Search in this guide

English

AWS Documentation AWS KMS Developer Guide

Feedback

Preferences

How Amazon Elastic Block Store (Amazon EBS) uses AWS KMS

PDF

Kindle

RSS

This topic discusses in detail how Amazon Elastic Block Store (Amazon EBS) uses AWS KMS to encrypt volumes and snapshots. For basic instructions about encrypting Amazon EBS volumes, see Amazon EBS Encryption.

Topics

Amazon EBS encryption

When you attach an encrypted Amazon EBS volume to a supported Amazon Elastic Compute Cloud (Amazon EC2) instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. The encryption occurs on the servers that host Amazon EC2 instances.

This feature is supported on all Amazon EBS volume types. You access encrypted volumes the same way you access other volumes; encryption and decryption are handled transparently and they require no additional action from you, your EC2 instance, or your application. Snapshots of encrypted volumes are automatically encrypted, and volumes that are created from encrypted snapshots are also automatically encrypted.

The encryption status of an EBS volume is determined when you create the volume. You cannot change the encryption status of an existing volume. However, you can migrate data between encrypted and unencrypted volumes and apply a new encryption status while copying a snapshot.

Using CMKs and data keys

When you create an encrypted Amazon EBS volume, you specify an AWS KMS customer master key (CMK). By default, Amazon EBS uses the AWS managed CMK for Amazon EBS in your account (`aws/ebs`). However, you can specify a customer managed CMK that you create and manage.

To use a customer managed CMK, you must give Amazon EBS permission to use the CMK on your behalf. For a list of required permissions, see **Permissions for IAM users** in the Amazon EC2 User Guide for Linux Instances or Amazon EC2 User Guide for Windows Instances.

> **Important**
>
> Amazon EBS supports only symmetric CMKs. You cannot use an asymmetric CMK to encrypt an Amazon EBS volume. For help determining whether a CMK is symmetric or asymmetric, see Identifying symmetric and asymmetric CMKs.

Amazon EBS uses the CMK that you specify to generate a unique data key for each volume. It stores an encrypted copy of the data key with the volume. Then, when you attach the volume to an Amazon EC2 instance, Amazon EBS uses the data key to encrypt all disk I/O to the volume.

The following explains how Amazon EBS uses your CMK:

1. When you create an encrypted EBS volume, Amazon EBS sends a GenerateDataKeyWithoutPlaintext request to AWS KMS, specifying the CMK that you chose for EBS volume encryption.

2. AWS KMS generates a new data key, encrypts it under the specified CMK, and then sends the encrypted data key to Amazon EBS to store with the volume metadata.
3. When you attach the encrypted volume to an EC2 instance, Amazon EC2 sends the encrypted data key to AWS KMS with a Decrypt request.
4. AWS KMS decrypts the encrypted data key and then sends the decrypted (plaintext) data key to Amazon EC2.
5. Amazon EC2 uses the plaintext data key in hypervisor memory to encrypt disk I/O to the EBS volume. The plaintext data key persists in memory as long as the EBS volume is attached to the EC2 instance.

Amazon EBS encryption context

In its GenerateDataKeyWithoutPlaintext and Decrypt requests to AWS KMS, Amazon EBS uses an encryption context with a name-value pair that identifies the volume or snapshot in the request. The name in the encryption context does not vary.

An encryption context is a set of key–value pairs that contain arbitrary nonsecret data. When you include an encryption context in a request to encrypt data, AWS KMS cryptographically binds the encryption context to the encrypted data. To decrypt the data, you must pass in the same encryption context.

For all volumes and for encrypted snapshots created with the Amazon EBS CreateSnapshot operation, Amazon EBS uses the volume ID as encryption context value. In the `requestParameters` field of a CloudTrail log entry, the encryption context looks similar to the following:

```
"encryptionContext": {
  "aws:ebs:id": "vol-0cfb133e847d28be9"
}
```

For encrypted snapshots created with the Amazon EC2 CopySnapshot operation, Amazon EBS uses the snapshot ID as encryption context value. In the `requestParameters` field of a CloudTrail log entry, the encryption context looks similar to the following:

```
"encryptionContext": {
  "aws:ebs:id": "snap-069a655b568de654f"
}
```

Detecting Amazon EBS failures

To create an encrypted EBS volume or attach the volume to an EC2 instance, Amazon EBS and the Amazon EC2 infrastructure must be able to use the CMK that you specified for EBS volume encryption. When the CMK is not usable—for example, when its key state is not `Enabled` —the volume creation or volume attachment fails.

In this case, Amazon EBS sends an *event* to Amazon CloudWatch Events to notify you about the failure. With CloudWatch Events, you can establish rules that trigger automatic actions in response to these events. For more information, see Amazon CloudWatch Events for Amazon EBS in the *Amazon EC2 User Guide for Linux Instances*, especially the following sections:

- Invalid Encryption Key on Volume Attach or Reattach
- Invalid Encryption Key on Create Volume

To fix these failures, ensure that the CMK that you specified for EBS volume encryption is enabled. To do this, first view the CMK to determine its current key state (the **Status** column in the AWS Management Console). Then, see the information at one of the following links:

- If the CMK's key state is disabled, enable it.
- If the CMK's key state is pending import, import key material.
- If the CMK's key state is pending deletion, cancel key deletion.

Using AWS CloudFormation to create encrypted Amazon EBS volumes

You can use AWS CloudFormation to create encrypted Amazon EBS volumes. For more information, see AWS::EC2::Volume in the *AWS CloudFormation User Guide*.

Did this page help you?

□ □

Provide feedback

Edit this page on GitHub

Previous topic: Amazon DynamoDB

Next topic: Amazon Elastic Transcoder

Need help?

- Try the forums
- Connect with an AWS IQ expert

Terms of Use
Privacy