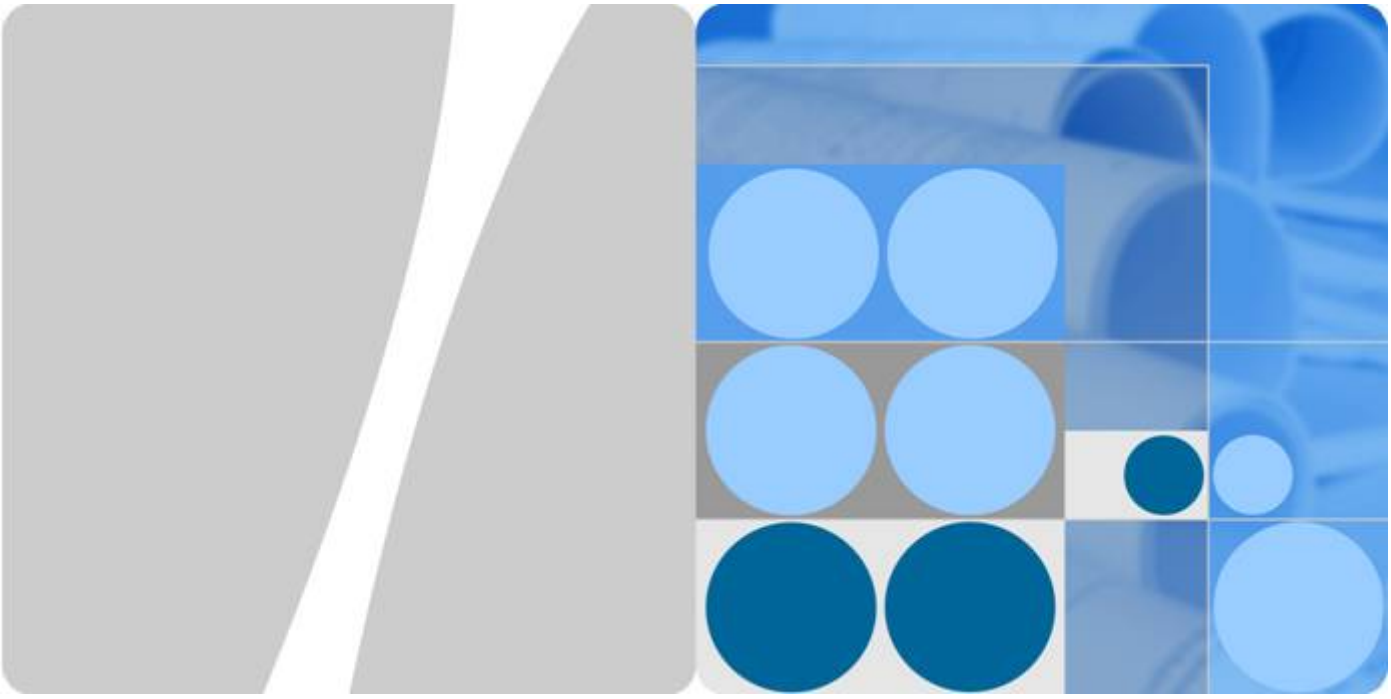


Key Management Service

User Guide

Date **2023-02-28**

Contents

1 Service Overview

1.1 Key Management Service

Key Management Service (KMS) is a secure, reliable, and easy-to-use service that helps users centrally manage and safeguard their Customer Master Keys (CMKs).

KMS uses hardware security modules (HSMs) to protect CMKs. HSMs help you create and control CMKs with ease. All CMKs are protected by root keys in HSMs to avoid leakage.

It also controls access to keys and records all operations on keys with traceable logs. In addition, it provides use records of all keys, meeting your audit and regulatory compliance requirements.

Table 1-1 Basic concepts

Item	Definition
Customer Master Key (CMK)	A CMK is a Key Encryption Key (KEK) created by a user using KMS. It is used to encrypt and protect Data Encryption Keys (DEKs). One CMK can be used to encrypt one or more DEKs.
Default Master Key (DMK)	A Default Master Key is automatically created by another cloud service using KMS, such as Object Storage Service (OBS). The alias of a Default Master Key ends with **/default**. For details about the corresponding cloud services, see **Default Master Keys**.
Data Encryption Key (DEK)	A data encryption key (DEK) is a key used for encrypting data.
Hardware Security Module (HSM)	A hardware device that securely produces, stores, manages, and uses keys and provides encryption services.

Item	Definition
True Random Number Generator (TRNG)	A device that generates random numbers through physical processes instead of computer programs.
Project	A project is used to group and isolate OpenStack resources, including computing, storage, and network resources. A project can be a department or a project team. Multiple projects can be created for one account.

Table 1-2 Default Master Keys

Alias	Cloud Service
obs/default	OBS
evs/default	Elastic Volume Service (EVS)
ims/default	Image Management Service (IMS)

NOTE

A Default Master Key is automatically created when a user employs the KMS encryption function for the first time in another cloud service.

1.2 KMS

1.2.1 Functions

KMS is a secure, reliable, and easy-to-use cloud service that helps users create, manage, and protect keys in a centralized manner.

It uses Hardware Security Modules (HSMs) to protect keys. All CMKs are protected by root keys in HSMs to avoid key leakage.

It also controls access to keys and records all operations on keys with traceable logs. In addition, it provides use records of all keys, meeting your audit and regulatory compliance requirements.

Functions

- On the KMS console, you can perform the following operations on CMKs:
  - Creating, querying, enabling, disabling, scheduling the deletion of, and canceling the deletion of CMKs

  - Modifying the alias and description of CMKs
  - Using the Online Tool to Encrypt and Decrypt Small-Size Data
  - Importing CMKs and deleting CMK material
  - Adding, searching for, editing, and deleting tags
  - Creating, canceling, and querying grants
- You can use the API to perform the following operations:
  - Creating, encrypting, or decrypting data encryption keys (DEKs)
  - Retiring grants

  For details, see the **Key Management Service API Reference**.
- Generate hardware true random number.

  You can generate 512-bit random numbers using the KMS API. The 512-bit hardware true random numbers can be used as or serve as basis for key materials and encryption parameters. For details, see the **Key Management Service API Reference**.

Cryptographic Algorithms Supported by KMS

Table 1-3 describes the key wrapping encryption and decryption algorithms supported by imported keys.

Table 1-3 Key wrapping algorithms

Algorithm	Description	Configuration
RSAES_OAEP_SHA_256	RSA encryption algorithm that uses OAEP and has the **SHA-256** hash function	Select an encryption algorithm based on your HSM functions. 1. If your HSM supports the **RSAES_OAEP_SHA_256** algorithm, use **RSAES_OAEP_SHA_256** to encrypt key materials. 2. If your HSM does not support **OAEP**, use **RSAES_PKCS1_V1_5** to encrypt key materials. **NOTICE** The **RSAES_OAEP_SHA_1** encryption algorithm is no longer secure. Exercise caution when performing this operation.
RSAES_PKCS1_V1_5	Rivest-Shamir_Adleman (RSA) encryption algorithm (v1.5) of Public-Key Cryptography Standards number 1 (PKCS #1)	
RSAES_OAEP_SHA_1	RSA encryption algorithm that uses Optimal Asymmetric Encryption Padding (OAEP) and has the **SHA-1** hash function	

1.2.2 Product Advantages

- Extensive Service Integration

  KMS can be integrated with Object Storage Service (OBS), Elastic Volume Service (EVS), and Image Management Service (IMS), to manage keys of these services on the KMS console, and encrypt and decrypt your local data by making the KMS API calls.

- Regulatory Compliance

  Keys are generated by third-party validated HSMs. Access to keys is controlled and all operations involving keys are traceable by logs, compliant with international laws and regulations.

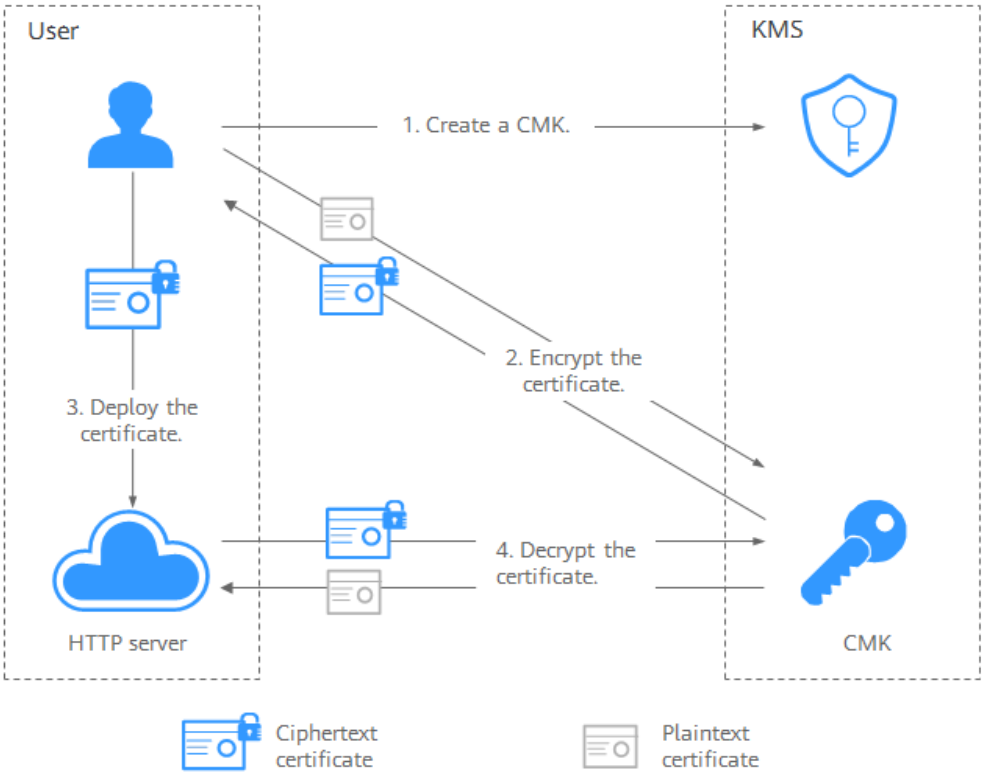
1.2.3 Application Scenarios

Small Data Encryption and Decryption

You can use the online tool on the KMS console or call KMS APIs to directly encrypt or decrypt a small amount of data, such as passwords, certificates, or phone numbers. Currently, a maximum of 4 KB of data can be encrypted or decrypted in this way.

