



Guía para desarrolladores

AWS Key Management Service



AWS Key Management Service: Guía para desarrolladores

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

AWS Key Management Service	1
¿Por qué AWS KMS usarlas?	1
AWS KMS in Regiones de AWS	2
AWS KMS precios	2
AWS KMS acuerdo de nivel de servicio	2
Acceder AWS Key Management Service	3
Consola de administración de AWS	3
Permisos necesarios para usar la AWS KMS consola	3
AWS Command Line Interface	4
AWS KMS REST API	4
AWS SDKs	4
Trabajando con AWS SDKs	5
AWS Encryption SDK	6
AWS KMS coherencia eventual	6
TLS híbrido postcuántico	7
Acerca del cifrado TLS postcuántico	8
Modo de uso	9
Configuración del cifrado TLS postcuántico híbrido	10
Más información	12
Conéctese a AWS KMS través de un punto final de VPC	12
Cree un punto final de VPC para AWS KMS	13
Conexión a un punto de conexión de VPC	14
Utilice los puntos finales de VPC para controlar el acceso a los recursos AWS KMS	16
Registro de AWS KMS solicitudes que utilizan un punto final de VPC	19
Puntos de conexión de doble pila	21
Las funciones no están disponibles en IPv6	21
Conceptos	22
Introducción	22
Objetivos de diseño	23
AWS KMS keys	24
Claves administradas por el cliente	28
Claves administradas por AWS	28
Claves propiedad de AWS	29
AWS KMS key jerarquía	30

Identificadores clave () KeyId	32
Claves asimétricas	35
Claves HMAC	37
Claves ML-DSA	40
Multi-Region llaves	40
Material de claves importado	52
Claves KMS en un almacén de claves de CloudHSM	59
Claves KMS en un almacén de claves externo	62
AWS KMS elementos básicos de criptografía	65
Entropía y generación de números aleatorios	65
Operaciones de clave simétrica (solo cifrado)	65
Operaciones de clave asimétrica (cifrado, firma digital y verificación de firmas)	66
Funciones de derivación de claves	67
AWS KMS uso interno de firmas digitales	67
Cifrado doble	67
Operaciones criptográficas	69
Algoritmos compatibles	71
Algoritmos criptográficos	71
Permisos y acceso a claves KMS	75
Políticas de clave de KMS	75
Concesiones de claves de KMS	76
Políticas de claves	77
Creación de una política de claves	77
Política de claves predeterminada	85
Visualización de políticas de claves	101
Cambio de una política de claves	104
Permisos para AWS los servicios	107
Políticas de IAM	108
Conceder permiso a varias entidades principales de IAM para acceder a una clave KMS	110
Prácticas recomendadas para las políticas de IAM	111
Especificación de claves KMS en declaraciones de políticas de IAM	114
Ejemplos	117
Políticas de control de recursos	124
Concesiones	126
Conceptos de concesión	128
Prácticas recomendadas	134

Control del acceso a las concesiones	136
Creación de concesiones	137
Visualización de concesiones	149
Uso de un token de concesión	151
Retiro y revocación de concesiones	152
Claves de condición	154
AWS claves de condición globales	154
AWS KMS claves de condición	158
AWS KMS claves de condición para plataformas certificadas	237
Aplicar permisos de privilegios mínimos	243
Implementación de permisos de privilegio mínimo	244
Control de acceso basado en atributos (ABAC)	248
Claves de condición ABAC para AWS KMS	249
¿Etiquetas o alias?	252
Solución de problemas de ABAC para AWS KMS	253
Control de acceso basado en roles (RBAC)	258
Acceso entre cuentas	260
Paso 1: Agregar una declaración de política de claves en la cuenta local	262
Paso 2: Agregar políticas de IAM a la cuenta externa	266
Permitir el uso de claves KMS externas con Servicios de AWS	268
Uso de claves KMS en otras cuentas	268
Control del acceso a claves de varias regiones	269
Conceptos básicos de autorización para claves de varias regiones	270
Autorización de administradores y usuarios clave de varias regiones	271
Determinar el acceso	276
Examinar la política de claves	276
Examen de las políticas de IAM	280
Examinar concesiones	282
Contexto de cifrado	284
Reglas de contexto de cifrado	285
Contexto de cifrado de las políticas	286
Contexto de cifrado de las concesiones	286
Registrar el contexto de cifrado	287
Almacenar el contexto de cifrado	287
Prueba de los permisos	288
¿Qué es DryRun?	288

Especificar DryRun con la API	289
Solución de problemas de AWS KMS permisos	290
Ejemplo 1: Se deniega al usuario el acceso a una clave KMS de su Cuenta de AWS	292
Ejemplo 2: El usuario asume el rol con permiso para usar una clave de KMS en otro Cuenta de AWS	294
Glosario	298
Autenticación	298
Autorización	298
Autenticación con identidades	298
Administración del acceso con políticas	300
AWS KMS recursos	302
Crear una clave de KMS.	303
Permisos para crear claves KMS	305
Elección del tipo de clave KMS que se va a crear	306
Creación de claves de KMS de cifrado simétricas	308
Creación de una clave de KMS asimétrica	314
Creación de una clave KMS HMAC	320
Creación de claves primarias de varias regiones	325
Creación de claves de réplica de varias regiones	331
Paso 1: Elegir Regiones de réplica	331
Paso 2: Crear claves de réplica	332
Creación de una clave KMS con material de claves importado	338
Permisos para importar material de clave	339
Requisitos para el material de claves importado	341
Paso 1: Crea un material AWS KMS key sin clave	344
Paso 2: descargar la clave pública de encapsulamiento y el token de importación	347
Paso 3: Cifrar el material de claves	357
Paso 4: Importar el material de claves	367
Crear una clave KMS en un almacén de AWS CloudHSM claves	375
Creación de una nueva clave KMS en su almacén de claves de CloudHSM	376
Creación de una clave KMS en un almacén de claves externo	382
Requisitos para una clave de KMS en un almacén de claves externo	384
Creación de una clave KMS en un almacén de claves externo	385
Identificación y visualización de claves	394
Encontrar el ID de clave y el ARN de clave	394
Acceso y enumeración de detalles de claves KMS	396

Identificación de diferentes tipos de claves	406
Identificación de claves KMS asimétricas	407
Identificación de claves KMS HMAC	408
Identificación de claves KMS de varias regiones	408
Identificación de claves KMS con material de claves importado	409
Identifique las claves de KMS en AWS CloudHSM almacenes de claves	410
Identificación de claves KMS en almacenes de claves externos	411
Personalización de la vista de la consola	412
Ordenar y filtrar las claves KMS	413
Personalización de las tablas de claves KMS	416
Encuentre las claves y el material clave de KMS en un almacén de AWS CloudHSM claves	418
Busque las claves KMS en un almacén de AWS CloudHSM claves	419
Busque todas las claves de un almacén de AWS CloudHSM claves	420
Busque la clave KMS de una AWS CloudHSM clave	422
Busque la AWS CloudHSM clave de una clave KMS	427
Habilitar y deshabilitar claves	431
Rotación de claves	434
¿Por qué rotar las claves de KMS?	436
Cómo funciona la rotación de claves	437
Habilitación de la rotación automática de claves	442
Deshabilitación de la rotación automática de claves	445
Realización de la rotación de claves bajo demanda	447
Inicio de la rotación de claves bajo demanda (consola)	448
Iniciar la rotación de claves bajo demanda (AWS KMS API)	449
Enumeración de las rotaciones y los materiales clave	450
Enumere las rotaciones y los materiales de claves (consola)	450
Enumere las rotaciones y los materiales clave (AWS KMS API)	452
Rotación manual de claves	453
Cambio de la clave principal en un conjunto de claves de varias regiones	455
Actualización de la región principal	457
Eliminación de claves	460
Acerca del período de espera	461
Consideraciones especiales	462
Control del acceso a la eliminación de claves	465
Permitir a los administradores de claves programar y cancelar la eliminación de claves	465
Programar la eliminación de claves	468

Cancelación de la eliminación de claves	470
Crear una alarma	471
Eliminación del material de claves importado	473
Generación de claves de datos	477
Crear una clave de datos	477
Cómo funcionan las operaciones criptográficas con claves de datos	478
Cifrar los datos con una clave de datos	478
Descifrar los datos con una clave de datos	479
Cómo afectan las claves de KMS obsoletas a las claves de datos	480
Generación de pares de claves de datos	482
Crear un par de clave de datos	482
Funcionamiento de las operaciones criptográficas con pares de claves de datos	483
Cifrar los datos con un par de claves de datos	484
Descifrar los datos con un par de claves de datos	484
Firmar los mensajes con un par de claves de datos	485
Verificar una firma con un par de claves de datos	486
Obtenga un secreto compartido con pares de claves de datos	487
Realice operaciones sin conexión con claves públicas	488
Consideraciones especiales para descargar claves públicas	489
Descarga de la clave pública	490
Ejemplo de operaciones sin conexión	492
Obtención de secretos compartidos sin conexión	492
Verificación fuera de línea con pares de claves ML-DSA	494
Verificación fuera de línea con pares de SM2 claves (solo en las regiones de China)	496
Monitorización de claves	502
Herramientas de supervisión	503
Herramientas automatizadas	503
Herramientas manuales	504
Iniciar sesión con AWS CloudTrail	505
Resultado AWS KMS entradas de registro en CloudTrail	506
Excluyendo AWS KMS eventos de un sendero	507
Ejemplos de AWS KMS entradas de registro	508
Supervise las claves con CloudWatch	597
AWS KMS métricas y dimensiones	597
Cree una CloudWatch alarma de caducidad del material clave importado	607
Cree CloudWatch alarmas para almacenes de claves externos	608

Determinación del uso anterior de una clave KMS	613
Examen de los permisos de clave KMS para determinar el ámbito de uso potencial	614
Examine la última operación criptográfica realizada con una clave KMS	614
Examine AWS CloudTrail los registros para comprobar el uso anterior	617
Monitorea las teclas con Amazon EventBridge	620
Rotación de CMK de KMS	621
Vencimiento del material de claves importado de KMS	621
Eliminación de CMK de KMS	622
Alias	624
Funcionamiento de los alias	625
Control del acceso a alias	628
km: CreateAlias	628
kms: ListAliases	629
km: UpdateAlias	630
kms: DeleteAlias	632
Limitar los permisos de alias	633
Crear alias	634
Buscar el nombre del alias y el ARN de alias	636
Actualización de alias	641
Eliminación de un alias	642
Uso de alias para controlar el acceso a las claves KMS	644
km: RequestAlias	645
kms: ResourceAliases	646
Obtención de información sobre cómo utilizar alias en las aplicaciones	648
Busque alias en AWS CloudTrail registros	650
Tags	652
Control del acceso a las etiquetas	653
Permisos de etiquetas en políticas	654
Limitar los permisos de etiqueta	656
Agregar etiquetas.	658
Agregar etiquetas al crear una clave KMS	658
Agregar etiquetas a claves KMS existentes	660
Editar etiquetas	661
Eliminación de etiquetas	662
Ver etiquetas	663
Uso de etiquetas para controlar el acceso a las claves KMS	665

Almacenes de claves	670
AWS KMS almacén de claves estándar	670
AWS KMS almacén de llaves estándar con material clave importado	670
AWS KMS almacenes de claves personalizados	672
AWS CloudHSM almacén de claves	673
Almacén de claves externo	673
AWS CloudHSM tiendas clave	673
AWS CloudHSM conceptos de almacén clave	677
Controle el acceso a su almacén de AWS CloudHSM llaves	680
Crear un almacén de AWS CloudHSM claves	682
Ver un almacén de AWS CloudHSM claves	689
Editar la configuración AWS CloudHSM del almacén de claves	692
Conecta un almacén de AWS CloudHSM claves	696
Desconectar un almacén de AWS CloudHSM claves	700
Eliminar un almacén de AWS CloudHSM claves	703
Resolver problemas de un almacén de claves personalizado	706
Almacenes de claves externos	723
Conceptos del almacén de claves externo	728
Cómo funcionan los almacenes de claves externos	737
Control de acceso al almacén de claves externo	739
Elección de una opción de conectividad proxy	745
Creación de un almacén de claves externo	759
Edición de las propiedades del almacén de claves externo	774
Visualización de almacenes de claves externos	781
Monitoreo de almacenes de claves externos	787
Conexión y desconexión de almacenes de claves externos	800
Eliminación de un almacén de claves externo	811
Solución de problemas de almacenes de claves externos	813
Seguridad	840
Protección de datos	841
Rotación del material de claves	841
Cifrado de datos	843
Privacidad entre redes	844
Identity and Access Management	845
AWS políticas gestionadas	846
Roles vinculados a servicios	849

Registro y supervisión	855
Validación de conformidad	857
Documentos de conformidad y seguridad	857
Más información	858
Resiliencia	858
Aislamiento regional	859
Diseño de varios inquilinos	859
Mejores prácticas de resiliencia en AWS KMS	860
Seguridad de la infraestructura	860
Aislamiento de Hosts físicos	862
Cuotas	863
Cuotas de recursos	863
AWS KMS keys: 100 000	864
Alias por clave KMS: 50	864
Concesiones por clave KMS: 50 000	865
Cuota de recursos de almacenes de claves personalizados: 10	865
Rotación bajo demanda: 25	866
Cuotas de solicitudes	866
Solicita cuotas para cada operación AWS KMS de la API	867
Aplicar cuotas de solicitudes	875
Cuotas compartidas para operaciones criptográficas	875
Solicitudes de la API realizadas en su nombre	877
Solicitudes entre cuentas	877
Cuotas de solicitudes del almacén de claves personalizado	877
Limitación controlada de solicitudes	879
Ejemplos de código	882
Conceptos básicos	883
Hola AWS KMS	884
Conceptos básicos	888
Acciones	962
Escenarios	1135
Trabajo con el cifrado de tablas	1136
Soporte de atestación criptográfica en AWS KMS	1139
Cómo realizar llamadas certificadas a AWS KMS	1141
Monitoreo de solicitudes certificadas	1141
Supervisión de las solicitudes para enclaves de Nitro	1142

Monitoreo de las solicitudes para NitroTPM	1147
Servicios de cifrado AWS	1153
Amazon Elastic Block Store (Amazon EBS)	1154
Cifrado de Amazon EBS	1154
Uso de claves KMS y claves de datos	1155
Contexto de cifrado de Amazon EBS	1155
Detección de errores de Amazon EBS	1156
Utilización AWS CloudFormation para crear volúmenes cifrados de Amazon EBS	1157
Amazon EMR	1157
Cifrar datos en el sistema de archivos EMR (EMRFS)	1158
Cifrar datos en los volúmenes de almacenamiento de los nodos de clúster	1161
Contexto de cifrado	1162
Amazon Redshift	1163
Cifrado de Amazon Redshift	1163
Contexto de cifrado	1164
Referencia	1165
Referencia de los estados de claves	1166
Estados clave y tipos de claves KMS	1166
Tabla estado de claves	1167
Referencia de tipos de claves	1175
Tabla de tipos de claves	1176
Tabla de características especiales	1184
Referencia de especificaciones de clave	1193
Especificación de clave SYMMETRIC_DEFAULT	1195
Especificaciones de clave de RSA	1196
Especificaciones de clave de curva elíptica	1200
Especificaciones de la clave para las claves KMS HMAC	1203
Especificaciones clave del ML-DSA	1204
SM2 especificación clave (solo regiones de China)	1204
Referencia de permisos	1205
Descripciones de las columnas	1256
AWS KMS operaciones internas	1258
Dominios y estado de dominio	1259
Seguridad de la comunicación interna	1263
Proceso de replicación de claves multirregión	1266
Protección de durabilidad	1267

Historial de documentos 1269

Actualizaciones recientes 1269

Actualizaciones anteriores 1277

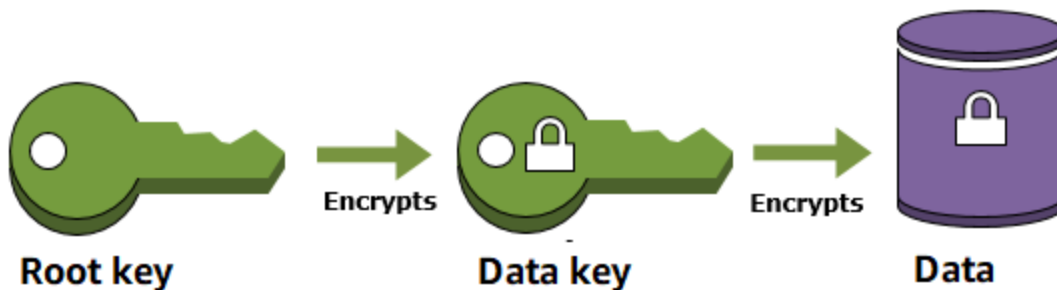
..... mcclxxxii

AWS Key Management Service

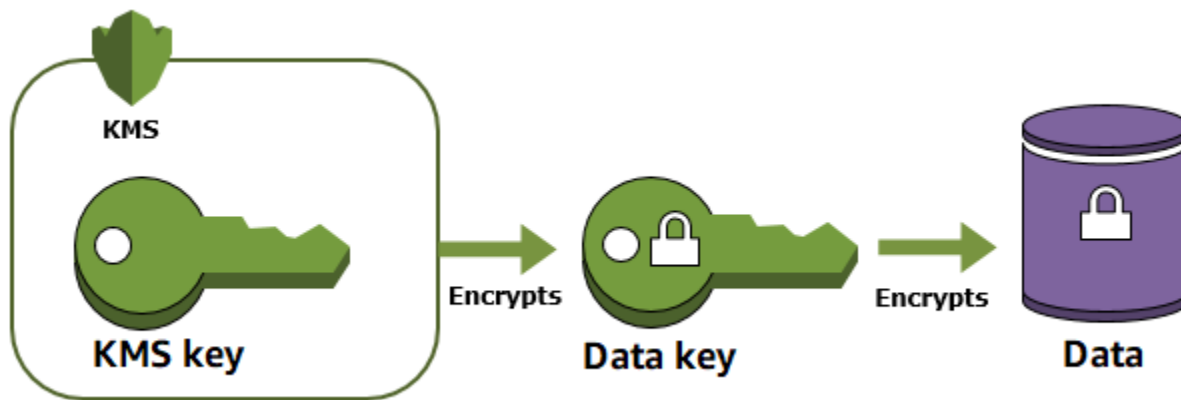
AWS Key Management Service (AWS KMS) es un servicio AWS gestionado que facilita la creación y el control de las claves que se utilizan para cifrar y firmar los datos. Las AWS KMS keys que cree AWS KMS están protegidas por [módulos de seguridad de hardware \(HSM\) validados por la norma FIPS 140-3 de nivel de seguridad 3](#). Nunca salen sin cifrar. AWS KMS Para usar o administrar sus claves KMS, usted interactúa con AWS KMS.

¿Por qué AWS KMS usarlas?

Cuando cifra datos, debe proteger la clave de cifrado. Si cifra la clave, debe proteger su clave de cifrado. En última instancia, debe proteger la clave de cifrado de nivel más alto (conocida como clave raíz) de la jerarquía que protege sus datos. Ahí es donde AWS KMS entra en juego.



AWS KMS protege tus claves raíz. Las claves de KMS se crean, administran, utilizan y eliminan por completo desde dentro AWS KMS. Nunca salen del servicio sin cifrar. Para usar o administrar sus claves KMS, debe llamar AWS KMS.



Además, puede crear y administrar [políticas clave para](#) garantizar que solo los usuarios de confianza tengan acceso a las claves de KMS. AWS KMS

AWS KMS in Regiones de AWS

Las Regiones de AWS opciones compatibles aparecen en [AWS Key Management Service Endpoints y Quotas](#). AWS KMS Si una AWS KMS función no es compatible con otra Región de AWS que sí la AWS KMS admite, la diferencia regional se describe en el tema sobre la función.

AWS KMS precios

Al igual que con otros AWS productos, su uso AWS KMS no requiere contratos ni compras mínimas. Para obtener más información sobre AWS KMS los precios, consulta [AWS Key Management Service Precios](#).

AWS KMS acuerdo de nivel de servicio

AWS Key Management Service está respaldado por un [acuerdo de nivel de servicio](#) que define nuestra política de disponibilidad de servicios.

Acceder AWS Key Management Service

Puede trabajar con él AWS KMS de las siguientes maneras:

Temas

- [Consola de administración de AWS](#)
- [AWS Command Line Interface](#)
- [AWS KMS REST API](#)
- [AWS SDKs](#)
- [Uso de este servicio con un AWS SDK](#)
- [AWS Encryption SDK](#)
- [AWS KMS coherencia eventual](#)
- [Uso de TLS poscuántico híbrido con AWS KMS](#)
- [Conéctese a AWS KMS través de un punto final de VPC](#)
- [Compatibilidad con puntos de conexión de doble pila](#)

Consola de administración de AWS

La consola es una interfaz de usuario basada en la web para la administración de AWS KMS los AWS recursos. Si te has registrado en una Cuenta de AWS, puedes acceder a la AWS KMS consola iniciando sesión en la página AWS KMS de Consola de administración de AWS inicio Consola de administración de AWS y eligiendo una opción.

Permisos necesarios para usar la AWS KMS consola

Para trabajar con la AWS KMS consola, los usuarios deben tener un conjunto mínimo de permisos que les permita trabajar con AWS KMS los recursos de la que disponen Cuenta de AWS. Además de estos permisos de AWS KMS , los usuarios también deben disponer de permisos para enumerar usuarios de IAM y los roles de IAM. Si crea una política de IAM que sea más restrictiva que el mínimo de permisos necesarios, la consola de AWS KMS no funcionará del modo esperado para los usuarios con esa política de IAM.

Para conocer los permisos mínimos necesarios para permitir a un usuario acceso de solo lectura a la consola de AWS KMS , consulte [Permita que un usuario vea las claves de KMS en la AWS KMS consola](#).

Para permitir que los usuarios trabajen con la AWS KMS consola para crear y administrar las claves de KMS, adjunte la política `AWSKeyManagementServicePowerUser` administrada al usuario, tal y como se describe en [AWS políticas gestionadas para AWS Key Management Service](#).

No es necesario que concedas permisos mínimos de consola a los usuarios que trabajan con la AWS KMS API a través de [AWS SDKsAWS Command Line Interface](#), o [Herramientas de AWS para PowerShell](#). Sin embargo, es necesario conceder permiso a estos usuarios para utilizar la API. Para obtener más información, consulte [Referencia de permisos](#).

AWS Command Line Interface

Puede usar las AWS CLI herramientas para emitir comandos o crear scripts en la línea de comandos de su sistema para realizar AWS (incluidas AWS KMS) tareas.

Para obtener más información sobre el AWS KMS uso de AWS CLI, consulte la [Referencia de AWS CLI comandos](#)

AWS KMS REST API

La arquitectura de AWS KMS está diseñada para ser neutral en cuanto al lenguaje de programación. REST APIEs una interfaz HTTP para. AWS KMS Con la REST API, puede utilizar solicitudes HTTP estándar para crear, recuperar y eliminar claves.

Para obtener más información sobre el uso de AWS KMS REST API, consulte la [referencia de la AWS Key Management Service API](#)

AWS SDKs

AWS proporciona SDKs (kits de desarrollo de software) que constan de bibliotecas y código de muestra para plataformas y lenguajes de programación comunes (Java JavaScript, C, Python, etc.). AWS SDKs Proporcionan una forma cómoda de crear un acceso programático a AWS KMS y AWS. AWS KMS es un REST servicio. Puedes enviar solicitudes a AWS KMS través de las bibliotecas del AWS SDK, que agrupan las tareas subyacentes AWS KMS REST API y simplifican tus tareas de programación. Para obtener información sobre ellas AWS SDKs, incluida la forma de descargarlas e instalarlas, consulta [Herramientas sobre las que basarse AWS](#).

[Ejemplos de código para AWS KMS usar AWS SDKs](#) Proporciona un buen punto de partida para su uso AWS KMS a través del AWS SDKs.

Uso de este servicio con un AWS SDK

AWS Los kits de desarrollo de software (SDKs) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK para C++	AWS SDK para C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK para Go	AWS SDK para Go ejemplos de código
AWS SDK para Java	AWS SDK para Java ejemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK para .NET	AWS SDK para .NET ejemplos de código
AWS SDK para PHP	AWS SDK para PHP ejemplos de código
Herramientas de AWS para PowerShell	Herramientas de AWS para PowerShell ejemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) ejemplos de código
AWS SDK para Ruby	AWS SDK para Ruby ejemplos de código
AWS SDK para Rust	AWS SDK para Rust ejemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP ejemplos de código
AWS SDK para Swift	AWS SDK para Swift ejemplos de código

Ejemplo de disponibilidad

¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

AWS Encryption SDK

AWS Encryption SDK Es una herramienta para implementar el cifrado del lado del cliente en su aplicación. No proporciona acceso completo a KMS, sino que se integra con AWS KMS el SDK independiente o se puede usar como un SDK independiente sin hacer referencia a las claves de KMS. Hay bibliotecas disponibles para Java JavaScript, C, Python y otros lenguajes de programación.

Para obtener más información, consulte la [Guía para desarrolladores de AWS Encryption SDK](#).

AWS KMS key políticas y políticas de IAM

AWS KMS coherencia eventual

La AWS KMS API sigue un modelo de [coherencia eventual](#) debido a la naturaleza distribuida del sistema. Como resultado, es posible que los cambios en AWS KMS los recursos no se vean inmediatamente en los siguientes comandos que ejecute.

Al realizar llamadas a la AWS KMS API, es posible que se produzca un breve retraso antes de que el cambio esté disponible en todo momento AWS KMS. Por lo general, el cambio tarda menos de unos segundos en propagarse por todo el sistema, pero en algunos casos puede tardar varios minutos. Es posible que se produzcan errores inesperados, como un error `NotFoundException` o `InvalidStateException`, en este periodo de tiempo. Por ejemplo, AWS KMS puede devolver un `NotFoundException` si llamas `GetParametersForImport` inmediatamente después de `llamarCreateKey`.

Le recomendamos que configure una estrategia de reintentos en sus AWS KMS clientes para reintentar automáticamente las operaciones tras un breve período de espera. Para obtener más información, consulte el [comportamiento de los reintentos](#) en la Guía de referencia de herramientas AWS SDKs y herramientas.

En el caso de las llamadas a la API relacionadas con las concesiones, puede [utilizar un token de concesión](#) para evitar posibles demoras y utilizar los permisos de una concesión de forma inmediata. Para obtener más información, consulte [Eventual consistency \(for grants\)](#).

Uso de TLS poscuántico híbrido con AWS KMS

AWS Key Management Service (AWS KMS) admite una opción híbrida de intercambio de claves poscuántico para el protocolo de cifrado de red Transport Layer Security (TLS). Puede utilizar esta opción de TLS cuando se conecte a los puntos de conexión de la API de AWS KMS. Estas características opcionales de intercambio híbrido postcuántico de claves son al menos tan seguras como el cifrado TLS que utilizamos hoy en día y es muy probable que aporten beneficios de seguridad adicionales. Sin embargo, afectan a la latencia y a la velocidad si las comparamos con los protocolos clásicos de intercambio de claves que se utilizan hoy en día.

Los datos que envías a AWS Key Management Service (AWS KMS) están protegidos en tránsito mediante el cifrado que proporciona una conexión de Transport Layer Security (TLS). Los conjuntos de cifrado clásicos que AWS KMS admite para las sesiones de TLS convierten en inviables los ataques de fuerza bruta en los mecanismos de intercambio de claves con la tecnología actual. Sin embargo, si la informática cuántica a gran escala tiene efectos prácticos en el futuro, los conjuntos de cifrado clásicos utilizados en los mecanismos de intercambio de claves TLS serán susceptibles a estos ataques. Si está desarrollando aplicaciones que se basan en la confidencialidad a largo plazo de los datos que se transmiten a través de una conexión TLS, debería plantearse un plan para migrar a la criptografía poscuántica antes de que se puedan utilizar ordenadores cuánticos a gran escala. AWS está trabajando para prepararse para este futuro y queremos que usted también esté bien preparado.

Para proteger los datos cifrados hoy contra posibles ataques futuros, AWS participa con la comunidad criptográfica en el desarrollo de algoritmos cuánticos resistentes o poscuánticos. Hemos implementado conjuntos de cifrado híbridos de intercambio de claves poscuántico AWS KMS que combinan elementos clásicos y poscuánticos para garantizar que su conexión TLS sea al menos tan sólida como lo sería con los conjuntos de cifrado clásicos.

[En la mayoría de los casos, estos conjuntos de cifrado híbridos están disponibles para su uso en sus cargas de trabajo de producción. Regiones de AWS](#) Sin embargo, dado que las características de rendimiento y los requisitos de ancho de banda de los conjuntos de cifrado híbridos son diferentes de los de los mecanismos de intercambio de claves clásicos, le recomendamos que [los pruebe en sus llamadas a la AWS KMS API en](#) diferentes condiciones.

Comentarios

Como siempre, agradecemos sus comentarios y su participación en nuestros repositorios de código abierto. Nos gustaría especialmente saber cómo interactúa su infraestructura con esta nueva variante del tráfico TLS.

- Para proporcionar comentarios sobre este tema, utilice el enlace Feedback (Comentarios) situado en la esquina superior derecha de esta página.
- Estamos desarrollando estos conjuntos de cifrado híbridos en código abierto en el [s2n-tls](#) repositorio de GitHub. Para proporcionar comentarios sobre la usabilidad de los conjuntos de cifrado, o para compartir nuevas condiciones o resultados de pruebas, [abra una incidencia](#) en el repositorio s2n-tls.
- Estamos escribiendo ejemplos de código para usar el TLS poscuántico híbrido AWS KMS en el repositorio. [aws-kms-pq-tls-example](#) GitHub. Para hacer preguntas o compartir ideas sobre la configuración de su cliente HTTP o AWS KMS cliente para que utilice los conjuntos de cifrado híbridos, [cree una edición](#) en el repositorio. [aws-kms-pq-tls-example](#)

Compatible Regiones de AWS

Post-Quantum TLS for AWS KMS está disponible en todos los dispositivos Regiones de AWS compatibles AWS KMS .

Para obtener una lista de los AWS KMS puntos finales de cada uno Región de AWS, consulte los [AWS Key Management Service puntos finales y las cuotas](#) en la Referencia general de Amazon Web Services. Para obtener más información acerca de los puntos de conexión FIPS, consulte [FIPS endpoints](#) en la Referencia general de Amazon Web Services.

Acerca del intercambio de claves postcuántico híbrido en TLS

AWS KMS admite conjuntos de cifrado híbridos de intercambio de claves poscuánticas. Puede utilizar Common Runtime AWS SDK for Java 2.x y AWS Common Runtime en sistemas Linux para configurar un cliente HTTP que utilice estos conjuntos de cifrado. A continuación, siempre que se conecte a un AWS KMS punto final con su cliente HTTP, se utilizarán los conjuntos de cifrado híbridos.

Este cliente HTTP utiliza [s2n-tls](#), que es una implementación de código abierto del protocolo TLS. Los conjuntos de cifrado híbridos que utiliza s2n-tls se implementan solo para el intercambio de claves, no para el cifrado de datos directo. Durante el intercambio de claves, el cliente y el servidor calculan la clave que utilizarán para cifrar y descifrar los datos en la red.

Los algoritmos que s2n-tls utiliza son un híbrido que combina la [curva elíptica de Diffie-Hellman](#) (ECDH), un algoritmo clásico de intercambio de claves que se utiliza hoy en día en TLS, con [Module-Lattice-Based Key-Encapsulation Mechanism](#) (ML-KEM) un algoritmo de cifrado de claves públicas y de establecimiento de claves que el Instituto Nacional de Estándares y Tecnología (NIST) [ha designado como su primer](#) algoritmo de acuerdo de claves poscuántico estándar. Este híbrido utiliza cada uno de los algoritmos de forma independiente para generar una clave. Luego combina las dos claves criptográficamente. Con s2n-tls, puede [configurar un cliente HTTP](#) con una preferencia por TLS postcuántico, que coloca a ECDH con ML-KEM en primer lugar en la lista de preferencias. Los algoritmos clásicos de intercambio de claves se incluyen en la lista de preferencias para garantizar la compatibilidad, pero están en una posición inferior en esta lista.

Uso de un TLS poscuántico híbrido con AWS KMS

Puede utilizar un TLS poscuántico híbrido para sus llamadas a AWS KMS. Cuando configure el entorno de prueba del cliente HTTP, tenga en cuenta la siguiente información:

Cifrado en tránsito

Los conjuntos de cifrado híbridos de s2n-tls se utilizan únicamente para el cifrado en tránsito. Protegen sus datos mientras viajan del cliente al punto final AWS KMS. AWS KMS no utiliza estos conjuntos de cifrado para cifrar los datos. AWS KMS keys

En cambio, cuando AWS KMS cifra los datos con claves KMS, utiliza criptografía simétrica con claves de 256 bits y el algoritmo AES-GCM (estándar de cifrado avanzado en modo contador de Galois), que ya es resistente a la tecnología cuántica. Los futuros e hipotéticos ataques de informática cuántica a gran escala a textos cifrados creados con claves AES-GCM de 256 bits [reducen la seguridad nominal de la clave a 128 bits](#). Este nivel de seguridad es suficiente para impedir los ataques de fuerza bruta contra textos cifrados. AWS KMS

Sistemas compatibles

El uso de los conjuntos de cifrado híbridos de s2n-tls solo es compatible actualmente con sistemas Linux. Además, estos conjuntos de cifrado solo son compatibles con AWS Common Runtime, como el SDKs AWS SDK for Java 2.x. Para ver un ejemplo, consulta [Configuración del cifrado TLS postcuántico híbrido](#).

AWS KMS Puntos de conexión

AWS KMS admite el TLS poscuántico híbrido en todos los puntos finales, incluidos los puntos finales validados por el [FIPS 140-3](#).

Configuración del cifrado TLS postcuántico híbrido

En este procedimiento, agregue una dependencia de Maven para el cliente HTTP de Common Runtime. AWS A continuación, configure un cliente HTTP que prefiera el TLS postcuántico. A continuación, cree un AWS KMS cliente que utilice el cliente HTTP.

Para ver ejemplos completos y funcionales de cómo configurar y usar el cifrado TLS híbrido postcuántico con AWS KMS, consulte el repositorio [aws-kms-pq-tls-example](#).

Note

El cliente HTTP AWS Common Runtime, que estaba disponible como versión preliminar, comenzó a estar disponible para el público en general en febrero de 2023. En esa versión, la clase `tlsCipherPreference` y el parámetro del método `tlsCipherPreference()` se sustituyen por el parámetro del método `postQuantumTlsEnabled()`. Si utilizó este ejemplo durante la vista previa, debe actualizar el código.

1. Agregue el cliente AWS Common Runtime a sus dependencias de Maven. Le recomendamos que utilice la última versión disponible.

Por ejemplo, esta instrucción agrega la versión 2.30.22 del cliente AWS Common Runtime a sus dependencias de Maven.

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

2. Para habilitar las suites de cifrado poscuántico híbridas, agréguelas AWS SDK for Java 2.x a su proyecto e inicialícelas. Luego habilite los conjuntos de cifrado postcuántico híbrido en su cliente HTTP como se muestra en el siguiente ejemplo.

Este código usa el parámetro del método `postQuantumTlsEnabled()` para configurar un [cliente HTTP de AWS en tiempo de ejecución común](#) que prefiera el conjunto de cifrado poscuántico híbrido recomendado, ECDH con ML-KEM. A continuación, utiliza el cliente HTTP configurado para crear una instancia del cliente AWS KMS asíncrono, [KmsAsyncClient](#).

Cuando se complete este código, todas las solicitudes de la [API de AWS KMS](#) de la instancia `KmsAsyncClient` utilizarán un TLS poscuántico 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. Pon a prueba tus AWS KMS llamadas con un TLS poscuántico híbrido.

Cuando llamas a las operaciones de la AWS KMS API en el AWS KMS cliente configurado, tus llamadas se transmiten al AWS KMS punto final mediante un TLS poscuántico híbrido. Para probar la configuración, llama a una AWS KMS API, como [ListKeys](#)

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

Prueba de su configuración de TLS postcuántico híbrido

Considere la posibilidad de ejecutar las siguientes pruebas con conjuntos de cifrado híbridos en las aplicaciones que AWS KMS llamen.

- Consulta la `tlsDetails` sección de la entrada de CloudTrail registro para ver una llamada a la AWS KMS API realizada por tu aplicación. El `keyExchange` campo debe mencionar un algoritmo híbrido, como `X25519MLKEM768`. Para ver un ejemplo, consulta [Descifra con una clave de cifrado simétrica estándar a través de una conexión TLS poscuántica](#).
- Realice pruebas comparativas utilizando un TLS poscuántico híbrido. El intercambio de claves híbridas aumenta el tamaño y el tiempo de procesamiento de algunos mensajes en el protocolo de enlace TLS, pero el impacto general en el rendimiento debería ser imperceptible en la mayoría de los casos.
- Intente conectarse desde diferentes ubicaciones. Según la ruta de red que utilice su solicitud, es posible que descubra que los servidores intermedios, proxies o firewalls antiguos con inspección profunda de paquetes (DPI) bloquean la solicitud. Esto puede deberse al uso de los nuevos grupos de intercambio de claves como [ClientHello](#) parte del protocolo de enlace TLS o a los mensajes de

intercambio de claves más grandes. Si tiene problemas para resolver estos problemas, colabore con su equipo de seguridad o con los administradores de TI para actualizar la configuración correspondiente y desbloquear los nuevos grupos de intercambio de claves TLS.

Obtenga más información sobre el TLS poscuántico en AWS KMS

Para obtener más información sobre el uso del TLS poscuántico híbrido en AWS KMS, consulta los siguientes recursos.

- [Para obtener más información sobre la criptografía poscuántica en AWS, incluidos enlaces a publicaciones de blogs y artículos de investigación, consulte Criptografía poscuántica.](#)
- Para obtener información acerca de s2n-tls, consulte [Presentación de s2n-tls, una nueva implementación de TLS de código abierto](#) y [Uso de s2n-tls](#).
- Para obtener información sobre el cliente HTTP de AWS Common Runtime, consulte [Configuración del cliente HTTP AWS basado en CRT en la](#) Guía para desarrolladores.AWS SDK for Java 2.x
- Para obtener información sobre el proyecto de criptografía postcuántica del Instituto Nacional de Estándares y Tecnología (NIST), consulte [Post-Quantum Cryptography](#).
- Para obtener información sobre la estandarización de la criptografía poscuántica del NIST, consulte [Estandarización de la criptografía poscuántica](#).

Conéctese a AWS KMS través de un punto final de VPC

Puede conectarse directamente a AWS KMS través de un punto final de interfaz privada en su nube privada virtual (VPC). Cuando utiliza un punto final de VPC de interfaz, la comunicación entre su VPC y la VPC AWS KMS se lleva a cabo íntegramente dentro de la red. AWS

AWS KMS es compatible con los puntos de conexión Amazon Virtual Private Cloud (Amazon VPC) con la tecnología de [AWS PrivateLink](#). Cada punto final de la VPC está representado por una o más [interfaces de red elásticas](#) (ENIs) con direcciones IP privadas en las subredes de la VPC.

El punto final de la interfaz de la VPC conecta su VPC directamente a AWS KMS sin una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una conexión. AWS Direct Connect Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse. AWS KMS

Regiones

AWS KMS admite puntos de enlace de VPC y políticas de puntos de enlace de VPC en todos los casos en las Regiones de AWS que es compatible. [AWS KMS](#)

Consideraciones sobre los puntos AWS KMS finales de VPC

Antes de configurar un punto final de la VPC de la interfaz AWS KMS, consulte el tema [Propiedades y limitaciones del punto final de la interfaz](#) en la AWS PrivateLink Guía.

AWS KMS el soporte para un punto final de VPC incluye lo siguiente.

- Puede utilizar el punto de conexión de VPC para llamar a todas las [operaciones de la API de AWS KMS](#) desde su VPC.
- Puede crear un punto final de VPC de interfaz que se conecte a un punto final de AWS KMS región o a un punto final de [AWS KMS FIPS](#).
- Puede usar AWS CloudTrail los registros para auditar el uso de las claves de KMS a través del punto de enlace de la VPC. Para obtener más información, consulte [Registro de AWS KMS solicitudes que utilizan un punto final de VPC](#).

Temas

- [Cree un punto final de VPC para AWS KMS](#)
- [Conéctese a un punto final AWS KMS de VPC](#)
- [Utilice los puntos finales de VPC para controlar el acceso a los recursos AWS KMS](#)
- [Registro de AWS KMS solicitudes que utilizan un punto final de VPC](#)

Cree un punto final de VPC para AWS KMS

Puede crear un punto de enlace de VPC AWS KMS mediante la consola de Amazon VPC o la API de Amazon VPC. Siga los procedimientos para [crear un punto de conexión de interfaz](#) mediante uno de los siguientes valores.

- Para crear un punto final de VPC para AWS KMS, utilice el siguiente nombre de servicio:

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

Por ejemplo, en la Región EE. UU. Oeste (Oregón) (us-west-2), el nombre del servicio sería:

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

- Para crear un punto de conexión de VPC que se conecte a un [punto de conexión FIPS de AWS KMS](#), utilice el siguiente nombre de servicio:

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

Por ejemplo, en la Región EE. UU. Oeste (Oregón) (us-west-2), el nombre del servicio sería:

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

Para facilitar el uso del punto de conexión de VPC, puede habilitar un [nombre de DNS privado](#) para el punto de conexión de VPC. Si selecciona la opción Enable DNS Name (Habilitar nombre DNS), el nombre del host de DNS de AWS KMS estándar se resuelve en el punto de conexión de VPC. Por ejemplo, `https://kms.us-west-2.amazonaws.com` se resolvería en un punto de conexión de VPC conectado al nombre del servicio `com.amazonaws.us-west-2.kms`.

Esta opción facilita el uso del punto de conexión de VPC. AWS SDKs Y AWS CLI utilizan el nombre de host AWS KMS DNS estándar de forma predeterminada, por lo que no es necesario especificar la URL del punto de conexión de la VPC en las aplicaciones y los comandos.

Para obtener más información, consulte [Acceso a un servicio a través de un punto de conexión de interfaz](#) en la Guía de AWS PrivateLink .

Conéctese a un punto final AWS KMS de VPC

Puede conectarse a AWS KMS través del punto de enlace de la VPC mediante un AWS SDK AWS CLI, el o. Herramientas de AWS para PowerShell Para especificar el punto de conexión de VPC, utilice su nombre de DNS.

Por ejemplo, este comando [list-keys](#) utiliza el parámetro `endpoint-url` para especificar el punto de conexión de VPC. Para utilizar un comando de este tipo, sustituya el ID del punto de conexión de VPC del ejemplo por uno de su cuenta.

```
$ aws kms list-keys --endpoint-url https://vpce-1234abcd5678c90a-09p7654s-us-east-1a.ec2.us-east-1.vpce.amazonaws.com
```

Permisos necesarios

Para que una AWS KMS solicitud que utiliza un punto de conexión de VPC se realice correctamente, el director requiere permisos de dos fuentes:

- Una [política de claves](#), [política de IAM](#), o bien [concesión](#) debe dar permiso a la entidad principal para llamar a la operación en el recurso (clave KMS o alias).
- Una política de extremo de VPC debe conceder a la entidad principal permiso para utilizar el punto de conexión para realizar la solicitud.

Por ejemplo, una política de clave puede conceder permiso a una entidad principal para llamar a [Descifrado](#) en una clave KMS en particular. Sin embargo, es posible que la política de punto de conexión de VPC no permita que la entidad principal llame a Decrypt en esa clave KMS mediante el punto de conexión.

O bien, una política de punto final de VPC podría permitir que un principal utilice el punto final para invocar determinadas claves [DisableKey](#) de KMS. Pero si la entidad principal no tiene esos permisos de una política de clave, política de IAM o concesión, se produce un error en la solicitud.

Puede crear una política de punto de conexión de VPC cuando cree el punto de conexión y puede cambiar la política de punto de conexión de VPC en cualquier momento. Utilice la consola de administración de VPC o las operaciones [CreateVpcEndpoint](#) [ModifyVpcEndpoint](#). También puede crear y cambiar una política de puntos finales de VPC [mediante una AWS CloudFormation plantilla](#). Para obtener ayuda sobre el uso de la consola de administración de la VPC, consulte [Creación de un punto de conexión de interfaz](#) y [Modificación de un punto de conexión de interfaz](#) en la Guía de AWS PrivateLink .

Nombres de host DNS privados

Si ha habilitado los nombres de host privados al crear el punto de conexión de VPC, no es necesario que especifique la URL de este en los comandos de la CLI ni en la configuración de la aplicación. El nombre del host de DNS de AWS KMS estándar se resuelve en el punto de conexión de VPC. AWS CLI Y SDKs usa este nombre de host de forma predeterminada para que puedas empezar a usar el punto de enlace de la VPC para conectarte a AWS KMS un punto de enlace regional sin cambiar nada en tus scripts y aplicaciones.

Para utilizar nombres de host privados, se deben establecer los atributos `enableDnsHostnames` y `enableDnsSupport` de la VPC en `true`. Para configurar estos atributos, utilice la [ModifyVpcAttribute](#) operación. Para obtener más información, consulte [Ver y actualizar los atributos de DNS de su VPC](#) en la Guía del usuario de Amazon VPC.

Utilice los puntos finales de VPC para controlar el acceso a los recursos AWS KMS

Puede controlar el acceso a AWS KMS los recursos y las operaciones cuando la solicitud proviene de la VPC o utiliza un punto final de la VPC. Para ello, utilice una de las siguientes [claves de condición global](#) en una [política de claves](#) o [política de IAM](#).

- Utilice la clave de condición `aws:sourceVpce` para conceder o restringir el acceso en función del punto de conexión de VPC.
- Utilice la clave de condición `aws:sourceVpc` para conceder o restringir el acceso en función de la VPC que aloja el punto de conexión privado.

Note

Actúe con precaución al crear políticas de IAM y de claves basadas en el punto de conexión de VPC. Si una declaración de política exige que las solicitudes provengan de una VPC o un punto de enlace de VPC en particular, es posible que se produzcan errores en las solicitudes de AWS los servicios integrados que utilizan un AWS KMS recurso en tu nombre. Para obtener ayuda, consulte [Uso de condiciones de punto final de VPC en políticas con AWS KMS permisos](#).

Además, la clave de condición `aws:sourceIP` no es efectiva si la solicitud procede de un [punto de conexión de Amazon VPC](#). Para restringir las solicitudes a un punto de conexión de VPC, utilice las claves de condición `aws:sourceVpce` o `aws:sourceVpc`. Para obtener más información, consulte [Administración de identidades y accesos para puntos de conexión de VPC y servicios de puntos de conexión de VPC](#) en la Guía de AWS PrivateLink .

Puedes usar estas claves de condición globales para controlar el acceso a AWS KMS keys (claves KMS), a los alias y a operaciones como esas [CreateKey](#) que no dependen de ningún recurso en particular.

Por ejemplo, la siguiente política de claves de ejemplo permite a un usuario realizar operaciones de cifrado con una clave KMS solo cuando la solicitud utiliza el punto de conexión de VPC especificado. Cuando un usuario realiza una solicitud a AWS KMS, el ID de punto final de la VPC de la solicitud se compara con el valor de la clave de `aws:sourceVpce` condición de la política. Si no coinciden, la solicitud se deniega.

Para usar una política como esta, reemplaza el Cuenta de AWS ID del marcador de posición y el IDs punto de enlace de VPC por valores válidos para tu cuenta.

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"
        }
      }
    }
  ]
}
```

También puede utilizar la clave de condición `aws:sourceVpc` para restringir el acceso a las claves KMS en función de la VPC en la que reside el punto de conexión.

La siguiente política de claves de ejemplo permite comandos que administran la clave KMS solo cuando proceden de `vpc-12345678`. Además, permite comandos que utilizan la clave KMS para operaciones criptográficas únicamente si proceden de `vpc-2b2b2b2b`. Podría utilizar una política como esta en caso de que una aplicación se ejecute en una VPC, pero utiliza una segunda VPC aislada para funciones de administración.

Para usar una política como esta, reemplaza el Cuenta de AWS ID del marcador de posición y el IDs punto de enlace de VPC por valores válidos para tu cuenta.

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": "*"
  }
]
}

```

Registro de AWS KMS solicitudes que utilizan un punto final de VPC

AWS CloudTrail registra todas las operaciones que utilizan el punto final de la VPC. Cuando una solicitud AWS KMS utiliza un punto de enlace de VPC, el ID del punto de enlace de VPC aparece en la entrada de [AWS CloudTrail registro que registra](#) la solicitud. Puede usar el ID del punto final para auditar el uso del punto final de la AWS KMS VPC.

Sin embargo, tus CloudTrail registros no incluyen las operaciones solicitadas por los responsables de otras cuentas ni las solicitudes de AWS KMS operaciones en las claves y los alias de KMS en otras cuentas. Además, para proteger su VPC, las solicitudes que son denegadas por una política

de puntos de conexión de la VPC, pero de lo contrario se habría permitido, no se registran en [AWS CloudTrail](#).

Por ejemplo, esta entrada de log de ejemplo registra una solicitud [GenerateDataKey](#) que ha utilizado el punto de enlace de la VPC. El campo `vpcEndpointId` aparece al final de la 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",
  "vpcEndpointId": "vpce-1234abcd5678c90a"
}
```

Compatibilidad con puntos de conexión de doble pila

AWS KMS proporciona un punto final público de doble pila que admite tanto a los clientes como IPv6 a IPv4 los clientes. Un punto final de doble pila permite a los clientes comunicarse AWS KMS con cualquiera IPv4 de las direcciones. IPv6 Para obtener más información sobre los puntos de AWS KMS conexión, consulte puntos de conexión y [AWS Key Management Service cuotas](#).

El terminal público AWS KMS de doble pila `https://kms.your-region.api.aws` es compatible con ambos IPv4 clientes. IPv6 AWS KMS también es accesible de forma privada a través IPv4 y IPv6 desde su nube privada virtual (VPC) mediante. AWS PrivateLink Para obtener más información sobre la creación de puntos de enlace de VPC de interfaz privada para AWS KMS, consulte. [Conéctese a AWS KMS través de un punto final de VPC](#)

Para obtener más información sobre sus IPv6 direcciones VPCs, consulte [Cómo funciona Amazon VPC](#) en la Guía del usuario de Amazon Virtual Private Cloud. Para obtener más información sobre cómo configurar su VPC para el modo de doble pila, consulte el [direccionamiento IP para su red VPCs y sus subredes](#) en la Guía del usuario de Amazon Virtual Private Cloud.

Las funciones no están disponibles en IPv6

AWS KMS no puede comunicarse IPv6 con almacenes de AWS CloudHSM claves o almacenes de claves externos. Esta limitación no le impide llamar al AWS KMS APIs extranjero IPv6.

AWS KMS conceptos

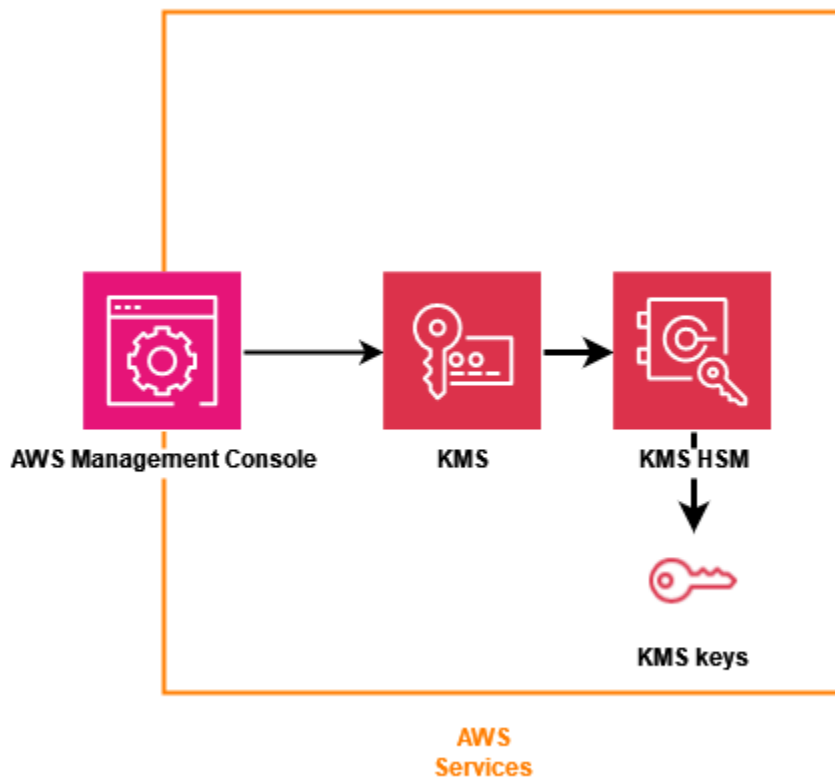
Conozca los términos y conceptos básicos que se utilizan en AWS Key Management Service (AWS KMS) y cómo funcionan juntos para ayudar a proteger sus datos.

Introducción a AWS KMS

AWS Key Management Service (AWS KMS) proporciona una interfaz web para generar y administrar claves criptográficas y funciona como proveedor de servicios criptográficos para proteger los datos. AWS KMS ofrece servicios de administración de claves tradicionales integrados con otros AWS servicios para proporcionar una visión uniforme de las claves de los clientes en todas sus áreas AWS, con administración y auditoría centralizadas.

AWS KMS incluye una interfaz web a través de la Consola de administración de AWS interfaz de línea de comandos y las operaciones de RESTful API para solicitar operaciones criptográficas de una flota distribuida de módulos de seguridad de hardware validados por el FIPS 140-3 (). HSMs El AWS KMS HSM es un dispositivo criptográfico de hardware independiente y multichip diseñado para proporcionar funciones criptográficas dedicadas a cumplir con los requisitos de seguridad y escalabilidad de. AWS KMS Puede establecer su propia jerarquía criptográfica basada en el HSM con claves que administra como AWS KMS keys. Estas claves están disponibles únicamente en la memoria HSMs y solo en ella durante el tiempo necesario para procesar su solicitud criptográfica. Puede crear varias claves de KMS, cada una representada por su ID de clave. Solo en las funciones y cuentas de AWS IAM administradas por cada cliente se pueden crear, eliminar o utilizar las claves de KMS administradas por el cliente para cifrar, descifrar, firmar o verificar datos. Puede definir los controles de acceso sobre quién puede administrar el and/or uso de las claves de KMS mediante la creación de una política adjunta a la clave. Estas políticas permiten definir usos específicos de la aplicación para las claves en cada operación de la API.

Además, la mayoría de AWS los servicios admiten el cifrado de datos en reposo mediante claves KMS. Esta capacidad permite a los clientes controlar cómo y cuándo los AWS servicios pueden acceder a los datos cifrados, controlando cómo y cuándo se puede acceder a las claves KMS.



AWS KMS es un servicio por niveles que consta de AWS KMS hosts orientados a la web y un nivel de HSMs. La agrupación de estos hosts en niveles forma la pila. AWS KMS. Todas las solicitudes AWS KMS deben realizarse a través del protocolo de seguridad de la capa de transporte (TLS) y terminar en un host. AWS KMS [los hosts solo permiten el TLS con un conjunto de cifrado que proporciona un secreto directo perfecto](#). AWS KMS autentica y autoriza sus solicitudes utilizando los mismos mecanismos de credenciales y políticas AWS Identity and Access Management (IAM) que están disponibles para todas las demás operaciones de la API. AWS

AWS KMS objetivos de diseño

AWS KMS está diseñado para cumplir los siguientes requisitos.

Durabilidad

La durabilidad de las claves criptográficas está diseñada para ser igual a la de los servicios de mayor durabilidad de los que disponen. AWS. Una única clave criptográfica puede cifrar grandes volúmenes de datos acumulados durante mucho tiempo.

Confiabilidad

El uso de las claves está protegido por las políticas de control de acceso que usted define y administra. No existe ningún mecanismo para exportar claves de KMS de texto no cifrado. La confidencialidad de las claves criptográficas es crucial. Se requieren varios empleados de Amazon con acceso específico a los controles de acceso basados en quóruns para realizar acciones administrativas en el. HSMs

Baja latencia y alto rendimiento

AWS KMS proporciona operaciones criptográficas con niveles de latencia y rendimiento adecuados para su uso en otros servicios en. AWS

Regiones independientes

AWS proporciona regiones independientes para los clientes que necesitan restringir el acceso a los datos en diferentes regiones. El uso de claves se puede aislar dentro de una Región de AWS.

Fuente segura de números aleatorios

Dado que una criptografía sólida depende de una generación de números aleatorios realmente impredecible, AWS KMS proporciona una fuente de números aleatorios validada y de alta calidad.

Auditoría

AWS KMS registra el uso y la administración de las claves criptográficas en AWS CloudTrail los registros. Puede utilizar AWS CloudTrail los registros para inspeccionar el uso de sus claves criptográficas, incluido el uso de las claves por parte de los AWS servicios que actúan en su nombre.

Para lograr estos objetivos, el AWS KMS sistema incluye un conjunto de AWS KMS operadores y operadores de hospedaje de servicios (denominados colectivamente «operadores») que administran los «dominios». Un dominio es un conjunto de AWS KMS servidores y operadores definido a nivel regional. HSMs Cada AWS KMS operador tiene un token de hardware que contiene un key pair de claves pública y privada que se utiliza para autenticar sus acciones. HSMs Tienen un par de claves públicas y privadas adicionales para establecer claves de cifrado que protegen la sincronización de estados del HSM.

AWS KMS keys

Las claves KMS que crea y administra para usarlas en sus propias aplicaciones criptográficas son de un tipo conocido como claves administradas por el cliente. Las claves administradas por el cliente

también se pueden usar junto con AWS los servicios que usan claves de KMS para cifrar los datos que el servicio almacena en su nombre. Las claves administradas por el cliente se recomiendan para los clientes que desean tener un control total sobre el ciclo de vida y el uso de sus claves. Disponer de una clave administrada por el cliente en su cuenta tiene asociado un coste mensual. Además, las solicitudes de uso y and/or gestión de la clave conllevan un coste de uso. Para obtener más información, consulte [AWS Key Management Service Precios](#).

Hay casos en los que un cliente puede querer un AWS servicio para cifrar sus datos, pero no quiere la sobrecarga que supone administrar las claves y no quiere pagar por una clave. Una Clave administrada de AWS es una clave de KMS que existe en su cuenta, pero que solo se puede usar en determinadas circunstancias. En concreto, solo la pueden usar en el contexto del AWS servicio en el que operas y solo la pueden usar los responsables de la cuenta en la que existe la clave. No puede administrar nada relacionado con el ciclo de vida o los permisos de estas claves. A medida que utilizas las funciones de cifrado en AWS los servicios, es posible que veas Claves administradas por AWS que utilizan un alias con el formato «aws<service code>». Por ejemplo, una clave aws/ebs solo se puede usar para cifrar los volúmenes de EBS y solo para los volúmenes utilizados por las entidades principales de IAM en la misma cuenta que la clave. Piensa en una Clave administrada de AWS que esté restringida para que solo los usuarios de tu cuenta la usen para los recursos de tu cuenta. No puedes compartir recursos cifrados en o Clave administrada de AWS con otras cuentas. Si bien la existencia de una en tu cuenta Clave administrada de AWS es gratuita, el AWS servicio asignado a la clave te cobrará por el uso de este tipo de clave.

Claves administradas por AWS son un tipo de clave heredado que dejará de crearse para nuevos AWS servicios a partir de 2021. En cambio, los AWS servicios nuevos (y antiguos) utilizan lo que se conoce como una Clave propiedad de AWS para cifrar los datos de los clientes de forma predeterminada. An Clave propiedad de AWS es una clave de KMS que se encuentra en una cuenta administrada por el AWS servicio, por lo que los operadores del servicio pueden administrar su ciclo de vida y sus permisos de uso. Al Claves propiedad de AWS utilizarlos, AWS los servicios pueden cifrar sus datos de forma transparente y permiten compartir fácilmente los datos entre cuentas o regiones sin que tenga que preocuparse por los permisos clave. Úselos Claves propiedad de AWS para encryption-by-default cargas de trabajo que proporcionan una protección de datos más sencilla y automatizada. Como estas claves son propiedad de y están administradas por él AWS, no se le cobra por su existencia o su uso, no puede cambiar sus políticas, no puede auditar las actividades relacionadas con estas claves y no puede eliminarlas. Utilice las claves administradas por el cliente cuando el control sea importante, pero utilice Claves propiedad de AWS cuando lo más importante sea la comodidad.

	Claves administradas por el cliente	Claves administradas por AWS	Claves propiedad de AWS
Política de claves	Controlada exclusivamente por el cliente	Controlada por el servicio; visible por el cliente	Controladas de forma exclusiva y solo visibles por el AWS servicio que cifra sus datos
Registro	CloudTrail almacén de datos de eventos o registros de clientes	CloudTrail almacén de datos de eventos o seguimiento de clientes	No visible por el cliente
Administración del ciclo de vida	El cliente administra la rotación, la eliminación y la ubicación regional	AWS KMS gestiona la rotación (anual), la eliminación y la ubicación regional	Servicio de AWS gestiona la rotación, la eliminación y la ubicación regional
Precios	Cuota mensual por existencia de claves (prorrataada por hora). También se cobra por el uso de claves	Sin cuota mensual, pero a la persona que llama se le cobra por el uso de la API en estas claves	Sin cargos para el cliente

Las claves KMS que usted crea son [claves administradas por el cliente](#). Los Servicios de AWS que utilizan claves KMS para cifrar los recursos de servicio a menudo crean claves para usted. Las claves KMS que se Servicios de AWS crean en su AWS cuenta son [Claves administradas por AWS](#). Las claves KMS que se Servicios de AWS crean en una cuenta de servicio son [Claves propiedad de AWS](#).

Tipo de clave KMS	Puede ver los metadatos clave KMS	Puede administrar la clave KMS	Creadas en mi Cuenta de AWS	Rotación automática	Precios
Clave administrada por clientes	Sí	Sí	Sí	Opcional.	Cuota mensual (prorrateada por hora) Tarifa por uso
Clave administrada de AWS	Sí	No	Sí	Obligatorio. Cada año (aproximadamente 365 días)	Sin cuota mensual Tarifa por uso (algunos Servicios de AWS pagan esta tarifa por ti)
Clave propiedad de AWS	No	No	No	Servicio de AWS Gestiona la estrategia de rotación	Sin cuotas

[AWS los servicios que se integran AWS KMS difieren en su compatibilidad con](#) las claves KMS.

Algunos AWS servicios cifran sus datos de forma predeterminada con una Clave propiedad de AWS o una Clave administrada de AWS. Algunos AWS servicios admiten claves administradas por el cliente. Otros AWS servicios admiten todos los tipos de claves de KMS para que pueda acceder fácilmente a una Clave propiedad de AWS clave gestionada por el cliente Clave administrada de AWS, verla o controlarla. Para obtener información detallada sobre las opciones de cifrado que ofrece un AWS servicio, consulte el tema El cifrado en reposo de la guía del usuario o la guía para desarrolladores del servicio.

Claves administradas por el cliente

Las claves KMS que usted crea son claves administradas por el cliente. Las claves administradas por el cliente son claves de KMS Cuenta de AWS que usted crea, posee y administra. Puede controlar por completo estas claves KMS, incluido el establecimiento y el mantenimiento de sus [políticas de claves](#), [políticas de IAM y concesiones](#), [su habilitación y deshabilitación](#), [la rotación de su material criptográfico](#), [la adición de etiquetas](#), [la creación de alias](#) que hacen referencia a la clave KMS y la [programación de las claves KMS para su eliminación](#).

Las claves administradas por el cliente aparecen en la página Claves administradas por el cliente de la Consola de administración de AWS de AWS KMS. Para identificar definitivamente una clave administrada por el cliente, utilice la operación [DescribeKey](#). En el caso de las claves administradas por el cliente, el valor del campo KeyManager de la respuesta DescribeKey es CUSTOMER.

Puede usar su clave administrada por el cliente en operaciones criptográficas y auditar su uso en los registros de AWS CloudTrail. Además, muchos [servicios de AWS integrados con AWS KMS](#) le permiten especificar una clave administrada por el cliente para proteger los datos que almacenan y administran para usted.

Las claves administradas por el cliente tienen una tarifa mensual y una tarifa por uso excesivo del nivel gratuito. Se tienen en cuenta para deducir de las AWS KMS [cuotas](#) de su cuenta. Consulte los [Precios de AWS Key Management Service](#) y [Cuotas](#) para obtener más información.

Claves administradas por AWS

Claves administradas por AWS son claves de KMS de su cuenta que un [AWS servicio integrado](#) en ella crea, administra y utiliza en su nombre AWS KMS.

Algunos AWS servicios le permiten elegir una clave gestionada por el cliente Clave administrada de AWS o una clave gestionada por el cliente para proteger los recursos de ese servicio. En general, a menos que se le pida que controle la clave de cifrado que protege sus recursos, una Clave administrada de AWS es una buena opción. No tiene que crear ni mantener la clave o su política de claves, y nunca hay una tarifa mensual por una Clave administrada de AWS.

Tiene permiso para [consultarlas Claves administradas por AWS](#) en su cuenta, [ver sus políticas clave](#) y [auditar su uso](#) en AWS CloudTrail los registros. Sin embargo, no puedes cambiar ninguna de sus propiedades Claves administradas por AWS, rotarlas, cambiar sus políticas clave ni programar su eliminación. Además, no puede utilizarlos directamente Claves administradas por AWS en operaciones criptográficas; el servicio que los crea los utiliza en su nombre.

[Las políticas de control de recursos](#) de su organización no se aplican a las Claves administradas por AWS.

Claves administradas por AWS aparecen en la Claves administradas por AWS página del formulario Consola de administración de AWS . AWS KMS También puede identificarlos Claves administradas por AWS por sus alias, que tienen el formato `aws/service-name`, por ejemplo `aws/redshift`. Para identificar definitivamente un Claves administradas por AWS, utilice la [DescribeKey](#) operación. En el caso de Claves administradas por AWS, el valor del campo `KeyManager` de la respuesta `DescribeKey` es AWS.

Todos Claves administradas por AWS se rotan automáticamente cada año. No puede cambiar esta programación de rotación.

Note

En mayo de 2022, AWS KMS cambiamos el programa de rotación Claves administradas por AWS de cada tres años (aproximadamente 1.095 días) a cada año (aproximadamente 365 días).

No hay una cuota mensual para Claves administradas por AWS. Pueden estar sujetos a cargos si se utilizan más allá del nivel gratuito, pero algunos AWS servicios cubren estos costes. Para obtener más detalles, consulte el tema Cifrado en reposo en la Guía del usuario o en la Guía para desarrolladores del servicio. Consulte los [Precios de AWS Key Management Service](#) para obtener más información.

Claves administradas por AWS no se tienen en cuenta las cuotas de recursos en la cantidad de claves de KMS en cada región de su cuenta. Sin embargo, cuando se utilizan en nombre de una entidad principal en su cuenta, estas claves KMS se contabilizan en las cuotas de solicitud. Para obtener más información, consulte [Cuotas](#).

Claves propiedad de AWS

Claves propiedad de AWS son un conjunto de claves de KMS que un AWS servicio posee y administra para su uso en varios Cuentas de AWS. Aunque no Claves propiedad de AWS estén en tus Cuenta de AWS manos, AWS cualquier servicio puede utilizarlas Clave propiedad de AWS para proteger los recursos de tu cuenta.

Algunos AWS servicios le permiten elegir una clave gestionada por el cliente Clave propiedad de AWS o una clave gestionada por el cliente. En general, a menos que tenga que auditar o controlar la

clave de cifrado que protege sus recursos, una clave propiedad de AWS es una buena opción. Claves propiedad de AWS son completamente gratuitos (sin cuotas mensuales ni de uso), no se descuentan de las [AWS KMS cuotas](#) de tu cuenta y son fáciles de usar. No es necesario crear ni mantener la clave ni su política de claves.

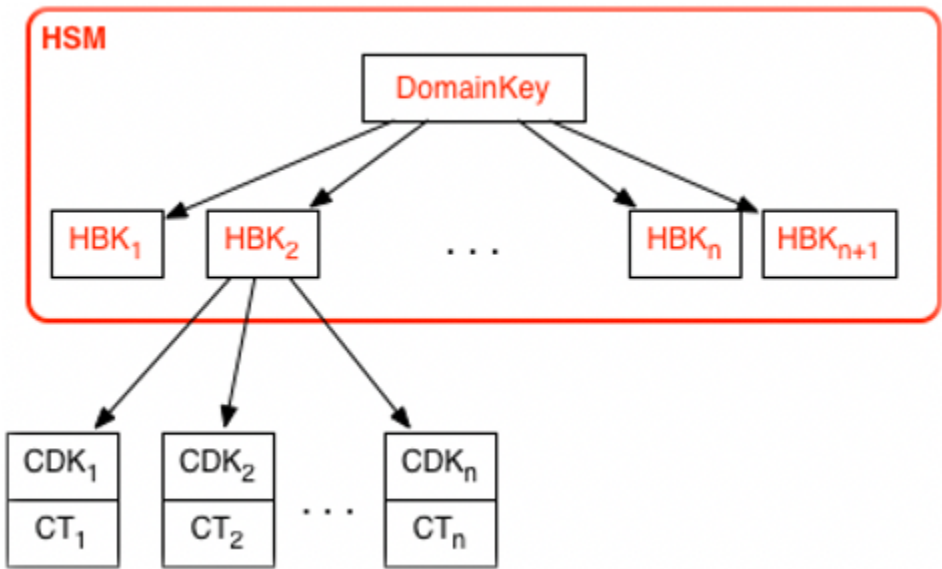
La rotación Claves propiedad de AWS varía según los servicios. Para obtener información sobre la rotación de un determinado servicio Clave propiedad de AWS, consulte el tema Cifrado en reposo en la guía del usuario o en la guía para desarrolladores del servicio.

AWS KMS key jerarquía

La jerarquía de claves comienza con una clave lógica de nivel superior, una AWS KMS key. Una clave de KMS representa un contenedor para material clave de nivel superior y está definida de forma única dentro del espacio de nombres del servicio de AWS con un nombre de recurso de Amazon (ARN). El ARN incluye un identificador de clave generado de forma única, un ID de clave. Se crea una clave KMS en función de una solicitud iniciada por el usuario mediante AWS KMS. Al recibirla, AWS KMS solicita la creación de una clave de respaldo HSM (HBK) inicial para colocarla en el contenedor de claves KMS. La HBK se genera en un HSM en el dominio y está diseñada para no exportarse nunca del HSM en texto sin formato. En su lugar, la HBK se exporta cifrada con claves de dominio administradas por HSM. Estos elementos exportados se denominan claves exportadas (EKTs).

El EKT se exporta a un almacenamiento de larga duración y de baja latencia. Por ejemplo, supongamos que recibe un ARN para la clave de KMS lógica. Esto representa la parte superior de una jerarquía de claves, o contexto criptográfico. Puede crear varias claves de KMS en su cuenta y establecer políticas en sus claves de KMS como cualquier otro recurso con AWS nombre.

Dentro de la jerarquía de una clave de KMS específica, la HBK puede considerarse como una versión de la clave de KMS. Cuando desee rotar la clave KMS AWS KMS, se creará un nuevo HBK que se asociará a la clave KMS como el HBK activo de la clave KMS. HBKs Las más antiguas se conservan y se pueden utilizar para descifrar y verificar datos previamente protegidos. Pero solo se puede usar la clave criptográfica activa para proteger nueva información.



Puede realizar solicitudes AWS KMS para usar sus claves de KMS para proteger directamente la información o solicitar claves adicionales generadas por HSM que estén protegidas por su clave de KMS. Estas claves se denominan claves de datos del cliente o. CDKs CDKs se pueden devolver cifradas como texto cifrado (CT), en texto plano o en ambos. Todos los objetos cifrados con una clave KMS (ya sean datos proporcionados por el cliente o claves generadas por HSM) solo se pueden descifrar en un HSM mediante una llamada directa. AWS KMS

El texto cifrado devuelto, o la carga útil descifrada, nunca se almacena en él. AWS KMS La información se le devuelve a través de su conexión TLS a AWS KMS. Esto también se aplica a las llamadas realizadas por los AWS servicios en su nombre.

La jerarquía de claves y las propiedades de clave específicas aparecen en la siguiente tabla.

Clave	Description (Descripción)	Ciclo de vida
Clave de dominio	Una clave AES-GCM de 256 bits solo en memoria de un HSM que se utiliza para ajustar versiones de las claves de KMS, las claves de backup de HSM.	Rotación diaria ¹
Clave de respaldo de HSM	Una clave simétrica de 256 bits, clave privada RSA o curva elíptica que se utiliza para proteger los datos y las claves del cliente y que	Rotación anual ² (config. opcional)

Clave	Description (Descripción)	Ciclo de vida
	se almacena de forma cifrada con claves de dominio. Una o más claves de backup de HSM comprenden la clave de KMS, representada por el keyId.	
Clave de cifrado derivada	Una clave AES-GCM de 256 bits solo en memoria de un HSM que se utiliza para cifrar datos y claves del cliente. Se deriva de una HBK para cada cifrado.	Se utiliza una vez por cifrado y se regenera al descifrarse.
Clave de datos del cliente	Clave simétrica o asimétrica delimitada por el usuario que se exporta desde un HSM en texto sin formato y texto cifrado. Se cifra con una clave de backup de HSM y se devuelve a los usuarios autorizados a través del canal TLS.	Rotación y uso controlado por aplicación

De vez en cuando, AWS KMS podría reducir la rotación de claves de dominio a, como máximo, una vez por semana para tener en cuenta las tareas de administración y configuración del dominio.

² Por defecto, las claves administradas por AWS se crean y gestionan en AWS KMS en tu nombre y se rotan automáticamente una vez al año.

Identificadores de clave (KeyId)

Los identificadores de clave actúan como nombres para las claves KMS. Le ayudan a reconocer sus claves KMS en la consola. Se utilizan para indicar qué claves KMS desea utilizar en operaciones de la API de AWS KMS, políticas de claves, políticas de IAM y concesiones. Los valores del identificador de clave no están relacionados en absoluto con el material de clave asociado a la clave KMS.

AWS KMS define varios identificadores de clave. Al crear una clave de KMS, AWS KMS genera un ARN de clave y un ID de clave, que son propiedades de la clave de KMS. Al crear un [alias](#), AWS KMS

genera un ARN de alias en función del nombre de alias que defina. Puede ver los identificadores de clave y alias en la API Consola de administración de AWS y en la AWS KMS misma.

En la AWS KMS consola, puede ver y filtrar las claves de KMS por su ARN de clave, ID de clave o nombre de alias, y ordenar por ID de clave y nombre de alias. Para obtener ayuda para encontrar los identificadores clave en la consola, consulte [the section called “Encontrar el ID de clave y el ARN de clave”](#).

En la AWS KMS API, los parámetros que se utilizan para identificar una clave de KMS tienen un nombre `KeyId` o una variante, como `TargetKeyId` o `DestinationKeyId`. Sin embargo, los valores de esos parámetros no se limitan a la clave IDs. Algunos pueden tomar cualquier identificador de clave válido. Para obtener información sobre los valores de cada parámetro, consulta la descripción del parámetro en la Referencia de la AWS Key Management Service API.

Note

Cuando utilices la AWS KMS API, ten cuidado con el identificador clave que utilices. APIs Los diferentes requieren identificadores clave diferentes. En general, utilice el identificador de clave más completo que sea práctico para su tarea.

AWS KMS admite los siguientes identificadores clave.

ARN de clave

El ARN de clave es el nombre de recurso de Amazon (ARN) de una clave KMS. Es un identificador único y completo para la clave KMS. Un ARN clave incluye la región y el Cuenta de AWS ID de clave. Para obtener ayuda para encontrar el ARN de clave de una clave KMS, consulte [the section called “Encontrar el ID de clave y el ARN de clave”](#).

El formato de un ARN de clave es el siguiente:

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

A continuación, se muestra un ARN de clave para una clave KMS de región única.

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

El *key-id* elemento de la clave ARNs de las [claves multirregionales](#) comienza con el `mrk-` prefijo. A continuación, se muestra un ARN de clave de ejemplo para una clave de múltiples regiones.

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

ID de clave

El ID de clave identifica de forma inequívoca una clave KMS dentro de una cuenta y región. Para obtener ayuda para encontrar el ID de clave de una clave KMS, consulte [the section called “Encontrar el ID de clave y el ARN de clave”](#).

A continuación, se muestra un ID de clave de ejemplo para una clave KMS de región única.

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

La clave IDs de las claves [multirregionales](#) comienza con el prefijo. `mrk-` A continuación, se muestra un ID de clave de ejemplo para una clave de múltiples regiones.

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

ARN de alias

El alias ARN es el nombre de recurso de Amazon (ARN) de un alias. AWS KMS Es un identificador único y completamente calificado para el alias, y para la clave KMS que representa. Un ARN de alias incluye la región Cuenta de AWS y el nombre del alias.

En cualquier momento dado, un ARN de alias identifica una clave KMS en particular. Sin embargo, dado que puede cambiar la clave KMS asociada con el alias, el ARN de alias puede identificar diferentes claves KMS en diferentes momentos. Para obtener ayuda para encontrar el ARN de alias de una clave KMS, consulte [Búsqueda del nombre del alias y el ARN de alias para una clave de KMS](#).

El formato de un ARN de alias es el siguiente:

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

A continuación, se muestra el ARN de alias de un `ExampleAlias` ficticio.

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

Nombre del alias

El nombre de alias es una cadena de hasta 256 caracteres. Identifica de forma inequívoca una clave KMS asociada dentro de una cuenta y región. En la AWS KMS API, los nombres de alias siempre comienzan `alias/` por. Para obtener ayuda para encontrar el nombre de alias de una clave KMS, consulte [Búsqueda del nombre del alias y el ARN de alias para una clave de KMS](#).

El formato de un nombre de alias es el siguiente:

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

Por ejemplo:

```
alias/ExampleAlias
```

El prefijo `aws/` de un nombre de alias está reservado para [Claves administradas por AWS](#). No se puede crear un alias con este prefijo. Por ejemplo, el nombre de alias Clave administrada de AWS de Amazon Simple Storage Service (Amazon S3) es el siguiente.

```
alias/aws/s3
```

Teclas asimétricas en AWS KMS

Una clave KMS asimétrica representa un par de claves privadas y públicas relacionadas matemáticamente. Puede entregar la clave pública a cualquiera, aunque no sea de confianza, pero la clave privada debe ser secreta.

En una clave KMS asimétrica, la clave privada se crea AWS KMS y nunca se queda AWS KMS sin cifrar. Para usar la clave privada, debe llamar a AWS KMS. Puedes usar la clave pública que contiene AWS KMS llamando a las operaciones de la AWS KMS API. O bien, puedes [descargar la clave pública](#) y utilizarla fuera de ella AWS KMS.

Si tu caso de uso requiere que los usuarios que no puedan llamar a AWS los cifren fuera del sistema AWS KMS, las claves KMS asimétricas son una buena opción. Sin embargo, si va a crear una clave KMS para cifrar los datos que almacena o administra en un AWS servicio, utilice una clave KMS de

cifrado simétrico. [AWS los servicios integrados AWS KMS utilizan](#) únicamente claves KMS de cifrado simétrico para cifrar los datos. Estos servicios no admiten cifrado con claves de KMS asimétricas.

Al firmar mensajes de más de 4 KB AWS KMS, debe aplicar un hash al mensaje AWS KMS antes de firmarlo. AWS KMS ofrece tres MessageType opciones para gestionar la introducción de mensajes: RAW para los mensajes de texto simple (donde se AWS KMS realiza el hash), DIGEST para los mensajes precodificados (donde se AWS KMS omite el paso de la codificación) y, EXTERNAL_MU específicamente, para las especificaciones clave del KMS de ML-DSA, en las que la entrada es un valor μ m representativo de 64 bytes. [En el caso de mensajes grandes que superen el límite de 4 KB, aplique un hash externo al mensaje y utilícelo MessageType :DIGEST\(o en el caso de las claves KMS de ML-DSA\) al realizar las operaciones de firma y MessageType :EXTERNAL_MUverificación.](#)

[AWS KMSAWS KMS](#)

AWS KMS admite varios tipos de claves KMS asimétricas.

Claves KMS de RSA

Una clave KMS con un par de claves RSA para el cifrado y el descifrado o para la firma y la verificación (pero no ambos). AWS KMS admite varias longitudes de clave para distintos requisitos de seguridad.

Para obtener información técnica sobre los algoritmos de cifrado y firma AWS KMS compatibles con las claves KMS de RSA, consulte las especificaciones de las claves de [RSA](#).

Claves KMS de curva elíptica (ECC)

Una clave KMS con un par de claves de curva elíptica para firmar y verificar o derivar secretos compartidos (pero no ambos). AWS KMS admite varias curvas de uso común.

Para obtener información técnica sobre los algoritmos de firma AWS KMS compatibles con las claves KMS de ECC, consulte las especificaciones de las claves de curva [elíptica](#).

Claves de KMS de ML-DSA

Una clave de KMS con un par de claves de ML-DSA para firma y verificación. El ML-DSA es un estándar de criptografía poscuántica desarrollado por el Instituto Nacional de Estándares y Tecnología (NIST) de EE. UU. para proteger contra las amenazas a la seguridad que plantea la computación cuántica. El ML-DSA es el algoritmo de firma digital recomendado para las organizaciones que están realizando el cambio de los algoritmos de firma digital RSA o de curva elíptica a la criptografía segura poscuántica.

AWS KMS admite varias longitudes de clave para diferentes requisitos de seguridad. Para obtener información técnica sobre los algoritmos de firma AWS KMS compatibles con las claves KMS de ML-DSA, consulte las especificaciones de claves de [ML-DSA](#).

SM2 Claves KMS (solo regiones de China)

Una clave KMS con un par de SM2 claves para el cifrado y el descifrado, la firma y la verificación o la obtención de secretos compartidos (debe elegir un [Key usage](#) tipo).

Para obtener información técnica sobre los algoritmos de cifrado y firma AWS KMS compatibles con las claves SM2 KMS (solo en las regiones de China), consulta las [especificaciones de las SM2 claves](#).

Para obtener ayuda para elegir la configuración de clave asimétrica, consulte [Elección del tipo de clave KMS que se va a crear](#).

Regiones

Las claves KMS asimétricas y los pares de claves de datos asimétricas son compatibles con todos Regiones de AWS los soportes. AWS KMS

Más información

- Para crear claves KMS asimétricas, consulte [Creación de una clave de KMS asimétrica](#).
- Para crear claves KMS asimétricas de varias regiones, consulte [Creación de claves primarias de varias regiones](#).
- Para aprender a firmar mensajes y verificar firmas con las claves KMS asimétricas, consulte [Firma digital con la nueva función de claves asimétricas de AWS KMS](#) en el Blog de seguridad de AWS .
- Para obtener más información sobre las consideraciones especiales a la hora de eliminar claves KMS asimétricas, consulte [Deleting asymmetric KMS keys](#).
- Para identificar y ver las claves KMS asimétricas, consulte [Identificación de claves KMS asimétricas](#).

Teclas HMAC en AWS KMS

Las claves KMS del código de autenticación de mensajes basado en hash (HMAC) son claves simétricas que se utilizan para generar y verificar su contenido. HMACs AWS KMS El material de claves exclusivo de cada clave HMAC de KMS proporciona la clave secreta que requieren los

algoritmos HMAC. Puede utilizar una clave KMS HMAC con las operaciones [GenerateMac](#) y [VerifyMac](#) para verificar la integridad y autenticidad de los datos dentro de AWS KMS.

Los algoritmos HMAC combinan una función hash criptográfica y una clave secreta compartida. Toman un mensaje y una clave secreta, como el material clave de una clave KMS HMAC, y devuelven un código o una etiqueta únicos de tamaño fijo. Si cambia incluso un carácter del mensaje, o si la clave secreta no es idéntica, la etiqueta resultante es totalmente diferente. Al requerir una clave secreta, el HMAC también proporciona autenticidad; es imposible generar una etiqueta HMAC idéntica sin la clave secreta. HMACs a veces se denominan firmas simétricas porque funcionan como firmas digitales, pero utilizan una sola clave tanto para la firma como para la verificación.

[Las claves HMAC KMS y los algoritmos HMAC que se AWS KMS utilizan cumplen con los estándares del sector definidos en el RFC 2104.](#) La AWS KMS [GenerateMac](#) operación genera etiquetas HMAC estándar. Las claves HMAC KMS se generan en módulos de seguridad de AWS KMS hardware que están certificados según el [Programa de validación de módulos criptográficos FIPS 140-3](#) (excepto en las regiones de China (Beijing) y China (Ningxia)) y nunca se dejan sin cifrar. AWS KMS Para utilizar una clave KMS HMAC, tiene que llamar a AWS KMS.

Puede utilizar claves KMS HMAC para determinar la autenticidad de un mensaje, como un token web JSON (JWT), información de tarjeta de crédito tokenizada o una contraseña enviada. También se pueden utilizar como funciones de derivación de claves seguras (KDFs), especialmente en aplicaciones que requieren claves deterministas.

Las claves HMAC KMS ofrecen una ventaja respecto HMACs a las del software de aplicación, ya que el material de claves se genera y se utiliza íntegramente desde dentro AWS KMS, sujeto a los controles de acceso que se establezcan en la clave.

Tip

Las prácticas recomendadas indican limitar el tiempo durante el cual cualquier mecanismo de firma, incluido un HMAC, es efectivo. Esto impide un ataque en el que el actor utiliza un mensaje firmado para establecer la validez repetidamente o mucho después de que se sustituya el mensaje. Las etiquetas del HMAC no incluyen una marca de hora, pero puede incluir una marca de hora en el token o mensaje para ayudarlo a detectar cuándo es hora de actualizar el HMAC.

Operaciones criptográficas admitidas

Las claves KMS HMAC solo admiten las operaciones criptográficas [GenerateMac](#) y [VerifyMac](#). No puede utilizar claves KMS HMAC para cifrar datos o firmar mensajes, ni utilizar ningún otro tipo de clave KMS en las operaciones de HMAC. Cuando utiliza la operación `GenerateMac`, proporciona un mensaje de hasta 4096 bytes, una clave KMS HMAC y el algoritmo MAC compatible con la especificación de clave HMAC, y `GenerateMac` computa la etiqueta HMAC. Para verificar una etiqueta HMAC, debe proporcionar la etiqueta HMAC y el mismo mensaje, clave KMS HMAC y algoritmo MAC que `GenerateMac` utilizó para computar la etiqueta HMAC original. La operación `VerifyMac` computa la etiqueta HMAC y verifica que es idéntica a la etiqueta HMAC suministrada. Si la entrada y las etiquetas HMAC calculadas no son idénticas, la verificación falla.

Las claves HMAC de KMS no admiten [rotación automática de claves](#) y no se puede crear una clave de KMS HMAC en un [almacén de claves personalizado](#).

Si va a crear una clave KMS para cifrar los datos de un AWS servicio, utilice una clave de cifrado simétrica. No puede utilizar una clave KMS HMAC.

Regiones

Las claves HMAC KMS son compatibles con todos Regiones de AWS los soportes. AWS KMS

Más información

- Para crear una clave KMS HMAC, consulte [Creación de una clave KMS HMAC](#).
- Para crear claves KMS HMAC de varias regiones, consulte [Multi-Region llaves en AWS KMS](#).
- Para examinar la diferencia en la política de claves predeterminada que la AWS KMS consola establece para las claves HMAC KMS, consulte [the section called “Permite a los usuarios de claves utilizar una clave KMS para las operaciones criptográficas”](#)
- Para identificar y ver claves KMS HMAC, consulte [Identificación de claves KMS HMAC](#).
- Para obtener más información sobre HMACs cómo crear tokens web JSON, consulte [Cómo proteger HMACs su interior AWS KMS](#) en el blog AWS de seguridad.
- Escuche un podcast: [Introducing HMACs for AWS Key Management Service](#) en The Official AWS Podcast.

ML-DSA entra AWS KMS

AWS Key Management Service (AWS KMS) admite el algoritmo de firma digital Module-Lattice (ML-DSA) para firmas criptográficas poscuánticas. Esta implementación sigue el [estándar 204 de los Estándares Federales de Procesamiento de Información \(FIPS\)](#) para ayudar a estar protegido contra las futuras amenazas de la computación cuántica. AWS KMS crea y protege todas las claves y operaciones de firma ML-DSA en módulos de seguridad de hardware validados por el FIPS 140-3 de nivel de seguridad 3. Para ayudar a equilibrar la seguridad con el rendimiento, el ML-DSA in AWS KMS ofrece tres niveles de seguridad distintos con diferentes especificaciones clave: ML_DSA_44, ML_DSA_65 y ML_DSA_87.

AWS KMS admite firmas de clave asimétricas RAW para mensajes de hasta 4 KB según el tipo de mensaje. Para mensajes más grandes, debe calcular externamente la μ de representación del mensaje de 64 bytes utilizada en la firma ML-DSA, tal como se define en la sección 6.2 del NIST FIPS 204. Utilice el tipo de EXTERNAL_MU mensaje de la operación de AWS KMS [firma](#) para especificar este mensaje de 64 bytes preprocesado. Las firmas generadas por la μ calculada externamente son las mismas las de RAW cuando se utiliza el mismo mensaje y la misma clave privada. Tenga en cuenta que esta firma es diferente del MLDSA “previo al hash” o del HashML-DSA definido en la sección 5.4 de NIST FIPS 204.

Para obtener más información sobre el uso de ML-DSA y el tipo de mensaje EXTERNAL_MU, consulte [Especificaciones clave del ML-DSA](#).

Para ver un ejemplo del uso de ML-DSA y el tipo de mensaje EXTERNAL_MU, consulte [Verificación fuera de línea con pares de claves ML-DSA](#).

Multi-Region llaves en AWS KMS

AWS KMS admite claves multirregionales, que son AWS KMS keys diferentes Regiones de AWS y se pueden usar indistintamente, como si tuvieras la misma clave en varias regiones. Cada conjunto de claves multirregionales relacionadas tiene el mismo material de clave y el mismo [identificador de clave](#), por lo que puede cifrar los datos en uno Región de AWS y descifrarlos en otro diferente Región de AWS sin necesidad de volver a cifrarlos ni de realizar llamadas entre regiones. AWS KMS

Como todas las claves KMS, las claves multirregionales nunca se quedan sin cifrar. AWS KMS Puede crear claves de varias regiones simétricas o asimétricas para el cifrado o la firma, y crear claves de varias regiones HMAC para generar y verificar etiquetas HMAC, y crear [claves de varias regiones con material de claves importado](#) o material de claves que AWS KMS genera. Debe administrar cada clave de varias regiones de forma independiente, incluida la creación de alias y

etiquetas, el establecimiento de sus políticas y concesiones clave, y la habilitación y deshabilitación selectivas. Puede utilizar claves de varias regiones en todas las operaciones criptográficas que puede realizar con claves de una sola región.

Multi-Region las claves son una solución flexible y potente para muchos escenarios comunes de seguridad de datos.

Recuperación ante desastres

En una arquitectura de respaldo y recuperación, las claves multirregionales le permiten procesar datos cifrados sin interrupción, incluso en caso de que se produzca una Región de AWS interrupción. Los datos mantenidos en las regiones de copia de seguridad se pueden descifrar en la región de copia de seguridad, y los datos recién cifrados en la región de copia de seguridad se pueden descifrar en la región principal cuando se restaura esa región.

Administración de datos global

Las empresas que operan en todo el mundo necesitan datos distribuidos globalmente que estén disponibles de manera consistente en Regiones de AWS. Puede crear claves de varias regiones en todas las regiones donde residen los datos y, a continuación, utilizar las claves como si fueran una clave de una sola región sin la latencia de una llamada entre regiones o el costo de volver a cifrar datos bajo una clave diferente en cada región.

Aplicaciones de firma distribuidas

Las aplicaciones que requieren capacidades de firma entre regiones pueden utilizar claves de firma asimétricas de varias regiones para generar firmas digitales idénticas de forma consistente y repetida en diferentes Regiones de AWS.

Si utiliza el encadenamiento de certificado con un único almacén de confianza global (para una entidad de certificado [CA] de raíz única y las CA intermedias regionales firmadas por la CA raíz) no necesita claves de varias regiones. Sin embargo, si el sistema no admite CA intermedias, como la firma de aplicaciones, puede usar claves de varias regiones para dar coherencia a las certificaciones regionales.

Active-active aplicaciones que abarcan varias regiones

Algunas cargas de trabajo y aplicaciones pueden abarcar varias regiones en arquitecturas activa-activa. Para estas aplicaciones, las claves de varias regiones pueden reducir la complejidad al proporcionar el mismo material clave para operaciones simultáneas de cifrado y descifrado en datos que podrían estar moviéndose a través de los límites de la región.

Puede utilizar claves de varias regiones con bibliotecas de cifrado del cliente, como la [AWS Encryption SDK](#), el [SDK de cifrado de base de datos de AWS](#) y el [Cifrado del cliente de Amazon S3](#).

La mayoría de [AWS los servicios que se integran AWS KMS](#) para el cifrado en reposo o las firmas digitales actualmente tratan las claves multirregionales como si fueran claves de una sola región. Por ejemplo, la replicación entre regiones de Amazon S3 descifra y vuelve a cifrar las claves de datos utilizadas para cifrar los datos de objetos en la clave de KMS en la región de destino, incluso cuando la clave de KMS de ambas regiones es una clave multirregional relacionada. Consulte la documentación específica del servicio para saber cómo un servicio replica los datos cifrados y si trata las claves multirregionales de forma diferente.

Multi-Region las claves no son globales. Cree una clave principal de varias regiones y, a continuación, replíquela en las regiones que seleccione dentro de una [partición de AWS](#). Luego, administre la clave de varias regiones en cada región de forma independiente. Ni crea AWS ni AWS KMS replica automáticamente claves multirregionales en ninguna región en tu nombre. [Claves administradas por AWS](#), las claves de KMS que AWS los servicios crean en su cuenta para usted, son siempre claves de una sola región.

En las regiones de China, se puede utilizar la característica clave multirregión para replicar las claves de KMS dentro de la partición de las regiones de China (aws-cn). Por ejemplo, se puede replicar una clave de la región de China (Pekín) en la región de China (Ningxia) o al revés. Al replicar una clave de una región de China a otra, aceptas usar la AWS Key Management Service de la región de destino y cumplir con todos los términos del acuerdo aplicables a la región de destino. No puede replicar una clave de las regiones de Beijing y Ningxia en una AWS región fuera de la partición de las regiones de China. De manera similar, no puede replicar una clave de una región por fuera de la partición de regiones de China en las regiones de Pekín y Ningxia.

No puede convertir una clave de región única existente en una clave de varias regiones. Este diseño garantiza que todos los datos protegidos con claves de una sola región existentes mantengan las mismas propiedades de residencia y soberanía de datos.

Para la mayoría de las necesidades de seguridad de los datos, el aislamiento regional y la tolerancia a errores de los recursos regionales hacen que las claves de una AWS KMS sola región estándar sean la solución más adecuada. Sin embargo, cuando necesite cifrar o firmar datos en aplicaciones del cliente en varias regiones, es posible que las claves de varias regiones sean la solución.

Regiones

Multi-Region las claves son compatibles con todos los soportes Regiones de AWS . AWS KMS

Precios y cuotas

Cada clave de un conjunto de claves de varias regiones relacionadas cuenta como una clave KMS para precios y cuotas. Las [cuotas de AWS KMS](#) se calculan por separado por cada región de una cuenta. El uso y la administración de las claves de varias regiones en cada Región cuenta para las cuotas de dicha Región.

Tipos de claves KMS compatibles

Puede crear los siguientes tipos de claves KMS para varias regiones:

- Claves de KMS de cifrado simétrico
- Claves de KMS asimétricas
- Claves KMS HMAC
- Claves KMS con material de claves importado

No puede crear claves de varias regiones en un almacén de claves personalizado.

Más información

- Para obtener información sobre cómo controlar el acceso a las claves KMS de varias regiones, consulte [Control del acceso a claves de varias regiones](#).
- Para crear claves KMS principales de varias regiones de cualquier tipo, consulte [Creación de claves primarias de varias regiones](#).
- Para crear claves KMS de réplica de varias regiones, consulte [Creación de claves de réplica de varias regiones](#).
- Para actualizar la región principal, consulte [Cambio de la clave principal en un conjunto de claves de varias regiones](#).
- Para identificar y ver las claves KMS de varias regiones, consulte [Identificación de claves KMS HMAC](#).
- Para conocer las consideraciones especiales a la hora de eliminar claves KMS de varias regiones, consulte [Deleting multi-Region keys](#).

Terminología y conceptos

Los siguientes términos y conceptos se utilizan con claves de varias regiones.

Multi-Region clave

Una clave de varias regiones es una de un conjunto de claves KMS con el mismo ID de clave y material de claves (y otras [propiedades compartidas](#)) en diferentes Regiones de AWS. Cada clave de varias regiones es una clave KMS que funciona completamente que se puede utilizar independientemente de sus claves de varias regiones relacionadas. Como todas las claves multirregionales relacionadas tienen el mismo identificador de clave y el mismo material de clave, son interoperables, es decir, cualquier clave multirregional relacionada Región de AWS puede descifrar el texto cifrado por cualquier otra clave multirregional relacionada.

Usted configura la propiedad de varias regiones de una clave KMS cuando la crea. No se puede cambiar esta propiedad de varias regiones en una clave existente. No se puede convertir una clave de región única en clave de varias regiones ni convertir una clave de varias regiones en una clave de región única. Para mover cargas de trabajo existentes a escenarios de varias regiones, debe volver a cifrar los datos o crear nuevas firmas con nuevas claves de varias regiones.

[Una clave multirregional puede ser simétrica o asimétrica y puede utilizar material clave o material clave importado. AWS KMS](#) No puede crear claves de varias regiones en un [almacén de claves personalizado](#).

En un conjunto de claves de varias regiones relacionadas, hay exactamente una [clave principal](#) en cualquier momento. Puede crear [claves de réplica](#) de esa clave principal en otras Regiones de AWS. También puede [actualizar la región principal](#), que cambia la clave principal a una clave de réplica y cambia una clave de réplica especificada a la clave principal. Sin embargo, solo puede mantener una clave principal o una clave de réplica en cada una. Región de AWS Todas las regiones deben estar en la misma [partición de AWS](#).

Puede tener varios conjuntos de claves de varias regiones relacionadas en la misma o diferentes Regiones de AWS. Aunque las claves de varias regiones relacionadas son interoperables, las claves de varias regiones no relacionadas no son interoperables.

Clave principal

Una clave principal multirregional es una clave de KMS que se puede replicar Regiones de AWS en otra de la misma partición. Cada conjunto de claves de varias regiones tiene una sola clave principal.

Una clave principal difiere de una clave de réplica en las siguientes formas:

- Solo se puede [replicar](#) una clave principal.

- La clave principal es el origen de [propiedades compartidas](#) de su [claves de réplica](#), incluido el material de la clave y el ID de la clave.
- Puede habilitar y desactivar la [rotación automática de claves](#) solo en una clave principal.
- Puede [programar la eliminación de una clave principal](#) en cualquier momento. Pero no AWS KMS eliminará una clave principal hasta que se eliminen todas sus claves de réplica.

Sin embargo, las claves primarias y de réplica no difieren en ninguna propiedad criptográfica. Puede utilizar una clave principal y sus claves de réplica de forma intercambiable.

No es necesario replicar una clave principal. Puede usarla como lo haría con cualquier clave KMS y replicarla cuando sea útil. Sin embargo, dado que las claves de varias regiones tienen propiedades de seguridad diferentes a las claves de una sola región, se recomienda crear una clave de varias regiones solo cuando planee replicarla.

Clave de réplica

Una clave de réplica de varias regiones es una clave KMS que tiene el mismo [ID de clave](#) y material de claves que su [clave principal](#) y las claves de réplica relacionadas, pero existe en una Región de AWS diferente.

Una clave de réplica es una clave KMS completamente funcional con su propia política de clave, concesiones, alias, etiquetas y otras propiedades. No es una copia ni un puntero a la clave principal ni a ninguna otra clave. Puede utilizar una clave de réplica incluso si su clave principal y todas las claves de réplica relacionadas están deshabilitadas. También puede convertir una clave de réplica en una clave principal y una clave principal en una clave de réplica. Una vez creada, una clave de réplica se basa en su clave principal solo para la [rotación de claves](#) y la [actualización de la región principal](#).

Las claves principales y de réplica no difieren en ninguna propiedad criptográfica. Puede utilizar una clave principal y sus claves de réplica de forma intercambiable. Los datos cifrados por una clave principal o de réplica se pueden descifrar con la misma clave o mediante cualquier clave principal o de réplica relacionada.

Replicación

Puede replicar una [clave principal](#) multirregional Región de AWS en otra diferente de la misma partición. Al hacerlo, AWS KMS crea una [clave de réplica](#) multirregional en la región especificada con el mismo [ID de clave](#) y otras [propiedades compartidas](#) que su clave principal. En el caso de

las claves de KMS con AWS_KMS origen, transporta de AWS KMS forma segura el material clave a través de los límites de la región y lo asocia a la nueva clave de réplica, todo dentro. AWS KMS En el caso de las claves KMS con EXTERNAL origen, debe importar el mismo material clave que importó a la clave de región principal para cada réplica de clave de región de forma individual.

Propiedades compartidas

Las propiedades compartidas son propiedades de una clave principal multirregional que se comparten con sus claves de réplica. AWS KMS crea las claves de réplica con los mismos valores de propiedad compartidos que los de la clave principal. A continuación, sincroniza periódicamente los valores de propiedad compartida de la clave principal con sus claves de réplica. No puede establecer estas propiedades en una clave de réplica.

Las siguientes son las propiedades compartidas de claves de varias regiones.

- [ID de clave](#): (el elemento de la Region del [ARN de clave](#) difiere).
- [Material clave](#): las claves principal y réplica de un conjunto de claves multirregionales relacionadas comparten el mismo material clave. En el caso de las claves multirregionales cuyo material clave se genera por AWS KMS (AWS_KMSorigen), transporta de AWS KMS forma segura todos los materiales clave de la principal a cada réplica cuando se crea la réplica o cuando se crea un nuevo material clave mediante rotación automática o bajo demanda. En el caso de las claves multirregionales con material clave importado (EXTERNALorigen), AWS KMS sincroniza el identificador del material clave con la clave principal, pero debe importar el material clave a cada clave de réplica de forma independiente.
- [Origen del material de claves](#)
- [Especificación de clave](#) y algoritmos de cifrado
- [Uso de claves](#)
- [Rotación automática de claves](#): solo puede habilitar y desactivar la rotación automática de claves en la clave principal. Las nuevas claves de réplica se crean con todas las versiones del material de claves compartido. Para obtener más información, consulte [Rotating multi-Region keys](#).
- [On-demand rotación](#): solo puede realizar la rotación bajo demanda en la clave principal. En el caso de claves multirregionales cuyo material clave es generado por AWS KMS (AWS_KMSorigen), AWS KMS crea réplicas de claves con todas las versiones del material clave compartido. En el caso de las claves multirregionales con material clave importado (EXTERNALorigen), AWS KMS propaga el identificador del material clave y la descripción del material clave desde la clave principal a las claves de réplica, pero no al material clave. Debe importar el material clave correcto a cada clave de réplica de forma individual. Para obtener más información, consulte [Rotating multi-Region keys](#).

También puede pensar en las designaciones primarias y de réplica de claves de varias regiones relacionadas como propiedades compartidas. Al [crear nuevas claves de réplica](#) o [actualizar la clave principal](#), AWS KMS sincroniza el cambio con todas las claves multirregionales relacionadas. Cuando se completan estos cambios, todas las claves de varias regiones relacionadas muestran su clave principal y las claves de réplica con precisión.

[Todas las demás propiedades de las claves multirregionales son propiedades independientes, incluidas la descripción de la clave, la política de claves, las concesiones, los estados clave habilitados y deshabilitados, los alias y las etiquetas.](#) Puede establecer los mismos valores para estas propiedades en todas las claves de varias regiones relacionadas, pero si cambia el valor de una propiedad independiente, AWS KMS no lo sincroniza.

Puede realizar un seguimiento de la sincronización de las propiedades compartidas de las claves de varias regiones. Busca el evento en tu AWS CloudTrail registro. [SynchronizeMultiRegionKey](#)

Consideraciones sobre seguridad para claves de varias regiones

Usa una clave AWS KMS multirregional solo cuando la necesites. Multi-Region las claves proporcionan una solución flexible y escalable para las cargas de trabajo que mueven datos cifrados entre regiones de AWS o que necesitan acceso entre regiones. Considere una clave de varias regiones si debe compartir, mover o hacer una copia de seguridad de datos protegidos entre regiones o necesita crear firmas digitales idénticas de aplicaciones que operan en regiones diferentes.

Sin embargo, el proceso de creación de una clave de varias regiones mueve el material clave a través de límites de Región de AWS dentro de AWS KMS. El texto cifrado generado por una clave de varias regiones se puede descifrar potencialmente mediante varias claves relacionadas en varias ubicaciones geográficas. Los Regionally-isolated servicios y los recursos también ofrecen beneficios significativos. Cada Región de AWS es independiente y está aislada de las demás regiones. Las regiones proporcionan tolerancia a errores, estabilidad y resistencia, y también pueden reducir la latencia. Le permiten crear recursos redundantes que sigan estando disponibles y no resulten afectados por una interrupción en otra región. Además AWS KMS, garantizan que cada texto cifrado se pueda descifrar con una sola clave.

Multi-Region las claves también plantean nuevas consideraciones de seguridad:

- Controlar el acceso y aplicar la política de seguridad de datos es más complejo con las claves de varias regiones. Debe asegurarse de que la política se auditará de forma coherente en clave en

varias regiones aisladas. Y debe usar la política para imponer límites, en lugar de depender de claves separadas.

Por ejemplo, debe establecer condiciones de política en los datos para evitar que los equipos de nómina de una Región puedan leer los datos de nómina de una Región diferente. Además, debe usar el control de acceso para evitar un escenario en el que una clave de varias regiones en una región proteja los datos de un inquilino y una clave de varias regiones relacionada en otra región proteja los datos de un inquilino diferente.

- La auditoría de claves entre regiones también es más compleja. Con las claves de varias regiones, debe examinar y conciliar las actividades de auditoría en varias regiones para obtener una comprensión completa de las actividades clave en los datos protegidos.
- La conformidad de los mandatos de residencia de datos puede ser más complejo. Con Regiones aisladas, puede garantizar la residencia de datos y la conformidad de la soberanía de datos. Las claves KMS de una región determinada solo pueden descifrar información confidencial en esa región. Los datos cifrados en una región pueden permanecer completamente protegidos e inaccesibles en cualquier otra región.

Para verificar la residencia y la soberanía de los datos con claves multirregionales, debe implementar políticas de acceso y compilar AWS CloudTrail eventos en varias regiones.

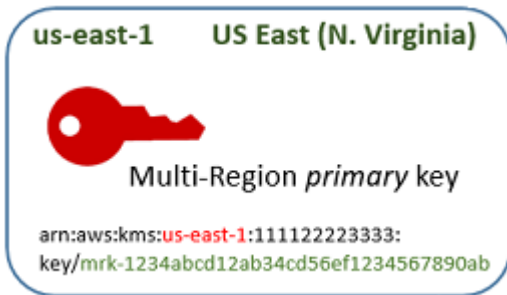
Para facilitar la administración del control de acceso de las claves multirregionales, el permiso para replicar una clave multirregional ([kms: ReplicateKey](#)) es independiente del permiso estándar para crear claves ([kms:](#)). `CreateKey` Además, AWS KMS admite varias condiciones políticas para las claves multirregionales `kms:MultiRegion`, como permitir o denegar el permiso para crear, usar o administrar claves multirregionales y `kms:ReplicaRegion` restringir las regiones en las que se puede replicar una clave multiregional. Para obtener más información, consulte [Control del acceso a claves de varias regiones](#).

Funcionamiento de las claves de varias regiones

Se empieza por crear una [clave principal multirregional](#) simétrica o asimétrica en una Región de AWS que AWS KMS sea compatible, como US East (Virginia del Norte). Usted decide si una clave es de una región o de varias regiones únicamente cuando la crea; no puede cambiar esta propiedad más adelante. Al igual que con cualquier clave KMS, debe establecer una política de clave para la clave de varias regiones, y se pueden crear concesiones y agregar alias y etiquetas para la categorización y autorización. (Estas son [propiedades independientes](#) que no están compartidas ni

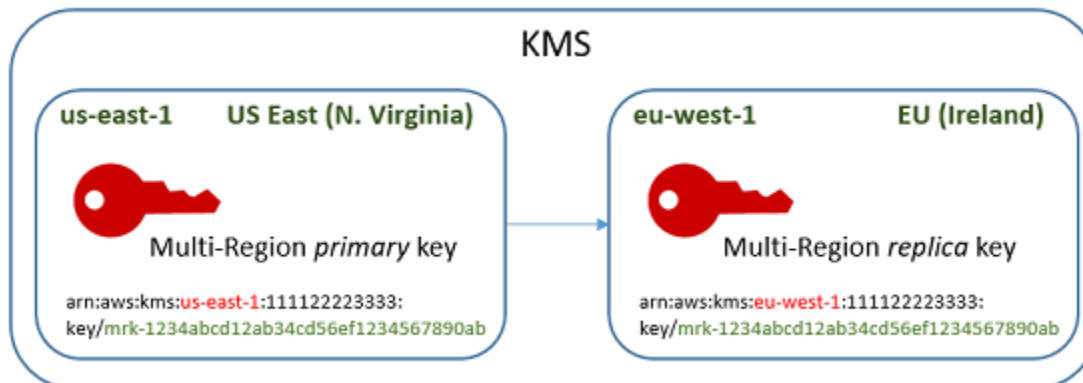
sincronizadas con otras claves). Puede utilizar la clave principal de varias regiones en operaciones criptográficas para el cifrado o la firma.

Puede [crear una clave principal multirregional](#) en la AWS KMS consola o mediante la [CreateKey](#) API con el `MultiRegion` parámetro establecido en `true`. Observe que las claves de varias regiones tienen un ID de clave distintivo que comienza con `mrk-`. Puede utilizar el prefijo `mrk-` para identificar los MRK mediante programación.



Si lo desea, puede [replicar](#) la clave principal multirregional en una o más unidades diferentes. Regiones de AWS de la misma [AWS partición](#), como Europa (Irlanda). Al hacerlo, AWS KMS crea una [clave de réplica](#) en la región especificada con el mismo ID de clave y otras [propiedades compartidas](#) que la clave principal. El resultado son dos claves de varias regiones relacionadas (una clave principal y una clave de réplica) que se pueden utilizar de forma intercambiable.

Puede [crear una clave de réplica multirregional](#) en la AWS KMS consola o mediante la [ReplicateKey](#) API.



El resultado de la [clave de réplica de varias regiones](#) es una clave KMS completamente funcional con las mismas [propiedades compartidas](#) que la clave principal. En todos los demás aspectos, es una clave KMS independiente con su propia descripción, política de clave, concesiones, alias y etiquetas. La habilitación o deshabilitación de una clave de varias regiones no tiene ningún efecto en las claves de varias regiones relacionadas. Puede utilizar las claves principal y de réplica de forma independiente en operaciones criptográficas o coordinar su uso. Por ejemplo, puede cifrar datos con

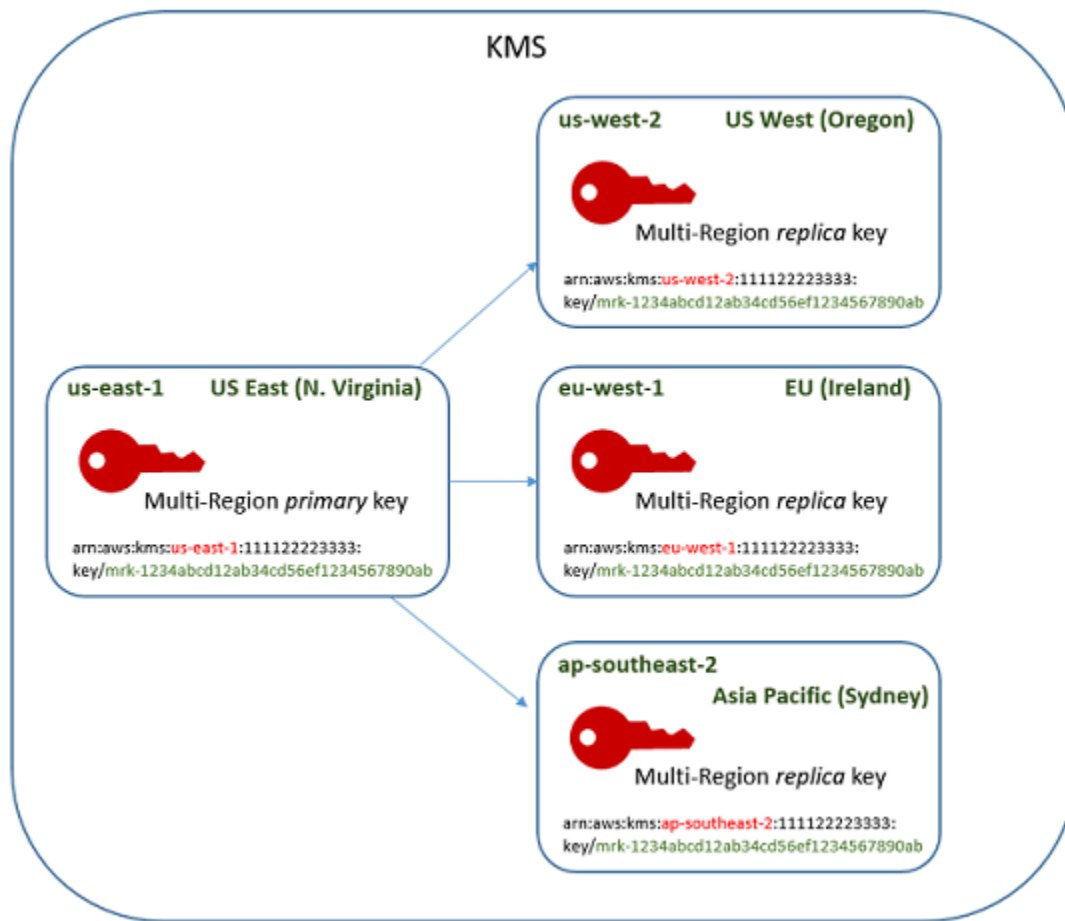
la clave principal en la región EE. UU. Este (Norte de Virginia), mover los datos a la región Europa (Irlanda) y usar la clave de réplica para descifrar los datos.

Las claves de varias regiones relacionadas tienen el mismo ID de clave. Sus ARN (nombres de recursos de Amazon) clave solo difieren en el campo Región. Por ejemplo, la clave principal de varias regiones y las claves de réplica pueden tener los siguientes ARN de clave de ejemplo. El ID de clave, que es el último elemento de la clave ARN, es idéntico. Ambas claves tienen el ID de clave distintivo de las claves de varias regiones, que comienza con `mrk-`.

```
Primary key: arn:aws:kms:us-east-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
Replica key: arn:aws:kms:eu-west-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
```

Se requiere tener el mismo ID de clave para lograr interoperabilidad. Al cifrar, AWS KMS vincula el ID de clave de la clave KMS al texto cifrado para que el texto cifrado solo se pueda descifrar con esa clave KMS o con una clave KMS con el mismo ID de clave. Esta característica también facilita el reconocimiento de las claves de varias regiones relacionadas y facilita su uso de forma intercambiable. Por ejemplo, cuando se utilizan en una aplicación, puede hacer referencia a las claves de varias regiones relacionadas por su ID de clave compartida. A continuación, si es necesario, especifique la Región o ARN para distinguirlos.

A medida que cambien sus necesidades de datos, puede replicar la clave principal Regiones de AWS en otra de la misma partición, como EE. UU. oeste (Oregón) y Asia Pacífico (Sídney). El resultado son cuatro claves de varias regiones relacionadas con el mismo material de claves y los mismos ID de claves, tal y como se muestra en el siguiente diagrama. Administra las claves de forma independiente. En el caso de las claves multirregionales con material clave importado, usted es responsable de importar el material clave a cada clave relacionada de forma individual. Puede usarlas de forma independiente o coordinada. Por ejemplo, puede cifrar datos con la clave de réplica en Asia-Pacífico (Sídney), mover los datos a EE. UU. Oeste (Oregón) y descifrarlos con la clave de réplica en EE. UU. Oeste (Oregón).



Otras consideraciones para las claves de varias regiones son las siguientes.

Sincronización de propiedades compartidas: [si una propiedad compartida de las claves multirregionales cambia, sincroniza AWS KMS automáticamente el cambio de la clave principal a todas sus claves de réplica](#). No puede solicitar ni forzar la sincronización de las propiedades compartidas. AWS KMS detecta y sincroniza todos los cambios por usted. Sin embargo, puede auditar la sincronización utilizando el [SynchronizeMultiRegionKey](#) evento en los CloudTrail registros.

Por ejemplo, si habilitas la rotación automática de claves en una clave principal simétrica multirregional con `AWS_KMS` origen, AWS KMS copia esa configuración en todas sus réplicas de claves. Cuando se gira el material clave, la rotación se sincroniza entre todas las claves de varias regiones relacionadas, de modo que sigan teniendo el mismo material clave actual y acceso a todas las versiones anteriores del material clave. Si crea una nueva clave de réplica, tiene el mismo material de clave actual de todas las claves de varias regiones relacionadas y acceso a todas las versiones anteriores del material claves. Para obtener más información, consulte [Rotating multi-Region keys](#).

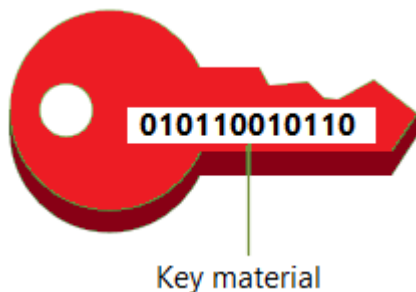
Cambio de la clave principal: cada conjunto de claves de varias regiones debe tener exactamente una clave principal. La [clave principal](#) es la única clave que se puede replicar. También es el origen de las propiedades compartidas de sus claves de réplica. Sin embargo, puede cambiar la clave principal a una réplica y promover una de las claves de réplica a primaria. Puede hacerlo para eliminar una clave principal de varias regiones de una región determinada o ubicar la clave principal en una región más cercana a los administradores del proyecto. Para obtener más información, consulte [Cambio de la clave principal en un conjunto de claves de varias regiones](#).

Eliminar claves multirregionales: al igual que todas las claves de KMS, debe programar la eliminación de claves multirregionales antes de eliminarlas. AWS KMS Mientras la clave está pendiente de eliminación, no puede utilizarla en ninguna operación criptográfica. Sin embargo, no AWS KMS eliminará una clave principal multirregional hasta que se eliminen todas sus claves de réplica. Para obtener información, consulte [Deleting multi-Region keys](#).

Importación de material clave para AWS KMS llaves

Puede crear una AWS KMS keys (clave KMS) con el material clave que suministre.

Una clave KMS es una representación lógica de una clave de datos. Los metadatos de una clave KMS incluyen el ID del material de claves que se utiliza para llevar a cabo operaciones criptográficas. Al [crear una clave de KMS](#), de forma predeterminada, se AWS KMS genera el material clave para esa clave de KMS. Pero puede crear una clave KMS sin material de claves y, a continuación, importar su propio material de claves en esa clave KMS, una característica que se conoce a menudo como "bring your own key" (BYOK) ("utilice su propia clave").



Note

AWS KMS no admite el descifrado de ningún AWS KMS texto cifrado mediante una clave KMS de cifrado simétrico fuera de ella AWS KMS, incluso si el texto cifrado se cifró con una clave KMS con material clave importado. AWS KMS no publica el formato de texto cifrado que requiere esta tarea y el formato puede cambiar sin previo aviso.

Cuando utiliza material clave importado, sigue siendo responsable del material clave y permite AWS KMS utilizar una copia del mismo. Puede hacerlo por uno o varios de los motivos siguientes:

- Para demostrar que ha generado el material de claves con un origen de entropía que cumple sus requisitos.
- Para utilizar el material clave de su propia infraestructura con AWS los servicios y AWS KMS para gestionar el ciclo de vida de ese material clave interno AWS.
- Para utilizar claves existentes y bien establecidas, como las claves para la firma de código AWS KMS, la firma de certificados de PKI y las aplicaciones con certificados anclados
- Establecer una fecha de caducidad para el material clave AWS y [eliminarlo manualmente](#), pero también hacer que vuelva a estar disponible en el futuro. Por el contrario, [programar la eliminación de claves](#) requiere un periodo de espera de 7 a 30 días, transcurrido el cual no puede recuperar la clave KMS eliminada.
- Ser propietario de la copia original del material clave y mantenerla fuera de ella AWS para garantizar una mayor durabilidad y recuperación ante desastres durante todo el ciclo de vida del material clave.
- En el caso de las claves asimétricas y las claves HMAC, la importación crea claves compatibles e interoperables que funcionan dentro y fuera de ellas. AWS

Tipos de claves KMS compatibles

AWS KMS admite material clave importado para los siguientes tipos de claves KMS. No puede importar material de claves a una clave de KMS de [almacenes de claves personalizados](#).

- [Claves de KMS de cifrado simétrico](#)
- [Claves de KMS asimétricas \(excepto claves de ML-DSA\)](#)
- [Claves KMS HMAC](#)
- [Claves de varias regiones](#) de todos los tipos compatibles.

Regiones

El material clave importado es compatible con todos Regiones de AWS los AWS KMS soportes.

En las regiones de China, los requisitos de material de claves en relación con claves KMS de cifrado simétrico difieren de los de otras regiones. Para obtener más información, consulte [Paso 3: Cifrar el material de claves](#).

Más información

- Para crear claves KMS con material de claves importado, consulte [Creación de una clave KMS con material de claves importado](#).
- Para crear una alarma de que le notifique cuando el material de claves importado en una clave KMS se acerca a su fecha de caducidad, consulte [Cree una CloudWatch alarma de caducidad del material clave importado](#).
- Para volver a importar el material de claves en una clave KMS, consulte [Volver a importar material de claves](#).
- Para importar material de claves nuevo a una clave de KMS para su rotación bajo demanda, consulte [Importación de nuevo material de claves](#) y [Realización de la rotación de claves bajo demanda](#).
- Para identificar y ver claves KMS con material de claves importado, consulte [Identificación de claves KMS con material de claves importado](#).
- Para obtener información acerca de las consideraciones especiales a la hora de eliminar claves KMS con material de claves importado, consulte [Deleting KMS keys with imported key material](#).

Consideraciones especiales en relación con el material de claves importado

Antes de decidir importar material clave a AWS KMS, debe comprender las siguientes características del material clave importado.

Generar el material de clave

Es su responsabilidad generar el material de claves con una fuente de aleatoriedad que cumpla sus requisitos de seguridad.

Usted es responsable de la disponibilidad y durabilidad

AWS KMS está diseñado para mantener una alta disponibilidad del material clave importado. Sin embargo, AWS KMS no mantiene la durabilidad del material clave importado al mismo nivel que el material clave que se genera en AWS KMS. Para obtener más información, consulte [Protección del material de claves importado](#).

Puede eliminar el material de clave

Puede [eliminar el material de claves importado](#) de una clave de KMS, al inutilizar inmediatamente la clave de KMS. Además, al importar material de claves en una clave de KMS, puede determinar si la clave vence y [establecer su fecha de vencimiento](#). Cuando llegue el momento de caducidad,

AWS KMS [elimina el material clave](#). Sin material de claves, la clave KMS no puede utilizarse en ninguna operación criptográfica. Para restaurar la clave, debe volver a importar el mismo material en la clave.

No puede cambiar el material de la clave para las claves asimétricas y HMAC

Al importar el material de claves en una clave de KMS, la clave de KMS se asocia de forma permanente a dicho material de claves. Puede [volver a importar el mismo material de claves](#), pero no puede importar material de claves diferente en esa clave KMS. Además, no puede [habilitar la rotación automática de claves](#) para una clave KMS con material de claves importado. Sin embargo, puede [rotar manualmente una clave KMS](#) con material de claves importado.

Puede realizar la rotación bajo demanda en claves de cifrado simétricas

Las claves de cifrado simétricas con material clave importado admiten la rotación bajo demanda. Puede [importar varios materiales de claves](#) a estas claves y utilizar la [rotación bajo demanda](#) para actualizar el material de claves actual. El material de claves actual se utiliza tanto para el cifrado como para el descifrado, pero otros materiales de claves (no actuales) solo se pueden utilizar para el descifrado.

No puede cambiar el origen del material de claves

Las claves KMS diseñadas para el material de claves importado tienen un valor de [origen](#) de EXTERNAL que no se puede cambiar. No puede convertir una clave KMS para material clave importado para utilizar material clave de ninguna otra fuente, ni siquiera. AWS KMS Del mismo modo, no puede convertir una clave KMS con material AWS KMS clave en una diseñada para material clave importado.

No puede exportar el material de claves

No puede exportar ningún material clave que haya importado. AWS KMS no puede devolverle el material clave importado de ninguna forma. Debe conservar una copia del material clave importado fuera de AWS, preferiblemente, en un administrador de claves, como un módulo de seguridad de hardware (HSM), de modo que pueda volver a importar el material clave si lo elimina o si caduca.

Puede crear claves de varias regiones con material de claves importado

Las múltiples regiones con material de claves importado tienen las características de las claves de KMS con material de claves importado y pueden interoperar entre Regiones de AWS. Para crear una clave de varias regiones con material de claves importado, debe importar el mismo material de claves en la clave de KMS principal y en cada clave de réplica. Para obtener más información sobre la importación de materiales clave para claves multirregionales, consulte.

Las claves asimétricas y las claves HMAC son portátiles e interoperables

Puede utilizar el material de clave asimétrico y el material de clave HMAC de forma externa AWS para interoperar con AWS KMS llaves del mismo material de clave importado.

A diferencia del texto cifrado AWS KMS simétrico, que está inextricablemente vinculado a la clave KMS utilizada en el algoritmo, AWS KMS utiliza formatos HMAC estándar y asimétricos para el cifrado, la firma y la generación de MAC. Como resultado, las claves son portátiles y admiten los escenarios tradicionales de claves de depósito de garantía.

Si su clave KMS tiene material clave importado, puede usar el material clave importado fuera de él para realizar las siguientes operaciones. AWS

- Claves HMAC: puede verificar una etiqueta HMAC generada por la clave HMAC de KMS con material de claves importado. También puede usar la clave HMAC KMS con el material clave importado para verificar una etiqueta HMAC generada por el material clave externo. AWS
- Claves de cifrado asimétricas: puede utilizar su clave de cifrado asimétrica privada AWS para descifrar un texto cifrado mediante la clave KMS con la clave pública correspondiente. También puedes usar tu clave KMS asimétrica para descifrar un texto cifrado asimétrico que se haya generado fuera de. AWS
- Claves de firma asimétrica: puedes usar tu clave KMS de firma asimétrica con material clave importado para verificar las firmas digitales generadas por tu clave de firma privada fuera de. AWS También puedes usar tu clave de firma pública asimétrica fuera de ella AWS para verificar las firmas generadas por tu clave KMS asimétrica.
- Claves de acuerdo de clave asimétrica: puede usar la clave KMS de acuerdo de clave asimétrica con material de claves importado para obtener secretos compartidos con un homólogo fuera de AWS.

Si importa el mismo material de claves en claves de KMS diferentes de la misma Región de AWS, esas claves son también interoperables. Para crear claves KMS interoperables en diferentes regiones Regiones de AWS, cree una clave multirregional con material clave importado.

Claves privadas RSA

- AWS KMS requiere que las claves privadas de RSA importadas tengan factores primos que cumplan con la prueba descrita en la sección A. 1.3 de la [FIPS 186-5](#). Es posible que otros programas o dispositivos utilicen algoritmos diferentes para validar estos factores principales de las claves privadas de RSA. En raras ocasiones, es posible que las claves validadas con otros algoritmos no sean aceptadas por AWS KMS.

Las claves de cifrado simétricas no son portátiles ni interoperables

Los textos cifrados simétricos que se AWS KMS producen no son portátiles ni interoperables. AWS KMS no publica el formato de texto cifrado simétrico que requiere la portabilidad y el formato puede cambiar sin previo aviso.

- AWS KMS no puede descifrar los textos cifrados simétricos que no estén cifrados AWS, incluso si utiliza material clave importado.
- AWS KMS no admite el descifrado de ningún texto cifrado AWS KMS simétrico que no sea AWS KMS, incluso si el texto cifrado se ha cifrado con una clave KMS con material clave importado.
- Las claves de KMS con el mismo material de claves importado no son interoperables. El texto cifrado simétrico que AWS KMS genera el texto cifrado específico de cada clave KMS. Este formato de texto cifrado garantiza que solo la clave de KMS que cifró los datos pueda descifrarlos.

Además, no puede utilizar ninguna AWS herramienta, como el [cifrado del lado del cliente AWS Encryption SDKo Amazon S3](#), para descifrar AWS KMS textos cifrados simétricos.

Por lo tanto, no puede utilizar claves con material clave importado para respaldar acuerdos de custodia de claves, en los que un tercero autorizado con acceso condicional al material clave puede descifrar determinados textos cifrados fuera de él. AWS KMS Para admitir el depósito de claves, utilice [AWS Encryption SDK](#) para cifrar su mensaje bajo una clave que es independiente de AWS KMS.

Protección del material de claves importado

El material de claves que importa está protegido en tránsito y en reposo. Antes de importar el material clave, cifra (o «envuelve») el material clave con la clave pública de un par de claves RSA generado en módulos de seguridad de AWS KMS hardware (HSMs) validados según el programa de validación de módulos [criptográficos FIPS 140-3](#). Puede cifrar el material de claves directamente con la clave pública que envuelve o cifrar el material de claves con una clave simétrica AES y, a continuación, cifrar la clave simétrica AES con la clave pública RSA.

Al recibirlo, AWS KMS descifra el material de claves con la clave privada correspondiente en un AWS KMS HSM y lo vuelve a cifrar con una clave simétrica AES que solo existe en la memoria volátil del HSM. El material de claves nunca sale del HSM en texto sin formato. Solo se descifra mientras está en uso y solo dentro. AWS KMS HSMs

El uso de la clave de KMS con el material de claves importado viene determinado únicamente por las [políticas de control de acceso](#) que se establezcan en la clave de KMS. Además, puede usar [alias](#) y [etiquetas](#) para identificar y [controlar el acceso](#) a la clave de KMS. Puede [habilitar y deshabilitar](#) la clave, [verla](#) y [supervisarla](#) mediante servicios como AWS CloudTrail.

Sin embargo, usted conserva la única copia de seguridad de su material de claves. A cambio de esta medida de control adicional, usted es responsable de la durabilidad y la disponibilidad general del material clave importado. AWS KMS está diseñado para mantener una alta disponibilidad del material clave importado. Sin embargo, AWS KMS no mantiene la durabilidad del material clave importado al mismo nivel que el material clave que se AWS KMS genera.

Esta diferencia relativa a la durabilidad es importante en los casos siguientes:

- Al [establecer una fecha de caducidad](#) para el material clave importado, AWS KMS elimina el material clave una vez que caduque. AWS KMS no elimina la clave KMS ni sus metadatos. Puedes [crear una CloudWatch alarma de Amazon](#) que te notifique cuando el material clave importado se acerca a su fecha de caducidad.

No puede eliminar el material clave que se AWS KMS genera para una clave de KMS ni puede configurar el material AWS KMS clave para que caduque.

- Al [eliminar manualmente el material clave importado](#), AWS KMS elimina el material clave pero no elimina la clave KMS ni sus metadatos. Por el contrario, [programar la eliminación de claves](#) requiere un período de espera de 7 a 30 días, tras el cual se eliminan AWS KMS permanentemente la clave de KMS, sus metadatos y su material clave.
- En el improbable caso de que se produzcan algunos fallos en toda la región AWS KMS (por ejemplo, una pérdida total de energía), AWS KMS no podrá restaurar automáticamente el material clave importado. Sin embargo, AWS KMS puede restaurar la clave KMS y sus metadatos.

Debe conservar una copia del material clave importado fuera del AWS sistema que controle. Se recomienda almacenar una copia exportable del material de claves importado en un sistema de administración de claves, como un HSM. Como práctica recomendada, debe almacenar una referencia al ARN de la clave de KMS y al ID del material de claves generado por AWS KMS junto con la copia exportable del material de claves. Si el material de claves importado se elimina o vence, la clave de KMS asociada quedará inutilizable hasta que vuelva a importar el mismo material de claves. Si el material de claves importado se pierde de forma permanente, cualquier texto cifrado con la clave de KMS será irrecuperable.

 Important

Las claves de cifrado simétricas pueden tener varios materiales clave asociados. La clave KMS completa queda inutilizable en cuanto se elimina uno de esos materiales clave o si alguno de esos materiales clave caduca (a menos que el material clave eliminado o que caduque). `PENDING_ROTATION` `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` Debe volver a importar cualquier material de claves caducado o eliminado asociado a dicha clave antes de que la clave pueda utilizarse para operaciones criptográficas.

Claves KMS en un almacén de claves de CloudHSM

Puede crear, ver, administrar, usar y programar su eliminación AWS KMS keys en un almacén de AWS CloudHSM claves. Los procedimientos son muy similares a los que usaría para otras claves de KMS. La única diferencia es que se especifica un almacén de AWS CloudHSM claves al crear la clave KMS. A continuación, AWS KMS crea material de claves no extraíble para la clave de KMS del AWS CloudHSM clúster que está asociado al almacén de AWS CloudHSM claves. Cuando se utiliza una clave KMS en un almacén de AWS CloudHSM claves, las [operaciones criptográficas](#) se realizan HSMS en el clúster.

Características admitidas

Además de los procedimientos descritos en esta sección, puede hacer lo siguiente con las claves KMS de un AWS CloudHSM almacén de claves:

- Utilice políticas de claves, políticas de IAM y concesiones para [autorizar el acceso](#) a las claves de KMS.
- [Habilite y deshabilite](#) las claves de KMS.
- Asigne [etiquetas](#), cree [alias](#) y utilice el control de acceso basado en atributos (ABAC) para autorizar el acceso a las claves de KMS.
- Utilice las claves KMS para realizar las siguientes operaciones criptográficas:
 - [Encrypt](#)
 - [Decrypt](#)
 - [GenerateDataKey](#)
 - [GenerateDataKeyWithoutPlaintext](#)
 - [ReEncrypt](#)

Las operaciones que generan pares de claves de datos asimétricos no [GenerateDataKeyPairWithoutPlaintext](#) se admiten en los almacenes de claves personalizados. [GenerateDataKeyPair](#)

- Utilice las claves de KMS con [servicios de AWS que se integran con AWS KMS](#) y que admiten las claves administradas por el cliente.
- Realice un seguimiento del uso de sus claves de KMS en [AWS CloudTrail los registros](#) y en las [herramientas CloudWatch de supervisión de Amazon](#).

Características no admitidas

- AWS CloudHSM Los almacenes de claves solo admiten claves KMS de cifrado simétrico. No puede crear claves HMAC KMS, claves KMS asimétricas ni pares de claves de datos asimétricas en un almacén de claves. AWS CloudHSM
- No puede [importar material clave](#) a una clave KMS de un almacén de claves. AWS CloudHSM AWS KMS genera el material clave para la clave KMS del AWS CloudHSM clúster.
- No puede habilitar ni deshabilitar la [rotación automática](#) del material clave de una clave KMS en un almacén de AWS CloudHSM claves.

Uso de claves KMS en un almacén de AWS CloudHSM claves

Cuando utilice la clave de KMS en una solicitud, identifique la clave de KMS por su ID o alias; no es necesario que especifique el almacén de AWS CloudHSM claves o el AWS CloudHSM clúster. La respuesta incluye los mismos campos que se devuelven para cualquier clave KMS de cifrado simétrica.

Sin embargo, cuando se utiliza una clave KMS en un almacén de AWS CloudHSM claves, la operación criptográfica se realiza íntegramente dentro del AWS CloudHSM clúster asociado al almacén de AWS CloudHSM claves. La operación utiliza el material de claves del clúster asociado con la clave KMS elegida.

Para ello, se deben cumplir las siguientes condiciones.

- El [estado de clave](#) de la clave KMS debe ser Enabled. Para encontrar el estado de la clave, utilice el campo Estado de la [AWS KMS consola](#) o el KeyState campo de la [DescribeKey](#) respuesta.
- El almacén de AWS CloudHSM claves debe estar conectado a su AWS CloudHSM clúster. Su estado en la [AWS KMS consola](#) o ConnectionState en la [DescribeCustomKeyStores](#) respuesta debe ser CONNECTED.

- El AWS CloudHSM clúster asociado al almacén de claves personalizado debe contener al menos un HSM activo. Para encontrar el número de activos HSMs en el clúster, utilice la [AWS KMS consola](#), la AWS CloudHSM consola o la [DescribeClusters](#) operación.
- El AWS CloudHSM clúster debe contener el material clave de la clave KMS. Si se ha eliminado el material de claves del clúster, o se ha creado un HSM a partir de una copia de seguridad que no incluía el material de claves, la operación criptográfica dará error.

Si no se cumplen estas condiciones, se produce un error en la operación criptográfica y se AWS KMS devuelve una `KMSInvalidStateException` excepción. Por lo general, solo tiene que [volver a conectar el almacén de AWS CloudHSM claves](#). Para obtener ayuda adicional, consulte [¿Cómo arreglar una clave KMS que produce error?](#).

Cuando utilice las claves de KMS en un almacén de AWS CloudHSM claves, tenga en cuenta que las claves de KMS de cada almacén de AWS CloudHSM claves comparten una [cuota de solicitud de almacén de claves personalizada](#) para las operaciones criptográficas. Si superas la cuota, AWS KMS devuelve un `ThrottlingException`. Si el AWS CloudHSM clúster asociado al almacén de AWS CloudHSM claves procesa numerosos comandos, incluidos los que no están relacionados con el almacén de AWS CloudHSM claves, es posible que obtenga `ThrottlingException` a una velocidad aún menor. Si recibe una excepción `ThrottlingException` para cualquier solicitud, baje la velocidad de solicitud e intente ejecutar los comandos de nuevo. Para obtener más información sobre la cuota de solicitudes del almacén de claves personalizado, consulte [Cuotas de solicitudes del almacén de claves personalizado](#).

Más información

- Para obtener más información sobre los almacenes de AWS CloudHSM claves, consulte [AWS CloudHSM tiendas clave](#).
- Para crear claves de KMS en un almacén de AWS CloudHSM claves, consulte [Crear una clave KMS en un almacén de AWS CloudHSM claves](#).
- Para identificar y ver las claves de KMS en un almacén de AWS CloudHSM claves, consulte [Identifique las claves de KMS en AWS CloudHSM almacenes de claves](#).
- Para encontrar las claves y el material clave de KMS en un almacén de AWS CloudHSM claves, consulte [Encuentre las claves y el material clave de KMS en un almacén de AWS CloudHSM claves](#).
- Para obtener más información sobre las consideraciones especiales a la hora de eliminar claves de KMS de un almacén de AWS CloudHSM claves, consulte [Eliminar claves de KMS de un almacén de AWS CloudHSM claves](#).

Claves KMS en un almacén de claves externo

Para crear, ver, administrar, usar y programar la eliminación de las claves de KMS en un almacén de claves externo, utilice procedimientos que son muy similares a los que usa para otras claves de KMS. Sin embargo, cuando crea una clave de KMS en un almacén de claves externo, especifica un [almacén de claves externo](#) y una [clave externa](#). Cuando usa una clave de KMS en un almacén de claves externo, el administrador de claves externo realiza [las operaciones de cifrado y descifrado](#) mediante la clave externa especificada.

AWS KMS no puede crear, ver, actualizar ni eliminar ninguna clave criptográfica en su administrador de claves externo. AWS KMS nunca accede directamente a su administrador de claves externo ni a ninguna clave externa. Todas las solicitudes de operaciones criptográficas están mediadas por su [proxy del almacén de claves externo](#). Para usar una clave de KMS en un almacén de claves externo, el almacén de claves externo que aloja la clave de KMS debe estar [conectado](#) a su proxy del almacén de claves externo.

Características admitidas

Además de los procedimientos tratados en esta sección, puede hacer lo siguiente con las claves de KMS en un almacén de claves personalizado:

- Utilice [políticas de claves](#), [políticas de IAM](#) y [concesiones](#) para controlar el acceso a las claves de KMS.
- [Habilite y deshabilite](#) las claves de KMS. Estas acciones no afectan a la clave externa del administrador de claves externo.
- Asigne [etiquetas](#), cree [alias](#) y utilice el [control de acceso basado en atributos](#) (ABAC) para autorizar el acceso a las claves de KMS.
- Utilice las claves KMS para realizar las siguientes operaciones criptográficas:
 - [Encrypt](#)
 - [Decrypt](#)
 - [GenerateDataKey](#)
 - [GenerateDataKeyWithoutPlaintext](#)
 - [ReEncrypt](#)

Las operaciones que generan pares de claves de datos asimétricos no se admiten en los almacenes de claves personalizados.

[GenerateDataKeyPairGenerateDataKeyPairWithoutPlaintext](#)

- Utilice las claves de KMS con [Servicios de AWS que se integran con AWS KMS](#) y que admiten las [claves administradas por el cliente](#).

Características no admitidas

- Los almacenes de claves externos solo admiten [claves de KMS de cifrado simétrico](#). No puede crear claves de KMS HMAC ni claves de KMS asimétricas en un almacén de claves externo.
- [GenerateDataKeyPair](#) y [GenerateDataKeyPairWithoutPlaintext](#) no se admiten en las claves de KMS de un almacén de claves externo.
- No puede usar una [AWS::KMS::Key CloudFormation plantilla](#) para crear un almacén de claves externo ni una clave KMS en un almacén de claves externo.
- [Las claves multirregionales](#) no se admiten en un almacén de claves externo.
- Las claves de KMS con [material de clave importado](#) no se admiten en un almacén de claves externo.
- [La rotación automática de claves](#) no es compatible con las claves de KMS en un almacén de claves externo.

Uso de claves KMS en un almacén de claves externo

Cuando utilice su clave de KMS en una solicitud, identifique la clave de KMS por su [ID de clave, ARN de clave, alias o ARN de alias](#). No tiene que especificar el almacén de claves externo. La respuesta incluye los mismos campos que se devuelven para cualquier clave KMS de cifrado simétrica. Sin embargo, cuando utiliza una clave de KMS en un almacén de claves externo, el administrador de claves externo realiza las operaciones de cifrado y descifrado mediante la clave externa especificada.

[Para garantizar que el texto cifrado con una clave KMS en un almacén de claves externo sea al menos tan seguro como cualquier texto cifrado con una clave KMS estándar, AWS KMS utiliza el doble cifrado.](#) Los datos se cifran primero con material clave. AWS KMS A continuación, su administrador de claves externo lo cifra utilizando la clave externa de la clave de KMS. Para descifrar el texto cifrado con doble cifrado, el administrador de claves externo descifra primero el texto cifrado mediante la clave externa de la clave de KMS. Luego se descifra AWS KMS utilizando el material AWS KMS clave para la clave KMS.

Para ello, se deben cumplir las siguientes condiciones.

- El [estado de clave](#) de la clave KMS debe ser Enabled. Para encontrar el estado de la clave, consulte el campo Estado de las claves administradas por el cliente, la [AWS KMS consola](#) o el KeyState campo de la [DescribeKey](#) respuesta.

- El almacén de claves externo que aloja la clave de KMS debe estar conectado a su [proxy del almacén de claves externo](#), es decir, el [estado de conexión](#) del almacén de claves externo debe ser CONNECTED.

Puede ver el estado de la conexión en la página de almacenes de claves externos de la AWS KMS consola o en la [DescribeCustomKeyStores](#) respuesta. El estado de la conexión del almacén de claves externo también se muestra en la página de detalles de la clave de KMS en la consola de AWS KMS. En la página de detalles, elija la pestaña Cryptographic configuration (Configuración criptográfica) y consulte el campo Connection state (Estado de la conexión) en la sección Custom key store (Almacén de claves personalizado).

Si el estado de la conexión es DISCONNECTED, primero debe conectarlo. Si el estado de la conexión es FAILED, debe resolver el problema, desconectar el almacén de claves externo y, a continuación, conectarlo. Para obtener instrucciones, consulte [Conexión y desconexión de almacenes de claves externos](#).

- El proxy del almacén de claves externo debe poder encontrar la clave externa.
- La clave externa debe estar habilitada y debe realizar el cifrado y el descifrado.

El estado de la clave externa es independiente y no se ve afectado por los cambios en el [estado de la clave de KMS](#), incluida la habilitación y deshabilitación de la clave de KMS. Del mismo modo, deshabilitar o eliminar la clave externa no cambia el estado de la clave de KMS, pero las operaciones criptográficas que utilicen la clave de KMS asociada fallarán.

Si no se cumplen estas condiciones, se produce un error en la operación criptográfica y se AWS KMS devuelve una `KMSInvalidStateException` excepción. Puede que tenga que [volver a conectar el almacén de claves externo](#) o utilizar las herramientas del administrador de claves externo para reconfigurar o reparar la clave externa. Para obtener ayuda adicional, consulte [the section called “Solución de problemas de almacenes de claves externos”](#).

Cuando utilice claves de KMS en un almacén de claves externo, tenga en cuenta que las claves de KMS de cada almacén de claves externo comparten una [cuota de solicitudes del almacén de claves personalizado](#) para operaciones criptográficas. Si supera la cuota, AWS KMS devuelve un `ThrottlingException`. Para obtener más información sobre la cuota de solicitudes del almacén de claves personalizado, consulte [Cuotas de solicitudes del almacén de claves personalizado](#).

Más información

- Para obtener más información sobre los almacenes de claves externos, consulte [Almacenes de claves externos](#).

- Para obtener más información sobre el material de claves de los almacenes de claves externos, consulte [Clave externa](#).
- Para crear claves KMS en un almacén de claves externo, consulte [Creación de una clave KMS en un almacén de claves externo](#).
- Para identificar y ver las claves KMS de un almacén de claves externo, consulte [Identificación de claves KMS en almacenes de claves externos](#).
- Para conocer las consideraciones especiales a la hora de eliminar claves KMS de un almacén de claves externo, consulte [Eliminación de claves KMS de un almacén de claves externo](#).

AWS KMS elementos básicos de criptografía

AWS KMS utiliza algoritmos criptográficos configurables para que el sistema pueda migrar rápidamente de un algoritmo o modo aprobado a otro. El conjunto inicial predeterminado de algoritmos criptográficos se ha seleccionado de algoritmos de los Estándares Federales de Procesamiento de la Información (aprobado por FIPS) para sus propiedades de seguridad y rendimiento.

Para obtener más información sobre los algoritmos criptográficos preferidos y aceptables, consulte [Algoritmos criptográficos compatibles](#).

Entropía y generación de números aleatorios

AWS KMS la generación de claves se realiza en. AWS KMS HSMS HSMS Implementan un generador híbrido de números aleatorios que utiliza el [NIST SP800-90A Deterministic Random Bit Generator \(DRBG\) CTR_DRBG using AES-256](#). Se encuentra sincronizado con un generador de bits aleatorio no determinista con 384 bits de entropía y actualizado con entropía adicional para proporcionar resistencia a la predicción en cada llamada de material criptográfico.

Operaciones de clave simétrica (solo cifrado)

Todos los comandos de cifrado con clave simétrica que se utilizan utilizan [estándares de cifrado avanzados \(AES\)](#) y, en el [modo contador de Galois \(GCM\)](#), utilizan claves de 256 bits. HSMS Las llamadas análogas para descifrar utilizan la función inversa.

AES-GCM es un esquema de cifrado autenticado. Además de cifrar texto sin formato a fin de producir texto cifrado, calcula una etiqueta de autenticación sobre el texto cifrado y los datos

adicionales para los que se requiere autenticación (datos autenticados adicionales o AAD). La etiqueta de autenticación ayuda a garantizar que los datos provengan de la supuesta fuente y que el texto cifrado y los AAD no se hayan modificado.

Con frecuencia, AWS omite la inclusión del AAD en nuestras descripciones, especialmente cuando se refiere al cifrado de claves de datos. En estos casos, el texto circundante implica que la estructura que se cifrará se divide entre el texto sin formato que se cifrará y los AAD de texto sin cifrar que se protegerán.

AWS KMS ofrece una opción para importar el material clave a un, en AWS KMS key lugar de confiar en él AWS KMS para generar el material clave. Este material clave importado se puede cifrar mediante el [RSAES-OAEP](#) para proteger la clave durante su transporte al HSM. AWS KMS Los pares de claves RSA se generan en el. AWS KMS HSMs El material clave importado se descifra en un AWS KMS HSM y se vuelve a cifrar con el AES-GCM antes de que el servicio lo almacene.

Operaciones de clave asimétrica (cifrado, firma digital y verificación de firmas)

AWS KMS admite el uso de operaciones de clave asimétricas para las operaciones de cifrado, firma digital y acuerdo de claves. Las operaciones de clave asimétrica se basan en un par de claves privadas y públicas relacionadas de forma matemática que puede utilizar para el cifrado y descifrado, la firma y verificación de la firma o la obtención de secretos compartidos. La clave privada nunca sale de AWS KMS sin cifrar. Puedes usar la clave pública interna de AWS KMS llamando a las operaciones de la AWS KMS API, o descargar la clave pública y usarla fuera de AWS KMS.

AWS KMS admite los siguientes cifrados asimétricos.

- RSA-OAEP (para el cifrado) y RSA-PSS y RSA-PKCS-#1-v1_5 (para la firma y verificación): es compatible con longitudes de clave de RSA (en bits): 2048, 3072 y 4096 para diferentes requisitos de seguridad.
- Curva elíptica (ECC): se utiliza para firmar y verificar o para obtener secretos compartidos, pero no para ambas cosas. Soporta curvas ECC: NIST P256, P384, P521, SECP 256k1, Ed25519.
- ML-DSA: se utiliza para la firma y la verificación. Las especificaciones de clave del ML-DSA compatibles son: ML_DSA_44, ML_DSA_65 y ML_DSA_87.
- SM2 (Solo en las regiones de China): se usa para cifrar y descifrar, firmar y verificar o obtener secretos compartidos, pero debes elegir un uso de clave. Soporta SM2 PKE para el cifrado y SM2 DSA para la firma.

Funciones de derivación de claves

Se utiliza una función de derivación de claves para obtener claves adicionales a partir de un secreto o clave inicial. AWS KMS utiliza una función de derivación de claves (KDF) para obtener claves por llamada para cada cifrado utilizado en un. AWS KMS key [Todas las operaciones de KDF utilizan el KDF en modo contador utilizando HMAC \[\] con \[\]. FIPS197 SHA256 FIPS180](#) La clave derivada de 256 bits se utiliza con AES-GCM para cifrar o descifrar datos y claves del cliente.

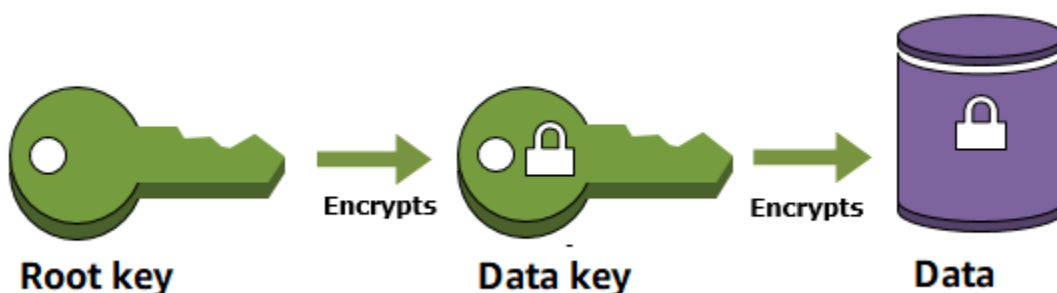
AWS KMS uso interno de firmas digitales

Las firmas digitales también se utilizan para autenticar comandos y comunicaciones entre entidades de AWS KMS . Todas las entidades de servicio tienen un par de claves de algoritmo de firma digital de curva elíptica (ECDSA). Emplean ECDSA como se define en el [Uso de algoritmos de criptografía de curva elíptica \(ECC\) en la sintaxis de mensajes criptográficos \(CMS\)](#) y X9.62-2005: Criptografía de clave pública para el sector de servicios financieros: el Algoritmo de firma digital de curva elíptica (ECDSA). Las entidades utilizan el algoritmo de hash seguro definido en las [publicaciones sobre normas federales de procesamiento de la información, FIPS PUB 180-4](#), conocido como. SHA384 Las claves se generan en la curva secp384r1 (NIST-P384).

Cifrado doble

Al cifrar sus datos, los datos están protegidos, pero tiene que proteger su clave de cifrado. Una estrategia consiste en cifrarla. El cifrado de sobres es la práctica de cifrar los datos en texto no cifrado con una clave de datos y, a continuación, cifrar la clave de datos con otra clave.

Incluso puede cifrar la clave de cifrado de datos con otra clave de cifrado y cifrar dicha clave de cifrado con otra clave de cifrado. Sin embargo, finalmente, una clave debe permanecer en texto no cifrado para que pueda descifrar las claves y los datos. Esta clave de cifrado de clave de texto no cifrado de nivel superior se conoce como clave raíz.



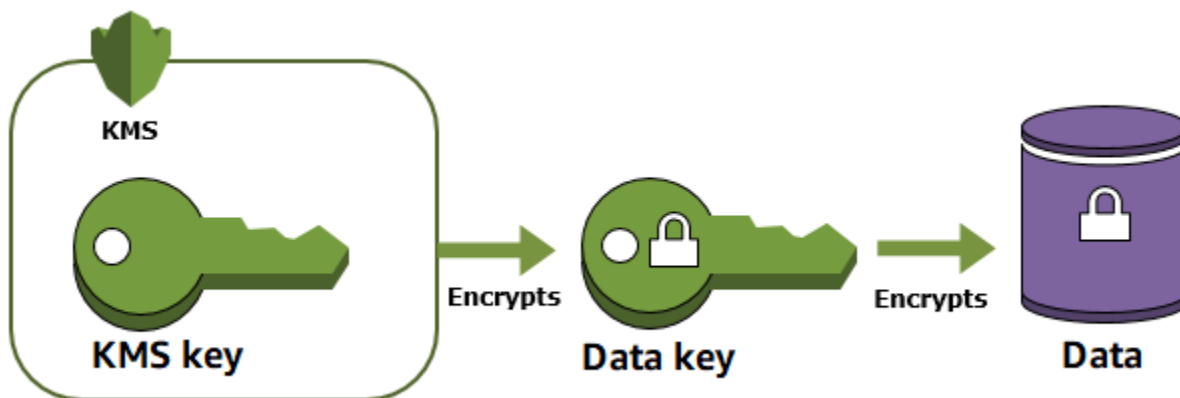
AWS KMS le ayuda a proteger sus claves de cifrado almacenándolas y administrándolas de forma segura. La clave raíz almacenada AWS KMS, conocida como AWS KMS keys, nunca deje sin cifrar los [módulos de seguridad de hardware validados por el AWS KMS FIPS 140-3 de nivel de seguridad 3](#). Para usar una clave KMS, debe llamar. AWS KMS

Una construcción básica utilizada en muchos sistemas criptográficos es el cifrado doble. El cifrado doble utiliza dos o más claves criptográficas para proteger un mensaje. Por lo general, una clave se deriva de una clave estática K de plazo más largo y otra clave es una clave por mensaje, msgKey, que se genera para cifrar el mensaje. El doble cifrado se forma al cifrar el mensaje: ciphertext = Encrypt(msgKey, message). A continuación, la clave del mensaje se cifra con la clave estática a largo plazo: encKey = Encrypt(k, msgKey). Por último, los dos valores (encKey, ciphertext) se empaquetan en una sola estructura, o mensaje con cifrado doble.

El destinatario, con acceso a la K, puede abrir el mensaje con doble cifrado al descifrar primero la clave cifrada y, a continuación, al descifrar el mensaje.

AWS KMS ofrece la posibilidad de gestionar estas claves estáticas de larga duración y automatizar el proceso de cifrado de sobres de sus datos.

Además de las funciones de cifrado incluidas en el AWS KMS servicio, el [SDK de cifrado proporciona bibliotecas de AWS cifrado](#) de sobres para el lado del cliente. Puede utilizar estas bibliotecas para proteger sus datos y las claves de cifrado que se utilizan a fin de cifrar esos datos.



El cifrado de sobre ofrece varios beneficios:

- Protección de las claves de datos

Al cifrar una clave de datos, no tiene que preocuparse del almacenamiento la clave de datos cifrada, porque la clave de datos está intrínsecamente protegida por el cifrado. Puede almacenar de forma segura la clave de datos cifrada junto con los datos cifrados.

- Cifrado de los mismos datos con varias claves múltiples

Las operaciones de cifrado pueden tardar mucho tiempo, en concreto cuando los datos que se cifran son objetos grandes. En vez de volver a cifrar los datos sin procesar varias veces con claves distintas, puede volver a cifrar solo las claves de datos que protegen los datos sin procesar.

- Combinación de los puntos fuertes de varios algoritmos

En general, los algoritmos de clave simétrica son más rápidos y producen textos cifrados más pequeños que los algoritmos de clave pública. Sin embargo, los algoritmos de clave pública proporcionan una separación inherente entre las funciones y facilitan la administración de las claves. El cifrado de sobre le permite combinar los puntos fuertes de cada estrategia.

Operaciones criptográficas

En AWS KMS, las operaciones criptográficas son operaciones de API que utilizan claves de KMS para proteger los datos. Como las claves de KMS permanecen dentro AWS KMS, debe llamar AWS KMS para usar una clave de KMS en una operación criptográfica.

Para realizar operaciones criptográficas con claves KMS, utilice AWS SDKs, AWS Command Line Interface (AWS CLI) o. Herramientas de AWS para PowerShell No puede realizar operaciones criptográficas en la consola de AWS KMS . Para ver ejemplos de cómo llamar a las operaciones criptográficas en varios lenguajes de programación, consulte [Ejemplos de código para AWS KMS usar AWS SDKs](#).

En la siguiente tabla se enumeran las operaciones AWS KMS criptográficas. También se muestra el tipo de clave y los requisitos de [uso de clave](#) para las claves KMS utilizadas en la operación.

Operación	Tipo de clave	Uso de clave
Decrypt	Simétrico o asimétrico	ENCRYPT_DECRYPT
DeriveSharedSecret	Asimétrica	KEY_AGREEMENT
Encrypt	Simétrico o asimétrico	ENCRYPT_DECRYPT
GenerateDataKey	Simétrica	ENCRYPT_DECRYPT
GenerateDataKeyPair	Simétrica [1]	ENCRYPT_DECRYPT

Operación	Tipo de clave	Uso de clave
	No es compatible con claves KMS en almacenes de claves personalizados.	
GenerateDataKeyPairWithoutPlaintext	Simétrica [1] No es compatible con claves KMS en almacenes de claves personalizados.	ENCRYPT_DECRYPT
GenerateDataKeyWithoutPlaintext	Simétrica	ENCRYPT_DECRYPT
GenerateMac	HMAC	GENERATE_VERIFY_MAC
GenerateRandom	N/A. Esta operación no utiliza una clave KMS.	N/A
ReEncrypt	Simétrico o asimétrico	ENCRYPT_DECRYPT
Sign	Asimétrica	SIGN_VERIFY
Verificar	Asimétrica	SIGN_VERIFY
VerifyMac	HMAC	GENERATE_VERIFY_MAC

[1] Genera un par de claves de datos asimétricas protegido por una clave de KMS de cifrado simétrico.

Para obtener información acerca de los permisos para operaciones criptográficas, consulte [the section called “Referencia de permisos”](#).

Para que todos los usuarios puedan AWS KMS responder a sus necesidades y sean altamente funcionales, AWS KMS establece cuotas en cuanto al número de operaciones criptográficas

llamadas por segundo. Para obtener información, consulte [the section called “Cuotas compartidas para operaciones criptográficas”](#).

Algoritmos criptográficos compatibles

Algoritmos criptográficos

En las siguientes tablas se resumen los algoritmos criptográficos, los cifrados, los modos y los tamaños de clave que se utilizan en sus servicios AWS para proteger sus datos. No deben considerarse una lista exhaustiva de todas las opciones de criptografía disponibles en AWS. Los algoritmos se dividen en dos categorías:

- Los algoritmos preferidos cumplen con los estándares de AWS seguridad y rendimiento.
- Se pueden usar algoritmos aceptables para garantizar la compatibilidad en algunas aplicaciones, pero no se prefieren.

Criptografía asimétrica

En la siguiente tabla se enumeran los algoritmos asimétricos compatibles para el cifrado, la concordancia de claves y las firmas digitales.

Tipo	Algoritmo	Status
Cifrado	RSA-OAEP (módulo de 2048 o 3072 bits)	Aceptable
Cifrado	HPKE (P-256 o P-384, HKDF y AES-GCM)	Aceptable
Acuerdo de claves	ML-KEM-768 o ML-KEM-1024	Preferido (resistente a la información cuántica)
Acuerdo de claves	ECDH(E) con P-384	Aceptable
Acuerdo de claves	ECDH(E) con P-256, P-521 o X25519	Aceptable

Tipo	Algoritmo	Status
Acuerdo de claves	ECDH(E) con Brainpool P256R1, Brainpool P384R1 o Brainpool P512R1	Aceptable
Firmas	ML-DSA-65 o ML-DSA-87	Preferido (resistente a la información cuántica)
Firmas	SLH-DSA	Preferido (firma con resistencia cuántica) software/firmware
Firmas	ECDSA con P-384	Aceptable
Firmas	ECDSA con P-256, P-521 o Ed25519	Aceptable
Firmas	RSA-2048 o RSA-3072	Aceptable

Criptografía simétrica

En la siguiente tabla se enumeran los algoritmos simétricos compatibles para el cifrado, el cifrado autenticado y el encapsulamiento de claves.

Tipo	Algoritmo	Status
Cifrado autenticado	AES-GCM-256	Preferido
Cifrado autenticado	AES-GCM-128	Aceptable
Cifrado autenticado	ChaCha20/Poly1305	Aceptable
Modos de cifrado	AES-XTS-256 (para almacenamiento en bloques)	Preferido
Modos de cifrado	AES-CBC/ CTR (modos no autenticados)	Aceptable
Encapsulamiento de claves	AES-GCM-256	Preferido

Tipo	Algoritmo	Status
Encapsulamiento de claves	AES-KW o AES-KWP con claves de 256 bits	Aceptable

Funciones criptográficas

En la siguiente tabla se enumeran los algoritmos compatibles para el hash, la derivación de claves, la autenticación de mensajes y el hash de contraseñas.

Tipo	Algoritmo	Status
Hashing	SHA2-384	Preferido
Hashing	SHA2-256	Aceptable
Hashing	SHA3	Aceptable
Derivación de claves	HKDF_Expand o HKDF con -256 SHA2	Preferido
Derivación de claves	Modo contador KDF con HMAC- -256 SHA2	Aceptable
Código de autenticación de mensajes	HMAC- -384 SHA2	Preferido
Código de autenticación de mensajes	HMAC- -256 SHA2	Aceptable
Código de autenticación de mensajes	KMAC	Aceptable
Hash de contraseñas	escribir con SHA384	Preferido
Hash de contraseñas	PBKDF2	Aceptable

Para obtener más información sobre los algoritmos criptográficos implementados en AWS, consulte [Algoritmos criptográficos](#) y. Servicios de AWS

Permisos y acceso a claves KMS

Para usarlo AWS KMS, debe tener credenciales que AWS pueda usar para autenticar sus solicitudes. Las credenciales deben incluir permisos para acceder a AWS los recursos AWS KMS keys y [alias](#). Ningún AWS director tiene permisos para acceder a una clave de KMS a menos que dicho permiso se otorgue de forma explícita y nunca se deniegue. No hay permisos implícitos ni automáticos para usar o administrar una clave KMS.

Para controlar el acceso a las claves KMS, puede utilizar los siguientes mecanismos de políticas.

- [Política de claves](#): cada clave KMS tiene una política de claves. Es el mecanismo principal para controlar el acceso a una clave KMS. Puede utilizar solo la política de claves para controlar el acceso, lo que significa que el ámbito completo de acceso a la clave KMS se define en un único documento (la política de claves). Para obtener más información sobre el uso de políticas de claves, consulte [Políticas de claves](#).
- [Políticas de IAM](#): puede utilizar políticas de IAM en combinación con la política de claves y subvenciones para controlar el acceso a una clave KMS. Este modo de controlar el acceso le permite administrar todos los permisos de las identidades de IAM en IAM. Para utilizar una política de IAM a fin de permitir el acceso a una clave KMS, la política de claves debe permitirlo explícitamente. Para obtener más información sobre el uso de políticas de IAM, consulte [Políticas de IAM](#).
- [Concesiones](#): puede utilizar concesiones en combinación con la política de claves y políticas de IAM para permitir el acceso a una clave KMS. Con este modo de controlar el acceso, puede permitir el acceso a la clave KMS en la política de claves y permitir que las identidades deleguen su acceso a otros. Para obtener más información sobre cómo usar concesiones, consulte [Subvenciones en AWS KMS](#).

Políticas de clave de KMS

La forma principal de administrar el acceso a AWS KMS los recursos es mediante políticas. Las políticas son documentos que describen qué entidades principales pueden acceder a qué recursos. Las políticas asociadas a una identidad de IAM se denominan políticas basadas en la identidad (o políticas de IAM) y las políticas asociadas a otros tipos de recursos se denominan políticas de recursos. AWS KMS las políticas de recursos para las claves de KMS se denominan políticas clave.

Todas las claves KMS tienen una política de claves. Si no proporciona una, AWS KMS crea una para usted. La [política de claves predeterminada](#) que se AWS KMS utiliza varía en función de si se crea la clave en la AWS KMS consola o se utiliza la AWS KMS API. Le recomendamos que edite la política de claves predeterminada para adaptarla a los requisitos de su organización en materia de [permisos de privilegio mínimo](#).

Puede utilizar únicamente la política de claves para controlar el acceso si la clave y la entidad principal de IAM están en la misma AWS cuenta, lo que significa que todo el alcance del acceso a la clave de KMS se define en un único documento (la política de claves). Sin embargo, cuando una persona que llama desde una cuenta tiene que acceder a una clave de otra cuenta, no puede utilizar únicamente la política de claves para conceder el acceso. En un escenario de varias cuentas, se debe adjuntar al usuario o rol de la persona que llama una política de IAM que permita explícitamente a dicha persona que llama realizar la llamada a la API.

También puede utilizar políticas de IAM en combinación con las concesiones y las políticas de claves para controlar el acceso a una clave KMS. Para utilizar una política de IAM a fin de controlar el acceso a una clave KMS, la política de claves debe conceder permiso a la cuenta para utilizar políticas de IAM. Puede especificar una [declaración de política de claves que habilite las políticas de IAM](#) o puede [especificar explícitamente la entidad principal permitida](#) en la política de claves.

Al redactar las políticas, asegúrese de contar con controles estrictos que restrinjan quién puede realizar las siguientes acciones:

- Actualizar, crear y eliminar las políticas de claves de IAM y KMS
- Adjuntar y separar políticas de IAM de los usuarios, los roles y los grupos
- Adjuntar y separar las políticas de claves KMS de sus claves KMS

Concesiones de claves de KMS

[Además de la IAM y las políticas clave, AWS KMS apoya las subvenciones.](#) Las subvenciones proporcionan una forma flexible y eficaz de gestionar qué directores de IAM o directores de AWS servicio pueden acceder a una clave sin modificar una política clave. Esto puede incluir casos de uso en los que los permisos solo sean necesarios mientras dure algún otro trabajo o recurso. Se llama a una API para crear la concesión y para retirarla o revocarla cuando ya no se necesita.

Para obtener más información, consulte [Subvenciones en AWS KMS](#).

Políticas clave en AWS KMS

Una política clave es una política de recursos para un. AWS KMS key Las políticas de claves son la forma principal de controlar el acceso a las claves KMS. Cada clave KMS debe tener exactamente una política de claves. Las declaraciones de políticas de claves determinan quién tiene permiso para usar la clave KMS y cómo debe usar dicho permiso. También puede utilizar [políticas de IAM](#) y [concesiones](#) para controlar el acceso a la clave KMS, pero cada clave KMS debe tener un documento de política de claves.

Ningún responsable AWS , ni siquiera el usuario raíz de la cuenta o el creador de la clave, tiene permisos para acceder a una clave de KMS, a menos que se autorice explícitamente, y nunca se deniegue, en una política de claves, una política de IAM o una concesión.

A menos que la política de claves lo permita explícitamente, no puede utilizar las políticas de IAM para permitir el acceso a una clave KMS. Sin el permiso de la política de claves, las políticas de IAM que conceden permisos no tienen ningún efecto. (Puede utilizar una política de IAM para denegar un permiso a una clave KMS sin el permiso de una política de claves). La política de claves predeterminada habilita las políticas de IAM. Para habilitar las políticas de IAM en la política de claves, agregue la declaración de política descrita en [Permite el acceso a las políticas de IAM Cuenta de AWS y las habilita](#).

A diferencia de las políticas de IAM, que son globales, las políticas de claves son regionales. Una política de claves controla el acceso solo a una clave KMS en la misma región. No tiene ningún efecto sobre las claves KMS de otras regiones.

Temas

- [Creación de una política de claves](#)
- [Política de claves predeterminada](#)
- [Visualización de políticas de claves](#)
- [Cambio de una política de claves](#)
- [Permisos para AWS los servicios en las políticas clave](#)

Creación de una política de claves

Puede crear y administrar políticas clave en la AWS KMS consola o mediante operaciones de AWS KMS API, como [CreateKeyReplicateKey](#), y [PutKeyPolicy](#).

Al crear una clave de KMS en la AWS KMS consola, la consola le explicará los pasos necesarios [para crear una política de claves basada en la política de claves predeterminada de la consola](#). Al utilizar la política de claves CreateKey o ReplicateKey APIs, si no se especifica, APIs se aplica la política de [claves predeterminada a las claves creadas mediante programación](#). Al usar la API PutKeyPolicy, es necesario especificar una política de claves.

Cada documento de política puede tener una o varias declaraciones de política. En el siguiente ejemplo se muestra un documento válido de política de claves con una declaración 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"
        }
      }
    }
  ]
}
```

Temas

- [Formato de la política de claves](#)
- [Elementos de una política de claves](#)
- [Ejemplo de política de claves de](#)

Formato de la política de claves

Un documento de política de claves debe cumplir las siguientes reglas:

- Pesar hasta 32 kilobytes (32 768 bytes)
- El elemento `Sid` de una declaración de política de claves puede incluir espacios. (Se prohíben los espacios en el elemento `Sid` de un documento de política de IAM).

Un documento de política de claves solo puede incluir los siguientes caracteres:

- Caracteres ASCII imprimibles
- Caracteres imprimibles del conjunto de caracteres Basic Latin y Latin-1 Supplement
- Caracteres especiales como la pestaña (`\u0009`), la fuente de línea (`\u000A`) y retorno de carro (`\u000D`)

Elementos de una política de claves

Un documento de política de claves debe tener los siguientes elementos:

Versión

Especifica la versión del documento de política de claves. Configure la versión en `2012-10-17` (la última versión).

Instrucción

Contiene las declaraciones de la política. Un documento de política de claves debe tener al menos una declaración.

Cada declaración de política de claves consta de hasta seis elementos. Los elementos `Effect`, `Principal`, `Action`, y `Resource` son necesarios.

Sid

(Opcional) El identificador de la declaración (`Sid`) una cadena arbitraria que puede utilizar para describir la declaración. El `Sid` en una política de claves puede incluir espacios. (No puede incluir espacios en un elemento `Sid` de política de IAM).

Efecto

(Obligatorio) Determina si desea permitir o denegar los permisos en la declaración de política. Los valores válidos son Allow o Deny. Si no permite el acceso de forma explícita a una clave KMS, el acceso se deniega implícitamente. También puede denegar explícitamente el acceso a una clave KMS. Puede hacer esto para asegurarse de que un usuario no pueda tener acceso a ella, aunque otra política le permita el acceso.

Principal

(Obligatorio) La [entidad principal](#) es la identidad que obtiene los permisos especificados en la declaración de política. Puede especificar Cuentas de AWS los usuarios de IAM, las funciones de IAM y algunos AWS servicios como principales en una política clave. Los [grupos de usuarios](#) de IAM no son una entidad principal válida en ningún tipo de política.

Un valor de asterisco, como "AWS": "*", representa todas las identidades de AWS de todas las cuentas.

Important

No establezca la Entidad principal en un asterisco (*) en ninguna declaración de política de claves que permita permisos a menos que utilice [condiciones](#) para limitar la política de claves. Un asterisco indica todas las identidades de cada Cuenta de AWS permiso para usar la clave de KMS, a menos que otra declaración de política lo deniegue explícitamente. Los usuarios de otras Cuentas de AWS pueden usar su clave de KMS siempre que tengan los permisos correspondientes en su propia cuenta.

Note

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;.

Cuando la entidad principal en una declaración de política de claves es una [entidad principal de Cuenta de AWS](#) se expresa como `arn:aws:iam::111122223333:root`, la declaración de la política no concede permisos a ninguna entidad principal de IAM.

En su lugar, otorga Cuenta de AWS permiso para usar las políticas de IAM para delegar los permisos especificados en la política clave. (Una entidad principal en formato `arn:aws:iam::111122223333:root` no representa al [usuario raíz de la cuenta de AWS](#), a pesar del uso de “root” en el identificador de la cuenta. Sin embargo, la entidad principal de la cuenta representa a la cuenta y a sus administradores, incluido el usuario raíz de la cuenta).

Cuando el principal es otro Cuenta de AWS o sus directores, los permisos solo entran en vigor cuando la cuenta está habilitada en la región con la clave y la política de claves de KMS. Para obtener información acerca de las regiones que no están habilitadas de forma predeterminada (“Regiones de adhesión”), consulte [Administración de Regiones de AWS](#) en la Referencia general de AWS.

Para permitir que otro Cuenta de AWS o sus principales usuarios utilicen una clave de KMS, debes conceder el permiso en una política de claves y en una política de IAM en la otra cuenta. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

Action

(Obligatorio) Especifique las operaciones de la API que se permitirán o denegarán. Por ejemplo, la `kms:Encrypt` acción corresponde a la operación de AWS KMS [cifrado](#). Puede enumerar varias acciones en una declaración de política. Para obtener más información, consulte [Referencia de permisos](#).

Note

Si falta el elemento `Action` requerido en una declaración de política de claves, la declaración de la política no tiene efecto. Una declaración de política de claves sin un elemento `Action` no se aplica a ninguna clave KMS.

Cuando a una declaración de política clave le falta su `Action` elemento, la AWS KMS consola informa correctamente de un error, pero lo [CreateKey](#) hace correctamente [PutKeyPolicy](#) APIs , aunque la declaración de política no sea efectiva.

Recurso

(Obligatorio) En una política de claves, el valor del elemento `Recurso` es “*”, que significa “esta clave KMS”. El asterisco (“*”) identifica la clave KMS a la que se adjunta la política de clave.

 Note

Si falta el elemento Resource requerido en una declaración de política de claves, la declaración de la política no tiene efecto. Una declaración de política de claves sin un elemento Resource no se aplica a ninguna clave KMS.

Cuando a una declaración de política clave le falta su Resource elemento, la AWS KMS consola informa correctamente de un error, pero lo hace [CreateKey](#) correctamente, [PutKeyPolicy](#) APIs a pesar de que la declaración de política no es efectiva.

Condición

(Opcional) Las condiciones especifican requisitos que deben cumplirse para que se aplique una política de claves. Con condiciones, AWS puede evaluar el contexto de una solicitud de API para determinar si la declaración de política se aplica o no.

Para especificar las condiciones, se utilizan claves de condición predefinidas. AWS KMS admite claves de [condición AWS globales y claves de AWS KMS condición](#). Para admitir el control de acceso basado en atributos (ABAC), AWS KMS proporciona claves de condición que controlan el acceso a una clave KMS en función de etiquetas y alias. Para obtener más información, consulte [ABAC para AWS KMS](#).

El formato de una condición es:

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

por ejemplo:

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

Para obtener más información sobre la sintaxis de las AWS políticas, consulte la [Referencia de políticas de AWS IAM en la Guía del usuario](#) de IAM.

Ejemplo de política de claves de

En el siguiente ejemplo se muestra una política de claves completa para una clave KMS de cifrado simétrica. Puede utilizarlo como referencia mientras lee acerca de los conceptos de política de claves

de este capítulo. Esta política de claves combina los ejemplos de declaraciones de política de la sección [política de claves predeterminada](#) anterior en una sola política de claves que lleva a cabo lo siguiente:

- Permite al ejemplo Cuenta de AWS 111122223333 tener acceso total a la clave KMS. Permite que la cuenta y sus administradores, incluido el usuario raíz de la cuenta (para emergencias), utilicen las políticas de IAM en la cuenta para permitir el acceso a la clave KMS.
- Permite que el rol de IAM ExampleAdminRole administre la clave de KMS.
- Permite que el rol de IAM ExampleUserRole utilice la clave 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 claves predeterminada

Al crear una clave KMS, puede especificar la política de claves para la nueva clave KMS. Si no la proporciona, AWS KMS crea una para usted. La política de claves predeterminada que se AWS KMS utiliza varía en función de si se crea la clave en la AWS KMS consola o se utiliza la AWS KMS API.

Política de claves predeterminada al crear una clave KMS mediante programación

Cuando se crea una clave de KMS mediante programación con la [AWS KMS API](#) (incluso mediante el uso de [AWS SDKs](#), [AWS Command Line Interface](#) o [Herramientas de AWS para PowerShell](#)) y no se especifica una política de claves, AWS KMS se aplica una política de claves predeterminada muy simple. Esta política de claves predeterminada tiene una declaración de política que otorga al propietario de la Cuenta de AWS clave de KMS permiso para usar las políticas de IAM a fin de permitir el acceso a todas las AWS KMS operaciones de la clave de KMS. Para obtener más información sobre esta declaración de política, consulte [Permite el acceso a las políticas de IAM Cuenta de AWS y las habilita](#).

Política de claves predeterminada al crear una clave KMS con Consola de administración de AWS

Al [crear una clave de KMS con la Consola de administración de AWS](#), la política clave comienza con la declaración de política que [permite el acceso a las políticas de IAM Cuenta de AWS y las habilita](#). A continuación, la consola añade una [declaración del administrador clave](#), una [declaración de los usuarios clave](#) y (para la mayoría de los tipos de claves) una declaración que permite a los directores utilizar la clave de KMS con [otros AWS](#) servicios. Puede utilizar las funciones de la AWS KMS consola para especificar los usuarios de IAM, las funciones de IAM y Cuentas de AWS quiénes son los administradores clave y los usuarios clave (o ambos).

Permisos

- [Permite el acceso a las políticas de IAM Cuenta de AWS y las habilita](#)
- [Permite que los administradores de claves administren la clave KMS](#)
- [Permite a los usuarios de claves utilizar la clave KMS](#)
 - [Permite a los usuarios de claves utilizar una clave KMS para las operaciones criptográficas](#)
 - [Permite a los usuarios clave utilizar la clave KMS con AWS los servicios](#)

Permite el acceso a las políticas de IAM Cuenta de AWS y las habilita

La siguiente declaración de política de claves predeterminada es fundamental.

- Otorga al Cuenta de AWS propietario de la clave KMS acceso completo a la clave KMS.

A diferencia de otras políticas de AWS recursos, una política AWS KMS clave no otorga automáticamente permisos a la cuenta ni a ninguna de sus identidades. Para conceder permisos a los administradores de cuentas, la política de claves debe incluir una declaración explícita que proporcione este permiso, de forma similar a ésta.

- Permite que la cuenta utilice las políticas de IAM para permitir el acceso a la clave KMS, además de la política de claves.

Sin este permiso, las políticas de IAM que permiten el acceso a la clave son ineficaces, aunque las políticas de IAM que deniegan el acceso a la clave siguen siendo eficaces.

- Reduce el riesgo de que la clave deje de poder administrarse dando permiso de control de acceso a los administradores de la cuenta, incluido el usuario raíz de la cuenta, que no se puede eliminar.

La siguiente declaración de política de claves es toda la política de claves predeterminada para las claves KMS creadas mediante programación. Es la primera declaración de política de la política de claves predeterminada para las claves de KMS que se crean en la AWS KMS consola.

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

Permite que las políticas de IAM permitan el acceso a la clave KMS.

La declaración de política clave Cuenta de AWS que se muestra arriba otorga al propietario de la clave permiso para usar las políticas de IAM, así como las políticas clave, para permitir todas las acciones (`kms : *`) en la clave de KMS.

La entidad principal en esta declaración de política de claves es la [entidad principal](#) de la cuenta, que está representada por un ARN en este formato: `arn:aws:iam::account-id:root`. El principal de la cuenta representa a la AWS cuenta y a sus administradores.

Cuando la entidad principal de una declaración de política de claves es la entidad principal de la cuenta, la declaración de política no concede permisos a ninguna entidad principal de IAM para utilizar la clave KMS. En su lugar, permite que la cuenta utilice las políticas de IAM para delegar los permisos especificados en la declaración de la política. Esta declaración de política de claves predeterminada permite a la cuenta utilizar las políticas de IAM para delegar el permiso para todas las acciones (`kms : *`) en la clave KMS.

reduce el riesgo de que la clave KMS deje de poder administrarse.

A diferencia de otras políticas de AWS recursos, una política AWS KMS clave no otorga automáticamente permisos a la cuenta ni a ninguno de sus directores. Para dar permisos a cualquier entidad principal, incluida la [entidad principal de la cuenta](#), debe utilizar una declaración de política de claves que proporcione los permisos de forma explícita. No es necesario dar a la entidad principal de la cuenta, ni a ninguna entidad principal, acceso a la clave KMS. Sin embargo, dar acceso a la entidad principal de la cuenta le ayuda a evitar que la clave deje de poder administrarse.

Por ejemplo, supongamos que se crea una política de claves que da acceso a la clave KMS a un solo usuario. Si posteriormente elimina ese usuario, la clave deja de poder administrarse y deberá [ponerse en contacto con el servicio de asistencia de AWS](#) para recuperar el acceso a la clave KMS.

La declaración de política de claves que se muestra arriba otorga a la [entidad principal de la cuenta](#) permiso para controlar la clave. El principal de la cuenta representa al usuario raíz de la cuenta Cuenta de AWS y a sus administradores, incluido el [usuario raíz de la cuenta](#). La cuenta de usuario raíz es la única entidad principal que no se puede eliminar a menos que se elimine la cuenta de Cuenta de AWS. Las prácticas recomendadas de IAM desaconsejan actuar en nombre del usuario raíz de la cuenta, excepto en caso de emergencia. Sin embargo, es posible que tenga que actuar como usuario raíz de la cuenta si elimina todos los demás usuarios y funciones con acceso a la clave KMS.

Permite que los administradores de claves administren la clave KMS

La política de claves predeterminada creada por la consola le permite elegir usuarios y roles de IAM de la cuenta y convertirlos en administradores de claves. Esta declaración se denomina la

declaración de los administradores de claves. Los administradores de claves tienen permisos para administrar la clave KMS, pero no para utilizar la clave KMS en [operaciones criptográficas](#). Puede agregar usuarios y roles de IAM a la lista de administradores de claves al crear la clave KMS en la vista predeterminada o en la vista de política.

 Warning

Como los administradores de claves tienen permiso para cambiar la política de claves y crear concesiones, pueden concederse a sí mismos y a otras personas AWS KMS permisos no especificados en esta política.

Las entidades principales que tienen permiso para administrar etiquetas y alias también pueden controlar el acceso a una clave KMS. Para obtener más información, consulte [ABAC para AWS KMS](#).

 Note

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;.

En el siguiente ejemplo se muestra la declaración de los administradores de claves en la vista predeterminada de la consola de 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 continuación se muestra un ejemplo de declaración de los administradores de claves en la vista de política de la consola de AWS KMS . Esta declaración de los administradores de claves es para una clave de KMS de cifrado simétrica de una sola región.

Note

La AWS KMS consola agrega administradores clave a la política clave bajo el identificador de la declaración "Allow access for Key Administrators". La modificación de este identificador de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

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

La declaración de los administradores de claves predeterminada para la clave KMS más común, una clave KMS de cifrado simétrica de una región única, permite los siguientes permisos. Para obtener información detallada sobre cada permiso, consulte la [AWS KMS permisos](#).

Al usar la AWS KMS consola para crear una clave KMS, la consola agrega los usuarios y roles que especifique al `Principal` elemento en la declaración del administrador clave.

Muchos de estos permisos contienen el carácter comodín (*), que permite todos los permisos que empiezan por el verbo especificado. Como resultado, cuando se AWS KMS añaden nuevas operaciones de API, los administradores de claves pueden utilizarlas automáticamente. No es necesario actualizar las políticas de claves para incluir las nuevas operaciones. Si prefiere limitar sus administradores de claves a un conjunto fijo de operaciones de API, puede [cambiar la política de claves](#).

kms:Create*

Permite [kms:CreateAlias](#) y [kms:CreateGrant](#). (El permiso `kms:CreateKey` solo es válido en una política de IAM).

kms:Describe*

Permite [kms:DescribeKey](#). Se requiere el permiso `kms:DescribeKey` para ver la página de detalles de clave de una clave KMS en la Consola de administración de AWS.

kms:Enable*

Permite [kms:EnableKey](#). Para las claves KMS de cifrado simétricas, también se permite [kms:EnableKeyRotation](#).

kms:List*

Permite [kms:ListGrants](#), [kms:ListKeyPolicies](#) y [kms:ListResourceTags](#). (Los permisos `kms:ListAliases` y `kms:ListKeys`, necesarios para ver las claves KMS en la Consola de administración de AWS, solo son válidos en las políticas de IAM.)

kms:Put*

Permite [kms:PutKeyPolicy](#). Este permiso permite a los administradores de claves cambiar la política de claves de esta clave KMS.

kms:Update*

Permite [kms:UpdateAlias](#) y [kms:UpdateKeyDescription](#). Para las claves de varias regiones, permite [kms:UpdatePrimaryRegion](#) en esta clave KMS.

kms:Revoke*

Permite [kms:RevokeGrant](#), que permite a los administradores de claves [eliminar una concesión](#) incluso si no son una [entidad principal retirada](#) en la concesión.

kms:Disable*

Permite [kms:DisableKey](#). Para las claves KMS de cifrado simétricas, también se permite [kms:DisableKeyRotation](#).

kms:Get*

Permite [kms:GetKeyPolicy](#) y [kms:GetKeyRotationStatus](#). Para claves KMS con material de claves importado, permite [kms:GetParametersForImport](#). Para claves KMS asimétricas, permite [kms:GetPublicKey](#). Se requiere el permiso `kms:GetKeyPolicy` para ver la política de claves de una clave KMS en la Consola de administración de AWS.

kms:Delete*

Permite [kms:DeleteAlias](#). Para claves con material de claves importado, permite [kms:DeleteImportedKeyMaterial](#). El permiso `kms:Delete*` no permite a los administradores de claves eliminar la clave KMS (`ScheduleKeyDeletion`).

kms:TagResource

Permite [kms:TagResource](#), que permite a los administradores de claves agregar etiquetas a la clave KMS. Dado que las etiquetas también se pueden utilizar para controlar el acceso a la clave KMS, este permiso permite a los administradores permitir o denegar el acceso a la clave KMS. Para obtener más información, consulte [ABAC para AWS KMS](#).

kms:UntagResource

Permite [kms:UntagResource](#), que permite a los administradores de claves eliminar etiquetas de la clave KMS. Dado que las etiquetas se pueden utilizar para controlar el acceso a la clave, este permiso permite a los administradores permitir o denegar el acceso a la clave KMS. Para obtener más información, consulte [ABAC para AWS KMS](#).

kms:ScheduleKeyDeletion

Permite [kms:ScheduleKeyDeletion](#), que permite a los administradores de claves [eliminar esta clave KMS](#). Para eliminar este permiso, desactive la opción Allow key administrators to delete this key (Permitir a los administradores de claves eliminar esta clave).

kms:CancelKeyDeletion

Permite [kms:CancelKeyDeletion](#), que permite a los administradores de claves [cancelar la eliminación de esta clave KMS](#). Para eliminar este permiso, desactive la opción Allow key administrators to delete this key (Permitir a los administradores de claves eliminar esta clave).

kms:RotateKeyOnDemand

Permite [kms:RotateKeyOnDemand](#), que permite a los administradores de claves [realizar la rotación bajo demanda del material de claves de esta clave KMS](#).

AWS KMS añade los siguientes permisos a la declaración de administradores de claves predeterminada al crear claves de uso especial.

kms:ImportKeyMaterial

El permiso [kms:ImportKeyMaterial](#) permite a los administradores de claves importar material de claves KMS. Este permiso solo se incluye en la política de claves cuando [crea una clave KMS sin material de claves](#).

kms:ReplicateKey

El [kms:ReplicateKey](#) permiso permite a los administradores de claves [crear una réplica de una clave principal multirregional en otra región](#). AWS Este permiso solo se incluye en la política de claves cuando crea una clave principal o de réplica de varias regiones.

kms:UpdatePrimaryRegion

El permiso [kms:UpdatePrimaryRegion](#) permite a los administradores de claves [cambiar una clave de réplica de varias regiones a una clave principal de varias regiones](#). Este permiso solo se incluye en la política de claves cuando crea una clave principal o de réplica de varias regiones.

Permite a los usuarios de claves utilizar la clave KMS

La política de claves predeterminada que la consola crea para las claves de KMS permite elegir los usuarios y roles de IAM en la cuenta y los externos Cuentas de AWS, y convertirlos en usuarios clave.

La consola agrega dos declaraciones de política a la política de claves para los usuarios de claves.

- [Utilice la clave KMS directamente](#): la primera declaración de política de claves da a los usuarios de claves permiso para usar la clave KMS directamente para todas las [operaciones criptográficas](#) de ese tipo de clave KMS.
- [Use la clave de KMS con AWS los servicios](#): la segunda declaración de política otorga a los usuarios clave permiso para permitir que los AWS servicios que están integrados usen la clave de KMS en su nombre para proteger los recursos, como los buckets de Amazon S3 y las tablas de Amazon DynamoDB. AWS KMS

Puede añadir usuarios de IAM, funciones de IAM y otros Cuentas de AWS a la lista de usuarios clave al crear la clave de KMS. También puede editar la lista con la vista predeterminada de la consola para las políticas de claves, tal como se muestra en la siguiente imagen. La vista predeterminada de las políticas de claves está disponible en la página de detalles de clave. Para obtener más información sobre cómo permitir que los usuarios de otros países Cuentas de AWS usen la clave de KMS, consulte. [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#)

Note

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de 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

Las declaraciones de usuarios de claves predeterminadas para una clave simétrica de región única permite los siguientes permisos. Para obtener información detallada sobre cada permiso, consulte la [AWS KMS permisos](#).

Cuando usa la AWS KMS consola para crear una clave KMS, la consola agrega los usuarios y roles que especifique al Principal elemento en la declaración de cada usuario clave.

Note

La AWS KMS consola agrega los usuarios clave a la política clave bajo los identificadores de la declaración "Allow use of the key" y "Allow attachment of persistent

`resources`". La modificación de estos identificadores de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

```
{
  "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 a los usuarios de claves utilizar una clave KMS para las operaciones criptográficas

Los usuarios de claves tienen permiso para usar la clave KMS directamente en todas las [operaciones criptográficas](#) admitidas en la clave KMS. También pueden usar la

[DescribeKey](#) operación para obtener información detallada sobre la clave de KMS en la AWS KMS consola o mediante las operaciones de la AWS KMS API.

De forma predeterminada, la AWS KMS consola agrega las declaraciones de los usuarios clave, como las de los ejemplos siguientes, a la política de claves predeterminada. Debido a que son compatibles con diferentes operaciones de la API, las acciones en las declaraciones de la política para las claves KMS de cifrado simétricas, claves KMS HMAC, claves KMS asimétricas para el cifrado de claves público y las claves KMS asimétricas para la firma y la verificación son ligeramente diferentes.

Claves KMS de cifrado simétricas

La consola agrega la siguiente declaración a la política de claves para claves KMS de cifrado simétricas.

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

Claves KMS HMAC

La consola agrega la siguiente declaración a la política de claves para claves KMS 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": "*"
}
```

```
"Resource": "*"
}
```

Claves KMS asimétricas para el cifrado de claves públicas

La consola agrega la siguiente declaración a la política de claves para claves KMS asimétricas con un uso de claves de Encrypt and decrypt (Cifrar y descifrar).

