricas são usadas para gerar os grafos na seção de monitoramento da página de detalhes de cada armazenamento de chaves externas. O tópico a seguir detalha como usar os grafos para identificar e solucionar problemas operacionais e de configuração que afetam o armazenamento de chaves externas. Recomendamos usar as CloudWatch métricas para definir alarmes que notifiquem você quando seu armazenamento externo de chaves não estiver funcionando conforme o esperado. Para obter mais informações, consulte [Monitoramento com a Amazon CloudWatch](#).

Tópicos

- [Visualizar os grafos](#)
- [Interpretar os grafos](#)

Visualizar os grafos

Você pode visualizar os grafos em diferentes níveis de detalhes. Por padrão, cada grafo usa um intervalo de três horas e um [período](#) de agregação de cinco minutos. Você pode ajustar a visualização do grafo no console, mas suas alterações serão revertidas para as configurações padrão quando a página de detalhes do armazenamento de chaves externas for fechada ou quando o navegador for atualizado. Para obter ajuda com a CloudWatch terminologia da Amazon, consulte os [CloudWatch conceitos da Amazon](#).

Visualizar detalhes do ponto de dados

Os dados de cada grafo são coletados por [métricas do AWS KMS](#). Para visualizar mais informações sobre um ponto de dados específico, pause o mouse sobre o ponto de dados no grafo linha. Isso exibirá um pop-up com mais informações sobre a métrica da qual o grafo foi derivado. Cada item da lista exibe o valor de [dimensão](#) registrado no ponto de dados. O pop-up exibirá um valor nulo (–) se não houver dados métricos disponíveis para o valor da dimensão no ponto de dados. Alguns grafos registram várias dimensões e valores para um único ponto de dados. Outros grafos, como o [grafo de confiabilidade](#), usam os dados coletados pela métrica para calcular um valor exclusivo. Cada item da lista é associado a uma cor de grafo linha diferente.

Modificar o intervalo de tempo

Para alterar o [intervalo de tempo](#), selecione um dos intervalos de tempo predefinidos no canto superior direito da seção de monitoramento. Os intervalos de tempo predefinidos variam de uma hora a uma semana (1h, 3h, 12h, 1d, 3d ou 1w). Isso ajusta o intervalo de tempo para todos os grafos. Se você quiser visualizar um gráfico específico em um intervalo de tempo diferente, ou se quiser definir um intervalo de tempo personalizado, amplie o gráfico ou visualize-o no CloudWatch console da Amazon.

Ampliar grafos

Você pode usar o [recurso de zoom de minimapa](#) para focar em seções de grafos linhas e grafos de área empilhada sem alterações entre as visualizações com zoom aumentado e zoom diminuído. Por exemplo, é possível usar o recurso de zoom de minimapa para focar em um pico em um grafo linha para comparar o pico com outras métricas no painel usando a mesma linha do tempo.

1. Escolha e arraste a área do gráfico em que deseja focar e, em seguida, solte-a.
2. Para redefinir o zoom, escolha o ícone Redefinir zoom igual a uma lupa com um símbolo de menos (-) no interior.

Ampliar um grafo

Para ampliar um grafo, selecione o ícone de menu no canto superior direito de um grafo individual e escolha Enlarge (Ampliar). Você também pode selecionar o ícone de ampliação exibido ao lado do ícone de menu ao passar o mouse sobre um grafo.

Ampliar um grafo permite modificar ainda mais a visualização de um grafo especificando um período diferente, um intervalo de tempo personalizado ou um intervalo de atualização. Essas alterações serão revertidas para as configurações padrão quando você fechar a visualização ampliada.

Modificar o período

1. Escolha o menu Period options (Opções do período). Por padrão, esse menu exibe o valor: 5 minutes (5 minutos).
2. Escolha um período, os períodos predefinidos variam de um segundo a 30 dias.

Por exemplo, escolha uma exibição de um minuto, que pode ser útil durante a solução de problemas. Ou escolha uma visualização menos detalhada de uma hora. Isso pode ser útil ao visualizar um intervalo de tempo mais amplo (por exemplo, 3 dias) para que possa ver

tendências ao longo do tempo. Para obter mais informações, consulte [Períodos](#) no Guia CloudWatch do usuário da Amazon.

Modificar o intervalo de tempo ou fuso horário

1. Selecione um dos intervalos de tempo predefinidos, que variam de uma hora a uma semana: (1h, 3h, 12h, 1d, 3d ou 1w). Como alternativa, você pode escolher Custom (Personalizado) para definir seu próprio intervalo de tempo.
2. Escolha Custom (Personalizado)
 - a. Intervalo de tempo: selecione a guia Absolute (Absoluto) no canto superior esquerdo da caixa. Use o seletor de calendário ou as caixas de campos de texto para especificar um intervalo de tempo.
 - b. Fuso horário: escolha a lista suspensa no canto superior direito da caixa. Você pode alterar o fuso horário para UTC ou Local time zone (Fuso horário local).
3. Depois de especificar um período, escolha Apply (Aplicar).

Modifique a frequência com que os dados do grafo serão atualizados

1. No canto superior direito, escolha o menu Refresh options (Opções de atualização).
2. Escolha um intervalo de atualização: Off (Desligado), 10 Seconds (10 segundos), 1 Minute (1 minuto), 2 Minutes (2 minutos), 5 Minutes (5 minutos) ou 15 Minutes (15 minutos).

Visualize gráficos no console da Amazon CloudWatch

Os gráficos na seção de monitoramento são derivados de métricas predefinidas que são AWS KMS publicadas na Amazon. CloudWatch Você pode abri-los no CloudWatch console e salvá-los nos CloudWatch painéis. Se você tiver vários armazenamentos de chaves externos, poderá abrir seus respectivos gráficos CloudWatch e salvá-los em um único painel para comparar sua integridade e uso.

Adicionar ao CloudWatch painel

Selecione Adicionar ao painel no canto superior direito para adicionar todos os gráficos a um CloudWatch painel da Amazon. Você pode selecionar um painel existente ou criar outro. Para obter informações sobre como usar esse painel para criar visualizações personalizadas dos gráficos e alarmes, consulte Usando [CloudWatch painéis da Amazon no Guia do usuário](#) da Amazon CloudWatch .

Exibir nas CloudWatch métricas

Selecione o ícone do menu no canto superior direito de um gráfico individual e escolha Exibir em métricas para visualizar esse gráfico no CloudWatch console da Amazon. No CloudWatch console, você pode adicionar esse único gráfico a um painel e modificar intervalos de tempo, períodos e intervalos de atualização. Para obter mais informações, consulte [Representação gráfica de métricas](#) no Guia do CloudWatch usuário da Amazon.

Interpretar os grafos

AWS KMS fornece vários gráficos para monitorar a integridade do seu armazenamento externo de chaves no AWS KMS console. Esses grafos são configurados automaticamente e derivados de [métricas do AWS KMS](#).

Os dados de grafos são coletados como parte das chamadas que você faz para o armazenamento de chaves externas e chaves externas. Você pode ver dados preenchendo gráficos durante um intervalo de tempo em que você não fez nenhuma chamada. Esses dados vêm das GetHealthStatus chamadas periódicas feitas em seu nome para verificar o status do proxy externo do armazenamento de chaves e do gerenciador de chaves externo. AWS KMS Se os grafos exibirem a mensagem No data available (Nenhum dado disponível), não houve chamadas gravadas durante esse intervalo de tempo ou o armazenamento de chaves externas está no estado [DISCONNECTED](#). Talvez você consiga identificar a hora em que o armazenamento de chaves externas foi desconectado [ajustando a visualização](#) para um intervalo de tempo mais amplo.

Tópicos

- [Total requests](#)
- [Confiabilidade](#)
- [Latência](#)
- [As cinco principais exceções](#)
- [Dias para o certificado expirar](#)

Total requests

O número total de AWS KMS solicitações recebidas para um armazenamento de chaves externo específico durante um determinado intervalo de tempo. Use esse grafo para determinar se você corre o risco de controle de utilização.

AWS KMS recomenda que seu gerenciador de chaves externo seja capaz de lidar com até 1800 solicitações de operações criptográficas por segundo. Se você abordar 540 mil chamadas em um período de cinco minutos, correrá o risco de controle de utilização.

Você pode monitorar o número de solicitações de operações criptográficas em chaves KMS em seu armazenamento de chaves externo que acelera com a AWS KMS métrica. [ExternalKeyStoreThrottle](#)

Se você estiver recebendo erros `KMSInvalidStateException` muito frequentes com uma mensagem explicando que a solicitação foi rejeitada “due to a very high request rate” [devido a uma taxa de solicitação muito alta], isso pode indicar que o gerenciador de chaves externas ou o proxy de armazenamento de chaves externas não consegue acompanhar a taxa de solicitação atual. Se possível, reduza a taxa de solicitação. Considere também solicitar uma redução no valor da cota de solicitação de armazenamento de chaves personalizado. Diminuir esse valor de cota pode aumentar a limitação, mas indica que AWS KMS está rejeitando solicitações em excesso rapidamente antes de serem enviadas ao proxy externo do armazenamento de chaves ou ao gerenciador de chaves externo. Para solicitar uma redução de cota, acesse o [AWS Support Center](#) e crie um caso.

O grafo do total de solicitações é derivado da métrica [XksProxyErrors](#), que coleta dados sobre as respostas bem-sucedidas e malsucedidas que o AWS KMS recebe de seu proxy de armazenamento de chaves externas. Quando você [visualiza um ponto de dados específico](#), o pop-up exibe o valor da `CustomKeyId` dimensão junto com o número total de AWS KMS solicitações registradas nesse ponto de dados. `CustomKeyId` será sempre o mesmo.

Confiabilidade

A porcentagem de AWS KMS solicitações para as quais o proxy externo do armazenamento de chaves retornou uma resposta bem-sucedida ou um erro que não pode ser repetido. Use esse grafo para avaliar a integridade operacional do proxy de armazenamento de chaves externas.

Ao exibir um valor menor que 100%, o grafo indica casos em que o proxy não respondeu ou respondeu com um erro com nova tentativa. Isso pode indicar problemas com a rede, lentidão do proxy de armazenamento de chaves externas ou do gerenciador de chaves externas ou bugs de implementação.

Se a solicitação incluir uma credencial inválida e seu proxy responder com `AuthenticationFailedException`, o grafo ainda indicará 100% de confiabilidade porque o proxy identificou um valor incorreto na [solicitação da API do proxy de armazenamento de chaves externas](#) e, portanto, a falha é esperada. Se a porcentagem do grafo de confiabilidade for de 100%, o proxy de armazenamento de chaves externas estará respondendo conforme o esperado. Se o grafo

exibir um valor menor que 100%, o proxy respondeu com um erro com nova tentativa ou atingiu o tempo limite. Por exemplo, se o proxy responder com `ThrottlingException` devido a uma taxa de solicitação muito alta, ele exibirá uma porcentagem de confiabilidade menor porque o proxy não conseguiu identificar um problema específico na solicitação que causou a falha. Isso ocorre porque os erros com nova tentativa provavelmente são problemas transitórios que podem ser resolvidos ao repetir a solicitação.

As respostas de erro a seguir reduzirão a porcentagem de confiabilidade. Você pode usar o grafo [As cinco principais exceções](#) e a métrica [XksProxyErrors](#) para monitorar ainda mais a frequência com que seu proxy retorna cada erro com nova tentativa.

- `InternalException`
- `DependencyTimeoutException`
- `ThrottlingException`
- `XksProxyUnreachableException`

O gráfico de confiabilidade é derivado da [XksProxyErrors](#) métrica, que coleta dados sobre as respostas bem-sucedidas e malsucedidas AWS KMS recebidas do proxy externo do armazenamento de chaves. A porcentagem de confiabilidade só diminuirá se a resposta tiver um valor `ErrorType` de `Retryable`. Quando você [visualiza um ponto de dados específico](#), o pop-up exibe o valor da `CustomKeyStoreId` dimensão junto com a porcentagem de confiabilidade das AWS KMS solicitações registradas nesse ponto de dados. `CustomKeyStoreId` será sempre o mesmo.

Recomendamos usar a [XksProxyErrors](#) métrica para criar um CloudWatch alarme que o notifique sobre possíveis problemas de rede, alertando-o quando mais de cinco erros repetidos forem registrados em um período de um minuto. Para obter mais informações, consulte [Criar um alarme para erros passíveis de nova tentativa](#).

Latência

O número de milissegundos necessários para que um proxy externo de armazenamento de chaves responda a uma AWS KMS solicitação. Use esse grafo para avaliar a performance do proxy de armazenamento de chaves externas e do gerenciador de chaves externas.

AWS KMS espera que o proxy externo do armazenamento de chaves responda a cada solicitação em 250 milissegundos. No caso de tempos limite de rede, AWS KMS tentará novamente a solicitação uma vez. Se o proxy falhar pela segunda vez, a latência registrada será o limite de

tempo limite combinado para as duas tentativas de solicitação, e o grafo exibirá aproximadamente 500 milissegundos. Em todos os outros casos em que o proxy não responder dentro do tempo limite de 250 milissegundos, a latência registrada será de 250 milissegundos. Se o proxy estiver frequentemente atingindo o tempo limite nas operações de criptografia e descryptografia, consulte seu administrador do proxy externo. Para obter ajuda na solução de problemas de latência, consulte [Erros de latência e de tempo limite](#).

Respostas lentas também podem indicar que seu gerenciador de chaves externo não consegue lidar com o tráfego de solicitações atual. AWS KMS recomenda que seu gerenciador de chaves externo seja capaz de lidar com até 1800 solicitações de operações criptográficas por segundo. Se o gerenciador de chaves externas não conseguir lidar com a taxa de 1800 solicitações por segundo, considere [solicitar uma redução na cota de solicitações de chaves do KMS em um armazenamento de chaves personalizado](#). As solicitações de operações de criptografia que usam as chaves do KMS em seu armazenamento de chaves externas se anteciparão à falha com uma [exceção de controle de utilização](#), em vez de serem processadas e posteriormente rejeitadas pelo proxy de armazenamento de chaves externas ou pelo gerenciador de chaves externas.

O grafo de latência é derivado da métrica [XksProxyLatency](#). Quando você [visualiza um ponto de dados específico](#), o pop-up exibe os valores correspondentes de dimensão `KmsOperation` e `XksOperation` junto com a latência média registrada para as operações no ponto de dados. Os itens da lista são ordenados da maior para a menor latência.

Recomendamos usar a [XksProxyLatency](#) métrica para criar um CloudWatch alarme que notifique você quando sua latência estiver se aproximando do limite de tempo limite. Para obter mais informações, consulte [Criar um alarme para tempo limite de resposta](#).

As cinco principais exceções

As cinco principais exceções para falhas nas operações de criptografia e de gerenciamento durante um intervalo de tempo específico. Use esse grafo para rastrear os erros mais frequentes e poder priorizar seus esforços de engenharia.

Essa contagem inclui as exceções AWS KMS recebidas do proxy externo do armazenamento de chaves e as `XksProxyUnreachableException` que AWS KMS retornam internamente quando não é possível estabelecer comunicação com o proxy externo do armazenamento de chaves.

Altas taxas de erros com nova tentativa podem indicar erros de redes, enquanto altas taxas de erros sem nova tentativa podem indicar um problema com a configuração do armazenamento de chaves externas. Por exemplo, um aumento `AuthenticationFailedExceptions` indica

uma discrepância entre as credenciais de autenticação configuradas AWS KMS e o proxy externo do armazenamento de chaves. Para visualizar a configuração do armazenamento de chaves externas, consulte [Visualizar repositórios de chaves externos](#). Para editar suas configurações de armazenamento de chaves externas, consulte [Editar propriedades do repositório de chaves externo](#).

As exceções AWS KMS recebidas do proxy externo do armazenamento de chaves são diferentes das exceções que AWS KMS retornam quando uma operação falha. AWS KMS as operações criptográficas retornam an `KMSInvalidStateException` para todas as falhas relacionadas à configuração externa ou ao estado da conexão do armazenamento de chaves externo. Para identificar o problema, use o texto da mensagem de erro que o acompanha.

A tabela a seguir mostra as exceções que podem aparecer no gráfico das 5 principais exceções e as exceções correspondentes que AWS KMS retornam para você.

Tipo de erro	Exceção exibida no grafo	Exceção que AWS KMS retornou para você
Sem nova tentativa	AccessDeniedException Para obter ajuda sobre a solução de problemas, consulte Problemas de autorização de proxy .	CustomKeyStoreInvalidStateException em resposta às operações <code>CreateKey</code> . KMSInvalidStateException em resposta às operações de criptografia.
Sem nova tentativa	AuthenticationFailedException Para obter ajuda sobre a solução de problemas, consulte Erros de credenciais de autenticação .	XksProxyIncorrectAuthenticationCredentialException em resposta às operações <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException em resposta às operações <code>CreateKey</code> .

Tipo de erro	Exceção exibida no grafo	Exceção que AWS KMS retornou para você
		KMSInvalidStateException em resposta às operações de criptografia.
Com nova tentativa	DependencyTimeoutException Para obter ajuda sobre a solução de problemas, consulte Erros de latência e de tempo limite .	XksProxyUriUnreachableException em resposta às operações <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException em resposta às operações <code>CreateKey</code> . KMSInvalidStateException em resposta às operações de criptografia.
Com nova tentativa	InternalException O proxy de armazenamento de chaves externas rejeitou a solicitação porque não consegue se comunicar com o gerenciador de chaves externas. Verifique se a configuração do proxy do armazenamento de chaves externas está correta e se o gerenciador de chaves externas está disponível.	XksProxyInvalidResponseException em resposta às operações <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException em resposta às operações <code>CreateKey</code> . KMSInvalidStateException em resposta às operações de criptografia.

Tipo de erro	Exceção exibida no grafo	Exceção que AWS KMS retornou para você
Sem nova tentativa	InvalidCiphertextException Para obter ajuda sobre a solução de problemas, consulte Erros de descrição de criptografia .	KMSInvalidStateException em resposta às operações de criptografia.
Sem nova tentativa	InvalidKeyUsageException Para obter ajuda sobre a solução de problemas, consulte Erros de operação de criptografia da chave externa .	XksKeyInvalidConfigurationException em resposta às operações <code>CreateKey</code> . KMSInvalidStateException em resposta às operações de criptografia.
Sem nova tentativa	InvalidStateException Para obter ajuda sobre a solução de problemas, consulte Erros de operação de criptografia da chave externa .	XksKeyInvalidConfigurationException em resposta às operações <code>CreateKey</code> . KMSInvalidStateException em resposta às operações de criptografia.

Tipo de erro	Exceção exibida no grafo	Exceção que AWS KMS retornou para você
Sem nova tentativa	InvalidUriPathException Para obter ajuda sobre a solução de problemas, consulte Erros gerais de configuração .	XksProxyInvalidConfigurationException em resposta às operações CreateCustomKeyStore e UpdateCustomKeyStore . CustomKeyStoreInvalidStateException em resposta às operações CreateKey . KMSInvalidStateException em resposta às operações de criptografia.
Sem nova tentativa	KeyNotFoundException Para obter ajuda sobre a solução de problemas, consulte Erros de chave externa .	XksKeyNotFoundException em resposta às operações CreateKey . KMSInvalidStateException em resposta às operações de criptografia.

Tipo de erro	Exceção exibida no grafo	Exceção que AWS KMS retornou para você
Com nova tentativa	ThrottlingException O proxy de armazenamento de chaves externas rejeitou a solicitação devido a uma taxa de solicitação muito alta. Reduza a frequência de suas chamadas usando chaves do KMS neste armazenamento de chaves externas.	XksProxyUriUnreachableException em resposta às operações <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException em resposta às operações <code>CreateKey</code> . KMSInvalidStateException em resposta às operações de criptografia.
Sem nova tentativa	UnsupportedOperationException Para obter ajuda sobre a solução de problemas, consulte Erros de operação de criptografia da chave externa.	XksKeyInvalidResponseException em resposta às operações <code>CreateKey</code> . KMSInvalidStateException em resposta às operações de criptografia.

Tipo de erro	Exceção exibida no grafo	Exceção que AWS KMS retornou para você
Sem nova tentativa	ValidationException Para obter ajuda sobre a solução de problemas, consulte Problemas de proxy .	XksProxyInvalidResponseException em resposta às operações <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException em resposta às operações <code>CreateKey</code> . KMSInvalidStateException em resposta às operações de criptografia.
Com nova tentativa	XksProxyUnreachableException Caso veja esse erro repetidamente, verifique se o proxy de armazenamento de chaves externas está ativo e conectado à rede e se o caminho do URI e o URI do endpoint ou o nome do serviço da VPC estão corretos no armazenamento de chaves externas.	XksProxyUriUnreachableException em resposta às operações <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException em resposta às operações <code>CreateKey</code> . KMSInvalidStateException em resposta às operações de criptografia.

O grafo das cinco principais exceções é derivado da métrica [XksProxyErrors](#). Quando você [visualiza um ponto de dados específico](#), o pop-up exibe o valor da dimensão `ExceptionName` junto com

o número de vezes que a exceção foi registrada no ponto de dados. Os cinco itens da lista são ordenados da exceção mais frequente para a menos frequente.

Recomendamos usar a [XksProxyErrors](#) métrica para criar um CloudWatch alarme que notifique você sobre possíveis problemas de configuração, alertando-o quando mais de cinco erros que não podem ser repetidos forem registrados em um período de um minuto. Para obter mais informações, consulte [Criar um alarme para erros não passíveis de nova tentativa](#).

Dias para o certificado expirar

O número de dias até o certificado TLS do endpoint do proxy de armazenamento de chaves externas (XksProxyUriEndpoint) expirar. Use esse grafo para monitorar a próxima expiração de seu certificado TLS.

Quando o certificado expira, AWS KMS não é possível se comunicar com o proxy externo do armazenamento de chaves. Todos os dados protegidos por chaves do KMS em seu armazenamento de chaves externas ficarão inacessíveis até você renovar o certificado.

O grafo de dias até a expiração do certificado é derivado da métrica [XksProxyCertificateDaysToExpire](#). É altamente recomendável usar essa métrica para criar um CloudWatch alarme que o notifique sobre o vencimento futuro. A expiração do certificado pode impedir que você acesse os recursos criptografados. Defina o alarme para dar tempo para a sua organização renovar o certificado antes que ele expire. Para obter mais informações, consulte [Criar um alarme para expiração de certificado](#).

Conectar e desconectar repositórios de chaves externos

Os novos armazenamentos de chaves externas não são conectados. Para criar e usar AWS KMS keys em seu armazenamento de chaves externo, você precisa conectar seu armazenamento de chaves externo ao [proxy de armazenamento de chaves externo](#). Você pode conectar e desconectar seu armazenamento de chaves externas a qualquer momento, e [visualizar seu estado de conexão](#).

Enquanto o armazenamento de chaves externo estiver desconectado, AWS KMS não poderá se comunicar com o proxy externo do armazenamento de chaves. Como resultado, você poderá visualizar e gerenciar o armazenamento de chaves externas e suas chaves do KMS existentes. No entanto, não é possível criar chaves do KMS em seu armazenamento de chaves externas nem usar suas chaves do KMS em operações de criptografia. Talvez seja necessário desconectar o armazenamento de chaves externas em algum momento, como ao editar as propriedades, mas planeje adequadamente. Desconectar o armazenamento de chaves pode interromper a operação dos AWS serviços que usam suas chaves KMS.

Não é necessário conectar seu armazenamento de chaves externas. Você pode deixar um armazenamento de chaves externas em estado desconectado indefinidamente e conectá-lo somente quando precisar usá-lo. No entanto, você pode desejar testar a conexão periodicamente para verificar se as configurações estão corretas e se ele pode ser conectado.

Quando você desconecta um armazenamento de chaves personalizado, as chaves do KMS do armazenamento de chaves tornam-se inutilizáveis imediatamente (sujeitas a consistência posterior). Porém, os recursos criptografados com [chaves de dados](#) protegidas pela chave do KMS não serão afetados até que a chave do KMS seja usada novamente, por exemplo, para descriptografar a chave de dados. Esse problema afeta os Serviços da AWS, pois muitos deles usam chaves de dados para proteger recursos. Para obter detalhes, consulte [Como as chaves do KMS inutilizáveis afetam as chaves de dados](#).

Note

Armazenamentos de chaves externas têm o estado DISCONNECTED somente quando nunca foram conectados ou quando você os desconecta explicitamente. Um estado CONNECTED não indica que o armazenamento de chaves externas ou seus componentes de apoio estejam operando com eficiência. Para obter informações sobre a performance dos componentes do armazenamento de chaves externas, consulte os grafos na seção Monitoring (Monitoramento) da página de detalhes de cada armazenamento de chaves externas. Para obter detalhes, consulte [Monitorar repositórios de chaves externos](#). Seu gerenciador de chaves externo pode fornecer métodos adicionais para interromper e reiniciar a comunicação entre o armazenamento de chaves AWS KMS externo e o proxy externo do armazenamento de chaves, ou entre o proxy externo do armazenamento de chaves e o gerenciador de chaves externo. Para obter detalhes, consulte a documentação do gerenciador de chaves externas.

Tópicos

- [Estado da conexão](#)
- [Conectar um repositório de chaves externo](#)
- [Desconectar um repositório de chaves externo](#)

Estado da conexão

A conexão e desconexão altera o estado da conexão do armazenamento de chaves personalizado. Os valores do estado da conexão são os mesmos para armazenamentos de AWS CloudHSM chaves e armazenamentos de chaves externos.

Para ver o estado da conexão do seu armazenamento de chaves personalizadas, use a [DescribeCustomKeyStores](#) operação ou o AWS KMS console. O estado da conexão é exibido em cada tabela de armazenamento de chaves personalizado, na seção General configuration (Configuração geral) da página de detalhes de cada armazenamento de chaves personalizado e na guia Cryptographic configuration (Configuração criptográfica) das chaves do KMS em um armazenamento de chaves personalizado. Para obter mais detalhes, consulte [Exibir uma loja de AWS CloudHSM chaves](#) e [Visualizar repositórios de chaves externos](#).

O armazenamento de chaves personalizado pode ter um dos estados de conexão a seguir:

- **CONNECTED:** o armazenamento de chaves personalizado está conectado a seu armazenamento de chaves de reserva. Você pode criar e usar chaves do KMS no armazenamento de chaves personalizado.

O armazenamento de chaves de apoio para um armazenamento de AWS CloudHSM chaves é seu AWS CloudHSM cluster associado. O armazenamento de chaves de reserva para um armazenamento de chaves externas é o proxy de armazenamento de chaves externas e o gerenciador de chaves externas ao qual oferece suporte.

Um estado CONNECTED significa que uma conexão foi bem-sucedida e que o armazenamento de chaves personalizado não foi desconectado intencionalmente. Isso não indica que a conexão está funcionando devidamente. Para obter informações sobre o status do AWS CloudHSM cluster associado ao seu armazenamento de AWS CloudHSM chaves, consulte [Obter CloudWatch métricas para AWS CloudHSM](#) no Guia do AWS CloudHSM usuário. Para obter informações sobre o status e a operação do armazenamento de chaves externas, consulte os grafos na seção Monitoring (Monitoramento) da página de detalhes de cada armazenamento de chaves externas. Para obter detalhes, consulte [Monitorar repositórios de chaves externos](#).

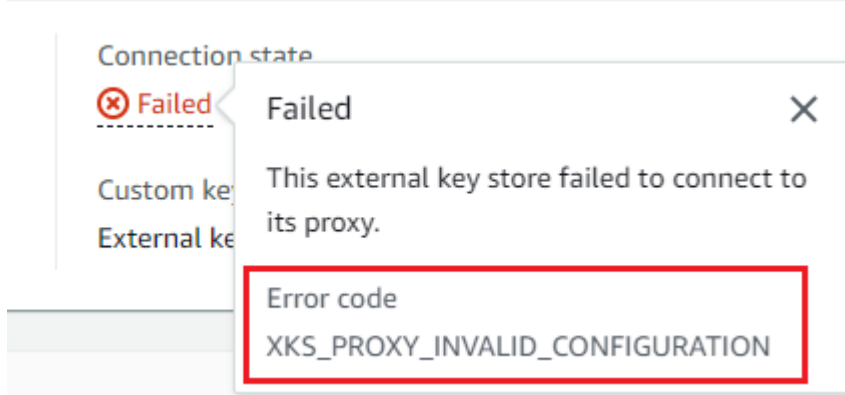
- **CONNECTING:** o processo de conexão de um armazenamento de chaves personalizado está em andamento. É um estado transitório.
- **DISCONNECTED:** o armazenamento de chaves personalizadas nunca foi conectado ao seu suporte ou foi desconectado intencionalmente usando o AWS KMS console ou a operação. [DisconnectCustomKeyStore](#)

- **DISCONNECTING**: o processo de desconexão de um armazenamento de chaves personalizado está em andamento. É um estado transitório.
- **FAILED**: falha na tentativa de conexão ao armazenamento de chaves personalizado. O `ConnectionErrorCode` na [DescribeCustomKeyStores](#) resposta indica o problema.

Para conectar um armazenamento de chaves personalizado, o estado de conexão deve ser **DISCONNECTED**. Se o estado da conexão for **FAILED**, use `ConnectionErrorCode` para identificar e resolver o problema. Desconecte o armazenamento de chaves personalizado antes de tentar conectá-lo novamente. Para obter ajuda com falhas de conexão, consulte [Erros de conexão do armazenamento de chaves externas](#). Para obter ajuda para responder a um código de erro de conexão, consulte [Códigos de erro de conexão para armazenamentos de chaves externas](#).

Para visualizar o código de erro de conexão:

- Na [DescribeCustomKeyStores](#) resposta, visualize o valor do `ConnectionErrorCode` elemento. Este elemento aparece na resposta de `DescribeCustomKeyStores` somente quando `ConnectionState` é **FAILED**.
- Para visualizar o código de erro de conexão no AWS KMS console, na página de detalhes do armazenamento externo de chaves e passe o mouse sobre o valor Falha.



Conectar um repositório de chaves externo

Quando seu armazenamento de chaves externas está conectado ao proxy de armazenamento de chaves externas, é possível [criar chaves do KMS no armazenamento de chaves externas](#) e usar chaves do KMS existentes em [operações de criptografia](#).

O processo que conecta um armazenamento de chaves externas ao proxy de armazenamento de chaves externas difere com base na conectividade do armazenamento de chaves externas.

- Quando você conecta um armazenamento de chaves externo com [conectividade de endpoint público](#), AWS KMS envia uma [GetHealthStatus solicitação](#) ao proxy externo do armazenamento de chaves para validar o [endpoint do URI do proxy](#), o [caminho do URI do proxy](#) e a [credencial de autenticação do proxy](#). Uma resposta bem-sucedida do proxy confirma que o [endpoint do URI do proxy](#) e o [caminho do URI do proxy](#) estão corretos e acessíveis e que o proxy autenticou a solicitação assinada com a [credencial de autenticação do proxy](#) para o armazenamento de chaves externas.
- Ao conectar um armazenamento de chaves externo com a [conectividade do serviço de endpoint VPC](#) ao proxy externo do armazenamento de chaves, AWS KMS faça o seguinte:
 - Confirma que o domínio do nome DNS privado especificado no [endpoint do URI do proxy](#) foi [verificado](#).
 - Cria um endpoint de interface de uma AWS KMS VPC para seu serviço de endpoint de VPC.
 - Cria uma zona hospedada privada para o nome DNS privado especificado no endpoint do URI do proxy
 - Envia uma [GetHealthStatussolicitação](#) para o proxy externo do armazenamento de chaves. Uma resposta bem-sucedida do proxy confirma que o [endpoint do URI do proxy](#) e o [caminho do URI do proxy](#) estão corretos e acessíveis e que o proxy autenticou a solicitação assinada com a [credencial de autenticação do proxy](#) para o armazenamento de chaves externas.

A operação de conexão inicia o processo de conexão do armazenamento de chaves personalizado, mas conectar um armazenamento de chaves externas ao proxy externo leva aproximadamente cinco minutos. Uma resposta bem-sucedida da operação de conexão não indica que o armazenamento de chaves externas esteja conectado. Para confirmar que a conexão foi bem-sucedida, use o AWS KMS console ou a [DescribeCustomKeyStores](#) operação para visualizar o [estado da conexão](#) externa do seu armazenamento de chaves.

Quando o estado da conexão é `FAILED`, um código de erro de conexão é exibido no AWS KMS console e adicionado à `DescribeCustomKeyStore` resposta. Para obter ajuda na interpretação dos códigos de erro de conexão, consulte [Códigos de erro de conexão para armazenamentos de chaves externas](#).

Conectar e reconectar ao repositório de chaves externo

Você pode conectar ou reconectar seu armazenamento de chaves externo no AWS KMS console ou usando a [ConnectCustomKeyStore](#) operação.

Usando o AWS KMS console

Você pode usar o AWS KMS console para conectar um armazenamento de chaves externo ao seu proxy de armazenamento de chaves externo.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Armazenamentos de chaves personalizados), External key stores (Armazenamentos de chaves externas).
4. Escolha a linha do armazenamento de chaves externas que deseja conectar.

Se o [estado de conexão](#) do armazenamento de chaves externas for FAILED, você deverá [desconectar o armazenamento de chaves externas](#) antes de conectá-lo.

5. No menu Key store actions (Ações do armazenamento de chaves), escolha Connect (Conectar).

O processo de conexão normalmente leva cerca de cinco minutos para ser concluído. Quando a operação é concluída, o [estado da conexão](#) é alterado para CONNECTED.

Se o estado da conexão for Failed, passe o mouse sobre o estado da conexão para ver o código de erro da conexão, que explica a causa do erro. Para obter ajuda para responder a um código de erro de conexão, consulte [Códigos de erro de conexão para armazenamentos de chaves externas](#). Para conectar um armazenamento de chaves externas com o estado de conexão Failed, você deve primeiro [desconectar o armazenamento de chaves personalizado](#).

Usando a AWS KMS API

Para conectar um armazenamento de chaves externo desconectado, use a [ConnectCustomKeyStore](#) operação.

Antes de se conectar, o [estado da conexão](#) do armazenamento de chaves externas deve ser DISCONNECTED. Se o estado da conexão atual for FAILED, [desconecte o armazenamento de chaves externas](#) e conecte-o novamente.

O processo de conexão leva cerca de cinco minutos para ser concluído. A menos que ocorra uma falha rapidamente, ConnectCustomKeyStore retornará uma resposta HTTP 200 e um objeto JSON sem propriedades. No entanto, essa resposta inicial não indica que a conexão foi bem-sucedida. Para determinar se o armazenamento de chaves externo está conectado, veja o estado da conexão na [DescribeCustomKeyStores](#) resposta.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Para identificar o armazenamento de chaves externas, use o ID do armazenamento de chaves personalizado. Você pode encontrar o ID na página Armazenamentos de chaves personalizadas no console ou usando a [DescribeCustomKeyStores](#) operação. Antes de executar esse exemplo, substitua o ID de exemplo por um válido.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

A operação `ConnectCustomKeyStore` não retorna `ConnectionState` na resposta. Para verificar se o armazenamento de chaves externo está conectado, use a [DescribeCustomKeyStores](#) operação. Por padrão, essa operação retorna todos os armazenamentos de chaves personalizados em sua conta e região. No entanto, você pode usar o parâmetro `CustomKeyStoreId` ou `CustomKeyStoreName` (mas não ambos) para limitar a resposta para determinados armazenamentos de chaves personalizados. Um valor `ConnectionState` de `CONNECTED` indica que o armazenamento de chaves externas está conectado ao proxy de armazenamento de chaves externas.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
  "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

Se o valor `ConnectionState` na resposta de `DescribeCustomKeyStores` for `FAILED`, o elemento `ConnectionErrorCode` indica o motivo da falha.

No exemplo a seguir, o `XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND` valor do `ConnectionErrorCode` indica que não é AWS KMS possível encontrar o serviço de endpoint da VPC que ele usa para se comunicar com o proxy externo do armazenamento de chaves. Verifique se `XksProxyVpcEndpointServiceName` está correto, se o principal de AWS KMS serviço é um principal permitido no serviço de endpoint da Amazon VPC e se o serviço de endpoint de VPC não exige a aceitação de solicitações de conexão. Para obter ajuda para responder a um código de erro de conexão, consulte [Códigos de erro de conexão para armazenamentos de chaves externas](#).

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

Desconectar um repositório de chaves externo

Quando você desconecta um armazenamento de chaves externas com [conectividade do serviço de endpoint da VPC](#) do proxy de armazenamento de chaves externas, o AWS KMS exclui o endpoint de interface para o serviço de endpoint da VPC e remove a infraestrutura de rede que ele criou para oferecer suporte à conexão. Nenhum processo equivalente é necessário para armazenamentos de chaves externas com conectividade de endpoint público. Essa ação não afeta o serviço de endpoint da VPC ou seus componentes de apoio nem afeta o proxy de armazenamento de chaves externas ou componentes externos.

Enquanto o armazenamento de chaves externo estiver desconectado, AWS KMS não envia nenhuma solicitação para o proxy externo do armazenamento de chaves. O estado da conexão do armazenamento de chaves externas é DISCONNECTED. As chaves do KMS no armazenamento de chaves externas desconectado estão no [estado de chave UNAVAILABLE](#) (a menos que estejam [pendentes de exclusão](#)), isto é, não podem ser usadas em operações de criptografia. Porém, você ainda poderá visualizar e gerenciar o armazenamento de chaves externas e suas chaves do KMS existentes.

O estado desconectado foi criado para ser temporário e reversível. Você pode reconectar o armazenamento de chaves externas a qualquer momento. Normalmente, não é necessária nenhuma reconfiguração. Porém, se alguma propriedade do proxy de armazenamento de chaves externas associado tiver sido alterada enquanto ele estava desconectado, como a alternância de sua [credencial de autenticação de proxy](#), você deverá [editar as configurações do armazenamento de chaves externas](#) antes de reconectar.

 Note

Enquanto um armazenamento de chaves personalizado estiver desconectado, todas as tentativas de criar chaves do KMS nele ou de usar chaves do KMS existentes em operações de criptografia falharão. Essa ação pode impedir que os usuários armazenem e acessem dados sigilosos.

Para avaliar melhor o efeito de desconectar o armazenamento de chaves externas, identifique as chaves do KMS no armazenamento de chaves externas e [determine seu uso anterior](#).

Você pode desconectar o armazenamento de chaves externas por motivos como estes:

- Para editar suas propriedades. Você pode editar o nome do armazenamento de chaves personalizado, o caminho do URI do proxy e a credencial de autenticação do proxy enquanto o armazenamento de chaves externas está conectado. No entanto, para editar o tipo de conectividade do proxy, o endpoint do URI do proxy ou o nome do serviço de endpoint da VPC, primeiro é necessário desconectar o armazenamento de chaves externas. Para obter detalhes, consulte [Editar propriedades do repositório de chaves externo](#).
- Para interromper toda a comunicação entre AWS KMS e o proxy externo do armazenamento de chaves. Você também pode interromper a comunicação entre AWS KMS e seu proxy desativando seu endpoint ou serviço de endpoint VPC. Além disso, seu proxy de armazenamento de chaves externo ou software de gerenciamento de chaves pode fornecer mecanismos adicionais para

AWS KMS impedir a comunicação com o proxy ou impedir que o proxy acesse seu gerenciador de chaves externo.

- Para desabilitar todas as chaves do KMS no armazenamento de chaves externas. Você pode [desativar e reativar as chaves KMS](#) em um armazenamento de chaves externo usando o AWS KMS console ou a [DisableKey](#) operação. Essas operações são concluídas rapidamente (sujeitas a consistência posterior), mas atuam em uma chave do KMS de cada vez. A desconexão do armazenamento de chaves externas altera o estado da chave de todas as chaves do KMS no armazenamento de chaves externas para Unavailable, impedindo que elas sejam usadas em operações de criptografia.
- Para reparar uma falha na tentativa de conexão. Se ocorrer falha em uma tentativa de conectar-se a um armazenamento de chaves externas (o estado da conexão do armazenamento de chaves personalizado apresentado é FAILED), você deve desconectar o armazenamento de chaves externas antes de tentar se conectar novamente.

Desconectar seu repositório de chaves externo

Você pode desconectar o armazenamento externo de chaves no AWS KMS console ou usando a [DisconnectCustomKeyStore](#) operação.

Usando o AWS KMS console

Você pode usar o AWS KMS console para conectar um armazenamento de chaves externo ao seu proxy de armazenamento de chaves externo. Esse processo leva cerca de cinco minutos para ser concluído.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Armazenamentos de chaves personalizados), External key stores (Armazenamentos de chaves externas).
4. Escolha a linha do armazenamento de chaves externas que deseja desconectar.
5. No menu Key store actions (Ações do armazenamento de chaves), escolha Disconnect (Desconectar).

Quando a operação é concluída, o estado de conexão é alterado de DISCONNECTING (DESCONECTANDO) para DISCONNECTED (DESCONECTADO). Se ocorrer falha na operação,

será exibida uma mensagem de erro descrevendo o problema e fornecendo ajuda para corrigi-lo. Se precisar de ajuda adicional, consulte [Erros de conexão do armazenamento de chaves externas](#).

Usando a AWS KMS API

Para desconectar um armazenamento de chaves externo conectado, use a [DisconnectCustomKeyStore](#) operação. Se a operação for bem-sucedida, AWS KMS retornará uma resposta HTTP 200 e um objeto JSON sem propriedades. O processo leva cerca de cinco minutos para ser concluído. Para encontrar o estado da conexão do armazenamento de chaves externo, use a [DescribeCustomKeyStores](#) operação.

Os exemplos nesta seção usam a [AWS Command Line Interface \(AWS CLI\)](#), mas você pode usar qualquer linguagem de programação compatível.

Este exemplo desconecta um armazenamento de chaves externas da conectividade do serviço de endpoint da VPC. Antes de executar este exemplo, substitua o ID de exemplo do armazenamento de chaves personalizado por um válido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Para verificar se o armazenamento de chaves externo está desconectado, use a [DescribeCustomKeyStores](#) operação. Por padrão, essa operação retorna todos os armazenamentos de chaves personalizados em sua conta e região. No entanto, você pode usar o parâmetro CustomKeyId e CustomKeyName (mas não ambos) para limitar a resposta para determinados armazenamentos de chaves personalizados. O valor ConnectionState de DISCONNECTED indica que esse exemplo de armazenamento de chaves externas não está mais conectado ao proxy de armazenamento de chaves externas.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-9876543210fedcba9",
      "CustomKeyName": "ExampleXksVpc",
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
```



```
    "UriPath": "/example/prefix/kms/xks/v1",
    "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
  }
}
```

Excluir um repositório de chaves externo

Quando você exclui um armazenamento de chaves externo, AWS KMS exclui todos os metadados sobre o armazenamento de chaves externo AWS KMS, incluindo informações sobre seu proxy de armazenamento de chaves externo. Essa operação não afeta o [proxy externo do armazenamento de chaves](#), o [gerenciador de chaves externo](#), [as chaves externas](#) ou quaisquer AWS recursos que você criou para dar suporte ao armazenamento de chaves externo, como um Amazon VPC ou um serviço de endpoint de VPC.

Antes de excluir um armazenamento de chaves externas, você deve [excluir todas as chaves do KMS](#) do armazenamento de chaves e [desconectar o armazenamento de chaves](#) do proxy de armazenamento de chaves externas. Senão, as tentativas de excluir o armazenamento de chaves falharão.

A exclusão de um armazenamento de chaves externas é irreversível, mas você pode criar um novo armazenamento de chaves externas e associá-lo ao mesmo proxy de armazenamento de chaves externas e ao mesmo gerenciador de chaves externas. No entanto, você não pode recriar as chaves KMS de criptografia simétrica no armazenamento de chaves externo, mesmo que tenha acesso ao mesmo material de chave externa. AWS KMS inclui metadados no texto cifrado simétrico exclusivo de cada chave KMS. Esse recurso de segurança garante que somente a chave do KMS que criptografou os dados poderá descriptografá-los.

Em vez de excluir o armazenamento de chaves externas, considere desconectá-lo. Enquanto um armazenamento de chaves externo estiver desconectado, você pode gerenciar o armazenamento de chaves externo e seu AWS KMS keys , mas não pode criar ou usar chaves KMS no armazenamento de chaves externo. Você pode reconectar o armazenamento de chaves externas a qualquer momento e continuar usando suas chaves do KMS para criptografar e descriptografar dados. Não há custo para um proxy de armazenamento de chaves externas desconectado ou chaves do KMS indisponíveis.

Você pode excluir seu armazenamento externo de chaves no AWS KMS console ou usando a [DeleteCustomKeyStore](#) operação.

Usando o AWS KMS console

Você pode usar o AWS KMS console para excluir um armazenamento de chaves externo.

1. Faça login no console Console de gerenciamento da AWS e abra o AWS Key Management Service (AWS KMS) em <https://console.aws.amazon.com/kms>.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. No painel de navegação, selecione Custom key stores (Armazenamentos de chaves personalizados), External key stores (Armazenamentos de chaves externas).
4. Encontre a linha que representa o armazenamento de chaves externas que você deseja excluir. Se o Connection state (Estado de conexão) do armazenamento de chaves externas não for DISCONNECTED, você deverá [desconectar o armazenamento de chaves externas](#) antes de excluí-lo.
5. No menu Key store actions (Ações de armazenamento de chaves), escolha Delete (Excluir).

Quando a operação for concluída, uma mensagem de êxito será exibida, e o armazenamento de chaves externas não será mais exibido na lista de armazenamentos de chaves. Se a operação não for bem-sucedida, será exibida uma mensagem de erro descrevendo o problema e fornecendo ajuda para corrigi-lo. Se precisar de ajuda adicional, consulte [Solução de problemas de armazenamentos de chaves externas](#).

Usando a AWS KMS API

Para excluir um armazenamento de chaves externo, use a [DeleteCustomKeyStore](#) operação. Se a operação for bem-sucedida, AWS KMS retornará uma resposta HTTP 200 e um objeto JSON sem propriedades.

Para começar, desconecte o armazenamento de chaves externas. Antes de executar um comando como esse, substitua o ID de exemplo do armazenamento de chaves personalizado por um válido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Depois que o armazenamento de chaves externo for desconectado, você poderá usar a [DeleteCustomKeyStore](#) operação para excluí-lo.

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Para confirmar que o armazenamento de chaves externo foi excluído, use a [DescribeCustomKeyStores](#) operação.

```
$ aws kms describe-custom-key-stores

{
  "CustomKeyStores": []
}
```

Se você especificar um nome ou ID de armazenamento de chaves personalizados que não exista mais, AWS KMS retornará uma `CustomKeyStoreNotFoundException` exceção.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
```

```
An error occurred (CustomKeyStoreNotFoundException) when calling the
DescribeCustomKeyStore operation:
```

Solução de problemas de armazenamentos de chaves externas

A resolução da maioria dos problemas com armazenamentos de chaves externos é indicada pela mensagem de erro AWS KMS exibida com cada exceção ou pelo [código de erro de conexão](#) que AWS KMS retorna quando uma tentativa de [conectar o armazenamento de chaves externo](#) ao proxy externo do armazenamento de chaves falha. Porém, alguns problemas são um pouco mais complexos.

Ao diagnosticar um problema com um armazenamento de chaves externas, primeiro localize a causa. Isso reduzirá a variação de soluções e tornará a solução de problemas mais eficiente.

- AWS KMS — O problema pode estar dentro AWS KMS, como um valor incorreto na [configuração do armazenamento de chaves externo](#).
- Externo — O problema pode ter origem externa AWS KMS, incluindo problemas com a configuração ou operação do proxy externo do armazenamento de chaves, do gerenciador de chaves externo, das chaves externas ou do serviço de endpoint VPC.
- Rede: pode ser um problema de conectividade ou de rede, como um problema com o endpoint do proxy, a porta, a pilha IP ou o nome ou domínio de DNS privado.

 Note

Quando as operações de gerenciamento em armazenamentos de chaves externas falham, elas geram várias exceções diferentes. Mas as operações AWS KMS criptográficas retornam `KMSInvalidStateException` para todas as falhas relacionadas à configuração externa ou ao estado da conexão do armazenamento de chaves externo. Para identificar o problema, use o texto da mensagem de erro que o acompanha.

A [ConnectCustomKeyStore](#) operação é bem-sucedida rapidamente antes que o processo de conexão seja concluído. Para determinar se o processo de conexão foi bem-sucedido, veja o [estado da conexão](#) do armazenamento de chaves externas. Se o processo de conexão falhar, o AWS KMS retornará um [código de erro de conexão](#) que explica a causa e sugere uma solução.

Tópicos

- [Ferramentas de solução de problemas de armazenamentos de chaves externas](#)
- [Erros de configuração](#)
- [Erros de conexão do armazenamento de chaves externas](#)
- [Erros de latência e de tempo limite](#)
- [Erros de credenciais de autenticação](#)
- [Erros de estado da chave](#)
- [Erros de descriptografia](#)
- [Erros de chave externa](#)
- [Problemas de proxy](#)
- [Problemas de autorização de proxy](#)

Ferramentas de solução de problemas de armazenamentos de chaves externas

AWS KMS fornece várias ferramentas para ajudá-lo a identificar e resolver problemas com seu armazenamento de chaves externo e suas chaves. Use essas ferramentas em conjunto com as ferramentas fornecidas com o proxy de armazenamento de chaves externas e o gerenciador de chaves externas.

Note

Seu proxy de armazenamento de chaves externas e seu gerenciador de chaves externas podem fornecer métodos mais fáceis de criar e manter seu armazenamento de chaves externas e as respectivas chaves do KMS. Para obter detalhes, consulte a documentação de suas ferramentas externas.

AWS KMS exceções e mensagens de erro

AWS KMS fornece uma mensagem de erro detalhada sobre qualquer problema encontrado. Você pode encontrar informações adicionais sobre AWS KMS exceções na [Referência da AWS Key Management Service API](#) e. AWS SDKs Mesmo se você estiver usando o AWS KMS console, talvez essas referências sejam úteis. Por exemplo, consulte a lista de [erros](#) da operação `CreateCustomKeyStores`.

Para otimizar o desempenho do proxy do repositório de chaves externo, o AWS KMS retorna exceções com base na confiabilidade do proxy em um período determinado de agregação de 5 minutos. No caso de um erro interno do servidor: 500, serviço indisponível: 503 ou tempo limite de conexão, um proxy com alta confiabilidade retorna `KMSInternalException` e aciona uma nova tentativa automática para garantir que as solicitações eventualmente obtenham sucesso. No entanto, um proxy com baixa confiabilidade retorna `KMSInvalidStateException`. Para obter mais informações, consulte [Monitoring an external key store](#).

Se o problema surgir em um AWS serviço diferente, como quando você usa uma chave KMS em seu armazenamento de chaves externo para proteger um recurso em outro AWS serviço, o AWS serviço pode fornecer informações adicionais para ajudá-lo a identificar o problema. Se o AWS serviço não fornecer a mensagem, você poderá ver a mensagem de erro nos [CloudTrail registros que registram](#) o uso da sua chave KMS.

[CloudTrail logs](#)

Cada operação AWS KMS da API, incluindo ações no AWS KMS console, é registrada em AWS CloudTrail registros. AWS KMS registra uma entrada de registro para operações bem-sucedidas e malsucedidas. Para operações com falha, a entrada de log inclui o nome da exceção do AWS KMS (`errorCode`) e a mensagem de erro (`errorMessage`). Use essas informações para ajudar a identificar e resolver o erro. Para ver um exemplo, consulte [Descriptografar falhas com uma chave do KMS em um armazenamento de chaves externas](#).

A entrada de log também inclui o ID da solicitação. Se a solicitação atingiu seu proxy de armazenamento de chaves externas, você poderá usar o ID da solicitação na entrada de log para encontrar a solicitação correspondente em seus logs de proxy, caso o proxy os forneça.

CloudWatch métricas

AWS KMS registra CloudWatch métricas detalhadas da Amazon sobre a operação e o desempenho do seu armazenamento de chaves externo, incluindo latência, limitação, erros de proxy, status do gerenciador de chaves externo, o número de dias até que seu certificado TLS expire e a idade relatada de suas credenciais de autenticação de proxy. Você pode usar essas métricas para desenvolver modelos de dados para a operação do seu armazenamento externo de chaves e CloudWatch alarmes que alertem você sobre problemas iminentes antes que eles ocorram.

Important

AWS KMS recomenda que você crie CloudWatch alarmes para monitorar as métricas externas do armazenamento de chaves. Esses alarmes alertarão sobre os primeiros sinais de problemas antes que eles se desenvolvam.

Grafos de monitoramento

AWS KMS exibe gráficos das CloudWatch métricas do armazenamento de chaves externo na página de detalhes de cada armazenamento de chaves externo no AWS KMS console. Você pode usar os dados nos gráficos para ajudar a localizar a origem dos erros, detectar problemas iminentes, estabelecer linhas de base e refinar seus limites de alarme. CloudWatch Para obter detalhes sobre como interpretar os grafos de monitoramento e usar seus dados, consulte [Monitorar repositórios de chaves externos](#).

Exibições de armazenamentos de chaves externas e chaves do KMS

AWS KMS exibe informações detalhadas sobre seus armazenamentos de chaves externos e as chaves KMS no armazenamento de chaves externo no AWS KMS console e na resposta às [DescribeCustomKeyStoresDescribeKey](#) operações e. Essas exibições incluem campos especiais para armazenamentos de chaves externas e chaves do KMS com informações que você pode usar para solucionar problemas, como o [estado da conexão](#) do armazenamento de chaves externas e o ID da chave externa associada à chave do KMS. Para obter detalhes, consulte [Visualizar repositórios de chaves externos](#).