Figure 1-1 shows an example about how to call the APIs to encrypt and decrypt an HTTPS certificate.

Figure 1-1 Encrypting and decrypting an HTTPS certificate



The procedure is as follows:

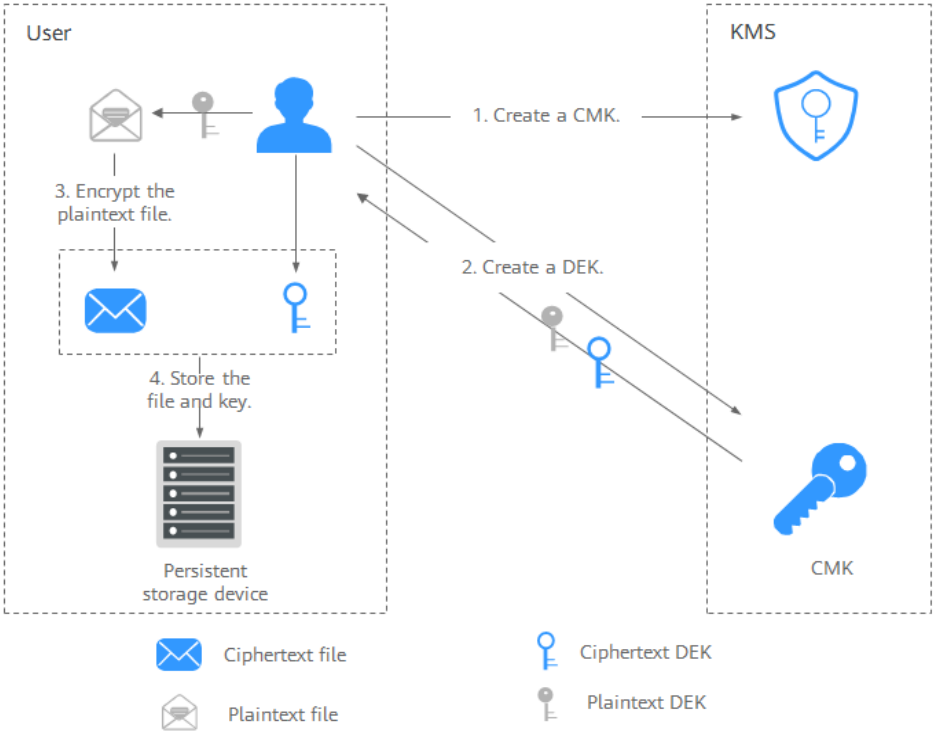
1. Create a CMK on KMS.
2. Call the **encrypt-data** API of KMS and use the CMK to encrypt the plaintext certificate.
3. Deploy the certificate onto a server.
4. The server calls the **decrypt-data** API of KMS to decrypt the ciphertext certificate.

Large Data Encryption and Decryption

If you want to encrypt or decrypt large volumes of data, such as pictures, videos, and database files, you can use the envelope encryption method, where the data does not need to be transferred over the network.

- **Figure 1-2** illustrates the process for encrypting a local file.

  Figure 1-2 Encrypting a local file

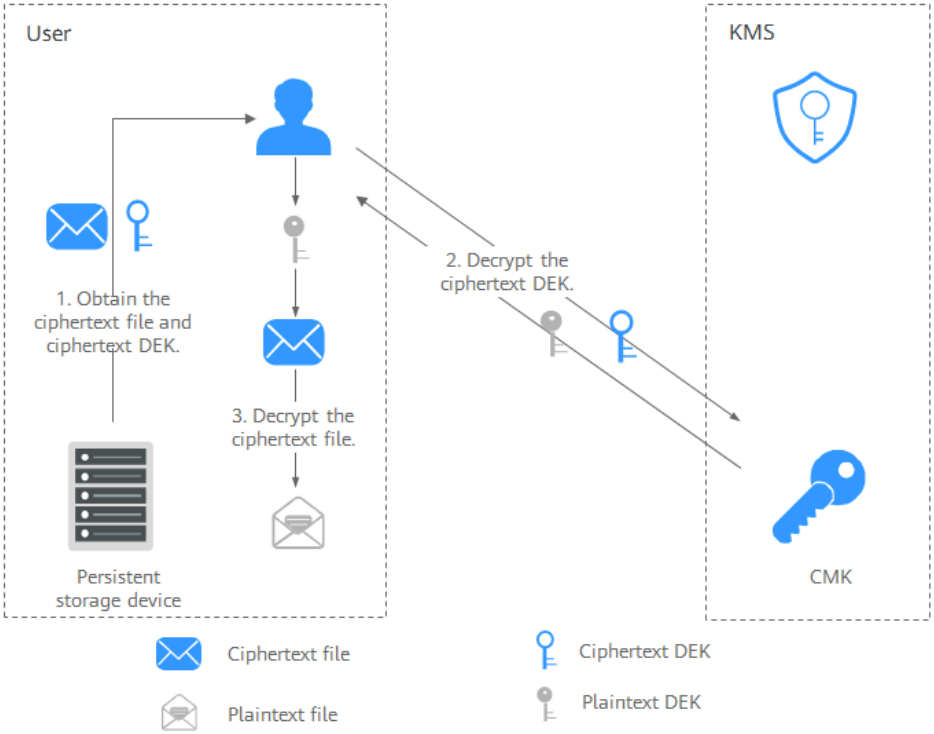
  

  The procedure is as follows:

  a. Create a CMK on KMS.
  b. Call the **create-datakey** API of KMS to create a DEK. Then you get a plaintext DEK and a ciphertext DEK. The ciphertext DEK is generated when you use a CMK to encrypt the plaintext DEK.
  c. Use the plaintext DEK to encrypt the file. A ciphertext file is generated.
  d. Save the ciphertext DEK and the ciphertext file together in a persistent storage device or a storage service.

- **Figure 1-3** illustrates the process for decrypting a local file.

Figure 1-3 Decrypting a local file



The procedure is as follows:

a. Obtain the ciphertext DEK and file from the persistent storage device or the storage service.

b. Call the **decrypt-datakey** API of KMS and use the corresponding CMK (the one used for encrypting the DEK) to decrypt the ciphertext DEK. Then you get the plaintext DEK.

   If the CMK is deleted, the decryption fails. Therefore, properly keep your CMKs.

c. Use the plaintext DEK to decrypt the ciphertext file.

1.2.4 Using KMS

Interacting with Cloud Services

Cloud services use the envelope encryption technology and call KMS APIs to encrypt service resources. Your CMKs are under your own management. With your grant, cloud services use a specific CMK of yours to encrypt data.

The encryption process is as follows:

1. Create a CMK on KMS.
2. Cloud services call the **create-datakey** API of the KMS to create a DEK. Then you get a plaintext DEK and a ciphertext DEK.

   NOTE

   Ciphertext DEKs are generated when you use a CMK to encrypt the plaintext DEKs.

3. Cloud services use the plaintext DEK to encrypt a plaintext file, generating a ciphertext file.
4. Cloud services store the ciphertext DEK and ciphertext file in a persistent storage device or a storage service.

NOTE

When users download the data from a cloud service, the service uses the CMK specified by KMS to decrypt the ciphertext DEK, uses the decrypted DEK to decrypt data, and then provides the decrypted data for users to download.

Table 1-4 List of cloud services that use KMS encryption

Service Name	Description
Object Storage Service (OBS)	You can upload objects to and download them from Object Storage Service (OBS) in common mode or server-side encryption mode. When you upload objects in encryption mode, data is encrypted at the server side and then securely stored on OBS in ciphertext. When you download encrypted objects, the data in ciphertext is decrypted at the server side and then provided to you in plaintext. OBS supports the server-side encryption with KMS-managed keys (SSE-KMS) mode. In SSE-KMS mode, OBS uses the keys provided by KMS for server-side encryption. For details about how to upload objects to OBS in SSE-KMS mode, see the **Object Storage Service User Guide**.
Elastic Volume Service (EVS)	If you enable the encryption function when creating an EVS disk, the disk will be encrypted with the DEK generated by using your CMK. Data stored in the EVS disk will be automatically encrypted. For details about how to use the encryption function of EVS, see **Elastic Volume Service User Guide**.
Image Management Service (IMS)	When creating a private image using an external image file, you can enable the private image encryption function and select a CMK provided by KMS to encrypt the image. For details about how to use the private image encryption function of Image Management Service (IMS), see **Image Management Service User Guide**.

Working with User Applications

To encrypt plaintext data, a user application can call the necessary KMS API to create a DEK. The DEK can then be used to encrypt the plaintext data. Then the application can store the encrypted data. In addition, the user application can call the KMS API to create CMKs. DEKs can be stored in ciphertext after being encrypted with the CMKs.

Envelope encryption is implemented, with CMKs stored in KMS and ciphertext DEKs in user applications. KMS is called to decrypt a ciphertext DEK only when necessary.

The encryption process is as follows:

1. The application calls the **create-key** API of KMS to create a CMK.
2. The application calls the **create-datakey** API of KMS to create a DEK. A plaintext DEK and a ciphertext DEK are generated.

   NOTE

   Ciphertext DEKs are generated when you use a CMK to encrypt the plaintext DEKs in 1.
3. The application uses the plaintext DEK to encrypt a plaintext file. A ciphertext file is generated.
4. The application saves the ciphertext DEK and the ciphertext file together in a persistent storage device or a storage service.

For details, see the **Key Management Service API Reference**.

