Guía del usuario

AWS CloudHSM

AWS CloudHSM: Guía del usuario

Table of Contents

¿Qué es AWS CloudHSM? 1
Casos de uso 2
Funcionamiento 4
Clústeres 5
Usuarios de HSM 5
Claves de HSM 6
SDK del cliente 7
Copias de seguridad 7
Regiones 9
Precios 9
Introducción 10
Creación de administradores de IAM 10
Creación de un grupo de usuarios de IAM y de administradores 11
Creación de una VPC 13
Crear un clúster 14
Revisión del grupo de seguridad del clúster 16
Lanzamiento de un cliente EC2 17
Configuración de los grupos de seguridad de la instancia EC2 20
Modificar el grupo de seguridad predeterminado 20
Conecte la instancia de Amazon EC2 al clúster AWS CloudHSM 21
Creación de un HSM 22
Verificar la identidad del HSM (opcional) 23
Información general 24
Obtención de los certificados del HSM8 26
Obtención de los certificados raíz 29
Verificación de las cadenas de certificados 29
Extracción y comparación de las claves públicas 30
Inicio del clúster 31
Obtención del CSR del clúster 32
Firmar la CSR 34
Inicio del clúster 36
Instalación de la CLI de CloudHSM 38
Instale las herramientas de línea de comandos AWS CloudHSM 38
Activación del clúster 42

Reconfigurar SSL (opcional) ... 45
Cree una clave, una CSR y, a continuación, firme la CSR. ... 45
Habilite el SSL personalizado para AWS CloudHSM ... 46
Creación de una aplicación ... 51
Prácticas recomendadas ... 53
Administración de clústeres ... 53
Escale su clúster para gestionar los picos de tráfico. ... 53
Diseñe su clúster para conseguir una alta disponibilidad. ... 53
Tenga, al menos, tres HSM para garantizar la durabilidad de las claves recién generadas. ... 54
Acceso seguro a su clúster ... 54
Reduzca los costos escalando en función de sus necesidades. ... 54
Administración de usuarios de HMS ... 55
Proteja las credenciales de sus usuarios de HSM. ... 55
Tenga, al menos, dos administradores para evitar bloqueos. ... 55
Habilite el cuórum para todas las operaciones de gestión de usuarios. ... 56
Cree varios usuarios de criptografía con permisos limitados. ... 56
Administración de claves HSM ... 56
Elija el tipo de clave correcto. ... 56
Gestione los límites de almacenamiento de claves. ... 57
Gestionar y proteger el empaquetado de claves. ... 57
Integración de aplicaciones ... 58
Inicie su SDK de cliente. ... 58
Autentíquese para realizar operaciones. ... 58
Gestione eficazmente las claves de su aplicación. ... 59
Emplee subprocesamiento múltiple. ... 60
Gestione los errores de limitación. ... 60
Integre los reintentos en las operaciones del clúster. ... 61
Implemente estrategias de recuperación de desastres. ... 61
Supervisión ... 62
Supervisión de registros de clientes ... 62
Monitoreo de registros de auditoría ... 62
Supervise AWS CloudTrail ... 63
Supervisa CloudWatch las métricas de Amazon ... 63
Administración de clústeres de ... 64
Arquitectura de clúster ... 64
Sincronización de clúster ... 65

Alta disponibilidad y balanceo de carga del clúster ... 66
Conexión al clúster ... 67
Colocación del certificado de emisión en cada instancia de EC2 ... 67
Especifique la ubicación del certificado de emisión. ... 68
Proceso de arranque del SDK de cliente ... 70
Cómo agregar o eliminar HSM ... 74
Agregar un HSM ... 74
Eliminación de un HSM ... 76
Eliminación de un clúster ... 77
Creación de clústeres a partir de las copias de seguridad ... 78
Cree clústeres a partir de copias de seguridad (consola) ... 79
Crear clústeres a partir de copias de seguridad (CLI) ... 80
Cree clústeres a partir de copias de seguridad (AWS CloudHSM API) ... 81
Administración de copias de seguridad ... 82
Trabajo con copias de seguridad ... 82
Eliminación de claves caducadas o usuarios inactivos ... 83
Consideración de recuperación de desastres ... 83
Eliminación y restauración de copias de seguridad ... 83
Eliminación y restauración de copias de seguridad (consola) ... 83
Eliminar y restaurar copias de seguridad (CLI) ... 84
Eliminar y restaurar las copias de seguridad (AWS CloudHSM API) ... 86
Configuración de retención de copias de seguridad ... 86
Descripción de la política de retención de copias de seguridad ... 86
Configurar la retención de copias de seguridad (consola) ... 87
Configurar la retención de copias de seguridad (CLI) ... 88
Configure la retención de copias de seguridad (AWS CloudHSM API) ... 90
Cómo clonar copias de seguridad entre regiones ... 90
Clonar copias de seguridad en diferentes regiones (consola) ... 91
Copiar copias de seguridad a diferentes regiones (CLI) ... 91
Copiar las copias de seguridad a diferentes regiones (AWS CloudHSM API) ... 92
Etiquetado de recursos ... 93
Adición o actualización de etiquetas ... 93
Enumeración de etiquetas ... 95
Eliminación de etiquetas ... 95
Administración de claves y de usuarios de HSM ... 97
Administración de usuarios de HSM ... 97

Utilización de la CLI de CloudHSM 97
Uso de CMU 148
Administración de claves 195
Sincronización y durabilidad de claves 195
Encapsulamiento de claves AES 204
Claves de confianza 208
Administración de claves con la CLI de CloudHSM 213
Gestión de claves con la KMU y la CMU 238
Administración de clústeres clonados 246
Obtención de una dirección IP para un HSM 248
Temas relacionados de 248
Herramientas de la línea de comandos 250
Mäs información sobre las herramientas de línea de comando 250
Herramienta de configuración 251
Herramienta de configuración más reciente 252
Herramienta de configuración anterior 279
CLI de CloudHSM 288
Plataformas admitidas 289
Introducción 290
Modos interactivo y de comando único 296
Atributos de clave 298
Migre de CMU y KMU a CloudHSM CLI 305
Configuraciones avanzadas 306
Referencia 313
Utilidad de administración de CloudHSM 517
Plataformas admitidas 517
Introducción 518
Instalar el cliente (Linux) 523
Instalación del cliente (Windows) 526
Referencia 527
Utilidad de administración de claves 590
Introducción 591
Instalar el cliente (Linux) 595
Instalación del cliente (Windows) 598
Referencia 599
SDK del cliente 727

Plataformas admitidas 727
Compatibilidad de Linux con SDK 5 de cliente 728
Compatibilidad de Windows con SDK 5 de cliente 729
Compatibilidad sin servidor para SDK 5 de cliente 729
Compatibilidad con componentes 729
Ventajas del último SDK 729
Migración a la versión más reciente del SDK 730
Biblioteca PKCS #11 731
Instalación de la biblioteca PKCS #11 732
Autenticación en la biblioteca PKCS #11 736
Tipos de clave 737
Mecanismos 737
Operaciones de la API 743
Atributos de clave 745
Ejemplos de código 771
Migre al SDK más reciente 772
Configuraciones avanzadas 775
Motor dinámico de OpenSSL 782
Instalación del motor dinámico de OpenSSL 783
Tipos de clave 787
Mecanismos 787
Migre al SDK más reciente 788
Configuraciones avanzadas 790
Proveedor de JCE 791
Instalación del proveedor de JCE 792
Tipos de clave 798
Mecanismos 799
Atributos de clave 808
Ejemplos de código 818
Javadocs 819
ClouDHSM KeyStore 819
Migre al SDK más reciente 823
Configuraciones avanzadas 834
Proveedores de KSP y CNG 842
Verificación de la instalación del proveedor 843
Requisitos previos 845

Asociación de una clave con un certificado 847
Ejemplo de código 849
SDK de cliente anterior 855
Compruebe su versión de SDK de cliente. 856
Comparación de componentes de SDK de cliente 857
Plataformas admitidas 858
Actualización de SDK 3 de cliente 861
Biblioteca PKCS #11 870
Motor dinámico de OpenSSL 914
Proveedor de JCE 917
Integración de aplicaciones de terceros 950
Descarga de SSL/TLS 950
Funcionamiento 951
Descarga de SSL/TLS en Linux 953
Descarga de SSL/TLS en Windows 1027
Agregar un equilibrador de carga (opcional) 1039
Entidad de certificación de Windows Server 1047
Requisitos previos 1047
Crear entidad de certificación de Windows Server 1049
Firmar una CSR 1051
Cifrado de Oracle Database 1052
Configuración de requisitos previos 1054
Configuración de la base de datos 1055
Microsoft SignTool 1058
Microsoft SignTool con AWS CloudHSM el paso 1: configurar los requisitos previos 1059
Microsoft SignTool con el AWS CloudHSM paso 2: crea un certificado de firma 1060
Microsoft SignTool con AWS CloudHSM el paso 3: firmar un archivo 1062
Java Keytool y Jarsigner 1063
Use SDK 5 de cliente para la integración con Java Keytool y Jarsigner. 1063
Use SDK 3 de cliente para la integración con Java Keytool y Jarsigner 1075
Otras integraciones de proveedores externos 1091
Supervisión 1093
Registros de SDK de cliente 1093
Registro de SDK 5 de cliente 1094
Registro de SDK 3 de cliente 1095
AWS CloudTrail 1097

AWS CloudHSM información en CloudTrail 1097
Descripción AWS CloudHSM de las entradas de los archivos de registro 1098
Registros de auditoría 1100
Cómo funciona el proceso de registro 1100
Visualización de registros 1101
Interpretación de registros 1104
Referencia de registro 1120
CloudWatch métricas 1123
Rendimiento 1125
Datos de rendimiento 1125
..... 1125
Limitación de HSM 1126
Seguridad 1127
Protección de datos 1128
Cifrado en reposo 1129
Cifrado en tránsito 1129
nd-to-end Cifrado electrónico 1129
Copias de seguridad de los clústeres 1131
Administración de identidades y accesos 1132
Concesión de permisos mediante políticas de IAM 1132
Acciones de la API para AWS CloudHSM 1133
Claves de condición para AWS CloudHSM 1134
Políticas administradas por AWS predefinidas para AWS CloudHSM 1134
Políticas gestionadas por el cliente para AWS CloudHSM 1135
Roles vinculados al servicio 1138
Conformidad 1140
Preguntas frecuentes sobre PCI-PIN 1141
Notificaciones de obsolescencia 1143
Resiliencia 1144
Seguridad de la infraestructura 1144
Aislamiento de red 1145
Autorización de usuarios 1145
Puntos de conexión de VPC (AWS PrivateLink) 1145
Consideraciones sobre los puntos AWS CloudHSM finales de VPC 1145
Creación de un punto de conexión de VPC de interfaz para AWS CloudHSM 1146
Crear una política de puntos de conexión de VPC para AWS CloudHSM 1146

Administración de actualizaciones 1147
Resolución de problemas 1148
Problemas conocidos 1148
Problemas conocidos para todas las instancias de HSM 1149
Problemas conocidos de la biblioteca PKCS #11 1153
Problemas conocidos para el SDK de JCE 1159
Problemas conocidos de OpenSSL Dynamic Engine 1164
Problemas conocidos para instancias de Amazon EC2 que ejecutan Amazon Linux 2 1167
Problemas conocidos para integrar aplicaciones de terceros 1167
Fallos de sincronización de clave en SDK 3 de cliente 1168
El SDK 3 de cliente comprueba el rendimiento. 1169
Recomendaciones de prueba 1170
Opciones configurables para la herramienta pkpspeed 1171
Pruebas que se pueden ejecutar con la herramienta pkpspeed 1171
Ejemplos 1172
El usuario de SDK 5 de cliente contiene valores inconsistentes. 1175
Se detectó un error durante la comprobación de disponibilidad de las claves. 1182
Extracción de claves con JCE 1183
GetEncoded o getPrivateExponent GetS devuelve null 1183
GetEncoded o getPrivateExponent GetS devuelven bytes clave fuera del HSM 1183
Limitación de HSM 1184
Resolución 1185
Mantener sincronizados los usuarios de HSM 1185
Conexión perdida 1186
Faltan registros de AWS CloudHSM auditoría CloudWatch 1189
Encapsulamientos de claves AES no compatibles 1189
Determine si el código genera claves encapsuladas irrecuperables. 1189
Acciones que debe adoptar si su código genera claves encapsuladas irrecuperables 1191
Solución de errores de creación de clústeres 1192
Agregar el permiso que falta. 1193
Crear el rol vinculado a un servicio manualmente. 1193
Uso de un usuario no federado 1193
Recuperación de registros de configuración de los clientes 1194
Herramienta de soporte para SDK 5 de cliente 1194
Herramienta de soporte para el SDK 3 de cliente 1196
Cuotas 1198

Recursos del sistema 1199
Descargas 1201
Descargas 1201
Versión más reciente 1201
Versión 5 del SDK de cliente: versión 5.12.0 1201
Versiones anteriores del SDK de cliente 1207
Versiones obsoletas 1225
Versiones obsoletas del Client SDK 5 1225
Versiones obsoletas del Client SDK 3 1240
Lanzamientos E nd-of-life 1249
Historial de documentos 1250
Actualizaciones recientes 1250
Actualizaciones anteriores 1256
..... mcclviii

¿Qué es AWS CloudHSM?

AWS CloudHSM combina las ventajas de la AWS nube con la seguridad de los módulos de seguridad de hardware (HSM). Un módulo de seguridad de hardware (HSM) es un dispositivo informático que procesa las operaciones criptográficas y proporciona almacenamiento seguro de las claves criptográficas. Con ello AWS CloudHSM, tiene el control total de los HSM de alta disponibilidad que se encuentran en la nube de AWS, tiene acceso de baja latencia y una base de confianza segura que automatiza la administración de los HSM (incluidas las copias de seguridad, el aprovisionamiento, la configuración y el mantenimiento).

AWS CloudHSM ofrece a los clientes una variedad de beneficios:

Los HSM están validados por la norma FIPS 140-2 de nivel 3.

> AWS CloudHSM utiliza HSM de uso general que cumplen con los estándares, son de un solo propietario y están validados por la norma FIPS 140-2 de nivel 3. Ofrecen más flexibilidad en comparación con los servicios de AWS totalmente gestionados, que tienen algoritmos y longitudes de clave predeterminados para su aplicación.

El cifrado E2E no es visible para AWS.

> Como su plano de datos está cifrado end-to-end (E2E) y AWS no lo ve, usted controla su propia administración de usuarios (fuera de las funciones de IAM). A cambio de este control, usted tiene más responsabilidad que si usara un servicio de AWS gestionado.

Control total de sus claves, algoritmos y desarrollo de aplicaciones.

> AWS CloudHSM le da el control total de los algoritmos y las claves que utiliza. Usted puede generar, almacenar, importar, exportar y administrar claves criptográficas, incluidas las claves de sesión, claves token, claves simétricas y pares de claves asimétricas. Además, AWS CloudHSM los SDK le proporcionan un control total sobre el desarrollo de las aplicaciones, el lenguaje de las aplicaciones, los subprocesos y la ubicación física de las aplicaciones.

Migre sus cargas de trabajo criptográficas a la nube.

> Los clientes que migren una infraestructura de clave pública que utilice los estándares de criptografía de clave pública #11 (PKCS #11), la extensión criptográfica de Java (JCE), la API de criptografía: próxima generación (CNG) o el proveedor de almacenamiento de claves (KSP) pueden migrar a ella con menos cambios en su aplicación. AWS CloudHSM

Acceso a clústeres FIPS y no FIPS

Para obtener más información sobre lo que puede hacer con él, consulte los siguientes temas. AWS CloudHSM Cuando esté listo para empezar AWS CloudHSM, consulteIntroducción.

> **Note**
>
> Si busca un servicio administrado para la creación y el control de las claves de cifrado, pero no desea o no necesita administrar su propio HSM, considere la posibilidad de usar AWS Key Management Service.
> Si busca un servicio elástico que gestione HSM de pago y claves de aplicaciones de procesamiento de pagos en la nube, puede usar AWS Payment Cryptography.

Contenido

- AWS CloudHSM casos de uso
- Cómo AWS CloudHSM funciona
- Precios

AWS CloudHSM casos de uso

AWS CloudHSM se puede utilizar para lograr una variedad de objetivos. El contenido de este tema proporciona una descripción general de lo que puede hacer con él AWS CloudHSM.

Cumplimiento de la normativa

Las empresas que necesiten ajustarse a los estándares de seguridad empresarial pueden utilizar AWS CloudHSM para gestionar las claves privadas que protegen los datos altamente confidenciales. Los HSM que proporciona AWS CloudHSM cuentan con la certificación FIPS 140-2 de nivel 3 y cumplen con la normativa PCI DSS. Además, AWS CloudHSM son compatibles con el PIN PCI y con el PCI-3DS. Para obtener más información, consulte Conformidad.

Cifrar y descifrar datos

Se utiliza AWS CloudHSM para gestionar las claves privadas que protegen los datos altamente confidenciales, el cifrado en tránsito y el cifrado en reposo. Además, AWS CloudHSM ofrece una integración compatible con los estándares con varios SDK criptográficos.

Firme y verifique documentos con claves públicas y privadas.

En criptografía, el uso de una clave privada para firmar un documento permite a los destinatarios utilizar una clave pública para verificar que usted (y no otra persona) ha enviado realmente el documento. Se utiliza AWS CloudHSM para crear pares de claves públicas y privadas asimétricas diseñadas específicamente para este propósito.

Autentique mensajes mediante HMAC y CMAC.

En criptografía, los códigos de autenticación de mensajes cifrados (CMAC) y los códigos de autenticación de mensajes basados en hash (HMAC) se usan para autenticar y garantizar la integridad de los mensajes enviados a través de redes no seguras. Con él AWS CloudHSM, puede crear y administrar de forma segura claves simétricas compatibles con HMAC y CMAC.

Aproveche los beneficios de y AWS CloudHSM AWS Key Management Service

Los clientes pueden combinar AWS CloudHSM y AWS KMSalmacenar el material clave en un entorno de un solo inquilino que cuente con la certificación FIPS 140-2 de nivel 3 y, al mismo tiempo, obtener las principales ventajas de administración, escalado e integración en la nube que ofrecen. AWS KMS Para obtener más información al respecto, consulte Almacenamiento de claves de AWS CloudHSM en la AWS Key Management Service Guía del desarrollador.

Descargar el procesamiento de SSL/TLS en los servidores web.

Para enviar datos de forma segura a través de Internet, los servidores web emplean pares de claves públicas/privadas y certificados de clave pública SSL/TLS para establecer sesiones HTTPS. Este proceso implica una gran cantidad de cálculos para los servidores web, pero puede reducir la carga computacional y, al mismo tiempo, brindar mayor seguridad al transferir parte de esta información a su clúster. AWS CloudHSM Para obtener información sobre cómo configurar la descarga de SSL/TLS con, consulte. AWS CloudHSMDescarga de SSL/TLS

Cifrado de datos transparente (TDE)

El cifrado de datos transparente (TDE) se usa para cifrar archivos de bases de datos. Con TDE, el software de base de datos cifra los datos antes de almacenarlos en el disco. Puede lograr una mayor seguridad almacenando la clave de cifrado maestra de TDE en los HSM de su AWS

CloudHSM. Para obtener información sobre la configuración de Oracle TDE con, consulte. AWS CloudHSMCifrado de Oracle Database

Gestionar las claves privadas de una autoridad de certificación (CA)

Una autoridad de certificación (CA) es una entidad de confianza que emite certificados digitales que vinculan una clave pública a una identidad (persona u organización). Para operar una CA, debe mantener la confianza con la protección de las claves privadas que firman los certificados emitidos por la CA. Puede almacenar dichas claves privadas en su AWS CloudHSM clúster y, a continuación, utilizar sus HSM para realizar operaciones de firma criptográfica.

Genere números aleatorios.

Generar números aleatorios para crear claves de cifrado es fundamental para la seguridad en línea. AWS CloudHSM se puede utilizar para generar números aleatorios de forma segura en los HSM que usted controla y que solo usted puede ver.

Cómo AWS CloudHSM funciona

En este tema se proporciona una descripción general de los conceptos básicos y la arquitectura que se utilizan para cifrar datos de forma segura y realizar operaciones criptográficas en los HSM. AWS CloudHSM opera en su propia Amazon Virtual Private Cloud (VPC). Antes de poder utilizarla AWS CloudHSM, primero debe crear un clúster, añadirle HSM, crear usuarios y claves y, a continuación, utilizar los SDK de cliente para integrar los HSM con la aplicación. Una vez hecho esto, utiliza los registros del SDK del cliente AWS CloudTrail, los registros de auditoría y Amazon CloudWatch para supervisar AWS CloudHSM.

Conozca AWS CloudHSM los conceptos básicos y cómo funcionan juntos para ayudar a proteger sus datos.

Temas

- AWS CloudHSM clústeres
- Usuarios de HSM
- Claves de HSM
- SDK de cliente
- AWS CloudHSM copias de seguridad en clúster

- Regiones

AWS CloudHSM clústeres

Hacer que los HSM individuales trabajen juntos en un clúster sincronizado, redundante y de alta disponibilidad puede resultar difícil, pero AWS CloudHSM supone todo el trabajo al proporcionar módulos de seguridad de hardware (HSM) en los clústeres. Un clúster es un conjunto de HSM individuales que se mantienen sincronizados. AWS CloudHSM Al realizar una tarea o una operación en un HSM en un clúster, el resto de los HSM de ese clúster se actualizan automáticamente. Para alcanzar sus objetivos de disponibilidad, durabilidad y escalabilidad, debe establecer la cantidad de HSM en su clúster en varias zonas de disponibilidad.

Puede crear un clúster que tenga de 1 a 28 HSM (el límite predeterminado es de 6 HSM por AWSAWS cuenta y región). Puede colocar los HSM en diferentes zonas de disponibilidad de una región. AWS Agregar más HSM a un clúster ofrece más desempeño. Distribuir clústeres en varias zonas de disponibilidad proporciona redundancia y alta disponibilidad.

Para obtener más información acerca de los clústeres , consulte Administrar AWS CloudHSM clústeres.

Para crear un clúster, consulte Introducción.

Usuarios de HSM

A diferencia de la mayoría de los AWS servicios y recursos, no utiliza usuarios AWS Identity and Access Management (de IAM) ni políticas de IAM para acceder a los recursos de su clúster. En su lugar, utiliza los usuarios de HSM directamente en los HSM de su clúster. AWS CloudHSM

Los usuarios de HSM son distintos de los usuarios de IAM. Los usuarios de IAM que dispongan de credenciales correctas pueden crear HSM interactuando con los recursos a través de la API de AWS. El cifrado E2E no es visible para AWS, por lo que debe usar las credenciales de usuario del HSM para autenticar las operaciones en el mismo, ya que las credenciales se gestionan directamente en el HSM. El HSM autentica a sus usuarios mediante credenciales que usted define y administra. Cada usuario del HSM tiene un tipo que determina las operaciones que puede realizar en el HSM. Cada HSM autentica a sus usuarios mediante las credenciales que usted ha definido en la CLI de CloudHSM.

Si usa la serie de versiones anteriores del SDK, tendrá que utilizar la utilidad de administración de CloudHSM (CMU).

Claves de HSM

AWS CloudHSM le permite generar, almacenar y administrar de forma segura sus claves de cifrado en los HSM de un solo inquilino que se encuentran en su clúster de AWS CloudHSM. Las claves pueden ser simétricas o asimétricas, pueden ser claves de sesión (claves efímeras) para sesiones únicas, claves token (claves persistentes) para uso a largo plazo, y pueden exportarse e importarse a AWS CloudHSM. Las claves también se pueden usar para completar tareas y funciones criptográficas comunes:

- Realice la firma de datos criptográficos y la verificación de firmas con algoritmos de cifrado simétricos y asimétricos.
- Trabajar con funciones hash para calcular compendios de mensajes y códigos de autenticación de mensajes basados en hash (HMAC).
- Encapsule y proteja otras claves.
- Accede a datos aleatorios criptográficamente seguros.

Además, AWS CloudHSM sigue algunos principios fundamentales para el uso y la administración de las claves:

Muchos tipos de claves y algoritmos entre los que elegir

Para que pueda personalizar sus propias soluciones, AWS CloudHSM ofrece muchos tipos de claves y algoritmos entre los que elegir. Los algoritmos admiten una variedad de tamaños de clave. Para obtener más información, consulte las páginas de atributos y mecanismos de cada AWS CloudHSM SDK de cliente.

Cómo se gestionan las claves

AWS CloudHSM las claves se administran mediante SDK y herramientas de línea de comandos. Para obtener información sobre cómo usar estas herramientas para administrar las claves, consulte Administrar claves en AWS CloudHSM y Mejores prácticas para AWS CloudHSM.

¿Quién posee las claves?

En AWS CloudHSM, el usuario criptográfico (CU) que crea la clave es el propietario de la misma. El propietario puede usar los comandos key share y key unshare para compartir y dejar de

compartir la clave con otras CU. Para obtener más información, consulte Uso de la CLI de CloudHSM para compartir y dejar de compartir claves.

El acceso y el uso se pueden controlar mediante el cifrado basado en atributos.

AWS CloudHSM permite utilizar el cifrado basado en atributos, una forma de cifrado que permite utilizar los atributos clave para controlar quién puede descifrar los datos en función de las políticas.

SDK de cliente

Al usarlo AWS CloudHSM, realiza operaciones criptográficas con los kits de desarrollo de software (SDK) de AWS CloudHSM cliente. AWS CloudHSM Los SDK de cliente incluyen:

- Estándars de criptografía de clave pública #11 (PKCS) #11
- Proveedor de JCE
- Motor dinámico de OpenSSL
- API de criptografía: próxima generación (CNG) y proveedor de almacenamiento de claves (key storage provider, KSP) para Microsoft Windows

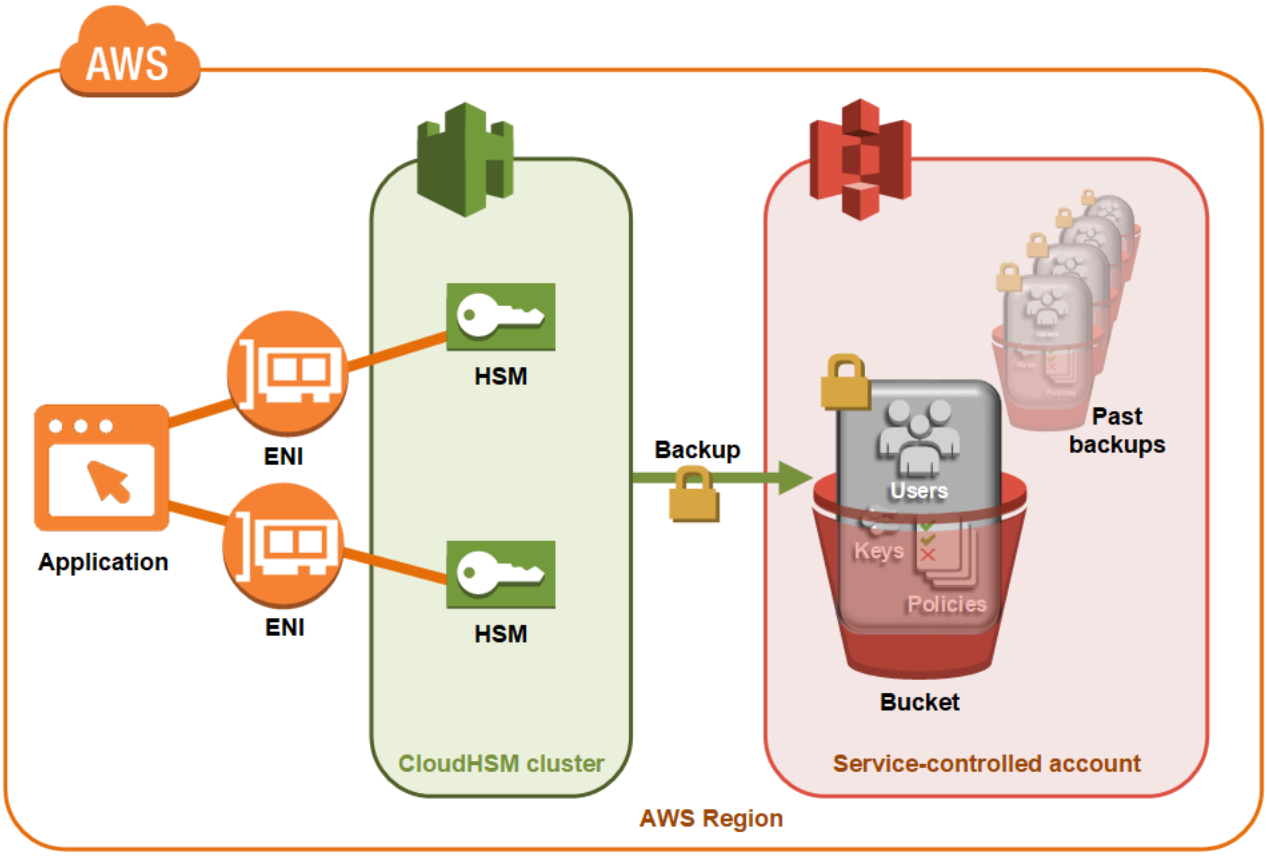
Puede usar alguno de estos SDK o todos ellos en su AWS CloudHSM clúster. Escriba el código de su aplicación para usar estos SDK para realizar operaciones criptográficas en sus HSM.

Las herramientas de utilidades y de línea de comandos son necesarias no solo para usar los SDK, sino también para configurar las credenciales, las políticas y los ajustes de su aplicación. Para obtener más información, consulte AWS CloudHSM herramientas de línea de comandos.

Para obtener más información sobre la instalación y el uso del SDK de cliente o sobre la seguridad de la conexión del cliente, consulte SDK del cliente y nd-to-end Cifrado electrónico.

AWS CloudHSM copias de seguridad en clúster

AWS CloudHSM realiza copias de seguridad periódicas de los usuarios, las claves y las políticas del clúster. Las copias de seguridad son seguras, duraderas y se actualizan según un cronograma predecible. En la siguiente ilustración, se muestra la relación de las copias de seguridad con el clúster.



Para obtener más información sobre cómo trabajar con copias de seguridad, consulte Administración de copias de seguridad.

Seguridad

Cuando AWS CloudHSM realiza una copia de seguridad desde el HSM, el HSM cifra todos sus datos antes de enviarlos a. AWS CloudHSM Los datos nunca salen del HSM en formato de texto no cifrado. Además, las copias de seguridad no se pueden descifrar AWS porque AWS no tiene acceso a la clave utilizada para descifrarlas. Para obtener más información, consulte Seguridad de las copias de seguridad del clúster.

Durabilidad

AWS CloudHSM almacena las copias de seguridad en un depósito de Amazon Simple Storage Service (Amazon S3) controlado por el servicio en la misma región que su clúster. Las copias de seguridad tienen un nivel de durabilidad del 99,999999999 %, equivalente al de cualquier objeto almacenado en Amazon S3.

Regiones

Para obtener información sobre las regiones compatibles AWS CloudHSM, consulte AWS CloudHSM Regiones y puntos finales en la tabla de regiones o en la Referencia general de AWStabla de regiones.

AWS CloudHSM puede que no esté disponible en todas las zonas de disponibilidad de una región determinada. Sin embargo, esto no debería afectar al rendimiento, ya que la carga AWS CloudHSM se equilibra automáticamente en todos los HSM de un clúster.

Como la mayoría de AWS los recursos, los clústeres y los HSM son recursos regionales. No es posible reutilizar o ampliar un clúster entre regiones. Debe realizar todos los pasos necesarios que se muestran en Empezar con AWS CloudHSM para crear un clúster en una nueva región.

Con fines de recuperación ante desastres, AWS CloudHSM le permite copiar copias de seguridad de su AWS CloudHSM clúster de una región a otra. Para obtener más información, consulte AWS CloudHSM copias de seguridad en clúster.

Precios

Con AWS CloudHSM, paga por horas sin compromisos a largo plazo ni pagos por adelantado. Para obtener más información, consulta AWS CloudHSM los precios en el AWS sitio web.

Empezar con AWS CloudHSM

Los siguientes temas le ayudan a crear, inicializar y activar un AWS CloudHSM clúster. Después de completar estos procedimientos, estará preparado para administrar usuarios y clústeres, y para utilizar las bibliotecas de software incluidas para realizar operaciones criptográficas.

Contenido

- Creación de grupos administrativos de IAM
- Cree una nube privada virtual (VPC).
- Crear un clúster
- Revisión del grupo de seguridad del clúster
- Lance una instancia de cliente de Amazon EC2.
- Configuración de los grupos de seguridad de la instancia de cliente de Amazon EC2
- Creación de un HSM
- Verificar la identidad y la autenticidad del HSM de un clúster (opcional)
- Inicio del clúster
- Instalación y configuración de la CLI de CloudHSM
- Activación del clúster
- Reconfigurar SSL con un nuevo certificado y clave privada (opcional)
- Creación de una aplicación

Creación de grupos administrativos de IAM

Como mejor práctica, no utilices tu Usuario raíz de la cuenta de AWS para interactuar con AWS, incluso AWS CloudHSM. En su lugar, utilice AWS Identity and Access Management (IAM) para crear un usuario de IAM, un rol de IAM o un usuario federado. Siga los pasos de la sección Creación de un grupo de usuarios de IAM y de administradores para crear un grupo de administradores y adjuntarle la AdministratorAccesspolítica. A continuación, cree un usuario administrador y agréguelo al grupo. Puede agregar usuarios adicionales al grupo según sea necesario. Cada usuario que añada hereda la AdministratorAccesspolítica del grupo.

Otra práctica recomendada es crear un grupo de AWS CloudHSM administradores que solo tenga los permisos necesarios para ejecutarse AWS CloudHSM. Puede agregar usuarios individuales a este grupo según sea necesario. Cada usuario hereda los permisos limitados que se han asociado

al grupo en lugar de tener acceso completo a AWS . La siguiente Políticas gestionadas por el cliente para AWS CloudHSM sección contiene la política que debe adjuntar a su grupo de AWS CloudHSM administradores.

AWS CloudHSM define un rol vinculado a un servicio para su cuenta. AWS El rol vinculado al servicio define actualmente los permisos que permiten a su cuenta registrar eventos. AWS CloudHSM Usted puede crear el rol de forma automática AWS CloudHSM o manual. No puede editar el rol, pero puede eliminarlo. Para obtener más información, consulte Funciones vinculadas al servicio para AWS CloudHSM.

Creación de un grupo de usuarios de IAM y de administradores

Para comenzar, cree un usuario de IAM y un grupo de administradores para ese usuario.

Inscríbase en un Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Signing in as the root user en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Para ver ejemplos de políticas AWS CloudHSM que puede adjuntar a su grupo de usuarios de IAM, consulte. Administración de identidad y acceso para AWS CloudHSM

Cree una nube privada virtual (VPC).

Si todavía no tiene una nube privada virtual (VPC), siga los pasos indicados en este tema para crear una.

> **Note**
>
> Si sigue estos pasos, se crearán subredes públicas y privadas.

Para crear una VPC

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En la barra de navegación, utilice el selector de regiones para elegir una de las AWS regiones compatibles AWS CloudHSM actualmente.
3. Seleccione el botón Crear VPC.
4. En Recursos para crear, elija VPC y más.
5. En Generación automática del nombre de la etiqueta, escriba un nombre identificable, como **`CloudHSM`**.
6. Deje el resto de opciones con sus valores predeterminados.
7. Seleccione Crear VPC.
8. Una vez creada la VPC, seleccione Ver VPC para ver la VPC que acaba de crear.

Crear un clúster

Un clúster es un conjunto de HSM individuales. AWS CloudHSM sincroniza los HSM de cada clúster para que funcionen como una unidad lógica.

Al crear un clúster, AWS CloudHSM crea un grupo de seguridad para el clúster en su nombre. Este grupo de seguridad controla el acceso de red a los HSM del clúster. Este grupo solamente permite las conexiones entrantes que proceden de las instancias Amazon Elastic Compute Cloud (Amazon EC2) que están en el grupo de seguridad. De forma predeterminada, el grupo de seguridad no contiene instancias. Posteriormente, debe lanzar una instancia de cliente y configurar el grupo de seguridad del clúster para permitir la comunicación y las conexiones con los HSM.

> **Important**
>
> Al crear un clúster, AWS CloudHSM crea un rol vinculado a un servicio denominado. AWSServiceRoleForCloudHSM Si AWS CloudHSM no puede crear el rol o el rol aún no existe, es posible que no pueda crear un clúster. Para obtener más información, consulte Solución de errores de creación de clústeres. Para obtener más información acerca de los roles vinculados a servicios, consulte Funciones vinculadas al servicio para AWS CloudHSM.

Puede crear un clúster desde la AWS CloudHSM consola, la AWS Command Line Interface (CLI) o la AWS CloudHSM API.

Para crear un clúster (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. En la barra de navegación, utilice el selector de regiones para elegir una de las AWS regiones compatibles AWS CloudHSM actualmente.
3. Elija Create cluster.
4. En la sección Cluster configuration, haga lo siguiente:
   a. Para VPC, seleccione la VPC que ha creado en Cree una nube privada virtual (VPC)..
   b. Para Zonas(s) de disponibilidad), junto a cada zona de disponibilidad, elija la subred privada que ha creado.

> **Note**
>
> Incluso si no AWS CloudHSM es compatible en una zona de disponibilidad determinada, el rendimiento no debería verse afectado, ya que la carga AWS CloudHSM se equilibra automáticamente entre todos los HSM de un clúster. Consulte AWS CloudHSM Regiones y puntos de conexión en Referencia general de AWSpara ver qué zonas de disponibilidad admiten. AWS CloudHSM

5. Elija Siguiente.
6. Especifique durante cuánto tiempo el servicio debe retener las copias de seguridad.

> **Note**
>
> Acepte el período de retención predeterminado de 90 días o escriba un nuevo valor de entre 7 y 379 días. El servicio eliminará automáticamente las copias de seguridad de este clúster que sean anteriores al valor que especifique aquí. Puede cambiar este valor posteriormente. Para obtener más información, consulte Configuración de retención de copias de seguridad.

7. Seleccione Siguiente.
8. (Opcional) Escriba una clave de etiqueta y un valor de etiqueta opcional. Para agregar más de una etiqueta al clúster, elija Agregar etiqueta.
9. Elija Revisar.
10. Revise la configuración del clúster y, a continuación, elija Crear clúster.

Para crear un clúster (CLI)

- En el símbolo del sistema, ejecute el comando create-cluster. Especifique el tipo de instancia de HSM, el período de retención de copias de seguridad y los ID de subred de las subredes donde piensa crear los HSM. Utilice los ID de las subredes privadas que ha creado. Especifique solo una subred por zona de disponibilidad.

```
$ aws cloudhsmv2 create-cluster --hsm-type hsm1.medium \
      --backup-retention-policy Type=DAYS,Value=<number of days> \
      --subnet-ids <subnet ID>
```

```
{
    "Cluster": {
        "BackupPolicy": "DEFAULT",
        "BackupRetentionPolicy": {
            "Type": "DAYS",
            "Value": 90
         },
        "VpcId": "vpc-50ae0636",
        "SubnetMapping": {
            "us-west-2b": "subnet-49a1bc00",
            "us-west-2c": "subnet-6f950334",
            "us-west-2a": "subnet-fd54af9b"
        },
        "SecurityGroup": "sg-6cb2c216",
        "HsmType": "hsm1.medium",
        "Certificates": {},
        "State": "CREATE_IN_PROGRESS",
        "Hsms": [],
        "ClusterId": "cluster-igklspoyj5v",
        "CreateTimestamp": 1502423370.069
    }
}
```

Para crear un clúster (AWS CloudHSM API)

- Envíe una solicitud CreateCluster. Especifique el tipo de instancia de HSM, la política de retención de copias de seguridad y los ID de subred de las subredes donde piensa crear los HSM. Utilice los ID de las subredes privadas que ha creado. Especifique solo una subred por zona de disponibilidad.

Si sus intentos de crear un clúster no tienen éxito, es posible que se debe a algún problema con las funciones vinculadas a servicios de AWS CloudHSM . Para ayudar a resolver el error, consulte Solución de errores de creación de clústeres.

Revisión del grupo de seguridad del clúster

Al crear un clúster, AWS CloudHSM crea un grupo de seguridad con ese nombre`cloudhsm-cluster-`*`clusterID`*`-sg`. Este grupo de seguridad contiene una regla TCP preconfigurada que permite la comunicaciones de entrada y de salida en el grupo de seguridad del clúster en los puertos

2223-2225. Este SG permite que las instancias de EC2 usen la VPC para comunicarse con los HSM de su clúster.

> **Warning**
>
> - No elimine ni modifique la regla TCP preconfigurada que existe en el grupo de seguridad del clúster. Esta regla puede evitar problemas de conectividad y el acceso no autorizado a los HSM.
> - El grupo de seguridad del clúster impide el acceso no autorizado a los HSM. Cualquier usuario que puede tener acceso a las instancias del grupo de seguridad también puede tener acceso a los HSM. La mayoría de las operaciones requieren que un usuario inicie sesión en el HSM. Sin embargo, es posible poner a cero los HSM sin autenticación, lo que destruye el material de claves, los certificados y los demás datos. Si sucede esto, los datos creados o modificados después de la copia de seguridad más reciente se pierden y no se pueden recuperar. Para evitar el acceso no autorizado, asegúrese de que solo los administradores de confianza puedan modificar o tener acceso a las instancias del grupo de seguridad predeterminado.

En el siguiente paso, puede lanzar una instancia de Amazon CE2 y conectarla a los HSM adjuntándole el grupo de seguridad del clúster.

Lance una instancia de cliente de Amazon EC2.

Para interactuar con el AWS CloudHSM clúster y las instancias de HSM y administrarlos, debe poder comunicarse con las interfaces de red elásticas de sus HSM. La forma más sencilla de hacerlo es utilizar una instancia EC2 en la misma VPC que el clúster. También puede utilizar los siguientes recursos de AWS para conectarse al clúster:

- Amazon VPC Peering
- AWS Direct Connect
- Conexiones de VPN

> **Note**
>
> Esta guía proporciona un ejemplo simplificado de cómo conectar una instancia EC2 a su clúster. AWS CloudHSM Para obtener información sobre las prácticas recomendadas en relación con las configuraciones de red seguras, consulte. Acceso seguro a su clúster

La AWS CloudHSM documentación suele suponer que está utilizando una instancia EC2 en la misma VPC y zona de disponibilidad (AZ) en la que creó el clúster.

Para crear una instancia EC2;

1. Abra la consola de EC2 en https://console.aws.amazon.com/ec2/.
2. Seleccione Lanzar instancia. En el menú desplegable, elija Lanzar instancia.
3. En el campo Nombre escriba un nombre para la instancia EC2.
4. En la sección Aplicaciones e imágenes de SO (imagen de máquina de Amazon), elija una imagen de máquina de Amazon (AMI) que corresponda a una plataforma compatible con CloudHSM. Para obtener más información, consulte Plataformas compatibles con SDK 5 de cliente.
5. En la sección Tipo de instancia, seleccione el tipo de instancia.
6. En la sección Par de claves, use un par de claves existente o seleccione Crear nuevo par de claves y siga estos pasos:
   a. En Nombre del par de claves, introduzca un nombre para el par de claves.
   b. En Tipo de par de claves, elija un tipo de par de claves.
   c. En Formato de archivo de la clave privada, elija el formato de la clave privada.
   d. Seleccione Crear par de claves.
   e. Descargue y guarde el archivo de clave privada.

> **Important**
>
> Esta es la única oportunidad que tiene de guardar el archivo de clave privada. Descargue el archivo y guárdelo en un lugar seguro. Proporcione el nombre del par de claves al lanzar una instancia. Además, debe proporcionar la clave privada

correspondiente cada vez que se conecte a la instancia y elegir el par de claves que creó al configurarla.

7. En Configuración de red, seleccione Editar.
8. En VPC, elija la VPC que ha creado anteriormente para el clúster.
9. En Subnet (Subred), elija la subred pública que ha creado para la VPC.
10. En Auto-assign Public IP (Autoasignar IP pública), elija Enable (Habilitar).
11. Elija Seleccionar un grupo de seguridad existente.
12. En Grupos de seguridad comunes, seleccione el grupo de seguridad predeterminado en el menú desplegable.
13. En Configurar almacenamiento, elija una configuración de almacenamiento en los menús desplegables.
14. En la ventana Resumen, seleccione Lanzar instancia.

> **Note**
>
> Al completar este paso se iniciará el proceso de creación de su instancia EC2.

Para obtener más información acerca de la creación de un cliente de Linux en Amazon EC2, consulte Introducción a las instancias de Linux en Amazon EC2. Para obtener información acerca de la conexión al cliente en ejecución, consulte los siguientes temas:

- Conexión a la instancia de Linux mediante SSH
- Conexión a la instancia Linux desde Windows utilizando PuTTY

La guía del usuario de Amazon EC2 contiene instrucciones detalladas para configurar y utilizar las instancias Amazon EC2. La siguiente lista proporciona información general sobre la documentación disponible para los clientes de Amazon EC2 de Linux y Windows:

- Para crear un cliente de Amazon EC2 de Linux, consulte Introducción a las instancias de Linux en Amazon EC2.

  Para obtener información acerca de la conexión al cliente en ejecución, consulte los siguientes temas:

  - Conexión a la instancia de Linux mediante SSH

  - Conexión a la instancia Linux desde Windows utilizando PuTTY
- Para crear un cliente de Amazon EC2 de Windows, consulte Introducción a las instancias de Windows en Amazon EC2. Para obtener más información acerca de cómo conectarse a su cliente de Windows, consulte Conectarse a su instancia de Windows.

> **Note**
>
> La instancia EC2 puede ejecutar todos los comandos CLI incluidos en esta guía. Si la CLI no está instalada, puede descargarla desde AWS Command Line Interface. Si utiliza Windows, puede descargar y ejecutar un instalador de Windows de 64 o 32 bits. Si utiliza Linux o macOS, puede instalar la CLI con pip.

Configuración de los grupos de seguridad de la instancia de cliente de Amazon EC2

Al lanzar una instancia de Amazon EC2, la asoció a un grupo de seguridad de Amazon VPC predeterminado. En este tema, se explica cómo asociar el grupo de seguridad del clúster a la instancia EC2. Esta asociación permite que el AWS CloudHSM cliente que se ejecuta en su instancia EC2 se comunique con sus HSM. Para conectar la instancia EC2 al AWS CloudHSM clúster, debe configurar correctamente el grupo de seguridad predeterminado de la VPC y asociar el grupo de seguridad del clúster a la instancia.

Modificar el grupo de seguridad predeterminado

Es necesario modificar el grupo de seguridad predeterminado para permitir la conexión SSH o RDP para descargar e instalar el software de cliente e interactuar con el HSM.

Para modificar el grupo de seguridad predeterminado

1. Abra el Panel EC2 en https://console.aws.amazon.com/ec2/.
2. Seleccione Instancias (en ejecución) y, a continuación, active la casilla de verificación situada junto a la instancia de EC2 en la que desee instalar el cliente. AWS CloudHSM
3. En la pestaña Seguridad, elija el grupo de seguridad denominado Predeterminado.
4. En la parte superior de la página, elija Actions (Acciones) y, a continuación, Edit inbound rules (Editar reglas de entrada).

5. Seleccione Add Rule (Añadir regla).
6. En Type (Tipo), realice una de las operaciones siguientes:
   - Para una instancia de Amazon EC2 de Windows Server, seleccione RDP. El puerto 3389 se rellena automáticamente.
   - Para una instancia de Amazon EC2 de Linux, seleccione SSH. El rango de puertos 22 se rellena automáticamente.
7. Para cualquiera de las opciones, defina Origen en Mi IP para poder comunicarse con su instancia de Amazon EC2.

   > **Important**
   >
   > No especifique 0.0.0.0/0 como rango CIDR, a fin de evitar permitir que cualquier persona tenga acceso a la instancia.

8. Seleccione Guardar.

Conecte la instancia de Amazon EC2 al clúster AWS CloudHSM

Debe asociar el grupo de seguridad del clúster a la instancia EC2, de modo que la instancia EC2 pueda comunicarse con los HSM del clúster. El grupo de seguridad del clúster contiene una regla preconfigurada que permite la comunicación entrante en los puertos 2223-2225.

Para conectar la instancia EC2 al clúster AWS CloudHSM

1. Abra la consola de EC2 en https://console.aws.amazon.com/ec2/.
2. Seleccione Instancias (en ejecución) y, a continuación, active la casilla de verificación de la instancia EC2 en la que desee instalar el AWS CloudHSM cliente.
3. En la parte superior de la página, seleccione Acciones, Seguridad y, a continuación, Cambiar grupos de seguridad.
4. Seleccione el grupo de seguridad cuyo nombre coincida con el ID del clúster, por ejemplo, `cloudhsm-cluster-`*`clusterID`*`-sg`.
5. Seleccione Agregar grupos de seguridad.
6. Seleccione Guardar.

> **Note**
>
> Puede asignar un máximo de cinco grupos de seguridad a una instancia de Amazon EC2. Si ha alcanzado el límite máximo, debe modificar el grupo de seguridad predeterminado de la instancia Amazon EC2 y el grupo de seguridad de clúster:
>
> En el grupo de seguridad predeterminado, haga lo siguiente:
>
> - Añada una regla de entrada para permitir el tráfico mediante el protocolo TCP en los puertos `2223-2225` desde el grupo de seguridad del clúster.
>
> En el grupo de seguridad del clúster, haga lo siguiente:
>
> - Añada una regla de entrada para permitir el tráfico mediante el protocolo TCP a través de puertos `2223-2225` desde el grupo de seguridad predeterminado.

Creación de un HSM

Después de crear un clúster, puede crear un HSM. Sin embargo, para poder crear un HSM en su clúster, este debe encontrarse en el estado sin inicializar. Para determinar el estado del clúster, consulte la página de clústeres en la AWS CloudHSM consola, utilice la CLI para ejecutar el describe-clusters comando o envíe una DescribeClusterssolicitud en la AWS CloudHSM API. Puede crear un HSM desde la AWS CloudHSM consola, la CLI o la AWS CloudHSM API.

Para crear un HSM (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Seleccione el botón de opción situado junto al ID del clúster para el que desea crear un HSM.
3. Seleccione Acciones. En el menú desplegable, elija Inicializar.
4. Elija una zona de disponibilidad (AZ) para el HSM que está creando.
5. Seleccione Crear.

Para crear un HSM (CLI)

- En el símbolo del sistema, ejecute el comando create-hsm. Especifique el ID del clúster que ha creado anteriormente y una zona de disponibilidad para el HSM. Especifique la zona de disponibilidad con el formato `us-west-2a`, `us-west-2b`, etc.

```
$ aws cloudhsmv2 create-hsm --cluster-id <cluster ID> --availability-
zone <Availability Zone>

{
    "Hsm": {
        "HsmId": "hsm-ted36yp5b2x",
        "EniIp": "10.0.1.12",
        "AvailabilityZone": "us-west-2a",
        "ClusterId": "cluster-igklspoyj5v",
        "EniId": "eni-5d7ade72",
        "SubnetId": "subnet-fd54af9b",
        "State": "CREATE_IN_PROGRESS"
    }
}
```

Para crear un HSM (API)AWS CloudHSM

- Envíe una solicitud CreateHsm. Especifique el ID del clúster que ha creado anteriormente y una zona de disponibilidad para el HSM.

Después de crear un clúster y un HSM, tiene la opción de verificar la identidad del HSM, o continuar directamente en Inicio del clúster.

Verificar la identidad y la autenticidad del HSM de un clúster (opcional)

Para inicializar el clúster, debe firmar una solicitud de firma de certificado (CSR) generada por el primer HSM del clúster. Antes de hacerlo, es posible que desee verificar la identidad y la autenticidad del HSM.

> **Note**
>
> Este proceso es opcional. Sin embargo, funciona únicamente hasta que se inicializa un clúster. Una vez que se inicializa el clúster, no puede utilizar este proceso para obtener los certificados o verificar los HSM.

Temas

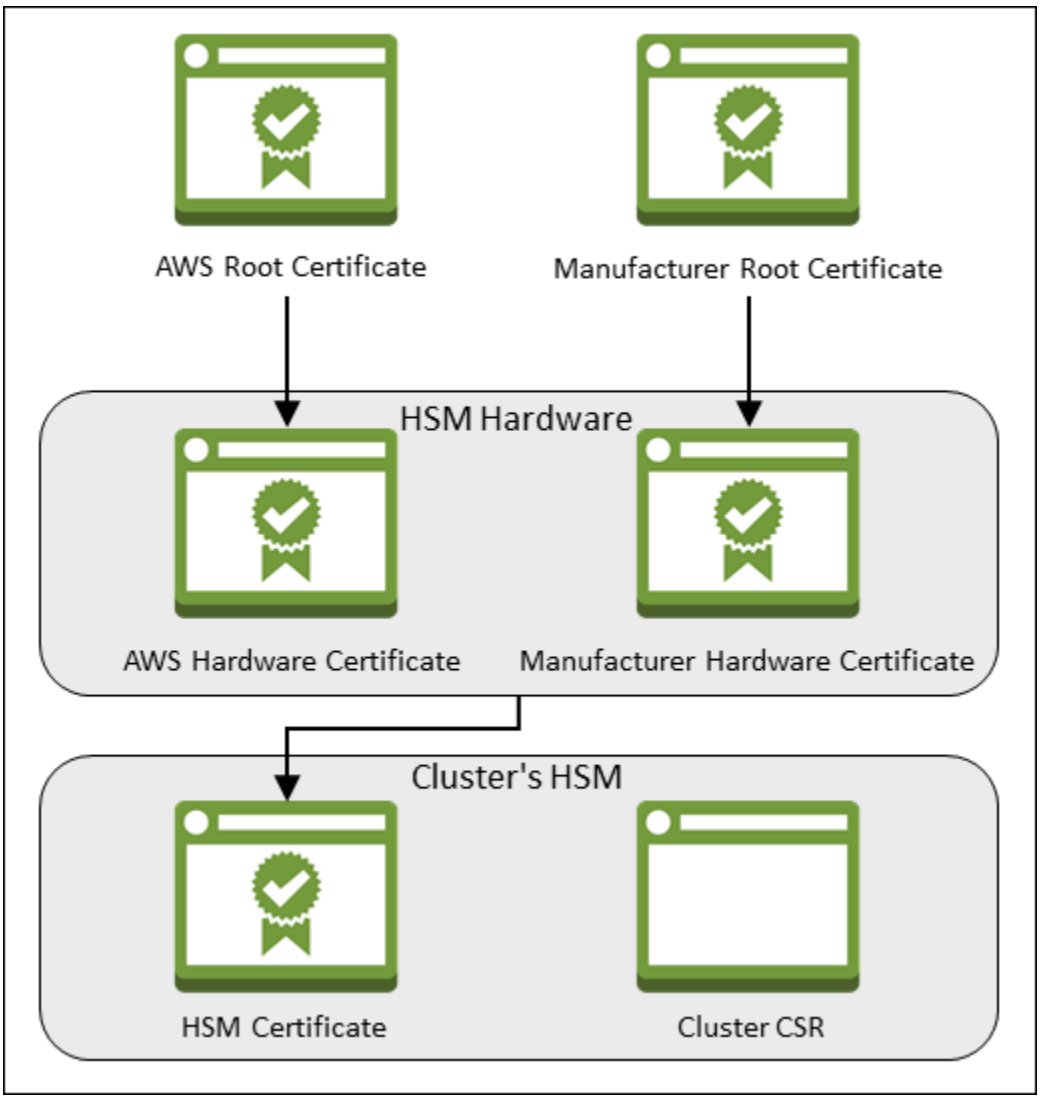
- Información general
- Obtención de los certificados del HSM8
- Obtención de los certificados raíz
- Verificación de las cadenas de certificados
- Extracción y comparación de las claves públicas

Información general

Para verificar la identidad del primer HSM del clúster, siga los pasos que se describen a continuación:

1. Obtener los certificados y las CSR: en este paso, recibirá tres certificados y una CSR desde el HSM. También obtendrá dos certificados raíz, uno del fabricante del hardware del HSM AWS CloudHSM y otro del fabricante del hardware.
2. Verificar las cadenas de certificados: en este paso, se crean dos cadenas de certificados, una para el certificado AWS CloudHSM raíz y otra para el certificado raíz del fabricante. A continuación, verifica el certificado HSM con estas cadenas de certificados para determinarlo AWS CloudHSM y el fabricante del hardware certifica la identidad y autenticidad del HSM.
3. Comparar las claves públicas: en este paso, extraerá y comparará las claves públicas del certificado del HSM y de la CSR del clúster, para asegurarse de que sean las mismas. Esto debería darle la seguridad de que la CSR fue generada por un HSM auténtico y de confianza.

En el siguiente diagrama se muestran la CSR, los certificados, y la relación que existe entre unos y otros. En la lista que le sigue, se definen los distintos certificados.



AWS Certificado raíz

Este AWS CloudHSM es el certificado raíz.

Certificado raíz del fabricante

Este es el certificado raíz del fabricante del hardware.

AWS Certificado de hardware

AWS CloudHSM creó este certificado cuando se añadió el hardware de HSM a la flota. Este certificado afirma que AWS CloudHSM es el propietario del hardware.

Certificado de hardware del fabricante

El fabricante del hardware del HSM creó este certificado cuando fabricó el hardware del HSM. Este certificado confirma que el fabricante creó el hardware.

Certificado del HSM

El certificado del HSM es generado por el hardware validado por FIPS cuando crea el primer HSM en el clúster. Este certificado confirma que el hardware del HSM creó el HSM.

CSR del clúster

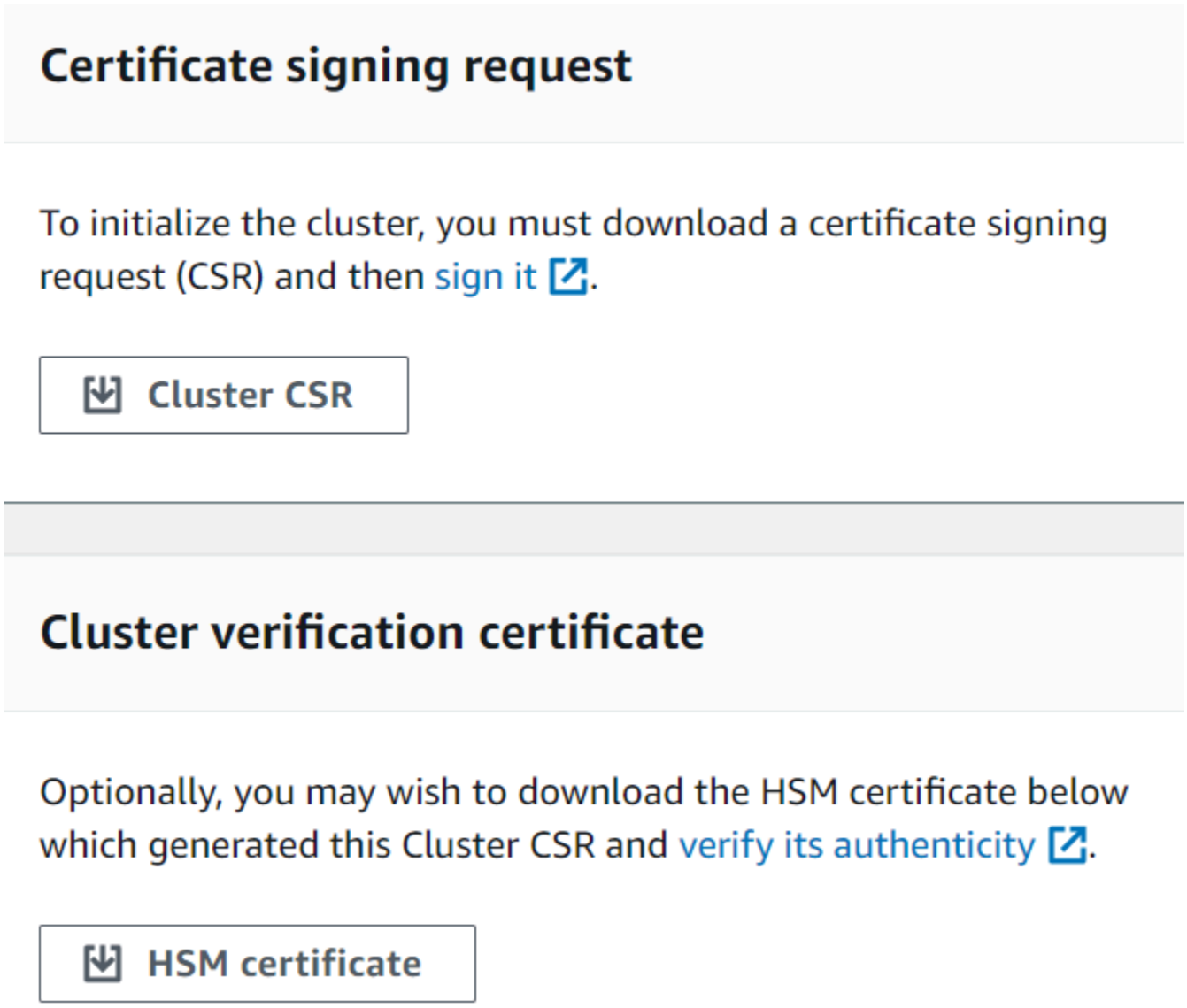
El primer HSM crea la CSR del clúster. Al firmar la CSR del clúster, solicita el clúster. A continuación, puede utilizar la CSR firmada para inicializar el clúster.

Obtención de los certificados del HSM8

Para verificar la identidad y la autenticidad del HSM, empiece por obtener una CSR y cinco certificados. Obtiene tres de los certificados del HSM, lo que puede hacer con la AWS CloudHSM consola, la AWS Command Line Interface (CLI) o la AWS CloudHSM API.

Para obtener la CSR y los certificados del HSM (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Seleccione el botón de opción situado junto a la ID del clúster con el HSM que desea verificar.
3. Seleccione Acciones. En el menú desplegable, elija Inicializar.
4. Si no ha completado el paso anterior para crear un HSM, elija una zona de disponibilidad (AZ) para el HSM que va a crear. A continuación, seleccione Crear.
5. Cuando los certificados y la CSR estén listos, verá enlaces para descargarlos.



6. Elija los enlaces necesarios para descargar y guardar la CSR y los certificados. Para simplificar los pasos posteriores, guarde todos los archivos en el mismo directorio y utilice los nombres de archivo predeterminados.

Para obtener los certificados CSR y HSM (CLI)

- En el símbolo del sistema, ejecute cuatro veces el comando describe-clusters para extraer la CSR y cada uno de los certificados y guardarlos en archivos.

    a. Escriba el siguiente comando para extraer la CSR del clúster. Sustituya *<ID de clúster>* por el ID del clúster que creó anteriormente.

```
$ aws cloudhsmv2 describe-clusters --filters clusterIds=<cluster ID> \
                                   --output text \
                                   --query 'Clusters[].Certificates.ClusterCsr'
 \
                                   > <cluster ID>_ClusterCsr.csr
```

b. Escriba el siguiente comando para extraer el certificado del HSM. Sustituya *<ID de clúster>* por el ID del clúster que creó anteriormente.

```
$ aws cloudhsmv2 describe-clusters --filters clusterIds=<cluster ID> \
                                   --output text \
                                   --query
 'Clusters[].Certificates.HsmCertificate' \
                                   > <cluster ID>_HsmCertificate.crt
```

c. Ejecute el siguiente comando para extraer el certificado de AWS hardware. Sustituya *<ID de clúster>* por el ID del clúster que creó anteriormente.

```
$ aws cloudhsmv2 describe-clusters --filters clusterIds=<cluster ID> \
                                   --output text \
                                   --query
 'Clusters[].Certificates.AwsHardwareCertificate' \
                                   > <cluster ID>_AwsHardwareCertificate.crt
```

d. Escriba el siguiente comando para extraer el certificado de hardware del fabricante. Sustituya *<ID de clúster>* por el ID del clúster que creó anteriormente.

```
$ aws cloudhsmv2 describe-clusters --filters clusterIds=<cluster ID> \
                                   --output text \
                                   --query
 'Clusters[].Certificates.ManufacturerHardwareCertificate' \
                                   > <cluster
 ID>_ManufacturerHardwareCertificate.crt
```

Para obtener los certificados CSR y HSM (API)AWS CloudHSM

- Envíe una solicitud DescribeClusters, extraiga la CSR y los certificados de la respuesta y guárdelos.

Obtención de los certificados raíz

Siga estos pasos para obtener los certificados raíz para AWS CloudHSM y para el fabricante. Guarde los archivos del certificado raíz en el directorio que contiene los archivos de los certificados de la CSR y del HSM.

Para obtener los certificados raíz AWS CloudHSM y del fabricante

1. Descargue el certificado AWS CloudHSM raíz: AWS_CloudHSM_Root-G1.zip
2. Descargue el certificado raíz del fabricante: liquid_security_certificate.zip.

   Para descargar el certificado desde la página de destino, https://www.marvell.com/products/security-solutions/ liquid-security-hsm-adapters -and-appliances/liquidsecurity-certificate.html y, a continuación, seleccione Descargar certificado.

   Es posible que tenga que hacer clic con el botón derecho del ratón en el enlace Descargar certificado y, a continuación, elegir Guardar enlace como... para guardar el archivo del certificado.
3. Después de descargar los archivos, extraiga (descomprima) el contenido.

Verificación de las cadenas de certificados

En este paso, se crean dos cadenas de certificados, una para el certificado AWS CloudHSM raíz y otra para el certificado raíz del fabricante. A continuación, utilice OpenSSL para verificar el certificado del HSM con cada una de las cadenas de certificados.

Para crear cadenas de certificados, abra un shell de Linux. Necesita OpenSSL, que está disponible en la mayoría de los shells de Linux, y necesita el certificado raíz y los archivos del certificado del HSM que descargó. Sin embargo, no necesita la CLI para este paso y no es necesario que el shell esté asociado a su AWS cuenta.

Para verificar el certificado HSM con el certificado AWS CloudHSM raíz

1. Desplácese hasta el directorio donde guardó el certificado raíz y los archivos del certificado del HSM que descargó. Los siguientes comandos presuponen que todos los certificados se encuentran en el directorio actual y utilizan los nombres de archivo predeterminados.

Utilice el siguiente comando para crear una cadena de certificados que incluya el certificado de AWS hardware y el certificado AWS CloudHSM raíz, en ese orden. Sustituya *<ID de clúster>* por el ID del clúster que creó anteriormente.

```
$ cat <cluster ID>_AwsHardwareCertificate.crt \
      AWS_CloudHSM_Root-G1.crt \
      > <cluster ID>_AWS_chain.crt
```

2. Utilice el siguiente comando de OpenSSL para verificar el certificado del HSM con la cadena de certificados de AWS . Sustituya *<ID de clúster>* por el ID del clúster que creó anteriormente.

```
$ openssl verify -CAfile <cluster ID>_AWS_chain.crt <cluster ID>_HsmCertificate.crt
<cluster ID>_HsmCertificate.crt: OK
```

Para verificar el certificado del HSM con el certificado raíz del fabricante

1. Utilice el siguiente comando para crear una cadena de certificados que incluya el certificado de hardware del fabricante y el certificado raíz del fabricante, en ese orden. Sustituya *<ID de clúster>* por el ID del clúster que creó anteriormente.

```
$ cat <cluster ID>_ManufacturerHardwareCertificate.crt \
      liquid_security_certificate.crt \
      > <cluster ID>_manufacturer_chain.crt
```

2. Utilice el siguiente comando de OpenSSL para verificar el certificado del HSM con la cadena de certificados del fabricante. Sustituya *<ID de clúster>* por el ID del clúster que creó anteriormente.

```
$ openssl verify -CAfile <cluster ID>_manufacturer_chain.crt <cluster
 ID>_HsmCertificate.crt
<cluster ID>_HsmCertificate.crt: OK
```

Extracción y comparación de las claves públicas

Utilice OpenSSL para extraer y comparar las claves públicas del certificado del HSM y de la CSR del clúster, con objeto de asegurarse de que sean las mismas.

Para comparar las claves públicas, utilice el shell de Linux. Necesita OpenSSL, que está disponible en la mayoría de los shells de Linux, pero no necesita la CLI para este paso. No es necesario que el shell esté asociado a su AWS cuenta.

Para extraer y comparar las claves públicas

1. Utilice el siguiente comando para extraer la clave pública del certificado del HSM.

```
$ openssl x509 -in <cluster ID>_HsmCertificate.crt -pubkey -noout > <cluster
 ID>_HsmCertificate.pub
```

2. Utilice el siguiente comando para extraer la clave pública de la CSR del clúster.

```
$ openssl req -in <cluster ID>_ClusterCsr.csr -pubkey -noout > <cluster
 ID>_ClusterCsr.pub
```

3. Utilice el siguiente comando para comparar las claves públicas. Si las claves públicas son idénticas, el siguiente comando no devuelve ningún resultado.

```
$ diff <cluster ID>_HsmCertificate.pub <cluster ID>_ClusterCsr.pub
```

Después de verificar la identidad y autenticidad del HSM, continúe con Inicio del clúster.

Inicio del clúster

Complete los pasos de los siguientes temas para inicializar el clúster AWS CloudHSM .

> **Note**
>
> Antes de inicializar el clúster, revise el proceso mediante el cual se puede verificar la identidad y autenticidad de los HSM. Este proceso es opcional y solo funciona hasta que se inicializa un clúster. Una vez que se inicializa el clúster, no puede utilizar este proceso para obtener sus certificados o verificar los HSM.

Temas

- Obtención del CSR del clúster
- Firmar la CSR

- Inicio del clúster

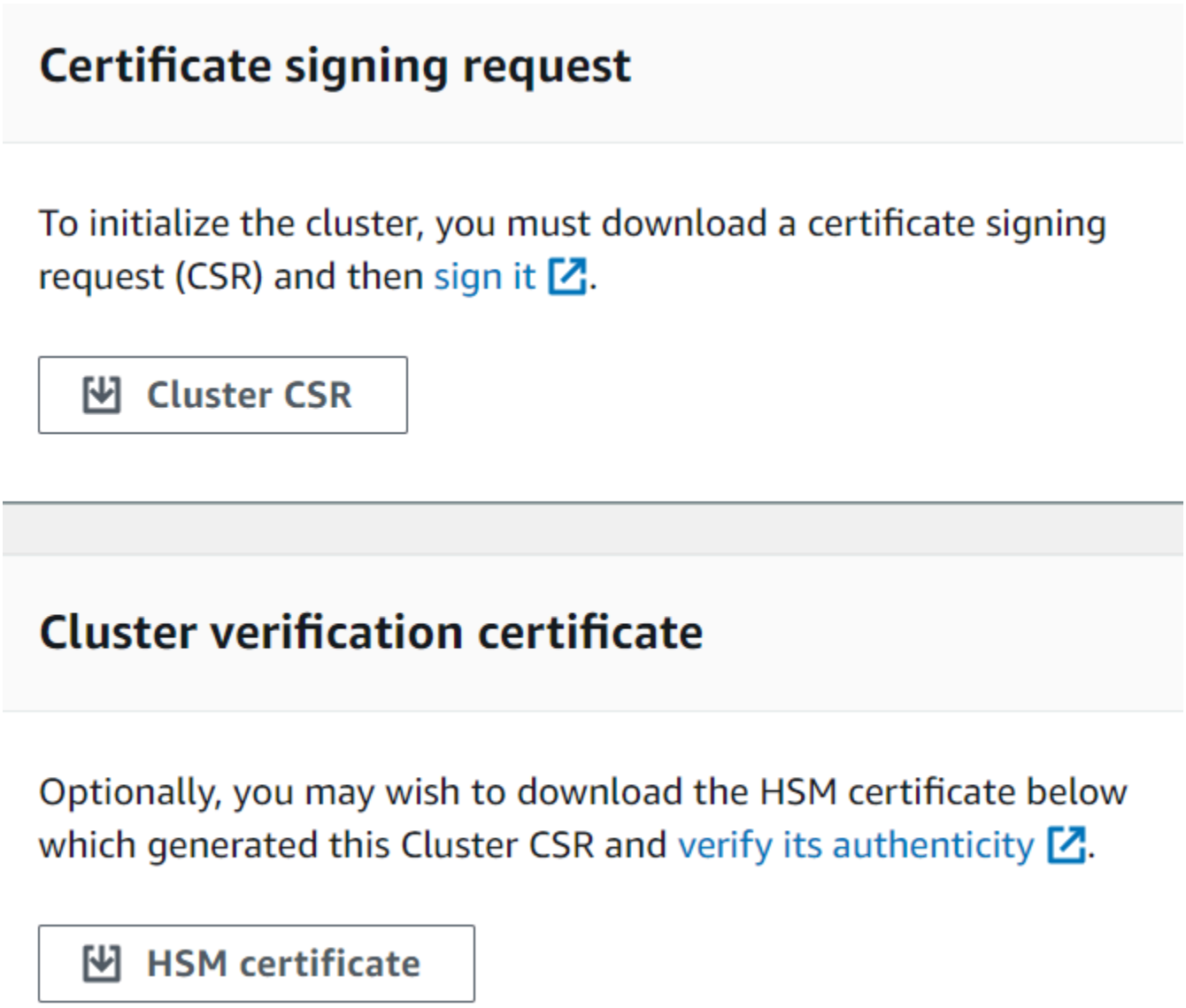
Obtención del CSR del clúster

Para poder inicializar el clúster, debe descargar y firmar una solicitud de firma de certificado (CSR) generada por el primer HSM del clúster. Si ha seguido los pasos para verificar la identidad del HSM del clúster, ya tiene la CSR y puede firmarla. De lo contrario, obtenga la CSR ahora mediante la AWS CloudHSM consola, la AWS Command Line Interface (CLI) o la AWS CloudHSM API.

> **Important**
>
> Para inicializar su clúster, su anclaje de confianza debe cumplir con RFC 5280 y satisfacer los siguientes requisitos:
>
> - Si usa extensiones X509v3, debe estar presente la extensión X509v3 Basic Constraints.
> - El anclaje de confianza debe ser un certificado autofirmado.
> - Los valores de extensión no deben entrar en conflicto entre sí.

Para obtener la CSR (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Seleccione el botón de opción situado junto a la ID del clúster con el HSM que desea verificar.
3. Seleccione Acciones. En el menú desplegable, elija Inicializar.
4. Si no ha completado el paso anterior para crear un HSM, elija una zona de disponibilidad (AZ) para el HSM que va a crear. A continuación, seleccione Crear.
5. Cuando la CSR esté lista, verá un enlace para descargarla.



6. Elija Cluster CSR para descargar y guardar la CSR.

Para obtener la CSR (CLI)

- En un símbolo del sistema, ejecute el siguiente comando describe-clusters, que extrae la CSR y la guarda en un archivo. Sustituya *<ID de clúster>* por el ID del clúster que creó anteriormente.

```
$ aws cloudhsmv2 describe-clusters --filters clusterIds=<cluster ID> \
                                   --output text \
                                   --query 'Clusters[].Certificates.ClusterCsr' \
                                   > <cluster ID>_ClusterCsr.csr
```

Para obtener la CSR (API)AWS CloudHSM

1. Envíe una solicitud DescribeClusters.
2. Extraiga y guarde la CSR de la respuesta.

Firmar la CSR

Actualmente, debe crear un certificado de firma autofirmado y utilizarlo para firmar la CSR del clúster. No necesita la CLI para este paso y no es necesario que el shell esté asociado a su AWS cuenta. Para firmar la CSR, haga lo siguiente:

1. Complete la sección anterior (consulte Obtención del CSR del clúster).
2. Crear un clave privada.
3. Utilice la clave privada para crear un certificado de firma.
4. Firme la CSR del clúster.

Crear un clave privada

> **Note**
>
> Para un clúster de producción, la clave debe crearse de forma segura mediante una fuente de asignación al azar. Recomendamos que utilice un HSM externo seguro que esté sin conexión o equivalente. Guarde la clave de forma segura. La clave define la identidad del clúster y su control exclusivo sobre los HSM que contiene.
> Durante las fases de desarrollo y pruebas, puede utilizar cualquier herramienta adecuada (como OpenSSL) para crear y firmar el certificado del clúster. En el ejemplo siguiente se muestra cómo crear una clave. Cuando haya utilizado la clave para crear un certificado autofirmado (vea el procedimiento a continuación), debe guardarla de forma segura. Para iniciar sesión en la AWS CloudHSM instancia, el certificado debe estar presente, pero la clave privada no.

Utilice el siguiente comando para crear una clave privada. Al inicializar un AWS CloudHSM clúster, debes usar el certificado RSA 2048 o el certificado RSA 4096.

```
$ openssl genrsa -aes256 -out customerCA.key 2048
```

```
Generating RSA private key, 2048 bit long modulus
........+++
............+++
e is 65537 (0x10001)
Enter pass phrase for customerCA.key:
Verifying - Enter pass phrase for customerCA.key:
```

Utilizar la clave privada para crear un certificado autofirmado

El hardware de confianza que se utiliza para crear la clave privada del clúster de producción también debe proporcionar una herramienta de software para generar un certificado autofirmado con dicha clave. En el ejemplo siguiente se utiliza OpenSSL y la clave privada que ha creado en el paso anterior para crear un certificado de firma. El certificado es válido durante 10 años (3652 días). Lea las instrucciones que aparecen en pantalla y siga las indicaciones.

```
$ openssl req -new -x509 -days 3652 -key customerCA.key -out customerCA.crt
Enter pass phrase for customerCA.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

Este comando crea un archivo de certificado denominado `customerCA.crt`. Coloque este certificado en todos los hosts desde los que se conectará al clúster. AWS CloudHSM Si asigna otro nombre al archivo o lo almacenarla en una ruta distinta de la raíz del host, debe editar el archivo de configuración del cliente según sea necesario. Utilice el certificado y la clave privada que acaba de crear para firmar la solicitud de firma de certificado (CSR) del clúster en el paso siguiente.

Firma de la CSR del clúster

El hardware de confianza que se utiliza para crear la clave privada del clúster de producción también debe proporcionar una herramienta para firmar la CSR con dicha clave. En el siguiente ejemplo se usa OpenSSL para firmar la CSR del clúster. En el ejemplo se utiliza la clave privada y el certificado autofirmado que ha creado en el paso anterior.

```
$ openssl x509 -req -days 3652 -in <cluster ID>_ClusterCsr.csr \
                               -CA customerCA.crt \
                               -CAkey customerCA.key \
                               -CAcreateserial \
                               -out <cluster ID>_CustomerHsmCertificate.crt
Signature ok
subject=/C=US/ST=CA/O=Cavium/OU=N3FIPS/L=SanJose/CN=HSM:<HSM
 identifer>:PARTN:<partition number>, for FIPS mode
Getting CA Private Key
Enter pass phrase for customerCA.key:
```

Este comando crea un archivo denominado *<cluster ID>*_CustomerHsmCertificate.crt. Utilice este archivo como el certificado firmado al inicializar el clúster.

Inicio del clúster

Use el certificado HSM firmado y su certificado de firma para inicializar el clúster. Puede usar la AWS CloudHSM consola, la CLI o la AWS CloudHSM API.

Para inicializar el clúster (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Seleccione el botón de opción situado junto a la ID del clúster con el HSM que desea verificar.
3. Seleccione Acciones. En el menú desplegable, elija Inicializar.
4. Si no ha completado el paso anterior para crear un HSM, elija una zona de disponibilidad (AZ) para el HSM que va a crear. A continuación, seleccione Crear.
5. En la página de descarga de solicitud de firma de certificado, elija Next (Siguiente). Si la opción Next no está disponible, primero elija uno de los enlaces de certificados o de CSR. A continuación, elija Next.
6. En la página Sign certificate signing request (CSR), elija Next.
7. En la página Upload the certificates, haga lo siguiente:

a. Junto a Cluster certificate, elija Upload file. A continuación, localice y seleccione el certificado del HSM que firmó anteriormente. Si ha realizado los pasos de la sección anterior, seleccione el archivo *<cluster ID>*_CustomerHsmCertificate.crt.

b. Junto a Issuing certificate, elija Upload file. A continuación, seleccione el certificado de firma. Si ha realizado los pasos de la sección anterior, seleccione el archivo customerCA.crt.

c. Elija Upload and initialize.

Para inicializar un clúster (CLI)

- En el símbolo del sistema, ejecute el comando initialize-cluster. Proporcione lo siguiente:
  - El ID del clúster que ha creado anteriormente.
  - El certificado del HSM que firmó anteriormente. Si ha realizado los pasos de la sección anterior, se encuentra en un archivo denominado *<cluster ID>*_CustomerHsmCertificate.crt.
  - Su certificado de firma. Si ha realizado los pasos de la sección anterior, el certificado de firma se guarda en un archivo denominado customerCA.crt.

```
$ aws cloudhsmv2 initialize-cluster --cluster-id <cluster ID> \
                                    --signed-cert file://<cluster
 ID>_CustomerHsmCertificate.crt \
                                    --trust-anchor file://customerCA.crt
{
    "State": "INITIALIZE_IN_PROGRESS",
    "StateMessage": "Cluster is initializing. State will change to INITIALIZED upon
 completion."
}
```

Para inicializar un clúster (API)AWS CloudHSM

- Envíe una solicitud InitializeCluster con lo siguiente:
  - El ID del clúster que ha creado anteriormente.
  - El certificado del HSM que firmó anteriormente.

- Su certificado de firma.

Instalación y configuración de la CLI de CloudHSM

Para interactuar con el HSM de su AWS CloudHSM clúster, necesita la CLI de CloudHSM.

Tareas

- Instale las herramientas de línea de comandos AWS CloudHSM

Instale las herramientas de línea de comandos AWS CloudHSM

Conéctese a su instancia de cliente y ejecute los siguientes comandos para descargar e instalar las herramientas de línea de AWS CloudHSM comandos. Para obtener más información, consulte Lance una instancia de cliente de Amazon EC2..

Utilice los comandos siguientes para descargar e instalar la CLI de CloudHSM.

Amazon Linux 2

Amazon Linux 2 en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-cli-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el7.x86_64.rpm
```

Amazon Linux 2 en arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-cli-latest.el7.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el7.aarch64.rpm
```

Amazon Linux 2023

Amazon Linux 2023 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/
cloudhsm-cli-latest.amzn2023.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.amzn2023.x86_64.rpm
```

Amazon Linux 2023 en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/
cloudhsm-cli-latest.amzn2023.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.amzn2023.aarch64.rpm
```

CentOS 7 (7.8+)

CentOS 7 en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-cli-
latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el7.x86_64.rpm
```

RHEL 7 (7.8+)

RHEL 7 en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-cli-
latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el7.x86_64.rpm
```

RHEL 8 (8.3+)

RHEL 8 en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-cli-
latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el8.x86_64.rpm
```

RHEL 9 (9.2+)

RHEL 9 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-cli-latest.el9.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el9.x86_64.rpm
```

RHEL 9 en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-cli-latest.el9.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el9.aarch64.rpm
```

Ubuntu 20.04 LTS

Ubuntu 20.04 LTS en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Focal/cloudhsm-cli_latest_u20.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-cli_latest_u20.04_amd64.deb
```

Ubuntu 22.04 LTS

Ubuntu 22.04 LTS en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/cloudhsm-cli_latest_u22.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-cli_latest_u22.04_amd64.deb
```

Ubuntu 22.04 LTS en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/cloudhsm-cli_latest_u22.04_arm64.deb
```

```
$ sudo apt install ./cloudhsm-cli_latest_u22.04_arm64.deb
```

Windows Server 2016

Para Windows Server 2016 con arquitectura x86_64, ábralo PowerShell como administrador y ejecute el siguiente comando:

```
PS C:\> wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Windows/
AWSCloudHSMCLI-latest.msi -Outfile C:\AWSCloudHSMCLI-latest.msi
```

```
PS C:\> Start-Process msiexec.exe -ArgumentList '/i C:\AWSCloudHSMCLI-latest.msi /
quiet /norestart /log C:\client-install.txt' -Wait
```

Windows Server 2019

Para Windows Server 2019 con una arquitectura x86_64, ábralo PowerShell como administrador y ejecute el siguiente comando:

```
PS C:\> wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Windows/
AWSCloudHSMCLI-latest.msi -Outfile C:\AWSCloudHSMCLI-latest.msi
```

```
PS C:\> Start-Process msiexec.exe -ArgumentList '/i C:\AWSCloudHSMCLI-latest.msi /
quiet /norestart /log C:\client-install.txt' -Wait
```

Use los siguientes comandos para configurar la CLI de CloudHSM.

Cómo iniciar el proceso de arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de los HSM de su clúster.

```
$ sudo /opt/cloudhsm/bin/configure-cli -a <The ENI IP addresses of the HSMs>
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de los HSM de su clúster.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-cli.exe" -a <The ENI IP addresses
 of the HSMs>
```

Activación del clúster

Al activar un AWS CloudHSM clúster, el estado del clúster cambia de inicializado a activo. A continuación, puede administrar los usuarios de hardware security module (HSM) y utilizar el HSM.

Important

Antes de poder activar el clúster, primero debe copiar el certificado de emisión en la ubicación predeterminada de la plataforma y en cada instancia de EC2 que se conecte al clúster (el certificado de emisión se crea al inicializar el clúster).

Linux

```
/opt/cloudhsm/etc/customerCA.crt
```

Windows

```
C:\ProgramData\Amazon\CloudHSM\customerCA.crt
```

Tras colocar el certificado de emisión, instale la CLI de CloudHSM y ejecute el comando cluster activate en su primer HSM. Observará que la cuenta de administrador del primer HSM del clúster tiene el rol de administrador desactivado. Se trata de un rol temporario que solo existe antes de la activación del clúster. Al activar el clúster, el rol de administrador desactivado cambia a administrador.

Cómo activar un clúster

1. Conéctese con la instancia de cliente que lanzó anteriormente. Para obtener más información, consulte Lance una instancia de cliente de Amazon EC2.. Puede lanzar una instancia de Linux o Windows Server.
2. Ejecute la CLI de CloudHSM en modo interactivo.

Linux

```
$ /opt/cloudhsm/bin/cloudhsm-cli interactive
```

Windows

```
C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
```

3. (Opcional) Utilice el comando user list para mostrar los usuarios existentes.

```
aws-cloudhsm > user list
{
  "error_code": 0,
  "data": {
    "users": [
      {
        "username": "admin",
        "role": "unactivated-admin",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      },
      {
        "username": "app_user",
        "role": "internal(APPLIANCE_USER)",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      }
    ]
  }
}
```

4. Utilice el comando cluster activate para establecer la contraseña de administrador inicial.

```
aws-cloudhsm > cluster activate
Enter
 password:<NewPassword>
Confirm password:<NewPassword>
{
  "error_code": 0,
```

```
  "data": "Cluster activation successful"
}
```

Le recomendamos que anote la contraseña nueva en una hoja de cálculo de contraseñas. No pierda la hoja de cálculo. Le recomendamos que imprima una copia de la hoja de cálculo de contraseñas, que utilice dicha hoja para registrar las contraseñas de HSM de importancia fundamental y que después la guarde en un lugar seguro. También es conveniente que guarde una copia de esta hoja de cálculo en un lugar seguro fuera de las instalaciones.

5. (Opcional) Utilice el comando user list para verificar que el tipo de usuario ha cambiado a administrador o CO.

```
aws-cloudhsm > user list
{
  "error_code": 0,
  "data": {
    "users": [
      {
        "username": "admin",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      },
       {
        "username": "app_user",
        "role": "internal(APPLIANCE_USER)",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      }
    ]
  }
}
```

6. Utilice el comando quit para detener la herramienta CLI de CloudHSM.

```
aws-cloudhsm > quit
```

Para obtener más información sobre cómo trabajar con la CLI o la CMU de CloudHSM, consulte Descripción de los usuarios de HSM y Descripción de la administración de usuarios de HSM con CMU.

Reconfigurar SSL con un nuevo certificado y clave privada (opcional)

AWS CloudHSM utiliza un certificado SSL para establecer una conexión con un HSM. Al instalar el cliente, se incluyen una clave predeterminada y un certificado SSL. Sin embargo, puede crear y usar los suyos. Tenga en cuenta que necesitará el certificado autofirmado (*customerCA.crt*) que creó al inicializar el clúster.

A un nivel alto, es un proceso que consta de dos pasos:

1. Primero, cree la clave privada; luego, utilice esa clave para crear una solicitud de firma de certificado (CSR). Utilice el certificado de emisión —el certificado que creó al inicializar el clúster— para firmar la CSR.
2. A continuación, utilice la herramienta de configuración para copiar la clave y el certificado en los directorios correspondientes.

Cree una clave, una CSR y, a continuación, firme la CSR.

Los pasos son los mismos para el SDK 3 de cliente o para el SDK 5 de cliente.

Para reconfigurar SSL con un nuevo certificado y clave privada

1. Cree una clave privada con el siguiente comando de OpenSSL:

```
openssl genrsa -out ssl-client.key 2048
Generating RSA private key, 2048 bit long modulus
........+++
............+++
e is 65537 (0x10001)
```

2. Utilice el siguiente comando de OpenSSL para crear una solicitud de firma de certificado (CSR). Se le harán varias preguntas sobre su certificado.

```
openssl req -new -sha256 -key ssl-client.key -out ssl-client.csr
```

```
Enter pass phrase for ssl-client.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

3. Firme la CSR con el certificado *`customerCA.crt`* que creó al inicializar el clúster.

```
openssl x509 -req -days 3652 -in ssl-client.csr \
        -CA customerCA.crt \
        -CAkey customerCA.key \
        -CAcreateserial \
        -out ssl-client.crt
Signature ok
subject=/C=US/ST=WA/L=Seattle/O=Example Company/OU=sales
Getting CA Private Key
```

Habilite el SSL personalizado para AWS CloudHSM

Los pasos son diferentes para el SDK 3 de cliente o para el SDK 5 de cliente. Para obtener más información acerca de cómo trabajar con la herramienta de línea de comandos, consulte ???.

Temas

- SSL personalizado para el SDK 3 de cliente

- SSL personalizado para SDK 5 de cliente

SSL personalizado para el SDK 3 de cliente

Use la herramienta de configuración del SDK 3 de cliente para habilitar el SSL personalizado. Para obtener más información sobre la herramienta de configuración para SDK 3 de cliente, consulte ???.

Cómo usar un certificado y una clave personalizados para la autenticación mutua cliente-servidor de TLS con SDK 3 de cliente en Linux

1. Copie la clave y el certificado en el directorio adecuado.

```
sudo cp ssl-client.crt /opt/cloudhsm/etc
sudo cp ssl-client.key /opt/cloudhsm/etc
```

2. Use la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
sudo /opt/cloudhsm/bin/configure --ssl \
 --pkey /opt/cloudhsm/etc/ssl-client.key \
 --cert /opt/cloudhsm/etc/ssl-client.crt
```

3. Agregue el certificado `customerCA.crt` al almacén de confianza. Cree un hash del nombre del firmante del certificado. De este modo se crea un índice que permite que busque el certificado por ese nombre.

```
openssl x509 -in /opt/cloudhsm/etc/customerCA.crt -hash | head -n 1
1234abcd
```

   Cree un directorio.

```
mkdir /opt/cloudhsm/etc/certs
```

   Cree un archivo que contenga el certificado con el nombre de hash.

```
sudo cp /opt/cloudhsm/etc/customerCA.crt /opt/cloudhsm/etc/certs/1234abcd.0
```

SSL personalizado para SDK 5 de cliente

Utilice cualquiera de las herramientas de configuración de SDK 5 de cliente para habilitar el SSL personalizado. Para obtener más información sobre la herramienta de configuración para SDK 5 de cliente, consulte ???.

PKCS #11 library

Cómo usar un certificado y una clave personalizados para la autenticación mutua entre cliente y servidor de TLS con SDK 5 de cliente en Linux

1. Copie la clave y el certificado en el directorio adecuado.

```
$ sudo cp ssl-client.crt /opt/cloudhsm/etc
sudo cp ssl-client.key /opt/cloudhsm/etc
```

2. Use la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 \
            --server-client-cert-file /opt/cloudhsm/etc/ssl-client.crt \
            --server-client-key-file /opt/cloudhsm/etc/ssl-client.key
```

Cómo usar un certificado y una clave personalizados para la autenticación mutua TLS cliente-servidor con SDK 5 de cliente en Windows

1. Copie la clave y el certificado en el directorio adecuado.

```
cp ssl-client.crt C:\ProgramData\Amazon\CloudHSM\ssl-client.crt
cp ssl-client.key C:\ProgramData\Amazon\CloudHSM\ssl-client.key
```

2. Con un PowerShell intérprete, utilice la herramienta de configuración para especificar `ssl-client.crt` y`ssl-client.key`.

```
& "C:\Program Files\Amazon\CloudHSM\bin\configure-pkcs11.exe" `
            --server-client-cert-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.crt `
            --server-client-key-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.key
```

OpenSSL Dynamic Engine

Cómo usar un certificado y una clave personalizados para la autenticación mutua entre cliente y servidor de TLS con SDK 5 de cliente en Linux

1. Copie la clave y el certificado en el directorio adecuado.

```
$ sudo cp ssl-client.crt /opt/cloudhsm/etc
sudo cp ssl-client.key /opt/cloudhsm/etc
```

2. Use la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
$ sudo /opt/cloudhsm/bin/configure-dyn \
            --server-client-cert-file /opt/cloudhsm/etc/ssl-client.crt \
            --server-client-key-file /opt/cloudhsm/etc/ssl-client.key
```

JCE provider

Cómo usar un certificado y una clave personalizados para la autenticación mutua entre cliente y servidor de TLS con SDK 5 de cliente en Linux

1. Copie la clave y el certificado en el directorio adecuado.

```
$ sudo cp ssl-client.crt /opt/cloudhsm/etc
sudo cp ssl-client.key /opt/cloudhsm/etc
```

2. Use la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
$ sudo /opt/cloudhsm/bin/configure-jce \
            --server-client-cert-file /opt/cloudhsm/etc/ssl-client.crt \
            --server-client-key-file /opt/cloudhsm/etc/ssl-client.key
```

Cómo usar un certificado y una clave personalizados para la autenticación mutua TLS cliente-servidor con SDK 5 de cliente en Windows

1. Copie la clave y el certificado en el directorio adecuado.

```
cp ssl-client.crt C:\ProgramData\Amazon\CloudHSM\ssl-client.crt
cp ssl-client.key C:\ProgramData\Amazon\CloudHSM\ssl-client.key
```

2. Con un PowerShell intérprete, utilice la herramienta de configuración para especificar `ssl-client.crt` y`ssl-client.key`.

```
& "C:\Program Files\Amazon\CloudHSM\bin\configure-jce.exe" `
            --server-client-cert-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.crt `
            --server-client-key-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.key
```

CloudHSM CLI

Cómo usar un certificado y una clave personalizados para la autenticación mutua entre cliente y servidor de TLS con SDK 5 de cliente en Linux

1. Copie la clave y el certificado en el directorio adecuado.

```
$ sudo cp ssl-client.crt /opt/cloudhsm/etc
sudo cp ssl-client.key /opt/cloudhsm/etc
```

2. Use la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
$ sudo /opt/cloudhsm/bin/configure-cli \
            --server-client-cert-file /opt/cloudhsm/etc/ssl-client.crt \
            --server-client-key-file /opt/cloudhsm/etc/ssl-client.key
```

Cómo usar un certificado y una clave personalizados para la autenticación mutua TLS cliente-servidor con SDK 5 de cliente en Windows

1. Copie la clave y el certificado en el directorio adecuado.

```
cp ssl-client.crt C:\ProgramData\Amazon\CloudHSM\ssl-client.crt
cp ssl-client.key C:\ProgramData\Amazon\CloudHSM\ssl-client.key
```

2. Con un PowerShell intérprete, utilice la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
& "C:\Program Files\Amazon\CloudHSM\bin\configure-cli.exe" `
            --server-client-cert-file C:\ProgramData\Amazon\CloudHSM\ssl-client.crt `
            --server-client-key-file C:\ProgramData\Amazon\CloudHSM\ssl-client.key
```

Creación de una aplicación

Cree aplicaciones y trabaje con claves utilizando AWS CloudHSM.

Para empezar a crear y usar claves en su nuevo clúster, primero debe crear un usuario del módulo de seguridad de hardware (HSM) con la herramienta de Utilidad de administración de CloudHSM (CMU). Para obtener más información, consulte Descripción de las tareas de administración de usuarios de HSM, Introducción a la interfaz de la línea de comandos (CLI) de AWS CloudHSM y Cómo administrar usuarios de HSM.

> **Note**
>
> Si utiliza SDK 3 de cliente, use Utilidad de administración de CloudHSM (CMU) en lugar de la CLI de CloudHSM.

Con los usuarios de HSM instalados, puede iniciar sesión en el HSM y crear y usar claves con cualquiera de las siguientes opciones:

- Uso de Utilidad de administración de claves, una herramienta de línea de comandos
- Creación de una aplicación C con la biblioteca PKCS #11
- Creación de una aplicación Java con el proveedor de JCE
- Utilización del motor dinámico de OpenSSL directamente desde la línea de comandos
- Utilización del motor dinámico de OpenSSL para la descarga de TLS con los servidores web NGINX y Apache
- Use los proveedores de GNC y KSP para usarlos con AWS CloudHSM Microsoft Windows Server Certificate Authority (CA)

- Utilice los proveedores de GNC y KSP para utilizarlos con AWS CloudHSM Microsoft Sign Tool
- Utilización de los proveedores de GNC y KSP para la descarga de TLS con el servidor web Internet Information Server (IIS)

Mejores prácticas para AWS CloudHSM

Siga las prácticas recomendadas de este tema para usar AWS CloudHSM de manera eficaz.

Contenido

- Administración de clústeres
- Administración de usuarios de HMS
- Administración de claves HSM
- Integración de aplicaciones
- Supervisión

Administración de clústeres

Siga las prácticas recomendadas de esta sección al crear el AWS CloudHSM clúster, acceder a él y administrarlo.

Escale su clúster para gestionar los picos de tráfico.

Existen varios factores que pueden influir en el rendimiento máximo que puede gestionar su clúster, tales como el tamaño de la instancia de cliente, el tamaño del clúster, la topografía de la red y las operaciones criptográficas que necesite para su caso de uso.

Como punto de partida, consulte el tema AWS CloudHSM Rendimiento para obtener más información sobre las estimaciones de rendimiento en los tamaños y configuraciones de clúster más comunes. Le recomendamos que realice una prueba de carga de su clúster con la carga máxima prevista para determinar si su arquitectura actual es resiliente y tiene la escala adecuada.

Diseñe su clúster para conseguir una alta disponibilidad.

Añada redundancia para tener en cuenta el mantenimiento: AWS puede sustituir su HSM para un mantenimiento programado o si detecta un problema. Como regla general, el tamaño del clúster debe tener una redundancia de, al menos, +1. Por ejemplo, si necesita dos HSM para que su servicio funcione en las horas punta, el tamaño ideal de su clúster será de tres. Si sigue las prácticas de disponibilidad recomendadas, estas sustituciones de HSM no deberían afectar a su servicio. Sin embargo, es posible que las operaciones en curso en el HSM sustituido fallen. En tal caso, deberán realizarse de nuevo.

Distribuya sus HSM en distintas zonas de disponibilidad: considere cómo funcionará su servicio si se produce una interrupción en la zona de disponibilidad. AWS recomienda que distribuya sus HSM en tantas zonas de disponibilidad como sea posible. En el caso de un clúster con tres HSM, debería distribuir los HSM en tres zonas de disponibilidad. En función de su sistema, es posible que necesite redundancia adicional.

Tenga, al menos, tres HSM para garantizar la durabilidad de las claves recién generadas.

En el caso de las aplicaciones que requieren la durabilidad de las claves recién generadas, le recomendamos al menos tres instancias de HSM distribuidas en todas las zonas de disponibilidad de una región.

Acceso seguro a su clúster

Use subredes privadas para limitar el acceso a su instancia: lance sus HSM e instancias de cliente en las subredes privadas de su VPC. Esto limita el acceso a sus HSM desde el exterior.

Utilice puntos de enlace de VPC para acceder a las API: el plano de AWS CloudHSM datos se diseñó para funcionar sin necesidad de acceder a Internet o a las API de AWS. Si tu instancia de cliente requiere acceso a la AWS CloudHSM API, puedes usar los puntos de enlace de la VPC para acceder a la API sin requerir acceso a Internet en tu instancia de cliente. Para obtener más información, consulte AWS CloudHSM y puntos finales de VPC.

Reconfigura el SSL para proteger la comunicación entre el cliente y el servidor: AWS CloudHSM usa TLS para establecer una conexión con tu HSM. Una vez inicializado el clúster, puede sustituir la clave y el certificado TLS predeterminados que se usan para establecer la conexión TLS externa. Para obtener más información, consulte Mejore la seguridad de su servidor web con la descarga de SSL/TLS en AWS CloudHSM.

Reduzca los costos escalando en función de sus necesidades.

Su uso de AWS CloudHSM no conlleva costos iniciales. Usted paga una tarifa por hora por cada HSM que lance hasta que cancele el HSM. Si su servicio no requiere un uso continuo AWS CloudHSM, puede reducir los costos reduciendo (eliminando) sus HSM a cero cuando no los necesite. Cuando vuelva a necesitar los HSM, podrá restaurarlos a partir de una copia de seguridad. Si, por ejemplo, tiene una carga de trabajo que requiere que firme el código una vez al mes, concretamente el último día del mes, puede escalar su clúster antes, reducirlo verticalmente

eliminando los HSM una vez finalizado el trabajo y, a continuación, restaurar el clúster para realizar de nuevo las operaciones de firma a finales del mes siguiente.

AWS CloudHSM realiza automáticamente copias de seguridad periódicas de los HSM del clúster. Cuando añada un nuevo HSM más adelante, AWS CloudHSM restaurará la última copia de seguridad en el nuevo HSM para que pueda reanudar su uso desde el mismo lugar en el que lo dejó. Para calcular los costes de su AWS CloudHSM arquitectura, consulte AWS CloudHSM los precios.

Recursos relacionados:

- Descripción general de las copias de seguridad
- Política de retención de copias de seguridad
- Copiar copias de seguridad entre AWS regiones

Administración de usuarios de HMS

Siga las prácticas recomendadas de esta sección para gestionar de forma eficaz los usuarios de su AWS CloudHSM clúster. Los usuarios de HSM son distintos de los usuarios de IAM. Los usuarios y las entidades de IAM que tengan una política basada en identidad con los permisos adecuados pueden crear HSM interactuando con los recursos a través de la API de AWS. Una vez creado el HSM, deberá introducir las credenciales de usuario de HSM para autenticar las operaciones del mismo. Para consultar una guía detallada de los usuarios de HSM, acceda a Administrar usuarios de HSM en AWS CloudHSM.

Proteja las credenciales de sus usuarios de HSM.

Es imprescindible que mantenga las credenciales de sus usuarios de HSM protegidas de forma segura, ya que los usuarios de HSM son las entidades que pueden acceder al mismo y realizar operaciones criptográficas y de gestión. AWS CloudHSM no tiene acceso a sus credenciales de usuario de HSM, y no podrá ayudarle si las pierde.

Tenga, al menos, dos administradores para evitar bloqueos.

Para evitar posibles bloqueos de acceso a su clúster, le recomendamos que tenga, al menos, dos administradores, por si se extravía una contraseña de administrador. En caso de que esto suceda, el otro administrador podrá restablecer la contraseña.

> **Note**
>
> Los administradores de SDK 5 de cliente son lo mismo que los responsables de criptografía (CO) en SDK 3 de cliente.

Habilite el cuórum para todas las operaciones de gestión de usuarios.

El cuórum le permite definir el número mínimo de administradores que deben aprobar una operación de gestión de usuarios para que se pueda llevar a cabo dicha operación. Debido a los privilegios que tienen los administradores, le recomendamos que habilite el cuórum para todas las operaciones de gestión de usuarios. Esta medida puede limitar las posibles repercusiones en caso de que una de sus contraseñas de administrador se vea comprometida. Para obtener más información, consulte Administración de cuórum.

Cree varios usuarios de criptografía con permisos limitados.

Al separar las responsabilidades de los usuarios de criptografía, ninguno de ellos puede controlar por completo el sistema. Por esta razón, le recomendamos que cree varios usuarios de criptografía con permisos limitados. Por lo general, deberá asignar responsabilidades y acciones claramente diferenciadas a los usuarios de criptografía. Por ejemplo, uno se encargará de generar y compartir claves con otros usuarios para que estos las usen en su aplicación.

Recursos relacionados:

- key share
- key unshare

Administración de claves HSM

Siga las prácticas recomendadas de esta sección para gestionar claves en AWS CloudHSM.

Elija el tipo de clave correcto.

Cuando use una clave de sesión, sus transacciones por segundo (TPS) se limitarán a un HSM donde exista la clave. Los HSM adicionales en el clúster no aumentarán el rendimiento de solicitudes de esa clave. Si usa una clave de token para la misma aplicación, se equilibrará la carga de sus

solicitudes entre todos los HSM disponibles en su clúster. Para obtener más información, consulte Ajustes clave de sincronización y durabilidad en AWS CloudHSM.

Gestione los límites de almacenamiento de claves.

El número máximo de claves de sesión y de token que se pueden almacenar a la vez en un HSM es limitado. Para obtener información sobre los límites de almacenamiento de claves, consulte AWS CloudHSM cuotas. Si su aplicación requiere una cantidad superior al límite, puede emplear una o varias de las siguientes estrategias para gestionar las claves de forma eficaz:

Use un empaquetado fiable para almacenar las claves en un almacén de datos externo: si emplea un encapsulado de claves fiable, puede superar el límite de almacenamiento de claves guardándolas todas en un almacén de datos externo. Cuando tenga que usar una clave, puede desencapsularla en el HSM como clave de sesión, usarla para la operación requerida y, a continuación, descartar la clave de sesión. Los datos de la clave original permanecerán almacenados de forma segura en su almacén de datos para usarla siempre que la necesite. El uso de claves fiables maximiza su protección.

Distribuya las claves entre los clústeres: otra estrategia para superar el límite de almacenamiento de claves consiste en almacenar las claves en múltiples clústeres. Con este método, deberá mapear las claves almacenadas en cada clúster. Utilice este mapeo para dirigir las solicitudes de sus clientes al clúster con la clave requerida. Para saber cómo conectarse a varios clústeres desde una misma aplicación de cliente, consulte los siguientes temas:

- Conexión a varios clústeres con el proveedor de JCE
- Conexión a varias ranuras con PKCS#11

Gestionar y proteger el empaquetado de claves.

Las claves pueden marcarse como extraíbles o no extraíbles mediante el atributo `EXTRACTABLE`. De forma predeterminada, las claves de HSM se marcan como extraíbles.

Las claves extraíbles son aquellas que se pueden exportar desde el HSM mediante encapsulado de claves. Las claves empaquetadas se cifran, y se deben desencapsular con la misma clave de encapsulado para poder utilizarlas. Las claves no extraíbles no se pueden exportar desde el HSM bajo ninguna circunstancia. No es posible convertir una clave no extraíble en extraíble. Por ello, es importante que considere la necesidad de que sus claves sean o no extraíbles para definir el correspondiente atributo en consecuencia.

Si necesita empaquetar claves en su aplicación, deberá emplear un encapsulado de claves fiable, que permita limitar la capacidad de los usuarios de HSM. Así, estos solo podrán encapsular y desencapsular las claves que un administrador haya marcado explícitamente como de confianza. Para obtener más información, consulte los temas sobre encapsulado de claves de confianza en Administrar claves en AWS CloudHSM.

Recursos relacionados

- Funciones de encapsulado y desencapsulado
- Funciones de cifrado para JCE
- Atributos Java admitidos
- Atributos de clave de la CLI de CloudHSM

Integración de aplicaciones

Siga las prácticas recomendadas de esta sección para optimizar la forma en que la aplicación se integra con el AWS CloudHSM clúster.

Inicie su SDK de cliente.

Debe iniciar su SDK de cliente antes de conectarlo a su clúster. Al iniciar las direcciones IP en el clúster, le recomendamos que use el parámetro `--cluster-id` siempre que sea posible. Este método rellena la configuración con todas las direcciones IP de HSM de su clúster sin necesidad de supervisar cada dirección individual. De este modo, se añade una mayor resiliencia a la inicialización de la aplicación en caso de que un HSM se encuentre en mantenimiento o se produzca una interrupción en la zona de disponibilidad. Para obtener más información, consulte Proceso de arranque del SDK de cliente.

Autentíquese para realizar operaciones.

En AWS CloudHSM, debe autenticarse en el clúster antes de poder realizar la mayoría de las operaciones, como las operaciones criptográficas.

Autenticación con la CLI de CloudHSM: puede autenticarse con la CLI de CloudHSM usando el modo de comando único o bien el modo interactivo. Ejecute el comando login para autenticarse en modo interactivo. Para autenticarse en modo de comando único, debe configurar las variables de entorno `CLOUDHSM_ROLE` y `CLOUDHSM_PIN`. Para obtener más información al respecto, consulte Modo de

comando único. AWS CloudHSM recomienda almacenar de forma segura sus credenciales de HSM cuando no las use su aplicación.

Autenticarse con PKCS #11: en PKCS #11, inicia sesión con la API C_Login después de abrir una sesión con C_. OpenSession Solo necesita ejecutar un C_Login por ranura (clúster). Una vez que haya iniciado sesión correctamente, podrá abrir sesiones adicionales con C_ OpenSession sin necesidad de realizar operaciones de inicio de sesión adicionales. Para ver ejemplos de autenticación en PKCS #11, consulte Ejemplos de código para la biblioteca PKCS #11.

Autenticarse con JCE: el proveedor de AWS CloudHSM JCE admite el inicio de sesión tanto implícito como explícito. El método que más le convenga del caso de uso. Siempre que sea posible, recomendamos usar el inicio de sesión implícito, ya que el SDK gestionará automáticamente la autenticación si la aplicación se desconecta de su clúster y necesita volver a autenticarse. El inicio de sesión implícito también le permite proporcionar credenciales a su aplicación cuando use una integración que no le permita controlar el código de la aplicación. Para obtener más información sobre los métodos de inicio de sesión, consulte Proporcione las credenciales al proveedor de JCE..

Autenticación con OpenSSL: con el motor dinámico de OpenSSL, las credenciales se proporcionan a través de variables de entorno. AWS CloudHSM recomienda almacenar de forma segura sus credenciales de HSM cuando la aplicación no las utilice. Si es posible, debe configurar su entorno para recuperar y configurar sistemáticamente estas variables de entorno sin necesidad de introducirlas manualmente. Para obtener más información sobre la autenticación con OpenSSL, consulte Instalación del motor dinámico de OpenSSL.

Gestione eficazmente las claves de su aplicación.

Use los atributos de clave para controlar lo que pueden hacer las claves: al generar una clave, use los atributos de clave para definir un conjunto de permisos que permitan o denieguen a esa clave realizar tipos específicos de operaciones. Recomendamos que las claves se generen con la cantidad mínima de atributos necesarios para completar la tarea. Por ejemplo, una clave AES que se usa para el cifrado no debería tener permisos para encapsular claves fuera del HSM. Para obtener más información, consulte nuestras páginas de atributos de los siguientes SDK de cliente:

- Atributos de clave de PKCS #11
- Atributos de clave JCE

Cuando sea posible, almacene en caché los objetos de clave para minimizar la latencia: las operaciones de búsqueda de claves consultarán todos los HSM del clúster. Esta operación es costosa, y no escala según el número de HSM de su clúster.

- Con PKCS #11, puede usar la API de `C_FindObjects` para encontrar claves.
- Con JCE, las claves se encuentran mediante. KeyStore

Para obtener un rendimiento óptimo, se AWS recomienda utilizar los comandos de búsqueda de teclas (como findKey ykey list) solo una vez durante el inicio de la aplicación y almacenar en caché el objeto clave devuelto en la memoria de la aplicación. Si necesita este objeto de clave más adelante, podrá recuperarlo de la memoria caché en lugar de consultarlo en cada operación, lo que aumentará considerablemente el rendimiento.

Emplee subprocesamiento múltiple.

AWS CloudHSM admite aplicaciones con varios subprocesos, pero hay ciertas cosas que se deben tener en cuenta con las aplicaciones con varios subprocesos.

Con PKCS #11, deberá inicializar la biblioteca PKCS #11 (llamando a `C_Initialize`) una sola vez. Debe asignar a cada proceso su propia sesión (`C_OpenSession`). No es recomendable usar la misma sesión en múltiples procesos.

Con JCE, el AWS CloudHSM proveedor debe inicializarse solo una vez. No comparta instancias de objetos SPI entre procesos. Por ejemplo, Cipher, Signature, Digest, Mac KeyFactory o KeyGenerator los objetos solo deben utilizarse en el contexto de su propio hilo.

Gestione los errores de limitación.

Es posible que se produzcan errores de limitación del HSM en las siguientes circunstancias:

- El clúster no está escalado correctamente para gestionar los picos de tráfico.
- El tamaño del clúster no tiene una redundancia de +1 durante los eventos de mantenimiento.
- Las interrupciones en la zona de disponibilidad reducen el número de HSM disponibles en el clúster.

Consulte Limitación de HSM para obtener más información sobre la mejor manera de gestionar este escenario.

Para asegurarse de que su clúster tiene el tamaño adecuado y no se agotará, le AWS recomienda que realice una prueba de carga en su entorno con los picos de tráfico esperados.

Integre los reintentos en las operaciones del clúster.

AWS puede reemplazar su HSM por motivos operativos o de mantenimiento. Para que su aplicación sea resistente a estas situaciones, le AWS recomienda implementar una lógica de reintentos del lado del cliente en todas las operaciones que se enruten a su clúster. Es de esperar que los posteriores reintentos de las operaciones fallidas debido a las sustituciones se realicen correctamente.

Implemente estrategias de recuperación de desastres.

Puede que sea necesario desviar el tráfico de todo un clúster o región en respuesta a un evento. En las siguientes secciones se describen varias estrategias para ello.

Use el emparejamiento de VPC para acceder a su clúster desde otra cuenta o región: puede utilizar el emparejamiento de VPC para acceder a su AWS CloudHSM clúster desde otra cuenta o región. Para obtener más información sobre la configuración, consulte ¿Qué es una conexión de emparejamiento de VPC? en la Guía de emparejamiento de VPC de Amazon. Una vez haya establecido las conexiones de emparejamiento y configurado los grupos de seguridad de forma adecuada, podrá comunicarse con las direcciones IP de los HSM de la misma manera que lo haría normalmente.

Conéctese a varios clústeres desde la misma aplicación: el proveedor JCE, la biblioteca PKCS #11 y la CLI del Client SDK 5 admiten la conexión a varios clústeres desde la misma aplicación. Por ejemplo, puede tener dos clústeres activos, cada uno en una región distinta. Su aplicación puede conectarse a ambos a la vez y equilibrar la carga entre los dos como parte de su operativa normal. Si su aplicación no usa SDK 5 de cliente (el SDK más reciente), no podrá conectarse a varios clústeres desde una misma aplicación. Como alternativa, puede mantener otro clúster en funcionamiento y, en caso de que se produzca una interrupción regional, transferir el tráfico al otro clúster para minimizar el tiempo de inactividad. Consulte las páginas correspondientes para obtener más información:

- Conexión a varias ranuras con PKCS#11
- Conexión a varios clústeres con el proveedor de JCE
- Conexión a varios clústeres con CLI

Restaure un clúster a partir de una copia de seguridad: puede crear un clúster nuevo a partir de una copia de seguridad de un clúster existente. Para obtener más información, consulte Administración de AWS CloudHSM copias de seguridad.

Supervisión

En esta sección se describen varios mecanismos con los que puede supervisar el clúster y la aplicación. Para obtener información adicional sobre la supervisión, consulte Monitorización AWS CloudHSM.

Supervisión de registros de clientes

SDK de cliente genera registros que usted puede supervisar. Para obtener más información sobre los registros de cliente, consulte Trabajo con los registros de SDK de cliente.

En plataformas diseñadas para ser efímeras, como Amazon ECS AWS Lambda, recopilar los registros de los clientes a partir de un archivo puede resultar difícil. En estas situaciones, se recomienda configurar el registro de SDK de cliente para generar los registros en la consola. La mayoría de los servicios recopilarán automáticamente este resultado y lo publicarán en Amazon CloudWatch Logs para que lo guardes y lo veas.

Si utiliza una integración de terceros además del SDK de AWS CloudHSM cliente, asegúrese de configurar ese paquete de software para que registre también su salida en la consola. Este paquete puede capturar el resultado del SDK del AWS CloudHSM cliente y, de lo contrario, escribirlo en su propio archivo de registro.

Consulte Herramienta de configuración de SDK 5 de cliente para obtener más información sobre cómo configurar las opciones de registro en su aplicación.

Monitoreo de registros de auditoría

AWS CloudHSM publica registros de auditoría en tu CloudWatch cuenta de Amazon. Los registros de auditoría provienen del HSM, y supervisan determinadas operaciones con fines de auditoría.

Puede usar los registros de auditoría para supervisar cualquier comando de administración invocado en su HSM. Por ejemplo, puede activar una alarma cuando detecte que se está realizando una operación de administración inesperada.

Consulte Cómo funciona el registro de auditoría de HSM para obtener más detalles.

Supervise AWS CloudTrail

AWS CloudHSM está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en AWS CloudHSM. AWS CloudTrail captura todas las llamadas a la API AWS CloudHSM como eventos. Las llamadas capturadas incluyen llamadas desde la AWS CloudHSM consola y llamadas en código a las operaciones de la AWS CloudHSM API.

Puede utilizarla AWS CloudTrail para auditar cualquier llamada a la API que se realice en el plano de AWS CloudHSM control para asegurarse de que no se esté produciendo ninguna actividad no deseada en su cuenta.

Para obtener más información, consulte Trabajar con AWS CloudTrail y AWS CloudHSM.

Supervisa CloudWatch las métricas de Amazon

Puedes usar CloudWatch las métricas de Amazon para monitorear tu AWS CloudHSM clúster en tiempo real. Las métricas se pueden agrupar por región, por ID de clúster y por ID de HSM.

Con CloudWatch las métricas de Amazon, puedes configurar CloudWatch las alarmas de Amazon para que te avisen de cualquier posible problema que pueda surgir y que pueda afectar a tu servicio. Recomendamos configurar alarmas para supervisar los siguientes aspectos:

- Aproximación al límite de claves de un HSM
- Aproximación al límite de número de sesiones de HSM en un HSM
- Aproximación al límite de número de usuarios de HSM en un HSM
- Diferencias en el recuento de claves o usuarios del HSM para identificar problemas de sincronización
- Los HSM en mal estado pueden escalar el clúster hasta que AWS CloudHSM puedan resolver el problema

Para obtener más información, consulte Trabajar con Amazon CloudWatch Logs y AWS CloudHSM Audit Logs.

Administrar AWS CloudHSM clústeres

Puede administrar AWS CloudHSM los clústeres desde la AWS CloudHSM consola o desde uno de los AWS SDK o herramientas de línea de comandos. Para obtener más información, consulte los siguientes temas.

Para crear un clúster, consulte Introducción.

Arquitectura de clúster

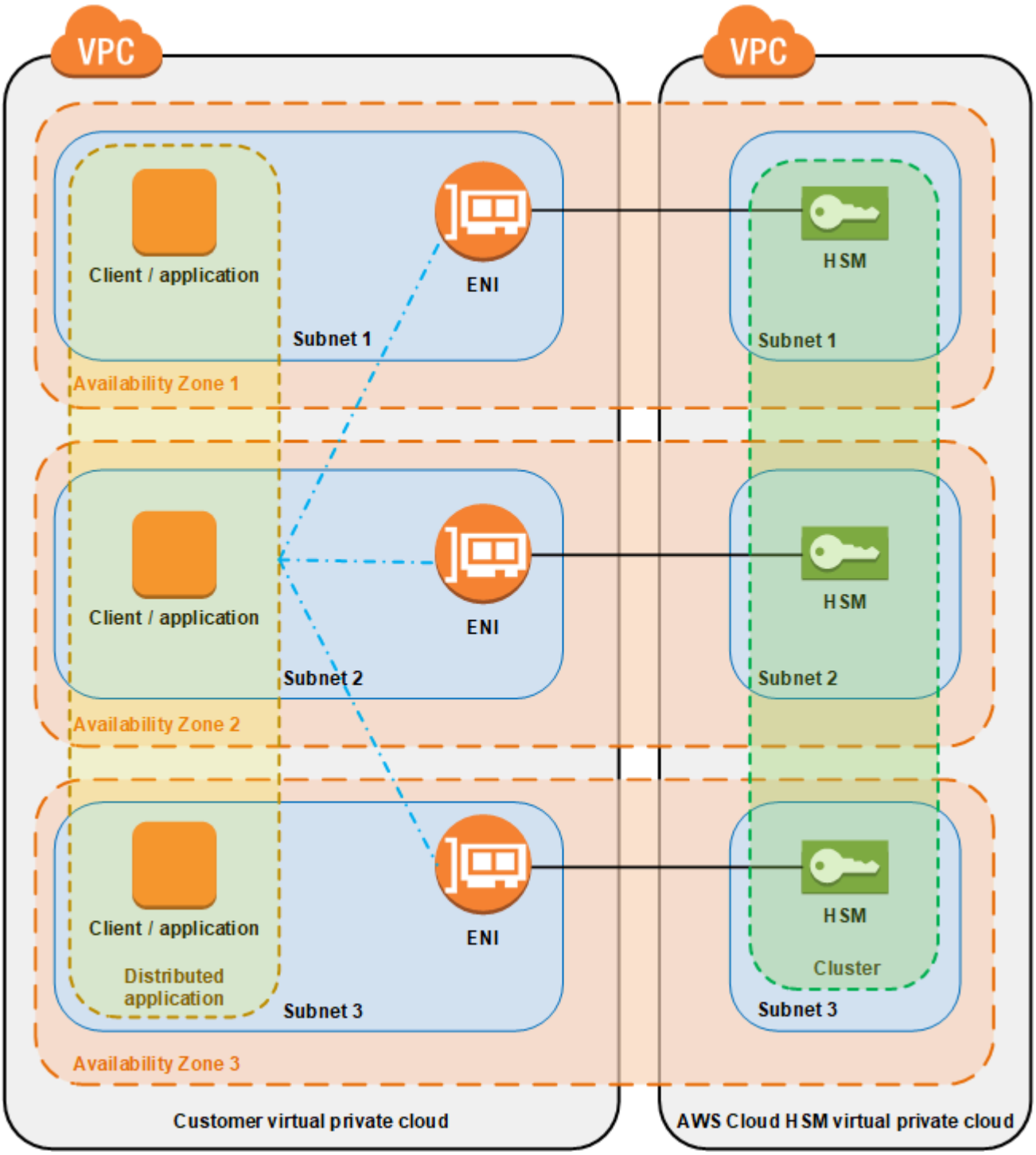
Al crear un clúster, debe especificar una Amazon Virtual Private Cloud (VPC) en su AWS cuenta y una o más subredes en esa VPC. Le recomendamos que cree una subred en cada zona de disponibilidad (AZ) de la región que elija. AWS Puede crear subredes privadas al crear una VPC. Para obtener más información, consulte Cree una nube privada virtual (VPC)..

Cada vez que crea un HSM, debe especificar el clúster y la zona de disponibilidad del HSM. Al colocar los HSM en diferentes zonas de disponibilidad, obtiene redundancia y alta disponibilidad en el caso de que una zona de disponibilidad no esté disponible.

Al crear un HSM, AWS CloudHSM coloca una interface de red elástica (ENI) en la subred especificada de su AWS cuenta. La interfaz de red elástica es la que se usa para interactuar con el HSM. El HSM reside en una VPC independiente en AWS una cuenta que es propiedad de. AWS CloudHSM El HSM y su interfaz de red correspondiente se encuentran en la misma zona de disponibilidad.

Para interactuar con los HSM de un clúster, necesita el AWS CloudHSM software de cliente. Por lo general, se instala al cliente en instancias de Amazon EC2 denominadas instancias de cliente, que residen en la misma VPC que los ENI de los HSM, tal y como se muestra en la siguiente ilustración. Esto no es necesario desde el punto de vista técnico; puede instalar el cliente en cualquier equipo compatible, siempre que pueda conectarse a las ENI de los HSM. El cliente se comunica con cada uno de los HSM del clúster a través de sus ENI.

La siguiente figura representa un AWS CloudHSM clúster con tres HSM, cada uno en una zona de disponibilidad diferente de la VPC.



Sincronización de clúster

En un AWS CloudHSM clúster, AWS CloudHSM mantiene sincronizadas las claves de los HSM individuales. No tiene que hacer nada para sincronizar las claves en los HSM. Para mantener sincronizados los usuarios y las políticas de cada HSM, actualice el archivo de configuración

del AWS CloudHSM cliente antes de administrar los usuarios de los HSM. Para obtener más información, consulte Mantener sincronizados los usuarios de HSM.

Al añadir un HSM nuevo a un clúster, AWS CloudHSM hace una copia de seguridad de todas las claves, usuarios y políticas de un HSM existente. A continuación, restaura ese backup en el nuevo HSM. De este modo, los dos HSM permanecen sincronizados.

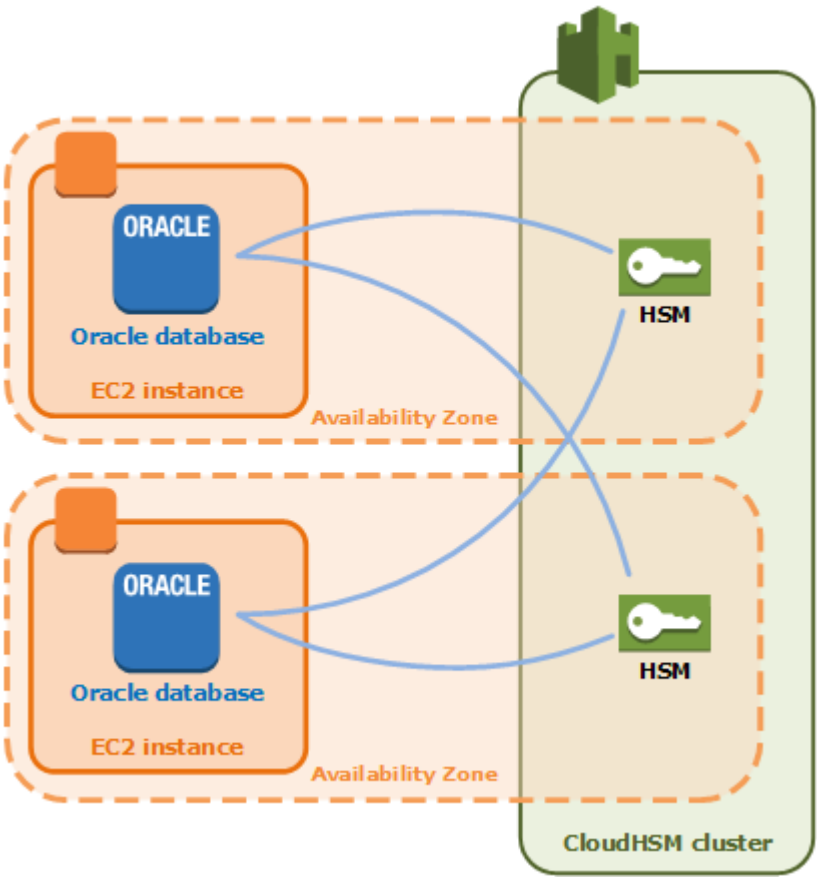
Si los HSM de un clúster no están sincronizados, los resincroniza AWS CloudHSM automáticamente. Para habilitarlo, AWS CloudHSM utiliza las credenciales del usuario del dispositivo. Este usuario existe en todos los HSM proporcionados por AWS CloudHSM y tiene permisos limitados. Puede obtener un hash de los objetos del HSM, así como insertar y extraer objetos enmascarados (cifrados). AWS no puede ver ni modificar los usuarios ni las claves y no puede realizar operaciones criptográficas utilizando estas claves.

Alta disponibilidad y balanceo de carga del clúster

Al crear un AWS CloudHSM clúster con más de un HSM, se obtiene automáticamente el equilibrio de carga. El balanceo de carga significa que el cliente de AWS CloudHSM distribuye las operaciones criptográficas entre todos los HSM del clúster en función de la capacidad de procesamiento adicional de cada uno de ellos.

Al crear los HSM en distintas zonas de AWS disponibilidad, se obtiene automáticamente una alta disponibilidad. Alta disponibilidad significa que obtiene mayor fiabilidad porque ningún HSM es un punto único de error. Se recomienda tener un mínimo de dos HSM en cada clúster, y que cada HSM se encuentre en distintas zonas de disponibilidad dentro de una región. AWS

Por ejemplo, en la figura siguiente se muestra una aplicación de base de datos Oracle que está distribuida en dos zonas de disponibilidad. Las instancias de base de datos almacenan sus claves maestras en un clúster que incluye un HSM en cada zona de disponibilidad. AWS CloudHSM sincroniza automáticamente las claves con ambos HSM para que sean inmediatamente accesibles y redundantes.



Conecta el SDK del cliente al AWS CloudHSM clúster

Para conectarse al clúster con el SDK 5 de cliente o el SDK 3 de cliente, primero debe hacer dos cosas:

- Contar con un certificado de emisión en la instancia EC2
- Inicio del arranque del SDK del cliente en el clúster

Colocación del certificado de emisión en cada instancia de EC2

Crea el certificado de emisión al inicializar el clúster. Copie el certificado de emisión en la ubicación predeterminada de la plataforma en cada instancia EC2 que se conecte al clúster.

Linux

```
/opt/cloudhsm/etc/customerCA.crt
```

Windows

```
C:\ProgramData\Amazon\CloudHSM\customerCA.crt
```

Especifique la ubicación del certificado de emisión.

Con SDK 5 de cliente, se utiliza la herramienta de configuración para especificar la ubicación del certificado de emisión.

PKCS #11 library

Cómo ubicar el certificado de emisión en Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 --hsm-ca-cert <customerCA certificate
 file>
```

Cómo ubicar el certificado de emisión en Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
"C:\Program Files\Amazon\CloudHSM\configure-pkcs11.exe" --hsm-ca-
cert <customerCA certificate file>
```

OpenSSL Dynamic Engine

Cómo ubicar el certificado de emisión en Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
$ sudo /opt/cloudhsm/bin/configure-dyn --hsm-ca-cert <customerCA certificate
 file>
```

JCE provider

Cómo ubicar el certificado de emisión en Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
$ sudo /opt/cloudhsm/bin/configure-jce --hsm-ca-cert <customerCA certificate
 file>
```

Cómo ubicar el certificado de emisión en Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
"C:\Program Files\Amazon\CloudHSM\configure-jce.exe" --hsm-ca-cert <customerCA
 certificate file>
```

CloudHSM CLI

Cómo ubicar el certificado de emisión en Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
$ sudo /opt/cloudhsm/bin/configure-cli --hsm-ca-cert <customerCA certificate
 file>
```

Cómo ubicar el certificado de emisión en Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
"C:\Program Files\Amazon\CloudHSM\configure-cli.exe" --hsm-ca-cert <customerCA
 certificate file>
```

Para obtener más información, consulte Herramienta de configuración.

Para obtener más información sobre la inicialización del clúster o la creación y firma del certificado, consulte Inicializar el clúster.

Proceso de arranque del SDK de cliente

El proceso de arranque es diferente según la versión del SDK de cliente que utilice, pero debe tener la dirección IP de uno de los módulos de seguridad de hardware (HSM) del clúster. Puede usar la dirección IP de cualquier HSM adjuntado al clúster. Una vez que el SDK de cliente se conecta, recupera las direcciones IP de cualquier HSM adicional y realiza las operaciones de equilibrio de carga y sincronización de claves del lado del cliente.

Cómo obtener una dirección IP para el clúster

Para obtener una dirección IP para un HSM (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Para cambiar la región de AWS, utilice el selector de regiones en la esquina superior derecha de la página.
3. Para abrir la página de detalles del clúster, en la tabla de clústeres, elija el ID del clúster.
4. Para obtener la dirección IP, vaya a la pestaña HSM y elija una de las direcciones IP que aparecen en la lista Dirección IP de ENI.

Para obtener una dirección IP para un HSM (CLI)

- Obtenga la dirección IP de un HSM mediante el describe-clusters comando de la CLI. En el resultado del comando, la dirección IP de los HSM son los valores de `EniIp`.

```
$  aws cloudhsmv2 describe-clusters


{
    "Clusters": [
        { ... }
            "Hsms": [
                {
...
                    "EniIp": "10.0.0.9",
...
                },
                {
...
                    "EniIp": "10.0.1.6",
...
```

Para obtener más información sobre las acciones de arranque, consulte la Herramienta de configuración.

Arranque de SDK 5 de cliente

PKCS #11 library

Arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM del clúster.

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 -a <HSM IP addresses>
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM del clúster.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-pkcs11.exe" -a <HSM IP
 addresses>
```

OpenSSL Dynamic Engine

Arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM del clúster.

```
$ sudo /opt/cloudhsm/bin/configure-dyn -a <HSM IP addresses>
```

JCE provider

Arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM del clúster.

```
$ sudo /opt/cloudhsm/bin/configure-jce -a <HSM IP addresses>
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM del clúster.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-jce.exe" -a <HSM IP addresses>
```

CloudHSM CLI

Arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de los HSM de su clúster.

```
$ sudo /opt/cloudhsm/bin/configure-cli -a <The ENI IP addresses of the HSMs>
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de los HSM de su clúster.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-cli.exe" -a <The ENI IP
 addresses of the HSMs>
```

> **Note**
>
> puede usar el parámetro `--cluster-id` en lugar de `-a <HSM_IP_ADDRESSES>`. Para ver los requisitos de uso de `--cluster-id`, consulte Herramienta de configuración de SDK 5 de cliente.

Arranque de SDK 3 de cliente

Arranque de una instancia EC2 de Linux para SDK 3 de cliente

- Se utiliza configure para especificar la dirección IP de un HSM del clúster.

```
sudo /opt/cloudhsm/bin/configure -a <IP address>
```

Arranque de una instancia EC2 de Windows para SDK 3 de cliente

- Se utiliza configure para especificar la dirección IP de un HSM del clúster.

```
C:\Program Files\Amazon\CloudHSM\bin\configure-jce.exe -a <HSM IP address>
```

Para obtener más información sobre la configuración, consulte ???.

Añadir o eliminar los HSM de un clúster AWS CloudHSM

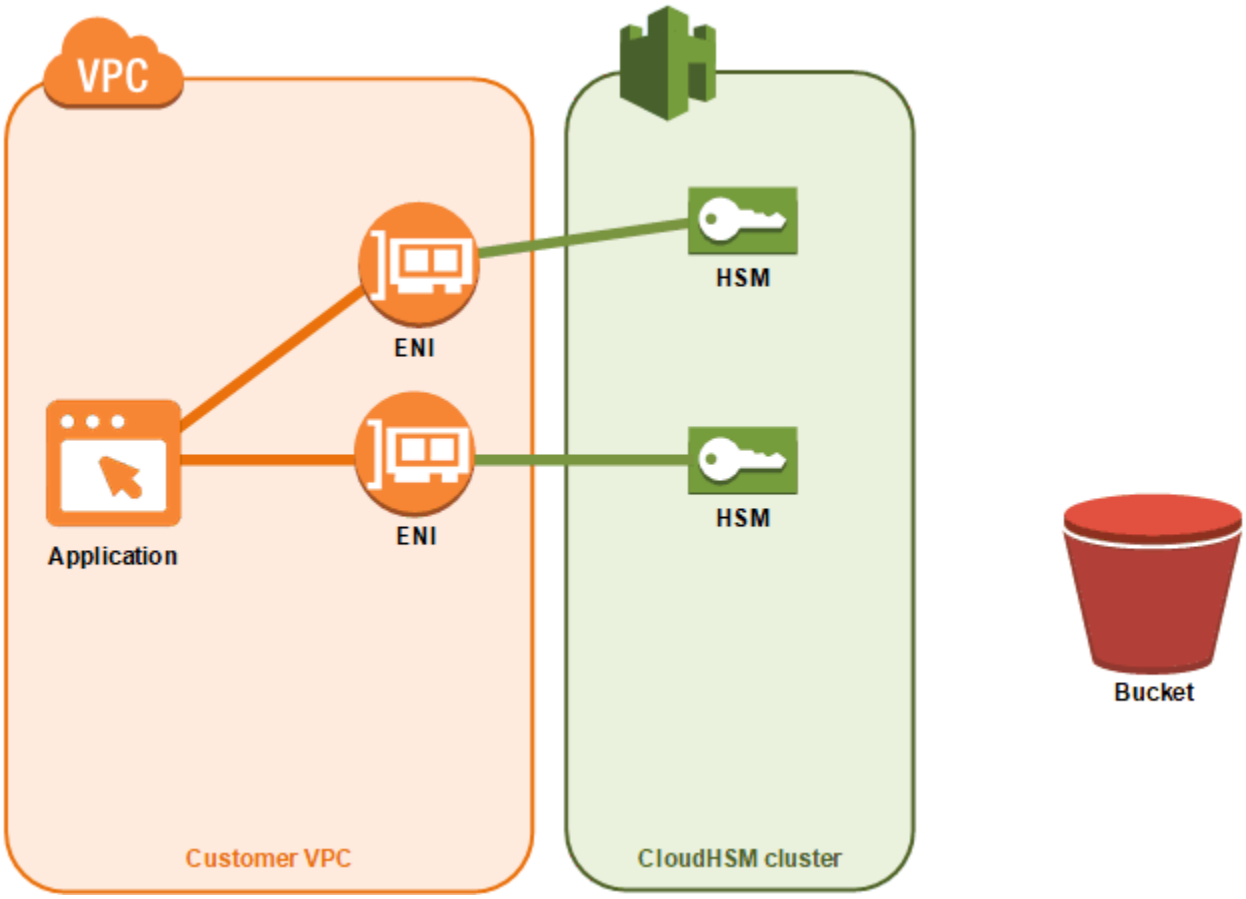
Para ampliar o reducir el tamaño del AWS CloudHSM clúster, añada o elimine los HSM mediante la AWS CloudHSM consola o uno de los AWS SDK o herramientas de línea de comandos. Recomendamos realizar pruebas de carga en el clúster para determinar el pico de carga previsto y, a continuación, añadir un HSM adicional para garantizar una alta disponibilidad.

Temas

- Agregar un HSM
- Eliminación de un HSM

Agregar un HSM

En la figura siguiente se muestran los eventos que se producen cuando se añade un HSM a un clúster.



1. Añada un HSM nuevo a un clúster. Los siguientes procedimientos explican cómo hacerlo desde la AWS CloudHSM consola, la AWS Command Line Interface (CLI) y la AWS CloudHSM API.

   Esta es la única acción que lleva a cabo. Los demás eventos se realizan de forma automática.
2. AWS CloudHSM hace una copia de seguridad de un HSM existente en el clúster. Para obtener más información, consulte Copias de seguridad.
3. AWS CloudHSM restaura la copia de seguridad en el nuevo HSM. Esto garantiza que el HSM esté sincronizado con los demás HSM del clúster.
4. Los HSM existentes en el clúster notifican al AWS CloudHSM cliente que hay un nuevo HSM en el clúster.
5. El cliente establece una conexión con el HSM nuevo.

Para añadir un HSM (consola)

1. Abre la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Elija un clúster para el HSM que va a añadir.

3. En la pestaña HSMs, elija Create HSM (Crear HSM).
4. Elija una zona de disponibilidad (AZ) para el HSM que está creando. A continuación, seleccione Crear.

Para agregar un HSM (CLI)

- En el símbolo del sistema, ejecute el comando create-hsm especificando un ID de clúster y una zona de disponibilidad para el HSM que va a crear. Si no sabe cuál es el ID de su clúster preferido, ejecute el comando describe-clusters. Especifique la zona de disponibilidad con el formato `us-east-2a`, `us-east-2b`, etc.

```
$ aws cloudhsmv2 create-hsm --cluster-id <cluster ID> --availability-
zone <Availability Zone>
{
    "Hsm": {
        "State": "CREATE_IN_PROGRESS",
        "ClusterId": "cluster-5a73d5qzrdh",
        "HsmId": "hsm-lgavqitns2a",
        "SubnetId": "subnet-0e358c43",
        "AvailabilityZone": "us-east-2c",
        "EniId": "eni-bab18892",
        "EniIp": "10.0.3.10"
    }
}
```

Para añadir un HSM (API)AWS CloudHSM

- Envíe una solicitud CreateHsm en la que se especifique el ID del clúster y una zona de disponibilidad para el HSM que está creando.

Eliminación de un HSM

Puede eliminar un HSM mediante la AWS CloudHSM consola, la CLI o la AWS CloudHSM API.

Para eliminar un HSM (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Elija el clúster que contiene el HSM que va a eliminar.

3. En la pestaña HSMs elija el HSM que va a eliminar. A continuación, elija Delete HSM (Eliminar HSM).
4. Confirme que desea eliminar el HSM. A continuación, elija Eliminar.

Para eliminar un HSM (CLI)

- En el símbolo del sistema, ejecute el comando delete-hsm. Pase el ID del clúster que contiene el HSM que está eliminando y uno de los siguientes identificadores de HSM:
  - El ID del HSM (`--hsm-id`)
  - La dirección IP del HSM (`--eni-ip`)
  - El ID de interfaz de red elástica del HSM (`--eni-id`)

  Si no sabe cuáles son los valores de estos identificadores, ejecute el comando describe-clusters.

```
$ aws cloudhsmv2 delete-hsm --cluster-id <cluster ID> --eni-ip <HSM IP address>
{
    "HsmId": "hsm-lgavqitns2a"
}
```

Para eliminar un HSM (API)AWS CloudHSM

- Envíe una solicitud DeleteHsm en la que especifique el ID del clúster y un identificador del HSM que está eliminando.

Eliminar un AWS CloudHSM clúster

Para eliminar un clúster, antes debe eliminar todos los HSM del clúster. Para obtener más información, consulte Eliminación de un HSM.

Después de eliminar todos los HSM, puede eliminar un clúster mediante la AWS CloudHSM consola, la AWS Command Line Interface (CLI) o la AWS CloudHSM API.

Para eliminar un clúster (consola)

1. Abre la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Elija el clúster que está eliminado. A continuación, elija Delete cluster (Eliminar clúster).

3. Confirme que desea eliminar el clúster y, a continuación, elija Delete (Eliminar).

Para eliminar un clúster (CLI)

- En el símbolo del sistema ejecute el comando delete-cluster, pasándole el ID del clúster que está eliminando. Si no sabe cuál es el ID del clúster, ejecute el comando describe-clusters.

```
$ aws cloudhsmv2 delete-cluster --cluster-id <cluster ID>
{
    "Cluster": {
        "Certificates": {
            "ClusterCertificate": "<certificate string>"
        },
        "SourceBackupId": "backup-rtq2dwi2gq6",
        "SecurityGroup": "sg-40399d28",
        "CreateTimestamp": 1504903546.035,
        "SubnetMapping": {
            "us-east-2a": "subnet-f1d6e798",
            "us-east-2c": "subnet-0e358c43",
            "us-east-2b": "subnet-40ed9d3b"
        },
        "ClusterId": "cluster-kdmrayrc7gi",
        "VpcId": "vpc-641d3c0d",
        "State": "DELETE_IN_PROGRESS",
        "HsmType": "hsm1.medium",
        "StateMessage": "The cluster is being deleted.",
        "Hsms": [],
        "BackupPolicy": "DEFAULT"
    }
}
```

Para eliminar un clúster de AWS CloudHSM (API)

- Envíe una solicitud DeleteCluster en la que se especifique el ID del clúster que está eliminando.

Crear AWS CloudHSM clústeres a partir de copias de seguridad

Para restaurar un AWS CloudHSM clúster a partir de una copia de seguridad, siga los pasos de este tema. Su clúster contendrá los mismos usuarios, material de claves, certificados, configuración y

políticas que tenía la copia de seguridad. Para obtener más información acerca de cómo administrar copias de seguridad, consulte Administración de copias de seguridad.

Cree clústeres a partir de copias de seguridad (consola)

1. Abre la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Elija Create cluster.
3. En la sección Cluster configuration, haga lo siguiente:
   a. En VPC elija una VPC para el clúster que está creando.
   b. En AZ (s) elija una subred privada para cada zona de disponibilidad que esté añadiendo al clúster.
4. En la sección Cluster source (Origen del clúster) haga lo siguiente:
   a. Elija Restore cluster from existing backup (Restaurar clúster a partir de una copia de seguridad ya existente).
   b. Elija la copia de seguridad que está restaurando.
5. Elija Siguiente: Revisar.
6. Revise la configuración del clúster y, a continuación, elija Create cluster (Crear clúster).
7. Especifique durante cuánto tiempo el servicio debe retener las copias de seguridad.

   Acepte el período de retención predeterminado de 90 días o escriba un nuevo valor de entre 7 y 379 días. El servicio eliminará automáticamente las copias de seguridad de este clúster que sean anteriores al valor que especifique aquí. Puede cambiar este valor posteriormente. Para obtener más información, consulte Configuración de retención de copias de seguridad.
8. Seleccione Siguiente.
9. (Opcional) Escriba una clave de etiqueta y un valor de etiqueta opcional. Para agregar más de una etiqueta al clúster, elija Agregar etiqueta.
10. Elija Revisar.
11. Revise la configuración del clúster y, a continuación, elija Crear clúster.

> **Tip**
>
> Para crear un HSM en este clúster que contenga los mismos usuarios, material clave, certificados, configuración y políticas que estaban en la copia de seguridad que restauró, añada un HSM al clúster.

Crear clústeres a partir de copias de seguridad (CLI)

Para determinar el ID de la copia de seguridad, emita el comando describe-backups.

- En el símbolo del sistema, ejecute el comando create-cluster. Especifique el tipo de instancia de HSM, los ID de las subredes donde piensa crear los HSM y el ID de la copia de seguridad que está restaurando.

```
$ aws cloudhsmv2 create-cluster --hsm-type hsm1.medium \
                                --subnet-ids <subnet ID 1> <subnet ID 2> <subnet ID
 N> \
                                --source-backup-id <backup ID>
{
    "Cluster": {
        "HsmType": "hsm1.medium",
        "VpcId": "vpc-641d3c0d",
        "Hsms": [],
        "State": "CREATE_IN_PROGRESS",
        "SourceBackupId": "backup-rtq2dwi2gq6",
        "BackupPolicy": "DEFAULT",
        "BackupRetentionPolicy": {
            "Type": "DAYS",
            "Value": 90
         },
        "SecurityGroup": "sg-640fab0c",
        "CreateTimestamp": 1504907311.112,
        "SubnetMapping": {
            "us-east-2c": "subnet-0e358c43",
            "us-east-2a": "subnet-f1d6e798",
            "us-east-2b": "subnet-40ed9d3b"
        },
        "Certificates": {
            "ClusterCertificate": "<certificate string>"
        },
        "ClusterId": "cluster-jxhlf7644ne"
```

```
    }
}
```

Cree clústeres a partir de copias de seguridad (AWS CloudHSM API)

Consulte el siguiente tema para obtener información sobre cómo crear clústeres a partir de copias de seguridad mediante la API.

- CreateCluster

Administración de AWS CloudHSM copias de seguridad

AWS CloudHSM realiza copias de seguridad periódicas del clúster al menos una vez cada 24 horas. Cada backup contiene copias cifradas de los siguientes datos:

- Usuarios (CO, CU y AU)
- Material y certificados de claves
- Configuración y políticas del módulo de seguridad de hardware (HSM)

No puede indicar al servicio que haga copias de seguridad, pero puede realizar ciertas acciones que obliguen al servicio a crear una copia de seguridad. El servicio realiza una copia de seguridad cuando usted efectúa alguna de las siguientes acciones:

- Activación de un clúster
- Cómo agregar un HSM a un clúster activo
- Cómo quitar un HSM de un clúster.

AWS CloudHSM elimina las copias de seguridad en función de la política de retención de copias de seguridad que haya establecido al crear los clústeres. Para obtener información sobre la administración de la política de retención de copias de seguridad, consulte Configuración de retención de copias de seguridad.

Temas

- Trabajo con copias de seguridad
- Eliminación y restauración de copias de seguridad
- Configurar la política AWS CloudHSM de retención de copias de seguridad
- Copiar copias de seguridad entre AWS regiones

Trabajo con copias de seguridad

Cuando agrega un HSM a un clúster que anteriormente contenía uno o varios HSM activos, el servicio restaura la copia de seguridad más reciente en el nuevo HSM. Utilice las copias de seguridad para administrar los HSM que utiliza con poca frecuencia. Cuando no necesite el HSM, elimínelo, para así activar una copia de seguridad. Más adelante, cuando necesite el HSM, cree uno

nuevo en el mismo clúster. Esta acción restaurará la copia de seguridad que creó anteriormente al eliminar el HSM.

Eliminación de claves caducadas o usuarios inactivos

Es posible que desee eliminar determinados materiales criptográficos de su entorno, como claves caducadas o usuarios inactivos. Se trata de un proceso de dos partes: En primer lugar, elimine estos materiales de su HSM. A continuación, elimine todas las copias de seguridad existentes. Seguir este proceso garantiza que no se restaure la información eliminada al inicializar un nuevo clúster a partir de una copia de seguridad. Para obtener más información, consulte the section called “Eliminación y restauración de copias de seguridad”.

Consideración de recuperación de desastres

Puede crear un clúster a partir de una copia de seguridad. Es posible que desee hacer esto para establecer un punto de recuperación para su clúster. Designe una copia de seguridad que contenga todos los usuarios, el material clave y los certificados que desee incluir en su punto de recuperación y, a continuación, utilice esa copia de seguridad para crear un nuevo clúster. Para obtener más información acerca de crear un clúster en una copia de seguridad, consulte Creación de clústeres a partir de las copias de seguridad.

También puede copiar una copia de seguridad de un clúster en otra región diferente, donde puede crear un nuevo clúster que sea un clon del original. Esto puede ser conveniente por diversas razones, como simplificar el proceso de recuperación de desastres, entre otras. Para obtener más información sobre cómo crear copias de seguridad en diferentes regiones, consulte Cómo clonar copias de seguridad entre regiones.

Eliminación y restauración de copias de seguridad

Tras eliminar una copia de seguridad, el servicio la retiene durante siete días, durante los cuales puede restaurarla. Transcurrido el período de siete días, ya no podrá restaurar la copia de seguridad. Para obtener más información acerca de cómo administrar copias de seguridad, consulte Administración de copias de seguridad.

Eliminación y restauración de copias de seguridad (consola)

Para eliminar una copia de seguridad (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.

2. Para cambiar la región de AWS, utilice el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Backups.
4. Elija una copia de seguridad que desea eliminar.
5. Para eliminar la copia de seguridad seleccionada, elija Acciones, Eliminar.

   Aparecerá el cuadro de diálogo Eliminar copias de seguridad.
6. Elija Eliminar.

   El estado de la copia de seguridad cambia a `PENDING_DELETE`. Puede restaurar una copia de seguridad que esté pendiente de eliminar durante un máximo de 7 días después de haber solicitado la eliminación.

Cómo restaurar una copia de seguridad (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Para cambiar la región de AWS, utilice el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Backups.
4. Elija una copia de seguridad en el estado `PENDING_DELETE` que desea restaurar.
5. Para restaurar la copia de seguridad seleccionada, elija Acciones, Restaurar.

Eliminar y restaurar copias de seguridad (CLI)

Compruebe el estado de una copia de seguridad o busque su ID mediante el describe-backups comando de la CLI.

Para eliminar una copia de seguridad (CLI)

- En el símbolo del sistema, ejecute el comando delete-backup, pasando el ID de la copia de seguridad que desea eliminar.

```
$ aws cloudhsmv2 delete-backup --backup-id <backup ID>
{
    "Backup": {
        "CreateTimestamp": 1534461854.64,
        "ClusterId": "cluster-dygnwhmscg5",
```

```
        "BackupId": "backup-ro5c4er4aac",
        "BackupState": "PENDING_DELETION",
        "DeleteTimestamp": 1536339805.522
    }
}
```

Para restaurar una copia de seguridad (CLI)

- Para restaurar una copia de seguridad, ejecute el comando restore-backup, pasando el ID de una copia de seguridad que tenga el estado `PENDING_DELETION`.

```
$ aws cloudhsmv2 restore-backup --backup-id <backup ID>
{
    "Backup": {
        "ClusterId": "cluster-dygnwhmscg5",
        "CreateTimestamp": 1534461854.64,
        "BackupState": "READY",
        "BackupId": "backup-ro5c4er4aac"
    }
}
```

Para enumerar las copias de seguridad (CLI)

- Para ver una lista de todas las copias de seguridad que tienen el estado `PENDING_DELETION`, ejecute el comando describe-backups e incluya `states=PENDING_DELETION` como filtro.

```
$ aws cloudhsmv2 describe-backups --filters states=PENDING_DELETION
{
    "Backups": [
        {
            "BackupId": "backup-ro5c4er4aac",
            "BackupState": "PENDING_DELETION",
            "CreateTimestamp": 1534461854.64,
            "ClusterId": "cluster-dygnwhmscg5",
            "DeleteTimestap": 1536339805.522,
        }
}
```

Eliminar y restaurar las copias de seguridad (AWS CloudHSM API)

Consulte los siguientes temas para aprender a eliminar y a restaurar copias de seguridad con la API.

- DeleteBackup
- RestoreBackup

Configurar la política AWS CloudHSM de retención de copias de seguridad

A excepción de los clústeres creados antes del 18 de noviembre de 2020, la política de retención de copias de seguridad predeterminada para los clústeres es de 90 días. Puede establecer este período en cualquier número entre 7 y 379 días. AWS CloudHSM no elimina la última copia de seguridad de un clúster. Para obtener más información acerca de cómo administrar copias de seguridad, consulte Administración de copias de seguridad.

Descripción de la política de retención de copias de seguridad

AWS CloudHSM purga las copias de seguridad en función de la política de retención de copias de seguridad que estableció al crear un clúster. La política de retención de copias de seguridad se aplica a los clústeres. Si transfiere una copia de seguridad a otra región, esa copia de seguridad ya no estará asociada a un clúster y no dispondrá de una política de retención de copias de seguridad. Debe eliminar manualmente las copias de seguridad que no estén asociadas a un clúster. AWS CloudHSM no elimina la última copia de seguridad de un clúster.

AWS CloudTrail informa sobre las copias de seguridad marcadas para su eliminación. Puede restaurar las copias de seguridad que el servicio purga del mismo modo que restauraría las copias de seguridad eliminadas manualmente. Para evitar una condición de carrera, debe cambiar la política de retención de copias de seguridad del clúster antes de restaurar una copia de seguridad eliminada por el servicio. Si desea mantener la misma política de retención y conservar determinadas copias de seguridad, puede especificar que el servicio excluya las copias de seguridad de la política de retención de copias de seguridad del clúster.

Exención de clústeres existentes

AWS CloudHSM lanzó la retención gestionada de copias de seguridad el 18 de noviembre de 2020. Los clústeres creados antes del 18 de noviembre de 2020 disponen de una política de retención de copias de seguridad de 90 días más la antigüedad del clúster. Por ejemplo, si creó un clúster el 18

de noviembre de 2019, el servicio le asignará una política de retención de copias de seguridad de un año más 90 días (455 días).

> **Note**
>
> Puede excluirse de la retención de copias de seguridad administrada, póngase en contacto con el servicio de asistencia (https://aws.amazon.com/support).

Configurar la retención de copias de seguridad (consola)

Cómo configurar una política de retención de copias de seguridad (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Para cambiar la región de AWS, utilice el selector de regiones en la esquina superior derecha de la página.
3. Haga clic en el ID de clúster de un clúster en estado activo para administrar la política de retención de copias de seguridad de ese clúster.
4. Para cambiar la política de retención de copias de seguridad, seleccione Acciones, Cambiar el período de retención de copias de seguridad.

   Aparecerá el cuadro de diálogo Cambiar el periodo de retención de copias de seguridad.
5. En Periodo de retención de copias de seguridad (en días), digite un valor entre 7 y 379 días.
6. Seleccione Cambiar el periodo de retención de copias de seguridad.

Cómo excluir o incluir una copia de seguridad de la política de retención de copias de seguridad (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Para ver las copias de seguridad, elija Copias de seguridad en el panel de navegación.
3. Haga clic en el ID de copia de seguridad de una copia de seguridad en estado Listo para excluirla o incluirla.
4. En la página Detalles de la copia de seguridad, realice una de las siguientes acciones.
   - Para excluir una copia de seguridad cuya fecha esté marcada como Hora de vencimiento, seleccione Acciones, Desactivar el vencimiento.

- Para incluir una copia de seguridad que no caduque, seleccione Acciones, Utilizar la política de retención de clústeres.

Configurar la retención de copias de seguridad (CLI)

Compruebe el estado de una copia de seguridad o busque su ID mediante el describe-backups comando de la CLI.

Para configurar la política de retención de copias de seguridad (CLI)

- En el símbolo del sistema, ejecute el comando modify-cluster. Especifique el ID del clúster y la política de retención de copias de seguridad.

```
$ aws cloudhsmv2 modify-cluster --cluster-id <cluster ID> \
                                --backup-retention-policy  Type=DAYS,Value=<number
 of days to retain backups>
{
   "Cluster": {
      "BackupPolicy": "DEFAULT",
      "BackupRetentionPolicy": {
         "Type": "DAYS",
         "Value": 90
      },
      "Certificates": {},
      "ClusterId": "cluster-kdmrayrc7gi",
      "CreateTimestamp": 1504903546.035,
      "Hsms": [],
      "HsmType": "hsm1.medium",
      "SecurityGroup": "sg-40399d28",
      "State": "ACTIVE",
      "SubnetMapping": {
         "us-east-2a": "subnet-f1d6e798",
         "us-east-2c": "subnet-0e358c43",
         "us-east-2b": "subnet-40ed9d3b"
      },
      "TagList": [
         {
            "Key": "Cost Center",
            "Value": "12345"
         }
      ],
      "VpcId": "vpc-641d3c0d"
```

```
    }
}
```

Para excluir una copia de seguridad de la política de retención de copias de seguridad (CLI)

- En el símbolo del sistema, ejecute el comando modify-backup-attributes. Especifique el ID de la copia de seguridad y establezca la marca Nunca vence para conservar la copia de seguridad.

```
$ aws cloudhsmv2 modify-backup-attributes --backup-id <backup ID> \
                                          --never-expires
{
   "Backup": {
      "BackupId": "backup-ro5c4er4aac",
      "BackupState": "READY",
      "ClusterId": "cluster-dygnwhmscg5",
      "NeverExpires": true
   }
}
```

Para incluir una copia de seguridad en la política de retención de copias de seguridad (CLI)

- En el símbolo del sistema, ejecute el comando modify-backup-attributes. Especifique el ID de la copia de seguridad y configure la no-never-expires marca para incluir la copia de seguridad en la política de retención de copias de seguridad, lo que significa que el servicio eventualmente eliminará la copia de seguridad.

```
$ aws cloudhsmv2 modify-backup-attributes --backup-id <backup ID> \
                                          --no-never-expires
{
   "Backup": {
      "BackupId": "backup-ro5c4er4aac",
      "BackupState": "READY",
      "ClusterId": "cluster-dygnwhmscg5",
      "NeverExpires": false
   }
}
```

Configure la retención de copias de seguridad (AWS CloudHSM API)

Consulte los siguientes temas para obtener información sobre cómo administrar la retención de copias de seguridad con la API.

- ModifyCluster
- ModifyBackupAttributes

Copiar copias de seguridad entre AWS regiones

Puede clonar copias de seguridad entre regiones por muchos motivos, como la resiliencia entre regiones, las cargas de trabajo globales y la recuperación de desastres. Después de clonar las copias de seguridad, aparecerán en la región de destino con un estado de `CREATE_IN_PROGRESS`. Una vez que su ejecución finaliza correctamente, el estado de la copia de seguridad clonada es `READY`. Si la copia falla, el estado de la copia de seguridad cambiará a `DELETED`. Compruebe los parámetros de entrada por si presentan algún error y asegúrese de que la copia de seguridad de origen especificada no se encuentre en el estado `DELETED` antes de volver a ejecutar la operación. Para obtener más información sobre cómo crear un clúster a partir de una copia de seguridad, consulte Administración de copias de seguridad o Creación de clústeres a partir de las copias de seguridad.

Tenga en cuenta lo siguiente:

- Para clonar la copia de seguridad de un clúster en una región de destino, la cuenta debe tener los permisos adecuados de la política de IAM. Con el fin de clonar la copia de seguridad en una región diferente, la política de IAM debe permitir el acceso a la región de origen en la que se encuentra la copia de seguridad. Una vez copiada de una región a otra, la política de IAM debe permitir el acceso a la región de destino con el fin de interaccionar con la copia de seguridad clonada, lo que incluye usar la operación CreateCluster. Para obtener más información, consulte Creación de administradores de IAM.
- El clúster original y el clúster que puede crearse a partir de una copia de seguridad en la región de destino no están vinculados. Deberá administrar cada uno de estos clústeres de manera independiente. Para obtener más información, consulte Administración de clústeres de .
- Las copias de seguridad no se pueden copiar entre regiones AWS restringidas y regiones estándar. Las copias de seguridad se pueden copiar entre las regiones AWS GovCloud (EE. UU. este) y AWS GovCloud (EE. UU., oeste).

Clonar copias de seguridad en diferentes regiones (consola)

Para clonar copias de seguridad en diferentes regiones (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Para cambiar la región de AWS, utilice el selector de regiones en la esquina superior derecha de la página.
3. En el panel de navegación, elija Backups.
4. Elija una copia de seguridad para clonarla en otra región.
5. Para clonar la copia de seguridad seleccionada, seleccione Acciones, copiar copia de seguridad a otra región.

   Aparece el cuadro de diálogo Copy backup to another region (Clonar copia de seguridad a otra región).
6. En Región de destino, elija una región en Seleccionar una región.
7. (Opcional) Escriba una clave de etiqueta y un valor de etiqueta opcional. Para agregar más de una etiqueta al clúster, elija Agregar etiqueta.
8. Elija Copy backup.

Copiar copias de seguridad a diferentes regiones (CLI)

Para determinar el ID de la copia de seguridad, ejecute el comando describe-backups.

Para copiar copias de seguridad a diferentes regiones (CLI)

- En el símbolo del sistema, ejecute el comando copy-backup-to-region. Especifique la región de destino y el ID de copia de seguridad de la copia de seguridad de origen. Si especifica un ID de copia de seguridad, se copiará la copia de seguridad asociada.

```
$ aws cloudhsmv2 copy-backup-to-region --destination-region <destination region> \
                                       --backup-id <backup ID>
```

Copiar las copias de seguridad a diferentes regiones (AWS CloudHSM API)

Consulte el siguiente tema para obtener información sobre cómo copiar copias de seguridad en diferentes regiones mediante la API.

- CopyBackupToRegion

Recursos de etiquetado AWS CloudHSM

Una etiqueta es una etiqueta que se asigna a un AWS recurso. Puede asignar etiquetas a los clústeres de AWS CloudHSM . Cada etiqueta consta de una clave de etiqueta y un valor de etiqueta, ambos definidos por el usuario. Por ejemplo, la clave de etiqueta puede ser Centro de costos y el valor de etiqueta puede ser 12345. Las claves de las etiquetas deben ser únicas para cada clúster.

Puede usar etiquetas para distintos fines. Uno de los usos habituales es categorizar y realizar el seguimiento de los costos de AWS . Puede aplicar etiquetas que representen categorías de negocio (p. ej., centros de costos, nombres de aplicación o propietarios) para estructurar los costos entre diferentes servicios. Al añadir etiquetas a AWS los recursos, AWS genera un informe de asignación de costes con el uso y los costes agregados por etiquetas. Puede usar este informe para ver sus AWS CloudHSM costos en términos de proyectos o aplicaciones, en lugar de ver todos AWS CloudHSM los costos como una sola partida.

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 .

Puede utilizar la consola de AWS CloudHSM o uno de los SDK o herramientas de línea de comandos de AWS para añadir, actualizar, generar una lista o quitar etiquetas.

Temas

- Adición o actualización de etiquetas
- Enumeración de etiquetas
- Eliminación de etiquetas

Adición o actualización de etiquetas

Puede añadir o actualizar etiquetas desde la AWS CloudHSM consola, la AWS Command Line Interface (CLI) o la AWS CloudHSM API.

Para añadir o actualizar etiquetas (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Elija el clúster que está etiquetando.
3. Seleccione Tags (Etiquetas).

4. Para añadir una etiqueta, haga lo siguiente:

   a. Seleccione Edit Tag (Editar etiqueta) y, a continuación, Add Tag (Añadir etiqueta).

   b. En Tag Key (Clave de etiqueta), escriba una clave para la etiqueta.

   c. (Opcional) En Value (Valor), escriba un valor para la etiqueta.

   d. Seleccione Guardar.

5. Para actualizar una etiqueta, haga lo siguiente:

   a. Seleccione Edit Tag (Editar etiqueta).

   > **Note**
   >
   > Si actualiza la clave de una etiqueta existente, la consola elimina la etiqueta existente y crea otra nueva.

   b. Escriba el nuevo valor de la etiqueta.

   c. Seleccione Guardar.

Para agregar o actualizar etiquetas (CLI)

1. En el símbolo del sistema, ejecute el comando tag-resource especificando las etiquetas y el ID del clúster que está etiquetando. Si no sabe cuál es el ID del clúster, ejecute el comando describe-clusters.

```
$ aws cloudhsmv2 tag-resource --resource-id <cluster ID> \
                              --tag-list Key="<tag key>",Value="<tag value>"
```

2. Para actualizar etiquetas, utilice el mismo comando, pero especifique una clave de etiqueta ya existente. Cuando especifique un valor de etiqueta nuevo para una etiqueta ya existente, la etiqueta se sobrescribe con el nuevo valor.

Para añadir o actualizar etiquetas (AWS CloudHSM API)

- Envíe una solicitud TagResource. Especifique las etiquetas y el ID del clúster que está etiquetando.

Enumeración de etiquetas

Puede enumerar las etiquetas de un clúster desde la AWS CloudHSM consola, la CLI o la AWS CloudHSM API.

Para generar una lista de etiquetas (consola)

1. Abre la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Elija el clúster cuyas etiquetas vaya a incluir en una lista.
3. Seleccione Tags (Etiquetas).

Para enumerar etiquetas (CLI)

- En el símbolo del sistema ejecute, el comando list-tags especificando el ID del clúster cuyas etiquetas quiere mostrar. Si no sabe cuál es el ID del clúster, ejecute el comando describe-clusters.

```
$ aws cloudhsmv2 list-tags --resource-id <cluster ID>
{
    "TagList": [
        {
            "Key": "Cost Center",
            "Value": "12345"
        }
    ]
}
```

Para enumerar etiquetas (AWS CloudHSM API)

- Envía una solicitud de ListTags en la que se especifique el ID del clúster cuyas etiquetas quiere mostrar.

Eliminación de etiquetas

Puede eliminar etiquetas de un clúster mediante la AWS CloudHSM consola, la CLI o la AWS CloudHSM API.

Para eliminar etiquetas (consola)

1. Abre la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Elija el clúster cuyas etiquetas va a eliminar.
3. Seleccione Tags (Etiquetas).
4. Seleccione Edit Tag (Editar etiqueta) y haga clic en la opción Remove tag (Eliminar etiqueta) de la etiqueta que desee eliminar.
5. Seleccione Guardar.

Para eliminar etiquetas (CLI)

- En el símbolo del sistema, ejecute el comando untag-resource especificando las claves de etiqueta de las etiquetas que está eliminando y el ID del clúster cuyas etiquetas está eliminando. Cuando utilice la CLI para eliminar etiquetas, especifique solo las claves de las etiquetas, no los valores de las etiquetas.

```
$ aws cloudhsmv2 untag-resource --resource-id <cluster ID> \
                                --tag-key-list "<tag key>"
```

Para eliminar etiquetas (AWS CloudHSM API)

- Envía una UntagResourcesolicitud en la AWS CloudHSM API especificando el ID del clúster y las etiquetas que vas a eliminar.

Administrar los usuarios y las claves de HSM AWS CloudHSM

Antes de poder usar el AWS CloudHSM clúster para el criptoprocesamiento, debe crear usuarios y claves en los HSM del clúster. Consulte los siguientes temas para obtener más información acerca de la gestión de usuarios y claves HSM en AWS CloudHSM. También puede aprender a utilizar la autenticación de cuórum (también conocida como "control de acceso M de N").

Temas

- Administrar usuarios de HSM en AWS CloudHSM
- Administrar claves en AWS CloudHSM
- Administración de clústeres clonados

Administrar usuarios de HSM en AWS CloudHSM

En AWS CloudHSM, debe usar las herramientas de línea de comandos de CloudHSM CLI o CloudHSM Management Utility (CMU) para crear y administrar los usuarios de su HSM. La CLI de CloudHSM está diseñada para su uso con la serie de versiones más reciente del SDK, mientras que la CMU está diseñada para su uso con la serie de versiones previas del SDK.

Temas

- Administración de usuarios de HSMcon la CLI de CloudHSM
- Administrar usuarios de HSM con la Utilidad de administración CloudHSM (CMU)

Administración de usuarios de HSMcon la CLI de CloudHSM

Utilice las herramientas de línea de comandos la CLI de CloudHSM para crear y administrar los usuarios de su HSM con el SDK más reciente.

Temas

- Más información sobre los usuarios de HSM
- Tabla de permisos de usuario de HSM
- Uso de la CLI de CloudHSM para administrar usuarios

- Uso de la CLI de CloudHSM para gestionar la MFA
- Usar la CLI de CloudHSM para gestionar la autenticación de cuórum (control de acceso M de N)

Más información sobre los usuarios de HSM

La mayoría de las operaciones que realiza en el HSM requieren las credenciales de un usuario de HSM. El HSM autentica a cada usuario del HSM y cada usuario del HSM tiene un tipo que determina las operaciones que puede realizar en el HSM como ese usuario.

> **Note**
>
> Los usuarios de HSM son distintos de los usuarios de IAM. Los usuarios de IAM que dispongan de credenciales correctas pueden crear HSM interactuando con los recursos a través de la API de AWS. Una vez creado el HSM, deberá introducir las credenciales de usuario de HSM para autenticar las operaciones del mismo.

Tipos de usuario

- Administrador desactivado
- Administrador
- Usuario de criptografía (CU)
- Usuario de dispositivos (AU)

Administrador desactivado

En la CLI de CloudHSM, el administrador desactivado es un usuario temporal que solo existe en el primer HSM de un clúster de AWS CloudHSM que nunca se ha activado. Para activar un clúster, ejecute el comandocluster activate en la CLI de CloudHSM. Tras ejecutar este comando, se solicita a los administradores que no estén activados que cambien la contraseña. Tras cambiar la contraseña, el administrador desactivado pasa a ser administrador.

Administrador

En la CLI de CloudHSM, el administrador puede realizar operaciones de administración de usuarios. Por ejemplo, pueden crear y eliminar usuarios, así como cambiar las contraseñas de los usuarios. Para obtener más información sobre los administradores, consulte Tabla de permisos de usuario de HSM.

Usuario de criptografía (CU)

Un usuario de criptografía (CU) puede realizar las siguientes operaciones de administración de claves y criptografía.

- Administración de claves: crear, eliminar, compartir, importar y exportar claves criptográficas.
- Operaciones criptográficas: utilizar las claves criptográficas para cifrado, descifrado, firma, verificación y mucho más.

Para obtener más información, consulte Tabla de permisos de usuario de HSM.

Usuario de dispositivos (AU)

El usuario del dispositivo (AU) puede realizar operaciones de clonación y sincronización en los HSM del clúster. AWS CloudHSM utiliza la AU para sincronizar los HSM de un clúster. AWS CloudHSM La AU existe en todos los HSM proporcionados por ellos y tiene AWS CloudHSM permisos limitados. Para obtener más información, consulte Tabla de permisos de usuario de HSM.

AWS no puede realizar ninguna operación en sus HSM. AWS no puede ver ni modificar sus usuarios o claves y no puede realizar ninguna operación criptográfica con esas claves.

Tabla de permisos de usuario de HSM

La siguiente tabla muestra las operaciones de HSM ordenadas por tipo de usuario o sesión de HSM que puede realizar la operación.

	Administrador	Usuario de criptografía (CU)	Usuario de dispositivos (AU)	Sesiones no autenticadas
Obtener información básica del clúster[1]	Sí	Sí	Sí	Sí
Cambiar su propia contraseña	Sí	Sí	Sí	No aplicable

	Administrador	Usuario de criptografía (CU)	Usuario de dispositivos (AU)	Sesiones no autenticadas
Cambiar la contraseña de cualquier usuario	 Sí	 No	 No	 No
Agregar o eliminar usuarios	 Sí	 No	 No	 No
Obtener el estado de sincronización[2]	 Sí	 Sí	 Sí	 No
Extraer o insertar objetos enmascarados[3]	 Sí	 Sí	 Sí	 No
Funciones de administración de claves[4]	 No	 Sí	 No	 No
Cifrar o descifrar	 No	 Sí	 No	 No
Firmar o verificar	 No	 Sí	 No	 No

	Administrador	Usuario de criptografía (CU)	Usuario de dispositivos (AU)	Sesiones no autenticadas
Generar resúmenes y HMAC	 No	 Sí	 No	 No

- [1] La información básica sobre el clúster incluye el número de HSM que hay en el clúster y la dirección IP, el modelo, el número de serie, el ID de dispositivo, el ID de firmware, etc. de cada HSM.
- [2] El usuario puede obtener un conjunto de resúmenes (hashes) que se corresponden con las claves del HSM. Una aplicación puede comparar estos conjuntos de resúmenes para conocer el estado de la sincronización de los HSM de un clúster.
- [3] Los objetos enmascarados son claves que se cifran antes de salir del HSM. No se pueden descifrar fuera del HSM. Solo se descifran después de insertarlos en un HSM que se encuentra en el mismo clúster que el HSM del que se extrajeron. Una aplicación puede extraer e insertar objetos enmascarados para sincronizar los HSM de un clúster.
- [4] Las funciones de administración de claves incluyen la creación, eliminación, encapsulación y modificación de los atributos de las claves.

Uso de la CLI de CloudHSM para administrar usuarios

En este tema se proporcionan step-by-step instrucciones sobre la administración de los usuarios del módulo de seguridad de hardware (HSM) con la CLI de CloudHSM. Para obtener más información sobre los usuarios de HSM o la CLI de CloudHSM, consulte CLI de CloudHSM y Utilización de la CLI de CloudHSM.

Secciones

- Más información de la gestión de usuarios de HSM con la CLI de CloudHSM
- Cómo descargar la CLI de CloudHSM
- Cómo gestionar los usuarios de HSM con la CLI de CloudHSM

Más información de la gestión de usuarios de HSM con la CLI de CloudHSM

Para administrar usuarios de HSM, inicie sesión en el HSM con el nombre de usuario y la contraseña de un administrador. Solo los administradores pueden administrar usuarios. El HSM contiene un administrador predeterminado denominado admin. Usted estableció la contraseña para admin cuando activó el clúster.

Para usar la CLI de CloudHSM, debe usar la herramienta de configuración para actualizar la configuración local. Para obtener instrucciones sobre cómo ejecutar la herramienta de configuración con la CLI de CloudHSM, consulte Introducción a la interfaz de la línea de comandos (CLI) de CloudHSM. El parámetro `-a` solicita agregar la dirección IP de un HSM en su clúster. Si tiene varios HSM, puede usar cualquier dirección IP. Esto garantiza que la CLI de CloudHSM pueda propagar cualquier cambio que realice en todo el clúster. Recuerde que la CLI de CloudHSM usa su archivo local para rastrear la información del clúster. Si el clúster ha cambiado desde la última vez que utilizó la CLI de CloudHSM desde un host concreto, debe añadir esos cambios al archivo de configuración local almacenado en ese host. Nunca extraiga un HSM mientras utilice la CLI de CloudHSM.

Para obtener una dirección IP para un HSM (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Para cambiar la región de AWS, utilice el selector de regiones en la esquina superior derecha de la página.
3. Para abrir la página de detalles del clúster, en la tabla de clústeres, elija el ID del clúster.
4. Para obtener la dirección IP, vaya a la pestaña HSM y elija una de las direcciones IP que aparecen en la lista Dirección IP de ENI.

Para obtener una dirección IP para un HSM (CLI)

- Obtenga la dirección IP de un HSM mediante el describe-clusters comando de la CLI. En el resultado del comando, la dirección IP de los HSM son los valores de `EniIp`.

```
$  aws cloudhsmv2 describe-clusters


{
    "Clusters": [
        { ... }
            "Hsms": [
                {
```

```
...
                    "EniIp": "10.0.0.9",
...
                },
                {
...
                    "EniIp": "10.0.1.6",
...
```

Cómo descargar la CLI de CloudHSM

La última versión de la CLI de CloudHSM está disponible para las tareas de administración de usuarios de HSM para SDK 5 de cliente. Para descargar e instalar la CLI de CloudHSM, siga las instrucciones recogidas en Instalación y configuración de la CLI de CloudHSM.

Cómo gestionar los usuarios de HSM con la CLI de CloudHSM

En esta sección se incluyen comandos básicos para administrar los usuarios de HSM con la CLI de CloudHSM.

> **Note**
>
> Nota: Los comandos de usuario de la CLI de CloudHSM se enumeran en la referencia de comandos de usuario de la CLI de CloudHSM.

Temas

- Cómo crear un administrador
- : cómo crear un usuario de criptografía
- Cómo enumerar todos los usuarios de HSM del clúster
- Cómo cambiar las contraseñas de los usuarios de HSM
- Cómo eliminar usuarios de HSM

Cómo crear un administrador

Siga estos pasos para crear un administrador.

1. Use el siguiente comando para iniciar el modo interactivo de la CLI de CloudHSM.

Linux

```
$ /opt/cloudhsm/bin/cloudhsm-cli interactive
```

Windows

```
C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
```

2. Ejecute el comando login e inicie sesión en el clúster como administrador.

```
aws-cloudhsm > login --username <USERNAME> --role admin
```

3. El sistema le solicitará su contraseña. Introduzca la contraseña y el resultado mostrará que el comando se ejecutó correctamente.

```
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

4. Escriba el siguiente comando para crear un administrador:

```
aws-cloudhsm > user create --username <USERNAME> --role admin
```

5. Ingrese la contraseña para un nuevo usuario.
6. Vuelva a introducir la contraseña para confirmar que la contraseña que ha introducido es correcta.

: cómo crear un usuario de criptografía

Siga estos pasos para crear un usuario.

1. Use el siguiente comando para iniciar el modo interactivo de la CLI de CloudHSM.

Linux

```
$ /opt/cloudhsm/bin/cloudhsm-cli interactive
```

Windows

```
C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
```

2. Ejecute el comando login e inicie sesión en el clúster como administrador.

```
aws-cloudhsm > login --username <USERNAME> --role admin
```

3. El sistema le solicitará su contraseña. Introduzca la contraseña y el resultado mostrará que el comando se ejecutó correctamente.

```
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

4. Escriba el siguiente comando para crear un usuario de criptografía:

```
aws-cloudhsm > user create --username <USERNAME> --role crypto-user
```

5. Ingrese la contraseña del usuario de criptografía.
6. Vuelva a introducir la contraseña para confirmar que la contraseña que ha introducido es correcta.

Cómo enumerar todos los usuarios de HSM del clúster

Utilice el comando user list para enumerar todos los usuarios del clúster. No es necesario iniciar sesión para ejecutar user list. Todos los tipos de usuarios pueden enumerar usuarios.

Siga estos pasos para enumerar todos los usuarios del clúster.

1. Use el siguiente comando para iniciar el modo interactivo de la CLI de CloudHSM.

Linux

```
$ /opt/cloudhsm/bin/cloudhsm-cli interactive
```

Windows

```
C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
```

2. Introduzca el siguiente comando para enumerar todos los usuarios del clúster:

```
aws-cloudhsm > user list
```

Para obtener más información sobre user list, consulte lista de usuarios.

Cómo cambiar las contraseñas de los usuarios de HSM

Utilice el comando user change-password para cambiar una contraseña.

En los tipos de usuario y las contraseñas se distingue entre mayúsculas y minúsculas, pero no en los nombres de usuario.

Los administradores, los usuarios de criptografía (CU) y los usuarios de dispositivos (AU)pueden cambiar su contraseña. Para cambiar la contraseña de otro usuario, debe iniciar sesión como administrador. No puede cambiar la contraseña de un usuario que actualmente haya iniciado sesión.

Cómo cambiar su propia contraseña

1. Use el siguiente comando para iniciar el modo interactivo de la CLI de CloudHSM.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm-cli interactive
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
   ```

2. Utilice el comando login e inicie sesión como el usuario con la contraseña que desea cambiar.

```
aws-cloudhsm > login --username <USERNAME> --role <ROLE>
```

3. Ingrese la contraseña del usuario.

```
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin1",
    "role": "admin"
  }
}
```

4. Escriba el comando user change-password.

```
aws-cloudhsm > user change-password --username <USERNAME> --role <ROLE>
```

5. Introduzca la nueva contraseña.
6. Vuelva a introducir la nueva contraseña.

Cómo cambiar la contraseña de otro usuario

1. Use el siguiente comando para iniciar el modo interactivo de la CLI de CloudHSM.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm-cli interactive
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
   ```

2. Ejecute el comando login e inicie sesión en el clúster como administrador.

```
aws-cloudhsm > login --username <USERNAME> --role admin
```

3. Introduzca la contraseña del administrador.

```
Enter password:
{
```

```
  "error_code": 0,
  "data": {
    "username": "admin1",
    "role": "admin"
  }
}
```

4. Introduzca el comando user change-password junto con el nombre del usuario cuya contraseña desea cambiar.

```
aws-cloudhsm > user change-password --username <USERNAME> --role <ROLE>
```

5. Introduzca la nueva contraseña.
6. Vuelva a introducir la nueva contraseña.

Para obtener más información sobre user change-password, consulte user change-password.

Cómo eliminar usuarios de HSM

Ejecute user delete para eliminar un usuario. Debe iniciar sesión como administrador para eliminar otro usuario.

> **Tip**
>
> No puede eliminar a los usuarios de criptografía (CU) que poseen claves.

Cómo eliminar un usuario

1. Use el siguiente comando para iniciar el modo interactivo de la CLI de CloudHSM.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm-cli interactive
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
   ```

2. Ejecute el comando login e inicie sesión en el clúster como administrador.

```
aws-cloudhsm > login --username <USERNAME> --role admin
```

3. El sistema le solicitará su contraseña. Introduzca la contraseña y el resultado mostrará que el comando se ejecutó correctamente.

```
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

4. Ejecute el comando user delete para eliminar un usuario.

```
aws-cloudhsm > user delete --username <USERNAME> --role <ROLE>
```

Para obtener más información acerca de user delete, consulte deleteUser.

Uso de la CLI de CloudHSM para gestionar la MFA

Para más seguridad, le recomendamos que configure la autenticación multifactor (MFA) para ayudar a proteger el clúster. Para obtener más información, consulte los siguientes temas.

Temas

- Más información sobre la MFA para los usuarios de HSM
- Trabajo con MFA para usuarios de HSM

Más información sobre la MFA para los usuarios de HSM

Cuando inicia sesión en un clúster con una cuenta de usuario de HSM habilitada para MFA, proporciona su contraseña a la CLI de CloudHSM (el primer factor, lo que sabe) y la CLI de CloudHSM le proporciona un token y le pide que firme el token.

Para proporcionar el segundo factor (el que ya tienes), firma el token con una clave privada de un par de claves que ya ha creado y asociado al usuario de HSM. Para acceder al clúster, debe proporcionar el token firmado a la CLI de CloudHSM.

Para obtener más información sobre la configuración de MFA para un usuario, consulte Configuración de MFA para la CLI de CloudHSM

Autenticación de cuórum y MFA

El clúster usa la misma clave para la autenticación de cuórum y para MFA. Esto significa que un usuario con MFA habilitado está registrado efectivamente para el control de acceso MoFN o Quroum. Para utilizar correctamente la autenticación de cuórum y la MFA para el mismo usuario de HSM, tenga en cuenta los siguientes puntos:

- Si actualmente usa la autenticación de cuórum para un usuario, debe usar el mismo par de claves que creó para el usuario de cuórum para habilitar la MFA para el usuario.
- Si agrega el requisito de MFA para un usuario que no es MFA y no es un usuario de autenticación de cuórum, entonces registre ese usuario como un usuario registrado de Quroum MoFN con autenticación de MFA.
- Si elimina el requisito de MFA o cambia la contraseña de un usuario de MFA que también es un usuario registrado de autenticación de cuórum, también eliminará el registro del usuario como usuario de cuórum MoFN.
- Si elimina el requisito de MFA o cambia la contraseña de un usuario de MFA que también es usuario de autenticación de cuórum, pero aún desea que ese usuario participe en la autenticación de cuórum, debe volver a registrar ese usuario como usuario de cuórum MoFN.

Para obtener más información acerca de la autenticación de cuórum, consulte Gestión del cuórum (M de N).

Trabajo con MFA para usuarios de HSM

En este tema se proporciona información e instrucciones para usar la CLI de CloudHSM para administrar la autenticación multifactor (MFA). Para obtener más información acerca de la CLI de CloudHSM, consulte Interfaz de la línea de comandos (CLI) de CloudHSM.

Temas

- Requisitos del par de claves de MFA
- Configuración de MFA para la CLI de CloudHSM
- Creación de usuarios con la MFA activada
- Inicio de sesión con usuarios con la MFA activada
- Rotación de las claves para los usuarios con la MFA activada

- Anulación del registro de una clave pública de MFA para los usuarios administradores cuando la clave pública de MFA esté registrada
- Referencia del archivo del token

Para obtener más información sobre el trabajo con usuarios de HSM, consulte Interfaz de la línea de comandos (CLI) de CloudHSM.

Requisitos del par de claves de MFA

Para habilitar la MFA para un usuario de HSM, puede crear un nuevo par de claves o utilizar una clave existente que cumpla los siguientes requisitos:

- Tipo de clave: asimétrica
- Uso de clave: firmar y verificar
- Especificaciones de clave: RSA_2048
- El algoritmo de firma incluye: sha256WithRSAEncryption

> **Note**
>
> Si utiliza la autenticación de cuórum o planea utilizarla, consulte Autenticación de cuórum y MFA

Puede usar la CLI de CloudHSM y el par de claves para crear un nuevo usuario administrador con la MFA activada.

Configuración de MFA para la CLI de CloudHSM

Siga estos pasos para configurar la MFA para la CLI de CloudHSM.

1. Para configurar la MFA mediante la estrategia de firma de tokens, primero debe generar una clave privada de RSA de 2048 bits y una clave pública asociada.

```
$ openssl genrsa -out officer1.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................+++++
.....................................................................+++++
e is 65537 (0x010001)
```

```
$ openssl rsa -in officer1.key -outform PEM -pubout -out officer1.pub
writing RSA key
```

2. Con la CLI de CloudHSM, inicie sesión en su cuenta de usuario.

```
$ cloudhsm-cli interactive
aws-cloudhsm > login --username admin --role admin --cluster-id <cluster ID>
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

3. A continuación, ejecute el comando para cambiar su estrategia de MFA. Debe proporcionar el parámetro `--token`. Este parámetro especifica un archivo en el que se escribirán tokens sin firmar.

```
aws-cloudhsm > user change-mfa token-sign --token unsigned-tokens.json --
username <USERNAME> --role crypto-user --change-quorum
Enter password:
Confirm password:
```

4. Ahora tiene un archivo con tokens sin firmar que deben firmarse: `unsigned-tokens.json`. La cantidad de tokens de este archivo depende de la cantidad de HSM del clúster. Cada token representa un HSM. Este archivo tiene formato JSON y contiene tokens que deben firmarse para demostrar que tiene una clave privada.

```
$ cat unsigned-tokens.json
{
  "version": "2.0",
  "tokens": [
{
    {
      "unsigned": "Vtf/9QOFY45v/E1osvpEMr59JsnP/hLDm4ItOO2vqL8=",
      "signed": ""
    },
    {
      "unsigned": "wVbC0/5IKwjyZK2NBpdFLyI7BiayZ24YcdUdlcxLwZ4=",
```

```
      "signed": ""
    },
    {
      "unsigned": "z6aW9RzErJBL5KqFG5h8lhTVt9oLbxppjod0Ebysydw=",
      "signed": ""
    }
  ]
}
```

5. El siguiente paso es firmar estos tokens con la clave privada creada en el paso 1. Vuelva a colocar las firmas en el archivo. En primer lugar, debe extraer y decodificar los tokens codificados en base64.

```
$ echo "Vtf/9QOFY45v/E1osvpEMr59JsnP/hLDm4ItOO2vqL8=" > token1.b64
$ echo "wVbC0/5IKwjyZK2NBpdFLyI7BiayZ24YcdUdlcxLwZ4=" > token2.b64
$ echo "z6aW9RzErJBL5KqFG5h8lhTVt9oLbxppjod0Ebysydw=" > token3.b64
$ base64 -d token1.b64 > token1.bin
$ base64 -d token2.b64 > token2.bin
$ base64 -d token3.b64 > token3.bin
```

6. Ahora tiene los tokens binarios que puede firmar con la clave privada RSA creada en el paso 1.

```
$ openssl pkeyutl -sign \
      -inkey officer1.key \
      -pkeyopt digest:sha256 \
      -keyform PEM \
      -in token1.bin \
      -out token1.sig.bin
$ openssl pkeyutl -sign \
      -inkey officer1.key \
      -pkeyopt digest:sha256 \
      -keyform PEM \
      -in token2.bin \
      -out token2.sig.bin
$ openssl pkeyutl -sign \
      -inkey officer1.key \
      -pkeyopt digest:sha256 \
      -keyform PEM \
      -in token3.bin \
      -out token3.sig.bin
```

7. Ahora tiene las firmas binarias de los tokens. Debe codificarlos con base64 y volver a colocarlos en su archivo del token.

```
$ base64 -w0 token1.sig.bin > token1.sig.b64
$ base64 -w0 token2.sig.bin > token2.sig.b64
$ base64 -w0 token3.sig.bin > token3.sig.b64
```

8. Por último, puede volver a copiar y pegar los valores de base64 en su archivo del token:

```
{
  "version": "2.0",
  "tokens": [
    {
      "unsigned": "1jqwxb9bJOUUQLiNb7mxXS1uBJsEXh0B9nj05BqnPsE=",
      "signed": "eiw3fZeCKIY50C4zPeg9Rt90M1Qlq3WlJh6Yw7xXm4nF6e9ETLE39+9M
+rUqDWMRZjaBfaMbg5d9yDkz5p13U7ch2tlF9LoYabsWutkT014KRq/rcYMvFsU9n/Ey/
TK0PVaxLN42X+pebV4juwMhN4mK4CzdFAJgM+UGBOj4yB9recpOBB9K8QFSpJZALSEdDgUc/
mS1eDq3rU0int6+4NKuLQjpR
+LSEIWRZ6g6+MND2vXGskxHjadCQ09L7Tz8VcWjKDbxJcBiGKvkqyozl9zrGo8fA3WHBmwiAgS61Merx77ZGY4PFR37
YMSC14prCN15DtMRv2xA1SGSb4w=="
    },
    {
      "unsigned": "LMMFc34ASPnvNPFzBbMbr9FProS/Zu2P8zF/xzk5hVQ=",
      "signed": "HBImKnHmw+6R2TpFEpfiAg4+hu2pFNwn43ClhKPkn2higbEhUD0JVi
+4MerSyvU/NN79iWVxDvJ9Ito+jpiRQjTfTGEoIteyuAr1v/Bzh+HjmrO53OQpZaJ/VXGIgApD0myuu/
ZGNKQTCSkkL7+V81FG7yR1Nm22jUeGa735zvm/E+cenvZdy0VVx6A7WeWrl3JEKKBweHbi+7BwbaW
+PTdCuIRd4Ug76Sy+cFhsvcG1k7cMwDh8MgXzIZ2m1f/hdy2j8qAxORTLlmwyUOYvPYOvUhc
+s83hx36QpGwGcD7RA0bPT5OrTx7PHd0N1CL+Wwy91We8yIOFBS6nxo1R7w=="
    },
    {
      "unsigned": "dzeHbwhiVXQqcUGj563z51/7sLUdxjL93SbOUyZRjH8=",
      "signed": "VgQPvrTsvGljVBFxHnswduq16x8ZrnxfcYVYGf/
N7gEzI4At3GDs2EVZWTRdvS0uGHdkFYp1apHgJZ7PDVmGcTkIXVD2lFYppcgNlSzkYlftr5EOjqS9ZjYEqgGuB4g//
MxaBaRbJai/6BlcE92NIdBusTtreIm3yTpjIXNAVoeRSnkfuw7wZcL96QoklNb1WUuSHw
+psUyeIVtIwFMHEfFoRC0t
+VhmnlnFnkjGPb9W3Aprw2dRRvFM3R2ZTDvMCiOYDzUCd43GftGq2LfxH3qSD51oFHglHQVOY0jyVzzlAvub5HQdtOQ
    }
  ]
}
```

9. Ahora que su archivo del token tiene todas las firmas necesarias, puede continuar. Introduzca el nombre del archivo que contiene los tokens firmados y pulse la tecla Intro. Por último, introduzca la ruta de su clave pública.

```
Enter signed token file path (press enter if same as the unsigned token file):
Enter public key PEM file path:officer1.pub
{
  "error_code": 0,
  "data": {
    "username": "<USERNAME>",
    "role": "crypto-user"
  }
}
```

   Ahora ha configurado su usuario con MFA.

```
{
    "username": "<USERNAME>",
    "role": "crypto-user",
    "locked": "false",
    "mfa": [
      {
        "strategy": "token-sign",
        "status": "enabled"
      }
    ],
    "cluster-coverage": "full"
},
```

Creación de usuarios con la MFA activada

Siga estos pasos para crear usuarios con la MFA activada.

1. Utilice la CLI de CloudHSM para iniciar sesión en el HSM como administrador.
2. Utilice el comando user create para crear el usuario que desee. A continuación, siga los pasos que se indican en Configuración de MFA para la CLI de CloudHSM para configurar la MFA para el usuario.

Inicio de sesión con usuarios con la MFA activada

Siga estos pasos para crear usuarios con la MFA activada.

1. Utilice el comando de login mfa-token-signde la CLI de CloudHSM para iniciar el proceso de inicio de sesión con MFA para un usuario que tenga la MFA habilitada.

```
aws-cloudhsm > login --username <USERNAME> --role <ROLE> mfa-token-sign --token
 unsigned-tokens.json
Enter password:
```

2. Introduzca la contraseña. A continuación, se le pedirá que introduzca la ruta al archivo de token que contiene los pares de tokens firmados y sin firmar, donde los tokens firmados son los que se generan con su clave privada.

```
aws-cloudhsm > login --username <USERNAME> --role <ROLE> mfa-token-sign --token
 unsigned-tokens.json
Enter password:
Enter signed token file path (press enter if same as the unsigned token file):
```

3. Mientras se le pide que introduzca la ruta al archivo de token firmado, puede inspeccionar el archivo de token sin firmar en una terminal independiente. Identifique el archivo con los tokens sin firmar que deben firmarse: `unsigned-tokens.json`. La cantidad de tokens de este archivo depende de la cantidad de HSM del clúster. Cada token representa un HSM. Este archivo tiene formato JSON y contiene tokens que deben firmarse para demostrar que tiene una clave privada.

```
$ cat unsigned-tokens.json
{
  "version": "2.0",
  "tokens": [
    {
      "unsigned": "Vtf/9QOFY45v/E1osvpEMr59JsnP/hLDm4ItOO2vqL8=",
      "signed": ""
    },
    {
      "unsigned": "wVbC0/5IKwjyZK2NBpdFLyI7BiayZ24YcdUdlcxLwZ4=",
      "signed": ""
    },
    {
      "unsigned": "z6aW9RzErJBL5KqFG5h8lhTVt9oLbxppjod0Ebysydw=",
      "signed": ""
```

```
    }
  ]
}
```

4. Firme los tokens sin firmar con la clave privada creada en el paso 2. En primer lugar, debe extraer y decodificar los tokens codificados en base64.

```
$ echo "Vtf/9QOFY45v/E1osvpEMr59JsnP/hLDm4ItOO2vqL8=" > token1.b64
$ echo "wVbC0/5IKwjyZK2NBpdFLyI7BiayZ24YcdUdlcxLwZ4=" > token2.b64
$ echo "z6aW9RzErJBL5KqFG5h8lhTVt9oLbxppjod0Ebysydw=" > token3.b64
$ base64 -d token1.b64 > token1.bin
$ base64 -d token2.b64 > token2.bin
$ base64 -d token3.b64 > token3.bin
```

5. Ahora tiene tokens binarios. Fírmelos con la clave privada de RSA que creó anteriormente en el paso 1 de la configuración de la MFA.

```
$ openssl pkeyutl -sign \
      -inkey officer1.key \
      -pkeyopt digest:sha256 \
      -keyform PEM \
      -in token1.bin \
      -out token1.sig.bin
$ openssl pkeyutl -sign \
      -inkey officer1.key \
      -pkeyopt digest:sha256 \
      -keyform PEM \
      -in token2.bin \
      -out token2.sig.bin
$ openssl pkeyutl -sign \
      -inkey officer1.key \
      -pkeyopt digest:sha256 \
      -keyform PEM \
      -in token3.bin \
      -out token3.sig.bin
```

6. Ahora tiene las firmas binarias de los tokens. Codifíquelas con base64 y vuelva a colocarlas en su archivo del token.

```
$ base64 -w0 token1.sig.bin > token1.sig.b64
$ base64 -w0 token2.sig.bin > token2.sig.b64
$ base64 -w0 token3.sig.bin > token3.sig.b64
```

7. Por último, vuelva a copiar y pegar los valores de base64 en su archivo del token:

```
{
  "version": "2.0",
  "tokens": [
    {
      "unsigned": "1jqwxb9bJOUUQLiNb7mxXS1uBJsEXh0B9nj05BqnPsE=",
      "signed": "eiw3fZeCKIY50C4zPeg9Rt90M1Qlq3WlJh6Yw7xXm4nF6e9ETLE39+9M
+rUqDWMRZjaBfaMbg5d9yDkz5p13U7ch2tlF9LoYabsWutkT014KRq/rcYMvFsU9n/Ey/
TK0PVaxLN42X+pebV4juwMhN4mK4CzdFAJgM+UGBOj4yB9recpOBB9K8QFSpJZALSEdDgUc/
mS1eDq3rU0int6+4NKuLQjpR
+LSEIWRZ6g6+MND2vXGskxHjadCQ09L7Tz8VcWjKDbxJcBiGKvkqyozl9zrGo8fA3WHBmwiAgS61Merx77ZGY4PFR37
YMSC14prCN15DtMRv2xA1SGSb4w=="
    },
    {
      "unsigned": "LMMFc34ASPnvNPFzBbMbr9FProS/Zu2P8zF/xzk5hVQ=",
      "signed": "HBImKnHmw+6R2TpFEpfiAg4+hu2pFNwn43ClhKPkn2higbEhUD0JVi
+4MerSyvU/NN79iWVxDvJ9Ito+jpiRQjTfTGEoIteyuAr1v/Bzh+HjmrO53OQpZaJ/VXGIgApD0myuu/
ZGNKQTCSkkL7+V81FG7yR1Nm22jUeGa735zvm/E+cenvZdy0VVx6A7WeWrl3JEKKBweHbi+7BwbaW
+PTdCuIRd4Ug76Sy+cFhsvcG1k7cMwDh8MgXzIZ2m1f/hdy2j8qAxORTLlmwyUOYvPYOvUhc
+s83hx36QpGwGcD7RA0bPT5OrTx7PHd0N1CL+Wwy91We8yIOFBS6nxo1R7w=="
    },
    {
      "unsigned": "dzeHbwhiVXQqcUGj563z51/7sLUdxjL93SbOUyZRjH8=",
      "signed": "VgQPvrTsvGljVBFxHnswduq16x8ZrnxfcYVYGf/
N7gEzI4At3GDs2EVZWTRdvS0uGHdkFYp1apHgJZ7PDVmGcTkIXVD2lFYppcgNlSzkYlftr5EOjqS9ZjYEqgGuB4g//
MxaBaRbJai/6BlcE92NIdBusTtreIm3yTpjIXNAVoeRSnkfuw7wZcL96QoklNb1WUuSHw
+psUyeIVtIwFMHEfFoRC0t
+VhmnlnFnkjGPb9W3Aprw2dRRvFM3R2ZTDvMCiOYDzUCd43GftGq2LfxH3qSD51oFHglHQVOY0jyVzzlAvub5HQdtOQ
    }
  ]
}
```

8. Ahora que su archivo del token tiene todas las firmas necesarias, puede continuar. Introduzca el nombre del archivo que contiene los tokens firmados y pulse la tecla Intro. Ahora debería iniciar sesión correctamente.