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

Claves KMS asimétricas para la firma y la verificación

La consola agrega la siguiente declaración a la política de claves para claves KMS asimétricas con un uso de claves de Sign and verify (Firmar y 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": "*"
}
```

Claves KMS asimétricas para la obtención de secretos compartidos

La consola agrega la siguiente declaración a la política de claves para claves KMS asimétricas con un uso de claves de Key agreement (Acuerdo de claves).

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

Las acciones de estas declaraciones proporcionan a los usuarios de clave los siguientes permisos.

[kms:Encrypt](#)

Permite a los usuarios de claves cifrar datos con esta clave KMS.

[kms:Decrypt](#)

Permite a los usuarios de claves descifrar los datos con esta clave KMS.

[kms:DeriveSharedSecret](#)

Permite a los usuarios de claves obtener secretos compartidos con esta clave KMS.

[kms:DescribeKey](#)

Permite a los usuarios de claves recuperar información sobre esta clave KMS, incluidos sus identificadores, fecha de creación, estado y mucho más. También permite a los usuarios clave mostrar detalles sobre la clave KMS en la AWS KMS consola.

kms:GenerateDataKey*

Permite a los usuarios de claves solicitar una clave de datos simétrica o un par de claves de datos asimétricos para operaciones criptográficas del cliente. La consola utiliza el carácter comodín * para representar el permiso para las siguientes operaciones de API: [GenerateDataKey](#), [GenerateDataKeyWithoutPlaintext](#), [GenerateDataKeyPair](#), y

[GenerateDataKeyPairWithoutPlaintext](#). Estos permisos solo son válidos en las claves KMS simétricas que cifran las claves de datos.

[kms: GenerateMac](#)

Permite a los usuarios de claves utilizar una clave KMS HMAC para generar una etiqueta HMAC.

[km: GetPublicKey](#)

Permite a los usuarios de claves descargar la clave pública de la clave KMS asimétrica. Las partes con las que compartes esta clave pública pueden cifrar los datos fuera de AWS KMS. Sin embargo, esos textos cifrados solo se pueden descifrar llamando a la operación [Descifrar](#) en AWS KMS.

[km: * ReEncrypt](#)

Permite a los usuarios de claves volver a cifrar los datos que se habían cifrado originalmente con esta clave KMS, o utilizar esta clave KMS para volver a cifrar los datos cifrados anteriormente. La [ReEncrypt](#) operación requiere acceso a las claves KMS de origen y destino. Para lograr esto, puede habilitar el permiso `kms:ReEncryptFrom` en la clave KMS fuente y el permiso `kms:ReEncryptTo` en la clave KMS de destino. Sin embargo, para simplificar, la consola permite `kms:ReEncrypt*` (con el carácter comodín `*`) en ambas claves KMS.

[kms:Sign](#)

Permite a los usuarios de claves firmar mensajes con esta clave KMS.

[kms:Verify](#)

Permite a los usuarios de claves verificar las firmas con esta clave KMS.

[kms: VerifyMac](#)

Permite a los usuarios de claves utilizar una clave KMS HMAC para verificar una etiqueta HMAC.

Permite a los usuarios clave utilizar la clave KMS con AWS los servicios

La política de claves predeterminada de la consola también proporciona a los usuarios clave los permisos de concesión que necesitan para proteger sus datos en AWS los servicios que utilizan concesiones. AWS los servicios suelen utilizar las concesiones para obtener un permiso específico y limitado para usar una clave de KMS.

Esta declaración de política clave permite al usuario clave crear, ver y revocar concesiones en la clave de KMS, pero solo cuando la solicitud de operación de concesión proviene de un [AWS servicio](#)

[integrado](#) en ella. AWS KMS La condición de `GrantIsForAWSResource` política [kms](#): no permite al usuario llamar directamente a estas operaciones de concesión. Cuando el usuario clave lo permite, un AWS servicio puede crear una concesión en nombre del usuario que permita al servicio utilizar la clave KMS para proteger los datos del usuario.

Los usuarios de claves requieren estos permisos de concesiones para utilizar la clave KMS con los servicios integrados, pero estos permisos no son suficientes. Los usuarios de claves también necesitan permiso para utilizar los servicios integrados. Para obtener más información sobre cómo dar a los usuarios acceso a un AWS servicio que se integra con él AWS KMS, consulte la documentación del servicio 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 ejemplo, los usuarios de claves pueden utilizar estos permisos en la clave KMS de las siguientes maneras.

- Utilice esta clave KMS con Amazon Elastic Block Store (Amazon EBS) y Amazon Elastic Compute Cloud (Amazon EC2) para adjuntar un volumen de EBS cifrado a una instancia EC2. El usuario de claves concede implícitamente a Amazon EC2 permiso para utilizar la clave KMS con el fin de asociar el volumen cifrado a la instancia. Para obtener más información, consulte [Cómo utiliza Amazon Elastic Block Store \(Amazon EBS\) AWS KMS](#).
- Utilice esta clave KMS con Amazon Redshift para lanzar un clúster cifrado. El usuario de claves concede implícitamente a Amazon Redshift permiso para utilizar la clave KMS con el fin de lanzar el clúster cifrado y crear instantáneas cifradas. Para obtener más información, consulte [Cómo utiliza Amazon Redshift AWS KMS](#).
- Utilice esta clave KMS con otros [servicios de AWS integrados con AWS KMS](#), que utilizan concesiones, para crear, administrar o usar recursos cifrados con esos servicios.

La política de claves predeterminada permite a los usuarios de claves delegar su permiso de concesión a todos los servicios integrados que utilizan concesiones. Sin embargo, puede crear una política de claves personalizada que restrinja el permiso a AWS servicios específicos. Para obtener más información, consulte la clave de condición [kms: ViaService](#).

Visualización de políticas de claves

Puedes ver la política clave de una [clave gestionada por el AWS KMS cliente](#) o [Clave administrada de AWS](#) de una de tu cuenta mediante la AWS KMS consola o la [GetKeyPolicy](#) operación de la AWS KMS API. No puede utilizar estas técnicas para ver la política de claves de una clave KMS en una cuenta de Cuenta de AWS distinta.

Para obtener más información sobre las políticas AWS KMS clave, consulta [Políticas clave en AWS KMS](#). Para obtener información acerca de cómo determinar qué usuarios y roles tienen acceso a un clave KMS, consulte [the section called "Determinar el acceso"](#).

Uso de la AWS KMS consola

Los usuarios autorizados pueden ver la política de claves para una [Clave administrada de AWS](#) o una [clave administrada por el cliente](#) en la pestaña Key policy (Política de claves) de la Consola de administración de AWS.

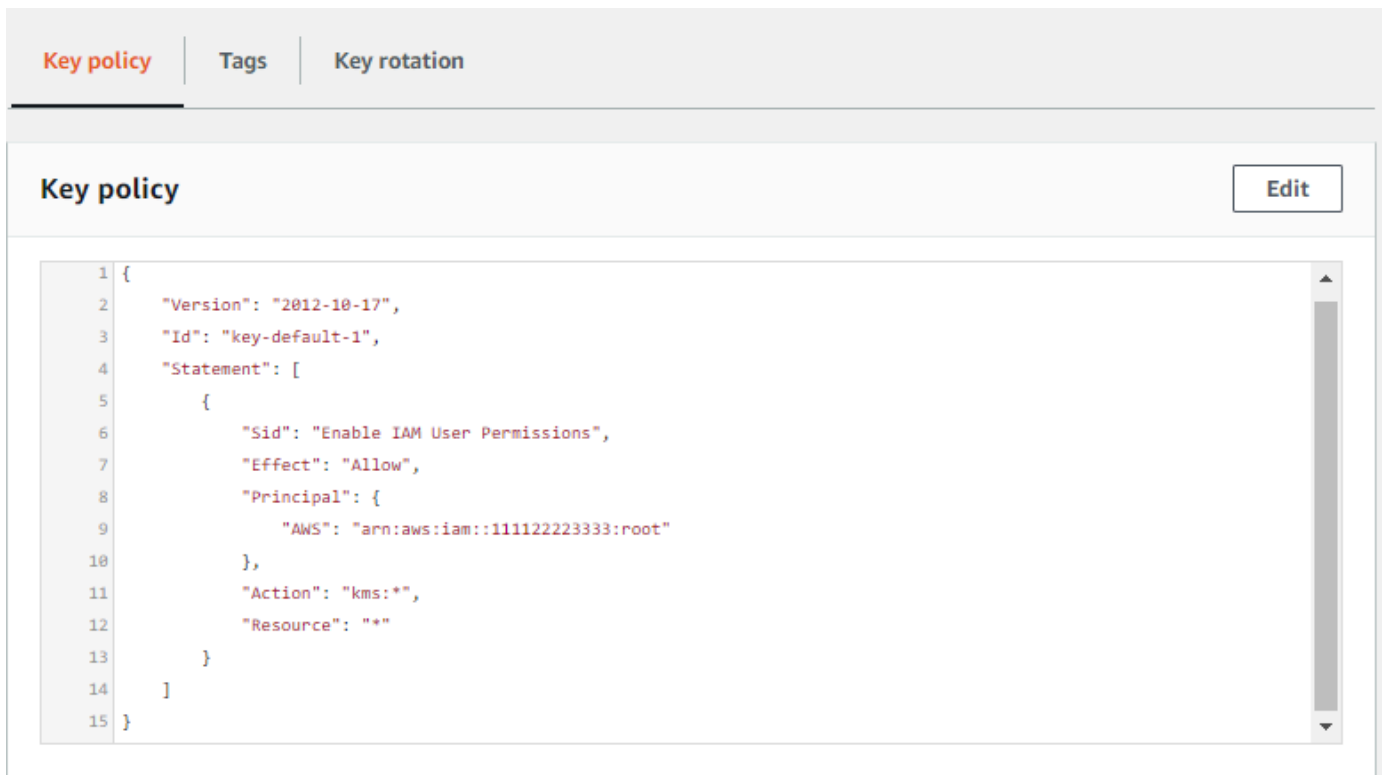
[Para ver la política clave de una clave de KMS en Consola de administración de AWS, debe tener los GetKeyPolicy permisos kms: DescribeKey, kms: y kms:. ListAliases](#)

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. Para ver las claves de la cuenta que se AWS crean y administran para usted, en el panel de navegación, seleccione las claves AWS administradas. Si desea ver las claves de la cuenta que usted crea y administra, elija en el panel de navegación, Claves administradas por el cliente.
4. En la lista de claves KMS, elija el alias o ID de clave de la clave KMS que desea examinar.
5. Seleccione la pestaña Key policy (Política de claves).

En la pestaña Key policy (Política de claves), es posible que vea el documento de política de claves. Esta es la vista de política. En las declaraciones de políticas de claves, puede ver

las entidades principales a las que la política de claves ha dado acceso a la clave KMS y las acciones que pueden realizar.

En el ejemplo siguiente se muestra la vista de política de la [política de claves predeterminada](#).



O bien, si creó la clave KMS en Consola de administración de AWS, verá la vista predeterminada con secciones para los administradores de claves, la eliminación de claves y los usuarios clave. Para ver el documento de políticas de claves, elija Switch to policy view (Cambiar a la vista de política).

En el ejemplo siguiente se muestra la vista predeterminada de la [política de claves predeterminada](#).

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			

Uso de la AWS KMS API

Para obtener la política de claves de una clave de KMS Cuenta de AWS, usa la [GetKeyPolicy](#) operación de la AWS KMS API. No puede utilizar esta operación para ver una política de claves en una cuenta distinta.

En el siguiente ejemplo, se usa el [get-key-policy](#) comando de AWS Command Line Interface (AWS CLI), pero puedes usar cualquier AWS SDK para realizar esta solicitud.

Tenga en cuenta que el parámetro `PolicyName` es obligatorio aunque `default` sea su único valor válido. Además, este comando solicita la salida en texto, en lugar de en JSON, para que sea más fácil de ver.

Antes de ejecutar este comando, reemplace el ID de clave de ejemplo por uno válido de su cuenta.

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

La respuesta debe ser similar a la siguiente, que devuelve la [política de claves predeterminada](#).

JSON

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

Cambio de una política de claves

Puede cambiar la política de claves de una clave de KMS suya Cuenta de AWS mediante la [PutKeyPolicy](#) operación Consola de administración de AWS o. No puede utilizar estas técnicas para cambiar la política de claves de una clave KMS en una cuenta de Cuenta de AWS distinta.

Cuando cambie una política de claves, tenga en cuenta las siguientes reglas:

- Puede ver la política de claves para una [Clave administrada de AWS](#) o una [clave administrada por el cliente](#), pero solo puede cambiar la política de claves para una clave KMS administrada por el cliente. Claves administradas por AWS Las políticas de las crea y administra el AWS servicio

que creó la clave KMS en su cuenta. No puede ver ni modificar la política de claves de una [Clave propiedad de AWS](#).

- Puede añadir o eliminar usuarios de IAM, funciones de IAM y Cuentas de AWS en la política clave, así como cambiar las acciones que se permiten o deniegan a esos responsables. Para obtener más información sobre las formas de especificar entidades principales y permisos en una política de claves, consulte [Políticas de claves](#).
- No puede agregar grupos de IAM a una política de claves, aunque puede agregar varios usuarios de IAM y roles de IAM. Para obtener más información, consulte [Conceder permiso a varias entidades principales de IAM para acceder a una clave KMS](#).
- Si añades una política externa Cuentas de AWS a una política clave, también debes usar políticas de IAM en las cuentas externas para conceder permisos a los usuarios, grupos o roles de IAM en esas cuentas. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).
- El documento de política de claves resultante no puede superar los 32 KB (32 768 bytes).

Cómo cambiar una política de claves

Puede cambiar una política de claves de tres formas diferentes, tal como se explica en las siguientes secciones.

Temas

- [Uso de la vista predeterminada Consola de administración de AWS](#)
- [Uso de la vista Consola de administración de AWS de políticas](#)
- [Uso de la AWS KMS API](#)

Uso de la vista predeterminada Consola de administración de AWS

Puede utilizar la consola para cambiar una política de claves con una interfaz gráfica denominada la vista predeterminada.

Si estos pasos no se corresponden con lo que aparece en la consola, puede significar que la consola no ha creado esta política de claves. O bien que la política de claves se ha modificado de un modo que no admite la vista predeterminada de la consola. En ese caso, siga los pasos de [Uso de la vista Consola de administración de AWS de políticas](#) o [Uso de la AWS KMS API](#).

1. Vea la política de claves para una clave administrada por el cliente tal y como se indica en [Uso de la AWS KMS consola](#). (No puede cambiar las políticas clave de Claves administradas por AWS.)
2. Decida lo que desea cambiar.
 - Para agregar o eliminar [administradores de claves](#) y para permitir o evitar que los administradores de claves [eliminen la clave KMS](#), utilice los controles de la sección Key administrators (Administradores de claves) de la página. Los administradores de claves administran la clave KMS, incluida su activación y desactivación, estableciendo la política de claves y [habilitando la rotación de claves](#).
 - Para añadir o eliminar [usuarios clave](#) y permitir o impedir que un usuario externo Cuentas de AWS utilice la clave KMS, utilice los controles de la sección Usuarios clave de la página. Los usuarios de claves pueden usar la clave KMS en [operaciones criptográficas](#), como cifrar, descifrar, volver a cifrar y generar claves de datos.

Uso de la vista Consola de administración de AWS de políticas

Puede utilizar la consola para cambiar un documento de política de claves con la vista de políticas de la consola.

1. Vea la política de claves para una clave administrada por el cliente tal y como se indica en [Uso de la AWS KMS consola](#). (No puede cambiar las políticas clave de Claves administradas por AWS.)
2. En la sección Política de claves, elija Cambiar a la vista de política.
3. Elija Edit (Edición de).
4. Decida lo que desea cambiar.
 - Para agregar una nueva declaración, seleccione Agregar nueva declaración. A continuación, puede seleccionar las acciones, las entidades principales y las condiciones de su nueva declaración de política de claves entre las opciones que se muestran en el panel del creador de declaraciones, o bien introducir manualmente los elementos de la declaración de política.
 - Para eliminar una declaración de su política de claves, selecciónela y, a continuación, elija Eliminar. Revise la declaración de política seleccionada y confirme que desea eliminarla. Si decide que no desea continuar con la eliminación de la declaración, seleccione Cancelar.

- Para editar una declaración de política de claves existente, selecciónela. A continuación, puede utilizar el panel generador de declaraciones para elegir los elementos específicos que desee modificar o editar de manera manual la declaración.

5. Seleccione Save changes (Guardar cambios).

Uso de la AWS KMS API

Puede utilizar la [PutKeyPolicy](#) operación para cambiar la política de claves de una clave de KMS en su Cuenta de AWS. No puede utilizar esta API en una clave KMS en una cuenta de Cuenta de AWS diferente.

1. Utilice la [GetKeyPolicy](#) operación para obtener el documento de política clave existente y, a continuación, guárdelo en un archivo. Para obtener un código de ejemplo en varios lenguajes de programación, consulte [Úselo GetKeyPolicy con un AWS SDK o CLI](#).
2. Abra el documento de políticas de claves en el editor de textos que prefiera, edítelo y, a continuación, guarde el archivo.
3. Utilice la [PutKeyPolicy](#) operación para aplicar el documento de política clave actualizado a la clave de KMS. Para obtener un código de ejemplo en varios lenguajes de programación, consulte [Úselo PutKeyPolicy con un AWS SDK o CLI](#).

Para ver un ejemplo de cómo copiar una política clave de una clave de KMS a otra, consulte el [GetKeyPolicy ejemplo](#) en la Referencia de AWS CLI comandos.

Permisos para AWS los servicios en las políticas clave

Muchos AWS servicios se utilizan AWS KMS keys para proteger los recursos que administran. Cuando un servicio usa [Claves propiedad de AWS](#) o [Claves administradas por AWS](#), el servicio establece y mantiene las políticas de claves para estas claves KMS.

Sin embargo, cuando usa una [clave administrada personalizada](#) con un servicio de AWS , establece y mantiene la política de claves. Esta política de claves debe permitir al servicio los permisos mínimos que necesita para proteger el recurso en su nombre. Le recomendamos que siga el principio de privilegio mínimo: otorgar al servicio solo los permisos que requiere. Puede hacerlo de manera eficaz aprendiendo qué permisos necesita el servicio y utilizando [claves de condición globales de AWS](#) y [claves de condición de AWS KMS](#) para refinar los permisos.

Para encontrar los permisos que requiere el servicio en una clave administrada por el cliente, consulte la documentación de cifrado del servicio. La siguiente lista incluye enlaces a la documentación de algunos servicios:

- AWS CloudTrailpermisos: [configure políticas AWS KMS clave para CloudTrail](#)
- Permisos de Amazon Elastic Block Store: guía del usuario de [Amazon EC2](#) y [Guía del usuario de Amazon EC2](#)
- Permisos de AWS Lambda: [cifrado de datos en reposo para Lambda](#)
- Permisos de Amazon Q: [cifrado de datos para Amazon Q](#)
- Permisos de servicio de base de datos relacional de Amazon: [administración de claves de AWS KMS](#)
- Permisos de AWS Secrets Manager: [autoriza el uso de la clave de KMS](#)
- Permisos de Amazon Simple Queue Service: [administración de claves de Amazon SQS](#)

Uso de políticas de IAM con AWS KMS

Puedes usar las políticas de IAM, junto con las [políticas clave](#), las [subvenciones](#) y las políticas de puntos finales de [VPC](#), para controlar el acceso a AWS KMS keys tu entrada. AWS KMS

Note

Para utilizar una política de IAM a fin de controlar el acceso a una clave KMS, la política de claves de la clave KMS debe conceder permiso a la cuenta para utilizar políticas de IAM. En concreto, la política de claves debe incluir la [declaración de política que habilita las políticas de IAM](#).

En esta sección se explica cómo utilizar las políticas de IAM para controlar el acceso a las operaciones. AWS KMS Para obtener más información sobre IAM, consulte la [Guía del usuario de IAM](#).

Todas las claves KMS deben tener una política de claves. Las políticas de IAM son opcionales. Para utilizar una política de IAM a fin de controlar el acceso a una clave KMS, la política de claves de la clave KMS debe conceder permiso a la cuenta para utilizar políticas de IAM. En concreto, la política de claves debe incluir la [declaración de política que habilita las políticas de IAM](#).

Las políticas de IAM pueden controlar el acceso a cualquier AWS KMS operación. A diferencia de las políticas clave, las políticas de IAM pueden controlar el acceso a varias claves de KMS y proporcionar permisos para las operaciones de varios servicios relacionados AWS . Sin embargo, las políticas de IAM son especialmente útiles para controlar el acceso a las operaciones [CreateKey](#), por ejemplo, que no pueden controlarse mediante una política clave porque no implican ninguna clave de KMS en particular.

Si accede a AWS KMS través de un punto de enlace de Amazon Virtual Private Cloud (Amazon VPC), también puede utilizar una política de punto de enlace de VPC para limitar el acceso a sus AWS KMS recursos cuando utilice el punto de enlace. Por ejemplo, cuando utilices el punto final de la VPC, es posible que solo permitas que los principales de tu cuenta accedan Cuenta de AWS a las claves gestionadas por el cliente. Para obtener más información, consulte las [políticas de punto de conexión de VPC](#).

Para obtener ayuda sobre cómo escribir y dar formato a un documento de política JSON, consulte la [Referencia de políticas JSON de IAM](#) en la Guía del usuario de IAM.

Puede las políticas de IAM de las siguientes formas:

- Adjuntar una política de permisos a un rol para federación o permisos en varias cuentas: puede adjuntar una política de IAM a un rol de IAM para habilitar la identidad federada, permitir los permisos en varias cuentas o conceder permisos a las aplicaciones que se ejecutan en instancias EC2. Para obtener más información sobre los diferentes casos de uso para roles de IAM, consulte [Roles de IAM](#) en la Guía del usuario de IAM.
- Asociar una política de permisos a un usuario o grupo: puede adjuntar una política que permita a un usuario o grupo de usuarios llamar a las operaciones de AWS KMS . Sin embargo, las prácticas recomendadas de IAM recomiendan utilizar identidades con credenciales temporales, como roles de IAM, siempre que sea posible.

En el siguiente ejemplo, se muestra una política de IAM con permisos. AWS KMS Esta política permite a las identidades de IAM a las que está asociada obtener todas las claves KMS y alias.

JSON

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

```
"Action": [
    "kms:ListKeys",
    "kms:ListAliases"
],
"Resource": "*"
}
```

Al igual que todas las políticas de IAM, esta política no tiene ningún elemento `Principal`. Cuando asocia una política de IAM a una identidad de IAM, esa identidad obtiene los permisos especificados en la política.

Para ver una tabla en la que se muestran todas las acciones de la AWS KMS API y los recursos a los que se aplican, consulte la [Referencia de permisos](#).

Conceder permiso a varias entidades principales de IAM para acceder a una clave KMS

Los grupos de IAM no son entidades principales válidas en una política de claves. Para permitir que varios usuarios y roles obtengan acceso a una clave KMS, realice una de las acciones siguientes:

- Utilice un rol de IAM como entidad principal en la política clave. Varios usuarios autorizados pueden asumir el rol según sea necesario. Para obtener más información, consulte la sección [Roles de IAM](#) en la Guía del usuario de IAM.

Si bien puede incluir varios usuarios de IAM en una política clave, no se recomienda esta práctica porque requiere que actualice la política clave cada vez que cambia la lista de usuarios autorizados. Además, las mejores prácticas de IAM desaconsejan el uso de usuarios de IAM con credenciales a largo plazo. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;

- Utilice una política de IAM para conceder permisos a un grupo de IAM. Para ello, asegúrese de que la política de claves incluya la declaración que [permite a las políticas de IAM permitir el acceso a la clave KMS](#), [cree una política de IAM](#) que permita el acceso a la clave KMS y, a continuación, [adjunte dicha política a un grupo de IAM](#) que contenga los usuarios de IAM autorizados. Con este enfoque, no es necesario cambiar ninguna política cuando cambie la lista de usuarios autorizados. En su lugar, solo debe agregar o eliminar dichos usuarios del grupo de IAM apropiado. Para obtener más información, consulte los [grupos de usuarios de IAM](#) en la Guía del usuario de IAM

Para obtener más información sobre cómo funcionan juntas las políticas AWS KMS clave y las políticas de IAM, consulte [Solución de problemas de AWS KMS permisos](#).

Prácticas recomendadas para las políticas de IAM

Asegurar el acceso a AWS KMS keys es fundamental para la seguridad de todos sus AWS recursos. Las claves KMS se utilizan para proteger muchos de los recursos más sensibles de su Cuenta de AWS. Tómese el tiempo para diseñar las [políticas de claves](#), las políticas de IAM, las [concesiones](#) y las políticas de punto de conexión de VPC que controlan el acceso a sus claves KMS.

En las declaraciones de política de IAM que controlan el acceso a las claves KMS, utilice el [principio del menos privilegiado](#). Proporcione a las entidades principales de IAM solo los permisos que necesitan y únicamente en las claves de KMS que deben usar o administrar.

Las siguientes prácticas recomendadas se aplican a las políticas de IAM que controlan el acceso a AWS KMS las claves y los alias. Para obtener información general sobre las prácticas recomendadas de política de IAM, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

Use políticas de claves

Siempre que sea posible, proporcione permisos en las políticas de clave que afecten a una clave KMS, en lugar de en una política de IAM que se pueda aplicar a muchas claves KMS, incluidas las de otras Cuentas de AWS. Esto es especialmente importante para los permisos confidenciales, como [kms: PutKeyPolicy](#) y [kms: ScheduleKeyDeletion](#), pero también para las operaciones criptográficas que determinan cómo se protegen los datos.

Limite el permiso CreateKey

Conceda permiso para crear claves ([kms: CreateKey](#)) solo a los directores que lo necesiten. Las principales entidades que crean una clave KMS también establecen su política de claves, de modo que puedan concederse a sí mismos y a otros permisos para usar y administrar las claves KMS que crean. Cuando permita este permiso, considere limitarlo mediante el uso de [condiciones de política](#). Por ejemplo, puede usar la KeySpec condición [kms:](#) para limitar el permiso a las claves KMS de cifrado simétrico.

Especifique claves KMS en una política de IAM

Como práctica recomendada, especifique la [ARN de clave](#) de cada clave KMS a la que se aplica el permiso en el elemento Resource de la declaración de política. Esta práctica restringe

el permiso a las claves KMS que requiere la entidad principal. Por ejemplo, este elemento `Resource` muestra solo las claves KMS que la entidad principal necesita 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"  
]
```

Si no es práctico especificar las claves de KMS, utilice un `Resource` valor que Cuenta de AWS limite el acceso a las claves de KMS en una región de confianza, como.

`arn:aws:kms:region:account:key/*` O limite el acceso a las claves de KMS en todas las regiones (*) de una región de confianza Cuenta de AWS, por ejemplo.

`arn:aws:kms:*:account:key/*`

No puede utilizar un [ID de clave](#), [nombre de alias](#) o bien [ARN de alias](#) para representar una clave de KMS en el campo `Resource` de una política de IAM. Si especifica un alias ARN, la política se aplica al alias, no a la clave KMS. Para obtener información general sobre las políticas de IAM, consulte [Control del acceso a alias](#).

Evite "Recurso": "*" en una política de IAM

Utilice los caracteres de comodín (*) con juicio. En una política de claves, el carácter de comodín en el elemento `Resource` representa la clave KMS a la que se adjunta la política de clave. Sin embargo, en una política de IAM, un carácter comodín solo en el `Resource` elemento ("`Resource`": "*") aplica los permisos a todas las claves de KMS en todos los casos en los Cuentas de AWS que la cuenta del principal tenga permiso de uso. Esto puede incluir [las claves de KMS en otras cuentas Cuentas de AWS](#), así como las claves de KMS de la cuenta del principal.

Por ejemplo, para usar una clave de KMS en otra Cuenta de AWS, el principal necesita el permiso de la política de claves de la clave de KMS de la cuenta externa y de una política de IAM de su propia cuenta. Supongamos que una cuenta arbitraria le dio a su Cuenta de AWS el permiso [kms:Decrypt](#) en sus claves KMS. Si es así, una política de IAM en su cuenta que le asigne permiso `kms:Decrypt` a un rol en todas las claves KMS ("`Resource`": "*") satisfaría la parte de IAM del requisito. Como resultado, las principales entidades que pueden asumir ese rol ahora pueden descifrar textos cifrados utilizando la clave KMS en la cuenta que no es de confianza. Las entradas de sus operaciones aparecen en los CloudTrail registros de ambas cuentas.

En particular, evite usar "Resource": "*" en una declaración de política que permita las siguientes operaciones de API. Estas operaciones se pueden realizar en las claves de KMS de otras Cuentas de AWS.

- [DescribeKey](#)
- [GetKeyRotationStatus](#)
- [Operaciones criptográficas \(cifrar, descifrar,,,,, GenerateDataKey,, GenerateDataKeyPair, GenerateDataKeyWithoutPlaintext, GenerateDataKeyPairWithoutPlaintext, firmar GetPublicKeyReEncrypt, verificar\)](#)
- [CreateGrant](#), [ListGrants](#), [ListRetirableGrants](#), [RetireGrant](#), [RevokeGrant](#)

Cuándo usar "Recurso": "*"

En una política de IAM, utilice un carácter comodín en el elemento Resource solo para los permisos que lo requieran. Solo los siguientes permisos requieren el elemento "Resource":
" * ".

- [kilómetros: CreateKey](#)
- [km: GenerateRandom](#)
- [km: ListAliases](#)
- [km: ListKeys](#)
- Permisos para almacenes de claves personalizados, como [kms: CreateCustomKeyStore](#) y [kms: ConnectCustomKeyStore](#).

 Note

Los permisos para las operaciones de alias ([kms: CreateAlias](#), [kms: UpdateAlias](#), [kms: DeleteAlias](#)) deben adjuntarse al alias y a la clave KMS. Puede utilizar "Resource": "*" en una política de IAM para representar los alias y las claves KMS, o especificar los alias y las claves KMS en el elemento Resource. Para ver ejemplos, consulte [Control del acceso a alias](#).

Los ejemplos de este tema proporcionan más información y orientación para diseñar políticas de IAM para claves KMS. Para conocer las prácticas recomendadas de IAM para todos los AWS recursos, consulte [las prácticas recomendadas de seguridad en IAM en](#) la Guía del usuario de IAM.

Especificación de claves KMS en declaraciones de políticas de IAM

Puede utilizar una política de IAM para permitir que una entidad principal utilice o administre claves KMS. Las claves KMS se especifican en el elemento Resource de la declaración de política.

- Para especificar una clave KMS en una declaración de política de IAM, debe utilizar su [ARN de clave](#). No puede utilizar un [ID de clave](#), un [nombre de alias](#) ni un [ARN de alias](#) para identificar una clave KMS en una declaración de política de IAM.

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

Para controlar el acceso a una clave de KMS en función de sus alias, utilice las claves de condición [kms: RequestAlias](#) o [kms: ResourceAliases](#). Para obtener más información, consulte [ABAC para AWS KMS](#).

Utilice un ARN de alias como recurso solo en una declaración de política que controle el acceso a las operaciones de alias, como [CreateAliasUpdateAlias](#), o. [DeleteAlias](#). Para obtener más información, consulte [Control del acceso a alias](#).

- Para especificar varias claves KMS en la cuenta y la región, utilice caracteres de comodín (*) en las posiciones de ID de región o recurso del ARN de clave.

Por ejemplo, para especificar todas las claves KMS de la región EE. UU. Oeste (Oregón) de una cuenta, utilice "Resource": "arn:aws:kms:us-west-2:111122223333:key/*".

Para especificar todas las claves KMS en todas las regiones de la cuenta, utilice "Resource": "arn:aws:kms:*:111122223333:key/*".

- Para representar todas las claves KMS, utilice un carácter de comodín solo ("*"). Utilice este formato para las operaciones que no utilicen ninguna clave de KMS concreta, es decir [CreateKey](#), [GenerateRandomListAliases](#), y [ListKeys](#).

Al escribir sus declaraciones de política, se trata de una [práctica recomendada](#) para especificar solo las claves KMS que la entidad principal necesita usar, en lugar de darles acceso a todas las claves KMS.

Por ejemplo, la siguiente declaración de política de IAM permite al director llamar a las operaciones [DescribeKey](#), [GenerateDataKey](#), [Decrypt](#) únicamente con las claves de KMS que figuran en el Resource elemento de la declaración de política. La especificación de claves KMS por ARN clave,

que es una práctica recomendada, garantiza que los permisos estén limitados únicamente a las claves 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 el permiso a todas las claves de KMS de una entidad de confianza concreta Cuenta de AWS, puede utilizar caracteres comodín (*) en las posiciones de la región y de los ID de las claves. Por ejemplo, la siguiente declaración de política permite a la entidad principal llamar a las operaciones especificadas en cualquier clave KMS de las dos cuentas de ejemplo de confianza.

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/*"
]
}
}
```

También puede utilizar un carácter comodín ("*") solo en el elemento `Resource`. Debido a que permite el acceso a todas las claves KMS que la cuenta tiene permiso para usar, se recomienda principalmente para operaciones sin una clave KMS concreta y declaraciones `Deny`. También puede utilizarlo en declaraciones de política que solo permiten operaciones de solo lectura menos sensibles. Para determinar si una AWS KMS operación implica una clave de KMS concreta, busque el valor de la clave de KMS en la columna Recursos de la tabla de [the section called "Referencia de permisos"](#)

Por ejemplo, la siguiente declaración de política utiliza un efecto `Deny` para prohibir que las principales entidades utilicen las operaciones especificadas en cualquier clave KMS. Utiliza un carácter comodín en el elemento `Resource` para representar todas las claves KMS.

JSON

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

La siguiente declaración de política utiliza un carácter comodín solo para representar todas las claves KMS. Pero solo permite operaciones de solo lectura menos sensibles y operaciones que no se aplican a ninguna clave KMS en particular.

JSON

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

Ejemplos de políticas de IAM

En esta sección, encontrará ejemplos de políticas de IAM que conceden permisos para varias acciones de AWS KMS .

Important

Algunos de los permisos de las siguientes políticas solo se permiten cuando la política de claves de la clave KMS también los permite. Para obtener más información, consulte [Referencia de permisos](#).

Para obtener ayuda sobre cómo escribir y dar formato a un documento de política JSON, consulte la [Referencia de políticas JSON de IAM](#) en la Guía del usuario de IAM.

Ejemplos

- [Permita que un usuario vea las claves de KMS en la AWS KMS consola](#)
- [Permitir a un usuario crear claves KMS](#)
- [Permita que un usuario cifre y descifre con cualquier clave KMS de una clave específica Cuenta de AWS](#)
- [Permite a un usuario cifrar y descifrar con cualquier clave de KMS de una región específica Cuenta de AWS](#)

- [Permitir a un usuario cifrar y descifrar con claves KMS específicas](#)
- [Impedir a un usuario desactivar o eliminar cualquier clave KMS](#)

Permita que un usuario vea las claves de KMS en la AWS KMS consola

La siguiente política de IAM permite a los usuarios el acceso de solo lectura a la consola. AWS KMS Los usuarios con estos permisos pueden ver todas las claves de KMS que contienen Cuenta de AWS, pero no pueden crear ni cambiar ninguna clave de KMS.

[Para ver las claves de KMS en las páginas de claves administradas por el cliente Claves administradas por AWS y las claves administradas por el cliente](#) `ListKeys`, los directores necesitan `GetResources` los permisos `kms: ListAliases`, `kms: y tag:`, incluso si las claves no tienen etiquetas `ni alias`. Los permisos restantes, especialmente `kms: DescribeKey`, son necesarios para ver las columnas y los datos opcionales de la tabla de claves de KMS en las páginas de detalles clave de KMS. Los `ListRoles` permisos `iam: ListUsers` e `iam:` son necesarios para mostrar la política clave en la vista predeterminada sin errores. Para ver los datos de la página de almacenes de claves personalizados y los detalles sobre las claves de KMS en los almacenes de claves personalizados, los directores también necesitan el permiso `kms: DescribeCustomKeyStores`

Si limita el acceso de la consola de un usuario a determinadas claves KMS, la consola muestra un error para cada clave KMS que no está visible.

Esta política incluye dos declaraciones de política. El elemento `Resource` en la primera declaración de política permite los permisos especificados en todas las claves KMS en todas las regiones del ejemplo Cuenta de AWS. Los lectores de la consola no necesitan acceso adicional porque la consola AWS KMS muestra solo las claves KMS en la cuenta de la entidad principal. Esto es cierto incluso si tienen permiso para ver las claves de KMS en otras Cuentas de AWS. Los permisos restantes AWS KMS y de IAM requieren un `"Resource": "*"` elemento porque no se aplican a ninguna clave de KMS en particular.

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 a un usuario crear claves KMS

La siguiente política de IAM permite a un usuario crear todos los tipos de claves KMS. El valor del Resource elemento se * debe a que la CreateKey operación no utiliza ningún AWS KMS recurso concreto (claves o alias de KMS).

[Para restringir al usuario a determinados tipos de claves de KMS, utilice las claves de KeyOrigin condición kms: KeyUsage, kms: y kms:. KeySpec](#)

JSON

```

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

```



```
}
}
```

Es posible que las principales entidades que crean claves necesiten algunos permisos relacionados.

- **kms: PutKeyPolicy** — Los directores que tengan `kms:CreateKey` permiso pueden establecer la política de claves inicial para la clave de KMS. Sin embargo, la `CreateKey` persona que llama debe tener el `PutKeyPolicy` permiso [kms:](#), que le permite cambiar la política de claves de KMS, o debe especificar el `BypassPolicyLockoutSafetyCheck` parámetro de `CreateKey`, lo cual no es recomendable. La persona que llama `CreateKey` puede obtener permiso `kms:PutKeyPolicy` para la clave KMS desde una política de IAM o pueden incluir este permiso en la política de claves de la clave KMS que están creando.
- **kms: TagResource** — Para añadir etiquetas a la clave KMS durante la `CreateKey` operación, la persona que `CreateKey` llama debe tener el `TagResource` permiso [kms:](#) en una política de IAM. Incluir este permiso en la política de claves de la nueva clave KMS no es suficiente. Sin embargo, si la persona que llama a `CreateKey` incluye `kms:TagResource` en la política de clave inicial, pueden agregar etiquetas en una llamada separada después de crear la clave KMS.
- **kms: CreateAlias** — Los responsables que creen una clave KMS en la AWS KMS consola deben tener el `CreateAlias` permiso [kms:](#) en la clave KMS y en el alias. (La consola realiza dos llamadas; una a `CreateKey` y una a `CreateAlias`). Debe proporcionar el permiso de alias en una política de IAM. Puede proporcionar estos permisos en una política de claves, una política de IAM o una concesión. Para obtener más información, consulte [Control del acceso a alias](#).

Además `kms:CreateKey`, la siguiente política de IAM otorga `kms:TagResource` permisos a todas las claves de KMS de la cuenta Cuenta de AWS y `kms:CreateAlias` a todos los alias de la cuenta. También incluye algunos permisos de solo lectura útiles que únicamente se pueden proporcionar en una política de IAM.

Esta política de IAM no incluye permiso `kms:PutKeyPolicy` ni ningún otro permiso que se pueda establecer en una política de clave. Es una [práctica recomendada](#) para establecer estos permisos en la política de claves donde se aplican exclusivamente a una clave 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": "*"
  }
]
```

Permita que un usuario cifre y descifre con cualquier clave KMS de una clave específica Cuenta de AWS

La siguiente política de IAM permite a los usuarios cifrar y descifrar datos con cualquier clave de KMS en 111122223333. Cuenta de AWS

JSON

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

```
    ],  
    "Resource": "arn:aws:kms:*:111122223333:key/*"  
  }  
}
```

Permite a un usuario cifrar y descifrar con cualquier clave de KMS de una región específica Cuenta de AWS

La siguiente política de IAM permite a los usuarios cifrar y descifrar datos con cualquier clave KMS Cuenta de AWS 111122223333 en la región EE.UU. Oeste (Oregón).

JSON

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

Permitir a un usuario cifrar y descifrar con claves KMS específicas

La siguiente política de IAM permite a un usuario cifrar y descifrar datos con las dos claves KMS especificadas en el elemento Resource. Al especificar una clave KMS en una declaración de política de IAM, debe utilizar la [ARN de clave](#) de la clave KMS.

JSON

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

```
"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 a un usuario desactivar o eliminar cualquier clave KMS

La siguiente política de IAM impide que un usuario deshabilite o elimine cualquier clave KMS, aunque otra política de IAM o política de claves conceda estos permisos. Una política que deniega de forma explícita los permisos anula todas las demás políticas, incluso aquellas que hayan permitido explícitamente los mismos permisos. Para obtener más información, consulte [Solución de problemas de AWS KMS permisos](#).

JSON

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

Políticas de control de recursos en AWS KMS

Las políticas de control de recursos (RCPs) son un tipo de política organizacional que puede usar para aplicar controles preventivos a AWS los recursos de su organización. RCPs le ayudan a restringir de forma centralizada y a gran escala el acceso externo a sus AWS recursos. RCPs complementar las políticas de control de servicios (SCPs). Si bien se SCPs puede utilizar para establecer de forma centralizada los permisos máximos para las funciones y los usuarios de IAM de su organización, se RCPs puede utilizar para establecer de forma centralizada los permisos máximos sobre los AWS recursos de su organización.

Puede usarlo RCPs para administrar los permisos de las claves de KMS administradas por el cliente en su organización. RCPs por sí solos no son suficientes para conceder permisos a las claves administradas por el cliente. Una RCP no concede ningún permiso. Una RCP define una barrera de protección de permisos o establece límites a las acciones que las identidades pueden llevar a cabo en los recursos de las cuentas afectadas. El administrador aún debe adjuntar políticas basadas en identidad a los usuarios o roles de IAM, o políticas de claves para conceder permisos realmente.

Note

Las políticas de control de recursos de su organización no se aplican a las [Claves administradas por AWS](#).

Claves administradas por AWS si son creadas, administradas y utilizadas en su nombre por un AWS servicio, no puede cambiar ni administrar sus permisos.

Más información

- Para obtener más información general RCPs, consulte [las políticas de control de recursos](#) en la Guía del AWS Organizations usuario.
- Para obtener detalles sobre cómo definir RCPs, incluidos ejemplos, consulte la [sintaxis de RCP](#) en la Guía del AWS Organizations usuario.

En el siguiente ejemplo, se muestra cómo utilizar una RCP para evitar que las entidades principales externas tengan acceso a las claves administradas por los clientes de su organización. Esta política es solo un ejemplo y debe modificarla para que se adapte a sus necesidades empresariales y de seguridad específicas. Por ejemplo, es posible que desee personalizar la política para permitir el

acceso de sus socios comerciales. Para obtener más información, consulte el [repositorio de ejemplos de políticas de perímetro de datos](#).

Note

El permiso `kms:RetireGrant` no es efectivo en una RCP, incluso si el elemento `Action` especifica un asterisco (*) como comodín.

Para obtener más información sobre cómo se determina el permiso para `kms:RetireGrant`, consulte [Retiro y revocación de concesiones](#).

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"
        }
      }
    }
  ]
}
```

El siguiente ejemplo de RCP exige que los directores de AWS servicio solo puedan acceder a las claves de KMS administradas por el cliente cuando la solicitud se origine en su organización. Esta política aplica el control solo a las solicitudes que tienen `aws:SourceAccount` presente. Esto garantiza que las integraciones de servicios que no requieren el uso de `aws:SourceAccount` no se

vean afectadas. Si la `aws:SourceAccount` está presente en el contexto de la solicitud, se evaluará la condición `Null` como `true`, lo que provocará que se aplique la clave `aws:SourceOrgID`.

Para obtener información general sobre el problema del suplente confuso, consulte [El problema del suplente confuso](#) en la Guía del usuario de 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"
        }
      }
    }
  ]
}
```

Subvenciones en AWS KMS

Una subvención es un instrumento de política que permite a [AWS los directores](#) o [directores AWS de servicio](#) utilizar claves KMS en operaciones criptográficas. También puede permitirles ver una clave KMS (`DescribeKey`) y crear y administrar concesiones. Al autorizar el acceso a una clave KMS, se consideran concesiones junto con [políticas de claves](#) y [políticas de IAM](#). Las concesiones se utilizan

a menudo para permisos temporales, ya que puede crear uno, utilizar sus permisos y eliminarlo sin cambiar las políticas de claves o las políticas de IAM.

Los AWS servicios que se integran para cifrar los datos en reposo suelen utilizar AWS KMS las subvenciones. El servicio crea una concesión en nombre de un usuario de la cuenta, utiliza sus permisos y retira la concesión tan pronto como finalice su tarea. Para obtener más información sobre cómo AWS los servicios utilizan las subvenciones, consulta el tema El cifrado en reposo en la guía del usuario o en la guía para desarrolladores del servicio.

Las concesiones son un mecanismo de control de acceso muy flexible y útil. Al crear una concesión para una clave de KMS, la concesión permite a las entidades principales beneficiarias llamar a las operaciones de concesión especificadas en la clave de KMS siempre que se cumplan todas las condiciones especificadas en la concesión.

- Cada concesión permite el acceso a exactamente una clave KMS. Puede crear una concesión para una clave KMS en una Cuenta de AWS diferente.
- Una concesión puede permitir el acceso a una clave KMS, pero no denegar el acceso.
- Cada subvención tiene un beneficiario, que puede ser un director del mismo (una identidad de AWS IAM) o un [director](#) de servicio del [concesionario \(un director de servicio\)](#). AWS Debe especificar `GranteePrincipal` o `GranteeServicePrincipal`, pero no ambos.
- Una concesión solo puede permitir [operaciones de concesión](#). Las operaciones de concesión deben estar respaldadas por la clave KMS de la concesión. Si especificas una operación no admitida, la solicitud fallará con una excepción. `CreateGrantValidationError`
- La entidad principal beneficiaria puede utilizar los permisos que la concesión le otorga sin especificar la concesión, tal como lo haría si los permisos procedieran de una política de clave o de una política de IAM. Sin embargo, dado que la AWS KMS API sigue un modelo de [coherencia eventual](#), al crear, retirar o revocar una subvención, es posible que se produzca un breve retraso antes de que el cambio esté disponible de forma definitiva. AWS KMS Para utilizar los permisos de una concesión inmediatamente, [use un token de concesión](#).
- Un beneficiario principal autorizado puede eliminar la concesión ([retirarla](#) o [revocarla](#)). Al eliminar una concesión se eliminan todos los permisos permitidos por la concesión. No es necesario averiguar qué políticas se deben agregar o quitar para deshacer la concesión.
- AWS KMS limita el número de concesiones en cada clave de KMS. Para obtener más información, consulte [Concesiones por clave KMS: 50 000](#).

Tenga cuidado al crear concesiones y al dar permiso a otros para crear concesiones. El permiso para crear subvenciones tiene implicaciones de seguridad, al igual que permitir el PutKeyPolicy permiso de [KMS](#) para establecer políticas.

- Los usuarios con permiso para crear concesiones para una clave de KMS (`kms:CreateGrant`) pueden usar una concesión para permitir que los usuarios y los roles, incluidos los AWS servicios, usen la clave de KMS. Los elementos principales pueden ser identidades propias Cuenta de AWS o identidades de una cuenta u organización diferente.
- Las subvenciones solo permiten un subconjunto de AWS KMS operaciones. Puede utilizar concesiones para permitir a los principales beneficiarios ver la clave KMS, utilizarla en operaciones criptográficas y crear y retirar concesiones. Para obtener más información, consulte [Operaciones de concesión](#). También puede utilizar [restricciones de la concesión](#) para limitar los permisos de una concesión.
- Las entidades principales pueden obtener permiso para crear concesiones a partir de una política de claves o de una política de IAM. Las entidades principales que reciben el permiso `kms:CreateGrant` de una política puede crear concesiones para cualquier [operación de concesión](#) en la clave KMS. Estas entidades principales no necesitan tener el permiso que conceden en la clave. Cuando se permite permiso `kms:CreateGrant` en una política, puede usar [condiciones de política](#) para limitar este permiso.
- Los principales beneficiarios también pueden obtener permiso para crear concesiones a partir de una concesión. Estas entidades principales solo pueden delegar los permisos que se les concedieron, incluso si tienen otros permisos de una política. Para obtener más información, consulte [Concesión CreateGrant del permiso](#).

Conceptos de concesión

Para utilizar las concesiones de manera efectiva, deberá comprender los términos y conceptos que AWS KMS utiliza.

Restricción de concesiones

Una condición que limita los permisos de una concesión. AWS KMS admite dos tipos de restricciones de concesión:

- Restricción del contexto de cifrado: restringe los permisos en función del [contexto de cifrado](#) incluido en la solicitud de una operación criptográfica. Esta restricción solo funciona con claves KMS de cifrado simétrico.

- **SourceArn restricción:** restringe la concesión de permisos a las solicitudes realizadas en nombre de un recurso específico. AWS De hecho, se trata de incluir una clave de condición [aws:SourceArn](#) global en la concesión. Esta restricción funciona con todos los tipos de claves KMS.

Para obtener más información, consulte [Uso de restricciones de concesiones](#).

ID de concesión

El identificador único de una concesión de una clave KMS. Puede usar un identificador de concesión, junto con un [identificador clave](#), para identificar una concesión en una [RevokeGrants](#) solicitud [RetireGrants](#) solicitud.

Operaciones de concesión

Las AWS KMS operaciones que puedes permitir en una subvención. Si especificas otras operaciones, la [CreateGrants](#) solicitud fallará con una `ValidationError` excepción. Estas son también las operaciones que aceptan un [token de concesión](#). Para obtener información detallada sobre estos permisos, consulte la [AWS KMS permisos](#).

Estas operaciones de concesión representan realmente permiso para usar la operación. Por lo tanto, para la operación `ReEncrypt`, puede especificar `ReEncryptFrom`, `ReEncryptTo` o ambos `ReEncrypt*`.

Las operaciones de concesión son:

- Operaciones criptográficas
 - [Decrypt](#)
 - [DeriveSharedSecret](#)
 - [Encrypt](#)
 - [GenerateDataKey](#)
 - [GenerateDataKeyPair](#)
 - [GenerateDataKeyPairWithoutPlaintext](#)
 - [GenerateDataKeyWithoutPlaintext](#)
 - [GenerateMac](#)
 - [ReEncryptFrom](#)
 - [ReEncryptTo](#)
 - [Sign](#)

- [Verificar](#)
- [VerifyMac](#)
- Otras operaciones
 - [CreateGrant](#)
 - [DescribeKey](#)
 - [GetPublicKey](#)
 - [RetireGrant](#)

Las operaciones de concesión que usted permita deben ser compatibles con la clave KMS de la concesión. Si especificas una operación no admitida, la [CreateGrant](#) solicitud fallará con una `ValidationError` excepción. Por ejemplo, las concesiones para claves KMS de cifrado simétricas no pueden permitir las operaciones [Firmar](#), [Verificar](#), [GenerateMac](#) o [VerifyMac](#). Las concesiones para claves CMK asimétricas no pueden permitir operaciones que generen claves de datos o pares de claves de datos.

Token de concesión

La AWS KMS API sigue un modelo de [coherencia eventual](#). Al crear una concesión, es posible que haya un breve retraso antes de que el cambio esté disponible a través de AWS KMS. Por lo general, el cambio tarda menos de unos segundos en propagarse por todo el sistema, pero en algunos casos puede tardar varios minutos. Si intenta utilizar una concesión antes de que se propague por completo por el sistema, es posible que obtenga un error de acceso denegado. Un token de concesión le permite hacer referencia a la concesión y utilizar los permisos de concesión inmediatamente.

Un token de concesión es una cadena única, no secreta, de longitud variable, codificada en base64 que representa una concesión. Puede usar el token de concesión para identificar la concesión en cualquier [operación de concesión](#). Sin embargo, debido a que el valor del token es un resumen hash, no revela ningún detalle sobre la concesión.

Un token de concesión está diseñado para utilizarse solo hasta que la concesión se haya propagado por completo a través de AWS KMS. Después de eso, el [principal beneficiario](#) puede utilizar el permiso de la concesión sin proporcionar un token de concesión ni cualquier otra prueba de la concesión. Puedes usar un token de concesión en cualquier momento, pero una vez que la concesión sea coherente, AWS KMS utiliza la concesión para determinar los permisos, no el token de concesión.

Por ejemplo, el siguiente comando llama a la [GenerateDataKey](#) operación. Utiliza un token de concesión para representar la concesión que da a la persona que llama (el principal beneficiario) permiso para llamar a `GenerateDataKey` en la clave KMS especificada.

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

También puede utilizar un token de concesión para identificar una concesión en operaciones que administran concesiones. Por ejemplo, el [director que se retira](#) puede usar un token de concesión en una llamada a la [RetireGrant](#) operación.

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

`CreateGrant` es la única operación que devuelve un token de concesión. No puede obtener un token de concesión de ninguna otra AWS KMS operación ni del [evento de CloudTrail registro](#) de la `CreateGrant` operación. Las [ListRetirableGrants](#) operaciones [ListGrants](#) devuelven el [ID de concesión](#), pero no un token de concesión.

Para obtener más información, consulte [Uso de un token de concesión](#).

Principal beneficiario

Un AWS principal (identidad de IAM) que obtiene los permisos especificados en la concesión.

El cesionario principal puede ser cualquier AWS principal, incluido un usuario Cuenta de AWS (root), un usuario de [IAM, un rol de IAM, un rol o usuario federado o un usuario con un rol](#) asumido. El principal del beneficiario puede estar en la misma cuenta que la clave KMS o en una cuenta diferente. Sin embargo, el principal beneficiario no puede ser una [entidad principal del servicio](#), un [grupo de IAM](#) o una [organización de AWS](#).

Para especificar el principal beneficiario, usa el parámetro de la solicitud. `GranteePrincipal` [CreateGrant](#)

Note

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que

proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;.

Director de servicio del concesionario

Un [director AWS de servicio](#) que obtiene los permisos especificados en la concesión. Para especificar el principal de servicio del concesionario, usa el `GranteeServicePrincipal` parámetro de la solicitud. [CreateGrant](#)

Al crear una subvención con un `GranteeServicePrincipal`, también debe incluir un. [SourceArn grant constraint](#) La `SourceArn` restricción garantiza que el director del servicio pueda usar la clave KMS solo cuando la solicitud se realice en nombre del AWS recurso especificado.

Al especificar `aGranteeServicePrincipal`, también debe especificar a [RetiringPrincipal](#) o [RetiringServicePrincipal](#).

Note

Debe especificar [GranteePrincipal](#) o [GranteeServicePrincipal](#), pero no ambos.

Retiro (de una concesión)

Termina una concesión. Retire una concesión cuando termine de usar los permisos.

La revocación y el retiro de una concesión eliminan la concesión. Sin embargo, el retiro se realiza por medio de una entidad principal especificada en la concesión. La revocación suele realizarla un administrador de claves. Para obtener más información, consulte [Retiro y revocación de concesiones](#).

Entidad principal que se va a dar de baja

Una entidad principal que puede [retirar una concesión](#). Puede especificar una entidad principal que se va a dar de baja en una concesión, pero no es necesario. El principal que se jubila puede ser cualquier AWS principal, incluidos los usuarios de IAM Cuentas de AWS, los roles de IAM, los usuarios federados y los usuarios con roles asumidos. La entidad principal que se va a dar de baja puede estar en la misma cuenta de que la clave KMS o en una cuenta diferente.

Para especificar el principal que se retira, utilice el parámetro de la `RetiringPrincipal` solicitud. [CreateGrant](#)

Note

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;.

Director de servicio que se jubila

Un [director AWS de servicio](#) que tiene permiso para [retirar una subvención](#). Para especificar el director del servicio que se retira, utilice el `RetiringServicePrincipal` parámetro de la `CreateGrants` solicitud.

Note

Puede especificar [RetiringPrincipal](#) o [RetiringServicePrincipal](#), pero no ambos.

Para obtener más información sobre quién puede retirar una subvención, consulte [Retiro y revocación de concesiones](#).

Revocación (de una concesión)

Termina una concesión. Puede revocar una concesión para denegar activamente los permisos que permite la concesión.

La revocación y el retiro de una concesión eliminan la concesión. Sin embargo, el retiro se realiza por medio de una entidad principal especificada en la concesión. La revocación suele realizarla un administrador de claves. Para obtener más información, consulte [Retiro y revocación de concesiones](#).

Consistencia final (para las concesiones)

La AWS KMS API sigue un modelo de [coherencia eventual](#). Al crear, retirar o revocar una concesión, es posible que haya un breve retraso antes de que el cambio esté disponible a través

de AWS KMS. Por lo general, el cambio tarda menos de unos segundos en propagarse por todo el sistema, pero en algunos casos puede tardar varios minutos.

Puede darse cuenta de este breve retraso si recibe errores inesperados. Por ejemplo, si intentas gestionar una nueva concesión o utilizas los permisos de una nueva concesión antes de que se conozca la concesión en su totalidad AWS KMS, es posible que se produzca un error de acceso denegado. Si retira o revoca una concesión, es posible que el principal beneficiario pueda seguir utilizando sus permisos durante un breve período hasta que la concesión se elimine por completo. La estrategia habitual consiste en volver a intentar la solicitud, y algunos AWS SDK incluyen una lógica de reintento y retroceso automáticos.

AWS KMS cuenta con funciones para mitigar este breve retraso.

- Para utilizar los permisos de una nueva concesión inmediatamente, utilice un [token de concesión](#). Puede usar un token de concesión para referirse a una concesión en cualquier [operación de concesión](#). Para obtener instrucciones, consulte [Uso de un token de concesión](#).
- La [CreateGrant](#) operación tiene un Name parámetro que impide que las operaciones de reintento generen concesiones duplicadas.

 Note

Los tokens de concesión reemplazan la validez de la concesión hasta que todos los puntos de conexión del servicio se hayan actualizado con el nuevo estado de concesión. En la mayoría de los casos, la consistencia final se logrará en cinco minutos.

Para obtener más información, consulte [Coherencia final de AWS KMS](#).

Mejores prácticas para AWS KMS subsidios

AWS KMS recomienda las siguientes prácticas recomendadas a la hora de crear, utilizar y gestionar las subvenciones.

- Limite los permisos de la concesión a los que requiera el concesionario. Utilice el principio de [acceso menos privilegiado](#). Sea lo más específico posible en todos los parámetros de la concesión:
- Especifique solo las [operaciones de subvención](#) que el concesionario necesita.

- [Usa un director específico del concesionario, como un rol de IAM, o un director de servicio específico del concesionario.](#)
- Utilice las [restricciones de subvención para restringir aún](#) más la subvención.
- Use el contexto de cifrado de [restricciones de concesión](#) para asegurarse de que las personas que llaman utilizan la clave KMS para el propósito previsto. Para obtener más información sobre cómo utilizar el contexto de cifrado en una solicitud para proteger sus datos, consulte [Cómo proteger la integridad de sus datos cifrados mediante el uso AWS Key Management Service y EncryptionContext](#) en el blog sobre AWS seguridad.

 Tip

Utilice la restricción de [EncryptionContextEqual](#)concesión siempre que sea posible. La restricción de [EncryptionContextSubset](#)concesión es más difícil de usar correctamente. Si necesita usarlo, lea detenidamente la documentación y pruebe la limitación de concesión para asegurarse de que funciona según lo previsto.

- Al crear subvenciones para AWS servicios compatibles, utilícelas [SourceArn grant constraint](#)para restringir la subvención a un recurso específico.
- Tenga cuidado con las subvenciones duplicadas. Las subvenciones duplicadas tienen los mismos parámetros, excepto el nombre de la subvención. La duplicación innecesaria puede hacer que alcance la [cuota clave de subvenciones por KMS](#). Para evitar duplicar concesiones al volver a intentar una solicitud CreateGrant, utilice el [parámetro Name](#). Para detectar concesiones duplicadas, utilice la [ListGrants](#)operación.

 Note

Algunos AWS servicios crean concesiones para distintos recursos que pueden parecer duplicados. Estas subvenciones tienen ciclos de vida vinculados a los diferentes recursos. Eliminar las subvenciones creadas por un AWS servicio puede ser perjudicial y requiere precauciones adicionales.

- Recuerde que las concesiones no caducan automáticamente. [Retire o revoque la concesiones](#) ni bien el permiso ya no sea necesario. Las concesiones que no se eliminan pueden crear un riesgo de seguridad para los recursos cifrados.

Control del acceso a las concesiones

Puede controlar el acceso a las operaciones que crean y administran concesiones en políticas de claves, políticas de IAM y concesiones. Las principales entidades que reciben el permiso `CreateGrant` de una concesión tienen [permisos de concesión más limitados](#).

Operación de la API	Política de claves o política de IAM	Concesión
CreateGrant	✓	✓
ListGrants	✓	-
ListRetirableGrants	✓	-
Retiro de concesiones	(Limitado. Consulte Retiro y revocación de concesiones)	✓
RevokeGrant	✓	-

Si utilizas una política clave o una política de IAM para controlar el acceso a las operaciones que crean y gestionan las subvenciones, puedes utilizar una o varias de las siguientes condiciones de la política para limitar el permiso. AWS KMS admite todas las siguientes claves de condición relacionadas con la concesión. Para obtener más detalles y ejemplos, consulte [AWS KMS claves de condición](#).

[km: GrantConstraintSourceArn](#)

Permite a los directores crear una subvención solo cuando la subvención incluye una `SourceArn` restricción con el valor especificado.

[km: GrantConstraintType](#)

Permite a las entidades principales crear una concesión solo cuando la concesión incluye la [restricción de concesiones](#) especificada.

[km: GranteePrincipal](#)

Permite a las entidades principales crear una concesión solo para el [beneficiario principal](#).

[km: GranteeServicePrincipal](#)

Permite a los directores crear una subvención solo cuando la subvención especifica un principal de servicio del concesionario en particular.

[km: GrantsForAWSResource](#)

Permite que los directores `CreateGrant` llamen o `RevokeGrant` solo cuando [un AWS servicio integrado AWS KMS envía](#) la solicitud en nombre del director. `ListGrants`

[km: GrantOperations](#)

Permite a las entidades principales crear una concesión, pero limita la concesión a las operaciones especificadas.

[km: RetiringPrincipal](#)

Permite a las entidades principales crear una concesión solo cuando la concesión especifica una [entidad principal que se retira](#) en particular.

[km: RetiringServicePrincipal](#)

Permite a los directores crear una subvención solo cuando la subvención especifica un director de servicio en particular que se jubila.

Creación de concesiones

Antes de crear una concesión, obtenga información sobre las opciones para personalizar su concesión. Puede usar restricciones de concesión para limitar los permisos de la concesión. Además, obtenga más información sobre la concesión de permiso `CreateGrant`. Las principales entidades que obtienen permiso para crear concesiones a partir de una concesión están limitadas en las concesiones que pueden crear.

Temas

- [Crear una concesión](#)
- [Concesión `CreateGrant` del permiso](#)

Crear una concesión

Para crear una subvención, llame a la [CreateGrant](#) operación. Especifique una clave de KMS, una lista de [las operaciones de subvención](#) permitidas y el director del [concesionario o el director](#) de servicio del [concesionario](#).

[Al especificar un cesionario principal, si lo desea, puede designar un principal que se jubile o un director de servicio que se jubile y utilizar el parámetro para definir las restricciones de las subvenciones. Constraints](#) [Al especificar un director de servicio cesionario, debe incluir una restricción de subvención. SourceArn](#) También debe especificar a o `aRetiringServicePrincipal`. `RetiringPrincipal`

Al crear, retirar o revocar una concesión, es posible que haya un breve retraso, normalmente menos de cinco minutos, antes de que el cambio esté disponible a través de AWS KMS. Para obtener más información, consulte [Eventual consistency \(for grants\)](#).

Por ejemplo, el siguiente comando `CreateGrant` crea una concesión que permite a los usuarios que están autorizados a asumir el rol `keyUserRole` para llamar a la operación [Decrypt](#) en la [clave KMS simétrica](#) especificada. La concesión utiliza el parámetro `RetiringPrincipal` para designar una entidad principal que puede retirar la concesión. También incluye una restricción de concesión que permite el permiso únicamente cuando el [contexto de cifrado](#) de la solicitud incluye `"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}
```

Si su código vuelve a intentar la operación `CreateGrant`, o utiliza una [AWS SDK que reintenta automáticamente las solicitudes](#), utilice el parámetro opcional [Name](#) (Nombre) para evitar la creación de concesiones duplicadas. Si AWS KMS recibe una `CreateGrant` solicitud de concesión con las mismas propiedades que una subvención existente, incluido el nombre, reconocerá la solicitud al volver a intentarlo y no creará una nueva concesión. No puede usar el valor `Name` para identificar la concesión en cualquier operación de AWS KMS .

Important

No incluya información confidencial en el nombre de la concesión. Puede aparecer en texto plano en CloudTrail los registros y otros resultados.

```
$ 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}
```

```
--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}
```

En [Úselo CreateGrant con un AWS SDK o CLI](#), puede consultar ejemplos de código que muestran cómo crear concesiones en varios lenguajes de programación.

Uso de restricciones de concesiones

Las [restricciones de concesión](#) establecen condiciones sobre los permisos que la concesión otorga a entidad principal beneficiaria. Las restricciones de concesión sustituyen a las [claves de condición](#) en una [política de claves](#) o una [política de IAM](#).

Important

No incluya información confidencial en este campo. Este campo puede mostrarse en texto plano en CloudTrail los registros y otros resultados.

AWS KMS admite dos tipos de restricciones de concesión: las restricciones del contexto de cifrado y la SourceArn restricción.

Restricciones del contexto de cifrado

Las restricciones del contexto de cifrado establecen los requisitos para el [contexto de cifrado](#) en una solicitud de una operación criptográfica. Cada valor de restricción de concesión de contexto de cifrado puede incluir hasta 8 pares de contextos de cifrado. El valor del contexto de cifrado de cada restricción no puede superar los 384 caracteres.

- Las restricciones de contexto de cifrado solo son válidas en una concesión para una clave KMS de cifrado simétrica. Las operaciones criptográficas con otras claves KMS no son compatibles con un contexto de cifrado.
- La restricción de contexto de cifrado se omite para las operaciones DescribeKey y RetireGrant. DescribeKey y RetireGrant no tienen un parámetro de contexto de cifrado, pero puede incluir estas operaciones en una concesión que tenga una restricción de contexto de cifrado.
- Puede usar una restricción de contexto de cifrado en una concesión para la operación CreateGrant. La restricción de contexto de cifrado requiere que cualquier concesión creada con

el permiso `CreateGrant` tenga una restricción de contexto de cifrado igualmente estricta o más estricta.

AWS KMS admite dos restricciones de concesión de contexto de cifrado.

EncryptionContextEquals

Use `EncryptionContextEquals` para especificar el contexto de cifrado exacto para las solicitudes permitidas.

`EncryptionContextEquals` requiere que los pares de contexto de cifrado de la solicitud coincidan exactamente, incluido el uso de mayúsculas y minúsculas, con los pares de contexto de cifrado de la restricción de concesión. Los pares pueden aparecer en cualquier orden, pero las claves y los valores de cada par no pueden variar.

Por ejemplo, si la limitación de la concesión `EncryptionContextEquals` requiere el par de restricción de concesión `"Department": "IT"`, la concesión permite solicitudes del tipo especificado cuando el contexto de cifrado de la solicitud es exactamente `"Department": "IT"`.

EncryptionContextSubset

Use `EncryptionContextSubset` para requerir que las solicitudes incluyan pares de contexto de cifrado particulares.

`EncryptionContextSubset` requiere que la solicitud incluya todos los pares de contexto de cifrado de la restricción de concesión (una coincidencia exacta, incluido el uso de mayúsculas y minúsculas), pero la solicitud también puede tener pares de contexto de cifrado adicionales. Los pares pueden aparecer en cualquier orden, pero las claves y los valores de cada par no pueden variar.

Por ejemplo, si la limitación de la concesión `EncryptionContextSubset` requiere el par de contexto de concesión `Department=IT`, la concesión permite solicitudes del tipo especificado cuando el contexto de cifrado de la solicitud es `"Department": "IT"`, o incluye `"Department": "IT"` junto con otros pares de contexto de cifrado, como `"Department": "IT", "Purpose": "Test"`.

Para especificar una restricción de contexto de cifrado en una concesión de una clave KMS de cifrado simétrico, utilice el `Constraints` parámetro en la [CreateGrant](#) operación. La concesión que crea este comando concede a los usuarios autorizados asumir el permiso del rol `keyUserRole`

para llamar a la operación [Decrypt](#). Sin embargo, ese permiso solo es efectivo cuando el contexto de cifrado de la solicitud Decrypt es un par de contexto de cifrado "Department": "IT".

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

La concesión resultante se parece a la siguiente. Tenga en cuenta que el permiso concedido al rol keyUserRole solo es efectivo cuando la solicitud Decrypt utiliza el mismo par de contexto de cifrado especificado en la restricción de concesiones. Para encontrar las concesiones de una clave KMS, utilice la [ListGrants](#) operación.

```
$ 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 satisfacer la limitación de concesión EncryptionContextEquals, el contexto de cifrado en la solicitud para la operación Decrypt debe ser un par "Department": "IT".

Una solicitud como la siguiente del principal beneficiario satisfaría la restricción de concesiones `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
```

Cuando la restricción de concesión es `EncryptionContextSubset`, los pares de contexto de cifrado de la solicitud deben incluir los pares de contexto de cifrado de la restricción de concesión, pero la solicitud también puede incluir otros pares de contexto de cifrado. La siguiente restricción de concesión requiere que uno de los pares de contexto de cifrado en la solicitud sea "Department": "IT".

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

La siguiente solicitud del principal beneficiario sería capaz de satisfacer tanto las restricciones de concesión `EncryptionContextEqual` y `EncryptionContextSubset` en este ejemplo.

```
$ 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
```

Sin embargo, una solicitud como la siguiente del principal beneficiario podría satisfacer la limitación de concesión `EncryptionContextSubset`, pero se produciría un error en la restricción de concesiones `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 los servicios suelen utilizar restricciones de contexto de cifrado en las concesiones que les dan permiso para usar las claves de KMS en su Cuenta de AWS. Por ejemplo, Amazon DynamoDB utiliza una concesión como la siguiente para obtener permiso para utilizar la [Clave administrada de AWS](#) para DynamoDB de su cuenta. La restricción `EncryptionContextSubset` de esta concesión hace que los permisos de la concesión solo sean efectivos cuando el contexto de cifrado de la solicitud contiene pares `"tableName": "Services"` y `"subscriberID": "111122223333"`. Esta restricción de concesión significa que la concesión permite a DynamoDB utilizar la clave KMS especificada solo en una determinada tabla de su Cuenta de AWS.

Para obtener este resultado, ejecute la [ListGrants](#) operación en DynamoDB de su cuenta. Clave administrada de AWS

```
$ aws kms list-grants --key-id 0987dcba-09fe-87dc-65ba-ab0987654321

{
  "Grants": [
    {
      "Operations": [
        "Decrypt",
        "Encrypt",
        "GenerateDataKey",
        "ReEncryptFrom",
        "ReEncryptTo",
        "RetireGrant",
        "DescribeKey"
      ],
      "IssuingAccount": "arn:aws:iam::111122223333:root",
      "Constraints": {
        "EncryptionContextSubset": {
          "aws:dynamodb:tableName": "Services",
          "aws:dynamodb:subscriberId": "111122223333"
        }
      },
      "CreationDate": 1518567315.0,
      "KeyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
      "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
      "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
      "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
      "GrantId":
        "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59"
    }
  ]
}
```



```
]
}
```

SourceArn restricción

SourceArn

La SourceArn restricción restringe la concesión de permisos a las solicitudes realizadas en nombre de un AWS recurso específico, identificado por su nombre de recurso de Amazon (ARN). La restricción de SourceArn concesión consiste, de hecho, en introducir una clave de condición [aws:SourceArn](#) global en la concesión y solo se evalúa cuando la solicitud la realiza un servicio.

AWS

La SourceArn restricción siempre es necesaria cuando la subvención especifica un. `GranteeServicePrincipal` Opcionalmente, se puede usar con `GranteePrincipal`.

A diferencia de las restricciones del contexto de cifrado, la SourceArn restricción se admite en las concesiones de todos los tipos de claves KMS.

Para especificar una SourceArn restricción en una concesión, utilice el `Constraints` parámetro en la [CreateGrant](#) operación. En el siguiente ejemplo, se crea una concesión que permite a un director de AWS servicio llamar a Decrypt las operaciones Encrypt and, pero solo cuando la solicitud se realiza en nombre de la tabla de DynamoDB especificada. La SourceArn restricción es necesaria cuando la concesión especifica un. `GranteeServicePrincipal`

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

La concesión resultante tiene el siguiente aspecto. Los permisos de la concesión solo entran en vigor cuando la solicitud incluye un ARN de origen que coincide con el SourceArn valor de la restricción de concesión.

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

```

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

```

También puedes combinar varias restricciones en una sola concesión. Sin embargo, puede especificar una `EncryptionContextEquals` o ambas `EncryptionContextSubset`, pero no ambas.

Cuando `GranteeServicePrincipal` se especifica, la `SourceArn` restricción es obligatoria, por lo que las combinaciones válidas para crear una concesión con múltiples restricciones son las siguientes:

- `EncryptionContextEquals` por `SourceArn`
- `EncryptionContextSubset` por `SourceArn`

El siguiente ejemplo crea una concesión con `GranteeServicePrincipal` y `RetiringServicePrincipal` incluye tanto una `EncryptionContextSubset` restricción como una `SourceArn` restricción.

```

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

```

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

La concesión resultante tiene el siguiente aspecto. Observe que la respuesta incluye RetiringServicePrincipal campos GranteeServicePrincipal y, además, el Constraints campo contiene ambos campos EncryptionContextSubset y SourceArn.

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

Concesión CreateGrant del permiso

Una concesión puede incluir permiso para llamar a la operación CreateGrant. Pero cuando un [principal beneficiario](#) obtiene permiso para llamar a CreateGrant desde una concesión, en lugar de una política, ese permiso es limitado.

- El principal beneficiario solo puede crear concesiones que permitan algunas o todas las operaciones de la concesión principal.
- Las [restricciones de concesión](#) en las concesiones que crean deben ser al menos tan estrictas como las de la concesión principal.

Estas limitaciones no se aplican a las entidades principales que obtienen permiso CreateGrant de una política, aunque sus permisos pueden estar limitados por [condiciones de política](#).

Por ejemplo, supongamos que existe una concesión que permite a la entidad principal beneficiaria llamar a las operaciones GenerateDataKey, Decrypt y CreateGrant. Llamamos a una concesión que permita un permiso CreateGrant una concesión principal.

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

```
    }
  ]
}
```

La entidad principal beneficiaria, `exampleUser`, puede utilizar este permiso para crear una concesión que incluya cualquier subconjunto de las operaciones especificadas en la concesión principal, como `CreateGrant` y `Decrypt`. La concesión secundaria no puede incluir otras operaciones, como `ScheduleKeyDeletion` o `ReEncrypt`.

Además, las [restricciones de concesión](#) de las concesiones secundarias deben ser restrictivas o más restrictivas que las de la concesión principal. Por ejemplo, la concesión secundaria puede agregar pares a una restricción `EncryptionContextSubset` de la concesión principal, pero no puede eliminarlos. La concesión secundaria puede cambiar una restricción `EncryptionContextSubset` por una restricción `EncryptionContextEquals`, pero no al revés.

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;

Por ejemplo, el principal beneficiario puede usar el permiso `CreateGrant` que obtuvo de la concesión principal para crear la siguiente concesión secundaria. Las operaciones del subsidio por hijos son un subconjunto de las operaciones del subsidio parental y las restricciones de la concesión son más restrictivas.

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

El principal beneficiario en la concesión secundaria, `anotherUser`, puede usar su permiso `CreateGrant` para crear concesiones. Sin embargo, las concesiones que `anotherUser` crea deben incluir las operaciones en su concesión principal o en un subconjunto, y las restricciones de concesión deben ser las mismas o más estrictas.

Visualización de concesiones

Para ver la concesión, utilice la [ListGrants](#) operación. Debe especificar la clave KMS a la que se aplican las concesiones. También puedes filtrar la lista de subvenciones por identificador de subvención, director del concesionario o director de servicio del concesionario. Para obtener más ejemplos, consulte [Úselo ListGrants con un AWS SDK o CLI](#).

Para ver todas las subvenciones de la región Cuenta de AWS y con un director de servicio o [director de servicio en particular que se jubile](#), utilice. [ListRetirableGrants](#) Las respuestas incluyen detalles sobre cada concesión.

Note

El campo `GranteePrincipal` de la respuesta `ListGrants` generalmente contiene el principal beneficiario de la concesión. Sin embargo, cuando el principal beneficiario de la subvención es un AWS servicio, el `GranteePrincipal` campo contiene el [director](#) del servicio, que puede representar a varios directores distintos del concesionario.

Note

Cuando se crea una concesión con el [GranteeServicePrincipal](#) parámetro, la `ListGrants` respuesta incluye un campo en lugar de. `GranteeServicePrincipal` `GranteePrincipal` Esto distingue las subvenciones que se crearon explícitamente

`GranteeServicePrincipal` para un [director de AWS servicio](#) de las subvenciones en las que un AWS servicio está representado `GranteePrincipal` sobre el terreno.

Por ejemplo, el siguiente comando muestra todas las concesiones de una clave 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"
      ]
    }
  ]
}
```

En el siguiente ejemplo, se muestra `ListGrants` la respuesta a una concesión creada con `GranteeServicePrincipal`. Observe que la respuesta incluye un `GranteeServicePrincipal` campo en lugar de `GranteePrincipal` un valor y el `Constraints` campo contiene un `SourceArn` valor.

```
$ aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "Grants": [
    {
      "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```

        "CreationDate": 1718567315.0,
        "GrantId":
"abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
        "Constraints": {
            "EncryptionContextSubset": {
                "Department": "IT"
            },
            "SourceArn": "arn:aws:dynamodb:us-east-1:111122223333:table/
ExampleTable"
        },
        "RetiringServicePrincipal": "service-name.amazonaws.com",
        "Name": "",
        "IssuingAccount": "arn:aws:iam::111122223333:root",
        "GranteeServicePrincipal": "service-name.amazonaws.com",
        "Operations": [
            "Encrypt",
            "Decrypt",
            "GenerateDataKey"
        ]
    }
]
}

```

Uso de un token de concesión

La AWS KMS API sigue un modelo de [coherencia eventual](#). Al crear una concesión, es posible que la concesión no sea efectiva inmediatamente. Es posible que haya un breve retraso antes de que el cambio esté disponible a través de AWS KMS. Por lo general, el cambio tarda menos de unos segundos en propagarse por todo el sistema, pero en algunos casos puede tardar varios minutos. Una vez que el cambio se haya propagado totalmente a través del sistema, la entidad principal beneficiaria puede utilizar los permisos de la concesión sin especificar el token de concesión ni ninguna prueba de la concesión. Sin embargo, si una concesión es tan nueva que aún no la conocen todos AWS KMS, es posible que la solicitud no se realice correctamente y se produzca un `AccessDeniedException` error.

Para utilizar los permisos de una nueva concesión inmediatamente, utilice el [token de concesión](#) para la concesión. Guarde el token de concesión que devuelve la [CreateGrant](#) operación. A continuación, envíe el token de concesión en la solicitud de la AWS KMS operación. Puedes enviar un token de subvención a cualquier [operación de AWS KMS subvención](#) y puedes enviar varios tokens de subvención en la misma solicitud.

En el siguiente ejemplo, se utiliza la `CreateGrant` operación para crear una concesión que permita las operaciones de [descifrado `GenerateDataKey` descifrado](#). Guarda el token de concesión que `CreateGrant` devuelve en la variable `token`. Luego, en una llamada a la operación `GenerateDataKey`, utiliza el token de concesión en la variable `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
```

Las entidades principales con permiso también pueden usar un token de concesión para retirar una nueva concesión incluso antes de que la concesión esté disponible en AWS KMS. (La operación `RevokeGrant` no acepta un token de concesión). Para obtener más información, consulte [Retiro y revocación de concesiones](#).

```
# Retire the grant
$ aws kms retire-grant --grant-token $token
```

Retiro y revocación de concesiones

Para eliminar una concesión, retírela o revoquela.

Las [RevokeGrant](#) operaciones [RetireGrant](#) son muy similares entre sí. Ambas operaciones eliminan una concesión, lo que elimina los permisos que permite la concesión. La principal diferencia entre estas operaciones es cómo se autorizan.

RevokeGrant

Como la mayoría de AWS KMS las operaciones, el acceso a la `RevokeGrant` operación se controla mediante [políticas clave y políticas](#) de [IAM](#). Cualquier director puede llamar a la

[RevokeGrant](#) API con `kms:RevokeGrant` permiso. Este permiso está incluido en los permisos estándar otorgados a los administradores de claves. Normalmente, los administradores revocan una concesión para denegar los permisos que permite la concesión.

RetireGrant

La concesión determina quién puede retirarla. Este diseño le permite controlar el ciclo de vida de una concesión sin cambiar las políticas de claves o las políticas de IAM. Normalmente, usted retira una concesión cuando ha terminado de usar sus permisos.

Se puede retirar una subvención mediante cualquiera de las siguientes formas:

- El [director o el director](#) de [servicio que se jubila](#) especificados en la subvención.
- El director del [concesionario o el director](#) de [servicio del concesionario, si la subvención](#) incluye la operación. `RetireGrant`
- El lugar Cuenta de AWS en el que se creó la subvención.

Hay un permiso `kms:RetireGrant` que se puede utilizar en las políticas de IAM, pero tiene una utilidad limitada. Las entidades principales especificadas en la concesión pueden retirar una concesión sin el permiso `kms:RetireGrant`. El permiso `kms:RetireGrant` por sí solo no permite a las entidades principales retirar una concesión. El permiso `kms:RetireGrant` no entra en vigor en una [política clave](#) o en [una política de control de recursos](#).

- Para denegar el permiso de retirar una concesión, puede usar una acción `Deny` con el permiso `kms:RetireGrant` en sus políticas de IAM.
- El Cuenta de AWS propietario de la clave KMS puede delegar el `kms:RetireGrant` permiso a un principal de IAM de la cuenta.
- Si el principal que se retira es otro Cuenta de AWS, los administradores de la otra cuenta pueden delegar el permiso `kms:RetireGrant` para retirar la concesión a un director de IAM de esa cuenta.

La AWS KMS API sigue un modelo de [coherencia eventual](#). Al crear, retirar o revocar una concesión, es posible que haya un breve retraso antes de que el cambio esté disponible a través de AWS KMS. Por lo general, el cambio tarda menos de unos segundos en propagarse por todo el sistema, pero en algunos casos puede tardar varios minutos. Si necesitas eliminar una nueva subvención inmediatamente, antes de que esté disponible en todo momento AWS KMS, [usa un token de subvención](#) para retirarla. No puede usar un token de concesión para revocar una concesión.

Claves de estado para AWS KMS

Puede especificar las condiciones en las [políticas clave y en las políticas](#) de [IAM](#) que controlan el acceso a AWS KMS los recursos. La declaración de política solo será efectiva si se cumplen las condiciones. Por ejemplo, es posible que desee que solo entre en vigor una declaración de política después de una fecha específica. O puede que desee que una declaración de política controle el acceso solo cuando aparezca un valor específico en una solicitud de la API.

Para especificar condiciones, se utilizan claves de condición en el [elemento Condition](#) de la declaración de una política con [operadores de condiciones de IAM](#). Algunas claves de condición se aplican de forma general AWS; otras son específicas de AWS KMS.

Los valores de las claves de condición deben cumplir las reglas de caracteres y codificación de las políticas AWS KMS clave y las políticas de IAM. Para obtener más información sobre las reglas del documento de política de claves, consulte [Formato de la política de claves](#). Para obtener más información sobre las reglas del documento de política de IAM, consulte [Requisitos de nombres de IAM](#) en la Guía del usuario de IAM.

Temas

- [AWS claves de condición globales](#)
- [AWS KMS claves de condición](#)
- [AWS KMS claves de condición para plataformas certificadas](#)

AWS claves de condición globales

AWS define [las claves de condición globales](#), un conjunto de claves de condiciones de política para todos los AWS servicios que utilizan IAM para el control de acceso. AWS KMS admite todas las claves de condición globales. Puede utilizarlos en políticas AWS KMS clave y políticas de IAM.

Por ejemplo, puede usar la clave de condición PrincipalArn global [aws:](#) para permitir el acceso a una AWS KMS key (clave KMS) solo cuando el principal de la solicitud esté representado por el nombre de recurso de Amazon (ARN) en el valor de la clave de condición. Para respaldar el [control de acceso basado en atributos](#) (ABAC) AWS KMS, puede utilizar la clave de condición global [aws:ResourceTag/tag-key](#) en una política de IAM para permitir el acceso a las claves de KMS con una etiqueta determinada.

Para evitar que un AWS servicio se utilice de forma confusa en una política en la que el director es el director del [AWS servicio, puede utilizar las claves](#) de condición o globales.

[aws:SourceArns:SourceAccount](#) Para obtener más información, consulte [Uso de claves de condición aws:SourceArn o aws:SourceAccount](#).

Para obtener información sobre las claves de condición AWS globales, incluidos los tipos de solicitudes en las que están disponibles, consulte [las claves de contexto de condición AWS globales](#) en la Guía del usuario de IAM. Para obtener ejemplos del uso de claves de condición globales en las políticas de IAM, consulte [Control del acceso a las solicitudes](#) y [Control de claves de etiqueta](#) en la Guía del usuario de IAM.

En los temas siguientes se proporcionan instrucciones especiales para el uso de claves de condición basadas en direcciones IP y puntos de conexión de VPC.

Temas

- [Uso de la condición de dirección IP en políticas con AWS KMS permisos](#)
- [Uso de condiciones de punto final de VPC en políticas con AWS KMS permisos](#)
- [Utilizar direcciones IPv6 en IAM y AWS KMS políticas de claves](#)

Uso de la condición de dirección IP en políticas con AWS KMS permisos

Puede utilizarla AWS KMS para proteger sus datos en un [AWS servicio integrado](#). Sin embargo, tenga cuidado al especificar la [dirección IP, la condición, los operadores](#) o la clave de `aws:SourceIp` condición en la misma declaración de política a la que se permite o deniega el acceso AWS KMS. Por ejemplo, la política de [Denega el acceso en AWS función de la IP de origen](#) restringe AWS las acciones a las solicitudes del rango de IP especificado.

Considere esta situación:

1. Adjunta a una identidad de IAM una política como la que se muestra en [AWS: Denega el acceso a una identidad de IAM en AWS función de la IP de origen](#). A continuación, establece el valor de la clave de condición `aws:SourceIp` en el rango de direcciones IP de la empresa del usuario. Esta identidad de IAM tiene otras políticas adjuntadas que le permiten usar Amazon EBS, Amazon EC2 y AWS KMS.
2. La identidad intenta asociar un volumen de EBS cifrado a una instancia de EC2. Esta acción falla con un error de autorización a pesar de que el usuario tiene permiso para utilizar todos los servicios pertinentes.

El paso 2 falla porque la solicitud AWS KMS para descifrar la clave de datos cifrados del volumen proviene de una dirección IP asociada a la infraestructura de Amazon EC2. Para que la solicitud se realice correctamente, debe provenir de la dirección IP del usuario que la origina. Dado que la política del paso 1 deniega explícitamente todas las solicitudes de las direcciones IP que no sean las especificadas, a Amazon EC2 se le deniega el permiso para descifrar la clave de datos cifrada del volumen de EBS.

Además, la clave de condición `aws:SourceIP` no es efectiva si la solicitud procede de un [punto de conexión de Amazon VPC](#). Para restringir las solicitudes a un punto de conexión de VPC, incluido un [punto de conexión de VPC de AWS KMS](#), utilice las claves de condición `aws:SourceVpce` o `aws:SourceVpc`. Para obtener más información, consulte [Puntos de conexión de VPC - Control del uso de los puntos de conexión](#) en la Guía del usuario de Amazon VPC.

Uso de condiciones de punto final de VPC en políticas con AWS KMS permisos

[AWS KMS es compatible con los puntos de conexión de Amazon Virtual Private Cloud \(Amazon VPC\) que funcionan](#) con la tecnología de [AWS PrivateLink](#). Puede usar las siguientes [claves de condición globales](#) en las políticas clave y en las políticas de IAM para controlar el acceso a AWS KMS los recursos cuando la solicitud proviene de una VPC o utiliza un punto de enlace de la VPC. Para obtener más información, consulte [Utilice los puntos finales de VPC para controlar el acceso a los recursos AWS KMS](#).

- `aws:SourceVpc` limita el acceso a las solicitudes procedentes de la VPC especificada.
- `aws:SourceVpce` limita el acceso a las solicitudes procedentes del punto de conexión de VPC especificado.

Si utilizas estas claves de condición para controlar el acceso a las claves de KMS, podrías denegar inadvertidamente el acceso a los AWS servicios que se utilizan en tu nombre. AWS KMS

Procure evitar una situación como la del ejemplo de [claves de condición de dirección IP](#). Si restringes las solicitudes de una clave de KMS a una VPC o un punto final de VPC, es posible que se produzcan errores en las llamadas AWS KMS desde un servicio integrado, como Amazon S3 o Amazon EBS. Esto puede ocurrir incluso si la solicitud de origen procede en última instancia de la VPC o del punto de conexión de VPC.

Utilizar direcciones IPv6 en IAM y AWS KMS políticas de claves

Antes de intentar acceder a AWS KMS través de IPv6, asegúrese de que todas las políticas clave y de IAM que contengan restricciones de direcciones IP estén actualizadas para incluir los rangos

de direcciones IPv6. Si no se actualizan las políticas basadas en direcciones IP para gestionar direcciones IPv6, los clientes pueden perder u obtener incorrectamente el acceso cuando comienzan a utilizar IPv6. Para obtener información general sobre los controles de acceso de KMS, consulte [Permisos y acceso a claves KMS](#). Para obtener más información sobre la compatibilidad de KMS y la doble pila, consulte [Compatibilidad con puntos de conexión de doble pila](#).

 Important

Estas declaraciones no permiten ninguna acción. Utilice estas declaraciones en combinación con otras declaraciones que permiten acciones específicas.

La siguiente declaración niega explícitamente el acceso a todos los permisos de KMS a las solicitudes que se originen en el rango 192.0.2.* de direcciones IPv4. Cualquier dirección IP fuera de este rango no recibe permisos de KMS de forma explícita. Dado que todas las direcciones IPv6 están fuera del rango denegado, esta declaración no niega de forma explícita permisos de KMS para ninguna dirección IPv6.

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

Puede modificar el elemento Condition para permitir los rangos de direcciones IPv4 (192.0.2.0/24) e IPv6 (2001:db8:1234::/32) como se muestra en el siguiente ejemplo.

```
{
  "Sid": "DenyKMSPermissions",
  "Effect": "Deny",
```

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

AWS KMS claves de condición

AWS KMS proporciona un conjunto de claves de condición que puede utilizar en las políticas clave y en las políticas de IAM. Estas claves de condición son específicas de AWS KMS. Por ejemplo, puede utilizar la clave de condición `kms:EncryptionContext:context-key` para exigir un determinado [contexto de cifrado](#) al controlar el acceso a una clave KMS de cifrado simétrica.

Condiciones para una solicitud de operación de la API

Muchas claves de AWS KMS condición controlan el acceso a una clave de KMS en función del valor de un parámetro de la solicitud de una AWS KMS operación. Por ejemplo, puede usar la clave de KeySpec condición [kms:](#) en una política de IAM para permitir el uso de la [CreateKey](#) operación solo cuando el valor del KeySpec parámetro de la CreateKey solicitud sea RSA_4096 el mismo.

Este tipo de condición funciona incluso cuando el parámetro no aparece en la solicitud, como cuando se utiliza el valor predeterminado del parámetro. Por ejemplo, puede usar la clave de KeySpec condición [kms:](#) para permitir que los usuarios usen la CreateKey operación solo cuando el valor del KeySpec parámetro sea SYMMETRIC_DEFAULT, que es el valor predeterminado. Esta condición permite las solicitudes que tienen el parámetro KeySpec con el valor SYMMETRIC_DEFAULT y las solicitudes que no tienen el parámetro KeySpec.

Condiciones para claves KMS utilizadas en operaciones de la API

Algunas claves de AWS KMS condición pueden controlar el acceso a las operaciones en función de una propiedad de la clave KMS que se utiliza en la operación. Por ejemplo, puede usar la KeyOrigin condición [kms:](#) para permitir que los directores [GenerateDataKey](#) invoquen una clave KMS solo

cuando `Origin` la clave KMS lo sea `AWS_KMS`. Para averiguar si una clave de condición se puede utilizar de esta manera, consulte la descripción de la clave de condición.

La operación debe ser una operación de recursos de clave KMS, es decir, una operación que está autorizada para una clave KMS en particular. Para identificar las operaciones de recursos clave de KMS, en la [tabla de acciones y recursos](#), busque un valor de `KMS key` en la columna `Resources` para la operación. Si utiliza este tipo de clave de condición con una operación que no está autorizada para un recurso clave de KMS concreto, por ejemplo [ListKeys](#), el permiso no entra en vigor porque la condición nunca se puede cumplir. No existe ningún recurso de clave KMS involucrado en la autorización de la operación `ListKeys` y tampoco la propiedad `KeySpec`.

Los siguientes temas describen cada clave de AWS KMS condición e incluyen ejemplos de declaraciones de política que muestran la sintaxis de las políticas.

Uso de operadores de conjuntos con claves de condición

Cuando una condición de política compara dos conjuntos de valores, como el conjunto de etiquetas de una solicitud y el conjunto de etiquetas de una política, es necesario saber AWS cómo comparar los conjuntos. IAM define dos operadores de conjunto, `ForAnyValue` y `ForAllValues` con este fin. Utilice operadores de conjunto solo con claves de condición de varios valores, que los requieren. No utilice operadores de conjunto con claves de condición de un solo valor. Como siempre, pruebe sus declaraciones de políticas minuciosamente antes de usarlas en entornos de producción.

Las claves de condición tienen un valor único o un valor múltiple. Para determinar si una clave de AWS KMS condición es de un solo valor o de varios valores, consulte la columna Tipo de valor en la descripción de la clave de condición.

- **Single-valued** Las claves de condición tienen como máximo un valor en el contexto de la autorización (la solicitud o el recurso). Por ejemplo, dado que cada llamada a la API solo puede originarse desde una Cuenta de AWS, [kms: CallerAccount](#) es una clave de condición de un solo valor. No utilice un operador de conjunto con una clave de condición de un solo valor.
- **Multi-valued** Las claves de condición tienen varios valores en el contexto de autorización (la solicitud o el recurso). Por ejemplo, dado que cada clave de KMS puede tener varios alias, [kms: ResourceAliases](#) puede tener varios valores. Multi-valued las claves de condición requieren un operador de conjunto.

Tenga en cuenta que la diferencia entre las claves de condición de un solo valor y de varios valores depende del número de valores en el contexto de autorización, no del número de valores de la condición de política.

Warning

El uso de un operador de conjunto con una clave de condición de un solo valor puede crear una declaración de política excesivamente permisiva (o excesivamente restrictiva). Utilice operadores de conjunto solo con claves de condición de varios valores.

Si crea o actualiza una política que incluye un operador `ForAllValues` set con la clave de contexto o la clave de `aws:RequestTag/tag-key` condición `kmsEncryptionContext::`, AWS KMS devuelve el siguiente mensaje de error:

```
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 obtener información detallada acerca de los operadores de conjuntos `ForAnyValue` y `ForAllValues`, consulte [Uso de múltiples claves y valores](#) en la Guía del usuario de IAM. Para obtener información sobre el riesgo de utilizar el operador `ForAllValues` set con una condición de un solo valor, consulte [Advertencia de seguridad: ForAllValues con clave de un solo valor](#) en la Guía del usuario de IAM.

Temas

- [km: BypassPolicyLockoutSafetyCheck](#)
- [km: CallerAccount](#)
- [kms: CustomerMasterKeySpec \(obsoleto\)](#)
- [kms: CustomerMasterKeyUsage \(obsoleto\)](#)
- [km: DataKeyPairSpec](#)
- [km: EncryptionAlgorithm](#)
- [kmsEncryptionContext:clave de contexto](#)
- [kms: EncryptionContextKeys](#)
- [km: ExpirationModel](#)
- [kms: GrantConstraintType](#)
- [km: GrantsForAWSResource](#)
- [kms: GrantOperations](#)
- [km: GranteePrincipal](#)

- [kms: KeyAgreementAlgorithm](#)
- [km: KeyOrigin](#)
- [kms: KeySpec](#)
- [kms: KeyUsage](#)
- [km: MacAlgorithm](#)
- [km: MessageType](#)
- [km: MultiRegion](#)
- [kms: MultiRegionKeyType](#)
- [kms: PrimaryRegion](#)
- [km: ReEncryptOnSameKey](#)
- [kms: RequestAlias](#)
- [km: ResourceAliases](#)
- [kms: ReplicaRegion](#)
- [km: RetiringPrincipal](#)
- [kms: GrantConstraintSourceArn](#)
- [kms: GranteeServicePrincipal](#)
- [kms: RetiringServicePrincipal](#)
- [kms: RotationPeriodInDays](#)
- [km: ScheduleKeyDeletionPendingWindowInDays](#)
- [km: SigningAlgorithm](#)
- [kms: TrailingDaysWithoutKeyUsage](#)
- [km: ValidTo](#)
- [kms: ViaService](#)
- [km: WrappingAlgorithm](#)
- [km: WrappingKeySpec](#)

km: BypassPolicyLockoutSafetyCheck

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:BypassPolicyLockoutSafetyCheck	Booleano	Single-valued	CreateKey PutKeyPolicy	Solo políticas de IAM Políticas de claves y políticas de IAM

La clave de kms:BypassPolicyLockoutSafetyCheck condición controla el acceso a [PutKeyPolicy](#) las operaciones [CreateKey](#) en función del valor del BypassPolicyLockoutSafetyCheck parámetro de la solicitud.

En el siguiente ejemplo, la declaración de la política de IAM impide que los usuarios eludan la comprobación de seguridad de bloqueo de la política denegándoles el permiso para crear claves KMS cuando el valor del parámetro BypassPolicyLockoutSafetyCheck de la solicitud CreateKey es true.

```
{
  "Effect": "Deny",
  "Action": [
    "kms:CreateKey",
    "kms:PutKeyPolicy"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

También puede utilizar la clave de condición kms:BypassPolicyLockoutSafetyCheck en una política de IAM o en una política de claves para controlar el acceso a la operación PutKeyPolicy.

En el ejemplo siguiente, la declaración de una política de claves impide que los usuarios eludan la comprobación de seguridad de bloqueo cuando se modifica la política de una clave KMS.

En lugar de utilizar una operación Deny explícita, esta declaración de la política utiliza Allow con [el operador de condición Null](#) para permitir únicamente el acceso cuando la solicitud no contiene el parámetro BypassPolicyLockoutSafetyCheck. Cuando no se utiliza el parámetro, el valor predeterminado es false. Esta declaración de la política es algo más débil y puede anularse en el caso improbable de que sea necesario eludirla.

```
{
  "Effect": "Allow",
  "Action": "kms:PutKeyPolicy",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

Véase también

- [kms: KeySpec](#)
- [km: KeyOrigin](#)
- [kms: KeyUsage](#)

km: CallerAccount

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:CallerAccount	Cadena	Single-valued	Operaciones de recursos de claves KMS Operaciones de almacén de	Políticas de claves y políticas de IAM

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política

claves personali
zadas

Puede utilizar esta clave de condición para permitir o denegar el acceso a todas las identidades (usuarios y roles) de una Cuenta de AWS. En las políticas de claves, se usa el elemento `Principal` para especificar las identidades a las que se aplica la declaración de política. La sintaxis del elemento `Principal` no proporciona una forma de especificar todas las identidades en una Cuenta de AWS. Sin embargo, puede lograr este efecto combinando esta clave de condición con un `Principal` elemento que especifique todas AWS las identidades.

Puede usarla para controlar el acceso a cualquier operación de recurso clave de KMS, es decir, cualquier AWS KMS operación que utilice una clave de KMS concreta. Para identificar las operaciones de recursos clave KMS, en la [Tabla de acciones y recursos](#), busque un valor de `KMS key` en la columna `Resources` para la operación. También es válido para operaciones que administran [almacenes de claves personalizados](#).

Por ejemplo, la siguiente declaración de política de claves demuestra cómo utilizar la clave de condición `kms:CallerAccount`. Esta declaración de política se encuentra en la política clave `Clave administrada de AWS de Amazon EBS`. Combina un `Principal` elemento que especifica todas las AWS identidades con la clave de `kms:CallerAccount` condición para permitir el acceso efectivo a todas las identidades en Cuenta de AWS `111122223333`. Contiene una clave de AWS KMS condición adicional (`kms:ViaService`) para limitar aún más los permisos al permitir únicamente las solicitudes que llegan a través de Amazon EBS. Para obtener más información, 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)

La clave de condición kms:CustomerMasterKeySpec está obsoleta. En su lugar, utilice la clave de KeySpec condición [kms:](#).

Las claves de condición kms:CustomerMasterKeySpec y kms:KeySpec funcionan de la misma forma. Solo los nombres difieren. Le recomendamos que utilice kms:KeySpec. Sin embargo, para evitar cambios irrelevantes, AWS KMS admite ambas claves de condición.

kms: CustomerMasterKeyUsage (obsoleto)

La clave de condición kms:CustomerMasterKeyUsage está obsoleta. En su lugar, utilice la clave de KeyUsage condición [kms:](#).

Las claves de condición kms:CustomerMasterKeyUsage y kms:KeyUsage funcionan de la misma forma. Solo los nombres difieren. Le recomendamos que utilice kms:KeyUsage. Sin embargo, para evitar cambios irrelevantes, AWS KMS admite ambas claves de condición.

km: DataKeyPairSpec

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:DataK eyPairSpec	Cadena	Single-valued	Generated ataKeyPair Generated ataKeyPai	Políticas de claves y políticas de IAM

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
			GenerateDataKeyPairWithoutPlaintext	

Puede utilizar esta clave de condición para controlar el acceso a las [GenerateDataKeyPairWithoutPlaintext](#) operaciones [GenerateDataKeyPair](#) en función del valor del `KeyPairSpec` parámetro de la solicitud. Por ejemplo, puede permitir que un usuario genere solo determinados tipos de pares de claves de datos.

El siguiente ejemplo de declaración de política de claves utiliza la clave de condición `kms:DataKeyPairSpec` para permitir a los usuarios utilizar la clave KMS para generar solo pares de claves de datos RSA.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:DataKeyPairSpec": "RSA*"
    }
  }
}
```

Véase también

- [kms: KeySpec](#)
- [the section called “km: EncryptionAlgorithm”](#)
- [the section called “kmsEncryptionContext:clave de contexto”](#)
- [the section called “kms: EncryptionContextKeys”](#)

km: EncryptionAlgorithm

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:EncryptionAlgorithm	Cadena	Single-valued	Decrypt Encrypt GenerateDataKey GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext GenerateDataKeyWithoutPlaintext ReEncrypt	Políticas de claves y políticas de IAM

Puede utilizar la clave de condición `kms:EncryptionAlgorithm` para controlar el acceso a operaciones criptográficas en función del algoritmo de cifrado que se utiliza en la operación. [Para las ReEncrypt operaciones de cifrado, descifrado y, por el contrario, controla el acceso en función del valor del EncryptionAlgorithm parámetro de la solicitud.](#) Para operaciones que generan claves de datos y pares de claves de datos, controla el acceso basado en el algoritmo de cifrado que se utiliza para cifrar la clave de datos.

Esta clave de condición no afecta a las operaciones que se realizan fuera de AWS KMS, como el cifrado con la clave pública de un par de claves KMS asimétricas fuera de. AWS KMS

EncryptionAlgorithm parámetro en una solicitud

Para permitir a los usuarios utilizar solo un algoritmo de cifrado determinado con una clave KMS, utilice una declaración de política con un efecto Deny y un operador de condición `StringNotEquals`. Por ejemplo, la siguiente declaración de política de claves de ejemplo prohíbe a las entidades principales que pueden asumir el rol `ExampleRole` utilizar esta clave KMS en las operaciones criptográficas especificadas, a menos que el algoritmo de cifrado de la solicitud sea `RSAES_OAEP_SHA_256`, un algoritmo de cifrado asimétrico que se usa con claves KMS RSA.

A diferencia de una declaración de política que permite a un usuario utilizar un algoritmo de cifrado determinado, una declaración de política con un doble negativo como este impide que otras políticas y concesiones para esta clave KMS permitan que este rol utilice otros algoritmos de cifrado. Deny en esta declaración de política tiene prioridad sobre cualquier otra política de claves o política de IAM con un efecto Allow y tiene prioridad sobre todas las concesiones para esta clave KMS y sus entidades principales.

```
{
  "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 cifrado utilizado para la operación

También puede utilizar la clave de condición `kms:EncryptionAlgorithm` para controlar el acceso a operaciones según el algoritmo de cifrado utilizado en la operación, incluso cuando el algoritmo no se especifica en la solicitud. Esto le permite requerir o prohibir el algoritmo `SYMMETRIC_DEFAULT`, que puede que no se especifique en una solicitud porque es el valor predeterminado.

Esta característica le permite usar la clave de condición `kms:EncryptionAlgorithm` para controlar el acceso a las operaciones que generan claves de datos y pares de claves de datos. Estas operaciones solo utilizan claves KMS de cifrado simétricas y el algoritmo `SYMMETRIC_DEFAULT`.

Por ejemplo, esta política de IAM limita sus entidades principales al cifrado simétrico. Deniega el acceso a cualquier clave KMS en la cuenta de ejemplo para operaciones criptográficas a menos que el algoritmo de cifrado especificado en la solicitud o utilizado en la operación sea `SYMMETRIC_DEFAULT`. Incluye `GenerateDataKey*` [GenerateDataKey](#) adiciones [GenerateDataKeyWithoutPlaintext](#), [GenerateDataKeyPair](#), y [GenerateDataKeyPairWithoutPlaintext](#) a los permisos. La condición no tiene ningún efecto en estas operaciones porque siempre utilizan un algoritmo de cifrado simétrico.

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

Véase también

- [the section called “km: MacAlgorithm”](#)
- [km: SigningAlgorithm](#)

kmsEncryptionContext:clave de contexto

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:EncryptionContext: <i>context-key</i>	Cadena	Single-valued	CreateGrant Encrypt Decrypt GenerateDataKey GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext GenerateDataKeyWithoutPlaintext ReEncrypt RetireGrant	Políticas de claves y políticas de IAM

Puede usar la clave de condición `kms:EncryptionContext:context-key` para controlar el acceso a una [clave KMS de cifrado simétrica](#) en función del [contexto de cifrado](#) de una solicitud para una [operación criptográfica](#). Utilice esta clave de condición para evaluar la clave y el valor del par de contexto de cifrado. Para evaluar solo las claves de contexto de cifrado o para requerir un contexto de cifrado independientemente de las claves o los valores, utilice la clave `EncryptionContextKeys` condicionada [kms:](#).

 Note

Los valores de claves de condición deben ajustarse a las reglas de caracteres de políticas de claves y políticas de IAM. Algunos caracteres válidos en un contexto de cifrado no son válidos en las políticas. Es posible que no pueda utilizar esta clave de condición para expresar todos los valores de contexto de cifrado válidos. Para obtener más información sobre las reglas del documento de política de claves, consulte [Formato de la política de claves](#). Para obtener más información sobre las reglas del documento de política de IAM, consulte [Requisitos de nombres de IAM](#) en la Guía del usuario de IAM.

No puede especificar un contexto de cifrado en una operación criptográfica con una [clave KMS asimétrica](#) o una [clave KMS HMAC](#). Los algoritmos asimétricos y los algoritmos MAC no son compatibles con un contexto de cifrado.

Para usar la clave de contexto `kms:EncryptionContext::`, sustituya el *context-key* marcador de posición por la clave de contexto de cifrado. Sustituya el marcador de posición *context-value* por el valor de contexto de cifrado.

```
"kms:EncryptionContext:context-key": "context-value"
```

Por ejemplo, la siguiente clave de condición especifica un contexto de cifrado en el que la clave es `AppName` y el valor es `ExampleApp` (`AppName = ExampleApp`).

```
"kms:EncryptionContext:AppName": "ExampleApp"
```

Esta es una [clave de condición de un solo valor](#). La clave de la clave de condición especifica una clave de contexto de cifrado determinada (*context-key*). Aunque puede incluir varios pares de contexto de cifrado en cada solicitud de la API, el par de contexto de cifrado con el *context-key* puede tener solo un valor. Por ejemplo, la clave de condición `kms:EncryptionContext:Department` solo se aplica a los pares de contexto de cifrado con una clave `Department`, y cualquier par de contexto de cifrado dado con la clave `Department` solo puede tener un valor.

No utilice un operador de conjunto con la clave de condición `kms:EncryptionContext:context-key`. Si crea una declaración de política con una acción `Allow`, la clave de condición `kms:EncryptionContext:context-key` y el operador de conjunto `ForAllValues`, la condición

permite solicitudes sin contexto de cifrado y solicitudes con pares de contexto de cifrado que no se especifican en la condición de política.

Warning

No utilice un operador de conjunto `ForAnyValue` o `ForAllValues` con esta clave de condición de un solo valor. Estos operadores de conjunto pueden crear una condición de política que no requiera valores que pretenda requerir y permite valores que pretende prohibir.

Si crea o actualiza una política que incluye un operador `ForAllValues` set con la clave de contexto `kms:EncryptionContext:`, AWS KMS devuelve el siguiente mensaje de error: `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 requerir un par de contexto de cifrado concreto, utilice la clave de condición `kms:EncryptionContext:context-key` con el operador `StringEquals`.

En la siguiente declaración de política de claves de ejemplo se permite a las entidades principales que pueden asumir la función utilizar la clave KMS en una solicitud `GenerateDataKey` solo cuando el contexto de cifrado de la solicitud contiene el par `AppName:ExampleApp`. Se permiten otros pares de contexto de cifrado.

El nombre de las claves distingue entre mayúsculas y minúsculas. La distinción de mayúsculas y minúsculas del valor se determina mediante el operador de condición, como `StringEquals`. Para obtener más información, consulte [Uso de mayúsculas y minúsculas en las condiciones de contexto de cifrado](#).

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

```

    "kms:EncryptionContext:AppName": "ExampleApp"
  }
}

```

Para exigir un par de contextos de cifrado y prohibir todos los demás pares de contextos de cifrado, utilice `kms:EncryptionContext: context-key` y [kms:EncryptionContextKeys](#) en la declaración de política. En la siguiente declaración de política de claves se utiliza la condición `kms:EncryptionContext:AppName` para requerir el par de contexto de cifrado `AppName=ExampleApp` en la solicitud. También utiliza una clave de condición `kms:EncryptionContextKeys` con el operador de conjunto `ForAllValues` para permitir solo la clave de contexto de cifrado `AppName`.

El operador de conjunto `ForAllValues` limita las claves de contexto de cifrado en la solicitud a `AppName`. Si la condición `kms:EncryptionContextKeys` con el operador de conjunto `ForAllValues` se utilizó solo en una declaración de política, este operador de conjunto permitiría solicitudes sin contexto de cifrado. Sin embargo, si la solicitud no tenía contexto de cifrado, la condición `kms:EncryptionContext:AppName` fallaría. Para conocer los detalles sobre el operador de conjunto `ForAllValues`, consulte [Uso de múltiples claves y valores](#) en la Guía del usuario de 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"
      ]
    }
  }
}

```

También puede utilizar esta clave de condición para denegar el acceso a una clave KMS para una operación concreta. En la siguiente declaración de política de claves de ejemplo se utiliza un efecto Deny para prohibir a la entidad principal utilizar la clave KMS si el contexto de cifrado de la solicitud contiene un par de contexto de cifrado Stage=Restricted. Esta condición permite una solicitud con otros pares de contexto de cifrado, incluidos los pares de contexto de cifrado con la clave Stage y otros valores, como Stage=Test.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": "Restricted"
    }
  }
}
```

Uso de varios pares de contexto de cifrado

Puede requerir o prohibir varios pares de contexto de cifrado. También puede requerir uno de varios pares de contexto de cifrado. Para conocer los detalles de la lógica utilizada para interpretar estas condiciones, consulte [Crear una condición con varias claves o valores](#) en la Guía del usuario de IAM.

Note

Las versiones anteriores de este tema mostraban declaraciones de política que utilizaban los operadores ForAnyValue y ForAllValues set con la clave de condición kms:EncryptionContext: context-key. Usando un operador de conjunto con una [clave de condición de un solo valor](#) puede dar lugar a políticas que permiten solicitudes sin contexto de cifrado y pares de contexto de cifrado no especificados.

Por ejemplo, una condición de política con el efecto Allow, el operador de conjunto ForAllValues y la clave de condición "kms:EncryptionContext:Department": "IT" no limita el contexto de cifrado al par "Department=IT". Permite solicitudes sin contexto de cifrado y solicitudes con pares de contexto de cifrado no especificados, como Stage=Restricted.

Revise sus políticas y elimine el operador set de cualquier condición con `kms:EncryptionContext: context-key`. Los intentos de crear o actualizar una política con este formato fallan con una excepción `OverlyPermissiveCondition`. Para resolver el error, elimine el operador de conjunto.

Para requerir varios pares de contexto de cifrado, enumere los pares en la misma condición. En la siguiente declaración de política de claves de ejemplo se requieren dos pares de contexto de cifrado, `Department=IT` y `Project=Alpha`. Debido a que las condiciones tienen diferentes claves (`kms:EncryptionContext:Department` y `kms:EncryptionContext:Project`), están implícitamente conectados por un operador AND. Otros pares de contexto de cifrado están permitidos, pero no son necesarios.

```
{
  "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 requerir un par de contexto de cifrado U otro par, coloque cada clave de condición en una declaración de política independiente. En la siguiente política de claves de ejemplo se requiere `Department=IT` o `Project=Alpha` pares, o ambos. Otros pares de contexto de cifrado están permitidos, pero no son necesarios.

```
{
  "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 cifrado específicos y excluir todos los demás pares de contextos de cifrado, utilice tanto la clave de contexto `kms:EncryptionContext:` como la de la declaración de política [kms:EncryptionContextKeys](#). La siguiente declaración de política clave utiliza la condición de clave contextual `kmsEncryptionContext::` para requerir un contexto de cifrado con ambos pares y. *Department=IT Project=Alpha* Utiliza una clave de condición `kms:EncryptionContextKeys` con el operador de conjuntos `ForAllValues` para permitir solo las claves de contexto de cifrado `Department` y `Project`.

El operador de conjuntos `ForAllValues` limita las claves de contexto de cifrado en la solicitud a `Department` y `Project`. Si se usara solo en una condición, este operador de conjunto permitiría solicitudes sin contexto de cifrado, pero en esta configuración, la clave de contexto `kms:EncryptionContext:` en esta condición fallaría.

```

{
  "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"
    ]
  }
}

```

También puede prohibir varios pares de contexto de cifrado. En la siguiente declaración de política de claves de ejemplo se utiliza un efecto Deny para prohibir a la entidad principal utilizar las claves KMS si el contexto de cifrado de la solicitud contiene un par Stage=Restricted o Stage=Production.

Múltiples valores (Restricted y Production) para la misma clave (kms:EncryptionContext:Stage) están implícitamente conectados por un OR. Para conocer los detalles, consulte la [Lógica de evaluación para condiciones con múltiples claves o valores](#) en la Guía del usuario de IAM.

```

{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": [
        "Restricted",
        "Production"
      ]
    }
  }
}

```

Uso de mayúsculas y minúsculas en las condiciones de contexto de cifrado

El contexto de cifrado que se especifica en una operación de descifrado debe coincidir exactamente, incluido el uso de mayúsculas y minúsculas, con el contexto de cifrado que se especifica en la operación de cifrado. Solo puede variar el orden de los pares de un contexto de cifrado con varios pares.

Sin embargo, en las condiciones de políticas, la clave de condición no distingue entre mayúsculas y minúsculas. La distinción de mayúsculas y minúsculas del valor de la condición se determina por el [operador de condición de política](#) que utilice, como `StringEquals` o `StringEqualsIgnoreCase`.

Por tanto, la clave de condición, que consta del prefijo `kms:EncryptionContext:` y el valor sustituto de *context-key*, no distingue mayúsculas de minúsculas. Una política que utiliza esta condición no comprueba el uso de mayúsculas o minúsculas de ninguno de los elementos de la clave de condición. La distinción de mayúsculas y minúsculas del valor, es decir *context-value*, lo determina el operador de la política de condición.

Por ejemplo, la siguiente declaración de política permite la operación cuando el contexto de cifrado incluye una clave Appname, independientemente de si está en mayúsculas o minúsculas. La condición `StringEquals` requiere que `ExampleApp` esté en mayúsculas cuando se especifique.

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

Para requerir una clave de contexto de cifrado que distinga entre mayúsculas y minúsculas, utilice la condición [kms: EncryptionContextKeys policy](#) con un operador de condición que distinga entre mayúsculas y minúsculas, como `StringEquals`. En esta condición de política, como la clave de contexto de cifrado es el valor de la condición de política, el operador de condición determina si se distingue entre mayúsculas y minúsculas.

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

Para exigir una evaluación de la clave y el valor del contexto de cifrado que distinga entre mayúsculas y minúsculas, utilice las condiciones de política `kms:EncryptionContextKeys` y la clave de contexto `kms:EncryptionContext::` juntas en la misma declaración de política. El operador de condición sensible a mayúsculas y minúsculas (como `StringEquals`) siempre se aplica al valor de la condición. La clave de contexto de cifrado (como `AppName`) es el valor de la condición `kms:EncryptionContextKeys`. El valor del contexto de cifrado (por ejemplo `ExampleApp`) es el valor de la condición de clave contextual `kms:EncryptionContext::`.

Por ejemplo, en la siguiente declaración de política de clave de ejemplo, como el operador `StringEquals` distingue entre mayúsculas y minúsculas, tanto la clave de contexto de cifrado como el valor de contexto de cifrado distinguen entre mayúsculas y 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"
    }
  }
}
```

```
}
```

Uso de variables en una condición de contexto de cifrado

La clave y el valor de un par de contexto de cifrado deben ser cadenas literales simples. No pueden ser números enteros ni objetos, ni ningún tipo que no esté totalmente resuelto. Si utiliza un tipo diferente, como un entero o un flotante, lo AWS KMS interpreta como una cadena literal.

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

Sin embargo, el valor de la clave de condición `kms:EncryptionContext:context-key` puede ser una [variable de política de IAM](#). Estas variables de política se resuelven en tiempo de ejecución con arreglo a los valores de la solicitud. Por ejemplo, `aws:CurrentTime` se resuelve en la hora de la solicitud y `aws:username` se resuelve en el nombre descriptivo del autor de la llamada.

Puede utilizar estas variables de política para crear una declaración de política con una condición que requiera información muy específica de un contexto de cifrado, como el nombre de usuario del autor de la llamada. Como contiene una variable, puede utilizar la misma declaración de política con todos los usuarios que puedan adoptar ese rol. No tiene que escribir una declaración de política diferente para cada usuario.

Imagine una situación en la que desea que todos los usuarios que puedan adoptar un rol utilicen la misma clave KMS para cifrar y descifrar los datos. Sin embargo, solo quiere que puedan descifrar los datos que ellos han cifrado. Comience por exigir que cada solicitud AWS KMS incluya un contexto de cifrado en el que la clave esté `user` y el valor sea el nombre de AWS usuario de la persona que llama, como el siguiente.

```
"encryptionContext": {  
  "user": "bob"  
}
```

A continuación, para forzar la aplicación de este requisito, puede utilizar una declaración de política como la del siguiente ejemplo. Esta declaración de política concede al rol `TestTeam` permiso para cifrar y descifrar datos con la clave KMS. Sin embargo, el permiso solo es válido cuando el contexto de cifrado de la solicitud incluye un par `"user": "<username>"`. Para representar el nombre de usuario, la condición utiliza la variable de política [aws:username](#).

Cuando se evalúa la solicitud, la variable de condición se sustituye por el nombre de usuario del autor de la llamada. Por tanto, la condición necesita el contexto de cifrado "user": "bob" para "bob" y "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}"
    }
  }
}
```

Puede utilizar una variable de política de IAM únicamente en el valor del par de la clave de condición `kms:EncryptionContext:context-key`. No puede utilizar una variable en la clave.

También puede utilizar [claves de contexto específicas del proveedor](#) en las variables. Estas claves de contexto identifican de forma exclusiva a los usuarios que han iniciado sesión AWS mediante la federación de identidades web.

Al igual que todas las variables, estas solo se pueden utilizar en la condición de política `kms:EncryptionContext:context-key`, no en el contexto de cifrado real. Y solo se pueden utilizar en el valor de la condición, no en la clave.

Por ejemplo, la siguiente declaración de política de clave es similar a la anterior. Sin embargo, la condición requiere un contexto de cifrado en el que la clave sea sub y el valor identifique de forma inequívoca a un usuario que ha iniciado sesión en un grupo de usuarios de Amazon Cognito. Para obtener más información acerca de la identificación de usuarios y roles en Amazon Cognito, consulte [Roles de IAM](#) en la [Guía para desarrolladores de 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}"
    }
  }
}
```

Véase también

- [the section called “kms: EncryptionContextKeys”](#)
- [the section called “kms: GrantConstraintType”](#)

kms: EncryptionContextKeys

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Encry ptionCont extKeys	Cadena (lista)	Multi-valued	CreateGrant Decrypt Encrypt Generated ataKey Generated ataKeyPair Generated ataKeyPai	Políticas de claves y políticas de IAM

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
			rWithoutP laintext Generated ataKeyWit houtPlain text ReEncrypt RetireGrant	

Puede usar la clave de condición `kms:EncryptionContextKeys` para controlar el acceso a una [clave KMS de cifrado simétrica](#) en función del [contexto de cifrado](#) de una solicitud para una operación criptográfica. Utilice esta clave de condición para evaluar únicamente la clave de cada par de contexto de cifrado. Utilice esta clave de condición `kms:EncryptionContext:context-key` para evaluar la clave y el valor del par de contexto de cifrado.

No puede especificar un contexto de cifrado en una operación criptográfica con una [clave KMS asimétrica](#) o una [clave KMS HMAC](#). Los algoritmos asimétricos y los algoritmos MAC no son compatibles con un contexto de cifrado.

Note

Los valores de las claves de condición, incluida una clave de contexto de cifrado, deben ajustarse a las reglas de caracteres y codificación de las políticas de AWS KMS claves. Es posible que no pueda utilizar esta clave de condición para expresar todas las claves de contexto de cifrado válidas. Para obtener más información sobre las reglas del documento de política de claves, consulte [Formato de la política de claves](#). Para obtener más información sobre las reglas del documento de política de IAM, consulte [Requisitos de nombres de IAM](#) en la Guía del usuario de IAM.

Esta es una [clave de condición multivalor](#). Puede especificar varios pares de contexto de cifrado en cada solicitud de la API. `kms:EncryptionContextKeys` compara las claves de contexto de cifrado de la solicitud con el conjunto de claves de contexto de cifrado de la política. Para determinar cómo se comparan estos conjuntos, debe proporcionar un operador de conjuntos `ForAnyValue` o `ForAllValues` en la condición de política. Para conocer los detalles sobre los operadores de conjuntos, consulte [Uso de múltiples claves y valores](#) en la Guía del usuario de IAM.

- `ForAnyValue`: al menos una clave de contexto de cifrado en la solicitud debe coincidir con una clave de contexto de cifrado en la condición de política. Se permiten otras claves de contexto de cifrado. Si la solicitud no tiene contexto de cifrado, la condición no se cumple.
- `ForAllValues`: cada clave de contexto de cifrado de la solicitud debe coincidir con una clave de contexto de cifrado en la condición de política. Este operador de conjunto limita las claves de contexto de cifrado a aquellas en la condición de política. No requiere ninguna clave de contexto de cifrado, pero prohíbe las claves de contexto de cifrado no especificadas.

En la siguiente declaración de política de claves de ejemplo se utiliza la condición `kms:EncryptionContextKeys` con la clave de condición del operador de conjuntos `ForAnyValue`. Esta declaración de la política utiliza la clave KMS para las operaciones especificadas, pero solamente cuando al menos uno de los pares de contexto de cifrado de la solicitud incluye la clave `AppName`, independientemente de su valor.

Por ejemplo, esta declaración de política de claves permite una solicitud `GenerateDataKey` con dos pares de contexto de cifrado, `AppName=Helper` y `Project=Alpha`, porque el primer par de contexto de cifrado cumple con la condición. Una solicitud con solo `Project=Alpha` o sin contexto de cifrado fallaría.

Como la operación de la [StringEquals](#) condición distingue entre mayúsculas y minúsculas, esta declaración de política exige que la clave de contexto de cifrado esté escrita y escrita en mayúsculas y minúsculas. Sin embargo, puede utilizar un operador de condición que omita el uso de mayúsculas y minúsculas de la clave, como `StringEqualsIgnoreCase`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
```

```
"kms:GenerateDataKey*"
],
"Resource": "*",
"Condition": {
  "ForAnyValue:StringEquals": {
    "kms:EncryptionContextKeys": "AppName"
  }
}
}
```

También puede utilizar la clave de condición `kms:EncryptionContextKeys` para solicitar un contexto de cifrado (cualquier contexto de cifrado) en las operaciones criptográficas que utilizan la clave KMS.

La siguiente declaración de política de ejemplo utiliza la clave de condición `kms:EncryptionContextKeys` con el [operador de condición Null](#) para permitir el acceso para utilizar una clave KMS de solo cuando el contexto de cifrado de la solicitud de la API no sea nulo. Esta condición no comprueba las claves ni los valores del contexto de cifrado. Solo verifica que existe el contexto de cifrado.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContextKeys": false
    }
  }
}
```

Véase también

- [kmsEncryptionContext:clave de contexto](#)
- [kms: GrantConstraintType](#)

km: ExpirationModel

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:ExpirationModel	Cadena	Single-valued	ImportKeyMaterial	Políticas de claves y políticas de IAM

La clave de kms:ExpirationModel condición controla el acceso a la [ImportKeyMaterial](#) operación en función del valor del [ExpirationModel](#) parámetro de la solicitud.

ExpirationModel es un parámetro opcional que determina si el material de claves importado vence. Los valores válidos son KEY_MATERIAL_EXPIRES y KEY_MATERIAL_DOES_NOT_EXPIRE. El valor predeterminado es KEY_MATERIAL_EXPIRES.

La fecha y la hora de caducidad vienen determinadas por el valor del [ValidTo](#) parámetro. El parámetro ValidTo es necesario a menos que el valor del parámetro ExpirationModel sea KEY_MATERIAL_DOES_NOT_EXPIRE. También puede usar la clave de ValidTo condición [kms:](#) para exigir una fecha de caducidad determinada como condición de acceso.

La siguiente declaración de política de ejemplo utiliza la clave de condición kms:ExpirationModel para permitir a los usuarios importar solamente material de claves en una clave KMS cuando la solicitud incluye el parámetro ExpirationModel y su valor sea 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"
    }
  }
}
```

```
}

```

También puede utilizar la clave de condición `kms:ExpirationModel` para que un usuario solamente pueda importar material de claves cuando expire el material de claves. En el ejemplo siguiente, la declaración de la política de clave utiliza la clave de condición `kms:ExpirationModel` con el [operador de condición Null](#) para que un usuario solamente pueda importar material de claves cuando la solicitud no contenga un parámetro `ExpirationModel`. El valor predeterminado `ExpirationModel` es `KEY_MATERIAL_EXPIRES`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:ExpirationModel": true
    }
  }
}
```

Véase también

- [km: ValidTo](#)
- [km: WrappingAlgorithm](#)
- [km: WrappingKeySpec](#)

kms: GrantConstraintType

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Grant Constrain tType	Cadena	Single-valued	CreateGrant RetireGrant	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para controlar el acceso a la [CreateGrant](#) operación en función del tipo de [restricción de concesión](#) de la solicitud.

Al crear una concesión, también puede especificar una restricción de concesión para permitir las operaciones que permite la concesión solo cuando esté presente un determinado [contexto de cifrado](#). La restricción de concesión puede ser uno de estos dos tipos: `EncryptionContextEquals` o `EncryptionContextSubset`. Puede usar esta clave de condición para comprobar que la solicitud contiene un tipo u otro.

 Note

La clave de `kms:GrantConstraintType` condición solo evalúa las restricciones de concesión del contexto de cifrado (`EncryptionContextEquals` y `EncryptionContextSubset`). Para controlar el acceso en función de la restricción de `SourceArn` concesión, utilice la clave de `GrantConstraintSourceArn` condición [kms:](#).

 Important

No incluya información confidencial en este campo. Este campo puede mostrarse en texto plano en CloudTrail los registros y otros resultados.

En la siguiente declaración de política de clave utiliza la clave de condición `kms:GrantConstraintType` para permitir que un usuario cree concesiones solo cuando la solicitud incluya una restricción de concesión `EncryptionContextEquals`. En el ejemplo se muestra una declaración de política en una política de claves.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GrantConstraintType": "EncryptionContextEquals"
    }
  }
}
```

```

    }
  }
}
```

Véase también

- [kmsEncryptionContext:clave de contexto](#)
- [kms: EncryptionContextKeys](#)
- [kms: GrantConstraintSourceArn](#)
- [km: GranteePrincipal](#)
- [kms: GranteeServicePrincipal](#)
- [km: GrantIsForAWSResource](#)
- [kms: GrantOperations](#)
- [km: RetiringPrincipal](#)
- [kms: RetiringServicePrincipal](#)

km: GrantIsForAWSResource

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Grant IsForAWSR esource	Booleano	Single-valued	CreateGrant ListGrants RevokeGrant	Políticas de claves y políticas de IAM

Permite o deniega el [CreateGrant](#) permiso para [RevokeGrant](#) las operaciones o solo cuando un [AWS servicio integrado AWS KMS](#) llama a la operación en nombre del usuario. [ListGrants](#) Esta condición de política no permite al usuario realizar estas operaciones de concesión directamente.

La siguiente declaración de política de ejemplo utiliza la clave de condición kms:GrantIsForAWSResource. Permite a AWS los servicios integrados AWS KMS, como Amazon EBS, crear concesiones en esta clave de KMS en nombre del principal especificado.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
}
```

Véase también

- [kms: GrantConstraintSourceArn](#)
- [kms: GrantConstraintType](#)
- [km: GranteePrincipal](#)
- [kms: GranteeServicePrincipal](#)
- [kms: GrantOperations](#)
- [km: RetiringPrincipal](#)
- [kms: RetiringServicePrincipal](#)

kms: GrantOperations

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Grant Operations	Cadena	Multi-valued	CreateGrant	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para controlar el acceso a la [CreateGrant](#) operación en función de [las operaciones de concesión](#) de la solicitud. Por ejemplo, puede permitir que los usuarios

creen concesiones que deleguen el permiso para cifrar pero no para descifrar. Para obtener más información acerca de concesiones, consulte [Uso de concesiones](#).

Esta es una [clave de condición de varios valores](#). `kms:GrantOperations` compara el conjunto de operaciones de concesión en la solicitud `CreateGrant` al conjunto de operaciones de concesión de la política. Para determinar cómo se comparan estos conjuntos, debe proporcionar un operador de conjuntos `ForAnyValue` o `ForAllValues` en la condición de política. Para conocer los detalles sobre los operadores de conjuntos, consulte [Uso de múltiples claves y valores](#) en la Guía del usuario de IAM.

- `ForAnyValue`: al menos una operación de concesión en la solicitud debe coincidir con una de las operaciones de concesión en la condición de política. Se permiten otras operaciones de concesión.
- `ForAllValues`: Cada operación de subvención incluida en la solicitud debe coincidir con una operación de subvención incluida en la condición de la política. Este operador de conjuntos limita las operaciones de concesión a las especificadas en la condición de política. No requiere ninguna operación de concesión, pero prohíbe operaciones de concesión no especificadas.

`ForAllValues` también devuelve el valor `true` cuando no hay operaciones de subvención en la solicitud, pero `CreateGrant` no las permite. Si el parámetro `Operations` falta o tiene un valor nulo, la solicitud `CreateGrant` falla.

La siguiente declaración de política de clave de ejemplo utiliza la clave de condición `kms:GrantOperations` para crear concesiones solo cuando las operaciones de concesión sean `Encrypt`, `ReEncryptTo` o ambas. Si la concesión incluye cualquier otra operación, la solicitud `CreateGrant` falla.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Encrypt",
        "ReEncryptTo"
      ]
    }
  }
}
```



```

    ]
  }
}
}

```

Si cambia el operador de conjuntos en la condición de política a `ForAnyValue`, la declaración de política requerirá que al menos una de las operaciones de concesión sea `Encrypt` o `ReEncryptTo`, pero permitirá otras operaciones de concesión, como `Decrypt` o `ReEncryptFrom`.

Véase también

- [kms: GrantConstraintSourceArn](#)
- [kms: GrantConstraintType](#)
- [km: GrantsForAWSResource](#)
- [km: GranteePrincipal](#)
- [kms: GranteeServicePrincipal](#)
- [km: RetiringPrincipal](#)
- [kms: RetiringServicePrincipal](#)

km: GranteePrincipal

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
<code>kms:GranteePrincipal</code>	Cadena	Single-valued	<code>CreateGrant</code>	Políticas de IAM y de claves

Puede utilizar esta clave de condición para controlar el acceso a la [CreateGrant](#) operación en función del valor del [GranteePrincipal](#) parámetro de la solicitud. Por ejemplo, puede crear concesiones para utilizar una clave KMS cuando la entidad principal del beneficiario en la solicitud `CreateGrant` coincida con la entidad principal especificada en la declaración de la condición.

Para especificar la entidad principal del beneficiario utilice el nombre de recurso de Amazon (ARN) de una entidad principal de AWS . Entre los principales válidos se incluyen Cuentas de AWS los

usuarios de IAM, los roles de IAM, los usuarios federados y los usuarios con roles asumidos. Para obtener ayuda con la sintaxis de ARN de una entidad principal, consulte [ARN de IAM](#) en la Guía del usuario de IAM.

En el ejemplo siguiente, la declaración de la política de claves utiliza la clave de condición `kms:GranteePrincipal` para crear concesiones de una clave KMS cuando la entidad principal del beneficiario de la concesión sea `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"
    }
  }
}
```

Véase también

- [kms: GrantConstraintSourceArn](#)
- [kms: GrantConstraintType](#)
- [km: GrantsForAWSResource](#)
- [kms: GrantOperations](#)
- [kms: GranteeServicePrincipal](#)
- [km: RetiringPrincipal](#)
- [kms: RetiringServicePrincipal](#)

kms: KeyAgreementAlgorithm

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:KeyAgreementAlgorithm	Cadena	Single-valued	DeriveSharedSecret	Políticas de claves y políticas de IAM

Puede utilizar la clave de kms:KeyAgreementAlgorithm condición para controlar el acceso a la [DeriveSharedSecret](#) operación en función del valor del KeyAgreementAlgorithm parámetro de la solicitud. Para KeyAgreementAlgorithm, el único valor válido es ECDH.

Por ejemplo, la siguiente declaración de política clave utiliza la clave de kms:KeyAgreementAlgorithm condición para denegar todo acceso a DeriveSharedSecret menos que lo KeyAgreementAlgorithm sea ECDH.

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

Véase también

- [the section called “kms: KeyUsage”](#)

km: KeyOrigin

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:KeyOrigin	Cadena	Single-valued	CreateKey Operaciones de recursos de claves KMS	Políticas de IAM Políticas de claves y políticas de IAM

La clave de condición kms:KeyOrigin controla el acceso a las operaciones en función del valor de la propiedad Origin de clave KMS creada por la operación o utilizada en ella. Funciona como una condición de recurso o una condición de solicitud.

Puede utilizar esta clave de condición para controlar el acceso a la [CreateKey](#) operación en función del valor del parámetro [Origin](#) de la solicitud. Los valores válidos para Origin son AWS_KMS, AWS_CLOUDHSM, EXTERNAL_KEY_STORE y EXTERNAL.

Por ejemplo, puede crear una clave KMS solo cuando el material clave se genere en AWS KMS (AWS_KMS), solo cuando el material clave se genere en un AWS CloudHSM clúster que esté asociado a un almacén de [claves personalizado de CloudHSM](#) AWS_CLOUDHSM (), solo cuando el material clave se genere en [un almacén de claves externo](#) EXTERNAL_KEY_STORE () o solo cuando [el material clave se](#) importe de una fuente externa (). EXTERNAL

El siguiente ejemplo de declaración de política clave usa la clave de kms:KeyOrigin condición para crear una clave KMS solo cuando AWS KMS crea el material clave.

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"
      }
    }
  }
]
}

```

También puede utilizar la clave de condición `kms:KeyOrigin` para controlar el acceso a las operaciones que utilizan o administran una clave de KMS en función de la propiedad `Origin` de la clave de KMS utilizada para la operación. La operación debe ser una operación de recursos de clave KMS, es decir, una operación que está autorizada para una clave KMS en particular. Para identificar las operaciones de recursos clave KMS, en la [Tabla de acciones y recursos](#), busque un valor de KMS key en la columna `Resources` para la operación.

Por ejemplo, la siguiente política de IAM permite a las entidades principales realizar las operaciones de recursos de clave KMS especificadas, pero solo con las claves KMS de la cuenta que se crearon en un almacén de claves 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"
    }
  }
}
```

Véase también

- [km: BypassPolicyLockoutSafetyCheck](#)
- [kms: KeySpec](#)
- [kms: KeyUsage](#)

kms: KeySpec

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:KeySpec	Cadena	Single-valued	CreateKey	Políticas de IAM

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
			Operaciones de recursos de claves KMS	Políticas de claves y políticas de IAM

La clave de condición `kms:KeySpec` controla el acceso a las operaciones en función del valor de la propiedad `KeySpec` de clave KMS creada por la operación o utilizada en ella.

Puede utilizar esta clave de condición en una política de IAM para controlar el acceso a la [CreateKey](#) operación en función del valor del [KeySpec](#) parámetro de una `CreateKey` solicitud. Por ejemplo, puede utilizar esta condición para permitir a los usuarios crear solo claves KMS de cifrado simétricas o solo claves KMS HMAC.

El siguiente ejemplo la declaración de política de IAM utiliza la clave de condición `kms:KeySpec` para permitir a las entidades principales crear solo claves KMS asimétricas RSA. El permiso solo es válido cuando el `KeySpec` en la solicitud comienza con `RSA_`.

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

También puede utilizar la clave de condición `kms:KeySpec` para controlar el acceso a las operaciones que utilizan o administran una clave de KMS en función de la propiedad `KeySpec` de la clave de KMS utilizada para la operación. La operación debe ser una operación de recursos de clave KMS, es decir, una operación que está autorizada para una clave KMS en particular. Para identificar las operaciones de recursos clave de KMS, en la [tabla de acciones y recursos](#), busque un valor de `KMS key` en la columna `Resources` para la operación.

Por ejemplo, la siguiente política de IAM permite a las entidades principales realizar las operaciones de recursos de clave KMS especificadas, pero solo con las claves KMS de cifrado simétricas de la cuenta.

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

Véase también

- [km: BypassPolicyLockoutSafetyCheck](#)
- [kms: CustomerMasterKeySpec \(obsoleto\)](#)
- [km: DataKeyPairSpec](#)
- [km: KeyOrigin](#)
- [kms: KeyUsage](#)

kms: KeyUsage

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:KeyUs age	Cadena	Single-valued	CreateKey	Políticas de IAM Políticas de claves y políticas de IAM

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
			Operaciones de recursos de claves KMS	

La clave de condición `kms:KeyUsage` controla el acceso a las operaciones en función del valor de la propiedad `KeyUsage` de clave KMS creada por la operación o utilizada en ella.

Puede utilizar esta clave de condición para controlar el acceso a la [CreateKey](#) operación en función del valor del [KeyUsage](#) parámetro de la solicitud. Los valores válidos para `KeyUsage` son `ENCRYPT_DECRYPT`, `SIGN_VERIFY`, `GENERATE_VERIFY_MAC` y `KEY_AGREEMENT`.

Por ejemplo, puede crear una clave KMS solo cuando `KeyUsage` sea `ENCRYPT_DECRYPT` o denegar un permiso de usuario cuando `KeyUsage` sea `SIGN_VERIFY`.

En el siguiente ejemplo la declaración de política de IAM utiliza la clave de condición `kms:KeyUsage` para crear una clave KMS solo cuando `KeyUsage` sea `ENCRYPT_DECRYPT`.

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

También puede utilizar la clave de condición `kms:KeyUsage` para controlar el acceso a las operaciones que utilizan o administran una clave KMS en función de la propiedad `KeyUsage` de la clave KMS utilizada para la operación. La operación debe ser una operación de recursos de clave KMS, es decir, una operación que está autorizada para una clave KMS en particular. Para identificar las operaciones de recursos clave KMS, en la [Tabla de acciones y recursos](#), busque un valor de KMS key en la columna `Resources` para la operación.

Por ejemplo, la siguiente política de IAM permite a las entidades principales realizar las operaciones de recursos de clave KMS especificadas, pero solo con las claves KMS de la cuenta que se utilizan para la firma y la verificación.

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

Véase también

- [km: BypassPolicyLockoutSafetyCheck](#)
- [kms: CustomerMasterKeyUsage \(obsoleto\)](#)
- [km: KeyOrigin](#)
- [kms: KeySpec](#)

km: MacAlgorithm

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:MacAl gorithm	Cadena	Single-valued	GenerateMac VerifyMac	Políticas de claves y políticas de IAM

Puede utilizar la clave de `kms:MacAlgorithm` condición para controlar el acceso a las [VerifyMac](#) operaciones [GenerateMac](#) en función del valor del `MacAlgorithm` parámetro de la solicitud.

El siguiente ejemplo de política de claves permite a los usuarios que pueden asumir el rol de `testers` utilizar la clave KMS HMAC para generar y verificar etiquetas HMAC solo cuando el algoritmo MAC de la solicitud es `HMAC_SHA_384` o `HMAC_SHA_512`. Esta política utiliza dos declaraciones de políticas independientes, cada una con su propia condición. Si especifica más de un algoritmo MAC en una única declaración de condición, la condición requiere ambos algoritmos, en lugar de uno u otro.

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"
      }
    }
  }
}
```

Véase también

- [the section called “km: EncryptionAlgorithm”](#)
- [km: SigningAlgorithm](#)

km: MessageType

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Messa geType	Cadena	Single-valued	Sign Verify	Políticas de claves y políticas de IAM

La clave de condición kms:MessageType controla el acceso a las operaciones [Sign](#) y [Verify](#) en función del valor del parámetro MessageType de la solicitud. Los valores válidos MessageType son RAW, DIGEST y EXTERNAL_MU.

Por ejemplo, la siguiente declaración de política de claves utiliza la clave de condición kms:MessageType para utilizar una clave KMS asimétrica para firmar un mensaje, pero no un resumen de mensajes.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
}
```

```
"Action": "kms:Sign",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "kms:MessageType": "RAW"
  }
}
```

Véase también

- [the section called “km: SigningAlgorithm”](#)

km: MultiRegion

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Multi Region	Booleano	Single-valued	CreateKey Operaciones de recursos de claves KMS	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para permitir operaciones solo en claves de una región o solo en [claves de varias regiones](#). La clave de kms:MultiRegion condición controla el acceso a AWS KMS las operaciones en las claves de KMS y a la [CreateKey](#) operación en función del valor de la MultiRegion propiedad de la clave de KMS. Los valores válidos son true (de varias regiones) y false (de una sola región). Todas las claves KMS tienen una propiedad MultiRegion.

El siguiente ejemplo la declaración de política de IAM utiliza la clave de condición kms:MultiRegion para permitir a las entidades principales crear solo claves de una sola región.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
```

```

    "Bool": {
      "kms:MultiRegion": false
    }
  }
}

```

kms: MultiRegionKeyType

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:MultiRegionKeyType	Cadena	Single-valued	CreateKey Operaciones de recursos de claves KMS	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para permitir operaciones solo en [claves principales de varias regiones](#) o solo en [claves de réplicas de varias regiones](#). La clave de kms:MultiRegionKeyType condición controla el acceso a AWS KMS las operaciones en las claves de KMS y la [CreateKey](#) operación en función de la MultiRegionKeyType propiedad de la clave de KMS. Los valores válidos son PRIMARY y REPLICA. Solo las claves de varias regiones tienen una propiedad MultiRegionKeyType.

Normalmente, utilice la clave de condición kms:MultiRegionKeyType en una política de IAM para controlar el acceso a varias claves KMS. Sin embargo, dado que una clave de varias regiones puede cambiar a principal o réplica, es posible que desee utilizar esta condición en una política de clave para permitir una operación solo cuando la clave de varias regiones concreta sea una clave principal o de réplica.

Por ejemplo, la siguiente declaración de política de IAM utiliza la clave de condición kms:MultiRegionKeyType para permitir que las principales entidades programen y cancelen la eliminación de claves solo en claves de réplica de varias regiones en la Cuenta de AWS especificada.

```

{
  "Effect": "Allow",

```

```
"Action": [
  "kms:ScheduleKeyDeletion",
  "kms:CancelKeyDeletion"
],
"Resource": "arn:aws:kms:*:111122223333:key/*",
"Condition": {
  "StringEquals": {
    "kms:MultiRegionKeyType": "REPLICA"
  }
}
```

Para permitir o denegar el acceso a todas las claves de varias regiones, puede usar ambos valores o un valor nulo con kms:MultiRegionKeyType. Sin embargo, se recomienda utilizar la clave de MultiRegion condición [kms:](#) para ello.

kms: PrimaryRegion

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Prima ryRegion	Cadena (lista)	Single-valued	UpdatePri maryRegion	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para limitar las regiones de destino de una [UpdatePrimaryRegion](#) operación. Estas son las Regiones de AWS que pueden alojar las claves principales de varias regiones.

La clave de kms:PrimaryRegion condición controla el acceso a la [UpdatePrimaryRegion](#) operación en función del valor del PrimaryRegion parámetro. El PrimaryRegion parámetro especifica la [clave Región de AWS de réplica multirregional](#) que se va a convertir en principal. El valor de la condición es uno o más Región de AWS nombres, como us-east-1 o ap-southeast-2, o patrones de nombres de regiones, como eu-*

Por ejemplo, la siguiente declaración de política utiliza la clave de condición kms:PrimaryRegion para permitir que las entidades principales actualicen la región principal de una clave de varias regiones a una de las cuatro regiones 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"
      ]
    }
  }
}
```

km: ReEncryptOnSameKey

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:ReEnc ryptOnSam eKey	Booleano	Single-valued	ReEncrypt	Políticas de claves y políticas de IAM

Puede usar esta clave de condición para controlar el acceso a la [ReEncrypt](#) operación en función de si la solicitud especifica una clave KMS de destino que sea la misma que se utilizó para el cifrado original.

Por ejemplo, la siguiente declaración de política de clave utiliza la clave de condición kms:ReEncryptOnSameKey para volver a cifrar únicamente cuando la clave KMS de destino sea la misma que se utilizó para el cifrado original.

```
{
  "Effect": "Allow",
```



```

"Principal": {
  "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
},
"Action": "kms:ReEncrypt*",
"Resource": "*",
"Condition": {
  "Bool": {
    "kms:ReEncryptOnSameKey": true
  }
}
}

```

kms: RequestAlias

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Reque stAlias	Cadena (lista)	Single-valued	Operaciones criptográficas DescribeKey GetPublicKey	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para permitir una operación solo cuando la solicitud utiliza un alias determinado para identificar la clave KMS. La clave de condición `kms:RequestAlias` controla el acceso a una clave KMS utilizada en una operación criptográfica, `GetPublicKey`, o bien `DescribeKey` en función del [alias](#) que identifica esa clave KMS en la solicitud. (Esta condición de política no afecta a la [GenerateRandom](#) operación porque la operación no utiliza una clave o alias de KMS).

Esta condición admite el [control de acceso basado en atributos](#) (ABAC) AWS KMS, que permite controlar el acceso a las claves de KMS en función de las etiquetas y los alias de una clave de KMS. Puede utilizar etiquetas y alias para permitir o denegar el acceso a una clave KMS sin cambiar las políticas o las concesiones. Para obtener más información, consulte [ABAC para AWS KMS](#).

Para especificar el alias en esta condición de política, utilice un [nombre del alias](#), como, por ejemplo, `alias/project-alpha`, o un patrón de nombre de alias, como `alias/*test*`. No puede especificar una [ARN de alias](#) en el valor de esta clave de condición.

Para satisfacer esta condición, el valor del parámetro `KeyId` de la solicitud debe tener un nombre de alias o ARN de alias coincidente. Si la solicitud utiliza un [identificador de clave](#) diferente, no cumple con la condición, incluso si identifica la misma clave KMS.

Por ejemplo, la siguiente declaración de política clave permite al director llamar a la [GenerateDataKey](#) operación mediante la clave KMS. Sin embargo, esto solo está permitido cuando el valor del parámetro `KeyId` de la solicitud es `alias/finance-key` o un ARN de alias con ese nombre de alias, como por ejemplo `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"
    }
  }
}
```

No puede usar esta clave de condición para controlar el acceso a las operaciones de alias, como [CreateAlias](#) o [DeleteAlias](#). Para obtener más información sobre cómo controlar el acceso a operaciones de alias, consulte [Control del acceso a alias](#).

km: ResourceAliases

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Resou rceAliases	Cadena (lista)	Multi-valued	Operaciones de recursos de claves KMS	Solo políticas de IAM

Utilice esta clave de condición para controlar el acceso a una clave KMS según los [alias](#) que estén asociados a la clave KMS. La operación debe ser una operación de recursos de clave KMS, es decir, una operación que está autorizada para una clave KMS en particular. Para identificar las operaciones de recursos clave de KMS, en la [tabla de acciones y recursos](#), busque un valor de KMS key en la columna Resources para la operación.

Esta condición admite el control de acceso basado en atributos (ABAC) en AWS KMS. Con ABAC, puede controlar el acceso a las claves KMS en función de las etiquetas asignadas a una clave KMS y los alias asociados a una clave KMS. Puede utilizar etiquetas y alias para permitir o denegar el acceso a una clave KMS sin cambiar las políticas o las concesiones. Para obtener más información, consulte [ABAC para AWS KMS](#).

El alias debe ser único en una región Cuenta de AWS y, pero esta condición le permite controlar el acceso a varias claves de KMS de la misma región (mediante el operador de StringLike comparación) o a varias claves Regiones de AWS de KMS en distintas cuentas.

Note

La ResourceAliases condición [kms:](#) solo entra en vigor cuando la clave KMS se ajusta a la cuota de [alias por clave de KMS](#). Si una clave KMS supera esta cuota, las entidades principales que están autorizadas a usar la clave KMS mediante la condición kms:ResourceAliases se deniega el acceso a la clave KMS.

Para especificar el alias en esta condición de política, utilice un [nombre del alias](#), como, por ejemplo, alias/project-alpha, o un patrón de nombre de alias, como alias/*test*. No puede especificar una [ARN de alias](#) en el valor de esta clave de condición. Para satisfacer la condición, la

clave KMS utilizada en la operación debe tener el alias especificado. No importa si la clave KMS se identifica o cómo se identifica en la solicitud de la operación.

Se trata de una clave de condición multivalor que compara el conjunto de alias asociado a una clave KMS con el conjunto de alias de la política. Para determinar cómo se comparan estos conjuntos, debe proporcionar un operador de conjuntos `ForAnyValue` o `ForAllValues` en la condición de política. Para conocer los detalles sobre los operadores de conjuntos, consulte [Uso de múltiples claves y valores](#) en la Guía del usuario de IAM.

- `ForAnyValue`: Al menos un alias asociado a la clave de KMS debe coincidir con un alias de la condición de la política. Se permiten otros alias. Si la clave KMS no tiene alias, la condición no se cumple.
- `ForAllValues`: Todos los alias asociados a la clave de KMS deben coincidir con un alias de la política. Este operador de conjuntos limita los alias asociados con la clave KMS a los que se encuentran en la condición de política. No requiere ningún alias, pero prohíbe los alias no especificados.

Por ejemplo, la siguiente declaración de política de IAM permite al director llamar a la [GenerateDataKey](#) operación desde cualquier clave de KMS especificada Cuenta de AWS que esté asociada al `finance-key` alias. (Las políticas de clave de las claves KMS afectadas también deben permitir que la cuenta de la entidad principal las utilice para esta operación.) Para indicar que la condición se cumple cuando uno de los muchos alias que podrían estar asociados con la clave KMS es `alias/finance-key`, la condición utiliza el operador de conjuntos `ForAnyValue`.

Ya que la condición `kms:ResourceAliases` se basa en el recurso, no en la solicitud, se realiza con éxito una llamada a `GenerateDataKey` para cualquier clave de KMS asociada con el alias `finance-key`, incluso si la solicitud utiliza un [ID de clave](#) o [ARN de clave](#) para identificar la clave de 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"
  }
}
}

```

El siguiente ejemplo de declaración de política de IAM permite que la principal entidad habilite y deshabilite las claves KMS, pero solo cuando todos los alias de las claves KMS incluyen “Test”. Esta declaración de política utiliza dos condiciones. La condición con el operador de conjuntos `ForAllValues` requiere que todos los alias asociados con la clave KMS incluyan “Test” (Prueba). La condición con el operador de conjuntos `ForAnyValue` requiere que la clave KMS tenga al menos un alias con “Test” (Prueba). Sin la condición `ForAnyValue`, esta declaración de política habría permitido a la entidad principal utilizar claves KMS que no tenían alias.

```

{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": [
    "kms:EnableKey",
    "kms:DisableKey"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "ForAllValues:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    },
    "ForAnyValue:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    }
  }
}

```

kms:ReplicaRegion

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:ReplicaRegion	Cadena (lista)	Single-valued	Replicate Key	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para limitar el número de Regiones de AWS veces que un principal puede replicar una [clave multirregional](#). La clave de kms:ReplicaRegion condición controla el acceso a la [ReplicateKey](#) operación en función del valor del [ReplicaRegion](#) parámetro de la solicitud. Este parámetro especifica la Región de AWS para la nueva [clave de réplica](#).

El valor de la condición es uno o más Región de AWS nombres, como us-east-1 o ap-southeast-2, o patrones de nombres, como eu-*. Para obtener una lista de los nombres de Regiones de AWS esos AWS KMS soportes, consulte los [AWS Key Management Service puntos finales y las cuotas](#) en. Referencia general de AWS

Por ejemplo, la siguiente declaración de política clave utiliza la clave de kms:ReplicaRegion condición para permitir que los directores llamen a la [ReplicateKey](#) operación solo cuando el valor del ReplicaRegion parámetro es una de las regiones 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"
      ]
    }
  }
}
```

```
}
}
```

Esta clave de condición controla el acceso únicamente a la [ReplicateKey](#) operación. Para controlar el acceso a la [UpdatePrimaryRegion](#) operación, utilice la clave de PrimaryRegion condición [kms:](#).

km: RetiringPrincipal

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:RetiringPrincipal	Cadena (lista)	Single-valued	CreateGrant	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para controlar el acceso a la [CreateGrant](#) operación en función del valor del [RetiringPrincipal](#) parámetro de la solicitud. Por ejemplo, puede crear concesiones para utilizar una clave KMS cuando el RetiringPrincipal de la solicitud CreateGrant coincida con el RetiringPrincipal de la declaración de la condición.

Para especificar la entidad principal de retirada utilice el nombre de recurso de Amazon (ARN) de una entidad principal de AWS . Entre los principales válidos se incluyen Cuentas de AWS los usuarios de IAM, los roles de IAM, los usuarios federados y los usuarios con roles asumidos. Para obtener ayuda con la sintaxis de ARN de una entidad principal, consulte [ARN de IAM](#) en la Guía del usuario de IAM.

El siguiente ejemplo de declaración de política de claves permite a un usuario crear concesiones para la clave KMS. La clave de condición kms:RetiringPrincipal restringe el permiso a las solicitudes CreateGrant en las que la entidad principal a la cual se va a retirar la concesión sea el 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"
  }
}
```

Véase también

- [kms: GrantConstraintSourceArn](#)
- [kms: GrantConstraintType](#)
- [km: GranteePrincipal](#)
- [kms: GranteeServicePrincipal](#)
- [km: GrantsForAWSResource](#)
- [kms: GrantOperations](#)
- [kms: RetiringServicePrincipal](#)

kms: GrantConstraintSourceArn

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Grant Constrain tSourceArn	ARN	Single-valued	CreateGrant	Políticas de claves y políticas de IAM

Puede usar esta clave de condición para controlar el acceso a la [CreateGrant](#) operación en función del SourceArn valor del parámetro [Constraints](#) de la solicitud. La restricción de SourceArn concesión consiste, de hecho, en introducir una clave de condición [aws:SourceArn](#) global en la concesión.

El siguiente ejemplo de declaración de política clave utiliza la clave de kms:GrantConstraintSourceArn condición para permitir la creación de concesiones para una

clave de KMS solo cuando la SourceArn restricción de la concesión coincide con un patrón ARN específico de una tabla de DynamoDB.

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

Véase también

- [kms: GrantConstraintType](#)
- [km: GranteePrincipal](#)
- [kms: GranteeServicePrincipal](#)
- [km: GrantsForAWSResource](#)
- [kms: GrantOperations](#)
- [km: RetiringPrincipal](#)
- [kms: RetiringServicePrincipal](#)

kms: GranteeServicePrincipal

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:Grant eeService Principal	Cadena	Single-valued	CreateGrant	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para controlar el acceso a la [CreateGrant](#) operación en función del valor del [GranteeServicePrincipal](#) parámetro de la solicitud. El valor es el [principal de AWS servicio](#) especificado como beneficiario de la subvención (por ejemplo, `service-name.amazonaws.com`).

El siguiente ejemplo de declaración de política clave usa la clave de `kms:GranteeServicePrincipal` condición para permitir la creación de concesiones para una clave de KMS solo cuando el principal de servicio del concesionario de la concesión lo es. `service-name.amazonaws.com`

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

Véase también

- [kms: GrantConstraintSourceArn](#)
- [kms: GrantConstraintType](#)
- [km: GranteePrincipal](#)
- [km: GrantsForAWSResource](#)
- [kms: GrantOperations](#)
- [km: RetiringPrincipal](#)
- [kms: RetiringServicePrincipal](#)

kms:RetiringServicePrincipal

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:RetiringServicePrincipal	Cadena	Single-valued	CreateGrant	Políticas de claves y políticas de IAM

Puede utilizar esta clave de condición para controlar el acceso a la [CreateGrant](#) operación en función del valor del [RetiringServicePrincipal](#) parámetro de la solicitud. El valor es el [principal de AWS servicio](#) especificado como el principal de la subvención que se retira (por ejemplo, `service-name.amazonaws.com`).

El siguiente ejemplo de declaración de política clave usa la clave de `kms:RetiringServicePrincipal` condición para permitir la creación de concesiones para una clave de KMS solo cuando el principal de servicio de la concesión se retira. `service-name.amazonaws.com`

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

Véase también

- [kms: GrantConstraintSourceArn](#)
- [kms: GrantConstraintType](#)

- [km: GranteePrincipal](#)
- [kms: GranteeServicePrincipal](#)
- [km: GrantIsForAWSResource](#)
- [kms: GrantOperations](#)
- [km: RetiringPrincipal](#)

kms: RotationPeriodInDays

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:RotationPeriodInDays	Numérico	Single-valued	EnableKeyRotation	Políticas de claves y políticas de IAM

Puede usar esta clave de condición para limitar los valores que los directores pueden especificar en el `RotationPeriodInDays` parámetro de una [EnableKeyRotation](#) solicitud.

`RotationPeriodInDays` Especifica el número de días entre cada fecha de rotación automática de claves. AWS KMS permite especificar un período de rotación de entre 90 y 2560 días, pero puede utilizar la clave de `kms:RotationPeriodInDays` condición para restringir aún más el período de rotación, por ejemplo, imponiendo un período de rotación mínimo dentro del rango válido.

Por ejemplo, la siguiente declaración de política de claves utiliza la clave de condición `kms:RotationPeriodInDays` para impedir que las entidades principales habiliten rotación de claves si el periodo de rotación es inferior o igual a 180 días.

```
{
  "Effect": "Deny",
  "Action": "kms:EnableKeyRotation",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
    "NumericLessThanEquals" : {
      "kms:RotationPeriodInDays" : "180"
    }
  }
}
```

```
}
}
```

km: ScheduleKeyDeletionPendingWindowInDays

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:ScheduleKeyDeletionPendingWindowInDays	Numérico	Single-valued	ScheduleKeyDeletion	Políticas de claves y políticas de IAM

Puede usar esta clave de condición para limitar los valores que los directores pueden especificar en el PendingWindowInDays parámetro de una [ScheduleKeyDeletion](#) solicitud.

PendingWindowInDaysEspecifica el número de días que deben AWS KMS transcurrir antes de eliminar una clave. AWS KMS permite especificar un período de espera de entre 7 y 30 días, pero puede usar la clave de kms:ScheduleKeyDeletionPendingWindowInDays condición para restringir aún más el período de espera, por ejemplo, imponiendo un período de espera mínimo dentro del rango válido.

Por ejemplo, la siguiente declaración de política de claves utiliza la clave de condición kms:ScheduleKeyDeletionPendingWindowInDays para impedir que las entidades principales programen la eliminación de claves si el periodo de espera es inferior o igual a 21 días.

```
{
  "Effect": "Deny",
  "Action": "kms:ScheduleKeyDeletion",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
    "NumericLessThanEquals" : {
      "kms:ScheduleKeyDeletionPendingWindowInDays" : "21"
    }
  }
}
```

}

km: SigningAlgorithm

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:SigningAlgorithm	Cadena	Single-valued	Sign Verify	Políticas de claves y políticas de IAM

Puede usar la clave de `kms:SigningAlgorithm` condición para controlar el acceso a las operaciones de [firma](#) y [verificación](#) en función del valor del [SigningAlgorithm](#) parámetro de la solicitud. Esta clave de condición no afecta a las operaciones realizadas fuera de AWS KMS, como la verificación de firmas con la clave pública en un par de claves KMS asimétricas fuera de. AWS KMS

La siguiente política de claves de ejemplo permite a los usuarios que puedan asumir el rol `testers` utilizar la clave KMS para firmar mensajes solo cuando el algoritmo de firma utilizado para la solicitud sea un 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*"
    }
  }
}
```

Véase también

- [km: EncryptionAlgorithm](#)
- [the section called “km: MacAlgorithm”](#)

- [the section called “km: MessageType”](#)

kms: TrailingDaysWithoutKeyUsage

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:TrailingDaysWithoutKeyUsage	Numérico	Single-valued	DisableKey ScheduleKeyDeletion	Políticas de claves y políticas de IAM

La clave de kms:TrailingDaysWithoutKeyUsage condición representa el número de días finales sin operaciones criptográficas en una clave de KMS, calculados a partir de la última operación criptográfica realizada correctamente o a partir de la fecha de creación de la clave de KMS o TrackingStartDate si la clave nunca se ha utilizado. Para obtener más información sobre la fecha de inicio del seguimiento, consulte. [Comprenda el período de seguimiento del uso](#) Puede utilizar esta clave de condición en las políticas clave y en las políticas de IAM para controlar el acceso a [ScheduleKeyDeletion](#) las [DisableKey](#) operaciones.

Esta clave de condición es una condición numérica de un solo valor. El valor de los últimos días siempre se redondea a la baja. Por ejemplo, si una clave se utilizó por última vez hace 89,9 días, el valor es 89.

El kms:TrailingDaysWithoutKeyUsage se calcula de la siguiente manera:

- Para las claves creadas el mismo día o después TrackingStartDate que no se hayan utilizado desde que se inició el seguimiento:

Fórmula: Fecha actual - KeyCreationDate

Por ejemplo, si la fecha actual es el 30 de abril de 2026 y la clave se creó el 20 de abril de 2026, el valor es 10.

- En el caso de las claves creadas antes TrackingStartDate que no se hayan utilizado desde que se inició el seguimiento:

Fórmula: Fecha actual - TrackingStartDate

Por ejemplo, si la fecha actual es el 30 de abril de 2026 y el TrackingStartDate 15 de abril de 2026, el valor es 15, independientemente de cuándo se creó la clave.

- Para las claves que se han utilizado desde que se inició el seguimiento, independientemente de cuándo se creó la clave:

Fórmula: Fecha actual: Timestamp de KeyLastUsage

El siguiente ejemplo de declaración de política clave deniega las DisableKey operaciones ScheduleKeyDeletion y cuando la clave se ha utilizado en los últimos 90 días o cuando la clave no tiene un uso registrado y se creó hace 90 días o menos o el seguimiento se inició hace 90 días o menos. Esto evita que se eliminen o deshabiliten accidentalmente las claves que se utilizan activamente o las que se han creado recientemente.

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

Véase también

- [Examine la última operación criptográfica realizada con una clave KMS](#)
- [km: ScheduleKeyDeletionPendingWindowInDays](#)

km: ValidTo

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:ValidTo	Timestamp	Single-valued	ImportKey Material	Políticas de claves y políticas de IAM

La clave de kms:ValidTo condición controla el acceso a la [ImportKeyMaterial](#) operación en función del valor del [ValidTo](#) parámetro de la solicitud, que determina cuándo caduca el material clave importado. El valor se expresa en [tiempo Unix](#).

De forma predeterminada, el parámetro ValidTo es obligatorio en las solicitudes ImportKeyMaterial. Sin embargo, si el valor del [ExpirationModel](#) parámetro es KEY_MATERIAL_DOES_NOT_EXPIRE, no es válido. ValidTo También puede usar la clave de ExpirationModel condición [kms:](#) para requerir el ExpirationModel parámetro o un valor de parámetro específico.

La siguiente declaración de política de ejemplo permite a un usuario importar material de claves en una clave KMS. La clave de condición kms:ValidTo limita el permiso a las solicitudes ImportKeyMaterial en las que el valor ValidTo sea menor o igual que 1546257599.0 (31 de diciembre de 2018 11:59:59 p. m.).

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "NumericLessThanEquals": {
      "kms:ValidTo": "1546257599.0"
    }
  }
}
```

Véase también

- [km: ExpirationModel](#)
- [km: WrappingAlgorithm](#)
- [km: WrappingKeySpec](#)

kms: ViaService

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:ViaService	Cadena	Single-valued	Operaciones de recursos de claves KMS	Políticas de claves y políticas de IAM

La clave de kms:ViaService condición limita el uso de una clave KMS a las solicitudes especificadas Servicios de AWS. Esta clave de condición solo se aplica a las [sesiones de acceso directo](#). Puede especificar uno o varios servicios en cada clave de condición kms:ViaService. La operación debe ser una operación de recursos de clave KMS, es decir, una operación que está autorizada para una clave KMS en particular. Para identificar las operaciones de recursos clave KMS, en la [Tabla de acciones y recursos](#), busque un valor de KMS key en la columna Resources para la operación.

Por ejemplo, la siguiente declaración de una política de claves utiliza la clave de condición kms:ViaService para permitir que se use una [clave administrada por el cliente](#) para las acciones especificadas solo cuando la solicitud provenga de Amazon EC2 o Amazon RDS en la región EE.UU. Oeste (Oregón) en nombre 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"
      ]
    }
  }
}

```

También puede utilizar un clave de condición `kms:ViaService` para denegar permisos para usar una clave KMS cuando la solicitud provenga de determinados servicios. Por ejemplo, la siguiente declaración de una política de claves utiliza una clave de condición `kms:ViaService` para impedir que se utilice una clave administrada por el cliente para las operaciones `Encrypt` cuando la solicitud provenga de AWS Lambda en nombre 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

Cuando se utiliza la clave de condición `kms:ViaService`, el servicio realiza la solicitud en nombre de una entidad principal de la Cuenta de AWS. Estas entidades principales deben tener los siguientes permisos:

- Permiso para usar la clave KMS. La entidad principal debe conceder estos permisos al servicio integrado para que pueda utilizar la clave administrada por el cliente en nombre de la entidad principal. Para obtener más información, consulte [Uso del AWS KMS cifrado con AWS servicios](#).
- Permiso para utilizar el servicio integrado. Para obtener más información sobre cómo dar a los usuarios acceso a un AWS servicio que se integra con él AWS KMS, consulte la documentación del servicio integrado.

Todas las [Claves administradas por AWS](#) utilizan una clave de condición `kms:ViaService` incluida en su documento de política de claves. Esta condición permite que solo se utilice la clave KMS para las solicitudes que proceden del servicio que ha creado la clave KMS. Para ver la política clave de un Clave administrada de AWS, utilice la [GetKeyPolicy](#) operación.

La clave de condición `kms:ViaService` es válida en IAM y las declaraciones de políticas de clave. Los servicios que especifique deben estar [integrados con AWS KMS](#) y admitir la clave de condición `kms:ViaService`.

Servicios que admiten el **kms:** clave de `ViaService` condición

En la siguiente tabla se enumeran AWS los servicios que están integrados con la clave de `kms:ViaService` condición en las claves administradas por el cliente AWS KMS y que permiten su uso. Es posible que los servicios de esta tabla no estén disponibles en todas las regiones. Utilice el `.amazonaws.com` sufijo del AWS KMS `ViaService` nombre en todas las AWS particiones.

 Note

Es posible que deba desplazarse horizontal o verticalmente para ver todos los datos de esta tabla.

Nombre del servicio	AWS KMS ViaService nombre
Amazon AI Operations	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 Transform MGN	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
Copia del modelo de Amazon Bedrock	bedrock. <i>AWS_region</i> .amazonaws.com
Amazon Chime SDK	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 de Amazon	codeguru-reviewer. <i>AWS_region</i> .amazonaws.com

Nombre del servicio	AWS KMS ViaService nombre
Amazon Comprehend	comprehend. <i>AWS_region</i> .amazonaws.com
Amazon Connect Customer	connect. <i>AWS_region</i> .amazonaws.com
Perfiles de clientes de Amazon Connect Customer	profile. <i>AWS_region</i> .amazonaws.com
Amazon Q en 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 (EBS solo)
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

Nombre del servicio	AWS KMS ViaService nombre
Amazon ElastiCache	Incluye ambos ViaService nombres en el valor de la clave de condición: • 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 Resolución de la entidad	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

Nombre del servicio	AWS KMS ViaService nombre
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 Pruebas de aplicaciones	apptest. <i>AWS_region</i> .amazonaws.com
Amazon Managed Blockchain	managedblockchain. <i>AWS_region</i> .amazonaws.com

Nombre del servicio	AWS KMS ViaService nombre
Amazon Managed Streaming para Apache Kafka (Amazon MSK)	kafka. <i>AWS_region</i> .amazonaws.com
Amazon Managed Workflows para 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 Servicio Amazon	es. <i>AWS_region</i> .amazonaws.com , aoss. <i>AWS_region</i> .amazonaws.com
Paquetes OpenSearch personalizados de 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
Información de rendimiento de Amazon RDS	rds. <i>AWS_region</i> .amazonaws.com
Amazon Redshift	redshift. <i>AWS_region</i> .amazonaws.com
Editor de consultas de Amazon Redshift V2	sqlworkbench. <i>AWS_region</i> .amazonaws.com

Nombre del servicio	AWS KMS ViaService nombre
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
Almacén de datos replicados de Amazon	ards. <i>AWS_region</i> .amazonaws.com
Amazon SageMaker AI	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
Tablas de 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

Nombre del servicio	AWS KMS ViaService nombre
AWS Storage Gateway	storagegateway. <i>AWS_region</i> <i>n</i> .amazonaws.com
Administrador de incidentes de AWS Systems Manager	ssm-incidents. <i>AWS_region</i> .amazonaws.com
Administrador de incidentes de AWS Systems Manager Contactos	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
Acceso verificado de 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

km: WrappingAlgorithm

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:WrappingAlgorithm	Cadena	Single-valued	GetParametersForImport	Políticas de claves y políticas de IAM

Esta clave de condición controla el acceso a la [GetParametersForImport](#) operación en función del valor del [WrappingAlgorithm](#) parámetro de la solicitud. Puede utilizar esta condición para exigir que las entidades principales usen un algoritmo para cifrar el material durante el proceso de importación. Las solicitudes de la clave pública y el token de importación necesarios no se realizan cuando especifican un algoritmo de encapsulamiento diferente.

La siguiente declaración de política de clave de ejemplo utiliza la clave de condición kms:WrappingAlgorithm para conceder al usuario de ejemplo permiso para llamar a la operación GetParametersForImport, pero le impide utilizar el algoritmo de encapsulamiento RSAES_OAEP_SHA_1. Cuando WrappingAlgorithm en la solicitud GetParametersForImport es RSAES_OAEP_SHA_1, se produce un error en la operación.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:WrappingAlgorithm": "RSAES_OAEP_SHA_1"
    }
  }
}
```

Véase también

- [km: ExpirationModel](#)

- [km: ValidTo](#)
- [km: WrappingKeySpec](#)

km: WrappingKeySpec

AWS KMS claves de condición	Tipo de condición	Tipo de valor	Operaciones de la API	Tipo de política
kms:WrappingKeySpec	Cadena	Single-valued	GetParametersForImport	Políticas de claves y políticas de IAM

Esta clave de condición controla el acceso a la [GetParametersForImport](#) operación en función del valor del [WrappingKeySpec](#) parámetro de la solicitud. Puede utilizar esta condición para exigir que las entidades principales usen un determinado tipo de clave pública durante el proceso de importación. Si la solicitud especifica un tipo de clave diferente, produce un error.

Como el único valor válido del parámetro WrappingKeySpec es RSA_2048, al impedir que los usuarios utilicen este valor, se evita que utilicen la operación GetParametersForImport.

En el ejemplo siguiente, la declaración de la política utiliza la clave de condición kms:WrappingAlgorithm para requerir que el parámetro WrappingKeySpec de la solicitud sea RSA_4096.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:WrappingKeySpec": "RSA_4096"
    }
  }
}
```

Véase también

- [km: ExpirationModel](#)
- [km: ValidTo](#)
- [km: WrappingAlgorithm](#)

AWS KMS claves de condición para plataformas certificadas

AWS KMS proporciona claves de condición para respaldar la certificación criptográfica para Nitro Enclaves y [AWS NitroTPM](#). AWS Nitro Enclaves es una funcionalidad de Amazon EC2 que le permite crear entornos informáticos aislados [denominados](#) enclaves para proteger y procesar datos altamente confidenciales. NitroTPM extiende una funcionalidad de certificación similar a las instancias de EC2.

Cuando realiza operaciones de [descifrado](#), [DeriveSharedSecretGenerateDataKeyGenerateDataKeyPair](#), o [GenerateRandom](#) API con un documento de certificación firmado, estas API cifran el texto sin formato de la respuesta con la clave pública del documento de certificación y devuelven texto cifrado en lugar de texto simple. Este texto cifrado solo se puede descifrar con la clave privada del enclave. Para obtener más información, consulte [Soporte de atestación criptográfica en AWS KMS](#).

Note

Si no proporciona una política de claves al crear una clave, crea una para usted. AWS KMS Esta [política de claves predeterminada](#) concede a las Cuentas de AWS propietarias de la clave de KMS acceso completo a ella y permite que la cuenta utilice políticas de IAM para permitir el acceso a la clave. Esta política permite realizar todas las acciones, como [desencriptar](#). AWS recomienda aplicar la entidad principal de [Aplicar permisos de privilegios mínimos](#) a las políticas de claves de KMS. También puede restringir el acceso al [modificar la acción de la política de claves de KMS](#) para kms : * en [NotAction:kms:Decrypt](#).

Las siguientes claves de condición permiten limitar los permisos para estas operaciones en función del contenido del documento de conformidad firmado. Antes de permitir una operación, AWS KMS compara el documento de certificación con los valores de estas claves de AWS KMS condición.

Claves de condición para Nitro Enclaves

Las siguientes claves de condición son específicas de la certificación de Nitro Enclaves:

km: RecipientAttestation:ImageSha384

AWS KMS Claves de estado	Tipo de condición	Tipo de valor	Operaciones de API	Tipo de política
kms:RecipientAttestation:ImageSha384	Cadena	Single-valued	Decrypt DeriveSharedSecret GenerateDataKey GenerateDataKeyPair GenerateRandom	Políticas de claves y políticas de IAM

La clave de condición kms:RecipientAttestation:ImageSha384 controla el acceso a Decrypt, DeriveSharedSecret, GenerateDataKey, GenerateDataKeyPair y GenerateRandom con una clave KMS cuando el resumen de la imagen del documento de certificación firmado de la solicitud coincide con el valor de la clave de condición. El valor ImageSha384 corresponde a PCR0 en el documento de certificación. Esta clave de condición solo entra en vigor cuando el Recipient parámetro de la solicitud especifica un documento de certificación firmado para un enclave de AWS Nitro.

Este valor también se incluye en los [CloudTrail eventos](#) relacionados con las solicitudes de enclaves de AWS KMS Nitro.

Por ejemplo, la siguiente declaración de política clave permite al data-processing rol usar la clave KMS para [descifrar](#), [DeriveSharedSecret](#), [GenerateDataKey](#), [GenerateDataKeyPair](#), y las operaciones. [GenerateRandom](#) La clave de condición kms:RecipientAttestation:ImageSha384 permite

las operaciones solo cuando el valor de resumen de imagen (PCR0) del documento de certificación en la solicitud coincida con el valor de resumen de imagen de la condición. Esta clave de condición solo entra en vigor cuando el Recipient parámetro de la solicitud especifica un documento de certificación firmado para un AWS enclave de Nitro.

Si la solicitud no incluye un documento de certificación válido de un enclave de AWS Nitro, se deniega el permiso porque no se cumple esta condición.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:DeriveSharedSecret",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
      "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

kms: PCR_ID RecipientAttestation:PCR < >

AWS KMS Claves de condición	Tipo de condición	Tipo de valor	Operaciones de API	Tipo de política
kms:Recip ientAttes tation:PC R<PCR_ID>	Cadena	Single-valued	Decrypt DeriveSha redSecret	Políticas de claves y políticas de IAM

AWS KMS Claves de condición	Tipo de condición	Tipo de valor	Operaciones de API	Tipo de política
			Generated ataKey	
			Generated ataKeyPair	
			GenerateR andom	

La clave de condición `kms:RecipientAttestation:PCR<PCR_ID>` controla el acceso a `Decrypt`, `DeriveSharedSecret`, `GenerateDataKey`, `GenerateDataKeyPair` y `GenerateRandom` con una clave KMS solo cuando los registros de configuración de la plataforma (PCR) del documento de certificación firmado en la solicitud coincidan con los PCR de la clave de condición. Esta clave de condición solo entra en vigor cuando el `Recipient` parámetro de la solicitud especifica un documento de certificación firmado desde un enclave de AWS Nitro.

Este valor también se incluye en los [CloudTrail eventos](#) que representan solicitudes de acceso a AWS KMS enclaves de Nitro.

Para especificar un valor de PCR, utilice el siguiente formato. Concatene el ID de PCR con el nombre de la clave de condición. Puede especificar un ID de PCR que identifique una de las [seis mediciones del enclave](#) o un ID de PCR personalizado que haya definido para un caso de uso específico. El valor de PCR debe ser una cadena hexadecimal en minúsculas de hasta 96 bytes.

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

Por ejemplo, la siguiente clave de condición especifica un valor particular para PCR1, que corresponde al hash del kernel utilizado para el enclave y el proceso de arranque.

```
kms:RecipientAttestation:PCR1:
  "abc1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcde
```

Por ejemplo, la siguiente declaración de política de claves permite al rol `data-processing` utilizar la clave de KMS para la operación [Decrypt](#).

La clave de condición kms:RecipientAttestation:PCR en esta declaración permite la operación solo cuando el valor PCR1 del documento de conformidad firmado en la solicitud coincide con el valor kms:RecipientAttestation:PCR1 de la condición. Use el operador de política StringEqualsIgnoreCase para requerir una comparación entre mayúsculas y minúsculas de los valores de PCR.

Si la solicitud no incluye un documento de certificación, se deniega el permiso porque esta condición no se cumple.

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

Claves de condición de NitroTPM

Las siguientes claves de condición son específicas de la certificación de NitroTPM:

kms: PCR_ID RecipientAttestation:NitroTPMPCR < >

AWS KMS Claves de condición	Tipo de condición	Tipo de valor	Operaciones de API	Tipo de política
kms:Recip ientAttes tation:Ni troTPMPCR <PCR_ID>	Cadena	Single-valued	Decrypt DeriveSha redSecret	Políticas de claves y políticas de IAM

AWS KMS Claves de condición	Tipo de condición	Tipo de valor	Operaciones de API	Tipo de política
			Generated ataKey	
			Generated ataKeyPair	
			GenerateR andom	

La clave de condición `kms:RecipientAttestation:NitroTPMPCR<PCR_ID>` controla el acceso a `Decrypt`, `DeriveSharedSecret`, `GenerateDataKey`, `GenerateDataKeyPair` y `GenerateRandom` con una clave KMS solo cuando los registros de configuración de la plataforma (PCR) del documento de certificación firmado en la solicitud coincidan con los PCR de la clave de condición. Esta clave de condición solo entra en vigor cuando el parámetro `Recipient` de la solicitud especifica un documento de certificación firmado de NitroTPM.

Este valor también se incluye en [CloudTraillos eventos](#) que representan AWS KMS solicitudes a NitroTPM.

Para especificar un valor de PCR, utilice el siguiente formato. Concatene el ID de PCR con el nombre de la clave de condición. El valor de PCR debe ser una cadena hexadecimal en minúsculas de hasta 96 bytes.

```
"kms:RecipientAttestation:NitroTPMPCRPCR_ID": "PCR_value"
```

Por ejemplo, la siguiente clave de condición especifica un valor particular para PCR4:

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

Por ejemplo, la siguiente declaración de política de claves permite al rol `data-processing` utilizar la clave de KMS para la operación [Decrypt](#).

La clave de `kms:RecipientAttestation:NitroTPMPCR` condición de esta declaración permite la operación solo cuando el valor de PCR4 del documento de certificación firmado de la solicitud

coincide con el valor de la condición. `kms:RecipientAttestation:NitroTPMPCR4` Use el operador de política `StringEqualsIgnoreCase` para requerir una comparación entre mayúsculas y minúsculas de los valores de PCR.

Si la solicitud no incluye un documento de certificación, se deniega el permiso porque esta condición no se cumple.

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

Aplicar permisos de privilegios mínimos

Dado que sus claves KMS protegen la información confidencial, le recomendamos que siga el principio del acceso de privilegio mínimo. Cuando defina las políticas de claves, delegue los permisos mínimos necesarios para realizar una tarea. Solo permita todas las acciones (`kms:*`) en una política de claves KMS si tiene previsto restringir aún más los permisos con políticas de IAM adicionales. Si tiene previsto administrar los permisos con políticas de IAM, limite quién tiene la capacidad de crear y adjuntar políticas de IAM a las entidades principales de IAM y [supervise los cambios en las políticas](#).

Si permite todas las acciones (`kms:*`) tanto en la política de claves como en la política de IAM, la entidad principal tendrá permisos administrativos y de uso para la clave KMS. Como práctica recomendada de seguridad, se recomienda delegar estos permisos únicamente a entidades principales específicas. Puede hacerlo nombrando explícitamente a la entidad principal en la política de claves o limitando las entidades principales a las que se adjunta la política de IAM. También puede usar [claves de condición](#) para restringir los permisos. Por ejemplo, puede usar la [aws:PrincipalTag](#) para permitir todas las acciones si la entidad principal que realiza la llamada a la API tiene la etiqueta especificada en la regla de condición.

Para obtener ayuda para comprender cómo se evalúan las declaraciones de políticas AWS, consulte la [lógica de evaluación de políticas](#) en la Guía del usuario de IAM. Recomendamos revisar este tema antes de redactar políticas para reducir la posibilidad de que su política tenga efectos no deseados, como proporcionar acceso a entidades principales que no deberían tener acceso.

 Tip

Cuando pruebe una aplicación en un entorno que no sea de producción, utilice [Analizador de acceso de IAM](#) para aplicar los privilegios mínimos a sus políticas de IAM.

Si utiliza usuarios de IAM en lugar de funciones de IAM, le recomendamos encarecidamente que habilite la autenticación AWS [multifactor](#) (MFA) para mitigar la vulnerabilidad de las credenciales a largo plazo. Puede utilizar la MFA para hacer lo siguiente:

- Requerir que los usuarios validen sus credenciales con MFA antes de realizar acciones privilegiadas, como programar la eliminación de claves.
- Dividir la propiedad de una contraseña de cuenta de administrador y el dispositivo de MFA entre varias personas para implementar la autorización dividida.

Más información

- [AWS políticas gestionadas para las funciones laborales](#)
- [Techniques for writing least privilege IAM policies](#)

Implementación de permisos de privilegio mínimo

Cuando concedas permiso a un AWS servicio para usar una clave de KMS, asegúrate de que el permiso sea válido solo para los recursos a los que el servicio debe acceder en tu nombre. Esta estrategia de privilegios mínimos ayuda a evitar el uso no autorizado de una clave KMS cuando las solicitudes se transfieren entre AWS servicios.

Para implementar una estrategia de privilegios mínimos, se recomienda utilizar las claves de condición del contexto de AWS KMS cifrado y las claves de condición del ARN de origen global o de la cuenta de origen.

Uso de claves de condición de contexto de cifrado

La forma más eficaz de implementar los permisos con privilegios mínimos cuando se utilizan AWS KMS recursos es incluir las claves [kms:EncryptionContext:context-key](#) o [kms:EncryptionContextKeys](#) condicionales en la política que permite a los directores realizar operaciones AWS KMS criptográficas. Estas claves de condición son especialmente eficaces porque asocian el permiso al [contexto de cifrado](#) que está enlazado al texto cifrado cuando el recurso está cifrado.

[Utilice las claves de condiciones del contexto de cifrado únicamente cuando la acción de la declaración de política CreateGrant sea una operación criptográfica AWS KMS simétrica que utilice un EncryptionContext parámetro, como operaciones como Decrypt, GenerateDataKey](#) (Para ver una lista de las operaciones admitidas, consulte [kms:EncryptionContext:context-key](#) o [kms:EncryptionContextKeys](#)). Si utiliza estas claves de condición para permitir otras operaciones, por ejemplo [DescribeKey](#), se denegará el permiso.

Establezca el valor en el contexto de cifrado que utiliza el servicio cuando cifra el recurso. Esta información generalmente suele estar disponible en el capítulo Seguridad de la documentación del servicio. Por ejemplo, el [contexto de cifrado de AWS Proton](#) identifica el recurso AWS Proton y su plantilla asociada. El [contexto de cifrado AWS Secrets Manager](#) identifica el secreto y su versión. El [contexto de cifrado para Amazon Location](#) identifica el rastreador o la recopilación.

El siguiente ejemplo de declaración de política de claves permite a Amazon Location Service crear concesiones en nombre de usuarios autorizados. Esta declaración de política limita el permiso mediante el uso de las claves [kms: ViaService](#), [kms: CallerAccount](#) y `kms:EncryptionContext:context-key` condition para vincular el permiso a un recurso de rastreo concreto.

```
{
  "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 claves de condición `aws:SourceArn` o `aws:SourceAccount`

Cuando la entidad principal de una declaración de política de claves es una [entidad principal del servicio de AWS](#), recomendamos encarecidamente que utilice las claves de condición globales `aws:SourceArn` o `aws:SourceAccount`, además de la clave de condición `kms:EncryptionContext:context-key`. El ARN y los valores de la cuenta se incluyen en el contexto de autorización solo cuando una solicitud proviene de AWS KMS de otro servicio de AWS. Esta combinación de condiciones implementa los permisos de privilegio mínimo y evita un potencial [escenario suplente confuso](#). Las entidades principales de servicio no suelen utilizarse como entidades principales en una política de claves, pero algunos servicios de AWS, como AWS CloudTrail, por ejemplo, sí lo requieren.

Para utilizar las claves de condición globales `aws:SourceArn` o `aws:SourceAccount`, establezca el valor en el nombre de recurso de Amazon (ARN) o la cuenta del recurso que se está cifrando. Por ejemplo, en una declaración de política de claves que da permiso a AWS CloudTrail para cifrar una traza, establezca el valor de `aws:SourceArn` al ARN de la traza. Siempre que sea posible, utilice `aws:SourceArn`, que es más específico. Establezca el valor en el ARN o un patrón ARN con caracteres comodín. Si no conoce el ARN del recurso, utilice `aws:SourceAccount` en su lugar.

Note

Si el ARN de un recurso incluye caracteres que no están permitidos en una política de AWS KMS, no puede usar ese ARN de recurso en el valor de la clave de condición. `aws:SourceArn` En cambio, utilice la clave de condición `aws:SourceAccount`. Para obtener más información sobre las reglas del documento de política de claves, consulte [Formato de la política de claves](#).

En el siguiente ejemplo de política de claves, la entidad principal que obtiene los permisos es la entidad principal del servicio AWS CloudTrail, `cloudtrail.amazonaws.com`. Para implementar el privilegio mínimo, esta política utiliza las claves de condición `aws:SourceArn` y `kms:EncryptionContext:context-key`. La declaración de política permite a CloudTrail utilizar

la clave KMS para [generar la clave de datos](#) que se utiliza para cifrar un rastro. Las condiciones `aws:SourceArn` y `kms:EncryptionContext:context-key` se evalúan de forma independiente. Cualquier solicitud de uso de la clave KMS para la operación especificada debe cumplir ambas condiciones.

Para restringir el permiso del servicio a la traza `finance` en la cuenta de ejemplo (111122223333) y la región `us-west-2`, esta declaración de política establece la clave de condición `aws:SourceArn` del ARN de una traza concreta. La declaración de condición utiliza el [ArnEquals](#) operador para garantizar que cada elemento del ARN se evalúe de forma independiente al coincidir. En el ejemplo también se utiliza la clave de condición `kms:EncryptionContext:context-key` para limitar el permiso a las trazas de una cuenta y región en particular.

Antes de utilizar esta política de claves, reemplace el ID de la cuenta de ejemplo, la región y el nombre de traza por valores válidos de su cuenta.

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

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. AWS KMS es compatible con ABAC, ya que le permite controlar el acceso a las claves gestionadas por sus clientes en función de las etiquetas y los alias asociados a las claves del KMS. Las claves de condición de etiqueta y alias que permiten a ABAC AWS KMS ofrecer una forma eficaz y flexible de autorizar a los directores a utilizar las claves de KMS sin necesidad de modificar las políticas ni gestionar las concesiones. Pero debe usar esta característica con cuidado para que a las principales entidades no se les permita o se les deniegue el acceso inadvertidamente.

Si utiliza ABAC, tenga en cuenta que el permiso para administrar etiquetas y alias es ahora un permiso de control de acceso. Asegúrese de conocer las etiquetas y alias existentes en todas las claves KMS antes de implementar una política que dependa de etiquetas o alias. Tome las precauciones razonables al agregar, eliminar y actualizar alias, y al etiquetar y desetiquetar claves. Otorgue permisos para administrar etiquetas y alias solo a las principales entidades que los necesiten y limite las etiquetas y alias que puedan administrar.

Notas

Cuando utilices ABAC para AWS KMS, ten cuidado a la hora de dar permiso a los directores para gestionar etiquetas y alias. Cambiar una etiqueta o alias podría permitir o denegar el permiso a una clave KMS. Los administradores de claves que no tienen permiso para cambiar políticas de claves o crear concesiones pueden controlar el acceso a claves KMS si tienen permiso para administrar etiquetas o alias.

Puede que transcurran cinco minutos hasta que los cambios de etiqueta y alias afecten a la autorización de clave KMS. Los cambios recientes pueden ser visibles en las operaciones de API antes de que afecten a la autorización.

Para controlar el acceso a una clave KMS basándose en su alias, debe utilizar una clave de condición. No puede utilizar un alias para representar una clave KMS en el elemento `Resource` de una declaración de política. Cuando aparece un alias en el elemento `Resource`, la declaración de política se aplica al alias, no a la clave KMS asociada.

Más información

- Para obtener más información sobre la AWS KMS compatibilidad con ABAC, incluidos ejemplos, consulte y. [Uso de alias para controlar el acceso a las claves KMS](#) [Uso de etiquetas para controlar el acceso a las claves KMS](#)
- Para obtener más información general sobre el uso de etiquetas para controlar el acceso a AWS los recursos, consulte [¿Para qué sirve ABAC? AWS](#) y Cómo [controlar el acceso a AWS los recursos mediante etiquetas de recursos](#) en la Guía del usuario de IAM.

Claves de condición ABAC para AWS KMS

Para autorizar el acceso a claves KMS en función de sus etiquetas y alias, utilice las siguientes claves de condición en una política de claves o política de IAM.

Clave de condición de ABAC	Description (Descripción)	Tipo de política	AWS KMS operaciones
leyes: ResourceTag	La etiqueta (clave y valor) en la clave KMS coincide con la etiqueta (clave y valor) o el patrón de etiqueta en la política	Política de IAM únicamente	Operaciones de recursos clave KMS ²
aws:RequestTag/tag-key	La etiqueta (clave y valor) en la solicitud coincide con la etiqueta (clave y valor) o el patrón de etiqueta en la política	Políticas de claves y políticas de IAM ¹	TagResource , UntagResource
leyes: TagKeys	Las claves de etiqueta de la solicitud coinciden con las claves de etiqueta de la política.	Políticas de claves y políticas de IAM ¹	TagResource , UntagResource

Clave de condición de ABAC	Description (Descripción)	Tipo de política	AWS KMS operaciones
kms: ResourceAliases	Los alias asociados a la clave KMS coinciden con los alias o patrones de alias de la política	Política de IAM únicamente	Operaciones de recursos clave KMS ²
kms: RequestAlias	El alias que representa a la clave KMS en la solicitud coincide con los patrones de alias o alias de la política.	Políticas de claves y políticas de IAM ¹	Operaciones criptográficas , DescribeKey , GetPublicKey

¹Cualquier clave de condición que se pueda utilizar en una política de clave también se puede utilizar en una política de IAM, pero solo si [la política de claves lo permite](#).

²Una operación de recursos de clave KMS es una operación autorizada para una clave KMS en particular. Para identificar las operaciones de recursos clave KMS, en la [tabla de permisos AWS KMS](#), busque un valor de la clave KMS en la columna Resources para la operación.

Por ejemplo, puede utilizar estas claves de condición para crear las siguientes políticas.

- Una política de IAM con `kms:ResourceAliases` que habilita el permiso para usar claves KMS con un alias o patrón de alias en particular. Esto es un poco diferente de las políticas que se basan en etiquetas: aunque puede utilizar patrones de alias en una política, cada alias debe ser único en una región Cuenta de AWS y. Esto le permite aplicar una política a un conjunto selecto de claves de KMS sin incluir la clave ARNs de las claves de KMS en la declaración de política. Para agregar o quitar claves KMS del conjunto, cambie el alias de la clave KMS.
- Una política de claves con `kms:RequestAlias` que permite a las principales entidades usar una clave KMS en una operación Encrypt, pero solo cuando la solicitud Encrypt utiliza ese alias para identificar la clave KMS.
- Una política de IAM con `aws:ResourceTag/tag-key` que deniega permiso para utilizar claves KMS con una clave de etiqueta y un valor de etiqueta en concreto. Esto le permite aplicar una política a un conjunto selecto de claves de KMS sin incluir la clave ARNs de las claves de KMS en

la declaración de política. Para agregar o quitar claves KMS del conjunto, etiqueta o desmarca de la clave KMS.

- Una política de IAM con `aws:RequestTag/tag-key` que permite a las principales entidades eliminar solo etiquetas de claves KMS de `"Purpose"="Test"`
- Una política de IAM con `aws:TagKeys` que deniega el permiso para etiquetar o desetiquetar una clave KMS con una clave de etiqueta `Restricted`.

ABAC hace que la administración de accesos sea flexible y escalable. Por ejemplo, puede utilizar la clave de condición `aws:ResourceTag/tag-key` para crear una política de IAM que permita a las principales entidades utilizar una clave KMS para operaciones especificadas solo cuando la clave KMS tenga una etiqueta `Purpose=Test`. La política se aplica a todas las claves KMS de todas las regiones de la Cuenta de AWS.

Cuando se adjunta a un usuario o rol, la siguiente política de IAM permite a las principales entidades utilizar todas las claves KMS existentes con una etiqueta `Purpose=Test` para las operaciones especificadas. Para proporcionar este acceso a claves KMS nuevas o existentes, no es necesario cambiar la política. Solo tiene que adjuntar la etiqueta `Purpose=Test` a las claves KMS. Del mismo modo, para eliminar este acceso de las claves KMS con una etiqueta `Purpose=Test`, edite o elimine la 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"
        }
      }
    }
  ]
}
```

```
}  
}  
]  
}
```

Sin embargo, si utiliza esta función, tenga cuidado al administrar etiquetas y alias. Agregar, cambiar o eliminar una etiqueta o alias puede permitir o denegar inadvertidamente el acceso a una clave KMS. Los administradores de claves que no tienen permiso para cambiar políticas de claves o crear concesiones pueden controlar el acceso a claves KMS si tienen permiso para administrar etiquetas y alias. Para mitigar este riesgo, considere [limitar permisos para administrar etiquetas](#) y [alias](#). Por ejemplo, es posible que exija solo las principales entidades administren etiquetas Purpose=Test. Para más detalles, consulte [Uso de alias para controlar el acceso a las claves KMS](#) y [Uso de etiquetas para controlar el acceso a las claves KMS](#).

¿Etiquetas o alias?

AWS KMS admite ABAC con etiquetas y alias. Ambas opciones proporcionan una estrategia de control de acceso flexible y escalable, pero son ligeramente diferentes entre sí.

Puede decidir usar etiquetas o alias en función de sus patrones de uso particulares AWS . Por ejemplo, si ya ha otorgado permisos de etiquetado a la mayoría de los administradores, podría ser más fácil controlar una estrategia de autorización basada en alias. O bien, si está cerca de la cuota de [alias por clave KMS](#), es posible que prefiera una estrategia de autorización basada en etiquetas.

Los siguientes beneficios son de interés general.

Beneficios del control de acceso basado en etiquetas

- El mismo mecanismo de autorización para distintos tipos de AWS recursos.

Puede utilizar la misma etiqueta o clave de etiqueta para controlar el acceso a varios tipos de recursos, como un clúster de Amazon Relational Database Service (Amazon RDS), un volumen de Amazon Elastic Block Store (Amazon EBS) y una clave KMS. Esta función permite varios modelos de autorización diferentes que son más flexibles que el control de acceso basado en roles tradicional.

- Autorizar el acceso a un grupo de claves KMS.

Puede utilizar etiquetas para administrar el acceso a un grupo de claves KMS en la misma región y Cuenta de AWS . Asigne la misma etiqueta o clave de etiqueta a las claves KMS que elija. A

continuación, cree una declaración easy-to-maintain de política sencilla que se base en la etiqueta o la clave de la etiqueta. Para agregar o quitar una clave KMS de su grupo de autorización, agregue o elimine la etiqueta; no necesita editar la política.

Beneficios del control de acceso basado en alias

- Autorizar el acceso a operaciones criptográficas basadas en alias.

La mayoría de las condiciones políticas de atributos basadas en solicitudes, incluida [aws:RequestTag/tag-key](#), solo afectan a las operaciones que agregan, editan o eliminan el atributo. Sin embargo, la clave de RequestAlias condición [kms:](#) controla el acceso a las operaciones criptográficas en función del alias utilizado para identificar la clave KMS de la solicitud. Por ejemplo, puede concederle permiso a una entidad principal para usar una clave KMS en una operación Encrypt pero solo cuando el valor del parámetro KeyId sea `alias/restricted-key-1`. Para satisfacer esta condición se requiere todo lo siguiente:

- La clave KMS debe estar asociada a ese alias.
- La solicitud debe utilizar el alias para identificar la clave KMS.
- La entidad principal debe tener permiso para utilizar la clave KMS en función de la condición `kms:RequestAlias`.

Esto resulta especialmente útil si las aplicaciones suelen utilizar nombres de alias o alias ARNs para hacer referencia a las claves de KMS.

- Proporcione permisos muy limitados.

Un alias debe ser único en una región Cuenta de AWS y. Como resultado, dar acceso a las principales entidades a una clave KMS basada en un alias puede ser mucho más restrictivo que darles acceso basado en una etiqueta. A diferencia de los alias, las etiquetas se pueden asignar a varias claves KMS de la misma cuenta y región. Si lo desea, puede usar un patrón de alias, como `alias/test*`, para dar acceso a las entidades principales a un grupo de claves KMS en la misma cuenta y Región. Sin embargo, permitir o denegar acceso a un alias en concreto permite un control muy estricto de claves KMS.

Solución de problemas de ABAC para AWS KMS

Controlar el acceso a las claves KMS en función de sus etiquetas y alias es conveniente y potente. Sin embargo, es propenso a algunos errores predecibles que querrá evitar.

Acceso cambiado debido al cambio de etiqueta

Si se elimina una etiqueta o se cambia su valor, se denegará el acceso a la clave KMS a las principales entidades que tengan acceso a una clave KMS basada únicamente en esa etiqueta. Esto también puede ocurrir cuando una etiqueta incluida en una declaración de política de denegación se agrega a una clave KMS. Agregar una etiqueta relacionada con la política a una clave KMS puede permitir el acceso a entidades principales a las que se debe denegar el acceso a una clave KMS.

Por ejemplo, supongamos que una entidad principal tiene acceso a una clave KMS basada en la etiqueta `Project=Alpha`, como el permiso proporcionado por la siguiente declaración de política de IAM de ejemplo.

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"
        }
      }
    }
  ]
}
```

Si se elimina la etiqueta de esa clave KMS o se cambia el valor de la etiqueta, la entidad principal ya no tiene permiso para usar la clave KMS para las operaciones especificadas. Esto puede resultar evidente cuando el director intenta leer o escribir datos en un AWS servicio que utiliza una clave gestionada por el cliente. Para rastrear el cambio de etiqueta, revise CloudTrail los registros [TagResource](#) o [UntagResource](#) las entradas.

Para restaurar el acceso sin actualizar la política, cambie las etiquetas de la clave KMS. Esta acción tiene un impacto mínimo aparte de un breve período de tiempo mientras está surtiendo efecto a lo largo de AWS KMS. Para evitar un error como este, otorgue permisos de etiquetado y desetiquetado solo a las principales entidades que lo necesiten y [limite sus permisos de etiquetado](#) a las etiquetas que necesitan administrar. Antes de cambiar una etiqueta, busque políticas para detectar el acceso que depende de la etiqueta y obtenga claves KMS en todas las regiones que tengan la etiqueta. Podrías considerar la posibilidad de crear una CloudWatch alarma de Amazon cuando cambies determinadas etiquetas.

Cambio de acceso debido al cambio de alias

Si se elimina un alias o se asocia a una clave KMS diferente, se denegará el acceso a la clave KMS a las principales entidades que tengan acceso a la clave KMS basándose únicamente en ese alias. Esto también puede ocurrir cuando se incluye un alias asociado con una clave KMS en una declaración de política de denegación. Agregar un alias relacionado con la política a una clave KMS también puede permitir el acceso a entidades principales a las que se debe denegar el acceso a una clave KMS.

Por ejemplo, la siguiente declaración de política de IAM utiliza la clave de ResourceAliases condición [kms:](#) para permitir el acceso a las claves de KMS en distintas regiones de la cuenta con cualquiera de los alias 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 el cambio de alias, revise CloudTrail los registros y [DeleteAlias](#) las [CreateAliasUpdateAlias](#) entradas.

Para restaurar el acceso sin actualizar la política, cambie el alias asociado con la clave KMS. Dado que cada alias se puede asociar a una sola clave KMS en una cuenta y región, administrar alias es un poco más difícil que administrar etiquetas. Restaurar el acceso a algunas entidades principales en una clave KMS puede denegar el acceso a la misma u otras entidades principales a una clave KMS diferente.

Para evitar este error, otorgue permisos de administración de alias solo a las principales entidades que lo necesiten y [limite sus permisos de administración de alias](#) a los alias que necesitan administrar. Antes de actualizar o eliminar un alias, busque políticas para detectar el acceso que depende del alias y busque claves KMS en todas las regiones asociadas al alias.

Acceso denegado debido a la cuota de alias

Los usuarios que estén autorizados a usar una clave de KMS con una ResourceAliases condición de [kms](#): recibirán una AccessDenied excepción si la clave de KMS supera los [alias predeterminados por cuota de claves de KMS](#) para esa cuenta y región.

Para restaurar el acceso, elimine los alias asociados a la clave KMS para que cumpla con la cuota. O utilice un mecanismo alternativo para dar a los usuarios acceso a la clave KMS.

Cambio de autorización retrasado

Los cambios que realice en las etiquetas y alias pueden tardar hasta cinco minutos en afectar a la autorización de claves KMS. Como resultado, un cambio de etiqueta o alias podría reflejarse en las respuestas de las operaciones de API antes de que afecten a la autorización. Es probable que este retraso sea superior al breve retraso de coherencia que, eventualmente, afecta a la mayoría de AWS KMS las operaciones.

Por ejemplo, puede que tenga una política de IAM que permita a ciertas entidades principales utilizar cualquier clave KMS con una etiqueta "Purpose"="Test". Luego, agrega la etiqueta "Purpose"="Test" de una clave KMS. Aunque la [TagResource](#) operación se complete y la [ListResourceTags](#) respuesta confirme que la etiqueta está asignada a la clave de KMS, es posible que las entidades principales no tengan acceso a la clave de KMS durante un máximo de cinco minutos.

Para evitar errores, construya este retraso esperado en su código.

Solicitudes fallidas debido a actualizaciones de alias

Cuando se actualiza un alias, se asocia un alias existente con una clave KMS diferente.

El [descifrado](#) y [ReEncrypt](#) las solicitudes que especifican el [nombre del alias o el ARN](#) del alias pueden fallar porque el alias ahora está asociado a una clave KMS que no cifró el texto cifrado. Esta situación normalmente devuelve un `IncorrectKeyException` o `NotFoundException`. O si la solicitud no tiene parámetro `KeyId` o `DestinationKeyId`, la operación puede fallar con la excepción `AccessDenied` dado que la persona que llama ya no tiene acceso a la clave KMS que cifró el texto cifrado.

Puede rastrear el cambio consultando los CloudTrail registros y las entradas de [CreateAlias](#) registro. [UpdateAliasDeleteAlias](#) También puede usar el valor del `LastUpdatedDate` campo en la [ListAliases](#) respuesta para detectar un cambio.

Por ejemplo, en el siguiente [ListAliases](#) ejemplo de respuesta se muestra que se actualizó el `ProjectAlpha_Test` alias de la `kms:ResourceAliases` condición. Como resultado, las principales entidades que tienen acceso basado en el alias pierden el acceso a la clave KMS previamente asociada. En su lugar, tienen acceso a la clave KMS recién asociada.

```
$ 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
  }
]
```

La solución para este cambio no es simple. Puede actualizar el alias nuevamente para asociarlo a la clave KMS original. Sin embargo, antes de actuar, debe considerar el efecto de ese cambio en la clave KMS asociada actualmente. Si las principales entidades utilizan la última clave KMS en operaciones criptográficas, es posible que necesiten acceso continuo a ella. En este caso, es posible que desee actualizar la política para asegurarse de que las principales entidades tienen permiso para usar ambas claves KMS.

Puede evitar un error como el siguiente: antes de actualizar un alias, busque políticas para detectar el acceso que depende del alias. A continuación, obtenga claves KMS en todas las regiones asociadas con el alias. De permisos de administración de alias solo a las principales entidades que lo necesiten y [limite sus permisos de administración de alias](#) a los alias que necesitan administrar.

RBAC para AWS KMS

El control de acceso basado en roles (RBAC) es una estrategia de autorización que solo proporciona a los usuarios los permisos necesarios para realizar sus tareas laborales, y nada más. AWS KMS es compatible con el RBAC, ya que le permite controlar el acceso a sus claves especificando permisos detallados sobre el uso de las claves dentro de las [políticas de claves](#). Las políticas de claves especifican un recurso, una acción, un efecto, una entidad principal y unas condiciones opcionales para conceder el acceso a las claves.

Para implementar el RBAC en AWS KMS, recomendamos separar los permisos de los usuarios clave de los administradores clave.

Key users

El siguiente ejemplo de política de claves permite al rol de IAM de `ExampleUserRole` utilizar la clave 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": "*"
  }
```

Es posible que sus usuarios de claves necesiten menos permisos que el usuario de este ejemplo. Asigne solo los permisos que necesite el usuario. Utilice las siguientes preguntas para ajustar con más precisión los permisos.

- ¿Qué entidades principales de IAM (roles o usuarios) necesitan acceder a la clave?
- ¿Qué acciones debe realizar cada entidad principal con la clave? Por ejemplo, ¿la entidad principal solo necesita permisos de cifrado y firma?
- ¿El usuario es una persona o un servicio? AWS Si se trata de un AWS servicio, puede usar la [clave de condición](#) para restringir el uso de la clave a un AWS servicio específico.

Key administrators

El siguiente ejemplo de política de claves permite al rol de IAM de ExampleAdminRole administrar la clave 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": "*"
}
```

Es posible que los administradores de claves necesiten menos permisos que el administrador de este ejemplo. Asigne únicamente los permisos que necesiten sus administradores de claves.

Otorgue a los usuarios únicamente los permisos que necesitan para cumplir con sus roles. Los permisos de un usuario pueden variar en función de si la clave se utiliza en entornos de prueba o de producción. Si utiliza permisos menos restrictivos en determinados entornos que no son de producción, implemente un proceso para probar las políticas antes de lanzarlas a producción.

Más información

- [Identidades de IAM \(usuarios, grupos de usuarios y roles\)](#)
- [Tipos de control de acceso](#)

Permitir a los usuarios de otras cuentas utilizar una clave KMS

Puede permitir que los usuarios o roles de una cuenta diferente Cuenta de AWS usen una clave de KMS en su cuenta. El acceso entre cuentas requiere permiso en la política de claves de la clave KMS y en una política de IAM en la cuenta del usuario externo.

El permiso entre cuentas solo es efectivo para las siguientes operaciones:

- [Operaciones criptográficas](#)
- [CreateGrant](#)
- [DescribeKey](#)

- [GetKeyRotationStatus](#)
- [GetPublicKey](#)
- [ListGrants](#)
- [RetireGrant](#)
- [RevokeGrant](#)

Si concede a un usuario de otra cuenta permiso para otras operaciones, esos permisos no surten efecto. Por ejemplo, si concedes a un responsable de otra cuenta el ListKeys permiso [kms:](#) en una política de IAM, o el ScheduleKeyDeletion permiso [kms:](#) en una política de IAM, en una clave de KMS, los intentos del usuario de realizar esas operaciones con tus recursos seguirán fallando.

Para obtener más información sobre el uso de claves de KMS en distintas cuentas para AWS KMS realizar operaciones, consulta la columna Uso multicuenta en los [AWS KMS permisos](#) campos y. [Uso de claves KMS en otras cuentas](#) También hay una sección Cross-account use (Uso entre cuentas) en cada descripción de la API en la [Referencia de la API de AWS Key Management Service](#).

 Warning

Tenga cuidado al conceder permisos a las principales entidades para usar las claves KMS. Siempre que sea posible, siga el principio del mínimo privilegio. Proporcione a los usuarios acceso solo a las claves KMS que necesitan para las operaciones que requieren. Además, tenga cuidado con el uso de cualquier clave KMS desconocida, especialmente una clave KMS en una cuenta diferente. Es posible que los usuarios malintencionados le den permisos para usar su clave KMS para obtener información sobre usted o su cuenta. Para obtener más información sobre el uso de políticas de para proteger los recursos de su cuenta, consulte [Prácticas recomendadas para las políticas de IAM](#).

Para conceder a permiso para utilizar una clave KMS a los usuarios y roles de otra cuenta, debe utilizar dos tipos diferentes de políticas:

- La política de claves de la clave KMS debe conceder a la cuenta externa (o a los usuarios y roles de la cuenta externa) permiso para utilizar la clave KMS. La política de claves está en la cuenta propietaria de la clave KMS.
- Las políticas de IAM de la cuenta externa debe delegar los permisos de política de claves de a sus usuarios y roles. Estas políticas se establecen en la cuenta externa y otorgan permisos para los usuarios y roles de esa cuenta.

La política de claves determina quién puede tener acceso a la clave KMS. La política de IAM determina quién sí tiene acceso a la clave KMS. Ni la política de claves ni la política de IAM solas son suficientes, debe cambiar ambas.

Para editar la política clave, puede usar la [vista de políticas](#) en las [PutKeyPolicy](#) operaciones Consola de administración de AWS [CreateKey](#) o usar.

Para obtener ayuda con la edición de políticas de IAM, consulte [Uso de políticas de IAM con AWS KMS](#).

Para ver un ejemplo que muestra cómo la política de claves y las políticas de IAM se combinan para permitir el uso de una clave KMS en una cuenta diferente, consulte [Ejemplo 2: El usuario asume el rol con permiso para usar una clave de KMS en otro Cuenta de AWS](#).

Puede ver las AWS KMS operaciones entre cuentas resultantes en la clave KMS de sus [AWS CloudTrail registros](#). Las operaciones que utilizan claves KMS en otras cuentas se registran tanto en la cuenta del autor de la llamada como en la cuenta del propietario de la clave KMS.

Temas

- [Paso 1: Agregar una declaración de política de claves en la cuenta local](#)
- [Paso 2: Agregar políticas de IAM a la cuenta externa](#)
- [Permitir el uso de claves KMS externas con Servicios de AWS](#)
- [Uso de claves KMS en otras cuentas](#)

Note

En los ejemplos de este tema, se muestra cómo utilizar juntas una política de claves y una política de IAM para proporcionar y limitar el acceso a una clave KMS. Estos ejemplos genéricos no pretenden representar los permisos que una clave de KMS Servicio de AWS requiere en concreto. Para obtener información sobre los permisos que se Servicio de AWS requieren, consulte el tema del cifrado en la documentación del servicio.

Paso 1: Agregar una declaración de política de claves en la cuenta local

La política de claves de una clave KMS es el principal determinante de quién puede obtener acceso a la clave KMS y qué operaciones puede realizar. La política de claves siempre se define en la

cuenta propietaria de la clave KMS. A diferencia de las políticas de IAM, las políticas de claves no especifican ningún recurso. El recurso es la clave KMS asociada a la política de claves. Al proporcionar permiso entre cuentas, la política de claves de la clave KMS debe conceder a la cuenta externa (o a los usuarios y roles de la cuenta externa) permiso para utilizar la clave KMS.

Para conceder a una cuenta externa permiso para utilizar la clave KMS, agregue una declaración a la política de claves que especifique la cuenta externa. En el elemento `Principal` de la política de claves, escriba el nombre de recurso de Amazon (ARN) de la cuenta externa.

Al especificar una cuenta externa en una política de claves, los administradores de IAM de la cuenta externa pueden utilizar políticas de IAM para delegar esos permisos a cualquier usuario y rol de la cuenta externa. También pueden decidir qué acciones especificadas en la política de claves pueden realizar los usuarios y roles.

Los permisos otorgados a la cuenta externa y sus entidades principales solo son efectivos si la cuenta externa está habilitada en la región que aloja la clave KMS y su política de clave. Para obtener información acerca de las regiones que no están habilitadas de forma predeterminada ("Regiones de adhesión"), consulte [Administración de Regiones de AWS](#) en la Referencia general de AWS.

Por ejemplo, suponga que desea permitir que la cuenta 444455556666 utilice una clave KMS de cifrado simétrica en la cuenta 111122223333. Para ello, agregue una declaración de política como la del siguiente ejemplo a la política de claves de la clave KMS de la cuenta 111122223333. Esta declaración de política concede a la cuenta externa, 444455556666, permiso para utilizar la clave KMS en operaciones criptográficas para claves KMS de cifrado simétricas.