1.2.5 Cloud Services with KMS Integrated

1.2.5.1 Encrypting Data in OBS

- When using Object Storage Service (OBS) to upload files with server-side encryption, you can select KMS encryption and use the key provided by KMS to encrypt the files to be uploaded. For more information about OBS, see the **Object Storage Service User Guide**.

  There are two types of CMKs that can be used:

  - The default master key **obs/default** created by KMS
  - CMKs that you create on the KMS console using KMS-generated key materials
- Alternatively, you can call OBS APIs to upload a file with server-side encryption using KMS-managed keys (SSE-KMS). For details, see the **Object Storage Service API Reference**.

1.2.5.2 Encrypting Data in EVS

- When purchasing a disk, you can choose **Advanced Settings** > **Configure** > **Encryption** to encrypt the disk using the key provided by KMS. For more information about EVS, see the **Elastic Volume Service User Guide**.

  NOTE

  Before you use the encryption function, EVS must be granted the permission to access KMS. If you have the right to grant the permission, you can grant the permission directly. If you do not have the permission, contact a user with the security administrator permissions to add the security administrator permission for you. Then, you can grant the permission. For more information about EVS, see the **Elastic Volume Service User Guide**.

  There are two types of CMKs that can be used:

  - The default master key **evs/default** created by KMS
  - CMKs that you create on the KMS console using KMS-generated key materials
- You can also call EVS APIs to create encrypted EVS disks. For details, see the **Elastic Volume Service API Reference**.

1.2.5.3 Encrypting Data in IMS

- When uploading an image file to Image Management Service (IMS), you can choose to encrypt the image file using a key provided by KMS to protect the file. For details, see the **Image Management Service User Guide**.

  There are two types of CMKs that can be used:

  - The default master key **ims/default** created by KMS
  - CMKs that you create on the KMS console using KMS-generated key materials
- You can also call IMS APIs to create encrypted image files. For details, see **Image Management Service API Reference**.

1.3 Permissions Management

If you want to assign different access permissions to employees in an enterprise for the KMS resources purchased on the cloud platform, you can use Identity and Access Management (IAM) to perform refined permission management. IAM provides identity authentication, permissions management, and access control, helping you secure the access to your cloud resources.

With IAM, you can use your account to create IAM users for your employees, and assign permissions to control their access to specific resource types. For example, if you have software developers and you want to assign them the permission to access KMS but not to delete KMS or its resources, then you can create an IAM policy to assign the developers the permission to access KMS but prevent them from deleting KMS related data.

If the system account has met your requirements and you do not need to create an independent IAM user for permission control, then you can skip this section. This will not affect other functions of KMS.

KMS Permissions

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups they are added to and can perform specified operations on cloud services based on the permissions.

KMS is a project-level service deployed and accessed in specific physical regions. To assign permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. Users need to switch to the authorized region when accessing KMS.

You can grant users permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you must also assign other roles that the permissions depend on to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under

certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant KMS users only the permissions for managing a certain type of cloud servers. Most policies contain permissions for specific APIs, and permissions are defined using API actions.

For more information, see **Table 1-5**.

Table 1-5 KMS permissions

Role/Policy Name	Description	Type	Dependency
KMS Administrator	Administrator permissions for the encryption key	System role	None

The following table describes the common operations supported by each system-defined permission of KMS. Select the permissions as needed.

Table 1-6 Common operations supported by each system-defined policy or role

Operation	KMS Administrator
Create a key	√
Enable a key	√
Disable a key	√
Schedule key deletion	√
Cancel scheduled key deletion	√
Modify a key alias	√
Modify key description	√
Generate a random number	√
Create a DEK	√
Create a plaintext-free DEK	√
Encrypt a DEK	√
Decrypt a DEK	√
Obtain parameters for importing a key	√
Import key materials	√
Delete key materials	√
Create a grant	√
Revoking a grant	√
Retire a grant	√

Operation	KMS Administrator
Query the grant list	√
Query retirable grants	√
Encrypt data	√
Decrypt data	√
Enable key rotation	√
Modify key rotation interval	√
Disable key rotation	√
Query key rotation status	√
Query CMK instances	√
Query key tags	√
Query project tags	√
Batch add or delete key tags	√
Add tags to a key	√
Delete key tags	√
Query the key list	√
Query key details	√
Query instance quantity	√
Query quotas	√

Helpful Links

- Two types of permission policies are provided by default: default policies and custom policies. Default policies are pre-defined by IAM and cannot be modified. If default policies do not meet your requirements, you can create custom policies for fine-grained permission control.
- Configure permission policies for a user group and add users to the group so that these users can obtain operation permissions defined in the policies.

1.4 How to Access

- Management console

  Log in to the management console. In the upper left corner, click . Select a region or project. Click  and choose **Security** > **Key Management Service**.

- API

  You can access KMS using the API. For details, see the *Key Management Service API Reference*.

1.5 Related Services

OBS

Object Storage Service (OBS) is a cloud storage service optimized for storing massive amounts of data. It provides unlimited, secure, and highly reliable storage capabilities at a relatively low cost. KMS provides central management and control capabilities of CMKs for OBS. It is used for server-side encryption with KMS-managed keys (SSE-KMS) on OBS.

EVS

Elastic Volume Service (EVS) offers scalable block storage for cloud servers. With high reliability, high performance, and rich specifications, EVS disks can be used for distributed file systems, development and test environments, data warehouse applications, and high-performance computing (HPC) scenarios to meet diverse service requirements. KMS provides central management and control capabilities of CMKs for EVS. It is used for encryption in EVS.

IMS

Image Management Service (IMS) supports lifecycle management for images. KMS provides central management and control capabilities of CMKs for Image Management Service (IMS). It is used for private image encryption in IMS.

ECS

Elastic Cloud Server (ECS) is a basic computing component that consists of CPUs, memory, OS, and EVS. After creating an ECS, you can use it like your local computer or physical server.

Dedicated HSM can encrypt sensitive data in the service systems on your ECS. You can control the generation, storage, and access authorization of keys to ensure the integrity and confidentiality of data during transmission and storage.

CTS

Cloud Trace Service (CTS) provides you with a history of KMS operations. After the CTS service is enabled, you can view all generated traces to review and audit performed KMS operations. For details, see the **Cloud Trace Service User Guide**.

Table 1-7 KMS operations supported by CTS

Operation	Resource Type	Trace Name
Creating a CMK	cmk	createKey
Creating a DEK	cmk	createDataKey

Operation	Resource Type	Trace Name
Creating a plaintext-free DEK	cmk	createDataKeyWithoutPlaintext
Enabling a CMK	cmk	enableKey
Disabling a CMK	cmk	disableKey
Encrypting a DEK	cmk	encryptDatakey
Decrypting a DEK	cmk	decryptDatakey
Scheduling the deletion of a CMK	cmk	scheduleKeyDeletion
Canceling the scheduled deletion of a CMK	cmk	cancelKeyDeletion
Generating random numbers	rng	genRandom
Changing the alias of a CMK	cmk	updateKeyAlias
Changing the description of a CMK	cmk	updateKeyDescription
Prompting risks about CMK deletion	cmk	deleteKeyRiskTips
Importing key material	cmk	importKeyMaterial
Deleting key material	cmk	deleteImportedKeyMaterial
Creating a grant	cmk	createGrant
Retiring a grant	cmk	retireGrant
Revoking a grant	cmk	revokeGrant
Encrypting data	cmk	encryptData
Decrypting data	cmk	decryptData
Adding a tag	cmk	createKeyTag
Deleting a tag	cmk	deleteKeyTag
Adding or deleting tags in batches	cmk	batchCreateKeyTags
Batch deleting tags	cmk	batchDeleteKeyTags
Enabling key rotation	cmk	enableKeyRotation
Modifying key rotation interval	cmk	updateKeyRotationInterval
Disabling key rotation	cmk	disableKeyRotation

IAM

Identity and Access Management (IAM) provides the permission management function for KMS.

Only users who have KMS Administrator permissions can use KMS.

To apply for permissions, contact a user with Security Administrator permissions. For details, see the **Identity and Access Management User Guide**.

2 Key Management Service

2.1 Key Types

CMKs can be categorized into symmetric keys and asymmetric keys.

Symmetric keys are commonly used for data encryption. Asymmetric keys are used for digital signature verification or sensitive information encryption in systems where the trust relationship is not mutual. An asymmetric key consists of a public key and a private key. The public key can be sent to anyone. The private key must be securely stored and only accessible to trusted users.

An asymmetric key can be used to generate and verify a signature. To securely transfer data, a signer sends the public key to a receiver, uses the private key to sign data, and then sends the data and signature to the receiver. The receiver can use the public key to verify the signature.

Table 2-1 Key algorithms supported by KMS

Key Type	Algorithm Type	Key Specifications	Description	Usage
Symmetric key	AES	AES_256	AES symmetric key	Encrypts and decrypts a small amount of data or data keys.
Asymmetric key	RSA	• RSA_2048 • RSA_3072 • RSA_4096	RSA asymmetric password	Encrypts and decrypts a small amount of data or creates digital signatures.
	ECC	• EC_P256 • EC_P384	Elliptic curve recommended by NIST	Digital signature

2.2 Creating a CMK

This section describes how to create a CMK on the KMS console.

Constraints

- You can create up to 100 CMKs, excluding default master keys.
- Aliases of default master keys end with **/default**. Therefore, in choosing aliases for your CMKs, do not use aliases ending with **/default**.
- KMS does not limit the number of times that a CMK can be called.

Scenarios

- Encrypt data in OBS
- Encrypt data in EVS
- Encrypt data in IMS
- Encrypt an RDS DB instance
- Direct encryption and decryption of small volumes of data
- DEK encryption and decryption for user applications

Creating a CMK

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click **Create Key** in the upper right corner.

Step 5 Configure parameters in the **Create Key** dialog box.

- **Alias** is the alias of the CMK to be created.