```
aws-cloudhsm > login --username <USERNAME> --role <ROLE> mfa-token-sign --token
 unsigned-tokens.json
Enter password:
Enter signed token file path (press enter if same as the unsigned token file):
{
  "error_code": 0,
```

```
  "data": {
    "username": "<USERNAME>",
    "role": "<ROLE>"
  }
}
```

Rotación de las claves para los usuarios con la MFA activada

Siga estos pasos para crear usuarios con la MFA activada.

<result>

Ha firmado el archivo del token con formato JSON generado con su clave privada y ha registrado una nueva clave pública de MFA.
</result>

1. Utilice la CLI de CloudHSM para iniciar sesión en el HSM como cualquier administrador o como el usuario específico que tiene habilitada la MFA (consulte el apartado Iniciar sesión de usuarios con MFA habilitada para obtener más información).
2. A continuación, ejecute el comando para cambiar su estrategia de MFA. Debe proporcionar el parámetro `--token`. Este parámetro especifica un archivo en el que se escribirán tokens sin firmar.

```
aws-cloudhsm > user change-mfa token-sign --token unsigned-tokens.json --
username <USERNAME> --role crypto-user --change-quorum
Enter password:
Confirm password:
```

3. Identifique el archivo con los tokens sin firmar que deben firmarse: `unsigned-tokens.json`. La cantidad de tokens de este archivo depende de la cantidad de HSM del clúster. Cada token representa un HSM. Este archivo tiene formato JSON y contiene tokens que deben firmarse para demostrar que tiene una clave privada. Esta será la nueva clave privada del nuevo par de claves pública/privada de RSA que desee usar para rotar la clave pública actualmente registrada.

```
$cat unsigned-tokens.json
{
  "version": "2.0",
  "tokens": [
    {
      "unsigned": "Vtf/9QOFY45v/E1osvpEMr59JsnP/hLDm4ItOO2vqL8=",
```

```
      "signed": ""
    },
    {
      "unsigned": "wVbC0/5IKwjyZK2NBpdFLyI7BiayZ24YcdUdlcxLwZ4=",
      "signed": ""
    },
    {
      "unsigned": "z6aW9RzErJBL5KqFG5h8lhTVt9oLbxppjod0Ebysydw=",
      "signed": ""
    }
  ]
}
```

4. Firme estos tokens con la clave privada que creó anteriormente durante la configuración. En primer lugar, debe extraer y decodificar los tokens codificados en base64.

```
$ echo "Vtf/9QOFY45v/E1osvpEMr59JsnP/hLDm4ItOO2vqL8=" > token1.b64
$ echo "wVbC0/5IKwjyZK2NBpdFLyI7BiayZ24YcdUdlcxLwZ4=" > token2.b64
$ echo "z6aW9RzErJBL5KqFG5h8lhTVt9oLbxppjod0Ebysydw=" > token3.b64
$ base64 -d token1.b64 > token1.bin
$ base64 -d token2.b64 > token2.bin
$ base64 -d token3.b64 > token3.bin
```

5. Ahora tiene tokens binarios. Fírmelos con la clave privada de RSA que creó anteriormente durante la configuración.

```
$ openssl pkeyutl -sign \
      -inkey officer1.key \
      -pkeyopt digest:sha256 \
      -keyform PEM \
      -in token1.bin \
      -out token1.sig.bin
$ openssl pkeyutl -sign \
      -inkey officer1.key \
      -pkeyopt digest:sha256 \
      -keyform PEM \
      -in token2.bin \
      -out token2.sig.bin
$ openssl pkeyutl -sign \
      -inkey officer1.key \
      -pkeyopt digest:sha256 \
```

```
      -keyform PEM \
      -in token3.bin \
      -out token3.sig.bin
```

6. Ahora tiene las firmas binarias de los tokens. Codifíquelas con base64 y vuelva a colocarlas en su archivo del token.

```
$ base64 -w0 token1.sig.bin > token1.sig.b64
$ base64 -w0 token2.sig.bin > token2.sig.b64
$ base64 -w0 token3.sig.bin > token3.sig.b64
```

7. Por último, vuelva a copiar y pegar los valores de base64 en su archivo del token:

```
{
  "version": "2.0",
  "tokens": [
    {
      "unsigned": "1jqwxb9bJOUUQLiNb7mxXS1uBJsEXh0B9nj05BqnPsE=",
      "signed": "eiw3fZeCKIY50C4zPeg9Rt90M1Qlq3WlJh6Yw7xXm4nF6e9ETLE39+9M
+rUqDWMRZjaBfaMbg5d9yDkz5p13U7ch2tlF9LoYabsWutkT014KRq/rcYMvFsU9n/Ey/
TK0PVaxLN42X+pebV4juwMhN4mK4CzdFAJgM+UGBOj4yB9recpOBB9K8QFSpJZALSEdDgUc/
mS1eDq3rU0int6+4NKuLQjpR
+LSEIWRZ6g6+MND2vXGskxHjadCQ09L7Tz8VcWjKDbxJcBiGKvkqyozl9zrGo8fA3WHBmwiAgS61Merx77ZGY4PFR37
YMSC14prCN15DtMRv2xA1SGSb4w=="
    },
    {
      "unsigned": "LMMFc34ASPnvNPFzBbMbr9FProS/Zu2P8zF/xzk5hVQ=",
      "signed": "HBImKnHmw+6R2TpFEpfiAg4+hu2pFNwn43ClhKPkn2higbEhUD0JVi
+4MerSyvU/NN79iWVxDvJ9Ito+jpiRQjTfTGEoIteyuAr1v/Bzh+HjmrO53OQpZaJ/VXGIgApD0myuu/
ZGNKQTCSkkL7+V81FG7yR1Nm22jUeGa735zvm/E+cenvZdy0VVx6A7WeWrl3JEKKBweHbi+7BwbaW
+PTdCuIRd4Ug76Sy+cFhsvcG1k7cMwDh8MgXzIZ2m1f/hdy2j8qAxORTLlmwyUOYvPYOvUhc
+s83hx36QpGwGcD7RA0bPT5OrTx7PHd0N1CL+Wwy91We8yIOFBS6nxo1R7w=="
    },
    {
      "unsigned": "dzeHbwhiVXQqcUGj563z51/7sLUdxjL93SbOUyZRjH8=",
      "signed": "VgQPvrTsvGljVBFxHnswduq16x8ZrnxfcYVYGf/
N7gEzI4At3GDs2EVZWTRdvS0uGHdkFYp1apHgJZ7PDVmGcTkIXVD2lFYppcgNlSzkYlftr5EOjqS9ZjYEqgGuB4g//
MxaBaRbJai/6BlcE92NIdBusTtreIm3yTpjIXNAVoeRSnkfuw7wZcL96QoklNb1WUuSHw
+psUyeIVtIwFMHEfFoRC0t
+VhmnlnFnkjGPb9W3Aprw2dRRvFM3R2ZTDvMCiOYDzUCd43GftGq2LfxH3qSD51oFHglHQVOY0jyVzzlAvub5HQdtOQ
    }
```

```
  ]
}
```

8. Ahora que su archivo del token tiene todas las firmas necesarias, puede continuar. Introduzca el nombre del archivo que contiene los tokens firmados y pulse la tecla Intro. Por último, introduzca la ruta de su clave pública. Ahora verá lo siguiente como parte del resultado de la lista de usuarios.

```
Enter signed token file path (press enter if same as the unsigned token file):
Enter public key PEM file path:officer1.pub
{
  "error_code": 0,
  "data": {
    "username": "<USERNAME>",
    "role": "crypto-user"
  }
}
```

   Ahora ha configurado su usuario con MFA.

```
{
    "username": "<USERNAME>",
    "role": "crypto-user",
    "locked": "false",
    "mfa": [
      {
        "strategy": "token-sign",
        "status": "enabled"
      }
    ],
    "cluster-coverage": "full"
},
```

Anulación del registro de una clave pública de MFA para los usuarios administradores cuando la clave pública de MFA esté registrada

Siga estos pasos para anular el registro de una clave pública de MFA para los usuarios administradores cuando la clave pública de MFA esté registrada.

1. Utilice la CLI de CloudHSM para iniciar sesión en el HSM como administrador con MFA habilitada.
2. Utilice el comando user change-mfa token-sign para eliminar la MFA de un usuario.

```
aws-cloudhsm > user change-mfa token-sign --username <USERNAME> --role admin --
deregister --change-quorum
Enter password:
Confirm password:
{
  "error_code": 0,
  "data": {
    "username": "<USERNAME>",
    "role": "admin"
  }
}
```

Referencia del archivo del token

El archivo del token que se genera al registrar una clave pública de MFA o al intentar iniciar sesión mediante MFA consta de lo siguiente:

- Tokens: una matriz de pares de tokens firmados o sin firmar codificados en base64 en forma de objetos literales JSON.
- Sin firmar: un token codificado en base64 y cifrado en SHA256.
- Signed: un token firmado (firma) codificado en base64 del token sin firmar que emplea la clave privada RSA de 2048 bits.

```
{
  "version": "2.0",
  "tokens": [
    {
      "unsigned": "1jqwxb9bJOUUQLiNb7mxXS1uBJsEXh0B9nj05BqnPsE=",
      "signed": "eiw3fZeCKIY50C4zPeg9Rt90M1Qlq3WlJh6Yw7xXm4nF6e9ETLE39+9M
+rUqDWMRZjaBfaMbg5d9yDkz5p13U7ch2tlF9LoYabsWutkT014KRq/rcYMvFsU9n/Ey/TK0PVaxLN42X
+pebV4juwMhN4mK4CzdFAJgM+UGBOj4yB9recpOBB9K8QFSpJZALSEdDgUc/mS1eDq3rU0int6+4NKuLQjpR
+LSEIWRZ6g6+MND2vXGskxHjadCQ09L7Tz8VcWjKDbxJcBiGKvkqyozl9zrGo8fA3WHBmwiAgS61Merx77ZGY4PFR37+j/
YMSC14prCN15DtMRv2xA1SGSb4w=="
    },
    {
```

```
      "unsigned": "LMMFc34ASPnvNPFzBbMbr9FProS/Zu2P8zF/xzk5hVQ=",
      "signed": "HBImKnHmw+6R2TpFEpfiAg4+hu2pFNwn43ClhKPkn2higbEhUD0JVi
+4MerSyvU/NN79iWVxDvJ9Ito+jpiRQjTfTGEoIteyuAr1v/Bzh+HjmrO53OQpZaJ/VXGIgApD0myuu/
ZGNKQTCSkkL7+V81FG7yR1Nm22jUeGa735zvm/E+cenvZdy0VVx6A7WeWrl3JEKKBweHbi+7BwbaW
+PTdCuIRd4Ug76Sy+cFhsvcG1k7cMwDh8MgXzIZ2m1f/hdy2j8qAxORTLlmwyUOYvPYOvUhc
+s83hx36QpGwGcD7RA0bPT5OrTx7PHd0N1CL+Wwy91We8yIOFBS6nxo1R7w=="
    },
    {
      "unsigned": "dzeHbwhiVXQqcUGj563z51/7sLUdxjL93SbOUyZRjH8=",
      "signed": "VgQPvrTsvGljVBFxHnswduq16x8ZrnxfcYVYGf/
N7gEzI4At3GDs2EVZWTRdvS0uGHdkFYp1apHgJZ7PDVmGcTkIXVD2lFYppcgNlSzkYlftr5EOjqS9ZjYEqgGuB4g//
MxaBaRbJai/6BlcE92NIdBusTtreIm3yTpjIXNAVoeRSnkfuw7wZcL96QoklNb1WUuSHw
+psUyeIVtIwFMHEfFoRC0t
+VhmnlnFnkjGPb9W3Aprw2dRRvFM3R2ZTDvMCiOYDzUCd43GftGq2LfxH3qSD51oFHglHQVOY0jyVzzlAvub5HQdtOQdErI
    }
  ]
}
```


Usar la CLI de CloudHSM para gestionar la autenticación de cuórum (control de acceso M de N)

Los HSM de su AWS CloudHSM clúster admiten la autenticación de quórum, que también se conoce como control de acceso M of N. Con la autenticación de cuórum ningún usuario único del HSM puede realizar operaciones controladas mediante cuórum en el HSM. En su lugar, para llevar a cabo estas operaciones debe cooperar un número mínimo de usuarios del HSM (al menos 2). Con la autenticación de cuórum, puede añadir una capa adicional de protección al exigir la aprobación de más de un usuario del HSM.

La autenticación de cuórum puede controlar las siguientes operaciones:

- Gestión de usuarios de HSM por el administrador: creación y eliminación de usuarios de HSM y cambio de la contraseña de otro usuario de HSM. Para obtener más información, consulte Uso de la autenticación de cuórum para administradores.

Los siguientes temas contienen más información acerca de la autenticación de cuórum en AWS CloudHSM.

Temas

- Descripción general de la estrategia de autenticación de cuórum con firma simbólica
- Detalles adicionales sobre la autenticación de cuórum

- Nombres y tipos de servicios que admiten la autenticación de cuórum
- Uso de la autenticación de cuórum para administradores: configuración por primera vez
- Uso de la autenticación de cuórum para administradores
- Cambio del valor mínimo de cuórum para los administradores

Descripción general de la estrategia de autenticación de cuórum con firma simbólica

En los pasos siguientes se resumen los procesos de autenticación de cuórum. Para informarse de las herramientas y los específicos, consulte Uso de la autenticación de cuórum para administradores.

1. Cada usuario del HSM crea una clave asimétrica para la firma. Los usuarios lo hacen fuera del HSM, teniendo cuidado de proteger la clave adecuadamente.
2. Cada usuario del HSM se conecta al HSM y registra la parte pública de su clave de firma (la clave pública) en el HSM.
3. Cuando un usuario del HSM quiere realizar una operación controlada mediante cuórum, inicia sesión en el HSM y obtiene un token de cuórum.
4. El usuario del HSM pasa el token de cuórum a uno o varios usuarios de ese HSM y les pide su aprobación.
5. Los otros usuarios del HSM dan su aprobación utilizando sus claves para firmar criptográficamente el token de cuórum. Esto se produce fuera del HSM.
6. Cuando el usuario del HSM tiene el número de aprobaciones requerido, el mismo usuario inicia sesión en el HSM y ejecuta la operación controlada por cuórum con el argumento --approval, proporcionando el archivo del token de cuórum firmado, que contiene todas las aprobaciones (firmas) necesarias.
7. El HSM utiliza las claves públicas registradas de cada firmante para verificar las firmas. Si las firmas son válidas, el HSM aprueba el token y se lleva a cabo la operación controlada por cuórum.

Detalles adicionales sobre la autenticación de cuórum

Tenga en cuenta la siguiente información adicional acerca del uso de la autenticación de cuórum en AWS CloudHSM.

- Un usuario del HSM puede firmar su propio token de cuórum; es decir, el usuario solicitante puede proporcionar una de las aprobaciones exigidas para la autenticación de cuórum.

- Elija el número mínimo de aprobadores de cuórum para las operaciones controladas mediante cuórum. El número más pequeño que puede elegir es dos (2) y el número más grande que puede elegir es ocho (8).
- El HSM puede almacenar hasta 1024 tokens de cuórum. Si el HSM ya tiene 1024 tokens cuando intenta crear uno nuevo, el HSM eliminará uno de los tokens que haya caducado. De forma predeterminada, los tokens caducan diez minutos después de su creación.
- Si la MFA está habilitada, el clúster usa la misma clave para la autenticación de cuórum y para la autenticación multifactor (MFA) Para obtener más información sobre el uso de la autenticación de cuórum y la 2FA, consulte el apartado Uso de la CLI de CloudHSM para gestionar el MFA.
- Cada HSM solo puede contener un token por servicio a la vez.

Nombres y tipos de servicios que admiten la autenticación de cuórum

Servicios de administración: la autenticación de cuórum se utiliza para los servicios con privilegios de administrador, como la creación y eliminación de usuarios, el cambio de las contraseñas de los usuarios, la configuración de los valores de cuórum y la desactivación de las capacidades de cuórum y MFA.

Además, cada tipo de servicio se divide en un nombre de servicio válido que contiene un conjunto específico de operaciones de servicio compatibles con cuórum que se pueden realizar.

Nombre del servicio	Tipo de servicio	Operaciones de servicio
usuario	Administrador	• user create • user delete • user change-password • user change-mfa
quorum	Administrador	• signo simbólico de quórum set-quorum-value

Uso de la autenticación de cuórum para administradores: configuración por primera vez

Los siguientes temas describen los pasos que debe seguir para configurar su módulo de seguridad de hardware (HSM) de forma que los administradores puedan usar la autenticación de cuórum. Debe seguir estos pasos solo una vez cuando configure por primera vez la autenticación de cuórum para

administradores. Después de completar estos pasos, consulte Uso de la autenticación de cuórum para administradores.

Temas

- Requisitos previos
- Creación y registro de una clave de firma
- Cómo establecer el valor mínimo de cuórum en el HSM

Requisitos previos

Para comprender este ejemplo, debe conocer la CLI de CloudHSM. En este ejemplo, el AWS CloudHSM clúster tiene dos HSM, cada uno con los mismos administradores, como se muestra en el siguiente resultado del comando. user list Para obtener más información sobre la creación de usuarios, consulte Utilización de la CLI de CloudHSM.

```
aws-cloudhsm>user list
{
  "error_code": 0,
  "data": {
    "users": [
      {
        "username": "admin",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [],
        "cluster-coverage": "full"
      },
      {
        "username": "admin2",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [],
        "cluster-coverage": "full"
      },
      {
        "username": "admin3",
        "role": "admin",
        "locked": "false",
        "mfa": [],
```

```
        "quorum": [],
        "cluster-coverage": "full"
      },
      {
        "username": "admin4",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [],
        "cluster-coverage": "full"
      },
      {
        "username": "app_user",
        "role": "internal(APPLIANCE_USER)",
        "locked": "false",
        "mfa": [],
        "quorum": [],
        "cluster-coverage": "full"
      }
    ]
  }
}
```

Creación y registro de una clave de firma

Para usar la autenticación de cuórum, cada administrador debe completar todos los pasos indicados:

Temas

- Creación de un par de claves RSA
- Creación y firma de un token de registro
- Cómo registrar una clave pública con el HSM

Creación de un par de claves RSA

Hay muchas formas de crear y proteger un par de claves. El siguiente ejemplo muestra cómo hacerlo con OpenSSL.

Example : creación de una clave privada con OpenSSL

El siguiente ejemplo ilustra cómo utilizar OpenSSL para crear una clave RSA de 2048 bits que está protegida por una frase de contraseña. Para utilizar este ejemplo, sustituya *<admin.key>* por el nombre del archivo donde desea almacenar la clave.

```
$ openssl genrsa -out <admin.key> -aes256 2048
Generating RSA private key, 2048 bit long modulus
.....................................+++
.+++
e is 65537 (0x10001)
Enter pass phrase for admin.key:
Verifying - Enter pass phrase for admin.key:
```

A continuación, genere la clave pública usando la clave privada que acaba de crear.

Example : creación de una clave pública con OpenSSL

El siguiente ejemplo muestra cómo usar OpenSSL para crear una clave pública a partir de la clave privada que acaba de crear.

```
$ openssl rsa -in admin.key -outform PEM -pubout -out admin1.pub
Enter pass phrase for admin.key:
writing RSA key
```

Creación y firma de un token de registro

Cree un token y fírmelo con la clave privada que acaba de generar en el paso anterior.

Example Creación de un token de registro

1. Use el siguiente comando para iniciar la CLI de CloudHSM:

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm-cli interactive
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
   ```

2. Cree un token de registro ejecutando el comando quorum token-sign generate:

```
aws-cloudhsm > quorum token-sign generate --service registration --token /path/
tokenfile
{
  "error_code": 0,
  "data": {
    "path": "/path/tokenfile"
  }
}
```

3. El comando quorum token-sign generate genera un token de registro en la ruta de archivo especificada. Inspeccione el archivo del token:

```
$ cat /path/tokenfile{
  "version": "2.0",
  "tokens": [
    {
      "approval_data": <approval data in base64 encoding>,
      "unsigned": <unsigned token in base64 encoding>,
      "signed": ""
    }
  ]
}
```

El archivo del token consta de lo siguiente:

- approval_data: un token de datos aleatorios codificado en base64 cuyos datos sin procesar no exceden el máximo de 245 bytes.
- unsigned: un token de approval_data codificado en base64 y con hash SHA-256
- signed: un token firmado (firma) codificado en base64 del token sin firmar que emplea la clave privada RSA de 2048 bits generada anteriormente con OpenSSL.

Firme el token sin firmar con la clave privada para demostrar que tiene acceso a la clave privada. Necesitará completar el archivo del token de registro con una firma y la clave pública para registrar al administrador como usuario del quórum en el clúster. AWS CloudHSM

Example : firme el token de registro sin firmar.

1. Decodifique el token sin firmar codificado en base64 e introdúzcalo en un archivo binario:

```
$ echo -n '6BMUj6mUjjko6ZLCEdzGlWpR5sILhFJfqhW1ej3Oq1g=' | base64 -d > admin.bin
```

2. Use OpenSSL y la clave privada para firmar el token de registro ahora binario sin firmar, y cree un archivo de firma binaria:

```
$ openssl pkeyutl -sign \
-inkey admin.key \
-pkeyopt digest:sha256 \
-keyform PEM \
-in admin.bin \
-out admin.sig.bin
```

3. Codifique la firma binaria en base64:

```
$ base64 -w0 admin.sig.bin > admin.sig.b64
```

4. Copie y pegue la firma codificada en base64 en el archivo del token:

```
{
  "version": "2.0",
  "tokens": [
    {
      "approval_data": <approval data in base64 encoding>,
      "unsigned": <unsigned token in base64 encoding>,
      "signed": <signed token in base64 encoding>
    }
  ]
}
```

Cómo registrar una clave pública con el HSM

Tras crear una clave, el administrador debe registrar la clave pública en el AWS CloudHSM clúster.

Para registrar una clave pública en el HSM

1. Use el siguiente comando para iniciar la CLI de CloudHSM:

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm-cli interactive
   ```

Windows

```
C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
```

2. Inicie sesión en la CLI de CloudHSM como administrador.

```
aws-cloudhsm > login --username admin --role admin
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

3. Utilice el comando user change-quorum token-sign register para registrar la clave pública. Para obtener más información, consulte el siguiente ejemplo o utilice el comando help user change-quorum token-sign register.

Example — Registrar una clave pública con el AWS CloudHSM clúster

En el siguiente ejemplo, se explica cómo se utiliza el comando user change-quorum token-sign register la CLI de CloudHSM para registrar una clave pública del administrador con el HSM. Para utilizar este comando, el administrador tiene que haber iniciado sesión en el HSM. Reemplace estos valores por sus propios valores:

```
aws-cloudhsm > user change-quorum token-sign register --public-key </path/admin.pub> --
signed-token </path/tokenfile>
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

> **Note**
>
> /path/admin.pub: ruta del archivo PEM de clave pública

Obligatorio: sí
/path/tokenfile: ruta del archivo con el token firmado por la clave privada del usuario
Obligatorio: sí

Una vez que todos los administradores hayan registrado sus claves públicas, el resultado del comando user list mostrará lo siguiente en el campo de cuórum e indicará la estrategia de cuórum habilitada, tal como se muestra a continuación:

```
aws-cloudhsm > user list
{
  "error_code": 0,
  "data": {
    "users": [
      {
        "username": "admin",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
        "cluster-coverage": "full"
      },
      {
        "username": "admin2",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
        "cluster-coverage": "full"
      },
      {
        "username": "admin3",
```

```
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
        "cluster-coverage": "full"
      },
      {
        "username": "admin4",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
        "cluster-coverage": "full"
      },
      {
        "username": "app_user",
        "role": "internal(APPLIANCE_USER)",
        "locked": "false",
        "mfa": [],
        "quorum": [],
        "cluster-coverage": "full"
      }
    ]
  }
}
```

Cómo establecer el valor mínimo de cuórum en el HSM

Para utilizar autenticación de cuórum para administradores, un administrador debe iniciar sesión en el HSM y, a continuación, establecer el valor mínimo de cuórum. Este es el número mínimo de aprobaciones del administrador necesarias para realizar las operaciones de administración de usuarios de HSM. Cualquier administrador en el HSM puede establecer el valor mínimo de cuórum, incluidos los administradores que no han registrado una clave para firmar. Puede cambiar el valor

mínimo del cuórum en cualquier momento; para obtener más información, consulte Cambio del valor mínimo.

Para establecer el valor mínimo de cuórum en el HSM

1. Use el siguiente comando para iniciar la CLI de CloudHSM:

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm-cli interactive
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
   ```

2. Inicie sesión en la CLI de CloudHSM como administrador.

   ```
   aws-cloudhsm > login --username admin --role admin
   Enter password:
   {
     "error_code": 0,
     "data": {
       "username": "admin",
       "role": "admin"
     }
   }
   ```

3. Utilice el comando signo simbólico de quórum set-quorum-value para establecer el valor mínimo de cuórum. Para obtener más información, consulte el siguiente ejemplo o utilice el comando help quorum token-sign set-quorum-value.

Example : establecimiento del valor mínimo de cuórum del HSM

Este ejemplo utiliza un valor mínimo de cuórum de dos (2). Puede elegir cualquier valor entre dos (2) y ocho (8), hasta alcanzar el número total de administradores en el HSM. En este ejemplo, el HSM tiene cuatro (4) administradores, por lo que el valor máximo posible es cuatro (4).

Para utilizar el siguiente comando de ejemplo, reemplace el número final (*<2>*) por el valor mínimo de cuórum preferido.

```
aws-cloudhsm > quorum token-sign set-quorum-value --service user --value <2>
```

```
{
  "error_code": 0,
  "data": "Set quorum value successful"
}
```

En este ejemplo, el servicio identifica el servicio HSM cuyo valor mínimo de quórum está establecido. El comando signo simbólico de quórum list-quorum-values muestra los tipos, nombres y descripciones de los servicios de HSM que se incluyen en el servicio.

Servicios de administración: la autenticación de cuórum se utiliza para los servicios con privilegios de administrador, como la creación y eliminación de usuarios, el cambio de las contraseñas de los usuarios, la configuración de los valores de cuórum y la desactivación de las capacidades de cuórum y MFA.

Además, cada tipo de servicio se divide en un nombre de servicio válido que contiene un conjunto específico de operaciones de servicio compatibles con cuórum que se pueden realizar.

Nombre del servicio	Tipo de servicio	Operaciones de servicio
usuario	Administrador	• user create • user delete • user change-password • user change-mfa
quorum	Administrador	• signo simbólico de quórum set-quorum-value

Utilice el comando quorum token-sign list-quorum-values para obtener el valor mínimo de cuórum del servicio.

```
aws-cloudhsm > quorum token-sign list-quorum-values
{
  "error_code": 0,
  "data": {
    "user": 2,
    "quorum": 1
  }
}
```

El resultado del comando quorum token-sign list-quorum-values anterior muestra que el valor mínimo de cuórum para las operaciones de gestión de usuarios de HSM ahora es dos (2). Después de completar estos pasos, consulte Uso del cuórum (M de N).

Uso de la autenticación de cuórum para administradores

Un administrador del HSM puede configurar la autenticación de quórum para las siguientes operaciones del clúster: AWS CloudHSM

- user create
- user delete
- user change-password
- user change-mfa

Una vez configurado el AWS CloudHSM clúster para la autenticación de quórum, los administradores no pueden realizar las operaciones de administración de usuarios del HSM por sí mismos. El siguiente ejemplo muestra el resultado cuando un administrador intenta crear un usuario nuevo en el HSM. La ejecución del comando devuelve un error, indicando la obligatoriedad de la autenticación de cuórum.

```
aws-cloudhsm > user create --username user1 --role crypto-user
Enter password:
Confirm password:
{
  "error_code": 1,
  "data": "Quorum approval is required for this operation"
}
```

Para realizar una operación de administración de usuarios de HSM, un administrador debe completar las siguientes tareas:

Temas

- Cómo obtener un token de cuórum
- Obtención de firmas de los administradores responsables de la aprobación
- Apruebe el token del AWS CloudHSM clúster y ejecute una operación de administración de usuarios

Cómo obtener un token de cuórum

En primer lugar, el administrador deberá usar la CLI de CloudHSM para solicitar un token de cuórum.

Para obtener un token de cuórum

1. Use el siguiente comando para iniciar la CLI de CloudHSM.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm-cli interactive
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
   ```

2. Ejecute el comando login e inicie sesión en el clúster como administrador.

   ```
   aws-cloudhsm>login --username admin --role admin
   ```

3. Utilice el comando quorum token-sign generate para generar un token de cuórum. Para obtener más información, consulte el siguiente ejemplo o utilice el comando help quorum token-sign generate.

Example : genera un token de cuórum.

Este ejemplo obtiene un token de cuórum para el administrador con nombre de usuario `admin` y guarda el token en un archivo denominado `admin.token`. Para utilizar el comando de ejemplo, sustituya estos valores por los suyos:

- *`<admin>`*: el nombre del administrador que obtiene el token. Debe ser el mismo administrador que ha iniciado sesión en el HSM y está ejecutando este comando.
- *`<admin.token>`*: el nombre del archivo que se debe utilizar para almacenar el token de cuórum.

En el siguiente comando, `user` identifica el nombre del servicio para el que puede utilizar el token que está obteniendo. En este caso, el token es para las operaciones de administración de usuarios de HSM (servicio `user`).

```
aws-cloudhsm > login --username <ADMIN> --role <ADMIN> --password <PASSWORD>
```

```
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}

aws-cloudhsm > quorum token-sign generate --service user --token </path/admin.token>
{
  "error_code": 0,
  "data": {
    "path": "/home/tfile"
  }
}
```

El comando quorum token-sign generate genera un token de cuórum de servicio de usuario en la ruta de archivo especificada. El archivo del token se puede inspeccionar:

```
$cat </path/admin.token>
{
  "version": "2.0",
  "approval_data": "AAEAAwAAABgAAAAAAAAAAJ9eFkfcP3mNzJAlfK
+OWbNhZG1pbgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABj5vbeAAAAAAAAAAAAAQADAAAAFQAAAAAAAAAW/
v5Euk83amq1fij0zyvD2FkbWluAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGPm9t4AAAAAAAAAAAABAAMAAAAUA
+b23gAAAAAAAAAA",
  "token": "0l2LZkmAHZyAc1hPhyckOoVW33aGrgG77qmDHWQ3CJ8=",
  "signatures": []
}
```

El archivo del token consta de lo siguiente:

- approval_data: un token de datos base64 sin procesar generado por el HSM.
- token: un token de approval_data codificado en base64 y con hash SHA-256
- signatures: un conjunto de tokens firmados (firmas) y codificados en base64 del token sin firmar, donde cada firma de un aprobador tiene la forma de un objeto JSON literal:

  ```
  {
        "username": "<APPROVER_USERNAME>",
        "signature": "<APPROVER_RSA2048_BIT_SIGNATURE>"
  }
  ```

La firma se crea cuando un aprobador usa su correspondiente clave privada RSA de 2048 bits cuya clave pública se ha registrado en el HSM.

Se puede confirmar que el token de cuórum del servicio de usuario generado existe en el clúster de CloudHSM ejecutando el comando quorum token-sign list:

```
aws-cloudhsm > quorum token-sign list
{
  "error_code": 0,
  "data": {
    "tokens": [
      {
        "username": "admin",
        "service": "user",
        "approvals-required": {
          "value": 2
        },
        "number-of-approvals": {
          "value": 0
        },
        "token-timeout-seconds": {
          "value": 597
        },
        "cluster-coverage": "full"
      }
    ]
  }
}
```

El tiempo `token-timeout-seconds` indica el tiempo de espera en segundos para que un token generado se apruebe antes de que caduque.

Obtención de firmas de los administradores responsables de la aprobación

Un administrador que tiene un token de cuórum debe obtener la aprobación del token de otros administradores. Para dar su aprobación, los otros administradores utilizan su clave de firma para firmar criptográficamente el token. Lo hacen fuera del HSM.

Existen muchas maneras diferentes de firmar el token. El siguiente ejemplo muestra cómo hacerlo con OpenSSL. Para utilizar otra herramienta de firma, asegúrese de que la herramienta utiliza la clave privada del administrador (clave de firma) para firmar un resumen SHA-256 del token.

Example : obtenga firmas de los administradores responsables de la aprobación.

En este ejemplo, el administrador que tiene el token (`admin`) necesita al menos dos (2) aprobaciones. Los siguientes comandos de ejemplo muestran cómo dos (2) administradores pueden utilizar OpenSSL para firmar el token criptográficamente.

1. Decodifique el token sin firmar codificado en base64 e introdúzcalo en un archivo binario:

```
$echo -n '0l2LZkmAHZyAc1hPhyckOoVW33aGrgG77qmDHWQ3CJ8=' | base64 -d > admin.bin
```

2. Use OpenSSL y la clave privada correspondiente del aprobador (`admin3`) para firmar el token sin firmar de cuórum, ahora binario, para el servicio de usuario y crear un archivo de firma binaria:

```
$openssl pkeyutl -sign \
-inkey admin3.key \
-pkeyopt digest:sha256 \
-keyform PEM \
-in admin.bin \
-out admin.sig.bin
```

3. Codifique la firma binaria en base64:

```
$base64 -w0 admin.sig.bin > admin.sig.b64
```

4. Por último, copie y pegue la firma codificada en base64 en el archivo de token, siguiendo el formato literal del objeto JSON especificado anteriormente para la firma del aprobador:

```
{
  "version": "2.0",
  "approval_data": "AAEAAwAAABgAAAAAAAAAAJ9eFkfcP3mNzJAlfK
+OWbNhZG1pbgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABj5vbeAAAAAAAAAAAAQADAAAAFQAAAAAAAAAW
v5Euk83amq1fij0zyvD2FkbWluAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGPm9t4AAAAAAAAAAABAAMAA
+b23gAAAAAAAAAA",
  "token": "0l2LZkmAHZyAc1hPhyckOoVW33aGrgG77qmDHWQ3CJ8=",
  "signatures": [
    {
      "username": "admin2",
      "signature": "O6qx7/mUaVkYYVr1PW7l8JJko+Kh3e8zBIqdk3tAiNy+1rW
+OsDtvYujhEU4aOFVLcrUFmyB/CX9OQmgJLgx/pyK+ZPEH+GoJGqk9YZ7X1nOXwZRP9g7hKV
+7XCtg9TuDFtHYWDpBfz2jWiu2fXfX4/
jTs4f2xIfFPIDKcSP8fhxjQ63xEcCf1jzGha6rDQMu4xUWWdtDgfT7um7EJ9dXNoHqLB7cTzphaubNaEFbFPXQ1siGm
```

```
ssktwyruGFLpXs1n0tJOEglGhx2qbYTs+omKWZdORl5WIWEXW3IXw/
Dg5vVObrNpvG0eZKO8nSMc27+cyPySc+ZbNw=="
    },
    {
      "username": "admin3",
      "signature": "O6qx7/mUaVkYYVr1PW7l8JJko+Kh3e8zBIqdk3tAiNy+1rW
+OsDtvYujhEU4aOFVLcrUFmyB/CX9OQmgJLgx/pyK+ZPEH+GoJGqk9YZ7X1nOXwZRP9g7hKV
+7XCtg9TuDFtHYWDpBfz2jWiu2fXfX4/
jTs4f2xIfFPIDKcSP8fhxjQ63xEcCf1jzGha6rDQMu4xUWWdtDgfT7um7EJ9dXNoHqLB7cTzphaubNaEFbFPXQ1siGm
ssktwyruGFLpXs1n0tJOEglGhx2qbYTs+omKWZdORl5WIWEXW3IXw/
Dg5vVObrNpvG0eZKO8nSMc27+cyPySc+ZbNw=="
    }
  ]
}
```


Apruebe el token del AWS CloudHSM clúster y ejecute una operación de administración de usuarios

Una vez que el administrador cuente con las aprobaciones o firmas necesarias, tal y como se detalla en la sección anterior, puede proporcionar ese token al clúster de AWS CloudHSM junto con una de las siguientes operaciones de administración de usuarios:

- créate
- eliminar
- change-password
- user change-mfa

Para obtener más información acerca del uso de estos comandos, consulte Utilización de la CLI de CloudHSM.

Durante la transacción, el token se aprobará dentro del AWS CloudHSM clúster y ejecutará la operación de administración de usuarios solicitada. El éxito de la operación de administración de usuarios depende, por un lado, de un token de cuórum válido y aprobado y, por otro, de una operación de administración de usuarios válida.

El administrador puede utilizar el token para una sola operación. Cuando dicha operación se realiza correctamente, el token ya no es válido. Para realizar otra operación de administración de usuarios de HSM, el administrador deberá repetir el proceso anteriormente descrito. Es decir, el administrador debe generar un nuevo token de cuórum, obtener firmas nuevas de los aprobadores y usar el token nuevo en el HSM con la administración de usuarios solicitada.

> **Note**
>
> El token de cuórum solo será válido mientras se mantenga la sesión actual iniciada. Si cierra sesión en la CLI de CloudHSM o si la red se desconecta, el token perderá su validez. Del mismo modo, un token autorizado solo puede usarse en la CLI de CloudHSM. No puede usarse para autenticarse en una aplicación diferente.

Example Creación de un nuevo usuario como administrador

En el siguiente ejemplo, un administrador conectado crea un nuevo usuario en el HSM.

```
aws-cloudhsm > user create --username user1 --role crypto-user --approval /path/
admin.token
Enter password:
Confirm password:
{
  "error_code": 0,
  "data": {
    "username": "user1",
    "role": "crypto-user"
  }
}
```

A continuación, el administrador introduce el comando user list para confirmar la creación del nuevo usuario:

```
aws-cloudhsm > user list{
  "error_code": 0,
  "data": {
    "users": [
      {
        "username": "admin",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
```

```
        "cluster-coverage": "full"
      },
      {
        "username": "admin2",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
        "cluster-coverage": "full"
      },
      {
        "username": "admin3",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
        "cluster-coverage": "full"
      },
      {
        "username": "admin4",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
        "cluster-coverage": "full"
      },
      {
        "username": "user1",
        "role": "crypto-user",
```

```
        "locked": "false",
        "mfa": [],
        "quorum": [],
        "cluster-coverage": "full"
      },
      {
        "username": "app_user",
        "role": "internal(APPLIANCE_USER)",
        "locked": "false",
        "mfa": [],
        "quorum": [],
        "cluster-coverage": "full"
      }
    ]
  }
}
```

Si el administrador intenta realizar otra operación de administración de usuarios de HSM, da error con un error de autenticación de cuórum.

```
aws-cloudhsm > user delete --username user1 --role crypto-user
{
  "error_code": 1,
  "data": "Quorum approval is required for this operation"
}
```

Como se muestra a continuación, el comando quorum token-sign list muestra que el administrador no tiene ningún token aprobado. Para realizar otra operación de administración de usuarios de HSM, el administrador debe generar un nuevo token de cuórum, conseguir nuevas firmas de los aprobadores y ejecutar la operación de administración de usuarios deseada con el argumento --approval, para proporcionar el token de cuórum que se aprobará y consumirá durante la ejecución de la operación de administración de usuarios.

```
aws-cloudhsm > quorum token-sign list
{
  "error_code": 0,
  "data": {
    "tokens": []
  }
}
```

Cambio del valor mínimo de cuórum para los administradores

Después de establecer el valor mínimo de cuórum para que los administradores puedan utilizar la autenticación de cuórum, es posible que desee cambiar el valor mínimo de cuórum. El HSM le permite cambiar el valor mínimo de cuórum solo cuando el número de aprobadores es igual o superior al actual valor mínimo de cuórum. Por ejemplo, si el valor mínimo del cuórum es dos (2), al menos dos (2) administradores deben aprobar el cambio del valor mínimo del cuórum.

> **Note**
>
> El valor de cuórum del servicio de usuario siempre debe ser inferior al valor de cuórum del servicio de cuórum. Para obtener información sobre los nombres de los servicios, como el servicio de cuórum y el servicio de usuario, consulte Nombres y tipos de servicios que admiten la autenticación de cuórum.

Para obtener aprobación de cuórum para cambiar el valor mínimo de cuórum, necesita un token de cuórum para quorum service usando el comando quorum token-sign set-quorum-value. Para generar un token de cuórum para el quorum service que utiliza el comando quorum token-sign set-quorum-value, el servicio de cuórum debe ser superior a uno (1). Esto significa que antes de que pueda cambiar el valor mínimo de cuórum para el servicio de usuario, puede que necesite cambiar el valor mínimo de cuórum para el servicio de cuórum.

Cómo cambiar el valor mínimo de cuórum para los administradores

1. Use el siguiente comando para iniciar el modo interactivo de la CLI de CloudHSM.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm-cli interactive
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
   ```

2. Ejecute el comando login e inicie sesión en el clúster como administrador.

   ```
   aws-cloudhsm>login --username <admin> --role admin
   ```

3. Utilice el comando quorum token-sign list-quorum-values para obtener los valores mínimos de cuórum para todos los nombres de servicio. Para más información, consulte el ejemplo siguiente.
4. Si el valor mínimo de quórum para el servicio de quórum es inferior al valor para el servicio de usuario, utilice el comando quorum token-sign set-quorum-value para cambiar el valor para el servicio de cuórum. Cambie el valor del servicio de cuórum a uno (1) que sea igual o superior al valor del servicio de usuario. Para obtener más información, consulte el siguiente ejemplo.
5. Genere un token de cuórum, teniendo cuidado de especificar el servicio de cuórum como el servicio para el que puede utilizar el token.
6. Obtenga aprobaciones (firmas) de otros administradores..
7. Apruebe el token en el AWS CloudHSM clúster y ejecute una operación de administración de usuarios. .
8. Utilice el comando quorum token-sign set-quorum-value para cambiar el valor mínimo de cuórum del servicio.

Example : obtener los valores mínimos de cuórum y cambiar el valor del servicio de quórum.

El siguiente comando de ejemplo muestra que el valor mínimo de cuórum para el servicio al usuario es actualmente dos (2).

```
aws-cloudhsm > quorum token-sign list-quorum-values{
  "error_code": 0,
  "data": {
    "user": 2,
    "quorum": 1
  }
}
```

Para cambiar el valor mínimo de cuórum para el servicio de cuórum, utilice el comando, quorum token-sign set-quorum-value estableciendo un valor que sea igual o superior al valor para el servicio de usuario. El siguiente ejemplo establece el valor mínimo de cuórum para el servicio de cuórum en dos (2), el mismo valor que se establece para el servicio de usuario.

```
aws-cloudhsm > quorum token-sign set-quorum-value --service quorum --value 2{
  "error_code": 0,
  "data": "Set quorum value successful"
}
```

El siguiente comando muestra que el valor mínimo de quórum es ahora dos (2) para el servicio de usuario y el servicio de cuórum.

```
aws-cloudhsm > quorum token-sign list-quorum-values{
  "error_code": 0,
  "data": {
    "user": 2,
    "quorum": 2
  }
}
```

Administrar usuarios de HSM con la Utilidad de administración CloudHSM (CMU)

En AWS CloudHSM, debe usar las herramientas de línea de comandos de CloudHSM CLI o CloudHSM Management Utility (CMU) para crear y administrar los usuarios de su HSM. La CLI de CloudHSM está diseñada para usarse con el SDK más reciente, mientras que la CMU está diseñada para usarse con los SDK anteriores.

Temas

- Más información sobre los usuarios de HSM
- Tabla de permisos de usuario de HSM
- Uso de la utilidad de administración de CloudHSM (CMU) para administrar usuarios
- Uso de la utilidad de administración (CMU) de CloudHSM para administrar la autenticación de dos factores (2FA) para los de responsables de criptografía
- Usar la Utilidad de administración de CloudHSM (CMU) para administrar la autenticación de cuórum (control de acceso M de N)

Más información sobre los usuarios de HSM

La mayoría de las operaciones que realiza en el HSM requieren las credenciales de un usuario de HSM. El HSM autentica a cada usuario del HSM y cada usuario del HSM tiene un tipo que determina las operaciones que puede realizar en el HSM como ese usuario.

> **Note**
>
> Los usuarios de HSM son distintos de los usuarios de IAM. Los usuarios de IAM que dispongan de credenciales correctas pueden crear HSM interactuando con los recursos a través de la API de AWS. Una vez creado el HSM, deberá introducir las credenciales de usuario de HSM para autenticar las operaciones del mismo.

Tipos de usuario

- Responsable de criptografía previa (PRECO)
- Responsable de criptografía (CO)
- Usuario de criptografía (CU)
- Usuario de dispositivos (AU)

Responsable de criptografía previa (PRECO)

Tanto en la utilidad de administración en la nube (CMU) como en la utilidad de administración de claves (KMU), el PRECO es un usuario temporal que solo existe en el primer HSM de un clúster de AWS CloudHSM . El primer HSM de un clúster nuevo contiene un usuario de PRECO, lo que indica que este clúster nunca se ha activado. Para activar un clúster, ejecute cloudhsm-cli y el comando cluster activate. Inicie sesión en el HSM y cambie la contraseña de PRECO. Al cambiar la contraseña, el usuario se convierte en un responsable de criptografía (CO).

Responsable de criptografía (CO)

Tanto en la utilidad de administración en la nube (CMU) como en la utilidad de administración de claves (KMU), un responsable de criptografía (CO) puede realizar operaciones de administración de usuarios. Por ejemplo, pueden crear y eliminar usuarios, así como cambiar las contraseñas de los usuarios. Para obtener más información sobre el usuario CO, consulte Tabla de permisos de usuario de HSM. Cuando se activa un clúster nuevo, el usuario cambia de responsable de criptografía previa (PRECO) a responsable de criptografía (CO).-->

Usuario de criptografía (CU)

Un usuario de criptografía (CU) puede realizar las siguientes operaciones de administración de claves y criptografía.

- Administración de claves: crear, eliminar, compartir, importar y exportar claves criptográficas.

- Operaciones criptográficas: utilizar las claves criptográficas para cifrado, descifrado, firma, verificación y mucho más.

Para obtener más información, consulte Tabla de permisos de usuario de HSM.

Usuario de dispositivos (AU)

El usuario del dispositivo (AU) puede realizar operaciones de clonación y sincronización en los HSM del clúster. AWS CloudHSM utiliza la AU para sincronizar los HSM de un clúster. AWS CloudHSM La AU existe en todos los HSM proporcionados por ellos y tiene AWS CloudHSM permisos limitados. Para obtener más información, consulte Tabla de permisos de usuario de HSM.

AWS no puede realizar ninguna operación en sus HSM. AWS no puede ver ni modificar sus usuarios o claves y no puede realizar ninguna operación criptográfica con esas claves.

Tabla de permisos de usuario de HSM

La siguiente tabla muestra las operaciones de HSM ordenadas por tipo de usuario o sesión de HSM que puede realizar la operación.

	Responsable de criptografía (CO)	Usuario de criptografía (CU)	Usuario de dispositivos (AU)	Sesiones no autenticadas
Obtener información básica del clúster[1]	Sí	Sí	Sí	Sí
Cambiar su propia contraseña	Sí	Sí	Sí	No aplicable
Cambiar la contraseña de cualquier usuario	Sí	No	No	No

	Responsable de criptografía (CO)	Usuario de criptografía (CU)	Usuario de dispositivos (AU)	Sesiones no autenticadas
Agregar o eliminar usuarios	 Sí	 No	 No	 No
Obtener el estado de sincronización[2]	 Sí	 Sí	 Sí	 No
Extraer o insertar objetos enmascarados[3]	 Sí	 Sí	 Sí	 No
Funciones de administración de claves[4]	 No	 Sí	 No	 No
Cifrar o descifrar	 No	 Sí	 No	 No
Firmar o verificar	 No	 Sí	 No	 No
Generar resúmenes y HMAC	 No	 Sí	 No	 No

- [1] La información básica sobre el clúster incluye el número de HSM que hay en el clúster y la dirección IP, el modelo, el número de serie, el ID de dispositivo, el ID de firmware, etc. de cada HSM.
- [2] El usuario puede obtener un conjunto de resúmenes (hashes) que se corresponden con las claves del HSM. Una aplicación puede comparar estos conjuntos de resúmenes para conocer el estado de la sincronización de los HSM de un clúster.
- [3] Los objetos enmascarados son claves que se cifran antes de salir del HSM. No se pueden descifrar fuera del HSM. Solo se descifran después de insertarlos en un HSM que se encuentra en el mismo clúster que el HSM del que se extrajeron. Una aplicación puede extraer e insertar objetos enmascarados para sincronizar los HSM de un clúster.
- [4] Las funciones de administración de claves incluyen la creación, eliminación, encapsulación y modificación de los atributos de las claves.

Uso de la utilidad de administración de CloudHSM (CMU) para administrar usuarios

En este tema se proporcionan step-by-step instrucciones sobre la administración de los usuarios del módulo de seguridad de hardware (HSM) con CloudHSM Management Utility (CMU), una herramienta de línea de comandos que se incluye con el SDK del cliente. Para obtener más información acerca de los usuarios de CMU o HSM, consulte Utilidad de administración de CloudHSM y Más información sobre los usuarios de HSM.

Secciones

- Comprender la administración de usuarios de HSM con CMU
- Descarga de la utilidad de administración de CloudHSM
- ¿Cómo administrar los usuarios de HSM con CMU?

Comprender la administración de usuarios de HSM con CMU

Para administrar usuarios de HSM, inicie sesión en el HSM con el nombre de usuario y la contraseña de un responsable de criptografía (CO). Solo los CO pueden administrar otros usuarios. El HSM contiene un CO predeterminado llamado admin. Usted estableció la contraseña para admin cuando activó el clúster.

Para utilizar la CMU, debe usar la herramienta de configuración para actualizar la configuración local. La CMU crea su propia conexión con el clúster y esta conexión no es compatible con el clúster. Para realizar un seguimiento de la información del clúster, la CMU mantiene un archivo de

configuración local. Esto significa que cada vez que utilice la CMU, primero debe actualizar el archivo de configuración ejecutando la herramienta de línea de comandos configurar con el parámetro `--cmu`. Si usa la versión 3.2.1 o anteriores de SDK de cliente, debe usar un parámetro diferente a `--cmu`. Para obtener más información, consulte the section called “Uso de la CMU con la versión 3.2.1 y anteriores de SDK de cliente”.

El parámetro `--cmu` solicita agregar la dirección IP de un HSM en su clúster. Si tiene varios HSM, puede usar cualquier dirección IP. Esto garantiza que la CMU pueda propagar cualquier cambio que realice en todo el clúster. Recuerde que la CMU usa el archivo local para rastrear la información del clúster. Si el clúster ha cambiado desde la última vez que utilizó la CMU desde un host concreto, debe añadir esos cambios al archivo de configuración local almacenado en ese host. Nunca añada ni elimine un HSM mientras esté utilizando la CMU.

Para obtener una dirección IP para un HSM (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Para cambiar la región de AWS, utilice el selector de regiones en la esquina superior derecha de la página.
3. Para abrir la página de detalles del clúster, en la tabla de clústeres, elija el ID del clúster.
4. Para obtener la dirección IP, vaya a la pestaña HSM y elija una de las direcciones IP que aparecen en la lista Dirección IP de ENI.

Para obtener una dirección IP para un HSM (CLI)

- Obtenga la dirección IP de un HSM mediante el describe-clusters comando de la CLI. En el resultado del comando, la dirección IP de los HSM son los valores de `EniIp`.

```
$  aws cloudhsmv2 describe-clusters


{
    "Clusters": [
        { ... }
            "Hsms": [
                {
...
                    "EniIp": "10.0.0.9",
...
                },
```

```
                {
...
                    "EniIp": "10.0.1.6",
...
```

Uso de la CMU con la versión 3.2.1 y anteriores de SDK de cliente

Con Client SDK 3.3.0, AWS CloudHSM se agregó la compatibilidad con el `--cmu` parámetro, lo que simplifica el proceso de actualización del archivo de configuración de la CMU. Si utiliza una versión de CMU de 3.2.1 o anterior del SDK de cliente, debe seguir utilizando los parámetros `-a` y `-m` para actualizar el archivo de configuración. Para obtener más información acerca de estos parámetros, consulte Herramientas de configuración .

Descarga de la utilidad de administración de CloudHSM

La última versión de CMU está disponible para las tareas de administración de usuarios de HSM, tanto si utiliza SDK 5 de cliente como SDK 3 de cliente.

Descarga e instalación de la CMU

- Descargue e instale la CMU.

  Amazon Linux

  ```
  $ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-mgmt-util-latest.el6.x86_64.rpm
  ```

  ```
  $ sudo yum install ./cloudhsm-mgmt-util-latest.el6.x86_64.rpm
  ```

  Amazon Linux 2

  ```
  $ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-mgmt-util-latest.el7.x86_64.rpm
  ```

  ```
  $ sudo yum install ./cloudhsm-mgmt-util-latest.el7.x86_64.rpm
  ```

CentOS 7.8+

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-mgmt-util-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-mgmt-util-latest.el7.x86_64.rpm
```

CentOS 8.3+

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-mgmt-util-latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-mgmt-util-latest.el8.x86_64.rpm
```

RHEL 7 (7.8+)

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-mgmt-util-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-mgmt-util-latest.el7.x86_64.rpm
```

RHEL 8 (8.3+)

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-mgmt-util-latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-mgmt-util-latest.el8.x86_64.rpm
```

Ubuntu 16.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/cloudhsm-mgmt-util_latest_amd64.deb
```

```
$ sudo apt install ./cloudhsm-mgmt-util_latest_amd64.deb
```

Ubuntu 18.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Bionic/
cloudhsm-mgmt-util_latest_u18.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-mgmt-util_latest_u18.04_amd64.deb
```

Windows Server 2012

1. Descargue la Utilidad de administración de CloudHSM.
2. Ejecute el instalador de CMU (AWSCloudHSMManagementUtil-latest.msi) con privilegios administrativos de Windows.

Windows Server 2012 R2

1. Descargue la Utilidad de administración de CloudHSM.
2. Ejecute el instalador de CMU (AWSCloudHSMManagementUtil-latest.msi) con privilegios administrativos de Windows.

Windows Server 2016

1. Descargue la Utilidad de administración de CloudHSM.
2. Ejecute el instalador de CMU (AWSCloudHSMManagementUtil-latest.msi) con privilegios administrativos de Windows.

¿Cómo administrar los usuarios de HSM con CMU?

En esta sección se incluyen comandos básicos para administrar los usuarios de HSM con CMU.

Cómo crear usuarios HSM

Utilice createUser para crear nuevos usuarios en el HSM. Debe iniciar sesión como CO para crear un usuario.

Cómo crear un nuevo usuario

1. Utilice la herramienta de configuración para actualizar la configuración de la CMU.

Linux

```
$ sudo /opt/cloudhsm/bin/configure --cmu <IP address>
```

Windows

```
C:\Program Files\Amazon\CloudHSM\bin\ configure.exe --cmu <IP address>
```

2. Iniciar CMU.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM> .\cloudhsm_mgmt_util.exe C:\ProgramData\Amazon
   \CloudHSM\data\cloudhsm_mgmt_util.cfg
   ```

3. Inicie sesión en HSM como usuario CO.

   ```
   aws-cloudhsm>loginHSM CO admin co12345
   ```

   Asegúrese de que la cantidad de conexiones que enumera la CMU coincida con la cantidad de HSM del clúster. Si no es así, cierre la sesión y comience de nuevo.

4. Utilice createUser para crear un usuario CO cuyo nombre sea **`example_officer`** y la contraseña sea **`password1`**.

   ```
   aws-cloudhsm>createUser CO example_officer password1
   ```

   La CMU le solicita información sobre la operación de creación de usuario.

   ```
   ************************CAUTION*********************************
   This is a CRITICAL operation, should be done on all nodes in the
   cluster. AWS does NOT synchronize these changes automatically with the
   nodes on which this operation is not executed or failed, please
   ensure this operation is executed on all nodes in the cluster.
   ****************************************************************
   ```

```
Do you want to continue(y/n)?
```

5. Escriba **y**.

Cómo crear un nuevo usuario CU

1. Utilice la herramienta de configuración para actualizar la configuración de la CMU.

   Linux

   ```
   $ sudo /opt/cloudhsm/bin/configure --cmu <IP address>
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\ configure.exe --cmu <IP address>
   ```

2. Iniciar CMU.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM> .\cloudhsm_mgmt_util.exe C:\ProgramData\Amazon
   \CloudHSM\data\cloudhsm_mgmt_util.cfg
   ```

3. Inicie sesión en HSM como usuario CO.

   ```
   aws-cloudhsm>loginHSM CO admin co12345
   ```

   Asegúrese de que la cantidad de conexiones que enumera la CMU coincida con la cantidad de HSM del clúster. Si no es así, cierre la sesión y comience de nuevo.

4. Utilice createUser para crear un nombre de usuario de CU **example_user** con una contraseña de **password1**.

   ```
   aws-cloudhsm>createUser CU example_user password1
   ```

La CMU le solicita información sobre la operación de creación de usuario.

```
************************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
****************************************************************

Do you want to continue(y/n)?
```

5. Escriba **y**.

Para obtener más información sobre createUser, consulte createUser.

Cómo enumerar todos los usuarios de HSM del clúster

Utilice el comando listUsers para mostrar a todos los usuarios en el clúster. No es necesario iniciar sesión para ejecutar listUsers y todos los tipos de usuarios pueden enumerarlos.

Cómo enumerar todos los usuarios en el clúster

1. Utilice la herramienta de configuración para actualizar la configuración de la CMU.

   Linux

   ```
   $ sudo /opt/cloudhsm/bin/configure --cmu <IP address>
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\ configure.exe --cmu <IP address>
   ```

2. Iniciar CMU.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

Windows

```
C:\Program Files\Amazon\CloudHSM> .\cloudhsm_mgmt_util.exe C:\ProgramData\Amazon
\CloudHSM\data\cloudhsm_mgmt_util.cfg
```

3. Ejecute listUsers para listar todos los usuarios del clúster.

```
aws-cloudhsm>listUsers
```

En la CMU, se muestran todos los usuarios del clúster.

```
Users on server 0(10.0.2.9):
Number of users found:4

    User Id             User Type       User Name
 MofnPubKey    LoginFailureCnt         2FA
         1              AU              app_user                                NO
               0               NO
         2              CO              example_officer                         NO
               0               NO
         3              CU              example_user                            NO
               0               NO
Users on server 1(10.0.3.11):
Number of users found:4

    User Id             User Type       User Name
 MofnPubKey    LoginFailureCnt         2FA
         1              AU              app_user                                NO
               0               NO
         2              CO              example_officer                         NO
               0               NO
         3              CU              example_user                            NO
               0               NO
Users on server 2(10.0.1.12):
Number of users found:4

    User Id             User Type       User Name
 MofnPubKey    LoginFailureCnt         2FA
         1              AU              app_user                                NO
               0               NO
```

```
        2               CO              example_officer                 NO
              0               NO
        3               CU              example_user                    NO
              0               NO
```

Para obtener más información sobre listUsers, consulte listUsers..

Cómo cambiar las contraseñas de los usuarios de HSM

Utilice changePswd para cambiar una contraseña.

En los tipos de usuario y las contraseñas se distingue entre mayúsculas y minúsculas, pero no en los nombres de usuario.

Los CO, los usuarios de criptografía (CU) y los usuarios de dispositivos (AU) solo pueden cambiar su propia contraseña. Para cambiar la contraseña de otro usuario, debe iniciar sesión como CO. No puede cambiar la contraseña de un usuario que actualmente haya iniciado sesión.

Cómo cambiar su propia contraseña

1. Utilice la herramienta de configuración para actualizar la configuración de la CMU.

   Linux

   ```
   $ sudo /opt/cloudhsm/bin/configure --cmu <IP address>
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\ configure.exe --cmu <IP address>
   ```

2. Iniciar CMU.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM> .\cloudhsm_mgmt_util.exe C:\ProgramData\Amazon
   \CloudHSM\data\cloudhsm_mgmt_util.cfg
   ```

3. Iniciar sesión en HSM.

```
aws-cloudhsm>loginHSM CO admin co12345
```

Asegúrese de que la cantidad de conexiones que enumera la CMU coincida con la cantidad de HSM del clúster. Si no es así, cierre la sesión y comience de nuevo.

4. Utilice changePswd para cambiar su propia contraseña.

```
aws-cloudhsm>changePswd CO example_officer <new password>
```

CMU le preguntará acerca de la operación de cambio de contraseña.

```
**********CAUTION**********
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
**********

Do you want to continue(y/n)?
```

5. Escriba **y**.

CMU le preguntará acerca de la operación de cambio de contraseña.

```
Changing password for example_officer(CO) on 3 nodes
```

Cómo cambiar la contraseña de otro usuario

1. Utilice la herramienta de configuración para actualizar la configuración de la CMU.

Linux

```
$ sudo /opt/cloudhsm/bin/configure --cmu <IP address>
```

Windows

```
C:\Program Files\Amazon\CloudHSM\bin\ configure.exe --cmu <IP address>
```

2. Iniciar CMU.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM> .\cloudhsm_mgmt_util.exe C:\ProgramData\Amazon
   \CloudHSM\data\cloudhsm_mgmt_util.cfg
   ```

3. Inicie sesión en HSM como usuario CO.

   ```
   aws-cloudhsm>loginHSM CO admin co12345
   ```

   Asegúrese de que la cantidad de conexiones que enumera la CMU coincida con la cantidad de HSM del clúster. Si no es así, cierre la sesión y comience de nuevo.

4. Utilice changePswd para cambiar la contraseña de otro usuario.

   ```
   aws-cloudhsm>changePswd CU example_user <new password>
   ```

   CMU le preguntará acerca de la operación de cambio de contraseña.

   ```
   
   ************************CAUTION*******************************
   This is a CRITICAL operation, should be done on all nodes in the
   cluster. AWS does NOT synchronize these changes automatically with the
   nodes on which this operation is not executed or failed, please
   ensure this operation is executed on all nodes in the cluster.
   **************************************************************
   
   Do you want to continue(y/n)?
   ```

5. Escriba **y**.

   CMU le preguntará acerca de la operación de cambio de contraseña.

```
Changing password for example_user(CU) on 3 nodes
```

Para más información acerca de changePswd, consulte changePswd .

Cómo eliminar usuarios de HSM

Ejecute deleteUser para eliminar un usuario. Debe iniciar sesión como CO para eliminar otro usuario.

> **Tip**
>
> No puede eliminar a los usuarios de criptografía (CU) que poseen claves.

Cómo eliminar un usuario

1. Utilice la herramienta de configuración para actualizar la configuración de la CMU.

   Linux

   ```
   $ sudo /opt/cloudhsm/bin/configure --cmu <IP address>
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\ configure.exe --cmu <IP address>
   ```

2. Iniciar CMU.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM> .\cloudhsm_mgmt_util.exe C:\ProgramData\Amazon
   \CloudHSM\data\cloudhsm_mgmt_util.cfg
   ```

3. Inicie sesión en HSM como usuario CO.

```
aws-cloudhsm>loginHSM CO admin co12345
```

Asegúrese de que la cantidad de conexiones que enumera la CMU coincida con la cantidad de HSM del clúster. Si no es así, cierre la sesión y comience de nuevo.

4. Ejecute deleteUser para eliminar un usuario.

```
aws-cloudhsm>deleteUser CO example_officer
```

CMU elimina el usuario.

```
Deleting user example_officer(CO) on 3 nodes
deleteUser success on server 0(10.0.2.9)
deleteUser success on server 1(10.0.3.11)
deleteUser success on server 2(10.0.1.12)
```

Para obtener más información acerca de deleteUser, consulte deleteUser.

Uso de la utilidad de administración (CMU) de CloudHSM para administrar la autenticación de dos factores (2FA) para los de responsables de criptografía

Para aumentar la seguridad, puede configurar la autenticación de dos factores (2FA), y así contribuir con la protección del clúster. Solo puede habilitar la 2FA para los responsables de criptografía (CO).

> **Note**
>
> No puede habilitar la 2FA para los usuarios de criptografía (CU) ni para las aplicaciones. La autenticación de dos factores (2FA) es solo para usuarios con rol de CO.

Temas

- Descripción de la 2FA para los usuarios de HSM
- Trabajo con la 2FA para los usuarios de HSM

Descripción de la 2FA para los usuarios de HSM

Al iniciar sesión en un clúster con una cuenta de módulo de servicio de hardware (HSM) compatible con la 2FA, le proporciona su contraseña a cloudhsm_mgmt_util (CMU) (el primer factor, es decir, el que conoce), y la CMU le proporciona un token y le solicita que lo firme. Para proporcionar el segundo factor (el que ya tienes), firma el token con una clave privada de un par de claves que ya ha creado y asociado al usuario de HSM. Para acceder al clúster, debe proporcionar el token firmado a la CMU.

Autenticación del cuórum y 2FA

El clúster usa la misma clave para la autenticación de cuórum y de la 2FA. Esto significa que un usuario con la 2FA habilitada está registrado efectivamente en M-of-n-access-control (MoFN). Para utilizar correctamente la autenticación de cuórum y la autenticación de 2FA para el mismo usuario de HSM, tenga en cuenta los siguientes aspectos:

- Si actualmente utiliza la autenticación de cuórum para un usuario, debe usar el mismo par de claves que creó para el usuario de quorum para habilitar la 2FA para el usuario.
- Si agrega el requisito de 2FA a un usuario que no cuenta con 2FA y que tampoco cuenta con autenticación de cuórum, registre a ese usuario como un usuario de MoFN con autenticación de 2FA.
- Si elimina el requisito de 2FA o cambia la contraseña de un usuario con 2FA que también es usuario con autenticación de cuórum, también eliminará el registro del usuario de quorum como usuario de MoFN.
- Si elimina el requisito de 2FA o cambia la contraseña de un usuario con 2FA que también es un usuario con autenticación de cuórum, pero aún desea que ese usuario participe en la autenticación de cuórum, debe volver a registrar ese usuario como usuario de MoFN.

Para obtener más información acerca de la autenticación de cuórum, consulte Uso de CMU para gestionar la autenticación de cuórum.

Trabajo con la 2FA para los usuarios de HSM

En esta sección se describe cómo trabajar con la 2FA para los usuarios de HSM, incluida la creación de usuarios de HSM de 2FA, la rotación de las claves y el inicio de sesión en el HSM como usuarios habilitados para la 2FA. Para obtener más información sobre cómo trabajar con usuarios de HSM, consulte ???, ???, ???, ???, y ???.

Creación de usuarios con 2FA

Para que un usuario de HSM pueda usar la 2FA, utilice una clave que cumpla los siguientes requisitos.

Requisitos de pares de claves de 2FA

Puede crear un nuevo par de claves o utilizar una clave existente que cumpla los siguientes requisitos.

- Tipo de clave: asimétrica
- Uso de clave: firmar y verificar
- Especificaciones de clave: RSA_2048
- El algoritmo de firma incluye:
  - `sha256WithRSAEncryption`

> **Note**
>
> Si utiliza la autenticación de cuórum o planea utilizarla, consulte the section called “Autenticación del cuórum y 2FA”.

Utilice la CMU y el par de claves para crear un nuevo usuario con rol de CO y que tenga la 2FA habilitada.

Cómo crear usuarios con rol de CO con la 2FA habilitada

1. En una terminal, lleve a cabo uno de los siguientes pasos:

   a. Acceda a su HSM e inicie sesión en la utilidad de administración de CloudHSM:

   ```
   /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

   b. Inicie sesión como CO y use el siguiente comando para crear un nuevo usuario de MFA con 2FA:

   ```
   aws-cloudhsm>createUser CO MFA <CO USER NAME> -2fa /home/ec2-user/authdata
   ************************CAUTION*********************************This is a
    CRITICAL operation, should be done on all nodes in the
   ```

```
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
**************************************************************

Do you want to continue(y/n)?

yCreating User exampleuser3(CO) on 1 nodesAuthentication data written to: "/
home/ec2-user/authdata"Generate Base64-encoded signatures for SHA256 digests in
 the authentication datafile.
To generate the signatures, use the RSA private key, which is the second factor
 ofauthentication for this user. Paste the signatures and the corresponding
 public keyinto the authentication data file and provide
the file path below.Leave this field blank to use the path initially
 provided.Enter filename:
```

c. Deje la terminal anterior en este estado. No presione enter ni introduzca ningún nombre de archivo.

2. En otra terminal, realice los siguientes pasos:

a. Acceda a su HSM e inicie sesión en la utilidad de administración de CloudHSM:

```
/opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
```

b. Genere un par de claves público-privadas con los siguientes comandos:

```
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt
 rsa_keygen_bits:2048
```

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

c. Ejecute el siguiente comando para instalar una característica de consulta de json para extraer el resumen del archivo authdata:

```
sudo yum install jq
```

d. Para extraer el valor de resumen, busque primero los siguientes datos en el archivo authdata:

```
{
  "Version":"1.0",
```

```
  "PublicKey":"",
  "Data":[
    {
      "HsmId": <"HSM ID">,
      "Digest": <"DIGEST">,
      "Signature":""
    }
  ]
}
```

> **Note**
>
> El resumen obtenido está codificado en base64; sin embargo, para firmarlo, es necesario decodificar primero el archivo y, a continuación, firmarlo. El siguiente comando decodificará el resumen y almacenará el contenido decodificado en 'digest1.bin'
>
> ```
> cat authdata | jq '.Data[0].Digest' | cut -c2- | rev | cut -c2- | rev |
>  base64 -d > digest1.bin
> ```

e. Convierta el contenido de la clave pública, añada "\n" y elimine los espacios, tal como se muestra a continuación:

```
-----BEGIN PUBLIC KEY-----\n<PUBLIC KEY>\n-----END PUBLIC KEY-----
```

> **Important**
>
> El comando anterior muestra cómo se agrega «\n» inmediatamente después de BEGIN PUBLIC KEY-----, se eliminan los espacios entre «\n» y el primer carácter de la clave pública, se agrega «\n» antes de -----END PUBLIC KEY y se eliminan los espacios entre «\n» y la terminación de la clave pública.

Este es el formato PEM para la clave pública que se admite en el archivo authdata.

f. Pegue el contenido del formato PEM de clave pública en la sección de clave pública del archivo authdata.

```
vi authdata
```

```
{
  "Version":"1.0",
  "PublicKey":"-----BEGIN PUBLIC KEY-----\n<"PUBLIC KEY">\n-----END PUBLIC
 KEY-----",
  "Data":[
{
  "HsmId":<"HSM ID">,
  "Digest":<"DIGEST">,
  "Signature":””
}
]
}
```

g. Firme el archivo de token con el siguiente comando:

```
openssl pkeyutl -sign -in digest1.bin -inkey private_key.pem -pkeyopt
 digest:sha256 | base64

Output Expected:
<"THE SIGNATURE">
```

> **Note**
>
> Como se muestra en el comando anterior, utilice openssl pkeyutl en lugar de openssl dgst para firmar.

h. Agregue el resumen firmado en el archivo de datos de autenticación en el campo "Firma".

```
vi authdata
```

```
{
    "Version": "1.0",
    "PublicKey": "-----BEGIN PUBLIC KEY----- ... -----END PUBLIC KEY-----",
    "Data": [
        {
            "HsmId": <"HSM ID">,
            "Digest": <"DIGEST">,
```

```
            "Signature": "Kkdl ... rkrvJ6Q=="
        },
        {
            "HsmId": <"HSM ID">,
            "Digest": <"DIGEST">,
            "Signature": "K1hxy ... Q261Q=="
        }
    ]
}
```

3. Regrese a la primera terminal y pulse **`Enter`**:

```
Generate Base64-encoded signatures for SHA256 digests in the authentication
 datafile. To generate the signatures, use the RSA private key,
which is the second factor ofauthentication for this user. Paste the signatures and
 the corresponding public keyinto the authentication data file and provide the file
 path below. Leave this field blank to use the path initially provided.
Enter filename: >>>>> Press Enter here

createUser success on server 0(10.0.1.11)
```

Administración de la 2FA para los usuarios de HSM

Utilice el cambio de contraseña para cambiar la contraseña de un usuario con 2FA, para activar o desactivar la 2FA, o para rotar la clave de 2FA. Cada vez que habilite la 2FA, debe proporcionar una clave pública para los inicios de sesión de 2FA.

El cambio de contraseña realiza cualquiera de los siguientes escenarios:

- Cómo cambiar la contraseña para un usuario con 2FA
- Cómo cambiar la contraseña para un usuario sin 2FA
- Cómo agregar 2FA a un usuario sin 2FA
- Cómo quitar la 2FA de un usuario con 2FA
- Cómo rotar la clave para un usuario con 2FA

También puede combinar tareas. Por ejemplo, puede eliminar la 2FA de un usuario y cambiar la contraseña al mismo tiempo, o puede rotar la clave de 2FA y cambiar la contraseña del usuario.

Cómo cambiar las contraseñas o rotar las claves para los usuarios con rol CO y con la 2FA habilitada

1. Utilice la CMU para iniciar sesión en el HSM como CO con la 2FA habilitada.
2. Utilice changePswd para cambiar la contraseña o rotar la clave de los usuarios con rol de CO y con la 2FA habilitada. Utilice el `-2fa` parámetro e incluya una ubicación en el sistema de archivos para que el sistema escriba el archivo `authdata`. Este archivo incluye un resumen de cada HSM del clúster.

   ```
   aws-cloudhsm>changePswd CO example-user <new-password> -2fa /path/to/authdata
   ```

   La CMU le solicita que utilice la clave privada para firmar los resúmenes del archivo `authdata` y que envíe las firmas con la clave pública.
3. Utilice la clave privada para firmar los resúmenes del archivo `authdata`, agregue las firmas y la clave pública al archivo `authdata` con formato JSON y, a continuación, proporcione a CMU la ubicación del archivo `authdata`. Para obtener más información, consulte the section called “Referencia de la configuración”.

   > **Note**
   >
   > El clúster usa la misma clave para la autenticación de cuórum y la 2FA. Si utiliza la autenticación de cuórum o planea utilizarla, consulte the section called “Autenticación del cuórum y 2FA”.

Cómo deshabilitar la 2FA de los usuarios con rol de CO y con la 2FA habilitada

1. Utilice la CMU para iniciar sesión en el HSM como CO con la 2FA habilitada.
2. Utilice changePswd para quitar la 2FA de los usuarios con rol de CO y que tengan la 2FA habilitada.

   ```
   aws-cloudhsm>changePswd CO example-user <new password>
   ```

   La CMU le solicita que confirme la operación de cambio de contraseña.

> **Note**
>
> Si elimina el requisito de 2FA o cambia la contraseña de un usuario con 2FA que también es usuario con autenticación de cuórum, también eliminará el registro del usuario de cuórum como usuario de MoFN. Para obtener más información acerca de los usuarios con 2FA, consulte the section called “Autenticación del cuórum y 2FA”.

3. Escriba **y**.

   La CMU confirma la operación de cambio de contraseña.

Referencia de la configuración

A continuación, se muestra un ejemplo de las propiedades de 2FA del archivo `authdata`, tanto para la solicitud generada por la CMU como para sus respuestas.

```
{
    "Version": "1.0",
    "PublicKey": "-----BEGIN PUBLIC KEY----- ... -----END PUBLIC KEY-----",
    "Data": [
        {
            "HsmId": "hsm-lgavqitns2a",
            "Digest": "k5O1p3f6foQRVQH7S8Rrjcau6h3TYqsSdr16A54+qG8=",
            "Signature": "Kkdl ... rkrvJ6Q=="
        },
        {
            "HsmId": "hsm-lgavqitns2a",
            "Digest": "IyBcx4I5Vyx1jztwvXinCBQd9lDx8oQe7iRrWjBAi1w=",
            "Signature": "K1hxy ... Q261Q=="
        }
    ]
}
```

Data

Nodo de nivel superior. Contiene un nodo subordinado para cada HSM del clúster. Aparece en las solicitudes y respuestas de todos los comandos de 2FA.

Resumir

Esto es lo que debe firmar para proporcionar el segundo factor de autenticación. La CMU se genera en las solicitudes de todos los comandos de 2FA.

HsmId

El ID de su HSM. Aparece en las solicitudes y respuestas de todos los comandos de 2FA.

PublicKey

La porción de clave pública del par de claves que generó se insertó como una cadena con formato PEM. Debe introducirla en las respuestas para createUser y changePswd.

Signature

El resumen firmado y codificado de Base 64. Debe introducirlo en las respuestas de todos los comandos de 2FA.

Versión

La versión del archivo de datos de autenticación con formato JSON Aparece en las solicitudes y respuestas de todos los comandos de 2FA.

Usar la Utilidad de administración de CloudHSM (CMU) para administrar la autenticación de cuórum (control de acceso M de N)

Los HSM de su AWS CloudHSM clúster admiten la autenticación de quórum, que también se conoce como control de acceso M of N. Con la autenticación de cuórum ningún usuario único del HSM puede realizar operaciones controladas mediante cuórum en el HSM. En su lugar, para llevar a cabo estas operaciones debe cooperar un número mínimo de usuarios del HSM (al menos 2). Con la autenticación de cuórum, puede añadir una capa adicional de protección al exigir la aprobación de más de un usuario del HSM.

La autenticación de cuórum puede controlar las siguientes operaciones:

- Administración de usuarios del HSM realizada mediante responsables de criptografía (CO): creación y eliminación de usuarios de HSM y cambio de la contraseña de otro usuario del HSM. Para obtener más información, consulte Uso de la autenticación de cuórum para responsables de criptografía.

Los siguientes temas contienen más información acerca de la autenticación de cuórum en AWS CloudHSM.

Temas

- Información general sobre la autenticación de cuórum
- Detalles adicionales sobre la autenticación de cuórum
- Uso de la autenticación de cuórum para responsables de criptografía: primera configuración
- Uso de la autenticación de cuórum para responsables de criptografía
- Cambio del valor mínimo de cuórum para responsables de criptografía

Información general sobre la autenticación de cuórum

En los pasos siguientes se resumen los procesos de autenticación de cuórum. Para informarse de las herramientas y los específicos, consulte Uso de la autenticación de cuórum para responsables de criptografía.

1. Cada usuario del HSM crea una clave asimétrica para la firma. Lo hacen fuera del HSM, teniendo cuidado de proteger la clave adecuadamente.
2. Cada usuario del HSM se conecta al HSM y registra la parte pública de su clave de firma (la clave pública) en el HSM.
3. Cuando un usuario del HSM desea realizar una operación controlada por cuórum, cada usuario inicia sesión en el HSM y obtiene un token de cuórum.
4. El usuario del HSM pasa el token de cuórum a uno o varios usuarios de ese HSM y les pide su aprobación.
5. Los otros usuarios del HSM dan su aprobación utilizando sus claves para firmar criptográficamente el token de cuórum. Esto se produce fuera del HSM.
6. Cuando el usuario del HSM tiene el número necesario de aprobaciones, el mismo usuario inicia sesión en el HSM y entrega el token de quórum y las aprobaciones (firmas) al HSM.
7. El HSM utiliza las claves públicas registradas de cada firmante para verificar las firmas. Si las firmas son válidas, el HSM aprueba el token.
8. Ahora el usuario del HSM ya puede realizar la operación controlada mediante cuórum.

Detalles adicionales sobre la autenticación de cuórum

Tenga en cuenta la siguiente información adicional acerca del uso de la autenticación de cuórum en AWS CloudHSM.

- Un usuario de HSM puede firmar su propio token de quórum, es decir, el usuario solicitante puede proporcionar una de las aprobaciones necesarias para la autenticación de quórum.
- Elija el número mínimo de aprobadores de cuórum para las operaciones controladas mediante cuórum. El número más pequeño que puede elegir es dos (2) y el número más grande que puede elegir es ocho (8).
- El HSM puede almacenar hasta 1024 tokens de cuórum. Si el HSM ya tiene 1024 tokens cuando intenta crear uno nuevo, el HSM eliminará uno de los tokens que haya caducado. De forma predeterminada, los tokens caducan diez minutos después de su creación.
- El clúster usa la misma clave para la autenticación de cuórum y para la autenticación de dos factores (2FA). Para obtener más información sobre el uso de la autenticación de cuórum y la 2FA, consulte Autenticación de cuórum y 2FA.

Uso de la autenticación de cuórum para responsables de criptografía: primera configuración

Los siguientes temas describen los pasos que debe seguir para configurar el módulo de seguridad de hardware (HSM) de forma que los responsables de criptografía (CO) puedan usar la autenticación de cuórum. Debe seguir estos pasos solo una vez cuando configure por primera vez la autenticación de cuórum para CO. Después de completar estos pasos, consulte Uso de la autenticación de cuórum para responsables de criptografía.

Temas

- Requisitos previos
- Creación y registro de una clave de firma
- Cómo establecer el valor mínimo de cuórum en el HSM

Requisitos previos

Para comprender este ejemplo, debe estar familiarizado con la cloudhsm_mgmt_util herramienta de línea de comandos (CMU). En este ejemplo, el AWS CloudHSM clúster tiene dos HSM, cada uno con el mismo CoS, como se muestra en el siguiente resultado del comando. listUsers Para obtener más información sobre la creación de usuarios, consulte Administración de usuarios de HSM.

```
aws-cloudhsm>listUsers
Users on server 0(10.0.2.14):
Number of users found:7

    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
          1             PRECO           admin                                NO
            0              NO
          2             AU              app_user                             NO
            0              NO
          3             CO              officer1                             NO
            0              NO
          4             CO              officer2                             NO
            0              NO
          5             CO              officer3                             NO
            0              NO
          6             CO              officer4                             NO
            0              NO
          7             CO              officer5                             NO
            0              NO
Users on server 1(10.0.1.4):
Number of users found:7

    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
          1             PRECO           admin                                NO
            0              NO
          2             AU              app_user                             NO
            0              NO
          3             CO              officer1                             NO
            0              NO
          4             CO              officer2                             NO
            0              NO
          5             CO              officer3                             NO
            0              NO
          6             CO              officer4                             NO
            0              NO
          7             CO              officer5                             NO
            0              NO
```

Creación y registro de una clave de firma

Para utilizar la autenticación de quórum, cada OC debe realizar todos los pasos siguientes:

Temas

- Creación de un par de claves RSA
- Creación y firma de un token de registro
- Cómo registrar una clave pública con el HSM

Creación de un par de claves RSA

Hay muchas formas de crear y proteger un par de claves. El siguiente ejemplo muestra cómo hacerlo con OpenSSL.

Example : creación de una clave privada con OpenSSL

El siguiente ejemplo ilustra cómo utilizar OpenSSL para crear una clave RSA de 2048 bits que está protegida por una frase de contraseña. Para utilizar este ejemplo, sustituya *officer1.key* por el nombre del archivo donde desea almacenar la clave.

```
$ openssl genrsa -out officer1.key -aes256 2048
Generating RSA private key, 2048 bit long modulus
.....................................+++
.+++
e is 65537 (0x10001)
Enter pass phrase for officer1.key:
Verifying - Enter pass phrase for officer1.key:
```

A continuación, genere la clave pública usando la clave privada que acaba de crear.

Example : creación de una clave pública con OpenSSL

El siguiente ejemplo muestra cómo usar OpenSSL para crear una clave pública a partir de la clave privada que acaba de crear.

```
$ openssl rsa -in officer1.key -outform PEM -pubout -out officer1.pub
Enter pass phrase for officer1.key:
writing RSA key
```

Creación y firma de un token de registro

Cree un token y fírmelo con la clave privada que acaba de generar en el paso anterior.

Example : cree un token.

El token de registro es solo un archivo con cualquier dato aleatorio que no supere el tamaño máximo de 245 bytes. Firme el token con la clave privada para demostrar que tiene acceso a la clave privada. El siguiente comando usa eco para redirigir una cadena a un archivo.

```
$ echo "token to be signed" > officer1.token
```

Firme el token y guárdelo en un archivo de firma. Necesitará el token firmado, el token sin firmar y la clave pública para registrar al CO como usuario de MoFN en el HSM.

Example : firme el token.

Utilice OpenSSL y la clave privada para firmar el token de registro y crear el archivo de firma.

```
$ openssl dgst -sha256 \
    -sign officer1.key \
    -out officer1.token.sig officer1.token
```

Cómo registrar una clave pública con el HSM

Después de crear una clave, el CO debe registrar la parte pública de la clave (la clave pública) en el HSM.

Para registrar una clave pública en el HSM

1. Utilice el siguiente comando para iniciar la herramienta de línea de comandos cloudhsm_mgmt_util.

   ```
   $ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

2. Utilice el comando loginHSM para iniciar sesión en los HSM como CO. Para obtener más información, consulte ???.
3. Utilice el comando registerQuorumPubKey para registrar la clave pública. Para obtener más información, consulte el siguiente ejemplo o utilice el comando help registerQuorumPubKey.

Example : registro de una clave pública con el HSM

En el siguiente ejemplo, se explica cómo se utiliza el comando registerQuorumPubKey en la herramienta de línea de comandos cloudhsm_mgmt_util para registrar una clave pública del CO con

el HSM. Para utilizar este comando, el CO tiene que haber iniciado sesión en el HSM. Reemplace estos valores por sus propios valores:

```
aws-cloudhsm> registerQuorumPubKey CO <officer1> <officer1.token> <officer1.token.sig>
 <officer1.pub>

************************CAUTION*******************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
**************************************************************

Do you want to continue(y/n)?y
registerQuorumPubKey success on server 0(10.0.2.14)
```

<officer1.token>

La ruta a un archivo que contiene un token de registro sin firmar. Puede contener cualquier dato aleatorio con un tamaño máximo de archivo de 245 bytes.

Obligatorio: sí

<officer1.token.sig>

La ruta a un archivo que contiene el hash firmado por el mecanismo SHA256_PKCS del token de registro.

Obligatorio: sí

<officer1.pub>

La ruta al archivo que contiene la clave pública de un par de claves asimétricas RSA-2048. Utilice la clave privada para firmar el token de registro.

Obligatorio: sí

Una vez que todos los CO registran las claves públicas, el resultado del comando listUsers lo muestra en la columna `MofnPubKey`, tal y como puede verse en el siguiente ejemplo.

```
aws-cloudhsm>listUsers
Users on server 0(10.0.2.14):
Number of users found:7
```

```
    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
         1              PRECO           admin                             NO
           0               NO
         2              AU              app_user                          NO
           0               NO
         3              CO              officer1                        YES
           0               NO
         4              CO              officer2                        YES
           0               NO
         5              CO              officer3                        YES
           0               NO
         6              CO              officer4                        YES
           0               NO
         7              CO              officer5                        YES
           0               NO
Users on server 1(10.0.1.4):
Number of users found:7

    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
         1              PRECO           admin                             NO
           0               NO
         2              AU              app_user                          NO
           0               NO
         3              CO              officer1                        YES
           0               NO
         4              CO              officer2                        YES
           0               NO
         5              CO              officer3                        YES
           0               NO
         6              CO              officer4                        YES
           0               NO
         7              CO              officer5                        YES
           0               NO
```

Cómo establecer el valor mínimo de cuórum en el HSM

Para utilizar autenticación de cuórum para CO, un CO debe iniciar sesión en el HSM y, a continuación, establecer el valor mínimo de cuórum, también conocido como "valor m". Este es el número mínimo de aprobaciones del CO necesarias para realizar las operaciones de administración de usuarios de HSM. Cualquier CO en el HSM puede establecer el valor mínimo de cuórum,

incluidos los CO que no han registrado una clave para firmar. Puede cambiar el valor mínimo del cuórum en cualquier momento; para obtener más información, consulte Cambio del valor mínimo.

Para establecer el valor mínimo de cuórum en el HSM

1. Utilice el siguiente comando para iniciar la herramienta de línea de comandos cloudhsm_mgmt_util.

```
$ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
```

2. Utilice el comando loginHSM para iniciar sesión en los HSM como CO. Para obtener más información, consulte ???.
3. Utilice el comando setMValue para establecer el valor mínimo de cuórum. Para obtener más información, consulte el siguiente ejemplo o utilice el comando help setMValue.

Example : establecimiento del valor mínimo de cuórum del HSM

Este ejemplo utiliza un valor mínimo de cuórum de dos. Puede elegir cualquier valor entre dos (2) y ocho (8), hasta el número total de CO del HSM. En este ejemplo, el HSM tiene seis CO, por lo que el valor máximo posible es seis.

Para utilizar el siguiente comando de ejemplo, reemplace el número final (*2*) por el valor mínimo de cuórum preferido.

```
aws-cloudhsm>setMValue 3 2
*************************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
*****************************************************************

Do you want to continue(y/n)?y
Setting M Value(2) for 3 on 2 nodes
```

En el ejemplo anterior, el primer número (3) identifica el servicio de HSM cuyo valor mínimo de cuórum está estableciendo.

En la siguiente tabla se enumeran los identificadores del servicio de HSM junto con sus nombres, descripciones y los comandos que se incluyen en el servicio.

Identificador del servicio	Nombre del servicio	Descripción del servicio	Comandos de HSM
3	USER_MGMT	Gestión de usuarios HSM	• createUser • deleteUser • changePswd (solo se aplica al cambiar la contraseña de otro usuario de HSM)
4	MISC_CO	Servicio de CO misceláneo	• setMValue

Para obtener el valor mínimo de cuórum de un servicio, utilice el comando getMValue, tal y como se muestra en el siguiente ejemplo.

```
aws-cloudhsm>getMValue 3
MValue of service 3[USER_MGMT] on server 0 : [2]
MValue of service 3[USER_MGMT] on server 1 : [2]
```

El resultado del comando getMValue anterior muestra que el valor mínimo de cuórum para las operaciones de administración de usuarios de HSM (servicio 3) ahora es dos.

Después de completar estos pasos, consulte Uso de la autenticación de cuórum para responsables de criptografía.

Uso de la autenticación de cuórum para responsables de criptografía

Un responsable de criptografía (CO) en el HSM puede configurar la autenticación de cuórum para las siguientes operaciones en el HSM:

- Creación de usuarios de HSM
- Eliminación de usuarios de HSM
- Cambio de la contraseña de otro usuario de HSM

Después de la configuración de HSM para la autenticación de cuórum, los CO no puede realizar operaciones de administración de usuarios de HSM por su cuenta. El siguiente ejemplo muestra el resultado cuando un CO intenta crear un usuario nuevo en el HSM. Se produce un error en el comando, `RET_MXN_AUTH_FAILED`, lo que indica un error en la autenticación de cuórum.

```
aws-cloudhsm>createUser CU user1 password
*************************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
*****************************************************************

Do you want to continue(y/n)?y
Creating User user1(CU) on 2 nodes
createUser failed: RET_MXN_AUTH_FAILED
creating user on server 0(10.0.2.14) failed

Retry/Ignore/Abort?(R/I/A):A
```

Para realizar una operación de administración de usuarios de HSM, un CO debe completar las siguientes tareas:

1. Obtenga un token de cuórum.
2. Obtenga aprobaciones (firmas) de otros CO.
3. Apruebe el token en el HSM.
4. Realice la operación de administración de usuarios de HSM.

Si aún no ha configurado el HSM para la autenticación de cuórum para CO, hágalo ahora. Para obtener más información, consulte Configuración por primera vez.

Cómo obtener un token de cuórum

En primer lugar, el CO debe utilizar la herramienta de línea de comandos cloudhsm_mgmt_util para solicitar un token de quórum.

Para obtener un token de cuórum

1. Utilice el siguiente comando para iniciar la herramienta de línea de comandos cloudhsm_mgmt_util.

```
$ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
```

2. Utilice el comando loginHSM para iniciar sesión en los HSM como CO. Para obtener más información, consulte ???.
3. Utilice el comando getToken para obtener un token de cuórum. Para obtener más información, consulte el siguiente ejemplo o utilice el comando help getToken.

Example : obtener un token de cuórum

Este ejemplo obtiene un token de cuórum para el CO con nombre de usuario officer1 y guarda el token en un archivo denominado `officer1.token`. Para utilizar el comando de ejemplo, sustituya estos valores por los suyos:

- *`officer1`*: el nombre del CO que está obteniendo el token. Debe ser el mismo CO que ha iniciado sesión en el HSM y está ejecutando este comando.
- *`officer1.token`*: el nombre del archivo que se debe utilizar para almacenar el token de cuórum.

En el siguiente comando, 3 identifica el servicio para el que puede utilizar el token que está obteniendo. En este caso, el token es para las operaciones de administración de usuarios de HSM (servicio 3). Para obtener más información, consulte Cómo establecer el valor mínimo de cuórum en el HSM.

```
aws-cloudhsm>getToken 3 officer1 officer1.token
getToken success on server 0(10.0.2.14)
Token:
Id:1
Service:3
Node:1
Key Handle:0
User:officer1
getToken success on server 1(10.0.1.4)
Token:
Id:1
Service:3
Node:0
Key Handle:0
User:officer1
```

Cómo obtener firmas de CO responsables de la aprobación

Un CO que tiene un token de cuórum debe obtener la aprobación del token de otros CO. Para dar su aprobación, los otros CO utilizan su clave de firma para firmar criptográficamente el token. Lo hacen fuera del HSM.

Existen muchas maneras diferentes de firmar el token. El siguiente ejemplo muestra cómo hacerlo con OpenSSL. Para utilizar otra herramienta de firma, asegúrese de que la herramienta utiliza la clave privada del CO (clave de firma) para firmar un resumen SHA-256 del token.

Example : obtener firmas de los CO responsables de la aprobación.

En este ejemplo, el CO que tiene el token (officer1) necesita al menos dos aprobaciones. Los siguientes comandos de ejemplo muestran cómo pueden dos CO utilizar OpenSSL para firmar el token criptográficamente.

En el primer comando, officer1 firma su propio token. Para utilizar los siguientes comandos de ejemplo, sustituya estos valores por los suyos:

- *officer1.key* y *officer2.key*: el nombre del archivo que contiene la clave de firma del CO.
- *officer1.token.sig1* y *officer1.token.sig2*: el nombre del archivo que se debe utilizar para almacenar la firma. Asegúrese de guardar cada firma en un archivo diferente.
- *officer1.token*: el nombre del archivo que contiene el token que el CO está firmando.

```
$ openssl dgst -sha256 -sign officer1.key -out officer1.token.sig1 officer1.token
Enter pass phrase for officer1.key:
```

En el siguiente comando, officer2 firma el mismo token.

```
$ openssl dgst -sha256 -sign officer2.key -out officer1.token.sig2 officer1.token
Enter pass phrase for officer2.key:
```

Aprobación del token firmado en el HSM

Una vez que un CO obtiene el número mínimo de aprobaciones (firmas) de otros CO, debe aprobar el token firmado en el HSM.

Para aprobar el token firmado en el HSM

1. Cree un archivo de aprobación del token. Para obtener más información, consulte el siguiente ejemplo.
2. Utilice el siguiente comando para iniciar la herramienta de línea de comandos cloudhsm_mgmt_util.

```
$ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
```

3. Utilice el comando loginHSM para iniciar sesión en los HSM como CO. Para obtener más información, consulte ???.
4. Utilice el comando approveToken para aprobar el token firmado, transmitiendo el archivo de aprobación del token. Para obtener más información, consulte el siguiente ejemplo.

Example : creación de un archivo de aprobación del token y aprobar el token firmado en el HSM

El archivo de aprobación del token es un archivo de texto en un formato en particular que el HSM necesita. El archivo contiene información sobre el token, los aprobadores y las firmas de los aprobadores. Se muestra a continuación un archivo de aprobación del token de ejemplo.

```
# For "Multi Token File Path", type the path to the file that contains
# the token. You can type the same value for "Token File Path", but
# that's not required. The "Token File Path" line is required in any
# case, regardless of whether you type a value.
Multi Token File Path = officer1.token;
Token File Path = ;

# Total number of approvals
Number of Approvals = 2;

# Approver 1
# Type the approver's type, name, and the path to the file that
# contains the approver's signature.
Approver Type = 2; # 2 for CO, 1 for CU
Approver Name = officer1;
Approval File = officer1.token.sig1;

# Approver 2
# Type the approver's type, name, and the path to the file that
# contains the approver's signature.
Approver Type = 2; # 2 for CO, 1 for CU
```

```
Approver Name = officer2;
Approval File = officer1.token.sig2;
```

Después de crear el archivo de aprobación del token, el CO utiliza la herramienta de línea de comandos cloudhsm_mgmt_util para iniciar sesión en el HSM. El CO utiliza después el comando approveToken para aprobar el token, tal y como se muestra en el siguiente ejemplo. Sustituya *approval.txt* por el nombre del archivo de aprobación del token.

```
aws-cloudhsm>approveToken approval.txt
approveToken success on server 0(10.0.2.14)
approveToken success on server 1(10.0.1.4)
```

Cuando este comando se ejecuta correctamente, el HSM ha aprobado el token de cuórum. Para comprobar el estado de un token, utilice el comando listTokens, tal y como se muestra en el siguiente ejemplo. El resultado del comando muestra que el token tiene el número necesario de aprobaciones.

El tiempo de validez del token indica durante cuánto tiempo se garantiza que el token persista en el HSM. Incluso después de que transcurra el tiempo de validez del token (cero segundos), puede seguir usando el token.

```
aws-cloudhsm>listTokens

====================
    Server 0(10.0.2.14)
====================
-------- Token - 0 ----------
Token:
Id:1
Service:3
Node:1
Key Handle:0
User:officer1
Token Validity: 506 sec
Required num of approvers : 2
Current num of approvals : 2
Approver-0: officer1
Approver-1: officer2
Num of tokens = 1

====================
    Server 1(10.0.1.4)
```

```
====================
-------- Token - 0 ----------
Token:
Id:1
Service:3
Node:0
Key Handle:0
User:officer1
Token Validity: 506 sec
Required num of approvers : 2
Current num of approvals : 2
Approver-0: officer1
Approver-1: officer2
Num of tokens = 1

listTokens success
```

Utilización del token para operaciones de administración de usuarios

Una vez que un CO tiene un token con el número necesario de aprobaciones, tal y como se muestra en la sección anterior, el CO puede realizar una de las siguientes operaciones de administración de usuarios de HSM:

- Crear un usuario de HSM con el comando createUser
- Eliminar un usuario de HSM con el comando deleteUser
- Cambiar la contraseña de un usuario de HSM diferente con el comando changePswd

Para obtener más información acerca del uso de estos comandos, consulte Administración de usuarios de HSM.

El CO puede utilizar el token para una sola operación. Cuando dicha operación se realiza correctamente, el token ya no es válido. Para hacer otra operación de administración de usuarios de HSM, el CO tiene que obtener un token de cuórum nuevo, obtener firmas nuevas de los aprobadores y aprobar el token nuevo en el HSM.

> **Note**
>
> El token M de N solo será válido mientras se mantenga la sesión iniciada actual. Si cierra sesión en la cloudhsm_mgmt_util o si la red se desconecta, el token perderá su validez. Del

mismo modo, un token autorizado solo se puede usar en cloudhsm_mgmt_util y no se puede usar para autenticarse en otra aplicación.

En el siguiente comando de ejemplo, el CO crea un usuario nuevo en el HSM.

```
aws-cloudhsm>createUser CU user1 password
**************************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
******************************************************************

Do you want to continue(y/n)?y
Creating User user1(CU) on 2 nodes
```

Una vez que el comando anterior se ejecuta correctamente, otro comando listUsers posterior muestra al usuario nuevo.

```
aws-cloudhsm>listUsers
Users on server 0(10.0.2.14):
Number of users found:8

    User Id             User Type       User Name                      MofnPubKey
  LoginFailureCnt         2FA
        1               PCO             admin                           NO
          0               NO
        2               AU              app_user                        NO
          0               NO
        3               CO              officer1                       YES
          0               NO
        4               CO              officer2                       YES
          0               NO
        5               CO              officer3                       YES
          0               NO
        6               CO              officer4                       YES
          0               NO
        7               CO              officer5                       YES
          0               NO
        8               CU              user1                           NO
          0               NO
```

```
Users on server 1(10.0.1.4):
Number of users found:8

    User Id             User Type       User Name                           MofnPubKey
  LoginFailureCnt         2FA
        1               PCO             admin                                NO
          0               NO
        2               AU              app_user                             NO
          0               NO
        3               CO              officer1                            YES
          0               NO
        4               CO              officer2                            YES
          0               NO
        5               CO              officer3                            YES
          0               NO
        6               CO              officer4                            YES
          0               NO
        7               CO              officer5                            YES
          0               NO
        8               CU              user1                                NO
          0               NO
```

Si el CO intenta realizar otra operación de administración de usuarios de HSM, da error con un error de autenticación de cuórum, tal y como se muestra en el siguiente ejemplo.

```
aws-cloudhsm>deleteUser CU user1
Deleting user user1(CU) on 2 nodes
deleteUser failed: RET_MXN_AUTH_FAILED
deleteUser failed on server 0(10.0.2.14)

Retry/rollBack/Ignore?(R/B/I):I
deleteUser failed: RET_MXN_AUTH_FAILED
deleteUser failed on server 1(10.0.1.4)

Retry/rollBack/Ignore?(R/B/I):I
```

El comando listTokens muestra que el CO no tiene ningún token aprobado, tal y como puede verse en el siguiente ejemplo. Para realizar otra operación de administración de usuarios de HSM, el CO tiene que obtener un token de cuórum nuevo, obtener firmas nuevas de los aprobadores y aprobar el token nuevo en el HSM.

```
aws-cloudhsm>listTokens
```

```
====================
    Server 0(10.0.2.14)
====================
Num of tokens = 0

====================
    Server 1(10.0.1.4)
====================
Num of tokens = 0

listTokens success
```

Cambio del valor mínimo de cuórum para responsables de criptografía

Después de establecer el valor mínimo de cuórum para que los responsables de criptografía (CO) puedan usar autenticación de cuórum, sería aconsejable que cambiara el valor mínimo de cuórum. El HSM le permite cambiar el valor mínimo de cuórum solo cuando el número de aprobadores es igual o superior al actual valor mínimo de cuórum. Por ejemplo, si el valor mínimo de cuórum es dos, al menos dos CO deben dar su aprobación para cambiar el valor mínimo de cuórum.

Para obtener aprobación de cuórum para cambiar el valor mínimo de cuórum, necesita un token de cuórum para el comando setMValue (servicio 4). Si necesita obtener un token de cuórum para el comando setMValue (servicio 4), el valor mínimo de cuórum del servicio 4 debe ser superior a uno. Esto significa que para poder cambiar el valor mínimo de cuórum para CO (servicio 3), es posible que tenga que cambiar el valor mínimo de cuórum para el servicio 4.

En la siguiente tabla se enumeran los identificadores del servicio de HSM junto con sus nombres, descripciones y los comandos que se incluyen en el servicio.

Identificador del servicio	Nombre del servicio	Descripción del servicio	Comandos de HSM
3	USER_MGMT	Gestión de usuarios HSM	• createUser • deleteUser • changePswd (solo se aplica al cambiar la contraseña de

Identificador del servicio	Nombre del servicio	Descripción del servicio	Comandos de HSM
			otro usuario de HSM)
4	`MISC_CO`	Servicio de CO misceláneo	• setMValue

Para cambiar el valor mínimo de cuórum para responsables de criptografía

1. Utilice el siguiente comando para iniciar la herramienta de línea de comandos cloudhsm_mgmt_util.

   ```
   $ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

2. Utilice el comando loginHSM para iniciar sesión en los HSM como CO. Para obtener más información, consulte ???.
3. Utilice el comando getMValue para obtener el valor mínimo de cuórum del servicio 3. Para obtener más información, consulte el siguiente ejemplo.
4. Utilice el comando getMValue para obtener el valor mínimo de cuórum del servicio 4. Para obtener más información, consulte el siguiente ejemplo.
5. Si el valor mínimo de cuórum del servicio 4 es inferior al valor del servicio 3, utilice el comando setMValue para cambiar el valor del servicio 4. Cambie el valor para el servicio 4 a uno que sea igual o superior al valor para el servicio 3. Para obtener más información, consulte el siguiente ejemplo.
6. Obtenga un token de cuórum, teniendo cuidado de especificar el servicio 4 como el servicio para el que puede utilizar el token.
7. Obtenga aprobaciones (firmas) de otros CO.
8. Apruebe el token en el HSM.
9. Utilice el comando setMValue para cambiar el valor mínimo de cuórum del servicio 3 (operaciones de administración de usuarios realizadas por CO).

Example : obtención de valores mínimos de cuórum y cambiar el valor del servicio 4

El siguiente comando de ejemplo muestra que el valor mínimo de cuórum para el servicio 3 es actualmente dos.

```
aws-cloudhsm>getMValue 3
MValue of service 3[USER_MGMT] on server 0 : [2]
MValue of service 3[USER_MGMT] on server 1 : [2]
```

El siguiente comando de ejemplo muestra que el valor mínimo de cuórum para el servicio 4 es actualmente uno.

```
aws-cloudhsm>getMValue 4
MValue of service 4[MISC_CO] on server 0 : [1]
MValue of service 4[MISC_CO] on server 1 : [1]
```

Para cambiar el valor mínimo de cuórum del servicio 4, utilice el comando setMValue y establezca un valor que sea igual o superior al valor del servicio 3. El siguiente ejemplo establece el valor mínimo de cuórum para el servicio 4 en dos (2), el mismo valor establecido para el servicio 3.

```
aws-cloudhsm>setMValue 4 2
************************CAUTION*******************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
**************************************************************

Do you want to continue(y/n)?y
Setting M Value(2) for 4 on 2 nodes
```

El siguiente comando muestra que el valor mínimo de cuórum es ahora dos para el servicio 3 y para el servicio 4.

```
aws-cloudhsm>getMValue 3
MValue of service 3[USER_MGMT] on server 0 : [2]
MValue of service 3[USER_MGMT] on server 1 : [2]
```

```
aws-cloudhsm>getMValue 4
MValue of service 4[MISC_CO] on server 0 : [2]
```

```
MValue of service 4[MISC_CO] on server 1 : [2]
```

Administrar claves en AWS CloudHSM

En AWS CloudHSM, utilice cualquiera de las siguientes opciones para administrar las claves de los HSM del clúster:

- Biblioteca PKCS #11
- Proveedor de JCE
- Proveedores de KSP y CNG
- La CLI de CloudHSM

Para administrar las claves, inicie sesión en el HSM con el nombre de usuario y la contraseña de un usuario de criptografía (CU). Solo una CU puede crear una clave. La CU que crea una clave es la propietaria y administradora.

Temas

- Ajustes clave de sincronización y durabilidad en AWS CloudHSM
- Empaquetado de llaves AES AWS CloudHSM
- Uso de claves de confianza en AWS CloudHSM
- Administración de claves con la CLI de CloudHSM
- Gestión de claves con la KMU y la CMU

Ajustes clave de sincronización y durabilidad en AWS CloudHSM

En este tema se describen los ajustes de sincronización de claves AWS CloudHSM, los problemas habituales a los que se enfrentan los clientes al trabajar con las claves de un clúster y las estrategias para aumentar la durabilidad de las claves.

Temas

- Conceptos
- Más información sobre la sincronización de claves
- Trabajo con las configuraciones de durabilidad de las claves del cliente
- Sincronización de claves entre clústeres clonados

Conceptos

Claves de token

Claves persistentes que se crean durante las operaciones de generación, importación o desempaquetado de claves. AWS CloudHSM sincroniza las claves simbólicas en un clúster.

Clave de sesión

Claves efímeras que solo existen en un módulo de seguridad de hardware (HSM) del clúster. AWS CloudHSM no sincroniza las claves de sesión en un clúster.

Sincronización de claves del lado del cliente

Proceso del lado del cliente que clona las claves simbólicas que se crean durante las operaciones de generación, importación o desencapsulamiento de claves. Para hacer que las claves de token sean más duraderas, ejecute un clúster con un mínimo de dos HSM.

Sincronización de claves del lado del servidor

Clona periódicamente las claves de todos los HSM del clúster. No requiere administración.

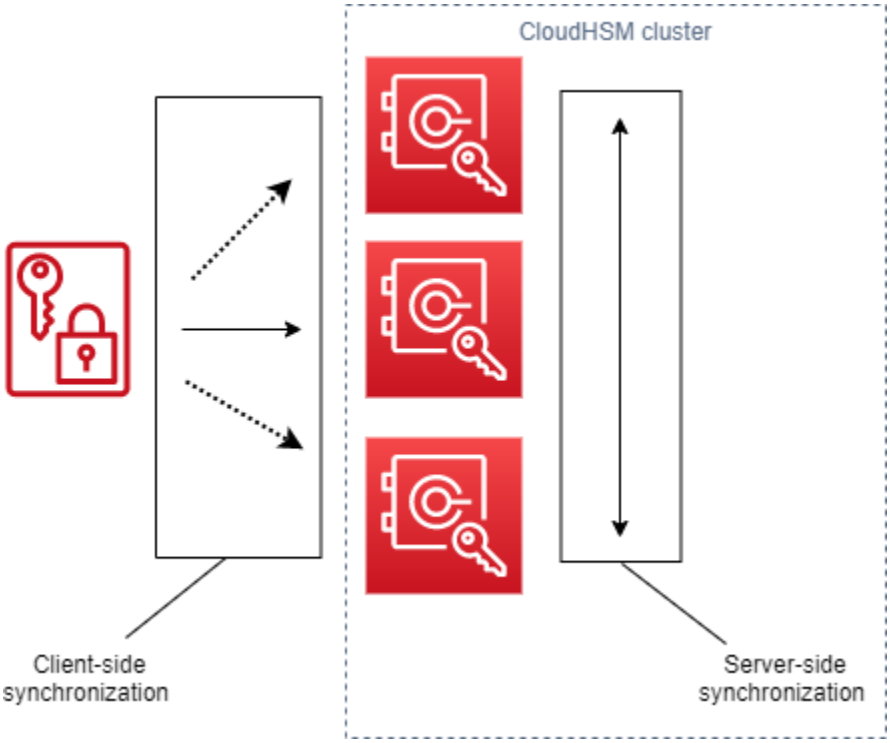
Configuración de durabilidad de claves del cliente

Los ajustes que usted configura en el cliente y que afectan a la durabilidad de la clave. Estos ajustes funcionan de forma diferente en SDK 5 de cliente y SDK 3 de cliente.

- En SDK 5 de cliente, utilice esta configuración para ejecutar un único clúster de HSM.
- En SDK 3 de cliente, utilice esta configuración para especificar el número de HSM necesarios para que las operaciones de creación de claves se realicen correctamente.

Más información sobre la sincronización de claves

AWS CloudHSM utiliza la sincronización de claves para clonar claves simbólicas en todos los HSM de un clúster. Las claves simbólicas se crean como claves persistentes durante las operaciones de generación, importación o desencapsulamiento de claves. Para distribuir estas claves en el clúster, CloudHSM ofrece sincronización de claves del lado del cliente y del servidor.



El objetivo de la sincronización de claves, tanto del lado del servidor como del lado del cliente, es distribuir las nuevas claves en el clúster lo más rápido posible después de crearlas. Esto es importante porque las llamadas posteriores que realice para usar nuevas claves se pueden enrutar a cualquier HSM disponible en el clúster. Si la llamada que realiza se dirige a un HSM sin la clave, la llamada fallará. Puede mitigar este tipo de errores especificando que sus aplicaciones reintenten las llamadas posteriores realizadas después de las operaciones de creación de claves. El tiempo necesario para la sincronización puede variar en función de la carga de trabajo del clúster y de otros elementos intangibles. Utilice CloudWatch métricas para determinar el tiempo que debe emplear su aplicación en este tipo de situaciones. Para obtener más información, consulte CloudWatch Métricas.

El desafío de la sincronización de claves en un entorno de nube es la durabilidad de las claves. Las claves se crean en un único HSM y, a menudo, se empiezan a utilizar esas claves de forma inmediata. Si el HSM en el que se crean las claves falla antes de clonarlas en otro HSM del clúster, se pierden las claves y se pierde el acceso a todo lo cifrado por ellas. Para mitigar este riesgo, ofrecemos sincronización del lado del cliente. La sincronización del lado del cliente es un proceso que clona las claves simbólicas que se crean durante las operaciones de generación, importación o desencapsulamiento de claves. Al clonar las claves a medida que se crean, las hace más duraderas. Por supuesto, no se pueden clonar las claves de un clúster con un solo HSM. Para que las claves sean más duraderas, también le recomendamos que configure el clúster para que utilice un mínimo de dos HSM. Con la sincronización del lado del cliente y un clúster con dos HSM, puede superar el desafío de la durabilidad de las claves en un entorno de nube.

Trabajo con las configuraciones de durabilidad de las claves del cliente

La sincronización de claves es, en su mayoría, un proceso automático, pero puede gestionar la configuración de durabilidad de las claves del lado del cliente. La configuración de durabilidad de las claves del lado del cliente funciona de forma diferente en SDK 5 de cliente y SDK 3 de cliente.

- En SDK 5 de cliente, presentamos el concepto de cuórums de disponibilidad de claves, que requiere que se ejecuten clústeres con un mínimo de dos HSM. Puede utilizar la configuración de durabilidad clave del lado del cliente para excluirse del requisito de los dos HSM. Para obtener más información sobre los cuórums, consulte the section called “Conceptos de SDK 5 de cliente”.
- En SDK 3 de cliente, se utiliza la configuración de durabilidad de las claves del lado del cliente para especificar el número de HSM en los que la creación de claves debe realizarse correctamente para que la operación general se considere un éxito.

Configuración de durabilidad de las claves de cliente de SDK 5 de cliente

En SDK 5 de cliente, la sincronización de claves es un proceso totalmente automático. Con el cuórum de disponibilidad de claves, las claves recién creadas deben existir en dos HSM del clúster para que la aplicación pueda usar la clave. Para utilizar el cuórum de disponibilidad de claves, el clúster debe tener un mínimo de dos HSM.

Si la configuración del clúster no cumple los requisitos de durabilidad clave, cualquier intento de crear o utilizar una clave simbólica fallará y aparecerá el siguiente mensaje de error en los registros:

```
Key <key handle> does not meet the availability requirements - The key must be
 available on at least 2 HSMs before being used.
```

Puede usar los ajustes de configuración del cliente para excluirse del cuórum de disponibilidad de claves. Por ejemplo, puede optar por no ejecutar un clúster con un solo HSM.

Conceptos de SDK 5 de cliente

Cuórum de disponibilidad de claves

AWS CloudHSM especifica el número de HSM de un clúster en los que deben existir las claves para que la aplicación pueda utilizarlas. Requiere clústeres con un mínimo de dos HSM.

Gestión de la configuración de durabilidad de la clave del cliente

Para administrar la configuración de durabilidad de las claves del cliente, debe utilizar la herramienta de configuración de SDK 5 de cliente.

PKCS #11 library

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Linux

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 --disable-key-availability-check
```

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Windows

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-pkcs11.exe" --disable-key-
availability-check
```

OpenSSL Dynamic Engine

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Linux

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
$ sudo /opt/cloudhsm/bin/configure-dyn --disable-key-availability-check
```

JCE provider

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Linux

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
$ sudo /opt/cloudhsm/bin/configure-jce --disable-key-availability-check
```

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Windows

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-jce.exe" --disable-key-
availability-check
```

CloudHSM CLI

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Linux

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
$ sudo /opt/cloudhsm/bin/configure-cli --disable-key-availability-check
```

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Windows

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-cli.exe" --disable-key-
availability-check
```

Configuración de durabilidad de las claves de cliente de SDK 3 de cliente

En SDK 3 de cliente, la sincronización de claves es en su mayoría un proceso automático, pero puede usar la configuración de durabilidad de las claves del cliente para aumentar la durabilidad de las claves. Debe especificar el número de HSM en los que la creación de claves debe realizarse correctamente para que la operación general se considere un éxito. La sincronización del lado del cliente siempre hace todo lo posible por clonar las claves de todos los HSM del clúster, independientemente de la configuración que elija. La configuración impone la creación de claves en el número de HSM que especifique. Si especifica un valor y el sistema no puede replicar la clave en ese número de HSM, el sistema eliminará automáticamente el material clave no deseado y podrá volver a intentarlo.

> **Important**
>
> Si no establece la configuración de durabilidad de las claves del cliente (o si usa el valor predeterminado de 1), sus claves son vulnerables a la pérdida. Si su HSM actual falla antes de que el servicio del servidor haya clonado esa clave en otro HSM, perderá el material de la clave.

Para maximizar la durabilidad de la clave, considere la posibilidad de especificar al menos dos HSM para la sincronización del lado del cliente. Recuerde que, independientemente del número de HSM que especifique, la carga de trabajo del clúster sigue siendo la misma. La sincronización del lado del cliente siempre hace todo lo posible por clonar las claves de todos los HSM del clúster.

Recomendaciones

- Mínimo: dos HSM por clúster
- Máximo: uno menos que el número total de HSM del clúster

Si se produce un error en la sincronización del lado del cliente, el servicio del cliente elimina las claves no deseadas que puedan haberse creado y que ahora no son necesarias. Esta limpieza es la respuesta más sencilla y puede que no siempre funcione. Si la limpieza no se realiza correctamente,

es posible que tenga que eliminar el material de clave no deseado. Para obtener más información, consulte Errores de sincronización de claves.

Configuración del archivo de configuración para garantizar la durabilidad de la clave del cliente

Para especificar la configuración de durabilidad de la clave del cliente, debe editar `cloudhsm_client.cfg`.

Cómo editar el archivo de configuración del cliente

1. Abra `cloudhsm_client.cfg`.

   Linux:

   ```
   /opt/cloudhsm/etc/cloudhsm_client.cfg
   ```

   Windows:

   ```
   C:\ProgramData\Amazon\CloudHSM\data\cloudhsm_client.cfg
   ```

2. En el `client` nodo del archivo, añada `create_object_minimum_nodes` y especifique un valor para el número mínimo de HSM en los que AWS CloudHSM deben crearse correctamente las claves para que las operaciones de creación de claves se realicen correctamente.

   ```
   "create_object_minimum_nodes" : 2
   ```

   > **Note**
   >
   > La herramienta de línea de comandos key_mgmt_util (KMU) tiene una configuración adicional para la durabilidad de las claves de cliente. Para obtener más información, consulte the section called “KMU y sincronización del lado del cliente”.

Referencia de la configuración

Estas son las propiedades de sincronización del lado del cliente, que se muestran en un extracto del `cloudhsm_client.cfg`:

```
{
```

```
    "client": {
        "create_object_minimum_nodes" : 2,
        ...
    },

    ...
}
```

create_object_minimum_nodes

Especifica el número mínimo de HSM necesarios para considerar que las operaciones de generación, importación o desencapsulamiento de claves se han realizado correctamente. Si está establecido, el valor predeterminado es 1. Esto significa que, para cada operación de creación de claves, el servicio del cliente intenta crear claves en todos los HSM del clúster, pero para que se realice correctamente, solo necesita crear una clave única en un HSM del clúster.

KMU y sincronización del lado del cliente

Si crea claves con la herramienta de línea de comandos key_mgmt_util (KMU), utilizará un parámetro de línea de comandos opcional (*-min_srv*) para limitar el número de HSM en los que clonar las claves. Si especifica el parámetro de la línea de comandos y un valor en el archivo de configuración, AWS CloudHSM respeta el mayor de los dos valores.

Para obtener más información, consulte los temas siguientes:

- GenDSA KeyPair
- GeneCC KeyPair
- Género A KeyPair
- genSymKey
- importPrivateKey
- importPubKey
- imSymKey
- insertMaskedObject
- unWrapKey

Sincronización de claves entre clústeres clonados

La sincronización del lado del cliente y del lado del servidor solo sirve para sincronizar claves dentro del mismo clúster. Si clona una copia de seguridad de un clúster en otra región, puede usar el comando syncKey de cloudhsm_mgmt_util (CMU) para sincronizar las claves entre los clústeres. Puede utilizar clústeres clonados para obtener redundancia entre regiones o para simplificar el proceso de recuperación de desastres. Para obtener más información, consulte syncKey.

Empaquetado de llaves AES AWS CloudHSM

En este tema se describen las opciones de empaquetado de claves AES AWS CloudHSM. El encapsulamiento de claves AES utiliza una clave AES (la clave de encapsulamiento) para encapsular otra clave de cualquier tipo (la clave de destino). Utilice el encapsulamiento de claves para proteger las claves almacenadas o transmitir claves a través de redes poco seguras.

Temas

- Algoritmos admitidos
- Mediante el empaquetado de claves AES AWS CloudHSM

Algoritmos admitidos

AWS CloudHSM ofrece tres opciones para empaquetar claves AES, cada una de ellas basada en cómo se rellena la clave de destino antes de empaquetarla. El relleno se realiza automáticamente, en función del algoritmo que se utilice, cuando se invoca el encapsulamiento de claves. En la siguiente tabla, se muestran los algoritmos admitidos y detalles relacionados para ayudarle a elegir un mecanismo de encapsulamiento adecuado para su aplicación.

Algoritmo de encapsulamiento de claves AES	Especificación	Tipos de claves de destino admitidos	Esquema de relleno	AWS CloudHSM Disponibilidad del cliente
Encapsulamiento de claves AES con relleno de ceros	RFC 5649 y SP 800 - 38F	Todos	Agrega ceros después de los bits de la clave, si es necesario, para alinear los bloques	SDK 3.1 y versiones posteriores

Algoritmo de encapsulamiento de claves AES	Especificación	Tipos de claves de destino admitidos	Esquema de relleno	AWS CloudHSM Disponibilidad del cliente
Encapsulamiento de claves AES sin relleno	RFC 3394 y SP 800 - 38F	Claves alineadas en bloques, como AES y 3DES	Ninguna	SDK 3.1 y versiones posteriores
Encapsulamiento de claves AES con relleno PKCS #5	Ninguna	Todos	Se agregan al menos 8 bytes según el esquema de relleno PKCS #5 para alinear los bloques	Todos

Para obtener más información acerca de cómo puede utilizar en su aplicación los algoritmos de encapsulamiento de claves AES de la tabla anterior, consulte Uso del encapsulamiento de claves AES en AWS CloudHSM.

Descripción de los vectores de inicialización del encapsulamiento de claves AES

Antes de realizar el encapsulamiento, CloudHSM agrega un vector de inicialización (IV) a la clave de destino para garantizar la integridad de los datos. Cada algoritmo de encapsulamiento de claves tiene una restricciones específicas sobre el tipo de IV permitido. Para instalar la sonda intravenosa AWS CloudHSM, tiene dos opciones:

- IV implícito: el IV se establece en NULL y CloudHSM utiliza el valor predeterminado de ese algoritmo en las operaciones de encapsulamiento y desencapsulamiento (recomendado)
- IV explicito: el IV se establece pasando su valor predeterminado a la función de encapsulamiento de claves.

> **Important**
>
> Debe saber qué IV está utilizando en su aplicación. Para desencapsular la clave, debe proporcionar el mismo IV que utilizó para encapsularla. Si usa un IV implícito para

encapsularla, use un IV implícito para desencapsularla. Con un IV implícito, CloudHSM usará el valor predeterminado para realizar el desencapsulamiento.

En la tabla siguiente, se describen los valores permitidos para los IV, lo que determina el algoritmo de encapsulamiento.

Algoritmo de encapsulamiento de claves AES	IV implícito	IV explícito
Encapsulamiento de claves AES con relleno de ceros	Obligatoria Valor predeterminado: (IV calculado internamente en función de la especificación)	No permitido
Encapsulamiento de claves AES sin relleno	Permitido (recomendado) Valor predeterminado: `0xA6A6A6A6A6A6A6A6`	Permitida Solo se acepta este valor: `0xA6A6A6A6A6A6A6A6`
Encapsulamiento de claves AES con relleno PKCS #5	Permitido (recomendado) Valor predeterminado: `0xA6A6A6A6A6A6A6A6`	Permitida Solo se acepta este valor: `0xA6A6A6A6A6A6A6A6`

Mediante el empaquetado de claves AES AWS CloudHSM

Las claves se encapsular y desencapsulan del modo siguiente:

- En biblioteca PCKS #11, seleccione el mecanismo apropiado para las funciones `C_WrapKey` y `C_UnWrapKey`, tal como se muestra en la siguiente tabla.
- En proveedor de JCE, seleccione el algoritmo, el modo y la combinación de relleno apropiados, e implemente los métodos de cifrado `Cipher.WRAP_MODE` y `Cipher.UNWRAP_MODE`, tal como se muestra en la siguiente tabla.
- En la CLI de CloudHSM, elija el algoritmo adecuado de la lista envoltorio para llaves de algoritmos y algoritmos desempaquetar llaves compatibles, tal y como se muestra en la siguiente tabla.

- En key_mgmt_util (KMU), utilice los comandos wrapKey y unWrapKey con los valores m apropiados, tal y como se muestra en la siguiente tabla.

Algoritmo de encapsulamiento de claves AES	Mecanismo de PKCS #11	Método de Java	Subcomando CLI	Argumento de la utilidad de administración de claves (KMU)
Encapsulamiento de claves AES con relleno de ceros	• `CKM_CLOUDHSM_AES_KEY_WRAP_ZERO_PAD` (mecanismo definido por el proveedor)	`AESWrap/ECB/ZeroPadding`	aes-zero-pad	m = 6
Encapsulamiento de claves AES sin relleno	• `CKM_CLOUDHSM_AES_KEY_WRAP_NO_PAD` (mecanismo definido por el proveedor)	`AESWrap/ECB/NoPadding`	aes-no-pad	m = 5
Encapsulamiento de claves AES con relleno PKCS #5	• `CKM_CLOUDHSM_AES_KEY_WRAP_PKCS5_PAD` (mecanismo definido por el proveedor)	`AESWrap/ECB/PKCS5Padding`	aes-pkcs5-pad	m = 4

Uso de claves de confianza en AWS CloudHSM

AWS CloudHSM admite el empaquetado de claves confiable para proteger las claves de datos de las amenazas internas. En este tema se describe cómo crear claves de confianza para proteger los datos.

Temas

- Más información sobre las claves de confianza
- Atributos de clave de confianza
- Cómo usar claves de confianza para encapsular claves de datos
- ¿Cómo desencapsular una clave de datos con una clave de confianza?

Más información sobre las claves de confianza

Una clave de confianza es una clave que se utiliza para encapsular otras claves y que los administradores y los responsables de criptografía (CO) identifican específicamente como de confianza mediante el atributo `CKA_TRUSTED`. Además, los administradores y los oficiales de criptografía (CO) utilizan `CKA_UNWRAP_TEMPLATE` y los atributos relacionados para especificar qué acciones pueden realizar las claves de datos una vez que están encapsuladas en una clave de confianza. Las claves de datos separadas por la clave de confianza también deben contener estos atributos para que la operación de desencapsulamiento se realice correctamente, lo que ayuda a garantizar que las claves de datos no encapsuladas solo estén permitidas para el uso que se pretenda.

Utilice el atributo `CKA_WRAP_WITH_TRUSTED` para identificar todas las claves de datos que desee encapsular con claves de confianza. De este modo, podrá restringir las claves de datos para que las aplicaciones solo puedan utilizar claves de confianza para desencapsularlas. Una vez establecido este atributo en las claves de datos, el atributo pasa a ser de solo lectura y no se puede modificar. Una vez establecidos estos atributos, las aplicaciones solo pueden desencapsular las claves de datos con las claves en las que confíen, y al desencapsularlas siempre se obtienen claves de datos con atributos que limitan el uso de dichas claves.

Atributos de clave de confianza

Los siguientes atributos permiten marcar una clave como de confianza, especificar que una clave de datos solo se pueda encapsular y desencapsular con una clave de confianza y controlar lo que puede hacer una clave de datos una vez desencapsulada:

- `CKA_TRUSTED`: aplique este atributo (además de `CKA_UNWRAP_TEMPLATE`) a la clave que encapsulará las claves de datos para especificar que un administrador o un oficial de cifrado (CO) ha realizado las diligencias necesarias y confía en esta clave. Solo un administrador o un CO pueden configurar `CKA_TRUSTED`. El usuario de criptografía (CU) es el propietario de la clave, pero solo un CO puede establecer su atributo `CKA_TRUSTED`.
- `CKA_WRAP_WITH_TRUSTED`: aplique este atributo a una clave de datos exportable para especificar que solo puede encapsular esta clave con claves marcadas como `CKA_TRUSTED`. Una vez establecido `CKA_WRAP_WITH_TRUSTED` como true, el atributo pasa a ser de solo lectura y no se puede cambiar ni eliminar.
- `CKA_UNWRAP_TEMPLATE`: aplique este atributo a la clave de encapsulamiento (además de `CKA_TRUSTED`) para especificar los nombres y valores de los atributos que el servicio debe aplicar automáticamente a las claves de datos que desencapsula el servicio. Cuando una aplicación envía una clave para desencapsular, la aplicación también puede proporcionar su propia plantilla de desencapsulamiento. Si especifica una plantilla de desencapsulamiento y la aplicación proporciona su propia plantilla de desencapsulamiento, el HSM utiliza ambas plantillas para aplicar los nombres y valores de los atributos a la clave. Sin embargo, si un valor en la `CKA_UNWRAP_TEMPLATE` para la clave de encapsulamiento entra en conflicto con un atributo proporcionado por la aplicación durante la solicitud de desencapsulamiento, se produce un error en la solicitud de desencapsulamiento.

Para más información sobre los atributos, consulte los siguientes temas:

- Atributos de clave de PKCS #11
- Atributos clave del JCE
- Atributos de clave de la CLI de CloudHSM

Cómo usar claves de confianza para encapsular claves de datos

Para usar una clave de confianza para encapsular una clave de datos, debe completar tres pasos básicos:

1. Para la clave de datos que planea encapsular con una clave de confianza, defina su atributo `CKA_WRAP_WITH_TRUSTED` como verdadero.
2. Para la clave de datos que planea encapsular con una clave de confianza, defina su atributo `CKA_TRUSTED` como true.

3. Use la clave de confianza para encapsular la clave de datos.

Paso 1: establezca la clave de datos **CKA_WRAP_WITH_TRUSTED** como true

Para la clave de datos que desea encapsular, elija una de las siguientes opciones para establecer el atributo CKA_WRAP_WITH_TRUSTED de la clave como true. Esto restringe la clave de los datos para que las aplicaciones solo puedan utilizar claves de confianza para encapsularlos.

Opción 1: si se genera una clave nueva, establezca **CKA_WRAP_WITH_TRUSTED** como true.

Genere una clave mediante PKCS #11, JCE o CloudHSM CLI. Consulte los siguientes ejemplos para obtener más información.

PKCS #11

Para generar una clave con PKCS #11, debe establecer el atributo CKA_WRAP_WITH_TRUSTED de clave como true. Como se muestra en el siguiente ejemplo, hágalo incluyendo este atributo en el CK_ATTRIBUTE template de la clave y, a continuación, establezca el atributo como true:

```
CK_BYTE_PTR label = "test_key";
CK_ATTRIBUTE template[] = {
        {CKA_WRAP_WITH_TRUSTED, &true_val,         sizeof(CK_BBOOL)},
        {CKA_LABEL,             label,             strlen(label)},
        ...
};
```

Para obtener más información, consulte nuestros ejemplos públicos que muestran la generación de claves con el PKCS #11.

JCE

Para generar una clave con JCE, debe establecer el atributo WRAP_WITH_TRUSTED de la clave en true. Como se muestra en el siguiente ejemplo, hágalo incluyendo este atributo en el KeyAttributesMap de la clave y, a continuación, establezca el atributo como true:

```
final String label = "test_key";
final KeyAttributesMap keySpec = new KeyAttributesMap();
keySpec.put(KeyAttribute.WRAP_WITH_TRUSTED, true);
keySpec.put(KeyAttribute.LABEL, label);
...
```

Para obtener más información, consulte nuestros ejemplos públicos que muestran la generación de claves con JCE.

CloudHSM CLI

Para generar una clave con la CLI de CloudHSM, debe establecer el atributo `wrap-with-trusted` de clave como true. Para ello, incluya `wrap-with-trusted=true` en el argumento apropiado para el comando de generación de claves:

- En el caso de claves simétricas, añada `wrap-with-trusted` al argumento `attributes`.
- En el caso de las claves públicas, añada `wrap-with-trusted` al argumento `public-attributes`.
- Para claves privadas, añada `wrap-with-trusted` al argumento `private-attributes`.

Para más información sobre la generación de pares de claves, consulte clave generate-asymmetric-pair .

Para obtener más información sobre la generación de claves simétricas, consulte key generate-symmetric .

Opción 2: si utiliza una clave existente, utilice la CLI de CloudHSM para establecer su **`CKA_WRAP_WITH_TRUSTED`** como true.

Para establecer el atributo `CKA_WRAP_WITH_TRUSTED` de una clave existente como verdadero, siga estos pasos:

1. Utilice el comando login para iniciar sesión como usuario de criptografía (CU).
2. Utilice el comando key set-attribute para establecer el atributo `wrap-with-trusted` de la clave como verdadero.

```
aws-cloudhsm > key set-attribute --filter attr.label=test_key --name wrap-with-
trusted --value true
{
  "error_code": 0,
  "data": {
    "message": "Attribute set successfully"
  }
}
```

Paso 2: establecer la clave de confianza **CKA_TRUSTED** como true

Para convertir una clave en una clave de confianza, su atributo CKA_TRUSTED debe estar establecido como true. Para ello, puede utilizar la CLI de CloudHSM o la utilidad de administración de CloudHSM (CMU).

- Si utiliza la CLI de CloudHSM para establecer el atributo de CKA_TRUSTED de una clave, consulte Cómo marcar una clave como de confianza con la CLI de CloudHSM.
- Si utiliza la CMU para establecer el atributo de una clave CKA_TRUSTED, consulte Cómo marcar una clave como de confianza con la CMU.

Paso 3. Uso de la clave de confianza para encapsular la clave de datos

Para encapsular la clave de datos a la que se hace referencia en el paso 1 con la clave de confianza que configuró en el paso 2, consulte los siguientes enlaces para ver ejemplos de código. En cada uno de ellos se muestra cómo encapsular las claves.

- AWS CloudHSM Ejemplos de PKCS #11
- AWS CloudHSM Ejemplos de JCE

¿Cómo desencapsular una clave de datos con una clave de confianza?

Para desencapsular una clave de datos, necesita una clave de confianza que tenga configurado CKA_UNWRAP como true. Para ser una clave de este tipo, también debe cumplir los siguientes criterios:

- El atributo CKA_TRUSTED de la clave se debe establecer como true.
- La clave debe utilizar CKA_UNWRAP_TEMPLATE y los atributos relacionados para especificar qué acciones pueden realizar las claves de datos una vez desencapsuladas. Si, por ejemplo, desea que una clave desencapsulada no se pueda exportar, debe configurar CKA_EXPORTABLE = FALSE como parte de CKA_UNWRAP_TEMPLATE.

> **Note**
>
> CKA_UNWRAP_TEMPLATE solo está disponible con PKCS #11.

Cuando una aplicación envía una clave para desencapsularla, también puede proporcionar su propia plantilla de encapsulamiento. Si especifica una plantilla de desencapsulamiento y la aplicación proporciona su propia plantilla de desencapsulamiento, el HSM utiliza ambas plantillas para aplicar los nombres y valores de los atributos a la clave. Sin embargo, si durante la solicitud de desencapsulamiento un valor de `CKA_UNWRAP_TEMPLATE` de la clave de confianza entra en conflicto con un atributo proporcionado por la aplicación, la solicitud de desencapsulamiento fallará.

Para ver un ejemplo de cómo desencapsular una clave de datos con una clave de confianza, consulte este ejemplo de PKCS #11.

Administración de claves con la CLI de CloudHSM

Si utilizas la última serie de versiones del SDK, usa la CLI de CloudHSM para administrar las claves AWS CloudHSM del clúster. Para obtener más información, consulte los temas que aparecen a continuación.

- El uso de claves de confianza describe cómo utilizar la CLI de CloudHSM para crear claves de confianza para proteger los datos.
- En Generar claves se proporcionan instrucciones sobre cómo generar claves, incluidas claves simétricas, claves RSA y claves EC.
- En Eliminar claves se proporciona información sobre cómo los propietarios de las claves eliminan las claves.
- En Compartir y dejar de compartir claves se detalla cómo los propietarios de las claves comparten y dejan de compartir las claves.
- En Filtrar claves se ofrecen pautas sobre cómo usar los filtros para buscar claves.

Uso de la CLI de CloudHSM para generar claves

Antes de poder generar una clave, debe iniciar la CLI de CloudHSM e iniciar sesión como usuario de criptografía (CU). Para generar claves en el HSM, utilice el comando que corresponde al tipo de clave que desea generar.

Temas

- Generación de claves simétricas
- Generación de claves asimétricas
- Temas relacionados de

Generación de claves simétricas

Use los comandos enumerados en key generate-symmetric para generar claves simétricas. Para ver todas las opciones disponibles, utilice el comando help key generate-symmetric.

Generación de una clave AES

Ejecute el comando key generate-symmetric aes para generar claves AES. Para ver todas las opciones disponibles, utilice el comando help key generate-symmetric aes.

Example

El siguiente ejemplo genera una clave AES de 32 bytes.

```
aws-cloudhsm > key generate-symmetric aes \
    --label aes-example \
    --key-length-bytes 32
```

Argumentos

<LABEL>

Especifica la etiqueta de clave AES definida por el usuario.

Obligatorio: sí

<KEY-LENGTH-BYTES>

Especifica la longitud de la clave en bytes.

Valores válidos:

- 16, 24 y 32

Obligatorio: sí

<KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios que se debe configurar para la clave AES generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`).

Para obtener una lista de los atributos AWS CloudHSM clave compatibles, consulteAtributos de clave de la CLI de CloudHSM.

Obligatorio: no

<SESSION>

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión. Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use key set-attribute.

Las claves se generan como claves persistentes/token de forma predeterminada. El uso del parámetro <SESSION> modifica esta opción, creando una clave de sesión o efímera

Obligatorio: no

Generación de una clave secreta genérica

Ejecute el comando key generate-symmetric generic-secret para generar claves secretas genéricas. Para ver todas las opciones disponibles, utilice el comando help key generate-symmetric generic-secret.

Example

El siguiente ejemplo genera una clave secreta genérica de 32 bytes.

```
aws-cloudhsm > key generate-symmetric generic-secret \
    --label generic-secret-example \
    --key-length-bytes 32
```

Argumentos

<LABEL>

Especifica una etiqueta definida por el usuario para la clave secreta genérica.

Obligatorio: sí

<KEY-LENGTH-BYTES>

Especifica la longitud de la clave en bytes.

Valores válidos:

- 1 a 800

Obligatorio: sí

<KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios que se deben establecer para la clave secreta genérica creada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`)

Para obtener una lista de los atributos AWS CloudHSM clave compatibles, consulteAtributos de clave de la CLI de CloudHSM.

Obligatorio: no

<SESSION>

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión. Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use key set-attribute.

Las claves se generan como claves persistentes/token de forma predeterminada. El uso del parámetro <SESSION> modifica esta opción, creando una clave de sesión o efímera

Obligatorio: no

Generación de claves asimétricas

Use los comandos enumerados en clave generate-asymmetric-pair para generar pares de claves asimétricas.

Generación de claves RSA

Utilice el comando key generate-asymmetric-pair rsa para generar un par de claves RSA. Para ver todas las opciones disponibles, utilice el comando help key generate-asymmetric-pair rsa.

Example

El siguiente ejemplo genera un par de claves RSA de 2048 bits.

```
aws-cloudhsm > key generate-asymmetric-pair rsa \
```

```
    --public-exponent 65537 \
    --modulus-size-bits 2048 \
    --public-label rsa-public-example \
    --private-label rsa-private-example
```

Argumentos

<PUBLIC_LABEL>

Especifica la etiqueta de clave pública definida por el usuario.

Obligatorio: sí

<PRIVATE_LABEL>

Especifica la etiqueta de clave privada definida por el usuario.

Obligatorio: sí

<MODULUS_SIZE_BITS>

Especifica la longitud del módulo en bits. El valor mínimo es 2048.

Obligatorio: sí

<PUBLIC_EXPONENT>

Especifica el exponente público. El valor debe ser un número impar superior o igual a 65537.

Obligatorio: sí

<PUBLIC_KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios que se deben establecer para la clave pública RSA generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`).

Para obtener una lista de los atributos AWS CloudHSM clave compatibles, consulteAtributos de clave de la CLI de CloudHSM.

Obligatorio: no

<SESSION>

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión. Utilice este parámetro cuando necesite una clave solo brevemente, por

ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use key set-attribute.

Las claves se generan como claves persistentes/token de forma predeterminada. El uso del parámetro <SESSION> modifica esta opción, creando una clave de sesión o efímera

Obligatorio: no

Generación de pares de claves EC (criptografía de curva elíptica)

Utilice el comando key generate-asymmetric-pair ec para generar un par de claves EC. Para ver todas las opciones disponibles, incluida una lista de las curvas elípticas admitidas, use el comando help key generate-asymmetric-pair ec.

Example

El siguiente ejemplo genera un par de claves EC usando la curva elíptica Secp384r1.

```
aws-cloudhsm > key generate-asymmetric-pair ec \
    --curve secp384r1 \
    --public-label ec-public-example \
    --private-label ec-private-example
```

Argumentos

<PUBLIC_LABEL>

Especifica la etiqueta de clave pública definida por el usuario. El tamaño máximo permitido `label` es de 127 caracteres para el SDK de cliente 5.11 y versiones posteriores. El SDK de cliente 5.10 y versiones anteriores tiene un límite de 126 caracteres.

Obligatorio: sí

<PRIVATE_LABEL>

Especifica la etiqueta de clave privada definida por el usuario. El tamaño máximo permitido `label` es de 127 caracteres para el SDK de cliente 5.11 y versiones posteriores. El SDK de cliente 5.10 y versiones anteriores tiene un límite de 126 caracteres.

Obligatorio: sí

<CURVE>

Especifica el identificador de la curva elíptica.

Valores válidos:

- prime256v1
- secp256r1
- secp224r1
- secp384r1
- secp256k1
- secp521r1

Obligatorio: sí

<PUBLIC_KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios que se deben establecer para la clave pública EC generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`)

Para obtener una lista de los atributos AWS CloudHSM clave compatibles, consulteAtributos de clave de la CLI de CloudHSM.

Obligatorio: no

<PRIVATE_KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios que se deben establecer para la clave privada EC generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`)

Para obtener una lista de los atributos AWS CloudHSM clave compatibles, consulteAtributos de clave de la CLI de CloudHSM.

Obligatorio: no

<SESSION>

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión. Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra

clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use key set-attribute.

De forma predeterminada, las claves que se generan son claves persistentes (token). La transferencia a <SESSION> cambia este estado, lo que garantiza que la clave generada con este argumento sea una clave de sesión (efímera).

Obligatorio: no

Temas relacionados de

- Atributos de clave de la CLI de CloudHSM
- clave generate-asymmetric-pair
- key generate-symmetric

Uso de la CLI de CloudHSM para eliminar claves

Use el ejemplo del presente tema para eliminar una clave con la CLI de CloudHSM. Solo el propietario de la clave puede eliminar claves.

Temas

- Ejemplo: eliminar una clave
- Temas relacionados de

Ejemplo: eliminar una clave

1. Ejecute el comando key list para identificar la clave que desea eliminar:

```
aws-cloudhsm > key list --filter attr.label="my_key_to_delete" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000540011",
        "key-info": {
          "key-owners": [
```

```
            {
              "username": "my_crypto_user",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "rsa",
          "label": "my_key_to_delete",
          "id": "",
          "check-value": "0x29bbd1",
          "class": "private-key",
          "encrypt": false,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 1217,
          "public-exponent": "0x010001",
          "modulus":
"0x8b3a7c20618e8be08220ed8ab2c8550b65fc1aad8d4cf04fbf2be685f97eeb78fcbbad9b02cd91a3b15e990
          "modulus-size-bits": 2048
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
 }
```

2. Tras identificar la clave, ejecute el comando key delete con el atributo `label` exclusivo de la clave para eliminarla:

```
aws-cloudhsm > key delete --filter attr.label="my_key_to_delete"
{
  "error_code": 0,
  "data": {
    "message": "Key deleted successfully"
  }
}
```

3. Ejecute el comando key list con el atributo `label` exclusivo de la clave y confirme que la clave se ha eliminado. Como se muestra en el siguiente ejemplo, en el clúster de HSM no hay ninguna clave con la etiqueta `my_key_to_delete`:

```
aws-cloudhsm > key list --filter attr.label="my_key_to_delete"
{
  "error_code": 0,
  "data": {
    "matched_keys": [],
    "total_key_count": 0,
    "returned_key_count": 0
  }
}
```

Temas relacionados de

- Atributos de clave de la CLI de CloudHSM
- eliminar clave

Uso de la CLI de CloudHSM para compartir y dejar de compartir claves

Use los comandos de este tema para compartir y dejar de compartir claves en la CLI de CloudHSM. En AWS CloudHSM, el usuario criptográfico (CU) que crea la clave es el propietario de la misma. El propietario puede usar los comandos key share y key unshare para compartir y dejar de compartir la clave con otras CU. Los usuarios con quienes se comparte la clave pueden utilizarla en operaciones criptográficas, pero no pueden exportarla, eliminarla ni compartirla con otros usuarios.

Antes de que pueda compartir una clave, usted debe iniciar sesión en el HSM como el usuario de criptografía (CU) que posee la clave.

Temas

- Ejemplo: compartir y dejar de compartir una clave
- Temas relacionados de

Ejemplo: compartir y dejar de compartir una clave

Example

El siguiente ejemplo muestra cómo compartir y dejar de compartir una clave con un usuario de criptografía (CU). `alice` Junto con los comandos key share y key unshare, los comandos para compartir y dejar de compartir también requieren una clave específica mediante los filtros de claves de la CLI de CloudHSM y el nombre de usuario específico del usuario con el que se compartirá o dejará de compartir la clave.

1. Comience por ejecutar el comando key list con un filtro para obtener una clave específica y ver con quién ya se ha compartido la clave.

```
aws-cloudhsm > key list --filter attr.label="rsa_key_to_share" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000001c0686",
        "key-info": {
          "key-owners": [
            {
              "username": "cu3",
              "key-coverage": "full"
            }
          ],
          "shared-users": [
            {
              "username": "cu2",
              "key-coverage": "full"
            },
            {
              "username": "cu1",
```

```
            "key-coverage": "full"
          },
          {
            "username": "cu4",
            "key-coverage": "full"
          },
          {
            "username": "cu5",
            "key-coverage": "full"
          },
          {
            "username": "cu6",
            "key-coverage": "full"
          },
          {
            "username": "cu7",
            "key-coverage": "full"
          },
        ],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "rsa_key_to_share",
        "id": "",
        "check-value": "0xae8ff0",
        "class": "private-key",
        "encrypt": false,
        "decrypt": true,
        "token": true,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": true,
        "verify": false,
        "wrap": false,
```

```
          "wrap-with-trusted": false,
          "key-length-bytes": 1219,
          "public-exponent": "0x010001",
          "modulus":
 "0xa8855cba933cec0c21a4df0450ec31675c024f3e65b2b215a53d2bda6dcd191f75729150b59b4d86df58254
          "modulus-size-bits": 2048
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

2. Vea el resultado `shared-users` para identificar con quién se comparte actualmente la clave.
3. Para compartir esta clave con el usuario de criptografía (CU)`alice`, ingrese el siguiente comando:

```
aws-cloudhsm > key share --filter attr.label="rsa_key_to_share" attr.class=private-
key --username alice --role crypto-user
{
  "error_code": 0,
  "data": {
    "message": "Key shared successfully"
  }
}
```

   Tenga en cuenta que, junto con el comando key share, este comando utiliza la etiqueta única de la clave y el nombre del usuario con el que se compartirá la clave.
4. Ejecute el comando key list para confirmar que la clave se ha compartido con `alice`:

```
aws-cloudhsm > key list --filter attr.label="rsa_key_to_share" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000001c0686",
        "key-info": {
          "key-owners": [
            {
              "username": "cu3",
```

```
            "key-coverage": "full"
          }
        ],
        "shared-users": [
          {
            "username": "cu2",
            "key-coverage": "full"
          },
          {
            "username": "cu1",
            "key-coverage": "full"
          },
          {
            "username": "cu4",
            "key-coverage": "full"
          },
          {
            "username": "cu5",
            "key-coverage": "full"
          },
          {
            "username": "cu6",
            "key-coverage": "full"
          },
          {
            "username": "cu7",
            "key-coverage": "full"
          },
          {
            "username": "alice",
            "key-coverage": "full"
          }
        ],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "rsa_key_to_share",
        "id": "",
        "check-value": "0xae8ff0",
        "class": "private-key",
        "encrypt": false,
        "decrypt": true,
        "token": true,
```

```
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 1219,
          "public-exponent": "0x010001",
          "modulus":
 "0xa8855cba933cec0c21a4df0450ec31675c024f3e65b2b215a53d2bda6dcd191f75729150b59b4d86df58254
          "modulus-size-bits": 2048
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

5. Para dejar de compartir la misma clave con `alice`, ejecute el siguiente comando unshare:

```
aws-cloudhsm > key unshare --filter attr.label="rsa_key_to_share"
 attr.class=private-key --username alice --role crypto-user
{
  "error_code": 0,
  "data": {
    "message": "Key unshared successfully"
  }
}
```

Tenga en cuenta que, junto con el comando key unshare, este comando utiliza la etiqueta única de la clave y el nombre del usuario con el que se compartirá la clave.

6. Vuelva a ejecutar el comando key list y confirme que la clave no se ha compartido con el usuario de criptografía `alice`:

```
aws-cloudhsm > key list --filter attr.label="rsa_key_to_share" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000001c0686",
        "key-info": {
          "key-owners": [
            {
              "username": "cu3",
              "key-coverage": "full"
            }
          ],
          "shared-users": [
            {
              "username": "cu2",
              "key-coverage": "full"
            },
            {
              "username": "cu1",
              "key-coverage": "full"
            },
            {
              "username": "cu4",
              "key-coverage": "full"
            },
            {
              "username": "cu5",
              "key-coverage": "full"
            },
            {
              "username": "cu6",
              "key-coverage": "full"
            },
            {
              "username": "cu7",
              "key-coverage": "full"
            },
          ],
```

```
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "rsa",
          "label": "rsa_key_to_share",
          "id": "",
          "check-value": "0xae8ff0",
          "class": "private-key",
          "encrypt": false,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 1219,
          "public-exponent": "0x010001",
          "modulus":
 "0xa8855cba933cec0c21a4df0450ec31675c024f3e65b2b215a53d2bda6dcd191f75729150b59b4d86df58254
          "modulus-size-bits": 2048
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```


Temas relacionados de

- Atributos de clave de la CLI de CloudHSM

- key share
- key unshare
- Uso de la CLI de CloudHSM para filtrar claves

Uso de la CLI de CloudHSM para filtrar claves

Use los comandos de clave del presente tema para emplear los mecanismos de filtrado de claves estandarizados para la CLI de CloudHSM.

- key list
- key delete
- key share
- key unshare
- key set-attribute

Para seleccionar y/o filtrar claves con la CLI de CloudHSM, los comandos de clave emplean un mecanismo de filtrado estandarizado basado en Atributos de clave de la CLI de CloudHSM. Se puede especificar una tecla o un conjunto de teclas en los comandos de teclado mediante uno o más AWS CloudHSM atributos que pueden identificar una o varias teclas. El mecanismo de filtrado de claves solo funciona con las claves que el usuario que ha iniciado sesión actualmente posee y comparte, así como con todas las claves públicas del AWS CloudHSM clúster.

Temas

- Requisitos
- Filtrado para encontrar una única clave
- Errores de filtración
- Temas relacionados de

Requisitos

Para filtrar las claves, debe iniciar sesión como usuario de criptografía (CU).

Filtrado para encontrar una única clave

Tenga en cuenta que, en los siguientes ejemplos, cada atributo que se use como filtro debe escribirse en formato `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE`. Por ejemplo, si desea filtrar por el atributo de etiqueta, deberá escribir `attr.label=my_label`.

Example Use un solo atributo para encontrar una única clave.

Este ejemplo demuestra cómo filtrar para obtener una única clave mediante un solo atributo de identificación.

```
aws-cloudhsm > key list --filter attr.label="my_unique_key_label" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000001c0686",
        "key-info": {
          "key-owners": [
            {
              "username": "cu1",
              "key-coverage": "full"
            }
          ],
          "shared-users": [
            {
              "username": "alice",
              "key-coverage": "full"
            }
          ],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "rsa",
          "label": "my_unique_key_label",
          "id": "",
          "check-value": "0xae8ff0",
          "class": "private-key",
          "encrypt": false,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
```

```
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 1219,
          "public-exponent": "0x010001",
          "modulus":
 "0xa8855cba933cec0c21a4df0450ec31675c024f3e65b2b215a53d2bda6dcd191f75729150b59b4d86df58254c8f5
          "modulus-size-bits": 2048
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

Example Use múltiples atributos para encontrar una única clave.

En el siguiente ejemplo se muestra cómo encontrar una única clave mediante varios atributos de clave.

```
aws-cloudhsm > key list --filter attr.key-type=rsa attr.class=private-key attr.check-
value=0x29bbd1 --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000540011",
        "key-info": {
          "key-owners": [
            {
```

```
            "username": "cu3",
            "key-coverage": "full"
          }
        ],
        "shared-users": [
          {
            "username": "cu2",
            "key-coverage": "full"
          }
        ],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "my_crypto_user",
        "id": "",
        "check-value": "0x29bbd1",
        "class": "my_test_key",
        "encrypt": false,
        "decrypt": true,
        "token": true,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": true,
        "verify": false,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 1217,
        "public-exponent": "0x010001",
        "modulus":
"0x8b3a7c20618e8be08220ed8ab2c8550b65fc1aad8d4cf04fbf2be685f97eeb78fcbbad9b02cd91a3b15e990c2a7
        "modulus-size-bits": 2048
      }
    }
  ],
```

```
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

Example Filtrado para encontrar un conjunto de claves

En el siguiente ejemplo se muestra cómo filtrar para encontrar un conjunto de claves RSA privadas.

```
aws-cloudhsm > key list --filter attr.key-type=rsa attr.class=private-key --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000001c0686",
        "key-info": {
          "key-owners": [
            {
              "username": "my_crypto_user",
              "key-coverage": "full"
            }
          ],
          "shared-users": [
            {
              "username": "cu2",
              "key-coverage": "full"
            },
            {
              "username": "cu1",
              "key-coverage": "full"
            },
          ],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "rsa",
          "label": "rsa_key_to_share",
          "id": "",
          "check-value": "0xae8ff0",
          "class": "private-key",
          "encrypt": false,
          "decrypt": true,
          "token": true,
```

```
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 1219,
          "public-exponent": "0x010001",
          "modulus":
  "0xa8855cba933cec0c21a4df0450ec31675c024f3e65b2b215a53d2bda6dcd191f75729150b59b4d86df58254c8f5
          "modulus-size-bits": 2048
        }
      },
      {
        "key-reference": "0x0000000000540011",
        "key-info": {
          "key-owners": [
            {
              "username": "my_crypto_user",
              "key-coverage": "full"
            }
          ],
          "shared-users": [
            {
              "username": "cu2",
              "key-coverage": "full"
            }
          ],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "rsa",
          "label": "my_test_key",
          "id": "",
          "check-value": "0x29bbd1",
```

```
          "class": "private-key",
          "encrypt": false,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 1217,
          "public-exponent": "0x010001",
          "modulus":
 "0x8b3a7c20618e8be08220ed8ab2c8550b65fc1aad8d4cf04fbf2be685f97eeb78fcbbad9b02cd91a3b15e990c2a7
          "modulus-size-bits": 2048
        }
      }
    ],
    "total_key_count": 2,
    "returned_key_count": 2
  }
}
```


Errores de filtración

Algunas operaciones de claves solo pueden realizarse con una única clave cada vez. En estas operaciones, la CLI de CloudHSM generará un error en caso de que los criterios de filtrado no estén lo suficientemente definidos o en caso de que existan varias claves coincidentes con los criterios. A continuación, se muestra un ejemplo de este tipo con la eliminación de clave.

Example Error de filtrado por coincidencia con demasiadas claves

```
aws-cloudhsm > key delete --filter attr.key-type=rsa
{
```

```
  "error_code": 1,
  "data": "Key selection criteria matched 48 keys. Refine selection criteria to select
 a single key."
}
```

Temas relacionados de

- Atributos de clave de la CLI de CloudHSM

Cómo marcar una clave como de confianza con la CLI de CloudHSM

El contenido de esta sección proporciona instrucciones sobre el uso de la CLI de CloudHSM para marcar una clave como de confianza.

1. Con el comando login de la CLI de CloudHSM, inicie sesión como usuario de criptografía (CU).
2. Utilice el comando key list para identificar la referencia clave de la clave que quiere marcar como de confianza. En el siguiente ejemplo, se muestra la clave con la etiqueta `key_to_be_trusted`.

   ```
   aws-cloudhsm > key list --filter attr.label=test_aes_trusted
           {
     "error_code": 0,
     "data": {
       "matched_keys": [
         {
           "key-reference": "0x0000000000200333",
           "attributes": {
             "label": "test_aes_trusted"
           }
         }
       ],
       "total_key_count": 1,
       "returned_key_count": 1
     }
   }
   ```

3. Con el comando logout, cierre sesión como usuario de criptografía (CU).
4. Con el comando login, inicie sesión como administrador.
5. Con el comando key set-attribute y la referencia clave que identificó en el paso 2, configure el valor de confianza de la clave como verdadero:

```
aws-cloudhsm > key set-attribute --filter key-reference=<Key Reference> --name
 trusted --value true
{
  "error_code": 0,
  "data": {
    "message": "Attribute set successfully"
  }
}
```

Gestión de claves con la KMU y la CMU

Si utilizas la última serie de versiones del SDK, usa la CLI de CloudHSM para administrar las claves AWS CloudHSM del clúster.

Si utilizas la serie de versiones anteriores del SDK, puedes administrar las claves de los HSM de tu AWS CloudHSM clúster mediante la herramienta de línea de comandos key_mgmt_util. Para poder administrar las claves, debe iniciar el AWS CloudHSM cliente, iniciar key_mgmt_util e iniciar sesión en los HSM. Para obtener más información, consulte Introducción a key_mgmt_util.

- En Usar claves de confianza se describe cómo usar los atributos de la biblioteca PKCS #11 y la CMU para crear claves de confianza para proteger los datos.
- En Generar claves se proporcionan instrucciones sobre cómo generar claves, incluidas claves simétricas, claves RSA y claves EC.
- En Importar claves se proporciona información sobre cómo los propietarios de las claves importan las claves.
- En Exportar claves se proporciona información sobre cómo los propietarios de las claves exportan las claves.
- En Eliminar claves se proporciona información sobre cómo los propietarios de las claves eliminan las claves.
- En Compartir y dejar de compartir claves se detalla cómo los propietarios de las claves comparten y dejan de compartir las claves.

Generación de claves

Para generar claves en el HSM, utilice el comando que corresponde al tipo de clave que desea generar.

Temas

- Generación de claves simétricas
- Generación de pares de claves RSA
- Generación de pares de claves ECC (criptografía de curva elíptica)

Generación de claves simétricas

Utilice el genSymKeycomando para generar claves AES y otros tipos de claves simétricas. Para ver todas las opciones disponibles, utilice el comando genSymKey -h.

El siguiente ejemplo crea una clave AES de 256 bits.

```
Command: genSymKey -t 31 -s 32 -l aes256
CfM3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

Symmetric Key Created.  Key Handle: 524295

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Generación de pares de claves RSA

Para generar un key pair de RSA, utilice el comando KeyPairGenRSA. Para ver todas las opciones disponibles, utilice el comando genRSAKeyPair -h.

El siguiente ejemplo genera un par de claves RSA de 2048 bits.

```
Command: genRSAKeyPair -m 2048 -e 65537 -l rsa2048
CfM3GenerateKeyPair returned: 0x00 : HSM Return: SUCCESS

CfM3GenerateKeyPair:    public key handle: 524294    private key handle: 524296

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Generación de pares de claves ECC (criptografía de curva elíptica)

Para generar un key pair ECC, utilice el comando GeneCC KeyPair. Para ver todas las opciones disponibles, incluida una lista de las curvas elípticas admitidas, use el comando genECCKeyPair -h.

En el ejemplo siguiente se genera un par de claves ECC con la curva elíptica P-384 definida en la publicación de NIST FIPS 186-4.

```
Command: genECCKeyPair -i 14 -l ecc-p384
Cfm3GenerateKeyPair returned: 0x00 : HSM Return: SUCCESS

Cfm3GenerateKeyPair:    public key handle: 524297    private key handle: 524298

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Cómo importar claves

Para importar claves secretas —es decir, claves simétricas y claves privadas asimétricas— en el HSM, primero debe crear una clave encapsulada en el HSM. Puede importar claves públicas directamente sin una clave de encapsulamiento.

Temas

- Cómo importar claves secretas
- Cómo importar claves públicas

Cómo importar claves secretas

Complete los pasos siguientes para importar una clave secreta. Antes de importar una clave secreta, guárdela en un archivo. Guarde las claves simétricas como bytes sin procesar y las claves privadas asimétricas en formato PEM.

En este ejemplo se muestra cómo importar una clave secreta sin cifrar desde un archivo hasta el HSM. Para importar una clave cifrada de un archivo al HSM, utilice el comando. unWrapKey

Para importar una clave secreta

1. Utilice el genSymKeycomando para crear una clave de empaquetado. El siguiente comando crea una clave de encapsulamiento AES de 128 bits que solo es válida durante la sesión actual. Puede utilizar una clave de sesión o una clave persistente como clave de encapsulación.

```
Command: genSymKey -t 31 -s 16 -sess -l import-wrapping-key
Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

Symmetric Key Created.  Key Handle: 524299

Cluster Error Status
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

2. Utilice uno de los siguientes comandos, en función del tipo de clave secreta que va a importar.

   - Para importar una clave simétrica, utilice el imSymKeycomando. El siguiente comando importa una clave AES de un archivo denominado `aes256.key` utilizando la clave de encapsulamiento creada en el paso anterior. Para ver todas las opciones disponibles, utilice el comando imSymKey -h.

```
Command: imSymKey -f aes256.key -t 31 -l aes256-imported -w 524299
Cfm3WrapHostKey returned: 0x00 : HSM Return: SUCCESS

Cfm3CreateUnwrapTemplate returned: 0x00 : HSM Return: SUCCESS

Cfm3UnWrapKey returned: 0x00 : HSM Return: SUCCESS

Symmetric Key Unwrapped.  Key Handle: 524300

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

   - Para importar una clave privada asimétrica, utilice el importPrivateKeycomando. El siguiente comando importa una clave privada de un archivo denominado `rsa2048.key` utilizando la clave de encapsulamiento creada en el paso anterior. Para ver todas las opciones disponibles, utilice el comando importPrivateKey -h.

```
Command: importPrivateKey -f rsa2048.key -l rsa2048-imported -w 524299
```

```
BER encoded key length is 1216

Cfm3WrapHostKey returned: 0x00 : HSM Return: SUCCESS

Cfm3CreateUnwrapTemplate returned: 0x00 : HSM Return: SUCCESS

Cfm3UnWrapKey returned: 0x00 : HSM Return: SUCCESS

Private Key Unwrapped.  Key Handle: 524301

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Cómo importar claves públicas

Utilice el importPubKeycomando para importar una clave pública. Para ver todas las opciones disponibles, utilice el comando importPubKey -h.

El siguiente ejemplo importa una clave pública RSA de un archivo denominado `rsa2048.pub`.

```
Command: importPubKey -f rsa2048.pub -l rsa2048-public-imported
Cfm3CreatePublicKey returned: 0x00 : HSM Return: SUCCESS

Public Key Handle: 524302

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Cómo exportar claves

Para exportar claves secretas —es decir, claves simétricas y claves privadas asimétricas— desde el HSM, primero debes crear una clave encapsulada. Puede exportar claves públicas directamente sin una clave de encapsulamiento.

Solo el propietario de la clave puede exportarla. Los usuarios con los que se comparte la clave pueden utilizarla en operaciones criptográficas, pero no pueden exportarla. Cuando ejecute este ejemplo, asegúrese de que exporta una clave que haya creado.

> **Important**
>
> El exSymKeycomando escribe una copia en texto plano (sin cifrar) de la clave secreta en un archivo. El proceso de exportación requiere una clave de encapsulación, pero la clave que hay en el archivo no es una clave encapsulada. Para exportar una copia encapsulada (cifrada) de una clave, utilice el comando wrapKey.

Temas

- Cómo exportar claves secretas
- Cómo exportar claves públicas

Cómo exportar claves secretas

Complete los pasos siguientes para exportar una clave secreta.

Para exportar una clave secreta

1. Utilice el genSymKeycomando para crear una clave de empaquetado. El siguiente comando crea una clave de encapsulamiento AES de 128 bits que solo es válida durante la sesión actual.

```
Command: genSymKey -t 31 -s 16 -sess -l export-wrapping-key
Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

Symmetric Key Created.  Key Handle: 524304

Cluster Error Status
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

2. Utilice uno de los siguientes comandos, en función del tipo de clave secreta que va a exportar.

   - Para exportar una clave simétrica, utilice el exSymKeycomando. El siguiente ejemplo exporta una clave AES a un archivo denominado `aes256.key.exp`. Para ver todas las opciones disponibles, utilice el comando exSymKey -h.

```
Command: exSymKey -k 524295 -out aes256.key.exp -w 524304
Cfm3WrapKey returned: 0x00 : HSM Return: SUCCESS

Cfm3UnWrapHostKey returned: 0x00 : HSM Return: SUCCESS
```

```
Wrapped Symmetric Key written to file "aes256.key.exp"
```

> **Note**
>
> La salida del comando indica que se ha escrito una "Wrapped Symmetric Key" (clave simétrica encapsulada) en el archivo de salida. Sin embargo, el archivo de salida contiene una clave sin cifrar (sin encapsular). Para exportar una clave encapsulada (cifrada) a un archivo, utilice el comando wrapKey.

- Para exportar una clave privada, utilice el comando exportPrivateKey. El siguiente comando exporta una clave privada a un archivo denominado `rsa2048.key.exp`. Para ver todas las opciones disponibles, utilice el comando exportPrivateKey -h.

```
Command: exportPrivateKey -k 524296 -out rsa2048.key.exp -w 524304
Cfm3WrapKey returned: 0x00 : HSM Return: SUCCESS

Cfm3UnWrapHostKey returned: 0x00 : HSM Return: SUCCESS

PEM formatted private key is written to rsa2048.key.exp
```

Cómo exportar claves públicas

Utilice el comando exportPubKey para exportar una clave pública. Para ver todas las opciones disponibles, utilice el comando exportPubKey -h.

El siguiente ejemplo exporta una clave pública RSA a un archivo denominado `rsa2048.pub.exp`.

```
Command: exportPubKey -k 524294 -out rsa2048.pub.exp
PEM formatted public key is written to rsa2048.pub.key

Cfm3ExportPubKey returned: 0x00 : HSM Return: SUCCESS
```

Eliminación de claves

Utilice el comando deleteKey para eliminar una clave, como en el siguiente ejemplo. Solo el propietario de la clave puede eliminar una clave.

```
Command: deleteKey -k 524300
```

```
Cfm3DeleteKey returned: 0x00 : HSM Return: SUCCESS

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Cómo compartir y dejar de compartir claves

En AWS CloudHSM, la CU que crea la clave es la propietaria de la misma. El propietario administra la clave, puede exportarla y eliminarla y, además, puede utilizar la clave en operaciones criptográficas. El propietario también puede compartir la clave con otros usuarios CU. Los usuarios con quien se comparte la clave pueden utilizar la clave en operaciones criptográficas, pero no pueden exportarla ni eliminarla ni tampoco pueden compartirla con otros usuarios.

Puede compartir las claves con otros usuarios de la CU al crear la clave, por ejemplo, mediante el `-u` parámetro de los comandos genSymKeyo GenRSA KeyPair. Para compartir claves existentes con otro usuario del HSM, utilice la herramienta de línea de comandos cloudhsm_mgmt_util. Este proceso es distinto de la mayoría de las tareas documentadas en esta sección, que utilizan la herramienta de línea de comandos key_mgmt_util.

Para poder compartir una clave, debe iniciar cloudhsm_mgmt_util, habilitar el cifrado e iniciar sesión en los HSM. end-to-end Para compartir una clave, inicie sesión en el HSM como el usuario de criptografía (CU) que posee la clave. Solo los propietarios de clave pueden compartir una clave.

Utilice el comando shareKey para compartir o dejar de compartir una clave especificando el identificador de la clave y el ID del usuario o usuarios. Para compartir o dejar de compartir con varios usuarios, especifique una lista separada por comas de ID de usuario. Para compartir una clave, utilice 1 como último parámetro del comando, como se muestra en el siguiente ejemplo. Para dejar de compartirla, utilice `0`.

```
aws-cloudhsm>shareKey 524295 4 1
************************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
************************************************************

Do you want to continue(y/n)?y
```

```
shareKey success on server 0(10.0.2.9)
shareKey success on server 1(10.0.3.11)
shareKey success on server 2(10.0.1.12)
```

A continuación, se muestra la sintaxis del comando shareKey.

```
aws-cloudhsm>shareKey <key handle> <user ID> <Boolean: 1 for share, 0 for unshare>
```

Cómo marcar una clave como de confianza con la CMU

El contenido de esta sección proporciona instrucciones sobre el uso de la CMU para marcar una clave como de confianza.

1. Con el comando LoginHsm, inicie sesión como responsable de criptografía (CO).
2. Use el comando setAttribute con `OBJ_ATTR_TRUSTED` (valor `134`) establecido en true (`1`).

   ```
   setAttribute <Key Handle> 134 1
   ```

Administración de clústeres clonados

Utilice la Utilidad de administración de CloudHSM (CMU) para sincronizar un clúster en una región remota, si el clúster de esa región se creó originalmente a partir de la copia de seguridad de un clúster de otra región. Supongamos que ha copiado un clúster en otra región (destino) y, posteriormente, desea sincronizar los cambios del clúster original (origen). En escenarios como este, se utiliza la CMU para sincronizar los clústeres. Para ello, debe crear un nuevo archivo de configuración de la CMU, especificar los módulos de seguridad de hardware (HSM) de ambos clústeres del nuevo archivo y, a continuación, utilizar la CMU para conectarse al clúster con ese archivo.

Cómo usar CMU en clústeres clonados

1. Cree una copia del archivo de configuración actual y cambie el nombre de la copia por otro.

   Por ejemplo, utilice las siguientes ubicaciones de archivos para buscar y crear una copia del archivo de configuración actual y, a continuación, cambie el nombre de la copia de `cloudhsm_mgmt_config.cfg` a `syncConfig.cfg`.

   - Linux: `/opt/cloudhsm/etc/cloudhsm_mgmt_config.cfg`

- Windows: `C:\ProgramData\Amazon\CloudHSM\data\cloudhsm_mgmt_config.cfg`

2. En la copia cuyo nombre ha cambiado, añada la IP de la interfaz de red elástica (ENI) del HSM de destino (el HSM de la región extranjera que debe sincronizarse). Se recomienda añadir el HSM de destino debajo del HSM de origen.

```
{
    ...
    "servers": [
        {
            ...
            "hostname": "<ENI Source IP>",
            ...
        },
        {
            ...
            "hostname": "<ENI Destination IP>",
            ...
        }
    ]
}
```

   Para obtener más información sobre los filtros de direcciones IP, consulte the section called "Obtención de una dirección IP para un HSM".

3. Inicialice la CMU con el nuevo archivo de configuración:

   Linux

```
$ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/userSync.cfg
```

   Windows

```
C:\Program Files\Amazon\CloudHSM>cloudhsm_mgmt_util.exe C:\ProgramData\Amazon
\CloudHSM\data\userSync.cfg
```

4. Compruebe los mensajes de estado devueltos con el fin de asegurarse de que está conectado a todos los HSM y determinar cuál de las direcciones IP de la ENI devueltas corresponde a cada clúster. Use syncUser y SyncKey para sincronizar manualmente los usuarios y las claves. Para obtener más información, consulte syncUser y syncKey.

Obtención de una dirección IP para un HSM

Utilice esta sección para obtener una dirección IP para un HSM.

Para obtener una dirección IP para un HSM (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.
2. Para cambiar la región de AWS, utilice el selector de regiones en la esquina superior derecha de la página.
3. Para abrir la página de detalles del clúster, en la tabla de clústeres, elija el ID del clúster.
4. Para obtener la dirección IP, vaya a la pestaña HSM y elija una de las direcciones IP que aparecen en la lista Dirección IP de ENI.

Para obtener una dirección IP para un HSM (CLI)

- Obtenga la dirección IP de un HSM mediante el describe-clusters comando de la CLI. En el resultado del comando, la dirección IP de los HSM son los valores de `EniIp`.

```
$  aws cloudhsmv2 describe-clusters


{
    "Clusters": [
        { ... }
            "Hsms": [
                {
...
                    "EniIp": "10.0.0.9",
...
                },
                {
...
                    "EniIp": "10.0.1.6",
...
```

Temas relacionados de

- syncUser

- syncKey
- Copia de una copia de seguridad entre regiones

AWS CloudHSM herramientas de línea de comandos

En este tema se describen las herramientas de línea de comandos disponibles para administrar y usar AWS CloudHSM.

Temas

- Mäs información sobre las herramientas de línea de comando
- Herramienta de configuración
- Interfaz de la línea de comandos (CLI) de CloudHSM
- Utilidad de administración de CloudHSM (CMU)
- Utilidad de administración de claves (KMU)

Mäs información sobre las herramientas de línea de comando

Además de la interfaz de línea de comandos (CLI) de AWS que utiliza para administrar los recursos de AWS, AWS CloudHSM ofrece herramientas de línea de comandos para crear y administrar los usuarios y claves de los HSM en sus HSM. En AWS CloudHSM , usa la conocida CLI para administrar su clúster y las herramientas de línea de comandos de CloudHSM para administrar su HSM.

Estas son las distintas herramientas de línea de comandos:

Cómo gestionar clústeres y HSM

Comandos de CloudHSMv2 en los cmdlets CLI y PowerShell HSM2 del módulo AWSPowerShell

- Estas herramientas obtienen, crean, eliminan y etiquetan clústeres y HSM: AWS CloudHSM
- Para usar los comandos de CloudHSMv2 en la CLI, debe instalar y configurar la CLI.
- Los PowerShell cmdlets HSM2 del AWSPowerShell módulo están disponibles en un módulo de Windows PowerShell y en un módulo Core multiplataforma. PowerShell

Cómo gestionar usuarios HSM

La CLI de CloudHSM

- Utilice la CLI de CloudHSM para crear usuarios, eliminar usuarios, enumerar usuarios, cambiar las contraseñas de los usuarios y actualizar la autenticación multifactor (MFA) de los usuarios. No se incluye en el software cliente de AWS CloudHSM. Para obtener instrucciones sobre la instalación de esta herramienta, consulte Instalación y configuración de la CLI de CloudHSM.

Herramientas ayudantes

Hay dos herramientas que le ayudan a utilizar herramientas y bibliotecas de software AWS CloudHSM :

- La herramienta de configuración actualiza sus archivos de configuración del clienteCloudHSM. Esto permite AWS CloudHSM sincronizar los HSM en un clúster.

  AWS CloudHSM ofrece dos versiones principales y Client SDK 5 es la más reciente. Ofrece diversas ventajas con respecto a la versión SDK 3 de cliente (la serie anterior).
- pkpspeed mide el desempeño del hardware del HSM con independencia de las bibliotecas de software.

Herramientas para los SDK anteriores

Utilice la herramienta de administración de claves (KMU) para crear, eliminar, importar y exportar claves simétricas y pares de claves asimétricas:

- key_mgmt_util. Esta herramienta está incluida en el software cliente de AWS CloudHSM .

Utilice la herramienta de administración de CloudHSM (CMU) para crear y eliminar usuarios de HSM, incluida la implementación de la autenticación de cuórum en las tareas de administración de usuarios.

- cloudhsm_mgmt_util. Esta herramienta está incluida en el software cliente de AWS CloudHSM .

Herramienta de configuración

AWS CloudHSM sincroniza automáticamente los datos entre todos los módulos de seguridad de hardware (HSM) de un clúster. La herramienta configure actualiza los datos de los HSM en

los archivos de configuración que utilizan los mecanismos de sincronización. Utilice configure para actualizar los datos de los HSM antes de utilizar las herramientas de línea de comandos, especialmente cuando los HSM del clúster han cambiado.

AWS CloudHSM incluye dos versiones principales del SDK de cliente:

- SDK 5 de cliente: este es nuestro SDK de cliente más reciente y predeterminado. Para obtener información sobre los beneficios y las ventajas que ofrece, consulte Ventajas del SDK 5 de cliente.
- SDK 3 de cliente: este es nuestro SDK de cliente anterior. Incluye un completo paquete de componentes para la compatibilidad de aplicaciones basadas en lenguaje y plataforma, así como herramientas de gestión.

Para obtener instrucciones sobre cómo migrar del SDK de cliente 3 al SDK de cliente 5, consulteMigración del SDK 3 de cliente al SDK 5 de cliente.

Temas

- Herramienta de configuración de SDK 5 de cliente
- Herramienta de configuración de SDK 3 de cliente

Herramienta de configuración de SDK 5 de cliente

Utilice la herramienta de configuración SDK 5 de cliente para actualizar los archivos de configuración del lado del cliente.

Cada componente de SDK 5 de cliente incluye una herramienta de configuración con un designador del componente en el nombre de archivo de la herramienta de configuración. Por ejemplo, la biblioteca PKCS #11 de SDK 5 de cliente incluye una herramienta de configuración denominada `configure-pkcs11` en Linux o `configure-pkcs11.exe` en Windows.

Sintaxis

PKCS #11

```
configure-pkcs11[ .exe ]
              -a <ENI IP address>
              [--hsm-ca-cert <customerCA certificate file path>]
              [--cluster-id <cluster ID>]
              [--endpoint <endpoint>]
```

```
            [--region <region>]
            [--server-client-cert-file <client certificate file path>]
            [--server-client-key-file <client key file path>]
            [--log-level <error | warn | info | debug | trace>]
                 Default is <info>
            [--log-rotation <daily | weekly>]
                 Default is <daily>
            [--log-file <file name with path>]
                 Default is </opt/cloudhsm/run/cloudhsm-pkcs11.log>
                 Default for Windows is <C:\\Program Files\\Amazon\\CloudHSM\
\cloudhsm-pkcs11.log>
            [--log-type <file | term>]
                 Default is <file>
            [-h | --help]
            [-V | --version]
            [--disable-key-availability-check]
            [--enable-key-availability-check]
            [--disable-validate-key-at-init]
            [--enable-validate-key-at-init]
                 This is the default for PKCS #11
```

OpenSSL

```
configure-dyn[ .exe ]
            -a <ENI IP address>
            [--hsm-ca-cert <customerCA certificate file path>]
            [--cluster-id <cluster ID>]
            [--endpoint <endpoint>]
            [--region <region>]
            [--server-client-cert-file <client certificate file path>]
            [--server-client-key-file <client key file path>]
            [--log-level <error | warn | info | debug | trace>]
                 Default is <error>
            [--log-type <file | term>]
                 Default is <term>
            [-h | --help]
            [-V | --version]
            [--disable-key-availability-check]
            [--enable-key-availability-check]
            [--disable-validate-key-at-init]
                 This is the default for OpenSSL
            [--enable-validate-key-at-init]
```

JCE

```
configure-jce[ .exe ]
             -a <ENI IP address>
             [--hsm-ca-cert <customerCA certificate file path>]
             [--cluster-id <cluster ID>]
             [--endpoint <endpoint>]
             [--region <region>]
             [--server-client-cert-file <client certificate file path>]
             [--server-client-key-file <client key file path>]
             [--log-level <error | warn | info | debug | trace>]
                  Default is <info>
             [--log-rotation <daily | weekly>]
                  Default is <daily>
             [--log-file <file name with path>]
                  Default is </opt/cloudhsm/run/cloudhsm-jce.log>
                  Default for Windows is <C:\\Program Files\\Amazon\\CloudHSM\
\cloudhsm-jce.log>
             [--log-type <file | term>]
                  Default is <file>
             [-h | --help]
             [-V | --version]
             [--disable-key-availability-check]
             [--enable-key-availability-check]
             [--disable-validate-key-at-init]
                  This is the default for JCE
             [--enable-validate-key-at-init]
```

CloudHSM CLI

```
configure-cli[ .exe ]
             -a <ENI IP address>
             [--hsm-ca-cert <customerCA certificate file path>]
             [--cluster-id <cluster ID>]
             [--endpoint <endpoint>]
             [--region <region>]
             [--server-client-cert-file <client certificate file path>]
             [--server-client-key-file <client key file path>]
             [--log-level <error | warn | info | debug | trace>]
                  Default is <info>
             [--log-rotation <daily | weekly>]
                  Default is <daily>
             [--log-file <file name with path>]
```

```
                Default for Linux is </opt/cloudhsm/run/cloudhsm-cli.log>
                Default for Windows is <C:\\Program Files\\Amazon\\CloudHSM\
\cloudhsm-cli.log>
            [--log-type <file | term>]
                Default setting is <file>
            [-h | --help]
            [-V | --version]
            [--disable-key-availability-check]
            [--enable-key-availability-check]
            [--disable-validate-key-at-init]
                This is the default for CloudHSM CLI
            [--enable-validate-key-at-init]
```

Configuraciones avanzadas

Para obtener una lista de las configuraciones avanzadas específicas de la herramienta de configuración de SDK 5 de cliente, consulte Configuraciones avanzadas de la herramienta de configuración de SDK 5 de cliente.

> **Important**
>
> Después de realizar cualquier cambio en la configuración, necesita reiniciar la aplicación para que los cambios surtan efecto.

Ejemplos

Estos ejemplos muestran cómo utilizar la herramienta de configuración para SDK 5 de cliente.

Iniciar SDK 5 de cliente

Example

Este ejemplo utiliza el parámetro `-a` para actualizar los datos de HSM para SDK 5 de cliente. Para usar el parámetro `-a`, debe tener la dirección IP de uno de los HSM de su clúster.

PKCS #11 library

Arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM de su clúster.

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 -a <HSM IP addresses>
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM del clúster.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-pkcs11.exe" -a <HSM IP
 addresses>
```

OpenSSL Dynamic Engine

Arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM del clúster.

```
$ sudo /opt/cloudhsm/bin/configure-dyn -a <HSM IP addresses>
```

JCE provider

Arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM del clúster.

```
$ sudo /opt/cloudhsm/bin/configure-jce -a <HSM IP addresses>
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM del clúster.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-jce.exe" -a <HSM IP addresses>
```

CloudHSM CLI

Arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de los HSM de su clúster.

```
$ sudo /opt/cloudhsm/bin/configure-cli -a <The ENI IP addresses of the HSMs>
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de los HSM de su clúster.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-cli.exe" -a <The ENI IP
 addresses of the HSMs>
```

> **Note**
>
> puede usar el parámetro `--cluster-id` en lugar de `-a <HSM_IP_ADDRESSES>`. Para ver los requisitos de uso de `--cluster-id`, consulte Herramienta de configuración de SDK 5 de cliente.

Para obtener más información sobre el parámetro `-a`, consulte the section called “Parámetros”.

Especifique el clúster, la región y el punto de conexión de SDK 5 de cliente

Example

En este ejemplo, se utiliza el parámetro `cluster-id` para iniciar SDK 5 de cliente mediante una llamada `DescribeClusters`.

PKCS #11 library

Arranque de una instancia EC2 de Linux para SDK 5 de cliente con **`cluster-id`**

- Utilice el ID del clúster `cluster-1234567` para especificar la dirección IP de un HSM del clúster.

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 --cluster-id cluster-1234567
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente con **`cluster-id`**

- Use el ID del clúster `cluster-1234567` para especificar la dirección IP de un HSM del clúster.

```
"C:\Program Files\Amazon\CloudHSM\configure-pkcs11.exe" --cluster-
id cluster-1234567
```

OpenSSL Dynamic Engine

Arranque de una instancia EC2 de Linux para SDK 5 de cliente con **`cluster-id`**

- Use el ID del clúster `cluster-1234567` para especificar la dirección IP de un HSM del clúster.

```
$ sudo /opt/cloudhsm/bin/configure-dyn --cluster-id cluster-1234567
```

JCE provider

Arranque de una instancia EC2 de Linux para SDK 5 de cliente con **`cluster-id`**

- Use el ID del clúster `cluster-1234567` para especificar la dirección IP de un HSM del clúster.

```
$ sudo /opt/cloudhsm/bin/configure-jce --cluster-id cluster-1234567
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente con **`cluster-id`**

- Use el ID del clúster `cluster-1234567` para especificar la dirección IP de un HSM del clúster.

```
"C:\Program Files\Amazon\CloudHSM\configure-jce.exe" --cluster-
id cluster-1234567
```

CloudHSM CLI

Arranque de una instancia EC2 de Linux para SDK 5 de cliente con **`cluster-id`**

- Use el ID del clúster `cluster-1234567` para especificar la dirección IP de un HSM del clúster.

```
$ sudo /opt/cloudhsm/bin/configure-cli --cluster-id cluster-1234567
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente con **`cluster-id`**

- Use el ID del clúster `cluster-1234567` para especificar la dirección IP de un HSM del clúster.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-cli.exe" --cluster-
id cluster-1234567
```

Puede usar los parámetros `--region` y `--endpoint` junto con el parámetro `cluster-id` para especificar la forma en que el sistema realiza la llamada `DescribeClusters`. Por ejemplo, si la región del clúster es diferente a la que está configurada como predeterminada de la CLI de AWS, debe usar el parámetro `--region` para usar esa región. Además, puede especificar el punto de enlace de la AWS CloudHSM API que se va a utilizar para la llamada, lo que puede ser necesario para varias configuraciones de red, como el uso de puntos de enlace de la interfaz de VPC que no utilizan el nombre de host de DNS predeterminado. AWS CloudHSM

PKCS #11 library

Cómo iniciar una instancia EC2 de Linux con un punto de conexión y una región personalizados

- Usa la herramienta de configuración para especificar la dirección IP de un HSM de tu clúster con una región y un punto final personalizados.

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 --cluster-id cluster-1234567 --
region us-east-1 --endpoint https://cloudhsmv2.us-east-1.amazonaws.com
```

Cómo iniciar una instancia EC2 de Windows con un punto de conexión y una región

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM de su clúster con una región y un punto final personalizados.

```
C:\Program Files\Amazon\CloudHSM\configure-pkcs11.exe --cluster-
id cluster-1234567--region us-east-1 --endpoint https://cloudhsmv2.us-
east-1.amazonaws.com
```

OpenSSL Dynamic Engine

Cómo iniciar una instancia EC2 de Linux con un punto de conexión y una región personalizados

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM de su clúster con una región y un punto final personalizados.

```
$ sudo /opt/cloudhsm/bin/configure-dyn --cluster-id cluster-1234567 --region us-
east-1 --endpoint https://cloudhsmv2.us-east-1.amazonaws.com
```

JCE provider

Cómo iniciar una instancia EC2 de Linux con un punto de conexión y una región personalizados

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM de su clúster con una región y un punto final personalizados.

```
$ sudo /opt/cloudhsm/bin/configure-jce --cluster-id cluster-1234567 --region us-
east-1 --endpoint https://cloudhsmv2.us-east-1.amazonaws.com
```

Cómo iniciar una instancia EC2 de Windows con un punto de conexión y una región

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM de su clúster con una región y un punto final personalizados.

```
"C:\Program Files\Amazon\CloudHSM\configure-jce.exe" --cluster-
id cluster-1234567 --region us-east-1 --endpoint https://cloudhsmv2.us-
east-1.amazonaws.com
```

CloudHSM CLI

Cómo iniciar una instancia EC2 de Linux con un punto de conexión y una región personalizados

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM de su clúster con una región y un punto final personalizados.

```
$ sudo /opt/cloudhsm/bin/configure-cli --cluster-id cluster-1234567 --region us-east-1 --endpoint https://cloudhsmv2.us-east-1.amazonaws.com
```

Cómo iniciar una instancia EC2 de Windows con un punto de conexión y una región

- Utilice la herramienta de configuración para especificar la dirección IP de un HSM de su clúster con una región y un punto final personalizados.

```
"C:\Program Files\Amazon\CloudHSM\configure-cli.exe" --cluster-id cluster-1234567 --region us-east-1 --endpoint https://cloudhsmv2.us-east-1.amazonaws.com
```

Para obtener más información acerca de los parámetros `--cluster-id`, `--region` y `--endpoint`, consulte the section called “Parámetros”.

Actualización del certificado y la clave del cliente para la autenticación mutua de TLS cliente-servidor

Example

En este ejemplo se muestra cómo utilizar los `--server-client-key-file` parámetros `server-client-cert-file` y para reconfigurar el SSL especificando una clave personalizada y un certificado SSL para AWS CloudHSM

PKCS #11 library

Cómo usar un certificado y una clave personalizados para la autenticación mutua entre cliente y servidor de TLS con SDK 5 de cliente en Linux

1. Copie la clave y el certificado en el directorio adecuado.

```
$ sudo cp ssl-client.crt /opt/cloudhsm/etc
sudo cp ssl-client.key /opt/cloudhsm/etc
```

2. Use la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 \
            --server-client-cert-file /opt/cloudhsm/etc/ssl-client.crt \
            --server-client-key-file /opt/cloudhsm/etc/ssl-client.key
```

Cómo usar un certificado y una clave personalizados para la autenticación mutua TLS cliente-servidor con SDK 5 de cliente en Windows

1. Copie la clave y el certificado en el directorio adecuado.

```
cp ssl-client.crt C:\ProgramData\Amazon\CloudHSM\ssl-client.crt
cp ssl-client.key C:\ProgramData\Amazon\CloudHSM\ssl-client.key
```

2. Con un PowerShell intérprete, utilice la herramienta de configuración para especificar `ssl-client.crt` y`ssl-client.key`.

```
& "C:\Program Files\Amazon\CloudHSM\bin\configure-pkcs11.exe" `
            --server-client-cert-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.crt `
            --server-client-key-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.key
```

OpenSSL Dynamic Engine

Cómo usar un certificado y una clave personalizados para la autenticación mutua entre cliente y servidor de TLS con SDK 5 de cliente en Linux

1. Copie la clave y el certificado en el directorio adecuado.

```
$ sudo cp ssl-client.crt /opt/cloudhsm/etc
sudo cp ssl-client.key /opt/cloudhsm/etc
```

2. Use la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
$ sudo /opt/cloudhsm/bin/configure-dyn \
            --server-client-cert-file /opt/cloudhsm/etc/ssl-client.crt \
            --server-client-key-file /opt/cloudhsm/etc/ssl-client.key
```

JCE provider

Cómo usar un certificado y una clave personalizados para la autenticación mutua entre cliente y servidor de TLS con SDK 5 de cliente en Linux

1. Copie la clave y el certificado en el directorio adecuado.

```
$ sudo cp ssl-client.crt /opt/cloudhsm/etc
sudo cp ssl-client.key /opt/cloudhsm/etc
```

2. Use la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
$ sudo /opt/cloudhsm/bin/configure-jce \
            --server-client-cert-file /opt/cloudhsm/etc/ssl-client.crt \
            --server-client-key-file /opt/cloudhsm/etc/ssl-client.key
```

Cómo usar un certificado y una clave personalizados para la autenticación mutua TLS cliente-servidor con SDK 5 de cliente en Windows

1. Copie la clave y el certificado en el directorio adecuado.

```
cp ssl-client.crt C:\ProgramData\Amazon\CloudHSM\ssl-client.crt
cp ssl-client.key C:\ProgramData\Amazon\CloudHSM\ssl-client.key
```

2. Con un PowerShell intérprete, utilice la herramienta de configuración para especificar `ssl-client.crt` y`ssl-client.key`.

```
& "C:\Program Files\Amazon\CloudHSM\bin\configure-jce.exe" `
            --server-client-cert-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.crt `
            --server-client-key-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.key
```

CloudHSM CLI

Cómo usar un certificado y una clave personalizados para la autenticación mutua entre cliente y servidor de TLS con SDK 5 de cliente en Linux

1. Copie la clave y el certificado en el directorio adecuado.

```
$ sudo cp ssl-client.crt /opt/cloudhsm/etc
sudo cp ssl-client.key /opt/cloudhsm/etc
```

2. Use la herramienta de configuración para especificar `ssl-client.crt` y `ssl-client.key`.

```
$ sudo /opt/cloudhsm/bin/configure-cli \
            --server-client-cert-file /opt/cloudhsm/etc/ssl-client.crt \
            --server-client-key-file /opt/cloudhsm/etc/ssl-client.key
```

Cómo usar un certificado y una clave personalizados para la autenticación mutua TLS cliente-servidor con SDK 5 de cliente en Windows

1. Copie la clave y el certificado en el directorio adecuado.

```
cp ssl-client.crt C:\ProgramData\Amazon\CloudHSM\ssl-client.crt
cp ssl-client.key C:\ProgramData\Amazon\CloudHSM\ssl-client.key
```

2. Con un PowerShell intérprete, utilice la herramienta de configuración para especificar `ssl-client.crt` y`ssl-client.key`.

```
& "C:\Program Files\Amazon\CloudHSM\bin\configure-cli.exe" `
            --server-client-cert-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.crt `
            --server-client-key-file C:\ProgramData\Amazon\CloudHSM\ssl-
client.key
```

Para obtener más información acerca de los parámetros `server-client-cert-file` y `--server-client-key-file`, consulte the section called “Parámetros”.

Desactivación de la configuración de durabilidad de las claves del cliente

Example

En este ejemplo, se utiliza el parámetro `--disable-key-availability-check` para deshabilitar la configuración de durabilidad de la clave del cliente. Para ejecutar un clúster con un solo HSM, debe deshabilitar la configuración de durabilidad de la clave de cliente.

PKCS #11 library

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Linux

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 --disable-key-availability-check
```

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Windows

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-pkcs11.exe" --disable-key-
availability-check
```

OpenSSL Dynamic Engine

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Linux

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
$ sudo /opt/cloudhsm/bin/configure-dyn --disable-key-availability-check
```

JCE provider

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Linux

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
$ sudo /opt/cloudhsm/bin/configure-jce --disable-key-availability-check
```

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Windows

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-jce.exe" --disable-key-
availability-check
```

CloudHSM CLI

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Linux

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
$ sudo /opt/cloudhsm/bin/configure-cli --disable-key-availability-check
```

Cómo deshabilitar la durabilidad de la clave de cliente para SDK 5 de cliente en Windows

- Use la herramienta de configuración para deshabilitar los ajustes de durabilidad de las claves de cliente.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-cli.exe" --disable-key-
availability-check
```

Para obtener más información sobre el parámetro `--disable-key-availability-check`, consulte the section called “Parámetros”.

Administración de las opciones de registro

Example

SDK 5 de cliente usa los parámetros `log-file`, `log-level`, `log-rotation` y `log-type` para administrar el registro.

> **Note**
>
> Para configurar el SDK para entornos sin servidor, como AWS Fargate o AWS Lambda, le recomendamos que configure AWS CloudHSM el tipo de registro en. `term` Los registros del cliente se enviarán al grupo de CloudWatch registros configurado para ese entorno `stderr` y se capturarán en él.

PKCS #11 library

Localización de registros predeterminada

- Si no especifica una ubicación para el archivo, el sistema escribirá los registros en la ubicación predeterminada siguiente:

  Linux

  ```
  /opt/cloudhsm/run/cloudhsm-pkcs11.log
  ```

  Windows

  ```
  C:\Program Files\Amazon\CloudHSM\cloudhsm-pkcs11.log
  ```

Cómo configurar el nivel de registro y dejar el resto de opciones de registro configuradas de forma predeterminada

- ```
  $ sudo /opt/cloudhsm/bin/configure-pkcs11 --log-level info
  ```

Cómo configurar las opciones de registro del archivo

- ```
  $ sudo /opt/cloudhsm/bin/configure-pkcs11 --log-type file --log-file <file name
   with path> --log-rotation daily --log-level info
  ```

Cómo configurar las opciones de registro del terminal

- ```
  $ sudo /opt/cloudhsm/bin/configure-pkcs11 --log-type term --log-level info
  ```

OpenSSL Dynamic Engine

Localización de registros predeterminada

- Si no especifica una ubicación para el archivo, el sistema escribirá los registros en la ubicación predeterminada siguiente:
  ```

Linux

```
stderr
```

Cómo configurar el nivel de registro y dejar el resto de opciones de registro configuradas de forma predeterminada

- ```
  $ sudo /opt/cloudhsm/bin/configure-dyn --log-level info
  ```

Cómo configurar las opciones de registro del archivo

- ```
  $ sudo /opt/cloudhsm/bin/configure-dyn --log-type <file name> --log-file file --
  log-rotation daily --log-level info
  ```

Cómo configurar las opciones de registro del terminal

- ```
  $ sudo /opt/cloudhsm/bin/configure-dyn --log-type term --log-level info
  ```

JCE provider

Localización de registros predeterminada

- Si no especifica una ubicación para el archivo, el sistema escribirá los registros en la ubicación predeterminada siguiente:

  Linux

  ```
  /opt/cloudhsm/run/cloudhsm-jce.log
  ```

  Windows

  ```
  C:\Program Files\Amazon\CloudHSM\cloudhsm-jce.log
  ```
  ```

Cómo configurar el nivel de registro y dejar el resto de opciones de registro configuradas de forma predeterminada

- ```
  $ sudo /opt/cloudhsm/bin/configure-jce --log-level info
  ```

Cómo configurar las opciones de registro del archivo

- ```
  $ sudo /opt/cloudhsm/bin/configure-jce --log-type file --log-file <file name> --log-rotation daily --log-level info
  ```

Cómo configurar las opciones de registro del terminal

- ```
  $ sudo /opt/cloudhsm/bin/configure-jce --log-type term --log-level info
  ```

CloudHSM CLI

Localización de registros predeterminada

- Si no especifica una ubicación para el archivo, el sistema escribirá los registros en la ubicación predeterminada siguiente:

  Linux

  ```
  /opt/cloudhsm/run/cloudhsm-cli.log
  ```

  Windows

  ```
  C:\Program Files\Amazon\CloudHSM\cloudhsm-cli.log
  ```

Cómo configurar el nivel de registro y dejar el resto de opciones de registro configuradas de forma predeterminada

- ```
  $ sudo /opt/cloudhsm/bin/configure-cli --log-level info
  ```

Cómo configurar las opciones de registro del archivo

- ```
  $ sudo /opt/cloudhsm/bin/configure-cli --log-type file --log-file <file name> --log-rotation daily --log-level info
  ```

Cómo configurar las opciones de registro del terminal

- ```
  $ sudo /opt/cloudhsm/bin/configure-cli --log-type term --log-level info
  ```

Para obtener más información acerca de los parámetros `log-file`, `log-level`, `log-rotation` y `log-type` consulte the section called “Parámetros”.

Coloque el certificado de emisión de SDK 5 de cliente

Example

En este ejemplo, se utiliza el parámetro `--hsm-ca-cert` para actualizar la ubicación del certificado de emisión de SDK 5 de cliente.

PKCS #11 library

Cómo ubicar el certificado de emisión en Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

  ```
  $ sudo /opt/cloudhsm/bin/configure-pkcs11 --hsm-ca-cert <customerCA certificate file>
  ```

Cómo ubicar el certificado de emisión en Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
"C:\Program Files\Amazon\CloudHSM\configure-pkcs11.exe" --hsm-ca-
cert <customerCA certificate file>
```

OpenSSL Dynamic Engine

Cómo ubicar el certificado de emisión en Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
$ sudo /opt/cloudhsm/bin/configure-dyn --hsm-ca-cert <customerCA certificate
 file>
```

JCE provider

Cómo ubicar el certificado de emisión en Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
$ sudo /opt/cloudhsm/bin/configure-jce --hsm-ca-cert <customerCA certificate
 file>
```

Cómo ubicar el certificado de emisión en Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
"C:\Program Files\Amazon\CloudHSM\configure-jce.exe" --hsm-ca-cert <customerCA
 certificate file>
```

CloudHSM CLI

Cómo ubicar el certificado de emisión en Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
$ sudo /opt/cloudhsm/bin/configure-cli --hsm-ca-cert <customerCA certificate
 file>
```

Cómo ubicar el certificado de emisión en Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar una ubicación para el certificado de emisión.

