AWS Key Management Store (KMS)

is a managed service that enables you to easily encrypt your data.

AWS KMS provides a highly available key storage, management, and auditing solution for you to encrypt data within your own applications and control the encryption of stored data across AWS services.

AWS KMS allows you to centrally manage and securely store your keys. These are known as AWS KMS keys (formerly known as customer master keys (CMKs).

## **AWS KMS Keys**

A KMS key consists of:

* Alias.
* Creation date.
* Description.
* Key state.
* Key material (either customer provided or AWS provided).

KMS keys are the primary resources in AWS KMS.

The KMS key includes metadata, such as the key ID, creation date, description, and key state.

The KMS key also contains the key material used to encrypt and decrypt data.

AWS KMS supports symmetric and asymmetric KMS keys.

KMS keys are created in AWS KMS. Symmetric KMS keys and the private keys of asymmetric KMS keys never leave AWS KMS unencrypted.

By default, AWS KMS creates the key material for a KMS key.

A KMS key can encrypt data up to 4KB in size.

A KMS key can generate, encrypt, and decrypt Data Encryption Keys (DEKs).

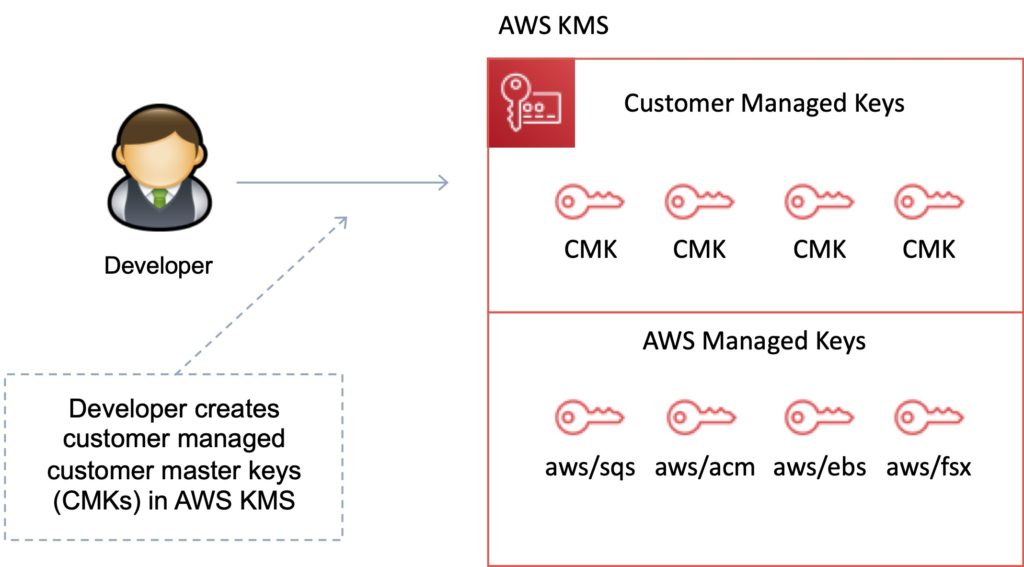
A KMS key can never be exported from KMS (CloudHSM allows this).

AWS Managed KMS keys:

* KMS keys managed by AWS are used by AWS services that interact with KMS to encrypt data.
* They can only be used by the service that created them within a particular region.
* They are created on the first time you implement encryption using that service.

Customer managed KMS keys:

* These provide the ability to implement greater flexibility.
* You can perform rotation, governing access, and key policy configuration.
* You are able to enable and disable the key when it is no longer required.



## **Customer Managed KMS keys**

Customer managed KMS keys are KMS keys in your AWS account that you create, own, and manage.

You have full control over these KMS keys, including establishing and maintaining their key policies, IAM policies, and grants, enabling and disabling them, rotating their cryptographic material, adding tags, creating aliases that refer to the KMS key, and scheduling the KMS keys for deletion.

Customer managed KMS keys incur a monthly fee and a fee for use in excess of the free tier.

## **AWS Managed KMS keys**

AWS managed KMS keys are KMS keys in your account that are created, managed, and used on your behalf by an AWS service that is integrated with AWS KMS.

You cannot manage these KMS keys, rotate them, or change their key policies.

You also cannot use AWS managed KMS keys in cryptographic operations directly; the service that creates them uses them on your behalf.

You do not pay a monthly fee for AWS managed KMS keys. They can be subject to fees for use in excess of the free tier, but some AWS services cover these costs for you.

## **AWS Owned KMS Keys**

AWS owned KMS keys are a collection of KMS keys that an AWS service owns and manages for use in multiple AWS accounts.