  NOTE

  - You can enter digits, letters, underscores (_), hyphens (-), colons (:), and slashes (/).
  - You can enter up to 255 characters.

- **Key Algorithm**: Select a key algorithm. For more information, see **Table 2-2**.

Table 2-2 Key algorithms supported by KMS

Key Type	Algorithm Type	Key Specifications	Description	Usage
Symmetric key	AES	AES_256	AES symmetric key	Encrypts and decrypts a small amount of data or data keys.
Asymmetric key	RSA	– RSA_2048 – RSA_3072 – RSA_4096	RSA asymmetric password	Encrypts and decrypts a small amount of data or creates digital signatures.
	ECC	– EC_P256 – EC_P384	Elliptic curve recommended by NIST	Digital signature

- **Usage**: Select **SIGN_VERIFY** or **ENCRYPT_DECRYPT**.
  - For a symmetric key, the default value is **ENCRYPT_DECRYPT**.
  - For RSA asymmetric keys, select **ENCRYPT_DECRYPT** or **SIGN_VERIFY**. The default value is **SIGN_VERIFY**.
  - For an ECC asymmetric key, the default value is **SIGN_VERIFY**.

  NOTE

  The key usage can only be configured during key creation and cannot be modified afterwards.
- (Optional) **Description** is the description of the CMK.

  NOTE

  You can enter up to 255 characters.

Step 6 (Optional) Add tags to the CMK as needed, and enter the tag key and tag value.

NOTE

- When a CMK has been created without any tag, you can add a tag to the CMK later as necessary. Click the alias of the CMK, click the **Tags** tab, and click **Add Tag**.
- The same tag (including tag key and tag value) can be used for different CMKs. However, under the same CMK, one tag key can have only one tag value.
- A maximum of 20 tags can be added for one CMK.
- If you want to delete a tag from the tag list when adding multiple tags, you can click **Delete** in the row where the tag to be added is located to delete the tag.

Step 7 Click **OK**. A message is displayed in the upper right corner of the page, indicating that the key is created successfully.

In the CMK list, you can view created CMKs. The default status of a CMK is **Enabled**.

----End

Related Operations

- For details about how to upload objects with server-side encryption, see section "Uploading a File with Server-Side Encryption" in the **Object Storage Service User Guide**.
- For details about how to encrypt data on EVS disks, see section "Creating an EVS Disk" in the **Elastic Volume Service User Guide**.
- For details about how to encrypt private images, see section "Encrypting an Image" in the **Image Management Service User Guide**.
- For details about how to create a DEK and a plaintext-free DEK, see sections "Creating a DEK" and "Creating a Plaintext-Free DEK" in the **Key Management Service API Reference**.
- For details about how to encrypt and decrypt a DEK for a user application, see sections "Encrypting a DEK" and "Decrypting a DEK" in the **Key Management Service API Reference**.

2.3 Creating CMKs Using Imported Key Materials

2.3.1 Overview

A CMK contains key metadata (key ID, key alias, description, key status, and creation date) and key materials used for encrypting and decrypting data.

- When a user uses the KMS console to create a CMK, the KMS automatically generates a key material for the CMK.
- If you want to use your own key material, you can use the key import function on the KMS console to create a CMK whose key material is empty, and import the key material to the CMK.

Important Notes

- Security

  You need to ensure that random sources meet your security requirements when using them to generate key materials. When using the import key function, you need to be responsible for the security of your key materials. Save the original backup of the key material so that the backup key material can be imported to the KMS in time when the key material is deleted accidentally.

- Availability and Durability

  Before importing the key material into KMS, you need to ensure the availability and durability of the key material.

  Differences between the imported key material and the key material generated by KMS are shown in **Table 2-3**.

Table 2-3 Differences between the imported key material and the key material generated by KMS

Key Material Source	Difference
CMKs using imported key materials	• You can delete the key material, but cannot delete the CMK and its metadata. • Such keys cannot be rotated. • When importing the key material, you can set the expiration time of the key material. After the key material expires, the KMS automatically deletes the key material within 24 hours, but does not delete the CMK and its metadata. It is recommended that you save a copy of the material on your local device because it may be used for re-import in cases of invalid key materials or key material mis-deletion.
CMKs using KMS generated key materials	• The key material cannot be manually deleted. • Symmetric keys can be rotated. • You cannot set the expiration time for key material.

- Association

  When a key material is imported to a CMK, the CMK is permanently associated with the key material. Other key materials cannot be imported into the CMK.

- Uniqueness

  If you use the CMK created using the imported key material to encrypt data, the encrypted data can be decrypted only by the CMK that has been used to encrypt the data, because the metadata and key material of the CMK must be consistent.

2.3.2 Importing Key Materials

If you want to use your own key materials instead of the KMS-generated materials, you can use the console to import your key materials to KMS. CMKs created using imported materials and KMS-generated materials are managed together by KMS.

This section describes how to import key materials on the KMS console.

NOTE

- A CMK with imported material works in the same way as one using KMS-generated material, that is, you enable and disable them as well as schedule their deletion and cancel their scheduled deletion in the same way.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click **Import Key**. The **Import Key** dialog box is displayed.

Step 5 Configure key parameters.

- **Alias** is the alias of the CMK to be created.

  NOTE

  - You can enter digits, letters, underscores (_), hyphens (-), colons (:), and slashes (/).
  - You can enter up to 255 characters.

- **Key Algorithm**: Select a key algorithm. For more information, see **Table 2-4**.

  Table 2-4 Key algorithms supported by KMS

Key Type	Algorithm Type	Key Specifications	Description	Usage
Symmetric key	AES	AES_256	AES symmetric key	Encrypts and decrypts a small amount of data or data keys.
Asymmetric key	RSA	– RSA_2048 – RSA_3072 – RSA_4096	RSA asymmetric password	Encrypts and decrypts a small amount of data or creates digital signatures.
	ECC	– EC_P256 – EC_P384	Elliptic curve recommended by NIST	Digital signature

- **Usage**: Select **SIGN_VERIFY** or **ENCRYPT_DECRYPT**.
  - For a symmetric key, the default value is **ENCRYPT_DECRYPT**.
  - For RSA asymmetric keys, select **ENCRYPT_DECRYPT** or **SIGN_VERIFY**. The default value is **SIGN_VERIFY**.

  - For an ECC asymmetric key, the default value is **SIGN_VERIFY**.

    NOTE

    The key usage can only be configured during key creation and cannot be modified afterwards.

- (Optional) **Description** is the description of the CMK.

  NOTE

  You can enter up to 255 characters.

Step 6 (Optional) Add tags to the CMK as needed, and enter the tag key and tag value.

NOTE

- If a CMK was created without any tag, you can add a tag to the CMK later as necessary. Click the alias of the CMK, click the **Tags** tab, and click **Add Tag**.
- The same tag (including tag key and tag value) can be used for different CMKs. However, under the same CMK, one tag key can have only one tag value.
- A maximum of 20 tags can be added for each CMK.
- If you want to delete a tag from the tag list when adding multiple tags, you can click **Delete** in the row where the tag to be added is located to delete the tag.

Step 7 Click **security and durability** to understand the security and durability of the imported key.

Step 8 Select **I understand the security and durability of using an imported key**, and create a CMK whose key material is empty.

Step 9 Click **Next** to go to the **Download the Import Items** step. Select a key wrapping algorithm based on **Table 2-5**.

Table 2-5 Key wrapping algorithms

Algorithm	Description	Configuration
RSAES_OAEP_SHA_256	RSA encryption algorithm that uses OAEP and has the **SHA-256** hash function	Select an encryption algorithm based on your HSM functions. 1. If the HSMs support the **RSAES_OAEP_SHA_256** algorithm, use **RSAES_OAEP_SHA_256** to encrypt key materials. 2. If the HSMs do not support **OAEP**, use **RSAES_PKCS1_V1_5** to encrypt key materials. **NOTICE** The **RSAES_OAEP_SHA_1** encryption algorithm is no longer secure. Exercise caution when performing this operation.
RSAES_PKCS1_V1_5	Rivest-Shamir_Adleman (RSA) encryption algorithm (v1.5) of Public-Key Cryptography Standards number 1 (PKCS #1)	
RSAES_OAEP_SHA_1	RSA encryption algorithm that uses Optimal Asymmetric Encryption Padding (OAEP) and has the **SHA-1** hash function	

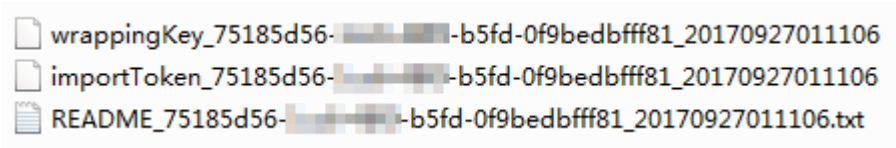
NOTE

If you stop a key material import process and want to try again, click **Import Key Material** in the row of the required CMK, and import key material in the dialog box that is displayed.

Step 10 Obtain the wrapping key and import token.

1. Obtain the wrapping key and import token.
   - Method 1: Click **Download**. The downloaded files include the wrapping key, import token, and description file, as shown below.

     Figure 2-1 Downloading a file

     - **wrappingKey**_*KeyID_DownloadTime* is the wrapping key. It is encoded in binary format and used to encrypt the wrapping key of the key material.
     - **importToken**_*KeyID_DownloadTime* is a token used to import key materials to KMS.
     - **README**_*KeyID_DownloadTime* is a description file recording information such as a CMK's serial number, wrapping algorithm, wrapping key name, token file name, and the expiration time of the token file and wrapping key.

     NOTICE

     The wrapping key and import token expire in 24 hours. If they have expired, download them again.

   - Method 2: Obtain the wrapping key and import token by calling APIs.

     i. Call the **get-parameters-for-import** API to obtain the wrapping key and import token.
        - **public_key**: content of the wrapping key (Base-64 encoding) returned after the API call
        - **import_token**: content of the import token (Base-64 encoding) returned after the API call

        The following example describes how to obtain the wrapping key and import token of a CMK (ID: **43f1ffd7-18fb-4568-9575-602e009b7ee8**; encryption algorithm: **RSAES_OAEP_SHA_256**).