Cliente de teste do proxy XKS

AWS KMS fornece um cliente de teste de código aberto que verifica se o proxy externo do armazenamento de chaves está em conformidade com a especificação da [API AWS KMS External Key Store Proxy](#). Você pode usar esse cliente de teste para identificar e resolver problemas com seu proxy de armazenamento de chaves externas.

Erros de configuração

Ao criar um armazenamento de chaves externas, você especifica valores de propriedade que abrangem a configuração do armazenamento de chaves externas, como a [credencial de autenticação do proxy](#), o [endpoint do URI do proxy](#), o [caminho do URI do proxy](#) e o [nome do serviço de endpoint da VPC](#). Quando AWS KMS detecta um erro no valor de uma propriedade, a operação falha e retorna um erro que indica o valor defeituoso.

Muitos problemas de configuração podem ser resolvidos ao corrigir o valor incorreto. Você pode corrigir um caminho do URI do proxy inválido ou uma credencial de autenticação do proxy sem desconectar o armazenamento de chaves externas. Para obter definições desses valores, inclusive requisitos de exclusividade, consulte [Organizar os pré-requisitos](#). Para obter instruções sobre como atualizar esses valores, consulte [Editar propriedades do repositório de chaves externo](#).

Para evitar erros com o caminho do URI do proxy e os valores da credencial de autenticação do proxy, ao criar ou atualizar o armazenamento de chaves externas, carregue um [arquivo de configuração de proxy](#) para o console do AWS KMS. É um arquivo baseado em JSON com caminho do URI do proxy e valores de credencial de autenticação de proxy que é fornecido pelo proxy de armazenamento de chaves externas ou pelo gerenciador de chaves externas. Você não pode usar um arquivo de configuração de proxy com operações de AWS KMS API, mas pode usar os valores no arquivo para ajudá-lo a fornecer valores de parâmetros para suas solicitações de API que correspondam aos valores em seu proxy.

Erros gerais de configuração

Exceções: CustomKeyStoreInvalidStateException (CreateKey),
KMSInvalidStateException (operações de criptografia),
XksProxyInvalidConfigurationException (operações de gerenciamento, exceto para CreateKey)

Códigos de erro de conexão: XKS_PROXY_INVALID_CONFIGURATION,
XKS_PROXY_INVALID_TLS_CONFIGURATION

Para armazenamentos de chaves externos com [conectividade de endpoint público](#), AWS KMS testa os valores da propriedade ao criar e atualizar o armazenamento de chaves externo. Para armazenamentos de chaves externas com [conectividade de serviço de endpoint da VPC](#), o AWS KMS testa os valores das propriedades ao criar e atualizar o armazenamento de chaves externas.

 Note

A operação `ConnectCustomKeyStore`, que é assíncrona, pode ser bem-sucedida mesmo que a tentativa de conectar o armazenamento de chaves externas ao proxy de armazenamento de chaves externas falhe. Nesse caso, não há exceção, mas o estado da conexão do armazenamento de chaves externas é `Failed`, e um código de erro de conexão explica a mensagem de erro. Para obter mais informações, consulte [Erros de conexão do armazenamento de chaves externas](#).

Se AWS KMS detectar um erro no valor de uma propriedade, a operação falhará e retornará `XksProxyInvalidConfigurationException` com uma das seguintes mensagens de erro.

O proxy de armazenamento de chaves externas rejeitou a solicitação devido a um caminho de URI inválido. Verifique o caminho do URI para o armazenamento de chaves externas e atualize, se necessário.

- O [caminho do URI do proxy](#) é o caminho base para AWS KMS solicitações ao proxy APIs. Se o caminho estiver incorreto, todas as solicitações ao proxy falharão. Para [visualizar o caminho atual do URI do proxy](#) para o armazenamento de chaves externas, use o console do AWS KMS ou a operação `DescribeCustomKeyStores`. Para encontrar o caminho correto do URI do proxy, consulte a documentação do proxy de armazenamento de chaves externas. Para obter ajuda na correção do valor do caminho do URI do proxy, consulte [Editar propriedades do repositório de chaves externo](#).
- O caminho do URI do proxy para o proxy de armazenamento de chaves externas pode ser alterado com as atualizações no proxy de armazenamento de chaves externas ou no gerenciador de chaves externas. Para obter informações sobre essas alterações, consulte a documentação do proxy de armazenamento de chaves externas ou do gerenciador de chaves externas.

XKS_PROXY_INVALID_TLS_CONFIGURATION

O AWS KMS não consegue estabelecer uma conexão TLS com o proxy de armazenamento de chaves externas. Verifique a configuração do TLS, inclusive o certificado.

- Todos os proxies de armazenamento de chaves externas precisam de um certificado TLS. O certificado TLS deve ser emitido por uma autoridade de certificação (CA) pública com suporte para armazenamentos de chaves externas. Para obter uma lista dos compatíveis CAs, consulte [Autoridades de certificação confiáveis](#) na especificação da API AWS KMS External Key Store Proxy.
- Para conectividade de endpoints públicos, o nome comum (CN) da entidade no certificado TLS deve corresponder ao nome do domínio no [endpoint do URI do proxy](#) do armazenamento de chaves externas. Por exemplo, se o endpoint público for `https://myproxy.xks.example.com`, o TLS, o CN no certificado TLS deverá ser `myproxy.xks.example.com` ou `*.xks.example.com`.
- Para conectividade de serviços de endpoint da VPC, o nome comum (CN) do assunto no certificado TLS deve corresponder ao nome DNS privado do [serviço de endpoint da VPC](#). Por exemplo, se o nome DNS privado for `myproxy-private.xks.example.com`, o CN no certificado TLS deverá ser `myproxy-private.xks.example.com` ou `*.xks.example.com`.
- O certificado TLS não pode ter expirado. Para obter o prazo de validade de um certificado TLS, use ferramentas SSL, como o [OpenSSL](#). Para monitorar a data de expiração de um certificado TLS associado a um armazenamento de chaves externo, use a [XksProxyCertificateDaysToExpire](#) CloudWatch métrica. O número de dias até a data de expiração da certificação TLS também aparece na [seção Monitoramento](#) do AWS KMS console.
- Se você estiver usando [conectividade de endpoint público](#), use ferramentas de teste SSL para testar a configuração de SSL. Os erros de conexão TLS podem ser resultado do encadeamento incorreto de certificados.

Erros de configuração de conectividade do serviço de endpoint da VPC

Exceções: `XksProxyVpcEndpointServiceNotFoundException`,
`XksProxyVpcEndpointServiceInvalidConfigurationException`

Além dos problemas gerais de conectividade, você pode encontrar os seguintes problemas ao criar, conectar ou atualizar um armazenamento de chaves externo com a conectividade do serviço de endpoint VPC. AWS KMS testa os valores das propriedades de um armazenamento de chaves externo com a conectividade do serviço de endpoint VPC ao [criar](#), [conectar](#) e [atualizar](#) o armazenamento de chaves externo. Quando as operações de gerenciamento falham devido a erros de configuração, elas geram as seguintes exceções:

XksProxyVpcEndpointServiceNotFoundException

A causa pode ser uma das seguintes:

- Um nome de serviço de endpoint da VPC incorreto. Verifique se o nome do serviço de endpoint da VPC para o armazenamento de chaves externas está correto e corresponde ao valor do endpoint do URI do proxy para o armazenamento de chaves externas. Para encontrar o nome do serviço do VPC endpoint, use o console Amazon [VPC](#) ou a operação. [DescribeVpcEndpointServices](#) Para encontrar o nome do serviço do VPC endpoint e o endpoint do URI do proxy de um armazenamento de chaves externo existente, use o AWS KMS console ou a operação. [DescribeCustomKeyStores](#) Para obter detalhes, consulte [Visualizar repositórios de chaves externos](#).
- O serviço de endpoint da VPC pode estar em um local Região da AWS diferente do armazenamento de chaves externo. Verifique se o serviço de endpoint da VPC e o armazenamento de chaves externas estão na mesma região. (O nome externo do nome da região, como, faz parte do nome do serviço do VPC endpointus-east-1, como com.amazonaws.vpce.us-east-1.vpce-svc-example.) Para obter uma lista dos requisitos do serviço de endpoint da VPC para um armazenamento de chaves externas, consulte [Serviço de VPC endpoint](#). Você não pode migrar um serviço de endpoint da VPC ou um armazenamento de chaves externas para uma região diferente. Contudo, você pode criar um novo armazenamento de chaves externas na mesma região do serviço de endpoint da VPC. Para obter mais detalhes, consulte [Configurar a conectividade do serviço de endpoint da VPC](#) e [Criar um repositório de chaves externo](#).
- AWS KMS não é um principal permitido para o serviço de endpoint da VPC. A lista de entidades principais autorizadas para o serviço de endpoint da VPC deve incluir o valor de `cks.kms.<region>.amazonaws.com`, como `cks.kms.eu-west-3.amazonaws.com`. Para obter instruções sobre como adicionar esse valor, consulte [Manage permissions](#) (Gerenciar permissões) no Guia do AWS PrivateLink.

XksProxyVpcEndpointServiceInvalidConfigurationException

Esse erro ocorre quando o serviço de endpoint da VPC não atende a um dos seguintes requisitos:

- A VPC requer pelo menos duas sub-redes privadas, cada uma em uma zona de disponibilidade diferente. Para obter ajuda para adicionar uma sub-rede à sua VPC, consulte [Crie uma sub-rede na VPC](#) no Guia do usuário da Amazon VPC.
- Seu [tipo de serviço de endpoint da VPC](#) deve usar um balanceador de carga de rede, não um balanceador de carga de gateway.
- A aceitação não deve ser obrigatória para o serviço de endpoint da VPC (Acceptance required [Aceitação obrigatória] deve ser false). Se a aceitação manual de cada solicitação de conexão for necessária, AWS KMS não será possível usar o serviço de VPC endpoint para se conectar ao proxy externo do armazenamento de chaves. Para obter detalhes, consulte [Accept or reject connection requests](#) (Aceitar ou rejeitar solicitações de conexão) no Guia do AWS PrivateLink .
- O serviço de endpoint da VPC deve ter um nome DNS privado que seja o subdomínio de um domínio público. Por exemplo, se o nome DNS privado for `https://myproxy-private.xks.example.com`, os domínios `xks.example.com` ou `example.com` devem ter um servidor DNS público. Para visualizar ou alterar o nome DNS privado do serviço de endpoint da VPC, consulte [Manage DNS names for VPC endpoint services](#) (Gerenciar nomes de DNS para serviços de endpoint da VPC) no Guia do AWS PrivateLink .
- O status da verificação do domínio do domínio de seu nome DNS privado deverá ser `verified`. Para visualizar e atualizar o status de verificação do domínio do nome DNS privado, consulte [Etapa 5: verificar o domínio do seu nome de DNS privado](#). Pode levar alguns minutos para que o status de verificação atualizado seja exibido depois que você adicionar o registro de texto obrigatório.

 Note

O domínio DNS privado só poderá ser verificado se for o subdomínio de um domínio público. Caso contrário, o status de verificação do domínio DNS privado não será alterado, mesmo depois de adicionar o registro TXT obrigatório.

- Certifique-se de que qualquer firewall entre AWS KMS e o proxy externo do armazenamento de chaves permita o tráfego de e para a porta 443 no proxy. AWS KMS se comunica na porta 443. IPv4 Esse valor não é configurável.
- O nome DNS privado do serviço de endpoint da VPC deve corresponder ao valor do [endpoint do URI do proxy](#) para o armazenamento de chaves externas. Para um armazenamento de chaves externas com conectividade de serviço de endpoint da VPC, o endpoint do URI do proxy deve ser `https://` seguido pelo nome DNS privado do serviço de endpoint da VPC. Para visualizar o valor

do endpoint do URI do proxy, consulte [Visualizar repositórios de chaves externos](#). Para alterar o valor do endpoint do URI do proxy, consulte [Editar propriedades do repositório de chaves externo](#).

Erros de conexão do armazenamento de chaves externas

O [processo de conexão de um armazenamento de chaves externas](#) ao proxy de armazenamento de chaves externas leva cerca de cinco minutos para ser concluído. A menos que se antecipe à falha, a operação `ConnectCustomKeyStore` retornará uma resposta HTTP 200 e um objeto JSON sem propriedades. No entanto, essa resposta inicial não indica que a conexão foi bem-sucedida. Para determinar se o armazenamento de chaves externas está conectado, consulte o [estado da conexão](#). Se a conexão falhar, o estado da conexão do armazenamento de chaves externo mudará para `FAILED` e AWS KMS retornará um [código de erro de conexão](#) que explica a causa da falha.

Note

Quando o estado de conexão de um armazenamento de chaves personalizado é `FAILED`, você deve desconectar o armazenamento de chaves personalizado antes de tentar reconectá-lo. Não é possível conectar um armazenamento de chaves personalizado a um status de conexão `FAILED`.

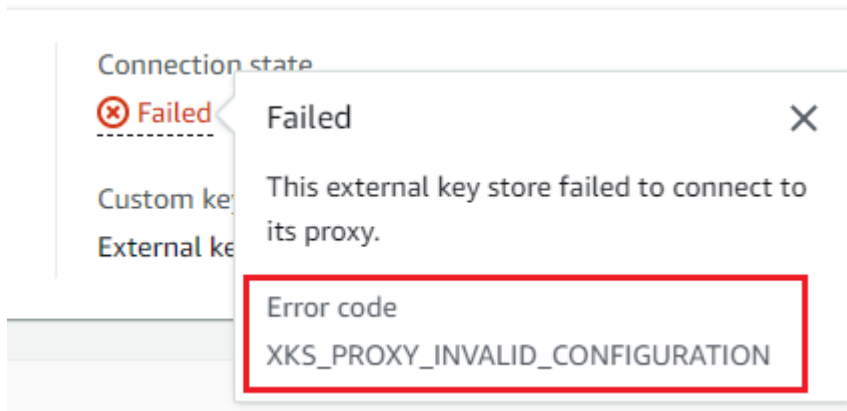
Para visualizar o estado da conexão de um armazenamento de chaves externas:

- Na [DescribeCustomKeyStores](#) resposta, visualize o valor do `ConnectionState` elemento.
- No AWS KMS console, o estado da conexão aparece na tabela de armazenamento de chaves externo. Além disso, na página de detalhes de cada armazenamento de chaves externas, o estado da conexão é exibido na seção General configuration (Configuração geral).

Quando o estado da conexão é `FAILED`, o código de erro de conexão ajuda a explicar o erro.

Para visualizar o código de erro de conexão:

- Na [DescribeCustomKeyStores](#) resposta, visualize o valor do `ConnectionErrorCode` elemento. Este elemento aparece na resposta de `DescribeCustomKeyStores` somente quando `ConnectionState` é `FAILED`.
- Para visualizar o código de erro de conexão no AWS KMS console, na página de detalhes do armazenamento externo de chaves e passe o mouse sobre o valor Falha.



Códigos de erro de conexão para armazenamentos de chaves externas

Os códigos de erro de conexão a seguir se aplicam a armazenamentos de chaves externas

INTERNAL_ERROR

AWS KMS não foi possível concluir a solicitação devido a um erro interno. Repetir a solicitação . Para solicitações ConnectCustomKeyStore, desconecte o armazenamento de chaves personalizado antes de tentar se conectar novamente.

INVALID_CREDENTIALS

Um ou ambos os valores de XksProxyAuthenticationCredential não são válidos no proxy de armazenamento de chaves externas especificado.

NETWORK_ERRORS

Os erros de rede estão AWS KMS impedindo a conexão do armazenamento de chaves personalizadas ao armazenamento de chaves de apoio.

XKS_PROXY_ACCESS_DENIED

AWS KMS as solicitações têm acesso negado ao proxy externo do armazenamento de chaves. Se o proxy de armazenamento de chaves externas tiver regras de autorização, verifique se elas permitem que o AWS KMS se comunique com o proxy em seu nome.

XKS_PROXY_INVALID_CONFIGURATION

Um erro de configuração está impedindo que o armazenamento de chaves externas se conecte ao proxy. Verifique o valor de XksProxyUriPath.

XKS_PROXY_INVALID_RESPONSE

AWS KMS não é possível interpretar a resposta do proxy externo do armazenamento de chaves. Caso veja esse código de erro de conexão repetidamente, notifique seu fornecedor de proxy de armazenamento de chaves externas.

XKS_PROXY_INVALID_TLS_CONFIGURATION

AWS KMS não é possível se conectar ao proxy externo do armazenamento de chaves porque a configuração do TLS é inválida. Verifique se o proxy de armazenamento de chaves externas é compatível com TLS 1.2 ou 1.3. Além disso, verifique se o certificado TLS não expirou, se corresponde ao nome do host no valor `XksProxyUriEndpoint` e se está assinado por uma autoridade de certificação confiável incluída na lista de [autoridades de certificação confiáveis](#).

XKS_PROXY_NOT_REACHABLE

AWS KMS não consegue se comunicar com seu proxy externo de armazenamento de chaves. Verifique se `XksProxyUriEndpoint` e `XksProxyUriPath` estão corretos. Use as ferramentas do proxy de armazenamento de chaves externas para verificar se o proxy está ativo e disponível em sua rede. Além disso, verifique se as instâncias do gerenciador de chaves externas estão operando corretamente. As tentativas de conexão falharão com esse código de erro de conexão se o proxy relatar que todas as instâncias do gerenciador de chaves externas estão indisponíveis.

XKS_PROXY_TIMED_OUT

AWS KMS pode se conectar ao proxy externo do armazenamento de chaves, mas o proxy não responde AWS KMS no tempo alocado. Caso veja esse código de erro de conexão repetidamente, notifique seu fornecedor de proxy de armazenamento de chaves externas.

XKS_VPC_ENDPOINT_SERVICE_INVALID_CONFIGURATION

A configuração do serviço de endpoint do Amazon VPC não está em conformidade com os requisitos de um AWS KMS armazenamento de chaves externo.

- O serviço de endpoint da VPC deve ser um serviço de endpoint para endpoints de interface na Conta da AWS do autor da chamada.
- Deve ter um balanceador de carga de rede (NLB) conectado a pelo menos duas sub-redes, cada uma em uma zona de disponibilidade diferente.
- A `Allow principals` lista deve incluir o principal AWS KMS de serviço da região, `cks.kms.<region>.amazonaws.com`, tal como `cks.kms.us-east-1.amazonaws.com`.
- Não deve exigir a [aceitação](#) de solicitações de conexão.

- Ele deve ter um nome DNS privado. O nome DNS privado de um armazenamento de chaves externas com conectividade VPC_ENDPOINT_SERVICE deve ser exclusivo na Região da AWS.
- O domínio do nome DNS privado deve ter o [status de verificação](#) `verified`.
- O [certificado TLS](#) especifica o nome de host DNS privado no qual o endpoint é acessível.

XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND

AWS KMS não consegue encontrar o serviço de endpoint VPC que ele usa para se comunicar com o proxy externo do armazenamento de chaves. Verifique se `XksProxyVpcEndpointServiceName` está correto e se a entidade principal do serviço do AWS KMS tem permissões de consumidor no serviço de endpoint da Amazon VPC.

Erros de latência e de tempo limite

Exceções: `CustomKeyStoreInvalidStateException` (`CreateKey`),
`KMSInvalidStateException` (operações de criptografia),
`XksProxyUriUnreachableException` (operações de gerenciamento)

[Códigos de erro de conexão](#): `XKS_PROXY_NOT_REACHABLE`, `XKS_PROXY_TIMED_OUT`

Quando não é possível contatar o proxy dentro do intervalo de tempo limite de 250 milissegundos, ele retorna uma exceção. `CreateCustomKeyStore` e `UpdateCustomKeyStore` retornam `XksProxyUriUnreachableException`. As operações criptográficas retornam o padrão `KMSInvalidStateException` com uma mensagem de erro que descreve o problema. Se `ConnectCustomKeyStore` falhar, AWS KMS retorna um [código de erro de conexão](#) que descreve o problema.

Os erros de tempo limite poderão ser problemas transitórios que podem ser resolvidos ao repetir a solicitação. Se o problema persistir, verifique se o proxy de armazenamento de chaves externas está ativo e conectado à rede e se o endpoint do URI do proxy, o caminho do URI do proxy e o nome do serviço de endpoint da VPC (se houver) estão corretos no armazenamento de chaves externas. Além disso, verifique se o gerenciador de chaves externo está próximo ao da Região da AWS armazenamento de chaves externo. Se precisar atualizar qualquer um desses valores, consulte [Editar propriedades do repositório de chaves externo](#).

Para rastrear os padrões de latência, use a [XksProxyLatency](#) CloudWatch métrica e o gráfico de latência média (com base nessa métrica) na [seção Monitoramento](#) do AWS KMS console. Seu proxy de armazenamento de chaves externas também pode gerar logs e métricas que rastreiam a latência e os tempos limite.

XksProxyUriUnreachableException

AWS KMS não pode se comunicar com o proxy externo do armazenamento de chaves. Pode ser um problema de rede transitório. Caso veja esse erro repetidamente, verifique se o proxy de armazenamento de chaves externas está ativo e conectado à rede e se o URI do endpoint está correto no armazenamento de chaves externas.

- O proxy externo do armazenamento de chaves não respondeu a uma solicitação de API de AWS KMS proxy dentro do intervalo de tempo limite de 250 milissegundos. Isso pode indicar um problema de rede transitório ou um problema operacional ou de performance com o proxy. Se uma nova tentativa não resolver o problema, notifique o administrador do proxy de armazenamento de chaves externas.

Erros de latência e de tempo limite geralmente se manifestam como falhas de conexão. Quando a [ConnectCustomKeyStore](#) operação falha, o estado da conexão do armazenamento de chaves externo muda para FAILED e AWS KMS retorna um código de erro de conexão que explica o erro. Para obter uma lista de códigos de erro de conexão e sugestões para resolver os erros, consulte [Códigos de erro de conexão para armazenamentos de chaves externas](#). As listas de códigos de conexão de todos os armazenamentos de chaves personalizados e Armazenamentos de chaves externas se aplicam aos armazenamentos de chaves externas. Os erros de conexão a seguir estão relacionados à latência e aos tempos limite.

XKS_PROXY_NOT_REACHABLE

- ou -

CustomKeyStoreInvalidStateException , KMSInvalidStateException ,
XksProxyUriUnreachableException

AWS KMS não pode se comunicar com o proxy externo do armazenamento de chaves. Verifique se o proxy de armazenamento de chaves externas está ativo e conectado à rede e se o caminho do URI e o URI do endpoint ou o nome do serviço da VPC estão corretos no armazenamento de chaves externas.

Esse erro poderá ocorrer pelos seguintes motivos:

- O proxy do armazenamento de chaves externas não está ativo ou não está conectado à rede.

- Há um erro nos valores do [endpoint do URI do proxy](#), do [caminho do URI do proxy](#) ou do [nome do serviço de endpoint da VPC](#) (se aplicável) na configuração do armazenamento de chaves externas. Para visualizar a configuração do armazenamento de chaves externo, use a [DescribeCustomKeyStores](#) operação ou [visualize a página de detalhes](#) do armazenamento de chaves externo no AWS KMS console.
- Pode haver um erro de configuração de rede, como um erro de porta, no caminho de rede entre AWS KMS e o proxy externo do armazenamento de chaves. AWS KMS se comunica com o proxy externo do armazenamento de chaves na porta 443 over. IPv4 Esse valor não é configurável.
- Quando o proxy externo do armazenamento de chaves relata (em uma [GetHealthStatus](#) resposta) que todas as instâncias externas do gerenciador de chaves são UNAVAILABLE, a [ConnectCustomKeyStore](#) operação falha com um `ConnectionErrorCode` de `XKS_PROXY_NOT_REACHABLE`. Para obter ajuda, consulte a documentação do gerenciador de chaves externas.
- Esse erro pode resultar de uma longa distância física entre o gerenciador de chaves externo e Região da AWS o armazenamento de chaves externo. A latência de ping (tempo de ida e volta da rede (RTT)) entre o Região da AWS e o gerenciador de chaves externo não deve ser superior a 35 milissegundos. Talvez seja necessário criar um armazenamento de chaves externo em um Região da AWS que esteja mais próximo do gerenciador de chaves externo ou mover o gerenciador de chaves externo para um data center mais próximo do Região da AWS.

XKS_PROXY_TIMED_OUT

- ou -

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ,
`XksProxyUriUnreachableException`

O AWS KMS rejeitou a solicitação porque o proxy de armazenamento de chaves externas não respondeu a tempo. Repetir a solicitação . Caso veja esse erro repetidamente, informe-o ao administrador do proxy de armazenamento de chaves externas.

Esse erro poderá ocorrer pelos seguintes motivos:

- Esse erro pode resultar de uma longa distância física entre o gerenciador de chaves externas e o proxy de armazenamento de chaves externas. Se possível, mova o proxy de armazenamento de chaves externas para mais perto do gerenciador de chaves externas.

- Erros de tempo limite podem ocorrer quando o proxy não foi projetado para lidar com o volume e a frequência das solicitações de AWS KMS. Se suas CloudWatch métricas indicarem um problema persistente, notifique o administrador externo do proxy do armazenamento de chaves.
- Podem ocorrer erros de tempo limite quando a conexão entre o gerenciador de chaves externas e a Amazon VPC para o armazenamento de chaves externas não está funcionando corretamente. Se você estiver usando AWS Direct Connect, verifique se sua VPC e seu gerenciador de chaves externo podem se comunicar de forma eficaz. Para obter ajuda para resolver qualquer problema, consulte [Solução de problemas AWS Direct Connect](#) no Guia do Direct Connect usuário.

XKS_PROXY_TIMED_OUT

- ou -

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ,
`XksProxyUriUnreachableException`

O proxy de armazenamento de chaves externas não respondeu à solicitação no tempo estipulado. Repetir a solicitação . Caso veja esse erro repetidamente, informe-o ao administrador do proxy de armazenamento de chaves externas.

- Esse erro pode resultar de uma longa distância física entre o gerenciador de chaves externas e o proxy de armazenamento de chaves externas. Se possível, mova o proxy de armazenamento de chaves externas para mais perto do gerenciador de chaves externas.

Erros de credenciais de autenticação

Exceções: `CustomKeyStoreInvalidStateException` (`CreateKey`),
`KMSInvalidStateException` (operações de criptografia),
`XksProxyIncorrectAuthenticationCredentialException` (operações de gerenciamento, exceto `CreateKey`)

Você estabelece e mantém uma credencial de autenticação AWS KMS em seu proxy externo de armazenamento de chaves. Em seguida, você informa os valores AWS KMS das credenciais ao criar um armazenamento de chaves externo. Para alterar a credencial de autenticação, faça a alteração no proxy de armazenamento de chaves externas. Em seguida, [atualize a credencial](#) do armazenamento de chaves externas. Se o proxy alternar a credencial, você deverá [atualizar a credencial](#) do armazenamento de chaves externas.

Se o proxy de armazenamento de chaves externas não autenticar uma solicitação assinada com a [credencial de autenticação do proxy](#) para seu armazenamento de chaves externas, o efeito dependerá da solicitação:

- `CreateCustomKeyStore` e `UpdateCustomKeyStore` falham com uma `XksProxyIncorrectAuthenticationCredentialException`.
- `ConnectCustomKeyStore` é bem-sucedido, mas a conexão falha. O estado da conexão é `FAILED`, e o código de erro da conexão é `INVALID_CREDENTIALS`. Para obter detalhes, consulte [Erros de conexão do armazenamento de chaves externas](#).
- As operações criptográficas retornam `KMSInvalidStateException` para todos os erros de configuração externa e erros de estado de conexão em um repositório de chaves externo. A mensagem de erro que acompanha descreve o problema.

O proxy de armazenamento de chaves externas rejeitou a solicitação porque não conseguiu autenticar o AWS KMS. Verifique as credenciais para o armazenamento de chaves externas e atualize-as, se necessário.

Esse erro poderá ocorrer pelos seguintes motivos:

- O ID da chave de acesso ou a chave de acesso secreta para o armazenamento de chaves externas não correspondem aos valores estabelecidos no proxy de armazenamento de chaves externas.

Para corrigir esse erro, [atualize a credencial de autenticação do proxy](#) para o armazenamento de chaves externas. Você pode fazer essa alteração sem desconectar o armazenamento de chaves externas.

- Um proxy reverso entre AWS KMS e o proxy externo do armazenamento de chaves pode estar manipulando cabeçalhos HTTP de forma a invalidar as assinaturas SigV4. Para corrigir esse erro, notifique o administrador do proxy.

Erros de estado da chave

Exceções: `KMSInvalidStateException`

Utiliza-se `KMSInvalidStateException` para duas finalidades distintas para as chaves do KMS em armazenamentos de chaves personalizados.

- Quando uma operação de gerenciamento, como `CancelKeyDeletion`, falha e retorna essa exceção, isso indica que o [estado](#) da chave do KMS não é compatível com a operação.
- Quando uma [operação de criptografia](#) em uma chave do KMS em um armazenamento de chaves personalizado falha com `KMSInvalidStateException`, pode indicar um problema no estado da chave do KMS. Mas a operação AWS KMS criptográfica retorna `KMSInvalidStateException` para todos os erros de configuração externa e erros de estado de conexão em um armazenamento de chaves externo. Para identificar o problema, use o texto da mensagem de erro que acompanha a exceção.

Para encontrar o estado de chave necessário para as operações de uma AWS KMS API, consulte [Principais estados das AWS KMS chaves](#). Para encontrar o estado de chave de uma chave do KMS, na página Customer managed keys (Chaves gerenciadas pelo cliente), visualize o campo Status da chave do KMS. Ou use a [DescribeKey](#) operação e visualize o `KeyState` elemento na resposta. Para obter detalhes, consulte [Identificar e visualizar chaves](#).

 Note

O estado da chave de uma chave do KMS em um armazenamento de chaves externas não indica nada sobre o status da [chave externa](#) associada. Para obter informações sobre o status da chave externa, use o gerenciador de chaves externas e as ferramentas de proxy de armazenamento de chaves externas.

`CustomKeyStoreInvalidStateException` refere-se ao [estado da conexão](#) do armazenamento de chaves externas, não ao [estado da chave](#) de uma chave do KMS.

Uma operação criptográfica em uma chave do KMS em um armazenamento personalizado poderá falhar se o estado da chave do KMS for `Unavailable` ou `PendingDeletion`. (As chaves desativadas retornam `DisabledException`.)

- Uma chave KMS tem um estado de `Disabled` chave somente quando você desativa intencionalmente a chave KMS no AWS KMS console ou usando a operação. [DisableKey](#) Enquanto uma chave do KMS está desabilitada, é possível visualizar e gerenciar a chave, mas não é possível usá-la para operações de criptografia. Para corrigir o problema, habilite a chave. Para obter detalhes, consulte [Habilitar e desabilitar chaves](#).
- Uma chave do KMS tem um estado da chave `Unavailable` quando o armazenamento de chaves externas é desconectado do proxy de armazenamento de chaves externas. Para corrigir uma chave do KMS indisponível, [reconecte o armazenamento de chaves externas](#). Após a

reconexão do armazenamento de chaves externas, o estado de chave das chaves do KMS nesse armazenamento de chaves externas é automaticamente restaurado ao estado anterior, como Enabled ou Disabled.

Uma chave do KMS tem um estado de chave PendingDeletion quando foi programada para exclusão e está no período de espera. Um erro de estado da chave em uma chave do KMS que está pendente de exclusão indica que a chave não deve ser excluída, seja porque está sendo usada para criptografia, seja porque é obrigatória para descriptografia. Para reabilitar a chave do KMS, cancele a exclusão programada e [habilite a chave](#). Para obter detalhes, consulte [Agendar exclusão de chave](#).

Erros de descriptografia

Exceções: KMSInvalidStateException

Quando uma operação de [descriptografia](#) com uma chave KMS em um armazenamento de chaves externo falha, AWS KMS retorna o padrão KMSInvalidStateException que as operações criptográficas usam para todos os erros de configuração externa e erros de estado de conexão em um armazenamento de chaves externo. A mensagem de erro indica o problema.

Para descriptografar um texto cifrado que foi criptografado usando [criptografia dupla](#), o gerenciador de chaves externas usa primeiro a chave externa para descriptografar a camada externa do texto cifrado. Em seguida, AWS KMS usa o material AWS KMS chave na chave KMS para decifrar a camada interna do texto cifrado. O gerenciador de chaves externas ou o AWS KMS podem rejeitar um texto cifrado inválido ou corrompido.

As mensagens de erro a seguir acompanham a KMSInvalidStateException quando a descriptografia falha. Isso indica um problema no texto cifrado ou no contexto de criptografia opcional da solicitação.

O proxy de armazenamento de chaves externas rejeitou a solicitação porque o texto cifrado especificado ou os dados autenticados adicionais estão corrompidos, ausentes ou são inválidos.

- Quando o proxy externo do armazenamento de chaves ou o gerenciador de chaves externo relatam que um texto cifrado ou seu contexto de criptografia é inválido, isso normalmente indica um problema com o texto cifrado ou o contexto de criptografia na solicitação enviada para.

Decrypt AWS KMS Para Decrypt operações, AWS KMS envia ao proxy o mesmo texto cifrado e contexto de criptografia que ele recebe na Decrypt solicitação.

Esse erro pode ser causado por um problema de redes em trânsito, como um bit invertido. Repetir a solicitação Decrypt. Se o problema persistir, verifique se o texto cifrado não foi alterado ou corrompido. Além disso, verifique se o contexto de criptografia na Decrypt solicitação AWS KMS corresponde ao contexto de criptografia na solicitação que criptografou os dados.

O texto cifrado que o proxy de armazenamento de chaves externas enviou para descriptografia, ou o contexto de criptografia, está corrompido, ausente ou é inválido.

- Quando AWS KMS rejeita o texto cifrado recebido do proxy, indica que o gerenciador de chaves externo ou proxy retornou um texto cifrado inválido ou corrompido para AWS KMS

Esse erro pode ser causado por um problema de redes em trânsito, como um bit invertido. Repetir a solicitação Decrypt. Se o problema persistir, verifique se o gerenciador de chaves externo está funcionando corretamente e se o proxy externo do armazenamento de chaves não altera o texto cifrado que recebe do gerenciador de chaves externo antes de retorná-lo. AWS KMS

Erros de chave externa

A [chave externa](#) é uma chave de criptografia do gerenciador de chaves externas que serve como material de chave externa para uma chave do KMS. O AWS KMS não consegue acessar diretamente a chave externa. Ele deve solicitar que o gerenciador de chaves externas (por meio do proxy de armazenamento de chaves externas) use a chave externa para criptografar dados ou descriptografar um texto cifrado.

Você especifica o ID da chave externa em seu gerenciador de chaves externas ao criar uma chave do KMS em seu armazenamento de chaves externas. Você não pode alterar o ID da chave externa depois que a chave do KMS é criada. Para evitar problemas com a chave do KMS, a operação `CreateKey` solicita que o proxy de armazenamento de chaves externas verifique o ID e a configuração da chave externa. Se a chave externa não [atender aos requisitos](#) de uso com uma chave do KMS, a operação `CreateKey` falhará com uma exceção e uma mensagem de erro que identificam o problema.

No entanto, poderão ocorrer problemas depois que a chave do KMS for criada. Se uma operação criptográfica falhar devido a um problema com a chave externa, a operação falhará e retornará um `KMSInvalidStateException` com uma mensagem de erro que indica o problema.

CreateKey erros na chave externa

Exceções: `XksKeyAlreadyInUseException`, `XksKeyNotFoundException`, `XksKeyInvalidConfigurationException`

A [CreateKey](#) operação tenta verificar a ID e as propriedades da chave externa que você fornece no parâmetro ID da chave externa (console) ou `XksKeyId` (API). Essa prática foi criada para detectar erros antes de você tentar usar a chave externa com a chave do KMS.

Chave externa em uso

Cada chave do KMS em um armazenamento de chaves externas deve usar uma chave externa diferente. Quando `CreateKey` reconhece que o ID da chave externa (`XksKeyId`) de uma chave KMS não é exclusivo no armazenamento de chaves externo, ele falha com um `XksKeyAlreadyInUseException`.

Se você usar várias IDs para a mesma chave externa, `CreateKey` não reconhecerá a duplicata. No entanto, as chaves KMS com a mesma chave externa não são interoperáveis porque têm materiais de AWS KMS chave e metadados diferentes.

Chave externa não encontrada

Quando o proxy externo do armazenamento de chaves relata que não consegue encontrar a chave externa usando o ID da chave externa (`XksKeyId`) da chave KMS, a `CreateKey` operação falha e retorna `XksKeyNotFoundException` com a seguinte mensagem de erro.

O proxy de armazenamento de chaves externas rejeitou a solicitação porque não conseguiu localizar a chave externa.

Esse erro poderá ocorrer pelos seguintes motivos:

- Talvez o ID da chave externa (`XksKeyId`) da chave do KMS seja inválido. Para encontrar o ID usado pelo proxy de chaves externas para identificar a chave externa, consulte a documentação do proxy de armazenamento de chaves externas ou do gerenciador de chaves externas.

- A chave externa pode ter sido excluída do gerenciador de chaves externas. Para investigar, use suas ferramentas de gerenciamento de chaves externas. Se a chave externa for excluída permanentemente, use outra chave externa com a chave do KMS. Para obter uma lista ou requisitos da chave externa, consulte [Requisitos para uma chave do KMS em um armazenamento de chaves externas](#).

Requisitos de chaves externas não atendidos

Quando o proxy de armazenamento de chaves externas relata que a chave externa não [atende aos requisitos](#) de uso com uma chave do KMS, a operação CreateKey apresenta falhas e retorna XksKeyInvalidConfigurationException com uma das mensagens de erro abaixo.

A especificação de chave da chave externa deve ser AES_256. A especificação chave da chave externa especificada é *<key-spec>* .

- A chave externa deve ser uma chave de criptografia simétrica de 256 bits com uma especificação de chave AES_256. Se a chave externa especificada for de um tipo diferente, especifique o ID de uma chave externa que atenda a esse requisito.

O status da chave externa deve ser ENABLED. O status da chave externa especificada é *<status>*.

- A chave externa deve estar habilitada no gerenciador de chaves externas. Se a chave externa especificada não estiver habilitada, use as ferramentas do gerenciador de chaves externas para habilitá-la ou especifique uma chave externa habilitada.

O uso de chave da chave externa deve incluir ENCRYPT e DECRYPT. O uso chave da chave externa especificada é *<key-usage>* .

- A chave externa deve estar configurada para criptografia e descryptografia no gerenciador de chaves externas. Se a chave externa especificada não incluir essas operações, use as

ferramentas do gerenciador de chaves externas para alterar as operações ou especifique outra chave externa.

Erros de operação de criptografia da chave externa

Exceções: `KMSInvalidStateException`

Quando o proxy de armazenamento de chaves externas não consegue encontrar a chave externa associada à chave do KMS, ou quando a chave externa não [atende aos requisitos](#) de uso com uma chave do KMS, a operação de criptografia falha.

Os problemas de chave externa detectados durante uma operação de criptografia são mais difíceis de resolver do que os problemas de chave externa detectados antes de criar a chave do KMS. Você não pode alterar o ID da chave externa depois que a chave do KMS é criada. Se a chave do KMS ainda não tiver criptografado nenhum dado, você poderá excluir a chave do KMS e criar uma nova com outro ID de chave externa. No entanto, o texto cifrado gerado com a chave KMS não pode ser descryptografado por nenhuma outra chave KMS, mesmo com a mesma chave externa, porque as chaves terão metadados e materiais de chave diferentes. AWS KMS Em vez disso, na medida do possível, use suas ferramentas de gerenciamento de chaves externas para resolver o problema com a chave externa.

Quando o proxy de armazenamento de chaves externas relata um problema com a chave externa, as operações criptográficas retornam `KMSInvalidStateException` com uma mensagem de erro que identifica o problema.

Chave externa não encontrada

Quando o proxy externo do armazenamento de chaves relata que não consegue encontrar a chave externa usando o ID da chave externa (`XksKeyId`) da chave KMS, as operações criptográficas retornam a `KMSInvalidStateException` com a seguinte mensagem de erro.

O proxy de armazenamento de chaves externas rejeitou a solicitação porque não conseguiu localizar a chave externa.

Esse erro poderá ocorrer pelos seguintes motivos:

- O ID da chave externa (`XksKeyId`) da chave do KMS não é mais válido.

Para encontrar o ID da chave externa associada à sua chave do KMS, [visualize os detalhes da chave do KMS](#). Para encontrar o ID que o proxy de chaves externas usa para identificar a chave externa, consulte a documentação do proxy de armazenamento de chaves externas ou do gerenciador de chaves externas.

AWS KMS verifica o ID da chave externa ao criar uma chave KMS em um armazenamento de chaves externo. Porém, o ID poderá se tornar inválido, sobretudo se o valor do ID da chave externa for um alias ou nome mutável. Você não pode alterar o ID da chave externa associada a uma chave do KMS existente. Para descriptografar qualquer texto cifrado criptografado sob a chave do KMS, é necessário associar novamente a chave externa ao ID da chave externa existente.

Se você ainda não usou a chave do KMS para criptografar dados, poderá criar uma nova chave do KMS com um ID de chave externa válido. No entanto, se você gerou texto cifrado com a chave do KMS, não poderá usar nenhuma outra chave do KMS para descriptografar o texto cifrado, mesmo usando a mesma chave externa.

- A chave externa pode ter sido excluída do gerenciador de chaves externas. Para investigar, use suas ferramentas de gerenciamento de chaves externas. Se possível, tente [recuperar o material de chave](#) de uma cópia ou backup do gerenciador de chaves externas. Se a chave externa for excluída permanentemente, todo texto cifrado criptografado sob a chave do KMS associada se tornará irrecoverável.

Erros de configuração de chave externa

Quando o proxy de armazenamento de chaves externas relata que a chave externa não [atende aos requisitos](#) de uso com uma chave do KMS, a operação criptográfica retorna `KMSInvalidStateException` com uma das mensagens de erro abaixo.

O proxy de armazenamento de chaves externas rejeitou a solicitação porque a chave externa não é compatível com a operação solicitada.

- A chave externa deve oferecer suporte a criptografia e descriptografia. Se o uso da chave não incluir criptografia e descriptografia, use suas ferramentas de gerenciamento de chaves externas para alterar o uso da chave.

O proxy de armazenamento de chaves externas rejeitou a solicitação porque a chave externa não está habilitada no gerenciador de chaves externas.

- A chave externa deve estar habilitada e disponível para uso no gerenciador de chaves externas. Se o status da chave externa não for `Enabled`, use as ferramentas do gerenciador de chaves externas para habilitá-la.

Problemas de proxy

Exceções:

`CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (operações de criptografia), `UnsupportedOperationException`, `XksProxyUriUnreachableException`, `XksProxyInvalidResponseException` (operações de gerenciamento, exceto `CreateKey`)

O proxy externo do armazenamento de chaves medeia toda a comunicação entre AWS KMS e o gerenciador de chaves externo. Ele traduz AWS KMS solicitações genéricas em um formato que seu gerente de chaves externo possa entender. Se o proxy externo do armazenamento de chaves não estiver em conformidade com a [especificação da API AWS KMS External Key Store Proxy](#), se não estiver operando corretamente ou não conseguir se comunicar com ele AWS KMS, você não poderá criar ou usar chaves KMS no seu armazenamento de chaves externo.

Embora muitos erros mencionem o proxy de armazenamento de chaves externas por causa de seu papel essencial na arquitetura de armazenamento de chaves externas, esses problemas podem se originar no gerenciador de chaves externas ou na chave externa.

Os problemas nesta seção estão relacionados a problemas no design ou na operação do proxy de armazenamento de chaves externas. A solução desses problemas pode exigir uma alteração no software do proxy. Consulte o administrador do proxy. Para ajudar a diagnosticar problemas de proxy, o AWS KMS fornece o [XKS Proxy Text Client](#), um cliente de teste de código aberto que verifica se seu proxy de armazenamento de chaves externas está em conformidade com a [AWS KMS External Key Store Proxy API Specification](#) (Especificação da API de proxy de armazenamento de chaves externas do).

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ou `XksProxyUriUnreachableException`

O proxy de armazenamento de chaves externas está no estado não íntegro. Caso veja essa mensagem repetidamente, notifique o administrador do proxy de armazenamento de chaves externas.

- Esse erro pode indicar um problema operacional ou erro de software no proxy de armazenamento de chaves externas. Você pode encontrar entradas de CloudTrail registro para a operação AWS KMS da API que gerou cada erro. É possível resolver esse erro repetindo a operação. No entanto, se persistir, notifique o administrador do proxy de armazenamento de chaves externas.
- Quando o proxy do armazenamento de chaves externo relata (em [GetHealthStatus](#) resposta) que todas as instâncias externas do gerenciador de chaves são UNAVAILABLE, as tentativas de criar ou atualizar um armazenamento de chaves externo falham com essa exceção. Se esse erro persistir, consulte a documentação do gerenciador de chaves externas.

`CustomKeyStoreInvalidStateException`, `KMSInvalidStateException` ou `XksProxyInvalidResponseException`

AWS KMS não é possível interpretar a resposta do proxy externo do armazenamento de chaves. Caso veja esse erro repetidamente, consulte o administrador do proxy de armazenamento de chaves externas.

- AWS KMS as operações geram essa exceção quando o proxy retorna uma resposta indefinida que AWS KMS não pode ser analisada ou interpretada. Esse erro poderá ocorrer ocasionalmente devido a problemas externos temporários ou a erros de rede esporádicos. Porém, se persistir, isso pode indicar que o proxy do armazenamento de chaves externas não está em conformidade com a [AWS KMS External Key Store Proxy API Specification](#) (Especificação da API de proxy de armazenamento de chaves externas do). Notifique o administrador ou fornecedor do armazenamento de chaves externas.

`CustomKeyStoreInvalidStateException`, `KMSInvalidStateException` ou `UnsupportedOperationException`

O proxy de armazenamento de chaves externas rejeitou a solicitação porque não é compatível com a operação de criptografia solicitada.

- O proxy externo do armazenamento de chaves deve oferecer suporte a todos os [proxy APIs](#) definidos na [Especificação da API AWS KMS External Key Store Proxy](#). Esse erro indica que o proxy não é compatível com a operação relacionada à solicitação. Notifique o administrador ou fornecedor do armazenamento de chaves externas.

Problemas de autorização de proxy

Exceções: `CustomKeyStoreInvalidStateException`, `KMSInvalidStateException`

Alguns proxies de armazenamento de chaves externas implementam requisitos de autorização para o uso de suas chaves externas. O proxy de armazenamento de chaves externas é permitido, mas não obrigatório, para criar e implementar um esquema de autorização que permita que usuários específicos solicitem operações específicas sob certas condições. Por exemplo, o proxy pode dar ao usuário A permissão para criptografar com uma chave externa específica, mas não para descriptografar com ela. Para obter mais informações, consulte [Autorização de proxy de armazenamento de chaves externas \(opcional\)](#).

A autorização de proxy é baseada nos metadados AWS KMS incluídos em suas solicitações ao proxy. Os campos `awsSourceVpc` e `awsSourceVpce` são incluídos nos metadados somente quando a solicitação vem de um endpoint da VPC e somente quando o autor da chamada está na mesma conta da chave do KMS.

```
"requestMetadata": {  
  "awsPrincipalArn": string,  
  "awsSourceVpc": string, // optional  
  "awsSourceVpce": string, // optional  
  "kmsKeyArn": string,  
  "kmsOperation": string,  
  "kmsRequestId": string,  
  "kmsViaService": string // optional  
}
```

Quando o proxy rejeita uma solicitação devido a uma falha na autorização, a AWS KMS operação relacionada falha. `CreateKey` devolve `CustomKeyStoreInvalidStateException`. AWS KMS as operações criptográficas retornam `KMSInvalidStateException`. Ambos usam esta mensagem de erro:

O proxy de armazenamento de chaves externas negou o acesso à operação. Verifique se o usuário e a chave externa estão autorizados para essa operação e repita a solicitação.

- Para resolver o erro, use o gerenciador de chaves externas ou as ferramentas de proxy de armazenamento de chaves externas para determinar por que a autorização falhou. Em seguida, atualize o procedimento que causou a solicitação não autorizada ou use suas ferramentas de proxy de armazenamento de chaves externas para atualizar a política de autorização. Você não pode solucionar este erro no AWS KMS.

Segurança do AWS Key Management Service

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de um data center e de uma arquitetura de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como a segurança da nuvem e a segurança na nuvem:

- **Segurança da nuvem** — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade que se aplicam a AWS Key Management Service (AWS KMS), consulte [AWS Serviços no escopo do programa de conformidade AWS](#) .
- **Segurança na nuvem** — Sua responsabilidade é determinada pelo AWS serviço que você usa. Além de sua configuração e uso de AWS KMS keys, você é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e regulamentações aplicáveis AWS KMS

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar AWS Key Management Service. Ele mostra como configurar para atender AWS KMS aos seus objetivos de segurança e conformidade.

Tópicos

- [Proteção de dados em AWS Key Management Service](#)
- [Gerenciamento de identidade e acesso para AWS Key Management Service](#)
- [Registro e monitoramento em AWS Key Management Service](#)
- [Validação de conformidade para AWS Key Management Service](#)
- [Resiliência em AWS Key Management Service](#)
- [Segurança da infraestrutura em AWS Key Management Service](#)

Proteção de dados em AWS Key Management Service

AWS Key Management Service armazena e protege suas chaves de criptografia para torná-las altamente disponíveis e, ao mesmo tempo, fornecer a você um controle de acesso forte e flexível.

Tópicos

- [Proteger material de chave](#)
- [Criptografia de dados](#)
- [Privacidade do tráfego entre redes](#)

Proteger material de chave

Por padrão, AWS KMS gera e protege o material da chave criptográfica para chaves KMS. Além disso, AWS KMS oferece opções para materiais essenciais criados e protegidos externamente AWS KMS.

Protegendo o material essencial gerado em AWS KMS

Quando você cria uma chave KMS, por padrão, AWS KMS gera e protege o material criptográfico da chave KMS.

Para proteger o material chave das chaves KMS, AWS KMS conta com uma frota distribuída de módulos de segurança de hardware validados pelo [FIPS 140-3 Security Level 3](#) (). HSMs Cada AWS KMS HSM é um dispositivo de hardware dedicado e autônomo projetado para fornecer funções criptográficas dedicadas para atender aos requisitos de segurança e escalabilidade da AWS KMS (Os HSMs que são AWS KMS usados nas regiões da China são certificados pela [OSCCA](#) e estão em conformidade com todas as regulamentações chinesas pertinentes, mas não são validados pelo Programa de Validação do Módulo Criptográfico FIPS 140-3.)

O material de chave de uma chave do KMS é criptografado por padrão quando é gerado no HSM. O material da chave é descriptografado somente na memória volátil do HSM e somente por alguns milissegundos necessários para ser usado em uma operação criptográfica. Sempre que o material chave não está em uso ativo, ele é criptografado no HSM e transferido para um armazenamento persistente [altamente durável](#) (99,999999999%) e de baixa latência, onde permanece separado e isolado do. HSMs O material de chave em texto nunca deixa o [limite de segurança](#) de HSM; ele nunca é gravado em disco ou em qualquer mídia de armazenamento persistente. (A única exceção é a chave pública de um par de chaves assimétricas, que não é secreta.)

AWS afirma como princípio fundamental de segurança que não há interação humana com material de chave criptográfica de texto simples de qualquer tipo em nenhum. AWS service (Serviço da AWS) Não há mecanismo para que ninguém, incluindo AWS service (Serviço da AWS) operadores, visualize, acesse ou exporte material chave em texto simples. Esse princípio se aplica mesmo durante falhas catastróficas e eventos de recuperação de desastres. O material de chave do cliente em texto simples AWS KMS é usado para operações criptográficas dentro do AWS KMS FIPS 140-3, validadas HSMs somente em resposta às solicitações autorizadas feitas ao serviço pelo cliente ou seu representante.

Para [chaves gerenciadas pelo cliente](#), Conta da AWS quem cria a chave é o proprietário único e intransferível da chave. A conta proprietária tem controle total e exclusivo sobre as políticas de autorização que controlam o acesso à chave. Pois Chaves gerenciadas pela AWS, Conta da AWS tem controle total sobre as políticas do IAM que autorizam solicitações para o. AWS service (Serviço da AWS)

Protegendo o material essencial gerado fora do AWS KMS

AWS KMS fornece alternativas aos principais materiais gerados em AWS KMS.

[Armazenamentos de chaves personalizadas](#), um AWS KMS recurso opcional, permitem criar chaves KMS apoiadas por material de chaves que é gerado e usado externamente. AWS KMS As chaves KMS nos [armazenamentos de AWS CloudHSM chaves](#) são apoiadas por chaves nos módulos AWS CloudHSM de segurança de hardware que você controla. Eles HSMs são certificados no [FIPS 140-2 Security Level 3 ou 140-3 Security Level 3](#). As chaves KMS em [armazenamentos de chaves externos](#) são apoiadas por chaves em um gerenciador de chaves externo que você controla e gerencia externamente AWS, como um HSM físico em seu data center privado.