Note

El siguiente ejemplo representa un ejemplo de política de claves para compartir una clave KMS con otra cuenta. Sustituya los valores de `Sid`, `Principal` y `Action` de ejemplo por valores válidos para el uso previsto de su clave 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": "*" ]  
}
```

En lugar de conceder permiso a la cuenta externa, puede especificar usuarios y roles externos concretos en la política de claves. Sin embargo, esos usuarios y roles no pueden utilizar la clave KMS hasta que los administradores de IAM de la cuenta externa asocien las políticas de IAM adecuadas a sus identidades. Las políticas de IAM pueden conceder permiso a todos o a un subconjunto de los usuarios y roles externos que se especifican en la política de claves. Y pueden permitir todas o un subconjunto de las acciones especificadas en la política de claves.

Especificar identidades en una política de claves restringe los permisos que los administradores de IAM de la cuenta externa pueden proporcionar. Sin embargo, hace que la administración de políticas con dos cuentas sea más compleja. Por ejemplo, suponga que necesita agregar un usuario o rol. Debe agregar dicha identidad a la política de claves en la cuenta propietaria de la clave KMS y crear políticas de IAM en la cuenta de la identidad.

Para especificar usuarios o roles externos concretos en una política de claves, en el elemento `Principal`, escriba el nombre de recurso de Amazon (ARN) de un usuario o rol en la cuenta externa.

Por ejemplo, la siguiente declaración de política de claves de ejemplo permite a `ExampleRole` de la cuenta 444455556666 utilizar una clave KMS en la cuenta 111122223333. Esta declaración de política de claves concede a la cuenta externa, 444455556666, permiso para utilizar la clave KMS en operaciones criptográficas para claves KMS de cifrado simétricas.

Note

El siguiente ejemplo representa un ejemplo de política de claves para compartir una clave KMS con otra cuenta. Sustituya los valores de `Sid`, `Principal` y `Action` de ejemplo por valores válidos para el uso previsto de su clave 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

No establezca la Entidad principal en un asterisco (*) en ninguna declaración de política de claves que permita permisos a menos que utilice [condiciones](#) para limitar la política de claves. Un asterisco indica todas las identidades de cada uno de los Cuentas de AWS permisos para usar la clave KMS, a menos que otra declaración de política lo deniegue explícitamente. Los usuarios de otros usuarios Cuentas de AWS pueden usar su clave de KMS siempre que tengan los permisos correspondientes en su propia cuenta.

También debe decidir qué permisos desea conceder a la cuenta externa. Por ejemplo, es posible que desee conceder a los usuarios permiso para descifrar pero no cifrar, o permiso para ver la clave KMS pero no utilizarla. Para obtener una lista de permisos sobre las claves KMS, consulte [AWS KMS permisos](#).

Definición de la política de claves predeterminada al crear una clave KMS

Cuando usas la [CreateKey](#) operación para crear una clave de KMS, puedes usar su Policy parámetro para especificar una política de claves que conceda permiso a una cuenta externa, o a usuarios y roles externos, para usar la clave de KMS.

Al crear una clave de KMS en Consola de administración de AWS, también se crea su política de claves. Al seleccionar identidades en las secciones Key Administrators (Administradores de claves) y Key Users (Usuarios de claves), AWS KMS agrega declaraciones de política para esas

identidades a la política de claves de la clave KMS. La sección Key Users (Usuarios de claves) también le permite agregar cuentas externas como usuarios de claves.

Al introducir el identificador de cuenta de una cuenta externa, se AWS KMS añaden dos declaraciones a la política clave. Esta acción solo afecta a la política de claves. Los usuarios y roles de la cuenta externa no pueden utilizar la clave KMS hasta que se asocien políticas de IAM para concederles algunos o todos estos permisos.

La primera declaración de la política de claves concede a la cuenta externa permiso para utilizar la clave KMS en operaciones criptográficas. La segunda declaración de política clave permite a la cuenta externa crear, ver y revocar las concesiones en la clave de KMS, pero solo cuando la solicitud proviene de un [AWS servicio que esté integrado con AWS KMS ella](#). Estos permisos permiten que otros AWS servicios que cifran los datos de los usuarios utilicen la clave KMS. Estos permisos están diseñados para las claves KMS que cifran los datos de usuario en los servicios de AWS

Paso 2: Agregar políticas de IAM a la cuenta externa

La política de claves de la cuenta propietaria de la clave KMS establece el rango válido de permisos. Sin embargo, los usuarios y roles de la cuenta externa no pueden utilizar la clave KMS hasta que adjunte políticas de IAM que deleguen esos permisos o utilice concesiones para administrar el acceso a la clave KMS. Las políticas de IAM se establecen en la cuenta externa.

Si la política de claves concede permiso a la cuenta externa, puede asociar políticas de IAM a cualquier usuario o rol de la cuenta. Sin embargo, si la política de claves concede permiso a usuarios o roles especificados, la política de IAM solo puede conceder esos permisos a todos o a un subconjunto de los usuarios y roles especificados. Si una política de IAM concede a la clave KMS acceso a otros usuarios o roles externos, no tiene ningún efecto.

La política de claves también limita las acciones de la política de IAM. La política de IAM puede delegar todas o un subconjunto de las acciones especificadas en la política de claves. Si la política de IAM enumera acciones que no se especifican en la política de claves, esos permisos no son efectivos.

La siguiente política de IAM de ejemplo permite a la entidad principal utilizar la clave KMS en la cuenta de 111122223333 para operaciones criptográficas. Para conceder este permiso a los usuarios y roles de la cuenta 444455556666, [asocie la política](#) a los usuarios o roles de la cuenta 444455556666.

 Note

El siguiente ejemplo representa un ejemplo de política de IAM para compartir una clave KMS con otra cuenta. Sustituya los valores de `Sid`, `Resource` y `Action` de ejemplo por valores válidos para el uso previsto de su clave 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"
}
```

Tenga en cuenta los siguientes detalles sobre esta política:

- A diferencia de las políticas de claves, las declaraciones de política de IAM no contienen el elemento `Principal`. En las políticas de IAM, la entidad principal es la identidad a la que está asociada la política.
- El elemento de `Resource` de la política de IAM identifica la clave KMS que la entidad principal puede utilizar. Para especificar una clave KMS, agregue el [ARN de la clave](#) al elemento `Resource`.
- Puede especificar más de una clave KMS en el elemento `Resource`. Sin embargo, si no especifica determinadas claves KMS en el elemento `Resource`, es posible que conceda acceso de forma inadvertida a más clave KMS de las que pretendía.
- Para permitir al usuario externo utilizar la clave KMS con [servicios de AWS que se integran con AWS KMS](#), es posible que tenga que agregar permisos a la política de claves o a la política de IAM. Para obtener más información, consulte [Permitir el uso de claves KMS externas con Servicios de AWS](#).

Para obtener más información sobre el uso de las políticas de IAM, consulte [Políticas de IAM](#).

Permitir el uso de claves KMS externas con Servicios de AWS

Puede conceder a un usuario de otra cuenta permiso para utilizar su clave KMS con un servicio integrado con AWS KMS. Por ejemplo, un usuario de una cuenta externa puede utilizar su clave KMS para cifrar los objetos en un bucket de Amazon S3 o para cifrar los secretos que almacenan en AWS Secrets Manager.

La política de claves debe conceder al usuario externo o a la cuenta del usuario externo permiso para utilizar la clave KMS. Además, debe asociar políticas de IAM a la identidad que concede al usuario permiso para utilizar el Servicio de AWS. Además, el servicio podría requerir que los usuarios tengan permisos adicionales en la política de claves o política de IAM. Para obtener una lista de los permisos que se Servicio de AWS requieren para una clave administrada por el cliente, consulte el tema sobre protección de datos en el capítulo sobre seguridad de la guía del usuario o la guía para desarrolladores del servicio.

Uso de claves KMS en otras cuentas

Si tiene permiso para usar una clave de KMS en otra Cuenta de AWS, puede usar la clave de KMS en Consola de administración de AWS, AWS SDKs, AWS CLI, y Herramientas de AWS para PowerShell.

Para identificar una clave KMS en una cuenta diferente en un comando de shell o solicitud de la API, utilice los siguientes [identificadores clave](#).

- Para [las operaciones criptográficas DescribeKey](#), y [GetPublicKey](#), utilice la [clave ARN](#) o el [alias ARN](#) de la clave KMS.
- Para [CreateGrant](#), [GetKeyRotationStatusListGrants](#), y [RevokeGrant](#), utilice la clave ARN de la clave KMS.

Si solo introduce un identificador de clave o un nombre de alias, se AWS supone que la clave de KMS está en su cuenta.

La AWS KMS consola no muestra las claves de KMS en otras cuentas, incluso si tiene permiso para usarlas. Además, las listas de claves KMS mostradas en las consolas de otros servicios de AWS no incluyen claves KMS en otras cuentas.

Para especificar una clave de KMS en una cuenta diferente de la consola de un AWS servicio, debe introducir la clave ARN o el alias ARN de la clave de KMS. El identificador de clave requerido varía

según el servicio y podría diferir entre la consola de servicio y sus operaciones de API. Para conocer detalles, consulte la documentación del servicio.

Control del acceso a claves de varias regiones

Puede usar claves de varias regiones en escenarios de conformidad, recuperación de desastres y copia de seguridad que serían más complejos con claves de una sola región. Sin embargo, dado que las propiedades de seguridad de las claves de varias regiones son significativamente diferentes de las de las claves de una sola región, recomendamos tener precaución al autorizar la creación, la administración y el uso de claves de varias regiones.

Note

Las declaraciones de la política de IAM existentes con caracteres comodín en el campo `Resource` ahora se aplican a las claves de una sola región y de varias regiones. Para restringirlas a claves KMS de una sola región o claves de varias regiones, utilice la clave de MultiRegion condición [kms:](#).

Utilice las herramientas de autorización para evitar la creación y el uso de claves de varias regiones en cualquier escenario en el que una sola región sea suficiente. Permita que los directores repliquen una clave multirregional solo en Regiones de AWS aquellos lugares donde las necesiten. De permiso para las claves de varias regiones solo a las entidades principales que las necesiten y solo para las tareas que las requieran.

Puedes usar políticas clave, políticas de IAM y subvenciones para permitir que los directores de IAM administren y usen tus claves multirregionales. Cuenta de AWS Cada clave de varias regiones es un recurso independiente con un ARN clave única y una política clave. Debe establecer y mantener una política clave para cada clave y asegurarse de que las políticas de IAM nuevas y existentes implementen su estrategia de autorización.

Para admitir claves multirregionales, AWS KMS utiliza un rol vinculado al servicio de IAM. Este rol le da a AWS KMS los permisos que necesita para sincronizar [propiedades compartidas](#). Para obtener más información, consulte [Autoriza la sincronización de claves AWS KMS multirregionales](#).

Temas

- [Conceptos básicos de autorización para claves de varias regiones](#)
- [Autorización de administradores y usuarios clave de varias regiones](#)

Conceptos básicos de autorización para claves de varias regiones

Al diseñar políticas de claves y políticas de IAM para claves de varias regiones, tenga en cuenta los siguientes principios.

- **Política de claves:** cada clave de varias regiones es un recurso clave KMS independiente con su propia [política de claves](#). Puede aplicar la misma política de clave o una política de clave diferente para cada clave del conjunto de claves de varias regiones relacionadas. Las políticas clave no son [propiedades compartidas de las](#) claves multirregionales. AWS KMS no copia ni sincroniza las políticas clave entre las claves multirregionales relacionadas.

Al crear una clave de réplica en la AWS KMS consola, la consola muestra la política de claves actual de la clave principal para mayor comodidad. Puede utilizar esta política de claves, editarla o eliminarla y reemplazarla. Pero incluso si acepta la política de clave principal sin cambios, AWS KMS no sincroniza las políticas. Por ejemplo, si cambia la política de clave de la clave principal, la política de clave de la clave de réplica sigue siendo la misma.

- **Política de claves predeterminada:** al crear claves multirregionales mediante las `ReplicateKey` operaciones [CreateKey](#), se aplica la [política de claves predeterminada](#), a menos que especifique una política de claves en la solicitud. Esta es la misma política de clave predeterminada que se aplica a las claves de una sola región.
- **Políticas de IAM:** al igual que con todas las claves KMS, puede usar políticas de IAM para controlar el acceso a las claves de varias regiones solo cuando la [política clave lo permite](#). [Las políticas de IAM](#) se aplican a todos de forma Regiones de AWS predeterminada. Sin embargo, puede utilizar claves de condición, como [aws: RequestedRegion](#), para limitar los permisos a una región concreta.

Para crear claves primarias y de réplica, las entidades principales deben tener permiso `kms:CreateKey` en una política de IAM que se aplica a la región donde se crea la clave.

- **Subvenciones:** AWS KMS [las subvenciones](#) son regionales. Cada concesión da permisos para una clave KMS. Puede utilizar concesiones para dar permisos a una clave principal de varias regiones o clave de réplica. Sin embargo, no puede utilizar una sola concesión para dar permisos a varias claves KMS, incluso si se trata de claves de varias regiones relacionadas.
- **ARN de clave:** cada clave de varias regiones tiene un [ARN de clave única](#). La clave ARNs de las claves multirregionales relacionadas tiene la misma partición, cuenta e ID de clave, pero diferentes regiones.

Para aplicar una declaración de política de IAM a una clave concreta de varias regiones, utilice su ARN clave o un patrón ARN clave que incluya la región. Para aplicar una declaración de política de

IAM a todas las claves de varias regiones, utilice un comodín (*) en el elemento Región del ARN, como se muestra en el siguiente ejemplo.

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

Para aplicar una declaración de política a todas las claves multirregionales de su país Cuenta de AWS, puede utilizar la condición de MultiRegion política [kms:](#) o un patrón de identificador de clave que incluya el prefijo distintivo `mrk-`.

- Función vinculada al servicio: [los directores que crean claves principales multirregionales deben tener el permiso iam: CreateServiceLinkedRole](#)

[Para sincronizar las propiedades compartidas de las claves multirregionales relacionadas, asume una función vinculada al servicio de IAM.](#) AWS KMS crea el rol vinculado al servicio Cuenta de AWS cada vez que se crea una clave principal multirregional. (Si la función existe, AWS KMS lo recrea, que no tiene ningún efecto nocivo). El rol es válido en todas las regiones. [Para poder crear \(o volver AWS KMS a crear\) el rol vinculado al servicio, los directores que creen claves principales multirregionales deben tener el permiso iam: CreateServiceLinkedRole](#)

Autorización de administradores y usuarios clave de varias regiones

Las entidades principales que crean y administran claves de varias regiones necesitan los siguientes permisos en las regiones principal y de réplica:

- `kms:CreateKey`
- `kms:ReplicateKey`
- `kms:UpdatePrimaryRegion`
- `iam:CreateServiceLinkedRole`

Creación de una clave principal

Para [crear una clave principal multirregional](#), el director necesita `CreateServiceLinkedRole` permisos `kms: CreateKey` e `iam: en una política de IAM` que sea efectiva en la región de la clave principal. Las entidades principales que tienen estos permisos pueden crear claves de una sola región y de varias regiones a menos que restrinja sus permisos.

El `iam:CreateServiceLinkedRole` permiso permite crear el [AWSServiceRoleForKeyManagementServiceMultiRegionKeysrol AWS KMS](#) para sincronizar las [propiedades compartidas de las claves multirregionales relacionadas](#).

Por ejemplo, esta política de IAM permite a una entidad principal crear claves de varias regiones, adjuntar políticas para esas claves y asignar funciones vinculadas al servicio para las claves de varias regiones.

JSON

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

Para permitir o denegar el permiso para crear claves principales multirregionales, utilice la clave de condición [kms: MultiRegion](#). Los valores válidos son `true` (clave de varias regiones) o `false` (clave de una sola región). Por ejemplo, la siguiente declaración de política de IAM utiliza una acción `Deny` con la clave de condición `kms:MultiRegion` para evitar que las entidades principales creen claves de varias regiones.

JSON

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

```

    "Statement": {
      "Action": "kms:CreateKey",
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:MultiRegion": true
        }
      }
    }
  }
}

```

Claves de replicación

Para [crear una clave de réplica de varias regiones](#), la entidad principal necesita los siguientes permisos:

- [kms: ReplicateKey](#) permiso en la política de claves de la clave principal.
- [kms: CreateKey](#) permiso en una política de IAM que está en vigor en la región de claves réplicas.

Tenga cuidado al permitir estos permisos. Permiten a las entidades principales crear claves KMS y las políticas de claves que autorizan su uso. El permiso `kms:ReplicateKey` también autoriza la transferencia de material clave a través de los límites de la región dentro de AWS KMS.

Para restringir los campos Regiones de AWS en los que se puede replicar una clave multirregional, utilice la clave de condición [kms: ReplicaRegion](#). Limita solo el permiso `kms:ReplicateKey`. De lo contrario, no tiene ningún efecto. Por ejemplo, la siguiente política de claves permite a la entidad principal replicar esta clave principal, pero solo en las regiones 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"
    ]
}
}
}

```

Actualización de la región principal

Las entidades principales autorizadas pueden convertir una clave de réplica en una clave principal, lo que cambia la clave principal anterior en una réplica. Esta acción se denomina [actualización de la región principal](#). Para actualizar la región principal, el director necesita el UpdatePrimaryRegion permiso [kms](#): en ambas regiones. Puede proporcionar estos permisos en una política de claves o una política de IAM.

- kms:UpdatePrimaryRegion en la clave principal. Este permiso debe ser efectivo en la región de clave principal.
- kms:UpdatePrimaryRegion en la clave de réplica. Este permiso debe ser efectivo en la región clave de réplica.

Por ejemplo, la siguiente política de clave otorga a los usuarios que pueden asumir el permiso de rol de Administrador para actualizar la región principal de la clave KMS. Esta clave KMS puede ser la clave principal o una clave de réplica en esta operación.

```

{
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:UpdatePrimaryRegion"
}

```

Para restringir Regiones de AWS lo que puede alojar una clave principal, utilice la clave de PrimaryRegion condición [kms](#):. Por ejemplo, la siguiente declaración de política de IAM permite a los directores actualizar la región principal de las claves multirregionales de la región Cuenta de AWS, pero solo cuando la nueva región principal sea una de las regiones 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"
    ]
  }
}
```

Uso y administración de claves de varias regiones

De forma predeterminada, las principales entidades que tienen permiso para usar y administrar claves KMS en una Cuenta de AWS y Región también tienen permiso para usar y administrar claves de varias regiones. Sin embargo, puede utilizar la clave de MultiRegion condición [kms:](#) para permitir solo las claves de una sola región o solo las claves de varias regiones. O bien, utilice la clave de MultiRegionKeyType condición [kms:](#) para permitir solo las claves principales de varias regiones o solo las claves de réplica. Ambas claves de condición controlan el acceso a la [CreateKey](#) operación y a cualquier operación que utilice una clave KMS existente, como [Encrypt](#) o [EnableKey](#).

En el siguiente ejemplo de declaración de política de IAM se utiliza la clave de condición `kms:MultiRegion` para evitar que las entidades principales utilicen o administren cualquier clave de varias regiones.

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

Esta declaración de política de IAM de ejemplo utiliza la condición `kms:MultiRegionKeyType` para permitir que las entidades principales programen y cancelen la eliminación de claves, pero solo en las claves de réplica de varias regiones.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": {
    "arn:aws:kms:us-west-2:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": "kms:MultiRegionKeyType": "REPLICA"
  }
}
```

Determinar el acceso a AWS KMS keys

Para determinar el alcance total de quién o qué tiene acceso actualmente a una clave AWS KMS key, debes examinar la política clave de la clave de KMS, todas las [concesiones](#) que se aplican a la clave de KMS y, posiblemente, todas las políticas AWS Identity and Access Management (de IAM). Puede hacerlo para determinar el ámbito del uso potencial de una clave KMS o como ayuda para cumplir los requisitos de conformidad o auditoría. Los siguientes temas pueden ayudarle a generar una lista completa de las entidades principales de AWS (identidades) que actualmente tienen acceso a una clave KMS.

Temas

- [Examinar la política de claves](#)
- [Examen de las políticas de IAM](#)
- [Examinar concesiones](#)

Examinar la política de claves

Las [políticas de claves](#) son la forma principal de controlar el acceso a las claves KMS. Cada clave KMS tiene exactamente una política de claves.

Cuando una política de claves consta de la [política de claves predeterminada](#) o la incluye, la política de claves permite a los administradores de IAM de la cuenta utilizar políticas de IAM para controlar el acceso a la clave KMS. Además, si la política de claves da permiso a [otra Cuenta de AWS](#) para que utilice la clave KMS, los administradores de IAM de la cuenta externa pueden utilizar políticas de IAM para delegar dichos permisos. Para determinar la lista completa de entidades principales que pueden obtener acceso a la clave KMS, [examine las políticas de IAM](#).

Para ver la política clave de una [clave gestionada por el AWS KMS cliente](#) o [Clave administrada de AWS](#) de tu cuenta, utiliza la [GetKeyPolicy](#) operación Consola de administración de AWS o la de la AWS KMS API. Para ver la política de claves, debe tener permisos `kms:GetKeyPolicy` para la clave KMS. Para obtener declaraciones acerca de cómo ver la política de claves para una clave KMS, consulte [the section called "Visualización de políticas de claves"](#).

Examine el documento de políticas de claves y anote todas las entidades principales especificadas en el elemento `Principal` de cada declaración de política. En una declaración de política que `Allow` surta efecto, los usuarios de IAM, las funciones de IAM y Cuentas de AWS el `Principal` elemento tienen acceso a esta clave de KMS.

Note

No establezca la Entidad principal en un asterisco (*) en ninguna declaración de política de claves que permita permisos a menos que utilice [condiciones](#) para limitar la política de claves. Un asterisco indica todas las identidades de cada Cuenta de AWS permiso para usar la clave de KMS, a menos que otra declaración de política lo deniegue explícitamente. Los usuarios de otras Cuentas de AWS pueden usar su clave de KMS siempre que dispongan de los permisos correspondientes en su propia cuenta.

En los siguientes ejemplos se utilizan las declaraciones de política existentes en la [política de claves predeterminada](#) para demostrar cómo hacerlo.

Example Declaración de política 1

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

```
}
```

En la declaración de política 1, `arn:aws:iam::111122223333:root` es un [principal de AWS cuenta](#) que hace referencia al Cuenta de AWS 111122223333. (No es el usuario raíz de la cuenta). De forma predeterminada, una declaración de política como esta se incluye en el documento de política clave cuando se crea una nueva clave de KMS con la o se crea una nueva clave de KMS mediante programación Consola de administración de AWS, pero no se proporciona una política clave.

Un documento de política clave con una declaración que permite el acceso a las [políticas de IAM de la cuenta Cuenta de AWS permite el acceso a la](#) clave de KMS. Esto significa que los usuarios y las roles de la cuenta podrían tener acceso a la clave KMS aunque no se indiquen explícitamente como entidades principales en el documento de políticas de claves. Asegúrese de [examinar todas las políticas de IAM](#) de todas las que Cuentas de AWS figuran como principales para determinar si permiten el acceso a esta clave de KMS.

Example Declaración 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": "*"
}
```

En la declaración de política 2, `arn:aws:iam::111122223333:role/KMSKeyAdmins` hace referencia a la función de IAM nombrada KMSKeyAdmins en 111122223333. Cuenta de AWS Los

usuarios que están autorizados a asumir este rol pueden realizar las acciones enumeradas en la declaración de política, que son las acciones administrativas para administrar una clave KMS.

Example Declaración 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": "*"
}
```

En la declaración de política 3, `arn:aws:iam::111122223333:role/EncryptionApp` hace referencia a la función de IAM nombrada en 111122223333. EncryptionApp Cuenta de AWS Las entidades principales que están autorizadas a asumir este rol tienen permiso para realizar las acciones enumeradas en la declaración de política, que incluyen las [operaciones criptográficas](#) para una clave KMS de cifrado simétrico.

Example Declaración 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}}
}
```

En la declaración de política 4, `arn:aws:iam::111122223333:role/EncryptionApp` hace referencia a la función de IAM nombrada en 111122223333. EncryptionApp Cuenta de AWS Las

entidades principales que están autorizadas a asumir este rol tienen permiso para realizar las acciones enumeradas en la declaración de política. Estas acciones, cuando se combinan con las acciones permitidas en Ejemplo de declaración de política 3, son las necesarias para delegar el uso de la clave KMS a la mayoría de los servicios de [AWS que se integran con AWS KMS](#), específicamente los servicios que usan [concesiones](#). El `GrantIsForAWSResource` valor `kms:` del `Condition` elemento garantiza que la delegación solo esté permitida cuando el delegado sea un AWS servicio que se integre con las concesiones de autorización AWS KMS y las utilice.

Para conocer las distintas formas en las que puede especificar una entidad principal en un documento de políticas de claves, consulte [Especificación de un elemento principal](#) en la Guía del usuario de IAM.

Para obtener más información sobre las políticas AWS KMS clave, consulte [Políticas clave en AWS KMS](#).

Examen de las políticas de IAM

Además de la política de claves y concesiones, también puede utilizar [políticas de IAM](#) para permitir el acceso a una clave KMS. Para obtener más información sobre cómo funcionan las políticas de IAM y las políticas de claves de forma conjunta, consulte [Solución de problemas de AWS KMS permisos](#).

Para determinar qué entidades principales tienen acceso a una clave KMS a través de las políticas de IAM, puede utilizar la herramienta del [simulador de políticas de IAM](#) o puede realizar solicitudes a la API de IAM.

Formas de examinar las políticas de IAM

- [Examen de las políticas de IAM con el simulador de políticas de IAM](#)
- [Examen de políticas de IAM con la API de IAM](#)

Examen de las políticas de IAM con el simulador de políticas de IAM

El simulador de políticas de IAM puede ayudarle a saber qué entidades principales tienen acceso a una clave KMS mediante una política de IAM.

Para utilizar el simulador de políticas de IAM para determinar el acceso a una clave KMS

1. Inicie sesión en el simulador de políticas de IAM Consola de administración de AWS y, a continuación, abra el simulador de políticas de IAM en <https://policysim.aws.amazon.com/>.

2. En el panel Users, Groups, and Roles, elija el usuario, grupo o rol cuyas políticas desee simular.
3. (Opcional) Desactive la casilla de verificación situada junto a la política que desee omitir en la simulación. Para simular todas las políticas, deje todas las políticas seleccionadas.
4. En el panel Policy Simulator, haga lo siguiente:
 - a. En Select service, elija Key Management Service.
 - b. Para simular AWS KMS acciones específicas, en Seleccionar acciones, elija las acciones que desee simular. Para simular todas AWS KMS las acciones, elija Seleccionar todo.
5. (Opcional) El simulador de políticas simula el acceso a todas las claves KMS de forma predeterminada. Para simular el acceso a una clave KMS específica, elija Simulation Settings (Configuraciones de simulación) y, a continuación, escriba el nombre de recurso de Amazon (ARN) de la clave KMS que se simulará.
6. Elija Run Simulation (Ejecutar la simulación).

Puede ver los resultados de la simulación en la sección Results. Repita los pasos del 2 al 6 por cada usuario, grupo y rol de la Cuenta de AWS.

Examen de políticas de IAM con la API de IAM

Puede utilizar la API de IAM para examinar políticas de IAM mediante programación. En los pasos siguientes se ofrece información general sobre cómo hacerlo:

1. Utilice las [ListRoles](#) operaciones y de la API de IAM para obtener todos los usuarios [ListUsers](#) y funciones de la [AWS cuenta para cada entidad que Cuenta de AWS figure como principal](#) en la política clave (es decir, cada principal de cuenta especificado en este formato "Principal": {"AWS": "arn:aws:iam::111122223333:root"}).
2. Para cada usuario y rol de la lista, usa la [SimulatePrincipalPolicy](#) operación en la API de IAM e introduce los siguientes parámetros:
 - Para PolicySourceArn, especifique el nombre de recurso de Amazon (ARN) de un usuario o rol de la lista. Puede especificar solo un PolicySourceArn para cada solicitud SimulatePrincipalPolicy, de modo que debe llamar a esta operación varias veces, una vez por cada usuario y rol de la lista.
 - En la ActionNames lista, especifique todas las acciones de la AWS KMS API que desee simular. Para simular todas las acciones de la AWS KMS API, utilice kms : *. Para probar las acciones individuales de la AWS KMS API, ponga "kms : «, por ejemplo, »" antes de cada acción

de la `APIkms:ListKeys`. Si desea ver una lista completa de las acciones de la API de AWS KMS, consulte [Acciones](#) en la Referencia de la API de AWS Key Management Service.

- (Opcional) Para determinar si los usuarios o roles tienen acceso a determinadas claves KMS, use el parámetro `ResourceArns` para especificar una lista de los nombres de recurso de Amazon (ARN) de las claves KMS. Para determinar si los usuarios o roles tienen acceso a cualquier clave KMS, no use el parámetro `ResourceArns`.

IAM responde a cada solicitud de `SimulatePrincipalPolicy` con una decisión de evaluación: `allowed`, `explicitDeny` o `implicitDeny`. Para cada respuesta que contenga una decisión de evaluación de `allowed`, la respuesta incluye el nombre de la operación de AWS KMS API específica que está permitida. También incluye el ARN de la clave KMS utilizado en la evaluación, si se ha realizado.

Examinar concesiones

Las concesiones son mecanismos avanzados para especificar los permisos que usted o un AWS servicio integrado AWS KMS pueden usar para especificar cómo y cuándo se puede usar una clave de KMS. Las concesiones están asociadas a una clave KMS, y cada concesión contiene la entidad principal que recibe el permiso de usar la clave KMS y una lista de las operaciones permitidas. Las concesiones son una alternativa a la política de claves, y son útiles para casos de uso específicos. Para obtener más información, consulte [Subvenciones en AWS KMS](#).

Para obtener una lista de las concesiones de una clave KMS, utilice la AWS KMS [ListGrants](#) operación. Puede examinar las concesiones de una clave KMS para determinar quién o qué tiene acceso actualmente para utilizar la clave KMS a través de dichas concesiones. En el caso de las subvenciones creadas con `GrantPrincipal`, busque el "GrantPrincipal" campo de `ListGrants` respuesta. En el caso de las subvenciones creadas con `GrantServicePrincipal`, busca el "GrantServicePrincipal" campo de `ListGrants` respuesta.

Por ejemplo, a continuación se ofrece una representación JSON de una concesión que se ha obtenido del comando [list-grants](#) en la 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 averiguar quién o qué tiene acceso para utilizar la clave KMS, busque el elemento `"GranteePrincipal"`. En el ejemplo anterior, la entidad principal beneficiaria es un usuario de rol asumido que está asociado con la instancia EC2 `i-5d476fab`, que la infraestructura de EC2 usa para asociar el volumen de EBS cifrado `vol-5cccfb4e` a la instancia. En este caso, el rol de infraestructura EC2 tiene permiso para usar la clave KMS porque anteriormente ha creado un volumen de EBS cifrado que está protegido por esta clave KMS y, después, ha asociado el volumen a una instancia EC2.

A continuación, se ofrece otro ejemplo de una representación JSON de una concesión que se ha obtenido del comando [list-grants](#) en la AWS CLI. En el siguiente ejemplo, el director del concesionario es otro. Cuenta de 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
}]]

```

En el siguiente ejemplo se muestra una subvención creada con `GranteeServicePrincipal`. Observe que la respuesta incluye un `GranteeServicePrincipal` campo en lugar de `GranteePrincipal` un recurso específico y que el `Constraints` campo contiene un `SourceArn` valor que restringe la concesión a un AWS recurso específico.

```

{"Grants": [{
  "Operations": ["Encrypt", "Decrypt", "GenerateDataKey", "DescribeKey"],
  "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",

```

```

    "Name": "",
    "GranteeServicePrincipal": "service-name.amazonaws.com",
    "GrantId": "a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2",
    "IssuingAccount": "arn:aws:iam::111122223333:root",
    "RetiringServicePrincipal": "service-name.amazonaws.com",
    "Constraints": {
      "SourceArn": "arn:aws:dynamodb:us-east-1:111122223333:table/ExampleTable"
    },
    "CreationDate": 1.718567315E9
  }
}

```

Contexto de cifrado

Note

No puede especificar un contexto de cifrado en una operación criptográfica con una [clave KMS asimétrica](#) o una [clave KMS HMAC](#). Los algoritmos asimétricos y los algoritmos MAC no son compatibles con un contexto de cifrado.

Todas [las operaciones AWS KMS criptográficas](#) con [claves KMS de cifrado simétrico](#) aceptan un contexto de cifrado, un conjunto opcional de pares clave-valor no secretos que pueden contener información contextual adicional sobre los datos. Puede insertar el contexto de cifrado en Encrypt las operaciones AWS KMS para mejorar la autorización y la auditabilidad de las llamadas de descifrado de la API. AWS KMS utiliza el contexto de cifrado como datos autenticados adicionales (AAD) para respaldar el cifrado autenticado. El contexto de cifrado se vincula criptográficamente al texto cifrado, de tal forma que se requiera el mismo contexto de cifrado para descifrar los datos.

El contexto de cifrado no es secreto y no está cifrado. Aparece en texto no cifrado en los [registros de AWS CloudTrail](#) para que pueda utilizarlo para identificar y clasificar las operaciones criptográficas. El contexto de cifrado no debe incluir información confidencial. Le recomendamos que el contexto de cifrado describa los datos que se van a cifrar o descifrar. Por ejemplo, cuando cifre un archivo, puede usar parte de la ruta del archivo como contexto de cifrado.

```

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

```

Por ejemplo, al cifrar volúmenes e instantáneas creados con la operación Amazon [Elastic Block Store](#) (Amazon EBS) [CreateSnapshot](#), Amazon EBS utiliza el ID del volumen como valor de contexto de cifrado.

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

También puede usar el contexto de cifrado para refinar o limitar el acceso a su AWS KMS keys cuenta. Puede utilizar el contexto de cifrado [como una restricción en concesiones](#) y como una [condición en declaraciones de política](#). Las claves de contexto de cifrado y sus valores pueden ser cadenas arbitrarias con aws. Estos valores contrastan con las [etiquetas generadas por AWS](#), como [aws:cloudformation:stack-name](#). Para obtener más información, consulte [kmsEncryptionContext:clave de contexto](#)

Para obtener información sobre cómo utilizar el contexto de cifrado para proteger la integridad de los datos cifrados, consulte la publicación [Cómo proteger la integridad de sus datos cifrados mediante el uso AWS Key Management Service y EncryptionContext](#) en el blog sobre AWS seguridad.

Reglas de contexto de cifrado

AWS KMS aplica las siguientes reglas para las claves y los valores del contexto de cifrado.

- La clave y el valor de un par de contexto de cifrado deben ser cadenas literales simples. Si utiliza un tipo diferente, como un número entero o flotante, AWS KMS lo interpretará como una cadena.
- Las claves y los valores de un contexto de cifrado pueden incluir caracteres Unicode. Si un contexto de cifrado incluye caracteres que no están permitidos en las políticas clave o en las políticas de IAM, no podrá especificar el contexto de cifrado en las claves de condición de la política, como [kms:EncryptionContext:context-key](#) y [kms:EncryptionContextKeys](#). Para obtener más información sobre las reglas clave de los documentos de política de claves, consulte [Formato de la política de claves](#). Para obtener más información sobre las reglas de documentos de políticas de IAM, consulte [Requisitos de nombres de IAM](#) en la Guía del usuario de IAM.

Contexto de cifrado de las políticas

El contexto de cifrado se utiliza principalmente para verificar la integridad y la autenticidad. Sin embargo, también puede utilizar el contexto de cifrado para controlar el acceso a AWS KMS keys de cifrado simétricas en las políticas de claves y las políticas de IAM.

Las claves [EncryptionContextKeyscondicionales kmsEncryptionContext::](#) y [kms:](#) permiten (o deniegan) un permiso solo cuando la solicitud incluye claves de contexto de cifrado o pares clave-valor determinados.

Por ejemplo, la siguiente declaración de política de claves permite al rol `RoleForExampleApp` utilizar la KMS en operaciones `Decrypt`. Utiliza la clave de condición `kms:EncryptionContext:context-key` para conceder este permiso solo cuando el contexto de cifrado de la solicitud incluye un par de contexto de cifrado `AppName:ExampleApp`.

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

Para obtener más información sobre estas claves de condición de contexto de cifrado, consulte [Claves de estado para AWS KMS](#).

Contexto de cifrado de las concesiones

Al [crear una concesión](#), puede incluir [restricciones de concesión](#) que establezcan las condiciones para la concesión de permisos. AWS KMS admite dos restricciones de concesión del contexto de cifrado `EncryptionContextEquals` y `EncryptionContextSubset` ambas implican el [contexto de cifrado](#) en una solicitud de una operación criptográfica. Al utilizar estas restricciones de concesión, los permisos de la concesión solo son efectivos cuando el contexto de cifrado de la solicitud de la operación criptográfica cumple los requisitos de las restricciones de concesión.

Por ejemplo, puede añadir una restricción de `EncryptionContextEquals` concesión a una concesión que permita la [GenerateDataKey](#) operación. Con esta restricción, la concesión solo permite la operación cuando el contexto de cifrado de la solicitud coincide con mayúsculas y minúsculas con el contexto de cifrado de la restricción de concesión.

```
$ aws kms create-grant \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::111122223333:user/exampleUser \  
  --retiring-principal arn:aws:iam::111122223333:role/adminRole \  
  --operations GenerateDataKey \  
  --constraints EncryptionContextEquals={Purpose=Test}
```

Una solicitud como la siguiente del principal beneficiario satisfaría la restricción de `EncryptionContextEquals`.

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

Para obtener más detalles sobre las restricciones de concesiones y , consulte [Uso de restricciones de concesiones](#). Para obtener información detallada sobre las concesiones, consulte [the section called “Concesiones”](#).

Registrar el contexto de cifrado

AWS KMS AWS CloudTrail se utiliza para registrar el contexto de cifrado para que pueda determinar a qué claves y datos del KMS se ha accedido. La entrada de registro muestra exactamente qué clave KMS se ha usado para cifrar o descifrar los datos específicos a los que hace referencia el contexto de cifrado en la entrada de registro.

Important

Como el contexto de cifrado se ha registrado, no debe contener información confidencial.

Almacenar el contexto de cifrado

Para simplificar el uso de cualquier contexto de cifrado al llamar a las operaciones [Decrypt](#) o [ReEncrypt](#), puede almacenar el contexto de cifrado junto con los datos cifrados. Es recomendable

que almacene solo la parte suficiente del contexto de cifrado para crear todo el contexto cuando sea necesario para el cifrado o el descifrado.

Por ejemplo, si el contexto de cifrado es la ruta completa a un archivo, almacene solo parte de esa ruta con el contenido del archivo cifrado. A continuación, cuando necesite todo el contexto de cifrado, reconstrúyalo a partir del fragmento almacenado. Si alguien intenta manipular el archivo (por ejemplo, cambiarlo de nombre o moverlo a otra ubicación), el valor de contexto de cifrado cambia y la solicitud de descifrado produce un error.

Prueba de los permisos

Para poder AWS KMS utilizarlas, debes tener credenciales que AWS puedas utilizar para autenticar tus solicitudes de API. Las credenciales deben incluir permisos para obtener acceso a las claves de KMS y alias. Los permisos vienen determinados por las políticas de claves, las políticas de IAM, las concesiones y los controles de acceso entre cuentas. Además de controlar el acceso a las claves de KMS, puede controlar el acceso a su CloudHSM y a los almacenes de claves personalizados.

Puede especificar el parámetro `DryRun` de la API para comprobar que dispone de los permisos necesarios para utilizar las claves de AWS KMS . También puedes utilizarlas `DryRun` para comprobar que los parámetros de la solicitud en una llamada a la AWS KMS API estén especificados correctamente.

Temas

- [¿Cuál es el DryRun parámetro?](#)
- [Especificar DryRun con la API](#)

¿Cuál es el DryRun parámetro?

`DryRun` es un parámetro de API opcional que se especifica para comprobar que las llamadas a la API de AWS KMS se realizan correctamente. Use `DryRun` para probar la llamada a la API antes de hacer una llamada a AWS KMS. Puede comprobar lo siguiente:

- Que cuenta con los permisos necesarios para utilizar las claves de AWS KMS .
- Que ha especificado correctamente los parámetros de la llamada.

AWS KMS admite el uso del `DryRun` parámetro en determinadas acciones de la API:

- [CreateGrant](#)
- [Decrypt](#)
- [DeriveSharedSecret](#)
- [Encrypt](#)
- [GenerateDataKey](#)
- [GenerateDataKeyPair](#)
- [GenerateDataKeyPairWithoutPlaintext](#)
- [GenerateDataKeyWithoutPlaintext](#)
- [GenerateMac](#)
- [ReEncrypt](#)
- [RetireGrant](#)
- [RevokeGrant](#)
- [Sign](#)
- [Verificar](#)
- [VerifyMac](#)

El uso del parámetro `DryRun` generará cargos y se facturará como una solicitud a la API estándar. Para obtener más información sobre AWS KMS los precios, consulta [AWS Key Management Service Precios](#).

Todas las solicitudes a la API que utilizan el parámetro `DryRun` se aplican a la cuota de solicitudes a la API y pueden dar lugar a una excepción de limitación si se supera una cuota de solicitudes a la API. Por ejemplo, llamar a [Decrypt](#) con `DryRun` o sin `DryRun` cuenta para la misma cuota de operaciones criptográficas. Consulte [Limitar las solicitudes AWS KMS](#) para obtener más información.

Cada llamada a una operación de AWS KMS API se captura como un evento y se registra en un AWS CloudTrail registro. El resultado de cualquier operación que especifique el `DryRun` parámetro aparece en el CloudTrail registro. Para obtener más información, consulte [Registro AWS KMS Llamadas a la API con AWS CloudTrail](#).

Especificar DryRun con la API

Para usarlo `DryRun`, especifique el `--dry-run` parámetro en los AWS CLI comandos y las llamadas a la AWS KMS API que admiten el parámetro. Cuando lo hagas, AWS KMS verificará si la llamada

se ha realizado correctamente. AWS KMS las llamadas que utilice siempre DryRun fallarán y devolverán un mensaje con información sobre el motivo por el que se produjo el error en la llamada. El mensaje puede incluir las siguientes excepciones:

- `DryRunOperationException`: la solicitud se realizaría correctamente si DryRun no se especificara.
- `ValidationException`: se produjo un error en la solicitud al especificar un parámetro de API incorrecto.
- `AccessDeniedException`: no tiene permisos para realizar la acción de API especificada en el recurso de KMS.

Por ejemplo, el siguiente comando usa la [CreateGrant](#) operación y crea una concesión que permite a los usuarios autorizados a asumir la `keyUserRole` función llamar a la operación de [descifrado](#) con una clave [KMS simétrica](#) especificada. Se especifica el parámetro DryRun.

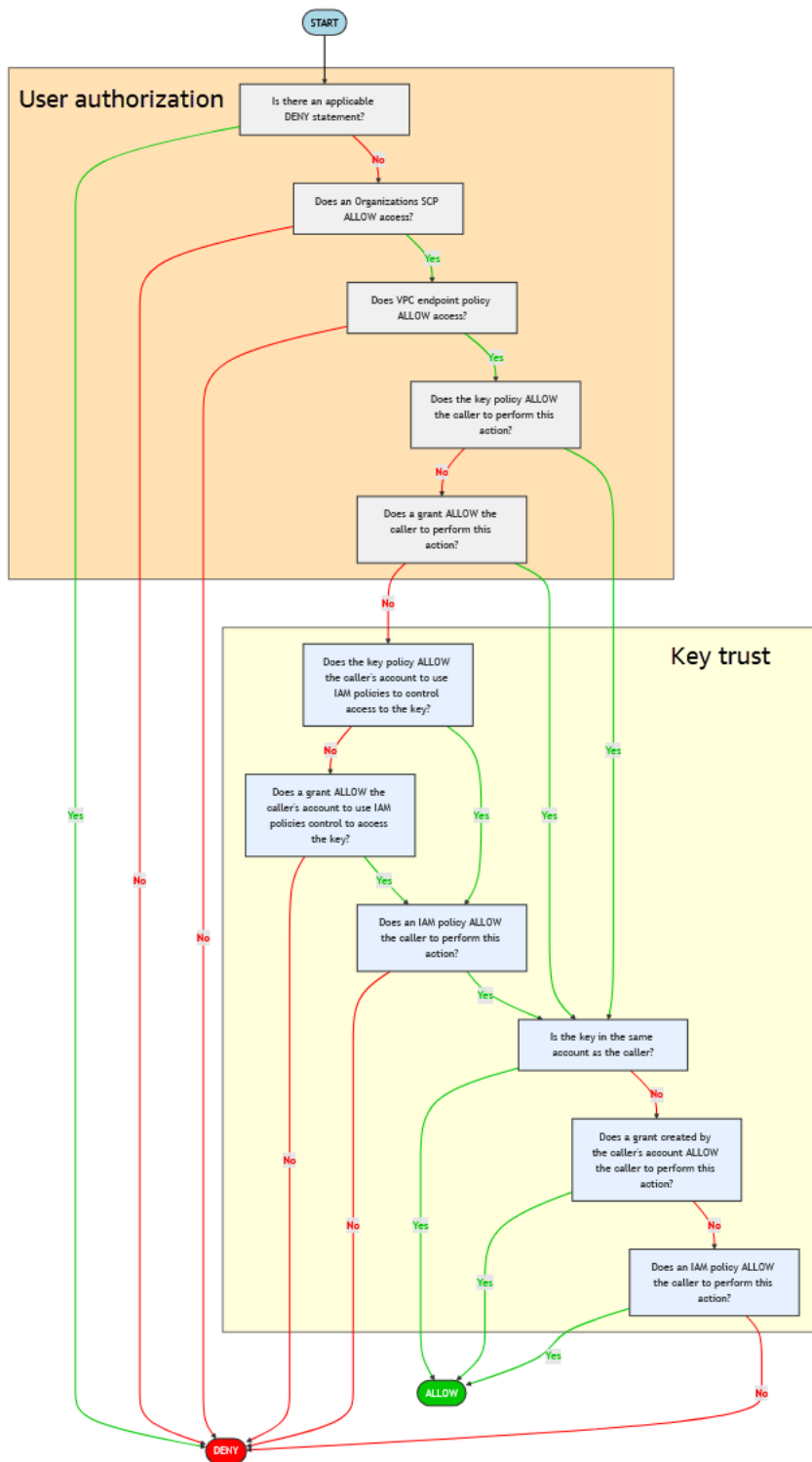
```
$ aws kms create-grant \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \  
  --operations Decrypt \  
  --dry-run
```

Solución de problemas de AWS KMS permisos

Al autorizar el acceso a una clave KMS, AWS KMS evalúa lo siguiente:

- La [política de claves](#) asociada a la clave KMS. La política de claves siempre se define en la región Cuenta de AWS y propietaria de la clave de KMS.
- Todas las [políticas de IAM](#) asociadas al usuario o rol que realiza la solicitud. Las políticas de IAM que rigen el uso de una clave KMS por una entidad principal siempre se definen en la Cuenta de AWS.
- Todas las [concesiones](#) que afectan a la clave KMS.
- Otros tipos de políticas que podrían aplicarse a la solicitud de usar la clave KMS, como [Políticas de control de servicios de AWS Organizations](#) y [Políticas de punto de enlace de la VPC](#). Estas políticas son opcionales y permiten todas las acciones de forma predeterminada, pero puede usarlas para restringir los permisos otorgados a las entidades principales.

AWS KMS evalúa estos mecanismos de política en conjunto para determinar si se permite o deniega el acceso a la clave de KMS. Para ello, AWS KMS utiliza un proceso similar al que se muestra en el siguiente diagrama de flujo. El siguiente diagrama de flujo ofrece una representación visual del proceso de evaluación de las políticas.



Este diagrama de flujo se divide en dos partes. Las partes parecen secuenciales, pero se suelen evaluar al mismo tiempo.

- El uso de la autorización determina si puede utilizar una clave KMS en función de su política de claves, sus políticas de IAM, sus concesiones y otras políticas aplicables.
- La confianza en la clave determina si debe confiar en una clave KMS que se le ha permitido utilizar. En general, usted confía en los recursos que tiene. Cuenta de AWS Sin embargo, también puede estar seguro de utilizar las claves de KMS de otra forma Cuenta de AWS si una política de concesión o de IAM de su cuenta le permite utilizar la clave de KMS.

Puede utilizar este diagrama de flujo para saber por qué se ha permitido o denegado que un intermediario utilice una clave KMS. También puede utilizarlo para evaluar sus políticas y concesiones. Por ejemplo, el diagrama de flujo muestra que se puede negar el acceso a un intermediario mediante una declaración DENY explícita, o por la ausencia de una declaración ALLOW explícita, en la política de claves, la política de IAM o la concesión.

El diagrama de flujo puede explicar algunos escenarios comunes de permisos.

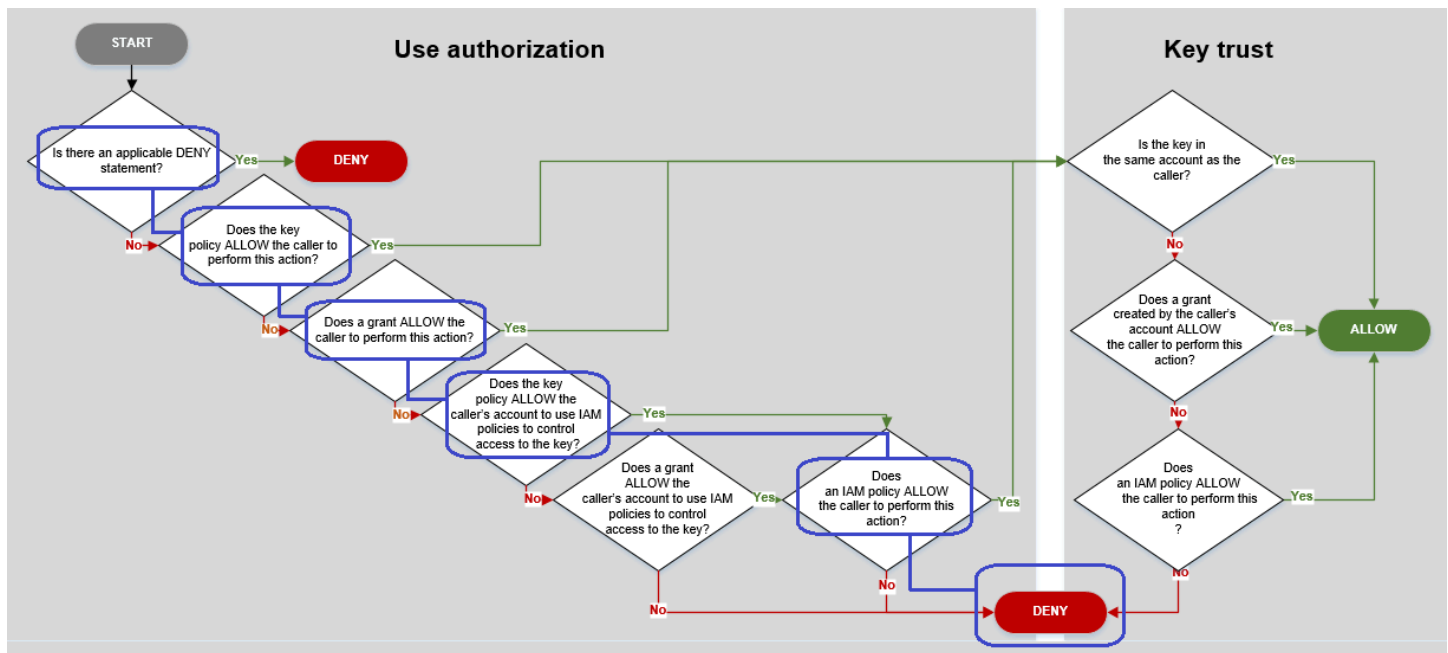
Ejemplos de permisos

- [Ejemplo 1: Se deniega al usuario el acceso a una clave KMS de su Cuenta de AWS](#)
- [Ejemplo 2: El usuario asume el rol con permiso para usar una clave de KMS en otro Cuenta de AWS](#)

Ejemplo 1: Se deniega al usuario el acceso a una clave KMS de su Cuenta de AWS

Alice es una usuaria de IAM en el Cuenta de AWS 111122223333. Se le ha denegado el acceso a una clave KMS en la misma Cuenta de AWS. ¿Por qué no puede utilizar Alice la clave KMS?

En este caso, se le deniega el acceso a Alice a la clave KMS porque no hay ninguna política de claves, política de IAM o concesión que proporcione los permisos necesarios. La política clave de la clave de KMS permite utilizar políticas de IAM Cuenta de AWS para controlar el acceso a la clave de KMS, pero ninguna política de IAM permite a Alice usar la clave de KMS.



Considere las políticas relevantes para este ejemplo.

- La clave KMS que Alice quiere utilizar tiene la [política de claves predeterminada](#). Esta política [permite que la Cuenta de AWS](#) que posee la clave KMS para utilizar políticas de IAM para controlar el acceso a la clave KMS. Esta política de claves satisface la condición ¿PERMITE la política de claves que la cuenta de los intermediarios utilice políticas de IAM para controlar el acceso a clave? del diagrama de flujo.

JSON

```

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

- Sin embargo, no hay ninguna política de claves, política de IAM ni concesión que conceda a Alice permiso para usar la clave KMS. Por lo tanto, se deniega a Alice el permiso para utilizar la clave KMS.

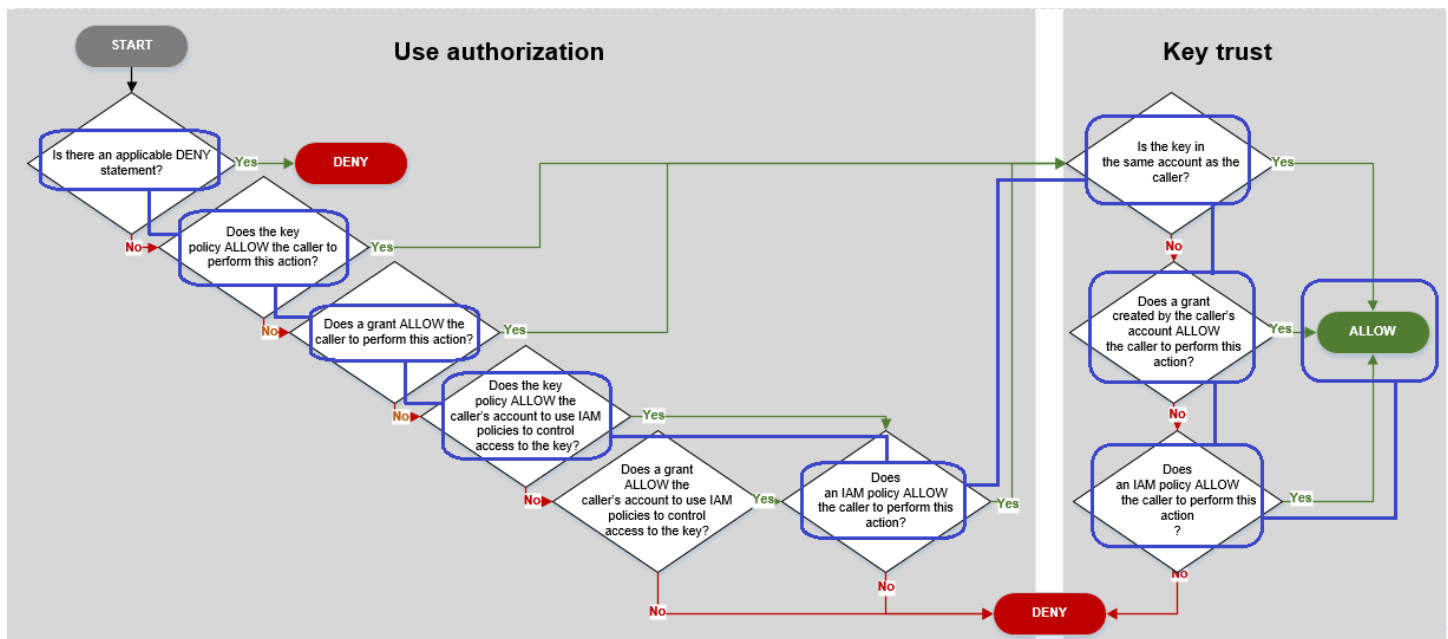
Ejemplo 2: El usuario asume el rol con permiso para usar una clave de KMS en otro Cuenta de AWS

Roberto es un usuario de la cuenta 1 (111122223333). Se le permite usar una clave KMS en la cuenta 2 (444455556666) en [operaciones criptográficas](#). ¿Cómo es posible?

Tip

Al evaluar los permisos entre cuentas, recuerde que la política de claves se especifica en la cuenta de la clave KMS. La política de IAM se especifica en la cuenta del intermediario, incluso cuando el intermediario está en una cuenta diferente. Para obtener más detalles sobre cómo proporcionar acceso entre cuentas a claves KMS, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

- La política de claves de la clave KMS de la cuenta 2 permite que la cuenta 2 utilice políticas de IAM para controlar el acceso a la clave KMS.
- La política de claves de la clave KMS de la cuenta 2 permite que la cuenta 1 utilice la clave KMS en operaciones criptográficas. Sin embargo, la cuenta 1 debe utilizar políticas de IAM para conceder acceso a la clave KMS a sus entidades principales.
- Una política de IAM de la cuenta 1 permite al rol Engineering utilizar la clave KMS de la cuenta 2 para las operaciones criptográficas.
- Roberto, un usuario de la cuenta 1, tiene permiso para asumir el rol Engineering.
- Roberto confía en esta clave KMS, porque, aunque no se encuentre en su cuenta, una política de IAM de su cuenta le otorga permiso explícito para utilizar esta clave KMS.



Considere las políticas que permiten a Roberto, un usuario de la cuenta 1, utilizar la clave KMS de la cuenta 2.

- La política de claves de la clave KMS permite que la cuenta 2 (444455556666, la cuenta propietaria de la clave KMS) utilice políticas de IAM para controlar el acceso a la clave KMS. Esta política de claves permite también que la cuenta 1 (111122223333) utilice la clave KMS en operaciones criptográficas (especificado en el elemento `Action` de la declaración de política). Sin embargo, ninguna persona de la cuenta 1 puede usar la clave KMS de la cuenta 2 hasta que la cuenta 1 defina políticas de IAM que concedan a las entidades principales acceso a la clave KMS.

En el diagrama de flujo, esta política de claves de la cuenta 2 satisface la condición ¿PERMITE la política de claves que la cuenta de los intermediarios utilice políticas de IAM para controlar el acceso a la clave?

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": "*"
  }
]
}

```

- Una política de IAM en la persona que llama Cuenta de AWS (cuenta 1, 111122223333) otorga al principal permiso para realizar operaciones criptográficas con la clave KMS en la cuenta 2 (444455556666). El elemento `Action` delega a la entidad principal los mismos permisos que la política de claves de la cuenta 2 concedió a la cuenta 1. Para conceder estos permisos al rol Engineering en la cuenta 1, [esta política en línea está incrustada](#) en el rol Engineering.

Las políticas de IAM entre cuentas como esta son eficaces solo cuando la política de claves de la clave KMS de la cuenta 2 concede a la cuenta 1 permiso para utilizar la clave KMS. Además, la cuenta 1 solo puede conceder permiso a sus entidades principales para realizar las acciones que la política de claves concedió a la cuenta.

En el diagrama de flujo, esto satisface la condición ¿Permite una política de IAM que el intermediario realice esta acción?

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"
        ]
      }
    ]
  }
}

```

- El último elemento necesario es la definición del rol Engineering en la cuenta 1. El elemento AssumeRolePolicyDocument del rol permite a Roberto asumir el rol 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 glosario de control de acceso

En el siguiente tema se describen los términos y conceptos importantes del control de AWS KMS acceso.

Autenticación

La autenticación es el proceso de verificación de su identidad. Para enviar una solicitud a AWS KMS, debe iniciar sesión AWS con sus AWS credenciales.

Autorización

La autorización proporciona el permiso para enviar solicitudes para crear, administrar o usar AWS KMS recursos. Por ejemplo, debe estar autorizado a utilizar una clave KMS en una operación criptográfica.

Para controlar el acceso a sus AWS KMS recursos, utilice [las políticas clave, las políticas](#) de [IAM y las subvenciones](#). Cada clave de KMS debe tener una política de claves. Si la política de claves lo permite, también puede utilizar las políticas y autorizaciones de IAM para conceder acceso a la clave KMS a las entidades principales. Para restringir la autorizaciones, puede utilizar [claves de condición](#) que pueden permitir o denegar el acceso solo cuando una solicitud o recurso cumple las condiciones que especifica. También puede permitir el acceso a las entidades principales en las que confía en [otras cuentas de Cuentas de AWS](#).

Autenticación con identidades

La autenticación es la forma de iniciar sesión para AWS usar tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte [Cómo iniciar sesión en la Cuenta de AWS](#) en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte [AWS Signature Version 4 para solicitudes de API](#) en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear un Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz que tiene acceso completo a todos Servicios de AWS los recursos. Se recomienda encarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver las tareas que requieren credenciales de usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio empresarial, del proveedor de identidades web o al Directory Service que se accede Servicios de AWS mediante credenciales de una fuente de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, se recomienda AWS IAM Identity Center. Para obtener más información, consulte [¿Qué es el Centro de identidades de IAM?](#) en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte [Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte [Casos de uso para usuarios de IAM](#) en la Guía del usuario de IAM.

Roles de IAM

Un [Rol de IAM](#) es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol [cambiando de un rol de usuario a uno de IAM \(consola\)](#) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte [Métodos para asumir un rol](#) en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Administración del acceso con políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte [Información general de políticas de JSON](#) en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de política de permisos JSON que asocia a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte [Definición de permisos de IAM personalizados con políticas administradas por el cliente](#) en la Guía del usuario de IAM.

Las políticas basadas en identidad pueden ser políticas insertadas (incrustadas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte [Selección entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

Políticas basadas en recursos

Una [política AWS KMS clave es una política](#) basada en recursos que controla el acceso a una clave de KMS. Cada clave de KMS debe tener una política de claves. Puede utilizar otro mecanismo de autorizaciones para permitir el acceso a la clave KMS, pero solo si la política de claves lo permite. (Puede utilizar una política de IAM para denegar el acceso a una clave KMS incluso si la política de claves no lo permite explícitamente).

Las políticas basadas en recursos son documentos de política JSON que puede asociar a un recurso, como, por ejemplo, una clave KMS, para controlar el acceso a un recurso específico. Las políticas basadas en recursos definen las acciones que puede realizar una entidad principal en ese recurso y en qué condiciones. No se especifica el recurso en una política basada en recursos, pero se debe especificar un principal, como cuentas, usuarios, roles, usuarios federados o. Servicios de AWS Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio que administra el recurso. No puede usar políticas AWS administradas de IAM, como la [política `AWSKeyManagementServicePowerUser` administrada, en una política basada](#) en recursos.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte [Políticas de control de servicios](#) en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCPs): establece los permisos máximos disponibles para los recursos de tus cuentas. Para obtener más información, consulte [Políticas de control de recursos \(RCPs\)](#) en la Guía del AWS Organizations usuario.
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

AWS KMS recursos

En AWS KMS, el recurso principal es un AWS KMS key. AWS KMS también admite un [alias](#), un recurso independiente que proporciona un nombre descriptivo para una clave de KMS. Algunas AWS KMS operaciones permiten usar un alias para identificar una clave de KMS.

Cada instancia de una clave KMS o alias tiene un único [nombre de recurso de Amazon](#) (ARN) con un formato estándar. En AWS KMS los recursos, el nombre del AWS servicio es kms.

- AWS KMS key

Formato de ARN:

```
arn:AWS partition name:AWS service name:Región de AWS:Cuenta de AWS ID:key/key ID
```

Ejemplo de ARN:

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

- Alias

Formato de ARN:

```
arn:AWS partition name:AWS service name:Región de AWS:Cuenta de AWS ID:alias/alias name
```

Ejemplo de ARN:

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

AWS KMS proporciona un conjunto de operaciones de API para que funcionen con sus AWS KMS recursos. Para obtener más información sobre cómo identificar las claves de KMS en las operaciones de AWS KMS API Consola de administración de AWS y las operaciones de API, consulte [Identificadores clave \(\) KeyId](#). Para obtener una lista de AWS KMS operaciones, consulta la [referencia de la AWS Key Management Service API](#).

Crear una clave de KMS.

Puede crearla AWS KMS keys en Consola de administración de AWS, o mediante la [CreateKey](#) operación o el [AWS::KMS::Key AWS CloudFormation recurso](#). Durante este proceso, establece la política de claves para la clave KMS, que puede cambiar en cualquier momento. También puede seleccionar los siguientes valores que definen el tipo de clave KMS que va a crear. No puede cambiar estas propiedades después de que se cree la clave de KMS.

Tipo de clave KMS

El tipo de clave es una propiedad que determina el tipo de clave criptográfica que se crea. AWS KMS ofrece tres tipos de claves para proteger los datos:

- Claves simétricas del estándar de cifrado avanzado (AES)

Claves de 256 bits que se utilizan en el modo Galois Counter Mode (GCM) del AES para proporcionar datos autenticados de menos de 4 KB encryption/decryption de tamaño. Este es el tipo de clave más común y se utiliza para proteger otras claves de cifrado de datos utilizadas en sus aplicaciones y, de este modo, cifrar los datos en su Servicios de AWS nombre.

- RSA, curva elíptica o claves asimétricas (solo para regiones de SM2 China)

Estas claves están disponibles en varios tamaños y admiten muchos algoritmos. Se pueden usar para cifrar y descifrar, firmar y verificar u obtener operaciones de secretos compartidos según el algoritmo elegido.

- Claves simétricas para realizar operaciones de códigos de autenticación de mensajes basados en hash (HMAC)

Estas claves son claves de 256 bits que se utilizan para las operaciones de firma y verificación.

Las claves KMS no se pueden exportar desde el servicio en texto sin formato. Las generan los módulos de seguridad de hardware (HSMs) utilizados por el servicio y solo se pueden usar dentro de ellos. Esta es la propiedad de seguridad fundamental AWS KMS para garantizar que las claves no se vean comprometidas.

Uso de clave

El uso de la clave es una propiedad que determina las operaciones criptográficas que esta admite. Las claves KMS pueden tener los siguientes usos de clave: ENCRYPT_DECRYPT, SIGN_VERIFY, GENERATE_VERIFY_MAC o KEY_AGREEMENT. Cada clave KMS solo puede tener un uso. Esto sigue las prácticas recomendadas de uso de claves según la [publicación especial](#)

[800-57 del Instituto Nacional de Estándares y Tecnología \(NIST\), Recomendaciones para la administración de claves](#), sección 5.2, Uso de claves. El uso de una clave KMS para más de un tipo de operaciones hace que el producto de ambas operaciones sea más vulnerable a ataques.

Especificación de clave

La especificación de clave es una propiedad que representa la configuración criptográfica de la clave. El significado de la especificación de clave difiere con el tipo de clave.

Para las claves KMS, la especificación de clave determina si la clave KMS es simétrica o asimétrica. También determina el tipo de su material de clave y los algoritmos que admite.

La especificación de clave predeterminada, [SYMMETRIC_DEFAULT](#), representa una clave de cifrado simétrico de 256 bits. Para obtener una descripción detallada de todas las especificaciones de clave compatibles, consulte [Referencia de especificaciones de clave](#).

Origen del material de claves

El origen del material de claves es una propiedad de la clave KMS que identifica el origen del material de claves en la clave KMS. Usted elija el origen del material de claves cuando crea la clave KMS y no se puede cambiar. La fuente del material de clave afecta a las características de seguridad, durabilidad, disponibilidad, latencia y rendimiento de la clave de KMS.

Cada clave KMS incluye una referencia al material de clave en los metadatos. El origen del material de claves de cifrado simétrico, las claves KMS pueden variar. Puede utilizar el material de claves que genera AWS KMS, el material de claves que se genera en un [almacén de claves personalizado](#), o bien [importar su propio material de claves](#).

De forma predeterminada, cada clave KMS tiene un material de claves único. Sin embargo, puede crear un conjunto de [claves de varias regiones](#) con el mismo material de claves.

Las claves de KMS pueden tener uno de los siguientes valores de origen del material clave: AWS_KMS, EXTERNAL ([material de clave importado](#)), AWS_CLOUDHSM ([clave de KMS en un almacén de AWS CloudHSM claves](#)) o EXTERNAL_KEY_STORE ([clave de KMS en un almacén de claves externo](#)).

Temas

- [Permisos para crear claves KMS](#)
- [Elección del tipo de clave KMS que se va a crear](#)
- [Creación de claves de KMS de cifrado simétricas](#)

- [Creación de una clave de KMS asimétrica](#)
- [Creación de una clave KMS HMAC](#)
- [Creación de claves primarias de varias regiones](#)
- [Creación de claves de réplica de varias regiones](#)
- [Creación de una clave KMS con material de claves importado](#)
- [Crear una clave KMS en un almacén de AWS CloudHSM claves](#)
- [Creación de una clave KMS en un almacén de claves externo](#)

Permisos para crear claves KMS

Para crear una clave KMS en la consola o mediante la APIs, debe tener el siguiente permiso en una política de IAM. Siempre que sea posible, use [claves de condición](#) para limitar los permisos. Por ejemplo, puede usar la clave de KeySpec condición [kms:](#) en una política de IAM para permitir que los principales creen solo claves de cifrado simétricas.

Para obtener un ejemplo de una política de IAM para las principales entidades que crean claves, consulte [Permitir a un usuario crear claves KMS](#).

Note

Tenga cuidado al dar permiso a las entidades principales para administrar etiquetas y alias. El cambio de etiqueta o alias puede permitir o denegar permiso a la clave administrada por el cliente. Para obtener más información, consulte [ABAC para AWS KMS](#).

- [kms: CreateKey](#) es obligatorio.
- [kms: CreateAlias](#) es necesario para crear una clave KMS en la consola, donde se requiere un alias para cada nueva clave KMS.
- [kms: TagResource](#) es necesario añadir etiquetas al crear la clave KMS.
- [iam: CreateServiceLinkedRole](#) es necesario para crear claves principales multirregionales. Para obtener más información, consulte [Control del acceso a claves de varias regiones](#).

El PutKeyPolicy permiso [kms:](#) no es necesario para crear la clave KMS. El permiso `kms:CreateKey` incluye permiso para establecer la política de clave inicial. Pero debe agregar este permiso a la política de clave mientras crea la clave KMS para asegurarse de que puede controlar el acceso

a la clave KMS. La alternativa es utilizar el [BypassLockoutSafetyCheck](#) parámetro, lo cual no se recomienda.

Las claves KMS pertenecen a la AWS cuenta en la que se crearon. El usuario de IAM que crea una clave KMS no se considera el propietario de la clave y no tiene permiso automáticamente para usar o administrar la clave KMS que creó. Al igual que cualquier otra entidad principal, el creador de la clave necesita obtener permiso a través de una política de claves, una política de IAM o una concesión. Sin embargo, las entidades principales que tienen el permiso de `kms:CreateKey` pueden establecer la política de clave inicial y darse permiso a ellas mismas para usar o administrar la clave.

Elección del tipo de clave KMS que se va a crear

El tipo de clave KMS que crea varía en gran medida en función de cómo tiene previsto utilizar la clave KMS, los requisitos de seguridad y los requisitos de autorización. El tipo de clave y el uso de la clave de una clave KMS determinan qué operaciones criptográficas puede realizar dicha clave. Cada clave KMS tiene solo un uso de claves. El uso de una clave KMS para más de un tipo de operaciones hace que el producto de todas las operaciones sea más vulnerable a ataques.

Para permitir que los directores creen claves de KMS solo para un uso de clave determinado, utilice la clave de KeyUsage condición [kms:](#). También puede utilizar la clave de condición `kms:KeyUsage` para permitir que las entidades principales llamen a las operaciones de la API para una clave KMS basada en su uso de claves. Por ejemplo, puede permitir un permiso que deshabilite una clave KMS solo si su uso de clave es `SIGN_VERIFY`.

Utilice las siguientes directrices para determinar qué tipo de clave KMS necesita en función de su caso de uso.

Cifrar y descifrar datos

Utilice una [clave KMS simétrica](#) para la mayoría de los casos de uso que requieren cifrar y descifrar datos. El algoritmo de cifrado simétrico que utiliza AWS KMS es rápido, eficaz y asegura la confidencialidad y la autenticidad de los datos. Admite el cifrado autenticado con datos autenticados adicionales (AAD), definidos como un [contexto de cifrado](#). Este tipo de clave KMS requiere que tanto el remitente como el destinatario de los datos cifrados dispongan de AWS credenciales válidas para realizar llamadas AWS KMS.

Si su caso de uso requiere que los usuarios que no pueden llamar AWS los cifren fuera o no AWS KMS, [las claves KMS asimétricas](#) son una buena opción. Puede distribuir la parte pública de la clave KMS asimétrica para permitir que estos usuarios cifren los datos. Las aplicaciones que

necesitan descifrar estos datos pueden utilizar la parte privada de la clave KMS asimétrica en AWS KMS.

Firmar mensajes y verificar firmas

Para firmar mensajes y verificar firmas, tiene que utilizar una [clave KMS asimétrica](#). Puede usar una clave KMS con una [especificación clave que represente un par de claves RSA, un par de claves de curva elíptica \(ECC\)](#), un par de claves ML-DSA o un par de claves (solo para regiones de China SM2). La especificación de clave que seleccione la determina el algoritmo de firma que desea utilizar. Se recomiendan los algoritmos de firma ECDSA que admiten los pares de claves ECC en lugar de los algoritmos de firma RSA. Utilice un par de claves de ML-DSA al migrar de claves de RSA o de ECC a claves poscuánticas. Sin embargo, es posible que necesite utilizar una especificación clave y un algoritmo de firma específicos para ayudar a los usuarios a verificar las firmas fuera de AWS.

Cifrado con pares de claves asimétricas

Para cifrar los datos con un par de claves asimétricas, debe usar una clave [KMS asimétrica con una especificación de clave RSA o una especificación de clave \(solo para regiones de SM2 China\)](#). Para cifrar los datos de AWS KMS con la clave pública de un par de claves KMS, utilice la operación [Encrypt](#) (Cifrar). También puede [descargar la clave pública](#) y compartirla con las partes que necesiten cifrar los datos fuera de ella. AWS KMS

Cuando descarga la clave pública de una clave KMS asimétrica, puede utilizarla fuera de AWS KMS. Sin embargo, ya no está sujeta a los controles de seguridad que protegen la clave KMS. AWS KMS Por ejemplo, no puede usar políticas o concesiones AWS KMS clave para controlar el uso de la clave pública. Tampoco puede controlar si la clave se utiliza únicamente para el cifrado y el descifrado mediante los algoritmos de cifrado AWS KMS compatibles. Para obtener más información, consulte [Consideraciones especiales para la descarga de claves públicas](#).

Para descifrar los datos que se cifraron con la clave pública externa AWS KMS, ejecute la operación de [descifrado](#). La operación Decrypt falla si los datos se cifran con una clave pública de una clave KMS con un uso de la clave de SIGN_VERIFY. También se producirá un error si se cifró mediante un algoritmo que AWS KMS no sea compatible con la especificación de clave que ha seleccionado. Para obtener más información sobre las especificaciones de claves y los algoritmos compatibles, consulte [Referencia de especificaciones de clave](#).

Para evitar estos errores, cualquier persona que utilice una clave pública fuera de ella AWS KMS debe almacenar la configuración de la clave. La AWS KMS consola y la [GetPublicKey](#) respuesta proporcionan la información que debe incluir al compartir la clave pública.

Obtención de secretos compartidos

Para obtener secretos compartidos, utilice una clave KMS con una [curva elíptica estándar del NIST](#) o material clave (solo para las regiones de [SM2China](#)). AWS KMS utiliza el [cofactor criptográfico de curva elíptica Diffie-Hellman Primitive](#) (ECDH) para establecer un acuerdo de claves entre dos pares mediante la obtención de un secreto compartido a partir de sus pares de claves público-privadas de curva elíptica. Puede utilizar el secreto compartido sin procesar que devuelve la [DeriveSharedSecret](#) operación para obtener una clave simétrica que pueda cifrar y descifrar los datos que se envían entre dos partes, o generarlos y verificarlos. HMACs AWS KMS recomienda que siga las [recomendaciones del NIST para la derivación de claves](#) cuando utilice el secreto compartido sin procesar para obtener una clave simétrica.

Generación y verificación de códigos HMAC

Para generar y verificar códigos de autenticación de mensajes basados en hash, utilice una clave KMS HMAC. Al crear una clave HMAC AWS KMS, AWS KMS crea y protege el material de la clave y se asegura de utilizar los algoritmos MAC correctos para la clave. Los códigos HMAC también se pueden utilizar como números pseudoaleatorios y, en ciertos escenarios, para la firma simétrica y la tokenización.

Las claves KMS HMAC son claves simétricas. Al crear una clave KMS HMAC en la consola de AWS KMS, elija el tipo de clave `Symmetric`.

Úselo con servicios AWS

Para crear una clave KMS para usarla con un [AWS servicio integrado AWS KMS](#), consulte la documentación del servicio. AWS los servicios que cifran los datos requieren una clave [KMS de cifrado simétrico](#).

Además de estas consideraciones, las operaciones criptográficas en las claves KMS con diferentes especificaciones de clave tienen diferentes precios y diferentes cuotas de solicitud. Para obtener información sobre precios de AWS KMS, consulte [AWS Key Management Service precios](#). Para obtener más información acerca de las cuotas de solicitud, consulte [Cuotas de solicitudes](#).

Creación de claves de KMS de cifrado simétricas

En este tema se explica cómo crear la clave KMS básica, una clave [KMS de cifrado simétrico](#) para una sola región con material de clave de. AWS KMS Puede utilizar esta clave KMS para proteger sus recursos en un Servicio de AWS.

[Puede crear claves KMS de cifrado simétrico en la AWS KMS consola, mediante la CreateKeyAPI o mediante la AWS::KMS::Key CloudFormation plantilla.](#)

La especificación de clave predeterminada, [SYMMETRIC_DEFAULT](#), es la especificación de claves para las claves KMS de cifrado simétricas. Al seleccionar el tipo de clave simétrica y el uso de claves de cifrado y descifrado en la AWS KMS consola, se selecciona la especificación de la clave. SYMMETRIC_DEFAULT En la [CreateKey](#) operación, si no especifica ningún KeySpec valor, se selecciona SYMMETRIC_DEFAULT. Si no tiene un motivo para utilizar una especificación de clave diferente, SYMMETRIC_DEFAULT es una buena opción.

Para obtener información acerca de las cuotas que se aplican a las claves KMS, consulte [Cuotas](#).

Uso de la consola AWS KMS

Puede usar el Consola de administración de AWS para crear AWS KMS keys (claves KMS).

Important

No incluya información confidencial en el alias, la descripción ni las etiquetas. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Elija Crear clave.
5. Para crear una clave KMS de cifrado simétrica, para Key type (Tipo de clave) seleccione Symmetric (Simétrica).
6. En Key usage (Uso de claves), se selecciona la opción Encrypt and decrypt (Cifrar y descifrar) para usted.
7. Elija Siguiente.
8. Escriba un alias para la clave KMS. El nombre del alias no puede empezar por **aws/**. Amazon Web Services se reserva el **aws/** prefijo para representarlo Claves administradas por AWS en su cuenta.

 Note

Agregar, eliminar o actualizar un alias puede permitir o denegar el permiso a la clave KMS. Para más detalles, consulte [ABAC para AWS KMS](#) y [Uso de alias para controlar el acceso a las claves KMS](#).

Un alias es un nombre de visualización que puede usar para identificar a una clave KMS. Le recomendamos que elija un alias que indique el tipo de datos que piensa proteger o la aplicación que piensa usar con la clave KMS.

Los alias son necesarios para crear una clave KMS en la Consola de administración de AWS. Son opcionales cuando se utiliza la [CreateKey](#) operación.

9. (Opcional) Escriba una descripción de la clave KMS.

Puede agregar una descripción ahora o actualizarla en cualquier momento, a menos que el [estado de la clave](#) sea Pending Deletion o Pending Replica Deletion. Para añadir, cambiar o eliminar la descripción de una clave gestionada por el cliente existente, edite la descripción en la página de detalles de la clave KMS de la operación Consola de administración de AWS o utilice la [UpdateKeyDescription](#) operación.

10. (Opcional) Escriba una clave de etiqueta y un valor de etiqueta opcional. Para agregar más de una etiqueta a la clave KMS, elija Add tag (Agregar etiqueta).

 Note

Etiquetar o quitar las etiquetas de la clave KMS puede permitir o denegar permiso a la clave KMS. Para más detalles, consulte [ABAC para AWS KMS](#) y [Uso de etiquetas para controlar el acceso a las claves KMS](#).

Al añadir etiquetas a AWS los recursos, AWS genera un informe de asignación de costes con el uso y los costes agregados por etiquetas. Las etiquetas también pueden utilizarse para controlar el acceso a una clave KMS. Para obtener información acerca del etiquetado de claves KMS, consulte [Etiquetas en AWS KMS](#) y [ABAC para AWS KMS](#).

11. Elija Siguiente.

12. Seleccione los usuarios y roles de IAM que pueden administrar la clave de KMS.

Notas

Esta política clave proporciona el control Cuenta de AWS total de esta clave de KMS. Permite a los administradores de cuentas utilizar las políticas de IAM para dar permiso a otras entidades principales para administrar la clave KMS. Para obtener más información, consulte [the section called “Política de claves predeterminada”](#).

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;. La AWS KMS consola agrega administradores de claves a la política de claves bajo el identificador de la declaración "Allow access for Key Administrators". La modificación de este identificador de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

13. (Opcional) Para evitar que los usuarios y los roles de IAM seleccionados eliminen esta clave de KMS, en la sección Eliminación de claves situada en la parte inferior de la página, desactive la casilla Permitir que los administradores de claves eliminen esta clave.

14. Elija Siguiente.

15. Seleccione los usuarios y roles de IAM que pueden usar la clave en [operaciones criptográficas](#)

Notas

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;. La AWS KMS consola agrega los usuarios clave a la política clave bajo los identificadores de la declaración "Allow use of the key" y "Allow attachment of persistent resources". La modificación de estos identificadores de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

16. (Opcional) Puede permitir que otras Cuentas de AWS usen esta clave KMS para operaciones criptográficas. Para ello, en la Cuentas de AWS sección Otros, en la parte inferior de la página,

selecciona **Añadir otra Cuenta de AWS** e introduce el número de Cuenta de AWS identificación de una cuenta externa. Para agregar varias cuentas externas, repita este paso.

 Note

Para permitir que las entidades principales de las cuentas externas usen la clave KMS, los administradores de la cuenta externa también deben crear las políticas de IAM que proporcionan estos permisos. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

17. Elija **Siguiente**.
18. Revise las declaraciones de política de claves para ver la clave. Seleccione **Editar** para realizar cambios en la política de claves.
19. Elija **Siguiente**.
20. Revise los ajustes de clave que ha elegido. Aún puede volver atrás y cambiar todos los ajustes.
21. Elija **Finalizar** para crear la clave de KMS.

Uso de la AWS KMS API

Puede utilizar la [CreateKey](#) operación para crear AWS KMS keys de todo tipo. Estos ejemplos utilizan la [AWS Command Line Interface \(AWS CLI\)](#). Para ver ejemplos en varios lenguajes de programación, consulte [Úselo CreateKey con un AWS SDK o CLI](#).

 Important

No incluya información confidencial en los campos `Description` o `Tags`. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

La siguiente operación crea una clave de cifrado simétrica en una única región respaldada por material de claves generado por AWS KMS. Esta operación no tiene parámetros obligatorios. Sin embargo, es posible que también desee utilizar el parámetro `Policy` para especificar una política de claves. Puede cambiar la política clave ([PutKeyPolicy](#)) y añadir elementos opcionales, como una [descripción](#) y [etiquetas](#), en cualquier momento. También puede crear [claves asimétricas](#), [claves de varias regiones](#), claves con [material de claves importado](#), y claves en [almacenes de claves personalizados](#). Para crear claves de datos para el cifrado del lado del cliente, utilice la [GenerateDataKey](#) operación.

La `CreateKey` operación no le permite especificar un alias, pero puede utilizarla para crear un alias para la [CreateAlias](#) nueva clave de KMS.

A continuación se muestra un ejemplo de una llamada a la operación `CreateKey` sin parámetros. Este comando utiliza todos los valores predeterminados. Crea una clave KMS de cifrado simétrica para con material de claves generado por 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"
    ],
  },
}
```

Si no especifica una política de claves para su nueva clave KMS, la [política de claves predeterminada](#) que aplica `CreateKey` es diferente de la política de claves predeterminada que aplica la consola cuando se utiliza para crear una nueva clave KMS.

Por ejemplo, esta llamada a la [GetKeyPolicy](#) operación devuelve la política clave que `CreateKey` se aplica. Da Cuenta de AWS acceso a la clave KMS y le permite crear políticas AWS Identity and Access Management (IAM) para la clave KMS. Para obtener información detallada sobre las políticas de IAM y las políticas de claves para claves KMS, consulte [Permisos y acceso a claves 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" : "*"
  } ]
}
```

Creación de una clave de KMS asimétrica

[Puede crear claves KMS asimétricas en la AWS KMS consola, mediante la CreateKeyAPI o mediante la AWS::KMS::Key CloudFormation plantilla.](#) Una clave KMS asimétrica representa un par de claves pública y privada que se puede utilizar para el cifrado, la firma o la obtención de secretos compartidos. La clave privada permanece dentro AWS KMS. Para descargar la clave pública para usarla fuera de ella AWS KMS, consulte [Descarga de la clave pública](#).

Cuando crea una clave KMS asimétrica, debe seleccionar una especificación de clave. A menudo, la especificación de clave que seleccione está determinada por requisitos normativos, de seguridad o empresariales. También puede estar influenciado por el tamaño de los mensajes que necesita cifrar o firmar. En general, las claves de cifrado más largas son más resistentes a los ataques de fuerza bruta. Para obtener una descripción detallada de todas las especificaciones de clave admitidas, consulte [Referencia de especificaciones de clave](#).

AWS los servicios que se integran AWS KMS no admiten claves KMS asimétricas. Si desea crear una clave KMS que cifre los datos que almacena o administra en un AWS servicio, [cree una clave KMS de cifrado simétrico](#).

Para obtener información sobre los permisos necesarios para crear claves de KMS, consulte [Permisos para crear claves KMS](#).

Uso de la consola AWS KMS

Puede utilizar el Consola de administración de AWS para crear claves asimétricas AWS KMS keys (claves KMS). Cada clave KMS asimétrica representa un par de claves públicas y privadas.

Important

No incluya información confidencial en el alias, la descripción ni las etiquetas. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Elija Crear clave.
5. Para crear una clave KMS asimétrica, en Key Type (Tipo de clave), seleccione Asymmetric (Asimétrica).
6. Para crear una clave KMS asimétrica para el cifrado de claves públicas, en Key Usage (Uso de claves), elija Encrypt and decrypt (Cifrar y descifrar).

Para crear una clave KMS asimétrica para firmar mensajes y verificar firmas, en Key Usage (Uso de claves), elija Sign and verify (Firmar y verificar).

Para crear una clave de KMS asimétrica para obtener secretos compartidos, en Uso de claves, elija Acuerdo de claves.

Para obtener ayuda sobre cómo elegir un valor de uso de clave, consulte [Elección del tipo de clave KMS que se va a crear](#).

7. Seleccione una especificación (Key spec [Especificación de clave]) para su clave KMS asimétrica.
8. Elija Siguiente.
9. Escriba un [alias](#) para la clave KMS El nombre del alias no puede empezar por **aws/**. El prefijo **aws/** está reservado para Amazon Web Services y representa las Claves administradas por AWS de su cuenta.

Un alias es un nombre descriptivo que se puede utilizar para identificar la clave KMS en la consola y en algunas otras. AWS KMS APIs Le recomendamos que elija un alias que indique el tipo de datos que piensa proteger o la aplicación que piensa usar con la clave KMS.

Los alias son necesarios para crear una clave KMS en la Consola de administración de AWS. No puede especificar un alias al usar la [CreateKey](#) operación, pero puede usar la consola o la [CreateAlias](#) operación para crear un alias para una clave de KMS existente. Para obtener más información, consulte [Alias en AWS KMS](#).

10. (Opcional) Escriba una descripción de la clave KMS.

Escriba una descripción que explique el tipo de datos que piensa proteger o la aplicación que piensa usar con la clave KMS.

Puede agregar una descripción ahora o actualizarla en cualquier momento, a menos que el [estado de la clave](#) sea Pending Deletion o Pending Replica Deletion. Para añadir, cambiar o eliminar la descripción de una clave gestionada por el cliente existente, edite la descripción en la página de detalles de la clave KMS de la operación Consola de administración de AWS o utilice la [UpdateKeyDescription](#) operación.

11. (Opcional) Escriba una clave de etiqueta y un valor de etiqueta opcional. Para agregar más de una etiqueta a la clave KMS, elija Add tag (Agregar etiqueta).

Al añadir etiquetas a AWS los recursos, AWS genera un informe de asignación de costes con el uso y los costes agregados por etiquetas. Las etiquetas también pueden utilizarse para controlar el acceso a una clave KMS. Para obtener información acerca del etiquetado de claves KMS, consulte [Etiquetas en AWS KMS](#) y [ABAC para AWS KMS](#).

12. Elija Siguiente.

13. Seleccione los usuarios y roles de IAM que pueden administrar la clave de KMS.

 Notas

Esta política clave proporciona el control Cuenta de AWS total de esta clave de KMS. Permite a los administradores de cuentas utilizar las políticas de IAM para dar permiso a otras entidades principales para administrar la clave KMS. Para obtener más información, consulte [the section called “Política de claves predeterminada”](#).

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que

proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;. La AWS KMS consola agrega administradores de claves a la política de claves bajo el identificador de la declaración "Allow access for Key Administrators". La modificación de este identificador de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

14. (Opcional) Para evitar que los usuarios y los roles de IAM seleccionados eliminen esta clave de KMS, en la sección Eliminación de claves situada en la parte inferior de la página, desactive la casilla Permitir que los administradores de claves eliminen esta clave.
15. Elija Siguiente.
16. Seleccione los usuarios y roles de IAM que pueden usar la clave de KMS en [operaciones criptográficas](#).

Notas

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;. La AWS KMS consola agrega los usuarios clave a la política clave bajo los identificadores de la declaración "Allow use of the key" y "Allow attachment of persistent resources". La modificación de estos identificadores de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

17. (Opcional) Puede permitir que otras Cuentas de AWS usen esta clave KMS para operaciones criptográficas. Para ello, en la parte inferior de la página de la sección Otras Cuentas de AWS, elija Agregar otra Cuenta de AWS e ingrese el número de identificación de Cuenta de AWS de una cuenta externa. Para agregar varias cuentas externas, repita este paso.

Note

Para permitir que las entidades principales de las cuentas externas usen la clave KMS, los administradores de la cuenta externa también deben crear las políticas de IAM que

proporcionan estos permisos. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

18. Elija Siguiente.
19. Revise las declaraciones de política de claves para ver la clave. Seleccione Editar para realizar cambios en la política de claves.
20. Elija Siguiente.
21. Revise los ajustes de clave que ha elegido. Aún puede volver atrás y cambiar todos los ajustes.
22. Elija Finalizar para crear la clave de KMS.

Uso de la API AWS KMS

Puede utilizar la [CreateKey](#) operación para crear una asimétrica AWS KMS key. En estos ejemplos, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Al crear una clave KMS asimétrica, debe especificar el parámetro KeySpec, que determina el tipo de claves que cree. Además, debe especificar un valor KeyUsage de ENCRYPT_DECRYPT, SIGN_VERIFY o KEY_AGREEMENT. No puede cambiar estas propiedades después de que se cree la clave de KMS.

La CreateKey operación no le permite especificar un alias, pero puede utilizarla para crear un alias para la [CreateAlias](#) nueva clave de KMS.

Important

No incluya información confidencial en los campos Description o Tags. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

Creación de una clave KMS asimétrica para el cifrado público

En el siguiente ejemplo se utiliza la operación CreateKey para crear una clave KMS asimétrica de claves RSA de 4096 bits diseñadas para el cifrado de claves públicas.

```
$ 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
    }
  }
}

```

Creación de una clave KMS asimétrica para la firma y la verificación

El comando del siguiente ejemplo crea una clave KMS asimétrica que representa un par de claves ECC utilizadas para la firma y verificación. No se puede crear un par de claves de curva elíptica para el cifrado y el descifrado.

```

$ 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"
  }
}

```

Creación de un par de claves KMS asimétricas para obtener secretos compartidos

El comando del siguiente ejemplo crea una clave KMS asimétrica que representa un par de claves ECDH utilizadas para la obtención de secretos compartidos. No se puede crear un par de claves de curva elíptica para el cifrado y el descifrado.

```

$ 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
  }
}

```

Creación de una clave KMS HMAC

[Puede crear claves HMAC KMS en la AWS KMS consola, mediante la CreateKeyAPI o mediante la AWS::KMS::Key CloudFormation plantilla.](#)

Al crear una clave HMAC KMS, debe seleccionar una especificación clave. AWS KMS admite varias [especificaciones clave para las claves HMAC KMS](#). La especificación de clave que seleccione podría estar determinada por requisitos normativos, de seguridad o empresariales. En general, las claves más largas son más resistentes a los ataques de fuerza bruta.

Para obtener información sobre los permisos necesarios para crear claves de KMS, consulte [Permisos para crear claves KMS](#).

Uso de la consola AWS KMS

Puede utilizarla Consola de administración de AWS para crear claves HMAC KMS. Las claves KMS HMAC son claves simétricas con un uso de claves de Generate and verify MAC (Generar y verificar MAC). También puede crear claves HMAC de varias regiones.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Elija Crear clave.
5. En Key type (Tipo de clave), elija Symmetric (Simétrica).

Las claves KMS HMAC son simétricas. Utilice la misma clave para generar y verificar etiquetas HMAC.

6. Para Key usage (Uso de claves), elija Generate and verify MAC (Generar y verificar MAC).

Generar y verificar que MAC es el único uso de clave válido para las claves KMS HMAC.

Note

Key usage (Uso de claves) se muestra para las claves simétricas solo cuando las claves KMS HMAC son compatibles en la región seleccionada.

7. Seleccione una especificación (Key spec [Especificación de clave]) para su clave KMS HMAC.

La especificación de clave que seleccione se puede determinar mediante requisitos normativos, de seguridad o empresariales. En general, las claves más largas son más seguras.

8. Para crear una clave HMAC [de varias regiones](#) primaria, en Advanced options (Opciones avanzadas), elija Multi-Region key (Clave de varias regiones). Las [propiedades compartidas](#) que defina para esta clave KMS, como su tipo de clave y uso de claves, se compartirán con sus claves de réplica.

No puede utilizar este procedimiento para crear una clave de réplica. Para crear una clave HMAC réplica de varias regiones, siga las [instrucciones para crear una clave de réplica](#).

9. Elija Siguiente.
10. Ingrese un [alias](#) para la clave KMS. El nombre del alias no puede empezar por **aws/**. El prefijo **aws/** está reservado para Amazon Web Services y representa las Claves administradas por AWS de su cuenta.

Le recomendamos que utilice un alias que identifique la clave KMS como clave HMAC, como HMAC/test-key. De este modo, le resultará más fácil identificar sus claves HMAC en la AWS KMS consola, donde podrá ordenar y filtrar las claves por etiquetas y alias, pero no por las especificaciones o el uso de las claves.

Los alias son necesarios para crear una clave de KMS en la Consola de administración de AWS. No puede especificar un alias al usar la [CreateKey](#) operación, pero puede usar la consola o la [CreateAlias](#) operación para crear un alias para una clave de KMS existente. Para obtener más información, consulte [Alias en AWS KMS](#).

11. (Opcional) Ingrese una descripción de la clave KMS.

Escriba una descripción que explique el tipo de datos que piensa proteger o la aplicación que piensa usar con la clave de KMS.

Puede agregar una descripción ahora o actualizarla en cualquier momento, a menos que el [estado de la clave](#) sea Pending Deletion o Pending Replica Deletion. Para añadir, cambiar o eliminar la descripción de una clave gestionada por el cliente existente, edite la descripción en la página de detalles de la clave KMS Consola de administración de AWS en la operación Consola de administración de AWS o utilice la [UpdateKeyDescription](#) operación.

12. (Opcional) Ingrese una clave de etiqueta y un value de etiqueta opcional. Para agregar más de una etiqueta a la clave de KMS, elija Agregar etiqueta.

Considere agregar una etiqueta que identifique la clave como una clave HMAC, por ejemplo Type=HMAC. Esto le permitirá identificar más fácilmente sus claves HMAC en la AWS KMS consola, donde podrá ordenar y filtrar las claves por etiquetas y alias, pero no por especificaciones o uso de las claves.

Al añadir etiquetas a AWS los recursos, AWS genera un informe de asignación de costes con el uso y los costes agregados por etiquetas. Las etiquetas también pueden utilizarse para controlar el acceso a una clave KMS. Para obtener información acerca del etiquetado de claves KMS, consulte [Etiquetas en AWS KMS](#) y [ABAC para AWS KMS](#).

13. Elija Siguiente.

14. Seleccione los usuarios y roles de IAM que pueden administrar la clave de KMS.

Notas

Esta política clave proporciona el control Cuenta de AWS total de esta clave de KMS. Permite a los administradores de cuentas utilizar las políticas de IAM para dar permiso a otras entidades principales para administrar la clave KMS. Para obtener más información, consulte [the section called “Política de claves predeterminada”](#).

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;. La AWS KMS consola agrega administradores de claves a la política de claves bajo el identificador de la declaración "Allow access for Key Administrators". La modificación de este identificador de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

15. (Opcional) Para evitar que los usuarios y los roles de IAM seleccionados eliminen esta clave de KMS, en la sección Eliminación de claves situada en la parte inferior de la página, desactive la casilla Permitir que los administradores de claves eliminen esta clave.

16. Elija Siguiente.

17. Seleccione los usuarios y roles de IAM que pueden usar la clave de KMS en [operaciones criptográficas](#).

Notas

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;.

La AWS KMS consola agrega los usuarios clave a la política clave bajo los identificadores de la declaración "Allow use of the key" y "Allow attachment of persistent resources". La modificación de estos identificadores de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

18. (Opcional) Puede permitir que otras Cuentas de AWS usen esta clave KMS para operaciones criptográficas. Para ello, en la parte inferior de la página de la sección Otras Cuentas de AWS, elija Agregar otra Cuenta de AWS e ingrese el número de identificación de Cuenta de AWS de una cuenta externa. Para agregar varias cuentas externas, repita este paso.

 Note

Para permitir que las entidades principales de las cuentas externas usen la clave KMS, los administradores de la cuenta externa también deben crear las políticas de IAM que proporcionan estos permisos. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

19. Elija Siguiente.
20. Revise las declaraciones de política de claves para ver la clave. Seleccione Editar para realizar cambios en la política de claves.
21. Elija Siguiente.
22. Revise los ajustes de clave que ha elegido. Aún puede volver atrás y cambiar todos los ajustes.
23. Elija Finish (Finalizar) para crear la clave KMS HMAC.

Uso de la API AWS KMS

Puede utilizar la [CreateKey](#) operación para crear una clave HMAC KMS. En estos ejemplos, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Al crear una clave KMS HMAC, debe especificar el parámetro KeySpec, que determina el tipo de clave KMS. Además, debe especificar un valor KeyUsage de GENERATE_VERIFY_MAC, aunque sea el único valor de uso de claves válido para claves HMAC. Para crear una clave KMS HMAC de [múltiples regiones](#), agregue el parámetro MultiRegion con un valor de true. No puede cambiar estas propiedades después de que se cree la clave de KMS.

La `CreateKey` operación no le permite especificar un alias, pero puede utilizarla para crear un alias para la [CreateAlias](#) nueva clave KMS. Le recomendamos que utilice un alias que identifique la clave KMS como clave HMAC, como `HMAC/test-key`. Esto le permitirá identificar sus claves HMAC con más facilidad en la AWS KMS consola, donde podrá ordenar y filtrar las claves por alias, pero no por especificación o uso de claves.

Si intenta crear una clave HMAC KMS Región de AWS en una que no sea compatible con las claves HMAC, la operación devuelve un `CreateKey UnsupportedOperationException`

El siguiente ejemplo utiliza la operación `CreateKey` para crear una clave KMS 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
  }
}
```

Creación de claves primarias de varias regiones

Puede crear una [clave principal multirregional](#) en la AWS KMS consola o mediante la AWS KMS API. Puede crear la clave principal en cualquier Región de AWS lugar que AWS KMS admita claves multirregionales.

Para crear una clave principal multirregional, el director necesita los [mismos permisos](#) que necesita para crear cualquier clave de KMS, incluido el CreateKey permiso [kms:](#) en una política de IAM. El director también necesita el permiso [iam:: CreateServiceLinkedRole](#). Puede usar la clave de MultiRegionKeyType condición [kms:](#) para permitir o denegar el permiso para crear claves principales multirregionales.

Note

Al crear su clave principal de varias regiones, considere detenidamente los usuarios y roles de IAM que seleccione para administrar y usar la clave. Las políticas de IAM pueden otorgar permisos a otros usuarios y roles de IAM para que administren la clave KMS. Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;.

Uso de la consola AWS KMS

Para crear una clave principal multirregional en la AWS KMS consola, utilice el mismo proceso que utilizaría para crear cualquier clave de KMS. Seleccione una clave de varias regiones en Advanced options (Opciones avanzadas). Para obtener instrucciones completas, consulte [Crear una clave de KMS..](#)

Important

No incluya información confidencial en el alias, la descripción ni las etiquetas. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Elija Crear clave.

5. Seleccione un tipo de clave [simétrica o asimétrica](#). Las claves simétricas son las predeterminadas.

Puede crear claves simétricas y asimétricas de varias regiones, incluidas las claves de KMS HMAC de varias regiones, que son simétricas.

6. Seleccione el uso de claves. Encrypt and decrypt (Cifrar y descifrar) es el valor predeterminado.

Para obtener ayuda, consulte [Crear una clave de KMS](#), [the section called “Creación de una clave de KMS asimétrica”](#) o [the section called “Creación de una clave KMS HMAC”](#).

7. Expanda Advanced options (Opciones avanzadas).
8. En Origen del material clave, para AWS KMS generar el material clave que compartirán la clave principal y la de réplica, elija KMS. Si [importa material de claves](#) en las claves principal y de réplica, elija External (Import key material) Externo (material de claves importado).
9. En Regionalidad, seleccione Clave de varias regiones.

No puede cambiar esta configuración después de crear la clave KMS.

10. Escriba un [alias](#) para la clave principal.

Los alias no son una propiedad compartida de claves de varias regiones. Puede asignar a la clave principal multirregional y a sus réplicas el mismo alias o alias diferentes. AWS KMS no sincroniza los alias de las claves multirregionales.

 Note

Agregar, eliminar o actualizar un alias puede permitir o denegar el permiso a la clave KMS. Para más detalles, consulte [ABAC para AWS KMS](#) y [Uso de alias para controlar el acceso a las claves KMS](#).

11. (Opcional) Escriba una descripción de la clave primaria.

Las descripciones no son una propiedad compartida de las claves de varias regiones. Puede dar a la clave principal multirregional y a sus réplicas la misma descripción o descripciones diferentes. AWS KMS no sincroniza las descripciones clave de las claves multirregionales.

12. (Opcional) Escriba una clave de etiqueta y un valor de etiqueta opcional. Para asignar más de una etiqueta a la clave principal, elija Add tag (Agregar etiqueta).

Las etiquetas no son una propiedad compartida de las claves de varias regiones. Puede asignar a su clave principal de varias regiones y sus réplicas las mismas etiquetas o etiquetas diferentes.

AWS KMS no sincroniza las etiquetas de las claves de varias regiones. Puede cambiar las etiquetas de las claves KMS en cualquier momento.

 Note

Etiquetar o quitar las etiquetas de la clave KMS puede permitir o denegar permiso a la clave KMS. Para más detalles, consulte [ABAC para AWS KMS](#) y [Uso de etiquetas para controlar el acceso a las claves KMS](#).

13. Seleccione los usuarios y roles de IAM que pueden administrar la clave principal.

 Notas

- Este paso inicia el proceso de creación de una [política de claves](#) para la clave principal. Las políticas de clave no son una propiedad compartida de las claves de varias regiones. Puede asignar a la clave principal multirregional y a sus réplicas la misma política clave o políticas clave diferentes. AWS KMS no sincroniza las políticas clave de las claves multirregionales. Puede cambiar la política de claves de una clave KMS en cualquier momento.
- Al crear una clave principal de varias regiones, considere la posibilidad de utilizar la [política de claves predeterminada](#) generada por la consola. Si se modifica esta política, la consola no proporcionará los pasos necesarios para seleccionar los administradores y usuarios clave al crear las réplicas de claves, ni agregará las declaraciones de política correspondientes. Como resultado, tendrá que agregarlos de manera manual.
- La AWS KMS consola agrega administradores clave a la política clave bajo el identificador de la declaración. "Allow access for Key Administrators" La modificación de este identificador de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

14. (Opcional) Para evitar que los usuarios y los roles de IAM seleccionados eliminen esta clave de KMS, en la sección Eliminación de claves situada en la parte inferior de la página, desactive la casilla Permitir que los administradores de claves eliminen esta clave.

15. Elija Siguiente.

16. Seleccione los usuarios y roles de IAM que pueden usar la clave de KMS en [operaciones criptográficas](#).

Notas

La AWS KMS consola agrega los usuarios clave a la política clave bajo los identificadores de la declaración "Allow use of the key" y "Allow attachment of persistent resources". La modificación de estos identificadores de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

17. (Opcional) Puede permitir que otras Cuentas de AWS usen esta clave KMS para operaciones criptográficas. Para ello, en la parte inferior de la página de la sección Otras Cuentas de AWS, elija Agregar otra Cuenta de AWS e ingrese el número de identificación de Cuenta de AWS de una cuenta externa. Para agregar varias cuentas externas, repita este paso.

Note

Para permitir que las entidades principales de las cuentas externas usen la clave KMS, los administradores de la cuenta externa también deben crear las políticas de IAM que proporcionan estos permisos. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

18. Elija Siguiente.
19. Revise las declaraciones de política de claves para ver la clave. Seleccione Editar para realizar cambios en la política de claves.
20. Elija Siguiente.
21. Revise los ajustes de clave que ha elegido. Aún puede volver atrás y cambiar todos los ajustes.
22. Seleccione Finalizar para crear la clave principal de varias regiones.

Uso de la API AWS KMS

Para crear una clave principal multirregional, utilice la [CreateKey](#) operación. Utilice el parámetro `MultiRegion` con un valor de `True`.

Por ejemplo, el siguiente comando crea una clave principal multirregión en la persona que llama (Región de AWS us-east-1). Acepta valores predeterminados para todas las demás propiedades, incluida la política de claves. Los valores predeterminados para las claves principales de varias regiones son los mismos que los valores predeterminados para todas las demás claves KMS,

incluida la [política de claves predeterminada](#). Este procedimiento crea una clave de cifrado simétrica, la clave KMS predeterminada.

La respuesta incluye el elemento `MultiRegion` y el elemento `MultiRegionConfiguration` con subelementos y valores típicos para una clave principal de varias regiones sin claves de réplica. La [ID de clave](#) de una clave de varias regiones siempre comienza con `mrk-`.

 Important

No incluya información confidencial en los campos `Description` o `Tags`. Estos campos pueden aparecer en texto plano en los CloudTrail registros y otros resultados.

```
$ 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": [ ]
    }
  }
}
```

```
}  
}
```

Creación de claves de réplica de varias regiones

Puede crear una clave de réplica multirregional en la AWS KMS consola mediante la [ReplicateKey](#) operación o mediante una `AWS::KMS::ReplicaKey` CloudFormation plantilla. No puede utilizar la [CreateKey](#) operación para crear una clave de réplica.

Puede utilizar estos procedimientos para replicar cualquier clave principal de varias regiones, incluida una [clave simétrica de KMS de cifrado](#), una [clave asimétrica de KMS](#) o una [clave KMS HMAC](#).

Cuando se completa esta operación, la nueva clave de réplica tiene un valor transitorio de [estado clave](#) de `Creating`. Este estado de clave cambia a `Enabled` (o `PendingImport` si se crea una clave de varias regiones con [material de claves importado](#)) después de unos segundos cuando se haya completado el proceso de creación de la nueva clave de réplica. Mientras el estado de la clave es `Creating`, puede administrar las claves, pero aún no puede usarlas en operaciones criptográficas. Si va a crear y utilizar la clave de réplica mediante programación, vuelva a intentarlo `KMSInvalidStateException` o llame [DescribeKey](#) para comprobar su `KeyState` valor antes de utilizarla.

En caso de que elimine por error una clave de réplica, puede utilizar este procedimiento para volver a crearla. En caso de que replique la misma clave primaria en la misma región, la nueva clave de réplica que cree tendrá las mismas [propiedades compartidas](#) que la clave de réplica original.

Important

No incluya información confidencial en el alias, la descripción ni las etiquetas. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

Para usar una AWS CloudFormation plantilla para crear una clave de réplica, consulte [AWS::KMS::ReplicaKey](#) la Guía del AWS CloudFormation usuario.

Paso 1: Elegir Regiones de réplica

Por lo general, se opta por replicar una clave multirregional en una en Región de AWS función de su modelo de negocio y de los requisitos normativos. Por ejemplo, puede replicar una clave en

Regiones donde guarda sus recursos. O bien, para cumplir con un requisito de recuperación de desastres, puede replicar una clave en regiones geográficamente distantes.

Los siguientes son los AWS KMS requisitos para las réplicas de regiones. Si la región que eliges no cumple con estos requisitos, se producirá un error en los intentos de replicar una clave.

- Una clave de varias regiones relacionada por región: no se puede crear una clave de réplica en la misma Región que su clave principal, o en la misma Región que otra réplica de la clave principal.

En caso de que intente replicar una clave primaria en una región que ya tiene una réplica de esa clave primaria, el intento producirá un error. En caso de que la clave de réplica actual de la región se encuentre en el [estado de clave PendingDeletion](#), puede [cancelar la eliminación de la clave de réplica](#) o esperar hasta que se elimine.

- Múltiples claves de varias regiones no relacionadas en la misma región: puede tener varias claves de varias regiones no relacionadas en la misma región. Por ejemplo, puede tener dos claves principales de varias regiones en la región us-east-1. Cada una de las claves principales puede tener una clave de réplica en la región us-west-2.
- Regiones en la misma partición: la región de la clave de réplica debe estar en la misma [partición de AWS](#) que la región de claves principal.
- La región debe estar habilitada: si una región está [deshabilitada de forma predeterminada](#), no puede crear ningún recurso en esa región hasta que esté habilitado para su Cuenta de AWS.

Paso 2: Crear claves de réplica

Note

Al crear claves de réplica, considere detenidamente los usuarios y roles de IAM que seleccione para administrar y usar la clave de réplica. Las políticas de IAM pueden otorgar permisos a otros usuarios y roles de IAM para que administren la clave KMS.

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;.

Uso de la AWS KMS consola

En la AWS KMS consola, puede crear una o varias réplicas de una clave principal multirregional en la misma operación.

Este procedimiento es similar a la creación de una clave KMS de una sola región estándar en la consola. Sin embargo, dado que una clave de réplica se basa en la clave principal, no se seleccionan valores para [propiedades compartidas](#), como la especificación de la clave (simétrica o asimétrica), el uso de la clave u origen de la clave.

Se especifican propiedades que no se comparten, como un alias, etiquetas, una descripción y una política de clave. Como conveniencia, la consola muestra los valores de propiedad actuales de la clave principal, pero puede cambiarlos. Aunque mantenga los valores de la clave principal, AWS KMS no los mantiene sincronizados.

Important

No incluya información confidencial en el alias, la descripción ni las etiquetas. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Seleccione el alias o el ID de clave de una [clave principal de varias regiones](#). Se abrirá la página de detalles de clave de la clave KMS.

Para identificar una clave principal de varias regiones, utilice el icono de herramienta situado en la esquina superior derecha para agregar la columna Regionality (Regionalidad) de la tabla.

5. Elija la pestaña Regionality (Regionalidad).
6. En la sección Related multi-Region keys (Claves de varias regiones relacionadas), elija Create new replica keys (Crear nuevas claves de réplica).

La sección Related multi-Region keys (Claves de varias regiones relacionadas) muestra la región de la clave principal y sus claves de réplica. Puede utilizar esta pantalla para ayudarle a elegir la región para su nueva clave de réplica.

7. Seleccione una o más Regiones de AWS. Este procedimiento crea una clave de réplica en cada una de las regiones seleccionadas.

El menú incluye solo las regiones de la misma AWS partición que la clave principal. Las regiones que ya tienen una clave de varias regiones relacionadas se muestran, pero no se pueden seleccionar. Es posible que no tenga permiso para replicar una clave en todas las regiones del menú.

Cuando haya terminado de elegir Regiones, cierre el menú. Aparecerán las regiones elegidas. Para cancelar la replicación en una región, seleccione la X que está junto al nombre de la región.

8. Escriba un [alias](#) para la clave de réplica.

La consola muestra uno de los alias actuales de la clave principal, pero puede cambiarlo. Puede asignar a su clave principal de varias regiones y sus réplicas el mismo alias o a alias diferentes. Los alias no son una [propiedad compartida de las claves](#) multirregionales. AWS KMS no sincroniza los alias de las claves multirregionales.

Agregar, eliminar o actualizar un alias puede permitir o denegar el permiso a la clave KMS. Para más detalles, consulte [ABAC para AWS KMS](#) y [Uso de alias para controlar el acceso a las claves KMS](#).

9. (Opcional) Escriba una descripción de la clave de réplica.

La consola muestra la descripción actual de la clave principal, pero puede cambiarla. Las descripciones no son una propiedad compartida de las claves de varias regiones. Puede dar a la clave principal multirregional y a sus réplicas la misma descripción o descripciones diferentes. AWS KMS no sincroniza las descripciones clave de las claves multirregionales.

10. (Opcional) Escriba una clave de etiqueta y un valor de etiqueta opcional. Para asignar más de una etiqueta a la clave de réplica, elija Add tag (Agregar etiqueta).

La consola muestra las etiquetas actualmente conectadas a la clave principal, pero puede cambiarlas. Las etiquetas no son una propiedad compartida de las claves de varias regiones. Puede asignar a la clave principal multirregional y a sus réplicas las mismas etiquetas o etiquetas diferentes. AWS KMS no sincroniza las etiquetas de las claves multirregionales.

Etiquetar o quitar las etiquetas de la clave KMS puede permitir o denegar permiso a la clave KMS. Para más detalles, consulte [ABAC para AWS KMS](#) y [Uso de etiquetas para controlar el acceso a las claves KMS](#).

11. Seleccione los usuarios y roles de IAM que pueden administrar la clave de réplica.

 Notas

- Si modificó la política de claves predeterminada al crear la clave principal de varias regiones, la consola no le presentará una petición para que seleccione administradores o usuarios de claves (pasos 11 a 15) durante la creación de la clave de réplica. En este caso, se deberá agregar manualmente los permisos necesarios para los administradores y usuarios de claves a la política de claves. Para ello, seleccione Editar en el paso Editar la política de claves (paso 17).
- Este paso comienza el proceso de creación de una [política de claves](#) para la clave de réplica. La consola muestra la política de clave actual de la clave principal, pero puede cambiarla. Las políticas de clave no son una propiedad compartida de las claves de varias regiones. Puede asignar a su clave principal de varias regiones y sus réplicas la misma política de claves o políticas de claves diferentes. AWS KMS no sincroniza las políticas de claves. Puede cambiar la política de claves de cualquier clave KMS en cualquier momento.
- La AWS KMS consola agrega administradores de claves a la política de claves bajo el identificador de la declaración. "Allow access for Key Administrators" La modificación de este identificador de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

12. (Opcional) Para evitar que los usuarios y los roles de IAM seleccionados eliminen esta clave de KMS, en la sección Eliminación de claves situada en la parte inferior de la página, desactive la casilla Permitir que los administradores de claves eliminen esta clave.
13. Elija Siguiente.
14. Seleccione los usuarios y roles de IAM que pueden usar la clave de KMS en [operaciones criptográficas](#).

 Nota

La AWS KMS consola agrega los usuarios clave a la política clave bajo los identificadores de la declaración "Allow use of the key" y "Allow attachment of persistent resources". La modificación de estos identificadores de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

15. (Opcional) Puede permitir que otras Cuentas de AWS usen esta clave KMS para operaciones criptográficas. Para ello, en la parte inferior de la página de la sección Otras Cuentas de AWS, elija Agregar otra Cuenta de AWS e ingrese el número de identificación de Cuenta de AWS de una cuenta externa. Para agregar varias cuentas externas, repita este paso.

 Note

Para permitir que las entidades principales de las cuentas externas usen la clave KMS, los administradores de la cuenta externa también deben crear las políticas de IAM que proporcionan estos permisos. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

16. Elija Siguiente.
17. Revise las declaraciones de política de claves para ver la clave. Seleccione Editar para realizar cambios en la política de claves.
18. Elija Siguiente.
19. Revise los ajustes de clave que ha elegido. Aún puede volver atrás y cambiar todos los ajustes.
20. Seleccione Finalizar para crear la clave de réplica de varias regiones.

Uso de la API AWS KMS

Para crear una clave de réplica multirregional, utilice la [ReplicateKey](#) operación. No puede utilizar la [CreateKey](#) operación para crear una clave de réplica. Esta operación crea las claves de réplica de una en una. La región que especifique debe cumplir con los [Requisitos de región](#) para las claves de réplica.

Cuando utiliza la operación `ReplicateKey`, no especifique valores para ninguna [propiedad compartida](#) de claves de varias regiones. Los valores de propiedad compartida se copian de la clave principal y se mantienen sincronizados. Sin embargo, puede especificar valores para las propiedades que no se comparten. De lo contrario, AWS KMS aplica los valores predeterminados estándar para las claves de KMS, no los valores de la clave principal.

 Note

Si no especifica valores para los Tags parámetros `DescriptionKeyPolicy`, o AWS KMS crea la clave de réplica con una descripción de cadena vacía, la [política de claves predeterminada](#) y sin etiquetas.

No incluya información confidencial en los campos `Description` o `Tags`. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

Por ejemplo, el comando siguiente crea una clave de réplica de varias regiones en la región Asia Pacífico (Sdney) (ap-southeast-2). Esta clave de réplica está modelada en la clave principal de la región EE. UU. Este (Norte de Virginia) (us-east-1), que se identifica mediante el valor del parámetro `KeyId`. En este ejemplo se aceptan valores predeterminados para todas las demás propiedades, incluida la política de claves.

La respuesta describe la nueva clave de réplica. Incluye campos para propiedades compartidas, como el `KeyId`, `KeySpec`, `KeyUsage` y el origen del material clave (`Origin`). También incluye propiedades que son independientes de la clave principal, como la `Description`, la política de claves (`ReplicaKeyPolicy`), y las etiquetas (`ReplicaTags`).

La respuesta también incluye el ARN clave y la región de la clave principal y todas sus claves de réplica, incluida la que se acaba de crear en la región ap-southeast-2. En este ejemplo, el elemento `ReplicaKey` muestra que esta clave principal ya se replicó en la región 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\": []
}

```

Creación de una clave KMS con material de claves importado

El material clave importado le permite proteger sus AWS recursos con las claves criptográficas que genere. En la siguiente información general se explica cómo importar el material de claves en AWS KMS. Para obtener más información sobre cada paso del proceso, consulte los temas correspondiente.

1. [Cree una clave de KMS sin material de claves](#): el origen debe ser EXTERNAL. Un origen de clave de EXTERNAL indica que la clave está diseñada para material clave importado e AWS KMS impide que se genere material clave para la clave KMS. En un paso posterior importará su propio material de claves en esta clave KMS.

El material clave que importe debe ser compatible con la especificación clave de la AWS KMS clave asociada. Para obtener más información sobre la compatibilidad, consulte [the section called “Requisitos para el material de claves importado”](#).

2. [Descargar la clave pública de encapsulamiento y el token de importación](#): después de completar el paso 1, descargue una clave pública de encapsulamiento y un token de importación. Estos elementos protegen el material clave mientras se importa a AWS KMS.

En este paso, elige el tipo (“especificación de clave”) de la clave de encapsulamiento RSA y el algoritmo de encapsulamiento que utilizará para cifrar los datos en tránsito en AWS KMS. Puede elegir una especificación de clave de encapsulamiento y un algoritmo de clave de encapsulamiento diferentes cada vez que importe o vuelva a importar el mismo material de claves.

3. [Cifrar el material de claves](#): utilice la clave pública de encapsulamiento que ha descargado en el paso 2 para cifrar el material de claves que ha creado en su propio sistema.
4. [Importar el material de claves](#): cargue el material de claves cifrado que ha creado en el paso 3 y el token de importación que ha descargado en el paso 2.

En esta etapa, puede [establecer una fecha de vencimiento opcional](#). Cuando el material clave importado caduca, lo AWS KMS elimina y la clave KMS queda inutilizable. Para seguir usando la clave de KMS, debe volver a importar el mismo material de claves.

Cuando la operación de importación se completa de manera correcta, el estado de la clave KMS cambia de `PendingImport` a `Enabled`. Ahora puede usar la clave KMS en operaciones criptográficas.

AWS KMS registra una entrada en el AWS CloudTrail registro al [crear la clave KMS, descargar la clave pública empaquetadora y el token de importación e importar el material clave](#). AWS KMS también registra una entrada cuando se elimina el material clave importado o cuando se AWS KMS [elimina el material clave caducado](#).

Permisos para importar material de clave

Para crear y administrar claves KMS con material de claves importado, el usuario necesita permiso para las operaciones de este proceso. Puede proporcionar los permisos `kms:GetParametersForImport`, `kms:ImportKeyMaterial`, y `kms:DeleteImportedKeyMaterial` en la política de claves al crear la clave KMS. En la AWS

KMS consola, estos permisos se añaden automáticamente a los administradores de claves al crear una clave con un origen de material clave externo.

Para crear claves KMS con material de claves importado, la entidad principal necesita los siguientes permisos.

- [kms: CreateKey](#) (política de IAM)
 - Para limitar este permiso a las claves de KMS con material clave importado, utilice la condición [kms: KeyOrigin](#) policy con un valor de EXTERNAL.

```
{
  "Sid": "CreateKMSKeysWithoutKeyMaterial",
  "Effect": "Allow",
  "Resource": "*",
  "Action": "kms:CreateKey",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL"
    }
  }
}
```

- [kms: GetParametersForImport](#) (política clave o política de IAM)
 - Para limitar este permiso a las solicitudes que utilizan un algoritmo de empaquetado y una especificación clave de empaquetado determinados, usa las condiciones de la WrappingKeySpec política [kms: WrappingAlgorithm](#) y [kms:](#).
- [kms: ImportKeyMaterial](#) (política clave o política de IAM)
 - Para permitir o prohibir que el material clave caduque y controlar la fecha de caducidad, utilice las condiciones de la ValidTo política [kms: ExpirationModel](#) y [kms:](#).

Para volver a importar el material clave importado, el director necesita los ImportKeyMaterial permisos [kms: GetParametersForImport](#) y [kms:](#).

Para eliminar el material clave importado, el director necesita el DeleteImportedKeyMaterial permiso [kms:](#).

Por ejemplo, para dar permiso KMSAdminRole al ejemplo y que pueda administrar todos los aspectos de una clave KMS con material de claves importado, incluya una declaración de política clave como la siguiente en la política clave de la clave 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 el material de claves importado

El material de claves que importa debe ser compatible con la [especificación de clave](#) de la clave de KMS asociada. Para los pares de claves asimétricas, importe solo la clave privada del par. AWS KMS deriva la clave pública de la clave privada.

AWS KMS admite las siguientes especificaciones clave para las claves de KMS con material de claves importado.

- Claves de cifrado simétricas
 - Especificación de la clave:
 - SYMMETRIC_DEFAULT.
 - Requisitos:
 - 256 bits (32 bytes) de datos binarios
 - En las regiones de China, debe ser un dato binario de 128 bits (16 bytes).
- Claves HMAC
 - Especificaciones de la clave:
 - HMAC_224
 - HMAC_256
 - HMAC_384
 - HMAC_512
 - Requisitos:
 - El material de claves HMAC debe cumplir con la [RFC 2104](#).

- La longitud de la clave debe coincidir, como mínimo, con la longitud especificada en la especificación de la clave. La longitud máxima de la clave es de 1024 bits.
- Si el material clave supera los 1024 bits, puede utilizar el hash del material clave e importar la salida del hash. El algoritmo de hash debe coincidir con la especificación de la clave de la clave de KMS de HMAC que está creando.
- Ejemplo:
 - Para importar 2048 bits de material de claves a una clave HMAC_256, primero se calcula el hash SHA-256 del material de claves de 2048 bits y, a continuación, se importa el resultado del hash de 256 bits a la clave de KMS.
- Longitudes de clave válidas:
 - HMAC_224: 224 a 1024 bits
 - HMAC_256: 256 a 1024 bits
 - HMAC_384: 384 a 1024 bits
 - HMAC_512: 512 a 1024 bits
- Clave privada asimétrica RSA
 - Especificaciones de la clave:
 - RSA_2048
 - RSA_3072
 - RSA_4096
 - Requisitos:
 - La clave privada asimétrica RSA que importe debe formar parte de un par de claves que cumpla con la [RFC 3447](#).
 - Módulo: 2048 bits, 3072 bits o 4096 bits
 - Número de números primos: 2 (no se admiten claves RSA de varios números primos)
 - El material de claves asimétricas debe estar codificado en BER o DER en el formato #8 de los estándares de criptografía de clave pública (PKCS) que cumple con la [RFC 5208](#).
- Clave privada asimétrica de curva elíptica
 - Especificaciones de la clave:
 - ECC_NIST_P256 (secp256r1)
 - ECC_NIST_P384 (secp384r1)
 - ECC_NIST_P521 (secp521r1)

- ECC_SECG_P256K1 (secp256k1)
- EDWARDS25519 ECC_NIST_ (ed25519)
- Requisitos:
 - La clave privada asimétrica ECC que importe debe formar parte de un par de claves que cumpla con la [RFC 5915](#).
 - Curva: NIST P-256, NIST P-384, NIST P-521, SecP256k1, NIST Ed25519.
 - Parámetros: solo curvas con nombre (se rechazan las claves de ECC con parámetros explícitos)
 - Coordenadas de puntos públicos: pueden estar comprimidas, descomprimidas o ser proyectivas
 - El material de claves asimétricas debe estar codificado en BER o DER en el formato #8 de los estándares de criptografía de clave pública (PKCS) que cumple con la [RFC 5208](#).
- Clave de ML-DSA
 - Especificaciones de la clave:
 - ML_DSA_44
 - ML_DSA_65
 - ML_DSA_87

 Important

No se admite la importación de claves de ML-DSA.

- SM2 clave privada asimétrica (solo regiones de China)
- Requisitos:
 - La clave privada SM2 asimétrica que importe debe formar parte de un key pair que cumpla con 0003. GM/T
 - Curva: SM2
 - Parámetros: solo curva con nombre (se rechazan SM2 las claves con parámetros explícitos).
 - Coordenadas de puntos públicos: pueden estar comprimidas, descomprimidas o ser proyectivas
 - El material de claves asimétricas debe estar codificado en BER o DER en el formato #8 de los estándares de criptografía de clave pública (PKCS) que cumple con la [RFC 5208](#).

Paso 1: Crea un material AWS KMS key sin clave

De forma predeterminada, AWS KMS crea material clave automáticamente al crear una clave de KMS. Para importar su propio material de claves, comience con la creación de una clave de KMS sin material de claves. Después, importe el material de claves. Para crear una clave KMS sin material clave, utilice la AWS KMS consola o la [CreateKey](#) operación.

Para crear una clave sin material de claves, especifique un [origen](#) de EXTERNAL. La propiedad de origen de una clave de KMS es inmutable. Una vez creada, no podrá convertir una clave de KMS diseñada para material de claves importado en una clave de KMS con material de clave de origen AWS KMS o de cualquier otra fuente.

El [estado de claves](#) de una clave de KMS con un origen EXTERNAL y ningún material de claves es PendingImport. Una clave de KMS puede permanecer en estado PendingImport indefinidamente. Sin embargo, no puede utilizar una clave de KMS en estado PendingImport en operaciones criptográficas. Cuando importa material de claves, el estado de la clave de KMS cambia a Enabled y puede usar la clave de KMS en operaciones criptográficas.

AWS KMS registra un evento en el AWS CloudTrail registro al [crear la clave KMS, descargar la clave pública y el token de importación e importar el material de la clave](#). AWS KMS también registra un CloudTrail evento cuando se [elimina el material clave importado](#) o cuando se AWS KMS [elimina el material clave caducado](#).

Temas

- [Crear una clave KMS sin material de claves \(consola\)](#)
- [Crear una clave de KMS sin material clave \(AWS KMS API\)](#)

Crear una clave KMS sin material de claves (consola)

Solo tiene que crear una clave de KMS para el material de claves importado una vez. Puede importar y volver a importar el mismo material de claves en la clave de KMS existente siempre que lo necesite, pero no puede importar material de claves diferente en una clave de KMS. Para obtener más información, consulte [Paso 2: descargar la clave pública de encapsulamiento y el token de importación](#).

Para encontrar las claves de KMS existentes con material de claves importado en la tabla de claves gestionadas por el cliente, utilice el icono con forma de engranaje situado en la esquina superior

derecha para mostrar la columna Origen de la lista de claves de KMS. Las claves importadas tienen el valor Origen de Externo (Importar material de claves).

Para crear una clave KMS con material de claves importado, siga las [instrucciones para crear una clave KMS del tipo de clave que prefiera](#), con la siguiente excepción.

Después de elegir el uso de la clave, haga lo siguiente:

1. Expanda Advanced options (Opciones avanzadas).
2. En Origen del material de claves, elija Externo (Material de claves importado).
3. Elija la casilla de verificación situada junto a Entiendo las implicaciones de seguridad y durabilidad del uso de una clave importada para indicarnos que entiende las implicaciones de utilizar material de claves importado. Para leer más información acerca de estas implicaciones, consulte [Protección del material de claves importado](#).
4. Opcional: para crear una [clave de KMS de varias regiones](#) con material de claves importado, en Regionalidad, seleccione Clave de varias regiones.
5. Vuelva a las instrucciones básicas. Los pasos restantes del procedimiento básico son los mismos para todas las claves de KMS de ese tipo.

Al elegir Finalizar, habrá creado una clave de KMS sin material de claves y con un estado ([estado de la clave](#)) de Pendiente de importación.

Sin embargo, en lugar de volver a la tabla de claves administradas por el cliente, la consola muestra una página en la que puede descargar la clave pública y el token de importación que necesita para importar el material de claves. Puede continuar con el paso de descarga ahora o seleccionar Cancelar para detenerse en este momento. Puede volver a este paso de descarga en cualquier momento.

Siguiente: [Paso 2: descargar la clave pública de encapsulamiento y el token de importación](#).

Crear una clave de KMS sin material clave (AWS KMS API)

Para usar la [AWS KMS API](#) para crear una clave KMS de cifrado simétrico sin material clave, envíe una [CreateKey](#) solicitud con el Origin parámetro establecido en. EXTERNAL El siguiente ejemplo muestra cómo hacerlo con la [AWS Command Line Interface \(AWS CLI\)](#).

```
$ aws kms create-key --origin EXTERNAL
```

Si el comando se ejecuta correctamente, verá un resultado parecido al siguiente. La AWS KMS clave Origin es EXTERNAL y KeyState es PendingImport.

i Tip

Si el comando no se ejecuta correctamente, es posible que aparezca `KMSInvalidStateException` o `NotFoundException`. Puede reintentar la solicitud.

```
{
  "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 el valor KeyId del resultado del comando para usarlo en pasos posteriores y, a continuación, vaya a [Paso 2: descargar la clave pública de encapsulamiento y el token de importación](#).

i Note

Este comando crea una clave de KMS de cifrado simétrico con una KeySpec de SYMMETRIC_DEFAULT y un KeyUsage de ENCRYPT_DECRYPT. Puede usar los parámetros opcionales `--key-spec` y `--key-usage` crear una clave de KMS asimétrica o HMAC. Para obtener más información, consulte la operación [CreateKey](#).

Paso 2: descargar la clave pública de encapsulamiento y el token de importación

Tras [crear un material AWS KMS key sin clave](#), descarga una clave pública empaquetadora y un token de importación para esa clave de KMS mediante la AWS KMS consola o la [GetParametersForImportAPI](#). La clave pública de encapsulamiento y el token de importación son un conjunto indivisible que deben usarse juntos.

Utilizará la clave pública de encapsulamiento para [cifrar el material de claves](#) para su transferencia. [Antes de descargar un par de claves de empaquetado RSA, seleccione la longitud \(especificación de clave\) del par de claves de empaquetado RSA y el algoritmo de empaquetado que utilizará para cifrar el material clave importado para su transporte en el paso 3.](#) AWS KMS también es compatible con la especificación clave de empaquetado SM2 (solo en las regiones de China).

Cada conjunto de claves públicas de encapsulamiento y token de importación es válido durante 24 horas. Si no los utiliza para importar material de claves en un plazo de 24 horas después de descargarlos, debe descargar un nuevo conjunto. Puede descargar nuevos conjuntos de claves públicas de encapsulamiento y tokens de importación en cualquier momento. Esto le permite cambiar la longitud de la clave de encapsulamiento RSA (“especificación de clave”) o sustituir un conjunto perdido.

También puede descargar un conjunto de claves públicas de encapsulamiento y un conjunto de tokens de importación para [volver a importar el mismo material de claves](#) en una clave de KMS. Puede hacerlo para establecer o cambiar el tiempo de vencimiento del material de claves o para restaurar el material de claves vencidas o eliminadas. Debe descargar y volver a cifrar el material clave cada vez que lo importe a AWS KMS.

Uso de la clave pública de encapsulamiento

La descarga incluye una clave pública exclusiva para usted Cuenta de AWS, también denominada clave pública empaquetadora.

Antes de importar el material clave, cifra el material clave con la clave de empaquetado pública y, a AWS KMS continuación, carga el material clave cifrado en. Cuando AWS KMS recibe el material clave cifrado, lo descifra con la clave privada correspondiente y, a continuación, lo vuelve a cifrar con una clave simétrica AES, todo ello dentro de un módulo de seguridad de AWS KMS hardware (HSM).

Uso del token de importación

La descarga incluye un token de importación que contiene metadatos para garantizar que el material de claves se importa correctamente. Al cargar el material de claves cifradas en AWS KMS, debe cargar el mismo token de importación que descargó en este paso.

Selección de una especificación de clave pública de encapsulamiento

Para proteger el material clave durante la importación, debes cifrarlo mediante la clave pública de empaquetado desde AWS KMS la que lo descargaste y un [algoritmo de empaquetado](#) compatible. Debe seleccionar una especificación de claves antes de descargar la clave pública de encapsulamiento y el token de importación. Todos los pares de claves de empaquetado se generan en módulos AWS KMS de seguridad de hardware (HSM). La clave privada nunca sale del HSM en texto sin formato.

Especificaciones de claves de encapsulamiento RSA

La especificación de clave de la clave pública de encapsulamiento determina la longitud de las claves del par de claves RSA que protege el material de la clave durante su transferencia a AWS KMS. En general, se recomienda utilizar la clave pública de encapsulamiento más larga que resulte práctica. Ofrecemos varias especificaciones de encapsulamiento de claves públicas para admitir una variedad de HSM y administradores de claves.

AWS KMS admite las siguientes especificaciones clave para las claves de empaquetado RSA que se utilizan para importar material clave de todo tipo, excepto cuando se indique lo contrario.

- RSA_4096 (recomendado)
- RSA_3072
- RSA_2048

Note

NO se admite la siguiente combinación: material de claves ECC_NIST_P521, la especificación de clave pública de encapsulamiento RSA_2048 y un algoritmo de encapsulamiento RSAES_OAEP_SHA_*.

No se puede encapsular directamente el material de claves ECC_NIST_P521 con una clave pública de encapsulamiento RSA_2048. Utilice una clave de encapsulamiento más grande o un algoritmo de encapsulamiento RSA_AES_KEY_WRAP_SHA_*.

Especificación de claves de encapsulamiento SM2 (solo en las regiones de China)

AWS KMS admite las siguientes especificaciones clave para las claves de empaquetado SM2 que se utilizan para importar material de clave asimétrico.

- SM2

Seleccionar un algoritmo de encapsulamiento

Para proteger su material de claves durante la importación, deberá cifrarlo con la clave pública de encapsulamiento descargada y un algoritmo de encapsulamiento admitido.

AWS KMS admite varios algoritmos de empaquetado RSA estándar y un algoritmo de empaquetado híbrido de dos pasos. En general, se recomienda utilizar el algoritmo de encapsulamiento más seguro que sea compatible con el material de claves y la [especificación de clave de encapsulamiento](#) importados. Normalmente, se elige un algoritmo admitido por el módulo de seguridad de hardware (HSM) o el sistema de administración de claves que protege el material de claves.

En la siguiente tabla se muestran los algoritmos de encapsulamiento compatibles con cada tipo de material de claves y clave de KMS. Los algoritmos se muestran en el orden de preferencia.

Material de claves	Algoritmo y especificaciones de encapsulamiento compatibles
Clave de cifrado simétrica	Algoritmos de encapsulamiento:
Clave AES de 256 bits	RSAES_OAEP_SHA_256
Clave SM4 de 128 bits (solo en las regiones de China)	RSAES_OAEP_SHA_1
	Algoritmos de encapsulamiento obsoletos:
	RSAES_PKCS1_V1
 Note A partir del 10 de octubre de 2023, AWS KMS no es compatible con el algoritmo de empaquetado RSAES_PKCS1_V1_5.	

Material de claves	Algoritmo y especificaciones de encapsulamiento compatibles
	Especificaciones de claves de encapsulamiento: - RSA_2048 - RSA_3072 - RSA_4096
Clave privada RSA asimétrica	Algoritmos de encapsulamiento: - RSA_AES_KEY_WRAP_SHA_256 - RSA_AES_KEY_WRAP_SHA_1 - SM2PKE (solo en regiones de China) Especificaciones de claves de encapsulamiento: - RSA_2048 - RSA_3072 - RSA_4096 - SM2 (solo en regiones de China)

Material de claves	Algoritmo y especificaciones de encapsulamiento compatibles
Clave privada de curva elíptica asimétrica (ECC) No puede utilizar los algoritmos de encapsulamiento RSAES_OAEP_SHA_* con la especificación de clave de encapsulamiento RSA_2048 para encapsular el material de claves ECC_NIST_P521.	Algoritmos de encapsulamiento: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (solo en regiones de China) Especificaciones de claves de encapsulamiento: RSA_2048 RSA_3072 RSA_4096 SM2 (solo en regiones de China)

Material de claves	Algoritmo y especificaciones de encapsulamiento compatibles
Clave privada SM2 asimétrica (solo en las regiones de China)	Algoritmos de encapsulamiento: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (solo en regiones de China) Especificaciones de claves de encapsulamiento: RSA_2048 RSA_3072 RSA_4096 SM2 (solo en regiones de China)
Clave HMAC	Algoritmos de encapsulamiento: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Especificaciones de claves de encapsulamiento: RSA_2048 RSA_3072 RSA_4096

 Note

Los algoritmos de encapsulamiento RSA_AES_KEY_WRAP_SHA_256 y RSA_AES_KEY_WRAP_SHA_1 no se admiten en las regiones de China.

- **RSA_AES_KEY_WRAP_SHA_256**: un algoritmo de encapsulamiento híbrido de dos pasos que combina el cifrado del material de claves con una clave simétrica AES que el usuario genere con el cifrado posterior de la clave simétrica AES con la clave de encapsulamiento pública RSA descargada y el algoritmo de encapsulamiento RSAES_OAEP_SHA_256.

Se necesita un algoritmo de encapsulamiento **RSA_AES_KEY_WRAP_SHA_*** para encapsular el material de claves privadas RSA, excepto en las regiones de China, donde se debe usar el algoritmo de encapsulamiento SM2PKE.

- **RSA_AES_KEY_WRAP_SHA_1**: un algoritmo de encapsulamiento híbrido de dos pasos que combina el cifrado del material de claves con una clave simétrica AES que el usuario genere con el cifrado posterior de la clave simétrica AES con la clave de encapsulamiento pública RSA descargada y el algoritmo de encapsulamiento RSAES_OAEP_SHA_1.

Se necesita un algoritmo de encapsulamiento **RSA_AES_KEY_WRAP_SHA_*** para encapsular el material de claves privadas RSA, excepto en las regiones de China, donde se debe usar el algoritmo de encapsulamiento SM2PKE.

- **RSAES_OAEP_SHA_256**— El algoritmo de cifrado RSA con un relleno de cifrado asimétrico óptimo (OAEP) con función hash. SHA-256
- **RSAES_OAEP_SHA_1**— El algoritmo de cifrado RSA con un relleno de cifrado asimétrico óptimo (OAEP) con función hash. SHA-1
- **RSAES_PKCS1_V1_5**(En desuso; desde el 10 de octubre de 2023, AWS KMS no es compatible con el algoritmo de empaquetado **RSAES_PKCS1_V1_5**): el algoritmo de cifrado RSA con el formato de relleno definido en la versión 1.5 del PKCS #1.
- **SM2PKE**(Solo regiones de China): algoritmo de cifrado basado en curvas elípticas definido por la OSCCA en GM/T 0003.4-2012.

Temas

- [Descarga de la clave pública de encapsulamiento y el token de importación \(consola\)](#)
- [Descargar la clave pública de empaquetado y el token de importación \(AWS KMS API\)](#)

Descarga de la clave pública de encapsulamiento y el token de importación (consola)

Puede usar la AWS KMS consola para descargar la clave pública de empaquetado y el token de importación.

1. Si acaba de completar los pasos para [crear una clave KMS sin material de claves](#) y se encuentra en la página Download wrapping key and import token (Descargar la clave de encapsulamiento y el token de importación), pase a [Step 10](#).
2. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
3. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
4. En el panel de navegación, elija Claves administradas por el cliente.

 Tip

Solo puede importar el material de claves a una clave de KMS con un Origen de Externo (importar material de claves). Esto indica que la clave KMS se ha creado sin material de claves. Para agregar la columna Origin (Origen) a la tabla, en la esquina superior derecha de la página, elija el icono de configuración



Active Origin (Origen) y, a continuación, elija Confirm (Confirmar).

5. Elija el alias o el ID de clave de la clave KMS que está pendiente de importación.
6. Elija la pestaña Cryptographic configuration (Configuración criptográfica) y vea sus valores. Las pestañas están debajo de la sección General configuration (Configuración general).

Solo se puede importar material de claves a claves de KMS con un Origen de Externo (importar material de claves). Para obtener información acerca de cómo crear claves KMS con material de claves importado, consulte [Importación de material clave para AWS KMS llaves](#).

7. Elige la pestaña adecuada según tu tipo de clave.
 - Para las claves asimétricas y HMAC, seleccione la pestaña Material clave.
 - Para las claves de cifrado simétricas, seleccione la pestaña Material y rotaciones de la clave.
8. Elija la acción de importación.
 - Para las claves asimétricas y HMAC, selecciona Importar material clave.
 - Para las claves de cifrado simétricas, elija una de las siguientes opciones:
 - Importe el material clave inicial (si aún no se ha importado ningún material clave)
 - Importe material clave nuevo (para añadir material nuevo para la rotación)

- Vuelva a importar el material clave (disponible en el menú Acciones de la tabla de materiales clave)

 Note

En el caso de las claves multirregionales, primero debe importar el nuevo material clave a la clave de región principal. A continuación, importe el mismo material clave en cada réplica de clave de región.

Para las claves principales multirregionales, la tabla de materiales clave incluye una columna de estado de importación de réplicas que muestra el estado de importación en todas las regiones de réplica (por ejemplo, «0 de 3 importadas»). Elija el valor del estado de importación de la réplica para abrir un modal que muestre el estado de importación de cada región de la réplica. El modal proporciona enlaces de importación de materiales clave para las regiones de réplica en las que no se ha importado el nuevo material clave.

9. En Seleccionar especificación de la clave de encapsulamiento, elija la configuración de su clave de KMS. Después de crear esta clave, no puede cambiar las especificaciones de clave.
10. En Select wrapping algorithm, elija la opción que usará para cifrar el material de claves. Para obtener más información acerca de estas opciones, consulte [Seleccionar un algoritmo de encapsulamiento](#).
11. Elija Descargar clave de encapsulamiento y token de importación y, a continuación, guarde el archivo.

Si dispone de la opción Next (Siguiente) para continuar con el proceso ahora, elija Next (Siguiente). Para continuar más adelante, elija Cancel (Cancelar).

12. Descomprima el archivo .zip que ha guardado en el paso anterior (Import_Parameters_<key_id>_<timestamp>).

La carpeta contiene los siguientes archivos:

- Una clave pública de encapsulamiento en un archivo llamado WrappingPublicKey.bin.
- Un token de importación en un archivo llamado ImportToken.bin.
- Un archivo de texto llamado README.txt. Este archivo contiene información sobre la clave pública de encapsulamiento, el algoritmo de encapsulamiento que se usará para cifrar el material de claves y la fecha y hora en que vencen la clave pública de encapsulamiento y el token de importación.

13. Para continuar el proceso, consulte [cifrar el material de claves](#).

Descargar la clave pública de empaquetado y el token de importación (AWS KMS API)

Para descargar la clave pública y el token de importación, usa la [GetParametersForImport](#) API. Especifique la clave de KMS que se asociará al material de claves importado. Esta clave de KMS debe tener un valor para [Origin](#) de EXTERNAL.

Note

No puedes importar material clave para las claves de ML-DSA KMS.

En este ejemplo se especifica el algoritmo de encapsulamiento RSA_AES_KEY_WRAP_SHA_256, la especificación de clave pública de encapsulamiento RSA_3072 y un ID de clave de ejemplo. Sustituya estos valores de ejemplo por valores válidos para la descarga. En el ID de la clave puede usar el [ID de la clave](#) o el [ARN de la clave](#), pero no puede usar un [nombre de alias](#) ni un [ARN de alias](#) en esta operación.

```
$ 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
```

Si el comando se ejecuta correctamente, verá un resultado parecido al siguiente:

```
{
  "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 los datos para el siguiente paso, base64 decodifica la clave pública e importa el token y guarda los valores decodificados en los archivos.

Para decodificar en base64 la clave pública y el token de importación:

1. Copie la clave pública codificada en base64 (representada *public key (base64 encoded)* en el resultado del ejemplo), péguela en un archivo nuevo y, a continuación, guárdelo. Póngale al archivo un nombre descriptivo, como por ejemplo `PublicKey.b64`.
2. Utilice [OpenSSL](#) para decodificar en base64 el contenido del archivo y guarde los datos descodificados en un nuevo archivo. El siguiente ejemplo descodifica los datos del archivo que ha guardado en el paso anterior (`PublicKey.b64`) y guarda el resultado en un nuevo archivo denominado `WrappingPublicKey.bin`.

```
$ openssl enc -d -base64 -A -in PublicKey.b64 -out WrappingPublicKey.bin
```

3. Copie el token de importación codificado *import token (base64 encoded)* en base64 (representado por el resultado del ejemplo), péguelo en un archivo nuevo y, a continuación, guárdelo. Asigne un nombre descriptivo al archivo, por ejemplo, `importtoken.b64`.
4. Utilice [OpenSSL](#) para decodificar en base64 el contenido del archivo y guarde los datos descodificados en un nuevo archivo. El siguiente ejemplo descodifica los datos del archivo que ha guardado en el paso anterior (`ImportToken.b64`) y guarda el resultado en un nuevo archivo denominado `ImportToken.bin`.

```
$ openssl enc -d -base64 -A -in importtoken.b64 -out ImportToken.bin
```

Continúe en [Paso 3: Cifrar el material de claves](#).

Paso 3: Cifrar el material de claves

Después de [descargar la clave pública y el token de importación](#), cifre el material de claves con la clave pública que descargó y el algoritmo de encapsulamiento que especificó. Si necesita reemplazar la clave pública o el token de importación, o cambiar el algoritmo de encapsulamiento, debe descargar una nueva clave pública y un token de importación. Para obtener información sobre las claves públicas y los algoritmos de empaquetado AWS KMS compatibles, consulte [Selección de una especificación de clave pública de encapsulamiento](#) y [Seleccionar un algoritmo de encapsulamiento](#).

El material de claves debe estar en formato binario. Para obtener información detallada, consulta [Requisitos para el material de claves importado](#).

 Note

Para los pares de claves asimétricas, cifra e importa solo la clave privada. AWS KMS deriva la clave pública de la clave privada.

NO se admite la siguiente combinación: material de claves ECC_NIST_P521, la especificación de clave pública de encapsulamiento RSA_2048 y un algoritmo de encapsulamiento RSAES_OAEP_SHA_*.

No se puede encapsular directamente el material de claves ECC_NIST_P521 con una clave pública de encapsulamiento RSA_2048. Utilice una clave de encapsulamiento más grande o un algoritmo de encapsulamiento RSA_AES_KEY_WRAP_SHA_*.

Los algoritmos de encapsulamiento RSA_AES_KEY_WRAP_SHA_256 y RSA_AES_KEY_WRAP_SHA_1 no se admiten en las regiones de China.

Normalmente, el material de claves se cifra al exportarlo desde el módulo de seguridad de hardware (HSM) o el sistema de administración de claves. Para obtener información sobre cómo exportar el material de claves en formato binario, consulte la documentación de su HSM o sistema de administración de claves. También puede consultar la siguiente sección que proporciona una demostración de prueba de concepto con OpenSSL.

Al cifrar el material de claves, utilice el mismo algoritmo de encapsulamiento que especificó al [descargar la clave pública y el token de importación](#). Para encontrar el algoritmo de empaquetado que especificó, consulte el evento de CloudTrail registro de la [GetParametersForImport](#) solicitud asociada.

Generación de material de claves para realizar pruebas

Los siguientes comandos de OpenSSL generan material de claves de cada tipo compatible para realizar pruebas. Estos ejemplos se proporcionan únicamente para pruebas y proof-of-concept demostraciones. En el caso de los sistemas de producción, utilice un método más seguro para generar el material de claves, como un módulo de seguridad de hardware o un sistema de administración de claves.

Para convertir las claves privadas de los pares de claves asimétricas a un formato cifrado en DER, transfiera el comando de generación de material de claves al siguiente comando `openssl pkcs8`. El parámetro `topk8` indica a OpenSSL que tome una clave privada como entrada y devuelva una clave con formato PKCS #8. (El comportamiento predeterminado es el contrario).

```
openssl pkcs8 -topk8 -outform der -nocrypt
```

Los siguientes comandos generan material de claves de prueba para cada uno de los tipos de clave compatibles.

- Clave de cifrado simétrica (32 bytes)

Este comando genera una clave simétrica de 256 bits (cadena aleatoria de 32 bytes) y la guarda en el archivo `PlaintextKeyMaterial.bin`. No es necesario cifrar este material de claves.

```
openssl rand -out PlaintextKeyMaterial.bin 32
```

Solo en las regiones de China, debe generar una clave simétrica de 128 bits (cadena aleatoria de 16 bytes).

```
openssl rand -out PlaintextKeyMaterial.bin 16
```

- Claves HMAC

Este comando genera una cadena de bytes aleatorios del tamaño especificado. No es necesario cifrar este material de claves.

La longitud de la clave HMAC debe coincidir con la longitud definida en la especificación de clave de la clave de KMS. Por ejemplo, si la clave de KMS es HMAC_384, debe importar una clave 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
```

- Claves privadas RSA

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:2048 | openssl pkcs8 -  
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
```

- Claves 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 claves privadas (solo regiones de China)

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:sm2 | openssl pkcs8 -topk8 -outform der -nocrypt > SM2_PrivateKey.der
```

Ejemplos de cifrado de material de claves con OpenSSL

Los siguientes ejemplos muestran cómo utilizar [OpenSSL](#) para cifrar el material de claves con la clave pública que ha descargado. Para cifrar el material clave con una clave SM2 pública (solo para las regiones de China), utilice la [SM2OfflineOperationHelper](#) clase. Para obtener más información sobre los tipos de materiales de claves que admite cada algoritmo de encapsulamiento, consulte [the section called “Seleccionar un algoritmo de encapsulamiento”](#).

Important

Estos ejemplos son solo una demostración de la prueba de concepto. En el caso de los sistemas de producción, utilice un método más seguro (como un HSM o un sistema de administración de claves comercial) para generar y almacenar el material de claves.

NO se admite la siguiente combinación: material de claves ECC_NIST_P521, la especificación de clave pública de encapsulamiento RSA_2048 y un algoritmo de encapsulamiento RSAES_OAEP_SHA_*.

No se puede encapsular directamente el material de claves ECC_NIST_P521 con una clave pública de encapsulamiento RSA_2048. Utilice una clave de encapsulamiento más grande o un algoritmo de encapsulamiento RSA_AES_KEY_WRAP_SHA_*.

RSAES_OAEP_SHA_1

AWS KMS admite la RSAES_OAEP_SHA_1 para claves de cifrado simétricas (SYMMETRIC_DEFAULT), claves privadas de curva elíptica (ECC), claves privadas y claves HMAC. SM2

RSAES_OAEP_SHA_1 no es compatible con las claves privadas RSA. Además, no puede usar una clave de encapsulamiento pública RSA_2048 con ningún algoritmo de encapsulamiento RSAES_OAEP_SHA_* para encapsular una clave privada ECC_NIST_P521 (secp521r1). Debe usar una clave de pública de encapsulamiento más grande o un algoritmo de encapsulamiento RSA_AES_KEY_WRAP.

El siguiente ejemplo cifra el material de claves con la [clave pública que ha descargado](#) y el algoritmo de encapsulamiento RSAES_OAEP_SHA_1, y lo guarda en el archivo EncryptedKeyMaterial.bin.

En este ejemplo:

- *WrappingPublicKey.bin* es el archivo que contiene la clave pública de encapsulamiento descargada.
- *PlaintextKeyMaterial.bin* es el archivo que contiene el material de claves que está cifrando, como PlaintextKeyMaterial.bin, HMAC_384_PlaintextKey.bin o ECC_NIST_P521_PrivateKey.der.

```
$ openssl pkeyutl \  
  -encrypt \  
  -in PlaintextKeyMaterial.bin \  
  -out EncryptedKeyMaterial.bin \  
  -inkey WrappingPublicKey.bin \  
  -keyform DER \  

```

```
-pubin \  
-pkeyopt rsa_padding_mode:oaep \  
-pkeyopt rsa_oaep_md:sha1
```

RSAES_OAEP_SHA_256

AWS KMS admite el RSAES_OAEP_SHA_256 para claves de cifrado simétricas (SYMMETRIC_DEFAULT), claves privadas de curva elíptica (ECC), claves privadas y claves HMAC. SM2

RSAES_OAEP_SHA_256 no es compatible con las claves privadas RSA. Además, no puede usar una clave de encapsulamiento pública RSA_2048 con ningún algoritmo de encapsulamiento RSAES_OAEP_SHA_* para encapsular una clave privada ECC_NIST_P521 (secp521r1). Debe usar una clave pública más grande o un algoritmo de encapsulamiento RSA_AES_KEY_WRAP.

El siguiente ejemplo cifra el material de claves con la [clave pública que ha descargado](#) y el algoritmo de encapsulamiento RSAES_OAEP_SHA_256, y lo guarda en el archivo EncryptedKeyMaterial.bin.

En este ejemplo:

- *WrappingPublicKey.bin* es el archivo que contiene la clave de encapsulamiento pública descargada. Si ha descargado la clave pública desde la consola, este archivo se denomina wrappingKey_KMS_key_key_ID_timestamp (por ejemplo, wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909).
- *PlaintextKeyMaterial.bin* es el archivo que contiene el material de claves que está cifrando, como PlaintextKeyMaterial.bin, HMAC_384_PlaintextKey.bin o ECC_NIST_P521_PrivateKey.der.

```
$ openssl pkeyutl \  
-encrypt \  
-in PlaintextKeyMaterial.bin \  
-out EncryptedKeyMaterial.bin \  
-inkey WrappingPublicKey.bin \  
-keyform DER \  
-pubin \  
-pkeyopt rsa_padding_mode:oaep \  
-pkeyopt rsa_oaep_md:sha256 \  
-pkeyopt rsa_mgf1_md:sha256
```

RSA_AES_KEY_WRAP_SHA_1

El algoritmo de encapsulamiento RSA_AES_KEY_WRAP_SHA_1 implica dos operaciones de cifrado.

1. Cifre el material de claves con una clave simétrica AES que genere y un algoritmo de cifrado simétrico AES.
2. Cifre la clave simétrica AES que utilizó con la clave pública que descargó y el algoritmo de encapsulamiento RSAES_OAEP_SHA_1.

El algoritmo de encapsulamiento RSA_AES_KEY_WRAP_SHA_1 requiere la versión 3.x de OpenSSL o posterior.

1. Generación de una clave de cifrado simétrica AES de 256 bits

Este comando genera una clave de cifrado simétrica AES que consta de 256 bits aleatorios y la guarda en el archivo `aes-key.bin`

```
# Generate a 32-byte AES symmetric encryption key
$ openssl rand -out aes-key.bin 32
```

2. Cifrado del material de claves con la clave de cifrado simétrica AES

Este comando cifra el material de claves con la clave de cifrado simétrica AES y guarda el material de claves cifrado en el archivo `key-material-wrapped.bin`.

En este comando de ejemplo:

- *PlaintextKeyMaterial.bin* es el archivo que contiene el material de claves que está importando, como `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der` o `ECC_NIST_P521_PrivateKey.der`.
- *aes-key.bin* es el archivo que contiene la clave de cifrado simétrica AES de 256 bits que generó en el 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. Cifrado de la clave de cifrado simétrica AES con la clave pública

Este comando cifra la clave de cifrado simétrica AES con la clave pública que ha descargado y el algoritmo de encapsulamiento RSAES_OAEP_SHA_1, la cifra en DER y la guarda en el archivo `aes-key-wrapped.bin`.

En este comando de ejemplo:

- *WrappingPublicKey.bin* es el archivo que contiene la clave de encapsulamiento pública descargada. Si ha descargado la clave pública desde la consola, este archivo se denomina `wrappingKey_KMS key_key_ID_timestamp` (por ejemplo, `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`).
- *aes-key.bin* es el archivo que contiene la clave de cifrado simétrica AES de 256 bits que generó en el primer comando de esta secuencia de ejemplo.

```
# 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. Generación del archivo que se va a importar

Concatene el archivo con el material de la clave cifrada y el archivo con la clave AES cifrada. Guárdelos en el archivo `EncryptedKeyMaterial.bin`, que es el archivo que va a importar en [Paso 4: Importar el material de claves](#).

En este comando de ejemplo:

- *key-material-wrapped.bin* es el archivo que contiene el material de claves cifrado.
- *aes-key-wrapped.bin* es el archivo que contiene la clave de cifrado AES cifrada.

```
# 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

El algoritmo de encapsulamiento RSA_AES_KEY_WRAP_SHA_256 implica dos operaciones de cifrado.

1. Cifre el material de claves con una clave simétrica AES que genere y un algoritmo de cifrado simétrico AES.
2. Cifre la clave simétrica AES que utilizó con la clave pública que descargó y el algoritmo de encapsulamiento RSAES_OAEP_SHA_256.

El algoritmo de encapsulamiento RSA_AES_KEY_WRAP_SHA_256 requiere la versión 3.x de OpenSSL o posterior.

1. Generación de una clave de cifrado simétrica AES de 256 bits

Este comando genera una clave de cifrado simétrica AES que consta de 256 bits aleatorios y la guarda en el archivo `aes-key.bin`

```
# Generate a 32-byte AES symmetric encryption key
$ openssl rand -out aes-key.bin 32
```

2. Cifrado del material de claves con la clave de cifrado simétrica AES

Este comando cifra el material de claves con la clave de cifrado simétrica AES y guarda el material de claves cifrado en el archivo `key-material-wrapped.bin`.

En este comando de ejemplo:

- *PlaintextKeyMaterial.bin* es el archivo que contiene el material de claves que está importando, como `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der` o `ECC_NIST_P521_PrivateKey.der`.
- *aes-key.bin* es el archivo que contiene la clave de cifrado simétrica AES de 256 bits que generó en el 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. Cifrado de la clave de cifrado simétrica AES con la clave pública

Este comando cifra la clave de cifrado simétrica AES con la clave pública que ha descargado y el algoritmo de encapsulamiento RSAES_OAEP_SHA_256, la cifra en DER y la guarda en el archivo `aes-key-wrapped.bin`.

En este comando de ejemplo:

- *WrappingPublicKey.bin* es el archivo que contiene la clave de encapsulamiento pública descargada. Si ha descargado la clave pública desde la consola, este archivo se denomina `wrappingKey_KMS key_key_ID_timestamp` (por ejemplo, `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`).
- *aes-key.bin* es el archivo que contiene la clave de cifrado simétrica AES de 256 bits que generó en el primer comando de esta secuencia de ejemplo.

```
# 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. Generación del archivo que se va a importar

Concatene el archivo con el material de la clave cifrada y el archivo con la clave AES cifrada. Guárdelos en el archivo `EncryptedKeyMaterial.bin`, que es el archivo que va a importar en [Paso 4: Importar el material de claves](#).

En este comando de ejemplo:

- `key-material-wrapped.bin` es el archivo que contiene el material de claves cifrado.
- `aes-key-wrapped.bin` es el archivo que contiene la clave de cifrado AES cifrada.

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

Continúe en [Paso 4: Importar el material de claves](#).

Paso 4: Importar el material de claves

Después de [cifrar el material de claves](#), puede importar el material de claves para usarlo con una AWS KMS key. Para importar el material de claves, debe cargar el material de claves cifrado desde [Paso 3: Cifrar el material de claves](#) y el token de importación que ha descargado en [Paso 2: descargar la clave pública de encapsulamiento y el token de importación](#). Debe importar material de claves en la misma clave KMS que ha especificado al [descargar la clave pública y el token de importación](#). Cuando se importa el material de claves, el [estado de la clave](#) de KMS cambia a `Enabled`. Puede usar la clave de KMS en operaciones criptográficas.

Al importar el material de claves de importación, puede [establecer una fecha de vencimiento opcional](#) para el material de claves. Cuando vence el material de claves, AWS KMS lo elimina y ya no se puede utilizar la clave KMS. Después de importar el material de claves, no puede establecer, cambiar ni cancelar la fecha de caducidad de la importación actual. Para cambiar estos valores, debe [volver a importar](#) el mismo material de claves.

En el caso de todas las claves de KMS con origen `EXTERNAL`, el primer material de claves importado pasa a ser el actual y se le asocia permanentemente. Las claves de cifrado simétricas con `EXTERNAL` origen admiten la rotación bajo demanda. Puede asociar varios materiales clave con claves importadas que admiten la rotación bajo demanda. El proceso de importación de material clave nuevo es diferente para las claves de una sola región y para las de varias regiones, tal como se describe en la sección [Importación de material clave nuevo](#). Debe establecer el

`importType` parámetro en `NEW_KEY_MATERIAL` junto con la [ImportKeyMaterial](#) acción para asociar el nuevo material clave a una clave KMS. El valor predeterminado del parámetro `ImportType` opcional es `EXISTING_KEY_MATERIAL`. Al omitir el parámetro `ImportType` o especificarlo como `EXISTING_KEY_MATERIAL`, se debe importar un material de claves que esté previamente asociado a la clave de KMS.

En el caso de las claves asimétricas o HMAC KMS con `EXTERNAL` origen, solo se puede asociar un material clave a la clave. AWS KMS rechazará las solicitudes [ImportKeyMaterial](#) de API con el `ImportType` parámetro.

Cuando se importan todos los materiales de claves asociados permanentemente a una clave de KMS, la clave de KMS está disponible para su uso en operaciones criptográficas. Si alguno de estos materiales de claves se elimina o se deja que caduque, el estado de la clave de KMS cambia a `PendingImport` y la clave no se puede utilizar para operaciones criptográficas.

Para importar material clave, puedes usar la [AWS KMS consola](#) o la [ImportKeyMaterial](#) API. Puedes usar la API directamente realizando solicitudes HTTP o usando un [AWS SDKs](#), [AWS Command Line Interface](#) o [Herramientas de AWS para PowerShell](#).

Al importar el material clave, se agrega una [ImportKeyMaterial](#) entrada al AWS CloudTrail registro para registrar la `ImportKeyMaterial` operación. La CloudTrail entrada es la misma tanto si utilizas la AWS KMS consola como la AWS KMS API.

Configuración de una fecha de vencimiento (opcional)

Al importar el material de claves para su clave de KMS, puede establecer una fecha y hora de vencimiento opcionales para el material de claves de hasta 365 días a partir de la fecha de importación. Cuando el material clave importado caduca, lo AWS KMS elimina. Esta acción cambia el [estado de clave](#) de la clave de KMS a `PendingImport`, lo que evita que se la utilice en operaciones criptográficas. Para usar la clave de KMS, debe [volver a importar una copia del material de claves original](#).

Garantizar que el material de claves importado venza con frecuencia puede ayudarlo a cumplir con los requisitos normativos, pero supone un riesgo adicional para los datos cifrados con la clave de KMS. Hasta que no se vuelva a importar una copia del material de claves original, no se podrá utilizar una clave de KMS con material de claves vencido y no se podrá acceder a los datos cifrados con la clave de KMS. Si no vuelve a importar el material de claves por cualquier motivo, incluida la pérdida de la copia del material de claves original, la clave de KMS quedará inutilizable de forma permanente y los datos cifrados con la clave de KMS no se podrán recuperar.

Para mitigar este riesgo, asegúrese de poder acceder a su copia del material clave importado y diseñe un sistema para eliminar y volver a importar el material clave antes de que caduque e interrumpa su carga de trabajo. AWS Le recomendamos que [active una alarma](#) que le indique el vencimiento del material de claves importado para que tenga tiempo suficiente para volver a importarlo antes de que venza. También puede utilizar sus CloudTrail registros para auditar las operaciones de [importación \(y reimportación\) de material clave y eliminar material clave importado, así como la AWS KMS operación de eliminación del material clave caducado](#).

AWS KMS no puede restaurar, recuperar ni reproducir el material clave eliminado. En lugar de establecer una fecha de vencimiento, puede [eliminar](#) y [volver a importar](#) periódicamente el material de claves importado mediante programación, pero los requisitos para retener una copia del material de claves original son los mismos.

Usted determina si el material de claves importado vence y cuándo lo hace cuando importa el material de claves. Puede activar y desactivar el vencimiento o establecer una nueva fecha de vencimiento al volver a importar el material de claves. Utilice el `ExpirationModel` parámetro [ImportKeyMaterial](#) para activar () y desactivar (`KEY_MATERIAL_EXPIRES`) la caducidad y el `ValidTo` parámetro para establecer la hora de caducidad. `KEY_MATERIAL_DOES_NOT_EXPIRE` El tiempo máximo es de 365 días a partir de los datos de importación; no hay un mínimo, pero la fecha debe ser en el futuro.

Establecimiento de la descripción de material de claves

Las claves de cifrado simétricas con `EXTERNAL` origen pueden tener varios materiales clave asociados. Puede especificar una descripción del material de claves opcional al importarlo a dichas claves. La descripción se puede utilizar para hacer un seguimiento de dónde se guarda de forma duradera el material de claves correspondiente fuera de AWS KMS.

En el caso de las claves multirregionales, puede establecer o cambiar la descripción del material de la clave únicamente en la clave de región principal. AWS KMS propaga automáticamente la descripción del material clave a las réplicas de las claves de región.

Importación de nuevo material de claves

Para realizar la rotación bajo demanda de una clave de KMS de cifrado simétrico con material de claves importado, primero tendrá que importar un material de claves nuevo, que no esté asociado previamente a la clave.

- Claves de una sola región

- Utilice la [ImportKeyMaterial](#) operación con el `ImportType` parámetro establecido en `NEW_KEY_MATERIAL` para realizar esta tarea. Este material clave no está asociado permanentemente a la tecla hasta que no se realice la [RotateKeyOnDemand](#) operación o se gire la llave en el Consola de administración de AWS. Hasta entonces, este material de claves estará en el estado `PENDING_ROTATION`. Una clave de KMS puede tener como máximo un material clave en el estado `PENDING_ROTATION` en cualquier momento. Un material clave en `PENDING_ROTATION` estado se puede eliminar sin que ello afecte a la usabilidad de la clave en las operaciones criptográficas.
- Claves de varias regiones
 - Para importar material clave a una clave multirregional, primero debe importar el nuevo material clave a la clave de región principal. No puede importar directamente nuevos materiales clave para replicar las claves de región. Tras importar el nuevo material clave a la clave de región principal, puede importar los mismos materiales clave a las réplicas de las claves de región.
 - Utilice la [ImportKeyMaterial](#) operación con el `ImportType` parámetro establecido en `NEW_KEY_MATERIAL` para la clave de región principal para realizar esta tarea. Para la clave de región de réplica, utilice el `EXISTING_KEY_MATERIAL` parámetro `ImportType` for para la `ImportKeyMaterial` operación.
 - El material clave para el cifrado simétrico Las claves multirregionales deben importarse a todas las réplicas de claves de región y a las claves de región principales antes de que el estado del material clave cambie a estado. `PENDING_ROTATION` Hasta entonces, el estado del nuevo material clave es. `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` Una clave KMS puede tener como máximo un material clave `PENDING_ROTATION` o un `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` estado en cualquier momento (consulte la `KeyMaterialState` descripción en [RotationsListEntry](#)). Un material clave en `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` o `PENDING_ROTATION` estado no está asociado permanentemente a la clave y se puede eliminar sin que ello afecte a la usabilidad de la clave en las operaciones criptográficas.

Volver a importar material de claves

Si administra una clave KMS con material de clave importado, es posible que tenga que reimportar el material de clave. Puede volver a importar el material de claves para reemplazar el material de claves vencido o eliminado o para cambiar el modelo de vencimiento o la fecha de vencimiento del material de claves.

Puede volver a importar material de claves en cualquier momento y en cualquier horario que cumpla con sus requisitos de seguridad. No tiene que esperar hasta que el material de claves esté en su fecha de vencimiento o cerca de ella.

El procedimiento para volver a importar material de claves es el mismo procedimiento que se utilizó para importar el material de claves la primera vez, con las siguientes excepciones.

- Utilice una clave KMS existente en lugar de crear una nueva clave KMS. Puede omitir el [paso 1](#) del procedimiento de importación.
- Al volver a importar el material de clave, puede cambiar el modelo de vencimiento y la fecha de vencimiento. En el caso de las claves de cifrado simétricas, también puede cambiar la descripción del material de la clave.

En el caso de las claves multirregionales, puede establecer o cambiar la descripción del material de la clave solo en la clave de región principal. AWS KMS propaga automáticamente la descripción del material clave a las réplicas de las claves de región.

Cada vez que se importa material de claves en una clave KMS, es necesario [descargar y utilizar una nueva clave de encapsulamiento y un nuevo token de importación](#) para la clave KMS. El procedimiento de encapsulamiento no afecta al contenido del material de claves, por lo que puede utilizar distintas claves públicas de encapsulamiento y diferentes algoritmos de encapsulamiento para importar el mismo material de claves.

Importar el material de claves (consola)

Puede utilizarlos Consola de administración de AWS para importar material clave.

1. Si se encuentra en la página Cargar material de claves encapsulado, vaya a [Step 10](#).
2. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
3. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
4. En el panel de navegación, elija Claves administradas por el cliente.
5. Elija el ID de clave o el alias de la clave KMS para los que ha descargado la clave pública y el token de importación.

6. Elija la pestaña **Cryptographic configuration** (Configuración criptográfica) y vea sus valores. Las pestañas se encuentran en la página de detalles de una clave KMS debajo de la sección **General configuration** (Configuración general).

Solo se puede importar material de claves a claves de KMS con un Origen de Externo (importar material de claves). Para obtener información sobre cómo crear claves KMS con material de claves importado, consulte [Importación de material clave para AWS KMS llaves](#).

7. Elige la pestaña adecuada según tu tipo de clave.
 - Para las claves asimétricas y HMAC, seleccione la pestaña **Material clave**.
 - Para las claves de cifrado simétricas, seleccione la pestaña **Material** y rotaciones de la clave.
8. Elija la acción de importación.
 - Para las claves asimétricas y HMAC, selecciona **Importar material clave**.
 - Para las claves de cifrado simétricas, elija una de las siguientes opciones:
 - **Importe el material clave inicial** (si aún no se ha importado ningún material clave)
 - **Importe material clave nuevo** (para añadir material nuevo para la rotación)
 - **Vuelva a importar el material clave** (disponible en el menú **Acciones** de la tabla de materiales clave)

 Note

En el caso de las claves multirregionales, primero debe importar el nuevo material clave a la clave de región principal. A continuación, importe el mismo material clave en cada réplica de clave de región.

Para las claves principales multirregionales, la tabla de materiales clave incluye una columna de estado de importación de réplicas que muestra el estado de importación en todas las regiones de réplica (por ejemplo, «0 de 3 importadas»). Elija el valor del estado de importación de la réplica para abrir un modal que muestre el estado de importación de cada región de la réplica. El modal proporciona enlaces de importación de materiales clave para las regiones de réplica en las que no se ha importado el nuevo material clave.

9. Si descargó el material de claves, importó el token y cifró el material de claves, seleccione **Siguiente**.

 Note

En el caso de las claves multirregionales, primero debe importar el nuevo material clave a la clave de región principal. A continuación, puede importar el mismo material clave a las réplicas de las claves de región.

10. En la sección Token de importación y material de claves cifrado, haga lo siguiente:
 - a. En Material de claves encapsulado, seleccione Elegir archivo. A continuación, especifique el archivo que contiene el material de claves encapsulado (cifrado).
 - b. En Token de importación, seleccione Elegir archivo. Suba el archivo que contenga el token de importación que ha [descargado](#).
11. En la sección Expiration option (Opción de vencimiento), determina si vence el material de claves. Para establecer una fecha y una hora de vencimiento, elija El material de claves caduca, y seleccione una fecha y una hora en el calendario. Puede especificar una fecha de hasta 365 días a partir de la hora y fecha actuales.
12. En el caso de las claves de cifrado simétricas, si lo desea, puede especificar una descripción del material de claves que se va a importar.
13. Seleccione Importar material de claves.

Importe material clave (AWS KMS API)

Para importar material clave, utilice la [ImportKeyMaterial](#) operación. Los ejemplos siguientes utilizan la [AWS CLI](#), pero puede usar cualquier lenguaje de programación admitido.

Para usar este ejemplo:

1. Sustituya `1234abcd-12ab-34cd-56ef-1234567890ab` por un ID de clave de la clave KMS que usó cuando descargó la clave pública y el token de importación. Para identificar la clave KMS, utilice su [ID de clave](#) o su [ARN de clave](#). No puede utilizar un [nombre de alias](#) o un [ARN de alias](#) para esta operación.
2. Sustituya `EncryptedKeyMaterial.bin` por el nombre del archivo que contiene el material de claves cifradas.
3. Sustituya `ImportToken.bin` por el nombre del archivo que contiene el token de importación.
4. Si desea que el material de claves importado caduque, defina el valor del parámetro `expiration-model` a su valor predeterminado, `KEY_MATERIAL_EXPIRES`, u omita el parámetro

expiration-model. A continuación, sustituya el valor del parámetro valid-to con la fecha y la hora en que desea que caduque el material de claves. La fecha y la hora pueden ser hasta 365 días a partir del momento de la solicitud.

```
$ 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
```

Si no desea que el material de claves importado caduque, defina el valor del parámetro expiration-model a KEY_MATERIAL_DOES_NOT_EXPIRE y omita el parámetro valid-to del 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. Si desea importar material de claves nuevo, no asociado anteriormente a la clave de KMS, defina el parámetro ImportType en NEW_KEY_MATERIAL. Esta opción solo se puede utilizar con claves de cifrado simétricas. Para estas claves, también puede usar el parámetro KeyMaterialDescription opcional para establecer una descripción del material de claves importado en el siguiente ejemplo de línea de comandos:

```
$ 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. En el caso de las claves multirregionales, puede establecer o cambiar la descripción del material de la clave únicamente en la clave de región principal. AWS KMS propaga automáticamente la descripción del material clave a las réplicas de las claves de región.

 Tip

Si el comando no se ejecuta correctamente, es posible que aparezca `KMSInvalidStateException` o `NotFoundException`. Puede reintentar la solicitud.

Crear una clave KMS en un almacén de AWS CloudHSM claves

Después de crear un almacén de AWS CloudHSM claves, puede crearlo AWS KMS keys en su almacén de claves. Deben ser [claves KMS de cifrado simétrico con el material de claves](#) que se AWS KMS generen. No puede crear [claves KMS asimétricas](#), [claves KMS HMAC](#) ni claves KMS con [material clave importado](#) en un almacén de claves personalizado. Además, no puede utilizar claves KMS de cifrado simétricas en un almacén de claves personalizado para generar pares de claves de datos asimétricos. El KMS no puede comunicarse IPv6 con los almacenes de AWS CloudHSM claves.

Para crear una clave KMS en un AWS CloudHSM almacén de AWS CloudHSM claves, el almacén de claves debe estar [conectado al AWS CloudHSM clúster asociado](#) y el clúster debe contener al menos dos activos HSMs en distintas zonas de disponibilidad. Para averiguar el estado y el número de la conexión HSMs, consulte la [página de almacenes de AWS CloudHSM claves](#) en Consola de administración de AWS. Cuando utilice las operaciones de la API, utilice la [DescribeCustomKeyStores](#) operación para comprobar que el almacén de AWS CloudHSM claves esté conectado. Para verificar la cantidad de activos HSMs en el clúster y sus zonas de disponibilidad, utilice la AWS CloudHSM [DescribeClusters](#) operación.

Al crear una clave de KMS en el almacén de AWS CloudHSM claves, AWS KMS crea la clave de KMS AWS KMS. Sin embargo, crea el material clave para la clave KMS en el AWS CloudHSM clúster asociado. En concreto, AWS KMS inicia sesión en el clúster como la [kmsuserCU que creó](#). A continuación crea una clave simétrica Advanced Encryption Standard (AES) de 256 bits, no extraíble y persistente en el clúster. AWS KMS establece el valor del [atributo de la etiqueta de claves](#), que solo es visible en el clúster, en el Nombre de recurso de Amazon (ARN) de la clave KMS.

Cuando el comando se ejecuta correctamente, el [estado de clave](#) de la nueva clave KMS es `Enabled` y su origen es `AWS_CLOUDHSM`. No se puede cambiar el origen de ninguna clave KMS después de crearla. Al ver una clave de KMS en un almacén de AWS CloudHSM claves de la AWS KMS consola o al utilizar la [DescribeKey](#) operación, puede ver las propiedades típicas, como el identificador de clave, el estado de la clave y la fecha de creación. Pero también puede ver el ID del almacén de claves personalizado y (de forma opcional) el ID del clúster de AWS CloudHSM .

Si se produce un error al intentar crear una clave KMS en el almacén de AWS CloudHSM claves, utilice el mensaje de error como ayuda para determinar la causa. Puede indicar que el almacén de AWS CloudHSM claves no está conectado (`CustomKeyStoreInvalidStateException`) o HSMs que el AWS CloudHSM clúster asociado no tiene los dos activos necesarios para esta operación (`CloudHsmClusterInvalidConfigurationException`). Para obtener ayuda, consulte [Resolver problemas de un almacén de claves personalizado](#).

Para ver un ejemplo del AWS CloudTrail registro de la operación que crea una clave KMS en un almacén de AWS CloudHSM claves, consulte [CreateKey](#).

Creación de una nueva clave KMS en su almacén de claves de CloudHSM

Puede crear una clave KMS de cifrado simétrico en el almacén de AWS CloudHSM claves de la AWS KMS consola o mediante la [CreateKey](#) operación.

Uso de la consola AWS KMS

Utilice el siguiente procedimiento para crear una clave KMS de cifrado simétrico en un almacén de AWS CloudHSM claves.

Note

No incluya información confidencial en el alias, la descripción ni las etiquetas. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Elija Crear clave.
5. Seleccione Symmetric (Simétrica).
6. En Key usage (Uso de claves), se selecciona la opción Encrypt and decrypt (Cifrar y descifrar) para usted. No la cambie.
7. Elija Advanced options (Opciones avanzadas).
8. En Origen del material de claves, elija Almacén de claves de AWS CloudHSM .

No puede crear una clave multirregional en un AWS CloudHSM almacén de claves.

9. Elija Siguiente.
10. Seleccione un almacén de AWS CloudHSM claves para su nueva clave KMS. Para crear un nuevo almacén de AWS CloudHSM claves, elija Crear un almacén de claves personalizado.

El almacén de AWS CloudHSM claves que seleccione debe tener el estado Conectado. Su AWS CloudHSM clúster asociado debe estar activo y contener al menos dos activos HSMs en distintas zonas de disponibilidad.

Si necesita ayuda para conectar un almacén de AWS CloudHSM claves, consulte [Desconectar un almacén de AWS CloudHSM claves](#). Para obtener ayuda sobre cómo añadir un HSM HSMs, consulte [Añadir un HSM](#) en la Guía del AWS CloudHSM usuario.

11. Elija Siguiente.
12. Escriba un alias y, si lo desea, una descripción para la clave KMS.
13. (Opcional). En la página agregar etiquetas, añada etiquetas que identifiquen o categoricen la clave KMS.

Al añadir etiquetas a AWS los recursos, AWS genera un informe de asignación de costes con el uso y los costes agregados por etiquetas. Las etiquetas también pueden utilizarse para controlar el acceso a una clave KMS. Para obtener información acerca del etiquetado de claves KMS, consulte [Etiquetas en AWS KMS](#) y [ABAC para AWS KMS](#).

14. Elija Siguiente.
15. En la sección administradores de claves, seleccione los usuarios y roles de IAM que pueden administrar la clave KMS. Para obtener más información, consulte [Permite que los administradores de claves administren la clave de KMS](#).

Notas

Las políticas de IAM pueden otorgar permisos para usar la clave KMS a otros usuarios y roles de IAM.

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;. La AWS KMS consola agrega los administradores clave a la política clave bajo el identificador de la declaración "Allow access for Key Administrators". La

modificación de este identificador de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

16. (Opcional) Para evitar que estos administradores de claves eliminen esta clave KMS, desactive la casilla Allow key administrators to delete this key (Permitir que los administradores de claves eliminen esta clave) situada en la parte inferior de la página.
17. Elija Siguiente.
18. En la sección Esta cuenta, seleccione los usuarios y roles de IAM Cuenta de AWS que pueden usar la clave KMS en operaciones [criptográficas](#). Para obtener más información, consulte [Permite a los usuarios de claves utilizar la clave de KMS](#).

Notas

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;. La AWS KMS consola añade los usuarios clave a la política clave bajo los identificadores "Allow use of the key" de la declaración y "Allow attachment of persistent resources". La modificación de estos identificadores de declaración puede afectar a la forma en que la consola muestra las actualizaciones que realice en la declaración.

19. (Opcional) Puede permitir que otras Cuentas de AWS usen esta clave KMS para operaciones criptográficas. Para ello, en la Cuentas de AWS sección Otros, en la parte inferior de la página, selecciona Añadir otra Cuenta de AWS e introduce el Cuenta de AWS ID de una cuenta externa. Para agregar varias cuentas externas, repita este paso.

Note

Los administradores de la otra también Cuentas de AWS deben permitir el acceso a la clave de KMS mediante la creación de políticas de IAM para sus usuarios. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

20. Elija Siguiente.

21. Revise las declaraciones de política de claves para ver la clave. Seleccione Editar para realizar cambios en la política de claves.
22. Elija Siguiente.
23. Revise los ajustes de clave que ha elegido. Aún puede volver atrás y cambiar todos los ajustes.
24. Cuando haya acabado, elija Finish (Finalizar) para crear la clave.

Cuando el procedimiento se complete correctamente, la pantalla mostrará la nueva clave KMS en el almacén de AWS CloudHSM claves que haya elegido. Al elegir el nombre o el alias de la nueva clave de KMS, la pestaña de configuración criptográfica de su página de detalles muestra el origen de la clave de KMS (AWS CloudHSM), el nombre, el ID y el tipo del almacén de claves personalizado y el ID del clúster. AWS CloudHSM Si el procedimiento falla, aparecerá un mensaje de error que describe el motivo del error.

Tip

Para facilitar la identificación de las claves KMS en un almacén de claves personalizado, en la página Customer managed keys (Claves administradas por el cliente) agregue la columna Custom key store ID (ID del almacén de claves personalizado). Haga clic en el icono de engranaje en la esquina superior derecha y, a continuación, seleccione Custom key store ID (ID del almacén de claves personalizado). Para obtener más información, consulte [Personalización de la vista de la consola](#).

Uso de la API AWS KMS

Para crear una nueva AWS KMS key (clave KMS) en su almacén de AWS CloudHSM claves, utilice la [CreateKey](#) operación. Use el parámetro CustomKeyId para identificar su almacén de claves personalizado y en el valor Origin especifique AWS_CLOUDHSM.

Es posible que también desee utilizar el parámetro Policy para especificar una política de claves. Puede cambiar la política de claves ([PutKeyPolicy](#)) y añadir elementos opcionales, como una [descripción](#) y [etiquetas](#), en cualquier momento.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

El siguiente ejemplo comienza con una llamada a la [DescribeCustomKeyStores](#) operación para comprobar que el almacén de AWS CloudHSM claves está conectado al AWS CloudHSM

clúster asociado. De forma predeterminada, esta operación devuelve los almacenes de claves personalizados de su cuenta y región. Para describir únicamente un almacén de AWS CloudHSM claves concreto, utilice su CustomKeyName parámetro CustomKeyId o (pero no ambos).

Antes de ejecutar este comando, reemplace el ID del almacén de claves personalizado de ejemplo por un ID válido.

 Note

No incluya información confidencial en los campos Description o Tags. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleKeyStore",
      "CustomKeyType": "AWS CloudHSM key store",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
    }
  ],
}
```

El siguiente comando de ejemplo utiliza la [DescribeClusters](#) operación para comprobar que el AWS CloudHSM clúster asociado al ExampleKeyStore (cluster-1a23b4cdefg) tiene al menos dos activos. HSMs Si el clúster tiene menos de dos, se produce un error en la operación. HSMs CreateKey

```
$ aws cloudhsmv2 describe-clusters
{
  "Clusters": [
    {
      "SubnetMapping": {
        ...
      },
      "CreateTimestamp": 1507133412.351,
      "ClusterId": "cluster-1a23b4cdefg",
      "SecurityGroup": "sg-865af2fb",
    }
  ]
}
```

```

    "HsmType": "hsm1.medium",
    "VpcId": "vpc-1a2b3c4d",
    "BackupPolicy": "DEFAULT",
    "Certificates": {
      "ClusterCertificate": "-----BEGIN CERTIFICATE-----\...\n-----END
CERTIFICATE-----\n"
    },
    "Hsms": [
      {
        "AvailabilityZone": "us-west-2a",
        "EniIp": "10.0.1.11",
        "ClusterId": "cluster-1a23b4cdefg",
        "EniId": "eni-ea8647e1",
        "StateMessage": "HSM created.",
        "SubnetId": "subnet-a6b10bd1",
        "HsmId": "hsm-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 comando de ejemplo usa la [CreateKey](#) operación para crear una clave KMS en un almacén de AWS CloudHSM claves. Para crear una clave KMS en un almacén de AWS CloudHSM claves, debe proporcionar el ID del almacén de claves personalizado del almacén de AWS CloudHSM claves y especificar un `Origin` valor de `AWS_CLOUDHSM`.

La respuesta incluye el almacén IDs de claves personalizado y el AWS CloudHSM clúster.

Antes de ejecutar este comando, reemplace el ID del almacén de claves personalizado de ejemplo por un 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",
    "CustomKeyId": "cks-1234567890abcdef0",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ]
  }
}
```

Creación de una clave KMS en un almacén de claves externo

Una vez [creado](#) y [conectado](#) el almacén de claves externo, puede crearlo AWS KMS keys en el almacén de claves. Deben ser [claves de KMS de cifrado simétrico](#) con un valor de origen de Almacén de claves externo (EXTERNAL_KEY_STORE). No puede crear [claves KMS asimétricas](#), [claves KMS HMAC](#) ni claves KMS con [material clave importado](#) en un almacén de claves personalizado. Además, no puede utilizar claves KMS de cifrado simétricas en un almacén de claves personalizado para generar pares de claves de datos asimétricos.

Una clave de KMS de un almacén de claves externo puede tener una latencia, durabilidad y disponibilidad más bajas que una clave de KMS estándar porque depende de componentes ubicados fuera de AWS. Antes de crear o utilizar una clave de KMS en un almacén de claves externo, compruebe que necesita una clave con las propiedades del almacén de claves externo.

 Note

Algunos administradores de claves externos proporcionan un método más sencillo para crear claves de KMS en un almacén de claves externo. Para obtener más detalles, consulte la documentación del administrador de claves externo.

Para crear una clave de KMS en su almacén de claves externo, especifique lo siguiente:

- El ID de su almacén de claves externo.
- Un [origen de material de claves](#) del almacén de claves externo (EXTERNAL_KEY_STORE).
- El ID de una [clave externa](#) existente en el [administrador de claves externo](#) asociado al almacén de claves externo. Esta clave externa sirve como material de claves para la clave de KMS. No se puede cambiar el ID de clave externa después de crear la clave de KMS.

AWS KMS proporciona el identificador de clave externo al proxy del almacén de claves externo en las solicitudes de operaciones de cifrado y descifrado. AWS KMS no puede acceder directamente a su administrador de claves externo ni a ninguna de sus claves criptográficas.

Además de la clave externa, una clave KMS de un almacén de claves externo también contiene material AWS KMS clave. Todos los datos cifrados con la clave KMS se cifran primero AWS KMS con el material clave de la AWS KMS clave y, a continuación, con el administrador de claves externo con la clave externa. Este proceso de [doble cifrado](#) garantiza que el texto cifrado protegido por una clave de KMS en un almacén de claves externo sea tan seguro como el texto cifrado protegido solo por AWS KMS. Para obtener más información, consulte [Cómo funcionan los almacenes de claves externos](#).

Cuando la operación CreateKey se realiza correctamente, el [estado de la clave](#) de la nueva clave de KMS es Enabled. Al [ver una clave de KMS en un almacén de claves externo](#), puede ver las propiedades habituales, como el ID de la clave, la [especificación de la clave](#), el [uso de la clave](#), el [estado de la clave](#) y la fecha de creación. Sin embargo, también puede ver el ID y el [estado de la conexión](#) del almacén de claves externo y el ID de la clave externa.

Si intenta crear una clave de KMS en su almacén de claves externo sin éxito, utilice el mensaje de error para identificar la causa. Puede indicar que el almacén de claves externo no está conectado (CustomKeyStoreInvalidStateException), que el proxy del almacén de claves externo no puede encontrar una clave externa con el ID de clave externa especificado

(XksKeyNotFoundException) o que la clave externa ya está asociada a una clave de KMS en el mismo almacén de claves externo (XksKeyAlreadyInUseException).

Para ver un ejemplo del AWS CloudTrail registro de la operación que crea una clave KMS en un almacén de claves externo, consulte [CreateKey](#).

Temas

- [Requisitos para una clave de KMS en un almacén de claves externo](#)
- [Creación de una clave KMS en un almacén de claves externo](#)

Requisitos para una clave de KMS en un almacén de claves externo

Para crear una clave de KMS en un almacén de claves externo, se requieren las siguientes propiedades del almacén de claves externo, la clave de KMS y la clave externa que sirve como material de clave criptográfica externa para la clave de KMS.

Requisitos de almacenes de claves externos

- Debe estar conectado a su proxy del almacén de claves externo.

Para ver el [estado de conexión](#) de un almacén de claves externo, consulte [Visualización de almacenes de claves externos](#). Para conectar su almacén de claves externo, consulte [Conexión y desconexión de almacenes de claves externos](#).

Requisitos de la clave de KMS

No puede cambiar estas propiedades después de crear la clave de KMS.

- Especificación de clave: SYMMETRIC_DEFAULT
- Uso de clave: ENCRYPT_DECRYPT
- Origen del material de claves: EXTERNAL_KEY_STORE
- Multirregión: FALSE

Requisitos de clave externa

- Clave criptográfica AES de 256 bits (256 bits aleatorios). El KeySpec de la clave externa debe ser AES_256.

- Habilitado y disponible para usarse. El Status de la clave externa debe ser ENABLED.
- Configurado para el cifrado y descifrado. El KeyUsage de la clave externa debe incluir ENCRYPT y DECRYPT.
- Se usa solo con esta clave de KMS. Cada KMS key de un almacén de claves externo debe estar asociada a una clave externa diferente.

AWS KMS también recomienda que la clave externa se utilice exclusivamente para el almacén de claves externo. Esta restricción facilita la identificación y la resolución de problemas con la clave.

- Accesible mediante el [proxy del almacén de claves externo](#) para el almacén de claves externo.

Si el proxy del almacén de claves externo no puede encontrar la clave con el ID de clave externa especificado, se produce un error en la operación CreateKey.

- Puede gestionar el tráfico previsto que Servicios de AWS genera su uso. AWS KMS recomienda que las claves externas estén preparadas para gestionar hasta 1800 solicitudes por segundo.

Creación de una clave KMS en un almacén de claves externo

Puede crear una nueva clave de KMS en el almacén de claves externo de la AWS KMS consola o mediante la [CreateKey](#) operación.

Uso de la AWS KMS consola

Hay dos formas de crear una clave de KMS en un almacén de claves externo.

- Método 1 (recomendado): elija un almacén de claves externo y, a continuación, cree una clave de KMS en ese almacén de claves externo.
- Método 2: cree una clave de KMS y, a continuación, indique que está en un almacén de claves externo.

Si usa el método 1, en el que elige el almacén de claves externo antes de crear la clave, AWS KMS elige automáticamente todas las propiedades de clave de KMS requeridas y completa el ID del almacén de claves externo. Este método evita los errores que puede cometer al crear la clave de KMS.

 Note

No incluya información confidencial en el alias, la descripción ni las etiquetas. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

Método 1 (recomendado): comience en su almacén de claves externo

Para usar este método, elija su almacén de claves externo y, a continuación, cree una clave de KMS. La AWS KMS consola selecciona automáticamente todas las propiedades necesarias y rellena el identificador del almacén de claves externo. Este método evita muchos de los errores que puede cometer al crear la clave de KMS.

1. Inicia sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrela en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Custom key stores (Almacenes de claves personalizados), External key stores (Almacenes de claves externos).
4. Elija el nombre del almacén de claves externo.
5. En la esquina superior derecha, seleccione Create a KMS key in this key store (Crear una clave de KMS en este almacén de claves).

Si el almacén de claves externo no está conectado, se le pedirá que lo conecte. Si el intento de conexión falla, debe resolver el problema y conectar el almacén de claves externo para poder crear una nueva claves de KMS en él.

Si el almacén de claves externo está conectado, se redirigirá a la página de Customer managed keys (Claves gestionadas por el cliente) para crear una clave. Los valores de Configuración de claves requeridos ya están seleccionados. Además, se rellena el identificador del almacén de claves personalizado de su almacén de claves externo, aunque puede cambiarlo.

6. Introduzca el identificador de clave de una [clave externa](#) en su [administrador de claves externo](#). Esta clave externa debe [cumplir los requisitos](#) para su uso con una clave de KMS. No puede cambiar este valor después de crear la clave.

Si la clave externa tiene varias IDs, introduzca el ID de clave que el proxy del almacén de claves externo utiliza para identificar la clave externa.

7. Confirme que va a crear una clave de KMS en el almacén de claves externo especificado.
8. Elija Siguiente.

El resto de este procedimiento es el mismo que [para crear una clave de KMS estándar](#).

9. Escriba un alias (requerido) y, una descripción (opcional) para la clave de KMS.
10. (Opcional). En la página agregar etiquetas, añada etiquetas que identifiquen o categoricen la clave KMS.

Al añadir etiquetas a AWS los recursos, AWS genera un informe de asignación de costes con el uso y los costes agregados por etiquetas. Las etiquetas también pueden utilizarse para controlar el acceso a una clave KMS. Para obtener información acerca del etiquetado de claves KMS, consulte [Etiquetas en AWS KMS](#) y [ABAC para AWS KMS](#).

11. Elija Siguiente.
12. En la sección administradores de claves, seleccione los usuarios y roles de IAM que pueden administrar la clave KMS. Para obtener más información, consulte [Permite que los administradores de claves administren la clave de KMS](#).

 Note

Las políticas de IAM pueden otorgar permisos para usar la clave KMS a otros usuarios y roles de IAM.

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;.

13. (Opcional) Para evitar que estos administradores de claves eliminen esta claves de KMS, desactive la casilla Allow key administrators to delete this key (Permitir que los administradores de claves eliminen esta clave).

Eliminar una clave de KMS es una operación destructiva e irreversible que puede hacer que el texto cifrado sea irrecuperable. No puede volver a crear una clave de KMS simétrica en un almacén de claves externo, aunque disponga del material de claves externas. Sin embargo, eliminar una clave de KMS no afecta a su clave externa asociada. Para obtener información sobre cómo eliminar una clave KMS de un almacén de claves externo, consulte [Consideraciones especiales para eliminar claves](#).

14. Elija Siguiente.

15. En la sección Esta cuenta, seleccione los usuarios y roles de IAM Cuenta de AWS que pueden usar la clave KMS en operaciones [criptográficas](#). Para obtener más información, consulte [Permite a los usuarios de claves utilizar la clave de KMS](#).

 Note

Las políticas de IAM pueden otorgar permisos para usar la clave KMS a otros usuarios y roles de IAM.

Las prácticas recomendadas de IAM desalientan el uso de usuarios de IAM con credenciales a largo plazo. Siempre que sea posible, utilice los roles de IAM, que proporcionan credenciales temporales. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM;.

16. (Opcional) Puede permitir que otros Cuentas de AWS usen esta clave KMS para operaciones criptográficas. Para ello, en la Cuentas de AWS sección Otros, en la parte inferior de la página, selecciona Añadir otra Cuenta de AWS e introduce el Cuenta de AWS ID de una cuenta externa. Para agregar varias cuentas externas, repita este paso.

 Note

Los administradores de la otra también Cuentas de AWS deben permitir el acceso a la clave de KMS mediante la creación de políticas de IAM para sus usuarios. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

17. Elija Siguiente.
18. Revise los ajustes de clave que ha elegido. Aún puede volver atrás y cambiar todos los ajustes.
19. Cuando haya acabado, elija Finish (Finalizar) para crear la clave.

Método 2: Comience con las claves gestionadas por el cliente

Este procedimiento es el mismo que el procedimiento para crear una clave de cifrado simétrica con material AWS KMS clave. Sin embargo, en este procedimiento, se especifica el ID del almacén de claves personalizado del almacén de claves externo y el ID de clave de la clave externa. También debe especificar los [valores de propiedad requeridos](#) para una clave de KMS en un almacén de claves externo, como la especificación de la clave y el uso de la clave.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Elija Crear clave.
5. Seleccione Symmetric (Simétrica).
6. En Key usage (Uso de claves), se selecciona la opción Encrypt and decrypt (Cifrar y descifrar) para usted. No la cambie.
7. Elija Advanced options (Opciones avanzadas).
8. En Key material origin (Origen del material de claves), elija External key store (Almacenes de claves externos).
9. Confirme que va a crear una clave de KMS en el almacén de claves externo especificado.
10. Elija Siguiente.
11. Elija la fila que representa el almacén de claves externo para la nueva clave de KMS.

No puede elegir un almacén de claves externo desconectado. Para conectar un almacén de claves que está desconectado, elija el nombre del almacén de claves y, a continuación, en Key store actions (Acciones del almacén de claves), elija Connect (Conectar). Para obtener más información, consulte [Uso de la consola AWS KMS](#).

12. Introduzca el identificador de clave de una [clave externa](#) en su [administrador de claves externo](#). Esta clave externa debe [cumplir los requisitos](#) para su uso con una clave de KMS. No puede cambiar este valor después de crear la clave.

Si la clave externa tiene varias IDs, introduzca el ID de clave que el proxy del almacén de claves externo utiliza para identificar la clave externa.

13. Elija Siguiente.

El resto de este procedimiento es el mismo que [para crear una clave de KMS estándar](#).

14. Escriba un alias y, si lo desea, una descripción para la clave KMS.
15. (Opcional). En la página agregar etiquetas, añada etiquetas que identifiquen o categoricen la clave KMS.

Al añadir etiquetas a AWS los recursos, AWS genera un informe de asignación de costes con el uso y los costes agregados por etiquetas. Las etiquetas también pueden utilizarse para controlar

el acceso a una clave KMS. Para obtener información acerca del etiquetado de claves KMS, consulte [Etiquetas en AWS KMS](#) y [ABAC para AWS KMS](#).

16. Elija Siguiente.
17. En la sección administradores de claves, seleccione los usuarios y roles de IAM que pueden administrar la clave KMS. Para obtener más información, consulte [Permite que los administradores de claves administren la clave de KMS](#).

 Note

Las políticas de IAM pueden otorgar permisos para usar la clave KMS a otros usuarios y roles de IAM.

18. (Opcional) Para evitar que estos administradores de claves eliminen esta claves de KMS, desactive la casilla Allow key administrators to delete this key (Permitir que los administradores de claves eliminen esta clave).

Eliminar una clave de KMS es una operación destructiva e irreversible que puede hacer que el texto cifrado sea irrecuperable. No puede volver a crear una clave de KMS simétrica en un almacén de claves externo, aunque disponga del material de claves externas. Sin embargo, eliminar una clave de KMS no afecta a su clave externa asociada. Para obtener información sobre cómo eliminar una clave de KMS de un almacén de claves externo, consulte [Eliminar un AWS KMS key](#).

19. Elija Siguiente.
20. En la sección Esta cuenta, seleccione los usuarios y roles de IAM Cuenta de AWS que pueden usar la clave KMS en operaciones [criptográficas](#). Para obtener más información, consulte [Permite a los usuarios de claves utilizar la clave de KMS](#).

 Note

Las políticas de IAM pueden otorgar permisos para usar la clave KMS a otros usuarios y roles de IAM.

21. (Opcional) Puede permitir que otros Cuentas de AWS usen esta clave KMS para operaciones criptográficas. Para ello, en la Cuentas de AWS sección Otros, en la parte inferior de la página, selecciona Añadir otra Cuenta de AWS e introduce el Cuenta de AWS ID de una cuenta externa. Para agregar varias cuentas externas, repita este paso.

 Note

Los administradores de la otra también Cuentas de AWS deben permitir el acceso a la clave de KMS mediante la creación de políticas de IAM para sus usuarios. Para obtener más información, consulte [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#).

22. Elija Siguiente.
23. Revise los ajustes de clave que ha elegido. Aún puede volver atrás y cambiar todos los ajustes.
24. Cuando haya acabado, elija Finish (Finalizar) para crear la clave.

Si el procedimiento se realiza correctamente, la pantalla mostrará la nueva clave de KMS en el almacén de claves externo de su elección. Al elegir el nombre o alias de la nueva clave de KMS, la pantalla Cryptographic configuration (Configuración criptográfica) de su página de detalles mostrará el origen de la clave de KMS (External key store [Almacén de claves externo]), el nombre, el ID y el tipo del almacén de claves personalizados, y el ID, el uso de claves y el estado de la clave externa. Si el procedimiento falla, aparecerá un mensaje de error que describe el motivo del error. Para , consulte [Solución de problemas de almacenes de claves externos](#).

 Tip

Para facilitar la identificación de las claves de KMS en un almacén de claves personalizado, en la página Customer managed keys (Claves administradas por el cliente), agregue la columna Origin (Origen) y Custom key store ID (ID del almacén de claves personalizado). Para cambiar los campos de la tabla, seleccione el icono de engranaje en la esquina superior derecha de la página. Para obtener más información, consulte [Personalización de la vista de la consola](#).

Uso de la API AWS KMS

Para crear una nueva clave de KMS en un almacén de claves externo, utilice la [CreateKey](#) operación. Se requieren los siguientes parámetros:

- El valor `Origin` debe ser `EXTERNAL_KEY_STORE`.

- El parámetro `CustomKeyStoreId` identifica el almacén de claves externo. El [ConnectionState](#) del almacén de claves externo especificado debe ser `CONNECTED`. Para encontrar el `CustomKeyStoreId` y `ConnectionState`, utilice la operación `DescribeCustomKeyStores`.
- El parámetro `XksKeyId` identifica las claves externas. Esta clave externa debe [cumplir con los requisitos](#) para la asociación con una clave de KMS.

También puede utilizar cualquiera de los parámetros opcionales de la operación `CreateKey`, como los parámetros `Policy` o [Tags](#) (Etiquetas).

Note

No incluya información confidencial en los campos `Description` o `Tags`. Estos campos pueden aparecer en texto plano en CloudTrail los registros y otros resultados.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Este comando de ejemplo utiliza la [CreateKey](#) operación para crear una clave KMS en un almacén de claves externo. La respuesta incluye las propiedades de las claves de KMS, el ID del almacén de claves externo y el ID, el uso y el estado de la clave externa.