        - Example request

```
{
   "key_id": "43f1ffd7-18fb-4568-9575-602e009b7ee8",
   "wrapping_algorithm":"RSAES_OAEP_SHA_256"
}
```

- Example response

```
{
   "key_id": "43f1ffd7-18fb-4568-9575-602e009b7ee8",
   "public_key":"public key base64 encoded data",
   "import_token":"import token base64 encoded data",
   "expiration_time":1501578672
}
```

ii. Save the wrapping key and convert its format. Only the key material encrypted using the converted wrapping key can be imported to the management console.

1) Copy the content of the wrapping key **public_key**, paste it to a .txt file, and save the file as **PublicKey.b64**.

2) Use OpenSSL to run the following command to perform Base-64 coding on the content of the **PublicKey.b64** file to generate binary data, and save the converted file as **PublicKey.bin**:

openssl enc -d -base64 -A -in PublicKey.b64 -out PublicKey.bin

iii. Save the import token, copy the content of the **import_token** token, paste it to a .txt file, and save the file as **ImportToken.b64**.

2. Use the wrapping key to encrypt the key material.

NOTE

After performing this step, you will obtain either of the following files:

Symmetric key scenario: **EncryptedKeyMaterial.bin** (key material)

Asymmetric key scenario: **EncryptedKeyMaterial.bin** (temporary key material) and **out_rsa_private_key.der** (private key ciphertext)

Method 1: Use the downloaded wrapping key to encrypt key materials on your HSM.

Method 2: Use OpenSSL to generate a key material and use the downloaded wrapping key to encrypt the key material.

NOTE

If you need to run the **openssl pkeyutl** command, ensure your OpenSSL version is 1.0.2 or later.

a. Generate a key material (256-bit symmetric key) and save it as **PlaintextKeyMaterial.bin**.

- If the AES256 symmetric key algorithm is used, run the following command on the client where the OpenSSL tool has been installed:

openssl rand -out *PlaintextKeyMaterial.bin* 32

- If the SA and ECC asymmetric key algorithms are used, run the following command on the client where the OpenSSL tool has been installed:

1) Generate a hexadecimal AES256 key.

openssl rand -out 0xPlaintextKeyMaterial.bin -hex 32

2) Convert the hexadecimal AES256 key to the binary format.

cat 0xPlaintextKeyMaterial.bin | xxd -r -ps > PlaintextKeyMaterial.bin

b. Use the downloaded wrapping key to encrypt the key material and save the encrypted key material as **EncryptedKeyMaterial.bin**.

If the wrapping key was downloaded from the console, replace ***PublicKey.bin*** in the following command with the wrapping key name *wrappingKey_keyID_DownloadTime*.

Table 2-6 Encrypting the generated key material using the downloaded wrapping key

Wrapping Key Algorithm	Key Material Encryption
RSAES_OAEP_SHA_256	**openssl pkeyutl** **-in *PlaintextKeyMaterial.bin*** **-inkey *PublicKey.bin*** **-out *EncryptedKeyMaterial.bin*** **-keyform der** **-pubin -encrypt** **-pkeyopt rsa_padding_mode:oaep -pkeyopt rsa_oaep_md:sha256**

c. (Optional) To import an asymmetric key, generate an asymmetric private key, use the temporary key material (**EncryptedKeyMaterial.bin**) to encrypt the private key, and import the encrypted file as the private key ciphertext.

- Take the RSA4096 algorithm as an example. Perform the following operations:

  1) Generate a private key.

     openssl genrsa -out rsa_private_key.pem 4096

  2) Convert the key to DER format.

     openssl pkcs8 -topk8 -inform PEM -outform DER -in rsa_private_key.pem -out rsa_private_key.der -nocrypt

  3) Use a temporary key material to encrypt the private key.

     openssl enc -id-aes256-wrap-pad -K $(cat 0xPlaintextKeyMaterial.bin) -iv A65959A6 -in rsa_private_key.der -out out_rsa_private_key.der

     NOTE

     By default, the -id-aes256-wrap-pad algorithm is not enabled in OpenSSL. To wrap a key, upgrade OpenSSL to the latest version and patch it first. For details, see **Why Can't I Wrap Asymmetric Keys by Using -id-aes256-wrap-pad in OpenSSL?**.

Step 11 Click **Next**.The **Import Key Material** page is displayed.

Table 2-7 Parameters for importing key materials (for symmetric keys)

Parameter	Description
Key ID	Random ID of a CMK generated during the CMK creation
Key material	Import a key material. For example, use the **EncryptedKeyMaterial.bin** file in **Step 10.2.b**.

Table 2-8 Parameters for importing key materials (for asymmetric keys)

Parameter	Description
Key ID	Random ID of a CMK generated during the CMK creation
Temporary key material	Import a temporary key material. For example, select the **EncryptedKeyMaterial.bin** file in **Step 10.2.b**.
Private key ciphertext	Select private key ciphertext. For example, select the **out_rsa_private_key.der** file in **Step 10.2.c**.

Step 12 Click **Next** to go to the **Import Key Token** step. Configure the parameters as described in **Table 2-9**.

Table 2-9 Parameters for importing a key token

Parameter	Description
Key ID	Random ID of a CMK generated during the CMK creation
Key import token	Select the token downloaded in **Step 10.1**.
Key material expiration mode	• **Key material will never expire**: You use this option to specify that key materials will not expire after import. • **Key material will expire**: You use this option to specify the expiration time of the key materials. By default, key materials expire in 24 hours after import. After the key material expires, the system automatically deletes the key material within 24 hours. Once the key material is deleted, the key cannot be used and its status changes to **Pending import**.

Step 13 Click **OK**. When the **Key imported successfully** message is displayed in the upper right corner, the materials are imported.

NOTICE

Key materials can be successfully imported when they match the corresponding CMK ID and token.

Your imported materials are displayed in the list of CMKs. The default status of an imported CMK is **Enabled**.

----End

2.3.3 Deleting Key Materials

When importing key materials, you can specify their expiration time. After the key material expires, KMS deletes it, and the status of CMK changes to **Pending import**. You can manually delete the key materials as needed. The effect of expiration of the key material is the same as that of manual deletion of the key material.

This section describes how to delete imported key materials on the KMS console.

NOTE

- To re-import a deleted key material, ensure the imported material is the same as the deleted one.
- Data encrypted using a CMK cannot be decrypted if the key material of the CMK was deleted. To decrypt the data, re-import the key material.

Prerequisites

- You have imported key materials for a CMK.
- The material source of the CMK is **External**.
- The CMK status is **Enabled** or **Disabled**.

Constraints

- To re-import a deleted key material, ensure the imported material is the same as the deleted one.
- Data encrypted using a CMK cannot be decrypted if the key material of the CMK was deleted. To decrypt the data, re-import the key material.
- After the deletion, the CMK will become unavailable and its status will change to **Pending import**.
- The key materials of asymmetric keys cannot be directly deleted. To delete them, perform the instructions in **Deleting One or More CMKs**.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 In the row containing the desired CMK, click **Delete Key Material**.

Step 5 In the dialog box that is displayed, click **Yes**. When **Key material deleted successfully** is displayed in the upper right corner, the key materials are successfully deleted.

After the deletion, the CMK will become unavailable and its status changes to **Pending import**.

----End

2.4 Managing CMKs

2.4.1 Viewing a CMK

This section describes how to view the information about the master key on the KMS console, including the key alias, status, ID, and creation time. The status of a CMK can be **Enabled**, **Disabled**, **Pending deletion**, or **Pending import**.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Check the key list.

Table 2-10 Key list parameters

Parameter	Description
Alias	Alias of a CMK
Status	Status of a CMK, which can be one of the following: • **Enabled** The CMK is enabled. • **Disabled** The CMK is disabled. • **Pending deletion** The CMK is scheduled for deletion. • **Pending import** If your CMK does not have materials, its status is **Pending import**.
ID	Random ID of a CMK generated during the CMK creation
Creation Time	Creation time of the CMK

Parameter	Description
Key Algorithm and Usage	Key algorithm selected during key creation and its usage
Origin	Source of key material, which can be one of the following: • **External** The key is imported to the KMS from an external system. • **Key Management Service** The key is a default master key or created in KMS.
Operation	Operations you can perform on the CMK, such as disable, delete, import key material, or cancel deletion.

Step 5 You can click the alias of a CMK to view its details.

NOTE

To change the alias or description of the CMK, click next to the value of **Alias** or **Description**.

- A default master key (the alias suffix of which is **/default**) does not allow alias and description changes.
- The alias and description of a CMK cannot be changed if the CMK is in **Pending deletion** status.

----End

2.4.2 Enabling One or More CMKs

This section describes how to use the KMS console to enable one or more CMKs. Only enabled CMKs can be used to encrypt or decrypt data. A new CMK is in the **Enabled** state by default.

Prerequisites

The CMK you want to enable is in **Disabled** status.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 In the row containing the desired CMK, click **Enable**.

Step 5 In the dialog box that is displayed, click **Yes** to enable the CMK.

NOTE

To enable multiple CMKs at a time, select them and click **Enable** in the upper left corner of the list.

----End

2.4.3 Disabling One or More CMKs

This section describes how to use the KMS console to disable one or more CMKs, thereby protecting data in urgent cases.

After being disabled, a CMK cannot be used to encrypt or decrypt any data. Before using a disabled CMK to encrypt or decrypt data, you must enable it by following instructions in **Enabling One or More CMKs**.

Prerequisites

The CMK you want to disable is in **Enabled** status.

Constraints

- Default master keys created by KMS cannot be disabled.
- A disabled CMK is still billable. It will stop incurring charges if it is deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 In the row containing the desired CMK, click **Disable**.

Step 5 In the dialog box that is displayed, select **I understand the impact of disabling keys** and click **Yes**.