Outro recurso opcional permite [importar o material de chave](#) para uma chave do KMS. Para proteger o material de chaves importado enquanto ele está em trânsito AWS KMS, você criptografa o material de chaves usando uma chave pública de um par de chaves RSA gerado em um AWS KMS HSM. O material da chave importada é descriptografado em um AWS KMS HSM e recriptografado com uma chave simétrica no HSM. Como todo material de AWS KMS chave, o material de chave importado em texto simples nunca sai da HSMs criptografia. No entanto, o cliente que forneceu o material de chave é responsável pelo uso seguro, a durabilidade e a manutenção do material de chave fora do AWS KMS.

Criptografia de dados

Os dados AWS KMS contidos consistem no material da chave de criptografia que eles representam. AWS KMS keys Esse material chave existe em texto simples somente nos módulos de segurança de AWS KMS hardware (HSMs) e somente quando está em uso. Caso contrário, o material de chave é criptografado e armazenado em armazenamento persistente durável.

O material de chaves AWS KMS gerado para as chaves KMS nunca sai do limite do AWS KMS HSMs não criptografado. Ele não é exportado ou transmitido em nenhuma operação de AWS KMS API. A exceção é para [chaves multirregionais](#), em que AWS KMS usa um mecanismo de replicação entre regiões para copiar o material da chave de uma chave multirregional de um HSM em um Região da AWS para um HSM em outro. Região da AWS Para obter detalhes, consulte [Processo de replicação para chaves multirregionais](#) em Detalhes AWS Key Management Service criptográficos.

Tópicos

- [Criptografia em repouso](#)
- [Criptografia em trânsito](#)

Criptografia em repouso

AWS KMS gera material essencial para módulos de segurança AWS KMS keys de hardware compatíveis com [FIPS 140-3 Security Level 3](#) (). HSMs A única exceção são as regiões da China, onde as HSMs que AWS KMS usam para gerar chaves KMS estão em conformidade com todas as regulamentações chinesas pertinentes, mas não são validadas pelo Programa de Validação de Módulos Criptográficos FIPS 140-3. Quando não estiver em uso, o material de chave é criptografado por uma chave do HSM e gravado em um armazenamento durável e persistente. O material chave das chaves KMS e as chaves de criptografia que protegem o material da chave nunca saem do formato HSMs de texto simples.

A criptografia e o gerenciamento de material de chave para chaves do KMS são tratados inteiramente pelo AWS KMS.

Para obter mais detalhes, consulte [Trabalhando com AWS KMS keys](#) detalhes AWS Key Management Service criptográficos.

Criptografia em trânsito

O material de chaves AWS KMS gerado para chaves KMS nunca é exportado ou transmitido em operações de AWS KMS API. AWS KMS usa [identificadores de chave](#) para representar as chaves

do KMS nas operações da API. Da mesma forma, o material de chaves KMS em [armazenamentos de chaves AWS KMS personalizadas](#) não é exportável e nunca é transmitido em nossas operações AWS KMS de AWS CloudHSM API.

No entanto, algumas operações de AWS KMS API retornam [chaves de dados](#). Além disso, os clientes podem usar operações de API para [importar material de chave](#) para chaves do KMS selecionadas.

Todas as chamadas de AWS KMS API devem ser assinadas e transmitidas usando o Transport Layer Security (TLS). AWS KMS requer o TLS 1.2 e recomenda o TLS 1.3 em todas as regiões. AWS KMS também suporta TLS híbrido pós-quântico para terminais AWS KMS de serviço em todas as regiões, exceto nas regiões da China. AWS KMS não suporta TLS híbrido pós-quântico para endpoints FIPS em. AWS GovCloud (US) Chamadas para o AWS KMS também exigem um pacote de codificação moderno que seja compatível com sigilo de encaminhamento perfeito, o que significa que o comprometimento de qualquer segredo, como uma chave privada, não compromete também a chave de sessão.

Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para usar AWS KMS endpoints padrão ou terminais AWS KMS FIPS, os clientes devem oferecer suporte ao TLS 1.2 ou posterior. Para saber mais sobre os endpoints FIPS disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#). Para obter uma lista de endpoints AWS KMS FIPS, consulte [AWS Key Management Service endpoints e cotas](#) no. Referência geral da AWS

As comunicações entre os hosts do AWS KMS serviço HSMs são protegidas usando criptografia de curva elíptica (ECC) e padrão avançado de criptografia (AES) em um esquema de criptografia autenticado. Para obter mais detalhes, consulte [Segurança da comunicação interna](#) em Detalhes AWS Key Management Service criptográficos.

Privacidade do tráfego entre redes

AWS KMS suporta um Console de gerenciamento da AWS e um conjunto de operações de API que permitem criar, gerenciar AWS KMS keys e usá-las em operações criptográficas.

AWS KMS suporta duas opções de conectividade de rede da sua rede privada para AWS.

- Uma conexão IPSec VPN pela internet
- [AWS Direct Connect](#), que conecta sua rede interna a um Direct Connect local por meio de um cabo de fibra óptica Ethernet padrão.

Todas as chamadas de AWS KMS API devem ser assinadas e transmitidas usando o Transport Layer Security (TLS). As chamadas também exigem um conjunto de codificação moderno que seja compatível com o [sigilo de encaminhamento perfeito](#). O tráfego para os módulos de segurança de hardware (HSMs) que armazenam material de chaves para chaves KMS é permitido somente de hosts de AWS KMS API conhecidos na rede AWS interna.

Para se conectar AWS KMS diretamente à sua nuvem privada virtual (VPC) sem enviar tráfego pela Internet pública, use VPC endpoints, desenvolvidos por [AWS PrivateLink](#). Para obter mais informações, consulte [Conecte-se AWS KMS por meio de um VPC endpoint](#).

AWS KMS também oferece suporte a uma opção [híbrida de troca de chaves pós-quântica](#) para o protocolo de criptografia de rede Transport Layer Security (TLS). Você pode usar essa opção com o TLS ao se conectar aos endpoints AWS KMS da API.

Gerenciamento de identidade e acesso para AWS Key Management Service

AWS Identity and Access Management (IAM) ajuda você a controlar com segurança o acesso aos AWS recursos. Os administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar AWS KMS os recursos. Para obter mais informações, consulte [Usando políticas do IAM com AWS KMS](#).

As [políticas de chaves](#) são o principal mecanismo para controlar o acesso às chaves KMS em AWS KMS. Cada chave do KMS deve ter uma política de chaves. Você também pode usar [políticas do IAM](#) e [concessões](#), juntamente com políticas de chaves, para controlar o acesso às suas chaves do KMS. Para obter mais informações, consulte [Acesso e permissões de chave do KMS](#).

Se você estiver usando uma Amazon Virtual Private Cloud (Amazon VPC), você pode [criar uma interface de VPC endpoint](#) baseada na tecnologia Powered by. AWS KMS [AWS PrivateLink](#). Você também pode usar políticas de VPC endpoint para determinar quais diretores podem acessar seu AWS KMS endpoint, quais chamadas de API eles podem fazer e qual chave KMS eles podem acessar.

Tópicos

- [AWS políticas gerenciadas para AWS Key Management Service](#)
- [Usando funções vinculadas a serviços para AWS KMS](#)

AWS políticas gerenciadas para AWS Key Management Service

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque elas estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as [políticas gerenciadas pelo cliente](#) que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para saber mais, consulte [AWS Políticas gerenciadas pela](#) no Guia do usuário do IAM.

AWS política gerenciada: AWSKey ManagementServicePowerUser

É possível anexar a política AWSKeyManagementServicePowerUser às suas identidades do IAM.

É possível usar a política gerenciada pela AWSKeyManagementServicePowerUser para conceder às entidades principais do IAM em sua conta as permissões de um usuário avançado. Usuários avançados podem criar chaves do KMS, usar e gerenciar as chaves do KMS que eles criam e visualizar todas as chaves do KMS e identidades do IAM. Entidades principais que têm a política gerenciada AWSKeyManagementServicePowerUser também podem obter permissões de outras origens, incluindo políticas de chave, outras políticas do IAM e concessões.

AWSKeyManagementServicePowerUser é uma política AWS gerenciada do IAM. Para obter mais informações sobre políticas AWS gerenciadas, consulte [políticas AWS gerenciadas](#) no Guia do usuário do IAM.

Note

Permissões nesta política específicas para uma chave do KMS, como `kms:TagResource` e `kms:GetKeyRotationStatus`, só entram em vigor quando a política de chaves para essa chave do KMS [permite explicitamente que a Conta da AWS use as políticas do IAM](#) para

controlar o acesso à chave. Para determinar se uma permissão é específica para uma chave do KMS, consulte [AWS KMS permissões](#) e procure um valor de KMS key (Chave do KMS) na coluna Resources (Recursos).

Essa política concede a um usuário avançado permissões sobre qualquer chave do KMS com uma política de chave que permita a operação. Para permissões entre contas, como `kms:DescribeKey` e `kms:ListGrants`, isso pode incluir chaves do KMS em Contas da AWS não confiáveis. Para obter mais detalhes, consulte [Práticas recomendadas para políticas do IAM](#) e [Permitir que usuários de outras contas usem uma chave do KMS](#). Para determinar se uma permissão é válida em chaves do KMS em outras contas, consulte [AWS KMS permissões](#) e procure por um valor Yes (Sim) na coluna Cross-account use (Uso entre contas).

Para permitir que os diretores visualizem o AWS KMS console sem erros, o diretor precisa da [tag: GetResources](#) permission, que não está incluída na `AWSKeyManagementServicePowerUser` política. Você pode permitir essa permissão em uma política do IAM separada.

A política do IAM gerenciada [AWSKeyManagementServicePowerUser](#) deve incluir as permissões a seguir.

- Permite que as entidades principais criem chaves do KMS. Como esse processo inclui a definição da política de chaves, os usuários avançados podem conceder a si mesmos e a outros permissão para usar e gerenciar as chaves do KMS criadas por eles.
- Permite que as entidades principais criem e excluam [alias](#)es e [tags](#) em todas as chaves do KMS. Alterar uma etiqueta ou um alias pode conceder ou negar uma permissão para usar e gerenciar a chave do KMS. Para obter detalhes, consulte [ABAC para AWS KMS](#).
- Permite que as entidades principais obtenham informações detalhadas sobre todas as chaves do KMS, incluindo seu ARN de chave, configuração criptográfica, política de chaves, alias, tags e [status de alternância](#).
- Permite que as entidades principais listem usuários, grupos e funções do IAM.
- Essa política não permite que as entidades principais usem nem gerenciem chaves do KMS não criadas por eles. No entanto, eles podem alterar alias e tags em todas as chaves do KMS, o que pode permitir ou negar permissão para usar ou gerenciar uma chave do KMS.

Para ver as permissões dessa política, consulte [AWSKeyManagementServicePowerUser](#) na Referência de política AWS gerenciada.

AWS política gerenciada: AWSServiceRoleForKeyManagementServiceCustomKeyStores

Não é possível anexar a `AWSServiceRoleForKeyManagementServiceCustomKeyStores` às entidades do IAM. Essa política está vinculada a uma função vinculada ao serviço que dá AWS KMS permissão para visualizar os AWS CloudHSM clusters associados ao seu armazenamento de AWS CloudHSM chaves e criar a rede para oferecer suporte a uma conexão entre seu armazenamento de chaves personalizado e seu AWS CloudHSM cluster. Para obter mais informações, consulte [Autorização AWS KMS para gerenciar recursos AWS CloudHSM do Amazon EC2](#).

AWS política gerenciada: AWSServiceRoleForKeyManagementServiceMultiRegionKeys

Não é possível anexar a `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` às entidades do IAM. Essa política é anexada a um perfil vinculado ao serviço que concede ao AWS KMS permissão para sincronizar todas as alterações no material de chave de uma chave primária multirregional com suas chaves-réplicas. Para obter mais informações, consulte [Autorizando a sincronização AWS KMS de chaves multirregionais](#).

AWS KMS atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas AWS KMS desde que esse serviço começou a rastrear essas alterações. Para obter alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed RSS na página AWS KMS [Histórico do documento](#).

Alteração	Descrição	Data
AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy : atualizar para uma política existente.	AWS KMS adicionou um campo ID de declaração (Sid) à política gerenciada na versão v2 da política.	21 de novembro de 2024
AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy : atualizar para uma política existente.	AWS KMS adicionou as <code>ec2:DescribeNetworkInterfaces</code> permissões <code>ec2:DescribeVpcs</code> <code>ec2:DescribeNetworkAcls</code> e para	10 de novembro de 2023

Alteração	Descrição	Data
	monitorar as alterações na VPC que contém seu AWS CloudHSM cluster para que AWS KMS possa fornecer mensagens de erro claras em caso de falhas.	
AWS KMS começou a rastrear as alterações	AWS KMS começou a rastrear as mudanças em suas políticas AWS gerenciadas.	10 de novembro de 2023

Usando funções vinculadas a serviços para AWS KMS

AWS Key Management Service usa funções [vinculadas ao serviço AWS Identity and Access Management](#) (IAM). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente a. AWS KMS As funções vinculadas ao serviço são definidas AWS KMS e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Uma função vinculada ao serviço facilita a configuração AWS KMS porque você não precisa adicionar manualmente as permissões necessárias. AWS KMS define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido de outra forma, só AWS KMS pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Você pode excluir um perfil vinculado ao serviço somente depois de excluir os atributos relacionados. Isso protege seus AWS KMS recursos porque você não pode remover inadvertidamente a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com perfis vinculados a serviços, consulte [Serviços da AWS compatíveis com o IAM](#) e procure os serviços que contenham Sim na coluna Service-Linked Role. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Para ver detalhes sobre as atualizações de perfis vinculados ao serviço discutido neste tópico, consulte [AWS KMS atualizações nas políticas AWS gerenciadas](#).

Tópicos

- [Autorização AWS KMS para gerenciar recursos AWS CloudHSM do Amazon EC2](#)
- [Autorizando a sincronização AWS KMS de chaves multirregionais](#)

Autorização AWS KMS para gerenciar recursos AWS CloudHSM do Amazon EC2

Para oferecer suporte aos seus AWS CloudHSM principais armazenamentos, AWS KMS precisa de permissão para obter informações sobre seus AWS CloudHSM clusters. Ele também precisa de permissão para criar a infraestrutura de rede que conecta seu armazenamento de AWS CloudHSM chaves ao AWS CloudHSM cluster. Para obter essas permissões, AWS KMS crie a função `AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculada ao serviço em seu. Conta da AWS Os usuários que criam armazenamentos de AWS CloudHSM chaves devem ter a `iam:CreateServiceLinkedRole` permissão que lhes permita criar funções vinculadas a serviços.

Para ver detalhes sobre atualizações na política

`AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy` gerenciada, consulte [AWS KMS atualizações nas políticas AWS gerenciadas](#).

Tópicos

- [Sobre a função AWS KMS vinculada ao serviço](#)
- [Criar a função vinculada ao serviço](#)
- [Editar a descrição de uma função vinculada ao serviço](#)
- [Excluir a função vinculada ao serviço](#)

Sobre a função AWS KMS vinculada ao serviço

Uma [função vinculada ao serviço](#) é uma função do IAM que dá permissão a um AWS serviço para chamar outros AWS serviços em seu nome. Ele foi projetado para facilitar o uso dos recursos de vários AWS serviços integrados sem precisar criar e manter políticas complexas de IAM. Para obter mais informações, consulte [Usando funções vinculadas a serviços para AWS KMS](#).

Para armazenamentos de AWS CloudHSM chaves, AWS KMS cria a função `AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculada ao serviço com a política `AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy` gerenciada. Essa política concede as seguintes permissões à função:

- [cloudHSM:describe*](#) — detecta alterações no AWS CloudHSM cluster que está anexado ao seu armazenamento de chaves personalizadas.

- [ec2: CreateSecurityGroup](#) — usado quando você [conecta um armazenamento de AWS CloudHSM chaves](#) para criar o grupo de segurança que permite o fluxo de tráfego de rede entre AWS KMS e seu AWS CloudHSM cluster.
- [ec2: AuthorizeSecurityGroupIngress](#) — usado quando você [conecta um armazenamento de AWS CloudHSM chaves](#) para permitir o acesso à rede AWS KMS a partir da VPC que contém AWS CloudHSM seu cluster.
- [ec2: CreateNetworkInterface](#) — usado quando você [conecta um armazenamento de AWS CloudHSM chaves](#) para criar a interface de rede usada para comunicação entre AWS KMS e o AWS CloudHSM cluster.
- [ec2: RevokeSecurityGroupEgress](#) — usado quando você [conecta um armazenamento de AWS CloudHSM chaves](#) para remover todas as regras de saída do grupo de segurança criado. AWS KMS
- [ec2: DeleteSecurityGroup](#) — usado quando você [desconecta um armazenamento de AWS CloudHSM chaves](#) para excluir grupos de segurança que foram criados quando você conectou o armazenamento de AWS CloudHSM chaves.
- [ec2: DescribeSecurityGroups](#) — usado para monitorar alterações no grupo de segurança AWS KMS criado na VPC que contém AWS CloudHSM seu cluster para AWS KMS que possa fornecer mensagens de erro claras em caso de falhas.
- [ec2: DescribeVpcs](#) — usado para monitorar alterações na VPC que contém AWS CloudHSM seu cluster para AWS KMS que possa fornecer mensagens de erro claras em caso de falhas.
- [ec2: DescribeNetworkAcls](#) — usado para monitorar alterações na rede da VPC que contém AWS CloudHSM seu cluster ACLs para AWS KMS que possa fornecer mensagens de erro claras em caso de falhas.
- [ec2: DescribeNetworkInterfaces](#) — usado para monitorar alterações nas interfaces de rede AWS KMS criadas na VPC que contém AWS CloudHSM seu cluster para AWS KMS que possa fornecer mensagens de erro claras em caso de falhas.

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": "*"
}
]
}

```

Como a função `AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculada ao serviço é confiável somente `cks.kms.amazonaws.com`, somente AWS KMS pode assumir essa função vinculada ao serviço. Essa função é limitada às operações que AWS KMS precisam visualizar seus AWS CloudHSM clusters e conectar um armazenamento de AWS CloudHSM chaves ao AWS CloudHSM cluster associado. Ele não fornece AWS KMS nenhuma permissão adicional. Por exemplo, AWS KMS não tem permissão para criar, gerenciar ou excluir seus AWS CloudHSM clusters ou backups. HSMs

Regiões

Assim como o recurso de lojas AWS CloudHSM principais, a `AWSServiceRoleForKeyManagementServiceCustomKeyStores` função é suportada em todos os Regiões da AWS lugares AWS KMS e AWS CloudHSM está disponível. Para obter uma lista do Regiões da AWS que cada serviço suporta, consulte [AWS Key Management Service Endpoints e cotas e AWS CloudHSM endpoints e cotas](#) no. Referência geral da Amazon Web Services

Para obter mais informações sobre como AWS os serviços usam funções vinculadas a serviços, consulte Como [usar funções vinculadas a serviços no Guia do usuário](#) do IAM.

Criar a função vinculada ao serviço

AWS KMS cria automaticamente a função

`AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculada ao serviço em sua Conta da AWS quando você cria um armazenamento de AWS CloudHSM chaves, se a função ainda não existir. Não é possível criar ou criar outra vez essa função vinculada a serviço diretamente.

Editar a descrição de uma função vinculada ao serviço

Você não pode editar o nome da função ou as declarações da política na função vinculada ao serviço `AWSServiceRoleForKeyManagementServiceCustomKeyStores`, mas você pode editar a descrição da função. Para obter mais informações, consulte [Editar um perfil vinculado ao serviço](#), no Guia do usuário do IAM.

Excluir a função vinculada ao serviço

AWS KMS não exclui a função

`AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculada ao serviço de sua Conta da AWS mesmo que você tenha [excluído todos os seus armazenamentos de AWS CloudHSM chaves](#). Embora atualmente não haja nenhum procedimento para excluir a função `AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculada ao serviço, não assume essa função AWS KMS nem usa suas permissões, a menos que você tenha armazenamentos de chaves ativos AWS CloudHSM .

Autorizando a sincronização AWS KMS de chaves multirregionais

Para oferecer suporte a [chaves multirregionais](#), AWS KMS precisa de permissão para sincronizar as [propriedades compartilhadas](#) de uma chave primária multirregional com suas chaves de réplica. Para obter essas permissões, AWS KMS crie a função `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` vinculada ao serviço em seu. Conta da AWS Usuários que criam chaves multirregionais devem ter a permissão `iam:CreateServiceLinkedRole` para criar perfis vinculados a serviços.

Você pode ver o [SynchronizeMultiRegionKey](#) CloudTrail evento que registra a AWS KMS sincronização de propriedades compartilhadas em seus AWS CloudTrail registros.

Para ver detalhes sobre atualizações na política

`AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy` gerenciada, consulte [AWS KMS atualizações nas políticas AWS gerenciadas](#).

Tópicos

- [Sobre a função vinculada ao serviço para chaves de várias regiões](#)
- [Criar a função vinculada ao serviço](#)
- [Editar a descrição de uma função vinculada ao serviço](#)
- [Excluir a função vinculada ao serviço](#)

Sobre a função vinculada ao serviço para chaves de várias regiões

Uma [função vinculada ao serviço](#) é uma função do IAM que dá permissão a um AWS serviço para chamar outros AWS serviços em seu nome. Ele foi projetado para facilitar o uso dos recursos de vários AWS serviços integrados sem precisar criar e manter políticas complexas de IAM.

Para chaves multirregionais, AWS KMS cria a função `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` vinculada ao serviço com a `AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy` política gerenciada. Essa política concede à função a permissão `kms:SynchronizeMultiRegionKey`, que permite sincronizar as propriedades compartilhadas de chaves de várias regiões.

Como a função `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` vinculada ao serviço é confiável somente `kms.amazonaws.com`, somente AWS KMS pode assumir essa função vinculada ao serviço. Essa função é limitada às operações que AWS KMS precisam sincronizar propriedades compartilhadas de várias regiões. Ele não fornece AWS KMS nenhuma permissão adicional. Por exemplo, AWS KMS não tem permissão para criar, replicar ou excluir nenhuma chave KMS.

Para obter mais informações sobre como AWS os serviços usam funções vinculadas ao serviço, consulte Como [usar funções vinculadas ao serviço no Guia do usuário](#) do IAM.

JSON

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

Criar a função vinculada ao serviço

AWS KMS cria automaticamente a função

`AWSServiceRoleForKeyManagementServiceMultiRegionKeys` vinculada ao serviço em sua Conta da AWS quando você cria uma chave multirregional, se a função ainda não existir. Não é possível criar ou criar outra vez essa função vinculada a serviço diretamente.

Editar a descrição de uma função vinculada ao serviço

Você não pode editar o nome da função ou as declarações de política na função

`AWSServiceRoleForKeyManagementServiceMultiRegionKeys` vinculada ao serviço, mas pode editar a descrição da função. Para obter mais informações, consulte [Editar uma função vinculada ao serviço](#), no Manual do usuário do IAM.

Excluir a função vinculada ao serviço

AWS KMS não exclui a função

`AWSServiceRoleForKeyManagementServiceMultiRegionKeys` vinculada ao serviço da sua Conta da AWS e você não pode excluí-la. No entanto, AWS KMS não assume a `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` função nem usa nenhuma de suas permissões, a menos que você tenha chaves multirregionais em sua Conta da AWS região.

Registro e monitoramento em AWS Key Management Service

O monitoramento é uma parte importante para entender a disponibilidade, o estado e o uso das suas AWS KMS keys no AWS KMS. O monitoramento ajuda a manter a segurança, a confiabilidade, a disponibilidade e o desempenho de suas AWS soluções. AWS fornece várias ferramentas para monitorar suas chaves KMS.

AWS CloudTrail Registros

Cada chamada para uma operação de AWS KMS API é capturada como um evento em um AWS CloudTrail registro. Esses registros registram todas as chamadas de API do AWS KMS console e as chamadas feitas por AWS KMS outros AWS serviços. As chamadas de API entre contas, como uma chamada para usar uma chave KMS em outra Conta da AWS, são registradas nos CloudTrail registros de ambas as contas.

Ao solucionar problemas ou fazer auditorias, você pode usar o log para reconstruir o ciclo de vida de uma chave do KMS. Também é possível visualizar seu gerenciamento e uso da chave do KMS

em operações de criptografia. Para obter mais informações, consulte [the section called “Fazendo login com AWS CloudTrail”](#).

CloudWatch Registros da Amazon

Monitore, armazene e acesse seus arquivos de AWS CloudTrail log de outras fontes. Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).

For AWS KMS, CloudWatch armazena informações úteis que ajudam você a evitar problemas com suas chaves KMS e os recursos que elas protegem. Para obter mais informações, consulte [the section called “Monitore as teclas com CloudWatch”](#).

Amazon EventBridge

AWS KMS gera EventBridge eventos quando sua chave KMS é [girada](#) ou [excluída](#) ou o [material da chave importada em sua chave](#) KMS expira. Pesquise AWS KMS eventos (operações de API) e encaminhe-os para uma ou mais funções ou fluxos de destino para capturar informações de estado. Para obter mais informações, consulte [the section called “Monitore as chaves com a Amazon EventBridge”](#) o [Guia do EventBridge usuário da Amazon](#).

CloudWatch Métricas da Amazon

Você pode monitorar suas chaves KMS usando CloudWatch métricas, que coletam e processam dados brutos AWS KMS em métricas de desempenho. Os dados são registrados em intervalos de duas semanas para que você possa visualizar tendências de informações atuais e históricas. Isso ajuda você a entender como as suas chaves do KMS são usadas e como seu uso muda ao longo do tempo. Para obter informações sobre o uso de CloudWatch métricas para monitorar chaves KMS, consulte [AWS KMS métricas e dimensões](#).

CloudWatch Alarmes da Amazon

Observe uma única métrica mudar ao longo de um período que você especificar. Depois, execute uma ou mais ações com base no valor da métrica relativa a um limite por um número de períodos. Por exemplo, você pode criar um CloudWatch alarme que é acionado quando alguém tenta usar uma chave KMS programada para ser excluída em uma operação criptográfica. Isso indica que a chave do KMS ainda está sendo usada e provavelmente não deve ser excluída. Para obter mais informações, consulte [the section called “Criar um alarme”](#).

AWS Security Hub CSPM

Você pode monitorar seu AWS KMS uso dos padrões do setor de segurança e da conformidade com as melhores práticas usando AWS Security Hub CSPM. O CSPM do Security Hub usa controles de segurança para avaliar configurações de recursos e padrões de segurança que

ajudam você a atender a vários frameworks de conformidade. Para obter mais informações, consulte [Controles do AWS Key Management Service](#) no Manual do usuário do AWS Security Hub .

Validação de conformidade para AWS Key Management Service

Audidores terceirizados avaliam a segurança e a conformidade AWS Key Management Service como parte de vários programas de AWS conformidade. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Tópicos

- [Documentos de conformidade e segurança](#)
- [Saiba mais](#)

Documentos de conformidade e segurança

Os seguintes documentos de conformidade e segurança abrangem AWS KMS. Para visualizá-los, use [AWS Artifact](#).

- Cloud Computing Compliance Controls Catalogue (C5)
- ISO 27001:2013 Statement of Applicability (SoA)
- Certificado ISO 27001:2013
- ISO 27017:2015 Statement of Applicability (SoA)
- Certificado ISO 27017:2015
- ISO 27018:2015 Statement of Applicability (SoA)
- Certificado ISO 27018:2014
- Certificado ISO 9001:2015
- Certificado de Conformidade do PCI DSS (AOC) e Resumo de Responsabilidade
- Relatório Service Organization Controls (SOC) 1
- Relatório Service Organization Controls (SOC) 2
- Relatório de Confidencialidade Service Organization Controls (SOC) 2
- FedRAMP-High

Para obter ajuda sobre AWS Artifact como usar, consulte [Como baixar relatórios no AWS Artifact](#).

Saiba mais

Sua responsabilidade de conformidade ao usar AWS KMS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. Se seu uso de AWS KMS estiver sujeito à conformidade com um padrão publicado, AWS fornece recursos para ajudar a:

- [AWS Serviços no escopo por programa de conformidade](#) — Esta página lista AWS os serviços que estão no escopo de programas de conformidade específicos. Para obter informações gerais, consulte [Programas de conformidade da AWS](#).
- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos focados em segurança e conformidade em AWS.
- [AWS Recursos de conformidade](#) — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Config](#) — Esse AWS serviço avalia se suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub CSPM](#) — Este AWS serviço fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub CSPM usa controles de segurança para avaliar seus AWS recursos e verificar sua conformidade com os padrões e as melhores práticas do setor de segurança. Para acessar uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub CSPM](#).

Resiliência em AWS Key Management Service

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Além da infraestrutura AWS global, AWS KMS oferece vários recursos para ajudar a suportar suas necessidades de resiliência e backup de dados. Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Isolamento regional

AWS Key Management Service (AWS KMS) é um serviço regional autossustentável que está disponível em todos. Regiões da AWS O design isolado regionalmente AWS KMS garante que um problema de disponibilidade em uma Região da AWS não afete a AWS KMS operação em nenhuma outra região. AWS KMS foi projetado para garantir zero tempo de inatividade planejado, com todas as atualizações de software e operações de escalonamento realizadas de forma perfeita e imperceptível.

O [Acordo AWS KMS de Nível de Serviço](#) (SLA) inclui um compromisso de serviço de 99,999% para todo o KMS. APIs Para cumprir esse compromisso, o AWS KMS garante que todos os dados e informações de autorização necessários para executar uma solicitação de API estejam disponíveis em todos os hosts regionais que recebem a solicitação.

A AWS KMS infraestrutura é replicada em pelo menos três zonas de disponibilidade (AZs) em cada região. Para garantir que várias falhas de host não afetem o AWS KMS desempenho, foi AWS KMS projetado para atender o AZs tráfego de clientes de qualquer região.

As alterações feitas nas propriedades ou permissões de uma chave do KMS são replicadas para todos os hosts na região a fim de garantir que a solicitação subsequente possa ser processada corretamente por qualquer host na região. As solicitações de [operações criptográficas](#) usando sua chave KMS são encaminhadas para uma frota de módulos de segurança de AWS KMS hardware (HSMs), qualquer um dos quais pode realizar a operação com a chave KMS.

Design de vários locatários

O design multilocatário AWS KMS permite que ele cumpra o SLA de disponibilidade de 99,999% e mantenha altas taxas de solicitação, ao mesmo tempo em que protege a confidencialidade de suas chaves e dados.

Há vários mecanismos de imposição de integridade implantados para garantir que a chave do KMS especificada para a operação criptográfica sempre seja a chave usada.

O material da chave em texto não criptografado para suas chaves do KMS é extensivamente protegido. Assim que é criado, o material da chave é criptografado no HSM e o material de chave criptografado é imediatamente movido para armazenamento seguro e de baixa latência. A chave criptografada é recuperada e descriptografada no HSM no momento do uso. A chave em texto não criptografado permanece na memória do HSM apenas pelo tempo necessário para a conclusão da operação criptográfica. Em seguida, ela é criptografada novamente no HSM e a chave criptografada

é devolvida para o armazenamento. O material chave em texto simples nunca sai do HSMs; nunca é gravado no armazenamento persistente.

Melhores práticas de resiliência em AWS KMS

Para otimizar a resiliência de seus AWS KMS recursos, considere as estratégias a seguir.

- Para apoiar sua estratégia de backup e recuperação de desastres, considere o uso de chaves de várias regiões, que são chaves do KMS criadas em uma Região da AWS e replicadas somente para regiões que você especifica. Com chaves multirregionais, você pode mover recursos criptografados entre Regiões da AWS (dentro da mesma partição) sem nunca expor o texto simples e descriptografar o recurso, quando necessário, em qualquer uma das regiões de destino. As chaves de várias regiões relacionadas são interoperáveis porque compartilham o mesmo material de chave e ID de chave, mas têm políticas independentes de chave para controle de acesso em alta resolução. Para obter detalhes, consulte [Chaves de várias regiões em AWS KMS](#).
- Para proteger suas chaves em um serviço multilocatário AWS KMS, como, certifique-se de usar controles de acesso, incluindo políticas de [chaves e políticas](#) de [IAM](#). Além disso, você pode enviar suas solicitações para AWS KMS usar um endpoint de interface VPC desenvolvido por AWS PrivateLink. Quando você faz isso, toda a comunicação entre sua Amazon VPC AWS KMS é conduzida inteiramente dentro da AWS rede usando um AWS KMS endpoint dedicado restrito à sua VPC. É possível proteger adicionalmente essas solicitações criando uma camada adicional de autorização usando [políticas de endpoint da VPC](#). Para obter mais detalhes, consulte [Conectar-se ao AWS KMS por meio de um endpoint da VPC](#).

Segurança da infraestrutura em AWS Key Management Service

Como um serviço gerenciado, AWS Key Management Service (AWS KMS) é protegido pelos procedimentos AWS globais de segurança de rede descritos em [Amazon Web Services: Visão geral dos processos de segurança](#).

Para acessar AWS KMS pela rede, você pode chamar as operações de AWS KMS API descritas na [Referência da AWS Key Management Service API](#). AWS KMS requer o TLS 1.2 e recomenda o TLS 1.3 em todas as regiões. AWS KMS também suporta TLS híbrido pós-quântico para terminais AWS KMS de serviço em todas as regiões, exceto as regiões da China. AWS KMS não suporta TLS híbrido pós-quântico para endpoints FIPS em AWS GovCloud (US). Para usar os [endpoints padrão do AWS KMS](#) ou [endpoints do FIPS do AWS KMS](#), os clientes devem ser compatíveis com TLS 1.2 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward

secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos, como o Java 7 e versões posteriores, oferece suporte a esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Você pode chamar essas operações de API de qualquer local de rede, mas AWS KMS suporta condições de política global que permitem controlar o acesso a uma chave KMS com base no endereço IP de origem, na VPC e no VPC endpoint. É possível usar essas chaves de condição em políticas de chaves e em políticas do IAM. No entanto, essas condições podem AWS impedir o uso da chave KMS em seu nome. Para obter detalhes, consulte [AWS chaves de condição globais](#).

Por exemplo, a declaração de política chave a seguir permite que os usuários que podem assumir a `KMSTestRole` função a usem AWS KMS key para as [operações criptográficas](#) especificadas, a menos que o endereço IP de origem seja um dos endereços IP especificados na política.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:role/KMSTestRole" },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24",
            "203.0.113.0/24"
          ]
        }
      }
    }
  ]
}
```

```
}  
}  
}  
}
```

Isolamento de hosts físicos

A segurança da infraestrutura física que AWS KMS usa está sujeita aos controles descritos na seção Segurança física e ambiental do [Amazon Web Services: Visão geral dos processos de segurança](#). É possível encontrar mais detalhes em relatórios de conformidade e em descobertas de auditoria de terceiros listados na seção anterior.

AWS KMS é suportado por módulos dedicados de segurança de hardware reforçado (HSMs) projetados com controles específicos para resistir a ataques físicos. HSMs São dispositivos físicos que não têm uma camada de virtualização, como um hipervisor, que compartilha o dispositivo físico entre vários inquilinos lógicos. O material da chave AWS KMS keys é armazenado somente na memória volátil do HSMs, e somente enquanto a chave KMS está em uso. Essa memória é apagada quando o HSM sai do estado operacional, incluindo desligamentos e definições intencionais e não intencionais. Para obter informações detalhadas sobre a operação do AWS KMS HSMs, consulte [Detalhes AWS Key Management Service criptográficos](#).

Cotas

Para tornar AWS KMS responsivo e eficiente para todos os usuários, AWS KMS aplica dois tipos de cotas: cotas de recursos e cotas de solicitação. Cada cota é calculada independentemente para cada Região de cada Conta da AWS.

Todas as AWS KMS cotas são ajustáveis, exceto a cota de [recursos de rotação sob demanda e a cota](#) de solicitação do [armazenamento de AWS CloudHSM chaves](#). Para solicitar um aumento da cota, consulte [Requesting a quota increase](#) no Guia do usuário do Service Quotas. [Para solicitar uma redução de cota, alterar uma cota que não está listada nas Cotas de Serviço ou alterar uma cota em uma área em que as Cotas de Região da AWS Serviço não estejam disponíveis, visite AWS Support o Centro e crie um caso. AWS KMS](#)

Tópicos

- [Cotas de recurso](#)
- [Cotas de solicitações](#)
- [Solicitações de AWS KMS limitação](#)

Cotas de recurso

AWS KMS estabelece cotas de recursos para garantir que possa fornecer um serviço rápido e resiliente a todos os nossos clientes. Algumas cotas de recursos se aplicam somente aos recursos que você cria, mas não aos recursos que os AWS serviços criam para você. Os recursos que você usa, mas que não estão na sua Conta da AWS, como o [Chaves pertencentes à AWS](#), não contam em relação a essas cotas.

Se você exceder um limite de recurso, as solicitações para criar um recurso adicional desse tipo geram uma mensagem de erro `LimitExceededException`.

Todas as cotas de AWS KMS recursos são ajustáveis, exceto a cota de [recursos de rotação sob demanda](#). Para solicitar um aumento da cota, consulte [Requesting a quota increase](#) no Guia do usuário do Service Quotas. [Para solicitar uma redução de cota, alterar uma cota que não está listada nas Cotas de Serviço ou alterar uma cota em uma área em que as Cotas de Região da AWS Serviço não estejam disponíveis, visite AWS Support o Centro e crie um caso. AWS KMS](#)

A tabela a seguir lista e descreve as cotas de AWS KMS recursos em cada Conta da AWS região.

Nome da cota	Valor padrão	Aplica-se a	Ajustável
AWS KMS keys	100.000	Chaves gerenciadas pelo cliente	Sim
Aliases por chave do KMS	50	Aliases criados pelo cliente	Sim
Concessões por chave do KMS	50.000	Chaves gerenciadas pelo cliente	Sim
Cota de recurso de armazenamento de chaves personalizadas	10	Conta da AWS e região	Sim
Alternância sob demanda	25	Chaves gerenciadas pelo cliente	Não

Além das cotas de recursos, AWS KMS usa cotas de solicitação para garantir a capacidade de resposta do serviço. Para obter detalhes, consulte [the section called “Cotas de solicitações”](#).

AWS KMS keys: 100,000

Você pode ter até 100.000 [chaves gerenciadas pelo cliente](#) em cada região da sua Conta da AWS. Esta cota é aplicável a todas as chaves gerenciadas pelo cliente em todas as Regiões da AWS , independentemente da [especificação da chave](#) ou do [estado da chave](#). Cada chave do KMS é considerada um recurso. [Chaves gerenciadas pela AWS](#) e [Chaves pertencentes à AWS](#) não contam para essa cota.

Aliases por chave do KMS: 50

É possível associar até 50 [aliases](#) a cada [chave gerenciada pelo cliente](#). Os aliases AWS associados a [Chaves gerenciadas pela AWS](#) não são contabilizados nessa cota. É possível encontrar essa cota ao [criar](#) ou [atualizar](#) um alias.

 Note

A ResourceAliases condição [kms:](#) é efetiva somente quando a chave KMS está em conformidade com essa cota. Se uma chave do KMS exceder essa cota, as entidades principais autorizadas a usar essa chave pela condição `kms:ResourceAliases` terão acesso negado a ela. Para obter detalhes, consulte [Acesso negado devido à cota de aliases](#).

A cota de aliases por chave KMS substitui a cota de aliases por região, que limitava o número total de aliases em cada região de um. Conta da AWS AWS KMS eliminou a cota de aliases por região.

Concessões por chave do KMS: 50.000

Cada [chave gerenciada pelo cliente](#) pode ter até 50.000 [concessões](#), incluindo as concessões criadas pelos [serviços da AWS integrados ao AWS KMS](#). Essa cota não se aplica a [Chaves gerenciadas pela AWS](#) ou a [Chaves pertencentes à AWS](#).

Um efeito dessa cota é que não é possível executar mais de 50.000 operações autorizadas por concessão que usam a mesma chave do KMS ao mesmo tempo. Depois de atingir a cota, será possível criar concessões na chave do KMS apenas quando uma concessão ativa for retirada ou revogada.

Por exemplo, quando você anexa um volume do Amazon Elastic Block Store (Amazon EBS) a uma instância do Amazon Elastic Compute Cloud (Amazon EC2), o volume é descriptografado para poder ser lido. Para obter permissão para descriptografar os dados, o Amazon EBS cria uma concessão para cada volume. Portanto, se todos os volumes do Amazon EBS usarem a mesma chave do KMS, não será possível anexar mais de 50.000 volumes de uma vez.

Cota de recursos de armazenamento de chaves personalizado: 10

Você pode criar até 10 [lojas de chaves personalizadas](#) em Conta da AWS cada região. Se você tentar criar mais, a [CreateCustomKeyStore](#) operação falhará.

Essa cota se aplica ao número total de armazenamentos de chaves personalizados em cada conta e região, inclusive todos os [armazenamentos de chaves do AWS CloudHSM](#) e [armazenamentos de chaves externas](#), independentemente do estado da conexão.

Rotação sob demanda: 25

Você pode realizar a [rotação de chaves sob demanda](#) no máximo 25 vezes por chave KMS. Se você tentar realizar mais rotações sob demanda, a [RotateKeyOnDemand](#) operação falhará.

Esta cota não é ajustável. Você não pode aumentá-lo usando Quotas de Serviço ou criando um caso em. AWS Support Para evitar atingir a cota de alternância sob demanda, recomendamos usar a [alternância automática de chaves](#) sempre que possível.

Cotas de solicitações

AWS KMS estabelece cotas para o número de operações de API solicitadas em cada segundo. As cotas de solicitação diferem de acordo com a operação da API Região da AWS, a e outros fatores, como o tipo de chave KMS. Quando você excede uma cota de solicitação de API, AWS KMS [limita a solicitação](#).

Todas as cotas de AWS KMS solicitação são ajustáveis, exceto a [cota de solicitação do armazenamento de AWS CloudHSM chaves](#). Para solicitar um aumento da cota, consulte [Requesting a quota increase](#) no Guia do usuário do Service Quotas. [Para solicitar uma redução de cota, alterar uma cota que não está listada nas Cotas de Serviço ou alterar uma cota em uma área em que as Cotas de Região da AWS Serviço não estejam disponíveis, visite AWS Support o Centro e crie um caso. AWS KMS](#)

Se você estiver excedendo a cota de solicitação para a [GenerateDataKey](#) operação, considere usar o recurso de armazenamento em [cache da chave de dados](#) do. AWS Encryption SDK A reutilização de chaves de dados pode reduzir a frequência de suas solicitações para o. AWS KMS

Além de solicitar cotas, AWS KMS usa cotas de recursos para garantir a capacidade de todos os usuários. Para obter detalhes, consulte [Cotas de recurso](#).

Para visualizar as tendências em suas taxas de solicitação, use o [console do Service Quotas](#). Você também pode criar um CloudWatch alarme [da Amazon](#) que o alerta quando sua taxa de solicitação atingir uma determinada porcentagem do valor da cota. Para obter detalhes, consulte [Gerenciar suas taxas de solicitação de AWS KMS API usando Service Quotas e Amazon CloudWatch](#) no Blog de AWS Segurança.

Tópicos

- [Solicite cotas para cada operação de AWS KMS API](#)

- [Aplicar cotas de solicitações](#)
- [Cotas compartilhadas para operações de criptografia](#)
- [Solicitações de API dofeitas em seu nome](#)
- [Solicitações entre contas](#)
- [Cotas de solicitação de armazenamento de chaves personalizadas](#)

Solicite cotas para cada operação de AWS KMS API

Esta tabela lista o código da cota de [Service Quotas](#) e o valor padrão para cada AWS KMS cota de solicitação. Todas as cotas de AWS KMS solicitação são ajustáveis, exceto a [cota de solicitação do armazenamento de AWS CloudHSM chaves](#).

Note

Talvez seja necessário rolar horizontalmente ou verticalmente para ver todos os dados nessa tabela.

Nome da cota	Valor padrão (solicitações por segundo)
Cryptographic operations (symmetric) request rate Aplica-se a: • Decrypt • Encrypt • GenerateDataKey • GenerateDataKeyWithoutPlainText • GenerateMac • GenerateRandom • ReEncrypt • VerifyMac	Essas cotas compartilhadas variam de acordo com o tipo Região da AWS e o tipo de chave KMS usada na solicitação. Cada cota é calculada separadamente. • 10.000 (compartilhada) • 20.000 (compartilhada) nas seguintes regiões: • Leste dos EUA (Ohio), us-east-2 • Ásia-Pacífico (Singapura), ap-southeast-1 • Ásia-Pacífico (Sydney), ap-southeast-2 • Ásia-Pacífico (Tóquio), ap-northeast-1 • Europa (Frankfurt), eu-central-1 • Europa (Londres), eu-west-2

Nome da cota	Valor padrão (solicitações por segundo)
	• 100.000 (compartilhada) nas seguintes regiões: • Leste dos EUA (Norte da Virgínia), us-east-1 • Oeste dos EUA (Oregon), us-west-2 • Europa (Irlanda), eu-west-1
Cryptographic operations (RSA) request rate Aplica-se a: • Decrypt • Encrypt • ReEncrypt • Sign • Verify	1.000 (compartilhada) para chaves do KMS RSA
Cryptographic operations (ML-DSA) request rate Aplica-se a: • Sign • Verify	1.000 (compartilhado) para chaves do KMS ML-DSA

Nome da cota	Valor padrão (solicitações por segundo)
Cryptographic operations (ECC and SM2) request rate Aplica-se a: • Decrypt—compatível somente com chaves SM2 KMS (somente para regiões da China) • DeriveSharedSecret • Encrypt—compatível somente com chaves SM2 KMS (somente para regiões da China) • ReEncrypt —compatível somente com chaves SM2 KMS (somente para regiões da China) • Sign • Verify	1.000 (compartilhadas) para chaves KMS de curva elíptica (ECC) e (somente regiões da SM2 China)
Custom key store request quotas Aplica-se a: • Decrypt • DeriveSharedSecret • Encrypt • GenerateDataKey • GenerateDataKeyWithoutPlainText • GenerateRandom • ReEncrypt	As cotas de solicitação de armazenamento de chaves personalizadas são calculadas separadamente para cada armazenamento de chaves personalizadas. • 1.800 (compartilhados) para cada armazenamento de AWS CloudHSM chaves • 1.800 (compartilhado) para cada armazenamento de chaves externas
CancelKeyDeletion request rate	5
ConnectCustomKeyStore request rate	5

Nome da cota	Valor padrão (solicitações por segundo)
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
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	100
Aplica-se a: - GenerateDataKeyPair - GenerateDataKeyPairWithoutPlaintext	

Nome da cota	Valor padrão (solicitações por segundo)
<code>GenerateDataKeyPair (ECC_NIST_P384) request rate</code> Aplica-se a: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100
<code>GenerateDataKeyPair (ECC_NIST_P521) request rate</code> Aplica-se a: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100
<code>GenerateDataKeyPair (ECC_SECG_P256K1) request rate</code> Aplica-se a: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100
<code>GenerateDataKeyPair (ECC_NIST_EDWARDS25519) request rate</code> Aplica-se a: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100

Nome da cota	Valor padrão (solicitações por segundo)
GenerateDataKeyPair (RSA_2048) request rate Aplica-se a: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	20
GenerateDataKeyPair (RSA_3072) request rate Aplica-se a: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	4
GenerateDataKeyPair (RSA_4096) request rate Aplica-se a: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	1
GenerateDataKeyPair (SM2 – China Regions only) request rate Aplica-se a: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	25
GetKeyPolicy request rate	1000

Nome da cota	Valor padrão (solicitações por segundo)
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
ListKeyRotations request rate	100
ListResourceTags request rate	2000
ListRetirableGrants request rate	100
PutKeyPolicy request rate	15
ReplicateKey request rate	5
Uma operação ReplicateKey conta como uma solicitação ReplicateKey na região da chave primária e duas solicitações CreateKey na região da réplica. Uma das solicitações CreateKey é uma simulação para detectar possíveis problemas antes de criar a chave.	
RetireGrant request rate	50
RevokeGrant request rate	50
RotateKeyOnDemand request rate	5

Nome da cota	Valor padrão (solicitações por segundo)
ScheduleKeyDeletion request rate	15
TagResource request rate	10
UntagResource request rate	5
UpdateAlias request rate	5
UpdateCustomKeyStore request rate	5
UpdateKeyDescription request rate	5
UpdatePrimaryRegion request rate	5
Uma operação UpdatePrimaryRegion conta como duas solicitações UpdatePrimaryRegion ; uma solicitação em cada uma das duas regiões afetadas.	

Aplicar cotas de solicitações

Ao analisar as cotas de solicitações, tenha em mente as seguintes informações.

- Cotas de solicitações aplicam-se a [chaves gerenciadas pelo cliente](#) e a [Chaves gerenciadas pela AWS](#). O uso de [Chaves pertencentes à AWS](#) não conta nas cotas de solicitação para você Conta da AWS, mesmo quando elas são usadas para proteger recursos em sua conta.
- Cotas de solicitações aplicam-se a solicitações enviadas a endpoints FIPS e não FIPS. Para obter uma lista de endpoints de AWS KMS serviço, consulte [AWS Key Management Service endpoints e cotas](#) no. Referência geral da AWS
- O controle de utilização é baseado em todas as solicitações em chaves do KMS de todos os tipos na região. Esse total inclui solicitações de todos os diretores do Conta da AWS, incluindo solicitações de AWS serviços em seu nome.
- Cada cota de solicitações é calculada de maneira independente. Por exemplo, as solicitações da [CreateKey](#) operação não têm efeito na cota de solicitações da [CreateAlias](#) operação. Se as solicitações CreateAlias forem limitadas, as solicitações CreateKey ainda poderão ser concluídas com êxito.

- Embora as operações de criptografia compartilhem uma cota, a cota compartilhada é calculada independentemente das cotas de outras operações. Por exemplo, as chamadas para as operações [Encrypt](#) e [Decrypt](#) compartilham uma cota de solicitação, mas essa cota é independente da cota para operações de gerenciamento, como. [EnableKey](#) Por exemplo, na região Europa (Londres), é possível executar 10.000 operações de criptografia em chaves do KMS simétricas mais 5 operações EnableKey por segundo sem que haja limitação.

Cotas compartilhadas para operações de criptografia

AWS KMS [operações criptográficas compartilham cotas](#) de solicitação. É possível solicitar qualquer combinação de operações de criptografia compatíveis com a chave do KMS, para que o número total de operações de criptografia não exceda a cota de solicitações desse tipo de chave do KMS. As exceções são [GenerateDataKeyPair](#) [GenerateDataKeyPairWithoutPlaintext](#), que compartilham uma cota separada.

As cotas para diferentes tipos de chaves do KMS são calculadas de maneira independente. Cada cota se aplica a todas as solicitações dessas operações na região Conta da AWS e com o tipo de chave determinado em cada intervalo de um segundo.

- A taxa de solicitações de operações de criptografia (simétricas) é a cota de solicitações compartilhadas para operações de criptografia que usam chaves do KMS simétricas em uma conta e região. Essa cota se aplica a operações criptográficas com chaves de criptografia simétrica e chaves de Hash-based message authentication code (HMAC – Código de autenticação de mensagem por hash), que também são simétricas.

Por exemplo, você pode estar usando [chaves KMS simétricas](#) em uma Região da AWS cota compartilhada de 10.000 solicitações por segundo. Quando você faz 7.000 [GenerateDataKey](#) solicitações por segundo e 2.000 solicitações [Decrypt](#) por segundo, AWS KMS não restringe suas solicitações. No entanto, quando você faz 9.500 solicitações [GenerateDataKey](#) e 1.000 solicitações [Encrypt](#) por segundo, o AWS KMS limita as solicitações porque elas excedem a cota compartilhada.

As operações criptográficas nas [chaves do KMS de criptografia simétrica](#) em um [armazenamento de chaves personalizadas](#) contam tanto para a taxa de solicitação de operações criptográficas (simétricas) da conta quanto para a [cota de solicitação de armazenamento de chaves personalizadas](#) para o armazenamento de chaves personalizadas.

- A taxa de solicitações de operações de criptografia (RSA) é a cota de solicitações compartilhadas de operações de criptografia que usam [chaves do KMS assimétricas RSA](#).

Por exemplo, com uma cota de solicitações de 1.000 operações por segundo, é possível fazer 400 solicitações [Encrypt](#) e 200 solicitações [Decrypt](#) com chaves do KMS RSA que podem criptografar e descriptografar, além de 250 solicitações [Sign](#) e 150 solicitações [Verify](#) com chaves do KMS RSA que podem assinar e verificar.

- A taxa de solicitações de operações de criptografia (ECC) é a cota compartilhada de solicitações de operações de criptografia que usam [chaves do KMS assimétricas de curva elíptica \(ECC\)](#) e [chaves do KMS assimétricas SM](#).

Por exemplo, com uma cota de solicitações de 1.000 operações por segundo, você pode fazer 400 solicitações de assinatura e 200 solicitações de verificação com chaves ECC KMS que podem assinar e verificar, além de 250 solicitações de assinatura e 150 solicitações de verificação com chaves SM2 KMS que podem [assinar](#) e [verificar](#).

- A cota de solicitação de armazenamento de chaves personalizadas corresponde à cota de solicitação compartilhada para operações criptográficas em chaves do KMS em um armazenamento de chaves personalizadas. Essa cota é calculada separadamente para cada armazenamento de chaves personalizado.