```
"C:\Program Files\Amazon\CloudHSM\configure-cli.exe" --hsm-ca-cert <customerCA
 certificate file>
```

Para obtener más información sobre el parámetro `--hsm-ca-cert`, consulte the section called “Parámetros”.

Parámetros

-a ***<ENI IP address>***

Agregue la dirección IP especificada a los archivos de configuración de SDK 5 de cliente. Introduzca cualquier dirección IP de ENI de un HSM del clúster. Para obtener más información acerca de cómo usar esta opción, consulte Iniciar SDK 5 de cliente.

Obligatorio: sí

-- hsm-ca-cert <customerCA certificate file path>

Ruta al directorio que almacena el certificado de la entidad de certificación (CA) que se utiliza para conectar las instancias de cliente de EC2 al clúster. Este archivo se crea al inicializar el clúster. De forma predeterminada, el sistema busca este archivo en la siguiente ubicación:

Linux

```
/opt/cloudhsm/etc/customerCA.crt
```

Windows

```
C:\ProgramData\Amazon\CloudHSM\customerCA.crt
```

Para obtener más información sobre la inicialización del clúster o la colocación del certificado, consulte ??? y???.

Obligatorio: no

--cluster-id <cluster ID>

Realiza una llamada `DescribeClusters` para buscar todas las direcciones IP de la interfaz de red elástica (ENI) de HSM en el clúster asociado al ID del clúster. El sistema añade las direcciones IP del ENI a los archivos AWS CloudHSM de configuración.

> **Note**
>
> Si utiliza el `--cluster-id` parámetro de una instancia EC2 dentro de una VPC que no tiene acceso a la Internet pública, debe crear un punto final de la VPC de interfaz al que conectarse. AWS CloudHSM Para obtener más información acerca de los puntos de conexión de VPC, consulte ???.

Obligatorio: no

--punto de conexión ***<endpoint>***

Especifique el punto final AWS CloudHSM de la API utilizado para realizar la llamada. `DescribeClusters` Debe configurar esta opción en combinación con `--cluster-id`.

Obligatorio: no

--region ***<region>***

Especifique la región de su clúster. Debe configurar esta opción en combinación con `--cluster-id`.

Si no proporciona el parámetro `--region`, el sistema elige la región intentando leer las variables de entorno `AWS_DEFAULT_REGION` o `AWS_REGION`. Si esas variables no están configuradas, el sistema comprueba la región asociada a su perfil en el archivo AWS Config (normalmente `~/.aws/config`), a menos que haya especificado un archivo diferente en la variable de entorno de `AWS_CONFIG_FILE`. Si no se establece ninguna de las opciones anteriores, el sistema utilizará la región `us-east-1` de forma predeterminada.

Obligatorio: no

-- server-client-cert-file <client certificate file path>

Ruta al certificado de cliente utilizado para la autenticación mutua de TLS cliente-servidor.

Utilice esta opción únicamente si no desea utilizar la clave y el certificado SSL/TLS predeterminados que incluimos en SDK 5 de cliente. Debe configurar esta opción en combinación con `--server-client-key-file`.

Obligatorio: no

-- server-client-key-file <client key file path>

Ruta a la clave de cliente utilizada para la autenticación mutua entre cliente y servidor con TLS.

Utilice esta opción únicamente si no desea utilizar la clave y el certificado SSL/TLS predeterminados que incluimos en SDK 5 de cliente. Debe configurar esta opción en combinación con `--server-client-cert-file`.

Obligatorio: no

--log-level ***<error | warn | info | debug | trace>***

Especifica el nivel de registro mínimo que el sistema debe escribir en el archivo de registro. Cada nivel incluye los niveles anteriores, con el error como nivel mínimo y el seguimiento como nivel máximo. Esto significa que si especifica errores, el sistema solo escribirá los errores en el registro. Si especifica el seguimiento, el sistema escribe los errores, las advertencias y los mensajes informativos (información) y de depuración en el registro. Para obtener más información, consulte Registro de Cliente SDK 5.

Obligatorio: no

--log-rotation ***<daily | weekly>***

Especifica la frecuencia con la que el sistema rota los registros. Para obtener más información, consulte Registro de Cliente SDK 5.

Obligatorio: no

--log-file ***<file name with path>***

Especifica dónde escribirá el sistema el archivo de registro. Para obtener más información, consulte Registro de Cliente SDK 5.

Obligatorio: no

--log-type ***<term | file>***

Especifica si el sistema escribirá el registro en un archivo o terminal. Para obtener más información, consulte Registro de Cliente SDK 5.

Obligatorio: no

-h | --help

Muestra ayuda.

Obligatorio: no

-v | --versión

Muestra la versión.

Obligatorio: no

--disable-key-availability-check

Marcador para deshabilitar el cuórum de disponibilidad de claves. Use este indicador para indicar que se AWS CloudHSM debe deshabilitar el quórum de disponibilidad de claves y puede usar claves que solo existan en un HSM del clúster. Para obtener más información sobre el uso de este marcador para establecer el cuórum de disponibilidad de claves, consulte ???.

Obligatorio: no

--enable-key-availability-check

Marcador para habilitar el cuórum de disponibilidad de claves. Use este indicador para indicar que AWS CloudHSM debe utilizarse el quórum de disponibilidad de claves y no permitirle usar claves hasta que esas claves estén en dos HSM del clúster. Para obtener más información sobre el uso de este marcador para establecer el cuórum de disponibilidad de claves, consulte ???.

Está habilitado de forma predeterminada.

Obligatorio: no

-- -init disable-validate-key-at

Mejora el rendimiento al especificar que puede omitir una llamada de inicialización para comprobar los permisos de una clave en llamadas posteriores. Utilice esta opción con precaución.

Antecedentes: algunos mecanismos de la biblioteca PKCS #11 admiten operaciones de varias partes, en las que una llamada de inicialización verifica si se puede utilizar la clave para llamadas posteriores. Esto requiere una llamada de verificación al HSM, lo que añade latencia a la operación general. Esta opción le permite deshabilitar la llamada posterior y, potencialmente, mejorar el rendimiento.

Obligatorio: no

-- -inicio enable-validate-key-at

Especifica que debe usar una llamada de inicialización para verificar los permisos de una clave para las llamadas posteriores. Esta es la opción predeterminada. Utilice `enable-validate-key-at-init` para reanudar estas llamadas de inicialización después de utilizar `disable-validate-key-at-init` para suspenderlas.

Obligatorio: no

Temas relacionados de

- DescribeClusters Operación de la API
- describe-clusters CLI de AWS
- Get-HSM2Cluster PowerShell cmdlet
- Inicio de SDK 5 de cliente.
- AWS CloudHSM Puntos de conexión de VPC
- Administrar la configuración de durabilidad de la clave de SDK 5 de cliente
- Registro de SDK 5 de cliente

Configuraciones avanzadas para la herramienta de configuración SDK 5 de cliente

La herramienta de configuración SDK 5 de cliente incluye configuraciones avanzadas que no forman parte de las características generales que utilizan la mayoría de los clientes. Las configuraciones avanzadas proporcionan capacidades adicionales.

- Configuraciones avanzadas para PKCS #11

- Conexión a varias ranuras con PKCS#11
- Comandos de reintento para PKCS #11
- Configuraciones avanzadas para JCE
  - Conexión a varios clústeres con el proveedor de JCE
  - Comandos de reintento para JCE
  - Extracción de claves mediante JCE
- Configuraciones avanzadas para OpenSSL
  - Comandos de reintento para OpenSSL
- Configuraciones avanzadas para la interfaz de línea de comandos (CLI)
  - Conexión a varios clústeres con CLI

Herramienta de configuración de SDK 3 de cliente

Utilice la herramienta de configuración SDK 3 de cliente para arrancar el daemon del cliente y configurar la Utilidad de administración de CloudHSM.

Sintaxis

```
configure -h | --help
          -a <ENI IP address>
          -m [-i <daemon_id>]
          --ssl --pkey <private key file> --cert <certificate file>
          --cmu <ENI IP address>
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza la herramienta configure.

Example : Actualice los datos del HSM del cliente y de AWS CloudHSM key_mgmt_util

En este ejemplo, se utiliza el `-a` parámetro de configure para actualizar los datos del HSM del cliente y de key_mgmt_util. AWS CloudHSM Para usar el parámetro `-a`, debe tener la dirección IP de uno de los HSM de su clúster. Utilice la consola o la CLI de AWS para obtener la dirección IP.

Para obtener una dirección IP para un HSM (consola)

1. Abra la AWS CloudHSM consola en https://console.aws.amazon.com/cloudhsm/home.

2. Para cambiar la región de AWS, utilice el selector de regiones en la esquina superior derecha de la página.
3. Para abrir la página de detalles del clúster, en la tabla de clústeres, elija el ID del clúster.
4. Para obtener la dirección IP, vaya a la pestaña HSM y elija una de las direcciones IP que aparecen en la lista Dirección IP de ENI.

Para obtener una dirección IP para un HSM (CLI)

- Obtenga la dirección IP de un HSM mediante el describe-clusters comando de la CLI. En el resultado del comando, la dirección IP de los HSM son los valores de `EniIp`.

```
$  aws cloudhsmv2 describe-clusters


{
    "Clusters": [
        { ... }
            "Hsms": [
                {
...
                    "EniIp": "10.0.0.9",
...
                },
                {
...
                    "EniIp": "10.0.1.6",
...
```

Cómo actualizar los datos del HSM

1. Antes de actualizar el `-a` parámetro, detenga el AWS CloudHSM cliente. De esta forma, evitará los conflictos que podrían producirse mientras configure edita el archivo de configuración del cliente. Si el cliente ya se ha detenido, este comando no tiene ningún efecto, por lo que puede utilizarlo en un script.

   Amazon Linux

   ```
   $ sudo stop cloudhsm-client
   ```

Amazon Linux 2

```
$ sudo service cloudhsm-client stop
```

CentOS 7

```
$ sudo service cloudhsm-client stop
```

CentOS 8

```
$ sudo service cloudhsm-client stop
```

RHEL 7

```
$ sudo service cloudhsm-client stop
```

RHEL 8

```
$ sudo service cloudhsm-client stop
```

Ubuntu 16.04 LTS

```
$ sudo service cloudhsm-client stop
```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client stop
```

Windows

- Para la versión 1.1.2 y posteriores del cliente de Windows:

  ```
  C:\Program Files\Amazon\CloudHSM>net.exe stop AWSCloudHSMClient
  ```

- Para la versión 1.1.1 y anteriores de clientes de Windows:

  Use Ctrl + C en la ventana de comandos donde inició el AWS CloudHSM cliente.

2. Este paso utiliza el parámetro `-a` de configure para añadir la dirección IP de la ENI `10.0.0.9` a los archivos de configuración.

   Amazon Linux

   ```
   $ sudo /opt/cloudhsm/bin/configure -a 10.0.0.9
   ```

   Amazon Linux 2

   ```
   $ sudo /opt/cloudhsm/bin/configure -a 10.0.0.9
   ```

   CentOS 7

   ```
   $ sudo /opt/cloudhsm/bin/configure -a 10.0.0.9
   ```

   CentOS 8

   ```
   $ sudo /opt/cloudhsm/bin/configure -a 10.0.0.9
   ```

   RHEL 7

   ```
   $ sudo /opt/cloudhsm/bin/configure -a 10.0.0.9
   ```

   RHEL 8

   ```
   $ sudo /opt/cloudhsm/bin/configure -a 10.0.0.9
   ```

   Ubuntu 16.04 LTS

   ```
   $ sudo /opt/cloudhsm/bin/configure -a 10.0.0.9
   ```

   Ubuntu 18.04 LTS

   ```
   $ sudo /opt/cloudhsm/bin/configure -a 10.0.0.9
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\ configure.exe -a 10.0.0.9
   ```

3. A continuación, reinicie el AWS CloudHSM cliente. Cuando el cliente de se inicia, utiliza la dirección IP de ENI del archivo de configuración para realizar consultas en el clúster. A continuación, escribe las direcciones IP de ENI de todos los HSM del clúster en el archivo `cluster.info`.

   Amazon Linux

   ```
   $ sudo start cloudhsm-client
   ```

   Amazon Linux 2

   ```
   $ sudo service cloudhsm-client start
   ```

   CentOS 7

   ```
   $ sudo service cloudhsm-client start
   ```

   CentOS 8

   ```
   $ sudo service cloudhsm-client start
   ```

   RHEL 7

   ```
   $ sudo service cloudhsm-client start
   ```

   RHEL 8

   ```
   $ sudo service cloudhsm-client start
   ```

   Ubuntu 16.04 LTS

   ```
   $ sudo service cloudhsm-client start
   ```

   Ubuntu 18.04 LTS

   ```
   $ sudo service cloudhsm-client start
   ```

Windows

- Para la versión 1.1.2 y posteriores del cliente de Windows:

```
C:\Program Files\Amazon\CloudHSM>net.exe start AWSCloudHSMClient
```

- Para la versión 1.1.1 y anteriores de clientes de Windows:

```
C:\Program Files\Amazon\CloudHSM>start "cloudhsm_client" cloudhsm_client.exe
 C:\ProgramData\Amazon\CloudHSM\data\cloudhsm_client.cfg
```

Cuando se complete el comando, los datos del HSM que utilizan el AWS CloudHSM cliente y key_mgmt_util estarán completos y serán precisos.

Example : actualización de los datos de HSM para la CMU desde la versión 3.2.1 y anteriores del SDK del cliente

En este ejemplo, se utiliza el comando `-m`configure para copiar los datos actualizados de los HSM del archivo `cluster.info` en el archivo `cloudhsm_mgmt_util.cfg` que utiliza cloudhsm_mgmt_util. Úselo con la CMU que se incluye con la versión 3.2.1 y anteriores del SDK del cliente.

- Antes de ejecutar el`-m`, detenga el AWS CloudHSM cliente, ejecute el `-a` comando y, a continuación, reinícielo, como se muestra en el AWS CloudHSM ejemplo anterior. Esto garantiza que los datos que se copian en el archivo `cloudhsm_mgmt_util.cfg` desde el archivo `cluster.info` sean completos y precisos.

  Linux

  ```
  $ sudo /opt/cloudhsm/bin/configure -m
  ```

  Windows

  ```
  C:\Program Files\Amazon\CloudHSM\bin\ configure.exe -m
  ```

Example : actualización de los datos de HSM para la CMU desde la versión 3.3.0 y posteriores del SDK del cliente

Este ejemplo utiliza el parámetro `--cmu` del comando configure para actualizar los datos HSM para CMU. Úselo con la CMU que se incluye con la versión 3.3.0 y posteriores del SDK del cliente. Para obtener más información sobre el uso de la CMU, consulte Usar la Utilidad de administración de CloudHSM (CMU) para administrar usuarios y Usar la CMU con la versión 3.2.1. y anteriores del SDK del cliente.

- Utilice el parámetro `--cmu` para transferir la dirección IP de un HSM de su clúster.

  Linux

  ```
  $ sudo /opt/cloudhsm/bin/configure --cmu <IP address>
  ```

  Windows

  ```
  C:\Program Files\Amazon\CloudHSM\bin\ configure.exe --cmu <IP address>
  ```

Parámetros

-h | --help

Muestra la sintaxis del comando.

Obligatorio: sí

-a *<ENI IP address>*

Agrega la dirección IP de la interfaz de red elástica (ENI) del HSM especificado a los archivos de configuración de AWS CloudHSM . Escriba la dirección IP de ENI de cualquiera de los HSM del clúster. No importa qué HSM seleccione.

Para obtener las direcciones IP ENI de los HSM del clúster, utilice la DescribeClustersoperación, el comando CLI describe-clusters o el cmdlet. Get-HSM2Cluster PowerShell

> **Note**
>
> Antes de ejecutar el comando, detenga el cliente `-a`configure. AWS CloudHSM A continuación, cuando se complete el `-a` comando, reinicie el AWS CloudHSM cliente. Para obtener información detallada, consulte los ejemplos.

Este parámetro edita los archivos de configuración siguientes:

- `/opt/cloudhsm/etc/cloudhsm_client.cfg`: Utilizado por el AWS CloudHSM cliente y key_mgmt_util.
- `/opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg`: Utilizado por cloudhsm_mgmt_util.

Cuando el AWS CloudHSM cliente se inicia, utiliza la dirección IP de ENI en su archivo de configuración para consultar el clúster y actualizar el `cluster.info` archivo (`/opt/cloudhsm/daemon/1/cluster.info`) con las direcciones IP de ENI correctas para todos los HSM del clúster.

Obligatorio: sí

-m

Actualiza las direcciones IP de ENI de HSM que figuran en el archivo de configuración que CMU utiliza.

> **Note**
>
> El parámetro `-m` se utiliza con la CMU de la versión 3.2.1 y anteriores del SDK del cliente. Para ver la CMU de la versión 3.3.0 y posteriores del SDK del cliente, consulte el parámetro `--cmu`, que simplifica el proceso de actualización de los datos del HSM para la CMU.

Al actualizar el `-a` parámetro del AWS CloudHSM cliente configure y, a continuación, iniciarlo, el daemon del cliente consulta el clúster y actualiza los `cluster.info` archivos con las direcciones IP de los HSM correctas para todos los HSM del clúster. Al ejecutar el comando configure `-m`, se completa la actualización copiando las direcciones IP de los HSM del archivo `cluster.info` en el archivo de configuración `cloudhsm_mgmt_util.cfg` que utiliza cloudhsm_mgmt_util.

Asegúrese de ejecutar el `-a` configure comando y reiniciar el AWS CloudHSM cliente antes de ejecutar el comando. `-m` De esta forma se asegura de que los datos que se copien en el archivo `cloudhsm_mgmt_util.cfg` desde el archivo `cluster.info` sean completos y precisos.

Obligatorio: sí

-i

Especifica un daemon del cliente alternativo. El valor predeterminado representa el cliente de AWS CloudHSM .

Valor predeterminado: 1

Obligatorio: no

--ssl

Sustituye el certificado y la clave SSL del clúster por la clave privada y el certificado especificados. Cuando se utiliza este parámetro, los parámetros `--pkey` y `--cert` son obligatorios.

Obligatorio: no

--pkey

Especifica la clave privada nueva. Introduzca la ruta y el nombre del archivo que contiene la clave privada.

Obligatorio: sí, en caso de que se especifique --ssl. De lo contrario, no debe usarse.

--cert

Especifica el certificado nuevo. Introduzca la ruta y el nombre del archivo que contiene el certificado. El certificado debe encadenarse hasta el certificado `customerCA.crt`, el certificado autofirmado que se utiliza para inicializar el clúster. Para obtener más información, consulte Inicializar el clúster.

Obligatorio: sí, en caso de que se especifique --ssl. De lo contrario, no debe usarse.

--cmu ***<ENI IP address>***

Combina los parámetros `-a` y `-m` en un solo parámetro. Agrega la dirección IP de la interfaz elastic network interface (ENI) de HSM especificada a los archivos de AWS CloudHSM configuración y, a continuación, actualiza el archivo de configuración de la CMU. Introduzca una

dirección IP de cualquier HSM en el clúster. Para la versión 3.2.1 y anteriores del SDK de cliente, consulte Usar la CMU con la versión 3.2.1 y anteriores de ClientSDK.

Obligatorio: sí

Temas relacionados de

- Configurar key_mgmt_util

Interfaz de la línea de comandos (CLI) de CloudHSM

La CLI de CloudHSM ayuda a los administradores a gestionar los usuarios y a los usuarios de criptomonedas a gestionar las claves de su clúster. Incluye herramientas que se pueden usar para crear, eliminar y enumerar usuarios, cambiar las contraseñas de los usuarios y actualizar la autenticación multifactor (MFA) de los usuarios. También incluye comandos que generan, eliminan, importan y exportan claves, obtienen y establecen atributos, buscan claves y realizan operaciones criptográficas.

Para obtener una lista definida de los usuarios de la CLI de CloudHSM, consulte Administración de usuarios de HSMcon la CLI de CloudHSM . Para obtener una lista definida de los atributos clave de la CLI de CloudHSM, consulte. Atributos de clave de la CLI de CloudHSM Para obtener información sobre cómo usar la CLI de CloudHSM para administrar las claves, consulte. Administración de claves con la CLI de CloudHSM

Para un inicio rápido, consulte Introducción a la interfaz de la línea de comandos (CLI) de CloudHSM. Para obtener información detallada y ejemplos de uso de los comandos CLI de CloudHSM, consulte Referencia para los comandos de la CLI de CloudHSM.

Temas

- Plataformas compatibles con la interfaz de la línea de comandos (CLI) de CloudHSM
- Introducción a la interfaz de la línea de comandos (CLI) de CloudHSM
- Modos interactivo y de comando único
- Atributos de clave de la CLI de CloudHSM
- Migre de Client SDK 3 CMU y KMU a Client SDK 5 CloudHSM CLI
- Configuraciones avanzadas para CLI
- Referencia para los comandos de la CLI de CloudHSM

Plataformas compatibles con la interfaz de la línea de comandos (CLI) de CloudHSM

Compatibilidad con Linux

Plataformas admitidas	Arquitectura X86_64	Arquitectura ARM
Amazon Linux 2	Sí	Sí
Amazon Linux 2023	Sí	Sí
CentOS 7 (7,8+)	Sí	No
Red Hat Enterprise Linux 7 (7.8+)	Sí	No
Red Hat Enterprise Linux 8 (8.3 o superior)	Sí	No
Red Hat Enterprise Linux 9 (9.2+)	Sí	Sí
Ubuntu 20.04 LTS	Sí	No
Ubuntu 22.04 LTS	Sí	Sí

Nota: El SDK 5.4.2 fue la última versión que proporcionó soporte para la plataforma CentOS 8. Para obtener más información, consulte el sitio web de CentOS.

Compatibilidad con Windows

- Microsoft Windows Server 2016
- Microsoft Windows Server 2019

Introducción a la interfaz de la línea de comandos (CLI) de CloudHSM

La interfaz de línea de comandos (CLI) de CloudHSM le permite administrar los usuarios de su clúster. AWS CloudHSM Utilice este tema para empezar con las tareas básicas de administración de usuarios del HSM, como la creación de usuarios, la lista de usuarios y la conexión de la CLI de CloudHSM al clúster.

Instalación de la CLI de CloudHSM

Utilice los comandos siguientes para descargar e instalar la CLI de CloudHSM.

Amazon Linux 2

Amazon Linux 2 en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-cli-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el7.x86_64.rpm
```

Amazon Linux 2 en arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-cli-latest.el7.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el7.aarch64.rpm
```

Amazon Linux 2023

Amazon Linux 2023 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/cloudhsm-cli-latest.amzn2023.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.amzn2023.x86_64.rpm
```

Amazon Linux 2023 en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/
cloudhsm-cli-latest.amzn2023.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.amzn2023.aarch64.rpm
```

CentOS 7 (7.8+)

CentOS 7 en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-cli-
latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el7.x86_64.rpm
```

RHEL 7 (7.8+)

RHEL 7 en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-cli-
latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el7.x86_64.rpm
```

RHEL 8 (8.3+)

RHEL 8 en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-cli-
latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el8.x86_64.rpm
```

RHEL 9 (9.2+)

RHEL 9 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-cli-
latest.el9.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el9.x86_64.rpm
```

RHEL 9 en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-cli-latest.el9.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-cli-latest.el9.aarch64.rpm
```

Ubuntu 20.04 LTS

Ubuntu 20.04 LTS en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Focal/cloudhsm-cli_latest_u20.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-cli_latest_u20.04_amd64.deb
```

Ubuntu 22.04 LTS

Ubuntu 22.04 LTS en arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/cloudhsm-cli_latest_u22.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-cli_latest_u22.04_amd64.deb
```

Ubuntu 22.04 LTS en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/cloudhsm-cli_latest_u22.04_arm64.deb
```

```
$ sudo apt install ./cloudhsm-cli_latest_u22.04_arm64.deb
```

Windows Server 2016

Para Windows Server 2016 con arquitectura x86_64, ábralo PowerShell como administrador y ejecute el siguiente comando:

```
PS C:\> wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Windows/
AWSCloudHSMCLI-latest.msi -Outfile C:\AWSCloudHSMCLI-latest.msi
```

```
PS C:\> Start-Process msiexec.exe -ArgumentList '/i C:\AWSCloudHSMCLI-latest.msi /
quiet /norestart /log C:\client-install.txt' -Wait
```

Windows Server 2019

Para Windows Server 2019 con una arquitectura x86_64, ábralo PowerShell como administrador y ejecute el siguiente comando:

```
PS C:\> wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Windows/
AWSCloudHSMCLI-latest.msi -Outfile C:\AWSCloudHSMCLI-latest.msi
```

```
PS C:\> Start-Process msiexec.exe -ArgumentList '/i C:\AWSCloudHSMCLI-latest.msi /
quiet /norestart /log C:\client-install.txt' -Wait
```

Use los siguientes comandos para configurar la CLI de CloudHSM.

Cómo iniciar el proceso de arranque de una instancia EC2 de Linux para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de los HSM de su clúster.

```
$ sudo /opt/cloudhsm/bin/configure-cli -a <The ENI IP addresses of the HSMs>
```

Arranque de una instancia EC2 de Windows para SDK 5 de cliente

- Utilice la herramienta de configuración para especificar la dirección IP de los HSM de su clúster.

```
"C:\Program Files\Amazon\CloudHSM\bin\configure-cli.exe" -a <The ENI IP addresses
 of the HSMs>
```

Utilización de la CLI de CloudHSM

1. Use el siguiente comando para iniciar la CLI de CloudHSM.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm-cli interactive
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
   ```

2. Utilice el comando login para iniciar sesión en el clúster. Todos los usuarios pueden usar este comando.

   El comando del siguiente ejemplo inicia sesión en admin, que es la cuenta de administrador por defecto. La contraseña de este usuario se establece al activar el clúster.

   ```
   aws-cloudhsm > login --username admin --role admin
   ```

   El sistema le solicitará su contraseña. Introduzca la contraseña y la salida mostrará que el comando se ha ejecutado correctamente.

   ```
   Enter password:
   {
     "error_code": 0,
     "data": {
       "username": "admin",
       "role": "admin"
     }
   }
   ```

3. Ejecute el comando user list para listar todos los usuarios del clúster.

   ```
   aws-cloudhsm > user list
   {
     "error_code": 0,
     "data": {
       "users": [
         {
           "username": "admin",
   ```

```
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      },
      {
        "username": "app_user",
        "role": "internal(APPLIANCE_USER)",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      }
    ]
  }
}
```

4. Utilice user create para crear un usuario de CU denominado **example_user**.

   Puede crear CU porque en un paso anterior inició sesión como usuario administrador. Solo los usuarios administradores pueden realizar tareas de administración de usuarios, como crear y eliminar usuarios y cambiar las contraseñas de otros usuarios.

```
aws-cloudhsm > user create --username example_user --role crypto-user
Enter password:
Confirm password:
{
 "error_code": 0,
 "data": {
   "username": "example_user",
   "role": "crypto-user"
 }
}
```

5. Ejecute user list para listar todos los usuarios del clúster.

```
aws-cloudhsm > user list
{
  "error_code": 0,
  "data": {
    "users": [
      {
        "username": "admin",
```

```
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      },
      {
        "username": "example_user",
        "role": "crypto_user",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      },
      {
        "username": "app_user",
        "role": "internal(APPLIANCE_USER)",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      }
    ]
  }
}
```

6. Use el logout comando para cerrar sesión en el clúster. AWS CloudHSM

```
aws-cloudhsm > logout
{
  "error_code": 0,
  "data": "Logout successful"
}
```

7. Utilice la CLI para ejecutar el comando quit.

```
aws-cloudhsm > quit
```

Modos interactivo y de comando único

En la CLI de CloudHSM, puede ejecutar comandos de dos maneras diferentes: en modo de comando único y en modo interactivo. El modo interactivo está diseñado para los usuarios y el modo de comando único está diseñado para los scripts.

> **Note**
>
> Todos los comandos funcionan en modo interactivo y en modo de comando único.

Modo interactivo

Use los siguientes comandos para iniciar el modo interactivo de la CLI de CloudHSM.

Linux

```
$ /opt/cloudhsm/bin/cloudhsm-cli interactive
```

Windows

```
C:\Program Files\Amazon\CloudHSM\bin\> .\cloudhsm-cli.exe interactive
```

Al utilizar la CLI en modo interactivo, puede iniciar sesión en una cuenta de usuario mediante el comando login.

Para mostrar una lista de todos los comandos de la CLI de CloudHSM, ejecute el comando siguiente:

```
aws-cloudhsm > help
```

Para obtener la sintaxis de un comando de la CLI de CloudHSM, ejecute el siguiente comando:

```
aws-cloudhsm >  help <command-name>
```

Para obtener una lista de los usuarios de los HSM, introduzca user list.

```
aws-cloudhsm > user list
```

Para finalizar la sesión de la CLI de CloudHSM, ejecute el comando siguiente:

```
aws-cloudhsm > quit
```

Modo de comando único

Si ejecuta la CLI de CloudHSM mediante el modo de comando único, debe configurar dos variables de entorno para proporcionar las credenciales: CLOUDHSM_PIN y CLOUDHSM_ROLE:

```
$ export CLOUDHSM_ROLE=admin
```

```
$ export CLOUDHSM_PIN=admin_username:admin_password
```

Después de hacer esto, puede ejecutar comandos con las credenciales almacenadas en su entorno.

```
$ cloudhsm-cli user change-password --username alice --role crypto-user
Enter password:
Confirm password:
{
    "error_code": 0,
    "data": {
      "username": "alice",
      "role": "crypto-user"
    }
}
```

Atributos de clave de la CLI de CloudHSM

En este tema se describe cómo utilizar la CLI de CloudHSM para configurar atributos de clave. Un atributo de clave de la CLI de CloudHSM puede definir el tipo de clave, cómo puede funcionar o cómo se etiqueta una clave. Algunos atributos definen características únicas (por ejemplo, el tipo de clave). Otros atributos se pueden configurar como true o false; al cambiarlos, se activa o desactiva una parte de la funcionalidad de la clave.

Para ver ejemplos que muestran cómo usar los atributos de clave, consulte los comandos que aparecen debajo del comando principal key.

Atributos admitidos

Es recomendable que solamente establezca valores para los atributos que desee hacer más restrictivos. Si no se especifica ningún valor, la CLI de CloudHSM utilizará el valor predeterminado que se indica en la tabla siguiente.

En la siguiente tabla se enumeran los atributos de clave, los valores posibles, los valores predeterminados y las notas relacionadas. Una celda vacía en la columna Valor indica que no hay ningún valor predeterminado específico asignado al atributo.

Atributo de la CLI de CloudHSM	Valor	Modificable con un conjunto de claves y atributos	Configurable en el momento de la creación de la clave
`always-sensitive`	El valor es `True` si `sensitive` siempre se ha establecido como `True` y nunca ha cambiado.	No	No
`check-value`	Valor de comprobación de la clave. Para obtener más información, consulte Detalles adicionales.	No	No
`class`	Valores posibles: `secret-key` , `public-key` y `private-key` .	No	Sí
`curve`	Curva elíptica utilizada para generar el par de claves de EC. Los valores aceptados son: `secp224r1` , `secp256r1` , `prime256v1` , `secp384r1` ,	No	Se puede configurar con RSA, no se puede configurar con EC.

Atributo de la CLI de CloudHSM	Valor	Modificable con un conjunto de claves y atributos	Configurable en el momento de la creación de la clave
	`secp256k1` y `secp521r1` .		
`decrypt`	Valor predeterminado: `False`	Sí	Sí
`derive`	Valor predeterminado: `False`	Sí	Sí
`destroyable`	Valor predeterminado: `True`	Sí	Sí
`ec-point`	Para las claves de EC, codificación DER del valor ECPoint ANSI X9.62 «Q» en formato hexadecimal. Para otros tipos de clave, este atributo no existe.	No	No
`encrypt`	Valor predeterminado: `False`	Sí	Sí
`extractable`	Valor predeterminado: `True`	No	Sí
`id`	Valor predeterminado: vacío	No	Sí

Atributo de la CLI de CloudHSM	Valor	Modificable con un conjunto de claves y atributos	Configurable en el momento de la creación de la clave
`key-length-bytes`	Necesaria para generar una clave de AES. Valores válidos: 16, 24 y 32 bytes.	No	No
`key-type`	Valores posibles: `aes`, `rsa` y `ec`	No	Sí
`label`	Valor predeterminado: vacío	Sí	Sí
`local`	Predeterminado: `True` para las claves generadas en el HSM, `False` para las claves importadas en el HSM.	No	No
`modifiable`	Valor predeterminado: `True`	No	No
`modulus`	El módulo que se utilizó para generar un par de claves RSA. Para otros tipos de clave, este atributo no existe.	No	No

Atributo de la CLI de CloudHSM	Valor	Modificable con un conjunto de claves y atributos	Configurable en el momento de la creación de la clave
`modulus-size-bits`	Necesario para generar un par de claves de RSA. El valor mínimo es `2048`.	No	Se puede configurar con RSA, no se puede configurar con EC.
`never-extractable`	El valor es `True` si nunca se ha establecido la opción extraíble como `False`. El valor es `False` si nunca se ha establecido la opción extraíble como `True`.	No	No
`private`	Valor predeterminado: `True`	No	Sí
`public-exponent`	Necesario para generar un par de claves de RSA. Valores válidos: el valor debe ser un número impar superior o igual a `65537`.	No	Se puede configurar con RSA, no se puede configurar con EC.

Atributo de la CLI de CloudHSM	Valor	Modificable con un conjunto de claves y atributos	Configurable en el momento de la creación de la clave
`sensitive`	Predeterminado: • El valor es `True` para las claves de AES y las claves privadas de EC y RSA. • El valor es `False` para las claves públicas de EC y RSA.	No	Se puede configurar con claves privadas, no se puede configurar con claves públicas.
`sign`	Predeterminado: • El valor es `True` para las claves de AES. • El valor es `False` para claves RSA y EC.	Sí	Sí
`token`	Valor predeterminado: `False`	No	Sí
`trusted`	Valor predeterminado: `False`	Sí	No
`unwrap`	Valor predeterminado: `False`	Sí	Sí

Atributo de la CLI de CloudHSM	Valor	Modificable con un conjunto de claves y atributos	Configurable en el momento de la creación de la clave
`unwrap-template`	Los valores deben usar la plantilla de atributo aplicada a cualquier clave desencapsulada mediante esta clave de encapsulamiento.	Sí	No
`verify`	Predeterminado: • El valor es `True` para las claves de AES. • El valor es `False` para claves RSA y EC.	Sí	Sí
`wrap`	Valor predeterminado: `False`	Sí	Sí
`wrap-template`	Los valores deben usar la plantilla de atributo para coincidir con la clave encapsulada usando esta clave de encapsulamiento.	Sí	No
`wrap-with-trusted`	Valor predeterminado: `False`	Sí	Sí

Detalles adicionales

Compruebe el valor.

El valor de comprobación (KCV) es un hash o suma de comprobación de 3 bytes de una clave que se genera cuando el HSM importa o genera una clave. También puede calcular un valor de comprobación fuera del HSM, por ejemplo, después de exportar una clave. A continuación, puede comparar los valores de comprobación para confirmar la identidad y la integridad de la clave. Para obtener el valor de comprobación de una clave, utilice la lista de claves con el marcador Verbose.

AWS CloudHSM utiliza los siguientes métodos estándar para generar un valor de comprobación:

- Claves simétricas: los primeros 3 bytes del resultado obtenido al cifrar un bloque cero con la clave.
- Pares de claves asimétricas: los primeros 3 bytes del hash SHA-1 de la clave pública.
- Claves HMAC: por el momento, no se admite el uso del KCV con claves HMAC.

Temas relacionados de

- key
- Referencia para los comandos de la CLI de CloudHSM

Migre de Client SDK 3 CMU y KMU a Client SDK 5 CloudHSM CLI

Utilice este tema para migrar los flujos de trabajo que utilizan las herramientas de línea de comandos de Client SDK 3, la utilidad de administración de CloudHSM (CMU) y la utilidad de administración de claves (KMU), para utilizar en su lugar la herramienta de línea de comandos de Client SDK 5, CloudHSM CLI.

En AWS CloudHSM, las aplicaciones de los clientes realizan operaciones criptográficas mediante el kit de desarrollo de software (SDK) de AWS CloudHSM cliente. El SDK de cliente 5 es el SDK principal al que se le siguen añadiendo nuevas funciones y compatibilidad con plataformas. En este tema se proporcionan detalles específicos sobre la migración del SDK de cliente 3 al SDK de cliente 5 para herramientas de línea de comandos.

El SDK de cliente 3 incluye dos herramientas de línea de comandos independientes: la CMU para administrar los usuarios y la KMU para administrar las claves y realizar operaciones con ellas. El

Client SDK 5 consolida las funciones de la CMU y la KMU (herramientas que se ofrecían con el Client SDK 3) en una sola herramienta, la. Interfaz de la línea de comandos (CLI) de CloudHSM Las operaciones de administración de usuarios se encuentran en los subcomandos y. usuario quorum Las operaciones de administración de claves se encuentran en el subcomando key, y las operaciones criptográficas, en el subcomando crypto. Consulte Referencia para los comandos de la CLI de CloudHSM para obtener una lista completa de comandos.

> **Note**
>
> Si en Client SDK 3 confiaba en syncKey una syncUser funcionalidad para la sincronización entre clústeres, siga utilizando la CMU. La CLI de CloudHSM en el SDK de cliente 5 no admite actualmente esta funcionalidad.

Para obtener instrucciones sobre cómo migrar al SDK de cliente 5, consulte. Migración del SDK 3 de cliente al SDK 5 de cliente Para obtener información sobre las ventajas de la migración, consulte. Ventajas del SDK 5 de cliente

Configuraciones avanzadas para CLI

La interfaz de línea de AWS CloudHSM comandos (CLI) incluye la siguiente configuración avanzada, que no forma parte de las configuraciones generales que utilizan la mayoría de los clientes. Estas configuraciones proporcionan capacidades adicionales.

- Conexión a múltiples clústeres

Conexión a varios clústeres con CLI

Con Client SDK 5, puede configurar la AWS CloudHSM CLI para permitir las conexiones a varios clústeres de CloudHSM desde una sola instancia de CLI.

Siga las instrucciones de este tema para utilizar la interfaz de línea de AWS CloudHSM comandos (CLI) y utilizar la funcionalidad de varios clústeres para conectarse con varios clústeres.

Temas

- Requisitos previos para varios clústeres
- Configurar la CLI para la funcionalidad de varios clústeres
- configure-cli add-cluster

- configure-cli remove-cluster
- Uso de varios clústeres

Requisitos previos para varios clústeres

- Dos o más AWS CloudHSM clústeres a los que te gustaría conectarte, junto con sus certificados de clúster.
- Una instancia de EC2 con grupos de seguridad configurados correctamente para conectarse a todos los clústeres anteriores. Para obtener más información sobre cómo configurar un clúster y la instancia de cliente, consulta Cómo empezar con AWS CloudHSM.
- Para configurar la funcionalidad de varios clústeres, ya debe haber descargado e instalado la AWS CloudHSM CLI. Si todavía no ha hecho esto, consulte las instrucciones en ???.
- No podrá acceder a un clúster configurado con, `./configure-cli[.exe] -a` ya que no estará asociado a un`cluster-id`. Puede volver a configurarlo siguiendo las instrucciones que `config-cli add-cluster` se describen en esta guía.

Configurar la CLI para la funcionalidad de varios clústeres

Para configurar la AWS CloudHSM CLI para la funcionalidad de varios clústeres, siga estos pasos:

1. Identifique los clústeres a los que desea conectarse.
2. Agregue estos clústeres a la configuración de AWS CloudHSM CLI mediante el subcomando configure-cli, tal y `add-cluster` como se describe a continuación.
3. Reinicie todos los procesos AWS CloudHSM CLI para que la nueva configuración surta efecto.

configure-cli add-cluster

Cuando se conecte a varios clústeres, utilice el `configure-cli add-cluster` comando para añadir un clúster a la configuración.

Sintaxis

```
configure-cli add-cluster [OPTIONS]
        --cluster-id <CLUSTER ID>
        [--region <REGION>]
        [--endpoint <ENDPOINT>]
        [--hsm-ca-cert <HSM CA CERTIFICATE FILE>]
```

```
        [--server-client-cert-file <CLIENT CERTIFICATE FILE>]
        [--server-client-key-file <CLIENT KEY FILE>]
        [-h, --help]
```

Ejemplos

Cómo agregar un clúster mediante el parámetro **`cluster-id`**

Example

Utilice el `configure-cli add-cluster` junto con el parámetro `cluster-id` para agregar un clúster (con el ID de `cluster-1234567`) a su configuración.

Linux

```
$ sudo /opt/cloudhsm/bin/configure-cli add-cluster --cluster-id cluster-1234567
```

Windows

```
C:\Program Files\Amazon\CloudHSM\> .\configure-cli.exe add-cluster --cluster-
id cluster-1234567
```

> **Tip**
>
> Si al utilizar `configure-cli add-cluster` con el parámetro `cluster-id` el clúster no se agrega, consulte el siguiente ejemplo para obtener una versión más larga de este comando, que también requiere de los parámetros `--region` y `--endpoint` para identificar el clúster que se va a agregar. Si, por ejemplo, la región del clúster es diferente a la que está configurada como predeterminada de la AWS CLI, debe usar el parámetro `--region` para usar la región correcta. Además, puede especificar el punto de enlace de la AWS CloudHSM API que se utilizará para la llamada, lo que puede ser necesario para varias configuraciones de red, como el uso de puntos de enlace de la interfaz de VPC para los que no se utiliza el nombre de host DNS predeterminado. AWS CloudHSM

Cómo agregar un clúster mediante los parámetros **`cluster-id`**, **`endpoint`** y **`region`**

Example

Utilice el parámetro `configure-cli add-cluster` junto con los parámetros `cluster-id`, `endpoint` y `region`para agregar un clúster (con el ID de `cluster-1234567`) a la configuración.

Linux

```
$ sudo /opt/cloudhsm/bin/configure-cli add-cluster --cluster-id cluster-1234567 --
region us-east-1 --endpoint https://cloudhsmv2.us-east-1.amazonaws.com
```

Windows

```
C:\Program Files\Amazon\CloudHSM\> .\configure-cli.exe add-cluster --cluster-
id cluster-1234567 --region us-east-1 --endpoint https://cloudhsmv2.us-
east-1.amazonaws.com
```

Para obtener más información acerca de los parámetros `--cluster-id`, `--region` y `--endpoint`, consulte the section called “Parámetros”.

Parámetros

--cluster-id ***<Cluster ID>***

Realiza una llamada `DescribeClusters` para buscar todas las direcciones IP de la interfaz de red elástica (ENI) de HSM en el clúster asociado al ID del clúster. El sistema añade las direcciones IP de ENI a los archivos de configuración. AWS CloudHSM

> **Note**
>
> Si utiliza el `--cluster-id` parámetro de una instancia EC2 dentro de una VPC que no tiene acceso a la Internet pública, debe crear un punto final de la VPC de interfaz al que conectarse. AWS CloudHSM Para obtener más información acerca de los puntos de conexión de VPC, consulte ???.

Obligatorio: sí

--punto de conexión ***<endpoint>***

Especifique el punto final AWS CloudHSM de la API utilizado para realizar la llamada. `DescribeClusters` Debe configurar esta opción en combinación con `--cluster-id`.

Obligatorio: no

-- hsm-ca-cert <HsmCA Certificate Filepath>

Especifica la ruta del archivo al certificado CA de HSM.

Obligatorio: no

--region ***<region>***

Especifique la región de su clúster. Debe configurar esta opción en combinación con `--cluster-id`.

Si no proporciona el parámetro `--region`, el sistema elige la región intentando leer las variables de entorno `AWS_DEFAULT_REGION` o `AWS_REGION`. Si esas variables no están configuradas, el sistema comprueba la región asociada a su perfil en el archivo AWS Config (normalmente `~/.aws/config`), a menos que haya especificado un archivo diferente en la variable de entorno de `AWS_CONFIG_FILE`. Si no se establece ninguna de las opciones anteriores, el sistema utilizará la región `us-east-1` de forma predeterminada.

Obligatorio: no

-- server-client-cert-file <Client Certificate Filepath>

Ruta al certificado de cliente utilizado para la autenticación mutua de TLS cliente-servidor.

Utilice esta opción únicamente si no desea utilizar la clave y el certificado SSL/TLS predeterminados que incluimos en SDK 5 de cliente. Debe configurar esta opción en combinación con `--server-client-key-file`.

Obligatorio: no

-- server-client-key-file <Client Key Filepath>

Ruta a la clave de cliente utilizada para la autenticación mutua entre cliente y servidor con TLS.

Utilice esta opción únicamente si no desea utilizar la clave y el certificado SSL/TLS predeterminados que incluimos en SDK 5 de cliente. Debe configurar esta opción en combinación con `--server-client-cert-file`.

Obligatorio: no

configure-cli remove-cluster

Cuando se conecte a varios clústeres con CLI, utilice el `configure-cli remove-cluster` comando para eliminar un clúster de la configuración.

Sintaxis

```
configure-cli remove-cluster [OPTIONS]
        --cluster-id <CLUSTER ID>
        [-h, --help]
```

Ejemplos

Eliminación de un clúster mediante el parámetro **`cluster-id`**

Example

Utilice el parámetro `configure-cli remove-cluster` junto con el parámetro `cluster-id` para eliminar un clúster (con el ID de `cluster-1234567`) de su configuración.

Linux

```
$ sudo /opt/cloudhsm/bin/configure-cli remove-cluster --cluster-id cluster-1234567
```

Windows

```
C:\Program Files\Amazon\CloudHSM\> .\configure-cli.exe remove-cluster --cluster-
id cluster-1234567
```

Para obtener más información sobre el parámetro `--cluster-id`, consulte the section called “Parámetros”.

Parámetro

--cluster-id ***<Cluster ID>***

El ID del clúster que se va a eliminar de la configuración.

Obligatorio: sí

Uso de varios clústeres

Después de configurar varios clústeres con AWS CloudHSM CLI, utilice el `cloudhsm-cli` comando para interactuar con ellos.

Ejemplos

Establecer un valor predeterminado **`cluster-id`** cuando se utiliza el modo interactivo

Example

Utilice el parámetro ??? junto con el `cluster-id` parámetro para establecer un clúster predeterminado (con el ID de`cluster-1234567`) a partir de su configuración.

Linux

```
$ cloudhsm-cli interactive --cluster-id cluster-1234567
```

Windows

```
C:\Program Files\Amazon\CloudHSM\> .\cloudhsm-cli.exe interactive --cluster-
id cluster-1234567
```

Establecer el valor **`cluster-id`** cuando se ejecuta un solo comando

Example

Utilice el `cluster-id` parámetro para configurar el clúster (con el ID de`cluster-1234567`) del que se va ??? a obtener.

Linux

```
$ cloudhsm-cli cluster hsm-info --cluster-id cluster-1234567
```

Windows

```
C:\Program Files\Amazon\CloudHSM\> .\cloudhsm-cli.exe cluster hsm-info --cluster-
id cluster-1234567
```

Referencia para los comandos de la CLI de CloudHSM

La CLI de CloudHSM ayuda a los administradores a gestionar los usuarios de su clúster. AWS CloudHSM La CLI de CloudHSM se puede ejecutar en dos modos: modo interactivo y modo de comando único. Para un inicio rápido, consulte Introducción a la interfaz de la línea de comandos (CLI) de CloudHSM.

Para ejecutar la mayoría de los comandos de la CLI de CloudHSM, debe iniciar la CLI de CloudHSM e iniciar sesión en el HSM. Si agrega o elimina los HSM, actualice los archivos de configuración de la CLI de CloudHSM. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

En los temas siguientes, se describen los comandos de la CLI de CloudHSM.

Comando	Descripción	Tipo de usuario
cluster activate	Activa un clúster de CloudHSM y confirma que el clúster es nuevo. Esto debe hacerse antes de poder realizar cualquier otra operación.	Administrador desactivado
clúster hsm-info	Enumere los HSM de su clúster.	Todos [1], incluidos los usuarios no autenticados. No es necesario iniciar sesión.
signo criptográfico ecdsa	Genera una firma mediante una clave privada EC y el mecanismo de firma ECDSA.	Usuarios de criptografía (CU)
signo criptográfico rsa-pkcs	Genera una firma mediante una clave privada RSA y el mecanismo de firma RSA-PKCS.	CU
signo criptográfico rsa-pkcs-pss	Genera una firma mediante una clave privada de RSA y	CU

Comando	Descripción	Tipo de usuario
	el mecanismo de firma RSA-PKCS-PSS.	
verificación criptográfica ecdsa	Confirma que un archivo se ha firmado en el HSM con una clave pública determinada. Comprueba que la firma se generó mediante el mecanismo de firma ECDSA. Compara un archivo firmado con un archivo fuente y determina si ambos están relacionados criptográficamente en función de una clave pública y un mecanismo de firma del ECDSA determinados.	CU
verificación criptográfica rsa-pkcs	Confirma que un archivo se ha firmado en el HSM con una clave pública determinada. Comprueba que la firma se generó mediante el mecanismo de firma RSA-PKCS. Compara un archivo firmado con un archivo fuente y determina si ambos están relacionados criptográficamente en función de una clave pública rsa y un mecanismo de firma determinados.	CU

Comando	Descripción	Tipo de usuario
verificación criptográfica rsa-pkcs-pss	Confirma que un archivo se ha firmado en el HSM con una clave pública determina da. Comprueba que la firma se generó mediante el mecanismo de firma RSA-PKCS-PSS. Compara un archivo firmado con un archivo fuente y determina si ambos están relaciona dos criptográficamente en función de una clave pública rsa y un mecanismo de firma determinados.	CU
eliminar clave	Elimina una clave del clúster. AWS CloudHSM	CU
key generate-file	Genera un archivo de claves en el AWS CloudHSM clúster.	CU
clave generate-asymmetric-pair rsa	Genera un key pair de claves RSA asimétrico en el clúster AWS CloudHSM .	CU
clave, etc. generate-asymmetric-pair	Genera un key pair de curvas elípticas (EC) asimétricas en el clúster. AWS CloudHSM	CU
key generate-symmetric aes	Genera una clave AES simétrica en el clúster. AWS CloudHSM	CU
key generate-symmetric generic-secret	Genera una clave secreta genérica simétrica en el clúster. AWS CloudHSM	CU

Comando	Descripción	Tipo de usuario
importación de claves (pem)	Importa una clave de formato PEM a un HSM. Puede utilizarlo para importar claves públicas que se han generado fuera del HSM.	CU
key list	Busca todas las claves del usuario actual presente en el clúster AWS CloudHSM .	CU
clave: replicar	Replica una clave de un clúster de origen a un clúster de destino clonado.	CU
key set-attribute	Establece los atributos de las claves del AWS CloudHSM clúster.	Las CU pueden ejecutar este comando y los administradores pueden establecer el atributo de confianza.
key share	Comparte una clave con otras CU AWS CloudHSM del clúster.	CU
key unshare	Deja de compartir una clave con otras CU del AWS CloudHSM clúster.	CU
desempaquetar llaves aes-gcm	Desempaqueta una clave de carga útil en el clúster mediante la clave de empaquetado AES y el mecanismo de desempaquetado AES-GCM.	CU

Comando	Descripción	Tipo de usuario
abrir llaves aes-no-pad	Desempaqueta una clave de carga útil en el clúster mediante la clave de empaquetado AES y el mecanismo de desempaquetado AES-NO-PAD.	CU
desempaquetador de teclas aes-pkcs5-pad	Desempaqueta una clave de carga mediante la clave de empaquetado AES y el mecanismo de desempaquetado AES-PKCS5-PAD.	CU
abrir llaves aes-zero-pad	Desempaqueta una clave de carga útil en el clúster mediante la clave de empaquetado AES y el mecanismo de desempaquetado AES-ZERO-PAD.	CU
desempaquetar llaves cloudhsm-aes-gcm	Desempaqueta una clave de carga útil en el clúster mediante la clave de empaquetado AES y el mecanismo de desempaquetado CLOUDHSM-AES-GCM.	CU
desempaquetar claves rsa-aes	Desempaqueta una clave de carga mediante una clave privada de RSA y el mecanismo de desempaquetado de RSA-AES.	CU

Comando	Descripción	Tipo de usuario
desempaquetar claves rsa-oaep	Desempaqueta una clave de carga mediante la clave privada RSA y el mecanismo de desempaquetado RSA-OAEP.	CU
desempaquetar claves rsa-pkcs	Desempaqueta una clave de carga mediante la clave privada de RSA y el mecanismo de desempaquetado de RSA-PKCS.	CU
envoltorio para llaves aes-gcm	Envuelve una clave de carga mediante una clave AES en el HSM y el mecanismo de empaquetado AES-GCM.	CU
envoltorio para llaves aes-no-pad	Envuelve una clave de carga mediante una clave AES en el HSM y el mecanismo de empaquetado AES-NO-PAD.	CU
funda para llaves aes-pkcs5-pad	Envuelve una clave de carga mediante una clave AES en el HSM y el mecanismo de empaquetado AES-PKCS5-PAD.	CU
envoltorio para llaves aes-zero-pad	Envuelve una clave de carga mediante una clave AES en el HSM y el mecanismo de empaquetado AES-ZERO-PAD.	CU

Comando	Descripción	Tipo de usuario
envoltorio para llaves cloudhsm-aes-gcm	Envuelve una clave de carga mediante una clave AES en el HSM y el mecanismo de empaquetado CLOUDHSM-AES-GCM.	CU
envoltorio para llaves rsa-aes	Envuelve una clave de carga útil mediante una clave pública RSA en el HSM y el mecanismo de empaquetado RSA-AES.	CU
envoltorio de llaves rsa-oaep	Envuelve una clave de carga mediante una clave pública RSA en el HSM y el mecanismo de empaquetado RSA-OAEP.	CU

Comando	Descripción	Tipo de usuario
El key wrap rsa-pkcs comando agrupa una clave de carga mediante una clave pública RSA en el HSM y el mecanismo de empaquetado. RSA-PKCS El `extractable` atributo de la clave de carga útil debe estar establecido en. `true` Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas. Para usar el key wrap rsa-pkcs comando, primero debe tener una clave RSA en el clúster AWS CloudHSM . Puede generar un par de claves RSA mediante el clave generate-asymmetric-pair comando y el `wrap` atributo establecidos en. `true` Tipo de usuario Los tipos de usuarios siguientes pueden ejecutar este comando. • Usuarios de criptografía (CU) Requisitos • Para ejecutar este comando, debe iniciar	Envuelve una clave de carga mediante una clave pública RSA en el HSM y el mecanismo de empaquetado RSA-PKCS.	CU

Comando	Descripción	Tipo de usuario
login	AWS CloudHSM Inicie sesión en su clúster.	Administrador, usuario de criptografía (CU) y usuario de dispositivos (AU)
logout	Cierre sesión en su AWS CloudHSM clúster.	Administrador, CU y usuario de dispositivos (AU)
quorum token-sign delete	Elimina uno o más tokens de un servicio autorizado de cuórum.	Administrador
quorum token-sign generate	Genera un token para un servicio autorizado de cuórum.	Administrador
quorum token-sign list	Muestra todos los tokens de cuórum con firma de token presentes en el clúster de CloudHSM.	Todos [1], incluidos los usuarios no autenticados. No es necesario iniciar sesión.
signo simbólico de quórum list-quorum-values	Muestra los valores de quórum establecidos en el clúster de CloudHSM.	Todos [1], incluidos los usuarios no autenticados. No es necesario iniciar sesión.
quorum token-sign list-timeouts	Obtiene el tiempo de espera del token en segundos para todos los tipos de token.	Administrador y usuario de criptografía
signo simbólico de quórum set-quorum-value	Establece un nuevo valor de quorum para un servicio autorizado de cuórum.	Administrador
quorum token-sign set-timeout	Establece el tiempo de espera del token en segundos para cada tipo de token.	Administrador

Comando	Descripción	Tipo de usuario
user change-mfa	Cambia la estrategia de autenticación multifactor (MFA) de un usuario.	Administrador, CU
user change-password	Cambia las contraseñas de los usuarios en los HSM. Cualquier usuario puede cambiar su propia contraseña. Los CO pueden cambiar la contraseña de cualquier persona.	Administrador, CU
user create	Crea un usuario en el clúster. AWS CloudHSM	Administrador
user delete	Elimina un usuario del AWS CloudHSM clúster.	Administrador
user list	Muestra los usuarios del AWS CloudHSM clúster.	Todos [1], incluidos los usuarios no autenticados. No es necesario iniciar sesión.
user change-quorum token-sign register	Registra la estrategia de cuórum con firma de token para un usuario.	Administrador

Annotations

- [1] Todos los usuarios incluyen todos los roles de la lista y los usuarios que no han iniciado sesión.

Clúster

cluster es una categoría principal para un grupo de comandos que, en combinación con la categoría principal, crean un comando específico para los usuarios. Actualmente, la categoría de usuario consta de los siguientes comandos:

- cluster activate
- clúster hsm-info

cluster activate

Use el comando cluster activate en la CLI de CloudHSM para activar un nuevo clúster. Este comando debe ejecutarse antes de usar el clúster para llevar a cabo operaciones criptográficas.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Administrador desactivado

Sintaxis

Este comando no tiene parámetros.

```
aws-cloudhsm > help cluster activate
Activate a cluster

This command will set the initial Admin password. This process will cause your CloudHSM
 cluster to
move into the ACTIVE state.

USAGE:
    cloudhsm-cli cluster activate [OPTIONS] [--password <PASSWORD>]

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error

      --password <PASSWORD>
          Optional: Plaintext activation password If you do not include this argument
 you will be prompted for it

  -h, --help
          Print help (see a summary with '-h')
```

Ejemplo

Este comando activa el clúster estableciendo la contraseña inicial de su usuario administrador.

```
aws-cloudhsm > cluster activate
Enter password:
Confirm password:
{
  "error_code": 0,
  "data": "Cluster activation successful"
}
```

Temas relacionados de

- user create
- user delete
- cambio de la contraseña de un usuario

clúster hsm-info

Utilice el comando cluster hsm-info de la CLI de CloudHSM para enumerar los HSM del clúster. No es necesario haber iniciado sesión en la CLI de CloudHSM para ejecutar este comando.

> **Note**
>
> Si agrega o elimina los HSM, actualice los archivos de configuración que utilizan el AWS CloudHSM cliente y las herramientas de línea de comandos. De lo contrario, es posible que los cambios que realice no se hagan efectivos en todos los HSM del clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios. No es preciso haber iniciado sesión para ejecutar este comando.

Sintaxis

```
aws-cloudhsm > help cluster hsm-info
List info about each HSM in the cluster
```

```
Usage: cloudhsm-cli cluster hsm-info [OPTIONS]

Options:
      --cluster-id <CLUSTER_ID>  Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
  -h, --help                     Print help
```

Ejemplo

Este comando muestra los HSM presentes en el clúster. AWS CloudHSM

```
aws-cloudhsm > cluster hsm-info
{
  "error_code": 0,
  "data": {
    "hsms": [
      {
        "vendor": "Marvell Semiconductors, Inc.",
        "model": "NITROX-III CNN35XX-NFBE",
        "serial-number": "5.3G1941-ICM000590",
        "hardware-version-major": "5",
        "hardware-version-minor": "3",
        "firmware-version-major": "2",
        "firmware-version-minor": "6",
        "firmware-build-number": "16",
        "firmware-id": "CNN35XX-NFBE-FW-2.06-16"
        "fips-state": "2 [FIPS mode with single factor authentication]"
      },
      {
        "vendor": "Marvell Semiconductors, Inc.",
        "model": "NITROX-III CNN35XX-NFBE",
        "serial-number": "5.3G1941-ICM000625",
        "hardware-version-major": "5",
        "hardware-version-minor": "3",
        "firmware-version-major": "2",
        "firmware-version-minor": "6",
        "firmware-build-number": "16",
        "firmware-id": "CNN35XX-NFBE-FW-2.06-16"
        "fips-state": "2 [FIPS mode with single factor authentication]"
      },
```

```
      {
        "vendor": "Marvell Semiconductors, Inc.",
        "model": "NITROX-III CNN35XX-NFBE",
        "serial-number": "5.3G1941-ICM000663",
        "hardware-version-major": "5",
        "hardware-version-minor": "3",
        "firmware-version-major": "2",
        "firmware-version-minor": "6",
        "firmware-build-number": "16",
        "firmware-id": "CNN35XX-NFBE-FW-2.06-16"
        "fips-state": "2 [FIPS mode with single factor authentication]"
      }
    ]
  }
}
```

La salida tiene los siguientes atributos:

- Proveedor: el nombre del proveedor del HSM.
- Modelo: el número del modelo del HSM.
- Número de serie: el número de serie del HSM. Esto puede cambiar debido a las sustituciones.
- H ardware-version-major: La versión de hardware principal.
- H ardware-version-minor: La versión de hardware secundaria.
- F irmware-version-major: La versión principal del firmware.
- F irmware-version-minor: La versión de firmware secundaria.
- F irmware-build-number: El número de versión del firmware.
- Firmware-id: el ID del firmware, que incluye las versiones principales y secundarias junto con la compilación.

Temas relacionados de

- cluster activate

crypto

cryptoes una categoría principal para un grupo de comandos que, cuando se combinan con la categoría principal, crean un comando específico para las operaciones criptográficas. Actualmente, esta categoría consta de los siguientes comandos:

- signo criptográfico
  - signo criptográfico ecdsa
  - signo criptográfico rsa-pkcs
  - signo criptográfico rsa-pkcs-pss
- verificación criptográfica
  - verificación criptográfica ecdsa
  - verificación criptográfica rsa-pkcs
  - verificación criptográfica rsa-pkcs-pss

signo criptográfico

crypto signes una categoría principal para un grupo de comandos que, cuando se combina con la categoría principal, utiliza una clave privada elegida en el AWS CloudHSM clúster para generar una firma. crypto signtiene los siguientes subcomandos:

- signo criptográfico ecdsa
- signo criptográfico rsa-pkcs
- signo criptográfico rsa-pkcs-pss

Para poder crypto sign utilizarla, debe tener una clave privada en su HSM. Puedes generar una clave privada con los siguientes comandos:

- clave, generate-asymmetric-pair etc.
- clave generate-asymmetric-pair rsa

signo criptográfico ecdsa

El crypto sign ecdsa comando genera una firma mediante una clave privada EC y el mecanismo de firma ECDSA.

Para usar el crypto sign ecdsa comando, primero debe tener una clave privada EC en su AWS CloudHSM clúster. Puede generar una clave privada EC mediante el clave, generate-asymmetric-pair etc. comando con el `sign` atributo establecido en`true`.

> **Note**
>
> Las firmas se pueden verificar AWS CloudHSM con verificación criptográfica subcomandos.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help crypto sign ecdsa
Sign with the ECDSA mechanism

Usage: crypto sign ecdsa --key-filter [<KEY_FILTER>>...] --hash-
function <HASH_FUNCTION> <--data-path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --key-filter [<KEY_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 matching key
      --hash-function <HASH_FUNCTION>
          [possible values: sha1, sha224, sha256, sha384, sha512]
      --data-path <DATA_PATH>
          The path to the file containing the data to be signed
      --data <DATA>
          Base64 Encoded data to be signed
  -h, --help
          Print help
```

Ejemplo

Estos ejemplos muestran cómo crypto sign ecdsa generar una firma mediante el mecanismo de firma ECDSA y SHA256 la función hash. Este comando usa una clave privada en el HSM.

Example Ejemplo: generar una firma para los datos codificados en base 64

```
aws-cloudhsm > crypto sign ecdsa --key-filter attr.label=ec-private --hash-function
 sha256 --data YWJjMTIz
{
  "error_code": 0,
  "data": {
    "key-reference": "0x0000000000007808dd",
    "signature": "4zki+FzjhP7Z/KqoQvh4ueMAxQQVp7FQguZ2wOS3Q5bzk
+Hc5irV5iTkuxQbropPttVFZ8V6FgR2fz+sPegwCw=="
  }
}
```

Example Ejemplo: generar una firma para un archivo de datos

```
aws-cloudhsm > crypto sign ecdsa --key-filter attr.label=ec-private --hash-function
 sha256 --data-path data.txt
{
  "error_code": 0,
  "data": {
    "key-reference": "0x0000000000007808dd",
    "signature": "4zki+FzjhP7Z/KqoQvh4ueMAxQQVp7FQguZ2wOS3Q5bzk
+Hc5irV5iTkuxQbropPttVFZ8V6FgR2fz+sPegwCw=="
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<DATA>

Datos codificados en Base64 para firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<DATA_PATH>

Especifica la ubicación de los datos que se van a firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<HASH_FUNCTION>

Especifica la función hash.

Valores válidos:

- sha1
- sha224
- sha256
- sha384
- sha512

Obligatorio: sí

<KEY_FILTER>

Referencia clave (por ejemplo, key-reference=0xabc) o lista separada por espacios de atributos clave en forma de attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE para seleccionar una clave coincidente.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos clave de la CLI de CloudHSM.

Obligatorio: sí

Temas relacionados de

- signo criptográfico
- verificación criptográfica

signo criptográfico rsa-pkcs

El crypto sign rsa-pkcs comando genera una firma mediante una clave privada de RSA y el mecanismo de firma RSA-PKCS.

Para usar el crypto sign rsa-pkcs comando, primero debe tener una clave privada RSA en el clúster. AWS CloudHSM Puede generar una clave privada RSA mediante el clave generate-asymmetric-pair rsa comando con el `sign` atributo establecido en. `true`

> **Note**
>
> Las firmas se pueden verificar AWS CloudHSM con verificación criptográfica subcomandos.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help crypto sign rsa-pkcs
Sign with the RSA-PKCS mechanism

Usage: crypto sign rsa-pkcs --key-filter [<KEY_FILTER>>...] --hash-
function <HASH_FUNCTION> <--data-path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --key-filter [<KEY_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 matching key
      --hash-function <HASH_FUNCTION>
          [possible values: sha1, sha224, sha256, sha384, sha512]
      --data-path <DATA_PATH>
          The path to the file containing the data to be signed
      --data <DATA>
```

```
          Base64 Encoded data to be signed
  -h, --help
          Print help
```

Ejemplo

Estos ejemplos muestran cómo generar una firma mediante el mecanismo de firma RSA-PKCS y la función hash. crypto sign rsa-pkcs SHA256 Este comando usa una clave privada en el HSM.

Example Ejemplo: generar una firma para los datos codificados en base 64

```
aws-cloudhsm > crypto sign rsa-pkcs --key-filter attr.label=rsa-private --hash-function
 sha256 --data YWJjMTIz
{
  "error_code": 0,
  "data": {
    "key-reference": "0x00000000007008db",
    "signature": "XJ7mRyHnDRYrDWTQuuNb
+5mhoXx7VTsPMjgOQW4iMN7E42eNHj2Q0oovMmBdHUEH0F4HYG8FBJOBhvGuM8J/
z6y41GbowVpUT6WzjnIQs79K9i7i6oR1TYjLnIS3r/zkimuXcS8/ZxyDzru+GO9BUT9FFU/
of9cvu4Oyn6a5+IXuCbKNQs19uASuFARUTZ0a0Ny1CB1MulxUpqGTmI91J6evlP7k/2khwDmJ5E8FEar5/
Cvbn9t21p3Uj561ngTXrYbIZ2KHpef9jQh/cEIvFLG61sexJjQi8EdTxeDA
+I3ITO0qrvvESvA9+Sj7kdG2ceIicFS8/8LwyxiIC31UHQ=="
  }
}
```

Example Ejemplo: generar una firma para un archivo de datos

```
aws-cloudhsm > crypto sign rsa-pkcs --key-filter attr.label=rsa-private --hash-function
 sha256 --data-path data.txt
{
  "error_code": 0,
  "data": {
    "key-reference": "0x00000000007008db",
    "signature": "XJ7mRyHnDRYrDWTQuuNb
+5mhoXx7VTsPMjgOQW4iMN7E42eNHj2Q0oovMmBdHUEH0F4HYG8FBJOBhvGuM8J/
z6y41GbowVpUT6WzjnIQs79K9i7i6oR1TYjLnIS3r/zkimuXcS8/ZxyDzru+GO9BUT9FFU/
of9cvu4Oyn6a5+IXuCbKNQs19uASuFARUTZ0a0Ny1CB1MulxUpqGTmI91J6evlP7k/2khwDmJ5E8FEar5/
Cvbn9t21p3Uj561ngTXrYbIZ2KHpef9jQh/cEIvFLG61sexJjQi8EdTxeDA
+I3ITO0qrvvESvA9+Sj7kdG2ceIicFS8/8LwyxiIC31UHQ=="
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<DATA>

Datos codificados en Base64 para firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<DATA_PATH>

Especifica la ubicación de los datos que se van a firmar.

Obligatorio: Sí (a menos que se proporcione a través de los datos)

<HASH_FUNCTION>

Especifica la función hash.

Valores válidos:

- sha1
- sha224
- sha256
- sha384
- sha512

Obligatorio: sí

<KEY_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos clave de la CLI de CloudHSM.

Obligatorio: sí

Temas relacionados de

- signo criptográfico
- verificación criptográfica

signo criptográfico rsa-pkcs-pss

El crypto sign rsa-pkcs-pss comando genera una firma mediante una clave privada RSA y el mecanismo de `RSA-PKCS-PSS` firma.

Para usar el crypto sign rsa-pkcs-pss comando, primero debe tener una clave privada RSA en el clúster AWS CloudHSM . Puede generar una clave privada RSA mediante el clave generate-asymmetric-pair rsa comando con el `sign` atributo establecido en. `true`

> **Note**
>
> Las firmas se pueden verificar AWS CloudHSM con verificación criptográfica subcomandos.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help crypto sign rsa-pkcs-pss
Sign with the RSA-PKCS-PSS mechanism

Usage: crypto sign rsa-pkcs-pss [OPTIONS] --key-filter [<KEY_FILTER>...] --
hash-function <HASH_FUNCTION> --mgf <MGF> --salt-length <SALT_LENGTH> <--data-
path <DATA_PATH>|--data <DATA>>

Options:
```

```
      --cluster-id <CLUSTER_ID>        Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
      --key-filter [<KEY_FILTER>...]   Key reference (e.g. key-
reference=0xabc) or space separated list of key attributes in the form of
 attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a matching key
      --hash-function <HASH_FUNCTION>  [possible values: sha1, sha224, sha256, sha384,
 sha512]
      --data-path <DATA_PATH>          The path to the file containing the data to be
 signed
      --data <DATA>                    Base64 Encoded data to be signed
      --mgf <MGF>                      The mask generation function [possible values:
 mgf1-sha1, mgf1-sha224, mgf1-sha256, mgf1-sha384, mgf1-sha512]
      --salt-length <SALT_LENGTH>      The salt length
  -h, --help                           Print help
```

Ejemplo

Estos ejemplos muestran cómo crypto sign rsa-pkcs-pss generar una firma mediante el mecanismo de firma y la RSA-PKCS-PSS función SHA256 hash. Este comando usa una clave privada en el HSM.

Example Ejemplo: generar una firma para los datos codificados en base 64

```
aws-cloudhsm > crypto sign rsa-pkcs-pss --key-filter attr.label=rsa-private --hash-
function sha256 --data YWJjMTIz --salt-length 10 --mgf mgf1-sha256
{
  "error_code": 0,
  "data": {
    "key-reference": "0x0000000007008db",
    "signature": "H/z1rYVMzNAa31K4amE5MTiwGxDdCTgQXCJXRBKVOVm7ZuyI0fGE4sT/BUN
+977mQEV2TqtWpTsiF2IpwGM1VfSBRt7h/g4o6YERm1tTQLl7q+AJ7uGGK37zCsWQrAo7Vy8NzPShxekePo/
ZegrB1aHWN1fE8H3IPUKqLuMDI9o1Jq6kM986ExS7YmeOIclcZkyykTWqHLQVL2C3+A2bHJZBqRcM5XoIpk8HkPypjpN
+m4FNUds30GAemoOMl6asSrEJSthaZWV53OBsDOqzA8Rt8JdhXS+GZp3vNLdL1OTBELDPweXVgAu4dBX0FOvpw/
gg6sNvuaDK4YOBv2fqKg=="
  }
}
```

Example Ejemplo: generar una firma para un archivo de datos

```
aws-cloudhsm > crypto sign rsa-pkcs-pss --key-filter attr.label=rsa-private --hash-
function sha256 --data-path data.txt --salt-length 10 --mgf mgf1-sha256
{
  "error_code": 0,
```

```
  "data": {
    "key-reference": "0x00000000007008db",
    "signature": "H/z1rYVMzNAa31K4amE5MTiwGxDdCTgQXCJXRBKVOVm7ZuyI0fGE4sT/BUN
+977mQEV2TqtWpTsiF2IpwGM1VfSBRt7h/g4o6YERm1tTQLl7q+AJ7uGGK37zCsWQrAo7Vy8NzPShxekePo/
ZegrB1aHWN1fE8H3IPUKqLuMDI9o1Jq6kM986ExS7YmeOIclcZkyykTWqHLQVL2C3+A2bHJZBqRcM5XoIpk8HkPypjpN
+m4FNUds30GAemoOMl6asSrEJSthaZWV53OBsDOqzA8Rt8JdhXS+GZp3vNLdL1OTBELDPweXVgAu4dBX0FOvpw/
gg6sNvuaDK4YOBv2fqKg=="
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<DATA>

Datos codificados en Base64 para firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<DATA_PATH>

Especifica la ubicación de los datos que se van a firmar.

Obligatorio: Sí (a menos que se proporcione a través de los datos)

<HASH_FUNCTION>

Especifica la función hash.

Valores válidos:

- sha1
- sha224
- sha256
- sha384
- sha512

Obligatorio: sí

<KEY_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos clave de la CLI de CloudHSM.

Obligatorio: sí

<MGF>

Especifica la función de generación de máscaras.

> **Note**
>
> La función hash de la función de generación de máscaras debe coincidir con la función hash del mecanismo de firma.

Valores válidos:

- mgf1-sha1
- mgf1-sha224
- mgf1-sha256
- mgf1-sha384
- mgf1-sha512

Obligatorio: sí

<SALT_LENGTH>

Especifica la longitud de la sal.

Obligatorio: sí

Temas relacionados de

- signo criptográfico
- verificación criptográfica

Temas relacionados de

- verificación criptográfica

verificación criptográfica

crypto verifyes una categoría principal para un grupo de comandos que, cuando se combina con la categoría principal, confirma si un archivo se ha firmado con una clave determinada. crypto verifytiene los siguientes subcomandos:

- crypto verify ecdsa
- crypto verify rsa-pkcs
- verificación criptográfica rsa-pkcs-pss

El crypto verify comando compara un archivo firmado con un archivo fuente y analiza si están relacionados criptográficamente en función de una clave pública y un mecanismo de firma determinados.

> **Note**
>
> Se puede iniciar sesión en los archivos AWS CloudHSM con la signo criptográfico operación.

verificación criptográfica ecdsa

El crypto verify ecdsa comando se utiliza para completar las siguientes operaciones:

- Confirme que un archivo se haya firmado en el HSM con una clave pública determinada.
- Compruebe que la firma se haya generado mediante el mecanismo de firma ECDSA.
- Compare un archivo firmado con un archivo fuente y determine si ambos están relacionados criptográficamente en función de una clave pública y un mecanismo de firma del ECDSA determinados.

Para usar el crypto verify ecdsa comando, primero debe tener una clave pública EC en su clúster. AWS CloudHSM Puede importar una clave pública EC mediante el pem de importación de claves comando con el `verify` atributo establecido en`true`.

> **Note**
>
> Puede generar una firma en la CLI de CloudHSM con subcomandossigno criptográfico.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help crypto verify ecdsa
Verify with the ECDSA mechanism

Usage: crypto verify ecdsa --key-filter [<KEY_FILTER>...] --hash-
function <HASH_FUNCTION> <--data-path <DATA_PATH>|--data <DATA>> <--signature-
path <SIGNATURE_PATH>|--signature <SIGNATURE>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --key-filter [<KEY_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 matching key
      --hash-function <HASH_FUNCTION>
          [possible values: sha1, sha224, sha256, sha384, sha512]
      --data-path <DATA_PATH>
          The path to the file containing the data to be verified
      --data <DATA>
          Base64 encoded data to be verified
      --signature-path <SIGNATURE_PATH>
          The path to where the signature is located
```

```
      --signature <SIGNATURE>
          Base64 encoded signature to be verified
  -h, --help
          Print help
```

Ejemplo

Estos ejemplos muestran cómo verificar una firma que se generó mediante el mecanismo de firma ECDSA y la función hash. crypto verify ecdsa SHA256 Este comando usa una clave pública en el HSM.

Example Ejemplo: compruebe una firma codificada en Base64 con datos codificados en Base64

```
aws-cloudhsm > crypto verify ecdsa --hash-function sha256 --key-filter attr.label=ec-
public --data YWJjMTIz --signature 4zki+FzjhP7Z/KqoQvh4ueMAxQQVp7FQguZ2wOS3Q5bzk
+Hc5irV5iTkuxQbropPttVFZ8V6FgR2fz+sPegwCw==
{
  "error_code": 0,
  "data": {
    "message": "Signature verified successfully"
  }
}
```

Example Ejemplo: compruebe un archivo de firma con un archivo de datos

```
aws-cloudhsm > crypto verify ecdsa --hash-function sha256 --key-filter attr.label=ec-
public --data-path data.txt --signature-path signature-file
{
   "error_code": 0,
  "data": {
    "message": "Signature verified successfully"
  }
}
```

Example Ejemplo: demostrar una relación de firma falsa

Este comando verifica si los datos ubicados en `/home/data` fueron firmados por una clave pública con la etiqueta `ecdsa-public` utilizando el mecanismo de firma ECDSA para generar la firma ubicada en. `/home/signature` Como los argumentos dados no constituyen una verdadera relación de firma, el comando devuelve un mensaje de error.

```
aws-cloudhsm > crypto verify ecdsa --hash-function sha256 --
key-filter attr.label=ec-public --data aW52YWxpZA== --signature
 +ogk7M7S3iTqFg3SndJfd91dZFr5Qo6YixJl8JwcvqqVgsVuO6o+VKvTRjz0/V05kf3JJbBLr87Q
+wLWcMAJfA==
{
  "error_code": 1,
  "data": "Signature verification failed"
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<DATA>

Datos codificados en Base64 para firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<DATA_PATH>

Especifica la ubicación de los datos que se van a firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<HASH_FUNCTION>

Especifica la función hash.

Valores válidos:

- sha1
- sha224
- sha256
- sha384
- sha512

Obligatorio: sí

<KEY_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos clave de la CLI de CloudHSM.

Obligatorio: sí

<SIGNATURE>

Firma codificada en Base64.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de firma)

<SIGNATURE_PATH>

Especifica la ubicación de la firma.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de firma)

Temas relacionados de

- signo criptográfico
- verificación criptográfica

verificación criptográfica rsa-pkcs

El crypto verify rsa-pkcs comando se utiliza para completar las siguientes operaciones:

- Confirme que un archivo se haya firmado en el HSM con una clave pública determinada.
- Compruebe que la firma se haya generado mediante el mecanismo de `RSA-PKCS` firma.
- Compare un archivo firmado con un archivo fuente y determine si ambos están relacionados criptográficamente en función de una clave pública rsa y un mecanismo de firma determinados.

Para usar el crypto verify rsa-pkcs comando, primero debe tener una clave pública RSA en el clúster. AWS CloudHSM

> **Note**
>
> Puede generar una firma mediante la CLI de CloudHSM con los subcomandossigno criptográfico.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help crypto verify rsa-pkcs
Verify with the RSA-PKCS mechanism

Usage: crypto verify rsa-pkcs --key-filter [<KEY_FILTER>...] --hash-
function <HASH_FUNCTION> <--data-path <DATA_PATH>|--data <DATA>> <--signature-
path <SIGNATURE_PATH>|--signature <SIGNATURE>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --key-filter [<KEY_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 matching key
      --hash-function <HASH_FUNCTION>
          [possible values: sha1, sha224, sha256, sha384, sha512]
      --data-path <DATA_PATH>
          The path to the file containing the data to be verified
      --data <DATA>
          Base64 encoded data to be verified
      --signature-path <SIGNATURE_PATH>
```

```
          The path to where the signature is located
      --signature <SIGNATURE>
          Base64 encoded signature to be verified
  -h, --help
          Print help
```

Ejemplo

Estos ejemplos muestran cómo comprobar una firma que crypto verify rsa-pkcs se generó mediante el mecanismo de firma RSA-PKCS y la función hash. SHA256 Este comando usa una clave pública en el HSM.

Example Ejemplo: compruebe una firma codificada en Base64 con datos codificados en Base64

```
aws-cloudhsm > crypto verify rsa-pkcs --hash-function sha256 --key-filter
 attr.label=rsa-public --data YWJjMTIz --signature XJ7mRyHnDRYrDWTQuuNb
+5mhoXx7VTsPMjgOQW4iMN7E42eNHj2Q0oovMmBdHUEH0F4HYG8FBJOBhvGuM8J/
z6y41GbowVpUT6WzjnIQs79K9i7i6oR1TYjLnIS3r/zkimuXcS8/ZxyDzru+GO9BUT9FFU/
of9cvu4Oyn6a5+IXuCbKNQs19uASuFARUTZ0a0Ny1CB1MulxUpqGTmI91J6evlP7k/2khwDmJ5E8FEar5/
Cvbn9t21p3Uj561ngTXrYbIZ2KHpef9jQh/cEIvFLG61sexJjQi8EdTxeDA
+I3ITO0qrvvESvA9+Sj7kdG2ceIicFS8/8LwyxiIC31UHQ==
{
  "error_code": 0,
  "data": {
    "message": "Signature verified successfully"
  }
}
```

Example Ejemplo: compruebe un archivo de firma con un archivo de datos

```
aws-cloudhsm > crypto verify rsa-pkcs --hash-function sha256 --key-filter
 attr.label=rsa-public --data-path data.txt --signature-path signature-file
{
  "error_code": 0,
  "data": {
    "message": "Signature verified successfully"
  }
}
```

Example Ejemplo: demostrar una relación de firma falsa

Este comando verifica si los datos no válidos se firmaron mediante una clave pública con la etiqueta `rsa-public` mediante el mecanismo de firma RSAPKCS para generar la firma ubicada en. `/home/signature` Como los argumentos dados no constituyen una verdadera relación de firma, el comando devuelve un mensaje de error.

```
aws-cloudhsm > crypto verify rsa-pkcs --hash-function sha256 --key-filter
 attr.label=rsa-public --data aW52YWxpZA== --signature XJ7mRyHnDRYrDWTQuuNb
+5mhoXx7VTsPMjgOQW4iMN7E42eNHj2Q0oovMmBdHUEH0F4HYG8FBJOBhvGuM8J/
z6y41GbowVpUT6WzjnIQs79K9i7i6oR1TYjLnIS3r/zkimuXcS8/ZxyDzru+GO9BUT9FFU/
of9cvu4Oyn6a5+IXuCbKNQs19uASuFARUTZ0a0Ny1CB1MulxUpqGTmI91J6evlP7k/2khwDmJ5E8FEar5/
Cvbn9t21p3Uj561ngTXrYbIZ2KHpef9jQh/cEIvFLG61sexJjQi8EdTxeDA
+I3ITO0qrvvESvA9+Sj7kdG2ceIicFS8/8LwyxiIC31UHQ==
{
  "error_code": 1,
  "data": "Signature verification failed"
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<DATA>

Datos codificados en Base64 para firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<DATA_PATH>

Especifica la ubicación de los datos que se van a firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<HASH_FUNCTION>

Especifica la función hash.

Valores válidos:

- sha1
- sha224
- sha256
- sha384
- sha512

Obligatorio: sí

<KEY_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos clave de la CLI de CloudHSM.

Obligatorio: sí

<SIGNATURE>

Firma codificada en Base64.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de firma)

<SIGNATURE_PATH>

Especifica la ubicación de la firma.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de firma)

Temas relacionados de

- signo criptográfico
- verificación criptográfica

verificación criptográfica rsa-pkcs-pss

El crypto sign rsa-pkcs-pss comando se utiliza para completar las siguientes operaciones.

- Confirme que un archivo se haya firmado en el HSM con una clave pública determinada.

- Compruebe que la firma se haya generado mediante el mecanismo de firma RSA-PKCS-PSS.
- Compare un archivo firmado con un archivo fuente y determine si ambos están relacionados criptográficamente en función de una clave pública rsa y un mecanismo de firma determinados.

Para usar el crypto verify rsa-pkcs-pss comando, primero debe tener una clave pública RSA en el clúster. AWS CloudHSM Puede importar una clave pública RSA mediante el comando key import pem (ADD UNWRAP LINK HERE) con el `verify` atributo establecido en. `true`

> **Note**
>
> Puede generar una firma mediante la CLI de CloudHSM con los subcomandossigno criptográfico.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help crypto verify rsa-pkcs-pss
Verify with the RSA-PKCS-PSS mechanism

Usage: crypto verify rsa-pkcs-pss --key-filter [<KEY_FILTER>...] --hash-
function <HASH_FUNCTION> --mgf <MGF> --salt-length >SALT_LENGTH< <--data-
path <DATA_PATH>|--data <DATA> <--signature-path <SIGNATURE_PATH>|--
signature <SIGNATURE>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
```

```
      --key-filter [<KEY_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 matching key
      --hash-function <HASH_FUNCTION>
          [possible values: sha1, sha224, sha256, sha384, sha512]
      --data-path <DATA_PATH>
          The path to the file containing the data to be verified
      --data <DATA>
          Base64 encoded data to be verified
      --signature-path <SIGNATURE_PATH>
          The path to where the signature is located
      --signature <SIGNATURE>
          Base64 encoded signature to be verified
      --mgf <MGF>
          The mask generation function [possible values: mgf1-sha1, mgf1-sha224, mgf1-
sha256, mgf1-sha384, mgf1-sha512]
      --salt-length <SALT_LENGTH>
          The salt length
  -h, --help
          Print help
```

Ejemplo

Estos ejemplos muestran cómo comprobar una firma que se crypto verify rsa-pkcs-pss generó mediante el mecanismo de firma RSA-PKCS-PSS y la función hash. SHA256 Este comando usa una clave pública en el HSM.

Example Ejemplo: compruebe una firma codificada en Base64 con datos codificados en Base64

```
aws-cloudhsm > crypto verify rsa-pkcs-pss --key-filter attr.label=rsa-public
 --hash-function sha256 --data YWJjMTIz --salt-length 10 --mgf mgf1-sha256
 --signature H/z1rYVMzNAa31K4amE5MTiwGxDdCTgQXCJXRBKVOVm7ZuyI0fGE4sT/BUN
+977mQEV2TqtWpTsiF2IpwGM1VfSBRt7h/g4o6YERm1tTQLl7q+AJ7uGGK37zCsWQrAo7Vy8NzPShxekePo/
ZegrB1aHWN1fE8H3IPUKqLuMDI9o1Jq6kM986ExS7YmeOIclcZkyykTWqHLQVL2C3+A2bHJZBqRcM5XoIpk8HkPypjpN
+m4FNUds30GAemoOMl6asSrEJSthaZWV53OBsDOqzA8Rt8JdhXS+GZp3vNLdL1OTBELDPweXVgAu4dBX0FOvpw/
gg6sNvuaDK4YOBv2fqKg==
{
  "error_code": 0,
  "data": {
    "message": "Signature verified successfully"
  }
}
```

Example Ejemplo: compruebe un archivo de firma con un archivo de datos

```
aws-cloudhsm > crypto verify rsa-pkcs-pss --key-filter attr.label=rsa-public --hash-
function sha256 --data-path data.txt --salt-length 10 --mgf mgf1-sha256 --signature
 signature-file
{
  "error_code": 0,
  "data": {
    "message": "Signature verified successfully"
  }
}
```

Example Ejemplo: demostrar una relación de firma falsa

Este comando verifica si los datos no válidos se firmaron mediante una clave pública con la etiqueta `rsa-public` mediante el mecanismo de firma RSAPKCSPSS para generar la firma ubicada en. `/home/signature` Como los argumentos dados no constituyen una verdadera relación de firma, el comando devuelve un mensaje de error.

```
aws-cloudhsm > crypto verify rsa-pkcs-pss --key-filter attr.label=rsa-public
 --hash-function sha256 --data aW52YWxpZA== --salt-length 10 --mgf mgf1-sha256
 --signature H/z1rYVMzNAa31K4amE5MTiwGxDdCTgQXCJXRBKVOVm7ZuyI0fGE4sT/BUN
+977mQEV2TqtWpTsiF2IpwGM1VfSBRt7h/g4o6YERm1tTQLl7q+AJ7uGGK37zCsWQrAo7Vy8NzPShxekePo/
ZegrB1aHWN1fE8H3IPUKqLuMDI9o1Jq6kM986ExS7YmeOIclcZkyykTWqHLQVL2C3+A2bHJZBqRcM5XoIpk8HkPypjpN
+m4FNUds30GAemoOMl6asSrEJSthaZWV53OBsDOqzA8Rt8JdhXS+GZp3vNLdL1OTBELDPweXVgAu4dBX0FOvpw/
gg6sNvuaDK4YOBv2fqKg==
{
  "error_code": 1,
  "data": "Signature verification failed"
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<DATA>

Datos codificados en Base64 para firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<DATA_PATH>

Especifica la ubicación de los datos que se van a firmar.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de datos)

<HASH_FUNCTION>

Especifica la función hash.

Valores válidos:

- sha1
- sha224
- sha256
- sha384
- sha512

Obligatorio: sí

<KEY_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos clave de la CLI de CloudHSM.

Obligatorio: sí

<MFG>

Especifica la función de generación de máscaras.

> **Note**
>
> La función hash de la función de generación de máscaras debe coincidir con la función hash del mecanismo de firma.

Valores válidos:

- mgf1-sha1
- mgf1-sha224
- mgf1-sha256
- mgf1-sha384
- mgf1-sha512

Obligatorio: sí

<SIGNATURE>

Firma codificada en Base64.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de firma)

<SIGNATURE_PATH>

Especifica la ubicación de la firma.

Obligatorio: Sí (a menos que se proporcione a través de la ruta de firma)

Temas relacionados de

- signo criptográfico
- verificación criptográfica

key

key es una categoría principal para un grupo de comandos que, cuando se combinan con la categoría principal, crean un comando específico para las claves. Actualmente, esta categoría consta de los siguientes comandos:

- eliminar clave
- key generate-file
- clave generate-asymmetric-pair
  - clave generate-asymmetric-pair rsa
  - clave generate-asymmetric-pair , etc.
- key generate-symmetric
  - key generate-symmetric aes

  - key generate-symmetric generic-secret
- pem de importación de claves
- key list
- réplica clave
- key set-attribute
- key share
- key unshare
- desempaquetar llaves
- envoltorio para llaves

eliminar clave

Use el key delete comando de la CLI de CloudHSM para eliminar una clave AWS CloudHSM de un clúster. Solo puede eliminar las claves de una en una. La eliminación de una clave de un par de claves no influye en la otra clave del par.

Solo el CU que creó la clave y, por tanto, es su propietario, puede borrarla. Los usuarios que comparten la clave, pero no la poseen, pueden utilizarla en operaciones criptográficas, pero no pueden borrarla.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key delete
Delete a key in the HSM cluster

Usage: key delete [OPTIONS] --filter [<FILTER>...]
```

```
Options:
      --cluster-id <CLUSTER_ID>  Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
      --filter [<FILTER>...]     Key reference (e.g. key-reference=0xabc)
 or space separated list of key attributes in the form of
 attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a matching key for deletion
  -h, --help                     Print help
```

Ejemplo

```
aws-cloudhsm > key delete --filter attr.label="ec-test-public-key"
{
  "error_code": 0,
  "data": {
    "message": "Key deleted successfully"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente para su eliminación.

Para obtener una lista de los atributos de clave CLI de CloudHSM admitidos, consulte Atributos de clave de la CLI de CloudHSM

Obligatorio: sí

Temas relacionados de

- key list
- key generate-file

- key unshare
- Atributos de clave de la CLI de CloudHSM
- Uso de la CLI de CloudHSM para filtrar claves

key generate-file

El key generate-file comando exporta una clave asimétrica del HSM. Si el destino es una clave privada, la referencia a la clave privada se exportará en un formato PEM falso. Si el destino es una clave pública, los bytes de la clave pública se exportarán en formato PEM.

El archivo PEM falso, que no contiene el material de clave privada propiamente dicho, sino que hace referencia a la clave privada del HSM, se puede utilizar para establecer la transferencia de SSL/TLS desde el servidor web a. AWS CloudHSM Para obtener más información, consulte Descarga de SSL/TLS.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key generate-file
Generate a key file from a key in the HSM cluster. This command does not export any
 private key data from the HSM

Usage: key generate-file --encoding <ENCODING> --path <PATH> --filter [<FILTER>...]

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --encoding <ENCODING>
          Encoding format for the key file
```

```
          Possible values:
          - reference-pem: PEM formatted key reference (supports private keys)
          - pem:           PEM format (supports public keys)

      --path <PATH>
          Filepath where the key file will be written

      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 matching key for file generation

  -h, --help
          Print help (see a summary with '-h')
```

Ejemplo

En este ejemplo, se muestra cómo se utiliza para generar un archivo de claves en el key generate-file clúster. AWS CloudHSM

Example

```
aws-cloudhsm > key generate-file --encoding reference-pem --path /tmp/ec-private-
key.pem --filter attr.label="ec-test-private-key"
{
  "error_code": 0,
  "data": {
    "message": "Successfully generated key file"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente para su eliminación.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos de clave de la CLI de CloudHSM

Obligatorio: no

<ENCODING>

Especifica el formato de codificación del archivo de claves.

Obligatorio: sí

<PATH>

Especifica la ruta del archivo en la que se escribirá el archivo de claves.

Obligatorio: sí

Temas relacionados de

- Atributos de clave de la CLI de CloudHSM
- Uso de la CLI de CloudHSM para filtrar claves
- clave generate-asymmetric-pair
- key generate-symmetric

clave generate-asymmetric-pair

key generate-asymmetric-pair es una categoría principal de un grupo de comandos que, cuando se combinan con la categoría principal, crean un comando que genera pares de claves asimétricas. Actualmente, esta categoría consta de los siguientes comandos:

- clave, generate-asymmetric-pair etc.
- clave generate-asymmetric-pair rsa

clave, generate-asymmetric-pair etc.

Utilice el key asymmetric-pair ec comando de la CLI de CloudHSM para generar un par de claves de curva elíptica (EC) asimétrica en el clúster. AWS CloudHSM

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key generate-asymmetric-pair ec
Generate an Elliptic-Curve Cryptography (ECC) key pair

Usage: key generate-asymmetric-pair ec [OPTIONS] --public-label <PUBLIC_LABEL> --
private-label <PRIVATE_LABEL> --curve <CURVE>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --public-label <PUBLIC_LABEL>
          Label for the public key
      --private-label <PRIVATE_LABEL>
          Label for the private key
      --session
          Creates a session key pair that exists only in the current session. The key
 cannot be recovered after the session ends
      --curve <CURVE>
          Elliptic curve used to generate the key pair [possible values: prime256v1,
 secp256r1, secp224r1, secp384r1, secp256k1, secp521r1]
      --public-attributes [<PUBLIC_KEY_ATTRIBUTES>...]
          Space separated list of key attributes to set for the generated EC public key
 in the form of KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE
      --private-attributes [<PRIVATE_KEY_ATTRIBUTES>...]
          Space separated list of key attributes to set for the generated EC private
 key in the form of KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE
  -h, --help
          Print help
```

Ejemplos

Estos ejemplos muestran cómo utilizar el comando key generate-asymmetric-pair ec para crear un par de claves EC.

Example Ejemplo: creación de un par de claves CE

```
aws-cloudhsm > key generate-asymmetric-pair ec \
    --curve secp224r1 \
    --public-label ec-public-key-example \
    --private-label ec-private-key-example
{
  "error_code": 0,
  "data": {
    "public_key": {
      "key-reference": "0x000000000012000b",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
      "attributes": {
        "key-type": "ec",
        "label": "ec-public-key-example",
        "id": "",
        "check-value": "0xd7c1a7",
        "class": "public-key",
        "encrypt": false,
        "decrypt": false,
        "token": false,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": false,
        "sign": false,
        "trusted": false,
        "unwrap": false,
        "verify": false,
        "wrap": false,
```

```
        "wrap-with-trusted": false,
        "key-length-bytes": 57,
        "ec-point":
 "0x047096513df542250a6b228fd9cb67fd0c903abc93488467681974d6f371083fce1d79da8ad1e9ede745fb9f38a
        "curve": "secp224r1"
      }
    },
"private_key": {
      "key-reference": "0x000000000012000c",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
      "attributes": {
        "key-type": "ec",
        "label": "ec-private-key-example",
        "id": "",
        "check-value": "0xd7c1a7",
        "class": "private-key",
        "encrypt": false,
        "decrypt": false,
        "token": false,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": false,
        "trusted": false,
        "unwrap": false,
        "verify": false,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 122,
```

```
        "ec-point":
 "0x047096513df542250a6b228fd9cb67fd0c903abc93488467681974d6f371083fce1d79da8ad1e9ede745fb9f38a
        "curve": "secp224r1"
      }
    }
  }
}
```

Example Ejemplo: creación de un par de claves CE con atributos opcionales

```
aws-cloudhsm > key generate-asymmetric-pair ec \
    --curve secp224r1 \
    --public-label ec-public-key-example \
    --private-label ec-private-key-example \
    --public-attributes token=true encrypt=true \
    --private-attributes token=true decrypt=true
{
  "error_code": 0,
  "data": {
    "public_key": {
      "key-reference": "0x0000000000002806eb",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "ec",
        "label": "ec-public-key-example",
        "id": "",
        "check-value": "0xedef86",
        "class": "public-key",
        "encrypt": true,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
```

```
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": false,
        "sign": false,
        "trusted": false,
        "unwrap": false,
        "verify": false,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 57,
        "ec-point":
"0x0487af31882189ec29eddf17a48e8b9cebb075b7b5afc5522fe9c83a029a450cc68592889a1ebf45f32240da514
        "curve": "secp224r1"
      }
    },
    "private_key": {
      "key-reference": "0x0000000000280c82",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "ec",
        "label": "ec-private-key-example",
        "id": "",
        "check-value": "0xedef86",
        "class": "private-key",
        "encrypt": false,
        "decrypt": true,
        "token": true,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
```

```
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": false,
        "trusted": false,
        "unwrap": false,
        "verify": false,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 122,
        "ec-point":
 "0x0487af31882189ec29eddf17a48e8b9cebb075b7b5afc5522fe9c83a029a450cc68592889a1ebf45f32240da514
        "curve": "secp224r1"
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<CURVE>

Especifica el identificador de la curva elíptica.

- prime256v1
- secp256r1
- secp224r1
- secp384r1
- secp256k1
- secp521r1

Obligatorio: sí

<PUBLIC_KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios para establecer la clave pública EC generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`)

Para obtener una lista de los atributos de clave compatibles, consulte Atributos de clave de la CLI de CloudHSM.

Obligatorio: no

<PUBLIC_LABEL>

Especifica una etiqueta definida por el usuario para la clave pública. El tamaño máximo permitido `label` es de 127 caracteres para el SDK de cliente 5.11 y versiones posteriores. El SDK de cliente 5.10 y versiones anteriores tiene un límite de 126 caracteres.

Obligatorio: sí

<PRIVATE_KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios para establecer la clave privada EC generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`).

Para obtener una lista de los atributos de clave compatibles, consulte Atributos de clave de la CLI de CloudHSM.

Obligatorio: no

<PRIVATE_LABEL>

Especifica la etiqueta de clave privada definida por el usuario. El tamaño máximo permitido `label` es de 127 caracteres para el SDK de cliente 5.11 y versiones posteriores. El SDK de cliente 5.10 y versiones anteriores tiene un límite de 126 caracteres.

Obligatorio: sí

<SESSION>

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

De forma predeterminada, las claves que se generan son claves persistentes (token). La transferencia a <SESSION> cambia este estado, lo que garantiza que la clave generada con este argumento sea una clave de sesión (efímera).

Obligatorio: no

Temas relacionados de

- Atributos de clave de la CLI de CloudHSM
- Uso de la CLI de CloudHSM para filtrar claves

clave generate-asymmetric-pair rsa

Utilice el key generate-asymmetric-pair rsa comando para generar un key pair de claves RSA asimétrico en su AWS CloudHSM clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key generate-asymmetric-pair rsa
Generate an RSA key pair

Usage: key generate-asymmetric-pair rsa [OPTIONS] --public-label <PUBLIC_LABEL>
 --private-label <PRIVATE_LABEL> --modulus-size-bits <MODULUS_SIZE_BITS> --public-
exponent <PUBLIC_EXPONENT>

Options:
      --cluster-id <CLUSTER_ID>
```

```
        Unique Id to choose which of the clusters in the config file to run the
operation against. If not provided, will fall back to the value provided when
interactive mode was started, or error
     --public-label <PUBLIC_LABEL>
         Label for the public key
     --private-label <PRIVATE_LABEL>
         Label for the private key
     --session
         Creates a session key pair that exists only in the current session. The key
cannot be recovered after the session ends
     --modulus-size-bits <MODULUS_SIZE_BITS>
         Modulus size in bits used to generate the RSA key pair
     --public-exponent <PUBLIC_EXPONENT>
         Public exponent used to generate the RSA key pair
     --public-attributes [<PUBLIC_KEY_ATTRIBUTES>...]
         Space separated list of key attributes to set for the generated RSA public
key in the form of KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE
     --private-attributes [<PRIVATE_KEY_ATTRIBUTES>...]
         Space separated list of key attributes to set for the generated RSA private
key in the form of KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE
 -h, --help
         Print help
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza `key generate-asymmetric-pair rsa` para crear un par de claves de RSA.

Example Ejemplo: creación de un par de claves RSA

```
aws-cloudhsm > key generate-asymmetric-pair rsa \
--public-exponent 65537 \
--modulus-size-bits 2048 \
--public-label rsa-public-key-example \
--private-label rsa-private-key-example
{
  "error_code": 0,
  "data": {
    "public_key": {
      "key-reference": "0x0000000000160010",
      "key-info": {
        "key-owners": [
          {
```

```
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "rsa-public-key-example",
        "id": "",
        "check-value": "0x498e1f",
        "class": "public-key",
        "encrypt": false,
        "decrypt": false,
        "token": false,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": false,
        "sign": false,
        "trusted": false,
        "unwrap": false,
        "verify": false,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 512,
        "public-exponent": "0x010001",
        "modulus":
 "0xdfca0669dc8288ed3bad99509bd21c7e6192661407021b3f4cdf4a593d939dd24f4d641af8e4e73b04c847731c6
e89a065e7d1a46ced96b46b909db2ab6be871ee700fd0a448b6e975bb64cae77c49008749212463e37a577baa57ce3e
bcebb7d20bd6df1948ae336ae23b52d73b7f3b6acc2543edb6358e08d326d280ce489571f4d34e316a2ea1904d513ca
        "modulus-size-bits": 2048
      }
    },
"private_key": {
      "key-reference": "0x0000000000160011",
      "key-info": {
        "key-owners": [
```

```
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "rsa-private-key-example",
        "id": "",
        "check-value": "0x498e1f",
        "class": "private-key",
        "encrypt": false,
        "decrypt": false,
        "token": false,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": false,
        "trusted": false,
        "unwrap": false,
        "verify": false,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 1217,
        "public-exponent": "0x010001",
        "modulus":
 "0xdfca0669dc8288ed3bad99509bd21c7e6192661407021b3f4cdf4a593d939dd24f4d641af8e4e73b04c847731c6
        "modulus-size-bits": 2048
      }
    }
  }
}
```

Example Ejemplo: creación de un par de claves RSA con atributos opcionales

```
aws-cloudhsm > key generate-asymmetric-pair rsa \
--public-exponent 65537 \
--modulus-size-bits 2048 \
--public-label rsa-public-key-example \
--private-label rsa-private-key-example \
--public-attributes token=true encrypt=true \
--private-attributes token=true decrypt=true
{
  "error_code": 0,
  "data": {
    "public_key": {
      "key-reference": "0x0000000000280cc8",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "rsa-public-key-example",
        "id": "",
        "check-value": "0x01fe6e",
        "class": "public-key",
        "encrypt": true,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": false,
        "sign": false,
        "trusted": false,
```

```
        "unwrap": false,
        "verify": false,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 512,
        "public-exponent": "0x010001",
        "modulus":
 "0xb1d27e857a876f4e9fd5de748a763c539b359f937eb4b4260e30d1435485a732c878cdad9c72538e2215351b1d4
73a80fdb457aa7b20cd61e486c326e2cfd5e124a7f6a996437437812b542e3caf85928aa866f0298580f7967ee6aa01
f6e6296d6c116d5744c6d60d14d3bf3cb978fe6b75ac67b7089bafd50d8687213b31abc7dc1bad422780d29c851d510
133022653225bd129f8491101725e9ea33e1ded83fb57af35f847e532eb30cd7e726f23910d2671c6364092e834697e
ac3160f0ca9725d38318b7",
        "modulus-size-bits": 2048
      }
    },
    "private_key": {
      "key-reference": "0x0000000000280cc7",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "rsa-private-key-example",
        "id": "",
        "check-value": "0x01fe6e",
        "class": "private-key",
        "encrypt": false,
        "decrypt": true,
        "token": true,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
```

```
        "sensitive": true,
        "sign": false,
        "trusted": false,
        "unwrap": false,
        "verify": false,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 1217,
        "public-exponent": "0x010001",
        "modulus":
 "0xb1d27e857a876f4e9fd5de748a763c539b359f937eb4b4260e30d1435485a732c878cdad9c72538e2215351b1d4
        "modulus-size-bits": 2048
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<MODULUS_SIZE_BITS>

Especifica la longitud del módulo en bits. El valor mínimo es 2048.

Obligatorio: sí

<PRIVATE_KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios para establecer la clave privada RSA generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`).

Para obtener una lista de los atributos de clave compatibles, consulte Atributos de clave de la CLI de CloudHSM.

Obligatorio: no

<PRIVATE_LABEL>

Especifica la etiqueta de clave privada definida por el usuario. El tamaño máximo permitido `label` es de 127 caracteres para el SDK de cliente 5.11 y versiones posteriores. El SDK de cliente 5.10 y versiones anteriores tiene un límite de 126 caracteres.

Obligatorio: sí

<PUBLIC_EXPONENT>

Especifica el exponente público. El valor debe ser un número impar superior o igual a 65537.

Obligatorio: sí

<PUBLIC_KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios para establecer la clave pública RSA generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`).

Para obtener una lista de los atributos de clave compatibles, consulte Atributos de clave de la CLI de CloudHSM.

Obligatorio: no

<PUBLIC_LABEL>

Especifica una etiqueta definida por el usuario para la clave pública. El tamaño máximo permitido `label` es de 127 caracteres para el SDK de cliente 5.11 y versiones posteriores. El SDK de cliente 5.10 y versiones anteriores tiene un límite de 126 caracteres.

Obligatorio: sí

<SESSION>

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

De forma predeterminada, las claves que se generan son claves persistentes (token). La transferencia a <SESSION> cambia este estado, lo que garantiza que la clave generada con este argumento sea una clave de sesión (efímera).

Obligatorio: no

Temas relacionados de

- Atributos de clave de la CLI de CloudHSM
- Uso de la CLI de CloudHSM para filtrar claves

key generate-symmetric

key generate-symmetric es una categoría principal para un grupo de comandos que, cuando se combinan con la categoría principal, crean un comando que genera claves simétricas. Actualmente, esta categoría consta de los siguientes comandos:

- key generate-symmetric aes
- key generate-symmetric generic-secret

key generate-symmetric aes

El key generate-symmetric aes comando genera una clave AES simétrica en el clúster AWS CloudHSM .

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key generate-symmetric aes
Generate an AES key

Usage: key generate-symmetric aes [OPTIONS] --label <LABEL> --key-length-
bytes <KEY_LENGTH_BYTES>
```

```
Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --label <LABEL>
          Label for the key
      --session
          Creates a session key that exists only in the current session. The key cannot
 be recovered after the session ends
      --key-length-bytes <KEY_LENGTH_BYTES>
          Key length in bytes
      --attributes [<KEY_ATTRIBUTES>...]
          Space separated list of key attributes to set for the generated AES key in
 the form of KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE
  -h, --help
          Print help
```

Ejemplos

En estos ejemplos se muestra cómo utilizar el comando key generate-symmetric aes para crear una clave AES.

Example Ejemplo: creación de una clave AES

```
aws-cloudhsm > key generate-symmetric aes \
--label example-aes \
--key-length-bytes 24
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000002e06bf",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
```

```
      "attributes": {
        "key-type": "aes",
        "label": "example-aes",
        "id": "",
        "check-value": "0x9b94bd",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": false,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 24
      }
    }
  }
}
```

Example Ejemplo: creación de una clave AES con atributos opcionales

```
aws-cloudhsm > key generate-symmetric aes \
--label example-aes \
--key-length-bytes 24 \
--attributes decrypt=true encrypt=true
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000002e06bf",
      "key-info": {
        "key-owners": [
```

```
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
      "attributes": {
        "key-type": "aes",
        "label": "example-aes",
        "id": "",
        "check-value": "0x9b94bd",
        "class": "secret-key",
        "encrypt": true,
        "decrypt": true,
        "token": true,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 24
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios que se debe configurar para la clave AES generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`).

Para obtener una lista de los atributos de clave compatibles, consulte Atributos de clave de la CLI de CloudHSM.

Obligatorio: no

<KEY-LENGTH-BYTES>

Especifica la longitud de la clave en bytes.

Valores válidos:

- 16, 24 y 32

Obligatorio: sí

<LABEL>

Especifica la etiqueta de clave AES definida por el usuario. El tamaño máximo permitido `label` es de 127 caracteres para el SDK de cliente 5.11 y versiones posteriores. El SDK de cliente 5.10 y versiones anteriores tiene un límite de 126 caracteres.

Obligatorio: sí

<SESSION>

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

De forma predeterminada, las claves que se generan son claves persistentes (token). La transferencia a <SESSION> cambia este estado, lo que garantiza que la clave generada con este argumento sea una clave de sesión (efímera).

Obligatorio: no

Temas relacionados de

- Atributos de clave de la CLI de CloudHSM
- Uso de la CLI de CloudHSM para filtrar claves

key generate-symmetric generic-secret

El comando key generate-asymmetric-pair genera una clave secreta genérica y simétrica en su clúster de AWS CloudHSM.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > key help generate-symmetric generic-secret
Generate a generic secret key

Usage: key generate-symmetric generic-secret [OPTIONS] --label <LABEL> --key-length-
bytes <KEY_LENGTH_BYTES>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --label <LABEL>
          Label for the key
      --session
          Creates a session key that exists only in the current session. The key cannot
 be recovered after the session ends
      --key-length-bytes <KEY_LENGTH_BYTES>
          Key length in bytes
      --attributes [<KEY_ATTRIBUTES>...]
```

```
          Space separated list of key attributes to set for the generated generic
 secret key in the form of KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE
  -h, --help
          Print help
```

Ejemplos

En estos ejemplos se muestra cómo usar el comando key generate-symmetric generic-secret para crear una clave secreta genérica.

Example Ejemplo: crear una clave secreta genérica

```
aws-cloudhsm > key generate-symmetric generic-secret \
--label example-generic-secret \
--key-length-bytes 256
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000002e08fd",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
      "attributes": {
        "key-type": "generic-secret",
        "label": "example-generic-secret",
        "id": "",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": false,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
```

```
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 256
      }
    }
  }
}
```

Example Ejemplo: cree una clave secreta genérica con atributos opcionales

```
aws-cloudhsm > key generate-symmetric generic-secret \
--label example-generic-secret \
--key-length-bytes 256 \
--attributes token=true encrypt=true
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000002e08fd",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
      "attributes": {
        "key-type": "generic-secret",
        "label": "example-generic-secret",
        "id": "",
        "class": "secret-key",
        "encrypt": true,
```

```
        "decrypt": false,
        "token": true,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 256
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<KEY_ATTRIBUTES>

Especifica una lista de atributos de clave separados por espacios que se debe configurar para la clave AES generada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo, `token=true`).

Para obtener una lista de los atributos de clave compatibles, consulte Atributos de clave de la CLI de CloudHSM.

Obligatorio: no

<KEY-LENGTH-BYTES>

Especifica la longitud de la clave en bytes.

Valores válidos:

- 1 a 800

Obligatorio: sí

<LABEL>

Especifica una etiqueta definida del usuario para la clave secreta genérica. El tamaño máximo permitido `label` es de 127 caracteres para el SDK de cliente 5.11 y versiones posteriores. El SDK de cliente 5.10 y versiones anteriores tiene un límite de 126 caracteres.

Obligatorio: sí

<SESSION>

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

De forma predeterminada, las claves que se generan son claves persistentes (token). La transferencia a <SESSION> cambia este estado, lo que garantiza que la clave generada con este argumento sea una clave de sesión (efímera).

Obligatorio: no

Temas relacionados de

- Atributos de clave de la CLI de CloudHSM
- Uso de la CLI de CloudHSM para filtrar claves

pem de importación de claves

El key import pem comando de AWS CloudHSM importa una clave de formato PEM a un HSM. Puede utilizarlo para importar claves públicas que se han generado fuera del HSM.

> **Note**
>
> Utilice el key generate-file comando para crear un archivo PEM estándar a partir de una clave pública o para crear un archivo PEM de referencia a partir de una clave privada.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key import pem
Import key from a PEM file

Usage: key import pem [OPTIONS] --path <PATH> --label <LABEL> --key-type-
class <KEY_TYPE_CLASS>
Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --path PATH>
          Path where the key is located in PEM format
      --label LABEL>
          Label for the imported key
      --key-type-class KEY_TYPE_CLASS>
          Key type and class of the imported key [possible values: ec-public, rsa-
public]
      --attributes [IMPORT_KEY_ATTRIBUTES>...]
          Space separated list of key attributes in the form of
 KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE for the imported key
  -h, --help
          Print help
```

Ejemplos

En este ejemplo se muestra cómo utilizar el key import pem comando para importar una clave pública RSA desde un archivo en formato PEM.

Example Ejemplo: importar una clave pública RSA

```
aws-cloudhsm > key import pem --path /home/example --label example-imported-key --key-
type-class rsa-public
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001e08e3",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "example-imported-key",
        "id": "0x",
        "check-value": "0x99fe93",
        "class": "public-key",
        "encrypt": false,
        "decrypt": false,
        "token": false,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": false,
        "sign": false,
        "trusted": false,
```

```
        "unwrap": false,
        "verify": false,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 512,
        "public-exponent": "0x010001",
        "modulus":
 "0x8e9c172c37aa22ed1ce25f7c3a7c936dadc532201400128b044ebb4b96#··3e4930ab910df5a2896eaeb8853cfe
        "modulus-size-bits": 2048
      }
    },
    "message": "Successfully imported key"
  }
}
```

Example Ejemplo: importar una clave pública RSA con atributos opcionales

```
aws-cloudhsm > key import pem --path /home/example --label example-imported-key-with-
attributes --key-type-class rsa-public --attributes verify=true
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001e08e3",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "session"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "example-imported-key-with-attributes",
        "id": "0x",
        "check-value": "0x99fe93",
        "class": "public-key",
        "encrypt": false,
        "decrypt": false,
        "token": false,
```

```
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": false,
        "sign": false,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 512,
        "public-exponent": "0x010001",
        "modulus":
 "0x8e9c172c37aa22ed1ce25f7c3a7c936dadc532201400128b044ebb4b96#··3e4930ab910df5a2896eaeb8853cfe
        "modulus-size-bits": 2048
      }
    },
    "message": "Successfully imported key"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PATH>

Especifica la ruta del archivo en la que se encuentra el archivo clave.

Obligatorio: sí

<LABEL>

Especifica una etiqueta definida por el usuario para la clave importada. El tamaño máximo permitido para `label` es de 126 caracteres.

Obligatorio: sí

<KEY_TYPE_CLASS>

Tipo de clave y clase de llave envuelta.

Valores posibles:

- ec-public
- rsa-public

Obligatorio: sí

<IMPORT_KEY_ATTRIBUTES>

Especifica una lista de atributos clave separados por espacios para establecer para la clave importada en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` (por ejemplo,`token=true`). Para obtener una lista de los atributos de clave compatibles, consulte Atributos de clave de la CLI de CloudHSM.

Obligatorio: no

Temas relacionados de

- signo criptográfico
- verificación criptográfica

key list

El key list comando busca todas las claves del usuario actual presentes en el AWS CloudHSM clúster. El resultado incluye claves que el usuario posee y comparte, así como todas las claves públicas en el clúster de CloudHSM.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Sintaxis

```
aws-cloudhsm > help key list
```

```
List the keys the current user owns, shares, and all public keys in the HSM cluster

Usage: key list [OPTIONS]

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select
 matching key(s) to list
      --max-items <MAX_ITEMS>
          The total number of items to return in the command's output. If the total
 number of items available is more than the value specified, a next-token is provided
 in the command's output. To resume pagination, provide the next-token value in the
 starting-token argument of a subsequent command [default: 10]
      --starting-token <STARTING_TOKEN>
          A token to specify where to start paginating. This is the next-token from a
 previously truncated response
  -v, --verbose
          If included, prints all attributes and key information for each matched key.
 By default each matched key only displays its key-reference and label attribute
  -h, --help
          Print help
```

Ejemplos

Los siguientes ejemplos muestran las diferentes formas de ejecutar el comando key list.

Example Ejemplo: búsqueda de todas las claves - de forma predeterminada

Este comando muestra las claves del usuario que ha iniciado sesión y que están presentes en el AWS CloudHSM clúster.

> **Note**
>
> De forma predeterminada, solo se muestran 10 claves del usuario que ha iniciado sesión y solo se muestran las `key-reference` y `label`. Utilice las opciones de paginación adecuadas para mostrar más o menos claves.

```
aws-cloudhsm > key list
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000000003d5",
        "attributes": {
          "label": "test_label_1"
        }
      },
      {
        "key-reference": "0x0000000000000626",
        "attributes": {
          "label": "test_label_2"
        }
      },.
      ...8 keys later...
    ],
    "total_key_count": 56,
    "returned_key_count": 10,
    "next_token": "10"
  }
}
```

Example Ejemplo: búsqueda de todas las claves - de forma detallada

El resultado incluye claves que el usuario posee y comparte, así como todas las claves públicas en los HSM.

> **Note**
>
> Nota: de forma predeterminada, solo se muestran 10 claves del usuario que ha iniciado sesión. Utilice las opciones de paginación adecuadas para mostrar más o menos claves.

```
aws-cloudhsm > key list --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
```

```
        "key-reference": "0x000000000012000c",
        "key-info": {
          "key-owners": [
            {
              "username": "cu1",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "session"
        },
        "attributes": {
          "key-type": "ec",
          "label": "ec-test-private-key",
          "id": "",
          "check-value": "0x2a737d",
          "class": "private-key",
          "encrypt": false,
          "decrypt": false,
          "token": false,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": false,
          "trusted": false,
          "unwrap": false,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 122,
          "ec-point":
"0x0442d53274a6c0ec1a23c165dcb9ccdd72c64e98ae1a9594bb5284e752c746280667e11f1e983493c1c605e0a80
          "curve": "secp224r1"
        }
      },
      {
        "key-reference": "0x000000000012000d",
        "key-info": {
```

```
          "key-owners": [
            {
              "username": "cu1",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "session"
        },
        "attributes": {
          "key-type": "ec",
          "label": "ec-test-public-key",
          "id": "",
          "check-value": "0x2a737d",
          "class": "public-key",
          "encrypt": false,
          "decrypt": false,
          "token": false,
          "always-sensitive": false,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": false,
          "sign": false,
          "trusted": false,
          "unwrap": false,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 57,
          "ec-point":
"0x0442d53274a6c0ec1a23c165dcb9ccdd72c64e98ae1a9594bb5284e752c746280667e11f1e983493c1c605e0a80
          "curve": "secp224r1"
        }
      }
    ],
      ...8 keys later...
    "total_key_count": 1580,
    "returned_key_count": 10
  }
```

```
}
```

Example Ejemplo: devolución paginada

El siguiente ejemplo muestra un subconjunto paginado de claves que muestra solo dos claves. A continuación, el ejemplo proporciona una llamada posterior para mostrar las dos claves siguientes.

```
aws-cloudhsm > key list --verbose --max-items 2
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000000030",
        "key-info": {
          "key-owners": [
            {
              "username": "cu1",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "98a6688d1d964ed7b45b9cec5c4b1909",
          "id": "",
          "check-value": "0xb28a46",
          "class": "secret-key",
          "encrypt": false,
          "decrypt": false,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
```

```
          "trusted": false,
          "unwrap": false,
          "verify": true,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      },
      {
        "key-reference": "0x0000000000000042",
        "key-info": {
          "key-owners": [
            {
              "username": "cu1",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "4ad6cdcbc02044e09fa954143efde233",
          "id": "",
          "check-value": "0xc98104",
          "class": "secret-key",
          "encrypt": true,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": false,
          "unwrap": true,
          "verify": true,
          "wrap": true,
          "wrap-with-trusted": false,
```

```
          "key-length-bytes": 16
        }
      }
    ],
    "total_key_count": 1580,
    "returned_key_count": 2,
    "next_token": "2"
  }
}
```

Para mostrar las dos claves siguientes, se puede realizar una llamada posterior:

```
aws-cloudhsm > key list --verbose --max-items 2 --starting-token 2
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000000081",
        "key-info": {
          "key-owners": [
            {
              "username": "cu1",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "6793b8439d044046982e5b895791e47f",
          "id": "",
          "check-value": "0x3f986f",
          "class": "secret-key",
          "encrypt": false,
          "decrypt": false,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
```

```
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": false,
          "unwrap": false,
          "verify": true,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      },
      {
        "key-reference": "0x0000000000000089",
        "key-info": {
          "key-owners": [
            {
              "username": "cu1",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "56b30fa05c6741faab8f606d3b7fe105",
          "id": "",
          "check-value": "0xe9201a",
          "class": "secret-key",
          "encrypt": false,
          "decrypt": false,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
```

```
          "trusted": false,
          "unwrap": false,
          "verify": true,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      }
    ],
    "total_key_count": 1580,
    "returned_key_count": 2,
    "next_token": "4"
  }
}
```

Para ver más ejemplos que demuestran cómo funciona el mecanismo de filtrado de claves en la CLI de CloudHSM, consulte Uso de la CLI de CloudHSM para filtrar claves.

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separada por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar las claves coincidentes de la lista.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos de clave de la CLI de CloudHSM

Obligatorio: no

<MAX_ITEMS>

El número total de elementos que se devuelven en la salida del comando. Si el número total de elementos disponible es mayor que el valor especificado, se proporciona un next-token en la salida del comando. Para reanudar la paginación, proporcione el valor del next-token en el argumento starting-token de un comando posterior.

Obligatorio: no

<STARTING_TOKEN>

Un token destinado a especificar dónde iniciar la paginación. Este es el next-token de una respuesta previamente truncada.

Obligatorio: no

<VERBOSE>

Si se incluye, muestra todos los atributos y la información clave de cada clave de coincidencia. De forma predeterminada, cada clave de coincidencia solo muestra su referencia clave y su atributo de etiqueta.

Obligatorio: no

Temas relacionados de

- eliminar clave
- key generate-file
- key unshare
- Atributos de clave de la CLI de CloudHSM
- Uso de la CLI de CloudHSM para filtrar claves

réplica clave

El key replicate comando replica una clave de un clúster de origen a un AWS CloudHSM clúster de destino AWS CloudHSM .

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

> **Note**
>
> Los usuarios criptográficos deben poseer la clave para usar este comando.

Requisitos

- Los clústeres de origen y destino deben ser clones. Esto significa que uno se creó a partir de una copia de seguridad del otro o que ambos se crearon a partir de una copia de seguridad común. Para obtener más información, consulte Creación de clústeres a partir de las copias de seguridad.
- El propietario de la clave debe existir en el clúster de destino. Además, si la clave se comparte con algún usuario, esos usuarios también deben existir en el clúster de destino.
- Para ejecutar este comando, debe iniciar sesión como CU en los clústeres de origen y de destino.
  - En el modo de comando único, el comando utilizará las variables de entorno CLOUDHSM_PIN y CLOUDHSM_ROLE para autenticarse en el clúster de origen. Para obtener más información, consulte Modo de comando único. Para proporcionar las credenciales del clúster de destino, debe establecer dos variables de entorno adicionales: DESTINATION_CLOUDHSM_PIN y DESTINATION_CLOUDHSM_ROLE:

    ```
    $ export DESTINATION_CLOUDHSM_ROLE=crypto-user
    ```

    ```
    $ export DESTINATION_CLOUDHSM_PIN=username:password
    ```

  - En el modo interactivo, los usuarios deberán iniciar sesión de forma explícita en los clústeres de origen y de destino.

Sintaxis

```
aws-cloudhsm > help key replicate
Replicate a key from a source to a destination cluster

Usage: key replicate --filter [<FILTER>...] --source-cluster-id <SOURCE_CLUSTER_ID> --
destination-cluster-id <DESTINATION_CLUSTER_ID>

Options:
      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select
 matching key on the source cluster
      --source-cluster-id <SOURCE_CLUSTER_ID>
          Source cluster ID
      --destination-cluster-id <DESTINATION_CLUSTER_ID>
          Destination cluster ID
  -h, --help
```

```
          Print help
```

Ejemplos

Example Ejemplo: replicar la clave

Este comando replica una clave de un clúster de origen en un clúster de destino clonado.

```
crypto-user-1@cluster-1234abcdefg > key replicate \
      --filter attr.label=example-key \
      --source-cluster-id cluster-1234abcdefg \
      --destination-cluster-id cluster-2345bcdefgh
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x0000000000300006",
      "key-info": {
        "key-owners": [
          {
            "username": "crypto-user-1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "example-key",
        "id": "0x",
        "check-value": "0x5e118e",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": true,
        "private": true,
```

```
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    },
    "message": "Successfully replicated key"
  }
}
```

Argumentos

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente en el clúster de origen.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos de clave de la CLI de CloudHSM

Obligatorio: sí

<SOURCE_CLUSTER_ID>

El ID del clúster de origen.

Obligatorio: sí

<DESTINATION_CLUSTER_ID>

El ID del clúster de destino.

Obligatorio: sí

Temas relacionados de

- Conexión a varios clústeres con CLI

key set-attribute

Use el key set-attribute comando para establecer los atributos de las claves del AWS CloudHSM clúster. Solamente el CU que creó la clave y que, por lo tanto, es su propietario, puede cambiar los atributos de la clave.

Para obtener una lista de los atributos clave que se pueden usar en la CLI de CloudHSM, consulte Atributos de clave de la CLI de CloudHSM.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Solo los usuarios de criptografía (CU) pueden ejecutar este comando.
- Los administradores pueden establecer el atributo de confianza.

Requisitos

Para ejecutar este comando, debe iniciar sesión como CU. Para establecer el atributo de confianza, debe iniciar sesión como usuario administrador.

Sintaxis

```
aws-cloudhsm > help key set-attribute
Set an attribute for a key in the HSM cluster

Usage: cloudhsm-cli key set-attribute [OPTIONS] --filter [<FILTER>...] --
name <KEY_ATTRIBUTE> --value <KEY_ATTRIBUTE_VALUE>

Options:
      --cluster-id <CLUSTER_ID>         Unique Id to choose which of the clusters in
 the config file to run the operation against. If not provided, will fall back to the
 value provided when interactive mode was started, or error
      --filter [<FILTER>...]            Key reference (e.g. key-
reference=0xabc) or space separated list of key attributes in the form of
 attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a matching key to modify
      --name <KEY_ATTRIBUTE>            Name of attribute to be set
      --value <KEY_ATTRIBUTE_VALUE>...  Attribute value to be set
  -h, --help                            Print help
```

Ejemplo: configuración de un atributo clave

El siguiente ejemplo muestra cómo utilizar el comando key set-attribute para establecer la etiqueta.

Example

1. Utilice la clave con la etiqueta `my_key`, como se muestra a continuación:

```
aws-cloudhsm > key set-attribute --filter attr.label=my_key --name encrypt --value
 false
{
  "error_code": 0,
  "data": {
    "message": "Attribute set successfully"
  }
}
```

2. Utilice el comando key list para confirmar que el atributo `encrypt` ha cambiado:

```
aws-cloudhsm > key list --filter attr.label=my_key --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000006400ec",
        "key-info": {
          "key-owners": [
            {
              "username": "bob",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "my_key",
          "id": "",
          "check-value": "0x6bd9f7",
          "class": "secret-key",
          "encrypt": false,
          "decrypt": true,
```

```
          "token": true,
          "always-sensitive": true,
          "derive": true,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": true,
          "unwrap": true,
          "verify": true,
          "wrap": true,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```


Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<KEY_ATTRIBUTE>

Especifica el nombre del atributo de la clave.

Obligatorio: sí

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente para su eliminación.

Para obtener una lista de los atributos clave de la CLI de CloudHSM compatibles, consulte Atributos de clave de la CLI de CloudHSM

Obligatorio: no

<KEY_ATTRIBUTE_VALUE>

Especifica el valor del atributo de la clave.

Obligatorio: sí

<KEY_REFERENCE>

Representación hexadecimal o decimal de la clave (como el identificador de una clave).

Obligatorio: no

Temas relacionados de

- Uso de la CLI de CloudHSM para filtrar claves
- Atributos de clave de la CLI de CloudHSM

key share

El key share comando comparte una clave con otras CU AWS CloudHSM del clúster.

Solo el CU que ha creado la clave (y, por tanto, la posee) puede compartirla. Los usuarios con quien se comparte la clave pueden utilizar la clave en operaciones criptográficas, pero no pueden exportarla ni eliminarla ni tampoco pueden compartirla o dejar de compartirla con otros usuarios. Además, estos usuarios no pueden cambiar los atributos de clave.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key share
Share a key in the HSM cluster with another user

Usage: key share --filter [<FILTER>...] --username <USERNAME> --role <ROLE>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error

      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 matching key for sharing

      --username <USERNAME>
          A username with which the key will be shared

      --role <ROLE>
          Role the user has in the cluster

          Possible values:
          - crypto-user: A CryptoUser has the ability to manage and use keys
          - admin:       An Admin has the ability to manage user accounts

  -h, --help
          Print help (see a summary with '-h')
```

Ejemplo: compartir una clave con otro CU

El siguiente ejemplo muestra cómo usar el comando key share para compartir una clave con el CU `alice`.

Example

1. Ejecute el comando key share para compartir la clave con `alice`.

```
aws-cloudhsm > key share --filter attr.label="rsa_key_to_share" attr.class=private-
key --username alice --role crypto-user
{
```

```
  "error_code": 0,
  "data": {
    "message": "Key shared successfully"
  }
}
```

2. Ejecute el comando key list.

```
aws-cloudhsm > key list --filter attr.label="rsa_key_to_share" attr.class=private-
key --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000001c0686",
        "key-info": {
          "key-owners": [
            {
              "username": "cu3",
              "key-coverage": "full"
            }
          ],
          "shared-users": [
            {
              "username": "cu2",
              "key-coverage": "full"
            },
            {
              "username": "cu1",
              "key-coverage": "full"
            },
            {
              "username": "cu4",
              "key-coverage": "full"
            },
            {
              "username": "cu5",
              "key-coverage": "full"
            },
            {
              "username": "cu6",
              "key-coverage": "full"
            },
```

```
            {
              "username": "cu7",
              "key-coverage": "full"
            },
            {
              "username": "alice",
              "key-coverage": "full"
            }
          ],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "rsa",
          "label": "rsa_key_to_share",
          "id": "",
          "check-value": "0xae8ff0",
          "class": "private-key",
          "encrypt": false,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": true,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": true,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 1219,
          "public-exponent": "0x010001",
          "modulus":
"0xa8855cba933cec0c21a4df0450ec31675c024f3e65b2b215a53d2bda6dcd191f75729150b59b4d86df58254
          "modulus-size-bits": 2048
        }
      }
    ],
    "total_key_count": 1,
```

```
    "returned_key_count": 1
  }
}
```

3. En la lista anterior, compruebe que `alice` aparece en la lista de `shared-users`

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente para su eliminación.

Para obtener una lista de los atributos de clave compatibles, consulte Atributos de clave de la CLI de CloudHSM.

Obligatorio: sí

<USERNAME>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_). En este comando, el nombre de usuario no distingue entre mayúsculas y minúsculas; el nombre de usuario siempre se muestra en minúsculas.

Obligatorio: sí

<ROLE>

Especifica el rol asignado a este usuario. Este parámetro es obligatorio. Para obtener el rol del usuario, ejecute el comando user list. Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Más información sobre los usuarios de HSM.

Obligatorio: sí

Temas relacionados de

- Uso de la CLI de CloudHSM para filtrar claves
- Atributos de clave de la CLI de CloudHSM

key unshare

El key unshare comando deja de compartir una clave con otras CU del clúster AWS CloudHSM .

Solo el CU que ha creado la clave (y, por tanto, la posee) puede dejar de compartirla. Los usuarios con quien se comparte la clave pueden utilizar la clave en operaciones criptográficas, pero no pueden exportarla ni eliminarla ni tampoco pueden compartirla o dejar de compartirla con otros usuarios. Además, estos usuarios no pueden cambiar los atributos de clave.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key unshare
Unshare a key in the HSM cluster with another user

Usage: key unshare --filter [<FILTER>...] --username <USERNAME> --role <ROLE>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error

      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 matching key for unsharing
```

```
    --username <USERNAME>
        A username with which the key will be unshared

    --role <ROLE>
        Role the user has in the cluster

        Possible values:
        - crypto-user: A CryptoUser has the ability to manage and use keys
        - admin:       An Admin has the ability to manage user accounts

-h, --help
        Print help (see a summary with '-h')
```

Ejemplo: dejar de compartir una clave con otro CU

El siguiente ejemplo muestra cómo usar el comando key unshare para dejar de compartir una clave con el CU `alice`.

Example

1. Ejecute el comando key list y filtre la clave específica que desea dejar de compartir con `alice`.

```
aws-cloudhsm > key list --filter attr.label="rsa_key_to_share" attr.class=private-
key --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000001c0686",
        "key-info": {
          "key-owners": [
            {
              "username": "cu3",
              "key-coverage": "full"
            }
          ],
          "shared-users": [
            {
              "username": "cu2",
              "key-coverage": "full"
            },
            {
```

```
            "username": "cu1",
            "key-coverage": "full"
          },
          {
            "username": "cu4",
            "key-coverage": "full"
          },
          {
            "username": "cu5",
            "key-coverage": "full"
          },
          {
            "username": "cu6",
            "key-coverage": "full"
          },
          {
            "username": "cu7",
            "key-coverage": "full"
          },
          {
            "username": "alice",
            "key-coverage": "full"
          }
        ],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "rsa",
        "label": "rsa_key_to_share",
        "id": "",
        "check-value": "0xae8ff0",
        "class": "private-key",
        "encrypt": false,
        "decrypt": true,
        "token": true,
        "always-sensitive": true,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": true,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
```

```
          "sign": true,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": false,
          "wrap-with-trusted": false,
          "key-length-bytes": 1219,
          "public-exponent": "0x010001",
          "modulus":
 "0xa8855cba933cec0c21a4df0450ec31675c024f3e65b2b215a53d2bda6dcd191f75729150b59b4d86df58254
          "modulus-size-bits": 2048
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

2. Confirme que `alice` aparece en los resultados de `shared-users` y ejecute el comando key unshare para dejar de compartir la clave con `alice`.

```
aws-cloudhsm > key unshare --filter attr.label="rsa_key_to_share"
 attr.class=private-key --username alice --role crypto-user
{
  "error_code": 0,
  "data": {
    "message": "Key unshared successfully"
  }
}
```

3. Vuelva a ejecutar el comando `key list` para confirmar que la clave ha dejado de compartirse con `alice`.

```
aws-cloudhsm > key list --filter attr.label="rsa_key_to_share" attr.class=private-
key --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000001c0686",
        "key-info": {
```

```
          "key-owners": [
            {
              "username": "cu3",
              "key-coverage": "full"
            }
          ],
          "shared-users": [
            {
              "username": "cu2",
              "key-coverage": "full"
            },
            {
              "username": "cu1",
              "key-coverage": "full"
            },
            {
              "username": "cu4",
              "key-coverage": "full"
            },
            {
              "username": "cu5",
              "key-coverage": "full"
            },
            {
              "username": "cu6",
              "key-coverage": "full"
            },
            {
              "username": "cu7",
              "key-coverage": "full"
            },
          ],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "rsa",
          "label": "rsa_key_to_share",
          "id": "",
          "check-value": "0xae8ff0",
          "class": "private-key",
          "encrypt": false,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
```

```
            "derive": false,
            "destroyable": true,
            "extractable": true,
            "local": true,
            "modifiable": true,
            "never-extractable": false,
            "private": true,
            "sensitive": true,
            "sign": true,
            "trusted": false,
            "unwrap": true,
            "verify": false,
            "wrap": false,
            "wrap-with-trusted": false,
            "key-length-bytes": 1219,
            "public-exponent": "0x010001",
            "modulus":
 "0xa8855cba933cec0c21a4df0450ec31675c024f3e65b2b215a53d2bda6dcd191f75729150b59b4d86df58254
            "modulus-size-bits": 2048
          }
        }
      ],
      "total_key_count": 1,
      "returned_key_count": 1
    }
}
```


Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave coincidente para su eliminación.

Para obtener una lista de los atributos de clave compatibles, consulte Atributos de clave de la CLI de CloudHSM.

Obligatorio: sí

<USERNAME>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_). En este comando, el nombre de usuario no distingue entre mayúsculas y minúsculas; el nombre de usuario siempre se muestra en minúsculas.

Obligatorio: sí

<ROLE>

Especifica el rol asignado a este usuario. Este parámetro es obligatorio. Para obtener el rol del usuario, ejecute el comando user list. Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Más información sobre los usuarios de HSM.

Obligatorio: sí

Temas relacionados de

- Uso de la CLI de CloudHSM para filtrar claves
- Atributos de clave de la CLI de CloudHSM

desempaquetar llaves

El comando key unwrap principal de la CLI de CloudHSM importa una clave privada simétrica o asimétrica cifrada (empaquetada) de un archivo al HSM. Este comando está diseñado para importar claves cifradas empaquetadas por el envoltorio para llaves comando, pero también se puede usar para desempaquetar claves empaquetadas con otras herramientas. Sin embargo, en esas situaciones, le recomendamos que utilice las bibliotecas de software de PKCS #11 o JCE para desencapsular la clave.

- desempaquetar llaves aes-gcm
- abrir llaves aes-no-pad
- desempaquetador de teclas aes-pkcs5-pad
- abrir llaves aes-zero-pad
- desempaquetar llaves cloudhsm-aes-gcm
- desempaquetar claves rsa-aes

- desempaquetar claves rsa-oaep
- desempaquetar claves rsa-pkcs

desempaquetar llaves aes-gcm

El key unwrap aes-gcm comando desempaqueta una clave de carga útil en el clúster mediante la clave de empaquetado AES y el mecanismo de desempaquetado. `AES-GCM`

Las claves no empaquetadas se pueden usar de la misma manera que las claves generadas por. AWS CloudHSM Para indicar que no se generaron localmente, su `local` atributo se establece en. `false`

Para usar el key unwrap aes-gcm comando, debe tener la clave de empaquetado AES en su AWS CloudHSM clúster y su `unwrap` atributo debe estar establecido en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key unwrap aes-gcm
Usage: key unwrap aes-gcm [OPTIONS] --filter [<FILTER>...] --tag-length-
bits <TAG_LENGTH_BITS> --key-type-class <KEY_TYPE_CLASS> --label <LABEL> --iv <IV> <--
data-path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a key
 to unwrap with
```

```
      --data-path <DATA_PATH>
          Path to the binary file containing the wrapped key data
      --data <DATA>
          Base64 encoded wrapped key data
      --attributes [<UNWRAPPED_KEY_ATTRIBUTES>...]
          Space separated list of key attributes in the form of
 KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE for the unwrapped key
      --aad <AAD>
          Aes GCM Additional Authenticated Data (AAD) value, in hex
      --tag-length-bits <TAG_LENGTH_BITS>
          Aes GCM tag length in bits
      --key-type-class <KEY_TYPE_CLASS>
          Key type and class of wrapped key [possible values: aes, des3, ec-private,
 generic-secret, rsa-private]
      --label <LABEL>
          Label for the unwrapped key
      --session
          Creates a session key that exists only in the current session. The key cannot
 be recovered after the session ends
      --iv <IV>
          Initial value used to wrap the key, in hex
  -h, --help
          Print help
```

Ejemplos

En estos ejemplos se muestra cómo utilizar el key unwrap aes-gcm comando mediante una clave AES con el valor del `unwrap` atributo establecido en`true`.

Example Ejemplo: Separe una clave de carga útil de los datos clave empaquetados codificados en Base64

```
aws-cloudhsm > key unwrap aes-gcm --key-type-class aes --label aes-unwrapped
 --filter attr.label=aes-example --tag-length-bits 64  --aad 0x10 --iv
 0xf90613bb8e337ec0339aad21 --data xvslgrtg8kHzrvekny97tLSIeokpPwV8
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001808e4",
      "key-info": {
        "key-owners": [
          {
```

```
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Example Ejemplo: desempaquetar una clave de carga útil proporcionada a través de una ruta de datos

```
aws-cloudhsm > key unwrap aes-gcm --key-type-class aes --label aes-unwrapped
 --filter attr.label=aes-example --tag-length-bits 64  --aad 0x10 --iv
 0xf90613bb8e337ec0339aad21 --data-path payload-key.pem
```

```
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001808e4",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
```

```
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave con la que desempaquetarlos.

Obligatorio: sí

<DATA_PATH>

Ruta al archivo binario que contiene los datos clave empaquetados.

Obligatorio: Sí (a menos que se proporcione a través de datos codificados en Base64)

<DATA>

Datos clave empaquetados codificados en Base64.

Obligatorio: Sí (a menos que se proporcione a través de una ruta de datos)

<ATTRIBUTES>

Lista de atributos clave separados por espacios en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para la clave empaquetada.

Obligatorio: no

<AAD>

Aes el valor de datos autenticados adicionales (AAD) de GCM, en hexadecimal.

Obligatorio: no

<TAG_LENGTH_BITS>

Longitud de la etiqueta Aes GCM en bits.

Obligatorio: sí

<KEY_TYPE_CLASS>

Tipo de clave y clase de clave empaquetada [valores posibles:`aes`,`des3`,`ec-private`,`generic-secret`,`rsa-private`].

Obligatorio: sí

<LABEL>

Etiqueta para la llave sin empaquetar.

Obligatorio: sí

<SESSION>

Crea una clave de sesión que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Obligatorio: no

<IV>

Valor inicial utilizado para envolver la clave, en hexadecimal.

Obligatorio: no

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

abrir llaves aes-no-pad

El key unwrap aes-no-pad comando desempaqueta una clave de carga útil en el clúster mediante la clave de empaquetado AES y el mecanismo de desempaquetado. `AES-NO-PAD`

Las claves no empaquetadas se pueden usar de la misma manera que las claves generadas por. AWS CloudHSM Para indicar que no se generaron localmente, su `local` atributo se establece en. `false`

Para usar el key unwrap aes-no-pad comando, debe tener la clave de empaquetado AES en su AWS CloudHSM clúster y su `unwrap` atributo debe estar establecido en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key unwrap aes-no-pad
Usage: key unwrap aes-no-pad [OPTIONS] --filter [<FILTER>...] --key-type-
class <KEY_TYPE_CLASS> --label <LABEL> <--data-path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a key
 to unwrap with
      --data-path <DATA_PATH>
          Path to the binary file containing the wrapped key data
      --data <DATA>
          Base64 encoded wrapped key data
      --attributes [<UNWRAPPED_KEY_ATTRIBUTES>...]
          Space separated list of key attributes in the form of
 KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE for the unwrapped key
      --key-type-class <KEY_TYPE_CLASS>
          Key type and class of wrapped key [possible values: aes, des3, ec-private,
 generic-secret, rsa-private]
      --label <LABEL>
          Label for the unwrapped key
      --session
          Creates a session key that exists only in the current session. The key cannot
 be recovered after the session ends
  -h, --help
          Print help
```

Ejemplos

Estos ejemplos muestran cómo utilizar el key unwrap aes-no-pad comando mediante una clave AES con el valor del `unwrap` atributo establecido en`true`.

Example Ejemplo: separe una clave de carga útil de los datos clave empaquetados codificados en Base64

```
aws-cloudhsm > key unwrap aes-no-pad --key-type-class aes --label aes-unwrapped --
filter attr.label=aes-example --data eXK3PMAOnKM9y3YX6brbhtMoC060EOH9
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001c08ec",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
```

```
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Example Ejemplo: desempaquetar una clave de carga útil proporcionada a través de una ruta de datos

```
aws-cloudhsm > key unwrap aes-no-pad --key-type-class aes --label aes-unwrapped --
filter attr.label=aes-example --data-path payload-key.pem
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001c08ec",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
```

```
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave con la que desempaquetarlos.

Obligatorio: sí

<DATA_PATH>

Ruta al archivo binario que contiene los datos clave empaquetados.

Obligatorio: Sí (a menos que se proporcione a través de datos codificados en Base64)

<DATA>

Datos clave empaquetados codificados en Base64.

Obligatorio: Sí (a menos que se proporcione a través de una ruta de datos)

<ATTRIBUTES>

Lista de atributos clave separados por espacios en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para la clave empaquetada.

Obligatorio: no

<KEY_TYPE_CLASS>

Tipo de clave y clase de clave empaquetada [valores posibles:`aes`,`des3`,`ec-private`,`generic-secret`,`rsa-private`].

Obligatorio: sí

<LABEL>

Etiqueta para la llave sin empaquetar.

Obligatorio: sí

<SESSION>

Crea una clave de sesión que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Obligatorio: no

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

desempaquetador de teclas aes-pkcs5-pad

El key unwrap aes-pkcs5-pad comando desempaqueta una clave de carga útil mediante la clave de empaquetado AES y el mecanismo de desempaquetado. `AES-PKCS5-PAD`

Las claves no empaquetadas se pueden usar de la misma manera que las claves generadas por. AWS CloudHSM Para indicar que no se generaron localmente, su `local` atributo se establece en. `false`

Para usar el key unwrap aes-pkcs5-pad comando, debe tener la clave de empaquetado AES en su AWS CloudHSM clúster y su `unwrap` atributo debe estar establecido en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key unwrap aes-pkcs5-pad
Usage: key unwrap aes-pkcs5-pad [OPTIONS] --filter [<FILTER>...] --key-type-
class <KEY_TYPE_CLASS> --label <LABEL> <--data-path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a key
 to unwrap with
      --data-path <DATA_PATH>
          Path to the binary file containing the wrapped key data
      --data <DATA>
          Base64 encoded wrapped key data
      --attributes [<UNWRAPPED_KEY_ATTRIBUTES>...]
          Space separated list of key attributes in the form of
 KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE for the unwrapped key
      --key-type-class <KEY_TYPE_CLASS>
          Key type and class of wrapped key [possible values: aes, des3, ec-private,
 generic-secret, rsa-private]
      --label <LABEL>
          Label for the unwrapped key
      --session
          Creates a session key that exists only in the current session. The key cannot
 be recovered after the session ends
  -h, --help
          Print help
```

Ejemplos

En estos ejemplos se muestra cómo utilizar el key unwrap aes-pkcs5-pad comando mediante una clave AES con el valor del `unwrap` atributo establecido en`true`.

Example Ejemplo: Separe una clave de carga útil de los datos clave empaquetados codificados en Base64

```
aws-cloudhsm > key unwrap aes-pkcs5-pad --key-type-class aes --label aes-unwrapped --
filter attr.label=aes-example --data MbuYNresfOKyGNnxKWen88nSfX+uUE/0qmGofSisicY=
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001c08e3",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
```

```
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Example Ejemplo: desempaquetar una clave de carga útil proporcionada a través de una ruta de datos

```
aws-cloudhsm > key unwrap aes-pkcs5-pad --key-type-class aes --label aes-unwrapped --
filter attr.label=aes-example --data-path payload-key.pem
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001c08e3",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
```

```
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave con la que desempaquetarlos.

Obligatorio: sí

<DATA_PATH>

Ruta al archivo binario que contiene los datos clave empaquetados.

Obligatorio: Sí (a menos que se proporcione a través de datos codificados en Base64)

<DATA>

Datos clave empaquetados codificados en Base64.

Obligatorio: Sí (a menos que se proporcione a través de una ruta de datos)

<ATTRIBUTES>

Lista de atributos clave separados por espacios en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para la clave empaquetada.

Obligatorio: no

<KEY_TYPE_CLASS>

Tipo de clave y clase de clave empaquetada [valores posibles:`aes`,`des3`,`ec-private`,`generic-secret`,`rsa-private`].

Obligatorio: sí

<LABEL>

Etiqueta para la llave sin empaquetar.

Obligatorio: sí

<SESSION>

Crea una clave de sesión que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Obligatorio: no

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

abrir llaves aes-zero-pad

El key unwrap aes-zero-pad comando desempaqueta una clave de carga útil en el clúster mediante la clave de empaquetado AES y el mecanismo de desempaquetado. `AES-ZERO-PAD`

Las claves no empaquetadas se pueden usar de la misma manera que las claves generadas por. AWS CloudHSM Para indicar que no se generaron localmente, su `local` atributo se establece en. `false`

Para usar el key unwrap aes-no-pad comando, debe tener la clave de empaquetado AES en su AWS CloudHSM clúster y su `unwrap` atributo debe estar establecido en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key unwrap aes-zero-pad
Usage: key unwrap aes-zero-pad [OPTIONS] --filter [<FILTER>...] --key-type-
class <KEY_TYPE_CLASS> --label <LABEL> <--data-path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a key
 to unwrap with
      --data-path <DATA_PATH>
          Path to the binary file containing the wrapped key data
      --data <DATA>
          Base64 encoded wrapped key data
      --attributes [<UNWRAPPED_KEY_ATTRIBUTES>...]
          Space separated list of key attributes in the form of
 KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE for the unwrapped key
      --key-type-class <KEY_TYPE_CLASS>
          Key type and class of wrapped key [possible values: aes, des3, ec-private,
 generic-secret, rsa-private]
      --label <LABEL>
          Label for the unwrapped key
      --session
          Creates a session key that exists only in the current session. The key cannot
 be recovered after the session ends
  -h, --help
          Print help
```

Ejemplos

Estos ejemplos muestran cómo utilizar el key unwrap aes-zero-pad comando mediante una clave AES con el valor del `unwrap` atributo establecido en`true`.

Example Ejemplo: separe una clave de carga útil de los datos clave empaquetados codificados en Base64

```
aws-cloudhsm > key unwrap aes-zero-pad --key-type-class aes --label aes-unwrapped --
filter attr.label=aes-example --data L1wVlL/YeBNVAw6Mpk3owFJZXBzDLONt
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001c08e7",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
```

```
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Example Ejemplo: desempaquetar una clave de carga útil proporcionada a través de una ruta de datos

```
aws-cloudhsm > key unwrap aes-zero-pad --key-type-class aes --label aes-unwrapped --
filter attr.label=aes-example --data-path payload-key.pem
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001c08e7",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
```

```
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave con la que desempaquetarlos.

Obligatorio: sí

<DATA_PATH>

Ruta al archivo binario que contiene los datos clave empaquetados.

Obligatorio: Sí (a menos que se proporcione a través de datos codificados en Base64)

<DATA>

Datos clave empaquetados codificados en Base64.

Obligatorio: Sí (a menos que se proporcione a través de una ruta de datos)

<ATTRIBUTES>

Lista de atributos clave separados por espacios en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para la clave empaquetada.

Obligatorio: no

<KEY_TYPE_CLASS>

Tipo de clave y clase de clave empaquetada [valores posibles:`aes`,`des3`,`ec-private`,`generic-secret`,`rsa-private`].

Obligatorio: sí

<LABEL>

Etiqueta para la llave sin empaquetar.

Obligatorio: sí

<SESSION>

Crea una clave de sesión que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Obligatorio: no

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

desempaquetar llaves cloudhsm-aes-gcm

El key unwrap cloudhsm-aes-gcm comando desempaqueta una clave de carga útil en el clúster mediante la clave de empaquetado AES y el mecanismo de desempaquetado. `CLOUDHSM-AES-GCM`

Las claves no empaquetadas se pueden usar de la misma manera que las claves generadas por. AWS CloudHSM Para indicar que no se generaron localmente, su `local` atributo se establece en. `false`

Para usar el key unwrap cloudhsm-aes-gcm comando, debe tener la clave de empaquetado AES en su AWS CloudHSM clúster y su `unwrap` atributo debe estar establecido en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key unwrap cloudhsm-aes-gcm
Usage: key unwrap cloudhsm-aes-gcm [OPTIONS] --filter [<FILTER>...] --tag-length-
bits <TAG_LENGTH_BITS> --key-type-class <KEY_TYPE_CLASS> --label <LABEL> <--data-
path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a key
 to unwrap with
      --data-path <DATA_PATH>
          Path to the binary file containing the wrapped key data
      --data <DATA>
          Base64 encoded wrapped key data
      --attributes [<UNWRAPPED_KEY_ATTRIBUTES>...]
          Space separated list of key attributes in the form of
 KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE for the unwrapped key
      --aad <AAD>
          Aes GCM Additional Authenticated Data (AAD) value, in hex
      --tag-length-bits <TAG_LENGTH_BITS>
          Aes GCM tag length in bits
      --key-type-class <KEY_TYPE_CLASS>
          Key type and class of wrapped key [possible values: aes, des3, ec-private,
 generic-secret, rsa-private]
      --label <LABEL>
          Label for the unwrapped key
```

```
      --session
          Creates a session key that exists only in the current session. The key cannot
 be recovered after the session ends
  -h, --help
          Print help
```

Ejemplos

En estos ejemplos se muestra cómo utilizar el key unwrap cloudhsm-aes-gcm comando mediante una clave AES con el valor del `unwrap` atributo establecido en`true`.

Example Ejemplo: Separe una clave de carga útil de los datos clave empaquetados codificados en Base64

```
aws-cloudhsm > key unwrap cloudhsm-aes-gcm --key-type-class aes --label aes-
unwrapped --filter attr.label=aes-example --tag-length-bits 64  --aad 0x10 --data
 6Rn8nkjEriDYlnP3P8nPkYQ8hplOEJ899zsrF+aTB0i/fIlZ
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001408e8",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
```

```
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Example Ejemplo: desempaquetar una clave de carga útil proporcionada a través de una ruta de datos

```
aws-cloudhsm > key unwrap cloudhsm-aes-gcm --key-type-class aes --label aes-unwrapped
 --filter attr.label=aes-example --tag-length-bits 64  --aad 0x10 --data-path payload-
key.pem
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001408e8",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
```

```
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave con la que desempaquetarlos.

Obligatorio: sí

<DATA_PATH>

Ruta al archivo binario que contiene los datos clave empaquetados.

Obligatorio: Sí (a menos que se proporcione a través de datos codificados en Base64)

<DATA>

Datos clave empaquetados codificados en Base64.

Obligatorio: Sí (a menos que se proporcione a través de una ruta de datos)

<ATTRIBUTES>

Lista de atributos clave separados por espacios en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para la clave empaquetada.

Obligatorio: no

<AAD>

Aes el valor de datos autenticados adicionales (AAD) de GCM, en hexadecimal.

Obligatorio: no

<TAG_LENGTH_BITS>

Longitud de la etiqueta Aes GCM en bits.

Obligatorio: sí

<KEY_TYPE_CLASS>

Tipo de clave y clase de clave empaquetada [valores posibles:`aes`,`des3`,`ec-private`,`generic-secret`,`rsa-private`].

Obligatorio: sí

<LABEL>

Etiqueta para la llave sin empaquetar.

Obligatorio: sí

<SESSION>

Crea una clave de sesión que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Obligatorio: no

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

desempaquetar claves rsa-aes

El key unwrap rsa-aes comando desempaqueta una clave de carga útil mediante una clave privada RSA y el mecanismo de desempaquetado. `RSA-AES`

Las claves no empaquetadas se pueden usar de la misma manera que las claves generadas por. AWS CloudHSM Para indicar que no se generaron localmente, su `local` atributo se establece en. `false`

Para utilizarloskey unwrap rsa-aes, debe tener la clave privada RSA de la clave de empaquetado pública de RSA en su AWS CloudHSM clúster y su `unwrap` atributo debe estar establecido en. `true`

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key unwrap rsa-aes
Usage: key unwrap rsa-aes [OPTIONS] --filter [<FILTER>...] --hash-
function <HASH_FUNCTION> --mgf <MGF> --key-type-class <KEY_TYPE_CLASS> --label <LABEL>
 <--data-path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
```

```
        Unique Id to choose which of the clusters in the config file to run the
operation against. If not provided, will fall back to the value provided when
interactive mode was started, or error
     --filter [<FILTER>...]
         Key reference (e.g. key-reference=0xabc) or space separated list of key
attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a key
to unwrap with
     --data-path <DATA_PATH>
         Path to the binary file containing the wrapped key data
     --data <DATA>
         Base64 encoded wrapped key data
     --attributes [<UNWRAPPED_KEY_ATTRIBUTES>...]
         Space separated list of key attributes in the form of
KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE for the unwrapped key
     --hash-function <HASH_FUNCTION>
         Hash algorithm [possible values: sha1, sha224, sha256, sha384, sha512]
     --mgf <MGF>
         Mask Generation Function algorithm [possible values: mgf1-sha1, mgf1-sha224,
mgf1-sha256, mgf1-sha384, mgf1-sha512]
     --key-type-class <KEY_TYPE_CLASS>
         Key type and class of wrapped key [possible values: aes, des3, ec-private,
generic-secret, rsa-private]
     --label <LABEL>
         Label for the unwrapped key
     --session
         Creates a session key that exists only in the current session. The key cannot
be recovered after the session ends
 -h, --help
         Print help
```

Ejemplo

En estos ejemplos se muestra cómo utilizar el key unwrap rsa-aes comando mediante la clave privada de RSA con el valor del `unwrap` atributo establecido en. `true`

Example Ejemplo: separe una clave de carga útil de los datos de clave empaquetados codificados en Base64

```
aws-cloudhsm > key unwrap rsa-aes --key-type-class aes --label aes-unwrapped
 --filter attr.label=rsa-private-key-example --hash-function sha256 --
mgf mgf1-sha256 --data HrSE1DEyLjIeyGdPa9R+ebiqB5TIJGyamPker31ZebPwRA
+NcerbAJO8DJ1lXPygZcI21vIFSZJuWMEiWpe1R9D/5WSYgxLVKex30xCFqebtEzxbKuv4DOmU4meSofqREYvtb3EoIKwjy
```

```
+RL5WGXKe4nAboAkC5GO7veI5yHL1SaKlssSJtTL/CFpbSLsAFuYbv/NUCWwMY5mwyVTCSlw+HlgKK
+5TH1MzBaSi8fpfyepLT8sHy2Q/VRl6ifb49p6m0KQFbRVvz/OWUd6l4d97BdgtaEz6ueg==
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001808e2",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
```

```
  }
}
```

Example Ejemplo: desempaquetar una clave de carga útil proporcionada a través de una ruta de datos

```
aws-cloudhsm > key unwrap rsa-aes --key-type-class aes --label aes-unwrapped --filter
 attr.label=rsa-private-key-example --hash-function sha256 --mgf mgf1-sha256 --data-
path payload-key.pem
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001808e2",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
```

```
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave con la que desempaquetarlos.

Obligatorio: sí

<DATA_PATH>

Ruta al archivo binario que contiene los datos clave empaquetados.

Obligatorio: Sí (a menos que se proporcione a través de datos codificados en Base64)

<DATA>

Datos clave empaquetados codificados en Base64.

Obligatorio: Sí (a menos que se proporcione a través de una ruta de datos)

<ATTRIBUTES>

Lista de atributos clave separados por espacios en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para la clave empaquetada.

Obligatorio: no

<KEY_TYPE_CLASS>

Tipo de clave y clase de clave empaquetada [valores posibles:`aes`,`des3`,`ec-private`,`generic-secret`,`rsa-private`].

Obligatorio: sí

<HASH_FUNCTION>

Especifica la función hash.

Valores válidos:

- sha1
- sha224
- sha256
- sha384
- sha512

Obligatorio: sí

<MGF>

Especifica la función de generación de máscaras.

> **Note**
>
> La función hash de la función de generación de máscaras debe coincidir con la función hash del mecanismo de firma.

Valores válidos:

- mgf1-sha1
- mgf1-sha224
- mgf1-sha256
- mgf1-sha384
- mgf1-sha512

Obligatorio: sí

<LABEL>

Etiqueta para la llave sin envolver.

Obligatorio: sí

<SESSION>

Crea una clave de sesión que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Obligatorio: no

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

desempaquetar claves rsa-oaep

El key unwrap rsa-oaep comando desempaqueta una clave de carga mediante la clave privada RSA y el mecanismo de desempaquetado. `RSA-OAEP`

Las claves no empaquetadas se pueden usar de la misma manera que las claves generadas por. AWS CloudHSM Para indicar que no se generaron localmente, su `local` atributo se establece en. `false`

Para usar el key unwrap rsa-oaep comando, debe tener la clave privada RSA de la clave de empaquetado pública de RSA en su AWS CloudHSM clúster y su `unwrap` atributo debe estar establecido en. `true`

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key unwrap rsa-oaep
Usage: key unwrap rsa-oaep [OPTIONS] --filter [<FILTER>...] --hash-
function <HASH_FUNCTION> --mgf <MGF> --key-type-class <KEY_TYPE_CLASS> --label <LABEL>
 <--data-path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a key
 to unwrap with
      --data-path <DATA_PATH>
          Path to the binary file containing the wrapped key data
      --data <<DATA>
          Base64 encoded wrapped key data
      --attributes [<UNWRAPPED_KEY_ATTRIBUTES>...]
          Space separated list of key attributes in the form of
 KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE for the unwrapped key
      --hash-function <HASH_FUNCTION>
          Hash algorithm [possible values: sha1, sha224, sha256, sha384, sha512]
      --mgf <MGF>
          Mask Generation Function algorithm [possible values: mgf1-sha1, mgf1-sha224,
 mgf1-sha256, mgf1-sha384, mgf1-sha512]
      --key-type-class <KEY_TYPE_CLASS>
          Key type and class of wrapped key [possible values: aes, des3, ec-private,
 generic-secret, rsa-private]
      --label <LABEL>
          Label for the unwrapped key
      --session
          Creates a session key that exists only in the current session. The key cannot
 be recovered after the session ends
  -h, --help
          Print help
```

Ejemplos

En estos ejemplos se muestra cómo utilizar el key unwrap rsa-oaep comando con la clave privada de RSA con el valor del `unwrap` atributo establecido en. `true`

Example Ejemplo: separe una clave de carga útil de los datos de clave empaquetados codificados en Base64

```
aws-cloudhsm > key unwrap rsa-oaep --key-type-class aes --label aes-unwrapped --filter
 attr.label=rsa-private-example-key --hash-function sha256 --mgf mgf1-sha256 --data
 OjJe4msobPLz9TuSAdULEu17T5rMDWtSlLyBSkLbaZnYzzpdrhsbGLbwZJCtB/jGkDNdB4qyTAOQwEpggGf6v
+Yx6JcesNeKKNU8XZal/YBoHC8noTGUSDI2qr+u2tDc84NPv6d+F2KOONXsSxMhmxzzNG/
gzTVIJhOuy/B1yHjGP4mOXoDZf5+7f5M1CjxBmz4Vva/wrWHGCSG0yOaWblEvOiHAIt3UBdyKmU+/
My4xjfJv7WGGu3DFUUIZ06TihRtKQhUYU1M9u6NPf9riJJfHsk6QCuSZ9yWThDT9as6i7e3htnyDhIhGWaoK8JU855cN/
YNKAUqkNpC4FPL3iw==
{
  "data": {
    "key": {
      "key-reference": "0x00000000001808e9",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
```

```
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Example Ejemplo: desempaquetar una clave de carga útil proporcionada a través de una ruta de datos

```
aws-cloudhsm > key unwrap rsa-oaep --key-type-class aes --label aes-unwrapped --filter
 attr.label=rsa-private-example-key --hash-function sha256 --mgf mgf1-sha256 --data-
path payload-key.pem
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001808e9",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
```

```
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave con la que desempaquetarlos.

Obligatorio: sí

<DATA_PATH>

Ruta al archivo binario que contiene los datos clave empaquetados.

Obligatorio: Sí (a menos que se proporcione a través de datos codificados en Base64)

<DATA>

Datos clave empaquetados codificados en Base64.

Obligatorio: Sí (a menos que se proporcione a través de una ruta de datos)

<ATTRIBUTES>

Lista de atributos clave separados por espacios en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para la clave empaquetada.

Obligatorio: no

<KEY_TYPE_CLASS>

Tipo de clave y clase de clave empaquetada [valores posibles:`aes`,`des3`,`ec-private`,`generic-secret`,`rsa-private`].

Obligatorio: sí

<HASH_FUNCTION>

Especifica la función hash.

Valores válidos:

- sha1
- sha224
- sha256
- sha384
- sha512

Obligatorio: sí

<MGF>

Especifica la función de generación de máscaras.

> **Note**
>
> La función hash de la función de generación de máscaras debe coincidir con la función hash del mecanismo de firma.

Valores válidos:

- mgf1-sha1
- mgf1-sha224
- mgf1-sha256
- mgf1-sha384
- mgf1-sha512

Obligatorio: sí

<LABEL>

Etiqueta para la llave sin envolver.

Obligatorio: sí

<SESSION>

Crea una clave de sesión que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Obligatorio: no

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

desempaquetar claves rsa-pkcs

El key unwrap rsa-pkcs comando desempaqueta una clave de carga mediante la clave privada RSA y el mecanismo de desempaquetado. `RSA-PKCS`

Las claves no empaquetadas se pueden usar de la misma manera que las claves generadas por. AWS CloudHSM Para indicar que no se generaron localmente, su `local` atributo se establece en. `false`

Para usar el unwrap rsa-pkcs comando de teclado, debe tener la clave privada RSA de la clave de empaquetado pública de RSA en su AWS CloudHSM clúster y su `unwrap` atributo debe estar establecido en. `true`

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key unwrap rsa-pkcs
Usage: key unwrap rsa-pkcs [OPTIONS] --filter [<FILTER>...] --key-type-
class <KEY_TYPE_CLASS> --label <LABEL> <--data-path <DATA_PATH>|--data <DATA>>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --filter [<FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a key
 to unwrap with
      --data-path <DATA_PATH>
          Path to the binary file containing the wrapped key data
      --data <DATA>
          Base64 encoded wrapped key data
      --attributes [<UNWRAPPED_KEY_ATTRIBUTES>...]
          Space separated list of key attributes in the form of
 KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE for the unwrapped key
      --key-type-class <KEY_TYPE_CLASS>
          Key type and class of wrapped key [possible values: aes, des3, ec-private,
 generic-secret, rsa-private]
      --label <LABEL>
          Label for the unwrapped key
      --session
          Creates a session key that exists only in the current session. The key cannot
 be recovered after the session ends
  -h, --help
          Print help
```

Ejemplos

En estos ejemplos se muestra cómo utilizar el key unwrap rsa-oaep comando mediante una clave AES con el valor del `unwrap` atributo establecido en. `true`

Example Ejemplo: separe una clave de carga útil de los datos clave empaquetados codificados en Base64

```
aws-cloudhsm > key unwrap rsa-pkcs --key-type-class aes --label
 aes-unwrapped --filter attr.label=rsa-private-key-example --data
 am0Nc7+YE8FWs+5HvU7sIBcXVb24QA0l65nbNAD+1bK+e18BpSfnaI3P+r8Dp+pLu1ofoUy/
vtzRjZoCiDofcz4EqCFnGl4GdcJ1/3W/5WRvMatCa2d7cx02swaeZcjKsermPXYRO1lGlfq6NskwMeeTkV8R7Rx9artFrs1
c3XdFJ2+0Bo94c6og/
yfPcpOOobJlITCoXhtMRepSdO4OggYq/6nUDuHCtJ86pPGnNahyr7+sAaSI3a5ECQLUjwaIARUCyoRh7EFK3qPXcg==
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001c08ef",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
```

```
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Example Ejemplo: desempaquetar una clave de carga útil proporcionada a través de una ruta de datos

```
aws-cloudhsm > key unwrap rsa-pkcs --key-type-class aes --label aes-unwrapped --filter
 attr.label=rsa-private-key-example --data-path payload-key.pem
{
  "error_code": 0,
  "data": {
    "key": {
      "key-reference": "0x00000000001c08ef",
      "key-info": {
        "key-owners": [
          {
            "username": "cu1",
            "key-coverage": "full"
          }
        ],
        "shared-users": [],
        "cluster-coverage": "full"
      },
      "attributes": {
        "key-type": "aes",
        "label": "aes-unwrapped",
        "id": "0x",
        "check-value": "0x8d9099",
        "class": "secret-key",
        "encrypt": false,
```

```
        "decrypt": false,
        "token": true,
        "always-sensitive": false,
        "derive": false,
        "destroyable": true,
        "extractable": true,
        "local": false,
        "modifiable": true,
        "never-extractable": false,
        "private": true,
        "sensitive": true,
        "sign": true,
        "trusted": false,
        "unwrap": false,
        "verify": true,
        "wrap": false,
        "wrap-with-trusted": false,
        "key-length-bytes": 16
      }
    }
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave con la que desempaquetarlos.

Obligatorio: sí

<DATA_PATH>

Ruta al archivo binario que contiene los datos clave empaquetados.

Obligatorio: Sí (a menos que se proporcione a través de datos codificados en Base64)

<DATA>

Datos clave empaquetados codificados en Base64.

Obligatorio: Sí (a menos que se proporcione a través de una ruta de datos)

<ATTRIBUTES>

Lista de atributos clave separados por espacios en forma de `KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para la clave empaquetada.

Obligatorio: no

<KEY_TYPE_CLASS>

Tipo de clave y clase de clave empaquetada [valores posibles:`aes`,`des3`,`ec-private`,`generic-secret`,`rsa-private`].

Obligatorio: sí

<LABEL>

Etiqueta para la llave sin empaquetar.

Obligatorio: sí

<SESSION>

Crea una clave de sesión que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Obligatorio: no

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

envoltorio para llaves

El key wrap comando de la CLI de CloudHSM exporta una copia cifrada de una clave privada simétrica o asimétrica del HSM a un archivo. Cuando se ejecutakey wrap, se especifican dos cosas:

la clave que se va a exportar y el archivo de salida. La clave para exportar es una clave del HSM que cifrará (empaquetará) la clave que desee exportar.

El key wrap comando no elimina la clave del HSM ni impide su uso en operaciones criptográficas. Puede exportar la misma clave varias veces. Para volver a importar la clave cifrada al HSM, utilice. desempaquetar llaves Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios con los que se comparte la clave solo pueden utilizarla en operaciones criptográficas.

El key wrap comando consta de los siguientes subcomandos:

- envoltorio para llaves aes-gcm
- envoltorio para llaves aes-no-pad
- funda para llaves aes-pkcs5-pad
- envoltorio para llaves aes-zero-pad
- envoltorio para llaves cloudhsm-aes-gcm
- envoltorio para llaves rsa-aes
- envoltorio de llaves rsa-oaep
- envoltorio para llaves rsa-pkcs

envoltorio para llaves aes-gcm

El key wrap aes-gcm comando empaqueta una clave de carga mediante una clave AES en el HSM y en el mecanismo de empaquetado. `AES-GCM` El `extractable` atributo de la clave de carga útil debe estar establecido en. `true`

Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas.

Para usar el key wrap aes-gcm comando, primero debe tener una clave AES en el AWS CloudHSM clúster. Puede generar una clave AES para empaquetarla con el key generate-symmetric aes comando y el `wrap` atributo establecidos en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key wrap aes-gcm
Usage: key wrap aes-gcm [OPTIONS] --payload-filter [<PAYLOAD_FILTER>...] --wrapping-
filter [<WRAPPING_FILTER>...] --tag-length-bits <TAG_LENGTH_BITS>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --payload-filter [<PAYLOAD_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 payload key
      --wrapping-filter [<WRAPPING_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 wrapping key
      --path <PATH>
          Path to the binary file where the wrapped key data will be saved
      --aad <AAD>
          Aes GCM Additional Authenticated Data (AAD) value, in hex
      --tag-length-bits <TAG_LENGTH_BITS>
          Aes GCM tag length in bits
  -h, --help
          Print help
```

Ejemplo

En este ejemplo se muestra cómo utilizar el key wrap aes-gcm comando mediante una clave AES.

Example

```
aws-cloudhsm > key wrap aes-gcm --payload-filter attr.label=payload-key --wrapping-
filter attr.label=aes-example --tag-length-bits 64  --aad 0x10
{
  "error_code": 0,
```

```
  "data": {
    "payload_key_reference": "0x00000000001c08f1",
    "wrapping_key_reference": "0x00000000001c08ea",
    "iv": "0xf90613bb8e337ec0339aad21",
    "wrapped_key_data": "xvslgrtg8kHzrvekny97tLSIeokpPwV8"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PAYLOAD_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separada por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de carga útil.

Obligatorio: sí

<PATH>

Ruta al archivo binario donde se guardarán los datos clave empaquetados.

Obligatorio: no

<WRAPPING_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de ajuste.

Obligatorio: sí

<AAD>

Valor de datos autenticados adicionales (AAD) de AES GCM, en hexadecimal.

Obligatorio: no

<TAG_LENGTH_BITS>

Longitud de la etiqueta AES GCM en bits.

Obligatorio: sí

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

envoltorio para llaves aes-no-pad

El key wrap aes-no-pad comando empaqueta una clave de carga mediante una clave AES en el HSM y en el mecanismo de empaquetado. `AES-NO-PAD` El `extractable` atributo de la clave de carga útil debe estar establecido en. `true`

Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas.

Para usar el key wrap aes-no-pad comando, primero debe tener una clave AES en el AWS CloudHSM clúster. Puede generar una clave AES para empaquetarla mediante el key generate-symmetric aes comando y el `wrap` atributo establecidos en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key wrap aes-no-pad
Usage: key wrap aes-no-pad [OPTIONS] --payload-filter [<PAYLOAD_FILTER>...] --wrapping-
filter [<WRAPPING_FILTER>...]
```

```
Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --payload-filter [<PAYLOAD_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 payload key
      --wrapping-filter [<WRAPPING_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 wrapping key
      --path <PATH>
          Path to the binary file where the wrapped key data will be saved
  -h, --help
          Print help
```

Ejemplo

En este ejemplo se muestra cómo utilizar el key wrap aes-no-pad comando mediante una clave AES con el valor del `wrap` atributo establecido en`true`.

Example

```
aws-cloudhsm > key wrap aes-no-pad --payload-filter attr.label=payload-key --wrapping-
filter attr.label=aes-example
{
  "error_code": 0,
  "data": {
    "payload_key_reference": "0x00000000001c08f1",
    "wrapping_key_reference": "0x00000000001c08ea",
    "wrapped_key_data": "eXK3PMAOnKM9y3YX6brbhtMoC060EOH9"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PAYLOAD_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separada por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de carga útil.

Obligatorio: sí

<PATH>

Ruta al archivo binario donde se guardarán los datos clave empaquetados.

Obligatorio: no

<WRAPPING_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de ajuste.

Obligatorio: sí

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

funda para llaves aes-pkcs5-pad

El key wrap aes-pkcs5-pad comando empaqueta una clave de carga mediante una clave AES en el HSM y en el mecanismo de empaquetado. `AES-PKCS5-PAD` El `extractable` atributo de la clave de carga útil debe estar establecido en. `true`

Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas.

Para usar el key wrap aes-pkcs5-pad comando, primero debe tener una clave AES en el AWS CloudHSM clúster. Puede generar una clave AES para empaquetarla mediante el key generate-symmetric aes comando y el `wrap` atributo establecidos en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key wrap aes-pkcs5-pad
Usage: key wrap aes-pkcs5-pad [OPTIONS] --payload-filter [<PAYLOAD_FILTER>...] --
wrapping-filter [<WRAPPING_FILTER>...]

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --payload-filter [<PAYLOAD_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 payload key
      --wrapping-filter [<WRAPPING_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 wrapping key
      --path <PATH>
          Path to the binary file where the wrapped key data will be saved
  -h, --help
          Print help
```

Ejemplo

En este ejemplo se muestra cómo utilizar el key wrap aes-pkcs5-pad comando mediante una clave AES con el valor del `wrap` atributo establecido en`true`.

Example

```
aws-cloudhsm > key wrap aes-pkcs5-pad --payload-filter attr.label=payload-key --
wrapping-filter attr.label=aes-example
{
  "error_code": 0,
```

```
  "data": {
    "payload_key_reference": "0x00000000001c08f1",
    "wrapping_key_reference": "0x00000000001c08ea",
    "wrapped_key_data": "MbuYNresfOKyGNnxKWen88nSfX+uUE/0qmGofSisicY="
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PAYLOAD_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separada por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de carga útil.

Obligatorio: sí

<PATH>

Ruta al archivo binario donde se guardarán los datos clave empaquetados.

Obligatorio: no

<WRAPPING_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de ajuste.

Obligatorio: sí

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

envoltorio para llaves aes-zero-pad

El key wrap aes-zero-pad comando empaqueta una clave de carga mediante una clave AES en el HSM y en el mecanismo de empaquetado. `AES-ZERO-PAD` El `extractable` atributo de la clave de carga útil debe estar establecido en. `true`

Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas.

Para usar el key wrap aes-zero-pad comando, primero debe tener una clave AES en el AWS CloudHSM clúster. Puede generar una clave AES para empaquetarla mediante el key generate-symmetric aes comando con el `wrap` atributo establecido en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key wrap aes-zero-pad
Usage: key wrap aes-zero-pad [OPTIONS] --payload-filter [<PAYLOAD_FILTER>...] --
wrapping-filter [<WRAPPING_FILTER>...]

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --payload-filter [<PAYLOAD_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 payload key
      --wrapping-filter [<WRAPPING_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 wrapping key
```

```
      --path <PATH>
          Path to the binary file where the wrapped key data will be saved
  -h, --help
          Print help
```

Ejemplo

En este ejemplo se muestra cómo utilizar el key wrap aes-zero-pad comando mediante una clave AES con el valor del `wrap` atributo establecido en`true`.

Example

```
aws-cloudhsm > key wrap aes-zero-pad --payload-filter attr.label=payload-key --
wrapping-filter attr.label=aes-example
{
  "error_code": 0,
  "data": {
    "payload_key_reference": "0x00000000001c08f1",
    "wrapping_key_reference": "0x00000000001c08ea",
    "wrapped_key_data": "L1wVlL/YeBNVAw6Mpk3owFJZXBzDLONt"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PAYLOAD_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separada por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de carga útil.

Obligatorio: sí

<PATH>

Ruta al archivo binario donde se guardarán los datos clave empaquetados.

Obligatorio: no

<WRAPPING_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de ajuste.

Obligatorio: sí

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

envoltorio para llaves cloudhsm-aes-gcm

El key wrap cloudhsm-aes-gcm comando empaqueta una clave de carga mediante una clave AES en el HSM y en el mecanismo de empaquetado. `CLOUDHSM-AES-GCM` El `extractable` atributo de la clave de carga útil debe estar establecido en. `true`

Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas.

Para usar el key wrap cloudhsm-aes-gcm comando, primero debe tener una clave AES en el AWS CloudHSM clúster. Puede generar una clave AES para empaquetarla con el key generate-symmetric aes comando y el `wrap` atributo establecidos en`true`.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key wrap cloudhsm-aes-gcm
```

```
Usage: key wrap cloudhsm-aes-gcm [OPTIONS] --payload-filter [<PAYLOAD_FILTER>...] --
wrapping-filter [<WRAPPING_FILTER>...] --tag-length-bits <TAG_LENGTH_BITS>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --payload-filter [<PAYLOAD_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 payload key
      --wrapping-filter [<WRAPPING_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 wrapping key
      --path <PATH>
          Path to the binary file where the wrapped key data will be saved
      --aad <AAD>
          Aes GCM Additional Authenticated Data (AAD) value, in hex
      --tag-length-bits <TAG_LENGTH_BITS>
          Aes GCM tag length in bits
  -h, --help
          Print help
```

Ejemplo

En este ejemplo se muestra cómo utilizar el key wrap cloudhsm-aes-gcm comando mediante una clave AES.

Example

```
aws-cloudhsm > key wrap cloudhsm-aes-gcm --payload-filter attr.label=payload-key --
wrapping-filter attr.label=aes-example --tag-length-bits 64 --aad 0x10
{
  "error_code": 0,
  "data": {
    "payload_key_reference": "0x00000000001c08f1",
    "wrapping_key_reference": "0x00000000001c08ea",
    "wrapped_key_data": "6Rn8nkjEriDYlnP3P8nPkYQ8hplOEJ899zsrF+aTB0i/fIlZ"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PAYLOAD_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separada por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de carga útil.

Obligatorio: sí

<PATH>

Ruta al archivo binario donde se guardarán los datos clave empaquetados.

Obligatorio: no

<WRAPPING_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de ajuste.

Obligatorio: sí

<AAD>

Valor de datos autenticados adicionales (AAD) de AES GCM, en hexadecimal.

Obligatorio: no

<TAG_LENGTH_BITS>

Longitud de la etiqueta AES GCM en bits.

Obligatorio: sí

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

envoltorio para llaves rsa-aes

El key wrap rsa-aes comando empaqueta una clave de carga útil mediante una clave pública RSA en el HSM y el mecanismo de empaquetado RSA-AES. El atributo de la clave de carga útil debe estar establecido en. `extractable` `true`

Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas.

Para usar el key wrap rsa-aes comando, primero debe tener una clave RSA en el clúster AWS CloudHSM . Puede generar un par de claves RSA mediante el clave generate-asymmetric-pair comando y el `wrap` atributo establecidos en. `true`

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key wrap rsa-aes
Usage: key wrap rsa-aes [OPTIONS] --payload-filter [<PAYLOAD_FILTER>...] --wrapping-
filter [<WRAPPING_FILTER>...] --hash-function <HASH_FUNCTION> --mgf <MGF>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --payload-filter [<PAYLOAD_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 payload key
      --wrapping-filter [<WRAPPING_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 wrapping key
```

```
      --path <PATH>
          Path to the binary file where the wrapped key data will be saved
      --hash-function <HASH_FUNCTION>
          Hash algorithm [possible values: sha1, sha224, sha256, sha384, sha512]
      --mgf <MGF>
          Mask Generation Function algorithm [possible values: mgf1-sha1, mgf1-sha224,
 mgf1-sha256, mgf1-sha384, mgf1-sha512]
  -h, --help
          Print help
```

Ejemplo

En este ejemplo se muestra cómo utilizar el key wrap rsa-ae comando mediante una clave pública RSA con el valor del `wrap` atributo establecido en. `true`

Example

```
aws-cloudhsm > key wrap rsa-aes --payload-filter attr.label=payload-key --wrapping-
filter attr.label=rsa-public-key-example --hash-function sha256 --mgf mgf1-sha256
{
  "error_code": 0,
  "data": {
    "payload-key-reference": "0x00000000001c08f1",
    "wrapping-key-reference": "0x00000000007008da",
    "wrapped-key-data": "HrSE1DEyLjIeyGdPa9R+ebiqB5TIJGyamPker31ZebPwRA
+NcerbAJO8DJ1lXPygZcI21vIFSZJuWMEiWpe1R9D/5WSYgxLVKex30xCFqebtEzxbKuv4DOmU4meSofqREYvtb3EoIKwjy
+RL5WGXKe4nAboAkC5GO7veI5yHL1SaKlssSJtTL/CFpbSLsAFuYbv/NUCWwMY5mwyVTCSlw+HlgKK
+5TH1MzBaSi8fpfyepLT8sHy2Q/VRl6ifb49p6m0KQFbRVvz/OWUd6l4d97BdgtaEz6ueg=="
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PAYLOAD_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separada por espacios en forma de `attr.`KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE para seleccionar una clave de carga útil.

Obligatorio: sí

<PATH>

Ruta al archivo binario donde se guardarán los datos clave empaquetados.

Obligatorio: no

<WRAPPING_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de ajuste.

Obligatorio: sí

<MGF>

Especifica la función de generación de máscaras.

> **Note**
>
> La función hash de la función de generación de máscaras debe coincidir con la función hash del mecanismo de firma.

Valores válidos

- mgf1-sha1
- mgf1-sha224
- mgf1-sha256
- mgf1-sha384
- mgf1-sha512

Obligatorio: sí

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

envoltorio de llaves rsa-oaep

El key wrap rsa-oaep comando empaqueta una clave de carga mediante una clave pública RSA en el HSM y el mecanismo de empaquetado. `RSA-OAEP` El `extractable` atributo de la clave de carga útil debe estar establecido en. `true`

Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas.

Para usar el key wrap rsa-oaep comando, primero debe tener una clave RSA en el clúster AWS CloudHSM . Puede generar un par de claves RSA mediante el clave generate-asymmetric-pair comando y el `wrap` atributo establecidos en. `true`

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key wrap rsa-oaep
Usage: key wrap rsa-oaep [OPTIONS] --payload-filter [<PAYLOAD_FILTER>...] --wrapping-
filter [<WRAPPING_FILTER>...] --hash-function <HASH_FUNCTION> --mgf <MGF>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --payload-filter [<PAYLOAD_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 payload key
      --wrapping-filter [<WRAPPING_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 wrapping key
```

```
      --path <PATH>
          Path to the binary file where the wrapped key data will be saved
      --hash-function <HASH_FUNCTION>
          Hash algorithm [possible values: sha1, sha224, sha256, sha384, sha512]
      --mgf <MGF>
          Mask Generation Function algorithm [possible values: mgf1-sha1, mgf1-sha224,
 mgf1-sha256, mgf1-sha384, mgf1-sha512]
  -h, --help
          Print help
```

Ejemplo

En este ejemplo se muestra cómo utilizar el key wrap rsa-oaep comando mediante una clave pública RSA con el valor del `wrap` atributo establecido en. `true`

Example

```
aws-cloudhsm > key wrap rsa-oaep --payload-filter attr.label=payload-key --wrapping-
filter attr.label=rsa-public-key-example --hash-function sha256 --mgf mgf1-sha256
{
  "error_code": 0,
  "data": {
    "payload-key-reference": "0x00000000001c08f1",
    "wrapping-key-reference": "0x00000000007008da",
    "wrapped-key-data": "OjJe4msobPLz9TuSAdULEu17T5rMDWtSlLyBSkLbaZnYzzpdrhsbGLbwZJCtB/
jGkDNdB4qyTAOQwEpggGf6v+Yx6JcesNeKKNU8XZal/YBoHC8noTGUSDI2qr+u2tDc84NPv6d
+F2KOONXsSxMhmxzzNG/gzTVIJhOuy/B1yHjGP4mOXoDZf5+7f5M1CjxBmz4Vva/
wrWHGCSG0yOaWblEvOiHAIt3UBdyKmU+/
My4xjfJv7WGGu3DFUUIZ06TihRtKQhUYU1M9u6NPf9riJJfHsk6QCuSZ9yWThDT9as6i7e3htnyDhIhGWaoK8JU855cN/
YNKAUqkNpC4FPL3iw=="
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PAYLOAD_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separada por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de carga útil.

Obligatorio: sí

<PATH>

Ruta al archivo binario donde se guardarán los datos clave empaquetados.

Obligatorio: no

<WRAPPING_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de ajuste.

Obligatorio: sí

<MGF>

Especifica la función de generación de máscaras.

> **Note**
>
> La función hash de la función de generación de máscaras debe coincidir con la función hash del mecanismo de firma.

Valores válidos

- mgf1-sha1
- mgf1-sha224
- mgf1-sha256
- mgf1-sha384
- mgf1-sha512

Obligatorio: sí

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

envoltorio para llaves rsa-pkcs

El key wrap rsa-pkcs comando agrupa una clave de carga mediante una clave pública RSA en el HSM y el mecanismo de empaquetado. `RSA-PKCS` El `extractable` atributo de la clave de carga útil debe estar establecido en. `true`

Solo el propietario de una clave, es decir, el usuario criptográfico (CU) que creó la clave, puede empaquetarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas.

Para usar el key wrap rsa-pkcs comando, primero debe tener una clave RSA en el clúster AWS CloudHSM . Puede generar un par de claves RSA mediante el clave generate-asymmetric-pair comando y el `wrap` atributo establecidos en. `true`

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Requisitos

- Para ejecutar este comando, debe iniciar sesión como CU.

Sintaxis

```
aws-cloudhsm > help key wrap rsa-pkcs
Usage: key wrap rsa-pkcs [OPTIONS] --payload-filter [<PAYLOAD_FILTER>...] --wrapping-
filter [<WRAPPING_FILTER>...]

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
      --payload-filter [<PAYLOAD_FILTER>...]
```

```
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 payload key
      --wrapping-filter [<WRAPPING_FILTER>...]
          Key reference (e.g. key-reference=0xabc) or space separated list of key
 attributes in the form of attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE to select a
 wrapping key
      --path <PATH>
          Path to the binary file where the wrapped key data will be saved
  -h, --help
          Print help
```

Ejemplo

En este ejemplo, se muestra cómo utilizar el key wrap rsa-pkcs comando mediante una clave pública RSA.

Example

```
aws-cloudhsm > key wrap rsa-pkcs --payload-filter attr.label=payload-key --wrapping-
filter attr.label=rsa-public-key-example
{
  "error_code": 0,
  "data": {
    "payload_key_reference": "0x00000000001c08f1",
    "wrapping_key_reference": "0x00000000007008da",
    "wrapped_key_data": "am0Nc7+YE8FWs+5HvU7sIBcXVb24QA0l65nbNAD+1bK+e18BpSfnaI3P+r8Dp
+pLu1ofoUy/
vtzRjZoCiDofcz4EqCFnGl4GdcJ1/3W/5WRvMatCa2d7cx02swaeZcjKsermPXYRO1lGlfq6NskwMeeTkV8R7Rx9artFrs1
c3XdFJ2+0Bo94c6og/
yfPcpOOobJlITCoXhtMRepSdO4OggYq/6nUDuHCtJ86pPGnNahyr7+sAaSI3a5ECQLUjwaIARUCyoRh7EFK3qPXcg=="
  }
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PAYLOAD_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separada por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de carga útil.

Obligatorio: sí

<PATH>

Ruta al archivo binario donde se guardarán los datos clave empaquetados.

Obligatorio: no

<WRAPPING_FILTER>

Referencia clave (por ejemplo`key-reference=0xabc`) o lista de atributos clave separados por espacios en forma de `attr.KEY_ATTRIBUTE_NAME=KEY_ATTRIBUTE_VALUE` para seleccionar una clave de ajuste.

Obligatorio: sí

Temas relacionados de

- envoltorio para llaves
- desempaquetar llaves

login

Puede utilizar el comando login de la CLI de CloudHSM para iniciar y cerrar sesión en cada HSM de un clúster.

> **Note**
>
> Si se superan cinco intentos de inicio de sesión incorrectos, se bloquea la cuenta. Para desbloquear la cuenta, un administrador debe restablecer la contraseña mediante el comando user change-password de cloudhsm_cli.

Cómo solucionar problemas de inicio y cierre de sesión

Si tiene más de un HSM en el clúster, es posible que puedan realizarse intentos adicionales de inicio de sesión incorrectos antes de que se bloquee la cuenta. Esto se debe a que el cliente CloudHSM equilibra la carga entre los diversos HSM. Por lo tanto, el intento de inicio de sesión no puede comenzar en el mismo HSM cada vez. Si va a probar esta funcionalidad, recomendamos que lo haga en un clúster con un solo HSM activo.

Si creó el clúster antes de febrero de 2018, la cuenta se bloquea después de 20 intentos de inicio de sesión incorrectos.

Tipo de usuario

Los usuarios siguientes pueden ejecutar estos comandos.

- Administrador desactivado
- Administrador
- Usuario de criptografía (CU)

Sintaxis

```
aws-cloudhsm > help login
Login to your cluster

USAGE:
    cloudhsm-cli login [OPTIONS] --username <USERNAME> --role <ROLE> [COMMAND]

Commands:
  mfa-token-sign  Login with token-sign mfa
  help            Print this message or the help of the given subcommand(s)

OPTIONS:
        --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error

        --username <USERNAME>
            Username to access the Cluster

        --role <ROLE>
```

```
            Role the user has in the Cluster

           Possible values:
           - crypto-user: A CryptoUser has the ability to manage and use keys
           - admin:       An Admin has the ability to manage user accounts

         --password <PASSWORD>
            Optional: Plaintext user's password. If you do not include this argument you
 will be prompted for it

  -h, --help
          Print help (see a summary with '-h')
```

Ejemplo

Example

Con este comando, puede iniciar sesión en todos los HSM de un clúster con las credenciales de un usuario administrador llamado `admin1`.

```
aws-cloudhsm > login --username admin1 --role admin
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin1",
    "role": "admin"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<USERNAME>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_). En este comando, el nombre de usuario

no distingue entre mayúsculas y minúsculas; el nombre de usuario siempre se muestra en minúsculas.

Obligatorio: sí

<ROLE>

Especifica el rol asignado a este usuario. Este parámetro es obligatorio. Los valores válidos son admin y crypto-user.

Para obtener el rol del usuario, ejecute el comando user list. Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Entendiendo los usuarios de HSM .

<PASSWORD>

Especifica la contraseña del usuario que va a iniciar sesión en los HSM.

Temas relacionados de

- Introducción a la CLI de CloudHSM
- Activación del clúster

iniciar sesión mfa-token-sign

Utilice el comando login mfa-token-sign en el inicio de sesión de la CLI de CloudHSM de AWS CloudHSM mediante la autenticación multifactor. Para usar este comando, primero debe configurar la MFA para la CLI de CloudHSM.

Tipo de usuario

Los usuarios siguientes pueden ejecutar estos comandos.

- Administrador
- Usuario de criptografía (CU)

Sintaxis

```
aws-cloudhsm > help login mfa-token-sign
Login with token-sign mfa

USAGE:
    login --username <USERNAME> --role <ROLE> mfa-token-sign --token <TOKEN>
```

```
OPTIONS:
      --cluster-id <CLUSTER_ID>  Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
      --token <TOKEN>            Filepath where the unsigned token file will be written
  -h, --help                     Print help
```

Ejemplo

Example

```
aws-cloudhsm > login --username test_user --role admin mfa-token-sign --token /home/
valid.token
Enter password:
Enter signed token file path (press enter if same as the unsigned token file):
{
  "error_code": 0,
  "data": {
    "username": "test_user",
    "role": "admin"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<TOKEN>

Ruta de archivo en la que se escribirá el archivo de token sin firmar.

Obligatorio: sí

Temas relacionados de

- Introducción a la CLI de CloudHSM
- Activación del clúster
- Uso de la CLI de CloudHSM para gestionar la MFA

logout

Puede usar el comando logout de la CLI de CloudHSM para cerrar sesión en cada HSM de un clúster.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Administrador
- Usuario de criptografía (CU)

Sintaxis

```
aws-cloudhsm > help logout
Logout of your cluster

USAGE:
    logout

OPTIONS:
        --cluster-id <CLUSTER_ID> Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
    -h, --help                    Print help information
    -V, --version                 Print version information
```

Ejemplo

Example

Este comando cierra la sesión de todos los HSM de un clúster.

```
aws-cloudhsm > logout
{
  "error_code": 0,
  "data": "Logout successful"
}
```

Temas relacionados de

- Introducción a la CLI de CloudHSM

- Activación del clúster

usuario

user es una categoría principal para un grupo de comandos que, en combinación con la categoría principal, crean un comando específico para los usuarios. Actualmente, la categoría de usuario consta de los siguientes comandos:

- user change-mfa
- user change-password
- user create
- user delete
- user list

user change-mfa

Actualmente, esta categoría consta de los siguientes comandos:

- user change-mfa token-sign

user change-mfa token-sign

Utilice el comando user change-mfa de la CLI de CloudHSM para actualizar la configuración de la autenticación multifactor (MFA) de una cuenta de usuario. Cualquier cuenta de usuario puede ejecutar este comando. Las cuentas con el rol de administrador pueden ejecutar este comando para otros usuarios.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Administrador
- Usuario de criptografía

Sintaxis

Actualmente, solo hay una estrategia multifactorial disponible para los usuarios: firma de token.

```
aws-cloudhsm > help user change-mfa
Change a user's Mfa Strategy

Usage:
    user change-mfa <COMMAND>

Commands:
  token-sign  Register or Deregister a public key using token-sign mfa strategy
  help        Print this message or the help of the given subcommand(s)
```

Esta estrategia solicita un archivo de token donde escribir los tokens sin firmar.

```
aws-cloudhsm > help user change-mfa token-sign
Register or Deregister a public key using token-sign mfa strategy

Usage: user change-mfa token-sign [OPTIONS] --username <USERNAME> --role <ROLE> <--
token <TOKEN>|--deregister>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error

      --username <USERNAME>
          Username of the user that will be modified

      --role <ROLE>
          Role the user has in the cluster

          Possible values:
          - crypto-user: A CryptoUser has the ability to manage and use keys
          - admin:       An Admin has the ability to manage user accounts

      --change-password <CHANGE_PASSWORD>
          Optional: Plaintext user's password. If you do not include this argument you
 will be prompted for it

      --token <TOKEN>
          Filepath where the unsigned token file will be written. Required for enabling
 MFA for a user
```

```
      --approval <APPROVAL>
          Filepath of signed quorum token file to approve operation

      --deregister
          Deregister the MFA public key, if present

      --change-quorum
          Change the Quorum public key along with the MFA key

  -h, --help
          Print help (see a summary with '-h')
```

Ejemplo

Este comando escribirá un token sin firmar por cada HSM del clúster en el archivo especificado por `token`. Cuando se le solicite, firme los tokens del archivo.

Example : escriba un token sin firmar por cada HSM de su clúster.

```
aws-cloudhsm > user change-mfa token-sign --username cu1 --change-password password --
role crypto-user --token /path/myfile
Enter signed token file path (press enter if same as the unsigned token file):
Enter public key PEM file path:/path/mypemfile
{
  "error_code": 0,
  "data": {
    "username": "test_user",
    "role": "admin"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<ROLE>

Especifica la función asignada a la cuenta de usuario. Este parámetro es obligatorio. Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Entendiendo los usuarios de HSM .

Valores válidos

- Administrador: los administradores pueden administrar a los usuarios, pero no las claves.
- Usuario de criptografía: los usuarios de criptografía pueden crear y administrar claves, y usar claves en operaciones criptográficas.

<USERNAME>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_).

No puede cambiar el nombre de un usuario después de crearlo. En los comandos de la CLI de CloudHSM, el tipo de usuario y la contraseña distinguen entre mayúsculas y minúsculas, pero el nombre de usuario no.

Obligatorio: sí

<CHANGE_PASSWORD>

Especifica la nueva contraseña (en texto sin formato) del usuario cuya MFA se está registrando o anulando.

Obligatorio: sí

<TOKEN>

Ruta de archivo en la que se escribirá el archivo de token sin firmar.

Obligatorio: sí

<APPROVAL>

Especifica la ruta de archivo de token de cuórum firmado para aprobar la operación. Solo es obligatorio si el valor del cuórum del servicio de usuario es superior a 1.

<DEREGISTER>

Anula el registro de la clave pública de la MFA, si está presente.

<CHANGE-QUORUM>

Cambia la clave pública del cuórum junto con la clave de la MFA.

Temas relacionados de

- Descripción de la 2FA para los usuarios de HSM

user change-password

Use el user change-password comando de la CLI de CloudHSM para cambiar la contraseña de un usuario AWS CloudHSM existente en el clúster. Para habilitar la autenticación multifactor para un usuario, ejecute el comando `user change-mfa`.

Cualquier usuario puede cambiar su propia contraseña. Además, los usuarios con rol de administrador pueden cambiar la contraseña de otro usuario del clúster. No es necesario que escriba la contraseña actual para realizar el cambio.

> **Note**
>
> No puede cambiar la contraseña de un usuario que haya iniciado sesión en el clúster.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Administrador
- Usuario de criptografía (CU)

Sintaxis

> **Note**
>
> Para habilitar la autenticación multifactor (MFA) para un usuario, ejecute el comando user change-mfa.

```
aws-cloudhsm > help user change-password
Change a user's password

    Usage:
        cloudhsm-cli user change-password [OPTIONS] --username <USERNAME> --role <ROLE>
 [--password <PASSWORD>]

    Options:
      --cluster-id <CLUSTER_ID>
```

```
        Unique Id to choose which of the clusters in the config file to run the
operation against. If not provided, will fall back to the value provided when
interactive mode was started, or error

    --username <USERNAME>
        Username of the user that will be modified

    --role <ROLE>
        Role the user has in the cluster

        Possible values:
        - crypto-user: A CryptoUser has the ability to manage and use keys
        - admin:       An Admin has the ability to manage user accounts

    --password <PASSWORD>
        Optional: Plaintext user's password. If you do not include this argument you
will be prompted for it

    --approval <APPROVAL>
        Filepath of signed quorum token file to approve operation

    --deregister-mfa <DEREGISTER-MFA>
        Deregister the user's mfa public key, if present

    --deregister-quorum <DEREGISTER-QUORUM>
        Deregister the user's quorum public key, if present
-h, --help
        Print help (see a summary with '-h')
```

Ejemplo

Los siguientes ejemplos muestran cómo utilizar user change-password para restablecer la contraseña del usuario actual o de cualquier otro usuario del clúster.

Example : cambio de la contraseña

Cualquier usuario del clúster puede utilizar user change-password para cambiar su propia contraseña.

