

EBS Volume Encryption on AWS

Date published: 2019-12-17

Date modified: 2023-06-27



Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Configuring encryption for Data Hub's EBS volumes on AWS.....	4
Create a Data Hub with encrypted EBS volumes.....	4
Encrypting EBS volumes with a CMK generated from AWS CloudHSM.....	5
Cross-account role permissions.....	5
Create a CloudHSM cluster.....	5
Initialize the CloudHSM cluster.....	6
Activate the CloudHSM cluster.....	8
Create a CMK from CloudHSM.....	8
Encrypting data with a CMK on Data Hubs.....	9

Configuring encryption for Data Hub's EBS volumes on AWS

You can configure encryption for Amazon Elastic Block Store (EBS) volumes used by the Data Hub cluster's VM instances to store data.

Amazon offers the option to encrypt EBS volumes and RDS instances using the default key from Amazon's Key Management System (KMS) or using an external customer-managed KMS (CMK). By default, Data Hubs use the same default key from Amazon's KMS or CMK as the parent environment but you have an option to pass a different CMK during Data Hub creation.

Encryption is configured for block devices and root devices. When encryption is configured for a given cluster, it is automatically applied to all the disk devices of any new VM instances added as a result of cluster scaling or repair.

Environment and Data Hub encryption options

To learn about encryption options that CDP offers for Data Lake, FreeIPA, and Data Hubs, refer to [Environment and Data Hub encryption options](#).

AWS prerequisites for using a CMK

You can use your existing AWS CMK or create a new AWS CMK. For detailed requirements, refer to [AWS Requirements: Customer managed encryption keys](#).

Create a Data Hub with encrypted EBS volumes


EBS encryption can be configured for a Data Hub cluster on the Hardware and Storage page of the advanced create cluster wizard in the CDP web interface.

Before you begin

Note the following:

- If you previously configured encryption on environment level, the CMK that was provided will be used by default for all Data Hubs running in that environment. You can overwrite it for a specific Data Hub by providing another CMK during Data Hub creation. If no CMK was provided during environment registration, a default key from Amazon's KMS is used by default to encrypt Data Hubs running in this environment. See [Adding a customer managed encryption key to a CDP environment running on AWS](#).
- During Data Hub creation, you can specify an encryption key Customer Managed Key (CMK).
- The Encryption configuration option is available per Data Hub host group. The default setting is Encryption: Not encrypted. To enable encryption, perform the following steps in the create cluster wizard for all host groups for which you would like to use encryption.

Procedure

1. Under Instance Type you can see Encryption Supported next to all instance types for which encryption is supported. Ensure that encryption is supported for the instance type that you would like to use.
2. Click on the  icon next to the chosen host group.
3. Enable the Enable Customer-Managed Keys toggle button.
4. Under Select Encryption key, select the CMK that you would like to use.

What to do next

Once the cluster is running, you can confirm that encryption is enabled by navigating to the details of the block devices or root devices in the EC2 console on AWS. The device should be marked as “Encrypted” and the “KMS Key ARN” is listed.

Encrypting EBS volumes with a CMK generated from AWS CloudHSM

This section provides steps for encrypting a Data Hub with a CMK generated from a custom key store on AWS CloudHSM.

AWS CloudHSM is a cloud-based hardware security module (HSM) that enables you to easily generate and use their own encryption keys in AWS. CloudHSM runs in your VPC and provides you with the flexibility to integrate with many applications.

When using a custom key store, the default KMS key store is replaced by a dedicated key store, which is hosted on a single tenant CloudHSM cluster on AWS. When a user generates a CMK (Customer Managed Key), the key material is generated and stored in the CloudHSM and KMS forwards all the data encryption requests to the CloudHSM.

For more general information about CloudHSM, see [AWS Security Blog](#).

To set this up, you need to create, initialize, and activate the HSM cluster, and then create a CMK from HSM. Once done, you can use the CMK for encrypting Data Hubs in CDP. Detailed steps are provided below.