As operações criptográficas nas [chaves do KMS de criptografia simétrica](#) em um [repositório de chaves personalizado](#) contam tanto para a taxa de solicitação de operações criptográficas (simétricas) da conta quanto para a [cota de solicitações ao repositório de chaves personalizado](#) para o repositório de chaves personalizado.

As cotas para diferentes tipos de chave também são calculadas de forma independente. Por exemplo, na região Ásia-Pacífico (Singapura), se usar chaves do KMS simétricas e assimétricas, você poderá fazer até 10.000 chamadas por segundo com chaves do KMS simétricas (incluindo chaves de HMAC) mais até 500 chamadas adicionais por segundo com suas chaves do KMS assimétricas RSA, mais até 300 solicitações adicionais por segundo com suas chaves do KMS baseadas em ECC.

Solicitações de API dofeitas em seu nome

Você pode fazer solicitações de API diretamente ou usando um AWS serviço integrado que faz solicitações de API AWS KMS em seu nome. A cota se aplica a ambos os tipos de solicitações.

Por exemplo, você pode armazenar dados no Amazon S3 usando a criptografia no lado do servidor com uma chave do KMS (SSE-KMS). Sempre que você carrega ou baixa um objeto do S3 criptografado com SSE-KMS, o Amazon S3 faz uma solicitação `GenerateDataKey` (para uploads)

ou Decrypt (para downloads) em seu nome. AWS KMS Essas solicitações contam para sua cota, portanto, limita AWS KMS as solicitações se você exceder um total combinado de 5.500 (ou 10.000 ou 50.000, dependendo da sua Região da AWS) upload ou download por segundo de objetos do S3 criptografados com SSE-KMS.

Solicitações entre contas

Quando um aplicativo em um Conta da AWS usa uma chave KMS de propriedade de uma conta diferente, isso é conhecido como solicitação entre contas. Nas solicitações entre contas, o AWS KMS limita a conta que faz as solicitações, e não a conta que possui a chave do KMS. Por exemplo, se uma aplicação na conta A usar uma chave do KMS na conta B, o uso da chave do KMS será aplicado somente às cotas na conta A.

Cotas de solicitação de armazenamento de chaves personalizadas

AWS KMS mantém cotas de solicitação para [operações criptográficas](#) nas chaves KMS em um armazenamento de chaves [personalizado](#). Essas cotas de solicitação são calculadas separadamente para cada armazenamento de chaves personalizadas.

Cota de solicitações do armazenamento de chaves personalizado	Valor padrão (solicitações por segundo) para cada armazenamento de chaves personalizadas	Ajustável
AWS CloudHSM cota de solicitação de armazenamento de chaves	1800	Não
Cota de solicitação de armazenamento de chaves externas	1800	Não

Note

AWS KMS as [cotas de solicitação de armazenamento de chaves personalizadas](#) não aparecem no console Service Quotas. Não é possível visualizar ou gerenciar essas cotas usando as operações da API do Service Quotas.

Se o AWS CloudHSM cluster associado a um armazenamento de AWS CloudHSM chaves estiver processando vários comandos, incluindo aqueles não relacionados ao armazenamento de chaves personalizadas, você poderá receber um AWS KMS `ThrottlingException` em uma `lower-than-expected` taxa. Se isso ocorrer, reduza sua taxa de solicitação AWS KMS, reduza a carga não relacionada ou use um AWS CloudHSM cluster dedicado para seu armazenamento de AWS CloudHSM chaves.

AWS KMS relata a limitação de solicitações externas de armazenamento de chaves na [ExternalKeyStoreThrottle](#) CloudWatch métrica. É possível usar essa métrica para visualizar padrões de controle de utilização, criar alarmes e ajustar a cota de solicitação de armazenamento de chaves externas.

Uma solicitação para uma [operação criptográfica](#) em uma chave do KMS em um armazenamento de chaves personalizadas é contabilizada para duas cotas:

- Cota de taxa de solicitação de operações criptográficas (simétricas) (por conta)

As solicitações de operações criptográficas em chaves do KMS em um armazenamento de chaves personalizadas são contabilizadas na cota `Cryptographic operations (symmetric) request rate` para cada região e Conta da AWS. Por exemplo, no Leste dos EUA (Norte da Virgínia) (us-east-1), cada Conta da AWS pode ter até 100.000 solicitações por segundo em chaves do KMS de criptografia simétrica, incluindo as solicitações que usam uma chave do KMS em um repositório de chaves personalizado.

- Cota de solicitação de armazenamento de chaves personalizadas (por armazenamento de chaves personalizadas)

As solicitações de operações criptográficas em chaves do KMS em um armazenamento de chaves personalizadas também são contabilizadas para `Custom key store request quota` de 1.800 operações por segundo. Essas cotas são calculadas separadamente para cada armazenamento de chaves personalizadas. Eles podem incluir solicitações de várias pessoas Contas da AWS que usam chaves KMS no armazenamento de chaves personalizadas.

Por exemplo, uma operação [Encrypt](#) em uma chave do KMS em um repositório de chaves personalizado (dos dois tipos) na região Leste dos EUA (Norte da Virgínia) (us-east-1) contará para a cota em nível de conta de `Cryptographic operations (symmetric) request rate` (100.000 mil solicitações por segundo) para sua conta e região, e para a `Custom key store request quota` (1.800 solicitações por segundo) para seu repositório de chaves personalizado. No

entanto, uma solicitação para uma operação de gerenciamento [PutKeyPolicy](#), como em uma chave KMS em um armazenamento de chaves personalizadas, se aplica somente à cota em nível de conta (15 solicitações por segundo).

Solicitações de AWS KMS limitação

Para garantir que AWS KMS possa fornecer respostas rápidas e confiáveis às solicitações de API de todos os clientes, ele limita as solicitações de API que excedem certos limites.

A limitação ocorre quando AWS KMS rejeita uma solicitação que, de outra forma, poderia ser válida e retorna um `ThrottlingException` erro como o seguinte.

```
You have exceeded the rate at which you may call KMS. Reduce the frequency of your calls.
(Service: AWSKMS; Status Code: 400; Error Code: ThrottlingException; Request ID: <ID>
```

AWS KMS limita as solicitações para as seguintes condições.

- A taxa de solicitações por segundo excede a [cota de AWS KMS solicitações](#) de uma conta e região.

Por exemplo, se os usuários da sua conta enviarem 1.000 `DescribeKey` solicitações em um segundo, AWS KMS limitará todas as `DescribeKey` solicitações subsequentes nesse segundo.

Para responder a um controle de utilização, use uma [estratégia de recuo e repetição](#). Essa estratégia é implementada automaticamente para erros de HTTP 400 em alguns AWS SDKs.

- Uma taxa intermitente ou alta sustentada de solicitações para alterar o estado da mesma chave do KMS. Essa condição é muitas vezes conhecida como "tecla de atalho".

Por exemplo, se um aplicativo em sua conta enviar uma série persistente de `DisableKey` solicitações `EnableKey` e solicitações para a mesma chave KMS, as solicitações serão AWS KMS limitadas. Essa limitação ocorre mesmo que as solicitações não excedam o limite de request-per-second solicitações para as operações `EnableKey` e `DisableKey`.

Para responder ao controle de utilização, ajuste a lógica da aplicação para que ela faça apenas as solicitações necessárias ou consolide as solicitações de várias funções.

- As solicitações de operações em chaves KMS em um [armazenamento de AWS CloudHSM chaves](#) podem ser limitadas a uma lower-than-expected taxa quando o AWS CloudHSM cluster associado

ao armazenamento de AWS CloudHSM chaves está processando vários comandos, incluindo aqueles não relacionados ao armazenamento de chaves. AWS CloudHSM

(AWS KMS não limita mais as solicitações de operações em chaves KMS em um AWS CloudHSM armazenamento de chaves quando não há sessões PKCS #11 disponíveis para o cluster. AWS CloudHSM Em vez disso, ele lança um `KMSInternalException` e recomenda que você repita sua solicitação.)

Para visualizar as tendências em suas taxas de solicitação, use o [console do Service Quotas](#). Você também pode criar um CloudWatch alarme [da Amazon](#) que o alerta quando sua taxa de solicitação atingir uma determinada porcentagem do valor da cota. Para obter detalhes, consulte [Gerenciar suas taxas de solicitação de AWS KMS API usando Service Quotas e Amazon CloudWatch](#) no Blog de AWS Segurança.

Todas as AWS KMS cotas são ajustáveis, exceto a cota de [recursos de rotação sob demanda e a cota](#) de solicitação do [armazenamento de AWS CloudHSM chaves](#). Para solicitar um aumento da cota, consulte [Requesting a quota increase](#) no Guia do usuário do Service Quotas. [Para solicitar uma redução de cota, alterar uma cota que não está listada nas Cotas de Serviço ou alterar uma cota em uma área em que as Cotas de Região da AWS Serviço não estejam disponíveis, visite AWS Support o Centro e crie um caso. AWS KMS](#)

 Note

AWS KMS as [cotas de solicitação de armazenamento de chaves personalizadas](#) não aparecem no console Service Quotas. Não é possível visualizar ou gerenciar essas cotas usando as operações da API do Service Quotas.

Exemplos de código para AWS KMS usar AWS SDKs

Os exemplos de código a seguir mostram como usar AWS KMS com um kit AWS de desenvolvimento de software (SDK).

As noções básicas são exemplos de código que mostram como realizar as operações essenciais em um serviço.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar perfis de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cenários são exemplos de código que mostram como realizar tarefas específicas chamando várias funções dentro de um serviço ou combinadas com outros Serviços da AWS.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- [Exemplos básicos de AWS KMS uso AWS SDKs](#)
 - [Olá AWS Key Management Service](#)
 - [Aprenda o básico AWS KMS com um SDK AWS](#)
 - [Ações para AWS KMS usar AWS SDKs](#)
 - [Use CreateAlias com um AWS SDK ou CLI](#)
 - [Use CreateGrant com um AWS SDK ou CLI](#)
 - [Use CreateKey com um AWS SDK ou CLI](#)
 - [Use Decrypt com um AWS SDK ou CLI](#)
 - [Use DeleteAlias com um AWS SDK ou CLI](#)
 - [Use DescribeKey com um AWS SDK ou CLI](#)
 - [Use DisableKey com um AWS SDK ou CLI](#)
 - [Use EnableKey com um AWS SDK ou CLI](#)
 - [Use EnableKeyRotation com um AWS SDK ou CLI](#)
 - [Use Encrypt com um AWS SDK ou CLI](#)
 - [Use GenerateDataKey com um AWS SDK ou CLI](#)

- [Use GenerateDataKeyWithoutPlaintext com um AWS SDK ou CLI](#)
- [Use GenerateRandom com um AWS SDK ou CLI](#)
- [Use GetKeyPolicy com um AWS SDK ou CLI](#)
- [Use ListAliases com um AWS SDK ou CLI](#)
- [Use ListGrants com um AWS SDK ou CLI](#)
- [Use ListKeyPolicies com um AWS SDK ou CLI](#)
- [Use ListKeys com um AWS SDK ou CLI](#)
- [Use PutKeyPolicy com um AWS SDK ou CLI](#)
- [Use ReEncrypt com um AWS SDK ou CLI](#)
- [Use RetireGrant com um AWS SDK ou CLI](#)
- [Use RevokeGrant com um AWS SDK ou CLI](#)
- [Use ScheduleKeyDeletion com um AWS SDK ou CLI](#)
- [Use Sign com um AWS SDK ou CLI](#)
- [Use TagResource com um AWS SDK ou CLI](#)
- [Use UpdateAlias com um AWS SDK ou CLI](#)
- [Use Verify com um AWS SDK ou CLI](#)
- [Cenários para AWS KMS usar AWS SDKs](#)
 - [Trabalhe com a criptografia de tabelas do DynamoDB usando a v2 AWS Command Line Interface](#)

Exemplos básicos de AWS KMS uso AWS SDKs

Os exemplos de código a seguir mostram como usar o básico do AWS Key Management Service with AWS SDKs.

Exemplos

- [Olá AWS Key Management Service](#)
- [Aprenda o básico AWS KMS com um SDK AWS](#)
- [Ações para AWS KMS usar AWS SDKs](#)
 - [Use CreateAlias com um AWS SDK ou CLI](#)
 - [Use CreateGrant com um AWS SDK ou CLI](#)
 - [Use CreateKey com um AWS SDK ou CLI](#)

- [Use Decrypt com um AWS SDK ou CLI](#)
- [Use DeleteAlias com um AWS SDK ou CLI](#)
- [Use DescribeKey com um AWS SDK ou CLI](#)
- [Use DisableKey com um AWS SDK ou CLI](#)
- [Use EnableKey com um AWS SDK ou CLI](#)
- [Use EnableKeyRotation com um AWS SDK ou CLI](#)
- [Use Encrypt com um AWS SDK ou CLI](#)
- [Use GenerateDataKey com um AWS SDK ou CLI](#)
- [Use GenerateDataKeyWithoutPlaintext com um AWS SDK ou CLI](#)
- [Use GenerateRandom com um AWS SDK ou CLI](#)
- [Use GetKeyPolicy com um AWS SDK ou CLI](#)
- [Use ListAliases com um AWS SDK ou CLI](#)
- [Use ListGrants com um AWS SDK ou CLI](#)
- [Use ListKeyPolicies com um AWS SDK ou CLI](#)
- [Use ListKeys com um AWS SDK ou CLI](#)
- [Use PutKeyPolicy com um AWS SDK ou CLI](#)
- [Use ReEncrypt com um AWS SDK ou CLI](#)
- [Use RetireGrant com um AWS SDK ou CLI](#)
- [Use RevokeGrant com um AWS SDK ou CLI](#)
- [Use ScheduleKeyDeletion com um AWS SDK ou CLI](#)
- [Use Sign com um AWS SDK ou CLI](#)
- [Use TagResource com um AWS SDK ou CLI](#)
- [Use UpdateAlias com um AWS SDK ou CLI](#)
- [Use Verify com um AWS SDK ou CLI](#)

Olá AWS Key Management Service

O exemplo de código a seguir mostra como começar a usar o AWS Key Management Service.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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());
        }
    }
}

```

- Para obter detalhes da API, consulte [ListKeys](#) a Referência AWS SDK for Java 2.x da API.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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;
}
```

- Para obter detalhes da API, consulte [ListKeys](#) a Referência AWS SDK para PHP da API.

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

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.
```

- Para obter detalhes da API, consulte a [ListKeys](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Aprenda o básico AWS KMS com um SDK AWS

Os exemplos de código a seguir mostram como:

- Criar uma chave do KMS.
- Listar chaves KMS para sua conta e obter detalhes sobre elas.
- Habilitar e desabilitar chaves do KMS.
- Gerar uma chave de dados simétrica que possa ser usada para criptografia do lado do cliente.
- Gere uma chave assimétrica usada para assinar dados digitalmente.
- Marcar chaves com tags.
- Excluir chaves do KMS.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

Execute um cenário em um prompt de comando.

```
import software.amazon.awssdk.core.SdkBytes;  
import software.amazon.awssdk.regions.Region;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;
```

```
import software.amazon.awssdk.services.kms.model.AlreadyExistsException;
import software.amazon.awssdk.services.kms.model.DisabledException;
import software.amazon.awssdk.services.kms.model.EnableKeyRotationResponse;
import software.amazon.awssdk.services.kms.model.KmsException;
import software.amazon.awssdk.services.kms.model.NotFoundException;
import software.amazon.awssdk.services.kms.model.RevokeGrantResponse;
import java.util.List;
import java.util.Scanner;
import java.util.concurrent.CompletableFuture;
import java.util.concurrent.CompletionException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class KMSScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
    "-");
    private static String accountId = "";

    private static final Logger logger =
    LoggerFactory.getLogger(KMSScenario.class);

    static KMSActions kmsActions = new KMSActions();

    static Scanner scanner = new Scanner(System.in);

    static String aliasName = "alias/dev-encryption-key";

    public static void main(String[] args) {
        final String usage = ""
            Usage: <granteePrincipal>

            Where:
                granteePrincipal - The principal (user, service account, or group)
to whom the grant or permission is being given.
            """;
    }
```

```
    if (args.length != 1) {
        logger.info(usage);
        return;
    }
    String granteePrincipal = args[0];
    String policyName = "default";

    accountId = kmsActions.getAccountId();
    String keyDesc = "Created by the AWS KMS API";

    logger.info(DASHES);
    logger.info("
        Welcome to the AWS Key Management SDK Basics scenario.

        This program demonstrates how to interact with AWS Key Management
        using the AWS SDK for Java (v2).
        The AWS Key Management Service (KMS) is a secure and highly available
        service that allows you to create
            and manage AWS KMS keys and control their use across a wide range of
        AWS services and applications.
        KMS provides a centralized and unified approach to managing
        encryption keys, making it easier to meet your
            data protection and regulatory compliance requirements.

        This Basics scenario creates two key types:

        - A symmetric encryption key is used to encrypt and decrypt data.
        - An asymmetric key used to digitally sign data.

        Let's get started...
        ");
    waitForInputToContinue(scanner);

    try {
        // Run the methods that belong to this scenario.
        String targetKeyId = runScenario(granteePrincipal, keyDesc, policyName);
        requestDeleteResources(aliasName, targetKeyId);

    } catch (Throwable rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
        {} ", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
```



```

        logger.info("An unexpected error occurred: " + rt.getMessage());
    }
}

private static String runScenario(String granteePrincipal, String keyDesc,
String policyName) throws Throwable {
    logger.info(DASHES);
    logger.info("1. Create a symmetric KMS key\n");
    logger.info("First, the program will creates a symmetric KMS key that you
can used to encrypt and decrypt data.");
    waitForInputToContinue(scanner);
    String targetKeyId;
    try {
        CompletableFuture<String> futureKeyId =
kmsActions.createKeyAsync(keyDesc);
        targetKeyId = futureKeyId.join();
        logger.info("A symmetric key was successfully created " +
targetKeyId);

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("""
        2. Enable a KMS key

        By default, when the SDK creates an AWS key, it is enabled. The next
bit of code checks to
        determine if the key is enabled.
        """);
    waitForInputToContinue(scanner);
    boolean isEnabled;
    try {

```

```

        CompletableFuture<Boolean> futureIsKeyEnabled =
kmsActions.isKeyEnabledAsync(targetKeyId);
        isEnabled = futureIsKeyEnabled.join();
        logger.info("Is the key enabled? {}", isEnabled);

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        throw cause;
    }

    if (!isEnabled)
        try {
            CompletableFuture<Void> future =
kmsActions.enableKeyAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " +
rt.getMessage());
            }
            throw cause;
        }

    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("3. Encrypt data using the symmetric KMS key");
    String plaintext = "Hello, AWS KMS!";
    logger.info(""""
        One of the main uses of symmetric keys is to encrypt and decrypt
data.

        Next, the code encrypts the string {} with the SYMMETRIC_DEFAULT
encryption algorithm.
        """, plaintext);

```

```

        waitForInputToContinue(scanner);
        SdkBytes encryptedData;
        try {
            CompletableFuture<SdkBytes> future =
kmsActions.encryptDataAsync(targetKeyId, plaintext);
            encryptedData = future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof DisabledException kmsDisabledEx) {
                logger.info("KMS error occurred due to a disabled
key: Error message: {}, Error code {}", kmsDisabledEx.getMessage(),
kmsDisabledEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("4. Create an alias");
        logger.info("""

        The alias name should be prefixed with 'alias/'.
        The default, 'alias/dev-encryption-key'.
        """);
        waitForInputToContinue(scanner);

        try {
            CompletableFuture<Void> future =
kmsActions.createCustomAliasAsync(targetKeyId, aliasName);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof AlreadyExistsException kmsExistsEx) {
                if (kmsExistsEx.getMessage().contains("already exists")) {
                    logger.info("The alias '" + aliasName + "' already exists.
Moving on...");
                }
            } else {

```

```

        logger.error("An unexpected error occurred: " + rt.getMessage(),
rt);
        deleteKey(targetKeyId);
        throw cause;
    }
}
waitForInputToContinue(scanner);

logger.info(DASHES);
logger.info("5. List all of your aliases");
waitForInputToContinue(scanner);
try {
    CompletableFuture<Object> future = kmsActions.listAllAliasesAsync();
    future.join();

} catch (RuntimeException rt) {
    Throwable cause = rt.getCause();
    if (cause instanceof KmsException kmsEx) {
        logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
    } else {
        logger.info("An unexpected error occurred: " + rt.getMessage());
    }
    deleteAliasName(aliasName);
    deleteKey(targetKeyId);
    throw cause;
}
waitForInputToContinue(scanner);

logger.info(DASHES);
logger.info("6. Enable automatic rotation of the KMS key");
logger.info(""""

```

By default, when the SDK enables automatic rotation of a KMS key, KMS rotates the key material of the KMS key one year (approximately 365 days) from the enable date and every year thereafter.

```

    """);
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<EnableKeyRotationResponse> future =
kmsActions.enableKeyRotationAsync(targetKeyId);
        future.join();
    }
}

```

```

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("""
        7. Create a grant

        A grant is a policy instrument that allows Amazon Web Services
        principals to use KMS keys.
        It also can allow them to view a KMS key (DescribeKey) and create and
        manage grants.
        When authorizing access to a KMS key, grants are considered along
        with key policies and IAM policies.
        """);

    waitForInputToContinue(scanner);
    String grantId = null;
    try {
        CompletableFuture<String> futureGrantId =
kmsActions.grantKeyAsync(targetKeyId, granteePrincipal);
        grantId = futureGrantId.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteKey(targetKeyId);
        throw cause;
    }
}

```

```

        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("8. List grants for the KMS key");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Object> future =
kmsActions.displayGrantIdsAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("9. Revoke the grant");
        logger.info("""
            The revocation of a grant immediately removes the permissions and
            access that the grant had provided.
            This means that any principal (user, role, or service) that was
            granted access to perform specific
            KMS operations on a KMS key will no longer be able to perform those
            operations.
            """);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<RevokeGrantResponse> future =
kmsActions.revokeKeyGrantAsync(targetKeyId, grantId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {

```

```

        if (kmsEx.getMessage().contains("Grant does not exist")) {
            logger.info("The grant ID '" + grantId + "' does not exist.
Moving on...");
        } else {
            logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            throw cause;
        }
    } else {
        logger.info("An unexpected error occurred: " + rt.getMessage());
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
}

waitForInputToContinue(scanner);

logger.info(DASHES);
logger.info("10. Decrypt the data\n");
logger.info("""
    Lets decrypt the data that was encrypted in an early step.
    The code uses the same key to decrypt the string that we encrypted
earlier in the program.
    """);
waitForInputToContinue(scanner);
String decryptedData = "";
try {
    CompletableFuture<String> future =
kmsActions.decryptDataAsync(encryptedData, targetKeyId);
    decryptedData = future.join();
    logger.info("Decrypted data: " + decryptedData);

} catch (RuntimeException rt) {
    Throwable cause = rt.getCause();
    if (cause instanceof KmsException kmsEx) {
        logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
    } else {
        logger.info("An unexpected error occurred: " + rt.getMessage());
    }
    deleteAliasName(aliasName);
    deleteKey(targetKeyId);
    throw cause;
}

```

```
logger.info("Decrypted text is: " + decryptedData);
waitForInputToContinue(scanner);
```

```
logger.info(DASHES);
logger.info("11. Replace a key policy\n");
logger.info("""
```

A key policy is a resource policy for a KMS key. Key policies are the primary way to control access to KMS keys. Every KMS key must have exactly one key policy. The statements in the key policy determine who has permission to use the KMS key and how they can use it.

You can also use IAM policies and grants to control access to the KMS key, but every KMS key must have a key policy.

By default, when you create a key by using the SDK, a policy is created that gives the AWS account that owns the KMS key full access to the KMS key.

Let's try to replace the automatically created policy with the following policy.

```
    "Version": "2012-10-17",
    "Statement": [{
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::000000000000:root"},
      "Action": "kms:*",
      "Resource": "*"
    }]
  """);
```

```
waitForInputToContinue(scanner);
try {
    CompletableFuture<Boolean> future =
kmsActions.replacePolicyAsync(targetKeyId, policyName, accountId);
    boolean success = future.join();
    if (success) {
        logger.info("Key policy replacement succeeded.");
    } else {
        logger.error("Key policy replacement failed.");
    }
}
```



```

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("12. Get the key policy\n");
    logger.info("The next bit of code that runs gets the key policy to make
sure it exists.");
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<String> future =
kmsActions.getKeyPolicyAsync(targetKeyId, policyName);
        String policy = future.join();
        if (!policy.isEmpty()) {
            logger.info("Retrieved policy: " + policy);
        }
    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("13. Create an asymmetric KMS key and sign your data\n");
    logger.info(""""

```

Signing your data with an AWS key can provide several benefits that make it an attractive option for your data signing needs. By using an AWS KMS key, you can leverage the security controls and compliance features provided by AWS, which can help you meet various regulatory requirements and enhance the overall security posture of your organization.

```

        """);
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<Boolean> future = kmsActions.signVerifyDataAsync();
        boolean success = future.join();
        if (success) {
            logger.info("Sign and verify data operation succeeded.");
        } else {
            logger.error("Sign and verify data operation failed.");
        }
    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("14. Tag your symmetric KMS Key\n");
    logger.info("
        By using tags, you can improve the overall management, security, and
governance of your
        KMS keys, making it easier to organize, track, and control access to
your encrypted data within
        your AWS environment
        """);
    waitForInputToContinue(scanner);
    try {

```

```

        CompletableFuture<Void> future =
kmsActions.tagKMSKeyAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);
    return targetKeyId;
}

// Deletes KMS resources with user input.
private static void requestDeleteResources(String aliasName, String
targetKeyId) {
    logger.info(DASHES);
    logger.info("15. Schedule the deletion of the KMS key\n");
    logger.info("""
        By default, KMS applies a waiting period of 30 days,
        but you can specify a waiting period of 7-30 days. When this
operation is successful,
        the key state of the KMS key changes to PendingDeletion and the key
can't be used in any
        cryptographic operations. It remains in this state for the duration
of the waiting period.

        Deleting a KMS key is a destructive and potentially dangerous
operation. When a KMS key is deleted,
        all data that was encrypted under the KMS key is unrecoverable.
        """);
    logger.info("Would you like to delete the Key Management resources? (y/
n)");
    String delAns = scanner.nextLine().trim();
    if (delAns.equalsIgnoreCase("y")) {
        logger.info("You selected to delete the AWS KMS resources.");
        waitForInputToContinue(scanner);
    }
}

```

```
        try {
            CompletableFuture<Void> future =
kmsActions.deleteSpecificAliasAsync(aliasName);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " +
rt.getMessage());
            }
        }
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Void> future =
kmsActions.disableKeyAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " +
rt.getMessage());
            }
        }

        try {
            CompletableFuture<Void> future =
kmsActions.deleteKeyAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
```

```
        logger.info("An unexpected error occurred: " +
rt.getMessage());
    }
}

} else {
    logger.info("The Key Management resources will not be deleted");
}

logger.info(DASHES);
logger.info("This concludes the AWS Key Management SDK scenario");
logger.info(DASHES);
}

// This method is invoked from Exceptions to clean up the resources.
private static void deleteKey(String targetKeyId) {
    try {
        CompletableFuture<Void> future =
kmsActions.disableKeyAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
    }

    try {
        CompletableFuture<Void> future =
kmsActions.deleteKeyAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
    }
}
```

```

    }
}

// This method is invoked from Exceptions to clean up the resources.
private static void deleteAliasName(String aliasName) {
    try {
        CompletableFuture<Void> future =
kmsActions.deleteSpecificAliasAsync(aliasName);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
    }
}

private static void waitForInputToContinue(Scanner scanner) {
    while (true) {
        logger.info("");
        logger.info("Enter 'c' followed by <ENTER> to continue:");
        String input = scanner.nextLine();

        if (input.trim().equalsIgnoreCase("c")) {
            logger.info("Continuing with the program...");
            logger.info("");
            break;
        } else {
            // Handle invalid input.
            logger.info("Invalid input. Please try again.");
        }
    }
}
}
}

```

Defina uma classe que encapsule as ações do KMS.

```
public class KMSActions {
```

```

    private static final Logger logger =
        LoggerFactory.getLogger(KMSActions.class);
    private static KmsAsyncClient kmsAsyncClient;

    /**
     * Retrieves an asynchronous AWS Key Management Service (KMS) client.
     * <p>
     * This method creates and returns a singleton instance of the KMS async
     client, with the following configurations:
     * <ul>
     * <li>Max concurrency: 100</li>
     * <li>Connection timeout: 60 seconds</li>
     * <li>Read timeout: 60 seconds</li>
     * <li>Write timeout: 60 seconds</li>
     * <li>API call timeout: 2 minutes</li>
     * <li>API call attempt timeout: 90 seconds</li>
     * <li>Retry policy: up to 3 retries</li>
     * <li>Credentials provider: environment variable credentials provider</li>
     * </ul>
     * <p>
     * If the client instance has already been created, it is returned instead of
     creating a new one.
     *
     * @return the KMS async client instance
     */
    private static KmsAsyncClient getAsyncClient() {
        if (kmsAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                .maxConcurrency(100)
                .connectionTimeout(Duration.ofSeconds(60))
                .readTimeout(Duration.ofSeconds(60))
                .writeTimeout(Duration.ofSeconds(60))
                .build();

            ClientOverrideConfiguration overrideConfig =
                ClientOverrideConfiguration.builder()
                    .apiCallTimeout(Duration.ofMinutes(2))
                    .apiCallAttemptTimeout(Duration.ofSeconds(90))
                    .retryPolicy(RetryPolicy.builder()
                        .numRetries(3)
                        .build())
                    .build();

            kmsAsyncClient = KmsAsyncClient.builder()

```

```

        .httpClient(httpClient)
        .overrideConfiguration(overrideConfig)
        .build();
    }
    return kmsAsyncClient;
}

/**
 * Creates a new symmetric encryption key asynchronously.
 *
 * @param keyDesc the description of the key to be created
 * @return a {@link CompletableFuture} that completes with the ID of the
newly created key
 * @throws RuntimeException if an error occurs while creating the key
 */
public CompletableFuture<String> createKeyAsync(String keyDesc) {
    CreateKeyRequest keyRequest = CreateKeyRequest.builder()
        .description(keyDesc)
        .keySpec(KeySpec.SYMMETRIC_DEFAULT)
        .keyUsage(KeyUsageType.ENCRYPT_DECRYPT)
        .build();

    return getAsyncClient().createKey(keyRequest)
        .thenApply(resp -> resp.keyMetadata().keyId())
        .exceptionally(ex -> {
            throw new RuntimeException("An error occurred while creating the
key: " + ex.getMessage(), ex);
        });
}

/**
 * Asynchronously checks if a specified key is enabled.
 *
 * @param keyId the ID of the key to check
 * @return a {@link CompletableFuture} that, when completed, indicates
whether the key is enabled or not
 *
 * @throws RuntimeException if an exception occurs while checking the key
state
 */
public CompletableFuture<Boolean> isKeyEnabledAsync(String keyId) {
    DescribeKeyRequest keyRequest = DescribeKeyRequest.builder()
        .keyId(keyId)
        .build();

```



```

        CompletableFuture<DescribeKeyResponse> responseFuture =
getAsyncClient().describeKey(keyRequest);
        return responseFuture.whenComplete((resp, ex) -> {
            if (resp != null) {
                KeyState keyState = resp.keyMetadata().keyState();
                if (keyState == KeyState.ENABLED) {
                    logger.info("The key is enabled.");
                } else {
                    logger.info("The key is not enabled. Key state: {}",
keyState);
                }
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(resp -> resp.keyMetadata().keyState() == KeyState.ENABLED);
    }

    /**
     * Asynchronously enables the specified key.
     *
     * @param keyId the ID of the key to enable
     * @return a {@link CompletableFuture} that completes when the key has been
enabled
     */
    public CompletableFuture<Void> enableKeyAsync(String keyId) {
        EnableKeyRequest enableKeyRequest = EnableKeyRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<EnableKeyResponse> responseFuture =
getAsyncClient().enableKey(enableKeyRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Key with ID [{}] has been enabled.", keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while enabling
key: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while enabling key: " + exception.getMessage(), exception);
                }
            }
        })
    }

```

```

    });

    return responseFuture.thenApply(response -> null);
}

/**
 * Encrypts the given text asynchronously using the specified KMS client and
 * key ID.
 *
 * @param keyId the ID of the KMS key to use for encryption
 * @param text the text to encrypt
 * @return a CompletableFuture that completes with the encrypted data as an
 * SdkBytes object
 */
public CompletableFuture<SdkBytes> encryptDataAsync(String keyId, String
text) {
    SdkBytes myBytes = SdkBytes.fromUtf8String(text);
    EncryptRequest encryptRequest = EncryptRequest.builder()
        .keyId(keyId)
        .plaintext(myBytes)
        .build();

    CompletableFuture<EncryptResponse> responseFuture =
getAsyncClient().encrypt(encryptRequest).toCompletableFuture();
    return responseFuture.whenComplete((response, ex) -> {
        if (response != null) {
            String algorithm = response.encryptionAlgorithm().toString();
            logger.info("The string was encrypted with algorithm {}.\"",
algorithm);
        } else {
            throw new RuntimeException(ex);
        }
    }).thenApply(EncryptResponse::ciphertextBlob);
}

/**
 * Creates a custom alias for the specified target key asynchronously.
 *
 * @param targetKeyId the ID of the target key for the alias
 * @param aliasName the name of the alias to create
 * @return a {@link CompletableFuture} that completes when the alias creation
 * operation is finished
 */

```

```

    public CompletableFuture<Void> createCustomAliasAsync(String targetKeyId,
String aliasName) {
        CreateAliasRequest aliasRequest = CreateAliasRequest.builder()
            .aliasName(aliasName)
            .targetKeyId(targetKeyId)
            .build();

        CompletableFuture<CreateAliasResponse> responseFuture =
getAsyncClient().createAlias(aliasRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("{} was successfully created.", aliasName);
            } else {
                if (exception instanceof ResourceExistsException) {
                    logger.info("Alias [{}] already exists. Moving on...",
aliasName);
                } else if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while creating
alias: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while creating alias: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(response -> null);
    }

    /**
     * Asynchronously lists all the aliases in the current AWS account.
     *
     * @return a {@link CompletableFuture} that completes when the list of
aliases has been processed
     */
    public CompletableFuture<Object> listAllAliasesAsync() {
        ListAliasesRequest aliasesRequest = ListAliasesRequest.builder()
            .limit(15)
            .build();

        ListAliasesPublisher paginator =
getAsyncClient().listAliasesPaginator(aliasesRequest);
        return paginator.subscribe(response -> {
            response.aliases().forEach(alias ->

```

```

        logger.info("The alias name is: " + alias.aliasName())
    );
})
.thenApply(v -> null)
.exceptionally(ex -> {
    if (ex.getCause() instanceof KmsException) {
        KmsException e = (KmsException) ex.getCause();
        throw new RuntimeException("A KMS exception occurred: " +
e.getMessage());
    } else {
        throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage());
    }
});
}

/**
 * Enables key rotation asynchronously for the specified key ID.
 *
 * @param keyId the ID of the key for which to enable key rotation
 * @return a CompletableFuture that represents the asynchronous operation of
enabling key rotation
 * @throws RuntimeException if there was an error enabling key rotation,
either due to a KMS exception or an unexpected error
 */
public CompletableFuture<EnableKeyRotationResponse>
enableKeyRotationAsync(String keyId) {
    EnableKeyRotationRequest enableKeyRotationRequest =
EnableKeyRotationRequest.builder()
        .keyId(keyId)
        .build();

    CompletableFuture<EnableKeyRotationResponse> responseFuture =
getAsyncClient().enableKeyRotation(enableKeyRotationRequest);
    responseFuture.whenComplete((response, exception) -> {
        if (exception == null) {
            logger.info("Key rotation has been enabled for key with id [{}]",
keyId);
        } else {
            if (exception instanceof KmsException kmsEx) {
                throw new RuntimeException("Failed to enable key rotation: "
+ kmsEx.getMessage(), kmsEx);
            } else {

```

```

        throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
    }
    });

    return responseFuture;
}

/**
 * Grants permissions to a specified principal on a customer master key (CMK)
asynchronously.
 *
 * @param keyId          The unique identifier for the customer master key
(CMK) that the grant applies to.
 * @param granteePrincipal The principal that is given permission to perform
the operations that the grant permits on the CMK.
 * @return A {@link CompletableFuture} that, when completed, contains the ID
of the created grant.
 * @throws RuntimeException If an error occurs during the grant creation
process.
 */
public CompletableFuture<String> grantKeyAsync(String keyId, String
granteePrincipal) {
    List<GrantOperation> grantPermissions = List.of(
        GrantOperation.ENCRYPT,
        GrantOperation.DECRYPT,
        GrantOperation.DESCRIBE_KEY
    );

    CreateGrantRequest grantRequest = CreateGrantRequest.builder()
        .keyId(keyId)
        .name("grant1")
        .granteePrincipal(granteePrincipal)
        .operations(grantPermissions)
        .build();

    CompletableFuture<CreateGrantResponse> responseFuture =
getAsyncClient().createGrant(grantRequest);
    responseFuture.whenComplete((response, ex) -> {
        if (ex == null) {
            logger.info("Grant created successfully with ID: " +
response.grantId());
        } else {

```

```

        if (ex instanceof KmsException kmsEx) {
            throw new RuntimeException("Failed to create grant: " +
kmsEx.getMessage(), kmsEx);
        } else {
            throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage(), ex);
        }
    }
});

return responseFuture.thenApply(CreateGrantResponse::grantId);
}

/**
 * Asynchronously displays the grant IDs for the specified key ID.
 *
 * @param keyId the ID of the AWS KMS key for which to list the grants
 * @return a {@link CompletableFuture} that, when completed, will be null
if the operation succeeded, or will throw a {@link RuntimeException} if the
operation failed
 * @throws RuntimeException if there was an error listing the grants, either
due to an {@link KmsException} or an unexpected error
 */
public CompletableFuture<Object> displayGrantIdsAsync(String keyId) {
    ListGrantsRequest grantsRequest = ListGrantsRequest.builder()
        .keyId(keyId)
        .limit(15)
        .build();

    ListGrantsPublisher paginator =
getAsyncClient().listGrantsPaginator(grantsRequest);
    return paginator.subscribe(response -> {
        response.grants().forEach(grant -> {
            logger.info("The grant Id is: " + grant.grantId());
        });
    })
        .thenApply(v -> null)
        .exceptionally(ex -> {
            Throwable cause = ex.getCause();
            if (cause instanceof KmsException) {
                throw new RuntimeException("Failed to list grants: " +
cause.getMessage(), cause);
            } else {

```

```

        throw new RuntimeException("An unexpected error occurred: " +
cause.getMessage(), cause);
    }
    });
}

/**
 * Revokes a grant for the specified AWS KMS key asynchronously.
 *
 * @param keyId The ID or key ARN of the AWS KMS key.
 * @param grantId The identifier of the grant to be revoked.
 * @return A {@link CompletableFuture} representing the asynchronous
operation of revoking the grant.
 *
 * The {@link CompletableFuture} will complete with a {@link
RevokeGrantResponse} object
 *
 * if the operation is successful, or with a {@code null} value if an
error occurs.
 */
public CompletableFuture<RevokeGrantResponse> revokeKeyGrantAsync(String
keyId, String grantId) {
    RevokeGrantRequest grantRequest = RevokeGrantRequest.builder()
        .keyId(keyId)
        .grantId(grantId)
        .build();

    CompletableFuture<RevokeGrantResponse> responseFuture =
getAsyncClient().revokeGrant(grantRequest);
    responseFuture.whenComplete((response, exception) -> {
        if (exception == null) {
            logger.info("Grant ID: [" + grantId + "] was successfully
revoked!");
        } else {
            if (exception instanceof KmsException kmsEx) {
                if (kmsEx.getMessage().contains("Grant does not exist")) {
                    logger.info("The grant ID '" + grantId + "' does not
exist. Moving on...");
                } else {
                    throw new RuntimeException("KMS error occurred: " +
kmsEx.getMessage(), kmsEx);
                }
            } else {
                throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
            }
        }
    });
}

```

```

        }
    });

    return responseFuture;
}

/**
 * Asynchronously decrypts the given encrypted data using the specified key
 * ID.
 *
 * @param encryptedData The encrypted data to be decrypted.
 * @param keyId The ID of the key to be used for decryption.
 * @return A CompletableFuture that, when completed, will contain the
 *         decrypted data as a String.
 *
 *         If an error occurs during the decryption process, the
 *         CompletableFuture will complete
 *
 *         exceptionally with the error, and the method will return an empty
 *         String.
 */
public CompletableFuture<String> decryptDataAsync(SdkBytes encryptedData,
String keyId) {
    DecryptRequest decryptRequest = DecryptRequest.builder()
        .ciphertextBlob(encryptedData)
        .keyId(keyId)
        .build();

    CompletableFuture<DecryptResponse> responseFuture =
getAsyncClient().decrypt(decryptRequest);
    responseFuture.whenComplete((decryptResponse, exception) -> {
        if (exception == null) {
            logger.info("Data decrypted successfully for key ID: " + keyId);
        } else {
            if (exception instanceof KmsException kmsEx) {
                throw new RuntimeException("KMS error occurred while
decrypting data: " + kmsEx.getMessage(), kmsEx);
            } else {
                throw new RuntimeException("An unexpected error occurred
while decrypting data: " + exception.getMessage(), exception);
            }
        }
    });
}

```



```

        return responseFuture.thenApply(decryptResponse ->
decryptResponse.plaintext().asString(StandardCharsets.UTF_8));
    }

    /**
     * Asynchronously replaces the policy for the specified KMS key.
     *
     * @param keyId      the ID of the KMS key to replace the policy for
     * @param policyName the name of the policy to be replaced
     * @param accountId  the AWS account ID to be used in the policy
     * @return a {@link CompletableFuture} that completes with a boolean
indicating
     *         whether the policy replacement was successful or not
     */
    public CompletableFuture<Boolean> replacePolicyAsync(String keyId, String
policyName, String accountId) {
        String policy = ""
    {
        "Version": "2012-10-17",
        "Statement": [{
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::%s:root"},
            "Action": "kms:*",
            "Resource": "*"
        }]
    }
    """.formatted(accountId);

        PutKeyPolicyRequest keyPolicyRequest = PutKeyPolicyRequest.builder()
            .keyId(keyId)
            .policyName(policyName)
            .policy(policy)
            .build();

        // First, get the current policy to check if it exists
        return getAsyncClient().getKeyPolicy(r ->
r.keyId(keyId).policyName(policyName))
            .thenCompose(response -> {
                logger.info("Current policy exists. Replacing it...");
                return getAsyncClient().putKeyPolicy(keyPolicyRequest);
            })
            .thenApply(putPolicyResponse -> {
                logger.info("The key policy has been replaced.");
                return true;
            });
    }

```

```

    })
    .exceptionally(throwable -> {
        if (throwable.getCause() instanceof LimitExceededException) {
            logger.error("Cannot replace policy, as only one policy is
allowed per key.");
            return false;
        }
        throw new RuntimeException("Error replacing policy", throwable);
    });
}

/**
 * Asynchronously retrieves the key policy for the specified key ID and
policy name.
 *
 * @param keyId      the ID of the AWS KMS key for which to retrieve the
policy
 * @param policyName the name of the key policy to retrieve
 * @return a {@link CompletableFuture} that, when completed, contains the key
policy as a {@link String}
 */
public CompletableFuture<String> getKeyPolicyAsync(String keyId, String
policyName) {
    GetKeyPolicyRequest policyRequest = GetKeyPolicyRequest.builder()
        .keyId(keyId)
        .policyName(policyName)
        .build();

    return getAsyncClient().getKeyPolicy(policyRequest)
        .thenApply(response -> {
            String policy = response.policy();
            logger.info("The response is: " + policy);
            return policy;
        })
        .exceptionally(ex -> {
            throw new RuntimeException("Failed to get key policy", ex);
        });
}

/**
 * Asynchronously signs and verifies data using AWS KMS.
 *
 * <p>The method performs the following steps:

```

```

* <ol>
*   <li>Creates an AWS KMS key with the specified key spec, key usage, and
origin.</li>
*   <li>Signs the provided message using the created KMS key and the
RSASSA-PSS-SHA-256 algorithm.</li>
*   <li>Verifies the signature of the message using the created KMS key
and the RSASSA-PSS-SHA-256 algorithm.</li>
* </ol>
*
* @return a {@link CompletableFuture} that completes with the result of the
signature verification,
*       {@code true} if the signature is valid, {@code false} otherwise.
* @throws KmsException if any error occurs during the KMS operations.
* @throws RuntimeException if an unexpected error occurs.
*/
public CompletableFuture<Boolean> signVerifyDataAsync() {
    String signMessage = "Here is the message that will be digitally signed";

    // Create an AWS KMS key used to digitally sign data.
    CreateKeyRequest createKeyRequest = CreateKeyRequest.builder()
        .keySpec(KeySpec.RSA_2048)
        .keyUsage(KeyUsageType.SIGN_VERIFY)
        .origin(OriginType.AWS_KMS)
        .build();

    return getAsyncClient().createKey(createKeyRequest)
        .thenCompose(createKeyResponse -> {
            String keyId = createKeyResponse.keyMetadata().keyId();

            SdkBytes messageBytes = SdkBytes.fromString(signMessage,
Charset.defaultCharset());
            SignRequest signRequest = SignRequest.builder()
                .keyId(keyId)
                .message(messageBytes)
                .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                .build();

            return getAsyncClient().sign(signRequest)
                .thenCompose(signResponse -> {
                    byte[] signedBytes =
signResponse.signature().asByteArray();

                    VerifyRequest verifyRequest = VerifyRequest.builder()
                        .keyId(keyId)

```

```

.message(SdkBytes.fromByteArray(signMessage.getBytes(Charset.defaultCharset()))

.signature(SdkBytes.fromByteBuffer(ByteBuffer.wrap(signedBytes)))

.signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
    .build();

        return getAsyncClient().verify(verifyRequest)
            .thenApply(verifyResponse -> {
                return (boolean) verifyResponse.signatureValid();
            });
    });
}

.throwable);
});
}

/**
 * Asynchronously tags a KMS key with a specific tag.
 *
 * @param keyId the ID of the KMS key to be tagged
 * @return a {@link CompletableFuture} that completes when the tagging
operation is finished
 */
public CompletableFuture<Void> tagKMSKeyAsync(String keyId) {
    Tag tag = Tag.builder()
        .tagKey("Environment")
        .tagValue("Production")
        .build();

    TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
        .keyId(keyId)
        .tags(tag)
        .build();

    return getAsyncClient().tagResource(tagResourceRequest)
        .thenRun(() -> {
            logger.info("{} key was tagged", keyId);
        })
        .exceptionally(throwable -> {

```

```
        throw new RuntimeException("Failed to tag the KMS key",
throwable);
    });
}

/**
 * Deletes a specific KMS alias asynchronously.
 *
 * @param aliasName the name of the alias to be deleted
 * @return a {@link CompletableFuture} representing the asynchronous
operation of deleting the specified alias
 */
public CompletableFuture<Void> deleteSpecificAliasAsync(String aliasName) {
    DeleteAliasRequest deleteAliasRequest = DeleteAliasRequest.builder()
        .aliasName(aliasName)
        .build();

    return getAsyncClient().deleteAlias(deleteAliasRequest)
        .thenRun(() -> {
            logger.info("Alias {} has been deleted successfully", aliasName);
        })
        .exceptionally(throwable -> {
            throw new RuntimeException("Failed to delete alias: " +
aliasName, throwable);
        });
}

/**
 * Asynchronously disables the specified AWS Key Management Service (KMS)
key.
 *
 * @param keyId the ID or Amazon Resource Name (ARN) of the KMS key to be
disabled
 * @return a CompletableFuture that, when completed, indicates that the key
has been disabled successfully
 */
public CompletableFuture<Void> disableKeyAsync(String keyId) {
    DisableKeyRequest keyRequest = DisableKeyRequest.builder()
        .keyId(keyId)
        .build();

    return getAsyncClient().disableKey(keyRequest)
        .thenRun(() -> {
            logger.info("Key {} has been disabled successfully",keyId);
        });
}
```

```

        })
        .exceptionally(throwable -> {
            throw new RuntimeException("Failed to disable key: " + keyId,
throwable);
        });
    }

    /**
     * Deletes a KMS key asynchronously.
     *
     * <p><strong>Warning:</strong> Deleting a KMS key is a destructive and
potentially dangerous operation.
     * When a KMS key is deleted, all data that was encrypted under the KMS key
becomes unrecoverable.
     * This means that any files, databases, or other data that were encrypted
using the deleted KMS key
     * will become permanently inaccessible. Exercise extreme caution when
deleting KMS keys.</p>
     *
     * @param keyId the ID of the KMS key to delete
     * @return a {@link CompletableFuture} that completes when the key deletion
is scheduled
     */
    public CompletableFuture<Void> deleteKeyAsync(String keyId) {
        ScheduleKeyDeletionRequest deletionRequest =
ScheduleKeyDeletionRequest.builder()
            .keyId(keyId)
            .pendingWindowInDays(7)
            .build();

        return getAsyncClient().scheduleKeyDeletion(deletionRequest)
            .thenRun(() -> {
                logger.info("Key {} will be deleted in 7 days", keyId);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to schedule key deletion for
key ID: " + keyId, throwable);
            });
    }

    public String getAccountId(){
        try (StsClient stsClient = StsClient.create()){

```

```
        GetCallerIdentityResponse callerIdentity =  
        stsClient.getCallerIdentity();  
        return callerIdentity.account();  
    }  
}  
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x .
 - [CreateAlias](#)
 - [CreateGrant](#)
 - [CreateKey](#)
 - [Decrypt](#)
 - [DescribeKey](#)
 - [DisableKey](#)
 - [EnableKey](#)
 - [Encrypt](#)
 - [GetKeyPolicy](#)
 - [ListAliases](#)
 - [ListGrants](#)
 - [ListKeys](#)
 - [RevokeGrant](#)
 - [ScheduleKeyDeletion](#)
 - [Sign](#)
 - [TagResource](#)

PHP

SDK para PHP

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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;
    }
}

}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para PHP .
 - [CreateAlias](#)
 - [CreateGrant](#)
 - [CreateKey](#)
 - [Decrypt](#)
 - [DescribeKey](#)
 - [DisableKey](#)
 - [EnableKey](#)
 - [Encrypt](#)
 - [GetKeyPolicy](#)
 - [ListAliases](#)
 - [ListGrants](#)
 - [ListKeys](#)
 - [RevokeGrant](#)
 - [ScheduleKeyDeletion](#)
 - [Sign](#)
 - [TagResource](#)

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
class KMSScenario:
    """Runs an interactive scenario that shows how to get started with KMS."""

    def __init__(
        self,
        key_manager: KeyManager,
        key_encryption: KeyEncrypt,
        alias_manager: AliasManager,
        grant_manager: GrantManager,
        key_policy: KeyPolicy,
    ):
        self.key_manager = key_manager
        self.key_encryption = key_encryption
        self.alias_manager = alias_manager
        self.grant_manager = grant_manager
        self.key_policy = key_policy
        self.key_id = ""
        self.alias_name = ""
        self.asymmetric_key_id = ""

    def kms_scenario(self):
        key_description = "Created by the AWS KMS API"

        print(DASHES)
        print(
            """
Welcome to the AWS Key Management SDK Basics scenario.

This program demonstrates how to interact with AWS Key Management using the AWS
SDK for Python (Boto3).
The AWS Key Management Service (KMS) is a secure and highly available service
that allows you to create
```

and manage AWS KMS keys and control their use across a wide range of AWS services and applications.

KMS provides a centralized and unified approach to managing encryption keys, making it easier to meet your data protection and regulatory compliance requirements.

This Basics scenario creates two key types:

- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...

```

        """
    )
    q.ask("Press Enter to continue...")

    print(DASHES)
    print(f"1. Create a symmetric KMS key\n")
    print(
        f"First, the program will creates a symmetric KMS key that you can
used to encrypt and decrypt data."
    )
    q.ask("Press Enter to continue...")
    self.key_id = self.key_manager.create_key(key_description)["KeyId"]
    print(f"A symmetric key was successfully created {self.key_id}.")
    q.ask("Press Enter to continue...")
    print(DASHES)
    print(
        """

```

2. Enable a KMS key

By default, when the SDK creates an AWS key, it is enabled. The next bit of code checks to

determine if the key is enabled.

```

        """
    )
    q.ask("Press Enter to continue...")
    is_enabled = self.is_key_enabled(self.key_id)
    print(f"Is the key enabled? {is_enabled}")
    if not is_enabled:
        self.key_manager.enable_key(self.key_id)
    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"3. Encrypt data using the symmetric KMS key")

```

```

    plain_text = "Hello, AWS KMS!"
    print(
        f"""

```

One of the main uses of symmetric keys is to encrypt and decrypt data.

Next, the code encrypts the string "{plain_text}" with the SYMMETRIC_DEFAULT encryption algorithm.

```

        """
    )
    q.ask("Press Enter to continue...")
    encrypted_text = self.key_encryption.encrypt(self.key_id, plain_text)
    print(DASHES)
    print(f"4. Create an alias")
    print(
        """

```

Now, the program will create an alias for the KMS key. An alias is a friendly name that you

can associate with a KMS key. The alias name should be prefixed with 'alias/'.