Antes de ejecutar este comando, reemplace el ID del almacén de claves personalizado de ejemplo por un 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"
}
}
```

Identificación y visualización de claves

Puede usar [Consola de administración de AWS](#) la [API AWS Key Management Service \(AWS KMS\)](#) para ver AWS KMS keys en cada cuenta y región, incluidas las claves de KMS que administra y las claves de KMS que administra AWS.

Temas

- [Encontrar el ID de clave y el ARN de clave](#)
- [Acceso y enumeración de detalles de claves KMS](#)
- [Identificación de diferentes tipos de claves](#)
- [Personalización de la vista de la consola](#)
- [Encuentre las claves y el material clave de KMS en un almacén de AWS CloudHSM claves](#)

Encontrar el ID de clave y el ARN de clave

Para identificar un AWS KMS key, puede utilizar el [ID de clave](#) o el nombre del recurso de Amazon ([ARN clave](#)). En [operaciones criptográficas](#), también puede utilizar el [nombre de alias](#) o el [ARN de alias](#).

Puede usar la [AWS KMS consola](#) o la [ListKeys](#) operación para identificar el ID de clave y el ARN de clave de cada clave de KMS de su cuenta y región.

Para obtener información detallada sobre los identificadores clave de KMS compatibles AWS KMS, consulte. [Identificadores clave \(\) KeyId](#) Para obtener ayuda para buscar un nombre de alias y un ARN de alias, consulte [Búsqueda del nombre del alias y el ARN de alias para una clave de KMS](#).

Uso de AWS KMS consola

1. Abra la AWS KMS consola en. <https://console.aws.amazon.com/kms>
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. Si desea ver las claves de la cuenta que usted crea y administra, elija en el panel de navegación, Claves administradas por el cliente. Para ver las claves de la cuenta que se AWS crean y administran para usted, en el panel de navegación, seleccione las claves AWS administradas.

- Para buscar el [ID de clave](#) de una clave KMS, consulte la fila que empieza por el alias de clave KMS.

La columna Key ID (ID de clave) aparece en las tablas de forma predeterminada. Si la columna Key ID (ID de clave) no aparece en la tabla, utilice el procedimiento que se describe en [the section called “Personalización de la vista de la consola”](#) para restaurarla. También puede ver el ID de clave de una clave KMS en su página de detalles.

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 buscar el nombre de recurso de Amazon (ARN) de la clave KMS, elija el ID o alias de la clave. El [ARN de clave](#) aparece en la sección General Configuration (Configuración general).

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	

Uso de AWS KMS API

Para encontrar el [ID de clave y el ARN](#) de clave de un AWS KMS key, utilice la [ListKeys](#) operación.

La operación ListKeys devuelve el ID de clave y el nombre de recurso de Amazon (ARN) de todas las claves KMS de la cuenta y la región de la persona que llama.

Por ejemplo, esta llamada a la operación ListKeys devuelve el ID y el ARN de todas las claves KMS de esta cuenta ficticia. Para ver ejemplos en varios lenguajes de programación, consulte [Úselo ListKeys con un AWS SDK o CLI](#).

```
$ aws kms list-keys
{
```

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

Acceso y enumeración de detalles de claves KMS

Puede usar la [AWS KMS consola](#) o la [DescribeKey](#) operación para acceder a información detallada sobre las claves de KMS de la cuenta y la región y enumerarla.

Los siguientes procedimientos muestran cómo acceder a los detalles de claves KMS, como el ID de la clave, las especificaciones de la clave o el uso de la clave, entre otros.

Uso de AWS KMS consola

En la página de detalles de cada clave KMS se muestran las propiedades de la clave KMS. Difiere ligeramente de los diferentes tipos de claves KMS.

Para ver información detallada sobre una clave de KMS, en la página Claves administradas por AWS o de claves administradas por el cliente, elija el alias o el ID de clave de la clave de KMS.

La página de detalles de una clave KMS incluye una Configuración general en la que se muestran las propiedades básicas de la clave KMS. También incluye pestañas en las que puede ver y editar las propiedades de la clave de KMS, como la política de claves, la configuración criptográfica, las etiquetas, el material y las rotaciones de las claves (en el caso de las claves de KMS que admiten la rotación automática o bajo demanda), la regionalidad (en el caso de las claves de varias regiones) y la clave pública (en el caso de las claves de KMS asimétricas).

Note

La AWS KMS consola muestra las claves KMS que tiene [permiso para ver](#) en su cuenta y región. Las claves de KMS de otras Cuentas de AWS no aparecen en la consola, incluso

si tiene permiso para verlas, administrarlas y usarlas. Para ver las claves de KMS de otras cuentas, utilice la [DescribeKey](#) operación.

Navegar a la página de detalles de clave de una clave KMS.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. Si desea ver las claves de la cuenta que usted crea y administra, elija en el panel de navegación, Claves administradas por el cliente. Para ver las claves de la cuenta que se AWS crean y administran para usted, en el panel de navegación, seleccione las claves AWS administradas.
4. Para abrir la página de detalles de clave, elija el ID de clave o el alias de la clave KMS.

Si la clave KMS tiene varios alias, un resumen de alias (+n más) aparece junto al nombre de uno de los alias. Elegir el resumen de alias le llevará directamente a la pestaña Alias en la página de detalles de clave.

KMS > Customer managed keys > Key ID: 2g6838ee-b221-409f-70ae-f70b81293326h

2g6838ee-b221-409f-70ae-f70b81293326h

Key actions Edit

General configuration

Alias
kms

ARN
arn:aws:kms:us-east-1:111222333444:key/2g6838ee-b221-409f-70ae-f70b81293326h

Current key material ID
0182670028d107a245d79f3bgf110cb02568ad5e7789e0748917ac3bf764e14g

Status
Enabled

Description
-

Creation date
Mar 27, 2026 11:22 PDT

Last used
10 days ago

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

En la siguiente lista se describen los campos de la visualización detallada, incluido el campo de las pestañas. Algunos de estos campos también están disponibles como columnas en la visualización de tabla.

Acceso y enumeración de detalles de claves KMS

397

Alias

Dónde: Pestañas de Alias

Un nombre fácil de recordar para la clave KMS. Puede usar un alias para identificar la clave KMS en la consola y en algunas AWS KMS API. Para obtener más información, consulte [Alias en AWS KMS](#).

La pestaña Alias muestra todos los alias asociados a la clave de KMS en la región Cuenta de AWS y.

ARN

Dónde: sección de configuración general

El Nombre de recurso de Amazon (ARN) de la clave de cifrado. Este valor únicamente identifica la clave KMS. Puede utilizarlo para identificar la clave KMS en las operaciones de la API de AWS KMS .

Estado de la conexión

Indica si un [almacén de claves personalizado](#) está conectado a su almacén de claves de respaldo. Este campo aparece únicamente cuando la clave de KMS se crea en un almacén de claves personalizado.

Para obtener información sobre los valores de este campo, consulte la referencia [ConnectionState](#) de la AWS KMS API.

Fecha de creación

Dónde: sección de configuración general

La fecha y hora en que se creó la clave KMS. Este valor se muestra en la hora local del dispositivo. La zona horaria no varía en función de la región.

A diferencia de Expiration (Vencimiento), la creación se refiere únicamente a la clave KMS, no a su material de claves.

ID del clúster de CloudHSM

Dónde: pestaña de Configuración criptográfica

El ID de clúster del AWS CloudHSM clúster que contiene el material clave de la clave KMS. Este campo aparece únicamente cuando la clave de KMS se crea en un [almacén de claves personalizado](#).

Si elige el ID de clúster de CloudHSM, se abre la página Clústeres en la consola. AWS CloudHSM

Material de claves actual

Dónde: sección de configuración general

Las claves de cifrado simétricas con origen AWS_KMS admiten la rotación automática y bajo demanda. Las claves de cifrado simétricas con EXTERNAL origen admiten la rotación bajo demanda. Estas claves pueden tener varios materiales de claves asociados a la clave. El material de claves rotado más recientemente se puede utilizar tanto para el cifrado como para el descifrado. Este material de claves se identifica como el material de claves actual. Otros materiales de claves solo pueden usarse para el descifrado. La rotación automática o bajo demanda de una clave de KMS cambia su material de claves actual.

ID del almacén de claves personalizadas

Dónde: pestaña de Configuración criptográfica

ID del [almacén de claves personalizadas](#) que contiene la clave KMS. Este campo aparece únicamente cuando la clave de KMS se crea en un almacén de claves personalizado.

Si eliges el ID del almacén de claves personalizado, se abre la página de almacenes de claves personalizados en la AWS KMS consola.

Nombre del almacén de claves personalizadas

Dónde: pestaña de Configuración criptográfica

El nombre del [almacén de claves personalizadas](#) que contiene la clave KMS. Este campo aparece únicamente cuando la clave de KMS se crea en un almacén de claves personalizado.

Tipo de almacén de claves personalizado

Dónde: pestaña de Configuración criptográfica

Indica si el almacén de claves personalizado es un [almacén de claves de AWS CloudHSM](#) o un [almacén de claves externo](#). Este campo aparece únicamente cuando la clave de KMS se crea en un [almacén de claves personalizado](#).

Description (Descripción)

Dónde: sección de configuración general

Una descripción breve y opcional de la clave KMS que puede escribir y editar. Para agregar o actualizar la descripción de una clave administrada por el cliente, sobre General Configuration (Configuración general), seleccione Edit (Editar).

Algoritmos de cifrado

Dónde: pestaña de Configuración criptográfica

Enumera los algoritmos de cifrado que se pueden utilizar con la clave KMS en AWS KMS. Este campo aparece únicamente cuando el Key type (Tipo de clave) es Asymmetric (Asimétrico) y cuando el Key usage (Uso de la clave) es Encrypt and decrypt (Cifrar y descifrar). Para obtener información sobre los algoritmos de cifrado AWS KMS compatibles, consulte [Especificación de clave SYMMETRIC_DEFAULT](#) y [Especificaciones de clave de RSA para el cifrado y el descifrado](#).

Fecha de vencimiento

Dónde: pestaña de Material de clave

Fecha y hora en la que el material de claves de la clave KMS vence. Este campo aparece únicamente en las claves KMS con [material de claves importado](#), es decir, cuando el Origin (Origen) es External (Externo) y la clave KMS contiene material de claves que vence. Las claves de cifrado simétricas pueden tener varios materiales clave asociados. Para este tipo de claves, este campo indica la fecha y hora más tempranas en que caduca uno de los materiales de claves asociados.

ID de clave externa

Dónde: pestaña de Configuración criptográfica

El ID de la [clave externa](#) que está asociado a una clave de KMS en un [almacén de claves externo](#). Este campo solo aparece para las claves de KMS de un almacén de claves externo.

Estado de clave externa

Dónde: pestaña de Configuración criptográfica

El estado más reciente que el [proxy del almacén de claves externo](#) informó para la [clave externa](#) asociada a la clave de KMS. Este campo solo aparece para las claves de KMS de un almacén de claves externo.

Uso de clave externa

Dónde: pestaña de Configuración criptográfica

Las operaciones criptográficas que están habilitadas en la [clave externa](#) asociada a la clave de KMS. Este campo solo aparece para las claves de KMS de un almacén de claves externo.

Utilizado por última vez

Dónde: sección de configuración general

La última vez que se utilizó la clave KMS en una operación criptográfica, se mostró como un valor relativo, por ejemplo, hace 5 días. Este valor solo cubre el uso desde que se AWS KMS comenzó a rastrear el uso de esta clave. Pase el ratón sobre el valor para ver la marca horaria exacta, la fecha en la que se AWS KMS empezó a rastrear el uso, la AWS KMS operación que se realizó y un enlace al evento asociado. AWS CloudTrail

[Este campo solo rastrea las operaciones criptográficas.](#) Non-cryptographic operaciones como las que se [GetKeyPolicy](#) rastrean [DescribeKey](#) no se rastrean.

Política de claves

Dónde: pestaña de Política de clave

Controla el acceso a la clave KMS junto con las [políticas de IAM](#) y las [concesiones](#). Cada clave KMS tiene una política de claves. Es el único elemento de autorización obligatorio. Para cambiar la política de claves de una clave administrada por el cliente, en la pestaña Key policy (Política de claves), seleccione Edit (Editar). Para obtener más información, consulte [the section called “Políticas de claves”](#).

Rotación y material de claves

Dónde: pestaña Rotación y material de claves

Esta pestaña solo aparece para las claves de cifrado simétricas con origen AWS_KMS (que admiten la rotación automática y bajo demanda), así como para las claves de cifrado simétricas de una sola región con origen EXTERNAL (que admiten la rotación bajo demanda).

La pestaña tiene tres paneles:

Rotación automática: habilita y deshabilita la [rotación automática](#) del material de claves en una [clave de KMS administrada por el cliente](#). Para cambiar el estado de la rotación de claves de una [clave administrada por el cliente](#), utilice la casilla de verificación. No puede habilitar ni desactivar la rotación del material de claves en una [Clave administrada de AWS](#). Las Claves administradas por AWS rotan automáticamente cada año.

On-demand rotación: inicie una [rotación bajo demanda](#) del material clave de una [clave gestionada por el cliente](#). En el caso de las claves importadas, debe haber ya un material de claves importado en estado PENDING_ROTATION para que la opción Rotar ahora esté disponible.

Materiales de claves: enumera todos los materiales de clave asociados a la clave de KMS. Cada material de claves tiene un identificador único y su fila muestra información adicional sobre el material de claves, como la fecha de rotación cuando el material de claves estuvo disponible para su uso en KMS. En el caso de las claves importadas, cada fila también tiene un menú de acciones que se puede utilizar para eliminar un material de claves específico o volver a importarlo a la clave de KMS.

Especificación de clave

Dónde: pestaña de Configuración criptográfica

El tipo de material clave de la clave KMS. AWS KMS admite claves KMS de cifrado simétrico (SYMMETRIC_DEFAULT), claves HMAC KMS de diferentes longitudes, claves KMS para claves RSA de diferentes longitudes y claves de curva elíptica con diferentes curvas. Para obtener más información, consulte [Key spec](#).

Tipo de clave

Dónde: pestaña de Configuración criptográfica

Indica si la clave KMS es Symmetric (Simétrica) o Asymmetric (Asimétrica).

Uso de clave

Dónde: pestaña de Configuración criptográfica

Indica si una clave KMS se puede utilizar para Encrypt and decrypt (Cifrar y descifrar), Sign and verify (Firmar y verificar) o Generate and verify MAC (Generar y verificar MAC). Para obtener más información, consulte [Key usage](#).

Origen

Dónde: pestaña de Configuración criptográfica

El origen del material de claves de la clave KMS. Los valores válidos son los siguientes:

- AWS KMS para material de claves que genera AWS KMS
- AWS CloudHSM para claves de KMS en el [almacén de claves de AWS CloudHSM](#)
- Externo para [material de claves importado](#) (BYOK)

- Almacén de claves externo para claves de KMS en un [almacén de claves externo](#)

Algoritmos de MAC

Dónde: pestaña de Configuración criptográfica

Enumera los algoritmos MAC que se pueden utilizar con una clave KMS HMAC en AWS KMS. Este campo aparece únicamente cuando la Especificación de la clave es una especificación de la clave HMAC (HMAC_*). Para obtener información acerca de los algoritmos MAC de firma que admite AWS KMS, consulte [Especificaciones de la clave para las claves KMS HMAC](#).

Clave principal

Dónde: pestaña de Regionalidad

Indica que esta clave KMS es una [clave principal de varias regiones](#). Los usuarios autorizados pueden utilizar esta sección para [cambiar la clave principal](#) a una clave de varias regiones relacionada diferente. Este campo aparece únicamente cuando la clave KMS es una clave principal de varias regiones.

Clave pública

Dónde: pestaña de Clave pública

Muestra la clave pública de una clave KMS asimétrica. Los usuarios autorizados pueden utilizar esta pestaña para [copiar y descargar la clave pública](#).

Regionalidad

Dónde: pestañas Sección de configuración general y Regionalidad

Indica si una clave KMS es una clave de una sola región, una [clave principal de varias regiones](#) o una [clave de réplica de varias regiones](#). Este campo aparece únicamente cuando la clave KMS es una clave de varias regiones.

Claves de varias regiones relacionadas

Dónde: pestaña de Regionalidad

Muestra todas las [claves de réplica y primaria de varias regiones](#) relacionadas, excepto la clave KMS actual. Este campo aparece únicamente cuando la clave KMS es una clave de varias regiones.

En la sección Claves de varias regiones relacionadas de una clave principal, los usuarios autorizados pueden [crear nuevas claves de réplica](#).

Clave de réplica

Dónde: pestaña de Regionalidad

Indica que esta clave KMS es una [clave de réplica de varias regiones](#). Este campo aparece únicamente cuando la clave KMS es una clave de réplica de varias regiones.

Algoritmos de firma

Dónde: pestaña de Configuración criptográfica

Enumera los algoritmos de firma que se pueden utilizar con la clave KMS en AWS KMS. Este campo aparece únicamente cuando el Key type (Tipo de clave) es Asymmetric (Asimétrico) y cuando el Key usage (Uso de la clave) es Sign and verify (Firmar y verificar). Para obtener información sobre los algoritmos de firma compatibles, consulte y. AWS KMS [Especificaciones de clave de RSA para la firma y la verificación](#) [Especificaciones de clave de curva elíptica](#)

Status

Dónde: sección de configuración general

El estado de la clave KMS. Puede utilizar la clave KMS en [operaciones criptográficas](#) solo cuando el estado es Enabled (Habilitado). Para obtener una descripción detallada de cada estado de clave KMS y su impacto en las operaciones que puede ejecutar en la clave KMS, consulte [Estados clave de AWS KMS claves](#).

Tags

Dónde: pestaña Etiquetas

Pares clave-valor opcionales que describen la clave KMS. Para agregar o cambiar las etiquetas de una clave KMS, en la pestaña Tags (Etiquetas), seleccione Edit (Editar).

Al añadir etiquetas a AWS los recursos, AWS genera un informe de asignación de costes con el uso y los costes agregados por etiquetas. Las etiquetas también pueden utilizarse para controlar el acceso a una clave KMS. Para obtener información acerca del etiquetado de claves KMS, consulte [Etiquetas en AWS KMS](#) y [ABAC para AWS KMS](#).

Uso de AWS KMS API

La [DescribeKey](#) operación devuelve detalles sobre la clave KMS especificada. Para identificar la clave KMS, utilice el [ID de la clave](#), [ARN de la clave](#), [nombre de alias](#) o [ARN del alias](#).

A diferencia de la [ListKeys](#) operación, que muestra solo las claves KMS de la cuenta y región de la persona que llama, los usuarios autorizados pueden usar la `DescribeKey` operación para obtener detalles sobre las claves KMS de otras cuentas.

 Note

La respuesta `DescribeKey` incluye tanto `KeySpec` como `CustomerMasterKeySpec` con los mismos valores. Este miembro `CustomerMasterKeySpec` está obsoleto.

Por ejemplo, esta llamada a `DescribeKey` devuelve información acerca de una clave KMS de cifrado simétrica. Los campos de la respuesta varían según la [especificación de la AWS KMS key](#), el [estado de la clave](#) y el [origen del material de la clave](#). Para ver ejemplos en varios lenguajes de programación, consulte [Úselo DescribeKey con un AWS SDK o CLI](#).

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

{
  "KeyMetadata": {
    "Origin": "AWS_KMS",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Description": "",
    "KeyManager": "CUSTOMER",
    "Enabled": true,
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "KeyState": "Enabled",
    "CreationDate": 1499988169.234,
    "MultiRegion": false,
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "CurrentKeyMaterialId":
      "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  }
}
```

Este ejemplo llama a la operación `DescribeKey` en una clave KMS asimétrica utilizada para la firma y la verificación. La respuesta incluye los algoritmos de firma que AWS KMS admite para esta clave 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"
    ]
  }
}
```

Identificación de diferentes tipos de claves

En los temas siguientes se explica cómo identificar los distintos tipos de claves en la AWS KMS consola y [DescribeKey](#) las respuestas.

Para obtener ayuda para acceder a la pestaña de configuración criptográfica de la página de detalles de una clave KMS, consulte [the section called “Acceso y enumeración de detalles de claves KMS”](#).

Temas

- [Identificación de claves KMS asimétricas](#)
- [Identificación de claves KMS HMAC](#)
- [Identificación de claves KMS de varias regiones](#)

- [Identificación de claves KMS con material de claves importado](#)
- [Identifique las claves de KMS en AWS CloudHSM almacenes de claves](#)
- [Identificación de claves KMS en almacenes de claves externos](#)

Identificación de claves KMS asimétricas

En la AWS KMS consola

Las columnas Key type (Tipo de clave) de la tabla Customer managed keys (Claves administradas por el cliente) muestran si cada clave KMS es simétrica o asimétrica. Puede filtrar la tabla por el valor de Tipo de clave para mostrar solo las claves de KMS asimétricas. Para obtener más información, consulte [the section called “Ordenar y filtrar las claves KMS”](#).

La pestaña Cryptographic configuration (Configuración criptográfica) en la página de detalles para una clave KMS muestra el Key type (Tipo de clave), que indica si la clave es simétrica o asimétrica. También muestra Key Usage (Uso de claves), que indica si la clave KMS asimétrica se utiliza para cifrar y descifrar, firmar y verificar o para obtener secretos compartidos.

En DescribeKey las respuestas

Cuando se llama a la operación DescribeKey en una clave KMS asimétrica, la respuesta incluye los valores de KeySpec y KeyUsage, que se pueden usar para determinar si una clave KMS es simétrica o asimétrica.

Si el valor de KeySpec es SYMMETRIC_DEFAULT, la clave es una clave KMS de cifrado simétrica. Para obtener más información sobre las especificaciones de las claves asimétricas, consulte [Referencia de especificaciones de clave](#).

Si el valor de KeyUsage es SIGN_VERIFY o KEY_AGREEMENT, la clave es una clave KMS asimétrica.

La operación DescribeKey también devuelve los siguientes detalles para claves KMS asimétricas.

- En el caso de las claves KMS asimétricas con un valor KeyUsage de ENCRYPT_DECRYPT, la operación devuelve EncryptionAlgorithms, que muestra los algoritmos de cifrado válidos para la clave.
- En el caso de las claves KMS asimétricas con un valor KeyUsage de SIGN_VERIFY, la operación devuelve SigningAlgorithms, que muestra los algoritmos de firma válidos para la clave.

- En el caso de las claves KMS asimétricas con un valor KeyUsage de KEY_AGREEMENT, la operación devuelve KeyAgreementAlgorithms, que muestra los algoritmos de acuerdo de claves válidos para la clave.

Para obtener más información sobre claves KMS, consulte [the section called “Claves asimétricas”](#).

Identificación de claves KMS HMAC

En la AWS KMS consola

Las claves de KMS HMAC se incluyen en la tabla Claves administradas por el cliente, pero no puede ordenar ni filtrar esta tabla por las especificaciones de clave ni por valores de uso de claves que identifican las claves de HMAC. Para facilitar la búsqueda de las claves HMAC, asígneles un alias o una etiqueta distintiva. A continuación, puede ordenar o filtrar por el alias o la etiqueta.

La pestaña Cryptographic configuration (Configuración criptográfica) en la página de detalles para una clave KMS muestra el Key type (Tipo de clave), que indica si la clave es simétrica o asimétrica. Las claves KMS HMAC son simétricas. La pestaña Cryptographic configuration (Configuración criptográfica) también muestra Key Usage (Uso de claves). Para las claves KMS HMAC, Generar y verificar MAC es siempre el valor uso de clave.

En DescribeKey las respuestas

Cuando llama a la operación DescribeKey en una clave KMS HMAC, la respuesta incluye los valores KeySpec y KeyUsage. En el caso de las claves KMS HMAC, el valor de uso de clave es siempre GENERATE_VERIFY_MAC y el valor de la especificación de clave siempre empieza por HMAC_.

Para obtener más información sobre claves KMS HMAC, consulte [the section called “Claves HMAC”](#).

Identificación de claves KMS de varias regiones

En la AWS KMS consola

La tabla Customer managed keys (Claves administradas por clientes) solo muestra las claves KMS en la región seleccionada. Puede ver las claves principales y de réplica de varias regiones en la región seleccionada. Para cambiar la AWS región, usa el selector de regiones en la esquina superior derecha de la consola.

Para facilitar la identificación de las claves de varias regiones en la tabla Claves administradas por el cliente, añada la columna Regionalidad a la tabla. Para obtener ayuda, consulte [the section called “Personalización de las tablas de claves KMS”](#).

La página de detalles de las claves KMS de varias regiones incluye una pestaña Regionality (Regionalidad). La pestaña Regionality (Regionalidad) de una clave principal incluye los botones Change primary Region (Cambiar región principal) y Create new replica keys (Crear nuevas claves de réplica). (La pestaña Regionality (Regionalidad) de una clave de réplica no tiene ningún botón). La sección Related multi-Region keys (Claves de varias regiones relacionadas) enumera todas las claves de varias regiones relacionadas con la actual. Si la clave actual es una clave de réplica, esta lista incluye la clave principal.

Si selecciona una clave multirregional relacionada de la tabla de claves multirregionales relacionadas, la AWS KMS consola cambiará a la región de la clave seleccionada y abrirá la página de detalles de la clave. Por ejemplo, si elige la clave de réplica de la sa-east-1 región en la sección de claves multirregionales relacionadas que aparece a continuación, la AWS KMS consola cambiará a la sa-east-1 región para mostrar la página de detalles de esa clave de réplica. Puede hacer esto para ver el alias o la política de clave de la clave de réplica. Para cambiar la región nuevamente, utilice el selector de regiones en la esquina superior derecha de la página.

En DescribeKey las respuestas

De forma predeterminada, las operaciones de la AWS KMS API son regionales y solo devuelven los recursos de la región actual o especificada. Sin embargo, cuando se llama a la DescribeKey operación en una clave KMS multirregional, la respuesta incluye todas las claves multirregionales relacionadas de otras AWS regiones del MultiRegionConfiguration elemento.

Para obtener más información sobre claves KMS de varias regiones, consulte [the section called “Multi-Region llaves”](#).

Identificación de claves KMS con material de claves importado

En la consola AWS KMS

Para facilitar la identificación de las claves KMS con material de claves importado en la tabla Customer managed keys (Claves administradas por el cliente), añada la columna Origin (Origen) a la tabla. La columna Origin facilita la identificación de las claves de KMS que tienen un valor

de propiedad de origen Externo (Importar material de claves). Para obtener ayuda, consulte [the section called “Personalización de las tablas de claves KMS”](#).

La pestaña Cryptographic configuration (Configuración criptográfica) de la página de detalles de una clave KMS muestra Origin (Origen), que identifica el origen del material de claves de la clave KMS. En el caso de las claves KMS con material de claves importado, el valor de origen es siempre External (Import Key material) [Externo (material de claves de importación)]. La página de detalles también incluye una pestaña Key material (Material de claves) que proporciona información detallada sobre el material de claves importado. Las claves de cifrado simétrico con EXTERNAL origen admiten rotaciones bajo demanda y pueden tener varios materiales clave asociados. Para este tipo de claves, la pestaña lleva la etiqueta Rotaciones y material de claves.

En las respuestas DescribeKey

Cuando se llama a la operación DescribeKey en una clave KMS con material de claves importado, la respuesta incluye los valores Origin, ExpirationModel y ValidTo. En el caso de las claves KMS con material de claves importado, el valor de origen es siempre EXTERNAL. El valor ExpirationModel indica si el material de claves está configurado para que caduque o no, y el valor ValidTo indica cuándo caducará el material de claves. Cuando hay varios materiales de claves asociados a una clave, el valor ValidTo indica la fecha de caducidad más temprana de todos los materiales de claves (excepto el que está pendiente de rotación) y ExpirationModel se establece en DOES_NOT_EXPIRE solo si ninguno de estos materiales de claves está programado para caducar. Para obtener más información, consulte [Configuración de una fecha de vencimiento \(opcional\)](#).

Para obtener información sobre las claves KMS con material de claves importado, consulte [the section called “Material de claves importado”](#).

Identifique las claves de KMS en AWS CloudHSM almacenes de claves

En la AWS KMS consola

Para facilitar la identificación de las claves de KMS en AWS CloudHSM los almacenes de claves de la tabla de claves gestionadas por el cliente, añada la columna Origin a la tabla. La columna Origen facilita la identificación de las claves de KMS que tienen un valor de propiedad de origen de AWS CloudHSM. Para obtener ayuda, consulte [the section called “Personalización de las tablas de claves KMS”](#).

La pestaña Configuración criptográfica de la página de detalles de una clave de KMS muestra Origen, que identifica el origen del material de claves de la clave de KMS. En AWS CloudHSM el caso de las claves KMS de los almacenes de claves, el valor de origen es siempre AWS CloudHSM.

En el caso de una clave KMS en un almacén de AWS CloudHSM claves, la pestaña de configuración criptográfica incluye una sección adicional, el almacén de claves personalizado, que proporciona información sobre el almacén de AWS CloudHSM claves y el AWS CloudHSM clúster asociados a la clave KMS.

En DescribeKey las respuestas

Cuando se llama a la operación DescribeKey con una clave KMS en un almacén de claves de AWS CloudHSM, la respuesta incluye Origin, que identifica el origen del material de claves. En el caso de las claves KMS de un almacén de AWS CloudHSM claves, el valor de origen es siempre AWS_CLOUDHSM. La operación también devuelve los siguientes campos especiales para las claves de KMS de los almacenes de AWS CloudHSM claves:

- CloudHsmClusterId
- CustomKeyStoreId

Para obtener más información sobre los almacenes de AWS CloudHSM claves, consulte [the section called “AWS CloudHSM tiendas clave”](#).

Identificación de claves KMS en almacenes de claves externos

En la AWS KMS consola

Para facilitar la identificación de las claves KMS en almacenes de claves externos en la tabla Customer managed keys (Claves administradas por el cliente), añade la columna Origin (Origen) a la tabla. La columna Origin (Origen) facilita la identificación de las claves KMS que tienen un valor de propiedad de origen External key store (Almacén de claves externo). Para obtener ayuda, consulte [the section called “Personalización de las tablas de claves KMS”](#).

La pestaña Configuración criptográfica de la página de detalles de una clave de KMS muestra Origen, que identifica el origen del material de claves de la clave de KMS. Por ejemplo, el valor de origen para todas las claves KMS en el almacén de claves externo es siempre External key store (Almacén de claves externo).

Para una clave KMS en un almacén de claves externo, la pestaña Cryptographic configuration (Configuración criptográfica) incluye dos secciones adicionales: Custom key store (Almacén de claves personalizado) y External key (Clave externa). La tabla Custom key store (Almacén de claves personalizado) proporciona información sobre el almacén de claves externo asociado a la clave KMS. La tabla External key (Clave externa) aparece en la consola de AWS KMS solo para las claves KMS de los almacenes de claves externos. Proporciona información sobre la clave externa asociada a la clave de KMS. La [clave externa](#) es una clave criptográfica externa AWS que sirve como material clave para la clave KMS en el almacén de claves externo. Cuando cifra o descifra con la clave de KMS, la operación la realiza su [administrador de claves externo](#) utilizando la clave externa especificada.

Los siguientes valores aparecen en la sección External key (Clave externa).

ID de clave externa

El identificador de la clave externa en su administrador de claves externo. Este es el valor que utiliza el proxy del almacén de claves externo para identificar la clave externa. Usted especifica el ID de la clave externa cuando crea la clave de KMS y no puede cambiarla. Si el valor del identificador de clave externa que utilizó para crear la clave de KMS cambia o deja de ser válida, debe [programar la eliminación de la clave de KMS](#) y [crear una nueva clave de KMS](#) con el valor de ID de clave externa correcto.

En DescribeKey las respuestas

Cuando se llama a la operación DescribeKey con una clave KMS en un almacén de claves externo, la respuesta incluye `Origin`, que identifica el origen del material de claves. En el caso de las claves KMS de un almacén de AWS CloudHSM claves, el valor de origen es siempre `EXTERNAL_KEY_STORE`. La operación también devuelve el elemento `CustomKeyStoreId`, que identifica el almacén de claves externo asociado a las claves de KMS.

Para obtener más información sobre los almacenes de claves, consulte [the section called “Almacenes de claves externos”](#).

Personalización de la vista de la consola

Puede personalizar la vista de la AWS KMS consola para facilitar la búsqueda de las claves de KMS. Personalice las tablas que aparecen en las páginas Claves administradas por AWS y Customer managed keys (Claves administradas por el cliente) para mostrar la información que más necesita, o bien ordene y filtre las claves KMS que aparecen en las tablas.

Temas

- [Ordenar y filtrar las claves KMS](#)
- [Personalización de las tablas de claves KMS](#)

Ordenar y filtrar las claves KMS

Para facilitar la búsqueda de las claves KMS en la consola, puede ordenarlas y filtrarlas.

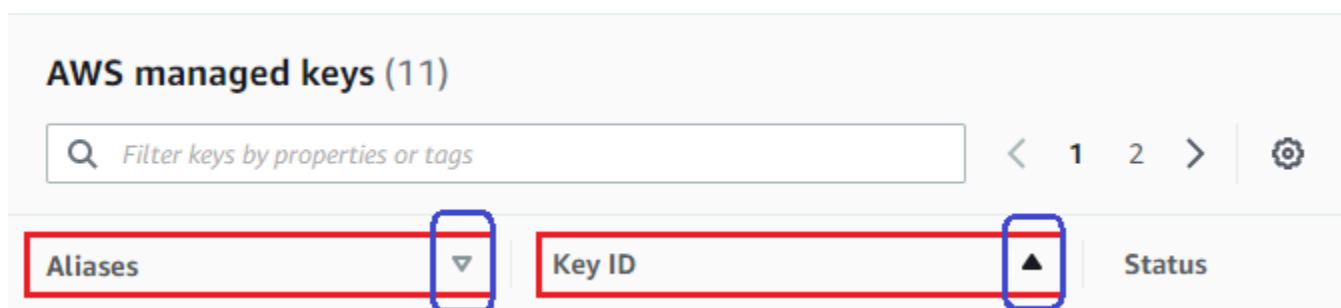
Ordenar

Puede ordenar las claves KMS administradas por el cliente en orden ascendente o descendente por sus valores de columna. Esta función ordena todas las claves KMS de la tabla, aunque no aparezcan en la página de la tabla actual.

Las columnas que se pueden ordenar aparecen indicadas con una flecha que aparece junto al nombre de la columna. En la página Claves administradas por AWS, puede ordenar por Alias o ID de clave. En la página Customer managed keys (Claves administradas por el cliente), puede ordenar por Alias, Key ID (ID de clave) o Key type (Tipo de clave).

Para ordenar en orden ascendente, seleccione el encabezado de la columna hasta que la flecha apunte hacia arriba. Para ordenar en orden descendente, seleccione el encabezado de la columna hasta que la flecha apunte hacia abajo. Puede ordenar solo por una columna cada vez.

Por ejemplo, puede ordenar las claves KMS en orden ascendente por ID de clave en lugar de por alias, que es el valor predeterminado.



Cuando ordena las claves KMS en la página Customer managed keys (Claves administradas por cliente) en orden ascendente por Key type (Tipo de clave), todas las claves asimétricas se muestran antes que todas las claves simétricas.

Filtro

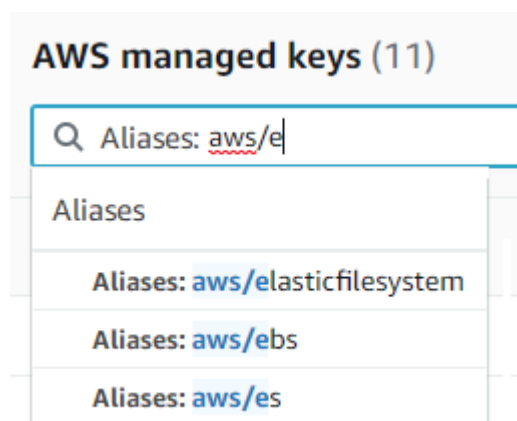
Puede filtrar las claves KMS por sus valores de propiedad o etiquetas. El filtro se aplica a todas las claves KMS de la tabla, aunque no aparezcan en la página de la tabla actual. El filtro no distingue entre mayúsculas y minúsculas.

Las propiedades que se pueden filtrar se enumeran en el cuadro de filtro. En la página Claves administradas por AWS , puede filtrar por alias e ID de clave. En la página Customer managed keys (Claves administradas por el cliente), puede filtrar por las propiedades de alias, ID de clave, tipo de clave y por etiquetas.

- En la página Claves administradas por AWS, puede filtrar por alias e ID de clave.
- En la página Customer managed keys (Claves administradas por el cliente), puede filtrar por etiquetas o por las propiedades de alias, ID de clave, tipo de clave o regionalidad.

Para filtrar por un valor de propiedad, elija el filtro, seleccione el nombre de propiedad y, a continuación, selecciónelo de la lista de valores de propiedad reales. Para filtrar por una etiqueta, elija la clave de etiqueta y, a continuación, elija de la lista de valores reales de etiqueta. Después de seleccionar una clave de propiedad o de etiqueta, también puede introducir una parte o el valor de la propiedad completo. Verá una vista previa de los resultados antes de tomar su decisión.

Por ejemplo, para mostrar las claves KMS con un nombre de alias que contenga aws/e, seleccione el cuadro de filtro, seleccione Alias, introduzca aws/e y, a continuación, pulse **Enter** o **Return** para agregar el filtro.



Filtros de tabla de claves KMS sugeridos

Filtrar por claves KMS asimétricas

Para mostrar solo las clave KMS asimétricas en la página Customer managed keys (Claves administradas por cliente), haga clic en el cuadro de filtro, elija Key type (Tipo de clave) y, a continuación, elija Key type: Asymmetric (Tipo de clave: Asimétrica). La opción Asymmetric (Asimétrica) solo aparece cuando tiene claves KMS asimétricas en la tabla.

Filtrar por claves de varias regiones

Para mostrar solo las claves multirregionales, en la página de claves gestionadas por el cliente, selecciona la casilla de filtro, selecciona Regionalidad y, a continuación, selecciona Regionalidad: Multi-Region La Multi-Region opción aparece solo cuando hay claves multirregionales en la tabla.

Filtrar por etiquetas

Para mostrar solo las claves KMS con una etiqueta concreta, elija el cuadro de filtro, elija la clave de etiqueta y, a continuación, elija entre los valores de etiqueta reales. También puede introducir una parte o el valor de la etiqueta completo.

La tabla resultante muestra todas las claves KMS con la etiqueta elegida. Sin embargo, no muestra la etiqueta. Para ver la etiqueta, elija el ID de clave o alias de la clave KMS y, en su página de detalles, elija la pestaña Tags (Etiquetas). Las pestañas aparecen debajo de la sección General configuration (Configuración general).

Este filtro requiere la clave y el valor de la etiqueta. No encontrará claves KMS escribiendo solo la clave de etiqueta o solo su valor. Para filtrar las etiquetas por la totalidad o parte de la clave o el valor de la etiqueta, utilice la [ListResourceTags](#) operación para obtener las claves de KMS etiquetadas y, a continuación, utilice las funciones de filtrado del lenguaje de programación.

Filtrar por texto

Para buscar texto, introduzca una parte o todo el alias, el ID de clave, el tipo de clave o la clave de etiqueta en el cuadro de filtro. (Después de seleccionar la clave de etiqueta, puede buscar un valor de etiqueta). Verá una vista previa de los resultados antes de tomar su decisión.

Por ejemplo, para mostrar claves KMS con test en sus claves de etiqueta o propiedades filtrables, escriba test en el cuadro de filtro. La vista previa muestra las claves KMS que seleccionará el filtro. En este caso, test solo aparece en la propiedad Alias.

Personalización de las tablas de claves KMS

Puede personalizar las tablas que aparecen en las páginas de claves administradas por el cliente Claves administradas por AWS y las que aparecen en ellas Consola de administración de AWS para adaptarlas a sus necesidades. Puede elegir las columnas de la tabla, el número de columnas de cada AWS KMS keys página (tamaño de página) y el formato del texto. La configuración que seleccione se guarda cuando la confirma y se vuelve a aplicar cuando abre las páginas.

Personalización de las tablas clave KMS

1. En la página Claves administradas por AWS o Claves administradas por el cliente, elija el icono de configuración



en la parte superior derecha de la página.

2. En la página Preferences (Preferencias), elija su configuración preferida y, a continuación, Confirm (Confirmar).

Considere utilizar la configuración Page size (Tamaño de página) para aumentar el número de clave KMS que se muestran en cada página, especialmente si suele utilizar un dispositivo que es fácil de desplazar.

Las columnas de datos que muestra pueden variar en función de la tabla, el rol de trabajo y los tipos de claves KMS de la cuenta y la región. En la siguiente tabla aparecen algunas sugerencias de configuración. Para obtener descripciones de las columnas, consulte [Uso de AWS KMS consola](#).

Configuraciones de tabla clave KMS sugeridas

Puede personalizar las columnas que aparecen en la tabla de claves KMS para mostrar la información que necesita sobre sus claves KMS.

Claves administradas por AWS

De forma predeterminada, la tabla Clave administrada de AWS muestra las columnas Alias, ID de clave y Estado. Estas columnas son ideales para la mayoría de casos de uso.

Claves KMS de cifrado simétricas

Si utiliza únicamente las claves de KMS de cifrado simétricas con el material de claves que genera AWS KMS, es probable que las columnas Alias, ID de clave, Estado y Fecha de creación sean las más útiles.

Claves KMS asimétricas

Si utiliza claves KMS asimétricas, además de las columnas Alias, ID de clave y Estado, debe considerar la adición de las columnas Tipo de clave, Especificación de clave y Uso de clave. Estas columnas le mostrarán si una clave KMS es simétrica o asimétrica, el tipo de material de claves y si la clave KMS se puede utilizar para el cifrado o la firma.

Claves KMS HMAC

Si utiliza claves KMS HMAC, además de las columnas Alias, ID de clave y Estado, debe considerar la adición de las columnas Especificación de clave y Uso de clave. Estas columnas le mostrarán si una clave KMS es una clave HMAC. Como no puede ordenar las claves de KMS por especificación o uso de clave, utilice alias y etiquetas para identificar las claves HMAC y, a continuación, utilice las [funciones de filtrado](#) de la AWS KMS consola para filtrar por alias o etiquetas.

Material de claves importado

Si tiene claves KMS con [material de claves importado](#), considere agregar las columnas Origen (Origen) y Expiration date (Fecha de vencimiento). Estas columnas le mostrarán si el material clave de una clave de KMS se ha importado o generado antes de que caduque el material clave, si es que caduca, si es que caduca. AWS KMS El campo Creation date (Fecha de creación) muestra la fecha en la que se creó la clave KMS (sin el material de claves). No refleja ninguna característica del material de claves.

Claves de los almacenes de claves personalizadas

Si tiene claves de KMS en [almacenes de claves personalizados](#), considere agregar las columnas Origen (Origen) y Custom key store ID (ID del almacén de claves personalizado). Estas columnas muestran que la clave de KMS se encuentra en un almacén de claves personalizado, indican qué tipo de almacén es y lo identifican.

Multi-Region claves

Si tiene [claves para varias regiones](#), considere agregar la columna Regionality (Regionalidad). Esto muestra si una clave KMS es una clave de una sola región, una [clave principal de varias regiones](#) o una [clave de réplica de varias regiones](#).

Encuentre las claves y el material clave de KMS en un almacén de AWS CloudHSM claves

Si administra un almacén de AWS CloudHSM claves, es posible que necesite identificar las claves de KMS en cada uno de los AWS CloudHSM. Por ejemplo, es probable que deba realizar algunas de las siguientes tareas.

- Realice un seguimiento de las claves de KMS en el almacén de AWS CloudHSM claves de AWS CloudTrail los registros.
- Prediga el efecto que tendrá en las claves de KMS la desconexión de un almacén de AWS CloudHSM claves.
- Programa la eliminación de las claves de KMS antes de eliminar un almacén de AWS CloudHSM claves.

Además, es posible que desee identificar las claves del AWS CloudHSM clúster que sirven como material clave para las claves de KMS. Aunque AWS KMS administra las claves de KMS y el material clave, usted sigue teniendo el control y la responsabilidad de la administración de su AWS CloudHSM clúster, así como de las copias de seguridad HSMs y las claves que contiene HSMs. Es posible que necesite identificar las claves para auditar el material clave, protegerlo de una eliminación accidental o eliminarlo de las copias de seguridad del clúster o de las copias de HSMs seguridad del clúster después de eliminar la clave de KMS.

Todo el material clave de las claves de KMS de tu almacén de AWS CloudHSM claves es propiedad del [usuario kmsuser criptográfico](#) (CU). AWS KMS establece el atributo de etiqueta clave, que solo se puede ver en AWS CloudHSM, en el nombre de recurso de Amazon (ARN) de la clave de KMS.

Para buscar las claves KMS y el material de claves, puede utilizar cualquiera de las técnicas siguientes.

- [Busque las claves KMS en un almacén de AWS CloudHSM claves](#)— Cómo identificar las claves KMS en uno o todos sus almacenes de AWS CloudHSM claves.
- [Busque todas las claves de un almacén de AWS CloudHSM claves](#): cómo encontrar todas las claves en su clúster que actúan como material de claves para las claves de KMS en su almacén de claves de AWS CloudHSM .

- [Busque la AWS CloudHSM clave de una clave KMS](#)— Cómo encontrar la clave en su clúster que sirve como material clave para una clave de KMS concreta en su almacén de AWS CloudHSM claves.
- [Busque la clave KMS de una AWS CloudHSM clave](#): Cómo encontrar la clave KMS para una clave concreta en su clúster.

Busque las claves KMS en un almacén de AWS CloudHSM claves

Si administra un almacén de AWS CloudHSM claves, es posible que necesite identificar las claves de KMS en cada uno AWS CloudHSM de ellos. Puede usar esta información para realizar un seguimiento de las operaciones de claves de KMS en AWS CloudTrail los registros, predecir el efecto de desconectar un almacén de claves personalizado en las claves de KMS o programar la eliminación de las claves de KMS antes de eliminar un almacén de AWS CloudHSM claves.

Para buscar las claves de KMS en un almacén de AWS CloudHSM claves (consola)

Para buscar las claves de KMS en un almacén de AWS CloudHSM claves concreto, en la página de claves administradas por el cliente, consulte los valores de los campos Nombre del almacén de claves personalizado o ID del almacén de claves personalizado. Para identificar las claves de KMS en cualquier almacén de AWS CloudHSM claves, busque las claves de KMS con un valor de origen de AWS CloudHSM. Para agregar columnas opcionales a la pantalla, elija el icono de engranaje en la esquina superior derecha de la página.

Para encontrar las claves de KMS en un almacén de AWS CloudHSM claves (API)

Para buscar las claves de KMS en un almacén de AWS CloudHSM claves, utilice las [DescribeKey](#) operaciones [ListKeys](#) y, a continuación, filtre por CustomKeyId valor. Antes de ejecutar los siguientes ejemplos, reemplace los valores ficticios del ID del almacén de claves personalizado por un ID válido.

Bash

Para encontrar las claves de KMS en un almacén de AWS CloudHSM claves concreto, obtén todas las claves de KMS de la cuenta y la región. A continuación, filtre por el ID del almacén de claves 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 obtener las claves de KMS en cualquier almacén de AWS CloudHSM claves de la cuenta y la región, busca CustomKeyType con un valor de AWS_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 buscar las claves de KMS en un almacén de AWS CloudHSM claves concreto, usa los KmsKey cmdlets Get KmsKeyList y Get para obtener todas las claves de KMS de la cuenta y la región.](#) A continuación, filtre por el ID del almacén de claves personalizado.

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyId -eq
'cks-1234567890abcdef0'
```

Para obtener las claves de KMS en cualquier almacén de AWS CloudHSM claves de la cuenta y la región, filtra por el CustomKeyType valor de. AWS_CLOUDHSM

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyType -eq 'AWS_CLOUDHSM'
```

Busque todas las claves de un almacén de AWS CloudHSM claves

Puede identificar las claves de su AWS CloudHSM clúster que sirven como material clave para su almacén de AWS CloudHSM claves. Para ello, utilice el comando [key list](#) en la CLI de CloudHSM.

También puede usar el comando key list para buscar la AWS KMS clave de una AWS CloudHSM clave. Cuando AWS KMS crea el material clave para una clave de KMS en el AWS CloudHSM clúster, escribe el nombre de recurso de Amazon (ARN) de la clave de KMS en la etiqueta de la clave. El comando key list devuelve el key-reference y el label.

Notas

Los siguientes procedimientos utilizan la herramienta de línea de comandos AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI de CloudHSM reemplaza key-handle por key-reference.

El 1 de enero de 2025, AWS CloudHSM dejará de ser compatible con las herramientas de línea de comandos Client SDK 3, la utilidad de administración CloudHSM (CMU) y la utilidad de administración de claves (KMU). Para obtener más información sobre las diferencias entre las herramientas de línea de comandos Client SDK 3 y Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) en la Guía del usuario de AWS CloudHSM .

Para ejecutar este procedimiento, debe desconectar temporalmente el almacén de AWS CloudHSM claves para poder iniciar sesión como CU. kmsuser

1. Desconecte el almacén de AWS CloudHSM claves, si aún no está desconectado, inicie sesión como se explica en [Cómo desconectar e iniciar sesión](#). kmsuser

 Note

Mientras un almacén de claves personalizado esté desconectado, todos los intentos de crear claves KMS en el almacén de claves personalizado o de usar claves KMS existentes en operaciones criptográficas fallarán. Esta acción puede impedir que los usuarios almacenen y accedan a datos confidenciales.

2. Utilice el comando [key list](#) de la CLI de CloudHSM para buscar todas las claves del usuario actual presentes AWS CloudHSM en el clúster.

De forma predeterminada, solo se muestran 10 claves del usuario que ha iniciado sesión y solo se muestran las key-reference y label. Para obtener más opciones, consulte la [lista de claves](#) en la Guía del usuario de 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. Cierre sesión y vuelva a conectar el almacén de AWS CloudHSM claves como se describe en [Cómo cerrar sesión y volver a conectar](#)

Busque la clave KMS de una AWS CloudHSM clave

Si conoce la referencia clave o el ID de una clave que kmsuser posee en el clúster, puede usar ese valor para identificar la clave de KMS asociada en su almacén de AWS CloudHSM claves.

Cuando AWS KMS crea el material clave para una clave de KMS en el AWS CloudHSM clúster, escribe el nombre de recurso de Amazon (ARN) de la clave de KMS en la etiqueta de la clave. A menos que haya cambiado el valor de la etiqueta, puede utilizar el comando [key list](#) en la CLI de CloudHSM para identificar la clave KMS asociada a la clave de AWS CloudHSM .

Notas

Los siguientes procedimientos utilizan la herramienta de línea de comandos AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI de CloudHSM reemplaza key-handle por key-reference.

El 1 de enero de 2025, AWS CloudHSM dejará de ser compatible con las herramientas de línea de comandos Client SDK 3, la utilidad de administración CloudHSM (CMU) y la utilidad de administración de claves (KMU). Para obtener más información sobre las diferencias entre las herramientas de línea de comandos Client SDK 3 y Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) en la Guía del usuario de AWS CloudHSM .

Para ejecutar estos procedimientos, debe desconectar temporalmente el almacén de AWS CloudHSM claves para poder iniciar sesión como CU. `kmsuser`

Note

Mientras un almacén de claves personalizado esté desconectado, todos los intentos de crear claves KMS en el almacén de claves personalizado o de usar claves KMS existentes en operaciones criptográficas fallarán. Esta acción puede impedir que los usuarios almacenen y accedan a datos confidenciales.

Temas

- [Identificación de la clave KMS asociada a una clave de referencia](#)
- [Identificación de la clave de KMS asociada a un ID de clave de respaldo](#)

Identificación de la clave KMS asociada a una clave de referencia

Los siguientes procedimientos muestran cómo utilizar el comando [key list](#) en la CLI de CloudHSM con el filtro de atributos `key-reference` para encontrar la clave en el clúster que actúa como material de claves para una clave KMS en particular en su almacén de claves de AWS CloudHSM .

1. Desconecte el almacén de AWS CloudHSM claves, si aún no está desconectado, inicie sesión como se explica en [Cómo desconectar e iniciar sesión](#). `kmsuser`
2. Utilice el comando [key list](#) en la CLI de CloudHSM para filtrar por el atributo `key-reference`. Especifique el argumento `verbose` para incluir todos los atributos y la información de claves de la clave coincidente. Si no especifica el argumento `verbose`, la operación `key list` solo devuelve la referencia de clave y el atributo de etiqueta de la clave coincidente.

Antes de ejecutar este comando, reemplace el `key-reference` de ejemplo por uno válido de su cuenta.

```
aws-cloudhsm > key list --filter attr.key-reference="0x00000000000120034" --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. Cierre la sesión y vuelva a conectar el almacén de AWS CloudHSM claves como se describe en [Cómo cerrar sesión y volver a conectar](#).

Identificación de la clave de KMS asociada a un ID de clave de respaldo

Todas las entradas de CloudTrail registro de operaciones criptográficas con una clave KMS en un almacén de AWS CloudHSM claves incluyen un `additionalEventData` campo con la `customKeyId` letra y. `backingKeyId` El valor devuelto en el campo `backingKeyId` se correlaciona con el atributo `id` de clave de CloudHSM. Puede filtrar la operación [key list](#) por el atributo `id` para identificar la clave KMS asociada a un `backingKeyId` específico.

1. Desconecte el almacén de AWS CloudHSM claves, si aún no está desconectado, e inicie sesión como `kmsuser` se explica en [Cómo desconectar e iniciar sesión](#).
2. Utilice el comando [key list](#) en la CLI de CloudHSM con el filtro de atributo para encontrar la clave en el clúster que actúa como material de claves para una clave KMS en particular en su almacén de claves de AWS CloudHSM .

En el siguiente ejemplo se muestra cómo filtrar por atributo `id`. AWS CloudHSM reconoce el valor de `id` como un valor hexadecimal. Para filtrar la operación de la lista de claves por `id` atributo, primero debe convertir el `backingKeyId` valor que identificó en la entrada de CloudTrail registro a un formato que AWS CloudHSM reconozca.

- a. Utilice el siguiente comando de Linux para convertir `backingKeyId` en una representación hexadecimal.

```
echo backingKeyId | tr -d '\n' | xxd -p
```

El siguiente ejemplo muestra cómo convertir la matriz de bytes de `backingKeyId` en una representación hexadecimal.

```
echo 5890723622dc15f699aa9ab2387a9f744b2b884c18b2186ee8ada4f556a2eb9d | tr -d '\n' | xxd -p
353839303732333632326463313566363939616139616232333837613966373434623262383834633138623
```

- b. Anexe la representación hexadecimal del `backingKeyId` con `0x`.

```
0x353839303732333632326463313566363939616139616232333837613966373434623262383834633138623
```

- c. Utilice el valor de `backingKeyId` convertido para filtrar por atributo `id`. Especifique el argumento `verbose` para incluir todos los atributos y la información de claves de la clave coincidente. Si no especifica el argumento `verbose`, la operación `key list` solo devuelve la referencia de clave y el atributo de etiqueta de la clave coincidente.

```
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":
"0x353839303732333632326463313566363939616139616232333837613966373434623262383463313
          "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. Cierre la sesión y vuelva a conectar el almacén de AWS CloudHSM claves como se describe en [Cómo cerrar sesión y volver a conectar](#).

Busque la AWS CloudHSM clave de una clave KMS

Puede usar el ID de clave de KMS de una clave de KMS de un AWS CloudHSM almacén de claves para identificar la clave del AWS CloudHSM clúster que sirve como material clave.

Cuando AWS KMS crea el material clave para una clave de KMS en el AWS CloudHSM clúster, escribe el nombre de recurso de Amazon (ARN) de la clave de KMS en la etiqueta de la clave. A menos que haya cambiado el valor de la etiqueta, puede usar el comando [key list](#) en la CLI de CloudHSM para encontrar el identificador y el recurso de clave del material de claves para la clave KMS.

Todas las entradas de CloudTrail registro para operaciones criptográficas con una clave KMS en un almacén de AWS CloudHSM claves incluyen un `additionalEventData` campo con la `customKeyStoreId` letra y. `backingKeyId` El valor devuelto en el `backingKeyId` campo es el atributo `id` AWS CloudHSM clave. Puede filtrar la operación AWS CloudHSM CLI de la lista de claves por el ARN de la clave de KMS para identificar el `id` atributo de clave de CloudHSM asociado a una clave de KMS específica.

Para ejecutar este procedimiento, debe desconectar temporalmente el almacén de AWS CloudHSM claves para poder iniciar sesión como `CU.kmsuser`

Notas

Los siguientes procedimientos utilizan la herramienta de línea de comandos AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI de CloudHSM reemplaza `key-handle` por `key-reference`.

El 1 de enero de 2025, AWS CloudHSM dejará de ser compatible con las herramientas de línea de comandos Client SDK 3, la utilidad de administración CloudHSM (CMU) y la utilidad de administración de claves (KMU). Para obtener más información sobre las diferencias entre las herramientas de línea de comandos Client SDK 3 y Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) en la Guía del usuario de AWS CloudHSM.

1. Desconecte el almacén de AWS CloudHSM claves, si aún no está desconectado, e inicie sesión como `kmsuser` se explica en [Cómo desconectar e iniciar sesión](#)

Note

Mientras un almacén de claves personalizado esté desconectado, todos los intentos de crear claves KMS en el almacén de claves personalizado o de usar claves KMS existentes en operaciones criptográficas fallarán. Esta acción puede impedir que los usuarios almacenen y accedan a datos confidenciales.

2. Utilice el comando [key list](#) de la CLI de CloudHSM y `label` filtre para encontrar la clave de KMS de una clave AWS CloudHSM concreta del clúster. Especifique el argumento `verbose` para incluir todos los atributos y la información de claves de la clave coincidente. Si no especifica el argumento `verbose`, la operación `key list` solo devuelve la referencia de clave y los atributos de etiqueta de la clave coincidente.

En el siguiente ejemplo se muestra cómo filtrar por el atributo `label` que almacena el ARN de clave KMS. Antes de ejecutar este comando, reemplace el ARN de clave KMS de ejemplo por uno válido de su cuenta.

```
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": "0x000000000000120034",
    "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. Cierre sesión y vuelva a conectar el almacén de AWS CloudHSM claves como se describe en. [Cómo cerrar sesión y volver a conectar](#)

Habilitar y deshabilitar claves

Puede habilitar y deshabilitar claves administradas por el cliente. Al crear una clave KMS, está habilitada de forma predeterminada. Si desactiva una clave KMS, no se puede utilizar en ninguna [operación criptográfica](#) hasta que lo vuelva a habilitar.

Debido a que es temporal y se deshace fácilmente, desactivar una clave KMS es una alternativa segura a eliminar una clave KMS, una acción destructiva e irreversible. Si está pensando en eliminar una clave KMS, deshabilítela primero y configure una [CloudWatch alarma](#) o un mecanismo similar para asegurarse de que nunca necesitará usar la clave para descifrar datos cifrados.

Al deshabilitar una clave de KMS, queda inutilizable de inmediato (sujeto a posible coherencia). Sin embargo, los recursos cifrados con [claves de datos](#) protegidas por la clave de KMS no se ven afectados hasta que se vuelva a utilizar la clave de KMS, por ejemplo, para descifrar la clave de datos. Este problema afecta a Servicios de AWS muchos de los cuales utilizan claves de datos para proteger sus recursos. Para obtener más información, consulte [Cómo afectan las claves de KMS obsoletas a las claves de datos](#).

No puede activar ni desactivar [Claves administradas por AWS](#) o [Claves propiedad de AWS](#). Claves administradas por AWS están habilitados permanentemente para su uso por parte de [los servicios que utilizan AWS KMS](#). Claves propiedad de AWS están gestionados únicamente por el servicio propietario de los mismos.

Note

AWS KMS no rota el material clave de las claves administradas por el cliente mientras están deshabilitadas. Para obtener más información, consulte [Cómo funciona la rotación de claves](#).

Uso de la AWS KMS consola

Puede utilizar la AWS KMS consola para activar y desactivar [las claves gestionadas por el cliente](#).

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.

3. En el panel de navegación, elija Claves administradas por el cliente.
4. Elija la casilla de verificación de las claves de KMS que quiere habilitar o deshabilitar.
5. Para habilitar una clave KMS, elija Key actions (Acciones de claves), Enable (Habilitar). Para deshabilitar una clave KMS, elija Key actions (Acciones de claves), Disable (Deshabilitar).

Uso de la API AWS KMS

La [EnableKey](#) operación habilita a un discapacitado AWS KMS key. En estos ejemplos, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido. El parámetro `key-id` es obligatorio.

Esta operación no devuelve ninguna salida. Para ver el estado de la clave, utilice la [DescribeKey](#) operación.

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

La [DisableKey](#) operación deshabilita una clave KMS habilitada. El parámetro `key-id` es obligatorio.

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

Esta operación no devuelve ninguna salida. Para ver el estado de la clave, utilice la [DescribeKey](#) operación y consulte el `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"  
    ]  
}
```

Rotate (Rotar) AWS KMS keys

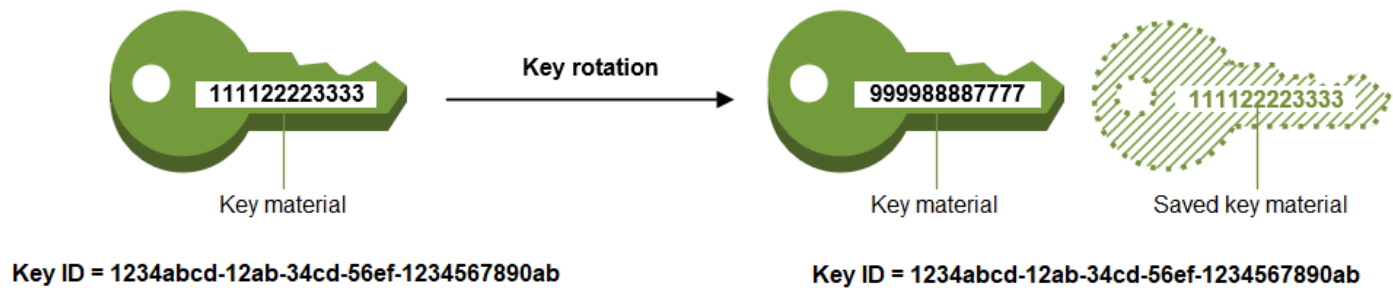
Para crear nuevo material criptográfico para sus [claves administradas por el cliente](#), puede crear nuevas claves KMS y, a continuación, cambiar sus aplicaciones o alias para que utilicen las claves. O bien, puede rotar el material de claves asociado a una clave KMS existente habilitando la rotación automática de claves o realizando la rotación bajo demanda.

De forma predeterminada, cuando se habilita la rotación automática de claves para una clave de KMS, se AWS KMS genera nuevo material criptográfico para la clave de KMS cada año. También puede especificar una opción personalizada [rotation-period](#) para definir el número de días después de activar la rotación automática de claves que AWS KMS rotará el material clave y el número de días que transcurrirán entre cada rotación automática a partir de entonces. Si necesita iniciar inmediatamente la rotación del material clave, puede realizar la rotación bajo demanda, independientemente de si la rotación automática de claves está habilitada o no. On-demand las rotaciones no cambian los programas de rotación automática existentes.

Puede [realizar un seguimiento de la rotación](#) del material clave de sus claves de KMS en Amazon CloudWatch y en la AWS Key Management Service consola. AWS CloudTrail También puede utilizar la [GetKeyRotationStatus](#) operación para comprobar si la rotación automática está habilitada para una clave de KMS e identificar cualquier rotación que se esté realizando bajo demanda. Puede utilizar la [ListKeyRotations](#) operación para ver los detalles de las rotaciones completadas.

La rotación de claves cambia únicamente el material de claves actual, que es el secreto criptográfico que se usa en las operaciones de cifrado. Cuando utiliza la clave KMS girada para descifrar el texto cifrado, AWS KMS utiliza el material clave que se utilizó para cifrarlo. No puede seleccionar un material de claves concreto para las operaciones de descifrado; AWS KMS elige automáticamente el material de claves correcto. Como se descifra de AWS KMS forma transparente con el material de clave adecuado, puede utilizar de forma segura una clave KMS girada en las aplicaciones y sin cambios de código. Servicios de AWS

La clave KMS es el mismo recurso lógico, independientemente de si el material de claves cambie o de cuántas veces lo haga. Las propiedades de la clave KMS no cambian, tal y como se muestra en la siguiente imagen.



Podría optar por crear una nueva clave KMS y utilizarla en lugar de la clave KMS original. Esto tiene el mismo efecto que rotar el material de claves de una clave KMS existente, así que a menudo se considera una [rotación manual de la clave](#). La rotación manual es una buena opción si desea rotar claves KMS que no cumplen los requisitos para la rotación automática o bajo demanda, incluidas las claves KMS [asimétricas](#), [las claves KMS HMAC](#) y [las claves KMS en almacenes de claves personalizados](#).

Note

La rotación de claves automática no afecta a los datos que la clave de KMS protege. No rota las claves de datos que ha generado la clave de KMS ni vuelve a cifrar los datos protegidos por la clave de KMS. La rotación de claves no mitigará el efecto de una clave de datos comprometida.

Rotación de claves y precios

AWS KMS cobra una cuota mensual por la primera y la segunda rotación del material clave conservado para tu clave KMS. Este aumento de precio está limitado a la segunda rotación y las rotaciones posteriores no se facturarán. Consulte los [Precios de AWS Key Management Service](#) para obtener más información.

Note

Puedes usar [AWS Cost Explorer Service](#) para ver un desglose de los cargos por almacenamiento de claves. Por ejemplo, puede filtrar la vista para ver los cargos totales de las claves facturadas como claves de KMS actuales y rotadas especificando \$REGION-KMS-Keys en Tipo de uso y agrupando los datos por Operación de la API.

Es posible que siga viendo instancias de la operación de la API Unknown heredada para fechas antiguas.

Rotación de claves y cuotas

Cada clave KMS cuenta como una clave para el cálculo de las cuotas de recursos clave, independientemente del número de versiones de material de claves rotadas.

Para obtener información detallada sobre el material de claves y la rotación, consulte los [Detalles criptográficos de AWS Key Management Service](#).

Temas

- [¿Por qué rotar las claves de KMS?](#)
- [Cómo funciona la rotación de claves](#)
- [Habilitación de la rotación automática de claves](#)
- [Deshabilitación de la rotación automática de claves](#)
- [Realización de la rotación de claves bajo demanda](#)
- [Enumeración de las rotaciones y los materiales clave](#)
- [Rotación manual de claves](#)
- [Cambio de la clave principal en un conjunto de claves de varias regiones](#)

¿Por qué rotar las claves de KMS?

Las mejores prácticas criptográficas desaconsejan la reutilización extensiva de las claves que cifran los datos directamente, como las [claves de datos](#) que se generan. AWS KMS Cuando las claves de datos de 256 bits cifran millones de mensajes, estas pueden fatigarse y empezar a producir texto cifrado con patrones sutiles que los actores inteligentes pueden aprovechar para descubrir los bits de la clave. Lo mejor es utilizar las claves de datos una vez o solo unas cuantas veces para mitigar el agotamiento de las claves.

Sin embargo, las claves de KMS se utilizan con mayor frecuencia como claves de encapsulamiento, también conocidas como claves de cifrado de claves. En lugar de cifrar los datos, las claves de encapsulamiento cifran las claves de datos que cifran los datos. Por lo tanto, se utilizan con mucha menos frecuencia que las claves de datos y casi nunca se reutilizan lo suficiente como para correr el riesgo de que se fatiguen las claves.

A pesar de que el riesgo de fatiga es muy bajo, es posible que tenga que cambiar sus claves de KMS debido a normas comerciales o contractuales o a normativas gubernamentales. Cuando se vea en la obligación de rotar las claves de KMS, le recomendamos que utilice la rotación automática de claves cuando se permita, la rotación bajo demanda si la rotación automática no se admite y la rotación manual de claves cuando no se admita la rotación automática de claves ni la rotación bajo demanda.

Podría considerar la realización de rotaciones bajo demanda para demostrar las capacidades de rotación de material de claves o para validar los scripts de automatización. Recomendamos utilizar rotaciones bajo demanda para las rotaciones no planificadas y utilizar la rotación automática de claves con un [periodo de rotación](#) personalizado siempre que sea posible.

Cómo funciona la rotación de claves

AWS KMS La rotación de claves está diseñada para ser transparente y fácil de usar. AWS KMS admite la rotación de claves automática y bajo demanda opcional solo para [las claves administradas por el cliente](#).

Rotación automática de claves

AWS KMS gira la clave KMS automáticamente en la siguiente fecha de rotación definida por su período de rotación. No necesita recordar ni programar la actualización.

La rotación automática de claves solo se admite en las claves KMS de cifrado simétrico con el material de clave AWS KMS generado (AWS_KMSorigen).

La rotación automática es opcional para las claves KMS administradas por el cliente. AWS KMS siempre rota el material clave de las claves de KMS AWS administradas cada año. La rotación de las claves KMS AWS propias la gestiona el Servicio de AWS propietario de la clave.

On-demand rotación

Inicie de inmediato la rotación del material de claves asociado a su clave KMS, independientemente de si la rotación automática de claves está habilitada o no.

On-demand la rotación de claves se admite en las claves KMS de cifrado simétrico con material de clave que AWS KMS genera (AWS_KMSorigen) y en las claves KMS de cifrado simétrico con material de clave importado (EXTERNALorigen).

Rotación manual

La rotación automática de claves no es compatible en los siguientes tipos de claves de KMS, pero puede [rotar estas claves de KMS de forma manual](#).

- [Claves de KMS asimétricas](#)
- [Claves KMS HMAC](#)
- Claves KMS en [almacenes de claves personalizados](#)

Administración de material de claves

AWS KMS conserva todo el material clave de una clave KMS con AWS_KMS origen, incluso si la rotación de claves está desactivada. AWS KMS elimina el material clave solo al eliminar la clave KMS.

Usted administra los materiales clave de las claves de cifrado simétrico con origen EXTERNAL. Puede eliminar cualquier material clave mediante la [DeleteImportedKeyMaterial](#) operación o establecer una fecha de caducidad al importar el material. La clave de KMS deja de poder utilizarse en cuanto alguno de sus materiales caduca o se elimina.

Uso de material de claves

Cuando utiliza una clave KMS girada para cifrar datos, AWS KMS utiliza el material clave actual. Cuando se utiliza la clave KMS rotada para descifrar el texto cifrado, AWS KMS utiliza la misma versión del material de claves utilizado para cifrarlo. No puede seleccionar una versión concreta del material clave para las operaciones de descifrado, sino que elige AWS KMS automáticamente la versión correcta.

Periodo de rotación

El período de rotación define el número de días después de activar la rotación automática de claves que AWS KMS rotarán el material clave y el número de días que transcurrirán entre cada rotación automática de claves a partir de entonces. Si no especifica un valor para `RotationPeriodInDays` al habilitar la rotación automática de claves, el valor predeterminado es de 365 días.

Puede utilizar la clave de `RotationPeriodInDays` condición [kms:](#) para restringir aún más los valores que los directores pueden especificar en el `RotationPeriodInDays` parámetro.

Fecha de rotación

La fecha de rotación refleja la fecha en que se actualizó el material clave actual de una clave de KMS, ya sea como resultado de una rotación automática (programada) o de una rotación de clave bajo demanda.

Fecha de rotación

AWS KMS gira automáticamente la clave KMS en la fecha de rotación definida por su período de rotación. El periodo de rotación predeterminado es de 365 días.

Claves administradas por el cliente

Como la rotación automática de claves es opcional en las [claves administradas por el cliente](#) y se puede habilitar y deshabilitar en cualquier momento, la fecha de rotación depende de la fecha en que se habilitó la rotación por última vez. La fecha puede cambiar si modifica el periodo de rotación de una clave en la que anteriormente había habilitado la rotación automática de claves. La fecha de rotación puede cambiar muchas veces durante la vida de la clave.

Por ejemplo, si crea una clave administrada por el cliente el 1 de enero de 2022 y habilita la rotación automática de claves con el periodo de rotación predeterminado de 365 días el 15 de marzo de 2022, AWS KMS rota el material de claves el 15 de marzo de 2023, el 15 de marzo de 2024 y, a partir de entonces, cada 365 días.

En los siguientes ejemplos se supone que la rotación automática de claves estaba habilitada con el periodo de rotación predeterminado de 365 días. Estos ejemplos muestran casos especiales que podrían afectar al periodo de rotación de una clave.

- Deshabilitar la rotación de claves: si [deshabilita la rotación automática de claves](#) en cualquier momento, la clave de KMS seguirá utilizando la versión del material de claves que utilizaba cuando la rotación estaba deshabilitada. Si vuelve a activar la rotación automática de claves, AWS KMS gira el material clave en función de la nueva fecha de activación de la rotación.
- Teclas KMS deshabilitadas: mientras una clave KMS esté deshabilitada, AWS KMS no la rota. Sin embargo, el estado de rotación de clave no cambia y no puede cambiarlo mientras la clave KMS esté deshabilitada. Cuando se vuelve a activar la clave KMS, si el material clave ha pasado su última fecha de rotación programada, lo AWS KMS rota inmediatamente. Si el material clave no ha perdido la última fecha de rotación programada, AWS KMS reanuda el programa de rotación clave original.
- Claves de KMS pendientes de eliminación: mientras una clave de KMS esté pendiente de eliminación, AWS KMS no la rota. El estado de rotación de clave se establece en `false` y no puede cambiarla mientras su eliminación está pendiente. Si se cancela la eliminación, se restaura el estado de rotación de clave anterior. Si el material clave ha pasado su última fecha de rotación programada, lo AWS KMS gira inmediatamente. Si el material clave no

ha perdido su última fecha de rotación programada, AWS KMS reanuda la programación de rotación clave original.

Claves administradas por AWS

AWS KMS rota automáticamente Claves administradas por AWS cada año (aproximadamente 365 días). No puede habilitar ni desactivar la rotación de claves de [Claves administradas por AWS](#).

El material clave de un objeto Clave administrada de AWS se rota primero un año después de su fecha de creación y, a partir de entonces, todos los años (aproximadamente 365 días a partir de la última rotación).

Note

En mayo de 2022, AWS KMS cambiamos el programa de rotación Claves administradas por AWS de cada tres años (aproximadamente 1095 días) a uno anual (aproximadamente 365 días).

Claves propiedad de AWS

No puede habilitar ni desactivar la rotación de claves de Claves propiedad de AWS. La estrategia de [rotación clave](#) de una Clave propiedad de AWS viene determinada por el AWS servicio que crea y administra la clave. Para obtener más detalles, consulte el tema Cifrado en reposo en la Guía del usuario o en la Guía para desarrolladores del servicio.

Rotación de claves de varias regiones

El comportamiento de rotación varía en función de si el material clave se genera por AWS KMS (AWS_KMSorigen) o se importa (EXTERNALorigen).

Multi-Region claves con **AWS_KMS** origen

Puede habilitar y deshabilitar la rotación automática de claves y realizar la rotación bajo demanda del material de claves en [claves de varias regiones](#) con origen AWS_KMS. La rotación de claves es una [propiedad compartida](#) de claves de varias regiones.

Solo habilita y deshabilita la rotación automática de claves en la clave principal. Inicia la rotación bajo demanda solo en la clave principal.

- Al AWS KMS sincronizar las claves multirregionales, copia el valor de la propiedad de rotación de claves de la clave principal a todas las claves de réplica relacionadas.

- Al AWS KMS girar el material clave, crea un nuevo material clave para la clave principal y, a continuación, lo copia más allá de los límites de la región en todas las réplicas de claves relacionadas. El material clave nunca sale AWS KMS sin cifrar. Este paso se controla cuidadosamente para garantizar que el material clave esté completamente sincronizado antes de utilizar cualquier clave en una operación criptográfica.
- AWS KMS no cifra ningún dato con el nuevo material clave hasta que dicho material clave esté disponible en la clave principal y en cada una de sus réplicas.
- Cuando se replica una clave principal que se ha rotado, la nueva clave de réplica tiene el material de clave actual y todas las versiones anteriores del material clave para sus claves de varias regiones relacionadas.

Este patrón garantiza que las claves de varias regiones relacionadas sean totalmente interoperables. Cualquier clave de varias regiones puede descifrar cualquier texto cifrado mediante una clave de varias regiones relacionada, incluso si el texto cifrado se cifró antes de que se creara la clave.

Multi-Region claves con origen **EXTERNAL**

Puede realizar la rotación bajo demanda del material clave en claves [multirregionales con cifrado simétrico con EXTERNAL origen](#). La rotación de claves es una [propiedad compartida](#) de claves de varias regiones.

La rotación bajo demanda solo se inicia en la clave principal después de importar el nuevo material de claves a la clave principal y a cada clave de réplica.

- Al importar un nuevo material clave a la clave principal, AWS KMS copia el identificador del material clave y la descripción del material clave de la clave principal a todas las claves de réplica relacionadas. No copia el material clave.
- Debe importar el mismo material clave en cada clave de réplica de forma individual. Una vez importado el material clave a todas las claves multirregionales relacionadas, puede iniciar la rotación bajo demanda de la clave principal. Esto garantiza que AWS KMS no se cifra ningún dato con el nuevo material clave hasta que dicho material clave esté disponible en la clave principal y en cada una de sus réplicas.
- Cada material clave de la clave principal o de cualquier clave de réplica puede caducar o eliminarse independientemente del resto del material clave de la misma clave o de cualquier otra clave multirregional relacionada.

AWS servicios

Puede activar la rotación automática de claves en [las claves administradas por el cliente](#) que utiliza para el cifrado del lado del servidor en AWS los servicios. La rotación anual es transparente y compatible con AWS los servicios.

Supervisión de la rotación de claves

[Cuando AWS KMS rota el material clave de una Clave administrada de AWS clave gestionada por el cliente, escribe un KMS CMK Rotation evento en Amazon EventBridge y otro RotateKey en tu AWS CloudTrail registro.](#) Puede usar estos registros para comprobar que la clave KMS se ha rotado.

Puede usar la AWS Key Management Service consola para ver el número de rotaciones pendientes bajo demanda y una lista de todas las rotaciones de materiales clave completadas para una clave de KMS.

Puede utilizar la [ListKeyRotations](#) operación para ver los detalles de las rotaciones completadas.

Consistencia final

La rotación de claves está sujeta a los mismos efectos de consistencia final que otras operaciones de administración de AWS KMS . Es posible que haya un ligero retraso antes de que el nuevo material de claves esté disponible en AWS KMS. Sin embargo, la rotación de material clave no causa ninguna interrupción o retraso en las operaciones criptográficas. El material clave actual se utiliza en operaciones criptográficas hasta que el nuevo material de claves esté disponible en toda la AWS KMS. Cuando el material clave de una clave multirregional se rota automáticamente, AWS KMS utiliza el material clave actual hasta que el nuevo material clave esté disponible en todas las regiones con una clave multirregional relacionada.

Habilitación de la rotación automática de claves

De forma predeterminada, cuando se habilita la rotación automática de claves para una clave de KMS, se AWS KMS genera nuevo material criptográfico para la clave de KMS cada año. También puede especificar una opción personalizada [rotation-period](#) para definir el número de días después de activar la rotación automática de claves que AWS KMS rotará el material clave y el número de días que transcurrirán entre cada rotación automática a partir de entonces.

La rotación automática de claves tiene las siguientes ventajas:

- Las propiedades de la clave KMS, incluido su [ID de clave](#), [ARN de clave](#), región, políticas y permisos, no cambian cuando se rota la clave.
- No necesita cambiar las aplicaciones ni los alias que hacen referencia al ID o ARN de la clave KMS.
- El material de claves de rotación no afecta al uso de la clave KMS en ningún Servicio de AWS.
- Tras activar la rotación de claves, AWS KMS gira la clave KMS automáticamente en la siguiente fecha de rotación definida por el período de rotación. No necesita recordar ni programar la actualización.

Puede activar la rotación automática de las teclas en la AWS KMS consola o mediante esta [EnableKeyRotation](#) operación. Para habilitar la rotación automática de claves, necesita permisos de `kms:EnableKeyRotation`. Para obtener más información sobre AWS KMS los permisos, consulte la [Referencia de permisos](#).

Uso de AWS KMS consola

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente. (No puede habilitar o desactivar la rotación de Claves administradas por AWS. Estas rotan cada año de forma automática).
4. Elija el alias o el ID de clave de una clave KMS.
5. Seleccione la pestaña Key Rotation (Rotación de claves).

La pestaña Rotación de claves solo aparece en la página de detalles de las claves de KMS de cifrado simétricas con el material de claves que AWS KMS genera (el Origen es AWS_KMS), incluidas las claves de KMS de cifrado simétricas [de varias regiones](#).

No puede rotar de forma automática las claves KMS asimétricas, las claves KMS HMAC, las claves KMS con [material de claves importado](#) o las claves KMS en los [almacenes de claves personalizados](#). Sin embargo, puede [rotarlas manualmente](#).

6. En la sección Rotación automática de claves, seleccione Editar.
7. En Key rotation (rotación de claves), seleccione Enable (Habilitar).

 Note

Si una clave de KMS está deshabilitada o pendiente de ser eliminada, AWS KMS no se rota el material clave y no se puede actualizar el estado de rotación automática de la clave ni el período de rotación. Para actualizar la configuración de la rotación automática de claves, habilite la clave KMS o cancele la eliminación. Para más detalles, consulte [Cómo funciona la rotación de claves](#) y [Estados clave de AWS KMS claves](#).

8. (Opcional) Escriba un periodo de rotación de entre 90 y 2560 días. El valor predeterminado es 365 días. Si no especifica un período de rotación personalizado, AWS KMS rotará el material clave todos los años.

Puede usar la clave de `RotationPeriodInDays` condición [kms:](#) para limitar los valores que los directores pueden especificar para el período de rotación.

9. Seleccione Save.

Uso de AWS KMS API

Puede utilizar la [API de AWS Key Management Service \(AWS KMS\)](#) para habilitar la rotación automática de claves y ver el estado de rotación actual de cualquier clave administrada por el cliente. En estos ejemplos, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

La [EnableKeyRotation](#) operación permite la rotación automática de claves para la clave KMS especificada. Para identificar la clave KMS en esta operación, utilice el [ID de la clave](#) o el [ARN de la clave](#). De forma predeterminada, la rotación de claves está desactivada para las claves KMS administradas por el cliente.

Puede usar la clave de `RotationPeriodInDays` condición [kms:](#) para limitar los valores que los directores pueden especificar para el `RotationPeriodInDays` parámetro de una `EnableKeyRotation` solicitud.

El siguiente ejemplo permite la rotación de claves con un período de rotación de 180 días en la clave KMS de cifrado simétrico especificada y utiliza la [GetKeyRotationStatus](#) operación para ver el 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"  
}
```

Deshabilitación de la rotación automática de claves

Tras habilitar la rotación automática de claves en una clave administrada por el cliente, puede optar por deshabilitarla en cualquier momento.

Si deshabilita la rotación automática de claves, la clave KMS seguirá utilizando la versión del material de claves que utilizaba cuando la rotación estaba deshabilitada. Si vuelve a activar la rotación automática de claves, AWS KMS rota el material de clave en función de la nueva fecha de activación de la rotación.

La deshabilitación de la rotación automática no afecta a su capacidad para [realizar rotaciones bajo demanda](#) ni cancela ninguna rotación bajo demanda en curso.

Puede desactivar la rotación automática de las teclas en la AWS KMS consola o mediante esta operación. [DisableKeyRotation](#) Para deshabilitar la rotación automática de claves, necesita permisos de `kms:DisableKeyRotation`. Para obtener más información sobre AWS KMS los permisos, consulte la [Referencia de permisos](#).

Uso de AWS KMS consola

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente. (No puede habilitar o desactivar la rotación de Claves administradas por AWS. Estas rotan cada año de forma automática).
4. Elija el alias o el ID de clave de una clave KMS.

5. Seleccione la pestaña Key Rotation (Rotación de claves).

La pestaña Rotación de claves solo aparece en la página de detalles de las claves de KMS de cifrado simétricas con el material de claves que AWS KMS genera (el Origen es AWS_KMS), incluidas las claves de KMS de cifrado simétricas [de varias regiones](#).

No puede rotar de forma automática las claves KMS asimétricas, las claves KMS HMAC, las claves KMS con [material de claves importado](#) o las claves KMS en los [almacenes de claves personalizados](#). Sin embargo, puede [rotarlas manualmente](#).

6. En la sección Rotación automática de claves, seleccione Editar.

7. En Key rotation (rotación de claves), seleccione Disable (Deshabilitar).

Note

Si una clave de KMS está deshabilitada o pendiente de ser eliminada, AWS KMS no se rota el material clave y no se puede actualizar el estado de rotación automática de la clave ni el período de rotación. Para actualizar la configuración de la rotación automática de claves, habilite la clave KMS o cancele la eliminación. Para más detalles, consulte [Cómo funciona la rotación de claves](#) y [Estados clave de AWS KMS claves](#).

8. Seleccione Save.

Uso de AWS KMS API

Puede utilizar la [API de AWS Key Management Service \(AWS KMS\)](#) para deshabilitar la rotación automática de claves y ver el estado de rotación actual de cualquier clave administrada por el cliente. En estos ejemplos se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

La [DisableKeyRotation](#) operación desactiva la rotación automática de la clave. Para identificar la clave KMS en esta operación, utilice el [ID de la clave](#) o el [ARN de la clave](#). De forma predeterminada, la rotación de claves está desactivada para las claves KMS administradas por el cliente.

El ejemplo siguiente desactiva la rotación automática de claves en la clave KMS de cifrado simétrico especificada y utiliza la [GetKeyRotationStatus](#) operación para ver el 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
}
```

Realización de la rotación de claves bajo demanda

Puede realizar la rotación bajo demanda del material de claves en las claves KMS administradas por el cliente, independientemente de si la rotación automática de claves está habilitada o no. La desactivación de la rotación automática ([DisableKeyRotation](#)) no afecta a su capacidad para realizar rotaciones bajo demanda ni cancela ninguna rotación bajo demanda en curso. On-demand las rotaciones no cambian los programas de rotación automática existentes. Por ejemplo, pensemos en una clave KMS que tenga habilitada la rotación automática de claves con un período de rotación de 730 días. Si la clave está programada para rotar automáticamente el 14 de abril de 2024 y realiza una rotación bajo demanda el 10 de abril de 2024, la clave rotará automáticamente, según lo programado, el 14 de abril de 2024 y, a partir de entonces, cada 730 días.

Puede realizar la rotación de claves bajo demanda un máximo de 25 veces por clave KMS. Puede utilizar la AWS KMS consola para ver el número de rotaciones bajo demanda restantes disponibles para una clave KMS.

On-demand La rotación de claves solo se admite en las claves [KMS de cifrado simétrico](#). No puede realizar la rotación bajo demanda de [claves KMS asimétricas](#), [claves KMS HMAC](#) o [claves KMS](#) en un almacén de claves [personalizado](#). Para realizar la rotación bajo demanda de un conjunto de [claves de varias regiones](#) relacionadas, invoque la rotación bajo demanda en la clave principal.

Los usuarios autorizados con `kms:GetKeyRotationStatus` permisos `kms:RotateKeyOnDemand` y permisos pueden usar la AWS KMS consola y la AWS KMS API para iniciar la rotación de claves bajo demanda y ver el estado de la rotación de claves. Se utiliza [ListKeyRotations](#) para ver las rotaciones completadas de una clave KMS.

Temas

- [Inicio de la rotación de claves bajo demanda \(consola\)](#)
- [Iniciar la rotación de claves bajo demanda \(AWS KMS API\)](#)

Inicio de la rotación de claves bajo demanda (consola)

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente. (No puede realizar la rotación bajo demanda de Claves administradas por AWS. Estas rotan cada año de forma automática).
4. Elija el alias o el ID de clave de una clave KMS.
5. Seleccione la pestaña Material y rotaciones de las claves.

La pestaña Material y rotaciones de las claves solo aparece en la página de detalles de las claves de KMS de cifrado simétrico que admiten la rotación automática o bajo demanda. Esto incluye las claves KMS con el material clave AWS KMS generado (origen AWS_KMS) y claves KMS con material clave importado (origen EXTERNO)

[No puede realizar la rotación bajo demanda de claves KMS asimétricas, claves KMS de HMAC o claves KMS en almacenes de claves personalizados.](#) Sin embargo, puede [rotarlas manualmente](#).

6. Elija Rotar ahora. Para las claves de cifrado simétricas con material de clave importado, la opción Rotar ahora solo está disponible si ha [importado anteriormente un material de clave nuevo](#) y se encuentra en estado de rotación pendiente.

Note

En el caso de las claves multirregionales, solo se puede girar la clave de región principal.

7. Lea y tenga en cuenta la advertencia y la información sobre el número de rotaciones bajo demanda restantes de la clave. También verá información como el identificador, la descripción y la fecha de caducidad del material de claves que se actualizará tras la rotación. Si decide que no desea continuar con la rotación bajo demanda, seleccione Cancelar.
8. Seleccione Rotar clave para confirmar la rotación bajo demanda.

 Note

On-demand la rotación está sujeta a los mismos posibles efectos de coherencia que otras operaciones AWS KMS de gestión. Es posible que haya un ligero retraso antes de que el nuevo material de claves esté disponible en AWS KMS. El banner en la parte superior de la consola le notifica cuando la rotación bajo demanda ha finalizado.

Iniciar la rotación de claves bajo demanda (AWS KMS API)

Puede utilizar la [API de AWS Key Management Service \(AWS KMS\)](#) para iniciar la rotación de claves bajo demanda y ver el estado de rotación actual de cualquier clave administrada por el cliente. En estos ejemplos se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

La [RotateKeyOnDemand](#) operación inicia inmediatamente la rotación de claves bajo demanda para la clave KMS especificada. Para identificar la clave KMS en estas operaciones, utilice el [ID de la clave](#) o el [ARN de la clave](#).

En el siguiente ejemplo, se inicia la rotación de claves bajo demanda en la clave KMS de cifrado simétrico especificada y se utiliza la [GetKeyRotationStatus](#) operación para comprobar que la rotación bajo demanda está en curso. La `OnDemandRotationStartDate` en la respuesta de `kms:GetKeyRotationStatus` identifica la fecha y la hora en que se inició una rotación bajo demanda en curso. En este ejemplo, la clave de KMS también tiene habilitada la rotación automática con un período de 365 días.

```
$ 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
}
```

Si la clave de KMS no admite la rotación automática o no tiene habilitada la rotación automática, la respuesta de `kms:GetKeyRotationStatus` tendría menos campos, como se muestra en el siguiente ejemplo:

```
$ 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"
}
```

Enumeración de las rotaciones y los materiales clave

Las claves de KMS que admiten la rotación de claves automática o bajo demanda pueden tener varios materiales clave asociados. Estas claves tienen un material clave inicial y un material clave adicional para cada rotación automática o bajo demanda.

Los usuarios autorizados con `kms:ListKeyRotations` permiso pueden usar la AWS KMS consola y la [ListKeyRotations](#) API para enumerar todos los materiales clave asociados a una clave KMS, incluidos los de las rotaciones automáticas y bajo demanda completadas.

Temas

- [Enumere las rotaciones y los materiales de claves \(consola\)](#)
- [Enumere las rotaciones y los materiales clave \(AWS KMS API\)](#)

Enumere las rotaciones y los materiales de claves (consola)

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.

4. Elija el alias o el ID de clave de una clave KMS.
5. Seleccione la pestaña Material y rotaciones de las claves.
 - La pestaña Material y rotaciones de las claves solo aparece en la página de detalles de las claves de KMS de cifrado simétrico que admiten la rotación automática o bajo demanda. Esto incluye las claves de KMS con el material clave que AWS KMS se generó (AWS_KMSorigen) y las claves de KMS con el material de clave importado (EXTERNALorigen).
 - La tabla de materiales de claves de la pestaña Material de claves y rotaciones muestra todos los materiales de claves asociados a la clave de KMS. Para cada material de claves, la entrada correspondiente muestra su identificador único asignado por AWS KMS, la fecha de rotación y el estado del material de claves. La fecha de rotación identifica cuándo se actualizó el material de claves tras una rotación de claves automática o bajo demanda. No hay ninguna fecha de rotación asociada al primer material ni al material de claves Pending rotation. El estado del material clave determina cómo se AWS KMS usa el material clave. El material clave actual se utiliza tanto para el cifrado como para el descifrado. Non-current el material clave solo se utiliza para el descifrado. Un estado del material de claves de Pending rotation indica que el material de claves está preparado para su rotación. Este material de claves no se utiliza para ninguna operación criptográfica hasta que una rotación de claves bajo demanda lo convierte en el material de claves actual. La información adicional que se muestra en el material de claves depende del tipo de clave de KMS.
 - En el caso de las claves de KMS de cifrado simétrico con origen AWS_KMS, cada fila también muestra el tipo de rotación, On-demand o Automatic.
 - Las claves KMS de cifrado simétrico con material clave importado (EXTERNALorigen) solo admiten la On-demand rotación, por lo que no hay ninguna columna de tipo de rotación. En su lugar, cada fila muestra un estado de importación, una descripción especificada por el usuario, información de caducidad y un menú Acciones. El estado de importación es Importado, lo que indica que el material clave está disponible en el interior AWS KMS o Pendiente de importación, lo que indica que el material clave no está disponible en el interior AWS KMS. El menú Acciones se puede utilizar para eliminar el material de claves importado o volver a importar el material de claves. La acción Eliminar material de claves está desactivada si el estado de importación del material de claves es Pendiente de importación. La acción Reimportar el material de claves siempre está disponible. No es necesario esperar a que un material de claves caduque o se elimine antes de volver a importarlo.

Enumere las rotaciones y los materiales clave (AWS KMS API)

Puede utilizar la [API de AWS Key Management Service \(AWS KMS\)](#) para iniciar la rotación de claves bajo demanda y ver el estado de rotación actual de cualquier clave administrada por el cliente. En estos ejemplos se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

La [ListKeyRotations](#) operación muestra todas las rotaciones y los materiales clave de la clave KMS especificada. Para identificar la clave KMS en estas operaciones, utilice el [ID de la clave](#) o el [ARN de la clave](#).

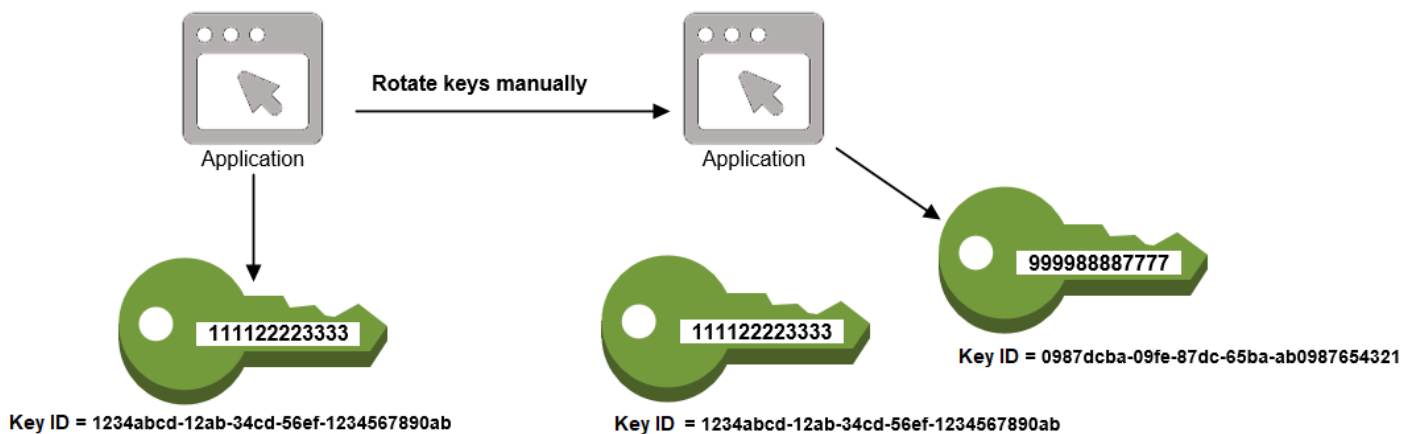
Esta operación admite un parámetro `IncludeKeyMaterial` opcional. El valor predeterminado de este parámetro es `ROTATIONS_ONLY`. Si omite este parámetro, AWS KMS devuelve información sobre los materiales clave creados mediante la rotación de claves automática o bajo demanda. Al especificar un valor de `ALL_KEY_MATERIAL`, AWS KMS agrega a la respuesta el primer material de claves y cualquier material de claves importado que esté pendiente de rotación. Este parámetro solo se puede usar con claves de KMS que admitan la rotación de claves automática o bajo 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
}
```

```
}
```

Rotación manual de claves

Es posible que le interese crear una nueva clave de KMS y utilizarla en sustitución de una clave de KMS actual en lugar de activar la rotación automática de claves. Cuando la nueva clave KMS tiene diferente material criptográfico que la clave KMS actual, el uso de la nueva clave KMS tiene el mismo efecto que cambiar el material de claves de una clave KMS existente. El proceso de sustitución de una clave KMS por otra se denomina rotación manual de claves.



La rotación manual es una buena opción si desea rotar claves KMS que no cumplen los requisitos para la rotación automática o bajo demanda, como las claves KMS asimétricas, las claves KMS HMAC y las claves KMS en almacenes de claves [personalizados](#).

Note

Cuando empiece a usar la nueva clave KMS, asegúrese de mantener habilitada la clave KMS original para AWS KMS poder descifrar los datos cifrados por la clave KMS original.

Cuando rota las claves KMS de forma manual, también debe actualizar las referencias al ID de la clave KMS o al ARN de la clave en sus aplicaciones. Los [alias](#), que asocian un nombre descriptivo a una clave KMS, pueden facilitar este proceso. Utilice un alias para hacer referencia a una clave KMS en sus aplicaciones. Después, cuando desee cambiar la clave KMS que utiliza la aplicación, en lugar editar su código de aplicación, cambie la clave KMS de destino del alias. Para obtener más información, consulte [Obtención de información sobre cómo utilizar alias en las aplicaciones](#).

Note

Los alias que apuntan a la última versión de una clave KMS girada manualmente son una buena solución para DescribeKey operaciones criptográficas como cifrar DeriveSharedSecret,,, GenerateDataKeyGenerateDataKeyPairGenerateMac, VerifyMacfirmar GetPublicKey verificar. Los alias no están permitidos en las operaciones que administran claves de KMS, como o. DisableKeyScheduleKeyDeletion. Al realizar la operación de descifrado en claves KMS de cifrado simétrico rotadas manualmente, omita el parámetro en el KeyId comando. AWS KMS utiliza automáticamente la clave KMS que cifró el texto cifrado.

El KeyId parámetro es obligatorio cuando se llama Decrypt o se verifica con una clave KMS asimétrica, o cuando se llama VerifyMac con una clave HMAC KMS. Estas solicitudes fallan cuando el valor del parámetro KeyId es un alias que ya no apunta a la clave de KMS que realizó la operación criptográfica, por ejemplo, cuando se rota una clave de forma manual. Para evitar este error, debe realizar un seguimiento y especificar la clave de KMS correcta para cada operación.

Para cambiar la clave KMS de destino de un alias, usa la UpdateAlias operación en la AWS KMS API. Por ejemplo, este comando actualiza el alias `alias/TestKey` para apuntar hacia una nueva clave KMS. Como la operación no devuelve ningún resultado, en el ejemplo se usa la ListAliases operación para mostrar que el alias ahora está asociado a una clave de KMS diferente y que el `LastUpdatedDate` campo está actualizado. Los ListAliases comandos utilizan el query parámetro de AWS CLI para obtener únicamente el `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
    },
  ]
}
```

Cambio de la clave principal en un conjunto de claves de varias regiones

Cada conjunto de claves de varias regiones relacionadas debe contar con una clave principal. Pero puede cambiar la clave principal. Esta acción, conocida como actualización de la región principal, convierte la clave principal actual en una clave de réplica y convierte una de las claves de réplica relacionadas en la clave principal. Puede hacerlo si necesita eliminar la clave principal actual mientras mantiene las claves de réplica, o para ubicar la clave principal en la misma región que los administradores de claves.

Puede seleccionar cualquier clave de réplica relacionada para que sea la nueva clave principal. Tanto la clave principal como la clave de réplica deben estar en el [estado clave](#) Enabled Cuando se inicia la operación.

El estado clave de **Updating**

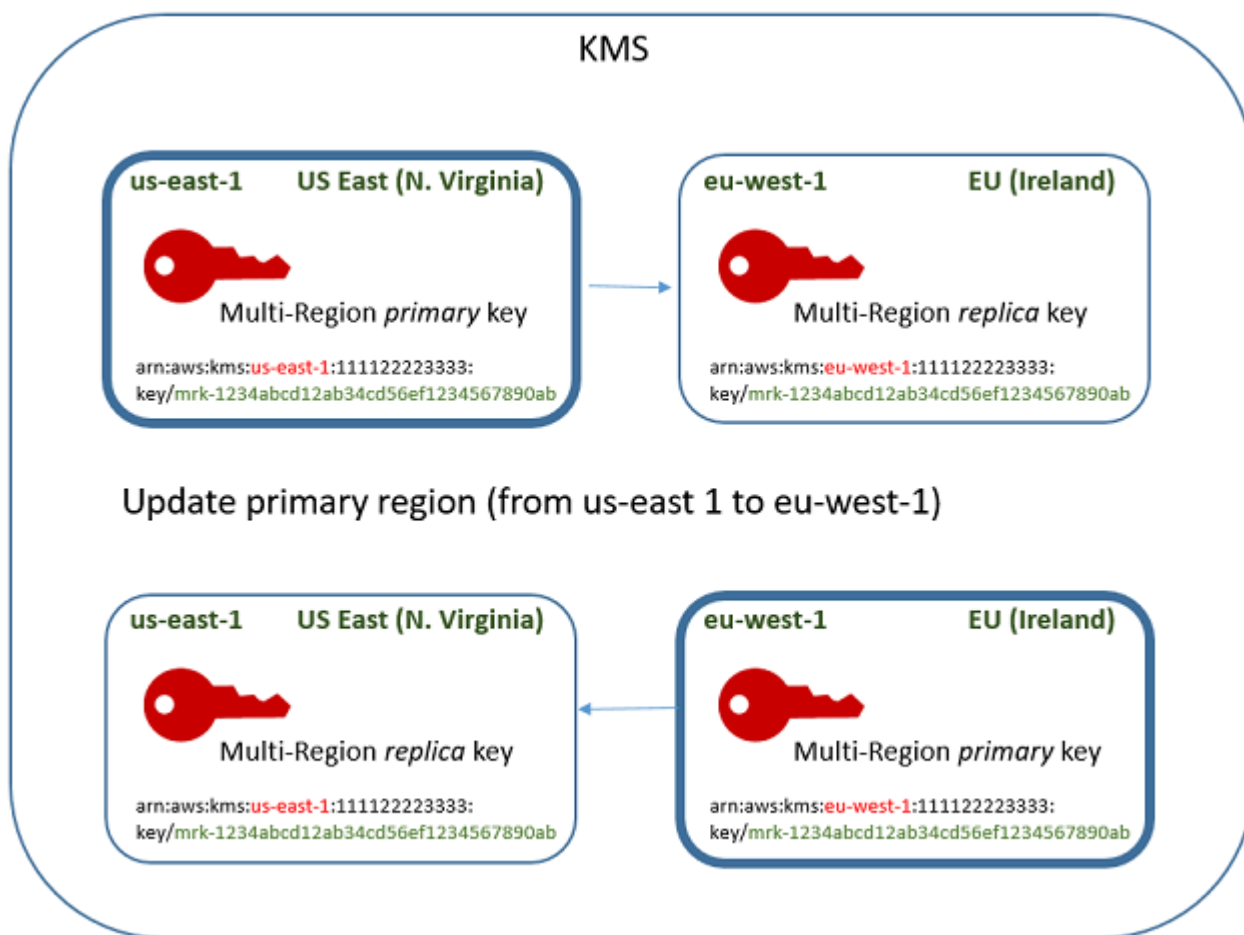
Incluso después de completar la operación `UpdatePrimaryRegion`, es posible que el proceso de actualización de la región principal siga en curso durante unos segundos más. Durante este tiempo, las claves principales antiguas y nuevas tienen un estado de clave transitoria de [Updating](#) (Actualizando). Mientras que el estado de clave es `Updating`, puede usar las claves en operaciones criptográficas, pero no puede replicar la nueva clave principal ni realizar determinadas operaciones de administración, como habilitar o desactivar estas claves. Operaciones como, por ejemplo, [DescribeKey](#) pueden mostrar las claves principales antiguas y

nuevas como réplicas. El estado de clave `Enabled` se restaura cuando se haya completado la actualización.

Para obtener información acerca del efecto del estado de clave `Updating`, consulte [Estados clave de AWS KMS claves](#).

Funcionamiento

Suponga que tiene una clave principal en EE. UU. Este (Norte de Virginia) (`us-east-1`) y una réplica de claves en Europa (Irlanda) (`eu-west-1`). Puede utilizar la función de actualización para cambiar la clave principal en EE. UU. Este (Norte de Virginia) (`us-east-1`) a una clave de réplica y cambiar la clave de réplica en Europa (Irlanda) (`eu-west-1`) a la clave principal.



Cuando se completa el proceso de actualización, la clave de varias regiones en la región Europa (Irlanda) (`eu-west-1`) es una clave principal de varias regiones y la clave en la región EE. UU. Este (Norte de Virginia) (`us-east-1`) es su clave de réplica. Si hay otras claves de réplica relacionadas, se convierten en réplicas de la nueva clave principal. La próxima vez que AWS KMS sincronice las propiedades compartidas de las claves multirregionales, obtendrá

las [propiedades compartidas](#) de la nueva clave principal y las copiará en sus claves de réplica, incluida la clave principal anterior.

La operación de actualización no modifica el [ARN de clave](#) de ninguna clave de varias regiones. Tampoco afecta a las propiedades compartidas, como el material clave, ni a las propiedades independientes, como la política de claves. Sin embargo, es posible que desee [actualice la política de claves](#) de la nueva clave principal. Por ejemplo, es posible que desee añadir el ReplicateKey permiso [kms:](#) para entidades de confianza a la nueva clave principal y eliminarlo de la nueva clave de réplica.

Actualización de la región principal

Puede convertir una clave de réplica en una clave principal, lo que cambia la clave principal anterior en una réplica. Para actualizar la región principal, necesitas el UpdatePrimaryRegion permiso [kms:](#) en ambas regiones.

Puede actualizar la región principal en la AWS KMS consola o mediante la [UpdatePrimaryRegion](#) operación.

Uso de la AWS KMS consola

Puede actualizar la clave principal de la AWS KMS consola. Comience con la página de detalles de clave de la clave principal actual.

1. Inicia sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrela en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Seleccione el alias o el ID de clave de la [clave principal de varias regiones](#). Esto abre la página de detalles de clave de la clave principal.

Para identificar una clave principal de varias regiones, utilice el icono de herramienta situado en la esquina superior derecha para agregar la columna Regionality (Regionalidad) de la tabla.

5. Elija la pestaña Regionality (Regionalidad).
6. En la sección Primary key (Clave principal), seleccione Change primary Region (Cambiar de región principal).

7. Elija la región de la nueva clave principal. Solo puede elegir una región del menú.

El menú Change primary Regions (Cambiar las regiones principales) incluye solo Regiones que tienen una clave de varias regiones relacionada. Es posible que no tenga [permiso para actualizar la región principal](#) en todas las regiones del menú.

8. Seleccione Change primary Region (Cambiar región principal).

Uso de la API AWS KMS

Para cambiar la clave principal de un conjunto de claves multirregionales relacionadas, utilice la [UpdatePrimaryRegion](#) operación.

Use el parámetro KeyId para identificar la clave principal actual. Utilice el PrimaryRegion parámetro para indicar Región de AWS la nueva clave principal. Si la clave principal aún no tiene una réplica en la nueva región principal, se produce un error en la operación.

En el ejemplo siguiente se cambia la clave principal de la clave de varias regiones en la región us-west-2 a su réplica en la región eu-west-1. El parámetro KeyId identifica la clave principal actual en la región us-west-2. El PrimaryRegion parámetro especifica Región de AWS la nueva clave principal, 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
```

Cuando se realiza correctamente, esta operación no devuelve ningún resultado; solo el código de estado HTTP. Para ver el efecto, ejecute la [DescribeKey](#) operación en cualquiera de las claves multirregionales. Es posible que desee esperar hasta que el estado de la clave vuelva a Enabled. Mientras que el estado de clave es [Updating](#) (Actualizando), los valores de la clave aún pueden estar en flujo.

Por ejemplo, la siguiente llamada DescribeKey obtiene los detalles acerca de la clave de varias regiones en la región eu-west-1. La salida muestra que la clave de varias regiones de la región eu-west-1 es ahora la clave principal. La clave de varias regiones relacionada (mismo ID de clave) en la región us-west-2 es ahora una clave 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"
        }
      ]
    }
  }
}
```

Eliminar un AWS KMS key

Eliminar un AWS KMS key es destructivo y potencialmente peligroso. Elimina el material de claves y todos los metadatos asociados con la clave KMS. Esta acción es irreversible. Una vez que se elimina una clave KMS, ya no pueden descifrar los datos que se habían cifrado con ella, lo que significa que no se pueden recuperar. (Las únicas excepciones son las [claves de réplica de varias regiones](#) y las claves HMAC de KMS y asimétricas con material de claves importado). Este riesgo es importante en el caso de [las claves KMS asimétricas que se utilizan para el cifrado](#), ya que, sin previo aviso ni error, los usuarios pueden seguir generando textos cifrados con la clave pública que no se pueden descifrar una vez eliminada la clave privada. AWS KMS

Debe eliminar una clave KMS solo cuando esté seguro de que ya no necesita usarla. Si no está seguro, considere la posibilidad de [desactivar la clave KMS](#) en lugar de eliminarla. Puede volver a habilitar una clave de KMS deshabilitada y [cancelar la eliminación programada](#) de una clave de KMS, pero no puede recuperar una clave de KMS eliminada.

Solo puede programar la eliminación de una clave administrada por el cliente. No puede eliminar o. Claves administradas por AWS Claves propiedad de AWS

Antes de eliminar una clave KMS, es posible que desee saber cuántos textos cifrados se cifraron con esa clave KMS. AWS KMS no almacena esta información ni almacena ninguno de los textos cifrados. Para obtener esta información, debe determinar por su cuenta el uso anterior de una clave KMS. Para obtener ayuda, consulte [Determinación del uso anterior de una clave KMS](#).

AWS KMS nunca elimina sus claves de KMS a menos que las programe explícitamente para que se eliminen y venza el período de espera obligatorio.

Sin embargo, puede decidir eliminar una clave KMS por uno o varios de los motivos siguientes:

- Para completar el ciclo de vida de clave de las claves KMS que ya no necesita
- Evitar los gastos generales de administración y los [costos](#) asociados con el mantenimiento de las claves KMS no usadas
- Para reducir el número de claves KMS que se contabilizan para su [cuota de recursos de clave KMS](#)

 Note

Si [cierras las tuyas Cuenta de AWS](#), tus claves de KMS se vuelven inaccesibles y ya no se te facturará por ellas.

AWS KMS registra una entrada en su AWS CloudTrail registro cuando [programa la eliminación](#) de la clave KMS y cuando se elimina [realmente la clave KMS](#).

Acerca del período de espera

Como eliminar una clave de KMS es destructivo y potencialmente peligroso, es AWS KMS necesario establecer un período de espera de 7 a 30 días. El periodo de espera predeterminado es de 30 días.

Sin embargo, el período de espera real puede ser hasta 24 horas más largo que el programado. Para obtener la fecha y la hora reales en las que se eliminará la clave KMS, utilice la [DescribeKey](#) operación. O en el consola AWS KMS , en la [página de detalles](#) para la clave KMS, en la sección Configuración general, consulte la eliminación programada. Asegúrese de anotar la zona horaria.

Durante el periodo de espera, el estado de la clave KMS y el estado de la clave son Pending deletion (Pendiente de eliminación).

- Una clave KMS que está pendiente de eliminación no puede utilizarse en ninguna [operación criptográfica](#).
- AWS KMS no [rota el material clave de las](#) claves de KMS que están pendientes de ser eliminadas.

Una vez finalizado el período de espera, AWS KMS elimina la clave KMS, sus alias y todos los metadatos relacionados AWS KMS .

Es posible que programar la eliminación de una clave de KMS no afecte inmediatamente a las claves de datos cifradas por la clave de KMS. Para obtener más información, consulte [Cómo afectan las claves de KMS obsoletas a las claves de datos](#).

Use el periodo de espera para asegurarse de que no necesita la clave KMS ahora ni en el futuro. Puedes [configurar una CloudWatch alarma de Amazon](#) para que te avise si una persona o aplicación intenta usar la clave KMS durante el período de espera. Para recuperar la clave KMS, puede

cancelar la eliminación de claves antes de que finalice el periodo de espera. Una vez finalizado el periodo de espera, no podrá cancelar la eliminación de la clave y AWS KMS eliminará la clave KMS.

Consideraciones especiales

Antes de programar la eliminación de las claves, revise las siguientes consideraciones especiales para eliminar las claves KMS de propósito especial.

Eliminación de claves KMS asimétricas

Los usuarios [autorizados](#) pueden eliminar las claves KMS simétricas y asimétricas. El procedimiento para programar la eliminación de estas claves KMS es el mismo para ambos tipos de claves. Sin embargo, dado que la [clave pública de una clave KMS asimétrica se puede descargar](#) y utilizar fuera de ella AWS KMS, la operación plantea riesgos adicionales importantes, especialmente en el caso de las claves KMS asimétricas que se utilizan para el cifrado (el uso de la clave es). ENCRYPT_DECRYPT

- Cuando planifica la eliminación de una clave KMS, el estado de la clave de la clave KMS cambia a Pending deletion (Pendiente de eliminación) y la clave KMS no se puede utilizar en [operaciones criptográficas](#). Sin embargo, programar la eliminación no tiene ningún efecto en las claves públicas ajenas a AWS KMS. Los usuarios que tengan las claves públicas pueden seguir utilizándolas para cifrar mensajes. No reciben ninguna notificación de que se haya cambiado el estado de la clave. A menos que se cancele la eliminación, el texto cifrado creado con la clave pública no se puede descifrar.
- Las alarmas, los registros y otras estrategias que detectan los intentos de uso de la clave KMS que está pendiente de eliminación no pueden detectar el uso de la clave pública fuera de AWS KMS.
- Cuando se elimina la clave KMS, todas AWS KMS las acciones relacionadas con esa clave KMS fallan. Sin embargo, los usuarios que tengan las claves públicas pueden seguir utilizándolas para cifrar mensajes. Estos textos cifrados no se pueden descifrar.

Si debe eliminar una clave KMS asimétrica con un uso de clave igual a ENCRYPT_DECRYPT, utilice las entradas de CloudTrail registro para determinar si la clave pública se ha descargado y compartido. Si ha sido así, compruebe que la clave pública no se utilice fuera de AWS KMS. Después, considere la posibilidad de [desactivar la clave KMS](#) en lugar de eliminarla.

El riesgo que supone eliminar una clave de KMS asimétrica se mitiga en el caso de las claves de KMS asimétricas con material de claves importado. Para obtener más información, consulte [Deleting KMS keys with imported key material](#).

Eliminación de claves de varias regiones

Para eliminar una clave principal, debe programar la eliminación de todas sus claves de réplica y esperar a que se eliminen las claves de réplica. El período de espera necesario para eliminar una clave principal comienza cuando se elimina la última de sus claves de réplica. Si debe eliminar una clave principal de una región concreta sin eliminar sus claves de réplica, cambie la clave principal por una clave de réplica mediante [actualización de la región principal](#).

Puede eliminar una clave réplica en cualquier momento. No depende del estado de la clave de ninguna otra clave KMS. Si elimina por error una clave de réplica, puede volver a crearla replicando la misma clave primaria en la misma región. La nueva clave de réplica que cree tendrá las mismas [propiedades compartidas](#) que la clave de réplica original.

Eliminación de claves KMS con material de claves importado

Eliminar el material de claves de una clave de KMS con material de claves importado es temporal y reversible. Para restaurar la clave, vuelva a importar su material de claves.

Por el contrario, eliminar una clave KMS es irreversible. Si [programa la eliminación de la clave](#) y vence el período de espera requerido, eliminará de AWS KMS forma permanente e irreversible la clave KMS, su material clave y todos los metadatos asociados a la clave KMS.

Sin embargo, el riesgo y las consecuencias de eliminar una clave de KMS con material de claves importado dependen del tipo (“especificación de clave”) de la clave de KMS.

- Claves de cifrado simétrico: si elimina una clave de KMS de cifrado simétrico, no se podrán recuperar los textos cifrados restantes cifrados con esa clave. No puede crear una nueva clave de KMS de cifrado simétrico que pueda descifrar los textos cifrados de una clave de KMS de cifrado simétrico eliminada, incluso si tiene el mismo material de claves. Los metadatos exclusivos de cada clave de KMS están enlazados criptográficamente a cada texto cifrado simétrico. Esta característica de seguridad garantiza que solo la clave de KMS que cifró el texto cifrado simétrico pueda descifrarlo, pero le impide volver a crear una clave de KMS equivalente.
- Claves asimétricas y HMAC: si dispone del material clave original, puede crear una nueva clave KMS con las mismas propiedades criptográficas que una clave KMS asimétrica o HMAC que se haya eliminado. AWS KMS genera firmas y textos cifrados RSA estándar, firmas ECC y etiquetas HMAC, que no incluyen ninguna característica de seguridad exclusiva. Además, puede utilizar una clave HMAC o la clave privada de un par de claves asimétricas fuera de AWS.

Una clave de KMS nueva que cree con el mismo material de clave asimétrica o HMAC tendrá un identificador de clave diferente. Tendrá que crear una nueva política de claves, volver a

crear los alias y actualizar las políticas y concesiones de IAM existentes para hacer referencia a la nueva clave.

Eliminar claves KMS de un almacén de claves AWS CloudHSM

Al programar la eliminación de una clave KMS de un almacén de AWS CloudHSM claves, su [estado de clave](#) cambia a Pending deletion (Eliminación pendiente) durante todo el periodo de espera, incluso si la clave de KMS deja de estar disponible porque [ha desconectado el almacén de claves personalizado](#). Esto permite cancelar la eliminación de la clave KMS en cualquier momento durante el período de espera.

Cuando vence el período de espera, AWS KMS elimina la clave KMS de AWS KMS. A continuación, AWS KMS hace todo lo posible por eliminar el material clave del AWS CloudHSM clúster asociado. Si AWS KMS no puede eliminar el material de claves, como, por ejemplo, cuando el almacén de claves está desconectado de AWS KMS, es probable que tenga que [eliminar el material de claves huérfano](#) manualmente del clúster.

AWS KMS no elimina el material clave de las copias de seguridad del clúster. Incluso si elimina la clave de KMS AWS KMS y elimina su material clave del AWS CloudHSM clúster, los clústeres creados a partir de las copias de seguridad pueden contener el material clave eliminado. Para eliminar permanentemente el material clave, utilice la [DescribeKey](#) operación para identificar la fecha de creación de la clave KMS. A continuación, [elimine todas las copias de seguridad del clúster](#) que puedan contener el material de claves.

Al programar la eliminación de una clave de KMS de un almacén de AWS CloudHSM claves, la clave de KMS queda inutilizable de inmediato (sujeto a una posible coherencia). Sin embargo, los recursos cifrados con [claves de datos](#) protegidas por la clave de KMS no se ven afectados hasta que se vuelva a utilizar la clave de KMS, por ejemplo, para descifrar la clave de datos. Este problema afecta a Servicios de AWS muchos de los cuales utilizan claves de datos para proteger sus recursos. Para obtener más información, consulte [Cómo afectan las claves de KMS obsoletas a las claves de datos](#).

Eliminación de claves KMS de un almacén de claves externo

La eliminación de una clave de KMS de un almacén de claves externo no afecta a la [clave externa](#) que sirvió como material de claves.

Cuando programa la eliminación de una clave de KMS de un almacén de claves externo, su [estado de clave](#) cambia a Pending deletion (Eliminación pendiente). La clave de KMS conservará el estado Pending deletion (Eliminación pendiente) durante todo el periodo de espera, incluso si

la clave de KMS deja de estar disponible porque [ha desconectado el almacén de claves externo](#). Esto permite cancelar la eliminación de la clave KMS en cualquier momento durante el período de espera. Cuando caduque el período de espera, AWS KMS elimina la clave KMS de AWS KMS.

Al programar la eliminación de una clave de KMS de un almacén de claves externo, la clave de KMS queda inutilizable de inmediato (sujeto a la posible coherencia). Sin embargo, los recursos cifrados con [claves de datos](#) protegidas por la clave de KMS no se ven afectados hasta que se vuelva a utilizar la clave de KMS, por ejemplo, para descifrar la clave de datos. Este problema afecta a los Servicios de AWS, muchos de los cuales utilizan claves de datos para proteger sus recursos. Para obtener más información, consulte [Cómo afectan las claves de KMS obsoletas a las claves de datos](#).

Control del acceso a la eliminación de claves

Si utiliza políticas de IAM para conceder AWS KMS permisos, las identidades de IAM que tienen acceso de AWS administrador ("Action": "*") o acceso AWS KMS total ("Action": "kms:*") ya pueden programar y cancelar la eliminación clave de las claves de KMS. Para permitir que los administradores de claves programen y cancelen la eliminación de claves en la política de claves, utilice la AWS KMS consola o la AWS KMS API.

Normalmente, solo los administradores de claves tienen permiso para programar o cancelar la eliminación de claves. Sin embargo, puede conceder estos permisos a otras identidades de IAM al agregar los permisos `kms:ScheduleKeyDeletion` y `kms:CancelKeyDeletion` a la política de claves o a una política de IAM. También puedes usar la clave de [kms:ScheduleKeyDeletionPendingWindowInDays](#) condición para restringir aún más los valores que los directores pueden especificar en el `PendingWindowInDays` parámetro de una [ScheduleKeyDeletion](#) solicitud.

Permitir a los administradores de claves programar y cancelar la eliminación de claves

Uso de la consola AWS KMS

Para otorgar permiso a los administradores de claves para programar y cancelar la eliminación de claves

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.

2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Elija el alias o el ID de clave de la clave KMS cuyos permisos desea cambiar.
5. Seleccione la pestaña key policy (política de claves).
6. El siguiente paso es diferente para la vista predeterminada y la vista de política de su política de claves. La vista predeterminada solo está disponible si utiliza la política de claves de consola predeterminada. De lo contrario, solo está disponible la vista de política.

Cuando la vista predeterminada está disponible, aparece el botón Switch to policy view (Cambiar a la vista de política) o Switch to default view (Cambiar a la vista predeterminada) en la pestaña Key policy (Política de claves).

- En la vista predeterminada:
 - En la sección Key deletion (Eliminación de la clave), seleccione Allow key administrators to delete this key (Permitir que los administradores de claves eliminen esta clave).
- En la vista de la política:
 - a. Elija Edit (Edición de).
 - b. En la declaración de política para los administradores de claves, agregue los permisos `kms:ScheduleKeyDeletion` y `kms:CancelKeyDeletion` al 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. Seleccione Save changes (Guardar cambios).

Uso de la API AWS KMS

Puede utilizar el AWS Command Line Interface para añadir permisos para programar y cancelar la eliminación de claves.

Para agregar permiso para programar y cancelar la eliminación de claves

1. Utilice el comando [aws kms get-key-policy](#) para recuperar la política de claves existente y, a continuación, guarde el documento de políticas en un archivo.
2. Abra el documento de políticas en el editor de textos que prefiera. En la declaración de política para los administradores de claves, agregue los permisos `kms:ScheduleKeyDeletion` y `kms:CancelKeyDeletion`. El siguiente ejemplo muestra una declaración de política con estos dos permisos:

```
{
  "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. Utilice el comando [aws kms put-key-policy](#) para aplicar la política de claves a la clave KMS.

Programar la eliminación de claves

Los siguientes procedimientos describen cómo programar la eliminación de claves y cancelar la eliminación de claves AWS KMS keys (claves de KMS) AWS KMS mediante la API Consola de administración de AWS y la AWS KMS API.

Warning

La eliminación de una clave KMS es un proceso destructivo y potencialmente peligroso. Solo debe hacerlo si está seguro de que ya no necesitará la clave KMS y no tendrá que usarla en el futuro. Si no está seguro, debe [desactivar la clave KMS](#) en lugar de eliminarla.

Para poder eliminar una clave KMS, debe disponer de permiso para hacerlo. Para obtener información sobre cómo conceder estos permisos a los administradores de claves, consulte [Control del acceso a la eliminación de claves](#). También puede usar la clave de condición [kms:ScheduleKeyDeletionPendingWindowInDays](#) para restringir aún más el periodo de espera, por ejemplo, para imponer un periodo de espera mínimo.

AWS KMS registra una entrada en el AWS CloudTrail registro cuando se [programa la eliminación](#) de la clave de KMS y cuando se [elimina realmente la clave de KMS](#).

Uso de la AWS KMS consola

En la Consola de administración de AWS, puede programar y cancelar la eliminación de varias claves de KMS a la vez.

Para programar la eliminación de claves

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.

No puede programar la eliminación de [Claves administradas por AWS](#) o [Claves propiedad de AWS](#).

4. Seleccione la casilla de verificación situada junto a la clave de KMS que desea eliminar.
5. Elija Key actions (Acciones de claves), Schedule key deletion (Programar la eliminación de claves).
6. Lea y tenga en cuenta la advertencia y la información sobre la cancelación de la eliminación durante el período de espera. La advertencia incluye el número de claves seleccionadas que se utilizaron recientemente en operaciones criptográficas. La tabla muestra una columna Utilizada por última vez que indica cuándo se utilizó por última vez cada clave de KMS. Si decide cancelar la eliminación, en la parte inferior de la página, elija Cancel (Cancelar).
7. En Waiting period (in days) [Período de espera (en días)], indique un número de días entre 7 y 30.
8. Revise las claves KMS que está eliminando.
9. Seleccione la casilla de verificación situada junto a Confirmo que deseo programar la eliminación de estas claves. Las claves quedarán inutilizables inmediatamente para las operaciones criptográficas. Después del período de espera de un **<number of days>** día, las claves se eliminarán permanentemente.
10. Elija Schedule deletion.

El estado de clave KMS cambia a Pending deletion (Eliminación pendiente).

Uso de la AWS KMS API

Utilice el comando [aws kms schedule-key-deletion](#) para programar la eliminación de una [clave administrada por el cliente](#), tal y como se muestra en el siguiente ejemplo.

No puede programar la eliminación de un Clave administrada de AWS o Clave propiedad de AWS.

```
$ aws kms schedule-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --  
pending-window-in-days 10
```

Si se usa correctamente, AWS CLI devuelve un resultado como el que se muestra en el siguiente ejemplo:

```
{
```

```
"KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
"DeletionDate": 1598304792.0,
"KeyState": "PendingDeletion",
"PendingWindowInDays": 10
}
```

Cancelación de la eliminación de claves

Después de [programar la eliminación de una clave KMS](#), puede cancelar la eliminación de la clave mientras aún esté en estado [pendiente de eliminación](#). Puede cancelar la eliminación de claves en la AWS KMS consola o mediante la [CancelKeyDeletion](#) operación. Tras cancelar la eliminación pendiente de una clave KMS, el estado de la clave KMS es Disabled. Para obtener más información sobre la habilitación de claves KMS, consulte [Habilitar y deshabilitar claves](#).

Uso de la AWS KMS consola

Para cancelar la eliminación de claves

1. Abra la AWS KMS consola en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Seleccione la casilla de verificación situada junto a la clave de KMS que desea recuperar.
5. Elija Key actions (Acciones de claves), Cancel key deletion (Cancelar eliminación de la clave).

El estado de clave KMS cambia de Pending deletion (Eliminación pendiente) a Disabled (Deshabilitado). Para utilizar la clave KMS, debe [habilitarla](#).

Uso de la API AWS KMS

Utilice el [aws kms cancel-key-deletion](#) comando para cancelar la eliminación de claves del AWS CLI , como se muestra en el siguiente ejemplo.

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

Si se utiliza correctamente, AWS CLI devuelve un resultado similar al que se muestra en el siguiente ejemplo:

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

El estado de la clave KMS cambia de Pending Deletion (Eliminación pendiente) a Disabled (Deshabilitada). Para utilizar la clave de KMS, debe [habilitarla](#).

Creación de una alarma que detecte el uso de una eliminación pendiente de una clave KMS

Puedes combinar las funciones de AWS CloudTrail Amazon CloudWatch Logs y Amazon Simple Notification Service (Amazon SNS) para crear una alarma de Amazon que te notifique cuando alguien de tu cuenta intente usar una clave de KMS que está pendiente de ser eliminada. CloudWatch Si recibe esta notificación, puede cancelar la eliminación de la clave KMS y reconsiderar su decisión de eliminarla.

Los siguientes procedimientos crean una alarma que le notifica cada vez que se escribe el mensaje de error **Key ARN is pending deletion** en sus CloudTrail archivos de registro. Este mensaje de error indica que una persona o aplicación ha intentado utilizar la clave KMS en una [operación criptográfica](#). Debido a que la notificación está vinculada al mensaje de error, no se activa cuando se utilizan operaciones de las API permitidas en las clave KMS que están pendientes de eliminación, como por ejemplo ListKeys, CancelKeyDeletion y PutKeyPolicy. Para ver una lista de las operaciones de la AWS KMS API que devuelven este mensaje de error, consulte [Estados clave de AWS KMS claves](#).

El correo electrónico de notificación que recibe no muestra la clave KMS o la operación criptográfica. Puede encontrar esa información en [su registro de CloudTrail](#). En lugar de ello, el correo electrónico informa de que el estado de la alarma ha cambiado de OK (Correcto) a Alarm (Alarma). Para obtener más información sobre CloudWatch las alarmas y los cambios de estado, consulta [Uso de CloudWatch las alarmas de Amazon](#) en la Guía del CloudWatch usuario de Amazon.

Warning

Esta CloudWatch alarma de Amazon no puede detectar el uso de la clave pública de una clave KMS asimétrica fuera de AWS KMS. Para obtener detalles acerca de los riesgos especiales de la eliminación de las claves KMS asimétricas utilizadas para la criptografía de

clave pública, incluida la creación de textos cifrados que no se pueden descifrar, consulte [Deleting asymmetric KMS keys](#).

En este procedimiento, se crea un filtro de métricas de grupos de CloudWatch registros que busca instancias de la excepción de eliminación pendiente. A continuación, crea una CloudWatch alarma basada en la métrica del grupo de registros. Para obtener información sobre los filtros de métricas de grupos de registros, consulte [Creación de métricas a partir de eventos de registro mediante filtros](#) en la Guía del usuario de Amazon CloudWatch Logs.

1. Cree un filtro de CloudWatch métricas que analice los CloudTrail registros.

Siga las instrucciones que se indican en [Crear un filtro de métricas para un grupo de registro](#) con los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Patrón de filtro	<code>{ \$.eventSource = kms* && \$.errorMessage = "* is pending deletion." }</code>
Valor de la métrica	1

2. Cree una CloudWatch alarma basada en el filtro de métricas que creó en el paso 1.

Siga las instrucciones de [Crear una CloudWatch alarma basada en un filtro métrico de grupos de registros](#) utilizando los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Filtro de métricas	El nombre del filtro de métricas que ha creado en el Paso 1.
Tipo de umbral	Estático
Condiciones	Siempre que sea <i>metric-name</i> mayor/igual que 1

Campo	Valor
Puntos de datos para alarma	1 fuera de 1
Tratamiento de datos que faltan	Treat missing data as good (not breaching threshold) (Tratar los datos que faltan como buenos [dentro del umbral])

Después de completar este procedimiento, recibirá una notificación cada vez que su nueva CloudWatch alarma entre en estado. ALARM Si recibe una notificación para esta alarma, puede significar que todavía se necesita una eliminación programada para cifrar o descifrar datos. En ese caso, [cancele la eliminación de la clave KMS](#) y reconsidere su decisión de eliminarla.

Eliminación del material de claves importado

Puede eliminar el material de claves importado desde una clave de KMS en cualquier momento. Además, cuando el material clave importado con fecha de caducidad caduca, lo AWS KMS elimina. En cualquier caso, cuando se elimina el material de claves, el [estado de la clave](#) de KMS cambia a importación pendiente, y la clave de KMS no puede utilizarse en ninguna operación criptográfica.

Las claves de cifrado simétricas pueden tener varios materiales clave asociados. Para estas claves, KMS asigna un identificador único a cada material de claves. Puede usar la [ListKeyRotationsAPI](#) para ver estos identificadores del material clave y el estado del material clave correspondiente (consulte [RotationsListEntry](#)). El estado del material clave es igual PENDING_ROTATION o PENDING_MULTI_REGION_IMPORT_AND_ROTATION indica que el material clave no está asociado permanentemente a la clave de KMS. Al eliminar o caducar cualquier material clave asociado de forma permanente, el estado de la clave pasa a Pendiente de importación. Para eliminar un material clave específico, especifique su identificador mediante el `key-material-id` parámetro de la [DeleteImportedKeyMaterialAPI](#).

Consideraciones sobre las claves multirregionales

- Al eliminar el material clave de una clave de región principal que esté en un PENDING_MULTI_REGION_IMPORT_AND_ROTATION estado PENDING_ROTATION o estado, también eliminará los materiales clave de las claves de región réplicas.

- Si elimina el material clave de una clave de región principal o de réplica, solo se verá afectada esa clave específica y las demás claves multirregionales relacionadas permanecerán inalteradas. Todas las claves de región principales o réplicas que tengan todos sus materiales clave asociados de forma permanente seguirán utilizándose en las operaciones criptográficas.

Warning

El parámetro `key-material-id` es opcional y, si no lo especifica, AWS KMS eliminará el material de claves actual.

Además de deshabilitar la clave de KMS y retirar los permisos, la eliminación del material de claves se puede utilizar como estrategia para detener el uso de la clave de KMS de forma rápida, pero temporal. Por el contrario, si se programa la eliminación de una clave de KMS con material de claves importado, también se detiene rápidamente el uso de la clave de KMS. Sin embargo, si la eliminación no se cancela durante el periodo de espera, la clave de KMS, el material de claves asociados y todos los metadatos clave se eliminan de forma permanente. Para obtener más información, consulte [Deleting KMS keys with imported key material](#).

Para eliminar el material clave, puede utilizar la AWS KMS consola o la operación de la [DeleteImportedKeyMaterial](#) API. AWS KMS registra una entrada en su AWS CloudTrail registro cuando [elimina el material clave importado](#) y cuando [AWS KMS elimina el material clave caducado](#).

Cómo afecta AWS a los servicios la eliminación de material clave

Cuando se elimina el material de claves, la clave de KMS se vuelve inutilizable de forma inmediata (sujeto a posible coherencia). Sin embargo, los recursos cifrados con [claves de datos](#) protegidas por la clave de KMS no se ven afectados hasta que se vuelva a utilizar la clave de KMS, por ejemplo, para descifrar la clave de datos. Este problema afecta a Servicios de AWS muchos de los cuales utilizan claves de datos para proteger sus recursos. Para obtener más información, consulte [Cómo afectan las claves de KMS obsoletas a las claves de datos](#).

Uso de la AWS KMS consola

Puede utilizar la AWS KMS consola para eliminar material clave.

1. Inicia sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrela en <https://console.aws.amazon.com/kms>.

2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.
4. Realice una de las siguientes acciones:
 - Seleccione la casilla de verificación de una clave KMS con material de claves importado. Elija Key actions, Delete key material. En el caso de las claves de cifrado simétricas que tienen varios materiales de claves asociados, se eliminará el material de claves actual.
 - Para claves de KMS de cifrado simétrico con material de clave importado, elija el alias o el ID de clave de una clave de KMS. Seleccione la pestaña Material y rotaciones de las claves. La tabla de materiales de claves mostrará una lista de todos los materiales de claves asociados a la clave. Seleccione Eliminar el material de claves en el menú Acciones de la fila correspondiente al material de claves que desee eliminar.
5. Confirme que desea eliminar el material de claves y, a continuación, seleccione Delete key material. El estado de la clave KMS, que corresponde a su [estado de clave](#), cambia a Pending import (Importación pendiente). Si el material de claves eliminado estaba en el estado PENDING_ROTATION, no habrá ningún cambio en el estado de la clave de KMS.

Uso de la API AWS KMS

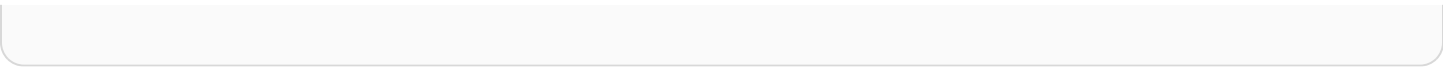
Para usar la [AWS KMS API](#) para eliminar material clave, envía una [DeleteImportedKeyMaterial](#) solicitud. El siguiente ejemplo muestra cómo hacerlo con la [AWS CLI](#).

Sustituya **1234abcd-12ab-34cd-56ef-1234567890ab** por el ID de clave de la clave KMS cuyo material de claves desea eliminar. Puede usar el ID de clave o el ARN de la clave KMS, pero no puede usar un alias para esta operación. El siguiente comando elimina el material de claves actual, que puede ser el único material de claves asociado a la clave.

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

Para eliminar un material de claves específico, especifique el material de claves identificado mediante el parámetro key-material-id. Sustituya **123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0** por el ID del material de claves que desea eliminar.

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --key-material-id 123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0
```



Generación de claves de datos

Las claves de datos son las claves simétricas que puede utilizar para cifrar los datos, incluidas grandes cantidades de datos y otras claves de cifrado de datos. A diferencia de las claves KMS simétricas, que no se pueden descargar, las claves de datos se le devuelven para su uso fuera de AWS KMS.

Cuando AWS KMS genera claves de datos, devuelve una clave de datos en texto simple para su uso inmediato (opcional) y una copia cifrada de la clave de datos que puede almacenar de forma segura junto con los datos. Cuando esté listo para descifrar los datos, primero pida AWS KMS que se descifre la clave de datos cifrada.

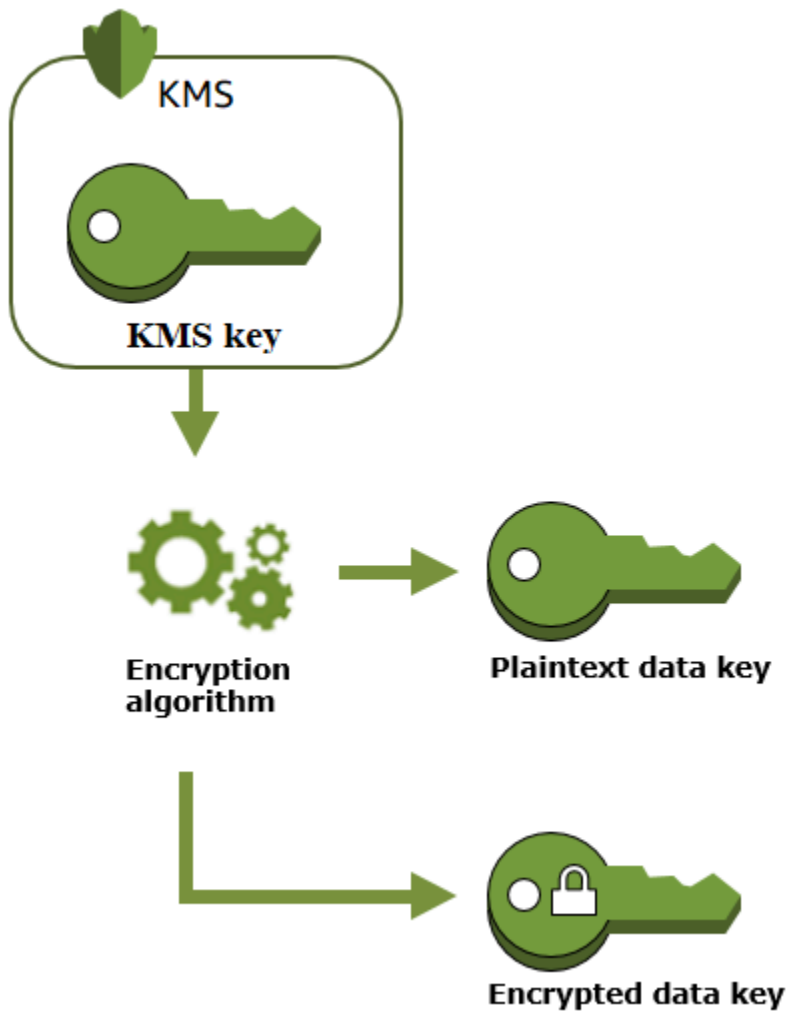
AWS KMS genera, cifra y descifra las claves de datos. Sin embargo, AWS KMS no almacena, administra ni rastrea las claves de datos, ni realiza operaciones criptográficas con las claves de datos. Debe usar y administrar las claves de datos fuera de AWS KMS. Para obtener ayuda para utilizar las claves de datos de forma segura, consulte el [AWS Encryption SDK](#).

Temas

- [Crear una clave de datos](#)
- [Cómo funcionan las operaciones criptográficas con claves de datos](#)
- [Cómo afectan las claves de KMS obsoletas a las claves de datos](#)

Crear una clave de datos

Para crear una clave de datos, llame a la [GenerateDataKey](#) operación. AWS KMS genera la clave de datos. Luego, cifra una copia de la clave de datos en una [clave KMS de cifrado simétrica](#) que especifique. Esta operación devuelve una copia de texto no cifrado de la clave de datos y una copia de la clave de datos que está cifrada con la clave KMS. En la imagen siguiente, se muestra esta operación.



AWS KMS también admite la [GenerateDataKeyWithoutPlaintext](#) operación, que devuelve solo una clave de datos cifrada. Cuando necesite usar la clave de datos, solicite AWS KMS [descifrarla](#).

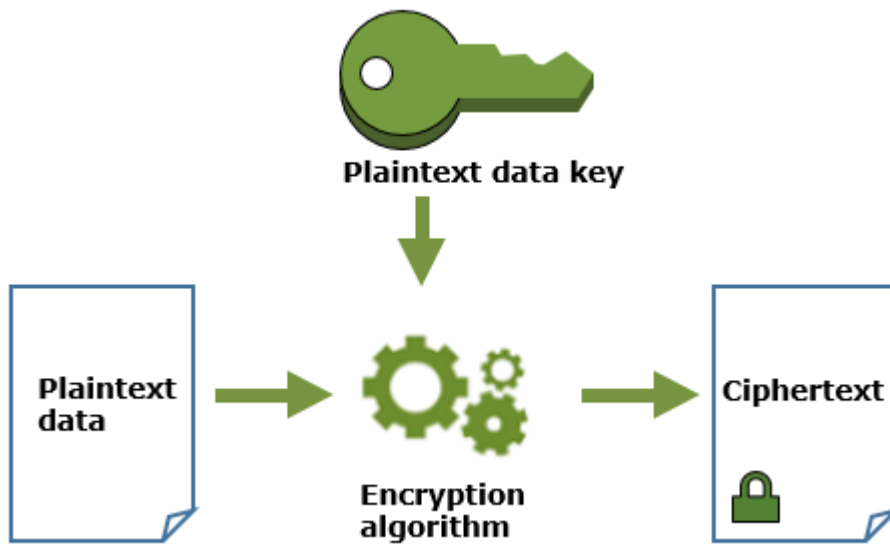
Cómo funcionan las operaciones criptográficas con claves de datos

En los temas siguientes se explica cómo funcionan las claves de datos generadas por una [GenerateDataKeyGenerateDataKeyWithoutPlaintext](#) operación.

Cifrar los datos con una clave de datos

AWS KMS no puede usar una clave de datos para cifrar datos. Sin embargo, puede utilizar la clave de datos fuera de AWS KMS, por ejemplo, mediante OpenSSL o una biblioteca criptográfica como la [AWS Encryption SDK](#)

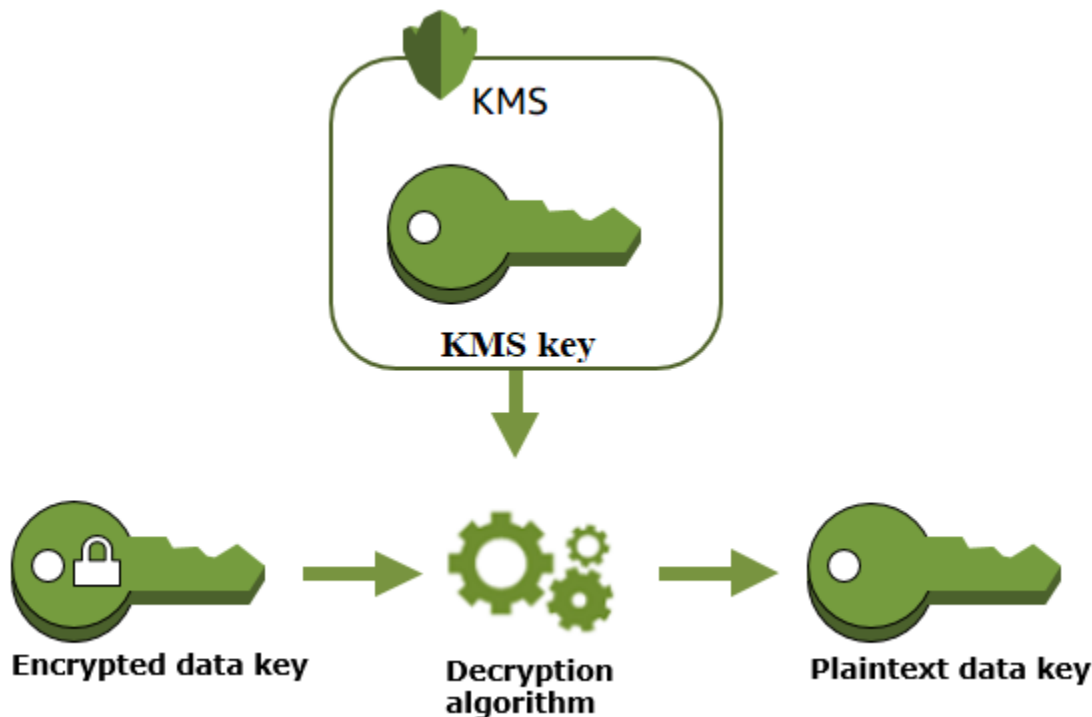
Después de utilizar la clave de datos en texto no cifrado para cifrar los datos, elimínela de la memoria tan pronto como sea posible. Puede almacenar de forma segura la clave de datos cifrada con los datos cifrados para que esté disponible para descifrar los datos.



Descifrar los datos con una clave de datos

Para descifrar los datos, pase la clave de datos cifrados a la operación de [descifrado](#). AWS KMS utiliza la clave KMS para descifrar la clave de datos y, a continuación, devuelve la clave de datos en texto plano. Utilice la clave de datos de texto no cifrado para descifrar los datos y, a continuación, elimine la clave de datos de texto no cifrado de la memoria tan pronto como sea posible.

En el siguiente diagrama, se muestra cómo se utiliza la operación `Decrypt` para descifrar una clave de datos cifrada.



Cómo afectan las claves de KMS obsoletas a las claves de datos

Cuando una clave de KMS queda obsoleta, el efecto es casi inmediato (sujeto a la posible coherencia). El [estado de clave](#) de la clave de KMS cambia para reflejar su nueva condición y todas las solicitudes para utilizar la clave de KMS en [operaciones criptográficas](#) fallan.

Sin embargo, el efecto en las claves de datos cifradas por la clave de KMS, y en los datos cifrados por la clave de datos, se retrasa hasta que se vuelva a utilizar la clave de KMS. Por ejemplo, para descifrar la clave de datos.

Las claves de KMS pueden quedar obsoletas por varios motivos. Entre ellos, se incluyen las siguientes acciones que puede realizar.

- [Desactivar la clave de KMS](#)
- [Programar la clave de KMS para eliminarla](#)
- [Eliminar el material de clave](#) de una clave de KMS con material de clave importado o permitir que el material de clave importado caduque. Si una clave de KMS con origen EXTERNAL tiene varios materiales clave asociados, la eliminación o caducidad de cualquier material clave hará que la clave quede inutilizable.
- [Desconectar el almacén de AWS CloudHSM claves](#) que aloja la clave de KMS o [eliminar la clave del AWS CloudHSM clúster](#) que sirve de material clave para la clave de KMS.

- [Desconectar el almacén de claves externo](#) que aloja la clave de KMS o realizar cualquier otra acción que interfiera con las solicitudes de cifrado y descifrado al proxy del almacén de claves externo, incluida la eliminación de la clave externa de su administrador de claves externo.

Este efecto es especialmente importante para las muchas personas Servicios de AWS que utilizan claves de datos para proteger los recursos que administra el servicio. El siguiente ejemplo utiliza Amazon Elastic Block Store (Amazon EBS) y Amazon Elastic Compute Cloud (EC2Amazon). Los diferentes Servicios de AWS utilizan las claves de datos de diferentes maneras. Para obtener más información, consulte la sección de Protección de datos del capítulo de Seguridad del Servicio de AWS.

Por ejemplo, considere esta situación:

1. [Usted crea un volumen de EBS cifrado](#) y especifica una clave de KMS para protegerlo. Amazon EBS solicita a AWS KMS que utilice su clave KMS para [generar una clave de datos cifrada](#) para el volumen. Amazon EBS almacena la clave de datos cifrada con los metadatos del volumen.
2. Cuando adjuntas el volumen de EBS a una EC2 instancia, Amazon EC2 utiliza tu clave de KMS para descifrar la clave de datos cifrados del volumen de EBS. Amazon EC2 utiliza la clave de datos del hardware Nitro, que se encarga de cifrar todos los discos I/O en el volumen de EBS. La clave de datos permanece en el hardware Nitro mientras el volumen de EBS esté conectado a la instancia. EC2
3. Usted realiza una acción que deja a la clave de KMS obsoleta. Esto no tiene ningún efecto inmediato en la EC2 instancia ni en el volumen de EBS. Amazon EC2 usa la clave de datos (no la clave KMS) para cifrar todo el disco I/O mientras el volumen está conectado a la instancia.
4. Sin embargo, cuando el volumen de EBS cifrado se separa de la EC2 instancia, Amazon EBS elimina la clave de datos del hardware de Nitro. La próxima vez que el volumen de EBS cifrado se adjunte a una EC2 instancia, se producirá un error en el adjunto, ya que Amazon EBS no podrá utilizar la clave de KMS para descifrar la clave de datos cifrados del volumen. Para volver a usar el volumen de EBS, debe hacer que la clave de KMS se pueda utilizar de nuevo.

Generación de pares de claves de datos

Una clave KMS asimétrica representa un par de claves de datos. Los pares de claves de datos son claves de datos asimétricos que constan de una clave privada y una clave pública relacionadas matemáticamente. Están diseñados para su uso en el cifrado y descifrado del lado del cliente, la firma y la verificación fuera de AWS KMS, o para establecer un secreto compartido entre dos pares.

A diferencia de los pares de claves de datos que generan herramientas como OpenSSL, AWS KMS protege la clave privada de cada par de claves de datos con una clave AWS KMS de cifrado simétrico que usted especifique. Sin embargo, AWS KMS no almacena, administra ni rastrea sus pares de claves de datos, ni realiza operaciones criptográficas con pares de claves de datos. Debe utilizar y administrar pares de claves de datos fuera de AWS KMS.

Temas

- [Crear un par de clave de datos](#)
- [Funcionamiento de las operaciones criptográficas con pares de claves de datos](#)

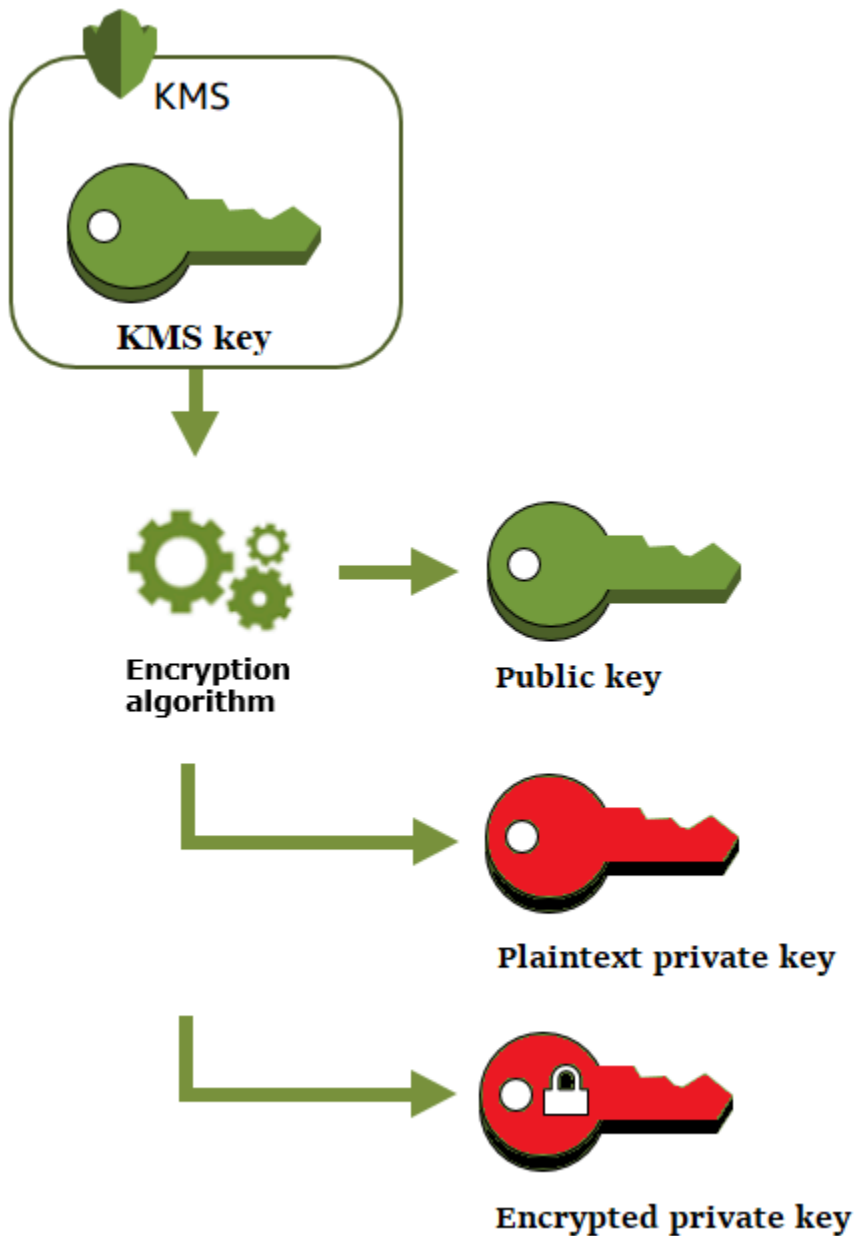
Crear un par de clave de datos

Para crear un par de claves de datos, llame a las [GenerateDataKeyPairWithoutPlaintext](#) operaciones [GenerateDataKeyPair](#). Especifique la [clave KMS de cifrado simétrica](#) que desea utilizar para cifrar la clave privada.

`GenerateDataKeyPair` devuelve una clave pública de texto no cifrado, una clave pública de texto no cifrado y una clave privada cifrada. Utilice esta operación cuando necesite una clave privada de texto no cifrado inmediatamente, por ejemplo, para generar una firma digital.

`GenerateDataKeyPairWithoutPlaintext` devuelve una clave pública de texto no cifrado y una clave privada de texto cifrado, pero no una clave privada de texto no cifrado. Utilice esta operación cuando no necesite una clave privada de texto no cifrado inmediatamente, por ejemplo, cuando esté cifrando con una clave pública. Más tarde, cuando necesite una clave privada de texto no cifrado para descifrar los datos, puede llamar a la operación [Decrypt](#).

En la imagen siguiente, se muestra la operación `GenerateDataKeyPair`. La operación `GenerateDataKeyPairWithoutPlaintext` omite la clave privada de texto no cifrado.



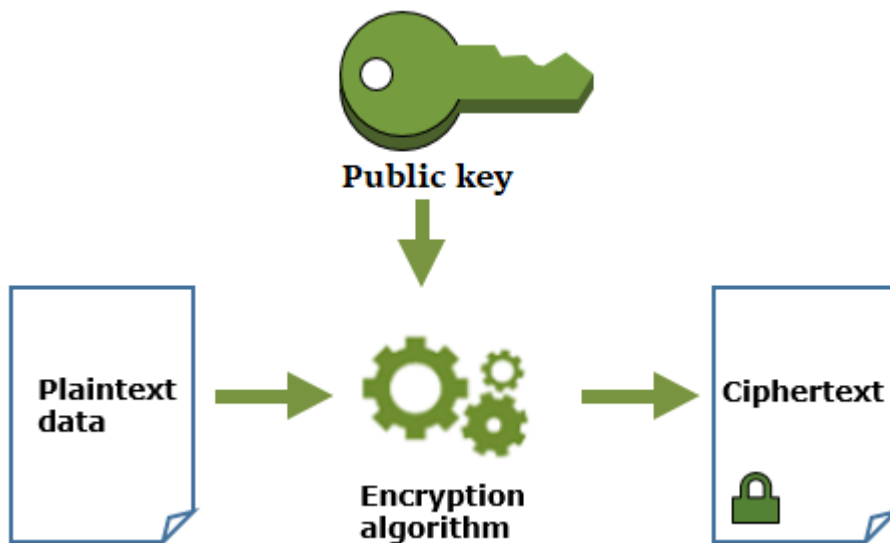
Funcionamiento de las operaciones criptográficas con pares de claves de datos

En los temas siguientes se explican qué operaciones criptográficas se pueden realizar con los pares de claves de datos generados por una [GenerateDataKeyPairWithoutPlaintext](#) operación [GenerateDataKeyPair](#) y cómo funcionan.

Cifrar los datos con un par de claves de datos

Cuando cifra con un par de claves de datos, utiliza la clave pública del par para cifrar los datos y la clave privada del mismo par para descifrar los datos. Normalmente, los pares de claves de datos se utilizan cuando muchas partes necesitan cifrar datos que solo la parte que posee la clave privada puede descifrar.

Las partes con la clave pública utilizan esa clave para cifrar datos, como se muestra en el siguiente diagrama.

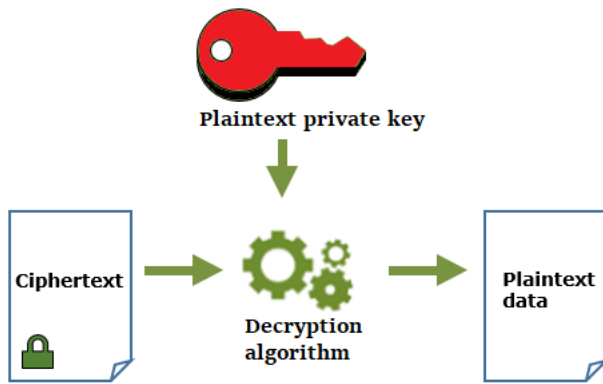


Descifrar los datos con un par de claves de datos

Para descifrar los datos, utilice la clave privada en el par de claves de datos. Para que la operación tenga éxito, las claves públicas y las privadas deben pertenecer al mismo par de claves de datos y debe utilizar el mismo algoritmo de cifrado.

Para descifrar la clave privada cifrada, pásala a la operación [Decrypt](#). Utilice la clave privada de texto no cifrado para descifrar los datos. A continuación, elimina la clave privada de texto no cifrado de la memoria lo antes posible.

El siguiente diagrama muestra cómo utilizar la clave privada en un par de claves de datos para descifrar el texto cifrado.



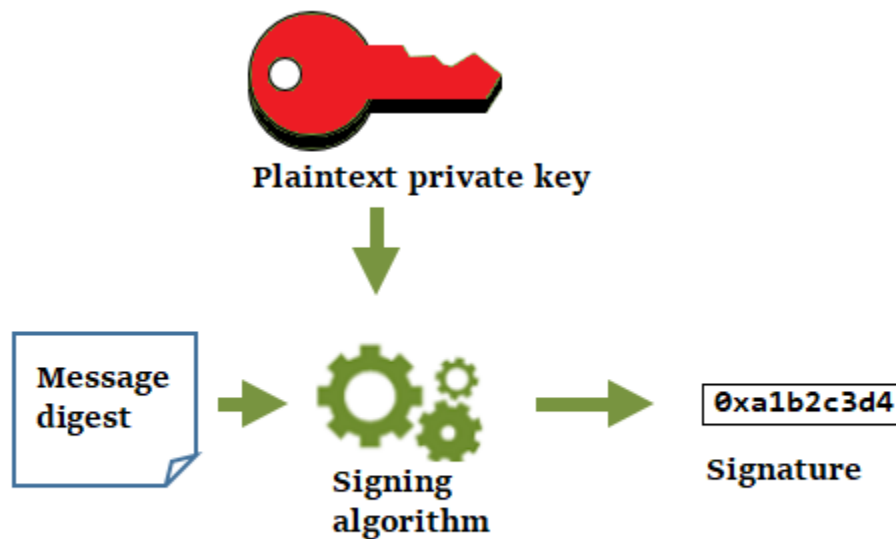
Firmar los mensajes con un par de claves de datos

Para generar una firma criptográfica para un mensaje, utilice la clave privada en el par de claves de datos. Cualquier persona con clave pública puede utilizarla para verificar que el mensaje se firmase con su clave privada y que no haya cambiado desde entonces.

Si cifra su clave privada, pase la clave privada cifrada a la operación de [descifrado](#). AWS KMS utiliza la clave KMS para descifrar la clave de datos y, a continuación, devuelve la clave privada en texto simple. Utilice la clave privada de texto no cifrado para generar la firma. A continuación, elimina la clave privada de texto no cifrado de la memoria lo antes posible.

Para firmar un mensaje, cree un resumen de mensaje con una función hash criptográfica, como el comando [dgst](#) en OpenSSL. Después, pase su clave privada no cifrada al algoritmo de firma. El resultado es una firma que representa los contenidos del mensaje. (Es posible que pueda firmar mensajes más cortos sin crear primero un resumen. El tamaño máximo del mensaje varía según la herramienta de firma que utilice).

El siguiente diagrama muestra cómo utilizar la clave privada en un par de claves de datos para firmar un mensaje.

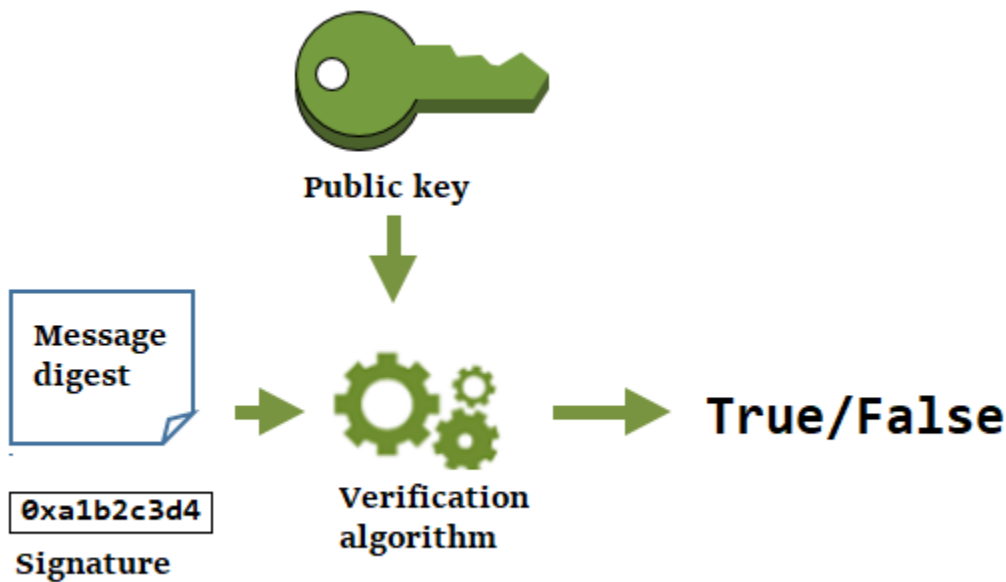


Verificar una firma con un par de claves de datos

Cualquier persona que tenga la clave pública en su par de claves de datos puede utilizarla para verificar la firma que generó con su clave privada. La verificación confirma que un usuario autorizado firmó el mensaje con el algoritmo de firma y la clave privada especificada y el mensaje no ha cambiado desde entonces.

Para tener éxito, la parte que verifique la firma debe generar el mismo tipo de resumen, utilizar el mismo algoritmo y utilizar la clave pública que se corresponde con la clave privada utilizada para firmar el mensaje.

El siguiente diagrama muestra cómo utilizar la clave pública en un par de claves de datos para verificar la firma de un mensaje.



Obtenga un secreto compartido con pares de claves de datos

El acuerdo de claves permite a dos homólogos, cada uno con un par de claves pública-privada de curva elíptica, establecer un secreto compartido a través de un canal no seguro. Para [obtener un secreto compartido](#), los dos homólogos deben intercambiar sus claves públicas a través de un canal de comunicación no seguro (como Internet). A continuación, cada parte utiliza su clave privada y la clave pública de su homólogo para calcular el mismo secreto compartido mediante un algoritmo de acuerdo de claves. Puede usar el valor secreto compartido para obtener una clave simétrica que pueda cifrar y descifrar los datos que se envían entre los dos pares, o que pueda generarlos y verificarlos. HMACs

Note

AWS KMS recomienda encarecidamente comprobar que la clave pública que recibe procede de la parte prevista antes de utilizarla para obtener un secreto compartido.

Realice operaciones sin conexión con claves públicas

En una clave KMS asimétrica, la clave privada se crea AWS KMS y nunca se queda AWS KMS sin cifrar. Para usar la clave privada, debe llamar. AWS KMS Puedes usar la clave pública que contiene AWS KMS llamando a las operaciones de la AWS KMS API. O bien, puedes [descargar la clave pública](#) y compartirla para usarla fuera de ella AWS KMS.

Puedes compartir una clave pública para que otros puedan cifrar datos ajenos a los AWS KMS que solo puedes descifrar con tu clave privada. O bien, para permitir que otros verifiquen una firma digital fuera del AWS KMS que haya generado con su clave privada. O bien, para compartir su clave pública con un compañero para obtener un secreto compartido.

Cuando utilizas la clave pública de tu clave KMS asimétrica interna AWS KMS, te beneficias de la autenticación, la autorización y el registro que forman parte de cada operación. AWS KMS También reduce el riesgo de cifrar datos que no se pueden descifrar. Estas funciones no son efectivas fuera de AWS KMS. Para obtener más información, consulte [Consideraciones especiales para descargar claves públicas](#).

Tip

¿Busca claves de datos o claves SSH? En este tema se explica cómo administrar claves asimétricas en AWS Key Management Service, donde la clave privada no es exportable. Para ver los pares de claves de datos exportables en los que la clave privada está protegida por una clave KMS de cifrado simétrico, consulte. [GenerateDataKeyPair](#) Para obtener ayuda sobre la descarga de la clave pública asociada a una instancia de Amazon EC2, consulte Recuperar la clave pública en la [Guía del usuario de Amazon EC2](#) y [Guía del usuario de Amazon EC2](#).

Temas

- [Consideraciones especiales para descargar claves públicas](#)
- [Descarga de la clave pública](#)
- [Ejemplo de operaciones sin conexión](#)

Consideraciones especiales para descargar claves públicas

Para proteger sus claves de KMS, AWS KMS proporciona controles de acceso, cifrado autenticado y registros detallados de cada operación. AWS KMS también le permite impedir el uso de claves KMS, de forma temporal o permanente. Por último, AWS KMS las operaciones están diseñadas para minimizar el riesgo de cifrar datos que no se pueden descifrar. Estas funciones no están disponibles cuando se utilizan claves públicas descargadas fuera de AWS KMS.

Autorización

[Las políticas clave y las políticas de IAM](#) que controlan el acceso a la clave KMS interna no AWS KMS tienen ningún efecto en las operaciones que se realizan fuera de AWS KMS. Cualquier usuario que pueda obtener la clave pública puede utilizarla fuera de ella, AWS KMS incluso si no tiene permiso para cifrar datos o verificar las firmas con la clave KMS.

Restricciones de uso de las claves

Las restricciones de uso de claves no entran en vigor fuera de AWS KMS. Si llama a la operación de [cifrado](#) con una clave KMS que tiene un KeyUsage de SIGN_VERIFY, se produce un error en la AWS KMS operación. Sin embargo, si cifra datos de forma externa AWS KMS con una clave pública de una clave KMS con un KeyUsage de SIGN_VERIFY o KEY_AGREEMENT, los datos no se pueden descifrar.

Restricciones del algoritmo

Las restricciones a los algoritmos de cifrado y firma que AWS KMS admite no son efectivas fuera de AWS KMS. Si cifra los datos con la clave pública de una clave de KMS externa y utiliza un algoritmo de AWS KMS cifrado que AWS KMS no lo admite, los datos no se pueden descifrar.

Desactivar y eliminar claves KMS

Las medidas que puede tomar para impedir el uso de la clave KMS en una operación criptográfica interna AWS KMS no impiden que nadie utilice la clave pública fuera de ella. AWS KMS Por ejemplo, desactivar una clave KMS, programar la eliminación de una clave KMS, eliminar una clave KMS o eliminar el material de claves de una clave KMS no tienen ningún efecto en una clave pública fuera de AWS KMS. Si eliminas una clave KMS asimétrica o eliminas o pierdes su material clave, los datos que no hayas cifrado con una clave pública no se podrán recuperar.

AWS KMS

Registro

AWS CloudTrail los registros que registran todas las AWS KMS operaciones, incluidas la solicitud, la respuesta, la fecha, la hora y el usuario autorizado, no registran el uso de la clave pública fuera de ella. AWS KMS

Verificación fuera de línea con pares de SM2 claves (solo en las regiones de China)

Para verificar una firma fuera o AWS KMS con una clave SM2 pública, debes especificar el identificador distintivo. De forma predeterminada, 1234567812345678 se AWS KMS utiliza como identificador distintivo. Para obtener más información, consulta [Verificación sin conexión con pares de SM2 claves \(solo regiones de China\)](#).

Descarga de la clave pública

Puede descargar la clave pública de un par de claves KMS asimétricas en la AWS KMS consola o mediante la [GetPublicKey](#) operación. Para descargar la clave pública, debe tener el permiso `kms:GetPublicKey` en la clave KMS asimétrica.

[La clave pública que se AWS KMS devuelve es una clave pública X.509 codificada en DER, también conocida como SubjectPublicKeyInfo \(SPKI\), según se define en el RFC 5280.](#) Cuando se utiliza la API HTTP o la, el valor está codificado en AWS CLI Base64. De lo contrario, no está codificado en Base64.

Para descargar la clave pública de un par de claves KMS asimétricas, necesita permisos `kms:GetPublicKey`. Para obtener más información sobre AWS KMS los permisos, consulta la [Referencia de permisos](#)

Uso de la AWS KMS consola

Puede utilizarla Consola de administración de AWS para ver, copiar y descargar la clave pública de una clave KMS asimétrica de su Cuenta de AWS. Para descargar la clave pública de una clave KMS asimétrica en otro lugar Cuenta de AWS, utilice la AWS KMS API.

1. Inicia sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrela en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente.

4. Elija el alias o el ID de clave de una clave KMS asimétrica.
5. Elija la pestaña Cryptographic configuration (Configuración criptográfica). Registre los valores de los campos Especificación de clave, Uso de claves y Algoritmos de cifrado o Algoritmos de firma. Deberás usar estos valores para usar la clave pública fuera de AWS KMS. Asegúrese de compartir esta información cuando comparta la clave pública.
6. Seleccione la pestaña Public key (Clave pública).
7. Para copiar la clave pública al portapapeles, elija Copy (Copiar). Para descargar la clave pública en un archivo, elija Download (Descargar).

Uso de la AWS KMS API

La [GetPublicKey](#) operación devuelve la clave pública en una clave KMS asimétrica. También devuelve información crítica que es necesaria para utilizar correctamente la clave pública AWS KMS, como el uso de claves y los algoritmos de cifrado. Asegúrese de guardar estos valores y compartirlos siempre que comparta la clave pública.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Para especificar una clave KMS, utilice el [ID de la clave](#), [ARN de la clave](#), [nombre de alias](#) o [ARN del alias](#). Cuando utilice un nombre del alias, utilice el prefijo alias/. Para especificar una clave de KMS en otra Cuenta de AWS, debe usar su clave ARN o su alias ARN.

Antes de ejecutar este comando, sustituya el nombre del alias de ejemplo por un identificador válido para la clave KMS. Para ejecutar este comando, debe tener permisos de `kms:GetPublicKey` en la clave 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..."
```

```
}
```

Ejemplo de operaciones sin conexión

Tras [descargar la clave pública](#) de su par de claves KMS asimétricas, puede compartirla con otras personas y utilizarla para realizar operaciones sin conexión.

AWS CloudTrail Los registros que registran todas las AWS KMS operaciones, incluidas la solicitud, la respuesta, la fecha, la hora y el usuario autorizado, no registran el uso de la clave pública fuera de ella. AWS KMS

En este tema se proporcionan ejemplos de operaciones sin conexión y se detallan las herramientas que AWS KMS proporciona para facilitar las operaciones sin conexión.

Temas

- [Obtención de secretos compartidos sin conexión](#)
- [Verificación fuera de línea con pares de claves ML-DSA](#)
- [Verificación fuera de línea con pares de SM2 claves \(solo en las regiones de China\)](#)

Obtención de secretos compartidos sin conexión

Puede [descargar la clave pública](#) del par de claves ECC para uso en operaciones sin conexión, es decir, operaciones fuera de AWS KMS.

En el siguiente tutorial de [OpenSSL](#) se muestra un método para obtener un secreto compartido sin utilizar la clave pública de un par AWS KMS de claves KMS ECC y una clave privada creada con OpenSSL.

1. Cree un key pair ECC en OpenSSL y prepárelo para usarlo con él. 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. Cree un par de claves de acuerdo de claves ECC AWS KMS y prepárelo para su uso con OpenSSL.

```
// Create a KMS key on the same curve as the key pair from step 1
// with a key usage of KEY_AGREEMENT
// Save its ARN in KMS_KEY1_ARN.
export KMS_KEY1_ARN=`aws kms create-key --key-spec ${KMS_CURVE_NAME} \
    --key-usage KEY_AGREEMENT | tee /dev/stderr | jq -r .KeyMetadata.Arn`

// Download the public key and save the Base64-encoded version in KMS_KEY1_PUB_BASE64

export KMS_KEY1_PUB_BASE64=`aws kms get-public-key --key-id ${KMS_KEY1_ARN} | \
    tee /dev/stderr | jq -r .PublicKey`

// Create a PEM file for the public KMS key for use with OpenSSL
export KMS_KEY1_PUB_PEM="aws_kms_ecdh_key1_pub.pem"
echo "-----BEGIN PUBLIC KEY-----" > ${KMS_KEY1_PUB_PEM}
echo ${KMS_KEY1_PUB_BASE64} | fold -w 64 >> ${KMS_KEY1_PUB_PEM}
echo "-----END PUBLIC KEY-----" >> ${KMS_KEY1_PUB_PEM}
```

3. Obtenga el secreto compartido en OpenSSL mediante la clave privada de OpenSSL y la clave pública de 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}
```

Verificación fuera de línea con pares de claves ML-DSA

AWS KMS admite una variante protegida de la firma ML-DSA, tal como se describe en la sección 3.4 de la [norma federal de procesamiento de información \(FIPS\) 204](#) para mensajes de hasta 4 KB de bytes.

Para firmar mensajes de más de 4 KB, realice el paso de preprocesamiento del mensaje fuera de AWS KMS. Este paso de uso de hash crea una μ representativa del mensaje de 64 bytes, tal como se define en la sección 6.2 de NIST FIPS 204.

AWS KMS tiene un tipo de mensaje denominado `EXTERNAL_MU` para mensajes de más de 4 KB. Si usa esto en lugar del tipo de `RAW` mensaje, AWS KMS:

- Asume que ya ha realizado el paso de uso de hash
- Omite su proceso de hash interno
- Funciona con mensajes de cualquier tamaño

Al verificar un mensaje, el método que se utilice dependerá de la restricción de tamaño del sistema o de la biblioteca externos y de si admite la μ representativa del mensaje de 64 bytes:

- Si el mensaje es de menor tamaño que la restricción de tamaño, utilice el tipo de mensaje `RAW`.
- Si el mensaje es mayor que la restricción de tamaño, utilice la μ representativa en el sistema externo.

En las siguientes secciones se muestra cómo firmar mensajes con OpenSSL AWS KMS y cómo verificarlos. Proporcionamos ejemplos de mensajes con un tamaño inferior o superior al límite de 4 KB impuesto por AWS KMS. OpenSSL no impone un límite al tamaño de los mensajes para su verificación.

Para ambos ejemplos, primero obtenga la clave pública de AWS KMS. Utilice el siguiente comando AWS CLI :

```
aws kms get-public-key \
  --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
  --output text \
  --query PublicKey | base64 --decode > public_key.der
```

Tamaño del mensaje inferior a 4 KB

Para mensajes de menos de 4 KB, usa el tipo de RAW mensaje con AWS KMS. Si bien puede usar EXTERNAL_MU, no es necesario para los mensajes que estén dentro del límite de tamaño.

Use el siguiente AWS CLI comando para firmar el mensaje:

```
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 el mensaje mediante OpenSSL utilice el siguiente comando:

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature
ExampleSignature.bin
```

Tamaño del mensaje superior a 4 KB

Para firmar mensajes de más de 4 KB, utilice el tipo de mensaje EXTERNAL_MU. Cuando utilice EXTERNAL_MU, codifique con hash previamente el mensaje de forma externa en una μ representativa de 64 bytes, tal como se define en la sección 6.2 de NIST FIPS 204, y lo transfiere a las operaciones de firma o verificación. Tenga en cuenta que es diferente del “MLDSA previo al hash” o del HASHML-DSA definido en la sección 5.4 de NIST FIPS 204.

1. En primer lugar, cree un prefijo de mensaje. El prefijo contiene un separador de dominios, la longitud de cualquier contexto y el contexto. El valor predeterminado para el separador de dominios y la longitud del contexto es cero.
2. Anexe el prefijo del mensaje al mensaje.
3. Se usa SHAKE256 para codificar la clave pública y anteponerla al resultado del paso 2.
4. Por último, haga un hash del resultado del paso 3 para obtener un EXTERNAL_MU 64 bytes.

El siguiente ejemplo utiliza OpenSSL 3.5 para crear 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
```

Tras crear el `mu.bin` archivo, llama a la AWS KMS API con el siguiente comando para firmar el mensaje:

```
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
```

La firma resultante es la misma que la firma RAW del mensaje original. Puede usar el mismo comando de OpenSSL 3.5 para verificar el mensaje:

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature
ExampleSignature.bin
```

Verificación fuera de línea con pares de SM2 claves (solo en las regiones de China)

Para verificar una firma fuera o AWS KMS con una clave SM2 pública, debes especificar el identificador distintivo. Al pasar un mensaje sin procesar [MessageType:RAW](#), a la API de [firma](#), AWS KMS utiliza el identificador distintivo predeterminado `1234567812345678`, definido por la OSCCA en 0009-2012. GM/T No puede especificar su propio identificador distintivo en AWS KMS.

Sin embargo, si vas a generar un resumen del mensaje fuera de AWS, puedes especificar tu propio identificador distintivo y, a continuación, pasarle el resumen del mensaje a, para que lo firme. [MessageType:DIGEST](#) AWS KMS Para ello, cambie el valor `DEFAULT_DISTINGUISHING_ID` en la clase `SM2OfflineOperationHelper`. El identificador distintivo que especifique puede ser cualquier cadena de hasta 8192 caracteres. Después de AWS KMS firmar el resumen del mensaje, necesitará el resumen del mensaje o el mensaje y el identificador distintivo utilizados para calcular el resumen para verificarlo sin conexión.

⚠ Important

El código de referencia `SM2OfflineOperationHelper` está diseñado para ser compatible con [Bouncy Castle](#) versión 1.68. Para obtener ayuda con otras versiones, póngase en contacto con bouncycastle.org.

Clase `SM2OfflineOperationHelper`

Para ayudarle a realizar operaciones con SM2 claves fuera de línea, la `SM2OfflineOperationHelper` clase de Java tiene métodos que realizan las tareas por usted. Puede usar esta clase auxiliar como modelo para otros proveedores de cifrado.

En su interior AWS KMS, las conversiones de texto cifrado sin procesar y los cálculos del resumen de los mensajes de la SM2 DSA se realizan automáticamente. No todos los proveedores de criptografía lo implementan de la misma SM2 manera. Algunas bibliotecas, como las versiones 1.1.1 y posteriores de [OpenSSL](#), realizan estas acciones automáticamente. AWS KMS confirmó este comportamiento en las pruebas con OpenSSL versión 3.0. Utilice la siguiente clase de `SM2OfflineOperationHelper` con bibliotecas, como [Bouncy Castle](#), que requieren que realice estas conversiones y cálculos manualmente.

La clase `SM2OfflineOperationHelper` proporciona métodos para las siguientes operaciones fuera de línea:

- Cálculo del resumen del mensaje

Para generar un resumen de mensajes sin conexión que puedas usar para la verificación sin conexión o al que puedas pasar AWS KMS para firmar, usa este método. `calculateSM2Digest` El `calculateSM2Digest` método genera un resumen del mensaje con el algoritmo SM3 de hash. La `GetPublicKey` API devuelve la clave pública en formato binario. Debe analizar la clave binaria para convertirla en una versión Java `PublicKey`. Proporcione la clave pública analizada con el mensaje. El método combina automáticamente el mensaje con el identificador distintivo predeterminado, 1234567812345678, pero puede establecer su propio identificador distintivo cambiando el valor `DEFAULT_DISTINGUISHING_ID`.

- Verificar

Para verificar una firma sin conexión, utilice el método `offlineSM2DSAVerify`. El método `offlineSM2DSAVerify` utiliza el resumen del mensaje calculado a partir del identificador

distintivo especificado y el mensaje original que proporciona para verificar la firma digital. La [GetPublicKey](#) API devuelve la clave pública en formato binario. Debe analizar la clave binaria para convertirla en una versión Java PublicKey. Proporcione la clave pública analizada con el mensaje original y la firma que desea verificar. Para obtener más información, consulte [Verificación sin conexión con pares de SM2 claves](#).

- Encrypt

Para cifrar texto sin formato sin conexión, utilice el método `offlineSM2PKEEncrypt`. Este método garantiza que el texto cifrado esté en un formato que se AWS KMS pueda descifrar. El `offlineSM2PKEEncrypt` método cifra el texto sin formato y, a continuación, convierte el texto cifrado sin procesar producido por PKE al formato ASN.1. SM2 La [GetPublicKey](#) API devuelve la clave pública en formato binario. Debe analizar la clave binaria para convertirla en una versión Java PublicKey. Proporcione la clave pública analizada con el texto sin formato que desea cifrar.

Si no está seguro de si necesita realizar la conversión, utilice la siguiente operación de OpenSSL para probar el formato del texto cifrado. Si la operación falla, debe convertir el texto cifrado al formato ASN.1.

```
openssl asn1parse -inform DER -in ciphertext.der
```

De forma predeterminada, la `SM2OfflineOperationHelper` clase utiliza el identificador distintivo predeterminado al generar resúmenes de mensajes para las operaciones de SM2 DSA. 1234567812345678

```
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");
    }
}

```

Supervisar AWS KMS keys

El monitoreo es una parte importante para comprender la disponibilidad, el estado y el uso de su AWS KMS keys información AWS KMS y para mantener la confiabilidad, la disponibilidad y el rendimiento de sus AWS soluciones. La recopilación de los datos de monitoreo de todas las partes de su solución de AWS le ayudará a depurar un error que se produce en distintas partes del código, en caso de que ocurra. No obstante, antes de comenzar a monitorizar las claves KMS, debe crear un plan de monitorización que incluya respuestas a las siguientes preguntas:

- ¿Cuáles son los objetivos de la supervisión?
- ¿Qué recursos va a supervisar?
- ¿Con qué frecuencia va a supervisar estos recursos?
- ¿Qué [herramientas de monitorización](#) va a utilizar?
- ¿Quién se encargará de realizar las tareas de supervisión?
- ¿Quién debería recibir una notificación cuando suceda algo?

El siguiente paso es monitorear las claves KMS a lo largo del tiempo para establecer un punto de referencia para el uso normal de AWS KMS y las expectativas en su entorno. A medida que monitorice las claves KMS, almacene los datos de monitorización históricos para que pueda compararlos con los datos actuales, identificar los patrones normales y las anomalías, así como desarrollar métodos para la resolución de problemas.

Por ejemplo, puede supervisar la actividad y los eventos de la AWS KMS API que afectan a sus claves de KMS. Cuando los datos están por encima o por debajo de las normas establecidas, es posible que efectuar una investigación o adoptar medidas correctivas.

Para establecer un punto de referencia para los patrones normales, monitorice los elementos siguientes:

- AWS KMS Actividad de la API para las operaciones del plano de datos. Se trata de [operaciones criptográficas](#) que utilizan una clave KMS, como [Decrypt](#), [Encrypt](#) y [ReEncrypt](#). [GenerateDataKey](#)
- AWS KMS Actividad de la API para las operaciones del plano de control que son importantes para usted. Estas operaciones administran una clave de KMS, y es posible que desee supervisar las que modifican la disponibilidad de una clave de KMS (por ejemplo [ScheduleKeyDeletion](#), [CancelKeyDeletion](#), [DisableKey](#), [EnableKeyImportKeyMaterial](#), y [DeletelImportedKeyMaterial](#)) o cambian el control de acceso de una clave de KMS (por ejemplo, [PutKeyPolicy](#) y [RevokeGrant](#)).

- Otras AWS KMS métricas (como el tiempo que queda hasta que caduque el [material clave importado](#)) y eventos (como la caducidad del material clave importado o la eliminación o rotación de claves de una clave KMS).

Herramientas de supervisión

AWS proporciona varias herramientas que puede utilizar para supervisar sus claves de KMS. Puede configurar algunas de estas herramientas para que monitoricen por usted, pero otras herramientas requieren intervención manual. Le recomendamos que automatice las tareas de monitorización en la medida de lo posible.

Herramientas de monitoreo automatizadas

Puede utilizar las siguientes herramientas de monitorización automatizada para monitorizar sus claves KMS e informar cuando haya algún cambio.

- AWS CloudTrail Supervisión de registros: comparta archivos de registro entre cuentas, supervise los archivos de CloudTrail registro en tiempo real enviándolos a CloudWatch Logs, cree aplicaciones de procesamiento de registros con la [biblioteca de CloudTrail procesamiento](#) y valide que los archivos de registro no hayan cambiado después de su entrega CloudTrail. Para obtener más información, consulte [Trabajar con archivos de CloudTrail registro](#) en la Guía del AWS CloudTrail usuario.
- Amazon CloudWatch Alarms: observe una sola métrica durante un período de tiempo que especifique y realice una o más acciones en función del valor de la métrica en relación con un umbral determinado durante varios períodos de tiempo. La acción es una notificación enviada a un tema del Servicio de Notificación Simple (Amazon SNS) o a una política de Amazon EC2 Auto Scaling. CloudWatch las alarmas no invocan acciones simplemente porque se encuentran en un estado determinado; el estado debe haber cambiado y se ha mantenido durante un número específico de períodos. Para obtener más información, consulte [Supervise las claves de KMS con Amazon CloudWatch](#).
- Amazon EventBridge: haga coincidir los eventos y diríjalos a una o más funciones o transmisiones de destino para capturar información de estado y, si es necesario, realizar cambios o tomar medidas correctivas. Para obtener más información, consulta [Supervise las claves de KMS con Amazon EventBridge](#) la [Guía del EventBridge usuario de Amazon](#).

- Amazon CloudWatch Logs: supervise, almacene y acceda a sus archivos de registro desde AWS CloudTrail u otras fuentes. Para obtener más información, consulta la [Guía del usuario CloudWatch de Amazon Logs](#).

Herramientas de supervisión manuales

Otra parte importante de la supervisión de las claves de KMS implica la supervisión manual de los elementos que CloudWatch las alarmas y los eventos no cubren. Los AWS paneles AWS KMS CloudWatch, AWS Trusted Advisor, y otros proporcionan una at-a-glance vista del estado de su AWS entorno.

Puede [personalizar](#) las Claves administradas por AWS y las páginas Claves administradas por el cliente de la [consola de AWS KMS](#) para mostrar la siguiente información sobre cada clave de KMS:

- ID de clave
- Status
- Fecha de creación
- Fecha de vencimiento (para las claves KMS con [material de claves importado](#))
- Origen
- ID de almacén de claves personalizado (para las claves KMS en [almacenes de claves personalizados](#))

El [panel de la consola de CloudWatch](#) muestra lo siguiente:

- Alarmas y estado actual
- Gráficos de alarmas y recursos
- Estado de los servicios

Además, puede utilizarlos CloudWatch para hacer lo siguiente:

- Crear [paneles personalizados](#) para monitorizar los servicios que le interesan
- Realizar un gráfico con los datos de las métricas para resolver problemas y descubrir tendencias
- Busque y explore todas sus métricas AWS de recursos
- Crear y editar las alarmas de notificación de problemas

AWS Trusted Advisor puede ayudarlo a monitorear sus AWS recursos para mejorar el rendimiento, la confiabilidad, la seguridad y la rentabilidad. Hay cuatro Trusted Advisor comprobaciones disponibles para todos los usuarios; hay más de 50 comprobaciones disponibles para los usuarios con un plan de soporte empresarial o empresarial. Para obtener más información, consulte [AWS Trusted Advisor](#).

Registro AWS KMS Llamadas a la API con AWS CloudTrail

AWS KMS está integrado con [AWS CloudTrail](#) un servicio que registra todas las llamadas realizadas a la API AWS KMS por los usuarios, los roles y otros AWS servicios. CloudTrail captura todas las llamadas a la API AWS KMS como eventos, incluidas las llamadas desde la AWS KMS consola, AWS KMS las API, las CloudFormation plantillas, el AWS Command Line Interface (AWS CLI) y Herramientas de AWS para PowerShell.

CloudTrail [registra todas AWS KMS las operaciones, incluidas las operaciones de solo lectura, como ListAliases y GetKeyRotationStatus, las operaciones que administran las claves de KMS, como CreateKey PutKeyPolicy, las operaciones criptográficas, como GenerateDataKeyDecrypt](#). También registra las operaciones internas que lo AWS KMS requieran, como,, [DeleteExpiredKeyMaterial](#) y [DeleteKey](#). [SynchronizeMultiRegionKeyRotateKey](#)

CloudTrail registra todas las operaciones realizadas correctamente y, en algunos casos, los intentos de llamadas fallidos, como cuando se deniega el acceso a un recurso a la persona que llama. [Cross-account las operaciones en las claves de KMS](#) se registran tanto en la cuenta de la persona que llama como en la cuenta del propietario de la clave de KMS. Sin embargo, AWS KMS las solicitudes multicuenta que se rechazan porque se deniega el acceso se registran únicamente en la cuenta de la persona que llama.

Por motivos de seguridad, algunos campos se omiten de las entradas de AWS KMS registro, como el Plaintext parámetro de una solicitud de [cifrado](#) y la respuesta a [GetKeyPolicy](#) cualquier operación criptográfica. Para facilitar la búsqueda de entradas de CloudTrail registro para determinadas claves de KMS, AWS KMS agrega el [ARN de clave](#) de la clave de KMS afectada al responseElements campo de las entradas de registro para algunas operaciones de administración de AWS KMS claves, incluso cuando la operación de API no devuelva el ARN de clave.

Aunque, de forma predeterminada, todas AWS KMS las acciones se registran como CloudTrail eventos, puedes AWS KMS excluirlas de un CloudTrail registro. Para obtener más información, consulte [Excluyendo AWS KMS eventos de un sendero](#).

Más información:

- Para ver ejemplos de AWS KMS operaciones de CloudTrail registro para plataformas certificadas, consulte [Monitoreo de solicitudes certificadas](#).

Temas

- [Resultado AWS KMS entradas de registro en CloudTrail](#)
- [Excluyendo AWS KMS eventos de un sendero](#)
- [Ejemplos de AWS KMS entradas de registro](#)

Resultado AWS KMS entradas de registro en CloudTrail

Para buscar entradas de CloudTrail registro, utilice la [CloudTrail consola](#) o la [CloudTrail LookupEvents](#) operación. CloudTrail admite numerosos [valores de atributos](#) para filtrar la búsqueda, incluidos el nombre del evento, el nombre de usuario y la fuente del evento.

Para ayudarle a buscar entradas de AWS KMS registro CloudTrail, AWS KMS rellena los siguientes campos de entrada de CloudTrail registro.

Note

A partir de diciembre de 2022, AWS KMS rellena los atributos Tipo de recurso y Nombre del recurso en todas las operaciones de administración que cambien una clave de KMS concreta. Estos valores de atributo pueden ser nulos en CloudTrail las entradas anteriores para las siguientes operaciones: [CreateAliasCreateGrantDeleteAlias](#), [DeleteImportedKeyMaterial](#), [ImportKeyMaterial](#), [ReplicateKey](#), [RetireGrant](#), [RevokeGrantUpdateAlias](#), y [UpdatePrimaryRegion](#).

Atributo	Valor	Entradas de registro
Fuente del evento (EventSource)	kms . amazonaws . com	Todas las operaciones.
Tipo de recurso (ResourceT ype)	AWS : : KMS : : Key	Operaciones de administr ación que cambian una clave de KMS en particular, como

Atributo	Valor	Entradas de registro
		CreateKey y EnableKey , pero no ListKeys.
Nombre del recurso (ResourceName)	ARN de clave (o ID de clave y ARN de clave)	Operaciones de administración que cambian una clave de KMS en particular, como CreateKey y EnableKey , pero no ListKeys.

Para ayudarle a encontrar entradas de registro para las operaciones de administración en determinadas claves de KMS, AWS KMS registra el ARN de clave de la clave de KMS afectada en el `responseElements.keyId` elemento de la entrada de registro, incluso cuando la operación de AWS KMS API no devuelva el ARN de clave.

Por ejemplo, una llamada correcta a la [DisableKey](#) operación no devuelve ningún valor en la respuesta, pero en lugar de un valor nulo, el `responseElements.keyId` valor de la [entrada de DisableKey registro](#) incluye la clave ARN de la clave KMS deshabilitada.

Esta función se agregó en diciembre de 2022 y afecta a las siguientes entradas de CloudTrail registro: [CreateAliasCreateGrantDeleteAlias](#), [DeleteKey](#), [DisableKey](#), [EnableKey](#), [EnableKeyRotation](#), [ImportKeyMaterial](#), [RotateKey](#), [SynchronizeMultiRegionKey](#), [TagResource](#), [UntagResourceUpdateAlias](#), y [UpdatePrimaryRegion](#).

Excluyendo AWS KMS eventos de un sendero

Para proporcionar un registro del uso y la administración de sus AWS KMS recursos, la mayoría de AWS KMS los usuarios se basan en los eventos de una CloudTrail ruta. El registro puede ser una fuente de datos valiosa para auditar eventos críticos, como la creación, inhabilitación y eliminación AWS KMS keys, el cambio de la política de claves y el uso de las claves de KMS por parte de AWS los servicios que actúan en su nombre. En algunos casos, los metadatos de una entrada de CloudTrail registro, como el [contexto de cifrado](#) de una operación de cifrado, pueden ayudarle a evitar o resolver errores.

Sin embargo, dado que AWS KMS puede generar una gran cantidad de eventos, AWS CloudTrail le permite excluir AWS KMS eventos de un registro. Esta configuración por ruta excluye todos los AWS KMS eventos; no puedes excluir eventos específicos AWS KMS .

⚠ Warning

Si se excluyen AWS KMS los eventos de un CloudTrail registro, se pueden ocultar las acciones que utilizan las claves de KMS. Actúe con precaución al conceder a las entidades principales el permiso `cloudtrail:PutEventSelectors` necesario para realizar esta operación.

Para excluir AWS KMS eventos de un registro:

- En la CloudTrail consola, utilice la configuración de eventos del Servicio de administración de claves de registro cuando [cree](#) o [actualice una ruta](#). Para obtener instrucciones, consulte [Registrar los eventos de administración Consola de administración de AWS en la](#) Guía del AWS CloudTrail usuario.
- En la CloudTrail API, utilice la [PutEventSelectors](#) operación. Agregue el atributo `ExcludeManagementEventSources` a sus selectores de eventos con un valor de `kms.amazonaws.com`. Para ver un ejemplo, consulte [Ejemplo: una ruta que no registra AWS Key Management Service eventos](#) en la Guía del AWS CloudTrail usuario.

Puede desactivar esta exclusión en cualquier momento cambiando la configuración de la consola o los selectores de eventos de un registro de seguimiento. A continuación, el sendero empezará a registrar AWS KMS los eventos. Sin embargo, no puede recuperar AWS KMS los eventos que ocurrieron mientras la exclusión estaba vigente.

Cuando excluyes AWS KMS eventos mediante la consola o la API, la operación de CloudTrail `PutEventSelectors` API resultante también se registra en tus CloudTrail registros. Si AWS KMS los eventos no aparecen en tus CloudTrail registros, busca un `PutEventSelectors` evento con el `ExcludeManagementEventSources` atributo establecido en `kms.amazonaws.com`.

Ejemplos de AWS KMS entradas de registro

AWS KMS escribe entradas en su CloudTrail registro cuando llama a una AWS KMS operación y cuando un AWS servicio llama a una operación en su nombre. AWS KMS también escribe una entrada cuando llama a una operación en tu nombre. Por ejemplo, escribe una entrada cuando [elimina una clave KMS](#) programado para su eliminación.

En los temas siguientes se muestran ejemplos de entradas de CloudTrail registro para AWS KMS las operaciones.

Para ver ejemplos de entradas de CloudTrail registro de solicitudes AWS KMS procedentes de plataformas certificadas, consulte [Monitoreo de solicitudes certificadas](#).

Temas

- [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](#)

- [GetKeyLastUsage](#)
- [GetKeyPolicy](#)
- [GetKeyRotationStatus](#)
- [GetParametersForImport](#)
- [ImportKeyMaterial](#)
- [ListAliases](#)
- [ListGrants](#)
- [ListKeyRotations](#)
- [PutKeyPolicy](#)
- [ReEncrypt](#)
- [ReplicateKey](#)
- [RetireGrant](#)
- [RevokeGrant](#)
- [RotateKey](#)
- [RotateKeyOnDemand](#)
- [ScheduleKeyDeletion](#)
- [Sign](#)
- [SynchronizeMultiRegionKey](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateAlias](#)
- [UpdateCustomKeyStore](#)
- [UpdateKeyDescription](#)
- [UpdatePrimaryRegion](#)
- [VerifyMac](#)
- [Verificar](#)
- [EC2 Ejemplo uno de Amazon](#)
- [EC2 Ejemplo dos de Amazon](#)

CancelKeyDeletion

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [CancelKeyDeletion](#) operación. Para obtener información sobre la eliminación AWS KMS keys, consulte [Eliminar un 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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [ConnectCustomKeyStore](#) operación. Para obtener información acerca conectar un almacén de claves personalizadas, consulte [Desconectar un almacén de AWS CloudHSM claves](#).

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

CreateAlias

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [CreateAlias](#) operación. El elemento `resources` incluye campos para los recursos de alias y clave KMS. Para obtener información sobre la creación de alias en AWS KMS, consulte [Crear alias](#).

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-08-14T23:08:31Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateAlias",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "aliasName": "alias/ExampleAlias",
    "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "caec1e0c-ce03-419e-bdab-6ab1f7c57c01",
  "eventID": "2dd6e784-8286-46a6-befd-d64e5a02fb28",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ]
}
```

```

    },
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

CreateCustomKeyStore

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [CreateCustomKeyStore](#) operación en un almacén de AWS CloudHSM claves. Para obtener información acerca de crear los almacenes de claves personalizadas, consulte [Crear un almacén de AWS CloudHSM claves](#).

```

{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [CreateGrant](#) operación. Para obtener información sobre la creación de subvenciones en AWS KMS, consulte [Subvenciones en AWS KMS](#).

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

En el siguiente ejemplo, se muestra una entrada de `CreateGrant` registro para una concesión con una restricción de contexto de cifrado.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:53:12Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "constraints": {
      "encryptionContextSubset": {
        "ContextKey1": "Value1"
      }
    }
  }
}
```

```

    }
  },
  "operations": [
    "Encrypt",
    "RetireGrant"
  ],
  "granteePrincipal": "service-name.amazonaws.com"
},
"responseElements": {
  "grantId": "f020fe75197b93991dc8491d6f19dd3cebb24ee62277a05914386724f3d48758",
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
},
"requestID": "f3c08808-63bc-11e4-bc2b-4198b6150d5c",
"eventID": "5d529779-2d27-42b5-92da-91aaea1fc4b5",
"readOnly": false,
"resources": [{
  "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "accountId": "111122223333"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

El siguiente ejemplo muestra una entrada de CreateGrant registro para una concesión principal de servicio. Esta concesión utiliza el `GranteeServicePrincipal` parámetro para especificar un AWS responsable de servicio como beneficiario e incluye una restricción de `SourceArn` concesión.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2026-03-04T18:22:45Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",

```

```

"sourceIPAddress": "192.0.2.0",
"userAgent": "AWS Internal",
"requestParameters": {
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "constraints": {
    "sourceArn": "arn:aws:dynamodb:us-east-1:111122223333:table/ExampleTable"
  },
  "operations": [
    "Encrypt",
    "Decrypt",
    "GenerateDataKey"
  ],
  "granteeServicePrincipal": "service-name.amazonaws.com",
  "retiringServicePrincipal": "service-name.amazonaws.com"
},
"responseElements": {
  "grantId": "a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2",
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
},
"requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
"readOnly": false,
"resources": [{
  "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "accountId": "111122223333"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

Note

Cuando se crea una concesión con el `GranteeServicePrincipal` parámetro, la entrada de CloudTrail registro de la `CreateGrant` operación incluye un `granteeServicePrincipal` campo en lugar de `granteePrincipal`. Del mismo modo, si `RetiringServicePrincipal` se especifica a, la entrada de registro incluye un `retiringServicePrincipal` campo en lugar de `retiringPrincipal`. Esto distingue las subvenciones que se crearon explícitamente `GranteeServicePrincipal` para un

[director de AWS servicio](#) de las subvenciones en las que un AWS servicio está representado `granteePrincipal` sobre el terreno.

CreateKey

Estos ejemplos muestran las entradas de AWS CloudTrail registro de la [CreateKey](#) operación.

Una entrada de CreateKey registro puede ser el resultado de una CreateKey solicitud o de la CreateKey operación de una [ReplicateKey](#) solicitud.

El siguiente ejemplo muestra una entrada de CloudTrail registro para una [CreateKey](#) operación que crea una [clave KMS de cifrado simétrico](#). Para obtener información sobre cómo crear claves KMS, consulte [Crear una clave de 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"
}

```

El siguiente ejemplo muestra el CloudTrail registro de una CreateKey operación que crea una clave KMS de cifrado simétrico en un almacén de [AWS CloudHSM claves](#).

```

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",

```



```

        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-14T17:39:50Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyUsage": "ENCRYPT_DECRYPT",
        "bypassPolicyLockoutSafetyCheck": false,
        "origin": "AWS_CLOUDHSM",
        "keySpec": "SYMMETRIC_DEFAULT",
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "customKeyStoreId": "cks-1234567890abcdef0",
        "description": ""
    },
    "responseElements": {
        "keyMetadata": {
            "awsAccountId": "111122223333",
            "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
            "creationDate": "Oct 14, 2021, 5:39:50 PM",
            "enabled": true,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Enabled",
            "origin": "AWS_CLOUDHSM",
            "customKeyStoreId": "cks-1234567890abcdef0",
            "cloudHsmClusterId": "cluster-1a23b4cdefg",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false
        }
    },
    "additionalEventData": {

```

```

    "backingKey": "{\\"backingKeyId\\":\\"backing-key-id\\"}"
  },
  "requestID": "4f0b185c-588c-4767-9e90-c618f7e13cad",
  "eventID": "c73964b8-703d-49e4-bd9e-f773d0ee1e65",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

El siguiente ejemplo muestra el CloudTrail registro de una CreateKey operación que crea una clave KMS de cifrado simétrico en un almacén de [claves 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,
    "customKeyStoreId": "cks-1234567890abcdef0",
    "xksKeyId": "bb8562717f809024"
  },
  "responseElements": {
    "keyMetadata": {
      "aWSAccountId": "111122223333",
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "creationDate": "Dec 7, 2022, 10:37:45 PM",
      "enabled": true,
      "description": "",
      "keyUsage": "ENCRYPT_DECRYPT",
      "keyState": "Enabled",
      "origin": "EXTERNAL_KEY_STORE",
      "customKeyStoreId": "cks-1234567890abcdef0",
      "keyManager": "CUSTOMER",
      "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "keySpec": "SYMMETRIC_DEFAULT",
      "encryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
      ],
      "multiRegion": false,
      "xksKeyConfiguration": {
        "id": "bb8562717f809024"
      }
    }
  },
  "requestID": "ba197c82-3ac7-487a-8ff4-7736bbbeb1316",
  "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

Estos ejemplos muestran las entradas de AWS CloudTrail registro de la operación de [descifrado](#).