NOTE

To disable multiple CMKs at a time, select them and click **Disable** in the upper left corner of the list.

----End

2.4.4 Deleting One or More CMKs

Before deleting the CMK, confirm that it is not in use and will not be used.

- Check the CMK permission to determine its possible usage scope. For details, see **Querying a Grant**.
- Check audit logs to determine the actual usage.

Prerequisites

- The CMK you want to schedule deletion for is in **Enabled** or **Disabled** status.

Constraints

- A key will not be deleted until its scheduled deletion period expires. You can set the period to a value within the range 7 to 1096 days.

  Before the specified deletion date, you can cancel the deletion if you want to use the CMK. Once the scheduled deletion has taken effect, the CMK will be deleted permanently and you will not be able to decrypt data encrypted by the CMK. Exercise caution when performing this operation.
- Default Master Keys created by KMS cannot be scheduled for deletion.
- A CMK in pending deletion status does not incur charges. If you cancel deletion, the charging resumes from the time when the CMK was scheduled to be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 In the row containing the desired CMK, click **Delete**.

Step 5 In the dialog box that is displayed, enter the number of days after which you want the deletion to take effect.

NOTE

- A key will not be deleted until its scheduled deletion period expires. You can set the period to a value within the range 7 to 1096 days. Before the specified deletion date, you can cancel the deletion if you want to use the CMK.
- A CMK in pending deletion status does not incur charges. If you cancel deletion, the charging resumes from the time when the CMK was scheduled to be deleted.

Step 6 In the dialog box that is displayed, select **I understand the impact of deleting keys** and click **Yes**.

NOTE

To schedule the deletion of multiple CMKs at a time, select them and click **Delete** in the upper left corner of the list.

----End

2.4.5 Canceling the Scheduled Deletion of One or More CMKs

This section describes how to use the KMS console to cancel the scheduled deletion of one or more CMKs prior to deletion execution. After the cancellation, the CMK is in **Disabled** status.

Prerequisites

The CMK for which you want to cancel the scheduled deletion is in **Pending deletion** status.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 In the row containing the desired CMK, click **Cancel Deletion**.

Step 5 In the dialog box that is displayed, click **OK** to cancel the scheduled deletion.

- If a key is created on the KMS console, the status of the key changes to **Disabled** after its scheduled deletion is canceled. For details about how to enable the key, see **Enabling One or More CMKs**.
- If the CMK is created using imported materials, its status becomes **Disabled** after the cancellation. To enable the CMK, see **Enabling One or More CMKs**.
- If the CMK is created using imported materials and no key materials have been imported for it, its status becomes **Pending import** after the cancellation. To use the CMK, perform **Creating CMKs Using Imported Key Materials**.

NOTE

To cancel the deletion of multiple CMKs at a time, select them and click **Cancel Deletion** in the upper left corner of the list.

----End

2.5 Using the Online Tool to Encrypt and Decrypt Small-Size Data

This section describes how to use the online tool to encrypt or decrypt small-size data (4 KB or smaller) on the KMS console.

Prerequisites

The desired CMK is in **Enabled** status.

Encrypting Data

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of the desired CMK to view its details, and go to the online tool for data encryption and decryption.

Step 5 Click **Encrypt**. In the text box on the left, enter the data to be encrypted.

Step 6 Click **Execute**. Ciphertext of the data is displayed in the text box on the right.

NOTE

- Use the current CMK to encrypt the data.
- You can click **Clear** to clear the entered data.
- You can click **Copy to Clipboard** to copy the ciphertext and save it in a local file.

----End

Decrypting Data

Step 1 Log in to the management console.

Step 2 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 3 You can click any CMK in **Enabled** status to go to the encryption and decryption page of the online tool.

Step 4 Click **Decrypt**. In the text box on the left, enter the data to be decrypted.

NOTE

- The tool will identify the original encryption CMK and use it to decrypt the data.
- However, if the CMK has been deleted, the decryption fails.

Step 5 Click **Execute**. Plaintext of the data is displayed in the text box on the right.

NOTE

You can click **Copy to Clipboard** to copy the plaintext and save it in a local file.

----End

2.6 Managing Tags

2.6.1 Adding a Tag

Tags are used to identify CMKs. You can add tags to CMKs so that you can classify CMKs, trace them, and collect their usage status according to the tags.

Constraints

Tags cannot be added to default master keys.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of the desired CMK to view its details.

Step 5 Click **Tags** to go to the tag management page.

Step 6 Click **Add Tag**. In the **Add Tag** dialog box, enter the tag key and tag value. **Table 2-11** describes the parameters.

NOTE

If you want to delete a tag to be added when adding multiple tags, you can click **Delete** in the row where the tag to be added is located to delete the tag.

Table 2-11 Tag parameters

Parameter	Description	Value	Example Value
Tag key	Name of a tag. The same tag (including tag key and tag value) can be used for different CMKs. However, under the same CMK, one tag key can have only one tag value. A maximum of 20 tags can be added for one CMK.	• Mandatory. • Each tag key must be unique under the same CMK. • 36 characters limit. • The following character types are allowed: – Uppercase letters – Lowercase letters – Digits – Special characters, including hyphens (-) and underscores (_)	cost

Parameter	Description	Value	Example Value
Tag value	Value of the tag	• This parameter can be empty. • 43 characters limit. • The following character types are allowed: – Uppercase letters – Lowercase letters – Digits – Special characters, including hyphens (-) and underscores (_)	100

Step 7 Click **OK** to complete.

----End

2.6.2 Searching for a CMK by Tag

This section describes how to search for a CMK by tag in a project on the KMS console.

Prerequisites

Tags have been added.

Constraints

- Multiple tags can be added for at one search. A maximum of 20 tags can be added for one search. If multiple tags are searched for at one time, each CMK in the search result meets the combined search criteria.
- If you want to delete an added tag from the search criteria, click ✕ next to the tag.
- You can click **Reset** to reset the search criteria.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click **Search by Tag** to show the search box.

Step 4 In the search box, enter or select a tag key and a tag value.

Step 5 Click  to add the input to the search criteria, and click **Search**. The list displays the CMKs that meet the search criteria.

NOTE

- Multiple tags can be added for at one search. A maximum of 20 tags can be added for one search. If multiple tags are searched for at one time, each CMK in the search result meets the combined search criteria.
- If you want to delete an added tag from the search criteria, click  next to the tag.
- You can click **Reset** to reset the search criteria.

----End

2.6.3 Modifying Tag Values

This section describes how to modify tag values on the KMS console.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the management console and select a region or project.

Step 3 Click  . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of the desired CMK to view its details.

Step 5 Click **Tags** to go to the tag management page.

Step 6 Click **Edit** of the target tag, and the **Edit Tag** dialog box is displayed.

Step 7 In the **Edit Tag** dialog box, enter a tag value, and click **OK** to complete the editing.

----End

2.6.4 Deleting Tags

This section describes how to delete tags on the KMS console.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the management console and select a region or project.

Step 3 Click  . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of the desired CMK to view its details.

Step 5 Click **Tags** to go to the tag management page.

Step 6 Click **Delete** of the target tag, and the **Delete Tag** dialog box is displayed.

Step 7 In the **Delete Tag** dialog box, click **Yes** to complete the deletion.

----End

2.7 Rotating CMKs

2.7.1 About Key Rotation

Purpose of Key Rotation

Keys that are widely or repeatedly used are insecure. To enhance the security of encryption keys, you are advised to periodically rotate keys and change their key materials.

The purposes of key rotation are:

- To reduce the amount of data encrypted by each key.

  A key will be insecure if it is used to encrypt a huge number of data. The amount of data encrypted a key refers to the total number of bytes or messages encrypted using the key.
- To enhance the capability of responding to security events.

  In your initial system security design, you shall design the key rotation function and use it for routine O&M, so that it will be at hand when an emergency occurs.
- To enhance the data isolation capability.

  The ciphertext data generated before and after key rotation will be isolated. You can identify the impact scope of a security event based on the key involved and take actions accordingly.

Key Rotation Methods

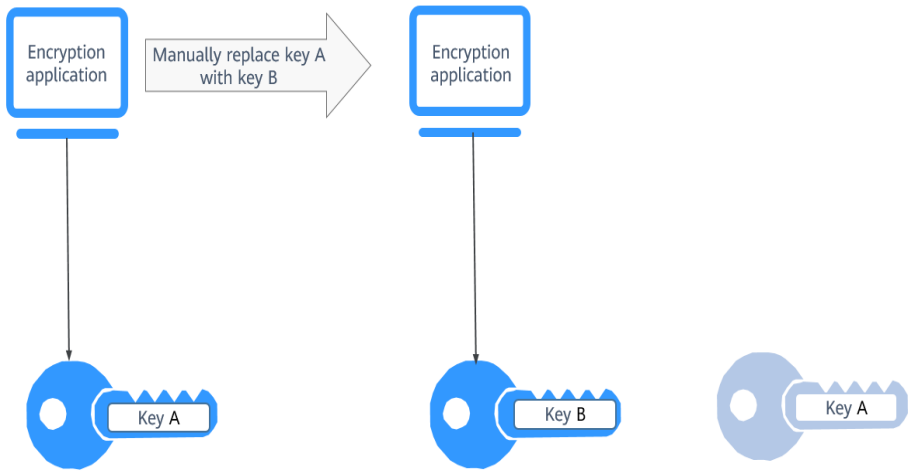
You can use either of the following key rotation methods:

- Manual key rotation

  Replace the key in use with a new key. For example, if key A is in use, you can create key B using a new encryption material, and replace key A with key B. This achieves the same outcome as changing the key material of key A.

  Take OBS as an example. To manually rotate a key, create a new CMK on the KMS console. Replace the old CMK with the new one on the OBS console.