```

        """
    )
    alias_name = q.ask("Enter an alias name: ", q.non_empty)
    self.alias_manager.create_alias(self.key_id, alias_name)
    print(f"{alias_name} was successfully created.")
    self.alias_name = alias_name
    print(DASHES)
    print(f"5. List all of your aliases")
    q.ask("Press Enter to continue...")
    self.alias_manager.list_aliases(10)
    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"6. Enable automatic rotation of the KMS key")
    print(
        """

```

By default, when the SDK enables automatic rotation of a KMS key, KMS rotates the key material of the KMS key one year (approximately 365 days) from the enable date and every year thereafter.

```

        """
    )
    q.ask("Press Enter to continue...")
    self.key_manager.enable_key_rotation(self.key_id)
    print(DASHES)
    print(f"Key rotation has been enabled for key with id {self.key_id}")
    print(

```

```

    """
7. Create a grant

```

A grant is a policy instrument that allows Amazon Web Services principals to use KMS keys.

It also can allow them to view a KMS key (DescribeKey) and create and manage grants.

When authorizing access to a KMS key, grants are considered along with key policies and IAM policies.

```

    """
    )
    print(
        """

```

To create a grant you must specify a `account_id`. To specify the grantee `account_id`, use the Amazon Resource Name (ARN)

of an AWS `account_id`. Valid principals include AWS accounts, IAM users, IAM roles, federated users, and assumed role users.

```

    """
    )
    account_id = q.ask(
        "Enter an account_id, or press enter to skip creating a grant... "
    )
    grant = None
    if account_id != "":
        grant = self.grant_manager.create_grant(
            self.key_id,
            account_id,
            [
                "Encrypt",
                "Decrypt",
                "DescribeKey",
            ],
        )
        print(f"Grant created successfully with ID: {grant['GrantId']}")

    q.ask("Press Enter to continue...")
    print(DASHES)
    print(DASHES)
    print(f"8. List grants for the KMS key")
    q.ask("Press Enter to continue...")
    self.grant_manager.list_grants(self.key_id)
    q.ask("Press Enter to continue...")
    print(DASHES)

```

```

    print(f"9. Revoke the grant")
    print(
        """

```

The revocation of a grant immediately removes the permissions and access that the grant had provided.

This means that any `account_id` (user, role, or service) that was granted access to perform specific

KMS operations on a KMS key will no longer be able to perform those operations.

```

        """
    )
    q.ask("Press Enter to continue...")

    if grant is not None:
        self.grant_manager.revoke_grant(self.key_id, grant["GrantId"])
        print(f"Grant ID: {grant['GrantId']} was successfully revoked!")

```

```

    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"10. Decrypt the data\n")
    print(
        """

```

Lets decrypt the data that was encrypted in an early step.

The code uses the same key to decrypt the string that we encrypted earlier in the program.

```

        """
    )
    q.ask("Press Enter to continue...")
    decrypted_data = self.key_encryption.decrypt(self.key_id, encrypted_text)
    print(f"Data decrypted successfully for key ID: {self.key_id}")
    print(f"Decrypted data: {decrypted_data}")

```

```

    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"11. Replace a key policy\n")
    print(
        """

```

A key policy is a resource policy for a KMS key. Key policies are the primary way to control

access to KMS keys. Every KMS key must have exactly one key policy. The statements in the key policy

determine who has permission to use the KMS key and how they can use it.

You can also use IAM policies and grants to control access to the KMS key, but every KMS key

must have a key policy.

By default, when you create a key by using the SDK, a policy is created that gives the AWS account that owns the KMS key full access to the KMS key.

Let's try to replace the automatically created policy with the following policy.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"AWS": "arn:aws:iam::000000000000:root"},
    "Action": "kms:*",
    "Resource": "*"
  }]
}
```

```

    """
    )
    account_id = q.ask("Enter your account ID or press enter to skip: ")
    if account_id != "":
        policy = {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {"AWS": f"arn:aws:iam::{account_id}:root"},
                    "Action": "kms:*",
                    "Resource": "*",
                }
            ],
        }

        self.key_policy.set_new_policy(self.key_id, policy)
        print("Key policy replacement succeeded.")
        q.ask("Press Enter to continue...")
    else:
        print("Skipping replacing the key policy.")

    print(DASHES)
    print(f"12. Get the key policy\n")
    print(
        f"The next bit of code that runs gets the key policy to make sure it exists."
    )
    q.ask("Press Enter to continue...")
    policy = self.key_policy.get_policy(self.key_id)
```



```

print(f"The key policy is: {policy}")

q.ask("Press Enter to continue...")
print(DASHES)
print(f"13. Create an asymmetric KMS key and sign your data\n")
print(
    """
    Signing your data with an AWS key can provide several benefits that make
it an attractive option
    for your data signing needs. By using an AWS KMS key, you can leverage
the
    security controls and compliance features provided by AWS,
    which can help you meet various regulatory requirements and enhance the
overall security posture
    of your organization.
    """
)
q.ask("Press Enter to continue...")
print(f"Sign and verify data operation succeeded.")
self.asymmetric_key_id = self.key_manager.create_asymmetric_key()
message = "Here is the message that will be digitally signed"
signature = self.key_encryption.sign(self.asymmetric_key_id, message)
if self.key_encryption.verify(self.asymmetric_key_id, message,
signature):
    print("Signature verification succeeded.")
else:
    print("Signature verification failed.")

q.ask("Press Enter to continue...")
print(DASHES)
print(f"14. Tag your symmetric KMS Key\n")
print(
    """
    By using tags, you can improve the overall management, security, and
governance of your
    KMS keys, making it easier to organize, track, and control access to your
encrypted data within
    your AWS environment
    """
)
q.ask("Press Enter to continue...")
self.key_manager.tag_resource(self.key_id, "Environment", "Production")
self.clean_up()

```

```
def is_key_enabled(self, key_id: str) -> bool:
    """
    Check if the key is enabled or not.

    :param key_id: The key to check.
    :return: True if the key is enabled, otherwise False.
    """
    response = self.key_manager.describe_key(key_id)
    return response["Enabled"] is True
```

```
def clean_up(self):
    """
    Delete resources created by this scenario.
    """
    if self.alias_name != "":
        print(f"Deleting the alias {self.alias_name}.")
        self.alias_manager.delete_alias(self.alias_name)
    window = 7 # The window in days for a scheduled deletion.
    if self.key_id != "":
        print(
            """
```

Warning:

Deleting a KMS key is a destructive and potentially dangerous operation. When a KMS key is deleted, all data that was encrypted under the KMS key is unrecoverable.

```
        """
    )
    if q.ask(
        f"Do you want to delete the key with ID {self.key_id} (y/n)?",
        q.is_yesno,
    ):
        print(
            f"The key {self.key_id} will be deleted with a window of {window} days. You can cancel the deletion before"
        )
        print("the window expires.")
        self.key_manager.delete_key(self.key_id, window)
        self.key_id = ""

    if self.asymmetric_key_id != "":
        if q.ask(
            f"Do you want to delete the asymmetric key with ID {self.asymmetric_key_id} (y/n)?",
            q.is_yesno,
```

```

        ):
            print(
                f"The key {self.asymmetric_key_id} will be deleted with a
window of {window} days. You can cancel the deletion before"
            )
            print("the window expires.")
            self.key_manager.delete_key(self.asymmetric_key_id, window)
            self.asymmetric_key_id = ""

if __name__ == "__main__":
    kms_scenario = None
    try:
        kms_client = boto3.client("kms")
        a_key_manager = KeyManager(kms_client)
        a_key_encrypt = KeyEncrypt(kms_client)
        an_alias_manager = AliasManager(kms_client)
        a_grant_manager = GrantManager(kms_client)
        a_key_policy = KeyPolicy(kms_client)
        kms_scenario = KMSScenario(
            key_manager=a_key_manager,
            key_encryption=a_key_encrypt,
            alias_manager=an_alias_manager,
            grant_manager=a_grant_manager,
            key_policy=a_key_policy,
        )
        kms_scenario.kms_scenario()
    except Exception:
        logging.exception("Something went wrong with the demo!")
        if kms_scenario is not None:
            kms_scenario.clean_up()

```

Classe e métodos do wrapper para gerenciamento de chaves do KMS.

```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":

```

```

    """
    Creates a KeyManager instance with a default KMS client.

    :return: An instance of KeyManager initialized with the default KMS
client.
    """
    kms_client = boto3.client("kms")
    return cls(kms_client)

def create_key(self, key_description: str) -> dict[str, any]:
    """
    Creates a key with a user-provided description.

    :param key_description: A description for the key.
    :return: The key ID.
    """
    try:
        key = self.kms_client.create_key(Description=key_description)
["KeyMetadata"]
        self.created_keys.append(key)
        return key
    except ClientError as err:
        logging.error(
            "Couldn't create your key. Here's why: %s",
            err.response["Error"]["Message"],
        )
        raise

def describe_key(self, key_id: str) -> dict[str, any]:
    """
    Describes a key.

    :param key_id: The ARN or ID of the key to describe.
    :return: Information about the key.
    """

    try:
        key = self.kms_client.describe_key(KeyId=key_id)["KeyMetadata"]
        return key
    except ClientError as err:
        logging.error(
            "Couldn't get key '%s'. Here's why: %s",

```

```

        key_id,
        err.response["Error"]["Message"],
    )
    raise

def enable_key_rotation(self, key_id: str) -> None:
    """
    Enables rotation for a key.

    :param key_id: The ARN or ID of the key to enable rotation for.
    """
    try:
        self.kms_client.enable_key_rotation(KeyId=key_id)
    except ClientError as err:
        logging.error(
            "Couldn't enable rotation for key '%s'. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise

def create_asymmetric_key(self) -> str:
    """
    Creates an asymmetric key in AWS KMS for signing messages.

    :return: The ID of the created key.
    """
    try:
        key = self.kms_client.create_key(
            KeySpec="RSA_2048", KeyUsage="SIGN_VERIFY", Origin="AWS_KMS"
        )["KeyMetadata"]
        self.created_keys.append(key)
        return key["KeyId"]
    except ClientError as err:
        logger.error(
            "Couldn't create your key. Here's why: %s",
            err.response["Error"]["Message"],
        )
        raise

def tag_resource(self, key_id: str, tag_key: str, tag_value: str) -> None:

```

```

"""
Add or edit tags on a customer managed key.

:param key_id: The ARN or ID of the key to enable rotation for.
:param tag_key: Key for the tag.
:param tag_value: Value for the tag.
"""
try:
    self.kms_client.tag_resource(
        KeyId=key_id, Tags=[{"TagKey": tag_key, "TagValue": tag_value}]
    )
except ClientError as err:
    logging.error(
        "Couldn't add a tag for the key '%s'. Here's why: %s",
        key_id,
        err.response["Error"]["Message"],
    )
    raise


def delete_key(self, key_id: str, window: int) -> None:
    """
    Deletes a list of keys.

    Warning:
    Deleting a KMS key is a destructive and potentially dangerous operation.
    When a KMS key is deleted,
    all data that was encrypted under the KMS key is unrecoverable.

    :param key_id: The ARN or ID of the key to delete.
    :param window: The waiting period, in days, before the KMS key is
    deleted.
    """

    try:
        self.kms_client.schedule_key_deletion(
            KeyId=key_id, PendingWindowInDays=window
        )
    except ClientError as err:
        logging.error(
            "Couldn't delete key %s. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )

```

```
raise
```

Classe e métodos do wrapper para aliases de chaves do KMS.

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_alias(self, key_id: str, alias: str) -> None:
        """
        Creates an alias for the specified key.

        :param key_id: The ARN or ID of a key to give an alias.
        :param alias: The alias to assign to the key.
        """
        try:
            self.kms_client.create_alias(AliasName=alias, TargetKeyId=key_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "AlreadyExistsException":
                logger.error(
                    "Could not create the alias %s because it already exists.",
                    key_id
                )
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
```

```

        raise

def list_aliases(self, page_size: int) -> None:
    """
    Lists aliases for the current account.
    :param page_size: The number of aliases to list per page.
    """
    try:
        alias_paginator = self.kms_client.get_paginator("list_aliases")
        for alias_page in alias_paginator.paginate(
            PaginationConfig={"PageSize": page_size}
        ):
            print(f"Here are {page_size} aliases:")
            pprint(alias_page["Aliases"])
            if alias_page["Truncated"]:
                answer = input(
                    f"Do you want to see the next {page_size} aliases (y/n)?
"
                    )
                if answer.lower() != "y":
                    break
            else:
                print("That's all your aliases!")
    except ClientError as err:
        logging.error(
            "Couldn't list your aliases. Here's why: %s",
            err.response["Error"]["Message"],
        )
        raise

def delete_alias(self, alias: str) -> None:
    """
    Deletes an alias.

    :param alias: The alias to delete.
    """
    try:
        self.kms_client.delete_alias(AliasName=alias)
    except ClientError as err:
        logger.error(
            "Couldn't delete alias %s. Here's why: %s",
            alias,

```



```

        err.response["Error"]["Message"],
    )
    raise

```

Classe e métodos do wrapper para criptografia de chaves do KMS.

```

class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def encrypt(self, key_id: str, text: str) -> bytes:
        """
        Encrypts text by using the specified key.

        :param key_id: The ARN or ID of the key to use for encryption.
        :param text: The text to encrypt.
        :return: The encrypted version of the text.
        """
        try:
            response = self.kms_client.encrypt(KeyId=key_id,
            Plaintext=text.encode())
            print(
                f"The string was encrypted with algorithm
            {response['EncryptionAlgorithm']}"
            )
            return response["CiphertextBlob"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DisabledException":

```

```

        logger.error(
            "Could not encrypt because the key %s is disabled.", key_id
        )
    else:
        logger.error(
            "Couldn't encrypt text. Here's why: %s",
            err.response["Error"]["Message"],
        )
    raise

def decrypt(self, key_id: str, cipher_text: bytes) -> str:
    """
    Decrypts text previously encrypted with a key.

    :param key_id: The ARN or ID of the key used to decrypt the data.
    :param cipher_text: The encrypted text to decrypt.
    :return: The decrypted text.
    """
    try:
        return self.kms_client.decrypt(KeyId=key_id,
            CiphertextBlob=cipher_text)[
            "Plaintext"
        ].decode()
    except ClientError as err:
        logger.error(
            "Couldn't decrypt your ciphertext. Here's why: %s",
            err.response["Error"]["Message"],
        )
    raise

def sign(self, key_id: str, message: str) -> str:
    """
    Signs a message with a key.

    :param key_id: The ARN or ID of the key to use for signing.
    :param message: The message to sign.
    :return: The signature of the message.
    """
    try:
        return self.kms_client.sign(
            KeyId=key_id,
            Message=message.encode(),

```

```

        SigningAlgorithm="RSASSA_PSS_SHA_256",
    )["Signature"]
except ClientError as err:
    logger.error(
        "Couldn't sign your message. Here's why: %s",
        err.response["Error"]["Message"],
    )
    raise

def verify(self, key_id: str, message: str, signature: str) -> bool:
    """
    Verifies a signature against a message.

    :param key_id: The ARN or ID of the key used to sign the message.
    :param message: The message to verify.
    :param signature: The signature to verify.
    :return: True when the signature matches the message, otherwise False.
    """
    try:
        response = self.kms_client.verify(
            KeyId=key_id,
            Message=message.encode(),
            Signature=signature,
            SigningAlgorithm="RSASSA_PSS_SHA_256",
        )
        valid = response["SignatureValid"]
        print(f"The signature is {'valid' if valid else 'invalid'}.")
        return valid
    except ClientError as err:
        if err.response["Error"]["Code"] == "SignatureDoesNotMatchException":
            print("The signature is not valid.")
        else:
            logger.error(
                "Couldn't verify your signature. Here's why: %s",
                err.response["Error"]["Message"],
            )
        raise

```

Classe e métodos do wrapper para concessões de chaves do KMS.

```

class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_grant(
        self, key_id: str, principal: str, operations: [str]
    ) -> dict[str, str]:
        """
        Creates a grant for a key that lets a principal generate a symmetric data
        encryption key.

        :param key_id: The ARN or ID of the key.
        :param principal: The principal to grant permission to.
        :param operations: The operations to grant permission for.
        :return: The grant that is created.
        """
        try:
            return self.kms_client.create_grant(
                KeyId=key_id,
                GranteePrincipal=principal,
                Operations=operations,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create a grant on key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise

```

```
def list_grants(self, key_id):
    """
    Lists grants for a key.

    :param key_id: The ARN or ID of the key to query.
    :return: The grants for the key.
    """
    try:
        paginator = self.kms_client.get_paginator("list_grants")
        grants = []
        page_iterator = paginator.paginate(KeyId=key_id)
        for page in page_iterator:
            grants.extend(page["Grants"])

        print(f"Grants for key {key_id}:")
        pprint(grants)
        return grants
    except ClientError as err:
        logger.error(
            "Couldn't list grants for key %s. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise

def revoke_grant(self, key_id: str, grant_id: str) -> None:
    """
    Revokes a grant so that it can no longer be used.

    :param key_id: The ARN or ID of the key associated with the grant.
    :param grant_id: The ID of the grant to revoke.
    """
    try:
        self.kms_client.revoke_grant(KeyId=key_id, GrantId=grant_id)
    except ClientError as err:
        logger.error(
            "Couldn't revoke grant %s. Here's why: %s",
            grant_id,
            err.response["Error"]["Message"],
        )
        raise
```

Classe e métodos do wrapper para políticas de chaves do 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.")
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Python (Boto3).
 - [CreateAlias](#)
 - [CreateGrant](#)
 - [CreateKey](#)
 - [Decrypt](#)
 - [DescribeKey](#)
 - [DisableKey](#)

- [EnableKey](#)
- [Encrypt](#)
- [GetKeyPolicy](#)
- [ListAliases](#)
- [ListGrants](#)
- [ListKeys](#)
- [RevokeGrant](#)
- [ScheduleKeyDeletion](#)
- [Sign](#)
- [TagResource](#)

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Ações para AWS KMS usar AWS SDKs

Os exemplos de código a seguir demonstram como realizar AWS KMS ações individuais com AWS SDKs. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Esses trechos chamam a AWS KMS API e são trechos de código de programas maiores que devem ser executados em contexto. É possível ver as ações em contexto em [Cenários para AWS KMS usar AWS SDKs](#).

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a [Referência de APIs do AWS Key Management Service](#).

Exemplos

- [Use CreateAlias com um AWS SDK ou CLI](#)
- [Use CreateGrant com um AWS SDK ou CLI](#)
- [Use CreateKey com um AWS SDK ou CLI](#)
- [Use Decrypt com um AWS SDK ou CLI](#)
- [Use DeleteAlias com um AWS SDK ou CLI](#)

- [Use DescribeKey com um AWS SDK ou CLI](#)
- [Use DisableKey com um AWS SDK ou CLI](#)
- [Use EnableKey com um AWS SDK ou CLI](#)
- [Use EnableKeyRotation com um AWS SDK ou CLI](#)
- [Use Encrypt com um AWS SDK ou CLI](#)
- [Use GenerateDataKey com um AWS SDK ou CLI](#)
- [Use GenerateDataKeyWithoutPlaintext com um AWS SDK ou CLI](#)
- [Use GenerateRandom com um AWS SDK ou CLI](#)
- [Use GetKeyPolicy com um AWS SDK ou CLI](#)
- [Use ListAliases com um AWS SDK ou CLI](#)
- [Use ListGrants com um AWS SDK ou CLI](#)
- [Use ListKeyPolicies com um AWS SDK ou CLI](#)
- [Use ListKeys com um AWS SDK ou CLI](#)
- [Use PutKeyPolicy com um AWS SDK ou CLI](#)
- [Use ReEncrypt com um AWS SDK ou CLI](#)
- [Use RetireGrant com um AWS SDK ou CLI](#)
- [Use RevokeGrant com um AWS SDK ou CLI](#)
- [Use ScheduleKeyDeletion com um AWS SDK ou CLI](#)
- [Use Sign com um AWS SDK ou CLI](#)
- [Use TagResource com um AWS SDK ou CLI](#)
- [Use UpdateAlias com um AWS SDK ou CLI](#)
- [Use Verify com um AWS SDK ou CLI](#)

Use **CreateAlias** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `CreateAlias`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

.NET

SDK para .NET

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.");
        }
    }
}
```

- Para obter detalhes da API, consulte [CreateAlias](#) na Referência AWS SDK para .NET da API.

CLI

AWS CLI

Como criar um alias para uma chave do KMS

O comando `create-alias` a seguir cria um alias com o nome `example-alias` para a chave do KMS identificada pelo ID de chave `1234abcd-12ab-34cd-56ef-1234567890ab`.

Os nomes de alias devem começar com `alias/`. Não use nomes de alias que comecem com `alias/aws`; eles são reservados para uso por AWS.

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

Esse comando não retorna resultados. Use o `list-aliases` comando para visualizar o novo alias.

Para obter mais informações, consulte [Using aliases](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [CreateAlias](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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);  
}
```

- Para obter detalhes da API, consulte [CreateAlias](#) a Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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")  
    }  
}
```

- Para obter detalhes da API, consulte a [CreateAlias](#) referência da API AWS SDK for Kotlin.

PHP

SDK para PHP

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Para obter detalhes da API, consulte [CreateAlias](#) a Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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

```

- Para obter detalhes da API, consulte a [CreateAlias](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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.

```


- Para obter detalhes da API, consulte a [CreateAlias](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **CreateGrant** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o CreateGrant.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

.NET

SDK para .NET

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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}");
    }
}

```

- Para obter detalhes da API, consulte [CreateGrant](#) na Referência AWS SDK para .NET da API.

CLI

AWS CLI

Como criar uma concessão

O exemplo de `create-grant` a seguir cria uma concessão que permite que o usuário `exampleUser` use o comando `decrypt` na chave do KMS `1234abcd-12ab-34cd-56ef-1234567890ab` de exemplo. A entidade principal descontinuada é o perfil `adminRole`. A concessão usa a restrição de concessão `EncryptionContextSubset` para conceder essa permissão apenas quando o contexto de criptografia na solicitação `decrypt` incluir o par de chave/valor `"Department": "IT"`.

```

aws kms create-grant \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --grantee-principal arn:aws:iam::123456789012:user/exampleUser \

```

```
--operations Decrypt \
--constraints EncryptionContextSubset={Department=IT} \
--retiring-principal arn:aws:iam::123456789012:role/adminRole
```

Saída:

```
{
  "GrantId":
    "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
  "GrantToken": "<grant token here>"
}
```

Use o `list-grants` comando para visualizar informações detalhadas sobre a concessão.

Para obter mais informações, consulte [Concessões no AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [CreateGrant](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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);
    }

```

- Para obter detalhes da API, consulte [CreateGrant](#) na Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
suspend fun createNewGrant(
    keyIdVal: String?,
    granteePrincipalVal: String?,
    operation: String,
): String? {
    val operationObj = GrantOperation.fromValue(operation)
    val grantOperationList = ArrayList<GrantOperation>()
    grantOperationList.add(operationObj)

    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
    }
}
```

- Para obter detalhes da API, consulte a [CreateGrant](#) referência da API AWS SDK for Kotlin.

PHP

SDK para PHP

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Para obter detalhes da API, consulte [CreateGrant](#) Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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

```

- Para obter detalhes da API, consulte a [CreateGrant](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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.
```

- Para obter detalhes da API, consulte a [CreateGrant](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **CreateKey** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o CreateKey.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- [Conheça os conceitos básicos](#)
- [Trabalhar com a criptografia de tabelas](#)

.NET

SDK para .NET

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.");
        }
    }
}

```

- Para obter detalhes da API, consulte [CreateKey](#) a Referência AWS SDK para .NET da API.

CLI

AWS CLI

Exemplo 1: Para criar uma chave KMS gerenciada pelo cliente no AWS KMS

O exemplo de `create-key` a seguir cria uma chave do KMS de criptografia simétrica.

Não é necessário especificar parâmetros para criar a chave básica do KMS (uma chave de criptografia simétrica). Os valores padrão desses parâmetros criam uma chave de criptografia simétrica.

Como esse comando não especifica uma política de chave, a chave do KMS adota a [política de chave padrão](#) para chaves do KMS criadas programaticamente. Use o comando `get-key-policy` para visualizar a política de chave. Use o comando `put-key-policy` para alterar a política de chave.

```
aws kms create-key
```

O comando `create-key` retorna os metadados da chave, incluindo o ID da chave e o ARN da nova chave do KMS. Você pode usar esses valores para identificar a chave KMS em outras operações do AWS KMS. A saída não inclui as tags. Use `list-resource-tags command` para visualizar as tags de uma chave do KMS.

Saída:

```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": "2017-07-05T14:04:55-07:00",
    "CurrentKeyMaterialId":
    "0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
```

```

        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_KMS"
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}

```

Observação: o comando `create-key` não permite especificar um alias. Use o comando `create-alias` para especificar um alias para a nova chave do KMS.

Para obter mais informações, consulte [Creating keys](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 2: como criar uma chave RSA assimétrica do KMS para criptografia e decodificação

O exemplo de `create-key` a seguir cria uma chave do KMS que contém um par de chaves RSA assimétrico para criptografia e descryptografia. A especificação da chave e o uso da chave não podem ser alterados depois que a chave é criada.

```

aws kms create-key \
  --key-spec RSA_4096 \
  --key-usage ENCRYPT_DECRYPT

```

Saída:

```

{
  "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"
  }
}
```

Para obter mais informações, consulte [Chaves assimétricas no AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

Exemplo 3: como criar uma chave do KMS de curva elíptica assimétrica para assinatura e verificação

Como criar uma chave do KMS assimétrica que contém um par de chaves com curva elíptica (ECC) assimétrica para assinatura e verificação. O parâmetro `--key-usage` é obrigatório, embora `SIGN_VERIFY` seja o único valor válido para chaves ECC do KMS. A especificação da chave e o uso da chave não podem ser alterados depois que a chave é criada.

```
aws kms create-key \
  --key-spec ECC_NIST_P521 \
  --key-usage SIGN_VERIFY
```

Saída:

```
{
  "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"
        ]
    }
}

```

Para obter mais informações, consulte [Chaves assimétricas no AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

Exemplo 4: criar uma chave ML-DSA assimétrica do KMS para assinatura e verificação

Este exemplo cria uma chave do algoritmo de assinatura digital baseado em rede de módulos (ML-DSA) para assinatura e verificação. O parâmetro `key-usage` é obrigatório, embora `SIGN_VERIFY` seja o único valor válido para chaves ML-DSA.

```

aws kms create-key \
  --key-spec ML_DSA_65 \
  --key-usage SIGN_VERIFY

```

Saída:

```

{
  "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"
    ]
  }
}

```

Para obter mais informações, consulte [Chaves assimétricas no AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

Exemplo 5: criar uma chave HMAC do KMS

O exemplo de `create-key` a seguir cria uma chave do KMS com HMAC de 384 bits. O valor `GENERATE_VERIFY_MAC` é obrigatório para o parâmetro `--key-usage`, mesmo que seja o único valor válido para chaves do KMS com HMAC.

```
aws kms create-key \  
  --key-spec HMAC_384 \  
  --key-usage GENERATE_VERIFY_MAC
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Chaves HMAC no AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 6: criar uma chave do KMS primária multirregional

O exemplo de `create-key` a seguir cria uma chave primária de criptografia simétrica multirregional. Como os valores padrão de todos os parâmetros criam uma chave de criptografia simétrica, somente o parâmetro `--multi-region` é necessário para a chave do KMS. Na AWS CLI, para indicar que um parâmetro booleano é verdadeiro, basta especificar o nome do parâmetro.

```
aws kms create-key \  
  --multi-region
```

Saída:

```
{  
  "KeyMetadata": {  
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/  
mrk-1234abcd12ab34cd56ef12345678990ab",  
    "AWSAccountId": "111122223333",  
    "CreationDate": "2021-09-02T016:15:21-09:00",  
    "CurrentKeyMaterialId":  
    "0b7fd7ddb6c6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",  
    "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"  
  }  
}
```



```
}
```

Para obter mais informações, consulte [Chaves assimétricas no AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

Exemplo 7: criar uma chave do KMS para material de chave importada

O exemplo de create-key a seguir cria uma chave do KMS sem material de chave. Quando a operação for concluída, você poderá importar seu próprio material de chave na chave do KMS. Defina o parâmetro --origin como EXTERNAL para criar a chave do KMS.

```
aws kms create-key \  
  --origin EXTERNAL
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Importação de material de chaves em chaves AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 6: Para criar uma chave KMS em um armazenamento de chaves do AWS CloudHSM

O `create-key` exemplo a seguir cria uma chave KMS no armazenamento de chaves especificado do AWS CloudHSM. A operação cria a chave KMS e seus metadados no AWS KMS e cria o material da chave no cluster do CloudHSM associado ao AWS armazenamento de chaves personalizadas. Os parâmetros `--custom-key-store-id` e `--origin` são obrigatórios.

```
aws kms create-key \  
  --origin AWS_CLOUDHSM \  
  --custom-key-store-id cks-1234567890abcdef0
```

Saída:

```
{  
  "KeyMetadata": {  
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
    "AWSAccountId": "111122223333",  
    "CloudHsmClusterId": "cluster-1a23b4cdefg",  
    "CreationDate": "2019-12-02T07:48:55-07:00",  
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",  
    "CustomKeyId": "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"  
  }  
}
```

Para obter mais informações, consulte [AWS CloudHSM key stores](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 8: criar uma chave do KMS em um repositório de chaves externo

O exemplo de `create-key` a seguir cria uma chave do KMS no repositório de chaves externo especificado. Os parâmetros `--custom-key-store-id`, `--origin` e `--xks-key-id` são obrigatórios neste comando.

O parâmetro `--xks-key-id` especifica o ID de uma chave de criptografia simétrica existente no gerenciador de chaves externo. Essa chave serve como o material de chave externo para a chave do KMS. O valor do parâmetro `--origin` deve ser `EXTERNAL_KEY_STORE`. O parâmetro `custom-key-store-id` deve identificar um repositório de chaves externo conectado ao proxy do repositório de chaves externo.

```
aws kms create-key \  
  --origin EXTERNAL_KEY_STORE \  
  --custom-key-store-id cks-9876543210fedcba9 \  
  --xks-key-id bb8562717f809024
```

Saída:

```
{  
  "KeyMetadata": {  
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
    "AWSAccountId": "111122223333",  
    "CreationDate": "2022-12-02T07:48:55-07:00",  
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",  
    "CustomKeyId": "cks-9876543210fedcba9",  
    "Description": "",  
    "Enabled": true,  
    "EncryptionAlgorithms": [  
      "SYMMETRIC_DEFAULT"  
    ],  
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
    "KeyManager": "CUSTOMER",  
    "KeySpec": "SYMMETRIC_DEFAULT",  
    "KeyState": "Enabled",  
    "KeyUsage": "ENCRYPT_DECRYPT",  
    "MultiRegion": false,  
    "Origin": "EXTERNAL_KEY_STORE",  
    "XksKeyConfiguration": {  
      "Id": "bb8562717f809024"  
    }  
  }  
}
```

```
}
```

Para obter mais informações, consulte [External key stores](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [CreateKey](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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);
        });
}
```

- Para obter detalhes da API, consulte [CreateKey](#) a Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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
    }
}
```

- Para obter detalhes da API, consulte a [CreateKey](#) referência da API AWS SDK for Kotlin.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
```

```

    * @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;
        }
    }
}

```

- Para obter detalhes da API, consulte [CreateKey](#) a Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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
```

- Para obter detalhes da API, consulte a [CreateKey](#) Referência da API AWS SDK for Python (Boto3).

Ruby

SDK para Ruby

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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
```

- Para obter detalhes da API, consulte [CreateKey](#) a Referência AWS SDK para Ruby da API.

Rust

SDK para Rust

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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(())  
}
```

- Para obter detalhes da API, consulte a [CreateKey](#) referência da API AWS SDK for Rust.

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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_kmslimitexceededx.
```

```
MESSAGE 'Limit exceeded for KMS resources.' TYPE 'E'.  
ENDTRY.
```

- Para obter detalhes da API, consulte a [CreateKey](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **Decrypt** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o Decrypt.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

CLI

AWS CLI

Exemplo 1: como descriptografar uma mensagem criptografada com uma chave simétrica do KMS (Linux e macOS)

O exemplo de decrypt comando a seguir demonstra a forma recomendada de descriptografar dados com a CLI. AWS Esta versão mostra como descriptografar dados com uma chave simétrica do KMS.

Forneça o texto cifrado em um arquivo. No valor do parâmetro `--ciphertext-blob`, use o prefixo `fileb://`, que instrui a CLI a ler os dados de um arquivo binário. Se o arquivo não estiver no diretório atual, digite o caminho completo para o arquivo. Para obter mais informações sobre a leitura dos valores dos parâmetros da AWS CLI de um arquivo, consulte Carregando parâmetros da AWS CLI de um arquivo < <https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html>> no Guia do usuário da interface de linha de AWS comando e práticas recomendadas para parâmetros de arquivos locais< <https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/>> no blog da

ferramenta de linha de comando. Especifique a chave KMS para descriptografar o texto cifrado. O parâmetro não é necessário ao descriptografar com um AWS KMS simétrico Chave MS. --key-id AWS O KMS pode obter o ID da chave KMS que foi usada para criptografar os dados dos metadados no texto cifrado. Porém, sempre é uma prática recomendada especificar a chave do KMS que você está usando. Esta prática garante que você use a chave do KMS desejada e impede que você descriptografe um texto cifrado acidentalmente usando uma chave do KMS em que você não confia. Solicite a saída de texto simples como um valor de texto. O parâmetro --query instrui a CLI obter somente o valor do campo Plaintext da saída. O parâmetro --output retorna a saída como texto. Decodifique o texto simples em Base64 e salve-o em um arquivo. O seguinte exemplo canaliza (|) o valor do parâmetro Plaintext para o utilitário Base64, que o decodifica. Em seguida, ele redireciona (>) a saída decodificada para o arquivo ExamplePlaintextFile.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
aws kms decrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --output text \
  --query Plaintext | base64 \
  --decode > ExamplePlaintextFile
```

Este comando não produz saída. A saída do comando decrypt é decodificada em base64 e salva em um arquivo.

Para obter mais informações, consulte [Decrypt](#) na Referência da API do AWS Key Management Service.

Exemplo 2: como descriptografar uma mensagem criptografada com uma chave simétrica do KMS (prompt de comando do Windows)

O exemplo a seguir é igual ao anterior, mas ele usa o utilitário certutil para decodificar em Base64 os dados em texto simples. Esse procedimento requer dois comandos, conforme mostrado nos exemplos a seguir.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
aws kms decrypt ^
```

```
--ciphertext-blob fileb://ExampleEncryptedFile ^  
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^  
--output text ^  
--query Plaintext > ExamplePlaintextFile.base64
```

Execute o comando certutil.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Saída:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

Para obter mais informações, consulte [Decrypt](#) na Referência da API do AWS Key Management Service.

Exemplo 3: como descriptografar uma mensagem criptografada com uma chave assimétrica do KMS (Linux e macOS)

O exemplo de comando decrypt a seguir mostra como descriptografar dados criptografados sob uma chave RSA assimétrica do KMS.

Ao usar uma chave assimétrica do KMS, o parâmetro `encryption-algorithm`, que especifica o algoritmo usado para criptografar o texto simples, é necessário.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
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
```

Este comando não produz saída. A saída do comando decrypt é decodificada em base64 e salva em um arquivo.

Para obter mais informações, consulte [Chaves assimétricas no AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

- Consulte detalhes da API em [Decrypt](#) na Referência de comandos da AWS CLI .

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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));
}

```

- Consulte detalhes da API em [Decrypt](#) na Referência da API AWS SDK for Java 2.x .

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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)
    }
}
```

- Consulte detalhes da API em [Decrypt](#) na Referência da API AWS SDK para Kotlin.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}

```

- Consulte detalhes da API em [Decrypt](#) na Referência da API AWS SDK para PHP .

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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

```

- Consulte detalhes da API em [Decrypt](#) na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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

```

- Consulte detalhes da API em [Decrypt](#) na Referência da API AWS SDK para Ruby .

Rust

SDK para Rust

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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(())
}

```

- Consulte detalhes da API em [Descriptografar](#) na Referência da API AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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.
```

- Para obter detalhes da API, consulte a referência da API [Decrypt](#) in AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **DeleteAlias** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o DeleteAlias.

CLI

AWS CLI

Para excluir um alias do AWS KMS

O exemplo de delete-alias a seguir exclui o alias alias/example-alias. O nome de alias deve começar com alias/.

```
aws kms delete-alias \  
    --alias-name alias/example-alias
```

Este comando não produz saída. Use o list-aliases comando para localizar o alias.

Para obter mais informações, consulte [Deleting an alias](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [DeleteAlias](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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);
        });
}
```

- Para obter detalhes da API, consulte [DeleteAlias](#) na Referência AWS SDK for Java 2.x da API.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Para obter detalhes da API, consulte [DeleteAlias](#) Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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
```

- Para obter detalhes da API, consulte a [DeleteAlias](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [DeleteAlias](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **DescribeKey** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o DescribeKey.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

.NET

SDK para .NET

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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}");
    }
}
```

- Para obter detalhes da API, consulte [DescribeKey](#) na Referência AWS SDK para .NET da API.

CLI

AWS CLI

Exemplo 1: como localizar informações detalhadas sobre uma chave do KMS

O `describe-key` exemplo a seguir obtém informações detalhadas sobre a chave AWS gerenciada para o Amazon S3 na conta e região do exemplo. Você pode usar esse comando para encontrar detalhes sobre chaves AWS gerenciadas e chaves gerenciadas pelo cliente.

Use o parâmetro `key-id` para especificar a chave do KMS. Este exemplo usa um valor de nome de alias, mas você pode usar um ID de chave, o ARN da chave, o nome do alias ou o ARN do alias neste comando.

```
aws kms describe-key \
  --key-id alias/aws/s3
```

Saída:

```
{
  "KeyMetadata": {
    "AWSAccountId": "846764612917",
    "KeyId": "b8a9477d-836c-491f-857e-07937918959b",
    "Arn": "arn:aws:kms:us-west-2:846764612917:key/b8a9477d-836c-491f-857e-07937918959b",
    "CurrentKeyMaterialId":
      "0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "CreationDate": 2017-06-30T21:44:32.140000+00:00,
    "Enabled": true,
    "Description": "Default KMS key that protects my S3 objects when no other
key is defined",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "KeyState": "Enabled",
    "Origin": "AWS_KMS",
    "KeyManager": "AWS",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
```

```

        "SYMMETRIC_DEFAULT"
    ]
}
}

```

Para obter mais informações, consulte [Viewing keys](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 2: como obter detalhes sobre uma chave RSA assimétrica do KMS

O exemplo de `describe-key` a seguir obtém informações detalhadas sobre uma chave RSA assimétrica do KMS usada para assinatura e verificação.

```

aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

```

Saída:

```

{
  "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"
    ]
  }
}

```

```
}
```

Exemplo 3: como obter detalhes sobre uma chave de réplica multirregional

O exemplo de `describe-key` a seguir os obtém metadados de uma chave de réplica multirregional. Essa chave multirregional é uma chave de criptografia simétrica. A saída de um comando `describe-key` para qualquer chave multirregional retorna informações sobre a chave primária e todas as suas réplicas.

```
aws kms describe-key \
  --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Saída:

```
{
  "KeyMetadata": {
    "MultiRegion": true,
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "CreationDate": "2021-06-28T21:09:16.114000+00:00",
    "CurrentKeyMaterialId":
    "0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "Description": "",
    "Enabled": true,
    "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "Origin": "AWS_KMS",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "MultiRegionConfiguration": {
      "MultiRegionKeyType": "PRIMARY",
      "PrimaryKey": {
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "us-west-2"
      },
```

```

        "ReplicaKeys": [
            {
                "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "eu-west-1"
            },
            {
                "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "ap-northeast-1"
            },
            {
                "Arn": "arn:aws:kms:sa-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "sa-east-1"
            }
        ]
    }
}

```

Exemplo 4: como obter detalhes sobre uma chave do KMS com HMAC

O exemplo de `describe-key` a seguir obtém informações detalhadas sobre uma chave do KMS com HMAC.

```

aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

```

Saída:

```

{
  "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
}
}

```

- Para obter detalhes da API, consulte [DescribeKey](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

/**
 * 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);
}

```

- Para obter detalhes da API, consulte [DescribeKey](#) a Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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}")
    }
}

```

- Para obter detalhes da API, consulte a [DescribeKey](#) referência da API AWS SDK for Kotlin.

PHP

SDK para PHP

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Para obter detalhes da API, consulte [DescribeKey](#) a Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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
```

- Para obter detalhes da API, consulte a [DescribeKey](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [DescribeKey](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **DisableKey** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DisableKey`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

.NET

SDK para .NET

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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}");
}
}
```

- Para obter detalhes da API, consulte [DisableKey](#) Referência AWS SDK para .NET da API.

CLI

AWS CLI

Como desativar uma chave do KMS temporariamente

O exemplo `disable-key` a seguir desabilita uma chave do KMS gerenciada pelo cliente. Use o `enable-key` comando para reabilitar a chave do KMS.

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

Este comando não produz saída.

Para obter mais informações, consulte [Enabling and Disabling Keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [DisableKey](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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);
        });
}
```

- Para obter detalhes da API, consulte [DisableKey](#) Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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")
    }
}
```

- Para obter detalhes da API, consulte a [DisableKey](#) referência da API AWS SDK for Kotlin.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}

```

- Para obter detalhes da API, consulte [DisableKey](#) a Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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
```

- Para obter detalhes da API, consulte a [DisableKey](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```


- Para obter detalhes da API, consulte a [DisableKey](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **EnableKey** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o EnableKey.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

.NET

SDK para .NET

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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}");
}
}
```

- Para obter detalhes da API, consulte [EnableKey](#) a Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para habilitar uma chave do KMS

O exemplo de `enable-key` a seguir habilita uma chave gerenciada pelo cliente. Você pode usar um comando como esse para habilitar uma chave do KMS que foi temporariamente desabilitada usando o comando `disable-key`. Você também pode usá-lo para habilitar uma chave do KMS que está desabilitada porque foi programada para exclusão, mas a exclusão foi cancelada.

Use o parâmetro `key-id` para especificar a chave do KMS. Este exemplo usa um valor de ID de chave, mas você pode usar um ID de chave ou o valor do ARN da chave neste comando.

Antes de executar esse comando, substitua o ID da chave de exemplo por um ID válido.

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

Este comando não produz saída. Use o `describe-key` comando para verificar se a chave KMS está habilitada. Consulte os valores dos campos `KeyState` e `Enabled` na saída `describe-key`.

Para obter mais informações, consulte [Enabling and Disabling Keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [EnableKey](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**  
 * 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);
}
```

- Para obter detalhes da API, consulte [EnableKey](#) a Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.")
    }
}
```

- Para obter detalhes da API, consulte a [EnableKey](#) referência da API AWS SDK for Kotlin.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Para obter detalhes da API, consulte [EnableKey](#) a Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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
```

- Para obter detalhes da API, consulte a [EnableKey](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [EnableKey](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **EnableKeyRotation** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `EnableKeyRotation`.

CLI

AWS CLI

Para habilitar a alternância automática de uma chave do KMS

O exemplo de `enable-key-rotation` a seguir viabiliza a alternância automática de uma chave do KMS gerenciada pelo cliente com um período de alternância de 180 dias. A chave do KMS será alternada em um ano (aproximadamente 365 dias) a partir da data em que esse comando for concluído e todos os anos a partir de então.

O parâmetro `--key-id` identifica a chave do KMS. Este exemplo usa o valor do ARN da chave, mas é possível usar o ID da chave ou o ARN da chave do KMS. O parâmetro `--rotation-period-in-days` especifica o número de dias entre as datas de cada rotação. Especifique um valor entre 90 e 2.560 dias. Se nenhum valor for especificado, o valor padrão será de 365 dias.

```
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
```

Este comando não produz saída. Use o `get-key-rotation-status` comando para verificar se a chave KMS está habilitada.

Para obter mais informações, consulte [Rotacionar chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [EnableKeyRotation](#) em Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).


```

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

```

- Para obter detalhes da API, consulte a [EnableKeyRotation](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [EnableKeyRotation](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **Encrypt** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o Encrypt.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

CLI

AWS CLI

Exemplo 1: como criptografar o conteúdo de um arquivo no Linux ou no macOS

O `encrypt` comando a seguir demonstra a forma recomendada de criptografar dados com a CLI AWS .

```
aws kms encrypt \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --plaintext fileb://ExamplePlaintextFile \
  --output text \
  --query CiphertextBlob | base64 \
  --decode > ExampleEncryptedFile
```

O comando realiza diversas ações:

Usa o parâmetro `--plaintext` para indicar os dados a serem criptografados. Esse valor de parâmetro deve ser codificado em Base64. O valor do `plaintext` parâmetro deve ser codificado em base64 ou você deve usar o prefixo `fileb://`, que instrui a AWS CLI a ler dados binários do arquivo. Se o arquivo não estiver no diretório atual, digite o caminho completo para o arquivo. Por exemplo: `fileb:///var/tmp/ExamplePlaintextFile` ou `fileb://C:\Temp\ExamplePlaintextFile`. [Para obter mais informações sobre como ler os valores dos parâmetros da AWS CLI de um arquivo, consulte Carregando parâmetros de um arquivo no Guia do usuário da interface de linha de AWS comando e as melhores práticas para parâmetros de arquivos locais no blog da ferramenta de linha de AWS comando. Usa os parâmetros --output e para controlar a saída do comando. Esses --query parâmetros extraem os dados criptografados, chamados de texto cifrado, da saída do comando. Para obter mais informações sobre como controlar a saída, consulte Controlando o comando Saída no Guia do usuário da interface de linha de AWS comando.](#) Usa o `base64` utilitário para decodificar a saída extraída em dados binários. O texto cifrado retornado por um comando bem-sucedido é um texto codificado em base64. `encrypt` Você deve decodificar esse texto antes de poder usar a AWS CLI para descriptografá-lo. Salva o texto cifrado binário em um arquivo. A parte final do comando `> ExampleEncryptedFile ()` salva o texto cifrado binário em um arquivo para facilitar a decodificação. Para ver um exemplo de comando que usa a AWS CLI para descriptografar dados, consulte os exemplos de descriptografia.

Exemplo 2: Usando a AWS CLI para criptografar dados no Windows

Esse exemplo é igual ao exemplo anterior, mas usa a ferramenta `certutil` em vez do `base64`. Esse procedimento requer dois comandos, conforme mostrado no exemplo a seguir.

```
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
```

Exemplo 3: criptografia com uma chave do KMS assimétrica

O comando `encrypt` a seguir mostra como criptografar texto simples com uma chave assimétrica do KMS. O parâmetro `--encryption-algorithm` é obrigatório. Como em todos os comandos da `encrypt` CLI, o `plaintext` parâmetro deve ser codificado em `base64` ou você deve usar o `fileb://` prefixo, que instrui a AWS CLI a ler dados binários do arquivo.

```
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
```

Este comando não produz saída.

- Consulte detalhes da API em [Encrypt](#) na Referência de comandos da AWS CLI .

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
```

```
    * 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);
    }
```

- Consulte detalhes da API em [Encrypt](#) na Referência da API AWS SDK for Java 2.x .

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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)
    }
}
```

- Consulte detalhes da API em [Encrypt](#) na Referência da API AWS SDK para Kotlin.

PHP

SDK para PHP

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Consulte detalhes da API em [Encrypt](#) na Referência da API AWS SDK para PHP .

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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
```

- Consulte detalhes da API em [Encrypt](#) na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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*')

```

- Consulte detalhes da API em [Encrypt](#) na Referência da API AWS SDK para Ruby .

Rust

SDK para Rust

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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()  
}
```

- Consulte detalhes da API em [Criptografar](#) na Referência da API AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte [Criptografar](#) no AWS SDK para referência da API SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **GenerateDataKey** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `GenerateDataKey`.

CLI

AWS CLI

Exemplo 1: como gerar uma chave de dados simétrica de 256 bits

O `generate-data-key` exemplo a seguir solicita uma chave de dados simétrica de 256 bits para uso fora do AWS. O comando retorna uma chave de dados em texto simples para uso e exclusão imediatos e uma cópia da chave de dados criptografada na chave do KMS especificada. A chave de dados criptografada pode ser armazenada com segurança junto com os dados criptografados.

Use o parâmetro `key-spec` com um valor de `AES_256` para solicitar uma chave de dados de 256 bits. Use o parâmetro `key-spec` com um valor de `AES_128` para solicitar uma chave de dados de 128 bits. Para todos os outros comprimentos de chave de dados, use o parâmetro `number-of-bytes`.

A chave do KMS especificada deve ser uma chave de criptografia simétrica do KMS, ou seja, uma chave do KMS com um valor de especificação de chave de `SYMMETRIC_DEFAULT`.

```
aws kms generate-data-key \
  --key-id alias/ExampleAlias \
  --key-spec AES_256
```

Saída:

```
{
  "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",
  "KeyId": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyMaterialId":
    "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
```

```

    "CiphertextBlob":
      "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
      +YdhV8MrkBQPeac0ReRVNDt9qlAt+SHgIRF8P0H+7U="
  }

```

A Plaintext (chave de dados em texto simples) e a CiphertextBlob (chave de dados criptografada) são retornadas no formato codificado em base64.

Para obter mais informações, consulte [Data keys](#) no Guia do desenvolvedor do AWS Key Management Service. Exemplo 2: como gerar uma chave de dados simétrica de 512 bits

O exemplo de generate-data-key a seguir solicita uma chave de dados simétrica de 512 bits para criptografia e descriptografia. O comando retorna uma chave de dados em texto simples para uso e exclusão imediatos e uma cópia da chave de dados criptografada na chave do KMS especificada. A chave de dados criptografada pode ser armazenada com segurança junto com os dados criptografados.

Use o parâmetro number-of-bytes para solicitar um comprimento de chave diferente de 128 ou 256 bits. Para solicitar uma chave de dados de 512 bits, o exemplo a seguir usa o parâmetro number-of-bytes com um valor de 64 (bytes).

A chave do KMS especificada deve ser uma chave de criptografia simétrica do KMS, ou seja, uma chave do KMS com um valor de especificação de chave de SYMMETRIC_DEFAULT.

OBSERVAÇÃO: os valores na saída deste exemplo são truncados para exibição.

```

aws kms generate-data-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --number-of-bytes 64

```

Saída:

```

{
  "CiphertextBlob": "AQIBAHi6LtupRpdKl2aJTzkk6Fbh0tQkMlQJJH3PdtHvS/y+hAEnX/
  QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSIb3DQEHBqCCA1UwggJRAgEAMIICSgYJKoZ...",
  "Plaintext": "ty8Lr0Bk60F07M2Bwt6qbFdNB
  +G00ZLtf5MSEb4a13R2UKWGOp06njAwy2n72VRm2m7z/
  Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm40ovEXiVfBveyX3DQxDzRSwbKDPk/...",
  "KeyId": "arn:aws:kms:us-
  west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyMaterialId":
    "0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}

```

```
}
```

Plaintext (chave de dados em texto simples) e CiphertextBlob (chave de dados criptografada) são retornadas no formato codificado em base64.

Para obter mais informações, consulte [Data keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [GenerateDataKey](#) em Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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)

```

- Para obter detalhes da API, consulte a [GenerateDataKey](#) Referência da API AWS SDK for Python (Boto3).

Rust

SDK para Rust

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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(())
}
```

- Para obter detalhes da API, consulte a [GenerateDataKey](#) referência da API AWS SDK for Rust.

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [GenerateDataKey](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **GenerateDataKeyWithoutPlaintext** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `GenerateDataKeyWithoutPlaintext`.

CLI

AWS CLI

Como gerar uma chave de dados simétrica de 256 bits sem uma chave de texto simples

O exemplo de `generate-data-key-without-plaintext` a seguir solicita uma cópia criptografada de uma chave de dados simétrica de 256 bits para uso externo à AWS. Você pode chamar o AWS KMS para descriptografar a chave de dados quando estiver pronto para usá-la.

Use o parâmetro `key-spec` com um valor de `AES_256` para solicitar uma chave de dados de 256 bits. Use o parâmetro `key-spec` com um valor de `AES_128` para solicitar uma chave de dados de 128 bits. Para todos os outros comprimentos de chave de dados, use o parâmetro `number-of-bytes`.

A chave do KMS especificada deve ser uma chave de criptografia simétrica do KMS, ou seja, uma chave do KMS com um valor de especificação de chave de `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
```

Saída:

```
{
  "CiphertextBlob":
  "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
  "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyMaterialId":
  "0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

A `CiphertextBlob` (chave de dados criptografada) é retornada em um formato codificado em base64.

Para obter mais informações, consulte [Data keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [GenerateDataKeyWithoutPlaintext](#) em Referência de AWS CLI Comandos.