La entrada de CloudTrail registro de una Decrypt operación siempre incluye el `valorencryptionAlgorithm`, `requestParameters` incluso si el algoritmo de cifrado no se especificó en la solicitud. El texto cifrado de la solicitud y el texto sin formato de la respuesta se omiten.

Temas

- [Descifrar con una clave de cifrado simétrica estándar](#)
- [Error de descifrado con una clave de cifrado simétrica estándar](#)
- [Descifra con una clave KMS en un AWS CloudHSM almacén de claves](#)
- [Descifrar con una clave de KMS en un almacén de claves externo](#)
- [Error de descifrado con una clave de KMS en un almacén de claves externo](#)
- [Descifra con una clave de cifrado simétrica estándar a través de una conexión TLS poscuántica](#)

Descifrar con una clave de cifrado simétrica estándar

A continuación se muestra un ejemplo de entrada de CloudTrail registro para una Decrypt operación con una clave de cifrado simétrica estándar.

```
{  
  "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"
    }
  }
}

```

Error de descifrado con una clave de cifrado simétrica estándar

El siguiente ejemplo de entrada de CloudTrail registro registra una Decrypt operación fallida con una clave KMS de cifrado simétrico estándar. La excepción (errorCode) y el mensaje de error (errorMessage) incluidos lo ayudan a resolver el error.

En este caso, la clave de KMS de cifrado simétrica especificada en la solicitud Decrypt no era la clave de KMS de cifrado simétrica que se utilizó para cifrar los datos.

```
{
  "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"
}

```

Descifra con una clave KMS en un AWS CloudHSM almacén de claves

El siguiente ejemplo de entrada de CloudTrail registro registra una Decrypt operación con una clave KMS en un [almacén de AWS CloudHSM claves](#). Todas las entradas de registro para operaciones criptográficas con clave KMS en un almacén de claves personalizado incluyen un campo `additionalEventData` con el `customKeyStoreId` y `backingKeyId`. El valor devuelto en el campo `backingKeyId` es el atributo `id` de clave de CloudHSM. Los `additionalEventData` no están especificados en la solicitud.

```

{
  "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": {
      "customKeyId": "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"
  }
}

```

Descifrar con una clave de KMS en un almacén de claves externo

El siguiente ejemplo de entrada de CloudTrail registro registra una Decrypt operación con una clave KMS en un [almacén de claves externo](#). Además de customKeyId, el campo additionalEventData incluye el [ID de clave externa](#) (XksKeyId). Los additionalEventData no están especificados en la solicitud.

```

{
  "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"
  }

```

Error de descifrado con una clave de KMS en un almacén de claves externo

El siguiente ejemplo de entrada de CloudTrail registro registra una solicitud fallida de una Decrypt operación con una clave KMS en un [almacén de claves externo](#). CloudWatch registra las solicitudes que fallan, además de las solicitudes correctas. Al registrar un error, la entrada del CloudTrail registro incluye la excepción (ErrorCode) y el mensaje de error correspondiente (ErrorMessage).

Si la solicitud fallida llegó a su proxy del almacén de claves externo, como en este ejemplo, puede usar el valor requestId para asociar la solicitud fallida a la solicitud correspondiente que registre el proxy de su almacén de claves externo, si los proporciona.

Para obtener ayuda con las solicitudes de Decrypt en almacenes de claves externos, consulte [Errores de descifrado](#).

```
{
  "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"
}

```

Descifra con una clave de cifrado simétrica estándar a través de una conexión TLS poscuántica

A continuación se muestra un ejemplo de entrada de CloudTrail registro para una Decrypt operación con una clave de cifrado simétrica estándar a través de una conexión TLS poscuántica. El campo KeyExchange de la sección menciona. [tlsDetails X25519MLKEM768](#) [Se trata de un algoritmo híbrido que combina la curva elíptica Diffie-Hellman \(ECDH\) sobre la curva 25519, un algoritmo clásico de intercambio de claves que se utiliza actualmente en el TLS, con el mecanismo de encapsulación de claves basado en módulos \(\), un algoritmo de cifrado y establecimiento de claves públicas que el Instituto Nacional de Estándares y Tecnología \(NISTML-KEM\) ha designado como su primer algoritmo estándar de acuerdo de claves poscuántico.](#)

```

{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [DeleteAlias](#) operación. Para obtener información sobre la eliminación de alias, consulte [Eliminación de un alias](#).

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

```
{
```

```

    "eventVersion": "1.02",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::111122223333:user/Alice",
      "accountId": "111122223333",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice",
      "sessionContext": {
        "attributes": {
          "mfaAuthenticated": "false",
          "creationDate": "2014-11-04T00:52:27Z"
        }
      }
    },
    "eventTime": "2014-11-04T00:52:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteAlias",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "aliasName": "alias/my_alias"
    },
    "responseElements": {
      "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d9542792-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "12f48554-bb04-4991-9cfc-e7e85f68eda0",
    "readOnly": false,
    "resources": [{
      "ARN": "arn:aws:kms:us-east-1:111122223333:alias/my_alias",
      "accountId": "111122223333"
    },
    {
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

DeleteCustomKeyStore

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [DeleteCustomKeyStore](#) operación. Para obtener información acerca de crear los almacenes de claves personalizadas, consulte [Eliminar un almacén de AWS CloudHSM claves](#).

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-21T20:17:32Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DeleteCustomKeyStore",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "customKeyStoreId": "cks-1234567890abcdef0"
  },
  "responseElements": null,
  "additionalEventData": {
    "customKeyStoreName": "ExampleKeyStore",
    "clusterId": "cluster-1a23b4cdefg"
  },
  "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
  "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333"
}
```

DeleteExpiredKeyMaterial

Al importar material clave a una AWS KMS key (clave KMS), puede establecer una fecha y hora de caducidad para ese material clave. AWS KMS registra una entrada en el CloudTrail registro al

[importar el material clave](#) (con la configuración de caducidad) y al AWS KMS eliminar el material clave caducado. Para obtener información sobre cómo crear clave KMS con material de claves importado, consulte [Importación de material clave para AWS KMS llaves](#).

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro que se genera al AWS KMS eliminar el material clave caducado.

```
{
  "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

Si importa material clave a una clave de KMS, puede eliminar el material clave importado en cualquier momento mediante la [DeleteImportedKeyMaterial](#) operación. Cuando se elimina el material de claves importado de una clave de KMS, el estado de la clave de KMS cambia a `PendingImport` y no se la puede usar en ninguna operación criptográfica. Para obtener más información, consulte [Eliminación del material de claves importado](#).

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada para la `DeleteImportedKeyMaterial` operación.

```
{
  "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

Estos ejemplos muestran la entrada de AWS CloudTrail registro que se genera cuando se elimina una clave de KMS. Para eliminar una clave KMS, utilice la [ScheduleKeyDeletion](#) operación. Una vez transcurrido el período de espera especificado, AWS KMS elimina la clave KMS y registra una entrada como la siguiente en el CloudTrail registro para registrar ese evento.

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

Para ver un ejemplo de la entrada de CloudTrail registro de la [ScheduleKeyDeletion](#) operación, consulte [ScheduleKeyDeletion](#). Para obtener más información acerca de cómo eliminar claves KMS, consulte [Eliminar un AWS KMS key](#).

El siguiente ejemplo de entrada de CloudTrail registro registra una `DeleteKey` operación de una clave KMS con el material clave incluido AWS KMS.

```

{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2020-07-31T00:07:00Z",

```

```

    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "b25f9cda-74e1-4458-847b-4972a0bf9668",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
  }

```

La siguiente entrada de CloudTrail registro registra la DeleteKey operación de una clave KMS en un [almacén de claves 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": {
      "customKeyId": "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [DescribeCustomKeyStores](#) operación. Para obtener información acerca de la visualización de los almacenes de claves personalizadas, consulte [Ver un almacén de AWS CloudHSM claves](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-21T20:17:32Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeCustomKeyStores",
  "awsRegion": "us-east-1",

```

```

    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "2ea1735f-628d-43e3-b2ee-486d02913a78",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}

```

DescribeKey

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [DescribeKey](#) operación. AWS KMS registra una entrada como la siguiente al llamar a la DescribeKey operación o al [ver las claves de KMS](#) en la AWS KMS consola. Esta llamada es el resultado de ver una clave en la consola AWS KMS de administración.

```

{
    "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [DisableKey](#) operación. Para obtener información sobre cómo activar y desactivar la AWS KMS keys entrada AWS KMS, consulte [Habilitar y deshabilitar claves](#).

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

```

{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:43Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DisableKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": {

```

```

    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "12345126-30d5-4b28-98b9-9153da559963",
  "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
  "readOnly": false,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

DisableKeyRotation

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [DisableKeyRotation](#) operación. Para obtener información acerca de la rotación de claves, consulte [Rotate \(Rotar\) 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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [DisconnectCustomKeyStore](#) operación. Para obtener información sobre cómo desconectar un almacén de claves personalizado, consulte [Desconectar un almacén de AWS CloudHSM claves](#).

```

{
  "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": {
    "customKeyId": "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [EnableKey](#) operación. Para obtener información sobre cómo habilitar y deshabilitar la AWS KMS keys entrada AWS KMS, consulte.. [Habilitar y deshabilitar claves](#)

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

```

{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:20Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "EnableKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}

```



```

    },
    "requestID": "d528a6fb-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "be393928-3629-4370-9634-567f9274d52e",
    "readOnly": false,
    "resources": [{
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}

```

EnableKeyRotation

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una llamada a la [EnableKeyRotation](#) operación. Para ver un ejemplo de la entrada de CloudTrail registro que se escribe cuando se gira la clave, consulte [RotateKey](#). Para obtener más información acerca de la rotación AWS KMS keys, consulte [Rotate \(Rotar\) AWS KMS keys](#).

Note

[rotation-period](#) es un parámetro de solicitud opcional. Si no especifica un período de rotación al habilitar la rotación automática de claves, el valor predeterminado es de 365 días.

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen el ARN clave de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva el ARN de clave.

```

{
    "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la operación de [cifrado](#).

```

{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [GenerateDataKey](#) operación.

```
{
```

```

    "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [GenerateDataKeyPair](#) operación. En este ejemplo se registra una operación que genera un par de claves de RSA cifrado bajo una AWS KMS key de cifrado 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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [GenerateDataKeyPairWithoutPlaintext](#) operación. En este ejemplo se registra una operación que genera un par de claves de RSA cifrado bajo un AWS KMS key de cifrado simétrico.

```

{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [GenerateDataKeyWithoutPlaintext](#) operación.

```

{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [GenerateMac](#) operación.


```
{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [GenerateRandom](#) operación. Como esta operación no utiliza un AWS KMS key, el `resources` campo está vacío.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateRandom",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
  "readOnly": true,
  "resources": [],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

GetKeyLastUsage

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [GetKeyLastUsage](#) operación. AWS KMS registra una entrada como la siguiente al llamar a la [GetKeyLastUsage](#) operación o al [ver los detalles clave del KMS](#) en la AWS KMS consola.

```
{
  "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": "2014-11-04T00:52:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GetKeyLastUsage",
  "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"
}
```

```

    "eventTime": "2026-04-27T17:11:06Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetKeyLastUsage",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "93746dd6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "4aa7e4d5-d047-452a-a5a6-2cce282a7e82",
    "readOnly": true,
    "resources": [{
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }

```

GetKeyPolicy

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [GetKeyPolicy](#) operación. Para obtener información acerca de cómo ver la política de claves de una clave KMS, consulte [Visualización de políticas de claves](#).

```

{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de la [GetKeyRotationStatus](#) operación. Para obtener información acerca de la rotación automática y bajo demanda de material de claves para una clave KMS, consulte [Rotate \(Rotar\) 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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al utilizar la [GetParametersForImport](#) operación. Esta operación devuelve la clave pública y el token de importación que se utiliza al importar material de claves en una clave KMS. La misma CloudTrail entrada se registra cuando se utiliza la `GetParametersForImport` operación o se utiliza la AWS KMS consola para [descargar la clave pública y el token de importación](#).

```

{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al utilizar la [ImportKeyMaterial](#) operación. La misma CloudTrail entrada se graba cuando se utiliza la ImportKeyMaterial operación o se utiliza la AWS KMS consola para [importar material clave](#) a un AWS KMS key.

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

```

{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",

```

```

        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-21T05:42:31Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ImportKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "validTo": "May 21, 2025, 5:47:45 AM",
        "expirationModel": "KEY_MATERIAL_EXPIRES",
        "importType": "NEW_KEY_MATERIAL",
        "keyMaterialDescription": "ExampleKeyMaterialA"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "89e10ee7-a612-414d-95a2-a128346969fd",
    "eventID": "c7abd205-a5a2-4430-bbfa-fc10f3e2d79f",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}

```

}

ListAliases

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [ListAliases](#) operación. Como esta operación no utiliza ningún alias concreto AWS KMS key, el `resources` campo está vacío. Para obtener información sobre la visualización de los alias en AWS KMS, consulte [Búsqueda del nombre del alias y el ARN de alias para una clave de 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": "  
"eyJjIjoiaWxpYXNvZTUyMmVtYTMwNC00YzEwLTliZWltYTJjZjA3NmJlbnQ2OTJhIiwibSI6ImFsaWFzL2U1NGNjbMTkzMjkzLnRlc2UiLCJlbGljaXRpbGUiOiJkaW8xIn0=  
    },  
    "responseElements": null,  
    "requestID": "bfef6c190-63bc-11e4-bc2b-4198b6150d5c",  
    "eventId": "a27dda7b-76f1-4ac3-8b40-42dfba77bcd6",  
    "readOnly": true,  
    "resources": [],  
    "eventType": "AwsApiCall",  
    "recipientAccountId": "111122223333"  
}
```


ListKeyRotations

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de la [ListKeyRotations](#) operación. Para obtener información acerca de la rotación automática y bajo demanda de material de claves para una clave KMS, consulte [Rotate \(Rotar\) 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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [PutKeyPolicy](#) operación. Para obtener información sobre cómo actualizar una política de claves, consulte [Cambio de una política de claves](#).

```

{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [ReEncrypt](#) operación. El `resources` campo de esta entrada de registro especifica dos AWS KMS keys, la clave KMS de origen y la clave KMS de destino, en ese orden.

```

{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [ReplicateKey](#) operación. Una ReplicateKey solicitud da como resultado una ReplicateKey operación y una [CreateKey](#) operación.

Para obtener información acerca de la replicación de claves de varias regiones, consulte [Creación de claves de réplica de varias regiones](#).

```
{
  "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    \n    \"Effect\": \"Allow\", \n    \"Principal\": {\"AWS\": \"arn:aws:iam::012345678901:user/Bob\"}, \n    \"Action\": \"kms:CreateGrant\", \n    \"Resource\": \"*\" \n  }, {\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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [RetireGrant](#) operación. Para obtener más información sobre las concesiones que van a retirarse, consulte [Retiro y revocación de concesiones](#).

```
{
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [RevokeGrant](#) operación. Para obtener más información sobre las concesiones que van a retirarse, consulte [Retiro y revocación de concesiones](#).

```
{
  "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

Estos ejemplos muestran las entradas de AWS CloudTrail registro de las operaciones que giran AWS KMS keys. Para obtener más información acerca de rotación de claves KMS, consulte [Rotate \(Rotar\) AWS KMS keys](#).

El siguiente ejemplo muestra una entrada de CloudTrail registro para la operación que rota una clave KMS de cifrado simétrico en la que está habilitada la rotación automática de claves. Para obtener información acerca de cómo habilitar la rotación automática, consulte [Rotate \(Rotar\) AWS KMS keys](#).

Para ver un ejemplo de la entrada de CloudTrail registro que registra la `EnableKeyRotation` operación, 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"
}

```

El siguiente ejemplo muestra una entrada de CloudTrail registro para una rotación bajo demanda iniciada por la [RotateKeyOnDemand](#) operación. Para obtener información acerca de la rotación de claves de KMS de cifrado simétricas bajo demanda, consulte [Realización de la rotación de claves bajo demanda](#).

Para ver un ejemplo de la entrada de CloudTrail registro que registra la RotateKeyOnDemand operación, 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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [RotateKeyOnDemand](#) operación. Para ver un ejemplo de la entrada de CloudTrail registro que se escribe cuando se gira la clave, consulte [RotateKey](#). Para obtener información acerca de la rotación bajo demanda de material de claves para una clave KMS, consulte [Realización de la rotación de claves bajo 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

En estos ejemplos se muestran las entradas de AWS CloudTrail registro de la [ScheduleKeyDeletion](#) operación.

Para ver un ejemplo de la entrada de CloudTrail registro que se escribe cuando se elimina la clave, consulte [DeleteKey](#). Para obtener información acerca de cómo eliminar AWS KMS keys, consulte [Eliminar un AWS KMS key](#).

En el siguiente ejemplo se registra una solicitud de ScheduleKeyDeletion para una clave KMS de una región.

```

{
  "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"
  }
}

```

En el siguiente ejemplo se registra una solicitud de `ScheduleKeyDeletion` para una clave KMS de varias regiones con claves de réplica.

Como AWS KMS no se eliminará una clave multirregional hasta que se eliminen todas sus claves de réplica, en el `responseElements` campo, el `keyState` es `PendingReplicaDeletion` y el `deletionDate` campo se omite.

```

{
  "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"
}

```

En el siguiente ejemplo, se registra una `ScheduleKeyDeletion` solicitud de clave KMS en un almacén de [claves AWS CloudHSM personalizado](#).

```
{
```

```

"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

En estos ejemplos se muestran las entradas de AWS CloudTrail registro de la operación de [firma](#).

El siguiente ejemplo muestra una entrada de CloudTrail registro para una operación de [firma](#) que utiliza una clave RSA KMS asimétrica para generar una firma digital para un archivo.

```
{  
  "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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al AWS KMS sincronizar una clave [multirregional](#). La sincronización implica llamadas entre regiones para copiar las [propiedades compartidas](#) de una clave principal multirregional en sus claves de réplica. AWS KMS sincroniza las claves multirregionales periódicamente para garantizar que todas las claves multirregionales relacionadas tengan el mismo material clave.

El `resources` elemento de la entrada de CloudTrail registro incluye la clave ARN de la clave principal multirregional, incluida la suya. Región de AWS Las claves de réplica de varias regiones relacionadas y sus Regiones no se enumeran en esta entrada de registro.

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2020-11-18T02:04:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "SynchronizeMultiRegionKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "12345681-de97-42e9-bed0-b02ae1abd8dc",
}

```

```

    "eventID": "abcdec99-2b5c-4670-9521-ddb8f031e146",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }
}

```

TagResource

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una llamada a la [TagResource](#) operación para agregar una etiqueta con una clave de etiqueta Department y un valor de etiqueta de IT.

Para ver un ejemplo de una entrada de UntagResource CloudTrail registro que se escribe cuando se gira la clave, consulte [UntagResource](#). Para obtener información sobre el etiquetado AWS KMS keys, consulte. [Etiquetas en 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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una llamada a la [UntagResource](#) operación para eliminar una etiqueta con una clave de etiqueta de Dept.

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelve la clave ARN.

Para ver un ejemplo de una entrada de TagResource CloudTrail registro, consulte [TagResource](#). Para obtener más información acerca del etiquetado AWS KMS keys, consulte [Etiquetas en 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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [UpdateAlias](#) operación. El elemento `resources` incluye campos para los recursos de alias y clave KMS. Para obtener información sobre la creación de alias en AWS KMS, consulte [Crear alias](#).

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-11-13T23:18:15Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "UpdateAlias",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "aliasName": "alias/my_alias",
    "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "d9472f40-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "f72d3993-864f-48d6-8f16-e26e1ae8dff0",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:alias/my_alias"
    },
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ]
}
```

```
],  
  "eventType": "AwsApiCall",  
  "recipientAccountId": "111122223333"  
}
```

UpdateCustomKeyStore

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [UpdateCustomKeyStore](#) operación para actualizar el ID de clúster de un almacén de claves personalizado. Para obtener información acerca de cómo editar los almacenes de claves personalizadas, consulte [Editar la configuración AWS CloudHSM del almacén de claves](#).

```
{  
  "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": {  
    "customKeyId": "cks-1234567890abcdef0",  
    "clusterId": "cluster-1a23b4cdefg"  
  },  
  "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"  
}
```

```
}
```

UpdateKeyDescription

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro generada al llamar a la [UpdateKeyDescription](#) operación.

```
{
  "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-4658-8427-9dc5b1ea791",
  "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

El siguiente ejemplo muestra las entradas de AWS CloudTrail registro que se generan al llamar a la [UpdatePrimaryRegion](#) operación con una [clave multirregional](#).

La UpdatePrimaryRegion operación escribe dos entradas de CloudTrail registro: una en la región con la clave principal multirregional que se convierte en una clave de réplica y otra en la región con una clave de réplica multirregional que se convierte en clave principal.

CloudTrail las entradas de registro de esta operación registradas a partir de diciembre de 2022 incluyen la clave ARN de la clave KMS afectada en el `responseElements.keyId` valor, aunque esta operación no devuelva la clave ARN.

El siguiente ejemplo muestra una entrada de CloudTrail registro para la región UpdatePrimaryRegion en la que la clave multirregional pasó de ser una clave principal a una clave de réplica (us-west-2). El campo `primaryRegion` muestra la región que ahora aloja la clave principal (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"
}

```

El siguiente ejemplo representa la entrada de CloudTrail registro de la región UpdatePrimaryRegion en la que la clave multirregional pasó de ser una clave de réplica a una clave principal (ap-northeast-1). Esta entrada de registro no identifica la región principal 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

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro para la [VerifyMac](#) operación.

```

{
  "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

Estos ejemplos muestran las entradas de AWS CloudTrail registro de la operación [Verify](#).

El siguiente ejemplo muestra una entrada de CloudTrail registro para una operación de [verificación](#) que utiliza una clave RSA KMS asimétrica para verificar una firma 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"
  }
}

```

EC2 Ejemplo uno de Amazon

El siguiente ejemplo registra a una entidad principal de IAM que crea un volumen cifrado con la clave de volumen predeterminada de la consola de EC2 administración de Amazon.

El siguiente ejemplo muestra una entrada de CloudTrail registro en la que la usuaria Alice crea un volumen cifrado con una clave de volumen predeterminada en la consola EC2 de administración de Amazon. El EC2 registro del archivo de registro incluye un `volumeId` campo con un valor de `vol-13439757`. El AWS KMS registro contiene un `encryptionContext` campo con un valor de `aws:ebs:id: vol-13439757`. Del mismo modo, coinciden `principalId` y `accountId` entre los dos registros. Los registros reflejan el hecho de que crear un volumen cifrado genera una clave de datos que se utiliza para cifrar el contenido del volumen.

```

{
  "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"
  }
]
}

```

EC2 Ejemplo dos de Amazon

En el siguiente ejemplo, un principal de IAM que ejecuta una EC2 instancia de Amazon crea y monta un volumen de datos que está cifrado con una clave KMS. Esta acción genera varios CloudTrail registros de registro. Para obtener más información sobre Amazon EBS y el cifrado, consulte [Requisitos del cifrado de Amazon EBS](#).

Cuando se crea el volumen, Amazon EC2, actuando en nombre del cliente, obtiene una clave de datos cifrada de AWS KMS (`GenerateDataKeyWithoutPlaintext`). Luego crea una concesión (`CreateGrant`) que le permite descifrar la clave de datos. Cuando se monta el volumen, Amazon EC2 llama AWS KMS para descifrar la clave de datos (`Decrypt`).

El `instanceId` de la EC2 instancia de Amazon, "i-81e2f56c", aparece en el `RunInstances` evento. El mismo ID de instancia califica el `granteePrincipal` de la concesión que se crea ("111122223333:aws:ec2-infrastructure:i-81e2f56c") y el rol asumido que es la entidad principal en la llamada `Decrypt` ("arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/i-81e2f56c").

El [ARN clave](#) de la clave KMS que protege el volumen de datos aparece en las tres AWS KMS llamadas (CreateGrantGenerateDataKeyWithoutPlaintext, yDecrypt). `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`

```
{
  "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-bbc8-3d1b7cfff49b0",
  "readOnly": true,
  "resources": [
    {
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
]

```

```
}
```

Supervise las claves de KMS con Amazon CloudWatch

Puedes monitorizar tu AWS KMS keys uso de [Amazon CloudWatch](#), un AWS servicio que recopila y procesa datos sin procesar para AWS KMS convertirlos en métricas legibles y prácticamente en tiempo real. Estos datos se registran durante un periodo de dos semanas para que pueda obtener acceso a información histórica y conocer mejor el uso de sus claves KMS y sus cambios a lo largo del tiempo.

Puedes usar Amazon CloudWatch para avisarte de eventos importantes, como los siguientes.

- El material clave importado de una clave KMS se acerca a su fecha de vencimiento.
- Se sigue utilizando una clave KMS pendiente de eliminación.
- El material clave de una clave KMS se rotó automáticamente.
- Se ha eliminado una clave KMS.

También puedes crear una CloudWatch alarma de [Amazon](#) que te avise cuando tu porcentaje de solicitudes alcance un porcentaje determinado del valor de la cuota. Para obtener más información, consulta [Gestiona tus tasas de solicitudes de AWS KMS API mediante Service Quotas y Amazon CloudWatch](#) en el blog AWS de seguridad.

AWS KMS métricas y dimensiones

AWS KMS predefine CloudWatch las métricas de Amazon para que le resulte más fácil monitorear los datos críticos y crear alarmas. Puedes ver las AWS KMS métricas mediante la API de Amazon Consola de administración de AWS y la CloudWatch API.

En esta sección, se enumeran todas AWS KMS las métricas y las dimensiones de cada una de ellas, y se proporciona una guía básica para crear CloudWatch alarmas en función de estas métricas y dimensiones.

Note

Nombre del grupo de dimensiones:

Para ver una métrica en la CloudWatch consola de Amazon, en la sección Métricas, selecciona el nombre del grupo de dimensiones. Luego, puede filtrar por Nombre de la

métrica. Este tema incluye el nombre de la métrica y el nombre del grupo de dimensiones de cada métrica de AWS KMS .

Puedes ver AWS KMS las métricas mediante la API de Amazon Consola de administración de AWS y la CloudWatch API. Para obtener más información, consulta [Ver las métricas disponibles](#) en la Guía del CloudWatch usuario de Amazon.

Temas

- [SuccessfulRequest](#)
- [SecondsUntilKeyMaterialExpiration](#)
- [Nube HSMKey StoreThrottle](#)
- [ExternalKeyStoreThrottle](#)
- [XksProxyCertificateDaysToExpire](#)
- [XksProxyCredentialAge](#)
- [XksProxyErrors](#)
- [XksExternalKeyManagerStates](#)
- [XksProxyLatency](#)

SuccessfulRequest

El número de solicitudes correctas de operaciones criptográficas en una clave de KMS específica. Al usar la `SuccessfulRequest` métrica, puedes aplicar un filtrado a nivel clave al uso de la AWS KMS API en. CloudWatch La estadística Sum de esta métrica define el número total de solicitudes satisfactorias durante el periodo.

Utilice esta métrica para identificar qué claves de KMS consumen la mayor parte de su cuota de solicitudes o son las que más contribuyen a los cargos de la API. También puedes crear una CloudWatch alarma basada en la `SuccessfulRequest` métrica para que te notifique los patrones anómalos de uso de la AWS KMS API. Estas alertas pueden ayudar a identificar los flujos de trabajo ineficientes que podrían superar, de manera no intencional, las cuotas de solicitudes o generar cargos inesperados.

Dimensiones para **SuccessfulRequest**

Dimensión	Description (Descripción)
KeyArn	Valor para cada clave de KMS.
Operación	Valor para cada operación AWS KMS de API. Esta métrica se aplica únicamente a operaciones criptográficas.

En el caso de [ReEncrypt](#) las operaciones, la `SuccessfulRequest` métrica incluye dimensiones para las claves KMS de origen y destino.

Dimensión	Description (Descripción)
SourceKeyArn	Valor de la clave de KMS que descifró el texto cifrado.
DestinationKeyArn	Valor de la clave de KMS que volvió a cifrar los datos.
Operación	Valor para cada operación de AWS KMS API, en este caso, <code>ReEncrypt</code> .

SecondsUntilKeyMaterialExpiration

La cantidad de segundos que restan hasta la primera fecha de caducidad del [material de clave importado](#) de una clave de KMS. Esta métrica es válida solo para las claves de KMS con material de clave importado (un [origen de material de clave](#) de `EXTERNAL`) y una fecha de caducidad.

Utilice esta métrica para realizar un seguimiento del tiempo que resta hasta la primera fecha de caducidad del material de claves importado. Cuando ese tiempo cae por debajo de un umbral que usted define, debe volver a importar el material de claves con una nueva fecha de caducidad para mantener la clave de KMS en uso. La métrica `SecondsUntilKeyMaterialExpiration` es específica de una clave de KMS. No puede usar esta métrica para supervisar varias claves de KMS o claves de KMS que pueda crear en el futuro. Si necesita ayuda para crear una CloudWatch alarma para supervisar esta métrica, consulte [Cree una CloudWatch alarma de caducidad del material clave importado](#).

La estadística más útil para esta métrica es **Minimum**, que le indica la menor cantidad de tiempo restante para todos los puntos de datos en el período estadístico especificado. La única unidad válida para esta métrica es **Seconds**.

Nombre del grupo de dimensiones: Per-Key Metrics

Dimensiones para **SecondsUntilKeyMaterialExpiration**

Dimensión	Descripción; relacionada con AWS
KeyId	Valor para cada clave de KMS.

Al programar una [eliminación de claves](#) de una clave KMS, AWS KMS aplica un periodo de espera antes de eliminar la clave KMS. Puede usar el periodo de espera para asegurarse de que no necesita la clave KMS ni ahora ni en el futuro. También puede configurar una CloudWatch alarma para que le avise si una persona o aplicación intenta utilizar la clave KMS en una [operación criptográfica](#) durante el período de espera. Si recibe una notificación de una alarma de este tipo, puede cancelar la eliminación de la clave KMS.

Para obtener instrucciones, consulte [Creación de una alarma que detecte el uso de una eliminación pendiente de una clave KMS](#).

Nube HSMKey StoreThrottle

El número de solicitudes de operaciones criptográficas en las claves de KMS de cada almacén de AWS CloudHSM claves que se AWS KMS limita (responde con una). `ThrottlingException` Esta métrica solo se aplica a los almacenes de claves AWS CloudHSM .

La `CloudHSMKeyStoreThrottle` métrica se aplica solo a las claves de KMS de un almacén de AWS CloudHSM claves y solo a las solicitudes de [operaciones criptográficas](#). AWS KMS [limita estas solicitudes cuando la tasa de solicitudes](#) supera la [cuota de solicitudes del almacén de claves personalizado de su AWS CloudHSM almacén](#) de claves. Esta métrica también incluye la limitación por clúster. AWS CloudHSM

Nombre del grupo de dimensiones: Keystore Throttle Metrics

Dimensión	Description (Descripción)
CustomKeyStoreId	Valor para cada almacén de AWS CloudHSM claves.
KmsOperation	Valor para cada operación AWS KMS de API. Esta métrica se aplica solo a las operaciones criptográficas en las claves de KMS en un almacén de claves de AWS CloudHSM .
KeySpec	Valor para cada tipo de clave de KMS. La única especificación clave admitida para las claves de KMS en un almacén de claves es AWS CloudHSM SYMMETRIC_DEFAULT.

ExternalKeyStoreThrottle

El número de solicitudes de operaciones criptográficas en las claves de KMS en cada almacén de claves externo que se limita (responde con AWS KMS un). `ThrottlingException` Esta métrica se aplica únicamente a los [almacenes de claves externos](#).

[La ExternalKeyStoreThrottle métrica solo se aplica a las claves KMS de un almacén de claves externo y solo a las solicitudes de operaciones criptográficas.](#) AWS KMS [limita estas solicitudes cuando la tasa de solicitudes](#) supera la [cuota de solicitudes del almacén de claves personalizado del almacén](#) de claves externo. Esta métrica no incluye la limitación por parte del proxy del almacén de claves externo ni del administrador de claves externo.

Usa esta métrica para revisar y ajustar el valor de la cuota de solicitudes de su almacén de claves personalizado. Si esta métrica indica que las solicitudes de estas claves de KMS AWS KMS se limitan con frecuencia, podría considerar la posibilidad de solicitar un aumento en el valor de la cuota de solicitudes del almacén de claves personalizado. Para conocer más detalles, consulte [Solicitud de un aumento de cuota](#) en la Guía del usuario de Service Quotas.

Si con frecuencia recibe errores `KMSInvalidStateException` con un mensaje que explica que la solicitud fue rechazada “debido a una tasa de solicitudes muy alta” o “debido a que el proxy del almacén de claves externo no respondió a tiempo”, podría indicar que su administrador de claves externo o el proxy de almacén de claves externo no puede seguir el ritmo de la tasa de solicitudes actual. Si es posible, reduzca el porcentaje de solicitudes. También podría considerar solicitar una disminución en el valor de la cuota de solicitudes del almacén de claves personalizado. Reducir

este valor de cuota puede aumentar la limitación (y el valor de la `ExternalKeyStoreThrottle` métrica), pero indica que AWS KMS se está rechazando el exceso de solicitudes rápidamente antes de enviarlas al proxy del almacén de claves externo o al administrador de claves externo. Para solicitar una reducción de la cuota, visite el [Centro de AWS Support](#) y crea un caso.

Nombre del grupo de dimensiones: Keystore Throttle Metrics

Dimensión	Description (Descripción)
CustomKeyStoreId	Valor para cada almacén de claves externo.
KmsOperation	Valor para cada operación de API AWS KMS . Esta métrica se aplica solo a las operaciones criptográficas en las claves de KMS en un almacén de claves externo.
KeySpec	Valor para cada tipo de clave de KMS. La única especificación de clave admitida para las claves de KMS en un almacén de claves externo es SYMMETRIC_DEFAULT.

XksProxyCertificateDaysToExpire

El número de días que faltan para que venza el certificado TLS del [punto de conexión del proxy del almacén de claves externo](#) (`XksProxyUriEndpoint`). Esta métrica se aplica únicamente a los [almacenes de claves externos](#).

Usa esta métrica para crear una CloudWatch alarma que te notifique la próxima caducidad de tu certificado TLS. Cuando el certificado caduque, AWS KMS no podrá comunicarse con el proxy del almacén de claves externo. No se podrá acceder a todos los datos protegidos por las claves de KMS en su almacén de claves externo hasta que renueve el certificado.

Una alarma previene que caduque la certificación que le puede impedir a usted acceder a sus recursos encriptados. Configure la alarma para que su organización tenga tiempo de renovar el certificado antes de que venza.

Nombre del grupo de dimensiones: XKS Proxy Certificate Metrics

Dimensión	Description (Descripción)
CustomKey StoreId	Valor para cada almacén de claves externo.
CertificateName	Nombre del sujeto (CN) en el certificado TLS.

Puede crear CloudWatch alarmas en función de las métricas de los almacenes de claves externos y de las claves de KMS de los almacenes de claves externos. Para obtener instrucciones, consulte [Monitoreo de almacenes de claves externos](#).

XksProxyCredentialAge

La cantidad de días que transcurrieron desde que la [credencial de autenticación del proxy](#) del almacén de claves externo actual (XksProxyAuthenticationCredential) se asoció con el almacén de claves externo. Este recuento comienza cuando introduce la credencial de autenticación como parte para crear o actualizar su almacén de claves externo. Esta métrica se aplica únicamente a los [almacenes de claves externos](#).

Este valor está diseñado para recordarle la antigüedad de su credencial de autenticación. Sin embargo, dado que empezamos el recuento cuando asocia la credencial a su almacén de claves externo, no cuando crea su credencial de autenticación en el proxy del almacén de claves externo, es posible que este no sea un indicador preciso de la antigüedad de las credenciales en el proxy.

Utilice esta métrica para crear una CloudWatch alarma que le recuerde que debe cambiar la credencial de autenticación del proxy del almacén de claves externo.

Nombre del grupo de dimensiones: Per-Keystore Metrics

Dimensión	Description (Descripción)
CustomKey StoreId	Valor para cada almacén de claves externo.

Puede crear CloudWatch alarmas en función de las métricas de los almacenes de claves externos y de las claves KMS de los almacenes de claves externos. Para obtener instrucciones, consulte [Monitoreo de almacenes de claves externos](#).

XksProxyErrors

El número de excepciones relacionadas con AWS KMS las solicitudes a su [proxy de almacén de claves externo](#). Este recuento incluye las excepciones a las que vuelve el proxy del almacén de claves externo AWS KMS y los errores de tiempo de espera que se producen cuando el proxy del almacén de claves externo no responde AWS KMS dentro del intervalo de tiempo de espera de 250 milisegundos. Esta métrica se aplica únicamente a los [almacenes de claves externos](#).

Puede utilizar esta métrica para realizar un seguimiento del porcentaje de errores de claves de KMS en su almacén de claves externo. Revela los errores más frecuentes, para que pueda priorizar sus esfuerzos de ingeniería. Por ejemplo, las claves de KMS que generan altas tasas de errores no reintentables pueden indicar un problema con la configuración de su almacén de claves externo. Para ver la configuración de su almacén de claves externo, consulte [Visualización de almacenes de claves externos](#). Para editar la configuración de su almacén de claves externo, consulte [Edición de las propiedades del almacén de claves externo](#).

Nombre del grupo de dimensiones: XKS Proxy Error Metrics

Dimensión	Description (Descripción)
CustomKeyStoreId	Valor para cada almacén de claves externo.
KmsOperation	Valor para cada operación de AWS KMS API que generó una solicitud al proxy XKS.
XksOperation	Valor para cada operación de la API del proxy del almacén de claves externo .
KeySpec	Valor para cada tipo de clave de KMS. La única especificación de clave admitida para las claves de KMS en un almacén de claves externo es SYMMETRIC_DEFAULT.
ErrorType	Valores:

Dimensión	Description (Descripción)
	- Errores que se pueden volver a intentar: es probable que sean transitorios, como los errores de red. - Errores que no se pueden volver a intentar: pueden indicar un problema con la configuración del almacén de claves personalizado o con componentes externos. - N/A: Solicitud exitosa; sin errores
Exception Name	Valores:
	- Nombre de la excepción - Ninguna: solicitud exitosa; sin errores

Puede crear CloudWatch alarmas en función de las métricas de los almacenes de claves externos y de las claves de KMS de los almacenes de claves externos. Para obtener instrucciones, consulte [Monitoreo de almacenes de claves externos](#).

XksExternalKeyManagerStates

Un recuento de la cantidad de [instancias de un administrador de claves externo](#) en cada uno de los siguientes estados de condición: `Active`, `Degraded` y `Unavailable`. La información de esta métrica proviene del proxy del almacén de claves externo asociado a cada almacén de claves externo. Esta métrica se aplica únicamente a los [almacenes de claves externos](#).

Los siguientes son los estados de condición para las instancias del administrador de claves externo asociadas a un almacén de claves externo. Cada proxy de almacén de claves externo puede utilizar diferentes indicadores para medir el estado de su administrador de claves externo. Para obtener más información, consulte la documentación del proxy del almacén de claves externo.

- `Active`: el administrador de claves externo está en buen estado.
- `Degraded`: el administrador de claves externo no está en buen estado, pero aún puede atender el tráfico.
- `Unavailable`: el administrador de claves externo no puede atender el tráfico.

Utilice esta métrica para crear una CloudWatch alarma que le avise de las instancias del administrador de claves externo degradadas y no disponibles. Para determinar en qué estado se

encuentra cada instancia del administrador de claves externo, consulte sus registros de proxy del almacén de claves externo.

Nombre del grupo de dimensiones: XKS External Key Manager Metrics

Dimensión	Description (Descripción)
CustomKeyStoreId	Valor para cada almacén de claves externo.
XksExternalKeyManagerState	Valor para cada estado de condición.

Puede crear CloudWatch alarmas en función de las métricas de los almacenes de claves externos y de las claves de KMS de los almacenes de claves externos. Para obtener instrucciones, consulte [Monitoreo de almacenes de claves externos](#).

XksProxyLatency

La cantidad de milisegundos que tarda un proxy del almacén de claves externo en responder a una solicitud de AWS KMS . Si se agotó el tiempo de espera de la solicitud, el valor registrado es un límite de tiempo de espera de 250 milisegundos. Esta métrica se aplica únicamente a los [almacenes de claves externos](#).

Utilice esta métrica para evaluar el rendimiento de su proxy de almacén de claves externo y de su administrador de claves externo. Por ejemplo, si el tiempo del proxy se agota con frecuencia en las operaciones de cifrado y descifrado, consulte a su administrador de proxy externo.

Las respuestas lentas también pueden indicar que tu administrador de claves externo no puede gestionar el tráfico de solicitudes actual. AWS KMS recomienda que su administrador de claves externo pueda gestionar hasta 1800 solicitudes de operaciones criptográficas por segundo. Si su administrador de claves externo no puede gestionar la tasa de 1800 solicitudes por segundo, considere solicitar una reducción de su [cuota de solicitudes de claves de KMS en un almacén de claves personalizado](#). Las solicitudes de operaciones criptográficas que utilizan las claves de KMS en su almacén de claves externo van a responder rápido a los errores con una [excepción de limitación](#), en lugar de ser procesadas y luego rechazadas por su proxy del almacén de claves externo o administrador de claves externo.

Nombre del grupo de dimensiones: XKS Proxy Latency Metrics

Dimensión	Description (Descripción)
CustomKeyStoreId	Valor para cada almacén de claves externo.
KmsOperation	Valor para cada operación de AWS KMS API que generó una solicitud al proxy XKS.
XksOperation	Valor para cada operación de la API del proxy del almacén de claves externo .
KeySpec	Valor para cada tipo de clave de KMS. La única especificación de clave admitida para las claves de KMS en un almacén de claves externo es SYMMETRIC_DEFAULT.

Puede crear CloudWatch alarmas en función de las métricas de los almacenes de claves externos y de las claves de KMS de los almacenes de claves externos. Para obtener instrucciones, consulte [Monitoreo de almacenes de claves externos](#).

Cree una CloudWatch alarma de caducidad del material clave importado

Puede crear una CloudWatch alarma que le notifique cuando el material clave importado en una clave de KMS se acerca a su fecha de caducidad. Por ejemplo, la alarma puede avisarle cuando el plazo de caducidad esté a menos de 30 días.

Al [importar material de claves en una clave KMS](#), también puede especificar de manera opcional una fecha y una hora en la que vence el material de claves. Cuando el material clave caduca, lo AWS KMS elimina y la clave KMS queda inutilizable. Para volver a usar la clave KMS, debe [volver a importar el material de claves](#). Sin embargo, si vuelve a importar el material de claves antes de que caduque, puede evitar interrumpir los procesos que utilizan esa clave de KMS.

Esta alarma utiliza la [SecondsUntilKeyMaterialExpires métrica](#) que se AWS KMS publica en el caso de las claves CloudWatch de KMS cuyo material clave importado caduque. Cada alarma usa esta métrica para supervisar el material de claves importado para una clave de KMS en particular. No puede crear una sola alarma para todas las claves de KMS con material de claves que caduque ni una alarma para las claves de KMS que pueda crear en el futuro.

Requisitos

Se necesitan los siguientes recursos para una CloudWatch alarma que supervise la caducidad del material clave importado.

- Una clave KMS con material de claves importado que caduca.
- Un tema de Amazon SNS. Para obtener más información, consulte el [tema Creación de Amazon SNS](#) en la Guía CloudWatch del usuario de Amazon.

Crear la alarma

Siga las instrucciones de Cómo [crear una CloudWatch alarma basada en un umbral estático](#) utilizando los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Seleccionar métrica	Elija KMS y, a continuación, seleccione Per-Key Metrics (Métricas por clave). Seleccione la fila con la clave de KMS y la métrica <code>SecondsUntilKeyMaterialExpires</code> . A continuación, elija Select metric (Seleccionar métrica). La lista Metrics (Métricas) muestra la métrica <code>SecondsUntilKeyMaterialExpires</code> solo para claves KMS con material de claves importado que caduca. Si no tiene claves de KMS con estas propiedades en la cuenta y la región, esta lista está vacía.
Estadística	Mínimo
Periodo	1 minuto
Tipo de umbral	Estático
Whenever...	Siempre que <i>metric-name</i> sea mayor que 1

Cree CloudWatch alarmas para almacenes de claves externos

Puedes crear CloudWatch alarmas de Amazon basadas en métricas externas del almacén de claves para que te notifiquen cuando el valor de una métrica supere un umbral que hayas especificado. La

alarma puede enviar el mensaje a un [tema de Amazon Simple Notification Service \(Amazon SNS\)](#) o a una [política de Amazon EC2 Auto Scaling](#). Para obtener información detallada sobre CloudWatch las alarmas, consulta [Uso de CloudWatch las alarmas de Amazon](#) en la Guía del CloudWatch usuario de Amazon.

Antes de crear una CloudWatch alarma de Amazon, necesitas un tema de Amazon SNS. Para obtener más información, consulte el [tema Creación de Amazon SNS](#) en la Guía CloudWatch del usuario de Amazon.

Temas

- [Creación de una alarma para el vencimiento del certificado](#)
- [Creación de una alarma para el tiempo de espera de respuesta](#)
- [Creación de una alarma para los errores reintentables](#)
- [Creación de una alarma para los errores no reintentables](#)

Creación de una alarma para el vencimiento del certificado

Esta alarma utiliza la [XksProxyCertificateDaysToExpire](#) métrica que se AWS KMS publica CloudWatch para registrar la caducidad prevista del certificado TLS asociado al punto de conexión proxy del almacén de claves externo. No puede crear una sola alarma para todos los almacenes de claves externos de su cuenta ni una alarma para los almacenes de claves externos que pueda crear en el futuro.

Le recomendamos configurar la alarma para que le avise 10 días antes de que venza su certificado, pero debe establecer el umbral que mejor se adapte a sus necesidades.

Crear la alarma

Siga las instrucciones de [Crear una CloudWatch alarma basada en un umbral estático](#) utilizando los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Seleccionar métrica	Elija KMS y, a continuación, elija XKS Proxy Certificate Metrics (Métricas del certificado de proxy XKS).

Campo	Valor
	Seleccione la casilla de verificación junto al XksProxyCertificateName que desea monitorear. A continuación, elija Select metric (Seleccionar métrica).
Estadística	Mínimo
Periodo	5 minutos
Tipo de umbral	Estático
Whenever...	Siempre que XksProxyCertificateDaysToExpire sea Lower que 10.

Creación de una alarma para el tiempo de espera de respuesta

Esta alarma utiliza la [XksProxyLatency](#) métrica en la que se AWS KMS publica CloudWatch para registrar el número de milisegundos que tarda un proxy de almacén de claves externo en responder a una AWS KMS solicitud. No puede crear una sola alarma para todos los almacenes de claves externos de su cuenta ni una alarma para los almacenes de claves externos que pueda crear en el futuro.

AWS KMS espera que el proxy del almacén de claves externo responda a cada solicitud en un plazo de 250 milisegundos. Recomendamos configurar una alarma para que le avise cuando su proxy del almacén de claves externo tarde más de 200 milisegundos en responder, pero debe establecer el umbral que mejor se adapte a sus necesidades.

Crear la alarma

Siga las instrucciones de [Crear una CloudWatch alarma basada en un umbral estático](#) utilizando los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Seleccionar métrica	Elija KMS y, a continuación, elija XKS Proxy Latency Metrics (Métricas del certificado de proxy XKS).

Campo	Valor
	Seleccione la casilla de verificación junto al <code>KmsOperation</code> que desea monitorear. A continuación, elija <code>Select metric</code> (Seleccionar métrica).
Estadística	Media
Periodo	5 minutos
Tipo de umbral	Estático
Whenever...	Siempre que <code>XksProxyLatency</code> sea Greater que 200.

Creación de una alarma para los errores reintentables

Esta alarma utiliza la [XksProxyErrors](#) métrica en la que se AWS KMS publica CloudWatch para registrar el número de excepciones relacionadas con AWS KMS las solicitudes al proxy del almacén de claves externo. No puede crear una sola alarma para todos los almacenes de claves externos de su cuenta ni una alarma para los almacenes de claves externos que pueda crear en el futuro.

Los errores reintentables reducirán el porcentaje de fiabilidad y pueden indicar errores de red. Le recomendamos configurar una alarma para que le avise cuando se registren más de cinco errores reintentables en un periodo de un minuto, pero debe establecer el umbral que mejor se adapte a sus necesidades.

Siga las instrucciones de [Crear una CloudWatch alarma basada en un umbral estático](#) utilizando los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Seleccionar métrica	Elija la pestaña <code>Queries</code> (Consultas). Elija <code>AWS/KMS</code> para <code>Namespace</code> (Espacio de nombres). Ingrese <code>SUM(XksProxyErrors)</code> para <code>Metric name</code> (Nombre de la métrica). Ingrese <code>ErrorType = Retryable</code> para <code>Filter by</code> (Filtrar por).

Campo	Valor
	Seleccione Ejecutar. A continuación, elija Select metric (Seleccionar métrica).
Etiqueta	<i>Retryable errors</i>
Periodo	1 minuto
Tipo de umbral	Estático
Whenever...	Siempre que q1 sea Greater que 5.

Creación de una alarma para los errores no reintentables

Esta alarma utiliza la [XksProxyErrors](#) métrica que se AWS KMS publica CloudWatch para registrar el número de excepciones relacionadas con AWS KMS las solicitudes al proxy del almacén de claves externo. No puede crear una sola alarma para todos los almacenes de claves externos de su cuenta ni una alarma para los almacenes de claves externos que pueda crear en el futuro.

Los errores no reintentables pueden indicar un problema con la configuración del almacén de claves externo. Le recomendamos configurar una alarma para que le avise cuando se registren más de cinco errores no reintentables en un periodo de un minuto, pero debe establecer el umbral que mejor se adapte a sus necesidades.

Siga las instrucciones de [Crear una CloudWatch alarma basada en un umbral estático](#) utilizando los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Seleccionar métrica	Elija la pestaña Queries (Consultas). Elija AWS/KMS para Namespace (Espacio de nombres). Ingrese SUM(XksProxyErrors) para Metric name (Nombre de la métrica). Ingrese ErrorType = Non-retryable para Filter by (Filtrar por). Seleccione Ejecutar. A continuación, elija Select metric (Seleccionar métrica).

Campo	Valor
Etiqueta	<i>Non-retryable errors</i>
Periodo	1 minuto
Tipo de umbral	Estático
Whenever...	Siempre que q1 sea Greater que 5.

Determinación del uso anterior de una clave KMS

Para fines de supervisión y auditoría, es posible que desee saber cómo se ha utilizado una clave de KMS en el pasado. Por ejemplo, es posible que desee determinar si una clave de KMS se sigue utilizando activamente antes de inhabilitarla o programar su eliminación, o bien identificar las claves no utilizadas de su cuenta. Las siguientes estrategias pueden ayudarle a determinar el uso anterior de una clave de KMS.

Warning

Estas estrategias para determinar el uso anterior solo son efectivas para AWS los directores y AWS KMS las operaciones. No pueden detectar el uso de la clave que no implique llamadas a AWS KMS la API. Una vez que se genera una clave de datos a partir de una clave KMS simétrica, su uso posterior para el cifrado local o el descifrado fuera de ella no AWS KMS se refleja en la última información de uso. Del mismo modo, estas estrategias no pueden detectar el uso de la clave pública de una clave KMS asimétrica fuera de. AWS KMS Para obtener detalles acerca de los riesgos especiales de la eliminación de las claves KMS asimétricas utilizadas para la criptografía de clave pública, incluida la creación de textos cifrados que no se pueden descifrar, consulte [Deleting asymmetric KMS keys](#).

Temas

- [Examen de los permisos de clave KMS para determinar el ámbito de uso potencial](#)
- [Examine la última operación criptográfica realizada con una clave KMS](#)
- [Examine AWS CloudTrail los registros para comprobar el uso anterior](#)

Examen de los permisos de clave KMS para determinar el ámbito de uso potencial

Determinar quién o qué tiene acceso actualmente a una clave de KMS puede ayudarle a determinar hasta qué punto se utiliza la clave de KMS y si sigue siendo necesaria. Para determinar quién o qué tiene acceso actualmente a una clave de KMS, consulte [Determinar el acceso a AWS KMS keys](#).

Examine la última operación criptográfica realizada con una clave KMS

AWS KMS proporciona información de uso sobre la última operación criptográfica realizada correctamente con cada clave de KMS junto con el identificador de CloudTrail evento asociado. Esto puede facilitar el proceso de identificación de las claves de KMS no utilizadas. También puede usar la clave de `TrailingDaysWithoutKeyUsage` condición [kms:](#) en las políticas clave para evitar que las claves utilizadas recientemente se deshabiliten accidentalmente o se programe su eliminación.

Puede ver la última operación criptográfica realizada correctamente con una clave de KMS mediante la Consola de administración de AWS, AWS CLI, o la API. AWS KMS

Note

Algunos AWS servicios crean recursos que dependen de una clave KMS para la protección de datos, pero no invocan operaciones criptográficas en esa clave con frecuencia. Por ejemplo, el servicio Amazon EC2 llama AWS KMS para descifrar la clave de datos de un volumen cifrado de Amazon EBS solo cuando el volumen está adjunto a una instancia. En estos casos, no debe basarse únicamente en la última información de uso para determinar si se puede eliminar una clave de KMS. Si se elimina la clave KMS que protege un volumen de Amazon EBS, no se interrumpirá el volumen de Amazon EBS que ya está adjunto, pero los intentos posteriores de adjuntar ese volumen de Amazon EBS cifrado a otra instancia de Amazon EC2 fallarán.

Comprenda el período de seguimiento del uso

AWS KMS rastrea solo la última operación criptográfica realizada correctamente con cada clave de KMS. Puede haber un retraso de hasta una hora entre el momento en que se produce una operación criptográfica y el momento en que se registra el uso.

Al comprobar la última información de uso de una clave KMS, la respuesta incluye una fecha de inicio del seguimiento. `TrackingStartDate` es la fecha a partir de la cual AWS KMS comenzó a

registrarse la actividad criptográfica de esa clave. Use esta fecha junto con la fecha de creación de la clave para determinar su historial de uso comparando la fecha de creación de la clave con la fecha de inicio del seguimiento:

- Si hay información sobre el último uso, la clave se ha utilizado para una operación criptográfica desde que se inició el seguimiento. La respuesta incluye el tipo de operación, la marca de tiempo y el identificador de evento asociado AWS CloudTrail .
- Si la información del último uso está vacía, la clave no ha registrado operaciones criptográficas desde que se inició el seguimiento. Compare la fecha de creación de la clave con la fecha `TrackingStartDate` para determinar lo que esto significa:
 - Si la clave se creó el mismo día o después del `TrackingStartDate`, no se ha utilizado para ninguna operación criptográfica desde que se creó.
 - Si la clave se creó antes del `TrackingStartDate`, no hay registro de que la clave se haya utilizado desde que se inició el seguimiento. Sin embargo, es posible que la clave se haya utilizado antes de que se iniciara el seguimiento. Para determinar si la clave se usó anteriormente, examine sus AWS CloudTrail registros anteriores.

Warning

Al eliminar las claves no utilizadas, no se base únicamente en la información sobre el último uso. En su lugar, [desactive primero la clave](#) y controle AWS CloudTrail las `DisabledException` entradas que indiquen los intentos de utilizar la clave mientras estaba desactivada. Esto ayuda a identificar posibles dependencias y errores en la carga de trabajo.

Operaciones criptográficas rastreadas

Solo se rastrean y registran las siguientes operaciones criptográficas exitosas para generar la última información de uso. Se excluyen las operaciones no criptográficas.

- `Decrypt`
- `DeriveSharedSecret`
- `Encrypt`
- `GenerateDataKey`
- `GenerateDataKeyPair`
- `GenerateDataKeyPairWithoutPlaintext`

- `GenerateDataKeyWithoutPlaintext`
- `GenerateMac`
- `ReEncrypt`
- `Sign`
- `Verify`
- `VerifyMac`

Otras consideraciones

Tenga en cuenta lo siguiente cuando utilice la información de uso:

- Claves KMS multirregionales: en el caso de las claves KMS multirregionales, las claves principales y réplicas registran la información del último uso de forma independiente. Cada clave de un conjunto de claves multirregional mantiene su propia información de último uso.
- ReEncrypt operaciones: la ReEncrypt operación utiliza dos claves: una clave de origen para el descifrado y una clave de destino para el cifrado. La información del último uso se registra para ambas claves de forma independiente, cada una con el ID de CloudTrail evento de la cuenta del propietario de la clave correspondiente.

Puedes ver la última información de uso mediante los siguientes métodos:

Uso de AWS KMS la consola

Puede ver la última operación criptográfica realizada correctamente con una clave KMS en la página de detalles de cada clave KMS. Para conocer los procedimientos sobre cómo ver la página de detalles de una clave KMS, consulte [Acceso y enumeración de detalles de claves KMS](#).

Uso de AWS KMS la API

La [GetKeyLastUsage](#) operación devuelve información de uso de la última operación criptográfica realizada con la clave KMS especificada. Para identificar la clave KMS, utilice el [ID de clave](#) o el [ARN de clave](#).

Por ejemplo, la siguiente llamada `GetKeyLastUsage` recupera la información de uso de una clave de KMS con el ID de clave. `1234abcd-12ab-34cd-56ef-1234567890ab`

```
$ aws kms get-key-last-usage --key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

```
{
  "KeyCreationDate": 1773253425.56,
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "TrackingStartDate": 1773253425.56,
  "KeyLastUsage": {
    "Timestamp": 1773253497.0,
    "Operation": "Encrypt",
    "KmsRequestId": "040cce3e-9ef3-4651-b8cf-e47c9bafdc9b",
    "CloudTrailEventId": "2cfd5892-ea8c-4342-ad49-4b9594b06a8b"
  }
}
```

Por el contrario, la siguiente llamada a `GetKeyLastUsage` revela información de uso de una clave de KMS con el ID `0987dcba-09fe-87dc-65ba-ab0987654321` de clave.

```
$ aws kms get-key-last-usage --key-id "0987dcba-09fe-87dc-65ba-ab0987654321"
{
  "KeyCreationDate": 1672531200.0,
  "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
  "TrackingStartDate": 1773253425.56,
  "KeyLastUsage": {}
}
```

Examine AWS CloudTrail los registros para comprobar el uso anterior

Puede utilizar el historial de uso de una clave KMS para determinar si tiene textos cifrados con una clave KMS concreta.

Toda la actividad AWS KMS de la API se registra en archivos de AWS CloudTrail registro. Si ha [creado un registro en CloudTrail](#) la región en la que se encuentra su clave de KMS, puede examinar los archivos de CloudTrail registro para ver un historial de toda la actividad de la AWS KMS API de una clave de KMS concreta. Si no tienes un registro, puedes ver los eventos recientes en tu [historial de CloudTrail eventos](#). Para obtener más información sobre cómo se AWS KMS usa CloudTrail, consulte [Registro AWS KMS Llamadas a la API con AWS CloudTrail](#).

Los siguientes ejemplos muestran las entradas de CloudTrail registro que se generan cuando se utiliza una clave de KMS para proteger un objeto almacenado en Amazon Simple Storage Service (Amazon S3). En este ejemplo, el objeto se carga en Amazon S3 utilizando [Protección de datos mediante cifrado del lado del servidor con claves KMS \(SSE-KMS\)](#). Al cargar un objeto en Amazon S3 con SSE-KMS, especifique la clave KMS que se usará para proteger el objeto. Amazon S3 utiliza

la AWS KMS [GenerateDataKey](#) operación para solicitar una clave de datos única para el objeto, y este evento de solicitud se registra CloudTrail con una entrada similar a la siguiente:

```
{
  "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": "2015-09-10T23:58:18Z",
  "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"
}

```

Cuando descargue más adelante este objeto de Amazon S3, Amazon S3 enviará una Decrypt solicitud AWS KMS a para descifrar la clave de datos del objeto mediante la clave de KMS especificada. Al hacerlo, los archivos de CloudTrail registro incluyen una entrada similar a la siguiente:

```

{
  "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"
}
```

Al evaluar estas entradas de registro, puede determinar el uso anterior de una determinada clave KMS y esto puede ayudarle a determinar si desea eliminarla.

Para ver más ejemplos de cómo aparece la actividad de la AWS KMS API en tus archivos de CloudTrail registro, visita [Registro AWS KMS Llamadas a la API con AWS CloudTrail](#). Para obtener más información, CloudTrail consulta la [Guía AWS CloudTrail del usuario](#).

Supervise las claves de KMS con Amazon EventBridge

Puede usar Amazon EventBridge (anteriormente Amazon CloudWatch Events) para que le avise de los siguientes eventos importantes en el ciclo de vida de sus claves de KMS.

- El material clave de una clave de KMS tuvo rotación automática o a pedido.
- El material de clave importado en una clave KMS se ha vencido.
- Se eliminó una clave KMS cuya eliminación estaba programada.

AWS KMS se integra con Amazon EventBridge para notificarle los eventos importantes que afectan a sus claves de KMS. Cada evento se representa en [JSON \(notación de JavaScript objetos\)](#) e incluye el nombre del evento, la fecha y hora en que se produjo el evento y las personas afectadas. Puede recopilar estos eventos y establecer reglas que los dirijan a uno o más destinos, como AWS Lambda funciones, temas de Amazon SNS, colas de Amazon SQS, transmisiones en Amazon Kinesis Data Streams o destinos integrados.

Para obtener más información sobre su uso EventBridge con otros tipos de eventos, incluidos los que se emiten AWS CloudTrail cuando se registra una solicitud de read/write API, consulta la [Guía del EventBridge usuario de Amazon](#).

En los temas siguientes se describen los EventBridge eventos que se AWS KMS generan.

Rotación de CMK de KMS

AWS KMS admite la [rotación automática y bajo demanda](#) del material clave en las claves KMS de cifrado simétrico.

Cada vez que AWS KMS rota el material clave, envía un KMS CMK Rotation evento a EventBridge AWS KMS genera este evento haciendo el mejor esfuerzo posible.

A continuación se muestra un ejemplo de este 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"
  }
}
```

Vencimiento del material de claves importado de KMS

Al [importar material de claves en una clave KMS](#), también puede especificar una hora en la que vence el material de claves. Cuando el material clave caduque, lo AWS KMS elimina y envía el

KMS Imported Key Material Expiration evento correspondiente a. EventBridge AWS KMS genera este evento haciendo el mejor esfuerzo posible.

A continuación se muestra un ejemplo de este 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"
  }
}
```

Eliminación de CMK de KMS

Al programar una [eliminación de claves](#) de una clave KMS, AWS KMS aplica un periodo de espera antes de eliminar la clave KMS. Una vez finalizado el período de espera, AWS KMS elimina la clave KMS y envía un KMS CMK Deletion evento a. EventBridge AWS KMS garantiza este EventBridge evento. Debido a los reintentos, puede generar varios eventos en unos segundos que eliminan la misma clave KMS.

A continuación se muestra un ejemplo de este 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"  
  }  
}
```


Alias en AWS KMS

Un alias es un nombre fácil de recordar para una AWS KMS key. Por ejemplo, un alias le permite referirse a una clave KMS como test-key en lugar de 1234abcd-12ab-34cd-56ef-1234567890ab.

Puede usar un alias para identificar una clave KMS en la AWS KMS consola, en la [DescribeKey](#) operación y en [las operaciones criptográficas](#), como [Encrypt](#) y [GenerateDataKey](#). Los alias también facilitan el reconocimiento de un [Clave administrada de AWS](#). Los alias de estas claves KMS siempre tienen la forma aws/<service-name>. Por ejemplo, el alias de Amazon aws/dynamodb DynamoDB es. Clave administrada de AWS Puede establecer estándares de alias similares para sus proyectos, como anteponer los alias con el nombre de un proyecto o categoría.

También puede permitir y denegar el acceso a claves KMS en función de sus alias sin editar políticas ni administrar concesiones. Esta función forma parte de la AWS KMS compatibilidad con el [control de acceso basado en atributos](#) (ABAC). Para obtener más información, consulte [Uso de alias para controlar el acceso a las claves KMS](#).

Gran parte de la potencia de los alias proviene de su capacidad de cambiar la clave KMS asociada a un alias en cualquier momento. Los alias pueden hacer que su código sea más fácil de escribir y mantener. Por ejemplo, supongamos que utiliza un alias para hacer referencia a una clave KMS concreta y desea cambiar la clave KMS. En ese caso, simplemente asocie el alias con una clave KMS diferente. No es necesario realizar cambios en el código.

Los alias también facilitan la reutilización del mismo código en diferentes Regiones de AWS. Cree alias con el mismo nombre en varias regiones y asocie cada alias a una clave KMS en su región. Cuando el código se ejecuta en cada región, el alias hace referencia a la clave KMS asociada en esa región. Para ver un ejemplo, consulta [Obtención de información sobre cómo utilizar alias en las aplicaciones](#).

Puede crear un alias para una clave de KMS en la AWS KMS consola mediante la [CreateAlias](#) API o mediante la plantilla [AWS: :KMS: :Alias CloudFormation](#).

La AWS KMS API proporciona un control total de los alias de cada cuenta y región. La API incluye operaciones para crear un alias ([CreateAlias](#)), ver los nombres y los ARN de los alias ([ListAliases](#)), cambiar la clave de KMS asociada a un alias ([UpdateAlias](#)) y eliminar un alias ([DeleteAlias](#)).

Funcionamiento de los alias

Obtenga información sobre cómo funcionan los alias en AWS KMS.

Un alias es un recurso independiente AWS

Un alias no es propiedad de una clave KMS. Las acciones que realice en el alias no afectan a su clave KMS asociada. Puede crear un alias para una clave KMS y, a continuación, actualizar el alias para que esté asociado a una clave KMS diferente. Incluso puede eliminar el alias sin ningún efecto en la clave KMS asociada. Sin embargo, si elimina una clave KMS, se eliminan todos los alias asociados a esa clave KMS.

Si especifica un alias como recurso en una política de IAM, la política hace referencia al alias, no a la clave KMS asociada.

Cada alias tiene dos formatos

Al crear un alias, se especifica el nombre del alias. AWS KMS crea el alias ARN por usted.

- Un [ARN de alias](#) es un nombre de recurso de Amazon (ARN) que identifica de forma exclusiva el alias.

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

- El [nombre de alias](#) debe ser único en la cuenta y la región. En la AWS KMS API, el nombre del alias siempre lleva el prefijo. `alias/` Este prefijo se omite en la consola. AWS KMS

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

Los alias no son secretos

Los alias pueden mostrarse en texto plano en los CloudTrail registros y otros resultados. No incluya información confidencial en el nombre del alias.

Cada alias está asociado a una clave KMS a la vez

El alias y la clave KMS deben estar en la misma cuenta y región.

Puede asociar un alias a cualquier [clave gestionada por el cliente](#) en la misma región Cuenta de AWS . Sin embargo, no tiene permiso para asociar un alias con una [Clave administrada de AWS](#).

Por ejemplo, este [ListAliases](#) resultado muestra que el test-key alias está asociado exactamente a una clave de KMS de destino, que se representa mediante la TargetKeyId propiedad.

```
{
  "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
}
```

Se pueden asociar varios alias con la misma clave KMS

Por ejemplo, puede asociar los alias test-key y project-key con la misma clave 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
}
```

El alias debe ser único en la cuenta y región.

Por ejemplo, solo puede tener un alias test-key en cada cuenta y región. Los alias distinguen entre mayúsculas y minúsculas, pero los alias que solo difieren en sus mayúsculas son muy propensos a errores. No puede cambiar un nombre de alias. Sin embargo, puede eliminar el alias y crear un nuevo alias con el nombre deseado.

Puede crear alias con el mismo nombre en diferentes regiones

Por ejemplo, puede tener un alias finance-key en EE. UU. Este (Norte de Virginia) y un alias finance-key en Europa (Fráncfort). Cada alias se asociaría a una clave KMS en su región.

Si su código se refiere a un nombre de alias como `alias/finance-key`, puede ejecutarlo en varias regiones. En cada región, utiliza una clave KMS diferente. Para obtener más información, consulte [Obtención de información sobre cómo utilizar alias en las aplicaciones](#).

Puede cambiar la clave KMS asociada a un alias

Puede utilizar la [UpdateAlias](#) operación para asociar un alias a una clave de KMS diferente. Por ejemplo, si el alias `finance-key` está asociado con la clave KMS `1234abcd-12ab-34cd-56ef-1234567890ab`, puede actualizarla para que esté asociada con la clave KMS `0987dcba-09fe-87dc-65ba-ab0987654321`.

Sin embargo, la clave de KMS actual y la nueva deben ser del mismo tipo (ambas simétricas o ambas asimétricas o ambas HMAC) y deben tener el mismo [uso de clave](#) (`ENCRYPT_DECRYPT` o `SIGN_VERIFY` o `GENERATE_VERIFY_MAC`). Esta restricción evita errores en el código que utiliza alias. Si debe asociar un alias a otro tipo de clave y ha mitigado los riesgos, puede eliminar el alias y volver a crearlo.

Algunas claves KMS no tienen alias

Al crear una clave KMS en la AWS KMS consola, debe asignarle un nuevo alias. Sin embargo, no se requiere un alias cuando se utiliza la [CreateKey](#) operación para crear una clave de KMS. Además, puede utilizar la [UpdateAlias](#) operación para cambiar la clave de KMS asociada a un alias y la [DeleteAlias](#) operación para eliminar un alias. Como resultado, algunas claves KMS pueden tener varios alias, y algunas podrían tener ninguno.

AWS crea alias en tu cuenta

AWS crea alias en tu cuenta para [Claves administradas por AWS](#). Estos alias tienen nombres del formulario `alias/aws/<service-name>`, como, por ejemplo, `alias/aws/s3`.

Algunos AWS alias no tienen clave KMS. Estos alias predefinidos suelen estar asociados a un Clave administrada de AWS cuando se empieza a utilizar el servicio.

Usar alias para identificar claves KMS

Puede usar un [nombre de alias](#) o un [ARN de alias](#) para identificar una clave de KMS en [las operaciones criptográficas](#), y [DescribeKey](#). [GetPublicKey](#) (Si la [clave KMS está en una Cuenta de AWS diferente](#), debe usar su [ARN de clave](#) o ARN de alias). Los alias no son identificadores válidos para las claves KMS en otras operaciones de AWS KMS. Para obtener información sobre los [identificadores clave](#) válidos para cada operación de la AWS KMS API, consulte las descripciones de los *KeyId* parámetros en la referencia de la AWS Key Management Service API.

No puede utilizar un nombre de alias ni un ARN de alias para [identificar una clave KMS en una política de IAM](#). Para controlar el acceso a una clave KMS en función de sus alias, usa las claves [ResourceAliasescondicionales kms: RequestAlias](#) o [kms:](#). Para obtener más información, consulte [ABAC para AWS KMS](#).

Control del acceso a alias

Al crear o cambiar un alias, afectará al alias y a su clave KMS asociada. Por lo tanto, las entidades principales que administran alias deben tener permiso para llamar a la operación de alias en el alias y en todas las claves KMS afectadas. Puede proporcionar estos permisos utilizando [políticas de claves](#), [políticas de IAM](#) y [concesiones](#).

Note

Tenga cuidado al dar permiso a las entidades principales para administrar etiquetas y alias. El cambio de etiqueta o alias puede permitir o denegar permiso a la clave administrada por el cliente. Para más detalles, consulte [ABAC para AWS KMS](#) y [Uso de alias para controlar el acceso a las claves KMS](#).

Para obtener información sobre cómo controlar el acceso a todas AWS KMS las operaciones, consulte [Referencia de permisos](#).

Los permisos para crear y administrar alias funcionan de la siguiente manera.

km: CreateAlias

Para crear un alias, la entidad principal necesita los siguientes permisos tanto para el alias como para la clave KMS asociada.

- `kms:CreateAlias` para el alias. Proporcione este permiso en una política de IAM adjunta a la entidad principal que tiene permiso para crear el alias.

En la siguiente declaración de política de ejemplo se especifica un alias en particular en un elemento `Resource`. Pero puede enumerar varios ARN de alias o especificar un patrón de alias, como `"test*"`. También puede especificar un valor `Resource` de `"*"` para permitir que la entidad principal cree cualquier alias en la cuenta y región. El permiso para crear un alias también se puede incluir en un permiso `kms:Create*` para todos los recursos de una cuenta y región.

```
{
  "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 la clave KMS. Este permiso debe proporcionarse en una política de claves o en una política de IAM que se delega desde la política de claves.

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

Puede utilizar claves de condición para limitar las claves KMS que puede asociar a un alias. Por ejemplo, puede usar la clave de KeySpec condición [kms:](#) para permitir que el director cree alias únicamente en claves KMS asimétricas. Para obtener una lista completa de las claves de condiciones que puede utilizar para limitar los permisos `kms:CreateAlias` en recursos de clave KMS, consulte [AWS KMS permisos](#).

kms: ListAliases

Para enumerar los alias de la cuenta y la región, la entidad principal debe tener el permiso `kms:ListAliases` en una política de IAM. Dado que esta política no está relacionada con ninguna clave o recurso de alias KMS en particular, el valor del elemento de recurso de la política [debe ser](#) `"*"`.

Por ejemplo, la siguiente declaración de política de IAM otorga el permiso a la entidad principal para enumerar todas las claves y alias KMS en la cuenta y la región.

JSON

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

km: UpdateAlias

Para cambiar la clave KMS asociada a un alias, la entidad principal necesita tres elementos de permiso: uno para el alias, uno para la clave KMS actual y otro para la nueva clave KMS.

Por ejemplo, suponga que desea cambiar el alias `test-key` de la clave KMS con el ID de clave `1234abcd-12ab-34cd-56ef-1234567890ab` a la clave KMS con el ID de clave `0987dcba-09fe-87dc-65ba-ab0987654321`. En ese caso, incluya declaraciones de política similares a las de los ejemplos de esta sección.

- `kms:UpdateAlias` para el alias. Puede proporcionar este permiso en una política de IAM que se asocia a la entidad principal. La siguiente política de IAM especifica un alias concreto. Pero puede enumerar varios ARN de alias o especificar un patrón de alias, como `"test*"`. También puede especificar un valor `Resource` de `"*"` para permitir que la entidad principal actualice cualquier alias en la cuenta y región.

```
{
  "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 la clave KMS que está actualmente asociada con el alias. Este permiso debe proporcionarse en una política de claves o en una política de IAM que se delega desde la política de claves.

```

{
  "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 la clave KMS que la operación asocia con el alias. Este permiso debe proporcionarse en una política de claves o en una política de IAM que se delega desde la política de claves.

```

{
  "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": "*"
}

```

Puede usar claves de condición para limitar una o ambas claves KMS en una operación `UpdateAlias`. Por ejemplo, puede usar una clave de `ResourceAliases` condición [kms:](#) para permitir que el director actualice los alias solo cuando la clave KMS de destino ya tenga un alias concreto. Para obtener una lista completa de las claves de condiciones que puede utilizar para limitar el permiso `kms:UpdateAlias` en un recurso de clave KMS, consulte [AWS KMS permisos](#).

kms: DeleteAlias

Para eliminar un alias, la entidad principal necesita permiso para el alias y para la clave KMS asociada.

Como siempre, debe tener precaución al dar permiso a las entidades principales para eliminar un recurso. Sin embargo, eliminar un alias no afecta a la clave KMS asociada. Aunque puede causar un error en una aplicación que se basa en el alias, si elimina un alias por error, puede volver a crearlo.

- `kms:DeleteAlias` para el alias. Proporcione este permiso en una política de IAM adjunta la entidad principal que tiene permiso para eliminar el alias.

En la siguiente declaración de política de ejemplo se especifica el alias de un elemento Resource. Pero puede enumerar varios ARN de alias o especificar un patrón de alias, como `"test*"`. También puede especificar un valor Resource de `"*"` para permitir que la entidad principal elimine cualquier alias en la cuenta y región.

```
{
  "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 la clave KMS asociada. Este permiso debe proporcionarse en una política de claves o en una política de IAM que se delega desde la política de claves.

```
{
  "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 los permisos de alias

Puede utilizar claves de condición para limitar los permisos de alias cuando el recurso es una clave KMS. Por ejemplo, la siguiente política de IAM permite operaciones de alias en claves KMS en una cuenta y región concretas. Sin embargo, utiliza la clave de KeyOrigin condición [kms:](#) para limitar aún más los permisos a las claves de KMS con material clave de AWS KMS.

Para obtener una lista completa de las claves de condiciones que puede utilizar para limitar el permiso de alias en un recurso de clave KMS, consulte [AWS KMS permisos](#).

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

No puede usar claves de condición en una declaración de política donde el recurso es un alias. Para limitar los alias que una entidad principal puede administrar, utilice el valor del elemento Resource de la declaración de política de IAM que controla el acceso al alias. Por ejemplo, las siguientes declaraciones de política permiten al director crear, actualizar o eliminar cualquier alias de la región Cuenta de AWS y, a menos que el alias comience porRestricted.

```
{  
  "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*"
}
```

Crear alias

Puede crear alias en la AWS KMS consola o mediante las operaciones de la AWS KMS API.

El alias debe ser una cadena de 1-256 caracteres. Solo puede contener caracteres alfanuméricos, barras (/), guiones bajos (_) y guiones (-). El nombre de alias de un [clave administrada por el cliente](#) no puede comenzar con `alias/aws/`. El prefijo `alias/aws/` se reserva para el uso de [Clave administrada de AWS](#).

Puede crear un alias para una nueva clave KMS o para una clave KMS existente. Puede agregar un alias para que se utilice una clave KMS concreta en un proyecto o aplicación.

También puede usar una AWS CloudFormation plantilla para crear un alias para una clave de KMS. Para obtener más información, consulte [AWS::KMS::Alias](#) en la Guía del usuario de AWS CloudFormation .

Uso de AWS KMS consola

Al [crear una clave de KMS](#) en la AWS KMS consola, debe crear un alias para la nueva clave de KMS. Para crear un alias para una clave KMS existente, utilice la pestaña Aliases (Alias) en la página de detalles de la clave KMS.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.

2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente. No puede administrar los alias de o. Claves administradas por AWS Claves propiedad de AWS
4. En la tabla, elija el ID de clave o el alias de la clave KMS. A continuación, en la página de detalles de la clave KMS, elija la pestaña Aliases (Alias).

Si una clave KMS tiene varios alias, la columna Aliases (Alias) de la tabla muestra un alias y un resumen de alias, como (+n más). Al elegir el resumen de alias, se le llevará directamente a la pestaña Aliases (Alias) en la página de detalles de la clave KMS.

5. En la pestaña Aliases (Alias), elija Create alias (Creación de alias). Introduzca un nombre de alias y elija Create alias (Creación de alias).

Important

No incluya información confidencial en este campo. Es posible que este campo se muestre en texto plano en CloudTrail los registros y otros resultados.

Note

No agregue el prefijo alias/. La consola lo agrega automáticamente. Si ingresa alias/ExampleAlias, el nombre de alias real será alias/alias/ExampleAlias.

Uso de AWS KMS API

Para crear un alias, utilice la [CreateAlias](#) operación. A diferencia del proceso de creación de claves de KMS en la consola, la [CreateKey](#) operación no crea un alias para una clave de KMS nueva.

Important

No incluya información confidencial en este campo. Este campo puede mostrarse en texto plano en CloudTrail los registros y otros resultados.

Puede utilizar la operación `CreateAlias` para crear un alias para una nueva clave KMS sin alias. También puede utilizar la operación `CreateAlias` para agregar un alias a cualquier clave KMS existente o para volver a crear un alias que se eliminó accidentalmente.

En las operaciones de la AWS KMS API, el nombre del alias debe empezar por `alias/` seguido de un nombre, por ejemplo. `alias/ExampleAlias`. El alias debe ser único en la cuenta y en la región de . Para buscar los nombres de alias que ya están en uso, utilice la [ListAliases](#) operación. El nombre del alias distingue entre mayúsculas y minúsculas.

La `TargetKeyId` puede ser cualquier [clave administrada por el cliente](#) en la misma Región de AWS. Para identificar la clave KMS, utilice su [ID de clave](#) o su [ARN de clave](#). No puede usar otro alias.

En el ejemplo siguiente, se crea el alias `example-key` y lo asocia con la clave KMS especificada. En estos ejemplos se utiliza el AWS Command Line Interface (AWS CLI). Para ver ejemplos en varios lenguajes de programación, consulte [Úselo CreateAlias con un AWS SDK o CLI](#).

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

`CreateAlias` no devuelve ningún resultado. Para ver el nuevo alias, utilice la operación `ListAliases`. Para obtener más información, consulte [Uso de AWS KMS API](#).

Búsqueda del nombre del alias y el ARN de alias para una clave de KMS

Los alias facilitan el reconocimiento de las claves de KMS en la AWS KMS consola. Puede ver los alias de una clave KMS en la AWS KMS consola o mediante esta operación. [ListAliases](#) La [DescribeKey](#) operación, que devuelve las propiedades de una clave KMS, no incluye los alias.

Los siguientes procedimientos muestran cómo ver e identificar los alias asociados a una clave de KMS mediante la AWS KMS consola y AWS KMS la API. Los ejemplos de AWS KMS API usan [AWS Command Line Interface \(AWS CLI\)](#), pero puedes usar cualquier lenguaje de programación compatible.

Uso de AWS KMS consola

La AWS KMS consola muestra los alias asociados a la clave KMS.

1. Abra la AWS KMS consola en. <https://console.aws.amazon.com/kms>
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. Si desea ver las claves de la cuenta que usted crea y administra, elija en el panel de navegación, Claves administradas por el cliente. Para ver las claves de la cuenta que se AWS crean y administran para usted, en el panel de navegación, seleccione las claves AWS administradas.
4. La columna Aliases (Alias) muestra el alias de cada clave KMS. Si una clave KMS no tiene un alias, aparece un guion (-) en la columna Aliases (Alias).

Si una clave KMS tiene varios alias, la columna Aliases (Alias) también tiene un resumen de alias, como (+n más). Por ejemplo, la siguiente clave KMS tiene dos alias, uno de los cuales es key-test.

Para buscar el nombre del alias y el ARN de alias de la clave KMS, utilice la pestaña Aliases (Alias).

- Para ir directamente a la pestaña Aliases (Alias), en la columna Aliases (Alias), elija el resumen de alias (+n más). Un resumen de alias sólo aparece si la clave KMS tiene más de un alias.
- O bien, elija el alias o el ID de clave de la clave KMS (que abre la página de detalles de la clave KMS) y, a continuación, elija la pestaña Aliases (Alias). Las pestañas se encuentran debajo de la sección General configuration (Configuración general).

Customer managed keys (16)			
Filter keys by aliases, key ID, or key type			Key actions ▼
Create key			
< 1 2 >			
⚙️			
☐	Aliases ▼	Key ID ▼	Status
☐	key-test (+1 more)	1234abcd-12ab-34cd-56ef-1234567890ab	Enabled
☐	-	1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d	Enabled

5. La pestaña Aliases (Alias) muestra el nombre del alias y el ARN del alias de todos los alias de una clave KMS. También puede crear y eliminar alias para la clave KMS en esta pestaña.

Key policy

Cryptographic configuration

Key material

Tags

Public key

Aliases

Aliases

Info

Delete

Create new alias

Q

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

Claves administradas por AWS

Puede usar el alias para reconocer un Clave administrada de AWS, como se muestra en esta página de ejemplo. Claves administradas por AWS Los alias de Claves administradas por AWS siempre tienen el formato: `aws/<service-name>`. Por ejemplo, el alias de Amazon aws/dynamodb DynamoDB es. Clave administrada de AWS

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

Uso de AWS KMS API

La [ListAliases](#) operación devuelve el nombre del alias y el ARN del alias de la cuenta y la región. El resultado incluye los alias para Claves administradas por AWS y para las claves administradas por el cliente. Los alias de Claves administradas por AWS deben tener el formato `aws/<service-name>`, como, por ejemplo, `aws/dynamodb`.

La respuesta también podría incluir los alias que no tienen el campo `TargetKeyId`. Se trata de alias predefinidos que se AWS han creado pero que aún no se han asociado a una clave de 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 obtener todos los alias que están asociados con una determinada clave KMS, utilice el parámetro `KeyId` de la operación `ListAliases`. El parámetro `KeyId` toma el [ID de clave](#) o el [ARN de clave](#) de la clave KMS.

En este ejemplo se obtienen todos los alias asociados con el `0987dcba-09fe-87dc-65ba-ab0987654321` de la clave KMS.

```

$ aws kms list-aliases --key-id 0987dcba-09fe-87dc-65ba-ab0987654321
{
  "Aliases": [
    {
      "AliasName": "alias/access-key",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "CreationDate": "2018-01-20T15:23:10.194000-07:00",
      "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
    },
    {
      "AliasName": "alias/finance-project",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "CreationDate": 1604958290.014,
      "LastUpdatedDate": 1604958290.014
    }
  ]
}

```

El parámetro `KeyId` no toma caracteres comodín, pero puede usar las características de su lenguaje de programación para filtrar la respuesta.

Por ejemplo, el siguiente AWS CLI comando solo obtiene los alias de. Claves administradas por AWS

```

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

```

El siguiente comando obtiene sólo el alias de access-key. El nombre del alias distingue entre mayúsculas y 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"
  }
]
```

Actualización de alias

Dado que un alias es un recurso independiente, puede cambiar la clave KMS asociada a un alias. Por ejemplo, si el test-key alias está asociado a una clave de KMS, puede utilizar la [UpdateAlias](#) operación para asociarlo a una clave de KMS diferente. Esta es una de las varias maneras de [girar manualmente una clave KMS](#) sin cambiar su material clave. También puede actualizar una clave KMS para que una aplicación que estaba utilizando una clave KMS para nuevos recursos utilice ahora una clave KMS diferente.

No puede actualizar un alias en la AWS KMS consola. Además, no puede utilizar UpdateAlias (o cualquier otra operación) para cambiar un nombre de alias. Para cambiar un nombre de alias, elimine el alias actual y, a continuación, cree un alias nuevo para la clave KMS.

Al actualizar un alias, la clave de KMS actual y la nueva clave de KMS deben ser del mismo tipo (ambas simétricas o asimétricas o HMAC). También deben tener el mismo uso de claves (ENCRYPT_DECRYPT o SIGN_VERIFY o GENERATE_VERIFY_MAC). Esta restricción evita errores criptográficos en el código que utiliza alias.

El siguiente ejemplo comienza con la [ListAliases](#) operación para mostrar que el test-key alias está asociado actualmente a la clave de KMS1234abcd-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
  }
]
```

A continuación, utiliza la operación `UpdateAlias` para cambiar la clave KMS que está asociada con el alias `test-key` a la clave KMS `0987dcba-09fe-87dc-65ba-ab0987654321`. No es necesario especificar la clave KMS asociada actualmente, solo la nueva («destino») clave KMS. El nombre del alias distingue entre mayúsculas y minúsculas.

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

Para comprobar que el alias se asocia ahora con la clave KMS de destino, utilice la operación `ListAliases` de nuevo. Este AWS CLI comando usa el `--query` parámetro para obtener solo el `test-key` alias. Los campos `TargetKeyId` y `LastUpdatedDate` se actualizan.

```
$ 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
  }
]
```

Eliminación de un alias

Puede eliminar un alias en la AWS KMS consola o mediante la [DeleteAlias](#) operación. Antes de eliminar un alias, asegúrese de que no esté en uso. Aunque eliminar un alias no afecta a la clave KMS asociada, puede crear problemas para cualquier aplicación que utilice el alias. Si elimina un alias por error, puede crear un nuevo alias con el mismo nombre y asociarlo a la misma clave KMS o a otra.

Si elimina una clave KMS, se eliminan todos los alias asociados a esa clave KMS.

Uso de AWS KMS consola

Para eliminar un alias de la AWS KMS consola, utilice la pestaña Alias de la página de detalles de la clave KMS. Puede eliminar varios alias para una clave KMS a la vez.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente. No puede administrar los alias de o. Claves administradas por AWS Claves propiedad de AWS
4. En la tabla, elija el ID de clave o el alias de la clave KMS. A continuación, en la página de detalles de la clave KMS, elija la pestaña Aliases (Alias).

Si una clave KMS tiene varios alias, la columna Aliases (Alias) de la tabla muestra un alias y un resumen de alias, como (+n más). Al elegir el resumen de alias, se le llevará directamente a la pestaña Aliases (Alias) en la página de detalles de la clave KMS.

5. En la pestaña Aliases (Alias), seleccione la casilla de verificación situada junto a los alias que desea eliminar. A continuación, elija Eliminar.

Uso de AWS KMS API

Para eliminar un alias, utilice la [DeleteAlias](#) operación. Esta operación elimina un alias a la vez. El nombre del alias distingue entre mayúsculas y minúsculas y debe estar precedido por el prefijo `alias/`.

Por ejemplo, el siguiente comando elimina el alias `test-key`. El comando no devuelve ningún resultado.

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

Para comprobar que se ha eliminado el alias, utilice la [ListAliases](#) operación. El comando siguiente utiliza el `--query` parámetro de AWS CLI para obtener únicamente el `test-key` alias. Los corchetes vacíos en la respuesta indican que la respuesta `ListAliases` no incluyó un alias `test-key`. Para eliminar los corchetes, utilice el parámetro y valor `--output text`.

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

[]

Uso de alias para controlar el acceso a las claves KMS

Puede controlar el acceso a las claves KMS basándose en los alias asociados a la clave KMS. Para ello, utilice las claves de ResourceAliases condición [kms: RequestAlias y kms:](#). Esta función forma parte de la AWS KMS compatibilidad con el [control de acceso basado en atributos](#) (ABAC).

La clave de condición `kms:RequestAlias` permite o deniega el acceso a una clave KMS basándose en el alias de una solicitud. La clave de condición `kms:ResourceAliases` permite o deniega el acceso a una clave KMS en función de los alias asociados a la clave KMS.

Estas características no le permiten identificar una clave KMS utilizando un alias en el elemento `resource` de una declaración de política. Cuando un alias es el valor de un elemento `resource`, la política se aplica al recurso de alias, no a ninguna clave KMS que pueda estar asociada con él.

Note

Puede que transcurran cinco minutos hasta que los cambios de etiqueta y alias afecten a la autorización de clave KMS. Los cambios recientes pueden ser visibles en las operaciones de API antes de que afecten a la autorización.

Cuando utilice alias para controlar el acceso a las claves KMS, tenga en cuenta lo siguiente:

- Utilice alias para reforzar las prácticas recomendadas de [acceso con privilegio mínimo](#). Dar a las entidades principales de IAM sólo los permisos que necesitan para las claves KMS que deben usar o administrar. Por ejemplo, utilice alias para identificar las claves KMS utilizadas en un proyecto. A continuación, dé permiso al equipo del proyecto para usar solo claves KMS con los alias del proyecto.
- Tenga cuidado al dar a las entidades principales los permisos `kms:CreateAlias`, `kms:UpdateAlias`, o `kms>DeleteAlias` que les permiten agregar, editar y eliminar alias. Cuando utiliza alias para controlar el acceso a las claves KMS, cambiar un alias puede dar permiso a las entidades principales para usar claves KMS que de otro modo no tenían permiso para usar. También puede denegar el acceso a las claves KMS que otras entidades principales requieren para realizar sus trabajos.
- Revisa los directores Cuenta de AWS que actualmente tienen permiso para administrar los alias y ajusta los permisos, si es necesario. Los administradores de claves que no tienen permiso para

cambiar políticas de claves o crear concesiones pueden controlar el acceso a claves KMS si tienen permiso para administrar alias.

Por ejemplo, la consola [default key policy for key administrators \(política de claves predeterminada para administradores de claves\)](#) incluye los permisos `kms:CreateAlias`, `kms>DeleteAlias`, y `kms:UpdateAlias`. Las políticas de IAM pueden dar permisos de alias para todas las claves KMS de su Cuenta de AWS. Por ejemplo, la política [AWSKeyManagementServicePowerUser](#) administrada permite a los directores crear, eliminar y enumerar los alias de todas las claves de KMS, pero no actualizarlos.

- Antes de establecer una política que dependa de un alias, revisa los alias de las claves de KMS de tu. Cuenta de AWS Asegúrese de que la política sólo se aplica a los alias que desea incluir. Usa [CloudTrail registros](#) y [CloudWatch alarmas](#) para avisarte de los cambios en el alias que puedan afectar al acceso a tus claves de KMS. Además, la [ListAliases](#) respuesta incluye la fecha de creación y la fecha de la última actualización de cada alias.
- Las condiciones de política de alias utilizan la coincidencia de patrones; no están vinculadas a una instancia concreta de un alias. Una política que utiliza claves de condición basadas en alias afecta a todos los alias nuevos y existentes que coincidan con el patrón. Si elimina y vuelve a crear un alias que coincida con una condición de política, la condición se aplica al nuevo alias, tal como lo hizo con el anterior.

La clave de condición `kms:RequestAlias` se basa en el alias especificado explícitamente en una solicitud de operación. La clave de condición `kms:ResourceAliases` depende de los alias asociados a una clave KMS, aunque no aparezcan en la solicitud.

km: RequestAlias

Permitir o denegar el acceso a una clave KMS basada en el alias que identifica la clave KMS en una solicitud. Puede utilizar la clave de RequestAlias condición [kms:](#) en una [política clave o en una política](#) de IAM. Se aplica a las operaciones que utilizan un alias para identificar una clave de KMS en una solicitud, es decir, [las operaciones criptográficas DescribeKey](#), y. [GetPublicKey](#) No es válido para operaciones de alias, como [CreateAlias](#)o [DeleteAlias](#).

Important

No se recomienda incluir una declaración de Deny política con `kms:RequestAlias` esta condición, ya que las personas que llaman pueden omitirla utilizando el [ID de clave, el ARN](#) de la clave o un alias diferente para identificar la clave de KMS en la solicitud. Para

denegar el acceso basándose en los alias asociados a una clave de KMS, utilice la clave de ResourceAliases condición [kms:](#) en su lugar.

En la clave de condición, especifique un [nombre de alias](#) o patrón de nombre de alias. No puede especificar un [ARN de alias](#).

Por ejemplo, la siguiente declaración de política permite a la entidad principal utilizar las operaciones especificadas en la clave KMS. El permiso solo será efectivo cuando la solicitud utiliza un alias que incluye alpha para identificar la clave 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*"
    }
  }
}
```

La siguiente solicitud de ejemplo de una entidad principal autorizada cumpliría la condición. Sin embargo, una solicitud que utilizó un [ID de clave](#), un [ARN de clave](#), o un alias diferente no cumpliría la condición, incluso si estos valores identificaban la misma clave KMS.

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

kms: ResourceAliases

Permitir o denegar el acceso a una clave KMS basándose en los alias asociados a la clave KMS, incluso si el alias no se utiliza en una solicitud. La clave de ResourceAliases condición [kms:](#) le

permite especificar un alias o un patrón de alias, por ejemplo `alias/test*`, para poder utilizarlos en una política de IAM para controlar el acceso a varias claves de KMS en la misma región. Es válida para cualquier AWS KMS operación que utilice una clave de KMS.

Por ejemplo, la siguiente política de IAM permite a las entidades principales llamar a las operaciones especificadas en las claves de KMS de dos en dos. Cuentas de AWS Sin embargo, el permiso sólo se aplica a las claves KMS asociadas con alias que comienzan con `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*"
        }
      }
    }
  ]
}
```

La condición `kms:ResourceAliases` es una condición del recurso, no de la solicitud. Como tal, una solicitud que no especifique el alias puede cumplir la condición.

La siguiente solicitud de ejemplo, que especifica un alias coincidente, satisface la condición.

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


Sin embargo, la solicitud de ejemplo siguiente también satisface la condición, siempre que la clave KMS especificada tenga un alias que comience con `restricted`, incluso si ese alias no se usa en la solicitud.

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

Obtención de información sobre cómo utilizar alias en las aplicaciones

Puede utilizar un alias para representar una clave KMS en el código de la aplicación. El `KeyId` parámetro en [las operaciones AWS KMS criptográficas](#) y [GetPublicKey](#) acepta un nombre de alias o un ARN de alias. [DescribeKey](#)

Por ejemplo, el siguiente comando `GenerateDataKey` utiliza un nombre de alias (`alias/finance`) para identificar una clave KMS. El nombre del alias es el valor del parámetro `KeyId`.

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

Si la clave de KMS está en otro Cuenta de AWS lugar, debe utilizar un ARN de clave o un alias ARN en estas operaciones. Cuando utilice un alias ARN, recuerde que el alias de una clave KMS se define en la cuenta que posee la clave KMS y puede diferir en cada región. Para obtener ayuda para encontrar el ARN de alias, consulte [Búsqueda del nombre del alias y el ARN de alias para una clave de KMS](#).

Por ejemplo, el siguiente comando `GenerateDataKey` utiliza una clave KMS que no está en la cuenta de la persona que llama. El alias `ExampleAlias` está asociado a la clave KMS en la cuenta y región especificadas.

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

Uno de los usos más potentes de los alias es en aplicaciones que se ejecutan en múltiples Regiones de AWS. Por ejemplo, puede tener una aplicación global que utiliza un RSA [clave KMS asimétrica](#) para la firma y la verificación.

- En EE.UU. Oeste (Oregón) (`us-west-2`), desea usar `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`.

- En Europa (Fráncfort) (eu-central-1), desea usar `arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321`
- En Asia Pacífico (Singapur) (ap-southeast-1), desea usar `arn:aws:kms:ap-southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d`.

Puede crear una versión diferente de su aplicación en cada región o utilizar un diccionario o una declaración `switch` para seleccionar la clave KMS correcta para cada región. Pero es mucho más fácil crear un alias con el mismo nombre de alias en cada región. El nombre del alias distingue entre mayúsculas y 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
```

A continuación, usa el alias en tu código. Cuando el código se ejecuta en cada región, el alias hará referencia a su clave KMS asociada en esa región. Por ejemplo, este código llama a la operación [Sign \(Firmar\)](#) con un nombre de alias.

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

Sin embargo, existe el riesgo de que el alias se elimine o actualice para que se asocie a una clave KMS diferente. En ese caso, los intentos de la aplicación de verificar firmas utilizando el nombre de alias fallarán y es posible que deba volver a crearlo o actualizarlo.

Para mitigar este riesgo, tenga cuidado al conceder permiso a las entidades principales para administrar los alias que utiliza en la aplicación. Para obtener más información, consulte [Control del acceso a alias](#).

Hay varias otras soluciones para aplicaciones que cifran datos en múltiples Regiones de AWS, incluido el [AWS Encryption SDK](#).

Busque alias en AWS CloudTrail registros

Puedes usar un alias para representar una operación AWS KMS key en una AWS KMS API. Cuando lo hace, el alias y la clave ARN de la clave KMS se registran en la entrada de AWS CloudTrail registro del evento. El alias aparece en el campo `requestParameters`. El ARN de clave aparece en el campo `resources`. Esto es cierto incluso cuando un AWS servicio utiliza una Clave administrada de AWS en su cuenta.

Por ejemplo, la siguiente [GenerateDataKey](#) solicitud usa el `project-key` alias para representar una clave de KMS.

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

Cuando esta solicitud se registra en el CloudTrail registro, la entrada del registro incluye el alias y la clave ARN de la clave de KMS real que se utilizó.

```
{
  "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 obtener más información sobre AWS KMS las operaciones de registro en CloudTrail los registros, consulte [Registro AWS KMS Llamadas a la API con AWS CloudTrail](#).

Etiquetas en AWS KMS

Una etiqueta es una etiqueta de metadatos opcional que puedes asignar (o AWS puedes asignar) a un AWS recurso. Cada etiqueta consta de una clave de etiqueta y a valor de etiqueta, que distinguen entre mayúsculas y minúsculas. El valor de la etiqueta puede ser una cadena vacía (nula). Cada etiqueta de un recurso debe tener una clave de etiqueta diferente, pero puedes añadir la misma etiqueta a varios AWS recursos. Cada recurso puede tener un máximo de 50 etiquetas creadas por el usuario.

No incluya información confidencial en la clave ni en el valor de la etiqueta. Muchas personas pueden acceder a las etiquetas Servicios de AWS, incluida la facturación.

En AWS KMS, puedes añadir etiquetas a una clave gestionada por el cliente al crear la clave de KMS y etiquetar o desetiquetar las claves de KMS existentes, a menos que estén [pendientes de ser eliminadas](#). No puedes etiquetar alias, Claves administradas por AWS almacenes de claves personalizadas ni claves de KMS en otros.Claves propiedad de AWS Cuentas de AWS Las etiquetas son opcionales, pero pueden ser muy útiles.

Por ejemplo, puede agregar una etiqueta "Project"="Alpha" para todas las claves KMS y los buckets de Amazon S3 que utilice para el proyecto Alpha.

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

Para obtener información general sobre las etiquetas, incluidos el formato y la sintaxis, consulte [AWS los recursos de etiquetado](#) en. Referencia general de Amazon Web Services

Las etiquetas lo ayudan a hacer lo siguiente:

- Identifique y organice sus AWS recursos. Muchos AWS servicios admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados. Por ejemplo, puede asignar la misma etiqueta a una clave de KMS y a un volumen o secreto de Amazon Elastic Block Store (Amazon EBS). AWS Secrets Manager También puede utilizar etiquetas para identificar claves KMS para la automatización.
- Realice un seguimiento de sus AWS costes. Cuando agrega etiquetas a sus AWS recursos, AWS genera un informe de asignación de costos con el uso y los costos agregados por etiquetas. Puede usar esta función para realizar un seguimiento de AWS KMS los costos de un proyecto, aplicación o centro de costos.

Para obtener más información sobre el uso de etiquetas para la asignación de costos, consulte [Uso de etiquetas de asignación de costos](#) en la Guía del usuario de AWS Billing . Para obtener información sobre las reglas que se aplican a las claves y los valores de las etiquetas, consulte [Restricciones de las etiquetas definidas por el usuario](#) en la Guía del usuario AWS Billing .

- Controle el acceso a sus AWS recursos. Permitir y denegar el acceso a las claves de KMS en función de sus etiquetas forma parte de la AWS KMS compatibilidad con el [control de acceso basado en atributos](#) (ABAC). Para obtener información sobre cómo controlar el acceso a AWS KMS keys en función de sus etiquetas, consulte. [Uso de etiquetas para controlar el acceso a las claves KMS](#) Para obtener más información general sobre el uso de etiquetas para controlar el acceso a AWS los recursos, consulte Control del [acceso a AWS los recursos mediante etiquetas de recursos](#) en la Guía del usuario de IAM.

AWS KMS escribe una entrada en el AWS CloudTrail registro cuando se utilizan las [ListResourceTags](#) operaciones [TagResource](#) [UntagResource](#), o.

Temas

- [Control del acceso a las etiquetas](#)
- [Inclusión de etiquetas a una clave de KMS](#)
- [Editar las etiquetas asociadas a una clave KMS](#)
- [Eliminación de las etiquetas asociadas a una clave de KMS](#)
- [Ver etiquetas asociadas a una clave KMS](#)
- [Uso de etiquetas para controlar el acceso a las claves KMS](#)

Control del acceso a las etiquetas

Para añadir, ver y eliminar etiquetas, ya sea en la AWS KMS consola o mediante la API, los directores necesitan permisos de etiquetado. Puede proporcionar estos permisos en una [política de claves](#). También puede proporcionarlos en las políticas de IAM (incluyendo [Políticas de punto de enlace de la VPC](#)), pero solo si [la política de claves lo permite](#). La política [AWSKeyManagementServicePowerUser](#) administrada permite a los directores etiquetar, desetiquetar y enumerar las etiquetas de todas las claves de KMS a las que puede acceder la cuenta.

También puede limitar estos permisos mediante claves de condición AWS globales para las etiquetas. En AWS KMS, estas condiciones pueden controlar el acceso a las operaciones de etiquetado, como [TagResource](#) y [UntagResource](#).

Note

Tenga cuidado al dar permiso a las entidades principales para administrar etiquetas y alias. El cambio de etiqueta o alias puede permitir o denegar permiso a la clave administrada por el cliente. Para más detalles, consulte [ABAC para AWS KMS](#) y [Uso de etiquetas para controlar el acceso a las claves KMS](#).

Para obtener más información y políticas de ejemplo, consulte [Control del acceso en función de las claves de etiqueta](#) en la Guía del usuario de IAM.

Los permisos para crear y administrar etiquetas funcionan de la siguiente manera.

km: TagResource

Permite a las entidades principales agregar o editar etiquetas. Para agregar etiquetas al crear una clave KMS, la entidad principal debe tener permiso en una política de IAM que no esté restringida a determinadas claves KMS.

km: ListResourceTags

Permite a las entidades principales ver etiquetas en claves KMS.

km: UntagResource

Permite a las entidades principales eliminar etiquetas de las claves KMS.

Permisos de etiquetas en políticas

Puede proporcionar permisos de etiquetas en una política de claves o una política de IAM. Por ejemplo, la siguiente política de claves de ejemplo ofrece permiso de etiquetar a los usuarios seleccionados en la clave KMS. Da permiso a todos los usuarios que pueden asumir los roles de administrador o desarrollador de ejemplo para ver 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 permiso de etiquetado de entidades principales en varias claves KMS, puede usar una política de IAM. Para que esta política sea efectiva, la política de claves de cada clave KMS debe permitir a la cuenta utilizar políticas de IAM para controlar el acceso a clave KMS.

Por ejemplo, la siguiente política de IAM permite a las entidades principales crear claves KMS. También les permite crear y administrar etiquetas en todas las claves KMS de la cuenta especificada. Esta combinación permite a los directores utilizar el parámetro [Etiquetas](#) de la [CreateKey](#) operación para añadir etiquetas a una clave de KMS mientras la crean.

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 los permisos de etiqueta

Puede limitar los permisos de etiquetado mediante [condiciones de política](#). Las siguientes condiciones de política se pueden aplicar a los permisos `kms:TagResource` y `kms:UntagResource`. Por ejemplo, puede utilizar la condición `aws:RequestTag/tag-key` para permitir que una entidad principal agregue solo etiquetas particulares, o impedir que una entidad principal agregue etiquetas con claves de etiqueta concretas. También puede utilizar la condición `kms:KeyOrigin` para evitar que las entidades principales etiqueten o desetiqueten claves KMS con [material de claves importado](#).

- [AWS: RequestTag](#)
- [aws:ResourceTag/tag-key](#) (solo políticas de IAM)
- [AWS: TagKeys](#)
- [km: CallerAccount](#)

- [km: KeySpec](#)
- [km: KeyUsage](#)
- [km: KeyOrigin](#)
- [km: ViaService](#)

Como práctica recomendada cuando utilice etiquetas para controlar el acceso a claves KMS, utilice la clave de condición `aws:RequestTag/tag-key` o `aws:TagKeys` para determinar qué etiquetas (o claves de etiqueta) están permitidas.

Por ejemplo, la siguiente política IAM es similar a la anterior. Sin embargo, esta política permite a las entidades principales crear etiquetas (`TagResource`) y eliminar etiquetas `UntagResource` solo para etiquetas con una clave de etiqueta `Project`.

Como `TagResource` las `UntagResource` solicitudes pueden incluir varias etiquetas, debe especificar un operador `ForAllValues` o `ForAnyValue` configurarlo con la `TagKeys` condición [aws:](#). El operador `ForAnyValue` requiere que al menos una de las claves de etiqueta de la solicitud coincida con una de las claves de etiqueta de la política. El operador `ForAllValues` requiere que todas las claves de etiqueta de la solicitud coincidan con una de las claves de etiqueta de la política. El `ForAllValues` operador también devuelve el `true` mensaje si no hay etiquetas en la solicitud, pero `TagResource` no lo `UntagResource` hace si no se especifica ninguna etiqueta. Para obtener información detallada sobre los operadores de conjunto, consulte [Usar varias claves y valores](#) en la Guía del usuario de 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"}
    }
  }
]
```

Inclusión de etiquetas a una clave de KMS

Las etiquetas ayudan a identificar y organizar AWS los recursos. Puede agregar etiquetas a una clave administrada por el cliente cuando [crea la clave KMS](#) o agregar etiquetas a claves KMS existentes. No puede etiquetar Claves administradas por AWS.

Los siguientes procedimientos muestran cómo añadir etiquetas a las claves administradas por el cliente mediante la AWS KMS consola y la AWS KMS API. Los ejemplos de AWS KMS API usan el [AWS Command Line Interface \(AWS CLI\)](#), pero puedes usar cualquier lenguaje de programación compatible.

Temas

- [Agregar etiquetas al crear una clave KMS](#)
- [Agregar etiquetas a claves KMS existentes](#)

Agregar etiquetas al crear una clave KMS

Puede agregar etiquetas a una clave de KMS al crear la clave mediante la AWS KMS consola o la [CreateKey](#) operación. Para agregar etiquetas al crear una clave KMS debe tener el permiso `kms:TagResource` en una política de IAM, además de los permisos necesarios para crear claves KMS. Como mínimo, el permiso debe cubrir todas las claves KMS de la cuenta y la región. Para obtener más información, consulte [Control del acceso a las etiquetas](#).

Uso de la AWS KMS consola

Para agregar etiquetas al crear una clave KMS en la consola debe tener los permisos necesarios para ver claves KMS en la consola, además de los permisos necesarios para etiquetar y crear claves KMS. Como mínimo, el permiso debe cubrir todas las claves KMS de la cuenta y la región.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente. (No puede administrar las etiquetas de una Clave administrada de AWS)
4. Elija el tipo de clave, a continuación, elija Next (Siguiente).
5. Escriba un alias y una descripción opcional.
6. Introduzca una clave de etiqueta y un valor de etiqueta opcional. Para agregar otras etiquetas, elija Add tag (Agregar etiqueta). Para quitar una etiqueta, seleccione Remove (Eliminar). Cuando termine de etiquetar su nueva clave KMS, elija Next (Siguiente).
7. Termine de crear su clave KMS

Uso de la API AWS KMS

Para especificar etiquetas al crear claves mediante la [CreateKey](#) operación, utilice el Tags parámetro de la operación.

El valor del parámetro Tags de CreateKey es una colección de pares de claves y valores de etiqueta que distinguen mayúsculas y minúsculas. Cada etiqueta de una clave KMS debe tener un nombre de etiqueta diferente. El valor de etiqueta puede ser una cadena vacía o nula.

Por ejemplo, el siguiente AWS CLI comando crea una clave KMS de cifrado simétrico con una Project:Alpha etiqueta. Cuando especifique más de un par de clave-valor, utilice un espacio para separar cada par.

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

Cuando este comando se ejecuta correctamente, devuelve un objeto KeyMetadata con información sobre la nueva clave KMS. Sin embargo, KeyMetadata no incluye etiquetas. Para obtener las etiquetas, utilice la [ListResourceTags](#) operación.

Agregar etiquetas a claves KMS existentes

Puede añadir etiquetas a las claves de KMS administradas por el cliente existentes en la AWS KMS consola o mediante la [TagResource](#) operación. Para agregar etiquetas necesita permiso de etiquetado en la clave KMS. Puede obtener este permiso de la política de clave para la clave KMS o, si la política de clave lo permite, de una política de IAM que incluya la clave KMS.

Uso de la AWS KMS consola

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente. (No puede administrar las etiquetas de una Clave administrada de AWS)
4. Puede utilizar el filtro de tabla para mostrar sólo claves KMS con etiquetas concretas. Para obtener más información, consulte [Ver etiquetas mediante la consola de AWS KMS](#).
5. Seleccione la casilla de verificación situada junto al alias de una clave KMS.
6. Elija Key actions, Add or edit tags.
7. En la página de detalles de la clave KMS, seleccione la pestaña Tags (Etiquetas).
 - Para crear la primera etiqueta, elija Create tag (Crear etiqueta), escriba el nombre (obligatorio) y el valor (opcional) de la etiqueta y, por último, elija Save (Guardar).

Si deja el valor de etiqueta en blanco, el valor de etiqueta real es una cadena nula o vacía.

 - Para agregar una etiqueta, elija Edit (Editar), Add tag (Agregar etiqueta), escriba el nombre y el valor de la etiqueta y, por último, elija Save (Guardar).
8. Para guardar los cambios, elija Guardar cambios.

Uso de la API AWS KMS

La [TagResource](#) operación agrega una o más etiquetas a una clave de KMS. No puede usar esta operación para agregar o editar etiquetas en otra Cuenta de AWS. También puede utilizar la TagResource operación para editar las etiquetas existentes. Para obtener más información, consulte [the section called “Editar etiquetas”](#).

Para agregar una etiqueta, especifique una clave de etiqueta nueva y un valor de la etiqueta. Cada etiqueta de una clave KMS debe tener una clave de etiqueta distinta. El valor de etiqueta puede ser una cadena vacía o nula.

Por ejemplo, el siguiente comando agrega las etiquetas **Purpose** y **Department** a una clave KMS de ejemplo.

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

Si este comando se realiza correctamente, no devuelve ningún resultado. Para ver las etiquetas de una clave KMS, utilice la [ListResourceTags](#) operación.

Editar las etiquetas asociadas a una clave KMS

Las etiquetas ayudan a identificar y organizar AWS los recursos. Puede editar las etiquetas asociadas a las claves de KMS administradas por el cliente en la AWS KMS consola o mediante esta [TagResource](#) operación. No puede editar las etiquetas de un Clave administrada de AWS.

Los siguientes procedimientos muestran cómo editar las etiquetas asociadas a una clave KMS. En estos ejemplos de API de AWS KMS , se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Uso de la AWS KMS consola

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente. (No puede editar las etiquetas de un) Clave administrada de AWS
4. Puede utilizar el filtro de tabla para mostrar sólo claves KMS con etiquetas concretas. Para obtener más información, consulte [Ver etiquetas mediante la consola de AWS KMS](#).
5. Seleccione la casilla de verificación situada junto al alias de una clave KMS.
6. Elija Key actions, Add or edit tags.
7. En la página de detalles de la clave KMS, seleccione la pestaña Tags (Etiquetas).

- Para modificar el nombre o el valor de una etiqueta, elija Edit (Editar), aplique los cambios y elija Save (Guardar).

8. Para guardar los cambios, elija Guardar cambios.

Uso de la AWS KMS API

La [TagResource](#) operación agrega una o más etiquetas a una clave administrada por el cliente; Sin embargo, también pueden utilizar TagResource para cambiar los valores de una etiqueta existente. No puede usar esta operación para agregar o editar etiquetas en una Cuenta de AWS diferente.

Para editar una etiqueta, especifique una clave de etiqueta existente y un nuevo valor de etiqueta. Cada etiqueta de una clave KMS debe tener una clave de etiqueta distinta. El valor de etiqueta puede ser una cadena vacía o nula.

Por ejemplo, este comando cambia el valor de la etiqueta Purpose de Pretest a Test.

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

Eliminación de las etiquetas asociadas a una clave de KMS

Las etiquetas ayudan a identificar y organizar sus AWS recursos. Puede eliminar las etiquetas asociadas a las claves de KMS administradas por el cliente en la AWS KMS consola o mediante esta [UntagResource](#) operación. No puede editar ni eliminar las etiquetas de un Clave administrada de AWS.

En los siguientes procedimientos se muestra cómo eliminar etiquetas de una clave KMS. En estos ejemplos de API de AWS KMS , se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Uso de la AWS KMS consola

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.

3. En el panel de navegación, elija Claves administradas por el cliente. (No puede administrar las etiquetas de una Clave administrada de AWS)
4. Puede utilizar el filtro de tabla para mostrar sólo claves KMS con etiquetas concretas. Para obtener más información, consulte [Ver etiquetas mediante la consola de AWS KMS](#).
5. Seleccione la casilla de verificación situada junto al alias de una clave KMS.
6. Elija Key actions, Add or edit tags.
7. En la página de detalles de la clave KMS, seleccione la pestaña Tags (Etiquetas).
 - Para eliminar una etiqueta, elija Edit (Editar). En la fila de la etiqueta, elija Remove (Eliminar) y, luego, Save (Guardar).
8. Para guardar los cambios, elija Guardar cambios.

Uso de la API AWS KMS

La [UntagResource](#) operación elimina las etiquetas de una clave de KMS. Para identificar las etiquetas que desea eliminar, especifique las claves de etiqueta. No puede usar esta operación para eliminar etiquetas de claves KMS una Cuenta de AWS diferente.

Cuando tiene éxito, la operación `UntagResource` no devuelve ningún resultado. Además, si la clave de etiqueta especificada no se encuentra en la clave KMS, no arroja una excepción ni devuelve una respuesta. Para confirmar que la operación ha funcionado, utilice la [ListResourceTags](#) operación.

Por ejemplo, este comando elimina la etiqueta **Purpose** y todos sus valores de la clave KMS especificada.

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

Ver etiquetas asociadas a una clave KMS

Las etiquetas ayudan a identificar y organizar AWS los recursos. Puede ver las etiquetas asociadas a las claves de KMS administradas por el cliente en la AWS KMS consola o mediante la [ListResourceTags](#) operación.

Los siguientes procedimientos muestran cómo encontrar las etiquetas asociadas a una clave KMS específica. Los ejemplos de AWS KMS API usan [AWS Command Line Interface \(AWS CLI\)](#), pero puedes usar cualquier lenguaje de programación compatible.

Uso de la AWS KMS consola

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Claves administradas por el cliente. (No puede administrar las etiquetas de una Clave administrada de AWS)
4. Puede utilizar el filtro de tabla para mostrar sólo claves KMS con etiquetas concretas.

Para mostrar solo las claves KMS con una etiqueta concreta, elija el cuadro de filtro, elija la clave de etiqueta y, a continuación, elija entre los valores de etiqueta reales. También puede introducir una parte o el valor de la etiqueta completo.

La tabla resultante muestra todas las claves KMS con la etiqueta elegida. Sin embargo, no muestra la etiqueta. Para ver la etiqueta, elija el ID de clave o alias de la clave KMS y, en su página de detalles, elija la pestaña Tags (Etiquetas). Las pestañas aparecen debajo de la sección General configuration (Configuración general).

Este filtro requiere la clave y el valor de la etiqueta. No encontrará claves KMS escribiendo solo la clave de etiqueta o solo su valor. Para filtrar las etiquetas por la totalidad o parte de la clave o el valor de la etiqueta, utilice la [ListResourceTags](#) operación para obtener las claves KMS etiquetadas y, a continuación, utilice las funciones de filtrado del lenguaje de programación.

5. Seleccione la casilla de verificación situada junto al alias de una clave KMS.
6. Elija Key actions, Add or edit tags.
7. En la página de detalles de la clave KMS, seleccione la pestaña Tags (Etiquetas).

Uso de la AWS KMS API

La [ListResourceTags](#) operación obtiene las etiquetas de una clave KMS. El parámetro KeyId es obligatorio. No puede usar esta operación para ver las etiquetas de claves KMS en una Cuenta de AWS diferente.

Por ejemplo, el comando siguiente obtiene las etiquetas para una clave KMS de ejemplo.

```
$ 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"
  }
]
```

Uso de etiquetas para controlar el acceso a las claves KMS

Puede controlar el acceso a en AWS KMS keys función de las etiquetas de la clave KMS. Por ejemplo, puede escribir una política de IAM que permita a las entidades principales habilitar y desactivar solo las claves KMS que tienen una etiqueta concreta. O bien, puede utilizar una política de IAM para evitar que las principales entidades utilicen claves KMS en operaciones criptográficas, a menos que la clave KMS tenga una etiqueta concreta.

Esta función forma parte de la AWS KMS compatibilidad con el [control de acceso basado en atributos](#) (ABAC). Para obtener información sobre el uso de etiquetas para controlar el acceso a AWS los recursos, consulte [¿Para qué sirve el ABAC? AWS y Cómo controlar el acceso a AWS los recursos mediante etiquetas de recursos](#) en la Guía del usuario de IAM. Para obtener ayuda para resolver problemas de acceso relacionados con ABAC, consulte [Solución de problemas de ABAC para AWS KMS](#).

Note

Puede que transcurran cinco minutos hasta que los cambios de etiqueta y alias afecten a la autorización de clave KMS. Los cambios recientes pueden ser visibles en las operaciones de API antes de que afecten a la autorización.

AWS KMS admite la clave [contextual de condición global aws:ResourceTag/tag-key](#), que permite controlar el acceso a las claves de KMS en función de las etiquetas de la clave de KMS. Dado que varias claves KMS pueden tener la misma etiqueta, esta función le permite aplicar el permiso a un conjunto seleccionado de claves KMS. También puede cambiar fácilmente las claves KMS del conjunto cambiando sus etiquetas.

En AWS KMS, la clave de `aws:ResourceTag/tag-key` condición solo se admite en las políticas de IAM. No se admite en las políticas clave, que solo se aplican a una clave de KMS, ni en las operaciones que no utilizan una clave de KMS concreta, como las [ListAliases](#) operaciones [ListKeys](#).

Controlar el acceso con etiquetas proporciona una forma sencilla, escalable y flexible de administrar los permisos. Sin embargo, si no está diseñado y administrado correctamente, puede permitir o denegar el acceso a sus claves KMS inadvertidamente. Si utiliza etiquetas para controlar el acceso, tenga en cuenta las siguientes prácticas.

- Utilice etiquetas para reforzar la práctica recomendada de [acceso menos privilegiado](#). Proporcione a las entidades principales de IAM solo los permisos que necesitan y únicamente en las claves KMS que deben usar o administrar. Por ejemplo, utilice etiquetas para etiquetar las claves KMS utilizadas en un proyecto. A continuación, dé permiso al equipo del proyecto para usar solo claves KMS con la etiqueta de proyecto.
- Tenga cuidado al dar a las principales entidades los permisos `kms:TagResource` y `kms:UntagResource` que les permiten agregar, editar y eliminar etiquetas. Cuando utiliza etiquetas para controlar el acceso a las claves KMS, cambiar una etiqueta puede dar permiso a las principales entidades para usar claves KMS que de otro modo no tenían permiso para usar. También puede denegar el acceso a las claves KMS que otras entidades principales requieren para realizar sus trabajos. Los administradores de claves que no tienen permiso para cambiar políticas de claves o crear concesiones pueden controlar el acceso a claves KMS si tienen permiso para administrar etiquetas.

Siempre que sea posible, utilice una condición de política, como `aws:RequestTag/tag-key` o `aws:TagKeys` para [limitar los permisos de etiquetado de una entidad](#) a determinadas etiquetas o patrones de etiquetas en determinadas claves KMS.

- Revisa las entidades principales Cuenta de AWS que actualmente tienen permisos para etiquetar y desetiquetar y ajústalas si es necesario. Por ejemplo, la consola [Política de claves predeterminada para administradores de claves](#) incluye el permiso `kms:TagResource` y `kms:UntagResource` en esa clave KMS. Las políticas de IAM pueden habilitar permisos de etiqueta y desetiqueta en todas las claves KMS. Por ejemplo, la política

[AWSKeyManagementServicePowerUser](#) administrada permite a los directores etiquetar, desetiquetar y enumerar las etiquetas de todas las claves de KMS.

- Antes de establecer una política que dependa de una etiqueta, revisa las etiquetas de las claves KMS de tu. Cuenta de AWS Asegúrese de que su política solo se aplique a las etiquetas que desea incluir. Usa [CloudTrail registros](#) y [CloudWatch alarmas](#) para avisarte de los cambios en las etiquetas que puedan afectar al acceso a tus claves de KMS.
- Las condiciones de política basadas en etiquetas utilizan la coincidencia de patrones; no están vinculadas a una instancia concreta de una etiqueta. Una política que utiliza claves de condición basadas en etiquetas afecta a todas las etiquetas nuevas y existentes que coincidan con el patrón. Si elimina y vuelve a crear una etiqueta que coincida con una condición de política, la condición se aplica a la nueva etiqueta, igual que a la anterior.

Por ejemplo, tomemos el siguiente ejemplo de política de IAM. Permite a los directores realizar llamadas a las operaciones [GenerateDataKeyWithoutPlaintext](#) y [descifrar](#) únicamente en las claves de KMS de su cuenta que pertenezcan a la región de Asia Pacífico (Singapur) y tengan una etiqueta. "Project"="Alpha" Puede adjuntar esta política a roles del ejemplo de proyecto Alpha.

JSON

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

La siguiente política de IAM de ejemplo permite a la entidad principal utilizar la clave KMS en la cuenta para operaciones criptográficas. Pero prohíbe a las principales entidades usar estas operaciones criptográficas en claves KMS con una etiqueta "Type"="Reserved" o sin 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"
    }
  }
}
```

Almacenes de claves

Un almacén de claves es una ubicación segura para almacenar y utilizar claves criptográficas. El almacén de claves predeterminado AWS KMS también admite métodos para generar y administrar las claves que almacena. De forma predeterminada, el material de claves criptográficas para el AWS KMS keys que se crea AWS KMS se genera y se protege mediante módulos de seguridad de hardware (HSMs) que forman parte del programa de validación de módulos [criptográficos FIPS 140-3](#). El material clave de sus claves KMS nunca se deja sin cifrar. HSMs

AWS KMS admite varios tipos de almacenes de claves para proteger el material de claves cuando se utiliza AWS KMS para crear y administrar las claves de cifrado. Todas las opciones de almacenamiento de claves que ofrece AWS KMS se validan continuamente según la norma FIPS 140-3 en el nivel de seguridad 3 y están diseñadas para impedir que cualquier persona, incluidos AWS los operadores, acceda a sus claves en texto simple o las utilice sin su permiso.

AWS KMS almacén de claves estándar

De forma predeterminada, se crea una clave KMS mediante el AWS KMS HSM estándar. Este tipo de HSM puede considerarse como una flota multiusuario HSMs que permite disponer del almacén de claves más escalable, económico y fácil de gestionar desde tu perspectiva. Si va a crear una clave KMS para utilizarla en uno o varios Servicios de AWS servidores para que el servicio pueda cifrar sus datos por usted, creará una clave simétrica. Si utiliza una clave KMS para el diseño de su propia aplicación, puede optar por crear una clave de cifrado simétrica, una clave asimétrica o una clave HMAC.

En la opción de almacenamiento de claves estándar, AWS KMS crea la clave y, a continuación, la cifra con las claves que el servicio administra internamente. A continuación, se almacenan varias copias de las versiones cifradas de sus claves en sistemas diseñados para ser duraderos. Al generar y proteger el material de claves en el tipo de almacén de claves estándar, podrá aprovechar al máximo la escalabilidad, la disponibilidad y la durabilidad de los almacenes de AWS claves AWS KMS con la carga operativa y el coste más bajos.

AWS KMS almacén de llaves estándar con material clave importado

En lugar de solicitar AWS KMS que se generen y almacenen las únicas copias de una clave determinada, puede optar por importar el material clave AWS KMS, lo que le permite generar su

propia clave de cifrado simétrica de 256 bits, una clave RSA o de curva elíptica (ECC) o una clave de código de autenticación de mensajes basado en hash (HMAC) y aplicarla a un identificador de clave KMS (keyID). A veces, esto se denomina traiga su propia clave (BYOK). El material clave importado de su sistema de administración de claves local debe protegerse mediante una clave pública emitida por AWS KMS, un algoritmo de empaquetado criptográfico compatible y un token de importación basado en el tiempo proporcionado por AWS KMS. Este proceso verifica que un HSM de AWS KMS solo pueda descifrar la clave importada y cifrada una vez que haya salido del entorno.

El material clave importado puede ser útil si tiene requisitos específicos en relación con el sistema que genera las claves o si desea una copia de la clave AWS como copia de seguridad. Sigue siendo responsable de la disponibilidad y la durabilidad generales del material de claves. Si bien AWS KMS tiene una copia de tu clave importada y estará disponible en cualquier momento mientras la necesites, las claves importadas ofrecen una API especial para eliminarla: `DeleteImportedKeyMaterial`. Esta API eliminará inmediatamente todas las copias del material de claves AWS KMS importado que contenga, sin opción de AWS recuperar la clave. Además, puede establecer un período de caducidad para una clave importada, después del cual la clave quedará inutilizable. Para que la clave vuelva a ser útil AWS KMS, tendrás que volver a importar el material clave y asignarlo al mismo KeyID. Esta acción de eliminación de las claves importadas es diferente a la de las claves estándar que se AWS KMS generan y almacenan por ti mismo. En el caso habitual, el proceso de eliminación de claves tiene un período de espera obligatorio en el que primero se bloquea el uso de una clave cuya eliminación está programada. Esta acción te permite ver los errores de acceso denegado en los registros de cualquier aplicación o AWS servicio que pueda necesitar esa clave para acceder a los datos. Si ve dichas solicitudes de acceso, puede optar por cancelar la eliminación programada y volver a habilitar la clave. Tras un período de espera configurable (entre 7 y 30 días) y solo entonces, KMS eliminará realmente el material de claves, el KeyID y todos los metadatos asociados a la clave. Para obtener más información sobre la disponibilidad y la durabilidad, consulte [Protección del material clave importado](#) en la Guía para desarrolladores de AWS KMS .

Hay que tener en cuenta que el material de claves importado presenta algunas limitaciones adicionales. Como AWS KMS no puede generar material de claves nuevo, no hay forma de configurar la rotación automática de las claves importadas. Deberá crear una nueva clave de KMS con un nuevo keyID y, a continuación, importar el nuevo material de claves para lograr una rotación efectiva. Además, los textos cifrados creados AWS KMS con una clave simétrica importada no se pueden descifrar fácilmente utilizando una copia local de la clave externa a. AWS Esto se debe a que el formato de cifrado autenticado que utiliza AWS KMS agrega metadatos adicionales al texto cifrado para garantizar, durante la operación de descifrado, que el texto cifrado se creó con la clave KMS

prevista en una operación de cifrado anterior. La mayoría de los sistemas criptográficos externos no sabrán cómo analizar estos metadatos para acceder al texto cifrado sin procesar y poder utilizar su copia de una clave simétrica. Los textos cifrados creados AWS KMS con claves asimétricas importadas (por ejemplo, RSA o ECC) se pueden utilizar fuera de la parte correspondiente (pública o privada) de la clave, ya que no se añaden metadatos adicionales AWS KMS al texto cifrado.

AWS KMS almacenes de claves personalizados

Sin embargo, si necesita un control aún mayor del HSMs, puede crear un almacén de claves personalizado.

Un almacén de claves personalizado es un almacén de claves interno AWS KMS que está respaldado por un administrador de claves externo AWS KMS, del que usted es propietario y administra. Los almacenes de claves personalizados combinan la práctica y completa interfaz de administración de claves AWS KMS con la capacidad de poseer y controlar el material clave y las operaciones criptográficas. Al utilizar una clave de KMS en un almacén de claves personalizado, el administrador de claves realiza las operaciones criptográficas usando sus claves criptográficas. Como resultado, usted asume una mayor responsabilidad por la disponibilidad y durabilidad de las claves criptográficas y por el funcionamiento de las HSMs.

Ser propietario de la suya HSMs puede resultar útil para cumplir con ciertos requisitos normativos que aún no permiten que los servicios web multiusuario, como el almacén de claves KMS estándar, almacenen sus claves criptográficas. Los almacenes de claves personalizados no son más seguros que los almacenes de claves de KMS que se utilizan de forma AWS gestionada HSMs, pero tienen implicaciones de gestión y costes diferentes (y superiores). Como resultado, usted asume una mayor responsabilidad por la disponibilidad y durabilidad de las claves criptográficas y por el funcionamiento de las HSMs. Independientemente de si utiliza el almacén de claves estándar AWS KMS HSMs o uno personalizado, el servicio está diseñado para que nadie, ni siquiera AWS los empleados, pueda recuperar sus claves en texto simple o utilizarlas sin su permiso. AWS KMS admite dos tipos de almacenes de claves personalizados, los almacenes de AWS CloudHSM claves y los almacenes de claves externos.

Características no admitidas

AWS KMS no admite las siguientes funciones en los almacenes de claves personalizados.

- [Claves de KMS asimétricas](#)
- [Claves KMS HMAC](#)

- [Claves KMS con material de claves importado](#)
- [Rotación automática de claves](#)
- [Claves de varias regiones](#)

AWS CloudHSM almacén de claves

Puede crear una clave de KMS en un [AWS CloudHSM](#) almacén de claves, donde se generan, almacenan y utilizan las claves de usuario raíz en un AWS CloudHSM clúster que usted posee y administra. Las solicitudes AWS KMS para usar una clave para alguna operación criptográfica se reenvían al AWS CloudHSM clúster para realizar la operación. Si bien un AWS CloudHSM clúster lo aloja AWS, es una solución de un solo inquilino que usted administra y opera directamente. Usted es el propietario de gran parte de la disponibilidad y el rendimiento de las claves de KMS de un AWS CloudHSM clúster. Para ver si un almacén de claves AWS CloudHSM personalizado es adecuado para sus necesidades, consulte [¿Son los almacenes de claves AWS KMS personalizados adecuados para usted?](#) en el blog AWS de seguridad.

Almacén de claves externo

Puede configurarlo AWS KMS para utilizar un almacén de claves externo (XKS), donde las claves de los usuarios raíz se generan, almacenan y utilizan en un sistema de gestión de claves ajeno al Nube de AWS. Las solicitudes a AWS KMS para utilizar una clave para alguna operación criptográfica se reenvían al sistema alojado externamente para realizar la operación. En concreto, las solicitudes se reenvían a un proxy XKS de su red, que luego las reenvía al sistema criptográfico que usted utilice. El proxy de XKS es una especificación de código abierto con la que cualquiera puede integrarse. Muchos proveedores comerciales de administración de claves admiten la especificación XKS Proxy. Como un almacén de claves externo está alojado por usted o por un tercero, usted es el responsable de toda la disponibilidad, durabilidad y rendimiento de las claves del sistema. Para comprobar si un almacén de claves externo es adecuado para sus necesidades, lea [Anunciar el almacén de claves AWS KMS externo \(XKS\)](#) en el blog de AWS noticias.

AWS CloudHSM tiendas clave

Un almacén de AWS CloudHSM claves es un [almacén de claves personalizado](#) respaldado por un [AWS CloudHSM clúster](#). Al crear uno AWS KMS key en un almacén de claves personalizado, AWS KMS genera y almacena material de claves no extraíble para la clave de KMS en un AWS CloudHSM clúster que es de su propiedad y que administra. Cuando utilizas una clave KMS en un

almacén de claves personalizado, las [operaciones criptográficas](#) se realizan HSMs en el clúster. Esta función combina la comodidad y la amplia integración AWS KMS con el control adicional de un AWS CloudHSM clúster en su Cuenta de AWS entorno.

AWS KMS proporciona soporte completo de consola y API para crear, usar y administrar sus almacenes de claves personalizados. Puede usar las claves KMS de su almacén de claves personalizado de la misma manera que usa cualquier clave KMS. Por ejemplo, puede usar las claves KMS para generar claves de datos y para cifrar datos. También puede usar las claves de KMS en su almacén de claves personalizado con AWS servicios que admitan las claves administradas por el cliente.

¿Necesito un almacén de claves personalizado?

Para la mayoría de los usuarios, el almacén de AWS KMS claves predeterminado, que está protegido por [módulos criptográficos validados por el FIPS 140-3](#), cumple sus requisitos de seguridad. No es necesario agregar una capa extra de responsabilidad de mantenimiento o una dependencia de un servicio adicional.

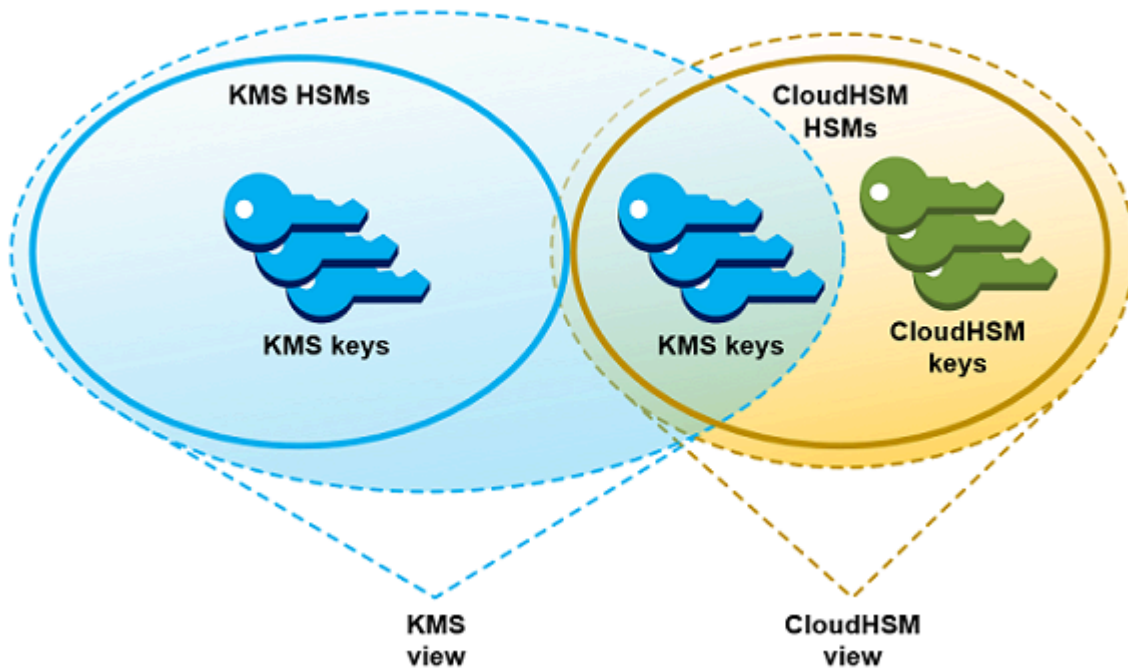
Sin embargo, puede plantearse crear un almacén de claves personalizado si su organización cumple alguno de los siguientes requisitos:

- Tiene claves que deben protegerse de forma explícita en un HSM de un solo inquilino o en un HSM sobre el que tiene control directo.
- Necesita poder eliminar inmediatamente el material clave de. AWS KMS
- Debe poder auditar todo el uso de sus claves de forma independiente AWS KMS o AWS CloudTrail.

¿Cómo funcionan los almacenes de claves personalizados?

Cada almacén de claves personalizado está asociado a un AWS CloudHSM clúster de su Cuenta de AWS. Al conectar el almacén de claves personalizado a su clúster, AWS KMS crea la infraestructura de red necesaria para respaldar la conexión. A continuación, inicia sesión en el AWS CloudHSM cliente de claves del clúster con las credenciales de un [usuario criptográfico dedicado](#) del clúster.

Puede crear y administrar sus almacenes de claves personalizados AWS KMS y crear y administrar sus clústeres de HSM en AWS CloudHSM. Al crear AWS KMS keys en un almacén de claves AWS KMS personalizado, puede ver y administrar las claves de KMS en AWS KMS. Pero también puede ver y administrar su material de claves en AWS CloudHSM, de igual modo que haría con otras claves en el clúster.



Puede [crear claves KMS de cifrado simétrico con el material de claves](#) generado AWS KMS en su almacén de claves personalizado. A continuación, utilice las mismas técnicas para ver y administrar las claves de KMS del almacén de claves personalizado que utiliza para las claves de KMS del almacén de AWS KMS claves. Puede controlar el acceso con políticas de claves y de IAM, crear etiquetas y alias, habilitar y desactivar las claves KMS y programar la eliminación de claves. Puede usar las claves de KMS para [operaciones criptográficas](#) y utilizarlas con AWS servicios que se integren con AWS KMS ellas.

Además, tiene el control total sobre el AWS CloudHSM clúster, lo que incluye la creación, la eliminación HSMs y la administración de las copias de seguridad. Puede usar el AWS CloudHSM cliente y las bibliotecas de software compatibles para ver, auditar y administrar el material clave de sus claves de KMS. Mientras el almacén de claves personalizadas esté desconectado, AWS KMS no podrá acceder a él y los usuarios no podrán utilizar las claves de KMS del almacén de claves personalizadas para operaciones criptográficas. Esta capa de control agregada convierte a los almacenes de claves personalizados en una solución potente para las organizaciones que la necesitan.

¿Por dónde empiezo?

Para crear y administrar un almacén de AWS CloudHSM claves, utilice las funciones de AWS KMS y AWS CloudHSM.

1. Comience en AWS CloudHSM. [Cree un clúster de AWS CloudHSM activo](#) o seleccione uno existente. El clúster debe tener al menos dos zonas de disponibilidad activas HSMs en diferentes zonas de disponibilidad. A continuación, cree una [cuenta de usuario de criptografía \(CU\) dedicado](#) en dicho clúster para AWS KMS.
2. En AWS KMS, [cree un almacén de claves personalizado](#) que esté asociado al AWS CloudHSM clúster seleccionado. AWS KMS proporciona una interfaz de administración completa que le permite crear, ver, editar y eliminar sus almacenes de claves personalizados.
3. Cuando esté listo para usar su almacén de claves personalizado, [conéctelo al AWS CloudHSM clúster asociado](#). AWS KMS crea la infraestructura de red que necesita para soportar la conexión. A continuación, se registra en el clúster con las credenciales de la cuenta de usuario de criptografía dedicado para que pueda generar y administrar material de claves en el clúster.
4. Ahora, puede [crear claves KMS de cifrado simétricas en su almacén de claves personalizado](#). Únicamente debe especificar el almacén de claves personalizado al crear la clave KMS.

Si se queda bloqueado en algún momento, puede buscar ayuda en el tema [Resolver problemas de un almacén de claves personalizado](#). Si su pregunta no tiene respuesta, utilice el enlace de comentarios en la parte inferior de cada página de esta guía o escriba en el [Foro de discusión de AWS Key Management Service](#).

Cuotas

AWS KMS permite hasta [10 almacenes de claves personalizados](#) en cada Cuenta de AWS región, incluidos los almacenes de [AWS CloudHSM claves y los almacenes](#) de [claves externos](#), independientemente del estado de conexión. Además, hay cuotas de AWS KMS solicitud [para el uso de claves KMS en un almacén de AWS CloudHSM claves](#).

Precios

Para obtener información sobre el costo de los almacenes de claves AWS KMS personalizados y las claves administradas por el cliente en un almacén de claves personalizado, consulte [AWS Key Management Service los precios](#). Para obtener información sobre el costo de AWS CloudHSM los clústeres HSMs, consulte [AWS CloudHSM los precios](#).

Regiones

AWS KMS admite tiendas AWS CloudHSM clave en todos los Regiones de AWS lugares donde AWS KMS sea compatible, excepto en Asia Pacífico (Melbourne), China (Beijing), China (Ningxia) y Europa (España).

Características no admitidas

AWS KMS no admite las siguientes funciones en los almacenes de claves personalizadas.

- [Claves de KMS asimétricas](#)
- [Claves KMS HMAC](#)
- [Claves KMS con material de claves importado](#)
- [Rotación automática de claves](#)
- [Claves de varias regiones](#)

AWS CloudHSM conceptos de almacén clave

En este tema se explican algunos de los términos y conceptos que se utilizan en las tiendas AWS CloudHSM clave.

AWS CloudHSM almacén de claves

Un almacén de AWS CloudHSM claves es un [almacén de claves personalizado](#) asociado a un AWS CloudHSM clúster que usted posee y administra. AWS CloudHSM los clústeres están respaldados por módulos de seguridad de hardware (HSMs) certificados según las normas [FIPS 140-2 o FIPS 140-3](#) de nivel 3.

Cuando crea una clave KMS en su almacén de claves, AWS KMS genera una AWS CloudHSM clave simétrica del estándar de cifrado avanzado (AES) de 256 bits, persistente y no exportable en el clúster asociado. AWS CloudHSM Este material clave nunca lo deja sin cifrar. HSMs Cuando se utiliza una clave KMS en un almacén de AWS CloudHSM claves, las operaciones criptográficas se realizan HSMs en el clúster.

AWS CloudHSM los almacenes de claves combinan la práctica y completa interfaz de administración de claves AWS KMS con los controles adicionales que proporciona un AWS CloudHSM clúster de su Cuenta de AWS empresa. Esta función integrada le permite crear, administrar y usar las claves de KMS y, AWS KMS al mismo tiempo, mantener el control total de las HSMs que almacenan su material clave, incluida la administración de los clústeres y las copias de seguridad. HSMs Puede usar la AWS KMS consola APIs para administrar el almacén de AWS CloudHSM claves y sus claves KMS. También puede usar la AWS CloudHSM consola APIs, el software de cliente y las bibliotecas de software asociadas para administrar el clúster asociado.

Puede [ver y administrar](#) su almacén de AWS CloudHSM claves, [editar sus propiedades](#) y [conectarlo](#) y [desconectarlo](#) del AWS CloudHSM clúster asociado. Si necesita [eliminar un almacén de AWS CloudHSM claves](#), primero debe eliminar las claves de KMS del almacén de claves programando su eliminación y esperando a que caduque el período de gracia. AWS CloudHSM Al eliminar el almacén de AWS CloudHSM claves, se elimina el recurso del AWS CloudHSM clúster AWS KMS, pero esto no afecta a él.

AWS CloudHSM clúster

Cada almacén de AWS CloudHSM claves está asociado a un AWS CloudHSM clúster. Cuando crea uno AWS KMS key en su almacén de AWS CloudHSM claves, AWS KMS crea su material clave en el clúster asociado. Al utilizar una clave de KMS en el almacén de claves de AWS CloudHSM , la operación criptográfica se realiza en el clúster asociado.

Cada AWS CloudHSM clúster solo se puede asociar a un almacén de AWS CloudHSM claves. El clúster que elija no se puede asociar a otro almacén de AWS CloudHSM claves ni compartir un historial de copias de seguridad con un clúster que esté asociado a otro almacén de AWS CloudHSM claves. El clúster debe estar inicializado y activo, y debe estar en la misma Cuenta de AWS región que el almacén de AWS CloudHSM claves. Puede crear un clúster nuevo o utilizar uno existente. AWS KMS no necesita el uso exclusivo del clúster. Para crear claves de KMS en el almacén de AWS CloudHSM claves, su clúster asociado, debe contener al menos dos claves activas HSMs. El resto de operaciones solo precisan un HSM.

El AWS CloudHSM clúster se especifica al crear el almacén de AWS CloudHSM claves y no se puede cambiar. Sin embargo, puede sustituir cualquier clúster que comparta un historial de copias de seguridad con el clúster original. De este modo podrá eliminar el clúster, en caso necesario, y reemplazarlo por uno creado a partir de una de sus copias de seguridad. Usted conserva el control total del AWS CloudHSM clúster asociado para poder administrar los usuarios y las claves, crear y eliminar HSMs, y usar y administrar las copias de seguridad.

Cuando esté listo para usar su almacén de AWS CloudHSM claves, lo conectará al AWS CloudHSM clúster asociado. Puede conectar y desconectar su almacén de claves personalizado en cualquier momento. Cuando el almacén de claves personalizado está conectado, se pueden crear y utilizar sus claves KMS. Cuando esté desconectado, podrá ver y administrar el almacén de AWS CloudHSM claves y sus claves KMS. Sin embargo, no puede crear nuevas claves de KMS ni utilizar las claves de KMS del almacén de AWS CloudHSM claves para operaciones criptográficas.

Usuario de criptografía de **kmsuser**

Para crear y administrar el material clave en el AWS CloudHSM clúster asociado en su nombre, AWS KMS utilice un [usuario AWS CloudHSM criptográfico](#) (CU) dedicado en el clúster denominado **kmsuser**. La **kmsuser** CU es una cuenta de CU estándar que se sincroniza automáticamente con todos los componentes del clúster y se guarda HSMs en las copias de seguridad del clúster.

Antes de crear el almacén de AWS CloudHSM claves, debe [crear una cuenta de **kmsuser** CU](#) en el AWS CloudHSM clúster mediante el comando [user create de](#) la CLI de CloudHSM. A continuación, al [crear el almacén de AWS CloudHSM claves](#), debe proporcionar la contraseña de la **kmsuser** cuenta a. AWS KMS Al [conectar el almacén de claves personalizado](#), AWS KMS inicia sesión en el clúster como **kmsuser** CU y cambia su contraseña. AWS KMS cifra la **kmsuser** contraseña antes de guardarla de forma segura. Cuando se gira la contraseña, la nueva contraseña se cifra y se almacena de la misma manera.

AWS KMS permanece conectado **kmsuser** mientras el almacén de AWS CloudHSM claves esté conectado. No debería usar esta cuenta de CU para otros fines. Sin embargo, conservará el control total de la cuenta del CU **kmsuser**. Podrá [buscar las claves](#) de propiedad de **kmsuser** en todo momento. Si es necesario, puede [desconectar el almacén de claves personalizado](#), cambiar la contraseña de **kmsuser**, [iniciar sesión en el clúster como **kmsuser**](#), y ver y administrar las claves propiedad de **kmsuser**.

Para obtener instrucciones sobre cómo crear la cuenta del CU **kmsuser**, consulte [Crear el usuario de criptografía **kmsuser**](#).

Claves KMS en un almacén de AWS CloudHSM claves

Puede usar la AWS KMS API AWS KMS o para crear una AWS KMS keys en un almacén de AWS CloudHSM claves. Utilice la misma técnica que utilizaría en cualquier clave KMS. La única diferencia es que debe identificar el almacén de AWS CloudHSM claves y especificar que el origen del material de claves es el AWS CloudHSM clúster.

Cuando se [crea una clave de KMS en un almacén de AWS CloudHSM claves](#), se AWS KMS crea la clave de KMS AWS KMS y se genera un material de clave simétrica del estándar de cifrado avanzado (AES) persistente y no exportable de 256 bits en el clúster asociado. Cuando se utiliza la AWS KMS clave en una operación criptográfica, la operación se realiza en el clúster mediante la clave AES basada en el AWS CloudHSM clúster. Aunque AWS CloudHSM admite claves simétricas

y asimétricas de distintos tipos, los almacenes de claves solo admiten claves de AWS CloudHSM cifrado simétricas AES.

Puede ver las claves KMS en un AWS CloudHSM almacén de claves de la AWS KMS consola y utilizar las opciones de la consola para mostrar el ID del almacén de claves personalizado. También puede utilizar la [DescribeKey](#) operación para buscar el ID del almacén de AWS CloudHSM claves y el ID del AWS CloudHSM clúster.

Las claves de KMS de un almacén de claves funcionan igual que cualquier otra AWS CloudHSM clave de KMS de un almacén de claves AWS KMS. Los usuarios autorizados necesitan los permisos para usar y administrar las claves KMS. Se utilizan los mismos procedimientos de consola y las mismas operaciones de API para ver y administrar las claves de KMS en un almacén de AWS CloudHSM claves. Puede habilitar y desactivar claves KMS, crear y usar etiquetas y alias, y configurar y cambiar las políticas de claves y de IAM. Puede utilizar las claves de KMS de un almacén de AWS CloudHSM claves para realizar operaciones criptográficas y utilizarlas con [AWS servicios integrados](#) que admitan el uso de claves gestionadas por el cliente. Sin embargo, no puede activar la [rotación automática de claves](#) ni [importar material de claves a una clave](#) de KMS de un almacén de AWS CloudHSM claves.

También puede utilizar el mismo proceso para [programar la eliminación](#) de una clave KMS en un almacén de AWS CloudHSM claves. Una vez transcurrido el período de espera, AWS KMS elimina la clave de KMS de KMS. A continuación, hace todo lo posible por eliminar el material clave de la clave KMS del AWS CloudHSM clúster asociado. Sin embargo, es posible que deba [eliminar el material de claves huérfano](#) manualmente del clúster y de sus copias de seguridad.

Controle el acceso a su almacén de AWS CloudHSM llaves

Las políticas de IAM se utilizan para controlar el acceso al almacén de AWS CloudHSM claves y al AWS CloudHSM clúster. Puede utilizar políticas clave, políticas de IAM y concesiones para controlar el acceso a su almacén AWS KMS keys de AWS CloudHSM claves. Le recomendamos que únicamente otorgue a los usuarios, grupos y roles los permisos necesarios para las tareas que es probable que se vayan a realizar.

Para respaldar sus almacenes de AWS CloudHSM claves, AWS KMS necesita permiso para obtener información sobre sus AWS CloudHSM clústeres. También necesita permiso para crear la infraestructura de red que conecta el almacén de AWS CloudHSM claves con su AWS CloudHSM clúster. Para obtener estos permisos, AWS KMS crea el rol `AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculado al servicio en su. Cuenta

de AWS Para obtener más información, consulte [Autorización AWS KMS para gestionar los recursos AWS CloudHSM de Amazon EC2](#).

Al diseñar su almacén de AWS CloudHSM claves, asegúrese de que los responsables que lo utilizan y administran solo tengan los permisos que necesitan. En la siguiente lista se describen los permisos mínimos necesarios para los administradores y usuarios del almacén de AWS CloudHSM claves.

- Los responsables que crean y administran el almacén de AWS CloudHSM claves necesitan el siguiente permiso para utilizar las operaciones de la API del almacén de AWS CloudHSM claves.
 - `cloudhsm:DescribeClusters`
 - `kms:CreateCustomKeyStore`
 - `kms:ConnectCustomKeyStore`
 - `kms>DeleteCustomKeyStore`
 - `kms:DescribeCustomKeyStores`
 - `kms:DisconnectCustomKeyStore`
 - `kms:UpdateCustomKeyStore`
 - `iam:CreateServiceLinkedRole`
- Los responsables que crean y administran el AWS CloudHSM clúster asociado a tu almacén de AWS CloudHSM claves necesitan permiso para crear e inicializar un clúster. AWS CloudHSM Esto incluye el permiso para crear o usar una Amazon Virtual Private Cloud (VPC), crear subredes y crear una instancia de Amazon. EC2 Es posible que también necesiten crear HSMs, eliminar y gestionar las copias de seguridad. Para obtener una lista de los permisos necesarios, consulte [Administración de identidades y accesos para AWS CloudHSM](#) en la Guía del usuario de AWS CloudHSM .
- Los responsables que crean y administran AWS KMS keys tu almacén de AWS CloudHSM claves necesitan [los mismos permisos](#) que quienes crean y administran cualquier clave de KMS. AWS KMS La [política de claves predeterminada](#) para una clave de KMS en un almacén de AWS CloudHSM claves es idéntica a la política de claves predeterminada para las claves de KMS en AWS KMS. [El control de acceso basado en atributos](#) (ABAC), que utiliza etiquetas y alias para controlar el acceso a las claves de KMS, también es efectivo en las claves de KMS de los almacenes de claves. AWS CloudHSM
- [Los responsables que utilizan las claves KMS de su almacén de claves para realizar operaciones criptográficas necesitan permiso para realizar la operación criptográfica con la AWS CloudHSM clave KMS, como KMS:Decrypt](#). Puede proporcionar estos permisos en una política de claves o

una política de IAM. Sin embargo, no necesitan ningún permiso adicional para usar una clave KMS en un almacén de claves. AWS CloudHSM

Crear un almacén de AWS CloudHSM claves

Puede crear uno o varios almacenes de AWS CloudHSM claves en su cuenta. Cada almacén de AWS CloudHSM claves está asociado a un AWS CloudHSM clúster de la misma Cuenta de AWS región. Antes de crear un almacén de claves de AWS CloudHSM, debe [cumplir los requisitos previos](#). A continuación, antes de poder usar el almacén de AWS CloudHSM claves, debe [conectarlo](#) a su AWS CloudHSM clúster.

Notas

El KMS no puede comunicarse IPv6 con los almacenes de AWS CloudHSM claves. Si intenta crear un almacén de AWS CloudHSM claves con todos los mismos valores de propiedades que un almacén de AWS CloudHSM claves desconectado existente, AWS KMS no crea un nuevo almacén de AWS CloudHSM claves y no genera ninguna excepción ni muestra un error. En su lugar, AWS KMS reconoce el duplicado como consecuencia probable de un reintento y devuelve el identificador del almacén de AWS CloudHSM claves existente.

No es necesario que conecte su almacén de AWS CloudHSM claves inmediatamente. Puede dejarlo desconectado hasta que lo necesite. Sin embargo, para comprobar que se ha configurado correctamente, debería [conectarlo](#), [ver su estado de conexión](#) y, a continuación, [desconectarlo](#).

Temas

- [Cumplir los requisitos previos](#)
- [Cree un nuevo almacén de claves AWS CloudHSM](#)

Cumplir los requisitos previos

Cada almacén de AWS CloudHSM claves está respaldado por un AWS CloudHSM clúster. Para crear un almacén de AWS CloudHSM claves, debe especificar un AWS CloudHSM clúster activo que no esté ya asociado a otro almacén de claves. También debe crear un usuario criptográfico (CU)

dedicado en el clúster HSMs que AWS KMS pueda usar para crear y administrar las claves en su nombre.

Antes de crear un almacén de AWS CloudHSM claves, haz lo siguiente:

Seleccione un AWS CloudHSM clúster

Cada almacén de AWS CloudHSM claves está [asociado exactamente a un AWS CloudHSM clúster](#). Cuando crea AWS KMS keys en su almacén de AWS CloudHSM claves, AWS KMS crea los metadatos de la clave de KMS, como un ID y un nombre de recurso de Amazon (ARN). AWS KMS a continuación, crea el material clave en HSMs el clúster asociado. Puede [crear un AWS CloudHSM clúster nuevo](#) o utilizar uno existente. AWS KMS no requiere acceso exclusivo al clúster.

El AWS CloudHSM clúster que seleccione está asociado permanentemente al almacén de AWS CloudHSM claves. Tras crear el almacén de AWS CloudHSM claves, puede [cambiar el ID de clúster](#) del clúster asociado, pero el clúster que especifique debe compartir un historial de respaldo con el clúster original. Para usar un clúster no relacionado, debe crear un nuevo almacén de AWS CloudHSM claves.

El AWS CloudHSM clúster que seleccione debe tener las siguientes características:

- El clúster debe estar activo.

Debe crear el clúster, inicializarlo, instalar el software de AWS CloudHSM cliente para su plataforma y, a continuación, activar el clúster. Para obtener instrucciones, consulte [Introducción a AWS CloudHSM](#) en la Guía del usuario de AWS CloudHSM .

- No está habilitado Mutual TLS (mTLS).

KMS no admite mTLS para los clústeres. Esta configuración no debe estar habilitada.

- El clúster debe estar en la misma cuenta y región que el almacén de AWS CloudHSM claves. No puede asociar un almacén de AWS CloudHSM claves de una región a un clúster de otra región. Para crear una infraestructura de claves en varias regiones, debe crear clústeres y almacenes de AWS CloudHSM claves en cada región.
- El clúster no puede estar asociado con otro almacén de claves personalizado en la misma cuenta y región. Cada almacén de AWS CloudHSM claves de la cuenta y la región debe estar asociado a un AWS CloudHSM clúster diferente. No puede especificar un clúster que ya esté asociado a un almacén de claves personalizado o a un clúster que comparte historial de copias de seguridad con un clúster asociado. Los clústeres que comparten un historial de copias de

seguridad deben tener el mismo certificado del clúster. Para ver el certificado de clúster de un clúster, utilice la AWS CloudHSM consola o la [DescribeClusters](#) operación.

Si la [copia de seguridad de un clúster de AWS CloudHSM en una región diferente](#), se considera un clúster diferente y puede asociar la copia de seguridad a un almacén de claves personalizado en su región. Sin embargo, las claves KMS de los dos almacenes de claves personalizados no son interoperables, incluso si tienen la misma clave de respaldo. AWS KMS vincula los metadatos al texto cifrado para que solo se puedan descifrar con la clave KMS que los cifró.

- El clúster debe configurarse con [subredes privadas](#) en al menos dos zonas de disponibilidad de la región. Como no AWS CloudHSM es compatible con todas las zonas de disponibilidad, le recomendamos que cree subredes privadas en todas las zonas de disponibilidad de la región. No puede volver a configurar las subredes para un clúster existente, pero puede [crear un clúster a partir de una copia de seguridad](#) con subredes distintas en la configuración del clúster.

 Important

Después de crear el almacén de AWS CloudHSM claves, no elimine ninguna de las subredes privadas configuradas para su AWS CloudHSM clúster. Si AWS KMS no puede encontrar todas las subredes de la configuración del clúster, los intentos de [conexión al almacén de claves personalizado](#) fallan y se produce un estado de error de SUBNET_NOT_FOUND conexión. Para obtener más información, consulte [Cómo arreglar un error de conexión](#).

- El [grupo de seguridad del clúster](#) (cloudhsm-cluster-*<cluster-id>*-sg) debe incluir reglas de entrada y reglas de salida que permitan el tráfico TCP a través de los puertos IPv4 2223-2225. El origen de las reglas de entrada y el destino de las reglas de salida deben coincidir con el ID del grupo de seguridad. Estas reglas se establecen de forma predeterminada al crear el clúster. No las elimine ni las cambie.
- El clúster debe contener al menos dos activos HSMs en distintas zonas de disponibilidad. Para comprobar el número de HSMs, utilice la AWS CloudHSM consola o la [DescribeClusters](#) operación. Si es necesario, puede [agregar un HSM](#).

Buscar el certificado de anclaje de confianza

Al crear un almacén de claves personalizado, debe cargar el certificado de anclaje de confianza del AWS CloudHSM clúster a AWS KMS. AWS KMS necesita el certificado de anclaje de

confianza para conectar el almacén de AWS CloudHSM claves al AWS CloudHSM clúster asociado.

Cada AWS CloudHSM clúster activo tiene un certificado de anclaje de confianza. Al [inicializar el clúster](#), se genera el certificado. Guárdelo en el archivo `customerCA.crt` y cópielo en alojamientos que se conecten con el clúster.

Cree el usuario **kmsuser** criptográfico para AWS KMS

Para administrar su almacén de AWS CloudHSM claves, AWS KMS inicie sesión en la cuenta [de usuario kmsuser criptográfico](#) (CU) del clúster seleccionado. Antes de crear el almacén de AWS CloudHSM claves, debe crear la kmsuser CU. A continuación, al crear el almacén de AWS CloudHSM claves, debe proporcionar la contraseña kmsuser para AWS KMS. Siempre que conectes el almacén de AWS CloudHSM claves a su AWS CloudHSM clúster asociado, AWS KMS inicia sesión con la kmsuser contraseña y rota la contraseña kmsuser

 Important

No especifique la opción 2FA al crear el CU kmsuser. Si lo hace, AWS KMS no podrá iniciar sesión y su almacén de AWS CloudHSM claves no podrá conectarse a este AWS CloudHSM clúster. Después de especificar 2FA, ya no podrá deshacer dicha acción. En su lugar, deberá eliminar el CU y crearlo de nuevo.

 Notas

Los siguientes procedimientos utilizan la herramienta de línea de comandos AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI de CloudHSM reemplaza `key-handle` por `key-reference`.

El 1 de enero de 2025, AWS CloudHSM dejará de ser compatible con las herramientas de línea de comandos Client SDK 3, la utilidad de administración CloudHSM (CMU) y la utilidad de administración de claves (KMU). Para obtener más información sobre las diferencias entre las herramientas de línea de comandos Client SDK 3 y Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) en la Guía del usuario de AWS CloudHSM .

1. Siga los procedimientos de introducción descritos en el tema [Getting started with CloudHSM Command Line Interface \(CLI\)](#) de la Guía del usuario de AWS CloudHSM .

2. Utilice el comando [user create](#) para crear un CU denominado kmsuser.

La contraseña debe contener entre 7 y 32 caracteres alfanuméricos, distingue entre mayúsculas y minúsculas, y no puede tener caracteres especiales.

El siguiente ejemplo de comando crea un CU de kmsuser.

```
aws-cloudhsm > user create --username kmsuser --role crypto-user
Enter password:
Confirm password:
{
  "error_code": 0,
  "data": {
    "username": "kmsuser",
    "role": "crypto-user"
  }
}
```

Cree un nuevo almacén de claves AWS CloudHSM

Tras [reunir los requisitos previos](#), puede crear un nuevo almacén de AWS CloudHSM claves en la AWS KMS consola o mediante la [CreateCustomKeyStore](#) operación.

Uso de la consola AWS KMS

Al crear un almacén de AWS CloudHSM claves en Consola de administración de AWS, puede añadir y crear los [requisitos previos](#) como parte de su flujo de trabajo. Sin embargo, el proceso es más rápido si los compila previamente.

1. Inicia sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrela en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Almacenes de claves personalizados, Almacenes de claves de AWS CloudHSM .
4. Seleccione Create a key store (Crear un m almacén de claves).
5. Escriba un nombre fácil de recordar para el almacén de claves personalizado. El nombre debe ser único entre todos los almacenes de claves personalizados de su cuenta.

 Important

No incluya información confidencial en este campo. Este campo puede mostrarse en texto plano en los CloudTrail registros y otros resultados.

6. Seleccione [un AWS CloudHSM clúster](#) para el almacén de AWS CloudHSM claves. O bien, para crear un AWS CloudHSM clúster nuevo, elija el enlace Crear un AWS CloudHSM clúster.

El menú muestra los AWS CloudHSM clústeres de su cuenta y región que aún no están asociados a un almacén de AWS CloudHSM claves. El clúster debe [cumplir los requisitos](#) de asociación con un almacén de claves personalizado.

7. Seleccione Elegir archivo y, a continuación, cargue el certificado de anclaje de confianza del AWS CloudHSM clúster que haya elegido. Este es el archivo `customerCA.crt` que creó al [inicializar el clúster](#).
8. Escriba la contraseña [del kmsuser usuario de criptografía](#) (CU) que creó en el clúster seleccionado.
9. Seleccione Crear.

Cuando el procedimiento se complete correctamente, el nuevo almacén de AWS CloudHSM claves aparecerá en la lista de almacenes de AWS CloudHSM claves de la cuenta y la región. Si el procedimiento da error, aparecerá un mensaje de error donde se describe el problema y se explica cómo resolverlo. Si necesita más ayuda, consulte [Resolver problemas de un almacén de claves personalizado](#).

Si intenta crear un almacén de AWS CloudHSM claves con todos los mismos valores de propiedades que un almacén de AWS CloudHSM claves desconectado existente, AWS KMS no se crea un almacén de AWS CloudHSM claves nuevo y no se produce ninguna excepción ni se muestra ningún error. En su lugar, AWS KMS reconoce el duplicado como consecuencia probable de un reintento y devuelve el identificador del almacén de AWS CloudHSM claves existente.

Siguiente: Los nuevos almacenes de AWS CloudHSM claves no se conectan automáticamente. Para poder crear AWS KMS keys en el almacén de AWS CloudHSM claves, debe [conectar el almacén de claves personalizado](#) a su AWS CloudHSM clúster asociado.

Uso de la AWS KMS API

Puede usar la [CreateCustomKeyStore](#) operación para crear un nuevo almacén de AWS CloudHSM claves asociado a un AWS CloudHSM clúster de la cuenta y la región. En estos ejemplos se utiliza la AWS Command Line Interface (AWS CLI), pero puede usar cualquier lenguaje de programación admitido.

La operación `CreateCustomKeyStore` requiere los siguientes valores de parámetro.

- `CustomKeyName` — Un nombre descriptivo para el almacén de claves personalizado que es único en la cuenta.

Important

No incluya información confidencial en este campo. Este campo puede mostrarse en texto plano en CloudTrail los registros y otros resultados.

- `CloudHsmClusterId` — El ID de clúster de un AWS CloudHSM clúster que [cumple los requisitos de](#) un almacén de AWS CloudHSM claves.
- `KeyStorePassword` — La contraseña de la cuenta kmsuser CU del clúster especificado.
- `TrustAnchorCertificate` — El contenido del `customerCA.crt` archivo que creó al [inicializar el clúster](#).

En el siguiente ejemplo se usa un ID de clúster ficticio. Antes de ejecutar el comando, reemplácelo por un ID de clúster válido.

```
$ 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>
```

Si utiliza el AWS CLI, puede especificar el archivo de certificado de anclaje de confianza, en lugar de su contenido. En el siguiente ejemplo, el archivo `customerCA.crt` se encuentra en el directorio raíz.

```
$ 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
```

Si la operación se ejecuta correctamente, `CreateCustomKeyStore` devolverá el ID del almacén de claves personalizado, tal y como se muestra en la siguiente respuesta de ejemplo.

```
{  
  "CustomKeyId": cks-1234567890abcdef0  
}
```

Si la operación falla, corrija el error que indica la excepción e inténtelo de nuevo. Para obtener ayuda adicional, consulte [Resolver problemas de un almacén de claves personalizado](#).

Si intenta crear un almacén de AWS CloudHSM claves con todos los mismos valores de propiedades que un almacén de AWS CloudHSM claves desconectado existente, AWS KMS no se crea un almacén de AWS CloudHSM claves nuevo y no se produce ninguna excepción ni se muestra un error. En su lugar, AWS KMS reconoce el duplicado como consecuencia probable de un reintento y devuelve el identificador del almacén de AWS CloudHSM claves existente.

A continuación: Para usar el almacén de AWS CloudHSM claves, [conéctelo a su AWS CloudHSM clúster](#).

Ver un almacén de AWS CloudHSM claves

Puede ver los almacenes de AWS CloudHSM claves de cada cuenta y región mediante la AWS KMS consola o la [DescribeCustomKeyStores](#) operación.

Uso de la AWS KMS consola

Al ver los almacenes de AWS CloudHSM claves del Consola de administración de AWS, puede ver lo siguiente:

- El nombre y el ID del almacén de claves personalizado
- El ID del AWS CloudHSM clúster asociado
- El número de HSMs en el clúster
- El estado de conexión actual

Un valor de estado de conexión (estado) de desconectado indica que el almacén de claves personalizado es nuevo y nunca se ha conectado, o que se ha [desconectado intencionadamente de](#)

[su AWS CloudHSM clúster](#). Sin embargo, si sus intentos de utilizar una clave KMS en un almacén de claves personalizadas conectado fallan, podría indicar que hay un problema con el almacén de claves personalizado o su AWS CloudHSM clúster. Para obtener ayuda, consulte [¿Cómo arreglar una clave KMS que produce error?](#).

Para ver los almacenes de AWS CloudHSM claves de una cuenta y región determinadas, utilice el siguiente procedimiento.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Almacenes de claves personalizados, Almacenes de claves de AWS CloudHSM .

Para personalizar la pantalla, haga clic en el icono de engranaje que aparece por debajo del botón Create key store (Crear almacén de claves).

Uso de la API AWS KMS

Para ver sus almacenes de AWS CloudHSM claves, utilice la [DescribeCustomKeyStores](#) operación. De forma predeterminada, esta operación devuelve los almacenes de claves personalizados de la cuenta y región. Pero puede usar el parámetro CustomKeyId o CustomKeyName (pero no ambos) para limitar el resultado de un almacén de claves personalizado determinado. En el caso de los almacenes de claves, la salida consiste en el nombre y el ID del almacén de claves personalizados, el ID del AWS CloudHSM clúster asociado y el estado de la conexión. AWS CloudHSM Si el estado de conexión indica un error, el resultado también incluirá un código de error que describe el motivo del error.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Por ejemplo, el siguiente comando devuelve todos los almacenes de claves personalizados de la cuenta y la región. Puede usar los parámetros Limit y Marker para desplazarse por los almacenes de claves personalizados del resultado.

```
$ aws kms describe-custom-key-stores
```

El siguiente comando de ejemplo usa el parámetro `CustomKeyStoreName` para obtener únicamente el almacén de claves personalizado con el nombre fácil de recordar `ExampleCloudHSMKeyStore`. Puede usar el parámetro `CustomKeyStoreName` o `CustomKeyStoreId` (pero no ambos) en cada comando.

El siguiente ejemplo de salida representa un almacén de AWS CloudHSM claves que está conectado a su AWS CloudHSM clúster.

 Note

El `CustomKeyStoreType` campo se agregó a la `DescribeCustomKeyStores` respuesta para distinguir los almacenes de AWS CloudHSM claves de los almacenes de claves 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>"
    }
  ]
}
```

Un `ConnectionState` número de `Disconnected` indica que un almacén de claves personalizado nunca se ha conectado o que se ha [desconectado intencionalmente de su AWS CloudHSM clúster](#). Sin embargo, si los intentos de utilizar una clave KMS en un almacén de AWS CloudHSM claves conectado fallan, podría indicar que hay un problema con el almacén de AWS CloudHSM claves o su AWS CloudHSM clúster. Para obtener ayuda, consulte [¿Cómo arreglar una clave KMS que produce error?](#).

Si el `ConnectionState` del almacén de claves personalizado es `FAILED`, la respuesta `DescribeCustomKeyStores` incluirá un elemento `ConnectionErrorCode` que explica el motivo del error.

Por ejemplo, en el resultado siguiente, el valor `INVALID_CREDENTIALS` indica que la conexión del almacén de claves personalizado ha fallado porque la contraseña de [kmsuser no es válida](#). Para obtener ayuda con esto y los errores de conexión, consulte [Resolver problemas de un almacén de claves 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>"
    }
  ]
}
```

Más información:

- [Visualización de almacenes de claves externos](#)
- [Identifique las claves de KMS en AWS CloudHSM almacenes de claves](#)
- [Registro AWS KMS Llamadas a la API con AWS CloudTrail](#)

Editar la configuración AWS CloudHSM del almacén de claves

Puede cambiar la configuración de un almacén de AWS CloudHSM claves existente. El almacén de claves personalizado debe estar desconectado de su AWS CloudHSM clúster.

Para editar la configuración del almacén de AWS CloudHSM claves:

1. [Desconecte el almacén de claves personalizado](#) de su clúster de AWS CloudHSM .

Mientras el almacén de claves personalizadas esté desconectado, no podrá crear AWS KMS keys (claves de KMS) en el almacén de claves personalizadas ni podrá utilizar las claves de KMS que contiene para [operaciones criptográficas](#).

2. Edite una o varias de las configuraciones del almacén de AWS CloudHSM claves.

Puede editar la siguiente configuración en un almacén de claves personalizado:

El nombre fácil de recordar del almacén de claves personalizado.

Escriba un nuevo nombre fácil de recordar. El nuevo nombre debe ser único entre todos los almacenes de claves personalizadas de su Cuenta de AWS.

 Important

No incluya información confidencial en este campo. Este campo puede mostrarse en texto plano en CloudTrail los registros y otros resultados.

El ID de clúster del AWS CloudHSM clúster asociado.

Edite este valor para sustituir el AWS CloudHSM clúster original por un clúster relacionado. Puede utilizar esta función para reparar un almacén de claves personalizado si su AWS CloudHSM clúster se daña o se elimina.

Especifique un AWS CloudHSM clúster que comparta un historial de copias de seguridad con el clúster original y que [cumpla los requisitos](#) para asociarse a un almacén de claves personalizado, incluidos dos activos HSMs en distintas zonas de disponibilidad. Los clústers que comparten un historial de copias de seguridad deben tener el mismo certificado del clúster. Para ver el certificado de clúster de un clúster, utilice la [DescribeClusters](#) operación. No puede usar la característica de edición para asociar el almacén de claves personalizado con un clúster de AWS CloudHSM sin relación.

La contraseña actual del [kmsuser usuario de criptografía](#) (CU).

Indica AWS KMS la contraseña actual de la kmsuser CU del AWS CloudHSM clúster. Esta acción no cambia la contraseña de la kmsuser CU del AWS CloudHSM clúster.

Si cambia la contraseña de la kmsuser CU del AWS CloudHSM clúster, utilice esta función para indicar AWS KMS la nueva kmsuser contraseña. De lo contrario, AWS KMS no podrá iniciar sesión en el clúster y todos los intentos de conexión del almacén de claves personalizado al clúster darán error.

3. [Vuelva a conectar el almacén de claves personalizado](#) a su clúster de AWS CloudHSM .

Edición de su configuración del almacén de claves

Puede editar la configuración del almacén de AWS CloudHSM claves en la AWS KMS consola o mediante esta [UpdateCustomKeyStore](#) operación.

Uso de la AWS KMS consola

Al editar un almacén de AWS CloudHSM claves, puede cambiar cualquiera de los valores configurables o alguno de ellos.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Almacenes de claves personalizados, Almacenes de claves de AWS CloudHSM .
4. Elija la fila del almacén de AWS CloudHSM claves que desee editar.

Si el valor de la columna Connection state (Estado de conexión) no es Disconnected (Desconectado), deberá desconectar el almacén de claves personalizado antes de editarlo. [En el menú Key store actions (Acciones del almacén de claves), seleccione Disconnect (Desconectar)].

Mientras un almacén de AWS CloudHSM claves esté desconectado, puede administrar el almacén de AWS CloudHSM claves y sus claves de KMS, pero no puede crear ni usar claves de KMS en el almacén de AWS CloudHSM claves.

5. Desde el menú Key store actions (Acciones del almacén de claves), seleccione Edit (Editar).
6. Realice una o más de las siguientes acciones.
 - Escriba un nuevo nombre fácil de recordar para el almacén de claves personalizado.
 - Escriba el ID de clúster de un AWS CloudHSM clúster relacionado.
 - Escriba la contraseña actual del usuario kmsuser criptográfico del AWS CloudHSM clúster asociado.
7. Seleccione Save.

Si el procedimiento se ejecuta correctamente, aparecerá un mensaje donde se describe la configuración que ha editado. Si el procedimiento da error, aparecerá un mensaje de error

donde se describe el problema y se explica cómo resolverlo. Si necesita más ayuda, consulte [Resolver problemas de un almacén de claves personalizado](#).

8. [Vuelva a conectar el almacén de claves personalizado](#).

Para usar el almacén de AWS CloudHSM claves, debe volver a conectarlo después de editarlo. Puede dejar el almacén de claves de AWS CloudHSM desconectado. Sin embargo, mientras esté desconectado, no podrá crear claves de KMS en el almacén de AWS CloudHSM claves ni utilizar las claves de KMS del almacén de AWS CloudHSM claves en operaciones [criptográficas](#).

Uso de la API AWS KMS

Para cambiar las propiedades de un almacén de AWS CloudHSM claves, utilice la [UpdateCustomKeyStore](#) operación. Puede cambiar varias propiedades de un almacén de claves personalizado en el mismo comando. Si la operación se realiza correctamente, AWS KMS devuelve una respuesta HTTP 200 y un objeto JSON sin propiedades. Para comprobar que los cambios son efectivos, utilice la [DescribeCustomKeyStores](#) operación.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Comience por [DisconnectCustomKeyStore](#) desconectar el almacén de claves personalizado de su AWS CloudHSM clúster. Reemplace el ID del almacén de claves personalizado de ejemplo, cks-1234567890abcdef0, por un ID real.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

En el primer ejemplo, [UpdateCustomKeyStore](#) se utiliza para cambiar el nombre descriptivo del almacén de AWS CloudHSM claves a DevelopmentKeys. El comando usa el CustomKeyId parámetro para identificar el almacén de AWS CloudHSM claves y CustomKeyName para especificar el nuevo nombre del almacén de claves personalizado.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-custom-key-store-name DevelopmentKeys
```

En el siguiente ejemplo, se cambia el clúster asociado a un almacén de AWS CloudHSM claves por otra copia de seguridad del mismo clúster. El comando usa el CustomKeyId parámetro para identificar el almacén de AWS CloudHSM claves y el CloudHsmClusterId parámetro para especificar el nuevo ID de clúster.


```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

El siguiente ejemplo indica AWS KMS que la kmsuser contraseña actual esExamplePassword. El comando utiliza el CustomKeyStoreId parámetro para identificar el almacén de AWS CloudHSM claves y el KeyStorePassword parámetro para especificar la contraseña actual.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-store-password ExamplePassword
```

El último comando vuelve a conectar el almacén de AWS CloudHSM claves a su AWS CloudHSM clúster. Puede dejar el almacén de claves personalizado desconectado, pero deberá conectarlo antes de crear claves KMS nuevas o para utilizar las claves KMS existentes para [operaciones criptográficas](#). Reemplace el ID del almacén de claves personalizado de ejemplo por un ID real.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Conecta un almacén de AWS CloudHSM claves

Los nuevos almacenes de AWS CloudHSM claves no están conectados. Para poder crearlo y usarlo AWS KMS keys en su almacén de AWS CloudHSM claves, debe conectarlo al AWS CloudHSM clúster asociado. Puedes conectar y desconectar tu almacén de AWS CloudHSM claves en cualquier momento y [ver su estado de conexión](#).

No es necesario que conecte su almacén de AWS CloudHSM claves. Puede dejar un almacén de AWS CloudHSM claves desconectado indefinidamente y conectarlo solo cuando necesite usarlo. Sin embargo, le recomendamos que compruebe la conexión de forma periódica para verificar que la configuración es correcta y que se puede conectar.

Note

AWS CloudHSM los almacenes de claves tienen un estado de DISCONNECTED conexión solo cuando el almacén de claves nunca se ha conectado o si lo desconectas explícitamente. Si el estado de conexión del almacén de AWS CloudHSM claves es CONNECTED pero tiene problemas para usarlo, asegúrese de que el AWS CloudHSM clúster asociado esté activo y contenga al menos uno activo HSMs. Si desea ayuda con las conexiones que dan error, consulte [the section called “Resolver problemas de un almacén de claves personalizado”](#).

Al conectar un almacén de AWS CloudHSM claves, AWS KMS busca el AWS CloudHSM clúster asociado, se conecta a él, inicia sesión en el AWS CloudHSM cliente como [usuario kmsuser criptográfico](#) (CU) y, a continuación, cambia la kmsuser contraseña. AWS KMS permanece conectado al AWS CloudHSM cliente mientras el almacén de AWS CloudHSM claves esté conectado.

Para establecer la conexión, AWS KMS crea un [grupo de seguridad](#) denominado kms-*<custom key store ID>* en la nube privada virtual (VPC) del clúster. El grupo de seguridad tiene una sola regla que permite el tráfico entrante desde el grupo de seguridad del clúster. AWS KMS también crea una [interface de red elástica](#) (ENI) en cada zona de disponibilidad de la subred privada del clúster. AWS KMS agrega el ENIs al grupo kms-*<cluster ID>* de seguridad y al grupo de seguridad del clúster. La descripción de cada ENI es KMS managed ENI for cluster *<cluster-ID>*.

El proceso de conexión puede tardar bastante en completarse, unos 20 minutos.

Antes de conectar el almacén de AWS CloudHSM claves, compruebe que cumple los requisitos.

- Su AWS CloudHSM clúster asociado debe contener al menos un HSM activo. Para encontrar el número de HSMs en el clúster, visualice el clúster en la AWS CloudHSM consola o utilice la [DescribeClusters](#) operación. Si es necesario, puede [agregar un HSM](#).
- El clúster debe tener una cuenta de [usuario kmsuser criptográfico](#) (CU), pero esa CU no puede iniciar sesión en el clúster cuando se conecta al almacén de AWS CloudHSM claves. Para obtener ayuda con el cierre de sesión, consulte [Cómo cerrar sesión y volver a conectar](#).
- El estado de conexión del almacén de AWS CloudHSM claves no puede ser DISCONNECTING o FAILED. Para ver el estado de la conexión, utilice la AWS KMS consola o la [DescribeCustomKeyStores](#) respuesta. Si el estado de conexión es FAILED, desconecte el almacén de claves personalizado, resuelva el problema y conéctelo de nuevo.

Si desea ayuda con las conexiones que dan error, consulte [Cómo arreglar un error de conexión](#).

Cuando su almacén de AWS CloudHSM claves esté conectado, podrá [crear claves de KMS en él y utilizar las claves](#) de KMS existentes en las [operaciones criptográficas](#).

Conéctese y vuelva a conectarse a su almacén de AWS CloudHSM llaves

Puede conectar o volver a conectar el almacén de AWS CloudHSM llaves en la AWS KMS consola o mediante esta operación. [ConnectCustomKeyStore](#)

Uso de la consola AWS KMS

Para conectar un almacén de AWS CloudHSM claves al Consola de administración de AWS, comience por seleccionar el almacén de AWS CloudHSM claves en la página Almacenes de claves personalizados. El proceso de conexión puede tardar hasta 20 minutos en completarse.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Almacenes de claves personalizados, Almacenes de claves de AWS CloudHSM .
4. Elija la fila del almacén de AWS CloudHSM claves que desee conectar.

Si el estado de conexión del almacén de AWS CloudHSM claves es Fallido, debe [desconectar el almacén de claves personalizado](#) antes de conectarlo.

5. En el menú Key store actions (Acciones del almacén de claves), seleccione Connect (Conectar).

AWS KMS inicia el proceso de conexión del almacén de claves personalizado. Encuentra el clúster del AWS CloudHSM asociado, genera la infraestructura de red necesaria, se conecta a él, inicia sesión en el clúster de AWS CloudHSM con el CU kmsuser y rota la contraseña kmsuser. Cuando la operación finalizada, el estado de conexión cambia a Connected (Conectado).

Si la operación falla, aparecerá un mensaje de error que describe el motivo del error. Antes de volver a intentar conectarse, [consulte el estado de conexión](#) del almacén de AWS CloudHSM claves. Si es Failed (Error), deberá [desconectar el almacén de claves personalizado](#) antes de conectarlo de nuevo. Si necesita más ayuda, consulte [Resolver problemas de un almacén de claves personalizado](#).

Siguiente: [the section called “Crear una clave KMS en un almacén de AWS CloudHSM claves”](#).

Uso de la AWS KMS API

Para conectar un almacén de AWS CloudHSM claves desconectado, utilice la [ConnectCustomKeyStore](#) operación. El AWS CloudHSM clúster asociado debe contener al menos un HSM activo y el estado de la conexión no puede ser FAILED el mismo.

El proceso de conexión puede tardar bastante en completarse, unos 20 minutos. A menos que el error sea rápido, la operación devuelve una respuesta HTTP 200 y un objeto JSON sin propiedades. Sin embargo, esta respuesta inicial no indica que la conexión se haya realizado

correctamente. Para determinar el estado de conexión del almacén de claves personalizado, consulte la [DescribeCustomKeyStores](#) respuesta.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Para identificar el almacén de AWS CloudHSM claves, utilice su ID de almacén de claves personalizado. Puede encontrar el ID en la página de almacenes de claves personalizados de la consola o mediante la [DescribeCustomKeyStores](#) operación sin parámetros. Antes de ejecutar este ejemplo, reemplace el ID de ejemplo por uno válido.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Para comprobar que el almacén de AWS CloudHSM claves está conectado, utilice la [DescribeCustomKeyStores](#) operación. De forma predeterminada, esta operación devuelve los almacenes de claves personalizados de su cuenta y región. Pero puede usar el parámetro CustomKeyId o CustomKeyName (pero no ambos) para limitar la respuesta a almacenes de claves personalizados determinados. El valor ConnectionState de CONNECTED indica que el almacén de claves personalizado está conectado a su clúster de AWS CloudHSM .

 Note

El CustomKeyType campo se agregó a la DescribeCustomKeyStores respuesta para distinguir los almacenes de AWS CloudHSM claves de los almacenes de claves externos.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleCloudHSMKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CustomKeyType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
    }
  ],
}
```

Si el valor `ConnectionState` da error, el elemento `ConnectionErrorCode` indicará el motivo del error. En este caso, no se pudo encontrar un AWS CloudHSM clúster en su cuenta con el ID del clúster `cluster-1a23b4cdefg`. Si ha eliminado el clúster, puede [restaurarlo a partir de una copia de seguridad](#) del clúster original y, a continuación, [editar el ID del clúster](#) para el almacén de claves personalizado. Para obtener ayuda sobre cómo responder a un código de error de conexión, consulte [Cómo arreglar un error de conexión](#).

```
$ 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 un almacén de AWS CloudHSM claves

Al desconectar un almacén de AWS CloudHSM claves, se AWS KMS cierra la sesión del AWS CloudHSM cliente, se desconecta del AWS CloudHSM clúster asociado y se elimina la infraestructura de red que creó para admitir la conexión.

Mientras un almacén de AWS CloudHSM claves está desconectado, puede administrar el almacén de AWS CloudHSM claves y sus claves de KMS, pero no puede crear ni usar claves de KMS en el almacén de AWS CloudHSM claves. El estado de conexión del almacén de claves es `DISCONNECTED` y el [estado de clave](#) de las claves de KMS en el almacén de claves personalizado es `Unavailable`, a menos que sean `PendingDeletion`. Puede volver a conectar el almacén de AWS CloudHSM claves en cualquier momento.

Note

AWS CloudHSM los almacenes de claves tienen un estado de `DISCONNECTED` conexión solo cuando el almacén de claves nunca se ha conectado o si se desconecta explícitamente. Si el estado de conexión del almacén de AWS CloudHSM claves es `CONNECTED` pero tiene problemas para usarlo, asegúrese de que el AWS CloudHSM clúster asociado esté activo.

y contenga al menos uno activo HSMs. Si desea ayuda con las conexiones que dan error, consulte [the section called “Resolver problemas de un almacén de claves personalizado”](#).

Al desconectar un almacén de claves personalizado, las claves de KMS del almacén de claves quedan inutilizables de inmediato (sujeto a posible coherencia). Sin embargo, los recursos cifrados con [claves de datos](#) protegidas por la clave de KMS no se ven afectados hasta que se vuelva a utilizar la clave de KMS, por ejemplo, para descifrar la clave de datos. Este problema afecta a los Servicios de AWS, muchos de los cuales utilizan claves de datos para proteger sus recursos. Para obtener más información, consulte [Cómo afectan las claves de KMS obsoletas a las claves de datos](#).

 Note

Mientras un almacén de claves personalizado esté desconectado, todos los intentos de crear claves KMS en el almacén de claves personalizado o de usar claves KMS existentes en operaciones criptográficas fallarán. Esta acción puede impedir que los usuarios almacenen y accedan a datos confidenciales.

Para realizar una mejor estimación del efecto de desconectar el almacén de claves personalizado, [identifique las claves de KMS](#) en el almacén de claves personalizado y [determine su uso en el pasado](#).

Puede desconectar un almacén de AWS CloudHSM claves por motivos como los siguientes:

- Para rotar la contraseña **kmsuser**. AWS KMS cambia la contraseña de **kmsuser** cada vez que se conecta al clúster de AWS CloudHSM. Para forzar la rotación de contraseñas, desconecte y vuelva a conectar.
- Para auditar el material clave de las claves de KMS del AWS CloudHSM clúster. Al desconectar el almacén de claves personalizado, AWS KMS cierra sesión en la cuenta de [usuario kmsuser criptográfico](#) del AWS CloudHSM cliente. Esto le permite iniciar sesión en el clúster con el CU **kmsuser** y auditar y administrar el material de claves para la clave KMS.
- Para desactivar de inmediato todas las claves de KMS en un almacén de claves de AWS CloudHSM. Puede [deshabilitar y volver a habilitar las claves de KMS](#) en un almacén de AWS CloudHSM claves mediante la [DisableKey](#) operación Consola de administración de AWS o. Estas operaciones se completan rápidamente, pero actúan en una clave KMS cada vez. Al desconectar el almacén de AWS CloudHSM claves, se cambia inmediatamente el estado de todas las claves

KMS del almacén de AWS CloudHSM clavesUnavailable, lo que impide que se utilicen en cualquier operación criptográfica.

- Para reparar un error en la conexión. Si se produce un error al intentar conectar un almacén de AWS CloudHSM claves (el estado de conexión del almacén de claves personalizado es el mismoFAILED), debe desconectar el almacén de AWS CloudHSM claves antes de intentar volver a conectarlo.

Desconecte el almacén de AWS CloudHSM claves

Puede desconectar el almacén de AWS CloudHSM llaves en la AWS KMS consola o mediante esta [DisconnectCustomKeyStore](#) operación.

Desconéctelo mediante la AWS KMS consola

Para desconectar un almacén de AWS CloudHSM claves conectado de la AWS KMS consola, comience por elegir el almacén de AWS CloudHSM claves en la página Almacenes de claves personalizados.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Almacenes de claves personalizados, Almacenes de claves de AWS CloudHSM .
4. Elija la fila del almacén de claves externo que desee desconectar.
5. En el menú Key store actions (Acciones del almacén de claves), seleccione Disconnect (Desconectar).

Cuando la operación finaliza, el estado de conexión cambia de Disconnecting (Desconectando) a Disconnected (Desconectado). Si la operación da error, aparecerá un mensaje de error donde se describe el problema y se explica cómo resolverlo. Si necesita más ayuda, consulte [Resolver problemas de un almacén de claves personalizado](#).

Desconéctese mediante la API AWS KMS

Para desconectar un almacén de AWS CloudHSM claves conectado, utilice la [DisconnectCustomKeyStore](#) operación. Si la operación se realiza correctamente, AWS KMS devuelve una respuesta HTTP 200 y un objeto JSON sin propiedades.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

En este ejemplo, se desconecta un almacén de AWS CloudHSM claves. Antes de ejecutar este ejemplo, reemplace el ID de ejemplo por uno válido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Para comprobar que el almacén de AWS CloudHSM claves está desconectado, utilice la [DescribeCustomKeyStores](#) operación. De forma predeterminada, esta operación devuelve los almacenes de claves personalizados de su cuenta y región. Pero puede usar el parámetro CustomKeyId o CustomKeyName (pero no ambos) para limitar la respuesta a almacenes de claves personalizados determinados. El ConnectionState valor de DISCONNECTED indica que este almacén de AWS CloudHSM claves de ejemplo no está conectado a su AWS CloudHSM clúster.

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

Eliminar un almacén de AWS CloudHSM claves

Al eliminar un almacén de AWS CloudHSM claves, AWS KMS elimina todos los metadatos sobre el almacén de AWS CloudHSM claves de KMS, incluida la información sobre su asociación con

un AWS CloudHSM clúster. Esta operación no afecta al AWS CloudHSM clúster HSMs, ni a sus usuarios. Puede crear un nuevo almacén de AWS CloudHSM claves asociado al mismo AWS CloudHSM clúster, pero no puede deshacer la operación de eliminación.

Solo puede eliminar un almacén de AWS CloudHSM claves que esté desconectado de su AWS CloudHSM clúster y que no contenga ninguno AWS KMS keys. Antes de eliminar un almacén de claves personalizado, haga lo siguiente.

- Compruebe que no necesitará nunca utilizar ninguna de las claves KMS del almacén de claves para ninguna [operación criptográfica](#). A continuación, [programe la eliminación](#) de todas las claves KMS del almacén de claves. Para obtener ayuda para encontrar las claves de KMS en un almacén de AWS CloudHSM claves, consulte [Busque las claves KMS en un almacén de AWS CloudHSM claves](#).
- Confirme que se han eliminado todas las claves KMS. Para ver las claves KMS de un almacén de AWS CloudHSM claves, consulte [the section called “Identifique las claves de KMS en AWS CloudHSM almacenes de claves”](#).
- [Desconecte el almacén de AWS CloudHSM claves](#) de su AWS CloudHSM clúster.

En lugar de eliminar el almacén de AWS CloudHSM claves, considere la posibilidad de [desconectarlo](#) del AWS CloudHSM clúster asociado. Mientras un almacén de AWS CloudHSM claves esté desconectado, puede administrar el almacén de AWS CloudHSM claves y su AWS KMS keys. Sin embargo, no puede crear ni usar claves KMS en el almacén de AWS CloudHSM claves. Puede volver a conectar el almacén de AWS CloudHSM claves en cualquier momento.

Elimina tu almacén de AWS CloudHSM claves

Puede eliminar el almacén de AWS CloudHSM claves en la AWS KMS consola o mediante esta [DeleteCustomKeyStore](#) operación.

Uso de la AWS KMS consola

Para eliminar un almacén de AWS CloudHSM claves del Consola de administración de AWS, comience por seleccionarlo en la AWS CloudHSM página Almacenes de claves personalizados.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.

3. En el panel de navegación, elija Almacenes de claves personalizados, Almacenes de claves de AWS CloudHSM .
4. Busque la fila que representa el almacén de AWS CloudHSM claves que desea eliminar. Si el estado de conexión del almacén de AWS CloudHSM claves no es Desconectado, debe [desconectar el almacén de AWS CloudHSM claves](#) antes de eliminarlo.
5. En el menú Key store actions (Acciones del almacén de claves), seleccione Delete (Eliminar).

Cuando se complete la operación, aparecerá un mensaje de éxito y el almacén de AWS CloudHSM claves ya no aparecerá en la lista de almacenes de claves. Si la operación da error, aparecerá un mensaje de error donde se describe el problema y se explica cómo resolverlo. Si necesita más ayuda, consulte [Resolver problemas de un almacén de claves personalizado](#).

Uso de la AWS KMS API

Para eliminar un almacén de AWS CloudHSM claves, utilice la [DeleteCustomKeyStore](#) operación. Si la operación se realiza correctamente, AWS KMS devuelve una respuesta HTTP 200 y un objeto JSON sin propiedades.

Para empezar, compruebe que el almacén de AWS CloudHSM claves no contiene ninguna AWS KMS keys. No puede eliminar un almacén de claves personalizado que contenga claves KMS. El primer comando de ejemplo utiliza [ListKeys](#) y [DescribeKey](#) busca AWS KMS keys en el almacén de AWS CloudHSM claves con el ID de almacén de claves ***cks-1234567890abcdef0*** personalizado del ejemplo. En ese caso, el comando no devuelve ninguna clave KMS. Si es así, utilice la [ScheduleKeyDeletion](#) operación para programar la eliminación de cada una de las claves del KMS.

Bash

```
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
```

PowerShell

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyId -eq  
'cks-1234567890abcdef0'
```

A continuación, desconecte el almacén de AWS CloudHSM claves. Este comando de ejemplo utiliza la [DisconnectCustomKeyStore](#) operación para desconectar un almacén de AWS CloudHSM claves de su AWS CloudHSM clúster. Antes de ejecutar este comando, reemplace el ID del almacén de claves personalizado de ejemplo por uno válido.

Bash

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Disconnect-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Una vez desconectado el almacén de claves personalizado, puede utilizar la [DeleteCustomKeyStore](#) operación para eliminarlo.

Bash

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Remove-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Resolver problemas de un almacén de claves personalizado

AWS CloudHSM los almacenes de llaves están diseñados para estar disponibles y ser resistentes. Sin embargo, hay algunas condiciones de error que puede que tengas que reparar para que tu almacén de AWS CloudHSM claves siga funcionando.

Temas

- [¿Cómo arreglar las claves KMS no disponibles?](#)
- [¿Cómo arreglar una clave KMS que produce error?](#)
- [Cómo arreglar un error de conexión](#)
- [Cómo responder ante un error de operación criptográfica](#)

- [Cómo arreglar las credenciales de kmsuser no válidas](#)
- [Cómo eliminar material de claves huérfano](#)
- [¿Cómo recuperar el material de claves eliminado de una clave KMS?](#)
- [Cómo iniciar sesión como kmsuser](#)

¿Cómo arreglar las claves KMS no disponibles?

El [estado clave](#) AWS KMS keys de un almacén de AWS CloudHSM claves suele ser `Enabled`. Como todas las claves de KMS, el estado de las claves cambia al deshabilitar las claves de KMS en un almacén de AWS CloudHSM claves o al programar su eliminación. Sin embargo, a diferencia de otras claves KMS, las claves KMS de un almacén de claves personalizado también pueden tener el [estado de clave](#) `Unavailable`.

El estado de clave `Unavailable` indica que la clave de KMS está en un almacén de claves personalizado que se ha [desconectado](#) de forma intencional y que los intentos de conectarlo de nuevo han fallado. Mientras una clave KMS no esté disponible, puede consultarla y administrarla, pero no puede usarla en [operaciones criptográficas](#).

Para buscar el estado de clave de una clave KMS en la página Customer managed keys (Claves administradas por el cliente) consulte el campo Status (Estado) de la clave KMS. O bien, utilice la [DescribeKey](#) operación y visualice el `KeyState` elemento en la respuesta. Para obtener más información, consulte [Identificación y visualización de claves](#).

Las claves KMS en un almacén de claves personalizado desconectado tendrá el estado de clave `Unavailable` o `PendingDeletion`. Las claves de KMS programadas para su eliminación de un almacén de claves personalizado tienen el estado de clave `Pending Deletion`, incluso si el almacén de claves personalizado está desconectado. Esto permite cancelar la eliminación programada de la clave sin volver a conectar el almacén de claves personalizado.

Para arreglar una clave KMS no disponible, [vuelva a conectar el almacén de claves personalizado](#). Después de volver a conectar el almacén de claves personalizado, el estado de clave de las claves KMS en el almacén de claves personalizado se restaura automáticamente al estado anterior, como `Enabled` o `Disabled`. Las claves KMS pendientes de eliminación seguirán teniendo el estado `PendingDeletion`. Sin embargo, si el problema persiste, [habilitar y desactivar una clave KMS no disponible](#) no cambia su estado de clave. La acción de habilitar o desactivar solo será efectiva cuando la clave esté disponible.

Si desea ayuda con las conexiones que dan error, consulte [Cómo arreglar un error de conexión](#).

¿Cómo arreglar una clave KMS que produce error?

Los problemas relacionados con la creación y el uso de las claves de KMS en AWS CloudHSM los almacenes de claves pueden deberse a un problema con el almacén de AWS CloudHSM claves, el AWS CloudHSM clúster asociado, la clave de KMS o su material de claves.

Cuando un almacén de AWS CloudHSM claves se desconecta de su AWS CloudHSM clúster, el estado de las claves de KMS en el almacén de claves personalizado es el mismo `Unavailable`. Todas las solicitudes para crear claves de KMS en un almacén de AWS CloudHSM claves desconectado devuelven una `CustomKeyStoreInvalidStateException` excepción. Todas las solicitudes para cifrar, descifrar, volver a cifrar o generar claves de datos devuelve una excepción `KMSInvalidStateException`. Para solucionar el problema, [vuelva a conectar el almacén de AWS CloudHSM claves](#).

Sin embargo, sus intentos de utilizar una clave KMS en un almacén de AWS CloudHSM claves para [operaciones criptográficas](#) pueden fallar incluso si su estado de clave es `Enabled` y el estado de conexión del almacén de AWS CloudHSM claves es el mismo. `Connected` Esto puede deberse a una de las siguientes condiciones.

- Puede que se haya eliminado el material de claves para la clave KMS del clúster de AWS CloudHSM asociado. Para investigar, [busque el identificador de claves](#) del material de claves para una clave KMS y, si es necesario, intente [recuperar el material de claves](#).
- Todas HSMs se eliminaron del AWS CloudHSM clúster asociado al almacén de AWS CloudHSM claves. Para usar una clave KMS en un almacén de AWS CloudHSM claves en una operación criptográfica, su AWS CloudHSM clúster debe contener al menos un HSM activo. Para comprobar el número y el estado HSMs de un AWS CloudHSM clúster, [utilice la AWS CloudHSM consola o la `DescribeClusters` operación](#). Para añadir un HSM al clúster, utilice la AWS CloudHSM consola o la [`CreateHsm` operación](#).
- Se eliminó el AWS CloudHSM clúster asociado al almacén de AWS CloudHSM claves. Para resolver el problema, [cree un clúster a partir de una copia de seguridad](#) relacionada con el clúster original, como una copia de seguridad de un clúster original o una copia de seguridad utilizada para crear el clúster original. A continuación, [edite el ID del clúster](#) en la configuración del almacén de claves personalizado. Para obtener instrucciones, consulte [¿Cómo recuperar el material de claves eliminado de una clave KMS?](#).
- El AWS CloudHSM clúster asociado al almacén de claves personalizado no tenía ninguna sesión de PKCS #11 disponible. Esto ocurre, por lo general, cuando se producen períodos de ráfagas de tráfico elevado, en los que se necesitan sesiones adicionales para dar servicio al tráfico. Para

responder a una `KMSInternalException` con un mensaje de error sobre las sesiones de PKCS#11, regrese y vuelva a intentar la solicitud.

Cómo arreglar un error de conexión

Si intenta [conectar un almacén de AWS CloudHSM claves](#) a su AWS CloudHSM clúster, pero se produce un error en la operación, el estado de conexión del almacén de AWS CloudHSM claves cambia a `FAILED`. Para averiguar el estado de conexión de un almacén de AWS CloudHSM claves, utilice la AWS KMS consola o la [DescribeCustomKeyStores](#) operación.

Algunos intentos de conexión producen un error rápidamente debido a errores de configuración del clúster detectados fácilmente. En este caso, el estado de la conexión aún es `DISCONNECTED`. Estos errores devuelven un mensaje de error o una [excepción](#) que explica por qué el intento produjo un error. Revise la descripción de la excepción y [los requisitos del clúster](#), solucione el problema, [actualice el almacén de AWS CloudHSM claves](#), si es necesario, e intente conectarse de nuevo.