La siguiente salida muestra que Bob ha iniciado sesión como usuario de criptografía (CU).

```
aws-cloudhsm > user change-password --username bob --role crypto-user
Enter password:
```

```
Confirm password:
{
  "error_code": 0,
  "data": {
    "username": "bob",
    "role": "crypto-user"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<APPROVAL>

Especifica la ruta de archivo de token de cuórum firmado para aprobar la operación. Solo es obligatorio si el valor del cuórum del servicio de usuario es superior a 1.

<DEREGISTER-MFA>

Anula el registro de la clave pública de la MFA, si está presente.

<DEREGISTER-QUORUM>

Anula el registro de la clave pública de cuórum, si existe.

<PASSWORD>

Especifica la nueva contraseña del usuario en texto plano.

Obligatorio: sí

<ROLE>

Especifica la función asignada a la cuenta de usuario. Este parámetro es obligatorio. Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Entendiendo los usuarios de HSM .

Valores válidos

- Administrador: los administradores pueden administrar a los usuarios, pero no las claves.
- Usuario de criptografía: los usuarios de criptografía pueden crear y administrar claves, y usar claves en operaciones criptográficas.

<USERNAME>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_).

No puede cambiar el nombre de un usuario después de crearlo. En los comandos de la CLI de CloudHSM, el tipo de usuario y la contraseña distinguen entre mayúsculas y minúsculas, pero el nombre de usuario no.

Obligatorio: sí

Temas relacionados de

- user list
- user create
- user delete

change-quorum de usuario

user change-quorum es una categoría principal para un grupo de comandos que, en combinación con la categoría principal, crean un comando específico para cambiar el cuórum de los usuarios.

user change-quorum se utiliza para registrar la autenticación de cuórum de los usuarios mediante una estrategia de cuórum específica. A partir de SDK 5.8.0, solo hay una estrategia de cuórum disponible para los usuarios, como se muestra a continuación.

Actualmente, esta categoría consta de la categoría y el subcomando siguientes:

- token-sign
  - register

user change-quorum token-sign

user change-quorum token-sign es una categoría principal para comandos que, cuando se combinan con esta categoría principal, crean un comando específico para las operaciones de cuórum con firma de token.

Actualmente, esta categoría consta de los siguientes comandos:

- register

user change-quorum token-sign register

Utilice el comando user change-quorum token-sign register de la CLI de CloudHSM para registrar la estrategia de cuórum con firma simbólica para un usuario administrador.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Administrador

Sintaxis

```
aws-cloudhsm > help user change-quorum token-sign register
Register a user for quorum authentication with a public key

Usage: user change-quorum token-sign register --public-key <PUBLIC_KEY> --signed-
token <SIGNED_TOKEN>

Options:
      --cluster-id <CLUSTER_ID>      Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
      --public-key <PUBLIC_KEY>      Filepath to public key PEM file
      --signed-token <SIGNED_TOKEN>  Filepath with token signed by user private key
  -h, --help Print help (see a summary with '-h')
```

Ejemplo

Example

Para ejecutar este comando, tendrá que iniciar sesión como el usuario para el que desea register quorum token-sign.

```
aws-cloudhsm > login --username admin1 --role admin
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin1",
```

```
    "role": "admin"
  }
}
```

El comando user change-quorum token-sign register registrará su clave pública en el HSM. Como resultado, lo calificará como aprobador de cuórum para las operaciones que requieren cuórum y que requieren que un usuario obtenga las firmas de cuórum para alcanzar el umbral de cuórum necesario.

```
aws-cloudhsm > user change-quorum token-sign register \
    --public-key /home/mypemfile \
    --signed-token /home/mysignedtoken
{
  "error_code": 0,
  "data": {
    "username": "admin1",
    "role": "admin"
  }
}
```

Ahora puede ejecutar el comando user list y confirmar que se ha registrado el cuórum token-sign para este usuario.

```
aws-cloudhsm > user list
{
  "error_code": 0,
  "data": {
    "users": [
      {
        "username": "admin",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [],
        "cluster-coverage": "full"
      },
      {
        "username": "admin1",
        "role": "admin",
        "locked": "false",
        "mfa": [],
        "quorum": [
```

```
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
        "cluster-coverage": "full"
      }
    ]
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<PUBLIC-KEY>

Ruta de acceso al archivo PEM de clave pública.

Obligatorio: sí

<SIGNED-TOKEN>

Ruta de acceso al archivo con token firmado por la clave privada del usuario.

Obligatorio: sí

Temas relacionados de

- Uso de la CLI de CloudHSM para administrar la autenticación de cuórum
- Uso de la autenticación de cuórum para responsables de criptografía: primera configuración
- Cambio del valor mínimo de cuórum para los administradores
- Nombres y tipos de servicios que admiten la autenticación de cuórum

user create

El user create comando de la CLI de CloudHSM crea un usuario AWS CloudHSM en el clúster. Solo pueden ejecutar este comando las cuentas de usuario con rol de administrador.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Administrador

Requisitos

Para ejecutar este comando, debe iniciar sesión como usuario administrador.

Sintaxis

```
aws-cloudhsm > help user create
Create a new user

Usage: cloudhsm-cli user create [OPTIONS] --username <USERNAME> --role <ROLE> [--
password <PASSWORD>]

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error

      --username <USERNAME>
          Username to access the HSM cluster

      --role <ROLE>
          Role the user has in the cluster

          Possible values:
          - crypto-user: A CryptoUser has the ability to manage and use keys
          - admin:       An Admin has the ability to manage user accounts

      --password <PASSWORD>
          Optional: Plaintext user's password. If you do not include this argument you
 will be prompted for it

      --approval <APPROVAL>
          Filepath of signed quorum token file to approve operation

  -h, --help
          Print help (see a summary with '-h')
```

Ejemplo

En estos ejemplos, se muestra cómo se utiliza user create para crear usuarios nuevos en los HSM.

Example : creación de un usuario de criptografía

En este ejemplo, se crea una cuenta en el AWS CloudHSM clúster con la función de usuario criptográfico.

```
aws-cloudhsm > user create --username alice --role crypto-user
Enter password:
Confirm password:
{
  "error_code": 0,
  "data": {
    "username": "alice",
    "role": "crypto-user"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<USERNAME>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_). En este comando, el nombre de usuario no distingue entre mayúsculas y minúsculas; el nombre de usuario siempre se muestra en minúsculas.

Obligatorio: sí

<ROLE>

Especifica el rol asignado a este usuario. Este parámetro es obligatorio. Los valores válidos son admin y crypto-user.

Para obtener el rol del usuario, ejecute el comando user list. Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Entendiendo los usuarios de HSM .

<PASSWORD>

Especifica la contraseña del usuario que va a iniciar sesión en los HSM.

Obligatorio: sí

<APPROVAL>

Especifica la ruta de archivo de token de cuórum firmado para aprobar la operación. Solo es obligatorio si el valor del cuórum del servicio de usuario es superior a 1.

Temas relacionados de

- user list
- user delete
- cambio de la contraseña de un usuario

user delete

El user delete comando de la CLI de CloudHSM elimina un usuario del clúster. AWS CloudHSM Solo pueden ejecutar este comando las cuentas de usuario con rol de administrador. No puede eliminar un usuario que esté conectado actualmente a un HSM.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Administrador

Requisitos

- No puede eliminar las cuentas de usuario que posean claves.
- Su cuenta de usuario debe tener rol de administrador para poder ejecutar este comando.

Sintaxis

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
aws-cloudhsm > help user delete
Delete a user

Usage: user delete [OPTIONS] --username <USERNAME> --role <ROLE>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error

      --username <USERNAME>
          Username to access the HSM cluster

      --role <ROLE>
          Role the user has in the cluster

          Possible values:
          - crypto-user: A CryptoUser has the ability to manage and use keys
          - admin:       An Admin has the ability to manage user accounts

      --approval <APPROVAL>
          Filepath of signed quorum token file to approve operation
```

Ejemplo

```
aws-cloudhsm > user delete --username alice --role crypto-user
{
  "error_code": 0,
  "data": {
    "username": "alice",
    "role": "crypto-user"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<USERNAME>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_). En este comando, el nombre de usuario no distingue entre mayúsculas y minúsculas; el nombre de usuario siempre se muestra en minúsculas.

Obligatorio: sí

<ROLE>

Especifica el rol asignado a este usuario. Este parámetro es obligatorio. Los valores válidos son admin y crypto-user.

Para obtener el rol del usuario, ejecute el comando user list. Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Entendiendo los usuarios de HSM .

Obligatorio: sí

<APPROVAL>

Especifica la ruta de archivo de token de cuórum firmado para aprobar la operación. Solo es obligatorio si el valor del cuórum del servicio de usuario es superior a 1.

Obligatorio: sí

Temas relacionados de

- user list
- user create
- user change-password

user list

El comando user list de la CLI de CloudHSM muestra las cuentas de usuario presentes en su clúster de CloudHSM. No es necesario haber iniciado sesión en la CLI de CloudHSM para ejecutar este comando.

> **Note**
>
> Si agrega o elimina los HSM, actualice los archivos de configuración que utilizan el AWS CloudHSM cliente y las herramientas de línea de comandos. De lo contrario, es posible que los cambios que realice no se hagan efectivos en todos los HSM del clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios. No es preciso haber iniciado sesión para ejecutar este comando.

Sintaxis

```
aws-cloudhsm > help user list
List the users in your cluster

USAGE:
    user list

Options:
      --cluster-id <CLUSTER_ID>  Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
  -h, --help                     Print help
```

Ejemplo

Este comando muestra los HSM presentes en el clúster de CloudHSM.

```
aws-cloudhsm > user list
{
  "error_code": 0,
  "data": {
    "users": [
      {
        "username": "admin",
        "role": "admin",
        "locked": "false",
        "mfa": [],
```

```
        "cluster-coverage": "full"
      },
      {
        "username": "test_user",
        "role": "admin",
        "locked": "false",
        "mfa": [
          {
            "strategy": "token-sign",
            "status": "enabled"
          }
        ],
        "cluster-coverage": "full"
      },
      {
        "username": "app_user",
        "role": "internal(APPLIANCE_USER)",
        "locked": "false",
        "mfa": [],
        "cluster-coverage": "full"
      }
    ]
  }
}
```

La salida contiene los siguientes atributos de usuario:

- Username: muestra el nombre fácil de recordar definido por el usuario para el usuario. El nombre de usuario se muestra siempre en minúsculas.
- Role: determina las operaciones que el usuario puede realizar en el HSM.
- Bloqueado: indica si esta cuenta de usuario se ha bloqueado.
- MFA: indica los mecanismos de autenticación multifactor compatibles con esta cuenta de usuario.
- Cobertura de clúster: indica la disponibilidad de esta cuenta de usuario en todo el clúster.

Temas relacionados de

- listUsers en key_mgmt_util
- user create
- user delete

- cambio de la contraseña de un usuario

quorum

quorum es una categoría principal para un grupo de comandos que, cuando se combina con quorum, crea un comando específico para la autenticación de cuórum o para operaciones M de N. Actualmente, esta categoría consiste en la subcategoría token-sign, la cual consta de sus propios comandos. Haga clic en el siguiente enlace para obtener más detalles.

- token-sign

Servicios de administración: la autenticación de cuórum se utiliza para los servicios con privilegios de administrador, como la creación y eliminación de usuarios, el cambio de las contraseñas de los usuarios, la configuración de los valores de cuórum y la desactivación de las capacidades de cuórum y MFA.

Además, cada tipo de servicio se divide en un nombre de servicio válido que contiene un conjunto específico de operaciones de servicio compatibles con cuórum que se pueden realizar.

Nombre del servicio	Tipo de servicio	Operaciones de servicio
usuario	Administrador	• user create • user delete • user change-password • user change-mfa
quorum	Administrador	• signo simbólico de quórum set-quorum-value

Temas relacionados de

- Uso de la autenticación de cuórum para administradores: configuración por primera vez
- Usar la CLI de CloudHSM para gestionar la autenticación de cuórum (control de acceso M de N)

quorum token-sign

quorum token-sign es una categoría para un grupo de comandos que, cuando se combinan con quorum token-sign, crean un comando específico para la autenticación de cuórum o para operaciones M de N.

Actualmente, esta categoría consta de los siguientes comandos:

- eliminar
- generate
- list
- list-quorum-values
- list-timeouts
- set-quorum-value
- set-timeout

quorum token-sign delete

Utilice el comando quorum token-sign delete de la CLI de CloudHSM para eliminar uno o más tokens de un servicio autorizado de cuórum.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Administrador

Sintaxis

```
aws-cloudhsm > help quorum token-sign delete
Delete one or more Quorum Tokens

Usage: quorum token-sign delete --scope <SCOPE>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error
```

```
      --scope <SCOPE>
          Scope of which token(s) will be deleted

          Possible values:
          - user: Deletes all token(s) of currently logged in user
          - all:  Deletes all token(s) on the HSM
  -h, --help
          Print help (see a summary with '-h')
```

Ejemplo

En el siguiente ejemplo se muestra cómo se puede usar el comando quorum token-sign delete de la CLI de CloudHSM para eliminar uno o más tokens de un servicio autorizado de cuórum.

Example : elimine uno o más tokens de un servicio autorizado de cuórum.

```
aws-cloudhsm > quorum token-sign delete --scope all
{
  "error_code": 0,
  "data": "Deletion of quorum token(s) successful"
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<SCOPE>

El ámbito en el que se eliminarán los tokens del AWS CloudHSM clúster.

Valores válidos

- User: se utiliza para eliminar únicamente los tokens que son propiedad del usuario que ha iniciado sesión.
- Todos: se utiliza para eliminar todos los tokens del AWS CloudHSM clúster.

Temas relacionados de

- user list

- user create
- user delete

quorum token-sign generate

Utilice el comando quorum token-sign generate de la CLI de CloudHSM para generar un token para un servicio autorizado de cuórum.

Existe un límite para obtener un token activo por usuario y servicio en un clúster de HSM por usuario y cuórum de servicios.

> **Note**
>
> Solo los administradores pueden generar un token de servicio.

Servicios de administración: la autenticación de cuórum se utiliza para los servicios con privilegios de administrador, como la creación y eliminación de usuarios, el cambio de las contraseñas de los usuarios, la configuración de los valores de cuórum y la desactivación de las capacidades de cuórum y MFA.

Además, cada tipo de servicio se divide en un nombre de servicio válido que contiene un conjunto específico de operaciones de servicio compatibles con cuórum que se pueden realizar.

Nombre del servicio	Tipo de servicio	Operaciones de servicio
usuario	Administrador	• user create • user delete • user change-password • user change-mfa
quorum	Administrador	• signo simbólico de quórum set-quorum-value

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Administrador
- Usuario de criptografía (CU)

Sintaxis

```
aws-cloudhsm > help quorum token-sign generate
Generate a token

Usage: quorum token-sign generate --service <SERVICE> --token <TOKEN>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error

      --service <SERVICE>
          Service the token will be used for

          Possible values:
          - user:
            User management service is used for executing quorum authenticated user
 management operations
          - quorum:
            Quorum management service is used for setting quorum values for any quorum
 service
          - registration:
            Registration service is used for registering a public key for quorum
 authentication

      --token <TOKEN>
          Filepath where the unsigned token file will be written
  -h, --help                     Print help
```

Ejemplo

Este comando escribirá un token sin firmar por cada HSM del clúster en el archivo especificado por `token`.

Example : escriba un token sin firmar por cada HSM de su clúster

```
aws-cloudhsm > quorum token-sign generate --service user --token /home/tfile
```

```
{
  "error_code": 0,
  "data": {
    "filepath": "/home/tfile"
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<SERVICE>

Especifica el servicio autorizado de cuórum para el que se generará un token. Este parámetro es obligatorio.

Valores válidos

- usuario: el servicio de administración de usuarios que se utiliza para ejecutar las operaciones de administración de usuarios autorizadas por cuórum.
- cuórum: servicio de administración de quorum que se utiliza para establecer los valores de quorum autorizados para cualquier servicio autorizado de quorum.
- registro: genera un token sin firmar que se utiliza para registrar una clave pública para la autorización del cuórum.

Obligatorio: sí

<TOKEN>

Ruta de archivo en la que se escribirá el archivo de token sin firmar.

Obligatorio: sí

Temas relacionados de

- Nombres y tipos de servicios que admiten la autenticación de cuórum

quorum token-sign list

Utilice el quorum token-sign list comando de la CLI de CloudHSM para enumerar todos los tokens de quórum con signo de token presentes en el clúster. AWS CloudHSM

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Administrador
- Usuario de criptografía (CU)

Sintaxis

```
aws-cloudhsm > help quorum token-sign list
List the token-sign tokens in your cluster

Usage: quorum token-sign list

Options:
      --cluster-id <CLUSTER_ID>  Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
  -h, --help                     Print help
```

Ejemplo

Este comando mostrará una lista de todos los tokens con signos simbólicos presentes en su clúster. AWS CloudHSM

Example

```
aws-cloudhsm > quorum token-sign list
{
  "error_code": 0,
  "data": {
    "tokens": [
      {
        "username": "admin",
        "service": "quorum",
        "approvals-required": 2
        "number-of-approvals": 0
```

```
        "token-timeout-seconds": 397
        "cluster-coverage": "full"
      },
      {
        "username": "admin",
        "service": "user",
        "approvals-required": 2
        "number-of-approvals": 2
        "token-timeout-seconds": 588
        "cluster-coverage": "full"
      }
    ]
  }
}
```

Temas relacionados de

- quorum token-sign generate

signo simbólico de quórum list-quorum-values

Utilice el quorum token-sign list-quorum-values comando de la CLI de CloudHSM para enumerar los valores de quórum establecidos en el clúster. AWS CloudHSM

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios. No es preciso haber iniciado sesión para ejecutar este comando.

Sintaxis

```
aws-cloudhsm > help quorum token-sign list-quorum-values
List current quorum values

Usage: quorum token-sign list-quorum-values

Options:
      --cluster-id <CLUSTER_ID>  Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
  -h, --help                     Print help
```

Ejemplo

Este comando muestra los valores de quórum establecidos en el AWS CloudHSM clúster para cada servicio.

Example

```
aws-cloudhsm > quorum token-sign list-quorum-values
{
  "error_code": 0,
  "data": {
    "user": 1,
    "quorum": 1
  }
}
```

Temas relacionados de

- Nombres y tipos de servicios que admiten la autenticación de cuórum

quorum token-sign list-timeouts

Ejecute el comando quorum token-sign list-timeouts en la CLI de CloudHSM para definir el tiempo de espera del token (en segundos) para cada tipo de token.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios. No es preciso haber iniciado sesión para ejecutar este comando.

Sintaxis

```
aws-cloudhsm > help quorum token-sign list-timeouts
List timeout durations in seconds for token validity

Usage: quorum token-sign list-timeouts

Options:
      --cluster-id <CLUSTER_ID>  Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
```

```
  -h, --help                    Print help
```

Ejemplo

Example

```
aws-cloudhsm > quorum token-sign list-timeouts
{
  "error_code": 0,
  "data": {
    "generated": 600,
    "approved": 600
  }
}
```

La salida incluye lo siguiente:

- generated: tiempo de espera en segundos para que se apruebe un token generado.
- approved: tiempo de espera en segundos para que un token aprobado se utilice para ejecutar una operación autorizada por cuórum.

Temas relacionados de

- quorum token-sign set-timeout

signo simbólico de quórum set-quorum-value

Ejecute el comando quorum token-sign set-quorum-value en la CLI de CloudHSM para definir un nuevo valor de cuórum para un servicio autorizado por cuórum.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Administrador

Sintaxis

```
aws-cloudhsm > help quorum token-sign set-quorum-value
```

```
Set a quorum value

Usage: quorum token-sign set-quorum-value [OPTIONS] --service <SERVICE> --value <VALUE>

Options:
      --cluster-id <CLUSTER_ID>
          Unique Id to choose which of the clusters in the config file to run the
 operation against. If not provided, will fall back to the value provided when
 interactive mode was started, or error

      --service <SERVICE>
          Service the token will be used for

          Possible values:
          - user:
            User management service is used for executing quorum authenticated user
 management operations
          - quorum:
            Quorum management service is used for setting quorum values for any quorum
 service

      --value <VALUE>
          Value to set for service

      --approval <APPROVAL>
          Filepath of signed quorum token file to approve operation

  -h, --help
          Print help (see a summary with '-h')
```

Ejemplo

Example

En el siguiente ejemplo, este comando genera un token sin firmar por cada HSM del clúster en el archivo especificado por el token. Cuando se le solicite, firme los tokens del archivo.

```
aws-cloudhsm > quorum token-sign set-quorum-value --service quorum --value 2
{
  "error_code": 0,
  "data": "Set Quorum Value successful"
}
```

A continuación, puede ejecutar el comando list-quorum-values para confirmar que se ha definido el valor de cuórum para el servicio de administración de cuórum:

```
aws-cloudhsm > quorum token-sign list-quorum-values
{
  "error_code": 0,
  "data": {
    "user": 1,
    "quorum": 2
  }
}
```

Argumentos

<CLUSTER_ID>

El ID del clúster en el que se va a ejecutar esta operación.

Obligatorio: si se han configurado varios clústeres.

<APPROVAL>

La ruta del archivo de token firmado que se va a aprobar en el HSM.

<SERVICE>

Especifica el servicio autorizado de cuórum para el que se generará un token. Este parámetro es obligatorio. Para obtener más información sobre los tipos y nombres de servicio, consulte Nombres y tipos de servicios que admiten la autenticación de cuórum.

Valores válidos

- user: servicio de gestión de usuarios. Servicio usado para ejecutar las operaciones de administración de usuarios autorizadas por cuórum.
- cuórum: servicio de gestión de cuórum. Servicio usado para definir los valores de cuórum autorizado para cualquier servicio autorizado por cuórum.
- registration: genera un token sin firmar para usarlo al registrar una clave pública para la autorización de cuórum.

Obligatorio: sí

<VALUE>

Especifica el valor de cuórum que se va a definir. El valor máximo de cuórum es ocho (8).

Obligatorio: sí

Temas relacionados de

- signo simbólico de quórum list-quorum-values
- Nombres y tipos de servicios que admiten la autenticación de cuórum

quorum token-sign set-timeout

Ejecute el comando quorum token-sign set-timeout en la CLI de CloudHSM para definir el periodo de espera del token (en segundos) para cada tipo de token.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Administrador

Sintaxis

```
aws-cloudhsm > help quorum token-sign set-timeout
Set timeout duration in seconds for token validity

Usage: quorum token-sign set-timeout <--generated <GENERATED> |--approved <APPROVED>>

Options:
      --cluster-id <CLUSTER_ID>   Unique Id to choose which of the clusters in the
 config file to run the operation against. If not provided, will fall back to the value
 provided when interactive mode was started, or error
      --generated <GENERATED>     Timeout period in seconds for a generated (non-
approved) token to be approved
      --approved <APPROVED>      Timeout period in seconds for an approved token to be
 used to execute a quorum operation
  -h, --help    Print help (see a summary with '-h')
```

Ejemplo

Los siguientes ejemplos muestran cómo usar el comando quorum token-sign set-timeout para definir el periodo de tiempo de espera del token.

```
aws-cloudhsm > quorum token-sign set-timeout --generated 900
```

```
{
  "error_code": 0,
  "data": "Set token timeout successful"
}
```

Temas relacionados de

- quorum token-sign list-timeouts

Utilidad de administración de CloudHSM (CMU)

La herramienta de línea de comandos cloudhsm_mgmt_util ayuda a los responsables de criptografía a administrar los usuarios en los HSM. Incluye herramientas que crean, eliminan y enumeran los usuarios, y cambian las contraseñas de los usuarios.

KMU y CMU forman parte del conjunto de SDK 3 de cliente.

cloudhsm_mgmt_util también incluye comandos que permiten a los usuarios criptográficos (CU) compartir claves y obtener y establecer atributos de claves. Estos comandos complementan a los comandos de administración de claves de la herramienta de administración de clave principal, key_mgmt_util.

Para un inicio rápido, consulte Administración de clústeres clonados. Para obtener información detallada sobre los comandos cloudhsm_mgmt_util y ejemplos de uso de los comandos, consulte Referencia del comando cloudhsm_mgmt_util.

Temas

- Plataformas compatibles para la utilidad AWS CloudHSM de administración
- Introducción a la utilidad de administración de CloudHSM (CMU)
- Instalar y configurar el AWS CloudHSM cliente (Linux)
- Instalación y configuración del AWS CloudHSM cliente (Windows)
- Referencia del comando cloudhsm_mgmt_util

Plataformas compatibles para la utilidad AWS CloudHSM de administración

Compatibilidad con Linux

- Amazon Linux
- Amazon Linux 2
- CentOS 6.10+
- CentOS 7.3+
- CentOS 8
- Red Hat Enterprise Linux (RHEL) 6.10+
- Red Hat Enterprise Linux (RHEL) 7.9+
- Red Hat Enterprise Linux (RHEL) 8
- Ubuntu 16.04 LTS
- Ubuntu 18.04 LTS

Compatibilidad con Windows

- Microsoft Windows Server 2012
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019

Introducción a la utilidad de administración de CloudHSM (CMU)

La utilidad de administración CloudHSM (CMU) le permite administrar a los usuarios del módulo de seguridad de hardware (HSM). Utilice este tema para comenzar con las tareas básicas de administración de usuarios de HSM, como crear usuarios, enumerarlos y conectar la CMU al clúster.

1. Para usar la CMU, primero debe utilizar la herramienta de configuración para actualizar la configuración de la CMU local con el parámetro `--cmu` y una dirección IP de uno de los HSM del clúster. Haga esto cada vez que utilice la CMU para asegurarse de administrar los usuarios de HSM en todos los HSM del clúster.

   Linux

   ```
   $ sudo /opt/cloudhsm/bin/configure --cmu <IP address>
   ```

Windows

```
C:\Program Files\Amazon\CloudHSM\bin\ configure.exe --cmu <IP address>
```

2. Ingrese el siguiente comando para iniciar la CLI en modo interactivo.

   Linux

   ```
   $ /opt/cloudhsm/bin/cloudhsm_mgmt_util /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM> .\cloudhsm_mgmt_util.exe C:\ProgramData\Amazon
   \CloudHSM\data\cloudhsm_mgmt_util.cfg
   ```

   La salida será parecida a la que se muestra a continuación en función de la cantidad de HSM que tenga.

   ```
   Connecting to the server(s), it may take time
   depending on the server(s) load, please wait...

   Connecting to server '10.0.2.9': hostname '10.0.2.9', port 2225...
   Connected to server '10.0.2.9': hostname '10.0.2.9', port 2225.

   Connecting to server '10.0.3.11': hostname '10.0.3.11', port 2225...
   Connected to server '10.0.3.11': hostname '10.0.3.11', port 2225.

   Connecting to server '10.0.1.12': hostname '10.0.1.12', port 2225...
   Connected to server '10.0.1.12': hostname '10.0.1.12', port 2225.
   ```

   El símbolo cambia a `aws-cloudhsm>` cuando se está ejecutando cloudhsm_mgmt_util.

3. Utilice el comando loginHSM para iniciar sesión en el clúster. Cualquier usuario de cualquier tipo puede utilizar este comando para iniciar sesión en el clúster.

   El comando en el siguiente ejemplo inicia sesión como admin que es el responsable de criptografía (CO) predeterminado. La contraseña de este usuario se establece al activar el clúster. Puede usar el parámetro `-hpswd` para ocultar la contraseña.

```
aws-cloudhsm>loginHSM CO admin -hpswd
```

El sistema le solicitará su contraseña. Introduce la contraseña, el sistema la oculta y el resultado muestra que el comando se ha ejecutado correctamente y que usted se ha conectado a todos los HSM en el clúster.

```
Enter password:


loginHSM success on server 0(10.0.2.9)
loginHSM success on server 1(10.0.3.11)
loginHSM success on server 2(10.0.1.12)
```

4. Ejecute listUsers para listar todos los usuarios del clúster.

```
aws-cloudhsm>listUsers
```

En la CMU, se muestran todos los usuarios del clúster.

```
Users on server 0(10.0.2.9):
Number of users found:2

    User Id             User Type       User Name
 MofnPubKey    LoginFailureCnt         2FA
         1              CO              admin                                   NO
               0               NO
         2              AU              app_user                                NO
               0               NO
Users on server 1(10.0.3.11):
Number of users found:2

    User Id             User Type       User Name
 MofnPubKey    LoginFailureCnt         2FA
         1              CO              admin                                   NO
               0               NO
         2              AU              app_user                                NO
               0               NO
Users on server 2(10.0.1.12):
Number of users found:2
```

```
    User Id              User Type       User Name
 MofnPubKey    LoginFailureCnt         2FA
         1              CO              admin                                NO
               0              NO
         2              AU              app_user                             NO
               0              NO
```

5. Utilice createUser para crear un nombre de usuario de CU **example_user** con una contraseña de **password1**.

   Utilice usuarios de CU en las aplicaciones para realizar operaciones criptográficas y de administración de claves. Puede crear usuarios de CU porque en el paso 3 inició sesión como usuario de CO. Solo los usuarios de CO pueden realizar tareas de administración de usuarios con la CMU, como crear y eliminar usuarios y cambiar las contraseñas de otros usuarios.

   ```
   aws-cloudhsm>createUser CU example_user password1
   ```

   La CMU le solicita información sobre la operación de creación de usuario.

   ```
   
   ************************CAUTION*******************************
   This is a CRITICAL operation, should be done on all nodes in the
   cluster. AWS does NOT synchronize these changes automatically with the
   nodes on which this operation is not executed or failed, please
   ensure this operation is executed on all nodes in the cluster.
   **************************************************************
   
   Do you want to continue(y/n)?
   ```

6. Para crear el usuario de CU **example_user**, escriba **y**.
7. Ejecute listUsers para listar todos los usuarios del clúster.

   ```
   aws-cloudhsm>listUsers
   ```

   La CMU muestra todos los usuarios del clúster, incluido el nuevo usuario de CU que acaba de crear.

   ```
   
   Users on server 0(10.0.2.9):
   Number of users found:3
   ```

```
    User Id             User Type       User Name
 MofnPubKey    LoginFailureCnt         2FA
         1              CO              admin                                   NO
               0               NO
         2              AU              app_user                                NO
               0               NO
         3              CU              example_user                            NO
               0               NO
Users on server 1(10.0.3.11):
Number of users found:3

    User Id             User Type       User Name
 MofnPubKey    LoginFailureCnt         2FA
         1              CO              admin                                   NO
               0               NO
         2              AU              app_user                                NO
               0               NO
         3              CU              example_user                            NO
               0               NO
Users on server 2(10.0.1.12):
Number of users found:3

    User Id             User Type       User Name
 MofnPubKey    LoginFailureCnt         2FA
         1              CO              admin                                   NO
               0               NO
         2              AU              app_user                                NO
               0               NO
         3              CU              example_user                            NO
               0               NO
```

8. Utilice el comando logoutHSM para cerrar la sesión en los HSM.

```
aws-cloudhsm>logoutHSM


logoutHSM success on server 0(10.0.2.9)
logoutHSM success on server 1(10.0.3.11)
logoutHSM success on server 2(10.0.1.12)
```

9. Utilice el comando quit para detener cloudhsm_mgmt_util.

```
aws-cloudhsm>quit


disconnecting from servers, please wait...
```

Instalar y configurar el AWS CloudHSM cliente (Linux)

Para interactuar con el HSM de su AWS CloudHSM clúster, necesita el software de AWS CloudHSM cliente para Linux. Debe instalarlo en la instancia de cliente de Linux EC2 que creó anteriormente. También puede instalar un cliente si utiliza Windows. Para obtener más información, consulte Instalación y configuración del AWS CloudHSM cliente (Windows).

Tareas

- Instale el AWS CloudHSM cliente y las herramientas de línea de comandos
- Edición de la configuración del cliente

Instale el AWS CloudHSM cliente y las herramientas de línea de comandos

Conéctese a su instancia de cliente y ejecute los siguientes comandos para descargar e instalar el AWS CloudHSM cliente y las herramientas de línea de comandos.

Amazon Linux

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-latest.el6.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el6.x86_64.rpm
```

Amazon Linux 2

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-latest.el7.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el7.x86_64.rpm
```

CentOS 7

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-latest.el7.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el7.x86_64.rpm
```

CentOS 8

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-latest.el8.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el8.x86_64.rpm
```

RHEL 7

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-latest.el7.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el7.x86_64.rpm
```

RHEL 8

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-latest.el8.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el8.x86_64.rpm
```

Ubuntu 16.04 LTS

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/cloudhsm-client_latest_amd64.deb
```

```
sudo apt install ./cloudhsm-client_latest_amd64.deb
```

Ubuntu 18.04 LTS

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Bionic/cloudhsm-client_latest_u18.04_amd64.deb
```

```
sudo apt install ./cloudhsm-client_latest_u18.04_amd64.deb
```

Edición de la configuración del cliente

Antes de poder usar el AWS CloudHSM cliente para conectarse al clúster, debe editar la configuración del cliente.

Para editar la configuración del cliente

1. Si va a instalar SDK 3 de cliente en cloudhsm_mgmt_util, realice los siguientes pasos para asegurarse de que todos los nodos del clúster estén sincronizados.

   a. Ejecute configure -a ***<IP of one of the HSMs>***.

   b. Reinicie el servicio del cliente.

   c. Ejecute config -m.

2. Copie el certificado de emisión —el que utilizó para firmar el certificado del clúster— en la siguiente ubicación de la instancia de cliente: `/opt/cloudhsm/etc/customerCA.crt`. Necesita permisos de usuario raíz en la instancia de cliente para copiar el certificado en esta ubicación.

3. Utilice el siguiente comando configure para actualizar los archivos de configuración del AWS CloudHSM cliente y las herramientas de línea de comandos, especificando la dirección IP del HSM del clúster. Para obtener la dirección IP del HSM, consulte el clúster en la AWS CloudHSM consola o ejecute el comando describe-clusters CLI. En la salida del comando, la dirección IP del HSM es el valor del campo `EniIp`. Si tiene más de un HSM, elija la dirección IP de cualquiera de ellos; no importa el que elija.

```
sudo /opt/cloudhsm/bin/configure -a <IP address>

Updating server config in /opt/cloudhsm/etc/cloudhsm_client.cfg
Updating server config in /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
```

4. Vaya a Activación del clúster.

Instalación y configuración del AWS CloudHSM cliente (Windows)

Para trabajar con un HSM de su AWS CloudHSM clúster en Windows, necesita el software de AWS CloudHSM cliente para Windows. Debe instalarlo en la instancia de Windows Server que creó anteriormente.

Cómo instalar (o actualizar) el cliente más reciente en Windows y las herramientas de línea de comandos

1. Conéctese a su instancia de Windows Server.
2. Descargue el instalador AWSCloudHSMClient -latest.msi.
3. Si va a instalar SDK 3 de cliente en cloudhsm_mgmt_util, realice los siguientes pasos para asegurarse de que todos los nodos del clúster estén sincronizados.
   a. Ejecute configure -a ***<IP of one of the HSMs>***.
   b. Reinicie el servicio del cliente.
   c. Ejecute config -m.
4. Vaya a la ubicación de descarga y ejecute el instalador (AWSCloudHSMClient-latest.msi) con privilegios administrativos.
5. Siga las instrucciones del instalador y, a continuación, seleccione Cerrar cuando el instalador haya finalizado.
6. Copie el certificado de emisión autofirmado —el que utilizó para firmar el certificado del clúster — en la carpeta `C:\ProgramData\Amazon\CloudHSM`.
7. Ejecute el siguiente comando para actualizar los archivos de configuración. Asegúrese de detener e iniciar el cliente durante la reconfiguración si lo está actualizando:

```
C:\Program Files\Amazon\CloudHSM\bin\ configure.exe -a <HSM IP address>
```

8. Vaya a Activación del clúster.

Notas:

- Si está actualizando el cliente, los archivos de configuración existentes de las instalaciones anteriores no se sobrescribirán.
- El instalador del AWS CloudHSM cliente para Windows registra automáticamente la API de criptografía: Next Generation (CNG) y Key Storage Provider (KSP). Para desinstalar el software de cliente, vuelva a ejecutar el instalador y siga las instrucciones de desinstalación.
- Si utiliza Linux, puede instalar el software de cliente de Linux. Para obtener más información, consulte Instalar y configurar el AWS CloudHSM cliente (Linux).

Referencia del comando cloudhsm_mgmt_util

La herramienta de línea de comandos cloudhsm_mgmt_util ayuda a los responsables de criptografía a administrar los usuarios en los HSM. También incluye comandos que permiten a los usuarios de criptografía (CU) compartir claves, obtener y configurar atributos de las claves. Estos comandos complementan los comandos primarios de administración de claves en la herramienta de línea de comandos key_mgmt_util.

Para un inicio rápido, consulte Administración de clústeres clonados.

Para poder ejecutar cualquier comando de cloudhsm_mgmt_util, debe iniciar cloudhsm_mgmt_util e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de cuenta de usuario que pueda ejecutar los comandos que desea utilizar.

Para listar todos los comandos de cloudhsm_mgmt_util, ejecute el siguiente comando:

```
aws-cloudhsm> help
```

Para obtener la sintaxis de un comando de cloudhsm_mgmt_util, ejecute el siguiente comando:

```
aws-cloudhsm> help <command-name>
```

> **Note**
>
> Utilice la sintaxis como se indica en la documentación. Aunque la ayuda integrada del software puede proporcionar opciones adicionales, estas no deben considerarse compatibles y no deben utilizarse en el código de producción.

Para ejecutar un comando, introduzca el nombre del comando o una parte suficiente para distinguirlo de los nombres de los demás comandos de cloudhsm_mgmt_util.

Por ejemplo, para obtener una lista de los usuarios de los HSM, introduzca listUsers o listU.

```
aws-cloudhsm> listUsers
```

Para finalizar la sesión de cloudhsm_mgmt_util, ejecute el siguiente comando:

```
aws-cloudhsm> quit
```

Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Los siguientes temas describen los comandos de cloudhsm_mgmt_util.

> **Note**
>
> Algunos comandos de key_mgmt_util y cloudhsm_mgmt_util tienen el mismo nombre. Sin embargo, los comandos suelen tener una sintaxis diferente, un resultado diferente y una funcionalidad ligeramente diferente.

Comando	Descripción	Tipo de usuario
changePswd	Cambia las contraseñas de los usuarios en los HSM. Cualquier usuario puede cambiar su propia contraseña. Los CO pueden cambiar	CO

Comando	Descripción	Tipo de usuario
	la contraseña de cualquier persona.	
createUser	Crea usuarios de todo tipo en los HSM.	CO
deleteUser	Elimina usuarios de todo tipo de los HSM.	CO
findAllKeys	Obtiene las claves que un usuario posee o comparte. También obtiene un hash de la propiedad de la clave y del uso compartido de datos para todas las claves en cada HSM.	CO, AU
getAttribute	Obtiene el valor de un atributo para una AWS CloudHSM clave y lo escribe en un archivo o en una salida estándar (salida estándar).	CU
getCert	Obtiene el certificado de un HSM específico y lo guarda en el formato de certificado que se desee.	Todos.
getHSMInfo	Obtiene información sobre el hardware en el que se está ejecutando un HSM.	Todos. No es necesario iniciar sesión.
getKeyInfo	Obtiene propietarios, usuarios compartidos y el estado de autenticación de cuórum de una clave.	Todos. No es necesario iniciar sesión.

Comando	Descripción	Tipo de usuario
info	Obtiene información sobre un HSM, incluida la dirección IP, el nombre de host, el puerto y el usuario actual.	Todos. No es necesario iniciar sesión.
listUsers	Obtiene los usuarios de cada uno de los HSM, su ID y tipo de usuario y otros atributos.	Todos. No es necesario iniciar sesión.
loginHSM y logoutHSM	Permiten iniciar y cerrar la sesión en un HSM.	Todos.
quit	Cierra cloudhsm_mgmt_util.	Todos. No es necesario iniciar sesión.
servidor	Permite entrar y salir del modo de servidor en un HSM.	Todos.
registerQuorumPubClave	Asocia un usuario de HSM a un par de claves asimétrico RSA-2048.	CO
setAttribute	Cambia los valores de los atributos de la etiqueta, de cifrado, descifrado, encapsulado y desencapsulado de una clave existente.	CU
shareKey	Comparte una clave existente con otros usuarios.	CU
syncKey	Sincroniza una clave en los clústeres clonados AWS CloudHSM .	CU, CO

Comando	Descripción	Tipo de usuario
syncUser	Sincroniza un usuario en todos los clústeres clonados. AWS CloudHSM	CO

changePswd

El comando changePswd de cloudhsm_mgmt_util cambia la contraseña de un usuario existente en los HSM del clúster.

Cualquier usuario puede cambiar su propia contraseña. Además, los responsables de criptografía (CO y PCO) pueden cambiar la contraseña de otro CO o usuario de criptografía (CU). No es necesario que escriba la contraseña actual para realizar el cambio.

> **Note**
>
> No puede cambiar la contraseña de un usuario que haya iniciado sesión actualmente en el AWS CloudHSM cliente o en key_mgmt_util.

Solución de problemas de changePswd

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Responsables de criptografía (CO)
- Usuarios de criptografía (CU)

Sintaxis

Introduzca los argumentos en el orden especificado en el diagrama de sintaxis. Utilice el parámetro `-hpswd` para ocultar la contraseña. Para habilitar la autenticación de dos factores (2FA) para un usuario tipo CO, utilice el parámetro `-2fa` e incluya una ruta de archivo. Para obtener más información, consulte the section called “Argumentos”.

```
changePswd <user-type> <user-name> <password |-hpswd> [-2fa </path/to/authdata>]
```

Ejemplos

Los siguientes ejemplos muestran cómo utilizar changePassword para restablecer la contraseña del usuario actual o de cualquier otro usuario de los HSM.

Example : cambio de la contraseña

Cualquier usuario de los HSM puede utilizar changePswd para cambiar su propia contraseña. Antes de cambiar la contraseña, utilice info para obtener información sobre cada uno de los HSM del clúster, incluidos el nombre de usuario y el tipo de usuario del usuario que ha iniciado sesión.

La siguiente salida muestra que Bob ha iniciado sesión como usuario de criptografía (CU).

```
        aws-cloudhsm> info server 0

Id      Name                    Hostname         Port   State           Partition
  LoginState
0       10.1.9.193              10.1.9.193      2225    Connected       hsm-jqici4covtv
  Logged in as 'bob(CU)'

aws-cloudhsm> info server 1

Id      Name                    Hostname         Port   State           Partition
  LoginState
1       10.1.10.7               10.1.10.7       2225    Connected       hsm-ogi3sywxbqx
  Logged in as 'bob(CU)'
```

Para cambiar la contraseña, Bob ejecuta changePswd y, a continuación, introduce el tipo de usuario, el nombre de usuario y una contraseña nueva.

```
aws-cloudhsm> changePswd CU bob newPassword

*************************CAUTION*******************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
***************************************************************

Do you want to continue(y/n)?y
Changing password for bob(CU) on 2 nodes
```

Example : cambio de la contraseña de otro usuario

Debe ser un usuario de tipo CO o PCO para cambiar la contraseña de otro CO, o CU en los HSM. Antes de cambiar la contraseña de otro usuario, utilice el comando info para confirmar que su tipo de usuario es CO o PCO.

La siguiente salida confirma que Alice, que es usuaria de tipo CO, tiene una sesión iniciada.

```
aws-cloudhsm>info server 0

Id      Name             Hostname         Port   State           Partition
 LoginState
0       10.1.9.193       10.1.9.193        2225   Connected      hsm-jqici4covtv
 Logged in as 'alice(CO)'


aws-cloudhsm>info server 1

Id      Name             Hostname         Port   State           Partition
 LoginState
0       10.1.10.7        10.1.10.7        2225   Connected       hsm-ogi3sywxbqx
 Logged in as 'alice(CO)'
```

Alice quiere restablecer la contraseña de otro usuario, John. Antes de cambiar la contraseña, utiliza el comando listUsers para comprobar el tipo de usuario de John.

La siguiente salida muestra que John es un usuario de tipo CO.

```
aws-cloudhsm> listUsers
Users on server 0(10.1.9.193):
Number of users found:5

    User Id             User Type       User Name            MofnPubKey
 LoginFailureCnt         2FA
         1              PCO             admin                   YES              0
             NO
         2              AU              jane                     NO              0
             NO
         3              CU              bob                      NO              0
             NO
         4              CU              alice                    NO              0
             NO
         5              CO              john                     NO              0
             NO
Users on server 1(10.1.10.7):
Number of users found:5

    User Id             User Type       User Name            MofnPubKey
 LoginFailureCnt         2FA
         1              PCO             admin                   YES              0
             NO
         2              AU              jane                     NO              0
             NO
         3              CU              bob                      NO              0
             NO
         4              CO              alice                    NO              0
             NO
         5              CO              john                     NO              0
             NO
```

Para cambiar la contraseña, Alice ejecuta changePswd y, a continuación, introduce el tipo de usuario, el nombre de usuario y la contraseña nueva de John.

```
aws-cloudhsm>changePswd CO john newPassword

************************CAUTION*******************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
**************************************************************
```

```
Do you want to continue(y/n)?y
Changing password for john(CO) on 2 nodes
```

Argumentos

Introduzca los argumentos en el orden especificado en el diagrama de sintaxis. Utilice el parámetro `-hpswd` para ocultar la contraseña. Para habilitar la 2FA para un usuario de tipo CO, utilice el parámetro `-2fa` e incluya una ruta de archivo. Para obtener más información acerca de cómo trabajar con la 2FA, consulte Uso de la CMU para administrar la 2FA

```
changePswd <user-type> <user-name> <password |-hpswd> [-2fa </path/to/authdata>]
```

<user-type>

Especifica el tipo actual de usuario cuya contraseña está cambiando. No se puede utilizar changePswd para cambiar el tipo de usuario.

Los valores válidos son `CO`, `CU`, `PCO` y `PRECO`.

Para obtener el tipo de usuario, utilice listUsers. Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Más información sobre los usuarios de HSM.

Obligatorio: sí

<user-name>

Especifica el nombre fácil de recordar del usuario. Este parámetro no distingue entre mayúsculas y minúsculas. No puede utilizar changePswd para cambiar el nombre del usuario.

Obligatorio: sí

<password | -hpswd >

Especifica una contraseña nueva para el usuario. Escriba una cadena de entre 7 y 32 caracteres. Este valor distingue entre mayúsculas y minúsculas. La contraseña aparece en texto no cifrado cuando se escribe. Para ocultar la contraseña, utilice el parámetro `-hpswd` en lugar de la contraseña y siga las indicaciones.

Obligatorio: sí

[-2fa </path/to/authdata>]

Especifica la activación de la 2FA para este usuario de tipo CO. Para obtener los datos necesarios para configurar la 2FA, incluya una ruta a una ubicación del sistema de archivos con un nombre de archivo después del parámetro `-2fa`. Para obtener más información acerca de cómo trabajar con la 2FA, consulte Uso de la CMU para administrar la 2FA.

Obligatorio: no

Temas relacionados de

- info
- listUsers
- createUser
- deleteUser

createUser

El comando createUser en cloudhsm_mgmt_util crea un usuario en los HSM. Solo los responsables de criptografía (CO y PCO) pueden ejecutar este comando. Cuando el comando se ejecuta correctamente, crea el usuario en todos los HSM en el clúster.

Cómo solucionar problemas de createUser

Sin embargo, si configuración del HSM no es exacta, es posible que el usuario no se cree en todos los HSM. Para añadir el usuario a cualquier HSM en el que falte, utilice los comandos syncUser o createUser solo en los HSM en los que falte ese usuario. Para evitar errores de configuración, ejecute la herramienta configure con la opción `-m`.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Responsables de criptografía (CO, PRECO)

Sintaxis

Introduzca los argumentos en el orden especificado en el diagrama de sintaxis. Utilice el parámetro `-hpswd` para ocultar la contraseña. Para crear un usuario CO con autenticación de dos factores (2FA), utilice el parámetro `-2fa` e incluya una ruta de archivo. Para obtener más información, consulte the section called “Argumentos”.

```
createUser <user-type> <user-name> <password |-hpswd> [-2fa </path/to/authdata>]
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza createUser para crear usuarios nuevos en los HSM.

Example Creación de un responsable de criptografía

Este ejemplo crea a un responsable de criptografía (CO) en los HSM de un clúster. El primer comando utiliza loginHSM para iniciar sesión en los HSM como responsable de criptografía.

```
aws-cloudhsm> loginHSM CO admin 735782961

loginHSM success on server 0(10.0.0.1)
loginHSM success on server 1(10.0.0.2)
loginHSM success on server 1(10.0.0.3)
```

El segundo comando utiliza el comando createUser para crear `alice`, un nuevo responsable de criptografía en el HSM.

El mensaje de precaución explica que el comando crea usuarios en todos los HSM en el clúster. Sin embargo, si el comando produce un error en cualquier HSM, el usuario no existe en esos HSM. Para continuar, escriba `y`.

El resultado muestra que se creó el usuario nuevo en los tres HSM del clúster.

```
aws-cloudhsm> createUser CO alice 391019314

**********************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
```

```
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
*************************************************************

Do you want to continue(y/n)?Invalid option, please type 'y' or 'n'

Do you want to continue(y/n)?y
Creating User alice(CO) on 3 nodes
```

Cuando se completa el comando, `alice` tiene los mismos permisos en el HSM que el usuario CO `admin`, incluido el cambio de contraseña de cualquier usuario en los HSM.

El comando final utiliza el comando listUsers para verificar que `alice` existe en los tres HSM en el clúster. El resultado también muestra que se ha asignado a `alice` el ID de usuario `3`.. Utiliza el ID de usuario para identificarse `alice` en otros comandos, como findAllKeys.

```
aws-cloudhsm> listUsers
Users on server 0(10.0.0.1):
Number of users found:3

    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
         1              PRECO           admin                                YES
           0               NO
         2              AU              app_user                              NO
           0               NO
         3              CO              alice                                 NO
           0               NO
Users on server 1(10.0.0.2):
Number of users found:3

    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
         1              PRECO           admin                                YES
           0               NO
         2              AU              app_user                              NO
           0               NO
         3              CO              alice                                 NO
           0               NO

Users on server 1(10.0.0.3):
Number of users found:3
```

```
    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
        1               PRECO           admin                             YES
          0               NO
        2               AU              app_user                           NO
          0               NO
        3               CO              alice                              NO
          0               NO
```

Example : creación de un usuario de criptografía

Este ejemplo crea un usuario de criptografía (CU) bob, en el HSM. Los usuarios de criptografía pueden crear y administrar claves, pero no pueden administrar usuarios.

Después de escribir y para responder al mensaje de precaución, el resultado muestra que bob se creó en los tres HSM en el clúster. El nuevo CU puede iniciar sesión en el HSM para crear y administrar claves.

El comando utilizó el valor de contraseña `defaultPassword`. Más adelante, bob o cualquier CO pueden usar el comando changePswd para cambiar su contraseña.

```
aws-cloudhsm> createUser CU bob defaultPassword

************************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
****************************************************************

Do you want to continue(y/n)?Invalid option, please type 'y' or 'n'

Do you want to continue(y/n)?y
Creating User bob(CU) on 3 nodes
```

Argumentos

Introduzca los argumentos en el orden especificado en el diagrama de sintaxis. Utilice el parámetro `-hpswd` para ocultar la contraseña. Para crear un usuario CO con autenticación de dos factores (2FA), utilice el parámetro `-2fa` e incluya una ruta de archivo. Para obtener más información acerca de la 2FA, consulte Uso de la CMU para administrar la 2FA.

```
createUser <user-type> <user-name> <password |-hpswd> [-2fa </path/to/authdata>]
```

<user-type>

Especifica el tipo de usuario. Este parámetro es obligatorio.

Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Más información sobre los usuarios de HSM.

Valores válidos:

- CO: los responsables de criptografía pueden administrar usuarios, pero no pueden administrar claves.
- CU: los usuarios de criptografía pueden crear y administrar claves, y usar claves en operaciones criptográficas.

El PRECO se convierte en un CO cuando asigna una contraseña durante la activación de HSM.

Obligatorio: sí

<user-name>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_).

No puede cambiar el nombre de un usuario después de crearlo. En los comandos cloudhsm_mgmt_util, el tipo de usuario y la contraseña distinguen entre mayúsculas y minúsculas, pero el nombre de usuario no.

Obligatorio: sí

<password | -hpswd >

Especifica una contraseña para el usuario. Escriba una cadena de entre 7 y 32 caracteres. Este valor distingue entre mayúsculas y minúsculas. La contraseña aparece en texto no cifrado cuando se escribe. Para ocultar la contraseña, utilice el parámetro `-hpswd` en lugar de la contraseña y siga las indicaciones.

Para cambiar la contraseña de un usuario, utilice changePswd. Cualquier usuario de HSM puede cambiar su propia contraseña, pero los usuarios CO pueden cambiar la contraseña de cualquier usuario (de cualquier tipo) en los HSM.

Obligatorio: sí

[-2fa </path/to/authdata>]

Especifica la creación de un usuario CO con la 2FA habilitada. Para obtener los datos necesarios para configurar la autenticación de 2FA, incluya una ruta a una ubicación del sistema de archivos con un nombre de archivo después del parámetro `-2fa`. Para obtener información sobre la configuración y cómo trabajar con 2FA, consulte Uso de la CMU para administrar la 2FA.

Obligatorio: no

Temas relacionados de

- listUsers
- deleteUser
- syncUser
- changePswd

deleteUser

El comando deleteUser de cloudhsm_mgmt_util elimina un usuario de los módulos de seguridad de hardware (HSM). Solo los responsables de criptografía (CO) pueden ejecutar este comando. No puede eliminar un usuario que haya iniciado sesión actualmente en un HSM. Para obtener más información sobre cómo eliminar usuarios, consulte Cómo eliminar usuarios de HSM.

> **Tip**
>
> No puede eliminar a los usuarios de criptografía (CU) que poseen claves.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- CO

Sintaxis

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
deleteUser <user-type> <user-name>
```

Ejemplo

Este ejemplo elimina un responsable de criptografía (CO) de los HSM de un clúster. El primer comando utiliza listUsers para generar una lista de todos los usuarios de los HSM.

El resultado muestra que el usuario `3`, `alice`, es un CO en los HSM.

```
aws-cloudhsm> listUsers
Users on server 0(10.0.0.1):
Number of users found:3

    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
          1             PCO             admin                                YES
            0             NO
          2             AU              app_user                              NO
            0             NO
          3             CO              alice                                 NO
            0             NO
Users on server 1(10.0.0.2):
Number of users found:3

    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
          1             PCO             admin                                YES
            0             NO
          2             AU              app_user                              NO
            0             NO
          3             CO              alice                                 NO
            0             NO

Users on server 1(10.0.0.3):
Number of users found:3

    User Id             User Type       User Name                       MofnPubKey
  LoginFailureCnt         2FA
          1             PCO             admin                                YES
            0             NO
          2             AU              app_user                              NO
            0             NO
```

```
         3              CO             alice                                  NO
           0               NO
```

El segundo comando utiliza el comando deleteUser para eliminar `alice` de los HSM.

El resultado muestra que el comando se ha ejecutado correctamente en los tres HSM del clúster.

```
aws-cloudhsm> deleteUser CO alice
Deleting user alice(CO) on 3 nodes
deleteUser success on server 0(10.0.0.1)
deleteUser success on server 0(10.0.0.2)
deleteUser success on server 0(10.0.0.3)
```

El comando final utiliza el comando listUsers para verificar que `alice` se ha eliminado de los tres HSM del clúster.

```
aws-cloudhsm> listUsers
Users on server 0(10.0.0.1):
Number of users found:2

    User Id             User Type       User Name                          MofnPubKey
  LoginFailureCnt         2FA
         1              PCO             admin                                   YES
           0               NO
         2              AU              app_user                                NO
           0               NO
Users on server 1(10.0.0.2):
Number of users found:2

    User Id             User Type       User Name                          MofnPubKey
  LoginFailureCnt         2FA
         1              PCO             admin                                   YES
           0               NO
         2              AU              app_user                                NO
           0               NO
Users on server 1(10.0.0.3):
Number of users found:2

    User Id             User Type       User Name                          MofnPubKey
  LoginFailureCnt         2FA
         1              PCO             admin                                   YES
           0               NO
```

```
        2              AU              app_user                                NO
          0               NO
```

Argumentos

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
deleteUser <user-type> <user-name>
```

<user-type>

Especifica el tipo de usuario. Este parámetro es obligatorio.

> **Tip**
>
> No puede eliminar a los usuarios de criptografía (CU) que poseen claves.

Los valores válidos son CO, CU.

Para obtener el tipo de usuario, utilice listUsers. Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Más información sobre los usuarios de HSM.

Obligatorio: sí

<user-name>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_).

No puede cambiar el nombre de un usuario después de crearlo. En los comandos cloudhsm_mgmt_util, el tipo de usuario y la contraseña distinguen entre mayúsculas y minúsculas, pero el nombre de usuario no.

Obligatorio: sí

Temas relacionados de

- listUsers
- createUser

- syncUser
- changePswd

findAllKeys

El comando findAllKeys de cloudhsm_mgmt_util obtiene las claves que un usuario de criptografía (CU) especificado posee o comparte. También devuelve un valor hash de los datos de usuario en cada uno de los HSM. Puede utilizar el hash para determinar de un vistazo si los usuarios, la propiedad de la clave y los datos de uso compartido de la clave son los mismos en todos los HSM en el clúster. En la salida, las claves que son propiedad del usuario se marcan con `(o)`, mientras que las claves compartidas se marcan con `(s)`.

findAllKeys solo devuelve claves públicas cuando el CU especificado posee la clave, aunque todos los CU del HSM puedan utilizar cualquier clave pública. Este comportamiento es diferente del comando findKey de key_mgmt_util, que devuelve claves públicas para todos los usuarios de CU.

Solo los responsables de criptografía (CO y PCO) y los usuarios de dispositivos (AU) pueden ejecutar este comando. Los usuarios de criptografía (CU) pueden ejecutar los siguientes comandos:

- listUsers para encontrar todos los usuarios
- findKey en key_mgmt_util para encontrar las claves que pueden utilizar.
- getKeyInfoen key_mgmt_util para encontrar el propietario y los usuarios compartidos de una clave en particular que poseen o comparten

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Responsables de criptografía (CO, PCO)
- Usuarios de dispositivos (AU)

Sintaxis

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
findAllKeys <user id> <key hash (0/1)> [<output file>]
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza `findAllKeys` para encontrar todas las claves de un usuario y obtener un hash de la información del usuario de las claves en cada uno de los HSM.

Example : búsqueda de las claves para un CU

Este ejemplo utiliza findAllKeys para buscar las claves en los HSM que el usuario 4 posee y comparte. El comando utiliza un valor de `0` para el segundo argumento para suprimir el valor hash. Dado que se omite el nombre del archivo opcional, el comando escribe en stdout (salida estándar).

El resultado muestra que el usuario 4 puede utilizar 6 claves: 8, 9, 17, 262162, 19 y 31. La salida utiliza `(s)` para marcar las claves que el usuario comparte explícitamente. Las claves que posee el usuario se marcan con `(o)` y se componen tanto de claves simétricas y privadas que el usuario no comparte como de claves públicas que están disponibles para todos los usuarios de criptografía.

```
aws-cloudhsm> findAllKeys 4 0
Keys on server 0(10.0.0.1):
Number of keys found 6
number of keys matched from start index 0::6
8(s),9(s),17,262162(s),19(o),31(o)
findAllKeys success on server 0(10.0.0.1)

Keys on server 1(10.0.0.2):
Number of keys found 6
number of keys matched from start index 0::6
8(s),9(s),17,262162(s),19(o),31(o)
findAllKeys success on server 1(10.0.0.2)

Keys on server 1(10.0.0.3):
Number of keys found 6
number of keys matched from start index 0::6
8(s),9(s),17,262162(s),19(o),31(o)
findAllKeys success on server 1(10.0.0.3)
```

Example Verificación de que los datos del usuario están sincronizados

Este ejemplo utiliza findAllKeys para comprobar que todos los HSM del clúster contienen los mismos usuarios, la misma propiedad de las claves y los mismos valores de uso compartido de claves. Para ello, obtiene un hash de los datos de usuario de la clave en cada HSM y compara los valores hash.

Para obtener el hash de la clave, el comando utiliza un valor de 1 en el segundo argumento. El nombre del archivo opcional se omite, por lo que el comando escribe el hash de la clave en stdout.

El ejemplo especifica el usuario 6, pero el valor hash será el mismo para cualquier usuario que posea o comparta cualquiera de las claves en los HSM. Si el usuario especificado no posee ni comparte ninguna clave, como por ejemplo un CO, el comando no devuelve un valor hash.

El resultado muestra que el hash de la clave es idéntico para los dos HSM del clúster. Si uno de los HSM tuviera diferentes usuarios, diferentes propietarios de clave o diferentes usuarios compartidos, los valores del hash de la clave no serían iguales.

```
aws-cloudhsm> findAllKeys 6 1
Keys on server 0(10.0.0.1):
Number of keys found 3
number of keys matched from start index 0::3
8(s),9(s),11,17(s)
Key Hash:
55655676c95547fd4e82189a072ee1100eccfca6f10509077a0d6936a976bd49

findAllKeys success on server 0(10.0.0.1)
Keys on server 1(10.0.0.2):
Number of keys found 3
number of keys matched from start index 0::3
8(s),9(s),11(o),17(s)
Key Hash:
55655676c95547fd4e82189a072ee1100eccfca6f10509077a0d6936a976bd49

findAllKeys success on server 1(10.0.0.2)
```

Este comando demuestra que el valor hash representa los datos de usuario para todas las claves en el HSM. El comando utiliza findAllKeys para el usuario 3. A diferencia del usuario 6, que posee o comparte solo 3 claves, el usuario 3 es propietario o comparte 17 claves, pero el valor del hash de la clave es el mismo.

```
aws-cloudhsm> findAllKeys 3 1
Keys on server 0(10.0.0.1):
```

```
Number of keys found 17
number of keys matched from start index 0::17
6(o),7(o),8(s),11(o),12(o),14(o),262159(o),262160(o),17(s),262162(s),19(s),20(o),21(o),262177(o
Key Hash:
55655676c95547fd4e82189a072ee1100eccfca6f10509077a0d6936a976bd49

findAllKeys success on server 0(10.0.0.1)
Keys on server 1(10.0.0.2):
Number of keys found 17
number of keys matched from start index 0::17
6(o),7(o),8(s),11(o),12(o),14(o),262159(o),262160(o),17(s),262162(s),19(s),20(o),21(o),262177(o
Key Hash:
55655676c95547fd4e82189a072ee1100eccfca6f10509077a0d6936a976bd49

findAllKeys success on server 1(10.0.0.2)
```

Argumentos

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
findAllKeys <user id> <key hash (0/1)> [<output file>]
```

<user id>

Obtiene todas las claves que el usuario especificado posee o comparte. Escriba el ID del usuario en los HSM. Para encontrar los ID de todos los usuarios, utilice listUsers.

Todos los ID de usuario son válidos, pero `findAllKeys` solamente devuelve las claves de los usuarios de criptografía (CU).

Obligatorio: sí

<key hash>

Incluye (1) o excluye (0) un hash de la propiedad del usuario y de los datos de uso compartido para todas las claves en cada HSM.

Cuando el argumento `user id` representa a un usuario que posee o comparte claves, se rellena el hash de la clave. El valor del hash de la clave es idéntico para todos los usuarios que poseen o comparten claves en el HSM, aunque posean y compartan claves diferentes. Sin embargo, cuando el `user id` representa a un usuario que no posee ni comparte ninguna clave, como un CO, el valor hash no se rellena.

Obligatorio: sí

<output file>

Escribe la salida en el archivo especificado.

Obligatorio: no

Valor predeterminado: stdout

Temas relacionados de

- changePswd
- deleteUser
- listUsers
- syncUser
- findKey en key_mgmt_util
- getKeyInfoen key_mgmt_util

getAttribute

El comando getAttribute en cloudhsm_mgmt_util obtiene el valor de un atributo de una clave para todos los HSM del clúster y lo escribe en stdout (salida estándar) o en un archivo. Solo los usuarios de criptografía (CU) pueden ejecutar este comando.

Los atributos de la clave son las propiedades de una clave. Contienen características, como el tipo de clave, clase, etiqueta e ID y los valores que representan las acciones que puede desempeñar en la clave, como cifrar, descifrar, encapsular, firmar y verificar.

Solamente puede utilizar getAttribute en claves que sean de su propiedad y que hayan compartido con usted. Puede ejecutar este comando o el comando getAttribute en key_mgmt_util, que escribe uno o todos los valores de atributo de una clave en un archivo.

Para obtener una lista de los atributos y las constantes que los representan, ejecute el comando listAttributes. Para cambiar los valores de los atributos de las claves existentes, utilice setAttribute en key_mgmt_util y setAttribute en cloudhsm_mgmt_util. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Sintaxis

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
getAttribute <key handle> <attribute id> [<filename>]
```

Ejemplo

En este ejemplo se obtiene el valor del atributo extraíble de una clave de los HSM. Puede utilizar un comando de este tipo para determinar si puede exportar una clave desde los HSM.

El primer comando utiliza listAttributes para encontrar la constante que representa el atributo extraíble. La salida muestra que la constante de `OBJ_ATTR_EXTRACTABLE` es `354`. Encontrará también esta información con las descripciones de los atributos y sus valores, en la Referencia de los atributos de claves.

```
aws-cloudhsm> listAttributes

Following are the possible attribute values for getAttribute:

      OBJ_ATTR_CLASS                  = 0
      OBJ_ATTR_TOKEN                  = 1
      OBJ_ATTR_PRIVATE                = 2
      OBJ_ATTR_LABEL                  = 3
      OBJ_ATTR_TRUSTED                = 134
      OBJ_ATTR_KEY_TYPE               = 256
```

```
      OBJ_ATTR_ID                     = 258
      OBJ_ATTR_SENSITIVE              = 259
      OBJ_ATTR_ENCRYPT                = 260
      OBJ_ATTR_DECRYPT                = 261
      OBJ_ATTR_WRAP                   = 262
      OBJ_ATTR_UNWRAP                 = 263
      OBJ_ATTR_SIGN                   = 264
      OBJ_ATTR_VERIFY                 = 266
      OBJ_ATTR_DERIVE                 = 268
      OBJ_ATTR_LOCAL                  = 355
      OBJ_ATTR_MODULUS                = 288
      OBJ_ATTR_MODULUS_BITS           = 289
      OBJ_ATTR_PUBLIC_EXPONENT        = 290
      OBJ_ATTR_VALUE_LEN              = 353
      OBJ_ATTR_EXTRACTABLE            = 354
      OBJ_ATTR_NEVER_EXTRACTABLE      = 356
      OBJ_ATTR_ALWAYS_SENSITIVE       = 357
      OBJ_ATTR_DESTROYABLE            = 370
      OBJ_ATTR_KCV                    = 371
      OBJ_ATTR_WRAP_WITH_TRUSTED      = 528
      OBJ_ATTR_WRAP_TEMPLATE          = 1073742353
      OBJ_ATTR_UNWRAP_TEMPLATE        = 1073742354
      OBJ_ATTR_ALL                    = 512
```

El segundo comando utiliza getAttribute para obtener el valor del atributo extraíble de la clave que tiene el identificador de clave `262170` en los HSM. Para especificar el atributo extraíble, el comando utiliza `354`, la constante que representa el atributo. Como el comando no especifica el nombre de archivo, getAttribute escribe la salida en stdout.

La salida muestra que el valor del atributo extraíble es 1 en todos los HSM. Este valor indica que el propietario de la clave puede exportarla. Cuando el valor es 0 (0x0), no se puede exportar desde los HSM. El valor del atributo extraíble se establece al crear la clave, pero no se puede cambiar.

```
aws-cloudhsm> getAttribute 262170 354

Attribute Value on server 0(10.0.1.10):
OBJ_ATTR_EXTRACTABLE
0x00000001

Attribute Value on server 1(10.0.1.12):
OBJ_ATTR_EXTRACTABLE
0x00000001
```

```
Attribute Value on server 2(10.0.1.7):
OBJ_ATTR_EXTRACTABLE
0x00000001
```

Argumentos

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
getAttribute <key handle> <attribute id> [<filename>]
```

<key-handle>

Especifica el identificador de la clave de destino. Puede especificar una única clave en cada comando. Para obtener el identificador de una clave, use findKey en key_mgmt_util.

Debe ser propietario de la clave especificada o esta debe compartirse con usted. Para encontrar los usuarios de una clave, utilice getKeyInfokey_mgmt_util.

Obligatorio: sí

<attribute id>

Identifica el atributo. Escriba una constante que represente un atributo o 512, que representa todos los atributos. Por ejemplo, para obtener el tipo de clave, especifique 256, que es la constante del atributo OBJ_ATTR_KEY_TYPE.

Para generar una lista de los atributos y sus constantes, utilice listAttributes. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Obligatorio: sí

<filename>

Escribe la salida en el archivo especificado. Escriba una ruta de archivo.

Si el archivo especificado existe, getAttribute sobrescribe el archivo sin ningún tipo de advertencia.

Obligatorio: no

Valor predeterminado: stdout

Temas relacionados de

- getAttribute en key_mgmt_util
- listAttributes
- setAttribute en cloudhsm_mgmt_util
- setAttribute en key_mgmt_util
- Referencia de los atributos de claves

getCert

Con el comando getCert de cloudhsm_mgmt_util, puede recuperar los certificados de un HSM específico de un clúster. Al ejecutar el comando, debe designar el tipo de certificado que desea recuperar. Para ello, utilice el número entero correspondiente que se describe más adelante en la sección Argumentos. Para obtener más información sobre la función de cada uno de estos certificados, consulte Verificar la identidad del HSM.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios.

Requisitos previos

Antes de comenzar, debe entrar en el modo de servidor en el HSM de destino. Para obtener más información, consulte server.

Sintaxis

Para utilizar el comando getCert una vez en modo de servidor:

```
server> getCert <file-name> <certificate-type>
```

Ejemplo

En primer lugar, entre el modo de servidor. Este comando entra en el modo de servidor en un HSM con el número de servidor `0`.

```
aws-cloudhsm> server 0

Server is in 'E2' mode...
```

A continuación, utilice el comando getCert. En este ejemplo, utilizamos `/tmp/P0.crt` como nombre del archivo en el que se guardará el certificado y `4` (Certificado raíz del cliente) como tipo de certificado deseado:

```
server0> getCert /tmp/P0.crt 4
getCert Success
```

Argumentos

```
getCert <file-name> <certificate-type>
```

<file-name>

Especifica el nombre del archivo en el que se va a guardar el certificado.

Obligatorio: sí

<certificate-type>

Un número entero que especifica el tipo de certificado que se desea recuperar. Los números enteros y sus correspondientes tipos de certificados son los siguientes:

- 1: certificado raíz del fabricante
- 2: certificado de hardware del fabricante
- 4: certificado raíz del cliente
- 8: certificado del clúster (firmado por el certificado raíz del cliente)
- 16: certificado del clúster (enlazado al certificado raíz del fabricante)

Obligatorio: sí

Temas relacionados de

- servidor

getHSMInfo

El comando getHSMInfo de cloudhsm_mgmt_util obtiene información sobre el hardware en el que se ejecuta cada HSM, incluido el modelo, el número de serie, el estado de FIPS, la memoria, la temperatura y los números de versión del hardware y el firmware. La información también contiene el ID del servidor que cloudhsm_mgmt_util utiliza para consultar el HSM.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios. No tiene que haber iniciado sesión para ejecutar este comando.

Sintaxis

Este comando no tiene parámetros.

```
getHSMInfo
```

Ejemplo

En este ejemplo, se utiliza getHSMInfo para obtener información acerca de los HSM del clúster.

```
aws-cloudhsm> getHSMInfo
Getting HSM Info on 3 nodes
                *** Server 0 HSM Info ***

        Label                :cavium
```

```
        Model                :NITROX-III CNN35XX-NFBE

        Serial Number        :3.0A0101-ICM000001
        HSM Flags            :0
        FIPS state           :2 [FIPS mode with single factor authentication]

        Manufacturer ID      :
        Device ID            :10
        Class Code           :100000
        System vendor ID     :177D
        SubSystem ID         :10


        TotalPublicMemory    :560596
        FreePublicMemory     :294568
        TotalPrivateMemory   :0
        FreePrivateMemory    :0

        Hardware Major       :3
        Hardware Minor       :0

        Firmware Major       :2
        Firmware Minor       :03

        Temperature          :56 C

        Build Number         :13

        Firmware ID          :xxxxxxxxxxxxxxx

...
```

Temas relacionados de

- info

getKeyInfo

El comando getKeyInfo en la key_mgmt_util devuelve los ID de los usuarios de HSM que pueden utilizar la clave, incluidos el propietario y los usuarios de criptografía (CU) con quienes se comparte la clave. Cuando la autenticación de cuórum está habilitada en una clave, getKeyInfo también devuelve el número de usuarios que deben aprobar las operaciones criptográficas que utilizan la

clave. Solamente puede ejecutar getKeyInfo en las claves que son de su propiedad y han compartido con usted.

Cuando ejecuta getKeyInfo en claves públicas, getKeyInfo solamente devuelve el propietario de la clave, aunque todos los usuarios del HSM puedan utilizar la clave pública. Para encontrar los ID de HSM de los usuarios en sus HSM, utilice listUsers. Para buscar las claves de un usuario concreto, utilice findKey de `-u` en key_mgmt_util. Los oficiales de cifrado pueden usar findAllKeyscloudhsm_mgmt_util.

Es propietario de las claves que crea. Puede compartir una clave con otros usuarios cuando la crea. A continuación, para compartir o dejar de compartir una clave existente, utilice shareKey en cloudhsm_mgmt_util.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Sintaxis

```
getKeyInfo -k <key-handle> [<output file>]
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza getKeyInfo para obtener información sobre los usuarios de una clave.

Example : obtención de los usuarios de una clave asimétrica

Este comando obtiene los usuarios que pueden utilizar la clave AES (asimétrica) con identificador de clave 262162. El resultado muestra que el usuario 3 es propietario de la clave y la comparte con los usuarios 4 y 6.

Solo los usuarios 3, 4 y 6 pueden ejecutar getKeyInfo en la clave 262162.

```
aws-cloudhsm>getKeyInfo 262162
Key Info on server 0(10.0.0.1):

        Token/Flash Key,

        Owned by user 3

        also, shared to following 2 user(s):

                4
                6
Key Info on server 1(10.0.0.2):

        Token/Flash Key,

        Owned by user 3

        also, shared to following 2 user(s):

                4
                6
```

Example : obtención de los usuarios de un par de claves simétricas

Estos comandos utilizan getKeyInfo para obtener los usuarios que pueden utilizar las claves de un par de claves ECC (simétricas). La clave pública tiene el identificador de clave 262179. La clave privada tiene el identificador de clave 262177.

Cuando ejecuta getKeyInfo en la clave privada (262177), devuelve el propietario de la clave (3) y los usuarios de criptografía (CU) 4, con quienes se comparte la clave.

```
aws-cloudhsm>getKeyInfo -k 262177
Key Info on server 0(10.0.0.1):

        Token/Flash Key,

        Owned by user 3

        also, shared to following 1 user(s):

                4
```

```
Key Info on server 1(10.0.0.2):

        Token/Flash Key,

        Owned by user 3

        also, shared to following 1 user(s):

                4
```

Cuando ejecuta getKeyInfo en la clave pública (262179), devuelve solo el propietario de la clave, el usuario 3.

```
aws-cloudhsm>getKeyInfo -k 262179
Key Info on server 0(10.0.3.10):

        Token/Flash Key,

        Owned by user 3

Key Info on server 1(10.0.3.6):

        Token/Flash Key,

        Owned by user 3
```

Para confirmar que el usuario 4 puede utilizar la clave pública (y todas las claves públicas en el HSM), utilice el parámetro `-u` de findKey en key_mgmt_util.

El resultado muestra que el usuario 4 puede utilizar la clave pública (262179) y la clave privada (262177) en el par de claves. El usuario 4 también puede utilizar todas las demás claves públicas y cualquier clave privada creadas o que se hayan compartido con ellos.

```
Command:  findKey -u 4

Total number of keys present 8

 number of keys matched from start index 0::7
11, 12, 262159, 262161, 262162, 19, 20, 21, 262177, 262179

        Cluster Error Status
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

```
        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

Example : obtención del valor de autenticación de cuórum (m_value) para una clave

Este ejemplo muestra cómo obtener `m_value` para una clave. El m_value es el número de usuarios en el cuórum que debe aprobar las operaciones criptográficas que utilizan la clave y las operaciones para compartir y dejar de compartir la clave.

Cuando la autenticación de cuórum está habilitada en una clave, un cuórum de usuarios debe aprobar cualquier operación criptográfica que utilice la clave. Para habilitar la autenticación de cuórum y establecer el tamaño de cuórum, utilice el parámetro `-m_value` al crear la clave.

Este comando se utiliza genSymKeypara crear una clave AES de 256 bits que se comparte con el usuario 4. Utiliza el parámetro `m_value` para habilitar la autenticación de cuórum y establecer el tamaño de cuórum en dos usuarios. El número de usuarios debe ser lo suficientemente grande como para proporcionar las aprobaciones necesarias.

El resultado muestra que el comando ha creado la clave 10.

```
Command:  genSymKey -t 31 -s 32 -l aes256m2 -u 4 -m_value 2

        Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Created.  Key Handle: 10

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Este comando utiliza getKeyInfo en cloudhsm_mgmt_util para obtener información sobre los usuarios de la clave `10`. El resultado muestra que la clave es propiedad del usuario `3` y se comparte con el usuario `4`. También muestra que un cuórum de dos usuarios debe aprobar todas las operaciones criptográficas que utilizan la clave.

```
aws-cloudhsm>getKeyInfo 10

Key Info on server 0(10.0.0.1):

        Token/Flash Key,
```

```
        Owned by user 3

        also, shared to following 1 user(s):

                4
         2 Users need to approve to use/manage this key
Key Info on server 1(10.0.0.2):

        Token/Flash Key,

        Owned by user 3

        also, shared to following 1 user(s):

                4
         2 Users need to approve to use/manage this key
```

Argumentos

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
getKeyInfo -k <key-handle> <output file>
```

<key-handle>

Especifica el identificador de una clave en el HSM. Escriba el identificador de una clave de su propiedad o que comparte. Este parámetro es obligatorio.

Obligatorio: sí

<output file>

Escribe la salida en el archivo especificado, en lugar de stdout. Si el archivo existe, el comando lo sobrescribe sin ningún tipo de advertencia.

Obligatorio: no

Valor predeterminado: stdout

Temas relacionados de

- getKeyInfoen key_mgmt_util

- findKey en key_mgmt_util
- findAllKeysen cloudhsm_mgmt_util
- listUsers
- shareKey

info

El comando info de cloudhsm_mgmt_util obtiene información sobre cada uno de los HSM del clúster, incluido el nombre de host, el puerto, la dirección IP y el nombre y el tipo de usuario que ha iniciado sesión en cloudhsm_mgmt_util del HSM.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios. No tiene que haber iniciado sesión para ejecutar este comando.

Sintaxis

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
info server <server ID>
```

Ejemplo

En este ejemplo, se utiliza info para obtener información acerca de un HSM del clúster. El comando utiliza 0 para consultar el primer HSM del clúster. La salida muestra la dirección IP, el puerto y el tipo y el nombre del usuario actual.

```
aws-cloudhsm> info server 0
```

```
Id      Name                    Hostname         Port   State          Partition
         LoginState
0       10.0.0.1                10.0.0.1        2225    Connected      hsm-udw0tkfg1ab
         Logged in as 'testuser(CU)'
```

Argumentos

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
info server <server ID>
```

<server id>

Especifica el ID de servidor del HSM. El sistema asigna a los HSM números ordinales que representan el orden en el que se añaden al clúster, comenzando por 0. Para encontrar el ID de servidor de un HSM, utilice getHSMInfo.

Obligatorio: sí

Temas relacionados de

- getHSMInfo
- loginHSM y logoutHSM

listAttributes

El listAttributes comando de cloudhsm_mgmt_util muestra los atributos de una clave y las constantes que los representan. AWS CloudHSM Usted utiliza estas constantes para identificar los atributos en los comandos getAttribute y setAttribute.

Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios. No tiene que haber iniciado sesión para ejecutar este comando.

Sintaxis

```
listAttributes [-h]
```

Ejemplo

Este comando enumera los atributos de clave que se pueden obtener y cambiar en key_mgmt_util, así como las constantes que los representan. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves. Para representar todos los atributos, utilice 512.

```
Command: listAttributes

    Description
===========
The following are all of the possible attribute values for getAttribute.

      OBJ_ATTR_CLASS                = 0
      OBJ_ATTR_TOKEN                = 1
      OBJ_ATTR_PRIVATE              = 2
      OBJ_ATTR_LABEL                = 3
      OBJ_ATTR_TRUSTED              = 134
      OBJ_ATTR_KEY_TYPE             = 256
      OBJ_ATTR_ID                   = 258
      OBJ_ATTR_SENSITIVE            = 259
      OBJ_ATTR_ENCRYPT              = 260
      OBJ_ATTR_DECRYPT              = 261
      OBJ_ATTR_WRAP                 = 262
      OBJ_ATTR_UNWRAP               = 263
      OBJ_ATTR_SIGN                 = 264
      OBJ_ATTR_VERIFY               = 266
      OBJ_ATTR_DERIVE               = 268
      OBJ_ATTR_LOCAL                = 355
      OBJ_ATTR_MODULUS              = 288
      OBJ_ATTR_MODULUS_BITS         = 289
      OBJ_ATTR_PUBLIC_EXPONENT      = 290
      OBJ_ATTR_VALUE_LEN            = 353
      OBJ_ATTR_EXTRACTABLE          = 354
      OBJ_ATTR_NEVER_EXTRACTABLE    = 356
      OBJ_ATTR_ALWAYS_SENSITIVE     = 357
```

```
OBJ_ATTR_DESTROYABLE            = 370
OBJ_ATTR_KCV                     = 371
OBJ_ATTR_WRAP_WITH_TRUSTED       = 528
OBJ_ATTR_WRAP_TEMPLATE           = 1073742353
OBJ_ATTR_UNWRAP_TEMPLATE        = 1073742354
OBJ_ATTR_ALL                    = 512
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

Temas relacionados de

- getAttribute
- setAttribute
- Referencia de los atributos de claves

listUsers

El comando listUsers de cloudhsm_mgmt_util obtiene los usuarios de cada HSM, junto con el tipo de usuario y otros atributos. Este comando lo pueden ejecutar todo tipo de usuarios. Ni siquiera es preciso haber iniciado sesión en cloudhsm_mgmt_util para ejecutar este comando.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios. No es preciso haber iniciado sesión para ejecutar este comando.

Sintaxis

Este comando no tiene parámetros.

```
listUsers
```

Ejemplo

Este comando enumera los usuarios de cada uno de los HSM del clúster y muestra sus atributos. Puede utilizar el atributo `User ID` para identificar a los usuarios en otros comandos, como deleteUser, changePswd y findAllKeys.

```
aws-cloudhsm> listUsers
Users on server 0(10.0.0.1):
Number of users found:6

    User Id             User Type       User Name            MofnPubKey
 LoginFailureCnt         2FA
        1              PCO             admin                    YES              0
            NO
        2              AU              app_user                  NO              0
            NO
        3              CU              crypto_user1              NO              0
            NO
        4              CU              crypto_user2              NO              0
            NO
        5              CO              officer1                 YES              0
            NO
        6              CO              officer2                  NO              0
            NO
Users on server 1(10.0.0.2):
Number of users found:5

    User Id             User Type       User Name            MofnPubKey
 LoginFailureCnt         2FA
        1              PCO             admin                    YES              0
            NO
        2              AU              app_user                  NO              0
            NO
        3              CU              crypto_user1              NO              0
            NO
        4              CU              crypto_user2              NO              0
            NO
```

```
        5              CO             officer1                YES              0
          NO
```

La salida contiene los siguientes atributos de usuario:

- ID de usuario: identifica al usuario de los comandos key_mgmt_util y cloudhsm_mgmt_util.
- User type (Tipo de usuario): determina las operaciones que el usuario puede realizar en el HSM.
- User Name (Nombre de usuario): muestra el nombre fácil de recordar definido por el usuario para el usuario.
- MofnPubKey: Indica si el usuario ha registrado un key pair para firmar los tokens de autenticación de quórum.
- LoginFailureCnt: Indica el número de veces que el usuario ha iniciado sesión sin éxito.
- 2FA: indica que el usuario ha activado la autenticación multifactor.

Temas relacionados de

- listUsers en key_mgmt_util
- createUser
- deleteUser
- changePswd

loginHSM y logoutHSM

Puede usar los comandos loginHSM y logoutHSM en cloudhsm_mgmt_util para iniciar y cerrar sesión en cada HSM de un clúster. Todos los usuarios del tipo que sean pueden utilizar estos comandos.

> **Note**
>
> Si se superan cinco intentos de inicio de sesión incorrectos, se bloquea la cuenta. Para desbloquear la cuenta, un responsable de criptografía (CO) debe restablecer la contraseña mediante el comando changePswd en cloudhsm_mgmt_util.

Solución de problemas de loginHSM y logoutHSM

Antes de que ejecute estos comandos cloudhsm_mgmt_util, debe iniciar cloudhsm_mgmt_util.

Si agrega o elimina los HSM, actualice los archivos de configuración que utilizan el AWS CloudHSM cliente y las herramientas de línea de comandos. De lo contrario, es posible que los cambios que realice no se hagan efectivos en todos los HSM del clúster.

Si tiene más de un HSM en el clúster, es posible que puedan realizarse intentos adicionales de inicio de sesión incorrectos antes de que se bloquee la cuenta. Esto se debe a que el cliente CloudHSM equilibra la carga entre los diversos HSM. Por lo tanto, el intento de inicio de sesión no puede comenzar en el mismo HSM cada vez. Si va a probar esta funcionalidad, recomendamos que lo haga en un clúster con un solo HSM activo.

Si creó el clúster antes de febrero de 2018, la cuenta se bloquea después de 20 intentos de inicio de sesión incorrectos.

Tipo de usuario

Los usuarios siguientes pueden ejecutar estos comandos.

- Responsable de criptografía previa (PRECO)
- Responsable de criptografía (CO)
- Usuario de criptografía (CU)

Sintaxis

Introduzca los argumentos en el orden especificado en el diagrama de sintaxis. Utilice el parámetro `-hpswd` para ocultar la contraseña. Para iniciar sesión con la autenticación de dos factores (2FA), utilice el parámetro `-2fa` e incluya una ruta de archivo. Para obtener más información, consulte the section called “Argumentos”.

```
loginHSM <user-type> <user-name> <password |-hpswd> [-2fa </path/to/authdata>]
```

```
logoutHSM
```

Ejemplos

En estos ejemplos, se muestra cómo utilizar loginHSM y logoutHSM para iniciar y cerrar sesión en todos los HSM de un clúster.

Example : inicio de sesión en los HSM de un clúster

Este comando inicia sesión en todos los HSM de un clúster con las credenciales de un usuario de CO llamado `admin` y con la contraseña `co12345`. El resultado muestra que el comando se ha ejecutado correctamente y que se ha conectado a los HSM (que, en este caso, son `server 0` y `server 1`).

```
aws-cloudhsm>loginHSM CO admin co12345

loginHSM success on server 0(10.0.2.9)
loginHSM success on server 1(10.0.3.11)
```

Example : inicio de sesión con una contraseña oculta

Este comando es el mismo que en el ejemplo anterior, salvo que esta vez se especifica que el sistema debe ocultar la contraseña.

```
aws-cloudhsm>loginHSM CO admin -hpswd
```

El sistema le solicitará su contraseña. Introduzca la contraseña. El sistema la ocultará, y el resultado mostrará que el comando se ha ejecutado correctamente y se ha conectado a los HSM.

```
Enter password:

loginHSM success on server 0(10.0.2.9)
loginHSM success on server 1(10.0.3.11)

aws-cloudhsm>
```

Example : cierre de sesión de un HSM

Este comando cierra la sesión de los HSM en los que tenga la sesión iniciada (que, en este caso, son `server 0` y `server 1`). El resultado muestra que el comando se ha ejecutado correctamente y que usted se ha desconectado de los HSM.

```
aws-cloudhsm>logoutHSM

logoutHSM success on server 0(10.0.2.9)
logoutHSM success on server 1(10.0.3.11)
```

Argumentos

Introduzca los argumentos en el orden especificado en el diagrama de sintaxis. Utilice el parámetro `-hpswd` para ocultar la contraseña. Para iniciar sesión con la autenticación de dos factores (2FA), utilice el parámetro `-2fa` e incluya una ruta de archivo. Para obtener más información acerca de cómo trabajar con la 2FA, consulte Uso de la CMU para administrar la 2FA

```
loginHSM <user-type> <user-name> <password |-hpswd> [-2fa </path/to/authdata>]
```

<user type>

Especifica el tipo de usuario que inicia sesión en los HSM. Para obtener más información, consulte Tipo de usuario más arriba.

Obligatorio: sí

<user name>

Especifica el nombre de usuario del usuario que va a iniciar sesión en los HSM.

Obligatorio: sí

<password | -hpswd >

Especifica la contraseña del usuario que va a iniciar sesión en los HSM. Para ocultar la contraseña, utilice el parámetro `-hpswd` en lugar de la contraseña y siga las indicaciones.

Obligatorio: sí

[-2fa </path/to/authdata>]

Especifica que el sistema debe utilizar un segundo factor para autenticar a este usuario de CO con 2FA habilitado. Para obtener los datos necesarios para iniciar sesión con la 2FA, incluya una ruta a una ubicación en el sistema de archivos con un nombre de archivo después del parámetro `-2fa`. Para obtener más información acerca de cómo trabajar con la 2FA, consulte Uso de la CMU para administrar la 2FA.

Obligatorio: no

Temas relacionados de

- Introducción a cloudhsm_mgmt_util

- Activación del clúster

registerQuorumPubClave

El comando registerQuorumPubKey en cloudhsm_mgmt_util asocia usuarios del módulo de seguridad de hardware (HSM) con pares de claves asimétricas RSA-2048. Una vez que asocie los usuarios de HSM a las claves, esos usuarios pueden usar la clave privada para aprobar las solicitudes de cuórum y el clúster puede usar la clave pública registrada para comprobar que la firma proviene del usuario. Para obtener más información sobre la autenticación de cuórum, consulte Administrar la autenticación de cuórum (control de acceso M de N).

> **Tip**
>
> En la AWS CloudHSM documentación, la autenticación de quórum a veces se denomina M de N (MoFN), lo que significa un mínimo de M aprobadores de un número total de N aprobadores.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Responsables de criptografía (CO)

Sintaxis

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
registerQuorumPubKey <user-type> <user-name> <registration-token> <signed-registration-token> <public-key>
```

Ejemplos

En este ejemplo, se muestra cómo utilizar registerQuorumPubKey para registrar los responsables de criptografía (CO) como aprobadores en las solicitudes de autenticación por cuórum. Para ejecutar este comando, debe tener un par de claves asimétrico RSA-2048, un token firmado y un token no firmado. Para obtener más información acerca de los requisitos, consulte the section called “Argumentos”.

Example : Registre un usuario de HSM para la autenticación de quórum

En este ejemplo se registra un CO denominado `quorum_officer` como aprobador de la autenticación de cuórum.

```
aws-cloudhsm> registerQuorumPubKey CO <quorum_officer> </path/to/quorum_officer.token>
 </path/to/quorum_officer.token.sig> </path/to/quorum_officer.pub>

*************************CAUTION*******************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
***************************************************************

Do you want to continue(y/n)?y
registerQuorumPubKey success on server 0(10.0.0.1)
```

El comando final usa el comando listUsers para comprobar que `quorum_officer` está registrado como usuario de MoFN.

```
aws-cloudhsm> listUsers
Users on server 0(10.0.0.1):
Number of users found:3

    User Id             User Type       User Name                      MofnPubKey
  LoginFailureCnt         2FA
        1               PCO             admin                           NO
          0               NO
        2               AU              app_user                        NO
          0               NO
        3               CO              quorum_officer                  YES
          0               NO
```

Argumentos

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
registerQuorumPubKey <user-type> <user-name> <registration-token> <signed-registration-
token> <public-key>
```

<user-type>

Especifica el tipo de usuario. Este parámetro es obligatorio.

Para obtener información detallada sobre los tipos de usuario en un HSM, consulte Más información sobre los usuarios de HSM.

Valores válidos:

- CO: los responsables de criptografía pueden administrar usuarios, pero no pueden administrar claves.

Obligatorio: sí

<user-name>

Especifica un nombre fácil de recordar para el usuario. La longitud máxima es de 31 caracteres. El único carácter especial permitido es un guion bajo (_).

No puede cambiar el nombre de un usuario después de crearlo. En los comandos cloudhsm_mgmt_util, el tipo de usuario y la contraseña distinguen entre mayúsculas y minúsculas, pero el nombre de usuario no.

Obligatorio: sí

<registration-token>

Especifica la ruta a un archivo que contiene un token de registro sin firmar. Puede contener cualquier dato aleatorio con un tamaño máximo de archivo de 245 bytes. Para obtener más información sobre la creación de un token de registro sin firmar, consulte Crear y firmar un token de registro.

Obligatorio: sí

<signed-registration-token>

Especifica la ruta a un archivo que contiene el hash firmado por el mecanismo SHA256_PKCS del token de registro. Para obtener más información, consulte Crear y firmar un token de registro.

Obligatorio: sí

<public-key>

Especifica la ruta a un archivo que contiene la clave pública de un par de claves RSA-2048 asimétricas. Utilice la clave privada para firmar el token de registro. Para obtener más información, consulte Crear un par de claves RSA.

Obligatorio: sí

> **Note**
>
> El clúster usa la misma clave para la autenticación de cuórum y para la autenticación de dos factores (2FA). Esto significa que no puede rotar una clave de cuórum para un usuario que tenga habilitada la autenticación de dos factores con registerQuorumPubKey. Para rotar la clave, debe usar changePswd. Para obtener más información sobre el uso de la autenticación de cuórum y la 2FA, consulte Autenticación de cuórum y 2FA.

Temas relacionados de

- Creación de un par de claves RSA
- Creación y firma de un token de registro
- Registro de una clave pública con HSM
- Aplicación de la autenticación de cuórum (control de acceso M de N)
- Autenticación de cuórum y 2FA
- listUsers

server

Normalmente, cuando se emite un comando en cloudhsm_mgmt_util, el comando afecta a todos los HSM del clúster designado (modo global). Sin embargo, puede haber ocasiones en las que necesite emitir comandos en un único HSM. Por ejemplo, si se produce un error en la sincronización automática, puede que tenga que sincronizar las claves y los usuarios de un HSM con el fin de mantener la coherencia en el clúster. Puede utilizar el comando server de cloudhsm_mgmt_util para entrar en el server mode e interactuar directamente con una instancia específica de HSM.

Tras la inicialización correcta, el símbolo del sistema `aws-cloudhsm>` se sustituye por el símbolo del sistema `server>`.

Para salir del modo de servidor, utilice el comando `exit`. Después de salir correctamente, volverá al símbolo del sistema de cloudhsm_mgmt_util.

Antes de ejecutar cualquier comando de cloudhsm_mgmt_util, debe iniciar cloudhsm_mgmt_util.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios.

Requisitos previos

Para entrar en el modo de servidor, primero debe conocer el número de servidor del HSM de destino. Los números de servidor son los que se muestran en la salida de rastreo generada por cloudhsm_mgmt_util al iniciarse. Los números de servidor se asignan en el mismo orden en que aparecen los HSM en el archivo de configuración. En este ejemplo, suponemos que `server 0` es el servidor que se corresponde con el HSM deseado.

Sintaxis

Para entrar en el modo de servidor:

```
server <server-number>
```

Para salir del modo de servidor:

```
server> exit
```

Ejemplo

Este comando entra en el modo de servidor en un HSM con el número de servidor `0`.

```
aws-cloudhsm> server 0

Server is in 'E2' mode...
```

Para salir del modo de servidor, utilice el comando exit.

```
server0> exit
```

Argumentos

```
server <server-number>
```

<server-number>

Especifica el número de servidor del HSM de destino.

Obligatorio: sí

El comando `exit` no tiene argumentos.

Temas relacionados de

- syncKey
- createUser
- deleteUser

setAttribute

El comando setAttribute de cloudhsm_mgmt_util cambia el valor de la etiqueta y cifra, descifra, encapsula y desencapsula atributos de una clave en los HSM. También puede utilizar el comando setAttribute de key_mgmt_util para convertir la clave de una sesión en una clave persistente. Solo puede cambiar los atributos de claves de su propiedad.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Sintaxis

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
setAttribute <key handle> <attribute id>
```

Ejemplo

En este ejemplo, se muestra cómo deshabilitar la funcionalidad de descifrar de una clave simétrica. Puede utilizar un comando como este para configurar una clave de encapsulamiento que debe poder encapsular y desencapsular otras claves, pero no cifrar ni descifrar datos.

El primer paso consiste en crear la clave de encapsulamiento. Este comando usa genSymKeykey_mgmt_util para generar una clave simétrica AES de 256 bits. La salida muestra que la nueva clave tiene el identificador de clave 14.

```
$  genSymKey -t 31 -s 32 -l aes256

Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Created.  Key Handle: 14

        Cluster Error Status
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

A continuación, queremos confirmar el valor actual del atributo de descifrar. Para obtener el ID de atributo del atributo de descifrar, utilice listAttributes. La salida muestra que la constante que representa al atributo OBJ_ATTR_DECRYPT es 261. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

```
aws-cloudhsm> listAttributes

Following are the possible attribute values for getAttribute:

      OBJ_ATTR_CLASS                = 0
      OBJ_ATTR_TOKEN                = 1
      OBJ_ATTR_PRIVATE              = 2
      OBJ_ATTR_LABEL                = 3
      OBJ_ATTR_TRUSTED              = 134
      OBJ_ATTR_KEY_TYPE             = 256
      OBJ_ATTR_ID                   = 258
      OBJ_ATTR_SENSITIVE            = 259
      OBJ_ATTR_ENCRYPT              = 260
      OBJ_ATTR_DECRYPT              = 261
      OBJ_ATTR_WRAP                 = 262
```

```
      OBJ_ATTR_UNWRAP                 = 263
      OBJ_ATTR_SIGN                   = 264
      OBJ_ATTR_VERIFY                 = 266
      OBJ_ATTR_DERIVE                 = 268
      OBJ_ATTR_LOCAL                  = 355
      OBJ_ATTR_MODULUS                = 288
      OBJ_ATTR_MODULUS_BITS           = 289
      OBJ_ATTR_PUBLIC_EXPONENT        = 290
      OBJ_ATTR_VALUE_LEN              = 353
      OBJ_ATTR_EXTRACTABLE            = 354
      OBJ_ATTR_NEVER_EXTRACTABLE      = 356
      OBJ_ATTR_ALWAYS_SENSITIVE       = 357
      OBJ_ATTR_DESTROYABLE            = 370
      OBJ_ATTR_KCV                    = 371
      OBJ_ATTR_WRAP_WITH_TRUSTED      = 528
      OBJ_ATTR_WRAP_TEMPLATE          = 1073742353
      OBJ_ATTR_UNWRAP_TEMPLATE        = 1073742354
      OBJ_ATTR_ALL                    = 512
```

Para obtener el valor actual del atributo de descifrado de la clave 14, el siguiente comando utiliza getAttribute en cloudhsm_mgmt_util.

La salida muestra que el valor del atributo de descifrar es verdadera (1) en ambos HSM del clúster.

```
aws-cloudhsm> getAttribute 14 261

Attribute Value on server 0(10.0.0.1):
OBJ_ATTR_DECRYPT
0x00000001

Attribute Value on server 1(10.0.0.2):
OBJ_ATTR_DECRYPT
0x00000001
```

Este comando utiliza setAttribute para cambiar el valor del atributo de descifrado (atributo 261) de la clave 14 a 0. Esto desactiva la funcionalidad de descifrar en la clave.

La salida muestra que el comando se ha ejecutado correctamente en ambos HSM del clúster.

```
aws-cloudhsm> setAttribute 14 261 0
************************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
```

```
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
************************************************************

Do you want to continue(y/n)? y
setAttribute success on server 0(10.0.0.1)
setAttribute success on server 1(10.0.0.2)
```

El comando final repite el comando getAttribute. En este caso también obtiene el atributo de descifrar (atributo 261) de la clave 14.

Ahora la salida muestra que el valor del atributo de descifrar es falso (0) en ambos HSM del clúster.

```
aws-cloudhsm>getAttribute 14 261
Attribute Value on server 0(10.0.3.6):
OBJ_ATTR_DECRYPT
0x00000000

Attribute Value on server 1(10.0.1.7):
OBJ_ATTR_DECRYPT
0x00000000
```

Argumentos

```
setAttribute <key handle> <attribute id>
```

<key-handle>

Especifica el identificador de una clave de su propiedad. Puede especificar una única clave en cada comando. Para obtener el identificador de una clave, use findKey en key_mgmt_util. Para encontrar los usuarios de una clave, utilice. getKeyInfo

Obligatorio: sí

<attribute id>

Especifica la constante que representa el atributo que desea cambiar. Puede especificar un único atributo en cada comando. Para obtener los atributos y sus valores enteros, utilice listAttributes. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Valores válidos:

- 3.OBJ_ATTR_LABEL
- 1340BJ_ATTR_TRUSTED
- 260: OBJ_ATTR_ENCRYPT.
- 261: OBJ_ATTR_DECRYPT.
- 262: OBJ_ATTR_WRAP.
- 263: OBJ_ATTR_UNWRAP.
- 264: OBJ_ATTR_SIGN.
- 2660BJ_ATTR_VERIFY
- 2680BJ_ATTR_DERIVE
- 3700BJ_ATTR_DESTROYABLE
- 5280BJ_ATTR_WRAP_WITH_TRUSTED
- 1073742353: OBJ_ATTR_WRAP_TEMPLATE
- 1073742354: OBJ_ATTR_UNWRAP_TEMPLATE

Obligatorio: sí

Temas relacionados de

- setAttribute en key_mgmt_util
- getAttribute
- listAttributes
- Referencia de los atributos de claves

Quit

El comando quit en cloudhsm_mgmt_util sale de cloudhsm_mgmt_util. Todos los usuarios del tipo que sean pueden utilizar este comando.

Antes de ejecutar cualquier comando de cloudhsm_mgmt_util, debe iniciar cloudhsm_mgmt_util.

Tipo de usuario

Los usuarios siguientes pueden ejecutar este comando.

- Todos los usuarios. No es preciso haber iniciado sesión para ejecutar este comando.

Sintaxis

```
quit
```

Ejemplo

Este comando sale de cloudhsm_mgmt_util. Una vez que finaliza correctamente, se vuelve a la línea de comandos normal. Este comando no tiene parámetros de salida.

```
aws-cloudhsm> quit

disconnecting from servers, please wait...
```

Temas relacionados de

- Introducción a cloudhsm_mgmt_util

shareKey

El comando shareKey de cloudhsm_mgmt_util comparte y cancela el uso compartido de claves de su propiedad con otros usuarios de criptografía. Solo el propietario de la clave puede compartir y dejar de compartir una clave. También puede compartir una clave cuando la crea.

Los usuarios que comparten la clave pueden utilizar la clave en operaciones criptográficas, pero no pueden eliminar, exportar, compartir o dejar de compartir la clave, o cambiar sus atributos. Cuando la autenticación de cuórum está habilitada en una clave, el cuórum debe aprobar cualquier operación que comparta o deje de compartir la clave.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Usuarios de criptografía (CU)

Sintaxis

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

Tipo de usuario: usuario de criptografía (CU)

```
shareKey <key handle> <user id> <(share/unshare key?) 1/0>
```

Ejemplo

Los siguientes ejemplos muestran cómo utilizar shareKey para compartir y dejar de compartir claves que posee con otros usuarios de criptografía.

Example : compartir una clave

En este ejemplo, se utiliza shareKey para compartir una clave privada de ECC que el usuario actual posee con otro usuario de criptografía de los HSM. Las claves públicas están disponibles para todos los usuarios del HSM, por lo que no puede compartirlas o dejar de compartirlas.

El primer comando se utiliza getKeyInfopara obtener la información del usuario para la clave262177, una clave privada ECC en los HSM.

El resultado muestra que la clave 262177 es propiedad del usuario 3, pero no se comparte.

```
aws-cloudhsm>getKeyInfo 262177

Key Info on server 0(10.0.3.10):

        Token/Flash Key,

        Owned by user 3

Key Info on server 1(10.0.3.6):

        Token/Flash Key,

        Owned by user 3
```

Este comando utiliza shareKey para compartir la clave 262177 con el usuario 4, otro usuario de criptografía de los HSM. El último argumento utiliza un valor de 1 para indicar una operación compartida.

El resultado muestra que la operación ha tenido éxito en ambos HSM en el clúster.

```
aws-cloudhsm>shareKey 262177 4 1
**************************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
******************************************************************

Do you want to continue(y/n)?y
shareKey success on server 0(10.0.3.10)
shareKey success on server 1(10.0.3.6)
```

Para comprobar que la operación se ha realizado correctamente, el ejemplo repite el primer comando getKeyInfo.

El resultado muestra que la clave 262177 se comparte ahora con el usuario 4.

```
aws-cloudhsm>getKeyInfo 262177

Key Info on server 0(10.0.3.10):

        Token/Flash Key,

        Owned by user 3

        also, shared to following 1 user(s):

                4
Key Info on server 1(10.0.3.6):

        Token/Flash Key,

        Owned by user 3

        also, shared to following 1 user(s):
```

```
                    4
```

Example : dejar de compartir una clave

En este ejemplo, se deja de compartir una clave simétrica, es decir, se elimina un usuario de criptografía de la lista de usuarios compartidos para la clave.

Este comando utiliza shareKey para eliminar al usuario 4 de la lista de usuarios compartidos para la clave 6. El último argumento utiliza el valor `0` para indicar una operación para dejar de compartir.

El resultado muestra que el comando ha tenido éxito en ambos HSM. Como resultado, el usuario 4 ya no puede utilizar la clave 6 en operaciones criptográficas.

```
aws-cloudhsm>shareKey 6 4 0
**********************CAUTION*********************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. AWS does NOT synchronize these changes automatically with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
**************************************************************

Do you want to continue(y/n)?y
shareKey success on server 0(10.0.3.10)
shareKey success on server 1(10.0.3.6)
```

Argumentos

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
shareKey <key handle> <user id> <(share/unshare key?) 1/0>
```

<key-handle>

Especifica el identificador de una clave de su propiedad. Puede especificar una única clave en cada comando. Para obtener el identificador de una clave, use findKey en key_mgmt_util. Para comprobar que es propietario de una clave, utilice. getKeyInfo

Obligatorio: sí

<user id>

Especifica el ID del usuario de criptografía (CU) con quien está compartiendo o dejando de compartir la clave. Para encontrar el ID del usuario, utilice listUsers.

Obligatorio: sí

<share 1 o unshare 0>

Para compartir la clave con el usuario especificado, escriba 1. Para dejar de compartir la clave, es decir, para eliminar al usuario especificado de la lista de los usuarios compartidos para la clave, escriba 0.

Obligatorio: sí

Temas relacionados de

- getKeyInfo

syncKey

Puede utilizar el comando syncKey en cloudhsm_mgmt_util para sincronizar manualmente las claves entre las instancias de HSM de un clúster o entre clústeres clonados. En general, no es preciso utilizar este comando, puesto que las instancias de HSM dentro de un clúster sincronizan las claves automáticamente. Sin embargo, la sincronización de claves entre clústeres clonados debe realizarse manualmente. Los clústeres clonados suelen crearse en distintas AWS regiones para simplificar el escalamiento global y los procesos de recuperación ante desastres.

No puede utilizar syncKey para sincronizar claves entre clústeres arbitrarios: uno de los clústeres debe haberse creado a partir de una copia de seguridad del otro. Además, ambos clústeres deben tener las credenciales de CO y CU coherentes para que la operación se lleve a cabo correctamente. Para obtener más información, consulte Usuarios de HSM.

Para usarlosyncKey, primero debe crear un archivo de AWS CloudHSM configuración que especifique un HSM del clúster de origen y otro del clúster de destino. Esto permitirá a cloudhsm_mgmt_util conectarse a ambas instancias de HSM. Utilice este archivo de configuración para iniciar cloudhsm_mgmt_util. A continuación, inicie sesión con las credenciales de un CO o un CU que tenga las claves que desea sincronizar.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Responsables de criptografía (CO)
- Usuarios de criptografía (CU)

> **Note**
>
> Los CO pueden utilizar syncKey con todas las claves; en cambio, los CU solo pueden utilizar este comando con las claves que les pertenecen. Para obtener más información, consulte the section called “Más información sobre los usuarios de HSM”.

Requisitos previos

Antes de comenzar, debe conocer el `key handle` de la clave del HSM de origen que hay que sincronizar con el HSM de destino. Si desea buscar el `key handle`, utilice el comando listUsers para enumerar todos los identificadores de los usuarios designados. A continuación, utilice el findAllKeyscomando para buscar todas las claves que pertenecen a un usuario concreto.

También debe conocer los `server IDs` asignados a los HSM de origen y de destino, que se muestran en la salida de rastreo devuelta por cloudhsm_mgmt_util al iniciarse. Estos se asignan en el mismo orden con que aparecen los HSM en el archivo de configuración.

Siga las instrucciones de Uso de CMU en clústeres clonados e inicialice cloudhsm_mgmt_util con el archivo de configuración nuevo. A continuación, entre en el modo de servidor en el HSM; para ello, ejecute el comando server.

Sintaxis

> **Note**
>
> Para ejecutar syncKey, en primer lugar, entre en el modo de servidor en el HSM que contiene la clave que se va a sincronizar.

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

Tipo de usuario: usuario de criptografía (CU)

```
syncKey <key handle> <destination hsm>
```

Ejemplo

Ejecute el comando server para iniciar sesión en el HSM de origen y entrar en el modo de servidor. En este ejemplo, se supone que `server 0` es el HSM de origen.

```
aws-cloudhsm> server 0
```

A continuación, ejecute el comando syncKey. En este ejemplo, se supone que la clave `261251` se va a sincronizar con `server 1`.

```
aws-cloudhsm> syncKey 261251 1
syncKey success
```

Argumentos

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
syncKey <key handle> <destination hsm>
```

<key handle>

Especifica el identificador de la clave que se va a sincronizar. Puede especificar una única clave en cada comando. Para obtener el identificador de una clave, utilícelo findAllKeysmientras está conectado a un servidor HSM.

Obligatorio: sí

<destination hsm>

Especifica el número del servidor con el que se va a sincronizar una clave.

Obligatorio: sí

Temas relacionados de

- listUsers

- findAllKeys
- describe-clusters en CLI
- servidor

syncUser

Puedes usar el syncUser comando de cloudhsm_mgmt_util para sincronizar manualmente los usuarios criptográficos (CU) o los criptooficiales (CO) entre las instancias de HSM de un clúster o entre clústeres clonados. AWS CloudHSM no sincroniza automáticamente a los usuarios. Por lo general, los usuarios se administran en modo global con objeto de que todos los HSM de un clúster se actualicen conjuntamente. Es posible que tenga que utilizar syncUser si un HSM se desincroniza accidentalmente (por ejemplo, debido a los cambios de contraseña) o si desea rotar las credenciales de usuario en los clústeres clonados. Los clústeres clonados suelen crearse en distintas AWS regiones para simplificar el escalamiento global y los procesos de recuperación ante desastres.

Para poder ejecutar cualquier comando de CMU, debe iniciar la CMU e iniciar sesión en el HSM. Asegúrese de que inicia sesión con un tipo de usuario que pueda ejecutar los comandos que planea utilizar.

Si agrega o elimina uno o varios HSM, actualice los archivos de configuración de la CMU. De lo contrario, es posible que los cambios que realice no se hagan efectivos para todos los HSM del clúster.

Tipo de usuario

Los tipos de usuarios siguientes pueden ejecutar este comando.

- Responsables de criptografía (CO)

Requisitos previos

Antes de comenzar, debe conocer el `user ID` del usuario del HSM de origen que hay que sincronizar con el HSM de destino. Para encontrar el `user ID`, utilice el comando listUsers para obtener una lista de todos los usuarios de los HSM del clúster.

También debe conocer los `server ID` asignados a los HSM de origen y de destino, que se muestran en la salida de rastreo devuelta por cloudhsm_mgmt_util al iniciarse. Estos se asignan en el mismo orden con que aparecen los HSM en el archivo de configuración.

Si va a sincronizar HSM en clústeres clonados, siga las instrucciones de Uso de CMU en clústeres clonados e inicialice cloudhsm_mgmt_util con el nuevo archivo de configuración.

Cuando esté listo para ejecutar syncUser, emita el comando server para entrar en el modo de servidor en el HSM de origen.

Sintaxis

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
syncUser <user ID> <server ID>
```

Ejemplo

Ejecute el comando server para iniciar sesión en el HSM de origen y entrar en el modo de servidor. En este ejemplo, se supone que `server 0` es el HSM de origen.

```
aws-cloudhsm> server 0
```

Ejecute el comando syncUser. En este ejemplo, suponemos que se va a sincronizar el usuario 6 en el HSM de destino `server 1`.

```
server 0> syncUser 6 1
ExtractMaskedObject: 0x0 !
InsertMaskedObject: 0x0 !
syncUser success
```

Argumentos

Dado que estos comandos no tienen parámetros designados, debe introducir los argumentos en el orden especificado en los diagramas de sintaxis.

```
syncUser <user ID> <server ID>
```

<user ID>

Especifica el ID del usuario que se va a sincronizar. Solo se puede especificar un usuario en cada comando. Para obtener el ID de un usuario, utilice listUsers.

Obligatorio: sí

<server ID>

Especifica el número de servidor del HSM con el que se va a sincronizar un usuario.

Obligatorio: sí

Temas relacionados de

- listUsers
- describe-clusters en CLI
- servidor

Utilidad de administración de claves (KMU)

Utilidad de administración de claves (KMU) es una herramienta de línea de comandos que ayuda a los usuarios de criptografía (CU) a administrar las claves de los módulos de seguridad de hardware (HSM). KMU contiene varios comandos que generan, eliminan, importan y exportan claves, obtienen y establecen atributos, encuentran claves y realizan operaciones criptográficas.

KMU y CMU forman parte del conjunto de SDK 3 de cliente.

Para un inicio rápido, consulte Introducción a key_mgmt_util. Para obtener información detallada acerca de los comandos, consulte referencia del comando cloudhsm_mgmt_util. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Para utilizar key_mgmt_util si usa Linux, conéctese a la instancia de cliente y, a continuación, consulte Instalar y configurar el AWS CloudHSM cliente (Linux). Si usa Windows, consulte Instalación y configuración del AWS CloudHSM cliente (Windows).

Temas

- Introducción a key_mgmt_util
- Instalar y configurar el AWS CloudHSM cliente (Linux)
- Instalación y configuración del AWS CloudHSM cliente (Windows)
- referencia del comando cloudhsm_mgmt_util

Introducción a key_mgmt_util

AWS CloudHSM incluye dos herramientas de línea de comandos con el software AWS CloudHSM cliente. La herramienta cloudhsm_mgmt_util dispone de comandos para la administración de usuarios de HSM. La herramienta key_mgmt_util tiene comandos para administrar las claves. Para comenzar a utilizar la herramienta de línea de comandos key_mgmt_util, consulte los siguientes temas.

Temas

- Configurar key_mgmt_util
- Uso básico de key_mgmt_util

Si detecta un mensaje de error o resultados inesperados para un comando, consulte los temas de Solución de problemas AWS CloudHSM para obtener ayuda. Para obtener más detalles acerca de los comandos de key_mgmt_util, consulte referencia del comando cloudhsm_mgmt_util.

Configurar key_mgmt_util

Siga los pasos de configuración que se describen a continuación antes de utilizar key_mgmt_util.

Inicie el AWS CloudHSM cliente

Antes de usar key_mgmt_util, debe iniciar el cliente. AWS CloudHSM El cliente es un daemon que establece una comunicación end-to-end cifrada con los HSM del clúster. La herramienta key_mgmt_util utiliza la conexión del cliente para comunicarse con los HSM del clúster. Sin ella, key_mgmt_util no funciona.

Para iniciar el cliente AWS CloudHSM

Utilice el siguiente comando para iniciar el AWS CloudHSM cliente.

Amazon Linux

```
$ sudo start cloudhsm-client
```

Amazon Linux 2

```
$ sudo service cloudhsm-client start
```

CentOS 7

```
$ sudo service cloudhsm-client start
```

CentOS 8

```
$ sudo service cloudhsm-client start
```

RHEL 7

```
$ sudo service cloudhsm-client start
```

RHEL 8

```
$ sudo service cloudhsm-client start
```

Ubuntu 16.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client start
```

Windows

- Para la versión 1.1.2 y posteriores del cliente de Windows:

```
C:\Program Files\Amazon\CloudHSM>net.exe start AWSCloudHSMClient
```

- Para la versión 1.1.1 y anteriores de clientes de Windows:

```
C:\Program Files\Amazon\CloudHSM>start "cloudhsm_client" cloudhsm_client.exe C:
\ProgramData\Amazon\CloudHSM\data\cloudhsm_client.cfg
```

Iniciar key_mgmt_util

Tras iniciar el AWS CloudHSM cliente, utilice el siguiente comando para iniciar key_mgmt_util.

Amazon Linux

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

Amazon Linux 2

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

CentOS 7

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

CentOS 8

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

RHEL 7

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

RHEL 8

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

Ubuntu 16.04 LTS

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

Ubuntu 18.04 LTS

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

Windows

```
c:\Program Files\Amazon\CloudHSM> .\key_mgmt_util.exe
```

El símbolo cambia a `Command:` cuando se está ejecutando key_mgmt_util.

Si el comando produce un error, como, por ejemplo, devuelve un mensaje `Daemon socket connection error`, intente actualizar su archivo de configuración.

Uso básico de key_mgmt_util

Consulte los siguientes temas para conocer el uso básico de la herramienta key_mgmt_util.

Temas

- Iniciar sesión en los HSM
- Cierre de la sesión de los HSM
- Detener key_mgmt_util

Iniciar sesión en los HSM

Utilice el comando loginHSM para iniciar sesión en los HSM. El siguiente comando inicia sesión como un usuario de criptografía (CU) denominado `example_user`. El resultado indica un inicio de sesión correcto para los tres HSM del clúster.

```
Command:  loginHSM -u CU -s example_user -p <PASSWORD>
Cfm3LoginHSM returned: 0x00 : HSM Return: SUCCESS

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

A continuación, se muestra la sintaxis del comando loginHSM.

```
Command:  loginHSM -u <USER TYPE> -s <USERNAME> -p <PASSWORD>
```

Cierre de la sesión de los HSM

Utilice el comando logoutHSM para cerrar sesión en los HSM.

```
Command:  logoutHSM
Cfm3LogoutHSM returned: 0x00 : HSM Return: SUCCESS

Cluster Error Status
```

```
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Detener key_mgmt_util

Utilice el comando exit para detener key_mgmt_util.

```
Command:  exit
```

Instalar y configurar el AWS CloudHSM cliente (Linux)

Para interactuar con el HSM de su AWS CloudHSM clúster, necesita el software de AWS CloudHSM cliente para Linux. Debe instalarlo en la instancia de cliente de Linux EC2 que creó anteriormente. También puede instalar un cliente si utiliza Windows. Para obtener más información, consulte Instalación y configuración del AWS CloudHSM cliente (Windows).

Tareas

- Instale el AWS CloudHSM cliente y las herramientas de línea de comandos
- Edición de la configuración del cliente

Instale el AWS CloudHSM cliente y las herramientas de línea de comandos

Conéctese a su instancia de cliente y ejecute los siguientes comandos para descargar e instalar el AWS CloudHSM cliente y las herramientas de línea de comandos.

Amazon Linux

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-latest.el6.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el6.x86_64.rpm
```

Amazon Linux 2

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-latest.el7.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el7.x86_64.rpm
```

CentOS 7

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-latest.el7.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el7.x86_64.rpm
```

CentOS 8

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-latest.el8.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el8.x86_64.rpm
```

RHEL 7

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-latest.el7.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el7.x86_64.rpm
```

RHEL 8

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-latest.el8.x86_64.rpm
```

```
sudo yum install ./cloudhsm-client-latest.el8.x86_64.rpm
```

Ubuntu 16.04 LTS

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/cloudhsm-client_latest_amd64.deb
```

```
sudo apt install ./cloudhsm-client_latest_amd64.deb
```

Ubuntu 18.04 LTS

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Bionic/cloudhsm-client_latest_u18.04_amd64.deb
```

```
sudo apt install ./cloudhsm-client_latest_u18.04_amd64.deb
```

Edición de la configuración del cliente

Antes de poder usar el AWS CloudHSM cliente para conectarse al clúster, debe editar la configuración del cliente.

Para editar la configuración del cliente

1. Copie el certificado de emisión —el que utilizó para firmar el certificado del clúster— en la siguiente ubicación de la instancia de cliente: `/opt/cloudhsm/etc/customerCA.crt`. Necesita permisos de usuario raíz en la instancia de cliente para copiar el certificado en esta ubicación.
2. Utilice el siguiente comando configure para actualizar los archivos de configuración del AWS CloudHSM cliente y las herramientas de línea de comandos, especificando la dirección IP del HSM del clúster. Para obtener la dirección IP del HSM, consulte el clúster en la AWS CloudHSM consola o ejecute el comando describe-clusters CLI. En la salida del comando, la dirección IP del HSM es el valor del campo `EniIp`. Si tiene más de un HSM, elija la dirección IP de cualquiera de ellos; no importa el que elija.

   ```
   sudo /opt/cloudhsm/bin/configure -a <IP address>
   ```

```
Updating server config in /opt/cloudhsm/etc/cloudhsm_client.cfg
Updating server config in /opt/cloudhsm/etc/cloudhsm_mgmt_util.cfg
```

3. Vaya a Activación del clúster.

Instalación y configuración del AWS CloudHSM cliente (Windows)

Para trabajar con un HSM de su AWS CloudHSM clúster en Windows, necesita el software de AWS CloudHSM cliente para Windows. Debe instalarlo en la instancia de Windows Server que creó anteriormente.

Cómo instalar (o actualizar) el cliente más reciente en Windows y las herramientas de línea de comandos

1. Conéctese a su instancia de Windows Server.
2. Descargue la versión más reciente (AWSCloudHSMClient-latest.msi) de la página de descargas.
3. Vaya a la ubicación de descarga y ejecute el instalador (AWSCloudHSMClient-latest.msi) con privilegios administrativos.
4. Siga las instrucciones del instalador y, a continuación, seleccione Cerrar cuando el instalador haya finalizado.
5. Copie el certificado de emisión autofirmado —el que utilizó para firmar el certificado del clúster — en la carpeta `C:\ProgramData\Amazon\CloudHSM`.
6. Ejecute el siguiente comando para actualizar los archivos de configuración. Asegúrese de detener e iniciar el cliente durante la reconfiguración si lo está actualizando:

```
C:\Program Files\Amazon\CloudHSM\bin\ .\configure.exe -a <HSM IP address>
```

7. Vaya a Activación del clúster.

Notas:

- Si está actualizando el cliente, los archivos de configuración existentes de las instalaciones anteriores no se sobrescribirán.
- El instalador del AWS CloudHSM cliente para Windows registra automáticamente la API de criptografía: Next Generation (CNG) y Key Storage Provider (KSP). Para desinstalar el software de cliente, vuelva a ejecutar el instalador y siga las instrucciones de desinstalación.

- Si utiliza Linux, puede instalar el software de cliente de Linux. Para obtener más información, consulte Instalar y configurar el AWS CloudHSM cliente (Linux).

referencia del comando cloudhsm_mgmt_util

La herramienta de línea de comandos key_mgmt_util le ayuda a administrar claves en los HSM de un clúster, incluidas la creación, la eliminación y la búsqueda de claves y sus atributos. Contiene varios comandos, cada uno de los cuales se describe en detalle en este tema.

Para un inicio rápido, consulte Introducción a key_mgmt_util. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves. Para obtener información sobre la herramienta de línea de comandos cloudhsm_mgmt_util que contiene los comandos necesarios para administrar el HSM y los usuarios del clúster, consulte Utilidad de administración de CloudHSM (CMU).

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Para ver una lista de todos los comandos key_mgmt_util, escriba:

```
Command: help
```

Para obtener ayuda sobre un comando de key_mgmt_util específico, escriba:

```
Command: <command-name> -h
```

Para finalizar la sesión de key_mgmt_util, escriba:

```
Command: exit
```

En los temas siguientes, se describen los comandos de key_mgmt_util.

> **Note**
>
> Algunos comandos de key_mgmt_util y cloudhsm_mgmt_util tienen el mismo nombre. Sin embargo, los comandos suelen tener una sintaxis diferente, un resultado diferente y una funcionalidad ligeramente diferente.

Comando	Descripción
aesWrapUnwrap	Cifra y descifra el contenido de una clave de un archivo.
deleteKey	Elimina una clave de los HSM.
Error2String	Obtiene el error que corresponde al código de error hexadecimal key_mgmt_util.
exit	Sale de key_mgmt_util.
exportPrivateKey	Exporta una copia de una clave privada de un HSM a un archivo del disco.
exportPubKey	Exporta una copia de una clave pública de un HSM a un archivo.
exSymKey	Exporta una copia de texto no cifrado de una clave simétrica desde los HSM a un archivo.
extractMaskedObject	Extrae una clave de un HSM como un archivo de objeto enmascarado.
findKey	Busca claves en función de un valor de atributo de clave.
findSingleKey	Comprueba que todos los HSM del clúster tengan una clave.
GenDSA KeyPair	Genera un par de claves de algoritmo de firma digital (DSA) en los HSM.
GeneCC KeyPair	Genera un par de claves de criptografía de curva elíptica (ECC) en sus HSM.
Género A KeyPair	Genera un par de claves asimétricas RSA en sus HSM.
genSymKey	Genera una clave simétrica en sus HSM.

Comando	Descripción
getAttribute	Obtiene los valores de los atributos de una clave de AWS CloudHSM y los escribe en un archivo.
getCaviumPrivClave	Crea una versión en formato PEM falso de una clave privada y la exporta a un archivo.
getCert	Recupera los certificados de las particiones de un HSM y los guarda en un archivo.
getKeyInfo	Obtiene los ID de los usuarios de HSM que pueden utilizar la clave. Si la clave se controla mediante cuórum, obtiene el número de usuarios del cuórum.
help	Muestra información de ayuda para los comandos disponibles en key_mgmt_util.
importPrivateKey	Importa una clave privada en un HSM.
importPubKey	Importa una clave pública en un HSM.
imSymKey	Importa una copia de texto no cifrado de una clave simétrica desde un archivo a los HSM.
insertMaskedObject	Inserta un objeto enmascarado desde un archivo en disco a un HSM perteneciente al clúster relacionado con el clúster de origen del objeto. Un clúster relacionado es cualquier clúster generado a partir de una copia de seguridad del clúster de origen.
???	Determina si un archivo determinado contiene una clave privada verdadera o una clave PEM falsa.

Comando	Descripción
listAttributes	Muestra los atributos de una AWS CloudHSM clave y las constantes que los representan.
listUsers	Obtiene los usuarios de los HSM, su ID y tipo de usuario y otros atributos.
loginHSM y logoutHSM	Inicia y cierra sesión en los HSM de un clúster.
setAttribute	Convierte una clave de sesión en una clave persistente.
sign	Genera una firma para un archivo utilizando la clave privada elegida.
unWrapKey	Importa una clave encapsulada (cifrada) desde un archivo a los HSM.
verify	Verifica si se utilizó una clave determinada para firmar un archivo concreto.
wrapKey	Exporta una copia cifrada de una clave desde el HSM a un archivo.

aesWrapUnwrap

El comando aesWrapUnwrap cifra o descifra el contenido de un archivo en el disco. Este comando está diseñado para encapsular y desencapsular claves de cifrado, pero se puede utilizar en cualquier archivo que contenga menos de 4 KB (4096 bytes) de datos.

aesWrapUnwrap utiliza el encapsulado de claves AES. Emplea una clave AES en el HSM como clave de encapsulamiento o desencapsulamiento. A continuación escribe el resultado en otro archivo en el disco.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
aesWrapUnwrap -h

aesWrapUnwrap -m <wrap-unwrap mode>
              -f <file-to-wrap-unwrap>
              -w <wrapping-key-handle>
              [-i <wrapping-IV>]
              [-out <output-file>]
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza aesWrapUnwrap para cifrar y descifrar una clave de cifrado en un archivo.

Example : encapsulamiento de una clave de cifrado

Este comando utiliza aesWrapUnwrap para encapsular una clave simétrica triple DES que se ha exportado desde el HSM sin cifrar hasta el archivo `3DES.key`. Puede utilizar un comando similar para encapsular toda clave que esté guardada en un archivo.

El comando utiliza el parámetro `-m` con un valor de 1 para indicar el modo de encapsulamiento. Utiliza el parámetro `-w` para especificar una clave AES en el HSM (indicador de clave 6) como clave de encapsulamiento. Escribe la clave encapsulada obtenida en el archivo `3DES.key.wrapped`.

La salida muestra que el comando se ha ejecutado correctamente y que la operación ha utilizado el IV predeterminado, que es el preferido.

```
 Command:  aesWrapUnwrap -f 3DES.key -w 6 -m 1 -out 3DES.key.wrapped

        Warning: IV (-i) is missing.
                 0xA6A6A6A6A6A6A6A6 is considered as default IV
result data:
49 49 E2 D0 11 C1 97 22
17 43 BD E3 4E F4 12 75
8D C1 34 CF 26 10 3A 8D
6D 0A 7B D5 D3 E8 4D C2
79 09 08 61 94 68 51 B7

result written to file 3DES.key.wrapped
```

```
        Cfm3WrapHostKey returned: 0x00 : HSM Return: SUCCESS
```

Example : desencapsulamiento de una clave de cifrado

En este ejemplo, se muestra cómo se utiliza aesWrapUnwrap para desencapsular (descifrar) una clave encapsulada (cifrada) en un archivo. Puede que le interese realizar una operación de este tipo antes de importar una clave al HSM. Por ejemplo, si intenta utilizar el imSymKeycomando para importar una clave cifrada, devuelve un error porque la clave cifrada no tiene el formato necesario para una clave de texto sin formato de ese tipo.

El comando desencapsula la clave del archivo `3DES.key.wrapped` y escribe el texto sin cifrar en el archivo `3DES.key.unwrapped`. El comando utiliza el parámetro `-m` con un valor de `0` para indicar el modo de desencapsulamiento. Utiliza el parámetro `-w` para especificar una clave AES en el HSM (indicador de clave 6) como clave de encapsulamiento. Escribe la clave encapsulada obtenida en el archivo `3DES.key.unwrapped`.

```
 Command:  aesWrapUnwrap -m 0 -f 3DES.key.wrapped -w 6 -out 3DES.key.unwrapped

        Warning: IV (-i) is missing.
                 0xA6A6A6A6A6A6A6A6 is considered as default IV
result data:
14 90 D7 AD D6 E4 F5 FA
A1 95 6F 24 89 79 F3 EE
37 21 E6 54 1F 3B 8D 62

result written to file 3DES.key.unwrapped

        Cfm3UnWrapHostKey returned: 0x00 : HSM Return: SUCCESS
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-m

Especifica el modo. Para encapsular (cifrar) el contenido del archivo, escriba 1; para desencapsular (descifrar) el contenido del archivo, escriba `0`.

Obligatorio: sí

-f

Especifica el archivo que se va a encapsular. Especifique un archivo que contenga menos de 4 KB (4096 bytes) de datos. Esta operación está diseñada para encapsular y desencapsular claves de cifrado.

Obligatorio: sí

-w

Especifica la clave de encapsulamiento. Introduzca el identificador de clave de una clave AES en el HSM. Este parámetro es obligatorio. Para buscar identificadores de clave, use el comando findKey.

Para crear una clave de empaquetado, utilice genSymKeypara generar una clave AES (tipo 31).

Obligatorio: sí

-i

Especifica un valor inicial alternativo (IV) para el algoritmo. Utilice el valor predeterminado a menos que tenga una condición especial que requiera una alternativa.

Predeterminado: `0xA6A6A6A6A6A6A6A6`. El valor predeterminado se define en la especificación de algoritmo de encapsulamiento de claves AES.

Obligatorio: no

-out

Especifica un nombre alternativo para el archivo de salida que contiene la clave encapsulada o desencapsulada. El valor predeterminado es `wrapped_key` (para operaciones de encapsulamiento) y `unwrapped_key` (para operaciones de desencapsulamiento) en el directorio local.

Si el archivo existe, el comando aesWrapUnwrap lo sobrescribe sin ningún tipo de advertencia. Si se produce un error en el comando, aesWrapUnwrap crea un archivo de salida sin contenido.

Valor predeterminado: para el encapsulamiento: `wrapped_key`. Para el desencapsulamiento: `unwrapped_key`.

Obligatorio: no

Temas relacionados de

- exSymKey
- imSymKey
- unWrapKey
- wrapKey

deleteKey

El comando deleteKey de key_mgmt_util permite eliminar una clave del HSM. Solo puede eliminar las claves de una en una. La eliminación de una clave de un par de claves no influye en la otra clave del par.

Solo el propietario de la clave puede eliminar una clave. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas, pero no eliminarla.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
deleteKey -h

deleteKey -k
```

Ejemplos

En los ejemplos siguientes, se muestra cómo se utiliza deleteKey para eliminar claves de los HSM.

Example : eliminación de una clave

Este comando elimina la clave que tiene el identificador de clave 6. Cuando el comando se ejecuta correctamente, deleteKey devuelve mensajes de éxito desde cada HSM del clúster.

```
Command: deleteKey -k 6
```

```
        Cfm3DeleteKey returned: 0x00 : HSM Return: SUCCESS

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Example : eliminación de una clave (error)

Cuando se produce un error en el comando porque ninguna clave tiene el identificador de clave especificado, deleteKey devuelve un mensaje de error de identificador de objeto no válido.

```
Command: deleteKey -k 252126

        Cfm3FindKey returned: 0xa8 : HSM Error: Invalid object handle is passed to this
 operation

        Cluster Error Status
        Node id 1 and err state 0x000000a8 : HSM Error: Invalid object handle is passed
 to this operation
        Node id 2 and err state 0x000000a8 : HSM Error: Invalid object handle is passed
 to this operation
```

Cuando el comando genera un error porque el usuario actual no es el propietario de la clave, el comando devuelve un error de acceso denegado.

```
Command:  deleteKey -k 262152

Cfm3DeleteKey returned: 0xc6 : HSM Error: Key Access is denied.
```

Parámetros

-h

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

-k

Especifica el identificador de la clave que se va a eliminar. Para encontrar los identificadores de clave de las claves del HSM, utilice findKey.

Obligatorio: sí

Temas relacionados de

- findKey

Error2String

El comando de ayuda Error2String de key_mgmt_util devuelve el error que corresponde a un código de error hexadecimal de key_mgmt_util. Puede utilizar este comando cuando esté solucionando problemas de los comandos y los scripts.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
Error2String -h

Error2String -r <response-code>
```

Ejemplos

En estos ejemplos, se muestra cómo utilizar Error2String para obtener la cadena de error de un código de error de key_mgmt_util.

Example : obtención de una descripción del error

Este comando obtiene la descripción del error del código de error `0xdb`. En la descripción, se explica que un intento de iniciar sesión en key_mgmt_util generó un error porque el usuario no era del tipo adecuado. Solo los usuarios de criptografía (CU) pueden iniciar sesión en key_mgmt_util.

```
        Command:  Error2String -r 0xdb

        Error Code db maps to HSM Error: Invalid User Type.
```

Example : búsqueda del código de error

En este ejemplo, se muestra dónde se puede encontrar el código de error de key_mgmt_util. El código de error, `0xc6`, se muestra después de la cadena: `Cfm3`*`command-name`* `returned:` .

En este ejemplo, getKeyInfoindica que el usuario actual (usuario 4) puede utilizar la clave en operaciones criptográficas. Sin embargo, cuando el usuario intenta utilizar deleteKey para eliminar la clave, el comando devuelve el código de error `0xc6`.

```
        Command:  deleteKey -k 262162

        Cfm3DeleteKey returned: 0xc6 : HSM Error: Key Access is denied

        Cluster Error Status

        Command:  getKeyInfo -k 262162

        Cfm3GetKey returned: 0x00 : HSM Return: SUCCESS

       Owned by user 3

       also, shared to following 1 user(s):

               4
```

Si se le notifica el error `0xc6`, puede utilizar un comando Error2String como este para buscar el error. En este caso el comando `deleteKey` ha generado un error de acceso denegado porque la clave se comparte con el usuario actual, pero es propiedad de otro usuario. Solo los propietarios de la clave tienen permiso para eliminarla.

```
        Command:  Error2String -r 0xa8

        Error Code c6 maps to HSM Error: Key Access is denied
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-r

Especifica un código de error hexadecimal. El indicador `0x` hexadecimal es obligatorio.

Obligatorio: sí

exit

El comando exit de key_mgmt_util permite salir de key_mgmt_util. Después de salir correctamente, volverá a la línea de comandos estándar.

Para poder ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util.

Sintaxis

```
exit
```

Parámetros

Este comando no tiene parámetros.

Temas relacionados de

- Iniciar key_mgmt_util

exportPrivateKey

El comando exportPrivateKey de key_mgmt_util exporta una clave privada asimétrica de un HSM a un archivo. El HSM no permite la exportación directa de claves en texto sin cifrar. El comando encapsula la clave privada con una clave de encapsulamiento AES que usted especifique, descifra los bytes encapsulados y copia la clave privada de texto sin cifrar en un archivo.

El comando exportPrivateKey no elimina la clave del HSM, no cambia sus atributos de clave ni le impide a usted utilizar la clave en operaciones criptográficas posteriores. Puede exportar la misma clave varias veces.

Solo puede exportar claves privadas que tengan un atributo `OBJ_ATTR_EXTRACTABLE` con valor 1. Debe especificar una clave de encapsulamiento AES que tenga los atributos `OBJ_ATTR_WRAP` y `OBJ_ATTR_DECRYPT` con valor 1. Para buscar los atributos de una clave, utilice el comando getAttribute.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
exportPrivateKey -h

exportPrivateKey -k <private-key-handle
                 -w <wrapping-key-handle>
                 -out <key-file>
                 [-m <wrapping-mechanism>]
                 [-wk <wrapping-key-file>]
```

Ejemplos

Este ejemplo muestra cómo utilizar exportPrivateKey para exportar una clave privada de un HSM.

Example : exportación de una clave privada

Este comando exporta la clave privada con el identificador 15 utilizando una clave de encapsulación con el identificador 16 a un archivo PEM denominado `exportKey.pem`. Cuando el comando se ejecuta correctamente, exportPrivateKey devuelve un mensaje de confirmación.

```
Command: exportPrivateKey -k 15 -w 16 -out exportKey.pem

Cfm3WrapKey returned: 0x00 : HSM Return: SUCCESS

        Cfm3UnWrapHostKey returned: 0x00 : HSM Return: SUCCESS

PEM formatted private key is written to exportKey.pem
```

Parámetros

Este comando admite los siguientes parámetros.

`-h`

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

`-k`

Especifica el identificador de clave de la clave privada que se va a exportar.

Obligatorio: sí

-w

Especifica el identificador de la clave de encapsulamiento. Este parámetro es obligatorio. Para buscar identificadores de clave, utilice el comando findKey.

Para determinar si una clave se puede utilizar como clave de encapsulación, utilice getAttribute para obtener el valor del atributo OBJ_ATTR_WRAP (262). Para crear una clave de encapsulación, utilice genSymKey para crear una clave AES (de tipo 31).

Si utiliza el parámetro `-wk` para especificar una clave de desencapsulación externa, la clave de encapsulación `-w` se utiliza para encapsular, pero no para desencapsular, la clave durante la exportación.

Obligatorio: sí

-out

Especifica el nombre del archivo en el que se escribirá la clave privada exportada.

Obligatorio: sí

-m

Especifica el mecanismo de encapsulación que se aplicará a la clave privada que se va a exportar. El único valor válido es `4`, que representa el mecanismo `NIST_AES_WRAP mechanism.`

Valor predeterminado: 4 (`NIST_AES_WRAP`)

Obligatorio: no

-wk

Especifica la clave que se utilizará para desencapsular la clave que se está exportando. Escriba la ruta y el nombre de un archivo que contenga una clave AES sin cifrar.

Si se incluye este parámetro, exportPrivateKey utiliza la clave del archivo especificado en el parámetro `-w` para encapsular la clave que se va a exportar y utiliza la clave especificada con el parámetro `-wk` para desencapsularla.

Valor predeterminado: utilizar la clave de encapsulación especificada en el parámetro `-w` para encapsular y desencapsular.

Obligatorio: no

Temas relacionados de

- importPrivateKey
- wrapKey
- unWrapKey
- genSymKey

exportPubKey

El comando exportPubKey en key_mgmt_util exporta una clave pública de un HSM a un archivo. Puede utilizarlo para exportar las claves públicas que genere en un HSM. También puede utilizar este comando para exportar las claves públicas que se importaron en un HSM, como las que se importaron con el comando importPubKey.

El comando exportPubKey copia el material de claves en el archivo que se especifique. Sin embargo, no elimina la clave del HSM, no cambia sus atributos de clave ni le impide a usted utilizar la clave en operaciones criptográficas posteriores. Puede exportar la misma clave varias veces.

Solo puede exportar las claves públicas cuyo valor de `OBJ_ATTR_EXTRACTABLE` es 1. Para buscar los atributos de una clave, utilice el comando getAttribute.

Antes de ejecutar cualquier comando de `key_mgmt_util`, debe iniciar key_mgmt_util e iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
exportPubKey -h

exportPubKey -k <public-key-handle
             -out <key-file>
```

Ejemplos

Este ejemplo muestra cómo utilizar exportPubKey para exportar una clave pública de un HSM.

Example : exportar una clave pública.

Este comando exporta una clave pública con el identificador `10` a un archivo denominado `public.pem`. Cuando el comando se ejecuta correctamente, exportPubKey devuelve un mensaje de confirmación.

```
Command: exportPubKey -k 10 -out public.pem

PEM formatted public key is written to public.pem

Cfm3ExportPubKey returned: 0x00 : HSM Return: SUCCESS
```

Parámetros

Este comando admite los siguientes parámetros.

`-h`

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

`-k`

Especifica el identificador de clave de la clave pública que se va a exportar.

Obligatorio: sí

`-out`

Especifica el nombre del archivo en el que se escribirá la clave pública exportada.

Obligatorio: sí

Temas relacionados de

- importPubKey
- Generar claves

exSymKey

El comando exSymKey en la herramienta key_mgmt_util exporta una copia sin cifrar de una clave simétrica desde el HSM y la guarda en un archivo del disco. Para exportar una copia cifrada

(encapsulada) de una clave, utilice wrapKey. Para importar una clave de texto sin formato, como las que se exportan, utilice. `exSymKey` imSymKey

Durante el proceso de exportación, exSymKey utiliza la clave AES especificada (la clave de encapsulado) para encapsular (cifrar) y después desencapsular (descifrar) la clave que se va a exportar. Sin embargo, el resultado de la operación de exportación es una clave sin cifrar (desencapsulada) en el disco.

Solo el propietario de una clave, es decir, el usuario CU que creó la clave, puede exportarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas, pero no pueden exportarla.

La operación exSymKey copia el material de la clave en el archivo especificado, pero no elimina la clave del HSM, ni cambia sus atributos de clave, ni le impide utilizar la clave en operaciones criptográficas. Puede exportar la misma clave varias veces.

exSymKey exporta únicamente claves simétricas. Para exportar claves públicas, utilice exportPubKey. Para exportar claves privadas, utilice exportPrivateKey.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
exSymKey -h

exSymKey -k <key-to-export>
         -w <wrapping-key>
         -out <key-file>
         [-m 4]
         [-wk <unwrapping-key-file> ]
```

Ejemplos

En los ejemplos siguientes, se muestra cómo se utiliza exSymKey para exportar claves simétricas de su propiedad desde los HSM.

Example : exportación de una clave simétrica 3DES

Este comando exporta una clave simétrica Triple DES (3DES) (identificador de clave 7). Utiliza una clave AES ya existente (identificador de clave 6) del HSM como clave de encapsulación. A continuación, escribe el texto no cifrado de la clave 3DES en el archivo `3DES.key`.

La salida muestra que la clave 7 (la clave 3DES) se ha encapsulado y desencapsulado correctamente y que, a continuación, se ha escrito en el archivo `3DES.key`.

> **Warning**
>
> Aunque la salida indica que se ha escrito una "Wrapped Symmetric Key" (clave simétrica encapsulada) en el archivo de salida, este contiene una clave no cifrada (desencapsulada).

```
        Command: exSymKey -k 7 -w 6 -out 3DES.key

       Cfm3WrapKey returned: 0x00 : HSM Return: SUCCESS

        Cfm3UnWrapHostKey returned: 0x00 : HSM Return: SUCCESS


Wrapped Symmetric Key written to file "3DES.key"
```

Example : cómo exportar con una clave de encapsulación solo para la sesión

En este ejemplo se muestra cómo utilizar una clave que solo existe en la sesión como clave de encapsulación. Dado que la clave para la exportación se encapsula, se desencapsula inmediatamente y se entrega sin cifrar, no es necesario conservar la clave de encapsulación.

Esta serie de comandos exporta una clave AES con el identificador de clave 8 del HSM. Utiliza una clave de sesión AES creada especialmente para este fin.

El primer comando se utiliza genSymKeypara crear una clave AES de 256 bits. Utiliza el parámetro `-sess` para crear una clave que solo existe en la sesión actual.

La salida muestra que el HSM crea la clave 262168.

```
        Command:  genSymKey -t 31 -s 32 -l AES-wrapping-key -sess

        Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Created.  Key Handle: 262168

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
```

A continuación, el ejemplo comprueba que la clave 8, es decir, la clave que se va a exportar, sea una clave simétrica que se puede extraer. También comprueba que la clave de encapsulación, clave 262168, sea una clave AES que solo exista en la sesión. Puede ejecutar el comando findKey, pero este ejemplo exporta los atributos de ambas claves a archivos y, a continuación, utiliza `grep` para encontrar los valores de atributos pertinentes en el archivo.

Estos comandos utilizan `getAttribute` con un valor `-a` de 512 (todos) para obtener todos los atributos de las claves 8 y 262168. Para obtener información sobre los atributos de las claves, consulte la the section called “Referencia de los atributos de claves”.

```
getAttribute -o 8 -a 512 -out attributes/attr_8
getAttribute -o 262168 -a 512 -out attributes/attr_262168
```

Estos comandos ejecutan `grep` para verificar los atributos de la clave que se va a exportar (clave 8) y la clave de encapsulación solo para la sesión (clave 262168).

```
    // Verify that the key to be exported is a symmetric key.
    $  grep -A 1 "OBJ_ATTR_CLASS" attributes/attr_8
    OBJ_ATTR_CLASS
    0x04

    // Verify that the key to be exported is extractable.
    $  grep -A 1 "OBJ_ATTR_KEY_TYPE" attributes/attr_8
    OBJ_ATTR_EXTRACTABLE
    0x00000001

    // Verify that the wrapping key is an AES key
    $  grep -A 1 "OBJ_ATTR_KEY_TYPE" attributes/attr_262168
    OBJ_ATTR_KEY_TYPE
    0x1f

    // Verify that the wrapping key is a session key
    $  grep -A 1 "OBJ_ATTR_TOKEN" attributes/attr_262168
    OBJ_ATTR_TOKEN
    0x00

    // Verify that the wrapping key can be used for wrapping
     $  grep -A 1 "OBJ_ATTR_WRAP" attributes/attr_262168
    OBJ_ATTR_WRAP
    0x00000001
```

Por último, utilizamos un comando exSymKey para exportar una clave 8 utilizando la clave de sesión (clave 262168) como clave de encapsulado.

Cuando finaliza la sesión, la clave 262168 ya no existe.

```
        Command:  exSymKey -k 8 -w 262168 -out aes256_H8.key

        Cfm3WrapKey returned: 0x00 : HSM Return: SUCCESS

        Cfm3UnWrapHostKey returned: 0x00 : HSM Return: SUCCESS


Wrapped Symmetric Key written to file "aes256_H8.key"
```

Example : utilización de una clave de desencapsulación externa

En este ejemplo se muestra cómo utilizar una clave de desencapsulación externa para exportar una clave desde el HSM.

Cuando exporta una clave desde el HSM, usted especifica una clave AES en el HSM para que sea la clave de encapsulación. De forma predeterminada, esa clave de encapsulación se usa para encapsular y desencapsular la clave que se va a exportar. Sin embargo, puede utilizar el parámetro `-wk` para indicar a exSymKey que utilice una clave externa de un archivo del disco para desencapsularla. Si lo hace, la clave especificada por el parámetro `-w` encapsula la clave de destino y la clave del archivo especificado por el parámetro `-wk` desencapsula la clave.

Dado que la clave de encapsulación tiene que ser una clave AES, que es simétrica, la clave de encapsulación del HSM y la clave de desencapsulación del disco han de tener el mismo material de clave. Para ello, debe importar la clave de encapsulación al HSM o exportarla desde el HSM antes de la operación de exportación.

En este ejemplo se crea una clave fuera del HSM y se importa al HSM. Se utiliza la copia interna de la clave para encapsular una clave simétrica que se exporta y la copia de clave del archivo para desencapsularla.

El primer comando utiliza OpenSSL para generar una clave AES de 256 bits. Guarda la clave en el archivo `aes256-forImport.key`. El comando OpenSSL no devuelve una salida, pero puede utilizar varios comandos para confirmar que todo se ha realizado correctamente. En este ejemplo, se utiliza la herramienta wc (recuento de palabras), que confirma que el archivo contiene 32 bytes de datos.

```
$  openssl rand -out keys/aes256-forImport.key 32

$ wc keys/aes256-forImport.key
 0  2 32 keys/aes256-forImport.key
```

Este comando utiliza el imSymKeycomando para importar la clave AES del `aes256-forImport.key` archivo al HSM. Cuando se completa el comando, la clave existe en el HSM con el identificador de clave 262167 y en el archivo `aes256-forImport.key`.

```
Command:  imSymKey -f keys/aes256-forImport.key -t 31 -l aes256-imported -w 6

        Cfm3WrapHostKey returned: 0x00 : HSM Return: SUCCESS

        Cfm3CreateUnwrapTemplate returned: 0x00 : HSM Return: SUCCESS

        Cfm3UnWrapKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Unwrapped.  Key Handle: 262167

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Este comando utiliza la clave en una operación de exportación. El comando utiliza exSymKey para exportar la clave 21, una clave AES de 192 bits. Para encapsular la clave utiliza la clave 262167, que es la copia que se importó en el HSM. Para desencapsular la clave, utiliza el mismo material de clave en el archivo `aes256-forImport.key`. Cuando se completa el comando, la clave 21 se exporta al archivo `aes192_h21.key`.

```
        Command:  exSymKey -k 21 -w 262167 -out aes192_H21.key -wk aes256-forImport.key

        Cfm3WrapKey returned: 0x00 : HSM Return: SUCCESS

Wrapped Symmetric Key written to file "aes192_H21.key"
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-k

Especifica el identificador de la clave que se va a exportar. Este parámetro es obligatorio. Escriba el identificador de una clave simétrica de su propiedad. Este parámetro es obligatorio. Para buscar identificadores de clave, use el comando findKey.

Para verificar que se puede exportar una clave, ejecute el comando getAttribute para obtener el valor del atributo `OBJ_ATTR_EXTRACTABLE`, que se representa con la constante `354`. Además, puede exportar únicamente claves de su propiedad. Para encontrar el propietario de una clave, utilice el getKeyInfocomando.

Obligatorio: sí

-w

Especifica el identificador de la clave de encapsulamiento. Este parámetro es obligatorio. Para buscar identificadores de clave, use el comando findKey.

Una clave de encapsulación es una clave del HSM que se utiliza para cifrar (encapsular) y después descifrar (desencapsular) la clave que se va a exportar. Solo las claves AES se pueden utilizar como claves de encapsulación.

Puede utilizar cualquier clave AES (de cualquier tamaño) como clave de encapsulación. Dado que la clave de encapsulación encapsula y, a continuación, desencapsula inmediatamente la clave de destino, puede utilizar una clave AES solo de una sesión como clave de encapsulación. Para determinar si una clave se puede utilizar como clave de encapsulación, utilice getAttribute para obtener el valor del atributo `OBJ_ATTR_WRAP`, que se representa con la constante `262`. Para crear una clave de empaquetado, utilice genSymKeypara crear una clave AES (tipo 31).

Si utiliza el parámetro `-wk` para especificar una clave de desencapsulación externa, la clave de encapsulación `-w` se utiliza para encapsular, pero no desencapsular, la clave durante la exportación.

> **Note**
>
> La clave 4 representa una clave interna incompatible. Le recomendamos que use una clave AES que cree y administre como clave de encapsulamiento.

Obligatorio: sí

-out

Especifica la ruta y el nombre del archivo de salida. Cuando el comando se ejecuta correctamente, este archivo contiene la clave exportada sin cifrar. Si el archivo ya existe, el comando lo sobrescribe sin ningún tipo de advertencia.

Obligatorio: sí

-m

Especifica el mecanismo de encapsulación. El único valor válido es `4`, que representa el mecanismo `NIST_AES_WRAP`.

Obligatorio: no

Predeterminado: 4

-wk

Utilice la clave AES del archivo especificado para desencapsular la clave que se exporta. Escriba la ruta y el nombre de un archivo que contenga una clave AES sin cifrar.

Si se incluye este parámetro, `exSymKey` utiliza la clave del HSM que se especificó en el parámetro `-w` para encapsular la clave que se va a exportar y utiliza la clave del archivo `-wk` para desencapsularla. Los valores de parámetro `-w` y `-wk` deben resolverse en la misma clave sin cifrar.

Obligatorio: no

Valor predeterminado: utilice la clave de encapsulación del HSM para realizar la desencapsulación.

Temas relacionados de

- genSymKey
- imSymKey
- wrapKey

extractMaskedObject

El comando extractMaskedObject de key_mgmt_util extrae una clave de un HSM y la guarda en un archivo como un objeto enmascarado. Los objetos enmascarados son objetos clonados que

solo se pueden utilizar después volver a insertarlos en el clúster original mediante el comando insertMaskedObject. Solo puede insertar un objeto enmascarado en el mismo clúster desde el que se generó, o en un clon de ese clúster. Esto incluye cualquier versión clonada del clúster generada al copiar una copia de seguridad entre regiones y al utilizar la copia de seguridad para crear un clúster nuevo.

Los objetos enmascarados son una forma eficaz de descargar y sincronizar claves, incluidas las claves no extraíbles (es decir, las claves que tienen un valor de `OBJ_ATTR_EXTRACTABLE` igual a `0`). De esta forma, las claves se pueden sincronizar de forma segura en clústeres relacionados en diferentes regiones sin necesidad de actualizar el archivo de configuración. AWS CloudHSM

> **Important**
>
> Tras su inserción, los objetos enmascarados se descifran y se les asigna un identificador de clave que es distinto del identificador de clave de la clave original. Un objeto enmascarado incluye todos los metadatos asociados a la clave original, incluidos los atributos, la información de propiedad y uso compartido y la configuración de cuórum. Si necesita sincronizar claves entre los clústeres de una aplicación, utilice syncKey en cloudhsm_mgmt_util en su lugar.

Para poder ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e iniciar sesión en el HSM. El comando extractMaskedObject lo puede utilizar el CU que es el propietario de la clave o cualquier CO.

Sintaxis

```
extractMaskedObject -h

extractMaskedObject -o <object-handle>
                    -out <object-file>
```

Ejemplos

Este ejemplo muestra cómo utilizar extractMaskedObject para extraer una clave de un HSM como un objeto enmascarado.

Example : extraer un objeto enmascarado.

Este comando extrae un objeto enmascarado de un HSM a partir de la clave con el identificador `524295` y lo guarda como un archivo denominado `maskedObj`. Cuando el comando se ejecuta correctamente, extractMaskedObject devuelve un mensaje de confirmación.

```
Command: extractMaskedObject -o 524295 -out maskedObj

Object was masked and written to file "maskedObj"

        Cfm3ExtractMaskedObject returned: 0x00 : HSM Return: SUCCESS
```

Parámetros

Este comando admite los siguientes parámetros.

`-h`

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

`-o`

Especifica el identificador de la clave que se va a extraer como objeto enmascarado.

Obligatorio: sí

`-out`

Especifica el nombre del archivo en el que se guardará el objeto enmascarado.

Obligatorio: sí

Temas relacionados de

- insertMaskedObject
- syncKey
- Copia de una copia de seguridad entre regiones
- Creación de un AWS CloudHSM clúster a partir de un Backup anterior

findKey

Utilice el comando findKey en key_mgmt_util para buscar claves mediante los valores de los atributos de clave. Cuando una clave coincide con todos los criterios que ha establecido, findKey devuelve el identificador de clave. Si no se especifica ningún parámetro, findKey devuelve los identificadores de todas las claves que se pueden utilizar en el HSM. Para encontrar los valores de atributo de una clave en particular, utilice getAttribute.

Al igual que todos los comandos de key_mgmt_util, findKey es específico del usuario. Devuelve solo las claves que el usuario actual puede utilizar en las operaciones criptográficas. Esto incluye las claves que el usuario actual posee y claves que se han compartido con el usuario actual.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
findKey -h

findKey [-c <key class>]
        [-t <key type>]
        [-l <key label>]
        [-id <key ID>]
        [-sess (0 | 1)]
        [-u <user-ids>]
        [-m <modulus>]
        [-kcv <key_check_value>]
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza findKey para encontrar e identificar claves en los HSM.

Example : búsqueda de todas las claves

Este comando encuentra todas las claves para el usuario actual en el HSM. El resultado incluye claves que el usuario posee y comparte, y todas las claves públicas en los HSM.

Para obtener los atributos de una clave con un identificador de clave particular, utilice getAttribute. Para determinar si el usuario actual posee o comparte una clave en particular, utilice o en cloudhsm_mgmt_util. getKeyInfofindAllKeys

```
Command: findKey

Total number of keys present 13

 number of keys matched from start index 0::12
6, 7, 524296, 9, 262154, 262155, 262156, 262157, 262158, 262159, 262160, 262161, 262162

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS

        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

Example : búsqueda de claves por tipo, usuario y sesión

Este comando encuentra claves de AES persistentes que el usuario actual y el usuario 3 pueden utilizar. (Es posible que el usuario 3 pueda utilizar otras claves que el usuario actual no puede ver.)

```
Command: findKey -t 31 -sess 0 -u 3
```

Example : búsqueda de claves por clase y etiqueta

Este comando encuentra todas las claves públicas para el usuario actual con la etiqueta `2018-sept`.

```
Command: findKey -c 2 -l 2018-sept
```

Example : búsqueda de claves RSA por módulo

Este comando encuentra claves RSA (tipo 0) para el usuario actual que se crearon utilizando el módulo en el archivo `m4.txt`.

```
Command: findKey -t 0 -m m4.txt
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-t

Encuentra claves del tipo especificado. Escriba la constante que representa la clase de clave. Por ejemplo, para encontrar las claves 3DES, escriba `-t 21`.

Valores válidos:

- 0: RSA
- 1: DSA
- 3: EC
- 16: GENERIC_SECRET
- 18: RC4
- 21: Triple DES (3DES)
- 31: AES

Obligatorio: no

-c

Encuentra claves en la clase especificada. Escriba la constante que representa la clase de clave. Por ejemplo, para encontrar claves públicas, escriba `-c 2`.

Valores válidos para cada tipo de clave:

- 2: pública. Esta clase contiene las claves públicas de los pares de claves públicas-privadas.
- 3: privada. Esta clase contiene las claves privadas de los pares de claves públicas-privadas.
- 4: secreta. Esta clase contiene todas las claves simétricas.

Obligatorio: no

-l

Encuentra claves con la etiqueta especificada. Escriba la etiqueta exacta. No puede utilizar caracteres comodín ni expresiones regulares en el valor `--l`.

Obligatorio: no

-id

Encuentra la clave con el ID especificado. Escriba la cadena exacta del ID. No puede utilizar caracteres comodín ni expresiones regulares en el valor `-id`.

Obligatorio: no

-sess

Encuentra claves por estado de la sesión. Para encontrar claves que solo sean válidas en la sesión actual, escriba 1. Para encontrar claves persistentes, escriba `0`.

Obligatorio: no

-u

Encuentra claves que los usuarios especificados y el actual usuario comparten. Escriba una lista separada por comas de los ID de usuario de HSM, como `-u 3` o `-u 4,7`. Para encontrar los ID de los usuarios en un HSM, utilice listUsers.

Si se especifica un ID de usuario, findKey devuelve las claves de ese usuario. Si se especifican varios ID de usuario, findKey devuelve las claves que todos los usuarios especificados pueden utilizar.

Como findKey solo devuelve las claves que el usuario actual puede utilizar, los resultados de `-u` son siempre idénticos a las claves del usuario actual, o a un subconjunto de ellas. Para obtener todas las claves que son propiedad de cualquier usuario o que comparten con él, los oficiales de cifrado (CO) pueden utilizar cloudhsm_mgmt_util. findAllKeys

Obligatorio: no

-m

Encuentra claves que se crearon utilizando el módulo RSA en el archivo especificado. Escriba la ruta al archivo que almacena el módulo.

-m especifica el archivo binario que contiene el módulo RSA con el que debe coincidir (opcional).

Obligatorio: no

-kcv

Encuentra claves con el valor de comprobación de clave que se ha especificado.

El valor de comprobación de claves (KCV) es un hash o suma de comprobación de 3 bytes de una clave que se genera cuando el HSM importa o genera una clave. También puede calcular un KCV fuera del HSM, por ejemplo, después de exportar una clave. A continuación, puede comparar los valores del KCV para confirmar la identidad e integridad de la clave. Para obtener el KCV (valor de control de la clave), utilice getAttribute.

AWS CloudHSM utiliza el siguiente método estándar para generar un valor de comprobación clave:

- Claves simétricas: los primeros 3 bytes del resultado obtenido al cifrar un bloque cero con la clave.
- Pares de claves asimétricas: los primeros 3 bytes del hash SHA-1 de la clave pública.
- Claves HMAC: por el momento, no se admite el uso del KCV con claves HMAC.

Obligatorio: no

Salida

El resultado de findKey muestra el número total de claves coincidentes y sus identificadores de clave

```
        Command:  findKey
Total number of keys present 10

 number of keys matched from start index 0::9
6, 7, 8, 9, 10, 11, 262156, 262157, 262158, 262159

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS

        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

Temas relacionados de

- findSingleKey
- getKeyInfo
- getAttribute
- findAllKeysen cloudhsm_mgmt_util
- Referencia de los atributos de claves

findSingleKey

El comando findSingleKey de la herramienta key_mgmt_util comprueba si existe una clave en todos los HSM del clúster.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
findSingleKey -h

findSingleKey -k <key-handle>
```

Ejemplo

Example

Este comando verifica que la clave 252136 exista en los tres HSM del clúster.

```
Command: findSingleKey -k 252136
Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS

        Cluster Error Status
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-k

Especifica el identificador de una clave en el HSM. Este parámetro es obligatorio.

Para buscar identificadores de clave, use el comando findKey.

Obligatorio: sí

Temas relacionados de

- findKey

- getKeyInfo
- getAttribute

GendA KeyPair

El comando genDSAKeyPair de la herramienta key_mgmt_util genera un par de claves de algoritmo de firma digital (DSA) en los HSM. Debe especificar la longitud del módulo; el comando genera el valor del módulo. También puede asignar un ID, compartir la clave con otros usuarios del HSM, crear claves no extraíbles y claves que caduquen cuando la sesión finaliza. Cuando el comando se ejecuta correctamente, devuelve identificadores de clave que el HSM asigna a las claves públicas y privadas. Puede utilizar estos identificadores de clave para identificar las claves ante otros comandos.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

> **Tip**
>
> Para buscar los atributos de una clave que haya creado, como tipo, longitud, etiqueta e ID, use getAttribute. Para buscar las claves de un usuario concreto, utilice. getKeyInfo Para buscar claves en función de sus valores de atributo, use findKey.

Sintaxis

```
genDSAKeyPair -h

genDSAKeyPair -m <modulus length>
              -l <label>
              [-id <key ID>]
              [-min_srv <minimum number of servers>]
              [-m_value <0..8>]
              [-nex]
              [-sess]
              [-timeout <number of seconds> ]
              [-u <user-ids>]
              [-attest]
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza genDSAKeyPair para crear un par de claves de DSA.

Example : creación de un par de claves de DSA

Este comando crea un par de claves de DSA con una etiqueta DSA. La salida muestra que el identificador de clave de la clave pública es 19, mientras que el identificador de la clave privada es 21.

```
Command: genDSAKeyPair -m 2048 -l DSA

        Cfm3GenerateKeyPair: returned: 0x00 : HSM Return: SUCCESS

        Cfm3GenerateKeyPair:    public key handle: 19    private key handle: 21

        Cluster Error Status
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Example : creación de un par de claves de DSA solo para la sesión

Este comando crea un par de claves de DSA que es válido únicamente en la sesión actual. El comando asigna el ID exclusivo `DSA_temp_pair` además de la etiqueta obligatoria (que no es exclusiva). Es posible que le interese crear un par de claves de este tipo para firmar y verificar un token de una sola sesión. La salida muestra que el identificador de clave de la clave pública es 12, mientras que el identificador de la clave privada es 14.

```
Command: genDSAKeyPair -m 2048 -l DSA-temp -id DSA_temp_pair -sess

        Cfm3GenerateKeyPair: returned: 0x00 : HSM Return: SUCCESS

        Cfm3GenerateKeyPair:    public key handle: 12    private key handle: 14

        Cluster Error Status
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Para confirmar que el par de claves exista únicamente en la sesión, utilice el parámetro `-sess` de findKey con el valor 1 (true).

```
  Command: findKey -sess 1

  Total number of keys present 2

 number of keys matched from start index 0::1
12, 14
```

```
Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS

Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

Example : creación de un par de claves de DSA no extraíbles y compartidas

Este comando crea un par de claves de DSA. La clave privada se comparte con otros tres usuarios y no se puede exportar desde el HSM. Cualquier usuario puede utilizar las claves públicas y estas siempre se pueden extraer.

```
Command:  genDSAKeyPair -m 2048 -l DSA -id DSA_shared_pair -nex -u 3,5,6

Cfm3GenerateKeyPair: returned: 0x00 : HSM Return: SUCCESS

Cfm3GenerateKeyPair:    public key handle: 11    private key handle: 19

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Example : creación de un par de claves controladas mediante cuórum

Este comando crea un par de claves de DSA con la etiqueta `DSA-mV2`. El comando utiliza el parámetro `-u` para compartir la clave privada con los usuarios 4 y 6. Utiliza el parámetro `-m_value` para solicitar un cuórum de al menos dos aprobaciones para cualquier operación criptográfica que utilice la clave privada. El comando también utiliza el parámetro `-attest` para verificar la integridad del firmware en el que se genera el par de claves.

La salida muestra que el comando genera una clave pública con el identificador de clave 12 y una clave privada con el identificador de clave 17 y que la comprobación de declaración del firmware del clúster ha pasado.

```
Command:  genDSAKeyPair -m 2048 -l DSA-mV2 -m_value 2 -u 4,6 -attest

Cfm3GenerateKeyPair: returned: 0x00 : HSM Return: SUCCESS

Cfm3GenerateKeyPair:    public key handle: 12    private key handle: 17

Attestation Check : [PASS]
```

```
        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Este comando utiliza getKeyInfola clave privada (identificador de clave17). La salida confirma que la clave es propiedad del usuario actual (usuario 3) y que se comparte con los usuarios 4 y 6 (y nadie más). La salida también muestra que la autenticación de cuórum está habilitada y que el cuórum es de dos.

```
        Command:  getKeyInfo -k 17

        Cfm3GetKey returned: 0x00 : HSM Return: SUCCESS

        Owned by user 3

        also, shared to following 2 user(s):

                4
                6
         2 Users need to approve to use/manage this key
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-m

Especifica la longitud del módulo en bits. El único valor válido es `2048`.

Obligatorio: sí

-l

Especifica una etiqueta definida por el usuario para el par de claves. Tipo de cadena. Se aplica la misma etiqueta a las dos claves del par. El tamaño máximo permitido para `label` es de 127 caracteres.

Puede utilizar cualquier frase que le ayude a identificar la clave. La etiqueta no tiene por qué ser única, por lo que puede usarla para agrupar y clasificar las claves.

Obligatorio: sí

-id

Especifica un identificador definido por el usuario para el par de claves. Escriba una cadena que sea única en el clúster. El valor predeterminado es una cadena vacía. La ID que especifique se aplicará a las dos claves del par.

Predeterminado: sin valor de ID.

Obligatorio: no

-min_srv

Especifica el número mínimo de HSM en los que la clave importada se sincroniza antes de que caduque el valor del parámetro `-timeout`. Si la clave no está sincronizada con el número especificado de servidores en el tiempo asignado, no se creará.

AWS CloudHSM sincroniza automáticamente todas las claves con todos los HSM del clúster. Para acelerar el proceso, establezca el valor de `min_srv` en un número menor que el de HSM del clúster y establezca un valor bajo de tiempo de espera. Sin embargo, tenga en cuenta que puede que algunas solicitudes no generen ninguna clave.

Predeterminado: 1

Obligatorio: no

-m_value

Especifica el número de usuarios que deben aprobar cualquier operación criptográfica que utilice la clave importada en el par. Escriba un valor de `0` a `8`.

Este parámetro establece un requisito de autenticación de cuórum para la clave privada. El valor predeterminado, `0`, deshabilita la característica de autenticación de cuórum para la clave. Cuando la autenticación de cuórum esté habilitada, el número especificado de usuarios deberá firmar un token para aprobar las operaciones criptográficas que empleen clave privada y las operaciones de compartir o dejar de compartir la clave privada.

Para encontrar la clave `m_value` de una clave, utilice. getKeyInfo

Este parámetro solo es válido cuando el parámetro `-u` del comando comparte la clave con suficientes usuarios para satisfacer el requisito `m_value`.

Predeterminado: 0

Obligatorio: no

-nex

Hace que la clave privada no se pueda extraer. La clave privada que se genera no se podrá exportar desde el HSM. Las claves públicas siempre se pueden extraer.

Predeterminado: tanto las claves públicas como las privadas del par de claves se pueden extraer.

Obligatorio: no

-sess

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use setAttribute.

Predeterminado: la clave es persistente.

Obligatorio: no

-timeout

Especifica cuánto tiempo (en segundos) espera el comando para que una clave se sincronice con el número de HSM especificado por el parámetro `min_srv`.

Este parámetro solo es válido cuando también se usa el parámetro `min_srv` en el comando.

Predeterminado: sin tiempo de espera predeterminado. El comando espera indefinidamente y solo vuelve a aparecer cuando la clave está sincronizada con el número mínimo de servidores.

Obligatorio: no

-u

Comparte la clave privada del par con los usuarios especificados. Este parámetro concede permiso a otros usuarios de criptografía (CU) del HSM para usar esta clave privada en

operaciones criptográficas. Cualquier usuario puede utilizar las claves públicas sin necesidad de compartirlas.

Escriba una lista separada por comas de los ID de usuario de HSM, como `-u 5,6`. No incluya el ID de usuario de HSM del usuario actual. Para buscar los ID de usuario de HSM de los CU del HSM, utilice listUsers. Para compartir o dejar de compartir una clave existente, utilice shareKey en cloudhsm_mgmt_util.

Valor predeterminado: solo el usuario actual puede utilizar la clave privada.

Obligatorio: no

-attest

Ejecuta una comprobación de integridad que verifica que el firmware en el que se ejecuta el clúster no haya sufrido alguna manipulación.

Predeterminado: sin comprobación de certificación.

Obligatorio: no

Temas relacionados de

- Género A KeyPair
- genSymKey
- GeneCC KeyPair

GeneCC KeyPair

El comando `genECCKeyPair` de la herramienta key_mgmt_util genera un par de claves de criptografía de curva elíptica (ECC) en los HSM. Cuando ejecute el comando `genECCKeyPair`, debe especificar el identificador de curva elíptica y una etiqueta para el par de claves. También puede compartir la clave privada con otros usuarios de CU, crear claves no extraíbles, claves controladas mediante cuórum y claves que caduquen cuando finalice la sesión. Cuando el comando se ejecuta correctamente, devuelve identificadores de clave que el HSM asigna a las claves ECC públicas y privadas. Puede utilizar estos identificadores de clave para identificar las claves ante otros comandos.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

> **Tip**
>
> Para buscar los atributos de una clave que haya creado, como tipo, longitud, etiqueta e ID, use getAttribute. Para encontrar las claves de un usuario en particular, utilice. getKeyInfo Para buscar claves en función de sus valores de atributo, use findKey.

Sintaxis

```
genECCKeyPair -h

genECCKeyPair -i <EC curve id>
              -l <label>
              [-id <key ID>]
              [-min_srv <minimum number of servers>]
              [-m_value <0..8>]
              [-nex]
              [-sess]
              [-timeout <number of seconds> ]
              [-u <user-ids>]
              [-attest]
```

Ejemplos

En los siguientes ejemplos, se muestra cómo se utiliza genECCKeyPair para crear un par de claves ECC en los HSM.

Example : crear y examinar un par de claves ECC

Este comando utiliza una curva elíptica NID_secp384r1 y una etiqueta `ecc14` para crear un par de claves ECC. El resultado muestra que el identificador de clave de la clave privada es 262177, mientras que el identificador de clave de la clave pública es 262179. La etiqueta se aplica a las clave privada y pública.

```
Command: genECCKeyPair -i 14 -l ecc14

        Cfm3GenerateKeyPair returned: 0x00 : HSM Return: SUCCESS

        Cfm3GenerateKeyPair:    public key handle: 262179    private key handle: 262177
```

```
        Cluster Error Status
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Después de generar la clave, puede examinar sus atributos. Utilice getAttribute para escribir todos los atributos (representados por la constante 512) de la nueva clave privada de ECC en el archivo `attr_262177`.

```
Command: getAttribute -o 262177 -a 512 -out attr_262177
got all attributes of size 529 attr cnt 19
Attributes dumped into attr_262177

        Cfm3GetAttribute returned: 0x00 : HSM Return: SUCCESS
```

A continuación, utilice el comando `cat` para ver el contenido del archivo de atributos `attr_262177`. El resultado muestra que la clave es una clave privada de curva elíptica que puede usarse para firmar, pero no para cifrar, descifrar, encapsular, desencapsular o verificar. La clave es persistente y exportable.

```
$  cat attr_262177

OBJ_ATTR_CLASS
0x03
OBJ_ATTR_KEY_TYPE
0x03
OBJ_ATTR_TOKEN
0x01
OBJ_ATTR_PRIVATE
0x01
OBJ_ATTR_ENCRYPT
0x00
OBJ_ATTR_DECRYPT
0x00
OBJ_ATTR_WRAP
0x00
OBJ_ATTR_UNWRAP
0x00
OBJ_ATTR_SIGN
0x01
OBJ_ATTR_VERIFY
0x00
```

```
OBJ_ATTR_LOCAL
0x01
OBJ_ATTR_SENSITIVE
0x01
OBJ_ATTR_EXTRACTABLE
0x01
OBJ_ATTR_LABEL
ecc2
OBJ_ATTR_ID

OBJ_ATTR_VALUE_LEN
0x0000008a
OBJ_ATTR_KCV
0xbbb32a
OBJ_ATTR_MODULUS
044a0f9d01d10f7437d9fa20995f0cc742552e5ba16d3d7e9a65a33e20ad3e569e68eb62477a9960a87911e6121d112
OBJ_ATTR_MODULUS_BITS
0x0000019f
```

Example Uso de una curva de EEC no válida

Este comando intenta crear un par de claves ECC utilizando una curva NID_X9_62_prime192v1. Debido a que esta curva elíptica no es válida para los HSM de modo FIPS-mode HSM, el comando produce un error. El mensaje informa que un servidor en el clúster no está disponible, pero esto no suele indicar un problema con los HSM en el clúster.

```
Command:  genECCKeyPair -i 1 -l ecc1

       Cfm3GenerateKeyPair returned: 0xb3 : HSM Error: This operation violates the
 current configured/FIPS policies

       Cluster Error Status
       Node id 0 and err state 0x30000085 : HSM CLUSTER ERROR: Server in cluster is
 unavailable
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-i

Especifica el identificador de la curva elíptica. Escriba un identificador.

Valores válidos:

- 2: NID_X9_62_prime256v1
- 14: NID_secp384r1
- 16: NID_secp256k1

Obligatorio: sí

-l

Especifica una etiqueta definida por el usuario para el par de claves. Tipo de cadena. Se aplica la misma etiqueta a las dos claves del par. El tamaño máximo permitido para `label` es de 127 caracteres.

Puede utilizar cualquier frase que le ayude a identificar la clave. La etiqueta no tiene por qué ser única, por lo que puede usarla para agrupar y clasificar las claves.

Obligatorio: sí

-id

Especifica un identificador definido por el usuario para el par de claves. Escriba una cadena que sea única en el clúster. El valor predeterminado es una cadena vacía. La ID que especifique se aplicará a las dos claves del par.

Predeterminado: sin valor de ID.

Obligatorio: no

-min_srv

Especifica el número mínimo de HSM en los que la clave importada se sincroniza antes de que caduque el valor del parámetro `-timeout`. Si la clave no está sincronizada con el número especificado de servidores en el tiempo asignado, no se creará.

AWS CloudHSM sincroniza automáticamente todas las claves con todos los HSM del clúster. Para acelerar el proceso, establezca el valor de `min_srv` en un número menor que el de HSM del clúster y establezca un valor bajo de tiempo de espera. Sin embargo, tenga en cuenta que puede que algunas solicitudes no generen ninguna clave.

Predeterminado: 1

Obligatorio: no

-m_value

Especifica el número de usuarios que deben aprobar cualquier operación criptográfica que utilice la clave importada en el par. Escriba un valor de `0` a `8`.

Este parámetro establece un requisito de autenticación de cuórum para la clave privada. El valor predeterminado, `0`, deshabilita la característica de autenticación de cuórum para la clave. Cuando la autenticación de cuórum esté habilitada, el número especificado de usuarios deberá firmar un token para aprobar las operaciones criptográficas que empleen clave privada y las operaciones de compartir o dejar de compartir la clave privada.

Para encontrar la clave `m_value` de una clave, utilice. getKeyInfo

Este parámetro solo es válido cuando el parámetro `-u` del comando comparte la clave con suficientes usuarios para satisfacer el requisito `m_value`.

Predeterminado: 0

Obligatorio: no

-nex

Hace que la clave privada no se pueda extraer. La clave privada que se genera no se podrá exportar desde el HSM. Las claves públicas siempre se pueden extraer.

Predeterminado: tanto las claves públicas como las privadas del par de claves se pueden extraer.

Obligatorio: no

-sess

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use setAttribute.

Predeterminado: la clave es persistente.

Obligatorio: no

-timeout

Especifica cuánto tiempo (en segundos) espera el comando para que una clave se sincronice con el número de HSM especificado por el parámetro `min_srv`.

Este parámetro solo es válido cuando también se usa el parámetro `min_srv` en el comando.

Predeterminado: sin tiempo de espera predeterminado. El comando espera indefinidamente y solo vuelve a aparecer cuando la clave está sincronizada con el número mínimo de servidores.

Obligatorio: no

-u

Comparte la clave privada del par con los usuarios especificados. Este parámetro concede permiso a otros usuarios de criptografía (CU) del HSM para usar esta clave privada en operaciones criptográficas. Cualquier usuario puede utilizar las claves públicas sin necesidad de compartirlas.

Escriba una lista separada por comas de los ID de usuario de HSM, como `-u 5,6`. No incluya el ID de usuario de HSM del usuario actual. Para buscar los ID de usuario de HSM de los CU del HSM, utilice listUsers. Para compartir o dejar de compartir una clave existente, utilice shareKey en cloudhsm_mgmt_util.

Valor predeterminado: solo el usuario actual puede utilizar la clave privada.

Obligatorio: no

-attest

Ejecuta una comprobación de integridad que verifica que el firmware en el que se ejecuta el clúster no haya sufrido alguna manipulación.

Predeterminado: sin comprobación de certificación.

Obligatorio: no

Temas relacionados de

- genSymKey
- Género A KeyPair
- GenDSA KeyPair

Género A KeyPair

El comando genRSAKeyPair de la herramienta key_mgmt_util genera un par de claves asimétricas RSA. Deberá especificar el tipo de clave, la longitud del módulo y un exponente público. El comando genera un módulo de la longitud especificada y crea el par de claves. Puede asignar un ID, compartir la clave con otros usuarios del HSM, crear claves no extraíbles y claves que caduquen cuando la sesión finaliza. Cuando el comando se ejecuta correctamente, devuelve un identificador de clave que el HSM asigna a la clave. Puede utilizar el identificador de clave para identificar la clave ante otros comandos.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

> **Tip**
>
> Para buscar los atributos de una clave que haya creado, como tipo, longitud, etiqueta e ID, use getAttribute. Para buscar las claves de un usuario concreto, utilice. getKeyInfo Para buscar claves en función de sus valores de atributo, use findKey.

Sintaxis

```
genRSAKeyPair -h

genRSAKeyPair -m <modulus length>
              -e <public exponent>
              -l <label>
              [-id <key ID>]
              [-min_srv <minimum number of servers>]
              [-m_value <0..8>]
              [-nex]
              [-sess]
              [-timeout <number of seconds> ]
              [-u <user-ids>]
              [-attest]
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza genRSAKeyPair para crear pares de claves asimétricas en los HSM.

Example : creación y examen de un par de claves RSA

Este comando crea un par de claves RSA con un módulo de 2048 bits y un exponente de 65537. La salida muestra que el identificador de clave pública es 2100177, mientras que el identificador de la clave privada es 2100426.

```
Command: genRSAKeyPair -m 2048 -e 65537 -l rsa_test

Cfm3GenerateKeyPair returned: 0x00 : HSM Return: SUCCESS

       Cfm3GenerateKeyPair:    public key handle: 2100177    private key handle:
 2100426

       Cluster Status:
       Node id 0 status: 0x00000000 : HSM Return: SUCCESS
       Node id 1 status: 0x00000000 : HSM Return: SUCCESS
```

El siguiente comando utiliza getAttribute para obtener los atributos de la clave pública que acabamos de crear. Escribe la salida en el archivo `attr_2100177`. Va seguido de un comando cat que obtiene el contenido del archivo de atributos. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Los valores hexadecimales resultantes confirman que se trata de una clave pública (`OBJ_ATTR_CLASS 0x02`) con un tipo de RSA (`OBJ_ATTR_KEY_TYPE 0x00`). Puede utilizar esta clave pública para cifrar (`OBJ_ATTR_ENCRYPT 0x01`), pero no para descifrar (`OBJ_ATTR_DECRYPT 0x00`). Los resultados también incluyen la longitud de la clave (512, `0x200`), el módulo, la longitud del módulo (2048, `0x800`) y el exponente público (65537, `0x10001`).

```
Command:  getAttribute -o 2100177 -a 512 -out attr_2100177

Attribute size: 801, count: 26
Written to: attr_2100177 file

       Cfm3GetAttribute returned: 0x00 : HSM Return: SUCCESS

$  cat attr_2100177
OBJ_ATTR_CLASS
0x02
OBJ_ATTR_KEY_TYPE
0x00
OBJ_ATTR_TOKEN
0x01
```

```
OBJ_ATTR_PRIVATE
0x01
OBJ_ATTR_ENCRYPT
0x01
OBJ_ATTR_DECRYPT
0x00
OBJ_ATTR_WRAP
0x01
OBJ_ATTR_UNWRAP
0x00
OBJ_ATTR_SIGN
0x00
OBJ_ATTR_VERIFY
0x01
OBJ_ATTR_LOCAL
0x01
OBJ_ATTR_SENSITIVE
0x00
OBJ_ATTR_EXTRACTABLE
0x01
OBJ_ATTR_LABEL
rsa_test
OBJ_ATTR_ID

OBJ_ATTR_VALUE_LEN
0x00000200
OBJ_ATTR_KCV
0xc51c18
OBJ_ATTR_MODULUS
0xbb9301cc362c1d9724eb93da8adab0364296bde7124a241087d9436b9be57e4f7780040df03c2c
1c0fe6e3b61aa83c205280119452868f66541bbbffacbbe787b8284fc81deaeef2b8ec0ba25a077d
6983c77a1de7b17cbe8e15b203868704c6452c2810344a7f2736012424cf0703cf15a37183a1d2d0
97240829f8f90b063dd3a41171402b162578d581980976653935431da0c1260bfe756d85dca63857
d9f27a541676cb9c7def0ef6a2a89c9b9304bcac16fdf8183c0a555421f9ad5dfeb534cf26b65873
970cdf1a07484f1c128b53e10209cc6f7ac308669112968c81a5de408e7f644fe58b1a9ae1286fec
b3e4203294a96fae06f8f0db7982cb5d7f
OBJ_ATTR_MODULUS_BITS
0x00000800
OBJ_ATTR_PUBLIC_EXPONENT
0x010001
OBJ_ATTR_TRUSTED
0x00
OBJ_ATTR_WRAP_WITH_TRUSTED
0x00
```

```
OBJ_ATTR_DESTROYABLE
0x01
OBJ_ATTR_DERIVE
0x00
OBJ_ATTR_ALWAYS_SENSITIVE
0x00
OBJ_ATTR_NEVER_EXTRACTABLE
0x00
```

Example : generación de un par de claves RSA compartidas

Este comando genera un par de claves RSA y comparte la clave privada con el usuario 4, otro CU en el HSM. El comando utiliza el parámetro `m_value` para solicitar al menos dos aprobaciones para poder usar la clave privada en el par en una operación criptográfica. Al utilizar el parámetro `m_value`, también tiene que utilizar `-u` en el comando y el `m_value` no puede superar el número total de usuarios (número de valores en `-u` + propietario).

```
Command:  genRSAKeyPair -m 2048 -e 65537 -l rsa_mofn -id rsa_mv2 -u 4 -m_value 2

       Cfm3GenerateKeyPair returned: 0x00 : HSM Return: SUCCESS

       Cfm3GenerateKeyPair:    public key handle: 27    private key handle: 28

       Cluster Error Status
       Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
       Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-m

Especifica la longitud del módulo en bits. El valor mínimo es 2048.

Obligatorio: sí

-e

Especifica el exponente público. El valor debe ser un número impar superior o igual a 65537.

Obligatorio: sí

-l

Especifica una etiqueta definida por el usuario para el par de claves. Tipo de cadena. Se aplica la misma etiqueta a las dos claves del par. El tamaño máximo permitido para `label` es de 127 caracteres.

Puede utilizar cualquier frase que le ayude a identificar la clave. La etiqueta no tiene por qué ser única, por lo que puede usarla para agrupar y clasificar las claves.

Obligatorio: sí

-id

Especifica un identificador definido por el usuario para el par de claves. Escriba una cadena que sea única en el clúster. El valor predeterminado es una cadena vacía. La ID que especifique se aplicará a las dos claves del par.

Predeterminado: sin valor de ID.

Obligatorio: no

-min_srv

Especifica el número mínimo de HSM en los que la clave importada se sincroniza antes de que caduque el valor del parámetro `-timeout`. Si la clave no está sincronizada con el número especificado de servidores en el tiempo asignado, no se creará.

AWS CloudHSM sincroniza automáticamente todas las claves con todos los HSM del clúster. Para acelerar el proceso, establezca el valor de `min_srv` en un número menor que el de HSM del clúster y establezca un valor bajo de tiempo de espera. Sin embargo, tenga en cuenta que puede que algunas solicitudes no generen ninguna clave.

Predeterminado: 1

Obligatorio: no

-m_value

Especifica el número de usuarios que deben aprobar cualquier operación criptográfica que utilice la clave importada en el par. Escriba un valor de `0` a `8`.

Este parámetro establece un requisito de autenticación de cuórum para la clave privada. El valor predeterminado, `0`, deshabilita la característica de autenticación de cuórum para la clave. Cuando

la autenticación de cuórum esté habilitada, el número especificado de usuarios deberá firmar un token para aprobar las operaciones criptográficas que empleen clave privada y las operaciones de compartir o dejar de compartir la clave privada.

Para encontrar la clave `m_value` de una clave, utilice. getKeyInfo

Este parámetro solo es válido cuando el parámetro `-u` del comando comparte la clave con suficientes usuarios para satisfacer el requisito `m_value`.

Predeterminado: 0

Obligatorio: no

-nex

Hace que la clave privada no se pueda extraer. La clave privada que se genera no se podrá exportar desde el HSM. Las claves públicas siempre se pueden extraer.

Predeterminado: tanto las claves públicas como las privadas del par de claves se pueden extraer.

Obligatorio: no

-sess

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use setAttribute.

Predeterminado: la clave es persistente.

Obligatorio: no

-timeout

Especifica cuánto tiempo (en segundos) espera el comando para que una clave se sincronice con el número de HSM especificado por el parámetro `min_srv`.

Este parámetro solo es válido cuando también se usa el parámetro `min_srv` en el comando.

Predeterminado: sin tiempo de espera predeterminado. El comando espera indefinidamente y solo vuelve a aparecer cuando la clave está sincronizada con el número mínimo de servidores.

Obligatorio: no

-u

Comparte la clave privada del par con los usuarios especificados. Este parámetro concede permiso a otros usuarios de criptografía (CU) del HSM para usar esta clave privada en operaciones criptográficas. Cualquier usuario puede utilizar las claves públicas sin necesidad de compartirlas.

Escriba una lista separada por comas de los ID de usuario de HSM, como `-u 5,6`. No incluya el ID de usuario de HSM del usuario actual. Para buscar los ID de usuario de HSM de los CU del HSM, utilice listUsers. Para compartir o dejar de compartir una clave existente, utilice shareKey en cloudhsm_mgmt_util.

Valor predeterminado: solo el usuario actual puede utilizar la clave privada.

Obligatorio: no

-attest

Ejecuta una comprobación de integridad que verifica que el firmware en el que se ejecuta el clúster no haya sufrido alguna manipulación.

Predeterminado: sin comprobación de certificación.

Obligatorio: no

Temas relacionados de

- genSymKey
- GendA KeyPair
- GeneCC KeyPair

genSymKey

El comando genSymKey en la herramienta key_mgmt_util genera una clave simétrica en los HSM. Puede especificar el tipo y el tamaño de la clave, asignar un ID y una etiqueta, y compartir la clave con otros usuarios de HSM. También puede crear claves que no se pueden extraer y claves que caducan al finalizar la sesión. Cuando el comando se ejecuta correctamente, devuelve un identificador de clave que el HSM asigna a la clave. Puede utilizar el identificador de clave para identificar la clave ante otros comandos.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
genSymKey -h

genSymKey -t <key-type>
          -s <key-size>
          -l <label>
          [-id <key-ID>]
          [-min_srv <minimum-number-of-servers>]
          [-m_value <0..8>]
          [-nex]
          [-sess]
          [-timeout <number-of-seconds> ]
          [-u <user-ids>]
          [-attest]
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza genSymKey para crear claves simétricas en los HSM.

> **Tip**
>
> Para usar las claves que crea con estos ejemplos para las operaciones de HMAC, debe establecer `OBJ_ATTR_SIGN` y `OBJ_ATTR_VERIFY` como `TRUE` después de generar la clave. Para establecer estos valores, utilice setAttribute en la Utilidad de administración de CloudHSM (CMU). Para obtener más información, consultesetAttribute.

Example Generación de una clave AES

Este comando crea una clave AES de 256 bits con una etiqueta `aes256`. El resultado muestra que el identificador de clave de la clave nueva es 6.

```
Command: genSymKey -t 31 -s 32 -l aes256

        Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Created.  Key Handle: 6
```

```
        Cluster Error Status
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Example : creación de una clave de sesión

Este comando crea una clave AES de 192 bits no extraíble que es válida únicamente en la sesión actual. Puede que desee crear una clave como esta para encapsular (y, a continuación, desencapsular inmediatamente) una clave que se está exportando.

```
Command: genSymKey -t 31 -s 24 -l tmpAES -id wrap01 -nex -sess
```

Example : regreso rápido

Este comando crea una clave de 512 bytes genéricos con una etiqueta `IT_test_key`. El comando no espera a que la clave se sincronice en todos los HSM en el clúster. En su lugar, regresa tan pronto como se crea la clave en cualquier HSM (`-min_srv 1`) o en 1 segundo (`-timeout 1`), el periodo que sea más corto. Si la clave no se sincroniza con el número mínimo especificado de HSM antes de que venza el tiempo de espera, no se genera. Es posible que desee utilizar un comando como este en un script que crea numerosas claves, como el bucle `for` del siguiente ejemplo.

```
Command: genSymKey -t 16 -s 512 -l IT_test_key -min_srv 1 -timeout 1

$  for i in {1..30};
     do /opt/cloudhsm/bin/key_mgmt_util singlecmd loginHSM -u CU -s example_user -p
 example_pwd genSymKey -l aes -t 31 -s 32 -min_srv 1 -timeout 1;
 done;
```

Example : creación de una clave genérica autorizada mediante cuórum

Este comando crea una clave secreta genérica de 2048 bits con la etiqueta `generic-mV2`.
El comando utiliza el parámetro `-u` para compartir la clave con otro CU, el usuario 6. Utiliza el parámetro `-m_value` para solicitar un cuórum de al menos dos aprobaciones para cualquier operación criptográfica que utilice la clave. El comando también utiliza el parámetro `-attest` para verificar la integridad del firmware en el que se genera la clave.

El resultado muestra que el comando generó una clave con identificador de clave 9 y que se ha superado la comprobación de declaración del firmware del clúster.

```
                Command:  genSymKey -t 16 -s 2048 -l generic-mV2 -m_value 2 -u 6 -
attest
```

```
        Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Created.  Key Handle: 9

        Attestation Check : [PASS]

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Example :creación y examen de una clave

Este comando crea una clave Triple DES con una etiqueta `3DES_shared` y un ID de `IT-02`. La clave la pueden utilizar el usuario actual y los usuarios 4 y 5. Se produce un error en el comando si el ID no es único en el clúster o si el usuario actual es el usuario 4 o el usuario 5.

El resultado muestra que la clave nueva tiene el identificador de clave 7.

```
Command: genSymKey -t 21 -s 24 -l 3DES_shared -id IT-02 -u 4,5

       Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Created.  Key Handle: 7

        Cluster Error Status
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Para verificar que la clave 3DES nueva es propiedad del usuario actual y se comparte con los usuarios 4 y 5, utilice getKeyInfo. El comando utiliza el identificador que se asignó a la clave nueva (`Key Handle: 7`).

El resultado confirma que la clave es propiedad del usuario 3 y se comparte con los usuarios 4 y 5.

```
Command:  getKeyInfo -k 7

        Cfm3GetKey returned: 0x00 : HSM Return: SUCCESS

        Owned by user 3

        also, shared to following 2 user(s):
```

```
                4, 5
```

Para confirmar el resto de las propiedades de la clave, utilice getAttribute. El primer comando utiliza `getAttribute` para obtener todos los atributos (`-a 512`) del identificador de clave 7 (`-o 7`). Los escribe en el archivo `attr_7`. El segundo comando utiliza `cat` para obtener el contenido del archivo `attr_7`.

Este comando confirma que la clave 7 es una clave simétrica de 192 bits (`OBJ_ATTR_VALUE_LEN 0x00000018` o 24 bytes) 3DES (`OBJ_ATTR_KEY_TYPE 0x15`) (`OBJ_ATTR_CLASS 0x04`) con una etiqueta `3DES_shared` (`OBJ_ATTR_LABEL 3DES_shared`) y un ID `IT_02` (`OBJ_ATTR_ID IT-02`). La clave es persistente (`OBJ_ATTR_TOKEN 0x01`) y extraíble (`OBJ_ATTR_EXTRACTABLE 0x01`) y se puede utilizar para el cifrado, el descifrado y el encapsulado.

> **Tip**
>
> Para buscar los atributos de una clave que haya creado, como tipo, longitud, etiqueta e ID, use getAttribute. Para buscar las claves de un usuario concreto, utilice. getKeyInfo Para buscar claves en función de sus valores de atributo, use findKey.

Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

```
Command:  getAttribute -o 7 -a 512 -out attr_7

got all attributes of size 444 attr cnt 17
Attributes dumped into attr_7 file

        Cfm3GetAttribute returned: 0x00 : HSM Return: SUCCESS


$  cat attr_7

OBJ_ATTR_CLASS
0x04
OBJ_ATTR_KEY_TYPE
0x15
OBJ_ATTR_TOKEN
0x01
OBJ_ATTR_PRIVATE
0x01
```

```
OBJ_ATTR_ENCRYPT
0x01
OBJ_ATTR_DECRYPT
0x01
OBJ_ATTR_WRAP
0x00
OBJ_ATTR_UNWRAP
0x00
OBJ_ATTR_SIGN
0x00
OBJ_ATTR_VERIFY
0x00
OBJ_ATTR_LOCAL
0x01
OBJ_ATTR_SENSITIVE
0x01
OBJ_ATTR_EXTRACTABLE
0x01
OBJ_ATTR_LABEL
3DES_shared
OBJ_ATTR_ID
IT-02
OBJ_ATTR_VALUE_LEN
0x00000018
OBJ_ATTR_KCV
0x59a46e
```

> **Tip**
>
> Para usar las claves que crea con estos ejemplos para las operaciones de HMAC, debe establecer `OBJ_ATTR_SIGN` y `OBJ_ATTR_VERIFY` como `TRUE` después de generar la clave. Para establecer estos valores, utilice setAttribute en CMU. Para obtener más información, consultesetAttribute.

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-t

Especifica el tipo de clave simétrica. Escriba la constante que representa el tipo de clave. Por ejemplo, para crear una clave AES, escriba `-t 31`.

Valores válidos:

- 16: GENERIC_SECRET. Una clave secreta genérica es una matriz de bytes que no se ajusta a ningún estándar en particular, como, por ejemplo, los requisitos de una clave AES.
- 18: RC4. Las claves RC4 no son válidas en los HSM en modo FIPS
- 21: Triple DES (3DES). No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.
- 31: AES

Obligatorio: sí

-s

Especifica el tamaño de la clave en bytes. Por ejemplo, para crear una clave de 192 bits, escriba `24`.

Valores válidos para cada tipo de clave:

- AES: 16 (128 bits), 24 (192 bits), 32 (256 bits)
- 3DES: 24 (192 bits)
- Secreta genérica: <3584 (28672 bits)

Obligatorio: sí

-l

Especifica una etiqueta definida por el usuario para la clave. Tipo de cadena.

Puede utilizar cualquier frase que le ayude a identificar la clave. La etiqueta no tiene por qué ser única, por lo que puede usarla para agrupar y clasificar las claves.

Obligatorio: sí

-attest

Ejecuta una comprobación de integridad que verifica que el firmware en el que se ejecuta el clúster no haya sufrido alguna manipulación.

Predeterminado: sin comprobación de certificación.

Obligatorio: no

-id

Especifica un identificador definido por el usuario para la clave. Escriba una cadena que sea única en el clúster. El valor predeterminado es una cadena vacía.

Predeterminado: sin valor de ID.

Obligatorio: no

-min_srv

Especifica el número mínimo de HSM en los que la clave importada se sincroniza antes de que caduque el valor del parámetro `-timeout`. Si la clave no está sincronizada con el número especificado de servidores en el tiempo asignado, no se creará.

AWS CloudHSM sincroniza automáticamente todas las claves con todos los HSM del clúster. Para acelerar el proceso, establezca el valor de `min_srv` en un número menor que el de HSM del clúster y establezca un valor bajo de tiempo de espera. Sin embargo, tenga en cuenta que puede que algunas solicitudes no generen ninguna clave.

Predeterminado: 1

Obligatorio: no

-m_value

Especifica el número de usuarios que deben aprobar cualquier operación criptográfica que utilice la clave. Escriba un valor de `0` a `8`.

Este parámetro establece un requisito de autenticación de quórum para la clave. El valor predeterminado, `0`, deshabilita la característica de autenticación de cuórum para la clave. Cuando la autenticación de quórum está activada, el número especificado de usuarios debe firmar un token para aprobar las operaciones criptográficas que utilizan la clave, y las operaciones que comparten o dejan de compartir la clave.

Para encontrar la clave `m_value` de una clave, utilice. getKeyInfo

Este parámetro solo es válido cuando el parámetro `-u` del comando comparte la clave con suficientes usuarios para satisfacer el requisito `m_value`.

Predeterminado: 0

Obligatorio: no

-nex

Hace que la clave no se pueda extraer. La clave que se genera no se puede exportar desde el HSM.

Predeterminado: la clave se puede extraer.

Obligatorio: no

-sess

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use setAttribute.

Predeterminado: la clave es persistente.

Obligatorio: no

-timeout

Especifica cuánto tiempo (en segundos) espera el comando para que una clave se sincronice con el número de HSM especificado por el parámetro `min_srv`.

Este parámetro solo es válido cuando también se usa el parámetro `min_srv` en el comando.

Predeterminado: sin tiempo de espera predeterminado. El comando espera indefinidamente y solo vuelve a aparecer cuando la clave está sincronizada con el número mínimo de servidores.

Obligatorio: no

-u

Comparte la clave con los usuarios especificados. Este parámetro concede permiso a otros usuarios de criptografía (CU) de HSM para usar esta clave en operaciones criptográficas.

Escriba una lista separada por comas de los ID de usuario de HSM, como `-u 5,6`. No incluya el ID de usuario de HSM del usuario actual. Para buscar los ID de usuario de HSM de los CU del HSM, utilice listUsers. Para compartir o dejar de compartir una clave existente, utilice shareKey en cloudhsm_mgmt_util.

Predeterminado: solo el usuario actual puede utilizar la clave.

Obligatorio: no

Temas relacionados de

- exSymKey
- Género A KeyPair
- GenDSA KeyPair
- GeneCC KeyPair
- setAttribute

getAttribute

El getAttribute comando de key_mgmt_util escribe uno o todos los valores de atributo de una clave en un archivo. AWS CloudHSM Si el atributo que especifica no existe para el tipo de clave, como el módulo de una clave AES, getAttribute devuelve un error.

Los atributos de la clave son las propiedades de una clave. Contienen características, como el tipo de clave, clase, etiqueta e ID y los valores que representan las acciones que puede desempeñar con la clave, como cifrar, descifrar, encapsular, firmar y verificar.

