### 

[What is AWS KMS?](#_k5ii5m47q0w)

[What are the different types of encryption in AWS?](#_ko4025xyztc5)

[What are the differences?](#_jop4kd5kl57p)

[Let’s create an AWS KMS Customer-managed key](#_jyqm768yu4xr)

[Let’s use the key on a new S3 bucket](#_gohc73l1wsfz)

[Editing an existing S3 bucket with the same customer-managed key](#_ql5fdkjnaq40)

[Create a new AWS KMS customer-managed key & S3 bucket](#_igj6q9jcl3s5)

### 

### What is AWS KMS?

AWS KMS stands for Amazon Web Services Key Management Service. It is a managed service that enables you to create and control the encryption keys used to encrypt your data in AWS services and in your own applications.

Using KMS, you can create and manage keys that are used to encrypt data at rest in Amazon S3, Amazon EBS, Amazon RDS, and other AWS services. KMS also enables you to control access to your keys, audit key usage, and rotate keys regularly to improve security.

KMS uses industry-standard cryptographic algorithms to secure your data, and the keys used to encrypt your data are never stored in plaintext. Instead, KMS stores encrypted copies of your keys in secure, highly-available infrastructure.

Overall, KMS helps you to manage the encryption of your data by simplifying key creation and management, allowing you to control key usage, and helping you to maintain a strong security posture.

### What are the different types of encryption in AWS?

* Server-Side Encryption (SSE): SSE is a method of encrypting data stored in AWS services such as Amazon S3, Amazon EBS, and Amazon RDS. SSE provides automatic encryption of your data using keys managed by AWS KMS or by the service itself. There are three types of SSE in AWS: SSE-S3, SSE-KMS, and SSE-C.
* Client-Side Encryption: Client-Side Encryption is a method of encrypting data before it is uploaded to AWS services such as Amazon S3. With client-side encryption, you manage the encryption keys yourself and the data is encrypted before it leaves your system. AWS provides a client-side encryption SDK that you can use to implement client-side encryption in your applications.

### What are the differences?

The main difference between encryption with AWS managed keys and customer managed keys is who has control over the encryption keys and the level of control and customization that comes with each option.

AWS Managed Keys are encryption keys that are managed by AWS KMS, which means that AWS manages the keys for you. With AWS managed keys, you don't have to worry about key management tasks such as key generation, key storage, and key rotation. AWS manages these tasks for you, and you can use the keys to encrypt data in a wide variety of AWS services. AWS Managed Keys are designed to provide strong, secure encryption without the need for you to manage your own keys.

Customer Managed Keys, on the other hand, are encryption keys that are managed by you. With customer managed keys, you have full control over the key generation, storage, and rotation processes. You can use your own keys to encrypt data in AWS services that support customer managed keys. The main advantage of customer managed keys is that you have complete control over the keys, which allows you to customize your key management policies to meet your specific security and compliance requirements.

### Let’s create an AWS KMS Customer-managed key

* Log in to the AWS Management Console and navigate to the AWS KMS service.
* Click on "Customer managed keys".
* Click on the "Create key" button to create a new customer managed key.
* Select the appropriate key configuration, such as key type and usage permissions.
* Enter a name for your new key, and optionally add a description to help you identify the key later.
* Choose the key administrators, which are AWS IAM users or roles that are allowed to manage the key.
* Choose the key policy, which defines the permissions for who can use the key.
* Click on "Next" to review your key configuration, and then click on "Finish" to create the key.

### Let’s use the key on a new S3 bucket

* Log in to the AWS Management Console and navigate to the S3 service.
* Click on "Create bucket" to create a new S3 bucket.
* Enter a unique name for your bucket and choose the region where you want to create the bucket.
* Click on "Next" to configure the bucket settings.
* Under "Server-side encryption", select "AWS KMS managed key" as the encryption type.
* Select the customer managed key that you created in AWS KMS from the dropdown list.
* Choose the appropriate encryption options, such as whether to enable encryption for objects in transit or not.
* Click on "Next" to configure the bucket permissions.
* Choose the appropriate access permissions for your bucket, such as who is allowed to access it and what level of access they have.
* Click on "Next" to configure the bucket tags, which are optional metadata that can be used to categorize and track your resources.
* Click on "Review" to review your bucket configuration.
* Review your configuration and make any necessary changes, then click on "Create bucket" to create your new S3 bucket.

### Editing an existing S3 bucket with the same customer-managed key

* Log in to the AWS Management Console and navigate to the S3 service.
* Click on the name of the S3 bucket that you want to edit.
* Click on the "Properties" tab.
* Under "Default encryption", click on "Edit".
* Select "AWS KMS managed key" as the encryption type.
* Select the customer-managed KMS key that you want to use from the dropdown list.
* Choose the appropriate encryption options, such as whether to enable encryption for objects in transit or not.
* Click on "Save changes" to save your new encryption settings.

### Create a new AWS KMS customer-managed key & S3 bucket

| AWSTemplateFormatVersion: '2010-09-09' Resources:  MyKMSKey:  Type: 'AWS::KMS::Key'  Properties:  Description: 'My customer-managed KMS key'  KeyPolicy:  Version: '2012-10-17'  Statement:  - Sid: 'Enable IAM User Permissions'  Effect: 'Allow'  Principal:  AWS: !Sub 'arn:aws:iam::${AWS::AccountId}:root'  Action:  - 'kms:\*'  Resource: '\*'  - Sid: 'Allow Use of the Key'  Effect: 'Allow'  Principal:  AWS: !Sub 'arn:aws:iam::${AWS::AccountId}:user/myuser'  Action:  - 'kms:Encrypt'  - 'kms:Decrypt'  - 'kms:ReEncrypt\*'  - 'kms:GenerateDataKey\*'  Resource: '\*'  MyS3Bucket:  Type: 'AWS::S3::Bucket'  Properties:  BucketName: 'my-s3-bucket' #Change this accordingly  BucketEncryption:  ServerSideEncryptionConfiguration:  - ServerSideEncryptionByDefault:  KMSMasterKeyID: !Ref MyKMSKey  SSEAlgorithm: 'aws:kms' |
| --- |