Rust

SDK para Rust

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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::<(),  
}
```

- Para obter detalhes da API, consulte a [GenerateDataKeyWithoutPlaintext](#) referência da API AWS SDK for Rust.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **GenerateRandom** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `GenerateRandom`.

CLI

AWS CLI

Exemplo 1: como gerar uma string de bytes aleatória de 256 bits (Linux ou macOS)

O exemplo de `generate-random` a seguir gera uma string de bytes aleatórios de 256 bits (32 bytes) codificada em base64. O exemplo decodifica a string de bytes e a salva no arquivo aleatório.

Ao executar esse comando, use o parâmetro `number-of-bytes` para especificar o tamanho do valor aleatório em bytes.

Você não especifica uma chave do KMS ao executar esse comando. A sequência de bytes aleatória não está relacionada a nenhuma chave do KMS.

Por padrão, o AWS KMS gera o número aleatório. No entanto, se você especificar um [armazenamento de chaves personalizadas](#), a sequência de bytes aleatória será gerada no cluster do AWS CloudHSM associado ao armazenamento de chaves personalizadas.

Este exemplo usa os seguintes parâmetros e valores:

Ele usa o `--number-of-bytes` parâmetro necessário com um valor de 32 para solicitar uma string de 32 bytes (256 bits). Ele usa o `--output` parâmetro com um valor de `text` para direcionar a AWS CLI a retornar a saída como texto, em vez de JSON. Ele usa o `Plaintext` para extrair o valor da propriedade da resposta. Ele canaliza (`|`) a saída do comando `--query Plaintext` para o `base64` utilitário, que decodifica a saída extraída. Ele usa o operador de redirecionamento (`>`) para salvar a string de bytes decodificada no arquivo. Ele usa o operador de redirecionamento (`>`) `ExampleRandom` para salvar o texto cifrado binário em um arquivo.

```
aws kms generate-random \
  --number-of-bytes 32 \
  --output text \
  --query Plaintext | base64 --decode > ExampleRandom
```

Este comando não produz saída.

Para obter mais informações, consulte [GenerateRandom](#) a Referência da API do AWS Key Management Service.

Exemplo 2: como gerar um número aleatório de 256 bits (prompt de comando do Windows)

O exemplo a seguir usa o comando `generate-random` para gerar uma string aleatória de 256 bits (32 bytes) codificada em base64. O exemplo decodifica a string de bytes e a salva no arquivo aleatório. Esse exemplo é igual ao exemplo anterior, mas usa o utilitário `certutil` no Windows para decodificar em base64 a string de bytes aleatória antes de salvá-la em um arquivo.

Primeiro, gere uma string de bytes aleatória codificada em base64 e salve-a em um arquivo temporário (`ExampleRandom.base64`).

```
aws kms generate-random \
  --number-of-bytes 32 \
  --output text \
  --query Plaintext > ExampleRandom.base64
```

Como a saída do comando `generate-random` é salva em um arquivo, esse exemplo não produz nenhuma saída.

Use o comando `certutil -decode` para decodificar a string de bytes codificada em base64 no arquivo `ExampleRandom.base64`. Em seguida, ele salva a string de bytes decodificada no arquivo `ExampleRandom`.

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

Saída:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

Para obter mais informações, consulte [GenerateRandom](#) a Referência da API do AWS Key Management Service.

- Para obter detalhes da API, consulte [GenerateRandom](#) em Referência de AWS CLI Comandos.

Rust

SDK para Rust

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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(())  
}
```

- Para obter detalhes da API, consulte a [GenerateRandom](#) referência da API AWS SDK for Rust.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **GetKeyPolicy** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `GetKeyPolicy`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

CLI

AWS CLI

Como copiar uma política de chave de uma chave do KMS para outra chave do KMS

O exemplo de `get-key-policy` a seguir obtém a política de chave de uma chave do KMS e a salva em um arquivo de texto. Em seguida, ele substitui a política de uma chave diferente do KMS usando o arquivo de texto como a entrada da política.

Como o parâmetro `--policy` de `put-key-policy` requer uma string, você deve usar a opção `--output text` para retornar a saída como uma string de texto em vez de 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 \  
    --policy policy.txt
```

```
--key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
--policy file://policy.txt
```

Este comando não produz saída.

Para obter mais informações, consulte [PutKeyPolicy](#) a Referência da API AWS KMS.

- Para obter detalhes da API, consulte [GetKeyPolicy](#) em Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.")
```

- Para obter detalhes da API, consulte a [GetKeyPolicy](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [GetKeyPolicy](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListAliases** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListAliases`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

.NET

SDK para .NET

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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);
        }
    }

```

- Para obter detalhes da API, consulte [ListAliases](#) a Referência AWS SDK para .NET da API.

CLI

AWS CLI

Exemplo 1: Para listar todos os aliases em uma AWS conta e região

O exemplo a seguir usa o `list-aliases` comando para listar todos os aliases na região padrão da AWS conta. O resultado inclui aliases associados às chaves KMS AWS gerenciadas e às chaves KMS gerenciadas pelo cliente.

```
aws kms list-aliases
```

Saída:

```
{
  "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"
    },
    ...
  ]
}
```

Exemplo 2: como listar todos os aliases de uma chave específica do KMS

O exemplo a seguir usa o comando `list-aliases` e seu parâmetro `key-id` para listar todos os aliases associados a uma determinada chave do KMS.

Cada alias é associado a apenas um chave do KMS, mas uma chave do KMS pode ter vários aliases. Esse comando é muito útil porque o console AWS KMS lista somente um alias para cada chave KMS. Você deve usar o comando `list-aliases` para localizar todos os aliases de uma chave do KMS.

Este exemplo usa o ID de chave da chave do KMS no parâmetro `--key-id`, mas você pode usar um ID de chave, o ARN da chave, o nome do alias ou o ARN do alias neste comando.

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

Saída:

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

Para obter mais informações, consulte [Working with aliases](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [ListAliases](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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());
            }
        });
    }
}
```

- Para obter detalhes da API, consulte [ListAliases](#) na Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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}")
        }
    }
}
```

- Para obter detalhes da API, consulte a [ListAliases](#) referência da API AWS SDK for Kotlin.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}

```

- Para obter detalhes da API, consulte [ListAliases](#) a Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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
```

- Para obter detalhes da API, consulte a [ListAliases](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

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.
```

- Para obter detalhes da API, consulte a [ListAliases](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListGrants** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o ListGrants.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

.NET

SDK para .NET

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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);
        }
    }
}

```

- Para obter detalhes da API, consulte [ListGrants](#) na Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para visualizar as concessões em uma chave AWS KMS

O `list-grants` exemplo a seguir exibe todas as concessões na chave KMS AWS gerenciada especificada para o Amazon DynamoDB em sua conta. Essa concessão permite que o DynamoDB use a chave do KMS em seu nome para criptografar uma tabela do DynamoDB antes de gravá-la no disco. Você pode usar um comando como esse para visualizar as concessões nas chaves KMS AWS gerenciadas e nas chaves KMS gerenciadas pelo cliente na AWS conta e na região.

Este comando usa o parâmetro `key-id` com um ID de chave para identificar a chave do KMS. Você pode usar um ID de chave ou o ARN da chave para identificar a chave do KMS. Para obter o ID da chave ou o ARN da chave de uma chave KMS AWS gerenciada, use o `list-keys` comando ou `list-aliases`

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

O resultado mostra que a concessão fornece ao Amazon DynamoDB permissão para usar a chave do KMS para operações criptográficas e permite que ele visualize detalhes sobre a chave do KMS (`DescribeKey`) e descontinue concessões (`RetireGrant`). A restrição `EncryptionContextSubset` limita a permissão às solicitações que incluem os pares de contexto de criptografia especificados. Como resultado, as permissões na concessão são efetivas somente na conta especificada e na tabela do 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"
  }
]
}

```

Para obter mais informações, consulte [Concessões no AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [ListGrants](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

/**
 * 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);
            }
        });
    }
}

```

- Para obter detalhes da API, consulte [ListGrants](#) na Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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}")
        }
    }
}
```

- Para obter detalhes da API, consulte a [ListGrants](#) referência da API AWS SDK for Kotlin.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Para obter detalhes da API, consulte [ListGrants](#) na Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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

```

- Para obter detalhes da API, consulte a [ListGrants](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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.

```

- Para obter detalhes da API, consulte a [ListGrants](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListKeyPolicies** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListKeyPolicies`.

CLI

AWS CLI

Como obter os nomes das políticas de chave de uma chave do KMS

O exemplo de `list-key-policies` a seguir obtém os nomes das políticas de chave de uma chave gerenciada pelo cliente na conta e região de exemplo. Você pode usar esse comando para encontrar os nomes das políticas de chaves para chaves AWS gerenciadas e chaves gerenciadas pelo cliente.

Como o único nome de política de chave válido é `default`, esse comando não é muito útil.

Use o parâmetro `key-id` para especificar a chave do KMS. Este exemplo usa um valor de ID de chave, mas você pode usar um ID de chave ou o ARN da chave neste comando.

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

Saída:

```
{
  "PolicyNames": [
    "default"
  ]
}
```

Para obter mais informações sobre as políticas de chaves do AWS KMS, consulte [Usando políticas de chaves no AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [ListKeyPolicies](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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);
        });
}
```

- Para obter detalhes da API, consulte [ListKeyPolicies](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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)
```

- Para obter detalhes da API, consulte a [ListKeyPolicies](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [ListKeyPolicies](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListKeys** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListKeys`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

.NET

SDK para .NET

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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);
}
}
```

- Para obter detalhes da API, consulte [ListKeys](#) a Referência AWS SDK para .NET da API.

CLI

AWS CLI

Como obter chaves do KMS em uma conta e região

O exemplo `list-keys` a seguir obtém as chaves do KMS em uma conta e região. Esse comando retorna as chaves AWS gerenciadas e as chaves gerenciadas pelo cliente.

```
aws kms list-keys
```

Saída:

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

Para obter mais informações, consulte [Viewing Keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [ListKeys](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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());
    }
}
}

```

- Para obter detalhes da API, consulte [ListKeys](#) na Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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}")
        }
    }
}
```

- Para obter detalhes da API, consulte a [ListKeys](#) referência da API AWS SDK for Kotlin.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
```

```

    * @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;
        }
    }
}

```

- Para obter detalhes da API, consulte [ListKeys](#) na Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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"],
        )

```

- Para obter detalhes da API, consulte a [ListKeys](#) Referência da API AWS SDK for Python (Boto3).

Rust

SDK para Rust

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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(())
}
```

- Para obter detalhes da API, consulte a [ListKeys](#) referência da API AWS SDK for Rust.

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [ListKeys](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **PutKeyPolicy** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o PutKeyPolicy.

CLI

AWS CLI

Como alterar a política de chaves de uma chave do KMS

O exemplo de put-key-policy a seguir altera a política de chave de uma chave gerenciada pelo cliente.

Para começar, crie uma política de chave e salve-a em um arquivo JSON local. Neste exemplo, o arquivo é key_policy.json. Você também pode especificar a política de chave como o valor da string do parâmetro policy.

A primeira declaração nessa política de chaves dá à AWS conta permissão para usar políticas do IAM para controlar o acesso à chave KMS. A segunda instrução fornece permissão de usuário ao test-user para executar os comandos describe-key e list-keys na chave do KMS.

Conteúdo de 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" : "*"
  }
]
```

Este exemplo usa o ID de chave para identificar a chave do KMS, mas também é possível usar um ARN de chave. O comando usa o parâmetro `policy` para especificar a política de chave. Para indicar que a política está em um arquivo, ela usa o prefixo `file://` necessário. Esse prefixo é necessário para identificar arquivos em todos os sistemas operacionais compatíveis. Por fim, o comando usa o parâmetro `policy-name` com um valor `default`. Se não for especificado um nome de política, o valor padrão será `default`. O único valor válido é `default`.

```
aws kms put-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --policy file://key_policy.json
```

Esse comando não produz nenhuma saída. Use o comando `get-key-policy` para verificar se o comando está em vigor. O comando de exemplo a seguir recupera a política de chave da

mesma chave do KMS. O parâmetro `output` com o valor `text` retorna um formato de texto fácil de ler.

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

Saída:

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

Para obter mais informações, consulte [Changing a Key Policy](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [PutKeyPolicy](#) em Referência de AWS CLI Comandos.

PHP

SDK para PHP

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Para obter detalhes da API, consulte [PutKeyPolicy](#) a Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.")

```

- Para obter detalhes da API, consulte a [PutKeyPolicy](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [PutKeyPolicy](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ReEncrypt** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o ReEncrypt.

CLI

AWS CLI

Exemplo 1: como criptografar novamente uma mensagem criptografada com uma chave simétrica diferente do KMS (Linux e macOS).

O exemplo de `re-encrypt` comando a seguir demonstra a forma recomendada de recriptografar dados com a CLI. AWS

Forneça o texto cifrado em um arquivo. No valor do parâmetro `--ciphertext-blob`, use o prefixo `fileb://`, que instrui a CLI a ler os dados de um arquivo binário. Se o arquivo não estiver no diretório atual, digite o caminho completo para o arquivo. Para obter mais informações sobre a leitura dos valores dos parâmetros da AWS CLI de um arquivo, consulte Carregando [parâmetros da AWS CLI de um arquivo no Guia do](#) usuário da interface de linha de AWS comando e [as melhores práticas para parâmetros de arquivos locais](#) no blog da ferramenta de linha de AWS comando. Especifique a chave KMS de origem, que descriptografa o texto cifrado. O parâmetro não é necessário ao descriptografar com chaves KMS de criptografia simétrica. `--source-key-id` AWS O KMS pode obter a chave KMS usada para criptografar os dados dos metadados no blob de texto cifrado. Porém, sempre é uma prática recomendada especificar a chave do KMS que você está usando. Essa prática garante que você use a chave do KMS desejada e impede que você descriptografe um texto cifrado acidentalmente usando uma chave do KMS em que você não confia. Especifique a chave do KMS de destino, que criptografa os dados novamente. O parâmetro `--destination-key-id` é sempre obrigatório. Este exemplo usa um ARN de chave, mas você pode usar qualquer identificador de chave válido. Solicite a saída de texto simples como um valor de texto. O parâmetro `--query` informa à CLI para obter somente o valor do campo `Plaintext` da saída. O parâmetro `--output` retorna a saída como texto. Decodifique o texto simples em Base64 e salve-o em um arquivo. O seguinte exemplo canaliza (`|`) o valor do parâmetro `Plaintext` para o utilitário Base64, que o decodifica. Em seguida, ele redireciona (`>`) a saída decodificada para o arquivo `ExamplePlaintext`.

Antes de executar esse comando, substitua a chave IDs de exemplo por identificadores de chave válidos da sua AWS conta.

```
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
```

Este comando não produz saída. A saída do comando `re-encrypt` é decodificada em base64 e salva em um arquivo.

Para obter mais informações, consulte [ReEncrypta](#) Referência da API do AWS Key Management Service.

Exemplo 2: como criptografar novamente uma mensagem criptografada com uma chave simétrica diferente do KMS (prompt de comando do Windows).

O exemplo de comando `re-encrypt` a seguir é o mesmo que o anterior, mas ele usa o utilitário `certutil` para decodificar em Base64 os dados em texto simples. Esse procedimento requer dois comandos, conforme mostrado nos exemplos a seguir.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
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
```

Em seguida, use o utilitário `certutil`.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Saída:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

Para obter mais informações, consulte [ReEncrypta](#) Referência da API do AWS Key Management Service.

- Para obter detalhes da API, consulte [ReEncryptem](#) Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.")

```

- Para obter detalhes da API, consulte a [ReEncrypt](#) Referência da API AWS SDK for Python (Boto3).

Ruby

SDK para Ruby

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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*')
```

- Para obter detalhes da API, consulte [ReEncrypta](#) Referência AWS SDK para Ruby da API.

Rust

SDK para Rust

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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(())
}

```

- Para obter detalhes da API, consulte a [ReEncrypt](#) referência da API AWS SDK for Rust.

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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.

```

- Para obter detalhes da API, consulte a [ReEncrypt](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **RetireGrant** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o RetireGrant.

CLI

AWS CLI

Como descontinuar uma concessão em uma chave mestra do cliente

O exemplo de `retire-grant` a seguir exclui a concessão de uma chave do KMS.

O comando de exemplo a seguir especifica os parâmetros `grant-id` e `key-id`. O valor do parâmetro `key-id` deve ser o ARN de chave da chave do KMS.

```
aws kms retire-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando não produz saída. Use o comando `list-grants` para confirmar a descontinuação da concessão.

Para obter mais informações, consulte [Retiring and revoking grants](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [RetireGrant](#) em Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.")
```

- Para obter detalhes da API, consulte a [RetireGrant](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte a [RetireGrant](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **RevokeGrant** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `RevokeGrant`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

CLI

AWS CLI

Como revogar uma concessão em uma chave mestra do cliente

O exemplo de `revoke-grant` a seguir exclui a concessão de uma chave do KMS. O comando de exemplo a seguir especifica os parâmetros `grant-id` e `key-id`. O valor do parâmetro `key-id` pode ser o ID ou o ARN de chave da chave do KMS.

```
aws kms revoke-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando não produz saída. Use o comando `list-grants` para confirmar que a concessão foi revogada.

Para obter mais informações, consulte [Retiring and revoking grants](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [RevokeGrant](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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;  
}
```

- Para obter detalhes da API, consulte [RevokeGrant](#) Referência AWS SDK for Java 2.x da API.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**  
 * @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;
        }
    }
}

```

- Para obter detalhes da API, consulte [RevokeGrant](#) a Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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

```

- Para obter detalhes da API, consulte a [RevokeGrant](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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.
```

- Para obter detalhes da API, consulte a [RevokeGrant](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ScheduleKeyDeletion** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ScheduleKeyDeletion`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

CLI

AWS CLI

Como programar a exclusão de uma chave do KMS gerenciada pelo cliente.

O exemplo de `schedule-key-deletion` a seguir programa a chave do KMS gerenciada pelo cliente especificada para ser excluída em 15 dias.

O parâmetro `--key-id` identifica a chave do KMS. Este exemplo usa o valor do ARN da chave, mas você pode usar o ID da chave ou o ARN da chave do KMS. O parâmetro `--pending-window-in-days` especifica a duração do período de espera (7 a 30 dias). Por padrão, o período de espera é de 30 dias. Este exemplo especifica um valor de 15, que indica AWS a exclusão permanente da chave KMS 15 dias após a conclusão do comando.

```
aws kms schedule-key-deletion \  
    --key-id arn:aws:kms:us-  
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
    --pending-window-in-days 15
```

```
--pending-window-in-days 15
```

A resposta inclui o ARN da chave, o estado da chave, o período de espera (PendingWindowInDays) e a data de exclusão no horário Unix. Para ver a data de exclusão no horário local, use o console AWS KMS. As chaves do KMS no estado PendingDeletion não podem ser usadas em operações criptográficas.

```
{
  "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
}
```

Para obter mais informações, consulte [Deleting keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [ScheduleKeyDeletion](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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);
            });
    }

```

- Para obter detalhes da API, consulte [ScheduleKeyDeletion](#) na Referência AWS SDK for Java 2.x da API.

PHP

SDK para PHP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

/***
 * @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;
        }
    }
}

```

- Para obter detalhes da API, consulte [ScheduleKeyDeletion](#) na Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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

```

- Para obter detalhes da API, consulte a [ScheduleKeyDeletion](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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_pendingwindowdays = 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.
```

- Para obter detalhes da API, consulte a [ScheduleKeyDeletion](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **Sign** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o Sign.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

CLI

AWS CLI

Exemplo 1: gerar uma assinatura digital para uma mensagem

O exemplo de `sign` a seguir gera uma assinatura criptográfica para uma mensagem curta. A saída do comando inclui um campo `Signature` codificado em base 64 que pode ser verificado usando o comando `verify`.

É necessário especificar uma mensagem para assinar e um algoritmo de assinatura compatível com sua chave do KMS assimétrica. Para obter os algoritmos de assinatura para sua chave do KMS, use o comando `describe-key`.

Na AWS CLI v2, o valor do `message` parâmetro deve ser codificado em Base64. Ou você pode salvar a mensagem em um arquivo e usar o `fileb://` prefixo, que AWS instrui a CLI a ler dados binários do arquivo.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta. O ID da chave deverá representar uma chave assimétrica do KMS com o uso da chave `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
```

Saída:

```
{
  "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Signature": "ABCDEFhpyVYyTxbafe74ccSvEJLJr3zuoV1Hfymz4qv+/fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```


Para obter mais informações sobre o uso de chaves KMS assimétricas no AWS KMS, consulte Chaves [assimétricas no KMS no Guia do desenvolvedor do AWS Key Management Service](#).AWS

Exemplo 2: salvar uma assinatura digital em um arquivo (Linux e macOS)

O exemplo de `sign` a seguir gera uma assinatura criptográfica para uma mensagem curta armazenada em um arquivo local. O comando também obtém a `Signature` propriedade da resposta, decodifica em Base64 e a salva no arquivo. `ExampleSignature` É possível usar o arquivo de assinatura em um comando `verify` que verifique a assinatura.

O comando `sign` requer uma mensagem codificada em base64 e um algoritmo de assinatura compatível com sua chave assimétrica do KMS. Para obter os algoritmos de assinatura compatíveis com sua chave do KMS, use o comando `describe-key`.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta. O ID da chave deve representar uma chave KMS assimétrica com o uso da chave `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
```

Este comando não produz saída. Este exemplo extrai a propriedade `Signature` da saída e a salva em um arquivo.

Para obter mais informações sobre o uso de chaves KMS assimétricas no AWS KMS, consulte Chaves [assimétricas no KMS no Guia do desenvolvedor do AWS Key Management Service](#).AWS

- Consulte detalhes da API em [Sign](#) na Referência de comandos da AWS CLI .

Java

SDK para Java 2.x

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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);
    });
}

```

- Consulte detalhes da API em [Sign](#) na Referência da API do AWS SDK for Java 2.x .

PHP

SDK para PHP

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Consulte detalhes da API em [Sign](#) na Referência da API do AWS SDK para PHP .

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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

```

- Consulte detalhes da API em [Sign](#) na Referência API do AWS SDK para Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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.

```

- Para obter detalhes da API, consulte a referência da API [Sign](#) in AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **TagResource** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o TagResource.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Conheça os conceitos básicos](#)

CLI

AWS CLI

Para adicionar uma tag a uma chave do KMS

O exemplo de tag-resource a seguir adiciona as tags "Purpose": "Test" e "Dept": "IT" a uma chave do KMS gerenciada pelo cliente. É possível usar tags como essas para identificar chaves do KMS e criar categorias de chaves do KMS para permissões e auditoria.

Use o parâmetro key-id para especificar a chave do KMS. Este exemplo usa um valor de ID de chave, mas você pode usar um ID de chave ou o ARN da chave neste comando.

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

Este comando não produz saída. Para visualizar as tags em uma chave AWS KMS KMS, use o list-resource-tags comando.

Para obter mais informações sobre o uso de tags no AWS KMS, consulte Como [marcar chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [TagResource](#) em Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * 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);
        });
}
```

- Para obter detalhes da API, consulte [TagResource](#) a Referência AWS SDK for Java 2.x da API.

PHP

SDK para PHP

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
/**
 * @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;
    }
}
```

- Para obter detalhes da API, consulte [TagResource](#) na Referência AWS SDK para PHP da API.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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

```

- Para obter detalhes da API, consulte a [TagResource](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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.
```

- Para obter detalhes da API, consulte a [TagResource](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **UpdateAlias** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o UpdateAlias.

CLI

AWS CLI

Como associar um alias a uma chave do KMS diferente

O exemplo de update-alias a seguir associa o alias alias/test-key a uma chave do KMS diferente.

O parâmetro --alias-name especifica o alias. O valor do nome do alias deve começar com alias/. O parâmetro --target-key-id especifica a chave do KMS que deve ser associada ao alias. Não é necessário especificar a chave do KMS atual para o alias.

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

Este comando não produz saída. Use o list-aliases comando para localizar o alias.

Para obter mais informações, consulte [Updating aliases](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para obter detalhes da API, consulte [UpdateAlias](#) em Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.")

```

- Para obter detalhes da API, consulte a [UpdateAlias](#) Referência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```

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.
```

- Para obter detalhes da API, consulte a [UpdateAlias](#) referência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **Verify** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o **Verify**.

CLI

AWS CLI

Para verificar uma assinatura digital

O comando `verify` a seguir verifica uma assinatura criptográfica para uma mensagem curta codificada em base64. O ID da chave, a mensagem, o tipo de mensagem e o algoritmo de assinatura devem ser os mesmos usados para assinar a mensagem.

Na AWS CLI v2, o valor do `message` parâmetro deve ser codificado em Base64. Ou você pode salvar a mensagem em um arquivo e usar o `fileb://` prefixo, que AWS instrui a CLI a ler dados binários do arquivo.

A assinatura que especificada não pode ter codificação base64. Para obter ajuda na decodificação da assinatura retornada pelo comando `sign`, consulte os exemplos do comando `sign`.

A saída do comando inclui um campo booleano `SignatureValid` que indica que a assinatura foi verificada. Se a validação da assinatura falhar, o comando `verify` também falhará.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
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
```

Saída:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "SignatureValid": true,
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Para obter mais informações sobre o uso de chaves KMS assimétricas no AWS KMS, consulte [Usando chaves assimétricas no Guia do desenvolvedor do AWS Key Management Service](#).

- Consulte detalhes da API em [Verify](#) na Referência de comandos da AWS CLI .

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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

```

- Consulte detalhes da API em [Verify](#) na Referência da API do AWS SDK para Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

```
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.
```

- Para obter detalhes da API, consulte [Verificar](#) no AWS SDK para referência da API SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários para AWS KMS usar AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns em AWS KMS with AWS SDKs. Esses cenários mostram como realizar tarefas específicas chamando várias funções internas AWS KMS ou combinadas com outras Serviços da AWS. Cada cenário inclui um link para o código-fonte completo, onde podem ser encontradas instruções sobre como configurar e executar o código.

Os cenários têm como alvo um nível intermediário de experiência para ajudar você a compreender ações de serviço em contexto.

Exemplos

- [Trabalhe com a criptografia de tabelas do DynamoDB usando a v2 AWS Command Line Interface](#)

Trabalhe com a criptografia de tabelas do DynamoDB usando a v2 AWS Command Line Interface

O exemplo de código a seguir mostra como gerenciar opções de criptografia para tabelas do DynamoDB.

- Crie uma tabela com a criptografia padrão.
- Crie uma tabela com uma chave gerenciada pelo cliente (CMK).
- Atualize as configurações de criptografia da tabela.
- Descreva a criptografia da tabela.

Bash

AWS CLI com script Bash

Crie uma tabela com a criptografia padrão.

```
# 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
```

Crie uma tabela com uma chave gerenciada pelo cliente (CMK).

```
# Step 1: Create a customer managed key in KMS  
aws kms create-key \  
  --description "Key for DynamoDB table encryption" \  
  --key-usage ENCRYPT_DECRYPT \  
  --customer-master-key-spec SYMMETRIC_DEFAULT  
  
# Store the key ID for later use  
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for  
DynamoDB')].KeyId" --output text)  
  
# Step 2: Create a table with the customer managed key  
aws dynamodb create-table \  
  --table-name SensitiveData \  
  --attribute-definitions \  
    AttributeName=RecordID,AttributeType=S \  
  --key-schema \  
    AttributeName=RecordID,KeyType=HASH \  
  --billing-mode PAY_PER_REQUEST \  
  --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

Atualize a criptografia da tabela.

```
# 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
```

Descreva a criptografia da tabela.

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
  --table-name CustomerData \
  --query "Table.SSEDescription"
```

- Consulte detalhes da API nos tópicos a seguir na Referência de comandos da AWS CLI .
 - [CreateKey](#)
 - [CreateTable](#)
 - [DescribeTable](#)
 - [UpdateTable](#)

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Usando esse serviço com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Suporte de atestado criptográfico em AWS KMS

AWS KMS [suporta atestado criptográfico para AWS Nitro Enclaves e NitroTPM.AWS](#) Os aplicativos que oferecem suporte a esses métodos de atestado chamam as seguintes operações AWS KMS criptográficas com um documento de atestado assinado. AWS KMS verifica se o documento de atestação veio de uma fonte válida (um enclave Nitro ou NitroTPM). Depois, em vez de retornar dados em texto sem formatação na resposta, essas APIs criptografam o texto sem formatação com a chave pública do atestado e retornam texto cifrado que somente pode ser descriptografado pela chave privada correspondente no enclave ou na instância do EC2.

- [Decrypt](#)
- [DeriveSharedSecret](#)
- [GenerateDataKey](#)
- [GenerateDataKeyPair](#)
- [GenerateRandom](#)

A tabela a seguir mostra como a resposta a solicitações atestadas diferem da resposta padrão para cada operação de API.

AWS KMS operação	Resposta padrão	Resposta para solicitações atestadas
Decrypt	Retorna dados de texto simples	Retorna dados de texto simples criptografados pela chave pública do documento de atestado
DeriveSharedSecret	Retornar segredo compartilhado bruto	Retorna o segredo compartilhado bruto criptografado pela chave pública do documento de atestado
GenerateDataKey	Retorna uma cópia em texto simples da chave de dados	Retorna uma cópia da chave de dados criptografada pela chave pública do documento de atestado

AWS KMS operação	Resposta padrão	Resposta para solicitações atestadas
	(Também retorna uma cópia da chave de dados criptografada por uma chave do KMS)	(Também retorna uma cópia da chave de dados criptografada por uma chave do KMS)
GenerateDataKeyPair	Retorna uma cópia em texto simples da chave privada (Também retorna a chave pública e uma cópia da chave privada criptografada por uma chave do KMS)	Retorna uma cópia da chave privada criptografada pela chave pública do documento de atestado (Também retorna a chave pública e uma cópia da chave privada criptografada por uma chave do KMS)
GenerateRandom	Retorna uma string de bytes aleatória	Retorna a string de bytes aleatória criptografada pela chave pública do documento de atestado

AWS KMS oferece suporte a [chaves de condição de política](#) que você pode usar para permitir ou negar operações atestadas com uma AWS KMS chave com base no conteúdo do documento de atestado. Você também pode [monitorar solicitações atestadas AWS KMS](#) em seus AWS CloudTrail registros.

Saiba mais

- [Atestado criptográfico](#)
- [AWS KMS chaves de condição para plataformas certificadas](#)
- [Como fazer chamadas atestadas para AWS KMS](#)
- [Monitorar solicitações atestadas](#)

Como fazer chamadas atestadas para AWS KMS

Para fazer uma chamada atestada para AWS KMS, use o `Recipient` parâmetro na solicitação para fornecer o documento de atestado assinado e o algoritmo de criptografia a ser usado com a chave pública no documento de atestado. Quando uma solicitação inclui o parâmetro `Recipient` com um documento de atestado assinado, a resposta inclui um campo `CiphertextForRecipient` com o texto cifrado criptografado pela chave pública. O campo de texto simples é nulo ou vazio.

O `Recipient` parâmetro deve especificar um documento de atestação assinado de um AWS Nitro Enclaves ou NitroTPM. AWS KMS depende da assinatura digital do documento de atestado para provar que a chave pública na solicitação veio de uma fonte válida. Não é possível fornecer seu próprio certificado para assinar digitalmente o documento de atestado.

O SDK do AWS Nitro Enclaves, que é suportado somente em um enclave Nitro, adiciona automaticamente o `Recipient` parâmetro e seus valores a cada solicitação. AWS KMS

Para fazer solicitações atestadas no AWS SDKs, você precisa especificar o `Recipient` parâmetro e seus valores. [O documento de certificação pode ser recuperado do NitroTPM usando o nitro-tpm-attest utilitário ou do Nitro Secure Module \(NSM\) usando a API do NSM.](#)

AWS KMS oferece suporte a [chaves de condição de política](#) que você pode usar para permitir ou negar operações atestadas com uma AWS KMS chave com base no conteúdo do documento de atestado. Você também pode [monitorar solicitações atestadas AWS KMS](#) em seus AWS CloudTrail registros.

Para obter informações detalhadas sobre o `Recipient` parâmetro e o campo de `CiphertextForRecipient` resposta da AWS, consulte os [GenerateRandom](#) tópicos [Decrypt](#), [DeriveSharedSecret](#), [GenerateDataKey](#), e na Referência da AWS Key Management Service API [GenerateDataKeyPair](#), no SDK do [AWS Nitro Enclaves](#) ou em qualquer SDK. Para obter informações sobre como configurar seus dados e chaves de dados para criptografia, consulte [Usando o atestado criptográfico](#) com AWS KMS.

Monitorar solicitações atestadas

Você pode usar seus AWS CloudTrail registros para monitorar [GenerateRandom](#) operações de [criptografia](#), [DeriveSharedSecret](#), [GenerateDataKey](#), [GenerateDataKeyPair](#), e que usam atestado. Nessas entradas de log, o campo `additionalEventData` tem um campo `recipient` com informações do atestado na solicitação. Esses campos são incluídos somente quando o parâmetro `Recipient` da solicitação especifica um atestado assinado.

As informações específicas incluídas no CloudTrail registro dependem do método de atestado usado.

Solicitações de monitoramento para Nitro enclaves

Para o atestado do enclave Nitro, o CloudTrail registro inclui o ID do módulo (`attestationDocumentModuleId`), o resumo da imagem () e os registros de configuração da plataforma (`attestationDocumentEnclaveImageDigest`) do documento de atestado. PCRs

O ID do módulo é o [ID do enclave](#) do Nitro enclave. O resumo da imagem é o SHA384 hash da imagem do enclave. É possível usar o resumo da imagem e os valores de PCR em [condições para políticas de chave e políticas do IAM](#). Para obter informações sobre PCRs, consulte [Onde obter as medidas de um enclave no Guia do usuário do AWS Nitro Enclaves](#).

Esta seção mostra um exemplo de entrada de CloudTrail registro para cada uma das solicitações de enclave Nitro suportadas para AWS KMS

Decrypt (para um enclave)

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma operação de [descriptografia para um AWS enclave Nitro](#).

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-27T22:58:24Z",
  "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 (para um enclave)

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma [GenerateDataKey](#) operação para um enclave AWS Nitro.

```

{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:40Z",
  "eventSource": "kms.amazonaws.com",

```

```

    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "numberOfBytes": 32
    },
    "responseElements": null,
    "additionalEventData": {
      "recipient": {
        "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
        "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
        "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
        "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
        "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
        "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
        "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
      }
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }

```

GenerateDataKeyPair (para um enclave)

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma [GenerateDataKeyPair](#) operação para um 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": "2020-07-27T18:57:57Z",
    "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 (para um enclave)

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma [GenerateRandom](#) operação para um enclave AWS Nitro.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateRandom",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "recipient": {
      "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
      "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
      "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
      "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
      "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
      "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
      "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
    }
  },
  "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
  "readOnly": true,
  "resources": [],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Monitorar solicitações para o NitroTPM

Para o atestado NitroTPM, o CloudTrail registro inclui o ID do módulo (`attestationDocumentModuleId`) e os registros de configuração da plataforma () do documento de atestado. PCRs

O ID do módulo é o ID da instância do EC2 com NitroTPM com um identificador TPM. Você pode usar os valores da PCR em [condições para políticas de chave e políticas do IAM](#).

Esta seção mostra um exemplo de entrada de CloudTrail registro para cada uma das solicitações NitroTPM suportadas para. AWS KMS

Descriptografar (para um NitroTPM)

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma operação de [descriptografia para um NitroTPM](#).

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-27T22:58:24Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "additionalEventData": {
    "recipient": {
      "attestationDocumentModuleId": "i-123456789abcde123-tpm000000000000000000",
      "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",

```

```

        "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
        "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
        "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
        "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
        "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
    }
},
"requestID": "b4a65126-30d5-4b28-98b9-9153da559963",
"eventID": "e5a2f202-ba1a-467c-b4ba-f729d45ae521",
"readOnly": true,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

GenerateDataKey (para um NitroTPM)

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma [GenerateDataKey](#) operação para um NitroTPM.

```

{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",

```

```

"requestParameters": {
  "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "numberOfBytes": 32
},
"responseElements": null,
"additionalEventData": {
  "recipient": {
    "attestationDocumentModuleId": "i-123456789abcde123-tpm000000000000000000",
    "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
    "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
    "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
    "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
    "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
    "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
  }
},
"requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
"eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
"readOnly": true,
"resources": [{
  "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "accountId": "111122223333"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

GenerateDataKeyPair (para um NitroTPM)

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma [GenerateDataKeyPair](#) operação para um NitroTPM.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },

```



```

    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "keyPairSpec": "RSA_3072",
      "encryptionContext": {
        "Project": "Alpha"
      },
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
      "recipient": {
        "attestationDocumentModuleId": "i-123456789abcde123-tpm000000000000000000",
        "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
        "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
        "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
        "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
        "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
        "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
      }
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }

```

GenerateRandom (para um NitroTPM)

O exemplo a seguir mostra uma entrada de AWS CloudTrail registro de uma [GenerateRandom](#) operação para um 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"
}
```

Usando AWS KMS criptografia com AWS serviços

Com AWS Key Management Service, você pode fornecer chaves de criptografia para proteger dados em outros AWS serviços. O uso da AWS KMS criptografia com AWS serviços se refere ao processo de integração AWS KMS com outros AWS serviços para criptografar e descriptografar dados em repouso ou em trânsito. Desenvolvedores, administradores de sistemas e profissionais de segurança podem estar interessados neste tópico para proteger dados confidenciais armazenados ou transmitidos por meio de AWS serviços, atender aos requisitos de conformidade regulatória ou implementar as melhores práticas de criptografia. Casos de uso comuns incluem criptografia de volumes do Amazon EBS, buckets do Amazon S3 e bancos de dados do Amazon RDS. [As seções a seguir abordarão as etapas para configurar e gerenciar chaves de AWS KMS criptografia para AWS serviços específicos, garantindo a confidencialidade e a integridade dos dados em seu AWS ambiente. Para obter a lista completa dos AWS serviços integrados AWS KMS, consulte \[Integração de serviços.AWS\]\(#\)](#)

Os tópicos a seguir discutem detalhadamente como determinados serviços são usados AWS KMS, incluindo as chaves KMS que eles suportam, como gerenciam as chaves de dados, as permissões necessárias e como rastrear o uso das chaves KMS por cada serviço em sua conta.

Important

[AWS os serviços integrados AWS KMS](#) usam somente chaves KMS de criptografia simétrica para criptografar seus dados. Esses serviços não fornecem suporte para criptografia com chaves do KMS assimétricas. Para obter ajuda para determinar se uma chave do KMS é simétrica ou assimétrica, consulte [Identificar diferentes tipos de chaves](#).

Tópicos

- [Como o Amazon Elastic Block Store \(Amazon EBS\) usa AWS KMS](#)
- [Como o Amazon EMR usa AWS KMS](#)
- [Como o Amazon Redshift usa AWS KMS](#)

Como o Amazon Elastic Block Store (Amazon EBS) usa AWS KMS

Este tópico discute em detalhes como o [Amazon Elastic Block Store \(Amazon EBS\) usa AWS KMS para criptografar volumes](#) e snapshots. Para obter instruções básicas sobre a criptografia de volumes do Amazon EBS, consulte [Criptografia do Amazon EBS](#).

Tópicos

- [Criptografia de Amazon EBS](#)
- [Usar chaves do KMS e chaves de dados](#)
- [Contexto de criptografia do Amazon EBS](#)
- [Detectar falhas do Amazon EBS](#)
- [Usando AWS CloudFormation para criar volumes criptografados do Amazon EBS](#)

Criptografia de Amazon EBS

Quando você anexa um volume do Amazon EBS criptografado a um [tipo de instância do Amazon Elastic Compute Cloud \(Amazon EC2\) com suporte](#), os dados são armazenados em repouso no volume, E/S de disco e snapshots criados do volume são todos criptografados. A criptografia ocorre nos servidores que hospedam instâncias do Amazon EC2.

Esse recurso tem suporte em todos os [tipos de volume do Amazon EBS](#). Você acessa os volumes criptografados da mesma forma que acessa outros volumes; a criptografia e a descriptografia são manipuladas de forma transparente e elas não exigem ações adicionais de você, sua instância do EC2 ou sua aplicação. Snapshots de volumes criptografados são criptografados automaticamente, e os volumes que são criados dos snapshots criptografados também são criptografados automaticamente.

O status da criptografia de um volume do EBS é determinado quando você cria o volume. Não é possível alterar o status de criptografia de um volume existente. No entanto, você pode [migrar dados](#) entre os volumes criptografados e não criptografados, e aplicar um novo status de criptografia enquanto copia um snapshot.

Por padrão, o Amazon EBS é compatível com criptografia opcional. Você pode ativar a criptografia automaticamente em todos os novos volumes e cópias de snapshot do EBS na sua região Conta da AWS. Essa configuração não afeta volumes ou snapshots existentes. Para obter mais informações, consulte [Amazon EBS encryption](#) no Amazon EBS User Guide.

Usar chaves do KMS e chaves de dados

Ao [criar um volume do Amazon EBS criptografado](#), você especifica uma AWS KMS key. Por padrão, o Amazon EBS usa a [Chave gerenciada pela AWS](#) para o Amazon EBS na sua conta (aws/ebs). No entanto, você pode especificar uma [chave gerenciada pelo cliente](#) que você cria e gerencia.

Para usar uma chave gerenciada pelo cliente, você deve dar ao Amazon EBS a permissão para usar a chave do KMS em seu nome. Para obter mais informações, consulte [How Amazon EBS encryption works](#) no Amazon EBS User Guide.

Important

O Amazon EBS oferece suporte somente para [chaves do KMS simétricas](#). Não é possível usar uma [chave do KMS assimétrica](#) para criptografar um volume do Amazon EBS. Para obter ajuda para determinar se uma chave do KMS é simétrica ou assimétrica, consulte [Identificar diferentes tipos de chaves](#).

Para cada volume, o Amazon EBS solicita AWS KMS a geração de uma chave de dados exclusiva criptografada sob a chave KMS que você especificar. O Amazon EBS armazena a chave de dados criptografada com o volume. Então, quando você anexa o volume a uma instância do Amazon EC2, o Amazon EBS liga AWS KMS para descriptografar a chave de dados. O Amazon EBS usa a chave de dados de texto simples na memória do hipervisor para criptografar todo o disco no volume. I/O Para obter detalhes, consulte Funcionamento da criptografia do EBS no [Guia do usuário do Amazon EC2](#) ou no [Guia do usuário do Amazon EC2](#).

Contexto de criptografia do Amazon EBS

Em suas solicitações [GenerateDataKeyWithoutPlaintext](#) e [Decrypt](#), o AWS KMS Amazon EBS usa um contexto de criptografia com um par de nome-valor que identifica o volume ou o snapshot na solicitação. O nome no contexto de criptografia não varia.

Um [contexto de criptografia](#) é um conjunto de pares de chave-valor que contêm dados arbitrários não secretos. Quando você inclui um contexto de criptografia em uma solicitação para criptografar dados, vincula AWS KMS criptograficamente o contexto de criptografia aos dados criptografados. Para descriptografar os dados, é necessário passar o mesmo contexto de criptografia.

Para todos os volumes e para snapshots criptografados criados com a [CreateSnapshot](#) operação do Amazon EBS, o Amazon EBS usa o ID do volume como valor do contexto de criptografia. No campo

`requestParameters` de uma entrada de log do CloudTrail, o contexto de criptografia é parecido com o seguinte:

```
"encryptionContext": {  
  "aws:ebs:id": "vol-0cfb133e847d28be9"  
}
```

Para snapshots criptografados criados com a operação do Amazon [CopySnapshot](#)EC2, o Amazon EBS usa o ID do snapshot como valor de contexto de criptografia. No campo `requestParameters` de uma entrada de log do CloudTrail, o contexto de criptografia é parecido com o seguinte:

```
"encryptionContext": {  
  "aws:ebs:id": "snap-069a655b568de654f"  
}
```

Detectar falhas do Amazon EBS

Para criar um volume do EBS criptografado ou anexar o volume a uma instância do EC2, o Amazon EBS e a infraestrutura do Amazon EC2 devem poder usar a chave do KMS especificada para a criptografia de volume do EBS. Quando a chave do KMS não pode ser utilizada, por exemplo, quando seu [estado de chave](#) não está Enabled, há falha na criação ou na anexação do volume.

Nesse caso, o Amazon EBS envia um evento para a Amazon EventBridge (antigo CloudWatch Events) para notificá-lo sobre a falha. Em EventBridge, você pode estabelecer regras que acionam ações automáticas em resposta a esses eventos. Para obter mais informações, consulte [CloudWatch Eventos da Amazon para Amazon EBS](#) no Guia do usuário do Amazon EBS, especialmente nas seguintes seções:

- [Chave de criptografia inválida em Anexar ou reanexar volume](#)
- [Chave de criptografia inválida em Criar volume](#)

Para corrigir essas falhas, verifique se a chave do KMS especificada para criptografia do volume do EBS está habilitada. Para fazer isso, primeiro [visualize a chave KMS](#) para determinar o estado atual da chave (a coluna Status no Console de gerenciamento da AWS). Veja as informações em um dos links a seguir:

- Se o estado da chave do KMS estiver desabilitado, [habilite-o](#).
- Se o estado da chave do KMS for importação pendente, [importe material de chave](#).

- Se o estado de chave da chave do KMS for exclusão pendente, [cancele a exclusão da chave](#).

Usando AWS CloudFormation para criar volumes criptografados do Amazon EBS

Você pode usar o [AWS CloudFormation](#) para criar volumes criptografados do Amazon EBS. Consulte mais informações em [AWS::EC2::Volume](#) no Guia de Usuário AWS CloudFormation .

Como o Amazon EMR usa AWS KMS

Ao usar um cluster do [Amazon EMR](#), você pode configurá-lo para criptografar dados em repouso antes de salvá-los em um local de armazenamento persistente. Você pode criptografar dados em repouso no EMR File System (EMRFS), nos volumes de armazenamento de nós de cluster, ou em ambos. Para criptografar dados em repouso, use uma AWS KMS key. Os tópicos a seguir explicam como um cluster do Amazon EMR usa uma chave do KMS para criptografar dados em repouso.

Important

O Amazon EMR oferece suporte somente a [chaves do KMS simétricas](#). Não é possível usar uma [chave do KMS assimétrica](#) para criptografar dados em repouso em um cluster do Amazon EMR. Para obter ajuda para determinar se uma chave do KMS é simétrica ou assimétrica, consulte [Identificar diferentes tipos de chaves](#).

Os clusters do Amazon EMR também criptografam dados em trânsito, o que significa que o cluster criptografa os dados antes de enviá-los por meio da rede. Não é possível usar uma chave do KMS para criptografar dados em trânsito. Para obter mais informações, consulte [Criptografia de dados em trânsito](#), no Guia de gerenciamento do Amazon EMR.

Para obter mais informações sobre todas as opções de criptografia disponíveis no Amazon EMR, consulte [Opções de criptografia](#), no Guia de gerenciamento do Amazon EMR.

Tópicos

- [Criptografar dados no EMR File System \(EMRFS\)](#)
- [Criptografar dados nos volumes de armazenamento dos nós de cluster](#)
- [Contexto de criptografia](#)

Criptografar dados no EMR File System (EMRFS)

Os clusters do Amazon EMR usam dois sistemas de arquivos distribuídos:

- O Hadoop Distributed File System (HDFS). A criptografia do HDFS não usa uma chave do KMS no AWS KMS.
- O EMR File System (EMRFS). O EMRFS é uma implementação do HDFS que permite que os clusters do Amazon EMR armazenem dados no Amazon Simple Storage Service (Amazon S3). O EMRFS oferece suporte a quatro opções de criptografia, duas das quais usam uma chave do KMS no AWS KMS. Para obter mais informações sobre todas as quatro opções de criptografia do EMRFS, consulte [Opções de criptografia](#), no Guia de gerenciamento do Amazon EMR.

As duas opções de criptografia do EMRFS que usam uma chave do KMS usam os seguintes recursos de criptografia oferecidos pelo Amazon S3:

- [Protegendo dados usando criptografia do lado do servidor com AWS Key Management Service \(SSE-KMS\)](#). O cluster do Amazon EMR envia dados ao Amazon S3. O Amazon S3 usa uma chave do KMS para criptografar os dados antes de enviá-los a um bucket do S3. Para obter mais informações sobre como isso funciona, consulte [Processo para criptografar dados no EMRFS com o SSE-KMS](#).
- [Proteger dados usando a criptografia no lado do cliente](#) (CSE-KMS). Os dados em um Amazon EMR são criptografados sob um AWS KMS key antes de serem enviados ao Amazon S3 para armazenamento. Para obter mais informações sobre como isso funciona, consulte [Processo para criptografar dados no EMRFS com o CSE-KMS](#).

Ao configurar um cluster do Amazon EMR para criptografar dados no EMRFS com uma chave do KMS, escolha a chave do KMS que deseja que o Amazon S3 ou o cluster do Amazon EMR use. Com o SSE-KMS, é possível escolher a Chave gerenciada pela AWS para o Amazon S3 com o alias `aws/s3` ou uma chave simétrica gerenciada pelo cliente criada por você. Com a criptografia no lado do cliente, é necessário escolher uma chave simétrica gerenciada pelo cliente criada por você. Ao escolher uma chave gerenciada pelo cliente, é necessário garantir que o cluster do Amazon EMR tenha permissão para usar a chave do KMS. Para obter mais informações, consulte [Uso AWS KMS keys para criptografia](#) no Guia de gerenciamento do Amazon EMR.

Para a criptografia do lado do servidor e a criptografia do lado do cliente, a chave do KMS que você escolhe é a chave raiz em um fluxo de trabalho de [criptografia de envelope](#). Os dados são criptografados com uma [chave de dados](#) exclusiva que é criptografada sob a chave KMS em AWS

KMS. Os dados criptografados e uma cópia criptografada de sua chave de dados são armazenados em conjunto como um único objeto criptografado em um bucket do S3. Para obter mais informações sobre como isso funciona, consulte os tópicos a seguir.

Tópicos

- [Processo para criptografar dados no EMRFS com o SSE-KMS](#)
- [Processo para criptografar dados no EMRFS com o CSE-KMS](#)

Processo para criptografar dados no EMRFS com o SSE-KMS

Quando você configura um cluster do Amazon EMR para usar o SSE-KMS, o processo de criptografia funciona da seguinte forma:

1. O cluster envia os dados ao Amazon S3 para armazenamento em um bucket do S3.
2. O Amazon S3 envia uma [GenerateDataKey](#) solicitação para AWS KMS, especificando o ID da chave KMS que você escolheu ao configurar o cluster para usar o SSE-KMS. A solicitação inclui o contexto de criptografia, para obter mais informações, consulte [Contexto de criptografia](#).
3. AWS KMS gera uma chave de criptografia de dados exclusiva (chave de dados) e, em seguida, envia duas cópias dessa chave de dados para o Amazon S3. Uma cópia é não criptografada (texto simples), e a outra cópia é criptografada com a chave do KMS.
4. O Amazon S3 usa a chave de dados de texto não criptografado para criptografar os dados que ela recebeu na etapa 1 e remove a chave de dados de texto não criptografado da memória, assim que possível, após o uso.
5. O Amazon S3 armazena os dados criptografados e a cópia criptografada da chave de dados em conjunto como um único objeto criptografado em um bucket do S3.

O processo de descriptografia funciona desta forma:

1. O cluster solicita um objeto de dados criptografado de um bucket do S3.
2. [O Amazon S3 extrai a chave de dados criptografada do objeto S3 e, em seguida, envia a chave de dados criptografada para AWS KMS com uma solicitação Decrypt](#). A solicitação inclui um [contexto de criptografia](#).
3. AWS KMS descriptografa a chave de dados criptografada usando a mesma chave KMS usada para criptografá-la e, em seguida, envia a chave de dados descriptografada (texto sem formatação) para o Amazon S3.

4. O Amazon S3 usa a chave de dados de texto não criptografado para descriptografar os dados criptografados e, em seguida, remove a chave de dados de texto não criptografado da memória, assim que possível, após o uso.
5. O Amazon S3 envia os dados descriptografados ao cluster.

Processo para criptografar dados no EMRFS com o CSE-KMS

Quando você configura um cluster do Amazon EMR para usar o CSE-KMS, o processo de criptografia funciona da seguinte forma:

1. Quando estiver pronto para armazenar dados no Amazon S3, o cluster envia uma [GenerateDataKey](#) solicitação para AWS KMS, especificando o ID da chave KMS que você escolheu ao configurar o cluster para usar o CSE-KMS. A solicitação inclui o contexto de criptografia, para obter mais informações, consulte [Contexto de criptografia](#).
2. AWS KMS gera uma chave de criptografia de dados exclusiva (chave de dados) e, em seguida, envia duas cópias dessa chave de dados para o cluster. Uma cópia é não criptografada (texto simples), e a outra cópia é criptografada com a chave do KMS.
3. O cluster usa a chave de dados de texto simples para criptografar os dados e remove a chave de dados de texto simples da memória, assim que possível após o uso.
4. O cluster combina os dados criptografados e a cópia criptografada da chave de dados em conjunto em um único objeto criptografado.
5. O cluster envia o objeto criptografado para o Amazon S3 para armazenamento.

O processo de descriptografia funciona desta forma:

1. O cluster solicita o objeto de dados criptografado de um bucket do S3.
2. O Amazon S3 envia o objeto criptografado ao cluster.
3. O cluster extrai a chave de dados criptografada do objeto criptografado e, em seguida, envia a chave de dados criptografada para AWS KMS com uma solicitação [Decrypt](#). A solicitação inclui o [contexto de criptografia](#).
4. AWS KMS descriptografa a chave de dados criptografada usando a mesma chave KMS usada para criptografá-la e, em seguida, envia a chave de dados descriptografada (texto sem formatação) para o cluster.
5. O cluster usa a chave de dados de texto simples para descriptografar os dados criptografados e remove a chave de dados de texto simples da memória, assim que possível após o uso.

Criptografar dados nos volumes de armazenamento dos nós de cluster

Um cluster do Amazon EMR é um conjunto de instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Cada instância no cluster é chamada de um nó de cluster ou nó. Cada nó pode ter dois tipos de volumes de armazenamento: volumes de armazenamento de instância e volumes do Amazon Elastic Block Store (Amazon EBS). Você pode configurar o cluster para usar [Linux Unified Key Setup \(LUKS\)](#) para criptografar ambos os tipos de volumes de armazenamento nos nós (mas não o volume de inicialização de cada nó). Isso é chamado de criptografia de disco local.

Ao habilitar a criptografia de disco local para um cluster, você pode optar por criptografar a chave LUKS com uma chave do KMS no AWS KMS. É necessário selecionar uma [chave gerenciada pelo cliente](#) criada por você. Não é possível usar uma [Chave gerenciada pela AWS](#). Se você escolher uma chave gerenciada pelo cliente, será necessário garantir que o cluster do Amazon EMR tenha permissão para usar a chave do KMS. Para obter mais informações, consulte [Uso AWS KMS keys para criptografia](#) no Guia de gerenciamento do Amazon EMR.

Quando você habilita a criptografia de disco local usando uma chave do KMS, o processo de criptografia funciona desta forma:

1. Quando cada nó do cluster é iniciado, ele envia uma [GenerateDataKey](#) solicitação para AWS KMS, especificando o ID da chave KMS que você escolheu ao habilitar a criptografia de disco local para o cluster.
2. AWS KMS gera uma chave de criptografia de dados exclusiva (chave de dados) e, em seguida, envia duas cópias dessa chave de dados para o nó. Uma cópia é não criptografada (texto simples), e a outra cópia é criptografada com a chave do KMS.
3. O nó usa uma versão codificada em base64 da chave de dados em texto simples como a senha que protege a chave LUKS. O nó salva a cópia criptografada da chave de dados em seu volume de inicialização.
4. [Se o nó for reinicializado, o nó reinicializado enviará a chave de dados criptografada para AWS KMS com uma solicitação Decrypt.](#)
5. AWS KMS descriptografa a chave de dados criptografada usando a mesma chave KMS usada para criptografá-la e, em seguida, envia a chave de dados descriptografada (texto sem formatação) para o nó.
6. O nó usa uma versão codificada em base64 da chave de dados de texto simples como a senha para desbloquear a chave LUKS.

Contexto de criptografia

Cada AWS serviço integrado ao AWS KMS pode especificar um [contexto de criptografia](#) quando usado AWS KMS para gerar chaves de dados ou para criptografar ou descriptografar dados. O contexto de criptografia é uma informação adicional autenticada AWS KMS usada para verificar a integridade dos dados. Quando um serviço especifica um contexto de criptografia para uma operação de criptografia, ele deve especificar o mesmo contexto de criptografia para a operação de descriptografia correspondente ou a descriptografia não terá êxito. O contexto de criptografia também é gravado em arquivos de AWS CloudTrail log, o que pode ajudar você a entender por que uma chave KMS específica foi usada.

A seção a seguir explica o contexto de criptografia usado em cada cenário de criptografia do Amazon EMR que usa uma chave do KMS.

Contexto de criptografia para criptografia EMRFS com o SSE-KMS

Com o SSE-KMS, o cluster Amazon EMR; envia dados ao Amazon S3 e, em seguida, o Amazon S3 usa uma chave do KMS para criptografar esses dados antes de enviá-los a um bucket do S3. Nesse caso, o Amazon S3 usa o Amazon Resource Name (ARN) do objeto S3 como contexto de criptografia com cada solicitação [GenerateDataKey](#) e [Decrypt](#) para a qual ele envia. AWS KMS O exemplo a seguir mostra uma representação JSON do contexto de criptografia usado pelo Amazon S3.