Solamente puede utilizar getAttribute en claves que sean de su propiedad y que hayan compartido con usted. Puede ejecutar este comando o el comando getAttribute en cloudhsm_mgmt_util, que obtiene el valor de un atributo de una clave de todos los HSM de un clúster y lo escribe en stdout o en un archivo.

Para obtener una lista de los atributos y las constantes que los representan, ejecute el comando listAttributes. Para cambiar los valores de los atributos de las claves existentes, utilice setAttribute en key_mgmt_util y setAttribute en cloudhsm_mgmt_util. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
getAttribute -h
```

```
getAttribute -o <key handle>
             -a <attribute constant>
             -out <file>
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza getAttribute para obtener los atributos de las claves de los HSM.

Example : obtención del tipo de clave

En este ejemplo se obtiene el tipo de la clave, por ejemplo una AES, 3DES o una clave genérica, o una clave RSA o par de claves de curva elíptica.

El primer comando ejecuta listAttributes, que obtiene los atributos de una clave y las constantes que los representan. La salida muestra que la constante del tipo de clave es 256. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

```
Command: listAttributes

Description
===========
The following are all of the possible attribute values for getAttributes.

      OBJ_ATTR_CLASS                = 0
      OBJ_ATTR_TOKEN                = 1
      OBJ_ATTR_PRIVATE              = 2
      OBJ_ATTR_LABEL                = 3
      OBJ_ATTR_KEY_TYPE             = 256
      OBJ_ATTR_ID                   = 258
      OBJ_ATTR_SENSITIVE            = 259
      OBJ_ATTR_ENCRYPT              = 260
      OBJ_ATTR_DECRYPT              = 261
      OBJ_ATTR_WRAP                 = 262
      OBJ_ATTR_UNWRAP               = 263
      OBJ_ATTR_SIGN                 = 264
      OBJ_ATTR_VERIFY               = 266
      OBJ_ATTR_LOCAL                = 355
      OBJ_ATTR_MODULUS              = 288
      OBJ_ATTR_MODULUS_BITS         = 289
      OBJ_ATTR_PUBLIC_EXPONENT      = 290
      OBJ_ATTR_VALUE_LEN            = 353
```

```
      OBJ_ATTR_EXTRACTABLE            = 354
      OBJ_ATTR_KCV                    = 371
```

El segundo comando ejecuta getAttribute. Solicita el tipo de clave (atributo 256) del indicador de clave 524296 y lo escribe en el archivo `attribute.txt`.

```
Command: getAttribute -o 524296 -a 256 -out attribute.txt
Attributes dumped into attribute.txt file
```

El último comando obtiene el contenido del archivo de clave. La salida revela que el tipo de clave es `0x15` o 21, lo que es una clave triple DES (3DES). Para informarse de las definiciones de los valores de clase y tipo, consulte la sección de referencia de los atributos de clave.

```
$  cat attribute.txt
OBJ_ATTR_KEY_TYPE
0x00000015
```

Example : obtención de todos los atributos de una clave

Este comando obtiene todos los atributos de la clave que tiene el indicador 6 y los escribe en el archivo `attr_6`. Utiliza un valor de atributo de 512, que representa todos los atributos.

```
Command: getAttribute -o 6 -a 512 -out attr_6

got all attributes of size 444 attr cnt 17
Attributes dumped into attribute.txt file

        Cfm3GetAttribute returned: 0x00 : HSM Return: SUCCESS>
```

Este comando muestra el contenido de un archivo de atributo de muestra con todos los valores de atributo. En los valores indica que la clave es una clave AES de 256 bits con un ID de `test_01` y una etiqueta de `aes256`. La clave es extraíble y persistente, es decir, no es una clave de una única sesión. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

```
$  cat attribute.txt

OBJ_ATTR_CLASS
0x04
OBJ_ATTR_KEY_TYPE
```

```
0x15
OBJ_ATTR_TOKEN
0x01
OBJ_ATTR_PRIVATE
0x01
OBJ_ATTR_ENCRYPT
0x01
OBJ_ATTR_DECRYPT
0x01
OBJ_ATTR_WRAP
0x01
OBJ_ATTR_UNWRAP
0x01
OBJ_ATTR_SIGN
0x00
OBJ_ATTR_VERIFY
0x00
OBJ_ATTR_LOCAL
0x01
OBJ_ATTR_SENSITIVE
0x01
OBJ_ATTR_EXTRACTABLE
0x01
OBJ_ATTR_LABEL
aes256
OBJ_ATTR_ID
test_01
OBJ_ATTR_VALUE_LEN
0x00000020
OBJ_ATTR_KCV
0x1a4b31
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-o

Especifica el identificador de la clave de destino. Puede especificar una única clave en cada comando. Para obtener el identificador de una clave, use findKey.

Además, debe ser propietario de la clave especificada o esta debe compartirse con usted. Para buscar los usuarios de una clave, utilice. getKeyInfo

Obligatorio: sí

-a

Identifica el atributo. Escriba una constante que represente un atributo o `512`, que representa todos los atributos. Por ejemplo, para obtener el tipo de clave, escriba `256`, que es la constante del atributo `OBJ_ATTR_KEY_TYPE`.

Para generar una lista de los atributos y sus constantes, utilice listAttributes. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Obligatorio: sí

-out

Escribe la salida en el archivo especificado. Escriba una ruta de archivo. No puede escribir la salida en `stdout`.

Si el archivo especificado existe, getAttribute sobrescribe el archivo sin ningún tipo de advertencia.

Obligatorio: sí

Temas relacionados de

- getAttribute en cloudhsm_mgmt_util
- listAttributes
- setAttribute
- findKey
- Referencia de los atributos de claves

getCaviumPrivClave

El comando getCaviumPrivKey de key_mgmt_util exporta una clave privada de un HSM en un formato PEM falso. El archivo PEM falso, que no contiene el material de la clave privada sino que hace referencia a la clave privada en el HSM, se puede utilizar para establecer la descarga SSL/TLS

del servidor web a AWS CloudHSM. Para obtener más información, consulte Descarga de SSL/TLS en Linux.

Antes de ejecutar cualquier comando key_mgmt_util, debe iniciar key_mgmt_util e iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
getCaviumPrivKey -h

getCaviumPrivKey -k <private-key-handle
                 -out <fake-PEM-file>
```

Ejemplos

Este ejemplo muestra cómo utilizar getCaviumPrivKey para exportar una clave privada en formato PEM falso.

Example Exportar un archivo en formato PEM falso

Este comando crea y exporta una versión PEM falsa de una clave privada con el identificador 15 y la guarda en un archivo denominado `cavKey.pem`. Cuando el comando se ejecuta correctamente, exportPrivateKey devuelve un mensaje de confirmación.

```
Command: getCaviumPrivKey -k 15 -out cavKey.pem

Private Key Handle is written to cavKey.pem in fake PEM format

        getCaviumPrivKey returned: 0x00 : HSM Return: SUCCESS
```

Parámetros

Este comando admite los siguientes parámetros.

`-h`

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

-k

Especifica el identificador de clave de la clave privada que se va a exportar en formato PEM falso.

Obligatorio: sí

-out

Especifica el nombre del archivo en el que se escribirá la clave PEM falsa.

Obligatorio: sí

Temas relacionados de

- importPrivateKey
- Descarga de SSL/TLS en Linux

getCert

El comando getCert en key_mgmt_util recupera los certificados de partición de un HSM y los guarda en un archivo. Al ejecutar el comando, debe designar el tipo de certificado que desea recuperar. Para ello, utilice uno de los números enteros que se describen a continuación en la sección Parámetros. Para obtener más información sobre la función de cada uno de estos certificados, consulte Verificar la identidad del HSM.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
getCert -h

getCert -f <file-name>
        -t <certificate-type>
```

Ejemplo

En este ejemplo, se muestra cómo utilizar getCert para recuperar el certificado raíz del cliente de un clúster y guardarlo en un archivo.

Example : recuperación de un certificado raíz del cliente

Este comando exporta un certificado raíz del cliente (representado por el númeroentero 4) y lo guarda en un archivo denominado `userRoot.crt`. Cuando el comando se ejecuta correctamente, getCert devuelve un mensaje de confirmación.

```
Command: getCert -f userRoot.crt -s 4

Cfm3GetCert() returned 0 :HSM Return: SUCCESS
```

Parámetros

Este comando admite los siguientes parámetros.

`-h`

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

`-f`

Especifica el nombre del archivo en el que se guardará el certificado recuperado.

Obligatorio: sí

-s

Un número entero que especifica el tipo de certificado de partición que se desea recuperar. Los números enteros y sus correspondientes tipos de certificados son los siguientes:

- 1: certificado raíz del fabricante
- 2: certificado de hardware del fabricante
- 4: certificado raíz del cliente
- 8: certificado del clúster (firmado por el certificado raíz del cliente)
- 16: certificado del clúster (enlazado al certificado raíz del fabricante)

Obligatorio: sí

Temas relacionados de

- Verificar la identidad del HSM
- getCert (en cloudhsm_mgmt_util)

getKeyInfo

El comando getKeyInfo en la key_mgmt_util devuelve los ID de los usuarios de HSM que pueden utilizar la clave, incluidos el propietario y los usuarios de criptografía (CU) con quienes se comparte la clave. Cuando la autenticación de cuórum está habilitada en una clave, getKeyInfo también devuelve el número de usuarios que deben aprobar las operaciones criptográficas que utilizan la clave. Solamente puede ejecutar getKeyInfo en las claves que son de su propiedad y han compartido con usted.

Cuando ejecuta getKeyInfo en claves públicas, getKeyInfo solamente devuelve el propietario de la clave, aunque todos los usuarios del HSM puedan utilizar la clave pública. Para encontrar los ID de HSM de los usuarios en sus HSM, utilice listUsers. Para encontrar las claves de un usuario concreto, use findKey `-u`.

Es propietario de las claves que crea. Puede compartir una clave con otros usuarios cuando la crea. A continuación, para compartir o dejar de compartir una clave existente, utilice shareKey en cloudhsm_mgmt_util.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
getKeyInfo -h

getKeyInfo -k <key-handle>
```

Ejemplos

En estos ejemplos, se muestra cómo se utiliza getKeyInfo para obtener información sobre los usuarios de una clave.

Example : obtención de los usuarios para una clave simétrica

Este comando obtiene los usuarios que pueden utilizar la clave AES (simétrica) con identificador de clave 9. El resultado muestra que el usuario 3 es propietario de la clave y la comparte con el usuario 4.

```
Command:  getKeyInfo -k 9

       Cfm3GetKey returned: 0x00 : HSM Return: SUCCESS
```

```
       Owned by user 3

       also, shared to following 1 user(s):

                4
```

Example : obtención de los usuarios para un par de claves asimétricas

Estos comandos utilizan getKeyInfo para obtener los usuarios que pueden utilizar las claves de un par de claves RSA (simétricas). La clave pública tiene el identificador de clave 21. La clave privada tiene el identificador de clave 20.

Cuando ejecuta getKeyInfo en la clave privada (20), devuelve el propietario de la clave (3) y los usuarios de criptografía (CU) 4 y 5, con quienes se comparte la clave.

```
Command:  getKeyInfo -k 20

       Cfm3GetKey returned: 0x00 : HSM Return: SUCCESS

       Owned by user 3

       also, shared to following 2 user(s):

                4
                5
```

Cuando ejecuta getKeyInfo en la clave pública (21), solamente devuelve el propietario de la clave (3).

```
Command:  getKeyInfo -k 21

       Cfm3GetKey returned: 0x00 : HSM Return: SUCCESS

       Owned by user 3
```

Para confirmar que el usuario 4 puede utilizar la clave pública (y todas las claves públicas en el HSM), utilice el parámetro `-u` de findKey.

El resultado muestra que el usuario 4 puede utilizar la clave pública (21) y la clave privada (20) en el par de claves. El usuario 4 también puede utilizar todas las demás claves públicas y cualquier clave privada creadas o que se hayan compartido con ellos.

```
Command:  findKey -u 4
Total number of keys present 8

 number of keys matched from start index 0::7
11, 12, 262159, 262161, 262162, 19, 20, 21

        Cluster Error Status
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS

        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

Example : obtención del valor de autenticación de cuórum (m_value) para una clave

Este ejemplo muestra cómo obtener el `m_value` para una clave, esto es, el número de usuarios en el cuórum que debe aprobar las operaciones criptográficas que utilizan la clave.

Cuando la autenticación de cuórum está habilitada en una clave, un cuórum de usuarios debe aprobar cualquier operación criptográfica que utilice la clave. Para habilitar la autenticación de cuórum y establecer el tamaño de cuórum, utilice el parámetro `-m_value` al crear la clave.

Este comando usa GenRSA KeyPair para crear un key pair de claves RSA que se comparte con el usuario 4. Utiliza el parámetro `m_value` para habilitar la autenticación de cuórum en la clave privada en el par y establecer el tamaño de cuórum en dos usuarios. El número de usuarios debe ser lo suficientemente grande como para proporcionar las aprobaciones necesarias.

El resultado muestra que el comando creó la clave pública 27 y la clave privada 28.

```
Command:  genRSAKeyPair -m 2048 -e 195193 -l rsa_mofn -id rsa_mv2 -u 4 -m_value 2

        Cfm3GenerateKeyPair returned: 0x00 : HSM Return: SUCCESS

        Cfm3GenerateKeyPair:    public key handle: 27    private key handle: 28

        Cluster Error Status
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
```

Este comando utiliza getKeyInfo para obtener información sobre los usuarios de la clave privada. El resultado muestra que la clave es propiedad del usuario 3 y se comparte con el usuario 4. También muestra que un cuórum de dos usuarios debe aprobar todas las operaciones criptográficas que utilizan la clave.

```
Command:  getKeyInfo -k 28

        Cfm3GetKey returned: 0x00 : HSM Return: SUCCESS

        Owned by user 3

        also, shared to following 1 user(s):

                4
         2 Users need to approve to use/manage this key
```

Parámetros

-h

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

-k

Especifica el identificador de una clave en el HSM. Escriba el identificador de una clave de su propiedad o que comparte. Este parámetro es obligatorio.

Para buscar identificadores de clave, use el comando findKey.

Obligatorio: sí

Temas relacionados de

- getKeyInfoen cloudhsm_mgmt_util
- listUsers
- findKey
- findAllKeysen cloudhsm_mgmt_util

Help

El comando help en key_mgmt_util muestra información sobre todos los comandos disponibles de key_mgmt_util.

Antes de ejecutar el comando help, debe iniciar key_mgmt_util.

Sintaxis

```
help
```

Ejemplo

Este ejemplo muestra la salida del comando `help`.

Example

```
Command:  help

Help Commands Available:

Syntax: <command> -h


   Command               Description
   =======               ===========

   exit                   Exits this application
   help                   Displays this information

        Configuration and Admin Commands
   getHSMInfo             Gets the HSM Information
   getPartitionInfo       Gets the Partition Information
   listUsers              Lists all users of a partition
   loginStatus            Gets the Login Information
   loginHSM               Login to the HSM
   logoutHSM              Logout from the HSM

        M of N commands
   getToken               Initiate an MxN service and get Token
   delToken               delete Token(s)
   approveToken           Approves an MxN service
   listTokens             List all Tokens in the current partition

        Key Generation Commands

        Asymmetric Keys:
   genRSAKeyPair          Generates an RSA Key Pair
   genDSAKeyPair          Generates a DSA Key Pair
   genECCKeyPair          Generates an ECC Key Pair
```

```
     Symmetric Keys:
genPBEKey              Generates a PBE DES3 key
genSymKey              Generates a Symmetric keys

     Key Import/Export Commands
createPublicKey        Creates an RSA public key
importPubKey           Imports RSA/DSA/EC Public key
exportPubKey           Exports RSA/DSA/EC Public key
importPrivateKey       Imports RSA/DSA/EC private key
exportPrivateKey       Exports RSA/DSA/EC private key
imSymKey               Imports a Symmetric key
exSymKey               Exports a Symmetric key
wrapKey                Wraps a key from from HSM using the specified handle
unWrapKey              UnWraps a key into HSM using the specified handle

     Key Management Commands
deleteKey              Delete Key
setAttribute           Sets an attribute of an object
getKeyInfo             Get Key Info about shared users/sessions
findKey                Find Key
findSingleKey          Find single Key
getAttribute           Reads an attribute from an object

     Certificate Setup Commands
getCert                Gets Partition Certificates stored on HSM

     Key Transfer Commands
insertMaskedObject     Inserts a masked object
extractMaskedObject    Extracts a masked object

     Management Crypto Commands
sign                   Generates a signature
verify                 Verifies a signature
aesWrapUnwrap          Does NIST AES Wrap/Unwrap

     Helper Commands
Error2String           Converts Error codes to Strings
                       save key handle in fake PEM format
getCaviumPrivKey       Saves an RSA private key handle
                       in fake PEM format
IsValidKeyHandlefile   Checks if private key file has
                       an HSM key handle or a real key
listAttributes         List all attributes for getAttributes
```

```
   listECCCurveIds        List HSM supported ECC CurveIds
```

Parámetros

Este comando no tiene parámetros.

Temas relacionados de

- loginHSM y logoutHSM

importPrivateKey

El comando importPrivateKey de key_mgmt_util importa una clave privada asimétrica de un archivo a un HSM. El HSM no permite la importación directa de claves en texto no cifrado. El comando cifra la clave privada mediante una clave de encapsulamiento AES que usted especifique y desencapsula la clave dentro del HSM. Si intenta asociar una AWS CloudHSM clave a un certificado, consulte este tema.

> **Note**
>
> No puede importar una clave PEM protegida con contraseña mediante una clave simétrica o privada.

Debe especificar una clave de encapsulamiento AES que tenga un valor de atributo `OBJ_ATTR_UNWRAP` y `OBJ_ATTR_ENCRYPT` `1`. Para buscar los atributos de una clave, utilice el comando getAttribute.

> **Note**
>
> Este comando no ofrece la opción de marcar la clave importada como no exportable.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
importPrivateKey -h
```

```
importPrivateKey -l <label>
                 -f <key-file>
                 -w <wrapping-key-handle>
                 [-sess]
                 [-id <key-id>]
                 [-m_value <0...8>]
                 [min_srv <minimum-number-of-servers>]
                 [-timeout <number-of-seconds>]
                 [-u <user-ids>]
                 [-wk <wrapping-key-file>]
                 [-attest]
```

Ejemplos

Este ejemplo muestra cómo utilizar importPrivateKey para importar una clave privada en un HSM.

Example : importar una clave privada

Este comando importa la clave privada de un archivo denominado `rsa2048.key` con la etiqueta `rsa2048-imported` y una clave de encapsulación con el identificador `524299`. Cuando el comando se ejecuta correctamente, importPrivateKey devuelve un identificador de clave para la clave importada y un mensaje de confirmación.

```
Command: importPrivateKey -f rsa2048.key -l rsa2048-imported -w 524299

BER encoded key length is 1216

Cfm3WrapHostKey returned: 0x00 : HSM Return: SUCCESS

Cfm3CreateUnwrapTemplate returned: 0x00 : HSM Return: SUCCESS

Cfm3UnWrapKey returned: 0x00 : HSM Return: SUCCESS

Private Key Unwrapped.  Key Handle: 524301

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Parámetros

Este comando admite los siguientes parámetros.

`-h`

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

`-l`

Especifica la etiqueta de clave privada definida por el usuario.

Obligatorio: sí

`-f`

Especifica el nombre de archivo de la clave que se va a importar.

Obligatorio: sí

`-w`

Especifica el identificador de la clave de encapsulamiento. Este parámetro es obligatorio. Para buscar identificadores de clave, utilice el comando findKey.

Para determinar si una clave se puede utilizar como clave de encapsulación, utilice getAttribute para obtener el valor del atributo `OBJ_ATTR_WRAP` (262). Para crear una clave de encapsulación, utilice genSymKey para crear una clave AES (de tipo 31).

Si utiliza el parámetro `-wk` para especificar una clave de desencapsulación externa, la clave de encapsulación `-w` se utiliza para encapsular, pero no desencapsular, la clave durante la importación.

Obligatorio: sí

`-sess`

Especifica la clave importada como una clave de sesión.

Valor predeterminado: la clave importada se mantiene como clave persistente (token) en el clúster.

Obligatorio: no

`-id`

Especifica el ID de la clave que se va a importar.

Predeterminado: sin valor de ID.

Obligatorio: no

-m_value

Especifica el número de usuarios que deben aprobar cualquier operación criptográfica que utilice la clave importada. Especifique un valor entre **0** y **8**.

Este parámetro solo es válido cuando el parámetro `-u` del comando comparte la clave con suficientes usuarios para satisfacer el requisito `m_value`.

Predeterminado: 0

Obligatorio: no

-min_srv

Especifica el número mínimo de HSM en los que la clave importada se sincroniza antes de que caduque el valor del parámetro `-timeout`. Si la clave no está sincronizada con el número especificado de servidores en el tiempo asignado, no se creará.

AWS CloudHSM sincroniza automáticamente todas las claves con todos los HSM del clúster. Para acelerar el proceso, establezca el valor de `min_srv` en un número menor que el de HSM del clúster y establezca un valor bajo de tiempo de espera. Sin embargo, tenga en cuenta que puede que algunas solicitudes no generen ninguna clave.

Predeterminado: 1

Obligatorio: no

-timout

Especifica el número de segundos que se debe esperar hasta que la clave se sincronice entre los HSM cuando se incluye el parámetro `min-serv`. Si no se ningún número, el sondeo continúa indefinidamente.

Valor predeterminado: sin límite

Obligatorio: no

-u

Especifica la lista de usuarios con los que se va a compartir la clave privada importada. Este parámetro concede a otros usuarios de criptografía (CU) de HSM permiso para usar la clave importada en operaciones criptográficas.

Escriba una lista de identificadores de usuario de HSM separada por comas, como `-u 5,6`. No incluya el ID de usuario de HSM del usuario actual. Para buscar los ID de usuario de HSM de los CU del HSM, utilice listUsers.

Valor predeterminado: solo el usuario actual puede utilizar la clave importada.

Obligatorio: no

`-wk`

Especifica la clave que se utilizará para encapsular la clave que se está importando. Escriba la ruta y el nombre de un archivo que contenga una clave AES sin cifrar.

Si se incluye este parámetro, importPrivateKey utiliza la clave del archivo `-wk` para encapsular la clave que se va a importar. También utiliza la clave especificada por el parámetro `-w` para desencapsularla.

Valor predeterminado: utilizar la clave de encapsulación especificada en el parámetro `-w` para encapsular y desencapsular.

Obligatorio: no

`-attest`

Realiza una comprobación de conformidad de la respuesta del firmware para asegurarse de que el firmware en el que se ejecuta el clúster no ha sido manipulado.

Obligatorio: no

Temas relacionados de

- wrapKey
- unWrapKey
- genSymKey
- exportPrivateKey

importPubKey

El comando importPubKey de key_mgmt_util importa una clave pública con formato PEM en un HSM. Puede utilizarlo para importar claves públicas que se han generado fuera del HSM. También puede

usar el comando para importar claves que se exportaron desde un HSM, como las exportadas por el comando. exportPubKey

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
importPubKey -h

importPubKey -l <label>
             -f <key-file>
             [-sess]
             [-id <key-id>]
             [min_srv <minimum-number-of-servers>]
             [-timeout <number-of-seconds>]
```

Ejemplos

Este ejemplo muestra cómo utilizar importPubKey para importar una clave pública en un HSM.

Example : importar una clave pública

Este comando importa una clave pública desde un archivo denominado `public.pem` con la etiqueta `importedPublicKey`. Cuando el comando se ejecuta correctamente, importPubKey devuelve un identificador de clave para la clave importada y un mensaje de confirmación.

```
Command: importPubKey -l importedPublicKey -f public.pem

Cfm3CreatePublicKey returned: 0x00 : HSM Return: SUCCESS

Public Key Handle: 262230

        Cluster Error Status
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
```

Parámetros

Este comando admite los siguientes parámetros.

`-h`

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

`-l`

Especifica la etiqueta de clave pública definida por el usuario.

Obligatorio: sí

`-f`

Especifica el nombre de archivo de la clave que se va a importar.

Obligatorio: sí

`-sess`

Designa la clave importada como clave de sesión.

Valor predeterminado: la clave importada se mantiene como clave persistente (token) en el clúster.

Obligatorio: no

`-id`

Especifica el ID de la clave que se va a importar.

Predeterminado: sin valor de ID.

Obligatorio: no

`-min_srv`

Especifica el número mínimo de HSM en los que la clave importada se sincroniza antes de que el caduque valor del parámetro `-timeout`. Si la clave no está sincronizada con el número especificado de servidores en el tiempo asignado, no se creará.

AWS CloudHSM sincroniza automáticamente todas las claves con todos los HSM del clúster. Para acelerar el proceso, establezca el valor de `min_srv` en un número menor que el de HSM del clúster y establezca un valor bajo de tiempo de espera. Sin embargo, tenga en cuenta que puede que algunas solicitudes no generen ninguna clave.

Predeterminado: 1

Obligatorio: no

`-timeout`

Especifica el número de segundos que se debe esperar hasta que la clave se sincronice entre los HSM cuando se incluye el parámetro `min-serv`. Si no se ningún número, el sondeo continúa indefinidamente.

Valor predeterminado: sin límite

Obligatorio: no

Temas relacionados de

- exportPubKey
- Generar claves

imSymKey

El comando imSymKey de la herramienta key_mgmt_util importa una copia sin cifrar de una clave simétrica desde un archivo en el HSM. Puede usarla para importar las claves que genere mediante cualquier método ajeno al HSM y las claves que se exportaron desde un HSM, como las claves que el comando exSymKey, escribe en un archivo.

Durante el proceso de importación imSymKey utiliza la clave AES seleccionada (la clave de encapsulación) para encapsular (cifrar) y, a continuación desencapsular (descifrar) la clave que se va a importar. Sin embargo, imSymKey funciona solo en archivos que contienen claves no cifradas. Para exportar e importar claves cifradas, utilice WrapKey y unWrapKeylos comandos.

Además, el comando imSymKey importa únicamente claves simétricas. Para importar claves públicas, utilice importPubKey. Para importar claves privadas, utilice importPrivateKeyo WrapKey.

> **Note**
>
> No puede importar una clave PEM protegida con contraseña mediante una clave simétrica o privada.

Las claves importadas funcionan de forma muy parecida a las claves que se generan en el HSM. Sin embargo, el valor del atributo OBJ_ATTR_LOCAL es cero, lo que indica que no se generó

localmente. Puede utilizar el siguiente comando para compartir una clave simétrica al importarla. Puede utilizar el comando `shareKey` de cloudhsm_mgmt_util para compartir la clave después de importarla.

```
imSymKey -l aesShared -t 31 -f kms.key -w 3296 -u 5
```

Después de importar una clave, asegúrese de marcar o eliminar el archivo de clave. Este comando no le impide importar el mismo material de claves varias veces. El resultado, es decir, varias claves con diferentes identificadores de clave y el mismo material de claves, dificulta el seguimiento del uso del material relacionado con las claves e impide que supere sus límites criptográficos.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
imSymKey -h

imSymKey -f <key-file>
         -w <wrapping-key-handle>
         -t <key-type>
         -l <label>
         [-id <key-ID>]
         [-sess]
         [-wk <wrapping-key-file> ]
         [-attest]
         [-min_srv <minimum-number-of-servers>]
         [-timeout <number-of-seconds> ]
         [-u <user-ids>]
```

Ejemplos

En los ejemplos siguientes se muestra cómo utilizar imSymKey para importar claves simétricas a sus HSM.

Example : importar una clave simétrica AES

En este ejemplo se utiliza imSymKey para importar una clave simétrica AES a HSM.

El primer comando utiliza OpenSSL para generar una clave simétrica AES de 256 bits aleatoria. Guarda la clave en el archivo `aes256.key`.

```
$  openssl rand -out aes256-forImport.key 32
```

El segundo comando utiliza imSymKey para importar la clave AES desde el archivo `aes256.key` hasta los HSM. Utiliza la clave 20, una clave AES del HSM, como clave de encapsulación y especifica una etiqueta de tipo `imported`. A diferencia del ID, no es necesario que la etiqueta sea única en el clúster. El valor del parámetro `-t` (tipo) es `31`, que representa AES.

La salida muestra que la clave del archivo se encapsuló y se desencapsuló y después se importó al HSM, donde se asignó al identificador de clave 262180.

```
Command:  imSymKey -f aes256.key -w 20 -t 31 -l imported

        Cfm3WrapHostKey returned: 0x00 : HSM Return: SUCCESS

        Cfm3CreateUnwrapTemplate returned: 0x00 : HSM Return: SUCCESS

        Cfm3UnWrapKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Unwrapped.  Key Handle: 262180

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

El siguiente comando utiliza getAttribute para obtener el atributo OBJ_ATTR_LOCAL (atributo 355) de la clave que acaba de importar y lo escribe en el archivo `attr_262180`.

```
Command:  getAttribute -o 262180 -a 355 -out attributes/attr_262180
Attributes dumped into attributes/attr_262180_imported file

        Cfm3GetAttribute returned: 0x00 : HSM Return: SUCCESS
```

Cuando examine el archivo de atributos, verá que el valor del atributo `OBJ_ATTR_LOCAL` es cero, lo que indica que el material de clave no se ha generado en el HSM.

```
$  cat attributes/attr_262180_local
OBJ_ATTR_LOCAL
0x00000000
```

Example : desplazamiento de una clave simétrica entre clústeres

En este ejemplo se muestra cómo usar exSymKeyy imSymKey mover una clave AES de texto plano entre clústeres. Se puede utilizar un proceso como este para crear una encapsulación AES que exista en los HSM de ambos clústeres. Una vez que la clave de empaquetado compartida esté en su lugar, puede usar WrapKey y unWrapKeymover las claves cifradas entre los clústeres.

El usuario CU encargado de realizar esta operación deberá tener permiso para iniciar sesión en los HSM de ambos clústeres.

El primer comando se utiliza exSymKeypara exportar la clave 14, una clave AES de 32 bits, del clúster 1 al `aes.key` archivo. También utiliza la clave 6, una clave AES que está en los HSM del clúster 1, como clave de encapsulación.

```
Command: exSymKey -k 14 -w 6 -out aes.key

        Cfm3WrapKey returned: 0x00 : HSM Return: SUCCESS

        Cfm3UnWrapHostKey returned: 0x00 : HSM Return: SUCCESS


Wrapped Symmetric Key written to file "aes.key"
```

A continuación, el usuario inicia sesión en key_mgmt_util en el clúster 2 y ejecuta un comando imSymKey para importar la clave contenida en el archivo `aes.key` a los HSM del clúster 2. Este comando utiliza la clave 252152, una clave AES que está en los HSM del clúster 2, como clave de encapsulación.

Como las claves de empaquetado que exSymKeyimSymKeyutilizo empaquetan y desenvuelven inmediatamente las claves de destino, no es necesario que las claves de empaquetado de los distintos clústeres sean las mismas.

La salida muestra que la clave se ha importado correctamente al clúster 2 y que se le ha asignado el identificador de clave 21.

```
Command:  imSymKey -f aes.key -w 262152 -t 31 -l xcluster

        Cfm3WrapHostKey returned: 0x00 : HSM Return: SUCCESS

        Cfm3CreateUnwrapTemplate returned: 0x00 : HSM Return: SUCCESS
```

```
        Cfm3UnWrapKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Unwrapped.  Key Handle: 21

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Para demostrar que la clave 14 del clúster 1 y la clave 21 del clúster 2 tienen el mismo material de clave, obtenga el valor de comprobación clave (KCV) de cada clave. Si los valores de KCV son los mismos, el material de clave es el mismo.

El siguiente comando utiliza getAttribute en el clúster 1 para escribir el valor del atributo KCV (atributo 371) de la clave 14 en el archivo `attr_14_kcv`. A continuación, utiliza un comando cat para obtener el contenido del archivo `attr_14_kcv`.

```
Command:  getAttribute -o 14 -a 371 -out attr_14_kcv
Attributes dumped into attr_14_kcv file

$  cat attr_14_kcv
OBJ_ATTR_KCV
0xc33cbd
```

Este comando similar utiliza getAttribute en el clúster 2 para escribir el valor del atributo KCV (atributo 371) de la clave 21 en el archivo `attr_21_kcv`. A continuación, utiliza un comando cat para obtener el contenido del archivo `attr_21_kcv`.

```
Command:  getAttribute -o 21 -a 371 -out attr_21_kcv
Attributes dumped into attr_21_kcv file

$  cat attr_21_kcv
OBJ_ATTR_KCV
0xc33cbd
```

La salida muestra que los valores de KCV de las dos claves son los mismos, lo que demuestra que el material de clave es el mismo.

Dado que los HSM de ambos clústeres tienen el mismo material de clave, ahora puede compartir las claves cifradas entre los clústeres sin que se exponga la clave sin cifrar. Por ejemplo, puede utilizar el comando `wrapKey` con la clave de encapsulación 14 para exportar una clave cifrada desde el

clúster 1 y, a continuación, utilizar unWrapKey con la clave de encapsulación 21 para importar la clave cifrada al clúster 2.

Example : importación de una clave de sesión

Este comando utiliza los parámetros `-sess` de imSymKey para importar una clave triple DES de 192 bits que es válida únicamente en la sesión actual.

El comando utiliza el parámetro `-f` para especificar el archivo que contiene la clave para importar, el parámetro `-t` para especificar el tipo de clave y el parámetro `-w` para especificar la clave de encapsulación. También utiliza el parámetro `-l` para especificar una etiqueta que establece la categoría de la clave, el parámetro `-id` para crear un identificador descriptivo y único para la clave y el parámetro `-attest` para verificar el firmware que importa la clave.

La salida muestra que la clave se encapsuló y se desencapsuló correctamente, se importó al HSM y se le asignó el identificador de clave 37. Además, muestra que se pasó la comprobación de atestación, que indica que el firmware no ha sufrido ninguna manipulación.

```
Command:  imSymKey -f 3des192.key -w 6 -t 21 -l temp -id test01 -sess -attest

        Cfm3WrapHostKey returned: 0x00 : HSM Return: SUCCESS

        Cfm3CreateUnwrapTemplate returned: 0x00 : HSM Return: SUCCESS

        Cfm3UnWrapKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Unwrapped.  Key Handle: 37

        Attestation Check : [PASS]

        Cluster Error Status
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

A continuación, puede ejecutar los comandos getAttribute o findKey para verificar los atributos de la clave recién importada. El siguiente comando utiliza findKey para verificar que la clave 37 tenga el tipo, la etiqueta y el ID especificados por el comando y que sea una clave de sesión. Tal y como se muestra en la línea 5 de la salida, findKey informa que la única clave que coincide con todos los atributos es la clave 37.

```
Command:  findKey -t 21 -l temp -id test01 -sess 1
Total number of keys present 1
```

```
 number of keys matched from start index 0::0
37

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS

        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

Parámetros

-attest

Ejecuta una comprobación de integridad que verifica que el firmware en el que se ejecuta el clúster no haya sufrido alguna manipulación.

Predeterminado: sin comprobación de certificación.

Obligatorio: no

-f

Especifica el archivo que contiene la clave que se va a importar.

El archivo debe contener una copia sin cifrar de una clave AES o triple DES de la longitud especificada. Las claves RC4 y DES no son válidas en los HSM en modo FIPS.

- AES: 16, 24 o 32 bytes
- Triple DES (3DES): 24 bytes

Obligatorio: sí

-h

Muestra ayuda para el comando.

Obligatorio: sí

-id

Especifica un identificador definido por el usuario para la clave. Escriba una cadena que sea única en el clúster. El valor predeterminado es una cadena vacía.

Predeterminado: sin valor de ID.

Obligatorio: no

-l

Especifica una etiqueta definida por el usuario para la clave. Tipo de cadena.

Puede utilizar cualquier frase que le ayude a identificar la clave. La etiqueta no tiene por qué ser única, por lo que puede usarla para agrupar y clasificar las claves.

Obligatorio: sí

-min_srv

Especifica el número mínimo de HSM en los que la clave importada se sincroniza antes de que caduque el valor del parámetro `-timeout`. Si la clave no está sincronizada con el número especificado de servidores en el tiempo asignado, no se creará.

AWS CloudHSM sincroniza automáticamente todas las claves con todos los HSM del clúster. Para acelerar el proceso, establezca el valor de `min_srv` en un número menor que el de HSM del clúster y establezca un valor bajo de tiempo de espera. Sin embargo, tenga en cuenta que puede que algunas solicitudes no generen ninguna clave.

Predeterminado: 1

Obligatorio: no

-sess

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use setAttribute.

Predeterminado: la clave es persistente.

Obligatorio: no

-timeout

Especifica cuánto tiempo (en segundos) espera el comando para que una clave se sincronice con el número de HSM especificado por el parámetro `min_srv`.

Este parámetro solo es válido cuando también se usa el parámetro `min_srv` en el comando.

Predeterminado: sin tiempo de espera predeterminado. El comando espera indefinidamente y solo vuelve a aparecer cuando la clave está sincronizada con el número mínimo de servidores.

Obligatorio: no

-t

Especifica el tipo de clave simétrica. Escriba la constante que representa el tipo de clave. Por ejemplo, para crear una clave AES, escriba `-t 31`.

Valores válidos:

- 21: Triple DES (3DES).
- 31: AES

Obligatorio: sí

-u

Comparte la clave que va a importar con los usuarios especificados. Este parámetro concede permiso a otros usuarios de criptografía (CU) de HSM para usar esta clave en operaciones criptográficas.

Escriba un ID o una lista separada por comas de los ID de usuario de HSM, como -u `5,6`. No incluya el ID de usuario de HSM del usuario actual. Para encontrar el ID, puede utilizar el comando listUsers en la herramienta de la línea de comandos cloudhsm_mgmt_util o el comando listUsers de la herramienta de la línea de comandos key_mgmt_util.

Obligatorio: no

-w

Especifica el identificador de la clave de encapsulamiento. Este parámetro es obligatorio. Para buscar identificadores de clave, use el comando findKey.

Una clave de encapsulación es una clave del HSM que se utiliza para cifrar ("encapsular") y después descifrar ("desencapsular") la clave durante el proceso de importación. Solo las claves AES se pueden utilizar como claves de encapsulación.

Puede utilizar cualquier clave AES (de cualquier tamaño) como clave de encapsulación. Dado que la clave de encapsulación encapsula y, a continuación, desencapsula inmediatamente la

clave de destino, puede utilizar una clave AES solo de una sesión como clave de encapsulación. Para determinar si una clave se puede utilizar como clave de encapsulación, utilice getAttribute para obtener el valor del atributo `OBJ_ATTR_WRAP` (262). Para crear una clave de empaquetado, utilice esta opción genSymKeypara crear una clave AES (tipo 31).

Si utiliza el parámetro `-wk` para especificar una clave de encapsulación externa, la clave de encapsulación `-w` se utilizará para desencapsular, pero no encapsular, la clave que se va a importar.

> **Note**
>
> La clave 4 es una clave interna incompatible. Le recomendamos que use una clave AES que cree y administre como clave de encapsulamiento.

Obligatorio: sí

-wk

Utilice la clave AES del archivo especificado para encapsular la clave que se importa. Escriba la ruta y el nombre de un archivo que contenga una clave AES sin cifrar.

Si se incluye este parámetro, imSymKey utiliza la clave del archivo `-wk` para encapsular la clave que se va a importar y utiliza la clave del HSM especificado en el parámetro `-w` para desencapsularla. Los valores de parámetro `-w` y `-wk` deben resolverse en la misma clave sin cifrar.

Valor predeterminado: utilice la clave de encapsulación del HSM para realizar la desencapsulación.

Obligatorio: no

Temas relacionados de

- genSymKey
- exSymKey
- wrapKey
- unWrapKey
- exportPrivateKey

- exportPubKey

insertMaskedObject

El comando insertMaskedObject de key_mgmt_util inserta un objeto enmascarado desde un archivo en un HSM designado. Los objetos enmascarados son objetos clonados que se extraen de un HSM usando el comando extractMaskedObject. Solo se pueden utilizar después de insertarlos de nuevo en el clúster original. Solo puede insertar un objeto enmascarado en el mismo clúster desde el que se generó, o en un clon de ese clúster. Esto incluye cualquier versión clonada del clúster original generada al copiar una copia de seguridad entre regiones y utilizar la copia de seguridad para crear un clúster nuevo.

Los objetos enmascarados son una forma eficaz de descargar y sincronizar claves, incluidas las claves no extraíbles (es decir, las claves que tienen un valor de `OBJ_ATTR_EXTRACTABLE` igual a `0`). De esta forma, las claves se pueden sincronizar de forma segura en clústeres relacionados en diferentes regiones sin necesidad de actualizar el archivo de configuración. AWS CloudHSM

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
insertMaskedObject -h

insertMaskedObject -f <filename>
                   [-min_srv <minimum-number-of-servers>]
                   [-timeout <number-of-seconds>]
```

Ejemplos

Este ejemplo muestra cómo utilizar insertMaskedObject para insertar un archivo de objeto enmascarado en un HSM.

Example : insertar un objeto enmascarado

Este comando inserta un objeto enmascarado en un HSM desde un archivo denominado `maskedObj`. Cuando el comando se ejecuta correctamente, insertMaskedObject devuelve un identificador de clave para la clave descifrada del objeto enmascarado y un mensaje de confirmación.

```
Command: insertMaskedObject -f maskedObj
```

```
Cfm3InsertMaskedObject returned: 0x00 : HSM Return: SUCCESS
        New Key Handle: 262433

        Cluster Error Status
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
```

Parámetros

Este comando admite los siguientes parámetros.

`-h`

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

`-f`

Especifica el nombre de archivo del objeto enmascarado que se va a insertar.

Obligatorio: sí

`-min_srv`

Especifica el número mínimo de servidores en los que se sincroniza el objeto enmascarado insertado antes de que caduque el valor del parámetro `-timeout`. Si el objeto no se sincroniza con el número de servidores especificado en el tiempo asignado, no se insertará.

Predeterminado: 1

Obligatorio: no

`-timeout`

Especifica el número de segundos que se debe esperar para que la clave se sincronice entre los servidores cuando se incluye el parámetro `min-serv`. Si no se ningún número, el sondeo continúa indefinidamente.

Valor predeterminado: sin límite

Obligatorio: no

Temas relacionados de

- extractMaskedObject
- syncKey
- Copia de una copia de seguridad entre regiones
- Creación de un AWS CloudHSM clúster a partir de un Backup anterior

IsValidKeyHandlefile

El comando IsValidKeyHandlefile de key_mgmt_util se utiliza para averiguar si un archivo de clave contiene una clave privada verdadera o una clave RSA PEM falsa. Un archivo de clave PEM falso no contiene material de una clave privada real sino que hace referencia a la clave privada del HSM. Este tipo de archivo se puede utilizar para establecer descarga de SSL/TLS del servidor web a AWS CloudHSM. Para obtener más información, consulte Descarga de SSL/TLS en Linux.

> **Note**
>
> IsValidKeyHandlefile solo funciona para claves RSA.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
IsValidKeyHandlefile -h

IsValidKeyHandlefile -f <rsa-private-key-file>
```

Ejemplos

Estos ejemplos muestran cómo utilizar IsValidKeyHandlefile para determinar si un archivo de clave determinado contiene material de clave verdadero o material de clave PEM falso.

Example : validar una clave privada verdadera

Este comando confirma que el archivo `privateKey.pem` contiene material de clave verdadero.

```
Command: IsValidKeyHandlefile -f privateKey.pem

Input key file has real private key
```

Example : invalidar una clave PEM falsa

Este comando confirma que el archivo `caviumKey.pem` contiene material de clave PEM falso creado con el identificador de clave 15.

```
Command: IsValidKeyHandlefile -f caviumKey.pem

Input file has invalid key handle: 15
```

Parámetros

Este comando admite los siguientes parámetros.

`-h`

Muestra la ayuda de la línea de comando para el comando.

Obligatorio: sí

`-f`

Especifica el archivo de clave privada RSA que se comprobará para determinar si contiene material de clave válido.

Obligatorio: sí

Temas relacionados de

- getCaviumPrivClave
- Descarga de SSL/TLS en Linux

listAttributes

El listAttributes comando de key_mgmt_util muestra los atributos de una clave y las constantes que los representan. AWS CloudHSM Usted utiliza estas constantes para identificar los atributos en los

comandos getAttribute y setAttribute. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

Este comando no tiene parámetros.

```
listAttributes
```

Ejemplo

Este comando enumera los atributos de clave que se pueden obtener y cambiar en key_mgmt_util, así como las constantes que los representan. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Para representar a todos los atributos del comando getAttribute de key_mgmt_util, utilice 512.

```
Command: listAttributes

Following are the possible attribute values for getAttributes:

      OBJ_ATTR_CLASS                = 0
      OBJ_ATTR_TOKEN                = 1
      OBJ_ATTR_PRIVATE              = 2
      OBJ_ATTR_LABEL                = 3
      OBJ_ATTR_KEY_TYPE             = 256
      OBJ_ATTR_ENCRYPT              = 260
      OBJ_ATTR_DECRYPT              = 261
      OBJ_ATTR_WRAP                 = 262
      OBJ_ATTR_UNWRAP               = 263
      OBJ_ATTR_SIGN                 = 264
      OBJ_ATTR_VERIFY               = 266
      OBJ_ATTR_LOCAL                = 355
      OBJ_ATTR_MODULUS              = 288
      OBJ_ATTR_MODULUS_BITS         = 289
      OBJ_ATTR_PUBLIC_EXPONENT      = 290
      OBJ_ATTR_VALUE_LEN            = 353
      OBJ_ATTR_EXTRACTABLE          = 354
      OBJ_ATTR_KCV                  = 371
```

Temas relacionados de

- listAttributes en cloudhsm_mgmt_util
- getAttribute
- setAttribute
- Referencia de los atributos de claves

listUsers

El comando listUsers de key_mgmt_util obtiene los usuarios de los HSM, junto con el tipo de usuario y otros atributos.

En key_mgmt_util, listUsers devuelve un resultado que representa todos los HSM del clúster, aun cuando no sean coherentes. Para obtener información acerca de los usuarios de cada HSM, utilice el comando listUsers en cloudhsm_mgmt_util.

Los comandos de usuario de key_mgmt_util listUsers y getKeyInfoson comandos de solo lectura que los criptousuarios (CU) tienen permiso para ejecutar. Los demás comandos de administración de usuarios forman parte de cloudhsm_mgmt_util. Son ejecutados por responsables de criptografía (CO) que tienen permisos de administración de usuarios.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
listUsers

listUsers -h
```

Ejemplo

Este comando enumera los usuarios de HSM en el clúster y sus atributos. Puede usar el `User ID` atributo para identificar a los usuarios en otros comandos, como FindKey, GetAttribute y. getKeyInfo

```
Command:  listUsers

        Number Of Users found 4
```

```
        Index       User ID     User Type       User Name           MofnPubKey
LoginFailureCnt         2FA
        1               1       PCO             admin                   NO
   0                NO
        2               2       AU              app_user                NO
   0                NO
        3               3       CU              alice                   YES
   0                NO
        4               4       CU              bob                     NO
   0                NO
        5               5       CU              trent                   YES
   0                NO

        Cfm3ListUsers returned: 0x00 : HSM Return: SUCCESS
```

La salida contiene los siguientes atributos de usuario:

- ID de usuario: identifica al usuario de los comandos key_mgmt_util y cloudhsm_mgmt_util.
- User type (Tipo de usuario): determina las operaciones que el usuario puede realizar en el HSM.
- User Name (Nombre de usuario): muestra el nombre fácil de recordar definido por el usuario para el usuario.
- MofnPubKey: Indica si el usuario ha registrado un key pair para firmar los tokens de autenticación de quórum.
- LoginFailureCnt: Indica el número de veces que el usuario ha iniciado sesión sin éxito.
- 2FA: indica que el usuario ha activado la autenticación multifactor.

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

Temas relacionados de

- listUsers en cloudhsm_mgmt_util
- findKey
- getAttribute

- getKeyInfo

loginHSM y logoutHSM

Los comandos loginHSM y logoutHSM de key_mgmt_util permiten iniciar y cerrar sesión en los HSM de un clúster. Una vez que inicie sesión en los HSM, puede utilizar key_mgmt_util para realizar diversas operaciones de administración de claves, como la generación, sincronización y encapsulación de claves públicas y privadas.

Para poder ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util. Para administrar claves con key_mgmt_util, debe iniciar sesión en los HSM como usuario de criptografía (CU).

> **Note**
>
> Si se superan cinco intentos de inicio de sesión incorrectos, se bloquea la cuenta. Si creó el clúster antes de febrero de 2018, la cuenta se bloquea después de 20 intentos de inicio de sesión incorrectos. Para desbloquear la cuenta, un responsable de criptografía (CO) debe restablecer la contraseña mediante el comando changePswd en cloudhsm_mgmt_util.
> Si tiene más de un HSM en el clúster, es posible que puedan realizarse intentos adicionales de inicio de sesión incorrectos antes de que se bloquee la cuenta. Esto se debe a que el cliente CloudHSM equilibra la carga entre los diversos HSM. Por lo tanto, el intento de inicio de sesión no puede comenzar en el mismo HSM cada vez. Si va a probar esta funcionalidad, recomendamos que lo haga en un clúster con un solo HSM activo.

Sintaxis

```
loginHSM -h

loginHSM -u <user type>
         { -p | -hpswd } <password>
         -s <username>
```

Ejemplo

Este ejemplo muestra cómo iniciar y cerrar sesión en los HSM de un clúster con los comandos `loginHSM` y `logoutHSM`.

Example Iniciar sesión en los HSM

Este comando inicia sesión en los HSM como usuario de criptografía (CU) con el nombre de usuario `example_user` y la contraseña `aws`. El resultado muestra que ha iniciado sesión en todos los HSM del clúster.

```
Command:  loginHSM -u CU -s example_user -p aws

Cfm3LoginHSM returned: 0x00 : HSM Return: SUCCESS

Cluster Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Example : iniciar sesión con una contraseña oculta

Este comando es el mismo que en el ejemplo anterior, salvo que esta vez se especifica que el sistema debe ocultar la contraseña.

```
Command:  loginHSM -u CU -s example_user -hpswd
```

El sistema le solicitará su contraseña. Introduzca la contraseña. El sistema la ocultará, y el resultado mostrará que el comando se ha ejecutado correctamente y se ha conectado a los HSM.

```
Enter password:

Cfm3LoginHSM returned: 0x00 : HSM Return: SUCCESS

Cluster Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS

Command:
```

Example Cerrar sesión en los HSM

Este comando cierra sesión en los HSM. El resultado muestra que ha iniciado sesión en todos los HSM del clúster.

```
Command: logoutHSM
```

```
Cfm3LogoutHSM returned: 0x00 : HSM Return: SUCCESS

Cluster Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Parámetros

-h

Muestra ayuda para este comando.

-u

Especifica el tipo de usuario que inicia la sesión. Para poder utilizar key_mgmt_util, debe iniciar sesión como CU.

Obligatorio: sí

-s

Especifica el nombre del usuario que inicia la sesión.

Obligatorio: sí

{ -p | -hpswd }

Especifique la contraseña de inicio de sesión con `-p`. La contraseña aparece en texto no cifrado cuando se escribe. Para ocultar la contraseña, use el parámetro opcional `-hpswd` en lugar de `-p` y siga las instrucciones.

Obligatorio: sí

Temas relacionados de

- exit

setAttribute

El comando setAttribute de key_mgmt_util convierte una clave que es válida únicamente en la sesión actual en una clave persistente que existe hasta que la elimina. El comando realiza esta operación

cambiando el valor de atributo de token de la clave (`OBJ_ATTR_TOKEN`) de falso (`0`) a verdadero (`1`). Solo puede cambiar los atributos de claves de su propiedad.

También puede utilizar el comando setAttribute de cloudhsm_mgmt_util para cambiar la etiqueta, encapsular, desencapsular, cifrar y descifrar atributos.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
setAttribute -h

setAttribute -o <object handle>
             -a 1
```

Ejemplo

En este ejemplo se muestra cómo convertir una clave de sesión en una clave persistente.

El primer comando usa el `-sess` parámetro de genSymKeypara crear una clave AES de 192 bits que solo es válida en la sesión actual. La salida muestra que el indicador de la clave de la nueva sesión es `262154`.

```
Command: genSymKey -t 31 -s 24 -l tmpAES -sess

        Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS

        Symmetric Key Created.  Key Handle: 262154

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
```

Este comando utiliza findKey para encontrar las claves de sesión de la sesión actual. La salida verifica que la clave `262154` es una clave de sesión.

```
Command:  findKey -sess 1

Total number of keys present 1
```

```
 number of keys matched from start index 0::0
262154

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS

        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

Este comando utiliza setAttribute para convertir la clave de sesión 262154 en una clave persistente. Para ello, cambia el valor del atributo de token (OBJ_ATTR_TOKEN) de la clave, de 0 (falso) a 1 (verdadero). Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

El comando utiliza el parámetro -o para especificar el indicador de clave (262154) y el parámetro -a para especificar la constante que representa el atributo de token (1). Cuando ejecuta el comando, este le solicita un valor para el atributo de token. El único valor válido es 1 (verdadero), el valor de una clave persistente.

```
Command: setAttribute -o 262154 -a 1
         This attribute is defined as a boolean value.
          Enter the boolean attribute value (0 or 1):1

        Cfm3SetAttribute returned: 0x00 : HSM Return: SUCCESS

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Para confirmar que la clave 262154 es ahora persistente, este comando busca claves de sesión (-sess 1) y claves persistentes (-sess 0) utilizando findKey. Esta vez el comando no encuentra claves de sesión, pero devuelve 262154 en la lista de claves persistentes.

```
Command: findKey -sess 1

Total number of keys present 0

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

```
        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS



Command: findKey -sess 0

Total number of keys present 5

 number of keys matched from start index 0::4
6, 7, 524296, 9, 262154

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS

        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-o

Especifica el identificador de la clave de destino. Puede especificar una única clave en cada comando. Para obtener el identificador de una clave, use findKey.

Obligatorio: sí

-a

Especifica la constante que representa el atributo que desea cambiar. El único valor válido es 1, que representa el atributo de token `OBJ_ATTR_TOKEN`.

Para obtener los atributos y sus valores enteros, utilice listAttributes.

Obligatorio: sí

Temas relacionados de

- setAttribute en cloudhsm_mgmt_util

- getAttribute
- listAttributes
- Referencia de los atributos de claves

sign

El comando sign de key_mgmt_uti utiliza la clave privada elegida para generar una firma para un archivo.

Para poder utilizar sign, primero debe tener una clave privada en el HSM. Puede generar una clave privada con los comandos genSymKey, genRSAKeyPair o genECCKeyPair. También puede importarla con el comando importPrivateKey. Para obtener más información, consulte Generar claves.

El comando sign utiliza un mecanismo de firma designado por el usuario, representado por un número entero, para firmar un archivo de mensaje. Para obtener una lista de posibles mecanismos de firma, consulte Parámetros.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
sign -h

sign -f <file name>
     -k <private key handle>
     -m <signature mechanism>
     -out <signed file name>
```

Ejemplo

Este ejemplo muestra cómo utilizar sign para firmar un archivo.

Example Firmar un archivo

Este comando firma un archivo denominado `messageFile` con una clave privada con el identificador `266309`. Utiliza el mecanismo de firma `SHA256_RSA_PKCS` (1) y guarda el archivo firmado resultante como `signedFile`.

```
Command: sign -f messageFile -k 266309 -m 1 -out signedFile

Cfm3Sign returned: 0x00 : HSM Return: SUCCESS

signature is written to file signedFile

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Parámetros

Este comando admite los siguientes parámetros.

`-f`

El nombre del archivo que se va a firmar.

Obligatorio: sí

`-k`

El identificador de la clave privada que se va a utilizar para la firma.

Obligatorio: sí

`-m`

Un número entero que representa el mecanismo de firma que se va a utilizar para la firma. Los mecanismos posibles se corresponden con los siguientes valores enteros:

Mecanismo de firma	Número entero correspondiente
SHA1_RSA_PKCS	0
SHA256_RSA_PKCS	1
SHA384_RSA_PKCS	2
SHA512_RSA_PKCS	3
SHA224_RSA_PKCS	4

Mecanismo de firma	Número entero correspondiente
SHA1_RSA_PKCS_PSS	5
SHA256_RSA_PKCS_PSS	6
SHA384_RSA_PKCS_PSS	7
SHA512_RSA_PKCS_PSS	8
SHA224_RSA_PKCS_PSS	9
ECDSA_SHA1	15
ECDSA_SHA224	16
ECDSA_SHA256	17
ECDSA_SHA384	18
ECDSA_SHA512	19

Obligatorio: sí

-out

El nombre del archivo en el que se va a guardar el archivo firmado.

Obligatorio: sí

Temas relacionados de

- verify
- importPrivateKey
- Género A KeyPair
- GeneCC KeyPair
- genSymKey
- Generar claves

unWrapKey

El comando unWrapKey de la herramienta key_mgmt_util importa una clave privada o simétrica encapsulada (cifrada) de un archivo en el HSM. Está diseñado para importar claves de cifrado encapsuladas con el comando wrapKey de key_mgmt_util, pero también se puede utilizar para desencapsular claves encapsuladas con otras herramientas. Sin embargo, en esas situaciones, le recomendamos que utilice las bibliotecas de software de PKCS #11 o JCE para desencapsular la clave.

Las claves importadas funcionan igual que las generadas por. AWS CloudHSM Sin embargo, el valor del atributo OBJ_ATTR_LOCAL es cero, lo que indica que no se han generado localmente.

Después de importar una clave, asegúrese de marcar o eliminar el archivo de claves. Este comando no le impide importar el mismo material de claves varias veces. Los resultados, es decir, varias claves con diferentes identificadores de clave y el mismo material de claves, dificulta el seguimiento del uso del material relacionado con las claves e impide que supere sus límites criptográficos.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
unWrapKey -h

unWrapKey -f <key-file-name>
          -w <wrapping-key-handle>
          [-sess]
          [-min_srv <minimum-number-of-HSMs>]
          [-timeout <number-of-seconds>]
          [-aad <additional authenticated data filename>]
          [-tag_size <tag size>]
          [-iv_file <IV file>]
          [-attest]
          [-m <wrapping-mechanism>]
          [-t <hash-type>]
          [-nex]
          [-u <user id list>]
          [-m_value <number of users needed for approval>]
          [-noheader]
          [-l <key-label>]
          [-id <key-id>]
          [-kt <key-type>]
```

```
        [-kc <key-class]
        [-i <unwrapping-IV>]
```

Ejemplo

Estos ejemplos muestran cómo utilizar unWrapKey para importar una clave encapsulada desde un archivo en los HSM. En el primer ejemplo, desencapsulamos una clave que se ha encapsulado con el comando key_mgmt_util de wrapKey y que, por lo tanto, tiene un encabezado. En el segundo ejemplo, desencapsulamos una clave que se ha encapsulado fuera de key_mgmt_util y que, por lo tanto, no tiene encabezado.

Example : desencapsulamiento de una clave (con encabezado)

Este comando importa una copia de una clave simétrica 3DES en un HSM. La clave se desencapsula con una clave AES con la etiqueta 6, que es criptográficamente idéntica a la que se utilizó para encapsular la clave 3DES. El resultado muestra que la clave del archivo se ha desencapsulado e importado y que el identificador de la clave importada es 29.

```
        Command:  unWrapKey -f 3DES.key -w 6 -m 4

        Cfm3UnWrapKey returned: 0x00 : HSM Return: SUCCESS

        Key Unwrapped.  Key Handle: 29

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Example : desencapsulamiento de una clave (sin encabezado)

Este comando importa una copia de una clave simétrica 3DES en un HSM. La clave se desencapsula con una clave AES con la etiqueta 6, que es criptográficamente idéntica a la que se utilizó para encapsular la clave 3DES. Dado que esta clave 3DES no se encapsuló con key_mgmt_util, se especifica el parámetro `noheader`, junto con sus parámetros de acompañamiento necesarios: una etiqueta de clave (`unwrapped3DES`), una clase de clave (4) y un tipo de clave (21). El resultado muestra que la clave del archivo se ha desencapsulado e importado y que el identificador de la clave importada es 8.

```
Command: unWrapKey -f 3DES.key -w 6 -noheader -l unwrapped3DES -kc 4 -kt 21 -m 4
```

```
      Cfm3CreateUnwrapTemplate2 returned: 0x00 : HSM Return: SUCCESS
       Cfm2UnWrapWithTemplate3 returned: 0x00 : HSM Return: SUCCESS

       Key Unwrapped. Key Handle: 8

       Cluster Error Status
       Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
       Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-f

Especifica la ruta y el nombre del archivo que contiene la clave encapsulada.

Obligatorio: sí

-w

Especifica la clave de encapsulamiento. Introduzca el identificador de clave de una clave AES o RSA en el HSM. Este parámetro es obligatorio. Para buscar identificadores de clave, use el comando findKey.

Para crear una clave de empaquetado, utilice genSymKeypara generar una clave AES (tipo 31) o GenRSA KeyPair para generar un par de claves RSA (tipo 0). Si utiliza un par de claves RSA, asegúrese de encapsular la clave con una de las claves y desencapsularla con la otra. Para verificar si una clave se puede utilizar como clave de encapsulamiento, utilice getAttribute para obtener el valor del atributo `OBJ_ATTR_WRAP`, que se representa con la constante `262`.

Obligatorio: sí

-sess

Crea una clave que solo existe en la sesión actual. La clave no se podrá recuperar una vez finalizada la sesión.

Utilice este parámetro cuando necesite una clave solo brevemente, por ejemplo, una clave de encapsulamiento que cifre y, a continuación, descifre rápidamente otra clave. No utilice una clave de sesión para cifrar los datos que pueda necesitar descifrar una vez finalizada la sesión.

Para cambiar una clave de sesión por una clave persistente (token), use setAttribute.

Predeterminado: la clave es persistente.

Obligatorio: no

-min_srv

Especifica el número mínimo de HSM en los que la clave importada se sincroniza antes de que caduque el valor del parámetro `-timeout`. Si la clave no está sincronizada con el número especificado de servidores en el tiempo asignado, no se creará.

AWS CloudHSM sincroniza automáticamente todas las claves con todos los HSM del clúster. Para acelerar el proceso, establezca el valor de `min_srv` en un número menor que el de HSM del clúster y establezca un valor bajo de tiempo de espera. Sin embargo, tenga en cuenta que puede que algunas solicitudes no generen ninguna clave.

Predeterminado: 1

Obligatorio: no

-timeout

Especifica cuánto tiempo (en segundos) espera el comando para que una clave se sincronice con el número de HSM especificado por el parámetro `min_srv`.

Este parámetro solo es válido cuando también se usa el parámetro `min_srv` en el comando.

Predeterminado: sin tiempo de espera predeterminado. El comando espera indefinidamente y solo vuelve a aparecer cuando la clave está sincronizada con el número mínimo de servidores.

Obligatorio: no

-attest

Ejecuta una comprobación de integridad que verifica que el firmware en el que se ejecuta el clúster no haya sufrido alguna manipulación.

Predeterminado: sin comprobación de certificación.

Obligatorio: no

-nex

Hace que la clave no se pueda extraer. La clave que se genera no se puede exportar desde el HSM.

Predeterminado: la clave se puede extraer.

Obligatorio: no

-m

Valor que representa el mecanismo de encapsulamiento. CloudHSM admite los siguientes mecanismos:

Mecanismo	Valor
`AES_KEY_WRAP_PAD_PKCS5`	4
`NIST_AES_WRAP_NO_PAD`	5
`NIST_AES_WRAP_PAD`	6
`RSA_AES`	7
`RSA_OAEP` (para conocer el tamaño máximo de los datos, consulte la nota que se incluye más adelante en esta sección)	8
`AES_GCM`	10
`CLOUDHSM_AES_GCM`	11
`RSA_PKCS` (para conocer el tamaño máximo de los datos, consulte la nota que se incluye más adelante en esta sección). Consulte la nota 1 que aparece a continuación para ver los próximos cambios.	12

Obligatorio: sí

> **Note**
>
> Cuando se utiliza el mecanismo de `RSA_OAEP` empaquetado, el tamaño máximo de clave que se puede empaquetar viene determinado por el módulo de la clave RSA y

la longitud del hash especificado de la siguiente manera: Tamaño máximo de clave = modulusLengthIn bytes- (hashLengthIn2* bytes) -2.
Cuando se utiliza el mecanismo de empaquetado RSA_PKCS, el tamaño máximo de clave que se puede empaquetar viene determinado por el módulo de la clave RSA de la siguiente manera: Tamaño máximo de clave = (bytes -11). modulusLengthIn

-t

Algoritmo hash	Valor
SHA1	2
SHA256	3
SHA384	4
SHA512	5
SHA224 (válido para los mecanismos RSA_AES y RSA_OAEP)	6

Obligatorio: no

-noheader

Si va a desencapsular una clave que se ha encapsulado fuera de key_mgmt_util, debe especificar este parámetro y todos los demás parámetros asociados.

Obligatorio: no

> **Note**
>
> Si especifica este parámetro, también debe especificar los siguientes parámetros -noheader:

- -l

  Especifica la etiqueta que se van a añadir a la clave desencapsulada.

Obligatorio: sí

- -kc

  Especifica la clase de la clave que se va a desencapsular. A continuación, se muestran los valores aceptables:

  3 = clave privada de un par de claves pública y privada

  4 = clave secreta (simétrica).

  Obligatorio: sí

- -kt

  Especifica el tipo de clave que se va a desencapsular. A continuación, se muestran los valores aceptables:

  0 = `RSA`

  1 = `DSA`

  3 = `ECC`

  16 = `GENERIC_SECRET`

  21 = `DES3`

  31 = `AES`

  Obligatorio: sí

Si lo desea, también puede especificar los siguientes parámetros `-noheader`:

- -id

  El ID que se van a añadir a la clave desencapsulada.

  Obligatorio: no

- -i

  El vector de inicialización (IV) de desencapsulación que se va a utilizar.

  Obligatorio: no

[1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Temas relacionados de

- wrapKey
- exSymKey
- imSymKey

verificar

El comando verify de key_mgmt_util confirma si un archivo está firmado con una clave determinada. Para ello, el comando verify compara un archivo firmado con un archivo de origen y analiza si están relacionados criptográficamente en función de una clave pública y un mecanismo de firma determinados. Se puede iniciar sesión en los archivos con la operación. AWS CloudHSM sign

Los mecanismos de firma están representados por los números enteros indicados en la sección de parámetros.

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
verify -h

verify -f <message-file>
       -s <signature-file>
       -k <public-key-handle>
       -m <signature-mechanism>
```

Ejemplo

Estos ejemplos muestran cómo utilizar verify para comprobar si se ha utilizado una clave pública determinada para firmar un archivo concreto.

Example : verificación de la firma de un archivo

Este comando intenta verificar si un archivo denominado `hardwarCert.crt` se firmó con la clave pública `262276` mediante el mecanismo de firma `SHA256_RSA_PKCS` para producir el archivo

firmado `hardwareCertSigned`. Dado que los parámetros especificados representan una relación de firma verdadera, el comando devuelve un mensaje de confirmación.

```
Command: verify -f hardwareCert.crt -s hardwareCertSigned -k 262276 -m 1

Signature verification successful

Cfm3Verify returned: 0x00 : HSM Return: SUCCESS
```

Example : demostración de una relación de firma falsa

Este comando verifica si un archivo denominado `hardwareCert.crt` se firmó con la clave pública `262276` mediante el mecanismo de firma `SHA256_RSA_PKCS` para producir el archivo firmado `userCertSigned`. Dado que los parámetros especificados no constituyen una relación de firma verdadera, el comando devuelve un mensaje de error.

```
Command: verify -f hardwarecert.crt -s usercertsigned -k 262276 -m 1
Cfm3Verify returned: 0x1b

CSP Error: ERR_BAD_PKCS_DATA
```

Parámetros

Este comando admite los siguientes parámetros.

`-f`

El nombre del archivo de mensaje de origen.

Obligatorio: sí

`-s`

El nombre del archivo firmado.

Obligatorio: sí

`-k`

El identificador de la clave pública que se cree que se utilizó para firmar el archivo.

Obligatorio: sí

-m

Un número entero que representa el mecanismo de firma que se supone que se utilizó para firmar el archivo. Los mecanismos posibles se corresponden con los siguientes valores enteros:

Mecanismo de firma	Número entero correspondiente
SHA1_RSA_PKCS	0
SHA256_RSA_PKCS	1
SHA384_RSA_PKCS	2
SHA512_RSA_PKCS	3
SHA224_RSA_PKCS	4
SHA1_RSA_PKCS_PSS	5
SHA256_RSA_PKCS_PSS	6
SHA384_RSA_PKCS_PSS	7
SHA512_RSA_PKCS_PSS	8
SHA224_RSA_PKCS_PSS	9
ECDSA_SHA1	15
ECDSA_SHA224	16
ECDSA_SHA256	17
ECDSA_SHA384	18
ECDSA_SHA512	19

Obligatorio: sí

Temas relacionados de

- sign
- getCert
- Generar claves

wrapKey

El comando wrapKey de key_mgmt_util exporta una copia cifrada de una clave simétrica o privada desde el HSM a un archivo. Cuando ejecute wrapKey, debe especificar la clave que va a exportar, una clave del HSM para cifrar (encapsular) la clave que se va a exportar y el archivo de salida.

El comando `wrapKey` escribe la clave cifrada en el archivo especificado, pero no la elimina del HSM ni le impide utilizarla en operaciones criptográficas. Puede exportar la misma clave varias veces.

Solo el propietario de una clave, es decir, el usuario de criptografía (CU) que creó la clave, puede exportarla. Los usuarios que comparten la clave pueden utilizarla en operaciones criptográficas, pero no pueden exportarla.

Para volver a importar la clave cifrada al HSM, utilice. unWrapKey Para exportar una clave de texto sin formato desde un HSM, utilice exSymKeyo exportPrivateKeysegún proceda. El aesWrapUnwrapcomando no puede descifrar (separar) las claves que cifran. wrapKey

Antes de ejecutar cualquier comando de key_mgmt_util, debe iniciar key_mgmt_util e Iniciar sesión en el HSM como usuario de criptografía (CU).

Sintaxis

```
wrapKey -h

wrapKey -k <exported-key-handle>
        -w <wrapping-key-handle>
        -out <output-file>
        [-m <wrapping-mechanism>]
        [-aad <additional authenticated data filename>]
        [-t <hash-type>]
        [-noheader]
        [-i <wrapping IV>]
        [-iv_file <IV file>]
        [-tag_size <num_tag_bytes>>]
```

Ejemplo

Example

Este comando exporta una clave simétrica Triple DES (3DES) de 192 bits (identificador de clave 7). Utiliza una clave AES de 256 bits en el HSM (identificador de clave 14) para encapsular la clave 7. A continuación, escribe la clave 3DES cifrada en el archivo `3DES-encrypted.key`.

La salida muestra que la clave 7 (la clave 3DES) se ha encapsulado correctamente y que se ha escrito en el archivo especificado. La clave cifrada tiene 307 bytes de largo.

```
        Command:  wrapKey -k 7 -w 14 -out 3DES-encrypted.key -m 4

        Key Wrapped.

        Wrapped Key written to file "3DES-encrypted.key length 307

        Cfm2WrapKey returned: 0x00 : HSM Return: SUCCESS
```

Parámetros

-h

Muestra ayuda para el comando.

Obligatorio: sí

-k

Identificador de la clave que desea exportar. Escriba el identificador de una clave simétrica o privada de su propiedad. Para buscar identificadores de clave, use el comando findKey.

Para verificar que se puede exportar una clave, ejecute el comando getAttribute para obtener el valor del atributo `OBJ_ATTR_EXTRACTABLE`, que se representa con la constante `354`. Para obtener ayuda para interpretar los atributos de clave, consulte la Referencia de los atributos de claves.

Además, únicamente puede exportar las claves que son de su propiedad. Para encontrar el propietario de una clave, utilice el comando. getKeyInfo

Obligatorio: sí

-w

Especifica la clave de encapsulamiento. Introduzca el identificador de clave de una clave AES o RSA en el HSM. Este parámetro es obligatorio. Para buscar identificadores de clave, use el comando findKey.

Para crear una clave de empaquetado, utilice genSymKeypara generar una clave AES (tipo 31) o GenRSA KeyPair para generar un par de claves RSA (tipo 0). Si utiliza un par de claves RSA, asegúrese de encapsular la clave con una de las claves y desencapsularla con la otra. Para verificar si una clave se puede utilizar como clave de encapsulamiento, utilice getAttribute para obtener el valor del atributo `OBJ_ATTR_WRAP`, que se representa con la constante `262`.

Obligatorio: sí

-out

Ruta y nombre del archivo de salida. Cuando el comando se ejecuta correctamente, este archivo contiene una copia cifrada de la clave exportada. Si el archivo ya existe, el comando lo sobrescribe sin ningún tipo de advertencia.

Obligatorio: sí

-m

Valor que representa el mecanismo de encapsulamiento. CloudHSM admite los siguientes mecanismos:

Mecanismo	Valor
`AES_KEY_WRAP_PAD_PKCS5`	4
`NIST_AES_WRAP_NO_PAD`	5
`NIST_AES_WRAP_PAD`	6
`RSA_AES`	7
`RSA_OAEP` (para conocer el tamaño máximo de los datos, consulte la nota que se incluye más adelante en esta sección)	8

Mecanismo	Valor
AES_GCM	10
CLOUDHSM_AES_GCM	11
RSA_PKCS (para conocer el tamaño máximo de los datos, consulte la nota que se incluye más adelante en esta sección). Consulte la nota 1 que aparece a continuación para ver los próximos cambios.	12

Obligatorio: sí

> **Note**
>
> Cuando se utiliza el mecanismo de RSA_0AEP empaquetado, el tamaño máximo de clave que se puede empaquetar viene determinado por el módulo de la clave RSA y la longitud del hash especificado de la siguiente manera: Tamaño máximo de clave = (bytes-2* bytes-2). modulusLengthIn hashLengthIn
> Cuando se utiliza el mecanismo de empaquetado RSA_PKCS, el tamaño máximo de clave que se puede empaquetar viene determinado por el módulo de la clave RSA de la siguiente manera: Tamaño máximo de clave = (bytes -11). modulusLengthIn

-t

Valor que representa el algoritmo hash. CloudHSM admite los siguientes algoritmos:

Algoritmo hash	Valor
SHA1	2
SHA256	3
SHA384	4
SHA512	5

Algoritmo hash	Valor
SHA224 (válido para los mecanismos RSA_AES y RSA_OAEP)	6

Obligatorio: no

-aad

Nombre del archivo que contiene AAD.

> **Note**
>
> Válido solo para los mecanismos AES_GCM y CLOUDHSM_AES_GCM.

Obligatorio: no

-noheader

Omite el encabezado que especifica los atributos de clave específicos de CloudHSM. Utilice este parámetro solo si tiene previsto desencapsular la clave con herramientas distintas de key_mgmt_util.

Obligatorio: no

-i

Vector de inicialización (IV) (valor hexadecimal).

> **Note**
>
> Solo es válido cuando se pasa con el parámetro -noheader de los mecanismos CLOUDHSM_AES_KEY_WRAP y NIST_AES_WRAP.

Obligatorio: no

-iv_file

Archivo en el que va a escribir el valor del IV obtenido como respuesta.

> **Note**
>
> Solo es válido cuando se pasa con el parámetro `-noheader` del mecanismo `AES_GCM`.

Obligatorio: no

-tag_size

Tamaño de la etiqueta que se va a guardar junto con el blob encapsulado.

> **Note**
>
> Solo es válido cuando se pasa con el parámetro `-noheader` de los mecanismos `AES_GCM` y `CLOUDHSM_AES_GCM`. El tamaño mínimo de la etiqueta es ocho.

Obligatorio: no

[1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Temas relacionados de

- exSymKey
- imSymKey
- unWrapKey

Referencia de los atributos de claves

Los comandos key_mgmt_util utilizan constantes para representar los atributos de claves en un HSM. Este tema puede ayudarle a identificar los atributos, encontrar las constantes que los representan en comandos y comprender sus valores.

Usted configura los atributos de una clave cuando la crea. Para cambiar el atributo del token, que indica si una clave es persistente o solo existe en la sesión, utilice el comando setAttribute de key_mgmt_util. Para cambiar los atributos de etiquetado, encapsulado, desencapsulamiento, cifrado y descifrado, utilice el comando`setAttribute` de cloudhsm_mgmt_util.

Para obtener una lista de los atributos y sus constantes, utilice listAttributes. Para obtener los valores de atributo de una clave, utilice getAttribute.

En la siguiente tabla se enumeran los atributos clave, sus constantes y sus valores válidos.

Atributo	Constant	Valores
OBJ_ATTR_ALL	512	Representa todos los atributos .
OBJ_ATTR_ALWAYS_SENSITIVE	357	0: falso. 1: verdadero.
OBJ_ATTR_CLASS	0	2: clave pública de un par de claves público-privada. 3: clave privada de un par de claves público-privada. 4: clave secreta (simétrica).
OBJ_ATTR_DECRYPT	261	0: falso. 1: verdadero. La clave se puede utilizar para descifrar datos.
OBJ_ATTR_DERIVE	268	0: falso. 1: verdadero. La función deriva la clave.
OBJ_ATTR_DESTROYABLE	370	0: falso. 1: verdadero.
OBJ_ATTR_ENCRYPT	260	0: falso. 1: verdadero. La clave se puede utilizar para cifrar datos.

Atributo	Constant	Valores
OBJ_ATTR_EXTRACTABLE	354	0: falso. 1: verdadero. Es posible exportar la clave desde los HSM.
OBJ_ATTR_ID	258	Cadena definida por el usuario. Debe ser única en el clúster. El valor predeterminado es una cadena vacía.
OBJ_ATTR_KCV	371	Valor de comprobación de clave de la clave. Para obtener más información, consulte Detalles adicionales.
OBJ_ATTR_KEY_TYPE	256	0: RSA. 1: DSA. 3: EC. 16: secreto genérico. 18: RC4. 21: Triple DES (3DES). 31: AES.
OBJ_ATTR_LABEL	3	Cadena definida por el usuario. No tiene que ser única en el clúster.
OBJ_ATTR_LOCAL	355	0. False. La clave se importó a los HSM. 1: verdadero.

Atributo	Constant	Valores
OBJ_ATTR_MODULUS	288	El módulo que se utilizó para generar un par de claves RSA. Para las claves EC, este valor representa la codificación DER del valor ECpoint ANSI X9.62 "Q" en formato hexadecimal. Para otros tipos de clave, este atributo no existe.
OBJ_ATTR_MODULUS_BITS	289	La longitud del módulo que se utilizó para generar un par de claves RSA. En el caso de las claves EC, representa el ID de la curva elíptica utilizada para generar la clave. Para otros tipos de clave, este atributo no existe.
OBJ_ATTR_NEVER_EXTRACTABLE	356	0: falso. 1: verdadero. No es posible exportar la clave desde los HSM.
OBJ_ATTR_PUBLIC_EXPONENT	290	El exponente público que se utilizó para generar un par de claves RSA. Para otros tipos de clave, este atributo no existe.

Atributo	Constant	Valores
OBJ_ATTR_PRIVATE	2	0: falso. 1: verdadero. Este atributo indica si los usuarios sin autenticar pueden enumerar los atributos de la clave. Dado que el proveedor PKCS#11 de CloudHSM no admite actualmente las sesiones públicas, todas las claves (incluidas las claves públicas de un par de clave pública-privada) tienen este atributo establecido en 1.
OBJ_ATTR_SENSITIVE	259	0: falso. Clave pública de un par de claves público-privadas. 1: verdadero.
OBJ_ATTR_SIGN	264	0: falso. 1: verdadero. La clave se puede utilizar para firmar (claves privadas).
OBJ_ATTR_TOKEN	1	0: falso. Clave de sesión. 1: verdadero. Clave persistente.
OBJ_ATTR_TRUSTED	134	0: falso. 1: verdadero.

Atributo	Constant	Valores
OBJ_ATTR_UNWRAP	263	0: falso. 1: verdadero. La clave se puede utilizar para descifrar claves.
OBJ_ATTR_UNWRAP_TEMPLATE	1073742354	Los valores deben usar la plantilla de atributo aplicada a cualquier clave desencapsulada mediante esta clave de encapsulamiento.
OBJ_ATTR_VALUE_LEN	353	Longitud de la clave en bytes.
OBJ_ATTR_VERIFY	266	0: falso. 1: verdadero. La clave se puede utilizar para verificación (claves públicas).
OBJ_ATTR_WRAP	262	0: falso. 1: verdadero. La clave se puede utilizar para cifrar claves.
OBJ_ATTR_WRAP_TEMPLATE	1073742353	Los valores deben usar la plantilla de atributo para coincidir con la clave encapsulada usando esta clave de encapsulamiento.
OBJ_ATTR_WRAP_WITH_TRUSTED	528	0: falso. 1: verdadero.

Detalles adicionales

Valor de comprobación de claves (KCV)

El valor de comprobación de claves (KCV) es un hash o suma de comprobación de 3 bytes de una clave que se genera cuando el HSM importa o genera una clave. También puede calcular un KCV fuera del HSM, por ejemplo, después de exportar una clave. A continuación, puede comparar los valores del KCV para confirmar la identidad e integridad de la clave. Para obtener el KCV (valor de control de la clave), utilice getAttribute.

AWS CloudHSM utiliza el siguiente método estándar para generar un valor de comprobación clave:

- Claves simétricas: los primeros 3 bytes del resultado obtenido al cifrar un bloque cero con la clave.
- Pares de claves asimétricas: los primeros 3 bytes del hash SHA-1 de la clave pública.
- Claves HMAC: por el momento, no se admite el uso del KCV con claves HMAC.

AWS CloudHSM SDK de cliente

Utilice un SDK de cliente para transferir las operaciones criptográficas de las aplicaciones basadas en plataformas o lenguajes a los módulos de seguridad de hardware (HSM).

AWS CloudHSM ofrece dos versiones principales y Client SDK 5 es la más reciente. Ofrece diversas ventajas con respecto a la versión SDK 3 de cliente (la serie anterior). Para obtener más información, consulte Ventajas del SDK 5 de cliente. Para obtener información acerca de las plataformas admitidas, consulte Plataformas compatibles con SDK 5 de cliente.

Para obtener información acerca del uso del SDK 3 de cliente, consulte SDK de cliente anterior (SDK de cliente 3).

the section called “Biblioteca PKCS #11”

El PKCS #11 es un estándar para realizar operaciones criptográficas en módulos de seguridad de hardware (HSM). AWS CloudHSM ofrece implementaciones de la biblioteca PKCS #11 que son compatibles con la versión 2.40 del PKCS #11.

the section called “Motor dinámico de OpenSSL”

El motor dinámico de AWS CloudHSM OpenSSL le permite transferir las operaciones criptográficas a su clúster de CloudHSM a través de la API de OpenSSL.

the section called “Proveedor de JCE”

El proveedor de AWS CloudHSM JCE cumple con la arquitectura criptográfica de Java (JCA). El proveedor le permite realizar operaciones criptográficas en el HSM.

the section called “Proveedores de KSP y CNG”

El AWS CloudHSM cliente para Windows incluye proveedores de GNC y KSP. Actualmente, solo SDK 3 de cliente es compatible con los proveedores de GNC y KSP.

Plataformas compatibles con SDK 5 de cliente

El soporte básico es diferente para cada versión del SDK AWS CloudHSM de cliente. El soporte de plataforma para los componentes de un SDK suele coincidir con el soporte básico, pero no siempre es así. Para determinar el soporte de plataforma de un componente concreto, primero debe asegurarse de que la plataforma que desea esté incluida en la sección básica del SDK y, a

continuación, comprobar si existen exclusiones o cualquier otra información pertinente en la sección de componentes.

AWS CloudHSM solo admite sistemas operativos de 64 bits.

El soporte de plataforma varía con el tiempo. Es posible que las versiones anteriores del SDK de cliente de CloudHSM no sean compatibles con todos los sistemas operativos enumerados aquí. Utilice las notas de la versión para determinar la compatibilidad del sistema operativo con las versiones anteriores del SDK de cliente de CloudHSM. Para obtener más información, consulte Descargas para AWS CloudHSM Client SDK.

Para ver las plataformas compatibles con el anterior SDK de cliente, consulte Plataformas compatibles con SDK 3 de cliente

Client SDK 5 no requiere un daemon de cliente.

Compatibilidad de Linux con SDK 5 de cliente

Plataformas admitidas	Arquitectura X86_64	Arquitectura ARM
Amazon Linux 2	Sí	Sí
Amazon Linux 2023	Sí	Sí
CentOS 7 (7,8+)	Sí	No
Red Hat Enterprise Linux 7 (7.8+)	Sí	No
Red Hat Enterprise Linux 8 (8.3 o superior)	Sí	No
Red Hat Enterprise Linux 9 (9.2+)	Sí	Sí
Ubuntu 20.04 LTS	Sí	No
Ubuntu 22.04 LTS	Sí	Sí

Nota: El SDK 5.4.2 fue la última versión que proporcionó soporte para la plataforma CentOS 8. Para obtener más información, consulte el sitio web de CentOS.

Compatibilidad de Windows con SDK 5 de cliente

- Microsoft Windows Server 2016
- Microsoft Windows Server 2019

Compatibilidad sin servidor para SDK 5 de cliente

- AWS Lambda
- Docker/ECS

Compatibilidad con componentes

Biblioteca PKCS #11

La biblioteca PKCS #11 es un componente multiplataforma compatible con el soporte básico de SDK 5 de cliente para Linux y Windows. Para obtener más información, consulte the section called “Compatibilidad de Linux con SDK 5 de cliente” y the section called “Compatibilidad de Windows con SDK 5 de cliente”.

Motor dinámico de OpenSSL

El motor dinámico de OpenSSL es un componente exclusivo de Linux que requiere OpenSSL 1.0.2, 1.1.1 o 3.x.

Proveedor de JCE

El proveedor de JCE es un SDK de Java que es compatible con OpenJDK 8, OpenJDK 11, OpenJDK 17 y OpenJDK 21 en todas las plataformas compatibles.

Ventajas del SDK 5 de cliente

En comparación con SDK 3 de cliente, SDK 5 de cliente es más fácil de gestionar, más fiable y ofrece una capacidad de configuración superior. SDK 5 de cliente también ofrece algunas ventajas clave con las que no cuenta SDK 3 de cliente.

Diseñado para arquitecturas sin servidor

SDK 5 de cliente no requiere un daemon de cliente, por lo que ya no es necesario gestionar un servicio en segundo plano. Esto supone una ayuda para los usuarios en distintos aspectos:

- Simplifica el proceso de inicio de la aplicación. Lo único que necesita para empezar a usar CloudHSM es configurar el SDK antes de ejecutar su aplicación.
- No es necesario contar con un proceso en ejecución constante, lo que facilita la integración con componentes sin servidor como Lambda y Elastic Container Service (ECS).

Mejores integraciones de terceros y portabilidad simplificada

SDK 5 de cliente se ciñe estrictamente a las especificaciones de JCE y proporciona una portabilidad más sencilla entre distintos proveedores de JCE y una mejor integración con terceros.

Experiencia de usuario y capacidad de configuración mejoradas

SDK 5 de cliente mejora la legibilidad de los mensajes de registro y proporciona mecanismos de gestión de errores y excepciones más claros, lo que facilita en gran medida la clasificación de autoservicio a los usuarios. SDK 5 también ofrece distintas configuraciones que puede consultar en la Página de configuración de la herramienta.

Compatibilidad ampliada con plataformas

SDK 5 de cliente ofrece más compatibilidad con las plataformas operativas modernas. Incluye soporte para tecnologías ARM y mayor compatibilidad con JCE, PKCS#11 y OpenSSL. Para obtener más información, consulte Plataformas admitidas.

Características y mecanismos adicionales

SDK 5 de cliente incluye características y mecanismos adicionales que no están disponibles en SDK 3 de cliente. Además, SDK 5 de cliente seguirá añadiendo más mecanismos en el futuro.

Migración del SDK 3 de cliente al SDK 5 de cliente

Para obtener instrucciones detalladas sobre la migración del SDK de cliente 3 al SDK de cliente 5, consulte las instrucciones de migración de cada SDK de cliente individual:

- Migre su biblioteca PKCS #11 del SDK de cliente 3 al SDK de cliente 5
- Migre su motor dinámico OpenSSL del Client SDK 3 al Client SDK 5
- Migre su proveedor de JCE de Client SDK 3 a Client SDK 5
- Migre de Client SDK 3 CMU y KMU a Client SDK 5 CloudHSM CLI

Para conocer las funciones o los casos de uso que no son compatibles con la CLI de CloudHSM, póngase en contacto con el servicio de asistencia.

> **Note**
>
> La biblioteca PKCS #11 del SDK 5 de Client ahora es compatible con las plataformas Windows. Puede gestionar la mayoría de los casos de uso que los proveedores de GNC y KSP pueden y deben considerar como sustitutos. Actualmente, KSP solo está disponible en Client SDK 3.

Biblioteca PKCS #11

El PKCS #11 es un estándar para realizar operaciones criptográficas en módulos de seguridad de hardware (HSM). AWS CloudHSM ofrece implementaciones de la biblioteca PKCS #11 que son compatibles con la versión 2.40 del PKCS #11.

Para obtener más información sobre el arranque, consulte Conexión al clúster. Problemas conocidos de la biblioteca PKCS #11Para solucionar problemas, consulte.

Para obtener información acerca del uso del SDK 3 de cliente, consulte SDK de cliente anterior (SDK de cliente 3).

Temas

- Instalación de SDK 5 de cliente para la biblioteca PKCS #11
- Biblioteca PKCS #11
- Tipos de claves compatibles
- Mecanismos admitidos
- Operaciones de la API compatibles
- Uso de atributos clave admitidos
- Ejemplos de código para la biblioteca PKCS #11

- Migre su biblioteca PKCS #11 del SDK de cliente 3 al SDK de cliente 5
- Configuraciones avanzadas para PKCS #11

Instalación de SDK 5 de cliente para la biblioteca PKCS #11

En este tema se proporcionan instrucciones para instalar la versión más reciente de la biblioteca PKCS #11 de la serie de versiones de SDK 5 de cliente. Para obtener más información sobre el SDK de cliente o la biblioteca PKCS #11, consulte Usar el SDK de cliente y Biblioteca de PKCS #11.

Instalación

Con SDK 5 de cliente, no es necesario instalar ni ejecutar un daemon de cliente.

Para ejecutar un único clúster de HSM con SDK 5 de cliente, primero debe administrar la configuración de durabilidad de la clave del cliente configurando `disable_key_availability_check` en `True`. Para obtener más información, consulte Sincronización de claves y Herramienta de configuración de SDK 5 de cliente.

Para obtener más información sobre la biblioteca PKCS #11 en SDK 5 de cliente, consulte Biblioteca PKCS #11.

> **Note**
>
> Para ejecutar un único clúster de HSM con SDK 5 de cliente, primero debe administrar la configuración de durabilidad de la clave del cliente configurando `disable_key_availability_check` en `True`. Para obtener más información, consulte Sincronización de claves y Herramienta de configuración de SDK 5 de cliente.

Cómo instalar y configurar la biblioteca PKCS #11

1. Utilice los comandos siguientes para descargar e instalar la biblioteca PKCS #11.

   Amazon Linux 2

   Instale la biblioteca PKCS #11 para Amazon Linux 2 en arquitectura X86_64:

   ```
   $ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-pkcs11-latest.el7.x86_64.rpm
   ```

```
$ sudo yum install ./cloudhsm-pkcs11-latest.el7.x86_64.rpm
```

Instale la biblioteca PKCS #11 para Amazon Linux 2 en arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-pkcs11-latest.el7.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-pkcs11-latest.el7.aarch64.rpm
```

Amazon Linux 2023

Instale la biblioteca PKCS #11 para Amazon Linux 2023 en la arquitectura X86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/cloudhsm-pkcs11-latest.amzn2023.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-pkcs11-latest.amzn2023.x86_64.rpm
```

Instale la biblioteca PKCS #11 para Amazon Linux 2023 en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/cloudhsm-pkcs11-latest.amzn2023.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-pkcs11-latest.amzn2023.aarch64.rpm
```

CentOS 7 (7.8+)

Instale la biblioteca PKCS #11 para CentOS 7.8+ en arquitectura X86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-pkcs11-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-pkcs11-latest.el7.x86_64.rpm
```

RHEL 7 (7.8+)

Instale la biblioteca PKCS #11 para RHEL 7 en la arquitectura X86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-pkcs11-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-pkcs11-latest.el7.x86_64.rpm
```

RHEL 8 (8.3+)

Instale la biblioteca PKCS #11 para RHEL 8 en la arquitectura X86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-pkcs11-latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-pkcs11-latest.el8.x86_64.rpm
```

RHEL 9 (9.2+)

Instale la biblioteca PKCS #11 para RHEL 9 en la arquitectura X86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-pkcs11-latest.el9.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-pkcs11-latest.el9.x86_64.rpm
```

Instale la biblioteca PKCS #11 para RHEL 9 en una arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-pkcs11-latest.el9.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-pkcs11-latest.el9.aarch64.rpm
```

Ubuntu 20.04 LTS

Instale la biblioteca PKCS #11 para Ubuntu 20.04 LTS en arquitectura X86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Focal/
cloudhsm-pkcs11_latest_u20.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-pkcs11_latest_u20.04_amd64.deb
```

Ubuntu 22.04 LTS

Instale la biblioteca PKCS #11 para Ubuntu 22.04 LTS en arquitectura X86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/
cloudhsm-pkcs11_latest_u22.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-pkcs11_latest_u22.04_amd64.deb
```

Instale la biblioteca PKCS #11 para Ubuntu 22.04 LTS en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/
cloudhsm-pkcs11_latest_u22.04_arm64.deb
```

```
$ sudo apt install ./cloudhsm-pkcs11_latest_u22.04_arm64.deb
```

Windows Server 2016

Instale la biblioteca PKCS #11 para Windows Server 2016 en arquitectura X86_64:

1. Descargue la biblioteca PKCS #11 para SDK 5 de cliente.
2. Ejecute el instalador de la biblioteca PKCS #11 (AWSCloudHSMPKCS11-latest.msi) con privilegios administrativos de Windows.

Windows Server 2019

Instale la biblioteca PKCS #11 para Windows Server 2019 en arquitectura X86_64:

1. Descargue la biblioteca PKCS #11 para SDK 5 de cliente.
2. Ejecute el instalador de la biblioteca PKCS #11 (AWSCloudHSMPKCS11-latest.msi) con privilegios administrativos de Windows.

2. Utilice la herramienta de configuración para especificar la ubicación del certificado de emisión. Para ver instrucciones, consulte Especifique la ubicación del certificado de emisión..
3. Para conectarse a su clúster, consulte Proceso de arranque del SDK de cliente.
4. Puede encontrar los archivos de la biblioteca PKCS #11 en las siguientes ubicaciones:

   - Binarios, scripts de configuración y archivos de registro de Linux:

     ```
     /opt/cloudhsm
     ```

     Binarios de Windows:

     ```
     C:\ProgramFiles\Amazon\CloudHSM
     ```

     Archivos de registro y scripts de configuración de Windows:

     ```
     C:\ProgramData\Amazon\CloudHSM
     ```

Biblioteca PKCS #11

Cuando se utiliza la biblioteca PKCS #11, su aplicación se ejecuta como un determinado usuario de criptografía (CU) en los HSM. La aplicación solo puede ver y administrar las claves que posee y comparte el CU. Puede utilizar un CU existente en sus HSM o crear un nuevo CU para su aplicación. Para obtener más información sobre la administración de CU, consulte Administración de usuarios de HSM con la CLI de CloudHSM y Administración de usuarios de HSM con la utilidad de administración de CloudHSM (CMU).

Para especificar el CU para la biblioteca PKCS #11, utilice el parámetro de pin de la función C_Login de PKCS #11. Para AWS CloudHSM, el parámetro pin tiene el siguiente formato:

```
<CU_user_name>:<password>
```

Por ejemplo, el siguiente comando establece el pin de la biblioteca PKCS #11 para el CU con el nombre de usuario `CryptoUser` y la contraseña `CUPassword123!`.

```
CryptoUser:CUPassword123!
```

Tipos de claves compatibles

La biblioteca PKCS #11 admite los siguientes tipos de clave.

Tipo de clave	Descripción
AES	Genere claves AES de 128, 192 y 256 bits.
Triple DES (3DES, DESede)	Genere una clave DES triple de 192 bits. Consulte la nota 1 que aparece a continuación para ver los próximos cambios.
EC	Genere claves con las curvas secp224r1 (P-224), secp256r1 (P-256), secp256k1 (Blockchain), secp384r1 (P-384) y secp521r1 (P-521).
GENERIC_SECRET	Genere secretos genéricos de 1 a 800 bytes.
RSA	Genere claves RSA de 2048 a 4096 bits, en incrementos de 256 bits

[1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Mecanismos admitidos

La biblioteca PKCS #11 es compatible con la versión 2.40 de la especificación PKCS #11. Para invocar una característica criptográfica con PKCS #11, llame a una función con un mecanismo determinado. En la siguiente tabla, se resumen las combinaciones de funciones y mecanismos admitidos por AWS CloudHSM.

La biblioteca PKCS #11 admite los siguientes algoritmos:

- Cifrado y descifrado: AES-CBC, AES-CTR, AES-ECB, AES-GCM, DES3-CBC, DES3-ECB, RSA-OAEP y RSA-PKCS
- Firma y verificación: RSA, HMAC y ECDSA; con y sin hash

- Hash/digest: SHA1, SHA224, SHA256, SHA384 y SHA512
- Encapsulación de claves: encapsulación de claves AES[1], AES-GCM, RSA-AES y RSA-OAEP

Temas

- Generación de funciones de claves y pares de claves
- Firma y comprobación de las funciones
- Funciones de recuperación de firma y recuperación de verificación
- Funciones Digest
- Funciones de cifrado y descifrado
- Derivación de funciones de claves
- Funciones de encapsulado y desencapsulado
- Tamaño máximo de datos para cada mecanismo
- Notas del mecanismo

Generación de funciones de claves y pares de claves

La biblioteca de AWS CloudHSM software de la biblioteca PKCS #11 permite utilizar los siguientes mecanismos para las funciones de generación de claves y pares de claves.

- `CKM_RSA_PKCS_KEY_PAIR_GEN`
- `CKM_RSA_X9_31_KEY_PAIR_GEN`Este mecanismo es funcionalmente idéntico al mecanismo `CKM_RSA_PKCS_KEY_PAIR_GEN`, pero ofrece más garantías en la generación de p y q.
- `CKM_EC_KEY_PAIR_GEN`
- `CKM_GENERIC_SECRET_KEY_GEN`
- `CKM_AES_KEY_GEN`
- `CKM_DES3_KEY_GEN` próximo cambio en la nota a pie de página 5.

Firma y comprobación de las funciones

La biblioteca de AWS CloudHSM software de la biblioteca PKCS #11 permite utilizar los siguientes mecanismos para las funciones de firma y verificación. Con SDK 5 de cliente, los datos se codifican localmente en el software. Esto significa que no hay límite en cuanto al tamaño de los datos que el SDK puede codificar.

Con SDK 5 de cliente, el hash de RSA y ECDSA se realiza de forma local, por lo que no hay límite de datos. Con HMAC hay límite de datos. Consulte la nota a pie de página 2 para obtener más información.

RSA

- `CKM_RSA_X_509`
- `CKM_RSA_PKCS` Únicamente para operaciones de una sola parte.
- `CKM_RSA_PKCS_PSS` Únicamente para operaciones de una sola parte.
- `CKM_SHA1_RSA_PKCS`
- `CKM_SHA224_RSA_PKCS`
- `CKM_SHA256_RSA_PKCS`
- `CKM_SHA384_RSA_PKCS`
- `CKM_SHA512_RSA_PKCS`
- `CKM_SHA512_RSA_PKCS`
- `CKM_SHA1_RSA_PKCS_PSS`
- `CKM_SHA224_RSA_PKCS_PSS`
- `CKM_SHA256_RSA_PKCS_PSS`
- `CKM_SHA384_RSA_PKCS_PSS`
- `CKM_SHA512_RSA_PKCS_PSS`

ECDSA

- `CKM_ECDSA` Únicamente para operaciones de una sola parte.
- `CKM_ECDSA_SHA1`
- `CKM_ECDSA_SHA224`
- `CKM_ECDSA_SHA256`
- `CKM_ECDSA_SHA384`
- `CKM_ECDSA_SHA512`

HMAC

- `CKM_SHA_1_HMAC`[2]

- CKM_SHA224_HMAC[2]
- CKM_SHA256_HMAC[2]
- CKM_SHA384_HMAC[2]
- CKM_SHA512_HMAC[2]

CMAC

- CKM_AES_CMAC

Funciones de recuperación de firma y recuperación de verificación

SDK 5 de cliente no ofrece funciones de recuperación de firma y recuperación de verificación.

Funciones Digest

La biblioteca de AWS CloudHSM software de la biblioteca PKCS #11 permite utilizar los siguientes mecanismos para las funciones de resumen. Con SDK 5 de cliente, los datos se codifican localmente en el software. Esto significa que no hay límite en cuanto al tamaño de los datos que el SDK puede codificar.

- CKM_SHA_1
- CKM_SHA224
- CKM_SHA256
- CKM_SHA384
- CKM_SHA512

Funciones de cifrado y descifrado

La biblioteca de AWS CloudHSM software de la biblioteca PKCS #11 le permite utilizar los siguientes mecanismos para las funciones de cifrado y descifrado.

- CKM_RSA_X_509
- CKM_RSA_PKCS Únicamente para operaciones de una sola parte Próximo cambio en la nota a pie de página 5.
- CKM_RSA_PKCS_OAEP Únicamente para operaciones de una sola parte.

- CKM_AES_ECB
- CKM_AES_CTR
- CKM_AES_CBC
- CKM_AES_CBC_PAD
- CKM_DES3_CBC próximo cambio en la nota a pie de página 5.
- CKM_DES3_ECB próximo cambio en la nota a pie de página 5.
- CKM_DES3_CBC_PAD próximo cambio en la nota a pie de página 5.
- CKM_AES_GCM [1, 2]
- CKM_CLOUDHSM_AES_GCM[3]

Derivación de funciones de claves

La biblioteca AWS CloudHSM de software de la biblioteca PKCS #11 permite utilizar los siguientes mecanismos para las funciones de Derive.

- CKM_SP800_108_COUNTER_KDF

Funciones de encapsulado y desencapsulado

La biblioteca de AWS CloudHSM software de la biblioteca PKCS #11 le permite utilizar los siguientes mecanismos para las funciones Wrap y Unwrap.

Para obtener más opciones de encapsulamiento de claves AES, consulte Encapsulamiento de claves con AES.

- CKM_RSA_PKCS Únicamente para operaciones de una sola parte. Próximo cambio en la nota a pie de página 5.
- CKM_RSA_PKCS_OAEP[4]
- CKM_AES_GCM[1, 3]
- CKM_CLOUDHSM_AES_GCM[3]
- CKM_RSA_AES_KEY_WRAP
- CKM_CLOUDHSM_AES_KEY_WRAP_NO_PAD[3]
- CKM_CLOUDHSM_AES_KEY_WRAP_PKCS5_PAD[3]

- CKM_CLOUDHSM_AES_KEY_WRAP_ZERO_PAD[3]

Tamaño máximo de datos para cada mecanismo

En la tabla siguiente, se muestra el tamaño máximo de datos establecido para cada mecanismo:

Tamaño máximo de datos

Mecanismo	Tamaño máximo de datos en bytes
CKM_SHA_1_HMAC	16288
CKM_SHA224_HMAC	16256
CKM_SHA256_HMAC	16288
CKM_SHA384_HMAC	16224
CKM_SHA512_HMAC	16224
CKM_AES_CBC	16272
CKM_AES_GCM	16224
CKM_CLOUDHSM_AES_GCM	16224
CKM_DES3_CBC	16280

Notas del mecanismo

- [1] Al realizar el cifrado AES-GCM, el HSM no acepta los datos del vector de inicialización (IV) de la aplicación. Debe utilizar un vector de inicialización generado. El IV de 12 bytes proporcionado por el HSM se escribe en la referencia de memoria a la que apunta el elemento pIV de la estructura de parámetros CK_GCM_PARAMS especificada por el usuario. Para asegurarse de no generar confusión en el usuario, el SDK de PKCS#11 versión 1.1.1 y posteriores obliga a que el elemento pIV apunte a un búfer puesto a cero cuando se inicializa el cifrado AES-GCM.
- [2] Cuando se opera con datos mediante cualquiera de los mecanismos siguientes, si el búfer de datos supera el tamaño máximo de datos, la operación produce un error. Para estos mecanismos, todo el procesamiento de los datos debe realizarse dentro del HSM. Para obtener más información

sobre los tamaños máximos de conjuntos de datos para cada mecanismo, consulte Tamaño máximo de datos para cada mecanismo.

- [3] mecanismo definido por el proveedor. Para poder utilizar los mecanismos definidos por el proveedor de CloudHSM, las aplicaciones PKCS#11 deben incluir `/opt/cloudhsm/include/pkcs11t.h` durante la compilación.

  `CKM_CLOUDHSM_AES_GCM`: este mecanismo exclusivo es una alternativa programáticamente segura del estándar `CKM_AES_GCM`. Antepone el IV generado por el HSM al texto cifrado en lugar de volver a escribirlo en la estructura `CK_GCM_PARAMS` que se proporciona durante la inicialización del cifrado. Puede utilizar este mecanismo con las funciones `C_Encrypt`, `C_WrapKey`, `C_Decrypt` y `C_UnwrapKey`. Cuando se utiliza este mecanismo, la variable pIV de la estructura `CK_GCM_PARAMS` debe establecerse en `NULL`. Cuando se utiliza este mecanismo con `C_Decrypt` y `C_UnwrapKey`, se espera que el IV se anteponga al texto cifrado que se está desencapsulando.

  `CKM_CLOUDHSM_AES_KEY_WRAP_PKCS5_PAD`:Encapsulamiento de claves AES con relleno PKCS #5

  `CKM_CLOUDHSM_AES_KEY_WRAP_ZERO_PAD`:Encapsulamiento de claves AES con relleno de ceros

- [4] Los siguientes `CK_MECHANISM_TYPE` y `CK_RSA_PKCS_MGF_TYPE` se admiten como `CK_RSA_PKCS_OAEP_PARAMS` para `CKM_RSA_PKCS_OAEP`:
  - `CKM_SHA_1` con `CKG_MGF1_SHA1`
  - `CKM_SHA224` con `CKG_MGF1_SHA224`
  - `CKM_SHA256` con `CKG_MGF1_SHA256`
  - `CKM_SHA384` con `CKM_MGF1_SHA384`
  - `CKM_SHA512` con `CKM_MGF1_SHA512`
- [5] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Operaciones de la API compatibles

La biblioteca PKCS #11 admite las siguientes operaciones API de PKCS #11.

- `C_CloseAllSessions`

- C_CloseSession
- C_CreateObject
- C_Decrypt
- C_DecryptFinal
- C_DecryptInit
- C_DecryptUpdate
- C_DeriveKey
- C_DestroyObject
- C_Digest
- C_DigestFinal
- C_DigestInit
- C_DigestUpdate
- C_Encrypt
- C_EncryptFinal
- C_EncryptInit
- C_EncryptUpdate
- C_Finalize
- C_FindObjects
- C_FindObjectsFinal
- C_FindObjectsInit
- C_GenerateKey
- C_GenerateKeyPair
- C_GenerateRandom
- C_GetAttributeValue
- C_GetFunctionList
- C_GetInfo
- C_GetMechanismInfo
- C_GetMechanismList
- C_GetSessionInfo
- C_GetSlotInfo

- `C_GetSlotList`
- `C_GetTokenInfo`
- `C_Initialize`
- `C_Login`
- `C_Logout`
- `C_OpenSession`
- `C_Sign`
- `C_SignFinal`
- `C_SignInit`
- `C_SignUpdate`
- `C_UnWrapKey`
- `C_Verify`
- `C_VerifyFinal`
- `C_VerifyInit`
- `C_VerifyUpdate`
- `C_WrapKey`

Uso de atributos clave admitidos

Un objeto de clave puede ser una clave pública, privada o secreta. Las acciones permitidas en un objeto de clave se especifican mediante atributos. Los atributos se definen cuando se crea el objeto de clave. Cuando se utiliza la biblioteca PKCS #11, asignamos los valores predeterminados que se especifican en el estándar PKCS #11.

AWS CloudHSM no admite todos los atributos enumerados en la especificación PKCS #11. Seguimos esta especificación en todos los atributos que admitimos. Estos atributos se indican en sus respectivas tablas.

Las funciones criptográficas como `C_CreateObject`, `C_GenerateKey`, `C_GenerateKeyPair`, `C_UnwrapKey` y `C_DeriveKey` que crean, modifican o copian objetos toman una plantilla de atributos como uno de sus parámetros. Para obtener más información acerca de cómo pasar una plantilla de atributos durante la creación de objetos, consulte el ejemplo Generar claves mediante la biblioteca PKCS #11.

Interpretación de la tabla de atributos de la biblioteca PKCS #11

La tabla de la biblioteca PKCS #11 contiene una lista de atributos que difieren por tipo de clave. Indica si un atributo determinado es compatible con un tipo de clave concreto cuando se utiliza una función criptográfica específica con. AWS CloudHSM

Leyenda

- ✓ indica que CloudHSM admite el atributo para el tipo de clave específico.
- ✖ indica que CloudHSM no admite el atributo para el tipo de clave específico.
- R indica que el valor del atributo se establece en de solo lectura para el tipo de clave específico.
- S indica que `GetAttributeValue` no puede leer el atributo porque distingue entre mayúsculas y minúsculas.
- Una celda vacía en la columna Valor predeterminado indica que no hay ningún valor predeterminado específico asignado al atributo.

GenerateKeyPair

Atributo	Tipo de clave					Default Value (Valor predeterminado)
		EC privada	EC pública	RSA privada	RSA pública	
`CKA_CLASS`		✓	✓	✓	✓	
`CKA_KEY_TYPE`		✓	✓	✓	✓	
`CKA_LABEL`		✓	✓	✓	✓	
`CKA_ID`		✓	✓	✓	✓	

Atributo	Tipo de clave				Default Value (Valor predeterminado)
CKA_LOCAL	R	R	R	R	True
CKA_TOKEN	✓	✓	✓	✓	False
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT	✖	✓	✖	✓	False
CKA_DECRYPT	✓	✖	✓	✖	False
CKA_DERIVE	✓	✓	✓	✓	False
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_DESTROYABLE	✓	✓	✓	✓	True
CKA_SIGN	✓	✖	✓	✖	False
CKA_SIGN_RECOVER	✖	✖	✖	✖	
CKA_VERIFY	✖	✓	✖	✓	False

Atributo	Tipo de clave				Default Value (Valor predeterminado)
CKA_VERIFY_RECOVER	✖	✖	✖	✖	
CKA_WRAP	✖	✓	✖	✓	False
CKA_WRAP_TEMPLATE	✖	✓	✖	✓	
CKA_TRUSTED	✖	✓	✖	✓	False
CKA_WRAP_WITH_TRUSTED	✓	✖	✓	✖	False
CKA_UNWRAP	✓	✖	✓	✖	False
CKA_UNWRAP_TEMPLATE	✓	✖	✓	✖	
CKA_SENSITIVE	✓[1]	✖	✓[1]	✖	True
CKA_ALWAYS_SENSITIVE	R	✖	R	✖	
CKA_EXTRACTABLE	✓	✖	✓	✖	True

Atributo	Tipo de clave				Default Value (Valor predeterminado)
CKA_NEVER_EXTRACTABLE	R	✖	R	✖	
CKA_MODULUS	✖	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	✓[2]	
CKA_PRIME_1	✖	✖	✖	✖	
CKA_PRIME_2	✖	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✖	✖	
CKA_EXPONENT_1	✖	✖	✖	✖	
CKA_EXPONENT_2	✖	✖	✖	✖	
CKA_PRIVATE_EXPONENT	✖	✖	✖	✖	
CKA_PUBLIC_EXPONENT	✖	✖	✖	✓[2]	

Atributo	Tipo de clave				Default Value (Valor predeterminado)
CKA_EC_PARAMS	✖	✓[2]	✖	✖	
CKA_EC_POINT	✖	✖	✖	✖	
CKA_VALUE	✖	✖	✖	✖	
CKA_VALUE_LEN	✖	✖	✖	✖	
CKA_CHECK_VALUE	R	R	R	R	

GenerateKey

Atributo	Tipo de clave				Default Value (Valor predeterminado)
		AES	DES3	Secreto genérico	
CKA_CLASS		✓	✓	✓	
CKA_KEY_TYPE		✓	✓	✓	
CKA_LABEL		✓	✓	✓	

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_ID	✓	✓	✓	
CKA_LOCAL	R	R	R	True
CKA_TOKEN	✓	✓	✓	False
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT	✓	✓	✖	False
CKA_DECRYPT	✓	✓	✖	False
CKA_DERIVE	✓	✓	✓	False
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	True
CKA_DESTROYABLE	✓	✓	✓	True
CKA_SIGN	✓	✓	✓	True
CKA_SIGN_RECOVER	✖	✖	✖	
CKA_VERIFY	✓	✓	✓	True
CKA_VERIFY_RECOVER	✖	✖	✖	

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_WRAP	✓	✓	✖	False
CKA_WRAP_TEMPLATE	✓	✓	✖	
CKA_TRUSTED	✓	✓	✖	False
CKA_WRAP_WITH_TRUSTED	✓	✓	✓	False
CKA_UNWRAP	✓	✓	✖	False
CKA_UNWRAP_TEMPLATE	✓	✓	✖	
CKA_SENSITIVE	✓	✓	✓	True
CKA_ALWAYS_SENSITIVE	✖	✖	✖	
CKA_EXTRACTABLE	✓	✓	✓	True
CKA_NEVER_EXTRACTABLE	R	R	R	

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_MODULUS	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	
CKA_PRIME_1	✖	✖	✖	
CKA_PRIME_2	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✖	
CKA_EXPONENT_1	✖	✖	✖	
CKA_EXPONENT_2	✖	✖	✖	
CKA_PRIVATE_EXPONENT	✖	✖	✖	
CKA_PUBLIC_EXPONENT	✖	✖	✖	
CKA_EC_PARAMS	✖	✖	✖	
CKA_EC_POINT	✖	✖	✖	

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_VALUE	✖	✖	✖	
CKA_VALUE_LEN	✓[2]	✖	✓[2]	
CKA_CHECK_VALUE	R	R	R	

CreateObject

Atributo	Tipo de clave							Default Value (Valor predeterminado)
	EC privada	EC pública	RSA privada	RSA pública	AES	DES3	Secreto genérico	
CKA_CLASS	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	
CKA_KEY_TYPE	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	
CKA_LABEL	✓	✓	✓	✓	✓	✓	✓	
CKA_ID	✓	✓	✓	✓	✓	✓	✓	
CKA_LOCAL	R	R	R	R	R	R	R	False

Atributo	Tipo de clave							Default Value (Valor predeterminado)
CKA_TOKEN	✓	✓	✓	✓	✓	✓	✓	False
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT	✖	✖	✖	✓	✓	✓	✖	False
CKA_DECRYPT	✖	✖	✓	✖	✓	✓	✖	False
CKA_DERIVE	✓	✓	✓	✓	✓	✓	✓	False
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_DESTROYABLE	✓	✓	✓	✓	✓	✓	✓	True
CKA_SIGN	✓	✖	✓	✖	✓	✓	✓	False
CKA_SIGN_RECOVER	✖	✖	✖	✖	✖	✖	✖	False
CKA_VERIFY	✖	✓	✖	✓	✓	✓	✓	False
CKA_VERIFY_RECOVER	✖	✖	✖	✖	✖	✖	✖	

Atributo	Tipo de clave							Default Value (Valor predeterminado)
CKA_WRAP	✖	✖	✖	✓	✓	✓	✖	False
CKA_WRAP_TEMPLATE	✖	✓	✖	✓	✓	✓	✖	
CKA_TRUSTED	✖	✓	✖	✓	✓	✓	✖	False
CKA_WRAP_WITH_TRUSTED	✓	✖	✓	✖	✓	✓	✓	False
CKA_UNWRAP	✖	✖	✓	✖	✓	✓	✖	False
CKA_UNWRAP_TEMPLATE	✓	✖	✓	✖	✓	✓	✖	
CKA_SENSITIVE	✓	✖	✓	✖	✓	✓	✓	True
CKA_ALWAYS_SENSITIVE	R	✖	R	✖	R	R	R	
CKA_EXTRACTABLE	✓	✖	✓	✖	✓	✓	✓	True
CKA_NEVER_EXTRACTABLE	R	✖	R	✖	R	R	R	

Atributo	Tipo de clave							Default Value (Valor predeterminado)
CKA_MODULUS	✖	✖	✓[2]	✓[2]	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	✖	✖	✖	✖	
CKA_PRIME_1	✖	✖	✓	✖	✖	✖	✖	
CKA_PRIME_2	✖	✖	✓	✖	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✓	✖	✖	✖	✖	
CKA_EXPONENT_1	✖	✖	✓	✖	✖	✖	✖	
CKA_EXPONENT_2	✖	✖	✓	✖	✖	✖	✖	
CKA_PRIVATE_EXPONENT	✖	✖	✓[2]	✖	✖	✖	✖	
CKA_PUBLIC_EXPONENT	✖	✖	✓[2]	✓[2]	✖	✖	✖	
CKA_EC_PARAMS	✓[2]	✓[2]	✖	✖	✖	✖	✖	

Atributo	Tipo de clave							Default Value (Valor predeterminado)
CKA_EC_POINT	✖	✓[2]	✖	✖	✖	✖	✖	
CKA_VALUE	✓[2]	✖	✖	✖	✓[2]	✓[2]	✓[2]	
CKA_VALUE_LEN	✖	✖	✖	✖	✖	✖	✖	
CKA_CHECK_VALUE	R	R	R	R	R	R	R	

UnwrapKey

Atributo	Tipo de clave						Default Value (Valor predeterminado)
	EC privada	RSA privada	AES	DES3	Secreto genérico		
CKA_CLASS	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]		
CKA_KEY_TYPE	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]		
CKA_LABEL	✓	✓	✓	✓	✓		

Atributo	Tipo de clave					Default Value (Valor predeterminado)
CKA_ID	✓	✓	✓	✓	✓	
CKA_LOCAL	R	R	R	R	R	False
CKA_TOKEN	✓	✓	✓	✓	✓	False
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT	✖	✖	✓	✓	✖	False
CKA_DECRYPT	✖	✓	✓	✓	✖	False
CKA_DERIVE	✓	✓	✓	✓	✓	False
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_DESTROYABLE	✓	✓	✓	✓	✓	True
CKA_SIGN	✓	✓	✓	✓	✓	False
CKA_SIGN_RECOVER	✖	✖	✖	✖	✖	False
CKA_VERIFY	✖	✖	✓	✓	✓	False

Atributo	Tipo de clave					Default Value (Valor predeterminado)
CKA_VERIFY_RECOVER	✖	✖	✖	✖	✖	
CKA_WRAP	✖	✖	✓	✓	✖	False
CKA_UNWRAP	✖	✓	✓	✓	✖	False
CKA_SENSITIVE	✓	✓	✓	✓	✓	True
CKA_EXTRACTABLE	✓	✓	✓	✓	✓	True
CKA_NEVER_EXTRACTABLE	R	R	R	R	R	
CKA_ALWAYS_SENSITIVE	R	R	R	R	R	
CKA_MODULUS	✖	✖	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	✖	✖	
CKA_PRIME_1	✖	✖	✖	✖	✖	

Atributo	Tipo de clave					Default Value (Valor predeterminado)
CKA_PRIME_2	✖	✖	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✖	✖	✖	
CKA_EXPONENT_1	✖	✖	✖	✖	✖	
CKA_EXPONENT_2	✖	✖	✖	✖	✖	
CKA_PRIVATE_EXPONENT	✖	✖	✖	✖	✖	
CKA_PUBLIC_EXPONENT	✖	✖	✖	✖	✖	
CKA_EC_PARAMS	✖	✖	✖	✖	✖	
CKA_EC_POINT	✖	✖	✖	✖	✖	
CKA_VALUE	✖	✖	✖	✖	✖	
CKA_VALUE_LEN	✖	✖	✖	✖	✖	

Atributo	Tipo de clave					Default Value (Valor predeterminado)
CKA_CHECK_VALUE		R	R	R	R	R	

DeriveKey

Atributo	Tipo de clave				Default Value (Valor predeterminado)
		AES	DES3	Secreto genérico	
CKA_CLASS		✓[2]	✓[2]	✓[2]	
CKA_KEY_TYPE		✓[2]	✓[2]	✓[2]	
CKA_LABEL		✓	✓	✓	
CKA_ID		✓	✓	✓	
CKA_LOCAL		R	R	R	True
CKA_TOKEN		✓	✓	✓	False
CKA_PRIVATE		✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT		✓	✓	✖	False

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_DECRYPT	✓	✓	✖	False
CKA_DERIVE	✓	✓	✓	False
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	True
CKA_DESTROYABLE	✓[1]	✓[1]	✓[1]	True
CKA_SIGN	✓	✓	✓	False
CKA_SIGN_RECOVER	✖	✖	✖	
CKA_VERIFY	✓	✓	✓	False
CKA_VERIFY_RECOVER	✖	✖	✖	
CKA_WRAP	✓	✓	✖	False
CKA_UNWRAP	✓	✓	✖	False
CKA_SENSITIVE	R	R	R	True
CKA_EXTRACTABLE	✓	✓	✓	True

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_NEVER_EXTRACTABLE	R	R	R	
CKA_ALWAYS_SENSITIVE	R	R	R	
CKA_MODULUS	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	
CKA_PRIME_1	✖	✖	✖	
CKA_PRIME_2	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✖	
CKA_EXPONENT_1	✖	✖	✖	
CKA_EXPONENT_2	✖	✖	✖	
CKA_PRIVATE_EXPONENT	✖	✖	✖	

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_PUBLIC_EXPONENT	✖	✖	✖	
CKA_EC_PARAMS	✖	✖	✖	
CKA_EC_POINT	✖	✖	✖	
CKA_VALUE	✖	✖	✖	
CKA_VALUE_LEN	✓[2]	✖	✓[2]	
CKA_CHECK_VALUE	R	R	R	

GetAttributeValue

Atributo	Tipo de clave						
	EC privada	EC pública	RSA privada	RSA pública	AES	DES3	Secreto genérico
CKA_CLASS	✓	✓	✓	✓	✓	✓	✓
CKA_KEY_TYPE	✓	✓	✓	✓	✓	✓	✓
CKA_LABEL	✓	✓	✓	✓	✓	✓	✓

Atributo	Tipo de clave						
CKA_ID	✓	✓	✓	✓	✓	✓	✓
CKA_LOCAL	✓	✓	✓	✓	✓	✓	✓
CKA_TOKEN	✓	✓	✓	✓	✓	✓	✓
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]
CKA_ENCRYPT	✖	✖	✖	✓	✓	✓	✖
CKA_DECRYPT	✖	✖	✓	✖	✓	✓	✖
CKA_DERIVE	✓	✓	✓	✓	✓	✓	✓
CKA_MODIFIABLE	✓	✓	✓	✓	✓	✓	✓
CKA_DESTROYABLE	✓	✓	✓	✓	✓	✓	✓
CKA_SIGN	✓	✖	✓	✖	✓	✓	✓
CKA_SIGN_RECOVER	✖	✖	✓	✖	✖	✖	✖
CKA_VERIFY	✖	✓	✖	✓	✓	✓	✓
CKA_VERIFY_RECOVER	✖	✖	✖	✓	✖	✖	✖

Atributo	Tipo de clave						
CKA_WRAP	✖	✖	✖	✓	✓	✓	✖
CKA_WRAP_TEMPLATE	✖	✓	✖	✓	✓	✓	✖
CKA_TRUSTED	✖	✓	✖	✓	✓	✓	✓
CKA_WRAP_WITH_TRUSTED	✓	✖	✓	✖	✓	✓	✓
CKA_UNWRAP	✖	✖	✓	✖	✓	✓	✖
CKA_UNWRAP_TEMPLATE	✓	✖	✓	✖	✓	✓	✖
CKA_SENSITIVE	✓	✖	✓	✖	✓	✓	✓
CKA_EXTRACTABLE	✓	✖	✓	✖	✓	✓	✓
CKA_NEVER_EXTRACTABLE	✓	✖	✓	✖	✓	✓	✓
CKA_ALWAYS_SENSITIVE	R	R	R	R	R	R	R
CKA_MODULUS	✖	✖	✓	✓	✖	✖	✖

Atributo	Tipo de clave						
CKA_MODULUS_BITS	✖	✖	✖	✓	✖	✖	✖
CKA_PRIME_1	✖	✖	S	✖	✖	✖	✖
CKA_PRIME_2	✖	✖	S	✖	✖	✖	✖
CKA_COEFFICIENT	✖	✖	S	✖	✖	✖	✖
CKA_EXPONENT_1	✖	✖	S	✖	✖	✖	✖
CKA_EXPONENT_2	✖	✖	S	✖	✖	✖	✖
CKA_PRIVATE_EXPONENT	✖	✖	S	✖	✖	✖	✖
CKA_PUBLIC_EXPONENT	✖	✖	✓	✓	✖	✖	✖
CKA_EC_PARAMS	✓	✓	✖	✖	✖	✖	✖
CKA_EC_POINT	✖	✓	✖	✖	✖	✖	✖
CKA_VALUE	S	✖	✖	✖	✓	✓	✓
CKA_VALUE_LEN	✖	✖	✖	✖	✓	✖	✓

Atributo	Tipo de clave						
CKA_CHECK_VALUE	✓	✓	✓	✓	✓	✓	✖

Notas de atributo

- [1] Este atributo es parcialmente compatible con el firmware y debe configurarse de forma explícita únicamente en el valor predeterminado.
- [2] Atributo obligatorio.

Modificación de atributos

Algunos atributos de un objeto se pueden modificar una vez creado el objeto, mientras que otros no. Para modificar los atributos, utilice el comando setAttribute de cloudhsm_mgmt_util. También puede generar una lista de atributos y las constantes que los representan mediante el comando listAttribute de cloudhsm_mgmt_util.

En la siguiente lista se muestran los atributos que se pueden modificar después de crear un objeto:

- CKA_LABEL
- CKA_TOKEN

> **Note**
>
> La modificación solo se permite para cambiar una clave de sesión por una clave de token. Utilice el comando setAttribute de key_mgmt_util para cambiar el valor del atributo.

- CKA_ENCRYPT
- CKA_DECRYPT
- CKA_SIGN
- CKA_VERIFY
- CKA_WRAP
- CKA_UNWRAP
- CKA_LABEL

- `CKA_SENSITIVE`
- `CKA_DERIVE`

> **Note**
>
> Este atributo admite la derivación de claves. Debe ser `False` para todas las claves públicas y no puede establecerse en `True`. Para las claves secretas y privadas de EC, se puede establecer en `True` o `False`.

- `CKA_TRUSTED`

> **Note**
>
> Este atributo se puede establecer en `True` o `False` solo mediante Crypto Officer (CO).

- `CKA_WRAP_WITH_TRUSTED`

> **Note**
>
> Aplique este atributo a una clave de datos exportable para especificar que solo puede encapsular esta clave con claves marcadas como `CKA_TRUSTED`. Una vez establecido `CKA_WRAP_WITH_TRUSTED` como true, el atributo pasa a ser de solo lectura y no se puede cambiar ni eliminar.

Interpretación de los códigos de error

La especificación en la plantilla de un atributo que no es compatible con una clave específica produce un error. La siguiente tabla contiene los códigos de error que se generan cuando se infringen las especificaciones:

Código de error	Descripción
`CKR_TEMPLATE_INCONSISTENT`	Este error aparece cuando se especifica un atributo en la plantilla de atributos que cumple la especificación PKCS #11, pero no es compatible con CloudHSM.

Código de error	Descripción
`CKR_ATTRIBUTE_TYPE_INVALID`	Recibirá este error cuando recupere un valor de un atributo que cumple la especific ación PKCS #11, pero no es compatible con CloudHSM.
`CKR_ATTRIBUTE_INCOMPLETE`	Este error aparece cuando no se especifica el atributo obligatorio en la plantilla de atributos.
`CKR_ATTRIBUTE_READ_ONLY`	Este error aparece cuando se especifica un atributo de solo lectura en la plantilla de atributos.

Ejemplos de código para la biblioteca PKCS #11

Los ejemplos de código que aparecen a continuación GitHub muestran cómo realizar tareas básicas con la biblioteca PKCS #11.

Requisitos previos

Antes de ejecutar las muestras, siga estos pasos para configurar su entorno:

- Instale y configure la biblioteca PKCS #11 para SDK 5 de cliente.
- Configure un usuario de criptografía (CU). La aplicación usa esta cuenta de HSM para ejecutar los ejemplos de código en el HSM.

Ejemplos de código

Los ejemplos de código de la biblioteca de AWS CloudHSM software de PKCS #11 están disponibles en. GitHub Este repositorio contiene ejemplos acerca de cómo realizar operaciones comunes con PKCS#11, como el cifrado, el descifrado, la firma y la verificación.

- Generar claves (AES, RSA, EC)
- Mostrar atributos de clave
- Cifrado y descifrado de datos con AES-GCM
- Cifrado y descifrado de datos con AES_CTR

- Cifrado y descifrado de datos con 3DES
- Firmar y verificar datos con RSA
- Derivar claves usando HMAC KDF
- Encapsule y desencapsule las claves con AES utilizando el relleno PKCS #5
- Encapsule y desencapsule las claves con AES sin relleno
- Encapsule y desencapsule las claves con AES usando cero relleno
- Encapsulamiento y desencapsulamiento de claves con AES-GCM
- Cómo encapsular y desencapsular claves con RSA

Migre su biblioteca PKCS #11 del SDK de cliente 3 al SDK de cliente 5

Utilice este tema para migrar la biblioteca PKCS #11 del SDK de cliente 3 al SDK de cliente 5. Para obtener información sobre las ventajas de la migración, consulte. Ventajas del SDK 5 de cliente

En AWS CloudHSM, las aplicaciones de los clientes realizan operaciones criptográficas mediante el kit de desarrollo de software (SDK) para AWS CloudHSM clientes. El SDK de cliente 5 es el SDK principal al que se le siguen añadiendo nuevas funciones y compatibilidad con plataformas.

Para revisar las instrucciones de migración de todos los proveedores, consulteMigración del SDK 3 de cliente al SDK 5 de cliente.

Prepárese abordando los cambios más importantes

Revise estos cambios importantes y actualice su aplicación en su entorno de desarrollo en consecuencia.

Los mecanismos de empaquetado han cambiado

Mecanismo del SDK 3 del cliente	Mecanismo equivalente al Client SDK 5
`CKM_AES_KEY_WRAP`	`CKM_CLOUDHSM_AES_KEY_WRAP_PKCS5_PAD`
`CKM_AES_KEY_WRAP_PAD`	`CKM_CLOUDHSM_AES_KEY_WRAP_ZERO_PAD`

Mecanismo del SDK 3 del cliente	Mecanismo equivalente al Client SDK 5
CKM_CLOUDHSM_AES_KEY_WRAP_PKCS5_PAD	CKM_CLOUDHSM_AES_KEY_WRAP_PKCS5_PAD
CKM_CLOUDHSM_AES_KEY_WRAP_NO_PAD	CKM_CLOUDHSM_AES_KEY_WRAP_NO_PAD
CKM_CLOUDHSM_AES_KEY_WRAP_ZERO_PAD	CKM_CLOUDHSM_AES_KEY_WRAP_ZERO_PAD

ECDH

En el SDK de cliente 3, puede usar el ECDH y especificar un KDF. Esta funcionalidad no está disponible actualmente en el SDK de cliente 5. Si su aplicación necesita esta funcionalidad, póngase en contacto con el servicio de asistencia.

Los identificadores clave ahora son específicos de cada sesión

Para utilizar correctamente los identificadores de clave en SDK 5 de cliente, debe obtener los identificadores de clave cada vez que ejecute una aplicación. Si tiene aplicaciones existentes que utilizarán los mismos identificadores de clave en distintas sesiones, debe modificar el código para obtener el identificador de clave cada vez que ejecute la aplicación. Para obtener información sobre cómo recuperar los identificadores de clave, consulte este ejemplo de AWS CloudHSM PKCS #11. Este cambio cumple con la especificación PKCS #11 2.40.

Migre al SDK 5 de cliente

Siga las instrucciones de esta sección para migrar del SDK de cliente 3 al SDK de cliente 5.

> **Note**
>
> Amazon Linux, Ubuntu 16.04, Ubuntu 18.04, CentOS 6, CentOS 8 y RHEL 6 no son compatibles actualmente con el SDK de cliente 5. Si actualmente utiliza una de estas plataformas con el SDK de cliente 3, tendrá que elegir una plataforma diferente al migrar al SDK de cliente 5.

1. Desinstale la biblioteca PKCS #11 del Client SDK 3.

Amazon Linux 2

```
$ sudo yum remove cloudhsm-pkcs11
```

CentOS 7

```
$ sudo yum remove cloudhsm-pkcs11
```

RHEL 7

```
$ sudo yum remove cloudhsm-pkcs11
```

RHEL 8

```
$ sudo yum remove cloudhsm-pkcs11
```

2. Desinstale el Client Daemon para Client SDK 3.

Amazon Linux 2

```
$ sudo yum remove cloudhsm-client
```

CentOS 7

```
$ sudo yum remove cloudhsm-client
```

RHEL 7

```
$ sudo yum remove cloudhsm-client
```

RHEL 8

```
$ sudo yum remove cloudhsm-client
```

> **Note**
>
> Es necesario volver a habilitar las configuraciones personalizadas.

3. Instale la biblioteca PKCS #11 del SDK de cliente siguiendo los pasos que se indican. Instalación de SDK 5 de cliente para la biblioteca PKCS #11
4. El Client SDK 5 presenta un nuevo formato de archivo de configuración y una nueva herramienta de arranque desde la línea de comandos. Para arrancar la biblioteca PKCS #11 del Client SDK 5, siga las instrucciones que se indican en la guía del usuario que aparece a continuación. Proceso de arranque del SDK de cliente
5. En su entorno de desarrollo, pruebe la aplicación. Actualice el código existente para resolver los cambios importantes antes de la migración final.

Temas relacionados de

- Mejores prácticas para AWS CloudHSM

Configuraciones avanzadas para PKCS #11

El proveedor AWS CloudHSM PKCS #11 incluye la siguiente configuración avanzada, que no forma parte de las configuraciones generales que utilizan la mayoría de los clientes. Estas configuraciones proporcionan capacidades adicionales.

- Conectar a varias ranuras con el PKCS #11
- Vuelva a intentar la configuración del PKCS #11

Conexión a varias ranuras con PKCS#11

Una sola ranura en la biblioteca PKCS #11 del SDK 5 de cliente representa una única conexión a un clúster en AWS CloudHSM. Con SDK 5 de cliente, puede configurar su biblioteca PKCS11 para permitir que varias ranuras conecten a los usuarios a varios clústeres de CloudHSM desde una sola aplicación PKCS #11.

Siga las instrucciones de este tema para hacer que su aplicación utilice la funcionalidad de varias ranuras para conectarse a varios clústeres.

Temas

- Requisitos previos de varias ranuras
- Configuración de la biblioteca PKCS #11 para que funcione con varias ranuras
- configure-pkcs11 add-cluster
- configure-pkcs11 remove-cluster

Requisitos previos de varias ranuras

- Dos o más AWS CloudHSM clústeres a los que desee conectarse, junto con sus certificados de clúster.
- Una instancia de EC2 con grupos de seguridad configurados correctamente para conectarse a todos los clústeres anteriores. Para obtener más información sobre cómo configurar un clúster y la instancia de cliente, consulta Cómo empezar con AWS CloudHSM.
- Para configurar la funcionalidad de varias ranuras, debe haber descargado e instalado la biblioteca PKCS #11. Si todavía no ha hecho esto, consulte las instrucciones en ???.

Configuración de la biblioteca PKCS #11 para que funcione con varias ranuras

Para configurar la biblioteca PKCS #11 para que funcione con varias ranuras, siga estos pasos:

1. Identifique los clústeres a los que desea conectarse mediante la funcionalidad de varias ranuras.
2. Agregue estos clústeres a su configuración de PKCS #11 siguiendo las instrucciones en ???
3. La próxima vez que se ejecute la aplicación PKCS #11, tendrá la funcionalidad de varias ranuras.

configure-pkcs11 add-cluster

Cuando se conecte a varias ranuras con el PKCS #11, utilice el comando configure-pkcs11 add-cluster para agregar un clúster a la configuración.

Sintaxis

```
configure-pkcs11 add-cluster [OPTIONS]
        --cluster-id <CLUSTER ID>
        [--region <REGION>]
```

```
        [--endpoint <ENDPOINT>]
        [--hsm-ca-cert <HSM CA CERTIFICATE FILE>]
        [--server-client-cert-file <CLIENT CERTIFICATE FILE>]
        [--server-client-key-file <CLIENT KEY FILE>]
        [-h, --help]
```

Ejemplos

Cómo agregar un clúster mediante el parámetro **`cluster-id`**

Example

Utilice el configure-pkcs11 add-cluster junto con el parámetro `cluster-id` para agregar un clúster (con el ID de `cluster-1234567`) a su configuración.

Linux

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 add-cluster --cluster-id cluster-1234567
```

Windows

```
C:\Program Files\Amazon\CloudHSM\> .\configure-pkcs11.exe add-cluster --cluster-
id cluster-1234567
```

> **Tip**
>
> Si al utilizar configure-pkcs11 add-cluster con el parámetro `cluster-id` el clúster no se agrega, consulte el siguiente ejemplo para obtener una versión más larga de este comando, que también requiere de los parámetros `--region` y `--endpoint` para identificar el clúster que se va a agregar. Si, por ejemplo, la región del clúster es diferente a la que está configurada como predeterminada de la AWS CLI, debe usar el parámetro `--region` para usar la región correcta. Además, puede especificar el punto de enlace de la AWS CloudHSM API que se utilizará para la llamada, lo que puede ser necesario para varias configuraciones de red, como el uso de puntos de enlace de la interfaz de VPC para los que no se utiliza el nombre de host DNS predeterminado. AWS CloudHSM

Cómo agregar un clúster mediante los parámetros **`cluster-id`**, **`endpoint`** y **`region`**

Example

Utilice el parámetro configure-pkcs11 add-cluster junto con los parámetros `cluster-id`, `endpoint` y `region`para agregar un clúster (con el ID de `cluster-1234567`) a la configuración.

Linux

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 add-cluster --cluster-
id cluster-1234567 --region us-east-1 --endpoint https://cloudhsmv2.us-
east-1.amazonaws.com
```

Windows

```
C:\Program Files\Amazon\CloudHSM\> .\configure-pkcs11.exe add-cluster --
cluster-id cluster-1234567--region us-east-1 --endpoint https://cloudhsmv2.us-
east-1.amazonaws.com
```

Para obtener más información acerca de los parámetros `--cluster-id`, `--region` y `--endpoint`, consulte the section called “Parámetros”.

Parámetros

--cluster-id ***<Cluster ID>***

Realiza una llamada `DescribeClusters` para buscar todas las direcciones IP de la interfaz de red elástica (ENI) de HSM en el clúster asociado al ID del clúster. El sistema añade las direcciones IP de ENI a los archivos de configuración. AWS CloudHSM

> **Note**
>
> Si utiliza el `--cluster-id` parámetro de una instancia EC2 dentro de una VPC que no tiene acceso a la Internet pública, debe crear un punto final de la VPC de interfaz al que conectarse. AWS CloudHSM Para obtener más información acerca de los puntos de conexión de VPC, consulte ???.

Obligatorio: sí

--punto de conexión ***<endpoint>***

Especifique el punto final AWS CloudHSM de la API utilizado para realizar la llamada. `DescribeClusters` Debe configurar esta opción en combinación con `--cluster-id`.

Obligatorio: no

-- hsm-ca-cert <HsmCA Certificate Filepath>

Especifica la ruta del archivo al certificado CA de HSM.

Obligatorio: no

--region ***<region>***

Especifique la región de su clúster. Debe configurar esta opción en combinación con `--cluster-id`.

Si no proporciona el parámetro `--region`, el sistema elige la región intentando leer las variables de entorno `AWS_DEFAULT_REGION` o `AWS_REGION`. Si esas variables no están configuradas, el sistema comprueba la región asociada a su perfil en el archivo AWS Config (normalmente `~/.aws/config`), a menos que haya especificado un archivo diferente en la variable de entorno de `AWS_CONFIG_FILE`. Si no se establece ninguna de las opciones anteriores, el sistema utilizará la región `us-east-1` de forma predeterminada.

Obligatorio: no

-- server-client-cert-file <Client Certificate Filepath>

Ruta al certificado de cliente utilizado para la autenticación mutua de TLS cliente-servidor.

Utilice esta opción únicamente si no desea utilizar la clave y el certificado SSL/TLS predeterminados que incluimos en SDK 5 de cliente. Debe configurar esta opción en combinación con `--server-client-key-file`.

Obligatorio: no

-- server-client-key-file <Client Key Filepath>

Ruta a la clave de cliente utilizada para la autenticación mutua entre cliente y servidor con TLS.

Utilice esta opción únicamente si no desea utilizar la clave y el certificado SSL/TLS predeterminados que incluimos en SDK 5 de cliente. Debe configurar esta opción en combinación con `--server-client-cert-file`.

Obligatorio: no

configure-pkcs11 remove-cluster

Cuando se conecte a varias ranuras con el PKCS #11, utilice el comando configure-pkcs11 remove-cluster para eliminar un clúster de las ranuras PKCS #11 disponibles.

Sintaxis

```
configure-pkcs11 remove-cluster [OPTIONS]
        --cluster-id <CLUSTER ID>
        [-h, --help]
```

Ejemplos

Eliminación de un clúster mediante el parámetro **`cluster-id`**

Example

Utilice el parámetro configure-pkcs11 remove-cluster junto con el parámetro `cluster-id` para eliminar un clúster (con el ID de `cluster-1234567`) de su configuración.

Linux

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 remove-cluster --cluster-
id cluster-1234567
```

Windows

```
C:\Program Files\Amazon\CloudHSM\> .\configure-pkcs11.exe remove-cluster --cluster-
id cluster-1234567
```

Para obtener más información sobre el parámetro `--cluster-id`, consulte the section called "Parámetros".

Parámetro

--cluster-id ***<Cluster ID>***

ID del clúster que se va a eliminar de la configuración

Obligatorio: sí

Comandos de reintento para PKCS #11

La versión 5.8.0 y posteriores de SDK de cliente tienen una estrategia de reintento automático integrada que reintentará las operaciones limitadas por HSM desde el lado del cliente. Cuando un HSM limita las operaciones porque está demasiado ocupado realizando operaciones anteriores y no puede aceptar más solicitudes, los SDK de cliente intentarán reintentar las operaciones limitadas hasta 3 veces y, al mismo tiempo, se retrasarán exponencialmente. Esta estrategia de reintento automático se puede configurar en uno de estos dos modos: desactivado y estándar.

- desactivado: el SDK de cliente no realizará ninguna estrategia de reintentos para ninguna operación limitada por parte del HSM.
- estándar: este es el modo predeterminado para la versión 5.8.0 y posteriores de SDK de cliente. En este modo, los SDK de cliente reintentarán ejecutar automáticamente las operaciones restringidas y se reducirán exponencialmente.

Para obtener más información, consulte Limitación de HSM.

Configuración de los comandos de reintento en modo desactivado

Linux

Cómo configurar los comandos de reintento en off para SDK 5 de cliente en Linux

- Puede utilizar los siguientes comandos para administrar la configuración de reintento en modo off:

```
$ sudo /opt/cloudhsm/bin/configure-pkcs11 --default-retry-mode off
```

Windows

Cómo configurar los comandos de reintento en off para SDK 5 de cliente en Windows

- Puede utilizar los siguientes comandos para administrar la configuración de reintento en modo off:

```
C:\Program Files\Amazon\CloudHSM\bin\ .\configure-pkcs11.exe --default-retry-mode off
```

Motor dinámico de OpenSSL

El motor dinámico de AWS CloudHSM OpenSSL le permite transferir las operaciones criptográficas a su clúster de CloudHSM a través de la API de OpenSSL.

AWS CloudHSM proporciona un motor dinámico OpenSSL, sobre el que puede obtener información en. Descarga de SSL/TLS en Linux Para ver un ejemplo sobre el uso AWS CloudHSM con OpenSSL, consulte este blog de seguridad de AWS. Para obtener información sobre la compatibilidad de plataformas con los SDK, consulte the section called “Plataformas admitidas”. Para obtener información sobre la solución de problemas, consulte. Problemas conocidos de OpenSSL Dynamic Engine

Para obtener información acerca del uso del SDK 3 de cliente, consulte SDK de cliente anterior (SDK de cliente 3).

Para obtener más información, consulte los temas siguientes.

Temas

- Instalación del motor dinámico de OpenSSL
- Tipos de claves de OpenSSL Dynamic Engine
- Mecanismos de OpenSSL Dynamic Engine
- Migre su motor dinámico OpenSSL del Client SDK 3 al Client SDK 5
- Configuraciones avanzadas para OpenSSL

Instalación del motor dinámico de OpenSSL

> **Note**
>
> Para ejecutar un único clúster de HSM con SDK 5 de cliente, primero debe administrar la configuración de durabilidad de la clave del cliente configurando `disable_key_availability_check` en `True`. Para obtener más información, consulte Sincronización de claves y Herramienta de configuración de SDK 5 de cliente.

Instalación y configuración del motor dinámico de OpenSSL

1. Utilice los comandos siguientes para descargar e instalar el motor de OpenSSL.

   Amazon Linux 2

   Instale el motor dinámico OpenSSL para Amazon Linux 2 en una arquitectura x86_64:

   ```
   $ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-dyn-latest.el7.x86_64.rpm
   ```

   ```
   $ sudo yum install ./cloudhsm-dyn-latest.el7.x86_64.rpm
   ```

   Instale el motor dinámico OpenSSL para Amazon Linux 2 en una arquitectura ARM64:

   ```
   $ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-dyn-latest.el7.aarch64.rpm
   ```

   ```
   $ sudo yum install ./cloudhsm-dyn-latest.el7.aarch64.rpm
   ```

   Amazon Linux 2023

   Instale el motor dinámico OpenSSL para Amazon Linux 2023 en la arquitectura x86_64:

   ```
   $ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/cloudhsm-dyn-latest.amzn2023.x86_64.rpm
   ```

   ```
   $ sudo yum install ./cloudhsm-dyn-latest.amzn2023.x86_64.rpm
   ```

Instale el motor dinámico OpenSSL para Amazon Linux 2023 en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/
cloudhsm-dyn-latest.amzn2023.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-dyn-latest.amzn2023.aarch64.rpm
```

CentOS 7 (7.8+)

Instale el motor dinámico OpenSSL para Centos 7 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-
dyn-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-dyn-latest.el7.x86_64.rpm
```

RHEL 7 (7.8+)

Instale el motor dinámico OpenSSL para RHEL 7 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-
dyn-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-dyn-latest.el7.x86_64.rpm
```

RHEL 8 (8.3+)

Instale el motor dinámico OpenSSL para RHEL 8 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-
dyn-latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-dyn-latest.el8.x86_64.rpm
```

RHEL 9 (9.2+)

Instale el motor dinámico OpenSSL para RHEL 9 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-dyn-latest.el9.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-dyn-latest.el9.x86_64.rpm
```

Instale el motor dinámico OpenSSL para RHEL 9 en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-dyn-latest.el9.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-dyn-latest.el9.aarch64.rpm
```

Ubuntu 20.04 LTS

Instale el motor dinámico OpenSSL para Ubuntu 20.04 LTS en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Focal/cloudhsm-dyn_latest_u20.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-dyn_latest_u20.04_amd64.deb
```

Ubuntu 22.04 LTS

Instale el motor dinámico OpenSSL para Ubuntu 22.04 LTS en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/cloudhsm-dyn_latest_u22.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-dyn_latest_u22.04_amd64.deb
```

Instale el motor dinámico OpenSSL para Ubuntu 22.04 LTS en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/cloudhsm-dyn_latest_u22.04_arm64.deb
```

```
$ sudo apt install ./cloudhsm-dyn_latest_u22.04_arm64.deb
```

Ha instalado la biblioteca compartida del motor dinámico en `/opt/cloudhsm/lib/libcloudhsm_openssl_engine.so`.

2. Iniciar SDK 5 de cliente. Para obtener más información sobre las acciones de arranque, consulte Proceso de arranque del SDK de cliente.
3. Configure una variable de entorno con las credenciales de un usuario de criptografía (CU). Para obtener información sobre cómo crear CU, consulte Uso de CMU para administrar usuarios.

```
$ export CLOUDHSM_PIN=<HSM user name>:<password>
```

> **Note**
>
> El SDK 5 de cliente introduce la variable de entorno `CLOUDHSM_PIN` para almacenar las credenciales del CU. En SDK 3 de cliente, las credenciales del CU se almacenan en la variable de entorno `n3fips_password`. El SDK 5 de cliente admite ambas variables de entorno, pero recomendamos utilizar `CLOUDHSM_PIN`.

4. Conecte su instalación de motor dinámico de OpenSSL al clúster. Para obtener más información, consulte Conexión al clúster.
5. Arranque del SDK 5 de cliente. Para obtener más información, consulte the section called “Proceso de arranque del SDK de cliente”.

Comprobar el motor dinámico de OpenSSL para el SDK 5 de cliente

Utilice el siguiente comando para comprobar la instalación del motor dinámico de OpenSSL.

```
$ openssl engine -t cloudhsm
```

El siguiente resultado comprueba su configuración:

```
(cloudhsm) CloudHSM OpenSSL Engine
     [ available ]
```

Tipos de claves de OpenSSL Dynamic Engine

El motor dinámico AWS CloudHSM OpenSSL admite los siguientes tipos de claves.

Tipo de clave	Descripción
EC	Firma/verifica el ECDSA para los tipos de claves P-256, P-384 y secp256k1. Para generar claves EC que sean interoperables con el motor OpenSSL, consulte key generate-file.
RSA	Generación de claves RSA para claves de 2048, 3072 y 4096 bits. Firma/verificación RSA. La verificación se descarga al software OpenSSL.

Mecanismos de OpenSSL Dynamic Engine

Aprenda a utilizar los mecanismos del motor dinámico de AWS CloudHSM OpenSSL.

Firma y comprobación de las funciones

El motor dinámico AWS CloudHSM OpenSSL le permite utilizar los siguientes mecanismos para las funciones de firma y verificación.

Con SDK 5 de cliente, los datos se codifican localmente en el software. Esto significa que no hay límite en el tamaño de los datos que se pueden procesar con un hash.

Tipos de firma RSA

- SHA1withRSA
- SHA224withRSA
- SHA256withRSA
- SHA384withRSA
- SHA512withRSA

Tipos de firma ECDSA

- SHA1withECDSA
- SHA224withECDSA
- SHA256withECDSA
- SHA384withECDSA
- SHA512withECDSA

Migre su motor dinámico OpenSSL del Client SDK 3 al Client SDK 5

Utilice este tema para migrar el motor dinámico de OpenSSL del Client SDK 3 al Client SDK 5. Para obtener información sobre las ventajas de la migración, consulte. Ventajas del SDK 5 de cliente

En AWS CloudHSM, las aplicaciones de los clientes realizan operaciones criptográficas mediante el kit de desarrollo de software (SDK) para AWS CloudHSM clientes. El SDK de cliente 5 es el SDK principal al que se le siguen añadiendo nuevas funciones y compatibilidad con plataformas.

> **Note**
>
> La generación de números aleatorios no se admite actualmente en Client SDK 5 con OpenSSL Dynamic Engine.

Para revisar las instrucciones de migración de todos los proveedores, consulte. Migración del SDK 3 de cliente al SDK 5 de cliente

Migre al SDK 5 de cliente

Siga las instrucciones de esta sección para migrar del SDK de cliente 3 al SDK de cliente 5.

> **Note**
>
> Amazon Linux, Ubuntu 16.04, Ubuntu 18.04, CentOS 6, CentOS 8 y RHEL 6 no son compatibles actualmente con el SDK de cliente 5. Si actualmente utiliza una de estas plataformas con el SDK de cliente 3, tendrá que elegir una plataforma diferente al migrar al SDK de cliente 5.

1. Desinstale el motor dinámico OpenSSL para Client SDK 3.

Amazon Linux 2

```
$ sudo yum remove cloudhsm-dyn
```

CentOS 7

```
$ sudo yum remove cloudhsm-dyn
```

RHEL 7

```
$ sudo yum remove cloudhsm-dyn
```

RHEL 8

```
$ sudo yum remove cloudhsm-dyn
```

2. Desinstale el Client Daemon para Client SDK 3.

Amazon Linux 2

```
$ sudo yum remove cloudhsm-client
```

CentOS 7

```
$ sudo yum remove cloudhsm-client
```

RHEL 7

```
$ sudo yum remove cloudhsm-client
```

RHEL 8

```
$ sudo yum remove cloudhsm-client
```

> **Note**
>
> Es necesario volver a habilitar las configuraciones personalizadas.

3. Instale el motor dinámico OpenSSL del SDK del cliente siguiendo los pasos que se indican en. Instalación del motor dinámico de OpenSSL
4. El Client SDK 5 presenta un nuevo formato de archivo de configuración y una nueva herramienta de arranque desde la línea de comandos. Para arrancar el motor dinámico OpenSSL Dynamic Engine de Client SDK 5, siga las instrucciones que se indican en la guía del usuario que aparece más abajo. Proceso de arranque del SDK de cliente
5. En su entorno de desarrollo, pruebe la aplicación. Actualice el código existente para resolver los cambios importantes antes de la migración final.

Temas relacionados de

- Mejores prácticas para AWS CloudHSM

Configuraciones avanzadas para OpenSSL

El proveedor de AWS CloudHSM OpenSSL incluye la siguiente configuración avanzada, que no forma parte de las configuraciones generales que utilizan la mayoría de los clientes. Estas configuraciones proporcionan capacidades adicionales.

- Comandos de reintento para OpenSSL

Comandos de reintento para OpenSSL

La versión 5.8.0 y posteriores de SDK de cliente tienen una estrategia de reintento automático integrada que reintentará las operaciones limitadas por HSM desde el lado del cliente. Cuando un HSM limita las operaciones porque está demasiado ocupado realizando operaciones anteriores y no puede aceptar más solicitudes, los SDK de cliente intentarán reintentar las operaciones limitadas hasta 3 veces y, al mismo tiempo, se retrasarán exponencialmente. Esta estrategia de reintento automático se puede configurar en uno de estos dos modos: desactivado y estándar.

- desactivado: el SDK de cliente no realizará ninguna estrategia de reintentos para ninguna operación limitada por parte del HSM.
- estándar: este es el modo predeterminado para la versión 5.8.0 y posteriores de SDK de cliente. En este modo, los SDK de cliente reintentarán ejecutar automáticamente las operaciones restringidas y se reducirán exponencialmente.

Para obtener más información, consulte Limitación de HSM.

Configuración de los comandos de reintento en modo desactivado

Linux

Para configurar los comandos de reintento en off para SDK 5 de cliente en Linux

- Puede utilizar uno de los siguientes comandos para configurar los comandos de reintento en modo off:

```
$ sudo /opt/cloudhsm/bin/configure-dyn --default-retry-mode off
```

Windows

Para configurar los comandos de reintento en off para SDK 5 de cliente en Windows

- Puede utilizar uno de los siguientes comandos para configurar los comandos de reintento en modo off:

```
C:\Program Files\Amazon\CloudHSM\bin\ .\configure-dyn.exe --default-retry-mode
 off
```

Proveedor de JCE

El proveedor AWS CloudHSM JCE es una implementación de proveedor creada a partir del marco de proveedores de Java Cryptographic Extension (JCE). El JCE le permite llevar a cabo operaciones criptográficas usando el kit de desarrollo de Java (JDK). En esta guía, el proveedor de AWS CloudHSM JCE a veces se denomina proveedor de JCE. Utilice el proveedor de JCE y el JDK para transferir las operaciones criptográficas de descarga al HSM. Para solucionar problemas, consulte. Problemas conocidos para el SDK de JCE

Para obtener información acerca del uso del SDK 3 de cliente, consulte SDK de cliente anterior (SDK de cliente 3).

Temas

- Instalar y usar el proveedor AWS CloudHSM JCE para Client SDK 5
- Tipos de claves compatibles
- Mecanismos compatibles
- Atributos Java admitidos
- Ejemplos de código para la biblioteca de AWS CloudHSM software para Java
- AWS CloudHSM Proveedor de ICE Javadocs
- Uso de la clase AWS CloudHSM KeyStore Java
- Migre su proveedor de JCE de Client SDK 3 a Client SDK 5
- Configuraciones avanzadas para JCE

Instalar y usar el proveedor AWS CloudHSM JCE para Client SDK 5

El proveedor JCE es compatible con OpenJDK 8, OpenJDK 11, OpenJDK 17 y OpenJDK 21. Puede descargar ambos desde el sitio web de OpenJDK.

> **Note**
>
> Para ejecutar un único clúster de HSM con SDK 5 de cliente, primero debe administrar la configuración de durabilidad de la clave del cliente configurando `disable_key_availability_check` en `True`. Para obtener más información, consulte Sincronización de claves y Herramienta de configuración de SDK 5 de cliente.

Temas

- Instalación de los proveedores JCE
- Proporcione las credenciales al proveedor de JCE.
- Aspectos básicos de gestión de claves en el proveedor de JCE

Instalación de los proveedores JCE

1. Utilice el siguiente comando para descargar e instalar el proveedor de JCE.

Amazon Linux 2

Instale el proveedor JCE para Amazon Linux 2 en una arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-jce-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-jce-latest.el7.x86_64.rpm
```

Instale el proveedor JCE para Amazon Linux 2 en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-jce-latest.el7.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-jce-latest.el7.aarch64.rpm
```

Amazon Linux 2023

Instale el proveedor JCE para Amazon Linux 2023 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/cloudhsm-jce-latest.amzn2023.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-jce-latest.amzn2023.x86_64.rpm
```

Instale el proveedor JCE para Amazon Linux 2023 en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Amzn2023/cloudhsm-jce-latest.amzn2023.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-jce-latest.amzn2023.aarch64.rpm
```

CentOS 7 (7.8+)

Instale el proveedor JCE para Centos 7 en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-jce-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-jce-latest.el7.x86_64.rpm
```

RHEL 7 (7.8+)

Instale el proveedor JCE para RHEL 7 en una arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-jce-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-jce-latest.el7.x86_64.rpm
```

RHEL 8 (8.3+)

Instale el proveedor JCE para RHEL 8 en una arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-jce-latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-jce-latest.el8.x86_64.rpm
```

RHEL 9 (9.2+)

Instale el proveedor JCE para RHEL 9 (9.2+) en una arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-jce-latest.el9.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-jce-latest.el9.x86_64.rpm
```

Instale el proveedor JCE para RHEL 9 (9.2+) en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL9/cloudhsm-jce-latest.el9.aarch64.rpm
```

```
$ sudo yum install ./cloudhsm-jce-latest.el9.aarch64.rpm
```

Ubuntu 20.04 LTS

Instale el proveedor JCE para Ubuntu 20.04 LTS en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Focal/
cloudhsm-jce_latest_u20.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-jce_latest_u20.04_amd64.deb
```

Ubuntu 22.04 LTS

Instale el proveedor JCE para Ubuntu 22.04 LTS en la arquitectura x86_64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/
cloudhsm-jce_latest_u22.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-jce_latest_u22.04_amd64.deb
```

Instale el proveedor JCE para Ubuntu 22.04 LTS en la arquitectura ARM64:

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Jammy/
cloudhsm-jce_latest_u22.04_arm64.deb
```

```
$ sudo apt install ./cloudhsm-jce_latest_u22.04_arm64.deb
```

Windows Server 2016

Instale el proveedor JCE para Windows Server 2016 en una arquitectura x86_64, ábralo PowerShell como administrador y ejecute el siguiente comando:

```
PS C:\> wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/
Windows/AWSCloudHSMJCE-latest.msi -Outfile C:\AWSCloudHSMJCE-latest.msi
```

```
PS C:\> Start-Process msiexec.exe -ArgumentList '/i C:\AWSCloudHSMJCE-
latest.msi /quiet /norestart /log C:\client-install.txt' -Wait
```

Windows Server 2019

Instale el proveedor JCE para Windows Server 2019 en una arquitectura x86_64, ábralo PowerShell como administrador y ejecute el siguiente comando:

```
PS C:\> wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/
Windows/AWSCloudHSMJCE-latest.msi -Outfile C:\AWSCloudHSMJCE-latest.msi
```

```
PS C:\> Start-Process msiexec.exe -ArgumentList '/i C:\AWSCloudHSMJCE-
latest.msi /quiet /norestart /log C:\client-install.txt' -Wait
```

2. Iniciar SDK 5 de cliente. Para obtener más información sobre las acciones de arranque, consulte Proceso de arranque del SDK de cliente.
3. Localice los siguientes archivos de proveedor de JCE:

   Linux

   - `/opt/cloudhsm/java/cloudhsm-version.jar`
   - `/opt/cloudhsm/bin/configure-jce`
   - `/opt/cloudhsm/bin/jce-info`

   Windows

   - `C:\Program Files\Amazon\CloudHSM\java\cloudhsm-version.jar>`
   - `C:\Program Files\Amazon\CloudHSM\bin\configure-jce.exe`
   - `C:\Program Files\Amazon\CloudHSM\bin\jce_info.exe`

Proporcione las credenciales al proveedor de JCE.

Los HSM necesitan autenticar la aplicación de Java antes de que la aplicación pueda utilizarlos. Los HSM autentican una sesión mediante el método de inicio de sesión explícito o implícito.

Inicio de sesión explícito: este método le permite proporcionar las credenciales de AWS CloudHSM directamente en la aplicación. Utiliza el método `AuthProvider`, en el que se pasa el nombre de

usuario y la contraseña del CU en el patrón pin. Para obtener más información, consulte el ejemplo de código Inicio de sesión en un HSM.

Inicio de sesión implícito: este método le permite definir las credenciales de AWS CloudHSM en un nuevo archivo de propiedades, en las propiedades del sistema o como variables de entorno.

- Propiedades del sistema: defina las credenciales mediante las propiedades del sistema al ejecutar la aplicación. En los siguientes ejemplos, se muestran dos maneras diferentes de hacerlo:

  Linux

  ```
  $ java -DHSM_USER=<HSM user name> -DHSM_PASSWORD=<password>
  ```

  ```
  System.setProperty("HSM_USER","<HSM user name>");
  System.setProperty("HSM_PASSWORD","<password>");
  ```

  Windows

  ```
  PS C:\> java -DHSM_USER=<HSM user name> -DHSM_PASSWORD=<password>
  ```

  ```
  System.setProperty("HSM_USER","<HSM user name>");
  System.setProperty("HSM_PASSWORD","<password>");
  ```

- Variables de entorno: defina las credenciales como variables de entorno.

  Linux

  ```
  $ export HSM_USER=<HSM user name>
  $ export HSM_PASSWORD=<password>
  ```

  Windows

  ```
  PS C:\> $Env:HSM_USER="<HSM user name>"
  PS C:\> $Env:HSM_PASSWORD="<password>"
  ```

Es posible que las credenciales no estén disponibles si la aplicación no las proporciona o si se intenta realizar una operación antes de que el HSM autentique la sesión. En esos casos, la biblioteca de software de CloudHSM para Java busca las credenciales en el orden que se indica a continuación:

1. Propiedades del sistema
2. Variables de entorno

Aspectos básicos de gestión de claves en el proveedor de JCE

Los aspectos básicos de la administración de claves en el proveedor de JCE están relacionados con la importación o la exportación de claves, la carga de claves por identificador o la eliminación de claves. Para obtener más información acerca de la administración de claves, consulte el ejemplo de código de administración de claves.

También puede encontrar más ejemplos de código de proveedor de JCE en Ejemplos de código.

Tipos de claves compatibles

La biblioteca de AWS CloudHSM software para Java permite generar los siguientes tipos de claves.

Tipo de clave	Descripción
AES	Genere claves AES de 128, 192 y 256 bits.
Triple DES (3DES, DESede)	Genere una clave DES triple de 192 bits Consulte la nota a pie de página 1 para ver los próximos cambios.
EC	Genere pares de claves EC: curvas NIST secp224r1 (P-224), secp256r1 (P-256), secp256k1 (Blockchain), secp384r1 (P-384) y secp521r1 (P-521).
GENERIC_SECRET	Genere secretos genéricos de 1 a 800 bytes.
HMAC	Soporte de hash para SHA1, SHA224, SHA256, SHA384, SHA512.
RSA	Genere claves RSA de 2048 a 4096 bits, en incrementos de 256 bits

[1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Mecanismos compatibles

Para obtener información sobre las interfaces y las clases de motor de la arquitectura criptográfica de Java (JCA) compatibles AWS CloudHSM, consulte los temas siguientes.

Temas

- Generación de funciones de claves y pares de claves
- Funciones de cifrado
- Firma y comprobación de las funciones
- Funciones Digest
- Funciones de código de autenticación de mensajes basado en hash (HMAC).
- Funciones de código de autenticación de mensajes basados en cifrado (CMAC)
- Conversión de las claves en especificaciones clave con generadores de claves
- Notas del mecanismo

Generación de funciones de claves y pares de claves

La biblioteca de AWS CloudHSM software para Java le permite utilizar las siguientes operaciones para generar funciones de claves y de pares de claves.

- `RSA`
- `EC`
- `AES`
- `DESede (Triple DES)`[consulte la nota 1]
- `GenericSecret`

Funciones de cifrado

La biblioteca de AWS CloudHSM software para Java admite las siguientes combinaciones de algoritmo, modo y relleno.

Algoritmo	Mode	Rellenado	Notas

AES	CBC	`AES/CBC/NoPadding`
		
		`AES/CBC/PKCS5Padding`
			Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE`.
			
			Implementa `Cipher.UNWRAP_MODE` for `AES/CBC NoPadding`

AES	ECB	`AES/ECB/PKCS5Padding`
		
		`AES/ECB/NoPadding`
			Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE`.

AES	CTR	`AES/CTR/NoPadding`	Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE`.
AES	GCM	`AES/GCM/NoPadding`	Implementa `Cipher.WRAP_MODE`, `Cipher.UNWRAP_MODE`, `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE`. Al realizar el cifrado AES-GCM, el HSM

Algoritmo	Mode	Rellenado	Notas
			no tiene en cuenta el vector de inicializ ación (IV) de la solicitud y utiliza un IV que él mismo genera. Una vez que se ha completado la operación, deberá llamar a `Cipher.ge tIV()` para obtener el IV.

AESWrap	ECB	`AESWrap/ECB/ NoPadding`
		`AESWrap/ECB/ PKCS5Padding`
		`AESWrap/ECB/ ZeroPadding`
			Implementa `Cipher.WR AP_MODE` y `Cipher.UN WRAP_MODE` .

DESede (Triple DES)	CBC	`DESede/CBC/ PKCS5Padding` `DESede/CBC/ NoPadding`	Implementa `Cipher.EN CRYPT_MODE` y `Cipher.DE CRYPT_MODE` . Consulte la nota 1 que aparece a continuación para ver los próximos cambios.

Algoritmo	Mode	Rellenado	Notas

DESede (Triple DES)	ECB	`DESede/ECB/NoPadding`
		
		`DESede/ECB/PKCS5Padding`
			Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE` . Consulte la nota 1 que aparece a continuación para ver los próximos cambios.

Algoritmo	Mode	Rellenado	Notas

RSA	ECB	`RSA/ECB/PKCS1Padding` consulte la nota 1
		
		`RSA/ECB/OAEPPadding`
		
		`RSA/ECB/OAEPWithSHA-1ANDMGF1Padding`
		
		`RSA/ECB/OAEPWithSHA-224ANDMGF1Padding`
		
		`RSA/ECB/OAEPWithSHA-256ANDMGF1Padding`
		
		`RSA/ECB/OAEPWithSHA-384ANDMGF1Padding`
		
		`RSA/ECB/OAEPWithSHA-512ANDMGF1Padding`
			Implementa `Cipher.WRAP_MODE`, `Cipher.UNWRAP_MODE`, `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE`.

Algoritmo	Mode	Rellenado	Notas
RSA	ECB	`RSA/ECB/NoPadding`	Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE` .

RSAAESWrap	ECB	`RSAAESWrap/ECB/OAEPPadding`
		
		`RSAAESWrap/ECB/OAEPWithSHA-1ANDMGF1Padding`
		
		`RSAAESWrap/ECB/OAEPWithSHA-224ANDMGF1Padding`
		
		`RSAAESWrap/ECB/OAEPWithSHA-256ANDMGF1Padding`
		
		`RSAAESWrap/ECB/OAEPWithSHA-384ANDMGF1Padding`
		
		`RSAAESWrap/ECB/OAEPWithSHA-512ANDMGF1Padding`
			Implementa `Cipher.WRAP_MODE` y `Cipher.UNWRAP_MODE` .

Firma y comprobación de las funciones

La biblioteca de AWS CloudHSM software para Java admite los siguientes tipos de firma y verificación. Con el SDK 5 de cliente y los algoritmos de firma con hash, los datos se codifican localmente en el software antes de enviarlos al HSM para su firma o comprobación. Esto significa que no hay límite en cuanto al tamaño de los datos que el SDK puede codificar.

Tipos de firma RSA

- `NONEwithRSA`
- `RSASSA-PSS`
- `SHA1withRSA`
- `SHA1withRSA/PSS`
- `SHA1withRSAandMGF1`
- `SHA224withRSA`
- `SHA224withRSAandMGF1`
- `SHA224withRSA/PSS`
- `SHA256withRSA`
- `SHA256withRSAandMGF1`
- `SHA256withRSA/PSS`
- `SHA384withRSA`
- `SHA384withRSAandMGF1`
- `SHA384withRSA/PSS`
- `SHA512withRSA`
- `SHA512withRSAandMGF1`
- `SHA512withRSA/PSS`

Tipos de firma ECDSA

- `NONEwithECDSA`
- `SHA1withECDSA`
- `SHA224withECDSA`

- SHA256withECDSA
- SHA384withECDSA
- SHA512withECDSA

Funciones Digest

La biblioteca de AWS CloudHSM software para Java admite los siguientes resúmenes de mensajes. Con SDK 5 de cliente, los datos se codifican localmente en el software. Esto significa que no hay límite en cuanto al tamaño de los datos que el SDK puede codificar.

- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

Funciones de código de autenticación de mensajes basado en hash (HMAC).

La biblioteca AWS CloudHSM de software para Java admite los siguientes algoritmos HMAC.

- HmacSHA1 (Tamaño máximo de datos en bytes: 16288)
- HmacSHA224 (Tamaño máximo de datos en bytes: 16256)
- HmacSHA256 (Tamaño máximo de datos en bytes: 16288)
- HmacSHA384 (Tamaño máximo de datos en bytes: 16224)
- HmacSHA512 (Tamaño máximo de datos en bytes: 16224)

Funciones de código de autenticación de mensajes basados en cifrado (CMAC)

Los CMAC (códigos de autenticación de mensajes basados en cifrado) crean códigos de autenticación de mensajes (MAC) mediante un cifrado por bloques y una clave secreta. Se diferencian de los HMAC en que utilizan un método de clave simétrica de bloques para los MAC, en lugar de un método de hash.

La biblioteca AWS CloudHSM de software para Java admite los siguientes algoritmos CMAC.

- `AESCMAC`

Conversión de las claves en especificaciones clave con generadores de claves

Puede utilizar las fábricas de claves para convertir las claves en especificaciones clave. AWS CloudHSM tiene dos tipos de fábricas clave para JCE:

SecretKeyFactory: Se utiliza para importar o derivar claves simétricas. Con SecretKeyFactory él, puede pasar una clave compatible o una clave compatible KeySpec para importar o derivar claves simétricas. AWS CloudHSM Las siguientes son las especificaciones compatibles para KeyFactory:

- SecretKeyFactoryEl `generateSecret` método de For admite KeySpeclas siguientes clases:
  - KeyAttributesMapse puede usar para importar bytes de una clave con atributos adicionales como clave de CloudHSM. Puede encontrar un ejemplo aquí.
  - SecretKeySpecse puede usar para importar una especificación de clave simétrica como clave de CloudHSM.
  - AesCmacKdfParameterSpecse puede utilizar para derivar claves simétricas mediante otra clave AES de CloudHSM.

> **Note**
>
> SecretKeyFactoryEl `translateKey` método utiliza cualquier clave que implemente la interfaz clave.

KeyFactory: Se utiliza para importar claves asimétricas. Con KeyFactory él, puede pasar una clave compatible o se puede importar una clave asimétrica KeySpec a ella. AWS CloudHSM Para obtener más información, consulte los siguientes recursos:

- Para KeyFactory el `generatePublic` método de For, se admiten KeySpeclas siguientes clases:
- KeyAttributesMap CloudHSM para RSA y EC, que incluye: KeyTypes
  - KeyAttributesMap CloudHSM para el público de RSA y EC. KeyTypes Puede encontrar un ejemplo aquí.
  - X509 EncodedKeySpec para clave pública RSA y EC
  - RSA PublicKeySpec para clave pública RSA
  - EC PublicKeySpec para clave pública de EC

- Para KeyFactory el `generatePrivate` método de For, se admiten KeySpeclas siguientes clases:
- KeyAttributesMap CloudHSM para RSA y EC, que incluye: KeyTypes
  - KeyAttributesMap CloudHSM para el público de RSA y EC. KeyTypes Puede encontrar un ejemplo aquí.
  - PKCS8 EncodedKeySpec para claves privadas EC y RSA
  - RSA para clave privada RSA PrivateCrtKeySpec
  - EC PrivateKeySpec para clave privada EC

Para KeyFactory el `translateKey` método, incluye cualquier clave que implemente la interfaz clave.

Notas del mecanismo

[1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Atributos Java admitidos

Este tema describe cómo puede utilizar una extensión propia para el proveedor JCE para establecer atributos de clave. Utilice esta extensión para establecer los atributos de clave admitidos y sus valores durante estas operaciones:

- Generación de claves
- Importación de claves

Para ver ejemplos de cómo utilizar los atributos de clave, consulte the section called “Ejemplos de código”.

Temas

- Descripción de los atributos
- Atributos admitidos
- Configuración de atributos para claves

Descripción de los atributos

Los atributos de clave se utilizan para especificar qué acciones se permiten en objetos relacionados con las claves, como claves públicas, privadas o secretas. Los atributos y valores de clave se definen durante las operaciones de creación de objetos de clave.

Sin embargo, Java Cryptography Extension (JCE) no especifica cómo deben establecerse los valores de los atributos de clave, por lo que, de forma predeterminada, se permiten la mayoría de las acciones. Por el contrario, el estándar PKCS #11 define un completo conjunto de atributos con valores predeterminados más restrictivos. Empezando por el proveedor JCE 3.1, AWS CloudHSM proporciona una extensión patentada que permite establecer valores más restrictivos para los atributos de uso común.

Atributos admitidos

Puede establecer valores para los atributos que aparecen en la tabla siguiente. Es recomendable que solamente establezca valores para los atributos que desee hacer más restrictivos. Si no especifica ningún valor, AWS CloudHSM utiliza el valor predeterminado especificado en la tabla siguiente. Las celdas vacías de la columna «Valor predeterminado» indican que no hay ningún valor predeterminado específico asignado al atributo.

Atributo	Valor predeterminado			Notas
	Clave simétrica	Clave pública del par de claves	Clave privada del par de claves	
DECRYPT	TRUE		TRUE	True indica que la clave se puede utilizar para descifrar cualquier búfer. Por lo general, esto se establece como FALSE para una clave cuya propiedad WRAP está

Atributo	Valor predeterminado			Notas
	Clave simétrica	Clave pública del par de claves	Clave privada del par de claves	
				configurada como true.
DERIVE				Permite utilizar una clave para derivar otras claves.
ENCRYPT	TRUE	TRUE		True indica que la clave se puede utilizar para cifrar cualquier búfer.
EXTRACTABLE	TRUE		TRUE	True indica que esta clave se puede exportar fuera del HSM.
ID				Un valor definido por el usuario que se utiliza para identificar la clave.
KEY_TYPE				Se utiliza para identificar el tipo de clave (AES, DeSede, secreto genérico, EC o RSA).

Atributo	Valor predeterminado			Notas
	Clave simétrica	Clave pública del par de claves	Clave privada del par de claves	
LABEL				Una cadena definida por el usuario que le permite identificar cómodamente las claves de su HSM. Para seguir las mejores prácticas, utilice una etiqueta única para cada clave para que sea más fácil encontrarla más adelante.
LOCAL				Indica una clave generada por el HSM.
OBJECT_CLASS				Se utiliza para identificar la clase de objeto de una clave (SecretKey PublicKey o PrivateKey).

Atributo	Valor predeterminado			Notas
	Clave simétrica	Clave pública del par de claves	Clave privada del par de claves	
PRIVATE	TRUE	TRUE	TRUE	True indica que es posible que los usuarios no tengan acceso a la clave hasta que se autentiqu en. Para mayor claridad, los usuarios no pueden acceder a ninguna clave AWS CloudHSM hasta que se hayan autentica do, incluso si este atributo está establecido en FALSE.
SIGN	TRUE		TRUE	True indica que la clave se puede utilizar para firmar un resumen del mensaje. Normalmente, se utiliza el valor FALSE con las claves públicas y privadas que se han archivado.

Atributo	Valor predeterminado			Notas
	Clave simétrica	Clave pública del par de claves	Clave privada del par de claves	
SIZE				Atributo que define el tamaño de una clave. Para obtener más información sobre los tamaños de clave compatibles, consulte Mecanismos compatibles con SDK 5 de cliente.
TOKEN	FALSE	FALSE	FALSE	Clave permanente que se replica en todos los HSM del clúster y se incluye en las copias de seguridad. TOKEN = FALSE implica el uso de una clave efímera, que se borra automáticamente cuando se interrumpe la conexión con ese HSM o se cierra la sesión.

Atributo	Valor predeterminado			Notas
	Clave simétrica	Clave pública del par de claves	Clave privada del par de claves	
UNWRAP	TRUE		TRUE	True indica que la clave se puede utilizar para desencapsular (importar) otra clave.
VERIFY	TRUE	TRUE		True indica que la clave se puede utilizar para verificar una firma. Normalmente, se utiliza el valor FALSE con las claves privadas.
WRAP	TRUE	TRUE		True indica que la clave se puede utilizar para encapsular otra clave. Por lo general, se utilizará el valor FALSE con las claves privadas.

Atributo	Valor predeterminado			Notas
	Clave simétrica	Clave pública del par de claves	Clave privada del par de claves	
WRAP_WITH_TRUSTED	FALSE		FALSE	Si su valor es verdadero, indica que una clave solo se puede encapsular y desencapsular con claves que tengan el atributo TRUSTED establecido en true. Una vez que una clave se establece como WRAP_WITH_TRUSTED , ese atributo es de solo lectura y no se puede establecer como false. Para obtener más información sobre el encapsulamiento de claves de confianza, consulte Uso de claves de confianza para controlar el desencaps

Atributo	Valor predeterminado			Notas
	Clave simétrica	Clave pública del par de claves	Clave privada del par de claves	
				ulamiento de claves.

> **Note**
>
> La compatibilidad con los atributos de la biblioteca PKCS #11 es más amplia. Para obtener más información, consulte Atributos de PKCS #11 admitidos.

Configuración de atributos para claves

`KeyAttributesMap` es un objeto similar a Java Map, que puede usar para establecer valores de atributo en los objetos de clave. Los métodos de la función `KeyAttributesMap` son iguales que los métodos que se utilizan para manipular mapas de Java.

Si desea establecer valores personalizados en los atributos, tiene dos opciones:

- Utilizar los métodos que se indican en la tabla siguiente
- Utilizar los modelos de Builder que se ilustran más adelante en este documento

Los objetos de mapa de atributos admiten los siguientes métodos para establecer atributos:

Operación	Valor de retorno	Método de **`KeyAttributesMap`**
Obtener el valor de un atributo de clave para una clave existente	Objeto (que contiene el valor) o null	get(keyAttribute)
Rellenar el valor de un atributo de clave	Valor anterior asociado con el atributo de clave o null si no	put(keyAttribute, valor)

Operación	Valor de retorno	Método de **KeyAttributesMap**
	había ninguna asignación de un atributo de clave	
Rellenar valores en varios atributos de clave	N/A	PuTall () keyAttributesMap
Eliminar un par clave-valor del mapa de atributos	Valor anterior asociado con el atributo de clave o null si no había ninguna asignación de un atributo de clave	remove(keyAttribute)

> **Note**
>
> Los atributos que no se especifican explícitamente se establecen en los valores predeterminados que se indican en la tabla anterior de the section called “Atributos admitidos”.

Configuración de atributos para un par de claves

Utilice la clase `KeyPairAttributesMap` de Java para administrar los atributos de clave de un par de claves. `KeyPairAttributesMap` encapsula dos objetos `KeyAttributesMap`: uno para una clave pública y otro para una clave privada.

Para establecer por separado atributos específicos en la clave pública y en la clave privada, puede utilizar el método `put()` en el objeto de mapa `KeyAttributes` que corresponda a esa clave. Utilice el método `getPublic()` para recuperar el mapa de atributos de la clave pública y utilice `getPrivate()` para recuperar el mapa de atributos de la clave privada. Puede rellenar el valor de varios atributos de clave a la vez tanto de pares de claves públicas como de pares de claves privadas utilizando `putAll()` con un mapa de atributos de pares de claves como argumento.

Ejemplos de código para la biblioteca de AWS CloudHSM software para Java

Requisitos previos

Antes de ejecutar las muestras, debe configurar el entorno:

- Instale y configure el Proveedor de la Extensión Criptográfica de Java (JCE).
- Configure un nombre de usuario y contraseña de HSM válidos. Los permisos del usuario criptográfico (CU) son suficientes para estas tareas. La aplicación utiliza estas credenciales para iniciar sesión en el HSM en cada ejemplo.
- Decida cómo proporcionar las credenciales al proveedor de JCE.

Ejemplos de código

Los siguientes ejemplos de código muestran cómo utilizar el proveedor de JCE de AWS CloudHSM para realizar tareas básicas. Hay más ejemplos de código disponibles en GitHub.

- Inicio de sesión en un HSM
- Administración de claves
- Generación de claves simétricas
- Generación de claves asimétricas
- Cifrado y descifrado con AES-GCM
- Cifrado y descifrado con AES-CTR
- Encriptar y descifrar con DESede-ECB[ver nota 1]
- Firma y verificación con claves RSA
- Firma y verificación con claves de EC
- Usar atributos clave admitidos
- Uso del almacén de claves de CloudHSM

[1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

AWS CloudHSM Proveedor de ICE Javadocs

Utilice Javadoc, el proveedor de JCE, para obtener información de uso sobre los tipos y métodos de Java definidos en el SDK JCE de AWS CloudHSM. Para descargar los Javadocs más recientes AWS CloudHSM, consulte la Versión más reciente sección de la página de descargas.

Puede importar Javadocs a un entorno de desarrollo integrado (IDE) o visualizarlas en un navegador web.

Uso de la clase AWS CloudHSM KeyStore Java

La AWS CloudHSM `KeyStore` clase proporciona un almacén de claves PKCS12 para fines especiales. Este almacén de claves puede almacenar certificados junto con datos de la clave y relacionar estos certificados con los datos de clave que están almacenados en AWS CloudHSM. La AWS CloudHSM `KeyStore` clase implementa la interfaz de proveedor `KeyStore` de servicios (SPI) de la extensión de criptografía de Java (JCE). Para obtener más información sobre su uso`KeyStore`, consulte Class. KeyStore

> **Note**
>
> Dado que los certificados son información pública y, para maximizar la capacidad de almacenamiento de las claves criptográficas, no AWS CloudHSM admite el almacenamiento de certificados en los HSM.

Elección del almacén de claves apropiado

El proveedor de AWS CloudHSM Java Cryptographic Extension (JCE) ofrece un AWS CloudHSM específico. KeyStore La AWS CloudHSM `KeyStore` clase admite la transferencia de operaciones clave al HSM, el almacenamiento local de certificados y las operaciones basadas en certificados.

Cargue el CloudHSM de uso especial de la siguiente manera: KeyStore

```
KeyStore ks = KeyStore.getInstance("CloudHSM")
```

Inicializando AWS CloudHSM KeyStore

Inicie sesión de AWS CloudHSM KeyStore la misma forma en que inicia sesión en el proveedor de JCE. Puede usar variables de entorno o el archivo de propiedades del sistema, y debe iniciar sesión

antes de empezar a usar CloudHSM KeyStore. Para ver un ejemplo de inicio de sesión en un HSM mediante el proveedor JCE, consulte Inicio de sesión en un HSM.

Si lo desea, puede especificar una contraseña para cifrar el archivo PKCS12 local que contiene los datos del almacén de claves. Al crear el AWS CloudHSM almacén de claves, se establece la contraseña y se la proporciona cuando se utilizan los métodos load, set y get.

Cree una instancia de un nuevo objeto KeyStore CloudHSM de la siguiente manera:

```
ks.load(null, null);
```

Para escribir los datos del almacén de claves en un archivo, utilice el método `store`. A partir de ese momento, puede cargar el almacén de claves existente utilizando el método `load` con el archivo de origen y la contraseña de la siguiente manera:

```
ks.load(inputStream, password);
```

Usando AWS CloudHSM KeyStore

AWS CloudHSM KeyStore cumple con la KeyStore especificación de la clase JCE y proporciona las siguientes funciones.

- `load`

  Carga el almacén de claves a partir de la secuencia de entrada especificada. Si se estableció una contraseña al guardar el almacén de claves, debe proporcionarse esta misma contraseña para que la carga se realice correctamente. Establezca los dos parámetros en null para inicializar un nuevo almacén de claves vacío.

  ```
  KeyStore ks = KeyStore.getInstance("CloudHSM");
  ks.load(inputStream, password);
  ```

- `aliases`

  Devuelve una enumeración de los nombres de alias de todas las entradas de la instancia especificada del almacén de claves. Los resultados incluyen objetos almacenados localmente en el archivo PKCS12 y objetos residentes en el HSM.

  Código de muestra:

```
KeyStore ks = KeyStore.getInstance("CloudHSM");
for(Enumeration<String> entry = ks.aliases(); entry.hasMoreElements();) {
    String label = entry.nextElement();
    System.out.println(label);
}
```

- `containsalias`

  Devuelve true si el almacén de claves tiene acceso al menos a un objeto con el alias especificado. El almacén de claves comprueba los objetos almacenados localmente en el archivo PKCS12 y los objetos residentes en el HSM.

- `deleteEntry`

  Elimina una entrada de certificado del archivo PKCS12 local. No se admite la eliminación de datos clave almacenados en un HSM mediante el. AWS CloudHSM KeyStore Puede eliminar las claves mediante el método `destroy` de la interfaz Destruible.

  ```
  ((Destroyable) key).destroy();
  ```

- `getCertificate`

  Devuelve el certificado asociado a un alias, si está disponible. Si el alias no existe o hace referencia a un objeto que no es un certificado, la función devuelve NULL.

  ```
  KeyStore ks = KeyStore.getInstance("CloudHSM");
  Certificate cert = ks.getCertificate(alias);
  ```

- `getCertificateAlias`

  Devuelve el nombre (alias) de la primera entrada del almacén de claves cuyos datos coinciden con el certificado especificado.

  ```
  KeyStore ks = KeyStore.getInstance("CloudHSM");
  String alias = ks.getCertificateAlias(cert);
  ```

- `getCertificateChain`

  Devuelve la cadena de certificados asociada con el alias especificado. Si el alias no existe o hace referencia a un objeto que no es un certificado, la función devuelve NULL.

- `getCreationDate`

Devuelve la fecha de creación de la entrada identificada por el alias especificado. Si no hay disponible ninguna fecha de creación, la función devuelve la fecha en la que el certificado pasó a ser válido.

- `getKey`

  GetKey se pasa al HSM y devuelve un objeto clave correspondiente a la etiqueta dada. Como consulta `getKey` directamente al HSM, se puede utilizar para cualquier clave del HSM, independientemente de si fue generada por el. KeyStore

  ```
  Key key = ks.getKey(keyLabel, null);
  ```

- `isCertificateEntry`

  Comprueba si la entrada con el alias especificado representa una entrada de certificado.

- `isKeyEntry`

  Comprueba si la entrada con el alias especificado representa una entrada de clave. La acción busca el alias tanto en el archivo PKCS12 como en el HSM.

- `setCertificateEntry`

  Asigna el certificado especificado al alias proporcionado. Si el alias proporcionado ya se utiliza para identificar una clave o un certificado, se inicia una excepción `KeyStoreException`. Puede utilizar el código JCE para obtener el objeto clave y, a continuación, utilizar el KeyStore `SetKeyEntry` método para asociar el certificado a la clave.

- `setKeyEntry` con una clave `byte[]`

  Actualmente, esta API no es compatible con SDK 5 de cliente.

- `setKeyEntry` con un objeto `Key`

  Asigna la clave especificada al alias proporcionado y la almacena dentro del HSM. Si la clave aún no existe en el HSM, se importará al HSM como una clave de sesión extraíble.

  Si el objeto `Key` es de tipo `PrivateKey`, debe ir acompañado de la cadena de certificados correspondiente.

  Si el alias ya existe, la llamada a `SetKeyEntry` inicia una excepción `KeyStoreException` y evita que la clave se sobrescriba. Si es necesario sobrescribir la clave, utilice KMU o JCE para ese propósito.

- `engineSize`

  Devuelve el número de entradas del almacén de claves.

- `store`

  Guarda el almacén de claves en el flujo de salida especificado como un archivo PKCS12 y lo protege con la contraseña proporcionada. Además, conserva todas las claves cargadas (que se establecen mediante llamadas a `setKey`).

Migre su proveedor de JCE de Client SDK 3 a Client SDK 5

Utilice este tema para migrar su proveedor de JCE del SDK de cliente 3 al SDK de cliente 5. Para obtener información sobre las ventajas de la migración, consulte. Ventajas del SDK 5 de cliente

En AWS CloudHSM, las aplicaciones de los clientes realizan operaciones criptográficas mediante el kit de desarrollo de software (SDK) para AWS CloudHSM clientes. El SDK de cliente 5 es el SDK principal al que se le siguen añadiendo nuevas funciones y compatibilidad con plataformas.

El proveedor de JCE del Client SDK 3 utiliza clases y API personalizadas que no forman parte de la especificación JCE estándar. El SDK de cliente 5 para el proveedor de JCE cumple con la especificación de JCE y, en determinadas áreas, es incompatible con versiones anteriores del SDK de cliente 3. Es posible que las aplicaciones del cliente requieran cambios como parte de la migración al SDK de cliente 5. En esta sección se describen los cambios necesarios para una migración satisfactoria.

Para revisar las instrucciones de migración de todos los proveedores, consulteMigración del SDK 3 de cliente al SDK 5 de cliente.

Temas

- Prepárese abordando los cambios más importantes
- Migre al SDK 5 de cliente
- Temas relacionados de

Prepárese abordando los cambios más importantes

Revise estos cambios importantes y actualice su aplicación en su entorno de desarrollo en consecuencia.

La clase y el nombre del proveedor han cambiado

¿Qué ha cambiado	¿Qué había en Client SDK 3	¿Qué hay en Client SDK 5	Ejemplo
Clase y nombre del proveedor	Se llama a la clase de proveedor JCE en Client SDK 3 `CaviumProvider` y tiene el nombre `Cavium` de proveedor .	En el SDK de cliente 5, se llama a la clase Provider `CloudHsmProvider` y tiene el nombre `CloudHSM` de proveedor.	En el repositorio de AWS CloudHSM GitHub muestras hay un ejemplo de cómo inicializar el `CloudHsmProvider` objeto.

El inicio de sesión explícito ha cambiado, el implícito no

¿Qué ha cambiado	¿Qué había en Client SDK 3	¿Qué hay en Client SDK 5	Ejemplo
Inicio de sesión explícito	El SDK 3 del cliente usa la `LoginManager` clase para el inicio de sesión explícito [1].	En el SDK de cliente 5, el `CloudHSM` proveedor implementa el inicio `AuthProvider` de sesión explícito. `AuthProvider` es una clase estándar de Java y sigue la forma idiomática de Java de iniciar sesión en un proveedor. Con la administración mejorada del estado de inicio de sesión en Client SDK 5, las aplicaciones ya no necesitan monitorea	Para ver un ejemplo sobre cómo utilizar el inicio de sesión explícito con Client SDK 5, consulte el LoginRunner ejemplo en el repositorio de ejemplos de AWS GitHub CloudHSM.

¿Qué ha cambiado	¿Qué había en Client SDK 3	¿Qué hay en Client SDK 5	Ejemplo
		r ni iniciar sesión durante las reconexiones. [2]	
Inicio de sesión implícito	No es necesario realizar cambios para el inicio de sesión implícito. El mismo archivo de propiedades y todas las variables de entorno seguirán funcionando para el inicio de sesión implícito al migrar del SDK de cliente 3 al SDK de cliente 5.		Para ver un ejemplo sobre cómo usar el inicio de sesión implícito con Client SDK 5, consulta el LoginRunner ejemplo en el AWS CloudHSM GitHub repositorio de ejemplos.

- [1] Fragmento de código del Client SDK 3:

```
LoginManager lm = LoginManager.getInstance();

lm.login(partition, user, pass);
```

- [2] Fragmento de código del SDK 5 del cliente:

```
// Construct or get the existing provider object
AuthProvider provider = new CloudHsmProvider();

// Call login method on the CloudHsmProvider object
// Here loginHandler is a CallbackHandler
provider.login(null, loginHandler);
```

Para ver un ejemplo sobre cómo utilizar el inicio de sesión explícito con Client SDK 5, consulta el LoginRunner ejemplo en el repositorio de AWS CloudHSM GitHub ejemplos.

La generación de claves ha cambiado

¿Qué ha cambiado	¿Qué había en Client SDK 3	¿Qué hay en Client SDK 5	Ejemplo
Generación de claves	En Client SDK 3, `Cavium[Key-type]AlgorithmParameterSpec` se utiliza para especificar los parámetros de generación de claves. Para ver un fragmento de código, consulte la nota a pie de página. [1]	En Client SDK 5, `KeyAttributesMap` se utiliza para especificar los atributos de generación de claves. Para ver un fragmento de código, consulte la nota a pie de página. [2]	Para ver un ejemplo de cómo se utiliza `KeyAttributesMap` para generar una clave simétrica, consulte el ejemplo en el repositorio de SymmetricKeys muestras de Github de AWS CloudHSM.
Generación de pares de claves	En el SDK de cliente 3, `Cavium[Key-type]Algorithmpa rameterSpec` se utiliza para especificar los parámetros de generación de pares de claves. Para ver un fragmento de código, consulte la nota a pie de página. [3]	En Client SDK 5, `KeyPairAttributesMap` se utiliza para especificar estos parámetros. Para ver un fragmento de código, consulte la nota a pie de página. [4]	Para ver un ejemplo sobre cómo `KeyAttributesMap` generar una clave asimétrica, consulta el ejemplo en el repositorio de AsymmetricKeys ejemplos. AWS CloudHSM GitHub

- [1] Fragmento de código de generación de claves del SDK 3 de Client:

```
KeyGenerator keyGen = KeyGenerator.getInstance("AES", "Cavium");
CaviumAESKeyGenParameterSpec aesSpec = new CaviumAESKeyGenParameterSpec(
keySizeInBits,
keyLabel,
```

```
isExtractable,
isPersistent);
keyGen.init(aesSpec);
SecretKey aesKey = keyGen.generateKey();
```

- [2] Fragmento de código de generación de claves del Client SDK 5:

```
KeyGenerator keyGen = KeyGenerator.getInstance("AES",
CloudHsmProvider.PROVIDER_NAME);

final KeyAttributesMap aesSpec = new KeyAttributesMap();
aesSpec.put(KeyAttribute.LABEL, keyLabel);
aesSpec.put(KeyAttribute.SIZE, keySizeInBits);
aesSpec.put(KeyAttribute.EXTRACTABLE, isExtractable);
aesSpec.put(KeyAttribute.TOKEN, isPersistent);

keyGen.init(aesSpec);
SecretKey aesKey = keyGen.generateKey();
```

- [3] Fragmento de código de generación de pares de claves de Client SDK 3:

```
KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("rsa", "Cavium");
CaviumRSAKeyGenParameterSpec spec = new CaviumRSAKeyGenParameterSpec(
keySizeInBits,
new BigInteger("65537"),
label + ":public",
label + ":private",
isExtractable,
isPersistent);

keyPairGen.initialize(spec);

keyPairGen.generateKeyPair();
```

- [4] Fragmento de código de generación de 5 pares de claves del SDK de cliente:

```
KeyPairGenerator keyPairGen =
KeyPairGenerator.getInstance("RSA", providerName);

// Set attributes for RSA public key
final KeyAttributesMap publicKeyAttrsMap = new KeyAttributesMap();
publicKeyAttrsMap.putAll(additionalPublicKeyAttributes);
publicKeyAttrsMap.put(KeyAttribute.LABEL, label + ":Public");
```

```
publicKeyAttrsMap.put(KeyAttribute.MODULUS_BITS, keySizeInBits);
publicKeyAttrsMap.put(KeyAttribute.PUBLIC_EXPONENT,
new BigInteger("65537").toByteArray());

// Set attributes for RSA private key
final KeyAttributesMap privateKeyAttrsMap = new KeyAttributesMap();
privateKeyAttrsMap.putAll(additionalPrivateKeyAttributes);
privateKeyAttrsMap.put(KeyAttribute.LABEL, label + ":Private");

// Create KeyPairAttributesMap and use that to initialize the
// keyPair generator
KeyPairAttributesMap keyPairSpec =
new KeyPairAttributesMapBuilder()
.withPublic(publicKeyAttrsMap)
.withPrivate(privateKeyAttrsMap)
.build();

keyPairGen.initialize(keyPairSpec);
keyPairGen.generateKeyPair();
```

Se han modificado las claves de búsqueda, eliminación y referencia

Para encontrar una clave ya generada, se AWS CloudHSM debe utilizar la KeyStore. El SDK 3 del cliente tiene dos KeyStore tipos: `Cavium` y`CloudHSM`. El SDK de cliente 5 solo tiene un KeyStore tipo:`CloudHSM`.

Pasar de un `Cavium` KeyStore a `CloudHSM` KeyStore otro requiere un cambio de KeyStore tipo. Además, el SDK de cliente 3 usa identificadores de teclas para hacer referencia a las claves, mientras que el SDK de cliente 5 usa etiquetas de clave. Los cambios de comportamiento resultantes se muestran a continuación.

¿Qué ha cambiado	¿Qué había en Client SDK 3	¿Qué hay en Client SDK 5	Ejemplo
Referencias clave	Con Client SDK 3, las aplicaciones utilizan etiquetas o identific adores de teclas para hacer referencia a las claves del HSM.	En Client SDK 5, las aplicaciones pueden Uso de la clase AWS CloudHSM KeyStore Java utilizarla para buscar claves por	

¿Qué ha cambiado	¿Qué había en Client SDK 3	¿Qué hay en Client SDK 5	Ejemplo
	Utilizan etiquetas KeyStore para encontrar una clave o utilizan identific adores para crear `CaviumKey` objetos.	etiqueta. Para buscar las claves por identific ador, usa la tecla AWS CloudHSM `KeyStoreW ithAttributes` with AWS CloudHSM `KeyRefereneSpec` .	
Búsqueda de múltiples entradas	Al buscar una clave utilizando `getEntrygetKey`, o `getCertificate` en situaciones en las que existan varios elementos con los mismos criterios `Cavium` KeyStore, solo se devolverá la primera entrada encontrada.	Con la AWS CloudHSM `KeyStore` tecla y`KeyStoreW ithAttrib utes` , en este mismo escenario , se generará una excepción. Para solucionar este problema, se recomienda establece r etiquetas únicas para las claves mediante el key set-attribute comando de la CLI de CloudHSM. O `KeyStoreW ithAttrib utes#getKeys` utilícelas para devolver todas las claves que coincidan con los criterios.	

¿Qué ha cambiado	¿Qué había en Client SDK 3	¿Qué hay en Client SDK 5	Ejemplo
Encuentra todas las claves	En el SDK de cliente 3 es posible encontrar todas las claves del HSM utilizando`Util.findAllKeys()` .	El Client SDK 5 simplifica y hace más eficiente la búsqueda de claves mediante el uso de la `KeyStoreWithAttributes` clase. Cuando sea posible, almacene sus claves en caché para minimizar la latencia. Para obtener más información, consulte Gestione eficazmente las claves de su aplicación.. Cuando necesite recuperar todas las claves del HSM, utilícelas `KeyStoreWithAttributes#getKeys` con una vacía`KeyAttributesMap` .	Un ejemplo en el que se usa la `KeyStoreWithAttributes` clase para buscar una clave está disponible en el repositorio de muestras de AWS CloudHSM Github y en él se muestra un fragmento de código. 1

¿Qué ha cambiado	¿Qué había en Client SDK 3	¿Qué hay en Client SDK 5	Ejemplo
Eliminación de claves	El SDK de cliente 3 se usa `Util.deleteKey()` para eliminar una clave.	El `Key` objeto del Client SDK 5 implementa la `Destroyable` interfaz que permite eliminar las claves mediante el `destroy()` método de esta interfaz.	Puede encontrar un código de ejemplo que muestra la funcionalidad de eliminar claves en el repositorio de ejemplos de GitHub de CloudHSM. Se muestra un fragmento de muestra para cada SDK en. 2

- [1] A continuación se muestra un fragmento:

```
KeyAttributesMap findSpec = new KeyAttributesMap();
findSpec.put(KeyAttribute.LABEL, label);
findSpec.put(KeyAttribute.KEY_TYPE, keyType);
KeyStoreWithAttributes keyStore = KeyStoreWithAttributes.getInstance("CloudHSM");

keyStore.load(null, null);
keyStore.getKey(findSpec);
```

- [2] Eliminar una clave en el SDK de cliente 3:

```
Util.deleteKey(key);
```

  Eliminar una clave en el SDK de cliente 5:

```
((Destroyable) key).destroy();
```

Las operaciones de desempaquetado de cifrado han cambiado, otras operaciones de cifrado no

> **Note**
>
> No es necesario realizar cambios en las operaciones de cifrado, descifrado y empaquetado de Cipher.