Although AWS owned KMS keys are not in your AWS account, an AWS service can use its AWS owned KMS keys to protect the resources in your account.

You do not need to create or manage the AWS owned KMS keys.

However, you cannot view, use, track, or audit them.

You are not charged a monthly fee or usage fee for AWS owned KMS keys and they do not count against the AWS KMS quotas for your account.

## **Data Encryption Keys**

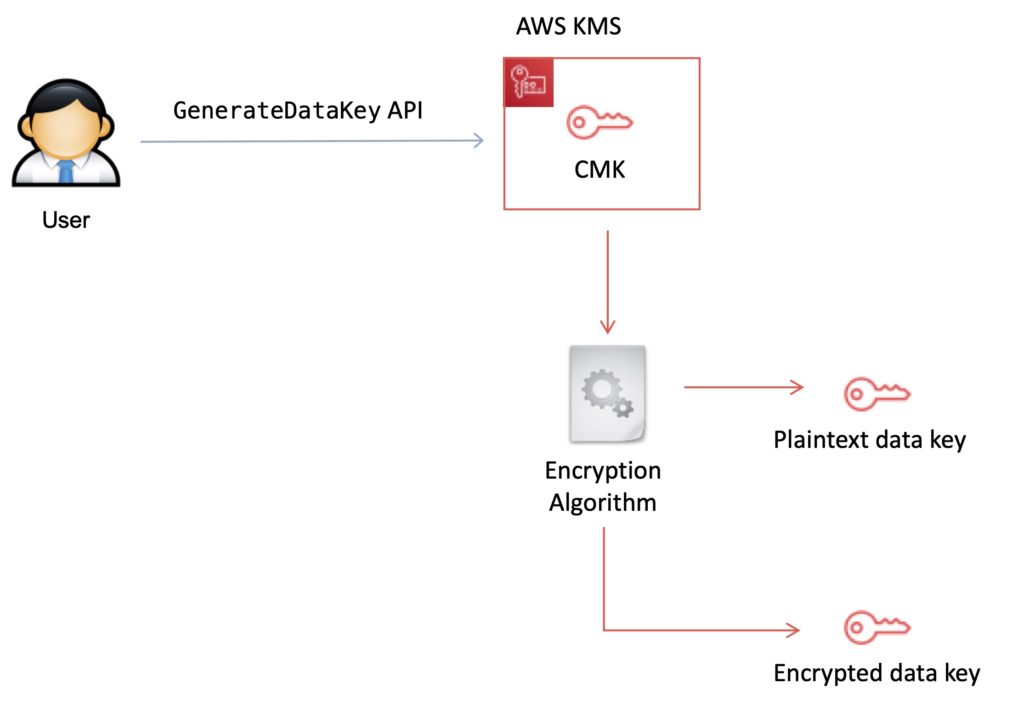
Data keys are encryption keys that you can use to encrypt data, including large amounts of data and other data encryption keys.

You can use AWS KMS keys to generate, encrypt, and decrypt data keys.

AWS KMS does not store, manage, or track your data keys, or perform cryptographic operations with data keys.

You must use and manage data keys outside of AWS KMS.

The GenerateDataKey API can be used to create a data encryption key using a KMS key:



## **KMS Details**

You set usage policies on the keys that determine which users can use them to encrypt and decrypt data and under which conditions.

Key material options:

* KMS generated.
* Import your own.

You can generate KMS keys in KMS, in an AWS CloudHSM cluster, or import them from your own key management infrastructure.

These master keys are protected by hardware security modules (HSMs) and are only ever used within those modules.

You can submit data directly to KMS to be encrypted or decrypted using these master keys.

KMS now has the option for symmetric and asymmetric keys.

KMS is for encryption at rest only (not in transit, use SSL).

KMS is tightly integrated into many AWS services like Lambda, S3, EBS, EFS, DynamoDB, SQS etc.

Data keys are not retained or managed by KMS.

AWS services encrypt your data and store an encrypted copy of the data key along with the data it protects.

When a service needs to decrypt your data they request KMS to decrypt the data key using your master key.

If the user requesting data from the AWS service is authorised to decrypt under your master key policy, the service will receive the decrypted data key from KMS with which it can decrypt your data and return it in plaintext.

All requests to use your master keys are logged in AWS CloudTrail so you can understand who used which key under which context and when they used it.

You can control who manages and accesses keys via IAM users and roles.

You can audit the use of keys via CloudTrail.

KMS differs from Secrets Manager as its purpose-built for encryption key management.

KMS is validated by many compliance schemes (e.g. PCI DSS Level 1, FIPS 140-2 Level 2).