Cross-account role permissions

If planning to use encryption, ensure that the cross-account IAM role used for the provisioning credential includes the permissions mentioned in [Permissions for using encryption](#).

Create a CloudHSM cluster

First, create a CloudHSM cluster.

Procedure

1. Log in to your AWS account and navigate to [CloudHSM](#).
2. In the top-right corner, verify that you are in the region where you would like to create the CloudHSM cluster.
3. Click on Create Cluster in the top corner.
4. On the “Cluster configuration” page:
 - a. Select the VPC in which you wish to create the cluster. This should be the VPC where your CDP environment should be running.
 - b. Under AZ(s), select the private subnets in the selected VPC (or create new private subnets If subnets are not present). Make sure that each subnet exists in a different availability zone.
5. Click Next.
6. On the “Backup retention” page, select a desired backup retention period (in days).
7. Click Next.
8. On the “Tags” page, add tags if required by your organization.
9. Review the cluster configuration and click on Create Cluster.

Results

This creates a cluster in an uninitialized state. You need to initialize the cluster before you can use it.



Note: When you create a cluster, AWS CloudHSM creates a security group with the name `cloudhsm-cluster-clusterID-sg`. This security group contains a preconfigured TCP rule that allows inbound and outbound communication within the cluster security group over ports 2223-2225. This rule allows HSMs in your cluster to communicate with each other.

Initialize the CloudHSM cluster

Now that you have created the CloudHSM cluster, you should initialize it.

Procedure

1. In the cluster created above, click on the Initialize button.
2. Select an Availability Zone (AZ) for the HSM and then choose Create.
3. Once the CSR is ready, you get a link to download the CSR. Click on Cluster CSR to download and save it.

Certificate signing request

To initialize the cluster, you must download a certificate signing request (CSR) and then [sign it](#).

 Cluster CSR

Cluster verification certificate

Optionally, you may wish to download the HSM certificate below which generated this Cluster CSR and [verify its authenticity](#).

 HSM certificate

Leave the browser tab open, as you will need to continue the steps later.

4. At this point, you need to sign the CSR to initialize the cluster. To sign the CSR the below steps are required:
 - a. Create a private key.

You can create a private key using OpenSSL for development and testing:

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

For example:

```
$ openssl genrsa -aes256 -out customerCA.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
```

```
e is 65537 (0x10001)
Enter pass phrase for customerCA.key:
Verifying - Enter pass phrase for customerCA.key:
```

For a production scenario AWS recommends creating the private key in a more secure fashion using an offline HSM or equivalent.

- b.** Use the private key to create a signing certificate.

You can use OpenSSL to sign the certificate. After running the command, you will be prompted to answer a few questions. The certificate will be valid for 10 years:

```
openssl req -new -x509 -days 3652 -key customerCA.key -out customerC
A.crt
```

This command creates a certificate file named customerCA.crt.

For example:

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

- c.** Put the certificate generated in the previous step on every host from which you are planning to connect to your AWS CloudHSM cluster.

If you rename the file or store it in a path other than the root of your host, you should edit your client configuration file accordingly.

- d.** Sign your cluster CSR using the CSR downloaded earlier and download your HSM certificate.

The following example uses OpenSSL to sign the cluster's CSR. The example uses your private key and the self-signed certificate that you created in the previous step:

```
openssl x509 -req -days 3652 -in <cluster ID>_ClusterCsr.csr \
  -CA customerCA.crt \
  -CAkey customerCA.key \
  -CAcreateserial \
  -out <cluster ID>_CustomerHsmCertificate.crt
```

Example command and output:

```
$ openssl x509 -req -days 3652 -in <cluster ID>_ClusterCsr.csr \
  -CA customerCA.crt \
  -CAkey customerCA.key \
  -CAcreateserial \
  -out <cluster ID>_CustomerHsmCertificate.crt
Signature ok
```

```
subject=/C=US/ST=CA/O=Cavium/OU=N3FIPS/L=SanJose/CN=HSM:<HSM identifier>:PARTN:<partition number>, for FIPS mode
Getting CA Private Key
Enter pass phrase for customerCA.key:
```

This command creates a file named <cluster ID>_CustomerHsmCertificate.crt. You will use this file as the signed certificate when you initialize the cluster.