Las operaciones de desempaquetado requieren que la `CaviumUnwrapParameterSpec` clase Client SDK 3 se sustituya por una de las siguientes clases específicas para las operaciones criptográficas enumeradas.

- `GCMUnwrapKeySpec`para unwrap `AES/GCM/NoPadding`
- `IvUnwrapKeySpec`para y `AESWrap unwrap` `AES/CBC/NoPadding unwrap`
- `OAEPUnwrapKeySpec` para `RSA OAEP unwrap`

Fragmento de ejemplo para: `OAEPUnwrapkeySpec`

```
OAEPParameterSpec oaepParameterSpec =
new OAEPParameterSpec(
        "SHA-256",
        "MGF1",
        MGF1ParameterSpec.SHA256,
        PSpecified.DEFAULT);

KeyAttributesMap keyAttributesMap =
        new KeyAttributesMap(KeyAttributePermissiveProfile.KEY_CREATION);
keyAttributesMap.put(KeyAttribute.TOKEN, true);
keyAttributesMap.put(KeyAttribute.EXTRACTABLE, false);

OAEPUnwrapKeySpec spec = new OAEPUnwrapKeySpec(oaepParameterSpec,
        keyAttributesMap);

Cipher hsmCipher =
        Cipher.getInstance(
                "RSA/ECB/OAEPPadding",
                CloudHsmProvider.PROVIDER_NAME);
hsmCipher.init(Cipher.UNWRAP_MODE, key, spec);
```

Las operaciones de firma no han cambiado

No es necesario realizar cambios en las operaciones de firma.

Migre al SDK 5 de cliente

Siga las instrucciones de esta sección para migrar del SDK de cliente 3 al SDK de cliente 5.

> **Note**
>
> Amazon Linux, Ubuntu 16.04, Ubuntu 18.04, CentOS 6, CentOS 8 y RHEL 6 no son compatibles actualmente con el SDK de cliente 5. Si actualmente utiliza una de estas plataformas con el SDK de cliente 3, tendrá que elegir una plataforma diferente al migrar al SDK de cliente 5.

1. Desinstale el proveedor de JCE para Client SDK 3.

   Amazon Linux 2

   ```
   $ sudo yum remove cloudhsm-jce
   ```

   CentOS 7

   ```
   $ sudo yum remove cloudhsm-jce
   ```

   RHEL 7

   ```
   $ sudo yum remove cloudhsm-jce
   ```

   RHEL 8

   ```
   $ sudo yum remove cloudhsm-jce
   ```

2. Desinstale el Client Daemon para Client SDK 3.

   Amazon Linux 2

   ```
   $ sudo yum remove cloudhsm-client
   ```

CentOS 7

```
$ sudo yum remove cloudhsm-client
```

RHEL 7

```
$ sudo yum remove cloudhsm-client
```

RHEL 8

```
$ sudo yum remove cloudhsm-client
```

> **Note**
>
> Es necesario volver a habilitar las configuraciones personalizadas.

3. Instale el proveedor JCE del SDK de cliente siguiendo los pasos que se indican. Instalar y usar el proveedor AWS CloudHSM JCE para Client SDK 5
4. El Client SDK 5 presenta un nuevo formato de archivo de configuración y una nueva herramienta de arranque desde la línea de comandos. Para iniciar su proveedor de JCE para el SDK 5 de Client, siga las instrucciones que se indican en la guía del usuario que aparece a continuación. Proceso de arranque del SDK de cliente
5. En su entorno de desarrollo, pruebe la aplicación. Actualice el código existente para resolver los cambios importantes antes de la migración final.

Temas relacionados de

- Mejores prácticas para AWS CloudHSM

Configuraciones avanzadas para JCE

El proveedor de AWS CloudHSM JCE incluye las siguientes configuraciones avanzadas, que no forman parte de las configuraciones generales que utilizan la mayoría de los clientes.

- Conexión a múltiples clústeres

- Extracción de claves mediante JCE

- Vuelva a intentar la configuración del JCE.

Conexión a varios clústeres con el proveedor de JCE

Esta configuración permite que una sola instancia de cliente se comunique con varios clústeres. En comparación con tener una sola instancia que solo se comunique con un único clúster, esta característica puede suponer un ahorro de costos en algunos casos de uso. La `CloudHsmProvider` clase AWS CloudHSM es la implementación de la clase Provider de Java Security. Cada instancia de esta clase representa una conexión a todo el AWS CloudHSM clúster. Cree una instancia de esta clase y añádala a la lista de proveedores de seguridad de Java para poder interactuar con ella mediante clases de JCE estándar.

En el siguiente ejemplo, se crea una instancia de esta clase y se añade a la lista de proveedores de seguridad de Java:

```
if (Security.getProvider(CloudHsmProvider.PROVIDER_NAME) == null) {
    Security.addProvider(new CloudHsmProvider());
}
```

Configuración de **`CloudHsmProvider`**

`CloudHsmProvider` puede configurarse de dos formas:

1. Configurar con un archivo (configuración predeterminada)
2. Configuración mediante código

Configurar con un archivo (configuración predeterminada)

Al iniciar `CloudHsmProvider` con el constructor predeterminado, de forma predeterminada buscará el archivo de configuración en la ruta `/opt/cloudhsm/etc/cloudhsm-jce.cfg` en Linux. Este archivo de configuración se puede configurar usando `configure-jce`.

Un objeto creado con el constructor predeterminado utilizará el nombre de proveedor de CloudHSM predeterminado `CloudHSM`. El nombre del proveedor es útil para interactuar con el JCE y decirle qué proveedor debe utilizar para las distintas operaciones. A continuación, se muestra un ejemplo de cómo utilizar el nombre del proveedor de CloudHSM para la operación de cifrado:

```
Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding", "CloudHSM");
```

Configuración mediante código

A partir de la versión 5.8.0 del SDK de cliente también puede configurar el `CloudHsmProvider` mediante código Java. La forma de hacerlo es utilizando un objeto de clase `CloudHsmProviderConfig`. Puede construir este objeto utilizando `CloudHsmProviderConfigBuilder`.

`CloudHsmProvider` tiene otro constructor que toma el objeto `CloudHsmProviderConfig`, como se muestra en el siguiente ejemplo.

Example

```
CloudHsmProviderConfig config = CloudHsmProviderConfig.builder()
                                    .withCluster(
                                        CloudHsmCluster.builder()
                                            .withHsmCAFilePath(hsmCAFilePath)

 .withClusterUniqueIdentifier("CloudHsmCluster1")
        .withServer(CloudHsmServer.builder().withHostIP(hostName).build())
                        .build())
        .build();
CloudHsmProvider provider = new CloudHsmProvider(config);
```

En este ejemplo, el nombre del proveedor de JCE es `CloudHsmCluster1`. Este es el nombre que la aplicación puede utilizar para interactuar con el JCE:

Example

```
Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding", "CloudHsmCluster1");
```

Como alternativa, las aplicaciones también pueden usar el objeto de proveedor creado anteriormente para que JCE sepa que debe usar ese proveedor para la operación:

```
Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding", provider);
```

Si no se especifica un identificador único con el método de `withClusterUniqueIdentifier`, se crea para usted un nombre de proveedor generado aleatoriamente. Para obtener este identificador

generado aleatoriamente, las aplicaciones pueden llamar `provider.getName()` para obtener el identificador.

Conexión a múltiples clústeres

Como se indicó anteriormente, cada `CloudHsmProvider` representa una conexión a su clúster de CloudHSM. Si desea comunicarse con otro clúster desde la misma aplicación, puede crear otro objeto de `CloudHsmProvider` con las configuraciones del otro clúster e interactuar con este otro clúster mediante el objeto del proveedor o el nombre del proveedor, como se muestra en el siguiente ejemplo.

Example

```
CloudHsmProviderConfig config = CloudHsmProviderConfig.builder()
                                    .withCluster(
                                        CloudHsmCluster.builder()
                                            .withHsmCAFilePath(hsmCAFilePath)

 .withClusterUniqueIdentifier("CloudHsmCluster1")
        .withServer(CloudHsmServer.builder().withHostIP(hostName).build())
                        .build())
        .build();
CloudHsmProvider provider1 = new CloudHsmProvider(config);

if (Security.getProvider(provider1.getName()) == null) {
    Security.addProvider(provider1);
}

CloudHsmProviderConfig config2 = CloudHsmProviderConfig.builder()
                                    .withCluster(
                                        CloudHsmCluster.builder()
                                            .withHsmCAFilePath(hsmCAFilePath2)

 .withClusterUniqueIdentifier("CloudHsmCluster2")
        .withServer(CloudHsmServer.builder().withHostIP(hostName2).build())
                        .build())
        .build();
CloudHsmProvider provider2 = new CloudHsmProvider(config2);

if (Security.getProvider(provider2.getName()) == null) {
    Security.addProvider(provider2);
}
```

Una vez que haya configurado los dos proveedores (ambos clústeres) anteriores, podrá interactuar con ellos mediante el objeto del proveedor o mediante el nombre del proveedor.

Si ampliamos este ejemplo que muestra cómo hablar con nosotros`cluster1`, podría utilizar el siguiente ejemplo para una operación NoPadding AES/GCM/:

```
Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding", provider1);
```

Y en la misma aplicación para generar la clave «AES» en el segundo clúster con el nombre del proveedor, también puede usar el siguiente ejemplo:

```
Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding", provider2.getName());
```

Comandos de reintento para JCE

La versión 5.8.0 y posteriores de SDK de cliente tienen una estrategia de reintento automático integrada que reintentará las operaciones limitadas por HSM desde el lado del cliente. Cuando un HSM limita las operaciones porque está demasiado ocupado realizando operaciones anteriores y no puede aceptar más solicitudes, los SDK de cliente intentarán reintentar las operaciones limitadas hasta 3 veces y, al mismo tiempo, se retrasarán exponencialmente. Esta estrategia de reintento automático se puede configurar en uno de estos dos modos: desactivado y estándar.

- desactivado: el SDK de cliente no realizará ninguna estrategia de reintentos para ninguna operación limitada por parte del HSM.
- estándar: este es el modo predeterminado para la versión 5.8.0 y posteriores de SDK de cliente. En este modo, los SDK de cliente reintentarán ejecutar automáticamente las operaciones restringidas y se reducirán exponencialmente.

Para obtener más información, consulte Limitación de HSM.

Configuración de los comandos de reintento en modo desactivado

Linux

Cómo configurar los comandos de reintento en off para SDK 5 de cliente en Linux

- Puede utilizar los siguientes comandos para administrar la configuración de reintento en modo off:

```
$ sudo /opt/cloudhsm/bin/configure-jce --default-retry-mode off
```

Windows

Cómo configurar los comandos de reintento en off para SDK 5 de cliente en Windows

- Puede utilizar los siguientes comandos para administrar la configuración de reintento en modo off:

```
C:\Program Files\Amazon\CloudHSM\bin\ .\configure-jce.exe --default-retry-mode
 off
```

Extracción de claves mediante JCE

La extensión de criptografía de Java (JCE) utiliza una arquitectura que permite conectar diferentes implementaciones de criptografía. AWS CloudHSM envía uno de esos proveedores de JCE que descarga las operaciones criptográficas al HSM. Para que la mayoría de los demás proveedores de JCE trabajen con claves almacenadas en AWS CloudHSM, deben extraer los bytes de clave de los HSM en texto claro y guardarlos en la memoria de la máquina para su uso. Por lo general, los HSM solo permiten extraer las claves como objetos encapsulados, no como texto transparente. Sin embargo, para respaldar los casos de uso de la integración entre proveedores, AWS CloudHSM permite una opción de configuración opcional que permite extraer los bytes clave de forma transparente.

> **Important**
>
> JCE descarga las operaciones a AWS CloudHSM cada vez que se especifique el proveedor de AWS CloudHSM o se utilice un objeto clave. AWS CloudHSM No necesita extraer las claves sin cifrar si espera que la operación se lleve a cabo dentro del HSM. La extracción de claves en texto no cifrado solo es necesaria cuando la aplicación no puede utilizar mecanismos seguros, como encapsular y desencapsular una clave, debido a las restricciones de una biblioteca externa o de un proveedor de JCE.

De forma predeterminada, el proveedor de AWS CloudHSM JCE permite la extracción de claves públicas para que funcionen con proveedores de JCE externos. Siempre se permiten los siguientes métodos:

Clase	Método	Formato (GetEncoded)
EcPublicKey	getEncoded()	X.509
	getW()	N/A
RSA PublicKey	getEncoded()	X.509
	getPublicExponent()	N/A
CloudHsmRsaPrivateCrtKey	getPublicExponent()	N/A

De forma predeterminada, el proveedor de AWS CloudHSM JCE no permite la extracción de los bytes de clave en blanco para las claves privadas o secretas. Si su caso de uso lo requiere, puede habilitar la extracción de los bytes de claves sin cifrar para claves privadas o secretas en las siguientes condiciones:

1. El atributo `EXTRACTABLE` para claves privadas y secretas se establece como true.
   - De forma predeterminada, el atributo `EXTRACTABLE` de las claves privadas y secretas está establecido como true. Las claves de `EXTRACTABLE` son claves que se pueden exportar fuera del HSM. Para obtener más información, consulte Atributos de Java admitidos para SDK 5 de cliente.
2. El atributo `WRAP_WITH_TRUSTED` para claves privadas y secretas se establece en false.
   - `getEncoded`, `getPrivateExponent` y `getS` no se pueden usar con claves privadas que no se puedan exportar sin cifrar. `WRAP_WITH_TRUSTED` no permite exportar sus claves privadas fuera del HSM de sin cifrar. Para obtener más información, consulte Cómo usar claves de confianza para controlar el desencapsulamiento de claves.

Permitir que el proveedor de AWS CloudHSM JCE extraiga secretos de claves privadas de AWS CloudHSM

> **Important**
>
> Este cambio de configuración permite extraer todos los bytes de claves `EXTRACTABLE` sin procesar del clúster de HSM. Para mejorar la seguridad, debería considerar la posibilidad de utilizar métodos de encapsulamiento de claves para extraer la clave del HSM de forma segura. Esto evita la extracción involuntaria de los bytes clave del HSM.

1. Utilice los siguientes comandos para permitir que sus claves privadas o secretas se extraigan en JCE:

   Linux

   ```
   $ /opt/cloudhsm/bin/configure-jce --enable-clear-key-extraction-in-software
   ```

   Windows

   ```
   C:\Program Files\Amazon\CloudHSM\> .\configure-jce.exe --enable-clear-key-extraction-in-software
   ```

2. Una vez que habilite la extracción de claves sin cifrar, se habilitarán los siguientes métodos para extraer las claves privadas de la memoria.

Clase	Método	Formato (GetEncoded)
Clave	getEncoded()	RAW
EC PrivateKey	getEncoded()	PKCS #8
	getS()	N/A
RSA PrivateCrtKey	getEncoded()	X.509
	getPrivateExponent()	N/A

Clase	Método	Formato (GetEncoded)
	getPrimeP()	N/A
	getPrimeQ()	N/A
	getPrimeExponent(P)	N/A
	getPrimeExponentQ ()	N/A
	getCrtCoefficient()	N/A

Si quiere restaurar el comportamiento predeterminado y no permitir que JCE exporte las claves en formato sin cifrar, ejecute el siguiente comando:

Linux

```
$ /opt/cloudhsm/bin/configure-jce --disable-clear-key-extraction-in-software
```

Windows

```
C:\Program Files\Amazon\CloudHSM\> .\configure-jce.exe --disable-clear-key-
extraction-in-software
```

API de criptografía: proveedores de próxima generación (CNG) y de almacenamiento de claves (KSP) para Microsoft Windows

El AWS CloudHSM cliente para Windows incluye proveedores de GNC y KSP. Actualmente, solo SDK 3 de cliente es compatible con los proveedores de GNC y KSP.

Los proveedores de almacenamiento de claves (KSP) permiten el almacenamiento y la recuperación de claves. Por ejemplo, si agrega el rol Servicios de certificados de Microsoft Active Directory (AD CS) a su servidor de Windows y elige crear una nueva clave privada de su entidad de certificación (CA), puede elegir el KSP que administrará el almacenamiento de claves. Al configurar el rol AD CS,

puede elegir este KSP. Para obtener más información, consulte Crear entidad de certificación de Windows Server.

API de criptografía: nueva generación (CNG) es una API criptográfica específica del sistema operativo Microsoft Windows. CNG permite a los desarrolladores utilizar técnicas criptográficas para proteger las aplicaciones basadas en Windows. A un alto nivel, la AWS CloudHSM implementación del GNC proporciona las siguientes funcionalidades:

- Primitivas criptográficas: le permiten realizar operaciones criptográficas fundamentales.
- Importación y exportación de claves: le permiten importar y exportar claves simétricas y asimétricas.
- API de protección de datos (CNG DPAPI): le permite cifrar y descifrar datos fácilmente.
- Almacenamiento y recuperación de claves: le permite almacenar y aislar de forma segura la clave privada de un par de claves asimétricas.

Temas

- Instalación de los proveedores de KSP y CNG para Windows
- AWS CloudHSM Requisitos previos de Windows
- Asociar una AWS CloudHSM clave a un certificado
- Ejemplo de código para proveedor CNG

Instalación de los proveedores de KSP y CNG para Windows

Los proveedores KSP y CNG se instalan al instalar el cliente de Windows AWS CloudHSM . Puede instalar el cliente siguiendo los pasos que se indican en Instalación del cliente (Windows).

Configuración y ejecución del cliente de AWS CloudHSM para Windows

Para iniciar el cliente de CloudHSM para Windows, debe cumplir los Requisitos previos. Posteriormente, debe actualizar los archivos de configuración que utilizan los proveedores e iniciar el cliente siguiendo los pasos que se indican a continuación. Tiene que realizar estos pasos la primera vez que utilice los proveedores de KSP y CNG y después de añadir o quitar HSM del clúster. De esta forma, AWS CloudHSM puede sincronizar los datos y mantener la coherencia en todos los HSM del clúster.

Paso 1: Detenga el cliente AWS CloudHSM

Antes de actualizar los archivos de configuración que utilizan los proveedores, detenga el AWS CloudHSM cliente. Si el cliente ya se ha detenido, la ejecución del comando stop no tiene ningún efecto.

- Para la versión 1.1.2 y posteriores del cliente de Windows:

```
C:\Program Files\Amazon\CloudHSM>net.exe stop AWSCloudHSMClient
```

- Para la versión 1.1.1 y anteriores de clientes de Windows:

  Use Ctrl + C en la ventana de comandos donde inició el AWS CloudHSM cliente.

Paso 2: actualice los archivos de AWS CloudHSM configuración

En este paso se utiliza el parámetro -a de la herramienta de configuración para añadir la dirección IP de la interfaz de red elástica (ENI) de uno de los HSM del clúster al archivo de configuración.

```
C:\Program Files\Amazon\CloudHSM configure.exe -a <HSM ENI IP>
```

Para obtener la dirección IP ENI de un HSM de su clúster, vaya a la AWS CloudHSM consola, elija los clústeres y seleccione el clúster deseado. También puede usar la DescribeClustersoperación, el comando describe-clusters o el cmdlet. Get-HSM2Cluster PowerShell Escriba una sola dirección IP de ENI. No importa qué dirección IP de ENI use.

Paso 3: iniciar el cliente AWS CloudHSM

A continuación, inicie o reinicie el AWS CloudHSM cliente. Cuando el AWS CloudHSM cliente se inicia, utiliza la dirección IP de ENI en su archivo de configuración para consultar el clúster. A continuación, añade las direcciones IP de ENI de todos los HSM en el clúster al archivo de información del clúster.

- Para la versión 1.1.2 y posteriores del cliente de Windows:

```
C:\Program Files\Amazon\CloudHSM>net.exe start AWSCloudHSMClient
```

- Para la versión 1.1.1 y anteriores de clientes de Windows:

```
C:\Program Files\Amazon\CloudHSM>start "cloudhsm_client" cloudhsm_client.exe C:
\ProgramData\Amazon\CloudHSM\data\cloudhsm_client.cfg
```

Comprobación de los proveedores de KSP y CNG

Puede utilizar cualquiera de los siguientes comandos para determinar qué proveedores están instalados en su sistema. Los comandos enumeran los proveedores de KSP y CNG registrados. No es necesario que el cliente de AWS CloudHSM esté en ejecución.

```
C:\Program Files\Amazon\CloudHSM>ksp_config.exe -enum
```

```
C:\Program Files\Amazon\CloudHSM>cng_config.exe -enum
```

Compruebe que los proveedores de KSP y CNG están instalados en la instancia EC2 de Windows Server. Debería ver las siguientes entradas en la lista:

```
Cavium CNG Provider
Cavium Key Storage Provider
```

Si falta el proveedor de CNG, ejecute el siguiente comando.

```
C:\Program Files\Amazon\CloudHSM>cng_config.exe -register
```

Si falta el proveedor de KSP, ejecute el siguiente comando.

```
C:\Program Files\Amazon\CloudHSM>ksp_config.exe -register
```

AWS CloudHSM Requisitos previos de Windows

Antes de poder iniciar el AWS CloudHSM cliente de Windows y utilizar los proveedores de KSP y CNG, debe configurar las credenciales de inicio de sesión del HSM en su sistema. Puede establecer las credenciales a través del Administrador de credenciales de Windows o de una variable de entorno del sistema. Le recomendamos que utilice el Administrador de credenciales de Windows para almacenar las credenciales. Esta opción está disponible con la versión 2.0.4 y AWS CloudHSM posteriores del cliente. La variable de entorno es más fácil de configurar, pero es menos segura que el Administrador de credenciales de Windows.

Administrador de credenciales de Windows

Puede emplear la utilidad `set_cloudhsm_credentials` o la interfaz del Administrador de credenciales de Windows.

- Con la utilidad **`set_cloudhsm_credentials`**:

  La utilidad `set_cloudhsm_credentials` está incluida en el instalador de Windows. Puede emplear esta utilidad para transferir cómodamente las credenciales de inicio de sesión de HSM al Administrador de credenciales de Windows. Si desea compilar esta utilidad a partir del código fuente, puede utilizar el código Python que se incluye en el instalador.

  1. Vaya a la carpeta `C:\Program Files\Amazon\CloudHSM\tools\`.
  2. Ejecute el archivo `set_cloudhsm_credentials.exe` con los parámetros de nombre de usuario y contraseña del CU.

     ```
     set_cloudhsm_credentials.exe --username <CU USER> --password <CU PASSWORD>
     ```

- Con la interfaz del Administrador de credenciales:

  Puede utilizar la interfaz del Administrador de credenciales para administrar manualmente las credenciales.

  1. Para abrir el Administrador de credenciales, escriba `credential manager` en el cuadro de búsqueda de la barra de tareas y seleccione Credential Manager (Administrador de credenciales).
  2. Seleccione Windows Credentials (Credenciales de Windows) para administrar las credenciales de Windows.
  3. Seleccione Add a generic credential (Agregar una credencial genérica) y rellene los detalles del modo siguiente:
     - En Internet o Network Address (Dirección de red o Internet), escriba el nombre de destino como `cloudhsm_client`.
     - En Username (Nombre de usuario) y Password (Contraseña), escriba las credenciales del CU.
     - Haga clic en OK (Aceptar).

Variables de entorno del sistema.

Puede configurar unas variables de entorno del sistema que identifiquen un HSM y un usuario de criptografía (CU) para su aplicación de Windows. Puede usar el comando setx para establecer variables de entorno del sistema, o bien establecer variables de entorno del sistema permanentes mediante programación o en la pestaña Opciones avanzadas del panel de control Propiedades del sistema de Windows.

> **Warning**
>
> Si establece las credenciales utilizando variables de entorno del sistema, la contraseña estará disponible como texto no cifrado en el sistema de los usuarios. Para solucionar este problema, utilice el Administrador de credenciales de Windows.

Configure las siguientes variables de entorno del sistema:

`n3fips_password=`*`CU USERNAME`*`:`*`CU PASSWORD`*

Identifica un usuario de criptografía (CU) en el HSM y proporciona toda la información de inicio de sesión necesaria. Su aplicación se autentica y ejecuta como este CU. La aplicación tiene los permisos de este CU y puede ver y administrar solo las claves que el CU posee y comparte. Para crear un nuevo CU, use createUser. Para encontrar los CU existentes, use listUsers.

Por ejemplo:

```
setx /m n3fips_password test_user:password123
```

Asociar una AWS CloudHSM clave a un certificado

Antes de poder utilizar AWS CloudHSM las claves con herramientas de terceros, como las de Microsoft SignTool, debe importar los metadatos de la clave al almacén de certificados local y asociarlos a un certificado. Para importar los metadatos de la clave, emplee la utilidad import_key.exe que se incluye en CloudHSM 3.0 y versiones posteriores. En los pasos siguientes, encontrará más información y un ejemplo de la salida.

Paso 1: Importar el certificado

En Windows, debería poder hacer doble clic en el certificado para importarlo en el almacén de certificados local.

Sin embargo, si el doble clic no funciona, utilice la herramienta Microsoft Certreq para importar el certificado en el administrador de certificados. Por ejemplo:

```
certreq -accept certificatename
```

Si esta acción no se realiza correctamente y aparece el error `Key not found`, continúe en el paso 2. Si el certificado aparece en el almacén de claves, la tarea se habrá completado correctamente y no será necesario realizar más acciones.

Paso 2: Recopilar información de identificación del certificado

Si el paso anterior no se realizó correctamente, deberá asociar la clave privada con un certificado. Sin embargo, antes de poder establecer esta asociación deberá buscar primero el nombre único del contenedor y el número de serie del certificado. Utilice una utilidad, como certutil, para mostrar la información necesaria del certificado. En el siguiente ejemplo de la salida de certutil, se muestra el nombre del contenedor y el número de serie.

```
================ Certificate 1 ================ Serial Number:
   72000000047f7f7a9d41851b4e000000000004Issuer: CN=Enterprise-CANotBefore: 10/8/2019
 11:50
   AM NotAfter: 11/8/2020 12:00 PMSubject: CN=www.example.com, OU=Certificate
 Management,
   O=Information Technology, L=Seattle, S=Washington, C=USNon-root CertificateCert
   Hash(sha1): 7f d8 5c 00 27 bf 37 74 3d 71 5b 54 4e c0 94 20 45 75 bc 65No key
 provider
   information Simple container name: CertReq-39c04db0-6aa9-4310-93db-db0d9669f42c
 Unique
   container name: CertReq-39c04db0-6aa9-4310-93db-db0d9669f42c
```

Paso 3: Asocie la clave AWS CloudHSM privada al certificado

Para asociar la clave al certificado, primero asegúrese de iniciar el daemon del AWS CloudHSM cliente. A continuación, utilice import_key.exe (que se incluye en CloudHSM 3.0 y versiones posteriores) para asociar la clave privada con el certificado. Cuando especifique el certificado, utilice

el nombre simple de contenedor. En el ejemplo siguiente, se muestra el comando y la respuesta. Esta acción solo copia los metadatos de la clave; la clave permanece en el HSM.

```
$> import_key.exe -RSA CertReq-39c04db0-6aa9-4310-93db-db0d9669f42c

Successfully opened Microsoft Software Key Storage Provider : 0NCryptOpenKey failed :
 80090016
```

Paso 4: Actualizar el almacén de certificados

Asegúrese de que el daemon del AWS CloudHSM cliente sigue ejecutándose. A continuación, utilice el verbo -repairstore de certutil para actualizar el número de serie del certificado. En el siguiente ejemplo, se muestra el comando y la salida. Consulte la documentación de Microsoft para obtener información sobre el verbo -repairstore.

```
C:\Program Files\Amazon\CloudHSM>certutil -f -csp "Cavium Key Storage Provider"-
repairstore my "72000000047f7f7a9d41851b4e000000000004"
my "Personal"
================ Certificate 1 ================
Serial Number: 72000000047f7f7a9d41851b4e000000000004
Issuer: CN=Enterprise-CA
NotBefore: 10/8/2019 11:50 AM
NotAfter: 11/8/2020 12:00 PM
Subject: CN=www.example.com, OU=Certificate Management, O=Information Technology,
 L=Seattle, S=Washington, C=US
Non-root CertificateCert Hash(sha1): 7f d8 5c 00 27 bf 37 74 3d 71 5b 54 4e c0 94 20 45
 75 bc 65
SDK Version: 3.0
Key Container = CertReq-39c04db0-6aa9-4310-93db-db0d9669f42c
Provider = Cavium Key Storage ProviderPrivate key is NOT exportableEncryption test
 passedCertUtil: -repairstore command completed successfully.
```

Tras actualizar el número de serie del certificado, puede utilizar este certificado y la clave AWS CloudHSM privada correspondiente con cualquier herramienta de firma de terceros en Windows.

Ejemplo de código para proveedor CNG

⚠ ** Código exclusivamente de ejemplo: no se debe utilizar con fines de producción **

Este ejemplo solo tiene fines ilustrativos. No ejecute este código en un entorno de producción.

El siguiente ejemplo muestra cómo enumerar los proveedores de servicios criptográficos registrados en su sistema para encontrar el proveedor CNG instalado con el cliente CloudHSM para Windows. En la muestra también se presenta cómo crear un par de claves asimétricas y cómo utilizar el par de claves para firmar datos.

Important

Antes de ejecutar este ejemplo, debe configurar las credenciales del HSM, tal y como se explica en los requisitos previos. Para obtener más información, consulte AWS CloudHSM Requisitos previos de Windows.

```
// CloudHsmCngExampleConsole.cpp : Console application that demonstrates CNG
 capabilities.
// This example contains the following functions.
//
//   VerifyProvider()          - Enumerate the registered providers and retrieve Cavium
 KSP and CNG providers.
//   GenerateKeyPair()         - Create an RSA key pair.
//   SignData()                - Sign and verify data.
//

#include "stdafx.h"
#include <Windows.h>

#ifndef NT_SUCCESS
#define NT_SUCCESS(Status) ((NTSTATUS)(Status) >= 0)
#endif

#define CAVIUM_CNG_PROVIDER L"Cavium CNG Provider"
#define CAVIUM_KEYSTORE_PROVIDER L"Cavium Key Storage Provider"

// Enumerate the registered providers and determine whether the Cavium CNG provider
// and the Cavium KSP provider exist.
//
```

```
bool VerifyProvider()
{
  NTSTATUS status;
  ULONG cbBuffer = 0;
  PCRYPT_PROVIDERS pBuffer = NULL;
  bool foundCng = false;
  bool foundKeystore = false;

  // Retrieve information about the registered providers.
  //   cbBuffer - the size, in bytes, of the buffer pointed to by pBuffer.
  //   pBuffer - pointer to a buffer that contains a CRYPT_PROVIDERS structure.
  status = BCryptEnumRegisteredProviders(&cbBuffer, &pBuffer);

  // If registered providers exist, enumerate them and determine whether the
  // Cavium CNG provider and Cavium KSP provider have been registered.
  if (NT_SUCCESS(status))
  {
    if (pBuffer != NULL)
    {
      for (ULONG i = 0; i < pBuffer->cProviders; i++)
      {
        // Determine whether the Cavium CNG provider exists.
        if (wcscmp(CAVIUM_CNG_PROVIDER, pBuffer->rgpszProviders[i]) == 0)
        {
          printf("Found %S\n", CAVIUM_CNG_PROVIDER);
          foundCng = true;
        }

        // Determine whether the Cavium KSP provider exists.
        else if (wcscmp(CAVIUM_KEYSTORE_PROVIDER, pBuffer->rgpszProviders[i]) == 0)
        {
          printf("Found %S\n", CAVIUM_KEYSTORE_PROVIDER);
          foundKeystore = true;
        }
      }
    }
  }
  else
  {
    printf("BCryptEnumRegisteredProviders failed with error code 0x%08x\n", status);
  }

  // Free memory allocated for the CRYPT_PROVIDERS structure.
  if (NULL != pBuffer)
```

```
  {
    BCryptFreeBuffer(pBuffer);
  }

  return foundCng == foundKeystore == true;
}

// Generate an asymmetric key pair. As used here, this example generates an RSA key
 pair
// and returns a handle. The handle is used in subsequent operations that use the key
 pair.
// The key material is not available.
//
// The key pair is used in the SignData function.
//
NTSTATUS GenerateKeyPair(BCRYPT_ALG_HANDLE hAlgorithm, BCRYPT_KEY_HANDLE *hKey)
{
  NTSTATUS status;

  // Generate the key pair.
  status = BCryptGenerateKeyPair(hAlgorithm, hKey, 2048, 0);
  if (!NT_SUCCESS(status))
  {
    printf("BCryptGenerateKeyPair failed with code 0x%08x\n", status);
    return status;
  }

  // Finalize the key pair. The public/private key pair cannot be used until this
  // function is called.
  status = BCryptFinalizeKeyPair(*hKey, 0);
  if (!NT_SUCCESS(status))
  {
    printf("BCryptFinalizeKeyPair failed with code 0x%08x\n", status);
    return status;
  }

  return status;
}

// Sign and verify data using the RSA key pair. The data in this function is hardcoded
// and is for example purposes only.
//
NTSTATUS SignData(BCRYPT_KEY_HANDLE hKey)
{
```

```
  NTSTATUS status;
  PBYTE sig;
  ULONG sigLen;
  ULONG resLen;
  BCRYPT_PKCS1_PADDING_INFO pInfo;

  // Hardcode the data to be signed (for demonstration purposes only).
  PBYTE message = (PBYTE)"d83e7716bed8a20343d8dc6845e57447";
  ULONG messageLen = strlen((char*)message);

  // Retrieve the size of the buffer needed for the signature.
  status = BCryptSignHash(hKey, NULL, message, messageLen, NULL, 0, &sigLen, 0);
  if (!NT_SUCCESS(status))
  {
    printf("BCryptSignHash failed with code 0x%08x\n", status);
    return status;
  }

  // Allocate a buffer for the signature.
  sig = (PBYTE)HeapAlloc(GetProcessHeap(), 0, sigLen);
  if (sig == NULL)
  {
    return -1;
  }

  // Use the SHA256 algorithm to create padding information.
  pInfo.pszAlgId = BCRYPT_SHA256_ALGORITHM;

  // Create a signature.
  status = BCryptSignHash(hKey, &pInfo, message, messageLen, sig, sigLen, &resLen,
BCRYPT_PAD_PKCS1);
  if (!NT_SUCCESS(status))
  {
    printf("BCryptSignHash failed with code 0x%08x\n", status);
    return status;
  }

  // Verify the signature.
  status = BCryptVerifySignature(hKey, &pInfo, message, messageLen, sig, sigLen,
BCRYPT_PAD_PKCS1);
  if (!NT_SUCCESS(status))
  {
    printf("BCryptVerifySignature failed with code 0x%08x\n", status);
    return status;
```

```
  }

  // Free the memory allocated for the signature.
  if (sig != NULL)
  {
    HeapFree(GetProcessHeap(), 0, sig);
    sig = NULL;
  }

  return 0;
}

// Main function.
//
int main()
{
  NTSTATUS status;
  BCRYPT_ALG_HANDLE hRsaAlg;
  BCRYPT_KEY_HANDLE hKey = NULL;

  // Enumerate the registered providers.
  printf("Searching for Cavium providers...\n");
  if (VerifyProvider() == false) {
    printf("Could not find the CNG and Keystore providers\n");
    return 1;
  }

  // Get the RSA algorithm provider from the Cavium CNG provider.
  printf("Opening RSA algorithm\n");
  status = BCryptOpenAlgorithmProvider(&hRsaAlg, BCRYPT_RSA_ALGORITHM,
 CAVIUM_CNG_PROVIDER, 0);
  if (!NT_SUCCESS(status))
  {
    printf("BCryptOpenAlgorithmProvider RSA failed with code 0x%08x\n", status);
    return status;
  }

  // Generate an asymmetric key pair using the RSA algorithm.
  printf("Generating RSA Keypair\n");
  GenerateKeyPair(hRsaAlg, &hKey);
  if (hKey == NULL)
  {
    printf("Invalid key handle returned\n");
    return 0;
```

```
  }
  printf("Done!\n");

  // Sign and verify [hardcoded] data using the RSA key pair.
  printf("Sign/Verify data with key\n");
  SignData(hKey);
  printf("Done!\n");

  // Remove the key handle from memory.
  status = BCryptDestroyKey(hKey);
  if (!NT_SUCCESS(status))
  {
    printf("BCryptDestroyKey failed with code 0x%08x\n", status);
    return status;
  }

  // Close the RSA algorithm provider.
  status = BCryptCloseAlgorithmProvider(hRsaAlg, NULL);
  if (!NT_SUCCESS(status))
  {
    printf("BCryptCloseAlgorithmProvider RSA failed with code 0x%08x\n", status);
    return status;
  }

  return 0;
}
```

SDK de cliente anterior (SDK de cliente 3)

AWS CloudHSM incluye dos versiones principales del SDK de cliente:

- SDK 5 de cliente: este es nuestro SDK de cliente más reciente y predeterminado. Para obtener información sobre los beneficios y las ventajas que ofrece, consulte Ventajas del SDK 5 de cliente.
- SDK 3 de cliente: este es nuestro SDK de cliente anterior. Incluye un completo paquete de componentes para la compatibilidad de aplicaciones basadas en lenguaje y plataforma, así como herramientas de gestión.

Para obtener instrucciones sobre cómo migrar del SDK de cliente 3 al SDK de cliente 5, consulteMigración del SDK 3 de cliente al SDK 5 de cliente.

Este tema incluye la documentación de SDK 3 de cliente.

Para descargar, vea Descargas.

Compruebe su versión de SDK de cliente.

Amazon Linux

Utilice el siguiente comando:

```
rpm -qa | grep ^cloudhsm
```

Amazon Linux 2

Utilice el siguiente comando:

```
rpm -qa | grep ^cloudhsm
```

CentOS 6

Utilice el siguiente comando:

```
rpm -qa | grep ^cloudhsm
```

CentOS 7

Utilice el siguiente comando:

```
rpm -qa | grep ^cloudhsm
```

CentOS 8

Utilice el siguiente comando:

```
rpm -qa | grep ^cloudhsm
```

RHEL 6

Utilice el siguiente comando:

```
rpm -qa | grep ^cloudhsm
```

RHEL 7

Utilice el siguiente comando:

```
rpm -qa | grep ^cloudhsm
```

RHEL 8

Utilice el siguiente comando:

```
rpm -qa | grep ^cloudhsm
```

Ubuntu 16.04 LTS

Utilice el siguiente comando:

```
apt list --installed | grep ^cloudhsm
```

Ubuntu 18.04 LTS

Utilice el siguiente comando:

```
apt list --installed | grep ^cloudhsm
```

Ubuntu 20.04 LTS

Utilice el siguiente comando:

```
apt list --installed | grep ^cloudhsm
```

Windows Server

Utilice el siguiente comando:

```
wmic product get name,version
```

Comparación de componentes de SDK de cliente

Además de las herramientas de línea de comandos, SDK 3 de cliente contiene componentes que permiten transferir las operaciones criptográficas al HSM desde diversas aplicaciones basadas

en lenguaje o plataforma. El SDK de cliente 5 tiene paridad con el SDK de cliente 3, excepto que todavía no es compatible con los proveedores de GNC y KSP. En la siguiente tabla encontrará una comparativa de la disponibilidad de componentes en SDK 3 de cliente y SDK 5 de cliente.

Componente	SDK 5 de cliente	SDK 3 de cliente
Biblioteca PKCS #11	Sí	Sí
Proveedor de JCE	Sí	Sí
Motor dinámico de OpenSSL	Sí	Sí
Proveedores de KSP y CNG		Sí
Utilidad de administración de CloudHSM (CMU)[1]	Sí	Sí
Utilidad de administración de claves (KMU)[1]	Sí	Sí
Herramienta de configuración	Sí	Sí

[1] Los componentes CMU y KMU están incluidos en la CLI de CloudHSM con SDK 5 de cliente.

Temas

- Plataformas compatibles con SDK 3 de cliente
- Actualización de SDK 3 de cliente en Linux
- Biblioteca PKCS #11 para SDK 3 de cliente
- Instalación del SDK 3 de cliente para el motor dinámico de OpenSSL
- SDK 3 de cliente para proveedor de JCE

Plataformas compatibles con SDK 3 de cliente

SDK 3 de cliente requiere un daemon de cliente y ofrece herramientas de línea de comandos, como la Utilidad de administración de CloudHSM (CMU), la Utilidad de administración de claves (KMU) y la herramienta de configuración.

El soporte básico es diferente para cada versión del SDK de AWS CloudHSM cliente. Por lo general, el soporte de plataforma para los componentes de un SDK coincide con el soporte básico, pero no siempre. Para determinar el soporte de plataforma de un componente concreto, primero debe asegurarse de que la plataforma que desea esté incluida en la sección básica del SDK y, a continuación, comprobar si existen exclusiones o cualquier otra información pertinente en la sección de componentes.

El soporte de plataforma varía con el tiempo. Es posible que las versiones anteriores del SDK de cliente de CloudHSM no sean compatibles con todos los sistemas operativos enumerados aquí. Utilice las notas de la versión para determinar la compatibilidad del sistema operativo con las versiones anteriores del SDK de cliente de CloudHSM. Para obtener más información, consulte Descargas para AWS CloudHSM Client SDK.

AWS CloudHSM solo admite sistemas operativos de 64 bits.

Contenido

- Compatibilidad con Linux
- Compatibilidad con Windows
- Compatibilidad con componentes
  - Biblioteca PKCS #11
  - Proveedor de JCE
  - Motor dinámico de OpenSSL
  - Proveedores de CNG y KSP

Compatibilidad con Linux

- Amazon Linux
- Amazon Linux 2
- CentOS 6,10+ [2]
- CentOS 7.3+
- CentOS 8 [1,4]
- Red Hat Enterprise Linux (RHEL) 6,10+ [2]
- Red Hat Enterprise Linux (RHEL) 7.3+
- Red Hat Enterprise Linux (RHEL) 8 [1]

- Ubuntu 16.04 LTS [3]
- Ubuntu 18.04 LTS [1]

[1] No es compatible con OpenSSL Dynamic Engine. Para obtener más información, consulte OpenSSL Dynamic Engine.

[2] Sin compatibilidad para la versión 3.3.0 y posteriores del SDK de cliente.

[3] SDK 3.4 es la última versión compatible con Ubuntu 16.04.

[4] SDK 3.4 es la última versión compatible con CentOS 8.3+.

Compatibilidad con Windows

- Microsoft Windows Server 2012
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019

Compatibilidad con componentes

Biblioteca PKCS #11

La biblioteca PKCS #11 es un componente exclusivo de Linux que coincide con el soporte básico de Linux. Para obtener más información, consulte the section called “Compatibilidad con Linux”.

Proveedor de JCE

El proveedor JCE es un componente exclusivo de Linux que coincide con el soporte base de Linux. Para obtener más información, consulte the section called “Compatibilidad con Linux”.

- Requiere OpenJDK 1.8

Motor dinámico de OpenSSL

El motor dinámico OpenSSL es un componente exclusivo de Linux que no coincide con el soporte básico de Linux. Vea las siguientes exclusiones.

- Requiere OpenSSL 1.0.2[f+]

Plataformas no compatibles:

- CentOS 8
- Red Hat Enterprise Linux (RHEL) 8
- Ubuntu 18.04 LTS

Estas plataformas incluyen una versión de OpenSSL incompatible con el motor dinámico de OpenSSL para SDK 3 de cliente. AWS CloudHSM es compatible con estas plataformas con el motor dinámico de OpenSSL para SDK 5 de cliente.

Proveedores de CNG y KSP

Los proveedores CNG y KSP son un componente exclusivo de Windows que coincide con el soporte básico de Windows. Para obtener más información, consulte Compatibilidad con Windows.

Actualización de SDK 3 de cliente en Linux

Con AWS CloudHSM Client SDK 3.1 y versiones posteriores, la versión del daemon del cliente y cualquier componente que instale deben coincidir para poder realizar la actualización. Para todos los sistemas basados en Linux, debe usar un comando único para actualizar por lotes el daemon del cliente con la misma versión de la biblioteca PKCS #11, el proveedor de extensión criptográfica de Java (JCE) o el motor dinámico de OpenSSL. Este requisito no es aplicable a los sistemas basados en Windows, ya que los binarios de los proveedores CNG y KSP ya están incluidos en el paquete daemon del cliente.

Cómo comprobar la versión daemon del cliente

- En sistemas Linux basados en Red Hat (incluidos Amazon Linux y CentOS), use el siguiente comando:

```
rpm -qa | grep ^cloudhsm
```

- En un sistema Linux basado en Debian, utilice el siguiente comando:

```
apt list --installed | grep ^cloudhsm
```

- En un sistema Windows, utilice el siguiente comando:

```
wmic product get name,version
```

Temas

- Requisitos previos
- Paso 1: detenga el daemon de cliente
- Paso 2: actualice el SDK de cliente.
- Paso 3: inicie el daemon de cliente

Requisitos previos

Descarga la última versión del daemon del AWS CloudHSM cliente y elige tus componentes.

> **Note**
>
> No es necesario instalar todos los componentes. Deberá actualizar cada uno de las componentes que tenga instalados para que coincida con la versión del daemon de cliente.

Daemon de cliente Linux más reciente

Amazon Linux

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-latest.el6.x86_64.rpm
```

Amazon Linux 2

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-latest.el7.x86_64.rpm
```

CentOS 7

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-latest.el7.x86_64.rpm
```

CentOS 8

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-latest.el8.x86_64.rpm
```

RHEL 7

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-latest.el7.x86_64.rpm
```

RHEL 8

```
sudo yum install wget
```

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-latest.el8.x86_64.rpm
```

Ubuntu 16.04 LTS

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/cloudhsm-client_latest_amd64.deb
```

Ubuntu 18.04 LTS

```
wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Bionic/cloudhsm-client_latest_u18.04_amd64.deb
```

Biblioteca PKCS #11 más reciente

Amazon Linux

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-pkcs11-latest.el6.x86_64.rpm
```

Amazon Linux 2

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
```

CentOS 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
```

CentOS 8

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-pkcs11-latest.el8.x86_64.rpm
```

RHEL 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
```

RHEL 8

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-pkcs11-latest.el8.x86_64.rpm
```

Ubuntu 16.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/cloudhsm-client-pkcs11_latest_amd64.deb
```

Ubuntu 18.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Bionic/cloudhsm-client-pkcs11_latest_u18.04_amd64.deb
```

Motor dinámico de OpenSSL más reciente

Amazon Linux

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-dyn-latest.el6.x86_64.rpm
```

Amazon Linux 2

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-dyn-latest.el7.x86_64.rpm
```

CentOS 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-dyn-latest.el7.x86_64.rpm
```

RHEL 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-dyn-latest.el7.x86_64.rpm
```

Ubuntu 16.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/cloudhsm-client-dyn_latest_amd64.deb
```

Proveedor de JCE más reciente

Amazon Linux

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-jce-latest.el6.x86_64.rpm
```

Amazon Linux 2

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-jce-latest.el7.x86_64.rpm
```

CentOS 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-jce-latest.el7.x86_64.rpm
```

CentOS 8

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-jce-latest.el8.x86_64.rpm
```

RHEL 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-jce-latest.el7.x86_64.rpm
```

RHEL 8

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-jce-latest.el8.x86_64.rpm
```

Ubuntu 16.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/cloudhsm-client-jce_latest_amd64.deb
```

Ubuntu 18.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Bionic/cloudhsm-client-jce_latest_u18.04_amd64.deb
```

Paso 1: detenga el daemon de cliente

Utilice el siguiente comando para detener el daemon de cliente.

Amazon Linux

```
$ sudo stop cloudhsm-client
```

Amazon Linux 2

```
$ sudo service cloudhsm-client stop
```

CentOS 7

```
$ sudo service cloudhsm-client stop
```

CentOS 8

```
$ sudo service cloudhsm-client stop
```

RHEL 7

```
$ sudo service cloudhsm-client stop
```

RHEL 8

```
$ sudo service cloudhsm-client stop
```

Ubuntu 16.04 LTS

```
$ sudo service cloudhsm-client stop
```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client stop
```

Paso 2: actualice el SDK de cliente.

El siguiente comando muestra la sintaxis necesaria para actualizar el daemon de cliente y los componentes. Antes de ejecutar el comando, quite los componentes que no quiera actualizar.

Amazon Linux

```
$ sudo yum install ./cloudhsm-client-latest.el6.x86_64.rpm \
              <./cloudhsm-client-pkcs11-latest.el6.x86_64.rpm> \
              <./cloudhsm-client-dyn-latest.el6.x86_64.rpm> \
              <./cloudhsm-client-jce-latest.el6.x86_64.rpm>
```

Amazon Linux 2

```
$ sudo yum install ./cloudhsm-client-latest.el7.x86_64.rpm \
              <./cloudhsm-client-pkcs11-latest.el7.x86_64.rpm> \
```

```
                <./cloudhsm-client-dyn-latest.el7.x86_64.rpm> \
                <./cloudhsm-client-jce-latest.el7.x86_64.rpm>
```

CentOS 7

```
$ sudo yum install ./cloudhsm-client-latest.el7.x86_64.rpm \
                <./cloudhsm-client-pkcs11-latest.el7.x86_64.rpm> \
                <./cloudhsm-client-dyn-latest.el7.x86_64.rpm> \
                <./cloudhsm-client-jce-latest.el7.x86_64.rpm>
```

CentOS 8

```
$ sudo yum install ./cloudhsm-client-latest.el8.x86_64.rpm \
                <./cloudhsm-client-pkcs11-latest.el8.x86_64.rpm> \
                <./cloudhsm-client-jce-latest.el8.x86_64.rpm>
```

RHEL 7

```
$ sudo yum install ./cloudhsm-client-latest.el7.x86_64.rpm \
                <./cloudhsm-client-pkcs11-latest.el7.x86_64.rpm> \
                <./cloudhsm-client-dyn-latest.el7.x86_64.rpm> \
                <./cloudhsm-client-jce-latest.el7.x86_64.rpm>
```

RHEL 8

```
$ sudo yum install ./cloudhsm-client-latest.el8.x86_64.rpm \
                <./cloudhsm-client-pkcs11-latest.el8.x86_64.rpm> \
                <./cloudhsm-client-jce-latest.el8.x86_64.rpm>
```

Ubuntu 16.04 LTS

```
$ sudo apt install ./cloudhsm-client_latest_amd64.deb \
                <cloudhsm-client-pkcs11_latest_amd64.deb> \
                <cloudhsm-client-dyn_latest_amd64.deb> \
                <cloudhsm-client-jce_latest_amd64.deb>
```

Ubuntu 18.04 LTS

```
$ sudo apt install ./cloudhsm-client_latest_u18.04_amd64.deb \
                <cloudhsm-client-pkcs11_latest_amd64.deb> \
                <cloudhsm-client-jce_latest_amd64.deb>
```

Paso 3: inicie el daemon de cliente

Utilice el siguiente comando para iniciar el daemon de cliente.

Amazon Linux

```
$ sudo start cloudhsm-client
```

Amazon Linux 2

```
$ sudo service cloudhsm-client start
```

CentOS 7

```
$ sudo service cloudhsm-client start
```

CentOS 8

```
$ sudo service cloudhsm-client start
```

RHEL 7

```
$ sudo service cloudhsm-client start
```

RHEL 8

```
$ sudo service cloudhsm-client start
```

Ubuntu 16.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 20.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

Biblioteca PKCS #11 para SDK 3 de cliente

El PKCS #11 es un estándar para la realización de operaciones criptográficas en módulos de seguridad de hardware (HSM).

Para obtener más información sobre el arranque, consulte Conexión al clúster.

Temas

- Instalación del SDK 3 de cliente para la biblioteca PKCS #11
- Autenticación en la biblioteca PKCS #11 (SDK 3 de cliente)
- Tipos de claves compatibles (SDK 3 de cliente)
- Mecanismos compatibles (SDK 3 de cliente)
- Operaciones de API compatibles (SDK 3 de cliente)
- Atributos de clave compatibles con SDK 3 de cliente
- Ejemplos de código para la biblioteca PKCS #11 (SDK 3 de cliente)

Instalación del SDK 3 de cliente para la biblioteca PKCS #11

Requisitos previos para el SDK 3 de cliente

La biblioteca PKCS #11 requiere el AWS CloudHSM cliente.

Si no ha instalado ni configurado el AWS CloudHSM cliente, hágalo ahora siguiendo los pasos que se indican enInstalar el cliente (Linux). Después de instalar y configurar el cliente, utilice el siguiente comando para iniciarlo.

Amazon Linux

```
$ sudo start cloudhsm-client
```

Amazon Linux 2

```
$ sudo systemctl cloudhsm-client start
```

CentOS 7

```
$ sudo systemctl cloudhsm-client start
```

CentOS 8

```
$ sudo systemctl cloudhsm-client start
```

RHEL 7

```
$ sudo systemctl cloudhsm-client start
```

RHEL 8

```
$ sudo systemctl cloudhsm-client start
```

Ubuntu 16.04 LTS

```
$ sudo systemctl cloudhsm-client start
```

Ubuntu 18.04 LTS

```
$ sudo systemctl cloudhsm-client start
```

Ubuntu 20.04 LTS

```
$ sudo systemctl cloudhsm-client start
```

Instale la biblioteca PKCS #11 para el SDK 3 de cliente

El siguiente comando descarga e instala la biblioteca PKCS #11.

Amazon Linux

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-pkcs11-latest.el6.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-pkcs11-latest.el6.x86_64.rpm
```

Amazon Linux 2

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
```

CentOS 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
```

CentOS 8

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-pkcs11-latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-pkcs11-latest.el8.x86_64.rpm
```

RHEL 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-pkcs11-latest.el7.x86_64.rpm
```

RHEL 8

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-pkcs11-latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-pkcs11-latest.el8.x86_64.rpm
```

Ubuntu 16.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/cloudhsm-client-pkcs11_latest_amd64.deb
```

```
$ sudo apt install ./cloudhsm-client-pkcs11_latest_amd64.deb
```

Ubuntu 18.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Bionic/cloudhsm-client-pkcs11_latest_u18.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-client-pkcs11_latest_u18.04_amd64.deb
```

- Si la instancia EC2 en la que instaló la biblioteca PKCS #11 no tiene instalados otros componentes del SDK 3 de cliente, debe iniciar el SDK 3 de cliente. Solo tiene que hacerlo una vez para cada instancia con un componente del SDK 3 de cliente.
- Puede encontrar los archivos de la biblioteca PKCS #11 en las siguientes ubicaciones:

  Binarios, scripts de configuración, certificados y archivos de registro de Linux:

  ```
  /opt/cloudhsm/lib
  ```

Autenticación en la biblioteca PKCS #11 (SDK 3 de cliente)

Cuando se utiliza la biblioteca PKCS #11, su aplicación se ejecuta como un determinado usuario de criptografía (CU) en los HSM. La aplicación solo puede ver y administrar las claves que posee y comparte el CU. Puede utilizar un CU existente en sus HSM o crear un nuevo CU. Para obtener más información sobre la administración de CU, consulte Administración de usuarios de HSM con la CLI de CloudHSM y Administración de usuarios de HSM con utilidad de administración de CloudHSM (CMU).

Para especificar el CU para la biblioteca PKCS #11, utilice el parámetro de pin de la función C_Login de PKCS #11. AWS CloudHSM En efecto, el parámetro pin tiene el siguiente formato:

```
<CU_user_name>:<password>
```

Por ejemplo, el siguiente comando establece el pin de la biblioteca PKCS #11 para el CU con el nombre de usuario `CryptoUser` y la contraseña `CUPassword123!`.

```
CryptoUser:CUPassword123!
```

Tipos de claves compatibles (SDK 3 de cliente)

La biblioteca PKCS #11 admite los siguientes tipos de clave.

Tipo de clave	Descripción
RSA	Genere claves RSA de 2048 a 4096 bits, en incrementos de 256 bits
EC	Genere claves con las curvas secp224r1 (P-224), secp256r1 (P-256), secp256k1 (Blockchain), secp384r1 (P-384) y secp521r1 (P-521).
AES	Genere claves AES de 128, 192 y 256 bits.
DES3 (Triple DES)	Genere claves DES3 de 192 bits. Consulte la nota 1 que aparece a continuación para ver los próximos cambios.
GENERIC_SECRET	Genere secretos genéricos de 1 a 64 bytes.

- [1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Mecanismos compatibles (SDK 3 de cliente)

La biblioteca PKCS #11 admite los siguientes algoritmos:

- Cifrado y descifrado: AES-CBC, AES-CTR, AES-ECB, AES-GCM, DES3-CBC, DES3-ECB, RSA-OAEP y RSA-PKCS
- Firma y verificación: RSA, HMAC y ECDSA; con y sin hash

- Hash/digest: SHA1, SHA224, SHA256, SHA384 y SHA512
- Encapsulación de claves: encapsulación de claves AES,[4] AES-GCM, RSA-AES y RSA-OAEP
- Derivación de claves: ECDH,[5] SP800-108 CTR KDF

Tabla de mecanismos y funciones de la biblioteca PKCS #11

La biblioteca PKCS #11 es compatible con la versión 2.40 de la especificación PKCS #11. Para invocar una característica criptográfica con PKCS #11, llame a una función con un mecanismo determinado. En la siguiente tabla, se resumen las combinaciones de funciones y mecanismos admitidos por AWS CloudHSM.

Interpretación de la tabla de mecanismos y funciones compatibles con PKCS #11

La marca ✓ indica que AWS CloudHSM es compatible con el mecanismo de la función. No se admiten todas las funciones posibles que se muestran en la especificación de PKCS #11. La marca ✖ indica que aún AWS CloudHSM no es compatible con el mecanismo para la función en cuestión, aunque el estándar PKCS #11 lo permita. Las celdas vacías indican que el estándar de PKCS #11 no admite el mecanismo para esa función.

Mecanismos y funciones compatibles con la biblioteca PKCS #11

Mecanismo	Funciones						
	Generar claves o pares de claves	Firmar y verificar	SR y VR	Resumir	Cifrar o descifrar	Derivar clave	Envolver y UnWrap
CKM_RSA_PKCS_KEY_PAIR_GEN	✓						
CKM_RSA_X9_31_KEY_PAIR_GEN	✓[2]						

Mecanismo	Funciones						
CKM_RSA_X_509		✓			✓		
CKM_RSA_PKCS consulte la nota 8		✓[1]	✖		✓[1]		✓[1]
CKM_RSA_PKCS_OAEP					✓[1]		✓[6]
CKM_SHA1_RSA_PKCS		✓[3.2]					
CKM_SHA224_RSA_PKCS		✓[3.2]					
CKM_SHA256_RSA_PKCS		✓[3.2]					
CKM_SHA384_RSA_PKCS		✓[2,3.2]					
CKM_SHA512_RSA_PKCS		✓[3.2]					
CKM_RSA_PKCS_PSS		✓[1]					
CKM_SHA1_RSA_PKCS_PSS		✓[3.2]					

Mecanismo	Funciones						
CKM_SHA224_RSA_PKCS_PSS		✓[3.2]					
CKM_SHA256_RSA_PKCS_PSS		✓[3.2]					
CKM_SHA384_RSA_PKCS_PSS		✓[2,3.2]					
CKM_SHA512_RSA_PKCS_PSS		✓[3.2]					
CKM_EC_KEY_PAIR_GEN	✓						
CKM_ECDSA		✓[1]					
CKM_ECDSA_SHA1		✓[3.2]					
CKM_ECDSA_SHA224		✓[3.2]					
CKM_ECDSA_SHA256		✓[3.2]					
CKM_ECDSA_SHA384		✓[3.2]					
CKM_ECDSA_SHA512		✓[3.2]					

Mecanismo	Funciones						
CKM_ECDH1_DERIVE						✓[5]	
CKM_SP800_108_COUNTER_KDF						✓	
CKM_GENERIC_SECRET_KEY_GEN	✓						
CKM_AES_KEY_GEN	✓						
CKM_AES_ECB					✓		✖
CKM_AES_CTR					✓		✖
CKM_AES_CBC					✓[3.3]		✖
CKM_AES_CBC_PAD					✓		✖
CKM_DES3_KEY_GEN consulte la nota 8	✓						
CKM_DES3_CBC consulte la nota 8					✓[3.3]		✖

Mecanismo	Funciones						
CKM_DES3_CBC_PAD consulte la nota 8					✓		✖
CKM_DES3_ECB consulte la nota 8					✓		✖
CKM_AES_GCM					✓3.3, 4		✓7.1
CKM_CLOUDHSM_AES_GCM					✓7.1		✓7.1
CKM_SHA_1				✓3.1			
CKM_SHA_1_HMAC		✓3.3					
CKM_SHA224				✓3.1			
CKM_SHA224_HMAC		✓3.3					
CKM_SHA256				✓3.1			
CKM_SHA256_HMAC		✓3.3					
CKM_SHA384				✓3.1			

Mecanismo	Funciones						
CKM_SHA384_HMAC		✓[3.3]					
CKM_SHA512				✓[3.1]			
CKM_SHA512_HMAC		✓[3.3]					
CKM_RSA_AES_KEY_WRAP							✓
CKM_AES_KEY_WRAP							✓
CKM_AES_KEY_WRAP_PAD							✓
CKM_CLOUDHSM_AES_KEY_WRAP_NO_PAD							✓[7.1]
CKM_CLOUDHSM_AES_KEY_WRAP_PKCS5_PAD							✓[7.1]
CKM_CLOUDHSM_AES_KEY_WRAP_ZERO_PAD							✓[7.1]

Notas del mecanismo

- [1] Únicamente para operaciones de una sola parte.
- [2] Este mecanismo es funcionalmente idéntico al mecanismo `CKM_RSA_PKCS_KEY_PAIR_GEN`, pero ofrece más garantías en la generación de p y q.
- [3.1] AWS CloudHSM aborda el hash de forma diferente en función del SDK del cliente. En el caso de SDK 3 de cliente, el hashing depende del tamaño de los datos y de si se utilizan operaciones de una o varias partes.

  Operaciones de una sola parte en SDK 3 de cliente

  En la tabla 3.1 se muestra el tamaño máximo del conjunto de datos para cada mecanismo de SDK 3 de cliente. El hash completo se calcula dentro del HSM. No admite tamaños de datos superiores a 16 KB.

  Tabla 3.1, tamaño máximo del conjunto de datos para operaciones de una sola parte

Mecanismo	Tamaño máximo de datos
`CKM_SHA_1`	16296
`CKM_SHA224`	16264
`CKM_SHA256`	16296
`CKM_SHA384`	16232
`CKM_SHA512`	16232

  Operaciones de varias partes en SDK 3 de cliente

  Admite tamaños de datos superiores a 16 KB, pero el tamaño de los datos determina el lugar del hashing. Los búferes de datos de menos de 16 KB se codifican con un hash dentro del HSM. Los búferes de entre 16 KB y el tamaño máximo de datos de su sistema se codifican localmente en el software. Recuerde: Las funciones de hash no requieren de secretos criptográficos, por lo que puede calcularlas de forma segura fuera del HSM.
- [3.2] AWS CloudHSM aborda el hash de forma diferente en función del SDK del cliente. En el caso de SDK 3 de cliente, el hashing depende del tamaño de los datos y de si se utilizan operaciones de una o varias partes.

  Operaciones de una sola parte en SDK 3 de cliente

En la tabla 3.2 se muestra el tamaño máximo del conjunto de datos para cada mecanismo de SDK 3 de cliente. No admite tamaños de datos superiores a 16 KB.

Tabla 3.2, tamaño máximo del conjunto de datos para operaciones de una sola parte

Mecanismo	Tamaño máximo de datos
CKM_SHA1_RSA_PKCS	16296
CKM_SHA224_RSA_PKCS	16264
CKM_SHA256_RSA_PKCS	16296
CKM_SHA384_RSA_PKCS	16232
CKM_SHA512_RSA_PKCS	16232
CKM_SHA1_RSA_PKCS_PSS	16296
CKM_SHA224_RSA_PKCS_PSS	16264
CKM_SHA256_RSA_PKCS_PSS	16296
CKM_SHA384_RSA_PKCS_PSS	16232
CKM_SHA512_RSA_PKCS_PSS	16232
CKM_ECDSA_SHA1	16296
CKM_ECDSA_SHA224	16264
CKM_ECDSA_SHA256	16296
CKM_ECDSA_SHA384	16232
CKM_ECDSA_SHA512	16232

Operaciones de varias partes en SDK 3 de cliente

Admite tamaños de datos superiores a 16 KB, pero el tamaño de los datos determina el lugar del hashing. Los búferes de datos de menos de 16 KB se codifican con un hash dentro del HSM. Los

búferes de entre 16 KB y el tamaño máximo de datos de su sistema se codifican localmente en el software. Recuerde: Las funciones de hash no requieren de secretos criptográficos, por lo que puede calcularlas de forma segura fuera del HSM.

- [3.3] Cuando se opera con datos mediante cualquiera de los mecanismos siguientes, si el búfer de datos supera el tamaño máximo de datos, la operación produce un error. Para estos mecanismos, todo el procesamiento de los datos debe realizarse dentro del HSM. En la tabla siguiente, se muestra el tamaño máximo de datos establecido para cada mecanismo:

  Tabla 3.3, tamaño máximo del conjunto de datos

Mecanismo	Tamaño máximo de datos
`CKM_SHA_1_HMAC`	16288
`CKM_SHA224_HMAC`	16256
`CKM_SHA256_HMAC`	16288
`CKM_SHA384_HMAC`	16224
`CKM_SHA512_HMAC`	16224
`CKM_AES_CBC`	16272
`CKM_AES_GCM`	16224
`CKM_CLOUDHSM_AES_GCM`	16224
`CKM_DES3_CBC`	16280

- [4] Al realizar el cifrado AES-GCM, el HSM no acepta los datos del vector de inicialización (IV) de la aplicación. Debe utilizar un vector de inicialización generado. El IV de 12 bytes proporcionado por el HSM se escribe en la referencia de memoria a la que apunta el elemento pIV de la estructura de parámetros `CK_GCM_PARAMS` especificada por el usuario. Para asegurarse de no generar confusión en el usuario, el SDK de PKCS#11 versión 1.1.1 y posteriores obliga a que el elemento pIV apunte a un búfer puesto a cero cuando se inicializa el cifrado AES-GCM.
- [5] Solo SDK 3 de cliente. Este mecanismo se implementa para admitir casos de descarga de SSL/TLS y solo se ejecuta parcialmente en el HSM. Antes de usar este mecanismo, consulte «Issue: ECDH key derivation is executed only partially within the HSM» en Problemas conocidos de la biblioteca PKCS #11.`CKM_ECDH1_DERIVE` no admite la curva secp521r1 (P-521).

- [6] Los siguientes `CK_MECHANISM_TYPE` y `CK_RSA_PKCS_MGF_TYPE` se admiten como `CK_RSA_PKCS_OAEP_PARAMS` para `CKM_RSA_PKCS_OAEP`:
  - `CKM_SHA_1` con `CKG_MGF1_SHA1`
  - `CKM_SHA224` con `CKG_MGF1_SHA224`
  - `CKM_SHA256` con `CKG_MGF1_SHA256`
  - `CKM_SHA384` con `CKM_MGF1_SHA384`
  - `CKM_SHA512` con `CKM_MGF1_SHA512`
- [7.1] Mecanismo definido por el proveedor. Para poder utilizar los mecanismos definidos por el proveedor de CloudHSM, las aplicaciones PKCS#11 deben incluir `/opt/cloudhsm/include/pkcs11t.h` durante la compilación.

  `CKM_CLOUDHSM_AES_GCM`: este mecanismo exclusivo es una alternativa programáticamente segura del estándar `CKM_AES_GCM`. Antepone el IV generado por el HSM al texto cifrado en lugar de volver a escribirlo en la estructura `CK_GCM_PARAMS` que se proporciona durante la inicialización del cifrado. Puede utilizar este mecanismo con las funciones `C_Encrypt`, `C_WrapKey`, `C_Decrypt` y `C_UnwrapKey`. Cuando se utiliza este mecanismo, la variable pIV de la estructura `CK_GCM_PARAMS` debe establecerse en `NULL`. Cuando se utiliza este mecanismo con `C_Decrypt` y `C_UnwrapKey`, se espera que el IV se anteponga al texto cifrado que se está desencapsulando.

  `CKM_CLOUDHSM_AES_KEY_WRAP_PKCS5_PAD`: encapsulamiento de claves AES con relleno PKCS #5

  `CKM_CLOUDHSM_AES_KEY_WRAP_ZERO_PAD`: encapsulamiento de claves AES con relleno de ceros

  Para obtener más opciones de encapsulamiento de claves AES, consulte Encapsulamiento de claves con AES.
- [8] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Operaciones de API compatibles (SDK 3 de cliente)

La biblioteca PKCS #11 admite las siguientes operaciones API de PKCS #11.

- `C_CloseAllSessions`

- C_CloseSession
- C_CreateObject
- C_Decrypt
- C_DecryptFinal
- C_DecryptInit
- C_DecryptUpdate
- C_DeriveKey
- C_DestroyObject
- C_Digest
- C_DigestFinal
- C_DigestInit
- C_DigestUpdate
- C_Encrypt
- C_EncryptFinal
- C_EncryptInit
- C_EncryptUpdate
- C_Finalize
- C_FindObjects
- C_FindObjectsFinal
- C_FindObjectsInit
- C_GenerateKey
- C_GenerateKeyPair
- C_GenerateRandom
- C_GetAttributeValue
- C_GetFunctionList
- C_GetInfo
- C_GetMechanismInfo
- C_GetMechanismList
- C_GetSessionInfo

- `C_GetSlotInfo`
- `C_GetSlotList`
- `C_GetTokenInfo`
- `C_Initialize`
- `C_Login`
- `C_Logout`
- `C_OpenSession`
- `C_Sign`
- `C_SignFinal`
- `C_SignInit`
- `C_SignRecover` (Solo compatible con SDK 3 de cliente)
- `C_SignRecoverInit` (Solo compatible con SDK 3 de cliente)
- `C_SignUpdate`
- `C_UnWrapKey`
- `C_Verify`
- `C_VerifyFinal`
- `C_VerifyInit`
- `C_VerifyRecover` (Solo compatible con SDK 3 de cliente)
- `C_VerifyRecoverInit` (Solo compatible con SDK 3 de cliente)
- `C_VerifyUpdate`
- `C_WrapKey`

Atributos de clave compatibles con SDK 3 de cliente

Un objeto de clave puede ser una clave pública, privada o secreta. Las acciones permitidas en un objeto de clave se especifican mediante atributos. Los atributos se definen cuando se crea el objeto de clave. Cuando se utiliza la biblioteca PKCS #11, asignamos los valores predeterminados que se especifican en el estándar PKCS #11.

AWS CloudHSM no admite todos los atributos enumerados en la especificación PKCS #11. Seguimos esta especificación en todos los atributos que admitimos. Estos atributos se indican en sus respectivas tablas.

Las funciones criptográficas como `C_CreateObject`, `C_GenerateKey`, `C_GenerateKeyPair`, `C_UnwrapKey` y `C_DeriveKey` que crean, modifican o copian objetos toman una plantilla de atributos como uno de sus parámetros. Para obtener más información acerca de cómo pasar una plantilla de atributos durante la creación de objetos, consulte el ejemplo Generar claves mediante la biblioteca de PKCS #11.

Interpretación de la tabla de atributos de la biblioteca PKCS #11

La tabla de la biblioteca PKCS #11 contiene una lista de atributos que difieren por tipo de clave. Indica si un atributo determinado es compatible con un tipo de clave concreto cuando se utiliza una función criptográfica específica con. AWS CloudHSM

Leyenda

- ✓ indica que CloudHSM admite el atributo para el tipo de clave específico.
- ✖ indica que CloudHSM no admite el atributo para el tipo de clave específico.
- R indica que el valor del atributo se establece en de solo lectura para el tipo de clave específico.
- S indica que `GetAttributeValue` no puede leer el atributo porque distingue entre mayúsculas y minúsculas.
- Una celda vacía en la columna Valor predeterminado indica que no hay ningún valor predeterminado específico asignado al atributo.

GenerateKeyPair

Atributo	Tipo de clave					Default Value (Valor predeterminado)
		EC privada	EC pública	RSA privada	RSA pública	
`CKA_CLASS`		✓	✓	✓	✓	

Atributo	Tipo de clave				Default Value (Valor predeterminado)
CKA_KEY_TYPE	✓	✓	✓	✓	
CKA_LABEL	✓	✓	✓	✓	
CKA_ID	✓	✓	✓	✓	
CKA_LOCAL	R	R	R	R	True
CKA_TOKEN	✓	✓	✓	✓	False
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT	✖	✓	✖	✓	False
CKA_DECRYPT	✓	✖	✓	✖	False
CKA_DERIVE	✓	✓	✓	✓	False
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_DESTROYABLE	✓	✓	✓	✓	True
CKA_SIGN	✓	✖	✓	✖	False

Atributo	Tipo de clave				Default Value (Valor predeterminado)
CKA_SIGN_RECOVER	✖	✖	✓[3]	✖	
CKA_VERIFY	✖	✓	✖	✓	False
CKA_VERIFY_RECOVER	✖	✖	✖	✓[4]	
CKA_WRAP	✖	✓	✖	✓	False
CKA_WRAP_TEMPLATE	✖	✓	✖	✓	
CKA_TRUSTED	✖	✓	✖	✓	False
CKA_WRAP_WITH_TRUSTED	✓	✖	✓	✖	False
CKA_UNWRAP	✓	✖	✓	✖	False
CKA_UNWRAP_TEMPLATE	✓	✖	✓	✖	
CKA_SENSITIVE	✓	✖	✓	✖	True

Atributo	Tipo de clave				Default Value (Valor predeterminado)
CKA_ALWAYS_SENSITIVE	R	✖	R	✖	
CKA_EXTRACTABLE	✓	✖	✓	✖	True
CKA_NEVER_EXTRACTABLE	R	✖	R	✖	
CKA_MODULUS	✖	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	✓[2]	
CKA_PRIME_1	✖	✖	✖	✖	
CKA_PRIME_2	✖	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✖	✖	
CKA_EXPONENT_1	✖	✖	✖	✖	
CKA_EXPONENT_2	✖	✖	✖	✖	

Atributo	Tipo de clave				Default Value (Valor predeterminado)
CKA_PRIVATE_EXPONENT	✖	✖	✖	✖	
CKA_PUBLIC_EXPONENT	✖	✖	✖	✓[2]	
CKA_EC_PARAMS	✖	✓[2]	✖	✖	
CKA_EC_POINT	✖	✖	✖	✖	
CKA_VALUE	✖	✖	✖	✖	
CKA_VALUE_LEN	✖	✖	✖	✖	
CKA_CHECK_VALUE	R	R	R	R	

GenerateKey

Atributo	Tipo de clave			Default Value (Valor predeterminado)
	AES	DES3	Secreto genérico	
CKA_CLASS	✓	✓	✓	
CKA_KEY_TYPE	✓	✓	✓	
CKA_LABEL	✓	✓	✓	
CKA_ID	✓	✓	✓	
CKA_LOCAL	R	R	R	True
CKA_TOKEN	✓	✓	✓	False
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT	✓	✓	✖	False
CKA_DECRYPT	✓	✓	✖	False
CKA_DERIVE	✓	✓	✓	False
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	True
CKA_DESTROYABLE	✓	✓	✓	True

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_SIGN	✓	✓	✓	True
CKA_SIGN_RECOVER	✖	✖	✖	
CKA_VERIFY	✓	✓	✓	True
CKA_VERIFY_RECOVER	✖	✖	✖	
CKA_WRAP	✓	✓	✖	False
CKA_WRAP_TEMPLATE	✓	✓	✖	
CKA_TRUSTED	✓	✓	✖	False
CKA_WRAP_WITH_TRUSTED	✓	✓	✓	False
CKA_UNWRAP	✓	✓	✖	False
CKA_UNWRAP_TEMPLATE	✓	✓	✖	
CKA_SENSITIVE	✓	✓	✓	True

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_ALWAYS_SENSITIVE	✖	✖	✖	
CKA_EXTRACTABLE	✓	✓	✓	True
CKA_NEVER_EXTRACTABLE	R	R	R	
CKA_MODULUS	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	
CKA_PRIME_1	✖	✖	✖	
CKA_PRIME_2	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✖	
CKA_EXPONENT_1	✖	✖	✖	
CKA_EXPONENT_2	✖	✖	✖	

Atributo	Tipo de clave							Default Value (Valor predeterminado)
CKA_PRIVATE_EXPONENT		✖	✖	✖				
CKA_PUBLIC_EXPONENT		✖	✖	✖				
CKA_EC_PARAMS		✖	✖	✖				
CKA_EC_POINT		✖	✖	✖				
CKA_VALUE		✖	✖	✖				
CKA_VALUE_LEN		✓[2]	✖	✓[2]				
CKA_CHECK_VALUE		R	R	R				

CreateObject

Atributo	Tipo de clave							Default Value (Valor predeterminado)
		EC privada	EC pública	RSA privada	RSA pública	AES	DES3	Secreto genérico

Atributo	Tipo de clave							Default Value (Valor predeterminado)
CKA_CLASS	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	
CKA_KEY_TYPE	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	
CKA_LABEL	✓	✓	✓	✓	✓	✓	✓	
CKA_ID	✓	✓	✓	✓	✓	✓	✓	
CKA_LOCAL	R	R	R	R	R	R	R	False
CKA_TOKEN	✓	✓	✓	✓	✓	✓	✓	False
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT	✖	✖	✖	✓	✓	✓	✖	False
CKA_DECRYPT	✖	✖	✓	✖	✓	✓	✖	False
CKA_DERIVE	✓	✓	✓	✓	✓	✓	✓	False
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	True

Atributo	Tipo de clave							Default Value (Valor predeterminado)
CKA_DESTROYABLE	✓	✓	✓	✓	✓	✓	✓	True
CKA_SIGN	✓	✖	✓	✖	✓	✓	✓	False
CKA_SIGN_RECOVER	✖	✖	✓[3]	✖	✖	✖	✖	False
CKA_VERIFY	✖	✓	✖	✓	✓	✓	✓	False
CKA_VERIFY_RECOVER	✖	✖	✖	✓[4]	✖	✖	✖	
CKA_WRAP	✖	✖	✖	✓	✓	✓	✖	False
CKA_WRAP_TEMPLATE	✖	✓	✖	✓	✓	✓	✖	
CKA_TRUSTED	✖	✓	✖	✓	✓	✓	✖	False
CKA_WRAP_WITH_TRUSTED	✓	✖	✓	✖	✓	✓	✓	False
CKA_UNWRAP	✖	✖	✓	✖	✓	✓	✖	False
CKA_UNWRAP_TEMPLATE	✓	✖	✓	✖	✓	✓	✖	

Atributo	Tipo de clave							Default Value (Valor predeterminado)
CKA_SENSITIVE	✓	✖	✓	✖	✓	✓	✓	True
CKA_ALWAYS_SENSITIVE	R	✖	R	✖	R	R	R	
CKA_EXTRACTABLE	✓	✖	✓	✖	✓	✓	✓	True
CKA_NEVER_EXTRACTABLE	R	✖	R	✖	R	R	R	
CKA_MODULUS	✖	✖	✓[2]	✓[2]	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	✖	✖	✖	✖	
CKA_PRIME_1	✖	✖	✓	✖	✖	✖	✖	
CKA_PRIME_2	✖	✖	✓	✖	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✓	✖	✖	✖	✖	
CKA_EXPONENT_1	✖	✖	✓	✖	✖	✖	✖	

Atributo	Tipo de clave							Default Value (Valor predeterminado)
CKA_EXPONENT_2	✖	✖	✓	✖	✖	✖	✖	
CKA_PRIVATE_EXPONENT	✖	✖	✓[2]	✖	✖	✖	✖	
CKA_PUBLIC_EXPONENT	✖	✖	✓[2]	✓[2]	✖	✖	✖	
CKA_EC_PARAMS	✓[2]	✓[2]	✖	✖	✖	✖	✖	
CKA_EC_POINT	✖	✓[2]	✖	✖	✖	✖	✖	
CKA_VALUE	✓[2]	✖	✖	✖	✓[2]	✓[2]	✓[2]	
CKA_VALUE_LEN	✖	✖	✖	✖	✖	✖	✖	
CKA_CHECK_VALUE	R	R	R	R	R	R	R	

UnwrapKey

Atributo	Tipo de clave					Default Value (Valor predeterminado)
	EC privada	RSA privada	AES	DES3	Secreto genérico	
CKA_CLASS	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	
CKA_KEY_TYPE	✓[2]	✓[2]	✓[2]	✓[2]	✓[2]	
CKA_LABEL	✓	✓	✓	✓	✓	
CKA_ID	✓	✓	✓	✓	✓	
CKA_LOCAL	R	R	R	R	R	False
CKA_TOKEN	✓	✓	✓	✓	✓	False
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT	✖	✖	✓	✓	✖	False
CKA_DECRYPT	✖	✓	✓	✓	✖	False
CKA_DERIVE	✓	✓	✓	✓	✓	False

Atributo	Tipo de clave					Default Value (Valor predeterminado)
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	True
CKA_DESTROYABLE	✓	✓	✓	✓	✓	True
CKA_SIGN	✓	✓	✓	✓	✓	False
CKA_SIGN_RECOVER	✖	✓[3]	✖	✖	✖	False
CKA_VERIFY	✖	✖	✓	✓	✓	False
CKA_VERIFY_RECOVER	✖	✖	✖	✖	✖	
CKA_WRAP	✖	✖	✓	✓	✖	False
CKA_UNWRAP	✖	✓	✓	✓	✖	False
CKA_SENSITIVE	✓	✓	✓	✓	✓	True
CKA_EXTRACTABLE	✓	✓	✓	✓	✓	True
CKA_NEVER_EXTRACTABLE	R	R	R	R	R	

Atributo	Tipo de clave					Default Value (Valor predeterminado)
CKA_ALWAYS_SENSITIVE	R	R	R	R	R	
CKA_MODULUS	✖	✖	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	✖	✖	
CKA_PRIME_1	✖	✖	✖	✖	✖	
CKA_PRIME_2	✖	✖	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✖	✖	✖	
CKA_EXPONENT_1	✖	✖	✖	✖	✖	
CKA_EXPONENT_2	✖	✖	✖	✖	✖	
CKA_PRIVATE_EXPONENT	✖	✖	✖	✖	✖	
CKA_PUBLIC_EXPONENT	✖	✖	✖	✖	✖	

Atributo	Tipo de clave					Default Value (Valor predeterminado)
CKA_EC_PARAMS		✖	✖	✖	✖	✖	
CKA_EC_POINT		✖	✖	✖	✖	✖	
CKA_VALUE		✖	✖	✖	✖	✖	
CKA_VALUE_LEN		✖	✖	✖	✖	✖	
CKA_CHECK_VALUE		R	R	R	R	R	

DeriveKey

Atributo	Tipo de clave				Default Value (Valor predeterminado)
		AES	DES3	Secreto genérico	
CKA_CLASS		✓[2]	✓[2]	✓[2]	
CKA_KEY_TYPE		✓[2]	✓[2]	✓[2]	
CKA_LABEL		✓	✓	✓	

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_ID	✓	✓	✓	
CKA_LOCAL	R	R	R	True
CKA_TOKEN	✓	✓	✓	False
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	True
CKA_ENCRYPT	✓	✓	✖	False
CKA_DECRYPT	✓	✓	✖	False
CKA_DERIVE	✓	✓	✓	False
CKA_MODIFIABLE	✓[1]	✓[1]	✓[1]	True
CKA_DESTROYABLE	✓[1]	✓[1]	✓[1]	True
CKA_SIGN	✓	✓	✓	False
CKA_SIGN_RECOVER	✖	✖	✖	
CKA_VERIFY	✓	✓	✓	False
CKA_VERIFY_RECOVER	✖	✖	✖	

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_WRAP	✓	✓	✖	False
CKA_UNWRAP	✓	✓	✖	False
CKA_SENSITIVE	✓	✓	✓	True
CKA_EXTRACTABLE	✓	✓	✓	True
CKA_NEVER_EXTRACTABLE	R	R	R	
CKA_ALWAYS_SENSITIVE	R	R	R	
CKA_MODULUS	✖	✖	✖	
CKA_MODULUS_BITS	✖	✖	✖	
CKA_PRIME_1	✖	✖	✖	
CKA_PRIME_2	✖	✖	✖	
CKA_COEFFICIENT	✖	✖	✖	

Atributo	Tipo de clave			Default Value (Valor predeterminado)
CKA_EXPONENT_1	✖	✖	✖	
CKA_EXPONENT_2	✖	✖	✖	
CKA_PRIVATE_EXPONENT	✖	✖	✖	
CKA_PUBLIC_EXPONENT	✖	✖	✖	
CKA_EC_PARAMS	✖	✖	✖	
CKA_EC_POINT	✖	✖	✖	
CKA_VALUE	✖	✖	✖	
CKA_VALUE_LEN	✓[2]	✖	✓[2]	
CKA_CHECK_VALUE	R	R	R	

GetAttributeValue

Atributo	Tipo de clave						
	EC privada	EC pública	RSA privada	RSA pública	AES	DES3	Secreto genérico
CKA_CLASS	✓	✓	✓	✓	✓	✓	✓
CKA_KEY_TYPE	✓	✓	✓	✓	✓	✓	✓
CKA_LABEL	✓	✓	✓	✓	✓	✓	✓
CKA_ID	✓	✓	✓	✓	✓	✓	✓
CKA_LOCAL	✓	✓	✓	✓	✓	✓	✓
CKA_TOKEN	✓	✓	✓	✓	✓	✓	✓
CKA_PRIVATE	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]	✓[1]
CKA_ENCRYPT	✖	✖	✖	✓	✓	✓	✖
CKA_DECRYPT	✖	✖	✓	✖	✓	✓	✖
CKA_DERIVE	✓	✓	✓	✓	✓	✓	✓
CKA_MODIFIABLE	✓	✓	✓	✓	✓	✓	✓

Atributo	Tipo de clave						
CKA_DESTROYABLE	✓	✓	✓	✓	✓	✓	✓
CKA_SIGN	✓	✖	✓	✖	✓	✓	✓
CKA_SIGN_RECOVER	✖	✖	✓	✖	✖	✖	✖
CKA_VERIFY	✖	✓	✖	✓	✓	✓	✓
CKA_VERIFY_RECOVER	✖	✖	✖	✓	✖	✖	✖
CKA_WRAP	✖	✖	✖	✓	✓	✓	✖
CKA_WRAP_TEMPLATE	✖	✓	✖	✓	✓	✓	✖
CKA_TRUSTED	✖	✓	✖	✓	✓	✓	✓
CKA_WRAP_WITH_TRUSTED	✓	✖	✓	✖	✓	✓	✓
CKA_UNWRAP	✖	✖	✓	✖	✓	✓	✖
CKA_UNWRAP_TEMPLATE	✓	✖	✓	✖	✓	✓	✖
CKA_SENSITIVE	✓	✖	✓	✖	✓	✓	✓

Atributo	Tipo de clave						
CKA_EXTRACTABLE	✓	✖	✓	✖	✓	✓	✓
CKA_NEVER_EXTRACTABLE	✓	✖	✓	✖	✓	✓	✓
CKA_ALWAYS_SENSITIVE	R	R;	R	R	R	R	R
CKA_MODULUS	✖	✖	✓	✓	✖	✖	✖
CKA_MODULUS_BITS	✖	✖	✖	✓	✖	✖	✖
CKA_PRIME_1	✖	✖	S	✖	✖	✖	✖
CKA_PRIME_2	✖	✖	S	✖	✖	✖	✖
CKA_COEFFICIENT	✖	✖	S	✖	✖	✖	✖
CKA_EXPONENT_1	✖	✖	S	✖	✖	✖	✖
CKA_EXPONENT_2	✖	✖	S	✖	✖	✖	✖
CKA_PRIVATE_EXPONENT	✖	✖	S	✖	✖	✖	✖

Atributo	Tipo de clave						
CKA_PUBLIC_EXPONENT	✖	✖	✓	✓	✖	✖	✖
CKA_EC_PARAMS	✓	✓	✖	✖	✖	✖	✖
CKA_EC_POINT	✖	✓	✖	✖	✖	✖	✖
CKA_VALUE	S	✖	✖	✖	✓[2]	✓[2]	✓[2]
CKA_VALUE_LEN	✖	✖	✖	✖	✓	✖	✓
CKA_CHECK_VALUE	✓	✓	✓	✓	✓	✓	✖

Notas de atributo

- [1] Este atributo es parcialmente compatible con el firmware y debe configurarse de forma explícita únicamente en el valor predeterminado.
- [2] Atributo obligatorio.
- [3] Solo SDK 3 de cliente. El atributo `CKA_SIGN_RECOVER` se obtiene del atributo `CKA_SIGN`. Si debe configurarse, solo se puede establecer en el mismo valor que el establecido para `CKA_SIGN`. Si no se establece, se obtiene el valor predeterminado de `CKA_SIGN`. Como CloudHSM solo admite los mecanismos de firma recuperable basados en RSA, este atributo solo se puede aplicar a clases públicas de RSA actualmente.
- [4] Solo SDK 3 de cliente. El atributo `CKA_VERIFY_RECOVER` se obtiene del atributo `CKA_VERIFY`. Si debe configurarse, solo se puede establecer en el mismo valor que el establecido para `CKA_VERIFY`. Si no se establece, se obtiene el valor predeterminado de `CKA_VERIFY`. Como CloudHSM solo admite los mecanismos de firma recuperable basados en RSA, este atributo solo se puede aplicar a clases públicas de RSA actualmente.

Modificación de atributos

Algunos atributos de un objeto se pueden modificar una vez creado el objeto, mientras que otros no. Para modificar los atributos, utilice el comando setAttribute de cloudhsm_mgmt_util. También puede generar una lista de atributos y las constantes que los representan mediante el comando listAttribute de cloudhsm_mgmt_util.

En la siguiente lista se muestran los atributos que se pueden modificar después de crear un objeto:

- `CKA_LABEL`
- `CKA_TOKEN`

> **Note**
>
> La modificación solo se permite para cambiar una clave de sesión por una clave de token. Utilice el comando setAttribute de key_mgmt_util para cambiar el valor del atributo.

- `CKA_ENCRYPT`
- `CKA_DECRYPT`
- `CKA_SIGN`
- `CKA_VERIFY`
- `CKA_WRAP`
- `CKA_UNWRAP`
- `CKA_LABEL`
- `CKA_SENSITIVE`
- `CKA_DERIVE`

> **Note**
>
> Este atributo admite la derivación de claves. Debe ser `False` para todas las claves públicas y no puede establecerse en `True`. Para las claves secretas y privadas de EC, se puede establecer en `True` o `False`.

- `CKA_TRUSTED`

> **Note**
>
> Este atributo se puede establecer en `True` o `False` solo mediante Crypto Officer (CO).

- `CKA_WRAP_WITH_TRUSTED`

> **Note**
>
> Aplique este atributo a una clave de datos exportable para especificar que solo puede encapsular esta clave con claves marcadas como `CKA_TRUSTED`. Una vez establecido `CKA_WRAP_WITH_TRUSTED` como true, el atributo pasa a ser de solo lectura y no se puede cambiar ni eliminar.

Interpretación de los códigos de error

La especificación en la plantilla de un atributo que no es compatible con una clave específica produce un error. La siguiente tabla contiene los códigos de error que se generan cuando se infringen las especificaciones:

Código de error	Descripción
`CKR_TEMPLATE_INCONSISTENT`	Este error aparece cuando se especifica un atributo en la plantilla de atributos que cumple la especificación PKCS #11, pero no es compatible con CloudHSM.
`CKR_ATTRIBUTE_TYPE_INVALID`	Recibirá este error cuando recupere un valor de un atributo que cumple la especific ación PKCS #11, pero no es compatible con CloudHSM.
`CKR_ATTRIBUTE_INCOMPLETE`	Este error aparece cuando no se especifica el atributo obligatorio en la plantilla de atributos.

Código de error	Descripción
`CKR_ATTRIBUTE_READ_ONLY`	Este error aparece cuando se especifica un atributo de solo lectura en la plantilla de atributos.

Ejemplos de código para la biblioteca PKCS #11 (SDK 3 de cliente)

Los ejemplos de código que aparecen a continuación GitHub muestran cómo realizar tareas básicas con la biblioteca PKCS #11.

Requisitos previos para el código de muestra

Antes de ejecutar las muestras, siga estos pasos para configurar su entorno:

- Instale y configure la biblioteca PKCS #11 para SDK 3 de cliente.
- Configure un usuario de criptografía (CU). La aplicación usa esta cuenta de HSM para ejecutar los ejemplos de código en el HSM.

Ejemplos de código

Los ejemplos de código de la biblioteca de AWS CloudHSM software de PKCS #11 están disponibles en. GitHub Este repositorio contiene ejemplos acerca de cómo realizar operaciones comunes con PKCS#11, como el cifrado, el descifrado, la firma y la verificación.

- Generar claves (AES, RSA, EC)
- Mostrar atributos de clave
- Cifrado y descifrado de datos con AES-GCM
- Cifrado y descifrado de datos con AES_CTR
- Cifrado y descifrado de datos con 3DES
- Firmar y verificar datos con RSA
- Derivar claves usando HMAC KDF
- Encapsule y desencapsule las claves con AES utilizando el relleno PKCS #5
- Encapsule y desencapsule las claves con AES sin relleno
- Encapsule y desencapsule las claves con AES usando cero relleno

- Encapsulamiento y desencapsulamiento de claves con AES-GCM
- Cómo encapsular y desencapsular claves con RSA

Instalación del SDK 3 de cliente para el motor dinámico de OpenSSL

El SDK 3 de cliente requiere un daemon del cliente para conectarse al clúster. Es compatible con:

- Generación de claves RSA para claves de 2048, 3072 y 4096 bits
- Firma o verificación de RSA
- Cifrado o descifrado de RSA
- Generación de números aleatorios criptográficamente segura y validada por FIPS

Temas

- Requisitos previos para el motor dinámico OpenSSL con SDK 3 de cliente
- Instalar el motor dinámico de OpenSSL para SDK 3 de cliente
- Utilizar el motor dinámico de OpenSSL para el SDK 3 de cliente

Requisitos previos para el motor dinámico OpenSSL con SDK 3 de cliente

Para obtener información acerca de las plataformas admitidas, consulte Plataformas compatibles con SDK 3 de cliente.

Antes de poder utilizar el motor AWS CloudHSM dinámico para OpenSSL, necesita el cliente. AWS CloudHSM

El cliente es un daemon que establece una comunicación end-to-end cifrada con los HSM del clúster y el motor OpenSSL se comunica localmente con el cliente. Para instalar y configurar el cliente, consulte. AWS CloudHSM Instalar el cliente (Linux) Luego use el siguiente comando para iniciarlo.

Amazon Linux

```
$ sudo start cloudhsm-client
```

Amazon Linux 2

```
$ sudo systemctl cloudhsm-client start
```

CentOS 6

```
$ sudo systemctl start cloudhsm-client
```

CentOS 7

```
$ sudo systemctl cloudhsm-client start
```

RHEL 6

```
$ sudo systemctl start cloudhsm-client
```

RHEL 7

```
$ sudo systemctl cloudhsm-client start
```

Ubuntu 16.04 LTS

```
$ sudo systemctl cloudhsm-client start
```

Instalar el motor dinámico de OpenSSL para SDK 3 de cliente

Los siguientes pasos describen cómo instalar y configurar el motor AWS CloudHSM dinámico para OpenSSL. Para obtener información acerca de la actualización, consulte Actualización de SDK 3 de cliente.

Instalación y configuración el motor OpenSSL

1. Utilice los comandos siguientes para descargar e instalar el motor de OpenSSL.

   Amazon Linux

   ```
   $ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-dyn-latest.el6.x86_64.rpm
   ```

   ```
   $ sudo yum install ./cloudhsm-client-dyn-latest.el6.x86_64.rpm
   ```

Amazon Linux 2

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-dyn-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-dyn-latest.el7.x86_64.rpm
```

CentOS 6

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-dyn-latest.el6.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-dyn-latest.el6.x86_64.rpm
```

CentOS 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-dyn-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-dyn-latest.el7.x86_64.rpm
```

RHEL 6

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-dyn-latest.el6.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-dyn-latest.el6.x86_64.rpm
```

RHEL 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-dyn-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-dyn-latest.el7.x86_64.rpm
```

Ubuntu 16.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/
cloudhsm-client-dyn_latest_amd64.deb
```

```
$ sudo apt install ./cloudhsm-client-dyn_latest_amd64.deb
```

El motor OpenSSL está instalado en `/opt/cloudhsm/lib/libcloudhsm_openssl.so`.

2. Utilice el siguiente comando para definir una variable de entorno denominada `n3fips_password` con las credenciales de un usuario de criptografía (CU).

```
$ export n3fips_password=<HSM user name>:<password>
```

Utilizar el motor dinámico de OpenSSL para el SDK 3 de cliente

Para utilizar el motor AWS CloudHSM dinámico de OpenSSL desde una aplicación integrada en OpenSSL, asegúrese de que la aplicación utilice el motor dinámico de OpenSSL denominado. `cloudhsm` La biblioteca compartida para el motor dinámico se encuentra en `/opt/cloudhsm/lib/libcloudhsm_openssl.so`.

Para utilizar el motor AWS CloudHSM dinámico para OpenSSL desde la línea de comandos de OpenSSL, utilice la `-engine` opción para especificar el nombre del motor dinámico de OpenSSL. `cloudhsm` Por ejemplo:

```
$ openssl s_server -cert server.crt -key server.key -engine cloudhsm
```

SDK 3 de cliente para proveedor de JCE

El proveedor AWS CloudHSM JCE es una implementación de proveedor creada a partir del marco de proveedores de Java Cryptographic Extension (JCE). El JCE le permite llevar a cabo operaciones criptográficas usando el kit de desarrollo de Java (JDK). En esta guía, el proveedor de AWS CloudHSM JCE a veces se denomina proveedor de JCE. Utilice el proveedor de JCE y el JDK para transferir las operaciones criptográficas de descarga al HSM.

Temas

- Instalar y usar el proveedor AWS CloudHSM JCE para Client SDK 3
- Mecanismos compatibles con SDK 3 de cliente
- Atributos de clave de Java compatibles con SDK 3 de cliente
- Ejemplos de código de la biblioteca de AWS CloudHSM software para Java for Client SDK 3
- Uso de la clase AWS CloudHSM KeyStore Java para Client SDK 3

Instalar y usar el proveedor AWS CloudHSM JCE para Client SDK 3

Para poder usar el proveedor de JCE, necesita el AWS CloudHSM cliente.

El cliente es un demonio que establece una comunicación end-to-end cifrada con los HSM del clúster. El proveedor de JCE se comunica localmente con el cliente. Si no ha instalado ni configurado el paquete de AWS CloudHSM cliente, hágalo ahora siguiendo los pasos que se indican en. Instalar el cliente (Linux) Después de instalar y configurar el cliente, utilice el siguiente comando para iniciarlo.

Nota: El proveedor JCE solo se admite en Linux y en sistemas operativos compatibles.

Amazon Linux

```
$ sudo start cloudhsm-client
```

Amazon Linux 2

```
$ sudo systemctl cloudhsm-client start
```

CentOS 7

```
$ sudo systemctl cloudhsm-client start
```

CentOS 8

```
$ sudo systemctl cloudhsm-client start
```

RHEL 7

```
$ sudo systemctl cloudhsm-client start
```

RHEL 8

```
$ sudo systemctl cloudhsm-client start
```

Ubuntu 16.04 LTS

```
$ sudo systemctl cloudhsm-client start
```

Ubuntu 18.04 LTS

```
$ sudo systemctl cloudhsm-client start
```

Ubuntu 20.04 LTS

```
$ sudo systemctl cloudhsm-client start
```

Temas

- Instalación del proveedor de JCE
- Validación de la instalación
- Cómo proporcionar credenciales al proveedor de JCE
- Aspectos básicos de gestión de claves en el proveedor de JCE

Instalación del proveedor de JCE

Utilice el siguiente comando para descargar e instalar el proveedor de JCE. Este proveedor solo se admite en Linux y en sistemas operativos compatibles.

> **Note**
>
> Para obtener actualizaciones, consulte Actualización de SDK 3 de cliente.

Amazon Linux

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL6/cloudhsm-client-jce-latest.el6.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-jce-latest.el6.x86_64.rpm
```

Amazon Linux 2

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-jce-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-jce-latest.el7.x86_64.rpm
```

CentOS 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-jce-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-jce-latest.el7.x86_64.rpm
```

CentOS 8

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-jce-latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-jce-latest.el8.x86_64.rpm
```

RHEL 7

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL7/cloudhsm-client-jce-latest.el7.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-jce-latest.el7.x86_64.rpm
```

RHEL 8

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/EL8/cloudhsm-client-jce-latest.el8.x86_64.rpm
```

```
$ sudo yum install ./cloudhsm-client-jce-latest.el8.x86_64.rpm
```

Ubuntu 16.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Xenial/cloudhsm-client-jce_latest_amd64.deb
```

```
$ sudo apt install ./cloudhsm-client-jce_latest_amd64.deb
```

Ubuntu 18.04 LTS

```
$ wget https://s3.amazonaws.com/cloudhsmv2-software/CloudHsmClient/Bionic/cloudhsm-client-jce_latest_u18.04_amd64.deb
```

```
$ sudo apt install ./cloudhsm-client-jce_latest_u18.04_amd64.deb
```

Después de ejecutar los comandos anteriores, encontrará los siguientes archivos en el proveedor de JCE:

- `/opt/cloudhsm/java/cloudhsm-version.jar`
- `/opt/cloudhsm/java/cloudhsm-test-version.jar`
- `/opt/cloudhsm/java/hamcrest-all-1.3.jar`
- `/opt/cloudhsm/java/junit.jar`
- `/opt/cloudhsm/java/log4j-api-2.17.1.jar`
- `/opt/cloudhsm/java/log4j-core-2.17.1.jar`
- `/opt/cloudhsm/lib/libcaviumjca.so`

Validación de la instalación

Realice operaciones básicas en el HSM para validar la instalación.

Cómo validar la instalación del proveedor de JCE

1. (Opcional) Si todavía no tiene instalado Java en su entorno, ejecute el comando siguiente para instalarlo.

Linux (and compatible libraries)

```
$ sudo yum install java-1.8.0-openjdk
```

Ubuntu

```
$ sudo apt-get install openjdk-8-jre
```

2. Utilice los siguientes comandos para definir las variables de entorno necesarias. Sustituya *<nombre de usuario de HSM>* y *<password>* con las credenciales de un usuario de criptografía (CU).

```
$ export LD_LIBRARY_PATH=/opt/cloudhsm/lib
```

```
$ export HSM_PARTITION=PARTITION_1
```

```
$ export HSM_USER=<HSM user name>
```

```
$ export HSM_PASSWORD=<password>
```

3. Utilice el siguiente comando para ejecutar la prueba de funcionalidad básica. Si se ejecuta correctamente, la salida del comando debería ser similar a la siguiente.

```
$ java8 -classpath "/opt/cloudhsm/java/*" org.junit.runner.JUnitCore
 TestBasicFunctionality

JUnit version 4.11
.2018-08-20 17:53:48,514 DEBUG [main] TestBasicFunctionality
 (TestBasicFunctionality.java:33) - Adding provider.
2018-08-20 17:53:48,612 DEBUG [main] TestBasicFunctionality
 (TestBasicFunctionality.java:42) - Logging in.
2018-08-20 17:53:48,612 INFO [main] cfm2.LoginManager (LoginManager.java:104) -
 Looking for credentials in HsmCredentials.properties
2018-08-20 17:53:48,612 INFO [main] cfm2.LoginManager (LoginManager.java:122) -
 Looking for credentials in System.properties
2018-08-20 17:53:48,613 INFO [main] cfm2.LoginManager (LoginManager.java:130) -
 Looking for credentials in System.env
 SDK Version: 2.03
```

```
2018-08-20 17:53:48,655 DEBUG [main] TestBasicFunctionality
 (TestBasicFunctionality.java:54) - Generating AES Key with key size 256.
2018-08-20 17:53:48,698 DEBUG [main] TestBasicFunctionality
 (TestBasicFunctionality.java:63) - Encrypting with AES Key.
2018-08-20 17:53:48,705 DEBUG [main] TestBasicFunctionality
 (TestBasicFunctionality.java:84) - Deleting AES Key.
2018-08-20 17:53:48,707 DEBUG [main] TestBasicFunctionality
 (TestBasicFunctionality.java:92) - Logging out.

Time: 0.205

OK (1 test)
```

Cómo proporcionar credenciales al proveedor de JCE

Los HSM necesitan autenticar la aplicación de Java antes de que la aplicación pueda utilizarlos. Cada aplicación puede utilizar una sesión. Los HSM autentican una sesión mediante el método de inicio de sesión explícito o implícito.

Inicio de sesión explícito: este método le permite proporcionar las credenciales de CloudHSM directamente en la aplicación. Utiliza el método `LoginManager.login()`, en el que se pasa el nombre de usuario y la contraseña del CU y el ID de la partición de HSM. Para obtener más información acerca de cómo utilizar el método de inicio de sesión explícito, consulte el ejemplo de código de inicio de sesión en un HSM.

Inicio de sesión implícito: este método le permite definir las credenciales de CloudHSM en un nuevo archivo de propiedades, en las propiedades del sistema o como variables de entorno.

- Nuevo archivo de propiedades: cree un nuevo archivo con el nombre `HsmCredentials.properties` y añádalo al `CLASSPATH` de su aplicación. El archivo debe contener lo siguiente:

```
HSM_PARTITION = PARTITION_1
HSM_USER = <HSM user name>
HSM_PASSWORD = <password>
```

- Propiedades del sistema: defina las credenciales mediante las propiedades del sistema al ejecutar la aplicación. En los siguientes ejemplos, se muestran dos maneras diferentes de hacerlo:

```
$ java -DHSM_PARTITION=PARTITION_1 -DHSM_USER=<HSM user name> -
DHSM_PASSWORD=<password>
```

```
System.setProperty("HSM_PARTITION","PARTITION_1");
System.setProperty("HSM_USER","<HSM user name>");
System.setProperty("HSM_PASSWORD","<password>");
```

- Variables de entorno: defina las credenciales como variables de entorno.

```
$ export HSM_PARTITION=PARTITION_1
$ export HSM_USER=<HSM user name>
$ export HSM_PASSWORD=<password>
```

Es posible que las credenciales no estén disponibles si la aplicación no las proporciona o si se intenta realizar una operación antes de que el HSM autentique la sesión. En esos casos, la biblioteca de software de CloudHSM para Java busca las credenciales en el orden que se indica a continuación:

1. `HsmCredentials.properties`
2. Propiedades del sistema
3. Variables de entorno

Control de errores

El control de errores es más fácil con el método de inicio de sesión explícito que con el de inicio de sesión implícito. Si utiliza la clase `LoginManager`, tendrá más control sobre el modo en que la aplicación gestiona los errores. Con el método de inicio de sesión implícito, la gestión de errores resulta difícil de comprender cuando las credenciales no son válidas o cuando los HSM tienen problemas en la sesión de autenticación.

Aspectos básicos de gestión de claves en el proveedor de JCE

Los aspectos básicos de la administración de claves en el proveedor de JCE están relacionados con la importación o la exportación de claves, la carga de claves por identificador o la eliminación de claves. Para obtener más información acerca de la administración de claves, consulte el ejemplo de código de administración de claves.

También puede encontrar más ejemplos de código de proveedor de JCE en Ejemplos de código.

Mecanismos compatibles con SDK 3 de cliente

Para obtener información sobre las interfaces y las clases de motor de la arquitectura criptográfica de Java (JCA) compatibles AWS CloudHSM, consulte los temas siguientes.

Temas

- Claves compatibles
- Cifrados compatibles
- Resúmenes compatibles
- Algoritmos de código de autenticación de mensajes basado en hash (HMAC) compatibles
- Mecanismos de firma y verificación compatibles
- Notas del mecanismo

Claves compatibles

La biblioteca de AWS CloudHSM software para Java permite generar los siguientes tipos de claves.

- AES: claves AES de 128, 192 y 256 bits.
- DESede: clave 3DES de 92 bits. Consulte la nota 1 que aparece a continuación para ver los próximos cambios.
- Pares de claves ECC para curvas de NIST secp256r1 (P-256), secp384r1 (P-384) y secp256k1 (Blockchain).
- RSA: claves RSA de 2048 a 4096 bits, en incrementos de 256 bits.

Además de los parámetros estándar, admitimos los siguientes parámetros para cada clave que se genere.

- Label: etiqueta de clave que puede utilizar para buscar claves.
- isExtractable: indica si la clave se puede exportar desde el HSM.
- isPersistent: indica si la clave permanece en el HSM cuando finaliza la sesión en curso.

> **Note**
>
> La versión 3.1 de la biblioteca de Java permite especificar parámetros con mayor detalle. Para obtener más información, consulte este artículo sobre los atributos de Java admitidos.

Cifrados compatibles

La biblioteca de AWS CloudHSM software para Java admite las siguientes combinaciones de algoritmo, modo y relleno.

Algoritmo	Mode	Rellenado	Notas
AES	CBC	`AES/CBC/NoPadding` `AES/CBC/PKCS5Padding`	Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE` .
AES	ECB	`AES/ECB/NoPadding` `AES/ECB/PKCS5Padding`	Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE` . Utiliza AES de transformación.
AES	CTR	`AES/CTR/NoPadding`	Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE` .
AES	GCM	`AES/GCM/NoPadding`	Implementa `Cipher.ENCRYPT_MODE` , `Cipher.DECRYPT_MOD`

Algoritmo	Mode	Rellenado	Notas
			`E` , `Cipher.WR AP_MODE` y `Cipher.UN WRAP_MODE` . Al realizar el cifrado AES-GCM, el HSM no tiene en cuenta el vector de inicializ ación (IV) de la solicitud y utiliza un IV que él mismo genera. Una vez que se ha completado la operación, deberá llamar a `Cipher.ge tIV()` para obtener el IV.

AESWrap	ECB	`AESWrap/ECB/ ZeroPadding`
		
		`AESWrap/ECB/ NoPadding`
		
		`AESWrap/ECB/ PKCS5Padding`
			Implementa `Cipher.WR AP_MODE` y `Cipher.UN WRAP_MODE` . Utiliza AES de transform ación.

Algoritmo	Mode	Rellenado	Notas

DESede (Triple DES)	CBC	`DESede/CBC/NoPadding`
		
		`DESede/CBC/PKCS5Padding`
			Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE` .
			
			Las rutinas de generación de claves aceptan un tamaño de 168 o 192 bits. Sin embargo, internamente, todas las claves DESede son de 192 bits.
			
			Consulte la nota 1 que aparece a continuación para ver los próximos cambios.

Algoritmo	Mode	Rellenado	Notas

DESede (Triple DES)	ECB	`DESede/ECB/NoPadding`
		
		`DESede/ECB/PKCS5Padding`
			Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE` .
			
			Las rutinas de generación de claves aceptan un tamaño de 168 o 192 bits. Sin embargo, internamente, todas las claves DESede son de 192 bits.
			
			Consulte la nota 1 que aparece a continuación para ver los próximos cambios.

RSA	ECB	`RSA/ECB/NoPadding`
		
		`RSA/ECB/PKCS1Padding`
			Implementa `Cipher.ENCRYPT_MODE` y `Cipher.DECRYPT_MODE` .
			
			Consulte la nota 1 que aparece a continuación para ver los próximos cambios.

Algoritmo	Mode	Rellenado	Notas

RSA	ECB	`RSA/ECB/OAEPPadding`
		
		`RSA/ECB/OAEPWithSHA-1ANDMGF1Padding`
		
		`RSA/ECB/OAEPWithSHA-224ANDMGF1Padding`
		
		`RSA/ECB/OAEPWithSHA-256ANDMGF1Padding`
		
		`RSA/ECB/OAEPWithSHA-384ANDMGF1Padding`
		
		`RSA/ECB/OAEPWithSHA-512ANDMGF1Padding`
			Implementa `Cipher.ENCRYPT_MODE`, `Cipher.DECRYPT_MODE`, `Cipher.WRAP_MODE` y `Cipher.UNWRAP_MODE`.
			
			`OAEPPadding` es OAEP con el tipo de rellenado `SHA-1`.

RSAAESWrap	ECB	`OAEPPADDING`	Implementa `Cipher.WRAP_Mode` y `Cipher.UNWRAP_MODE`.

Resúmenes compatibles

La biblioteca AWS CloudHSM de software para Java admite los siguientes resúmenes de mensajes.

- `SHA-1`
- `SHA-224`
- `SHA-256`
- `SHA-384`
- `SHA-512`

> **Note**
>
> A los datos con una longitud inferior a 16 KB se les aplica la función hash en el HSM, mientras que a los de mayor tamaño se les aplica localmente mediante software.

Algoritmos de código de autenticación de mensajes basado en hash (HMAC) compatibles

La biblioteca AWS CloudHSM de software para Java admite los siguientes algoritmos HMAC.

- `HmacSHA1`
- `HmacSHA224`
- `HmacSHA256`
- `HmacSHA384`
- `HmacSHA512`

Mecanismos de firma y verificación compatibles

La biblioteca de AWS CloudHSM software para Java admite los siguientes tipos de firma y verificación.

Tipos de firma RSA

- `NONEwithRSA`
- `SHA1withRSA`

- SHA224withRSA
- SHA256withRSA
- SHA384withRSA
- SHA512withRSA
- SHA1withRSA/PSS
- SHA224withRSA/PSS
- SHA256withRSA/PSS
- SHA384withRSA/PSS
- SHA512withRSA/PSS

Tipos de firma ECDSA

- NONEwithECDSA
- SHA1withECDSA
- SHA224withECDSA
- SHA256withECDSA
- SHA384withECDSA
- SHA512withECDSA

Notas del mecanismo

[1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Atributos de clave de Java compatibles con SDK 3 de cliente

En este tema, se describe cómo puede utilizar una extensión propia con la versión 3.1 de la biblioteca de Java para establecer atributos de clave. Utilice esta extensión para establecer los atributos de clave admitidos y sus valores durante estas operaciones:

- Generación de claves
- Importación de claves

- Desencapsulamiento de claves

> **Note**
>
> La extensión para establecer atributos de clave personalizados es una característica opcional. Si ya tiene un código que funciona en la versión 3.0 de la biblioteca de Java , no es necesario que lo modifique. Las claves que cree seguirán teniendo los mismos atributos que antes.

Temas

- Descripción de los atributos
- Atributos admitidos
- Configuración de atributos para claves
- Resumen global

Descripción de los atributos

Los atributos de clave se utilizan para especificar qué acciones se permiten en objetos relacionados con las claves, como claves públicas, privadas o secretas. Los atributos y valores de clave se definen durante las operaciones de creación de objetos de clave.

Sin embargo, Java Cryptography Extension (JCE) no especifica cómo deben establecerse los valores de los atributos de clave, por lo que, de forma predeterminada, se permiten la mayoría de las acciones. Por el contrario, el estándar PKCS #11 define un completo conjunto de atributos con valores predeterminados más restrictivos. A partir de la versión 3.1 de la biblioteca de Java, CloudHSM cuenta con una extensión propia que permite establecer valores más restrictivos para los atributos que se usan habitualmente.

Atributos admitidos

Puede establecer valores para los atributos que aparecen en la tabla siguiente. Es recomendable que solamente establezca valores para los atributos que desee hacer más restrictivos. Si no se especifica ningún valor, CloudHSM utilizará el valor predeterminado que se indica en la tabla siguiente. Las celdas vacías de la columna «Valor predeterminado» indican que no hay ningún valor predeterminado específico asignado al atributo.

Atributo	Valor predeterminado			Notas
	Clave simétrica	Clave pública del par de claves	Clave privada del par de claves	
CKA_TOKEN	FALSE	FALSE	FALSE	Clave permanente que se replica en todos los HSM del clúster y se incluye en las copias de seguridad. CKA_TOKEN = FALSE implica el uso de una clave de sesión, que se carga en un único HSM y se borra automáticamente cuando se interrumpe la conexión con ese HSM.
CKA_LABEL				Cadena definida por el usuario. Permite identificar fácilmente las claves en el HSM.
CKA_EXTRACTABLE	TRUE		TRUE	True indica que esta clave se puede exportar fuera del HSM.

Atributo	Valor predeterminado			Notas
CKA_ENCRYPT	TRUE	TRUE		True indica que la clave se puede utilizar para cifrar cualquier búfer.
CKA_DECRYPT	TRUE		TRUE	True indica que la clave se puede utilizar para descifrar cualquier búfer. Por lo general, cuando una clave tiene el valor True en CKA_WRAP, se utiliza el valor FALSE.
CKA_WRAP	TRUE	TRUE		True indica que la clave se puede utilizar para encapsular otra clave. Por lo general, se utilizará el valor FALSE con las claves privadas.
CKA_UNWRAP	TRUE		TRUE	True indica que la clave se puede utilizar para desencapsular (importar) otra clave.

Atributo	Valor predeterminado			Notas
CKA_SIGN	TRUE		TRUE	True indica que la clave se puede utilizar para firmar un resumen del mensaje. Normalmente, se utiliza el valor FALSE con las claves públicas y privadas que se han archivado.
CKA_VERIFY	TRUE	TRUE		True indica que la clave se puede utilizar para verificar una firma. Normalmente, se utiliza el valor FALSE con las claves privadas.

Atributo	Valor predeterminado			Notas
CKA_PRIVATE	TRUE	TRUE	TRUE	True indica que es posible que los usuarios no tengan acceso a la clave hasta que se autentiqu en. Es decir, los usuarios no pueden acceder a ninguna clave de CloudHSM hasta que se autentican, incluso aunque este atributo esté establecido en FALSE.

> **Note**
>
> La compatibilidad con los atributos de la biblioteca PKCS #11 es más amplia. Para obtener más información, consulte Atributos de PKCS #11 admitidos.

Configuración de atributos para claves

`CloudHsmKeyAttributesMap` es un objeto similar a Java Map, que puede usar para establecer valores de atributo en los objetos de clave. Los métodos de la función `CloudHsmKeyAttributesMap` son iguales que los métodos que se utilizan para manipular mapas de Java.

Si desea establecer valores personalizados en los atributos, tiene dos opciones:

- Utilizar los métodos que se indican en la tabla siguiente
- Utilizar los modelos de Builder que se ilustran más adelante en este documento

Los objetos de mapa de atributos admiten los siguientes métodos para establecer atributos:

Operación	Valor de retorno	Método de **CloudHSMKeyAttributesMap**
Obtener el valor de un atributo de clave para una clave existente	Objeto (que contiene el valor) o null	get(keyAttribute)
Rellenar el valor de un atributo de clave	Valor anterior asociado con el atributo de clave o null si no había ninguna asignación de un atributo de clave	put(keyAttribute, valor)
Rellenar valores en varios atributos de clave	N/A	PuTall () keyAttributesMap
Eliminar un par clave-valor del mapa de atributos	Valor anterior asociado con el atributo de clave o null si no había ninguna asignación de un atributo de clave	remove(keyAttribute)

> **Note**
>
> Los atributos que no se especifican explícitamente se establecen en los valores predeterminados que se indican en la tabla anterior de the section called “Atributos admitidos”.

Ejemplo de modelo de Builder

Por lo general, a los desarrolladores les resultará más cómodo utilizar las clases a través del modelo Builder. Por ejemplo:

```
import com.amazonaws.cloudhsm.CloudHsmKeyAttributes;
import com.amazonaws.cloudhsm.CloudHsmKeyAttributesMap;
import com.amazonaws.cloudhsm.CloudHsmKeyPairAttributesMap;
```

```
CloudHsmKeyAttributesMap keyAttributesSessionDecryptionKey =
   new CloudHsmKeyAttributesMap.Builder()
      .put(CloudHsmKeyAttributes.CKA_LABEL, "ExtractableSessionKeyEncryptDecrypt")
      .put(CloudHsmKeyAttributes.CKA_WRAP, false)
      .put(CloudHsmKeyAttributes.CKA_UNWRAP, false)
      .put(CloudHsmKeyAttributes.CKA_SIGN, false)
      .put(CloudHsmKeyAttributes.CKA_VERIFY, false)
      .build();

CloudHsmKeyAttributesMap keyAttributesTokenWrappingKey =
   new CloudHsmKeyAttributesMap.Builder()
      .put(CloudHsmKeyAttributes.CKA_LABEL, "TokenWrappingKey")
      .put(CloudHsmKeyAttributes.CKA_TOKEN, true)
      .put(CloudHsmKeyAttributes.CKA_ENCRYPT, false)
      .put(CloudHsmKeyAttributes.CKA_DECRYPT, false)
      .put(CloudHsmKeyAttributes.CKA_SIGN, false)
      .put(CloudHsmKeyAttributes.CKA_VERIFY, false)
      .build();
```

Es posible que los desarrolladores utilicen también conjunto de atributos para aplicar con mayor comodidad las prácticas recomendadas para las plantillas de claves. Por ejemplo:

```
//best practice template for wrapping keys

CloudHsmKeyAttributesMap commonKeyAttrs = new CloudHsmKeyAttributesMap.Builder()
    .put(CloudHsmKeyAttributes.CKA_EXTRACTABLE, false)
    .put(CloudHsmKeyAttributes.CKA_DECRYPT, false)
    .build();

// initialize a new instance of CloudHsmKeyAttributesMap by copying commonKeyAttrs
// but with an appropriate label

CloudHsmKeyAttributesMap firstKeyAttrs = new CloudHsmKeyAttributesMap(commonKeyAttrs);
firstKeyAttrs.put(CloudHsmKeyAttributes.CKA_LABEL, "key label");

// alternatively, putAll() will overwrite existing values to enforce conformance

CloudHsmKeyAttributesMap secondKeyAttrs = new CloudHsmKeyAttributesMap();
secondKeyAttrs.put(CloudHsmKeyAttributes.CKA_DECRYPT, true);
secondKeyAttrs.put(CloudHsmKeyAttributes.CKA_ENCRYPT, true);
secondKeyAttrs.put(CloudHsmKeyAttributes.CKA_LABEL, “safe wrapping key”);
secondKeyAttrs.putAll(commonKeyAttrs); // will overwrite CKA_DECRYPT to be FALSE
```

Configuración de atributos para un par de claves

Utilice la clase `CloudHsmKeyPairAttributesMap` de Java para administrar los atributos de clave de un par de claves. `CloudHsmKeyPairAttributesMap` encapsula dos objetos `CloudHsmKeyAttributesMap`: uno para una clave pública y otro para una clave privada.

Para establecer por separado atributos específicos en la clave pública y en la clave privada, puede utilizar el método `put()` en el objeto de mapa `CloudHsmKeyAttributes` que corresponda a esa clave. Utilice el método `getPublic()` para recuperar el mapa de atributos de la clave pública y utilice `getPrivate()` para recuperar el mapa de atributos de la clave privada. Puede rellenar el valor de varios atributos de clave a la vez tanto de pares de claves públicas como de pares de claves privadas utilizando `putAll()` con un mapa de atributos de pares de claves como argumento.

Ejemplo de modelo de Builder

Por lo general, a los desarrolladores les resultará más cómodo establecer los atributos de clave a través del modelo Builder. Por ejemplo:

```
import com.amazonaws.cloudhsm.CloudHsmKeyAttributes;
import com.amazonaws.cloudhsm.CloudHsmKeyAttributesMap;
import com.amazonaws.cloudhsm.CloudHsmKeyPairAttributesMap;

//specify attributes up-front
CloudHsmKeyAttributesMap keyAttributes =
    new CloudHsmKeyAttributesMap.Builder()
        .put(CloudHsmKeyAttributes.CKA_SIGN, false)
        .put(CloudHsmKeyAttributes.CKA_LABEL, "PublicCertSerial12345")
        .build();

CloudHsmKeyPairAttributesMap keyPairAttributes =
    new CloudHsmKeyPairAttributesMap.Builder()
        .withPublic(keyAttributes)
        .withPrivate(
            new CloudHsmKeyAttributesMap.Builder() //or specify them inline
                .put(CloudHsmKeyAttributes.CKA_LABEL, "PrivateCertSerial12345")
                .put (CloudHSMKeyAttributes.CKA_WRAP, FALSE)
                .build()
        )
        .build();
```

> **Note**
>
> Para obtener más información acerca de esta extensión propietaria, consulte el archivo Javadoc y el ejemplo en. GitHub Para ver el Javadoc, descargue y abra el archivo.

Resumen global

Para especificar atributos de clave con operaciones de claves, siga estos pasos:

1. Crear una instancia de `CloudHsmKeyAttributesMap` para las claves simétricas o de `CloudHsmKeyPairAttributesMap` para los pares de claves.
2. Defina el objeto attributes en el paso 1 con los atributos de clave y los valores que corresponda.
3. Cree una instancia de la clase `Cavium*ParameterSpec` que corresponda al tipo de clave específico y pase este objeto de atributos configurado a su constructor.
4. Pase este objeto `Cavium*ParameterSpec` a la clase o el método criptográfico que corresponda.

Si necesita más información, la tabla siguiente contiene las clases y métodos de `Cavium*ParameterSpec` que admiten atributos de clave personalizados.

Tipo de clave	Clase de especificación de parámetros	Constructores de ejemplo
Clase base	`CaviumKeyGenAlgorithmParameterSpec`	`CaviumKeyGenAlgorithmParameterSpec(CloudHsmKeyAttributesMap keyAttributesMap)`
DES	`CaviumDESKeyGenParameterSpec`	`CaviumDESKeyGenParameterSpec(int keySize, byte[] iv, CloudHsmKeyAttributesMap keyAttributesMap)`

Tipo de clave	Clase de especificación de parámetros	Constructores de ejemplo
RSA	`CaviumRSAKeyGenParameterSpec`	`CaviumRSAKeyGenParameterSpec(int keysize, BigInteger publicExponent, CloudHsmKeyPairAttributesMap keyPairAttributesMap)`
secreta	`CaviumGenericSecretKeyGenParameterSpec`	`CaviumGenericSecretKeyGenParameterSpec(int size, CloudHsmKeyAttributesMap keyAttributesMap)`
AES	`CaviumAESKeyGenParameterSpec`	`CaviumAESKeyGenParameterSpec(int keySize, byte[] iv, CloudHsmKeyAttributesMap keyAttributesMap)`
EC	`CaviumECGenParameterSpec`	`CaviumECGenParameterSpec(String stdName, CloudHsmKeyPairAttributesMap keyPairAttributesMap)`

Código de muestra: generar y encapsular una clave

En estos breves ejemplos de código, se muestran los pasos de dos operaciones diferentes: generación de claves y encapsulamiento de claves:

```
// Set up the desired key attributes

KeyGenerator keyGen = KeyGenerator.getInstance("AES", "Cavium");
CaviumAESKeyGenParameterSpec keyAttributes = new CaviumAESKeyGenParameterSpec(
    256,
    new CloudHsmKeyAttributesMap.Builder()
        .put(CloudHsmKeyAttributes.CKA_LABEL, "MyPersistentAESKey")
        .put(CloudHsmKeyAttributes.CKA_EXTRACTABLE, true)
        .put(CloudHsmKeyAttributes.CKA_TOKEN, true)
        .build()
);

// Assume we already have a handle to the myWrappingKey
// Assume we already have the wrappedBytes to unwrap

// Unwrap a key using Custom Key Attributes

CaviumUnwrapParameterSpec unwrapSpec = new
 CaviumUnwrapParameterSpec(myInitializationVector, keyAttributes);

Cipher unwrapCipher = Cipher.getInstance("AESWrap", "Cavium");
unwrapCipher.init(Cipher.UNWRAP_MODE, myWrappingKey, unwrapSpec);
Key unwrappedKey = unwrapCipher.unwrap(wrappedBytes, "AES", Cipher.SECRET_KEY);
```

Ejemplos de código de la biblioteca de AWS CloudHSM software para Java for Client SDK 3

Requisitos previos

Antes de ejecutar las muestras, debe configurar el entorno:

- Instale y configure el proveedor de la extensión criptográfica de Java (JCE) y el paquete del cliente de AWS CloudHSM.
- Configure un nombre de usuario y contraseña de HSM válidos. Los permisos del usuario criptográfico (CU) son suficientes para estas tareas. La aplicación utiliza estas credenciales para iniciar sesión en el HSM en cada ejemplo.
- Decida cómo proporcionar las credenciales al proveedor de JCE.

Ejemplos de código

Los siguientes ejemplos de código muestran cómo utilizar el proveedor de JCE de AWS CloudHSM para realizar tareas básicas. Hay más ejemplos de código disponibles en GitHub.

- Inicio de sesión en un HSM
- Administración de claves
- Generación de una clave AES
- Cifrado y descifrado con AES-GCM
- Cifrado y descifrado con AES-CTR
- Cifrado y descifrado con D3DES-ECB[1]
- Encapsulamiento y desencapsulamiento de claves con AES-GCM
- Cómo encapsular y desencapsular claves con AES
- Cómo encapsular y desencapsular claves con RSA
- Usar atributos clave admitidos
- Enumerar claves en el almacén de claves
- Usar el almacén de claves de CloudHSM
- Firma de mensajes en una muestra con varios subprocesos
- Firma y verificación con claves de EC

[1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Uso de la clase AWS CloudHSM KeyStore Java para Client SDK 3

La AWS CloudHSM **`KeyStore`** clase proporciona un almacén de claves PKCS12 específico que permite el acceso a AWS CloudHSM las claves a través de aplicaciones como keytool y jarsigner. Este almacén de claves puede almacenar certificados junto con datos de la clave y relacionar estos certificados con los datos de clave que están almacenados en AWS CloudHSM.

> **Note**
>
> Dado que los certificados son información pública y, para maximizar la capacidad de almacenamiento de claves criptográficas, no admite el almacenamiento de certificados en los HSM. AWS CloudHSM

La AWS CloudHSM `KeyStore` clase implementa la interfaz de proveedor `KeyStore` de servicios (SPI) de la extensión de criptografía de Java (JCE). Para obtener más información sobre su uso`KeyStore`, consulte Class. KeyStore

Elección del almacén de claves apropiado

El proveedor de la extensión criptográfica de AWS CloudHSM Java (JCE) incluye un almacén de claves de solo lectura y transferencia predeterminado que transfiere todas las transacciones al HSM. Este almacén de claves predeterminado es distinto del almacén de uso especial. AWS CloudHSM KeyStore En la mayoría de los casos, obtendrá mejor rendimiento en tiempo de ejecución con el valor predeterminado. Solo debe usarlo AWS CloudHSM KeyStore para aplicaciones en las que necesite soporte para certificados y operaciones basadas en certificados, además de delegar las operaciones clave al HSM.

Aunque los dos almacenes de claves utilizan el proveedor JCE para las operaciones, son entidades independientes y no intercambian información entre sí.

Cargue el almacén de claves predeterminado para la aplicación Java de la siguiente manera:

```
KeyStore ks = KeyStore.getInstance("Cavium");
```

Cargue el CloudHSM de uso especial de la siguiente manera: KeyStore

```
KeyStore ks = KeyStore.getInstance("CloudHSM")
```

Inicializando AWS CloudHSM KeyStore

Inicie sesión de AWS CloudHSM KeyStore la misma forma en que inicia sesión en el proveedor de JCE. Puede usar variables de entorno o el archivo de propiedades del sistema, y debe iniciar sesión antes de empezar a usar CloudHSM KeyStore. Para ver un ejemplo de inicio de sesión en un HSM mediante el proveedor JCE, consulte Inicio de sesión en un HSM.

Si lo desea, puede especificar una contraseña para cifrar el archivo PKCS12 local que contiene los datos del almacén de claves. Al crear el AWS CloudHSM almacén de claves, se establece la contraseña y se la proporciona cuando se utilizan los métodos load, set y get.

Cree una instancia de un nuevo objeto KeyStore CloudHSM de la siguiente manera:

```
ks.load(null, null);
```

Para escribir los datos del almacén de claves en un archivo, utilice el método `store`. A partir de ese momento, puede cargar el almacén de claves existente utilizando el método `load` con el archivo de origen y la contraseña de la siguiente manera:

```
ks.load(inputStream, password);
```

Usando AWS CloudHSM KeyStore

Por lo general, un objeto KeyStore CloudHSM se utiliza a través de una aplicación de terceros, como jarsigner o keytool. También puede obtener acceso al objeto directamente a través del código.

AWS CloudHSM KeyStore cumple con la KeyStore especificación de la clase JCE y proporciona las siguientes funciones.

- `load`

  Carga el almacén de claves a partir de la secuencia de entrada especificada. Si se estableció una contraseña al guardar el almacén de claves, debe proporcionarse esta misma contraseña para que la carga se realice correctamente. Establezca los dos parámetros en null para inicializar un nuevo almacén de claves vacío.

  ```
  KeyStore ks = KeyStore.getInstance("CloudHSM");
  ks.load(inputStream, password);
  ```

- `aliases`

  Devuelve una enumeración de los nombres de alias de todas las entradas de la instancia especificada del almacén de claves. Los resultados incluyen objetos almacenados localmente en el archivo PKCS12 y objetos residentes en el HSM.

  Código de muestra:

```
KeyStore ks = KeyStore.getInstance("CloudHSM");
for(Enumeration<String> entry = ks.aliases(); entry.hasMoreElements();)
 {
  String label = entry.nextElement();
  System.out.println(label);
 }
```

- `ContainsAlias`

  Devuelve true si el almacén de claves tiene acceso al menos a un objeto con el alias especificado. El almacén de claves comprueba los objetos almacenados localmente en el archivo PKCS12 y los objetos residentes en el HSM.

- `DeleteEntry`

  Elimina una entrada de certificado del archivo PKCS12 local. No se admite la eliminación de los datos clave almacenados en un HSM mediante el. AWS CloudHSM KeyStore Puede eliminar las claves con la herramienta key_mgmt_util de CloudHSM.

- `GetCertificate`

  Devuelve el certificado asociado a un alias, si está disponible. Si el alias no existe o hace referencia a un objeto que no es un certificado, la función devuelve NULL.

```
KeyStore ks = KeyStore.getInstance("CloudHSM");
Certificate cert = ks.getCertificate(alias)
```

- `GetCertificateAlias`

  Devuelve el nombre (alias) de la primera entrada del almacén de claves cuyos datos coinciden con el certificado especificado.

```
KeyStore ks = KeyStore.getInstance("CloudHSM");
String alias = ks.getCertificateAlias(cert)
```

- `GetCertificateChain`

  Devuelve la cadena de certificados asociada con el alias especificado. Si el alias no existe o hace referencia a un objeto que no es un certificado, la función devuelve NULL.

- `GetCreationDate`

Devuelve la fecha de creación de la entrada identificada por el alias especificado. Si no hay disponible ninguna fecha de creación, la función devuelve la fecha en la que el certificado pasó a ser válido.

- `GetKey`

  GetKey se pasa al HSM y devuelve un objeto clave correspondiente a la etiqueta dada. Como consulta `getKey` directamente al HSM, se puede utilizar para cualquier clave del HSM, independientemente de si fue generada por el. KeyStore

  ```
  Key key = ks.getKey(keyLabel, null);
  ```

- `IsCertificateEntry`

  Comprueba si la entrada con el alias especificado representa una entrada de certificado.

- `IsKeyEntry`

  Comprueba si la entrada con el alias especificado representa una entrada de clave. La acción busca el alias tanto en el archivo PKCS12 como en el HSM.

- `SetCertificateEntry`

  Asigna el certificado especificado al alias proporcionado. Si el alias proporcionado ya se utiliza para identificar una clave o un certificado, se inicia una excepción `KeyStoreException`. Puede utilizar el código JCE para obtener el objeto clave y, a continuación, utilizar el KeyStore `SetKeyEntry` método para asociar el certificado a la clave.

- `SetKeyEntry` con una clave `byte[]`

  Actualmente, esta API no es compatible con SDK 3 de cliente.

- `SetKeyEntry` con un objeto `Key`

  Asigna la clave especificada al alias proporcionado y la almacena dentro del HSM. Si el objeto `Key` no es de tipo `CaviumKey`, la clave se importa en el HSM como clave de sesión extraíble.

  Si el objeto `Key` es de tipo `PrivateKey`, debe ir acompañado de la cadena de certificados correspondiente.

  Si el alias ya existe, la llamada a `SetKeyEntry` inicia una excepción `KeyStoreException` y evita que la clave se sobrescriba. Si es necesario sobrescribir la clave, utilice KMU o JCE para ese propósito.

- `EngineSize`

  Devuelve el número de entradas del almacén de claves.

- `Store`

  Guarda el almacén de claves en el flujo de salida especificado como un archivo PKCS12 y lo protege con la contraseña proporcionada. Además, conserva todas las claves cargadas (que se establecen mediante llamadas a `setKey`).

Integración de las aplicaciones de terceros con AWS CloudHSM

Algunos de los casos de uso AWS CloudHSM implican la integración de aplicaciones de software de terceros con el HSM de su AWS CloudHSM clúster. Al integrar software de terceros AWS CloudHSM, puede lograr una variedad de objetivos relacionados con la seguridad. En los temas siguientes se describe cómo conseguir algunos de dichos objetivos.

Temas

- Mejore la seguridad de su servidor web con la descarga de SSL/TLS en AWS CloudHSM
- Configuración de Windows Server como entidad de certificación (CA) con AWS CloudHSM
- Cifrado de datos transparente (TDE) de Oracle Database con AWS CloudHSM
- Usa Microsoft SignTool con AWS CloudHSM para firmar archivos
- Java Keytool y Jarsigner
- Otras integraciones de proveedores externos

Mejore la seguridad de su servidor web con la descarga de SSL/TLS en AWS CloudHSM

Los servidores web y sus clientes (navegadores web) pueden usar los protocolos capa de sockets seguros (SSL) o seguridad de la capa de transporte (TLS) para confirmar la identidad del servidor web y establecer una conexión segura que envíe y reciba páginas web u otros datos a través de Internet. Esto se conoce comúnmente como HTTPS. El servidor web utiliza un par de claves público-privadas y un certificado de clave pública SSL/TLS para establecer una sesión HTTPS con cada cliente. Este proceso implica muchos cálculos para los servidores web, pero puede transferir parte de ellos a su AWS CloudHSM clúster, lo que se conoce como aceleración de SSL. La descarga reduce la carga informática del servidor web y proporciona seguridad adicional al almacenar claves privadas del servidor en un HSM.

En los siguientes temas se proporciona una descripción general del AWS CloudHSM funcionamiento de SSL/TLS offload with y tutoriales para configurar SSL/TLS offload with en las siguientes plataformas. AWS CloudHSM

Para Linux, utilice el motor dinámico de OpenSSL del software de servidor web NGINX o Apache HTTP Server .

Para Windows, utilice el software de servidor web Internet Information Services (IIS) for Windows Server.

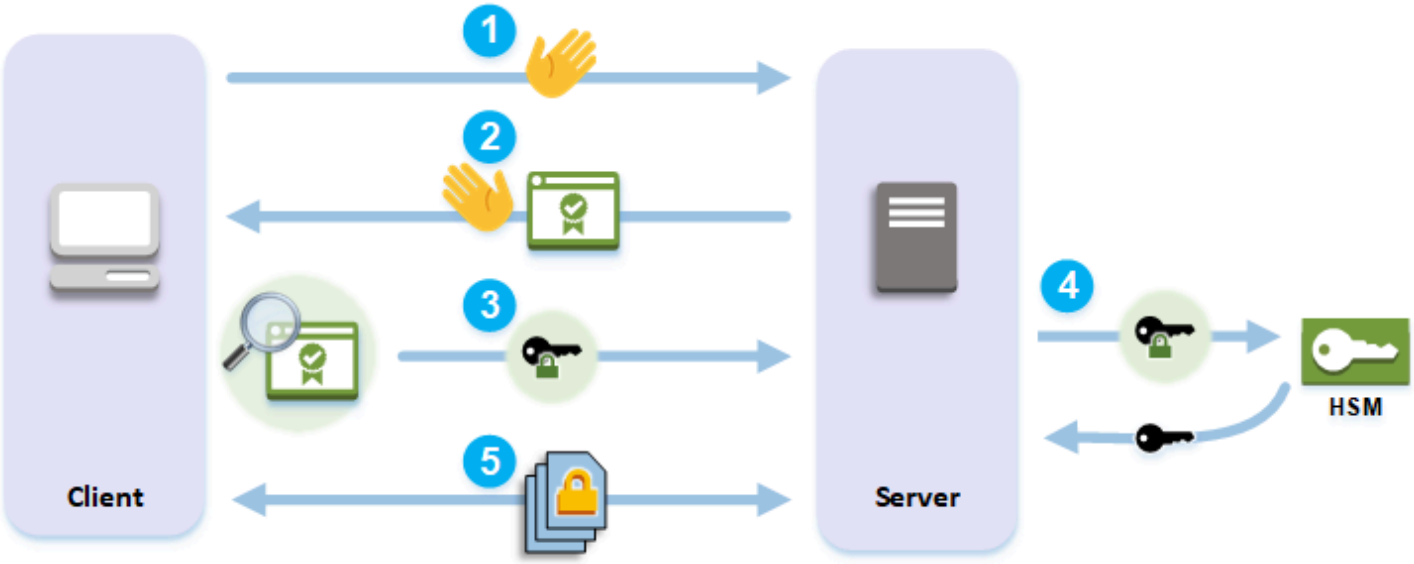
Temas

- Cómo funciona la descarga de SSL/TLS AWS CloudHSM
- Descarga de SSL/TLS en Linux
- Uso de IIS con CNG para la descarga de SSL/TLS en Windows
- Agregar un equilibrador de carga con Elastic Load Balancing (opcional)

Cómo funciona la descarga de SSL/TLS AWS CloudHSM

Para establecer una conexión HTTPS, el servidor web realiza un proceso de protocolo de enlace con los clientes. Como parte de este proceso, el servidor descarga parte del procesamiento criptográfico en los HSM, tal y como se muestra en la siguiente figura. Cada paso del proceso se explica debajo de la figura.

> **Note**
>
> En la imagen y el proceso siguientes se presupone que se utiliza RSA para la verificación del servidor y el intercambio de claves. El proceso es un tanto diferente cuando se utiliza Diffie–Hellman en lugar de RSA.



1. El cliente envía un mensaje de saludo al servidor.
2. El servidor responde con un mensaje de saludo y envía el certificado del servidor.
3. El cliente realiza las siguientes acciones:
   a. Verifica que el certificado del servidor SSL/TLS esté firmado por uno de los certificados raíz en los que confía el cliente.
   b. Extrae la clave pública del certificado del servidor.
   c. Genera un número secreto principal preliminar y lo cifra con la clave pública del servidor.
   d. Envía el número secreto principal preliminar cifrado al servidor.
4. Para descifrar el número secreto principal preliminar del cliente, el servidor lo envía al HSM. El HSM utiliza la clave privada del HSM para descifrar el número secreto principal preliminar y, a continuación, lo envía al servidor. Independientemente, el cliente y el servidor utilizan cada uno el número secreto principal preliminar e información de los mensajes de saludo para calcular un secreto maestro.
5. El proceso de protocolo de enlace finaliza. Durante el resto de la sesión, todos los mensajes enviados entre el cliente y el servidor se cifran con derivados del secreto maestro.

Para obtener información sobre cómo configurar la descarga de SSL/TLS con, consulte uno de los siguientes temas: AWS CloudHSM

- Descarga de SSL/TLS en Linux
- Uso de IIS con CNG para la descarga de SSL/TLS en Windows

Descarga de SSL/TLS en Linux

Con AWS CloudHSM, puede realizar la descarga de SSL/TLS en Linux con NGINX, Apache y Tomcat. Para obtener más información, consulte los siguientes temas relacionados:

Temas

- Uso de NGINX o Apache con OpenSSL para la descarga de SSL/TLS en Linux
- Uso de Tomcat con JSSE para la descarga de SSL/TLS en Linux

Uso de NGINX o Apache con OpenSSL para la descarga de SSL/TLS en Linux

En este tema se proporcionan step-by-step instrucciones para configurar la descarga de SSL/TLS en un servidor web Linux. AWS CloudHSM

Temas

- Información general
- Paso 1: configurar los requisitos previos
- Paso 2: generar o importar una clave privada y un certificado SSL/TLS
- Paso 3: configurar el servidor web
- Paso 4: habilitar el tráfico HTTPS y verificar el certificado

Información general

En Linux, las aplicaciones de servidor web NGINX y Apache HTTP Server se integran con OpenSSL para admitir HTTPS. El motor dinámico de AWS CloudHSM para OpenSSL proporciona una interfaz que permite al software del servidor web utilizar los HSM de un clúster para las descargas criptográficas y el almacenamiento de claves. El motor de OpenSSL es el puente que conecta el servidor web con su clúster de AWS CloudHSM .

Para completar este tutorial, primero debe decidir si va a utilizar el software de servidor web NGINX o Apache en Linux. A continuación, el tutorial le enseña a realizar las tareas siguientes:

- Instalar el software del servidor web en una instancia de Amazon EC2.
- Configurar el software del servidor web para que sea compatible con HTTPS mediante el uso de una clave privada almacenada en su clúster de AWS CloudHSM .

- (Opcional) Uso de Amazon EC2 para crear una segunda instancia de servidor web y Elastic Load Balancing para crear un equilibrador de carga. El uso de un equilibrador de carga puede mejorar el desempeño al distribuir la carga entre varios servidores. También puede proporcionar redundancia y una mayor disponibilidad si uno o más servidores funcionan mal.

Cuando esté listo para empezar, vaya al Paso 1: configurar los requisitos previos.

Paso 1: configurar los requisitos previos

Las diferentes plataformas requieren requisitos previos diferentes. Utilice la siguiente sección de requisitos previos que se ajuste a su plataforma.

Temas

- Requisitos previos para el SDK 5 de cliente
- Requisitos previos para el SDK 3 de cliente

Requisitos previos para el SDK 5 de cliente

Si desea configurar un servidor web para la descarga SSL/TLS con SDK 5 de cliente, necesita lo siguiente:

- Un AWS CloudHSM clúster activo con al menos dos módulos de seguridad de hardware (HSM)

> **Note**
>
> Puede usar un único clúster de HSM, pero primero debe deshabilitar la durabilidad de la clave de cliente. Para obtener más información, consulte Administrar la configuración de durabilidad de las claves de cliente y Herramienta de configuración del SDK 5 de cliente.

- Una instancia de Amazon EC2 que ejecute el sistema operativo Linux y tenga el siguiente software instalado:
  - Un servidor web (NGINX o Apache)
  - El motor dinámico de OpenSSL para el SDK 5 de cliente
- Un usuario de criptografía (CU) que sea el propietario y administre la clave privada del servidor web en el HSM.

Para configurar una instancia de servidor web de Linux y crear un CU en el HSM

1. Instale y configure el motor dinámico OpenSSL para. AWS CloudHSM Para obtener más información sobre la instalación del motor dinámico de OpenSSL, consulte Motor dinámico de OpenSSL para SDK 5 de cliente.
2. En una instancia Linux EC2 que tenga acceso a su clúster, instale el servidor web NGINX o Apache:

   Amazon Linux

   - NGINX

     ```
     $ sudo yum install nginx
     ```

   - Apache

     ```
     $ sudo yum install httpd24 mod24_ssl
     ```

   Amazon Linux 2

   - Para obtener información sobre cómo descargar la última versión de NGINX en Amazon Linux 2, consulte el sitio web de NGINX.

     La última versión de NGINX disponible para Amazon Linux 2 utiliza una versión de OpenSSL más reciente que la versión de sistema de OpenSSL. Después de instalar NGINX, debe crear un enlace simbólico desde la biblioteca AWS CloudHSM OpenSSL Dynamic Engine a la ubicación que espera esta versión de OpenSSL

     ```
     $ sudo ln -sf /opt/cloudhsm/lib/libcloudhsm_openssl_engine.so /usr/lib64/
     engines-1.1/cloudhsm.so
     ```

   - Apache

     ```
     $ sudo yum install httpd mod_ssl
     ```

CentOS 7

- Para obtener información sobre cómo descargar la última versión de NGINX en Centos 7, consulte el sitio web de NGINX.

  La última versión de NGINX disponible para CentOS 7 utiliza una versión de OpenSSL más reciente que la versión de sistema de OpenSSL. Después de instalar NGINX, debe crear un enlace simbólico desde la biblioteca AWS CloudHSM OpenSSL Dynamic Engine a la ubicación que espera esta versión de OpenSSL

  ```
  $ sudo ln -sf /opt/cloudhsm/lib/libcloudhsm_openssl_engine.so /usr/lib64/
  engines-1.1/cloudhsm.so
  ```

- Apache

  ```
  $ sudo yum install httpd mod_ssl
  ```

Red Hat 7

- Para obtener información sobre cómo descargar la última versión de NGINX en Red Hat 7, consulte el sitio web de NGINX.

  La última versión de NGINX disponible para Red Hat 7 utiliza una versión de OpenSSL más reciente que la versión de sistema de OpenSSL. Después de instalar NGINX, debe crear un enlace simbólico desde la biblioteca AWS CloudHSM OpenSSL Dynamic Engine a la ubicación que espera esta versión de OpenSSL

  ```
  $ sudo ln -sf /opt/cloudhsm/lib/libcloudhsm_openssl_engine.so /usr/lib64/
  engines-1.1/cloudhsm.so
  ```

- Apache

  ```
  $ sudo yum install httpd mod_ssl
  ```

CentOS 8

- NGINX

```
$ sudo yum install nginx
```

- Apache

```
$ sudo yum install httpd mod_ssl
```

Red Hat 8

- NGINX

```
$ sudo yum install nginx
```

- Apache

```
$ sudo yum install httpd mod_ssl
```

Ubuntu 18.04

- NGINX

```
$ sudo apt install nginx
```

- Apache

```
$ sudo apt install apache2
```

Ubuntu 20.04

- NGINX

```
$ sudo apt install nginx
```

- Apache

```
$ sudo apt install apache2
```

Ubuntu 22.04

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

3. Utilice la CLI de CloudHSM para crear una CU. Para obtener más información sobre la administración de los usuarios de HSM, consulte Administrar a los usuarios de HSM con la CLI de CloudHSM.

> **Tip**
>
> Realice un seguimiento del nombre de usuario y la contraseña del CU. Los necesitará más adelante cuando genere o importe el certificado y la clave privada de HTTPS para el servidor web.

Después de completar estos pasos, vaya a Paso 2: generar o importar una clave privada y un certificado SSL/TLS.

Notas

- Para usar Linux con seguridad mejorada (SELinux) y servidores web, debe permitir las conexiones TCP salientes en el puerto 2223, que es el puerto que el SDK 5 de cliente utiliza para comunicarse con el HSM.
- Para crear y activar un clúster y permitir que una instancia EC2 acceda al clúster, complete los pasos que se indican en Introducción a AWS CloudHSM. La introducción ofrece step-by-step instrucciones para crear un clúster activo con un HSM y una instancia de cliente Amazon EC2. Puede utilizar esta instancia de cliente como su servidor web.
- Para evitar deshabilitar la durabilidad de la clave de cliente, agregue más de un HSM a su clúster. Para obtener más información, consulte Agregar un HSM.
- Para conectarse a su instancia de cliente, puede utilizar SSH o PuTTY. Para obtener más información, consulte Conectarse a la instancia de Linux mediante SSH o Conectarse a la instancia de Linux desde Windows mediante PuTTY en la documentación de Amazon EC2.

Requisitos previos para el SDK 3 de cliente

Si desea configurar un servidor web para la descarga SSL/TLS con SDK 3 de cliente, necesita lo siguiente:

- Un AWS CloudHSM clúster activo con al menos un HSM.
- Una instancia de Amazon EC2 que ejecute el sistema operativo Linux y tenga el siguiente software instalado:
  - El AWS CloudHSM cliente y las herramientas de línea de comandos.
  - La aplicación del servidor web NGINX o Apache.
  - El motor AWS CloudHSM dinámico de OpenSSL.
- Un usuario de criptografía (CU) que sea el propietario y administre la clave privada del servidor web en el HSM.

Para configurar una instancia de servidor web de Linux y crear un CU en el HSM

1. Realice los pasos que se indican en Introducción. Así, dispondrá de un clúster activo con un HSM y una instancia de cliente de Amazon EC2. La instancia EC2 se configurará con las herramientas de línea de comandos. Utilice esta instancia de cliente como su servidor web.
2. Conéctese a su instancia de cliente. Para obtener más información, consulte Conectarse a la instancia de Linux mediante SSH o Conectarse a la instancia de Linux desde Windows mediante PuTTY en la documentación de Amazon EC2.
3. En una instancia Linux EC2 que tenga acceso a su clúster, instale el servidor web NGINX o Apache:

   Amazon Linux

   - NGINX

     ```
     $ sudo yum install nginx
     ```

   - Apache

     ```
     $ sudo yum install httpd24 mod24_ssl
     ```

   Amazon Linux 2

   - La versión 1.19 de NGINX es la última versión de NGINX compatible con el motor del SDK 3 de cliente de Amazon Linux 2.

Para obtener más información y descargar la versión 1.19 de NGINX, consulte el sitio web de NGINX.

- Apache

```
$ sudo yum install httpd mod_ssl
```

CentOS 7

- La versión 1.19 de NGINX es la última versión de NGINX compatible con el motor del SDK 3 de cliente de Centos 7.

  Para obtener más información y descargar la versión 1.19 de NGINX, consulte el sitio web de NGINX.

- Apache

```
$ sudo yum install httpd mod_ssl
```

Red Hat 7

- La versión 1.19 de NGINX es la última versión de NGINX compatible con el motor del SDK 3 de cliente de Red Hat 7.

  Para obtener más información y descargar la versión 1.19 de NGINX, consulte el sitio web de NGINX.

- Apache

```
$ sudo yum install httpd mod_ssl
```

Ubuntu 16.04

- NGINX

```
$ sudo apt install nginx
```

- Apache

```
$ sudo apt install apache2
```

Ubuntu 18.04

- NGINX

```
$ sudo apt install nginx
```

- Apache

```
$ sudo apt install apache2
```

4. (Opcional) Añada más HSM a su clúster. Para obtener más información, consulte Agregar un HSM.
5. Utilice cloudhsm_mgmt_util para crear un CU. Para obtener más información, consulte Administración de usuarios de HSM. Realice un seguimiento del nombre de usuario y la contraseña del CU. Los necesitará más adelante cuando genere o importe el certificado y la clave privada de HTTPS para el servidor web.

Después de completar estos pasos, vaya a Paso 2: generar o importar una clave privada y un certificado SSL/TLS.

Paso 2: generar o importar una clave privada y un certificado SSL/TLS

Para habilitar HTTPS, la aplicación del servidor web (NGINX o Apache) necesita una clave privada y un certificado SSL/TLS correspondiente. Para utilizar la descarga de SSL/TLS de un servidor web AWS CloudHSM, debe almacenar la clave privada en un HSM de su clúster. AWS CloudHSM Puede realizar esta operación de una de las siguientes formas:

- Si todavía no dispone de una clave privada y un certificado correspondiente, genere una clave privada en un HSM. Use la clave privada para crear una solicitud de firma de certificado (CSR), la cual utilizará para crear un certificado SSL/TLS.
- Si ya dispone de una clave privada y de su certificado correspondiente, puede importar la clave privada a un HSM.

Independientemente del método anterior que elija, se exporta una clave privada PEM falsa del HSM, que es un archivo de clave privada en formato PEM que contiene una referencia a la clave privada almacenada en el HSM (no es la clave privada real). El servidor web utiliza el archivo de clave privada PEM falso para identificar la clave privada en el HSM durante la descarga de SSL/TLS.

Realice una de las siguientes acciones siguientes:

- Generación de una clave privada y un certificado
- Importación de una clave privada y un certificado existentes

Generación de una clave privada y un certificado

Generación de una clave privada

En esta sección, se muestra cómo generar un par de claves mediante el uso de la herramienta Utilidad de administración de claves (KMU) del SDK 3 de cliente. Una vez que haya generado un par de claves dentro del HSM, puede exportarlo como un archivo PEM falso y generar el certificado correspondiente.

Las claves privadas generadas con la utilidad de administración de claves (KMU) se pueden usar tanto con el SSK 3 de cliente como con el SDK 5 de cliente.

Instalación y configuración de la herramienta Utilidad de administración de claves (KMU)

1. Conéctese a su instancia de cliente.
2. Instalación y configuración de SDK 3 de cliente.
3. Ejecute el siguiente comando para iniciar el cliente. AWS CloudHSM

   Amazon Linux

   ```
   $ sudo start cloudhsm-client
   ```

   Amazon Linux 2

   ```
   $ sudo service cloudhsm-client start
   ```

   CentOS 7

   ```
   $ sudo service cloudhsm-client start
   ```

CentOS 8

```
$ sudo service cloudhsm-client start
```

RHEL 7

```
$ sudo service cloudhsm-client start
```

RHEL 8

```
$ sudo service cloudhsm-client start
```

Ubuntu 16.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 20.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

4. Ejecute el siguiente comando para iniciar la herramienta de línea de comandos key_mgmt_util.

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

5. Ejecute el siguiente comando para iniciar sesión en el HSM. Reemplace *<nombre de usuario>* y *<password>* por el nombre de usuario y la contraseña del usuario criptográfico (CU).

```
Command: loginHSM -u CU -s <user name> -p <password>>
```

Generación una clave privada

Según su caso de uso, puede generar un par de claves RSA o EC. Realice una de las siguientes acciones siguientes:

- Cómo generar una clave privada RSA en un HSM

  Utilice el comando `genRSAKeyPair` para generar un par de claves RSA. Este ejemplo genera un par de claves RSA con un módulo de 2048, un exponente público de 65537 y una etiqueta de *`tls_rsa_keypair.`*

  ```
  Command: genRSAKeyPair -m 2048 -e 65537 -l tls_rsa_keypair
  ```

  Si el comando se ejecutó correctamente, debería ver el siguiente resultado que indica que ha generado correctamente un par de claves RSA.

  ```
  Cfm3GenerateKeyPair returned: 0x00 : HSM Return: SUCCESS

                  Cfm3GenerateKeyPair:    public key handle: 7    private key handle: 8

                  Cluster Status:
                  Node id 1 status: 0x00000000 : HSM Return: SUCCESS
  ```

- Cómo generar una clave privada EC en un HSM

  Utilice el comando `genECCKeyPair` para generar un par de claves EC. Este ejemplo genera un par de claves EC con un ID de curva de 2 (correspondiente a la curva `NID_X9_62_prime256v1`) y una etiqueta de *`tls_ec_keypair`*.

  ```
  Command: genECCKeyPair -i 2 -l tls_ec_keypair
  ```

  Si el comando se ejecutó correctamente, debería ver el siguiente resultado que indica que ha generado correctamente un par de claves EC.

  ```
  Cfm3GenerateKeyPair returned: 0x00 : HSM Return: SUCCESS

                  Cfm3GenerateKeyPair:    public key handle: 7    private key handle: 8

                  Cluster Status:
                  Node id 1 status: 0x00000000 : HSM Return: SUCCESS
  ```

Exportación de un archivo de clave privada PEM falso

Una vez que tenga una clave privada en el HSM, debe exportar un archivo de clave privada PEM falso. Este archivo no contiene los datos de clave reales, pero permite que el motor dinámico de OpenSSL identifique la clave privada en el HSM. A continuación, puede utilizar la clave privada para crear una solicitud de firma de certificado (CSR) y firmar la CSR para crear el certificado.

> **Note**
>
> Los archivos PEM falsos generados con la herramienta Utilidad de administración de claves (KMU) se pueden usar tanto con el SDK 3 de cliente como con el SDK 5 de cliente.

Identifique el identificador de claves que corresponde a la clave que desea exportar como un PEM falso y, a continuación, ejecute el siguiente comando para exportar la clave privada en un formato PEM falso y guárdela en un archivo. Reemplace los valores siguientes por sus propios valores.

- *<private_key_handle>*: identificador de la clave privada generada. Este identificador se generó por uno de los comandos de generación de claves del paso anterior. En el ejemplo anterior, el identificador de la clave privada es 8.
- *<web_server_fake_PEM.key>*: nombre del archivo en el que se escribirá la clave PEM falsa.

```
Command: getCaviumPrivKey -k <private_key_handle> -out <web_server_fake_PEM.key>
```

Exit (Salir)

Ejecute el comando siguiente para detener el key_mgmt_util.

```
Command: exit
```

Ahora debería tener un nuevo archivo en su sistema, ubicado en la ruta especificada*<web_server_fake_PEM.key>* del comando anterior. Este archivo es el archivo de clave privada PEM falso.

Generación de un certificado autofirmado

Una vez que haya generado una clave privada PEM falsa, puede usar este archivo para generar una solicitud de firma de certificado (CSR) y un certificado.

En un entorno de producción, normalmente se usa una entidad de certificación (CA) para crear un certificado de una CSR. No es necesaria una CA para un entorno de prueba. Si utiliza una CA, envíele el archivo CSR y utilice el certificado SSL/TLS firmado que le proporcione en su servidor web para HTTPS.

Como alternativa al uso de una CA, puede usar el motor dinámico AWS CloudHSM OpenSSL para crear un certificado autofirmado. Los navegadores no confían en certificados autofirmados y no deben utilizarse en entornos de producción. Se pueden usar en entornos de prueba.

> **Warning**
>
> Los certificados autofirmados deben utilizarse únicamente en entornos de prueba. En entornos de producción, utilice un método más seguro como, por ejemplo, una autoridad de certificación para crear un certificado.

Instalación y configuración del motor dinámico de OpenSSL

1. Conéctese a su instancia de cliente.
2. Para instalar y configurar, realice una de las siguientes acciones:
   - the section called “Instalación del motor dinámico de OpenSSL”
   - the section called “Motor dinámico de OpenSSL”

Generación de un certificado

1. Obtener una copia del archivo PEM falso generado en un paso anterior.
2. Creación de una CSR

   Ejecute el siguiente comando para usar el motor dinámico AWS CloudHSM OpenSSL para crear una solicitud de firma de certificado (CSR). Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa. Reemplace *<web_server.csr>* por el nombre del archivo que contiene la CSR.

   El comando `req` es interactivo. Responderá a cada campo. La información del campo se copia en su certificado SSL/TLS.

   ```
   $ openssl req -engine cloudhsm -new -key <web_server_fake_PEM.key> -
   out <web_server.csr>
   ```

3. Creación de un certificado autofirmado

   Ejecute el siguiente comando para usar el motor dinámico de AWS CloudHSM OpenSSL para firmar su CSR con su clave privada en su HSM. Esto creará un certificado autofirmado. Reemplace los siguientes valores en el comando por sus propios valores.

   - *<web_server.csr>*: nombre del archivo que contiene la CSR.
   - *<web_server_fake_PEM.key>*: nombre del archivo que contiene la clave privada PEM falsa.
   - *<web_server.crt>*: nombre del archivo que contendrá su certificado de servidor web.

   ```
   $ openssl x509 -engine cloudhsm -req -days 365 -in <web_server.csr> -
   signkey <web_server_fake_PEM.key> -out <web_server.crt>
   ```

Después de completar estos pasos, vaya a Paso 3: configurar el servidor web.

Importación de una clave privada y un certificado existentes

Es posible que ya disponga de una clave privada y un certificado SSL/TLS correspondiente que utiliza para HTTPS en su servidor web. En caso afirmativo, puede importar la clave a un HSM siguiendo los pasos indicados en esta sección.

> **Note**
>
> Algunas notas sobre la importación de claves privadas y la compatibilidad con el SDK de cliente:
>
> - La importación de una clave privada existente requiere el SDK 3 de cliente.
> - Puede usar claves privadas del SDK 3 de cliente con el SDK 5 de cliente.
> - El motor dinámico de Open SSL para el SDK 3 de cliente no es compatible con las plataformas Linux más recientes, pero sí lo es la implementación del motor dinámico de Open SSL para el SDK 5 de cliente. Puede importar una clave privada existente mediante la herramienta Utilidad de administración de claves (KMU) incluida con el SDK 3 de cliente y, a continuación, usar esa clave privada y la implementación del motor dinámico de OpenSSL con el SDK 5 de cliente para admitir la descarga de SSL/TLS en las plataformas Linux más recientes.

Cómo importar una clave privada existente a un HSM con SDK 3 de cliente

1. Conéctese a su instancia de cliente de Amazon EC2. Si es necesario, copie su clave privada y el certificado en la instancia.
2. Instalación and configuración de SDK 3 de cliente
3. Ejecute el siguiente comando para iniciar el cliente. AWS CloudHSM

   Amazon Linux

   ```
   $ sudo start cloudhsm-client
   ```

   Amazon Linux 2

   ```
   $ sudo service cloudhsm-client start
   ```

   CentOS 7

   ```
   $ sudo service cloudhsm-client start
   ```

   CentOS 8

   ```
   $ sudo service cloudhsm-client start
   ```

   RHEL 7

   ```
   $ sudo service cloudhsm-client start
   ```

   RHEL 8

   ```
   $ sudo service cloudhsm-client start
   ```

   Ubuntu 16.04 LTS

   ```
   $ sudo service cloudhsm-client start
   ```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 20.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

4. Ejecute el siguiente comando para iniciar la herramienta de línea de comandos key_mgmt_util.

```
$ /opt/cloudhsm/bin/key_mgmt_util
```

5. Ejecute el siguiente comando para iniciar sesión en el HSM. Reemplace *<nombre de usuario>* y *<password>* por el nombre de usuario y la contraseña del usuario criptográfico (CU).

```
Command: loginHSM -u CU -s <user name> -p <password>
```

6. Ejecute los siguientes comandos para importar la clave privada a un HSM.

   a. Ejecute el siguiente comando para crear una clave de encapsulación simétrica que sea válida únicamente para la sesión actual. El comando y el resultado se muestran aquí.

```
Command: genSymKey -t 31 -s 16 -sess -l wrapping_key_for_import

Cfm3GenerateSymmetricKey returned: 0x00 : HSM Return: SUCCESS
Symmetric Key Created.  Key Handle: 6
Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

   b. Ejecute el siguiente comando para importar la clave privada actual a un HSM. El comando y el resultado se muestran aquí. Reemplace los valores siguientes por sus propios valores:

   - *<web_server_existing.key>*: nombre del archivo que contiene su clave privada.
   - *<web_server_imported_key>*: etiqueta para su clave privada importada.

- *<wrapping_key_handle>*: identificador de clave de encapsulamiento generado en el comando anterior. En el ejemplo anterior el identificador de clave de encapsulación es 6.

```
Command: importPrivateKey -f <web_server_existing.key> -
l <web_server_imported_key> -w <wrapping_key_handle>

BER encoded key length is 1219
Cfm3WrapHostKey returned: 0x00 : HSM Return: SUCCESS
Cfm3CreateUnwrapTemplate returned: 0x00 : HSM Return: SUCCESS
Cfm3UnWrapKey returned: 0x00 : HSM Return: SUCCESS
Private Key Unwrapped.  Key Handle: 8
Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
```

7. Ejecute el comando siguiente para exportar la clave privada en formato PEM falso y guardarla en un archivo. Reemplace los valores siguientes por sus propios valores.

   - *<private_key_handle>*: identificador de la clave privada importada. Este identificador se generó con el segundo comando del paso anterior. En el ejemplo anterior, el identificador de la clave privada es 8.
   - *<web_server_fake_PEM.key>*: nombre del archivo que contiene su clave privada PEM falsa exportada.

   ```
   Command: getCaviumPrivKey -k <private_key_handle> -out <web_server_fake_PEM.key>
   ```

8. Ejecute el comando siguiente para detener key_mgmt_util.

   ```
   Command: exit
   ```

Después de completar estos pasos, vaya a Paso 3: configurar el servidor web.

Paso 3: configurar el servidor web

Actualice la configuración de software del servidor web para utilizar el certificado HTTPS y la correspondiente clave privada PEM falsa que ha creado en el paso anterior. Recuerde hacer una copia de seguridad de sus certificados y claves existentes antes de empezar. De este modo, concluirá la configuración del software del servidor web de Linux para la descarga SSL/TLS con AWS CloudHSM.

Complete los pasos indicados en una de las siguientes secciones.

Temas

- Configuración del servidor web NGINX
- Configuración del servidor web Apache

Configuración del servidor web NGINX

Use esta sección para configurar NGINX en las plataformas compatibles.

Para actualizar la configuración del servidor web para NGINX

1. Conéctese a su instancia de cliente.
2. Ejecute el siguiente comando para crear los directorios necesarios para el certificado del servidor web y la clave privada PEM falsa.

```
$ sudo mkdir -p /etc/pki/nginx/private
```

3. Ejecute el siguiente comando para copiar su certificado de servidor web en la ubicación necesaria. Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

```
$ sudo cp <web_server.crt> /etc/pki/nginx/server.crt
```

4. Ejecute el siguiente comando para copiar la clave privada PEM falsa en la ubicación necesaria. Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

```
$ sudo cp <web_server_fake_PEM.key> /etc/pki/nginx/private/server.key
```

5. Ejecute el siguiente comando para cambiar la propiedad de estos archivos para que el usuario denominado nginx pueda leerlos.

```
$ sudo chown nginx /etc/pki/nginx/server.crt /etc/pki/nginx/private/server.key
```

6. Ejecute el siguiente comando para hacer una copia de seguridad del archivo `/etc/nginx/nginx.conf`.

```
$ sudo cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.backup
```

7. Actualizar la configuración de NGINX

> **Note**
>
> Cada clúster puede soportar un máximo de 1000 procesos de trabajo de NGINX en todos los servidores web de NGINX.

Amazon Linux

Utilice un editor de texto para editar el archivo `/etc/nginx/nginx.conf`. Esto requiere permisos de raíz de Linux. En la parte superior del archivo, añada la siguiente línea:

- Si usa SDK 3 de cliente

```
ssl_engine cloudhsm;
env n3fips_password;
```

- Si usa SDK 5 de cliente

```
ssl_engine cloudhsm;
env CLOUDHSM_PIN;
```

Agregue lo siguiente a la sección TLS del archivo:

```
# Settings for a TLS enabled server.
server {
    listen       443 ssl http2 default_server;
    listen       [::]:443 ssl http2 default_server;
    server_name  _;
    root         /usr/share/nginx/html;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    # It is *strongly* recommended to generate unique DH parameters
    # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem 2048
    #ssl_dhparam "/etc/pki/nginx/dhparams.pem";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_protocols TLSv1.2;
```

```
    ssl_ciphers "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA";
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
    }

    error_page 404 /404.html;
    location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

Amazon Linux 2

Utilice un editor de texto para editar el archivo `/etc/nginx/nginx.conf`. Esto requiere permisos de raíz de Linux. En la parte superior del archivo, añada la siguiente línea:

- Si usa SDK 3 de cliente

```
ssl_engine cloudhsm;
env n3fips_password;
```

- Si usa SDK 5 de cliente

```
ssl_engine cloudhsm;
env CLOUDHSM_PIN;
```

Agregue lo siguiente a la sección TLS del archivo:

```
# Settings for a TLS enabled server.
server {
    listen       443 ssl http2 default_server;
    listen       [::]:443 ssl http2 default_server;
    server_name  _;
    root         /usr/share/nginx/html;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    # It is *strongly* recommended to generate unique DH parameters
    # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem 2048
    #ssl_dhparam "/etc/pki/nginx/dhparams.pem";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_protocols TLSv1.2;
    ssl_ciphers "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA";
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
    }

    error_page 404 /404.html;
    location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

CentOS 7

Utilice un editor de texto para editar el archivo `/etc/nginx/nginx.conf`. Esto requiere permisos de raíz de Linux. En la parte superior del archivo, añada la siguiente línea:

- Si usa SDK 3 de cliente

```
ssl_engine cloudhsm;
env n3fips_password;
```

- Si usa SDK 5 de cliente

```
ssl_engine cloudhsm;
env CLOUDHSM_PIN;
```

Agregue lo siguiente a la sección TLS del archivo:

```
# Settings for a TLS enabled server.
server {
    listen       443 ssl http2 default_server;
    listen       [::]:443 ssl http2 default_server;
    server_name  _;
    root         /usr/share/nginx/html;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    # It is *strongly* recommended to generate unique DH parameters
    # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem 2048
    #ssl_dhparam "/etc/pki/nginx/dhparams.pem";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_protocols TLSv1.2;
    ssl_ciphers "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA";
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
```

```
    include /etc/nginx/default.d/*.conf;

    location / {
    }

    error_page 404 /404.html;
    location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

CentOS 8

Utilice un editor de texto para editar el archivo `/etc/nginx/nginx.conf`. Esto requiere permisos de raíz de Linux. En la parte superior del archivo, añada la siguiente línea:

```
ssl_engine cloudhsm;
env CLOUDHSM_PIN;
```

Agregue lo siguiente a la sección TLS del archivo:

```
# Settings for a TLS enabled server.
server {
    listen       443 ssl http2 default_server;
    listen       [::]:443 ssl http2 default_server;
    server_name  _;
    root         /usr/share/nginx/html;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    # It is *strongly* recommended to generate unique DH parameters
    # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem 2048
    #ssl_dhparam "/etc/pki/nginx/dhparams.pem";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
```

```
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA";
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
    }

    error_page 404 /404.html;
    location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

Red Hat 7

Utilice un editor de texto para editar el archivo `/etc/nginx/nginx.conf`. Esto requiere permisos de raíz de Linux. En la parte superior del archivo, añada la siguiente línea:

- Si usa SDK 3 de cliente

```
ssl_engine cloudhsm;
env n3fips_password;
```

- Si usa SDK 5 de cliente

```
ssl_engine cloudhsm;
env CLOUDHSM_PIN;
```

Agregue lo siguiente a la sección TLS del archivo:

```
# Settings for a TLS enabled server.
server {
```

```
    listen       443 ssl http2 default_server;
    listen       [::]:443 ssl http2 default_server;
    server_name  _;
    root         /usr/share/nginx/html;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    # It is *strongly* recommended to generate unique DH parameters
    # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem 2048
    #ssl_dhparam "/etc/pki/nginx/dhparams.pem";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_protocols TLSv1.2;
    ssl_ciphers "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA";
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
    }

    error_page 404 /404.html;
    location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

Red Hat 8

Utilice un editor de texto para editar el archivo `/etc/nginx/nginx.conf`. Esto requiere permisos de raíz de Linux. En la parte superior del archivo, añada la siguiente línea:

```
ssl_engine cloudhsm;
```

```
env CLOUDHSM_PIN;
```

Agregue lo siguiente a la sección TLS del archivo:

```
# Settings for a TLS enabled server.
server {
    listen       443 ssl http2 default_server;
    listen       [::]:443 ssl http2 default_server;
    server_name  _;
    root         /usr/share/nginx/html;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    # It is *strongly* recommended to generate unique DH parameters
    # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem 2048
    #ssl_dhparam "/etc/pki/nginx/dhparams.pem";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA";
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
    }

    error_page 404 /404.html;
    location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

Ubuntu 16.04 LTS

Utilice un editor de texto para editar el archivo `/etc/nginx/nginx.conf`. Esto requiere permisos de raíz de Linux. En la parte superior del archivo, añada la siguiente línea:

```
ssl_engine cloudhsm;
    env n3fips_password;
```

Agregue lo siguiente a la sección TLS del archivo:

```
# Settings for a TLS enabled server.
    server {
        listen       443 ssl http2 default_server;
        listen       [::]:443 ssl http2 default_server;
        server_name  _;
        root         /usr/share/nginx/html;

        ssl_certificate "/etc/pki/nginx/server.crt";
        ssl_certificate_key "/etc/pki/nginx/private/server.key";
        # It is *strongly* recommended to generate unique DH parameters
        # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem
 2048
        #ssl_dhparam "/etc/pki/nginx/dhparams.pem";
        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout  10m;
        ssl_protocols TLSv1.2;
        ssl_ciphers "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-
SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-
SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-
RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-
GCM-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA";
        ssl_prefer_server_ciphers on;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        location / {
        }

        error_page 404 /404.html;
        location = /40x.html {
```

```
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }
```

Ubuntu 18.04 LTS

Utilice un editor de texto para editar el archivo `/etc/nginx/nginx.conf`. Esto requiere permisos de raíz de Linux. En la parte superior del archivo, añada la siguiente línea:

```
ssl_engine cloudhsm;
    env CLOUDHSM_PIN;
```

Agregue lo siguiente a la sección TLS del archivo:

```
# Settings for a TLS enabled server.
    server {
        listen       443 ssl http2 default_server;
        listen       [::]:443 ssl http2 default_server;
        server_name  _;
        root         /usr/share/nginx/html;

        ssl_certificate "/etc/pki/nginx/server.crt";
        ssl_certificate_key "/etc/pki/nginx/private/server.key";
        # It is *strongly* recommended to generate unique DH parameters
        # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem
 2048
        #ssl_dhparam "/etc/pki/nginx/dhparams.pem";
        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout  10m;
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_ciphers "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-
SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-
SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-
RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-
GCM-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA";
        ssl_prefer_server_ciphers on;
```

```
        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        location / {
        }

        error_page 404 /404.html;
        location = /40x.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }
```

Ubuntu 20.04 LTS

Utilice un editor de texto para editar el archivo `/etc/nginx/nginx.conf`. Esto requiere permisos de raíz de Linux. En la parte superior del archivo, añada la siguiente línea:

```
ssl_engine cloudhsm;
    env CLOUDHSM_PIN;
```

Agregue lo siguiente a la sección TLS del archivo:

```
# Settings for a TLS enabled server.
    server {
        listen       443 ssl http2 default_server;
        listen       [::]:443 ssl http2 default_server;
        server_name  _;
        root         /usr/share/nginx/html;

        ssl_certificate "/etc/pki/nginx/server.crt";
        ssl_certificate_key "/etc/pki/nginx/private/server.key";
        # It is *strongly* recommended to generate unique DH parameters
        # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem
 2048
        #ssl_dhparam "/etc/pki/nginx/dhparams.pem";
        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout  10m;
        ssl_protocols TLSv1.2 TLSv1.3;
```

```
        ssl_ciphers "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-
SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-
SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-
RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-
GCM-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA";
        ssl_prefer_server_ciphers on;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        location / {
        }

        error_page 404 /404.html;
        location = /40x.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

Guarde el archivo.

8. Haga una copia de seguridad del archivo de configuración `systemd` y, a continuación, establezca la ruta de `EnvironmentFile`.

Amazon Linux

No hay que hacer nada.

Amazon Linux 2

1. Haga una copia de seguridad del archivo `nginx.service`.

```
$ sudo cp /lib/systemd/system/nginx.service /lib/systemd/system/
nginx.service.backup
```

2. Abra el archivo `/lib/systemd/system/nginx.service` en un editor de texto y, a continuación, en la sección [Service], añada la siguiente ruta:

```
EnvironmentFile=/etc/sysconfig/nginx
```

CentOS 7

No hay que hacer nada.

CentOS 8

1. Haga una copia de seguridad del archivo `nginx.service`.

```
$ sudo cp /lib/systemd/system/nginx.service /lib/systemd/system/
nginx.service.backup
```

2. Abra el archivo `/lib/systemd/system/nginx.service` en un editor de texto y, a continuación, en la sección [Service], añada la siguiente ruta:

```
EnvironmentFile=/etc/sysconfig/nginx
```

Red Hat 7

No hay que hacer nada.

Red Hat 8

1. Haga una copia de seguridad del archivo `nginx.service`.

```
$ sudo cp /lib/systemd/system/nginx.service /lib/systemd/system/
nginx.service.backup
```

2. Abra el archivo `/lib/systemd/system/nginx.service` en un editor de texto y, a continuación, en la sección [Service], añada la siguiente ruta:

```
EnvironmentFile=/etc/sysconfig/nginx
```

Ubuntu 16.04

1. Haga una copia de seguridad del archivo `nginx.service`.

```
$ sudo cp /lib/systemd/system/nginx.service /lib/systemd/system/
nginx.service.backup
```

2. Abra el archivo `/lib/systemd/system/nginx.service` en un editor de texto y, a continuación, en la sección [Service], añada la siguiente ruta:

```
EnvironmentFile=/etc/sysconfig/nginx
```

Ubuntu 18.04

1. Haga una copia de seguridad del archivo `nginx.service`.

```
$ sudo cp /lib/systemd/system/nginx.service /lib/systemd/system/
nginx.service.backup
```

2. Abra el archivo `/lib/systemd/system/nginx.service` en un editor de texto y, a continuación, en la sección [Service], añada la siguiente ruta:

```
EnvironmentFile=/etc/sysconfig/nginx
```

Ubuntu 20.04 LTS

1. Haga una copia de seguridad del archivo `nginx.service`.

```
$ sudo cp /lib/systemd/system/nginx.service /lib/systemd/system/
nginx.service.backup
```

2. Abra el archivo `/lib/systemd/system/nginx.service` en un editor de texto y, a continuación, en la sección [Service], añada la siguiente ruta:

```
EnvironmentFile=/etc/sysconfig/nginx
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

9. Compruebe si existe el archivo `/etc/sysconfig/nginx` y, a continuación, realice una de las operaciones siguientes:

   - Si el archivo existe, haga una copia de seguridad del mismo ejecutando el siguiente comando:

     ```
     $ sudo cp /etc/sysconfig/nginx /etc/sysconfig/nginx.backup
     ```

   - Si el archivo no existe, abra un editor de texto y, a continuación, cree un archivo denominado `nginx` en la carpeta `/etc/sysconfig/`.

10. Configure el entorno NGINX.

> **Note**
>
> El SDK 5 de cliente introduce la variable de entorno `CLOUDHSM_PIN` para almacenar las credenciales del CU.

Amazon Linux

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

- Si usa SDK 3 de cliente

  ```
  n3fips_password=<CU user name>:<password>
  ```

- Si usa SDK 5 de cliente

  ```
  CLOUDHSM_PIN=<CU user name>:<password>
  ```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Guarde el archivo.

Amazon Linux 2

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

- Si usa SDK 3 de cliente

```
n3fips_password=<CU user name>:<password>
```

- Si usa SDK 5 de cliente

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Guarde el archivo.

CentOS 7

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

- Si usa SDK 3 de cliente

```
n3fips_password=<CU user name>:<password>
```

- Si usa SDK 5 de cliente

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Guarde el archivo.

CentOS 8

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Guarde el archivo.

Red Hat 7

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

- Si usa SDK 3 de cliente

```
n3fips_password=<CU user name>:<password>
```

- Si usa SDK 5 de cliente

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Guarde el archivo.

Red Hat 8

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Guarde el archivo.

Ubuntu 16.04 LTS

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

```
n3fips_password=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Guarde el archivo.

Ubuntu 18.04 LTS

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Guarde el archivo.

Ubuntu 20.04 LTS

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Guarde el archivo.

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

11. Inicie el servidor web NGINX.

Amazon Linux

Abra el archivo `/etc/sysconfig/nginx` en un editor de texto. Esto requiere permisos de raíz de Linux. Añada las credenciales del usuario de criptografía (CU):

```
$ sudo service nginx start
```

Amazon Linux 2

Detención de cualquier proceso de NGINX en ejecución

```
$ sudo systemctl stop nginx
```

Recarga de la configuración `systemd` para incluir los últimos cambios

```
$ sudo systemctl daemon-reload
```

Inicio del proceso de NGINX

```
$ sudo systemctl start nginx
```

CentOS 7

Detención de cualquier proceso de NGINX en ejecución

```
$ sudo systemctl stop nginx
```

Recarga de la configuración `systemd` para incluir los últimos cambios

```
$ sudo systemctl daemon-reload
```

Inicio del proceso de NGINX

```
$ sudo systemctl start nginx
```

CentOS 8

Detención de cualquier proceso de NGINX en ejecución

```
$ sudo systemctl stop nginx
```

Recarga de la configuración `systemd` para incluir los últimos cambios

```
$ sudo systemctl daemon-reload
```

Inicio del proceso de NGINX

```
$ sudo systemctl start nginx
```

Red Hat 7

Detención de cualquier proceso de NGINX en ejecución

```
$ sudo systemctl stop nginx
```

Recarga de la configuración `systemd` para incluir los últimos cambios

```
$ sudo systemctl daemon-reload
```

Inicio del proceso de NGINX

```
$ sudo systemctl start nginx
```

Red Hat 8

Detención de cualquier proceso de NGINX en ejecución

```
$ sudo systemctl stop nginx
```

Recarga de la configuración `systemd` para incluir los últimos cambios

```
$ sudo systemctl daemon-reload
```

Inicio del proceso de NGINX

```
$ sudo systemctl start nginx
```

Ubuntu 16.04 LTS

Detención de cualquier proceso de NGINX en ejecución

```
$ sudo systemctl stop nginx
```

Recarga de la configuración `systemd` para incluir los últimos cambios

```
$ sudo systemctl daemon-reload
```

Inicio del proceso de NGINX

```
$ sudo systemctl start nginx
```

Ubuntu 18.04 LTS

Detención de cualquier proceso de NGINX en ejecución

```
$ sudo systemctl stop nginx
```

Recarga de la configuración `systemd` para incluir los últimos cambios

```
$ sudo systemctl daemon-reload
```

Inicio del proceso de NGINX

```
$ sudo systemctl start nginx
```

Ubuntu 20.04 LTS

Detención de cualquier proceso de NGINX en ejecución

```
$ sudo systemctl stop nginx
```

Recarga de la configuración `systemd` para incluir los últimos cambios

```
$ sudo systemctl daemon-reload
```

Inicio del proceso de NGINX

```
$ sudo systemctl start nginx
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

12. (Opcional) Configure su plataforma para iniciar NGINX en el arranque.

Amazon Linux

```
$ sudo chkconfig nginx on
```

Amazon Linux 2

```
$ sudo systemctl enable nginx
```

CentOS 7

No hay que hacer nada.

CentOS 8

```
$ sudo systemctl enable nginx
```

Red Hat 7

No hay que hacer nada.

Red Hat 8

```
$ sudo systemctl enable nginx
```

Ubuntu 16.04 LTS

```
$ sudo systemctl enable nginx
```

Ubuntu 18.04 LTS

```
$ sudo systemctl enable nginx
```

Ubuntu 20.04 LTS

```
$ sudo systemctl enable nginx
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

Después de actualizar la configuración del servidor web, vaya a Paso 4: habilitar el tráfico HTTPS y verificar el certificado.

Configuración del servidor web Apache

Use esta sección para configurar Apache en las plataformas compatibles.

Para actualizar la configuración del servidor web para Apache

1. Conéctese a su instancia de cliente de Amazon EC2.
2. Defina las ubicaciones predeterminadas de los certificados y las claves privadas de su plataforma.

   Amazon Linux

   En el archivo `/etc/httpd/conf.d/ssl.conf`, asegúrese de que existan estos valores:

   ```
   SSLCertificateFile      /etc/pki/tls/certs/localhost.crt
   SSLCertificateKeyFile   /etc/pki/tls/private/localhost.key
   ```

   Amazon Linux 2

   En el archivo `/etc/httpd/conf.d/ssl.conf`, asegúrese de que existan estos valores:

   ```
   SSLCertificateFile      /etc/pki/tls/certs/localhost.crt
   SSLCertificateKeyFile   /etc/pki/tls/private/localhost.key
   ```

   CentOS 7

   En el archivo `/etc/httpd/conf.d/ssl.conf`, asegúrese de que existan estos valores:

   ```
   SSLCertificateFile      /etc/pki/tls/certs/localhost.crt
   SSLCertificateKeyFile   /etc/pki/tls/private/localhost.key
   ```

   CentOS 8

   En el archivo `/etc/httpd/conf.d/ssl.conf`, asegúrese de que existan estos valores:

   ```
   SSLCertificateFile      /etc/pki/tls/certs/localhost.crt
   SSLCertificateKeyFile   /etc/pki/tls/private/localhost.key
   ```

Red Hat 7

En el archivo `/etc/httpd/conf.d/ssl.conf`, asegúrese de que existan estos valores:

```
SSLCertificateFile      /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile   /etc/pki/tls/private/localhost.key
```

Red Hat 8

En el archivo `/etc/httpd/conf.d/ssl.conf`, asegúrese de que existan estos valores:

```
SSLCertificateFile      /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile   /etc/pki/tls/private/localhost.key
```

Ubuntu 16.04 LTS

En el archivo `/etc/apache2/sites-available/default-ssl.conf`, asegúrese de que existan estos valores:

```
SSLCertificateFile      /etc/ssl/certs/localhost.crt
SSLCertificateKeyFile   /etc/ssl/private/localhost.key
```

Ubuntu 18.04 LTS

En el archivo `/etc/apache2/sites-available/default-ssl.conf`, asegúrese de que existan estos valores:

```
SSLCertificateFile      /etc/ssl/certs/localhost.crt
SSLCertificateKeyFile   /etc/ssl/private/localhost.key
```

Ubuntu 20.04 LTS

En el archivo `/etc/apache2/sites-available/default-ssl.conf`, asegúrese de que existan estos valores:

```
SSLCertificateFile      /etc/ssl/certs/localhost.crt
SSLCertificateKeyFile   /etc/ssl/private/localhost.key
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

3. Copie el certificado de su servidor web en la ubicación requerida según su plataforma.

Amazon Linux

```
$ sudo cp <web_server.crt> /etc/pki/tls/certs/localhost.crt
```

Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

Amazon Linux 2

```
$ sudo cp <web_server.crt> /etc/pki/tls/certs/localhost.crt
```

Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

CentOS 7

```
$ sudo cp <web_server.crt> /etc/pki/tls/certs/localhost.crt
```

Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

CentOS 8

```
$ sudo cp <web_server.crt> /etc/pki/tls/certs/localhost.crt
```

Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

Red Hat 7

```
$ sudo cp <web_server.crt> /etc/pki/tls/certs/localhost.crt
```

Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

Red Hat 8

```
$ sudo cp <web_server.crt> /etc/pki/tls/certs/localhost.crt
```

Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

Ubuntu 16.04 LTS

```
$ sudo cp <web_server.crt> /etc/ssl/certs/localhost.crt
```

Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

Ubuntu 18.04 LTS

```
$ sudo cp <web_server.crt> /etc/ssl/certs/localhost.crt
```

Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

Ubuntu 20.04 LTS

```
$ sudo cp <web_server.crt> /etc/ssl/certs/localhost.crt
```

Sustituya *<web_server.crt>* por el nombre del certificado de servidor web.

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

4. Copie su clave privada PEM falsa en la ubicación requerida según su plataforma.

Amazon Linux

```
$ sudo cp <web_server_fake_PEM.key> /etc/pki/tls/private/localhost.key
```

Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

Amazon Linux 2

```
$ sudo cp <web_server_fake_PEM.key> /etc/pki/tls/private/localhost.key
```

Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

CentOS 7

```
$ sudo cp <web_server_fake_PEM.key> /etc/pki/tls/private/localhost.key
```

Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

CentOS 8

```
$ sudo cp <web_server_fake_PEM.key> /etc/pki/tls/private/localhost.key
```

Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

Red Hat 7

```
$ sudo cp <web_server_fake_PEM.key> /etc/pki/tls/private/localhost.key
```

Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

Red Hat 8

```
$ sudo cp <web_server_fake_PEM.key> /etc/pki/tls/private/localhost.key
```

Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

Ubuntu 16.04 LTS

```
$ sudo cp <web_server_fake_PEM.key> /etc/ssl/private/localhost.key
```

Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

Ubuntu 18.04 LTS

```
$ sudo cp <web_server_fake_PEM.key> /etc/ssl/private/localhost.key
```

Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

Ubuntu 20.04 LTS

```
$ sudo cp <web_server_fake_PEM.key> /etc/ssl/private/localhost.key
```

Reemplace *<web_server_fake_PEM.key>* por el nombre del archivo que contiene la clave privada de PEM falsa.

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

5. Cambie la propiedad de estos archivos si así lo requiere su plataforma.

Amazon Linux

```
$ sudo chown apache /etc/pki/tls/certs/localhost.crt /etc/pki/tls/private/
localhost.key
```

Proporciona permisos de lectura al usuario apache.

Amazon Linux 2

```
$ sudo chown apache /etc/pki/tls/certs/localhost.crt /etc/pki/tls/private/
localhost.key
```

Proporciona permisos de lectura al usuario apache.

CentOS 7

```
$ sudo chown apache /etc/pki/tls/certs/localhost.crt /etc/pki/tls/private/
localhost.key
```

Proporciona permisos de lectura al usuario apache.

CentOS 8

```
$ sudo chown apache /etc/pki/tls/certs/localhost.crt /etc/pki/tls/private/
localhost.key
```

Proporciona permisos de lectura al usuario apache.

Red Hat 7

```
$ sudo chown apache /etc/pki/tls/certs/localhost.crt /etc/pki/tls/private/
localhost.key
```

Proporciona permisos de lectura al usuario apache.

Red Hat 8

```
$ sudo chown apache /etc/pki/tls/certs/localhost.crt /etc/pki/tls/private/
localhost.key
```

Proporciona permisos de lectura al usuario apache.

Ubuntu 16.04 LTS

No hay que hacer nada.

Ubuntu 18.04 LTS

No hay que hacer nada.

Ubuntu 20.04 LTS

No hay que hacer nada.

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

6. Configure las directivas de Apache para su plataforma.

Amazon Linux

Localice el archivo SSL para esta plataforma:

```
/etc/httpd/conf.d/ssl.conf
```

Este archivo contiene las directivas de Apache que definen el funcionamiento del servidor. Las directivas se muestran a la izquierda, seguidas de un valor. Utilice un editor de texto para editar el archivo. Esto requiere permisos de raíz de Linux.

Actualice o introduzca las siguientes directivas con estos valores:

```
SSLCryptoDevice cloudhsm
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA
```

Guarde el archivo.

Amazon Linux 2

Localice el archivo SSL para esta plataforma:

```
/etc/httpd/conf.d/ssl.conf
```

Este archivo contiene las directivas de Apache que definen el funcionamiento del servidor. Las directivas se muestran a la izquierda, seguidas de un valor. Utilice un editor de texto para editar el archivo. Esto requiere permisos de raíz de Linux.

Actualice o introduzca las siguientes directivas con estos valores:

```
SSLCryptoDevice cloudhsm
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA
```

Guarde el archivo.

CentOS 7

Localice el archivo SSL para esta plataforma:

```
/etc/httpd/conf.d/ssl.conf
```

Este archivo contiene las directivas de Apache que definen el funcionamiento del servidor. Las directivas se muestran a la izquierda, seguidas de un valor. Utilice un editor de texto para editar el archivo. Esto requiere permisos de raíz de Linux.

Actualice o introduzca las siguientes directivas con estos valores:

```
SSLCryptoDevice cloudhsm
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA
```

Guarde el archivo.

CentOS 8

Localice el archivo SSL para esta plataforma:

```
/etc/httpd/conf.d/ssl.conf
```

Este archivo contiene las directivas de Apache que definen el funcionamiento del servidor. Las directivas se muestran a la izquierda, seguidas de un valor. Utilice un editor de texto para editar el archivo. Esto requiere permisos de raíz de Linux.

Actualice o introduzca las siguientes directivas con estos valores:

```
SSLCryptoDevice cloudhsm
SSLProtocol TLSv1.2 TLSv1.3
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA
SSLProxyCipherSuite HIGH:!aNULL
```

Guarde el archivo.

Red Hat 7

Localice el archivo SSL para esta plataforma:

```
/etc/httpd/conf.d/ssl.conf
```

Este archivo contiene las directivas de Apache que definen el funcionamiento del servidor. Las directivas se muestran a la izquierda, seguidas de un valor. Utilice un editor de texto para editar el archivo. Esto requiere permisos de raíz de Linux.

Actualice o introduzca las siguientes directivas con estos valores:

```
SSLCryptoDevice cloudhsm
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA
```

Guarde el archivo.

Red Hat 8

Localice el archivo SSL para esta plataforma:

```
/etc/httpd/conf.d/ssl.conf
```

Este archivo contiene las directivas de Apache que definen el funcionamiento del servidor. Las directivas se muestran a la izquierda, seguidas de un valor. Utilice un editor de texto para editar el archivo. Esto requiere permisos de raíz de Linux.

Actualice o introduzca las siguientes directivas con estos valores:

```
SSLCryptoDevice cloudhsm
SSLProtocol TLSv1.2 TLSv1.3
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
```

```
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA
SSLProxyCipherSuite HIGH:!aNULL
```

Guarde el archivo.

Ubuntu 16.04 LTS

Localice el archivo SSL para esta plataforma:

```
/etc/apache2/mods-available/ssl.conf
```

Este archivo contiene las directivas de Apache que definen el funcionamiento del servidor. Las directivas se muestran a la izquierda, seguidas de un valor. Utilice un editor de texto para editar el archivo. Esto requiere permisos de raíz de Linux.

Actualice o introduzca las siguientes directivas con estos valores:

```
SSLCryptoDevice cloudhsm
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA
```

Guarde el archivo.

Habilite el módulo SSL y la configuración predeterminada del sitio SSL:

```
$ sudo a2enmod ssl
$ sudo a2ensite default-ssl
```

Ubuntu 18.04 LTS

Localice el archivo SSL para esta plataforma:

```
/etc/apache2/mods-available/ssl.conf
```

Este archivo contiene las directivas de Apache que definen el funcionamiento del servidor. Las directivas se muestran a la izquierda, seguidas de un valor. Utilice un editor de texto para editar el archivo. Esto requiere permisos de raíz de Linux.

Actualice o introduzca las siguientes directivas con estos valores:

```
SSLCryptoDevice cloudhsm
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA
SSLProtocol TLSv1.2 TLSv1.3
```

Guarde el archivo.

Habilite el módulo SSL y la configuración predeterminada del sitio SSL:

```
$ sudo a2enmod ssl
$ sudo a2ensite default-ssl
```

Ubuntu 20.04 LTS

Localice el archivo SSL para esta plataforma:

```
/etc/apache2/mods-available/ssl.conf
```

Este archivo contiene las directivas de Apache que definen el funcionamiento del servidor. Las directivas se muestran a la izquierda, seguidas de un valor. Utilice un editor de texto para editar el archivo. Esto requiere permisos de raíz de Linux.

Actualice o introduzca las siguientes directivas con estos valores:

```
SSLCryptoDevice cloudhsm
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES256-GCM-
```

```
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA
SSLProtocol TLSv1.2 TLSv1.3
```

Guarde el archivo.

Habilite el módulo SSL y la configuración predeterminada del sitio SSL:

```
$ sudo a2enmod ssl
$ sudo a2ensite default-ssl
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

7. Configure un archivo de valores de entorno para su plataforma.

Amazon Linux

No hay que hacer nada. Los valores de entorno se introducen en `/etc/sysconfig/httpd`

Amazon Linux 2

Abra el archivo de servicio httpd:

```
/lib/systemd/system/httpd.service
```

Añada lo siguiente a la sección `[Service]`:

```
EnvironmentFile=/etc/sysconfig/httpd
```

CentOS 7

Abra el archivo de servicio httpd:

```
/lib/systemd/system/httpd.service
```

Añada lo siguiente a la sección `[Service]`:

```
EnvironmentFile=/etc/sysconfig/httpd
```

CentOS 8

Abra el archivo de servicio httpd:

```
/lib/systemd/system/httpd.service
```

Añada lo siguiente a la sección `[Service]`:

```
EnvironmentFile=/etc/sysconfig/httpd
```

Red Hat 7

Abra el archivo de servicio httpd:

```
/lib/systemd/system/httpd.service
```

Añada lo siguiente a la sección `[Service]`:

```
EnvironmentFile=/etc/sysconfig/httpd
```

Red Hat 8

Abra el archivo de servicio httpd:

```
/lib/systemd/system/httpd.service
```

Añada lo siguiente a la sección `[Service]`:

```
EnvironmentFile=/etc/sysconfig/httpd
```

Ubuntu 16.04 LTS

No hay que hacer nada. Los valores de entorno se introducen en `/etc/sysconfig/httpd`

Ubuntu 18.04 LTS

No hay que hacer nada. Los valores de entorno se introducen en `/etc/sysconfig/httpd`

Ubuntu 20.04 LTS

No hay que hacer nada. Los valores de entorno se introducen en `/etc/sysconfig/httpd`

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

8. En el archivo que almacena las variables de entorno de su plataforma, defina una variable de entorno que contenga las credenciales del usuario de criptografía (CU):

Amazon Linux

Utilice un editor de texto para editar el `/etc/sysconfig/httpd`.

- Si usa SDK 3 de cliente

```
n3fips_password=<CU user name>:<password>
```

- Si usa SDK 5 de cliente

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Amazon Linux 2

Utilice un editor de texto para editar el `/etc/sysconfig/httpd`.

- Si usa SDK 3 de cliente

```
n3fips_password=<CU user name>:<password>
```

- Si usa SDK 5 de cliente

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

CentOS 7

Utilice un editor de texto para editar el `/etc/sysconfig/httpd`.