```
{ "aws:s3:arn" : "arn:aws:s3:::S3_bucket_name/S3_object_key" }
```

Contexto de criptografia para criptografia EMRFS com o CSE-KMS

Com o CSE-KMS, o cluster do Amazon EMR usa uma chave do KMS para criptografar os dados antes de os enviar ao Amazon S3 para armazenamento. Nesse caso, o cluster usa o Amazon Resource Name (ARN) da chave KMS como contexto de criptografia com cada solicitação [GenerateDataKey](#) e [Decrypt](#) para a qual ele envia. AWS KMS O exemplo a seguir mostra uma representação JSON do contexto de criptografia que o cluster usa.

```
{ "kms_cmek_id" : "arn:aws:kms:us-east-2:111122223333:key/0987ab65-43cd-21ef-09ab-87654321cdef" }
```

Contexto de criptografia para criptografia de disco local com LUKS

Quando um cluster do Amazon EMR usa criptografia de disco local com LUKS, os nós do cluster não especificam o contexto de criptografia com as solicitações [GenerateDataKey](#) e [Decrypt](#) para as quais eles enviam. AWS KMS

Como o Amazon Redshift usa AWS KMS

Este tópico discute como o Amazon Redshift AWS KMS usa para criptografar dados.

Tópicos

- [Criptografia do Amazon Redshift](#)
- [Contexto de criptografia](#)

Criptografia do Amazon Redshift

Um data warehouse do Amazon Redshift é um conjunto de recursos de computação chamados nós, que são organizados em um grupo chamado cluster. Cada cluster executa um mecanismo do Amazon Redshift e contém um ou mais bancos de dados.

O Amazon Redshift usa para criptografia uma arquitetura de quatro níveis baseada em chaves. A arquitetura consiste em chaves de criptografia dos dados, uma chave de banco de dados, uma chave de cluster e uma chave raiz. Você pode usar um AWS KMS key como chave raiz.

As chaves de criptografia dos dados criptografam blocos de dados do cluster. Cada bloco de dados recebe uma chave AES-256 gerada aleatoriamente. Essas chaves são criptografadas com a chave do banco de dados do cluster.

A chave do banco de dados criptografa as chaves de criptografia dos dados do cluster. A chave do banco de dados é uma chave AES-256 gerada aleatoriamente. Ela é armazenada em disco em uma rede separada do cluster do Amazon Redshift e passada para o cluster por meio de um canal seguro.

A chave do cluster criptografa a chave do banco de dados do cluster do Amazon Redshift. Você pode usar AWS KMS AWS CloudHSM, ou um módulo de segurança de hardware externo (HSM) para gerenciar a chave do cluster. Para obter mais detalhes, consulte a documentação de [Criptografia de banco de dados do Amazon Redshift](#).

Para solicitar criptografia, marque a caixa apropriada no console do Amazon Redshift. Para especificar uma [chave gerenciada pelo cliente](#), escolha uma na lista que aparece abaixo da caixa de criptografia. Se você não especificar uma chave gerenciada pelo cliente, o Amazon Redshift usará a [Chave gerenciada pela AWS](#) na sua conta.

 Important

O Amazon Redshift só é compatível com chaves do KMS de criptografia simétrica. Não é possível usar uma chave do KMS assimétrica em um fluxo de trabalho de criptografia do Amazon Redshift. Para obter ajuda para determinar se uma chave do KMS é simétrica ou assimétrica, consulte [Identificar diferentes tipos de chaves](#).

Contexto de criptografia

Cada serviço integrado AWS KMS especifica um [contexto de criptografia](#) ao solicitar chaves de dados, criptografar e descriptografar. O contexto de criptografia são dados adicionais autenticados (AAD) AWS KMS usados para verificar a integridade dos dados. Ou seja, quando um contexto de criptografia é especificado para uma operação de criptografia, o serviço também o especifica para a operação de descriptografia ou a descriptografia não terá êxito. O Amazon RedShift usa o ID do cluster e a hora de criação no contexto de criptografia. No `requestParameters` campo de um arquivo de CloudTrail log, o contexto de criptografia será semelhante a este.