**Exam tip:** Encryption keys are regional.

## **Key Management with KMS**

You can perform the following key management functions in AWS KMS:

* Create keys with a unique alias and description.
* Import your own key material.
* Define which IAM users and roles can manage keys.
* Define which IAM users and roles can use keys to encrypt and decrypt data.
* Choose to have AWS KMS automatically rotate your keys on an annual basis.
* Temporarily disable keys so they cannot be used by anyone.
* Re-enable disabled keys.
* Delete keys that you no longer use.
* Audit use of keys by inspecting logs in AWS CloudTrail.
* Create custom key stores\*.
* Connect and disconnect custom key stores\*.
* Delete custom key stores\*.

\* The use of custom key stores requires CloudHSM resources to be available in your account.

## **Data Encryption Scenarios**

Typically, data is encrypted in one of the following three scenarios:

1. You can use KMS APIs directly to encrypt and decrypt data using your master keys stored in KMS.
2. You can choose to have AWS services encrypt your data using your master keys stored in KMS. In this case data is encrypted using data keys that are protected by your master keys in KMS.
3. You can use the AWS Encryption SDK that is integrated with AWS KMS to perform encryption within your own applications, whether they operate in AWS or not.

## **Custom Key Store**

The AWS KMS custom key store feature combines the controls provided by AWS CloudHSM with the integration and ease of use of AWS KMS.

You can configure your own CloudHSM cluster and authorise KMS to use it as a dedicated key store for your keys rather than the default KMS key store.

When you create keys in KMS you can chose to generate the key material in your CloudHSM cluster. Master keys that are generated in your custom key store never leave the HSMs in the CloudHSM cluster in plaintext and all KMS operations that use those keys are only performed in your HSMs.

In all other respects master keys stored in your custom key store are consistent with other KMS keys.

## **Key Deletion**

You can schedule a customer master key and associated metadata that you created in AWS KMS for deletion, with a configurable waiting period from 7 to 30 days.

This waiting period allows you to verify the impact of deleting a key on your applications and users that depend on it.

The default waiting period is 30 days.

You can cancel key deletion during the waiting period.

## **AWS KMS API’s**

The following APIs are useful to know for the exam:

Encrypt (aws kms encrypt):

* Encrypts plaintext into ciphertext by using a customer master key (KMS key).
* You can encrypt small amounts of arbitrary data, such as a personal identifier or database password, or other sensitive information.
* You can use the Encrypt operation to move encrypted data from one AWS region to another.

Decrypt (aws kms decrypt):

* Decrypts ciphertext that was encrypted by an AWS KMS key using any of the following operations:
  + Encrypt
  + GenerateDataKey
  + GenerateDataKeyPair
  + GenerateDataKeyWithoutPlaintext
  + GenerateDataKeyPairWithoutPlaintext

Re-encrypt (aws kms re-encrypt):

* Decrypts ciphertext and then re-encrypts it entirely within AWS KMS.
* You can use this operation to change the customer master key (KMS key) under which data is encrypted, such as when you manually rotate a KMS key or change the KMS key that protects a ciphertext.
* You can also use it to re-encrypt ciphertext under the same KMS key, such as to change the encryption context of a ciphertext.

Enable-key-rotation:

* Enables automatic rotation of the key material for the specified symmetric customer master key (KMS key).
* You cannot perform this operation on a KMS key in a different AWS account.

GenerateDataKey (aws kms generate-data-key):

* Enables automatic rotation of the key material for the specified symmetric customer master key (KMS key).
* You cannot perform this operation on a KMS key in a different AWS account.

GenerateDataKeyWithoutPlaintext (generate-data-key-without-plaintext):

* Generates a unique symmetric data key.
* This operation returns a data key that is encrypted under a customer master key (KMS key) that you specify.
* To request an asymmetric data key pair, use the GenerateDataKeyPair or GenerateDataKeyPairWithoutPlaintext operations.

## **KMS Envelope Encryption**

AWS KMS is integrated with AWS services and client-side toolkits that use a method known as envelope encryption to encrypt your data.

Under this method, KMS generates data keys which are used to encrypt data and are themselves encrypted using your master keys in KMS:

* A KMS key is used to encrypt the data key (envelope key).
* The envelope key is used to decrypt the data.

## **Limits**

You can create up to 1000 KMS keys per account per region.

As both enabled and disabled KMS keys count towards the limit, AWS recommend deleting disabled keys that you no longer use.

AWS managed master keys created on your behalf for use within supported AWS services do not count against this limit.

There is no limit to the number of data keys that can be derived using a master key and used in your application or by AWS services to encrypt data on your behalf.

### **Related posts:**

* ****