- Si usa SDK 3 de cliente

```
n3fips_password=<CU user name>:<password>
```

- Si usa SDK 5 de cliente

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

CentOS 8

Utilice un editor de texto para editar el `/etc/sysconfig/httpd`.

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Red Hat 7

Utilice un editor de texto para editar el `/etc/sysconfig/httpd`.

- Si usa SDK 3 de cliente

```
n3fips_password=<CU user name>:<password>
```

- Si usa SDK 5 de cliente

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Red Hat 8

Utilice un editor de texto para editar el `/etc/sysconfig/httpd`.

```
CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

> **Note**
>
> El SDK 5 de cliente introduce la variable de entorno `CLOUDHSM_PIN` para almacenar las credenciales del CU.

Ubuntu 16.04 LTS

Utilice un editor de texto para editar el `/etc/apache2/envvars`.

```
export n3fips_password=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

Ubuntu 18.04 LTS

Utilice un editor de texto para editar el `/etc/apache2/envvars`.

```
export CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

> **Note**
>
> El SDK 5 de cliente introduce la variable de entorno `CLOUDHSM_PIN` para almacenar las credenciales del CU. SDK 3 de cliente introduce la variable de entorno `n3fips_password` para almacenar las credenciales de CU. El SDK 5 de cliente admite ambas variables de entorno, pero recomendamos utilizar `CLOUDHSM_PIN`.

Ubuntu 20.04 LTS

Utilice un editor de texto para editar el `/etc/apache2/envvars`.

```
export CLOUDHSM_PIN=<CU user name>:<password>
```

Sustituya *<CU user name>* y *<password>* por las credenciales del usuario de criptografía.

> **Note**
>
> El SDK 5 de cliente introduce la variable de entorno `CLOUDHSM_PIN` para almacenar las credenciales del CU. SDK 3 de cliente introduce la variable de entorno `n3fips_password` para almacenar las credenciales de CU. El SDK 5 de cliente admite ambas variables de entorno, pero recomendamos utilizar `CLOUDHSM_PIN`.

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

9. Inicie el servidor web Apache.

Amazon Linux

```
$ sudo systemctl daemon-reload
$ sudo service httpd start
```

Amazon Linux 2

```
$ sudo systemctl daemon-reload
$ sudo service httpd start
```

CentOS 7

```
$ sudo systemctl daemon-reload
$ sudo service httpd start
```

CentOS 8

```
$ sudo systemctl daemon-reload
$ sudo service httpd start
```

Red Hat 7

```
$ sudo systemctl daemon-reload
$ sudo service httpd start
```

Red Hat 8

```
$ sudo systemctl daemon-reload
$ sudo service httpd start
```

Ubuntu 16.04 LTS

```
$ sudo service apache2 start
```

Ubuntu 18.04 LTS

```
$ sudo service apache2 start
```

Ubuntu 20.04 LTS

```
$ sudo service apache2 start
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

10. (Opcional) Configure su plataforma para iniciar Apache en el arranque.

Amazon Linux

```
$ sudo chkconfig httpd on
```

Amazon Linux 2

```
$ sudo chkconfig httpd on
```

CentOS 7

```
$ sudo chkconfig httpd on
```

CentOS 8

```
$ systemctl enable httpd
```

Red Hat 7

```
$ sudo chkconfig httpd on
```

Red Hat 8

```
$ systemctl enable httpd
```

Ubuntu 16.04 LTS

```
$ sudo systemctl enable apache2
```

Ubuntu 18.04 LTS

```
$ sudo systemctl enable apache2
```

Ubuntu 20.04 LTS

```
$ sudo systemctl enable apache2
```

Ubuntu 22.04 LTS

La compatibilidad con el motor dinámico de OpenSSL aún no está disponible.

Después de actualizar la configuración del servidor web, vaya a Paso 4: habilitar el tráfico HTTPS y verificar el certificado.

Paso 4: habilitar el tráfico HTTPS y verificar el certificado

Después de configurar el servidor web para la descarga de SSL/TLS, añada la instancia del servidor web a un grupo de seguridad que AWS CloudHSM permita el tráfico HTTPS entrante. Esto permite a los clientes, como, por ejemplo, navegadores web, establecer una conexión HTTPS con su servidor web. A continuación, establece una conexión HTTPS con tu servidor web y comprueba que utiliza el certificado con el que configuraste la descarga de SSL/TLS. AWS CloudHSM

Temas

- Habilitación de las conexiones HTTPS entrantes

- Verificación del uso del certificado configurado por parte de HTTPS

Habilitación de las conexiones HTTPS entrantes

Para conectarse a su servidor web desde un cliente (como, por ejemplo, un navegador web), cree un grupo de seguridad que permita conexiones HTTPS entrantes. En concreto, debería permitir conexiones TCP entrantes en el puerto 443. Asigne este grupo de seguridad a su servidor web.

Para crear un grupo de seguridad para HTTPS y asignarlo a su servidor web

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Elija Security Groups en el panel de navegación.
3. Elija Crear grupo de seguridad.
4. En Create Security Group (Crear grupo de seguridad), haga lo siguiente:
    a. Para Security group name (Nombre del grupo de seguridad), escriba un nombre para el grupo de seguridad que está creando.
    b. De manera opcional, escriba una descripción del grupo de seguridad que está creando.
    c. Para VPC, elija la VPC que contiene la instancia de su servidor web Amazon EC2.
    d. Seleccione Add Rule (Añadir regla).
    e. Para tipo, seleccione HTTPS en la ventana desplegable.
    f. Para Origen, introduzca una ubicación de origen.
    g. Elija Crear grupo de seguridad.
5. En el panel de navegación, seleccione Instancias.
6. Seleccione la casilla de verificación junto a la instancia del servidor web.
7. Seleccione las Acciones en el menú desplegable que se encuentra en la parte superior de la página. Seleccione Seguridad, a continuación, Cambiar grupos de seguridad.
8. Para Grupos de seguridad asociados, seleccione el cuadro de búsqueda y elija el grupo de seguridad que creó para HTTPS. A continuación, elija Añadir grupos de seguridad.
9. Seleccione Guardar.

Verificación del uso del certificado configurado por parte de HTTPS

Después de añadir el servidor web a un grupo de seguridad, puede verificar que la descarga de SSL/TLS utiliza su certificado autofirmado. Puede hacerlo mediante un navegador web o con una herramienta como OpenSSL s_client.

Para verificar la descarga de SSL/TLS con un navegador web

1. Utilice un navegador web para conectarse a su servidor web mediante el nombre de DNS público o la dirección IP del servidor. Asegúrese de que la dirección URL en la barra de direcciones comienza con https://. Por ejemplo, **`https://ec2-52-14-212-67.us-east-2.compute.amazonaws.com/`**.

   > **Tip**
   >
   > Puede utilizar un servicio DNS como, por ejemplo, Amazon Route 53, para dirigir el nombre de dominio de su sitio web (por ejemplo, https://www.ejemplo.com/) a su servidor web. Para obtener más información, consulte Direccionamiento del tráfico a una instancia de Amazon EC2 en la Guía para desarrolladores de Amazon Route 53 o en la documentación para su servicio DNS.

2. Utilice el navegador web para ver el certificado del servidor web. Para más información, consulte los siguientes temas:

   - Para Mozilla Firefox, consulte View a Certificate en el sitio web de Soporte de Mozilla.
   - Para Google Chrome, consulte Conocer los problemas de seguridad en el sitio web para desarrolladores de Google.

   Otros navegadores web pueden tener características similares que puede utilizar para ver el certificado del servidor web.

3. Asegúrese de que el certificado SSL/TLS es el que ha configurado para que utilice el servidor web.

Para verificar la descarga de SSL/TLS con OpenSSL s_client

1. Ejecute el siguiente comando OpenSSL para conectarse a su servidor web a través de HTTPS. Sustituya *`<server name>`* por el nombre de DNS público o la dirección IP de su servidor web.

```
openssl s_client -connect <server name>:443
```

> **Tip**
>
> Puede utilizar un servicio DNS como, por ejemplo, Amazon Route 53, para dirigir el nombre de dominio de su sitio web (por ejemplo, https://www.ejemplo.com/) a su servidor web. Para obtener más información, consulte Direccionamiento del tráfico a una instancia de Amazon EC2 en la Guía para desarrolladores de Amazon Route 53 o en la documentación para su servicio DNS.

2. Asegúrese de que el certificado SSL/TLS es el que ha configurado para que utilice el servidor web.

Ahora tiene un sitio web que se protege con HTTPS. La clave privada del servidor web se almacena en un HSM del clúster. AWS CloudHSM

Para agregar un equilibrador de carga, consulte Agregar un equilibrador de carga con Elastic Load Balancing (opcional).

Uso de Tomcat con JSSE para la descarga de SSL/TLS en Linux

En este tema se proporcionan step-by-step instrucciones para configurar la descarga de SSL/TLS mediante Java Secure Socket Extension (JSSE) con el SDK de JCE. AWS CloudHSM

Temas

- Información general
- Paso 1: configurar los requisitos previos
- Paso 2: generar o importar una clave privada y un certificado SSL/TLS
- Paso 3:cConfigurar el servidor web Tomcat
- Paso 4: habilitar el tráfico HTTPS y verificar el certificado

Información general

En AWS CloudHSM, los servidores web de Tomcat funcionan en Linux para admitir HTTPS. El SDK de AWS CloudHSM JCE proporciona una interfaz que se puede utilizar con JSSE (Java Secure Socket Extension) para permitir el uso de HSM en dichos servidores web. AWS CloudHSM JCE es

el puente que conecta JSSE con su clúster de AWS CloudHSM. JSSE es una API de Java para los protocolos Capa de conexión segura (SSL) y Seguridad de la capa de transporte (TLS).

Paso 1: configurar los requisitos previos

Siga estos requisitos previos para utilizar un servidor web Tomcat con descarga de SSL/TLS en Linux. AWS CloudHSM Estos requisitos previos deben cumplirse para configurar la descarga de SSL/TLS en un servidor web con el SDK 5 de cliente y un servidor web Tomcat.

> **Note**
>
> Las diferentes plataformas requieren requisitos previos diferentes. Siga siempre los pasos de instalación adecuados para su plataforma.

Requisitos previos

- Una instancia de Amazon EC2 que ejecuta un sistema operativo Linux con un servidor web Tomcat instalado.
- Un usuario de criptografía (CU) que sea el propietario y administre la clave privada del servidor web en el HSM.
- Un AWS CloudHSM clúster activo con al menos dos módulos de seguridad de hardware (HSM) que tengan instalado y configurado el JCE for Client SDK 5.

> **Note**
>
> Puede usar un único clúster de HSM, pero primero debe deshabilitar la durabilidad de la clave de cliente. Para obtener más información, consulte Administrar la configuración de durabilidad de las claves de cliente y Herramienta de configuración del SDK 5 de cliente.

Cómo cumplir los requisitos previos

1. Instale y configure el JCE AWS CloudHSM en un AWS CloudHSM clúster activo con al menos dos módulos de seguridad de hardware (HSM). Para obtener más información sobre la instalación, consulte JCE para SDK 5 de cliente.
2. En una instancia Linux EC2 que tenga acceso a su AWS CloudHSM clúster, siga las instrucciones de Apache Tomcat para descargar e instalar el servidor web Tomcat.

3. Utilice la CLI de CloudHSM para crear un usuario de criptografía (CU). Para obtener más información sobre la administración de los usuarios de HSM, consulte Administrar a los usuarios de HSM con la CLI de CloudHSM.

   > **Tip**
   >
   > Realice un seguimiento del nombre de usuario y la contraseña del CU. Los necesitará más adelante cuando genere o importe el certificado y la clave privada de HTTPS para el servidor web.

4. Para configurar JCE con Java Keytool, siga las instrucciones de Uso de SDK 5 de cliente para la integración con Java Keytool y Jarsigner.

Después de completar estos pasos, vaya a Paso 2: generar o importar una clave privada y un certificado SSL/TLS.

Notas

- Para usar Linux con seguridad mejorada (SELinux) y servidores web, debe permitir las conexiones TCP salientes en el puerto 2223, que es el puerto que el SDK 5 de cliente utiliza para comunicarse con el HSM.
- Para crear y activar un clúster y permitir que una instancia EC2 acceda al clúster, complete los pasos que se indican en Introducción a AWS CloudHSM. En esta sección se ofrecen step-by-step instrucciones para crear un clúster activo con un HSM y una instancia de cliente de Amazon EC2. Puede utilizar esta instancia de cliente como su servidor web.
- Para evitar deshabilitar la durabilidad de la clave de cliente, agregue más de un HSM a su clúster. Para obtener más información, consulte Agregar un HSM.
- Para conectarse a su instancia de cliente, puede utilizar SSH o PuTTY. Para obtener más información, consulte Conectarse a la instancia de Linux mediante SSH o Conectarse a la instancia de Linux desde Windows mediante PuTTY en la documentación de Amazon EC2.

Paso 2: generar o importar una clave privada y un certificado SSL/TLS

Para habilitar HTTPS, su aplicación de servidor web Tomcat necesita una clave privada y un certificado SSL/TLS correspondiente. Para utilizar la descarga de SSL/TLS de un servidor web AWS CloudHSM, debe almacenar la clave privada en un HSM de su clúster. AWS CloudHSM

> **Note**
>
> Si todavía no dispone de una clave privada y un certificado correspondiente, genere una clave privada en un HSM. Use la clave privada para crear una solicitud de firma de certificado (CSR), la cual utilizará para crear un certificado SSL/TLS.

Debe crear un AWS CloudHSM KeyStore archivo local que contenga una referencia a su clave privada en el HSM y al certificado asociado. El servidor web utiliza el AWS CloudHSM KeyStore archivo para identificar la clave privada en el HSM durante la descarga de SSL/TLS.

Temas

- Generación de una clave privada
- Generación de un certificado autofirmado

Generación de una clave privada

En esta sección se muestra cómo generar un par de claves utilizando el JDK. KeyTool Una vez que haya generado un par de claves dentro del HSM, puede exportarlo como un KeyStore archivo y generar el certificado correspondiente.

Según su caso de uso, puede generar un par de claves RSA o EC. Los siguientes pasos muestran cómo generar un par de claves de RSA.

Use el **`genkeypair`** comando in KeyTool para generar un key pair de RSA

1. Tras sustituir las siguientes *<VARIABLES>* por sus datos específicos, utilice el siguiente comando para generar un archivo de almacén de claves denominado `jsse_keystore.keystore`, que tendrá una referencia a su clave privada en el HSM.

```
$ keytool -genkeypair -alias <UNIQUE ALIAS FOR KEYS> -keyalg <KEY ALGORITHM> -
keysize <KEY SIZE> -sigalg <SIGN ALGORITHM> \
        -keystore <PATH>/<JSSE KEYSTORE NAME>.keystore -storetype CLOUDHSM \
        -dname CERT_DOMAIN_NAME \
        -J-classpath '-J'$JAVA_LIB'/*:/opt/cloudhsm/java/*:./*' \
        -provider "com.amazonaws.cloudhsm.jce.provider.CloudHsmProvider" \
        -providerpath "$CLOUDHSM_JCE_LOCATION" \
        -keypass <KEY PASSWORD> -storepass <KEYSTORE PASSWORD>
```

- ***<PATH>***: la ruta en la que desea generar su archivo de almacén de claves.
- ***<UNIQUE ALIAS FOR KEYS>***: se usa para identificar de forma exclusiva su clave en el HSM. Este alias se establecerá como atributo LABEL de la clave.
- ***<KEY PASSWORD>***: almacenamos la referencia de su clave en el archivo de almacén de claves local. Esta contraseña protege la referencia local.
- ***<KEYSTORE PASSWORD>***: esta es la contraseña del archivo del almacén de claves local.
- ***<JSSE KEYSTORE NAME>***: nombre del archivo de almacén de claves.
- ***<CERT DOMAIN NAME>***: nombre distintivo X.500.
- ***<KEY ALGORITHM>***: algoritmo clave para generar un par de claves (por ejemplo, RSA y EC).
- ***<KEY SIZE>***: tamaño de clave para generar el par de claves (por ejemplo, 2048, 3072 y 4096).
- ***<SIGN ALGORITHM>***: tamaño de la clave para generar el par de claves (por ejemplo, SHA1withRSA, SHA224withRSA, SHA256withRSA, SHA384withRSA y SHA512withRSA).

2. Para confirmar que el comando se ha utilizado correctamente, introduzca el siguiente comando y compruebe que ha generado correctamente un par de claves RSA.

```
$ ls <PATH>/<JSSE KEYSTORE NAME>.keystore
```

Generación de un certificado autofirmado

Una vez que haya generado una clave privada junto con el archivo del almacén de claves, puede usar este archivo para generar una solicitud de firma de certificado (CSR) y un certificado.

En un entorno de producción, normalmente se usa una entidad de certificación (CA) para crear un certificado de una CSR. No es necesaria una CA para un entorno de prueba. Si utiliza una CA, envíele el archivo CSR y utilice el certificado SSL/TLS firmado que le proporcione en su servidor web para HTTPS.

Como alternativa al uso de una CA, puede utilizarla KeyTool para crear un certificado autofirmado. Los navegadores no confían en certificados autofirmados y no deben utilizarse en entornos de producción. Se pueden usar en entornos de prueba.

> **Warning**
>
> Los certificados autofirmados deben utilizarse únicamente en entornos de prueba. En entornos de producción, utilice un método más seguro como, por ejemplo, una entidad de certificación para crear un certificado.

Generación de un certificado

1. Obtenga una copia del archivo de almacén de claves generado en un paso anterior.
2. Ejecute el siguiente comando para usar el y KeyTool crear una solicitud de firma de certificado (CSR).

```
$ keytool -certreq -keyalg RSA -alias unique_alias_for_key -file certreq.csr \
        -keystore <JSSE KEYSTORE NAME>.keystore -storetype CLOUDHSM \
        -J-classpath '-J$JAVA_LIB/*:/opt/cloudhsm/java/*:./*' \
        -keypass <KEY PASSWORD> -storepass <KEYSTORE PASSWORD>
```

> **Note**
>
> El archivo de salida de la solicitud de firma de certificado es `certreq.csr`.

Firma de un certificado

- Tras sustituir las siguientes *<VARIABLES>* por sus datos específicos, ejecute el siguiente comando para firmar su CSR con su clave privada en su HSM. Esto creará un certificado autofirmado.

```
$ keytool -gencert -infile certreq.csr -outfile certificate.crt \
    -alias <UNIQUE ALIAS FOR KEYS> -keypass <KEY_PASSWORD> -
storepass <KEYSTORE_PASSWORD> -sigalg SIG_ALG \
    -storetype CLOUDHSM -J-classpath '-J$JAVA_LIB/*:/opt/cloudhsm/java/*:./*' \
    -keystore jsse_keystore.keystore
```

> **Note**
>
> `certificate.crt` es el certificado firmado que usa la clave privada del alias.

Importación de un certificado en KeyStore

- Tras sustituir las siguientes *<VARIABLES>* por sus datos específicos, ejecute el siguiente comando para importar un certificado firmado como certificado de confianza. Este paso almacenará el certificado en la entrada del almacén de claves identificada por un alias.

```
$ keytool -import -alias <UNIQUE ALIAS FOR KEYS> -keystore jsse_keystore.keystore \
    -file certificate.crt -storetype CLOUDHSM \
    -v -J-classpath '-J$JAVA_LIB/*:/opt/cloudhsm/java/*:./*' \
    -keypass <KEY PASSWORD> -storepass <KEYSTORE_PASSWORD>
```

Conversión de un certificado en un PEM

- Ejecute el siguiente comando para convertir el archivo de certificado firmado (.crt) en un PEM. El archivo PEM se utilizará para enviar la solicitud desde el cliente http.

```
$ openssl x509 -inform der -in certificate.crt -out certificate.pem
```

Después de completar estos pasos, vaya al Paso 3: configurar el servidor web.

Paso 3:cConfigurar el servidor web Tomcat

Actualice la configuración de software del servidor web para utilizar el certificado HTTPS y el archivo PEM correspondiente que ha creado en el paso anterior. Recuerde hacer una copia de seguridad de sus certificados y claves existentes antes de empezar. De este modo, concluirá la configuración del software del servidor web de Linux para la descarga SSL/TLS con AWS CloudHSM. Para obtener más información, consulte Referencia de configuración de Apache Tomcat 9.

Detención del servidor

- Tras sustituir las siguientes *<VARIABLES>* por sus datos específicos, ejecute el siguiente comando para detener Tomcat Server antes de actualizar la configuración.

```
$ /<TOMCAT DIRECTORY>/bin/shutdown.sh
```

  - ***<TOMCAT DIRECTORY>***: el directorio de su instalación de Tomcat.

Actualización del classpath de Tomcat

1. Conéctese a su instancia de cliente.
2. Localice la carpeta de instalación de Tomcat.
3. Tras sustituir las siguientes *<VARIABLES>* por sus datos específicos, ejecute el siguiente comando para añadir la biblioteca de Java y la ruta de Java de Cloudhsm en el classpath de Tomcat, ubicado en el archivo Tomcat/bin/catalina.sh.

```
$ sed -i 's@CLASSPATH="$CLASSPATH""$CATALINA_HOME"\/bin\/
bootstrap.jar@CLASSPATH="$CLASSPATH""$CATALINA_HOME"\/bin\/bootstrap.jar:'"
        <JAVA LIBRARY>"'\/*:\/opt\/cloudhsm\/java\/*:.\/*@' <TOMCAT PATH> /bin/
catalina.sh
```

   - ***<JAVA LIBRARY>***: ubicación de la biblioteca de Java JRE.
   - ***<TOMCAT PATH>***: carpeta de instalación de Tomcat.

Añada un conector HTTPS a la configuración del servidor.

1. Acceda a la carpeta de instalación de Tomcat.
2. Tras sustituir las siguientes *<VARIABLES>* por sus datos específicos, ejecute el siguiente comando para añadir un conector HTTPS y usar los certificados generados como requisitos previos:

```
$ sed -i '/<Connector port="8080"/i <Connector port=\"443\" maxThreads=\"200\"
 scheme=\"https\" secure=\"true\" SSLEnabled=\"true\" keystoreType=\"CLOUDHSM\"
 keystoreFile=\"
        <CUSTOM DIRECTORY>/<JSSE KEYSTORE NAME>.keystore\" keystorePass=\"<KEYSTORE
 PASSWORD>\" keyPass=\"<KEY PASSWORD>
        \" keyAlias=\"<UNIQUE ALIAS FOR KEYS>" clientAuth=\"false\" sslProtocol=
\"TLS\"/>' <TOMCAT PATH>/conf/server.xml
```

   - ***<CUSTOM DIRECTORY>***: directorio en el que se encuentra el almacén de claves.
   - ***<JSSE KEYSTORE NAME>***: nombre del archivo de almacén de claves.
   - ***<KEYSTORE PASSWORD>***: esta es la contraseña del archivo del almacén de claves local.
   - ***<KEY PASSWORD>***: almacenamos la referencia de su clave en el archivo de almacén de claves local. Esta contraseña protege la referencia local.

- ***<UNIQUE ALIAS FOR KEYS>***: se usa para identificar de forma exclusiva su clave en el HSM. Este alias se establecerá como atributo LABEL de la clave.
- ***<TOMCAT PATH>***: ruta de la carpeta de Tomcat.

Inicio del servidor

- Tras sustituir las *<VARIABLES>* por sus datos específicos, ejecute el siguiente comando para detener el servidor Tomcat antes de actualizar la configuración

```
$ /<TOMCAT DIRECTORY>/bin/startup.sh
```

> **Note**
>
> ***<TOMCAT DIRECTORY>*** es el nombre del directorio de instalación de Tomcat.

Después de actualizar la configuración del servidor web, vaya a Paso 4: habilitar el tráfico HTTPS y verificar el certificado.

Paso 4: habilitar el tráfico HTTPS y verificar el certificado

Después de configurar el servidor web para la descarga de SSL/TLS, añada la instancia del servidor web a un grupo de seguridad que AWS CloudHSM permita el tráfico HTTPS entrante. Esto permite a los clientes, como, por ejemplo, navegadores web, establecer una conexión HTTPS con su servidor web. A continuación, establece una conexión HTTPS con tu servidor web y comprueba que utiliza el certificado con el que configuraste la descarga de SSL/TLS. AWS CloudHSM

Temas

- Habilitación de las conexiones HTTPS entrantes
- Verificación del uso del certificado configurado por parte de HTTPS

Habilitación de las conexiones HTTPS entrantes

Para conectarse a su servidor web desde un cliente (como, por ejemplo, un navegador web), cree un grupo de seguridad que permita conexiones HTTPS entrantes. En concreto, debería permitir conexiones TCP entrantes en el puerto 443. Asigne este grupo de seguridad a su servidor web.

Para crear un grupo de seguridad para HTTPS y asignarlo a su servidor web

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Elija Security Groups en el panel de navegación.
3. Elija Crear grupo de seguridad.
4. En Create Security Group (Crear grupo de seguridad), haga lo siguiente:
    a. Para Security group name (Nombre del grupo de seguridad), escriba un nombre para el grupo de seguridad que está creando.
    b. De manera opcional, escriba una descripción del grupo de seguridad que está creando.
    c. Para VPC, elija la VPC que contiene la instancia de su servidor web Amazon EC2.
    d. Seleccione Add Rule (Añadir regla).
    e. Para tipo, seleccione HTTPS en la ventana desplegable.
    f. Para Origen, introduzca una ubicación de origen.
    g. Elija Crear grupo de seguridad.
5. En el panel de navegación, seleccione Instancias.
6. Seleccione la casilla de verificación junto a la instancia del servidor web.
7. Seleccione las Acciones en el menú desplegable que se encuentra en la parte superior de la página. Seleccione Seguridad, a continuación, Cambiar grupos de seguridad.
8. Para Grupos de seguridad asociados, seleccione el cuadro de búsqueda y elija el grupo de seguridad que creó para HTTPS. A continuación, elija Añadir grupos de seguridad.
9. Seleccione Guardar.

Verificación del uso del certificado configurado por parte de HTTPS

Después de añadir el servidor web a un grupo de seguridad, puede verificar que la descarga de SSL/TLS utiliza su certificado autofirmado. Puede hacerlo mediante un navegador web o con una herramienta como OpenSSL s_client.

Para verificar la descarga de SSL/TLS con un navegador web

1. Utilice un navegador web para conectarse a su servidor web mediante el nombre de DNS público o la dirección IP del servidor. Asegúrese de que la dirección URL en la barra de direcciones comienza con https://. Por ejemplo, **`https://ec2-52-14-212-67.us-east-2.compute.amazonaws.com/`**.

> **Tip**
>
> Puede utilizar un servicio DNS como, por ejemplo, Amazon Route 53, para dirigir el nombre de dominio de su sitio web (por ejemplo, https://www.ejemplo.com/) a su servidor web. Para obtener más información, consulte Direccionamiento del tráfico a una instancia de Amazon EC2 en la Guía para desarrolladores de Amazon Route 53 o en la documentación para su servicio DNS.

2. Utilice el navegador web para ver el certificado del servidor web. Para más información, consulte los siguientes temas:

   - Para Mozilla Firefox, consulte View a Certificate en el sitio web de Soporte de Mozilla.
   - Para Google Chrome, consulte Conocer los problemas de seguridad en el sitio web para desarrolladores de Google.

   Otros navegadores web pueden tener características similares que puede utilizar para ver el certificado del servidor web.

3. Asegúrese de que el certificado SSL/TLS es el que ha configurado para que utilice el servidor web.

Para verificar la descarga de SSL/TLS con OpenSSL s_client

1. Ejecute el siguiente comando OpenSSL para conectarse a su servidor web a través de HTTPS. Sustituya *<server name>* por el nombre de DNS público o la dirección IP de su servidor web.

```
openssl s_client -connect <server name>:443
```

> **Tip**
>
> Puede utilizar un servicio DNS como, por ejemplo, Amazon Route 53, para dirigir el nombre de dominio de su sitio web (por ejemplo, https://www.ejemplo.com/) a su servidor web. Para obtener más información, consulte Direccionamiento del tráfico a una instancia de Amazon EC2 en la Guía para desarrolladores de Amazon Route 53 o en la documentación para su servicio DNS.

2. Asegúrese de que el certificado SSL/TLS es el que ha configurado para que utilice el servidor web.

Ahora tiene un sitio web que se protege con HTTPS. La clave privada del servidor web se almacena en un HSM del clúster. AWS CloudHSM

Para agregar un equilibrador de carga, consulte Agregar un equilibrador de carga con Elastic Load Balancing (opcional).

Uso de IIS con CNG para la descarga de SSL/TLS en Windows

Este tutorial proporciona step-by-step instrucciones para configurar la descarga de SSL/TLS en un servidor web de Windows. AWS CloudHSM

Temas

- Información general
- Paso 1: configurar los requisitos previos
- Paso 2: crear una solicitud de firma de certificado (CSR) y un certificado
- Paso 3: configurar el servidor web
- Paso 4: habilitar el tráfico HTTPS y verificar el certificado

Información general

En Windows, la aplicación del servidor web Internet Information Services (IIS) para Windows Server admite HTTPS de forma nativa. El proveedor de almacenamiento de claves (KSP) de AWS CloudHSM para la API de criptografía de nueva generación (CNG) de Microsoft proporciona la interfaz que permite a IIS utilizar los HSM del clúster para la descarga criptográfica y el almacenamiento de claves. El AWS CloudHSM KSP es el puente que conecta IIS con el clúster. AWS CloudHSM

Este tutorial le enseña a realizar las siguientes tareas:

- Instalar el software del servidor web en una instancia de Amazon EC2.
- Configurar el software del servidor web para que sea compatible con HTTPS mediante el uso de una clave privada almacenada en su clúster de AWS CloudHSM .
- (Opcional) Uso de Amazon EC2 para crear una segunda instancia de servidor web y Elastic Load Balancing para crear un equilibrador de carga. El uso de un equilibrador de carga puede mejorar el

desempeño al distribuir la carga entre varios servidores. También puede proporcionar redundancia y una mayor disponibilidad si uno o más servidores funcionan mal.

Cuando esté listo para empezar, vaya al Paso 1: configurar los requisitos previos.

Paso 1: configurar los requisitos previos

Para configurar la descarga de SSL/TLS en un servidor web, necesita lo siguiente: AWS CloudHSM

- Un AWS CloudHSM clúster activo con al menos un HSM.
- Una instancia de Amazon EC2 en la que se ejecuta un sistema operativo Windows y que tenga instalado el siguiente software:
  - El software de AWS CloudHSM cliente para Windows.
  - Internet Information Services (IIS) para Windows Server.
- Un usuario de criptografía (CU) que sea el propietario y administre la clave privada del servidor web en el HSM.

> **Note**
>
> Este tutorial usa Microsoft Windows Server 2016. También es posible utilizar Microsoft Windows Server 2012, pero no Microsoft Windows Server 2012 R2.

Para configurar una instancia de Windows Server y crear un CU en el HSM

1. Realice los pasos que se indican en Introducción. Cuando lance el cliente de Amazon EC2, seleccione una AMI de Windows Server 2016 o de Windows Server 2012. Cuando haya completado estos pasos, dispondrá de un clúster activo con al menos un HSM. También tiene una instancia de cliente Amazon EC2 que ejecuta Windows Server con el software AWS CloudHSM cliente para Windows instalado.
2. (Opcional) Añada más HSM a su clúster. Para obtener más información, consulte Agregar un HSM.
3. Conéctese al servidor de Windows. Para obtener más información, consulte Conectarse a la instancia en la Guía del usuario de Amazon EC2 para instancias de Windows.
4. Utilice la CLI de CloudHSM para crear un usuario de criptografía (CU). Realice un seguimiento del nombre de usuario y la contraseña del CU. Los necesitará para completar el paso siguiente.

> **Note**
>
> Para obtener información sobre la creación de un usuario, consulte Administrar usuarios de HSM con la CLI de CloudHSM.

5. Establezca las credenciales de inicio de sesión del HSM, utilizando el nombre de usuario y la contraseña del CU que creó en el paso anterior.
6. En el paso 5, si utilizó el Administrador de credenciales de Windows para configurar las credenciales de HSM, descargue `psexec.exe`desde SysInternals y ejecute el siguiente comando como NT Authority\ SYSTEM:

```
psexec.exe -s "C:\Program Files\Amazon\CloudHsm\tools\set_cloudhsm_credentials.exe"
 --username <USERNAME> --password <PASSWORD>
```

Reemplace *<NOMBRE DE USUARIO>* y *<CONTRASEÑA>* por las credenciales de HSM.

Para instalar IIS en Windows Server

1. Si aún no lo ha hecho, conéctese a su servidor de Windows. Para obtener más información, consulte Conectarse a la instancia en la Guía del usuario de Amazon EC2 para instancias de Windows.
2. En su servidor de Windows, inicie Administrador del servidor.
3. En el panel Administrador del servidor, elija Agregar roles y características.
4. Lea la información de Antes de comenzar y, a continuación, elija Siguiente.
5. En Installation Type, elija Instalación basada en características o en roles. A continuación, elija Next.
6. En Selección de servidor, elija Seleccionar un servidor del grupo de servidores. A continuación, elija Next.
7. En Roles de servidor, haga lo siguiente:
   a. Seleccione Servidor web (IIS).
   b. En Agregar características necesarias para Servidor web (IIS), elija Agregar características.
   c. Elija Siguiente para finalizar la selección de roles de servidor.
8. En Features (Características), acepte los valores predeterminados. A continuación, elija Next.

9. Lea la información sobre el Rol de servidor web (IIS). A continuación, elija Next.
10. En Seleccionar servicios de rol, acepte los valores predeterminados o cambie la configuración como desee. A continuación, elija Next.
11. En Confirmation (Confirmación), lea la información de confirmación. Después, seleccione Install (Instalar).
12. Cuando finalice la instalación, elija Cerrar.

Después de completar estos pasos, vaya a Paso 2: crear una solicitud de firma de certificado (CSR) y un certificado.

Paso 2: crear una solicitud de firma de certificado (CSR) y un certificado

Para habilitar HTTPS, el servidor web necesita un certificado SSL/TLS y la clave privada correspondiente. Para utilizar la descarga de SSL/TLS, debe almacenar la clave privada en el HSM de su clúster AWS CloudHSM. AWS CloudHSM Para ello, utilice el proveedor de almacenamiento de claves (KSP) de AWS CloudHSM para la API de criptografía de nueva generación (CNG) de Microsoft si desea crear una solicitud de firma de certificado (CSR). A continuación, debe proporcionar la CSR a una entidad de certificación (CA), para que firme la CSR y genere un certificado.

Temas

- Creación de una CSR
- Obtención e importación de un certificado firmado

Creación de una CSR

Utilice el AWS CloudHSM KSP de su servidor Windows para crear una CSR.

Para crear una CSR

1. Si aún no lo ha hecho, conéctese a su servidor de Windows. Para obtener más información, consulte Conectarse a la instancia en la Guía del usuario de Amazon EC2 para instancias de Windows.
2. Utilice el siguiente comando para iniciar el daemon del AWS CloudHSM cliente.

Amazon Linux

```
$ sudo start cloudhsm-client
```

Amazon Linux 2

```
$ sudo service cloudhsm-client start
```

CentOS 7

```
$ sudo service cloudhsm-client start
```

CentOS 8

```
$ sudo service cloudhsm-client start
```

RHEL 7

```
$ sudo service cloudhsm-client start
```

RHEL 8

```
$ sudo service cloudhsm-client start
```

Ubuntu 16.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client start
```

Windows

- Para la versión 1.1.2 y posteriores del cliente de Windows:

```
C:\Program Files\Amazon\CloudHSM>net.exe start AWSCloudHSMClient
```

- Para la versión 1.1.1 y anteriores de clientes de Windows:

```
C:\Program Files\Amazon\CloudHSM>start "cloudhsm_client" cloudhsm_client.exe
 C:\ProgramData\Amazon\CloudHSM\data\cloudhsm_client.cfg
```

3. En Windows Server, utilice un editor de texto para crear un archivo de solicitud de certificado denominado `IISCertRequest.inf`. A continuación, se muestra el contenido de un archivo `IISCertRequest.inf` de ejemplo. Para obtener más información sobre las secciones, las claves y los valores que puede especificar en el archivo, consulte la documentación de Microsoft. No cambie el valor de `ProviderName`.

```
[Version]
Signature = "$Windows NT$"
[NewRequest]
Subject = "CN=example.com,C=US,ST=Washington,L=Seattle,O=ExampleOrg,OU=WebServer"
HashAlgorithm = SHA256
KeyAlgorithm = RSA
KeyLength = 2048
ProviderName = "Cavium Key Storage Provider"
KeyUsage = 0xf0
MachineKeySet = True
[EnhancedKeyUsageExtension]
OID=1.3.6.1.5.5.7.3.1
```

4. Utilice el comando certreq de Windows para crear una CSR a partir del archivo `IISCertRequest.inf` que creó en el paso anterior. En el siguiente ejemplo, se guarda la CSR en un archivo denominado `IISCertRequest.csr`. Si utilizó un nombre de archivo diferente para el archivo de solicitud de certificado, sustituya el *`CertRequestarchivo.inf de IIS`* por el nombre de archivo adecuado. Si lo desea, puede sustituir el *`CertRequestarchivo.csr de IIS`* por un nombre de archivo diferente para el archivo CSR.

```
C:\>certreq -new IISCertRequest.inf IISCertRequest.csr
        SDK Version: 2.03

CertReq: Request Created
```

El archivo `IISCertRequest.csr` contiene la CSR. Necesita esta CSR para obtener un certificado firmado.

Obtención e importación de un certificado firmado

En un entorno de producción, normalmente se usa una entidad de certificación (CA) para crear un certificado de una CSR. No es necesaria una CA para un entorno de prueba. Si utiliza una CA, envíe el archivo de la CSR (`IISCertRequest.csr`) a la CA para que cree un certificado SSL/TLS firmado.

Como alternativa al uso de una CA, puede utilizar una herramienta como OpenSSL para crear un certificado autofirmado.

> **Warning**
>
> Los navegadores no confían en certificados autofirmados y no deben utilizarse en entornos de producción. Se pueden usar en entornos de prueba.

Los siguientes procedimientos muestran cómo crear un certificado autofirmado y cómo utilizarlo para firmar la CSR del servidor web.

Para crear un certificado autofirmado

1. Utilice el siguiente comando de OpenSSL para crear una clave privada. Si lo desea, puede sustituir *SelfSignedCA.key* por el nombre del archivo para que contenga su clave privada.

```
openssl genrsa -aes256 -out SelfSignedCA.key 2048
Generating RSA private key, 2048 bit long modulus
.....................................................................+++
.......................................+++
e is 65537 (0x10001)
Enter pass phrase for SelfSignedCA.key:
Verifying - Enter pass phrase for SelfSignedCA.key:
```

2. Utilice el siguiente comando de OpenSSL para crear un certificado autofirmado con la clave privada que ha creado en el paso anterior. Este es un comando interactivo. Lea las instrucciones que aparecen en pantalla y siga las indicaciones. Sustituya *SelfSignedCA.key* por el nombre del archivo que contiene la clave privada (si es diferente). Si lo desea, puede sustituir *Selfsignedca.crt* por el nombre del archivo para que contenga su certificado autofirmado.

```
openssl req -new -x509 -days 365 -key SelfSignedCA.key -out SelfSignedCA.crt
Enter pass phrase for SelfSignedCA.key:
You are about to be asked to enter information that will be incorporated
```

```
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

Para utilizar el certificado autofirmado para firmar la CSR del servidor web

- Utilice el siguiente comando de OpenSSL para utilizar la clave privada y el certificado autofirmado para firmar la CSR. Sustituya los nombres de los archivos siguientes por los que contienen los datos correspondientes (si son distintos).
  - *IIS CertRequest .csr*: nombre del archivo que contiene la CSR del servidor web
  - *SelfSignedCA.crt*: el nombre del archivo que contiene el certificado autofirmado
  - *SelfSignedCA.key*: el nombre del archivo que contiene la clave privada
  - *IISCert.crt*: nombre del archivo en que se va a guardar el certificado firmado del servidor web

```
openssl x509 -req -days 365 -in IISCertRequest.csr \
                            -CA SelfSignedCA.crt \
                            -CAkey SelfSignedCA.key \
                            -CAcreateserial \
                            -out IISCert.crt
Signature ok
subject=/ST=IIS-HSM/L=IIS-HSM/OU=IIS-HSM/O=IIS-HSM/CN=IIS-HSM/C=IIS-HSM
Getting CA Private Key
Enter pass phrase for SelfSignedCA.key:
```

Una vez que haya completado el paso anterior, tendrá un certificado firmado para el servidor web (`IISCert.crt`) y un certificado autofirmado (`SelfSignedCA.crt`). Cuando tenga estos archivos, vaya al Paso 3: configurar el servidor web.

Paso 3: configurar el servidor web

Actualice la configuración del sitio web de IIS para que utilice el certificado HTTPS que creó al final del paso anterior. De este modo, concluirá la configuración del software del servidor web de Windows (IIS) para la descarga de SSL/TLS con AWS CloudHSM.

Si utilizó un certificado autofirmado para firmar la CSR, primero debe importar el certificado autofirmado en las entidades de certificación raíz de confianza de Windows.

Para importar el certificado autofirmado en las entidades de certificación raíz de confianza de Windows

1. Si aún no lo ha hecho, conéctese a su servidor de Windows. Para obtener más información, consulte Conectarse a la instancia en la Guía del usuario de Amazon EC2 para instancias de Windows.
2. Copie el certificado autofirmado en el servidor de Windows.
3. En Windows Server, abra el Panel de control.
4. En Buscar en el Panel de control, escriba **`certificates`**. A continuación, elija Administrar certificados de equipo.
5. En la ventana Certificados (equipo local), haga doble clic en Entidades de certificación raíz de confianza.
6. Haga clic con el botón derecho en Certificados y, a continuación, elija Todas las tareas, Importar.
7. En el Asistente para importar certificados, elija Siguiente.
8. Elija Examinar y, a continuación, busque y seleccione el certificado autofirmado. Si creó el certificado autofirmado siguiendo las instrucciones del paso anterior de este tutorial, el certificado autofirmado se llama `SelfSignedCA.crt`. Elija Open.
9. Elija Next.
10. En Almacén de certificados, elija Colocar todos los certificados en el siguiente almacén. A continuación, asegúrese de que está seleccionada la opción Entidades de certificación raíz de confianza para Almacén de certificados.
11. Elija Next y, a continuación, elija Finish.

Para actualizar la configuración del sitio web de IIS

1. Si aún no lo ha hecho, conéctese a su servidor de Windows. Para obtener más información, consulte Conectarse a la instancia en la Guía del usuario de Amazon EC2 para instancias de Windows.
2. AWS CloudHSM Inicie el daemon del cliente.
3. Copie el certificado firmado del servidor web (el que creó al final del paso anterior de este tutorial) en el servidor de Windows.
4. En Windows Server, utilice el comando certreq de Windows para aceptar el certificado firmado, tal y como se muestra en el siguiente ejemplo. Sustituya *IISCert.crt* por el nombre del archivo que contiene el certificado firmado del servidor web.

   ```
   C:\>certreq -accept IISCert.crt
           SDK Version: 2.03
   ```

5. En su servidor de Windows, inicie Administrador del servidor.
6. En el panel Administrador del servidor, en la esquina superior derecha, elija Herramientas, Administrador de Internet Information Services (IIS).
7. En la ventana Administrador de Internet Information Services (IIS), haga doble clic en el nombre del servidor. A continuación, haga doble clic en Sitios. Seleccione el sitio web.
8. Seleccione Configuración de SSL. A continuación, en el lado derecho de la ventana, elija Enlaces.
9. En la ventana Enlaces de sitios, elija Agregar.
10. En Tipo, elija https. En Certificado SSL, elija el certificado HTTPS que creó al final del paso anterior de este tutorial.

    > **Note**
    >
    > Si se produce un error durante la vinculación de este certificado, reinicie el servidor y vuelva a intentar este paso.

11. Seleccione Aceptar.

Después de actualizar la configuración del sitio web, vaya al Paso 4: habilitar el tráfico HTTPS y verificar el certificado.

Paso 4: habilitar el tráfico HTTPS y verificar el certificado

Después de configurar el servidor web para la descarga de SSL/TLS, añada la instancia del servidor web a un grupo de seguridad que AWS CloudHSM permita el tráfico HTTPS entrante. Esto permite a los clientes, como, por ejemplo, navegadores web, establecer una conexión HTTPS con su servidor web. A continuación, establece una conexión HTTPS con tu servidor web y comprueba que utiliza el certificado con el que configuraste la descarga de SSL/TLS. AWS CloudHSM

Temas

- Habilitación de las conexiones HTTPS entrantes
- Verificación del uso del certificado configurado por parte de HTTPS

Habilitación de las conexiones HTTPS entrantes

Para conectarse a su servidor web desde un cliente (como, por ejemplo, un navegador web), cree un grupo de seguridad que permita conexiones HTTPS entrantes. En concreto, debería permitir conexiones TCP entrantes en el puerto 443. Asigne este grupo de seguridad a su servidor web.

Para crear un grupo de seguridad para HTTPS y asignarlo a su servidor web

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Elija Security Groups en el panel de navegación.
3. Elija Crear grupo de seguridad.
4. En Create Security Group (Crear grupo de seguridad), haga lo siguiente:
    a. Para Security group name (Nombre del grupo de seguridad), escriba un nombre para el grupo de seguridad que está creando.
    b. De manera opcional, escriba una descripción del grupo de seguridad que está creando.
    c. Para VPC, elija la VPC que contiene la instancia de su servidor web Amazon EC2.
    d. Seleccione Add Rule (Añadir regla).
    e. Para tipo, seleccione HTTPS en la ventana desplegable.
    f. Para Origen, introduzca una ubicación de origen.
    g. Elija Crear grupo de seguridad.
5. En el panel de navegación, seleccione Instancias.
6. Seleccione la casilla de verificación junto a la instancia del servidor web.

7. Seleccione las Acciones en el menú desplegable que se encuentra en la parte superior de la página. Seleccione Seguridad, a continuación, Cambiar grupos de seguridad.
8. Para Grupos de seguridad asociados, seleccione el cuadro de búsqueda y elija el grupo de seguridad que creó para HTTPS. A continuación, elija Añadir grupos de seguridad.
9. Seleccione Guardar.

Verificación del uso del certificado configurado por parte de HTTPS

Después de añadir el servidor web a un grupo de seguridad, puede verificar que la descarga de SSL/TLS utiliza su certificado autofirmado. Puede hacerlo mediante un navegador web o con una herramienta como OpenSSL s_client.

Para verificar la descarga de SSL/TLS con un navegador web

1. Utilice un navegador web para conectarse a su servidor web mediante el nombre de DNS público o la dirección IP del servidor. Asegúrese de que la dirección URL en la barra de direcciones comienza con https://. Por ejemplo, **`https://ec2-52-14-212-67.us-east-2.compute.amazonaws.com/`**.

   > **Tip**
   >
   > Puede utilizar un servicio DNS como, por ejemplo, Amazon Route 53, para dirigir el nombre de dominio de su sitio web (por ejemplo, https://www.ejemplo.com/) a su servidor web. Para obtener más información, consulte Direccionamiento del tráfico a una instancia de Amazon EC2 en la Guía para desarrolladores de Amazon Route 53 o en la documentación para su servicio DNS.

2. Utilice el navegador web para ver el certificado del servidor web. Para más información, consulte los siguientes temas:

   - Para Mozilla Firefox, consulte View a Certificate en el sitio web de Soporte de Mozilla.
   - Para Google Chrome, consulte Conocer los problemas de seguridad en el sitio web para desarrolladores de Google.

   Otros navegadores web pueden tener características similares que puede utilizar para ver el certificado del servidor web.

3. Asegúrese de que el certificado SSL/TLS es el que ha configurado para que utilice el servidor web.

Para verificar la descarga de SSL/TLS con OpenSSL s_client

1. Ejecute el siguiente comando OpenSSL para conectarse a su servidor web a través de HTTPS. Sustituya *<server name>* por el nombre de DNS público o la dirección IP de su servidor web.

```
openssl s_client -connect <server name>:443
```

> **Tip**
>
> Puede utilizar un servicio DNS como, por ejemplo, Amazon Route 53, para dirigir el nombre de dominio de su sitio web (por ejemplo, https://www.ejemplo.com/) a su servidor web. Para obtener más información, consulte Direccionamiento del tráfico a una instancia de Amazon EC2 en la Guía para desarrolladores de Amazon Route 53 o en la documentación para su servicio DNS.

2. Asegúrese de que el certificado SSL/TLS es el que ha configurado para que utilice el servidor web.

Ahora tiene un sitio web que se protege con HTTPS. La clave privada del servidor web se almacena en un HSM del clúster. AWS CloudHSM

Para agregar un equilibrador de carga, consulte Agregar un equilibrador de carga con Elastic Load Balancing (opcional).

Agregar un equilibrador de carga con Elastic Load Balancing (opcional)

Después de configurar la descarga de SSL/TLS con un servidor web, tiene la opción de crear más servidores web y un equilibrador de carga de Elastic Load Balancing que dirige el tráfico de HTTPS a los servidores web. Un equilibrador de carga puede reducir la carga de sus servidores web individuales equilibrando el tráfico entre dos o más servidores web. También puede aumentar la disponibilidad de su sitio web, ya que el equilibrador de carga monitorea el estado de los servidores web y solo dirige tráfico a servidores en buen estado. Si se produce un error en un servidor web, el equilibrador de carga deja automáticamente de dirigir tráfico hacia el servidor.

Temas

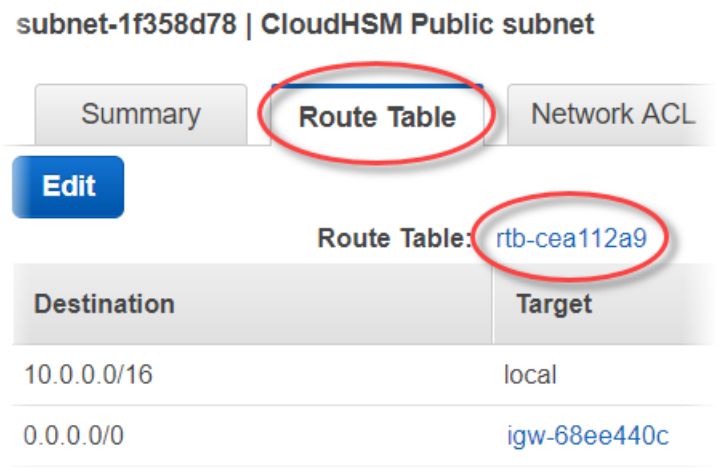
- Creación de una subred para el segundo servidor web
- Creación del segundo servidor web
- Creación del equilibrador de carga

Creación de una subred para el segundo servidor web

Antes de poder crear otro servidor web, debe crear una nueva subred en la misma VPC que contenga el clúster y el servidor AWS CloudHSM web existentes.

Para crear una nueva subred

1. Abra la sección Subredes de la consola de Amazon VPC.
2. Elija Create Subnet (Crear subred).
3. En el cuadro de diálogo Create Subnet, haga lo siguiente:
   a. En Name tag (Etiqueta de nombre), escriba un nombre para la subred.
   b. Para la VPC, elija la AWS CloudHSM VPC que contiene el clúster y el servidor web existentes. AWS CloudHSM
   c. Para Availability Zone (Zona de disponibilidad), elija una zona de disponibilidad diferente a la que contiene su servidor web existente.
   d. En Bloque de CIDR IPv4, escriba el bloque de CIDR que se usará para la subred. Por ejemplo, escriba **`10.0.10.0/24`**.
   e. Elija Sí, crear.
4. Seleccione la casilla de verificación situada junto a la subred pública que contiene su servidor web existente. Se trata de una subred diferente de la subred pública creada en el paso anterior.
5. En el panel de contenido, elija la pestaña Tabla de enrutamiento. A continuación, elija el enlace de la tabla de ruteo.



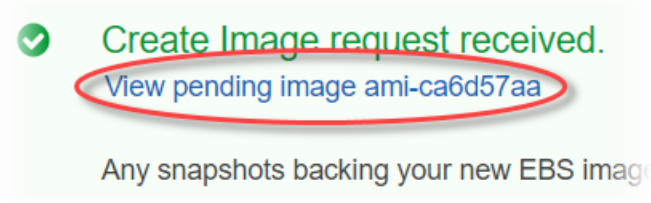
6. Seleccione la casilla de verificación que hay junto a la tabla de ruteo.
7. Elija la pestaña Asociaciones de subredes. A continuación, elija Edit.
8. Seleccione la casilla de verificación situada junto a la subred pública que ha creado anteriormente en este procedimiento. A continuación, elija Guardar.

Creación del segundo servidor web

Complete los siguientes pasos para crear un segundo servidor web con la misma configuración que su servidor web existente.

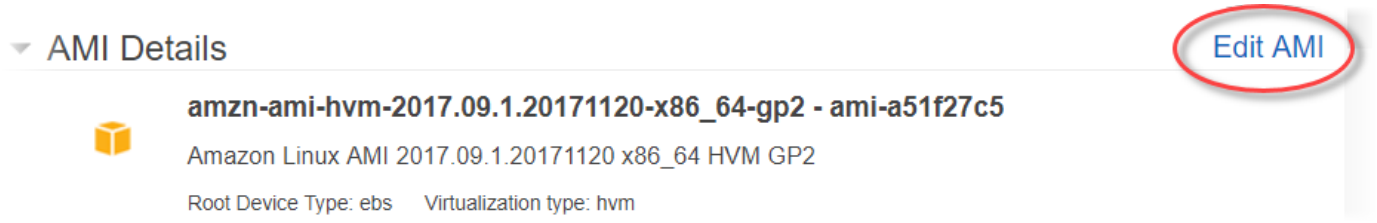
Para crear un segundo servidor web

1. Abra la sección de instancias de la consola de Amazon EC2.
2. Seleccione la casilla de verificación junto a la instancia de servidor web existente.
3. Elija Actions (Acciones), Image (Imagen) y, a continuación, Create Image (Crear imagen).
4. En el cuadro de diálogo Create Image (Crear imagen), haga lo siguiente:
   a. En Image name (Nombre de la imagen), escriba un nombre para la imagen.
   b. En Image description (Descripción de la imagen), escriba una descripción para la imagen.
   c. Elija Create Image (Crear imagen). Esta acción reinicia su servidor web existente.
   d. Elija el enlace Ver imagen pendiente ami-***<ID AMI>***.



En la columna Estado, observe el estado de la imagen. Cuando el estado de la imagen sea disponible (esto podría tardar varios minutos), vaya al siguiente paso.

5. En el panel de navegación, seleccione Instancias.
6. Seleccione la casilla de verificación junto al servidor web existente.
7. Elija Actions (Acciones) y después Launch More Like This (Lanzar más así).
8. Elija Edit AMI (Editar AMI).



9. En el panel de navegación izquierdo, elija Mis AMI. A continuación, borre el texto en el campo de búsqueda.
10. Junto a la imagen del servidor web, elija Select (Seleccionar).
11. Elija Sí, deseo continuar con esta AMI (***nombre de la imagen>*** - ami-***ID de AMI>***).
12. Elija Siguiente.
13. Seleccione un tipo de instancia y, a continuación, elija Next: Configure Instance Details.
14. En Step 3: Configure Instance Details (Paso 3: Configure los detalles de la instancia), haga lo siguiente:

    a. Para Network (Red), elija la VPC que contiene su servidor web existente.
    b. Para Subnet (Subred), elija la subred pública que creó para el segundo servidor web.
    c. En Auto-assign Public IP, elija Enable.
    d. Cambie los detalles restantes de la instancia como desee. A continuación, elija Next: Add Storage (Siguiente: Añadir almacenamiento).
15. Cambie la configuración de almacenamiento como desee. A continuación, elija Siguiente: Agregar etiquetas.
16. Agregue o edite las etiquetas que desee. A continuación, elija Next: Configure Security Group.

17. En Step 6: Configure Security Group, haga lo siguiente:

    a. En Assign a security group (Asignar un grupo de seguridad), seleccione Select an existing security group (Seleccionar un grupo de seguridad existente).

    b. Seleccione la casilla de verificación situada junto al grupo de seguridad denominado cloudhsm-***<cluster ID>***-sg. AWS CloudHSM creó este grupo de seguridad en su nombre al crear el clúster. Debe elegir este grupo de seguridad para permitir que la instancia del servidor web se conecte a los HSM del clúster.

    c. Seleccione la casilla de verificación situada junto al grupo de seguridad que permite tráfico HTTPS entrante. Ha creado previamente este grupo de seguridad.

    d. (Opcional) Seleccione la casilla de verificación situada junto a un grupo de seguridad que permite el tráfico entrante de SSH (para Linux) o de RDP (para Windows) entrante desde la red. Es decir, el grupo de seguridad debe permitir el tráfico entrante de TCP a través del puerto 22 (para SSH en Linux) o del puerto 3389 (para RDP en Windows). De lo contrario, no podrá conectarse a su instancia de cliente. Si no dispone de un grupo de seguridad de este tipo, debe crearlo y, a continuación, asignarlo a la instancia de cliente.

    Elija Revisar e iniciar.

18. Revise los detalles de la instancia y, a continuación, elija Launch.

19. Elija si desea lanzar la instancia con un par de claves existente, crear un nuevo par de claves o lanzar la instancia sin un par de claves.

    - Para utilizar un par de claves existente, haga lo siguiente:

      1. Elija Choose an existing key pair.
      2. En Select a key pair, elija el par de claves que desea usar.
      3. Seleccione la casilla de verificación situada junto a I acknowledge that I have access to the selected private key file (***private key file name***>.pem), and that without this file, I won't be able to log into my instance. (Confirmo que tengo acceso al archivo de clave privada seleccionado (nombre del archivo de clave privada>.pem) y que sin este archivo no podré iniciar sesión en mi instancia).

    - Para crear un nuevo par de claves, haga lo siguiente:

      1. Elija Crear un nuevo par de claves.
      2. En Key pair name (Nombre del par de claves), escriba un nombre para el par de claves.

3. Elija Download Key Pair y guarde el archivo de clave privada en una ubicación segura y accesible.

> **Warning**
>
> No puede volver a descargar el archivo de clave privada después de este punto. Si no descarga ahora el archivo de clave privada, no podrá obtener acceso a la instancia de cliente.

- Para lanzar la instancia sin un par de claves, haga lo siguiente:
  1. Elija Proceed without a key pair (Continuar sin un par de claves).
  2. Seleccione la casilla de verificación junto a I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI. (Confirmo que no podré conectarme a esta instancia salvo que ya sepa la contraseña integrada en esta AMI).

Elija Launch Instances.

Creación del equilibrador de carga

Complete los siguientes pasos para crear un equilibrador de carga de Elastic Load Balancing que dirige el tráfico HTTPS a los servidores web.

Cómo crear un equilibrador de carga

1. Abra la página de equilibradores de carga en la consola de Amazon EC2.
2. Elija Create Load Balancer (Crear equilibrador de carga).
3. En la sección Network Load Balancer, elija Create (Crear).
4. En Paso 1: Configurar el equilibrador de carga, haga lo siguiente:
   a. Para Nombre, escriba un nombre para el equilibrador de carga que está creando.
   b. En la sección Agentes de escucha, para Puerto Equilibrador de Carga, cambie el valor a **443**.
   c. En la sección Availability Zones (Zonas de disponibilidad), para VPC, elija la VPC que contiene sus servidores web.
   d. En la sección Availability Zones (Zonas de disponibilidad), elija las subredes que contienen sus servidores web.

e. Elija Next: Configure Routing (Siguiente: Configuración del enrutamiento).

5. En Step 2: Configure Routing (Paso 2: Configurar direccionamiento), haga lo siguiente:

   a. Para Name (Nombre), escriba un nombre para el grupo de destino que está creando.

   b. Para Puerto, cambie el valor a **443**.

   c. Elija Next: Register Targets (Siguiente: Registrar destinos).

6. Para Step 3: Register Targets (Paso 3: Registrar destinos), haga lo siguiente:

   a. En la sección Instancias, seleccione las casillas de verificación junto a las instancias de servidor web. A continuación, elija Add to registered (Añadir a registrados).

   b. Elija Siguiente: Revisar.

7. Revise los detalles del equilibrador de carga y, a continuación, elija Crear.

8. Cuando se haya creado correctamente el equilibrador de carga, elija Cerrar.

Una vez que haya finalizado los pasos anteriores, la consola de Amazon EC2 muestra su equilibrador de carga de Elastic Load Balancing.

Cuando el estado del equilibrador de carga sea activo, puede verificar que el equilibrador de carga está en ejecución. Es decir, puede verificar que está enviando tráfico HTTPS a sus servidores web con descarga de SSL/TLS con AWS CloudHSM. Puede hacerlo mediante un navegador web o una herramienta como OpenSSL s_client.

Cómo verificar que el equilibrador de carga se está ejecutando con un navegador web

1. En la consola de Amazon EC2, encuentre el nombre de DNS para el equilibrador de carga que acaba de crear. A continuación, seleccione el nombre de DNS y cópielo.

2. Utilice un navegador web como Mozilla Firefox o Google Chrome para conectarse a su equilibrador de carga con el nombre de DNS del equilibrador de carga. Asegúrese de que la dirección URL en la barra de direcciones comienza con https://.

> **Tip**
>
> Puede utilizar un servicio DNS como, por ejemplo, Amazon Route 53, para dirigir el nombre de dominio de su sitio web (por ejemplo, https://www.ejemplo.com/) a su servidor web. Para obtener más información, consulte Direccionamiento del tráfico a una

instancia de Amazon EC2 en la Guía para desarrolladores de Amazon Route 53 o en la documentación para su servicio DNS.

3. Utilice el navegador web para ver el certificado del servidor web. Para más información, consulte los siguientes temas:

   - Para Mozilla Firefox, consulte View a Certificate en el sitio web de Soporte de Mozilla.
   - Para Google Chrome, consulte Conocer los problemas de seguridad en el sitio web para desarrolladores de Google.

   Otros navegadores web pueden tener características similares que puede utilizar para ver el certificado del servidor web.

4. Asegúrese de que el certificado es el que ha configurado para que lo utilice el servidor web.

Cómo verificar que el equilibrador de carga se está ejecutando con OpenSSL s_client

1. Utilice el siguiente comando OpenSSL para conectarse a su equilibrador de carga a través de HTTPS. Sustituya *<DNS name>* por el nombre de DNS de su equilibrador de carga.

```
openssl s_client -connect <DNS name>:443
```

> **Tip**
>
> Puede utilizar un servicio DNS como, por ejemplo, Amazon Route 53, para dirigir el nombre de dominio de su sitio web (por ejemplo, https://www.ejemplo.com/) a su servidor web. Para obtener más información, consulte Direccionamiento del tráfico a una instancia de Amazon EC2 en la Guía para desarrolladores de Amazon Route 53 o en la documentación para su servicio DNS.

2. Asegúrese de que el certificado es el que ha configurado para que lo utilice el servidor web.

Ahora tiene un sitio web protegido con HTTPS, con la clave privada del servidor web almacenada en un HSM de su clúster. AWS CloudHSM Su página web tiene dos servidores web y un equilibrador de carga para ayudar a mejorar la eficiencia y la disponibilidad.

Configuración de Windows Server como entidad de certificación (CA) con AWS CloudHSM

En una infraestructura de clave pública (PKI), una entidad de certificación (CA) es una entidad de confianza que emite certificados digitales. Estos certificados digitales vinculan una clave pública a una identidad (una persona u organización) mediante criptografía de clave pública y firmas digitales. Para operar una CA, debe preservar la confianza mediante la protección de las claves privadas que firman los certificados emitidos por la CA. Puede almacenar las claves privadas en un HSM de su clúster de AWS CloudHSM y usar el HSM para realizar las operaciones de firma criptográfica.

En este tutorial, utilizará Windows Server y AWS CloudHSM configurará una CA. Tiene que instalar el software de cliente de AWS CloudHSM para Windows en el servidor de Windows y agregar después la función Servicios de certificados de Active Directory (AD CS) a Windows Server. Al configurar esta función, se utiliza un proveedor de almacenamiento de AWS CloudHSM claves (KSP) para crear y almacenar la clave privada de la CA en el AWS CloudHSM clúster. El KSP es el puente que conecta el servidor Windows con el clúster AWS CloudHSM . En el último paso, usted firma una solicitud de firma de certificado (CSR) con su entidad de certificación de Windows Server.

Para obtener más información, consulte los temas siguientes:

Temas

- Paso 1 de la entidad de certificación de Windows Server: configurar los requisitos previos
- Paso 2 de la entidad de certificación de Windows Server: crear una entidad de certificación de Windows Server con AWS CloudHSM
- Paso 3 de Windows Server CA: firme una solicitud de firma de certificado (CSR) con su CA de Windows Server con AWS CloudHSM

Paso 1 de la entidad de certificación de Windows Server: configurar los requisitos previos

Para configurar Windows Server como entidad de certificación (CA) con AWS CloudHSM, necesita lo siguiente:

- Un AWS CloudHSM clúster activo con al menos un HSM.
- Una instancia de Amazon EC2 que ejecuta un sistema operativo Windows Server con el software AWS CloudHSM cliente para Windows instalado. Este tutorial usa Microsoft Windows Server 2016.

- Un usuario criptográfico (CU) que sea el propietario y administre la clave privada de la entidad de certificación en el HSM.

Para configurar los requisitos previos de una CA de Windows Server con AWS CloudHSM

1. Realice los pasos que se indican en Introducción. Al lanzar el cliente de Amazon EC2, elija una AMI de Windows Server. Este tutorial usa Microsoft Windows Server 2016. Cuando haya completado estos pasos, dispondrá de un clúster activo con al menos un HSM. También tiene una instancia de cliente Amazon EC2 que ejecuta Windows Server con el software de AWS CloudHSM cliente para Windows instalado.
2. (Opcional) Añada más HSM a su clúster. Para obtener más información, consulte Agregar un HSM.
3. Conéctese a su instancia de cliente. Para obtener más información, consulte Conectarse a la instancia en la Guía del usuario de Amazon EC2 para instancias de Windows.
4. Cree un usuario de criptografía (CU) mediante la administración de usuarios de HSM con la CLI de CloudHSM o administración de usuarios de HSM con la utilidad de administración de CloudHSM (CMU) . Realice un seguimiento del nombre de usuario y la contraseña del CU. Los necesitará para completar el paso siguiente.
5. Establezca las credenciales de inicio de sesión del HSM, utilizando el nombre de usuario y la contraseña del CU que creó en el paso anterior.
6. En el paso 5, si utilizó el Administrador de credenciales de Windows para configurar las credenciales de HSM, descargue `psexec.exe`desde SysInternals y ejecute el siguiente comando como NT Authority\ SYSTEM:

   ```
   psexec.exe -s "C:\Program Files\Amazon\CloudHsm\tools\set_cloudhsm_credentials.exe"
    --username <USERNAME> --password <PASSWORD>
   ```

   Reemplace *<NOMBRE DE USUARIO>* y *<CONTRASEÑA>* por las credenciales de HSM.

Para crear una CA de Windows Server con AWS CloudHSM, vaya aCrear entidad de certificación de Windows Server.

Paso 2 de la entidad de certificación de Windows Server: crear una entidad de certificación de Windows Server con AWS CloudHSM

Para crear una entidad de certificación de Windows Server, usted agrega el rol Servicios de certificados de Active Directory (AD CS) a Windows Server. Al añadir esta función, se utiliza un proveedor de almacenamiento de AWS CloudHSM claves (KSP) para crear y almacenar la clave privada de la CA en el AWS CloudHSM clúster.

> **Note**
>
> Al crear tu entidad de certificación de Windows Server, puedes optar por crear una entidad de certificación o una entidad de certificación subordinada. Normalmente, usted toma esta decisión en función del diseño de su infraestructura de clave pública y las políticas de seguridad de su organización. En este tutorial se explica cómo crear una entidad de certificación raíz para simplificar.

Para agregar el rol AD CS a Windows Server y crear la clave privada de la entidad de certificación

1. Si aún no lo ha hecho, conéctese a su servidor de Windows. Para obtener más información, consulte Conectarse a la instancia en la Guía del usuario de Amazon EC2 para instancias de Windows.
2. En su servidor de Windows, inicie Administrador del servidor.
3. En el panel Administrador del servidor, elija Agregar roles y características.
4. Lea la información de Antes de comenzar y, a continuación, elija Siguiente.
5. En Installation Type, elija Instalación basada en características o en roles. A continuación, elija Next.
6. En Selección de servidor, elija Seleccionar un servidor del grupo de servidores. A continuación, elija Next.
7. En Roles de servidor, haga lo siguiente:
   a. Seleccione Servicios de certificados de Active Directory.
   b. En Agregar características necesarias para Servicios de certificados de Active Directory, elija Agregar características.
   c. Elija Siguiente para finalizar la selección de roles de servidor.
8. En Características, acepte los valores predeterminados y, a continuación, elija Siguiente.

9. En AD CS, haga lo siguiente:

    a. Elija Siguiente.

    b. Seleccione Entidad de certificación y, a continuación, elija Siguiente.

10. En Confirmación, lea la información de confirmación y, a continuación, elija Instalar. No cierre la ventana.
11. Elija el enlace Configurar Servicios de certificados de Active Directory en el servidor de destino resaltado.
12. En Credenciales, compruebe o cambie las credenciales mostradas. A continuación, elija Next.
13. En Servicios de rol, seleccione Entidad de certificación. A continuación, elija Next.
14. En Tipo de instalación, seleccione CA independiente. A continuación, elija Next.
15. En Tipo de CA, seleccione CA raíz. A continuación, elija Next.

    > **Note**
    >
    > Puede optar por crear una entidad de certificación raíz o una entidad de certificación subordinada en función del diseño de su infraestructura de clave pública y las políticas de seguridad de su organización. En este tutorial se explica cómo crear una entidad de certificación raíz para simplificar.

16. En Clave privada, seleccione Crear una nueva clave privada. A continuación, elija Next.
17. En Criptografía, haga lo siguiente:

    a. En Seleccionar un proveedor de servicios criptográficos, seleccione una de las opciones del proveedor de almacenamiento de claves de Cavium en el menú. Estos son los proveedores de almacenamiento de claves de AWS CloudHSM . Por ejemplo, puede elegir el proveedor de almacenamiento de claves de RSA#Cavium.

    b. En Longitud de la clave, elija una de las opciones de la longitud de la clave.

    c. En Seleccione el algoritmo de hash para firmar los certificados emitidos por esta CA, elija una de las opciones del algoritmo de hash.

    Elija Siguiente.

18. En Nombre de CA, haga lo siguiente:

    a. (Opcional) Edite el nombre común.

b. (Opcional) Escriba un sufijo de nombre distinguido.

Elija Siguiente.

19. En Periodo de validez, especifique un periodo en años, meses, semanas o días. A continuación, elija Next.
20. En Base de datos de certificados, puede aceptar los valores predeterminados o, de forma opcional, cambiar la ubicación y el registro de la base de datos. A continuación, elija Next.
21. En Confirmación, consulte la información acerca de su entidad de certificación; a continuación, elija Configurar.
22. Elija Cerrar y, a continuación, seleccione Cerrar de nuevo.

Ahora tiene una CA de Windows Server con AWS CloudHSM. Para aprender a firmar una solicitud de firma de certificado (CSR) con su entidad de certificación, vaya a Firmar una CSR.

Paso 3 de Windows Server CA: firme una solicitud de firma de certificado (CSR) con su CA de Windows Server con AWS CloudHSM

Puede usar su CA de Windows Server AWS CloudHSM para firmar una solicitud de firma de certificado (CSR). Para completar estos pasos, necesita una CSR válida. Puede crear una CSR de varias formas, incluidas las siguientes:

- Mediante OpenSSL
- Mediante el Administrador de Internet Information Services (IIS) de Windows Server
- Mediante el complemento de certificados en la consola de administración de Microsoft
- Mediante la utilidad de la línea de comandos certreq en Windows

Los pasos para crear una CSR quedan fuera del alcance de este tutorial. Al tener una CSR, puede firmarla con su entidad de certificación de Windows Server.

Para firmar una CSR con su entidad de certificación de Windows Server

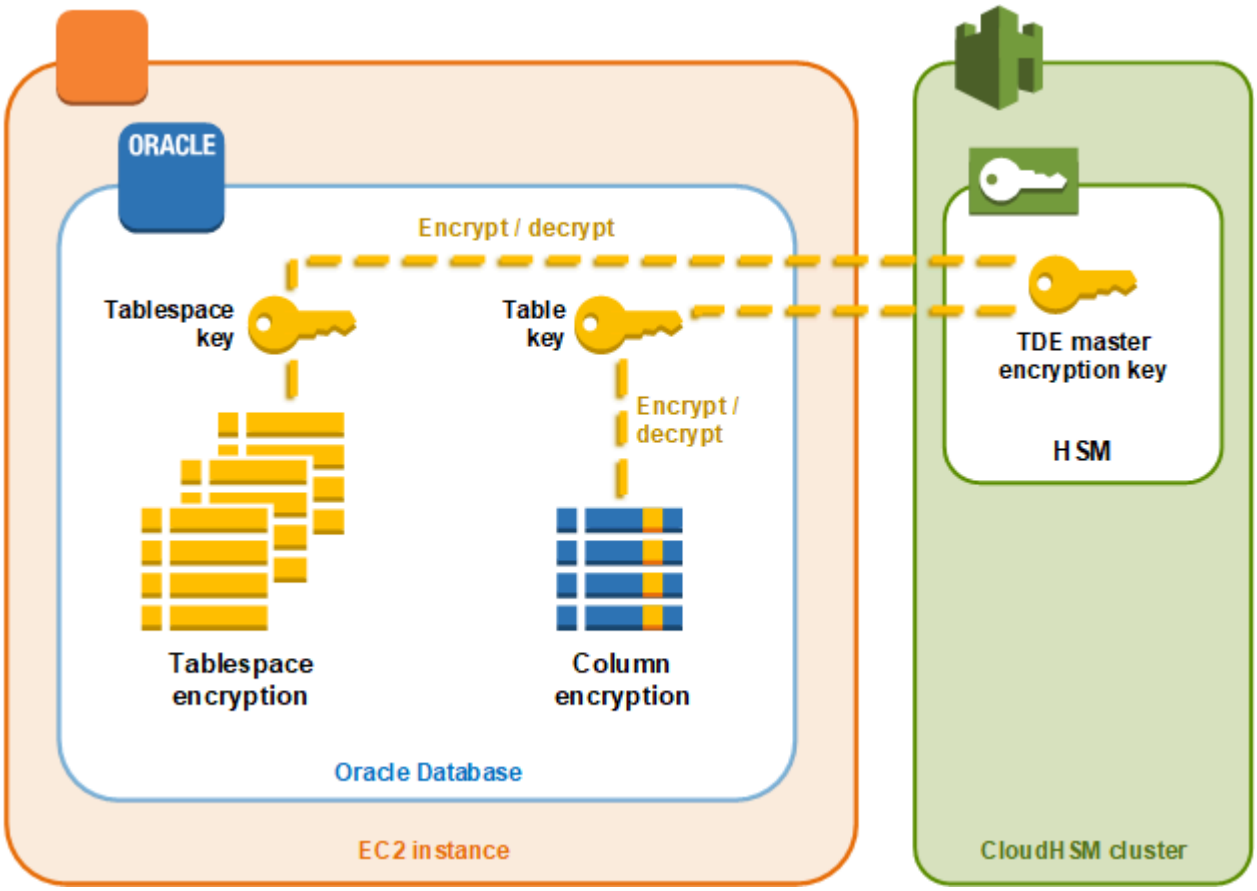
1. Si aún no lo ha hecho, conéctese a su servidor de Windows. Para obtener más información, consulte Conectarse a la instancia en la Guía del usuario de Amazon EC2 para instancias de Windows.
2. En su servidor de Windows, inicie Administrador del servidor.

3. En el panel Administrador del servidor, en la esquina superior derecha, elija Herramientas, Entidad de certificación.
4. En la ventana Entidad de certificación, elija el nombre de su equipo.
5. En el menú Acción, elija Todas las tareas, Enviar nueva solicitud.
6. Seleccione el archivo de la CSR y, a continuación, elija Abrir.
7. En la ventana Entidad de certificación, haga doble clic en Solicitudes pendientes.
8. Seleccione la solicitud pendiente. A continuación, en el menú Acción, elija Todas las tareas, Emitir.
9. En la ventana entidad de certificación, haga doble clic en Solicitudes emitidas para ver el certificado firmado.
10. (Opcional) Para exportar el certificado firmado a un archivo, complete los pasos siguientes:
    a. En la ventana Entidad de certificación, haga doble clic en el certificado.
    b. Elija la pestaña Detalles y, a continuación, elija Copiar en archivo.
    c. Siga las instrucciones en el Asistente para exportar certificados.

Ahora tiene una CA de Windows Server y un certificado válido firmado por la CA de Windows Server. AWS CloudHSM

Cifrado de datos transparente (TDE) de Oracle Database con AWS CloudHSM

El cifrado de datos transparente (TDE) se usa para cifrar archivos de bases de datos. Con TDE, el software de base de datos cifra los datos antes de almacenarlos en el disco. Los datos de las columnas de la tabla o de los espacios de tabla de la base de datos se cifran con una clave de tabla o de espacio de tabla. Algunas versiones del software de bases de datos de Oracle ofrecen TDE. En Oracle TDE, estas claves se cifran con la clave de cifrado maestra de TDE. Puede lograr una mayor seguridad almacenando la clave de cifrado maestra de TDE en los HSM de su clúster. AWS CloudHSM



En esta solución, su Oracle Database está instalado en una instancia Amazon EC2. Oracle Database se integra con la biblioteca de software de AWS CloudHSM para PKCS # 11 para almacenar la clave maestra de TDE en los HSM de su clúster.

> **Important**
>
> - Recomendamos instalar Oracle Database en una instancia de Amazon EC2.

Siga los pasos que se describen a continuación para realizar la integración del cifrado TDE de Oracle en AWS CloudHSM.

Para configurar la integración de Oracle TDE con AWS CloudHSM

1. Siga los pasos que se detallan en Configuración de requisitos previos para preparar el entorno.
2. Siga los pasos que se indican a continuación Configuración de la base de datos para configurar Oracle Database para que se integre con su AWS CloudHSM clúster.

Oracle TDE con AWS CloudHSM: configurar los requisitos previos

Para lograr la integración de Oracle TDE con AWS CloudHSM, necesita lo siguiente:

- Un AWS CloudHSM clúster activo con al menos un HSM.
- Una instancia de Amazon EC2 que ejecute el sistema operativo Linux y tenga el siguiente software instalado:
  - El AWS CloudHSM cliente y las herramientas de línea de comandos.
  - La biblioteca AWS CloudHSM de software para PKCS #11.
  - Base de datos Oracle. AWS CloudHSM admite la integración de Oracle TDE. La versión del SDK 5.6 de cliente y las versiones posteriores admiten el cifrado TDE de Oracle para la versión 19c de Oracle Database. El SDK 3 de cliente es compatible con el cifrado TDE de Oracle para las versiones 11g y 12c de Oracle Database.
- Un usuario criptográfico (CU) que sea el propietario de la clave de cifrado maestra del cifrado TDE y la administre en los HSM de su clúster.

Complete los siguientes pasos para configurar todos los requisitos previos.

Para configurar los requisitos previos para la integración de Oracle TDE con AWS CloudHSM

1. Realice los pasos que se indican en Introducción. A continuación, tendrá un clúster activo con un HSM. También tendrá una instancia de Amazon EC2 que se ejecuta en el sistema operativo de Amazon Linux. El AWS CloudHSM cliente y las herramientas de línea de comandos también se instalarán y configurarán.
2. (Opcional) Añada más HSM a su clúster. Para obtener más información, consulte Agregar un HSM.
3. Conéctese a la instancia de cliente de Amazon EC2 y haga lo siguiente:
   a. Instale la biblioteca AWS CloudHSM de software para PKCS #11.
   b. Instale Oracle Database. Para obtener más información, consulte la documentación de Oracle Database. La versión del SDK 5.6 de cliente y las versiones posteriores admiten el cifrado TDE de Oracle para la versión 19c de Oracle Database. El SDK 3 de cliente es compatible con el cifrado TDE de Oracle para las versiones 11g y 12c de Oracle Database.
   c. Utilice la herramienta de línea de comandos cloudhsm_mgmt_util para crear un usuario criptográfico (CU) en su clúster. Para obtener más información sobre la creación de un

CU, consulte Cómo administrar a los usuarios de HSM con una CMU y Administración de usuarios de HSM.

Una vez que haya completado estos pasos, podrá Configuración de la base de datos.

Oracle TDE con AWS CloudHSM: Configure la base de datos y genere la clave de cifrado maestra

Para integrar Oracle TDE con su AWS CloudHSM clúster, consulte los siguientes temas:

1. Actualización de la configuración de la base de datos de Oracle para utilizar los HSM en su clúster como el módulo de seguridad externa. Para obtener información acerca de módulos de seguridad externos, consulte Introduction to Transparent Data Encryption en la Oracle Database Advanced Security Guide.
2. Generación de claves de cifrado maestras de TDE de Oracle en los HSM en su clúster.

Actualización de la configuración de la base de datos de Oracle

Para actualizar la configuración de Oracle Database para utilizar un HSM en su clúster como el módulo de seguridad externo, complete los siguientes pasos. Para obtener información acerca de módulos de seguridad externos, consulte Introduction to Transparent Data Encryption en la Oracle Database Advanced Security Guide.

Para actualizar la configuración de Oracle

1. Conéctese a su instancia de cliente de Amazon EC2. Se trata de la instancia donde instaló Oracle Database.
2. Realice una copia de backup del archivo denominado `sqlnet.ora`. Para la ubicación de este archivo, consulte la documentación de Oracle.
3. Utilice un editor de texto para editar el archivo denominado `sqlnet.ora`. Añada la siguiente línea. Si una línea existente en el archivo comienza con `encryption_wallet_location`, sustituya la línea existente por la siguiente.

   ```
   encryption_wallet_location=(source=(method=hsm))
   ```

   Guarde el archivo.

4. Ejecute el siguiente comando para crear el directorio en el que Oracle Database espera encontrar el archivo de biblioteca de la biblioteca de software AWS CloudHSM PKCS #11.

```
sudo mkdir -p /opt/oracle/extapi/64/hsm
```

5. Ejecute el siguiente comando para copiar la biblioteca de AWS CloudHSM software del archivo PKCS #11 en el directorio que creó en el paso anterior.

```
sudo cp /opt/cloudhsm/lib/libcloudhsm_pkcs11.so /opt/oracle/extapi/64/hsm/
```

> **Note**
>
> El directorio `/opt/oracle/extapi/64/hsm` solo debe contener un archivo de biblioteca. Elimine cualquier otro archivo que exista en ese directorio.

6. Ejecute el siguiente comando para cambiar la propiedad del directorio `/opt/oracle` y todo lo que hay dentro de él.

```
sudo chown -R oracle:dba /opt/oracle
```

7. Inicie Oracle Database.

Generación de claves de cifrado maestras de TDE de Oracle

Para generar la clave maestra de TDE de Oracle en los HSM en su clúster, complete los pasos que se indican en el siguiente procedimiento.

Para generar la clave maestra

1. Utilice el siguiente comando para abrir Oracle SQL*Plus. Cuando se le solicite, escriba la contraseña del sistema que configuró cuando instaló Oracle Database.

```
sqlplus / as sysdba
```

> **Note**
>
> Para SDK 3 de cliente, debe establecer la variable de entorno `CLOUDHSM_IGNORE_CKA_MODIFIABLE_FALSE` cada vez que genere una clave

maestra. Esta variable solo es necesaria para generar claves maestras. Para obtener más información, consulte "Problemas: Oracle establece el atributo `CKA_MODIFIABLE` de PCKS #11 durante la generación de claves maestras, pero HSM no admite esto" en Problemas conocidos para integrar aplicaciones de terceros.

2. Ejecute la instrucción SQL que crea la clave de cifrado maestra tal y como se muestra en los siguientes ejemplos. Utilice la declaración que se corresponde con la versión de Oracle Database. Reemplace *<nombre de usuario CU>* por el nombre del usuario criptográfico (CU). Reemplace *<password>* por la contraseña del CU.

> **Important**
>
> Ejecute el siguiente comando solo una vez. Cada vez que se ejecuta el comando, se crea una nueva clave de cifrado maestra.

- Para Oracle Database versión 11, ejecute la siguiente instrucción SQL.

```
SQL> alter system set encryption key identified by "<CU user name>:<password>";
```

- Para Oracle Database versión 12 y versión 19c, ejecute la siguiente instrucción SQL.

```
SQL> administer key management set key identified by "<CU user
 name>:<password>";
```

Si la respuesta es `System altered` o `keystore altered`, entonces ha generado correctamente y establecido la clave maestra para Oracle TDE.

3. (Opcional) Ejecute el siguiente comando para comprobar el estado del monedero de Oracle.

```
SQL> select * from v$encryption_wallet;
```

Si el wallet no está abierto, utilice uno de los siguientes comandos para abrirlo. Reemplace *<nombre de usuario CU>* por el nombre del usuario criptográfico (CU). Reemplace *<password>* por la contraseña del CU.

- Para Oracle 11, ejecute el siguiente comando para abrir el wallet.

```
SQL> alter system set encryption wallet open identified by "<CU user
 name>:<password>";
```

Para cerrar manualmente el wallet, ejecute el siguiente comando.

```
SQL> alter system set encryption wallet close identified by "<CU user
 name>:<password>";
```

- Para Oracle 12 y Oracle 19c, ejecute el siguiente comando para abrir el wallet.

```
SQL> administer key management set keystore open identified by "<CU user
 name>:<password>";
```

Para cerrar manualmente el wallet, ejecute el siguiente comando.

```
SQL> administer key management set keystore close identified by "<CU user
 name>:<password>";
```

Usa Microsoft SignTool con AWS CloudHSM para firmar archivos

En criptografía y en la infraestructura de claves públicas (PKI), se utilizan firmas digitales para confirmar que los datos los ha enviado una entidad de confianza. Las firmas también indican que los datos no se han manipulado durante su transmisión. Una firma es un hash cifrado que se genera con la clave privada del remitente. El receptor puede verificar la integridad de los datos descifrando su firma de hash con la clave pública del remitente. A su vez, el remitente tiene la responsabilidad de mantener un certificado digital. El certificado digital demuestra que el remitente es el propietario de la clave privada y proporciona al destinatario la clave pública necesaria para realizar el descifrado. Siempre que la clave privada sea propiedad del remitente, se puede confiar en la firma. AWS CloudHSM proporciona hardware seguro validado por el FIPS 140-2 de nivel 3 para proteger estas claves con un acceso exclusivo de un solo inquilino.

Muchas organizaciones utilizan Microsoft SignTool, una herramienta de línea de comandos que firma, verifica y marca la hora de los archivos para simplificar el proceso de firma de código. Puede utilizarlos AWS CloudHSM para almacenar de forma segura sus pares de claves hasta que los necesite SignTool, creando así un flujo de trabajo fácilmente automatizable para la firma de datos.

Los siguientes temas proporcionan una descripción general de cómo utilizarlos SignTool con AWS CloudHSM:

Temas

- Microsoft SignTool con AWS CloudHSM el paso 1: configurar los requisitos previos
- Microsoft SignTool con el AWS CloudHSM paso 2: crea un certificado de firma
- Microsoft SignTool con AWS CloudHSM el paso 3: firmar un archivo

Microsoft SignTool con AWS CloudHSM el paso 1: configurar los requisitos previos

Para usar Microsoft SignTool con AWS CloudHSM, necesitas lo siguiente:

- Una instancia del cliente Amazon EC2 con el sistema operativo Windows.
- Una autoridad de certificación (CA) cuyo mantenimiento se realiza internamente o a través de un proveedor externo.
- Un AWS CloudHSM clúster activo en la misma nube pública virtual (VPC) que su instancia EC2. El clúster debe contener al menos un HSM.
- Un usuario criptográfico (CU) para poseer y administrar las claves del AWS CloudHSM clúster.
- Un archivo o un archivo ejecutable sin firma.
- El kit de desarrollo de software (SDK) de Microsoft Windows.

Para configurar los requisitos previos para su uso AWS CloudHSM con Windows SignTool

1. Siga las instrucciones de la sección Introducción de esta guía para lanzar una instancia EC2 de Windows y un clúster de AWS CloudHSM .
2. Si desea alojar su propia CA de Windows Server, siga los pasos 1 y 2 de Configuración de Windows Server como entidad emisora de certificados con AWS CloudHSM. De lo contrario, siga utilizando su CA de terceros de confianza pública.
3. Descargue e instale una de las siguientes versiones de SDK de Microsoft Windows en la instancia EC2 de Windows:

    - Microsoft Windows SDK 10
    - Microsoft Windows SDK 8.1
    - Microsoft Windows SDK 7

El archivo ejecutable de `SignTool` forma parte de la característica de instalación Windows SDK Signing Tools for Desktop Apps. Puede omitir la instalación de las demás funciones si no las necesita. La ubicación de instalación predeterminada es:

```
C:\Program Files (x86)\Windows Kits\<SDK version>\bin\<version number>\<CPU
 architecture>\signtool.exe
```

Ahora puede usar el SDK de Microsoft Windows, su AWS CloudHSM clúster y su CA para crear un certificado de firma.

Microsoft SignTool con el AWS CloudHSM paso 2: crea un certificado de firma

Ahora que ha descargado el SDK de Windows en la instancia EC2, puede utilizarlo para generar una solicitud de firma de certificado (CSR). La CSR es un certificado sin firma que se puede pasar a la CA para que lo firme. En este ejemplo, se utiliza el archivo ejecutable `certreq` que se incluye con el SDK de Windows para generar la CSR.

Para generar una CSR mediante el archivo ejecutable **`certreq`**

1. Si todavía no lo ha hecho, conéctese a la instancia EC2 de Windows. Para obtener más información, consulte Conectarse a la instancia en la Guía del usuario de Amazon EC2 para instancias de Windows.
2. Cree un archivo denominado `request.inf` que contenga las líneas siguientes. Sustituya la información de `Subject` por la de su organización. Para ver una explicación de cada uno de los parámetros, consulte la documentación de Microsoft.

```
[Version]
Signature= $Windows NT$
[NewRequest]
Subject = "C=<Country>,CN=<www.website.com>,O=<Organization>,OU=<Organizational-
Unit>,L=<City>,S=<State>"
RequestType=PKCS10
HashAlgorithm = SHA256
KeyAlgorithm = RSA
KeyLength = 2048
ProviderName = Cavium Key Storage Provider
```

```
KeyUsage = "CERT_DIGITAL_SIGNATURE_KEY_USAGE"
MachineKeySet = True
Exportable = False
```

3. Ejecute `certreq.exe`. En este ejemplo, se guarda la CSR como `request.csr`.

```
certreq.exe -new request.inf request.csr
```

Internamente, se genera un nuevo par de claves en el AWS CloudHSM clúster y la clave privada del par se utiliza para crear la CSR.

4. Envíe la CSR a su CA. Si utiliza una CA de Windows Server, siga estos pasos:

   a. Escriba el siguiente comando para abrir la herramienta de CA:

   ```
   certsrv.msc
   ```

   b. En la ventana nueva, haga clic con el botón derecho del ratón en el nombre del servidor de la CA. Elija Todas las tareas y, a continuación, elija Enviar solicitud nueva.

   c. Vaya a la ubicación de `request.csr`y elija Abrir.

   d. Navegue hasta la carpeta Solicitudes pendientes expandiendo el menú Entidad de certificación de Server. Haga clic con el botón derecho del ratón en la solicitud que acaba de crear y, en Todas las tareas, elija Emitir.

   e. Ahora vaya a la carpeta Certificados emitidos (situada encima de la carpeta Solicitudes pendientes).

   f. Elija Abrir para ver el certificado y, a continuación, elija la pestaña Detalles.

   g. Elija Copiar en archivo para iniciar el Asistente para exportación de certificados. Guarde en un lugar seguro el archivo X.509 con codificación DER como `signedCertificate.cer`.

   h. Salga de la herramienta de CA y utilice el siguiente comando para mover el archivo de certificado al almacén de certificados personales de Windows. Esto permite que puedan utilizarlo otras aplicaciones.

   ```
   certreq.exe -accept signedCertificate.cer
   ```

A partir de ahora, puede utilizar el certificado importado para Firmar un archivo .

Microsoft SignTool con AWS CloudHSM el paso 3: firmar un archivo

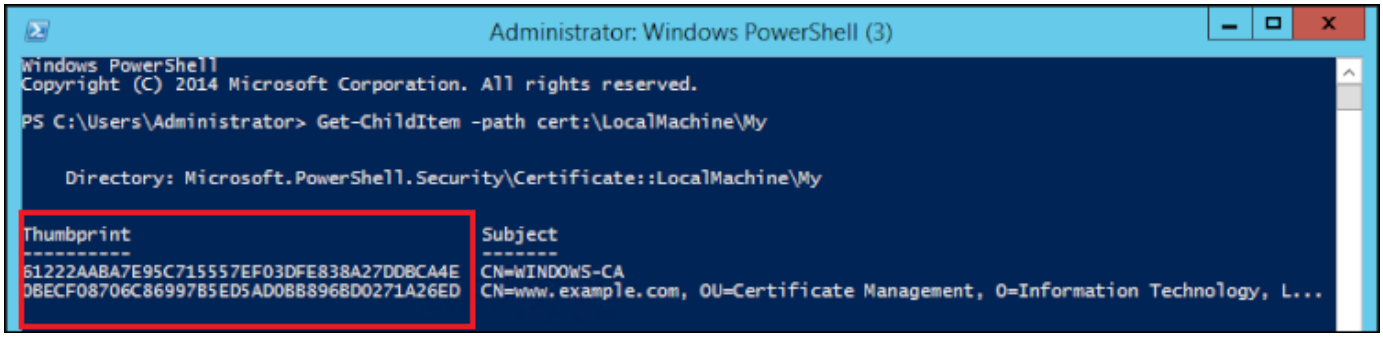
Ya está listo para usar SignTool su certificado importado para firmar su archivo de ejemplo. Para ello, debe conocer el hash SHA-1 o huella digital del certificado. La huella digital se utiliza para garantizar que SignTool solo se utilicen certificados verificados por AWS CloudHSM. En este ejemplo, se utiliza PowerShell para obtener el hash del certificado. También puede utilizar la GUI de la CA o el archivo ejecutable `certutil` del SDK de Windows.

Para obtener la huella digital de un certificado y utilizarla para firmar un archivo

1. PowerShell Ábrelo como administrador y ejecuta el siguiente comando:

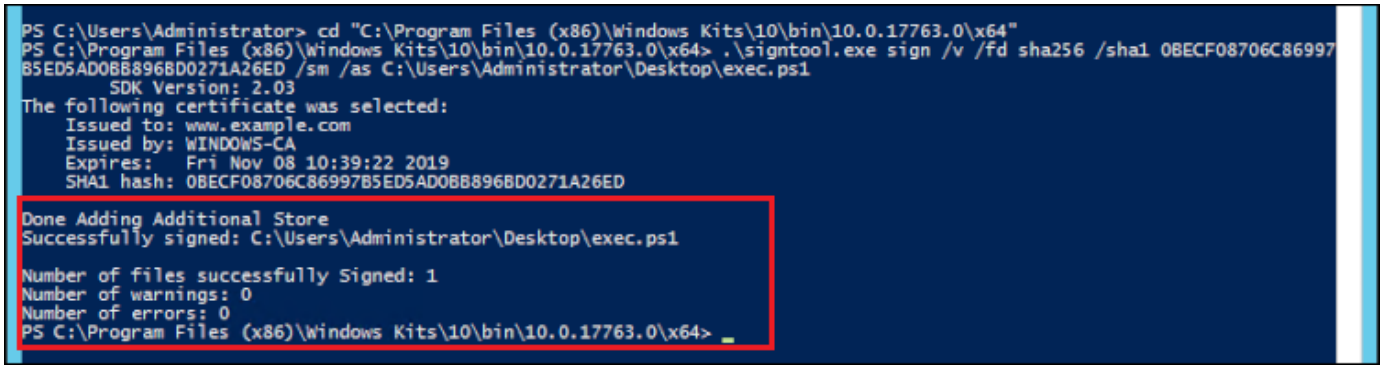
   ```
   Get-ChildItem -path cert:\LocalMachine\My
   ```

   Copie la huella digital que se visualiza debajo de `Thumbprint`.

2. Navegue hasta el directorio PowerShell que contiene`SignTool.exe`. La ubicación predeterminada es `C:\Program Files (x86)\Windows Kits\10\bin\10.0.17763.0\x64`.
3. Por último, firme el archivo ejecutando el comando siguiente. Si el comando se ejecuta correctamente, PowerShell devuelve un mensaje de confirmación.

   ```
   signtool.exe sign /v /fd sha256 /sha1 <thumbprint> /sm C:\Users\Administrator
   \Desktop\<test>.ps1
   ```



4. (Opcional) Para verificar la firma del archivo, utilice el comando siguiente:

```
signtool.exe verify /v /pa C:\Users\Administrator\Desktop\<test>.ps1
```

Java Keytool y Jarsigner

AWS CloudHSM ofrece la integración con las utilidades Java Keytool y Jarsigner mediante Client SDK 3 y Client SDK 5. Los pasos para usar estas herramientas variarán en función de la versión del SDK de cliente en la que haya descargado:

- Uso de SDK 5 de cliente para la integración con Java Keytool y Jarsigner
- Uso de SDK 3 de cliente para la integración con Java Keytool y Jarsigner

Uso de SDK 5 de cliente para la integración con Java Keytool y Jarsigner

AWS CloudHSM el almacén de claves es un almacén de claves JCE de uso especial que utiliza los certificados asociados a las claves de su HSM a través de herramientas de terceros, como y. `keytool` `jarsigner` AWS CloudHSM no almacena los certificados en el HSM, ya que los certificados son datos públicos y no confidenciales. El almacén de AWS CloudHSM claves almacena los certificados en un archivo local y los asigna a las claves correspondientes del HSM.

Cuando se utiliza el almacén de AWS CloudHSM claves para generar nuevas claves, no se genera ninguna entrada en el archivo del almacén de claves local; las claves se crean en el HSM. Del mismo modo, cuando utiliza el almacén de claves de AWS CloudHSM para buscar claves, la búsqueda se transfiere al HSM. Al almacenar los certificados en el almacén de AWS CloudHSM claves, el proveedor comprueba que existe un par de claves con el alias correspondiente en el HSM y, a continuación, asocia el certificado proporcionado al par de claves correspondiente.

Temas

- Requisitos previos
- Uso del almacén de AWS CloudHSM claves con keytool
- Uso del almacén de AWS CloudHSM claves con Jarsigner
- Problemas conocidos

Requisitos previos

Para usar el almacén de AWS CloudHSM claves, primero debe inicializar y configurar el AWS CloudHSM SDK de JCE.

Paso 1: Instalar JCE

Para instalar el JCE, incluidos los requisitos previos del AWS CloudHSM cliente, siga los pasos para instalar la biblioteca Java.

Paso 2: Agregar credenciales de inicio de sesión de HSM a variables de entorno

Configure las variables de entorno para que contengan las credenciales de inicio de sesión de HSM.

Linux

```
$ export HSM_USER=<HSM user name>
```

```
$ export HSM_PASSWORD=<HSM password>
```

Windows

```
PS C:\> $Env:HSM_USER=<HSM user name>
```

```
PS C:\> $Env:HSM_PASSWORD=<HSM password>
```

> **Note**
>
> El AWS CloudHSM JCE ofrece varias opciones de inicio de sesión. Para utilizar el almacén de AWS CloudHSM claves con aplicaciones de terceros, debe utilizar el inicio de sesión implícito con variables de entorno. Si desea utilizar el inicio de sesión explícito a través del

código de la aplicación, debe crear su propia aplicación con el almacén de AWS CloudHSM claves. Para obtener información adicional, consulte el artículo sobre el uso del almacén de AWS CloudHSM claves.

Paso 3: Registrar el proveedor de JCE

Para registrar el proveedor de JCE en la CloudProvider configuración de Java, siga estos pasos:

1. Abra el archivo de configuración `java.security` en su instalación de Java para editarlo.
2. En el archivo de configuración `java.security`, agregue `com.amazonaws.cloudhsm.jce.provider.CloudHsmProvider` como último proveedor. Por ejemplo, si hay nueve proveedores en el archivo `java.security`, agregue el siguiente proveedor como último proveedor de la sección:

   ```
   security.provider.10=com.amazonaws.cloudhsm.jce.provider.CloudHsmProvider
   ```

> **Note**
>
> Añadir al AWS CloudHSM proveedor como prioridad más alta puede afectar negativamente al rendimiento del sistema, ya que se dará prioridad al AWS CloudHSM proveedor en las operaciones que puedan transferirse de forma segura al software. Como práctica recomendada, especifique siempre el proveedor que desea utilizar para una operación, ya sea el proveedor AWS CloudHSM o un proveedor basado en software.

> **Note**
>
> Especificar las opciones de línea de comandos `-providerName`, `-providerclass` y `-providerpath` al generar claves mediante keytool con el almacén de claves de AWS CloudHSM puede provocar errores.

Uso del almacén de AWS CloudHSM claves con keytool

Keytool es una conocida utilidad de línea de comandos para tareas comunes de claves y certificados. La documentación de AWS CloudHSM no ofrece un tutorial completo sobre Keytool. En este artículo

se explican los parámetros específicos que debe utilizar con varias funciones de la herramienta clave cuando se utiliza AWS CloudHSM como raíz de confianza a través del almacén de AWS CloudHSM claves.

Cuando utilice keytool con el almacén de AWS CloudHSM claves, especifique los siguientes argumentos para cualquier comando de keytool:

Linux

```
-storetype CLOUDHSM -J-classpath< '-J/opt/cloudhsm/java/*'>
```

Windows

```
-storetype CLOUDHSM -J-classpath<'-J"C:\Program Files\Amazon\CloudHSM\java\*"'>
```

Si desea crear un nuevo archivo de almacén de claves mediante el almacén de AWS CloudHSM claves, consulte. Usando AWS CloudHSM KeyStore Si desea utilizar un almacén de claves existente, especifique el nombre (incluida la ruta) con el argumento keystore en keytool. Si especifica un archivo de almacén de claves que no existe en un comando de keytool, el almacén de AWS CloudHSM claves crea un nuevo archivo de almacén de claves.

Creación de nuevas claves con keytool

Puede usar keytool para generar cualquier tipo de claves RSA, AES y DESede admitidas por el SDK para JCE de AWS CloudHSM.

> **Important**
>
> Una clave generada mediante keytool se genera en el software y, a continuación, se importa AWS CloudHSM como una clave persistente y extraíble.

Le recomendamos encarecidamente que genere las claves no exportables fuera de keytool y que después importe los certificados correspondientes en el almacén de claves. Si utilizas claves RSA o EC extraíbles a través de keytool y Jarsigner, los proveedores exportan las claves desde AWS CloudHSM y luego las utilizan localmente para las operaciones de firma.

Si tiene varias instancias de cliente conectadas a su AWS CloudHSM clúster, tenga en cuenta que al importar un certificado al almacén de claves de una instancia de cliente, los certificados no estarán

disponibles automáticamente en otras instancias de cliente. Para registrar la clave y los certificados asociados en cada instancia del cliente, debe ejecutar una aplicación Java, tal y como se describe en the section called “Generación de CSR con keytool”. Si lo desea, también puede realizar los cambios necesarios en un cliente y copiar el archivo de almacén de claves resultante en las demás instancias del cliente.

Ejemplo 1: generar una clave AES-256 simétrica y guardarla en un archivo de almacén de claves llamado «my_keystore.store» del directorio de trabajo. Reemplace *<secret label>* por una etiqueta única.

Linux

```
$ keytool -genseckey -alias <secret label> -keyalg aes \
 -keysize 256 -keystore my_keystore.store \
 -storetype CloudHSM -J-classpath '-J/opt/cloudhsm/java/*' \
```

Windows

```
PS C:\> keytool -genseckey -alias <secret label> -keyalg aes `
 -keysize 256 -keystore my_keystore.store `
 -storetype CloudHSM -J-classpath '-J"C:\Program Files\Amazon\CloudHSM\java\*"'
```

Ejemplo 2: generar un par de claves RSA 2048 y guardarlo en un archivo de almacén de claves llamado “my_keystore.store” en el directorio de trabajo. Reemplace *<RSA key pair label>* por una etiqueta única.

Linux

```
$ keytool -genkeypair -alias <RSA key pair label> \
 -keyalg rsa -keysize 2048 \
 -sigalg sha512withrsa \
 -keystore my_keystore.store \
 -storetype CLOUDHSM \
 -J-classpath '-J/opt/cloudhsm/java/*'
```

Windows

```
PS C:\> keytool -genkeypair -alias <RSA key pair label> `
 -keyalg rsa -keysize 2048 `
 -sigalg sha512withrsa `
```

```
 -keystore my_keystore.store `
 -storetype CLOUDHSM `
 -J-classpath '-J"C:\Program Files\Amazon\CloudHSM\java\*"'
```

Encontrará una lista de los algoritmos de firma compatibles en la biblioteca de Java.

Eliminación de claves con keytool

El almacén de AWS CloudHSM claves no admite la eliminación de claves. Puede borrar las claves mediante el método de eliminación de la interfaz Destroyable.

```
((Destroyable) key).destroy();
```

Generación de CSR con keytool

Para tener la máxima flexibilidad al generar una solicitud de firma de certificado (CSR), utilice Motor dinámico de OpenSSL. El comando siguiente utiliza keytool para generar una CSR de un par de claves con el alias `my-key-pair`.

Linux

```
$ keytool -certreq -alias <key pair label> \
 -file my_csr.csr \
 -keystore my_keystore.store \
 -storetype CLOUDHSM \
 -J-classpath '-J/opt/cloudhsm/java/*'
```

Windows

```
PS C:\> keytool -certreq -alias <key pair label> `
 -file my_csr.csr `
 -keystore my_keystore.store `
 -storetype CLOUDHSM `
 -J-classpath '-J"C:\Program Files\Amazon\CloudHSM\java\*"'
```

> **Note**
>
> Para poder utilizar un par de claves de keytool, ese par de claves debe tener una entrada en el archivo de almacén de claves especificado. Si desea utilizar un par de claves generado

fuera de keytool, debe importar los metadatos de las claves y los certificados en el almacén de claves. Para obtener instrucciones sobre la importación de los datos del almacén de claves, consulte the section called “Uso de keytool para importar certificados intermedios y raíz al almacén de AWS CloudHSM claves ”.

Uso de keytool para importar certificados intermedios y raíz al almacén de AWS CloudHSM claves

Para importar un certificado de CA, debe habilitar la verificación de una cadena de certificados completa en un certificado recién importado. A continuación, se muestra un ejemplo del comando:

Linux

```
$ keytool -import -trustcacerts -alias rootCAcert \
 -file rootCAcert.cert -keystore my_keystore.store \
 -storetype CLOUDHSM \
 -J-classpath '-J/opt/cloudhsm/java/*'
```

Windows

```
PS C:\> keytool -import -trustcacerts -alias rootCAcert `
 -file rootCAcert.cert -keystore my_keystore.store `
 -storetype CLOUDHSM `
 -J-classpath '-J"C:\Program Files\Amazon\CloudHSM\java\*"'
```

Si conectas varias instancias de cliente a tu AWS CloudHSM clúster, la importación de un certificado al almacén de claves de una instancia de cliente no hará que el certificado esté disponible automáticamente en otras instancias de cliente. Es necesario importar el certificado en cada instancia del cliente.

Uso de keytool para eliminar certificados del almacén de AWS CloudHSM claves

En el comando siguiente, se muestra un ejemplo de cómo se elimina un certificado de un almacén de claves de keytool para Java.

Linux

```
$ keytool -delete -alias mydomain \
 -keystore my_keystore.store \
 -storetype CLOUDHSM \
```

```
 -J-classpath '-J/opt/cloudhsm/java/*'
```

Windows

```
PS C:\> keytool -delete -alias mydomain `
 -keystore my_keystore.store `
 -storetype CLOUDHSM `
 -J-classpath '-J"C:\Program Files\Amazon\CloudHSM\java\*"'
```

Si conectas varias instancias de cliente a tu AWS CloudHSM clúster, al eliminar un certificado del almacén de claves de una instancia de cliente, no se eliminará automáticamente el certificado de otras instancias de cliente. Es necesario eliminar el certificado en cada instancia de cliente.

Importación de un certificado en funcionamiento al almacén de AWS CloudHSM claves mediante keytool

Cuando se firma una solicitud de firma de certificado (CSR), es posible importarla en el almacén de claves de AWS CloudHSM y asociarla con el par de claves apropiado. Puede ver un ejemplo en el siguiente comando:

Linux

```
$ keytool -importcert -noprompt -alias <key pair label> \
 -file my_certificate.crt \
 -keystore my_keystore.store \
 -storetype CLOUDHSM \
 -J-classpath '-J/opt/cloudhsm/java/*'
```

Windows

```
PS C:\> keytool -importcert -noprompt -alias <key pair label> `
 -file my_certificate.crt `
 -keystore my_keystore.store `
 -storetype CLOUDHSM `
 -J-classpath '-J"C:\Program Files\Amazon\CloudHSM\java\*"'
```

El alias debe ser un par de claves con un certificado asociado del almacén de claves. Si la clave se genera fuera de keytool o en otra instancia del cliente, primero debe importar los metadatos de la clave y el certificado en el almacén de claves.

Es necesario que la cadena de certificados se pueda verificar. Si no puede verificar el certificado, es posible que deba importar el certificado de firma (entidad de certificación) en el almacén de claves para poder verificar la cadena.

Exportación de certificados mediante keytool

En el ejemplo siguiente, se genera un certificado en formato X.509 binario. Para exportar un certificado en un formato legible, añada `-rfc` en el comando `-exportcert`.

Linux

```
$ keytool -exportcert -alias <key pair label> \
 -file my_exported_certificate.crt \
 -keystore my_keystore.store \
 -storetype CLOUDHSM \
 -J-classpath '-J/opt/cloudhsm/java/*'
```

Windows

```
PS C:\> keytool -exportcert -alias <key pair label> `
 -file my_exported_certificate.crt `
 -keystore my_keystore.store `
 -storetype CLOUDHSM `
 -J-classpath '-J"C:\Program Files\Amazon\CloudHSM\java\*"'
```

Uso del almacén de AWS CloudHSM claves con Jarsigner

Jarsigner es una popular utilidad de línea de comandos para firmar archivos JAR mediante una clave almacenada de forma segura en un HSM. La documentación de AWS CloudHSM no ofrece un tutorial completo sobre Jarsigner. En esta sección, se explican los parámetros de Jarsigner que debe utilizar para firmar y verificar las firmas AWS CloudHSM como fuente de confianza en el almacén de claves. AWS CloudHSM

Configuración de claves y certificados

Para poder firmar archivos JAR con Jarsigner, no olvide configurar o completar los siguientes pasos:

1. Siga las instrucciones de los requisitos previos del almacén de claves de AWS CloudHSM.
2. Configure las claves de firma y los certificados y la cadena de certificados asociados, que deben almacenarse en el almacén de AWS CloudHSM claves de la instancia de servidor o cliente actual.

Cree las claves AWS CloudHSM y, a continuación, importe los metadatos asociados a su almacén de AWS CloudHSM claves. Si desea utilizar keytool para configurar las claves y los certificados, consulte the section called “Creación de nuevas claves con keytool”. Si utiliza varias instancias de cliente para firmar los JAR, cree la clave e importe la cadena de certificados. A continuación, copie el archivo de almacén de claves resultante en cada instancia del cliente. Si genera nuevas claves con frecuencia, es posible que le resulte más fácil importar los certificados individualmente en cada instancia del cliente.

3. Toda la cadena de certificados debe ser verificable. Para que la cadena de certificados sea verificable, es posible que deba agregar el certificado de CA y los certificados intermedios al almacén de AWS CloudHSM claves. Consulte el fragmento de código en the section called “Firmar un archivo JAR con AWS CloudHSM y Jarsigner” para obtener instrucciones acerca de cómo utilizar código Java para verificar la cadena de certificados. Si lo prefiere, puede utilizar keytool para importar los certificados. Para obtener instrucciones sobre cómo utilizar keytool, consulte the section called “Uso de keytool para importar certificados intermedios y raíz al almacén de AWS CloudHSM claves ”.

Firmar un archivo JAR con AWS CloudHSM y Jarsigner

Utilice el siguiente comando para firmar un archivo JAR:

Linux;

Para OpenJDK 8

```
jarsigner -keystore my_keystore.store \
 -signedjar signthisclass_signed.jar \
 -sigalg sha512withrsa \
 -storetype CloudHSM \
 -J-classpath '-J/opt/cloudhsm/java/*:/usr/lib/jvm/java-1.8.0/lib/tools.jar' \
 -J-Djava.library.path=/opt/cloudhsm/lib \
 signthisclass.jar <key pair label>
```

Para OpenJDK 11, OpenJDK 17 y OpenJDK 21

```
jarsigner -keystore my_keystore.store \
 -signedjar signthisclass_signed.jar \
 -sigalg sha512withrsa \
 -storetype CloudHSM \
 -J-classpath '-J/opt/cloudhsm/java/*' \
```

```
 -J-Djava.library.path=/opt/cloudhsm/lib \
 signthisclass.jar <key pair label>
```

Windows

Para OpenJDK8

```
jarsigner -keystore my_keystore.store `
 -signedjar signthisclass_signed.jar `
 -sigalg sha512withrsa `
 -storetype CloudHSM `
 -J-classpath '-JC:\Program Files\Amazon\CloudHSM\java\*;C:\Program Files\Java
\jdk1.8.0_331\lib\tools.jar' `
  "-J-Djava.library.path='C:\Program Files\Amazon\CloudHSM\lib\'" `
 signthisclass.jar <key pair label>
```

Para OpenJDK 11, OpenJDK 17 y OpenJDK 21

```
jarsigner -keystore my_keystore.store `
 -signedjar signthisclass_signed.jar `
 -sigalg sha512withrsa `
 -storetype CloudHSM `
 -J-classpath '-JC:\Program Files\Amazon\CloudHSM\java\*'`
  "-J-Djava.library.path='C:\Program Files\Amazon\CloudHSM\lib\'" `
 signthisclass.jar <key pair label>
```

Utilice el siguiente comando para verificar un JAR firmado:

Linux

Para OpenJDK8

```
jarsigner -verify \
 -keystore my_keystore.store \
 -sigalg sha512withrsa \
 -storetype CloudHSM \
 -J-classpath '-J/opt/cloudhsm/java/*:/usr/lib/jvm/java-1.8.0/lib/tools.jar' \
 -J-Djava.library.path=/opt/cloudhsm/lib \
 signthisclass_signed.jar <key pair label>
```

Para OpenJDK 11, OpenJDK 17 y OpenJDK 21

```
jarsigner -verify \
 -keystore my_keystore.store \
 -sigalg sha512withrsa \
 -storetype CloudHSM \
 -J-classpath '-J/opt/cloudhsm/java/*' \
 -J-Djava.library.path=/opt/cloudhsm/lib \
 signthisclass_signed.jar <key pair label>
```

Windows

Para OpenJDK 8

```
jarsigner -verify `
 -keystore my_keystore.store `
 -sigalg sha512withrsa `
 -storetype CloudHSM `
 -J-classpath '-JC:\Program Files\Amazon\CloudHSM\java\*;C:\Program Files\Java
\jdk1.8.0_331\lib\tools.jar' `
 "-J-Djava.library.path='C:\Program Files\Amazon\CloudHSM\lib\'" `
 signthisclass_signed.jar <key pair label>
```

Para OpenJDK 11, OpenJDK 17 y OpenJDK 21

```
jarsigner -verify `
 -keystore my_keystore.store `
 -sigalg sha512withrsa `
 -storetype CloudHSM `
 -J-classpath '-JC:\Program Files\Amazon\CloudHSM\java\*`
 "-J-Djava.library.path='C:\Program Files\Amazon\CloudHSM\lib\'" `
 signthisclass_signed.jar <key pair label>
```

Problemas conocidos

1. No admitimos las claves EC con Keytool y Jarsigner.

Uso de SDK 3 de cliente para la integración con Java Keytool y Jarsigner

AWS CloudHSM el almacén de claves es un almacén de claves JCE de uso especial que utiliza los certificados asociados a las claves de su HSM a través de herramientas de terceros, como y. `keytool` `jarsigner` AWS CloudHSM no almacena los certificados en el HSM, ya que los certificados son datos públicos y no confidenciales. El almacén de AWS CloudHSM claves almacena los certificados en un archivo local y los asigna a las claves correspondientes del HSM.

Cuando utiliza el almacén de AWS CloudHSM claves para generar nuevas claves, no se genera ninguna entrada en el archivo del almacén de claves local; las claves se crean en el HSM. Del mismo modo, cuando utiliza el almacén de claves de AWS CloudHSM para buscar claves, la búsqueda se transfiere al HSM. Al almacenar los certificados en el almacén de AWS CloudHSM claves, el proveedor comprueba que existe un par de claves con el alias correspondiente en el HSM y, a continuación, asocia el certificado proporcionado al par de claves correspondiente.

Temas

- Requisitos previos
- Uso del almacén de AWS CloudHSM claves con keytool
- Uso del almacén de AWS CloudHSM claves con jarsigner
- Problemas conocidos
- Registrar claves preexistentes con el almacén de claves AWS CloudHSM

Requisitos previos

Para usar el almacén de AWS CloudHSM claves, primero debe inicializar y configurar el AWS CloudHSM SDK de JCE.

Paso 1: Instalar JCE

Para instalar el JCE, incluidos los requisitos previos del AWS CloudHSM cliente, siga los pasos para instalar la biblioteca Java.

Paso 2: Agregar credenciales de inicio de sesión de HSM a variables de entorno

Configure las variables de entorno para que contengan las credenciales de inicio de sesión de HSM.

```
export HSM_PARTITION=PARTITION_1
export HSM_USER=<HSM user name>
export HSM_PASSWORD=<HSM password>
```

> **Note**
>
> La JCE de CloudHSM dispone de varias opciones de inicio de sesión. Para utilizar el almacén de AWS CloudHSM claves con aplicaciones de terceros, debe utilizar el inicio de sesión implícito con variables de entorno. Si desea utilizar el inicio de sesión explícito a través del código de la aplicación, debe crear su propia aplicación con el almacén de AWS CloudHSM claves. Para obtener información adicional, consulte el artículo sobre el uso del almacén de AWS CloudHSM claves.

Paso 3: Registrar el proveedor de JCE

Para registrar el proveedor de JCE, en la CloudProvider configuración de Java.

1. Abra el archivo de configuración java.security de la instalación de Java para editarlo.
2. En el archivo de configuración java.security, agregue `com.cavium.provider.CaviumProvider` como último proveedor. Por ejemplo, si hay nueve proveedores en el archivo java.security, agregue el siguiente proveedor como último proveedor de la sección. Si el proveedor de Cavium se agregara con una prioridad mayor, el rendimiento del sistema podría verse negativamente afectado.

   ```
   security.provider.10=com.cavium.provider.CaviumProvider
   ```

   > **Note**
   >
   > Es posible que los usuarios avanzados estén acostumbrados a especificar las opciones `-providerName`, `-providerclass` y `-providerpath` de la línea de comandos cuando utilizan keytool, en lugar de actualizar el archivo de configuración de seguridad. Si intenta especificar las opciones de la línea de comandos al generar claves con el almacén de AWS CloudHSM claves, se producirán errores.

Uso del almacén de AWS CloudHSM claves con keytool

Keytool es una conocida utilidad de línea de comandos para tareas comunes de claves y certificados en los sistemas Linux. La documentación de AWS CloudHSM no ofrece un tutorial completo

sobre keytool. En este artículo se explican los parámetros específicos que debe utilizar con varias funciones de la herramienta clave cuando se utiliza AWS CloudHSM como raíz de confianza a través del almacén de AWS CloudHSM claves.

Cuando utilice keytool con el almacén de AWS CloudHSM claves, especifique los siguientes argumentos para cualquier comando de keytool:

```
-storetype CLOUDHSM \
  -J-classpath '-J/opt/cloudhsm/java/*' \
  -J-Djava.library.path=/opt/cloudhsm/lib
```

Si desea crear un nuevo archivo de almacén de claves mediante el almacén de AWS CloudHSM claves, consulte. Usando AWS CloudHSM KeyStore Si desea utilizar un almacén de claves existente, especifique el nombre (incluida la ruta) con el argumento keystore en keytool. Si especifica un archivo de almacén de claves que no existe en un comando de keytool, el almacén de AWS CloudHSM claves crea un nuevo archivo de almacén de claves.

Creación de nuevas claves con keytool

Puede usar keytool para generar cualquier tipo de clave compatible con el SDK de JCE AWS CloudHSM. Vea la lista completa de claves junto con su longitud en el artículo de claves admitidas de la biblioteca de Java.

> **Important**
>
> Una clave generada a través de keytool se genera en el software y, a continuación, se importa AWS CloudHSM como una clave persistente y extraíble.

Las instrucciones para crear claves no extraíbles directamente en el HSM y luego usarlas con keytool o Jarsigner se muestran en el ejemplo de código de Registrar claves preexistentes con Key Store. AWS CloudHSM Le recomendamos encarecidamente que genere las claves no exportables fuera de keytool y que después importe los certificados correspondientes en el almacén de claves. Si utilizas claves RSA o EC extraíbles a través de keytool y jarsigner, los proveedores exportan las claves desde y luego las utilizan localmente para las operaciones de firma. AWS CloudHSM

Si tiene varias instancias de cliente conectadas al clúster de CloudHSM, tenga en cuenta que, aunque se importe un certificado en el almacén de claves de una instancia del cliente, los certificados

no estarán disponibles automáticamente en otras instancias del cliente. Para registrar la clave y los certificados asociados en cada instancia del cliente, debe ejecutar una aplicación Java, tal y como se describe en Generar una CSR con Keytool. Si lo desea, también puede realizar los cambios necesarios en un cliente y copiar el archivo de almacén de claves resultante en las demás instancias del cliente.

Ejemplo 1: generar una clave AES-256 simétrica y guardarla en un archivo de almacén de claves llamado «my_keystore.store» del directorio de trabajo. Reemplace *<secret label>* por una etiqueta única.

```
keytool -genseckey -alias <secret label> -keyalg aes \
  -keysize 256 -keystore my_keystore.store \
  -storetype CloudHSM -J-classpath '-J/opt/cloudhsm/java/*' \
  -J-Djava.library.path=/opt/cloudhsm/lib/
```

Ejemplo 2: generar un par de claves RSA 2048 y guardarlo en un archivo de almacén de claves llamado “my_keystore.store” en el directorio de trabajo. Reemplace *<RSA key pair label>* por una etiqueta única.

```
keytool -genkeypair -alias <RSA key pair label> \
        -keyalg rsa -keysize 2048 \
        -sigalg sha512withrsa \
        -keystore my_keystore.store \
        -storetype CLOUDHSM \
        -J-classpath '-J/opt/cloudhsm/java/*' \
        -J-Djava.library.path=/opt/cloudhsm/lib/
```

Ejemplo 3: generar una clave p256 ED y guardarla en un archivo de almacén de claves llamado «my_keystore.store» del directorio de trabajo. Reemplace *<ec key pair label>* por una etiqueta única.

```
keytool -genkeypair -alias <ec key pair label> \
        -keyalg ec -keysize 256 \
        -sigalg SHA512withECDSA \
        -keystore my_keystore.store \
        -storetype CLOUDHSM \
        -J-classpath '-J/opt/cloudhsm/java/*' \
        -J-Djava.library.path=/opt/cloudhsm/lib/
```

Encontrará una lista de los algoritmos de firma compatibles en la biblioteca de Java.

Eliminación de claves con keytool

El almacén de AWS CloudHSM claves no admite la eliminación de claves. Para eliminar la clave, debe usar la `deleteKey` función de la herramienta AWS CloudHSM de línea de comandos,deleteKey.

Generación de CSR con keytool

Para tener la máxima flexibilidad al generar una solicitud de firma de certificado (CSR), utilice Motor dinámico de OpenSSL. El comando siguiente utiliza keytool para generar una CSR de un par de claves con el alias `my-key-pair`.

```
keytool -certreq -alias <key pair label> \
        -file my_csr.csr \
        -keystore my_keystore.store \
        -storetype CLOUDHSM \
        -J-classpath '-J/opt/cloudhsm/java/*' \
        -J-Djava.library.path=/opt/cloudhsm/lib/
```

> **Note**
>
> Para poder utilizar un par de claves de keytool, ese par de claves debe tener una entrada en el archivo de almacén de claves especificado. Si desea utilizar un par de claves generado fuera de keytool, debe importar los metadatos de las claves y los certificados en el almacén de claves. Para obtener instrucciones sobre cómo importar los datos del almacén de claves, consulte Importación de certificados intermedios y raíz al almacén de AWS CloudHSM claves mediante Keytool.

Uso de keytool para importar certificados intermedios y raíz al almacén de claves AWS CloudHSM

Para importar un certificado de CA, debe habilitar la verificación de una cadena de certificados completa en un certificado recién importado. A continuación, se muestra un ejemplo del comando:

```
keytool -import -trustcacerts -alias rootCAcert \
        -file rootCAcert.cert -keystore my_keystore.store \
        -storetype CLOUDHSM \
        -J-classpath '-J/opt/cloudhsm/java/*' \
        -J-Djava.library.path=/opt/cloudhsm/lib/
```

Si conectas varias instancias de cliente a tu AWS CloudHSM clúster, la importación de un certificado al almacén de claves de una instancia de cliente no hará que el certificado esté disponible automáticamente en otras instancias de cliente. Es necesario importar el certificado en cada instancia del cliente.

Uso de keytool para eliminar certificados del almacén de AWS CloudHSM claves

En el comando siguiente, se muestra un ejemplo de cómo se elimina un certificado de un almacén de claves de keytool para Java.

```
keytool -delete -alias mydomain -keystore \
        -keystore my_keystore.store \
        -storetype CLOUDHSM \
        -J-classpath '-J/opt/cloudhsm/java/*' \
        -J-Djava.library.path=/opt/cloudhsm/lib/
```

Si conectas varias instancias de cliente a tu AWS CloudHSM clúster, al eliminar un certificado del almacén de claves de una instancia de cliente, no se eliminará automáticamente el certificado de otras instancias de cliente. Es necesario eliminar el certificado en cada instancia de cliente.

Importación de un certificado en funcionamiento al almacén de AWS CloudHSM claves mediante keytool

Cuando se firma una solicitud de firma de certificado (CSR), es posible importarla en el almacén de claves de AWS CloudHSM y asociarla con el par de claves apropiado. Puede ver un ejemplo en el siguiente comando:

```
keytool -importcert -noprompt -alias <key pair label> \
        -file my_certificate.crt \
        -keystore my_keystore.store
        -storetype CLOUDHSM \
        -J-classpath '-J/opt/cloudhsm/java/*' \
        -J-Djava.library.path=/opt/cloudhsm/lib/
```

El alias debe ser un par de claves con un certificado asociado del almacén de claves. Si la clave se genera fuera de keytool o en otra instancia del cliente, primero debe importar los metadatos de la clave y el certificado en el almacén de claves. Para obtener instrucciones sobre cómo importar los metadatos del certificado, consulte el ejemplo de código en Cómo registrar claves preexistentes en el almacén de AWS CloudHSM claves.

Es necesario que la cadena de certificados se pueda verificar. Si no puede verificar el certificado, es posible que deba importar el certificado de firma (entidad de certificación) en el almacén de claves para poder verificar la cadena.

Exportación de certificados mediante keytool

En el ejemplo siguiente, se genera un certificado en formato X.509 binario. Para exportar un certificado en un formato legible, añada `-rfc` en el comando `-exportcert`.

```
keytool -exportcert -alias <key pair label> \
        -file my_exported_certificate.crt \
        -keystore my_keystore.store \
        -storetype CLOUDHSM \
        -J-classpath '-J/opt/cloudhsm/java/*' \
        -J-Djava.library.path=/opt/cloudhsm/lib/
```

Uso del almacén de AWS CloudHSM claves con jarsigner

Jarsigner es una popular utilidad de línea de comandos para firmar archivos JAR mediante una clave almacenada de forma segura en un HSM. La documentación de AWS CloudHSM no ofrece un tutorial completo sobre Jarsigner. En esta sección, se explican los parámetros de Jarsigner que debe utilizar para firmar y verificar las firmas AWS CloudHSM como fuente de confianza en el almacén de claves. AWS CloudHSM

Configuración de claves y certificados

Para poder firmar archivos JAR con Jarsigner, no olvide configurar o completar los siguientes pasos:

1. Siga las instrucciones de los requisitos previos del almacén de claves de AWS CloudHSM.
2. Configure las claves de firma y los certificados y la cadena de certificados asociados, que deben almacenarse en el almacén de AWS CloudHSM claves de la instancia de servidor o cliente actual. Cree las claves AWS CloudHSM y, a continuación, importe los metadatos asociados a su almacén de AWS CloudHSM claves. Utilice el ejemplo de código que se muestra en Cómo registrar claves preexistentes en el almacén de AWS CloudHSM claves para importar los metadatos al almacén de claves. Si desea utilizar keytool para configurar las claves y los certificados, consulte Creación de nuevas claves con keytool. Si utiliza varias instancias de cliente para firmar los JAR, cree la clave e importe la cadena de certificados. A continuación, copie el archivo de almacén de claves resultante en cada instancia del cliente. Si genera nuevas claves con frecuencia, es posible que le resulte más fácil importar los certificados individualmente en cada instancia del cliente.

3. Toda la cadena de certificados debe ser verificable. Para que la cadena de certificados sea verificable, es posible que deba agregar el certificado de CA y los certificados intermedios al almacén de AWS CloudHSM claves. Consulte el fragmento de código en Firmar un archivo JAR con Jarsigner para obtener instrucciones sobre cómo usar AWS CloudHSM el código Java para verificar la cadena de certificados. Si lo prefiere, puede utilizar keytool para importar los certificados. Para obtener instrucciones sobre el uso de keytool, consulte Uso de Keytool para importar certificados intermedios y raíz a Key Store. AWS CloudHSM

Firma de archivos JAR con AWS CloudHSM y jarsigner

Utilice el siguiente comando para firmar un archivo JAR:

```
jarsigner -keystore my_keystore.store \
        -signedjar signthisclass_signed.jar \
        -sigalg sha512withrsa \
        -storetype CloudHSM \
        -J-classpath '-J/opt/cloudhsm/java/*:/usr/lib/jvm/java-1.8.0/lib/tools.jar' \
        -J-Djava.library.path=/opt/cloudhsm/lib \
        signthisclass.jar <key pair label>
```

Utilice el siguiente comando para verificar un JAR firmado:

```
jarsigner -verify \
        -keystore my_keystore.store \
        -sigalg sha512withrsa \
        -storetype CloudHSM \
        -J-classpath '-J/opt/cloudhsm/java/*:/usr/lib/jvm/java-1.8.0/lib/tools.jar' \
        -J-Djava.library.path=/opt/cloudhsm/lib \
        signthisclass_signed.jar <key pair label>
```

Problemas conocidos

En la siguiente lista encontrará los problemas conocidos actuales.

- Al generar claves con keytool, el primer proveedor de la configuración del proveedor no puede ser. CaviumProvider
- Cuando se generan claves con keytool, el primer proveedor (compatible) del archivo de configuración de seguridad se utiliza para generar la clave. Normalmente, es un proveedor de software. A continuación, se asigna un alias a la clave generada y se importa al AWS CloudHSM HSM como clave persistente (simbólica) durante el proceso de adición de claves.

- Cuando utilices keytool con un almacén de AWS CloudHSM claves, no especifiques `-providerName` ni `-providerpath` opciones en la línea de comandos. `-providerclass` Especifique estas opciones en el archivo del proveedor de seguridad tal y como se describe en los requisitos previos del almacén de claves.
- Cuando se usan claves de EC no extraíbles a través de keytool y Jarsigner, es necesario eliminar o deshabilitar el proveedor SunEC de la lista de proveedores del archivo java.security. Si utiliza claves EC extraíbles a través de keytool y Jarsigner, los proveedores exportan los bits clave del AWS CloudHSM HSM y utilizan la clave localmente para las operaciones de firma. No se recomienda utilizar claves exportables con keytool o Jarsigner.

Registrar claves preexistentes con el almacén de claves AWS CloudHSM

Para tener la máxima seguridad y flexibilidad con los atributos y el etiquetado, le recomendamos que genere sus claves de firma con key_mgmt_util. También puede usar una aplicación de Java para generar la clave en AWS CloudHSM.

En la siguiente sección se proporciona un ejemplo de código que muestra cómo generar un nuevo par de claves en el HSM y registrarlo con las claves existentes importadas al almacén de AWS CloudHSM claves. Las claves importadas están disponibles para que puedan utilizarse en herramientas de terceros como keytool y Jarsigner.

Si desea utilizar una clave preexistente, modifique el ejemplo de código para buscar una clave por su etiqueta en lugar de generar una clave nueva. El ejemplo de código para buscar una clave por etiqueta está disponible en el ejemplo KeyUtilitiesRunner .java de. GitHub

> **Important**
>
> Al registrar una clave almacenada en AWS CloudHSM un almacén de claves local, no se exporta la clave. Cuando se registra la clave, el almacén de claves registra el alias (o la etiqueta) de la clave y relaciona localmente los objetos del certificado del almacén con un par de claves de AWS CloudHSM. Si el par de claves se crea como no exportable, los bits de la clave no saldrán del HSM.

```
                    //
```

```
// Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
//
// Permission is hereby granted, free of charge, to any person obtaining a copy of
this
// software and associated documentation files (the "Software"), to deal in the
Software
// without restriction, including without limitation the rights to use, copy, modify,
// merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
// permit persons to whom the Software is furnished to do so.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
// INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
// PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
// HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
// OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
// SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
//

package com.amazonaws.cloudhsm.examples;

import com.cavium.key.CaviumKey;
import com.cavium.key.parameter.CaviumAESKeyGenParameterSpec;
import com.cavium.key.parameter.CaviumRSAKeyGenParameterSpec;
import com.cavium.asn1.Encoder;
import com.cavium.cfm2.Util;

import javax.crypto.KeyGenerator;

import java.io.ByteArrayInputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileNotFoundException;

import java.math.BigInteger;

import java.security.*;
import java.security.cert.Certificate;
import java.security.cert.CertificateException;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;
import java.security.interfaces.RSAPrivateKey;
import java.security.interfaces.RSAPublicKey;
import java.security.KeyStore.PasswordProtection;
import java.security.KeyStore.PrivateKeyEntry;
```

```
import java.security.KeyStore.Entry;

import java.util.Calendar;
import java.util.Date;
import java.util.Enumeration;

//
// KeyStoreExampleRunner demonstrates how to load a keystore, and associate a
 certificate with a
// key in that keystore.
//
// This example relies on implicit credentials, so you must setup your environment
 correctly.
//
// https://docs.aws.amazon.com/cloudhsm/latest/userguide/java-library-
install.html#java-library-credentials
//

public class KeyStoreExampleRunner {

    private static byte[] COMMON_NAME_OID = new byte[] { (byte) 0x55, (byte) 0x04,
 (byte) 0x03 };
    private static byte[] COUNTRY_NAME_OID = new byte[] { (byte) 0x55, (byte) 0x04,
 (byte) 0x06 };
    private static byte[] LOCALITY_NAME_OID = new byte[] { (byte) 0x55, (byte) 0x04,
 (byte) 0x07 };
    private static byte[] STATE_OR_PROVINCE_NAME_OID = new byte[] { (byte) 0x55,
 (byte) 0x04, (byte) 0x08 };
    private static byte[] ORGANIZATION_NAME_OID = new byte[] { (byte) 0x55, (byte)
 0x04, (byte) 0x0A };
    private static byte[] ORGANIZATION_UNIT_OID = new byte[] { (byte) 0x55, (byte)
 0x04, (byte) 0x0B };

    private static String helpString = "KeyStoreExampleRunner%n" +
           "This sample demonstrates how to load and store keys using a keystore.%n%n"
 +
           "Options%n" +
           "\t--help\t\t\tDisplay this message.%n" +
           "\t--store <filename>\t\tPath of the keystore.%n" +
           "\t--password <password>\t\tPassword for the keystore (not your CU
 password).%n" +
           "\t--label <label>\t\t\tLabel to store the key and certificate under.%n" +
           "\t--list\t\t\tList all the keys in the keystore.%n%n";
```

```
    public static void main(String[] args) throws Exception {
        Security.addProvider(new com.cavium.provider.CaviumProvider());
        KeyStore keyStore = KeyStore.getInstance("CloudHSM");

        String keystoreFile = null;
        String password = null;
        String label = null;
        boolean list = false;
        for (int i = 0; i < args.length; i++) {
            String arg = args[i];
            switch (args[i]) {
                case "--store":
                    keystoreFile = args[++i];
                    break;
                case "--password":
                    password = args[++i];
                    break;
                case "--label":
                    label = args[++i];
                    break;
                case "--list":
                    list = true;
                    break;
                case "--help":
                    help();
                    return;
            }
        }

        if (null == keystoreFile || null == password) {
            help();
            return;
        }

        if (list) {
            listKeys(keystoreFile, password);
            return;
        }

        if (null == label) {
            label = "Keystore Example Keypair";
        }

        //
```

```
        // This call to keyStore.load() will open the pkcs12 keystore with the supplied
        // password and connect to the HSM. The CU credentials must be specified using
        // standard CloudHSM login methods.
        //
        try {
            FileInputStream instream = new FileInputStream(keystoreFile);
            keyStore.load(instream, password.toCharArray());
        } catch (FileNotFoundException ex) {
            System.err.println("Keystore not found, loading an empty store");
            keyStore.load(null, null);
        }

        PasswordProtection passwd = new PasswordProtection(password.toCharArray());
        System.out.println("Searching for example key and certificate...");

        PrivateKeyEntry keyEntry = (PrivateKeyEntry) keyStore.getEntry(label, passwd);
        if (null == keyEntry) {
            //
            // No entry was found, so we need to create a key pair and associate a
certificate.
            // The private key will get the label passed on the command line. The
keystore alias
            // needs to be the same as the private key label. The public key will have
":public"
            // appended to it. The alias used in the keystore will We associate the
certificate
            // with the private key.
            //
            System.out.println("No entry found, creating...");
            KeyPair kp = generateRSAKeyPair(2048, label + ":public", label);
            System.out.printf("Created a key pair with the handles %d/%d%n",
((CaviumKey) kp.getPrivate()).getHandle(), ((CaviumKey) kp.getPublic()).getHandle());

            //
            // Generate a certificate and associate the chain with the private key.
            //
            Certificate self_signed_cert = generateCert(kp);
            Certificate[] chain = new Certificate[1];
            chain[0] = self_signed_cert;
            PrivateKeyEntry entry = new PrivateKeyEntry(kp.getPrivate(), chain);

            //
            // Set the entry using the label as the alias and save the store.
            // The alias must match the private key label.
```

```
            //
            keyStore.setEntry(label, entry, passwd);

            FileOutputStream outstream = new FileOutputStream(keystoreFile);
            keyStore.store(outstream, password.toCharArray());
            outstream.close();

            keyEntry = (PrivateKeyEntry) keyStore.getEntry(label, passwd);
        }

        long handle = ((CaviumKey) keyEntry.getPrivateKey()).getHandle();
        String name = keyEntry.getCertificate().toString();
        System.out.printf("Found private key %d with certificate %s%n", handle, name);
    }

    private static void help() {
        System.out.println(helpString);
    }

    //
    // Generate a non-extractable / non-persistent RSA keypair.
    // This method allows us to specify the public and private labels, which
    // will make KeyStore alises easier to understand.
    //
    public static KeyPair generateRSAKeyPair(int keySizeInBits, String publicLabel,
String privateLabel)
            throws InvalidAlgorithmParameterException, NoSuchAlgorithmException,
NoSuchProviderException {

        boolean isExtractable = false;
        boolean isPersistent = false;
        KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("rsa", "Cavium");
        CaviumRSAKeyGenParameterSpec spec = new
CaviumRSAKeyGenParameterSpec(keySizeInBits, new BigInteger("65537"), publicLabel,
privateLabel, isExtractable, isPersistent);

        keyPairGen.initialize(spec);

        return keyPairGen.generateKeyPair();
    }

    //
    // Generate a certificate signed by a given keypair.
    //
```

```
    private static Certificate generateCert(KeyPair kp) throws CertificateException {
        CertificateFactory cf = CertificateFactory.getInstance("X509");
        PublicKey publicKey = kp.getPublic();
        PrivateKey privateKey = kp.getPrivate();
        byte[] version = Encoder.encodeConstructed((byte) 0,
Encoder.encodePositiveBigInteger(new BigInteger("2"))); // version 1
        byte[] serialNo = Encoder.encodePositiveBigInteger(new BigInteger(1,
Util.computeKCV(publicKey.getEncoded())));

        // Use the SHA512 OID and algorithm.
        byte[] signatureOid = new byte[] {
            (byte) 0x2A, (byte) 0x86, (byte) 0x48, (byte) 0x86, (byte) 0xF7, (byte)
0x0D, (byte) 0x01, (byte) 0x01, (byte) 0x0D };
        String sigAlgoName = "SHA512WithRSA";

         byte[] signatureId = Encoder.encodeSequence(
                                        Encoder.encodeOid(signatureOid),
                                        Encoder.encodeNull());

         byte[] issuer = Encoder.encodeSequence(
                                    encodeName(COUNTRY_NAME_OID, "<Country>"),
                                    encodeName(STATE_OR_PROVINCE_NAME_OID, "<State>"),
                                    encodeName(LOCALITY_NAME_OID, "<City>"),
                                    encodeName(ORGANIZATION_NAME_OID,
"<Organization>"),
                                    encodeName(ORGANIZATION_UNIT_OID, "<Unit>"),
                                    encodeName(COMMON_NAME_OID, "<CN>")
                                );

         Calendar c = Calendar.getInstance();
         c.add(Calendar.DAY_OF_YEAR, -1);
         Date notBefore = c.getTime();
         c.add(Calendar.YEAR, 1);
         Date notAfter = c.getTime();
         byte[] validity = Encoder.encodeSequence(
                                        Encoder.encodeUTCTime(notBefore),
                                        Encoder.encodeUTCTime(notAfter)
                                    );
         byte[] key = publicKey.getEncoded();

         byte[] certificate = Encoder.encodeSequence(
                                        version,
                                        serialNo,
                                        signatureId,
```

```
                                        issuer,
                                        validity,
                                        issuer,
                                        key);
        Signature sig;
        byte[] signature = null;
        try {
            sig = Signature.getInstance(sigAlgoName, "Cavium");
            sig.initSign(privateKey);
            sig.update(certificate);
            signature = Encoder.encodeBitstring(sig.sign());

        } catch (Exception e) {
            System.err.println(e.getMessage());
            return null;
        }

        byte [] x509 = Encoder.encodeSequence(
                        certificate,
                        signatureId,
                        signature
                        );
        return cf.generateCertificate(new ByteArrayInputStream(x509));
    }

     //
     // Simple OID encoder.
     // Encode a value with OID in ASN.1 format
     //
     private static byte[] encodeName(byte[] nameOid, String value) {
        byte[] name = null;
        name = Encoder.encodeSet(
                    Encoder.encodeSequence(
                            Encoder.encodeOid(nameOid),
                            Encoder.encodePrintableString(value)
                    )
                );
        return name;
     }

    //
    // List all the keys in the keystore.
    //
```

```
    private static void listKeys(String keystoreFile, String password) throws Exception
 {
        KeyStore keyStore = KeyStore.getInstance("CloudHSM");

        try {
            FileInputStream instream = new FileInputStream(keystoreFile);
            keyStore.load(instream, password.toCharArray());
        } catch (FileNotFoundException ex) {
            System.err.println("Keystore not found, loading an empty store");
            keyStore.load(null, null);
        }

        for(Enumeration<String> entry = keyStore.aliases(); entry.hasMoreElements();) {
            System.out.println(entry.nextElement());
        }
    }

}
```

Otras integraciones de proveedores externos

Varios proveedores externos AWS CloudHSM lo respaldan como base de confianza. Esto significa que puede utilizar la solución de software de su elección al crear y almacenar las claves subyacentes en el clúster de CloudHSM. Como resultado, su carga de trabajo AWS puede confiar en las ventajas de latencia, disponibilidad, fiabilidad y elasticidad de CloudHSM. La siguiente lista incluye los proveedores externos compatibles con CloudHSM.

> **Note**
>
> AWS no avala ni avala a ningún proveedor externo.

- Vault de Hashicorpes una herramienta de administración de secretos diseñada para habilitar la colaboración y la gobernanza entre organizaciones. AWS CloudHSM Apoya AWS Key Management Service y constituye una base de confianza para una protección adicional.
- Thycotic Secrets Server ayuda a los clientes a administrar credenciales confidenciales en cuentras privilegiadas. Se apoya AWS CloudHSM como una raíz de confianza.

- El adaptador KMIP de P6R le permite utilizar sus AWS CloudHSM instancias a través de una interfaz KMIP estándar.
- PrimeKey EJBCA es una popular solución de código abierto para PKI. Le permite crear y almacenar pares de claves de forma segura con. AWS CloudHSM
- Box KeySafe proporciona la administración de claves de cifrado para el contenido en la nube a muchas organizaciones con estrictos requisitos de seguridad, privacidad y cumplimiento normativo. Además, los clientes pueden proteger KeySafe las claves directamente AWS Key Management Service o indirectamente AWS CloudHSM a través de AWS KMS Custom Key Store.
- Insyde Software es una AWS CloudHSM fuente de confianza para la firma de firmware.
- F5 BIG-IP LTM es AWS CloudHSM una base de confianza.
- Cloudera Navigator Key HSM le permite utilizar su clúster de CloudHSM para crear y almacenar claves para Cloudera Navigator Key Trustee Server.
- Venafi Trust Protection Platform proporciona una administración integral de identidades de máquinas para TLS, SSH y firma de código con la generación y protección de claves de AWS CloudHSM.

Monitorización AWS CloudHSM

Además de las funciones de registro integradas en el SDK del cliente, también puede utilizar AWS CloudTrail Amazon CloudWatch Logs y Amazon CloudWatch para supervisar AWS CloudHSM.

Registros del SDK del cliente

: Utilice el registro del SDK de cliente para supervisar la información de diagnóstico y solución de problemas de las aplicaciones que cree.

CloudTrail

: Úselo CloudTrail para monitorear todas las llamadas a la API de su AWS cuenta, incluidas las llamadas que realiza para crear y eliminar clústeres, módulos de seguridad de hardware (HSM) y etiquetas de recursos.

CloudWatch Registros

: Utilice CloudWatch los registros para supervisar los registros de sus instancias de HSM, que incluyen eventos para crear y eliminar usuarios de HSM, cambiar las contraseñas de los usuarios, crear y eliminar claves, etc.

CloudWatch

: Úselo CloudWatch para monitorear el estado de su clúster en tiempo real.

Temas

- Trabajo con los registros de SDK de cliente
- Trabajar con AWS CloudTrail y AWS CloudHSM
- Trabajar con Amazon CloudWatch Logs y AWS CloudHSM Audit Logs
- Obtener CloudWatch métricas para AWS CloudHSM

Trabajo con los registros de SDK de cliente

Puede recuperar los registros generados por el SDK del cliente. AWS CloudHSM ofrece una implementación del registro con el SDK de cliente 3 y el SDK de cliente 5.

Temas

- Registro de SDK 5 de cliente

- Registro de SDK 3 de cliente

Registro de SDK 5 de cliente

Los registros de SDK 5 de cliente contienen información sobre cada componente en un archivo con el nombre del componente. Puede usar la herramienta de configuración de SDK 5 de cliente para configurar el registro de cada componente.

Si no especifica una ubicación para el archivo, el sistema escribirá los registros en la ubicación predeterminada:

PKCS #11 library

- Linux

```
/opt/cloudhsm/run/cloudhsm-pkcs11.log
```

  Windows

```
C:\Program Files\Amazon\CloudHSM\cloudhsm-pkcs11.log
```

OpenSSL Dynamic Engine

- Linux

```
stderr
```

JCE provider

- Linux

```
/opt/cloudhsm/run/cloudhsm-jce.log
```

  Windows

```
C:\Program Files\Amazon\CloudHSM\cloudhsm-jce.log
```

Para obtener información sobre cómo configurar el registro para SDK 5 de cliente, consulte la Herramienta de configuración de SDK 5 de cliente.

Registro de SDK 3 de cliente

Los registros del SDK 3 del cliente contienen información detallada del daemon del AWS CloudHSM cliente. La ubicación de los registros depende del sistema operativo de la instancia cliente de Amazon EC2 en la que ejecute el daemon del cliente.

Amazon Linux

En Amazon Linux, los registros del AWS CloudHSM cliente se escriben en el archivo denominado`/opt/cloudhsm/run/cloudhsm_client.log`. Puede usar logrotate o una herramienta similar para rotar y administrar estos registros.

Amazon Linux 2

En Amazon Linux 2, los registros del AWS CloudHSM cliente se recopilan y almacenan en el diario. Puede usar journalctl para ver y administrar estos registros. Por ejemplo, utilice el siguiente comando para ver los registros del AWS CloudHSM cliente.

```
journalctl -f -u cloudhsm-client
```

CentOS 7

En Centos 7, los registros del AWS CloudHSM cliente se recopilan y almacenan en el diario. Puede usar journalctl para ver y administrar estos registros. Por ejemplo, utilice el siguiente comando para ver los registros del AWS CloudHSM cliente.

```
journalctl -f -u cloudhsm-client
```

CentOS 8

En Centos 8, los registros del AWS CloudHSM cliente se recopilan y almacenan en el diario. Puede usar journalctl para ver y administrar estos registros. Por ejemplo, utilice el siguiente comando para ver los registros del AWS CloudHSM cliente.

```
journalctl -f -u cloudhsm-client
```

RHEL 7

En Red Hat Enterprise Linux 7, los registros del AWS CloudHSM cliente se recopilan y almacenan en el diario. Puede usar journalctl para ver y administrar estos registros. Por ejemplo, utilice el siguiente comando para ver los registros del AWS CloudHSM cliente.

```
journalctl -f -u cloudhsm-client
```

RHEL 8

En Red Hat Enterprise Linux 8, los registros del AWS CloudHSM cliente se recopilan y almacenan en el diario. Puede usar journalctl para ver y administrar estos registros. Por ejemplo, utilice el siguiente comando para ver los registros del AWS CloudHSM cliente.

```
journalctl -f -u cloudhsm-client
```

Ubuntu 16.04

En Ubuntu 16.04, los registros del AWS CloudHSM cliente se recopilan y almacenan en el diario. Puede usar journalctl para ver y administrar estos registros. Por ejemplo, utilice el siguiente comando para ver los registros del AWS CloudHSM cliente.

```
journalctl -f -u cloudhsm-client
```

Ubuntu 18.04

En Ubuntu 18.04, los registros del AWS CloudHSM cliente se recopilan y almacenan en el diario. Puede usar journalctl para ver y administrar estos registros. Por ejemplo, utilice el siguiente comando para ver los registros del AWS CloudHSM cliente.

```
journalctl -f -u cloudhsm-client
```

Windows

- Para la versión 1.1.2 y posteriores del cliente de Windows:

  AWS CloudHSM Los registros del cliente se escriben en un `cloudhsm.log` archivo de la carpeta de archivos de AWS CloudHSM programa (`C:\Program Files\Amazon\CloudHSM \`). El nombre de cada archivo de registro tiene como sufijo una marca de tiempo que indica cuándo se inició el AWS CloudHSM cliente.

- Para la versión 1.1.1 y anteriores del cliente de Windows:

  El registros de cliente no se escriben en un archivo. Los registros se muestran en la línea de comandos o en la PowerShell ventana en la que se inició el cliente. AWS CloudHSM

Trabajar con AWS CloudTrail y AWS CloudHSM

AWS CloudHSM está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en AWS CloudHSM. CloudTrail captura todas las llamadas a la API AWS CloudHSM como eventos. Las llamadas capturadas incluyen llamadas desde la AWS CloudHSM consola y llamadas en código a las operaciones de la AWS CloudHSM API. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para AWS CloudHSM. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por usted CloudTrail, puede determinar a AWS CloudHSM qué dirección IP se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la Guía AWS CloudTrail del usuario. Para ver una lista completa de las operaciones de la AWS CloudHSM API, consulta las acciones en la referencia de la AWS CloudHSM API.

AWS CloudHSM información en CloudTrail

CloudTrail está habilitada en su AWS cuenta al crear la cuenta. Cuando se produce una actividad en AWS CloudHSM, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS . Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de tu cuenta AWS CloudHSM, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail Integraciones y servicios compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

CloudTrail registra todas AWS CloudHSM las operaciones, incluidas las operaciones de solo lectura, como `DescribeClusters` y`ListTags`, y las operaciones de administración, como `InitializeClusterCreatHsm`, y. `DeleteBackup`

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario root o AWS Identity and Access Management (IAM).
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el elemento CloudTrail UserIdentity.

Descripción AWS CloudHSM de las entradas de los archivos de registro

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que demuestra la AWS CloudHSM `CreateHsm` acción.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAJZVM5NEGZSTCITAMM:ExampleSession",
```

```
        "arn": "arn:aws:sts::111122223333:assumed-role/AdminRole/ExampleSession",
        "accountId": "111122223333",
        "accessKeyId": "ASIAIY22AX6VRYNBGJSA",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2017-07-11T03:48:44Z"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAJZVM5NEGZSTCITAMM",
                "arn": "arn:aws:iam::111122223333:role/AdminRole",
                "accountId": "111122223333",
                "userName": "AdminRole"
            }
        }
    },
    "eventTime": "2017-07-11T03:50:45Z",
    "eventSource": "cloudhsm.amazonaws.com",
    "eventName": "CreateHsm",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.251.233.179",
    "userAgent": "aws-internal/3",
    "requestParameters": {
        "availabilityZone": "us-west-2b",
        "clusterId": "cluster-fw7mh6mayb5"
    },
    "responseElements": {
        "hsm": {
            "eniId": "eni-65338b5a",
            "clusterId": "cluster-fw7mh6mayb5",
            "state": "CREATE_IN_PROGRESS",
            "eniIp": "10.0.2.7",
            "hsmId": "hsm-6lz2hfmnzbx",
            "subnetId": "subnet-02c28c4b",
            "availabilityZone": "us-west-2b"
        }
    },
    "requestID": "1dae0370-65ec-11e7-a770-6578d63de907",
    "eventID": "b73a5617-8508-4c3d-900d-aa8ac9b31d08",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Trabajar con Amazon CloudWatch Logs y AWS CloudHSM Audit Logs

Cuando un HSM de su cuenta recibe un comando de las herramientas de línea de AWS CloudHSM comandos o de las bibliotecas de software, registra la ejecución del comando en un registro de auditoría. Los registros de auditoría del HSM contienen todos los comandos de administración iniciados por el cliente, incluidos los que crean y eliminan el HSM, los que inician y cierran sesión en el HSM y los que administran usuarios y claves. Estos registros conforman un registro de confianza de las acciones que han cambiado el estado del HSM.

AWS CloudHSM recopila sus registros de auditoría de HSM y los envía a Amazon CloudWatch Logs en su nombre. Puede utilizar las funciones de los CloudWatch registros para gestionar sus registros de AWS CloudHSM auditoría, incluida la búsqueda y el filtrado de los registros y la exportación de los datos de registro a Amazon S3. Puede trabajar con sus registros de auditoría de HSM en la CloudWatch consola de Amazon o utilizar los comandos CloudWatch Logs de la CLI y los SDK de CloudWatch Logs.

Temas

- Cómo funciona el registro de auditoría de HSM
- Visualización de los registros de auditoría de HSM en CloudWatch los registros
- Interpretación de los registros de auditoría de HSM
- Referencia del registro de auditoría de HSM

Cómo funciona el registro de auditoría de HSM

El registro de auditoría se habilita automáticamente en todos los AWS CloudHSM clústeres. No se puede deshabilitar ni desactivar, y ninguna configuración puede AWS CloudHSM impedir la exportación de los registros a CloudWatch registros. Cada evento de registro tiene una marca temporal y un número de secuencia que indican el orden de los eventos y le ayudan a detectar cualquier intento de manipulación del registro.

Cada instancia de HSM genera su propio registro. Los registros de auditoría de los distintos HSM, incluidos aquellos que están en el mismo clúster, pueden ser diferentes. Por ejemplo, solo el primer HSM de cada clúster registra la inicialización del HSM. Los eventos de inicialización no aparecen en los registros de los HSM que se clonan de las copias de seguridad. Del mismo modo, cuando se

crea una clave, el HSM que la genera registra un evento de generación de clave. Los demás HSM del clúster registran un evento cuando reciben la clave a través de una sincronización.

AWS CloudHSM recopila los registros y los publica en los CloudWatch registros de tu cuenta. Para comunicarse con el servicio de CloudWatch registros en su nombre, AWS CloudHSM utiliza un rol vinculado al servicio. La política de IAM asociada a la función permite realizar solo las tareas necesarias AWS CloudHSM para enviar los registros de auditoría a Logs. CloudWatch

> **Important**
>
> Si creó un clúster antes del 20 de enero de 2018 pero no ha creado ninguna función vinculada al servicio asociada, debe crear una manualmente. Esto es necesario CloudWatch para recibir los registros de auditoría de su AWS CloudHSM clúster. Para obtener más información acerca de la creación de roles vinculados a servicios, consulte Descripción de los roles vinculados a servicios y Cómo crear un rol vinculado a un servicio en la guía del usuario de IAM.

Visualización de los registros de auditoría de HSM en CloudWatch los registros

Amazon CloudWatch Logs organiza los registros de auditoría en grupos de registros y, dentro de un grupo de registros, en flujos de registros. Cada entrada de registro es un evento. AWS CloudHSM crea un grupo de registros para cada clúster y un flujo de registros para cada HSM del clúster. No es necesario crear ningún componente de CloudWatch Logs ni cambiar ninguna configuración.

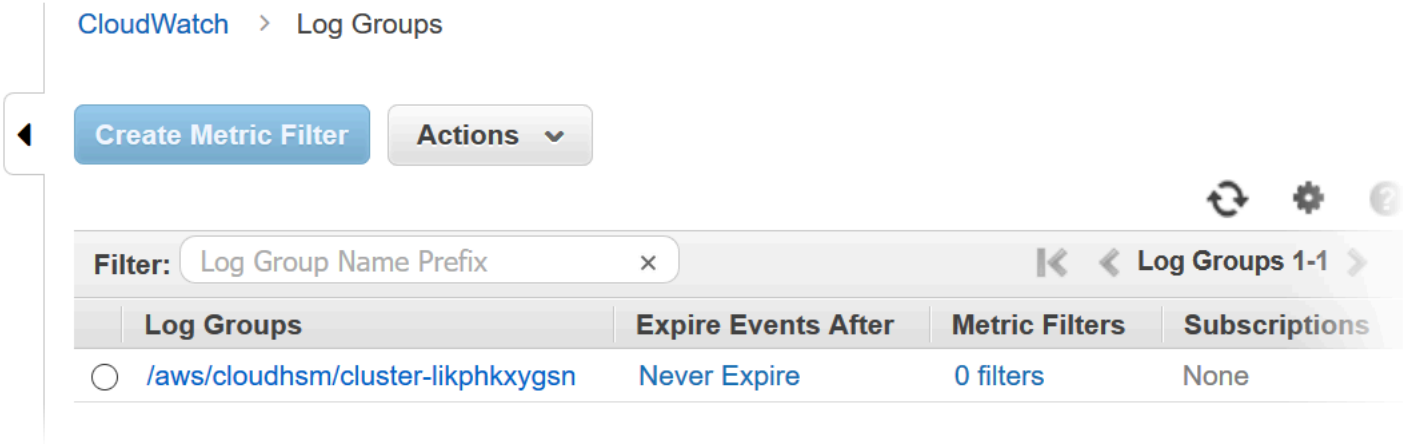
- El nombre del grupo de registros es `/aws/cloudhsm/<cluster ID>`; por ejemplo, `/aws/cloudhsm/cluster-likphkxygsn`. Cuando utilice el nombre del grupo de registros en una CLI o un PowerShell comando, asegúrese de ponerlo entre comillas dobles.
- El nombre de la secuencia de registros es el ID de HSM; por ejemplo, `hsm-nwbbiqbj4jk`.

  Por lo general, suele haber una secuencia de registros en cada HSM. Sin embargo, cualquier acción que cambie el ID de HSM, como cuando se produce un error en un HSM y se sustituye, crea una nueva secuencia de registros.

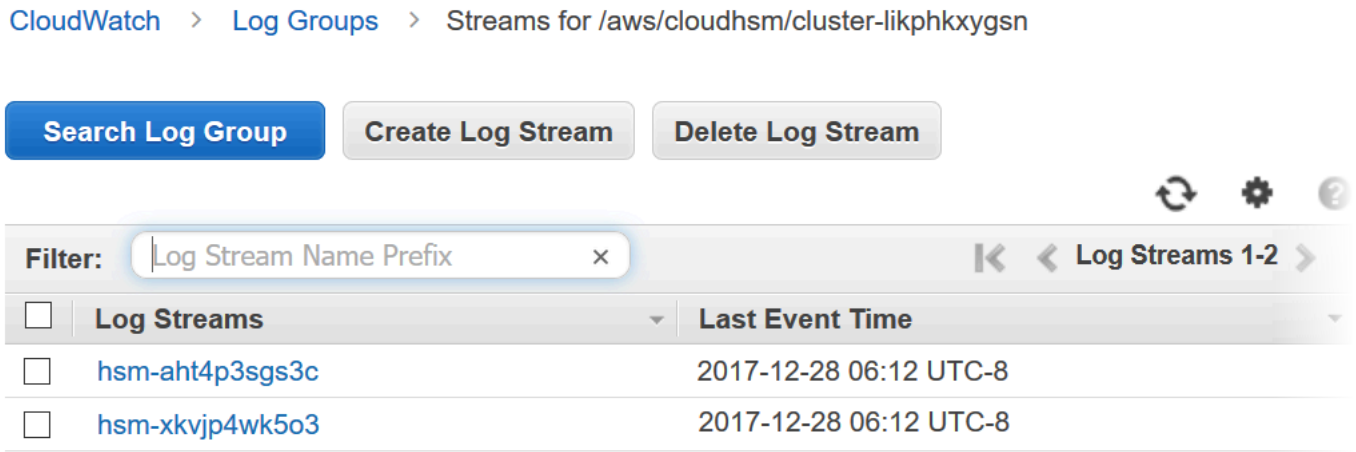
Para obtener más información sobre CloudWatch los conceptos de Logs, consulte Conceptos en la Guía del usuario de Amazon CloudWatch Logs.

Puede ver los registros de auditoría de un HSM desde la página de CloudWatch registros de AWS Management Console, los comandos de CloudWatch Logs de la CLI, los PowerShellcmdlets de CloudWatch Logs o los CloudWatch SDK de Logs. Para obtener instrucciones, consulta Ver datos de registro en la Guía del usuario de Amazon CloudWatch Logs.

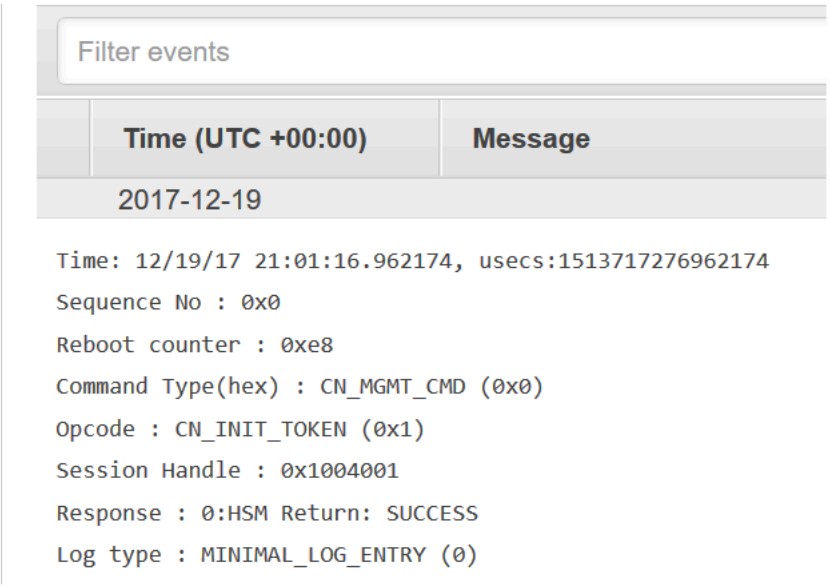
Por ejemplo, en la siguiente imagen se muestra el grupo de registros del clúster `cluster-likphkxygsn` de la AWS Management Console.



Cuando seleccione el nombre del grupo de registros del clúster, puede ver la secuencia de registros de cada uno de los HSM del clúster. En la imagen siguiente, se muestran las secuencias de registros de los HSM del clúster `cluster-likphkxygsn`.

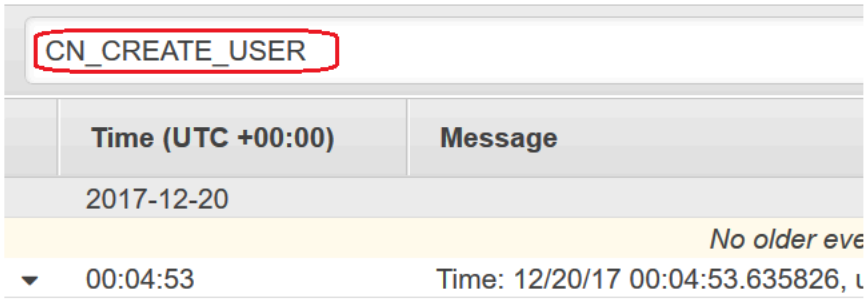


Cuando elige el nombre de una secuencia de registros de HSM, puede ver los eventos del registro de auditoría. Por ejemplo, este evento, cuyo número de secuencia es 0x0 y donde el valor de `Opcode` es `CN_INIT_TOKEN`, suele ser el primer evento del primer HSM de cada clúster. Este evento registra la inicialización del HSM del clúster.



Puede utilizar todas las funciones de los CloudWatch registros para gestionar sus registros de auditoría. Por ejemplo, puede utilizar la características Filter events (Filtrar eventos) para buscar un texto específico de un evento,como el `Opcode CN_CREATE_USER`.

Para buscar todos los eventos que no incluyen el texto especificado, añada un signo menos (-) delante del texto. Por ejemplo, para buscar eventos que no incluyen `CN_CREATE_USER`, escriba -CN_CREATE_USER.




```
Time: 12/20/17 00:04:53.635826, usecs:1513728293635826
Sequence No : 0x13a
Reboot counter : 0xe8
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_CREATE_USER (0x3)
Session Handle : 0x1014006
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : testuser
User Type : CN_CRYPTO_USER (1)
```

Interpretación de los registros de auditoría de HSM

Los eventos de los registros de auditoría del HSM tienen campos estándar. Algunos tipos de eventos contienen otros campos que capturan información útil sobre el evento. Por ejemplo, los eventos de inicio de sesión y administración de usuarios contienen el nombre del usuario y el tipo de usuario. Los comandos de administración de claves contienen el identificador de la clave.

Algunos de los campos proporcionan información importante. `Opcode` identifica el comando de administración que se está registrando. `Sequence No` identifica un evento de la secuencia de registros e indica el orden en que se registró.

Por ejemplo, el siguiente evento de ejemplo es el segundo evento (`Sequence No: 0x1`) de la secuencia de registros de un HSM. Muestra el HSM que generó una clave de cifrado de contraseña, lo que forma parte de la rutina de inicio.

```
Time: 12/19/17 21:01:17.140812, usecs:1513717277140812
Sequence No : 0x1
Reboot counter : 0xe8
```

```
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_GEN_PSWD_ENC_KEY (0x1d)
Session Handle : 0x1004001
Response : 0:HSM Return: SUCCESS
Log type : MINIMAL_LOG_ENTRY (0)
```

Los siguientes campos son comunes a todos los AWS CloudHSM eventos del registro de auditoría.

Tiempo

Hora a la que tuvo lugar el evento en la zona horaria UTC. La hora se muestra en lenguaje natural y en formato Unix en microsegundos.

Reboot counter (Contador de reinicios)

Contador ordinal persistente de 32 bits que aumenta cuando el hardware del HSM se reinicia.

Todos los eventos de una secuencia de registros tienen el mismo valor de contador de reinicios. Sin embargo, el contador de reinicios podría no ser el único en una secuencia de registros, ya que podría haber otros en las distintas instancias del HSM del mismo clúster.

Sequence No (Número de secuencia)

Contador ordinal de 64 bits que aumenta con cada evento del registro. El primer evento de cada secuencia de registros tiene el número de secuencia 0x0. No debe haber espacios en los valores de `Sequence No`. El número de secuencia solamente es único dentro de una secuencia de registros.

Command type (Tipo de comando)

Valor hexadecimal que representa la categoría del comando. Los comandos de las secuencias de registros de AWS CloudHSM tienen el tipo de comandos `CN_MGMT_CMD` (0x0) o `CN_CERT_AUTH_CMD` (0x9).

Opcode

Identifica el comando de administración ejecutado. Para obtener una lista de `Opcode` los valores de los registros de AWS CloudHSM auditoría, consulte Referencia del registro de auditoría de HSM.

Session handle (Identificador de sesión)

Identifica la sesión en la que se ejecutó el comando y se registró el evento.

Respuesta

Registra la respuesta del comando de administración. El campo `Response`, tendrá los valores `SUCCESS` y `ERROR`.

Tipo de registro

Indica el tipo de AWS CloudHSM registro del registro que registró el comando.

- `MINIMAL_LOG_ENTRY (0)`
- `MGMT_KEY_DETAILS_LOG (1)`
- `MGMT_USER_DETAILS_LOG (2)`
- `GENERIC_LOG`

Ejemplos de eventos de registro de auditoría

Los eventos de una secuencia de registros recopilan el historial del HSM, desde su creación hasta su eliminación. Puede utilizar el registro para revisar el ciclo de vida de los HSM y obtener información detallada acerca de su funcionamiento. Cuando interprete los eventos, tenga en cuenta el valor de `Opcode`, que indica la acción o el comando de administración, y el valor de `Sequence No`, que indica el orden de los eventos.

Temas

- Ejemplo: Inicialización del primer HSM de un clúster
- Eventos de inicio y cierre de sesión
- Ejemplo: Creación y eliminación de usuarios
- Ejemplo: Creación y eliminación de un par de claves
- Ejemplo: Creación y sincronización de una clave
- Ejemplo: Exportación de una clave
- Ejemplo: Importación de una clave
- Ejemplo: Compartir y dejar de compartir una clave

Ejemplo: Inicialización del primer HSM de un clúster

La secuencia de registros del primer HSM de cada clúster es muy diferente de la de los demás HMS del clúster. El registro de auditoría del primer HSM de cada clúster recopila su creación e

inicialización. Los registros de otros HSM del clúster generados a partir de copias de seguridad comienzan con un evento de inicio de sesión.

> **Important**
>
> Las siguientes entradas de inicialización no aparecerán en los CloudWatch registros de los clústeres inicializados antes del lanzamiento de la función de registro de auditorías de CloudHSM (30 de agosto de 2018). Para obtener más información, consulte el Historial de revisión.

Los siguientes eventos de ejemplo aparecen en la secuencia de registros del primer HSM de un clúster. El primer evento del registro, el que tiene el valor de `Sequence No 0x0`, representa el comando para inicializar el HSM (`CN_INIT_TOKEN`). La respuesta indica que el comando se ha realizado correctamente (`Response : 0: HSM Return: SUCCESS`).

```
Time: 12/19/17 21:01:16.962174, usecs:1513717276962174
Sequence No : 0x0
Reboot counter : 0xe8
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_INIT_TOKEN (0x1)
Session Handle : 0x1004001
Response : 0:HSM Return: SUCCESS
Log type : MINIMAL_LOG_ENTRY (0)
```

El segundo evento de esta secuencia de registros de ejemplo (`Sequence No 0x1`) registra el comando para crear la clave de cifrado de contraseñas que el HSM utiliza (`CN_GEN_PSWD_ENC_KEY`).

Esta secuencia de inicio es habitual entre los primeros HSM de cada clúster. Como los siguientes HSM del mismo clúster son clones del primero, utilizan la misma clave de cifrado de contraseñas.

```
Time: 12/19/17 21:01:17.140812, usecs:1513717277140812
Sequence No : 0x1
Reboot counter : 0xe8
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_GEN_PSWD_ENC_KEY (0x1d)
Session Handle : 0x1004001
Response : 0:HSM Return: SUCCESS
Log type : MINIMAL_LOG_ENTRY (0)
```

El tercer evento de esta secuencia de registros de ejemplo (`Sequence No 0x2`) es la creación del usuario de dispositivos (AU), que es el servicio AWS CloudHSM . Los eventos que implican a usuarios de HSM contienen campos adicionales para el nombre de usuario y el tipo de usuario.

```
Time: 12/19/17 21:01:17.174902, usecs:1513717277174902
Sequence No : 0x2
Reboot counter : 0xe8
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_CREATE_APPLIANCE_USER (0xfc)
Session Handle : 0x1004001
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : app_user
User Type : CN_APPLIANCE_USER (5)
```

El cuarto evento de esta secuencia de registros de ejemplo (`Sequence No 0x3`) registra el evento `CN_INIT_DONE`, que completa la inicialización del HSM.

```
Time: 12/19/17 21:01:17.298914, usecs:1513717277298914
Sequence No : 0x3
Reboot counter : 0xe8
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_INIT_DONE (0x95)
Session Handle : 0x1004001
Response : 0:HSM Return: SUCCESS
Log type : MINIMAL_LOG_ENTRY (0)
```

Puede seguir el resto de eventos en la secuencia de inicio. Estos eventos podrían incluir varios eventos de inicio y cierre de sesión y la generación de la clave de cifrado de claves (KEK). El siguiente evento registra el comando que cambia la contraseña del responsable de criptografía previa (PRECO). Este comando activa el clúster.

```
Time: 12/13/17 23:04:33.846554, usecs:1513206273846554
Sequence No: 0x1d
Reboot counter: 0xe8
Command Type(hex): CN_MGMT_CMD (0x0)
Opcode: CN_CHANGE_PSWD (0x9)
Session Handle: 0x2010003
Response: 0:HSM Return: SUCCESS
Log type: MGMT_USER_DETAILS_LOG (2)
User Name: admin
```

```
User Type: CN_CRYPTO_PRE_OFFICER (6)
```

Eventos de inicio y cierre de sesión

Cuando interprete el registro de auditoría, tenga en cuenta los eventos que registran inicios y cierres de sesión de los usuarios en HSM. Estos eventos le ayudan a determinar qué usuario es responsable de los comandos de administración de usuario que aparecen en orden entre los comandos de inicio y cierre de sesión.

Por ejemplo, esta entrada del registro recopila un inicio de sesión de un responsable de criptografía llamado `admin`. El número de secuencia, `0x0`, indica que este es el primer evento de esta secuencia de registros.

Cuando un usuario inicia sesión en un HSM, los demás HSM del clúster también registran un evento de inicio de sesión para el usuario. Puede buscar los eventos de inicio de sesión correspondientes en las secuencias de registros de otros HSM del clúster poco después del evento de inicio de sesión inicial.

```
Time: 01/16/18 01:48:49.824999, usecs:1516067329824999
Sequence No : 0x0
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_LOGIN (0xd)
Session Handle : 0x7014006
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : admin
User Type : CN_CRYPTO_OFFICER (2)
```

El siguiente evento de ejemplo recopila el cierre de sesión del responsable de criptografía `admin`. El número de secuencia, `0x2`, indica que este es el tercer evento de esta secuencia de registros.

Si el usuario conectado cierra la sesión sin desconectarse, la secuencia de registros contendrá un evento `CN_APP_FINALIZE` o de cierre de sesión (`CN_SESSION_CLOSE`) en lugar de un evento `CN_LOGOUT`. A diferencia del evento de inicio de sesión, este cierre de sesión solo suele registrarse en el HSM que ejecuta el comando.

```
Time: 01/16/18 01:49:55.993404, usecs:1516067395993404
Sequence No : 0x2
Reboot counter : 0x107
```

```
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_LOGOUT (0xe)
Session Handle : 0x7014000
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : admin
User Type : CN_CRYPTO_OFFICER (2)
```

Si se produce un error al intentar iniciar sesión porque el nombre de usuario no es válido, el HSM registra un evento `CN_LOGIN` con el nombre y el tipo de usuario proporcionados en el comando de inicio de sesión. La respuesta mostrará el mensaje de error 157, que indica que el nombre de usuario no existe.

```
Time: 01/24/18 17:41:39.037255, usecs:1516815699037255
Sequence No : 0x4
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_LOGIN (0xd)
Session Handle : 0xc008002
Response : 157:HSM Error: user isn't initialized or user with this name doesn't exist
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : ExampleUser
User Type : CN_CRYPTO_USER (1)
```

Si se produce un error al intentar iniciar sesión porque la contraseña no es válida, el HSM registra un evento `CN_LOGIN` con el nombre y el tipo de usuario proporcionados en el comando de inicio de sesión. La respuesta muestra el mensaje de error con el código `RET_USER_LOGIN_FAILURE`.

```
Time: 01/24/18 17:44:25.013218, usecs:1516815865013218
Sequence No : 0x5
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_LOGIN (0xd)
Session Handle : 0xc008002
Response : 163:HSM Error: RET_USER_LOGIN_FAILURE
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : testuser
User Type : CN_CRYPTO_USER (1)
```

Ejemplo: Creación y eliminación de usuarios

En este ejemplo, se muestran los eventos de registro que se escriben cuando un responsable de criptografía (CO) crea y elimina usuarios.

El primer evento registra un CO, `admin`, que inicia sesión en el HSM. El número de secuencia, `0x0`, indica que se trata del primer evento de la secuencia de registros. El nombre y el tipo de usuario que ha iniciado sesión se incluyen en el evento.

```
Time: 01/16/18 01:48:49.824999, usecs:1516067329824999
Sequence No : 0x0
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_LOGIN (0xd)
Session Handle : 0x7014006
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : admin
User Type : CN_CRYPTO_OFFICER (2)
```

El siguiente evento de la secuencia de registros (con el número `0x1`) registra el CO que crea un nuevo usuario de criptografía (CU). El nombre y el tipo del nuevo usuario se incluyen en el evento.

```
Time: 01/16/18 01:49:39.437708, usecs:1516067379437708
Sequence No : 0x1
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_CREATE_USER (0x3)
Session Handle : 0x7014006
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : bob
User Type : CN_CRYPTO_USER (1)
```

A continuación, el CO crea otro responsable de criptografía, `alice`. El número de secuencia indica que esta acción va detrás de la anterior sin que haya ninguna acción en medio.

```
Time: 01/16/18 01:49:55.993404, usecs:1516067395993404
Sequence No : 0x2
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_CREATE_CO (0x4)
```

```
Session Handle : 0x7014007
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : alice
User Type : CN_CRYPTO_OFFICER (2)
```

Después, el CO `admin` inicia sesión y elimina al responsable de criptografía llamado `alice`. El HSM registra un evento `CN_DELETE_USER`. El nombre y el tipo del usuario eliminado se incluyen en el evento.

```
Time: 01/23/18 19:58:23.451420, usecs:1516737503451420
Sequence No : 0xb
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_DELETE_USER (0xa1)
Session Handle : 0x7014007
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : alice
User Type : CN_CRYPTO_OFFICER (2)
```

Ejemplo: Creación y eliminación de un par de claves

En este ejemplo, se muestran los eventos que se registran en un registro de auditoría de HSM al crear y eliminar un par de claves.

El siguiente evento registra el usuario de criptografía (CU) `crypto_user`, que inicia sesión en la HSM.

```
Time: 12/13/17 23:09:04.648952, usecs:1513206544648952
Sequence No: 0x28
Reboot counter: 0xe8
Command Type(hex): CN_MGMT_CMD (0x0)
Opcode: CN_LOGIN (0xd)
Session Handle: 0x2014005
Response: 0:HSM Return: SUCCESS
Log type: MGMT_USER_DETAILS_LOG (2)
User Name: crypto_user
User Type: CN_CRYPTO_USER (1)
```

A continuación, el CU genera un par de claves (`CN_GENERATE_KEY_PAIR`). La clave privada tiene el identificador de clave `131079`. La clave pública tiene el identificador de clave `131078`.

```
Time: 12/13/17 23:09:04.761594, usecs:1513206544761594
Sequence No: 0x29
Reboot counter: 0xe8
Command Type(hex): CN_MGMT_CMD (0x0)
Opcode: CN_GENERATE_KEY_PAIR (0x19)
Session Handle: 0x2014004
Response: 0:HSM Return: SUCCESS
Log type: MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle: 131079
Public Key Handle: 131078
```

El CU elimina inmediatamente el par de claves. Un evento CN_DESTROY_OBJECT registra la eliminación de la clave pública (131078).

```
Time: 12/13/17 23:09:04.813977, usecs:1513206544813977
Sequence No: 0x2a
Reboot counter: 0xe8
Command Type(hex): CN_MGMT_CMD (0x0)
Opcode: CN_DESTROY_OBJECT (0x11)
Session Handle: 0x2014004
Response: 0:HSM Return: SUCCESS
Log type: MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle: 131078
Public Key Handle: 0
```

Después, un segundo evento `CN_DESTROY_OBJECT` registra la eliminación de la clave privada (`131079`).

```
Time: 12/13/17 23:09:04.815530, usecs:1513206544815530
Sequence No: 0x2b
Reboot counter: 0xe8
Command Type(hex): CN_MGMT_CMD (0x0)
Opcode: CN_DESTROY_OBJECT (0x11)
Session Handle: 0x2014004
Response: 0:HSM Return: SUCCESS
Log type: MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle: 131079
Public Key Handle: 0
```

Por último, el CU cierra la sesión.

```
Time: 12/13/17 23:09:04.817222, usecs:1513206544817222
Sequence No: 0x2c
Reboot counter: 0xe8
Command Type(hex): CN_MGMT_CMD (0x0)
Opcode: CN_LOGOUT (0xe)
Session Handle: 0x2014004
Response: 0:HSM Return: SUCCESS
Log type: MGMT_USER_DETAILS_LOG (2)
User Name: crypto_user
User Type: CN_CRYPTO_USER (1)
```

Ejemplo: Creación y sincronización de una clave

En este ejemplo se muestra el efecto de crear una clave en un clúster con varias HSM. La clave se genera en un HSM, se extrae del HSM como un objeto enmascarado y se inserta así en los demás HSM.

> **Note**
>
> Las herramientas de cliente podrían dar error al sincronizar la clave. O bien el comando podría incluir el parámetro min_srv, que sincroniza la clave únicamente en el número de HSM especificado. En cualquier caso, el AWS CloudHSM servicio sincroniza la clave con los demás HSM del clúster. Como los HSM solamente registran los comandos de administración del cliente en sus registros, la sincronización del servidor no se escribe en el registro de HSM.

En primer lugar, observe la secuencia de registros del HSM que recibe y ejecuta los comandos. La secuencia de registro tiene como nombre el ID del HSM, `hsm-abcde123456`, pero este ID no aparece en los eventos de registro.

Primero, el usuario de criptografía (CU) `testuser` inicia sesión en el HSM `hsm-abcde123456`.

```
Time: 01/24/18 00:39:23.172777, usecs:1516754363172777
Sequence No : 0x0
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_LOGIN (0xd)
Session Handle : 0xc008002
Response : 0:HSM Return: SUCCESS
```

```
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : testuser
User Type : CN_CRYPTO_USER (1)
```

La CU ejecuta un exSymKeycomando para generar una clave simétrica. El HSM `hsm-abcde123456` genera una clave simétrica con el identificador de clave `262152`. El HSM registra un evento `CN_GENERATE_KEY` en su registro.

```
Time: 01/24/18 00:39:30.328334, usecs:1516754370328334
Sequence No : 0x1
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_GENERATE_KEY (0x17)
Session Handle : 0xc008004
Response : 0:HSM Return: SUCCESS
Log type : MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle : 262152
Public Key Handle : 0
```

El siguiente evento de la secuencia de registros de `hsm-abcde123456` registra el primer paso del proceso de sincronización. La nueva clave (identificador de clave `262152`) se extrae del HSM como un objeto enmascarado.

```
Time: 01/24/18 00:39:30.330956, usecs:1516754370330956
Sequence No : 0x2
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_EXTRACT_MASKED_OBJECT_USER (0xf0)
Session Handle : 0xc008004
Response : 0:HSM Return: SUCCESS
Log type : MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle : 262152
Public Key Handle : 0
```

Ahora, observe la secuencia de registros del HSM `hsm-zyxwv987654`, otro HSM del mismo clúster. Esta secuencia de registros también contiene un evento de inicio de sesión del CU `testuser`. El valor de la hora indica que tuvo lugar poco después de que el usuario iniciara sesión en el HSM `hsm-abcde123456`.

```
Time: 01/24/18 00:39:23.199740, usecs:1516754363199740
```

```
Sequence No : 0xd
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_LOGIN (0xd)
Session Handle : 0x7004004
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : testuser
User Type : CN_CRYPTO_USER (1)
```

La secuencia de registros de este HSM no tiene ningún evento `CN_GENERATE_KEY`. Sin embargo, tiene un evento que registra la sincronización de la clave con este HSM. El evento `CN_INSERT_MASKED_OBJECT_USER` registra la recepción de la clave `262152` como un objeto enmascarado. Ahora, la clave `262152` existe en los dos HSM del clúster.

```
Time: 01/24/18 00:39:30.408950, usecs:1516754370408950
Sequence No : 0xe
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_INSERT_MASKED_OBJECT_USER (0xf1)
Session Handle : 0x7004003
Response : 0:HSM Return: SUCCESS
Log type : MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle : 262152
Public Key Handle : 0
```

Cuando el usuario CU cierra sesión, este evento `CN_LOGOUT` solamente aparece en la secuencia de registros del HSM que recibió los comandos.

Ejemplo: Exportación de una clave

En este ejemplo, se muestran los eventos de registro de auditoría que se registran cuando un usuario de criptografía (CU) exporta las claves de un clúster con varios HSM.

El evento siguiente registra el CU (`testuser`), que inicia sesión en key_mgmt_util.

```
Time: 01/24/18 19:42:22.695884, usecs:1516822942695884
Sequence No : 0x26
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_LOGIN (0xd)
```

```
Session Handle : 0x7004004
Response : 0:HSM Return: SUCCESS
Log type : MGMT_USER_DETAILS_LOG (2)
User Name : testuser
User Type : CN_CRYPTO_USER (1)
```

La CU ejecuta un exSymKeycomando para exportar la clave7, una clave AES de 256 bits. El comando utiliza la clave 6, que es una clave AES de 256 bits de los HSM, como clave de encapsulamiento.

El HSM que recibe el comando registra un evento CN_WRAP_KEY de la clave 7, la clave que se va a exportar.

```
Time: 01/24/18 19:51:12.860123, usecs:1516823472860123
Sequence No : 0x27
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_WRAP_KEY (0x1a)
Session Handle : 0x7004003
Response : 0:HSM Return: SUCCESS
Log type : MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle : 7
Public Key Handle : 0
```

A continuación, el HSM registra un evento CN_NIST_AES_WRAP para la clave de encapsulamiento, la clave 6. La clave se encapsula y de inmediato se desencapsula de nuevo, pero el HSM solamente registra un evento.

```
Time: 01/24/18 19:51:12.905257, usecs:1516823472905257
Sequence No : 0x28
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_NIST_AES_WRAP (0x1e)
Session Handle : 0x7004003
Response : 0:HSM Return: SUCCESS
Log type : MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle : 6
Public Key Handle : 0
```

El comando exSymKey escribe la clave exportada en un archivo, pero no cambia la clave del HSM. Por tanto, no habrá eventos correspondientes en los registros de otros HSM del clúster.

Ejemplo: Importación de una clave

En este ejemplo, se muestran los eventos de registro de auditoría que se escriben cuando se importan claves en los HSM de un clúster. En este ejemplo, el usuario criptográfico (CU) utiliza el imSymKeycomando para importar una clave AES a los HSM. El comando utiliza la clave 6 como clave de encapsulamiento.

El HSM que recibe el comando primero registra un evento `CN_NIST_AES_WRAP` para la clave 6, la clave de encapsulamiento.

```
Time: 01/24/18 19:58:23.170518, usecs:1516823903170518
Sequence No : 0x29
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_NIST_AES_WRAP (0x1e)
Session Handle : 0x7004003
Response : 0:HSM Return: SUCCESS
Log type : MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle : 6
Public Key Handle : 0
```

A continuación, el HSM registra un evento `CN_UNWRAP_KEY`, que representa la operación de importación. A la clave importada se le asigna el identificador de clave 11.

```
Time: 01/24/18 19:58:23.200711, usecs:1516823903200711
Sequence No : 0x2a
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_UNWRAP_KEY (0x1b)
Session Handle : 0x7004003
Response : 0:HSM Return: SUCCESS
Log type : MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle : 11
Public Key Handle : 0
```

Cuando se genera o importa una nueva clave, las herramientas del cliente intentan sincronizar automáticamente la nueva clave con otros HSM del clúster. En este caso, el HSM registra un evento `CN_EXTRACT_MASKED_OBJECT_USER` cuando la clave 11 se extrae del HSM como un objeto enmascarado.

```
Time: 01/24/18 19:58:23.203350, usecs:1516823903203350
```

```
Sequence No : 0x2b
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_EXTRACT_MASKED_OBJECT_USER (0xf0)
Session Handle : 0x7004003
Response : 0:HSM Return: SUCCESS
Log type : MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle : 11
Public Key Handle : 0
```

La secuencia de registros de otros HSM del clúster refleja la llegada de la nueva clave importada.

Por ejemplo, este evento se registró en la secuencia de registros de otro HSM del mismo clúster. Este evento `CN_INSERT_MASKED_OBJECT_USER` registra la llegada de un objeto enmascarado que representa la clave 11.

```
Time: 01/24/18 19:58:23.286793, usecs:1516823903286793
Sequence No : 0xb
Reboot counter : 0x107
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_INSERT_MASKED_OBJECT_USER (0xf1)
Session Handle : 0xc008004
Response : 0:HSM Return: SUCCESS
Log type : MGMT_KEY_DETAILS_LOG (1)
Priv/Secret Key Handle : 11
Public Key Handle : 0
```

Ejemplo: Compartir y dejar de compartir una clave

En este ejemplo, se muestra el evento del registro de auditoría que se genera cuando un usuario de criptografía (CU) comparte o deja de compartir una clave privada de ECC con otros usuarios de criptografía. El CU utiliza el comando shareKey y proporciona el identificador de clave, el ID de usuario y el valor 1 para compartir la clave o el valor `0` para dejar de compartirla.

En el siguiente ejemplo, el HSM que recibe el comando registra un evento `CM_SHARE_OBJECT` que representa la operación de compartición.

```
Time: 02/08/19 19:35:39.480168, usecs:1549654539480168
Sequence No : 0x3f
Reboot counter : 0x38
Command Type(hex) : CN_MGMT_CMD (0x0)
Opcode : CN_SHARE_OBJECT (0x12)
```

```
Session Handle : 0x3014007
Response : 0:HSM Return: SUCCESS
Log type : UNKNOWN_LOG_TYPE (5)
```

Referencia del registro de auditoría de HSM

AWS CloudHSM registra los comandos de administración de HSM en los eventos del registro de auditoría. Cada evento tiene un valor de código de operación (`0pcode`) que identifica la acción que se ha producido y su respuesta. Puede utilizar los valores de `0pcode` para buscar, clasificar y filtrar los registros.

La siguiente tabla define los `0pcode` valores de un registro de AWS CloudHSM auditoría.

Código de operación (Opcode)	Descripción
Inicio de sesión del usuario: estos eventos incluyen el nombre y el tipo de usuario.	
`CN_LOGIN (0xd)`	Inicio de sesión de usuario
`CN_LOGOUT (0xe)`	Cierre de sesión de usuario
`CN_APP_FINALIZE`	Se cerró la conexión con el HSM. Se eliminaron todas las claves de sesión o los símbolos de quórum de esta conexión.
`CN_CLOSE_SESSION`	Se cerró la sesión con el HSM. Se eliminaron todas las claves de sesión o los símbolos de quórum de esta sesión.
Administración del usuario: estos eventos incluyen el nombre y el tipo de usuario.	
`CN_CREATE_USER (0x3)`	Creación de un usuario de criptografía (CU).
`CN_CREATE_CO`	Creación de responsable de criptografía.
`CN_DELETE_USER`	Eliminar un usuario
`CN_CHANGE_PSWD`	Cambio de la contraseña de un usuario.
`CN_SET_M_VALUE`	Establezca la autenticación de quórum (M de N) para una acción del usuario

Código de operación (Opcode)	Descripción
CN_APPROVE_TOKEN	Apruebe un token de autenticación de quórum para una acción de usuario
CN_DELETE_TOKEN	Elimine uno o más tokens de quórum
CN_GET_TOKEN	Solicita un token de firma para iniciar una operación de quórum
Administración de claves: estos eventos contienen el identificador de la clave.	
CN_GENERATE_KEY	Generación de una clave simétrica.
CN_GENERATE_KEY_PAIR (0x19)	Generar un key pair asimétrico
CN_CREATE_OBJECT	Importación de una clave pública (sin encapsulamiento)
CN_MODIFY_OBJECT	Establezca un atributo clave
CN_DESTROY_OBJECT (0x11)	Eliminación de una clave de sesión
CN_TOMBSTONE_OBJECT	Eliminación de una clave simbólica
CN_SHARE_OBJECT	Compartir o dejar de compartir una clave.
CN_WRAP_KEY	Exportar una copia cifrada de una clave (wrapKey).
CN_UNWRAP_KEY	Importar una copia cifrada de una clave (unwrapKey).
CN_DERIVE_KEY	Derive una clave simétrica a partir de una clave existente
CN_NIST_AES_WRAP	Cifre o descifre una clave con una clave AES
CN_INSERT_MASKED_OBJECT_USER	Inserte una clave cifrada con atributos de otro HSM del clúster.

Código de operación (Opcode)	Descripción
CN_EXTRACT_MASKED_OBJECT_USER	Envuelve o cifra una clave con atributos del HSM para enviarla a otro HSM del clúster.
Back up HSMs	
CN_BACKUP_BEGIN	Comience el proceso de copia de seguridad
CN_BACKUP_END	Se completó el proceso de respaldo
CN_RESTORE_BEGIN	Comience a restaurar desde una copia de seguridad
CN_RESTORE_END	Se completó el proceso de restauración a partir de una copia de seguridad
Certificate-Based Authentication	
CN_CERT_AUTH_STORE_CERT	Almacena el certificado del clúster
HSM Instance Commands	
CN_INIT_TOKEN (0x1)	Inicie el proceso de inicialización del HSM
CN_INIT_DONE	El proceso de inicialización del HSM ha finalizado
CN_GEN_KEY_ENC_KEY	Generación de una clave de cifrado de claves (KEK).
CN_GEN_PSWD_ENC_KEY (0x1d)	Generación de una clave de cifrado de contraseñas (PEK).
HSM crypto commands	
CN_FIPS_RAND	Genere un número aleatorio compatible con FIPS

Obtener CloudWatch métricas para AWS CloudHSM

CloudWatch Úselo para monitorear su AWS CloudHSM clúster en tiempo real. Las métricas se pueden agrupar por región, por ID de clúster o por ID de clúster y por ID de HSM.

El espacio de nombres de `AWS/CloudHSM` incluye las siguientes métricas:

Métrica	Descripción
HsmUnhealthy	La instancia de HSM no funciona correctamente. AWS CloudHSM reemplaza automáticamente las instancias en mal estado. Puede optar por ampliar de forma proactiva el tamaño del clúster para reducir el impacto en el rendimien to mientras se sustituye el HSM.
HsmTemper ature	Temperatura del empalme del procesador de hardware. El sistema se apaga si la temperatura alcanza 110 grados centígrados.
HsmKeysSe ssionOccupied	Cantidad de claves de sesión que la instancia de HSM está utilizando.
HsmKeysTo kenOccupied	La cantidad de claves de token que la instancia de HSM y el clúster están utilizando.
HsmSslCtx sOccupied	El número de canales end-to-end cifrados actualmente establecidos para la instancia de HSM. Se permiten hasta 2048 canales.
HsmSessio nCount	Cantidad de conexiones abiertas a la instancia de HSM. Se permiten hasta 2048. De forma predeterminada, el daemon del cliente está configurado para abrir dos sesiones con cada instancia de HSM en un end-to-end canal cifrado. AWS CloudHSM también puede tener hasta 2 conexiones abiertas con el HSM para supervisar el estado de los HSM.
HsmUsersA vailable	Cantidad de usuarios adicionales que se pueden crear. Esto equivale al número máximo de usuarios (indicado en HsmUsersMax) menos los usuarios creados hasta la fecha.
HsmUsersMax	El máximo número de usuarios que se pueden crear en la instancia de HSM. La cantidad actual de usuarios es 1024.

Métrica	Descripción
InterfaceEth2 OctetsInput	La suma acumulada del tráfico entrante al HSM hasta la fecha.
InterfaceEth2 OctetsOutput	La suma acumulada del tráfico saliente al HSM hasta la fecha.

AWS CloudHSM Rendimiento

En el caso de los clústeres de producción, debe tener al menos dos instancias de HSM distribuidas en diferentes zonas de disponibilidad de una región. Recomendamos realizar pruebas de carga en el clúster para determinar el pico de carga previsto y, a continuación, añadir un HSM adicional para garantizar una alta disponibilidad. En el caso de las aplicaciones que requieren la durabilidad de las claves recién generadas, le recomendamos al menos tres instancias de HSM distribuidas en todas las zonas de disponibilidad de una región.

Datos de rendimiento

El rendimiento de los AWS CloudHSM clústeres varía en función de la carga de trabajo específica. En la siguiente tabla se muestra el rendimiento aproximado de los algoritmos criptográficos más comunes que se ejecutan en una instancia de EC2. Para aumentar el rendimiento, puede añadir instancias de HSM adicionales a sus clústeres. El rendimiento puede variar en función de la configuración, el tamaño de los datos y la carga adicional de aplicaciones en las instancias de EC2. Recomendamos realizar pruebas de carga de su aplicación para determinar las necesidades de escalado.

Operación	Clúster de dos HSM[1]	Clúster de tres HSM[2]	Clúster de seis HSM[3]
Firma RSA de 2048 bits	2000 operaciones/s	3000 operaciones/s	5000 operaciones/s
Firma EC P256	500 operaciones/s	750 operaciones/s	1500 operaciones/s

- [1] Un clúster de dos HSM con aplicación de procesamiento múltiple Java que se ejecuta en una instancia de EC2 c4.large y un HSM en la misma zona de disponibilidad que la instancia de EC2.
- [2] Un clúster de tres HSM con aplicación de procesamiento múltiple Java que se ejecuta en una instancia de EC2 c4.large y un HSM en la misma zona de disponibilidad que la instancia de EC2.
- [3] Un clúster de seis HSM con aplicación de procesamiento múltiple Java que se ejecuta en una instancia de EC2 c4.large y un HSM en la misma zona de disponibilidad que la instancia de EC2.

Limitación de HSM

Cuando su carga de trabajo supere la capacidad de HSM de su clúster, recibirá mensajes de error indicando que los HSM están ocupados o limitados. Para obtener información detallada sobre qué hacer en este caso, consulte Limitación de HSM.

Seguridad en AWS CloudHSM

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada 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 conformidad aplicables AWS CloudHSM, consulte Servicios de AWS dentro del alcance por programa de conformidad .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS CloudHSM. Los siguientes temas muestran cómo configurarlo AWS CloudHSM para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros servicios de AWS que le ayudan a supervisar y proteger sus AWS CloudHSM recursos.

Contenido

- Protección de datos en AWS CloudHSM
- Administración de identidad y acceso para AWS CloudHSM
- Conformidad
- Resiliencia en AWS CloudHSM
- Seguridad de la infraestructura en AWS CloudHSM
- AWS CloudHSM y puntos finales de VPC
- Gestión de actualizaciones en AWS CloudHSM

Protección de datos en AWS CloudHSM

El modelo de responsabilidad AWS compartida modelo se aplica a la protección de datos en AWS CloudHSM. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con AWS CloudHSM o Servicios de AWS utilizando la consola, la API, la CLI o AWS los SDK. Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación

o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya la información de las credenciales en la URL para validar la solicitud para ese servidor.

Cifrado en reposo

Al realizar AWS CloudHSM una copia de seguridad desde un HSM, el HSM cifra sus datos antes de enviarlos a él. AWS CloudHSM Los datos se cifran mediante una clave de cifrado única y efímera. Para obtener más información, consulte AWS CloudHSM copias de seguridad en clúster.