Cuando el estado de la conexión sea `FAILED`, ejecute la [DescribeCustomKeyStores](#) operación y observe el `ConnectionErrorCode` elemento en la respuesta.

Note

Cuando el estado de conexión de un almacén de AWS CloudHSM claves es `FAILED`, debe [desconectar el almacén de AWS CloudHSM claves](#) antes de intentar volver a conectarlo. No puede conectar un almacén de AWS CloudHSM claves con un estado de `FAILED` conexión.

- `CLUSTER_NOT_FOUND` indica que AWS KMS no puede encontrar un AWS CloudHSM clúster con el ID de clúster especificado. Esto puede deberse a que se ha proporcionado un ID de clúster erróneo a una operación de API o que se ha eliminado el clúster y no se ha reemplazado. Para corregir este error, compruebe el ID del clúster, por ejemplo, mediante la AWS CloudHSM consola o la [DescribeClusters](#) operación. Si se ha eliminado el clúster, [cree un clúster a partir de una copia de seguridad reciente](#) del original. A continuación, [desconecte el almacén de AWS CloudHSM claves](#), [edite la AWS CloudHSM configuración del ID del clúster del almacén](#) de claves y [vuelva a conectar el almacén de AWS CloudHSM claves](#) al clúster.
- `INSUFFICIENT_CLOUDHSM_HSMS` indica que el AWS CloudHSM clúster asociado no contiene ninguno de los HSMs. Para conectarse, el clúster debe tener al menos un HSM. Para encontrar el número de HSMs en el clúster, utilice la [DescribeClusters](#) operación. Para solucionar este error,

[agregue al menos un HSM](#) al clúster. Si agregas varias HSMs, es mejor crearlas en distintas zonas de disponibilidad.

- `INSUFFICIENT_FREE_ADDRESSES_IN_SUBNET` indica que no se ha AWS KMS podido conectar el almacén de AWS CloudHSM claves a su AWS CloudHSM clúster porque al menos una [subred privada asociada al clúster](#) no tiene ninguna dirección IP disponible. Una conexión a un almacén de AWS CloudHSM claves requiere una dirección IP libre en cada una de las subredes privadas asociadas, aunque es preferible tener dos.

[No se pueden añadir direcciones IP](#) (bloques CIDR) a una subred existente. Si es posible, se deben mover o eliminar otros recursos que estén utilizando las direcciones IP de la subred, como instancias de EC2 no utilizadas o interfaces de red elásticas. De lo contrario, puede [crear un clúster a partir de una copia de seguridad reciente](#) del AWS CloudHSM clúster con subredes privadas nuevas o existentes que tengan [más espacio libre de direcciones](#). A continuación, para asociar el nuevo clúster a su almacén de AWS CloudHSM claves, [desconecte el almacén de claves personalizado](#), [cambie el ID de clúster](#) del almacén de AWS CloudHSM claves por el ID del nuevo clúster e intente conectarse de nuevo.

 Tip

Para evitar [restablecer la kmsuser contraseña](#), utilice la copia de seguridad más reciente del AWS CloudHSM clúster.

- `INTERNAL_ERROR` indica que no se AWS KMS pudo completar la solicitud debido a un error interno. Intente realizar de nuevo la solicitud. En el caso de `ConnectCustomKeyStore` las solicitudes, desconecte el almacén de AWS CloudHSM claves antes de intentar conectarse de nuevo.
- `INVALID_CREDENTIALS` indica que AWS KMS no puede iniciar sesión en el AWS CloudHSM clúster asociado porque no tiene la contraseña de `kmsuser` cuenta correcta. Si desea ayuda para solucionar este error, consulte [Cómo arreglar las credenciales de kmsuser no válidas](#).
- `NETWORK_ERRORS` suele hacer referencia a problemas temporales de red. [Desconecte el almacén de AWS CloudHSM claves](#), espere unos minutos e intente conectarse de nuevo.
- `SUBNET_NOT_FOUND` indica que se ha eliminado al menos una subred de la configuración del AWS CloudHSM clúster. Si AWS KMS no puede encontrar todas las subredes de la configuración del clúster, se produce un error al intentar conectar el almacén de AWS CloudHSM claves al AWS CloudHSM clúster.

Para corregir este error, [cree un clúster a partir de una copia de seguridad reciente](#) del mismo AWS CloudHSM clúster. (Este proceso crea una nueva configuración de clúster con una VPC y subredes privadas). Compruebe que el nuevo clúster cumple los [requisitos de un almacén de claves personalizado](#) y anote el nuevo ID del clúster. A continuación, para asociar el nuevo clúster a su almacén de AWS CloudHSM claves, [desconecte el almacén de claves personalizado](#), [cambie el ID de clúster](#) del almacén de AWS CloudHSM claves por el ID del nuevo clúster e intente conectarse de nuevo.

 Tip

Para evitar [restablecer la kmsuser contraseña](#), utilice la copia de seguridad más reciente del AWS CloudHSM clúster.

- USER_LOCKED_OUT indica que la [cuenta del usuario de criptografía \(CU\) kmsuser](#) está bloqueada del clúster de AWS CloudHSM asociado porque se han realizado demasiados intentos fallidos de introducción de contraseña. Si desea ayuda para solucionar este error, consulte [Cómo arreglar las credenciales de kmsuser no válidas](#).

Para corregir este error, [desconecte el almacén de AWS CloudHSM claves](#) y utilice el comando [user change-password](#) de la CLI de CloudHSM para cambiar la contraseña de la cuenta. kmsuser A continuación, [edite la configuración de la contraseña de kmsuser](#) para el almacén de claves personalizado e intente conectarlo de nuevo. Para obtener ayuda, utilice el procedimiento descrito en el tema [Cómo arreglar las credenciales de kmsuser no válidas](#).

- USER_LOGGED_IN indica que la cuenta kmsuser CU ha iniciado sesión en el clúster asociado. AWS CloudHSM Esto AWS KMS impide rotar la contraseña de la kmsuser cuenta e iniciar sesión en el clúster. Para corregir este error, cierre la sesión del CU kmsuser del clúster. Si ha cambiado la kmsuser contraseña para iniciar sesión en el clúster, también debe actualizar el valor de la contraseña del almacén de AWS CloudHSM claves. Para obtener ayuda, consulte [Cómo cerrar sesión y volver a conectar](#).
- USER_NOT_FOUND indica que AWS KMS no puede encontrar una cuenta kmsuser CU en el AWS CloudHSM clúster asociado. Para corregir este error, [cree una cuenta kmsuser CU](#) en el clúster y, a continuación, [actualice el valor de la contraseña del almacén](#) de AWS CloudHSM claves del almacén de claves. Para obtener ayuda, consulte [Cómo arreglar las credenciales de kmsuser no válidas](#).

Cómo responder ante un error de operación criptográfica

Una operación criptográfica que utiliza una clave de KMS en un almacén de claves personalizado puede dar un error `KMSInvalidStateException`. Los siguientes mensajes de error pueden acompañar al error `KMSInvalidStateException`.

KMS no puede comunicarse con el clúster de CloudHSM. Esto puede ser un problema de red transitorio. Si ve este error repetidamente, compruebe que las reglas de red ACLs y del grupo de seguridad de la VPC del AWS CloudHSM clúster sean correctas.

- Aunque se trata de un error HTTPS 400, puede deberse a problemas de red transitorios. Para responder, comience por volver a intentar la solicitud. Sin embargo, si continúa fallando, examine la configuración de los componentes de red. Este error es probablemente causado por la configuración incorrecta de un componente de red, como una regla de firewall o una regla de grupo de seguridad de VPC que bloquea el tráfico saliente. Por ejemplo, el KMS no puede comunicarse con AWS CloudHSM los IPv6 clústeres existentes. Para obtener más información sobre los requisitos previos, consulte [Crear un almacén de AWS CloudHSM claves](#).

KMS no puede comunicarse con el AWS CloudHSM clúster porque el usuario `kmsuser` está bloqueado. Si ve este error repetidamente, desconecte el almacén de AWS CloudHSM claves y restablezca la contraseña de la cuenta `kmsuser`. Actualice la contraseña de `kmsuser` para el almacén de claves personalizado y pruebe a realizar la solicitud de nuevo.

- Este mensaje de error indica que la [cuenta del usuario de criptografía \(CU\) `kmsuser`](#) está bloqueada del clúster del AWS CloudHSM asociado porque se han realizado demasiados intentos fallidos de introducción de contraseña. Si desea ayuda para solucionar este error, consulte [Cómo desconectar e iniciar sesión](#).

Cómo arreglar las credenciales de **kmsuser** no válidas

Al [conectar un almacén de AWS CloudHSM claves](#), AWS KMS inicia sesión en el AWS CloudHSM clúster asociado como [usuario `kmsuser` criptográfico](#) (CU). Permanece conectado hasta que se desconecte el almacén de AWS CloudHSM claves. La respuesta [DescribeCustomKeyStores](#)

muestra un `ConnectionState` de `FAILED` y un valor de `ConnectionErrorCode` de `INVALID_CREDENTIALS`, como se muestra en el siguiente ejemplo.

Si desconecta el almacén de AWS CloudHSM claves y cambia la `kmsuser` contraseña, AWS KMS no podrá iniciar sesión en el AWS CloudHSM clúster con las credenciales de la cuenta `kmsuser` CU. Como resultado, todos los intentos de conectar el almacén de AWS CloudHSM claves fallan. La respuesta `DescribeCustomKeyStores` muestra un `ConnectionState` de `FAILED` y un valor de `ConnectionErrorCode` de `INVALID_CREDENTIALS`, como se muestra en el siguiente ejemplo.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "INVALID_CREDENTIALS",
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
    }
  ],
}
```

Además, después de 5 intentos de iniciar sesión en el clúster con una contraseña incorrecta, AWS CloudHSM bloquea la cuenta del usuario. Para iniciar sesión en el clúster, deberá cambiar la contraseña de la cuenta.

Si AWS KMS recibe una respuesta de bloqueo cuando intenta iniciar sesión en el clúster como `kmsuser` CU, se produce un error en la solicitud de conexión del almacén de AWS CloudHSM claves. La [DescribeCustomKeyStores](#) respuesta incluye un valor `ConnectionState` de `FAILED` y un `ConnectionErrorCode` valor de `USER_LOCKED_OUT`, como se muestra en el siguiente ejemplo.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "USER_LOCKED_OUT",
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
    }
  ],
}
```

```
],  
}
```

Para reparar cualquiera de estas condiciones use el procedimiento siguiente.

1. [Desconecte el almacén de AWS CloudHSM claves](#).
2. Ejecute la [DescribeCustomKeyStores](#) operación y visualice el valor del `ConnectionErrorCode` elemento en la respuesta.
 - Si el valor de `ConnectionErrorCode` es `INVALID_CREDENTIALS`, determine la contraseña actual para la cuenta de `kmsuser`. Si es necesario, use el comando [user change-password](#) en la CLI de CloudHSM para establecer la contraseña con un valor conocido.
 - Si el valor de `ConnectionErrorCode` es `USER_LOCKED_OUT`, deberá usar el comando [user change-password](#) en la CLI de CloudHSM para modificar la contraseña de `kmsuser`.
3. [Edite la configuración de la contraseña de kmsuser](#) para que coincida con la contraseña de `kmsuser` del clúster. Esta acción le dice a AWS KMS qué contraseña debe usar para iniciar sesión en el clúster. No cambia la contraseña de `kmsuser` en el clúster.
4. [Conecte el almacén de claves personalizado](#).

Cómo eliminar material de claves huérfano

Tras programar la eliminación de una clave de KMS de un almacén de AWS CloudHSM claves, es posible que tenga que eliminar manualmente el material de claves correspondiente del AWS CloudHSM clúster asociado.

Al crear una clave de KMS en un almacén de AWS CloudHSM claves, AWS KMS crea los metadatos de la clave de KMS AWS KMS y genera el material de claves en el AWS CloudHSM clúster asociado. Al programar la eliminación de una clave de KMS de un almacén de AWS CloudHSM claves, tras el período de espera, se AWS KMS eliminan los metadatos de la clave de KMS. A continuación, AWS KMS hace todo lo posible por eliminar el material clave correspondiente del AWS CloudHSM clúster. El intento podría fallar si AWS KMS no se puede acceder al clúster, por ejemplo, cuando se desconecta del almacén de AWS CloudHSM claves o si se cambia la `kmsuser` contraseña. AWS KMS no intenta eliminar el material clave de las copias de seguridad del clúster.

AWS KMS informa de los resultados de su intento de eliminar el material clave del clúster en la entrada de `DeleteKey` eventos de sus AWS CloudTrail registros. Aparece en el elemento `BackingKeysDeletionStatus` del elemento `additionalEventData`, tal y como se muestra en

la siguiente entrada de ejemplo. La entrada también incluye el ARN de la clave KMS, el ID del AWS CloudHSM clúster y el ID (backing-key-id) del material clave.

```
{
  "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"
}
```

Notas

Los siguientes procedimientos utilizan la herramienta de línea de comandos AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI de CloudHSM reemplaza `key-handle` por `key-reference`.

El 1 de enero de 2025, AWS CloudHSM dejará de ser compatible con las herramientas de línea de comandos Client SDK 3, la utilidad de administración CloudHSM (CMU) y la utilidad de administración de claves (KMU). Para obtener más información sobre las diferencias entre las herramientas de línea de comandos Client SDK 3 y Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) en la Guía del usuario de AWS CloudHSM.

Los siguientes procedimientos muestran cómo eliminar el material clave huérfano del clúster asociado. AWS CloudHSM

1. Desconecte el almacén de AWS CloudHSM claves, si aún no está desconectado, inicie [sesión](#), como se explica en [Cómo desconectar e iniciar sesión](#).

Note

Mientras un almacén de claves personalizado esté desconectado, todos los intentos de crear claves KMS en el almacén de claves personalizado o de usar claves KMS existentes en operaciones criptográficas fallarán. Esta acción puede impedir que los usuarios almacenen y accedan a datos confidenciales.

2. Utilice el comando [key delete](#) de la CLI de CloudHSM para eliminar la clave HSMs del clúster.

Todas las entradas de CloudTrail registro para operaciones criptográficas con una clave KMS en un almacén de AWS CloudHSM claves incluyen un `additionalEventData` campo con una `y.customKeyStoreId backingKey` El valor devuelto en el campo `backingKeyId` es el atributo `id` de clave de CloudHSM. Recomendamos filtrar la operación de eliminación de claves `id` para eliminar el material clave huérfano que identificó en sus CloudTrail registros.

AWS CloudHSM reconoce el `backingKeyId` valor como un valor hexadecimal. Para filtrar por `id`, debe anexar el `backingKeyId` con `0x`. Por ejemplo, si está `backingKeyId` en su CloudTrail registro `1a2b3c45678abcdef`, debe filtrar por `0x1a2b3c45678abcdef`.

En el siguiente ejemplo, se elimina una clave HSMs del clúster. Se `backing-key-id` muestra en la entrada de CloudTrail registro. Antes de ejecutar este comando, reemplace el `backing-key-id` de ejemplo por uno válido de su cuenta.

```
aws-cloudhsm key delete --filter attr.id="0x<backing-key-id>"
{
  "error_code": 0,
  "data": {
    "message": "Key deleted successfully"
  }
}
```

3. Cierre la sesión y vuelva a conectar el almacén de AWS CloudHSM claves como se describe en [Cómo cerrar sesión y volver a conectar](#).

¿Cómo recuperar el material de claves eliminado de una clave KMS?

Si AWS KMS key se elimina el material clave de un, la clave de KMS no se puede utilizar y no se puede descifrar todo el texto cifrado con la clave de KMS. Esto puede suceder si el material clave de una clave KMS de un almacén de AWS CloudHSM claves se elimina del clúster asociado. AWS CloudHSM Sin embargo, el material de claves se puede recuperar.

Al crear una AWS KMS key (clave KMS) en un almacén de AWS CloudHSM claves, AWS KMS inicia sesión en el AWS CloudHSM clúster asociado y crea el material clave para la clave KMS. También cambia la contraseña por un valor que solo él conoce y permanece conectado mientras el almacén de AWS CloudHSM claves esté conectado. Como solo el propietario de la clave, es decir, la CU que creó la clave, puede eliminarla, es poco probable que la clave se borre HSMs accidentalmente.

Sin embargo, si el material clave de una clave de KMS se elimina HSMs de un clúster, el estado de la clave de KMS finalmente cambia a `UNAVAILABLE`. Si intenta usar la clave de KMS para una operación criptográfica, la operación da error con una excepción `KMSInvalidStateException`. Lo más importante es que todos los datos cifrados con la clave KMS no pueden descifrarse.

Puede recuperar el material de claves eliminado, bajo determinadas circunstancias, [creando un clúster a partir de una copia de seguridad](#) que contenga el material de claves. Esta estrategia funciona únicamente si se creó al menos una copia de seguridad mientras existió la clave y antes de que se eliminara.

Utilice el siguiente proceso para recuperar el material de claves.

1. Busque una copia de seguridad del clúster que incluya el material de claves. La copia de seguridad también debe incluir todos los usuarios y claves necesarios para respaldar el clúster y sus datos cifrados.

Utilice la [DescribeBackups](#) operación para enumerar las copias de seguridad de un clúster. A continuación, use la marca temporal de la copia de seguridad para seleccionar una copia de seguridad. Para limitar la salida al clúster asociado al almacén de AWS CloudHSM claves, utilice el `Filters` parámetro, como se muestra en el siguiente ejemplo.

```
$ aws cloudhsmv2 describe-backups --filters clusterIds=<cluster ID>
{
  "Backups": [
    {
      "ClusterId": "cluster-1a23b4cdefg",
      "BackupId": "backup-9g87f6edcba",
      "CreateTimestamp": 1536667238.328,
      "BackupState": "READY"
    },
    ...
  ]
}
```

2. [Crear un clúster a partir de la copia de seguridad seleccionada](#). Compruebe que la copia de seguridad contiene la clave eliminada y otros usuarios y claves necesarios para el clúster.
3. [Desconecte el almacén de AWS CloudHSM claves](#) para poder editar sus propiedades.
4. [Edite el ID de clúster](#) del almacén de AWS CloudHSM claves. Escriba el ID del clúster creado a partir de la copia de seguridad. Puesto que el clúster comparte un historial de copias de seguridad con el clúster original, el nuevo ID del clúster debería ser válido.
5. [Vuelva a conectar el almacén de AWS CloudHSM claves](#).

Cómo iniciar sesión como **kmsuser**

Para crear y administrar el material clave del AWS CloudHSM clúster para su almacén de AWS CloudHSM claves, AWS KMS utilice la [cuenta de usuario kmsuser criptográfico \(CU\)](#). Al [crear el almacén de AWS CloudHSM claves](#), debe crear la cuenta [kmsuser CU](#) en el clúster y proporcionar su contraseña. AWS KMS

En general, AWS KMS administra la `kmsuser` cuenta. Sin embargo, para algunas tareas, debe desconectar el almacén de AWS CloudHSM claves, iniciar sesión en el clúster como `kmsuser` CU y utilizar la [interfaz de línea de comandos \(CLI\) de CloudHSM](#).

 Note

Mientras un almacén de claves personalizado esté desconectado, todos los intentos de crear claves KMS en el almacén de claves personalizado o de usar claves KMS existentes en operaciones criptográficas fallarán. Esta acción puede impedir que los usuarios almacenen y accedan a datos confidenciales.

En este tema se explica cómo [desconectar el almacén de AWS CloudHSM claves e iniciar sesión como `kmsuser`](#), ejecutar la herramienta de línea de AWS CloudHSM comandos, [cerrar sesión y volver a conectar AWS CloudHSM](#) el almacén de claves.

Temas

- [Cómo desconectar e iniciar sesión](#)
- [Cómo cerrar sesión y volver a conectar](#)

Cómo desconectar e iniciar sesión

Siga el siguiente procedimiento cada vez que deba iniciar sesión en un clúster asociado como usuario de criptografía de `kmsuser`.

 Notas

Los siguientes procedimientos utilizan la herramienta de línea de comandos AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI de CloudHSM reemplaza `key-handle` por `key-reference`.

El 1 de enero de 2025, AWS CloudHSM dejará de ser compatible con las herramientas de línea de comandos Client SDK 3, la utilidad de administración CloudHSM (CMU) y la utilidad de administración de claves (KMU). Para obtener más información sobre las diferencias entre las herramientas de línea de comandos Client SDK 3 y Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) en la Guía del usuario de AWS CloudHSM.

1. Desconecte el almacén de AWS CloudHSM claves, si aún no está desconectado. Puede usar la AWS KMS consola o la AWS KMS API.

Mientras su AWS CloudHSM clave esté conectada, AWS KMS iniciará sesión como `kmsuser`. Esto evita que inicie sesión como `kmsuser` o que cambie la contraseña de `kmsuser`.

Por ejemplo, este comando se utiliza [DisconnectCustomKeyStore](#) para desconectar un almacén de claves de ejemplo. Sustituya el ID del almacén de AWS CloudHSM claves del ejemplo por uno válido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

2. Ejecute el comando `login` para iniciar sesión como administrador. Utilice los procedimientos descritos en la sección [Uso de la CLI de CloudHSM](#) de la Guía del usuario de AWS CloudHSM .

```
aws-cloudhsm > login --username admin --role admin
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

3. Utilice el comando [user change-password](#) de la CLI de CloudHSM para cambiar la contraseña de `kmsuser` la cuenta por una que conozca. (AWS KMS rota la contraseña al conectar el almacén de claves). AWS CloudHSM La contraseña debe contener entre 7 y 32 caracteres alfanuméricos, distingue entre mayúsculas y minúsculas, y no puede tener caracteres especiales.
4. Inicie sesión como `kmsuser` con la contraseña que haya establecido. Si desea leer instrucciones detalladas, consulte la sección [Uso de la CLI de CloudHSM](#) de la Guía del usuario de AWS CloudHSM .

```
aws-cloudhsm > login --username kmsuser --role crypto-user
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "kmsuser",

```

```
"role": "crypto-user"
}
}
```

Cómo cerrar sesión y volver a conectar

Utilice el siguiente procedimiento cada vez que necesite cerrar sesión como usuario de criptografía de kmsuser y volver a conectar su almacén de claves.

Notas

Los siguientes procedimientos utilizan la herramienta de línea de comandos AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI de CloudHSM reemplaza key-handle por key-reference.

El 1 de enero de 2025, AWS CloudHSM dejará de ser compatible con las herramientas de línea de comandos Client SDK 3, la utilidad de administración CloudHSM (CMU) y la utilidad de administración de claves (KMU). Para obtener más información sobre las diferencias entre las herramientas de línea de comandos Client SDK 3 y Client SDK 5, consulte [Migrate from Client SDK 3 CMU and KMU to Client SDK 5 CloudHSM CLI](#) en la Guía del usuario de AWS CloudHSM .

1. Realice la tarea y, a continuación, utilice el comando [logout](#) de la CLI de CloudHSM para cerrar sesión. Si no cierra sesión, los intentos de volver a conectar el almacén de claves fallarán. AWS CloudHSM

```
aws-cloudhsm logout
{
  "error_code": 0,
  "data": "Logout successful"
}
```

2. [Edite la configuración de la contraseña de kmsuser](#) para el almacén de claves personalizado.

Esto indica AWS KMS la contraseña actual kmsuser del clúster. Si omite este paso, no AWS KMS podrá iniciar sesión en el clúster como tal kmsuser y todos los intentos de volver a conectar el almacén de claves personalizado fallarán. Puede utilizar la AWS KMS consola o el KeyStorePassword parámetro de la [UpdateCustomKeyStore](#) operación.

Por ejemplo, este comando indica AWS KMS que la contraseña actual es `tempPassword`. Reemplace la contraseña de ejemplo por una real.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-store-password tempPassword
```

3. Vuelva a conectar el almacén de AWS KMS claves a su AWS CloudHSM clúster. Sustituya el ID del almacén de AWS CloudHSM claves de ejemplo por uno válido. Durante el proceso de conexión, AWS KMS cambia la `kmsuser` contraseña por un valor que solo él conozca.

La [ConnectCustomKeyStore](#) operación vuelve rápidamente, pero el proceso de conexión puede tardar mucho tiempo. La respuesta inicial no indica que el proceso de conexión se haya realizado correctamente.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

4. Utilice la [DescribeCustomKeyStores](#) operación para comprobar que el almacén de AWS CloudHSM claves está conectado. Sustituya el ID del almacén de AWS CloudHSM claves del ejemplo por uno válido.

En este ejemplo, el campo de estado de la conexión muestra que el almacén de AWS CloudHSM claves ya 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"
    }
  ],
}
```

Almacenes de claves externos

Los almacenes de claves externos le permiten proteger sus AWS recursos mediante claves criptográficas externas. Esta función avanzada está diseñada para cargas de trabajo reguladas que debe proteger con claves de cifrado almacenadas en un sistema de administración de claves externo controlado por usted. Los almacenes de claves externos respaldan el [compromiso de soberanía AWS digital](#) y te dan el control soberano sobre tus datos AWS, incluida la posibilidad de cifrarlos con material clave que te pertenezca y que controle fuera de él. AWS

Un almacén de claves externo es un [almacén de claves personalizado](#) respaldado por un administrador de claves externo que usted posee y administra de AWS forma externa. El administrador de claves externo puede ser un módulo de seguridad de hardware físico o virtual (HSMs), o cualquier sistema basado en hardware o software capaz de generar y utilizar claves criptográficas. El administrador de claves externo realiza las operaciones de cifrado y descifrado que utilizan una clave KMS en un almacén de claves externo utilizando el material de claves criptográficas, una función conocida como guardar sus propias claves (HYOKs).

AWS KMS nunca interactúa directamente con su administrador de claves externo y no puede crear, ver, administrar ni eliminar sus claves. En su lugar, solo AWS KMS interactúa con el software de [proxy de almacenamiento de claves externo](#) (proxy XKS) que usted proporcione. El proxy del almacén de claves externo interviene en todas las comunicaciones entre AWS KMS y el administrador de claves externo. Transmite todas las solicitudes AWS KMS a su administrador de claves externo y transmite las respuestas de su administrador de claves externo a AWS KMS. El proxy del almacén de claves externo también traduce las solicitudes genéricas AWS KMS a un formato específico del proveedor que su administrador de claves externo pueda entender, lo que le permite utilizar almacenes de claves externos con administradores de claves de diversos proveedores.

Puede utilizar las claves de KMS en un almacén de claves externo para el cifrado del cliente, incluso con el [AWS Encryption SDK](#). Sin embargo, los almacenes de claves externos son un recurso importante para el cifrado del lado del servidor, ya que le permiten proteger sus AWS recursos en múltiples ubicaciones. Servicios de AWS con sus claves criptográficas fuera de ellas. AWS Servicios de AWS que admiten [claves administradas por el cliente](#) para el cifrado simétrico también admiten claves KMS en un almacén de claves externo. Para obtener más información sobre el soporte, consulte la [Integración de servicios de AWS](#).

Los almacenes de claves externos permiten utilizarlos AWS KMS para cargas de trabajo reguladas, en las que las claves de cifrado deben almacenarse y utilizarse fuera de ellas. AWS Sin embargo,

representan una desviación importante del modelo estándar de responsabilidad compartida y requieren cargas operativas adicionales. Para la mayoría de los clientes, el mayor riesgo para la disponibilidad y la latencia superará los beneficios de seguridad de los almacenes de claves externos.

Los almacenes de claves externos le permiten controlar la raíz de la confianza. Los datos cifrados con las claves de KMS de su almacén de claves externo solo se pueden descifrar mediante el administrador de claves externo que usted controle. Si revoca temporalmente el acceso a su administrador de claves externo, por ejemplo, desconectando el almacén de claves externo o desconectando su administrador de claves externo del proxy del almacén de claves externo, AWS pierde todo el acceso a sus claves criptográficas hasta que lo restaure. Durante ese intervalo, el texto cifrado encriptado con las claves de KMS no se puede descifrar. Si revoca permanentemente el acceso al administrador de claves externo, todo el texto cifrado encriptado con una clave de KMS en su almacén de claves externo no se podrá recuperar. Las únicas excepciones son los AWS servicios que almacenan en caché brevemente las claves de [datos protegidas por las claves](#) de KMS. Estas claves de datos seguirán en funcionamiento hasta que desactive el recurso o caduque la memoria caché. Para obtener más información, consulte [Cómo afectan las claves de KMS obsoletas a las claves de datos](#).

Los almacenes de claves externos desbloquean los pocos casos de uso para cargas de trabajo reguladas, en las que las claves de cifrado deben permanecer exclusivamente bajo su control y ser inaccesibles para ellas. AWS Sin embargo, se trata de un cambio importante en la forma en que opera la infraestructura basada en la nube y un cambio significativo en el modelo de responsabilidad compartida. Para la mayoría de las cargas de trabajo, la carga operativa adicional y los mayores riesgos para la disponibilidad y el rendimiento superarán los beneficios de seguridad de los almacenes de claves externos.

¿Necesito un almacén de claves externo?

Para la mayoría de los usuarios, el almacén de AWS KMS claves predeterminado, que está protegido por [módulos de seguridad de hardware validados por el FIPS 140-3 de nivel de seguridad 3](#), cumple con sus requisitos de seguridad, control y normativa. Los usuarios de almacenes de claves externos incurren en altos costos, cargas de mantenimiento y solución de problemas, además de riesgos para la latencia, la disponibilidad y la fiabilidad.

Si piensa en un almacén de claves externo, tómese un tiempo para comprender las alternativas, como un [almacén de AWS CloudHSM claves](#) respaldado por un AWS CloudHSM clúster del que sea

propietario y que usted administre, y claves de KMS con [material de claves importado](#) que genere usted mismo HSMs y que pueda eliminar de las claves de KMS cuando lo desee. En particular, la importación de material de clave con un intervalo de caducidad muy breve podría proporcionar un nivel de control similar sin los riesgos de rendimiento o disponibilidad.

Un almacén de claves externo puede ser la solución adecuada para su organización si tiene los siguientes requisitos:

- Debe utilizar claves criptográficas en su administrador de claves local o en un administrador de claves AWS que no esté bajo su control.
- Debe demostrar que sus claves criptográficas se conservan únicamente bajo su control fuera de la nube.
- Debe cifrar y descifrar mediante claves criptográficas con autorización independiente.
- El material de claves debe estar sujeto a una ruta de auditoría secundaria e independiente.

Si elige un almacén de claves externo, limite su uso a las cargas de trabajo que requieran protección con claves criptográficas fuera de AWS.

Modelo de responsabilidad compartida

Las claves KMS estándar utilizan material clave que se genera y se utiliza para su AWS KMS propiedad y administración. HSMs Usted establece las políticas de control de acceso en sus claves de KMS y las configura para Servicios de AWS que usen claves de KMS para proteger sus recursos. AWS KMS asume la responsabilidad de la seguridad, la disponibilidad, la latencia y la durabilidad del material clave de sus claves de KMS.

Las claves de KMS de los almacenes de claves externos se basan en el material de clave y las operaciones de su administrador de claves externo. Como tal, el equilibrio de responsabilidades cambia en su dirección. Usted es responsable de la seguridad, la fiabilidad, la durabilidad y el rendimiento de las claves criptográficas de su administrador de claves externo. AWS KMS es responsable de responder con prontitud a las solicitudes y de comunicarse con el proxy de su almacén de claves externo, así como de mantener nuestros estándares de seguridad. [Para garantizar que cada clave externa almacene el texto cifrado al menos con la misma seguridad que el AWS KMS texto cifrado estándar, AWS KMS primero cifra todo el texto sin formato con material AWS KMS clave específico de su clave KMS y, a continuación, lo envía a su administrador de claves externo para que lo cifre con su clave externa, un procedimiento conocido como doble cifrado.](#) Como

resultado, ni AWS KMS ni el propietario del material de clave externo pueden descifrar por sí solos el texto cifrado con doble cifrado.

Usted es responsable de mantener un administrador de claves externo que cumpla con sus estándares reglamentarios y de rendimiento. También es responsable de proporcionar y mantener un proxy del almacén de claves externo que cumpla con la [especificación de la API del proxy del almacén de claves externo de AWS KMS](#) y de garantizar la disponibilidad y la durabilidad del material de claves. También debe crear, configurar y mantener un almacén de claves externo. Cuando se produzcan errores causados por componentes que usted mantiene, debe estar preparado para identificarlos y resolverlos, de modo que los AWS servicios puedan acceder a sus recursos sin interrupciones indebidas. AWS KMS proporciona una [guía de solución](#) de problemas para ayudarle a determinar la causa de los problemas y las soluciones más probables.

Revisa las [CloudWatch métricas y dimensiones de Amazon](#) que AWS KMS registran los almacenes de claves externos. AWS KMS le recomienda encarecidamente que cree CloudWatch alarmas para supervisar su almacén de claves externo, de modo que pueda detectar los primeros signos de problemas operativos y de rendimiento antes de que se produzcan.

¿Qué está cambiando?

Los almacenes de claves externos solo admiten claves de KMS de cifrado simétrico. En su interior AWS KMS, las claves de KMS se utilizan y administran en un almacén de claves externo prácticamente de la misma manera que se administran otras [claves administradas por el cliente](#), lo que incluye la [configuración de políticas de control de acceso](#) y la [supervisión del uso de las claves](#). Para solicitar una operación criptográfica APIs con una clave de KMS en un almacén de claves externo, se utiliza lo mismo con los mismos parámetros que se utilizan para cualquier clave de KMS. El precio también es el mismo que el de las claves de KMS estándar. Para obtener más información, consulte [Claves KMS en un almacén de claves externo](#) y [Precios de AWS Key Management Service](#).

Sin embargo, con los almacenes de claves externos cambian los siguientes principios:

- Usted es responsable de la disponibilidad, la durabilidad y la latencia de las operaciones de claves.
- Usted es responsable de todos los costos de desarrollo, compra, operación y licencias de su sistema de administración de claves externo.
- Puede implementar la [autorización independiente](#) de todas las solicitudes desde AWS KMS su proxy de almacén de claves externo.
- Puede supervisar, auditar y registrar todas las operaciones de su proxy de almacén de claves externo y todas las operaciones de su administrador de claves externo relacionadas con AWS KMS las solicitudes.

¿Por dónde empiezo?

Para crear y administrar un almacén de claves externo, debe [elegir la opción de conectividad proxy del almacén de claves externo](#), [reunir los requisitos previos](#) y [crear y configurar el almacén de claves externo](#).

Cuotas

AWS KMS permite hasta [10 almacenes de claves personalizados](#) en cada Cuenta de AWS región, incluidos los almacenes de [AWS CloudHSM claves y los almacenes de claves externos](#), independientemente del estado de conexión. Además, hay cuotas de solicitudes de AWS KMS sobre el [uso de claves KMS en un almacén de claves externo](#).

Si eliges la [conectividad de proxy de VPC](#) para tu proxy de almacén de claves externo, es posible que también haya cuotas en los componentes necesarios VPCs, como las subredes y los balanceadores de carga de red. Para obtener más información sobre estas cuotas, utilice la [consola de Service Quotas](#).

Regiones

Para minimizar la latencia de la red, cree los componentes del almacén de claves externo en la Región de AWS más cercana a su [administrador de claves externo](#). Si es posible, elija una región con un tiempo de ida y vuelta (RTT) de la red de 35 milisegundos o menos.

Los almacenes de claves externos son compatibles Regiones de AWS en todos los países AWS KMS admitidos, excepto en China (Beijing) y China (Ningxia).

Características no admitidas

AWS KMS no admite las siguientes funciones en los almacenes de claves personalizadas.

- [Claves de KMS asimétricas](#)
- [Claves KMS HMAC](#)
- [Claves KMS con material de claves importado](#)
- [Rotación automática de claves](#)
- [Claves de varias regiones](#)

Más información:

- [Anunciamos el almacén de claves externo de AWS KMS](#) en el blog de noticias de AWS .

Conceptos del almacén de claves externo

Conozca los términos y conceptos básicos que se utilizan en almacenes de claves externos.

Almacén de claves externo

Un almacén de claves externo es un [almacén de claves AWS KMS personalizado](#) respaldado por un administrador de claves externo al AWS que usted posee y administra. Cada clave de KMS de un almacén de claves externo está asociada a una [clave externa](#) del administrador de claves externo. Cuando utiliza una clave de KMS en un almacén de claves externo para el cifrado o el descifrado, la operación se realiza en el administrador de claves externo mediante su clave externa, una función conocida como Guardar sus propias claves (HYOK). Esta función está diseñada para organizaciones que deben mantener las claves criptográficas en su propio administrador de claves externo.

Los almacenes de claves externos garantizan que las claves criptográficas y las operaciones que protegen sus AWS recursos permanezcan en el administrador de claves externo que está bajo su control. AWS KMS envía solicitudes a su administrador de claves externo para cifrar y descifrar datos, pero AWS KMS no puede crear, eliminar ni administrar ninguna clave externa. Todas las solicitudes AWS KMS a su administrador de claves externo están mediadas por un componente de software [proxy de almacenamiento de claves externo](#) que usted suministra, posee y administra.

AWS Los servicios que admiten [claves administradas por el AWS KMS cliente](#) pueden usar las claves KMS de su almacén de claves externo para proteger sus datos. Como resultado, sus datos se protegen en última instancia mediante sus claves mediante las operaciones de cifrado de su administrador de claves externo.

Las claves de KMS de un almacén de claves externo tienen modelos de confianza, [acuerdos de responsabilidad compartida](#) y expectativas de rendimiento fundamentalmente diferentes a los de las claves de KMS estándar. Con los almacenes de claves externos, usted es responsable de la seguridad e integridad del material de clave y de las operaciones criptográficas. La disponibilidad y la latencia de las claves de KMS en un almacén de claves externo se ven afectadas por el equipo, el software, los componentes de red y la distancia entre AWS KMS y el administrador de claves externo. También es probable que incurra en costes adicionales para el administrador de claves externo y para la infraestructura de redes y equilibrio de carga que necesita para comunicarse con su administrador de claves externo AWS KMS

Puede utilizar su almacén de claves externo como parte de una estrategia más abarcativa de protección de datos. Para cada AWS recurso que proteja, puede decidir cuáles requieren una clave KMS en un almacén de claves externo y cuáles pueden protegerse mediante una clave KMS estándar. Esto le brinda la flexibilidad de elegir claves de KMS para clasificaciones de datos, aplicaciones o proyectos específicos.

Administrador de claves externo

Un administrador de claves externo es un componente ajeno a AWS que puede generar claves simétricas AES de 256 bits y realizar un cifrado y descifrado simétricos. El administrador de claves externo para un almacén de claves externo puede ser un módulo de seguridad de hardware (HSM) físico, un HSM virtual o un administrador de claves de software con o sin un componente de HSM. Puede estar ubicado en cualquier lugar fuera de AWS, incluso en sus instalaciones, en un centro de datos local o remoto o en cualquier nube. Su almacén de claves externo puede estar respaldado por un único administrador de claves externo o por varias instancias de administrador de claves relacionadas que compartan claves criptográficas, como un clúster de HSM. Los almacenes de claves externos están diseñados para dar soporte a una variedad de administradores externos de diferentes proveedores. Para obtener más información sobre la conexión a su administrador de claves externo, consulte [Elección de una opción de conectividad proxy del almacén de claves externo](#).

Clave externa

Cada clave de KMS de un almacén de claves externo está asociada a una clave criptográfica del [administrador de claves externo](#) conocida como clave externa. Al cifrar o descifrar con una clave de KMS en su almacén de claves externo, la operación criptográfica se realiza en su [administrador de claves externo](#) utilizando su clave externa.

Warning

La clave externa es esencial para el funcionamiento de la clave de KMS. Si se pierde o se elimina la clave externa, el texto cifrado con la clave KMS asociada será irre recuperable.

Para los almacenes de claves externos, la clave externa debe ser una clave AES de 256 bits que esté habilitada y pueda realizar el cifrado y el descifrado. Para obtener información detallada sobre los requisitos de clave externa, consulte [Requisitos para una clave de KMS en un almacén de claves externo](#).

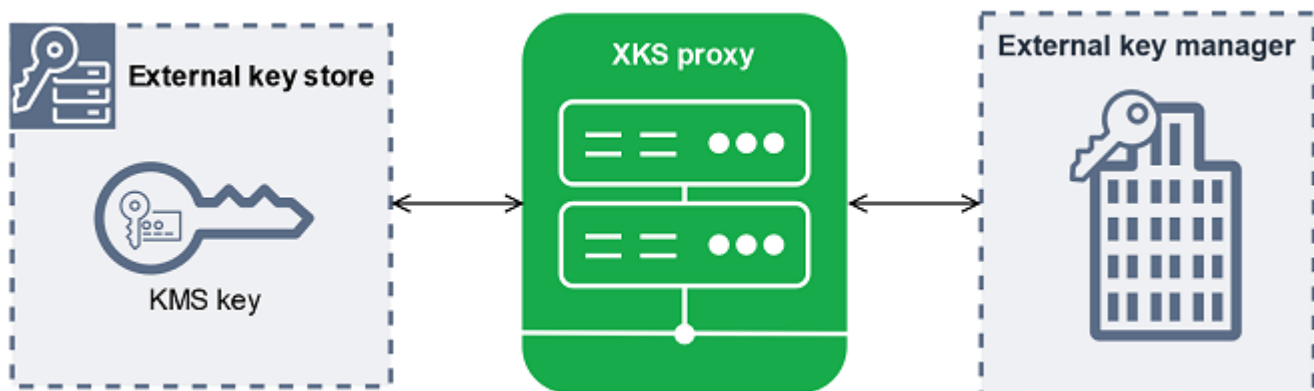
AWS KMS no puede crear, eliminar ni administrar ninguna clave externa. El material de claves criptográficas nunca sale de su administrador de claves externo. Cuando crea una clave de KMS en un almacén de claves externo, proporciona el ID de una clave externa (XksKeyId). No puede cambiar el ID de clave externa asociado a una clave de KMS, aunque el administrador de claves externo puede cambiar el material de clave asociado al ID de clave externa.

Además de la clave externa, una clave de KMS en un almacén de claves externo también contiene material de clave de AWS KMS. Los datos protegidos por la clave KMS se cifran primero AWS KMS con el material de la AWS KMS clave y, después, mediante el administrador de claves externo con la clave externa. Este proceso de [doble cifrado](#) garantiza que el texto cifrado protegido por su clave de KMS esté siempre tan seguro como el texto cifrado protegido solo por AWS KMS.

Muchas claves criptográficas tienen diferentes tipos de identificadores. Al crear una clave de KMS en un almacén de claves externo, proporcione el ID de la clave externa que el [proxy del almacén de claves externo](#) utiliza para hacer referencia a la clave externa. Si utiliza un identificador incorrecto, el intento de crear una clave de KMS en su almacén de claves externo no tendrá éxito.

Proxy del almacén de claves externo

El proxy de almacenamiento de claves externo («proxy XKS») es una aplicación de software propiedad y gestionada por el cliente que interviene en todas las comunicaciones entre AWS KMS el administrador de claves externo y el administrador de claves externo. También traduce AWS KMS las solicitudes genéricas a un formato que el administrador de claves externo específico del proveedor comprenda. Se necesita un proxy del almacén de claves externo para un almacén de claves externo. Cada almacén de claves externo está asociado a un proxy del almacén de claves externo.



AWS KMS no puede crear, eliminar ni administrar ninguna clave externa. El material de claves criptográficas nunca sale de su administrador de claves externo. Toda la comunicación entre su

administrador de claves externo AWS KMS y su administrador de claves externo está mediada por el proxy del almacén de claves externo. AWS KMS envía las solicitudes al proxy del almacén de claves externo y recibe las respuestas del proxy del almacén de claves externo. El proxy del almacén de claves externo es responsable de transmitir las solicitudes AWS KMS a su administrador de claves externo y de transmitir las respuestas de su administrador de claves externo a AWS KMS.

Usted posee y administra el proxy del almacén de claves externo de su almacén de claves externo, y es responsable de su mantenimiento y funcionamiento. Puede desarrollar su proxy de almacén de claves externo basándose en la [especificación de la API de proxy de almacén de claves externo](#) de código abierto que AWS KMS publica o compra una aplicación de proxy a un proveedor. Su proxy del almacén de claves externo puede estar incluido en su administrador de claves externo. Para facilitar el desarrollo de servidores proxy, AWS KMS también proporciona un ejemplo de proxy de almacén de claves externo ([aws-kms-xks-proxy](#)) y un cliente de prueba ([xks-kms-xksproxy-test-client](#)) que comprueba que el proxy de almacén de claves externo cumple con la especificación.

Para autenticarse, el proxy utiliza certificados AWS KMS TLS del lado del servidor. [Para autenticarse en su proxy, AWS KMS firme todas las solicitudes enviadas a su proxy de almacén de claves externo con una credencial de autenticación de proxy SigV4.](#)

El proxy del almacén de claves externo debe ser compatible con HTTP/1.1 o posterior y TLS 1.2 o posterior con al menos uno de los siguientes conjuntos de cifrado:

- TLS_AES_256_GCM_ (TLS 1.3SHA384)
- TLS___ CHACHA20 (POLY1305TLS 1.3SHA256)

 Note

No es AWS GovCloud (US) Region compatible con CHACHA20 TLS___ POLY1305 SHA256

- TLS_ECDHE_RSA_CON_AES_256_GCM_ SHA384 (TLS 1.2)
- TLS_ECDHE_ECDSA_CON_AES_256_GCM_ (TLS 1.2SHA384)

Para crear y utilizar las claves de KMS en el almacén de claves externo, primero debe [conectar el almacén de claves externo](#) a su proxy del almacén de claves externo. También puede desconectar el almacén de claves externo de su proxy cuando lo solicite. Cuando lo haga, todas las claves de KMS del almacén de claves externo dejarán de estar [disponibles](#); no se podrán utilizar en ninguna operación criptográfica.

Conectividad proxy del almacén de claves externo

La conectividad proxy del almacén de claves externo («conectividad proxy XKS») describe el método que se utiliza para comunicarse con el proxy del almacén de claves externo. AWS KMS

Especifica la opción de conectividad de proxy al crear el almacén de claves externo y pasa a ser una propiedad del almacén de claves externo. Puede cambiar la opción de conectividad del proxy al actualizar la propiedad del almacén de claves personalizado, pero debe asegurarse de que el proxy del almacén de claves externo pueda seguir accediendo a las mismas claves externas.

AWS KMS admite las siguientes opciones de conectividad.

- [Conectividad de punto final público](#): AWS KMS envía solicitudes para el proxy de su almacén de claves externo a través de Internet a un punto final público que usted controle. Esta opción es fácil de crear y mantener, pero es posible que no cumpla los requisitos de seguridad de todas las instalaciones.
- [Conectividad del servicio de punto final de VPC](#): AWS KMS envía solicitudes a un servicio de punto final de Amazon Virtual Private Cloud (Amazon VPC) que usted crea y mantiene. Puede alojar su proxy de almacén de claves externo dentro de su Amazon VPC o alojar su proxy de almacén de claves externo fuera AWS y usar la Amazon VPC solo para comunicarse. También se puede conectar el almacén de claves externo a un servicio de punto de conexión de Amazon VPC propiedad de otra Cuenta de AWS.

Para obtener más información sobre las opciones de conectividad de proxy del almacén de claves externo, consulte [Elección de una opción de conectividad proxy del almacén de claves externo](#).

Credencial de autenticación de proxy del almacén de claves externo

Para autenticarse en su proxy de almacén de claves externo, AWS KMS firme todas las solicitudes enviadas a su proxy de almacén de claves externo con una credencial de autenticación [Signature V4 \(SigV4\)](#). Establece y mantiene la credencial de autenticación en el proxy y, a continuación, se la proporciona AWS KMS al crear el almacén externo.

Note

La credencial SigV4 que se AWS KMS utiliza para firmar las solicitudes al proxy XKS no está relacionada con ninguna de las credenciales SigV4 asociadas a los principales de su

servidor. AWS Identity and Access Management Cuentas de AWS No reutilice ninguna credencial SigV4 de IAM para su proxy del almacén de claves externo.

Cada credencial de autenticación del proxy tiene dos partes. Debe proporcionar ambas partes al crear un almacén de claves externo o actualizar la credencial de autenticación del almacén de claves externo.

- ID de clave de acceso: identifica la clave de acceso secreta. Puede proporcionar este ID en texto sin formato.
- Clave de acceso secreta: la parte secreta de la credencial. AWS KMS cifra la clave de acceso secreta de la credencial antes de almacenarla.

Puede [editar la configuración de la credencial](#) en cualquier momento, por ejemplo, al introducir valores incorrectos, al cambiar la credencial en el proxy o cuando el proxy rota la credencial. Para obtener información técnica sobre la AWS KMS autenticación en el proxy del almacén de claves externo, consulte [Autenticación](#) en la especificación de la API del proxy del almacén de claves AWS KMS externo.

Para que pueda cambiar su credencial sin interrumpir las Servicios de AWS que utilizan claves KMS en su almacén de claves externo, le recomendamos que el proxy del almacén de claves externo admita al menos dos credenciales de autenticación válidas. AWS KMS Esto garantiza que la credencial anterior siga funcionando mientras usted proporciona su nueva credencial a AWS KMS.

Para ayudarlo a realizar un seguimiento de la antigüedad de su credencial de autenticación de proxy, AWS KMS define una CloudWatch métrica de Amazon, [XksProxyCredentialAge](#). Puede usar esta métrica para crear una CloudWatch alarma que le notifique cuando la antigüedad de su credencial alcance el umbral que usted establezca.

Proxy APIs

Para admitir un almacén de claves AWS KMS externo, un proxy de [almacén de claves externo debe implementar el proxy](#) requerido, tal APIs como se describe en la [especificación de la API de proxy del almacén de claves AWS KMS externo](#). Estas solicitudes de API de proxy son las únicas solicitudes que se AWS KMS envían al proxy. Aunque nunca envíe estas solicitudes directamente, conocerlas puede ayudarlo a solucionar cualquier problema que pueda surgir con el almacén de claves externo o el proxy. Por ejemplo, AWS KMS incluye información sobre la latencia y las tasas de éxito de estas

llamadas a la API en sus [CloudWatch métricas de Amazon](#) para almacenes de claves externos. Para obtener más información, consulte [Monitoreo de almacenes de claves externos](#).

En la siguiente tabla se enumeran y describen cada uno de los proxies APIs. También incluye las AWS KMS operaciones que desencadenan una llamada a la API de proxy y cualquier excepción de AWS KMS operación relacionada con la API de proxy.

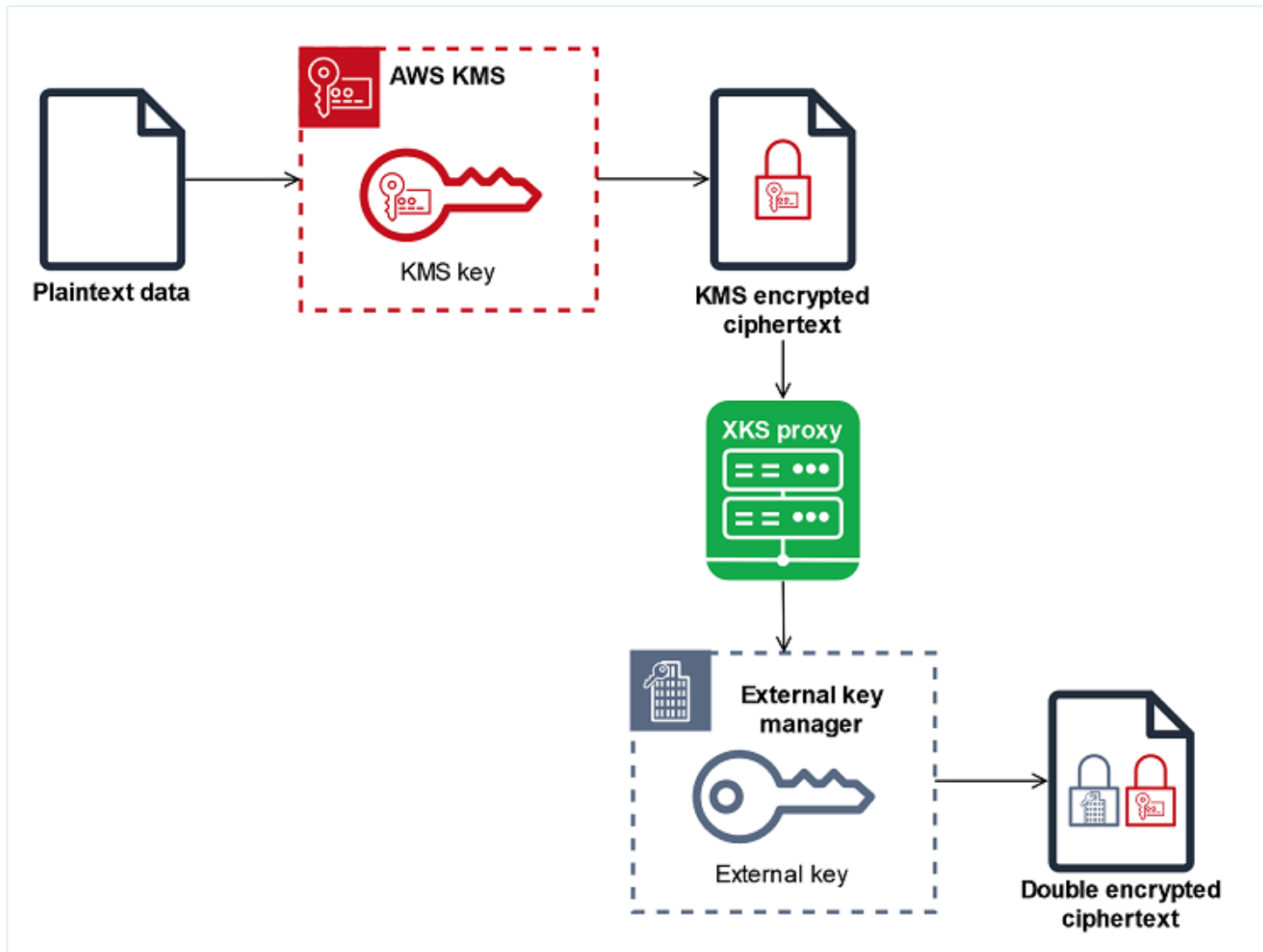
API de proxy	Description (Descripción)	AWS KMS Operaciones relacionadas
Decrypt	AWS KMS envía el texto cifrado que se va a descifrar y el identificador de la clave externa que se va a utilizar. El algoritmo de cifrado requerido es AES_GCM.	Descifrar , ReEncrypt
Encrypt	AWS KMS envía los datos que se van a cifrar y el ID de la clave externa que se va a utilizar. El algoritmo de cifrado requerido es AES_GCM.	Cifrar , GenerateDataKey , GenerateDataKeyWithoutPlaintextReEncrypt
GetHealthStatus	AWS KMS solicita información sobre el estado del proxy y de su administrador de claves externo. El estado de cada administrador de claves externo puede ser uno de los siguientes. • Active: en buen estado; puede servir al tráfico • Degraded: en mal estado, pero puede servir al tráfico • Unavailable : en mal estado; no sirve para el tráfico	CreateCustomKeyStore(para conectividad de punto final público), ConnectCustomKeyStore(para conectividad de servicio de punto final de VPC) Si todas las instancias del administrador de claves externo tienen estado <code>Unavailable</code> , fallan los intentos de crear o conectar el almacén de claves con XksProxyUriUnreachableException .
GetKeyMetadata	AWS KMS solicita información sobre la clave externa asociada a una clave	CreateKey Si la especificación de la clave no es AES_256, el uso de la clave

API de proxy	Description (Descripción)	AWS KMS Operaciones relacionadas
	de KMS en su almacén de claves externo. La respuesta incluye la especificación de la clave (AES_256), el uso de la clave ([ENCRYPT, DECRYPT]) y si la clave externa está ENABLED o DISABLED.	no es [ENCRYPT, DECRYPT] ni el estado es DISABLED, la operación de CreateKey fallará con XksKeyInvalidConfigurationException .

Cifrado doble

Los datos cifrados por una clave de KMS en un almacén de claves externo se cifran dos veces. En primer lugar, AWS KMS cifra los datos con material AWS KMS clave específico de la clave KMS. A continuación, su [administrador de claves externo](#) cifra el texto cifrado por AWS KMS con su [clave externa](#). Este proceso se conoce como doble cifrado.

El doble cifrado garantiza que los datos cifrados por una clave de KMS en un almacén de claves externo sean tan seguros como el texto cifrado con una clave de KMS estándar. También protege el texto sin formato en tránsito desde AWS KMS el proxy del almacén de claves externo. Con el doble cifrado, conserva el control total de sus textos cifrados. Si revoca el acceso de AWS a su clave externa de forma permanente a través de su proxy externo, todo el texto cifrado que quede en AWS se destruirá mediante una operación criptográfica.



Para habilitar el doble cifrado, cada clave de KMS de un almacén de claves externo tiene dos claves de respaldo criptográficas:

- Un material AWS KMS clave exclusivo de la clave KMS. Este material clave se genera y solo se utiliza en los módulos de [seguridad de hardware con certificación AWS KMS FIPS 140-3 de nivel 3](#) (HSMs).
- Una [clave externa](#) de su administrador de claves externo.

El doble cifrado tiene los siguientes efectos:

- AWS KMS no puede descifrar ningún texto cifrado mediante una clave KMS en un almacén de claves externo sin acceder a las claves externas a través del proxy del almacén de claves externo.

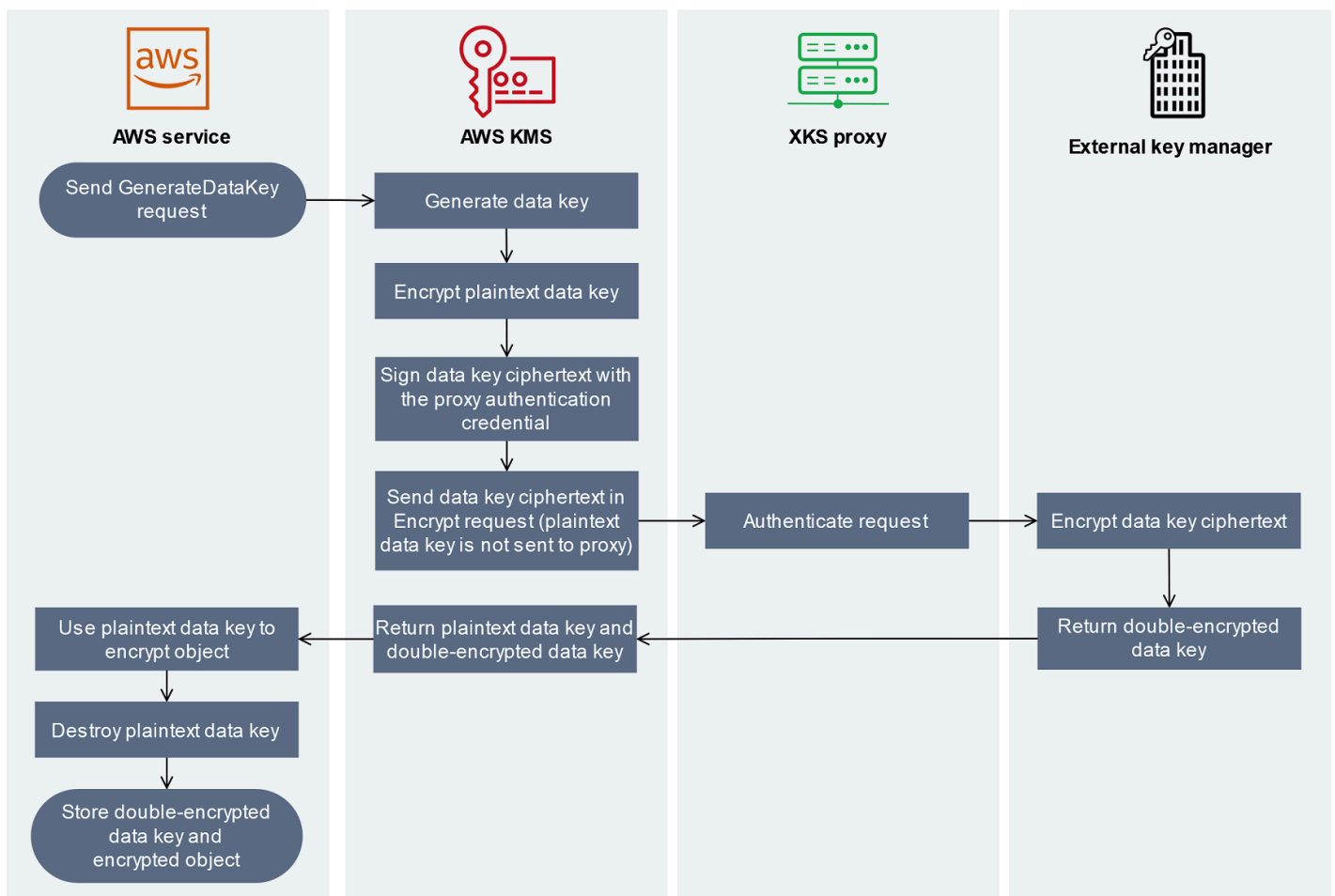
- No puede descifrar ningún texto cifrado mediante una clave KMS en un almacén de claves externo externo AWS, incluso si dispone de su material de clave externa.
- No puede volver a crear una clave de KMS que se haya eliminado de un almacén de claves externo, aunque disponga de su material de claves externas. Cada clave de KMS tiene metadatos únicos que incluye en el texto cifrado simétrico. Una nueva clave de KMS no podría descifrar el texto cifrado con la clave original, incluso si utilizara el mismo material de clave externa.

Para ver un ejemplo de doble cifrado en la práctica, consulte [Cómo funcionan los almacenes de claves externos](#).

Cómo funcionan los almacenes de claves externos

El [almacén de claves externo](#), el [proxy del almacén de claves externo](#) y el [administrador de claves externo](#) trabajan en conjunto para proteger los recursos de AWS . El siguiente procedimiento muestra el flujo de trabajo de cifrado típico de Servicio de AWS que cifra cada objeto con una clave de datos única protegida por una clave de KMS. En este caso, ha elegido una clave de KMS en un almacén de claves externo para proteger el objeto. El ejemplo muestra cómo se AWS KMS utiliza el [doble cifrado](#) para proteger la clave de datos en tránsito y garantizar que el texto cifrado generado por una clave KMS en un almacén de claves externo sea siempre al menos tan seguro como el texto cifrado mediante una clave KMS simétrica estándar con el material clave incluido. AWS KMS

Los métodos de cifrado utilizados por cada entidad que se integra varían. Servicio de AWS AWS KMS Para obtener más información, consulte el tema “Protección de datos” del capítulo Seguridad de la documentación de los Servicio de AWS .



1. Agrega un objeto nuevo a su Servicio de AWS recurso. Para cifrar el objeto, Servicio de AWS envía una [GenerateDataKey](#) solicitud para AWS KMS utilizar una clave KMS en su almacén de claves externo.
2. AWS KMS genera una clave de [datos simétrica de 256 bits y se prepara para enviar una copia de la clave](#) de datos en texto plano al administrador de claves externo a través del proxy del almacén de claves externo. AWS KMS inicia el proceso de [doble cifrado](#) cifrando la clave de datos en texto plano con el [material de clave asociado a la AWS KMS clave](#) KMS en el almacén de claves externo.
3. AWS KMS envía una solicitud de [cifrado al](#) proxy del almacén de claves externo asociado al almacén de claves externo. La solicitud incluye el texto cifrado de la clave de datos que se va a cifrar y el ID de la [clave externa](#) asociada a la clave KMS. AWS KMS firma la solicitud con la [credencial de autenticación proxy del proxy](#) del almacén de claves externo.

La copia en texto sin formato de la clave de datos no se envía al proxy del almacén de claves externo.

4. El proxy del almacén de claves externo autentica la solicitud y, a continuación, la pasa a su administrador de claves externo.

Algunos servidores proxy de almacenes de claves externos también implementan una [política de autorización](#) opcional que permite que solo las entidades principales seleccionadas realicen operaciones en condiciones específicas.

5. El administrador de claves externo cifra el texto cifrado de la clave de datos mediante la clave externa especificada. El administrador de claves externo devuelve la clave de datos con doble cifrado al proxy del almacén de claves externo, que la devuelve a AWS KMS.
6. AWS KMS devuelve la clave de datos en texto plano y la copia doblemente cifrada de esa clave de datos al Servicio de AWS
7. Servicio de AWS Utiliza la clave de datos de texto sin formato para cifrar el objeto de recurso, destruye la clave de datos de texto sin formato y almacena la clave de datos cifrada con el objeto cifrado.

Algunos Servicios de AWS pueden almacenar en caché la clave de datos de texto sin formato para utilizarla en varios objetos o reutilizarla mientras el recurso está en uso. Para obtener más información, consulte [Cómo afectan las claves de KMS obsoletas a las claves de datos](#).

Para descifrar el objeto cifrado, Servicio de AWS deben devolver la clave de datos cifrados a una solicitud AWS KMS de [descifrado](#). Para descifrar la clave de datos cifrada, AWS KMS debe devolverla al proxy del almacén de claves externo con el ID de la clave externa. Si la solicitud de descifrado al proxy del almacén de claves externo falla por algún motivo, AWS KMS no puede descifrar la clave de datos cifrados y Servicio de AWS no puede descifrar el objeto cifrado.

Control de acceso al almacén de claves externo

Todas las funciones de control de AWS KMS acceso ([políticas clave](#), [políticas de IAM](#) y [concesiones](#)) que se utilizan con las claves KMS estándar funcionan de la misma manera para las claves KMS de un almacén de claves externo. Puede usar las políticas de IAM para controlar el acceso a las operaciones de la API que crean y administran almacenes de claves externos. Las políticas de IAM y las políticas clave se utilizan para controlar el acceso al AWS KMS keys almacén de claves externo. También puedes usar políticas de [control de servicios para tu AWS organización y políticas de punto final de VPC para controlar el acceso a las](#) claves de KMS en tu almacén de claves externo.

Le recomendamos que únicamente otorgue a los usuarios y roles los permisos necesarios para las tareas que es probable que se vayan a realizar.

Temas

- [Autorizar a administradores de almacenes de claves externos](#)
- [Autorización de usuarios de claves de KMS en almacenes de claves externos](#)
- [AWS KMS Autorizar la comunicación con el proxy de tu almacén de claves externo](#)
- [Autorización del proxy del almacén de claves externo \(opcional\)](#)
- [Autenticación mTLS \(obsoleta\)](#)

Autorizar a administradores de almacenes de claves externos

Las entidades principales que crean y administran un almacén de claves externo necesitan permisos para las operaciones del almacén de claves personalizado. En la siguiente lista, se describen los permisos mínimos necesarios para los administradores del almacén de claves externo. Como un almacén de claves personalizado no es un AWS recurso, no puedes conceder permisos a un almacén de claves externo para los principales de otros. Cuentas de AWS

- kms:CreateCustomKeyStore
- kms:DescribeCustomKeyStores
- kms:ConnectCustomKeyStore
- kms:DisconnectCustomKeyStore
- kms:UpdateCustomKeyStore
- kms>DeleteCustomKeyStore

Para crear un almacén de claves externo con la [conectividad del servicio de puntos de enlace de Amazon VPC y el servicio](#) de puntos de enlace de VPC sea propiedad de otra persona Cuenta de AWS, también necesitará el siguiente permiso:

- ec2:DescribeVPCEndpointServices

Las entidades principales que crean un almacén de claves externo necesitan permiso para crear y configurar los componentes del almacén de claves externo. Las entidades principales pueden crear almacenes de claves externos solo en sus propias cuentas. Para crear un almacén de claves externo con [conectividad a los servicios de punto de conexión de VPC](#), las entidades principales deben tener permiso para crear los siguientes componentes:

- Una VPC de Amazon

- Subredes públicas y privadas
- Un equilibrador de carga de red y grupo de destino
- Un servicio de punto de conexión de Amazon VPC

Para obtener más información, consulte [Identity and Access Management para Amazon VPC](#), [Identity and Access Management para puntos de conexión de VPC y servicios de puntos de conexión de VPC](#) y [Elastic Load Balancing API permissions](#) (Permisos de la API de Elastic Load Balancing).

Autorización de usuarios de claves de KMS en almacenes de claves externos

Los responsables que crean y administran AWS KMS keys tu almacén de claves externo necesitan [los mismos permisos](#) que quienes crean y administran cualquier clave de KMS. AWS KMS La [política de claves predeterminada](#) para claves de KMS en un almacén de claves externo es idéntica a la política de claves predeterminada para claves de KMS en AWS KMS. El [control de acceso basado en atributos](#) (ABAC), que utiliza etiquetas y alias para controlar el acceso a las claves de KMS, también es efectivo para claves de KMS en almacenes de claves externos.

Las entidades principales que usan las claves KMS en el almacén de claves personalizado para [operaciones criptográficas](#) requieren permiso para realizar la operación criptográfica con la clave KMS, por ejemplo, [kms:Decrypt](#). Puede proporcionar estos permisos en una política de IAM o en una política de claves. Sin embargo, no necesitan más permisos para utilizar una clave KMS en un almacén de claves personalizado.

Para establecer un permiso que se aplique solo a las claves de KMS en un almacén de claves externo, utilice la condición de política [kms:KeyOrigin](#) con un valor de EXTERNAL_KEY_STORE. Puedes usar esta condición para limitar el CreateKey permiso de [KMS](#): o cualquier permiso específico de un recurso clave de KMS. Por ejemplo, la siguiente política de IAM permite a la identidad a la que está asociada llamar a las operaciones especificadas en cualquier clave de KMS de la cuenta, siempre que las claves de KMS se encuentren en un almacén de claves externo. Ten en cuenta que puedes limitar el permiso a las claves de KMS de un almacén de claves externo y a las claves de KMS de un almacén de claves externo concreto de la cuenta Cuenta de AWS, pero no a ningún almacén de claves externo en concreto.

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

AWS KMS Autorizar la comunicación con el proxy de tu almacén de claves externo

AWS KMS se comunica con su administrador de claves externo únicamente a través del [proxy de almacén de claves externo](#) que usted proporcione. AWS KMS se autentica en su proxy firmando sus solicitudes mediante el [proceso de firma de la versión 4 \(SiGv4\)](#) con la [credencial de autenticación del proxy del almacén de claves externo que especifique](#). Si utiliza la [conectividad de un punto final público](#) para su proxy de almacén de claves externo, AWS KMS no necesita ningún permiso adicional.

Sin embargo, si utiliza la [conectividad del servicio de puntos de enlace de VPC](#), debe conceder AWS KMS permiso para crear un punto de enlace de interfaz para su servicio de puntos de enlace de Amazon VPC. Este permiso es necesario independientemente de si el proxy del almacén de claves externo está en su VPC o si el proxy del almacén de claves externo está ubicado en otro lugar, pero utiliza el servicio de punto final de la VPC para comunicarse con él. AWS KMS

AWS KMS Para permitir la creación de un punto final de interfaz, utilice la [consola de Amazon VPC](#) o la [ModifyVpcEndpointServicePermissions](#) operación. Conceda permisos para la siguiente entidad principal: `cks.kms.<region>.amazonaws.com`.

Si su servicio de puntos de conexión de Amazon VPC es propiedad de una persona Cuenta de AWS distinta a la Cuenta de AWS propietaria del almacén de claves externo (XKS), también tendrá que permitir que XKS acceda al servicio de puntos de conexión de VPC. Para ello, incluya [el Cuenta de AWS ID de XKS en una lista de permitidos como principal](#) para el servicio de puntos de conexión de Amazon VPC.

Same Cuenta de AWS

Si su servicio de punto final de VPC es propiedad de la Cuenta de AWS misma propiedad que su almacén de claves externo, debe agregarlo AWS KMS a la lista Permitir entidades principales de su servicio de punto final de VPC.

En el siguiente ejemplo, se utiliza el [modify-vpc-endpoint-service-permissions](#) AWS CLI comando AWS KMS para permitir la conexión al servicio de punto final de la VPC especificado en la región EE.UU. Oeste (Oregón) (us-west-2). Antes de usar este comando, sustituya el ID de servicio de Amazon VPC por valores válidos para su configuración. Región de AWS

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com"]'
```

Para eliminar este permiso, utilice la [consola de Amazon VPC](#) o el parámetro [ModifyVpcEndpointServicePermissions](#)with. RemoveAllowedPrincipals

Cross Cuenta de AWS

Cuando su servicio de punto final de VPC sea propiedad del otro Cuenta de AWS, debe agregar ambos AWS KMS y su almacén de claves externo a la lista Permitir principales de su servicio de punto final de VPC.

En el siguiente ejemplo, se utiliza el [modify-vpc-endpoint-service-permissions](#) AWS CLI comando para permitir que tanto AWS KMS el almacén de claves externo (XKS) como el almacén de claves externo (XKS) se conecten al servicio de punto final de la VPC especificado en la región EE.UU. Oeste (Oregón) (us-west-2). Antes de ejecutar este comando, reemplace el ID de servicio de Amazon VPC, la Región de AWS y el ARN de la entidad principal de IAM con valores válidos para su configuración. El principal de IAM debe sustituirse por un principal del propietario del XKS. Cuenta de AWS

En este ejemplo, `arn:aws:iam::123456789012:role/cks_role` es la entidad principal de IAM en la cuenta de propietario de XKS, que se utilizará para crear, actualizar o conectar el XKS a su servicio de punto de conexión de VPC. Si desea permitir que todas las entidades principales de la cuenta de propietario de XKS accedan a su servicio de punto de conexión de VPC, puede 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 eliminar este permiso, utilice la [consola de Amazon VPC](#) o el parámetro [ModifyVpcEndpointServicePermissions](#) con `RemoveAllowedPrincipals`.

Autorización del proxy del almacén de claves externo (opcional)

Algunos proxy del almacén de claves externo implementan requisitos de autorización para el uso de sus claves externas. Se permite, pero no es obligatorio, utilizar un proxy del almacén de claves externo para diseñar e implementar un esquema de autorización que permita a determinados usuarios solicitar ciertas operaciones únicamente en determinadas condiciones. Por ejemplo, un proxy puede configurarse para permitir al usuario A cifrar con una clave externa determinada, pero no descifrar con ella.

La autorización de proxy es independiente de la [autenticación de proxy basada en SIGv4](#) que se requiere para todos los proxies de almacenamiento de claves externo. También es independiente de las políticas de clave, las políticas de IAM y las concesiones que autorizan el acceso a las operaciones que afectan al almacén de claves externo o a sus claves de KMS.

Para permitir la autorización por parte del proxy del almacén de claves externo, AWS KMS incluye metadatos en cada [solicitud de API de proxy](#), como la persona que llama, la clave de KMS, la AWS KMS operación y (si la hubiera). Los metadatos de solicitud para la versión 1 (v1) de la API del proxy de clave externa son los siguientes.

```
"requestMetadata": {  
  "awsPrincipalArn": string,  
  "awsSourceVpc": string, // optional  
  "awsSourceVpce": string, // optional  
  "kmsKeyArn": string,  
  "kmsOperation": string,  
  "kmsRequestId": string,  
  "kmsViaService": string // optional  
}
```

Por ejemplo, puedes configurar tu proxy para que permita las solicitudes de un director en particular (`awsPrincipalArn`), pero solo cuando la solicitud la realice en nombre del principal Servicio de AWS (`kmsViaService`).

Si se produce un error en la autorización del proxy, se produce un error en la AWS KMS operación relacionada y aparece un mensaje en el que se explica el error. Para obtener más información, consulte [Problemas de autorización de proxy](#).

Autenticación mTLS (obsoleta)

Las versiones anteriores de esta guía mencionaban la seguridad mutua de la capa de transporte (mTLS) como un mecanismo de autenticación secundario opcional para autenticar las solicitudes. AWS KMS Con los mTLS, ambas partes (AWS KMS como cliente y el proxy XKS como servidor) que se comunican a través de un canal TLS utilizan certificados para autenticarse mutuamente.

Sin embargo, los cambios en la [política del programa root de Chrome \(sección 4.2.2\)](#) prohíben que los usuarios root de Chrome que sean CAs de confianza pública emitan certificados con la extensión ClientAuth Extended Key Usage (EKU) después del 15 de junio de 2026. Como resultado, ya no AWS KMS se puede obtener un certificado de cliente adecuado para mTLS de [Amazon Trust Services](#). Cualquier proxy XKS que se utilice para crear un nuevo almacén de claves externo AWS KMS después del 16 de marzo de 2026 no debe requerir mTLS. Después del 15 de junio de 2026, ningún proxy XKS configurado para requerir mTLS no podrá comunicarse con él. AWS KMS Los clientes deben confiar en la autenticación SigV4 para verificar el origen de las solicitudes. AWS KMS Para obtener más información, consulte [Credencial de autenticación de proxy de almacén de claves externo](#).

Elección de una opción de conectividad proxy del almacén de claves externo

Antes de crear el almacén de claves externo, elija la opción de conectividad que determine cómo AWS KMS se comunica con los componentes del almacén de claves externo. La opción de conectividad que elija determina el resto del proceso de planificación.

Si va a crear un almacén de claves externo, debe determinar cómo AWS KMS se comunica con el [proxy del almacén de claves externo](#). Esta elección determinará qué componentes necesita y cómo los configura. AWS KMS admite las siguientes opciones de conectividad.

- [Conectividad de punto de conexión público](#)
- [Conectividad del servicio del punto de conexión de VPC](#)

Elija la opción que se ajuste a sus objetivos de rendimiento y seguridad.

Antes de empezar, [confirme que necesita un almacén de claves externo](#). La mayoría de los clientes pueden usar claves KMS respaldadas por material AWS KMS clave.

Consideraciones

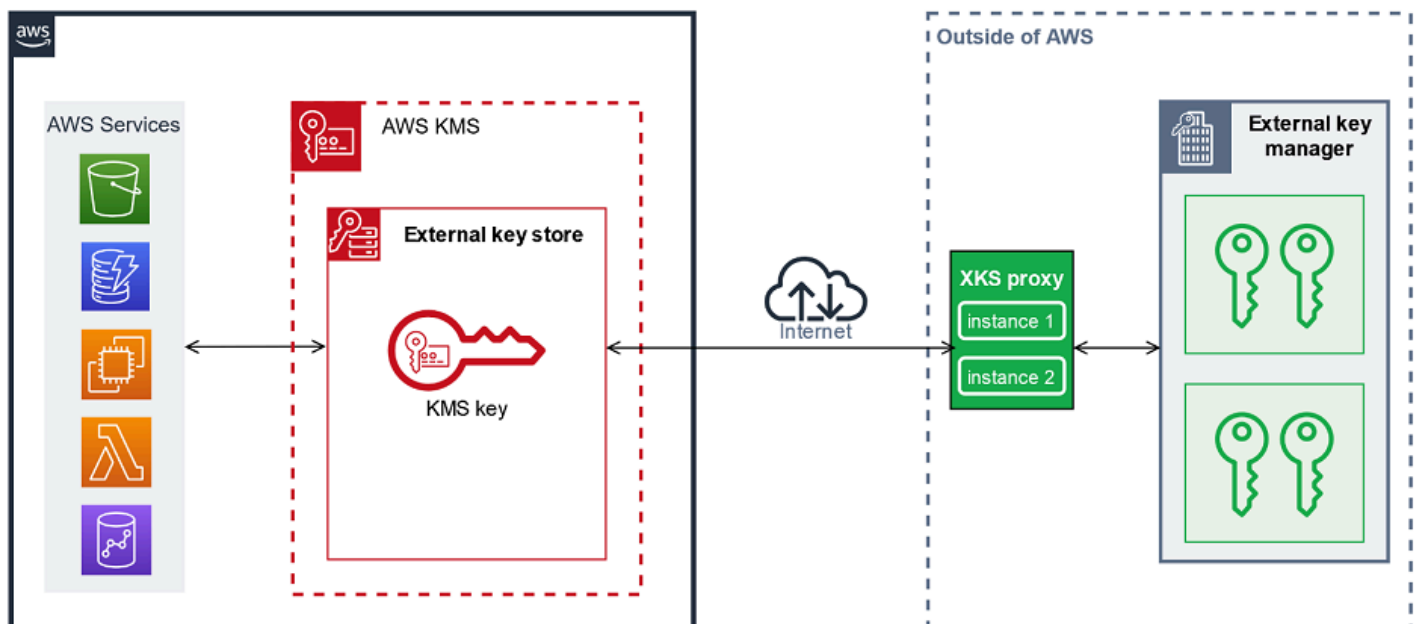
- Si el proxy del almacén de claves externo está integrado en el administrador de claves externo, es posible que la conectividad esté predeterminada. Para obtener orientación, consulte la documentación de su administrador de claves externo o proxy del almacén de claves externo.
- Puede [cambiar la opción de conectividad proxy del almacén de claves externo](#) incluso en un almacén de claves externo en funcionamiento. Sin embargo, el proceso debe planificarse y ejecutarse en detalle para minimizar las interrupciones, evitar errores y garantizar el acceso continuo a las claves criptográficas que cifran los datos.

Conectividad de punto de conexión público

AWS KMS se conecta al proxy del almacén de claves externo (proxy XKS) a través de Internet mediante un punto final público.

Esta opción de conectividad es más fácil de configurar y mantener, y se alinea bien con algunos modelos de administración de claves. Sin embargo, es posible que no cumpla con los requisitos de seguridad de algunas organizaciones.

XKS proxy connected by a public endpoint



Requisitos

Si elige la conectividad de punto de conexión público, se requiere lo siguiente.

- El proxy de su almacén de claves externo debe estar accesible en un punto de conexión que se pueda enrutar públicamente.
- Puede utilizar el mismo punto de conexión público para varios almacenes de claves externos, siempre que utilicen valores de [ruta URI de proxy](#) diferentes.
- No puede usar el mismo punto final para un almacén de claves externo con conectividad de punto final público y cualquier almacén de claves externo con conectividad de servicios de punto final de VPC en el mismo lugar Región de AWS, incluso si los almacenes de claves están en diferentes. Cuentas de AWS

- Debe obtener un certificado TLS emitido por una autoridad de certificación pública compatible con almacenes de claves externos. Para obtener una lista, consulte [Autoridades de certificación de confianza](#).

El nombre común del sujeto (CN) en el certificado TLS debe coincidir con el nombre de dominio en el [punto de conexión URI del proxy](#) para el proxy del almacén de claves externo. Por ejemplo, si el punto de conexión público es `https://myproxy.xks.example.com`, el TLS, el CN del certificado TLS debe ser `myproxy.xks.example.com` o `*.xks.example.com`.

- Asegúrese de que todos los firewalls que se encuentren entre el proxy del almacén de claves externo AWS KMS y el servidor proxy permitan el tráfico desde y hacia el puerto 443 del proxy. AWS KMS se comunica a través del puerto 443. IPv4 Este valor no se puede configurar.

Para conocer todos los requisitos de un almacén de claves externo, consulte [Ensamblaje de los requisitos previos](#).

Conectividad del servicio del punto de conexión de VPC

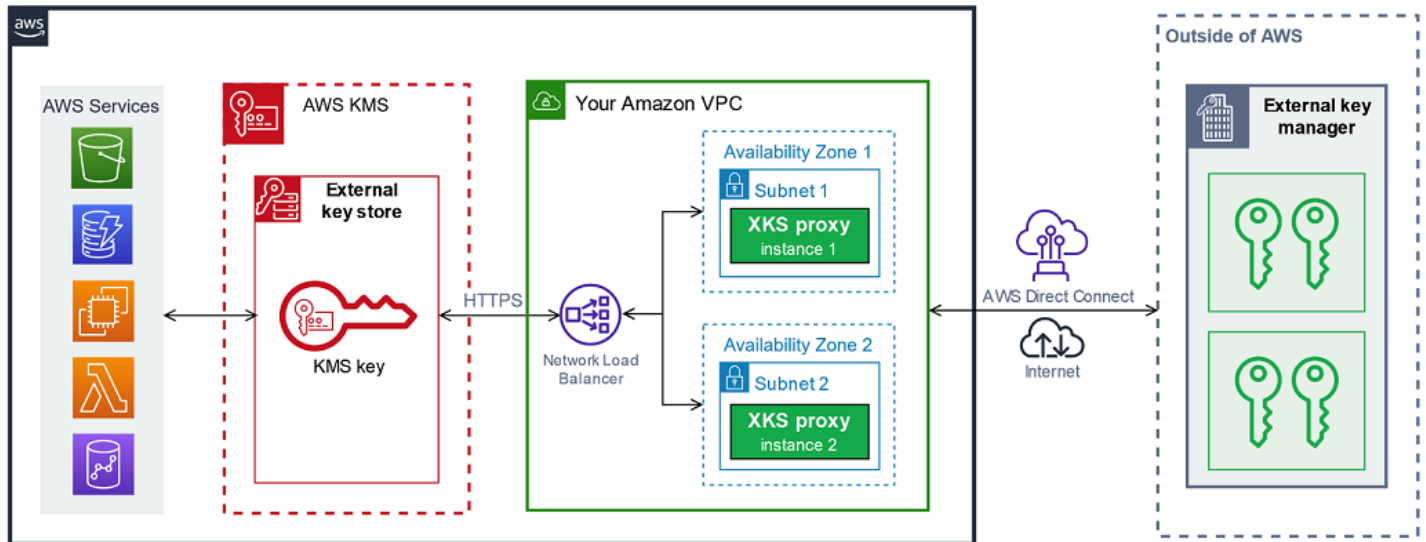
AWS KMS se conecta al proxy del almacén de claves externo (proxy XKS) mediante la creación de un punto de enlace de interfaz a un servicio de punto final de Amazon VPC que usted cree y configure. Usted es responsable de [crear el servicio de punto de conexión de VPC](#) y de conectar la VPC al administrador de claves externo.

Su servicio de punto final puede usar cualquiera de las [opciones de network-to-Amazon VPC compatibles para las](#) comunicaciones, incluidas. [AWS Direct Connect](#)

Esta opción de conectividad es más complicada de configurar y mantener. Pero lo utiliza AWS PrivateLink, lo que le permite AWS KMS conectarse de forma privada a su Amazon VPC y a su proxy de almacén de claves externo sin utilizar la Internet pública.

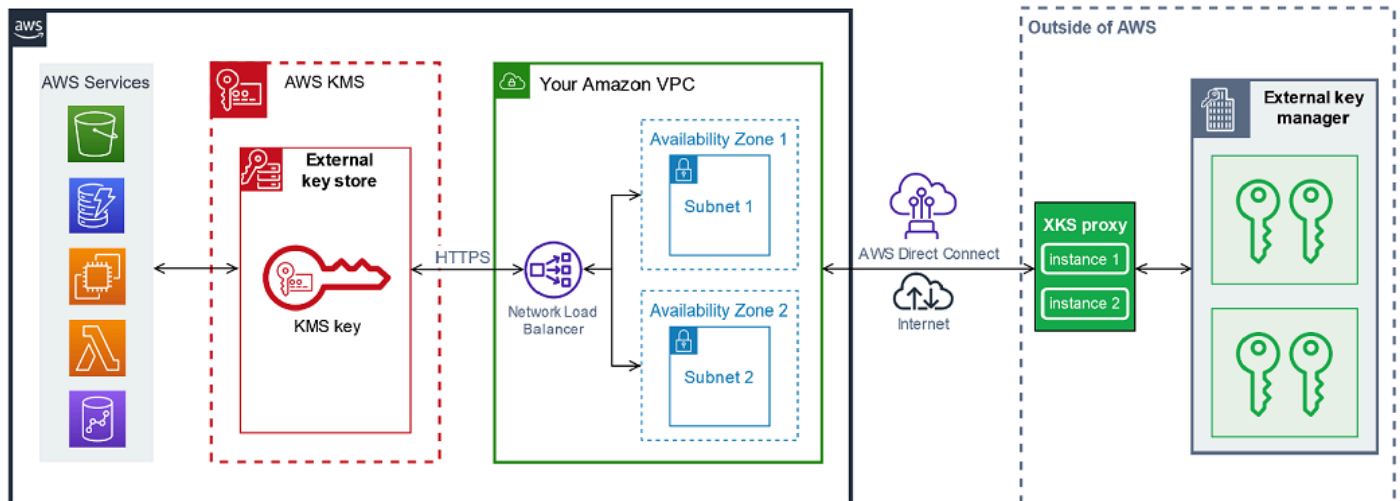
Puede localizar su proxy del almacén de claves externo en su VPC de Amazon.

XKS proxy hosted in Amazon VPC



Como alternativa, puede ubicar el proxy de su almacén de claves externo fuera del servicio de puntos de conexión de Amazon VPC Nube de AWS y utilizarlo únicamente para comunicarse de forma segura con él. AWS KMS

XKS proxy connected via Amazon VPC endpoint service



También puede conectar un almacén de claves externo a un servicio de punto final de Amazon VPC propiedad de otra persona. Cuenta de AWS Ambas Cuentas de AWS necesitan los [permisos necesarios](#) para permitir las comunicaciones entre AWS KMS y el servicio de punto final de la VPC.

Más información:

- Revise el proceso de creación de un almacén de claves externo, incluido [el ensamblaje de los requisitos previos](#). Lo ayudará a asegurarse de que dispone de todos los componentes que necesita para crear su almacén de claves externo.
- Aprenda a [controlar el acceso a su almacén de claves externo](#), incluidos los permisos que requieren los administradores y usuarios del almacén de claves externo.
- Obtén información sobre las [CloudWatch métricas y dimensiones de Amazon](#) que AWS KMS registran los almacenes clave externos. Le recomendamos encarecidamente que cree alarmas para monitorear su almacén de claves externo y así poder detectar los primeros signos de problemas operativos y de rendimiento.

Configuración de la conectividad del servicio de punto de conexión de VPC

Utilice las instrucciones de esta sección para crear y configurar AWS los recursos y los componentes relacionados necesarios para un almacén de claves externo que utilice la conectividad del [servicio de punto final de VPC](#). Los recursos que se enumeran para esta opción de conectividad complementan los [recursos necesarios para todos los almacenes de claves externos](#). Después de crear y configurar los recursos necesarios, puede [crear su almacén de claves externo](#).

Puede ubicar el proxy de su almacén de claves externo en su Amazon VPC o ubicar el proxy fuera de su servicio de punto final de VPC AWS y usarlo para comunicarse.

Antes de empezar, [confirme que necesita un almacén de claves externo](#). La mayoría de los clientes pueden usar claves de KMS respaldadas por material AWS KMS clave.

Note

Es posible que algunos de los elementos necesarios para la conectividad del servicio de punto de conexión de VPC estén incluidos en el administrador de claves externo. Además, es posible que el software tenga requisitos de configuración adicionales. Antes de crear y configurar los AWS recursos de esta sección, consulte la documentación del proxy y del administrador de claves.

Temas

- [Requisitos para la conectividad del servicio del punto de conexión de VPC](#)
- [Paso 1: Crear una VPC de Amazon y subredes](#)
- [Paso 2: Crear un grupo de destino](#)
- [Paso 3: Crear un equilibrador de carga de red](#)
- [Paso 4: Crear un servicio de punto de conexión de VPC](#)
- [Paso 5: Verificar el dominio de su nombre DNS privado](#)
- [Paso 6: Autorizar AWS KMS la conexión al servicio de punto final de la VPC](#)

Requisitos para la conectividad del servicio del punto de conexión de VPC

Si elige la conectividad del servicio de punto de conexión de VPC para su almacén de claves externo, necesitará los siguientes recursos.

- Una VPC de Amazon que esté conectada a su administrador de claves externo. Debe tener al menos dos [subredes](#) privadas en dos zonas de disponibilidad diferentes.

Puede utilizar una VPC de Amazon existente para su almacén de claves externo, siempre que [cumpla los requisitos](#) para su uso con un almacén de claves externo. Varios almacenes de claves externos pueden compartir una VPC de Amazon, pero cada almacén de claves externo debe tener su propio servicio de punto de conexión de VPC y un nombre DNS privado.

- Un [servicio de punto de conexión de VPC de Amazon proporcionado por un AWS PrivateLink](#) con un [Equilibrador de carga de red](#) y un [grupo de destino](#).

El servicio de punto de conexión no puede requerir aceptación. Además, debe agregar AWS KMS como entidad principal permitida. Esto permite AWS KMS crear puntos finales de interfaz para que pueda comunicarse con el proxy externo del almacén de claves.

- Un nombre de DNS privado para el servicio de punto de conexión de VPC que es único en su Región de AWS.

El nombre DNS privado debe ser un subdominio de un dominio público de nivel superior. Por ejemplo, si el nombre DNS privado es `myproxy-private.xks.example.com`, debe ser un subdominio de un dominio público, como `xks.example.com` o `example.com`.

Debe [verificar la propiedad](#) del dominio de DNS para el nombre DNS privado.

- Un certificado TLS emitido por una [autoridad de certificación pública admitida](#) para su proxy del almacén de claves externo.

El nombre común del sujeto (CN) en el certificado TLS debe coincidir con el nombre DNS privado. Por ejemplo, si el nombre DNS privado es `myproxy-private.xks.example.com`, el CN del certificado TLS debe ser `myproxy-private.xks.example.com` o `*.xks.example.com`.

- Para minimizar la latencia de la red, cree sus AWS componentes en el [soporte Región de AWS](#) que esté más cerca de su [administrador de claves externo](#). Si es posible, elija una región con un tiempo de ida y vuelta (RTT) de la red de 35 milisegundos o menos.

Para conocer todos los requisitos de un almacén de claves externo, consulte [Ensamblaje de los requisitos previos](#).

Paso 1: Crear una VPC de Amazon y subredes

La conectividad del servicio de punto de conexión de VPC requiere una VPC de Amazon que esté conectada a su administrador de claves externo con al menos dos subredes privadas. Puede crear una VPC de Amazon o utilizar una VPC de Amazon existente que cumpla con los requisitos para los almacenes de claves externos. Para obtener ayuda con la creación de una nueva VPC de Amazon, consulte [Crear una VPC](#) en la Guía del usuario de Amazon Virtual Private Cloud.

Requisitos para su VPC de Amazon

Para trabajar con almacenes de claves externos el servicio de punto de conexión de Amazon VPC debe tener las siguientes propiedades.

- Debe estar en la misma [región admitida](#) que su almacén de claves externo.
- Requiere al menos dos subredes privadas, cada una en una zona de disponibilidad diferente.
- El intervalo de direcciones IP privadas de su VPC de Amazon no debe superponerse con el intervalo de direcciones IP privadas del centro de datos que aloja su [administrador de claves externo](#).
- Se deben utilizar todos los componentes IPv4.

Tiene muchas opciones para conectar la VPC de Amazon a su proxy del almacén de claves externo. Elija la opción que se ajuste a sus necesidades de rendimiento y seguridad. Para ver una lista, consulte [Conectar la VPC a otras redes](#) y Opciones de conectividad de la [Network-to-Amazon VPC](#). Para obtener más detalles, consulte [Direct Connect](#) y la [Guía del usuario de AWS Site-to-Site VPN](#).

Creación de una VPC de Amazon para su almacén de claves externo

Use las siguientes instrucciones para crear la VPC de Amazon para su almacén de claves externo. Solo se requiere una VPC de Amazon si elige la opción de [conectividad del servicio de punto de conexión de VPC](#). Puede utilizar una VPC de Amazon existente que cumpla los requisitos para un almacén de claves externo.

Siga las instrucciones que se indican en el tema [Crear una VPC, subredes y otros recursos de la VPC](#) con los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
IPv4 Bloque CIDR	Introduzca las direcciones IP de la VPC. El intervalo de direcciones IP privadas de su VPC de Amazon no debe superponerse con el intervalo de direcciones IP privadas del centro de datos que aloja su administrador de claves externo .
Número de zonas de disponibilidad () AZs	2 o más
Número de subredes públicas	No se requiere ninguna (0)
Número de subredes privadas	Una para cada AZ
Puerta de enlace NAT	No se requiere ninguna.
Puntos de conexión de VPC	No se requiere ninguna.
Enable DNS hostnames	Sí

Campo	Valor
Habilitar la resolución de DNS	Sí

Asegúrese de probar la comunicación de la VPC. Por ejemplo, si su proxy de almacén de claves externo no se encuentra en su VPC de Amazon, cree una instancia de Amazon EC2 en su VPC de Amazon y compruebe que la VPC de Amazon pueda comunicarse con su proxy del almacén de claves externo.

Conexión de la VPC al administrador de claves externo

Conecte la VPC al centro de datos que aloja su administrador de claves externo mediante cualquiera de las [opciones de conectividad de red](#) que admite la VPC de Amazon. Asegúrese de que la instancia de Amazon EC2 en la VPC (o el proxy del almacén de claves externo, si está en la VPC) pueda comunicarse con el centro de datos y el administrador de claves externo.

Paso 2: Crear un grupo de destino

Antes de crear el servicio de punto de conexión de VPC requerido, cree los componentes necesarios, un equilibrador de carga de red (NLB) y un grupo de destino. El equilibrador de carga de red (NLB) distribuye las solicitudes entre varios objetivos en buen estado, cualquiera de los cuales puede atender la solicitud. En este paso, crea un grupo de destino con al menos dos servidores para su proxy del almacén de claves externo y registra sus direcciones IP en el grupo de destino.

Siga las instrucciones que se indican en el tema [Configuración de un grupo de destino](#) con los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Tipo de destino	Direcciones IP
Protocolo	TCP
Puerto	443

Campo	Valor
Tipo de dirección IP	IPv4
VPC	Elija la VPC en la que va a crear el servicio de punto de conexión de VPC para su almacén de claves externo.
Protocolo y ruta de comprobación de estado	El protocolo y la ruta de comprobación de estado variarán según la configuración del proxy del almacén de claves externo. Consulte la documentación de su administrador de claves externo o proxy del almacén de claves externo. Para obtener información general sobre la configuración de comprobaciones de estado para sus grupos de destino, consulte Comprobaciones de estado para sus grupos de destino en la Guía del usuario del equilibrador de carga elástico para el equilibrador de carga de red.
Network	Otra dirección IP privada
IPv4 dirección	Las direcciones privadas del proxy de su almacén de claves externo
Puertos	443

Paso 3: Crear un equilibrador de carga de red

El equilibrador de carga de red distribuye el tráfico de la red, incluidas las solicitudes de AWS KMS a su proxy del almacén de claves externo, a los destinos configurados.

Siga las instrucciones del tema [Configuración de un equilibrador de carga y un oyente](#) para configurar y agregar un oyente y crear un equilibrador de carga con los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Esquema	Internal
Tipo de dirección IP	IPv4

Campo	Valor
Asignación de redes	Elija la VPC en la que va a crear el servicio de punto de conexión de VPC para su almacén de claves externo.
Correspondencia	Elija las dos zonas de disponibilidad (al menos dos) que configuró para las subredes de VPC. Compruebe los nombres de las subredes y la dirección IP privada.
Protocolo	TCP
Puerto	443
Acción predeterminada: Reenviar a	Elija el grupo de destino para su equilibrador de carga de red.

Paso 4: Crear un servicio de punto de conexión de VPC

Por lo general, se crea un punto de conexión para un servicio. Sin embargo, cuando crea un servicio de punto final de VPC, usted es el proveedor y AWS KMS crea un punto final para su servicio. Para un almacén de claves externo, cree un servicio de punto de conexión de VPC con el equilibrador de carga de red que creó en el paso anterior. El servicio de punto final de la VPC puede estar en el Cuenta de AWS mismo almacén de claves externo o en uno diferente. Cuenta de AWS

Varios almacenes de claves externos pueden compartir una VPC de Amazon, pero cada almacén de claves externo debe tener su propio servicio de punto de conexión de VPC y un nombre DNS privado.

Siga las instrucciones del tema [Crear un servicio de punto de conexión](#) para crear un servicio de punto de conexión de VPC con los siguientes valores obligatorios. Para los demás campos, acepte los valores predeterminados y proporcione los nombres solicitados.

Campo	Valor
Tipo de equilibrador de carga	Network
Equilibradores de carga disponibles	Elija el equilibrador de carga de red que creó en el paso anterior. Si el nuevo equilibrador de carga no aparece en la lista, compruebe que su estado esté activo. Es posible que el estado del equilibrador de carga tarde unos minutos en cambiar de aprovisionamiento a activo.
Se requiere aceptación	False. Anule la selección de la casilla de verificación. No requieren aceptación. AWS KMS no se puede conectar al servicio de punto final de la VPC sin una aceptación manual. Si se requiere la aceptación, los intentos de crear el almacén de claves externo fallan con una excepción <code>XksProxyInvalidConfigurationException</code>.
Habilitación de nombre DNS privado	Asociar un nombre DNS privado con el servicio
Nombre de DNS privado	Introduzca un nombre DNS privado que sea único en su Región de AWS. El nombre DNS privado debe ser un subdominio de un dominio público de nivel superior. Por ejemplo, si el nombre DNS privado es <code>myproxy-private.xks.example.com</code>, debe ser un subdominio de un dominio público, como <code>xks.example.com</code> o <code>example.com</code>. Este nombre DNS privado debe coincidir con el nombre común del asunto (CN) del certificado TLS configurado en el proxy del almacén de claves externo. Por ejemplo, si el nombre DNS privado es <code>myproxy-private.xks.example.com</code>, el CN del certificado TLS debe ser <code>myproxy-private.xks.example.com</code> o <code>*.xks.example.com</code>. Si el certificado y el nombre DNS privado no coinciden, los intentos de conectar un almacén de claves externo a su proxy del almacén de claves externo fallan con un código de error de conexión de <code>XKS_PROXY_INVALID_</code>

Campo	Valor
	TLS_CONFIGURATION . Para obtener más información, consulte Errores de configuración general .
Tipos de direcciones IP compatibles	IPv4

Paso 5: Verificar el dominio de su nombre DNS privado

Al crear el servicio de punto de conexión de VPC, su estado de verificación de dominio es `pendingVerification`. Antes de utilizar el servicio de punto de conexión de VPC para crear un almacén de claves externo, este estado debe ser `verified`. Para verificar que es el propietario del dominio asociado con su nombre DNS privado, debe crear un registro TXT en un servidor DNS público.

Por ejemplo, si el nombre de DNS privado de su servicio de punto final de VPC es `esmyproxy-private.xks.example.com`, debe crear un registro TXT en un dominio público, como `xks.example.com` o `example.com`, el que sea público. AWS PrivateLink busca el registro TXT primero en `xks.example.com` y después en `example.com`.

Tip

Después de agregar un registro TXT, es posible que el estado de verificación del dominio tarde unos minutos en cambiar de `pendingVerification` a `verify`.

Para empezar, busque el estado de verificación de su dominio mediante uno de los siguientes métodos. Los valores válidos son `verified`, `pendingVerification` y `failed`.

- En la [consola de VPC de Amazon](#), seleccione Endpoint services (Servicios de punto de conexión) y elija su servicio de punto de conexión. En el panel de detalles, consulte Estado de verificación del dominio.
- Utilice la operación [DescribeVpcEndpointServiceConfigurations](#). El valor `State` está en el campo `ServiceConfigurations.PrivateDnsNameConfiguration.State`.

Si el estado de verificación no es `verified`, siga las instrucciones del tema [Verificación de la propiedad del dominio](#) para agregar un registro TXT al servidor de DNS de su dominio y comprobar que el registro TXT esté publicado. A continuación, vuelva a comprobar el estado de la verificación.

No es necesario crear un registro A para el nombre de dominio DNS privado. Cuando AWS KMS crea un punto de enlace de interfaz para el servicio de punto final de la VPC, crea AWS PrivateLink automáticamente una zona alojada con el registro A necesario para el nombre de dominio privado de la AWS KMS VPC. Para almacenes de claves externos con conectividad de servicio de punto de conexión de VPC, esto sucede cuando [conecta su almacén de claves externo](#) a su proxy de almacén de claves externo.

Paso 6: Autorizar AWS KMS la conexión al servicio de punto final de la VPC

Consulte los siguientes procedimientos para gestionar los permisos del servicio de punto de conexión de Amazon VPC. Cada paso depende de la conectividad y la configuración entre el almacén de claves externo, el servicio de punto de conexión de VPC y la Cuenta de AWS.

Same Cuenta de AWS

Si su servicio de punto final de VPC es propiedad de la Cuenta de AWS misma propiedad que su almacén de claves externo, debe agregarlo AWS KMS a la lista Permitir entidades principales de su servicio de punto final de VPC. Esto permite AWS KMS crear puntos de enlace de interfaz para su servicio de punto final de VPC. Si no AWS KMS es un principal permitido, los intentos de crear un almacén de claves externo fallarán con una `XksProxyVpcEndpointServiceNotFoundException` excepción.

Siga las instrucciones del tema [Administrar permisos](#) de la Guía de AWS PrivateLink . Use el siguiente valor obligatorio.

Campo	Valor
AWS KMS ARN	<code>cks.kms.<region>.amazonaws.com</code> Por ejemplo, <code>cks.kms.us-east-1.amazonaws.com</code>

Cross Cuenta de AWS

Si su servicio de punto final de VPC es propiedad de otra persona, Cuenta de AWS debe añadir ambos AWS KMS y su cuenta a la lista de usuarios principales permitidos. Esto

permite que AWS KMS su almacén de claves externo cree puntos de conexión de interfaz para su servicio de puntos de enlace de VPC. Si AWS KMS no es una entidad principal permitida, los intentos de crear un almacén de claves externo fallarán con una excepción `XksProxyVpcEndpointServiceNotFoundException`. Deberá proporcionar el Cuenta de AWS ARN en el que reside el almacén de claves externo.

Siga las instrucciones del tema [Administrar permisos](#) de la Guía de AWS PrivateLink . Use el siguiente valor obligatorio.

Campo	Valor
AWS KMS ARN	<code>cks.kms.<region>.amazonaws.com</code> Por ejemplo, <code>cks.kms.us-east-1.amazonaws.com</code>
Cuenta de AWS ARN	<code>arn:aws:iam:: 111122223333 :role/role_name</code> Por ejemplo, <code>arn:aws:iam:: 123456789012 :role/cks_role</code>

Siguiente: [Creación de un almacén de claves externo](#)

Creación de un almacén de claves externo

Puede crear uno o varios almacenes de claves externos en cada Cuenta de AWS región. Cada almacén de claves externo debe estar asociado a un administrador de claves externo y a un proxy de AWS almacenamiento de claves externo (proxy XKS) que medie la comunicación entre su administrador de claves externo AWS KMS y su administrador de claves externo. Para obtener más información, consulte [Elección de una opción de conectividad proxy del almacén de claves externo](#). Antes de empezar, [confirme que necesita un almacén de claves externo](#). La mayoría de los clientes pueden usar claves KMS respaldadas por material AWS KMS clave.

Tip

Algunos administradores de claves externos proporcionan un método más sencillo para crear un almacén de claves externo. Para obtener más detalles, consulte la documentación del administrador de claves externo.

Antes de crear un almacén de claves externo, debe [cumplir los requisitos previos](#). Durante el proceso de creación, usted especifica las propiedades del almacén de claves externo. Lo que es más importante, debe indicar si su almacén de claves externo AWS KMS utiliza un [punto final público](#) o un [servicio de punto final de VPC](#) para conectarse a su proxy de almacén de claves externo. También especificas los detalles de la conexión, incluido el extremo URI del proxy y la ruta dentro de ese punto final del proxy desde donde se AWS KMS envían las solicitudes de API al proxy.

Consideraciones

- El KMS no puede comunicarse IPv6 con almacenes de claves externos.
- Si utiliza la conectividad de un punto final público, asegúrese de que AWS KMS pueda comunicarse con su proxy a través de Internet mediante una conexión HTTPS. Esto incluye configurar el TLS en el proxy del almacén de claves externo y garantizar que cualquier firewall entre el proxy AWS KMS y el proxy permita el IPv4 tráfico hacia y desde el puerto 443 del proxy. Al crear un almacén de claves externo con conectividad de punto final público, AWS KMS prueba la conexión enviando una solicitud de estado al proxy del almacén de claves externo. Esta prueba comprueba que se puede acceder al punto de conexión y que el proxy del almacén de claves externo aceptará una solicitud firmada con la [credencial de autenticación del proxy del almacén de claves externo](#). Si se produce un error en esta solicitud de prueba, se produce un error en la operación para crear el almacén de claves externo.
- Si utiliza la conectividad al servicio de punto de conexión de VPC, asegúrese de que el equilibrador de carga de red, el nombre DNS privado y el servicio de punto de conexión de VPC estén configurados correctamente y en funcionamiento. Si el proxy del almacén de claves externo no está en la VPC, debe asegurarse de que el servicio de punto final de la VPC pueda comunicarse con el proxy del almacén de claves externo. (AWS KMS comprueba la conectividad del servicio de punto final de la VPC al [conectar el almacén de claves externo](#) a su proxy de almacén de claves externo).
- AWS KMS registra [CloudWatch las métricas y dimensiones de Amazon](#), especialmente para los almacenes de claves externos. Los gráficos de supervisión basados en algunas de estas métricas aparecen en la AWS KMS consola para cada almacén de claves externo. Le recomendamos encarecidamente que utilice estas métricas para crear alarmas que monitoreen su almacén de claves externo. Estas alarmas le avisan de las señales tempranas de problemas operativos y de rendimiento antes de que se produzcan. Para obtener instrucciones, consulte [Monitoreo de almacenes de claves externos](#).

- Los almacenes de claves externos están sujetos a [cuotas de recursos](#). El uso de claves de KMS en un almacén de claves externo está sujeto a [las cuotas de solicitud](#). Revise estas cuotas antes de diseñar la implementación de su almacén de claves externo.

Note

Revise su configuración para ver si hay dependencias circulares que puedan impedir que funcione.

Por ejemplo, si crea un proxy de almacén de claves externo con AWS recursos, asegúrese de que el funcionamiento del proxy no requiera la disponibilidad de una clave de KMS en un almacén de claves externo al que se acceda a través de ese proxy.

Todos los nuevos almacenes de claves externos se crean en un estado desconectado. Antes de poder crear claves de KMS en su almacén de claves externo, debe [conectarlo](#) a su proxy del almacén de claves externo. Para cambiar las propiedades del almacén de claves externo, [edite la configuración del almacén de claves externo](#).

Temas

- [Cumplir los requisitos previos](#)
- [Creación de un nuevo almacén de claves externo](#)

Cumplir los requisitos previos

Antes de crear un almacén de claves externo, debe reunir los componentes necesarios, incluido el [administrador de claves externo](#) que utilizará como soporte para el almacén de claves externo y el [proxy del almacén de claves externo](#), que traduce AWS KMS las solicitudes a un formato que el administrador de claves externo pueda entender.

Los siguientes componentes son necesarios para todos los almacenes de claves externos. Además de estos componentes, debe proporcionar los componentes que admitan la [opción de conectividad al proxy del almacén de claves externo](#) que elija.

 Tip

Su administrador de claves externo puede incluir algunos de estos componentes o puede que estén configurados para usted. Para obtener más detalles, consulte la documentación del administrador de claves externo.

Si va a crear su almacén de claves externo en la AWS KMS consola, tiene la opción de cargar un [archivo de configuración de proxy](#) basado en JSON que especifique la [ruta URI del proxy y la credencial](#) de [autenticación del proxy](#). Algunos proxy del almacén de claves externos generan este archivo por usted. Para obtener más información, consulte la documentación del proxy del almacén de claves externo o el administrador de claves externo.

Administrador de claves externo

Cada almacén de claves externo requiere al menos una instancia del [administrador de claves externo](#). Puede ser un módulo de seguridad de hardware (HSM) físico o virtual o un software de administración de claves.

Puede usar un único administrador de claves, pero le recomendamos al menos dos instancias del administrador de claves relacionadas que compartan claves criptográficas por motivos de redundancia. El almacén de claves externo no requiere el uso exclusivo del administrador de claves externo. Sin embargo, el administrador de claves externo debe tener la capacidad de gestionar la frecuencia prevista de las solicitudes de cifrado y descifrado de los AWS servicios que utilizan claves KMS en el almacén de claves externo para proteger sus recursos. El administrador de claves externo debe configurarse para administrar hasta 1800 solicitudes por segundo y responder dentro del tiempo de espera de 250 milisegundos para cada solicitud. Se recomienda ubicar el administrador de claves externo cerca de an para Región de AWS que el tiempo de ida y vuelta (RTT) de la red sea de 35 milisegundos o menos.

Si su proxy del almacén de claves externo lo permite, puede cambiar el administrador de claves externo que asocia a su proxy del almacén de claves externo, pero el nuevo administrador de claves externo debe ser una copia de seguridad o una instantánea con el mismo material de claves. Si la clave externa que asocia a una clave KMS ya no está disponible para el proxy del almacén de claves externo, AWS KMS no podrás descifrar el texto cifrado con la clave KMS.

El proxy del almacén de claves externo debe poder acceder al administrador de claves externo. Si la [GetHealthStatus](#) respuesta del proxy indica que todas las instancias del administrador de claves

externo lo estánUnavailable, todos los intentos de crear un almacén de claves externo fallan con un. [XksProxyUriUnreachableException](#)

Proxy del almacén de claves externo

Debe especificar un [proxy del almacén de claves externo](#) (proxy XKS) que cumpla con los requisitos de diseño de la [especificación de la API de proxy del almacén de claves externo de AWS KMS](#). Puede desarrollar o comprar un proxy de almacén de claves externo, o utilizar un proxy de almacén de claves externo proporcionado por su administrador de claves externo o integrado en él. AWS KMS recomienda que el proxy de almacenamiento de claves externo esté configurado para gestionar hasta 1800 solicitudes por segundo y responder dentro del tiempo de espera de 250 milisegundos para cada solicitud. Le recomendamos que ubique el administrador de claves externo cerca de un, de Región de AWS modo que el tiempo de ida y vuelta de la red (RTT) sea de 35 milisegundos o menos.

Puede utilizar un proxy del almacén de claves externo para más de un almacén de claves externo, pero cada almacén de claves externo debe tener un punto de conexión y una ruta de URI únicos dentro del proxy del almacén de claves externo para sus solicitudes.

Si utiliza la conectividad a un servicio de punto de conexión de VPC, puede localizar su proxy del almacén de claves externo en su VPC de Amazon, pero no es obligatorio. Puede ubicar su proxy fuera, por ejemplo AWS, en su centro de datos privado, y usar el servicio de punto final de VPC solo para comunicarse con el proxy.

Credencial de autenticación del proxy

Para crear un almacén de claves externo, debe especificar su credencial de autenticación del proxy del almacén de claves externo (XksProxyAuthenticationCredential).

Debe establecer una [credencial de autenticación](#) (XksProxyAuthenticationCredential) para su proxy AWS KMS de almacén de claves externo. AWS KMS se autentica en su proxy firmando sus solicitudes mediante el [proceso de firma de la versión 4 \(SiGv4\)](#) con la credencial de autenticación del proxy del almacén de claves externo. Debe especificar una credencial de autenticación cuando se crea el almacén de claves externo y [se puede cambiar](#) en cualquier momento. Si su proxy rota la credencial, asegúrese de actualizar los valores de la credencial del almacén de claves externo.

La credencial de autenticación del proxy tiene dos partes. Debe proporcionar ambas partes para su almacén de claves externo.

- ID de clave de acceso: identifica la clave de acceso secreta. Puede proporcionar este ID en texto sin formato.

- **Clave de acceso secreta:** la parte secreta de la credencial. AWS KMS cifra la clave de acceso secreta de la credencial antes de almacenarla.

La credencial de SigV4 que se AWS KMS utiliza para firmar las solicitudes al proxy del almacén de claves externo no está relacionada con ninguna de las credenciales de SigV4 asociadas a los principales de sus cuentas. AWS Identity and Access Management AWS No reutilice ninguna credencial SigV4 de IAM para su proxy del almacén de claves externo.

Conectividad de proxy

Para crear un almacén de claves externo, debe especificar su opción de conectividad del proxy del almacén de claves externo (`XksProxyConnectivity`).

AWS KMS puede comunicarse con el proxy de su almacén de claves externo mediante un [punto de conexión público](#) o un servicio de [punto final de Amazon Virtual Private Cloud \(Amazon VPC\)](#). Si bien un punto de conexión público es más sencillo de configurar y mantener, es posible que no cumpla con los requisitos de seguridad de todas las instalaciones. Si elige la opción de conectividad del servicio de punto de conexión de VPC de Amazon, debe crear y mantener los componentes necesarios, incluida una VPC de Amazon con al menos dos subredes en dos zonas de disponibilidad diferentes, un servicio del punto de conexión de VPC con un equilibrador de carga de red y un grupo de destino, y un nombre DNS privado para el servicio de punto de conexión de VPC.

Puede [cambiar la opción de conectividad del proxy](#) para su almacén de claves externo. Sin embargo, debe asegurar la disponibilidad continua del material de claves asociado a las claves de KMS en su almacén de claves externo. De lo contrario, AWS KMS no podrá descifrar ningún texto cifrado con esas claves de KMS.

Para obtener ayuda para decidir qué opción de conectividad de proxy es mejor para su almacén de claves externo, consulte [Elección de una opción de conectividad proxy del almacén de claves externo](#). Para obtener ayuda para crear una configuración de la conectividad del servicio del punto de conexión de VPC, consulte [Configuración de la conectividad del servicio de punto de conexión de VPC](#).

Punto de conexión URI del proxy

Para crear un almacén de claves externo, debe especificar el punto final (`XksProxyUriEndpoint`) que se AWS KMS utiliza para enviar las solicitudes al proxy del almacén de claves externo.

El protocolo debe ser HTTPS. AWS KMS se comunica a través IPv4 del puerto 443. No especifique el puerto en el valor del punto de conexión URI del proxy.

- [Conectividad al punto de conexión público](#): especifique el punto de conexión disponible públicamente para su proxy del almacén de claves externo. Se debe poder acceder a este punto de conexión antes de crear el almacén de claves externo.
- [Conectividad al servicio de punto de conexión de VPC](#): especifique `https://` seguido del nombre DNS privado del servicio de punto de conexión de VPC.

El certificado del servidor TLS configurado en el proxy del almacén de claves externo debe coincidir con el nombre de dominio del punto de conexión URI y debe ser emitido por una autoridad de certificación compatible con los almacenes de claves externos. Para obtener una lista, consulte [Autoridades de certificación de confianza](#). La autoridad de certificación exigirá una prueba de la propiedad del dominio antes de emitir el certificado TLS.

El nombre común del sujeto (CN) en el certificado TLS debe coincidir con el nombre DNS privado. Por ejemplo, si el nombre DNS privado es `myproxy-private.xks.example.com`, el CN del certificado TLS debe ser `myproxy-private.xks.example.com` o `*.xks.example.com`.

Puede [cambiar el punto de conexión de la URI del proxy](#), pero asegúrese de que el proxy del almacén de claves externo tenga acceso al material de claves asociado a las claves de KMS de su almacén de claves externo. De lo contrario, AWS KMS no podrá descifrar ningún texto cifrado con esas claves KMS.

Requisitos de singularidad

- El valor combinado del punto de conexión del URI del proxy (`XksProxyUriEndpoint`) y el valor de la ruta URI del proxy (`XksProxyUriPath`) deben ser únicos en la Cuenta de AWS y en la región.
- Los almacenes de claves externos con conectividad al punto de conexión público pueden compartir el mismo punto de conexión URI del proxy, siempre que tengan valores de ruta URI de proxy diferentes.
- Un almacén de claves externo con conectividad de punto final público no puede usar el mismo valor de punto final de URI de proxy que ningún almacén de claves externo con conectividad de servicios de punto final de VPC en el mismo nivel Región de AWS, incluso si los almacenes de claves están en diferentes lugares. Cuentas de AWS
- Cada almacén de claves externo con conectividad al punto de conexión de VPC debe tener su propio nombre DNS privado. El punto final del URI del proxy (nombre de DNS privado) debe ser único en la región Cuenta de AWS and.

Ruta URI del proxy

Para crear un almacén de claves externo, debe especificar la ruta base en el proxy del almacén de claves externo al [proxy requerido APIs](#). El valor debe empezar / y terminar con/kms/xks/v1, donde v1 representa la versión de la AWS KMS API del proxy del almacén de claves externo. Esta ruta puede incluir un prefijo opcional entre los elementos necesarios, por ejemplo /example-prefix/kms/xks/v1. Para encontrar este valor, consulte la documentación de su proxy del almacén de claves externo.

AWS KMS envía las solicitudes de proxy a la dirección especificada mediante la concatenación del punto final del URI del proxy y la ruta del URI del proxy. Por ejemplo, si el punto final del URI del proxy es `https://myproxy.xks.example.com` y la ruta del URI del proxy es `/kms/xks/v1`, AWS KMS envía sus solicitudes de API de proxy a `https://myproxy.xks.example.com/kms/xks/v1`.

Puede [cambiar la ruta URI del proxy](#), pero asegúrese de que el proxy del almacén de claves externo tenga acceso al material de claves asociado a las claves de KMS de su almacén de claves externo. De lo contrario, AWS KMS no podrá descifrar ningún texto cifrado con esas claves de KMS.

Requisitos de singularidad

- El valor combinado del punto de conexión del URI del proxy (`XksProxyUriEndpoint`) y el valor de la ruta URI del proxy (`XksProxyUriPath`) deben ser únicos en la Cuenta de AWS y en la región.

Servicio de punto de conexión de VPC

Especifica el nombre del servicio de punto de conexión de VPC de Amazon que se utiliza para comunicarse con el proxy del almacén de claves externo. Este componente solo es necesario para los almacenes de claves externos que utilizan la conectividad a servicios de punto de conexión de VPC. Para obtener ayuda para establecer y configurar el servicio de punto de conexión de VPC para un almacén de claves externo, consulte [Configuración de la conectividad del servicio de punto de conexión de VPC](#).

El servicio de punto de conexión de VPC debe tener las siguientes propiedades:

- El servicio de punto de conexión de VPC puede estar en la misma Cuenta de AWS que el almacén de claves externo o en una diferente.

- El servicio de punto final de la VPC debe residir en el mismo lugar Región de AWS que el almacén de claves externo.
- Deberás proporcionar el Cuenta de AWS ID del servicio de punto final de la VPC si reside en otro. Cuenta de AWS
- Debe tener un equilibrador de carga de red (NLB) conectado a al menos dos subredes, cada una en una zona de disponibilidad diferente.
- La lista de principales permitidos para el servicio de punto final de la VPC debe incluir AWS KMS el principal de servicio de la región `cks.kms.<region>.amazonaws.com`; por ejemplo, `cks.kms.us-east-1.amazonaws.com`
- Si su servicio de puntos de conexión de Amazon VPC es propiedad de una persona Cuenta de AWS distinta a la Cuenta de AWS propietaria del almacén de claves externo (XKS), también tendrá que permitir que XKS acceda al servicio de puntos de conexión de VPC. Para ello, incluya [el Cuenta de AWS ID de XKS en una lista de permitidos como principal](#) para el servicio de puntos de conexión de Amazon VPC.
- No debe requerir la aceptación de las solicitudes de conexión.
- Debe tener un nombre DNS privado dentro de un dominio público de nivel superior. Por ejemplo, podría tener un nombre DNS privado `myproxy-private.xks.example.com` en el dominio `xks.example.com` público.

El nombre DNS privado de un almacén de claves externo con conectividad al servicio de punto de conexión de VPC debe ser único en su Región de AWS.

- El [estado de verificación de dominio](#) del dominio de su nombre DNS privado debe ser `verified`.
- El certificado de servidor TLS configurado en el proxy del almacén de claves externo debe especificar el nombre de host DNS privado en el que se puede acceder al punto de conexión.

Requisitos de singularidad

- Los almacenes de claves externos con conectividad al punto de conexión de VPC pueden compartir una Amazon VPC, pero cada almacén de claves externo debe tener su propio servicio de punto de conexión de VPC y nombre DNS privado.

Archivo de configuración del proxy

Un archivo de configuración del proxy es un archivo opcional basado en JSON que contiene valores para la [ruta URI del proxy](#) y las propiedades de las [credenciales de autenticación del proxy](#)

del almacén de claves externo. Al crear o [editar un almacén de claves externo](#) en la consola de AWS KMS, puede cargar un archivo de configuración del proxy para proporcionar los valores de configuración del almacén de claves externo. El uso de este archivo evita errores al escribir y pegar, y garantiza que los valores del almacén de claves externo coincidan con los valores del proxy del almacén de claves externo.

Los archivos de configuración del proxy los genera el proxy del almacén de claves externo. Para saber si su proxy del almacén de claves externo ofrece un archivo de configuración del proxy, consulte la documentación del proxy del almacén de claves externo.

A continuación, se muestra un ejemplo de un archivo de configuración de proxy bien estructurado con valores ficticios.

```
{
  "XksProxyUriPath": "/example-prefix/kms/xks/v1",
  "XksProxyAuthenticationCredential": {
    "AccessKeyId": "ABCDE12345670EXAMPLE",
    "RawSecretAccessKey": "0000EXAMPLEFA5FT0mCc3DrGUe2sti527BitkQ0Zr9M09+vE="
  }
}
```

Solo puede cargar un archivo de configuración de proxy cuando cree o edite un almacén de claves externo en la consola. AWS KMS No puede usarlo con las [UpdateCustomKeyStore](#) operaciones [CreateCustomKeyStore](#), pero puede usar los valores del archivo de configuración del proxy para asegurarse de que los valores de los parámetros son correctos.

Creación de un nuevo almacén de claves externo

Una vez que haya reunido los requisitos previos necesarios, puede crear un nuevo almacén de claves externo en la AWS KMS consola o mediante la [CreateCustomKeyStore](#) operación.

Uso de la consola AWS KMS

Antes de crear un almacén de claves externo, [elija el tipo de conectividad de proxy](#) y asegúrese de haber creado y configurado todos los [componentes necesarios](#). Si necesita ayuda para encontrar alguno de los valores requeridos, consulte la documentación del proxy de su almacén de claves externo o del software de administración de claves.

 Note

Al crear un almacén de claves externo en Consola de administración de AWS, puede cargar un archivo de configuración de proxy basado en JSON con los valores de la [ruta URI del proxy y la credencial](#) de [autenticación del proxy](#). Algunos proxy generan este archivo por usted. No es obligatorio.

1. [Inicia sesión en la consola AWS Key Management Service \(AWS KMS\) Consola de administración de AWS y ábrela en /kms. https://console.aws.amazon.com](#)
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Custom key stores (Almacenes de claves personalizados), External key stores (Almacenes de claves externos).
4. Seleccione Create external key store (Crear almacén de claves externo).
5. Escriba un nombre fácil de recordar para el almacén de claves externo. El nombre debe ser único entre todos los almacenes de claves externos de su cuenta.

 Important

No incluya información confidencial en este campo. Este campo puede mostrarse en texto plano en los CloudTrail registros y otros resultados.

6. Elija el tipo de [conectividad del proxy](#).

Su elección de conectividad del proxy determina los [componentes necesarios](#) para su proxy del almacén de claves externo. Si necesita ayuda para tomar esta decisión, consulte [Elección de una opción de conectividad proxy del almacén de claves externo](#).

- a. Seleccione el servicio de punto de conexión de VPC entre cuentas si su servicio de punto de conexión de VPC reside en una Cuenta de AWS diferente. A continuación, introduzca el Cuenta de AWS ID del propietario del punto final de la VPC en el campo ID de la cuenta del propietario del servicio del punto final de la VPC.
- b. Elija o introduzca el nombre del [servicio de punto de conexión de VPC](#) para este almacén de claves externo. Este paso solo aparece cuando el tipo de conectividad al proxy del almacén de claves externo es el servicio de punto de conexión de VPC.

El servicio de punto final de la VPC y su VPCs deben cumplir los requisitos de un almacén de claves externo. Para obtener más información, consulte [the section called “Cumplir los requisitos previos”](#).

7. Elija o introduzca el nombre del [servicio de punto de conexión de VPC](#) para este almacén de claves externo. Este paso solo aparece cuando el tipo de conectividad al proxy del almacén de claves externo es el servicio de punto de conexión de VPC.

El servicio de punto final de la VPC y su VPCs deben cumplir los requisitos de un almacén de claves externo. Para obtener más información, consulte [the section called “Cumplir los requisitos previos”](#).

8. Introduzca el [punto de conexión de la URI de su proxy](#). El protocolo debe ser HTTPS. AWS KMS se comunica a través IPv4 del puerto 443. No especifique el puerto en el valor del punto de conexión URI del proxy.

Si AWS KMS reconoce el servicio de punto final de VPC que especificó en el paso anterior, completará este campo por usted.

Para la conectividad al punto de conexión público, introduzca una URI de punto de conexión disponible públicamente. Para la conectividad de punto de conexión de VPC, introduzca `https://` seguido del nombre DNS privado del servicio de punto de conexión de VPC.

9. Para introducir los valores del prefijo de [ruta URI del proxy](#) y de [la credencial de autenticación del proxy](#), cargue un archivo de configuración de proxy o introduzca los valores manualmente.
 - Si tiene un [archivo de configuración de proxy](#) opcional que contiene valores para la [ruta URI del proxy](#) y la [credencial de autenticación del proxy](#), elija Upload configuration file (Cargar archivo de configuración). Siga los pasos para cargar el archivo.

Cuando se carga el archivo, la consola muestra los valores del archivo en campos editables. Puede cambiar los valores ahora o [editarlos](#) después de crear el almacén de claves externo.

Para mostrar el valor de la clave de acceso secreta, seleccione Show secret access key (Mostrar clave de acceso secreta).

- Si no tiene un archivo de configuración de proxy, puede introducir la ruta URI del proxy y los valores de las credenciales de autenticación del proxy manualmente.
 - a. Si no dispone de un archivo de configuración de proxy, puede introducir la URI de proxy manualmente. La consola proporciona el valor/kms/xks/v1 requerido.

Si la [ruta URI de su proxy](#) incluye un prefijo opcional, como el `example-prefix` en `/example-prefix/kms/xks/v1`, introduzca el prefijo en el campo del prefijo de ruta URI del proxy. De lo contrario, deje el campo en blanco.

- b. Si no dispone de un archivo de configuración de proxy, puede introducir su [credencial de autenticación del proxy](#) manualmente. Se requieren tanto el ID de clave de acceso como la clave de acceso secreta.
 - En Proxy credential: Access key ID (Credencial de proxy: ID de clave de acceso), introduzca el ID de clave de acceso de la credencial de autenticación del proxy. El ID de clave de acceso identifica la clave de acceso secreta.
 - En Proxy credential: Secret access key (Credencial de proxy: clave de acceso secreta), introduzca el ID de clave de acceso secreta de la credencial de autenticación del proxy.

Para mostrar el valor de la clave de acceso secreta, seleccione Show secret access key (Mostrar clave de acceso secreta).

Este procedimiento no establece ni cambia la credencial de autenticación que estableció en su proxy del almacén de claves externo. Simplemente asocia estos valores con su almacén de claves externo. Para obtener información sobre la configuración, el cambio y la credencial de autenticación del proxy rotativo, consulte la documentación del proxy del almacén de claves externo o del software de administración de claves.

Si la credencial de autenticación del proxy cambia, [edite la configuración de credenciales](#) del almacén de claves externo.

10. Seleccione Create external key store (Crear almacén de claves externo).

Si el proceso se ejecuta correctamente, el nuevo almacén de claves externo aparecerá en la lista de almacenes de claves externos de la cuenta y la región. Si el procedimiento da error, aparecerá un mensaje de error donde se describe el problema y se explica cómo resolverlo. Si necesita más ayuda, consulte [CreateKey errores en la clave externa](#).

Siguiente: los nuevos almacenes de claves externos no se conectan de forma automática. Para poder crear AWS KMS keys en su almacén de claves externo, debe [conectar el almacén de claves externo](#) a su proxy de almacén de claves externo.

Uso de la AWS KMS API

Puede utilizar la [CreateCustomKeyStore](#) operación para crear un nuevo almacén de claves externo. Para obtener ayuda para encontrar los parámetros requeridos, consulte la documentación del proxy de su almacén de claves externo o del software de administración de claves.

Tip

No puede cargar un [archivo de configuración de proxy](#) cuando utilice la operación `CreateCustomKeyStore`. Sin embargo, puede utilizar los valores del archivo de configuración del proxy para asegurarse de que los valores de los parámetros son correctos.

Para crear un almacén de claves externo, la operación `CreateCustomKeyStore` requiere los siguientes valores de parámetros.

- `CustomKeyStoreName`: un nombre fácil de recordar para el almacén de claves externo que es exclusivo de la cuenta.

Important

No incluya información confidencial en este campo. Este campo puede mostrarse en texto plano en CloudTrail los registros y otros resultados.

- `CustomKeyStoreType`: especifique `EXTERNAL_KEY_STORE`.
- [XksProxyConnectivity](#): especifique `PUBLIC_ENDPOINT` o `VPC_ENDPOINT_SERVICE`.
- [XksProxyAuthenticationCredential](#): especifique el ID de clave de acceso y la clave de acceso secreta.
- [XksProxyUriEndpoint](#): el punto de conexión que AWS KMS utiliza para comunicarse con el proxy de su almacén de claves externo.
- [XksProxyUriPath](#)— La ruta desde el proxy hasta el proxy APIs.
- [XksProxyVpcEndpointServiceName](#): es obligatorio solo cuando su valor `XksProxyConnectivity` es `VPC_ENDPOINT_SERVICE`.

 Note

Si utiliza la AWS CLI versión 1.0, ejecute el siguiente comando antes de especificar un parámetro con un valor HTTP o HTTPS, como el XksProxyUriEndpoint parámetro.

```
aws configure set cli_follow_urlparam false
```

De lo contrario, la AWS CLI versión 1.0 reemplaza el valor del parámetro por el contenido que se encuentra en esa dirección URI, lo que provoca el siguiente error:

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
https:// : received non 200 status code of 404
```

En los siguientes ejemplos se utilizan valores ficticios. Antes de ejecutar el comando, sustituya los valores por valores válidos para el almacén de claves externo.

Cree un almacén de claves externo con conectividad a los puntos de conexión públicos.

```
$ 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>
```

Cree un almacén de claves externo con conectividad al servicio de punto de conexión de 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>
```

Si la operación se ejecuta correctamente, `CreateCustomKeyStore` devolverá el ID del almacén de claves personalizado, tal y como se muestra en la siguiente respuesta de ejemplo.

```
{
  "CustomKeyId": cks-1234567890abcdef0
}
```

Si la operación falla, corrija el error que indica la excepción e inténtelo de nuevo. Para obtener ayuda adicional, consulte [Solución de problemas de almacenes de claves externos](#).

Siguiente: para usar el almacén de claves externo, [conéctelo a su proxy del almacén de claves externo](#).

Edición de las propiedades del almacén de claves externo

Puede editar las propiedades seleccionadas de un almacén de claves externo existente.

Puede editar algunas propiedades mientras el almacén de claves externo esté conectado o desconectado. Para otras propiedades, primero debe [desconectar el almacén de claves externo](#) de su proxy del almacén de claves externo. El [estado de conexión](#) del almacén de claves externo debe ser `DISCONNECTED`. Mientras su almacén de claves externo esté desconectado, puede administrar el almacén de claves y sus claves de KMS, pero no puede crear ni usar las claves de KMS en el almacén de claves externo. Para averiguar el [estado de conexión](#) del almacén de claves externo, utilice la [DescribeCustomKeyStores](#) operación o consulte la sección Configuración general de la página de detalles del almacén de claves externo.

Antes de actualizar las propiedades del almacén de claves externo, AWS KMS envía una [GetHealthStatus](#) solicitud al proxy del almacén de claves externo con los nuevos valores. Si la solicitud se realiza correctamente, indica que puede conectarse y autenticarse en un proxy del almacén de claves externo con los valores de propiedad actualizados. Si la solicitud falla, la operación de edición falla con una excepción que identifica el error.

Cuando finalice la operación de edición, los valores de las propiedades actualizados del almacén de claves externo aparecerán en la AWS KMS consola y en la `DescribeCustomKeyStores` respuesta. No obstante, pueden pasar hasta cinco minutos hasta que los cambios surtan efecto.

Si edita el almacén de claves externo en la AWS KMS consola, tiene la opción de cargar un [archivo de configuración del JSON-based proxy](#) que especifique la [ruta del URI del proxy](#) y la [credencial de autenticación del proxy](#). Algunos proxy del almacén de claves externos generan este archivo por

usted. Para obtener más información, consulte la documentación del proxy del almacén de claves externo o el administrador de claves externo.

 Warning

Los valores actualizados de las propiedades deben conectar el almacén de claves externo a un proxy para el mismo administrador de claves externo que los valores anteriores, o para obtener una copia de seguridad o una instantánea del administrador de claves externo con las mismas claves criptográficas. Si su almacén de claves externo pierde permanentemente el acceso a las claves externas asociadas a sus claves de KMS, el texto cifrado encriptado con esas claves externas no se podrá recuperar. En concreto, cambiar la conectividad del proxy de un almacén de claves externo puede AWS KMS impedir el acceso a las claves externas.

 Tip

Algunos administradores de claves externos proporcionan un método más sencillo para editar las propiedades de un almacén de claves externo. Para obtener más detalles, consulte la documentación del administrador de claves externo.

Puede cambiar las siguientes propiedades de un almacén de claves externo.

Propiedades editables del almacén de claves externo	Cualquier estado de conexión	Estado Disconnected (Desconectado) obligatorio
Nombre del almacén de claves personalizadas		
Un nombre fácil de recordar obligatorio para el almacén de claves personalizado.		
 Important No incluya información confidencial en este campo. Este campo puede mostrarse en		

Propiedades editables del almacén de claves externo	Cualquier estado de conexión	Estado Disconnected (Desconectado) obligatorio
texto plano en CloudTrail los registros y otros resultados.		
Credencial de autenticación proxy () XksProxyAuthenticationCredential (Debe especificar el ID de clave de acceso y la clave de acceso secreta, incluso si cambia solo un elemento).	✓	
Ruta de URI del proxy () XksProxyUriPath	✓	
Conectividad proxy (XksProxyConnectivity) (También debe actualizar el punto de conexión de la URI del proxy. Si va a cambiar a la conectividad al servicio de punto de conexión de VPC, debe especificar un nombre de servicio de punto de conexión de VPC del proxy).		✓
Punto final del URI del proxy (XksProxyUriEndpoint) Si cambia la URI del punto de conexión del proxy, es posible que también tenga que cambiar el certificado TLS asociado.		✓
Nombre del servicio de punto final de VPC proxy () XksProxyVpcEndpointServiceName (Este campo es obligatorio para la conectividad al servicio de punto de conexión de VPC)		✓

Propiedades editables del almacén de claves externo	Cualquier estado de conexión	Estado Disconnected (Desconectado) obligatorio

[Propietario del servicio de punto final de VPC \(\)](#)

XksProxyVpcEndpointServiceOwner



(Este campo es obligatorio para la conectividad al servicio de punto de conexión de VPC)

Edición de sus propiedades del almacén de claves externo

Puede editar las propiedades del almacén de claves externo en la AWS KMS consola o mediante la [UpdateCustomKeyStore](#) operación.

Uso de AWS KMS consola

Al editar un almacén de claves, puede cambiar cualquiera de los valores editables. Algunos cambios requieren que el almacén de claves externo esté desconectado de su proxy del almacén de claves externo.

Si está editando la ruta URI del proxy o la credencial de autenticación del proxy, puede introducir los nuevos valores o cargar un [archivo de configuración del proxy](#) del almacén de claves externo que incluya los nuevos valores.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Custom key stores (Almacenes de claves personalizados), External key stores (Almacenes de claves externos).
4. Elija el almacén de claves que quiere editar.
 - Si es necesario, desconecte el almacén de claves externo de su proxy del almacén de claves externo. En el menú Key store actions (Acciones del almacén de claves), seleccione Disconnect (Desconectar).
5. Desde el menú Key store actions (Acciones del almacén de claves), seleccione Edit (Editar).

6. Cambie una o más de las propiedades editables del almacén de claves externo. También puede cargar un [archivo de configuración del proxy](#) con valores para la ruta URI del proxy y la credencial de autenticación del proxy. Puede utilizar un archivo de configuración de proxy incluso si algunos valores especificados en el archivo no han cambiado.
7. Elija Update external key store (Actualizar el almacén de claves externo).
8. Revise la advertencia y, si decide continuar, confirme la advertencia y, a continuación, seleccione Update external key store (Actualizar almacén de claves externo).

Si el procedimiento se ejecuta correctamente, aparecerá un mensaje donde se describen las propiedades que ha editado. Si el procedimiento da error, aparecerá un mensaje de error donde se describe el problema y se explica cómo resolverlo.

9. Si es necesario, vuelva a conectar el almacén de claves externo. En el menú Key store actions (Acciones del almacén de claves), seleccione Connect (Conectar).

Puede dejar el almacén de claves externo desconectado. Pero mientras esté desconectado, no podrá crear las claves de KMS en el almacén de claves externo ni usar las claves de KMS del almacén de claves externo en [operaciones criptográficas](#).

Uso de AWS KMS API

Para cambiar las propiedades de un almacén de claves externo, utilice la [UpdateCustomKeyStore](#) operación. Puede cambiar varias propiedades de un almacén de claves externo en la misma operación. Si la operación se realiza correctamente, AWS KMS devuelve una respuesta HTTP 200 y un objeto JSON sin propiedades.

Utilice el parámetro CustomKeyStoreId para identificar el almacén de claves externo. Utilice los demás parámetros para cambiar las propiedades. No puede usar un [archivo de configuración del proxy](#) con la operación UpdateCustomKeyStore. El archivo de configuración del proxy solo es compatible con la AWS KMS consola. Sin embargo, puede utilizar el archivo de configuración del proxy como ayuda para determinar los valores de los parámetros correctos para el proxy del almacén de claves externo.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Antes de empezar, [si es necesario, desconecte el almacén de claves externo](#) de su proxy del almacén de claves externo. Tras la actualización, si es necesario, puede [volver a conectar el almacén de claves externo](#) a su proxy del almacén de claves externo. Puede dejar el almacén de

claves externo en el estado desconectado, pero deberá volver a conectarlo antes de crear claves de KMS nuevas en el almacén de claves, o para utilizar las claves de KMS existentes en el almacén de claves para operaciones criptográficas.

Note

Si utiliza la AWS CLI versión 1.0, ejecute el siguiente comando antes de especificar un parámetro con un valor HTTP o HTTPS, como el `XksProxyUriEndpoint` parámetro.

```
aws configure set cli_follow_urlparam false
```

De lo contrario, la AWS CLI versión 1.0 reemplaza el valor del parámetro por el contenido que se encuentra en esa dirección URI, lo que provoca el siguiente error:

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve  
https:// : received non 200 status code of 404
```

Cambio del nombre del almacén de claves externo

En el primer ejemplo, se utiliza la [UpdateCustomKeyStore](#) operación para cambiar el nombre descriptivo del almacén de claves externo a `XksKeyStore`. El comando utiliza el parámetro `CustomKeyStoreId` para especificar el almacén de claves personalizado y `CustomKeyStoreName` para especificar el nuevo nombre del almacén de claves personalizado. Sustituya todos los valores de ejemplo por valores reales para su almacén de claves externo.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-  
custom-key-store-name XksKeyStore
```

Cambio de la credencial de autenticación del proxy

En el siguiente ejemplo, se actualiza la credencial de autenticación del proxy que AWS KMS utiliza para autenticarse en el proxy del almacén de claves externo. Puede usar un comando como este para actualizar la credencial si está rotada en su proxy.

Primero actualice la credencial del proxy del almacén de claves externo. A continuación, utilice esta función para informar del cambio a AWS KMS. (Su proxy admitirá brevemente la credencial antigua y la nueva para que tenga tiempo de actualizarla). AWS KMS

Siempre debe especificar el ID de clave de acceso y la clave de acceso secreta en la credencial, incluso si solo se cambia un valor.

Los dos primeros comandos configuran variables para conservar los valores de las credenciales. Las operaciones `UpdateCustomKeyStore` utilizan el parámetro `CustomKeyId` para identificar el almacén de claves externo. Utiliza el parámetro `XksProxyAuthenticationCredential` con sus campos `AccessKeyId` y `RawSecretAccessKey` para especificar la nueva credencial. Sustituya todos los valores de ejemplo por valores reales para su almacén de claves externo.

```
$ 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
```

Cambio de la ruta URI del proxy

El siguiente ejemplo actualiza la ruta URI del proxy (`XksProxyUriPath`). La combinación del punto final del URI del proxy y la ruta del URI del proxy debe ser única en la región Cuenta de AWS and. Sustituya todos los valores de ejemplo por valores reales para su almacén de claves externo.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
  --xks-proxy-uri-path /kms/xks/v1
```

Cambio a la conectividad del servicio del punto de conexión de VPC

En el siguiente ejemplo, se utiliza la [UpdateCustomKeyStore](#) operación para cambiar el tipo de conectividad del proxy del almacén de claves externo a `VPC_ENDPOINT_SERVICE`. Para realizar este cambio, debe especificar los valores necesarios para la conectividad del servicio de punto de conexión de VPC, incluido el nombre del servicio de punto de conexión de VPC (`XksProxyVpcEndpointServiceName`) y un valor de punto de conexión URI del proxy (`XksProxyUriEndpoint`) que incluya el nombre DNS privado del servicio de punto de conexión de VPC. Sustituya todos los valores de ejemplo por valores reales para su almacén de claves externo.

```
$ 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
```

Cambio a la conectividad de puntos de conexión públicos

En el siguiente ejemplo, se cambia el tipo de conectividad del proxy del almacén de claves externo a `PUBLIC_ENDPOINT`. Al realizar este cambio, debe actualizar el valor del punto de conexión URI (`XksProxyUriEndpoint`) del proxy. Sustituya todos los valores de ejemplo por valores reales para su almacén de claves externo.

Note

La conectividad al punto de conexión de VPC proporciona mayor seguridad que la conectividad al punto de conexión público. Antes de cambiar a la conectividad al punto de conexión público, considere otras opciones, como localizar el proxy del almacén de claves externo en las instalaciones y utilizar la VPC solo para la comunicación.

```
$ 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
```

Visualización de almacenes de claves externos

Puede ver los almacenes de claves externos de cada cuenta y región mediante la AWS KMS consola o mediante la [DescribeCustomKeyStores](#) operación.

Al consultar un almacén de claves externo, puede ver lo siguiente:

- Información básica sobre el almacén de claves, incluido su nombre fácil de recordar, ID, tipo de almacén de claves y fecha de creación.
- Información de configuración para el [proxy del almacén de claves externo](#), incluidos el [tipo de conectividad](#), el [punto de conexión URI del proxy](#) y su [ruta](#) y el [ID de clave de acceso](#) de su [credencial de autenticación de proxy](#) actual.
- Si el proxy del almacén de claves externo utiliza la [conectividad al servicio de punto de conexión de VPC](#), la consola muestra el nombre del servicio de punto de conexión de VPC.
- El [estado de conexión](#) actual.

Note

Un valor de estado de conexión Disconnected (Desconectado) indica que el almacén de claves externo nunca se ha conectado o que se ha desconectado intencionadamente de su proxy del almacén de claves externo. Sin embargo, si los intentos de usar una clave de KMS en un almacén de claves externo conectado no son fructíferos, puede deberse a un problema con el almacén de claves externo o el proxy. Para obtener ayuda, consulte [Errores de conexión del almacén de claves externo](#).

- Una sección de [monitorización](#) con gráficos de [CloudWatch las métricas de Amazon](#) diseñada para ayudarte a detectar y resolver problemas con tu almacén de claves externo. Si necesitas ayuda para interpretar los gráficos, utilizarlos en la planificación y la resolución de problemas y crear CloudWatch alarmas en función de las métricas de los gráficos, consulta [Monitoreo de almacenes de claves externos](#).

Propiedades del almacén de claves externo

Las siguientes propiedades de un almacén de claves externo están visibles en la AWS KMS consola y en la [DescribeCustomKeyStores](#) respuesta.

Propiedades del almacén de claves personalizado

Los siguientes valores aparecen en la sección de configuración general de la página de detalles de cada almacén de claves personalizado. Estas propiedades se aplican a todos los almacenes de claves personalizados, incluidos los almacenes de AWS CloudHSM claves y los almacenes de claves externos.

ID del almacén de claves personalizadas

Un identificador único que se AWS KMS asigna al almacén de claves personalizado.

Nombre del almacén de claves personalizadas

Un nombre fácil de recordar que asigna al almacén de claves personalizado al crearlo. Puede cambiar este valor en cualquier momento.

Tipo de almacén de claves personalizado

El tipo de almacén de claves personalizado. Los valores válidos son AWS CloudHSM (AWS_CLOUDHSM) o Almacén de claves externo (EXTERNAL_KEY_STORE). No se puede cambiar el tipo después de crear el almacén de claves personalizado.

Fecha de creación

Fecha en la que se creó el almacén de claves personalizado. Este valor se muestra en la hora local de la Región de AWS.

Estado de la conexión

Indica si el almacén de claves personalizado está conectado a su almacén de claves de respaldo. El estado de conexión será DISCONNECTED solo si el almacén de claves personalizado nunca se ha conectado al almacén de claves de respaldo o se ha desconectado intencionadamente. Para obtener más información, consulte [the section called “Estado de la conexión”](#).

Propiedades de configuración del almacén de claves externo

Los siguientes valores aparecen en la sección de configuración del proxy del almacén de claves externo de la página de detalles de cada almacén de claves externo y en el XksProxyConfiguration elemento de la [DescribeCustomKeyStores](#) respuesta. Para obtener una descripción detallada de cada campo, incluidos los requisitos de exclusividad y ayuda para determinar el valor correcto de cada campo, consulte [the section called “Cumplir los requisitos previos”](#) en el tema Creación de un almacén de claves externo.

Conectividad de proxy

Indica si el almacén de claves externo utiliza [conectividad de puntos de conexión públicos](#) o [conectividad del servicio de punto de conexión de VPC](#).

Punto de conexión URI del proxy

El punto final que se AWS KMS utiliza para conectarse al [proxy del almacén de claves externo](#).

Ruta URI del proxy

La ruta desde el punto final del URI del proxy desde el que se AWS KMS envían [las solicitudes de API del proxy](#).

Credencial de proxy: ID de clave de acceso

Parte de la [credencial de autenticación de proxy](#) que establece en el proxy de su almacén de claves externo. El ID de clave de acceso identifica la clave de acceso secreta de la credencial.

AWS KMS utiliza el proceso de firma SigV4 y la credencial de autenticación del proxy para firmar sus solicitudes en el proxy del almacén de claves externo. La credencial de la firma permite que el proxy del almacén de claves externo autentique las solicitudes en tu nombre. AWS KMS

Nombre del servicio de punto de conexión de VPC

El nombre del servicio de punto de conexión de VPC de Amazon que admite su almacén de claves externo. Este valor solo aparece cuando el almacén de claves externo utiliza la [conectividad del servicio de punto de conexión de VPC](#). Puede localizar su proxy de almacén de claves externo en la VPC o utilizar el servicio de punto de conexión de VPC para comunicarse de forma segura con su proxy de almacén de claves externo.

ID de propietario de servicio de punto de conexión de VPC

El nombre del servicio de punto de conexión de Amazon VPC que admite su almacén de claves externo. Este valor solo aparece cuando el almacén de claves externo utiliza la [conectividad del servicio de punto de conexión de VPC](#). Puede localizar su proxy de almacén de claves externo en la VPC o utilizar el servicio de punto de conexión de VPC para comunicarse de forma segura con su proxy de almacén de claves externo.

Visualización de sus propiedades del almacén de claves externo

Puede ver su almacén de claves externo y sus propiedades asociadas en la AWS KMS consola o mediante la [DescribeCustomKeyStores](#) operación.

Uso de la AWS KMS consola

Para ver los almacenes de claves externos de una cuenta y región determinados, use el siguiente procedimiento.

1. Inicie sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrala en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Custom key stores (Almacenes de claves personalizados), External key stores (Almacenes de claves externos).

4. Para ver información detallada sobre un almacén de claves externo, elija el nombre de almacén de claves.

Uso de la API AWS KMS

Para ver los almacenes de claves externos, utilice la [DescribeCustomKeyStores](#) operación. De forma predeterminada, esta operación devuelve los almacenes de claves personalizados de la cuenta y región. Pero puede usar el parámetro `CustomKeyStoreId` o `CustomKeyStoreName` (pero no ambos) para limitar el resultado de un almacén de claves personalizado determinado.

Para los almacenes de claves personalizados, el resultado consiste en el ID, el nombre y el tipo del almacén de claves personalizado y el [estado de conexión](#) del almacén de claves. Si el estado de conexión es `FAILED`, el resultado también incluirá un `ConnectionErrorCode` que describe el motivo del error. Para obtener ayuda para interpretar el `ConnectionErrorCode` para un almacén de claves externo, consulte [Códigos de error de conexión para almacenes de claves externos](#).

Para los almacenes de claves externos, el resultado también incluye el elemento `XksProxyConfiguration`. Este elemento incluye el [tipo de conectividad](#), el [punto de conexión URI del proxy](#), [la ruta URI del proxy](#) y el ID de clave de acceso de la [credencial de autenticación del proxy](#).

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Por ejemplo, el siguiente comando devuelve todos los almacenes de claves personalizados de la cuenta y la región. Puede usar los parámetros `Limit` y `Marker` para desplazarse por los almacenes de claves personalizados del resultado.

```
$ aws kms describe-custom-key-stores
```

El siguiente comando usa el parámetro `CustomKeyStoreName` para obtener únicamente el almacén de claves externo con el nombre fácil de recordar `ExampleXksPublic`. Este ejemplo de almacén de claves utiliza conectividad a puntos de conexión públicos. Está conectado a su proxy del almacén de claves externo.

```
$ 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"
  }
}
```

El siguiente comando obtiene un ejemplo de almacén de claves externo con conectividad del servicio de punto de conexión de VPC. En este ejemplo, el almacén de claves externo está conectado a su proxy del almacén de claves externo.

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

Un [ConnectionState](#) `Disconnected` indica que no se ha conectado nunca un almacén de claves externo o que se ha desconectado de su proxy del almacén de claves externo de forma

intencionada. Sin embargo, si los intentos de usar una clave de KMS en un almacén de claves externo conectado no son fructíferos, puede deberse a un problema con el proxy del almacén de claves externo u otros componentes externos.

Si el `ConnectionState` del almacén de claves externo es `FAILED`, la respuesta `DescribeCustomKeyStores` incluirá un elemento `ConnectionErrorCode` que explica el motivo del error.

Por ejemplo, en el siguiente resultado, el `XKS_PROXY_TIMED_OUT` valor indica que AWS KMS se puede conectar al proxy del almacén de claves externo, pero la conexión falló porque el proxy del almacén de claves externo no respondió AWS KMS en el tiempo asignado. Si ve este código de error de conexión varias veces, comuníquese al proveedor del proxy del almacén de claves externo. Para obtener ayuda con esto y los errores de conexión, consulte [Solución de problemas de almacenes de claves externos](#).

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-9876543210fedcba9",
      "CustomKeyName": "ExampleXksVpc",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "XKS_PROXY_TIMED_OUT",
      "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"
      }
    }
  ]
}
```

Monitoreo de almacenes de claves externos

AWS KMS recopila las métricas de cada interacción con un almacén de claves externo y las publica en su CloudWatch cuenta. Estas métricas se utilizan para generar los gráficos en la sección de seguimiento de la página de detalles para cada almacén de claves externo. El siguiente tema detalla

cómo usar los gráficos para identificar y solucionar problemas operativos y de configuración que afectan su almacén de claves externo. Te recomendamos que utilices las CloudWatch métricas para configurar alarmas que te notifiquen cuando tu almacén de claves externo no funcione como esperabas. Para obtener más información, consulta [Monitoring with Amazon CloudWatch](#).

Temas

- [Visualización de los gráficos](#)
- [Interpretación de los gráficos](#)

Visualización de los gráficos

Puede ver los gráficos en diferentes niveles de detalle. De forma predeterminada, cada gráfico utiliza un intervalo de tiempo de tres horas y un [periodo](#) de agregación de cinco minutos. Puede ajustar la vista del gráfico dentro de la consola, pero sus cambios volverán a la configuración predeterminada cuando se cierre la página de detalles del almacén de claves externo o se actualice el navegador. Para obtener ayuda con la CloudWatch terminología de Amazon, consulta [CloudWatch Conceptos de Amazon](#).

Visualización de detalles de puntos de datos

Los datos de cada gráfico se recopilan mediante [métricas de AWS KMS](#). Para ver más información sobre un punto de datos específico, coloque el ratón sobre el punto de datos del gráfico lineal. Esto mostrará una ventana emergente con más información sobre la métrica de la que se derivó el gráfico. Cada elemento de la lista muestra el valor de [dimensión](#) registrado en ese punto de datos. La ventana emergente muestra un valor nulo (–) si no hay datos métricos disponibles para el valor de la dimensión en ese punto de datos. Algunos gráficos registran varias dimensiones y valores para un único punto de datos. Otros gráficos, como el [gráfico de fiabilidad](#), utilizan los datos recopilados por la métrica para calcular un valor único. Cada elemento de la lista está asociado con un color de gráfico de líneas diferente.

Modificación del intervalo de tiempo

Para modificar el [intervalo de tiempo](#), seleccione uno de los intervalos de tiempo predefinidos en la esquina superior derecha de la sección de monitoreo. Los intervalos de tiempo predefinidos abarcan de 1 hora a 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 sem.). Esto ajusta el intervalo de tiempo de todos los gráficos. Si quieres ver un gráfico específico en un intervalo de tiempo diferente, o si quieres establecer un intervalo de tiempo personalizado, amplía el gráfico o visualízalo en la CloudWatch consola de Amazon.

Acercar un gráfico

Puede usar la [función de zoom del minimapa](#) para enfocarse en secciones de gráficos de líneas y porciones apiladas de los gráficos sin cambiar entre vistas ampliadas y alejadas. Por ejemplo, puede usar la función de zoom del minimapa para enfocarse en un pico en un gráfico, de modo que pueda comparar el pico con otros gráficos en la sección de monitoreo de la misma línea de tiempo.

1. Elija y arrastre el área del gráfico en la que desea centrarse y, a continuación, suéltela.
2. Para restablecer el tamaño del gráfico, elija el icono Reset zoom (Restablecer el zoom) que parece una lupa con un símbolo menos (-) en su interior.

Ampliación de un gráfico

Para ampliar un gráfico, seleccione el icono de menú situado en la esquina superior derecha de un gráfico individual y, a continuación, seleccione Enlarge (Ampliar). También puede seleccionar el icono de ampliación que aparece junto al icono del menú al pasar el ratón sobre un gráfico.

Al ampliar un gráfico, puede modificar aún más la vista de un gráfico especificando un periodo diferente, un intervalo de tiempo personalizado o un intervalo de actualización. Estos cambios volverán a la configuración predeterminada cuando cierre la vista ampliada.

Modificación del periodo

1. Seleccione el menú Period options (Opciones de periodo). De forma predeterminada, este menú muestra el valor: 5 minutos.
2. Elija un periodo, los periodos predefinidos van de 1 segundo a 30 días.

Por ejemplo, puede elegir una vista de un minuto, que puede ser útil a la hora de solucionar problemas. O bien puede elegir una vista menos detallada de una hora. Esto puede ser útil cuando desee ver un intervalo de tiempo mayor (por ejemplo, tres días) para poder identificar las tendencias a lo largo del tiempo. Para obtener más información, consulta [Períodos](#) en la Guía del CloudWatch usuario de Amazon.

Modificación del intervalo de tiempo o la zona horaria

1. Seleccione uno de los intervalos de tiempo predefinidos, que van desde 1 hora hasta 1 semana (1 h, 3 h, 12 h, 1 día, 3 días o 1 semana). También puede elegir Custom (Personalizado) para establecer su propio intervalo de tiempo.
2. Elija Custom (Personalizado).

- a. Intervalo de tiempo: seleccione la pestaña Absolute (Absoluto) en la esquina superior izquierda del cuadro. Utilice el selector de calendario o los cuadros de campos de texto para especificar el intervalo de tiempo.
 - b. Zona horaria: elija el menú desplegable ubicado en la esquina superior derecha del cuadro. Puede cambiar la zona horaria a UTC (UTC) o Local time zone (Zona horaria local).
3. Después de especificar un intervalo de tiempo, seleccione Apply (Aplicar).

Modifique la frecuencia con la que se actualizan los datos de su gráfico

1. Elija el menú Refresh options (Opciones de actualización) en la esquina superior derecha.
2. Elija un intervalo de actualización [Off (desactivado), 10 seconds (10 segundos), 1 minute (1 minuto), 2 minutes (2 minutos), 5 minutes (5 minutos) o 15 minutes (15 minutos)].

Ver gráficos en la CloudWatch consola de Amazon

Los gráficos de la sección de monitorización se derivan de métricas predefinidas que se AWS KMS publican en Amazon CloudWatch. Puede abrirlas en la CloudWatch consola y guardarlos en los CloudWatch paneles. Si tiene varios almacenes de claves externos, puede abrir sus gráficos respectivos CloudWatch y guardarlos en un único panel para comparar su estado y uso.

Añadir al CloudWatch panel

Selecciona Añadir al panel en la esquina superior derecha para añadir todos los gráficos a un CloudWatch panel de Amazon. Puede seleccionar un tablero existente o crear uno nuevo. Para obtener información sobre el uso de este panel para crear vistas personalizadas de los gráficos y las alarmas, consulte [Uso de los CloudWatch paneles de Amazon](#) en la Guía del CloudWatch usuario de Amazon.

Ver en métricas CloudWatch

Selecciona el icono de menú en la esquina superior derecha de un gráfico individual y selecciona Ver en métricas para ver este gráfico en la CloudWatch consola de Amazon. Desde la CloudWatch consola, puedes añadir este único gráfico a un panel y modificar los rangos de tiempo, los períodos y los intervalos de actualización. Para obtener más información, consulta Cómo [graficar métricas](#) en la Guía del CloudWatch usuario de Amazon.

Interpretación de los gráficos

AWS KMS proporciona varios gráficos para supervisar el estado del almacén de claves externo de la AWS KMS consola. Estos gráficos se configuran automáticamente y se derivan de [las métricas de AWS KMS](#).

Los datos del gráfico se recopilan como parte de las llamadas que realiza a su almacén de claves externo y claves externas. Es posible que vea datos rellenando los gráficos durante un intervalo de tiempo en el que no ha realizado ninguna llamada. Estos datos provienen de las `GetHealthStatus` llamadas periódicas que AWS KMS realiza en su nombre para comprobar el estado de su almacén de claves, proxy y administrador de claves externo. Si sus gráficos muestran el mensaje `No data available` (No hay datos disponibles), entonces no hubo llamadas registradas durante ese intervalo de tiempo o su almacén de claves externo está en un estado [DISCONNECTED](#). Es posible que pueda identificar la hora en que se desconectó su almacén de claves externo [ajustando su vista](#) a un intervalo de tiempo más amplio.

Temas

- [Número total de solicitudes](#)
- [Fiabilidad](#)
- [Latencia](#)
- [Las 5 excepciones principales](#)
- [Días para el vencimiento del certificado](#)

Número total de solicitudes

El número total de AWS KMS solicitudes que se reciben para un almacén de claves externo específico durante un intervalo de tiempo determinado. Utilice este gráfico para determinar si corre el riesgo de sufrir una limitación.

AWS KMS recomienda que su administrador de claves externo pueda gestionar hasta 1800 solicitudes de operaciones criptográficas por segundo. Si se acerca a las 540 000 llamadas en un periodo de cinco minutos, corre el riesgo de sufrir una limitación.

Puede controlar el número de solicitudes de operaciones criptográficas en las claves de KMS en su almacén de claves externo, que se acelera AWS KMS con la métrica. [ExternalKeyStoreThrottle](#)

Si recibe errores `KMSInvalidStateException` muy frecuentes con un mensaje que explica que la solicitud fue rechazada “debido a una tasa de solicitudes muy alta”, podría indicar que su

administrador de claves externo o el proxy del almacén de claves externo no pueden seguir el ritmo de la tasa de solicitudes actual. Si es posible, reduzca el porcentaje de solicitudes. También podría considerar solicitar una disminución en el valor de la cuota de solicitudes del almacén de claves personalizado. Reducir este valor de cuota puede aumentar la limitación, pero indica que se AWS KMS está rechazando el exceso de solicitudes rápidamente antes de enviarlas al proxy del almacén de claves externo o al administrador de claves externo. Para solicitar una reducción de la cuota, visite [el Centro AWS Support](#) y cree un caso.

El gráfico del total de solicitudes se obtiene de la métrica [XksProxyErrors](#), que recopila datos sobre las respuestas correctas y fallidas que AWS KMS recibe del proxy del almacén de claves externo. Al [ver un punto de datos específico](#), la ventana emergente muestra el valor de la CustomKeyId dimensión junto con el número total de AWS KMS solicitudes registradas en ese punto de datos. El CustomKeyId siempre será el mismo.

Fiabilidad

El porcentaje de AWS KMS solicitudes para las que el proxy del almacén de claves externo arrojó una respuesta correcta o un error que no se pudo volver a intentar. Utilice este gráfico para evaluar el estado operativo de su proxy del almacén de claves externo.

Cuando el gráfico muestra un valor inferior al 100 %, indica los casos en los que el proxy no respondió o respondió con un error reintentable. Esto puede indicar problemas con la red, lentitud del proxy del almacén de claves externo o del administrador de claves externo, o errores de implementación.

Si la solicitud incluye una credencial incorrecta y su proxy responde con una `AuthenticationFailedException`, el gráfico seguirá indicando una fiabilidad del 100 % porque el proxy identificó un valor incorrecto en la [solicitud de la API del proxy del almacén de claves externo](#) y, por lo tanto, es de esperar que se produzca un error. Si el porcentaje del gráfico de fiabilidad es del 100 %, entonces el proxy del almacén de claves externo responde según lo esperado. Si el gráfico muestra un valor inferior al 100 %, el proxy respondió con un error reintentable o se agotó el tiempo de espera. Por ejemplo, si el proxy responde con una respuesta de `ThrottlingException` debido a una tasa de solicitudes muy alta, mostrará un porcentaje de fiabilidad más bajo porque el proxy no pudo identificar un problema específico en la solicitud que provocó el error. Esto se debe a que los errores reintentables son probablemente problemas transitorios que se pueden resolver reintentando la solicitud.

Las siguientes respuestas de error reducirán el porcentaje de fiabilidad. Puede usar el gráfico [Las 5 excepciones principales](#) y la métrica [XksProxyErrors](#) para controlar con más detalle la frecuencia con la que su proxy devuelve cada error reintentable.

- `InternalException`
- `DependencyTimeoutException`
- `ThrottlingException`
- `XksProxyUnreachableException`

El gráfico de fiabilidad se deriva de la [XksProxyErrors](#) métrica, que recopila datos sobre las respuestas correctas y no satisfactorias que se AWS KMS reciben del proxy del almacén de claves externo. El porcentaje de fiabilidad solo disminuirá si la respuesta tiene un valor de `ErrorType` de `Retryable`. Al [ver un punto de datos específico](#), la ventana emergente muestra el valor de la `CustomKeyStoreId` dimensión junto con el porcentaje de confiabilidad de AWS KMS las solicitudes registradas en ese punto de datos. El `CustomKeyStoreId` siempre será el mismo.

Te recomendamos usar la [XksProxyErrors](#) métrica para crear una CloudWatch alarma que te avise de posibles problemas de red y te avise cuando se registren más de cinco errores reintentables en un período de un minuto. Para obtener más información, consulte [Creación de una alarma para los errores reintentables](#).

Latencia

El número de milisegundos que tarda un proxy de almacén de claves externo en responder a una solicitud. AWS KMS Utilice este gráfico para evaluar el rendimiento de su proxy del almacén de claves externo y de su administrador de claves externo.

AWS KMS espera que el proxy del almacén de claves externo responda a cada solicitud en un plazo de 250 milisegundos. En caso de que se agote el tiempo de espera de la red, AWS KMS volverá a intentar la solicitud una vez. Si el proxy falla por segunda vez, la latencia registrada es el límite de tiempo de espera combinado para ambos intentos de solicitud y el gráfico mostrará aproximadamente 500 milisegundos. En todos los demás casos en los que el proxy no responde dentro del límite de tiempo de espera de 250 milisegundos, la latencia registrada es de 250 milisegundos. Si el proxy se agota con frecuencia en las operaciones de cifrado y descifrado, consulte a su administrador de proxy externo. Para obtener ayuda con la solución de problemas de latencia, consulte [Errores de latencia y tiempo de espera](#).

Las respuestas lentas también pueden indicar que el administrador de claves externo no puede gestionar el tráfico de solicitudes actual. AWS KMS recomienda que su administrador de claves externo pueda gestionar hasta 1800 solicitudes de operaciones criptográficas por segundo. Si su administrador de claves externo no puede gestionar la tasa de 1800 solicitudes por segundo, considere solicitar una reducción de su [cuota de solicitudes de claves de KMS en un almacén de claves personalizado](#). Las solicitudes de operaciones criptográficas que utilizan las claves de KMS en su almacén de claves externo van a responder rápido a los errores con una [excepción de limitación](#), en lugar de ser procesadas y luego rechazadas por su proxy del almacén de claves externo o administrador de claves externo.

El gráfico de latencia se realiza a partir de la métrica [XksProxyLatency](#). Al consultar un [punto de datos específico](#), la ventana emergente muestra los valores de dimensión `KmsOperation` y `XksOperation` correspondientes junto con la latencia promedio registrada para las operaciones en ese punto de datos. Los elementos de la lista se ordenan de mayor a menor latencia.

Te recomendamos usar la [XksProxyLatency](#) métrica para crear una CloudWatch alarma que te notifique cuando la latencia se acerque al límite de tiempo de espera. Para obtener más información, consulte [Creación de una alarma para el tiempo de espera de respuesta](#).

Las 5 excepciones principales

Las cinco excepciones principales para operaciones criptográficas y de administración fallidas durante un intervalo de tiempo determinado. Utilice este gráfico para realizar un seguimiento de los errores más frecuentes, de modo que pueda priorizar sus esfuerzos de ingeniería.

Este recuento incluye las excepciones que se AWS KMS reciben del proxy del almacén de claves externo y las `XksProxyUnreachableException` que se AWS KMS devuelven internamente cuando no puede establecer comunicación con el proxy del almacén de claves externo.

Las altas tasas de errores reintentables pueden indicar errores de red, mientras que las altas tasas de errores no reintentables pueden indicar un problema con la configuración del almacén de claves externo. Por ejemplo, un pico de entrada `AuthenticationFailedExceptions` indica una discrepancia entre las credenciales de autenticación configuradas en AWS KMS el proxy del almacén de claves externo. Para ver la configuración de su almacén de claves externo, consulte [Visualización de almacenes de claves externos](#). Para editar la configuración de su almacén de claves externo, consulte [Edición de las propiedades del almacén de claves externo](#).

Las excepciones que se AWS KMS reciben del proxy del almacén de claves externo son diferentes de las que se muestran cuando se produce AWS KMS un error en una operación. AWS KMS Las operaciones criptográficas muestran un valor `KMSInvalidStateException` para todos los errores

relacionados con la configuración externa o el estado de conexión del almacén de claves externo. Para identificar el problema, utilice el texto del mensaje de error adjunto.

En la siguiente tabla se muestran las excepciones que pueden aparecer en el gráfico de las cinco excepciones principales y las excepciones correspondientes que aparecen AWS KMS en la pantalla.

Tipo de error	Excepción mostrada en el gráfico	Excepción que se te AWS KMS devolvió
No reintentable	AccessDeniedException Para obtener ayuda sobre la resolución de problemas , consulte Problemas de autorización de proxy .	CustomKeyStoreInvalidStateException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.
No reintentable	AuthenticationFailedException Para obtener ayuda sobre la resolución de problemas , consulte Errores en las credenciales de autenticación .	XksProxyIncorrectAuthenticationCredentialException en respuesta a operaciones <code>CreateCustomKeyStore</code> y <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.
Reintentable	DependencyTimeoutException	XksProxyUriUnreachableException en respuesta a operaciones <code>CreateCustomKeyStore</code>

Tipo de error	Excepción mostrada en el gráfico	Excepción que se te AWS KMS devolvió
	Para obtener ayuda sobre la resolución de problemas, consulte Errores de latencia y tiempo de espera.	y <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.
Reintentable	InternalException El proxy del almacén de claves externo rechazó la solicitud porque no puede comunicarse con el administrador de claves externo. Compruebe que la configuración del proxy del almacén de claves externo sea correcta y que el administrador de claves externo esté disponible.	XksProxyInvalidResponseException en respuesta a operaciones <code>CreateCustomKeyStore</code> y <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.
No reintentable	InvalidCiphertextException Para obtener ayuda sobre la resolución de problemas, consulte Errores de descifrado.	KMSInvalidStateException en respuesta a operaciones criptográficas.

Tipo de error	Excepción mostrada en el gráfico	Excepción que se te AWS KMS devolvió
No reintentable	InvalidKeyUsageException Para obtener ayuda sobre la resolución de problemas, consulte Errores de operación criptográfica para la clave externa .	XksKeyInvalidConfigurationException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.
No reintentable	InvalidStateException Para obtener ayuda sobre la resolución de problemas, consulte Errores de operación criptográfica para la clave externa .	XksKeyInvalidConfigurationException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.
No reintentable	InvalidUriPathException Para obtener ayuda sobre la resolución de problemas, consulte Errores de configuración general .	XksProxyInvalidConfigurationException en respuesta a operaciones <code>CreateCustomKeyStore</code> y <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.

Tipo de error	Excepción mostrada en el gráfico	Excepción que se te AWS KMS devolvió
No reintentable	KeyNotFoundException Para obtener ayuda sobre la resolución de problemas , consulte Errores de clave externa .	XksKeyNotFoundException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.
Reintentable	ThrottlingException El proxy del almacén de claves externo rechazó la solicitud debido a una tasa de solicitudes muy alta. Reduzca la frecuencia de las llamadas con las claves de KMS en este almacén de claves externo.	XksProxyUriUnreachableException en respuesta a operaciones <code>CreateCustomKeyStore</code> y <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.
No reintentable	UnsupportedOperationException Para obtener ayuda sobre la resolución de problemas, consulte Errores de operación criptográfica para la clave externa .	XksKeyInvalidResponseException en respuesta a operaciones <code>CreateKey</code> . KMSInvalidStateException en respuesta a operaciones criptográficas.

Tipo de error	Excepción mostrada en el gráfico	Excepción que se te AWS KMS devolvió
No reintentable	ValidationException Para obtener ayuda sobre la resolución de problemas , consulte Problemas con el proxy.	XksProxyInvalidResponseException en respuesta a operaciones CreateCustomKeyStore y UpdateCustomKeyStore . CustomKeyStoreInvalidStateException en respuesta a operaciones CreateKey . KMSInvalidStateException en respuesta a operaciones criptográficas.
Reintentable	XksProxyUnreachableException Si aparece este error varias veces, compruebe que el proxy del almacén de claves externo esté activo y conectado a la red, y que su ruta URI y su nombre de servicio de VPC o URI de punto de conexión sean correctos en su almacén de claves externo.	XksProxyUriUnreachableException en respuesta a operaciones CreateCustomKeyStore y UpdateCustomKeyStore . CustomKeyStoreInvalidStateException en respuesta a operaciones CreateKey . KMSInvalidStateException en respuesta a operaciones criptográficas.

El gráfico de las 5 excepciones principales se deriva de la métrica [XksProxyErrors](#). Al ver un [punto de datos específico](#), la ventana emergente muestra el valor de la dimensión ExceptionName junto

con la cantidad de veces que se registró la excepción en ese punto de datos. Los cinco elementos de la lista están ordenados de la excepción más frecuente a la menos frecuente.

Recomendamos utilizar la [XksProxyErrors](#) métrica para crear una CloudWatch alarma que le notifique los posibles problemas de configuración y le avise cuando se registren más de cinco errores que no se pueden volver a intentar en un período de un minuto. Para obtener más información, consulte [Creación de una alarma para los errores no reintentables](#).

Días para el vencimiento del certificado

El número de días que faltan para que venza el certificado TLS del punto de conexión del proxy del almacén de claves externo (`XksProxyUriEndpoint`). Utilice este gráfico para monitorear el próximo vencimiento de su certificado TLS.

Cuando el certificado caduque, AWS KMS no podrá comunicarse con el proxy del almacén de claves externo. No se podrá acceder a todos los datos protegidos por las claves de KMS en su almacén de claves externo hasta que renueve el certificado.

El gráfico de días para el vencimiento del certificado se deriva de la métrica [XksProxyCertificateDaysToExpire](#). Recomendamos encarecidamente utilizar esta métrica para crear una CloudWatch alarma que le notifique la próxima caducidad. El vencimiento del certificado puede impedirle acceder a los recursos cifrados. Configure la alarma para que su organización tenga tiempo de renovar el certificado antes de que venza. Para obtener más información, consulte [Creación de una alarma para el vencimiento del certificado](#).

Conexión y desconexión de almacenes de claves externos

Los nuevos almacenes de claves externos no están conectados. Para crear y usar AWS KMS keys en su almacén de claves externo, debe conectar su almacén de claves externo a su [proxy de almacén de claves externo](#). Puede conectar y desconectar su almacén de claves externo en cualquier momento y [ver su estado de conexión](#).

Mientras el almacén de claves externo esté desconectado, AWS KMS no podrá comunicarse con el proxy del almacén de claves externo. Como resultado, puede ver y administrar su almacén de claves externo y sus claves de KMS existentes. Sin embargo, no puede crear claves de KMS en su almacén de claves externo ni utilizar sus claves de KMS en operaciones criptográficas. Puede que tenga que desconectar el almacén de claves externo en algún momento, por ejemplo, al editar sus propiedades, por lo que debería planificar ante eventualidades. La desconexión del almacén de claves podría interrumpir el funcionamiento de AWS los servicios que utilizan sus claves KMS.

No es obligatorio conectar el almacén de claves externo. Puede dejar un almacén de claves externo desconectado de forma indefinida y conectarlo únicamente cuando tenga que usarlo. Sin embargo, le recomendamos que compruebe la conexión de forma periódica para verificar que la configuración es correcta y que se puede conectar.

Al desconectar un almacén de claves personalizado, las claves de KMS del almacén de claves quedan inutilizables de inmediato (sujeto a posible coherencia). Sin embargo, los recursos cifrados con [claves de datos](#) protegidas por la clave de KMS no se ven afectados hasta que se vuelva a utilizar la clave de KMS, por ejemplo, para descifrar la clave de datos. Este problema afecta a los Servicios de AWS, muchos de los cuales utilizan claves de datos para proteger sus recursos. Para obtener más información, consulte [Cómo afectan las claves de KMS obsoletas a las claves de datos](#).

Note

Los almacenes de claves externos tienen un estado DISCONNECTED solo cuando el almacén de claves nunca se ha conectado o lo desconecta explícitamente. Un estado CONNECTED no indica que el almacén de claves externo o sus componentes de soporte estén funcionando de manera eficiente. Para obtener información sobre el rendimiento de los componentes del almacén de claves externo, consulte los gráficos de la sección Monitoreo de la página de detalles de cada almacén de claves externo. Para obtener más información, consulte [Monitoreo de almacenes de claves externos](#).

El administrador de claves externo puede proporcionar métodos adicionales para detener y reiniciar la comunicación entre el almacén de claves AWS KMS externo y el proxy del almacén de claves externo, o entre el proxy del almacén de claves externo y el administrador de claves externo. Para obtener más detalles, consulte la documentación del administrador de claves externo.

Temas

- [Estado de la conexión](#)
- [Conexión de un almacén de claves externo](#)
- [Desconexión de un almacén de claves externo](#)

Estado de la conexión

La conexión y la desconexión cambian el estado de conexión del almacén de claves personalizado. Los valores del estado de conexión son los mismos para los almacenes de AWS CloudHSM claves y los almacenes de claves externos.

Para ver el estado de conexión del almacén de claves personalizado, utilice la [DescribeCustomKeyStores](#) operación o la AWS KMS consola. El Connection state (Estado de la conexión) aparece en cada tabla de almacén de claves personalizado, en la sección General configuration (Configuración general), y en la pestaña Cryptographic configuration (Configuración criptográfica) de las claves de KMS de un almacén de claves personalizado. Para más detalles, consulte [Ver un almacén de AWS CloudHSM claves](#) y [Visualización de almacenes de claves externos](#).

Un almacén de claves personalizado puede tener uno de los siguientes estados de conexión:

- **CONNECTED:** el almacén de claves personalizado está conectado a su almacén de claves de respaldo. Puede crear y usar claves de KMS en el almacén de claves personalizado.

El almacén de claves de respaldo de un almacén de AWS CloudHSM claves es su AWS CloudHSM clúster asociado. El almacén de claves de respaldo de un almacén de claves externo es el proxy del almacén de claves externo y el administrador de claves externo que admite.

El estado CONNECTED (CONECTADO) significa que la conexión se ha realizado correctamente y que el almacén de claves personalizado no se ha desconectado intencionadamente. No indica que la conexión funcione correctamente. Para obtener información sobre el estado del AWS CloudHSM clúster asociado a su almacén de AWS CloudHSM claves, consulte [Obtener CloudWatch métricas AWS CloudHSM](#) en la Guía del AWS CloudHSM usuario. Para obtener información sobre el estado y el funcionamiento del almacén de claves externo, consulte los gráficos de la sección Monitoreo de la página de detalles de cada almacén de claves externo. Para obtener más información, consulte [Monitoreo de almacenes de claves externos](#).

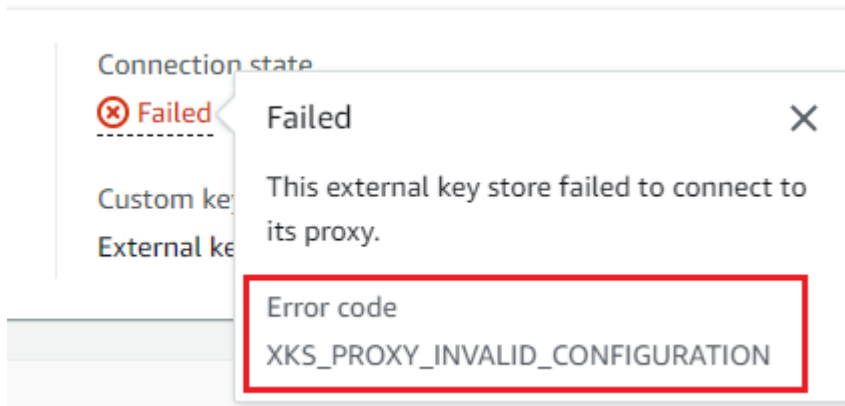
- **CONNECTING:** el proceso de conectar un almacén de claves personalizado está en curso. Este es un estado transitorio.
- **DISCONNECTED:** El almacén de claves personalizado nunca se conectó a su soporte o se desconectó intencionadamente mediante la AWS KMS consola o la [DisconnectCustomKeyStore](#) operación.
- **DISCONNECTING:** el proceso de desconexión de un almacén de claves personalizado está en curso. Este es un estado transitorio.

- FAILED: se produjo un error al intentar conectar el almacén de claves personalizado. Lo que `ConnectionErrorCode` aparece en la [DescribeCustomKeyStores](#) respuesta indica el problema.

Para conectar un almacén de claves personalizado, su estado de conexión debe ser DISCONNECTED. Si el estado de la conexión es FAILED, utilice el `ConnectionErrorCode` para identificar y resolver el problema. Luego, desconecte el almacén de claves personalizado antes de intentar conectarse de nuevo. Si desea ayuda con las conexiones que dan error, consulte [Errores de conexión del almacén de claves externo](#). Para obtener ayuda sobre cómo responder a un código de error de conexión, consulte [Códigos de error de conexión para almacenes de claves externos](#).

Para ver el código de error de conexión:

- En la [DescribeCustomKeyStores](#) respuesta, vea el valor del `ConnectionErrorCode` elemento. Este elemento aparece en la respuesta `DescribeCustomKeyStores` solo cuando el `ConnectionState` es FAILED.
- Para ver el código de error de conexión en la AWS KMS consola, vaya a la página de detalles del almacén de claves externo y coloque el cursor sobre el valor de error.



Conexión de un almacén de claves externo

Cuando el almacén de claves externo esté conectado, puede [crear las claves de KMS en él](#) y usar las claves de KMS existentes en [operaciones criptográficas](#).

El proceso que conecta un almacén de claves externo a su proxy del almacén de claves externo difiere en función de la conectividad del almacén de claves externo.

- Cuando conectas un almacén de claves externo con [conectividad de punto final público](#), AWS KMS envía una [GetHealthStatus solicitud](#) al proxy del almacén de claves externo para validar

el [extremo URI del proxy](#), la [ruta URI del proxy](#) y la [credencial de autenticación del proxy](#). Una respuesta correcta del proxy confirma que el [punto de conexión de la URI del proxy](#) y la [ruta de la URI del proxy](#) son precisos y accesibles, y que el proxy autenticó la solicitud firmada con la [credencial de autenticación del proxy](#) para el almacén de claves externo.

- Al conectar un almacén de claves externo con [conectividad del servicio de punto final de VPC](#) a su proxy de almacén de claves externo, AWS KMS hace lo siguiente:
 - Confirma que se ha [verificado](#) el dominio del nombre DNS privado especificado en el [punto de conexión de URI del proxy](#).
 - Crea un punto final de interfaz desde una AWS KMS VPC hasta su servicio de punto final de VPC.
 - Crea una zona alojada privada para el nombre DNS privado especificado en el punto de conexión de URI del proxy
 - Envía una [GetHealthStatussolicitud](#) al proxy del almacén de claves externo. Una respuesta correcta del proxy confirma que el [punto de conexión de la URI del proxy](#) y la [ruta de la URI del proxy](#) son precisos y accesibles, y que el proxy autenticó la solicitud firmada con la [credencial de autenticación del proxy](#) para el almacén de claves externo.

La operación de conexión inicia el proceso de conexión del almacén de claves personalizado, pero conectar un almacén de claves externo a su proxy externo tarda aproximadamente cinco minutos. Una respuesta correcta de la operación de conexión no indica que el almacén de claves externo esté conectado. Para confirmar que la conexión se ha realizado correctamente, utilice la AWS KMS consola o la [DescribeCustomKeyStores](#) operación para ver el [estado de la conexión](#) del almacén de claves externo.

Cuando el estado de la conexión es `FAILED`, se muestra un código de error de conexión en la AWS KMS consola y se añade a la `DescribeCustomKeyStore` respuesta. Para obtener ayuda para interpretar los códigos de error de conexión, consulte [Códigos de error de conexión para almacenes de claves externos](#).

Conexión y reconexión a su almacén de claves externo

Puede conectar o volver a conectar el almacén de claves externo en la AWS KMS consola o mediante esta [ConnectCustomKeyStore](#) operación.

Uso de la consola AWS KMS

Puede utilizar la AWS KMS consola para conectar un almacén de claves externo a su proxy de almacén de claves externo.

1. Inicia sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrela en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Custom key stores (Almacenes de claves personalizados), External key stores (Almacenes de claves externos).
4. Elija la fila del almacén de claves externo que desee conectar.

Si el [estado de conexión](#) del almacén de claves externo es FAILED (ERROR), deberá [desconectar el almacén de claves externo](#) antes de conectarlo.

5. En el menú Key store actions (Acciones del almacén de claves), seleccione Connect (Conectar).

El proceso de conexión suele tardar unos cinco minutos en completarse. Cuando se completa la operación, el [estado de la conexión](#) cambia a CONNECTED (CONECTADO).

Si el estado de conexión es Failed (Error), coloque el cursor sobre el estado de la conexión para ver el código de error de conexión, que explica la causa del error. Para obtener ayuda sobre cómo responder a un código de error de conexión, consulte [Códigos de error de conexión para almacenes de claves externos](#). Para conectar un almacén de claves externo con un estado de conexión Failed (Error), primero debe [desconectar el almacén de claves personalizado](#).

Uso de la API AWS KMS

Para conectar un almacén de claves externo desconectado, utilice la [ConnectCustomKeyStore](#) operación.

Antes de realizar la conexión, el [estado de conexión](#) del almacén de claves externo debe ser DISCONNECTED. Si el estado de conexión actual es FAILED, [desconecte el almacén de claves externo](#) y conéctelo de nuevo.

El proceso de conexión tarda hasta cinco minutos en completarse. A menos que el error sea rápido, la operación ConnectCustomKeyStore devuelve una respuesta HTTP 200 y un objeto JSON sin propiedades. Sin embargo, esta respuesta inicial no indica que la conexión se haya realizado

correctamente. Para determinar si el almacén de claves externo está conectado, consulte el estado de la conexión en la [DescribeCustomKeyStores](#) respuesta.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

Para identificar el almacén de claves externo, use el ID del almacén de claves personalizado. Puede encontrar el ID en la página de almacenes de claves personalizados de la consola o mediante la [DescribeCustomKeyStores](#) operación. Antes de ejecutar este ejemplo, reemplace el ID de ejemplo por uno válido.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

En cambio, la operación `ConnectCustomKeyStore` no devuelve el `ConnectionState` en su respuesta. Para comprobar que el almacén de claves externo está conectado, utilice la [DescribeCustomKeyStores](#) operación. De forma predeterminada, esta operación devuelve los almacenes de claves personalizados de su cuenta y región. Pero puede usar el parámetro `CustomKeyStoreId` o `CustomKeyStoreName` (pero no ambos) para limitar la respuesta a almacenes de claves personalizados determinados. El valor de `ConnectionState` `CONNECTED` indica que el almacén de claves externo está conectado a su proxy del almacén de claves externo.

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

Si el valor de `ConnectionState` en la respuesta `DescribeCustomKeyStores` es `FAILED`, el elemento `ConnectionErrorCode` indica el motivo del error.

En el siguiente ejemplo, el `XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND` valor de `ConnectionErrorCode` indica que no AWS KMS puede encontrar el servicio de punto final de la VPC que utiliza para comunicarse con el proxy del almacén de claves externo. Compruebe que `XksProxyVpcEndpointServiceName` es correcto, que el principal del AWS KMS servicio es un principal permitido en el servicio de puntos de enlace de Amazon VPC y que el servicio de puntos de enlace de VPC no requiere la aceptación de las solicitudes de conexión. Para obtener ayuda sobre cómo responder a un código de error de conexión, consulte [Códigos de error de conexión para almacenes de claves externos](#).

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

Desconexión de un almacén de claves externo

Al desconectar un almacén de claves externo con [conectividad a un servicio de punto de conexión de VPC](#) de su proxy del almacén de claves externo, AWS KMS elimina el punto de conexión de su interfaz al servicio del punto de conexión de VPC y elimina la infraestructura de red que creó para respaldar la conexión. No se requiere ningún proceso equivalente para los almacenes de claves externos con puntos de conexión públicos. Esta acción no afecta al servicio del punto de conexión

de VPC ni a ninguno de sus componentes de soporte, ni al proxy del almacén de claves externo ni a ningún componente externo.

Mientras el almacén de claves externo esté desconectado, AWS KMS no envía ninguna solicitud al proxy del almacén de claves externo. El estado de conexión del almacén de claves externo es `DISCONNECTED`. Las claves de KMS del almacén de claves externo desconectado se encuentran en el [estado de clave UNAVAILABLE](#) (a menos que estén [pendientes de eliminación](#)), lo que significa que no se pueden utilizar en operaciones criptográficas. Sin embargo, aún puede ver y administrar su almacén de claves externo y sus claves de KMS existentes.

El estado desconectado está diseñado para ser temporal y reversible. Puede volver a conectar el almacén de claves externo en cualquier momento. Por lo general, no es necesaria ninguna reconfiguración. Sin embargo, si alguna de las propiedades del proxy del almacén de claves externo asociado cambió mientras estaba desconectado, como la rotación de su [credencial de autenticación del proxy](#), debe [editar la configuración del almacén de claves externo](#) antes de volver a conectarse.

 Note

Mientras un almacén de claves personalizado esté desconectado, todos los intentos de crear claves KMS en el almacén de claves personalizado o de usar claves KMS existentes en operaciones criptográficas fallarán. Esta acción puede impedir que los usuarios almacenen y accedan a datos confidenciales.

Para realizar una mejor estimación del efecto de desconectar el almacén de claves externo, identifique las claves de KMS en el almacén de claves externo y [determine su uso en el pasado](#).

Puede desconectar el almacén de claves externo por los motivos siguientes:

- Para editar sus propiedades. Puede editar el nombre del almacén de claves personalizado, la ruta URI del proxy y la credencial de autenticación del proxy mientras el almacén de claves externo esté conectado. Sin embargo, para editar el tipo de conectividad del proxy, el punto de conexión URI del proxy o el nombre del servicio de punto de conexión de VPC, primero debe desconectar el almacén de claves externo. Para obtener más información, consulte [Edición de las propiedades del almacén de claves externo](#).
- Para detener todas las comunicaciones entre AWS KMS y el proxy del almacén de claves externo. También puede detener la comunicación entre AWS KMS y su proxy deshabilitando su punto final o el servicio de punto final de VPC. Además, el proxy del almacén de claves externo o el software

de administración de claves pueden proporcionar mecanismos adicionales para AWS KMS impedir la comunicación con el proxy o impedir que el proxy acceda a su administrador de claves externo.

- Para deshabilitar todas las claves de KMS en el almacén de claves externo. Puede [deshabilitar y volver a activar las claves KMS](#) en un almacén de claves externo mediante la AWS KMS consola o la [DisableKey](#) operación. Estas operaciones se completan rápidamente (sujeto a eventual consistencia), pero actúan en una clave de KMS cada vez. La desconexión del almacén de claves externo cambia el estado de clave de todas las claves de KMS en el almacén de claves externo a `Unavailable`, lo que evita que se utilicen en operaciones criptográficas.
- Para reparar un error en la conexión. Si el intento de conexión al almacén de claves externo falla (el estado de la conexión del almacén de claves externo es `FAILED`), deberá desconectar el almacén de claves externo antes de intentar conectarlo de nuevo.

Desconexión de su almacén de claves externo

Puede desconectar el almacén de claves externo de la AWS KMS consola o mediante esta [DisconnectCustomKeyStore](#) operación.

Uso de la AWS KMS consola

Puede utilizar la AWS KMS consola para conectar un almacén de claves externo a su proxy de almacén de claves externo. Este proceso tarda alrededor de 5 minutos en completarse.

1. Inicia sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrela en <https://console.aws.amazon.com/kms>.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Custom key stores (Almacenes de claves personalizados), External key stores (Almacenes de claves externos).
4. Elija la fila del almacén de claves externo que desee desconectar.
5. En el menú Key store actions (Acciones del almacén de claves), seleccione Disconnect (Desconectar).

Cuando la operación finalizada, el estado de conexión cambia de `DISCONNECTING` (`DESCONECTANDO`) a `DISCONNECTED` (`DESCONECTADO`). Si la operación da error, aparecerá un mensaje de error donde se describe el problema y se explica cómo resolverlo. Si necesita más ayuda, consulte [Errores de conexión del almacén de claves externo](#).

Uso de la API AWS KMS

Para desconectar un almacén de claves externo conectado, utilice la [DisconnectCustomKeyStore](#) operación. Si la operación se realiza correctamente, AWS KMS devuelve una respuesta HTTP 200 y un objeto JSON sin propiedades. El proceso tarda hasta cinco minutos en completarse. Para encontrar el estado de la conexión del almacén de claves externo, utilice la [DescribeCustomKeyStores](#) operación.

En los ejemplos de esta sección, se utiliza la [AWS Command Line Interface \(AWS CLI\)](#), pero puede usar cualquier lenguaje de programación admitido.

En este ejemplo, se desconecta un almacén de claves externo con la conectividad del servicio del punto de conexión de VPC. Antes de ejecutar este ejemplo, reemplace el ID del almacén de claves personalizado de ejemplo por uno válido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Para comprobar que el almacén de claves externo está desconectado, utilice la [DescribeCustomKeyStores](#) operación. De forma predeterminada, esta operación devuelve los almacenes de claves personalizados de su cuenta y región. Pero puede usar el parámetro CustomKeyId o CustomKeyName (pero no ambos) para limitar la respuesta a almacenes de claves personalizados determinados. El valor de ConnectionState DISCONNECTED indica que el almacén de claves externo ya no está conectado a su proxy del almacén de claves externo.

```
$ 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"
```

```
}  
}  
]  
}
```

Eliminación de un almacén de claves externo

Al eliminar un almacén de claves externo, AWS KMS elimina todos los metadatos sobre el almacén de claves externo AWS KMS, incluida la información sobre su proxy de almacén de claves externo. Esta operación no afecta al [proxy del almacén de claves externo](#), al [administrador de claves externo](#), a [las claves externas](#) ni a ningún AWS recurso que haya creado para respaldar el almacén de claves externo, como una Amazon VPC o un servicio de punto final de VPC.

Antes de eliminar un almacén de claves externo, debe [eliminar todas las claves de KMS del almacén de claves](#) y [desconectar el almacén de claves](#) de su proxy del almacén de claves externo. De lo contrario, fallarán los intentos de eliminar el almacén de claves.

Eliminar un almacén de claves externo es irreversible, pero puede crear un nuevo almacén de claves externo y asociarlo al mismo proxy del almacén de claves externo y al mismo administrador de claves externo. Sin embargo, no puede volver a crear las claves KMS de cifrado simétrico en el almacén de claves externo, aunque tenga acceso al mismo material de claves externas. AWS KMS incluye los metadatos en el texto cifrado simétrico exclusivo de cada clave KMS. Esta función de seguridad garantiza que solo la clave de KMS que cifró los datos pueda descifrarlos.

En lugar de eliminar el almacén de claves externo, considere la posibilidad de desconectarlo. Mientras un almacén de claves externo esté desconectado, puede administrar el almacén de claves externo y su almacenamiento, AWS KMS keys pero no puede crear ni usar claves KMS en el almacén de claves externo. Puede volver a conectar el almacén de claves externo en cualquier momento y reanudar el uso de sus claves de KMS para cifrar y descifrar datos. Un proxy del almacén de claves externo desconectado o sus claves de KMS no disponibles no incurren en ningún costo.

Puede eliminar el almacén de claves externo en la AWS KMS consola o mediante esta [DeleteCustomKeyStore](#) operación.

Uso de la AWS KMS consola

Puede utilizar la AWS KMS consola para eliminar un almacén de claves externo.

1. Inicia sesión en la consola AWS Key Management Service (AWS KMS) Consola de administración de AWS y ábrela en <https://console.aws.amazon.com/kms>.

2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Custom key stores (Almacenes de claves personalizados), External key stores (Almacenes de claves externos).
4. Busque la fila que representa el almacén de claves externo que quiere eliminar. Si el estado de conexión del almacén de claves externo no es DISCONNECTED (DESCONECTADO), debe [desconectar el almacén de claves externo](#) antes de eliminarlo.
5. En el menú Key store actions (Acciones del almacén de claves), seleccione Delete (Eliminar).

Cuando la operación finalice, aparecerá un mensaje de confirmación y el almacén de claves externo ya no aparecerá en la lista de almacenes de claves. Si la operación da error, aparecerá un mensaje de error donde se describe el problema y se explica cómo resolverlo. Si necesita más ayuda, consulte [Solución de problemas de almacenes de claves externos](#).

Uso de la API AWS KMS

Para eliminar un almacén de claves externo, utilice la [DeleteCustomKeyStore](#) operación. Si la operación se realiza correctamente, AWS KMS devuelve una respuesta HTTP 200 y un objeto JSON sin propiedades.

Para empezar, desconecte el almacén de claves externo. Antes de ejecutar este comando, reemplace el ID del almacén de claves personalizado de ejemplo por uno válido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Una vez desconectado el almacén de claves externo, puede utilizar la [DeleteCustomKeyStore](#) operación para eliminarlo.