```
"encryptionContext": {  
  "aws:redshift:arn": "arn:aws:redshift:region:account_ID:cluster:cluster_name",  
  "aws:redshift:createtime": "20150206T1832Z"  
},
```

Você pode pesquisar o nome do cluster em seus CloudTrail registros para entender quais operações foram realizadas usando uma AWS KMS key (chave KMS). As operações incluem a criptografia e descriptografia do cluster e a geração de chaves de dados.

AWS KMS Referência

O material de referência a seguir fornecem informações úteis sobre o uso e o gerenciamento de chaves do KMS.

- [Referência do tipo de chave](#). Lista o tipo de chave do KMS que oferece suporte a cada operação da API do AWS KMS .

Para encontrar: posso habilitar e desabilitar uma chave do KMS de assinatura RSA?

- [Tabela de estados de chaves](#). Mostra como o estado da chave de uma chave do KMS afeta seu uso em operações de API do AWS KMS .

Para encontrar: posso alterar o alias de uma chave do KMS cuja exclusão está pendente?

- [Referência de permissões da API do AWS KMS](#). Fornece informações sobre as permissões necessárias para cada operação de AWS KMS API.

Para descobrir: Posso [GetKeyPolicy](#) usar uma chave em uma AWS conta diferente? Posso conceder a permissão `kms:Decrypt` em uma política do IAM?

- [ViaService referência](#). Lista os serviços da AWS que oferecem suporte à chave de condição `kms:ViaService`.

Para descobrir: Posso usar a chave de `kms:ViaService` condição para permitir uma permissão somente quando ela vem da Amazon ElastiCache? E o Amazon Neptune?

- [AWS KMS preços](#). Lista e explica o preço das chaves do KMS.

Para encontrar: quanto custa para usar minhas chaves assimétricas?

- [AWS KMS solicitar cotas](#). Lista as cotas por segundo para solicitações de AWS KMS API em cada conta e região.

Para encontrar: quantas solicitações [Decrypt](#) posso executar em cada segundo? Quantas solicitações [Decrypt](#) posso executar em chaves do KMS no meu armazenamento de chaves personalizado?

- [AWS KMS cotas de recursos](#). Lista as cotas nos recursos da AWS KMS .

Para encontrar: quantas chaves KMS posso ter em cada região da minha conta? Quantos aliases posso ter em cada chave do KMS?

- [AWS serviços integrados com AWS KMS](#). Lista os AWS serviços que usam chaves KMS para proteger os recursos que eles criam, armazenam e gerenciam.

Para encontrar: o Amazon Connect usa chaves do KMS para proteger meus recursos do Connect?

Principais estados das AWS KMS chaves

E AWS KMS key sempre tem um estado chave. Operações na chave do KMS e seu ambiente podem alterar o estado da chave. Operações na chave do KMS e seu ambiente podem alterar esse estado de chave, transitoriamente ou até que outra operação altere seu estado de chave. Essas operações são feitas de modo assíncrono ou por uma chamada de API.

A tabela nesta seção mostra como os estados principais afetam as chamadas para operações de AWS KMS API. Como resultado de seu estado de chave, espera-se que uma operação em uma chave do KMS tenha êxito (#), apresente falhas (X) ou tenha êxito somente em certas condições (?). O resultado muitas vezes é diferente para chaves do KMS com material de chave importado.

Essa tabela inclui apenas as operações de API que usam uma chave do KMS existente. Outras operações, como [CreateKey](#) e [ListKeys](#), são omitidas.

Tópicos

- [Estados de chave e tipos de chaves do KMS](#)
- [Tabela de estados de chave](#)

Estados de chave e tipos de chaves do KMS

O tipo da chave do KMS determina os estados de chave que ela pode ter.

- Todas as chaves do KMS podem estar nos estados Enabled, Disabled e PendingDeletion.
- A maioria das chaves do KMS é criada no estado Enabled. Chaves com material de chave importado são criadas no estado PendingImport.
- O estado PendingImport aplica-se somente a chaves do KMS com [material de chave importado](#). Quando qualquer material de chave de uma chave importada é excluído ou expira, o estado muda de Enabled para PendingImport.
- O estado Unavailable aplica-se somente a uma chave do KMS em um [armazenamento de chaves personalizado](#). Uma chave KMS em um [armazenamento de AWS CloudHSM chaves](#)

[ocorre](#) Unavailable quando o armazenamento de chaves personalizado é intencionalmente desconectado de seu cluster. AWS CloudHSM Uma chave do KMS em um [armazenamento de chaves externas](#) está Unavailable quando o armazenamento de chaves personalizado foi desconectado do [proxy de armazenamento de chaves externas](#) intencionalmente. É possível visualizar e gerenciar chaves do KMS indisponíveis, mas não é possível usá-las em operações de criptografia.

O estado da chave de uma chave do KMS em um armazenamento de chaves personalizado não é afetado pelas alterações em sua chave de reserva. Uma chave KMS em um armazenamento de AWS CloudHSM chaves não é afetada pelas alterações no [material de chaves associado](#) no AWS CloudHSM cluster. Uma chave do KMS em um armazenamento de chaves externas não é afetada por alterações em sua [chave externa](#) em um gerenciador de chaves externas. Se a chave de reserva estiver desabilitada ou excluída, o estado da chave do KMS não será alterado, mas as operações de criptografia que usam a chave do KMS apresentarão falha.

- Os estados de chave Creating, Updating e PendingReplicaDeletion aplicam-se somente a [chaves de várias regiões](#).
 - Uma chave de réplica de várias regiões está no estado de chave Creating enquanto ela está sendo criada. Esse processo ainda pode estar em andamento quando a [ReplicateKey](#) operação for concluída. Quando o processo de replicação estiver concluído, a chave de réplica estará no estado Enabled ou PendingImport.
 - Chaves de várias regiões estão no estado de chave Updating transitório enquanto a região primária está sendo atualizada. Esse processo ainda pode estar em andamento quando a [UpdatePrimaryRegion](#) operação for concluída. Quando o processo de atualização estiver concluído, as chaves primária e de réplica retomarão o estado de chave Enabled.
 - Quando você programar a exclusão de uma chave primária de várias regiões contendo chaves de réplica, essa chave primária estará no estado PendingReplicaDeletion até que todas as suas chaves de réplica sejam excluídas. Seu estado de chave muda para PendingDeletion. Para obter detalhes, consulte [Deleting multi-Region keys](#).

Tabela de estados de chave

A tabela a seguir mostra como o estado de uma chave do KMS afeta operações do AWS KMS .

Descrições de notas de rodapé numeradas ([n]) estão no final deste tópico.

Note

Talvez seja necessário rolar horizontalmente ou verticalmente para ver todos os dados nessa tabela.

solicitações de	Habilitado	Desabilitado	Exclusão pendente Exclusão pendente de réplica	Importação pendente	Unavailable (Indisponível)	Criando	Atualização
CancelKey Deletion	 [4]	 [4]		 [4]	 [4], [13]	 [4]	 [4]
CreateAliases			 [3]				
CreateGrant		 [1]	 [2] ou [3]	 [5]		 [14]	
Decrypt		 [1]	 [2] ou [3]	 [5]	 [11]	 [14]	
DeleteAliases							
DeleteImportedKeyMaterial	 [9]	 [9]	 [9]		N/D		

solicitações de	Habilitado	Desabilitado	Exclusão pendente Exclusão pendente de réplica	Importação pendente	Unavailable (Indisponível)	Criando	Atualização
						[14]	[15]
DeriveSharedSecret	✓	✗ [1]	✗ [2] ou [3]	✗ [5]	N/D	✗ [14]	✓ [15]
DescribeKey	✓	✓	✓	✓	✓	✓	✓
DisableKey	✓	✓	✗ [3]	✗ [5]	✓ [12]	✗ [14]	✗ [15]
DisableKeyRotation	❓ [7]	✗ [1] ou [7]	✗ [3] ou [7]	✗ [6]	✗ [7]	✗ [14]	❓ [7]
EnableKey	✓	✓	✗ [3]	✗ [5]	✓ [12]	✗ [14]	✗ [15]
EnableKeyRotation	❓ [7]	✗ [1] ou [7]	✗ [3] ou [7]	✗ [6]	✗ [7]	✗ [14]	❓ [7]
Encrypt	✓	✗ [1]	✗ [2] ou [3]	✗ [5]	✗ [11]	✗ [14]	✓

solicitações de	Habilitado	Desabilitado	Exclusão pendente Exclusão pendente de réplica	Importação pendente	Unavailable (Indisponível)	Criando	Atualização
GenerateDataKey	✓	 [1]	 [2] ou [3]	 [5]	 [11]	 [14]	✓
GenerateDataKeyPair	✓	 [1]	 [2] ou [3]	 [5]	 [7]	 [14]	✓
GenerateDataKeyPairWithoutPlaintext	✓	 [1]	 [2] ou [3]	 [5]	 [7]	 [14]	✓
GenerateDataKeyWithoutPlaintext	✓	 [1]	 [2] ou [3]	 [5]	 [11]	 [14]	✓
GenerateMac	✓	 [1]	 [2] ou [3]	 [5]	N/D	 [14]	✓
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓
GetKeyRotationStatus	 [7]	 [7]	 [7]	 [6]	 [7]	 [7]	 [7]

solicitações de	Habilitado	Desabilitado	Exclusão pendente Exclusão pendente de réplica	Importação pendente	Unavailable (Indisponível)	Criando	Atualização
GetParametersForImport	 [9]	 [9]	 [8] ou [9]		 [9]	 [14]	 [15]
GetPublicKey			 [2] ou [3]		N/D	 [14]	
ImportKeyMaterial	 [9]	 [9]	 [9]		 [9]	 [14]	
ListAliases							
ListGrants							
ListKeyPolicies							
ListKeyRotations	 [7]	 [7]	 [7]		 [7]	 [7]	 [7]
ListResourceTags							

solicitações de	Habilitado	Desabilitado	Exclusão pendente Exclusão pendente de réplica	Importação pendente	Unavailable (Indisponível)	Criando	Atualização
PutKeyPolicy	✓	✓	✓	✓	✓	✓	✓
ReEncrypt	✓	✗ [1]	✗ [2] ou [3]	✗ [5]	✗ [11]	✗ [14]	✓
Replicate Key	✓	✗ [1]	✗ [2] ou [3]	✗ [5]	N/D	✗ [14]	✗ [15]
RetireGrant	✓	✓	✓	✓	✓	✓	✓
RevokeGrant	✓	✓	✓	✓	✓	✓	✓
RotateKey OnDemand	🔍 [7]	✗ [1] ou [7]	✗ [3] ou [7]	✗ [5]	✗ [7]	✗ [14]	🔍 [7]
ScheduleKeyDeletion	✓	✓	✗ [3]	✓	✓	✓	✗ [15]
Sign	✓	✗ [1]	✗ [2] ou [3]	✗ [5]	N/D	✗ [14]	✓

solicitações de	Habilitado	Desabilitado	Exclusão pendente Exclusão pendente de réplica	Importação pendente	Unavailable (Indisponível)	Criando	Atualização
TagResource	✓	✓	✗ [3]	✓	✓	✓	✓
UntagResource	✓	✓	✗ [3]	✓	✓	✓	✓
UpdateAliases	✓	✓	?	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	✗ [3]	✓	✓	✓	✓
UpdatePrimaryRegion	✓	✗ [1]	✗ [2] ou [3]	✗ [5]	N/D	✗ [14]	✓
Verificar	✓	✗ [1]	✗ [2] ou [3]	✗ [5]	N/D	✗ [14]	✓
VerifyMac	✓	✗ [1]	✗ [2] ou [3]	✗ [5]	N/D	✗ [14]	✓

Detalhes da tabela

- [1] DisabledException: `<key ARN>` is disabled.
- [2] DisabledException: `<key ARN>` is pending deletion (or pending replica deletion).
- [3] KMSInvalidStateException: `<key ARN>` is pending deletion (or pending replica deletion).
- [4] KMSInvalidStateException: `<key ARN>` is not pending deletion (or pending replica deletion).
- [5] KMSInvalidStateException: `<key ARN>` is pending import because no key material has ever been imported or one of the imported key materials is deleted or expired.
- [6] UnsupportedOperationException: `<key ARN>` origin is EXTERNAL which is not valid for this operation.
- [7] Se a chave do KMS estiver em um repositório de chaves personalizado: UnsupportedOperationException.
- [8] Se a chave do KMS tiver material de chave importado: KMSInvalidStateException
- [9] Se a chave do KMS não puder ter material de chave importado: UnsupportedOperationException.
- [10] Se a exclusão da chave do KMS de origem está pendente, o comando foi bem-sucedido. Se a exclusão da chave do KMS de destino está pendente, o comando falha com o erro: KMSInvalidStateException : `<key ARN>` is pending deletion.
- [11] KMSInvalidStateException: `<key ARN>` is unavailable. Não é possível executar essa operação em uma chave do KMS indisponível.
- [12] A operação é bem-sucedida, mas o estado da chave do KMS não muda até que ela se torne disponível.
- [13] Enquanto a exclusão de uma chave do KMS em um armazenamento de chaves personalizado estiver pendente, seu estado de chave permanecerá PendingDeletion, mesmo que a chave do KMS se torne indisponível. Isso permite que você cancele a exclusão da chave do KMS a qualquer momento durante o período de espera.
- [14] KMSInvalidStateException: `<key ARN>` is creating. AWS KMS lança essa exceção enquanto replica uma chave multirregional (). ReplicateKey

- [15] `KMSInvalidStateException`: `<key ARN> is updating`. AWS KMS lança essa exceção enquanto atualiza a região primária de uma chave multirregional (`UpdatePrimaryRegion`).

Referência de tipos de chaves

AWS KMS suporta recursos diferentes para diferentes tipos de chaves KMS. Por exemplo, somente é possível usar [chaves do KMS de criptografia simétrica](#) para [gerar chaves de dados simétricas](#) e [pares de chaves de dados assimétricas](#). Além disso, a [importação do material de chave](#) e o [rodízio de chaves automático](#) são compatíveis somente com chaves do KMS de criptografia simétrica, e somente é possível criar chaves do KMS de criptografia simétrica em um [repositório de chaves personalizado](#).

Esta referência contém duas tabelas.

- A [tabela de tipos de chave](#) lista as AWS KMS operações válidas para chaves KMS de criptografia simétrica, chaves KMS assimétricas e chaves HMAC KMS.
- A [tabela de recursos especiais](#) apresenta as operações do AWS KMS que são válidas para chaves do KMS multirregionais, chaves do KMS com material de chave importado e chaves do KMS em armazenamentos de chaves personalizados.

Tabela de tipos de chave

Talvez seja necessário rolar horizontalmente ou verticalmente para ver todos os dados nessa tabela.

AWS KMS Operação de API	Chaves do KMS de criptogra fia simétrica	Chaves do KMS de HMAC	Chaves do KMS assimétricas (ENCRYPT DECRYPT)	Chaves do KMS assimétricas (SIGN_VERIFY)	Chaves do KMS assimétricas (KEY_AGREEMENT)
CancelKeyDeletion	Sim	Sim	Sim	Sim	Sim
CreateAlias	Sim	Sim	Sim	Sim	Sim
CreateGrant	Sim	Sim	Sim	Sim	Sim

AWS KMS Operação de API	Chaves do KMS de criptogra fia simétrica	Chaves do KMS de HMAC	Chaves do KMS assimétricas (ENCRYPT DECRYPT)	Chaves do KMS assimétricas (SIGN_VERIFY)	Chaves do KMS assimétricas (KEY_AGREEMENT)
CreateKey	Sim	Sim	Sim	Sim	Sim
Decrypt	Sim	Não	Sim	Não	Não
DeleteAlias	Sim	Sim	Sim	Sim	Sim
DeleteImportedKeyMaterial É válido somente em chaves do KMS com material de chave importado (Origin é EXTERNAL).	Sim	Sim	Sim	Sim	Sim
DeriveSharedSecret	Não	Não	Não	Não	Sim
DescribeKey	Sim	Sim	Sim	Sim	Sim
DisableKey	Sim	Sim	Sim	Sim	Sim

AWS KMS Operação de API	Chaves do KMS de criptogra fia simétrica	Chaves do KMS de HMAC	Chaves do KMS assimétricas (ENCRYPT DECRYPT)	Chaves do KMS assimétricas (SIGN_VERIFY)	Chaves do KMS assimétricas (KEY_AGREEMENT)
DisableKeyRotation	Sim Válido somente em chaves KMS com material (Origins de AWS KMS chave.	Não	Não	Não	Não
EnableKey	Sim	Sim	Sim	Sim	Sim
EnableKeyRotation	Sim Válido somente em chaves KMS com material (Origins de AWS KMS chave.	Não	Não	Não	Não
Encrypt	Sim	Não	Sim	Não	Não

AWS KMS Operação de API	Chaves do KMS de criptogra fia simétrica	Chaves do KMS de HMAC	Chaves do KMS assimétri cas (ENCRYPT DECRYPT)	Chaves do KMS assimétricas (SIGN_VERIFY)	Chaves do KMS assimétricas (KEY_AGRE EMENT)
GenerateDataKey	Sim	Não	Não	Não	Não
GenerateDataKeyPair Gera um par de chaves de dados assimétricas que é protegido por uma chave do KMS de criptogra fia simétrica.	Sim Não é válido com chaves do KMS em armazena mentos de chaves personali zados.	Não	Não	Não	Não
GenerateDataKeyPairWithoutPlaintext Gera um par de chaves de dados assimétricas que é protegido por uma chave do KMS de criptogra fia simétrica.	Sim Não é válido com chaves do KMS em armazena mentos de chaves personali zados.	Não	Não	Não	Não

AWS KMS Operação de API	Chaves do KMS de criptogra fia simétrica	Chaves do KMS de HMAC	Chaves do KMS assimétricas (ENCRYPT DECRYPT)	Chaves do KMS assimétricas (SIGN_VERIFY)	Chaves do KMS assimétricas (KEY_AGREEMENT)
GenerateDataKeyWithPlaintext	Sim	Não	Não	Não	Não
GenerateMac	Não	Sim	Não	Não	Não
GetKeyPolicy	Sim	Sim	Sim	Sim	Sim
GetKeyRotationStatus	Sim	Sim (KeyRotationEnabled sempre será false.)	Sim (KeyRotationEnabled sempre será false.)	Sim (KeyRotationEnabled sempre será false.)	Sim (KeyRotationEnabled sempre será false.)
GetParametersForImport É válido somente em chaves do KMS com material de chave importado (Origin é EXTERNAL).	Sim	Sim	Sim	Sim	Sim
GetPublicKey	Não	Não	Sim	Sim	Sim

AWS KMS Operação de API	Chaves do KMS de criptogra fia simétrica	Chaves do KMS de HMAC	Chaves do KMS assimétri cas (ENCRYPT DECRYPT)	Chaves do KMS assimétricas (SIGN_VERIFY)	Chaves do KMS assimétricas (KEY_AGRE EMENT)
ImportKeyMaterial É válido somente em chaves do KMS com material de chave importado (Origin é EXTERNAL).	Sim	Sim	Sim	Sim	Sim
ListAliases	Sim	Sim	Sim	Sim	Sim
ListGrants	Sim	Sim	Sim	Sim	Sim
ListKeyPolicies	Sim	Sim	Sim	Sim	Sim
ListKeyRotations	Sim	Sim (O campo Rotation sempre será nulo ou vazio.)	Sim (O campo Rotation sempre será nulo ou vazio.)	Sim (O campo Rotations sempre será nulo ou vazio.)	Sim (O campo Rotations sempre será nulo ou vazio.)
ListResourceTags	Sim	Sim	Sim	Sim	Sim
ListRetirableGrants	Sim	Sim	Sim	Sim	Sim
PutKeyPolicy	Sim	Sim	Sim	Sim	Sim

AWS KMS Operação de API	Chaves do KMS de criptogra fia simétrica	Chaves do KMS de HMAC	Chaves do KMS assimétri cas (ENCRYPT DECRYPT)	Chaves do KMS assimétricas (SIGN_VERIFY)	Chaves do KMS assimétricas (KEY_AGRE EMENT)
ReEncrypt	Sim	Não	Sim	Não	Não
ReplicateKey	Sim	Sim	Sim	Sim	Sim
- Válido somente em chaves de várias regiões					
RetireGrant	Sim	Sim	Sim	Sim	Sim
RevokeGrant	Sim	Sim	Sim	Sim	Sim
RotateKey OnDemand	Sim Válido somente para chaves do KMS de criptogra fia simétrica gerenciad as pelo cliente, com a origem AWS_KMS ou EXTERNAL	Não	Não	Não	Não

AWS KMS Operação de API	Chaves do KMS de criptogra fia simétrica	Chaves do KMS de HMAC	Chaves do KMS assimétri cas (ENCRYPT DECRYPT)	Chaves do KMS assimétricas (SIGN_VERIFY)	Chaves do KMS assimétricas (KEY_AGRE EMENT)
ScheduleKeyDeletion	Sim	Sim	Sim	Sim	Sim
Sign	Não	Não	Não	Sim	Não
TagResource	Sim	Sim	Sim	Sim	Sim
UntagResource	Sim	Sim	Sim	Sim	Sim
UpdateAlias	Sim	Sim	Sim	Sim	Sim
A chave do KMS atual e a nova chave do KMS devem ser do mesmo tipo (ambas simétricas, assimétricas ou HMAC) e devem ter o mesmo uso de chave .					
UpdateKeyDescription	Sim	Sim	Sim	Sim	Sim
UpdateReplicaRegion	Sim	Sim	Sim	Sim	Sim
- Válido somente em chaves de várias regiões					

AWS KMS Operação de API	Chaves do KMS de criptogra fia simétrica	Chaves do KMS de HMAC	Chaves do KMS assimétri cas (ENCRYPT DECRYPT)	Chaves do KMS assimétricas (SIGN_VERIFY)	Chaves do KMS assimétricas (KEY_AGRE EMENT)
Verificar	Não	Não	Não	Sim	Não
VerifyMac	Não	Sim	Não	Não	Não

Tabela de recursos especiais

Esta tabela mostra as operações de AWS KMS API que são suportadas em cada tipo de chave de propósito especial.

Ao ler esta tabela, esteja ciente destas interações:

- [Chaves de várias regiões](#):
 - Chaves do KMS multirregionais podem ser chaves de criptografia do KMS simétricas, chaves do KMS assimétricas, chaves do KMS de HMAC e chaves do KMS com material de chave importado.
 - Não é possível criar chaves de várias Regiões em um armazenamento de chaves personalizado.
- [Material de chave importado](#)
 - É possível importar material de chave importado para chave do KMS simétricas, chaves do KMS assimétricas e chaves do KMS de HMAC.
 - É possível [chaves de várias regiões com material de chave importado](#).
 - Você não pode criar chaves do KMS com material de chave em um armazenamento de chaves personalizado.
 - A alternância automática de chaves (`EnableKeyRotation`, `DisableKeyRotation`) não é compatível com chaves do KMS com material de chave importado.
 - A rotação de chaves sob demanda (`RotateKeyOnDemand`) é compatível com chaves KMS de criptografia simétrica com material de chave importado.
- [Armazenamentos de chaves personalizados](#)

- Armazenamentos personalizados de chaves são compatíveis apenas com chaves do KMS de criptografia simétrica.
- Operações simétricas em pares de chaves assimétricas (`GenerateDataKeyPair`, `GenerateDataKeyPairWithoutPlaintext`) não são compatíveis com chaves do KMS em armazenamentos de chaves personalizados.
- A alternância automática de chaves (`EnableKeyRotation`, `DisableKeyRotation`) não é compatível com chaves do KMS em armazenamentos personalizados de chaves.
- Não é possível criar chaves de várias regiões em armazenamentos personalizados de chaves.

Talvez seja necessário rolar horizontalmente ou verticalmente para ver todos os dados nessa tabela.

AWS KMS Operação de API	Chaves de várias regiões	Material de chave importado	Chaves do KMS em um armazenamento de chaves personalizado
CancelKeyDeletion	✓	✓	✓
CreateAlias	✓	✓	✓
CreateGrant	✓	✓	✓
CreateKey Você pode usar <code>CreateKey</code> para criar uma chave primária multirregional, chaves do KMS com material de chave importado ou uma chave do KMS em um armazenamento de chaves personalizado. Para criar uma chave de réplica multirregional, use <code>ReplicateKey</code> .	✓	✓	✓
Decrypt	✓	✓	✓

AWS KMS Operação de API	Chaves de várias regiões	Material de chave importado	Chaves do KMS em um armazenamento de chaves personalizado
	Válido somente quando KeyUsage é ENCRYPT_DECRYPT		
DeleteAlias	✓	✓	✓
DeleteImportedKeyMaterial	✓	✓	✗
	É válido somente para chaves do KMS com material de chave importado (Origin é EXTERNAL)		
DeriveSharedSecret	✓	✓	✗
	Válido somente quando KeyUsage é KEY_AGREEMENT	Válido somente quando KeyUsage é KEY_AGREEMENT	
DescribeKey	✓	✓	✓
DisableKey	✓	✓	✓

AWS KMS Operação de API	Chaves de várias regiões	Material de chave importado	Chaves do KMS em um armazenamento de chaves personalizado
DisableKeyRotation	 Válido somente em chaves de criptografia simétricas com material (OriginsAWS_KMS de AWS KMS chave.		
EnableKey	 Válido somente em chaves do KMS de criptografia simétricas		
EnableKeyRotation	 Válido somente em chaves de criptografia simétricas com material (OriginsAWS_KMS de AWS KMS chave.		

AWS KMS Operação de API	Chaves de várias regiões	Material de chave importado	Chaves do KMS em um armazenamento de chaves personalizado
Encrypt	 Válido somente quando KeyUsage é ENCRYPT_D ECRYPT		
GenerateDataKey	 Válido somente em chaves do KMS de criptografia simétricas		
GenerateDataKeyPair	 Válido somente em chaves do KMS de criptografia simétricas		

AWS KMS Operação de API	Chaves de várias regiões	Material de chave importado	Chaves do KMS em um armazenamento de chaves personalizado
GenerateDataKeyPairWithoutPlaintext	 Válido somente em chaves do KMS de criptografia simétricas		
GenerateDataKeyWithoutPlaintext	 Válido somente em chaves do KMS de criptografia simétricas		
GenerateMac Válido somente em chaves do KMS de HMAC			
GetKeyPolicy			
GetKeyRotationStatus		 (KeyRotationEnabled sempre será false.)	

AWS KMS Operação de API	Chaves de várias regiões	Material de chave importado	Chaves do KMS em um armazenamento de chaves personalizado
GetParametersForImport	 É válido somente para chaves do KMS com material de chave importado (Origin é EXTERNAL).		
GetPublicKey Válido somente para chaves do KMS assimétricas .			
ImportKeyMaterial	 É válido somente para chaves do KMS com material de chave importado (Origin é EXTERNAL).		
ListAliases			
ListGrants			
ListKeyPolicies			

AWS KMS Operação de API	Chaves de várias regiões	Material de chave importado	Chaves do KMS em um armazenamento de chaves personalizado
ListKeyRotations	 Válido somente em chaves de criptografia simétricas com EXTERNAL origem AWS_KMS ou.	 Válido somente em chaves de criptografia simétricas.	
ListResourceTags			
ListRetirableGrants			
PutKeyPolicy			
ReEncrypt	 Válido somente quando KeyUsage é ENCRYPT_DECRYPT		

AWS KMS Operação de API	Chaves de várias regiões	Material de chave importado	Chaves do KMS em um armazenamento de chaves personalizado
ReplicateKey	✓ Válido somente em chaves primárias multirregionais.	✓ Válido somente em chaves primárias multirregionais.	✗
RetireGrant	✓	✓	✓
RevokeGrant	✓	✓	✓
RotateKeyOnDemand	✓ Válido somente em chaves de criptografia simétricas com EXTERNAL origem AWS_KMS ou.	✓ Válido somente em chaves de criptografia simétricas.	✗
ScheduleKeyDeletion	✓	✓	✓
Sign Válido apenas quando KeyUsage é SIGN_VERIFY .	✓	✓	✗
TagResource	✓	✓	✓

AWS KMS Operação de API	Chaves de várias regiões	Material de chave importado	Chaves do KMS em um armazenamento de chaves personalizado
UntagResource	✓	✓	✓
UpdateAlias - A chave do KMS atual e a nova chave do KMS devem ser do mesmo tipo (ambas simétricas, assimétricas ou HMAC) e devem ter o mesmo uso de chave .	✓	✓	✓
UpdateKeyDescription	✓	✓	✓
UpdateReplicaRegion	✓	✓	✗
		Válido somente em chaves multirregionais.	
Verificar Válido somente quando KeyUsage é SIGN_VERIFY .	✓	✓	✗
VerifyMac Válido somente em chaves do KMS de HMAC	✓	✓	✗

Referência de especificação de chave

Ao criar uma chave do KMS assimétrica ou uma chave do KMS de HMAC, selecione sua [especificação de chave](#). A especificação da chave, que é uma propriedade de cada AWS KMS key, representa a configuração criptográfica da sua chave KMS. Você escolhe a especificação de chave ao criar a chave do KMS e não pode alterá-la. Se você tiver escolhido a especificação de chave errada, [exclua a chave do KMS](#) e crie outra.

Note

A especificação de chave para uma chave do KMS era conhecida como “especificação de chave mestra do cliente”. O `CustomerMasterKeySpec` parâmetro da [CreateKey](#) operação está obsoleto. Em vez disso, use o parâmetro `KeySpec`. A resposta das [DescribeKey](#) operações `CreateKey` e inclui um `CustomerMasterKeySpec` membro `KeySpec` e com o mesmo valor.

A especificação da chave determina se a chave KMS é simétrica ou assimétrica, o tipo de material da chave na chave KMS e os algoritmos de criptografia, algoritmos de assinatura ou algoritmos de código de autenticação de mensagem (MAC) que AWS KMS suportam a chave KMS. A especificação de chave escolhida normalmente é determinada pelo caso de uso e pelos requisitos regulatórios. No entanto, as operações criptográficas em chaves do KMS com especificações de chave diferentes são cobradas de maneira diferente e estão sujeitas a cotas diferentes. Para obter detalhes sobre os preços, consulte [AWS Key Management Service Pricing](#) (Preços do). Para obter mais informações sobre cotas de solicitações, consulte [Cotas de solicitações](#).

Para limitar as principais especificações que os diretores podem usar ao criar chaves KMS, use a chave [kms: condition](#). `KeySpec` Você também pode usar a chave de `kms:KeySpec` condição para permitir que os diretores chamem AWS KMS operações somente em chaves KMS com uma especificação de chave específica. Por exemplo, é possível negar permissão para programar a exclusão de qualquer chave do KMS com uma especificação de chave `RSA_4096`.

AWS KMS suporta as seguintes especificações principais para chaves KMS:

[Especificações da chave de criptografia simétrica](#) (padrão)

- `SYMMETRIC_DEFAULT`

[Especificações de chave RSA](#) (criptografia e descriptografia ou assinatura e verificação)

- `RSA_2048`

- RSA_3072
- RSA_4096

Especificações da chave de curva elíptica

- [Pares de chaves de curva elíptica](#) padrão NIST assimétrica (assinatura e verificação ou derivação de segredos compartilhados)
 - ECC_NIST_P256 (secp256r1)
 - ECC_NIST_P384 (secp384r1)
 - ECC_NIST_P521 (secp521r1)
 - ECC_NIST_EDWARDS25519 (ed25519) - somente assinatura e verificação
 - Nota: Para chaves EDWARDS25519 KMS ECC_NIST_, o algoritmo de assinatura _SHA_512 exige, enquanto ED25519 _PH_SHA_512 exige. [MessageType:RAW](#) [ED25519MessageType:DIGEST](#) Esses tipos de mensagem não podem ser usados de forma intercambiável.
- Outros pares de chaves de curva elíptica assimétricas (assinatura e verificação)
 - ECC_SECG_P256K1 ([secp256k1](#)), normalmente usado para criptomoedas.

[SM2 especificação chave](#) (criptografia e descryptografia -ou- assinatura e verificação -ou- derivação de segredos compartilhados)

- SM2 (Somente regiões da China)

Especificações de chave de HMAC

- HMAC_224
- HMAC_256
- HMAC_384
- HMAC_512

Especificações de chave ML-DSA

- ML_DSA_44
- ML_DSA_65
- ML_DSA_87

Especificação da chave SYMMETRIC_DEFAULT

A especificação de chave padrão, SYMMETRIC_DEFAULT, é a especificação de chave para chaves do KMS de criptografia simétrica. Quando você seleciona o tipo de chave simétrica e o uso

da chave de criptografia e descriptografia no AWS KMS console, ele seleciona a especificação da chave. SYMMETRIC_DEFAULT Na [CreateKey](#) operação, se você não especificar um KeySpec valor, SYMMETRIC_DEFAULT será selecionado. Se você não tiver um motivo para usar uma especificação de chave diferente, SYMMETRIC_DEFAULT é uma boa escolha.

SYMMETRIC_DEFAULT representa o AES-256-GCM, um algoritmo simétrico com base no [Advanced Encryption Standard](#) (AES) em [Galois Counter Mode](#) (GCM) com chaves de 256 bits, um padrão do setor para criptografia segura. O texto cifrado que esse algoritmo gera oferece suporte a dados adicionais autenticados (ADD), como um [contexto de criptografia](#) e o GCM fornece uma verificação de integridade adicional no texto cifrado.

Os dados criptografados em AES-256-GCM está protegido agora e futuramente. Os criptógrafos consideram esse algoritmo como resistente a quânticos. Num futuro teórico, ataques de computação quântica em grande escala em textos cifrados criados sob chaves AES-GCM de 256 bits [reduzem a segurança efetiva da chave para 128 bits](#). Mas esse nível de segurança é suficiente para inviabilizar ataques de força bruta em textos AWS KMS cifrados.

A única exceção nas regiões da China, onde SYMMETRIC_DEFAULT representa uma chave simétrica de 128 bits que usa criptografia. SM4 Você só pode criar uma SM4 chave de 128 bits nas regiões da China. Uma chave do KMS AES-GCM de 256 bits não pode ser criada nas regiões da China.

Você pode usar uma chave KMS de criptografia simétrica AWS KMS para criptografar, descriptografar e recriptografar dados e para proteger chaves de dados e pares de chaves de dados gerados. AWS os serviços integrados AWS KMS usam chaves KMS de criptografia simétrica para criptografar seus dados em repouso. Você pode [importar seu próprio material de chave](#) para uma chave do KMS de criptografia simétrica e criar chaves do KMS de criptografia simétrica em [armazenamentos personalizados de chaves](#). Para acessar uma tabela comparando as operações que podem ser executadas em chaves do KMS simétricas e assimétricas, consulte [Comparar chaves do KMS simétricas e assimétricas](#).

Você pode usar uma chave KMS de criptografia simétrica AWS KMS para criptografar, descriptografar e recriptografar dados e gerar chaves de dados e pares de chaves de dados. Você pode criar chaves do KMS de criptografia simétrica [de várias regiões](#), [importar seu próprio material de chaves](#) para uma chave do KMS de criptografia simétrica e criar chaves do KMS de criptografia simétrica em [armazenamentos personalizados de chaves](#). Para obter uma tabela comparando as operações que podem ser realizadas em chaves do KMS de diferentes tipos, consulte [Referência de tipos de chaves](#).

Especificações da chave RSA

Quando você usa uma especificação de chave RSA, AWS KMS cria uma chave KMS assimétrica com um par de chaves RSA. A chave privada nunca sai AWS KMS sem criptografia. Você pode usar a chave pública dentro AWS KMS ou baixar a chave pública para uso externo AWS KMS.

Warning

Ao criptografar dados fora de AWS KMS, certifique-se de poder descriptografar seu texto cifrado. Se você usar a chave pública de uma chave do KMS que foi excluída do AWS KMS, a chave pública de uma chave do KMS configurada para assinatura e verificação ou um algoritmo de criptografia não compatível com a chave do KMS, os dados se tornarão irre recuperáveis.

Em AWS KMS, você pode usar chaves KMS assimétricas com pares de chaves RSA para criptografia e decodificação, ou assinatura e verificação, mas não ambas. Essa propriedade, conhecida como [Key usage](#), é determinada separadamente da especificação de chave, mas você deve tomar essa decisão antes de escolher uma especificação de chave.

AWS KMS suporta as seguintes especificações de chave RSA para criptografia e decodificação ou assinatura e verificação:

- RSA_2048
- RSA_3072
- RSA_4096

As especificações de chave RSA diferem no tamanho da chave RSA em bits. A especificação da chave RSA escolhida deve ser determinada pelos padrões de segurança ou pelos requisitos da tarefa. Em geral, use a maior chave que seja prática e acessível para sua tarefa. As operações criptográficas em chaves do KMS com diferentes especificações de chave RSA são cobradas de maneira diferente. Para obter informações sobre AWS KMS preços, consulte [AWS Key Management Service Pricing](#). Para obter mais informações sobre cotas de solicitações, consulte [Cotas de solicitações](#).

Especificações de chave RSA para criptografia e descriptografia

Quando uma chave do KMS RSA assimétrica é usada para criptografia e descriptografia, você criptografa com a chave pública e descriptografa com a chave privada. Quando você chama a `Encrypt` operação AWS KMS para obter uma chave RSA KMS, AWS KMS usa a chave pública no par de chaves RSA e o algoritmo de criptografia que você especifica para criptografar seus dados. Para descriptografar o texto cifrado, chame a `Decrypt` operação e especifique a mesma chave KMS e algoritmo de criptografia. AWS KMS em seguida, usa a chave privada no par de chaves RSA para descriptografar seus dados.

Você também pode baixar a chave pública e usá-la para criptografar dados externos. AWS KMS Certifique-se de usar um algoritmo de criptografia AWS KMS compatível com chaves RSA KMS. Para descriptografar o texto cifrado, chame a função `Decrypt` com a mesma chave do KMS e o mesmo algoritmo de criptografia.

AWS KMS suporta dois algoritmos de criptografia para chaves KMS com especificações de chave RSA. Esses algoritmos, que são definidos no [PKCS nº 1 v2.2](#), diferem na função de hash que usam internamente, [Em AWS KMS, os algoritmos RSAES_OAEP sempre usam a mesma função de hash para fins de hash e para a função de geração de máscara \(\)](#). MGF1 É obrigatório especificar um algoritmo de criptografia ao chamar as operações [Encrypt](#) e [Decrypt](#). É possível escolher um algoritmo diferente para cada solicitação.

Algoritmos de criptografia compatíveis com especificações de chave RSA

Algoritmo de criptografia	Descrição do algoritmo
RSAES_OAEP_SHA_1	PKCS nº 1 v2.2, Seção 7.1. Criptografia RSA com preenchimento OAEP usando SHA-1 para o hash e na função de geração de MGF1 máscara junto com um rótulo vazio.
RSAES_OAEP_SHA_256	PKCS nº 1, Seção 7.1. Criptografia RSA com preenchimento OAEP usando SHA-256 para o hash e na função de geração de MGF1 máscara junto com um rótulo vazio.

Não é possível configurar uma chave do KMS para usar um algoritmo de criptografia específico. No entanto, você pode usar a condição [kms: EncryptionAlgorithm](#) policy para especificar os algoritmos de criptografia que os principais podem usar com a chave KMS.

Para obter os algoritmos de criptografia de uma chave KMS, [visualize a configuração criptográfica](#) da chave KMS no AWS KMS console ou use a operação. [DescribeKey](#) AWS KMS também fornece a especificação da chave e os algoritmos de criptografia quando você baixa sua chave pública, seja no AWS KMS console ou usando a [GetPublicKey](#) operação.

Você pode escolher uma especificação de chave RSA com base no tamanho dos dados do texto não criptografado que pode ser criptografado em cada solicitação. A tabela a seguir mostra o tamanho máximo, em bytes, do texto não criptografado que pode ser criptografado em uma única chamada para a operação [Encrypt](#). Os valores diferem de acordo com a especificação de chave e com o algoritmo de criptografia. Para fins comparativos, é possível usar uma chave do KMS de criptografia simétrica para criptografar até 4.096 bytes de uma vez.

Para calcular o tamanho máximo de texto simples em bytes para esses algoritmos, use a seguinte fórmula: $(key\_size\_in\_bits/8) - (2 * hash\_length\_in\_bits / 8) - 2$. Por exemplo, para RSA_2048 com SHA-256, o tamanho máximo de texto não criptografado em bytes é de $(2048/8) - (2 * 256/8) - 2 = 190$.

Tamanho máximo de texto não criptografado (em bytes) em uma operação Encrypt

	Algoritmo de criptografia	
Especificação da chave	RSAES_OAEP_SHA_1	RSAES_OAEP_SHA_256
RSA_2048	214	190
RSA_3072	342	318
RSA_4096	470	446

Especificações de chave RSA para assinatura e verificação

Quando uma chave do KMS RSA assimétrica é usada para assinatura e verificação, você gera a assinatura para uma mensagem com uma chave privada e verifica a assinatura com a chave pública.

Quando você chama a Sign operação de uma chave KMS assimétrica, AWS KMS usa a chave privada no par de chaves RSA, a mensagem e o algoritmo de assinatura que você especifica para

gerar uma assinatura. AWS KMS Para verificar a assinatura, chame a operação [Verify](#). Especifique a assinatura, além da mesma chave KMS, mensagem e algoritmo de assinatura. AWS KMS em seguida, usa a chave pública no par de chaves RSA para verificar a assinatura. Você também pode baixar a chave pública e usá-la para verificar a assinatura fora do AWS KMS.

AWS KMS suporta os seguintes algoritmos de assinatura para todas as chaves KMS com uma especificação de chave RSA. É obrigatório especificar um algoritmo de assinatura ao chamar as operações [Sign](#) e [Verify](#). É possível escolher um algoritmo diferente para cada solicitação. Ao assinar com pares de chaves RSA, os algoritmos RSASSA-PSS são preferidos. Incluímos algoritmos RSASSA- PKCS1 -v1_5 para compatibilidade com aplicativos existentes.

Algoritmos de assinatura compatíveis com especificações de chave RSA

Algoritmo de assinatura	Descrição do algoritmo
RSASSA_PSS_SHA_256	PKCS #1 v2.2, Seção 8.1, Assinatura RSA com preenchimento PSS usando SHA-256 para o resumo da mensagem e a função de geração de máscara junto com um sal de 256 bits MGF1
RSASSA_PSS_SHA_384	PKCS #1 v2.2, Seção 8.1, Assinatura RSA com preenchimento PSS usando SHA-384 para o resumo da mensagem e a função de geração de máscara junto com um sal de 384 bits MGF1
RSASSA_PSS_SHA_512	PKCS #1 v2.2, Seção 8.1, Assinatura RSA com preenchimento PSS usando SHA-512 para o resumo da mensagem e a função de geração de máscara junto com um sal de 512 bits MGF1
PKCS1RASSA_ _V1_5_SHA_256	PKCS nº 1 v2.2, Seção 8.2, assinatura RSA com PKCS nº 1 v1.5 padding e SHA-256
PKCS1RASSA_ _V1_5_SHA_384	PKCS nº 1 v2.2, Seção 8.2, assinatura RSA com PKCS nº 1 v1.5 padding e SHA-384

Algoritmo de assinatura	Descrição do algoritmo
PKCS1RASSA_ _V1_5_SHA_512	PKCS nº 1 v2.2, Seção 8.2, assinatura RSA com PKCS nº 1 v1.5 padding e SHA-512

Não é possível configurar uma chave do KMS para usar algoritmos de assinatura específicos. No entanto, você pode usar a condição [kms: SigningAlgorithm](#) policy para especificar os algoritmos de assinatura que os diretores podem usar com a chave KMS.

Para obter os algoritmos de assinatura de uma chave KMS, [visualize a configuração criptográfica](#) da chave KMS no AWS KMS console ou usando a operação. [DescribeKey](#) AWS KMS também fornece a especificação da chave e os algoritmos de assinatura quando você baixa sua chave pública, seja no AWS KMS console ou usando a [GetPublicKey](#) operação.

Especificações da chave de curva elíptica

Quando você usa uma especificação de chave de curva elíptica (ECC), AWS KMS cria uma chave KMS assimétrica com um par de chaves ECC para assinatura e verificação ou derivação de segredos compartilhados (mas não ambos). A chave privada que gera assinaturas ou deriva segredos compartilhados nunca sai descriptografada do AWS KMS. Você pode usar a chave pública para [verificar as assinaturas](#) dentro AWS KMS ou [fazer o download da chave pública](#) para uso AWS KMS externo.

Quando você usa uma especificação de chave Edwards Curve, AWS KMS cria uma chave KMS assimétrica com um par de chaves Ed25519 para assinatura e verificação. A chave privada que gera assinaturas nunca sai AWS KMS sem criptografia. Você pode usar a chave pública para [verificar as assinaturas](#) dentro AWS KMS ou [fazer o download da chave pública](#) para uso AWS KMS externo.

AWS KMS suporta as seguintes especificações de chave ECC para chaves KMS assimétricas.

- Pares de chaves de curva elíptica padrão NIST assimétrica (assinatura e verificação ou derivação de segredos compartilhados)
 - ECC_NIST_P256 (secp256r1)
 - ECC_NIST_P384 (secp384r1)
 - ECC_NIST_P521 (secp521r1)
 - ECC_NIST_EDWARDS25519 (ed25519) - somente assinatura e verificação

- Nota: Para chaves EDWARDS25519 KMS ECC_NIST_, o algoritmo de assinatura _SHA_512 exige, enquanto ED25519 _PH_SHA_512 exige. [MessageType:RAW](#) [ED25519MessageType:DIGEST](#) Esses tipos de mensagem não podem ser usados de forma intercambiável.
- Outros pares de chaves de curva elíptica assimétricas (assinatura e verificação)
 - ECC_SECG_P256K1 ([secp256k1](#)), normalmente usada para criptomoedas.

A especificação da chave ECC escolhida deve ser determinada pelos padrões de segurança ou pelos requisitos da tarefa. Em geral, use a curva com mais pontos que seja prática e acessível para sua tarefa.

Se você estiver criando uma chave KMS assimétrica para [derivar segredos compartilhados](#), use uma das especificações-chave de curva elíptica padrão do NIST (exceto ECC_SECG_P256K1 e ECC_NIST_). EDWARDS25519 O único algoritmo de acordo de chave compatível para derivar segredos compartilhados é o [Elliptic Curve Cryptography Cofactor Diffie-Hellman Primitive](#) (ECDH). Para obter um exemplo de como derivar segredos compartilhados offline, consulte [the section called “Derivar segredos compartilhados offline”](#).

Se você estiver criando uma chave do KMS assimétrica para usar com criptomoedas, use a especificação de chave ECC_SECG_P256K1. Também é possível usar essa especificação de chave para outros fins, mas ela é exigida para Bitcoin e para outras criptomoedas.

A tabela a seguir mostra os algoritmos de assinatura AWS KMS compatíveis com cada uma das principais especificações do ECC. Não é possível configurar uma chave do KMS para usar algoritmos de assinatura específicos. No entanto, você pode usar a condição [kms: SigningAlgorithm](#) policy para especificar os algoritmos de assinatura que os diretores podem usar com a chave KMS.

Algoritmos de assinatura compatíveis para especificações de chave ECC

Especificação da chave	Algoritmo de assinatura	Descrição do algoritmo
ECC_NIST_P256	ECDSA_SHA_256	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-256 para o resumo da mensagem.

Especificação da chave	Algoritmo de assinatura	Descrição do algoritmo
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 para o resumo da mensagem.
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 para o resumo da mensagem.
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 para o resumo da mensagem.
ECC_NIST_EDWARDS25519	ED25519_SHA_512	NIST FIPS 186-5, Section 7, EdDSA signature using the curve specified by the key and SHA-512 para o resumo da mensagem. O KMS exige MessageType:RAW esse algoritmo.
ECC_NIST_EDWARDS25519	ED25519_PH_SHA_512	NIST FIPS 186-5, Section 7, EdDSA signature using the curve specified by the key and SHA-512 para o resumo da mensagem. O KMS exige MessageType:DIGEST esse algoritmo.

Especificações de chave para chaves do KMS de HMAC

AWS KMS suporta chaves HMAC simétricas em comprimentos variados. A especificação de chave que você escolhe depende de seus requisitos regulatórios, de segurança ou de negócios. O comprimento da chave determina o algoritmo MAC usado nas [GenerateMacVerifyMac](#) operações. Em geral, chaves mais longas são mais seguras. Use a chave mais longa que seja viável para o seu caso de uso.

Especificação de chave de HMAC	Algoritmo de MAC
HMAC_224	HMAC_SHA_224
HMAC_256	HMAC_SHA_256
HMAC_384	HMAC_SHA_384
HMAC_512	HMAC_SHA_512

Especificações de chave ML-DSA

Uma chave ML-DSA é uma chave criptográfica usada no Algoritmo de Assinatura Module-Lattice-Based Digital (ML-DSA), projetado para criptografia pós-quântica. Este algoritmo faz parte do esforço de padronização do NIST (National Institute of Standards and Technology), especificamente descrito nos [Federal Information Processing Standards \(FIPS\) 204](#).

As chaves ML-DSA são usadas em um sistema de par de chaves, pública e privada. A chave privada é usada para assinar dados, enquanto a chave pública é usada para verificar a assinatura. Esse sistema garante a autenticidade, integridade e não repúdio de mensagens ou documentos digitais, mesmo diante de possíveis ameaças de computação quântica.

Quando você cria uma chave com a especificação de chave ML-DSA, AWS KMS cria uma chave KMS assimétrica com um par de chaves ML-DSA para assinatura e verificação. A chave privada que gera assinaturas nunca sai AWS KMS sem criptografia. Você pode usar a chave pública para [verificar as assinaturas](#) dentro AWS KMS ou [fazer o download da chave pública](#) para uso AWS KMS externo.

AWS KMS suporta as seguintes especificações de chave ML-DSA para chaves KMS assimétricas:

- ML_DSA_44

- ML_DSA_65
- ML_DSA_87

AWS KMS suporta o algoritmo de assinatura ML_DSA_SHAKE_256 para todas as especificações principais do ML-DSA.

SM2 especificação chave (somente regiões da China)

A especificação SM2 chave é uma especificação chave de curva elíptica definida dentro da GM/T série de especificações publicadas pelo [Escritório de Administração Estatal de Criptografia Comercial \(OSCCA\) da China](#). A especificação SM2 principal está disponível somente nas regiões da China. Quando você usa a especificação de SM2 chave, AWS KMS cria uma chave KMS assimétrica com um par de chaves. SM2 Você pode usar seu par de SM2 chaves dentro AWS KMS ou fazer o download da chave pública para uso externo AWS KMS. Para obter mais informações, consulte [the section called “Verificação off-line com pares de SM2 chaves \(somente regiões da China\)”](#).

Cada chave do KMS pode ter somente um [Key usage](#). Você pode usar uma chave SM2 KMS para assinar e verificar, criptografar e descriptografar ou derivar segredos compartilhados. Você deve especificar o uso da chave ao criar a chave do KMS, e não será possível alterá-lo após a criação da chave.

Se você estiver criando uma chave KMS assimétrica para [derivar segredos compartilhados](#), use a especificação de chave. SM2 O único algoritmo de acordo de chave compatível para derivar segredos compartilhados é o [Elliptic Curve Cryptography Cofactor Diffie-Hellman Primitive](#) (ECDH).

AWS KMS suporta os seguintes algoritmos de SM2 criptografia e assinatura:

- SM2Algoritmo de criptografia PKE

SM2O PKE é um algoritmo de criptografia baseado em curva elíptica definido pela OSCCA em 0003.4-2012. GM/T

- SM2Algoritmo de assinatura DSA

SM2O DSA é um algoritmo de assinatura baseado em curva elíptica definido pela OSCCA em 0003.2-2012. GM/T SM2O DSA exige uma ID distintiva que seja codificada com o algoritmo de SM3 hash e depois combinada com a mensagem, ou resumo da mensagem, para a qual você passou. AWS KMS Esse valor concatenado é então codificado e assinado por. AWS KMS

AWS KMS permissões

Essa tabela foi criada para ajudar você a entender AWS KMS as permissões para que você possa controlar o acesso aos seus AWS KMS recursos. As definições dos cabeçalhos das colunas aparecem abaixo da tabela.

Você também pode aprender sobre AWS KMS permissões no AWS Key Management Service tópico [Ações, recursos e chaves de condição](#) do Service Authorization Reference. No entanto, esse tópico não lista todas as chaves de condição que você pode usar para refinar cada permissão.

Para obter mais informações sobre quais AWS KMS operações são válidas para chaves KMS de criptografia simétrica, chaves KMS assimétricas e chaves HMAC KMS, consulte o [Referência de tipos de chaves](#)

Note

Talvez seja necessário rolar horizontalmente ou verticalmente para ver todos os dados da tabela.

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
CancelKeyDeletion kms:CancelKeyDeletion	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
				kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService
ConnectCustomKeyStore kms:ConnectCustomKeyStore	Política do IAM	Não	*	kms: CallerAccount

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
CreateAlias kms:CreateAlias	Política do IAM (para o alias)	Não	Alias	Nenhum (quando controlar o acesso ao alias)
Para usar essa operação, o chamador precisa da permissão kms:CreateAlias em dois recursos: - O alias (em uma política do IAM) - A chave do KMS (em uma política de chaves) Para obter detalhes, consulte Controlar o acesso a aliases.	Política de chaves (para a chave do KMS)	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService
CreateCustomKeyStore kms:CreateCustomKeyStore	Política do IAM	Não	*	kms: CallerAccount

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
CreateGrant kms:CreateGrant	Política de chave	Sim	Chave do KMS	Condições para contexto de criptografia: kms:EncryptionContext: chave de contexto kms: EncryptionContextKeys Condições de concessão: kms: GrantConstraintType kms: GranteePrincipal kms: GrantsFor AWSResource kms: GrantOperations kms: RetiringPrincipal Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
				kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
CreateKey kms:CreateKey	Política do IAM	Não	*	kms: BypassPolicyLockoutSafetyCheck kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ViaService aws:RequestTag/tag-key (chave de condição AWS global) aws:ResourceTag/tag-key (chave de condição AWS global) aws: TagKeys (chave de condição AWS global)

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
Decrypt kms:Decrypt	Política de chave	Sim	Chave do KMS	Condições para operações criptográficas kms: EncryptionAlgorithm kms: RequestAlias Condições para contexto de criptografia: kms:EncryptionContext: chave de contexto kms: EncryptionContextKeys Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
				aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService
DeleteAlias kms:DeleteAlias Para usar essa operação, o chamador precisa da permissão kms:DeleteAlias em dois recursos: • O alias (em uma política do IAM) • A chave do KMS (em uma política de chaves) Para obter detalhes, consulte Controlar o acesso a aliases .	Política do IAM (para o alias)	Não	Alias	Nenhum (quando controlar o acesso ao alias)
	Política de chaves (para a chave do KMS)	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
DeleteCustomKeyStore kms:DeleteCustomKeyStore	Política do IAM	Não	*	kms: CallerAccount
DeleteImportedKeyMaterial kms:DeleteImportedKeyMaterial	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
DeriveSharedSecret kms:DeriveSharedSecret	Política de chave	Sim	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Condições para operações criptográficas: kms: KeyAgreementAlgorithm
DescribeCustomKeyStores kms:DescribeCustomKeyStores	Política do IAM	Não	*	kms: CallerAccount

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
DescribeKey kms:DescribeKey	Política de chave	Sim	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Outras condições: kms: RequestAlias

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
DisableKey kms:DisableKey	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
DisableKeyRotation kms:DisableKeyRotation	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService
DisconnectCustomKeyStore kms:DisconnectCustomKeyStore	Política do IAM	Não	*	kms: CallerAccount

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
EnableKey <code>kms:EnableKey</code>	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
EnableKeyRotation <code>kms:EnableKeyRotation</code>	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Condições de alternância automática de chaves: kms: RotationPeriodInDays

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
Encrypt kms:Encrypt	Política de chave	Sim	Chave do KMS	Condições para operações criptográficas kms: EncryptionAlgorithm kms: RequestAlias Condições para contexto de criptografia: kms:EncryptionContext: chave de contexto kms: EncryptionContextKeys Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
				aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
GenerateDataKey kms:GenerateDataKey	Política de chave	Sim	Chave do KMS	Condições para operações criptográficas kms: EncryptionAlgorithm kms: RequestAlias Condições para contexto de criptografia: kms:EncryptionContext: chave de contexto kms: EncryptionContextKeys Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
				aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
GenerateDataKeyPair <code>kms:GenerateDataKeyPair</code>	Política de chave	Sim	Chave KMS Gera um par de chaves de dados assimétricas que é protegido por uma chave do KMS de criptografia simétrica.	Condições para pares de chaves de dados: kms: DataKeyPairSpec Condições para operações criptográficas kms: EncryptionAlgorithm kms: RequestAlias Condições para contexto de criptografia: kms:EncryptionContext:chave de contexto kms: EncryptionContextKeys Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
				kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
GenerateDataKeyPairWithoutPlaintext <code>kms:GenerateDataKeyPairWithoutPlaintext</code>	Política de chave	Sim	Chave KMS Gera um par de chaves de dados assimétricas que é protegido por uma chave do KMS de criptografia simétrica.	Condições para pares de chaves de dados: kms: DataKeyPairSpec Condições para operações criptográficas kms: EncryptionAlgorithm kms: RequestAlias Condições para contexto de criptografia: kms:EncryptionContext: chave de contexto kms: EncryptionContextKeys Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
				kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
GenerateDataKeyWithoutPlaintext kms:GenerateDataKeyWithoutPlaintext	Política de chave	Sim	Chave do KMS	Condições para operações criptográficas kms: EncryptionAlgorithm kms: RequestAlias Condições para contexto de criptografia: kms:EncryptionContext: chave de contexto kms: EncryptionContextKeys Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
				aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService
GenerateMac kms:GenerateMac	Política de chave	Sim	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Condições para operações criptográficas: kms: MacAlgorithm kms: RequestAlias

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
GenerateRandom kms:GenerateRandom	Política do IAM	N/D	*	Nenhum
GetKeyPolicy kms:GetKeyPolicy	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
GetKeyRotationStatus kms:GetKeyRotationStatus	Política de chave	Sim	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
GetParametersForImport <code>kms:GetParametersForImport</code>	Política de chave	Não	Chave KMS	kms: WrappingAlgorithm kms: WrappingKeySpec Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
GetPublicKey <code>kms:GetPublicKey</code>	Política de chave	Sim	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Outras condições: kms: RequestAlias

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
ImportKeyMaterial <code>kms:ImportKeyMaterial</code>	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Outras condições: kms: ExpirationModel kms: ValidTo
ListAliases <code>kms:ListAliases</code>	Política do IAM	Não	*	Nenhum

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
ListGrants kms:ListGrants	Política de chave	Sim	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Outras condições: kms: GrantsForAWSResource

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
ListKeyPolicies kms:ListKeyPolicies	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
ListKeyRotations kms:ListKeyRotations	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService
ListKeys kms:ListKeys	Política do IAM	Não	*	Nenhum

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
ListResourceTags kms:ListResourceTags	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
ListRetirableGrants kms:ListRetirableGrants	Política do IAM	A entidade principal especificada deve estar na conta local, mas a operação retorna concessões em todas as contas.	*	Nenhum

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
PutKeyPolicy kms:PutKeyPolicy	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Outras condições: kms: BypassPolicyLockoutSafetyCheck

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
ReEncrypt <code>kms:ReEncryptFrom</code> <code>kms:ReEncryptTo</code> Para usar essa operação, o autor da chamada precisa de permissão em duas chaves do KMS: <code>kms:ReEncryptFrom</code> na chave do KMS usada para descriptografar <code>kms:ReEncryptTo</code> na chave do KMS usada para criptografar	Política de chave	Sim	Chave do KMS	Condições para operações criptográficas kms: EncryptionAlgorithm kms: RequestAlias Condições para contexto de criptografia: kms:EncryptionContext: chave de contexto kms: EncryptionContextKeys Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
				aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Outras condições: kms: ReEncryptOnSameKey

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
ReplicateKey <code>kms:ReplicateKey</code> Para usar essa operação, o autor da chamada precisa das seguintes permissões: <code>kms:ReplicateKey</code> na chave primária de várias regiões <code>kms:CreateKey</code> em uma política do IAM na região de réplica	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Outras condições: kms: ReplicaRegion

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
RetireGrant <code>kms:RetireGrant</code> A permissão para retirar uma concessão é determinada principalmente pela concessão. Uma política por si só não pode permitir o acesso a essa operação. Para obter mais informações, consulte Retirar e revogar concessões.	Política do IAM (Essa permissão não é eficaz em uma política de chave.)	Sim	Chave do KMS	Condições para contexto de criptografia: kms:EncryptionContext: chave de contexto kms: EncryptionContextKeys Condições de concessão: kms: GrantConstraintType Condições para operações de chave do KMS: kms: CallerAccount kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
RevokeGrant kms:RevokeGrant	Política de chave	Sim	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Outras condições: kms: GrantsForAWSResource

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
RotateKeyOnDemand <code>kms:RotateKeyOnDemand</code>	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
ScheduleKeyDeletion <code>kms:ScheduleKeyDeletion</code>	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
Sign kms:Sign	Política de chave	Sim	Chave KMS	Condições para assinatura e verificação: kms: MessageType kms: RequestAlias kms: SigningAlgorithm Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
TagResource kms:TagResource	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Condições para marcação: aws:RequestTag/tag-key (chave de condição AWS global) aws: TagKeys (chave de condição AWS global)

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
UntagResource kms:UntagResource	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Condições para marcação: aws:RequestTag/tag-key (chave de condição AWS global) aws: TagKeys (chave de condição AWS global)

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
UpdateAlias kms:UpdateAlias	Política do IAM (para o alias)	Não	Alias	Nenhum (quando controlar o acesso ao alias)
Para usar essa operação, o chamador precisa da permissão kms:UpdateAlias em três recursos: • O alias • A chave do KMS atualmente associada • A chave do KMS recém-associada Para obter detalhes, consulte Controlar o acesso a aliases.	Política de chaves (para as chaves do KMS)	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService
UpdateCustomKeyStore kms:UpdateCustomKeyStore	Política do IAM	Não	*	kms: CallerAccount

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
UpdateKeyDescription kms:UpdateKeyDescription	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
UpdatePrimaryRegion <code>kms:UpdatePrimaryRegion</code> Para usar essa operação, o autor da chamada precisa da permissão <code>kms:UpdatePrimaryRegion</code> na chave primária de várias regiões que se tornará uma chave de réplica e também na chave de réplica de várias regiões que se tornará a chave primária.	Política de chave	Não	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Outras condições kms: PrimaryRegion

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
Verificar kms:Verify	Política de chave	Sim	Chave KMS	Condições para assinatura e verificação: kms: MessageType kms: RequestAlias kms: SigningAlgorithm Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService

Ações e permissões	Tipo de política	Uso entre contas	Recursos (para políticas do IAM)	AWS KMS chaves de condição
VerifyMac kms:VerifyMac	Política de chave	Sim	Chave do KMS	Condições para operações de chave do KMS: kms: CallerAccount kms: KeySpec kms: KeyUsage kms: KeyOrigin kms: MultiRegion kms: MultiRegionKeyType kms: ResourceAliases aws:ResourceTag/tag-key (chave de condição AWS global) kms: ViaService Condições para operações criptográficas: kms: MacAlgorithm kms: RequestAlias

Descrições das colunas

As colunas nesta tabela fornecem as seguintes informações:

- Ações e permissões listam cada operação da AWS KMS API e a permissão que permite a operação. Você especifica a operação no elemento `Action` de uma instrução de política.
- Tipo de política indica se a permissão pode ser usada em uma política de chaves ou política do IAM.

Política de chaves significa que você pode especificar a permissão na política de chaves. Quando a política de chaves contém a [instrução de política que permite políticas do IAM](#), você pode especificar as permissões em uma política do IAM.

Política do IAM significa que você pode especificar a permissão apenas em uma política do IAM.

- Uso entre contas mostra as operações que os usuários autorizados podem executar em recursos em uma Conta da AWS diferente.

Um valor de Yes (Sim) significa que as entidades principais podem executar a operação em recursos em uma Conta da AWS diferente.

Um valor de No (Não) significa que as entidades principais podem executar a operação somente em recursos da própria Conta da AWS.

Se você conceder a uma entidade principal em uma conta diferente uma permissão que não pode ser usada em um recurso entre contas, essa permissão não será efetiva. Por exemplo, se você der `TagResource` permissão a um principal em uma conta diferente [kms](#): para uma chave KMS em sua conta, as tentativas dele de marcar a chave KMS em sua conta falharão.

- Recursos lista os AWS KMS recursos aos quais as permissões se aplicam. AWS KMS suporta dois tipos de recursos: uma chave KMS e um alias. Em uma política de chaves, o valor do elemento `Resource` é sempre `*`, o que indica a chave do KMS à qual a política de chaves está anexada.

Use os valores a seguir para representar um AWS KMS recurso em uma política do IAM.

Chave KMS

Quando o recurso for uma chave do KMS, use seu [ARN de chave](#). Para obter ajuda, consulte [the section called “Encontrar o ID e o ARN da chave”](#).

`arn:AWS_partition_name:kms:AWS_Region:AWS_account_ID:key/key_ID`

Por exemplo:

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

Alias

Quando o recurso for um alias, use seu [ARN do alias](#). Para obter ajuda, consulte [the section called “Encontrar o nome e o ARN do alias”](#).

`arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:alias/alias_name`

Por exemplo:

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

* (asterisco)

Quando a permissão não se aplicar a um recurso específico (chave do KMS ou alias), use um asterisco (*).

Em uma política do IAM para uma AWS KMS permissão, um asterisco no Resource elemento indica todos os AWS KMS recursos (chaves e aliases do KMS). Você também pode usar um asterisco no Resource elemento quando a AWS KMS permissão não se aplica a nenhuma chave ou alias KMS em particular. Por exemplo, ao permitir ou negar a permissão `kms:CreateKey` ou `kms:ListKeys`, você deve definir o elemento Resource como `*`.

- AWS KMS as chaves de AWS KMS condição listam as chaves de condição que você pode usar para controlar o acesso à operação. Especifique as condições no elemento Condition de uma política. Para obter mais informações, consulte [AWS KMS chaves de condição](#). Essa coluna também inclui [chaves de condição AWS globais](#) que são suportadas por AWS KMS, mas não por todos os AWS serviços.

AWS KMS operações internas

AWS Key Management Service (AWS KMS) fornece chaves criptográficas e operações protegidas por [módulos de segurança de hardware \(HSM\) validados pelo FIPS 140-3 Security Level 3](#) e dimensionados para a nuvem. AWS KMS chaves e funcionalidades são usadas por vários serviços AWS em nuvem, e você pode usá-las para proteger os dados em seus aplicativos. Este guia técnico fornece detalhes sobre as operações criptográficas que são executadas AWS durante o uso AWS KMS.

AWS KMS os internos são necessários para escalar e proteger um serviço HSMs de gerenciamento de chaves distribuído globalmente.

Tópicos

- [Domínios e estado do domínio](#)
- [Segurança de comunicação interna](#)
- [Processo de replicação para chaves em multirregiões](#)
- [Proteção de durabilidade](#)

Domínios e estado do domínio

Uma coleção cooperativa de AWS KMS entidades internas confiáveis dentro de um Região da AWS é chamada de domínio. Um domínio inclui um conjunto de entidades confiáveis, um conjunto de regras e um conjunto de chaves secretas, chamadas chaves de domínio. As chaves de domínio são HSMs compartilhadas entre os membros do domínio. Um estado de domínio consiste nos campos a seguir.

Nome

Um nome de domínio para identificar este domínio.

Membros

Uma lista dos HSMs membros do domínio, incluindo a chave pública de assinatura e as chaves públicas do contrato.

Operadores

Uma lista de entidades, chaves públicas de assinatura e uma função (AWS KMS operadora ou host do serviço) que representa os operadores desse serviço.

Regras

Uma lista de regras de quórum para cada comando que deve ser satisfeita para executar um comando no HSM.

Chaves de domínio

Uma lista de chaves de domínio (chaves simétricas) atualmente em uso no domínio.

O estado completo do domínio está disponível apenas no HSM. O estado do domínio é sincronizado entre os membros do domínio HSM como um token de domínio exportado.

Chaves de domínio

Todos os HSMs itens em um domínio compartilham um conjunto de chaves de domínio, $\{DK_r\}$. Essas chaves são compartilhadas por meio de uma rotina de exportação de estado de domínio. O estado do domínio exportado pode ser importado para qualquer HSM membro do domínio.

O conjunto de chaves de domínio, $\{DK_r\}$, sempre inclui uma chave de domínio ativa e várias chaves de domínio desativadas. As chaves de domínio são trocadas diariamente para garantir que estejam AWS em conformidade com a [Recomendação para Gerenciamento de Chaves - Parte 1](#). Durante a alternância das chaves de domínio, todas as chaves KMS existentes criptografadas sob a chave de domínio de saída são criptografadas novamente sob a nova chave de domínio ativa. A chave de domínio ativa é usada para criptografar qualquer novo EKTs. As chaves de domínio expiradas só podem ser usadas para descriptografar previamente criptografadas EKTs por um número de dias equivalente ao número de chaves de domínio alteradas recentemente.

Tokens de domínio exportados

Há uma necessidade regular de sincronizar o estado entre os participantes do domínio. Isso é feito através da exportação do estado do domínio sempre que uma alteração é feita no domínio. O estado do domínio é exportado como um token de domínio exportado.

Nome

Um nome de domínio para identificar este domínio.

Membros

Uma lista dos HSMs membros do domínio, incluindo suas chaves públicas de assinatura e contrato.

Operadores

Uma lista de entidades, chaves de assinatura pública e uma função que representa os operadores deste serviço.

Regras

Uma lista de regras de quórum para cada comando que deve ser satisfeita para executar um comando em um membro do domínio HSM.

Chaves de domínio criptografadas

Chaves de domínio criptografadas por envelope. As chaves de domínio são criptografadas pelo membro de assinatura para cada um dos membros listados acima, envoltas em sua chave de contrato público.

Signature

Uma assinatura no estado do domínio produzida por um HSM, necessariamente um membro do domínio que exportou o estado do domínio.

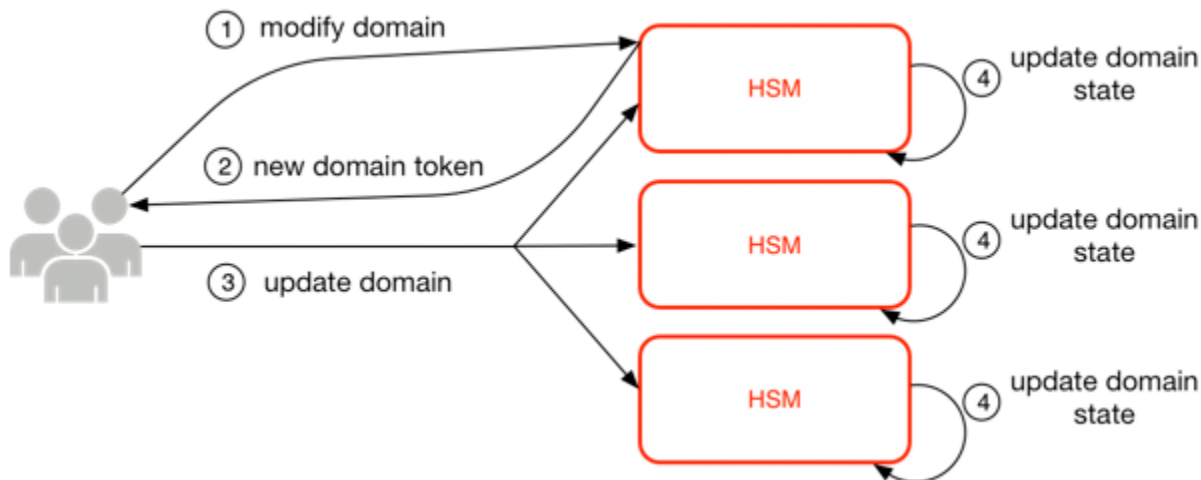
O token de domínio exportado forma a fonte fundamental de confiança para entidades que operam no domínio.

Gerenciar estados de domínio

O estado do domínio é gerenciado por meio de comandos autenticados por quórum. Essas alterações incluem modificar a lista de participantes confiáveis no domínio, modificar as regras de quórum para executar comandos HSM e alternar periodicamente as chaves de domínio. Esses comandos são autenticados um a um, e não por meio de operações de sessão autenticada, como mostrado na imagem a seguir.

Em seu estado inicializado e operacional, um HSM contém um conjunto de chaves de identidade assimétricas autogeradas, um par de chaves de assinatura e um par de chaves de estabelecimento de chave. Por meio de um processo manual, um AWS KMS operador pode estabelecer um domínio inicial a ser criado em um primeiro HSM em uma região. Este domínio inicial consiste em um estado de domínio completo, conforme definido anteriormente neste tópico. Ele é instalado por meio de um comando “join” para cada um dos membros do HSM definidos no domínio.

Depois que um HSM ingressou em um domínio inicial, ele fica vinculado às regras definidas nesse domínio. Essas regras regem os comandos que usam chaves criptográficas do cliente ou fazem alterações no estado do host ou do domínio. As operações de API de sessão autenticada que usam suas chaves criptográficas foram definidas anteriormente.



A imagem anterior mostra como um estado de domínio é modificado. O processo consiste em quatro etapas:

1. Um comando baseado em quórum é enviado para um HSM para modificar o domínio.
2. Um novo estado de domínio é gerado e exportado como um novo token de domínio exportado. O estado no HSM não é modificado, ou seja, a alteração não é promulgada no HSM.
3. Um segundo comando é enviado para cada um HSMs no token de domínio recém-exportado para atualizar o estado do domínio com o novo token de domínio.
4. O HSMs listado no novo token de domínio exportado pode autenticar o comando e o token de domínio. Eles também podem descompactar as chaves de domínio para atualizar o estado do domínio em tudo HSMs no domínio.

HSMs não se comunicam diretamente uns com os outros. Em vez disso, um quórum de operadores solicita uma alteração no estado do domínio que resulta em um novo token de domínio exportado. Um membro de host de serviço do domínio é usado para distribuir o novo estado de domínio para cada HSM no domínio.

A saída e a adesão a um domínio são feitas por meio das funções de gerenciamento do HSM. A modificação do estado do domínio é feita através das funções de gerenciamento de domínio.

Sair do domínio

Faz com que um HSM saia de um domínio, excluindo da memória todos os resquícios e chaves daquele domínio.

Ingressar no domínio

Faz com que um HSM ingresse em um novo domínio ou atualize seu estado de domínio atual para o novo estado de domínio. O domínio existente é usado como fonte do conjunto inicial de regras para autenticar esta mensagem.

Criar um domínio

Faz com que um novo domínio seja criado em um HSM. Retorna um primeiro token de domínio que pode ser distribuído ao membro HSMs do domínio.

Modificar operadores

Adiciona ou remove operadores da lista de operadores autorizados e suas funções no domínio.

Modificar membros

Adiciona ou remove um HSM da lista de autorizados HSMs no domínio.

Modificar regras

Modifica o conjunto de regras de quórum necessárias para executar comandos em um HSM.

Alternar chaves de domínio

Faz com que uma nova chave de domínio seja criada e marcada como a chave de domínio ativa. Isso move a chave ativa existente para uma chave desativada e remove a chave desativada mais antiga do estado do domínio.

Segurança de comunicação interna

Os comandos entre os hosts ou AWS KMS operadores de serviço e os HSMs são protegidos por meio de dois mecanismos descritos em [Sessões autenticadas](#): um método de solicitação com assinatura de quorum e uma sessão autenticada usando um protocolo de host de serviço HSM.

Os comandos assinados por quorum são projetados para que nenhum operador possa modificar as proteções críticas de segurança que eles fornecem. HSMs Os comandos executados nas sessões autenticadas ajudam a garantir que apenas operadores de serviço autorizados possam executar operações envolvendo chaves KMS. Todas as informações secretas vinculadas ao cliente são protegidas em toda a AWS infraestrutura.

Estabelecimento de chaves

Para proteger as comunicações internas, AWS KMS usa dois métodos diferentes de estabelecimento de chaves. O primeiro é definido como C(1, 2, ECC DH) na [Recomendação para Esquemas de Estabelecimento de Chave em Par usando Criptografia de Logaritmo Discreto](#). Este esquema tem um iniciador com uma chave de assinatura estática. O iniciador gera e assina uma chave efêmera curva elíptica Diffie-Hellman (ECDH) destinada a um destinatário com uma chave de acordo ECDH estática. Este método usa uma chave efêmera e duas chaves estáticas usando ECDH. Essa é a derivação do rótulo C(1, 2, ECC DH). Este método, às vezes, é chamado de ECDH de uma passagem.

O segundo método de estabelecimento de chave é [C\(2, 2, ECC, DH\)](#). Neste esquema, ambas as partes têm uma chave de assinatura estática e geram, assinam e trocam uma chave ECDH efêmera. Este método usa duas chaves estáticas e duas chaves efêmeras, cada uma usando ECDH. Essa é a derivação do rótulo C(2, 2, ECC, DH). Este método é, por vezes, chamado ECDH efêmero ou ECDHE. Todas as chaves ECDH são geradas na curva secp384r1 (NIST-P384).

Limite de segurança do HSM

O limite interno de segurança do AWS KMS é o HSM. O HSM tem uma interface proprietária e nenhuma outra interface física ativa em seu estado operacional. Um HSM operacional é implantado durante a inicialização com as chaves criptográficas necessárias para estabelecer sua função no domínio. Materiais criptográficos sensíveis do HSM são armazenados somente na memória volátil e apagados quando o HSM sai do estado operacional, incluindo desligamentos ou reinicializações intencionais ou não.

As operações da API do HSM são autenticadas por comandos individuais ou por uma sessão confidencial mutuamente autenticada estabelecida por um host de serviço.

Comandos assinados por quórum

Os comandos assinados por quorum são emitidos pelos operadores para HSMs. Esta seção descreve como os comandos baseados em quórum são criados, assinados e autenticados. Estas regras são bastante simples. Por exemplo, o comando Foo requer dois membros da função Bar para ser autenticado. Há três etapas na criação e verificação de um comando baseado em quórum. A primeira etapa é a criação inicial do comando; a segunda é o envio a operadores adicionais para assinar e a terceira é a verificação e execução.

Com a finalidade de introduzir os conceitos, suponha que existe um conjunto autêntico de chaves públicas e funções do operador $\{QOS_s\}$ e um conjunto de regras de quórum $QR = \{Comando_i,$

$Regra_{\{i, t\}}$ no qual cada Regra é um conjunto de funções e número mínimo N $\{Função_t, N_t\}$. Para que um comando satisfaça a regra de quórum, o conjunto de dados de comando deve ser assinado por um conjunto de operadores listados em $\{QOS_s\}$ de forma que atendam a uma das regras listadas para esse comando. Como mencionado anteriormente, o conjunto de regras de quórum e operadores são armazenados no estado de domínio e no token de domínio exportado.

Na prática, um signatário inicial assina o comando $Sig_1 = \text{Sign}(dO_{p1}, \text{Comando})$. Um segundo operador também assina o comando $Sig_2 = \text{Sign}(dO_{p2}, \text{Comando})$. A mensagem duplamente assinada é enviada para um HSM para execução. O HSM executa o seguinte:

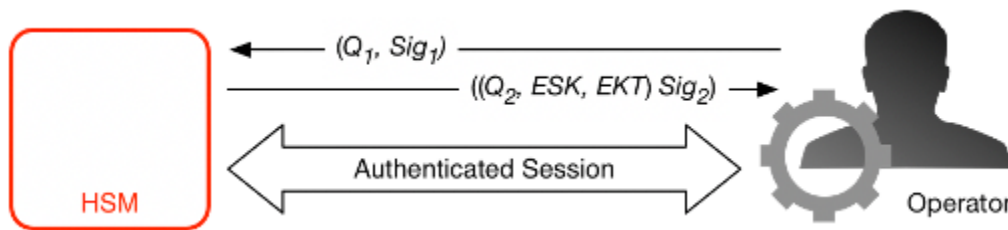
1. Para cada assinatura, ele extrai a chave pública do signatário do estado do domínio e verifica a assinatura no comando.
2. Ele verifica se o conjunto de signatários satisfaz a uma regra para o comando.

Sessões autenticadas

Suas principais operações são executadas entre os AWS KMS anfitriões voltados para o exterior e o. HSMs Esses comandos dizem respeito à criação e uso de chaves criptográficas e geração de números aleatórios seguros. Os comandos são executados em um canal autenticado por sessão entre os hosts do serviço e o. HSMs Além da necessidade de autenticação, essas sessões exigem confidencialidade. Os comandos executados nessas sessões incluem o retorno de chaves de dados de texto não criptografado e mensagens descriptografadas destinadas a você. Para garantir que essas sessões não possam ser subvertidas por meio de man-in-the-middle ataques, as sessões são autenticadas.

Este protocolo executa um contrato de chave ECDHE mutuamente autenticado entre o HSM e o host do serviço. A troca é iniciada pelo host do serviço e concluída pelo HSM. O HSM também retorna uma chave de sessão (SK) criptografada pela chave negociada e um token de chave exportado que contém a chave de sessão. O token de chave exportado contém um período de validade, após o qual o host de serviço deve renegociar uma chave de sessão.

Um host de serviços é membro do domínio e tem um par de chaves de assinatura de identidade (DHO_{si} , $QHOS_i$) e uma cópia autêntica das “chaves públicas de identidade HSMs”. Ele usa seu conjunto de chaves de assinatura de identidade para negociar com segurança uma chave de sessão que pode ser usada entre o host de serviço e qualquer HSM no domínio. Os tokens de chave exportados têm um período de validade associado após o qual uma nova chave deve ser negociada.



O processo começa com o reconhecimento do host de serviço que requer uma chave de sessão para enviar e receber fluxos de comunicação confidenciais entre ele e um membro do HSM do domínio.

1. Um host de serviço gera um par de chaves efêmero ECDH (d_1 , Q_1) e o assina com sua chave de identidade $Sig_1 = \text{Sign}(\text{dOS}, Q_1)$.
2. O HSM verifica a assinatura na chave pública recebida usando seu token de domínio atual e cria um par de chaves efêmero ECDH (d_2 , Q_2). Em seguida, ele conclui a [Recomendação ECDH-key-exchange para esquemas de estabelecimento de chaves em pares usando criptografia de logaritmo discreto para formar uma chave AES-GCM](#) negociada de 256 bits. O HSM gera uma nova chave de sessão AES-GCM de 256 bits. Ele criptografa a chave de sessão com a chave negociada para formar a chave de sessão criptografada (ESK). Ele também criptografa a chave de sessão sob a chave de domínio como um token de chave EKT exportado. Finalmente, ele assina um valor de retorno com seu par de chaves de identidade $Sig_2 = \text{Sign}(\text{dHsK}, (Q_2, \text{ESK}, \text{EKT}))$.
3. O host do serviço verifica a assinatura nas chaves recebidas usando seu token de domínio atual. O host do serviço conclui a troca de chaves ECDH de acordo com a [Recomendação para Esquemas de Estabelecimento de Chave em Par usando Criptografia de Logaritmo Discreto](#). Em seguida, ele descripta o ESK para obter a chave de sessão SK.

Durante o período de validade no EKT, o host de serviço pode usar a chave de sessão negociada SK para enviar comandos criptografados com envelope para o HSM. Cada service-host-initiated comando sobre essa sessão autenticada inclui o EKT. O HSM responde usando a mesma chave de sessão negociada SK.

Processo de replicação para chaves em multirregiões

AWS KMS usa um mecanismo de replicação entre regiões para copiar o material chave em uma chave KMS de um HSM em um Região da AWS para um HSM em outro. Região da AWS Para que esse mecanismo funcione, a chave KMS que está sendo replicada deve ser uma chave multirregional com origem. AWS_KMS Para chaves multirregionais com material de chave importado

(EXTERNALorigem), você é responsável por replicar o material de chave importando o mesmo material de chave em cada réplica individualmente.

Ao replicar uma chave KMS de uma região para outra, as HSMs regiões não podem se comunicar diretamente, porque estão em redes isoladas. Em vez disso, as mensagens trocadas durante a replicação entre regiões são entregues por um serviço proxy. Durante a replicação entre regiões, cada mensagem gerada por um AWS KMS HSM é assinada criptograficamente usando uma chave de assinatura de replicação. As chaves de assinatura de replicação (RSKs) são chaves ECDSA na curva NIST P-384. Cada região possui pelo menos um RSK, e o componente público de cada RSK é compartilhado com todas as outras regiões na mesma AWS partição.

O processo de replicação entre regiões para copiar material de chaves da Região A para a Região B funciona da seguinte forma:

1. O HSM na Região B gera uma chave ECDH efêmera na curva NIST P-384, Contrato de replicação da chave B (RAKB). O componente público do RAKB é enviado para um HSM na Região A pelo serviço de proxy.
2. O HSM na Região A recebe o componente público do RAKB e, em seguida, gera outra chave ECDH efêmera na curva NIST P-384, Contrato de replicação da Chave A (RAKA). O HSM executa o esquema de estabelecimento de chave ECDH no RAKA e no componente público do RAKB, e deriva uma chave simétrica da saída, o Empacotamento de replicação de chave (RWK). O RWK é usado para criptografar o material de chaves da chave do KMS de várias regiões que está sendo replicada.
3. O componente público do RAKA e o material de chaves criptografado com o RWK são enviados para o HSM na Região B por meio do serviço de proxy.
4. O HSM na Região B recebe o componente público do RAKA e o material de chaves criptografado usando o RWK. O HSM deriva pela RWK executando o regime de estabelecimento de chave ECDH no RAKB e o componente público do RAKA.
5. O HSM na Região B usa o RWK para descriptografar o material de chaves da Região A.

Proteção de durabilidade

A durabilidade adicional do serviço para chaves geradas pelo serviço é fornecida pelo uso de armazenamento offline HSMs e múltiplo não volátil de tokens de domínio exportados e armazenamento redundante de chaves KMS criptografadas. Os off-line HSMs são membros dos domínios existentes. Com exceção de não estar on-line e participar das operações regulares

de domínio, os off-line HSMs aparecem de forma idêntica no estado do domínio aos membros existentes do HSM.

O design de durabilidade visa proteger todas as chaves KMS em uma região que AWS sofram uma perda em grande escala das chaves on-line HSMs ou do conjunto de chaves KMS armazenadas em nosso sistema de armazenamento principal. AWS KMS keys com material de chave importado não estão incluídos nas proteções de durabilidade oferecidas por outras chaves KMS. No caso de uma falha regional AWS KMS, o material de chave importado pode precisar ser reimportado para uma chave KMS.

O offline e HSMs as credenciais para acessá-los são armazenados em cofres dentro de salas seguras monitoradas em várias localizações geográficas independentes. Cada cofre requer pelo menos um agente de AWS segurança e um AWS KMS operador, de duas equipes independentes AWS, para obter esses materiais. O uso desses materiais é regido por uma política interna que exige a presença de um quórum de AWS KMS operadores.

Histórico do documento

Este tópico descreve atualizações importantes no Guia do desenvolvedor do AWS Key Management Service .

Tópicos

- [Atualizações recentes](#)
- [Atualizações anteriores](#)

Atualizações recentes

A tabela a seguir descreve alterações significativas nesta documentação desde janeiro de 2018. Além das principais alterações listadas aqui, também atualizamos a documentação com frequência para melhorar as descrições e os exemplos e abordar os comentários que você nos envia. Para ser notificado sobre alterações significativas, inscreva-se no feed RSS.

Talvez seja necessário rolar horizontalmente ou verticalmente para ver todos os dados nessa tabela.

Alteração	Descrição	Data
Atualização de recurso	AWS KMS oferece suporte à rotação sob demanda de chaves multirregionais com criptografia simétrica com material de chave importado.	26 de novembro de 2025
Atualização de recurso	AWS KMS suporta um novo <code>keyExchange</code> campo na <code>tlsDetails</code> seção de CloudTrail eventos.	24 de novembro de 2025
Atualização de recurso	AWS KMS suporta o novo KeySpec EDWARDS25519 ECC_NIST_.	7 de novembro de 2025
Compatibilidade com pilha dupla	AWS KMS suporta pilha dupla.	18 de junho de 2025

Atualização de recurso	Inclui compatibilidade com assinaturas criptográficas pós-quânticas ML-DSA (Module-Lattice Digital Signature Algorithm).	13 de junho de 2025
Rodízio de chaves importadas	É possível realizar o rodízio sob demanda de chaves do KMS de criptografia simétrica com material de chave importado (origem EXTERNAL).	5 de junho de 2025
Atualização de recurso	Inclusão de compatibilidade com chaves do KMS multirregionais nas regiões China.	21 de novembro de 2024
AWS atualização de política gerenciada	A função AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy vinculada ao serviço foi atualizada adicionando uma ID de instrução (Sid) à política gerenciada com a versão v2 da política.	21 de novembro de 2024
Alteração de cota	Aumentou a taxa de solicitação padrão para DeleteImportedKeyMaterial solicitações ImportKeyMaterial e solicitações.	23 de julho de 2024

Alteração de cota	Aumentou a taxa de solicitação de operações criptográficas padrão para chaves KMS de criptografia simétrica, chaves RSA KMS e chaves ECC e KMS. SM2	8 de julho de 2024
Novo atributo	Foi adicionado um novo KeyUsage tipo KEY_AGREEMENT para chaves KMS de curva elíptica (ECC) e SM2 (somente para regiões da China) recomendadas pelo NIST e suporte adicional para derivar segredos compartilhados.	13 de junho de 2024
Atualizações na alternância de chaves	Adição de compatibilidade com períodos de personalizados alternância para alternâncias automáticas de chaves, alternâncias de chaves sob demanda e visibilidade das alternância de material de chave.	12 de abril de 2024
Atualizações da política gerenciada	Novas permissões adicionadas permitem AWS KMS monitorar as alterações na VPC que contém seu AWS CloudHSM cluster para que AWS KMS possa fornecer mensagens de erro claras em caso de falhas. <code>AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy</code>	10 de novembro de 2023

Atualização de recurso	Foi adicionado suporte para o parâmetro da API DryRun.	5 de julho de 2023
Atualização de recurso	Foi adicionado suporte para importar material de chaves para todos os tipos de AWS KMS chaves, exceto lojas de chaves personalizadas.	5 de junho de 2023
Atualização de recurso	Atualizações AWS KMS APIs para Nitro Enclaves	10 de março de 2023
Atualização de recurso	O algoritmo de RSAES_PKCS1_V1_5 empacotamento está obsoleto. AWS KMS encerrará todo o suporte até 1º de outubro de 2023, de acordo com as diretrizes de gerenciamento de chaves criptográficas do Instituto Nacional de Padrões e Tecnologia (NIST). Recomendamos que você comece a usar um algoritmo de empacotamento diferente imediatamente.	28 de fevereiro de 2023
Atualização de recurso	Foi adicionado suporte para armazenamentos externos de chaves, um recurso que permite proteger seus AWS recursos usando chaves criptográficas externas. AWS	29 de novembro de 2022

Alteração de cota	Aumentamos a cota de AWS KMS keys recursos para 100.000 chaves KMS em cada conta e região.	8 de julho de 2022
Atualização de recursos	Foi adicionado suporte para chaves HMAC KMS em mais Regiões da AWS	8 de julho de 2022
Novo tópico	O AWS Key Management Service tópico Resiliência foi adicionado ao capítulo Segurança do Guia do AWS KMS Desenvolvedor.	14 de junho de 2022
Novo atributo	Foi adicionado suporte para AWS KMS chaves e operações de API que geram e verificam códigos HMAC.	19 de abril de 2022
Alteração na documentação	Substituição da condição chave mestra do cliente (CMK) por AWS KMS key e chave do KMS.	30 de agosto de 2021
Novo recurso	Suporte adicionado para chaves de várias regiões , um conjunto de chaves do KMS interoperáveis em regiões diferentes que têm o mesmo ID de chave e material de chave. É possível usar chaves de várias regiões para criptografar dados em uma região e descriptografar dados em uma região diferente.	8 de junho de 2021

Novo recurso	Suporte adicionado para controle de acesso baseado em atributos (ABAC). Você pode usar tags e aliases para controlar o acesso ao seu AWS KMS keys.	17 de dezembro de 2020
Novo recurso	Adicionado suporte para políticas de endpoint da VPC.	9 de julho de 2020
Novo conteúdo	Explica as propriedades de segurança do AWS KMS.	18 de junho de 2020
Novo recurso	Foi adicionado suporte para chaves de dados assimétricas AWS KMS keys e assimétricas.	25 de novembro de 2019
Recurso atualizado	Você pode ver a política de chaves do Chaves gerenciadas pela AWS no AWS KMS console. Esse recurso costumava ser limitado a chaves gerenciadas pelo cliente.	15 de novembro de 2019
Novo recurso	Explica como usar algoritmos de troca de chave pós-quântica híbrida no TLS para suas chamadas para o AWS KMS.	4 de novembro de 2019
Alteração de cota	Aumentamos as cotas de recursos para alguns APIs que gerenciam chaves KMS.	18 de setembro de 2019

Alteração de cota	Alteradas as cotas de recursos para chaves do KMS, aliases e concessões por chave do KMS.	27 de março de 2019
Alteração de cota	Alterada a cota de solicitações compartilhadas por segundo para operações de criptografia que usam AWS KMS keys em um armazenamento de chaves personalizado.	7 de março de 2019
Novo recurso	Explica como criar e gerenciar armazenamentos de chaves AWS KMS personalizadas . Cada armazenamento de chaves é apoiado por um AWS CloudHSM cluster que você possui e controla.	26 de novembro de 2018
Novo console	Explica como usar o novo AWS KMS console, que é independente do console do IAM. O console original e as instruções para usá-lo permanecerão disponíveis por um breve período para que você tenha tempo para se familiarizar com o novo console.	7 de novembro de 2018
Alteração de cota	Alterou a cota de solicitações compartilhadas para uso de AWS KMS keys.	21 de agosto de 2018

Novo conteúdo	Explica como AWS Secrets Manager usa AWS KMS chaves para criptografar o valor secreto em um segredo.	13 de julho de 2018
Novo conteúdo	Explica como o DynamoDB AWS KMS AWS KMS keys usa para oferecer suporte à opção de criptografia do lado do servidor.	23 de maio de 2018
Novo recurso	Explica como usar um endpoint privado em sua VPC para se conectar diretamente AWS KMS, em vez de se conectar pela Internet.	22 de janeiro de 2018

Atualizações anteriores

A tabela a seguir descreve as mudanças importantes no Guia do AWS Key Management Service Desenvolvedor antes de 2018.

Talvez seja necessário rolar horizontalmente ou verticalmente para ver todos os dados nessa tabela.

Alteração	Descrição	Data
Novo conteúdo	Documentação adicionada sobre Etiquetas em AWS KMS .	15 de fevereiro de 2017
Novo conteúdo	Documentação adicionada sobre Monitorar AWS KMS keys e Monitore as chaves KMS com a Amazon CloudWatch .	31 de agosto de 2016

Alteração	Descrição	Data
Novo conteúdo	Documentação adicionada sobre Material de chave importado .	11 de agosto de 2016
Novo conteúdo	Adicionada a seguinte documentação: Políticas do IAM , Referência de permissões e Chaves de condição .	5 de julho de 2016
Atualizar	Partes da documentação atualizadas no capítulo Acesso e permissões de chave do KMS .	5 de julho de 2016
Atualizar	Atualizada a página Cotas para refletir as novas cotas padrão.	31 de maio de 2016
Atualizar	Atualização da página Cotas para refletir novas cotas padrão e atualização da documentação sobre o token de concessão para aumentar a clareza e a precisão.	11 de abril de 2016
Novo conteúdo	Documentação adicionada sobre Permitir que diversas entidades principais do IAM acessem uma chave do KMS e Usar a condição de endereço IP .	17 de fevereiro de 2016

Alteração	Descrição	Data
Atualizar	Atualizadas as páginas Políticas-chave em AWS KMS e Alterar uma política de chave para aumentar a clareza e a precisão.	17 de fevereiro de 2016
Atualizar	Atualização da página de tópicos sobre gerenciamento de chaves do KMS para esclarecimento.	5 de janeiro de 2016
Novo conteúdo	Documentação adicionada sobre CloudTrail.	18 de novembro de 2015
Novo conteúdo	Adicionadas instruções para Alterar uma política de chave .	18 de novembro de 2015
Atualizar	Atualizou a documentação sobre como o Amazon Relational Database Service AWS KMS usa.	18 de novembro de 2015
Novo conteúdo	Foi adicionada documentação sobre a Amazon WorkSpaces.	6 de novembro de 2015
Atualizar	Atualizada a página Políticas-chave em AWS KMS para aumentar a clareza.	22 de outubro de 2015
Novo conteúdo	Documentação adicionada sobre Excluir um AWS KMS key , inclusive a documentação de suporte sobre Criar um alarme e Determinar a utilização anterior de uma chave do KMS .	15 de outubro de 2015

Alteração	Descrição	Data
Novo conteúdo	Documentação adicionada sobre Determinando o acesso ao AWS KMS keys .	15 de outubro de 2015
Novo conteúdo	Documentação adicionada sobre Principais estados das AWS KMS chaves .	15 de outubro de 2015
Novo conteúdo	Adição de documentação sobre o Amazon Simple Email Service.	1º de outubro de 2015
Atualizar	Atualização da página Cotas para explicar as novas cotas de solicitações.	31 de agosto de 2015
Novo conteúdo	Foram adicionadas informações sobre as cobranças de uso AWS KMS. Consulte Definição de preço do AWS KMS .	14 de agosto de 2015
Novo conteúdo	Foram adicionadas cotas de solicitação ao AWS KMS Cotas .	11 de junho de 2015
Novo conteúdo	Adicionado um novo exemplo de código Java que demonstra o uso da operação UpdateAlias .	1º de junho de 2015
Atualizar	A tabela de regiões do AWS Key Management Service foi movida para a Referência geral da AWS.	29 de maio de 2015

Alteração	Descrição	Data
Novo conteúdo	Documentação adicionada sobre Como o Amazon EMR usa AWS KMS .	28 de janeiro de 2015
Novo conteúdo	Foi adicionada documentação sobre a Amazon WorkMail.	28 de janeiro de 2015
Novo conteúdo	Foi adicionada documentação sobre como o Amazon Relational Database Service AWS KMS usa.	6 de janeiro de 2015
Novo conteúdo	Adição de documentação sobre o Amazon Elastic Transcoder.	24 de novembro de 2014
Novo guia	Introdução do Guia do desenvolvedor do AWS Key Management Service .	12 de novembro de 2014

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.