Cifrado en tránsito

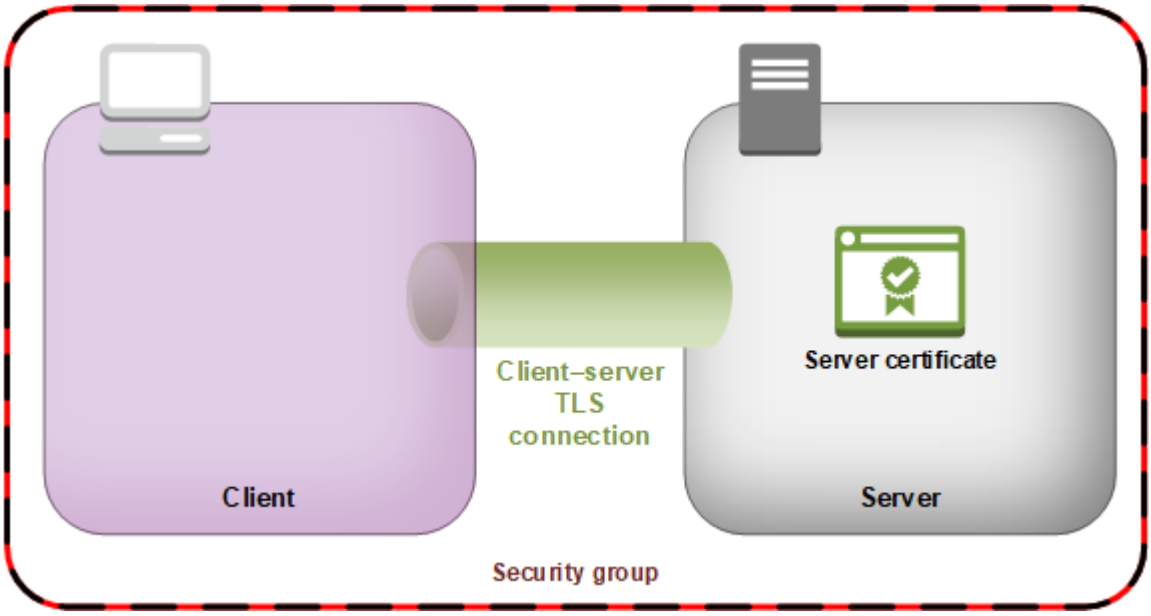
La comunicación entre el AWS CloudHSM cliente y el HSM del clúster se cifra de extremo a extremo. Esta comunicación solo la puede descifrar su cliente y sus HSM. Para obtener más información, consulte nd-to-end Cifrado electrónico.

AWS CloudHSM cifrado del cliente end-to-end

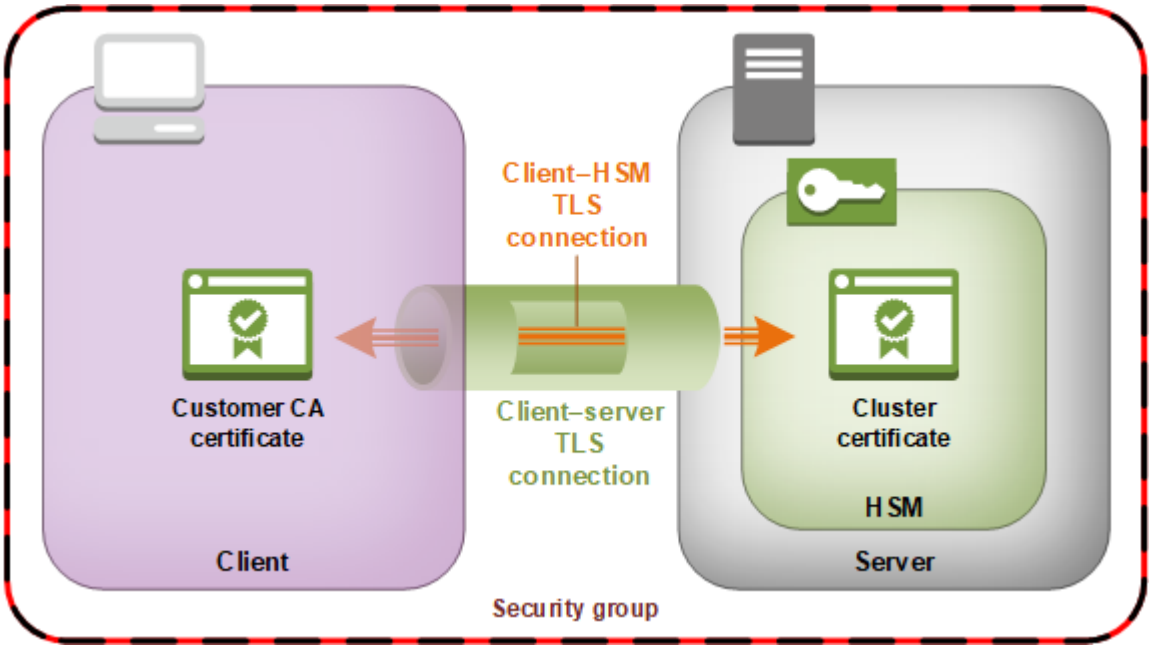
La comunicación entre la instancia del cliente y los HSM del clúster se realiza con cifrado integral. Solo pueden descifrarla el cliente y los HSM.

El siguiente proceso explica cómo el cliente establece una comunicación end-to-end cifrada con un HSM.

1. El cliente establece una conexión Transport Layer Security (TLS) con el servidor que aloja el hardware del HSM. El grupo de seguridad del clúster únicamente permite el tráfico entrante al servidor desde las instancias de cliente del grupo de seguridad. El cliente también comprueba el certificado del servidor para asegurarse de que es un servidor de confianza.



2. A continuación, el cliente establece una conexión cifrada con el hardware del HSM. El HSM tiene el certificado de clúster firmado por usted con su propia entidad de certificación (CA), y el cliente tiene el certificado raíz de la CA. Antes de que se establezca la conexión cifrada entre el cliente y el HSM, el cliente verifica el certificado de clúster del HSM con el certificado raíz. La conexión únicamente se establece cuando el cliente verifica correctamente que el HSM es de confianza.



Seguridad de las copias de seguridad del clúster

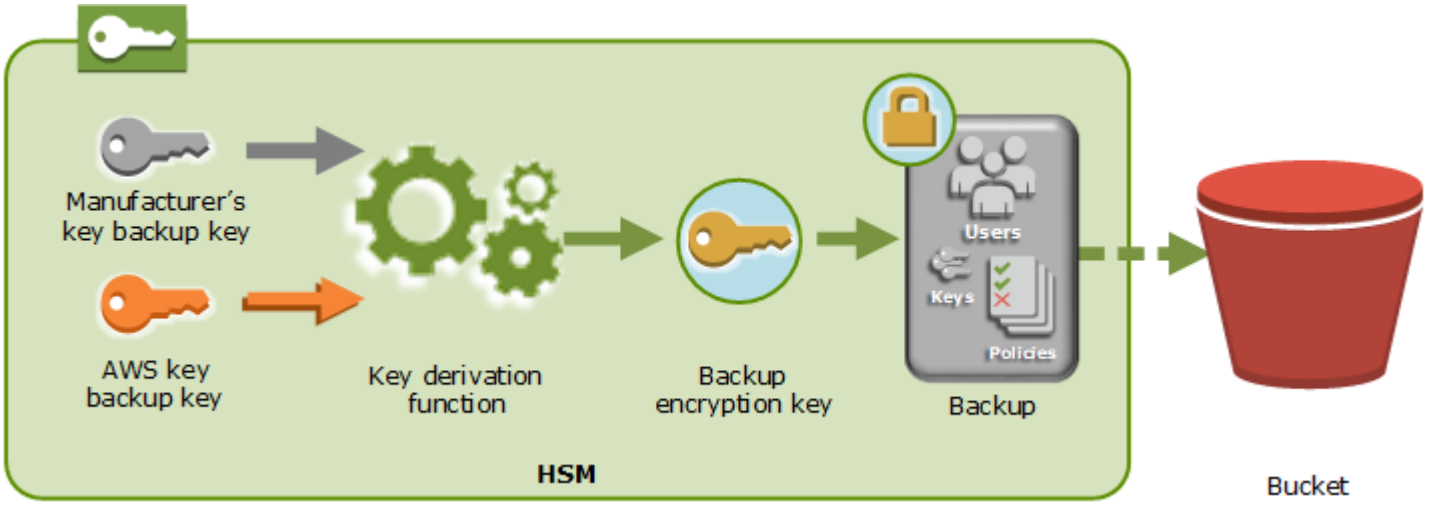
Cuando AWS CloudHSM hace una copia de seguridad desde el HSM, el HSM cifra todos sus datos antes de enviarlos a. AWS CloudHSM Los datos nunca salen del HSM en formato de texto no cifrado. Además, las copias de seguridad no se pueden descifrar AWS porque AWS no tiene acceso a la clave utilizada para descifrarlas.

Para cifrar los datos, el HSM utiliza una clave de cifrado única y efímera, conocida como la clave de backup efímera (EBK). La EBK es una clave de cifrado AES de 256 bits que se genera dentro del HSM al realizar una copia de seguridad. AWS CloudHSM El HSM genera la EBK y, a continuación, la utiliza para cifrar sus datos con un método de encapsulación de clave AES aprobado por FIPS que cumple la Publicación especial de NIST 800-38F. A continuación, el HSM proporciona los datos cifrados a. AWS CloudHSM Los datos cifrados contienen una copia cifrada de la EBK.

Para cifrar la EBK, el HSM utiliza otra clave de cifrado conocida como la clave de backup persistente (PBK). La PBK también es una clave de cifrado AES de 256 bits. Para generar la PBK, el HSM utiliza una función de derivación de clave (KDF) aprobada por FIPS en modo contador que cumple la Publicación especial de NIST 800-108. Los datos de entrada de esta KDF incluyen lo siguiente:

- Una clave de backup de clave de fabricante (MKBK), integrada permanentemente en el hardware del HSM por parte del fabricante.
- Una AWS clave clave de respaldo (AKBK), instalada de forma segura en el HSM cuando la configura inicialmente. AWS CloudHSM

Los procesos de cifrado se resumen en la siguiente figura. La clave de cifrado de backup representa la clave de backup persistente (PBK) y la clave de backup efímera (EBK).



AWS CloudHSM solo puede restaurar las copias de seguridad en los HSM de AWS propiedad del mismo fabricante. Dado que cada backup contiene todos los usuarios, claves y la configuración del HSM original, el HSM restaurado contiene las mismas protecciones y controles de acceso que el original. Los datos restaurados sobrescriben todos los demás datos que pudiera haber en el HSM antes de la restauración.

Un backup solo contiene datos cifrados. Antes de que el servicio almacene una copia de seguridad en Amazon S3, vuelve a cifrar la copia de seguridad mediante AWS Key Management Service (AWS KMS).

Administración de identidad y acceso para AWS CloudHSM

AWS utiliza credenciales de seguridad para identificarle y para concederle acceso a sus recursos de AWS. Puede utilizar las características de AWS Identity and Access Management (IAM) para permitir que otros usuarios, servicios y aplicaciones utilicen sus recursos de AWS de forma completa o limitada. Puede hacerlo sin compartir sus credenciales de seguridad.

De forma predeterminada, los usuarios de IAM no tienen permiso para crear, consultar ni modificar recursos de AWS. Para permitir que un usuario de IAM obtenga acceso a los recursos, como, por ejemplo, un equilibrador de carga, y realizar tareas, debe:

1. Crear una política de IAM que conceda permiso al usuario de IAM para utilizar los recursos específicos y las acciones de la API que necesita.
2. Asocie la política al usuario de IAM o al grupo al que pertenece el usuario de IAM.

Cuando se asocia una política a un usuario o grupo de usuarios, les otorga o deniega el permiso para realizar las tareas especificadas en los recursos indicados.

Por ejemplo, puede utilizar IAM para crear usuarios y grupos en su cuenta de AWS. Un usuario de IAM puede ser una persona, un sistema o una aplicación. A continuación, puede conceder permisos a los usuarios y grupos de tal forma que puedan llevar a cabo acciones concretas en determinados recursos especificados mediante una política de IAM.

Concesión de permisos mediante políticas de IAM

Cuando se asocia una política a un usuario o grupo de usuarios, les otorga o deniega el permiso para realizar las tareas especificadas en los recursos indicados.

Una política de IAM es un documento JSON que contiene una o varias instrucciones. Cada instrucción se estructura, tal y como se muestra en el siguiente ejemplo.

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect": "effect",
    "Action": "action",
    "Resource": "resource-arn",
    "Condition": {
      "condition": {
        "key":"value"
      }
    }
  }]
}
```

- Effect: el valor effect puede ser `Allow` o `Deny`. De forma predeterminada, los usuarios de IAM no tienen permiso para utilizar los recursos y las acciones de la API, por lo que se deniegan todas las solicitudes. Si se concede un permiso explícito se anula el valor predeterminado. Una denegación explícita invalida cualquier permiso concedido.
- Action: el valor de action es la acción de la API para la que concede o deniega permisos. Para obtener más información sobre cómo especificar action, consulte Acciones de la API para AWS CloudHSM.
- Recurso: el recurso al que afecta la acción. AWS CloudHSM no admite permisos a nivel de recurso. Debe utilizar el comodín * para especificar todos los recursos. AWS CloudHSM
- Condition: si lo desea, puede utilizar condiciones para controlar cuándo está en vigor la política. Para obtener más información, consulte Claves de condición para AWS CloudHSM.

Para obtener más información, consulte la Guía del usuario de IAM.

Acciones de la API para AWS CloudHSM

En el elemento Acción de su declaración de política de IAM, puede especificar cualquier acción de API que AWS CloudHSM ofrezca. El nombre de la acción debe llevar como prefijo la cadena en minúsculas `cloudhsm:`, tal y como se muestra en el ejemplo siguiente.

```
"Action": "cloudhsm:DescribeClusters"
```

Para especificar varias acciones en una misma instrucción, inclúyalas entre corchetes y sepárelas por comas, tal y como se muestra en el siguiente ejemplo.

```
"Action": [
    "cloudhsm:DescribeClusters",
    "cloudhsm:DescribeHsm"
]
```

También puede utilizar el carácter comodín * para especificar varias acciones. En el siguiente ejemplo, se especifican todos los nombres de las acciones de la API AWS CloudHSM que comienzan `List` por.

```
"Action": "cloudhsm:List*"
```

Para especificar todas las acciones de la API AWS CloudHSM, usa el comodín *, como se muestra en el siguiente ejemplo.

```
"Action": "cloudhsm:*"
```

Para ver la lista de acciones de la API para AWS CloudHSM, consulte AWS CloudHSM Acciones.

Claves de condición para AWS CloudHSM

Al crear una política, se pueden especificar las condiciones que controlan cuándo entra en vigor. Cada condición contiene uno o varios pares clave-valor. Existen claves de condición globales y claves de condición específicas del servicio.

AWS CloudHSM no tiene claves de contexto específicas del servicio.

Para obtener más información sobre las claves de condición globales, consulte Claves de contexto de condición de IAM y globales de AWS en la Guía del usuario de IAM.

Políticas administradas por AWS predefinidas para AWS CloudHSM

Las políticas administradas creadas por AWS conceden los permisos necesarios para casos de uso comunes. Puede asociar estas políticas a sus usuarios de IAM, en función del nivel de acceso que necesiten para AWS CloudHSM :

- AWSCloudHSMFullAccess— Otorga el acceso completo necesario para utilizar AWS CloudHSM las funciones.

- AWSCloudHSMReadOnlyAccess— Otorga acceso de solo lectura a las AWS CloudHSM funciones.

Políticas gestionadas por el cliente para AWS CloudHSM

Le recomendamos que cree un grupo de administradores de IAM AWS CloudHSM que contenga solo los permisos necesarios para ejecutarse AWS CloudHSM. Asocie la política con los permisos adecuados a este grupo. Puede agregar usuarios de IAM al grupo según sea necesario. Cada usuario que agregue hereda la política del grupo de administradores.

Además, le recomendamos que cree grupos de usuarios adicionales en función de los permisos que necesitan los usuarios. Esto garantiza que solo los usuarios de confianza tengan acceso a acciones críticas de la API. Por ejemplo, puede crear un grupo de usuarios que utilice para conceder acceso de solo lectura a clústeres y HSM. Dado que este grupo no permite a un usuario eliminar clústeres o HSM, un usuario que no sea de confianza no puede afectar a la disponibilidad de una carga de trabajo de producción.

A medida que se vayan añadiendo nuevas funciones de AWS CloudHSM administración con el tiempo, puede asegurarse de que solo los usuarios de confianza tengan acceso inmediato. Si asigna permisos limitados a las políticas en el momento de su creación, podrá asignarles manualmente los permisos de las nuevas características más adelante.

A continuación se muestran ejemplos de políticas para AWS CloudHSM. Para obtener información acerca de cómo crear una política y asociarla a un grupo de usuarios de IAM, consulte Creación de políticas en la pestaña JSON en la Guía del usuario de IAM.

Ejemplos

- Permisos de solo lectura
- Permisos de usuario avanzado
- Permisos de administrador

Example Ejemplo: permiso de solo lectura

Esta política permite el acceso a las acciones `DescribeClusters` y `DescribeBackups` de la API. También incluye permisos adicionales para acciones específicas de acciones de la API de Amazon EC2. No permite al usuario eliminar clústeres ni HSM.

```
{
   "Version": "2012-10-17",
   "Statement": {
      "Effect": "Allow",
      "Action": [
         "cloudhsm:DescribeClusters",
         "cloudhsm:DescribeBackups",
         "cloudhsm:ListTags"
      ],
      "Resource": "*"
   }
}
```

Example Ejemplo: permisos de usuario avanzado

Esta política permite el acceso a un subconjunto de las acciones de la AWS CloudHSM API. También incluye permisos adicionales para acciones de la API de Amazon EC2 específicas. No permite al usuario eliminar clústeres ni HSM. Debe incluir la `iam:CreateServiceLinkedRole` acción para AWS CloudHSM permitir la creación automática del rol AWSServiceRoleForCloudHSMvinculado al servicio en su cuenta. Este rol permite AWS CloudHSM registrar eventos. Para obtener más información, consulte Funciones vinculadas al servicio para AWS CloudHSM.

> **Note**
>
> Para ver los permisos específicos de cada API, consulte Acciones, recursos y claves de condición para AWS CloudHSM en la Referencia de autorizaciones de servicio.

```
{
   "Version": "2012-10-17",
   "Statement": {
      "Effect": "Allow",
      "Action": [
         "cloudhsm:DescribeClusters",
         "cloudhsm:DescribeBackups",
         "cloudhsm:CreateCluster",
         "cloudhsm:CreateHsm",
         "cloudhsm:RestoreBackup",
         "cloudhsm:CopyBackupToRegion",
         "cloudhsm:InitializeCluster",
```

```
            "cloudhsm:ListTags",
            "cloudhsm:TagResource",
            "cloudhsm:UntagResource",
            "ec2:CreateNetworkInterface",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeNetworkInterfaceAttribute",
            "ec2:DetachNetworkInterface",
            "ec2:DeleteNetworkInterface",
            "ec2:CreateSecurityGroup",
            "ec2:AuthorizeSecurityGroupIngress",
            "ec2:AuthorizeSecurityGroupEgress",
            "ec2:RevokeSecurityGroupEgress",
            "ec2:DescribeSecurityGroups",
            "ec2:DeleteSecurityGroup",
            "ec2:CreateTags",
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "iam:CreateServiceLinkedRole"
        ],
        "Resource": "*"
    }
}
```

Example Ejemplo: permisos de administrador

Esta política permite el acceso a todas las acciones de la AWS CloudHSM API, incluidas las acciones para eliminar los HSM y los clústeres. También incluye permisos adicionales para acciones de la API de Amazon EC2 específicas. Debe incluir la `iam:CreateServiceLinkedRole` acción para AWS CloudHSM permitir la creación automática del rol AWSServiceRoleForCloudHSMvinculado al servicio en su cuenta. Este rol permite AWS CloudHSM registrar eventos. Para obtener más información, consulte Funciones vinculadas al servicio para AWS CloudHSM.

```
{
   "Version":"2012-10-17",
   "Statement":{
      "Effect":"Allow",
      "Action":[
         "cloudhsm:*",
         "ec2:CreateNetworkInterface",
         "ec2:DescribeNetworkInterfaces",
         "ec2:DescribeNetworkInterfaceAttribute",
```

```
            "ec2:DetachNetworkInterface",
            "ec2:DeleteNetworkInterface",
            "ec2:CreateSecurityGroup",
            "ec2:AuthorizeSecurityGroupIngress",
            "ec2:AuthorizeSecurityGroupEgress",
            "ec2:RevokeSecurityGroupEgress",
            "ec2:DescribeSecurityGroups",
            "ec2:DeleteSecurityGroup",
            "ec2:CreateTags",
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "iam:CreateServiceLinkedRole"
        ],
        "Resource":"*"
    }
}
```

Funciones vinculadas al servicio para AWS CloudHSM

La política de IAM que creó anteriormente Políticas gestionadas por el cliente para AWS CloudHSM incluye la acción. `iam:CreateServiceLinkedRole` AWS CloudHSM define un rol vinculado al servicio denominado. AWSServiceRoleForCloudHSM El rol está predefinido AWS CloudHSM e incluye los permisos necesarios para AWS CloudHSM llamar a otros AWS servicios en su nombre. El rol facilita la configuración del servicio, ya que no es necesario agregar manualmente los permisos de las políticas de rol y de confianza.

La política de roles permite AWS CloudHSM crear grupos de CloudWatch registros y flujos de registros de Amazon Logs y escribir eventos de registro en su nombre. Puede verlo a continuación y en la consola de IAM.

```
{
    "Version": "2018-06-12",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
```

```
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        }
    ]
}
```

La política de confianza del AWSServiceRoleForCloudHSMrol AWS CloudHSM permite asumirlo.

```
{
  "Version": "2018-06-12",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudhsm.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Creación de un rol vinculado a un servicio (automático)

AWS CloudHSM crea el AWSServiceRoleForCloudHSMrol al crear un clúster si incluye la `iam:CreateServiceLinkedRole` acción en los permisos que definió al crear el grupo de AWS CloudHSM administradores. Consulte Políticas gestionadas por el cliente para AWS CloudHSM.

Si ya tienes uno o más clústeres y solo quieres añadir el AWSServiceRoleForCloudHSMrol, puedes usar la consola, el comando create-cluster o la operación de CreateClusterAPI para crear un clúster. A continuación, usa la consola, el comando delete-cluster o la operación de DeleteClusterAPI para eliminarlo. Al crear el nuevo clúster se crea la función vinculada al servicio y se aplica a todos los clústeres de la cuenta. También puede crear el rol manualmente. Consulte la siguiente sección para obtener más información.

> **Note**
>
> No necesita realizar todos los pasos descritos en la sección Empezar con AWS CloudHSM para crear un clúster si solo lo está creando para añadir el AWSServiceRoleForCloudHSMrol.

Creación de un rol vinculado a un servicio (manual)

Puede usar la consola de IAM o AWS CLI la API para crear el AWSServiceRoleForCloudHSMrol. Para obtener más información, consulte Crear un rol vinculado a un servicio en la Guía del usuario de IAM.

Edición del rol vinculado al servicio

AWS CloudHSM no permite editar el AWSServiceRoleForCloudHSMrol. Después de que se cree el rol, por ejemplo, no podrá cambiar su nombre porque varias entidades pueden hacer referencia al rol por su nombre. Además, no puede cambiar la política de rol. Sin embargo, puede usar IAM para editar la descripción del rol. Para obtener más información, consulte Editar un rol vinculado a un servicio en la Guía del usuario de IAM.

Eliminación del rol vinculado a un servicio

No puede eliminar una función vinculada a un servicio si todavía existe un clúster al que se haya aplicado. Para eliminar el rol, primero debe eliminar cada HSM de su clúster y, a continuación, eliminar el clúster. Se debe eliminar cada clúster de su cuenta. A continuación, puede utilizar la consola de IAM o la API para eliminar el rol. AWS CLI Para obtener más información acerca de la eliminación de un clúster, consulte Eliminar un AWS CloudHSM clúster. Para más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Conformidad

AWS CloudHSM proporciona políticas de seguridad aprobadas por la FIPS para los HSM. Además, AWS CloudHSM cumple con los requisitos de conformidad con los estándares PCI-PIN, PCI-3DS y SOC2. Confiar en un HSM validado por el FIPS puede ayudarle a cumplir los requisitos de conformidad corporativos, contractuales y normativos en materia de seguridad de los datos en la nube. AWS Para obtener más información, consulte la siguiente información.

Conformidad con FIPS 140-2:

> La Publicación 140-2 del Federal Information Processing Standard (Estándar Federal de Procesamiento de Información, FIPS) es un estándar de seguridad del Gobierno de EE. UU. que especifica los requisitos de seguridad para los módulos criptográficos que protegen información confidencial. Los HSM que proporciona cuentan con AWS CloudHSM la certificación FIPS 140-2 de nivel 3 (certificado #4218). Para obtener más información al respecto, consulte Validación de FIPS para hardware.

Conformidad con DSS de PCI

PCI DSS (siglas en inglés de estándar de seguridad de los datos del sector de tarjetas de pago) es un estándar de seguridad de la información propio administrado por el Consejo de estándares de seguridad del sector de tarjetas de pago. Los HSM suministrados AWS CloudHSM cumplen con la norma PCI DSS.

Conformidad con PIN PCI

El PCI PIN proporciona requisitos de seguridad y estándares de evaluación para la transmisión, el procesamiento y la administración de los datos del número de identificación personal (PIN), información que se utiliza para las transacciones en cajeros automáticos y terminales point-of-sale (POS). AWS CloudHSM cumple con PCI PIN desde enero de 2023. Para obtener más información, consulte el artículo AWS CloudHSM ahora cuenta con la certificación PCI PIN.

Conformidad con PCI-3DS

PCI 3DS (o Three Domain Secure, 3-D Secure) proporciona seguridad de datos para los pagos de comercio electrónico seguros de EMV 3D. PCI 3DS proporciona un nivel adicional de seguridad para las compras online. AWS CloudHSM es conforme a PCI-3DS.

SOC2

SOC2 es un marco que ayuda a las organizaciones de servicios a demostrar sus controles de seguridad en la nube y en los centros de datos. AWS CloudHSM ha implementado controles de SOC2 en áreas críticas para adherirse a los principios de servicio de confianza. Para obtener más información, consulte la Página de preguntas frecuentes sobre SOC de AWS.

AWS CloudHSM Preguntas frecuentes sobre el cumplimiento del PCI-PIN

El PCI PIN proporciona requisitos de seguridad y estándares de evaluación para la transmisión, el procesamiento y la administración de los datos del número de identificación personal (PIN), información que se utiliza para las transacciones en cajeros automáticos y terminales point-of-sale (POS).

Los clientes pueden acceder a la certificación de conformidad (AOC) y el resumen de responsabilidad de PCI-PIN a través de AWS Artifact, un portal de autoservicio que ofrece acceso bajo demanda a los informes de conformidad de AWS. Para obtener más información, inicie sesión en AWS Artifact desde la consola de administración de AWS, o consulte Introducción a AWS Artifact.

Preguntas frecuentes

P: ¿Qué son la certificación de conformidad y el resumen de responsabilidad?

La certificación de conformidad (AOC) la elabora un evaluador de PIN cualificado (QPA) que certifica el cumplimiento de los controles aplicables del estándar AWS CloudHSM PCI-PIN. La matriz resumida de responsabilidades describe los controles que son responsabilidad respectiva de sus clientes y de sus clientes. AWS CloudHSM

P: ¿Cómo puedo obtener la AWS CloudHSM certificación de conformidad?

Los clientes pueden acceder a la declaración de conformidad (AOC) de PCI-PIN a través de AWS Artifact, un portal de autoservicio que ofrece acceso bajo demanda a los informes de conformidad de AWS. Para obtener más información, inicie sesión en AWS Artifact desde la consola de administración de AWS, o consulte Introducción a AWS Artifact.

P: ¿Cómo puedo saber de qué controles PCI-PIN soy responsable?

Para obtener información detallada, consulte el «Resumen de responsabilidad del PIN de AWS CloudHSM PCI» del Paquete de conformidad con los PIN de PCI de AWS, disponible para los clientes a través de AWS Artifact, un portal de autoservicio para acceder bajo demanda a los informes de conformidad de AWS. Para obtener más información, inicie sesión en AWS Artifact desde la consola de administración de AWS, o consulte Introducción a AWS Artifact.

P: Como AWS CloudHSM cliente, ¿puedo confiar en la certificación de conformidad (AOC) del PCI-PIN?

Los clientes deben gestionar su propia conformidad con PCI-PIN. Debe someterse a un proceso formal de certificación de PCI-PIN a través de un asesor cualificado de PIN (QPA), que comprobará que su carga de trabajo de pagos satisface todos los controles y requisitos de PCI-PIN. Sin embargo, para los controles de los que es responsable AWS, su QPA puede confiar en la AWS CloudHSM certificación de conformidad (AOC) sin necesidad de realizar más pruebas.

P: ¿Es AWS CloudHSM responsable de los requisitos de PCI-PIN relacionados con el ciclo de vida de la administración de claves?

AWS CloudHSM es responsable del ciclo de vida de los dispositivos físicos de los HSM. Los clientes son responsables de satisfacer los requisitos del ciclo de vida de gestión de claves establecidos en el estándar PCI-PIN.

P: ¿Qué AWS CloudHSM controles cumplen con el PCI-PIN?

El AOC resume los AWS CloudHSM controles que evalúa la QPA. Los clientes pueden acceder al resumen de responsabilidad de PCI-PIN a través de AWS Artifact, un portal de autoservicio que ofrece acceso bajo demanda a los informes de conformidad de AWS.

P: ¿ AWS CloudHSM Admite funciones de pago como la traducción de códigos PIN y el DUKPT?

No, AWS CloudHSM proporciona HSM de uso general. Con el tiempo, es posible que proporcionemos funciones de pago. Si bien el servicio no realiza funciones de pago directamente, la certificación de conformidad con el PIN AWS CloudHSM PCI permite a los clientes cumplir su propia normativa con la PCI para los servicios que utilizan. AWS CloudHSM Si desea usar los servicios de criptografía de pagos de AWS para su carga de trabajo, consulte el blog "Traslade el procesamiento de pagos a la nube con la criptografía de pagos de AWS".

Notificaciones de obsolescencia

De vez en cuando, AWS CloudHSM puede dejar de funcionar para seguir cumpliendo con los requisitos de las normas FIPS 140, PCI-DSS, PCI-PIN, PCI-3DS y SOC2. Esta página detalla los cambios que se aplican actualmente.

Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024

El Instituto Nacional de Estándares y Tecnología (NIST) [1] recomienda no admitir el cifrado triple DES (DeSede, 3DES, DES3) ni el encapsulado y desencapsulado de claves RSA con padding PKCS #1 v1.5 a partir del 31 de diciembre de 2023. Por lo tanto, su compatibilidad finalizará el 1 de enero de 2024 en nuestras instancias conformes a la Normativa Federal de Procesamiento de Información (FIPS).

Esta guía se aplica a las siguientes operaciones criptográficas:

- Generación de claves Triple DES
  - `CKM_DES3_KEY_GEN` para la biblioteca PKCS #11
  - `DESede` Keygen para el proveedor de JCE
  - `genSymKey` con `-t=21` para KMU
- Cifrado con claves Triple DES (nota: se permiten operaciones de descifrado)
  - Para la biblioteca PKCS #11: cifrado `CKM_DES3_CBC`, cifrado `CKM_DES3_CBC_PAD` y cifrado `CKM_DES3_ECB`
  - Para el proveedor de JCE: cifrado `DESede/CBC/PKCS5Padding`, cifrado `DESede/CBC/NoPadding`, cifrado `DESede/ECB/Padding` y cifrado `DESede/ECB/NoPadding`

- Encapsulado, desencapsulado, cifrado y descifrado de claves RSA con padding PKCS#1 v1.5
  - Encapsulado, desencapsulado, cifrado y descifrado de `CKM_RSA_PKCS` para el SDK de PKCS#11
  - Encapsulado, desencapsulado, cifrado y descifrado de `RSA/ECB/PKCS1Padding` para el SDK de JCE
  - `wrapKey` y `unWrapKey` con `-m 12` para KMU (tenga en cuenta que `12` es el valor del mecanismo `RSA_PKCS`)

[1] Para obtener más información sobre este cambio, consulte las tablas 1 y 5 de la sección Transición en el uso de algoritmos criptográficos y longitudes de clave.

Resiliencia en AWS CloudHSM

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones 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 las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura global.AWS Para obtener más información acerca de las características de AWS CloudHSM que admiten la resiliencia, consulte Alta disponibilidad y balanceo de carga del clúster.

Seguridad de la infraestructura en AWS CloudHSM

Como servicio gestionado, AWS CloudHSM está protegido por los procedimientos de seguridad de red AWS global que se describen en el documento técnico Amazon Web Services: Overview of Security Processes.

Utiliza las llamadas a la API AWS publicadas para acceder a AWS CloudHSM través de la red. 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 de seguridad de IAM principal. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Aislamiento de red

Una nube privada virtual (VPC) es una red virtual en su propia área, aislada lógicamente en la nube de AWS. Puede crear un clúster en una subred privada en la VPC. Puede crear subredes privadas al crear una VPC. Para obtener más información, consulte Cree una nube privada virtual (VPC)..

Al crear un HSM, AWS CloudHSM coloque una interfaz de red elástica (ENI) en la subred para poder interactuar con los HSM. Para obtener más información, consulte Arquitectura de clúster.

AWS CloudHSM crea un grupo de seguridad que permite la comunicación entrante y saliente entre los HSM del clúster. Puede utilizar este grupo de seguridad para permitir que las instancias EC2 se comuniquen con los HSM en el clúster. Para obtener más información, consulte Configuración de los grupos de seguridad de la instancia de cliente de Amazon EC2.

Autorización de usuarios

En este AWS CloudHSM caso, las operaciones que se realizan en el HSM requieren las credenciales de un usuario del HSM autenticado. Para obtener más información, consulte the section called “Más información sobre los usuarios de HSM”.

AWS CloudHSM y puntos finales de VPC

Puede establecer una conexión privada entre su VPC y crear un punto final de AWS CloudHSM la VPC de interfaz. Los puntos de enlace de la interfaz funcionan con una tecnología que le permite acceder de forma privada a AWS CloudHSM las API sin una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una conexión AWS Direct Connect. AWS PrivateLink Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con AWS CloudHSM las API. El tráfico entre la VPC y AWS CloudHSM no sale de la red de Amazon.

Cada punto de conexión de la interfaz está representado por una o más interfaces de red elásticas en las subredes.

Para obtener más información, consulte Interface VPC Endpoints (AWS PrivateLink) en la Guía del usuario de Amazon VPC.

Consideraciones sobre los puntos AWS CloudHSM finales de VPC

Antes de configurar un punto de enlace de VPC de interfaz AWS CloudHSM, asegúrese de revisar las propiedades y limitaciones del punto de enlace de interfaz en la Guía del usuario de Amazon VPC.

- AWS CloudHSM admite realizar llamadas a todas sus acciones de API desde su VPC.

Creación de un punto de conexión de VPC de interfaz para AWS CloudHSM

Puede crear un punto de enlace de VPC para el AWS CloudHSM servicio mediante la consola de Amazon VPC o la (AWS Command Line Interface CLI). Para más información, consulte Creación de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

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

```
com.amazonaws.region.cloudhsmv2
```

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

```
com.amazonaws.us-west-2.cloudhsmv2
```

Para facilitar el uso del punto de enlace de la VPC, puede habilitar un nombre de alojamiento DNS privado para el punto de enlace de la VPC. Si seleccionas la opción Habilitar nombre DNS privado, el nombre de host AWS CloudHSM DNS estándar (`https://cloudhsmv2.<region>.amazonaws.com`) pasa a ser el punto de conexión de tu VPC.

Esta opción facilita el uso del punto de conexión de VPC. AWS Los SDK y la CLI utilizan el nombre de host AWS CloudHSM 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 más información, consulte Acceso a un servicio a través de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Crear una política de puntos de conexión de VPC para AWS CloudHSM

Puede asociar una política de punto de conexión con su punto de conexión de VPC que controla el acceso a AWS CloudHSM. La política especifica la siguiente información:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden llevar a cabo las acciones.

Para más información, consulte Control del acceso a los servicios con puntos de enlace de la VPC en la Guía del usuario de Amazon VPC.

Ejemplo: política de puntos finales de VPC para acciones AWS CloudHSM

El siguiente es un ejemplo de una política de puntos finales para AWS CloudHSM. Cuando se adjunta a un punto final, esta política otorga acceso a las AWS CloudHSM acciones enumeradas a todos los principales de todos los recursos. Administración de identidad y acceso para AWS CloudHSMConsulte otras AWS CloudHSM acciones y sus permisos de IAM correspondientes.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "cloudhsm:DescribeBackups",
            "cloudhsm:DescribeClusters",
            "cloudhsm:ListTags",
         ],
         "Resource":"*"
      }
   ]
}
```

Gestión de actualizaciones en AWS CloudHSM

AWS administra el firmware. Un tercero se encarga de mantener el firmware y NIST debe evaluar la conformidad con FIPS 140-2 de nivel 3. Solo se puede instalar el firmware que dispone de una firma criptográfica de la clave FIPS a la que AWS no tiene acceso.

Solución de problemas AWS CloudHSM

Si tiene problemas con ellos AWS CloudHSM, los siguientes temas pueden ayudarle a resolverlos.

Temas

- Problemas conocidos
- Fallos de sincronización de clave en SDK 3 de cliente
- SDK 3 de cliente: comprobar el rendimiento de HSM con la herramienta pkpspeed
- El usuario de SDK 5 de cliente contiene valores inconsistentes.
- Se detectó un error durante la comprobación de disponibilidad de las claves.
- Extracción de claves con JCE
- Limitación de HSM
- Mantener sincronizados los usuarios de los HSM del clúster
- Conexión perdida con el clúster
- Faltan registros de AWS CloudHSM auditoría CloudWatch
- IV personalizados con una longitud no compatible para el encapsulamiento de claves AES
- Solución de errores de creación de clústeres
- Recuperación de registros de configuración de los clientes

Problemas conocidos

AWS CloudHSM tiene los siguientes problemas conocidos. Elija un tema para obtener más información.

Temas

- Problemas conocidos para todas las instancias de HSM
- Problemas conocidos de la biblioteca PKCS #11
- Problemas conocidos para el SDK de JCE
- Problemas conocidos de OpenSSL Dynamic Engine
- Problemas conocidos para instancias de Amazon EC2 que ejecutan Amazon Linux 2
- Problemas conocidos para integrar aplicaciones de terceros

Problemas conocidos para todas las instancias de HSM

Los siguientes problemas afectan a todos AWS CloudHSM los usuarios, independientemente de si utilizan la herramienta de línea de comandos key_mgmt_util, el SDK PKCS #11, el SDK de JCE o el SDK de OpenSSL.

Temas

- Problema: el encapsulamiento de claves con AES utiliza el relleno PKCS#5 en vez de proporcionar una implementación compatible con los estándares de encapsulamiento de clave con cero relleno.
- Problema: el daemon de cliente requiere al menos una dirección IP válida en su archivo de configuración para conectarse correctamente al clúster.
- Problema: había un límite máximo de 16 KB de datos que se podían procesar y firmar AWS CloudHSM mediante Client SDK 3
- Problema: no se podían especificar las claves importadas como no exportables.
- Problema: Se ha eliminado el mecanismo predeterminado para WrapKey y unWrapKey los comandos de key_mgmt_util
- Problema: si tiene un único HSM en el clúster, la conmutación por error de HSM no funciona correctamente.
- Problema: si se supera la capacidad de claves de los HSM del clúster en un breve periodo de tiempo, el cliente entra en un estado de error no gestionado.
- Problema: no se admiten operaciones de resumen con claves de HMAC de un tamaño superior a 800 bytes.
- Problema: la herramienta client_info, que se distribuye con SDK 3 de cliente, elimina el contenido de la ruta especificada por el argumento de salida opcional.
- Problema: recibe un error al ejecutar la herramienta de configuración del SDK 5 con el argumento --cluster-id en entornos en contenedores.
- Problema: Aparece el error «No se pudo crear el certificado o la clave a partir del archivo pfx proporcionado». Error: 8 pulgadas NotPkcs

Problema: el encapsulamiento de claves con AES utiliza el relleno PKCS#5 en vez de proporcionar una implementación compatible con los estándares de encapsulamiento de clave con cero relleno.

Además, no se admite el encapsulamiento de clave sin relleno o con cero relleno.

- Impacto: no se produce ningún impacto si se empaqueta y se desempaqueta utilizando este algoritmo interno. AWS CloudHSM Sin embargo, las claves empaquetadas AWS CloudHSM no se pueden desempaquetar en otros HSM o software que cumplan con la especificación de no relleno. Esto se debe a que se pueden agregar ocho bytes de datos de relleno al final de los datos clave durante un desencapsulamiento compatible con los estándares. Las claves empaquetadas externamente no se pueden desempaquetar correctamente en una instancia. AWS CloudHSM
- Solución: para desencapsular externamente una clave encapsulada con el encapsulamiento de clave con AES con relleno PKCS #5 en una instancia de AWS CloudHSM, quite el relleno adicional antes de intentar utilizar la clave. Puede hacerlo si recorta los bytes adicionales en un editor de archivos o copia solo los bytes de clave en un nuevo búfer en el código.
- Estado de la resolución: con la versión de software y cliente 3.1.0, AWS CloudHSM ofrece opciones compatibles con los estándares para el encapsulamiento de claves con AES. Para obtener más información, consulte este artículo sobre el encapsulamiento de claves AES.

Problema: el daemon de cliente requiere al menos una dirección IP válida en su archivo de configuración para conectarse correctamente al clúster.

- Impacto: si elimina cada HSM en el clúster y, a continuación, añade otro HSM, que obtiene una nueva dirección IP, el daemon del cliente sigue buscando sus HSM en las direcciones IP originales.
- Solución alternativa: si ejecuta una carga de trabajo intermitente, le recomendamos que utilice el `IpAddress` argumento de la CreateHsmfunción para establecer la elastic network interface (ENI) en su valor original. Tenga en cuenta que la ENI es específica de la zona de disponibilidad (AZ). La alternativa consiste en eliminar el archivo `/opt/cloudhsm/daemon/1/cluster.info` y, a continuación, restablecer la configuración del cliente a la dirección IP de su nuevo HSM. Puede utilizar el comando `client -a <IP address>`. Para obtener más información, consulte Instalar y configurar el AWS CloudHSM cliente (Linux) o Instalar y configurar el AWS CloudHSM cliente (Windows).

Problema: había un límite máximo de 16 KB de datos que se podían procesar y firmar AWS CloudHSM mediante Client SDK 3

- Estado de resolución: los datos con un tamaño inferior a 16 KB siguen siendo enviados al HSM para aplicarles la función hash. Hemos añadido la posibilidad de aplicar la función hash localmente, mediante software, a los datos con un tamaño entre 16 KB y 64 KB. El SDK 5 del

cliente fallará explícitamente si el búfer de datos supera los 64 KB. Debe actualizar el cliente y los SDK a una versión superior a la 5.0.0 o superior para beneficiarse de la corrección.

Problema: no se podían especificar las claves importadas como no exportables.

- Estado de resolución: este problema se ha corregido. No es necesario que realice ninguna acción para beneficiarse de la corrección.

Problema: Se ha eliminado el mecanismo predeterminado para WrapKey y unWrapKey los comandos de key_mgmt_util

- Resolución: Al usar el WrapKey o unWrapKey los comandos, debe usar la `-m` opción para especificar el mecanismo. Consulte los ejemplos en WrapKey o en unWrapKeylos artículos para obtener más información.

Problema: si tiene un único HSM en el clúster, la conmutación por error de HSM no funciona correctamente.

- Impacto: si la única instancia de HSM del clúster pierde la conectividad, el cliente no se volverá a conectar con ella, incluso si la instancia de HSM se restaura posteriormente.
- Solución provisional: le recomendamos al menos dos instancias de HSM en cualquier clúster de producción. Si utiliza esta configuración, no se verá afectado por este problema. Para los clústeres de un solo HSM, reinicie el daemon del cliente para restaurar la conectividad.
- Estado de resolución: este problema se ha solucionado en la versión 1.1.2 del cliente AWS CloudHSM . Debe actualizar a este cliente para beneficiarse de la corrección.

Problema: si se supera la capacidad de claves de los HSM del clúster en un breve periodo de tiempo, el cliente entra en un estado de error no gestionado.

- Impacto: cuando el cliente encuentra el estado de error no gestionado, se bloquea y debe reiniciarse.
- Solución provisional: pruebe el rendimiento para asegurarse de que no está creando claves de sesión a una velocidad que el cliente no pueda gestionar. Puede reducir la velocidad añadiendo un HSM al clúster o ralentizando la creación de claves de sesión.

- Estado de resolución: este problema se ha solucionado en la versión 1.1.2 del cliente AWS CloudHSM . Debe actualizar a este cliente para beneficiarse de la corrección.

Problema: no se admiten operaciones de resumen con claves de HMAC de un tamaño superior a 800 bytes.

- Impacto: las claves de HMAC de más de 800 bytes se pueden generar o importar en el HSM. Sin embargo, si utiliza esta clave más grande en una operación de resumen a través de JCE o de key_mgmt_util, la operación no se realizará correctamente. Tenga en cuenta que si utiliza PKCS11 las claves HMAC se limitan a un tamaño de 64 bytes.
- Solución provisional: si va a utilizar claves de HMAC para operaciones de resumen en el HSM, asegúrese de que el tamaño es inferior a 800 bytes.
- Estado de resolución: ninguno por el momento.

Problema: la herramienta client_info, que se distribuye con SDK 3 de cliente, elimina el contenido de la ruta especificada por el argumento de salida opcional.

- Impacto: es posible que todos los archivos y subdirectorios existentes en la ruta de salida especificada se pierdan permanentemente.
- Solución alternativa: no utilice el argumento opcional `-output` *`path`* cuando utilice la herramienta `client_info`.
- Estado de la resolución: Este problema se ha resuelto en la versión 3.3.2 del SDK de cliente. Debe actualizar a este cliente para beneficiarse de la corrección.

Problema: recibe un error al ejecutar la herramienta de configuración del SDK 5 con el argumento **`--cluster-id`** en entornos en contenedores.

Aparece el siguiente error al usar el argumento --cluster-id con la herramienta de configuración:

```
No credentials in the property bag
```

Este error se debe a una actualización a la versión 2 del servicio de metadatos de la instancia (IMDSv2). Para obtener más información, consulte la documentación de IMDSv2.

- Impacto: este problema afectará a los usuarios que utilicen la herramienta de configuración en las versiones 5.5.0 y posteriores del SDK en entornos en contenedores y que utilicen los metadatos de las instancias de EC2 para proporcionar credenciales.
- Solución alternativa: establezca el límite de saltos de respuesta de PUT en al menos dos. Para obtener información sobre cómo hacerlo, consulte Configurar las opciones de metadatos de la instancia.

Problema: Aparece el error «No se pudo crear el certificado o la clave a partir del archivo pfx proporcionado». Error: 8 pulgadas NotPkcs

- Impacto: los usuarios del SDK 5.11.0 que reconfiguren el SSL con un certificado y una clave privada fallarán si sus claves privadas no están en formato PKCS8.
- Solución alternativa: puede convertir la clave privada SSL personalizada al formato PKCS8 con el comando openssl: `openssl pkcs8 -topk8 -inform PEM -outform PEM -in ssl_private_key -out ssl_private_key_pkcs8`
- Estado de la resolución: este problema se resolvió en la versión 5.12.0 del SDK del cliente. Debe actualizar a esta versión de cliente o posterior para beneficiarse de la solución.

Problemas conocidos de la biblioteca PKCS #11

Temas

- Problema: el encapsulamiento de la clave AES de la versión 3.0.0 de la biblioteca PKCS #11 no valida los IV antes de usarlos.
- Problema: el SDK 2.0.4 de PKCS #11 y las versiones anteriores siempre usaban el IV predeterminado de 0xA6A6A6A6A6A6A6A6 para el encapsulado y desencapsulado de claves AES.
- Problema: el atributo CKA_DERIVE no se admitía y no se gestionaba.
- Problema: el atributo CKA_SENSITIVE no se admitía y no se gestionaba.
- Problema: la función hash multiparte y la firma no son compatibles.
- Problema: C_GenerateKeyPair no gestiona CKA_MODULUS_BITS ni CKA_PUBLIC_EXPONENT en la plantilla privada de una forma que cumpla con los estándares.
- Problema: los búferes para las operaciones C_Encrypt y C_Decrypt API no pueden superar los 16 KB cuando se utiliza el mecanismo CKM_AES_GCM.

- Problema: la derivación de claves Diffie-Hellman (ECDH) de curva elíptica se ejecuta parcialmente en el HSM.
- Problema: la verificación de las firmas secp256k1 falla en las plataformas EL6, como CentOS6 y RHEL 6
- Problema: la secuencia incorrecta de las llamadas a las funciones arroja resultados indefinidos en lugar de fallar.
- Problema: la sesión de solo lectura no es compatible con el SDK 5
- Problema: el archivo de encabezado cryptoki.h es solo para Windows.

Problema: el encapsulamiento de la clave AES de la versión 3.0.0 de la biblioteca PKCS #11 no valida los IV antes de usarlos.

Si especifica un vector de inicialización menor que 8 bytes de longitud, se rellena con bytes impredecibles antes de su uso.

> **Note**
>
> Esto afecta a `C_WrapKey` solo con el mecanismo `CKM_AES_KEY_WRAP`.

- Impacto: si proporciona un IV que es más corto que 8 bytes en la versión 3.0.0 de la biblioteca PKCS #11, es posible que no pueda desencapsular la clave.
- Soluciones provisionales:
  - Le recomendamos encarecidamente que actualice a la versión 3.0.1 o superior de la biblioteca PKCS #11, que aplica correctamente la longitud IV durante el encapsulamiento de claves AES. Modifique su código de encapsulado para pasar un vector de inicialización NULO, o especifique el vector de inicialización predeterminado de `0xA6A6A6A6A6A6A6A6`. Para obtener más información, consulte IV personalizados con longitud no conforme para encapsulamiento de clave AES.
  - Si ha envuelto alguna clave con la versión 3.0.0 de la biblioteca PKCS #11 utilizando un IV inferior a 8 bytes, póngase en contacto con nosotros para obtener ayuda.
- Estado de resolución: este problema se ha resuelto en la versión 3.0.1 de la biblioteca PKCS #11. Para ajustar claves mediante el encapsulado de claves AES, especifique un vector de inicialización que sea NULO o de 8 bytes de longitud.

Problema: el SDK 2.0.4 de PKCS #11 y las versiones anteriores siempre usaban el IV predeterminado de **`0xA6A6A6A6A6A6A6A6`** para el encapsulado y desencapsulado de claves AES.

Los vectores de inicialización proporcionados por el usuario se han ignorado de forma silenciosa.

> **Note**
>
> Esto afecta a `C_WrapKey` solo con el mecanismo `CKM_AES_KEY_WRAP`.

- Impacto:
  - Si utilizó el SDK de PKCS #11 2.0.4 o una versión anterior y un vector de inicialización proporcionado por el usuario, las claves se encapsulan con el vector de inicialización predeterminado de `0xA6A6A6A6A6A6A6A6`.
  - Si utilizó el SDK de PKCS #11 3.0.0 o posterior y un vector de inicialización proporcionado por el usuario, las claves se encapsulan con el vector de inicialización proporcionado por el usuario.
- Soluciones provisionales:
  - Para desencapsular las claves encapsuladas con el SDK de PKCS #11 2.0.4 o anterior, utilice el vector de inicialización predeterminado de `0xA6A6A6A6A6A6A6A6`.
  - Para desencapsular claves encapsuladas con el SDK de PKCS #11 3.0.0 o posterior, utilice el vector de inicialización proporcionado por el usuario.
- Estado de resolución: le recomendamos que modifique el código de encapsulado y desencapsulado para pasar un vector de inicialización NULO o que especifique el vector de inicialización predeterminado de `0xA6A6A6A6A6A6A6A6`.

Problema: el atributo **`CKA_DERIVE`** no se admitía y no se gestionaba.

- Estado de resolución: hemos implementado correcciones para aceptar `CKA_DERIVE` si se ha establecido en `FALSE`. No se admitirá que `CKA_DERIVE` se establezca en `TRUE` hasta que comencemos a añadir compatibilidad con la función de derivación de claves en AWS CloudHSM. Debe actualizar el cliente y los SDK a la versión 1.1.1 o posterior para beneficiarse de la corrección.

Problema: el atributo **CKA_SENSITIVE** no se admitía y no se gestionaba.

- Estado de resolución: hemos implementado correcciones para aceptar y procesar correctamente el atributo `CKA_SENSITIVE`. Debe actualizar el cliente y los SDK a la versión 1.1.1 o posterior para beneficiarse de la corrección.

Problema: la función hash multiparte y la firma no son compatibles.

- Impacto: `C_DigestUpdate` y `C_DigestFinal` no se implementan. `C_SignFinal` tampoco se implementa y producirá un error con `CKR_ARGUMENTS_BAD` en los búferes no `NULL`.
- Solución alternativa: aplique un hash a los datos dentro de la aplicación y AWS CloudHSM utilícelos únicamente para firmar el hash.
- Estado de resolución: estamos corrigiendo el cliente y los SDK para implementar correctamente la función hash multiparte. Las actualizaciones se anunciarán en el foro de AWS CloudHSM y en la página de historial de versiones.

Problema: **C_GenerateKeyPair** no gestiona **CKA_MODULUS_BITS** ni **CKA_PUBLIC_EXPONENT** en la plantilla privada de una forma que cumpla con los estándares.

- Impacto: `C_GenerateKeyPair` debe devolver `CKA_TEMPLATE_INCONSISTENT` cuando la plantilla privada contiene `CKA_MODULUS_BITS` o `CKA_PUBLIC_EXPONENT`. En cambio, genera una clave privada en la que todos los campos se establecen en `FALSE`. La clave no se puede usar.
- Solución: recomendamos que su aplicación compruebe los valores del campo de uso además del código de error.
- Estado de resolución: estamos implementado soluciones para devolver el mensaje de error correcto cuando se usa una plantilla de clave privada incorrecta. La biblioteca PKCS #11 actualizada se anunciará en la página de historial de versiones.

Problema: los búferes para las operaciones **C_Encrypt** y **C_Decrypt** API no pueden superar los 16 KB cuando se utiliza el mecanismo **CKM_AES_GCM**.

AWS CloudHSM no admite el cifrado AES-GCM multiparte.

- Impacto: no puede usar el mecanismo `CKM_AES_GCM` para cifrar datos mayores de 16 KB.

- Solución: puede usar un mecanismo alternativo como `CKM_AES_CBC`,`CKM_AES_CBC_PAD` o puede dividir los datos en partes y cifrar cada parte usando `AES_GCM` individualmente. Si lo utiliza`AES_GCM`, debe gestionar la división de sus datos y el posterior cifrado. AWS CloudHSM no realiza el cifrado AES-GCM multiparte por usted. Tenga en cuenta que FIPS exige que el vector de inicialización (IV) para `AES-GCM` se genere en el HSM. Por lo tanto, el IV será diferente para cada parte de sus datos con cifrado AES-GCM.
- Estado de resolución: estamos corrigiendo los SDK para que produzcan un error de forma explícita si el búfer de datos es demasiado grande. Se devuelve `CKR_MECHANISM_INVALID` para las API de operaciones `C_EncryptUpdate` y `C_DecryptUpdate`. Estamos evaluando alternativas para admitir búferes más grandes sin recurrir al cifrado multiparte. Las actualizaciones se anunciarán en el AWS CloudHSM foro y en la página del historial de versiones.

Problema: la derivación de claves Diffie-Hellman (ECDH) de curva elíptica se ejecuta parcialmente en el HSM.

Su clave privada EC permanece dentro del HSM en todo momento, pero el proceso de generación de la clave se realiza en varios pasos. Como resultado, los resultados intermedios de cada paso están disponibles en el cliente.

- Impacto: en el SDK de cliente 3, la clave derivada mediante el `CKM_ECDH1_DERIVE` mecanismo está disponible primero en el cliente y, a continuación, se importa al HSM. Un identificador de clave se devuelve después a su aplicación.
- Solución: si implementa una descarga de SSL/TLS en AWS CloudHSM, es posible que esta limitación no represente un problema. Si su aplicación requiere que su clave permanezca dentro de un límite FIPS en todo momento, considere el uso de un protocolo alternativo que no se base en la generación de la clave ECDH.
- Estado de resolución: desarrollamos la opción para llevar a cabo la generación de la clave ECDH en su totalidad dentro del HSM. La implementación actualizada se anunciará en la página de historial de versiones una vez que esté disponible.

Problema: la verificación de las firmas secp256k1 falla en las plataformas EL6, como CentOS6 y RHEL 6

Esto se debe a que la biblioteca PKCS #11 de CloudHSM impide que se realice una llamada de red durante la inicialización de la operación de verificación mediante el uso de OpenSSL para verificar

los datos de la curva de EC. Como Secp256k1 no es compatible con el paquete OpenSSL en las plataformas EL6, la inicialización produce un error.

- Impacto: la verificación de firmas Secp256k1 producirá un error en las plataformas EL6. La llamada de verificación producirá el error `CKR_HOST_MEMORY`.
- Solución: le recomendamos que utilice Amazon Linux 1 o cualquier plataforma EL7 si su aplicación PKCS # 11 necesita verificar firmas secp256k1. Otra solución consiste en actualizar a una versión del paquete OpenSSL que admita la curva secp256k1.
- Estado de resolución: estamos implementando correcciones para revertir al HSM si la validación de curvas locales no está disponible. La biblioteca de PKCS #11 actualizada se anunciará en la página de historial de versiones.

Problema: la secuencia incorrecta de las llamadas a las funciones arroja resultados indefinidos en lugar de fallar.

- Impacto: si llama a una secuencia de funciones incorrecta, el resultado final es incorrecto aunque las llamadas individuales a las funciones se devuelvan correctamente. Por ejemplo, es posible que los datos descifrados no coincidan con el texto no cifrado original o que las firmas no se puedan verificar. Este problema afecta tanto a las operaciones de una sola parte como a las de varias partes.

  Ejemplos de secuencias de funciones incorrectas:

  - `C_EncryptInit`/`C_EncryptUpdate` seguido de `C_Encrypt`
  - `C_DecryptInit`/`C_DecryptUpdate` seguido de `C_Decrypt`
  - `C_SignInit`/`C_SignUpdate` seguido de `C_Sign`
  - `C_VerifyInit`/`C_VerifyUpdate` seguido de `C_Verify`
  - `C_FindObjectsInit` seguido de `C_FindObjectsInit`
- Solución alternativa: su aplicación debe, de conformidad con la especificación PKCS #11, utilizar la secuencia correcta de llamadas a funciones tanto para las operaciones de una sola parte como para las de varias partes. En estas circunstancias, su aplicación no debe confiar en la biblioteca de CloudHSM PKCS #11 para devolver un error.

Problema: la sesión de solo lectura no es compatible con el SDK 5

- Problema: el SDK 5 no admite la apertura de sesiones de solo lectura con `C_OpenSession`.

- Impacto: si intenta llamar a `C_OpenSession` sin proporcionar un `CKF_RW_SESSION`, la llamada fallará y aparecerá el error `CKR_FUNCTION_FAILED`.
- Solución alternativa: al abrir una sesión, debe pasar los marcadores de `CKF_SERIAL_SESSION | CKF_RW_SESSION` a la llamada de función `C_OpenSession`.

Problema: el archivo de encabezado **`cryptoki.h`** es solo para Windows.

- Problema: en las versiones 5.0.0 a 5.4.0 del AWS CloudHSM Client SDK 5 en Linux, el archivo de encabezado solo es compatible con los sistemas operativos Windows. `/opt/cloudhsm/include/pkcs11/cryptoki.h`
- Impacto: en los sistemas operativos basados en Linux, es posible que se produzcan problemas al intentar incluir este archivo de encabezado en la aplicación.
- Estado de la resolución: actualice a la versión 5.4.1 o superior del AWS CloudHSM Client SDK 5, que incluye una versión de este archivo de encabezado compatible con Linux.

Problemas conocidos para el SDK de JCE

Temas

- Problema: si trabaja con pares de claves asimétricas, verá que la capacidad para las claves está ocupada aunque no esté creando o importando las claves explícitamente.
- Problema: El JCE es de solo lectura KeyStore
- Problema: los búferes para el cifrado AES-GCM no pueden superar los 16 000 bytes.
- Problema: la derivación de claves Diffie-Hellman (ECDH) de curva elíptica se ejecuta parcialmente en el HSM.
- Problema: KeyGenerator e interpreta KeyAttribute incorrectamente el parámetro de tamaño de la clave como número de bytes en lugar de bits
- Problema: SDK 5 de cliente muestra la advertencia «Se ha producido una operación ilegal de acceso reflexivo».
- Problema: el grupo de sesiones de JCE está agotado.
- Problema: pérdida de memoria del SDK 5 del cliente al operar con GetKey

Problema: si trabaja con pares de claves asimétricas, verá que la capacidad para las claves está ocupada aunque no esté creando o importando las claves explícitamente.

- Impacto: este problema puede provocar que los HSM se queden sin espacio para las claves de forma inesperada y se produce cuando la aplicación utiliza un objeto de clave JCE estándar para las operaciones de criptografía en lugar de un objeto `CaviumKey`. Si utiliza un objeto de clave JCE estándar, `CaviumProvider` importa implícitamente esa clave en el HSM como una clave de sesión y no elimina esta clave hasta que se cierra la aplicación. Como resultado, las claves se acumulan mientras la aplicación se está ejecutando y pueden hacer que los HSM se queden sin espacio libre para las claves, lo que bloquea la aplicación.
- Solución: si utiliza la clase `CaviumSignature`, la clase `CaviumCipher`, la clase `CaviumMac` o la clase `CaviumKeyAgreement`, debe proporcionar la clave como un objeto `CaviumKey` en lugar que como un objeto de clave JCE estándar.

  Puede convertir manualmente una clave normal en un objeto `CaviumKey` utilizando la clase `ImportKey` y puede eliminar manualmente la clave una vez que se haya completado la operación.
- Estado de la resolución: estamos actualizando `CaviumProvider` para administrar correctamente las importaciones implícitas. La corrección se anunciará en la página del historial de versiones una vez que esté disponible.

Problema: El JCE es de solo lectura KeyStore

- Impacto: en la actualidad no puede almacenar un tipo de objeto que no sea compatible con HSM en el almacén de claves de JCE. En concreto, no puede almacenar certificados en el almacén de claves. Esto impide la interoperabilidad con herramientas como jarsigner, que espera encontrar el certificado en el almacén de claves.
- Solución: puede rehacer el código para cargar los certificados desde archivos locales o desde una ubicación de bucket de S3 en lugar de hacerlo desde el almacén de claves.
- Estado de resolución: estamos añadiendo compatibilidad con el almacenamiento de certificados en el almacén de claves. La característica se anunciará en la página de historial de versiones una vez que esté disponible.

Problema: los búferes para el cifrado AES-GCM no pueden superar los 16 000 bytes.

Además, no se admite el cifrado AES-GCM multiparte.

- Impacto: no puede usar AES-GCM para cifrar datos que superen los 16 000 bytes.
- Solución: puede usar un mecanismo alternativo como AES-CBC o puede dividir los datos en partes y cifrar cada parte individualmente. Si divide los datos, debe administrar el texto cifrado dividido y su descifrado. Como FIPS requiere que el vector de inicialización (IV) para AES-GCM se genere en el HSM, el IV de cada elemento de datos con cifrado AES-GCM será diferente.
- Estado de resolución: estamos corrigiendo los SDK para que produzcan un error de forma explícita si el búfer de datos es demasiado grande. Estamos evaluando alternativas que admitan búferes más grandes sin recurrir al cifrado multiparte. Las actualizaciones se anunciarán en el foro de AWS CloudHSM y en la página de historial de versiones.

Problema: la derivación de claves Diffie-Hellman (ECDH) de curva elíptica se ejecuta parcialmente en el HSM.

Su clave privada EC permanece dentro del HSM en todo momento, pero el proceso de generación de la clave se realiza en varios pasos. Como resultado, los resultados intermedios de cada paso están disponibles en el cliente. Encontrará una muestra de la derivación de claves de ECDH en los ejemplos de código de Java.

- Impacto: el SDK de cliente 3 añade la funcionalidad ECDH al JCE. Cuando se utiliza la `KeyAgreement` clase para derivar un SecretKey, primero está disponible en el cliente y, a continuación, se importa al HSM. Un identificador de clave se devuelve después a su aplicación.
- Solución alternativa: si va a implementar SSL/TLS Offload en AWS CloudHSM, es posible que esta limitación no suponga un problema. Si su aplicación requiere que su clave permanezca dentro de un límite FIPS en todo momento, considere el uso de un protocolo alternativo que no se base en la generación de la clave ECDH.
- Estado de resolución: desarrollamos la opción para llevar a cabo la generación de la clave ECDH en su totalidad dentro del HSM. Cuando esté disponible, anunciaremos la implementación actualizada en la página del historial de versiones.

Problema: KeyGenerator e interpreta KeyAttribute incorrectamente el parámetro de tamaño de la clave como número de bytes en lugar de bits

Al generar una clave mediante la `init` función de la KeyGenerator clase o el `SIZE` atributo de la AWS CloudHSM KeyAttribute enumeración, la API espera erróneamente que el argumento sea el número de bytes de la clave, cuando debería ser el número de bits de la clave.

- Impacto: las versiones 5.4.0 a 5.4.2 de SDK de cliente esperaban erróneamente que el tamaño de la clave se proporcionara a las API especificadas en bytes.
- Solución alternativa: Convierta el tamaño de la clave de bits a bytes antes de usar la KeyGenerator clase o KeyAttribute enumeración para generar claves con el proveedor AWS CloudHSM JCE si utiliza las versiones 5.4.0 a 5.4.2 del SDK de cliente.
- Estado de la resolución: actualice la versión 5.5.0 o posterior del SDK de su cliente, que incluye una corrección para calcular correctamente los tamaños de las claves en bits al utilizar la KeyGenerator clase o la enumeración para generar claves. KeyAttribute

Problema: SDK 5 de cliente muestra la advertencia «Se ha producido una operación ilegal de acceso reflexivo».

Al usar SDK 5 de cliente con Java 11, CloudHSM lanza la siguiente advertencia de Java:

```
```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by
 com.amazonaws.cloudhsm.jce.provider.CloudHsmKeyStore (file:/opt/cloudhsm/java/
cloudhsm-jce-5.6.0.jar) to field java.security .KeyStore.keyStoreSpi
WARNING: Please consider reporting this to the maintainers of
 com.amazonaws.cloudhsm.jce.provider.CloudHsmKeyStore
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective
 access operations
WARNING: All illegal access operations will be denied in a future release
```
```

Estas advertencias no tienen ningún efecto. Somos conscientes de este problema y estamos trabajando para resolverlo. No se necesita ninguna acción ni solución alternativa.

Problema: el grupo de sesiones de JCE está agotado.

Impacto: es posible que no pueda realizar operaciones en JCE después de ver el siguiente mensaje:

```
com.amazonaws.cloudhsm.jce.jni.exception.InternalException: There are too many
 operations
happening at the same time: Reached max number of sessions in session pool: 1000
```

Soluciones provisionales:

- Reinicie la aplicación JCE si está experimentando algún impacto.
- Al realizar una operación, es posible que tenga que finalizar la operación de JCE antes de perder la referencia a la operación.

> **Note**
>
> En función de la operación, puede ser necesario un método de finalización.

Operación	Método/s de finalización
Cifrado	`doFinal()` en modo de cifrado o descifrado `wrap()` en modo de encapsulamiento `unwrap()` en modo de desencapsulamiento
KeyAgreement	`generateSecret()` o `generateSecret(String)`
KeyPairGenerator	`generateKeyPair()` , `genKeyPair()` , o `reset()`
KeyStore	No se necesita ningún método.
MAC	`doFinal()` o `reset()`
MessageDigest	`digest()` o `reset()`
SecretKeyFactory	No se necesita ningún método.
SecureRandom	No se necesita ningún método.
Signature	`sign()` en modo de señal `verify()` en modo de verificación

Estado de la resolución: hemos resuelto este problema en el SDK del cliente 5.9.0 y versiones posteriores. Para solucionar este problema, actualice su SDK del cliente a una de estas versiones.

Problema: pérdida de memoria del SDK 5 del cliente al operar con GetKey

- Impacto: la `getKey` operación de la API tenía una pérdida de memoria en JCE en las versiones 5.10.0 y anteriores del SDK de cliente. Si utilizas la `getKey` API varias veces en tu aplicación, esto aumentará el crecimiento de la memoria y, en consecuencia, aumentará el consumo de memoria de la aplicación. Con el tiempo, esto puede provocar errores de limitación o requerir el reinicio de la aplicación.
- Solución alternativa: se recomienda actualizar a Client SDK 5.11.0. Si esto no se puede hacer, te recomendamos que no llames a la `getKey` API varias veces en tu aplicación. En su lugar, reutilice la clave devuelta anteriormente de la `getKey` operación anterior en la medida de lo posible.
- Estado de la resolución: actualice la versión del SDK de su cliente a la 5.11.0 o posterior, que incluye una solución para este problema.

Problemas conocidos de OpenSSL Dynamic Engine

Estos son los problemas conocidos de OpenSSL Dynamic Engine.

Temas

- Problema: no se puede instalar AWS CloudHSM OpenSSL Dynamic Engine en RHEL 6 y CentOS6
- Problema: solo se admite la descarga de RSA en HSM de forma predeterminada.
- Problema: no se admite el cifrado y descifrado RSA con relleno OAEP por medio de una clave en el HSM.
- Problema: solo se lleva a cabo en el HSM la generación de la clave privada para las claves RSA y ECC.
- Problema: no se puede instalar OpenSSL Dynamic Engine para SDK 3 de cliente en RHEL 8, CentOS 8 o Ubuntu 18.04 LTS.
- Problema: el SHA-1 Sign and Verify está obsoleto en RHEL 9 (9.2+)
- Problema: el motor dinámico de AWS CloudHSM OpenSSL no es compatible con el proveedor FIPS de OpenSSL v3.x

Problema: no se puede instalar AWS CloudHSM OpenSSL Dynamic Engine en RHEL 6 y CentOS6

- Impacto: el motor dinámico de OpenSSL solo admite OpenSSL 1.0.2 [f+]. De forma predeterminada, RHEL 6 y CentOS 6 vienen con OpenSSL 1.0.1.
- Solución: actualice la biblioteca OpenSSL de RHEL 6 y CentOS 6 a la versión 1.0.2 [f+].

Problema: solo se admite la descarga de RSA en HSM de forma predeterminada.

- Impacto: para maximizar el desempeño, el SDK no está configurada para descargar funciones adicionales, como la generación de números aleatorios o las operaciones EC-DH.
- Solución: contacte con nosotros a través de un caso de soporte si necesita descargar operaciones adicionales.
- Estado de resolución: estamos agregando compatibilidad al SDK para configurar las opciones de descarga a través de un archivo de configuración. La actualización se anunciará en la página del historial de versiones cuando esté disponible.

Problema: no se admite el cifrado y descifrado RSA con relleno OAEP por medio de una clave en el HSM.

- Impacto: cualquier llamada al cifrado y descifrado RSA con relleno OAEP produce un error. divide-by-zero Esto se debe a que el motor dinámico de OpenSSL llama a la operación localmente mediante el archivo PEM falso en lugar de descargar la operación al HSM.
- Solución: puede realizar este procedimiento mediante Biblioteca PKCS #11 o Proveedor de JCE.
- Estado de resolución: estamos añadiendo el servicio de soporte para el SDK para descargar correctamente esta operación. La actualización se anunciará en la página del historial de versiones cuando esté disponible.

Problema: solo se lleva a cabo en el HSM la generación de la clave privada para las claves RSA y ECC.

Para cualquier otro tipo de clave, el motor AWS CloudHSM OpenSSL no se utiliza para el procesamiento de llamadas. En su lugar se usa el motor de OpenSSL local. Esto genera una clave localmente mediante software.

- Impacto: debido a que la conmutación por error es silenciosa, no hay ninguna indicación que no ha recibido una clave generada de forma segura en el HSM. Verá un rastro de salida que contiene la cadena `"..........++++++"` si la clave se ha generado localmente por OpenSSL mediante

software. Este rastro no aparece cuando la operación se lleva a cabo en el HSM. Dado que la clave no se genera ni está almacenada en el HSM, no estará disponible para su uso futuro.

- Solución provisional: utilice el motor de OpenSSL para los tipos de claves que admite. Para todos los demás tipos de claves, utilice PKCS #11 o JCE en las aplicaciones o `key_mgmt_util` en la CLI.

Problema: no se puede instalar OpenSSL Dynamic Engine para SDK 3 de cliente en RHEL 8, CentOS 8 o Ubuntu 18.04 LTS.

- Impacto: de forma predeterminada, RHEL 8, CentOS 8 y Ubuntu 18.04 LTS incluyen una versión de OpenSSL que no es compatible con OpenSSL Dynamic Engine para SDK 3 de cliente.
- Solución alternativa: utilice una plataforma Linux que sea compatible con el motor dinámico de OpenSSL. Para obtener información acerca de las plataformas admitidas, consulte Plataformas admitidas.
- Estado de la resolución: AWS CloudHSM admite estas plataformas con OpenSSL Dynamic Engine for Client SDK 5. Para obtener más información, consulte Plataformas compatibles y Motor dinámico de OpenSSL.

Problema: el SHA-1 Sign and Verify está obsoleto en RHEL 9 (9.2+)

- Impacto: El uso del resumen de mensajes del SHA-1 con fines criptográficos ha quedado obsoleto en RHEL 9 (9.2+). Como resultado, las operaciones de firma y verificación con SHA-1 mediante el motor dinámico OpenSSL fallarán.
- Solución alternativa: si su situación requiere el uso del SHA-1 para firmar o verificar firmas criptográficas existentes o de terceros, consulte Mejorar la seguridad de RHEL: entender la obsolescencia del SHA-1 en las notas de la versión RHEL 9 (9.2+) y RHEL 9 (9.2+) para obtener más información.

Problema: el motor dinámico de AWS CloudHSM OpenSSL no es compatible con el proveedor FIPS de OpenSSL v3.x

- Impacto: Recibirá un error si intenta utilizar el motor dinámico de AWS CloudHSM OpenSSL cuando el proveedor FIPS esté activado para las versiones 3.x de OpenSSL.

- Solución alternativa: para utilizar el motor dinámico de AWS CloudHSM OpenSSL con las versiones 3.x de OpenSSL, asegúrese de que el proveedor «predeterminado» esté configurado. Obtenga más información sobre el proveedor predeterminado en el sitio web de OpenSSL.

Problemas conocidos para instancias de Amazon EC2 que ejecutan Amazon Linux 2

Problema: la versión 2018.07 de Amazon Linux 2 usa un **`ncurses`** paquete actualizado (versión 6) que actualmente no es compatible con los SDK AWS CloudHSM

Al ejecutar AWS CloudHSMcloudhsm_mgmt_util o key_mgmt_util, aparece el siguiente error:

```
/opt/cloudhsm/bin/cloudhsm_mgmt_util: error while loading shared libraries:
 libncurses.so.5: cannot open shared object file: No such file or directory
```

- Impacto: las instancias que se ejecutan en la versión 2018.07 de Amazon Linux 2 no podrán usar todas las AWS CloudHSM utilidades.
- Solución: emita el siguiente comando en sus instancias EC2 de Amazon Linux 2 para instalar el paquete compatible `ncurses` (versión 5):

  ```
  sudo yum update && yum install ncurses-compat-libs
  ```

- Estado de resolución: este problema se ha solucionado en la versión 1.1.2 del cliente AWS CloudHSM . Debe actualizar a este cliente para beneficiarse de la corrección.

Problemas conocidos para integrar aplicaciones de terceros

Problema: SDK 3 de cliente no admite que Oracle establezca el atributo PKCS #11 **`CKA_MODIFIABLE`** durante la generación de la clave maestra.

Este límite se define en la biblioteca PKCS #11. Para obtener más información, consulte la anotación 1 de Atributos de PKCS #11 admitidos.

- Impacto: se produce un error al crear claves maestras de Oracle.
- Solución: establezca la variable de entorno especial `CLOUDHSM_IGNORE_CKA_MODIFIABLE_FALSE` en TRUE cuando cree una nueva clave maestra.

Esta variable de entorno solo es necesaria para generar claves maestras; no la necesita para nada más. Por ejemplo, puede usar esta variable con la primera clave maestra que cree: Después, solo tendría que utilizarla si quisiera cambiar la edición de la clave maestra. Para obtener más información, consulte Generar la clave de cifrado maestra de TDE de Oracle.

- Estado de la resolución: estamos mejorando el firmware de HSM para que sea totalmente compatible con el atributo CKA_MODIFIABLE. Las actualizaciones se anunciarán en el AWS CloudHSM foro y en la página del historial de versiones

Fallos de sincronización de clave en SDK 3 de cliente

En Client SDK 3, si se produce un error en la sincronización del lado del cliente, AWS CloudHSM hace todo lo posible para limpiar las claves no deseadas que se hayan creado (y que ahora no sean deseadas). Este proceso implica eliminar de inmediato el material de claves no deseadas o marcarlo como material no deseado para su eliminación posterior. En ambos casos, la resolución no requiere ninguna acción por su parte. En el raro caso de que AWS CloudHSM no pueda eliminar ni marcar el material clave no deseado, debe eliminarlo.

Problema: al intentar realizar una operación de generación, importación o desenscapsulado de claves de token, recibe un mensaje indicando que se ha producido un error de tombstone.

```
2018-12-24T18:28:54Z liquidSecurity ERR: print_node_ts_status:
[create_object_min_nodes]Key: 264617 failed to tombstone on node:1
```

Causa: no AWS CloudHSM se pudo eliminar ni marcar el material clave no deseado.

Solución: un HSM de su clúster contiene material de claves no deseadas que no ha sido marcado como material no deseado. Debe eliminar manualmente el material de claves. Para eliminar manualmente el material de claves no deseadas, use key_mgmt_util (KMU) o una API de la biblioteca PKCS #11 o del proveedor de JCE. Para obtener más información, consulte deleteKey o SDK del cliente.

Para que las claves simbólicas sean más duraderas, AWS CloudHSM se produce un error en las operaciones de creación de claves que no se realizan correctamente en el número mínimo de HSM especificado en la configuración de sincronización del lado del cliente. Para obtener más información, consulte Sincronización de claves en AWS CloudHSM.

SDK 3 de cliente: comprobar el rendimiento de HSM con la herramienta pkpspeed

En este tema se describe cómo validar el rendimiento de HSM con el SDK 3 de cliente.

Para comprobar el rendimiento de los HSM del AWS CloudHSM clúster, puede utilizar la herramienta pkpspeed (Linux) o pkpspeed_blocking (Windows) que se incluye con el Client SDK 3. La herramienta pkpspeed se ejecuta en condiciones ideales y llama directamente al HSM para ejecutar las operaciones sin necesidad de utilizar un SDK, como el PKCS11. Recomendamos realizar pruebas de carga de la aplicación de forma independiente para determinar sus necesidades de escalamiento. No recomendamos realizar las siguientes pruebas: Random (I), ModExp (R) y EC point mul (Y).

Para obtener más información acerca de la instalación del cliente en una instancia de Linux EC2, consulte Instalar y configurar el AWS CloudHSM cliente (Linux). Para obtener más información acerca de la instalación del cliente en una instancia de Windows, consulte Instalación y configuración del AWS CloudHSM cliente (Windows).

Tras instalar y configurar el AWS CloudHSM cliente, ejecute el siguiente comando para iniciarlo.

Amazon Linux

```
$ sudo start cloudhsm-client
```

Amazon Linux 2

```
$ sudo service cloudhsm-client start
```

CentOS 7

```
$ sudo service cloudhsm-client start
```

CentOS 8

```
$ sudo service cloudhsm-client start
```

RHEL 7

```
$ sudo service cloudhsm-client start
```

RHEL 8

```
$ sudo service cloudhsm-client start
```

Ubuntu 16.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client start
```

Windows

- Para la versión 1.1.2 y posteriores del cliente de Windows:

```
C:\Program Files\Amazon\CloudHSM>net.exe start AWSCloudHSMClient
```

- Para la versión 1.1.1 y anteriores de clientes de Windows:

```
C:\Program Files\Amazon\CloudHSM>start "cloudhsm_client" cloudhsm_client.exe C:
\ProgramData\Amazon\CloudHSM\data\cloudhsm_client.cfg
```

Si ya ha instalado el software de cliente, es posible que deba descargar e instalar la última versión para obtener pkpspeed. Puede encontrar la herramienta pkpspeed `/opt/cloudhsm/bin/pkpspeed` en Linux o en `C:\Program Files\Amazon\CloudHSM\` en Windows.

Para utilizar pkpspeed, ejecute el comando pkpspeed o pkpspeed_blocking.exe especificando el nombre de usuario y la contraseña de un usuario de criptografía (CU) del HSM. A continuación, defina las opciones que se utilizarán mientras tiene en cuenta las siguientes recomendaciones.

Recomendaciones de prueba

- Para probar el desempeño de las operaciones de firma y comprobación de RSA y verificar las operaciones, elija el cifrado `RSA_CRT` en Linux o la opción B en Windows. No elija `RSA` (opción A en Windows). La cifrados son equivalentes, pero `RSA_CRT` está optimizado para el desempeño.
- Comience por un número reducido de subprocesos. Para probar el desempeño de AES suele bastar con un subproceso para mostrar el máximo desempeño. Para probar el desempeño de RSA (`RSA_CRT`), normalmente es suficiente con tres o cuatro subprocesos.

Opciones configurables para la herramienta pkpspeed

- Modo FIPS: siempre AWS CloudHSM está en modo FIPS (consulte las AWS CloudHSM preguntas frecuentes para obtener más información). Esto se puede verificar mediante las herramientas CLI, tal como se documenta en la Guía del AWS CloudHSM usuario, y ejecutando el getHSMInfo comando que indicará el estado del modo FIPS.
- Tipo de prueba (con bloqueo o sin bloqueo): especifica cómo se realizan las operaciones por subprocesos. Lo más probable es que consiga mejores números si utiliza el modo de no bloqueo. Esto se debe a que se utilizan subprocesos y simultaneidad.
- Número de subprocesos: número de subprocesos con los que se va a ejecutar la prueba.
- Tiempo en segundos para ejecutar la prueba (máximo = 600): pkpspeed produce los resultados medidos en "OPERACIONES/segundos" e informa este valor por cada segundo que se ejecuta la prueba. Por ejemplo, si la prueba se ejecuta durante 5 segundos, el resultado puede asemejarse a los siguientes valores de muestra:
  - `OPERATIONS/second 821/1`
  - `OPERATIONS/second 833/1`
  - `OPERATIONS/second 845/1`
  - `OPERATIONS/second 835/1`
  - `OPERATIONS/second 837/1`

Pruebas que se pueden ejecutar con la herramienta pkpspeed

- AES GCM: realiza una prueba el cifrado en modo AES GCM.
- Basic 3DES CBC: realiza una prueba del cifrado en modo CBC de 3DES. Consulte la nota 1 que aparece a continuación para ver los próximos cambios.
- Basic AES: realiza una prueba del cifrado AES CBC/ECB.
- Digest: realiza una prueba del resumen del hash.
- ECDSA Sign: realiza una prueba de la firma de ECDSA.
- ECDSA Verify: realiza una prueba de la verificación de ECDSA.
- FIPS Random: realiza una prueba de la generación de un número aleatorio conforme con las normas FIPS. (Nota: solo se puede usar en modo de bloqueo).
- HMAC: realiza una prueba del HMAC.
- Random: esta prueba no es relevante porque utilizamos los HSM del FIPS 140-2.

- RSA non-CRT versus RSA_CRT: realiza una prueba de las operaciones de firma y verificación de RSA.
- RSA OAEP Enc: realiza una prueba del cifrado RSA OAEP.
- RSA OAEP Dec: realiza una prueba del descifrado de RSA OAEP.
- RSA private dec non-CRT: realiza una prueba del cifrado de clave privada RSA (sin optimizar).
- RSA private key dec CRT: realiza una prueba del cifrado de clave privada RSA (optimizado).
- RSA PSS Sign: realiza una prueba de la firma de PSS de RSA.
- RSA PSS Verify: realiza una prueba de la verificación de PSS de RSA.
- RSA public key enc: realiza una prueba del cifrado de clave pública de RSA.

El cifrado de clave pública RSA, el descifrado privado RSA (no CRT) y el descifrado de clave privada RSA (CRT) también solicitarán al usuario que responda lo siguiente:

```
Do you want to use static key [y/n]
```

Si se introduce y, se importa una clave calculada previamente al HSM.

Si se introduce n, se genera una nueva clave.

[1] No autorizado después de 2023 para el cumplimiento de FIPS según las directrices del NIST. Para obtener más información, consulte Cumplimiento de la normativa FIPS 140: anulación de mecanismo 2024.

Ejemplos

En los ejemplos siguientes se muestran las opciones que puede elegir con pkpspeed (Linux) o pkpspeed_blocking (Windows) para probar el desempeño de HSM para las operaciones de RSA y AES.

Example : uso de pkpspeed para probar el desempeño de RSA

Puede ejecutar este ejemplo en Windows, Linux y sistemas operativos compatibles.

Linux

Utilice estas instrucciones para Linux y sistemas operativos compatibles.

```
/opt/cloudhsm/bin/pkpspeed -s CU user name -p password

SDK Version: 2.03

        Available Ciphers:
                AES_128
                AES_256
                3DES
                RSA  (non-CRT. modulus size can be 2048/3072)
                RSA_CRT (same as RSA)
For RSA, Exponent will be 65537

Current FIPS mode is: 00002
Enter the number of thread [1-10]: 3
Enter the cipher: RSA_CRT
Enter modulus length: 2048
Enter time duration in Secs: 60
Starting non-blocking speed test using data length of 245 bytes...
[Test duration is 60 seconds]

Do you want to use static key[y/n] (Make sure that KEK is available)?n
```

Windows

```
c:\Program Files\Amazon\CloudHSM>pkpspeed_blocking.exe -s CU user name -p password

Please select the test you want to run

RSA non-CRT------------------->A
RSA CRT----------------------->B
Basic 3DES CBC---------------->C
Basic AES--------------------->D
FIPS Random------------------->H
Random------------------------>I
AES GCM ---------------------->K

eXit------------------------->X
B

Running 4 threads for 25 sec

Enter mod size(2048/3072):2048
Do you want to use Token key[y/n]n
```

```
Do you want to use static key[y/n] (Make sure that KEK is available)?  n
OPERATIONS/second                821/1
OPERATIONS/second                833/1
OPERATIONS/second                845/1
OPERATIONS/second                835/1
OPERATIONS/second                837/1
OPERATIONS/second                836/1
OPERATIONS/second                837/1
OPERATIONS/second                849/1
OPERATIONS/second                841/1
OPERATIONS/second                856/1
OPERATIONS/second                841/1
OPERATIONS/second                847/1
OPERATIONS/second                838/1
OPERATIONS/second                843/1
OPERATIONS/second                852/1
OPERATIONS/second                837/
```

Example : uso de pkpspeed para probar el desempeño de AES

Linux

Utilice estas instrucciones para Linux y sistemas operativos compatibles.

```
/opt/cloudhsm/bin/pkpspeed -s <CU user name> -p <password>

SDK Version: 2.03

        Available Ciphers:
                AES_128
                AES_256
                3DES
                RSA  (non-CRT. modulus size can be 2048/3072)
                RSA_CRT (same as RSA)
For RSA, Exponent will be 65537

Current FIPS mode is: 00000002
Enter the number of thread [1-10]: 1
Enter the cipher: AES_256
Enter the data size [1-16200]: 8192
Enter time duration in Secs: 60
Starting non-blocking speed test using data length of 8192 bytes...
```

Windows

```
c:\Program Files\Amazon\CloudHSM>pkpspeed_blocking.exe -s CU user name -p password
login as USER
Initializing Cfm2 library
        SDK Version: 2.03

 Current FIPS mode is: 00000002
Please enter the number of threads [MAX=400] : 1
Please enter the time in seconds to run the test [MAX=600]: 20


Please select the test you want to run

RSA non-CRT------------------->A
RSA CRT----------------------->B
Basic 3DES CBC---------------->C
Basic AES--------------------->D
FIPS Random------------------->H
Random------------------------>I
AES GCM ---------------------->K

eXit------------------------->X
D

Running 1 threads for 20 sec

Enter the key size(128/192/256):256
Enter the size of the packet in bytes[1-16200]:8192
OPERATIONS/second                9/1
OPERATIONS/second                10/1
OPERATIONS/second                11/1
OPERATIONS/second                10/1
OPERATIONS/second                10/1
OPERATIONS/second                10/...
```

El usuario de SDK 5 de cliente contiene valores inconsistentes.

El comando `user list` devuelve una lista de todos los usuarios y las propiedades de los usuarios del clúster. Si alguna de las propiedades de un usuario tiene el valor “incoherente”, este usuario no está sincronizado en todo el clúster. Esto significa que el usuario existe con propiedades diferentes

en los distintos HSM del clúster. En función de qué propiedad no sea coherente, se pueden tomar diferentes medidas de reparación.

En la siguiente tabla se incluyen los pasos para resolver las incoherencias en el caso de un solo usuario. Si un solo usuario tiene varias incoherencias, resuélvalas siguiendo estos pasos de principio a fin. Si hay varios usuarios con incoherencias, revise esta lista para cada usuario y resuelva por completo las incoherencias de ese usuario antes de pasar al siguiente.

> **Note**
>
> Para realizar estos pasos, lo ideal es iniciar sesión como administrador. Si su cuenta de administrador no es coherente, siga estos pasos: inicie sesión con el administrador y repita los pasos hasta que todas las propiedades sean coherentes. Una vez que su cuenta de administrador sea coherente, puede usar esa cuenta de administrador para sincronizar a otros usuarios del clúster.

Propiedad incoherente	Ejemplo de salida de una lista de usuarios	Implicación	Método de recuperación
El «rol» del usuario es «incoherente»	`{ "username": "test_user", "role": "inconsistent ", "locked": "false", "mfa": [], "cluster-coverage": "full" }`	Este usuario es administrador CryptoUser en algunos HSM y administrador en otros HSM. Esto puede ocurrir si dos SDK intentan crear el mismo usuario, al mismo tiempo y con roles diferentes. Debe eliminar este usuario y volver a crearlo con el rol deseado.	1. Inicie sesión como administrador. 2. Elimine el usuario en todos los HSM: user delete --username **<user's name>** --role admin user delete --username **<user's name>** --role crypto-user 3. Cree el usuario con el rol deseado:

Propiedad incoherente	Ejemplo de salida de una lista de usuarios	Implicación	Método de recuperación
			user create --username **<user's name>** --role **<desired role>**

Propiedad incoherente	Ejemplo de salida de una lista de usuarios	Implicación	Método de recuperación

La «cobertura de clúster» del usuario es «incoherente»	<pre>{
	"username":
	"test_user",
	
	"role": "crypto-user",
	"locked":
	"false",
	"mfa": [],
	"cluster-coverage":
	"**inconsistent** "
	}</pre>
		Este usuario existe en un subconjunto de HSM del clúster. Esto puede suceder si un user create lo ha conseguido parcialmente o si un user delete lo ha hecho parcialmente.
		
		Debe finalizar la operación anterior, ya sea creando o quitando este usuario del clúster.
			Si el usuario no debería existir, siga estos pasos:
			
			1. Inicie sesión como administrador.
			2. Ejecute este comando:
			
			user delete --username**<user's name>** --role admin
			3. Ahora, ejecute el siguiente comando:
			
			user delete --username**<user's name>** --role crypto-user
			
			Si el usuario debería existir, siga estos pasos:
			
			1. Inicie sesión como administrador.
			2. Ejecute el siguiente comando:
			
			user create --username **<user's name>** --role **<desired role>**

Propiedad incoherente	Ejemplo de salida de una lista de usuarios	Implicación	Método de recuperación

El parámetro «bloqueado» del usuario es «incoherente» o «verdadero»	<pre>{
	"username": "test_user",
	"role": "crypto-user",
	"locked": **inconsistent** ,
	
	"mfa": [],
	"cluster-coverage": "full"
	}</pre>
		Este usuario está bloqueado en un subconjunto de HSM.
		
		Esto puede ocurrir si un usuario utiliza una contraseña incorrecta y solo se conecta a un subconjunto de HSM del clúster.
		
		Debe cambiar las credenciales del usuario para que sean coherentes en todo el clúster.
			Si el usuario tiene la MFA activada, siga estos pasos:
			
			1. Inicie sesión como administrador.
			2. Utilice el siguiente comando para desactivar temporalmente la MFA:
			
			user change-mfa token-sign --username **<user's name>** --role **<desired role>** --disable
			3. Cambie la contraseña del usuario para que pueda iniciar sesión en todos los HSM:
			
			user change-password --username **<user's name>** --role **<desired role>**
			
			Si MFA debe estar activo para el usuario, siga estos pasos:

Propiedad incoherente	Ejemplo de salida de una lista de usuarios	Implicación	Método de recuperación
			1. Haga que el usuario inicie sesión y vuelva a habilitar la MFA (esto requerirá que firme los tokens y proporcione su clave pública en un archivo PEM): user change-mfa token-sign --username **<user's name>** --role **<desired role>** —token <File>

Propiedad incoherente	Ejemplo de salida de una lista de usuarios	Implicación	Método de recuperación

El estado de la MFA es «incoherente»	{
	"username":
	"test_user",
	
	"role": "crypto-user",
	"locked":
	"false",
	"mfa": [
	{
	"strategy":
	"token-sign",
	"status":
	"**inconsistent** "
	}
	],
	"cluster-coverage":
	"full"
	}
		Este usuario tiene diferentes marcadores de MFA en los diferentes HSM del clúster.
		
		Esto puede suceder si una operación de MFA solo se completa en un subconjunto de HSM.
		
		Debe restablecer la contraseña del usuario y permitir que vuelva a activar la MFA.
			Si el usuario tiene la MFA activada, siga estos pasos:
			
			1. Inicie sesión como administrador.
			2. Utilice el siguiente comando para desactivar temporalmente la MFA:
			
			user change-mfa token-sign --username **<user's name>** --role **<desired role>** --disable
			3. También tendrá que cambiar la contraseña del usuario para que pueda iniciar sesión en todos los HSM:
			
			user change-password --username **<user's name>** --role **<desired role>**

Propiedad incoherente	Ejemplo de salida de una lista de usuarios	Implicación	Método de recuperación
			Si la MFA debe estar activa para el usuario, siga estos pasos: 1. Haga que el usuario inicie sesión y vuelva a habilitar la MFA (esto requerirá que firme los tokens y proporcione su clave pública en un archivo PEM): user change-mfa token-sign --username **<user's name>** --role **<desired role>** —token <File>

Se detectó un error durante la comprobación de disponibilidad de las claves.

Problema: un HSM devuelve el siguiente error:

```
Key <KEY HANDLE> does not meet the availability requirements - The key must be
 available on at least 2 HSMs before being used.
```

Causa: las comprobaciones de disponibilidad de claves buscan claves que, en raras ocasiones pero posibles, podrían perderse. Este error suele producirse en clústeres con un solo HSM o en clústeres con dos HSM durante un período en el que se reemplaza uno de ellos. En estas situaciones, es probable que las siguientes operaciones del cliente hayan provocado el error anterior:

- Se generó una nueva clave mediante un comando como key generate-symmetric o clave generate-asymmetric-pair .
- Se inició una operación de key list.
- Se inició una nueva instancia del SDK.

> **Note**
>
> OpenSSL bifurca con frecuencia nuevas instancias del SDK.

Resolución/recomendación: elija una de las siguientes acciones para evitar que se produzca este error:

- Utilice el parámetro --disable-key-availability-check para establecer la disponibilidad de la clave como falsa en el archivo de configuración de la herramienta de configuración. Para obtener más información, consulte la sección Parámetros de la herramienta de configuración.
- Si utiliza un clúster con dos HSM, evite realizar las operaciones que provocaron el error, excepto durante el código de inicialización.
- Aumente la cantidad de HSM en el clúster a al menos tres.

Extracción de claves con JCE

GetEncoded o getPrivateExponent GetS devuelve null

`getEncoded`, `getPrivateExponent` y `getS` devolverán un valor nulo porque están deshabilitados de forma predeterminada. Para habilitarlos, consulte Extracción de claves mediante JCE.

Si `getEncoded`, `getPrivateExponent` y `getS` devuelven un valor nulo después de activarlos, la clave no cumple los requisitos previos adecuados. Para obtener más información, consulte Extracción de claves mediante JCE.

GetEncoded o getPrivateExponent GetS devuelven bytes clave fuera del HSM

Usted o alguien con acceso a su sistema ha habilitado la extracción clara de claves. Consulte las páginas siguientes para obtener más información, como la forma de restablecer esta configuración al estado desactivado predeterminado.

- Extracción de claves mediante JCE
- Protección y extracción de claves de un HSM

Limitación de HSM

Cuando su carga de trabajo supere la capacidad de HSM de su clúster, recibirá mensajes de error indicando que los HSM están ocupados o limitados. Cuando esto sucede, es posible que vea una reducción del rendimiento o un aumento en la tasa de solicitudes de rechazo de los HSM. Además, los HSM pueden enviar los siguientes errores de ocupado.

Para SDK de cliente 5

- En PKCS11, los errores de ocupado se asignan a `CKR_FUNCTION_FAILED`. Este error puede producirse por varios motivos, pero si la limitación del HSM provoca este error, aparecerán en su registro las siguientes líneas de registro:
  - `[cloudhsm_provider::hsm1::hsm_connection::e2e_encryption::error] Failed to prepare E2E response. Error: Received error response code from Server. Response Code: 187`
  - `[cloudhsm_pkcs11::decryption::aes_gcm] Received error from the server. Error: This operation is already in progress. Internal error code: 0x000000BB`
- En JCE, los errores de ocupado se asignan a `com.amazonaws.cloudhsm.jce.jni.exception.InternalException: Unexpected error with the Provider: The HSM could not queue the request for processing.`.
- Los errores de ocupado de otros SDK muestran el siguiente mensaje: `Received error response code from Server. Response Code: 187`.

Para SDK de cliente 3

- En PKCS11, los errores de ocupado se asignan a errores `CKR_OPERATION_ACTIVE`.
- En JCE, los errores de ocupado se asignan a `CFM2Exception` con el estado de `0xBB (187)`. Las aplicaciones pueden usar la función `getStatus()` en `CFM2Exception` para comprobar qué estado devuelve el HSM.

- Los errores de ocupado de otros SDK mostrarán el siguiente mensaje: `HSM Error: HSM is already busy generating the keys(or random bytes) for another request.`

Resolución

Puede solucionar estos problemas realizando una o más de las siguientes acciones:

- Agregue comandos de reintento para las operaciones de HSM rechazadas en su capa de aplicaciones. Antes de activar los comandos de reintento, asegúrese de que el clúster tenga el tamaño adecuado para soportar los picos de carga.

  > **Note**
  >
  > En SDK de cliente 5.8.0 y versiones posteriores, los comandos de reintento están activados de forma predeterminada. Para obtener más información sobre la configuración de los comandos de reintento de cada SDK, consulte Configuraciones avanzadas para la herramienta de configuración SDK 5 de cliente.

- Añada más HSM a su clúster siguiendo las instrucciones que se indican en Añadir o eliminar los HSM de un clúster AWS CloudHSM.

  > **Important**
  >
  > Recomendamos realizar pruebas de carga en el clúster para determinar el pico de carga previsto y, a continuación, añadir un HSM adicional para garantizar una alta disponibilidad.

Mantener sincronizados los usuarios de los HSM del clúster

Para administrar los usuarios de sus HSM, utiliza una herramienta de línea de comandos conocida como cloudhsm_mgmt_util. AWS CloudHSM Solo se comunica con los HSM que están en el archivo de configuración de la herramienta. No tiene en cuenta otros HSM del clúster que no estén en el archivo de configuración.

AWS CloudHSM sincroniza las claves de los HSM con las de todos los demás HSM del clúster, pero no sincroniza los usuarios ni las políticas del HSM. Al utilizar cloudhsm_mgmt_util para administrar usuarios de HSM, estos cambios de usuarios podrían afectar únicamente a algunos de los HSM del clúster, es decir, los que están en el archivo de configuración de cloudhsm_mgmt_util. Esto puede

causar problemas al AWS CloudHSM sincronizar las claves entre los HSM del clúster, ya que es posible que los usuarios que poseen las claves no existan en todos los HSM del clúster.

Para evitar estos problemas, edite el archivo de configuración de cloudhsm_mgmt_util antes de administrar los usuarios. Para obtener más información, consulte ???.

Conexión perdida con el clúster

Al configurar el AWS CloudHSM cliente, proporcionó la dirección IP del primer HSM del clúster. Esta dirección IP se guarda en el archivo de configuración del AWS CloudHSM cliente. Cuando el cliente se inicia, intenta conectarse con esta dirección IP. Si no puede, por ejemplo, porque el HSM falla o porque usted lo ha eliminado, es posible que vea errores como los siguientes:

```
LIQUIDSECURITY: Daemon socket connection error
```

```
LIQUIDSECURITY: Invalid Operation
```

Para resolver estos errores, actualice el archivo de configuración con la dirección IP de un HSM activo y accesible del clúster.

Para actualizar el archivo de configuración del AWS CloudHSM cliente

1. Utilice una de las siguientes formas para buscar la dirección IP de un HSM activo de su clúster.

   - Consulte la pestaña HSMs de la página de detalles del clúster en la consola de AWS CloudHSM.
   - Utilice la AWS Command Line Interface (CLI) para describe-clustersejecutar el comando.

   Necesita esta dirección IP en un paso posterior.
2. Utilice el siguiente comando para detener el cliente.

   Amazon Linux

   ```
   $ sudo stop cloudhsm-client
   ```

   Amazon Linux 2

   ```
   $ sudo service cloudhsm-client stop
   ```

CentOS 7

```
$ sudo service cloudhsm-client stop
```

CentOS 8

```
$ sudo service cloudhsm-client stop
```

RHEL 7

```
$ sudo service cloudhsm-client stop
```

RHEL 8

```
$ sudo service cloudhsm-client stop
```

Ubuntu 16.04 LTS

```
$ sudo service cloudhsm-client stop
```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client stop
```

Windows

- Para la versión 1.1.2 y posteriores del cliente de Windows:

```
C:\Program Files\Amazon\CloudHSM>net.exe stop AWSCloudHSMClient
```

- Para la versión 1.1.1 y anteriores de clientes de Windows:

  Use Ctrl + C en la ventana de comandos donde inició el AWS CloudHSM cliente.

3. Utilice el siguiente comando para actualizar el archivo de configuración del cliente, proporcionando la dirección IP que obtuvo un paso anterior.

```
$ sudo /opt/cloudhsm/bin/configure -a <IP address>
```

4. Use el siguiente comando para iniciar el cliente de .

Amazon Linux

```
$ sudo start cloudhsm-client
```

Amazon Linux 2

```
$ sudo service cloudhsm-client start
```

CentOS 7

```
$ sudo service cloudhsm-client start
```

CentOS 8

```
$ sudo service cloudhsm-client start
```

RHEL 7

```
$ sudo service cloudhsm-client start
```

RHEL 8

```
$ sudo service cloudhsm-client start
```

Ubuntu 16.04 LTS

```
$ sudo service cloudhsm-client start
```

Ubuntu 18.04 LTS

```
$ sudo service cloudhsm-client start
```

Windows

- Para la versión 1.1.2 y posteriores del cliente de Windows:

```
C:\Program Files\Amazon\CloudHSM>net.exe start AWSCloudHSMClient
```

- Para la versión 1.1.1 y anteriores de clientes de Windows:

```
C:\Program Files\Amazon\CloudHSM>start "cloudhsm_client" cloudhsm_client.exe
 C:\ProgramData\Amazon\CloudHSM\data\cloudhsm_client.cfg
```

Faltan registros de AWS CloudHSM auditoría CloudWatch

Si creó un clúster antes del 20 de enero de 2018, tendrá que configurar manualmente una función vinculada al servicio para poder habilitar la entrega de los registros de auditoría de ese clúster. Para obtener instrucciones acerca de cómo habilitar una función vinculado al servicio en un clúster de HSM, consulte Descripción de las funciones vinculadas al servicio y Creación de una función vinculada al servicio en la guía Usuario de IAM.

IV personalizados con una longitud no compatible para el encapsulamiento de claves AES

En este tema de solución de problemas aprenderá a detectar si su aplicación genera claves encapsuladas irrecuperables. Si se ve afectado por este problema, utilice este tema para solucionar el problema.

Temas

- Determine si el código genera claves encapsuladas irrecuperables.
- Acciones que debe adoptar si su código genera claves encapsuladas irrecuperables

Determine si el código genera claves encapsuladas irrecuperables.

Solo se verás afectado si se dan todas las condiciones que se indican a continuación:

Condición	¿Cómo lo sé?
Su aplicación utiliza la biblioteca PKCS #11.	La biblioteca PKCS #11 se instala como el archivo `libpkcs11.so` en su carpeta `/opt/cloudhsm/lib` . Las aplicaciones escritas en

Condición	¿Cómo lo sé?
	lenguaje C suelen utilizar la biblioteca PKCS #11 directamente, mientras que las aplicacio nes escritas en Java pueden utilizar la bibliotec a indirectamente mediante una capa de abstracción de Java. Si utiliza Windows, esto NO le afecta, ya que la biblioteca PKCS #11 no está disponible actualmente para Windows.
Su aplicación usa específicamente la versión 3.0.0 de la biblioteca PKCS #11.	Si ha recibido un correo electrónico del AWS CloudHSM equipo, es probable que esté utilizando la versión 3.0.0 de la biblioteca PKCS #11. Para comprobar la versión del software en las instancias de su aplicación, utilice este comando: `rpm -qa | grep ^cloudhsm`
Las claves se encapsulan mediante el encapsulamiento de claves AES.	El encapsulamiento de claves AES significa que se utiliza una clave AES para encapsula r alguna otra clave. El nombre del mecanismo correspondiente es `CKM_AES_KEY_WRAP` . Se usa con la función `C_WrapKey` . Otros mecanismos de encapsulamiento basados en AES que utilizan vectores de inicialización (IV), como `CKM_AES_GCM` y `CKM_CLOUD HSM_AES_GCM` , no se ven afectados por este problema. Obtenga más información sobre funciones y mecanismos.

Condición	¿Cómo lo sé?
Se especifica un IV personalizado al solicitar el encapsulamiento de claves AES, y la longitud de este IV es inferior a 8	El encapsulamiento de claves AES generalmente se inicializa con una estructura de `CK_MECHANISM` como la siguiente: `CK_MECHANISM mech = {CKM_AES_KEY_WRAP, IV_POINTER, IV_LENGTH};` Esta cuestión solo le afecta a usted si: • IV_POINTER no es NULL • IV_LENGTH tiene menos de 8 bytes

Si no se dan todas las condiciones anteriores, puede dejar de leer. Las claves encapsuladas se pueden desencapsular correctamente y este problema no le afecta. De lo contrario, consulte the section called “Acciones que debe adoptar si su código genera claves encapsuladas irrecuperables”.

Acciones que debe adoptar si su código genera claves encapsuladas irrecuperables

Debe seguir los tres pasos siguientes:

1. Actualice inmediatamente su biblioteca PKCS #11 a una versión más reciente.
   - Biblioteca de PKCS #11 más reciente para Amazon Linux, Centos 6 y RHEL 6
   - Biblioteca PKCS #11 más reciente para Amazon Linux 2, CentOS 7 y RHEL 7
   - Biblioteca de PKCS #11 más reciente para Ubuntu 16.04 LTS
2. Actualice su software para usar un IV que cumpla con los estándares

   Le recomendamos encarecidamente que siga nuestro ejemplo de código y simplemente especifique un IV NULO, lo que provocará que el HSM utilice el IV predeterminado que cumple con los estándares. Como alternativa, puede especificar explícitamente el IV como `0xA6A6A6A6A6A6A6A6` con una longitud de IV correspondiente de 8. No recomendamos utilizar ningún otro IV para el encapsulamiento de claves AES y deshabilitaremos explícitamente los IV personalizados para el encapsulamiento de claves AES en una futura versión de la biblioteca PKCS #11.

En aes_wrapping.c on aparece un ejemplo de código para especificar correctamente el IV. GitHub

3. Identifique y recupere las claves encapsuladas existentes.

   Debe identificar las claves que encapsuló con la versión 3.0.0 de la biblioteca PKCS #11 y, a continuación, ponerse en contacto con el servicio de asistencia para obtener ayuda (https://aws.amazon.com/support) a la hora de recuperar estas claves.

> **Important**
>
> Este problema solo afecta a las claves encapsuladas en la versión 3.0.0 de la biblioteca PKCS #11. Puede encapsular las claves utilizando versiones anteriores (2.0.4 y paquetes con números inferiores) o versiones posteriores (3.0.1 y paquetes con números superiores) de la biblioteca PKCS #11.

Solución de errores de creación de clústeres

Al crear un clúster, AWS CloudHSM crea el rol AWSServiceRoleForCloudHSM vinculado al servicio, si el rol aún no existe. Si AWS CloudHSM no se puede crear el rol vinculado al servicio, es posible que se produzca un error al intentar crear un clúster.

En este tema se explica cómo resolver los problemas más habituales para que pueda crear un clúster correctamente. Tiene que crear este rol una única vez. Una vez que el rol vinculado al servicio se cree en su cuenta, podrá utilizar cualquiera de los métodos admitidos para crear y administrar clústeres adicionales.

En las secciones siguientes se ofrecen sugerencias para solucionar errores de creación de clústeres relacionados con el rol vinculado al servicio. Si prueba estas sugerencias pero sigue sin poder crear un clúster, póngase en contacto con AWS Support. Para obtener más información sobre el rol AWSServiceRoleForCloudHSM vinculado al servicio, consulte. Funciones vinculadas al servicio para AWS CloudHSM

Temas

- Agregar el permiso que falta.
- Crear el rol vinculado a un servicio manualmente.
- Uso de un usuario no federado

Agregar el permiso que falta.

Para crear un rol vinculado a un servicio, el usuario ha de tener el permiso `iam:CreateServiceLinkedRole`. Si el usuario de IAM que está creando el clúster no tiene este permiso, se producirá un error en el proceso de creación del clúster al intentar crear el rol vinculado al servicio en su cuenta. AWS

Cuando el error se produce porque falta un permiso, el mensaje de error contiene el texto siguiente.

```
This operation requires that the caller have permission to call
 iam:CreateServiceLinkedRole to create the CloudHSM Service Linked Role.
```

Para solucionar este error, dé al usuario de IAM que crea el clúster el permiso `AdministratorAccess` o añada el permiso `iam:CreateServiceLinkedRole` a la política de IAM del usuario. Para leer las instrucciones, consulte Agregar permisos a un usuario nuevo o existente.

A continuación, intente volver a crear el clúster.

Crear el rol vinculado a un servicio manualmente.

Puede usar la consola, la CLI o la API de IAM para crear el rol vinculado al AWSServiceRoleForCloudHSM servicio. Para obtener más información, consulte Crear un rol vinculado a un servicio en la Guía del usuario de IAM.

Uso de un usuario no federado

Los usuarios federados, cuyas credenciales se originan fuera de AWS, pueden realizar muchas de las tareas de un usuario no federado. No obstante, AWS no permite a los usuarios realizar llamadas a la API para crear un rol vinculado al servicio desde un punto de enlace federado.

Para solucionar este problema, cree un usuario no federado con el permiso `iam:CreateServiceLinkedRole` o conceda a un usuario no federado ya existente el permiso `iam:CreateServiceLinkedRole`. A continuación, pida a ese usuario que cree un clúster desde la CLI. Esto creará el rol vinculado al servicio en su cuenta.

Una vez que haya creado el rol vinculado al servicio, podrá eliminar el clúster que el usuario no federado creó. La eliminación del clúster no afecta al rol. A partir de entonces, cualquier usuario con los permisos necesarios, incluidos los usuarios federados, podrá crear AWS CloudHSM clústeres en su cuenta.

Para comprobar que se creó el rol, abra la consola de IAM en https://console.aws.amazon.com/iam/ y seleccione Roles. O utilice el comando get-role de IAM en la CLI.

```
$  aws iam get-role --role-name AWSServiceRoleForCloudHSM
{
    "Role": {
        "Description": "Role for CloudHSM service operations",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "cloudhsm.amazonaws.com"
                    }
                }
            ]
        },
        "RoleId": "AROAJ4I6WN5QVGG5G7CBY",
        "CreateDate": "2017-12-19T20:53:12Z",
        "RoleName": "AWSServiceRoleForCloudHSM",
        "Path": "/aws-service-role/cloudhsm.amazonaws.com/",
        "Arn": "arn:aws:iam::111122223333:role/aws-service-role/cloudhsm.amazonaws.com/
AWSServiceRoleForCloudHSM"
    }
}
```

Recuperación de registros de configuración de los clientes

AWS CloudHSM ofrece herramientas para Client SDK 3 y Client SDK 5 para recopilar información sobre su entorno para que AWS Support resuelva problemas.

Temas

- Herramienta de soporte para SDK 5 de cliente
- Herramienta de soporte para el SDK 3 de cliente

Herramienta de soporte para SDK 5 de cliente

El script extrae la siguiente información:

- El archivo de configuración para el componente de SDK 5 de cliente
- Archivos de registro disponibles
- Versión actual del sistema operativo
- Información de paquetes

Ejecutar la herramienta de información para SDK 5 de cliente

El SDK 5 de cliente incluye una herramienta de soporte al cliente para cada componente, pero todas las herramientas funcionan igual. Ejecute la herramienta para crear un archivo de salida con toda la información recopilada.

Las herramientas utilizan una sintaxis como la siguiente:

```
[ pkcs11 | dyn | jce ]_info
```

Por ejemplo, para recopilar información de soporte desde un host Linux que ejecuta la biblioteca PKCS #11 y hacer que el sistema escriba en el directorio predeterminado, ejecute este comando:

```
/opt/cloudhsm/bin/pkcs11_info
```

La herramienta crea el archivo de salida dentro del directorio `/tmp`.

PKCS #11 library

Para recopilar datos de soporte para la biblioteca PKCS #11 en Linux

- Utilice la herramienta de soporte para recopilar datos.

```
/opt/cloudhsm/bin/pkcs11_info
```

Para recopilar datos de soporte para la biblioteca PKCS #11 en Windows

- Utilice la herramienta de soporte para recopilar datos.

```
C:\Program Files\Amazon\CloudHSM\bin\pkcs11_info.exe
```

OpenSSL Dynamic Engine

Para recopilar datos de soporte para el motor dinámico de OpenSSL en Linux

- Utilice la herramienta de soporte para recopilar datos.

```
/opt/cloudhsm/bin/dyn_info
```

JCE provider

Para recopilar datos de soporte para el proveedor de JCE en Linux

- Utilice la herramienta de soporte para recopilar datos.

```
/opt/cloudhsm/bin/jce_info
```

Para recopilar datos de soporte para el proveedor de JCE en Windows

- Utilice la herramienta de soporte para recopilar datos.

```
C:\Program Files\Amazon\CloudHSM\bin\jce_info.exe
```

Recuperación de registros de un entorno sin servidor

Para configurar entornos sin servidor, como Fargate o Lambda, le recomendamos que configure AWS CloudHSM el tipo de registro en. `term` Una vez configurado`term`, el entorno sin servidor podrá generar resultados en. CloudWatch

Para obtener los registros del cliente CloudWatch, consulte Trabajar con grupos de registros y flujos de CloudWatch registros en la Guía del usuario de Amazon Logs.

Herramienta de soporte para el SDK 3 de cliente

El script extrae la siguiente información:

- El sistema operativo y la versión actual
- La información de configuración del cliente de los archivos `cloudhsm_client.cfg`, `cloudhsm_mgmt_util.cfg` y `application.cfg`

- Los registros del cliente de la ubicación específica de la plataforma
- Información sobre el clúster y el HSM mediante cloudhsm_mgmt_util
- Información de OpenSSL
- El cliente y la versión de compilación actuales
- La versión del instalador

Ejecutar la herramienta de información para SDK 3 de cliente

El script crea un archivo de salida con toda la información recopilada. El script crea el archivo de salida dentro del directorio `/tmp`.

Linux: `/opt/cloudhsm/bin/client_info`

Windows: `C:\Program Files\Amazon\CloudHSM\client_info`

> **Warning**
>
> Este script tiene un problema conocido en las versiones 3.1.0 a 3.3.1 de SDK 3 de cliente. Le recomendamos encarecidamente que efectúe la actualización a la versión 3.3.2, que incluye una solución para este problema. Antes de utilizar esta herramienta, consulte la página Problemas conocidos para obtener más información.

AWS CloudHSM cuotas

Las cuotas, anteriormente conocidas como límites, son los valores asignados a los AWS recursos. Las siguientes cuotas se aplican a sus AWS CloudHSM recursos por AWS región y AWS cuenta. La cuota predeterminada es el valor inicial aplicado y estos valores se muestran en la siguiente tabla. AWS Las cuotas ajustables pueden incrementarse por encima de la cuota predeterminada.

Service Quotas

Recurso	Cuota predeterminada	¿Ajustable?
Clústeres	4	Sí
HSM	6	Sí
HSM por clúster	28	No

La forma recomendada de solicitar un aumento de cuota es abrir la consola de cuotas de servicio. En la consola, elija el servicio y la cuota, y envíe la solicitud. Para obtener más información, consulte la documentación de las cuotas de servicio.

Las cuotas de la siguiente tabla de cuotas del sistema no son ajustables.

Cuotas del sistema

Recurso	Cuota
Máximo de claves por clúster	3300
Máximo de usuarios por clúster	1 024
Longitud máxima de un nombre de usuario	31 caracteres
Longitud de contraseña obligatoria	De 7 a 32 caracteres
Número máximo de conexiones de cliente simultáneas por clúster [1]	900

Recurso	Cuota
Máximo de sesiones de PKCS#11 por aplicación	1 024

[1] Para SDK 3 de cliente, una conexión de cliente es un daemon de cliente. Para SDK 5 de cliente, una conexión de cliente es una aplicación.

Para obtener más información, consulte Recursos del sistema.

Recursos del sistema

Las cuotas de recursos del sistema son cuotas de lo que el AWS CloudHSM cliente puede usar cuando se ejecuta.

Los descriptores de archivos son un mecanismo del sistema operativo para identificar y administrar los archivos abiertos en cada proceso.

El demonio del cliente de CloudHSM utiliza descriptores de archivos para administrar las conexiones entre las aplicaciones y el cliente, así como entre el cliente y el servidor.

De forma predeterminada, la configuración del cliente de CloudHSM asignará 3000 descriptores de archivo. Este valor predeterminado está diseñado para ofrecer una sesión óptima y brindar capacidad de subprocesamiento entre el demonio del cliente y los HSM.

En circunstancias excepcionales, si ejecuta el cliente en un entorno con recursos restringidos, tal vez sea necesario modificar estos valores predeterminados.

> **Note**
>
> Al cambiar estos valores, el rendimiento del cliente de CloudHSM podría verse afectado o la aplicación podría quedar inoperativa.

1. Edite el archivo `/etc/security/limits.d/cloudhsm.conf`.

```
#
```

```
# DO NOT EDIT THIS FILE
#
hsmuser soft nofile 3000
hsmuser hard nofile 3000
```

2. Modifique los valores numéricos según sea necesario.

> **Note**
>
> La cuota de `soft` debe ser inferior o igual a la cuota de `hard`.

3. Reinicie el proceso del demonio del cliente de CloudHSM.

> **Note**
>
> Esta opción de configuración no está disponible en las plataformas de Microsoft Windows.

Descargas para AWS CloudHSM Client SDK

Descargas

En marzo de 2021, AWS CloudHSM lanzó la versión 5.0.0 del SDK de cliente, que presenta un SDK de cliente completamente nuevo con diferentes requisitos, capacidades y soporte de plataforma.

El Client SDK 5 es totalmente compatible con los entornos de producción y ofrece los mismos componentes y el mismo nivel de soporte que el Client SDK 3, con la excepción del soporte para los proveedores de GNC y KSP. Para obtener más información, consulte Comparación de componentes de SDK de cliente.

> **Note**
>
> Para obtener información sobre las plataformas compatibles con cada SDK de cliente, consulte y. Plataformas compatibles con SDK 5 de cliente Plataformas compatibles con SDK 3 de cliente

Versión más reciente

En esta sección se incluye la versión más reciente del SDK de cliente.

Versión 5 del SDK de cliente: versión 5.12.0

Amazon Linux 2

Descargue la versión 5.12.0 del software para Amazon Linux 2 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 383baed4a861391eb0923c0d9cf451851c6dd02d7d6a9e9cc3638c60bf300ef2)
- Motor dinámico de OpenSSL (SHA256 checksum f7aba68787a4c975f3e9f4ead28c2c28adc787ca0babebc070a928d226ff330a)
- Proveedor de JCE (SHA256 checksum 1f75f1a5d428b18ce2dc6ce8e17923009895c2545e2d04d76dafd6da914c0b4e)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)

- CLI de CloudHSM (SHA256 checksum 4c27fae1ef5fd1642c04514ec84ad4cab78f59a32eb3fce59b51805c44b25295)

Descargue la versión 5.12.0 del software para Amazon Linux 2 en la arquitectura ARM64:

- Biblioteca PKCS #11 (SHA256 checksum c28a1f27e23e6ab1550dab6a353c6c9338a391a84d57f4ac99a1a3a9810c753f)
- Motor dinámico de OpenSSL (SHA256 checksum 7d2e864c31c13f55443c1b1d04589fbdd4558fe103954de4384691e2c429a872)
- Proveedor de JCE (SHA256 checksum e9a35eb87b2f257c47fb083d286deb835da45858b2d89759ca7d5bb4ef747b4b)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum 28b6f918912b5c63bf10018824b642a805b309c21947a1d0ebbdc44647e80554)

Amazon Linux 2023

Descargue la versión 5.12.0 del software para Amazon Linux 2023 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 02801365cba449c5238a4e5ad3df1ddf7edd00ade976f47e956e885286503f3f)
- Motor dinámico de OpenSSL (SHA256 checksum 0abed69a7c6acaafdaabdcc5fab7d56611ffd94f5480cade6f8beace9aeae056)
- Proveedor de JCE (SHA256 checksum 3d5d9a903d3a216eca40f92dbb0b4030b7a86ad7ceee8d62241c97a6e1881e25)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum f96671d882b862033bba0b3633448dc6a26e45a25063e29b79a5cd4b7fc4945c)

Descargue la versión 5.12.0 del software para Amazon Linux 2023 en la arquitectura ARM64:

- Biblioteca PKCS #11 (SHA256 checksum 53d05006b46bda8e9c1dd76e8307a780bfe0a67b10a9a87723c97f94e29f5b8e)
- Motor dinámico de OpenSSL (SHA256 checksum ec1cca8e01b3303ff9473eeef6b33dc85b6affac7a47387b098905f9f2fc85ba)
- Proveedor de JCE (SHA256 checksum c828ae56f46233215b9f35798b5859ebdac962af442acbc457081c3baaa44f11)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum ddd5dcd68d01f4fafaf13dc0b4ddcf98e3731ed51bdd51f85535b29353644a9f)

CentOS 7 (7.8+)

Descargue el software de la versión 5.12.0 para Centos 7 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 383baed4a861391eb0923c0d9cf451851c6dd02d7d6a9e9cc3638c60bf300ef2)
- Motor dinámico de OpenSSL (SHA256 checksum f7aba68787a4c975f3e9f4ead28c2c28adc787ca0babebc070a928d226ff330a)
- Proveedor de JCE (SHA256 checksum 1f75f1a5d428b18ce2dc6ce8e17923009895c2545e2d04d76dafd6da914c0b4e)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum 4c27fae1ef5fd1642c04514ec84ad4cab78f59a32eb3fce59b51805c44b25295)

RHEL 7 (7.8+)

Descargue la versión 5.12.0 del software para RHEL 7 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 383baed4a861391eb0923c0d9cf451851c6dd02d7d6a9e9cc3638c60bf300ef2)
- Motor dinámico de OpenSSL (SHA256 checksum f7aba68787a4c975f3e9f4ead28c2c28adc787ca0babebc070a928d226ff330a)
- Proveedor de JCE (SHA256 checksum 1f75f1a5d428b18ce2dc6ce8e17923009895c2545e2d04d76dafd6da914c0b4e)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum 4c27fae1ef5fd1642c04514ec84ad4cab78f59a32eb3fce59b51805c44b25295)

RHEL 8 (8.3+)

Descargue la versión 5.12.0 del software para RHEL 8 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 6e51e95122fd0991278888287f0c408808b26fb5f1196c46168477b9090fc478)
- Motor dinámico de OpenSSL (SHA256 checksum 1f1d52ff7af6c537d8cfeb5973c691a9d90a518accd685ff9b66cd78daf98928)
- Proveedor de JCE (SHA256 checksum 156944607de987d6b39bd8a2d21ccd294c01377a9e35f9f15f8b0f4c8bb90033)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum 351e802f79dd2d0b5f7d23bb74c146be05e5169b603c9aace24189094a45a35d)

RHEL 9 (9.2+)

Descargue la versión 5.12.0 del software para RHEL 9 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum d1b2f4ac7e6e0c18e788512e7726bc68b571d99a1442ce2f2e80f4b0f9956266)
- Motor dinámico de OpenSSL (SHA256 checksum cf86a3f17cd6c51969d4ce80c1e3ea6513b995611be7e2e72e5e5233c71d6add)
- Proveedor de JCE (SHA256 checksum ae89e256eb89ec6b4fa0f001e7a4e1d8f1c08530423e81aa74d69a17b25d9a99)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum dfe6fe5d890c33b2f5d38f906ade113b06c8c05f3427a327744c454e7302f1a5)

Descargue la versión 5.12.0 del software para RHEL 9 en la arquitectura ARM64:

- Biblioteca PKCS #11 (SHA256 checksum cad72a6ab2232b4c38b90d7c62147520b975d646773dd90d7be897fa0a537d2d)
- Motor dinámico de OpenSSL (SHA256 checksum ad751f756530a2317c3c64380ea3a07865b13e1874fab0e61ac530b21487c7fb)
- Proveedor de JCE (SHA256 checksum d204e69acfbb90996fb08ae3573607b65630b1124fb379e078c002d55ac07766)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum c0f412cc59bafd235e046cdc1a0c5d330f2d72f7d6434672e9522f86bc945090)

Ubuntu 20.04 LTS

Descargue la versión 5.12.0 del software para Ubuntu 20.04 LTS en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum d37b1f872eb2b1ab34303d5b8b803daa925902b645c57c6e15a28bb6321e0f42)
- Motor dinámico de OpenSSL (SHA256 checksum cdc6e737652556b57d26d8816b2bc9820128cb3919360660b6f7fe65f9d39e3f)
- Proveedor de JCE (SHA256 checksum f567a08344414a4776e1c5a9715657476925ca32695c4c2dd84a4f3fc5dc1615)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum f2ee5ad01c5018fc3670f602228fd71087228cd3923bf5b9bc73e4d7084dac6c)

Ubuntu 22.04 LTS

Descargue la versión 5.12.0 del software para Ubuntu 22.04 LTS en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 0e78928acd7a1662e4b07b15d5c3ccb88714ff89e47b991c8ab6e4c2229ee5aa)
- Motor dinámico de OpenSSL (SHA256 checksum 4f3168745edc5592234891a7b1d82b179a4947e87c72fade1be3bad58b7ed1a3)
- Proveedor de JCE (SHA256 checksum d4c3655cdc2b00d1ab5ceafac94dfbc5c5244ed20e10fdd9db9f4e741e013733)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum d00bbacb6f2e57bd92d832a2bd11cadede972f8e82cc402ec0684b9c6b23123c)

Descargue la versión 5.12.0 del software para Ubuntu 22.04 LTS en la arquitectura ARM64:

- Biblioteca PKCS #11 (SHA256 checksum 0c1121535c523acb864215338292bab32acee438357878b5fc0b6d268713b86f)
- Motor dinámico de OpenSSL (SHA256 checksum dc7a219302021570bc8c36674d2bd33165557bb2f9a0af8fdf114f1b85a70d84)
- Proveedor de JCE (SHA256 checksum af3834a10081f1e4e7894275c8b9c7b7649b8de3b6f0aeb0781a3358183a9046)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum baa253ac62c2fbcc5712561e0fb0feb25461efc3ce68cf86d4c7bf0af0f14a34)

Windows Server 2016

Descargue la versión 5.12.0 del software para Windows Server 2016 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 11c3255fcc90b47810cfe4b2f71d56a006d295efccdd90f0d3f2dec5d2bab893)
- Proveedor de JCE (SHA256 checksum 09001458196590f54352c0c8986f442003bfc2db71bac6392ce512899d386806)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum b446ad1387fe406dcc0a12b6de86fa98e9db4a18f9829b745efb87750c6e31ea)

Windows Server 2019

Descargue la versión 5.12.0 del software para Windows Server 2019 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 11c3255fcc90b47810cfe4b2f71d56a006d295efccdd90f0d3f2dec5d2bab893)
- Proveedor de JCE (SHA256 checksum 09001458196590f54352c0c8986f442003bfc2db71bac6392ce512899d386806)
  - Javadocs para AWS CloudHSM (SHA256 checksum 7158bc80e3b5b0915d83c39d4c060060a43a79cc407b1f783383b9e20bc5ff43)
- CLI de CloudHSM (SHA256 checksum b446ad1387fe406dcc0a12b6de86fa98e9db4a18f9829b745efb87750c6e31ea)

El Client SDK 5.12.0 añade compatibilidad con ARM a varias plataformas y mejora el rendimiento de todos los SDK. Se han agregado nuevas funciones al proveedor de CLI y JCE de CloudHSM.

Compatibilidad con plataformas

- Se agregó soporte para Amazon Linux 2023 en la arquitectura ARM64 para todos los SDK.
- Se agregó soporte para Red Hat Enterprise Linux 9 (9.2+) en la arquitectura ARM64 para todos los SDK.
- Se agregó soporte para Ubuntu 22.04 LTS en la arquitectura ARM64 para todos los SDK.

La CLI de CloudHSM

- Se agregó el siguiente comando:
  - réplica clave
- Se agregó soporte para conectarse a varios clústeres. Para obtener más información, consulte Conexión a varios clústeres con CLI.

Proveedor de JCE

- Se agregó `KeyReferenceSpec` para recuperar claves utilizando`KeyStoreWithAttributes`.
- Se agregó `getKeys` para recuperar varias claves a la vez usando. `KeyStoreWithAttributes`

Mejoras en el rendimiento

- Mejoras en el rendimiento del NoPadding funcionamiento del AES CBC en todos los SDK.

Versiones anteriores del SDK de cliente

En esta sección se enumeran las versiones anteriores del SDK de cliente.

Versión 5.11.0

Amazon Linux 2

Descargue la versión 5.11.0 del software para Amazon Linux 2 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 9fc0cd7cf003a7cb7e42dbd19671d58a97fc3b3d871d284dc6ae7fd226598772)
- Motor dinámico de OpenSSL (SHA256 checksum 1df6669c971440d446890b0fbeb74125a423df7b14e7ac4577347be7ef176572)
- Proveedor de JCE (SHA256 checksum 148a3f1de55a68e3bb525fb2994645333a52c2e9e46946dd8d90fcbc90ab64fd)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum a68f4a56d4c539cfcc8a1e56e19b5ff385bb24936ea5f349255b4e9bfbee9aab)

Descargue la versión 5.11.0 del software para Amazon Linux 2 en la arquitectura ARM64:

- Biblioteca PKCS #11 (SHA256 checksum 5ac16449ec149c9b5e7776865803245ab17d0f1ad56df80173840c5e8d257b19)
- Motor dinámico de OpenSSL (SHA256 checksum 28c2eb7f3f60172b0186e5c25f71bb7341537058a71f288673936766048083c1)
- Proveedor de JCE (SHA256 checksum 06c9d9d281c12b1d2bd9a7b601d6317e46cedf175706bbfa3e4dcaed6ba05448)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum 218982bb17aa751969a7866b0a9ff27e7aa5007a07817627d9cc1f7d60a78160)

Amazon Linux 2023

Descargue la versión 5.11.0 del software para Amazon Linux 2023 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 55310ab333d18bcfabdc4b74115b040386b4508934bdff93e1d054c4c4a6f9ea)
- Motor dinámico de OpenSSL (SHA256 checksum f3d4934dc872a9b5212a180b9814ca2af3eca01ee228a8725563f1770add0dce)

- Proveedor de JCE (SHA256 checksum 757d3abb515aeb08f4b1c83970ee0979399efee00ee78c9a9dbec05f4ed9768d)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum 22af8f0501ff9a45a9e0683a408a63771c2c06c66abf5478d310d6d32e013555)

CentOS 7 (7.8+)

Descargue el software de la versión 5.11.0 para Centos 7 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 9fc0cd7cf003a7cb7e42dbd19671d58a97fc3b3d871d284dc6ae7fd226598772)
- Motor dinámico de OpenSSL (SHA256 checksum 1df6669c971440d446890b0fbeb74125a423df7b14e7ac4577347be7ef176572)
- Proveedor de JCE (SHA256 checksum 148a3f1de55a68e3bb525fb2994645333a52c2e9e46946dd8d90fcbc90ab64fd)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum a68f4a56d4c539cfcc8a1e56e19b5ff385bb24936ea5f349255b4e9bfbee9aab)

RHEL 7 (7.8+)

Descargue la versión 5.11.0 del software para RHEL 7 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 9fc0cd7cf003a7cb7e42dbd19671d58a97fc3b3d871d284dc6ae7fd226598772)
- Motor dinámico de OpenSSL (SHA256 checksum 1df6669c971440d446890b0fbeb74125a423df7b14e7ac4577347be7ef176572)
- Proveedor de JCE (SHA256 checksum 148a3f1de55a68e3bb525fb2994645333a52c2e9e46946dd8d90fcbc90ab64fd)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum a68f4a56d4c539cfcc8a1e56e19b5ff385bb24936ea5f349255b4e9bfbee9aab)

RHEL 8 (8.3+)

Descargue la versión 5.11.0 del software para RHEL 8 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum b95b9f588656fb14fd08bb66ce0e0da807b96daa38348dec07a508c9bef7403a)
- Motor dinámico de OpenSSL (SHA256 checksum 7bb437b91a52e863b2b00ff7f427ce22522026daf757be873ee031ec6ffffd88)

- Proveedor de JCE (SHA256 checksum e0db887e05eb535314f4d99f21da12d87d35ebb8baf9726f4ce8f01d9df0ea01)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum 8485b5a6d679767ca9b4f611718159a643cf3e85090a8e4d20fe53c3707e25c3)

RHEL 9 (9.2+)

Descargue la versión 5.11.0 del software para RHEL 9 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 87b56a20accf67df53a203b7f115655b2acfaec4516682d4976d9475b10bec8e)
- Motor dinámico de OpenSSL (SHA256 checksum 83a6b58572e985df937beede4b10e867b0ac6050ace8010dc8d535be365d2747)
- Proveedor de JCE (SHA256 checksum ee95213d02d913250478d0793d6dd578e5c54d765e635c7468a49bdff4c2a6f3)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum 7e168ed3bef8e9c5110645e9960680e9a57f7b94e16aec71422e3c67ebc58fb5)

Ubuntu 20.04 LTS

Descargue la versión 5.11.0 del software para Ubuntu 20.04 LTS en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum abc3a339d1fe5850db65620804e9a910f8b4f913624ef9b7189f2f0df1825c01)
- Motor dinámico de OpenSSL (SHA256 checksum 075fc3f9974d552f27ad67fa92c8abff31b756b9add875b8cd4957e6801583a4)
- Proveedor de JCE (SHA256 checksum 5de45c519133a0dae8da3ac01809db7974be25c14c15eb773fc5c972c0178c13)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum 83e0e4505a063792c19feb3d4cfd032b9089091916168d92b0f51a967a007734)

Ubuntu 22.04 LTS

Descargue la versión 5.11.0 del software para Ubuntu 22.04 LTS en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum b8f20be125c8530b2a7bd945956e9c04296fba5634af408b40be4e03bdbad72a)

- Motor dinámico de OpenSSL (SHA256 checksum d728c156eb4ee5c67159e57d6b092785800baa5fb61c14d64f460a8b8f53a778)
- Proveedor de JCE (SHA256 checksum 44e943b8cd1176ad666e249342687744a280c6222df58b5a9f084c932f628284)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum 8ccf5389d459611be813e42d7f9d040090f94f3fe88f9d110bcfb25e9619e4a7)

Windows Server 2016

Descargue la versión 5.11.0 del software para Windows Server 2016 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum aa4bce5be15bbe0978b7205c619bb91c55a8e0f1f4636be311f24878f7709e07)
- Proveedor de JCE (SHA256 checksum 004cdb9ecb4a4d72458084997de7f562fb76a4e2f0567009f1dfafa7b2bded47)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum 679795db759fda4823232142297a281e21a7d6f32cb5ddd6ac4c479866fa33b7)

Windows Server 2019

Descargue la versión 5.11.0 del software para Windows Server 2019 en la arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum aa4bce5be15bbe0978b7205c619bb91c55a8e0f1f4636be311f24878f7709e07)
- Proveedor de JCE (SHA256 checksum 004cdb9ecb4a4d72458084997de7f562fb76a4e2f0567009f1dfafa7b2bded47)
  - Javadocs para AWS CloudHSM (SHA256 checksum fb469ae53b516338f3326b402b15b7b84912801a8c25a28cd31a5da0631cd3c5)
- CLI de CloudHSM (SHA256 checksum 679795db759fda4823232142297a281e21a7d6f32cb5ddd6ac4c479866fa33b7)

El SDK de cliente 5.11.0 añade nuevas funciones, mejora la estabilidad e incluye correcciones de errores para todos los SDK.

Compatibilidad con plataformas

- Se agregó compatibilidad con Amazon Linux 2023 y RHEL 9 (9.2+) para todos los SDK.
- Se ha eliminado la compatibilidad con Ubuntu 18.04 LTS debido a su reciente fin de vida útil.
- Se ha eliminado el soporte para Amazon Linux debido a su reciente fin de vida útil.

La CLI de CloudHSM

- Se agregaron los siguientes comandos:
  - signo criptográfico
  - verificación criptográfica
  - pem de importación de claves
  - desempaquetar llaves
  - envoltorio para llaves
- key generate-fileahora admite la exportación de claves públicas.

Motor dinámico de OpenSSL

- El motor dinámico de AWS CloudHSM OpenSSL ahora es compatible con las plataformas que vienen instaladas con una versión 3.x de la biblioteca OpenSSL. Esto incluye Amazon Linux 2023, RHEL 9 (9.2+) y Ubuntu 22.04.

JCE

- Se agregó soporte para JDK 17 y JDK 21.
- Se agregó soporte para las claves AES que se utilizarán en las operaciones de HMAC.
- Se agregó el nuevo atributo `ID` clave.
- Se introdujo una nueva `DataExceptionCause` variante para el agotamiento de las teclas:`DataExceptionCause.KEY_EXHAUSTED`.

Mejoras y correcciones de errores:

- Se ha aumentado la longitud máxima del `label` atributo de 126 a 127 caracteres.
- Se ha corregido un error que impedía abrir las llaves EC con este RsaOaep mecanismo.
- Se ha resuelto un problema conocido relacionado con la operación GetKey en el proveedor de JCE. Consulte Problema: pérdida de memoria del SDK 5 del cliente al operar con GetKey para obtener más información.
- Se ha mejorado el registro de todos los SDK para las claves Triple DES que han alcanzado su límite máximo de bloques de cifrado, según la norma FIPS 140-2.

- Se han añadido problemas conocidos para el motor dinámico de OpenSSL. Para obtener más información, consulte Problemas conocidos de OpenSSL Dynamic Engine.

Versión 5.10.0

Amazon Linux

Descargue la versión 5.10.0 del software para Amazon Linux en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum d63adf3e96c19c2d894b2defcbadd916dbb0398993050b1358bd93a36aa5acab)
- Motor dinámico de OpenSSL (SHA256 checksum 4daa3e591ffd5f7ce8ef3759c41deaa38867f5e5d21f15927aea83afb1678ac5)
- Proveedor de JCE (SHA256 checksum 6c1ac94d3080f1c609d9dafbcb14480911beef3a488c4ed6f2b11b377da9b477)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum c12617fcd7990ba53e96f477979b410e3a5f17842ca7a912861b8b820809b5b5)

Amazon Linux 2

Descargue la versión 5.10.0 del software para Amazon Linux 2 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum fc47e705e57a0bfd433f7b46c9477a70df5c442a8ad9c2969bcef38e328e4933)
- Motor dinámico de OpenSSL (SHA256 checksum 0aca262df6780995c9b884fcb8765bbd64acaf21b2286ec4d05a9a90edb3d4cb)
- Proveedor de JCE (SHA256 checksum b5be7f73c4bcffc5da6f89f324e6b3db5b091610464c8bd38dbddfff0484b2c2)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum e8cf09966890b88a61e695dc034874a445093300359d5d6a86b5a546803920bb)

Descargue la versión 5.10.0 del software para Amazon Linux 2 en arquitectura ARM64:

- Biblioteca PKCS #11 (SHA256 checksum 5d8dfd835f1ed5a7f5a4fcc8ecf81cfa29883aca7e2985de69b5db723ab663db)
- Motor dinámico de OpenSSL (SHA256 checksum 91fb8efe2646bf0dbd9087554baa09554714e9d56e9bfd5c0dc3023a9f485574)

- Proveedor de JCE (SHA256 checksum 99f6e55c37fdf00085a816d46835aeff54470797b3b71f4d28a70dc79c9caf44)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum 4a88ba9b4cf0dd5573f3dd88ab9dc257e4c486069cb529c5d554979ee2dd83af)

CentOS 7 (7.8+)

Descargue la versión 5.10.0 del software para CentOS 7 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum fc47e705e57a0bfd433f7b46c9477a70df5c442a8ad9c2969bcef38e328e4933)
- Motor dinámico de OpenSSL (SHA256 checksum 0aca262df6780995c9b884fcb8765bbd64acaf21b2286ec4d05a9a90edb3d4cb)
- Proveedor de JCE (SHA256 checksum b5be7f73c4bcffc5da6f89f324e6b3db5b091610464c8bd38dbddfff0484b2c2)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum e8cf09966890b88a61e695dc034874a445093300359d5d6a86b5a546803920bb)

RHEL 7 (7.8+)

Descargue la versión 5.10.0 del software para RHEL 7 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum fc47e705e57a0bfd433f7b46c9477a70df5c442a8ad9c2969bcef38e328e4933)
- Motor dinámico de OpenSSL (SHA256 checksum 0aca262df6780995c9b884fcb8765bbd64acaf21b2286ec4d05a9a90edb3d4cb)
- Proveedor de JCE (SHA256 checksum b5be7f73c4bcffc5da6f89f324e6b3db5b091610464c8bd38dbddfff0484b2c2)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum e8cf09966890b88a61e695dc034874a445093300359d5d6a86b5a546803920bb)

RHEL 8 (8.3+)

Descargue la versión 5.10.0 del software para RHEL 8 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 96afb7042a148ddc7a60ab6235b49e176d0460d1c2957bd76ca3d8406ac1cb03)

- Motor dinámico de OpenSSL (SHA256 checksum 2caad2bffea8aef73c91ad422d09772ef830fe7f80a7be19020e6a107eadfbe8)
- Proveedor de JCE (SHA256 checksum 3543551f08fbe3900821ea2d4ea148b4e86e2334bc94d7ffef6f3b831457cd71)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum 812eccaadfc490f13bcd0b0a835ef58f3a3d4344ad7e0a237de476dd24509525)

Ubuntu 18.04 LTS

Descargue la versión 5.10.0 del software para Ubuntu 18.04 LTS en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum be4c61766b8b46e1f6c14c3dcf90aaab9f38240fcd9c68b4009704276c5f6f4a)
- Motor dinámico de OpenSSL (SHA256 checksum 64bd8af827b6dc3786e8ad28858cbc4ef6a0fd42164a0945f427eddcf5f02858)
- Proveedor de JCE (SHA256 checksum 9fcbdf08e93641468588b608173f26f18781bbc029ed95b2e086da29a968cc00)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum 13808bddddb7eedeb2b8486d23a9976c7fa8d9220149a6b9400626bcaff3b513)

> **Note**
>
> Debido al reciente final de la vida útil de Ubuntu 18.04 LTS, ya no AWS CloudHSM será compatible con esta plataforma en la próxima versión.

Ubuntu 20.04 LTS

Descargue la versión 5.10.0 del software para Ubuntu 20.04 LTS en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 99ae96504580ff85ed4958a582903a847f666bdaafafbe887a5a76db58f24500)
- Motor dinámico de OpenSSL (SHA256 checksum 13e3f6fe086acf9617b163f66e3941f973daa583fb9322d16c396aa29fc3611d)
- Proveedor de JCE (SHA256 checksum 44562cebd9af1aa965840cd9bcb237e518d24c715b3c8bca1405c9c1871835e2)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)

- CLI de CloudHSM (SHA256 checksum ab71b4ec531c5e6d05c91539c7edc1c07e6c748052ebf6200f148cb6812538c5)

Ubuntu 22.04 LTS

Descargue la versión 5.10.0 del software para Ubuntu 22.04 LTS en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum ee331a44fbe4936ec98a3ae55d58e67ed38e8bbff0a4f4ce8b1bd8239b75877b)
- La compatibilidad con el motor dinámico de OpenSSL aún no está disponible para esta plataforma.
- Proveedor de JCE (SHA256 checksum 9e44d14dd33624f6fe36711633013e47e4a93f4d4635e08900546113ded56e3d)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum 2df361546848cd3f8965b1007dca42a0c959eb10d9e3f4995e8e1c852406751d)

Windows Server 2016

Descargue la versión 5.10.0 del software para Windows Server 2016 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 7aae9bfd99a6dd0f4d376c227c206c01847f83a9efd774d1063d76cc6fdaa89f)
- Proveedor de JCE (SHA256 checksum 1c58fd651e51be2ba59051a87aceca0452990b29837b8a7efabcd510ccbf8c1f)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum f745a2236c9eb9f6f128313eddc35795bd5e47fdf67332bedeb2554201b61a24)

Windows Server 2019

Descargue la versión 5.10.0 del software para Windows Server 2019 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 7aae9bfd99a6dd0f4d376c227c206c01847f83a9efd774d1063d76cc6fdaa89f)
- Proveedor de JCE (SHA256 checksum 1c58fd651e51be2ba59051a87aceca0452990b29837b8a7efabcd510ccbf8c1f)
  - Javadocs para AWS CloudHSM (SHA256 checksum dcbb870c6bd58c6770ba7a2b616c6103a5efb3bdeab831ce8f9c82cc09a9870f)
- CLI de CloudHSM (SHA256 checksum f745a2236c9eb9f6f128313eddc35795bd5e47fdf67332bedeb2554201b61a24)

SDK de cliente 5.10.0 mejora la estabilidad e incluye correcciones de errores para todos los SDK.

La CLI de CloudHSM

- Se han agregado nuevos comandos que permiten a los clientes administrar las claves mediante la CLI de CloudHSM, con funciones como:
  - Creación de claves simétricas y pares de claves asimétricas
  - Cómo compartir y dejar de compartir claves
  - Cómo enumerar y filtrar claves mediante atributos de clave
  - Mostrar atributos de clave
  - Cómo generar archivos de referencia de clave
  - Eliminación de claves
- Se ha mejorado el registro de errores.
- Se ha agregado compatibilidad con comandos Unicode multilínea en modo interactivo.

Mejoras y correcciones de errores:

- Se ha mejorado el rendimiento a la hora de importar, desencapsular, derivar y crear claves de sesión para todos los SDK.
- Se ha corregido un error en el proveedor de JCE que impedía eliminar los archivos temporales al salir.
- Se ha corregido un error que provocaba un error de conexión bajo determinadas condiciones tras sustituir los HSM del clúster.
- Se ha modificado el formato de salida `getVersion` de JCE para gestionar grandes números de versión menor e incluir el número de parche.

Compatibilidad con plataformas

- Se ha agregado compatibilidad para Ubuntu 22.04 con JCE, PKCS #11 y la CLI de CloudHSM (la compatibilidad con el motor dinámico de OpenSSL aún no está disponible).

Versión 5.9.0

Amazon Linux

Descargue la versión 5.9.0 del software para Amazon Linux en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 4f368be41f006b751ac41b14e1435c27841f60bbde0f032ec02a359fea637dcf)
- Motor dinámico de OpenSSL (SHA256 checksum 81af0d34683825cd6ff844ccacf9c8f4842a4ba76e3875a89121d09a286b4490)
- Proveedor de JCE (SHA256 checksum e8e5bc09d8e0b3cb24f30ab420fe08902a19073012335ac94382ec55fcc45abd)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum 17284144b45043204ce012fe8b62b1973f10068950abedbd9c2c6172ed0979c6)

Amazon Linux 2

Descargue la versión 5.9.0 del software para Amazon Linux 2 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum e5affca37abc4ff76369237649830feb32fccd3fa05199cc2021230137093c56)
- Motor dinámico de OpenSSL (SHA256 checksum 848a2e31550bbc2b0223468877baa2a8cda3131ef8537856b31db226d55c4170)
- Proveedor de JCE (SHA256 checksum 884f483ef3e9c7def92e3ff01b226e5cbf276d96dcb2f6f56009516f19d41dc0)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum 2e62d5a27cff46d9fb47d656afeccd9dbfb5413bfd2267dd3c8fb7960fef7f26)

Descargue la versión 5.9.0 del software para Amazon Linux 2 en arquitectura ARM64:

- Biblioteca PKCS #11 (SHA256 checksum 4337dca5a08c5194b1118fa197bb4a4f7988df4e1b961e6f2e367295ba99d61d)
- Motor dinámico de OpenSSL (SHA256 checksum 4f08689934e877662a7ce64554fb04eb4b2c213b936018609ff187d100e34a85)
- Proveedor de JCE (SHA256 checksum b337b80271a2d308949d5911971fe6ad35df4e34876a481fcac347f1d897fe39)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum a4d466e6b5f74dcd283ba32c9dd87441941d5e5a05936b7c2b4cc7ef85eb1071)

CentOS 7 (7.8+)

Descargue la versión 5.9.0 del software para CentOS 7 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum e5affca37abc4ff76369237649830feb32fccd3fa05199cc2021230137093c56)

- Motor dinámico de OpenSSL (SHA256 checksum 848a2e31550bbc2b0223468877baa2a8cda3131ef8537856b31db226d55c4170)
- Proveedor de JCE (SHA256 checksum 884f483ef3e9c7def92e3ff01b226e5cbf276d96dcb2f6f56009516f19d41dc0)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum 2e62d5a27cff46d9fb47d656afeccd9dbfb5413bfd2267dd3c8fb7960fef7f26)

RHEL 7 (7.8+)

Descargue la versión 5.9.0 del software para RHEL 7 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum e5affca37abc4ff76369237649830feb32fccd3fa05199cc2021230137093c56)
- Motor dinámico de OpenSSL (SHA256 checksum 848a2e31550bbc2b0223468877baa2a8cda3131ef8537856b31db226d55c4170)
- Proveedor de JCE (SHA256 checksum 884f483ef3e9c7def92e3ff01b226e5cbf276d96dcb2f6f56009516f19d41dc0)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum 2e62d5a27cff46d9fb47d656afeccd9dbfb5413bfd2267dd3c8fb7960fef7f26)

RHEL 8 (8.3+)

Descargue la versión 5.9.0 del software para RHEL 8 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 081887f6ea1d9df9d1e409b2b5bde83e965c42229acbeb1f950c8fe478361edc)
- Motor dinámico de OpenSSL (SHA256 checksum 6b0500a42fd57c39f076f14e5079f80145b6ebd2c441395761eb04600c07bda5)
- Proveedor de JCE (SHA256 checksum 2bc7ac26b259af92a65fbd5a30d5eb2a92ce0e70efe41feb53bf82f168aa90bb)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum 79ecbe9b4c5316ccf447d8c59b76b5ac2cc854bd79cd50c1f29197aa8cb080db)

Ubuntu 18.04 LTS

Descargue la versión 5.9.0 del software para Ubuntu 18.04 LTS en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum bc6d2227edd7b5a83fed32741fbacbb1756d5df89ebb3435d96f0609a180db65)
- Motor dinámico de OpenSSL (SHA256 checksum 2d6a26434fa6faf337f1dfb42de033220fa405a82d4540e279639a03b3ee6e9d)
- Proveedor de JCE (SHA256 checksum e12aef122f490e9026452ce31c25625b1accb9a5866b3d470488f10f047f1873)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum f0bcabe594db3e8ff86cc0f65c2a10858d34452eb6b9fc33d7aac05c0f5f4f30)

Ubuntu 20.04 LTS

Descargue la versión 5.9.0 del software para Ubuntu 20.04 LTS en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum 15dde8182f432de9e7d369b05e384e1f2d80dcca85db3b16ecc26cdef1a34bb9)
- Motor dinámico de OpenSSL (SHA256 checksum c8ba94a999038af87d4905b7c1feb4cc87e20d1776a32ef6f6d11ee000b5a896)
- Proveedor de JCE (SHA256 checksum de33cd3e8130a06d9da5207079533aac8276a1319ac435a3737b4f65bd8fb972)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum cfa31535ad9a99a5113496c06fbace38e9593491aca9bb031a18b51075973e68)

Windows Server 2016

Descargue la versión 5.9.0 del software para Windows Server 2016 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum ab5380805b0e17dd89dbbefd3fbda8b54da3c140f82e9f3d021850c31837bbe3)
- Proveedor de JCE (SHA256 checksum f0941d7a20193818133de8a742d3b848ea19abaf25f5a71ac65949ce5a37c533)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum 131530ffe5caff963d483f440d06dcfb41dc11b0f8d78f1dd07bb07f76aeb6d2)

Windows Server 2019

Descargue la versión 5.9.0 del software para Windows Server 2019 en arquitectura x86_64:

- Biblioteca PKCS #11 (SHA256 checksum ab5380805b0e17dd89dbbefd3fbda8b54da3c140f82e9f3d021850c31837bbe3)

- Proveedor de JCE (SHA256 checksum f0941d7a20193818133de8a742d3b848ea19abaf25f5a71ac65949ce5a37c533)
  - Javadocs para AWS CloudHSM (SHA256 checksum 6343427177180c8f61eec0341e827fbba29420ed2033c0e4b4803d49a3df7763)
- CLI de CloudHSM (SHA256 checksum 131530ffe5caff963d483f440d06dcfb41dc11b0f8d78f1dd07bb07f76aeb6d2)

SDK de cliente 5.9.0 mejora la estabilidad e incluye correcciones de errores para todos los SDK. Se han optimizado todos los SDK para informar a las aplicaciones de los fallos de funcionamiento inmediatamente cuando se determine que un HSM no está disponible. Esta versión incluye mejoras de rendimiento para JCE.

Proveedor de JCE

- Rendimiento mejorado
- Se ha corregido un problema conocido con el agotamiento del grupo de sesiones.

Versión 3.4.4

Para poder actualizar SDK 3 de cliente en plataformas Linux, debe utilizar un comando por lotes que actualice daemon de cliente y todas las bibliotecas al mismo tiempo. Para obtener más información, consulte este artículo sobre Actualización de SDK 3 de cliente.

Para descargar el software, elija la pestaña correspondiente al sistema operativo que prefiera y, a continuación, elija el enlace a cada paquete de software.

Amazon Linux

Descargue la versión 3.4.4 del software para Amazon Linux:

- AWS CloudHSM Cliente (SHA256 checksum 900de424d70f41e661aa636f256a6a79cc43bea6b0fe6eb95c2aaa63e5289505)
- Biblioteca PKCS #11 (SHA256 checksum a3f93f084d59fee5d7c859292bc02cb7e7f15fb06e971171ebf9b52bbd229c30)
- Motor dinámico de OpenSSL (SHA256 checksum 8db07b9843d49016b0b6fec46d39881d94e426fcaae1cee2747be14af9313bb0)
- Proveedor de JCE (SHA256 checksum 360617c55bf4caa8e6e78ede079ca68cf9ef11473e7918154c22ba908a219843)
- AWS CloudHSM Utilidad de administración (SHA256 checksum c9961ffe38921131bd6f3702e10d73588e68b8ab10fbb241723e676f4fa8c4fa)

Amazon Linux 2

Descargue la versión 3.4.4 del software para Amazon Linux 2:

- AWS CloudHSM Cliente (SHA256 checksum 7d61d835ae38c6ce121d102b516527f342a76ac31733768097d5cab8bc482610)
- Biblioteca PKCS #11 (SHA256 checksum 2099f324ff625e1a46d96c1d5084263ca1d650424d7465ead43fe767d6687f36)
- Motor dinámico de OpenSSL (SHA256 checksum 6d8e81ad1208652904fe4b6abc4f174e866303f2302a6551c3fbef617337e663)
- Proveedor de JCE (SHA256 checksum 70e3cdce143c45a76e155ffb5969841e0153e011f59eb9f2c6e6be0707030abf)
- AWS CloudHSM Utilidad de administración (SHA256 checksum 5a702fe5e50dc6055daa723df71a0874317c9ff5844eea30104587a61097ecf4)

CentOS 6

AWS CloudHSM no es compatible con CentOS 6 con la versión 3.4.4 del SDK de cliente.

Use the section called “Versión 3.2.1” para CentOS 6 o elija una plataforma compatible.

CentOS 7 (7.8+)

Descargue la versión 3.4.4 del software para CentOS 7:

- AWS CloudHSM Cliente (SHA256 checksum 7d61d835ae38c6ce121d102b516527f342a76ac31733768097d5cab8bc482610)
- Biblioteca PKCS #11 (SHA256 checksum 2099f324ff625e1a46d96c1d5084263ca1d650424d7465ead43fe767d6687f36)
- Motor dinámico de OpenSSL (SHA256 checksum 6d8e81ad1208652904fe4b6abc4f174e866303f2302a6551c3fbef617337e663)
- Proveedor de JCE (SHA256 checksum 70e3cdce143c45a76e155ffb5969841e0153e011f59eb9f2c6e6be0707030abf)
- AWS CloudHSM Utilidad de administración (SHA256 checksum 5a702fe5e50dc6055daa723df71a0874317c9ff5844eea30104587a61097ecf4)

CentOS 8

Descargue la versión 3.4.4 del software para CentOS 8:

- AWS CloudHSM Cliente (SHA256 checksum 81639c9ec83e501709c4117ba9d98b23dea7838a206ed244c9c6cc0d65130f8c)

- Biblioteca PKCS #11 (SHA256 checksum 9a15daa87b8616cf03a6bf6b375f53451ef448dbc54bf2c27fbc2be7823fc633)
- Proveedor de JCE (SHA256 checksum 2b1c4208992903cf7bcc669c1392c59a64fbfc82e010c626ffa58d0cb8e9126b)
- AWS CloudHSM Utilidad de administración (SHA256 checksum 3adbecc802e0854c23aa4b8d80540d1748903c8dba93b6c8042fb7885051c360)

> **Note**
>
> Debido al reciente final de la vida útil de CentOS 8, ya no podremos dar soporte a esta plataforma en la próxima versión.

RHEL 6

AWS CloudHSM no es compatible con RedHat Enterprise Linux 6 con la versión 3.4.4 del SDK de cliente.

the section called “Versión 3.2.1”Úselo para RedHat Enterprise Linux 6 o elija una plataforma compatible.

RHEL 7 (7.8+)

Descargue la versión 3.4.4 del software para RedHat Enterprise Linux 7:

- AWS CloudHSM Cliente (SHA256 checksum 7d61d835ae38c6ce121d102b516527f342a76ac31733768097d5cab8bc482610)
- Biblioteca PKCS #11 (SHA256 checksum 2099f324ff625e1a46d96c1d5084263ca1d650424d7465ead43fe767d6687f36)
- Motor dinámico de OpenSSL (SHA256 checksum 6d8e81ad1208652904fe4b6abc4f174e866303f2302a6551c3fbef617337e663)
- Proveedor de JCE (SHA256 checksum 70e3cdce143c45a76e155ffb5969841e0153e011f59eb9f2c6e6be0707030abf)
- AWS CloudHSM Utilidad de administración (SHA256 checksum 5a702fe5e50dc6055daa723df71a0874317c9ff5844eea30104587a61097ecf4)

RHEL 8 (8.3+)

Descargue la versión 3.4.4 del software para RedHat Enterprise Linux 8:

- AWS CloudHSM Cliente (SHA256 checksum 81639c9ec83e501709c4117ba9d98b23dea7838a206ed244c9c6cc0d65130f8c)

- Biblioteca PKCS #11 (SHA256 checksum 9a15daa87b8616cf03a6bf6b375f53451ef448dbc54bf2c27fbc2be7823fc633)
- Proveedor de JCE (SHA256 checksum 2b1c4208992903cf7bcc669c1392c59a64fbfc82e010c626ffa58d0cb8e9126b)
- AWS CloudHSM Utilidad de administración (SHA256 checksum 3adbecc802e0854c23aa4b8d80540d1748903c8dba93b6c8042fb7885051c360)

Ubuntu 16.04 LTS

Descargue la versión 3.4.4 del software para Ubuntu 16.04 LTS:

- AWS CloudHSM Cliente (SHA256 checksum 317c92c2e0b5d60afab1beb947f053d13ddaacb994cccc2c2b898e997ece29b9)
- Biblioteca PKCS #11 (SHA256 checksum 91451c420c51488a022569fd32f052a3b988a2883ea4c2ac952acb61a2fea37c)
- Motor dinámico de OpenSSL (SHA256 checksum 4098771ad0e38df9bf14d50520ca49b9395f819f0387e2bc3b0e61abb5888e66)
- Proveedor de JCE (SHA256 checksum e136ff183271c2f9590a9fccb8261a7eb809506686b070e3854df1b8686c6641)
- AWS CloudHSM Utilidad de administración (SHA256 checksum cbf24a4032f393a913a9898b1b27036392104e8e05d911cab84049b2bcca2541)

> **Note**
>
> Debido a la inminente EOL de Ubuntu 16.04, dejaremos de ofrecer soporte para esta plataforma en la próxima versión.

Ubuntu 18.04 LTS

Descargue la versión 3.4.4 del software para Ubuntu 18.04 LTS:

- AWS CloudHSM Cliente (SHA256 checksum cf57d5e0e95efbf032aac8887aebd59ac8cc80e97c69e7c39fdad40873374fe8)
- Biblioteca PKCS #11 (SHA256 checksum 428f8bdad7925db5401112f707942ee8f3ca554f4ab53fa92237996e69144d2f)
- Proveedor de JCE (SHA256 checksum 1ff17b8f7688e84f7f0bfc96383564dca598a1cab2f2c52c888d0361682f2b9e)
- AWS CloudHSM Utilidad de administración (SHA256 checksum afe253046146ed6177c520b681efc680dac1048c4a95b3d8ad0f305e79bbe93e)

Windows Server

AWS CloudHSM admite las versiones de 64 bits de Windows Server 2012, Windows Server 2012 R2, Windows Server 2016 y Windows Server 2019. El software cliente AWS CloudHSM 3.4.4 para Windows Server incluye los proveedores de GNC y KSP necesarios. Para obtener más información, consulte Instalación y configuración del AWS CloudHSM cliente (Windows). Descargue la versión más reciente (3.4.4) del software para Windows Server:

- AWS CloudHSM para Windows Server (SHA256 checksum d51a7db588e9121d8f0b0351606bd986e1c4de6547f2c8235200dc8a5ffbe53e)
- AWS CloudHSM Utilidad de administración (SHA256 checksum 0c12d7da9086735cdf189535937a8e036163009c5018dcaf2ee9cdbb6bd4c06f)

La versión 3.4.4 agrega actualizaciones al proveedor de JCE.

AWS CloudHSM Software de cliente

- Versión actualizada por coherencia.

Biblioteca PKCS #11

- Versión actualizada por coherencia.

Motor dinámico de OpenSSL

- Versión actualizada por coherencia.

Proveedor de JCE

- Actualice la versión de log4j a la 2.17.1.

Proveedores de KSP y CNG para Windows

- Versión actualizada por coherencia.

Versiones obsoletas

Las versiones 5.8.0 y anteriores están en desuso. No recomendamos usar versiones obsoletas en cargas de trabajo de producción. No proporcionamos actualizaciones retrocompatibles con versiones anteriores para las versiones obsoletas, ni las alojamos para su descarga. Si su producción se ve afectada por el uso de versiones obsoletas, deberá actualizarlas para corregir el software.

Versiones obsoletas del Client SDK 5

En esta sección se enumeran las versiones obsoletas de Client SDK 5.

Versión 5.8.0

La versión 5.8.0 incluye autenticación de cuórum para la CLI de CloudHSM, descarga de SSL/TLS con JSSE, compatibilidad con varias ranuras para PKCS #11, compatibilidad con varios clústeres y varios usuarios para JCE, extracción de claves con JCE, compatibilidad con KeyFactory para JCE y nuevas configuraciones de reintento para códigos de retorno no terminales, así como mejoras de estabilidad y correcciones de errores para todos los SDK.

Biblioteca PKCS #11

- Se ha agregado soporte para configuración de múltiples ranuras.

Proveedor de JCE

- Se ha agregado extracción de claves basada en configuración.
- Se ha agregado compatibilidad con configuraciones de múltiples clústeres y múltiples usuarios.
- Se ha agregado soporte para la descarga de SSL y TLS con JSSE.
- Se agregó el soporte de desempaquetado para AES/CBC/. NoPadding
- Se agregaron nuevos tipos de fábricas clave: y. SecretKeyFactory KeyFactory

La CLI de CloudHSM

- Se ha agregado compatibilidad con autenticación de cuórum.

Versión 5.7.0

La versión 5.7.0 presenta la CLI de CloudHSM e incluye un nuevo algoritmo de código de autenticación de mensajes basado en cifrado (CMAC). En esta versión se ha agregado la arquitectura ARM a Amazon Linux 2. Los Javadocs del proveedor de JCE ya están disponibles para AWS CloudHSM.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.
- Ahora es compatible con la arquitectura ARM en Amazon Linux 2.
- Algoritmos
  - CKM_AES_CMAC (firmar y verificar)

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.
- Ahora es compatible con la arquitectura ARM en Amazon Linux 2.

Proveedor de JCE

- Mejoras de estabilidad y correcciones de errores.
- Algoritmos
  - AESCMAC

Versión 5.6.0

La versión 5.6.0 incluye soporte de mecanismo para la biblioteca PKCS #11 y el proveedor de JCE. Además, la versión 5.6 es compatible con Ubuntu 20.04.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.
- Mecanismos
  - CKM_RSA_X_509, para modos de cifrado, descifrado, firma y verificación

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Proveedor de JCE

- Mejoras de estabilidad y correcciones de errores.
- Cifrados
  - RSA/ECB/, para los modos de cifrado NoPadding y descifrado

Claves compatibles

- EC con curvas secp224r1 y secp521r1

Compatibilidad con plataformas

- Incorpora la compatibilidad con Ubuntu 20.04.

Versión 5.5.0

La versión 5.5.0 añade soporte para la integración de OpenJDK 11, Keytool y Jarsigner, así como mecanismos adicionales para el proveedor de JCE. Resuelve un problema conocido relacionado con una KeyGenerator clase que interpreta incorrectamente el parámetro de tamaño de la clave como número de bytes en lugar de bits.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Proveedor de JCE

- Compatibilidad con las utilidades Keytool y Jarsigner
- Compatibilidad con OpenJDK 11 en todas las plataformas

- Cifrados
  - Modo AES/CBC/ NoPadding Cifrar y descifrar
  - Modos de cifrado y descifrado de AES/ECB/PKCS5Padding
  - Modo AES/CTR/ NoPadding Cifrar y descifrar
  - Modo AES/GCM/ Envolver y desempaquetar NoPadding
  - Modos de cifrado y descifrado de DESede/ECB/PKCS5Padding
  - Modo Desede/CBC/ Cifrar y descifrar NoPadding
  - Modo NoPadding AESWRAP/ECB/ Wrap y Unwrap
  - AESWrap/ECB/PKCS5Padding Modos de encapsulado y desencapsulado
  - Modo ZeroPadding AESWrap/ECB/Wrap y Unwrap
  - RSA/ECB/PKCS1Padding Modos de encapsulado y desencapsulado
  - RSA/ECB/OAEPPadding Modos de encapsulado y desencapsulado
  - RSA/ECB/OAEPWithSHA-1ANDMGF1Padding Modos de encapsulado y desencapsulado
  - RSA/ECB/OAEPWithSHA-224ANDMGF1Padding Modos de encapsulado y desencapsulado
  - RSA/ECB/OAEPWithSHA-256ANDMGF1Padding Modos de encapsulado y desencapsulado
  - RSA/ECB/OAEPWithSHA-384ANDMGF1Padding Modos de encapsulado y desencapsulado
  - RSA/ECB/OAEPWithSHA-512ANDMGF1Padding Modos de encapsulado y desencapsulado
  - RSAAESWrap/ECB/OAEPPadding Modos de encapsulado y desencapsulado
  - RSAAESWrap/ECB/OAEPWithSHA-1ANDMGF1Padding Modos de encapsulado y desencapsulado
  - RSAAESWrap/ECB/OAEPWithSHA-224ANDMGF1Padding Modos de encapsulado y desencapsulado
  - RSAAESWrap/ECB/OAEPWithSHA-256ANDMGF1Padding Modos de encapsulado y desencapsulado
  - RSAAESWrap/ECB/OAEPWithSHA-384ANDMGF1Padding Modos de encapsulado y desencapsulado
  - RSAAESWrap/ECB/OAEPWithSHA-512ANDMGF1Padding Modos de encapsulado y desencapsulado
- KeyFactory y SecretKeyFactory
  - RSA: claves RSA de 2048 a 4096 bits, en incrementos de 256 bits
  - AES: claves AES de 128, 192 y 256 bits

  - Pares de claves EC para curvas de NIST secp256r1 (P-256), secp384r1 (P-384) y secp256k1
  - DESede (3DES)
  - GenericSecret
  - HMAC: soporte de hash para SHA1, SHA224, SHA256, SHA384, SHA512.
- Firmar o verificar
  - RSASSA-PSS
  - SHA1withRSA/PSS
  - SHA224withRSA/PSS
  - SHA256withRSA/PSS
  - SHA384withRSA/PSS
  - SHA512withRSA/PSS
  - SHA1withRSAandMGF1
  - SHA224withRSAandMGF1
  - SHA256withRSAandMGF1
  - SHA384withRSAandMGF1
  - SHA512withRSAandMGF1

Versión 5.4.2

SDK de cliente 5.4.2 mejora la estabilidad e incluye correcciones de errores para todos los SDK. Esta es también la última versión para la plataforma CentOS 8. Para obtener más información, consulte el sitio web de CentOS.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Proveedor de JCE

- Mejoras de estabilidad y correcciones de errores.

versión 5.4.1

La versión 5.4.1 resuelve un problema conocido con la biblioteca PKCS #11. Esta es también la última versión para la plataforma CentOS 8. Para obtener más información, consulte el sitio web de CentOS.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Proveedor de JCE

- Mejoras de estabilidad y correcciones de errores.

Versión 5.4.0

La versión 5.4.0 añade soporte inicial para el proveedor de JCE en todas las plataformas. El proveedor de JCE es compatible con OpenJDK 8.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Proveedor de JCE

- Tipos de clave
  - RSA: claves RSA de 2048 a 4096 bits, en incrementos de 256 bits.
  - AES: claves AES de 128, 192 y 256 bits.
  - Pares de claves EC para curvas de NIST secp256r1 (P-256), secp384r1 (P-384) y secp256k1
  - DESede (3DES)

  - Soporte de hash para SHA1, SHA224, SHA256, SHA384, SHA512.
- Cifrados (solo cifrado y descifrado)
  - AES/GCM/ NoPadding
  - AES/ECB/ NoPadding
  - AES/CBC/PKCS5Padding
  - Desede/ECB/ NoPadding
  - DESede/CBC/PKCS5Padding
  - AES/CTR/ NoPadding
  - RSA/ECB/PKCS1Padding
  - RSA/ECB/OAEPPadding
  - RSA/ECB/OAEPWithSHA-1ANDMGF1Padding
  - RSA/ECB/OAEPWithSHA-224ANDMGF1Padding
  - RSA/ECB/OAEPWithSHA-256ANDMGF1Padding
  - RSA/ECB/OAEPWithSHA-384ANDMGF1Padding
  - RSA/ECB/OAEPWithSHA-512ANDMGF1Padding
- Digests
  - SHA-1
  - SHA-224
  - SHA-256
  - SHA-384
  - SHA-512
- Firmar o verificar
  - NONEwithRSA
  - SHA1withRSA
  - SHA224withRSA
  - SHA256withRSA
  - SHA384withRSA
  - SHA512withRSA
  - NONEwithECDSA
  - SHA1withECDSA

  - SHA224withECDSA
  - SHA256withECDSA
  - SHA384withECDSA
  - SHA512withECDSA
- Integración con Java KeyStore

Versión 5.3.0

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.

Motor dinámico de OpenSSL

- Compatibilidad añadida con la firma o verificación de ECDSA con las curvas P-256, P-384 y secp256k1.
- Añada soporte para las plataformas: Amazon Linux, Amazon Linux 2, Centos 7.8+, RHEL 7 (7.8+).
- Se ha agregado compatibilidad con OpenSSL versión 1.0.2.
- Mejoras de estabilidad y correcciones de errores.

Proveedor de JCE

- Tipos de clave
  - RSA: claves RSA de 2048 a 4096 bits, en incrementos de 256 bits.
  - AES: claves AES de 128, 192 y 256 bits.
  - Pares de claves EC para curvas de NIST secp256r1 (P-256), secp384r1 (P-384) y secp256k1
  - DESede (3DES)
  - Soporte de hash para SHA1, SHA224, SHA256, SHA384, SHA512.
- Cifrados (solo cifrado y descifrado)
  - AES/GCM/ NoPadding
  - AES/ECB/ NoPadding
  - AES/CBC/PKCS5Padding
  - Desede/ECB/ NoPadding

  - DESede/CBC/PKCS5Padding
  - AES/CTR/ NoPadding
  - RSA/ECB/PKCS1Padding
  - RSA/ECB/OAEPPadding
  - RSA/ECB/OAEPWithSHA-1ANDMGF1Padding
  - RSA/ECB/OAEPWithSHA-224ANDMGF1Padding
  - RSA/ECB/OAEPWithSHA-256ANDMGF1Padding
  - RSA/ECB/OAEPWithSHA-384ANDMGF1Padding
  - RSA/ECB/OAEPWithSHA-512ANDMGF1Padding
- Digests
  - SHA-1
  - SHA-224
  - SHA-256
  - SHA-384
  - SHA-512
- Firmar o verificar
  - NONEwithRSA
  - SHA1withRSA
  - SHA224withRSA
  - SHA256withRSA
  - SHA384withRSA
  - SHA512withRSA
  - NONEwithECDSA
  - SHA1withECDSA
  - SHA224withECDSA
  - SHA256withECDSA
  - SHA384withECDSA
  - SHA512withECDSA
- Integración con Java KeyStore

Versión 5.2.1

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Versión 5.2.0

La versión 5.2.0 agrega compatibilidad con tipos de claves y mecanismos adicionales a la biblioteca PKCS #11.

Biblioteca PKCS #11

Tipos de clave

- ECDSA: curvas P-224, P-256, P-384, P-521 y secp256k1
- Triple DES (3DES)

Mecanismos

- CKM_EC_KEY_PAIR_GEN
- CKM_DES3_KEY_GEN
- CKM_DES3_CBC
- CKM_DES3_CBC_PAD
- CKM_DES3_ECB
- CKM_ECDSA
- CKM_ECDSA_SHA1
- CKM_ECDSA_SHA224
- CKM_ECDSA_SHA256
- CKM_ECDSA_SHA384
- CKM_ECDSA_SHA512
- CKM_RSA_PKCS para cifrado/descifrado

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Versión 5.1.0

La versión 5.1.0 agrega compatibilidad con tipos de claves y mecanismos adicionales a la biblioteca PKCS #11.

Biblioteca PKCS #11

Mecanismos

- CKM_RSA_PKCS para encapsulado y desencapsulado
- CKM_RSA_PKCS_PSS
- CKM_SHA1_RSA_PKCS_PSS
- CKM_SHA224_RSA_PKCS_PSS
- CKM_SHA256_RSA_PKCS_PSS
- CKM_SHA384_RSA_PKCS_PSS
- CKM_SHA512_RSA_PKCS_PSS
- CKM_AES_ECB
- CKM_AES_CTR
- CKM_AES_CBC
- CKM_AES_CBC_PAD
- CKM_SP800_108_COUNTER_KDF
- CKM_GENERIC_SECRET_KEY_GEN
- CKM_SHA_1_HMAC
- CKM_SHA224_HMAC
- CKM_SHA256_HMAC
- CKM_SHA384_HMAC
- CKM_SHA512_HMAC
- Solo encapsulado y desencapsulado de CKM_RSA_PKCS_OAEP
- CKM_RSA_AES_KEY_WRAP

- CKM_CLOUDHSM_AES_KEY_WRAP_NO_PAD
- CKM_CLOUDHSM_AES_KEY_WRAP_PKCS5_PAD
- CKM_CLOUDHSM_AES_KEY_WRAP_ZERO_PAD

Operaciones de API

- C_ CreateObject
- C_ DeriveKey
- C_ WrapKey
- C_ UnWrapKey

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Versión 5.0.1

La versión 5.0.1 añade soporte inicial para el motor dinámico de OpenSSL.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.

Motor dinámico de OpenSSL

- Versión inicial del motor dinámico de OpenSSL.
- Esta versión ofrece soporte introductorio para los tipos de clave y las API de OpenSSL:
  - Generación de claves RSA para claves de 2048, 3072 y 4096 bits
  - API de OpenSSL:
    - Firma de RSA usando RSA PKCS con SHA1/224/256/384/512 y RSA PSS
    - Generación de claves de RSA

  Para obtener más información, consulte Motor dinámico de OpenSSL.
- Plataformas compatibles: CentOS 8.3+, Red Hat Enterprise Linux (RHEL) 8.3+ y Ubuntu 18.04 LTS

- Requiere: OpenSSL 1.1.1

  Para obtener más información, consulte Plataformas compatibles.

- Compatibilidad con descarga de SSL/TLS en CentOS 8.3+, Red Hat Enterprise Linux (RHEL) 8.3 y Ubuntu 18.04 LTS, incluido NGINX 1.19 (para paquetes de cifrado seleccionados).

  Para obtener más información, consulte Descarga de SSL/TLS en Linux.

Versión 5.0.0

La versión 5.0.0 es la primera versión.

Biblioteca PKCS #11

- Esta es la versión inicial.

Soporte introductorio para la biblioteca PKCS #11 en la versión 5.0.0 de SDK de cliente

En esta sección se detalla la compatibilidad con los tipos de clave, los mecanismos, las operaciones de API y los atributos SDK de cliente versión 5.0.0.

Tipos de clave:

- AES: claves AES de 128, 192 y 256 bits
- RSA: claves RSA de 2048 a 4096 bits, en incrementos de 256 bits

Mecanismos

- CKM_AES_GCM
- CKM_AES_KEY_GEN
- CKM_CLOUDHSM_AES_GCM
- CKM_RSA_PKCS
- CKM_RSA_X9_31_KEY_PAIR_GEN
- CKM_SHA1
- CKM_SHA1_RSA_PKCS
- CKM_SHA224

- CKM_SHA224_RSA_PKCS
- CKM_SHA256
- CKM_SHA256_RSA_PKCS
- CKM_SHA384
- CKM_SHA384_RSA_PKCS
- CKM_SHA512
- CKM_SHA512_RSA_PKCS

Operaciones de API:

- C_ CloseAllSessions
- C_ CloseSession
- C_Decrypt
- C_ DecryptFinal
- C_ DecryptInit
- C_ DecryptUpdate
- C_ DestroyObject
- C_Digest
- C_ DigestFinal
- C_ DigestInit
- C_ DigestUpdate
- C_Encrypt
- C_ EncryptFinal
- C_ EncryptInit
- C_ EncryptUpdate
- C_Finalize
- C_ FindObjects
- C_ FindObjectsFinal
- C_ FindObjectsInit
- C_ GenerateKey

- C_ GenerateKeyPair
- C_ GenerateRandom
- C_ GetAttributeValue
- C_ GetFunctionList
- C_ GetInfo
- C_ GetMechanismInfo
- C_ GetMechanismList
- C_ GetSessionInfo
- C_ GetSlotInfo
- C_ GetSlotList
- C_ GetTokenInfo
- C_Initialize
- C_Login
- C_Logout
- C_ OpenSession
- C_Sign
- C_ SignFinal
- C_ SignInit
- C_ SignUpdate
- C_Verify
- C_ VerifyFinal
- C_ VerifyInit
- C_ VerifyUpdate

Atributos:

- GenerateKeyPair
  - Todos los atributos de clave de RSA
- GenerateKey
  - Todos los atributos de clave de AES
- GetAttributeValue

- Todos los atributos de clave de RSA
- Todos los atributos de clave de AES

Ejemplos:

- Generar claves (AES, RSA, EC)
- Mostrar atributos de clave
- Cifrado y descifrado de datos con AES-GCM
- Firmar y verificar datos con RSA

Versiones obsoletas del Client SDK 3

En esta sección se enumeran las versiones obsoletas de Client SDK 3.

Versión 3.4.3

La versión 3.4.3 agrega actualizaciones al proveedor de JCE.

AWS CloudHSM Software de cliente

- Versión actualizada por coherencia.

Biblioteca PKCS #11

- Versión actualizada por coherencia.

Motor dinámico de OpenSSL

- Versión actualizada por coherencia.

Proveedor de JCE

- Actualice la versión de log4j a la 2.17.0.

Proveedores de KSP y CNG para Windows

- Versión actualizada por coherencia.

Versión 3.4.2

La versión 3.4.2 agrega actualizaciones al proveedor de JCE.

AWS CloudHSM Software de cliente

- Versión actualizada por coherencia.

Biblioteca PKCS #11

- Versión actualizada por coherencia.

Motor dinámico de OpenSSL

- Versión actualizada por coherencia.

Proveedor de JCE

- Actualice la versión de log4j a la 2.16.0.

Proveedores de KSP y CNG para Windows

- Versión actualizada por coherencia.

Versión 3.4.1

La versión 3.4.1 agrega actualizaciones al proveedor de JCE.

AWS CloudHSM Software de cliente

- Versión actualizada por coherencia.

Biblioteca PKCS #11

- Versión actualizada por coherencia.

Motor dinámico de OpenSSL

- Versión actualizada por coherencia.

Proveedor de JCE

- Actualice la versión de log4j a la 2.15.0.

Proveedores de KSP y CNG para Windows

- Versión actualizada por coherencia.

Versión 3.4.0

La versión 3.4.0 agrega actualizaciones a todos los componentes.

AWS CloudHSM Software de cliente

- Mejoras de estabilidad y correcciones de errores.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Proveedor de JCE

- Mejoras de estabilidad y correcciones de errores.

Proveedores de KSP y CNG para Windows

- Mejoras de estabilidad y correcciones de errores.

Versión 3.3.2

La versión 3.3.2 resuelve un problema con el script client_info.

AWS CloudHSM Software de cliente

- Versión actualizada por coherencia.

Biblioteca PKCS #11

- Versión actualizada por coherencia.

Motor dinámico de OpenSSL

- Versión actualizada por coherencia.

Proveedor de JCE

- Versión actualizada por coherencia.

Proveedores de KSP y CNG para Windows

- Versión actualizada por coherencia.

Versión 3.3.1

La versión 3.3.1 agrega actualizaciones a todos los componentes.

AWS CloudHSM Software de cliente

- Mejoras de estabilidad y correcciones de errores.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Proveedor de JCE

- Mejoras de estabilidad y correcciones de errores.

Proveedores de KSP y CNG para Windows

- Mejoras de estabilidad y correcciones de errores.

Versión 3.3.0

La versión 3.3.0 añade autenticación de dos factores (2FA) y otras mejoras.

AWS CloudHSM Software de cliente

- Se ha agregado autenticación 2FA para responsables de criptografía (CO). Para obtener más información, consulte Administración de autenticación de dos factores para responsables de criptografía.
- Se ha eliminado el soporte de plataforma para RedHat Enterprise Linux 6 y Centos 6. Para obtener más información, consulte Soporte de Linux.
- Se ha agregado una versión independiente de CMU para su uso con SDK 5 de cliente o SDK 3 de cliente. Esta es la versión de CMU incluida en el daemon del cliente de la versión 3.3.0. Ahora puede descargar CMU sin descargar el daemon de cliente.

Biblioteca PKCS #11

- Mejoras de estabilidad y correcciones de errores.
- Se ha eliminado el soporte de plataforma para RedHat Enterprise Linux 6 y Centos 6. Para obtener más información, consulte Soporte de Linux.

Motor dinámico de OpenSSL

- Versión actualizada por coherencia.
- Se ha eliminado el soporte de plataforma para RedHat Enterprise Linux 6 y Centos 6. Para obtener más información, consulte Soporte de Linux.

Proveedor de JCE

- Mejoras de estabilidad y correcciones de errores.
- Se ha eliminado el soporte de plataforma para RedHat Enterprise Linux 6 y Centos 6. Para obtener más información, consulte Soporte de Linux.

Proveedores de KSP y CNG para Windows

- Versión actualizada por coherencia.

Versión 3.2.1

La versión 3.2.1 añade un análisis de conformidad entre la AWS CloudHSM implementación de la biblioteca PKCS #11 y el estándar PKCS #11, nuevas plataformas y otras mejoras.

AWS CloudHSM Software de cliente

- Compatibilidad añadida con plataformas CentOS 8, RHEL 8 y Ubuntu 18.04 LTS. Para obtener más información, consulte ???.

Biblioteca PKCS #11

- Informe de conformidad de biblioteca PKCS #11 para SDK 3.2.1 de cliente.
- Compatibilidad añadida con plataformas CentOS 8, RHEL 8 y Ubuntu 18.04 LTS. Para obtener más información, consulte ???.

Motor dinámico de OpenSSL

- No ofrece compatibilidad con CentOS 8, RHEL 8 y Ubuntu 18.04 LTS. Para obtener más información, consulte ???.

Proveedor de JCE

- Compatibilidad añadida con plataformas CentOS 8, RHEL 8 y Ubuntu 18.04 LTS. Para obtener más información, consulte ???.

Proveedores de KSP y CNG para Windows

- Mejoras de estabilidad y correcciones de errores.

Versión 3.2.0

La versión 3.2.0 añade compatibilidad con el enmascaramiento de contraseñas y otras mejoras.

AWS CloudHSM Software de cliente

- Permite ocultar la contraseña cuando se utilizan herramientas de línea de comandos. Para obtener más información, consulte loginHSM y logoutHSM (cloudhsm_mgmt_util) y loginHSM y logoutHSM (key_mgmt_util).

Biblioteca PKCS #11

- Compatibilidad añadida con el cifrado mediante hash de datos de gran tamaño en el software para algunos mecanismos de PKCS #11 que antes no eran compatibles. Para obtener más información, consulte Mecanismos admitidos.

Motor dinámico de OpenSSL

- Mejoras de estabilidad y correcciones de errores.

Proveedor de JCE

- Versión actualizada por coherencia.

Proveedores de KSP y CNG para Windows

- Mejoras de estabilidad y correcciones de errores.

Versión 3.1.2

La versión 3.1.2 agrega actualizaciones al proveedor de JCE.

AWS CloudHSM Software de cliente

- Versión actualizada por coherencia.

Biblioteca PKCS #11

- Versión actualizada por coherencia.

Motor dinámico de OpenSSL

- Versión actualizada por coherencia.

Proveedor de JCE

- Actualice la versión de log4j a la 2.13.3.

Proveedores de KSP y CNG para Windows

- Versión actualizada por coherencia.

Versión 3.1.1

AWS CloudHSM Software de cliente

- Versión actualizada por coherencia.

Biblioteca PKCS #11

- Versión actualizada por coherencia.

Motor dinámico de OpenSSL

- Versión actualizada por coherencia.

Proveedor de JCE

- Correcciones de errores y mejoras de rendimiento.

Windows (CNG, KSP)

- Versión actualizada por coherencia.

Versión 3.1.0

La versión 3.1.0 añade mecanismos de encapsulamiento de claves AES que cumplen los estándares.

AWS CloudHSM Software de cliente

- Nuevo requisito para la actualización: la versión del cliente debe coincidir con la versión de cualquier biblioteca de software que esté utilizando. Para poder actualizar, debe utilizar un comando por lotes que actualice el cliente y todas las bibliotecas al mismo tiempo. Para obtener más información, consulte este artículo sobre Actualización de SDK 3 de cliente.
- Key_mgmt_util (KMU) incluye las siguientes actualizaciones:
  - Se han agregado dos nuevos métodos de encapsulamiento de claves AES con y sin relleno de ceros a la izquierda que cumplen los estándares. Para obtener más información, consulte wrapKey y unwrapKey.
  - Se ha deshabilitado la funcionalidad que permitía especificar un IV personalizado al encapsular una clave con AES_KEY_WRAP_PAD_PKCS5. Para obtener más información, consulte este artículo sobre el encapsulamiento de claves AES.

Biblioteca PKCS #11

- Se han agregado dos nuevos métodos de encapsulamiento de claves AES con y sin relleno de ceros a la izquierda que cumplen los estándares. Para obtener más información, consulte este artículo sobre el encapsulamiento de claves AES.
- Puede configurar la longitud de sal en las firmas RSA-PSS. Para obtener información sobre cómo utilizar esta función, consulte Longitud de sal configurable para las firmas RSA-PSS activadas. GitHub

Motor dinámico de OpenSSL

- CAMBIO IMPORTANTE: los conjuntos de cifrado TLS 1.0 y 1.2 con SHA1 no están disponibles en la versión 3.1.0 del motor de OpenSSL. Este problema se resolverá en breve.
- Si desea instalar la biblioteca del motor dinámico de OpenSSL en RHEL 6 o CentOS 6, infórmese sobre un problema conocido relacionado con la versión predeterminada de OpenSSL instalada en esos sistemas operativos.
- Mejoras de estabilidad y correcciones de errores

Proveedor de JCE

- CAMBIO IMPORTANTE: para solucionar los problemas de conformidad de Java Cryptography Extension (JCE), ahora, en el encapsulamiento y desencapsulamiento de AES, se utiliza correctamente el algoritmo de AESWrap en lugar del algoritmo de AES. Esto significa que

`Cipher.WRAP_MODE` y `Cipher.UNWRAP_MODE` ya no tienen éxito para los mecanismos AES/ECB y AES/CBC.

Para actualizar a la versión de cliente 3.1.0, debe actualizar el código. Si tiene claves encapsuladas existentes, debe prestar especial atención al mecanismo que usa para desencapsular y a cómo han cambiado los valores predeterminados del IV. Si encapsuló claves con la versión de cliente 3.0.0 o anterior, entonces en 3.1.1 debe usar AESWrap/ECB/PKCS5Padding para desencapsular sus claves existentes. Para obtener más información, consulte este artículo sobre el encapsulamiento de claves AES.

- Puede mostrar varias claves con la misma etiqueta desde la biblioteca de JCE. Para aprender a recorrer en iteraciones todas las claves disponibles, consulte Buscar todas las claves activadas. GitHub
- Al crear las claves, puede establecer valores más restrictivos en los atributos; por ejemplo, especificar etiquetas diferentes para claves públicas y privadas. Para obtener más información, consulte este artículo sobre los atributos de Java admitidos.

Windows (CNG, KSP)

- Mejoras de estabilidad y correcciones de errores.

Lanzamientos E nd-of-life

AWS CloudHSM anuncia el fin de la vida útil de las versiones que ya no son compatibles con el servicio. Para preservar la seguridad de tu aplicación, nos reservamos el derecho de rechazar activamente las conexiones desde las end-of-life versiones.

- Actualmente, no se ha publicado ninguna versión del SDK del end-of-life cliente.

Historial de documentos

En este tema se describen actualizaciones importantes en la Guía del usuario de AWS CloudHSM .

Temas

- Actualizaciones recientes
- Actualizaciones anteriores

Actualizaciones recientes

En la siguiente tabla se describen cambios importantes en esta documentación desde abril 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 importante, utilice el enlace de la esquina superior derecha para suscribirse a la fuente RSS.

Para obtener más información sobre las nuevas versiones, consulte Descargas para AWS CloudHSM Client SDK

Cambio	Descripción	Fecha
Se añadió una versión nueva.	Publicada la versión 5.12.0 del AWS CloudHSM cliente.	20 de marzo de 2024
Se añadió una versión nueva.	Publicada la versión AWS CloudHSM 5.11.0 del cliente.	17 de enero de 2024
Se añadió una versión nueva.	Publicada la versión AWS CloudHSM 5.10.0 del cliente.	28 de julio de 2023
Se añadió una versión nueva.	Publicada la versión AWS CloudHSM 5.9.0 del cliente.	23 de mayo de 2023
Se añadió una versión nueva.	Publicada la versión 5.8.0 del AWS CloudHSM cliente.	16 de marzo de 2023

Se añadió una versión nueva.	Publicada la versión 5.7.0 del AWS CloudHSM cliente.	16 de noviembre de 2022
Se añadió una versión nueva.	Publicada la versión 5.6.0 del AWS CloudHSM cliente.	1 de septiembre de 2022
Se añadió una versión nueva.	Publicada la versión 5.5.0 del AWS CloudHSM cliente.	13 de mayo de 2022
Se añadió una versión nueva.	Publicada la versión 5.4.2 del AWS CloudHSM cliente.	18 de marzo de 2022
Se añadió una versión nueva.	Publicada la versión 5.4.1 del AWS CloudHSM cliente.	10 de febrero de 2022
Se añadió una versión nueva.	Publicada la versión 5.4.0 del proveedor AWS CloudHSM JCE para plataformas Windows.	1 de febrero de 2022
Se añadió una versión nueva.	Publicó la versión 5.4.0 del AWS CloudHSM cliente, que añade soporte inicial para el proveedor JCE en todas las plataformas Linux.	28 de enero de 2022
Se añadió una versión nueva.	Publicada la versión AWS CloudHSM 5.3.0 del cliente.	3 de enero de 2022
Se añadió una versión nueva.	Publicada la versión 3.4.4 del AWS CloudHSM cliente.	3 de enero de 2022
Se añadió una versión nueva.	Publicada la versión 3.4.3 del AWS CloudHSM cliente.	20 de diciembre de 2021
Se añadió una versión nueva.	Publicada la versión 3.4.2 del AWS CloudHSM cliente.	15 de diciembre de 2021

Se añadió una versión nueva.	Publicada la versión 3.4.1 del AWS CloudHSM cliente.	10 de diciembre de 2021
Se añadió una versión nueva.	Publicada la versión 5.2.1 del AWS CloudHSM cliente.	4 de octubre de 2021
Se añadió una versión nueva.	Publicada la versión 3.4.0 del AWS CloudHSM cliente.	25 de agosto de 2021
Se añadió una versión nueva.	Publicada la versión 5.2.0 del AWS CloudHSM cliente.	3 de agosto de 2021
Se añadió una versión nueva.	Publicada la versión 3.3.2 del AWS CloudHSM cliente.	2 de julio de 2021
Se añadió una versión nueva.	Publicada la versión 5.1.0 del AWS CloudHSM cliente.	1 de junio de 2021
Se añadió una versión nueva.	Publicada la versión 3.3.1 del AWS CloudHSM cliente.	26 de abril de 2021
Se añadió una versión nueva.	Publicada la versión 5.0.1 del AWS CloudHSM cliente.	8 de abril de 2021
Se añadió una versión nueva.	Publicada la versión 5.0.0 del AWS CloudHSM cliente.	12 de marzo de 2021
Se añadió nuevo contenido.	Se agregó el punto de enlace VPC de interfaz, una función de AWS que le permite crear una conexión privada entre su VPC AWS CloudHSM sin necesidad de acceso a través de Internet o mediante un dispositivo NAT, una conexión VPN o una conexión. AWS Direct Connect	10 de febrero de 2021

Se añadió una versión nueva.	Publicada la versión AWS CloudHSM 3.3.0 del cliente.	3 de febrero de 2021
Agregar contenido nuevo.	Se agregó la retención administrada de copias de seguridad, una característica que elimina automáticamente las copias de seguridad antiguas.	18 de noviembre de 2020
Agregar contenido nuevo.	Se agregó un informe de conformidad que analiza la implementación 3.2.1 del SDK de AWS CloudHSM cliente de la biblioteca PKCS #11 con el estándar PKCS #11.	29 de octubre de 2020
Se añadió una versión nueva.	Publicada la versión 3.2.1 AWS CloudHSM del cliente.	8 de octubre de 2020
Se añadió nuevo contenido.	Se agregó documentación que describe la configuración de sincronización de claves en AWS CloudHSM.	1 de septiembre de 2020
Se añadió una versión nueva.	Publicada la versión 3.2.0 del AWS CloudHSM cliente.	31 de agosto de 2020
Se añadió una versión nueva.	Publicada la versión 3.1.2 del AWS CloudHSM cliente.	30 de julio de 2020
Se añadió una versión nueva.	Publicada la versión 3.1.1 del AWS CloudHSM cliente.	3 de junio de 2020
Se añadió una versión nueva.	Publicada la versión 3.1.0 del AWS CloudHSM cliente.	21 de mayo de 2020
Se añadió una versión nueva.	Publicada la versión 3.0.1 del AWS CloudHSM cliente.	20 de abril de 2020

Se añadió una versión nueva.	Publicada la versión 3.0.0 del AWS CloudHSM cliente para la plataforma Windows Server.	30 de octubre de 2019
Se añadió una versión nueva.	Publicada la versión 3.0.0 del AWS CloudHSM cliente para todas las plataformas, excepto Windows.	22 de octubre de 2019
Se añadió una versión nueva.	Publicada la versión 2.0.4 del AWS CloudHSM cliente.	26 de agosto de 2019
Se añadió una versión nueva.	Publicada la versión 2.0.3 del AWS CloudHSM cliente.	13 de mayo de 2019
Se añadió una versión nueva.	Publicada la versión 2.0.1 del AWS CloudHSM cliente.	21 de marzo de 2019
Se añadió una versión nueva.	Publicada la versión 2.0.0 del AWS CloudHSM cliente.	6 de febrero de 2019
Se agregó compatibilidad con regiones	Se agregó AWS CloudHSM soporte para las regiones de la UE (Estocolmo) y AWS GovCloud (EE. UU. Este).	19 de diciembre de 2018
Se añadió una versión nueva.	Publicada la versión 1.1.2 del AWS CloudHSM cliente para Windows.	20 de noviembre de 2018
Se actualizaron los problemas conocidos.	Se añadió contenido nuevo a la guía de solución de problemas.	8 de noviembre de 2018
Se añadió una versión nueva.	Publicada la versión 1.1.2 del AWS CloudHSM cliente para plataformas Linux.	8 de noviembre de 2018

Se agregó compatibilidad con regiones	Se agregó AWS CloudHSM soporte para las regiones de la UE (París) y Asia Pacífico (Seúl).	24 de octubre de 2018
Se añadió nuevo contenido.	Se ha añadido la posibilidad de eliminar y restaurar AWS CloudHSM copias de seguridad.	10 de septiembre de 2018
Se añadió nuevo contenido.	Se ha añadido la entrega automática de registros de auditoría a Amazon CloudWatch Logs.	13 de agosto de 2018
Se añadió nuevo contenido.	Se agregó la posibilidad de copiar una copia de seguridad de un AWS CloudHSM clúster en todas las regiones.	30 de julio de 2018
Se agregó compatibilidad con regiones	Se agregó AWS CloudHSM soporte para la región de la UE (Londres).	13 de junio de 2018
Se añadió nuevo contenido.	Se agregó soporte de AWS CloudHSM cliente y biblioteca para Amazon Linux 2, Red Hat Enterprise Linux (RHEL) 6, Red Hat Enterprise Linux (RHEL) 7, CentOS 6, CentOS 7 y Ubuntu 16.04 LTS.	10 de mayo de 2018
Se añadió una versión nueva.	Se agregó un cliente de Windows. AWS CloudHSM	30 de abril de 2018

Actualizaciones anteriores

En la siguiente tabla se describen los cambios importantes con respecto a los AWS CloudHSM anteriores a 2018.

Cambio	Descripción	Fecha
Nuevo contenido	Se agregó la autenticación de cuórum (control de acceso M de N) para los responsables de criptografía (CO). Para obtener más información, consulte Usar la Utilidad de administración de CloudHSM (CMU) para administrar la autenticación de cuórum (control de acceso M de N).	9 de noviembre de 2017
Actualización	Se agregó documentación acerca del uso de la herramienta de línea de comandos key_mgmt_util. Para obtener más información, consulte referencia del comando cloudhsm_mgmt_util.	9 de noviembre de 2017
Nuevo contenido	Se agregó el cifrado de datos transparente de Oracle. Para obtener más información, consulte Cifrado de Oracle Database.	25 de octubre de 2017
Nuevo contenido	Se agregó la descarga de SSL. Para obtener más información, consulte Descarga de SSL/TLS.	12 de octubre de 2017

Cambio	Descripción	Fecha
Nueva guía	Esta versión presenta AWS CloudHSM	14 de agosto de 2017

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.