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Para confirmar que se ha eliminado el almacén de claves externo, utilice la [DescribeCustomKeyStores](#) operación.

```
$ aws kms describe-custom-key-stores

{
  "CustomKeyStores": []
}
```

Si especifica un nombre o ID de almacén de claves personalizado que ya no existe, AWS KMS devuelve una `CustomKeyStoreNotFoundException` excepción.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
```

An error occurred (CustomKeyStoreNotFoundException) when calling the DescribeCustomKeyStore operation:

Solución de problemas de almacenes de claves externos

La solución de la mayoría de los problemas con los almacenes de claves externos se indica mediante el mensaje de error que AWS KMS aparece con cada excepción o mediante el [código de error de conexión](#) que aparece AWS KMS cuando se produce un error al intentar [conectar el almacén de claves externo](#) a su proxy de almacén de claves externo. Sin embargo, algunos problemas son un poco más complejos.

Al diagnosticar un problema con un almacén de claves externo, localice primero la causa. Esto reducirá la gama de soluciones y hará que la solución de problemas sea más eficiente.

- **AWS KMS** — El problema puede estar dentro AWS KMS, por ejemplo, un valor incorrecto en la [configuración del almacén de claves externo](#).
- **Externo**: el problema puede originarse fuera de AWS KMS, incluidos los problemas con la configuración o el funcionamiento del proxy del almacén de claves externo, el administrador de claves externo, las claves externas o el servicio de punto final de la VPC.
- **Redes**: puede tratarse de un problema de conectividad o de red, como un problema con el punto de conexión del proxy, el puerto, la pila de IP o el dominio o nombre DNS privado.

Note

Cuando las operaciones de administración en almacenes de claves externos fallan, se generan varias excepciones diferentes. Sin embargo, las operaciones AWS KMS criptográficas se `KMSInvalidStateException` repiten cuando se producen errores relacionados con la configuración externa o el estado de conexión del almacén de claves externo. Para identificar el problema, utilice el texto del mensaje de error adjunto.

La [ConnectCustomKeyStore](#) operación se realiza rápidamente antes de que se complete el proceso de conexión. Para determinar si el proceso de conexión se ha realizado correctamente, consulte el [estado de conexión](#) del almacén de claves externo. Si el proceso

de conexión falla, AWS KMS devuelve un [código de error de conexión](#) que explica la causa y sugiere una solución.

Temas

- [Herramientas de solución de problemas para almacenes de claves externos](#)
- [Errores de configuración](#)
- [Errores de conexión del almacén de claves externo](#)
- [Errores de latencia y tiempo de espera](#)
- [Errores en las credenciales de autenticación](#)
- [Errores de estados de las claves](#)
- [Errores de descifrado](#)
- [Errores de clave externa](#)
- [Problemas con el proxy](#)
- [Problemas de autorización de proxy](#)

Herramientas de solución de problemas para almacenes de claves externos

AWS KMS proporciona varias herramientas para ayudarle a identificar y resolver problemas con el almacén de claves externo y sus claves. Utilice estas herramientas junto con las herramientas proporcionadas con su proxy del almacén de claves externo y su administrador de claves externo.

Note

El proxy del almacén de claves externo y el administrador de claves externo pueden proporcionar métodos más sencillos para crear y mantener el almacén de claves externo y sus claves de KMS. Para obtener más detalles, consulte la documentación de sus herramientas externas.

AWS KMS excepciones y mensajes de error

AWS KMS proporciona un mensaje de error detallado sobre cualquier problema que encuentre. Puedes encontrar información adicional sobre AWS KMS las excepciones en la [referencia de la AWS Key Management Service API](#) y AWS SDKs. Incluso si utilizas la AWS KMS consola,

estas referencias pueden resultarte útiles. Por ejemplo, consulte la lista de [errores](#) de la operación `CreateCustomKeyStores`.

Para optimizar el rendimiento del proxy del almacén de claves externo, AWS KMS devuelve excepciones en función de la fiabilidad del proxy dentro de un período de agregación determinado de 5 minutos. En caso de que se obtenga una respuesta 500 Internal Server Error, 503 Service Unavailable o tiempo de espera de conexión agotado, un proxy de alta fiabilidad devuelve `KMSInternalException` funcionar y activa un reintento automático para garantizar que las solicitudes se realicen correctamente. Sin embargo, un proxy de baja fiabilidad devuelve `KMSInvalidStateException`. Para obtener más información, consulte [Monitoreo de un almacén de claves externo](#).

Si el problema surge en otro AWS servicio, por ejemplo, cuando utilizas una clave KMS en tu almacén de claves externo para proteger un recurso de otro AWS servicio, es posible que el AWS servicio proporcione información adicional que te ayude a identificar el problema. Si el AWS servicio no proporciona el mensaje, puedes ver el mensaje de error en los [CloudTrail registros](#) que registran el uso de tu clave KMS.

[CloudTrail logs](#)

Todas las operaciones de la AWS KMS API, incluidas las acciones de la AWS KMS consola, se registran en AWS CloudTrail los registros. AWS KMS registra una entrada de registro para las operaciones correctas y fallidas. Para las operaciones fallidas, la entrada del registro incluye el nombre de la excepción de AWS KMS (`errorCode`) y el mensaje de error (`errorMessage`). Puede utilizar esta información como ayuda para identificar y resolver el error. Para ver un ejemplo, consulta [Error de descifrado con una clave de KMS en un almacén de claves externo](#).

La entrada de registro también incluye el ID de la solicitud. Si la solicitud llegó a su proxy del almacén de claves externo, puede utilizar el ID de solicitud de la entrada de registro para buscar la solicitud correspondiente en sus registros de proxy, si su proxy los proporciona.

[CloudWatch métricas](#)

AWS KMS registra CloudWatch estadísticas detalladas de Amazon sobre el funcionamiento y el rendimiento de tu almacén de claves externo, como la latencia, la limitación, los errores de proxy, el estado del administrador de claves externo, el número de días que faltan para que caduque tu certificado TLS y la antigüedad declarada de tus credenciales de autenticación de proxy. Puede utilizar estas métricas para desarrollar modelos de datos para el funcionamiento de su almacén de claves externo y CloudWatch alarmas que le avisen de problemas inminentes antes de que se produzcan.

⚠ Important

AWS KMS recomienda crear CloudWatch alarmas para supervisar las métricas del almacén de claves externo. Estas alarmas le avisarán de las señales tempranas de problemas antes de que se presenten.

Gráficos de monitorización

AWS KMS muestra gráficos de las CloudWatch métricas del almacén de claves externo en la página de detalles de cada almacén de claves externo de la AWS KMS consola. Puede utilizar los datos de los gráficos para localizar el origen de los errores, detectar problemas inminentes, establecer líneas de base y refinar los umbrales de alarma. CloudWatch Para obtener más detalles sobre la interpretación de los gráficos de monitoreo y el uso de sus datos, consulte [Monitoreo de almacenes de claves externos](#).

Visualización de almacenes de claves externos y claves de KMS

AWS KMS muestra información detallada sobre los almacenes de claves externos y las claves de KMS del almacén de claves externo de la AWS KMS consola, así como sobre la respuesta a las operaciones y. [DescribeCustomKeyStoresDescribeKey](#) Estas pantallas incluyen campos especiales para almacenes de claves externos y claves de KMS con información que puede utilizar para solucionar problemas, como el [estado de conexión](#) del almacén de claves externo y el ID de clave externa asociada a la clave de KMS. Para obtener más información, consulte [Visualización de almacenes de claves externos](#).

XKS Proxy Test Client

AWS KMS proporciona un cliente de prueba de código abierto que verifica que el proxy del almacén de claves externo cumpla con la especificación de la [API de proxy del almacén de claves AWS KMS externo](#). Puede utilizar este cliente de prueba para identificar y resolver problemas con su proxy del almacén de claves externo.

Errores de configuración

Al crear un almacén de claves externo, especifica los valores de propiedades que comprenden la configuración del almacén de claves externo, como la [credencial de autenticación del proxy](#), el [punto de conexión URI del proxy](#), la [ruta URI del proxy](#) y el [nombre del servicio de punto de conexión de VPC](#). Cuando AWS KMS detecta un error en el valor de una propiedad, la operación no se realiza correctamente y devuelve un error que indica que el valor es incorrecto.

Muchos problemas de configuración se pueden resolver al corregir el valor incorrecto. Puede corregir una ruta URI del proxy o una credencial de autenticación del proxy no válidas sin desconectar el almacén de claves externo. Para obtener las definiciones de estos valores, incluidos los requisitos de exclusividad, consulte [Cumplir los requisitos previos](#). Para obtener instrucciones sobre cómo actualizar estos valores, consulte [Edición de las propiedades del almacén de claves externo](#).

Para evitar errores con la ruta URI del proxy y los valores de la credencial de autenticación del proxy, al crear o actualizar el almacén de claves externo, cargue un [archivo de configuración del proxy](#) en la consola de AWS KMS. Se trata de un archivo basado en JSON con la ruta URI del proxy y los valores de la credencial de autenticación del proxy que proporciona el proxy del almacén de claves externo o el administrador de claves externo. No puedes usar un archivo de configuración de proxy con las operaciones de la AWS KMS API, pero puedes usar los valores del archivo para ayudarte a proporcionar valores de parámetros para tus solicitudes de API que coincidan con los valores de tu proxy.

Errores de configuración general

Excepciones: `CustomKeyStoreInvalidStateException` (`CreateKey`),
`KMSInvalidStateException` (operaciones criptográficas),
`XksProxyInvalidConfigurationException` (operaciones de administración, excepto `CreateKey`)

[Códigos de error de conexión](#): `XKS_PROXY_INVALID_CONFIGURATION`,
`XKS_PROXY_INVALID_TLS_CONFIGURATION`

En el caso de los almacenes de claves externos con [conectividad de punto final público](#), AWS KMS comprueba los valores de las propiedades al crear y actualizar el almacén de claves externo. En el caso de los almacenes de claves externos con [conectividad al servicio de punto de conexión de VPC](#), AWS KMS prueba los valores de propiedades al conectar y actualizar el almacén de claves externo.

Note

La operación `ConnectCustomKeyStore`, que es asincrónica, puede realizarse correctamente aunque falle el intento de conectar el almacén de claves externo a su proxy del almacén de claves externo. En ese caso, no hay ninguna excepción, pero el estado de conexión del almacén de claves externo es `Failed` (Error) y aparece un código de error de conexión que explica el mensaje de error. Para obtener más información, consulte [Errores de conexión del almacén de claves externo](#).

Si AWS KMS detecta un error en el valor de una propiedad, se produce un error en la operación y se devuelve `XksProxyInvalidConfigurationException` con uno de los siguientes mensajes de error.

El proxy del almacén de claves externo rechazó la solicitud porque la ruta URI no era válida. Verifique la ruta URI del almacén de claves externo y actualícela si es necesario.

- La [ruta URI del proxy](#) es la ruta base para AWS KMS las solicitudes al proxy APIs. Si esta ruta es incorrecta, fallarán todas las solicitudes al proxy. Para [ver la ruta URI del proxy actual](#) del almacén de claves externo, utilice la consola de AWS KMS o la operación `DescribeCustomKeyStores`. Para encontrar la ruta URI del proxy correcta, consulte la documentación del proxy del almacén de claves externo. Para obtener ayuda para corregir el valor de la ruta URI del proxy, consulte [Edición de las propiedades del almacén de claves externo](#).
- La ruta URI del proxy del almacén de claves externo puede cambiar con las actualizaciones del proxy del almacén de claves externo o del administrador de claves externo. Para obtener información sobre estos cambios, consulte la documentación del proxy del almacén de claves externo o administrador de claves externo.

XKS_PROXY_INVALID_TLS_CONFIGURATION

AWS KMS no puede establecer una conexión TLS con el proxy del almacén de claves externo. Verifique la configuración de TLS, incluido su certificado.

- Todos los proxy del almacén de claves externo requieren un certificado TLS. El certificado TLS debe ser emitido por una autoridad de certificación (CA) pública que sea compatible con almacenes de claves externos. Para obtener una lista de las admitidas CAs, consulte [las autoridades certificadoras de confianza](#) en la especificación de la API de proxy del almacén de claves AWS KMS externo.
- Para la conectividad a puntos de conexión públicos, el nombre común (CN) del asunto del certificado TLS debe coincidir con el nombre de dominio del [punto de conexión URI del proxy](#) del proxy del almacén de claves externo. Por ejemplo, si el punto de conexión público es `https://myproxy.xks.example.com`, el TLS, el CN del certificado TLS debe ser `myproxy.xks.example.com` o `*.xks.example.com`.

- Para la conectividad al servicio de punto de conexión de VPC, el nombre común (CN) del asunto del certificado TLS debe coincidir con el nombre DNS privado del [servicio de punto de conexión de VPC](#). Por ejemplo, si el nombre DNS privado es `myproxy-private.xks.example.com`, el CN del certificado TLS debe ser `myproxy-private.xks.example.com` o `*.xks.example.com`.
- El certificado TLS no puede estar caducado. Para obtener la fecha de caducidad de un certificado TLS, utilice herramientas SSL, como [OpenSSL](#). Para controlar la fecha de caducidad de un certificado TLS asociado a un almacén de claves externo, usa la [XksProxyCertificateDaysToExpire](#) CloudWatch métrica. El número de días que faltan para la fecha de caducidad de la certificación TLS también aparece en la [sección Supervisión](#) de la AWS KMS consola.
- Si utiliza una [conectividad a puntos de conexión públicos](#), utilice las herramientas de prueba de SSL para probar la configuración de SSL. Los errores de conexión TLS pueden deberse a un encadenamiento incorrecto de certificados.

Errores de configuración de conectividad al servicio de punto de conexión de VPC

Excepciones: `XksProxyVpcEndpointServiceNotFoundException`,
`XksProxyVpcEndpointServiceInvalidConfigurationException`

Además de los problemas de conectividad generales, es posible que se produzcan los siguientes problemas al crear, conectar o actualizar un almacén de claves externo con la conectividad del servicio de punto final de VPC. AWS KMS prueba los valores de las propiedades de un almacén de claves externo con la conectividad del servicio de punto final de VPC al [crear](#), [conectar](#) y [actualizar](#) el almacén de claves externo. Cuando las operaciones de administración fallan debido a errores de configuración, generan las siguientes excepciones:

`XksProxyVpcEndpointServiceNotFoundException`

Esto podría deberse a una de las siguientes causas:

- Un nombre del servicio de punto de conexión de VPC incorrecto. Compruebe que el nombre del servicio de punto de conexión de VPC del almacén de claves externo sea correcto y que coincida con el valor del punto de conexión URI del proxy del almacén de claves externo. Para encontrar el nombre del servicio de punto final de la VPC, utilice la [consola de Amazon VPC](#) o la operación. [DescribeVpcEndpointServices](#) Para encontrar el nombre del servicio de punto final de la VPC y el extremo URI del proxy de un almacén de claves externo existente, utilice la AWS KMS consola o

la [DescribeCustomKeyStores](#) operación. Para obtener más información, consulte [Visualización de almacenes de claves externos](#).

- El servicio de punto final de la VPC puede estar en un almacén de claves Región de AWS diferente al externo. Verifique que el servicio de punto de conexión de VPC y el almacén de claves externo estén en la misma región. (El nombre externo del nombre de la región, por ejemplo, forma parte del nombre del servicio de punto final de la VPCus-east-1, como com.amazonaws.vpce.us-east-1.vpce-svc-example.) Para obtener una lista de los requisitos del servicio de punto de conexión de VPC para un almacén de claves externo, consulte [Servicio de punto de conexión de VPC](#). No puede mover un servicio de punto de conexión de VPC ni un almacén de claves externo a otra región. Sin embargo, puede crear un nuevo almacén de claves externo en la misma región que el servicio de punto de conexión de VPC. Para más detalles, consulte [Configuración de la conectividad del servicio de punto de conexión de VPC](#) y [Creación de un almacén de claves externo](#).
- AWS KMS no es un principal permitido para el servicio de punto final de la VPC. La lista de Entidades principales permitidas para el servicio de punto de conexión de VPC debe incluir el valor cks.kms.<region>.amazonaws.com, por ejemplo cks.kms.eu-west-3.amazonaws.com. Para obtener instrucciones sobre cómo agregar este valor, consulte [Administrar permisos](#) en la Guía de AWS PrivateLink.

XksProxyVpcEndpointServiceInvalidConfigurationException

Este error se produce cuando el servicio de punto de conexión de VPC no cumple uno de los siguientes requisitos:

- La VPC requiere al menos dos subredes privadas, cada una en una zona de disponibilidad diferente. Para agregar una subred a la VPC, consulte [Crear una subred en la VPC](#) en la Guía del usuario de Amazon VPC.
- El [tipo de servicio de punto de conexión de VPC](#) debe usar un equilibrador de carga de red, no un equilibrador de carga de puerta de enlace.
- No debe exigirse la aceptación del servicio de punto de conexión de VPC (la aceptación obligatoria debe ser falsa). Si se requiere la aceptación manual de cada solicitud de conexión, AWS KMS no podrá utilizar el servicio de punto final de la VPC para conectarse al proxy del almacén de claves externo. Para obtener más información, consulte [Aceptar o rechazar solicitudes de conexión](#) en la Guía de AWS PrivateLink.

- El servicio de punto de conexión de VPC debe tener un nombre DNS privado que sea un subdominio de un dominio público. Por ejemplo, si el nombre DNS privado es `https://myproxy-private.xks.example.com`, los dominios `xks.example.com` o `example.com` deben tener un servidor de DNS público. Para ver o cambiar el nombre DNS privado del servicio de punto de conexión de VPC, consulte [Administrar nombres DNS de los servicios de punto de conexión de VPC](#) en la Guía de AWS PrivateLink.
- El estado de verificación de dominio del dominio de su nombre DNS privado debe ser `verified`. Para ver y actualizar el estado de verificación del dominio de nombre DNS privado, consulte [Paso 5: Verificar el dominio de su nombre DNS privado](#). Es posible que el estado de verificación actualizado tarde unos minutos en aparecer después de agregar el registro de texto requerido.

 Note

Un dominio DNS privado solo se puede verificar si es el subdominio de un dominio público. De lo contrario, el estado de verificación del dominio de DNS privado no cambiará, incluso después de agregar el registro TXT requerido.

- Asegúrese de que los firewalls situados entre AWS KMS y el proxy del almacén de claves externo permitan el tráfico hacia y desde el puerto 443 del proxy. AWS KMS se comunica a través del puerto 443. IPv4 Este valor no se puede configurar.
- El nombre DNS privado del servicio de punto de conexión de VPC debe coincidir con el valor del [punto de conexión URI del proxy](#) para el almacén de claves externo. Para un almacén de claves externo con conectividad al servicio de punto de conexión de VPC, el punto de conexión URI del proxy debe ser `https://` seguido del nombre DNS privado del servicio de punto de conexión de VPC. Para ver el valor del punto de conexión URI del proxy, consulte [Visualización de almacenes de claves externos](#). Para cambiar el valor del punto de conexión URI del proxy, consulte [Edición de las propiedades del almacén de claves externo](#).

Errores de conexión del almacén de claves externo

El [proceso de conexión de un almacén de claves externo](#) a su proxy de almacén de claves externo tarda unos cinco minutos en completarse. A menos que el error sea rápido, la operación `ConnectCustomKeyStore` devuelve una respuesta HTTP 200 y un objeto JSON sin propiedades. Sin embargo, esta respuesta inicial no indica que la conexión se haya realizado correctamente. Para determinar si el almacén de claves externo está conectado, consulte su [estado de conexión](#). Si la

conexión falla, el estado de conexión del almacén de claves externo cambia FAILED y AWS KMS devuelve un [código de error de conexión](#) que explica la causa del error.

Note

Si el estado de conexión de un almacén de claves personalizado es FAILED, deberá desconectar el almacén de claves personalizado antes de conectarlo de nuevo. No puede conectar un almacén de claves personalizado que tenga el estado de conexión FAILED.

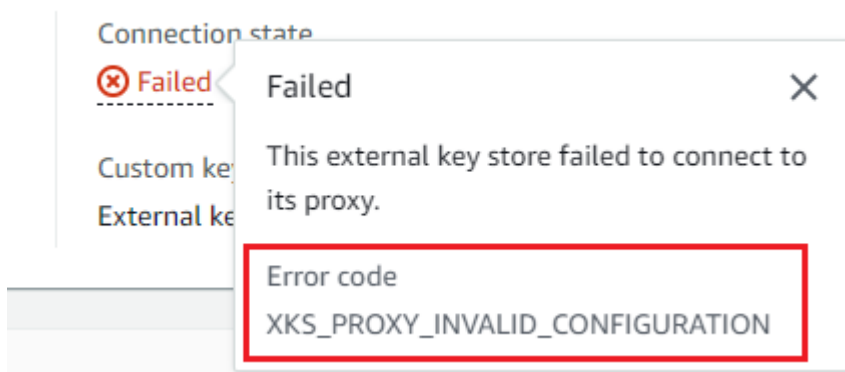
Para ver el estado de conexión de un almacén de claves externo:

- En la [DescribeCustomKeyStores](#) respuesta, vea el valor del ConnectionState elemento.
- En la AWS KMS consola, el estado de la conexión aparece en la tabla del almacén de claves externo. Además, en la página de detalles de cada almacén de claves externo, aparece el estado de conexión en la sección Configuración general.

Cuando el estado de la conexión es FAILED, el código de error de conexión ayuda a explicar el error.

Para ver el código de error de conexión:

- En la [DescribeCustomKeyStores](#) respuesta, vea el valor del ConnectionErrorCode elemento. Este elemento aparece en la respuesta DescribeCustomKeyStores solo cuando el ConnectionState es FAILED.
- Para ver el código de error de conexión en la AWS KMS consola, vaya a la página de detalles del almacén de claves externo y coloque el cursor sobre el valor de error.



Códigos de error de conexión para almacenes de claves externos

Los siguientes códigos de error de conexión aplican a los almacenes de claves externos

INTERNAL_ERROR

AWS KMS no se pudo completar la solicitud debido a un error interno. Intente realizar de nuevo la solicitud. Para las solicitudes `ConnectCustomKeyStore`, desconecte el almacén de claves personalizado antes de intentar conectarse de nuevo.

INVALID_CREDENTIALS

Uno o ambos valores de `XksProxyAuthenticationCredential` no son válidos en el proxy del almacén de claves externo especificado.

NETWORK_ERRORS

Los errores de red AWS KMS impiden conectar el almacén de claves personalizado al almacén de claves secundario.

XKS_PROXY_ACCESS_DENIED

AWS KMS a las solicitudes se les deniega el acceso al proxy del almacén de claves externo. Si el proxy del almacén de claves externo tiene reglas de autorización, compruebe que permitan a AWS KMS comunicarse con el proxy en su nombre.

XKS_PROXY_INVALID_CONFIGURATION

Un error de configuración impide que el almacén de claves externo se conecte a su proxy. Compruebe el valor de `XksProxyUriPath`.

XKS_PROXY_INVALID_RESPONSE

AWS KMS no puede interpretar la respuesta del proxy del almacén de claves externo. Si ve este código de error de conexión varias veces, comuníquese al proveedor del proxy del almacén de claves externo.

XKS_PROXY_INVALID_TLS_CONFIGURATION

AWS KMS no puede conectarse al proxy del almacén de claves externo porque la configuración de TLS no es válida. Compruebe que el proxy del almacén de claves externo sea compatible con TLS 1.2 o 1.3. Además, compruebe que el certificado TLS no esté caducado, que coincida con el nombre de host en el valor `XksProxyUriEndpoint` y que esté firmado por una autoridad de certificación de confianza incluida en la lista de [Autoridades de certificación de confianza](#).

XKS_PROXY_NOT_REACHABLE

AWS KMS no puede comunicarse con el proxy del almacén de claves externo. Compruebe que `XksProxyUriEndpoint` y `XksProxyUriPath` sean correctos. Utilice las herramientas de su proxy del almacén de claves externo para comprobar que el proxy esté activo y disponible en su red. Además, compruebe que las instancias del administrador de claves externo funcionen correctamente. Los intentos de conexión fallan con este código de error de conexión si el proxy informa que todas las instancias del administrador de claves externo no están disponibles.

XKS_PROXY_TIMED_OUT

AWS KMS puede conectarse al proxy del almacén de claves externo, pero el proxy no responde AWS KMS en el tiempo asignado. Si ve este código de error de conexión varias veces, comuníquese al proveedor del proxy del almacén de claves externo.

XKS_VPC_ENDPOINT_SERVICE_INVALID_CONFIGURATION

La configuración del servicio de puntos de conexión de Amazon VPC no cumple con los requisitos de un almacén de claves AWS KMS externo.

- El servicio de punto de conexión de VPC debe ser un servicio de punto de conexión para los puntos de conexión de la interfaz de la Cuenta de AWS de la persona que llama.
- Debe tener un equilibrador de carga de red (NLB) conectado a al menos dos subredes, cada una en una zona de disponibilidad diferente.
- La `Allow principals` lista debe incluir el principal AWS KMS de servicio de la región `cks.kms.<region>.amazonaws.com`, por ejemplo `cks.kms.us-east-1.amazonaws.com`.
- No debe requerir la [aceptación](#) de las solicitudes de conexión.
- Debe tener un nombre DNS privado. El nombre DNS privado de un almacén de claves externo con conectividad `VPC_ENDPOINT_SERVICE` debe ser único en su Región de AWS.
- El dominio del nombre DNS privado debe tener un [estado de verificación](#) `verified`.
- El [certificado TLS](#) especifica el nombre de host DNS privado en el que se puede acceder al punto de conexión.

XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND

AWS KMS no encuentra el servicio de punto final de la VPC que utiliza para comunicarse con el proxy del almacén de claves externo. Compruebe que `XksProxyVpcEndpointServiceName` es correcto y que la entidad principal del servicio de AWS KMS tiene permisos de consumidor del servicio en el servicio de punto de conexión de VPC de Amazon.

Errores de latencia y tiempo de espera

Excepciones: `CustomKeyStoreInvalidStateException` (`CreateKey`),
`KMSInvalidStateException` (operaciones criptográficas),
`XksProxyUriUnreachableException` (operaciones de administración)

Códigos de error de conexión: `XKS_PROXY_NOT_REACHABLE`, `XKS_PROXY_TIMED_OUT`

Si no AWS KMS puede contactar con el proxy dentro del intervalo de tiempo de espera de 250 milisegundos, devuelve una excepción. `CreateCustomKeyStore` `UpdateCustomKeyStore` volver. `XksProxyUriUnreachableException` Las operaciones criptográficas generan la `KMSInvalidStateException` estándar con un mensaje de error que describe el problema. Si `ConnectCustomKeyStore` falla, AWS KMS devuelve un código de error de conexión que describe el problema.

Los errores de tiempo de espera pueden ser problemas transitorios que se pueden resolver volviendo a intentar la solicitud. Si el problema persiste, compruebe que el proxy del almacén de claves externo esté activo y conectado a la red. Además, compruebe que el punto de conexión URI del proxy, la ruta URI del proxy y el nombre del servicio de punto de conexión de VPC (si lo hubiera) sean correctos en el almacén de claves externo. Compruebe también que el administrador de claves externo esté cerca del almacén Región de AWS de claves externo. Si necesita actualizar alguno de estos valores, consulte Edición de las propiedades del almacén de claves externo.

Para realizar un seguimiento de los patrones de latencia, utiliza la `XksProxyLatency` CloudWatch métrica y el gráfico de latencia media (basado en esa métrica) de la sección Supervisión de la AWS KMS consola. El proxy de su almacén de claves externo también puede generar registros y métricas que rastreen la latencia y los tiempos de espera.

`XksProxyUriUnreachableException`

AWS KMS no se puede comunicar con el proxy del almacén de claves externo. Esto puede ser un problema de red transitorio. Si aparece este error varias veces, compruebe que el proxy del almacén de claves externo esté activo y conectado a la red, y que el URI del punto de conexión sea correcto en el almacén de claves externo.

- El proxy del almacén de claves externo no respondió a una solicitud de API de AWS KMS proxy dentro del intervalo de espera de 250 milisegundos. Esto puede indicar un problema de red transitorio o un problema operativo o de rendimiento con el proxy. Si volver a intentarlo no resuelve el problema, notifíquelo al administrador de proxy del almacén de claves externo.

Los errores de latencia y tiempo de espera suelen manifestarse como errores de conexión. Cuando se produce un error en la [ConnectCustomKeyStore](#) operación, el estado de conexión del almacén de claves externo cambia FAILED y AWS KMS devuelve un código de error de conexión que explica el error. Para obtener una lista de los códigos de error de conexión y sugerencias para resolverlos, consulte [Códigos de error de conexión para almacenes de claves externos](#). Las listas de códigos de conexión de Todos los almacenes de claves personalizados y Almacenes de claves externos aplican a los almacenes de claves externos. Los siguientes errores de conexión están relacionados con la latencia y los tiempos de espera.

XKS_PROXY_NOT_REACHABLE

-o bien-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ,
`XksProxyUriUnreachableException`

AWS KMS no se puede comunicar con el proxy del almacén de claves externo. Compruebe que el proxy del almacén de claves externo esté activo y conectado a la red y que su ruta URI y su nombre del servicio de VPC o URI de punto de conexión sean correctos en su almacén de claves externo.

Este error puede producirse por las siguientes razones:

- El proxy del almacén de claves externo no está activo o no está conectado a la red.
- Hay un error en los valores del [punto de conexión URI del proxy](#), la [ruta URI del proxy](#) o el [nombre del servicio de punto de conexión de VPC](#) (si corresponde) en la configuración del almacén de claves externo. Para ver la configuración del almacén de claves externo, utilice la [DescribeCustomKeyStores](#) operación o [consulte la página de detalles](#) del almacén de claves externo en la AWS KMS consola.
- Es posible que haya un error de configuración de red, como un error de puerto, en la ruta de red entre AWS KMS y el proxy del almacén de claves externo. AWS KMS se comunica con el proxy del almacén de claves externo en el puerto 443 IPv4. Este valor no se puede configurar.
- Cuando el proxy del almacén de claves externo informa (en una [GetHealthStatus](#) respuesta) de que todas las instancias del administrador de claves externo UNAVAILABLE lo están, la [ConnectCustomKeyStore](#) operación falla con un `ConnectionErrorCode` de `XKS_PROXY_NOT_REACHABLE`. Para obtener ayuda, consulte la documentación del administrador de claves externo.

- Este error puede deberse a una gran distancia física entre el administrador de claves externo y el Región de AWS almacén de claves externo. La latencia del ping (tiempo de ida y vuelta de la red (RTT)) entre el administrador de claves externo Región de AWS y el administrador de claves externo no debe ser superior a 35 milisegundos. Puede que tenga que crear un almacén de claves externo en una Región de AWS que esté más cerca del administrador de claves externo o mover el administrador de claves externo a un centro de datos que esté más cerca del Región de AWS.

XKS_PROXY_TIMED_OUT

-o bien-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ,
`XksProxyUriUnreachableException`

AWS KMS rechazó la solicitud porque el proxy del almacén de claves externo no respondió a tiempo. Intente realizar de nuevo la solicitud . Si ve este error varias veces, infórmelo al administrador de proxy del almacén de claves externo.

Este error puede producirse por las siguientes razones:

- Este error puede deberse a una gran distancia física entre el administrador de claves externo y el proxy del almacén de claves externo. Si es posible, acerque el proxy del almacén de claves externo al administrador de claves externo.
- Los errores de tiempo de espera pueden producirse cuando el proxy no está diseñado para gestionar el volumen y la frecuencia de las solicitudes procedentes de AWS KMS de él. Si tus CloudWatch métricas indican un problema persistente, notifícalo al administrador del proxy externo del almacén de claves.
- Se pueden producir errores de tiempo de espera cuando la conexión entre el administrador de claves externo y Amazon VPC para el almacén de claves externo no funciona correctamente. Si lo está utilizando AWS Direct Connect, compruebe que la VPC y el administrador de claves externo se puedan comunicar de forma eficaz. Para obtener ayuda para resolver cualquier problema, consulte [Solución de problemas AWS Direct Connect](#) en la Guía del Direct Connect usuario.

XKS_PROXY_TIMED_OUT

-o bien-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ,
`XksProxyUriUnreachableException`

El proxy del almacén de claves externo no respondió a la solicitud en el tiempo establecido.

Intente realizar de nuevo la solicitud . Si ve este error varias veces, infórmelo al administrador de proxy del almacén de claves externo.

- Este error puede deberse a una gran distancia física entre el administrador de claves externo y el proxy del almacén de claves externo. Si es posible, acerque el proxy del almacén de claves externo al administrador de claves externo.

Errores en las credenciales de autenticación

Excepciones: `CustomKeyStoreInvalidStateException` (`CreateKey`),
`KMSInvalidStateException` (operaciones criptográficas),
`XksProxyIncorrectAuthenticationCredentialException` (operaciones de administración distintas de `CreateKey`)

Debe establecer y mantener una credencial de autenticación para su proxy AWS KMS de almacén de claves externo. A continuación, indica AWS KMS los valores de las credenciales al crear un almacén de claves externo. Para cambiar la credencial de autenticación, realice el cambio en el proxy del almacén de claves externo. A continuación, [actualice la credencial](#) del almacén de claves externo. Si su proxy rota la credencial, debe [actualizar la credencial](#) del almacén de claves externo.

Si el proxy del almacén de claves externo no autentica una solicitud firmada con la [credencial de autenticación de proxy](#) del almacén de claves externo, el efecto depende de la solicitud:

- `CreateCustomKeyStore` y `UpdateCustomKeyStore` fallan con una `XksProxyIncorrectAuthenticationCredentialException`.
- `ConnectCustomKeyStore` se realiza correctamente, pero se produce un error en la conexión. El estado de la conexión es `FAILED` y el código de error de conexión es `INVALID_CREDENTIALS`. Para obtener más información, consulte [Errores de conexión del almacén de claves externo](#).
- Las operaciones criptográficas devuelven `KMSInvalidStateException` para todos los errores de configuración y estado de conexión externos de un almacén de claves externo. El mensaje de error adjunto describe el problema.

El proxy del almacén de claves externo rechazó la solicitud porque no pudo autenticar AWS KMS. Compruebe las credenciales del almacén de claves externo y actualícelas si es necesario.

Este error puede producirse por las siguientes razones:

- El ID de clave de acceso o la clave de acceso secreta del almacén de claves externo no coincide con los valores establecidos en el proxy del almacén de claves externo.

Para corregir este error, [actualice la credencial de autenticación de proxy](#) del almacén de claves externo. Puede realizar este cambio sin desconectar el almacén de claves externo.

- Un proxy inverso entre AWS KMS y el proxy del almacén de claves externo podría manipular los encabezados HTTP de forma que se invalidaran las firmas de SiGv4. Para corregir este error, notifique al administrador de proxy.

Errores de estados de las claves

Excepciones: `KMSInvalidStateException`

`KMSInvalidStateException` se utiliza para dos propósitos distintos para las claves de KMS en almacenes de claves personalizados.

- Cuando una operación de administración, como `CancelKeyDeletion`, falla y devuelve esta excepción, indica que el [estado de la clave](#) de la clave de KMS no es compatible con la operación.
- Cuando se produce un error con `KMSInvalidStateException` en una [operación criptográfica](#) en una clave de KMS de un almacén de claves personalizado, puede indicar un problema con el estado de la clave de KMS. Sin embargo, la operación AWS KMS criptográfica se devuelve `KMSInvalidStateException` para todos los errores de configuración externa y de estado de conexión en un almacén de claves externo. Para identificar el problema, utilice el mensaje de error que acompaña a la excepción.

Para encontrar el estado clave necesario para las operaciones de una AWS KMS API, consulte [Estados clave de AWS KMS claves](#). Para buscar el estado de clave de una clave KMS en la página Customer managed keys (Claves administradas por el cliente) consulte el campo Status (Estado) de la clave KMS. O bien, utilice la [DescribeKey](#) operación y visualice el `KeyState` elemento en la respuesta. Para obtener más información, consulte [Identificación y visualización de claves](#).

 Note

El estado de clave de una clave de KMS en un almacén de claves externo no indica nada sobre el estado de la [clave externa](#) asociada. Para obtener información sobre el estado de la clave externa, utilice el administrador de claves externo y las herramientas de proxy del almacén de claves externo.

`CustomKeyStoreInvalidStateException` se refiere al [estado de conexión](#) del almacén de claves externo, no al [estado de clave](#) de una clave de KMS.

Una operación criptográfica en una clave KMS de un almacén de claves personalizado puede producir un error porque el estado de la clave KMS es `Unavailable` o `PendingDeletion`. (Las teclas deshabilitadas devuelven `DisabledException`).

- Una clave de KMS tiene un estado de `Disabled` clave solo cuando se deshabilita intencionadamente la clave de KMS en la AWS KMS consola o mediante esta [DisableKey](#) operación. Mientras una clave de KMS esté deshabilitada, puede consultarla y administrarla, pero no puede usarla en operaciones criptográficas. Para solucionar este problema, habilite la clave. Para obtener más información, consulte [Habilitar y deshabilitar claves](#).
- Una clave de KMS tiene un estado de clave `Unavailable` cuando el almacén de claves externo se desconecta de su proxy del almacén de claves externo. Para arreglar una clave de KMS no disponible, [vuelva a conectar el almacén de claves externo](#). Después de volver a conectar el almacén de claves externo, el estado de clave de las claves de KMS en el almacén de claves externo se restaura automáticamente al estado anterior, como `Enabled` o `Disabled`.

Una clave de KMS tiene un estado de clave `PendingDeletion` cuando se ha programado su eliminación y se encuentra en su periodo de espera. Un error de estado de clave en una clave de KMS que está pendiente de eliminación indica que la clave no se debe eliminar, ya sea porque se utiliza para el cifrado o porque es necesaria para el descifrado. Para volver a habilitar la clave de KMS, cancele la eliminación programada y, a continuación, [habilite la clave](#). Para obtener más información, consulte [Programar la eliminación de claves](#).

Errores de descifrado

Excepciones: `KMSInvalidStateException`

Cuando se produce un error en una operación de [descifrado](#) con una clave KMS de un almacén de claves externo, AWS KMS devuelve el estándar `KMSInvalidStateException` que utilizan las operaciones criptográficas para todos los errores de configuración externa y de estado de conexión en un almacén de claves externo. El mensaje de error indica el problema.

Para descifrar un texto cifrado que fue cifrado mediante [doble cifrado](#), el administrador de claves externo utiliza primero la clave externa para descifrar la capa exterior del texto cifrado. A continuación, AWS KMS utiliza el material AWS KMS clave de la clave KMS para descifrar la capa interna del texto cifrado. AWS KMS o un administrador de claves externo puede rechazar un texto cifrado no válido o dañado.

Cuando se produce un error en el descifrado, aparecen los siguientes mensajes de error junto a `KMSInvalidStateException`. Indica un problema con el texto cifrado o el contexto de cifrado opcional de la solicitud.

El proxy del almacén de claves externo rechazó la solicitud porque el texto cifrado especificado o los datos autenticados adicionales están dañados, faltan o no son válidos por algún motivo.

- Cuando el proxy del almacén de claves externo o el administrador de claves externo informan que un texto cifrado o su contexto de cifrado no son válidos, normalmente indica que hay un problema con el texto cifrado o el contexto de cifrado de la solicitud enviada a. Decrypt AWS KMS Para Decrypt las operaciones, AWS KMS envía al proxy el mismo texto cifrado y el mismo contexto de cifrado que recibe en la solicitud. Decrypt

Este error puede deberse a un problema de red durante el transporte, como un bit invertido. Intente realizar de nuevo la solicitud Decrypt. Si el problema persiste, compruebe que el texto cifrado no esté alterado ni dañado. Además, compruebe que el contexto de cifrado de la Decrypt solicitud AWS KMS coincide con el contexto de cifrado de la solicitud que cifró los datos.

El texto cifrado que el proxy del almacén de claves externo envió para su descifrado, o el contexto de cifrado, está dañado, falta o no es válido por algún motivo.

- Cuando AWS KMS rechaza el texto cifrado que recibió del proxy, indica que el administrador de claves externo o el proxy devolvieron un texto cifrado no válido o dañado. AWS KMS

Este error puede deberse a un problema de red durante el transporte, como un bit invertido. Intente realizar de nuevo la solicitud Decrypt. Si el problema persiste, compruebe que el administrador de claves externo funciona correctamente y que el proxy del almacén de claves externo no altera el texto cifrado que recibe del administrador de claves externo antes de devolverlo a AWS KMS.

Errores de clave externa

Una [clave externa](#) es una clave criptográfica del administrador de claves externo que sirve como material de clave externa para una clave de KMS. AWS KMS no puede acceder directamente a la clave externa. Debe solicitar al administrador de claves externo (a través del proxy del almacén de claves externo) que utilice la clave externa para cifrar datos o descifrar un texto cifrado.

Especifica el ID de clave externa en su administrador de claves externo cuando crea una clave de KMS en su almacén de claves externo. No se puede cambiar el ID de clave externa después de crear la clave de KMS. Para evitar problemas con la clave de KMS, la operación CreateKey solicita al proxy del almacén de claves externo que verifique el ID y la configuración de la clave externa. Si la clave externa no [cumple los requisitos](#) para su uso con una clave de KMS, se produce un error en la operación CreateKey y aparece un mensaje de excepción y error que identifica el problema.

Sin embargo, pueden producirse problemas después de que se crea la clave de KMS. Si una operación criptográfica falla debido a un problema con la clave externa, se produce un error y se genera una `KMSInvalidStateException` con un mensaje de error que indica el problema.

CreateKey errores en la clave externa

Excepciones: `XksKeyAlreadyInUseException`, `XksKeyNotFoundException`, `XksKeyInvalidConfigurationException`

La [CreateKey](#) operación intenta comprobar el identificador y las propiedades de la clave externa que se proporciona en el parámetro ID de clave externa (consola) o `XksKeyId` (API). Esta práctica está diseñada para detectar errores de forma temprana antes de intentar utilizar la clave externa con la clave de KMS.

Clave externa en uso

Cada clave de KMS de un almacén de claves externo debe usar una clave externa diferente. Cuando CreateKey reconoce que el identificador de clave externa (`XksKeyId`) de una clave de KMS no es único en el almacén de claves externo, se produce un error con `unXksKeyAlreadyInUseException`.

Si usa varios IDs para la misma clave externa, `CreateKey` no reconocerá el duplicado. Sin embargo, las claves de KMS con la misma clave externa no son interoperables porque tienen diferentes metadatos y materiales de AWS KMS claves.

No se encontró la clave externa

Cuando el proxy del almacén de claves externo informa que no puede encontrar la clave externa utilizando el identificador de clave externa (`XksKeyId`) de la clave KMS, la `CreateKey` operación falla y vuelve `XksKeyNotFoundException` con el siguiente mensaje de error.

El proxy del almacén de claves externo rechazó la solicitud porque no pudo encontrar la clave externa.

Este error puede producirse por las siguientes razones:

- Es posible que el ID de la clave externa (`XksKeyId`) de la clave de KMS no sea válido. Para encontrar el ID que su proxy de clave externa utiliza para identificar la clave externa, consulte la documentación sobre el proxy del almacén de claves externo o el administrador de claves externo.
- Puede que se haya eliminado la clave externa de su administrador de claves externo. Para investigar, utilice sus herramientas de administrador de claves externo. Si la clave externa se elimina permanentemente, utilice una clave externa diferente con la clave de KMS. Para obtener una lista de los requisitos de la clave externa, consulte [Requisitos para una clave de KMS en un almacén de claves externo](#).

No se cumplen los requisitos de clave externa

Cuando el proxy del almacén de claves externo informa que la clave externa no [cumple con los requisitos](#) para su uso con una clave KMS, la operación `CreateKey` falla y genera una `XksKeyInvalidConfigurationException` con uno de los siguientes mensajes de error.

La especificación de clave de la clave externa debe ser `AES_256`. La especificación clave de la clave externa especificada es `<key-spec>` .

- La clave externa debe ser una clave de cifrado simétrica de 256 bits con una especificación de clave de `AES_256`. Si la clave externa especificada es de otro tipo, especifique el ID de una clave externa que cumpla este requisito.

El estado de la clave externa debe ser ENABLED (HABILITADA). El estado de la clave externa especificada es `<status>`.

- La clave externa debe estar habilitada en el administrador de claves externo. Si la clave externa especificada no está habilitada, utilice las herramientas del administrador de claves externo para habilitarla o especifique una clave externa habilitada.

El uso de clave de la clave externa debe incluir ENCRYPT (CIFRAR) y DECRYPT (DESCIFRAR). El uso de la clave externa especificada es `< key-usage >`.

- La clave externa debe configurarse para el cifrado y el descifrado en el administrador de claves externo. Si la clave externa especificada no incluye estas operaciones, utilice las herramientas del administrador de claves externo para cambiar las operaciones o especifique una clave externa diferente.

Errores de operación criptográfica para la clave externa

Excepciones: `KMSInvalidStateException`

Cuando el proxy del almacén de claves externo no puede encontrar la clave externa asociada a la clave de KMS o la clave externa no [cumple los requisitos](#) para su uso con una clave de KMS, se produce un error en la operación criptográfica.

Los problemas de clave externa que se detectan durante una operación criptográfica son más difíciles de resolver que los problemas de clave externa detectados antes de crear la clave de KMS. No se puede cambiar el ID de clave externa después de crear la clave de KMS. Si la clave de KMS aún no ha cifrado ningún dato, puede eliminarla y crear una nueva con un ID de clave externa diferente. Sin embargo, el texto cifrado generado con la clave KMS no se puede descifrar con ninguna otra clave KMS, ni siquiera con la misma clave externa, ya que las claves tendrán metadatos y materiales clave diferentes AWS KMS . En su lugar, en la medida de lo posible, utilice las herramientas de administración de claves externas para resolver el problema con la clave externa.

Cuando el proxy del almacén de claves externo informa un problema con la clave externa, las operaciones criptográficas devuelven una `KMSInvalidStateException` con un mensaje de error que identifica el problema.

No se encontró la clave externa

Cuando el proxy del almacén de claves externo informa que no puede encontrar la clave externa utilizando el identificador de clave externa (`XksKeyId`) de la clave KMS, las operaciones criptográficas devuelven un mensaje de error `KMSInvalidStateException` con el siguiente mensaje de error.

El proxy del almacén de claves externo rechazó la solicitud porque no pudo encontrar la clave externa.

Este error puede producirse por las siguientes razones:

- El ID de la clave externa (`XksKeyId`) de la clave de KMS ya no es válido.

Para encontrar el ID de clave externa asociado a su clave de KMS, [consulte los detalles de la clave de KMS](#). Para encontrar el ID que su proxy de clave externa utiliza para identificar la clave externa, consulte la documentación sobre el proxy del almacén de claves externo o el administrador de claves externo.

AWS KMS verifica el ID de clave externa cuando crea una clave KMS en un almacén de claves externo. Sin embargo, el ID puede dejar de ser válido, especialmente si el valor del ID de clave externa es un alias o un nombre mutable. No puede cambiar el ID de clave externa asociado a una clave de KMS existente. Para descifrar cualquier texto cifrado con la clave de KMS, debe volver a asociar la clave externa con el ID de clave externa existente.

Si aún no ha utilizado la clave de KMS para cifrar datos, puede crear una nueva clave de KMS con un ID de clave externa válido. Sin embargo, si ha generado texto cifrado con la clave de KMS, no puede usar ninguna otra clave de KMS para descifrar el texto cifrado, incluso si usa la misma clave externa.

- Puede que se haya eliminado la clave externa de su administrador de claves externo. Para investigar, utilice sus herramientas de administrador de claves externo. Si es posible, intente [recuperar el material de clave](#) de una copia o respaldo de su administrador de claves externo. Si la clave externa se elimina de forma permanente, cualquier texto cifrado con la clave de KMS asociada será irrecuperable.

Errores de configuración de claves externas

Cuando el proxy del almacén de claves externo informa que la clave externa no [cumple con los requisitos](#) para su uso con una clave KMS, la operación criptográfica genera una `KMSInvalidStateException` con uno de los siguientes mensajes de error.

El proxy del almacén de claves externo rechazó la solicitud porque la clave externa no admite la operación solicitada.

- La clave externa debe admitir tanto el cifrado como el descifrado. Si el uso de la clave no incluye el cifrado y el descifrado, utilice las herramientas de administración de claves externas para cambiar el uso de la clave.

El proxy del almacén de claves externo rechazó la solicitud porque la clave externa no está habilitada en el administrador de claves externo.

- La clave externa debe estar habilitada y disponible para su uso en el administrador de claves externo. Si el estado de la clave externa no es `Enabled`, utilice las herramientas del administrador de claves externo para habilitarla.

Problemas con el proxy

Excepciones:

`CustomKeyStoreInvalidStateException (CreateKey)`, `KMSInvalidStateException` (operaciones criptográficas), `UnsupportedOperationException`, `XksProxyUriUnreachableException`, `XksProxyInvalidResponseException` (operaciones de administración distintas de `CreateKey`)

El proxy del almacén de claves externo interviene en todas las comunicaciones entre AWS KMS y el administrador de claves externo. Traduce AWS KMS las solicitudes genéricas a un formato que su administrador de claves externo pueda entender. Si el proxy del almacén de claves externo no cumple con la [especificación de la API de proxy del almacén de claves AWS KMS externo](#), o si no funciona correctamente o no se puede comunicar con él AWS KMS, no podrás crear ni usar claves de KMS en tu almacén de claves externo.

Si bien muchos errores mencionan el proxy del almacén de claves externo debido a su papel fundamental en la arquitectura del almacén de claves externo, esos problemas pueden originarse en el administrador de claves externo o en la clave externa.

Los problemas de esta sección se refieren a problemas con el diseño o el funcionamiento del proxy del almacén de claves externo. La resolución de estos problemas puede requerir un cambio en el software de proxy. Consulte a su administrador de proxy. Para ayudar a diagnosticar problemas con el proxy, AWS KMS proporciona [XKS Proxy Text Client](#), un cliente de prueba de código abierto que comprueba que el proxy del almacén de claves externo cumple con la [Especificación de la API de proxy del almacén de claves externo de AWS KMS](#).

```
CustomKeyStoreInvalidStateException , KMSInvalidStateException o  
XksProxyUriUnreachableException
```

El proxy del almacén de claves externo se encuentra en mal estado. Si ve este mensaje varias veces, notifíquelo al administrador de proxy del almacén de claves externo.

- Este error puede indicar un problema operativo o un error de software en el proxy del almacén de claves externo. Puedes encontrar las entradas de CloudTrail registro de la operación de AWS KMS API que generó cada error. Este error puede resolverse si vuelve a intentar la operación. Sin embargo, si persiste, notifíquelo al administrador de proxy del almacén de claves externo.
- Cuando el proxy del almacén de claves externo informa (en una [GetHealthStatus](#) respuesta) de que todas las instancias del administrador de claves externo lo están UNAVAILABLE, los intentos de crear o actualizar un almacén de claves externo fallan, con esta excepción. Si el error persiste, consulte la documentación del administrador de claves externo.

```
CustomKeyStoreInvalidStateException , KMSInvalidStateException o  
XksProxyInvalidResponseException
```

AWS KMS no puede interpretar la respuesta del proxy del almacén de claves externo. Si ve este error varias veces, consulte con el administrador de proxy del almacén de claves externo.

- AWS KMS las operaciones generan esta excepción cuando el proxy devuelve una respuesta indefinida que AWS KMS no se puede analizar ni interpretar. Este error puede producirse ocasionalmente debido a problemas externos temporales o a errores de red esporádicos. Sin embargo, si persiste, podría indicar que el proxy del almacén de claves externo no cumple con

la [Especificación de la API de proxy del almacén de claves externo de AWS KMS](#). Notifique a su proveedor o administrador del almacén de claves externo.

`CustomKeyStoreInvalidStateException`, `KMSInvalidStateException` o `UnsupportedOperationException`

El proxy del almacén de claves externo rechazó la solicitud porque no admite la operación criptográfica solicitada.

- El proxy del almacén de claves externo debe admitir todos los [proxies APIs](#) definidos en la [especificación de la API de proxy del almacén de claves AWS KMS externo](#). Este error indica que el proxy no admite la operación relacionada con la solicitud. Notifique a su proveedor o administrador del almacén de claves externo.

Problemas de autorización de proxy

Excepciones: `CustomKeyStoreInvalidStateException`, `KMSInvalidStateException`

Algunos proxy del almacén de claves externo implementan requisitos de autorización para el uso de sus claves externas. Se permite, pero no es obligatorio, utilizar un proxy de almacén de claves externo para diseñar e implementar un esquema de autorización que permita a determinados usuarios solicitar ciertas operaciones en determinadas condiciones. Por ejemplo, un proxy puede permitir al usuario cifrar con una clave externa determinada, pero no descifrar con ella. Para obtener más información, consulte [Autorización del proxy del almacén de claves externo \(opcional\)](#).

La autorización del proxy se basa en los metadatos que AWS KMS incluye en sus solicitudes al proxy. Los campos `awsSourceVpc` y `awsSourceVpce` se incluyen en los metadatos solo cuando la solicitud proviene de un punto de conexión de VPC y solo cuando la persona que llama está en la misma cuenta que la clave de KMS.

```
"requestMetadata": {
  "awsPrincipalArn": string,
  "awsSourceVpc": string, // optional
  "awsSourceVpce": string, // optional
  "kmsKeyArn": string,
  "kmsOperation": string,
  "kmsRequestId": string,
```

```
"kmsViaService": string // optional  
}
```

Cuando el proxy rechaza una solicitud debido a un error de autorización, se produce un error en la AWS KMS operación relacionada.

CreateKey devuelve `CustomKeyStoreInvalidStateException`. AWS KMS las operaciones criptográficas regresan `KMSInvalidStateException`. Ambos utilizan el siguiente mensaje de error:

El proxy del almacén de claves externo denegó el acceso a la operación. Compruebe que el usuario y la clave externa estén autorizados para esta operación e intente realizar la solicitud de nuevo.

- Para resolver el error, utilice el administrador de claves externo o las herramientas de proxy del almacén de claves externo para determinar por qué falló la autorización. A continuación, actualice el procedimiento que provocó la solicitud no autorizada o utilice las herramientas de proxy del almacén de claves externo para actualizar la política de autorización. No puede resolver este error en AWS KMS.

Seguridad de AWS Key Management Service

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de un centro de datos y una arquitectura de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS programas](#) de de . Para obtener más información sobre los programas de cumplimiento aplicables a AWS Key Management Service (AWS KMS), consulte [AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad AWS Servicios incluidos](#) .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Además de su configuración y uso AWS KMS keys, usted es responsable de otros factores, como la confidencialidad de sus datos, los requisitos de su empresa y las leyes y reglamentos aplicables AWS KMS

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando lo utilice AWS Key Management Service. Le muestra cómo configurarlo para AWS KMS cumplir sus objetivos de seguridad y conformidad.

Temas

- [Protección de datos en AWS Key Management Service](#)
- [Gestión de identidad y acceso para AWS Key Management Service](#)
- [Inicio de sesión y supervisión AWS Key Management Service](#)
- [Validación de conformidad para AWS Key Management Service](#)
- [Resiliencia en AWS Key Management Service](#)
- [Seguridad de la infraestructura en AWS Key Management Service](#)

Protección de datos en AWS Key Management Service

AWS Key Management Service almacena y protege sus claves de cifrado para que estén altamente disponibles y, al mismo tiempo, le proporciona un control de acceso sólido y flexible.

Temas

- [Rotación del material de claves](#)
- [Cifrado de datos](#)
- [Privacidad del tráfico entre redes](#)

Rotación del material de claves

De forma predeterminada, AWS KMS genera y protege el material de claves criptográficas para las claves KMS. Además, AWS KMS ofrece opciones para el material clave que se crea y protege fuera de AWS KMS.

Protegiendo el material clave generado en AWS KMS

Al crear una clave KMS, de forma predeterminada, se AWS KMS genera y protege el material criptográfico de la clave KMS.

Para proteger el material clave de las claves KMS, AWS KMS utiliza una flota distribuida de módulos de seguridad de hardware [validados por el FIPS 140-3 de nivel de seguridad 3 \(\)](#). HSMs Cada AWS KMS HSM es un dispositivo de hardware dedicado e independiente diseñado para proporcionar funciones criptográficas específicas a fin de cumplir con los requisitos de seguridad y escalabilidad de. AWS KMS (Los HSMs que se AWS KMS utilizan en las regiones de China están certificados por la [OSCCA](#) y cumplen con todas las normas chinas pertinentes, pero no están validados según el Programa de validación de módulos criptográficos FIPS 140-3).

El material de claves de una clave de KMS se cifra de forma predeterminada cuando se genera en el HSM. El material de claves se descifra solo en la memoria volátil del HSM y solo durante los pocos milisegundos que se necesitan para usarlo en una operación criptográfica. Siempre que el material clave no esté en uso activo, se cifra en el HSM y se transfiere a un almacenamiento persistente de baja latencia y [alta durabilidad](#) (99,19%), donde permanece separado y aislado del. HSMs El material de claves de texto sin formato nunca sale de los [límites de seguridad](#) del HSM; nunca se escribe en un disco ni permanece en ningún medio de almacenamiento. (La única excepción es la clave pública de un par de claves asimétricas, que no es secreta).

AWS afirma como principio de seguridad fundamental que no hay interacción humana con el material clave criptográfico de texto simple, sea cual sea su tipo. Servicio de AWS No existe ningún mecanismo que permita a nadie, ni siquiera a Servicio de AWS los operadores, ver, acceder o exportar el material clave en texto plano. Este principio se aplica incluso durante fallos catastróficos y eventos de recuperación de desastres. El material clave de cliente en texto simple AWS KMS se utiliza para operaciones criptográficas dentro del marco del AWS KMS FIPS 140-3 y HSMs solo se valida en respuesta a las solicitudes autorizadas que el cliente o su delegado realicen al servicio.

En el caso de [las claves administradas por el cliente](#), el Cuenta de AWS que crea la clave es el propietario único e intransferible de la clave. La cuenta propietaria tiene el control total y exclusivo sobre las políticas de autorización que controlan el acceso a la clave. Pues Claves administradas por AWS, Cuenta de AWS tiene el control total sobre las políticas de IAM que autorizan las solicitudes al Servicio de AWS

Protegiendo el material clave generado fuera de AWS KMS

AWS KMS proporciona alternativas al material clave generado en AWS KMS.

Los [almacenes de claves personalizados](#), una AWS KMS función opcional, permiten crear claves de KMS respaldadas por material clave generado y utilizado fuera de ellos AWS KMS. Las claves de KMS de los [almacenes de AWS CloudHSM claves](#) están respaldadas por claves de los módulos de seguridad de AWS CloudHSM hardware que usted controla. HSMs Están certificadas con el nivel de [seguridad FIPS 140-2 de nivel 3 o el 140-3 de nivel de seguridad](#) 3. Las claves de KMS de los [almacenes de claves externos](#) están respaldadas por las claves de un administrador de claves externo que usted controla y administra desde fuera AWS, como un HSM físico en su centro de datos privado.

Otra característica opcional le permite [importar el material de claves](#) para obtener una clave KMS. Para proteger el material clave importado mientras está en tránsito AWS KMS, se cifra el material clave con una clave pública de un par de claves RSA generado en un AWS KMS HSM. El material clave importado se descifra en un AWS KMS HSM y se vuelve a cifrar con una clave simétrica en el HSM. Como todo el material AWS KMS clave, el material clave importado en texto plano nunca sale del archivo sin cifrar. HSMs Sin embargo, el cliente que proporcionó el material de claves es responsable del uso seguro, la durabilidad y el mantenimiento del material de claves fuera de AWS KMS.

Cifrado de datos

Los datos se AWS KMS componen del material clave de cifrado que representan AWS KMS keys y del material que representan. Este material clave solo existe en texto plano en los módulos de seguridad de AWS KMS hardware (HSMs) y solo cuando está en uso. De lo contrario, el material de la clave se cifra y se almacena en almacenamiento persistente duradero.

El material clave que se AWS KMS genera para las claves de KMS nunca AWS KMS HSMs deja de estar cifrado. No se exporta ni transmite en ninguna operación de AWS KMS API. La excepción son [las claves multirregionales](#), donde se AWS KMS utiliza un mecanismo de replicación entre regiones para copiar el material clave de una clave multirregional de un HSM Región de AWS a un HSM de otro. Región de AWS Para obtener más información, consulte el [proceso de replicación de claves multirregionales en Detalles criptográficos](#). AWS Key Management Service

Temas

- [Cifrado en reposo](#)
- [Cifrado en tránsito](#)

Cifrado en reposo

AWS KMS genera material clave para los módulos de seguridad AWS KMS keys de hardware compatibles con el [nivel de seguridad 3 de la norma FIPS 140-3](#) (). HSMs La única excepción son las regiones de China, donde las HSMs que se AWS KMS utilizan para generar las claves KMS cumplen con todas las regulaciones chinas pertinentes, pero no están validadas según el Programa de validación de módulos criptográficos FIPS 140-3. Cuando no se utiliza, el material de claves se cifra mediante una clave de HSM y se escribe en un almacenamiento duradero y persistente. El material clave de las claves KMS y las claves de cifrado que protegen el material clave nunca están HSMs en formato de texto plano.

El cifrado y la administración del material de claves para las claves KMS está completamente a cargo de AWS KMS.

Para obtener más información, consulte Cómo [trabajar con AWS KMS keys](#) detalles AWS Key Management Service criptográficos.

Cifrado en tránsito

El material clave que se AWS KMS genera para las claves de KMS nunca se exporta ni transmite en las operaciones de la AWS KMS API. AWS KMS utiliza [identificadores clave](#) para representar

las claves de KMS en las operaciones de la API. Del mismo modo, el material clave de las claves de KMS en los [almacenes de claves AWS KMS personalizados](#) no se puede exportar y nunca se transmite a las operaciones de API AWS KMS ni AWS CloudHSM a través de ellas.

Sin embargo, algunas operaciones de AWS KMS API devuelven claves de [datos](#). Además, los clientes pueden usar las operaciones de la API para [importar material de claves](#) para las claves KMS seleccionadas.

Todas las llamadas a la AWS KMS API deben firmarse y transmitirse mediante Transport Layer Security (TLS). AWS KMS requiere TLS 1.2 y recomienda TLS 1.3 en todas las regiones. AWS KMS también es compatible con el TLS poscuántico híbrido para los puntos finales AWS KMS de servicio en todas las regiones, excepto en las regiones de China. AWS KMS no admite el TLS poscuántico híbrido para los terminales FIPS en. AWS GovCloud (US) Las llamadas a AWS KMS también requieren un paquete de cifrado moderno que admita secreto perfecto en el futuro, lo que significa que el compromiso de cualquier secreto, como una clave privada, no comprometa también la clave de sesión.

Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para utilizar puntos de conexión estándar o AWS KMS puntos de conexión AWS KMS FIPS, los clientes deben admitir TLS 1.2 o una versión posterior. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-3](#). [Para obtener una lista de los puntos finales de AWS KMS FIPS, consulte AWS Key Management Service los puntos de enlace y las cuotas en](#). Referencia general de AWS

Las comunicaciones entre los hosts del AWS KMS servicio HSMs están protegidas mediante criptografía de curva elíptica (ECC) y el estándar de cifrado avanzado (AES) en un esquema de cifrado autenticado. Para obtener más información, consulte [Seguridad de las comunicaciones internas](#) en Detalles criptográficos. AWS Key Management Service

Privacidad del tráfico entre redes

AWS KMS admite un Consola de administración de AWS conjunto de operaciones de API que permiten crearlas, administrarlas AWS KMS keys y utilizarlas en operaciones criptográficas.

AWS KMS admite dos opciones de conectividad de red, desde su red privada hasta AWS.

- Una conexión IPSec VPN a través de Internet
- [AWS Direct Connect](#), que conecta la red interna a una Direct Connect ubicación a través de un cable Ethernet de fibra óptica estándar.

Todas las llamadas a la AWS KMS API deben estar firmadas y transmitirse mediante Transport Layer Security (TLS). Las llamadas también requieren un paquete de cifrado moderno que admita el [secreto perfecto en el futuro](#). El tráfico a los módulos de seguridad de hardware (HSMs) que almacenan el material clave para las claves de KMS solo está permitido desde hosts de AWS KMS API conocidos a través de la red AWS interna.

Para conectarse directamente AWS KMS desde su nube privada virtual (VPC) sin enviar tráfico a través de la Internet pública, utilice los puntos de conexión de la VPC, con la tecnología de [AWS PrivateLink](#). Para obtener más información, consulte [Conéctese a AWS KMS través de un punto final de VPC](#).

AWS KMS también es compatible con una opción [híbrida de intercambio de claves poscuántico](#) para el protocolo de cifrado de red Transport Layer Security (TLS). Puede usar esta opción con TLS cuando se conecte a AWS KMS los puntos finales de la API.

Gestión de identidad y acceso para AWS Key Management Service

AWS Identity and Access Management (IAM) le ayuda a controlar de forma segura el acceso a AWS los recursos. Los administradores controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar AWS KMS los recursos. Para obtener más información, consulte [Uso de políticas de IAM con AWS KMS](#).

Las [políticas clave](#) son el mecanismo principal para controlar el acceso a las claves de KMS. AWS KMS Cada clave de KMS debe tener una política de claves. También puede utilizar [políticas de IAM](#) y [concesiones](#), junto con las políticas de claves para controlar el acceso a sus claves KMS. Para obtener más información, consulte [Permisos y acceso a claves KMS](#).

Si utiliza una Amazon Virtual Private Cloud (Amazon VPC), puede [crear una interfaz de punto final de VPC](#) con tecnología. AWS KMS [AWS PrivateLink](#). También puedes usar las políticas de punto final de la VPC para determinar qué entidades principales pueden acceder a tu AWS KMS punto final, qué llamadas a la API pueden realizar y a qué clave de KMS pueden acceder.

Temas

- [AWS políticas gestionadas para AWS Key Management Service](#)
- [Uso de roles vinculados a servicios para AWS KMS](#)

AWS políticas gestionadas para AWS Key Management Service

Una política AWS administrada es una política independiente creada y administrada por AWS. Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir [políticas administradas por el cliente](#) específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte [Políticas administradas por AWS](#) en la Guía del usuario de IAM.

AWS política gestionada: AWSKey ManagementServicePowerUser

Puede asociar la política AWSKeyManagementServicePowerUser a las identidades de IAM.

Puede utilizar la política administrada de AWSKeyManagementServicePowerUser para dar a las entidades principales de IAM en su cuenta los permisos de un usuario avanzado. Los usuarios avanzados pueden crear claves KMS, usar y administrar las claves KMS que crean y ver todas las claves KMS e identidades de IAM. Las entidades principales que tienen la política administrada AWSKeyManagementServicePowerUser también pueden obtener permisos de otras fuentes, incluidas las políticas de claves, otras políticas de IAM y las concesiones.

AWSKeyManagementServicePowerUser es una política de IAM AWS gestionada. Para obtener más información sobre las políticas AWS gestionadas, consulte las [políticas AWS gestionadas](#) en la Guía del usuario de IAM.

Note

Los permisos de esta política que son específicos para una clave KMS, como `kms:TagResource` y `kms:GetKeyRotationStatus`, solo son efectivos cuando la política

de la clave para esa clave KMS [permite explícitamente que la Cuenta de AWS utilice las políticas de IAM](#) para controlar el acceso a la clave. Para determinar si un permiso es específico para una clave KMS, consulte [AWS KMS permisos](#) y busque un valor de clave KMS en la columna Resources (Recursos).

Esta política proporciona al usuario avanzado permisos sobre cualquier clave KMS con una política de claves que permita la operación. En el caso de los permisos entre cuentas, como `kms:DescribeKey` y `kms:ListGrants`, esto podría incluir las claves KMS en Cuentas de AWS no confiables. Para más detalles, consulte [Prácticas recomendadas para las políticas de IAM](#) y [Permitir a los usuarios de otras cuentas utilizar una clave KMS](#). Para determinar si un permiso es válido en las claves KMS de otras cuentas, consulte [AWS KMS permisos](#) y busque un valor Yes (Sí) en la columna Cross-account use (Uso entre cuentas).

Para que los directores puedan ver la AWS KMS consola sin errores, el director necesita la [etiqueta: GetResources permission](#), que no está incluida en la `AWSKeyManagementServicePowerUser` política. Puede autorizar este permiso en una política de IAM independiente.

La política de IAM administrada [AWSKeyManagementServicePowerUser](#) debe incluir los siguientes permisos.

- Permite a las entidades principales crear claves KMS. Dado que este proceso incluye la configuración de la política de claves, los usuarios avanzados pueden concederse a sí mismos y a otros permisos para usar y administrar las claves KMS que crean.
- Permite a las entidades principales crear y eliminar [alias](#) y [etiquetas](#) en todas las claves KMS. Si cambia una etiqueta o un alias, puede permitir o denegar el permiso para usar y administrar la clave KMS. Para obtener más información, consulte [ABAC para AWS KMS](#).
- Permite a las entidades principales obtener información detallada sobre todas las claves KMS, incluyendo su ARN clave, configuración criptográfica, política de claves, alias, etiquetas y [estado de rotación](#).
- Permite a las entidades principales enumerar usuarios, grupos y roles de IAM.
- Esta política no permite a las entidades principales utilizar o administrar claves KMS que no hayan creado. Sin embargo, pueden cambiar los alias y las etiquetas de todas las claves KMS, lo que podría permitirles o denegarles el permiso para utilizar o administrar una clave KMS.

Para ver los permisos de esta política, consulte la Referencia [AWSKeyManagementServicePowerUser](#) de políticas AWS administradas.

AWS política gestionada: AWSServiceRoleForKeyManagementServiceCustomKeyStores

No puede asociar `AWSServiceRoleForKeyManagementServiceCustomKeyStores` a sus entidades IAM. Esta política está asociada a un rol vinculado a un servicio que permite AWS KMS ver los AWS CloudHSM clústeres asociados a su almacén de AWS CloudHSM claves y crear la red que permita una conexión entre su almacén de claves personalizado y su AWS CloudHSM clúster. Para obtener más información, consulte [Autorización AWS KMS para gestionar los recursos AWS CloudHSM de Amazon EC2](#).

AWS política gestionada: AWSServiceRoleForKeyManagementServiceMultiRegionKeys

No puede asociar `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` a sus entidades IAM. Esta política se adjunta a un rol vinculado a un servicio que otorga permiso de AWS KMS para sincronizar los cambios realizados en el material de claves de una clave principal de varias regiones con las claves de réplica. Para obtener más información, consulte [Autoriza la sincronización de claves AWS KMS multirregionales](#).

AWS KMS actualizaciones de las políticas AWS gestionadas

Consulte los detalles sobre las actualizaciones de las políticas AWS administradas AWS KMS desde que este servicio comenzó a rastrear estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de AWS KMS [Historial de documentos](#).

Cambio	Descripción	Fecha
AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy : actualización de una política actual	AWS KMS se agregó un campo de ID de declaración (Sid) a la política administrada en la versión v2 de la política.	21 de noviembre de 2024
AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy : actualización de una política actual	AWS KMS agregó los <code>ec2:DescribeNetworkInterfaces</code>	10 de noviembre de 2023

Cambio	Descripción	Fecha
	permisos <code>ec2:DescribeVpcs</code> <code>ec2:DescribeNetworkAcls</code> , y para monitorear los cambios en la VPC que contiene el AWS CloudHSM clúster, de modo que AWS KMS puedan proporcionar mensajes de error claros en caso de fallas.	
AWS KMS comenzó a rastrear los cambios	AWS KMS comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	10 de noviembre de 2023

Uso de roles vinculados a servicios para AWS KMS

AWS Key Management Service [usa roles vinculados al AWS Identity and Access Management servicio \(IAM\)](#). Un rol vinculado a un servicio es un tipo único de rol de IAM al que se vincula directamente. AWS KMS Los roles vinculados al servicio se definen AWS KMS e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

Un rol vinculado a un servicio facilita la configuración AWS KMS , ya que no es necesario añadir manualmente los permisos necesarios. AWS KMS define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo AWS KMS puede asumir sus funciones. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda asociar a ninguna otra entidad de IAM.

Solo puede eliminar un rol vinculado a servicios después de eliminar los recursos relacionados. Esto protege sus AWS KMS recursos porque no puede eliminar inadvertidamente el permiso de acceso a los recursos.

Para obtener información acerca de otros servicios que admiten roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#) y busque los servicios que muestran Sí en la columna Rol vinculado a un servicio. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Para consultar los detalles sobre las actualizaciones de las funciones vinculadas al servicio, consulte [AWS KMS actualizaciones de las políticas AWS gestionadas](#).

Temas

- [Autorización AWS KMS para gestionar los recursos AWS CloudHSM de Amazon EC2](#)
- [Autoriza la sincronización de claves AWS KMS multirregionales](#)

Autorización AWS KMS para gestionar los recursos AWS CloudHSM de Amazon EC2

Para respaldar sus almacenes de AWS CloudHSM claves, AWS KMS necesita permiso para obtener información sobre sus AWS CloudHSM clústeres. También necesita permiso para crear la infraestructura de red que conecta el almacén de AWS CloudHSM claves con su AWS CloudHSM clúster. Para obtener estos permisos, AWS KMS crea el rol `AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculado al servicio en su Cuenta de AWS. Los usuarios que crean almacenes de AWS CloudHSM claves deben tener el `iam:CreateServiceLinkedRole` permiso que les permita crear roles vinculados a servicios.

Para ver los detalles sobre las actualizaciones de la política `AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy` gestionada, consulte. [AWS KMS actualizaciones de las políticas AWS gestionadas](#)

Temas

- [Acerca de la función AWS KMS vinculada al servicio](#)
- [Creación del rol vinculado a servicios](#)
- [Editar la descripción del rol vinculado a un servicio](#)
- [Eliminar el rol vinculado a servicios](#)

Acerca de la función AWS KMS vinculada al servicio

Un [rol vinculado a un servicio es un rol](#) de IAM que da permiso a un AWS servicio para llamar a otros AWS servicios en tu nombre. Está diseñado para facilitar el uso de las funciones de varios AWS servicios integrados sin tener que crear y mantener políticas de IAM complejas. Para obtener más información, consulte [Uso de roles vinculados a servicios para AWS KMS](#).

En el AWS CloudHSM caso de los almacenes clave, AWS KMS crea la función `AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculada al servicio con la

AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy política gestionada. Esta política concede los siguientes permisos al rol:

- [CloudHSM:Describe*](#): detecta los cambios en el AWS CloudHSM clúster adjunto a su almacén de claves personalizado.
- [ec2: CreateSecurityGroup](#) — se utiliza cuando se [conecta un almacén de AWS CloudHSM claves](#) para crear el grupo de seguridad que permite el flujo de tráfico de red entre el clúster y el clúster. AWS KMS AWS CloudHSM
- [ec2: AuthorizeSecurityGroupIngress](#) — se utiliza cuando se [conecta un almacén de AWS CloudHSM claves](#) para permitir el acceso a la red desde AWS KMS la VPC que contiene AWS CloudHSM el clúster.
- [ec2: CreateNetworkInterface](#) — se utiliza cuando se [conecta un almacén de AWS CloudHSM claves](#) para crear la interfaz de red que se utiliza para la comunicación entre el clúster AWS KMS y el clúster. AWS CloudHSM
- [ec2: RevokeSecurityGroupEgress](#) — se utiliza cuando se [conecta un almacén de AWS CloudHSM claves](#) para eliminar todas las reglas de salida del grupo de seguridad que lo creó. AWS KMS
- [ec2: DeleteSecurityGroup](#) — se utiliza cuando se [desconecta un almacén de AWS CloudHSM claves](#) para eliminar los grupos de seguridad que se crearon al conectar el AWS CloudHSM almacén de claves.
- [ec2: DescribeSecurityGroups](#) — se usa para monitorear los cambios en el grupo de seguridad que se AWS KMS creó en la VPC que contiene AWS CloudHSM el clúster, de modo AWS KMS que pueda proporcionar mensajes de error claros en caso de fallas.
- [ec2: DescribeVpcs](#) — se usa para monitorear los cambios en la VPC que contiene AWS CloudHSM el clúster, de modo AWS KMS que pueda proporcionar mensajes de error claros en caso de fallas.
- [ec2: DescribeNetworkAcls](#) — se usa para monitorear los cambios en la red ACLs de la VPC que contiene AWS CloudHSM el clúster, de modo AWS KMS que pueda proporcionar mensajes de error claros en caso de fallas.
- [ec2: DescribeNetworkInterfaces](#) — se utiliza para supervisar los cambios en las interfaces de red que se AWS KMS crearon en la VPC que contiene AWS CloudHSM el clúster, de modo AWS KMS que puedan proporcionar mensajes de error claros en caso de que se produzcan errores.

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 el rol `AWSServiceRoleForKeyManagementServiceCustomKeyStores` vinculado al servicio solo es de confianza `zacks.kms.amazonaws.com`, solo AWS KMS puede asumir este rol vinculado al servicio. Esta función se limita a las operaciones necesarias para AWS KMS ver los AWS CloudHSM clústeres y conectar un almacén de AWS CloudHSM claves al clúster asociado. AWS CloudHSM No otorga AWS KMS ningún permiso adicional. Por ejemplo, AWS KMS no tiene permiso para crear, administrar o eliminar sus AWS CloudHSM clústeres o copias de seguridad. HSMs

Regiones

Al igual que la función de almacenes de AWS CloudHSM claves, la `AWSServiceRoleForKeyManagementServiceCustomKeyStores` función es compatible en todas Regiones de AWS partes AWS KMS y AWS CloudHSM está disponible. Para obtener una lista de los elementos Regiones de AWS que admite cada servicio, consulte [AWS Key Management Service Puntos finales y cuotas](#) y [AWS CloudHSM puntos finales y cuotas](#) en. Referencia general de Amazon Web Services

Para obtener más información sobre cómo AWS los servicios utilizan las funciones vinculadas a servicios, consulte [Uso de funciones vinculadas a servicios en la Guía del usuario de IAM](#).

Creación del rol vinculado a servicios

AWS KMS crea automáticamente el rol

AWSServiceRoleForKeyManagementServiceCustomKeyStores vinculado al servicio en su Cuenta de AWS al crear un almacén de AWS CloudHSM claves, si el rol aún no existe. No puede crear o volver a crear este rol vinculado a un servicio directamente.

Editar la descripción del rol vinculado a un servicio

No puede editar el nombre del rol o las instrucciones de la política en el rol vinculado a un servicio AWSServiceRoleForKeyManagementServiceCustomKeyStores, aunque sí puede editar la descripción del rol. Para obtener más información, consulte [Editar un rol vinculado a un servicio](#) en la Guía del usuario de IAM.

Eliminar el rol vinculado a servicios

AWS KMS no elimina el rol AWSServiceRoleForKeyManagementServiceCustomKeyStores vinculado al servicio de su cuenta, Cuenta de AWS incluso si ha [eliminado todos sus almacenes de claves](#). AWS CloudHSM Aunque actualmente no existe ningún procedimiento para eliminar la función AWSServiceRoleForKeyManagementServiceCustomKeyStores vinculada al servicio, AWS KMS no la asume ni utiliza sus permisos a menos que tenga almacenes de claves activos. AWS CloudHSM

Autoriza la sincronización de claves AWS KMS multirregionales

Para admitir [claves multirregionales](#), AWS KMS necesita permiso para sincronizar las [propiedades compartidas](#) de una clave principal multirregional con sus claves de réplica. Para obtener estos permisos, AWS KMS crea el rol vinculado al AWSServiceRoleForKeyManagementServiceMultiRegionKeys servicio en su Cuenta de AWS. Los usuarios que crean claves de varias regiones deben tener el permiso `iam:CreateServiceLinkedRole` que les permite crear roles vinculados a un servicio.

Puede ver el [SynchronizeMultiRegionKey](#) CloudTrail evento que registra la AWS KMS sincronización de propiedades compartidas en sus registros. AWS CloudTrail

Para ver los detalles sobre las actualizaciones de la política

AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy gestionada, consulte [AWS KMS actualizaciones de las políticas AWS gestionadas](#).

Temas

- [Acerca del rol vinculado a un servicio para claves de varias regiones](#)
- [Creación del rol vinculado a servicios](#)
- [Editar la descripción del rol vinculado a un servicio](#)
- [Eliminar el rol vinculado a servicios](#)

Acerca del rol vinculado a un servicio para claves de varias regiones

Una [función vinculada a un servicio es una función](#) de IAM que permite a un AWS servicio llamar a otros AWS servicios en su nombre. Está diseñado para facilitar el uso de las funciones de varios AWS servicios integrados sin tener que crear y mantener políticas de IAM complejas.

En el caso de las claves multirregionales, AWS KMS crea el rol `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` vinculado al servicio con la política gestionada `AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy`. Esta política le confiere al rol el permiso `kms:SynchronizeMultiRegionKey`, que le permite sincronizar las propiedades compartidas de claves de varias regiones.

Como el rol `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` vinculado al servicio solo es de confianza `arn:kms:amazonaws.com`, solo AWS KMS puede asumir este rol vinculado al servicio. Esta función se limita a las operaciones que se AWS KMS necesitan para sincronizar las propiedades compartidas de varias regiones. No otorga AWS KMS ningún permiso adicional. Por ejemplo, AWS KMS no tiene permiso para crear, replicar ni eliminar ninguna clave de KMS.

Para obtener más información sobre cómo AWS los servicios utilizan las funciones vinculadas a servicios, consulte [Uso de funciones vinculadas a servicios en la Guía del usuario](#) de IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Sid" : "KMSSynchronizeMultiRegionKey",
      "Effect" : "Allow",
      "Action" : [
        "kms:SynchronizeMultiRegionKey"
      ],
    }
  ]
}
```

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

Creación del rol vinculado a servicios

AWS KMS crea automáticamente el rol

AWSServiceRoleForKeyManagementServiceMultiRegionKeysvinculado al servicio en usted Cuenta de AWS al crear una clave multirregional, si el rol aún no existe. No puede crear o volver a crear este rol vinculado a un servicio directamente.

Editar la descripción del rol vinculado a un servicio

No puede editar el nombre del rol ni las declaraciones de política del rol

AWSServiceRoleForKeyManagementServiceMultiRegionKeysvinculado al servicio, pero sí puede editar la descripción del rol. Para obtener más información, consulte [Editar un rol vinculado a un servicio](#) en la Guía del usuario de IAM.

Eliminar el rol vinculado a servicios

AWS KMS no elimina el rol AWSServiceRoleForKeyManagementServiceMultiRegionKeysvinculado al servicio de su cuenta Cuenta de AWS y usted no puede eliminarlo. Sin embargo, AWS KMS no asume el AWSServiceRoleForKeyManagementServiceMultiRegionKeysrol ni usa ninguno de sus permisos a menos que tenga claves multirregionales en su Cuenta de AWS región.

Inicio de sesión y supervisión AWS Key Management Service

La monitorización es una parte importante de la comprensión de la disponibilidad, el estado y el uso de su AWS KMS keys en AWS KMS. La supervisión ayuda a mantener la seguridad, la confiabilidad, la disponibilidad y el rendimiento de sus AWS soluciones. AWS proporciona varias herramientas para supervisar las claves de KMS.

AWS CloudTrail Registros

Cada llamada a una operación de AWS KMS API se captura como un evento en un AWS CloudTrail registro. Estos registros registran todas las llamadas a la API desde la AWS KMS consola, así como las llamadas realizadas por AWS KMS otros AWS servicios. Las llamadas a la

API entre cuentas, como las llamadas para usar una clave de KMS en otra Cuenta de AWS, se registran en los CloudTrail registros de ambas cuentas.

Al solucionar problemas o auditar, puede utilizar el registro para reconstruir el ciclo de vida de una clave KMS. También puede ver su administración y uso de la clave KMS en operaciones criptográficas. Para obtener más información, consulte [the section called “Iniciar sesión con AWS CloudTrail”](#).

Amazon CloudWatch Logs

Supervise, almacene y acceda a sus archivos de registro desde AWS CloudTrail y otras fuentes. Para obtener más información, consulta la [Guía del CloudWatch usuario de Amazon](#).

Pues AWS KMS, CloudWatch almacena información útil que le ayuda a evitar problemas con sus claves de KMS y los recursos que protegen. Para obtener más información, consulte [the section called “Supervise las claves con CloudWatch”](#).

Amazon EventBridge

AWS KMS genera EventBridge eventos cuando se [rota](#) o se [elimina](#) su clave de KMS o cuando caduca el [material de clave importado](#) de su clave de KMS. Busque AWS KMS eventos (operaciones de API) y diríjalos a una o más funciones o flujos de destino para capturar información de estado. Para obtener más información, consulta [the section called “Monitorea las teclas con Amazon EventBridge”](#) la [Guía del EventBridge usuario de Amazon](#).

CloudWatch Métricas de Amazon

Puede supervisar sus claves de KMS mediante CloudWatch métricas, que recopilan y procesan datos sin procesar para AWS KMS convertirlos en métricas de rendimiento. Los datos se registran en intervalos de dos semanas para que pueda ver las tendencias de la información actual e histórica. Esto le ayuda a comprender cómo se utilizan sus claves KMS y cómo su uso cambia con el tiempo. Para obtener información sobre el uso de CloudWatch métricas para supervisar las claves de KMS, consulte [AWS KMS métricas y dimensiones](#).

CloudWatch Alarmas Amazon

Vea solo un cambio de alarma durante el periodo especificado. Luego, realice una o varias acciones según el valor de la métrica con respecto a un umbral durante varios períodos. Por ejemplo, puede crear una CloudWatch alarma que se active cuando alguien intente usar una clave de KMS que está programada para eliminarse en una operación criptográfica. Esto indica que la clave KMS todavía se está utilizando y probablemente no debería eliminarse. Para obtener más información, consulte [the section called “Crear una alarma”](#).

AWS Security Hub CSPM

Puede supervisar su AWS KMS uso para comprobar si cumple con AWS Security Hub CSPM los estándares del sector de la seguridad y las mejores prácticas. Security Hub CSPM utiliza controles de seguridad para evaluar las configuraciones de los recursos y los estándares de seguridad para ayudarlo a cumplir con varios marcos de conformidad. Para obtener más información, consulte [AWS Key Management Service controls](#) en la Guía del usuario de AWS Security Hub .

Validación de conformidad para AWS Key Management Service

Los auditores externos evalúan la seguridad y el cumplimiento AWS Key Management Service como parte de varios programas de AWS cumplimiento. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Temas

- [Documentos de conformidad y seguridad](#)
- [Más información](#)

Documentos de conformidad y seguridad

Los siguientes documentos de cumplimiento y seguridad cubren AWS KMS. Para verlos, utilice [AWS Artifact](#).

- Catálogo de controles de conformidad de informática en la nube (C5)
- Declaración de aplicabilidad (SoA) de ISO 27001:2013
- Certificación ISO 27001:2013
- Declaración de aplicabilidad (SoA) de ISO 27017:2015
- Certificación ISO 27017:2015
- Declaración de aplicabilidad (SoA) de ISO 27018:2015
- Certificación ISO 27018:2014
- Certificación ISO 9001:2015
- Declaración de conformidad (AOC) PCI DSS y resumen de responsabilidad
- Informe de controles de organizaciones de servicios (SOC) 1
- Informe de controles de organizaciones de servicios (SOC) 2

- Informe de controles de organizaciones de servicios (SOC) 2 para la confidencialidad
- FedRAMP-High

Para obtener ayuda al AWS Artifact respecto, consulta [Descarga de informes en AWS Artifact](#).

Más información

Su responsabilidad de cumplimiento al AWS KMS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. Si su uso de AWS KMS está sujeto al cumplimiento de una norma publicada, AWS proporciona recursos que le ayudarán a:

- [AWS Servicios incluidos en el ámbito de aplicación del programa de conformidad](#): en esta página se enumeran AWS los servicios que se encuentran dentro del ámbito de aplicación de programas de conformidad específicos. Para obtener información general, consulte [Programas de conformidad de AWS](#).
- [Guías de inicio rápido sobre seguridad y cumplimiento](#): estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en la seguridad y el cumplimiento. AWS
- [AWS Recursos de conformidad](#): esta colección de libros de trabajo y guías puede aplicarse a su sector y ubicación.
- [AWS Config](#)— Este AWS servicio evalúa en qué medida las configuraciones de sus recursos cumplen con las prácticas internas, las directrices del sector y las normativas.
- [AWS Security Hub CSPM](#)— Este AWS servicio proporciona una visión completa del estado de su seguridad interior AWS. Security Hub CSPM utiliza controles de seguridad para evaluar sus AWS recursos y comprobar su conformidad con los estándares y las mejores prácticas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la [Referencia de controles de Security Hub CSPM](#).

Resiliencia en AWS Key Management Service

La infraestructura AWS global se basa en zonas Regiones de AWS de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad

tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Además de la infraestructura AWS global, AWS KMS ofrece varias funciones para ayudarlo a satisfacer sus necesidades de respaldo y resiliencia de datos. Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte [Infraestructura AWS global](#).

Aislamiento regional

AWS Key Management Service (AWS KMS) es un servicio regional autosuficiente que está disponible en todas las Regiones de AWS. El diseño aislado regionalmente AWS KMS garantiza que un problema de disponibilidad en una región Región de AWS no pueda afectar al AWS KMS funcionamiento en ninguna otra región. AWS KMS está diseñado para garantizar que no haya ningún tiempo de inactividad planificado, ya que todas las actualizaciones de software y las operaciones de escalado se realizan de forma fluida e imperceptible.

El [acuerdo AWS KMS de nivel de servicio](#) (SLA) incluye un compromiso de servicio del 99,999% para todos los KMS. APIs Para cumplir este compromiso, AWS KMS garantiza que todos los datos y la información de autorización necesarios para ejecutar una solicitud de la API estén disponibles en todos los hosts regionales que reciben la solicitud.

La AWS KMS infraestructura se replica en al menos tres zonas de disponibilidad (AZs) en cada región. Para garantizar que los fallos en varios hosts no afecten al AWS KMS rendimiento, AWS KMS está diseñado para atender el tráfico de clientes desde cualquier parte AZs de una región.

Los cambios realizados en las propiedades o permisos de una clave de KMS se replican en todos los hosts de la región para garantizar que cualquier host de la región pueda procesar de manera correcta la solicitud posterior. Las solicitudes de [operaciones criptográficas](#) que utilizan la clave KMS se reenvían a una flota de módulos de seguridad de AWS KMS hardware (HSMs), cualquiera de los cuales puede realizar la operación con la clave KMS.

Diseño de varios inquilinos

El diseño multiusuario AWS KMS le permite cumplir con el SLA de disponibilidad del 99,999% y mantener altas tasas de solicitudes, al tiempo que protege la confidencialidad de sus claves y datos.

Se implementan varios mecanismos de cumplimiento de la integridad para garantizar que la clave de KMS especificada para la operación criptográfica sea siempre la que se utiliza.

El material de clave de texto sin formato para las claves de KMS está ampliamente protegido. El material de clave se cifra en el HSM tan pronto como se crea y el material de clave cifrado se mueve de inmediato al almacenamiento seguro y de baja latencia. La clave cifrada se recupera y se descifra dentro del HSM justo a tiempo para su uso. La clave de texto sin formato permanece en la memoria HSM solo durante el tiempo necesario para completar la operación criptográfica. Luego se vuelve a cifrar en el HSM y la clave cifrada se devuelve al almacenamiento. El material clave en texto simple nunca sale del archivo HSMs; nunca se escribe en un almacenamiento persistente.

Mejores prácticas de resiliencia en AWS KMS

Para optimizar la resiliencia de sus AWS KMS recursos, considere las siguientes estrategias.

- Para respaldar la estrategia de copia de seguridad y recuperación de desastres, considere las claves de varias regiones, que son claves de KMS creadas en una Región de AWS y se replican solo en las regiones que especifique. Con las claves multirregionales, puede mover los recursos cifrados Regiones de AWS (dentro de la misma partición) sin exponer nunca el texto sin formato, y descifrar el recurso, cuando sea necesario, en cualquiera de sus regiones de destino. Las claves de varias regiones relacionadas son interoperables porque comparten el mismo material de clave y el mismo ID de clave, pero tienen políticas de clave independientes para el control de acceso de alta resolución. Para obtener más información, consulte [Claves de varias regiones en AWS KMS](#).
- [Para proteger sus claves en un servicio multiusuario, por ejemplo AWS KMS, asegúrese de utilizar los controles de acceso, incluidas las políticas clave y las políticas de IAM](#). Además, puede enviar sus solicitudes a AWS KMS un punto final de interfaz de VPC con tecnología AWS PrivateLink. Cuando lo haga, toda la comunicación entre su VPC de Amazon y AWS KMS se realizará íntegramente dentro de la AWS red mediante un AWS KMS punto de conexión dedicado restringido a su VPC. Puede proteger aún más estas solicitudes al crear una capa de autorización adicional mediante [políticas de punto de conexión de VPC](#). Para obtener más información, consulte [Conexión a AWS KMS mediante un punto de conexión de VPC](#).

Seguridad de la infraestructura en AWS Key Management Service

Como servicio gestionado, AWS Key Management Service (AWS KMS) está protegido por los procedimientos de seguridad de la red AWS global que se describen en [Amazon Web Services: Descripción general de los procesos de seguridad](#).

Para acceder a AWS KMS través de la red, puede llamar a las operaciones de la AWS KMS API que se describen en la [referencia de la AWS Key Management Service API](#). AWS KMS requiere

TLS 1.2 y recomienda TLS 1.3 en todas las regiones. AWS KMS también es compatible con el TLS poscuántico híbrido para los puntos finales AWS KMS de servicio en todas las regiones, excepto en las regiones de China. AWS KMS no admite el TLS poscuántico híbrido para los puntos finales FIPS en. AWS GovCloud (US) Para utilizar los [puntos de conexión estándar de AWS KMS](#) o los [puntos de conexión FIPS de AWS KMS](#), los clientes deben admitir TLS 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de los sistemas modernos, como Java 7 y posteriores, son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar [AWS Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Puede llamar a estas operaciones de API desde cualquier ubicación de red, pero AWS KMS admite condiciones de política global que le permiten controlar el acceso a una clave de KMS en función de la dirección IP de origen, la VPC y el punto final de la VPC. Puede utilizar estas claves de condición en políticas de claves y en políticas de IAM. Sin embargo, estas condiciones pueden AWS impedir que utilices la clave de KMS en tu nombre. Para obtener más información, consulte [AWS claves de condición globales](#).

Por ejemplo, la siguiente declaración de política clave permite a los usuarios que pueden asumir la `KMSTestRole` función utilizarla AWS KMS key para las [operaciones criptográficas](#) especificadas, a menos que la dirección IP de origen sea una de las direcciones IP especificadas en la 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"
    ]
  }
}
```

Aislamiento de Hosts físicos

La seguridad de la infraestructura física que se AWS KMS utiliza está sujeta a los controles descritos en la sección Seguridad física y ambiental de [Amazon Web Services: Descripción general de los procesos de seguridad](#). Puede encontrar más detalles en los informes de conformidad y los resultados de auditoría de terceros enumerados en la sección anterior.

AWS KMS está respaldado por módulos de seguridad de hardware reforzados dedicados (HSMs) diseñados con controles específicos para resistir los ataques físicos. HSMs Se trata de dispositivos físicos que no tienen una capa de virtualización, como un hipervisor, que comparte el dispositivo físico entre varios elementos lógicos. El material clave AWS KMS keys se almacena únicamente en la memoria volátil del HSMs KMS y solo mientras esté en uso. Esta memoria se borra cuando el HSM sale del estado operativo, incluidos los cierres y reinicios previstos e imprevistos. Para obtener información detallada sobre el funcionamiento de AWS KMS HSMs, consulte [Detalles AWS Key Management Service criptográficos](#).

Cuotas

Para que todos los usuarios puedan AWS KMS responder y ofrecer un rendimiento óptimo, AWS KMS aplica dos tipos de cuotas: las cuotas de recursos y las cuotas de solicitud. Cada cuota se calcula de forma independiente para cada región de cada Cuenta de AWS.

Todas AWS KMS las cuotas son ajustables, excepto la cuota de [recursos de rotación bajo demanda y la cuota](#) de [solicitudes del almacén de AWS CloudHSM claves](#). Para solicitar un aumento de cuota, consulte [Solicitud de un aumento de cuota](#) en la Guía de usuario de Service Quotas. Para solicitar una reducción de cuota, cambiar una cuota que no aparece en Service Quotas o cambiar una cuota en una en la Región de AWS que Service Quotas for no AWS KMS esté disponible, visite [AWS Support Center](#) y cree un caso.

Temas

- [Cuotas de recursos](#)
- [Cuotas de solicitudes](#)
- [Limitar las solicitudes AWS KMS](#)

Cuotas de recursos

AWS KMS establece cuotas de recursos para garantizar que pueda brindar un servicio rápido y resiliente a todos nuestros clientes. Algunas cuotas de recursos se aplican solo a los recursos que usted crea, pero no a los recursos que AWS los servicios crean para usted. Los recursos que usa pero que no están en la cuenta de Cuenta de AWS, como las [Claves propiedad de AWS](#), no se contabilizan en estas cuotas.

Si ha superado un límite de recursos, las solicitudes para crear un recurso adicional de dicho tipo generan un mensaje de error `LimitExceededException`.

Todas las cuotas de AWS KMS recursos son ajustables, excepto la [cuota de recursos de rotación bajo demanda](#). Para solicitar un aumento de cuota, consulte [Solicitud de un aumento de cuota](#) en la Guía de usuario de Service Quotas. Para solicitar una reducción de cuota, cambiar una cuota que no aparece en Service Quotas o cambiar una cuota en una en la Región de AWS que Service Quotas for no AWS KMS esté disponible, visite [AWS Support Center](#) y cree un caso.

En la siguiente tabla se enumeran y describen las cuotas de AWS KMS recursos de cada Cuenta de AWS región.

Nombre de la cuota	Predeterminado	Aplica a	Ajustable
AWS KMS keys	100 000	Claves administradas por el cliente	Sí
Alias por clave KMS	50	Alias creados por el cliente	Sí
Concesiones por clave KMS	50 000	Claves administradas por el cliente	Sí
Cuota de recursos de almacenes de claves personalizados	10	Cuenta de AWS y región	Sí
Rotación bajo demanda	25	Claves administradas por el cliente	No

Además de las cuotas de recursos, AWS KMS utiliza las cuotas de solicitud para garantizar la capacidad de respuesta del servicio. Para obtener más información, consulte [the section called “Cuotas de solicitudes”](#).

AWS KMS keys: 100 000

Puede tener hasta 100 000 [claves administradas por el cliente](#) en cada región de su Cuenta de AWS. Esta cuota se aplica a todas las claves administradas por el cliente en todas las Regiones de AWS, independientemente de su [especificación](#) o de su [estado](#) de clave. Cada clave de KMS se considera un recurso. Las [Claves administradas por AWS](#) y [Claves propiedad de AWS](#) no se contabilizan para esta cuota.

Alias por clave KMS: 50

Puede asociar hasta 50 [alias](#) con cada [clave administrada por el cliente](#). Los alias a los que se AWS asocian [Claves administradas por AWS](#) no se tienen en cuenta para esta cuota. Es posible que encuentre esta cuota cuando [cree](#) o [actualice](#) un alias.

 Note

La ResourceAliases condición [kms:](#) solo entra en vigor cuando la clave KMS cumple con esta cuota. Si una clave KMS supera esta cuota, las entidades principales que están autorizadas a usar la clave KMS mediante la condición kms:ResourceAliases se deniega el acceso a la clave KMS. Para obtener más información, consulte [Acceso denegado debido a la cuota de alias](#).

La cuota de alias por clave de KMS reemplaza a la cuota de alias por región que limitaba el número total de alias en cada región de una. Cuenta de AWS AWS KMS ha eliminado la cuota de alias por región.

Concesiones por clave KMS: 50 000

Cada [clave administrada por el cliente](#) puede tener hasta 50 000 [concesiones](#), incluidas las concesiones creadas por los [servicios de AWS que están integrados con AWS KMS](#). Esta cuota no se aplica a [Claves administradas por AWS](#) o [Claves propiedad de AWS](#).

Un efecto de esta cuota es que no puede realizar más de 50 000 operaciones con concesiones autorizadas que utilicen la misma clave KMS al mismo tiempo. Después de alcanzar la cuota, puede crear nuevas concesiones para la clave KMS solo cuando se retire o se revoque una concesión activa.

Por ejemplo, cuando adjunta un volumen de Amazon Elastic Block Store (Amazon EBS) a una instancia de Amazon Elastic Compute Cloud (Amazon EC2), el volumen se descifra para que pueda leerlo. Para obtener permiso para descifrar los datos, Amazon EBS crea una concesión para cada volumen. Por lo tanto, si todos los volúmenes de Amazon EBS utilizan la misma clave KMS, no puede adjuntar más de 50 000 volúmenes a la vez.

Cuota de recursos de almacenes de claves personalizados: 10

Puede crear hasta 10 [almacenes de claves personalizados](#) en cada Cuenta de AWS región. Si intenta crear más, se produce un error en la [CreateCustomKeyStore](#) operación.

Esta cuota se aplica al número total de almacenes de claves personalizados en cada cuenta y región. Esto incluye todos los [almacenes de claves de AWS CloudHSM](#) y los [almacenes de claves externos](#), independientemente del estado de su conexión.

Rotación bajo demanda: 25

Puede realizar la [rotación de claves bajo demanda](#) un máximo de 25 veces por clave KMS. Si intenta realizar más rotaciones bajo demanda, se produce un error en la [RotateKeyOnDemand](#) operación.

Esta cuota no se puede ajustar. No puede aumentarlo mediante Service Quotas ni creando un caso en AWS Support. Para evitar alcanzar la cuota de rotación bajo demanda, recomendamos utilizar la [rotación de claves automática](#) siempre que sea posible.

Cuotas de solicitudes

AWS KMS establece cuotas para el número de operaciones de API solicitadas por segundo. Las cuotas de solicitud varían según el funcionamiento de la API Región de AWS, la API y otros factores, como el tipo de clave de KMS. Cuando superas una cuota de solicitud de API AWS KMS, [limita la](#) solicitud.

Todas las cuotas de AWS KMS solicitudes son ajustables, excepto la cuota de [solicitud del almacén de AWS CloudHSM claves](#). Para solicitar un aumento de cuota, consulte [Solicitud de un aumento de cuota](#) en la Guía de usuario de Service Quotas. Para solicitar una reducción de cuota, cambiar una cuota que no aparece en Service Quotas o cambiar una cuota en una en la Región de AWS que Service Quotas for no AWS KMS esté disponible, visite [AWS Support Center](#) y cree un caso.

Si supera la cuota solicitada para la [GenerateDataKey](#) operación, considere la posibilidad de utilizar la función de almacenamiento en [caché de claves de datos](#) del AWS Encryption SDK. La reutilización de las claves de datos podría reducir la frecuencia de sus solicitudes también. AWS KMS

Además de solicitar cuotas, AWS KMS utiliza cuotas de recursos para garantizar la capacidad de todos los usuarios. Para obtener más información, consulte [Cuotas de recursos](#).

Para ver las tendencias de las tarifas de solicitudes, utilice la [Consola Service Quotas](#). También puedes crear una CloudWatch alarma de [Amazon](#) que te avise cuando tu porcentaje de solicitudes alcance un porcentaje determinado del valor de la cuota. Para obtener más información, consulta [Gestiona tus tasas de solicitudes de AWS KMS API mediante Service Quotas y Amazon CloudWatch](#) en el blog AWS de seguridad.

Temas

- [Solicita cuotas para cada operación AWS KMS de la API](#)
- [Aplicar cuotas de solicitudes](#)
- [Cuotas compartidas para operaciones criptográficas](#)

- [Solicitudes de la API realizadas en su nombre](#)
- [Solicitudes entre cuentas](#)
- [Cuotas de solicitudes del almacén de claves personalizado](#)

Solicita cuotas para cada operación AWS KMS de la API

En esta tabla se muestra el código de [cuota de Service](#) Quotas y el valor predeterminado de cada cuota de AWS KMS solicitud. Todas las cuotas de AWS KMS solicitud son ajustables, excepto la [cuota de solicitud del almacén de AWS CloudHSM claves](#).

Note

Es posible que tenga que desplazarse horizontal o verticalmente para ver todos los datos de esta tabla.

Nombre de la cuota	Límite predeterminado (solicitudes por segundo)
Cryptographic operations (symmetric) request rate Válido para: • Decrypt • Encrypt • GenerateDataKey • GenerateDataKeyWithoutPlainText • GenerateMac • GenerateRandom • ReEncrypt • VerifyMac	Estas cuotas compartidas varían en función del Región de AWS tipo de clave KMS utilizada en la solicitud. Cada cuota se calcula por separado. • 10 000 (compartidas) • 20 000 (compartidas) en las siguientes Regiones: • EE. UU. Este (Ohio), us-east-2 • Asia Pacífico (Singapur), ap-southeast-1 • Asia Pacífico (Sídney), ap-southeast-2 • Asia Pacífico (Tokio), ap-northeast-1 • Europa (Fráncfort), eu-central-1 • Europa (Londres), eu-west-2 • 100 000 (compartidas) en las siguientes Regiones:

Nombre de la cuota	Límite predeterminado (solicitudes por segundo)
	• EE.UU. Este (Norte de Virginia) (us-east-1) • EE.UU. Oeste (Oregón) (us-west-2) • Europa (Irlanda), eu-west-1
Cryptographic operations (RSA) request rate Válido para: • Decrypt • Encrypt • ReEncrypt • Sign • Verify	1 000 (compartidas) para claves KMS de RSA
Cryptographic operations (ML-DSA) request rate Válido para: • Sign • Verify	1 000 (compartidas) para claves de KMS de ML-DSA

Nombre de la cuota	Límite predeterminado (solicitudes por segundo)
Cryptographic operations (ECC and SM2) request rate Válido para: • Decrypt—solo compatible con claves KMS SM2 (solo para regiones de China) • DeriveSharedSecret • Encrypt—solo compatible con claves KMS SM2 (solo para regiones de China) • ReEncrypt —solo compatible con claves KMS SM2 (solo para regiones de China) • Sign • Verify	1000 (compartidas) para claves KMS de curva elíptica (ECC) y (solo regiones de SM2 China)
Custom key store request quotas Válido para: • Decrypt • DeriveSharedSecret • Encrypt • GenerateDataKey • GenerateDataKeyWithoutPlainText • GenerateRandom • ReEncrypt	Las cuotas de solicitudes del almacén de claves personalizado se calculan por separado para cada almacén de claves personalizado. • 1800 (compartidas) por cada almacén de claves AWS CloudHSM • 1800 (compartidas) para cada almacén de claves externo.
CancelKeyDeletion request rate	5
ConnectCustomKeyStore request rate	5

Nombre de la cuota	Límite predeterminado (solicitudes 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

Nombre de la cuota	Límite predeterminado (solicitudes por segundo)
<code>GenerateDataKeyPair (ECC_NIST_P256) request rate</code> Válido para: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100
<code>GenerateDataKeyPair (ECC_NIST_P384) request rate</code> Válido para: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100
<code>GenerateDataKeyPair (ECC_NIST_P521) request rate</code> Válido para: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100
<code>GenerateDataKeyPair (ECC_SECG_P256K1) request rate</code> Válido para: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100

Nombre de la cuota	Límite predeterminado (solicitudes por segundo)
<code>GenerateDataKeyPair (ECC_NIST_EDWARDS25519) request rate</code> Válido para: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100
<code>GenerateDataKeyPair (RSA_2048) request rate</code> Válido para: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	20
<code>GenerateDataKeyPair (RSA_3072) request rate</code> Válido para: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	4
<code>GenerateDataKeyPair (RSA_4096) request rate</code> Válido para: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	1

Nombre de la cuota	Límite predeterminado (solicitudes por segundo)
GenerateDataKeyPair (SM2 – China Regions only) request rate Válido para: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	25
GetKeyLastUsage request rate	5
GetKeyPolicy request rate	1 000
GetKeyRotationStatus request rate	1 000
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

Nombre de la cuota	Límite predeterminado (solicitudes por segundo)
<code>ReplicateKey request rate</code> 5 Una operación <code>ReplicateKey</code> cuenta como una solicitud <code>ReplicateKey</code> en la Región de la clave principal y dos solicitudes <code>CreateKey</code> en la Región de la réplica. Solo una de las solicitudes <code>CreateKey</code> es una ejecución en seco para detectar posibles problemas antes de crear la clave.	
<code>RetireGrant request rate</code>	50
<code>RevokeGrant request rate</code>	50
<code>RotateKeyOnDemand request rate</code>	5
<code>ScheduleKeyDeletion request rate</code>	15
<code>TagResource request rate</code>	10
<code>UntagResource request rate</code>	5
<code>UpdateAlias request rate</code>	5
<code>UpdateCustomKeyStore request rate</code>	5
<code>UpdateKeyDescription request rate</code>	5
<code>UpdatePrimaryRegion request rate</code> 5 Un operación <code>UpdatePrimaryRegion</code> cuenta como dos solicitudes <code>UpdatePrimaryRegion</code> ; una solicitud en cada una de las dos Regiones afectadas.	

Aplicar cuotas de solicitudes

Al revisar las cuotas de solicitudes, tenga en cuenta la siguiente información.

- Las cuotas de solicitudes se aplican a [claves administradas por el cliente](#) y [Claves administradas por AWS](#). El uso de [Claves propiedad de AWS](#) no tiene en cuenta las cuotas solicitadas para usted Cuenta de AWS, incluso cuando se utilizan para proteger los recursos de su cuenta.
- Las cuotas de solicitud se aplican a las solicitudes enviadas a puntos de conexión FIPS y puntos de conexión no FIPS. Para obtener una lista de los puntos finales del AWS KMS servicio, consulte los [AWS Key Management Service puntos finales y las cuotas](#) en. Referencia general de AWS
- La limitación controlada se basa en todas las solicitudes de las claves KMS de todos los tipos de la Región. Este total incluye las solicitudes de todos los directores del Cuenta de AWS, incluidas las solicitudes de AWS servicios en su nombre.
- Cada cuota de solicitud se calcula de forma independiente. Por ejemplo, las solicitudes de la [CreateKey](#) operación no afectan a la cuota de solicitudes de la [CreateAlias](#) operación. Si las solicitudes CreateAlias se limitan de forma controlada, las solicitudes CreateKey aún pueden completarse correctamente.
- Aunque las operaciones criptográficas comparten una cuota, la cuota compartida se calcula de manera independiente de las cuotas para otras operaciones. [Por ejemplo, las llamadas a las operaciones de cifrado y descifrado comparten una cuota de solicitudes, pero esa cuota es independiente de la cuota de las operaciones de administración, por ejemplo. EnableKey](#) Por ejemplo, en la Región Europa (Londres), puede realizar 10 000 operaciones criptográficas en claves KMS simétricas más 5 operaciones EnableKey por segundo sin que se limiten de forma controlada.

Cuotas compartidas para operaciones criptográficas

AWS KMS las [operaciones criptográficas comparten cuotas de](#) solicitud. Puede solicitar cualquier combinación de las operaciones criptográficas admitidas por la clave KMS, solo para que el número total de operaciones criptográficas no supere la cuota de solicitud para ese tipo de clave KMS. Las excepciones son [GenerateDataKeyPair](#) [GenerateDataKeyPairWithoutPlaintext](#), que comparten una cuota independiente.

Las cuotas para distintos tipos de claves KMS se calculan de forma independiente. Cada cuota se aplica a todas las solicitudes de estas operaciones en la región Cuenta de AWS y con el tipo de clave indicado en cada intervalo de un segundo.

- La tasa de solicitudes de operaciones criptográficas (simétricas) es la cuota de solicitudes compartidas para operaciones criptográficas que utilizan claves KMS simétricas en una cuenta y región. Esta cuota se aplica a las operaciones criptográficas con claves de cifrado simétricas y claves HMAC, que también son simétricas.

Por ejemplo, es posible que esté utilizando [claves KMS simétricas](#) Región de AWS con una cuota compartida de 10 000 solicitudes por segundo. Cuando realizas 7 000 [GenerateDataKey](#) solicitudes por segundo y 2 000 solicitudes de [descifrado](#) por segundo, AWS KMS no se limitan tus solicitudes. Sin embargo, cuando realice 9500 solicitudes [GenerateDataKey](#) y 1000 solicitudes [Encrypt](#) por segundo, AWS KMS limita de forma controlada las solicitudes porque superan la cuota compartida.

Las operaciones criptográficas en las [claves KMS de cifrado simétrico](#) de un [almacén de claves personalizado](#) cuentan tanto para la tasa de solicitudes de operaciones criptográficas (simétricas) de la cuenta como para la [cuota de solicitudes del almacén de claves personalizado](#) para el almacén de claves personalizado.

- La tasa de solicitudes de operaciones criptográficas (RSA) es la cuota de solicitudes compartidas para operaciones criptográficas que utilizan [claves KMS asimétricas RSA](#).

Por ejemplo, con una cuota de solicitudes de 1 000 operaciones por segundo, puede realizar 400 solicitudes de [Cifrado](#) y 200 solicitudes de [Descifrado](#) con claves KMS RSA que pueden cifrar y descifrar, además de 250 solicitudes de [Firma](#) y 150 solicitudes de [Verificación](#) con claves KMS RSA que pueden firmar y verificar.

- La tasa de solicitudes de operaciones criptográficas (ECC) es la cuota de solicitudes compartidas para operaciones criptográficas que utilizan [claves KMS asimétricas de curva elíptica \(ECC\)](#) y [claves KMS asimétricas SM](#).

Por ejemplo, con una cuota de solicitudes de 1000 operaciones por segundo, puedes realizar 400 solicitudes de firma y 200 solicitudes de verificación con claves KMS de ECC que pueden firmar y verificar, además de 250 solicitudes de [firma](#) y 150 solicitudes de verificación con claves de SM2 KMS que pueden firmar y [verificar](#).

- La cuota de solicitud de almacén de claves personalizadas es la cuota de solicitud compartida para operaciones criptográficas en claves KMS en un almacén de claves personalizado. Esta cuota se calcula por separado para cada almacén de claves personalizado.

Las operaciones criptográficas en las [claves de KMS de cifrado simétrico](#) de un [almacén de claves personalizado](#) cuentan tanto para la tasa de solicitudes de operaciones criptográficas (simétricas)

de la cuenta como para la [cuota de solicitudes del almacén de claves personalizado](#) para el almacén de claves personalizado.

Las cuotas para distintos tipos de clave también se calculan de forma independiente. Por ejemplo, en la Región Asia Pacífico (Singapur), si utiliza claves KMS simétricas y asimétricas, puede realizar hasta 10 000 llamadas por segundo con claves KMS simétricas (incluidas claves HMAC) más un máximo de 500 llamadas adicionales por segundo con claves KMS asimétricas RSA, más un máximo de 300 solicitudes adicionales por segundo con claves KMS basadas en ECC.

Solicitudes de la API realizadas en su nombre

Puedes realizar solicitudes a la API directamente o mediante un AWS servicio integrado que realice las solicitudes a la API AWS KMS en tu nombre. La cuota se aplica a ambos tipos de solicitudes.

Por ejemplo, puede almacenar datos en Amazon S3 utilizando el cifrado del servidor con una clave KMS (SSE-KMS). Cada vez que carga o descarga un objeto de S3 cifrado con SSE-KMS, Amazon S3 realiza una solicitud `GenerateDataKey` (para cargas) o `Decrypt` (para descargas) AWS KMS en su nombre. Estas solicitudes se tienen en cuenta para su cuota, por lo AWS KMS que limita las solicitudes si supera un total combinado de 5 500 (o 10 000 o 50 000, según el volumen Región de AWS) de cargas o descargas por segundo de objetos de S3 cifrados con SSE-KMS.

Solicitudes entre cuentas

Cuando una de las aplicaciones de una aplicación Cuenta de AWS utiliza una clave de KMS propiedad de otra cuenta, se denomina solicitud multicuenta. En el caso de las solicitudes entre cuentas, AWS KMS limita de forma controlada la cuenta que realiza las solicitudes, no la cuenta que posee la clave KMS. Por ejemplo, si una aplicación de una cuenta A utiliza una clave KMS de la cuenta B, el uso de la clave KMS se aplica solo a las cuotas de la cuenta A.

Cuotas de solicitudes del almacén de claves personalizado

AWS KMS mantiene las cuotas de solicitud para [las operaciones criptográficas en las](#) claves de KMS en un almacén de [claves personalizado](#). Esta solicitud de cuotas se calcula por separado para cada almacén de claves personalizado.

Cuota de solicitudes del almacén de claves personalizado	Valor predeterminado (solicitudes por segundo) para cada almacén de claves personalizado	Ajustable
AWS CloudHSM cuota de solicitud del almacén de claves	1800	No
Cuota de solicitudes del almacén de claves externo	1800	No

Note

AWS KMS [las cuotas de solicitud de almacén de claves personalizadas](#) no aparecen en la consola de Service Quotas. No puede ver ni administrar estas cuotas mediante las operaciones de la API de Service Quotas.

Si el AWS CloudHSM clúster asociado a un almacén de AWS CloudHSM claves procesa numerosos comandos, incluidos aquellos que no están relacionados con el almacén de claves personalizado, es posible que obtengas un `AWS KMS ThrottlingException` lower-than-expected ritmo. Si esto ocurre, reduce la tasa de solicitudes a AWS KMS, reduce la carga no relacionada o usa un AWS CloudHSM clúster dedicado para tu almacén de AWS CloudHSM claves.

AWS KMS informa de la limitación de las solicitudes de almacenes de claves externos en la [ExternalKeyStoreThrottle](#) CloudWatch métrica. Puede usar esta métrica para ver los patrones de limitación, crear alarmas y ajustar su cuota de solicitudes del almacén de claves externo.

Una solicitud de una [operación criptográfica](#) en una clave KMS en un almacén de claves personalizado cuenta para dos cuotas:

- Cuota de tasas de solicitud de operaciones criptográficas (simétricas) (por cuenta)

Las solicitudes de operaciones criptográficas en las claves KMS de un almacén de claves personalizado se cuentan para la cuota `Cryptographic operations (symmetric) request rate` para cada Cuenta de AWS y región. Por ejemplo, en Este de EE. UU. (norte de

Virginia) (us-east-1), cada Cuenta de AWS puede tener hasta 100 000 solicitudes por segundo en claves de KMS de cifrado simétrico, incluidas las solicitudes que usan una clave de KMS en un almacén de claves personalizado.

- Cuota de solicitudes de almacén de claves personalizado (por almacén de claves personalizado)

Las solicitudes de operaciones criptográficas en claves KMS en un almacén de claves personalizado también cuentan para un Custom key store request cuota de 1800 operaciones por segundo. Estas cuotas se calculan por separado para cada almacén de claves personalizado. Pueden incluir solicitudes de varios Cuentas de AWS que utilizan claves de KMS en el almacén de claves personalizado.

Por ejemplo, si solicita una operación [Encrypt](#) en una clave de KMS dentro de un almacén de claves personalizado (de cualquier tipo) en la región Este de EE. UU. (norte de Virginia) (us-east-1), cuenta para la cuota a nivel de cuenta Cryptographic operations (symmetric) request rate (100 000 solicitudes por segundo) para su cuenta y región, y para una Custom key store request cuota (1800 solicitudes por segundo) para su almacén de claves personalizado. Sin embargo, una solicitud de una operación de administración [PutKeyPolicy](#), como una clave KMS de un almacén de claves personalizado, solo se aplica a su cuota a nivel de cuenta (15 solicitudes por segundo).

Limitar las solicitudes AWS KMS

Para garantizar que AWS KMS puede proporcionar respuestas rápidas y fiables a las solicitudes de API de todos los clientes, limita las solicitudes de API que superan ciertos límites.

La limitación se produce cuando se AWS KMS rechaza una solicitud que, de otro modo, podría ser válida y se produce un `ThrottlingException` error como el siguiente.

```
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 las solicitudes que cumplen las siguientes condiciones.

- La tasa de solicitudes por segundo supera la [cuota de AWS KMS solicitudes](#) de una cuenta y una región.

Por ejemplo, si los usuarios de tu cuenta envían 1000 `DescribeKey` solicitudes en un segundo, limita AWS KMS todas las `DescribeKey` solicitudes subsiguientes en ese segundo.

Para responder a la limitación controlada, utilice una [estrategia de interrupción y reintento](#). En algunos casos, esta estrategia se implementa automáticamente para los errores de HTTP 400. AWS SDKs

- Una velocidad alta o sostenida de solicitudes para cambiar el estado de la misma clave KMS. Esta condición se conoce a menudo como una “tecla de acceso rápido”.

Por ejemplo, si una aplicación de tu cuenta envía una oleada persistente de `DisableKey` solicitudes de la misma clave KMS `EnableKey` y las solicita, limita AWS KMS las solicitudes. Esta limitación se produce incluso si las solicitudes no superan el límite de solicitudes de las operaciones request-per-second y. `EnableKey` `DisableKey`

Para responder a la limitación controlada, ajuste la lógica de la aplicación para que solo realice solicitudes requeridas o consolide las solicitudes de varias funciones.

- Es posible que las solicitudes de operaciones con las claves de KMS de un [AWS CloudHSM almacén](#) de claves se limiten a un lower-than-expected ritmo similar al que el AWS CloudHSM clúster asociado al almacén de AWS CloudHSM claves procesa numerosos comandos, incluidos aquellos que no están relacionados con el almacén de claves. AWS CloudHSM

(ya AWS KMS no limita las solicitudes de operaciones con las claves de KMS de un AWS CloudHSM almacén de claves cuando no hay sesiones de PKCS #11 disponibles para el clúster. AWS CloudHSM En su lugar, lanza un `KMSInternalException` y recomienda que vuelvas a intentar la solicitud.)

Para ver las tendencias de las tarifas de solicitudes, utilice la [Consola Service Quotas](#). También puedes crear una CloudWatch alarma de [Amazon](#) que te avise cuando tu porcentaje de solicitudes alcance un porcentaje determinado del valor de la cuota. Para obtener más información, consulta [Gestiona tus tasas de solicitudes de AWS KMS API mediante Service Quotas y Amazon CloudWatch](#) en el blog AWS de seguridad.

Todas AWS KMS las cuotas son ajustables, excepto la cuota de [recursos de rotación bajo demanda y la cuota](#) de [solicitudes del almacén de AWS CloudHSM claves](#). Para solicitar un aumento de cuota, consulte [Solicitud de un aumento de cuota](#) en la Guía de usuario de Service Quotas. Para solicitar una reducción de cuota, cambiar una cuota que no aparece en Service Quotas o cambiar una cuota

en una en la Región de AWS que Service Quotas for no AWS KMS esté disponible, visite [AWS Support Center](#) y cree un caso.

 Note

AWS KMS [las cuotas de solicitud de almacén de claves personalizadas](#) no aparecen en la consola de Service Quotas. No puede ver ni administrar esta cuotas mediante las operaciones de la API de Service Quotas.

Ejemplos de código para AWS KMS usar AWS SDKs

Los siguientes ejemplos de código muestran cómo usarlo AWS KMS con un kit de desarrollo de AWS software (SDK).

Los conceptos básicos son ejemplos de código que muestran cómo realizar las operaciones esenciales dentro de un servicio.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- [Ejemplos básicos de AWS KMS uso AWS SDKs](#)
 - [Hola AWS Key Management Service](#)
 - [Aprenda los conceptos básicos del AWS KMS uso de un AWS SDK](#)
 - [Acciones de AWS KMS uso AWS SDKs](#)
 - [Úselo CreateAlias con un AWS SDK o CLI](#)
 - [Úselo CreateGrant con un AWS SDK o CLI](#)
 - [Úselo CreateKey con un AWS SDK o CLI](#)
 - [Úselo Decrypt con un AWS SDK o CLI](#)
 - [Úselo DeleteAlias con un AWS SDK o CLI](#)
 - [Úselo DescribeKey con un AWS SDK o CLI](#)
 - [Úselo DisableKey con un AWS SDK o CLI](#)
 - [Úselo EnableKey con un AWS SDK o CLI](#)
 - [Úselo EnableKeyRotation con un AWS SDK o CLI](#)
 - [Úselo Encrypt con un AWS SDK o CLI](#)
 - [Úselo GenerateDataKey con un AWS SDK o CLI](#)

- [Úselo `GenerateDataKeyWithoutPlaintext` con un AWS SDK o CLI](#)
- [Úselo `GenerateRandom` con un AWS SDK o CLI](#)
- [Úselo `GetKeyPolicy` con un AWS SDK o CLI](#)
- [Úselo `ListAliases` con un AWS SDK o CLI](#)
- [Úselo `ListGrants` con un AWS SDK o CLI](#)
- [Úselo `ListKeyPolicies` con un AWS SDK o CLI](#)
- [Úselo `ListKeys` con un AWS SDK o CLI](#)
- [Úselo `PutKeyPolicy` con un AWS SDK o CLI](#)
- [Úselo `ReEncrypt` con un AWS SDK o CLI](#)
- [Úselo `RetireGrant` con un AWS SDK o CLI](#)
- [Úselo `RevokeGrant` con un AWS SDK o CLI](#)
- [Úselo `ScheduleKeyDeletion` con un AWS SDK o CLI](#)
- [Úselo `Sign` con un AWS SDK o CLI](#)
- [Úselo `TagResource` con un AWS SDK o CLI](#)
- [Úselo `UpdateAlias` con un AWS SDK o CLI](#)
- [Úselo `Verify` con un AWS SDK o CLI](#)
- [Escenarios de AWS KMS uso AWS SDKs](#)
 - [Trabaje con el cifrado de tablas de DynamoDB mediante la versión 2 AWS Command Line Interface](#)

Ejemplos básicos de AWS KMS uso AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de AWS Key Management Service with AWS SDKs.

Ejemplos

- [Hola AWS Key Management Service](#)
- [Aprenda los conceptos básicos del AWS KMS uso de un AWS SDK](#)
- [Acciones de AWS KMS uso AWS SDKs](#)
 - [Úselo `CreateAlias` con un AWS SDK o CLI](#)
 - [Úselo `CreateGrant` con un AWS SDK o CLI](#)
 - [Úselo `CreateKey` con un AWS SDK o CLI](#)

- [Úselo Decrypt con un AWS SDK o CLI](#)
- [Úselo DeleteAlias con un AWS SDK o CLI](#)
- [Úselo DescribeKey con un AWS SDK o CLI](#)
- [Úselo DisableKey con un AWS SDK o CLI](#)
- [Úselo EnableKey con un AWS SDK o CLI](#)
- [Úselo EnableKeyRotation con un AWS SDK o CLI](#)
- [Úselo Encrypt con un AWS SDK o CLI](#)
- [Úselo GenerateDataKey con un AWS SDK o CLI](#)
- [Úselo GenerateDataKeyWithoutPlaintext con un AWS SDK o CLI](#)
- [Úselo GenerateRandom con un AWS SDK o CLI](#)
- [Úselo GetKeyPolicy con un AWS SDK o CLI](#)
- [Úselo ListAliases con un AWS SDK o CLI](#)
- [Úselo ListGrants con un AWS SDK o CLI](#)
- [Úselo ListKeyPolicies con un AWS SDK o CLI](#)
- [Úselo ListKeys con un AWS SDK o CLI](#)
- [Úselo PutKeyPolicy con un AWS SDK o CLI](#)
- [Úselo ReEncrypt con un AWS SDK o CLI](#)
- [Úselo RetireGrant con un AWS SDK o CLI](#)
- [Úselo RevokeGrant con un AWS SDK o CLI](#)
- [Úselo ScheduleKeyDeletion con un AWS SDK o CLI](#)
- [Úselo Sign con un AWS SDK o CLI](#)
- [Úselo TagResource con un AWS SDK o CLI](#)
- [Úselo UpdateAlias con un AWS SDK o CLI](#)
- [Úselo Verify con un AWS SDK o CLI](#)

Hola AWS Key Management Service

En los siguientes ejemplos de código se muestra cómo empezar a utilizar AWS Key Management Service.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /**
         * The `subscribe` method is required when using paginator methods in the
         * AWS SDK
         * because paginator methods return an instance of a `ListKeysPublisher`,
         * which is

```

```

        * based on a reactive stream. This allows asynchronous retrieval of
        paginated
        * results as they become available. By subscribing to the stream, we can
        process
        * each page of results as they are emitted.
        */
        ListKeysPublisher keysPublisher =
kmsAsyncClient.listKeysPaginator(listKeysRequest);
        CompletableFuture<Void> future = keysPublisher
            .subscribe(r -> r.keys().forEach(key ->
                System.out.println("The key ARN is: " + key.keyArn() + ". The key
Id is: " + key.keyId()))
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    System.err.println("Error occurred: " +
exception.getMessage());
                } else {
                    System.out.println("Successfully listed all keys.");
                }
            });

        try {
            future.join();
        } catch (Exception e) {
            System.err.println("Failed to list keys: " + e.getMessage());
        }
    }
}

```

- Para obtener más información sobre la API, consulta [ListKeys](#) la Referencia AWS SDK for Java 2.x de la API.

PHP

SDK para PHP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
include "vendor/autoload.php";

use Aws\Kms\KmsClient;

echo "This file shows how to connect to the KmsClient, uses a paginator to get
the keys for the account, and lists the KeyIds for up to 10 keys.\n";

$client = new KmsClient([]);

$pageLength = 10; // Change this value to change the number of records shown, or
to break up the result into pages.

$keys = [];
$keysPaginator = $client->getPaginator("ListKeys", ['Limit' => $pageLength]);
foreach($keysPaginator as $page){
    foreach($page['Keys'] as $index => $key){
        echo "The $index index Key's ID is: {$key['KeyId']}\n";
    }
    echo "End of page one of results. Alter the \$pageLength variable to see more
results.\n";
    break;
}
```

- Para obtener más información sobre la API, consulta [ListKeys](#) la Referencia AWS SDK para PHP de la API.

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

TRY.

```
oo_result = lo_kms->listkeys( ).
MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.
```



```
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [ListKeys](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Aprenda los conceptos básicos del AWS KMS uso de un AWS SDK

En el siguiente ejemplo de código, se muestra cómo:

- Creación de una clave de KMS.
- Enumerar las claves de KMS de su cuenta y obtener información sobre ellas.
- Habilitar y desactivar claves KMS.
- Generar una clave de datos simétrica que pueda usarse para el cifrado del cliente.
- Genere una clave asimétrica que se utilice para firmar digitalmente los datos.
- Tag keys (Claves de etiqueta).
- Eliminar las claves KMS.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

Ejecute un escenario en un símbolo del sistema.

```
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 una clase que ajuste las acciones de 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 obtener detalles sobre la API, consulte los siguientes temas en la Referencia de la API de 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

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
echo "\n";
echo "-----\n";
echo <<<WELCOME
Welcome to the AWS Key Management Service SDK Basics scenario.

This program demonstrates how to interact with AWS Key Management Service using
the AWS SDK for PHP (v3).
The AWS Key Management Service (KMS) is a secure and highly available service
that allows you to create
and manage AWS KMS keys and control their use across a wide range of AWS services
and applications.
KMS provides a centralized and unified approach to managing encryption keys,
making it easier to meet your
data protection and regulatory compliance requirements.

This KMS Basics scenario creates two key types:
- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...\n
WELCOME;
echo "-----\n";
$this->pressEnter();

$this->kmsClient = new KmsClient([]);
// Initialize the KmsService class with the client. This allows you to
override any defaults in the client before giving it to the service class.
$this->kmsService = new KmsService($this->kmsClient);

// 1. Create a symmetric KMS key.
echo "\n";
```

```
    echo "1. Create a symmetric KMS key.\n";
    echo "First, we will create a symmetric KMS key that is used to encrypt
and decrypt data by invoking createKey().\n";
    $this->pressEnter();

    $key = $this->kmsService->createKey();
    $this->resources['symmetricKey'] = $key['KeyId'];
    echo "Created a customer key with ARN {$key['Arn']}. \n";
    $this->pressEnter();

    // 2. Enable a KMS key.
    echo "\n";
    echo "2. Enable a KMS key.\n";
    echo "By default when you create an AWS key, it is enabled. The code
checks to
determine if the key is enabled. If it is not enabled, the code enables it.\n";
    $this->pressEnter();

    $keyInfo = $this->kmsService->describeKey($key['KeyId']);
    if(!$keyInfo['Enabled']){
        echo "The key was not enabled, so we will enable it.\n";
        $this->pressEnter();
        $this->kmsService->enableKey($key['KeyId']);
        echo "The key was successfully enabled.\n";
    }else{
        echo "The key was already enabled, so there was no need to enable it.
\n";
    }
    $this->pressEnter();

    // 3. Encrypt data using the symmetric KMS key.
    echo "\n";
    echo "3. Encrypt data using the symmetric KMS key.\n";
    echo "One of the main uses of symmetric keys is to encrypt and decrypt
data.\n";
    echo "Next, we'll encrypt the string 'Hello, AWS KMS!' with the
SYMMETRIC_DEFAULT encryption algorithm.\n";
    $this->pressEnter();
    $text = "Hello, AWS KMS!";
    $encryption = $this->kmsService->encrypt($key['KeyId'], $text);
    echo "The plaintext data was successfully encrypted with the algorithm:
{$encryption['EncryptionAlgorithm']}. \n";
    $this->pressEnter();
```

```
// 4. Create an alias.
echo "\n";
echo "4. Create an alias.\n";
$aliasInput = testable_readline("Please enter an alias prefixed with
\"alias/\" or press enter to use a default value: ");
if($aliasInput == ""){
    $aliasInput = "alias/dev-encryption-key";
}
$this->kmsService->createAlias($key['KeyId'], $aliasInput);
$this->resources['alias'] = $aliasInput;
echo "The alias \"$aliasInput\" was successfully created.\n";
$this->pressEnter();

// 5. List all of your aliases.
$aliasPageSize = 10;
echo "\n";
echo "5. List all of your aliases, up to $aliasPageSize.\n";
$this->pressEnter();
$aliasPaginator = $this->kmsService->listAliases();
foreach($aliasPaginator as $pages){
    foreach($pages['Aliases'] as $alias){
        echo $alias['AliasName'] . "\n";
    }
    break;
}
$this->pressEnter();

// 6. Enable automatic rotation of the KMS key.
echo "\n";
echo "6. Enable automatic rotation of the KMS key.\n";
echo "By default, when the SDK enables automatic rotation of a KMS key,
KMS rotates the key material of the KMS key one year (approximately 365 days)
from the enable date and every year
thereafter.";
$this->pressEnter();
$this->kmsService->enableKeyRotation($key['KeyId']);
echo "The key's rotation was successfully set for key:
{$key['KeyId']}\n";
$this->pressEnter();

// 7. Create a grant.
echo "7. Create a grant.\n";
echo "\n";
```

```
    echo "A grant is a policy instrument that allows Amazon Web Services
principals to use KMS keys.
It also can allow them to view a KMS key (DescribeKey) and create and manage
grants.
When authorizing access to a KMS key, grants are considered along with key
policies and IAM policies.\n";
    $granteeARN = testable_readline("Please enter the Amazon Resource Name
(ARN) of an Amazon Web Services principal. Valid principals include Amazon
Web Services accounts, IAM users, IAM roles, federated users, and assumed
role users. For help with the ARN syntax for a principal, see IAM ARNs in the
Identity and Access Management User Guide. \nTo skip this step, press enter
without any other values: ");
    if($granteeARN){
        $operations = [
            "ENCRYPT",
            "DECRYPT",
            "DESCRIBE_KEY",
        ];
        $grant = $this->kmsService->createGrant($key['KeyId'], $granteeARN,
$operations);
        echo "The grant Id is: {$grant['GrantId']}\n";
    }else{
        echo "Steps 7, 8, and 9 will be skipped.\n";
    }
    $this->pressEnter();

    // 8. List grants for the KMS key.
    if($granteeARN){
        echo "8. List grants for the KMS key.\n\n";
        $grantsPaginator = $this->kmsService->listGrants($key['KeyId']);
        foreach($grantsPaginator as $page){
            foreach($page['Grants'] as $grant){
                echo $grant['GrantId'] . "\n";
            }
        }
    }else{
        echo "Skipping step 8...\n";
    }
    $this->pressEnter();

    // 9. Revoke the grant.
    if($granteeARN) {
        echo "\n";
        echo "9. Revoke the grant.\n";
```

```

        $this->pressEnter();
        $this->kmsService->revokeGrant($grant['GrantId'], $keyInfo['KeyId']);
        echo "{$grant['GrantId']} was successfully revoked!\n";
    }else{
        echo "Skipping step 9...\n";
    }
    $this->pressEnter();

    // 10. Decrypt the data.
    echo "\n";
    echo "10. Decrypt the data.\n";
    echo "Let's decrypt the data that was encrypted before.\n";
    echo "We'll use the same key to decrypt the string that we encrypted
earlier in the program.\n";
    $this->pressEnter();
    $decryption = $this->kmsService->decrypt($keyInfo['KeyId'],
$encryption['CiphertextBlob'], $encryption['EncryptionAlgorithm']);
    echo "The decrypted text is: {$decryption['Plaintext']}\n";
    $this->pressEnter();

    // 11. Replace a Key Policy.
    echo "\n";
    echo "11. Replace a Key Policy.\n";
    echo "A key policy is a resource policy for a KMS key. Key policies are
the primary way to control access to KMS keys.\n";
    echo "Every KMS key must have exactly one key policy. The statements in
the key policy determine who has permission to use the KMS key and how they can
use it.\n";
    echo " You can also use IAM policies and grants to control access to the
KMS key, but every KMS key must have a key policy.\n";
    echo "We will replace the key's policy with a new one:\n";
    $stsClient = new StsClient([]);
    $result = $stsClient->getCallerIdentity();
    $accountId = $result['Account'];
    $keyPolicy = <<< KEYPOLICY
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::$accountId:root"},
        "Action": "kms:*",
        "Resource": "*"
    }]
}

```

```
KEYPOLICY;
    echo $keyPolicy;
    $this->pressEnter();
    $this->kmsService->putKeyPolicy($keyInfo['KeyId'], $keyPolicy);
    echo "The Key Policy was successfully replaced!\n";
    $this->pressEnter();

    // 12. Retrieve the key policy.
    echo "\n";
    echo "12. Retrieve the key policy.\n";
    echo "Let's get some information about the new policy and print it to the
screen.\n";
    $this->pressEnter();
    $policyInfo = $this->kmsService->getKeyPolicy($keyInfo['KeyId']);
    echo "We got the info! Here is the policy: \n";
    echo $policyInfo['Policy'] . "\n";
    $this->pressEnter();

    // 13. Create an asymmetric KMS key and sign data.
    echo "\n";
    echo "13. Create an asymmetric KMS key and sign data.\n";
    echo "Signing your data with an AWS key can provide several benefits that
make it an attractive option for your data signing needs.\n";
    echo "By using an AWS KMS key, you can leverage the security controls and
compliance features provided by AWS, which can help you meet various regulatory
requirements and enhance the overall security posture of your organization.\n";
    echo "First we'll create the asymmetric key.\n";
    $this->pressEnter();
    $keySpec = "RSA_2048";
    $keyUsage = "SIGN_VERIFY";
    $asymmetricKey = $this->kmsService->createKey($keySpec, $keyUsage);
    $this->resources['asymmetricKey'] = $asymmetricKey['KeyId'];
    echo "Created the key with ID: {$asymmetricKey['KeyId']}\n";
    echo "Next, we'll sign the data.\n";
    $this->pressEnter();
    $algorithm = "RSASSA_PSS_SHA_256";
    $sign = $this->kmsService->sign($asymmetricKey['KeyId'], $text,
$algorithm);
    $verify = $this->kmsService->verify($asymmetricKey['KeyId'], $text,
$sign['Signature'], $algorithm);
    echo "Signature verification result: {$sign['signature']}\n";
    $this->pressEnter();

    // 14. Tag the symmetric KMS key.
```



```

    echo "\n";
    echo "14. Tag the symmetric KMS key.\n";
    echo "By using tags, you can improve the overall management, security,
and governance of your KMS keys, making it easier to organize, track, and
control access to your encrypted data within your AWS environment.\n";
    echo "Let's tag our symmetric key as Environment->Production\n";
    $this->pressEnter();
    $this->kmsService->tagResource($key['KeyId'], [
        [
            'TagKey' => "Environment",
            'TagValue' => "Production",
        ],
    ]);
    echo "The key was successfully tagged!\n";
    $this->pressEnter();

    // 15. Schedule the deletion of the KMS key
    echo "\n";
    echo "15. Schedule the deletion of the KMS key.\n";
    echo "By default, KMS applies a waiting period of 30 days, but you can
specify a waiting period of 7-30 days.\n";
    echo "When this operation is successful, the key state of the KMS key
changes to PendingDeletion and the key can't be used in any cryptographic
operations.\n";
    echo "It remains in this state for the duration of the waiting period.\n
\n";

    echo "Deleting a KMS key is a destructive and potentially dangerous
operation. When a KMS key is deleted, all data that was encrypted under the KMS
key is unrecoverable.\n\n";

    $cleanUp = testable_readline("Would you like to delete the resources
created during this scenario, including the keys? (y/n): ");
    if($cleanUp == "Y" || $cleanUp == "y"){
        $this->cleanUp();
    }

    echo
    "-----
\n";
    echo "This concludes the AWS Key Management SDK Basics scenario\n";
    echo
    "-----
\n";

```

```
namespace Kms;

use Aws\Kms\Exception\KmsException;
use Aws\Kms\KmsClient;
use Aws\Result;
use Aws\ResultPaginator;
use AwsUtilities\AWSServiceClass;

class KmsService extends AWSServiceClass
{

    protected KmsClient $client;
    protected bool $verbose;

    /**
     * @param KmsClient|null $client
     * @param bool $verbose
     */
    public function __construct(KmsClient $client = null, bool $verbose = false)
    {
        $this->verbose = $verbose;
        if($client){
            $this->client = $client;
            return;
        }
        $this->client = new KmsClient([]);
    }

    /**
     * @param string $keySpec
     * @param string $keyUsage
     * @param string $description
     * @return array
     */
    public function createKey(string $keySpec = "", string $keyUsage = "", string
    $description = "Created by the SDK for PHP")
    {
        $parameters = ['Description' => $description];
        if($keySpec && $keyUsage){
            $parameters['KeySpec'] = $keySpec;
```

```

        $parameters['KeyUsage'] = $keyUsage;
    }
    try {
        $result = $this->client->createKey($parameters);
        return $result['KeyMetadata'];
    } catch (KmsException $caught){
        // Check for error specific to createKey operations
        if ($caught->getAwsErrorMessage() == "LimitExceededException"){
            echo "The request was rejected because a quota was exceeded. For
more information, see Quotas in the Key Management Service Developer Guide.";
        }
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @param string $ciphertext
 * @param string $algorithm
 * @return Result
 */
public function decrypt(string $keyId, string $ciphertext, string $algorithm
= "SYMMETRIC_DEFAULT")
{
    try{
        return $this->client->decrypt([
            'CiphertextBlob' => $ciphertext,
            'EncryptionAlgorithm' => $algorithm,
            'KeyId' => $keyId,
        ]);
    } catch (KmsException $caught){
        echo "There was a problem decrypting the data: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @param string $text

```

```

    * @return Result
    */
    public function encrypt(string $keyId, string $text)
    {
        try {
            return $this->client->encrypt([
                'KeyId' => $keyId,
                'Plaintext' => $text,
            ]);
        } catch (KmsException $caught) {
            if ($caught->getAwsErrorMessage() == "DisabledException") {
                echo "The request was rejected because the specified KMS key is
not enabled.\n";
            }
            throw $caught;
        }
    }

    /**
     * @param string $keyId
     * @param int $limit
     * @return ResultPaginator
     */
    public function listAliases(string $keyId = "", int $limit = 0)
    {
        $args = [];
        if ($keyId) {
            $args['KeyId'] = $keyId;
        }
        if ($limit) {
            $args['Limit'] = $limit;
        }
        try {
            return $this->client->getPaginator("ListAliases", $args);
        } catch (KmsException $caught) {
            if ($caught->getAwsErrorMessage() == "InvalidMarkerException") {
                echo "The request was rejected because the marker that specifies
where pagination should next begin is not valid.\n";
            }
            throw $caught;
        }
    }
}

```

```
/**
 * @param string $keyId
 * @param string $alias
 * @return void
 */
public function createAlias(string $keyId, string $alias)
{
    try{
        $this->client->createAlias([
            'TargetKeyId' => $keyId,
            'AliasName' => $alias,
        ]);
    }catch (KmsException $caught){
        if($caught->getAwsErrorMessage() == "InvalidAliasNameException"){
            echo "The request was rejected because the specified alias name
is not valid.";
        }
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @param string $granteePrincipal
 * @param array $operations
 * @param array $grantTokens
 * @return Result
 */
public function createGrant(string $keyId, string $granteePrincipal, array
$operations, array $grantTokens = [])
{
    $args = [
        'KeyId' => $keyId,
        'GranteePrincipal' => $granteePrincipal,
        'Operations' => $operations,
    ];
    if($grantTokens){
        $args['GrantTokens'] = $grantTokens;
    }
}
```

```

        try{
            return $this->client->createGrant($args);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidGrantTokenException"){
                echo "The request was rejected because the specified grant token
is not valid.\n";
            }
            throw $caught;
        }
    }

    /**
     * @param string $keyId
     * @return array
     */
    public function describeKey(string $keyId)
    {
        try {
            $result = $this->client->describeKey([
                "KeyId" => $keyId,
            ]);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
resource could not be found.\n";
            }
            throw $caught;
        }
    }

    /**
     * @param string $keyId
     * @return void
     */
    public function disableKey(string $keyId)
    {
        try {
            $this->client->disableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
resource could not be found.\n";
            }
            throw $caught;
        }
    }

```

```

    });
    }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 obtener detalles sobre la API, consulte los siguientes temas en la Referencia de la API de 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

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KMSScenario:
    """Runs an interactive scenario that shows how to get started with KMS."""

    def __init__(
        self,
        key_manager: KeyManager,
        key_encryption: KeyEncrypt,
        alias_manager: AliasManager,
        grant_manager: GrantManager,
        key_policy: KeyPolicy,
    ):
        self.key_manager = key_manager
        self.key_encryption = key_encryption
        self.alias_manager = alias_manager
        self.grant_manager = grant_manager
        self.key_policy = key_policy
        self.key_id = ""
        self.alias_name = ""
        self.asymmetric_key_id = ""

    def kms_scenario(self):
        key_description = "Created by the AWS KMS API"

        print(DASHES)
        print(
            """
Welcome to the AWS Key Management SDK Basics scenario.

This program demonstrates how to interact with AWS Key Management using the AWS
SDK for Python (Boto3).
The AWS Key Management Service (KMS) is a secure and highly available service
that allows you to create
```

and manage AWS KMS keys and control their use across a wide range of AWS services and applications.

KMS provides a centralized and unified approach to managing encryption keys, making it easier to meet your data protection and regulatory compliance requirements.

This Basics scenario creates two key types:

- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...

```

        """
    )
    q.ask("Press Enter to continue...")

    print(DASHES)
    print(f"1. Create a symmetric KMS key\n")
    print(
        f"First, the program will creates a symmetric KMS key that you can
used to encrypt and decrypt data."
    )
    q.ask("Press Enter to continue...")
    self.key_id = self.key_manager.create_key(key_description)["KeyId"]
    print(f"A symmetric key was successfully created {self.key_id}.")
    q.ask("Press Enter to continue...")
    print(DASHES)
    print(
        """

```

2. Enable a KMS key

By default, when the SDK creates an AWS key, it is enabled. The next bit of code checks to

determine if the key is enabled.

```

        """
    )
    q.ask("Press Enter to continue...")
    is_enabled = self.is_key_enabled(self.key_id)
    print(f"Is the key enabled? {is_enabled}")
    if not is_enabled:
        self.key_manager.enable_key(self.key_id)
    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"3. Encrypt data using the symmetric KMS key")

```

```

    plain_text = "Hello, AWS KMS!"
    print(
        f"""

```

One of the main uses of symmetric keys is to encrypt and decrypt data.

Next, the code encrypts the string "{plain_text}" with the SYMMETRIC_DEFAULT encryption algorithm.

```

        """
    )
    q.ask("Press Enter to continue...")
    encrypted_text = self.key_encryption.encrypt(self.key_id, plain_text)
    print(DASHES)
    print(f"4. Create an alias")
    print(
        """

```

Now, the program will create an alias for the KMS key. An alias is a friendly name that you

can associate with a KMS key. The alias name should be prefixed with 'alias/'.

```

        """
    )
    alias_name = q.ask("Enter an alias name: ", q.non_empty)
    self.alias_manager.create_alias(self.key_id, alias_name)
    print(f"{alias_name} was successfully created.")
    self.alias_name = alias_name
    print(DASHES)
    print(f"5. List all of your aliases")
    q.ask("Press Enter to continue...")
    self.alias_manager.list_aliases(10)
    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"6. Enable automatic rotation of the KMS key")
    print(
        """

```

By default, when the SDK enables automatic rotation of a KMS key,

KMS rotates the key material of the KMS key one year (approximately 365 days) from the enable date and every year thereafter.

```

        """
    )
    q.ask("Press Enter to continue...")
    self.key_manager.enable_key_rotation(self.key_id)
    print(DASHES)
    print(f"Key rotation has been enabled for key with id {self.key_id}")
    print(

```



```

    """
7. Create a grant

```

A grant is a policy instrument that allows Amazon Web Services principals to use KMS keys.

It also can allow them to view a KMS key (DescribeKey) and create and manage grants.

When authorizing access to a KMS key, grants are considered along with key policies and IAM policies.

```

    """
    )
    print(
        """

```

To create a grant you must specify a `account_id`. To specify the grantee `account_id`, use the Amazon Resource Name (ARN)

of an AWS `account_id`. Valid principals include AWS accounts, IAM users, IAM roles, federated users, and assumed role users.

```

    """
    )
    account_id = q.ask(
        "Enter an account_id, or press enter to skip creating a grant... "
    )
    grant = None
    if account_id != "":
        grant = self.grant_manager.create_grant(
            self.key_id,
            account_id,
            [
                "Encrypt",
                "Decrypt",
                "DescribeKey",
            ],
        )
        print(f"Grant created successfully with ID: {grant['GrantId']}")

    q.ask("Press Enter to continue...")
    print(DASHES)
    print(DASHES)
    print(f"8. List grants for the KMS key")
    q.ask("Press Enter to continue...")
    self.grant_manager.list_grants(self.key_id)
    q.ask("Press Enter to continue...")
    print(DASHES)

```

```

    print(f"9. Revoke the grant")
    print(
        """

```

The revocation of a grant immediately removes the permissions and access that the grant had provided.

This means that any `account_id` (user, role, or service) that was granted access to perform specific

KMS operations on a KMS key will no longer be able to perform those operations.

```

        """
    )
    q.ask("Press Enter to continue...")

    if grant is not None:
        self.grant_manager.revoke_grant(self.key_id, grant["GrantId"])
        print(f"Grant ID: {grant['GrantId']} was successfully revoked!")

```

```

    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"10. Decrypt the data\n")
    print(
        """

```

Lets decrypt the data that was encrypted in an early step.

The code uses the same key to decrypt the string that we encrypted earlier in the program.

```

        """
    )
    q.ask("Press Enter to continue...")
    decrypted_data = self.key_encryption.decrypt(self.key_id, encrypted_text)
    print(f"Data decrypted successfully for key ID: {self.key_id}")
    print(f"Decrypted data: {decrypted_data}")

```

```

    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"11. Replace a key policy\n")
    print(
        """

```

A key policy is a resource policy for a KMS key. Key policies are the primary way to control

access to KMS keys. Every KMS key must have exactly one key policy. The statements in the key policy

determine who has permission to use the KMS key and how they can use it.

You can also use IAM policies and grants to control access to the KMS key, but every KMS key

must have a key policy.

By default, when you create a key by using the SDK, a policy is created that gives the AWS account that owns the KMS key full access to the KMS key.

Let's try to replace the automatically created policy with the following policy.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"AWS": "arn:aws:iam::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()

```

Clase contenedora y métodos para la administración de claves 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
```

Clase contenedora y métodos para los alias de claves 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

```

Clase contenedora y métodos para el cifrado de claves 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

```

Clase contenedora y métodos para las concesiones de claves 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
```


Clase contenedora y métodos para las políticas de claves 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 obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Python (Boto3).
 - [CreateAlias](#)
 - [CreateGrant](#)
 - [CreateKey](#)
 - [Decrypt](#)
 - [DescribeKey](#)
 - [DisableKey](#)

- [EnableKey](#)
- [Encrypt](#)
- [GetKeyPolicy](#)
- [ListAliases](#)
- [ListGrants](#)
- [ListKeys](#)
- [RevokeGrant](#)
- [ScheduleKeyDeletion](#)
- [Sign](#)
- [TagResource](#)

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Acciones de AWS KMS uso AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar AWS KMS acciones individuales con AWS SDKs. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Estos extractos se denominan AWS KMS API y son fragmentos de código de programas más grandes que deben ejecutarse en su contexto. Puede ver las acciones en contexto en [Escenarios de AWS KMS uso AWS SDKs](#).

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la [Referencia de la API de AWS Key Management Service](#).

Ejemplos

- [Úselo CreateAlias con un AWS SDK o CLI](#)
- [Úselo CreateGrant con un AWS SDK o CLI](#)
- [Úselo CreateKey con un AWS SDK o CLI](#)
- [Úselo Decrypt con un AWS SDK o CLI](#)
- [Úselo DeleteAlias con un AWS SDK o CLI](#)

- [Úselo DescribeKey con un AWS SDK o CLI](#)
- [Úselo DisableKey con un AWS SDK o CLI](#)
- [Úselo EnableKey con un AWS SDK o CLI](#)
- [Úselo EnableKeyRotation con un AWS SDK o CLI](#)
- [Úselo Encrypt con un AWS SDK o CLI](#)
- [Úselo GenerateDataKey con un AWS SDK o CLI](#)
- [Úselo GenerateDataKeyWithoutPlaintext con un AWS SDK o CLI](#)
- [Úselo GenerateRandom con un AWS SDK o CLI](#)
- [Úselo GetKeyPolicy con un AWS SDK o CLI](#)
- [Úselo ListAliases con un AWS SDK o CLI](#)
- [Úselo ListGrants con un AWS SDK o CLI](#)
- [Úselo ListKeyPolicies con un AWS SDK o CLI](#)
- [Úselo ListKeys con un AWS SDK o CLI](#)
- [Úselo PutKeyPolicy con un AWS SDK o CLI](#)
- [Úselo ReEncrypt con un AWS SDK o CLI](#)
- [Úselo RetireGrant con un AWS SDK o CLI](#)
- [Úselo RevokeGrant con un AWS SDK o CLI](#)
- [Úselo ScheduleKeyDeletion con un AWS SDK o CLI](#)
- [Úselo Sign con un AWS SDK o CLI](#)
- [Úselo TagResource con un AWS SDK o CLI](#)
- [Úselo UpdateAlias con un AWS SDK o CLI](#)
- [Úselo Verify con un AWS SDK o CLI](#)

Úselo **CreateAlias** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar CreateAlias.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

.NET

SDK para .NET

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Creates an alias for an AWS Key Management Service (AWS KMS) key.
/// </summary>
public class CreateAlias
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The alias name must start with alias/ and can be
        // up to 256 alphanumeric characters long.
        var aliasName = "alias/ExampleAlias";

        // The value supplied as the TargetKeyId can be either
        // the key ID or key Amazon Resource Name (ARN) of the
        // AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

        var request = new CreateAliasRequest
        {
            AliasName = aliasName,
            TargetKeyId = keyId,
        };

        var response = await client.CreateAliasAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
```

```
        {  
            Console.WriteLine($"Alias, {aliasName}, successfully created.");  
        }  
        else  
        {  
            Console.WriteLine($"Could not create alias.");  
        }  
    }  
}
```

- Para obtener más información sobre la API, consulta [CreateAlias](#) la Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Creación de un alias de una clave de KMS

El siguiente comando `create-alias` crea un alias llamado `example-alias` de la clave de KMS identificada por el ID de clave `1234abcd-12ab-34cd-56ef-1234567890ab`.

Los nombres del alias no debe empezar por `alias/`. No utilice nombres de alias que comiencen por `alias/aws`; están reservados para su uso exclusivo AWS.

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

Este comando no devuelve ningún resultado. Para ver el nuevo alias, utilice el comando `list-aliases`.

Para obtener más información, consulte [Uso de alias](#) en la AWS Guía para desarrolladores de Key Management Service.

- Para obtener más información sobre la API, consulte [CreateAlias](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

 Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Creates a custom alias for the specified target key asynchronously.
 *
 * @param targetKeyId the ID of the target key for the alias
 * @param aliasName the name of the alias to create
 * @return a {@link CompletableFuture} that completes when the alias creation
 * operation is finished
 */
public CompletableFuture<Void> createCustomAliasAsync(String targetKeyId,
String aliasName) {
    CreateAliasRequest aliasRequest = CreateAliasRequest.builder()
        .aliasName(aliasName)
        .targetKeyId(targetKeyId)
        .build();

    CompletableFuture<CreateAliasResponse> responseFuture =
getAsyncClient().createAlias(aliasRequest);
    responseFuture.whenComplete((response, exception) -> {
        if (exception == null) {
            logger.info("{} was successfully created.", aliasName);
        } else {
            if (exception instanceof ResourceExistsException) {
                logger.info("Alias [{}] already exists. Moving on...",
aliasName);
            } else if (exception instanceof KmsException kmsEx) {
                throw new RuntimeException("KMS error occurred while creating
alias: " + kmsEx.getMessage(), kmsEx);
            } else {
                throw new RuntimeException("An unexpected error occurred
while creating alias: " + exception.getMessage(), exception);
            }
        }
    })
}
```

```
});  
  
    return responseFuture.thenApply(response -> null);  
}
```

- Para obtener más información sobre la API, consulta [CreateAlias](#) la Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
suspend fun createCustomAlias(  
    targetKeyIdVal: String?,  
    aliasNameVal: String?,  
) {  
    val request =  
        CreateAliasRequest {  
            aliasName = aliasNameVal  
            targetKeyId = targetKeyIdVal  
        }  
  
    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->  
        kmsClient.createAlias(request)  
        println("$aliasNameVal was successfully created")  
    }  
}
```

- Para obtener más información sobre la API, consulta [CreateAlias](#) la referencia sobre el AWS SDK para la API de Kotlin.

PHP

SDK para PHP

 Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $keyId
 * @param string $alias
 * @return void
 */
public function createAlias(string $keyId, string $alias)
{
    try{
        $this->client->createAlias([
            'TargetKeyId' => $keyId,
            'AliasName' => $alias,
        ]);
    }catch (KmsException $caught){
        if($caught->getAwsErrorMessage() == "InvalidAliasNameException"){
            echo "The request was rejected because the specified alias name
is not valid.";
        }
        throw $caught;
    }
}
```

- Para obtener más información sobre la API, consulta [CreateAlias](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

 Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_alias(self, key_id: str, alias: str) -> None:
        """
        Creates an alias for the specified key.

        :param key_id: The ARN or ID of a key to give an alias.
        :param alias: The alias to assign to the key.
        """
        try:
            self.kms_client.create_alias(AliasName=alias, TargetKeyId=key_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "AlreadyExistsException":
                logger.error(
                    "Could not create the alias %s because it already exists.",
                    key_id
                )
```

```

else:
    logger.error(
        "Couldn't encrypt text. Here's why: %s",
        err.response["Error"]["Message"],
    )
    raise

```

- Para obtener más información sobre la API, consulta [CreateAlias](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    " iv_alias_name = 'alias/my-key-alias'
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    lo_kms->createalias(
        iv_aliasname = iv_alias_name
        iv_targetkeyid = iv_key_id
    ).
    MESSAGE 'Alias created successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsalreadyexistsex.
    MESSAGE 'Alias already exists.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsinvalidaliasnameex.
    MESSAGE 'Invalid alias name.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [CreateAlias](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **CreateGrant** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar CreateGrant.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

.NET

SDK para .NET

Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
public static async Task Main()
{
    var client = new AmazonKeyManagementServiceClient();

    // The identity that is given permission to perform the operations
    // specified in the grant.
    var grantee = "arn:aws:iam::111122223333:role/ExampleRole";

    // The identifier of the AWS KMS key to which the grant applies. You
    // can use the key ID or the Amazon Resource Name (ARN) of the KMS
    key.
    var keyId = "7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";

    var request = new CreateGrantRequest
```

```

        {
            GranteePrincipal = grantee,
            KeyId = keyId,

            // A list of operations that the grant allows.
            Operations = new List<string>
            {
                "Encrypt",
                "Decrypt",
            },
        };

        var response = await client.CreateGrantAsync(request);

        string grantId = response.GrantId; // The unique identifier of the
grant.
        string grantToken = response.GrantToken; // The grant token.

        Console.WriteLine($"Id: {grantId}, Token: {grantToken}");
    }
}

```

- Para obtener más información sobre la API, consulta [CreateGrant](#) la Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Creación de una concesión

En el siguiente ejemplo de `create-grant` se crea una concesión que permite al usuario `exampleUser` utilizar el comando `decrypt` de la clave de KMS del ejemplo `1234abcd-12ab-34cd-56ef-1234567890ab`. La entidad principal que se va a dar de baja es el rol `adminRole`. La concesión utiliza la limitación de concesión `EncryptionContextSubset` para permitir este permiso solo cuando el contexto de cifrado de la solicitud `decrypt` incluye un par clave-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
```

Salida:

```
{
  "GrantId":
    "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
  "GrantToken": "<grant token here>"
}
```

Para ver información detallada sobre la concesión, utilice el comando `list-grants`.

Para obtener más información, consulte [las subvenciones en AWS KMS](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte [CreateGrant](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Grants permissions to a specified principal on a customer master key (CMK)
 * asynchronously.
 *
 * @param keyId          The unique identifier for the customer master key
 *                        (CMK) that the grant applies to.
 * @param granteePrincipal The principal that is given permission to perform
 *                        the operations that the grant permits on the CMK.
 * @return A {@link CompletableFuture} that, when completed, contains the ID
 *         of the created grant.
```

```

    * @throws RuntimeException If an error occurs during the grant creation
    process.
    */
    public CompletableFuture<String> grantKeyAsync(String keyId, String
    granteePrincipal) {
        List<GrantOperation> grantPermissions = List.of(
            GrantOperation.ENCRYPT,
            GrantOperation.DECRYPT,
            GrantOperation.DESCRIBE_KEY
        );

        CreateGrantRequest grantRequest = CreateGrantRequest.builder()
            .keyId(keyId)
            .name("grant1")
            .granteePrincipal(granteePrincipal)
            .operations(grantPermissions)
            .build();

        CompletableFuture<CreateGrantResponse> responseFuture =
    getAsyncClient().createGrant(grantRequest);
        responseFuture.whenComplete((response, ex) -> {
            if (ex == null) {
                logger.info("Grant created successfully with ID: " +
    response.grantId());
            } else {
                if (ex instanceof KmsException kmsEx) {
                    throw new RuntimeException("Failed to create grant: " +
    kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
    ex.getMessage(), ex);
                }
            }
        });

        return responseFuture.thenApply(CreateGrantResponse::grantId);
    }

```

- Para obtener más información sobre la API, consulta [CreateGrant](#) la Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
suspend fun createNewGrant(
    keyIdVal: String?,
    granteePrincipalVal: String?,
    operation: String,
): String? {
    val operation0b = GrantOperation.fromValue(operation)
    val grantOperationList = ArrayList<GrantOperation>()
    grantOperationList.add(operation0b)

    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 obtener más información sobre la API, consulta [CreateGrant](#) la referencia sobre el AWS SDK para la API de Kotlin.

PHP

SDK para PHP

 Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $keyId
 * @param string $granteePrincipal
 * @param array $operations
 * @param array $grantTokens
 * @return Result
 */
public function createGrant(string $keyId, string $granteePrincipal, array
$operations, array $grantTokens = [])
{
    $args = [
        'KeyId' => $keyId,
        'GranteePrincipal' => $granteePrincipal,
        'Operations' => $operations,
    ];
    if($grantTokens){
        $args['GrantTokens'] = $grantTokens;
    }
    try{
        return $this->client->createGrant($args);
    }catch(KmsException $caught){
        if($caught->getAwsErrorMessage() == "InvalidGrantTokenException"){
            echo "The request was rejected because the specified grant token
is not valid.\n";
        }
        throw $caught;
    }
}
```

- Para obtener más información sobre la API, consulta [CreateGrant](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_grant(
        self, key_id: str, principal: str, operations: [str]
    ) -> dict[str, str]:
        """
        Creates a grant for a key that lets a principal generate a symmetric data
encryption key.

        :param key_id: The ARN or ID of the key.
        :param principal: The principal to grant permission to.
        :param operations: The operations to grant permission for.
        :return: The grant that is created.
        """
```

```

try:
    return self.kms_client.create_grant(
        KeyId=key_id,
        GranteePrincipal=principal,
        Operations=operations,
    )
except ClientError as err:
    logger.error(
        "Couldn't create a grant on key %s. Here's why: %s",
        key_id,
        err.response["Error"]["Message"],
    )
    raise

```

- Para obtener más información sobre la API, consulta [CreateGrant](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    " iv_grantee_principal = 'arn:aws:iam::123456789012:role/my-role'
    " it_operations contains 'Encrypt', 'Decrypt', 'GenerateDataKey'
    oo_result = lo_kms->creategrant(
        iv_keyid = iv_key_id
        iv_granteeprincipal = iv_grantee_principal
        it_operations = it_operations
    ).
    MESSAGE 'Grant created successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.

```

```
MESSAGE 'The key is disabled.' TYPE 'E'.  
CATCH /aws1/cx_kmsnotfoundexception.  
MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [CreateGrant](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **CreateKey** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar CreateKey.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en los siguientes ejemplos de código:

- [Conceptos básicos](#)
- [Trabajo con el cifrado de tablas](#)

.NET

SDK para .NET

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
using System;  
using System.Threading.Tasks;  
using Amazon.KeyManagementService;  
using Amazon.KeyManagementService.Model;
```

```

    /// <summary>
    /// Shows how to create a new AWS Key Management Service (AWS KMS)
    /// key.
    /// </summary>
    public class CreateKey
    {
        public static async Task Main()
        {
            // Note that if you need to create a Key in an AWS Region
            // other than the Region defined for the default user, you need to
            // pass the Region to the client constructor.
            var client = new AmazonKeyManagementServiceClient();

            // The call to CreateKeyAsync will create a symmetrical AWS KMS
            // key. For more information about symmetrical and asymmetrical
            // keys, see:
            //
            // https://docs.aws.amazon.com/kms/latest/developerguide/symm-asymm-
            choose.html
            var response = await client.CreateKeyAsync(new CreateKeyRequest());

            // The KeyMetadata object contains information about the new AWS KMS
            key.
            KeyMetadata keyMetadata = response.KeyMetadata;

            if (keyMetadata is not null)
            {
                Console.WriteLine($"KMS Key: {keyMetadata.KeyId} was successfully
                created.");
            }
            else
            {
                Console.WriteLine("Could not create KMS Key.");
            }
        }
    }
}

```

- Para obtener más información sobre la API, consulta [CreateKey](#) la Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Ejemplo 1: Para crear una clave de KMS gestionada por el cliente en AWS KMS

En el siguiente ejemplo `create-key` se crea una clave de KMS de cifrado simétrico.

Para crear la clave de KMS básica, una clave de cifrado simétrica, no es necesario especificar ningún parámetro. Los valores predeterminados de esos parámetros crean una clave de cifrado simétrica.

Puesto que este comando no especifica una política de claves, la clave de KMS obtiene la [política de claves predeterminada](#) para las claves de KMS creadas mediante programación. Utilice el comando `get-key-policy` para ver la política de claves. Utilice el comando `put-key-policy` para cambiar la política de claves.

```
aws kms create-key
```

El comando `create-key` devuelve los metadatos de la clave, incluidos el ID de la clave y el ARN de la nueva clave de KMS. Puede usar estos valores para identificar la clave de KMS en otras operaciones de AWS KMS. La salida no incluye las etiquetas. Para ver las etiquetas de una clave de KMS, utilice `list-resource-tags` command.

Salida:

```
{
  "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"
        ]
    }
}

```

Nota: El comando `create-key` no permite especificar un alias. Para crear un alias para la nueva clave de KMS, utilice el comando `create-alias`.

Para obtener más información, consulte [Creación de claves](#) en la Guía para desarrolladores de AWS Key Management Service.

Ejemplo 2: Creación de una clave KMS de RSA asimétrica para el cifrado y el descifrado

El siguiente ejemplo de `create-key` se crea una clave de KMS que incluye un par de claves asimétricas RSA para el cifrado y el descifrado. No se puede cambiar la especificación de clave y el uso de la clave después de crear la clave:

```

aws kms create-key \
  --key-spec RSA_4096 \
  --key-usage ENCRYPT_DECRYPT

```

Salida:

```

{
  "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 obtener más información, consulte [Claves asimétricas en AWS KMS en](#) la Guía para desarrolladores del Servicio de administración de AWS claves.

Ejemplo 3: creación de una clave de KMS asimétrica de curva elíptica para la firma y la verificación

Creación de una clave de KMS asimétrica que contenga un par de claves asimétricas de curva elíptica (ECC) para la firma y la verificación. El parámetro `--key-usage` es obligatorio aunque `SIGN_VERIFY` sea el único valor válido para las claves KMS de ECC. No se puede cambiar la especificación de clave y el uso de la clave después de crear la clave:

```

aws kms create-key \
  --key-spec ECC_NIST_P521 \
  --key-usage SIGN_VERIFY

```

Salida:

```

{
  "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 obtener más información, consulte [Claves asimétricas en AWS KMS en](#) la Guía para desarrolladores del Servicio de administración de AWS claves.

Ejemplo 4: Creación de una clave de KMS ML-DSA asimétrica para la firma y la verificación

En este ejemplo, se crea una clave de algoritmo de firma digital de module-lattice (ML-DSA) para la firma y la verificación. El parámetro de clave de uso es obligatorio aunque SIGN_VERIFY sea el único valor válido para las claves de ML-DSA.

```
aws kms create-key \  
  --key-spec ML_DSA_65 \  
  --key-usage SIGN_VERIFY
```

Salida:

```
{  
  "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 obtener más información, consulte [Claves asimétricas en AWS KMS en](#) la Guía para desarrolladores del Servicio de administración de AWS claves.

Ejemplo 5: Creación de una clave de KMS HMAC

En el siguiente ejemplo de `create-key` se crea una clave de KMS HMAC de 384 bits. El valor `GENERATE_VERIFY_MAC` del parámetro `--key-usage` es obligatorio aunque sea el único valor válido para las claves KMS HMAC.

```
aws kms create-key \  
  --key-spec HMAC_384 \  
  --key-usage GENERATE_VERIFY_MAC
```

Salida:

```
{  
  "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 obtener más información, consulte [las claves HMAC en AWS KMS en](#) la Guía para desarrolladores del Servicio de administración de AWS claves.

Ejemplo 6: Creación de una clave de KMS principal de varias regiones

En el siguiente ejemplo de `create-key` se crea una clave principal de cifrado simétrica de varias regiones. Puesto que los valores predeterminados de todos los parámetros crean una clave de cifrado simétrica, solo es necesario el parámetro `--multi-region` para esta clave de KMS. En la AWS CLI, para indicar que un parámetro booleano es verdadero, simplemente especifique el nombre del parámetro.

```
aws kms create-key \  
  --multi-region
```

Salida:

```
{  
  "KeyMetadata": {  
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/  
mrk-1234abcd12ab34cd56ef12345678990ab",  
    "AWSAccountId": "111122223333",  
    "CreationDate": "2021-09-02T016:15:21-09:00",  
    "CurrentKeyMaterialId":  
    "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",  
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",  
    "Description": "",  
    "Enabled": true,  
    "EncryptionAlgorithms": [  
      "SYMMETRIC_DEFAULT"  
    ],  
    "KeyId": "mrk-1234abcd12ab34cd56ef12345678990ab",  
    "KeyManager": "CUSTOMER",  
    "KeySpec": "SYMMETRIC_DEFAULT",  
    "KeyState": "Enabled",  
    "KeyUsage": "ENCRYPT_DECRYPT",  
    "MultiRegion": true,  
    "MultiRegionConfiguration": {  
      "MultiRegionKeyType": "PRIMARY",  
      "PrimaryKey": {  
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/  
mrk-1234abcd12ab34cd56ef12345678990ab",  
        "Region": "us-west-2"  
      },  
      "ReplicaKeys": []  
    },  
    "Origin": "AWS_KMS"  
  }  
}
```

```
}
```

Para obtener más información, consulte [Claves asimétricas en AWS KMS en](#) la Guía para desarrolladores del servicio de administración de AWS claves.

Ejemplo 7: Creación de una clave de KMS para material de claves importado

En el siguiente ejemplo de `create-key` se crea una clave de KMS sin material de claves. Cuando se complete la operación, puede importar su propio material de claves en la clave de KMS. Para crear esta clave de KMS, establezca el parámetro `--origin` en `EXTERNAL`.

```
aws kms create-key \  
  --origin EXTERNAL
```

Salida:

```
{  
  "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 obtener más información, consulte [Importación de material clave en claves de AWS KMS](#) en la Guía para desarrolladores del Servicio de administración de AWS claves.

Ejemplo 6: Para crear una clave KMS en un almacén de claves de AWS CloudHSM

En el siguiente `create-key` ejemplo, se crea una clave KMS en el almacén de claves de AWS CloudHSM especificado. La operación crea la clave de KMS y sus metadatos en AWS KMS y crea el material de claves en el clúster de AWS CloudHSM asociado al almacén de claves personalizado. Los parámetros `--custom-key-store-id` y `--origin` son obligatorios.

```
aws kms create-key \  
  --origin AWS_CLOUDHSM \  
  --custom-key-store-id cks-1234567890abcdef0
```

Salida:

```
{  
  "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 obtener más información, consulte [Almacenes de claves de AWS CloudHSM](#) en la Guía para desarrolladores de AWS Key Management Service.

Ejemplo 8: Creación de una clave de KMS en un almacén de claves externo

En el siguiente ejemplo de `create-key` se crea una clave de KMS en el almacén de claves externo especificado. Los parámetros `--custom-key-store-id`, `--origin` y `--xks-key-id` son obligatorios en este comando.

El parámetro `--xks-key-id` especifica el ID de una clave de cifrado simétrica existente en el administrador de claves externo. Esta clave sirve como material de clave externa para la clave de KMS. El valor del parámetro `--origin` debe ser `EXTERNAL_KEY_STORE`. El parámetro `custom-key-store-id` debe identificar un almacén de claves externo que esté conectado a su proxy de almacén de claves externo.

```
aws kms create-key \  
  --origin EXTERNAL_KEY_STORE \  
  --custom-key-store-id cks-9876543210fedcba9 \  
  --xks-key-id bb8562717f809024
```

Salida:

```
{  
  "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": "ckd-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 obtener más información, consulte [Almacenes de claves externos](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte la Referencia [CreateKey](#) de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Creates a new symmetric encryption key asynchronously.
 *
 * @param keyDesc the description of the key to be created
 * @return a {@link CompletableFuture} that completes with the ID of the
 * newly created key
 * @throws RuntimeException if an error occurs while creating the key
 */
public CompletableFuture<String> createKeyAsync(String keyDesc) {
    CreateKeyRequest keyRequest = CreateKeyRequest.builder()
        .description(keyDesc)
        .keySpec(KeySpec.SYMMETRIC_DEFAULT)
        .keyUsage(KeyUsageType.ENCRYPT_DECRYPT)
        .build();

    return getAsyncClient().createKey(keyRequest)
        .thenApply(resp -> resp.keyMetadata().keyId())
        .exceptionally(ex -> {
            throw new RuntimeException("An error occurred while creating the
key: " + ex.getMessage(), ex);
        });
}
```

- Para obtener más información sobre la API, consulta [CreateKey](#) la Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
suspend fun createKey(keyDesc: String?): String? {
    val request =
        CreateKeyRequest {
            description = keyDesc
            customerMasterKeySpec = CustomerMasterKeySpec.SymmetricDefault
            keyUsage = KeyUsageType.fromValue("ENCRYPT_DECRYPT")
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val result = kmsClient.createKey(request)
        println("Created a customer key with id " + result.keyMetadata?.arn)
        return result.keyMetadata?.keyId
    }
}
```

- Para obtener más información sobre la API, consulta [CreateKey](#) la referencia sobre el AWS SDK para la API de Kotlin.

PHP

SDK para PHP

 Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $keySpec
 * @param string $keyUsage
 * @param string $description
 * @return array
 */
public function createKey(string $keySpec = "", string $keyUsage = "", string
 $description = "Created by the SDK for PHP")
{
    $parameters = ['Description' => $description];
    if($keySpec && $keyUsage){
        $parameters['KeySpec'] = $keySpec;
        $parameters['KeyUsage'] = $keyUsage;
    }
    try {
        $result = $this->client->createKey($parameters);
        return $result['KeyMetadata'];
    }catch(KmsException $caught){
        // Check for error specific to createKey operations
        if ($caught->getAwsErrorMessage() == "LimitExceededException"){
            echo "The request was rejected because a quota was exceeded. For
 more information, see Quotas in the Key Management Service Developer Guide.";
        }
        throw $caught;
    }
}
```

- Para obtener más información sobre la API, consulta [CreateKey](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_key(self, key_description: str) -> dict[str, any]:
        """
        Creates a key with a user-provided description.

        :param key_description: A description for the key.
        :return: The key ID.
        """
        try:
            key = self.kms_client.create_key(Description=key_description)
            ["KeyMetadata"]
            self.created_keys.append(key)
            return key
        except ClientError as err:
            logging.error(
                "Couldn't create your key. Here's why: %s",
```

```
        err.response["Error"]["Message"],
    )
    raise
```

- Para obtener más información sobre la API, consulta [CreateKey](#) la AWS Referencia de API de SDK for Python (Boto3).

Ruby

SDK para Ruby

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Create a AWS KMS key.
# As long we are only encrypting small amounts of data (4 KiB or less) directly,
# a KMS key is fine for our purposes.
# For larger amounts of data,
# use the KMS key to encrypt a data encryption key (DEK).

client = Aws::KMS::Client.new

resp = client.create_key({
  tags: [
    {
      tag_key: 'CreatedBy',
      tag_value: 'ExampleUser'
    }
  ]
})

puts resp.key_metadata.key_id
```

- Para obtener más información sobre la API, consulta [CreateKey](#) la Referencia AWS SDK para Ruby de la API.

Rust

SDK para Rust

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
async fn make_key(client: &Client) -> Result<(), Error> {  
    let resp = client.create_key().send().await?;  
  
    let id = resp.key_metadata.as_ref().unwrap().key_id();  
  
    println!("Key: {}", id);  
  
    Ok(())  
}
```

- Para obtener más información sobre la API, consulta [CreateKey](#) la referencia sobre la API de AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    " iv_description = 'Created by the AWS SDK for SAP ABAP'
```

```
oo_result = lo_kms->createkey( iv_description = iv_description ).
MESSAGE 'KMS key created successfully.' TYPE 'I'.
CATCH /aws1/cx_kmskmsinternalex.
MESSAGE 'An internal error occurred.' TYPE 'E'.
CATCH /aws1/cx_kmslimitexceedex.
MESSAGE 'Limit exceeded for KMS resources.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [CreateKey](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **Decrypt** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar Decrypt.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

CLI

AWS CLI

Ejemplo 1: Descifrado de un mensaje cifrado con una clave de KMS simétrica (Linux y macOS)

El siguiente ejemplo de decrypt comando muestra la forma recomendada de descifrar datos con la AWS CLI. Esta versión muestra cómo descifrar los datos en una clave de KMS simétrica.

Proporcione el texto cifrado de un archivo. En el valor del parámetro `--ciphertext-blob`, utilice el prefijo `fileb://`, que indica a la CLI que lea los datos de un archivo binario. Si el archivo no se encuentra en el directorio actual, escriba la ruta completa al archivo. Para obtener más información sobre la lectura de los valores de los parámetros AWS

CLI de un archivo, consulte [Carga de parámetros AWS CLI desde un archivo < https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html>](https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html) en la Guía del usuario de la interfaz de línea de AWS comandos y Prácticas recomendadas para los parámetros de archivos locales< [https://aws.amazon.com/blogs/ developer/ best-practices-for-local -file-parameters/>](https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/) en el blog de herramientas de línea de AWS comandos. Especifique la clave KMS para descifrar el texto cifrado. El parámetro no es necesario cuando se descifra con una clave KMS simétrica. --key-id AWS KMS puede obtener el identificador de clave de la clave KMS que se utilizó para cifrar los datos de los metadatos del texto cifrado. Pero siempre es una práctica recomendada especificar la clave KMS que está utilizando. Esta práctica garantiza que utilice la clave de KMS pretendida y le impide descifrar inadvertidamente un texto cifrado mediante una clave de KMS en la que no confía. Solicite la salida de texto sin formato como valor de texto. El parámetro --query indica a la CLI que obtenga únicamente el valor del campo Plaintext de la salida. El parámetro --output devuelve la salida como texto. Decodifique en base64 el texto sin formato y guárdelo en un archivo. El siguiente ejemplo canaliza (|) el valor del parámetro Plaintext a la utilidad Base64, que lo decodifica. A continuación, redirige (>) la salida descodificada al archivo ExamplePlaintext.

Antes de ejecutar este comando, sustituya el identificador de clave de ejemplo por un identificador de clave válido de su cuenta. AWS

```
aws kms decrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --output text \
  --query Plaintext | base64 \
  --decode > ExamplePlaintextFile
```

Este comando no genera ninguna salida. La salida del comando decrypt se descodifica en base64 y se guarda en un archivo.

Para obtener más información, consulte [Decrypt](#) en la Referencia de la API de AWS Key Management Service.

Ejemplo 2: Descifrado de un mensaje cifrado con una clave de KMS simétrica (símbolo del sistema de Windows)

El ejemplo siguiente es el mismo que el anterior, excepto que usa la utilidad `certutil` para descodificar en Base64 los datos de texto sin formato. Como se muestra en los siguientes ejemplos, este procedimiento requiere dos comandos.

Antes de ejecutar este comando, sustituya el identificador de clave de ejemplo por un identificador de clave válido de su AWS cuenta.

```
aws kms decrypt ^  
  --ciphertext-blob fileb://ExampleEncryptedFile ^  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^  
  --output text ^  
  --query Plaintext > ExamplePlaintextFile.base64
```

Ejecute el comando `certutil`.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Salida:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

Para obtener más información, consulte [Descifrado](#) en la AWS Referencia de la API de Key Management Service.

Ejemplo 3: descifrado de un mensaje cifrado con una clave de KMS asimétrica (Linux y macOS)

El siguiente comando `decrypt` de ejemplo muestra cómo descifrar los datos cifrados con una clave KMS de RSA asimétrica.

Cuando se utiliza una clave de KMS asimétrica, es obligatorio el parámetro `encryption-algorithm`, que especifica el algoritmo que se utiliza para cifrar el texto sin formato.

Antes de ejecutar este comando, sustituya el identificador de clave de ejemplo por un identificador de clave válido de su AWS cuenta.

```
aws kms decrypt \  
  --ciphertext-blob fileb://ExampleEncryptedFile \  
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
  --encryption-algorithm RSASOAES \  
  --output text \  
  --query Plaintext > ExamplePlaintextFile.base64
```

```
--encryption-algorithm RSAES_OAEP_SHA_256 \
--output text \
--query Plaintext | base64 \
--decode > ExamplePlaintextFile
```

Este comando no genera ninguna salida. La salida del comando `decrypt` se descodifica en base64 y se guarda en un archivo.

Para obtener más información, consulte [Claves asimétricas en AWS KMS en la Guía para desarrolladores del servicio de administración de AWS claves](#).

- Para obtener información sobre la API, consulte [Decrypt](#) en la Referencia de comandos de la AWS CLI .

Java

SDK para Java 2.x

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Asynchronously decrypts the given encrypted data using the specified key
 * ID.
 *
 * @param encryptedData The encrypted data to be decrypted.
 * @param keyId The ID of the key to be used for decryption.
 * @return A CompletableFuture that, when completed, will contain the
 *         decrypted data as a String.
 *
 *         If an error occurs during the decryption process, the
 *         CompletableFuture will complete
 *
 *         exceptionally with the error, and the method will return an empty
 *         String.
 */
public CompletableFuture<String> decryptDataAsync(SdkBytes encryptedData,
String keyId) {
    DecryptRequest decryptRequest = DecryptRequest.builder()
        .ciphertextBlob(encryptedData)
        .keyId(keyId)
```



```

        .build();

        CompletableFuture<DecryptResponse> responseFuture =
getAsyncClient().decrypt(decryptRequest);
        responseFuture.whenComplete((decryptResponse, exception) -> {
            if (exception == null) {
                logger.info("Data decrypted successfully for key ID: " + keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while
decrypting data: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while decrypting data: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(decryptResponse ->
decryptResponse.plaintext().asString(StandardCharsets.UTF_8));
    }

```

- Para obtener información sobre la API, consulte [Decrypt](#) en la Referencia de la API de AWS SDK for Java 2.x .

Kotlin

SDK para Kotlin

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

suspend fun encryptData(keyIdValue: String): ByteArray? {
    val text = "This is the text to encrypt by using the AWS KMS Service"
    val myBytes: ByteArray = text.toByteArray()

    val encryptRequest =

```

```
        EncryptRequest {
            keyId = keyIdValue
            plaintext = myBytes
        }

        KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
            val response = kmsClient.encrypt(encryptRequest)
            val algorithm: String = response.encryptionAlgorithm.toString()
            println("The encryption algorithm is $algorithm")

            // Return the encrypted data.
            return response.ciphertextBlob
        }
    }

suspend fun decryptData(
    encryptedDataVal: ByteArray?,
    keyIdVal: String?,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
        val myVal = decryptResponse.plaintext

        // Print the decrypted data.
        print(myVal)
    }
}
```

- Para obtener información sobre la API, consulte [Decrypt](#) en la Referencia de la API de AWS SDK para Kotlin.

PHP

SDK para PHP

 Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $keyId
 * @param string $ciphertext
 * @param string $algorithm
 * @return Result
 */
public function decrypt(string $keyId, string $ciphertext, string $algorithm
= "SYMMETRIC_DEFAULT")
{
    try{
        return $this->client->decrypt([
            'CiphertextBlob' => $ciphertext,
            'EncryptionAlgorithm' => $algorithm,
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
        echo "There was a problem decrypting the data: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}
```

- Para obtener información sobre la API, consulte [Decrypt](#) en la Referencia de la API de AWS SDK para PHP .

Python

SDK para Python (Boto3)

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def decrypt(self, key_id: str, cipher_text: bytes) -> str:
        """
        Decrypts text previously encrypted with a key.

        :param key_id: The ARN or ID of the key used to decrypt the data.
        :param cipher_text: The encrypted text to decrypt.
        :return: The decrypted text.
        """
        try:
            return self.kms_client.decrypt(KeyId=key_id,
CiphertextBlob=cipher_text)[
                "Plaintext"
            ].decode()
        except ClientError as err:
            logger.error(
                "Couldn't decrypt your ciphertext. Here's why: %s",
```

```
        err.response["Error"]["Message"],
    )
    raise
```

- Para obtener información sobre la API, consulte [Decrypt](#) en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Decrypted blob

blob =
  '01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593'
blob_packed = [blob].pack('H*')

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.decrypt({
  ciphertext_blob: blob_packed
})

puts 'Raw text: '
puts resp.plaintext
```

- Para obtener información sobre la API, consulte [Decrypt](#) en la Referencia de la API de AWS SDK para Ruby .

Rust

SDK para Rust

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
async fn decrypt_key(client: &Client, key: &str, filename: &str) -> Result<(),
Error> {
    // Open input text file and get contents as a string
    // input is a base-64 encoded string, so decode it:
    let data = fs::read_to_string(filename)
        .map(|input| {
            base64::decode(input).expect("Input file does not contain valid base
64 characters.")
        })
        .map(Blob::new);

    let resp = client
        .decrypt()
        .key_id(key)
        .ciphertext_blob(data.unwrap())
        .send()
        .await?;

    let inner = resp.plaintext.unwrap();
    let bytes = inner.as_ref();

    let s = String::from_utf8(bytes.to_vec()).expect("Could not convert to
UTF-8");

    println();
    println!("Decoded string:");
    println!("{}", s);

    Ok(())
}
```

- Para obtener información sobre la API, consulte [Decrypt](#) en la Referencia de la API de AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    " iv_ciphertext_blob contains the encrypted data  
    oo_result = lo_kms->decrypt(  
        iv_keyid = iv_key_id  
        iv_ciphertextblob = iv_ciphertext_blob  
    ).  
    MESSAGE 'Text decrypted successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsdisabledexception.  
    MESSAGE 'The key is disabled.' TYPE 'E'.  
CATCH /aws1/cx_kmsincorrectkeyex.  
    MESSAGE 'Incorrect key for decryption.' TYPE 'E'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulta [Cómo descifrar](#) en el AWS SDK para obtener información sobre la API ABAP de SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. [Uso de este servicio con un AWS SDK](#) En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **DeleteAlias** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar DeleteAlias.

CLI

AWS CLI

Para eliminar un alias de AWS KMS

En el siguiente ejemplo de delete-alias se elimina el alias alias/example-alias. El nombre de alias debe empezar por alias/.

```
aws kms delete-alias \  
  --alias-name alias/example-alias
```

Este comando no genera ninguna salida. Utilice el comando list-aliases para encontrar el alias.

Para obtener más información, consulte [Eliminación de un alias](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener información sobre la API, consulte [DeleteAlias](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**  
 * Deletes a specific KMS alias asynchronously.  
 *  
 * @param aliasName the name of the alias to be deleted  
 * @return a {@link CompletableFuture} representing the asynchronous  
 operation of deleting the specified alias
```



```
*/
public CompletableFuture<Void> deleteSpecificAliasAsync(String aliasName) {
    DeleteAliasRequest deleteAliasRequest = DeleteAliasRequest.builder()
        .aliasName(aliasName)
        .build();

    return getAsyncClient().deleteAlias(deleteAliasRequest)
        .thenRun(() -> {
            logger.info("Alias {} has been deleted successfully", aliasName);
        })
        .exceptionally(throwable -> {
            throw new RuntimeException("Failed to delete alias: " +
aliasName, throwable);
        });
}
```

- Para obtener más información sobre la API, consulta [DeleteAlias](#) la Referencia AWS SDK for Java 2.x de la API.

PHP

SDK para PHP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $aliasName
 * @return void
 */
public function deleteAlias(string $aliasName)
{
    try {
        $this->client->deleteAlias([
            'AliasName' => $aliasName,
        ]);
    }
```

```

        }catch(KmsException $caught){
            echo "There was a problem deleting the alias: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
}

```

- Para obtener más información sobre la API, consulta [DeleteAlias](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def delete_alias(self, alias: str) -> None:
        """

```

```

Deletes an alias.

:param alias: The alias to delete.
"""
try:
    self.kms_client.delete_alias(AliasName=alias)
except ClientError as err:
    logger.error(
        "Couldn't delete alias %s. Here's why: %s",
        alias,
        err.response["Error"]["Message"],
    )
    raise

```

- Para obtener más información sobre la API, consulta [DeleteAlias](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    " iv_alias_name = 'alias/my-key-alias'
    lo_kms->deletealias( iv_aliasname = iv_alias_name ).
    MESSAGE 'Alias deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Alias not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [DeleteAlias](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **DescribeKey** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar DescribeKey.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

.NET

SDK para .NET

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Retrieve information about an AWS Key Management Service (AWS KMS) key.
/// You can supply either the key Id or the key Amazon Resource Name (ARN)
/// to the DescribeKeyRequest KeyId property.
/// </summary>
public class DescribeKey
{
    public static async Task Main()
    {
```

```
var keyId = "7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
var request = new DescribeKeyRequest
{
    KeyId = keyId,
};

var client = new AmazonKeyManagementServiceClient();

var response = await client.DescribeKeyAsync(request);
var metadata = response.KeyMetadata;

    Console.WriteLine($"{metadata.KeyId} created on:
{metadata.CreationDate}");
    Console.WriteLine($"State: {metadata.KeyState}");
    Console.WriteLine($"{metadata.Description}");
}
}
```

- Para obtener más información sobre la API, consulta [DescribeKey](#) la Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Ejemplo 1: Búsqueda de información detallada sobre una clave de KMS

En el siguiente `describe-key` ejemplo, se obtiene información detallada sobre la clave AWS gestionada de Amazon S3 en la cuenta y la región de ejemplo. Puede usar este comando para obtener detalles sobre las claves AWS administradas y las claves administradas por el cliente.

Para especificar la clave de KMS, utilice el parámetro `key-id`. En este ejemplo, se utiliza un valor de nombre de alias, pero puede utilizar un ID de clave, un ARN de clave, un nombre de alias o un ARN de alias en este comando.

```
aws kms describe-key \
    --key-id alias/aws/s3
```

Salida:

```
{
  "KeyMetadata": {
    "AWSAccountId": "846764612917",
    "KeyId": "b8a9477d-836c-491f-857e-07937918959b",
    "Arn": "arn:aws:kms:us-west-2:846764612917:key/
b8a9477d-836c-491f-857e-07937918959b",
    "CurrentKeyMaterialId":
"0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "CreationDate": 2017-06-30T21:44:32.140000+00:00,
    "Enabled": true,
    "Description": "Default KMS key that protects my S3 objects when no other
key is defined",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "KeyState": "Enabled",
    "Origin": "AWS_KMS",
    "KeyManager": "AWS",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ]
  }
}
```

Para obtener más información, consulte [Visualización de claves](#) en la Guía para desarrolladores de AWS Key Management Service.

Ejemplo 2: Obtención de detalles sobre una clave KMS de RSA asimétrica

En el siguiente ejemplo de `describe-key` se obtiene información detallada acerca de una clave KMS de RSA asimétrica que se utiliza para la firma y la verificación.

```
aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Salida:

```
{
  "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"
    ]
}

```

Ejemplo 3: obtención de detalles sobre una clave de réplica de varias regiones

En el siguiente ejemplo de `describe-key` se obtienen los metadatos de una clave de réplica de varias regiones. Esta clave de varias regiones es una clave de cifrado simétrica. La salida de un comando `describe-key` para cualquier clave de varias regiones devuelve información sobre la clave principal y todas sus réplicas.

```

aws kms describe-key \
  --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab

```

Salida:

```

{
  "KeyMetadata": {
    "MultiRegion": true,
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "CreationDate": "2021-06-28T21:09:16.114000+00:00",

```

```

    "CurrentKeyMaterialId":
    "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "Description": "",
    "Enabled": true,
    "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "Origin": "AWS_KMS",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ],
    "MultiRegionConfiguration": {
        "MultiRegionKeyType": "PRIMARY",
        "PrimaryKey": {
            "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
            "Region": "us-west-2"
        },
        "ReplicaKeys": [
            {
                "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "eu-west-1"
            },
            {
                "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "ap-northeast-1"
            },
            {
                "Arn": "arn:aws:kms:sa-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "sa-east-1"
            }
        ]
    }
}

```

Ejemplo 4: obtención de detalles sobre una clave HMAC KMS

En el siguiente ejemplo de `describe-key` se obtiene información detallada acerca de una clave KMS de HMAC.

```
aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Salida:

```
{
  "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 obtener más información sobre la API, consulte [DescribeKey](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * Asynchronously checks if a specified key is enabled.
 *
 * @param keyId the ID of the key to check
 * @return a {@link CompletableFuture} that, when completed, indicates
whether the key is enabled or not
 *
 * @throws RuntimeException if an exception occurs while checking the key
state
 */
public CompletableFuture<Boolean> isKeyEnabledAsync(String keyId) {
    DescribeKeyRequest keyRequest = DescribeKeyRequest.builder()
        .keyId(keyId)
        .build();

    CompletableFuture<DescribeKeyResponse> responseFuture =
getAsyncClient().describeKey(keyRequest);
    return responseFuture.whenComplete((resp, ex) -> {
        if (resp != null) {
            KeyState keyState = resp.keyMetadata().keyState();
            if (keyState == KeyState.ENABLED) {
                logger.info("The key is enabled.");
            } else {
                logger.info("The key is not enabled. Key state: {}",
keyState);
            }
        } else {
            throw new RuntimeException(ex);
        }
    }).thenApply(resp -> resp.keyMetadata().keyState() == KeyState.ENABLED);
}

```

- Para obtener más información sobre la API, consulta [DescribeKey](#) la Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
suspend fun describeSpecifcKey(keyIdVal: String?) {  
    val request =  
        DescribeKeyRequest {  
            keyId = keyIdVal  
        }  
  
    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->  
        val response = kmsClient.describeKey(request)  
        println("The key description is ${response.keyMetadata?.description}")  
        println("The key ARN is ${response.keyMetadata?.arn}")  
    }  
}
```

- Para obtener más información sobre la API, consulta [DescribeKey](#) la referencia sobre el AWS SDK para la API de Kotlin.

PHP

SDK para PHP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**  
 * @param string $keyId
```

```

    * @return array
    */
    public function describeKey(string $keyId)
    {
        try {
            $result = $this->client->describeKey([
                "KeyId" => $keyId,
            ]);
            return $result['KeyMetadata'];
        } catch (KmsException $caught) {
            if ($caught->getAwsErrorMessage() == "NotFoundException") {
                echo "The request was rejected because the specified entity or
resource could not be found.\n";
            }
            throw $caught;
        }
    }
}

```

- Para obtener más información sobre la API, consulta [DescribeKey](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """

```

```
Creates a KeyManager instance with a default KMS client.

:return: An instance of KeyManager initialized with the default KMS
client.
"""
kms_client = boto3.client("kms")
return cls(kms_client)

def describe_key(self, key_id: str) -> dict[str, any]:
    """
    Describes a key.

    :param key_id: The ARN or ID of the key to describe.
    :return: Information about the key.
    """

    try:
        key = self.kms_client.describe_key(KeyId=key_id)["KeyMetadata"]
        return key
    except ClientError as err:
        logging.error(
            "Couldn't get key '%s'. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise
```

- Para obtener más información sobre la API, consulta [DescribeKey](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    oo_result = lo_kms->describekey( iv_keyid = iv_key_id ).  
    DATA(lo_key) = oo_result->get_keymetadata( ).  
    MESSAGE 'Retrieved key information successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [DescribeKey](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **DisableKey** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `DisableKey`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

.NET

SDK para .NET

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
using System;
```

```
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Disable an AWS Key Management Service (AWS KMS) key and then retrieve
/// the key's status to show that it has been disabled.
/// </summary>
public class DisableKey
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The identifier of the AWS KMS key to disable. You can use the
        // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

        var request = new DisableKeyRequest
        {
            KeyId = keyId,
        };

        var response = await client.DisableKeyAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            // Retrieve information about the key to show that it has now
            // been disabled.
            var describeResponse = await client.DescribeKeyAsync(new
DescribeKeyRequest
            {
                KeyId = keyId,
            });
            Console.WriteLine($"{describeResponse.KeyMetadata.KeyId} - state:
{describeResponse.KeyMetadata.KeyState}");
        }
    }
}
```

- Para obtener más información sobre la API, consulta [DisableKey](#) la Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Desactivación temporal de una clave de KMS

El siguiente comando `disable-key` desactiva una clave de KMS administrada por el cliente. Para volver a habilitar la clave de KMS, utilice el comando `enable-key`.

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

Este comando no genera ninguna salida.

Para obtener más información, consulte [Habilitar y deshabilitar claves](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulta [DisableKey](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Asynchronously disables the specified AWS Key Management Service (KMS)
 * key.
 *
 * @param keyId the ID or Amazon Resource Name (ARN) of the KMS key to be
 * disabled
 * @return a CompletableFuture that, when completed, indicates that the key
 * has been disabled successfully
```



```
*/
public CompletableFuture<Void> disableKeyAsync(String keyId) {
    DisableKeyRequest keyRequest = DisableKeyRequest.builder()
        .keyId(keyId)
        .build();

    return getAsyncClient().disableKey(keyRequest)
        .thenRun(() -> {
            logger.info("Key {} has been disabled successfully",keyId);
        })
        .exceptionally(throwable -> {
            throw new RuntimeException("Failed to disable key: " + keyId,
throwable);
        });
}
```

- Para obtener más información sobre la API, consulta [DisableKey](#) la Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
suspend fun disableKey(keyIdVal: String?) {
    val request =
        DisableKeyRequest {
            keyId = keyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        kmsClient.disableKey(request)
        println("$keyIdVal was successfully disabled")
    }
}
```

- Para obtener más información sobre la API, consulta [DisableKey](#) la referencia sobre el AWS SDK para la API de Kotlin.

PHP

SDK para PHP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $keyId
 * @return void
 */
public function disableKey(string $keyId)
{
    try {
        $this->client->disableKey([
            'KeyId' => $keyId,
        ]);
    } catch (KmsException $caught){
        echo "There was a problem disabling the key: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}
```

- Para obtener más información sobre la API, consulta [DisableKey](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def disable_key(self, key_id: str) -> None:
        try:
            self.kms_client.disable_key(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't disable key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Para obtener más información sobre la API, consulta [DisableKey](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    lo_kms->disablekey( iv_keyid = iv_key_id ).  
    MESSAGE 'KMS key disabled successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [DisableKey](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **EnableKey** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar EnableKey.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

.NET

SDK para .NET

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Enable an AWS Key Management Service (AWS KMS) key.
/// </summary>
public class EnableKey
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The identifier of the AWS KMS key to enable. You can use the
        // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

        var request = new EnableKeyRequest
        {
            KeyId = keyId,
        };

        var response = await client.EnableKeyAsync(request);
        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            // Retrieve information about the key to show that it has now
            // been enabled.
            var describeResponse = await client.DescribeKeyAsync(new
DescribeKeyRequest
            {
                KeyId = keyId,
```

```
        });  
        Console.WriteLine($"{describeResponse.KeyMetadata.KeyId} - state:  
{describeResponse.KeyMetadata.KeyState}");  
    }  
}
```

- Para obtener más información sobre la API, consulta [EnableKey](#) la Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Activación de una clave de KMS

En el siguiente ejemplo de `enable-key` se habilita una clave administrada por el cliente. Puede utilizar un comando como este para habilitar una clave de KMS que deshabilitó temporalmente mediante el comando `disable-key`. También puede usarlo para habilitar una clave de KMS que esté deshabilitada porque la eliminación estaba programada y esta se canceló.

Para especificar la clave de KMS, utilice el parámetro `key-id`. En este ejemplo, se utiliza un valor de ID de clave, pero puede utilizar un ID de clave o un valor de ARN de clave en este comando.

Antes de ejecutar este comando, reemplace el ID de claves de ejemplo por uno válido.

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

Este comando no genera ninguna salida. Utilice el comando `describe-key` para comprobar que la clave de KMS está habilitada. Consulte los valores de los campos `KeyState` y `Enabled` en la salida `describe-key`.

Para obtener más información, consulte [Habilitar y deshabilitar claves](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulta [EnableKey](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Asynchronously enables the specified key.
 *
 * @param keyId the ID of the key to enable
 * @return a {@link CompletableFuture} that completes when the key has been
 * enabled
 */
public CompletableFuture<Void> enableKeyAsync(String keyId) {
    EnableKeyRequest enableKeyRequest = EnableKeyRequest.builder()
        .keyId(keyId)
        .build();

    CompletableFuture<EnableKeyResponse> responseFuture =
getAsyncClient().enableKey(enableKeyRequest);
    responseFuture.whenComplete((response, exception) -> {
        if (exception == null) {
            logger.info("Key with ID [{}] has been enabled.", keyId);
        } else {
            if (exception instanceof KmsException kmsEx) {
                throw new RuntimeException("KMS error occurred while enabling
key: " + kmsEx.getMessage(), kmsEx);
            } else {
                throw new RuntimeException("An unexpected error occurred
while enabling key: " + exception.getMessage(), exception);
            }
        }
    });
}
```

```
        return responseFuture.thenApply(response -> null);  
    }
```

- Para obtener más información sobre la API, consulta [EnableKey](#) la Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
suspend fun enableKey(keyIdVal: String?) {  
    val request =  
        EnableKeyRequest {  
            keyId = keyIdVal  
        }  
  
    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->  
        kmsClient.enableKey(request)  
        println("$keyIdVal was successfully enabled.")  
    }  
}
```

- Para obtener más información sobre la API, consulta [EnableKey](#) la referencia sobre el AWS SDK para la API de Kotlin.

PHP

SDK para PHP

 Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $keyId
 * @return void
 */
public function enableKey(string $keyId)
{
    try {
        $this->client->enableKey([
            'KeyId' => $keyId,
        ]);
    } catch (KmsException $caught){
        if($caught->getAwsErrorMessage() == "NotFoundException"){
            echo "The request was rejected because the specified entity or
resource could not be found.\n";
        }
        throw $caught;
    }
}
```

- Para obtener más información sobre la API, consulta [EnableKey](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def enable_key(self, key_id: str) -> None:
        """
        Enables a key. Gets the key state after each state change.

        :param key_id: The ARN or ID of the key to enable.
        """
        try:
            self.kms_client.enable_key(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't enable key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Para obtener más información sobre la API, consulta [EnableKey](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    lo_kms->enablekey( iv_keyid = iv_key_id ).  
    MESSAGE 'KMS key enabled successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [EnableKey](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **EnableKeyRotation** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `EnableKeyRotation`.

CLI

AWS CLI

Cómo habilitar la rotación automática de una clave KMS

El siguiente ejemplo de `enable-key-rotation` permite la rotación automática de una clave de KMS administrada por el cliente con un período de rotación de 180 días. La clave de KMS se rotará un año (aproximadamente 365 días) a partir de la fecha en que se complete este comando y cada año a partir de entonces.

El parámetro `--key-id` identifica las claves KMS. En este ejemplo, se utiliza un valor de ARN clave, pero puede utilizar el ID de clave o el ARN de la clave de KMS. El parámetro `--rotation-period-in-days` especifica el número de días entre cada fecha de rotación. Especifique un valor comprendido entre 90 y 2560 días. El valor predeterminado es 365 días si no se especifica ningún valor.

```
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 no genera ninguna salida. Utilice el comando `get-key-rotation-status` para comprobar que la clave de KMS está habilitada.

Para obtener más información, consulte [Rotación de claves](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte [EnableKeyRotation](#) la Referencia de AWS CLI comandos.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def enable_key_rotation(self, key_id: str) -> None:
        """
        Enables rotation for a key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        """
        try:
            self.kms_client.enable_key_rotation(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't enable rotation for key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Para obtener más información sobre la API, consulta [EnableKeyRotation](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    lo_kms->enablekeyrotation( iv_keyid = iv_key_id ).  
    MESSAGE 'Key rotation enabled successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsdisabledexception.  
    MESSAGE 'The key is disabled.' TYPE 'E'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmsunsupportedopex.  
    MESSAGE 'Operation not supported for this key.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [EnableKeyRotation](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **Encrypt** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar Encrypt.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

CLI

AWS CLI

Ejemplo 1: Cifrado del contenido de un archivo en Linux o MacOS

El siguiente encrypt comando muestra la forma recomendada de cifrar datos con la AWS CLI.

```
aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob | base64 \  
  --decode > ExampleEncryptedFile
```

El comando hace varias cosas:

Utiliza el parámetro `--plaintext` para indicar los datos que se van a cifrar. El valor de este parámetro debe estar codificado en base64. El valor del `plaintext` parámetro debe estar codificado en base64 o debe usar el prefijo `fileb://`, que indica a la AWS CLI que lea los datos binarios del archivo. Si el archivo no está en el directorio actual, escriba la ruta completa al archivo. Por ejemplo: `fileb:///var/tmp/ExamplePlaintextFile` o `fileb://C:\Temp\ExamplePlaintextFile`. [Para obtener más información sobre la lectura de los valores de los parámetros de la AWS CLI de un archivo, consulte Carga de parámetros desde un archivo en la Guía del usuario de la interfaz de línea de AWS comandos y Prácticas recomendadas para los parámetros de archivos locales en el blog de herramientas de línea de AWS comandos.](#) Utiliza los `--query` parámetros `--output` y para controlar la salida del comando. Estos parámetros extraen los datos cifrados, denominados `texto cifrado`, de la salida del comando. [Para obtener más información sobre el control de la salida, consulte Control del comando Resultado de](#) la Guía del usuario de la interfaz de línea de AWS comandos. Utiliza la `base64` utilidad para decodificar el resultado extraído en datos binarios. El texto cifrado que devuelve un `encrypt` comando correcto es texto codificado en base64. Debe decodificar este texto antes de poder utilizar la AWS CLI para descifrarlo. Guarda el texto cifrado binario en un archivo. La parte final del comando (`> ExampleEncryptedFile`) guarda el texto cifrado binario en un archivo para facilitar el descifrado. Para ver un comando de ejemplo que usa la AWS CLI para descifrar datos, consulte los ejemplos de descifrado.

Ejemplo 2: Uso de la AWS CLI para cifrar datos en Windows

Este ejemplo es igual al anterior, excepto que utiliza la herramienta `certutil` en lugar de `base64`. Como se muestra en el siguiente ejemplo, este procedimiento requiere dos comandos.

```
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
```

Ejemplo 3: cifrado con una clave de KMS asimétrica

El siguiente comando `encrypt` muestra cómo cifrar texto sin formato con una clave de KMS asimétrica. El parámetro `--encryption-algorithm` es obligatorio. Como en todos los comandos de la `encrypt` CLI, el `plaintext` parámetro debe estar codificado en `base64` o debe usar el `fileb://` prefijo, que indica a la AWS CLI que lea los datos binarios del archivo.

```
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 no genera ninguna salida.

- Para obtener información sobre la API, consulte [Encrypt](#) en la Referencia de comandos de la AWS CLI .

Java

SDK para Java 2.x

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).