Figure 2-2 Manual key rotation



- Automatic key rotation

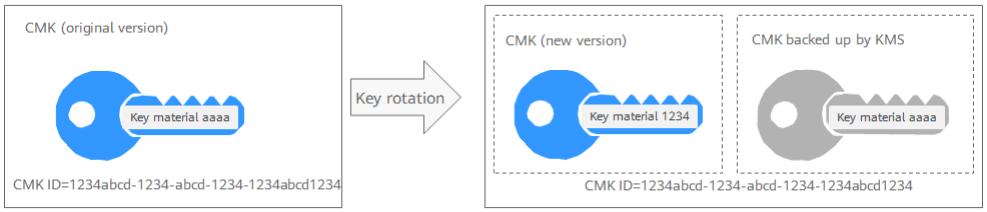
  KMS automatically rotates keys based on the configured rotation period (365 days by default). The system automatically generates a new key to replace the key in use. Automatic key rotation only changes the key material of a CMK. The logical attributes of the CMK will not change, including its key ID, alias, description, and permissions.

  Automatic key rotation has the following characteristics:

  a. Enable rotation for an existing CMK. KMS will automatically generate new key materials for the CMK.

  b. Data is not re-encrypted in an automatic key rotation. The DEK generated using the CMK is not automatically rotated, and data that has been encrypted using the CMK will not be encrypted again. If a DEK has been leaked, automatic rotation cannot contain the impact of the leakage.

Figure 2-3 Key rotation



NOTE

KMS retains all versions of a CMK, so that you can decrypt any ciphertext encrypted using the CMK.

- KMS uses the latest version of the CMK to encrypt data.
- When decrypting data, KMS uses the CMK version that was used to encrypt the data.

Rotation Modes

Table 2-12 Key rotation modes

Key Type	Rotation Mode
Default master key	Cannot be rotated.
User-defined key (imported CMK)	Can only be manually rotated. For more information about user-defined keys, see **CMK Overview**.
Symmetric key	Can be automatically or manually rotated.
Asymmetric key	Can only be manually rotated.
Disabled CMK	Disabled CMKs are not rotated. KMS keeps their rotation status unchanged. After a CMK is enabled, if it has been used for longer than the rotation period, KMS will immediately rotate keys. If the CMK has been used for shorter than the rotation period, KMS will implement the original rotation plan. For more information, see **Disabling One or More CMKs**.
CMKs in pending deletion state	KMS does not rotate CMKs in pending deletion status. After you cancel the deletion of a CMK, the previous key rotation status will be restored. If the CMK has been used for longer than the rotation period, KMS will immediately rotate keys. If the CMK has been used for shorter than the rotation period, KMS will implement the original rotation plan. For more information, see **Scheduling the Deletion of One or More Keys**.

NOTE

You can check the rotation details on the **Rotation Policy** page, including the last rotation time and number of rotations.

2.7.2 Enabling Key Rotation

This section describes how to enable rotation for a CMK on the KMS console.

By default, automatic key rotation is disabled for a CMK. Every time you enable key rotation, KMS automatically rotates CMKs based on the rotation period you set.

Prerequisites

- The CMK is enabled.
- The **Origin** of the CMK is **KMS**.

Constraints

A disabled CMK is never rotated, even if rotation is enabled for it.

KMS resumes rotation when this CMK is enabled. If you enable this CMK after one rotation period has passed, KMS will rotate it within 24 hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of the desired CMK to view its details.

Step 5 Click the **Rotation Policy** tab. The rotation switch is displayed.

Step 6 Click to enable key rotation.

Step 7 In the **Enable Rotation Policy** dialog box, set the rotation period and click **OK**.

- Set the rotation period (unit: day) to an integer in the range 30 to 365. The default value is **365**.
- After the setting takes effect, the new rotation period starts.
- Configure the period based on how often a CMK is used. If it is frequently used, configure a short period; otherwise, set a long one.

  NOTE

  - A disabled CMK is never rotated, even if rotation is enabled for it.
  - KMS resumes rotation when this CMK is enabled. If you enable this CMK after one rotation period has passed, KMS will rotate it within 24 hours.
  - You can click to change the rotation period. After the period is changed, KMS rotates the CMK by the new period.

----End

2.7.3 Disabling Key Rotation

This section describes how to disable rotation for a key on the KMS console.

Prerequisites

- The key is enabled.
- The **Origin** of the CMK is **KMS**.
- Key rotation has been enabled.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of a symmetric key.

Step 5 Click the **Rotation Policy** tab. The rotation switch is displayed.

Step 6 Click to disable key rotation.

Step 7 In the displayed confirmation dialog box, click **OK**.

Step 8 Check the rotation status.

----End

2.8 Managing a Grant

2.8.1 Creating a Grant

You can create grants for other users or accounts to use the CMK. You can create a maximum of 100 grants on a CMK.

Prerequisites

- You have obtained the ID of the grantee (user to whom permissions are to be authorized).
- The desired CMK is in **Enabled** status.

Constraints

The owner of a CMK can create a grant for the CMK on the KMS console or by calling APIs. The users or accounts who have the grant creation permission assigned by the owner of the CMK can create grants for the CMK only by calling APIs.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of the desired CMK to go to the page displaying its details to create a grant on it.

Step 5 Click the **Grants** tab.

Step 6 Click **Create Grant**. The **Create Grant** dialog box is displayed.

Step 7 In the dialog box that is displayed, enter the ID of the user to be authorized and select permissions to be granted. For more information, see **Table 2-13**.

NOTICE

A grantee can perform the authorized operations only by calling the necessary APIs. For details, see the *Key Management Service API Reference*.

Table 2-13 Parameter description

Parameter	Description	Example Value
Key ID	ID of a CMK (automatically read by the system)	-
User or Tenant	Whether a user or an account is authorized. • User User ID: Enter the IAM user ID. To obtain the ID, click the username in the upper right corner of the page, choose **My Credentials**. Choose **API Credentials** from the navigation pane, and copy the value of **IAM User ID**. After the authorization is complete, the IAM user can use the specified keys. • Account Account ID: Enter the IAM user ID. To obtain the ID, click the username in the upper right corner of the page, choose **My Credentials**. Choose **API Credentials** from the navigation pane, and copy the value of **Account ID**. After the authorization is complete, all IAM users under the account can use specified keys.	d9a6b2bdaedd4ba586cabe6372d1b312

Parameter	Description	Example Value
Operations	The following permissions can be authorized: **NOTE** • You can create multiple grants on a CMK to provide different permissions to the same user. The user's permissions on the CMK are the combination of all the grants. • This parameter cannot be left blank. • Selecting only **Create Grant** is not allowed. • **Create Data Key Without Plaintext** • **Create Data Key** • **Encrypt Data Key** • **Decrypt Data Key** • **Query Key Information** • **Create Grant** • **Retire Grant** – A grantee can retire a grant if the grantee does not need that permission. – If, before retiring a grant, the grantee has granted the permission to another user, that user's permission will not be affected by the grant retirement. • **Encrypt Data** • **Decrypt Data**	-

Step 8 Click **OK**. When message **Grant created successfully** is displayed in the upper right corner, the grant has been created.

In the list of grants, you can view the grant ID, grant type, grantee ID, granted operation, and creation time of the grant.

----End

2.8.2 Querying a Grant

This section describes how to view the details about a grant on the KMS console, such as the grant ID, grantee user ID, granted operation, and creation time.

Prerequisites

You have created a grant.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of the desired CMK to view its details.

Step 5 Click the **Grants** tab. Information about the CMK and grants created on it are displayed.

Table 2-14 describes the parameters.

Table 2-14 Parameter description

Parameter	Description
Grant ID	Randomly generated unique identification of a grant
Granted To	Whether permissions are granted to a user or account.
Grantee ID	ID of the authorized user or account.
Granted Operations	Authorized operations (such as **Create Data Key**) on the CMK
Creation Time	Creation time of the grant
Operation	Operations that can be performed on a grant. For example, you can revoke a grant.

Step 6 Click a grant ID to view the grant details.

----End

2.8.3 Revoking a Grant

You can revoke a grant on the KMS console in either of the following scenarios:

- A grantee does not need the grant. (The grantee can either tell the user who has created the grant to revoke the grant or call the necessary API to revoke the grant directly.)
- You do not want the grantee to have the grant.

When a grant is revoked, the grantee does not have the corresponding permission anymore. However, if the grantee has created the same grant to another user, permission of that user will not be affected.

This section describes how to revoke a grant on the KMS console.

Prerequisites

You have created a grant.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of the desired CMK to view its details.

Step 5 In the row of a grantee, click **Revoke Grant**.

Step 6 In the dialog box that is displayed, click **Yes**. When **Grant revoked successfully** is displayed in the upper right corner, the grant has been revoked.

----End

3 FAQs

3.1 KMS Related

3.1.1 What Is Key Management Service?

KMS is a secure, reliable, and easy-to-use cloud service that helps users create, manage, and protect keys in a centralized manner.

It uses Hardware Security Modules (HSMs) to protect keys. All CMKs are protected by root keys in HSMs to avoid key leakage.

It also controls access to keys and records all operations on keys with traceable logs. In addition, it provides use records of all keys, meeting your audit and regulatory compliance requirements.

3.1.2 What Is a Customer Master Key?

A Customer Master Key (CMK) is a Key Encryption Key (KEK) created by a user on KMS. It is used to encrypt and protect DEKs. One CMK can be used to encrypt one or more DEKs.

3.1.3 What Is a Default Master Key?

A Default Master Key is automatically created by another cloud service using KMS, such as Object Storage Service (OBS). The alias of a Default Master Key ends with **/default**.

You can use the management console to query but cannot disable or schedule the deletion of Default Master Keys.

Table 3-1 Default Master Keys

Alias	Cloud Service
obs/default	Object Storage Service (OBS)
evs/default	Elastic Volume Service (EVS)

Alias	Cloud Service
ims/default	Image Management Service (IMS)

NOTE

A Default Master Key is automatically created when a user employs the KMS encryption function for the first time in another cloud service.

3.1.4 What Are the Differences Between a Custom Key and a DMK?

Table 3-2 illustrates the differences between a custom key and a Default Master Key (DMK).