5. Navigate back to your browser window and initialize the cluster. You will need to use the signed HSM certificate and also the actual certificate to initialize the cluster. Click Next on the “Sign certificate Signing request” page.
6. On the upload certificates page, do the following:
 - a. Next to “Cluster certificate”, click on Upload file. Next, locate and select the HSM certificate. The file should be called <cluster ID>_CustomerHsmCertificate.crt.
 - b. Next to “Issuing certificate”, choose Upload file. Then select your signing certificate. If you completed the steps in the previous section, the file should be called customerCA.crt.
 - c. Choose Upload and initialize.

Activate the CloudHSM cluster

Once you have initialized the cluster, you need to activate it. Activating the cluster can be done through the command line using an Amazon EC2 instance with a CloudHSM client installed on it.

Before you begin

Install the CloudHSM client on an Amazon EC2 instance. The instructions to install the CloudHSM client on an Amazon EC2 instance are documented well in [Install and Configure the AWS CloudHSM Client \(Linux\)](#). Make sure that the EC2 instance is on the same VPC that our cluster is created on.

Procedure

1. Attach the EC2 instance to the security group that was created as a result of creating the HSM cluster.
2. To activate the CloudHSM cluster, you need to start the cloudhsm_mgmt_util through a command line utility, which will connect to the new HSM node and automatically enter the cloudHSM CLI. Follow the steps in [Activate the Cluster](#) (AWS docs).

What to do next

Once the Cloud HSM is active, you can start creating CMK’s from the custom key store.

Create a CMK from CloudHSM

Now that you set up CloudHSM cluster, you can create a CMK from CloudHSM.

Procedure

1. From the AWS console, navigate to the [Key Management Service \(KMS\)](#).
2. In the top-right corner, verify that you are in the region where you would like to create the CMK.
3. Select Customer Managed Keys (CMK) from the KMS page.
4. Click on Create Key in the top left corner.

5. In the “Configure key” wizard, select Symmetric and then in the “Advanced options” select the Custom key store (CloudHSM):

The screenshot shows the 'Configure key' wizard interface. It has two main sections: 'Key type' and 'Advanced options'. In the 'Key type' section, 'Symmetric' is selected with a radio button, and 'Asymmetric' is unselected. Below 'Key type' is the 'Advanced options' section, which is expanded. Under 'Key material origin', 'Custom key store (CloudHSM)' is selected with a radio button, while 'KMS' and 'External' are unselected. At the bottom right of the wizard are 'Cancel' and 'Next' buttons.

6. Click Next.
7. Add an alias to the key.
8. Click Next.
9. Provide required permissions to the key to encrypt and decrypt the data.
10. Click Next and then finish creating the key.

Results

Once you've performed these steps, you are ready to create Data Hubs with the CMK that you just created.

Encrypting data with a CMK on Data Hubs

Use these steps to set up encryption for your Data Hub cluster data with the CMK created earlier.

Procedure

1. Create an environment in the same VPC where the CloudHSM is hosted. Steps to register a new environment in AWS cloud are documented in [Register an AWS environment](#).
2. Once the environment is running, you can create Data Hubs within the environment as described in [Create a Data Hub Cluster on AWS](#). Make sure to select the following in the Advanced options of the cluster creation wizard:
 - a. Under Hardware and Storage, make sure to select the instance types that support encryption.
 - b. Under Encryption Key, select the CMK that you have created earlier. This CMK will be used to encrypt your data.
 - c. Repeat the process for all host groups where you would like to encrypt data.
3. Finish creating the Data Hub.

Results

Once the Data Hub is running, the specified CMK is used to encrypt data stored in the cluster.