```

/**
 * Encrypts the given text asynchronously using the specified KMS client and
 * key ID.
 *
 * @param keyId the ID of the KMS key to use for encryption
 * @param text the text to encrypt
 * @return a CompletableFuture that completes with the encrypted data as an
 * SdkBytes object
 */
public CompletableFuture<SdkBytes> encryptDataAsync(String keyId, String
text) {
    SdkBytes myBytes = SdkBytes.fromUtf8String(text);
    EncryptRequest encryptRequest = EncryptRequest.builder()
        .keyId(keyId)
        .plaintext(myBytes)
        .build();

    CompletableFuture<EncryptResponse> responseFuture =
getAsyncClient().encrypt(encryptRequest).toCompletableFuture();
    return responseFuture.whenComplete((response, ex) -> {
        if (response != null) {
            String algorithm = response.encryptionAlgorithm().toString();
            logger.info("The string was encrypted with algorithm {}.\"",
algorithm);
        } else {
            throw new RuntimeException(ex);
        }
    }).thenApply(EncryptResponse::ciphertextBlob);
}

```

- Para obtener información sobre la API, consulte [Encrypt](#) en la referencia de la API de AWS SDK for Java 2.x .

Kotlin

SDK para Kotlin

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
suspend fun encryptData(keyIdValue: String): ByteArray? {
    val text = "This is the text to encrypt by using the AWS KMS Service"
    val myBytes: ByteArray = text.toByteArray()

    val encryptRequest =
        EncryptRequest {
            keyId = keyIdValue
            plaintext = myBytes
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.encrypt(encryptRequest)
        val algorithm: String = response.encryptionAlgorithm.toString()
        println("The encryption algorithm is $algorithm")

        // Return the encrypted data.
        return response.ciphertextBlob
    }
}

suspend fun decryptData(
    encryptedDataVal: ByteArray?,
    keyIdVal: String?,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
        val myVal = decryptResponse.plaintext
    }
}
```

```
        // Print the decrypted data.
        print(myVal)
    }
}
```

- Para obtener información sobre la API, consulte [Encrypt](#) en la referencia de la API de AWS SDK para Kotlin.

PHP

SDK para PHP

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $keyId
 * @param string $text
 * @return Result
 */
public function encrypt(string $keyId, string $text)
{
    try {
        return $this->client->encrypt([
            'KeyId' => $keyId,
            'Plaintext' => $text,
        ]);
    } catch (KmsException $caught) {
        if ($caught->getAwsErrorMessage() == "DisabledException") {
            echo "The request was rejected because the specified KMS key is
not enabled.\n";
        }
        throw $caught;
    }
}
```

- Para obtener información sobre la API, consulte [Encrypt](#) en la referencia de la API de AWS SDK para PHP .

Python

SDK para Python (Boto3)

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def encrypt(self, key_id: str, text: str) -> bytes:
        """
        Encrypts text by using the specified key.

        :param key_id: The ARN or ID of the key to use for encryption.
        :param text: The text to encrypt.
        :return: The encrypted version of the text.
        """
        try:
```

```

        response = self.kms_client.encrypt(KeyId=key_id,
        Plaintext=text.encode())
        print(
            f"The string was encrypted with algorithm
{response['EncryptionAlgorithm']}"
        )
        return response["CiphertextBlob"]
    except ClientError as err:
        if err.response["Error"]["Code"] == "DisabledException":
            logger.error(
                "Could not encrypt because the key %s is disabled.", key_id
            )
        else:
            logger.error(
                "Couldn't encrypt text. Here's why: %s",
                err.response["Error"]["Message"],
            )
        raise

```

- Para obtener información sobre la API, consulte [Encrypt](#) en la referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

require 'aws-sdk-kms' # v2: require 'aws-sdk'

# ARN of the AWS KMS key.
#
# Replace the fictitious key ARN with a valid key ID

```

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

text = '1234567890'

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.encrypt({
    key_id: keyId,
    plaintext: text
})

# Display a readable version of the resulting encrypted blob.
puts 'Blob:'
puts resp.ciphertext_blob.unpack('H*')
```

- Para obtener información sobre la API, consulte [Encrypt](#) en la referencia de la API de AWS SDK para Ruby .

Rust

SDK para Rust

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
async fn encrypt_string(
    verbose: bool,
    client: &Client,
    text: &str,
    key: &str,
    out_file: &str,
) -> Result<(), Error> {
    let blob = Blob::new(text.as_bytes());

    let resp = client.encrypt().key_id(key).plaintext(blob).send().await?;
```

```
// Did we get an encrypted blob?
let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
let bytes = blob.as_ref();

let s = base64::encode(bytes);

let mut ofile = File::create(out_file).expect("unable to create file");
ofile.write_all(s.as_bytes()).expect("unable to write");

if verbose {
    println!("Wrote the following to {:?}" , out_file);
    println!("{}", s);
}

Ok(())
}
```

- Para obtener información sobre la API, consulte [Encrypt](#) en la Referencia de la API de AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    " iv_plaintext contains the data to encrypt
    oo_result = lo_kms->encrypt(
        iv_keyid = iv_key_id
        iv_plaintext = iv_plaintext
    ).
    MESSAGE 'Text encrypted successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.
```

```
MESSAGE 'The key is disabled.' TYPE 'E'.  
CATCH /aws1/cx_kmsnotfoundexception.  
MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [Cifrar](#) en el AWS SDK para obtener información sobre la API ABAP de SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. [Uso de este servicio con un AWS SDK](#) En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **GenerateDataKey** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar GenerateDataKey.

CLI

AWS CLI

Ejemplo 1: Generación de una clave de datos simétrica de 256 bits

En el siguiente generate-data-key ejemplo, se solicita una clave de datos simétrica de 256 bits para utilizarla fuera de. AWS El comando devuelve una clave de datos en texto sin formato para su uso y eliminación inmediatos, y una copia de esa clave de datos cifrada bajo la clave de KMS especificada. Puede almacenar de forma segura la clave de datos cifrada con los datos cifrados.

Para solicitar una clave de datos de 256 bits, utilice el parámetro key-spec con un valor de AES_256. Para solicitar una clave de datos de 128 bits, utilice el parámetro key-spec con un valor de AES_128. Para el resto longitudes de clave de datos, utilice el parámetro number-of-bytes.

La clave de KMS que especifique debe ser una clave de KMS de cifrado simétrica, es decir, una clave de KMS con un valor de especificación de clave SYMMETRIC_DEFAULT.

```
aws kms generate-data-key \
```



```
--key-id alias/ExampleAlias \  
--key-spec AES_256
```

Salida:

```
{  
  "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",  
  "KeyId": "arn:aws:kms:us-  
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "KeyMaterialId":  
    "0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",  
  "CiphertextBlob":  
    "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk  
+YdhV8MrkBQPeac0ReRVNDt9qleAt+SHgIRF8P0H+7U="
```

Plaintext (clave de datos de texto sin formato) y CiphertextBlob (clave de datos cifrada) se devuelven en formato codificado en base64.

Para obtener más información, consulte [Claves de datos](#) en la Guía para desarrolladores de AWS Key Management Service. Ejemplo 2: Generación de una clave de datos simétrica de 512 bits

En el siguiente ejemplo de `generate-data-key` se solicita una clave de datos simétrica de 512 bits para cifrado y descifrado. El comando devuelve una clave de datos en texto sin formato para su uso y eliminación inmediatos, y una copia de esa clave de datos cifrada bajo la clave de KMS especificada. Puede almacenar de forma segura la clave de datos cifrada con los datos cifrados.

Para solicitar una longitud de clave distinta de 128 o 256 bits, utilice el parámetro `number-of-bytes`. Para solicitar una clave de datos de 512 bits, en el siguiente ejemplo se utiliza el parámetro `number-of-bytes` con un valor de 64 (bytes).

La clave KMS que especifique debe ser una clave de KMS de cifrado simétrica, es decir, una clave de KMS con un valor de especificación de clave de `SYMMETRIC_DEFAULT`.

NOTA: Los valores de la salida de este ejemplo se han truncado para su visualización.

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

--number-of-bytes 64

Salida:

```
{
  "CiphertextBlob": "AQIBAHi6LtupRpdK12aJTzkK6Fbh0tQkMlQJJH3PdtHvS/y+hAEnX/
QQNmMwDfg2koRNMEc8AAACaDCCAmQGCSqGSib3DQEHbqCCA1UwggJRAgEAMIICSgYJKoZ...",
  "Plaintext": "ty8Lr0Bk60F07M2Bwt6qbFdNB
+G00ZLtf5MSEb4a13R2UKWG0p06njAwy2n72VRm2m7z/
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm40ovEXiVfBveyX3DQxDzRSwbKDPk/...",
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyMaterialId":
  "0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

Plaintext (clave de datos de texto sin formato) y CiphertextBlob (clave de datos cifrada) se devuelven en formato codificado en base64.

Para obtener más información, consulte [Claves de datos](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte la Referencia de [GenerateDataKey](#) comandos AWS CLI .

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []
```

```

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def generate_data_key(self, key_id):
        """
        Generates a symmetric data key that can be used for client-side
encryption.
        """
        answer = input(
            f"Do you want to generate a symmetric data key from key {key_id} (y/
n)? "
        )
        if answer.lower() == "y":
            try:
                data_key = self.kms_client.generate_data_key(
                    KeyId=key_id, KeySpec="AES_256"
                )
            except ClientError as err:
                logger.error(
                    "Couldn't generate a data key for key %s. Here's why: %s",
                    key_id,
                    err.response["Error"]["Message"],
                )
            else:
                pprint(data_key)

```

- Para obtener más información sobre la API, consulta [GenerateDataKey](#) la AWS Referencia de API de SDK for Python (Boto3).

Rust

SDK para Rust

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
async fn make_key(client: &Client, key: &str) -> Result<(), Error> {
    let resp = client
        .generate_data_key()
        .key_id(key)
        .key_spec(DataKeySpec::Aes256)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
}
```

- Para obtener más información sobre la API, consulta [GenerateDataKey](#) la referencia sobre la API de AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    " iv_keyspec = 'AES_256'
    oo_result = lo_kms->generatedatakey(
        iv_keyid = iv_key_id
        iv_keyspec = 'AES_256'
    ).
    MESSAGE 'Data key generated successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.
    MESSAGE 'The key is disabled.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [GenerateDataKey](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **GenerateDataKeyWithoutPlaintext** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `GenerateDataKeyWithoutPlaintext`.

CLI

AWS CLI

Generación de una clave de datos simétrica de 256 bits sin una clave de texto no cifrado

En el siguiente ejemplo de `generate-data-key-without-plaintext` se solicita una copia cifrada de una clave de datos simétrica de 256 bits para usarla fuera de AWS. Puede llamar a AWS KMS para descifrar la clave de datos cuando esté listo para usarla.

Para solicitar una clave de datos de 256 bits, utilice el parámetro `key-spec` con un valor de `AES_256`. Para solicitar una clave de datos de 128 bits, utilice el parámetro `key-spec` con un valor de `AES_128`. Para el resto longitudes de clave de datos, utilice el parámetro `number-of-bytes`.

La clave de KMS que especifique debe ser una clave de KMS de cifrado simétrica, es decir, una clave de KMS con un valor de especificación de clave `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
```

Salida:

```
{
  "CiphertextBlob":
"AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
  "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyMaterialId":
"0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

`CiphertextBlob` (clave de datos cifrada) se devuelve en formato codificado en base64.

Para obtener más información, consulte [Claves de datos](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte [GenerateDataKeyWithoutPlaintext](#) la Referencia de AWS CLI comandos.

Rust

SDK para Rust

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
async fn make_key(client: &Client, key: &str) -> Result<(), Error> {
    let resp = client
        .generate_data_key_without_plaintext()
        .key_id(key)
        .key_spec(DataKeySpec::Aes256)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
}
```

- Para obtener más información sobre la API, consulta [GenerateDataKeyWithoutPlaintext](#) la referencia sobre la API de AWS SDK para Rust.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **GenerateRandom** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `GenerateRandom`.

CLI

AWS CLI

Ejemplo 1: Generación de una cadena de bytes aleatoria de 256 bits (Linux o macOS)

En el siguiente ejemplo de `generate-random` se genera una cadena de bytes aleatoria de 256 bits (32 bytes) codificada en base64. El ejemplo decodifica la cadena de bytes y la guarda en el archivo aleatorio.

Al ejecutar este comando, debe usar el parámetro `number-of-bytes` para especificar la longitud del valor aleatorio en bytes.

No se especifica una clave de KMS cuando se ejecuta este comando. La cadena de bytes aleatoria no está relacionada con ninguna clave de KMS.

De forma predeterminada, AWS KMS genera el número aleatorio. Sin embargo, si especifica un [almacén de claves personalizado](#), la cadena de bytes aleatorios se genera en el clúster de AWS CloudHSM asociado al almacén de claves personalizado.

Este ejemplo utiliza los parámetros y valores siguientes:

Utiliza el `--number-of-bytes` parámetro necesario con un valor de 32 para solicitar una cadena de 32 bytes (256 bits). Utiliza el `--output` parámetro con un valor de `text` para indicar a la AWS CLI que devuelva la salida como texto, en lugar de JSON. Utiliza el `--query Plaintext` para extraer el valor de la `Plaintext` propiedad de la respuesta. Canaliza (`|`) la salida del comando `--query parameter` a la `base64 --decode` utilidad, que decodifica la salida extraída. Utiliza el operador de redirección (`>`) para guardar el byte decodificado cadena al archivo. Utiliza el operador de redirección (`>`) `ExampleRandom` para guardar el texto cifrado binario en un archivo.

```
aws kms generate-random \
  --number-of-bytes 32 \
  --output text \
  --query Plaintext | base64 --decode > ExampleRandom
```

Este comando no genera ninguna salida.

Para obtener más información, consulte la referencia de [GenerateRandom](#) la API del Servicio de administración de AWS claves.

Ejemplo 2: Generación de un número aleatorio de 256 bits (símbolo del sistema de Windows)

En el siguiente ejemplo se utiliza el comando `generate-random` para generar una cadena de bytes aleatoria codificada en base64 de 256 bits (32 bytes). El ejemplo decodifica la cadena de bytes y la guarda en el archivo aleatorio. Este ejemplo es igual al anterior, excepto que usa la utilidad `certutil` de Windows para decodificar en base64 la cadena de bytes aleatoria antes de guardarla en un archivo.

En primer lugar, genere una cadena de bytes aleatoria codificada en base64 y guárdela en un archivo temporal, `ExampleRandom.base64`.

```
aws kms generate-random \
  --number-of-bytes 32 \
  --output text \
  --query Plaintext > ExampleRandom.base64
```

Puesto que el resultado del comando `generate-random` se guarda en un archivo, este ejemplo no produce ningún resultado.

Ahora utilice el comando `certutil -decode` para decodificar la cadena de bytes codificada en base64 del archivo `ExampleRandom.base64`. A continuación, guarda la cadena de bytes decodificada en el archivo `ExampleRandom`.

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

Salida:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Para obtener más información, consulte la referencia [GenerateRandom](#) de la API del Servicio de administración de AWS claves.

- Para obtener más información sobre la API, consulte [GenerateRandom](#) la Referencia de AWS CLI comandos.

Rust

SDK para Rust

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
async fn make_string(client: &Client, length: i32) -> Result<(), Error> {
    let resp = client
        .generate_random()
        .number_of_bytes(length)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.plaintext.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
}
```

- Para obtener más información sobre la API, consulta [GenerateRandom](#) la referencia sobre la API de AWS SDK para Rust.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **GetKeyPolicy** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar GetKeyPolicy.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

CLI

AWS CLI

Copia de una política de claves de una clave de KMS a otra

En el siguiente ejemplo de `get-key-policy` se obtiene la política de claves de una clave de KMS y se guarda en un archivo de texto. A continuación, reemplaza la política de una clave de KMS diferente utilizando el archivo de texto como entrada de la política.

Puesto que el parámetro `--policy` de `put-key-policy` requiere una cadena, debe usar la opción `--output text` para devolver la salida como una cadena de texto en lugar 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 \
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
  --policy file://policy.txt
```

Este comando no genera ninguna salida.

Para obtener más información, consulte la referencia [PutKeyPolicy](#) de la API de AWS KMS.

- Para obtener más información sobre la API, consulte [GetKeyPolicy](#) la Referencia de AWS CLI comandos.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def get_policy(self, key_id: str) -> dict[str, str]:
        """
        Gets the policy of a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The key policy as a dict.
        """
        if key_id != "":
            try:
                response = self.kms_client.get_key_policy(
                    KeyId=key_id,
                )
                policy = json.loads(response["Policy"])
            except ClientError as err:
                logger.error(
                    "Couldn't get policy for key %s. Here's why: %s",

```

```

        key_id,
        err.response["Error"]["Message"],
    )
    raise
else:
    pprint(policy)
    return policy
else:
    print("Skipping get policy demo.")

```

- Para obtener más información sobre la API, consulta [GetKeyPolicy](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    oo_result = lo_kms->getkeypolicy(
        iv_keyid = iv_key_id
        iv_policyname = 'default'
    ).
    MESSAGE 'Retrieved key policy successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [GetKeyPolicy](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **ListAliases** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListAliases`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

.NET

SDK para .NET

Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// List the AWS Key Management Service (AWS KMS) aliases that have been
defined for
/// the keys in the same AWS Region as the default user. If you want to list
/// the aliases in a different Region, pass the Region to the client
/// constructor.
/// </summary>
public class ListAliases
{

```

```
public static async Task Main()
{
    var client = new AmazonKeyManagementServiceClient();
    var request = new ListAliasesRequest();
    var response = new ListAliasesResponse();

    do
    {
        response = await client.ListAliasesAsync(request);

        response.Aliases.ForEach(alias =>
        {
            Console.WriteLine($"Created: {alias.CreationDate} Last
Update: {alias.LastUpdatedDate} Name: {alias.AliasName}");
        });

        request.Marker = response.NextMarker;
    }
    while (response.Truncated);
}
```

- Para obtener más información sobre la API, consulta [ListAliases](#) la Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Ejemplo 1: Para enumerar todos los alias de una AWS cuenta y una región

En el siguiente ejemplo, se utiliza el `list-aliases` comando para enumerar todos los alias de la región predeterminada de la AWS cuenta. El resultado incluye los alias asociados a las claves de KMS AWS administradas y a las claves de KMS administradas por el cliente.

```
aws kms list-aliases
```

Salida:

```
{
```

```
"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"  
  },  
  ...  
]  
}
```

Ejemplo 2: Creación de una lista de todos los alias de una clave de KMS concreta

En el siguiente ejemplo se utiliza el comando `list-aliases` y su parámetro `key-id` para mostrar todos los alias que están asociados con una determinada clave de KMS.

Cada alias está asociado a una sola clave de KMS, pero una clave de KMS puede tener varios alias. Este comando es muy útil porque la consola de AWS KMS solo muestra un alias para cada clave de KMS. Debe utilizar el comando `list-aliases` para encontrar todos los alias de una clave de KMS.

En este ejemplo, se utiliza el ID de la clave de KMS del parámetro `--key-id`, pero puede utilizar un ID de clave, un ARN de clave, un nombre de alias o un ARN de alias en este comando.

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

Salida:


```
{
  "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 obtener más información, consulte [Uso de alias](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte [ListAliases](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Asynchronously lists all the aliases in the current AWS account.
 *
 * @return a {@link CompletableFuture} that completes when the list of
 * aliases has been processed
 */
public CompletableFuture<Object> listAllAliasesAsync() {
    ListAliasesRequest aliasesRequest = ListAliasesRequest.builder()
```

```

        .limit(15)
        .build();

    ListAliasesPublisher paginator =
getAsyncClient().listAliasesPaginator(aliancesRequest);
    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 obtener más información sobre la API, consulta [ListAliases](#) la Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

suspend fun listAllAliases() {
    val request =
        ListAliasesRequest {
            limit = 15

```

```

    }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listAliases(request)
        response.aliases?.forEach { alias ->
            println("The alias name is ${alias.aliasName}")
        }
    }
}

```

- Para obtener más información sobre la API, consulta [ListAliases](#) la referencia sobre el AWS SDK para la API de Kotlin.

PHP

SDK para PHP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * @param string $keyId
 * @param int $limit
 * @return ResultPaginator
 */
public function listAliases(string $keyId = "", int $limit = 0)
{
    $args = [];
    if($keyId){
        $args['KeyId'] = $keyId;
    }
    if($limit){
        $args['Limit'] = $limit;
    }
    try{
        return $this->client->getPaginator("ListAliases", $args);
    }
}

```

```

        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidMarkerException"){
                echo "The request was rejected because the marker that specifies
where pagination should next begin is not valid.\n";
            }
            throw $caught;
        }
    }
}

```

- Para obtener más información sobre la API, consulta [ListAliases](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

```

```

def list_aliases(self, page_size: int) -> None:
    """
    Lists aliases for the current account.
    :param page_size: The number of aliases to list per page.
    """
    try:
        alias_paginator = self.kms_client.get_paginator("list_aliases")
        for alias_page in alias_paginator.paginate(
            PaginationConfig={"PageSize": page_size}
        ):
            print(f"Here are {page_size} aliases:")
            pprint(alias_page["Aliases"])
            if alias_page["Truncated"]:
                answer = input(
                    f"Do you want to see the next {page_size} aliases (y/n)?
                "
                )
                if answer.lower() != "y":
                    break
            else:
                print("That's all your aliases!")
    except ClientError as err:
        logging.error(
            "Couldn't list your aliases. Here's why: %s",
            err.response["Error"]["Message"],
        )
        raise

```

- Para obtener más información sobre la API, consulta [ListAliases](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    oo_result = lo_kms->listalias( ).  
    MESSAGE 'Retrieved KMS aliases list.' TYPE 'I'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [ListAliases](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **ListGrants** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar ListGrants.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

.NET

SDK para .NET

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
using System;  
using System.Threading.Tasks;  
using Amazon.KeyManagementService;  
using Amazon.KeyManagementService.Model;
```

```
/// <summary>
/// List the AWS Key Management Service (AWS KMS) grants that are associated
with
/// a specific key.
/// </summary>
public class ListGrants
{
    public static async Task Main()
    {
        // The identifier of the AWS KMS key to disable. You can use the
        // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";
        var client = new AmazonKeyManagementServiceClient();
        var request = new ListGrantsRequest
        {
            KeyId = keyId,
        };

        var response = new ListGrantsResponse();

        do
        {
            response = await client.ListGrantsAsync(request);

            response.Grants.ForEach(grant =>
            {
                Console.WriteLine($"{grant.GrantId}");
            });

            request.Marker = response.NextMarker;
        }
        while (response.Truncated);
    }
}
```

- Para obtener más información sobre la API, consulta [ListGrants](#) la Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Para ver las concesiones de una clave de AWS KMS

En el siguiente `list-grants` ejemplo, se muestran todas las concesiones de la clave de KMS AWS gestionada especificada para Amazon DynamoDB en su cuenta. Esta concesión permite a DynamoDB utilizar la clave de KMS en su nombre para cifrar una tabla de DynamoDB antes de escribirla en el disco. Puede usar un comando como este para ver las concesiones de las claves de KMS AWS administradas y las claves de KMS administradas por el cliente en la AWS cuenta y la región.

Este comando usa el parámetro `key-id` con un ID de la clave para identificar la clave de KMS. Puede utilizar un ID o ARN de clave para identificar la clave de KMS. Para obtener el identificador de clave o el ARN de clave de una clave de KMS AWS administrada, utilice el comando `list-keys` o `list-aliases`.

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

La salida muestra que la concesión otorga a Amazon DynamoDB permiso para usar la clave de KMS para operaciones criptográficas y le da permiso para ver detalles sobre la clave de KMS (`DescribeKey`) y para retirar concesiones (`RetireGrant`). La restricción `EncryptionContextSubset` limita estos permisos a las solicitudes que incluyen los pares de contexto de cifrado especificados. Como resultado, los permisos de la concesión solo son efectivos en la cuenta y la tabla de DynamoDB especificadas.

```
{
  "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 obtener más información, consulte las [concesiones en AWS KMS en](#) la Guía para desarrolladores del servicio de administración de AWS claves.

- Para obtener más información sobre la API, consulte [ListGrants](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * Asynchronously displays the grant IDs for the specified key ID.
 *
 * @param keyId the ID of the AWS KMS key for which to list the grants
 * @return a {@link CompletableFuture} that, when completed, will be null
 * if the operation succeeded, or will throw a {@link RuntimeException} if the
 * operation failed

```

```

    * @throws RuntimeException if there was an error listing the grants, either
    due to an {@link KmsException} or an unexpected error
    */
    public CompletableFuture<Object> displayGrantIdsAsync(String keyId) {
        ListGrantsRequest grantsRequest = ListGrantsRequest.builder()
            .keyId(keyId)
            .limit(15)
            .build();

        ListGrantsPublisher paginator =
            getAsyncClient().listGrantsPaginator(grantsRequest);
        return paginator.subscribe(response -> {
            response.grants().forEach(grant -> {
                logger.info("The grant Id is: " + grant.grantId());
            });
        })
        .thenApply(v -> null)
        .exceptionally(ex -> {
            Throwable cause = ex.getCause();
            if (cause instanceof KmsException) {
                throw new RuntimeException("Failed to list grants: " +
                    cause.getMessage(), cause);
            } else {
                throw new RuntimeException("An unexpected error occurred: " +
                    cause.getMessage(), cause);
            }
        });
    }
}

```

- Para obtener más información sobre la API, consulta [ListGrants](#) la Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
suspend fun displayGrantIds(keyIdVal: String?) {
    val request =
        ListGrantsRequest {
            keyId = keyIdVal
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listGrants(request)
        response.grants?.forEach { grant ->
            println("The grant Id is ${grant.grantId}")
        }
    }
}
```

- Para obtener más información sobre la API, consulta [ListGrants](#) la referencia sobre el AWS SDK para la API de Kotlin.

PHP

SDK para PHP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $keyId
 * @return Result
 */
public function listGrants(string $keyId)
{
    try{
        return $this->client->listGrants([
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
```

```

        if($caught->getAwsErrorMessage() == "NotFoundException"){
            echo "    The request was rejected because the specified entity
or resource could not be found.\n";
        }
        throw $caught;
    }
}

```

- Para obtener más información sobre la API, consulta [ListGrants](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def list_grants(self, key_id):
        """

```

Lists grants for a key.

```
:param key_id: The ARN or ID of the key to query.
:return: The grants for the key.
"""
try:
    paginator = self.kms_client.get_paginator("list_grants")
    grants = []
    page_iterator = paginator.paginate(KeyId=key_id)
    for page in page_iterator:
        grants.extend(page["Grants"])

    print(f"Grants for key {key_id}:")
    pprint(grants)
    return grants
except ClientError as err:
    logger.error(
        "Couldn't list grants for key %s. Here's why: %s",
        key_id,
        err.response["Error"]["Message"],
    )
    raise
```

- Para obtener más información sobre la API, consulta [ListGrants](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

TRY.

```
" iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
```

```

        oo_result = lo_kms->listgrants( iv_keyid = iv_key_id ).
        MESSAGE 'Retrieved grants list.' TYPE 'I'.
    CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
    CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.

```

- Para obtener más información sobre la API, consulte [ListGrants](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **ListKeyPolicies** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListKeyPolicies`.

CLI

AWS CLI

Obtención de los nombres de las políticas de claves de una clave de KMS

En el siguiente ejemplo de `list-key-policies` se obtienen los nombres de las políticas de clave de una clave administrada por el cliente en la cuenta y la región del ejemplo. Puede usar este comando para buscar los nombres de las políticas clave para las claves AWS administradas y las claves administradas por el cliente.

Este comando no resulta útil, ya que el único nombre de la política de claves válido es `default`.

Para especificar la clave de KMS, utilice el parámetro `key-id`. En este ejemplo, se utiliza un valor de ID de clave, pero puede utilizar un ID de clave o un ARN de clave en este comando.

```

aws kms list-key-policies \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

```

Salida:

```
{
  "PolicyNames": [
    "default"
  ]
}
```

Para obtener más información sobre las políticas clave de AWS KMS, consulte [Uso de políticas clave en AWS KMS](#) en la Guía AWS para desarrolladores del Servicio de administración de claves.

- Para obtener más información sobre la API, consulte [ListKeyPolicies](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Asynchronously retrieves the key policy for the specified key ID and
 * policy name.
 *
 * @param keyId      the ID of the AWS KMS key for which to retrieve the
 * policy
 * @param policyName the name of the key policy to retrieve
 * @return a {@link CompletableFuture} that, when completed, contains the key
 * policy as a {@link String}
 */
public CompletableFuture<String> getKeyPolicyAsync(String keyId, String
policyName) {
    GetKeyPolicyRequest policyRequest = GetKeyPolicyRequest.builder()
        .keyId(keyId)
        .policyName(policyName)
        .build();

    return getAsyncClient().getKeyPolicy(policyRequest)
```

```

        .thenApply(response -> {
            String policy = response.policy();
            logger.info("The response is: " + policy);
            return policy;
        })
        .exceptionally(ex -> {
            throw new RuntimeException("Failed to get key policy", ex);
        });
    }
}

```

- Para obtener más información sobre la API, consulta [ListKeyPolicies](#) la Referencia AWS SDK for Java 2.x de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def list_policies(self, key_id):

```



```

"""
Lists the names of the policies for a key.

:param key_id: The ARN or ID of the key to query.
"""
try:
    policy_names = self.kms_client.list_key_policies(KeyId=key_id)[
        "PolicyNames"
    ]
except ClientError as err:
    logging.error(
        "Couldn't list your policies. Here's why: %s",
        err.response["Error"]["Message"],
    )
    raise
else:
    print(f"The policies for key {key_id} are:")
    pprint(policy_names)

```

- Para obtener más información sobre la API, consulta [ListKeyPolicies](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    oo_result = lo_kms->listkeypolicies( iv_keyid = iv_key_id ).
    MESSAGE 'Retrieved key policies list.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.

```

```
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [ListKeyPolicies](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **ListKeys** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar ListKeys.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

.NET

SDK para .NET

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
using System;  
using System.Threading.Tasks;  
using Amazon.KeyManagementService;  
using Amazon.KeyManagementService.Model;  
  
/// <summary>  
/// List the AWS Key Managements Service (AWS KMS) keys for the AWS Region  
/// of the default user. To list keys in another AWS Region, supply the  
Region
```

```
/// as a parameter to the client constructor.
/// </summary>
public class ListKeys
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();
        var request = new ListKeysRequest();
        var response = new ListKeysResponse();

        do
        {
            response = await client.ListKeysAsync(request);

            response.Keys.ForEach(key =>
            {
                Console.WriteLine($"ID: {key.KeyId}, {key.KeyArn}");
            });

            // Set the Marker property when response.Truncated is true
            // in order to get the next keys.
            request.Marker = response.NextMarker;
        }
        while (response.Truncated);
    }
}
```

- Para obtener más información sobre la API, consulta [ListKeys](#) la Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Obtención de las claves KMS de una cuenta y una región

En el siguiente ejemplo de `list-keys` se obtienen las claves de KMS de una cuenta y una región. Este comando devuelve tanto las claves AWS administradas como las claves administradas por el cliente.

```
aws kms list-keys
```

Salida:

```
{
  "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 obtener más información, consulte [Visualización de claves](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte [ListKeys](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
```

```
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /**
         * The `subscribe` method is required when using paginator methods in the
         AWS SDK
         * because paginator methods return an instance of a `ListKeysPublisher`,
         which is
         * based on a reactive stream. This allows asynchronous retrieval of
         paginated
         * results as they become available. By subscribing to the stream, we can
         process
         * each page of results as they are emitted.
         */
        ListKeysPublisher keysPublisher =
            kmsAsyncClient.listKeysPaginator(listKeysRequest);
        CompletableFuture<Void> future = keysPublisher
            .subscribe(r -> r.keys().forEach(key ->
                System.out.println("The key ARN is: " + key.keyArn() + ". The key
                Id is: " + key.keyId()))
            .whenComplete((result, exception) -> {
                if (exception != null) {
```

```

        System.err.println("Error occurred: " +
exception.getMessage());
    } else {
        System.out.println("Successfully listed all keys.");
    }
});

    try {
        future.join();
    } catch (Exception e) {
        System.err.println("Failed to list keys: " + e.getMessage());
    }
}
}

```

- Para obtener más información sobre la API, consulta [ListKeys](#) la Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

suspend fun listAllKeys() {
    val request =
        ListKeysRequest {
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listKeys(request)
        response.keys?.forEach { key ->
            println("The key ARN is ${key.keyArn}")
            println("The key Id is ${key.keyId}")
        }
    }
}

```

```
}  
}
```

- Para obtener más información sobre la API, consulta [ListKeys](#) la referencia sobre el AWS SDK para la API de Kotlin.

PHP

SDK para PHP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**  
 * @return array  
 */  
public function listKeys()  
{  
    try {  
        $contents = [];  
        $paginator = $this->client->getPaginator("ListKeys");  
        foreach($paginator as $result){  
            foreach ($result['Content'] as $object) {  
                $contents[] = $object;  
            }  
        }  
        return $contents;  
    }catch(KmsException $caught){  
        echo "There was a problem listing the keys: {$caught->getAwsErrorMessage()}\n";  
        throw $caught;  
    }  
}
```

- Para obtener más información sobre la API, consulta [ListKeys](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def list_keys(self):
        """
        Lists the keys for the current account by using a paginator.
        """
        try:
            page_size = 10
            print("\nLet's list your keys.")
            key_paginator = self.kms_client.get_paginator("list_keys")
            for key_page in key_paginator.paginate(PaginationConfig={"PageSize":
10}):
                print(f"Here are {len(key_page['Keys'])} keys:")
```



```

        pprint(key_page["Keys"])
        if key_page["Truncated"]:
            answer = input(
                f"Do you want to see the next {page_size} keys (y/n)? "
            )
            if answer.lower() != "y":
                break
        else:
            print("That's all your keys!")
    except ClientError as err:
        logging.error(
            "Couldn't list your keys. Here's why: %s",
            err.response["Error"]["Message"],
        )

```

- Para obtener más información sobre la API, consulta [ListKeys](#) la AWS Referencia de API de SDK for Python (Boto3).

Rust

SDK para Rust

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

async fn show_keys(client: &Client) -> Result<(), Error> {
    let resp = client.list_keys().send().await?;

    let keys = resp.keys.unwrap_or_default();

    let len = keys.len();

    for key in keys {
        println!("Key ARN: {}", key.key_arn.as_deref().unwrap_or_default());
    }
}

```

```
println!();
println!("Found {} keys", len);

Ok(())
}
```

- Para obtener más información sobre la API, consulta [ListKeys](#) la referencia sobre la API de AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.
    oo_result = lo_kms->listkeys( ).
    MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.
CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte [ListKeys](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **PutKeyPolicy** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar PutKeyPolicy.

CLI

AWS CLI

Cambio de la política de claves de una clave de KMS

En el siguiente ejemplo de `put-key-policy` se cambia la política de claves para una clave administrada por el cliente.

Para empezar, cree una política de clave y guárdela en un archivo JSON local. En este ejemplo, el archivo es `key_policy.json`. También puede especificar la política de claves como un valor de cadena del parámetro `policy`.

La primera declaración de esta política clave otorga a la AWS cuenta permiso para usar las políticas de IAM para controlar el acceso a la clave KMS. La segunda instrucción otorga permiso al usuario `test-user` para ejecutar los comandos `describe-key` y `list-keys` de la clave de KMS.

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

Para identificar la clave de KMS, en este ejemplo se utiliza el ID de clave, pero también se puede utilizar un ARN de la clave. El comando utiliza el parámetro `policy` para especificar las políticas de claves. Para indicar que la política se encuentra en un archivo, utiliza el prefijo `file://` necesario. Este prefijo es necesario para identificar los archivos de todos los sistemas operativos compatibles. Por último, el comando usa el parámetro `policy-name` con un valor de `default`. Si no se especifica ningún nombre de política, el valor predeterminado es `default`. El único valor válido es `default`.

```
aws kms put-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --policy file://key_policy.json
```

Este comando no produce ninguna salida. Para comprobar que se aplicó el comando, utilice el comando `get-key-policy`. El siguiente comando de ejemplo obtiene la política de claves para la misma clave de KMS. El parámetro `output` con un valor de `text` devuelve un formato de texto que es fácil de leer.

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

Salida:

```
{
  "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 obtener más información, consulte [Cambio de una política de claves](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte [PutKeyPolicy](#) la Referencia de AWS CLI comandos.

PHP

SDK para PHP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * @param string $keyId
 * @param string $policy
 * @return void
 */
public function putKeyPolicy(string $keyId, string $policy)
{
    try {
        $this->client->putKeyPolicy([
            'KeyId' => $keyId,
            'Policy' => $policy,

```

```

    });
    }catch(KmsException $caught){
        echo "There was a problem replacing the key policy: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

```

- Para obtener más información sobre la API, consulta [PutKeyPolicy](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def set_policy(self, key_id: str, policy: dict[str, any]) -> None:
        """

```

```

    Sets the policy of a key. Setting a policy entirely overwrites the
    existing
    policy, so care is taken to add a statement to the existing list of
    statements
    rather than simply writing a new policy.

    :param key_id: The ARN or ID of the key to set the policy to.
    :param policy: The existing policy of the key.
    :return: None
    """
    principal = input(
        "Enter the ARN of an IAM role to set as the principal on the policy:
"
    )
    if key_id != "" and principal != "":
        # The updated policy replaces the existing policy. Add a new
statement to
        # the list along with the original policy statements.
        policy["Statement"].append(
            {
                "Sid": "Allow access for ExampleRole",
                "Effect": "Allow",
                "Principal": {"AWS": principal},
                "Action": [
                    "kms:Encrypt",
                    "kms:GenerateDataKey*",
                    "kms:Decrypt",
                    "kms:DescribeKey",
                    "kms:ReEncrypt*",
                ],
                "Resource": "*",
            }
        )
    try:
        self.kms_client.put_key_policy(KeyId=key_id,
Policy=json.dumps(policy))
    except ClientError as err:
        logger.error(
            "Couldn't set policy for key %s. Here's why %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise
    else:

```

```

        print(f"Set policy for key {key_id}.")
    else:
        print("Skipping set policy demo.")

```

- Para obtener más información sobre la API, consulta [PutKeyPolicy](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    " iv_policy = '{"Version":"2012-10-17",          "Statement": [...]}'
    lo_kms->putkeypolicy(
        iv_keyid = iv_key_id
        iv_policyname = 'default'
        iv_policy = iv_policy
    ).
    MESSAGE 'Key policy updated successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsmalformedplydocex.
    MESSAGE 'Malformed policy document.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [PutKeyPolicy](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **ReEncrypt** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar ReEncrypt.

CLI

AWS CLI

Ejemplo 1: Cómo volver a cifrar un mensaje cifrado con una clave de KMS simétrica diferente (Linux y macOS).

El siguiente ejemplo de `re-encrypt` comando muestra la forma recomendada de volver a cifrar los datos con la CLI AWS .

Proporcione el texto cifrado de un archivo. En el valor del parámetro `--ciphertext-blob`, utilice el prefijo `fileb://`, que indica a la CLI que lea los datos de un archivo binario. Si el archivo no se encuentra en el directorio actual, escriba la ruta completa al archivo. Para obtener más información sobre la lectura de los valores de los parámetros AWS CLI de un archivo, consulte [Carga de parámetros AWS CLI desde un archivo](#) en la Guía del usuario de la interfaz de línea de AWS comandos y [Prácticas recomendadas para los parámetros de archivos locales](#) en el blog de herramientas de línea de AWS comandos. Especifique la clave KMS de origen, que descifra el texto cifrado. El `--source-key-id` parámetro no es necesario al descifrar con claves KMS de cifrado simétrico. AWS KMS puede obtener la clave KMS que se utilizó para cifrar los datos de los metadatos del blob de texto cifrado. Pero siempre es una práctica recomendada especificar la clave KMS que está utilizando. Esta práctica garantiza que utilice la clave de KMS pretendida y le impide descifrar inadvertidamente un texto cifrado mediante una clave de KMS en la que no confía. Especifique la clave de KMS de destino, que vuelve a cifrar los datos. El parámetro `--destination-key-id` siempre es obligatorio. En este ejemplo, se utiliza un ARN de clave, pero se puede utilizar cualquier identificador de clave válido. Solicite la salida de texto sin formato como un valor de texto. El parámetro `--query` indica a la CLI que obtenga únicamente el valor del campo `Plaintext` de la salida. El parámetro `--output` devuelve la salida como texto. Decodifique en base64 el texto sin formato y guárdelo en un archivo. El siguiente ejemplo canaliza (`|`) el valor del parámetro `Plaintext` a la utilidad

Base64, que lo decodifica. A continuación, dirige (>) la salida descodificada al archivo `ExamplePlaintext`.

Antes de ejecutar este comando, sustituya la clave de ejemplo por los identificadores de clave IDs válidos de su cuenta. AWS

```
aws kms re-encrypt \  
  --ciphertext-blob fileb://ExampleEncryptedFile \  
  --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
  --query CiphertextBlob \  
  --output text | base64 --decode > ExampleReEncryptedFile
```

Este comando no genera ninguna salida. La salida del comando `re-encrypt` se descodifica en base64 y se guarda en un archivo.

Para obtener más información, consulta la referencia [ReEncrypt](#) de la API del Servicio de administración de AWS claves.

Ejemplo 2: Cómo volver a cifrar un mensaje cifrado con una clave de KMS simétrica diferente (Linux y macOS).

El siguiente comando `re-encrypt` de ejemplo es el mismo que el anterior, excepto que usa la utilidad `certutil` para descodificar en Base64 los datos de texto sin formato. Como se muestra en los siguientes ejemplos, este procedimiento requiere dos comandos.

Antes de ejecutar este comando, sustituya la ID de clave de ejemplo por una ID de clave válida de su AWS cuenta.

```
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
```

A continuación, use la utilidad `certutil`

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Salida:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Para obtener más información, consulta la referencia [ReEncrypt](#) de la API del Servicio de administración de AWS claves.

- Para obtener más información sobre la API, consulte [ReEncrypt](#) la Referencia de AWS CLI comandos.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def re_encrypt(self, source_key_id, cipher_text):
        """
        Takes ciphertext previously encrypted with one key and reencrypt it by
        using
        another key.
```

```

        :param source_key_id: The ARN or ID of the original key used to encrypt
the
                                ciphertext.
        :param cipher_text: The encrypted ciphertext.
        :return: The ciphertext encrypted by the second key.
        """
        destination_key_id = input(
            f"Your ciphertext is currently encrypted with key {source_key_id}. "
            f"Enter another key ID or ARN to reencrypt it: "
        )
        if destination_key_id != "":
            try:
                cipher_text = self.kms_client.re_encrypt(
                    SourceKeyId=source_key_id,
                    DestinationKeyId=destination_key_id,
                    CiphertextBlob=cipher_text,
                )["CiphertextBlob"]
            except ClientError as err:
                logger.error(
                    "Couldn't reencrypt your ciphertext. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            else:
                print(f"Reencrypted your ciphertext as: {cipher_text}")
                return cipher_text
        else:
            print("Skipping reencryption demo.")

```

- Para obtener más información sobre la API, consulta [ReEncrypt](#) la AWS Referencia de API de SDK for Python (Boto3).

Ruby

SDK para Ruby

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Human-readable version of the ciphertext of the data to reencrypt.

blob =
  '01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593'
sourceCiphertextBlob = [blob].pack('H*')

# Replace the fictitious key ARN with a valid key ID

destinationKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.re_encrypt({
                        ciphertext_blob: sourceCiphertextBlob,
                        destination_key_id: destinationKeyId
                      })

# Display a readable version of the resulting re-encrypted blob.
puts 'Blob:'
puts resp.ciphertext_blob.unpack('H*')
```

- Para obtener más información sobre la API, consulta [ReEncrypt](#) la Referencia AWS SDK para Ruby de la API.

Rust

SDK para Rust

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
async fn reencrypt_string(
    verbose: bool,
    client: &Client,
    input_file: &str,
    output_file: &str,
    first_key: &str,
    new_key: &str,
) -> Result<(), Error> {
    // Get blob from input file
    // Open input text file and get contents as a string
    // input is a base-64 encoded string, so decode it:
    let data = fs::read_to_string(input_file)
        .map(|input_file| base64::decode(input_file).expect("invalid base 64"))
        .map(Blob::new);

    let resp = client
        .re_encrypt()
        .ciphertext_blob(data.unwrap())
        .source_key_id(first_key)
        .destination_key_id(new_key)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);
    let o = &output_file;

    let mut ofile = File::create(o).expect("unable to create file");
    ofile.write_all(s.as_bytes()).expect("unable to write");
}
```

```

    if verbose {
        println!("Wrote the following to {}: ", output_file);
        println!("{}", s);
    } else {
        println!("Wrote base64-encoded output to {}", output_file);
    }

    Ok(())
}

```

- Para obtener más información sobre la API, consulta [ReEncrypt](#) la referencia sobre la API de AWS SDK para Rust.

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    " iv_source_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    " iv_destination_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'
    " iv_ciphertext_blob contains the encrypted data
    oo_result = lo_kms->reencrypt(
        iv_sourcekeyid = iv_source_key_id
        iv_destinationkeyid = iv_destination_key_id
        iv_ciphertextblob = iv_ciphertext_blob
    ).
    MESSAGE 'Ciphertext reencrypted successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.
    MESSAGE 'The key is disabled.' TYPE 'E'.
CATCH /aws1/cx_kmsincorrectkeyex.
    MESSAGE 'Incorrect source key for decryption.' TYPE 'E'.

```

```
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [ReEncrypt](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **RetireGrant** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `RetireGrant`.

CLI

AWS CLI

Retirada de una concesión en una clave maestra de cliente

En el siguiente ejemplo de `retire-grant` se elimina una concesión de una clave de KMS.

El siguiente comando de ejemplo especifica los parámetros `grant-id` y `key-id`. El valor del parámetro `key-id` debe ser el ARN de la clave de 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 no genera ninguna salida. Utilice el comando `list-grants` para confirmar que la concesión se ha retirado.

Para obtener más información, consulte [Retiro y revocación de concesiones](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte [RetireGrant](#) la Referencia de AWS CLI comandos.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def retire_grant(self, grant):
        """
        Retires a grant so that it can no longer be used.

        :param grant: The grant to retire.
        """
        try:
            self.kms_client.retire_grant(GrantToken=grant["GrantToken"])
        except ClientError as err:
            logger.error(
                "Couldn't retire grant %s. Here's why: %s",
                grant["GrantId"],
                err.response["Error"]["Message"],
            )
        else:
            print(f"Grant {grant['GrantId']} retired.")
```

- Para obtener más información sobre la API, consulta [RetireGrant](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    " iv_grant_token = 'AQpAM2RhZ...'  
    lo_kms->retiregrant( iv_granttoken = iv_grant_token ).  
    MESSAGE 'Grant retired successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Grant not found.' TYPE 'E'.  
CATCH /aws1/cx_kmsinvgranttokenex.  
    MESSAGE 'Invalid grant token.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [RetireGrant](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **RevokeGrant** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar RevokeGrant.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

CLI

AWS CLI

Revocación de una concesión en una clave maestra de cliente

En el siguiente ejemplo de `revoke-grant` se elimina una concesión de una clave de KMS. El siguiente comando de ejemplo especifica los parámetros `grant-id` y `key-id`. El valor del parámetro `key-id` puede ser el ID de clave o el ARN de la clave de KMS.

```
aws kms revoke-grant \
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando no genera ninguna salida. Utilice el comando `list-grants` para confirmar que la concesión se ha revocado.

Para obtener más información, consulte [Retiro y revocación de concesiones](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte [RevokeGrant](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Revokes a grant for the specified AWS KMS key asynchronously.
```

```

*
* @param keyId The ID or key ARN of the AWS KMS key.
* @param grantId The identifier of the grant to be revoked.
* @return A {@link CompletableFuture} representing the asynchronous
operation of revoking the grant.
* The {@link CompletableFuture} will complete with a {@link
RevokeGrantResponse} object
* if the operation is successful, or with a {@code null} value if an
error occurs.
*/
public CompletableFuture<RevokeGrantResponse> revokeKeyGrantAsync(String
keyId, String grantId) {
    RevokeGrantRequest grantRequest = RevokeGrantRequest.builder()
        .keyId(keyId)
        .grantId(grantId)
        .build();

    CompletableFuture<RevokeGrantResponse> responseFuture =
getAsyncClient().revokeGrant(grantRequest);
    responseFuture.whenComplete((response, exception) -> {
        if (exception == null) {
            logger.info("Grant ID: [" + grantId + "] was successfully
revoked!");
        } else {
            if (exception instanceof KmsException kmsEx) {
                if (kmsEx.getMessage().contains("Grant does not exist")) {
                    logger.info("The grant ID '" + grantId + "' does not
exist. Moving on...");
                } else {
                    throw new RuntimeException("KMS error occurred: " +
kmsEx.getMessage(), kmsEx);
                }
            } else {
                throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
            }
        }
    });

    return responseFuture;
}

```

- Para obtener más información sobre la API, consulta [RevokeGrant](#) la Referencia AWS SDK for Java 2.x de la API.

PHP

SDK para PHP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * @param string $grantId
 * @param string $keyId
 * @return void
 */
public function revokeGrant(string $grantId, string $keyId)
{
    try{
        $this->client->revokeGrant([
            'GrantId' => $grantId,
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
        echo "There was a problem with revoking the grant: {$caught-
>getAwsErrorMessage()}.\\n";
        throw $caught;
    }
}
```

- Para obtener más información sobre la API, consulta [RevokeGrant](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def revoke_grant(self, key_id: str, grant_id: str) -> None:
        """
        Revokes a grant so that it can no longer be used.

        :param key_id: The ARN or ID of the key associated with the grant.
        :param grant_id: The ID of the grant to revoke.
        """
        try:
            self.kms_client.revoke_grant(KeyId=key_id, GrantId=grant_id)
        except ClientError as err:
            logger.error(
                "Couldn't revoke grant %s. Here's why: %s",
                grant_id,
                err.response["Error"]["Message"],
            )
            raise
```

- Para obtener más información sobre la API, consulta [RevokeGrant](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    " iv_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'  
    " iv_grant_id = '1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p'  
    lo_kms->revokegrant(  
        iv_keyid = iv_key_id  
        iv_grantid = iv_grant_id  
    ).  
    MESSAGE 'Grant revoked successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Grant or key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmsinvalidgrantidex.  
    MESSAGE 'Invalid grant ID.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [RevokeGrant](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **ScheduleKeyDeletion** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `ScheduleKeyDeletion`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

CLI

AWS CLI

Programación de la eliminación de una clave de KMS administrada por el cliente.

En el siguiente ejemplo de `schedule-key-deletion` se programa la eliminación de la clave de KMS administrada por el cliente especificada en 15 días.

El parámetro `--key-id` identifica las claves KMS. En este ejemplo, se utiliza un valor de ARN clave, pero puede utilizar el ID de clave o el ARN de la clave de KMS. El parámetro `--pending-window-in-days` especifica la duración del período de espera de 7 a 30 días. De forma predeterminada, el periodo de espera es de 30 días. En este ejemplo se especifica un valor de 15, que indica que hay que AWS eliminar permanentemente la clave KMS 15 días después de que se complete el comando.

```
aws kms schedule-key-deletion \
  --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
  --pending-window-in-days 15
```

La respuesta incluye el ARN de la clave, el estado de la clave, el período de espera (`PendingWindowInDays`) y la fecha de eliminación en tiempo Unix. Para ver la fecha de eliminación en hora local, utilice la consola AWS KMS. Las claves KMS en el estado de clave `PendingDeletion` no se pueden utilizar en operaciones 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 obtener más información, consulte [Eliminación de claves](#) en la Guía para desarrolladores de AWS Key Management Service.

- Para obtener más información sobre la API, consulte [ScheduleKeyDeletion](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * Deletes a KMS key asynchronously.
 *
 * <p><strong>Warning:</strong> Deleting a KMS key is a destructive and
potentially dangerous operation.
 * When a KMS key is deleted, all data that was encrypted under the KMS key
becomes unrecoverable.
 * This means that any files, databases, or other data that were encrypted
using the deleted KMS key
 * will become permanently inaccessible. Exercise extreme caution when
deleting KMS keys.</p>
 *
 * @param keyId the ID of the KMS key to delete
 * @return a {@link CompletableFuture} that completes when the key deletion
is scheduled
 */
public CompletableFuture<Void> deleteKeyAsync(String keyId) {
    ScheduleKeyDeletionRequest deletionRequest =
ScheduleKeyDeletionRequest.builder()

```

```

        .keyId(keyId)
        .pendingWindowInDays(7)
        .build();

return getAsyncClient().scheduleKeyDeletion(deletionRequest)
    .thenRun(() -> {
        logger.info("Key {} will be deleted in 7 days", keyId);
    })
    .exceptionally(throwable -> {
        throw new RuntimeException("Failed to schedule key deletion for
key ID: " + keyId, throwable);
    });
}

```

- Para obtener más información sobre la API, consulta [ScheduleKeyDeletion](#) la Referencia AWS SDK for Java 2.x de la API.

PHP

SDK para PHP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * @param string $keyId
 * @param int $pendingWindowInDays
 * @return void
 */
public function scheduleKeyDeletion(string $keyId, int $pendingWindowInDays =
7)
{
    try {
        $this->client->scheduleKeyDeletion([
            'KeyId' => $keyId,
            'PendingWindowInDays' => $pendingWindowInDays,

```

```

    });
    }catch(KmsException $caught){
        echo "There was a problem scheduling the key deletion: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

```

- Para obtener más información sobre la API, consulta [ScheduleKeyDeletion](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def delete_key(self, key_id: str, window: int) -> None:

```

```
"""
Deletes a list of keys.

Warning:
Deleting a KMS key is a destructive and potentially dangerous operation.
When a KMS key is deleted,
all data that was encrypted under the KMS key is unrecoverable.

:param key_id: The ARN or ID of the key to delete.
:param window: The waiting period, in days, before the KMS key is
deleted.
"""

try:
    self.kms_client.schedule_key_deletion(
        KeyId=key_id, PendingWindowInDays=window
    )
except ClientError as err:
    logging.error(
        "Couldn't delete key %s. Here's why: %s",
        key_id,
        err.response["Error"]["Message"],
    )
    raise
```

- Para obtener más información sobre la API, consulta [ScheduleKeyDeletion](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

TRY.

```

    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
    " iv_pending_window_days = 7
    oo_result = lo_kms->schedulekeydeletion(
        iv_keyid = iv_key_id
        iv_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 obtener más información sobre la API, consulte [ScheduleKeyDeletion](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **Sign** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar Sign.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

CLI

AWS CLI

Ejemplo 1: Cómo generar una firma digital para un mensaje

El siguiente ejemplo de `sign` genera una firma criptográfica para un mensaje corto. El resultado del comando incluye un campo de Signature codificado en Base64 que puede comprobar mediante el comando `verify`.

Debe especificar un mensaje para firmar y un algoritmo de firma que admita su clave KMS asimétrica. Para obtener los algoritmos de firma de su clave KMS, utilice el comando `describe-key`.

En AWS CLI v2, el valor del `message` parámetro debe estar codificado en Base64. O bien, puede guardar el mensaje en un archivo y usar el `fileb://` prefijo, que indica a la AWS CLI que lea los datos binarios del archivo.

Antes de ejecutar este comando, sustituya la ID de clave de ejemplo por una ID de clave válida de su AWS cuenta. El ID de clave debe representar una clave KMS asimétrica con un uso de clave de `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
```

Salida:

```
{
  "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 obtener más información sobre el uso de claves KMS asimétricas en AWS KMS, consulte Claves [asimétricas en AWS KMS en la Guía para desarrolladores del Servicio de administración de AWS claves](#).

Ejemplo 2: guardar una firma digital en un archivo (Linux y macOS)

El siguiente ejemplo de `sign` genera una firma criptográfica para un mensaje corto almacenado en un archivo local. El comando también obtiene la `Signature` propiedad de la respuesta, la decodifica en Base64 y la guarda en el archivo. `ExampleSignature` Puede usar el archivo de firmas en un comando `verify` que verifique la firma.

El comando `sign` requiere un mensaje codificado en Base64 y un algoritmo de firma que admita su clave KMS asimétrica. Para obtener los algoritmos de firma que admite su clave KMS, utilice el comando `describe-key`.

Antes de ejecutar este comando, sustituya la ID de clave de ejemplo por una ID de clave válida de su cuenta. AWS El ID de clave debe representar una clave KMS asimétrica con un uso de clave de `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 no genera ninguna salida. En este ejemplo, se extrae la propiedad `Signature` del resultado y se guarda en un archivo.

Para obtener más información sobre el uso de claves KMS asimétricas en AWS KMS, consulte Claves [asimétricas en AWS KMS en la Guía para desarrolladores del Servicio de administración de AWS claves](#).

- Para obtener detalles de la API, consulte [Sign](#) en la Referencia de comandos de la AWS CLI .

Java

SDK para Java 2.x

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**
 * Asynchronously signs and verifies data using AWS KMS.
```

```

*
* <p>The method performs the following steps:
* <ol>
*     <li>Creates an AWS KMS key with the specified key spec, key usage, and
origin.</li>
*     <li>Signs the provided message using the created KMS key and the
RSASSA-PSS-SHA-256 algorithm.</li>
*     <li>Verifies the signature of the message using the created KMS key
and the RSASSA-PSS-SHA-256 algorithm.</li>
* </ol>
*
* @return a {@link CompletableFuture} that completes with the result of the
signature verification,
*     {@code true} if the signature is valid, {@code false} otherwise.
* @throws KmsException if any error occurs during the KMS operations.
* @throws RuntimeException if an unexpected error occurs.
*/
public CompletableFuture<Boolean> signVerifyDataAsync() {
    String signMessage = "Here is the message that will be digitally signed";

    // Create an AWS KMS key used to digitally sign data.
    CreateKeyRequest createKeyRequest = CreateKeyRequest.builder()
        .keySpec(KeySpec.RSA_2048)
        .keyUsage(KeyUsageType.SIGN_VERIFY)
        .origin(OriginType.AWS_KMS)
        .build();

    return getAsyncClient().createKey(createKeyRequest)
        .thenCompose(createKeyResponse -> {
            String keyId = createKeyResponse.keyMetadata().keyId();

            SdkBytes messageBytes = SdkBytes.fromString(signMessage,
Charset.defaultCharset());
            SignRequest signRequest = SignRequest.builder()
                .keyId(keyId)
                .message(messageBytes)
                .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                .build();

            return getAsyncClient().sign(signRequest)
                .thenCompose(signResponse -> {
                    byte[] signedBytes =
signResponse.signature().asByteArray();

```



```

        VerifyRequest verifyRequest = VerifyRequest.builder()
            .keyId(keyId)

            .message(SdkBytes.fromByteArray(signMessage.getBytes(Charset.defaultCharset())))

            .signature(SdkBytes.fromByteBuffer(ByteBuffer.wrap(signedBytes)))

            .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                .build();

        return getAsyncClient().verify(verifyRequest)
            .thenApply(verifyResponse -> {
                return (boolean) verifyResponse.signatureValid();
            });
    });
}

.throwable);
}
}

```

- Para obtener información sobre la API, consulte [Sign](#) en la referencia de la API de AWS SDK for Java 2.x .

PHP

SDK para PHP

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * @param string $keyId
 * @param string $message
 * @param string $algorithm

```

```

    * @return Result
    */
    public function sign(string $keyId, string $message, string $algorithm)
    {
        try {
            return $this->client->sign([
                'KeyId' => $keyId,
                'Message' => $message,
                'SigningAlgorithm' => $algorithm,
            ]);
        } catch (KmsException $caught){
            echo "There was a problem signing the data: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
}

```

- Para obtener información sobre la API, consulte [Sign](#) en la referencia de la API de AWS SDK para PHP .

Python

SDK para Python (Boto3)

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

```

```

        :return: An instance of KeyEncrypt initialized with the default KMS
client.
    """
    kms_client = boto3.client("kms")
    return cls(kms_client)

def sign(self, key_id: str, message: str) -> str:
    """
    Signs a message with a key.

    :param key_id: The ARN or ID of the key to use for signing.
    :param message: The message to sign.
    :return: The signature of the message.
    """
    try:
        return self.kms_client.sign(
            KeyId=key_id,
            Message=message.encode(),
            SigningAlgorithm="RSASSA_PSS_SHA_256",
        )["Signature"]
    except ClientError as err:
        logger.error(
            "Couldn't sign your message. Here's why: %s",
            err.response["Error"]["Message"],
        )
        raise

```

- Para obtener información sobre la API, consulte [Sign](#) en la referencia de la API de AWS SDK para Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

TRY.
    " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
    " iv_message contains the message to sign
    " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
    oo_result = lo_kms->sign(
        iv_keyid = iv_key_id
        iv_message = iv_message
        iv_signingalgorithm = iv_signing_algorithm
    ).
    MESSAGE 'Message signed successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsdisabledexception.
    MESSAGE 'The key is disabled.' TYPE 'E'.
CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmsinvalidkeyusageex.
    MESSAGE 'Key cannot be used for signing.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte la referencia sobre el AWS SDK de inicio de [sesión](#) para la API ABAP de SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **TagResource** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar TagResource.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- [Conceptos básicos](#)

CLI

AWS CLI

Cómo agregar una etiqueta a una clave KMS

En el siguiente ejemplo de `tag-resource`, se agregan las etiquetas `"Purpose": "Test"` y `"Dept": "IT"` a una clave KMS administrada por el cliente. Puede usar etiquetas como estas para etiquetar las claves KMS y crear categorías de claves KMS para permisos y auditoría.

Para especificar la clave de KMS, utilice el parámetro `key-id`. En este ejemplo, se utiliza un valor de ID de clave, pero puede utilizar un ID de clave o un ARN de clave en este comando.

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

Este comando no genera ninguna salida. Para ver las etiquetas de una clave AWS KMS de KMS, utilice el `list-resource-tags` comando.

Para obtener más información sobre el uso de etiquetas en AWS KMS, consulte [Etiquetado de claves](#) en la Guía para desarrolladores del Servicio de administración de AWS claves.

- Para obtener más información sobre la API, consulte [TagResource](#) la Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
/**  
 * Asynchronously tags a KMS key with a specific tag.  
 *  
 * @param keyId the ID of the KMS key to be tagged
```

```

    * @return a {@link CompletableFuture} that completes when the tagging
    operation is finished
    */
    public CompletableFuture<Void> tagKMSKeyAsync(String keyId) {
        Tag tag = Tag.builder()
            .tagKey("Environment")
            .tagValue("Production")
            .build();

        TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
            .keyId(keyId)
            .tags(tag)
            .build();

        return getAsyncClient().tagResource(tagResourceRequest)
            .thenRun(() -> {
                logger.info("{} key was tagged", keyId);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to tag the KMS key",
throwable);
            });
    }
}

```

- Para obtener más información sobre la API, consulta [TagResource](#) la Referencia AWS SDK for Java 2.x de la API.

PHP

SDK para PHP

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

/**
 * @param string $keyId

```

```

    * @param array $tags
    * @return void
    */
    public function tagResource(string $keyId, array $tags)
    {
        try {
            $this->client->tagResource([
                'KeyId' => $keyId,
                'Tags' => $tags,
            ]);
        } catch (KmsException $caught){
            echo "There was a problem applying the tag(s): {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
}

```

- Para obtener más información sobre la API, consulta [TagResource](#) la Referencia AWS SDK para PHP de la API.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

```

```

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

def tag_resource(self, key_id: str, tag_key: str, tag_value: str) -> None:
    """
    Add or edit tags on a customer managed key.

    :param key_id: The ARN or ID of the key to enable rotation for.
    :param tag_key: Key for the tag.
    :param tag_value: Value for the tag.
    """
    try:
        self.kms_client.tag_resource(
            KeyId=key_id, Tags=[{"TagKey": tag_key, "TagValue": tag_value}]
        )
    except ClientError as err:
        logging.error(
            "Couldn't add a tag for the key '%s'. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise

```

- Para obtener más información sobre la API, consulta [TagResource](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).


```

DATA lt_tags TYPE /aws1/cl_kmstag=>tt_taglist.

TRY.
  " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
  " iv_tag_key = 'Environment'
  " iv_tag_value = 'Production'
  APPEND NEW /aws1/cl_kmstag(
    iv_tagkey = iv_tag_key
    iv_tagvalue = iv_tag_value
  ) TO lt_tags.

  lo_kms->tagresource(
    iv_keyid = iv_key_id
    it_tags = lt_tags
  ).
  MESSAGE 'Tag added to KMS key successfully.' TYPE 'I'.
CATCH /aws1/cx_kmsnotfoundexception.
  MESSAGE 'Key not found.' TYPE 'E'.
CATCH /aws1/cx_kmstagexception.
  MESSAGE 'Invalid tag format.' TYPE 'E'.
CATCH /aws1/cx_kmskmsinternalex.
  MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.

```

- Para obtener más información sobre la API, consulte [TagResource](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **UpdateAlias** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar UpdateAlias.

CLI

AWS CLI

Asociación de un alias a una clave de KMS diferente

En el siguiente ejemplo de `update-alias` se asocia el alias `alias/test-key` a una clave de KMS diferente.

El parámetro `--alias-name` especifica el alias. El valor del nombre del alias debe empezar por `alias/`. El parámetro `--target-key-id` especifica la clave de KMS que se va a asociar al alias. No es necesario especificar la clave de KMS actual para el alias.

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

Este comando no genera ninguna salida. Utilice el comando `list-aliases` para encontrar el alias.

Para obtener más información, consulte [Actualización de alias](#) en la AWS Guía para desarrolladores de Key Management Service.

- Para obtener más información sobre la API, consulte [UpdateAlias](#) la Referencia de AWS CLI comandos.

Python

SDK para Python (Boto3)

Note

Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class AliasManager:  
    def __init__(self, kms_client):  
        self.kms_client = kms_client  
        self.created_key = None  
  
    @classmethod  
    def from_client(cls) -> "AliasManager":  
        """  
        Creates an AliasManager instance with a default KMS client.
```

```

        :return: An instance of AliasManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def update_alias(self, alias, current_key_id):
        """
        Updates an alias by assigning it to another key.

        :param alias: The alias to reassign.
        :param current_key_id: The ARN or ID of the key currently associated with
the alias.
        """
        new_key_id = input(
            f"Alias {alias} is currently associated with {current_key_id}. "
            f"Enter another key ID or ARN that you want to associate with
{alias}: "
        )
        if new_key_id != "":
            try:
                self.kms_client.update_alias(AliasName=alias,
TargetKeyId=new_key_id)
            except ClientError as err:
                logger.error(
                    "Couldn't associate alias %s with key %s. Here's why: %s",
                    alias,
                    new_key_id,
                    err.response["Error"]["Message"],
                )
            else:
                print(f"Alias {alias} is now associated with key {new_key_id}.")
        else:
            print("Skipping alias update.")

```

- Para obtener más información sobre la API, consulta [UpdateAlias](#) la AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
TRY.  
    " iv_alias_name = 'alias/my-key-alias'  
    " iv_target_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'  
    lo_kms->updatealias(  
        iv_aliasname = iv_alias_name  
        iv_targetkeyid = iv_target_key_id  
    ).  
    MESSAGE 'Alias updated successfully.' TYPE 'I'.  
CATCH /aws1/cx_kmsnotfoundexception.  
    MESSAGE 'Alias or key not found.' TYPE 'E'.  
CATCH /aws1/cx_kmskmsinternalex.  
    MESSAGE 'An internal error occurred.' TYPE 'E'.  
ENDTRY.
```

- Para obtener más información sobre la API, consulte [UpdateAlias](#) la referencia sobre la API ABAP del AWS SDK para SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **Verify** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar Verify.

CLI

AWS CLI

Cómo verificar una firma digital

En el siguiente ejemplo de `verify`, se verifica una firma criptográfica para un mensaje corto codificado en Base64. El identificador de clave, el mensaje, el tipo de mensaje y el algoritmo de firma deben ser los mismos que se usaron para firmar el mensaje.

En AWS CLI v2, el valor del `message` parámetro debe estar codificado en Base64. O bien, puede guardar el mensaje en un archivo y usar el `fileb://` prefijo, que indica a la AWS CLI que lea los datos binarios del archivo.

La firma que especifique no puede estar codificada en base64. Para obtener ayuda para decodificar la firma que devuelve el comando `sign`, consulte los ejemplos de comandos `sign`.

El resultado del comando incluye un campo `SignatureValid` booleano que indica que se verificó la firma. Si se produce un error en la validación de la firma, también se produce un error en el comando `verify`.

Antes de ejecutar este comando, sustituya la ID de clave de ejemplo por una ID de clave válida de su AWS cuenta.

```
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
```

Salida:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "SignatureValid": true,
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Para obtener más información sobre el uso de claves KMS asimétricas en AWS KMS, consulte [Uso de claves asimétricas](#) en la Guía para desarrolladores del Servicio de administración de AWS claves.

- Para obtener información sobre la API, consulte [Verify](#) en la Referencia de comandos de la AWS CLI .

Python

SDK para Python (Boto3)

Note

Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def verify(self, key_id: str, message: str, signature: str) -> bool:
        """
        Verifies a signature against a message.

        :param key_id: The ARN or ID of the key used to sign the message.
        :param message: The message to verify.
        :param signature: The signature to verify.
        :return: True when the signature matches the message, otherwise False.
        """
```

```

try:
    response = self.kms_client.verify(
        KeyId=key_id,
        Message=message.encode(),
        Signature=signature,
        SigningAlgorithm="RSASSA_PSS_SHA_256",
    )
    valid = response["SignatureValid"]
    print(f"The signature is {'valid' if valid else 'invalid'}.")
    return valid
except ClientError as err:
    if err.response["Error"]["Code"] == "SignatureDoesNotMatchException":
        print("The signature is not valid.")
    else:
        logger.error(
            "Couldn't verify your signature. Here's why: %s",
            err.response["Error"]["Message"],
        )
    raise

```

- Para obtener información sobre la API, consulte [Verify](#) en la Referencia de la API de AWS SDK para Python (Boto3).

SAP ABAP

SDK para SAP ABAP

Note

Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el [Repositorio de ejemplos de código de AWS](#).

TRY.

```

" iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
" iv_message contains the original message
" iv_signature contains the signature to verify
" iv_signing_algorithm = 'RSASSA_PSS_SHA_256'

```

```

        oo_result = lo_kms->verify(
            iv_keyid = iv_key_id
            iv_message = iv_message
            iv_signature = iv_signature
            iv_signingalgorithm = iv_signing_algorithm
        ).
    DATA(lv_valid) = oo_result->get_signaturevalid( ).
    IF lv_valid = abap_true.
        MESSAGE 'Signature is valid.' TYPE 'I'.
    ELSE.
        MESSAGE 'Signature is invalid.' TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
    CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
    CATCH /aws1/cx_kmskmsinvalidsigex.
        MESSAGE 'Invalid signature.' TYPE 'E'.
    CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.

```

- Para obtener más información sobre la API, consulte [Verificar](#) en el AWS SDK para obtener información sobre la API ABAP de SAP.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Escenarios de AWS KMS uso AWS SDKs

Los siguientes ejemplos de código muestran cómo implementar escenarios comunes en AWS KMS with AWS SDKs. Estos escenarios muestran cómo realizar tareas específicas mediante la invocación de varias funciones internas AWS KMS o combinadas con otros Servicios de AWS. En cada escenario se incluye un enlace al código fuente completo, con instrucciones de configuración y ejecución del código.

Los escenarios requieren un nivel intermedio de experiencia para entender las acciones de servicio en su contexto.

Ejemplos

- [Trabaje con el cifrado de tablas de DynamoDB mediante la versión 2 AWS Command Line Interface](#)

Trabaje con el cifrado de tablas de DynamoDB mediante la versión 2 AWS Command Line Interface

En el siguiente ejemplo de código se muestra cómo administrar las opciones de cifrado de las tablas de DynamoDB.

- Cree una tabla con un cifrado predeterminado.
- Cree una tabla con una CMK administrada por el cliente.
- Actualice la configuración de cifrado de la tabla.
- Describa el cifrado de la tabla.

Bash

AWS CLI con el script Bash

Cree una tabla con un cifrado predeterminado.

```
# 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
```

Cree una tabla con una CMK administrada por el cliente.

```
# 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
```

Actualice el cifrado de la tabla.

```
# 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
```

Describa el cifrado de la tabla.

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
  --table-name CustomerData \
  --query "Table.SSEDescription"
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de comandos de AWS CLI .
 - [CreateKey](#)
 - [CreateTable](#)
 - [DescribeTable](#)
 - [UpdateTable](#)

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte [Uso de este servicio con un AWS SDK](#). En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Soporte de atestación criptográfica en AWS KMS

AWS KMS [admite la certificación criptográfica para Nitro Enclaves y NitroTPM](#). Las aplicaciones que admiten estos métodos de certificación realizan las siguientes operaciones criptográficas con un documento de certificación firmado. AWS KMS verifica que el documento de certificación proviene de una fuente válida (Nitro enclave o NitroTPM). A continuación, en lugar de devolver datos de texto sin formato en la respuesta, estas API cifran el texto sin formato con la clave pública del documento de certificación y devuelven texto cifrado solo con la clave privada correspondiente en el enclave o instancia EC2.

- [Decrypt](#)
- [DeriveSharedSecret](#)
- [GenerateDataKey](#)
- [GenerateDataKeyPair](#)
- [GenerateRandom](#)

En la siguiente tabla, se muestra en qué se diferencia la respuesta a las solicitudes certificadas de la respuesta estándar para cada operación de API.

AWS KMS operación	Respuesta estándar	Respuesta a las solicitudes certificadas
Decrypt	Devuelve datos de texto sin formato	Devuelve los datos de texto sin formato cifrados por la clave pública del documento de certificación
DeriveSharedSecret	Devuelve un secreto compartido sin formato	Devuelve el secreto compartido o sin formato por la clave pública del documento de certificación
GenerateDataKey	Devuelve una copia en texto sin formato de la clave de datos	Devuelve una copia de la clave de datos cifrada por la

AWS KMS operación	Respuesta estándar	Respuesta a las solicitudes certificadas
	(También devuelve una copia de la clave de datos cifrada por una clave de KMS)	clave pública del documento de certificación (También devuelve una copia de la clave de datos cifrada por una clave de KMS)
GenerateDataKeyPair	Devuelve una copia de texto sin formato de la clave privada (También devuelve la clave pública y una copia de la clave privada cifrada por una clave de KMS)	Devuelve una copia de la clave privada cifrada por la clave pública del documento de certificación (También devuelve la clave pública y una copia de la clave privada cifrada por una clave de KMS)
GenerateRandom	Devuelve una cadena de bytes aleatoria	Devuelve la cadena de bytes aleatoria cifrada por la clave pública del documento de certificación

AWS KMS admite [claves de condición de política](#) que se pueden utilizar para permitir o denegar operaciones certificadas con una AWS KMS clave basada en el contenido del documento de certificación. También puede [supervisar las solicitudes certificadas](#) en sus registros. AWS KMS AWS CloudTrail

Más información

- [Certificación criptográfica](#)
- [AWS KMS claves de condición para plataformas certificadas](#)
- [Cómo realizar llamadas certificadas a AWS KMS](#)
- [Monitoreo de solicitudes certificadas](#)

Cómo realizar llamadas certificadas a AWS KMS

Para realizar una llamada certificada a AWS KMS, utilice el `Recipient` parámetro de la solicitud para proporcionar el documento de certificación firmado y el algoritmo de cifrado que se utilizará con la clave pública en el documento de certificación. Cuando una solicitud incluye el parámetro `Recipient` con un documento de certificación firmado, la respuesta incluye un campo `CiphertextForRecipient` con el texto cifrado mediante la clave pública. El campo de texto sin formato es nulo o está vacío.

El `Recipient` parámetro debe especificar un documento de certificación firmado por Nitro Enclaves o NitroTPM. AWS KMS se basa en la firma digital del documento de certificación para demostrar que la clave pública de la solicitud proviene de una fuente válida. No puede proporcionar su propio certificado para firmar digitalmente el documento de certificación.

El SDK de AWS Nitro Enclaves, que solo es compatible con un enclave de Nitro, añade automáticamente el `Recipient` parámetro y sus valores a cada solicitud. AWS KMS

Para realizar solicitudes certificadas en AWS SDKs, debe especificar el `Recipient` parámetro y sus valores. [El documento de certificación se puede recuperar del NitroTPM mediante la `nitro-tpm-attest` utilidad o del Nitro Secure Module \(NSM\) mediante la API de NSM.](#)

AWS KMS admite [claves de condición de política](#) que se pueden utilizar para permitir o denegar operaciones certificadas con una AWS KMS clave basada en el contenido del documento de certificación. También puede [supervisar las solicitudes certificadas](#) en sus registros. AWS KMS AWS CloudTrail

Para obtener información detallada sobre el `Recipient` parámetro y el campo de `CiphertextForRecipient` respuesta de AWS, consulte [Decrypt](#), [DeriveSharedSecretGenerateDataKeyGenerateDataKeyPair](#), y [GenerateRandom](#) los temas de la AWS Key Management Service API Reference, el SDK de [AWS Nitro Enclaves o cualquier SDK](#). Para obtener información sobre cómo configurar los datos y las claves de datos para el cifrado, consulte [Uso de la atestación criptográfica con](#). AWS KMS

Monitoreo de solicitudes certificadas

Puede usar sus AWS CloudTrail registros para supervisar las [GenerateRandom](#) operaciones de [Decrypt](#), [DeriveSharedSecretGenerateDataKeyGenerateDataKeyPair](#), y que utilizan la atestación. En estas entradas de registro, el campo `additionalEventData` tiene un campo `recipient` con

información del documento de certificación en la solicitud. Estos campos se incluyen solo cuando el parámetro `Recipient` de la solicitud especifica un documento de certificación firmado.

La información específica incluida en el CloudTrail registro depende del método de certificación utilizado.

Supervisión de las solicitudes para enclaves de Nitro

Para la certificación del enclave Nitro, el CloudTrail registro incluye el ID del módulo (`attestationDocumentModuleId`), el resumen de la imagen () y los registros de configuración de la plataforma (PCRs) del documento de certificación.
`attestationDocumentEnclaveImageDigest`

El ID del módulo es el [ID del enclave](#) de Nitro. El resumen de la imagen es el SHA384 hash de la imagen del enclave. Puede utilizar el resumen de la imagen y los valores de PCR en [condiciones para las políticas de claves y las políticas de IAM](#). Para obtener información sobre el PCRs, consulte [Dónde obtener las medidas de un enclave](#) en la Guía del usuario de AWS Nitro Enclaves.

En esta sección se muestra un ejemplo de entrada de CloudTrail registro para cada una de las solicitudes de enclave de Nitro admitidas a. AWS KMS

Decrypt (para un enclave)

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una operación de [descifrado para un AWS enclave](#) de 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 un enclave)

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una [GenerateDataKey](#) operación para un enclave de 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 un enclave)

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una [GenerateDataKeyPair](#) operación para un enclave de 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 un enclave)

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una [GenerateRandom](#) operación para un enclave de 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"
}
```

Monitoreo de las solicitudes para NitroTPM

Para la certificación de NitroTPM, el CloudTrail registro incluye el ID del módulo (attestationDocumentModuleId) y los registros de configuración de la plataforma (PCRs) del documento de certificación.

El ID del módulo es el ID de la instancia EC2 con NitroTPM con un identificador de TPM. Puede utilizar los valores de PCR en [condiciones para las políticas de claves y las políticas de IAM](#).

En esta sección se muestra un ejemplo de entrada de CloudTrail registro para cada una de las solicitudes de NitroTPM compatibles. AWS KMS

Decrypt (para un NitroTPM)

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una operación de [descifrado para un NitroTPM](#).

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-27T22:58:24Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
```

```

    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "additionalEventData": {
    "recipient": {
      "attestationDocumentModuleId": "i-123456789abcde123-tpm0000000000000000",
      "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
      "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
      "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
      "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
      "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
      "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
    }
  },
  "requestID": "b4a65126-30d5-4b28-98b9-9153da559963",
  "eventID": "e5a2f202-ba1a-467c-b4ba-f729d45ae521",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

GenerateDataKey (para un NitroTPM)

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una [GenerateDataKey](#) operación de un 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 un NitroTPM)

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una [GenerateDataKeyPair](#) operación de un 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 un NitroTPM)

El siguiente ejemplo muestra una entrada de AWS CloudTrail registro de una [GenerateRandom](#) operación de un 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-tpm000000000000000000",
      "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"  
}
```

Uso del AWS KMS cifrado con AWS servicios

Con AWS Key Management Service, puede proporcionar claves de cifrado para proteger los datos de otros AWS servicios. El uso del AWS KMS cifrado con AWS los servicios se refiere al proceso de integración AWS KMS con otros AWS servicios para cifrar y descifrar los datos en reposo o en tránsito. Los desarrolladores, administradores de sistemas y profesionales de la seguridad podrían estar interesados en este tema para proteger los datos confidenciales almacenados o transmitidos a través de AWS los servicios, cumplir con los requisitos de conformidad normativa o implementar las mejores prácticas de cifrado. Los casos de uso más comunes incluyen el cifrado de volúmenes de Amazon EBS, buckets de Amazon S3 y bases de datos de Amazon RDS. En las siguientes secciones se describen los pasos para configurar y administrar las claves de AWS KMS cifrado para AWS servicios específicos, garantizando la confidencialidad e integridad de los datos en todo el AWS entorno. Para obtener una lista completa de los AWS servicios integrados en ellos AWS KMS, consulte Integración de [AWS servicios](#).

En los temas siguientes se describe en detalle cómo utilizan determinados servicios AWS KMS, incluidas las claves de KMS que admiten, cómo administran las claves de datos, los permisos que requieren y cómo realizar un seguimiento del uso de las claves de KMS de su cuenta por parte de cada servicio.

Important

AWS Los [servicios integrados AWS KMS utilizan](#) únicamente claves KMS de cifrado simétrico para cifrar los datos. Estos servicios no admiten cifrado con claves de KMS asimétricas. Para obtener ayuda para determinar si una clave KMS es simétrica o asimétrica, consulte [Identificación de diferentes tipos de claves](#).

Temas

- [Cómo utiliza Amazon Elastic Block Store \(Amazon EBS\) AWS KMS](#)
- [Cómo utiliza Amazon EMR AWS KMS](#)
- [Cómo utiliza Amazon Redshift AWS KMS](#)

Cómo utiliza Amazon Elastic Block Store (Amazon EBS) AWS KMS

En este tema se analiza en detalle cómo se AWS KMS utiliza [Amazon Elastic Block Store \(Amazon EBS\)](#) para cifrar volúmenes e instantáneas. Para obtener instrucciones básicas sobre el cifrado de volúmenes de Amazon EBS, consulte [Cifrado de Amazon EBS](#).

Temas

- [Cifrado de Amazon EBS](#)
- [Uso de claves KMS y claves de datos](#)
- [Contexto de cifrado de Amazon EBS](#)
- [Detección de errores de Amazon EBS](#)
- [Utilización AWS CloudFormation para crear volúmenes cifrados de Amazon EBS](#)

Cifrado de Amazon EBS

Cuando se asocia un volumen de Amazon EBS cifrado a un [tipo de instancia Amazon Elastic Compute Cloud \(Amazon EC2 compatible\)](#), se cifran los datos almacenados que están en reposo en el volumen, la E/S de disco y las instantáneas creadas a partir del volumen. El cifrado se produce en los servidores que alojan instancias de Amazon EC2.

Esta característica es compatible con todos los [tipos de volúmenes de Amazon EBS](#). A los volúmenes cifrados se obtiene acceso del mismo modo que a otros volúmenes; el cifrado y el descifrado se gestionan de forma transparente y no requieren ninguna acción de su parte, la instancia EC2 o su aplicación. Las instantáneas de los volúmenes cifrados se cifran automáticamente y los volúmenes que se crean a partir de las instantáneas cifradas también se cifran de forma automática.

El estado de cifrado de un volumen de EBS se determina al crear el volumen. No puede cambiar el estado de cifrado de un volumen existente. Sin embargo, puede [migrar datos](#) entre volúmenes cifrados y no cifrados, y aplicar un nuevo estado de cifrado al copiar una instantánea.

Amazon EBS admite el cifrado opcional de forma predeterminada. Puede activar el cifrado automáticamente en todos los volúmenes y copias instantáneas nuevos de EBS de su región y de su región. Cuenta de AWS Este ajuste de configuración no afecta a los volúmenes o instantáneas existentes. Para más información, consulte [Cifrado de Amazon EBS](#) en la Guía del usuario de Amazon EBS.

Uso de claves KMS y claves de datos

Cuando [crea un volumen Amazon EBS cifrado](#), se especifica un AWS KMS key. De manera predeterminada, Amazon EBS utiliza la [Clave administrada de AWS](#) para Amazon EBS en su cuenta (aws/ebs). Sin embargo, puede especificar una [clave administrada por el cliente](#) que puede crear y administrar.

Para utilizar una clave administrada por el cliente, debe dar permiso a Amazon EBS para usar la clave KMS en su nombre. Para obtener más información, consulte [Cómo funciona el cifrado de Amazon EBS](#) en la Guía del usuario de Amazon EBS.

Important

Amazon EBS solo admite [claves KMS simétricas](#). No se puede utilizar una [clave KMS asimétrica](#) para cifrar un volumen de Amazon EBS. Para obtener ayuda para determinar si una clave KMS es simétrica o asimétrica, consulte [Identificación de diferentes tipos de claves](#).

Para cada volumen, Amazon EBS solicita AWS KMS generar una clave de datos única cifrada con la clave de KMS que especifique. Amazon EBS almacena la clave de datos cifrada con el volumen. A continuación, al adjuntar el volumen a una instancia de Amazon EC2, Amazon EBS llama AWS KMS para descifrar la clave de datos. Amazon EBS utiliza la clave de datos de texto simple de la memoria del hipervisor para cifrar todos los discos del I/O volumen. Para obtener más información, consulte [Cómo funciona el cifrado de EBS en la Guía del usuario de Amazon EC2](#) o en la [Guía del usuario de Amazon EC2](#).

Contexto de cifrado de Amazon EBS

En sus solicitudes [GenerateDataKeyWithoutPlaintext](#) y en las de [Decrypt](#) AWS KMS, Amazon EBS utiliza un contexto de cifrado con un par nombre-valor que identifica el volumen o la instantánea de la solicitud. El nombre en el contexto de cifrado no varía.

Un [contexto de cifrado](#) es un conjunto de pares de clave-valor que contienen datos no secretos arbitrarios. Cuando incluye un contexto de cifrado en una solicitud de cifrado de datos, vincula AWS KMS criptográficamente el contexto de cifrado a los datos cifrados. Para descifrar los datos, es necesario pasar el mismo contexto de cifrado.

Para todos los volúmenes y para las instantáneas cifradas creadas con la [CreateSnapshot](#) operación Amazon EBS, Amazon EBS utiliza el ID del volumen como valor de contexto de cifrado. En el campo `requestParameters` de una entrada de registro de CloudTrail, el contexto de cifrado es similar al siguiente:

```
"encryptionContext": {  
  "aws:ebs:id": "vol-0cfb133e847d28be9"  
}
```

Para las instantáneas cifradas creadas con la operación Amazon [CopySnapshot](#) EC2, Amazon EBS utiliza el ID de la instantánea como valor de contexto de cifrado. En el campo `requestParameters` de una entrada de registro de CloudTrail, el contexto de cifrado es similar al siguiente:

```
"encryptionContext": {  
  "aws:ebs:id": "snap-069a655b568de654f"  
}
```

Detección de errores de Amazon EBS

Para crear un volumen de EBS cifrado o adjuntar el volumen a una instancia EC2, Amazon EBS y la infraestructura de Amazon EC2 deben poder usar la clave KMS especificada para el cifrado del volumen de EBS. Cuando la clave KMS no es utilizable, por ejemplo, cuando su [estado clave](#) no es `Enabled`: la creación del volumen o el adjunto al volumen fallan.

En este caso, Amazon EBS envía un evento a Amazon EventBridge (anteriormente CloudWatch Events) para notificarle el error. En EventBridge, puede establecer reglas que activen acciones automáticas en respuesta a estos eventos. Para obtener más información, consulte [Amazon CloudWatch Events para Amazon EBS](#) en la Guía del usuario de Amazon EBS, especialmente en las siguientes secciones:

- [Clave de cifrado no válida al asociar o volver a asociar un volumen](#)
- [Clave de cifrado no válida al crear el volumen](#)

Para solucionar estos errores, compruebe que esté habilitada la clave KMS que ha especificado para el cifrado del volumen de EBS. Para ello, [consulte primero la clave de KMS](#) para determinar su estado de clave actual (la columna Estado de). Consola de administración de AWS A continuación, consulte la información de uno estos enlaces:

- Si el estado de clave de la clave KMS está deshabilitado, [habilítelo](#).
- Si el estado de clave de la clave KMS está pendiente de importación, [importe el material de claves](#).
- Si el estado de clave de la clave KMS es pendiente de eliminación, [cancele la eliminación de la clave](#).

Utilización AWS CloudFormation para crear volúmenes cifrados de Amazon EBS

Puede usar [AWS CloudFormation](#) para crear volúmenes de Amazon EBS cifrados. Para obtener más información, consulte [AWS::EC2::Volume](#) en la Guía del usuario de AWS CloudFormation .

Cómo utiliza Amazon EMR AWS KMS

Cuando utilice un clúster de [Amazon EMR](#), puede configurarlo para cifrar los datos en reposo antes de guardarlos en una ubicación de almacenamiento persistente. Puede cifrar los datos en reposo en el sistema de archivos EMR (EMRFS), en los volúmenes de almacenamiento de nodos del clúster o en ambos. Para cifrar los datos en reposo, puede utilizar una AWS KMS key. En los siguientes temas se explica cómo un clúster Amazon EMR utiliza una clave KMS para cifrar los datos en reposo.

Important

Amazon EMR solo admite [claves KMS simétricas](#). No se puede utilizar una [clave KMS asimétrica](#) para cifrar datos en reposo en un clúster de Amazon EMR. Para obtener ayuda para determinar si una clave KMS es simétrica o asimétrica, consulte [Identificación de diferentes tipos de claves](#).

Los clústeres de Amazon EMR también cifran los datos en tránsito, lo que significa que el clúster cifra los datos antes de enviarlos a través de la red. No puede utilizar una clave KMS para cifrar los datos en tránsito. Para obtener más información, consulte [Cifrado de datos en tránsito](#) en la Guía de administración de Amazon EMR.

Para obtener más información sobre todas las opciones de cifrado disponibles en Amazon EMR, consulte [Opciones de cifrado](#) en la Guía de administración de Amazon EMR.

Temas

- [Cifrar datos en el sistema de archivos EMR \(EMRFS\)](#)
- [Cifrar datos en los volúmenes de almacenamiento de los nodos de clúster](#)
- [Contexto de cifrado](#)

Cifrar datos en el sistema de archivos EMR (EMRFS)

Los clústeres de Amazon EMR usan dos sistemas de archivos distribuidos:

- Hadoop Distributed File System (HDFS). El cifrado HDFS no utiliza una clave KMS en AWS KMS.
- Sistema de archivos EMR (EMRFS). EMRFS es una implementación de HDFS que permite a los clústeres de Amazon EMR almacenar datos en Amazon Simple Storage Service (Amazon S3). EMRFS admite cuatro opciones de cifrado, dos de las cuales utilizan una clave KMS en AWS KMS. Para obtener más información acerca de las cuatro opciones de cifrado de EMRFS, consulte [Opciones de cifrado](#) en la Guía de administración de Amazon EMR.

Las dos opciones de cifrado de EMRFS que utilizan una clave KMS usan las siguientes características de cifrado que ofrece Amazon S3:

- [Proteja los datos mediante el cifrado del lado del servidor con AWS Key Management Service \(SSE-KMS\)](#). El clúster de Amazon EMR envía datos a Amazon S3. Amazon S3 utiliza una clave KMS para cifrar los datos antes de guardarlos en un bucket de S3. Para obtener más información sobre cómo funciona, consulte [Proceso de cifrar datos en EMRFS con SSE-KMS](#).
- [Protección de datos mediante el cifrado del lado del cliente \(CSE-KMS\)](#). Los datos de Amazon EMR de Amazon se cifran bajo un AWS KMS key antes de su envío a Amazon S3 para el almacenamiento. Para obtener más información sobre cómo funciona, consulte [Proceso de cifrar datos en EMRFS con CSE-KMS](#).

Al configurar un clúster de Amazon EMR para cifrar datos en EMRFS con una clave KMS, hay que elegir la clave KMS que se desea que utilice Amazon S3 o el clúster de Amazon EMR. Con SSE-KMS, puede elegir la Clave administrada de AWS para Amazon S3 con el alias `aws/s3`, o una clave simétrica administrada por el cliente que haya creado. Con el cifrado del lado del cliente, debe elegir una clave simétrica administrada por el cliente que cree. Si elige una clave administrada por el cliente, debe asegurarse de que el clúster de Amazon EMR tiene permiso para utilizar la clave KMS. Para obtener más información, consulte [Uso AWS KMS keys para el cifrado](#) en la Guía de administración de Amazon EMR.

Tanto para el cifrado del lado del servidor como del lado del cliente, la clave KMS que elija es la clave raíz en un flujo de trabajo de [cifrado de sobre](#). Los datos se cifran con una [clave de datos](#) única que se cifra con la clave KMS in AWS KMS. Los datos cifrados y una copia cifrada de su clave de datos se almacenan juntos como un solo objeto cifrado en un bucket de S3. Para obtener más información sobre cómo funciona, consulte los siguientes temas.

Temas

- [Proceso de cifrar datos en EMRFS con SSE-KMS](#)
- [Proceso de cifrar datos en EMRFS con CSE-KMS](#)

Proceso de cifrar datos en EMRFS con SSE-KMS

Al configurar un clúster de Amazon EMR para usar SSE-KMS, el proceso de cifrado funciona del siguiente modo:

1. El clúster envía datos a Amazon S3 para almacenarlos en un bucket de S3.
2. Amazon S3 envía una [GenerateDataKey](#) solicitud a AWS KMS, especificando el ID de clave de KMS que eligió al configurar el clúster para usar SSE-KMS. La solicitud incluye el contexto de cifrado; para obtener más información, consulte [Contexto de cifrado](#).
3. AWS KMS genera una clave de cifrado de datos única (clave de datos) y, a continuación, envía dos copias de esta clave de datos a Amazon S3. Una copia está sin cifrar (texto no cifrado) y la otra se cifra con la clave KMS.
4. Amazon S3 utiliza la clave de datos de texto no cifrado para cifrar los datos que ha recibido en el paso 1 y, a continuación, elimina la clave de datos de texto no cifrado de la memoria tan pronto como sea posible después de utilizarla.
5. Amazon S3 almacena los datos cifrados y la copia cifrada de la clave de datos como un solo objeto cifrado en un bucket de S3.

El proceso de descifrado funciona de la siguiente manera:

1. El clúster solicita un objeto de datos cifrado de un bucket de S3.
2. Amazon S3 extrae la clave de datos cifrados del objeto S3 y, a continuación, envía la clave de datos cifrados AWS KMS con una solicitud de [descifrado](#). La solicitud incluye un [contexto de cifrado](#).
3. AWS KMS descifra la clave de datos cifrada con la misma clave de KMS que se utilizó para cifrarla y, a continuación, envía la clave de datos descifrada (texto sin formato) a Amazon S3.

4. Amazon S3 utiliza la clave de datos de texto no cifrado para descifrar los datos cifrados y, a continuación, elimina la clave de datos de texto no cifrado de la memoria tan pronto como sea posible después de utilizarla.
5. Amazon S3 envía los datos descifrados al clúster.

Proceso de cifrar datos en EMRFS con CSE-KMS

Al configurar un clúster de Amazon EMR para usar CSE-KMS, el proceso de cifrado funciona del siguiente modo:

1. Cuando esté listo para almacenar datos en Amazon S3, el clúster envía una [GenerateDataKey](#) solicitud a AWS KMS la que especifica el ID de clave de la clave de KMS que eligió al configurar el clúster para usar CSE-KMS. La solicitud incluye el contexto de cifrado; para obtener más información, consulte [Contexto de cifrado](#).
2. AWS KMS genera una clave de cifrado de datos única (clave de datos) y, a continuación, envía dos copias de esta clave de datos al clúster. Una copia está sin cifrar (texto no cifrado) y la otra copia se cifra con la clave KMS.
3. El clúster utiliza la clave de datos de texto no cifrado para cifrar los datos y, a continuación, elimina la clave de datos de texto no cifrado de la memoria tan pronto como sea posible después de utilizarla.
4. El clúster combina los datos cifrados y la copia cifrada de la clave de datos en un solo objeto cifrado.
5. El clúster envía el objeto cifrado a Amazon S3 para almacenarlo.

El proceso de descifrado funciona de la siguiente manera:

1. El clúster solicita el objeto de datos cifrado de un bucket de S3.
2. Amazon S3 envía el objeto cifrado al clúster.
3. El clúster extrae la clave de datos cifrados del objeto cifrado y, a continuación, envía la clave de datos cifrados AWS KMS con una solicitud de [descifrado](#). La solicitud incluye el [contexto de cifrado](#).
4. AWS KMS descifra la clave de datos cifrada con la misma clave de KMS que se utilizó para cifrarla y, a continuación, envía la clave de datos descifrada (texto sin formato) al clúster.

5. El clúster utiliza la clave de datos de texto no cifrado para descifrar los datos cifrados y, a continuación, elimina la clave de datos de texto no cifrado de la memoria tan pronto como sea posible después de utilizarla.

Cifrar datos en los volúmenes de almacenamiento de los nodos de clúster

Un clúster de Amazon EMR es una colección de instancias de Amazon Elastic Compute Cloud (Amazon EC2). Cada instancia del clúster se denomina un nodo de clúster o nodo. Cada nodo puede tener dos tipos de volúmenes de almacenamiento: volúmenes de almacén de instancias y volúmenes de Amazon Elastic Block Store (Amazon EBS). Puede configurar el clúster para utilizar la [Configuración de clave unificada de Linux \(LUKS\)](#) para cifrar ambos tipos de volúmenes de almacenamiento en los nodos (pero no el volumen de arranque de cada nodo). Se denomina cifrado de disco local.

Al habilitar el cifrado de disco local para un clúster, puede cifrar la clave LUKS con una clave KMS en AWS KMS. Debe elegir una [clave administrada por el cliente](#) que cree; no puede utilizar una [Clave administrada de AWS](#). Si elige una clave administrada por el cliente, debe asegurarse de que el clúster de Amazon EMR tiene permiso para utilizar la clave KMS. Para obtener más información, consulte [Uso AWS KMS keys para el cifrado](#) en la Guía de administración de Amazon EMR.

Si habilita el cifrado del disco local mediante una clave KMS, el proceso de cifrado funciona del siguiente modo:

1. Cuando se lanza cada nodo del clúster, envía una [GenerateDataKey](#) solicitud a AWS KMS la que especifica el ID de clave de la clave de KMS que eligió al habilitar el cifrado del disco local para el clúster.
2. AWS KMS genera una clave de cifrado de datos única (clave de datos) y, a continuación, envía dos copias de esta clave de datos al nodo. Una copia está sin cifrar (texto no cifrado) y la otra se cifra con la clave KMS.
3. El nodo utiliza una versión de la codificación de base64 de la clave de datos de texto no cifrado como la contraseña que protege la clave LUKS. El nodo guarda la copia cifrada de la clave de datos en su volumen de arranque.
4. [Si el nodo se reinicia, el nodo reiniciado envía la clave de datos cifrados AWS KMS con una solicitud de descifrado.](#)
5. AWS KMS descifra la clave de datos cifrada con la misma clave KMS que se utilizó para cifrarla y, a continuación, envía la clave de datos descifrada (texto sin formato) al nodo.

6. El nodo utiliza la versión de la codificación de base64 de la clave de datos de texto no cifrado como la contraseña para desbloquear la clave LUKS.

Contexto de cifrado

Cada AWS servicio integrado AWS KMS puede especificar un [contexto de cifrado](#) cuando el servicio lo utiliza AWS KMS para generar claves de datos o para cifrar o descifrar datos. El contexto de cifrado es información autenticada adicional que se AWS KMS utiliza para comprobar la integridad de los datos. Cuando un servicio especifica el contexto de cifrado para una operación de cifrado, debe especificar el mismo contexto de cifrado para la operación de descifrado correspondiente; de lo contrario, el descifrado no se realizará correctamente. El contexto de cifrado también se escribe en los archivos de AWS CloudTrail registro, lo que puede ayudarle a entender por qué se utilizó una clave KMS específica.

En la siguiente sección se explica el contexto de cifrado que se utiliza en cada situación de cifrado de Amazon EMR que utiliza una clave KMS.

Contexto de cifrado para el cifrado de EMRFS con SSE-KMS

Con SSE-KMS, el clúster de Amazon EMR envía datos a Amazon S3 y, a continuación, Amazon S3 utiliza una clave KMS para cifrar los datos antes de guardarlos en un bucket de S3. En este caso, Amazon S3 utiliza el nombre de recurso de Amazon (ARN) del objeto S3 como contexto de cifrado para cada [GenerateDataKey](#) solicitud de [Decrypt a](#) la que envía. AWS KMS El siguiente ejemplo muestra una representación JSON del contexto de cifrado que usa Amazon S3.

```
{ "aws:s3:arn" : "arn:aws:s3:::S3_bucket_name/S3_object_key" }
```

Contexto de cifrado para el cifrado de EMRFS con CSE-KMS

Con CSE-KMS, el clúster de Amazon EMR utiliza una clave KMS para cifrar los datos antes de enviarlos a Amazon S3 para almacenarlos. En este caso, el clúster utiliza el nombre de recurso de Amazon (ARN) de la clave de KMS como contexto de cifrado con cada [GenerateDataKey](#) solicitud de [Decrypt a](#) la que envía. AWS KMS El siguiente ejemplo muestra una representación JSON del contexto de cifrado que usa el clúster.

```
{ "kms_cmek_id" : "arn:aws:kms:us-east-2:111122223333:key/0987ab65-43cd-21ef-09ab-87654321cdef" }
```

Contexto de cifrado para el cifrado de disco local con LUKS

Cuando un clúster de Amazon EMR utiliza el cifrado de disco local con LUKS, los nodos del clúster no especifican el contexto de cifrado con las solicitudes [GenerateDataKey](#) y [Decrypt](#) a las que envían. AWS KMS

Cómo utiliza Amazon Redshift AWS KMS

En este tema se describe cómo Amazon Redshift cifra AWS KMS los datos.

Temas

- [Cifrado de Amazon Redshift](#)
- [Contexto de cifrado](#)

Cifrado de Amazon Redshift

Un almacén de datos de Amazon Redshift es una colección de recursos de computación denominados nodos que están organizados en un grupo llamado clúster. Cada clúster ejecuta un motor Amazon Redshift y contiene una o más bases de datos.

Amazon Redshift usa una arquitectura de cuatro niveles basada en claves para el cifrado. La arquitectura consta de claves de cifrado de datos, una clave de base de datos, una clave de clúster y una clave maestra. Puede utilizar una AWS KMS key como clave raíz.

Las claves de cifrado de datos cifran los bloques de datos del clúster. A cada bloque de datos se le asigna una clave AES-256 generada aleatoriamente. Estas claves se cifran mediante la clave de base de datos del clúster.

La clave de base de datos cifra las claves de cifrado de datos del clúster. La clave de base de datos es una clave AES-256 generada aleatoriamente. Se almacena en disco en una red diferente del clúster de Amazon Redshift y se pasa al clúster a través de un canal seguro.

La clave del clúster cifra la clave de base de datos del clúster de Amazon Redshift. Puede usar AWS KMS, AWS CloudHSM, o un módulo de seguridad de hardware (HSM) externo para administrar la clave del clúster. Consulte la documentación sobre [cifrado de base de datos de Amazon Redshift](#) para obtener más información.

Puede solicitar el cifrado marcando la casilla correspondiente en la consola de Amazon Redshift. Puede especificar una [clave administrada por el cliente](#) para usarla seleccionando una en la lista que

aparece debajo del cuadro de cifrado. Si no especifica una clave administrada por el cliente, Amazon Redshift utiliza el [Clave administrada de AWS](#) para Amazon Redshift en tu cuenta.

 Important

Amazon Redshift solo admite claves KMS de cifrado simétricas. No se puede utilizar una clave KMS asimétrica en un flujo de trabajo de Amazon Redshift cifrado. Para obtener ayuda para determinar si una clave KMS es simétrica o asimétrica, consulte [Identificación de diferentes tipos de claves](#).

Contexto de cifrado

Cada servicio integrado AWS KMS especifica un [contexto de cifrado](#) al solicitar claves de datos, cifrar y descifrar. El contexto de cifrado son datos autenticados adicionales (AAD) que se AWS KMS utilizan para comprobar la integridad de los datos. Es decir, cuando se especifica un contexto de cifrado para una operación de cifrado, el servicio también lo especifica en la operación de descifrado; de lo contrario, el descifrado no se realizará correctamente. Amazon Redshift utiliza el ID de clúster y la hora de creación para el contexto de cifrado. En el `requestParameters` campo de un archivo de CloudTrail registro, el contexto de cifrado tendrá un aspecto similar al siguiente.

```
"encryptionContext": {  
  "aws:redshift:arn": "arn:aws:redshift:region:account_ID:cluster:cluster_name",  
  "aws:redshift:createtime": "20150206T1832Z"  
},
```

Puede buscar el nombre del clúster en sus CloudTrail registros para saber qué operaciones se realizaron mediante una AWS KMS key (clave KMS). Las operaciones incluyen el cifrado del clúster, el descifrado del clúster y la generación de claves de datos.

AWS KMS Referencia

El siguiente material de referencia proporciona información útil sobre el uso y la administración de claves KMS.

- [Referencia de tipos de clave](#). Muestra el tipo de clave KMS que admite cada operación de la API de AWS KMS .

Para buscar: ¿Puedo habilitar y desactivar una clave KMS de firma RSA?

- [Tabla estado de claves](#). Muestra cómo afecta el estado de la clave de una clave KMS a su uso en operaciones de la API de AWS KMS .

Para buscar: ¿Puedo cambiar el alias de una clave KMS que está pendiente de eliminación?

- [Referencia de permisos de la API de AWS KMS](#). Proporciona información sobre los permisos necesarios para cada operación de la AWS KMS API.

Para averiguarlo: ¿Puedo ejecutar [GetKeyPolicy](#) una clave de otra AWS cuenta? ¿Puedo conceder el permiso `kms:Decrypt` en una política de IAM?

- [ViaService referencia](#). Enumera los servicios de AWS que admiten la clave de condición `kms:ViaService`.

Para buscar: ¿Puedo usar la clave de `kms:ViaService` condición para permitir un permiso solo cuando proviene de Amazon ElastiCache? ¿Qué ocurre con Amazon Neptune?

- [AWS KMS precios](#). Enumera y explica el precio de las claves KMS.

Para buscar: ¿Cuánto cuesta el uso de mis claves asimétricas?

- [AWS KMS solicitar cuotas](#). Muestra las cuotas por segundo para las solicitudes de AWS KMS API en cada cuenta y región.

Para buscar: ¿Cuántas solicitudes [Decrypt](#) puedo ejecutar en cada segundo? ¿Cuántas solicitudes [Decrypt](#) puedo ejecutar en claves KMS en mi almacén de claves personalizado?

- [AWS KMS cuotas de recursos](#). Enumera las cuotas de recursos de AWS KMS .

Para encontrar: ¿Cuántas claves KMS puedo tener en cada región de mi cuenta? ¿Cuántos alias puedo tener en cada clave KMS?

- [AWS servicios integrados con AWS KMS](#). Enumera los AWS servicios que utilizan claves de KMS para proteger los recursos que crean, almacenan y administran.

Para averiguar: ¿Connect Customer utiliza claves KMS para proteger mis recursos de Connect?

Estados clave de AWS KMS claves

A AWS KMS key siempre tiene un estado clave. Las operaciones en la clave de KMS y su entorno pueden cambiar el estado de la clave. El estado de la clave puede cambiar de forma transitoria, o hasta que otra operación cambie su estado de clave. Estas operaciones se realizan de forma asíncrona o mediante una llamada a la API.

La tabla de esta sección muestra cómo los estados clave afectan a las llamadas a las operaciones de la AWS KMS API. Como resultado de su estado clave, se espera que una operación en una clave KMS tenga éxito (#), falle (X), o tenga éxito solo bajo ciertas condiciones (?). El resultado a menudo difiere en el caso de las claves KMS con material de claves importado.

Esta tabla incluye sólo las operaciones de API que utilizan una clave KMS existente. Se omiten otras operaciones [ListKeys](#), como [CreateKey](#).

Temas

- [Estados clave y tipos de claves KMS](#)
- [Tabla estado de claves](#)

Estados clave y tipos de claves KMS

El tipo de clave KMS determina los estados clave que puede tener.

- Todas las claves KMS pueden estar en los estados Enabled, Disabled y PendingDeletion.
- La mayoría de las claves KMS se crean en el estado Enabled. Las claves con material de claves importado se crean en el estado PendingImport.
- El estado PendingImport solo se aplica a las claves KMS con [material de claves importado](#). Cuando el material de claves de una clave importada se elimina o caduca, el estado cambia de Enabled a PendingImport.
- El estado Unavailable se aplica solo a una clave KMS en un [almacén de claves personalizado](#). Una clave KMS en un [almacén de AWS CloudHSM claves](#) se produce Unavailable cuando el almacén de claves personalizado se desconecta intencionadamente de su AWS CloudHSM clúster. Una clave de KMS en un [almacén de claves externo](#) está Unavailable cuando el almacén de claves personalizado se desconecta de forma intencionada de su [proxy del almacén](#)

[de claves externo](#). Puede ver y administrar las claves KMS no disponibles, pero no puede usarlas en operaciones criptográficas.

El estado de clave de una clave de KMS de un almacén de claves personalizado no se ve afectado por los cambios realizados a su clave de respaldo. Una clave KMS de un almacén de AWS CloudHSM claves no se ve afectada por los cambios en su [material de claves asociado](#) en el AWS CloudHSM clúster. Una clave de KMS de un almacén de claves externo no se ve afectada por los cambios en su [clave externa](#) en un administrador de claves externo. Si la clave de respaldo está deshabilitada o eliminada, el estado de la clave de KMS no cambia, pero fallan las operaciones criptográficas que utilizan la clave de KMS.

- Los estados clave `Creating`, `Updating` y `PendingReplicaDeletion` solo se aplican a [claves de varias regiones](#).
 - Una clave de réplica de varias regiones está en el estado de clave `Creating` transitorio mientras se está creando. Es posible que este proceso siga en curso cuando se complete la [ReplicateKey](#) operación. Cuando se completa el proceso de replicación, la clave de réplica se encuentra en el estado `Enabled` o `PendingImport`.
 - Multi-Region las claves están en el estado de `Updating` clave transitoria mientras se actualiza la región principal. Es posible que este proceso siga en curso cuando se complete la [UpdatePrimaryRegion](#) operación. Cuando se completa el proceso de actualización, las claves principal y de réplica reanudan el estado de clave `Enabled`.
 - Cuando se programa la eliminación de una clave principal de varias regiones que tiene claves de réplica, la clave principal se encuentra en el estado `PendingReplicaDeletion` hasta que se eliminen todas sus claves de réplica. A continuación, el estado de clave cambia a `PendingDeletion`. Para obtener más información, consulte [Deleting multi-Region keys](#).

Tabla estado de claves

En la siguiente tabla se muestra cómo el estado de clave de una clave KMS afecta las operaciones de AWS KMS .

Las descripciones de las notas numeradas a pie de página ([n]) se encuentran al final de este tema.

Note

Es posible que tenga que desplazarse horizontal o verticalmente para ver todos los datos de esta tabla.

API	Habilitad o	Deshabili tado	Eliminaci ón pendiente Eliminaci ón pendiente de réplica	Importaci ón pendiente	No disponibl e	Creación	Actualiza ción
CancelKey Deletion	 [4]	 [4]		 [4]	 [4], [13]	 [4]	 [4]
CreateAli as			 [3]				
CreateGra nt		 [1]	 [2] o [3]	 [5]		 [14]	
Decrypt		 [1]	 [2] o [3]	 [5]	 [11]	 [14]	
DeleteAli as							
DeleteImp ortedKeyM aterial	 [9]	 [9]	 [9]		N/A	 [14]	 [15]
DeriveSha redSecret					N/A		

API	Habilitad o	Deshabili tado	Eliminaci ón pendiente Eliminaci ón pendiente de réplica	Importaci ón pendiente	No disponibl e	Creación	Actualiza ción
		[1]	[2] o [3]	[5]		[14]	
DescribeKey	✓	✓	✓	✓	✓	✓	✓
DisableKey	✓	✓	✗ [3]	✗ [5]	✓ [12]	✗ [14]	✗ [15]
DisableKeyRotation	❓ [7]	✗ [1] o [7]	✗ [3] o [7]	✗ [6]	✗ [7]	✗ [14]	❓ [7]
EnableKey	✓	✓	✗ [3]	✗ [5]	✓ [12]	✗ [14]	✗ [15]
EnableKeyRotation	❓ [7]	✗ [1] o [7]	✗ [3] o [7]	✗ [6]	✗ [7]	✗ [14]	❓ [7]
Encrypt	✓	✗ [1]	✗ [2] o [3]	✗ [5]	✗ [11]	✗ [14]	✓

API	Habilitad o	Deshabili tado	Eliminaci ón pendiente Eliminaci ón pendiente de réplica	Importaci ón pendiente	No disponibl e	Creación	Actualiza ción
GenerateDataKey	✓	 [1]	 [2] o [3]	 [5]	 [11]	 [14]	✓
GenerateDataKeyPair	✓	 [1]	 [2] o [3]	 [5]	 [7]	 [14]	✓
GenerateDataKeyPairWithoutPlaintext	✓	 [1]	 [2] o [3]	 [5]	 [7]	 [14]	✓
GenerateDataKeyWithoutPlaintext	✓	 [1]	 [2] o [3]	 [5]	 [11]	 [14]	✓
GenerateMac	✓	 [1]	 [2] o [3]	 [5]	N/A	 [14]	✓
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓

API	Habilitad o	Deshabili tado	Eliminaci ón pendiente Eliminaci ón pendiente de réplica	Importaci ón pendiente	No disponibl e	Creación	Actualiza ción
GetKeyRot ationStat us	 [7]	 [7]	 [7]	 [6]	 [7]	 [7]	 [7]
GetParam etersForIm port	 [9]	 [9]	 [8] o [9]		 [9]	 [14]	 [15]
GetPublic Key			 [2] o [3]		N/A	 [14]	
ImportKey Material	 [9]	 [9]	 [9]		 [9]	 [14]	
ListAlias es							
ListGrant s							
ListKeyPo licies							

API	Habilitad o	Deshabili tado	Eliminaci ón pendiente Eliminaci ón pendiente de réplica	Importaci ón pendiente	No disponibl e	Creación	Actualiza ción
ListKeyRo tations	 [7]	 [7]	 [7]		 [7]	 [7]	 [7]
ListResou rceTags							
PutKeyPol icy							
ReEncrypt		 [1]	 [2] o [3]	 [5]	 [11]	 [14]	
Replicate Key		 [1]	 [2] o [3]	 [5]	N/A	 [14]	 [15]
RetireGra nt							
RevokeGra nt							
RotateKey OnDemand	 [7]	 [1] o [7]	 [3] o [7]	 [5]	 [7]	 [14]	 [7]

API	Habilitad o	Deshabili tado	Eliminaci ón pendiente Eliminaci ón pendiente de réplica	Importaci ón pendiente	No disponibl e	Creación	Actualiza ción
ScheduleKeyDeletion	✓	✓	✗ [3]	✓	✓	✓	✗ [15]
Sign	✓	✗ [1]	✗ [2] o [3]	✗ [5]	N/A	✗ [14]	✓
TagResource	✓	✓	✗ [3]	✓	✓	✓	✓
UntagResource	✓	✓	✗ [3]	✓	✓	✓	✓
UpdateAliases	✓	✓	⓪ [10]	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	✗ [3]	✓	✓	✓	✓

API	Habilitad o	Deshabili tado	Eliminaci ón pendiente Eliminaci ón pendiente de réplica	Importaci ón pendiente	No disponibl e	Creación	Actualiza ción
UpdatePri maryRegio n		 [1]	 [2] o [3]	 [5]	N/A	 [14]	
Verificar		 [1]	 [2] o [3]	 [5]	N/A	 [14]	
VerifyMac		 [1]	 [2] o [3]	 [5]	N/A	 [14]	

Detalles de la tabla

- [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] Si la clave de KMS está en un almacén de claves personalizado: `UnsupportedOperationException`.
- [8] Si la clave KMS tiene material de claves importado: `KMSInvalidStateException`
- [9] Si la clave de KMS no puede tener o no tiene material de claves importado: `UnsupportedOperationException`.
- [10] Si la clave KMS de origen está pendiente de eliminación, el comando se ejecuta satisfactoriamente. Si la clave KMS de destino está pendiente de eliminación, el comando genera el error: `KMSInvalidStateException` : `<key ARN>` is pending deletion.
- [11] `KMSInvalidStateException`: `<key ARN>` is unavailable. No puede realizar esta operación en una clave KMS no disponible.
- [12] La operación se ha realizado correctamente pero el estado de clave de la clave KMS no cambiará hasta que esté disponible.
- [13] Mientras una clave KMS en el almacén de claves personalizado esté pendiente de eliminación, su estado de clave seguirá siendo `PendingDeletion` incluso si la clave KMS no está disponible. Esto permite cancelar la eliminación de la clave KMS en cualquier momento durante el período de espera.
- [14] `KMSInvalidStateException`: `<key ARN>` is creating. AWS KMS lanza esta excepción mientras se replica una clave multirregional (`()ReplicateKey`).
- [15] `KMSInvalidStateException`: `<key ARN>` is updating. AWS KMS lanza esta excepción mientras se actualiza la región principal de una clave multirregional (`()UpdatePrimaryRegion`).

Referencia de tipos de claves

AWS KMS admite diferentes funciones para diferentes tipos de claves KMS. Por ejemplo, solo puede utilizar las [claves de KMS de cifrado simétricas](#) para [generar claves de datos simétricas](#) y [pares de claves de datos asimétricas](#). Además, la [importación del material de claves](#) y la [rotación automática de claves](#) son compatibles únicamente con las claves de KMS de cifrado simétricas. Asimismo, solo puede crear claves de KMS de cifrado simétricas en un [almacén de claves personalizadas](#).

Esta referencia incluye dos tablas.

- La [tabla de tipos de clave](#) muestra las AWS KMS operaciones que son válidas para las claves KMS de cifrado simétrico, las claves KMS asimétricas y las claves KMS HMAC.
- La [tabla de características especiales](#) muestra las operaciones de AWS KMS que son válidas para las claves de KMS para varias regiones, las claves de KMS con material de claves importado y las claves de KMS de almacenes de claves personalizados.

Tabla de tipos de claves

Es posible que tenga que desplazarse horizontal o verticalmente para ver todos los datos de esta tabla.

AWS KMS Funcionamiento de la API	Claves de KMS de cifrado simétrico	Claves KMS HMAC	Claves de KMS asimétricas (ENCRYPT DECRYPT)	Claves de KMS asimétricas (SIGN_VERIFY)	Claves KMS asimétricas (KEY_AGREEMENT)
CancelKeyDeletion	Sí	Sí	Sí	Sí	Sí
CreateAlias	Sí	Sí	Sí	Sí	Sí
CreateGrant	Sí	Sí	Sí	Sí	Sí
CreateKey	Sí	Sí	Sí	Sí	Sí
Decrypt	Sí	No	Sí	No	No
DeleteAlias	Sí	Sí	Sí	Sí	Sí
DeleteImportedKeyMaterial	Sí	Sí	Sí	Sí	Sí
Válido solo en las claves de KMS con material de claves importado (Origin es EXTERNAL).					

AWS KMS Funcionamiento de la API	Claves de KMS de cifrado simétrico	Claves KMS HMAC	Claves de KMS asimétricas (ENCRYPT DECRYPT)	Claves de KMS asimétricas (SIGN_VERIFY)	Claves KMS asimétricas (KEY_AGREEMENT)
DeriveSharedSecret	No	No	No	No	Sí
DescribeKey	Sí	Sí	Sí	Sí	Sí
DisableKey	Sí	Sí	Sí	Sí	Sí
DisableKeyRotation	Sí Válido solo en claves de KMS con material (Origine AWS KMS clave.	No	No	No	No
EnableKey	Sí	Sí	Sí	Sí	Sí

AWS KMS Funcionamiento de la API	Claves de KMS de cifrado simétrico	Claves KMS HMAC	Claves de KMS asimétricas (ENCRYPT DECRYPT)	Claves de KMS asimétricas (SIGN_VERIFY)	Claves KMS asimétricas (KEY_AGREEMENT)
EnableKeyRotation	Sí	No	No	No	No
	Válido solo en claves KMS con material (Orígene AWS KMS clave.				
Encrypt	Sí	No	Sí	No	No
GenerateDataKey	Sí	No	No	No	No
GenerateDataKeyPair	Sí	No	No	No	No
Genera un par de claves de datos asimétricos protegido por una clave de KMS de cifrado simétrica.	No es válido en claves de KMS en almacene de claves personalizados.				

AWS KMS Funcionamiento de la API	Claves de KMS de cifrado simétrico	Claves KMS HMAC	Claves de KMS asimétricas (ENCRYPT DECRYPT)	Claves de KMS asimétricas (SIGN_VERIFY)	Claves KMS asimétricas (KEY_AGREEMENT)
GenerateDataKeyPairWithoutPlaintext Genera un par de claves de datos asimétricos protegido por una clave de KMS de cifrado simétrica.	Sí	No	No	No	No
GenerateDataKeyWithoutPlaintext	No es válido en claves de KMS en almacene de claves personalizados.				
GenerateMac	No	Sí	No	No	No
GetKeyPolicy	Sí	Sí	Sí	Sí	Sí
GetKeyRotationStatus	Sí	Sí (KeyRotationEnabled siempre será false.)	Sí (KeyRotationEnabled siempre será false.)	Sí (KeyRotationEnabled siempre será false.)	Sí (KeyRotationEnabled siempre será false.)

AWS KMS Funcionamiento de la API	Claves de KMS de cifrado simétrico	Claves KMS HMAC	Claves de KMS asimétricas (ENCRYPT DECRYPT)	Claves de KMS asimétricas (SIGN_VERIFY)	Claves KMS asimétricas (KEY_AGREEMENT)
GetParametersForImport Válido solo en las claves de KMS con material de claves importado (Origin es EXTERNAL).	Sí	Sí	Sí	Sí	Sí
GetPublicKey	No	No	Sí	Sí	Sí
ImportKeyMaterial Válido solo en las claves de KMS con material de claves importado (Origin es EXTERNAL).	Sí	Sí	Sí	Sí	Sí
ListAliases	Sí	Sí	Sí	Sí	Sí
ListGrants	Sí	Sí	Sí	Sí	Sí
ListKeyPolicies	Sí	Sí	Sí	Sí	Sí

AWS KMS Funcionamiento de la API	Claves de KMS de cifrado simétrico	Claves KMS HMAC	Claves de KMS asimétricas (ENCRYPT DECRYPT)	Claves de KMS asimétricas (SIGN_VERIFY)	Claves KMS asimétricas (KEY_AGREEMENT)
ListKeyRotations	Sí	Sí (El campo Rotations siempre será nulo o estará vacío).	Sí (El campo Rotations siempre será nulo o estará vacío).	Sí (El campo Rotations siempre será nulo o estará vacío).	Sí (El campo Rotations siempre será nulo o estará vacío).
ListResourceTags	Sí	Sí	Sí	Sí	Sí
ListRetirableGrants	Sí	Sí	Sí	Sí	Sí
PutKeyPolicy	Sí	Sí	Sí	Sí	Sí
ReEncrypt	Sí	No	Sí	No	No
ReplicateKey - Válido solo en claves de varias regiones	Sí	Sí	Sí	Sí	Sí
RetireGrant	Sí	Sí	Sí	Sí	Sí
RevokeGrant	Sí	Sí	Sí	Sí	Sí

AWS KMS Funcionamiento de la API	Claves de KMS de cifrado simétrico	Claves KMS HMAC	Claves de KMS asimétricas (ENCRYPT DECRYPT)	Claves de KMS asimétricas (SIGN_VERIFY)	Claves KMS asimétricas (KEY_AGREEMENT)
RotateKeyOnDemand	Sí Válido solo en claves KMS de cifrado simétrico administradas por el cliente con origen AWS_KMS o EXTERNAL	No	No	No	No
ScheduleKeyDeletion	Sí	Sí	Sí	Sí	Sí
Sign	No	No	No	Sí	No
TagResource	Sí	Sí	Sí	Sí	Sí
UntagResource	Sí	Sí	Sí	Sí	Sí

AWS KMS Funcionamiento de la API	Claves de KMS de cifrado simétrico	Claves KMS HMAC	Claves de KMS asimétricas (ENCRYPT DECRYPT)	Claves de KMS asimétricas (SIGN_VERIFY)	Claves KMS asimétricas (KEY_AGREEMENT)
UpdateAlias	Sí	Sí	Sí	Sí	Sí
La clave de KMS actual y la nueva clave de KMS deben ser del mismo tipo (ambas simétricas o ambas asimétricas o ambas HMAC) y deben tener el mismo uso de clave .					
UpdateKeyDescription	Sí	Sí	Sí	Sí	Sí
UpdateReplicaRegion	Sí	Sí	Sí	Sí	Sí
- Válido solo en claves de varias regiones					
Verificar	No	No	No	Sí	No
VerifyMac	No	Sí	No	No	No

Tabla de características especiales

En esta tabla se muestran las operaciones de AWS KMS API que se admiten en cada tipo de clave de uso especial.

Al leer esta tabla, debe tener en cuenta las siguientes interacciones:

- [Claves de varias regiones](#):
 - Las claves de varias regiones pueden ser claves de KMS de cifrado simétrico, claves de KMS asimétricas, claves de KMS HMAC y claves de KMS con material de claves importado.
 - No puede crear claves de varias regiones en un almacén de claves personalizado.
- [Material de claves importado](#)
 - Puede importar material de claves de KMS de cifrado simétrico, claves de KMS asimétricas y claves HMAC de KMS.
 - Puede crear [claves de varias regiones con material de claves importado](#).
 - No puede crear claves con material de claves importado en un almacén de claves personalizado.
 - La rotación automática de claves (`EnableKeyRotation`, `DisableKeyRotation`) no es compatible con las claves KMS con el material de claves importado.
 - La rotación de claves bajo demanda (`RotateKeyOnDemand`) se admite para las claves KMS de cifrado simétrico con material de clave importado.
- [Almacenes de claves personalizados](#)
 - Los almacenes de claves personalizados solo admiten claves KMS de cifrado simétricas.
 - Las operaciones simétricas en pares de claves asimétricas (`GenerateDataKeyPair`, `GenerateDataKeyPairWithoutPlaintext`) no se admiten en las claves de KMS de los almacenes de claves personalizados.
 - Las claves KMS de almacenes de claves personalizados no admiten la rotación automática de claves (`EnableKeyRotation`, `DisableKeyRotation`).
 - No puede crear claves de varias regiones en almacenes de claves personalizados.

Es posible que tenga que desplazarse horizontal o verticalmente para ver todos los datos de esta tabla.

AWS KMS Funcionamiento de la API	Claves de varias regiones	Material de claves importado	Las claves KMS en un almacén de claves personalizado
CancelKeyDeletion	✓	✓	✓
CreateAlias	✓	✓	✓
CreateGrant	✓	✓	✓
CreateKey Puede utilizar <code>CreateKey</code> para crear una clave principal de varias regiones, una clave de KMS con material de claves importado o una clave de KMS en un almacén de claves personalizado. Para crear una clave de réplica de varias regiones, utilice <code>ReplicateKey</code> .	✓	✓	✓
Decrypt	✓ Válido solo cuando <code>KeyUsage</code> es <code>ENCRYPT_D</code> <code>ECRYPT</code>	✓	✓
DeleteAlias	✓	✓	✓
DeleteImportedKeyMaterial	✓	✓	✗

AWS KMS Funcionamiento de la API	Claves de varias regiones	Material de claves importado	Las claves KMS en un almacén de claves personalizado
	Válido solo para claves con material de claves importado (Origin es EXTERNAL)		
DeriveSharedSecret	 Válido solo cuando KeyUsage es KEY_AGREEMENT)	 Válido solo cuando KeyUsage es KEY_AGREEMENT)	
DescribeKey			
DisableKey			
DisableKeyRotation	 Válido solo en claves de cifrado simétricas con material (OriginesAWS_KM AWS KMS clave.		

AWS KMS Funcionamiento de la API	Claves de varias regiones	Material de claves importado	Las claves KMS en un almacén de claves personalizado
EnableKey	 Válido solo con claves de cifrado de KMS simétricas		
EnableKeyRotation	 Válido solo en claves de cifrado simétricas con material (OriginesAWS_KM de AWS KMS clave.		
Encrypt	 Válido solo cuando KeyUsage es ENCRYPT_DECRYPT		

AWS KMS Funcionamiento de la API	Claves de varias regiones	Material de claves importado	Las claves KMS en un almacén de claves personalizado
GenerateDataKey	✓ Válido solo con claves de cifrado de KMS simétricas	✓	✓
GenerateDataKeyPair	✓ Válido solo con claves de cifrado de KMS simétricas	✓	✗
GenerateDataKeyPairWithoutPlaintext	✓ Válido solo con claves de cifrado de KMS simétricas	✓	✗
GenerateDataKeyWithoutPlaintext	✓ Válido solo con claves de cifrado de KMS simétricas	✓	✓
GenerateMac Válido solo en claves HMAC de KMS	✓	✓	✗

AWS KMS Funcionamiento de la API	Claves de varias regiones	Material de claves importado	Las claves KMS en un almacén de claves personalizado
GetKeyPolicy	✓	✓	✓
GetKeyRotationStatus	✓	✓ (KeyRotationEnabled siempre será false.)	✗
GetParametersForImport	✓ Válido solo para claves con material de claves importado (Origin es EXTERNAL).	✓	✗
GetPublicKey Válido solo para claves de KMS asimétricas .	✓	✓	✗

AWS KMS Funcionamiento de la API	Claves de varias regiones	Material de claves importado	Las claves KMS en un almacén de claves personalizado
ImportKeyMaterial	 Válido solo para claves con material de claves importado (Origin es EXTERNAL).		
ListAliases			
ListGrants			
ListKeyPolicies			
ListKeyRotations	 Válido solo en claves de cifrado simétricas con origen AWS_KMS u EXTERNAL origen.	 Válido solo en claves de cifrado simétricas.	
ListResourceTags			
ListRetirableGrants			

AWS KMS Funcionamiento de la API	Claves de varias regiones	Material de claves importado	Las claves KMS en un almacén de claves personalizado
PutKeyPolicy	✓	✓	✓
ReEncrypt	✓ Válido solo cuando KeyUsage es ENCRYPT_D ECRYPT	✓	✓
ReplicateKey	✓ Válido solo en claves principales de varias regiones.	✓ Válido solo en claves principales de varias regiones.	✗
RetireGrant	✓	✓	✓
RevokeGrant	✓	✓	✓
RotateKeyOnDemand	✓ Válido solo en claves de cifrado simétricas con origen AWS_KMS u EXTERNAL origen.	✓ Válido solo en claves de cifrado simétricas.	✗

AWS KMS Funcionamiento de la API	Claves de varias regiones	Material de claves importado	Las claves KMS en un almacén de claves personalizado
ScheduleKeyDeletion	✓	✓	✓
Sign Válido solo cuando KeyUsage es SIGN_VERIFY .	✓	✓	✗
TagResource	✓	✓	✓
UntagResource	✓	✓	✓
UpdateAlias La clave de KMS actual y la nueva clave de KMS deben ser del mismo tipo (ambas simétricas o ambas asimétricas o ambas HMAC) y deben tener el mismo uso de clave .	✓	✓	✓
UpdateKeyDescription	✓	✓	✓
UpdateReplicaRegion	✓	✓ Válido solo en claves de varias regiones.	✗
Verificar Solo es válido cuando KeyUsage es SIGN_VERIFY .	✓	✓	✗

AWS KMS Funcionamiento de la API	Claves de varias regiones	Material de claves importado	Las claves KMS en un almacén de claves personalizado
VerifyMac	✓	✓	✗
Válido solo en claves HMAC de KMS			

Referencia de especificaciones de clave

Cuando crea una clave KMS asimétrica o una clave KMS HMAC, seleccione su [especificación de clave](#). La especificación clave, que es una propiedad de cada AWS KMS key, representa la configuración criptográfica de su clave KMS. Se selecciona la especificación de clave al crear la clave KMS y no se puede cambiar. Si ha seleccionado una especificación de clave errónea, [elimine la clave KMS](#) y cree una nueva.

Note

La especificación de clave para una clave KMS se conocía como “especificación de clave maestra del cliente”. El `CustomerMasterKeySpec` parámetro de la [CreateKey](#) operación está obsoleto. En su lugar, utilice el parámetro `KeySpec`. La respuesta de las [DescribeKey](#) operaciones `CreateKey` y incluye un `CustomerMasterKeySpec` elemento `KeySpec` and con el mismo valor.

La especificación de clave determina si la clave de KMS es simétrica o asimétrica, el tipo de material de clave de la clave de la clave de KMS y los algoritmos de cifrado, de firma o de código de autenticación de mensajes (MAC) AWS KMS compatibles con la clave de KMS. La especificación de clave que seleccione suele estar determinada por el caso de uso y los requisitos normativos. Sin embargo, las operaciones criptográficas en las claves KMS con diferentes especificaciones de clave tienen un precio diferente y están sujetas a diferentes cuotas de solicitud. Para obtener más información sobre precios, consulte [precios de AWS Key Management Service](#). Para obtener más información acerca de las cuotas de solicitud, consulte [Cuotas de solicitudes](#).

Para limitar las especificaciones clave que los directores pueden utilizar al crear claves de KMS, utilice la clave de condición [kms:](#). KeySpec También puede usar la clave de kms :KeySpec condición para permitir que los directores llamen a AWS KMS las operaciones únicamente en claves de KMS con una especificación de clave determinada. Por ejemplo, puede denegar permiso para programar la eliminación de cualquier clave KMS con una especificación de clave RSA_4096.

AWS KMS admite las siguientes especificaciones clave para las claves de KMS:

[Especificaciones de la clave de cifrado simétricas](#) (predeterminado)

- SYMMETRIC_DEFAULT

[Especificaciones de clave de RSA](#) (cifrado y descifrado o firma y verificación)

- RSA_2048
- RSA_3072
- RSA_4096

[Especificaciones de clave de curva elíptica](#)

- [Pares de claves de curvas elípticas asimétricas según el estándar del NIST](#) (para firmar y verificar o derivar secretos compartidos)
 - ECC_NIST_P256 (secp256r1)
 - ECC_NIST_P384 (secp384r1)
 - ECC_NIST_P521 (secp521r1)
 - EDWARDS25519 ECC_NIST_ (ed25519): solo para firma y verificación
 - Nota: Para las claves EDWARDS25519 KMS ECC_NIST_, se requiere el algoritmo de firma _SHA_512, mientras que _PH_SHA_512 lo requiere. ED25519 [MessageType:RAW](#) [ED25519MessageType:DIGEST](#) Estos tipos de mensajes no se pueden usar indistintamente.
- Otros pares de claves de curva elíptica asimétricas (firma y verificación)
 - ECC_SECG_P256K1 ([secp256k1](#)), que se suele utilizar para las criptomonedas.

[SM2 especificación clave](#) (cifrado y descifrado, o firma y verificación, u obtención de secretos compartidos)

- SM2 (Solo regiones de China)

[Especificaciones de la clave HMAC](#)

- HMAC_224
- HMAC_256

- HMAC_384
- HMAC_512

Especificaciones de clave de ML-DSA

- ML_DSA_44
- ML_DSA_65
- ML_DSA_87

Especificación de clave SYMMETRIC_DEFAULT

La especificación de clave predeterminada, SYMMETRIC_DEFAULT, es la especificación de clave para las claves KMS de cifrado simétricas. Al seleccionar el tipo de clave simétrica y el uso de claves de cifrado y descifrado en la AWS KMS consola, se selecciona la especificación de la SYMMETRIC_DEFAULT clave. En la [CreateKey](#) operación, si no especifica ningún KeySpec valor, se selecciona SYMMETRIC_DEFAULT. Si no tiene un motivo para utilizar una especificación de clave diferente, SYMMETRIC_DEFAULT es una buena opción.

SYMMETRIC_DEFAULT representa AES-256-GCM, un algoritmo simétrico basado en el [estándar de cifrado avanzado](#) (AES) en el [modo de contador Galois](#) (GCM) con claves de 256 bits, un estándar del sector para conseguir un cifrado seguro. El texto cifrado que este algoritmo genera admite datos autenticados adicionales (AAD), como un [contexto de cifrado](#), y GCM ofrece una comprobación de integridad adicional en el texto cifrado.

Los datos cifrados con AES-256-GCM están protegidos ahora y en el futuro. Los criptógrafos consideran que este algoritmo es resistente a la informática cuántica. Los futuros e hipotéticos ataques de informática cuántica a gran escala a textos cifrados creados con claves AES-GCM de 256 bits [reducen la seguridad nominal de la clave a 128 bits](#). Sin embargo, este nivel de seguridad es suficiente para hacer inviables los ataques de fuerza bruta contra textos cifrados. AWS KMS

La única excepción en las regiones de China, donde SYMMETRIC_DEFAULT representa una clave simétrica de 128 bits que utiliza cifrado. SM4 Solo puedes crear una SM4 clave de 128 bits en las regiones de China. No puede crear una clave KMS con cifrado AES-GCM de 256 bits en las regiones de China.

Puede utilizar una clave KMS de cifrado simétrico para cifrar, descifrar y volver AWS KMS a cifrar los datos, y para proteger las claves de datos y los pares de claves de datos generados. AWS los servicios integrados AWS KMS utilizan claves KMS de cifrado simétrico para cifrar los datos

en reposo. Puede [importar su propio material de claves](#) en una clave KMS de cifrado simétrica y crear claves KMS de cifrado simétricas en [almacenes de claves personalizadas](#). Para obtener una tabla en la que se comparan las operaciones que puede realizar en las claves de KMS simétricas y asimétricas, consulte [Comparación de claves de KMS simétricas y asimétricas](#).

Puede utilizar una clave KMS de cifrado simétrico para cifrar, descifrar y volver AWS KMS a cifrar los datos, así como para generar claves y pares de claves de datos. Puede crear claves KMS de cifrado simétricas [de varias regiones](#), [importar su propio material de claves](#) en una clave KMS de cifrado simétrica y crear claves KMS de cifrado simétricas en [almacenes de claves personalizadas](#). Para obtener una tabla en la que se comparan las operaciones que puede realizar con claves KMS de diferentes tipos, consulte [Referencia de tipos de claves](#).

Especificaciones de clave de RSA

Cuando utiliza una especificación clave RSA, AWS KMS crea una clave KMS asimétrica con un par de claves RSA. La clave privada nunca sale sin cifrar. AWS KMS Puede usar la clave pública que contiene AWS KMS o descargar la clave pública para usarla fuera de AWS KMS ella.

Warning

Al cifrar datos de forma externa AWS KMS, asegúrese de poder descifrar el texto cifrado. Si utiliza la clave pública de una clave KMS que se ha eliminado de AWS KMS, la clave pública de una clave KMS configurada para firma y verificación o un algoritmo de cifrado que no es compatible con la clave KMS, los datos son irrecuperables.

En AWS KMS, puede utilizar claves KMS asimétricas con pares de claves RSA para el cifrado y el descifrado, o para la firma y la verificación, pero no para ambos. Esta propiedad, conocida como [Key usage](#), se determina independientemente de la especificación de clave, pero tiene que tomar esta decisión antes de seleccionar una especificación de clave.

AWS KMS admite las siguientes especificaciones de claves RSA para el cifrado y el descifrado o para la firma y la verificación:

- RSA_2048
- RSA_3072
- RSA_4096

Las especificaciones de clave de RSA varían en la longitud de la clave de RSA en bits. La especificación de clave de RSA que seleccione puede estar determinada por las normas de seguridad o los requisitos de la tarea. Por regla general, utilice la clave más grande que sea práctica y asequible para la tarea. Las operaciones criptográficas en claves KMS con diferentes especificaciones de clave de RSA tienen un precio diferente. Para obtener información sobre AWS KMS los precios, consulte los precios de los [servicios de administración de AWS claves](#). Para obtener más información acerca de las cuotas de solicitud, consulte [Cuotas de solicitudes](#).

Especificaciones de clave de RSA para el cifrado y el descifrado

Cuando se utiliza una clave KMS asimétrica de RSA para el cifrado y el descifrado, se cifra con la clave pública y se descifra con la clave privada. Cuando llamas a la Encrypt operación AWS KMS para obtener una clave RSA KMS, AWS KMS utiliza la clave pública del par de claves RSA y el algoritmo de cifrado que especifiques para cifrar los datos. Para descifrar el texto cifrado, llame a la Decrypt operación y especifique la misma clave KMS y el mismo algoritmo de cifrado. AWS KMS a continuación, utiliza la clave privada del par de claves RSA para descifrar los datos.

También puede descargar la clave pública y utilizarla para cifrar datos externos a AWS KMS. Asegúrese de utilizar un algoritmo de cifrado AWS KMS compatible con las claves KMS de RSA. Para descifrar el texto cifrado, llame a la función Decrypt con la misma clave KMS y el mismo algoritmo de cifrado.

AWS KMS admite dos algoritmos de cifrado para claves KMS con especificaciones de clave RSA. Estos algoritmos, que se definen en [PKCS #1 v2.2](#), difieren en la función hash que utilizan de forma interna. [En AWS KMS, los algoritmos RSAES_OAEP siempre utilizan la misma función de hash tanto para el procesamiento de hash como para la función de generación de máscaras \(\).](#) MGF1 Tiene que especificar un algoritmo de cifrado cuando llame a las operaciones [Encrypt](#) y [Decrypt](#). Puede elegir un algoritmo diferente para cada solicitud.

Algoritmos de cifrado compatibles con las especificaciones de clave de RSA

Algoritmo de cifrado	Descripción del algoritmo
RSAES_OAEP_SHA_1	PKCS #1 v2.2, sección 7.1. Cifrado RSA con relleno OAEP mediante SHA-1 tanto para la función de generación de hash como para la de generación de máscaras, junto con una etiqueta vacía. MGF1

Algoritmo de cifrado	Descripción del algoritmo
RSAES_OAEP_SHA_256	PKCS #1, sección 7.1. Cifrado RSA con relleno OAEP mediante SHA-256 tanto para la función de generación de hash como para la máscara, junto con una etiqueta vacía. MGF1

No puede configurar una clave KMS para utilizar un algoritmo de cifrado específico. Sin embargo, puede usar la condición de EncryptionAlgorithm política [kms:](#) para especificar los algoritmos de cifrado que los principales pueden usar con la clave KMS.

Para obtener los algoritmos de cifrado de una clave KMS, [consulte la configuración criptográfica](#) de la clave KMS en la AWS KMS consola o utilice la [DescribeKey](#) operación. AWS KMS también proporciona las especificaciones clave y los algoritmos de cifrado al descargar la clave pública, ya sea en la AWS KMS consola o mediante la [GetPublicKey](#) operación.

Puede elegir una especificación de clave de RSA en función de la longitud de los datos de texto no cifrado que puede cifrar en cada solicitud. En la siguiente tabla se muestra el tamaño máximo, en bytes, del texto no cifrado que puede cifrar en una única llamada a la operación [Encrypt](#). Los valores varían en función de la especificación de clave y el algoritmo de cifrado. Para realizar una comparación, puede utilizar una clave KMS de cifrado simétrica para cifrar hasta 4096 bytes a la vez.

Para calcular la longitud máxima del texto plano en bytes para estos algoritmos, utilice la siguiente fórmula: $(key\_size\_in\_bits/8) - (2 * hash\_length\_in\_bits/8) - 2$. Por ejemplo, para RSA_2048 con SHA-256, el tamaño máximo de texto no cifrado en bytes es $(2048/8) - (2 * 256/8) - 2 = 190$.

Tamaño máximo de texto no cifrado (en bytes) en una operación de cifrado

Especificación de clave	Algoritmo de cifrado	
	RSAES_OAEP_SHA_1	RSAES_OAEP_SHA_256
RSA_2048	214	190
RSA_3072	342	318
RSA_4096	470	446

Especificaciones de clave de RSA para la firma y la verificación

Cuando se utiliza una clave KMS asimétrica de RSA para la firma y la verificación, genera la firma para un mensaje con la clave privada y verifica la firma con la clave pública.

Cuando llamas a la `Sign` operación AWS KMS para obtener una clave KMS asimétrica, AWS KMS utiliza la clave privada del par de claves RSA, el mensaje y el algoritmo de firma que especifiques para generar una firma. Para verificar la firma, llame a la operación [Verify](#). Especifique la firma, además de la misma clave KMS, mensaje y algoritmo de firma. AWS KMS luego usa la clave pública del par de claves RSA para verificar la firma. También puede descargar la clave pública y utilizarla para verificar la firma fuera de AWS KMS.

AWS KMS admite los siguientes algoritmos de firma para todas las claves de KMS con una especificación de clave RSA. Tiene que especificar un algoritmo de firma cuando llame a las operaciones [Sign](#) y [Verify](#). Puede elegir un algoritmo diferente para cada solicitud. Al firmar con pares de claves de RSA, se prefieren los algoritmos RSASSA-PSS. Incluimos los algoritmos RSASSA-PKCS1-v1_5 para garantizar la compatibilidad con las aplicaciones existentes.

Algoritmos de firma compatibles con las especificaciones de clave de RSA

Algoritmo de firma	Descripción del algoritmo
RSASSA_PSS_SHA_256	PKCS #1 v2.2, sección 8.1, firma RSA con relleno PSS con SHA-256 tanto para el resumen del mensaje como para la función de generación de máscaras, junto con una sal de 256 bits MGF1
RSASSA_PSS_SHA_384	PKCS #1 v2.2, sección 8.1, firma RSA con relleno PSS con SHA-384 tanto para el resumen del mensaje como para la función de generación de máscaras, junto con una sal de 384 bits MGF1
RSASSA_PSS_SHA_512	PKCS #1 v2.2, sección 8.1, firma RSA con relleno PSS utilizando el SHA-512 tanto para el resumen del mensaje como para la función de generación de máscaras, junto con una sal de 512 bits MGF1

Algoritmo de firma	Descripción del algoritmo
RSASSA_ _V1_5_SHA_256_PKCS1	PKCS #1 v2.2, sección 8.2, firma de RSA con relleno PKCS #1 v1.5 y SHA-256
RSASSA_PKCS1_ _V1_5_SHA_384	PKCS #1 v2.2, sección 8.2, firma de RSA con relleno PKCS #1 v1.5 y SHA-384
RSASSA_PKCS1_ _V1_5_SHA_512	PKCS #1 v2.2, sección 8.2, firma de RSA con relleno PKCS #1 v1.5 y SHA-512

No puede configurar una clave de KMS para utilizar algoritmos de firma específicos. Sin embargo, puede usar la condición [kms: SigningAlgorithm](#) policy para especificar los algoritmos de firma que los directores pueden usar con la clave KMS.

Para obtener los algoritmos de firma de una clave KMS, [consulte la configuración criptográfica](#) de la clave KMS en la AWS KMS consola o utilice la [DescribeKey](#) operación. AWS KMS también proporciona las especificaciones clave y los algoritmos de firma al descargar la clave pública, ya sea en la AWS KMS consola o mediante la [GetPublicKey](#) operación.

Especificaciones de clave de curva elíptica

Cuando utiliza una especificación clave de curva elíptica (ECC), AWS KMS crea una clave KMS asimétrica con un par de claves ECC para firmar y verificar o para derivar secretos compartidos (pero no ambos). La clave privada que genera firmas u obtiene secretos compartidos nunca deja AWS KMS sin cifrar. Puede usar la clave pública para [verificar las firmas](#) internas AWS KMS o [descargar la clave pública para usarla fuera de ella](#). AWS KMS

Cuando utiliza una especificación de clave Edwards Curve, AWS KMS crea una clave KMS asimétrica con un par de claves Ed25519 para la firma y la verificación. La clave privada que genera las firmas nunca se queda sin cifrar. AWS KMS Puedes usar la clave pública para [verificar las firmas](#) internas AWS KMS o [descargar la clave pública](#) para usarla fuera de AWS KMS ella.

AWS KMS admite las siguientes especificaciones clave ECC para claves KMS asimétricas.

- Pares de claves de curvas elípticas asimétricas según el estándar del NIST (para firmar y verificar o derivar secretos compartidos)
 - ECC_NIST_P256 (secp256r1)

- ECC_NIST_P384 (secp384r1)
- ECC_NIST_P521 (secp521r1)
- EDWARDS25519 ECC_NIST_ (ed25519): solo para firma y verificación
 - Nota: Para las claves EDWARDS25519 KMS ECC_NIST_, se requiere el algoritmo de firma _SHA_512, mientras que _PH_SHA_512 lo requiere. ED25519 [MessageType:RAW](#) [ED25519MessageType:DIGEST](#) Estos tipos de mensajes no se pueden usar indistintamente.
- Otros pares de claves de curva elíptica asimétricas (firma y verificación)
 - ECC_SECG_P256K1 ([secp256k1](#)), que se suele utilizar para las criptomonedas.

La especificación de clave de ECC que selecciona puede estar determinada por las normas de seguridad o los requisitos de la tarea. Por regla general, utilice la curva con más puntos que sea práctica y asequible para la tarea.

Si va a crear una clave KMS asimétrica para [obtener secretos compartidos](#), utilice una de las especificaciones de clave de curva elíptica estándar del NIST (excepto ECC_SECG_P256K1 y ECC_NIST_). EDWARDS25519 El único algoritmo de acuerdo de claves compatible para obtener secretos compartidos es el [cofactor criptográfico de curva elíptica primitivo Diffie-Hellman](#) (ECDH). Para ver un ejemplo sobre cómo obtener secretos compartidos sin conexión, consulte [the section called “Obtención de secretos compartidos sin conexión”](#).

Si va a crear una clave KMS asimétrica para utilizarla con criptomonedas, utilice la especificación de clave ECC_SECG_P256K1. También puede utilizar esta especificación de clave para otros fines, pero es obligatoria para Bitcoin y otras criptomonedas.

En la siguiente tabla se muestran los algoritmos de firma compatibles con cada una de las especificaciones clave del ECC. AWS KMS No puede configurar una clave de KMS para utilizar algoritmos de firma específicos. Sin embargo, puede usar la condición de SigningAlgorithm política [kms:](#) para especificar los algoritmos de firma que los principales pueden usar con la clave KMS.

Algoritmos de firma compatibles con las especificaciones de clave de ECC

Especificación de clave	Algoritmo de firma	Descripción del algoritmo
ECC_NIST_P256	ECDSA_SHA_256	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and

Especificación de clave	Algoritmo de firma	Descripción del algoritmo
		SHA-256 para el resumen del mensaje.
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 el resumen del mensaje.
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 el resumen del mensaje.
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 el resumen del mensaje.
ECC_NIST_EDWARDS25519	ED25519_SHA_512	NIST FIPS 186-5, Section 7.6, EdDSA signature using the curve specified by the key and SHA-512 para el resumen del mensaje. KMS requiere MessageType:RAW este algoritmo.

Especificación de clave	Algoritmo de firma	Descripción del algoritmo
ECC_NIST_EDWARDS25519	ED25519_PH_SHA_512	NIST FIPS 186-5, Section 7.8, HashEdDSA signature using the curve specified by the key and SHA-512 para el resumen del mensaje. KMS requiere MessageType:DIGEST este algoritmo. Al especificar el algoritmo de firma ED25519_PH_SHA_512 con MessageType:DIGEST , AWS KMS sigue realizando o el prehash del SHA-512 descrito en el paso 1 de la sección 7.8.1 . Esto significa que la entrada se codifica dos veces: una por usted y otra por KMS.

Especificaciones de la clave para las claves KMS HMAC

AWS KMS admite claves HMAC simétricas de diferentes longitudes. La especificación de clave que seleccione puede depender de sus requisitos normativos, de seguridad o empresariales. La longitud de la clave determina el algoritmo MAC que se utiliza en las operaciones [GenerateMac](#) [VerifyMac](#). En general, las claves más largas son más seguras. Utilice la clave más larga que sea práctica para su caso de uso.

Especificación de la clave HMAC	Algoritmo MAC
HMAC_224	HMAC_SHA_224
HMAC_256	HMAC_SHA_256
HMAC_384	HMAC_SHA_384
HMAC_512	HMAC_SHA_512

Especificaciones clave del ML-DSA

Una clave ML-DSA es una clave criptográfica utilizada en el algoritmo de firma Module-Lattice-Based digital (ML-DSA), que está diseñado para la criptografía poscuántica. Este algoritmo forma parte del esfuerzo de estandarización del National Institute of Standards and Technology (Instituto Nacional de Estándares y Tecnología, NIST), descrito específicamente en las [Normas Federales de Procesamiento de la Información \(FIPS\) 204](#).

Las claves de ML-DSA se utilizan en un sistema de pares de claves público-privadas. La clave privada se usa para firmar datos, mientras que la clave pública se usa para verificar la firma. Este sistema garantiza la autenticidad, la integridad y el no rechazo de los mensajes o documentos digitales, incluso ante posibles amenazas informáticas de computación cuántica.

Al crear una clave con la especificación de clave ML-DSA, AWS KMS crea una clave KMS asimétrica con un par de claves ML-DSA para la firma y la verificación. La clave privada que genera las firmas nunca se queda sin cifrar. AWS KMS Puedes usar la clave pública para [verificar las firmas](#) internas AWS KMS o [descargar la clave pública](#) para usarla fuera de AWS KMS ella.

AWS KMS admite las siguientes especificaciones clave ML-DSA para claves KMS asimétricas:

- ML_DSA_44
- ML_DSA_65
- ML_DSA_87

AWS KMS admite el algoritmo de firma ML_DSA_SHAKE_256 para todas las especificaciones clave del ML-DSA.

SM2 especificación clave (solo regiones de China)

La especificación SM2 clave es una especificación clave de curva elíptica definida dentro de la GM/T serie de especificaciones publicadas por la [Oficina de Administración Estatal de Criptografía Comercial \(OSCCA\) de China](#). La especificación SM2 clave solo está disponible en las regiones de China. Cuando utiliza la especificación SM2 clave, AWS KMS crea una clave KMS asimétrica con un par de SM2 claves. Puede usar su par de SM2 claves dentro AWS KMS o descargar la clave pública para usarla fuera de ella AWS KMS. Para obtener más información, consulte [the section called “Verificación fuera de línea con pares de SM2 claves \(solo en las regiones de China\)”](#).

Cada clave de KMS solo puede tener un [Key usage](#). Puedes usar una clave SM2 KMS para firmar y verificar, cifrar y descifrar o para obtener secretos compartidos. Debe especificar el uso de claves al crear la clave KMS y no se la puede cambiar una vez creada.

Si va a crear una clave KMS asimétrica para [obtener secretos compartidos](#), utilice la especificación de la SM2 clave. El único algoritmo de acuerdo de claves compatible para obtener secretos compartidos es el [cofactor criptográfico de curva elíptica primitivo Diffie-Hellman](#) (ECDH).

AWS KMS admite los siguientes algoritmos de SM2 cifrado y firma:

- SM2Algoritmo de cifrado PKE

SM2EI PKE es un algoritmo de cifrado basado en curvas elípticas definido por la OSCCA en 0003.4-2012. GM/T

- SM2Algoritmo de firma DSA

SM2EI DSA es un algoritmo de firma basado en curvas elípticas definido por la OSCCA en 0003.2-2012. GM/T SM2EI DSA requiere un identificador distintivo que se compila con el algoritmo de cifrado y, a continuación, se combina con SM3 el mensaje, o resumen del mensaje, al que se ha hecho la transferencia. AWS KMS A continuación, este valor concatenado se codifica con un hash y se firma por. AWS KMS

AWS KMS permisos

Esta tabla está diseñada para ayudarle a entender AWS KMS los permisos para que pueda controlar el acceso a sus AWS KMS recursos. Las definiciones de los títulos de las columnas aparecen bajo la tabla.

También puede obtener información sobre AWS KMS los permisos en las [claves de acciones, recursos y condición del AWS Key Management Service tema de](#) la Referencia de autorización de servicios. Sin embargo, ese tema no enumera todas las claves de condición que puede utilizar para limitar cada permiso.

Para obtener más información sobre qué AWS KMS operaciones son válidas para las claves KMS de cifrado simétrico, las claves KMS asimétricas y las claves KMS HMAC, consulte la. [Referencia de tipos de claves](#)

Note

Es posible que tenga que desplazarse en forma horizontal o vertical para ver todos los datos de la tabla.

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
CancelKeyDeletion kms:CancelKeyDeletion	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService
ConnectCustomKeyStore kms:ConnectCustomKeyStore	Política de IAM	No	*	km: CallerAccount

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
CreateAlias kms:CreateAlias	Política de IAM (para el alias)	No	Alias	Ninguno (cuando se controla el acceso al alias)
Para utilizar esta operación, el intermediario necesita permiso kms:CreateAlias en dos recursos: • El alias (en una política de IAM) • La clave KMS (en una política de clave) Para obtener más información, consulte Control del acceso a alias.	Política de claves (para la clave KMS)	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService
CreateCustomKeyStore kms:CreateCustomKeyStore	Política de IAM	No	*	km: CallerAccount

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
CreateGrant kms:CreateGrant	Política de claves	Sí	Clave KMS	Condiciones de contexto de cifrado: kmsEncryptionContext: clave de contexto kms: EncryptionContextKeys Condiciones de concesión: km: GrantConstraintSourceArn km: GrantConstraintType km: GranteePrincipal km: GranteeServicePrincipal km: GrantIsForAWSResource km: GrantOperations km: RetiringPrincipal km: RetiringServicePrincipal Condiciones para las operaciones clave KMS: km: CallerAccount

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
				km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
CreateKey kms:CreateKey	Política de IAM	No	*	km: BypassPolicyLockoutSafetyCheck km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ViaService aws:RequestTag/tag-key (clave de condición AWS global) aws:ResourceTag/tag-key (clave de condición global)AWS aws: TagKeys (clave de condición AWS global)

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
Decrypt kms:Decrypt	Política de claves	Sí	Clave KMS	Condiciones para operaciones criptográficas kms: EncryptionAlgorithm km: RequestAlias Condiciones de contexto de cifrado: kmsEncryptionContext: clave de contexto kms: EncryptionContextKeys Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
				aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService
DeleteAlias kms:DeleteAlias	Política de IAM (para el alias)	No	Alias	Ninguno (cuando se controla el acceso al alias)
Para utilizar esta operación, el intermediario necesita permiso kms:DeleteAlias en dos recursos: - El alias (en una política de IAM) - La clave KMS (en una política de clave) Para obtener más información, consulte Control del acceso a alias.	Política de claves (para la clave KMS)	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
DeleteCustomKeyStore kms:DeleteCustomKeyStore	Política de IAM	No	*	km: CallerAccount
DeleteImportedKeyMaterial kms:DeleteImportedKeyMaterial	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
DeriveSharedSecret kms:DeriveSharedSecret	Política de claves	Sí	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Condiciones para operaciones criptográficas: km: KeyAgreementAlgorithm
DescribeCustomKeyStores kms:DescribeCustomKeyStores	Política de IAM	No	*	km: CallerAccount

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
DescribeKey kms:DescribeKey	Política de claves	Sí	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Otras condiciones: km: RequestAlias

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
DisableKey kms:DisableKey	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Última condición clave de uso: km: TrailingDaysWithoutKeyUsage

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
DisableKeyRotation kms:DisableKeyRotation	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService
DisconnectCustomKeyStore kms:DisconnectCustomKeyStore	Política de IAM	No	*	km: CallerAccount

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
EnableKey kms:EnableKey	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
EnableKeyRotation <code>kms:EnableKeyRotation</code>	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Condiciones de rotación automática de claves: km: RotationPeriodInDays

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
Encrypt kms:Encrypt	Política de claves	Sí	Clave KMS	Condiciones para operaciones criptográficas km: EncryptionAlgorithm km: RequestAlias Condiciones de contexto de cifrado: kmsEncryptionContext: clave de contexto kms: EncryptionContextKeys Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
				aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
GenerateDataKey kms:GenerateDataKey	Política de claves	Sí	Clave KMS	Condiciones para operaciones criptográficas km: EncryptionAlgorithm km: RequestAlias Condiciones de contexto de cifrado: kmsEncryptionContext: clave de contexto kms: EncryptionContextKeys Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
				aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
GenerateDataKeyPair <code>kms:GenerateDataKeyPair</code>	Política de claves	Sí	Clave de KMS Genera un par de claves de datos asimétricos protegido por una clave de KMS de cifrado simétrica.	Condiciones para pares de claves de datos: km: DataKeyPairSpec Condiciones para operaciones criptográficas km: EncryptionAlgorithm km: RequestAlias Condiciones de contexto de cifrado: kmsEncryptionContext: clave de contexto kms: EncryptionContextKeys Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
				km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
GenerateDataKeyPairWithoutPlaintext <code>kms:GenerateDataKeyPairWithoutPlaintext</code>	Política de claves	Sí	Clave de KMS Genera un par de claves de datos asimétricos protegido por una clave de KMS de cifrado simétrica.	Condiciones para pares de claves de datos: km: DataKeyPairSpec Condiciones para operaciones criptográficas km: EncryptionAlgorithm km: RequestAlias Condiciones de contexto de cifrado: kmsEncryptionContext: clave de contexto kms: EncryptionContextKeys Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
				km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
GenerateDataKeyWithoutPlaintext kms:GenerateDataKeyWithoutPlaintext	Política de claves	Sí	Clave KMS	Condiciones para operaciones criptográficas km: EncryptionAlgorithm km: RequestAlias Condiciones de contexto de cifrado: kmsEncryptionContext: clave de contexto kms: EncryptionContextKeys Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
				aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService
GenerateMac kms:GenerateMac	Política de claves	Sí	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Condiciones para operaciones criptográficas: km: MacAlgorithm km: RequestAlias

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
GenerateRandom kms:GenerateRandom	Política de IAM	N/A	*	Ninguno
GetKeyLastUsage kms:GetKeyLastUsage	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
GetKeyPolicy kms:GetKeyPolicy	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
GetKeyRotationStatus kms:GetKeyRotationStatus	Política de claves	Sí	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
GetParametersForImport kms:GetParametersForImport	Política de claves	No	Clave KMS	km: WrappingAlgorithm km: WrappingKeySpec Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
GetPublicKey kms:GetPublicKey	Política de claves	Sí	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Otras condiciones: km: RequestAlias

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
ImportKeyMaterial kms:ImportKeyMaterial	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Otras condiciones: km: ExpirationModel km: ValidTo
ListAliases kms:ListAliases	Política de IAM	No	*	Ninguno

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
ListGrants kms:ListGrants	Política de claves	Sí	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Otras condiciones: km: GrantIsForAWSResource

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
ListKeyPolicies kms:ListKeyPolicies	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
ListKeyRotations kms:ListKeyRotations	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService
ListKeys kms:ListKeys	Política de IAM	No	*	Ninguno

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
ListResourceTags <code>kms:ListResourceTags</code>	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
ListRetirableGrants kms:ListRetirableGrants	Política de IAM	La principal entidad especificada debe estar en la cuenta local, pero la operación devuelve concesiones en todas las cuentas.	*	Ninguno

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
PutKeyPolicy kms:PutKeyPolicy	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Otras condiciones: km: BypassPolicyLockoutSafetyCheck

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
ReEncrypt <code>kms:ReEncryptFrom</code> <code>kms:ReEncryptTo</code> Para utilizar esta operación, la persona que llama necesita permiso en dos claves KMS: <code>kms:ReEncryptFrom</code> en la clave KMS utilizada para descifrar <code>kms:ReEncryptTo</code> en la clave KMS utilizada para cifrar	Política de claves	Sí	Clave KMS	Condiciones para operaciones criptográficas km: EncryptionAlgorithm km: RequestAlias Condiciones de contexto de cifrado: kmsEncryptionContext: clave de contexto kms: EncryptionContextKeys Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
				aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Otras condiciones: km: ReEncrypt OnSameKey

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
ReplicateKey <code>kms:ReplicateKey</code> Para utilizar esta operación, la persona que llama necesita estos permisos: • <code>kms:ReplicateKey</code> en la clave principal de varias regiones • <code>kms:CreateKey</code> en una política de IAM en la región de réplica	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Otras condiciones: km: ReplicaRegion

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
RetireGrant <code>kms:RetireGrant</code> El permiso para retirar una concesión se determina principalmente por la concesión. Una política por sí sola no puede permitir el acceso a esta operación. Para obtener más información, consulte Retiro y revocación de concesiones.	Política de IAM (Este permiso no es efectivo en una política de clave.)	Sí	Clave KMS	Condiciones de contexto de cifrado: kmsEncryptionContext: clave de contexto kms: EncryptionContextKeys Condiciones de concesión: km: GrantConstraintType Condiciones para las operaciones clave KMS: km: CallerAccount km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
RevokeGrant <code>kms:RevokeGrant</code>	Política de claves	Sí	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Otras condiciones: km: GrantIsForAWSResource

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
RotateKeyOnDemand kms:RotateKeyOnDemand	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
ScheduleKeyDeletion kms:ScheduleKeyDeletion	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Última condición clave de uso: km: TrailingDaysWithoutKeyUsage

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
Sign kms:Sign	Política de claves	Sí	Clave de KMS	Condiciones para la firma y la verificación: km: MessageType km: RequestAlias km: SigningAlgorithm Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
TagResource kms:TagResource	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Condiciones para el etiquetado: aws:RequestTag/tag-key (clave de condición AWS global) aws: TagKeys (clave de condición AWS global)

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
UntagResource kms:UntagResource	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: kms: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Condiciones para el etiquetado: aws:RequestTag/tag-key (clave de condición AWS global) aws: TagKeys (clave de condición AWS global)

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
UpdateAlias kms:UpdateAlias	Política de IAM (para el alias)	No	Alias	Ninguno (cuando se controla el acceso al alias)
Para utilizar esta operación, el intermediario necesita permiso kms:UpdateAlias en tres recursos: • El alias • La clave KMS asociada actualmente • La clave KMS recién asociada Para obtener más información, consulte Control del acceso a alias.	Política de claves (para las claves KMS)	No	Clave KMS	Condiciones para las operaciones clave KMS: kms: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService
UpdateCustomKeyStore kms:UpdateCustomKeyStore	Política de IAM	No	*	km: CallerAccount

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
UpdateKeyDescription kms:UpdateKeyDescription	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
UpdatePrimaryRegion <code>kms:UpdatePrimaryRegion</code> Para utilizar esta operación, la persona que llama necesita permiso <code>kms:UpdatePrimaryRegion</code> tanto en la clave principal de varias regiones que se convertirá en una clave de réplica como en la clave de réplica de varias regiones que se convertirá en la clave principal.	Política de claves	No	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Otras condiciones km: PrimaryRegion

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
Verificar kms:Verify	Política de claves	Sí	Clave de KMS	Condiciones para la firma y la verificación: km: MessageType km: RequestAlias km: SigningAlgorithm Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService

Acciones y permisos	Tipo de política	Cross-account utilizar	Recursos (para políticas de IAM)	AWS KMS claves de condición
VerifyMac kms:VerifyMac	Política de claves	Sí	Clave KMS	Condiciones para las operaciones clave KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (clave de condición AWS global) kms: ViaService Condiciones para operaciones criptográficas: km: MacAlgorithm km: RequestAlias

Descripciones de las columnas

Las columnas de esta tabla proporcionan la siguiente información:

- Acciones y permisos muestra cada operación de la AWS KMS API y el permiso que permite la operación. Especifique la operación en el elemento `Action` de una declaración de política.
- Tipo de política indica si el permiso se puede utilizar en una política de claves o en una política de IAM.

Política de claves significa que puede especificar el permiso en la política de claves. Cuando la política de claves contiene la [declaración de política que permite las políticas de IAM](#), puede especificar el permiso en una política de IAM.

La política de IAM significa que puede especificar el permiso solo en la política de IAM.

- Cross-account Use muestra las operaciones que los usuarios autorizados pueden realizar en los recursos de un sitio diferente Cuenta de AWS.

Un valor de Yes (Sí) significa que las entidades principales pueden realizar la operación en los recursos en una Cuenta de AWS diferente.

Un valor de No significa que las principales entidades pueden realizar la operación solo en recursos en sus propios Cuenta de AWS.

Si otorga a una entidad principal de una cuenta diferente un permiso que no se puede utilizar en un recurso entre cuentas, el permiso no será efectivo. Por ejemplo, si le das `TagResource` permiso a un director de una cuenta diferente para [usar](#) una clave de KMS en tu cuenta, sus intentos de etiquetar la clave de KMS en tu cuenta fallarán.

- Recursos muestra los AWS KMS recursos a los que se aplican los permisos. AWS KMS admite dos tipos de recursos: una clave de KMS y un alias. En una política de claves, el valor del elemento `Resource` es siempre `*`, lo que indica la clave KMS a la que está asociada la política de claves.

Utilice los siguientes valores para representar un AWS KMS recurso en una política de IAM.

Clave de KMS

Cuando el recurso sea una clave KMS, utilice su [ARN de clave](#). Para obtener ayuda, consulte [the section called “Encontrar el ID de clave y el ARN de clave”](#).

`arn:AWS_partition_name:kms:AWS_Region:AWS_account_ID:key/key_ID`

Por ejemplo:

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

Alias

Cuando el recurso sea un alias, utilice su [ARN de alias](#). Para obtener ayuda, consulte [the section called “Buscar el nombre del alias y el ARN de alias”](#).

`arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:alias/alias_name`

Por ejemplo:

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

***** (asterisco)

Cuando el permiso no se aplique a un recurso determinado (clave KMS o alias), utilice un asterisco (*).

En una política de IAM para un AWS KMS permiso, un asterisco en el elemento indica todos los recursos (claves y alias de KMS). Resource AWS KMS También puedes usar un asterisco en el Resource elemento cuando el AWS KMS permiso no se aplique a ninguna clave o alias de KMS en concreto. Por ejemplo, al permitir o denegar un permiso `kms:CreateKey` o `kms:ListKeys`, debe establecer el elemento Resource en `*`.

- AWS KMS las claves de condición enumeran las claves de AWS KMS condición que puede usar para controlar el acceso a la operación. Las condiciones se especifican en un elemento Condition de la política. Para obtener más información, consulte [AWS KMS claves de condición](#). En esta columna también se incluyen [las claves de condición AWS globales](#) que son compatibles con todos los AWS servicios AWS KMS, pero no con todos ellos.

AWS KMS operaciones internas

AWS Key Management Service (AWS KMS) proporciona claves y operaciones criptográficas protegidas por [módulos de seguridad de hardware \(HSM\) validados por el FIPS 140-3 de nivel de seguridad \(HSM\)](#) y escalados para la nube. AWS KMS Varios servicios en la AWS nube utilizan las claves y las funciones, y usted puede utilizarlas para proteger los datos de sus aplicaciones. Esta guía técnica proporciona detalles sobre las operaciones criptográficas que se ejecutan AWS cuando se utilizan AWS KMS.

AWS KMS los componentes internos son necesarios para escalar y proteger HSMs un servicio de administración de claves distribuido a nivel mundial.

Temas

- [Dominios y estado de dominio](#)
- [Seguridad de la comunicación interna](#)
- [Proceso de replicación de claves multirregión](#)
- [Protección de durabilidad](#)

Dominios y estado de dominio

Un conjunto cooperativo de AWS KMS entidades internas de confianza dentro de un dominio. Región de AWS se denomina dominio. Un dominio incluye un conjunto de entidades de confianza, un conjunto de reglas y un conjunto de claves secretas, denominadas claves de dominio. Las claves de dominio se comparten entre HSMs los miembros del dominio. Un estado de dominio consta de los siguientes campos.

Name

Un nombre de dominio para identificar este dominio.

Miembros

Una lista de los HSMs que son miembros del dominio, incluida su clave de firma pública y sus claves de acuerdo públicas.

Operadores

Una lista de entidades, claves de firma públicas y un rol (AWS KMS operador o anfitrión del servicio) que representa a los operadores de este servicio.

Reglas

Una lista de reglas de quórum para cada comando que se deben cumplir para ejecutar un comando en el HSM.

Claves de dominio

Una lista de claves de dominio (claves simétricas) actualmente en uso dentro del dominio.

El estado de dominio completo solo está disponible en el HSM. El estado de dominio se sincroniza entre los miembros del dominio de HSM como un token de dominio exportado.

Claves de dominio

Todos los HSMs elementos de un dominio comparten un conjunto de claves de dominio, $\{DK_r\}$. Estas claves se comparten a través de una rutina de exportación de estado de dominio. El estado de dominio exportado se puede importar a cualquier HSM que sea miembro del dominio.

El conjunto de claves de dominio, $\{DK_r\}$, siempre incluye una clave de dominio activa y varias claves de dominio desactivadas. Las claves de dominio se rotan a diario para garantizar que AWS cumplen con la [recomendación para la gestión de claves \(parte 1\)](#). Durante la rotación de claves de dominio, todas las claves de KMS existentes cifradas con la clave de dominio saliente se vuelven a cifrar con la nueva clave de dominio activa. La clave de dominio activa se utiliza para cifrar cualquier clave nueva. Las claves de dominio caducadas solo se pueden usar para descifrar previamente cifradas EKTs durante un número de días equivalente al número de claves de dominio rotadas recientemente.

Tokens de dominio exportados

Existe una necesidad periódica de sincronizar el estado entre los participantes del dominio. Esto se logra al exportar el estado de dominio cada vez que se realiza un cambio en el dominio. El estado de dominio se exporta como un token de dominio exportado.

Name

Un nombre de dominio para identificar este dominio.

Miembros

Una lista de las HSMs que son miembros del dominio, incluidas sus claves públicas de firma y acuerdo.

Operadores

Una lista de entidades, claves de firma públicas y un rol que representa a los operadores de este servicio.

Reglas

Una lista de reglas de quórum para cada comando que se deben cumplir para ejecutar un comando en un miembro de dominio de HSM.

Claves de dominio cifradas

Claves de dominio con cifrado doble. El miembro firmante cifra las claves de dominio para cada uno de los miembros enumerados anteriormente y las vuelve a cifrar con su clave de acuerdo pública.

Signature

Una firma en el estado de dominio producida por un HSM, necesariamente el miembro del dominio que exportó el estado de dominio.

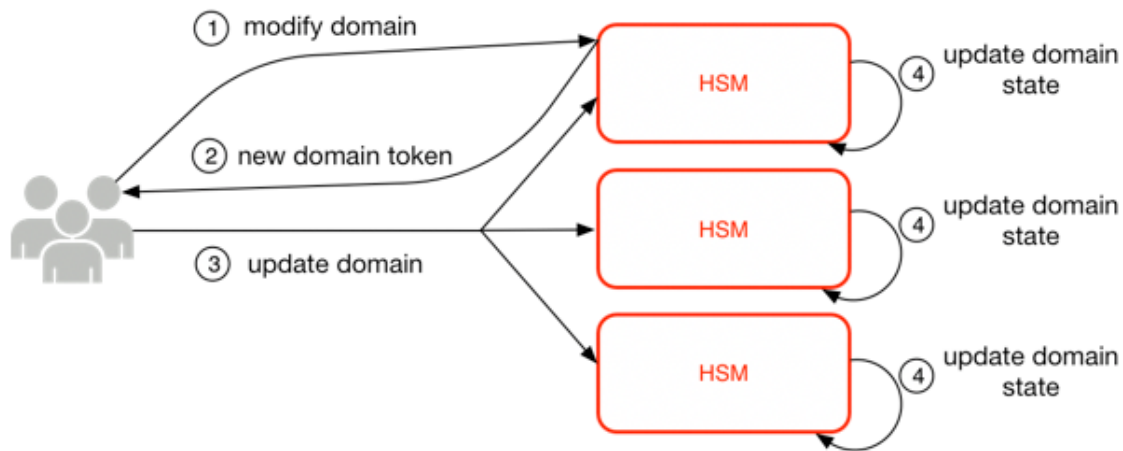
El token de dominio exportado forma la fuente fundamental de confianza para las entidades que operan dentro del dominio.

Administración de estados de dominio

El estado de dominio se administra a través de comandos autenticados con quórum. Estos cambios incluyen la actualización de la lista de participantes de confianza en el dominio, la modificación de las reglas de quórum para ejecutar comandos de HSM y la rotación periódica de las claves de dominio. Estos comandos se autentican por cada comando en lugar de operaciones de sesión autenticadas, como se muestra en la imagen siguiente.

Un HSM, en su estado inicializado y operativo, contiene un conjunto de claves de identidad asimétricas generadas de forma automática, un par de claves de firma y un par de claves de establecimiento de claves. Mediante un proceso manual, un AWS KMS operador puede establecer un dominio inicial que se creará en el primer HSM de una región. Este dominio inicial consta de un estado de dominio completo como se definió anteriormente en este tema. Se instala mediante un comando de unión para cada uno de los miembros del HSM definidos en el dominio.

Una vez que un HSM se haya unido a un dominio inicial, este estará vinculado a las reglas que se definen en ese dominio. Estas reglas rigen los comandos que utilizan claves criptográficas de cliente o realizan cambios en el estado del anfitrión o del dominio. Las operaciones de API de sesión autenticadas que utilizan las claves criptográficas se han definido anteriormente.



La imagen anterior muestra cómo se modifica un estado de dominio. El proceso consta de cuatro pasos:

1. Se envía un comando basado en quórum a un HSM para modificar el dominio.
2. El nuevo estado de dominio se crea y se exporta como un nuevo token de dominio exportado. El estado en el HSM no se modifica, lo que significa que el cambio no se implementa en el HSM.
3. Se envía un segundo comando a cada uno de los componentes del HSMs token de dominio recién exportado para actualizar el estado de su dominio con el nuevo token de dominio.
4. Lo que HSMs aparece en el nuevo token de dominio exportado puede autenticar el comando y el token de dominio. También pueden desempaquetar las claves de dominio para actualizar el estado del dominio HSMs en todo el dominio.

HSMs no se comunican directamente entre sí. En su lugar, un quórum de operadores solicita un cambio en el estado de dominio que da como resultado un nuevo token de dominio exportado. Un miembro del anfitrión de servicio del dominio se utiliza para distribuir el nuevo estado de dominio a cada HSM del dominio.

La salida y unión de un dominio se realizan a través de las funciones de gestión de HSM. La modificación del estado de dominio se realiza a través de las funciones de gestión del dominio.

Salir del dominio

Hace que un HSM salga de un dominio, lo que elimina todos los restos y claves de ese dominio de la memoria.

Unirse al dominio

Hace que un HSM se una a un nuevo dominio o actualice su estado de dominio actual al nuevo estado del dominio. El dominio existente se utiliza como fuente del conjunto inicial de reglas para autenticar este mensaje.

Crear un dominio

Hace que se cree un nuevo dominio en un HSM. Devuelve un primer token de dominio que se puede distribuir a los miembros HSMs del dominio.

Modificar operadores

Agrega o elimina operadores de la lista de operadores autorizados y sus respectivos roles en el dominio.

Modificar miembros

Agrega o elimina un HSM de la lista de autorizados HSMs en el dominio.

Modificar reglas

Modifica el conjunto de reglas de quórum necesario para ejecutar comandos en un HSM.

Rotar claves de dominio

Hace que se cree una nueva clave de dominio y que esta se marque como la clave de dominio activa. De esta manera, se mueve la clave activa existente a una clave desactivada y elimina la clave desactivada más antigua del estado de dominio.

Seguridad de la comunicación interna

Los comandos entre los hosts u AWS KMS operadores del servicio HSMs se protegen mediante dos mecanismos descritos en [Sesiones autenticadas](#): un método de solicitud firmado por quórum y una sesión autenticada mediante un protocolo de host de servicio HSM.

Los comandos firmados por quórum están diseñados para que ningún operador pueda modificar por sí solo las protecciones de seguridad críticas que proporcionan. HSMs Los comandos que se ejecutan en las sesiones autenticadas ayudan a garantizar que solo los operadores de servicio autorizados puedan realizar operaciones con claves de KMS. Toda la información secreta vinculada al cliente está protegida en toda la infraestructura. AWS

Establecimiento de claves

Para proteger las comunicaciones internas, AWS KMS utiliza dos métodos diferentes de establecimiento de claves. El primero se define como C (1, 2, ECC DH) en la [Recomendación para los esquemas de establecimiento de claves por pares que utilizan criptografía de logaritmos discretos](#). Este esquema tiene un iniciador con una clave de firma estática. El iniciador genera y firma una clave de ephemeral elliptic curve Diffie-Hellman (ECDH, curva elíptica de Diffie-Hellman efímera), dirigida a un destinatario con una clave de acuerdo de ECDH estática. Este método utiliza dos claves estáticas y dos claves efímeras, cada una mediante el uso de ECDH. Esa es la derivación de la etiqueta C (1, 2, ECC DH). Este método a veces se denomina ECDH directa.

El segundo método de establecimiento de claves es [C \(2, 2, ECC, DH\)](#). En este esquema, ambas partes tienen una clave de firma estática y generan, firman e intercambian una clave de ECDH efímera. Este método utiliza dos claves estáticas y dos claves efímeras, cada una mediante el uso de ECDH. Esa es la derivación de la etiqueta C (2, 2, ECC, DH). Este método a veces se denomina ECDH efímero o ECDHE. Todas las claves ECDH se generan en la curva secp384r1 (NIST-P384).

Límite de seguridad del HSM

El límite de seguridad interior de AWS KMS es el HSM. El HSM tiene una interfaz propietaria y ninguna otra interfaz de activación física en su estado operativo. Un HSM operativo se aprovisionará durante la inicialización con las claves criptográficas necesarias para establecer su rol en el dominio. Los materiales criptográficos sensibles del HSM solo se almacenan en la memoria volátil y se borran cuando el HSM sale del estado operativo, incluidos los bloqueos o reinicios previstos o no intencionados.

Las operaciones de la API del HSM se autentican mediante comandos individuales o mediante una sesión confidencial mutuamente autenticada establecida por un anfitrión de servicio.

Comandos firmados por quórum

Los operadores emiten comandos firmados por quórum para HSMs. En esta sección se describe cómo se crean, firman y autentican los comandos basados en quórum. Estas reglas son bastante simples. Por ejemplo, el comando Foo requiere dos miembros del rol Bar para que lo autentique. Hay tres pasos en la creación y verificación de un comando basado en quórum. El primer paso es la creación inicial del comando; el segundo es el envío a operadores adicionales para firmar y el tercero es la verificación y ejecución.

Con el fin de presentar los conceptos, suponga que existe un conjunto auténtico de claves públicas y roles del operador $\{QOS_s\}$ y un conjunto de reglas de quórum $QR = \{Command_i, Rule_{\{i, t\}}\}$ donde

cada regla es un conjunto de roles y un número mínimo $N \{Role_t, N_t\}$. A fin de que un comando satisfaga la regla de quórum, el conjunto de datos del comando debe estar firmado por un conjunto de operadores enumerados en $\{QOS_s\}$ de manera que cumplan con una de las reglas enumeradas para ese comando. Como se mencionó anteriormente, el conjunto de reglas de quórum y operadores se almacenan en el estado de dominio y el token de dominio exportado.

En la práctica, un firmante inicial firma el comando $Sig_1 = \text{Sign}(dO_{p1}, \text{Command})$. Un segundo operador también firma el comando $Sig_2 = \text{Sign}(dO_{p2}, \text{Command})$. El mensaje con doble firma se envía a un HSM para su ejecución. El HSM realiza las siguientes tareas:

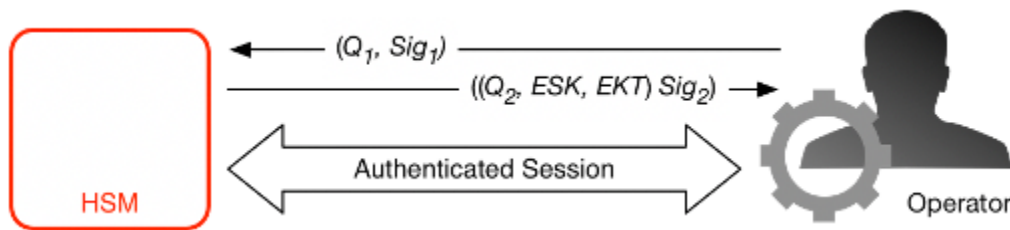
1. Para cada firma, extrae la clave pública del firmante del estado del dominio y verifica la firma en el comando.
2. Comprueba que el conjunto de firmantes cumpla una regla para el comando.

Sesiones autenticadas

Sus operaciones clave se ejecutan entre los AWS KMS hosts externos y el. HSMs Estos comandos se refieren a la creación y la utilización de claves criptográficas y a la generación segura de números aleatorios. Los comandos se ejecutan a través de un canal autenticado por sesión entre los hosts del servicio y el. HSMs Además de la necesidad de autenticidad, estas sesiones requieren confidencialidad. Los comandos que se ejecutan en estas sesiones incluyen la devolución de claves de datos de texto sin cifrar y mensajería descifrada destinados a usted. Para garantizar que estas sesiones no se puedan subvertir mediante man-in-the-middle ataques, las sesiones se autentican.

Este protocolo realiza un acuerdo de clave de ECDHE mutuamente autenticado entre el HSM y el anfitrión de servicio. El anfitrión de servicio inicia el intercambio y el HSM lo completa. El HSM también devuelve una session key (SK, clave de sesión) cifrada por la clave negociada y un token de clave exportado que contiene la clave de sesión. El token de clave exportado contiene un periodo de validez, después del cual el anfitrión de servicio debe renegociar una clave de sesión.

Un host de servicio es miembro del dominio y tiene un par de claves de firma de identidad ($DHos_i$, $QHOS_i$) y una copia auténtica de las claves públicas de «identidad HSMs». Utiliza el conjunto de claves de firma de identidad para negociar de forma segura una clave de sesión que se puede utilizar entre el anfitrión de servicio y cualquier HSM del dominio. Los tokens de clave exportados tienen un periodo de validación asociado a ellos, después del cual se debe negociar una nueva clave.



El proceso comienza con el reconocimiento del anfitrión de servicio de que requiere una clave de sesión para enviar y recibir flujos de comunicación sensibles entre sí y un miembro del HSM del dominio.

1. Un anfitrión de servicio genera un par de claves efímeras de ECDH (d_1 , Q_1) y lo firma con la clave de identidad $Sig_1 = \text{Sign}(\text{dOS}, Q_1)$.
2. El HSM verifica la firma en la clave pública recibida mediante el token de dominio actual y crea un par de claves efímeras de ECDH (d_2 , Q_2). A continuación, completa la información de ECDH-key-exchange acuerdo con la [recomendación para sistemas de establecimiento de claves por pares que utilicen criptografía de logaritmos discretos para formar una clave AES-GCM](#) negociada de 256 bits. El HSM genera una nueva clave de sesión AES-GCM de 256 bits. Cifra la clave de sesión con la clave negociada para formar la encrypted session key (ESK, clave de sesión cifrada). También cifra la clave de sesión bajo la clave de dominio como un token de clave exportado (EKT). Finalmente, firma un valor de retorno con el par de claves de identidad $Sig_2 = \text{Sign}(\text{dHSK}, [Q_2, \text{ESK}, \text{EKT}])$.
3. El anfitrión de servicio verifica la firma en las claves recibidas mediante el token de dominio actual. Luego, el anfitrión de servicio completa el intercambio de claves de ECDH según la [Recomendación para esquemas de establecimiento de claves por pares que utilizan criptografía de logaritmo discreta](#). A continuación, descifra la ESK a fin de obtener la clave de sesión (SK).

Durante el periodo de validación del EKT, el anfitrión de servicio puede utilizar la clave de sesión negociada SK para enviar comandos cifrados sobre el HSM. Todos los service-host-initiated comandos de esta sesión autenticada incluyen el EKT. El HSM responde con la utilización de la misma clave de sesión negociada SK.

Proceso de replicación de claves multirregión

AWS KMS utiliza un mecanismo de replicación entre regiones para copiar el material clave de una clave de KMS de un HSM Región de AWS a un HSM de otro. Región de AWS Para que este mecanismo funcione, la clave KMS que se está replicando debe ser una clave multirregional

con origen. **AWS_KMS** En el caso de las claves multirregionales con material clave importado (**EXTERNAL**origen), usted es responsable de replicar el material clave importando el mismo material clave en cada réplica de forma individual.

Al replicar una clave KMS de una región a otra, las HSMs regiones no pueden comunicarse directamente porque están en redes aisladas. En su lugar, los mensajes intercambiados durante la replicación entre regiones los entrega un servicio proxy. Durante la replicación entre regiones, todos los mensajes generados por un AWS KMS HSM se firman criptográficamente con una clave de firma de replicación. Las claves de firma de replicación (RSKs) son claves ECDSA en la curva P-384 del NIST. Cada región posee al menos un RSK y el componente público de cada RSK se comparte con todas las demás regiones de la misma partición. **AWS**

El proceso de replicación entre regiones para copiar material de clave de la región A a la región B funciona como se explica a continuación:

1. El HSM de la región B genera una clave ECDH efímera en la curva NIST P-384, Clave B del acuerdo de replicación (RAKB). El componente público de RAKB se envía a un HSM de la región A por el servicio proxy.
2. El HSM de la región A recibe el componente público de RAKB y, a continuación, genera otra clave ECDH efímera en la curva NIST P-384, Clave A del acuerdo de replicación (RAKA). El HSM ejecuta el esquema de establecimiento de claves ECDH en RAKA y el componente público de RAKB, y obtiene una clave simétrica de la salida, la Clave de encapsulamiento de replicación (RWK). La RWK se utiliza para cifrar el material de clave de la clave KMS multirregión que se está replicando.
3. El componente público de RAKA y el material de clave cifrado con la RWK se envían al HSM de la región B a través del servicio proxy.
4. El HSM de la región B recibe el componente público de RAKA y el material de clave cifrado mediante la RWK. El HSM se obtiene por la RWK ejecutando el esquema de establecimiento de claves ECDH en RAKB y el componente público de RAKA.
5. El HSM de la región B utiliza la RWK para descifrar el material de clave de la región A.

Protección de durabilidad

La durabilidad adicional del servicio para las claves generadas por el servicio se consigue mediante el uso del almacenamiento fuera de línea HSMs, el almacenamiento múltiple y no volátil de los tokens de dominio exportados y el almacenamiento redundante de claves KMS cifradas. Los que HSMs están fuera de línea son miembros de los dominios existentes. Con la excepción de no estar

en línea y participar en las operaciones normales del dominio, las personas sin conexión HSMs aparecen en el estado del dominio del mismo modo que los miembros del HSM existentes.

El diseño de durabilidad tiene por objeto proteger todas las claves KMS de una región en caso de que se AWS produzca una pérdida masiva de las claves KMS en línea HSMs o del conjunto de claves KMS almacenadas en nuestro sistema de almacenamiento principal. AWS KMS keys con material de clave importado, no están incluidas en las protecciones de durabilidad que ofrecen otras claves KMS. En caso de que se produzca un error en toda la región AWS KMS, es posible que sea necesario volver a importar el material clave importado a una clave KMS.

Los objetos sin conexión HSMs y las credenciales para acceder a ellos se guardan en cajas fuertes dentro de salas seguras supervisadas en varias ubicaciones geográficas independientes. Cada caja fuerte requiere al menos un oficial de AWS seguridad y un AWS KMS operador, de dos equipos independientes AWS, para obtener estos materiales. El uso de estos materiales se rige por una política interna que exige la presencia de un quórum de AWS KMS operadores.

Historial de documentos

En este tema se describen actualizaciones importantes en la Guía para desarrolladores de AWS Key Management Service .

Temas

- [Actualizaciones recientes](#)
- [Actualizaciones anteriores](#)

Actualizaciones recientes

En la siguiente tabla se describen cambios importantes en esta documentación desde enero de 2018. Además de los cambios importantes que se indican a continuación, también actualizamos la documentación con frecuencia para mejorar las descripciones y los ejemplos y para dar cuenta de los comentarios que nos envía. Si desea recibir notificaciones sobre cambios importantes, suscríbase a la fuente RSS.

Es posible que tenga que desplazarse horizontal o verticalmente para ver todos los datos de esta tabla.

Cambio	Descripción	Fecha
Subsidios principales de servicio	AWS KMS ahora apoya la creación de subvenciones para los directores AWS de servicio. Puede utilizar los <code>RetiringServicePrincipal</code> parámetros nuevos <code>GranteeServicePrincipal</code> y de la <code>CreateGrant</code> operación para especificar a un director de servicio como beneficiario o director saliente de una subvención. Esta versión también introduce la restricción de	6 de marzo de 2026

SourceArn concesión, que restringe la concesión de permisos a las solicitudes realizadas en nombre de un recurso específico. AWS Hay tres nuevas claves de condición disponibles para CreateGrant las solicitudes: kms:GranteeServicePrincipal kms:RetiringServicePrincipal , y. kms:GrantConstraintSourceArn Para obtener más información, consulte [Subvenciones en AWS KMS](#).

[Actualización de funciones](#)

AWS KMS admite la rotación bajo demanda de claves multirregionales y de cifrado simétrico con material clave importado.

26 de noviembre de 2025

[Actualización de funciones](#)

AWS KMS admite un nuevo keyExchange campo en la sección de eventos. tlsDetails CloudTrail

24 de noviembre de 2025

[Actualización de funciones](#)

AWS KMS es compatible con el nuevo KeySpec ECC_NIST_EDWARDS25519.

7 de noviembre de 2025

[Dual-stack apoyo](#)

AWS KMS admite doble pila.

18 de junio de 2025

[Actualización de funciones](#)

Añade compatibilidad con las firmas criptográficas poscuánticas del algoritmo de firma Module-Lattice digital (ML-DSA).

13 de junio de 2025

Rotación de claves importada	Puede realizar la rotación bajo demanda de las claves de KMS de cifrado simétrico con material de clave importado (origen EXTERNAL).	5 de junio de 2025
Actualización de funciones	Se agregó compatibilidad con las claves de KMS de varias regiones de China.	21 de noviembre de 2024
AWS actualización de política gestionada	Se actualizó la función <code>AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy</code> vinculada al servicio añadiendo un identificador de sentencia (<code>Sid</code>) a la política gestionada con la versión v2 de la política.	21 de noviembre de 2024
Cambio de cuota	Se ha aumentado la tasa de solicitudes ImportKeyMaterial y DeleteImportedKeyMaterial solicitudes predeterminada.	23 de julio de 2024
Cambio de cuota	Se ha aumentado la tasa de solicitudes de operaciones criptográficas predeterminada para las claves KMS de cifrado simétricas, las claves KMS de RSA y las claves KMS de ECC y SM2.	8 de julio de 2024

Nueva característica	Se agregó un nuevo KeyUsage tipo KEY_AGREEMENT para las claves KMS de curva NIST-recommended elíptica (ECC) y SM2 (solo para regiones de China) y se agregó soporte para derivar secretos compartidos.	13 de junio de 2024
Actualizaciones de la rotación de claves	Se ha agregado soporte a los períodos de rotación personalizados para las rotaciones de claves automáticas, las rotaciones de claves bajo demanda y la visibilidad sobre las rotaciones de materiales de claves.	12 de abril de 2024
Actualizaciones a políticas administradas	Se agregaron nuevos permisos AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy que permiten AWS KMS monitorear los cambios en la VPC que contiene su AWS CloudHSM clúster, de modo que AWS KMS pueden proporcionar mensajes de error claros en caso de fallas.	10 de noviembre de 2023
Actualización de funciones	Se ha agregado compatibilidad con el parámetro de API DryRun.	5 de julio de 2023

Actualización de funciones	Se agregó soporte para importar material clave para todos los tipos de AWS KMS claves, excepto para los almacenes de claves personalizados.	5 de junio de 2023
Actualización de funciones	Actualizaciones de las AWS KMS API para Nitro Enclaves	10 de marzo de 2023
Actualización de funciones	El algoritmo de RSAES_PKCS1_V1_5 empaquetado está obsoleto. AWS KMS pondrá fin a todo el soporte a más RSAES_PKCS1_V1_5 tardar el 1 de octubre de 2023, de conformidad con las directrices sobre la gestión de claves criptográficas del Instituto Nacional de Estándares y Tecnología (NIST). Le recomendamos que comience a utilizar un algoritmo de encapsulamiento diferente de inmediato.	28 de febrero de 2023
Actualización de funciones	Se ha añadido soporte para almacenes de claves externos, una función que te permite proteger tus AWS recursos mediante claves criptográficas externas. AWS	29 de noviembre de 2022
Cambio de cuota	Se aumentó la cuota AWS KMS keys de recursos a 100 000 claves de KMS en cada cuenta y región.	8 de julio de 2022

Actualización de funciones	Se agregó soporte para claves HMAC KMS en más Regiones de AWS	8 de julio de 2022
Nuevo tema	Se agregó el AWS Key Management Service tema Resiliencia al capítulo de seguridad de la Guía para AWS KMS desarrolladores.	14 de junio de 2022
Nueva característica	Se agregó soporte para AWS KMS claves y operaciones de API que generan y verifican códigos HMAC.	19 de abril de 2022
Cambio de documentación	Sustituir el término clave maestra del cliente (CMK) por AWS KMS key y clave KMS.	30 de agosto de 2021
Nueva característica	Se ha agregado compatibilidad para claves de varias regiones , un conjunto de claves KMS interoperables en diferentes regiones que tienen el mismo ID de clave y el mismo material de claves. Puede utilizar claves de varias regiones para cifrar datos en una región y descifrar datos en una región diferente.	8 de junio de 2021
Nueva característica	Se ha agregado compatibilidad con control de acceso basado en el atributo (ABAC). Puede usar etiquetas y alias para controlar el acceso a su. AWS KMS keys	17 de diciembre de 2020

Nueva característica	Se agregó compatibilidad para las políticas de punto de conexión de VPC.	9 de julio de 2020
Nuevo contenido	Explica las propiedades de seguridad de. AWS KMS	18 de junio de 2020
Nueva característica	Se agregó compatibilidad con claves de datos asimétricas AWS KMS keys y asimétricas.	25 de noviembre de 2019
Función actualizada	Puede ver la política clave de Claves administradas por AWS en la AWS KMS consola. Anteriormente, este rol se limitaba a las claves administradas por el cliente.	15 de noviembre de 2019
Nueva característica	Explica cómo utilizar algoritmos de intercambio híbrido postcuántico de claves en TLS para sus llamadas a AWS KMS.	4 de noviembre de 2019
Cambio de cuota	Se han aumentado las cuotas de recursos para algunas API que administran las claves KMS.	18 de septiembre de 2019
Cambio de cuota	Se han cambiado las cuotas de recursos en las claves KMS, los alias y las concesiones de cada clave KMS.	27 de marzo de 2019

Cambio de cuota	Se ha cambiado la cuota de solicitud por segundo para operaciones criptográficas que utilizan AWS KMS keys en un almacén de claves personalizadas.	7 de marzo de 2019
Nueva característica	Explica cómo crear y administrar almacenes de claves AWS KMS personalizados . Cada almacén de claves está respaldado por un AWS CloudHSM clúster que usted posee y controla.	26 de noviembre de 2018
Nueva consola	Explica cómo utilizar la nueva AWS KMS consola, que es independiente de la consola de IAM. La consola original y las instrucciones de uso estarán disponibles durante un breve periodo de tiempo para darle tiempo para familiarizarse con la nueva consola.	7 de noviembre de 2018
Cambio de cuota	Se modificó la cuota de solicitudes compartidas para su uso de AWS KMS keys.	21 de agosto de 2018
Nuevo contenido	Explica cómo se AWS Secrets Manager utilizan AWS KMS las claves para cifrar el valor secreto de un secreto.	13 de julio de 2018

[Nuevo contenido](#)

Explica cómo DynamoDB admite su opción de AWS KMS AWS KMS keys cifrado del lado del servidor.

23 de mayo de 2018

[Nueva característica](#)

Explica cómo [usar un punto de conexión privado en tu VPC](#) para conectarte directamente a AWS KMS, en lugar de hacerlo a través de Internet.

22 de enero de 2018

Actualizaciones anteriores

En la siguiente tabla se describen los cambios importantes introducidos en la Guía para AWS Key Management Service desarrolladores antes de 2018.

Es posible que tenga que desplazarse horizontal o verticalmente para ver todos los datos de esta tabla.

Cambio	Descripción	Fecha
Contenido nuevo	Se ha agregado documentación sobre Etiquetas en AWS KMS .	15 de febrero de 2017
Contenido nuevo	Se ha agregado documentación sobre Supervisar AWS KMS keys y Supervise las claves de KMS con Amazon CloudWatch .	31 de agosto de 2016
Contenido nuevo	Se ha agregado documentación sobre Material de claves importado .	11 de agosto de 2016
Contenido nuevo	Se ha agregado la siguiente documentación: Políticas de	5 de julio de 2016

Cambio	Descripción	Fecha
	IAM , Referencia de permisos , y Claves de condición .	
Actualización	Se han actualizado partes de la documentación en el capítulo Permisos y acceso a claves KMS .	5 de julio de 2016
Actualización	Se ha actualizado la página Cuotas para reflejar las nuevas cuotas predeterminadas.	31 de mayo de 2016
Actualización	Se ha actualizado la página Cuotas para reflejar las nuevas cuotas predeterminadas y se ha actualizado la documentación de token de concesión para mejorar la claridad y la precisión.	11 de abril de 2016
Contenido nuevo	Se ha agregado documentación sobre Conceder permiso a varias entidades principales de IAM para acceder a una clave KMS y Usar la condición de dirección IP .	17 de febrero de 2016
Actualización	Se han actualizado las páginas Políticas clave en AWS KMS y Cambio de una política de claves para mejorar la claridad y la precisión.	17 de febrero de 2016

Cambio	Descripción	Fecha
Actualización	Se han actualizado las páginas de temas sobre administración de KMS para mejorar la claridad.	5 de enero de 2016
Contenido nuevo	Se agregó documentación sobre CloudTrail.	18 de noviembre de 2015
Contenido nuevo	Se han agregado instrucciones para Cambio de una política de claves .	18 de noviembre de 2015
Actualización	Se actualizó la documentación sobre cómo utiliza Amazon Relational Database Service AWS KMS.	18 de noviembre de 2015
Contenido nuevo	Se ha añadido documentación sobre Amazon WorkSpaces.	6 de noviembre de 2015
Actualización	Se ha actualizado la página Políticas clave en AWS KMS para mejorar la claridad.	22 de octubre de 2015
Contenido nuevo	Se ha agregado documentación sobre Eliminar un AWS KMS key , incluida la documentación complementaria sobre Crear una alarma y Determinación del uso anterior de una clave KMS .	15 de octubre de 2015
Contenido nuevo	Se ha agregado documentación sobre Determinar el acceso a AWS KMS keys .	15 de octubre de 2015

Cambio	Descripción	Fecha
Contenido nuevo	Se ha agregado documentación sobre Estados clave de AWS KMS claves .	15 de octubre de 2015
Contenido nuevo	Documentación agregada sobre Amazon Simple Email Service.	1 de octubre de 2015
Actualización	Se ha actualizado la página Cuotas para explicar las nuevas cuotas de solicitud.	31 de agosto de 2015
Contenido nuevo	Se agregó información sobre los cargos por su uso AWS KMS. Consulte Precios de AWS KMS .	14 de agosto de 2015
Contenido nuevo	Se agregaron cuotas de solicitudes a AWS KMS Cuotas .	11 de junio de 2015
Contenido nuevo	Se ha agregado un nuevo ejemplo de código de Java que demuestra el uso de la operación UpdateAlias .	1 de junio de 2015
Actualización	Se ha movido la tabla de regiones de AWS Key Management Service a la Referencia general de AWS.	29 de mayo de 2015
Contenido nuevo	Se ha agregado documentación sobre Cómo utiliza Amazon EMR AWS KMS .	28 de enero de 2015
Contenido nuevo	Se ha añadido documentación sobre Amazon WorkMail.	28 de enero de 2015

Cambio	Descripción	Fecha
Contenido nuevo	Se agregó documentación sobre cómo utiliza Amazon Relational Database Service AWS KMS.	6 de enero de 2015
Contenido nuevo	Documentación agregada sobre Amazon Elastic Transcoder.	24 de noviembre de 2014
Nueva guía	Se ha presentado la Guía para desarrolladores de AWS Key Management Service .	12 de noviembre de 2014

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.