Table 3-2 Differences between a custom key and a DMK

Item	Definition	Difference
Custom key	A Key Encryption Key (KEK) created using KMS. The key is used to encrypt and protect DEKs. A CMK can encrypt multiple DEKs.	Can be disabled and scheduled for deletion.
Default Master Key	Automatically generated by the system when you use KMS to encrypt data in another cloud service for the first time. The suffix of the key is **/default**. Example: **evs/default**	Cannot be disabled or scheduled for deletion.

3.1.5 What Is a Data Encryption Key?

A data encryption key (DEK) is used to encrypt data.

3.1.6 Why Cannot I Delete a CMK Immediately?

The decision to delete a CMK should be considered with great caution. Before deletion, confirm that the CMK's encrypted data has all been migrated. As soon as the CMK is deleted, you will not be able to decrypt data with it. Therefore, KMS offers a user-specified period of 7 to 1096 days for the deletion to finally take effect. On the scheduled day of deletion, the CMK will be permanently deleted. However, prior to the scheduled day, you can still cancel the pending deletion. This is a means of precaution within KMS.

3.1.7 Which Cloud Services Can Use KMS for Encryption?

Object Storage Service (OBS), Elastic Volume Service (EVS), and Image Management Service (IMS) can use KMS for encryption.

Table 3-3 List of cloud services that use KMS encryption

Service Name	Description
Object Storage Service (OBS)	You can upload objects to and download them from Object Storage Service (OBS) in common mode or server-side encryption mode. When you upload objects in encryption mode, data is encrypted at the server side and then securely stored on OBS in ciphertext. When you download encrypted objects, the data in ciphertext is decrypted at the server side and then provided to you in plaintext. OBS supports the server-side encryption with KMS-managed keys (SSE-KMS) mode. In SSE-KMS mode, OBS uses the keys provided by KMS for server-side encryption. For details about how to upload objects to OBS in SSE-KMS mode, see the **Object Storage Service User Guide**.
Elastic Volume Service (EVS)	If you enable the encryption function when creating an EVS disk, the disk will be encrypted with the DEK generated by using your CMK. Data stored in the EVS disk will be automatically encrypted. For details about how to use the encryption function of EVS, see **Elastic Volume Service User Guide**.
Image Management Service (IMS)	When creating a private image using an external image file, you can enable the private image encryption function and select a CMK provided by KMS to encrypt the image. For details about how to use the private image encryption function of Image Management Service (IMS), see **Image Management Service User Guide**.

3.1.8 How Do Cloud Services Use KMS to Encrypt Data?

Services (such as OBS, IMS, EVS, SFS, DDS, and RDS) use the envelope encryption method provided by KMS to protect data.

NOTE

Envelope encryption is an encryption method that enables DEKs to be stored, transmitted, and used in "envelopes" of CMKs. As a result, CMKs do not directly encrypt and decrypt data.

- When you use a cloud service to encrypt data, you need to specify a CMK on KMS. The cloud service generates a plaintext DEK and a ciphertext DEK. The ciphertext DEK is generated by encrypting the plaintext DEK using the specified CMK. The cloud service uses the plaintext DEK to encrypt data and stores the encrypted ciphertext data and ciphertext DEK in the cloud service.

- When users download the data from the cloud, the cloud service uses the CMK specified by KMS to decrypt the ciphertext DEK, use the decrypted DEK to decrypt data, and then provide the decrypted data for users to download.

3.1.9 What Are the Benefits of Envelope Encryption?

Envelope encryption is the practice of encrypting data with a DEK and then encrypting the DEK with a root key that you can fully manage. In this case, CMKs are not required for encryption or decryption.

Benefits:

- Advantages over CMK encryption in KMS

  Users can use CMKs to encrypt and decrypt data on the KMS console or by calling KMS APIs.

  A CMK can encrypt and decrypt data no more than 4 KB. An envelope can encrypt and decrypt larger volumes of data.

  Data encrypted using envelopes does not need to be transferred. Only the DEKs need to be transferred to the KMS server.
- Advantages over encryption by using cloud services
  - Security

    Data transferred to the cloud for encryption is exposed to risks such as interception and phishing.

    During envelope encryption, KMS uses Hardware Security Modules (HSMs) to protect keys. All CMKs are protected by root keys in HSMs to avoid key leakage.
  - Trustworthiness

    You will worry about data security on the cloud. It is also difficult for cloud services to prove that they never misuse or disclose such data.

    If you choose envelope encryption, KMS will control access to keys and record all usages of and operations on keys with traceable logs, meeting your audit and regulatory compliance requirements.
  - Performance and cost

    To encrypt or decrypt data using a cloud service, you have to send the data to the encryption server and receive the processed data. This process seriously affects your service performance and incurs high costs.

    Envelope encryption allows you to generate DEKs online by calling KMS cryptographic algorithm APIs, and to encrypt a large amount of local data with the DEKs.

3.1.10 Is There a Limit on the Number of CMKs That I Can Create on KMS?

Yes.

You can create a maximum of 100 CMKs, including those in enabled, disabled, and pending deletion states. DMKs are not included.

3.1.11 Can I Export a CMK from KMS?

No.

To ensure CMK security, users can only create and use CMKs in KMS.

3.1.12 Can I Decrypt My Data if I Permanently Delete My CMK?

No.

If you have permanently deleted your CMK, the data encrypted using it cannot be decrypted. If the scheduled deletion date of the CMK has not arrived, you can cancel the scheduled deletion.

If the CMK is created using imported key material and only the key material is deleted, you can import the local backup of the key material to the CMK and reclaim the user data. If the key material is not backed up locally, user data cannot be reclaimed.

3.1.13 How Do I Use the Online Tool to Encrypt or Decrypt Small Volumes of Data?

You can use the online tool to encrypt or decrypt data in the following procedures:

Encrypting Data

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the management console and select a region or project.

Step 3 Click . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 4 Click the alias of the desired CMK to view its details, and go to the online tool for data encryption and decryption.

Step 5 Click **Encrypt**. In the text box on the left, enter the data to be encrypted.

Step 6 Click **Execute**. Ciphertext of the data is displayed in the text box on the right.

NOTE

- Use the current CMK to encrypt the data.
- You can click **Clear** to clear the entered data.
- You can click **Copy to Clipboard** to copy the ciphertext and save it in a local file.

----End

Decrypting Data

Step 1 Log in to the management console.

Step 2 Click  . Choose **Security** > **Key Management Service**. The **Key Management Service** page is displayed.

Step 3 You can click any CMK in **Enabled** status to go to the encryption and decryption page of the online tool.

Step 4 Click **Decrypt**. In the text box on the left, enter the data to be decrypted.

NOTE

- The tool will identify the original encryption CMK and use it to decrypt the data.
- However, if the CMK has been deleted, the decryption fails.

Step 5 Click **Execute**. Plaintext of the data is displayed in the text box on the right.

NOTE

You can click **Copy to Clipboard** to copy the plaintext and save it in a local file.

----End

3.1.14 Can I Update CMKs Created by KMS-Generated Key Materials?

No.

Keys created using KMS-generated materials cannot be updated. You can only use KMS to create new CMKs to encrypt and decrypt data.

3.1.15 Why Can't I Wrap Asymmetric Keys by Using -id-aes256-wrap-pad in OpenSSL?

Symptom

By default, the -id-aes256-wrap-pad algorithm is not enabled in OpenSSL. To wrap a key, upgrade OpenSSL to the latest version and patch it first.

Solution

Use bash commands to create a local copy of the existing OpenSSL. You do not need to delete or modify the default OpenSSL client installation configurations.

Step 1 Switch to the **root** user.

sudo su -

Step 2 Run the following command and record the OpenSSL version:

openssl version

Step 3 Run the following commands to create the **/root/build** directory. This directory will be used to store the latest OpenSSL binary file.

mkdir $HOME/build

mkdir -p $HOME/local/ssl

cd $HOME/build

Step 4 Download the latest OpenSSL version from https://www.openssl.org/source/.

Step 5 Download and decompress the binary file.

Step 6 Replace **openssl-1.1.1d.tar.gz** with the latest OpenSSL version downloaded in **step 4**.

curl -O https://www.openssl.org/source/openssl-1.1.1d.tar.gz

tar -zxf openssl-1.1.1d.tar.gz

Step 7 Use the **gcc** tool to patch the version, and compile the downloaded binary file.

yum install patch make gcc -y

NOTE

If you are using a version other than OpenSSL-1.1.1d, you may need to change the directory and commands used, or this patch may not work properly.

Step 8 Run the following commands:

```
sed -i "/BIO_get_cipher_ctx(benc, &ctx);/a\ EVP_CIPHER_CTX_set_flags(ctx,
EVP_CIPHER_CTX_FLAG_WRAP_ALLOW);" $HOME/build/openssl-1.1.1d/apps/enc.c
```

Step 9 Run the following commands to compile the OpenSSL **enc.c** file:

cd $HOME/build/openssl-1.1.1d/

./config --prefix=$HOME/local --openssldir=$HOME/local/ssl

make -j$(grep -c ^processor /proc/cpuinfo)

make install

Step 10 Configure the environment variable **LD_LIBRARY_PATH** to ensure that required libraries are available for OpenSSL. The latest version of OpenSSL has been dynamically linked to the binary file in the **$HOME/local/ssl/lib/** directory, and cannot be directly executed in shell.

Step 11 Create a script named **openssl.sh** to load the **$HOME/local/ssl/lib/** path before running the binary file.

cd $HOME/local/bin/

echo -e '#!/bin/bash \nenv LD_LIBRARY_PATH=$HOME/local/lib/ $HOME/local/bin/openssl "$@"' > ./openssl.sh

Step 12 Run the following command to configure an execute bit on the script:

chmod 755 ./openssl.sh

Step 13 Run the following command to start the patched OpenSSL version:

$HOME/local/bin/openssl.sh

----End

4 Change History

Released On	Description
2023-02-28	This is the first official release.