Amazon EBS Encryption Support in Avi Vantage

Avi Technical Reference (v20.1)

Amazon EBS Encryption Support in Avi Vantage

view online

Overview

Amazon EBS encryption is a solution offered to encrypt EBS volumes. Encrypting EBS volumes and attaching it to the supported instance type encrypts the data inside the volume, all data moving between the volume and the instance, all snapshots created from the volume, and all volumes created from those snapshots.

The data at rest within an Amazon S3 data center can be protected using AWS KMS. Server-side encryption is one way to use AMS KMS, in which you can protect your data using the customer master key. The three different modes of service-side encryption are: * SSE-S3 ? Amazon S3 manages the data and master encryption keys. * SSE-C ? User manages the encryption key. * SSE-KMS ? AWS manages the data key, but the user manages the master key in AWS KMS. For complete information on AWS KMS, refer to How Amazon Simple Storage Service (Amazon S3) Uses AWS KMS.

Starting release 17.2.3, Avi Vantage supports enabling EBS and S3 encryption using AWS SSE-KMS which encrypts the Amazon Machine Image (AMI). For more information on AMI, refer to Amazon Machine Images (AMI).

Configuring AWS Encryption for Avi Vantage

On deploying the Avi Controller instance in the AWS cloud, an Amazon Machine Image (AMI) is generated and uploaded to an Amazon Simple Storage Service (S3) bucket within the account. This Controller AMI is used to deploy the Service Engines as required.

Enabling encryption, encrypts both the Controller and Service Engine AMIs. As explained earlier, this encryption is done for the EBS volume and S3 bucket. Enabling encryption does not dynamically update any existing Service Engines and is applied only to the newly launched Service Engines.

CLI Configuration

To enable the encryption using CLI, enter the Controller bash and enter the following options under the cloud configuration mode:

```
configure cloud <i>aws_cloud</i>
    aws_configuration
        s3_encryption [<b>mode</b> | <b>key</b>]
        ebs_encryption [<b>mode</b> | <b>key</b>]
```

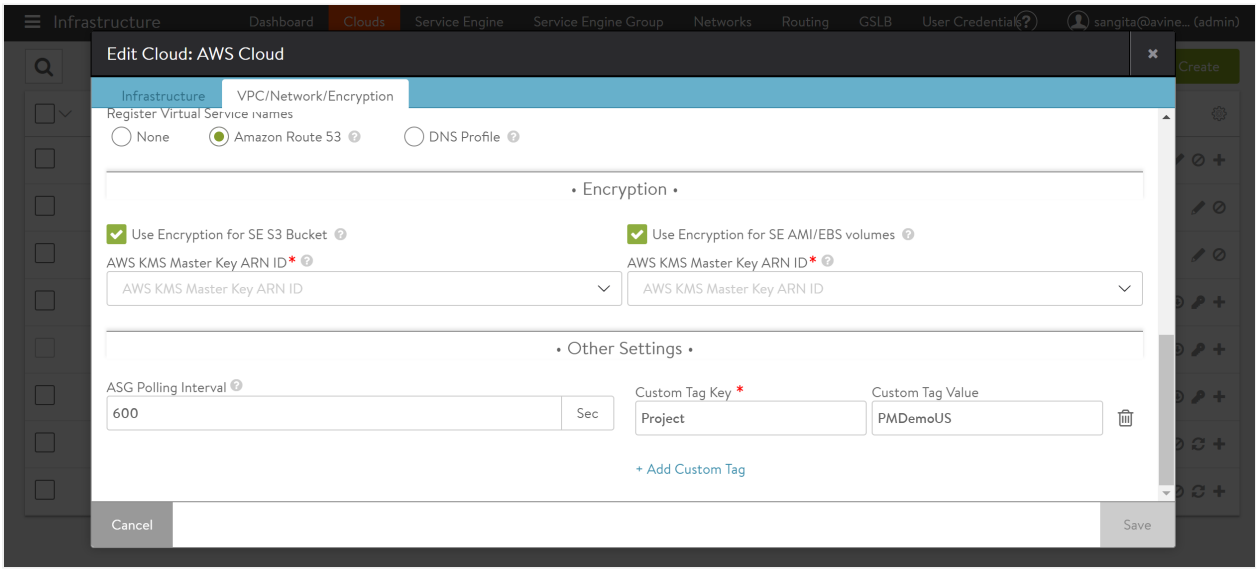
- Entering the mode keyword enables the SSE KMS mode of AWS encryption mode.
- Entering the key keyword allows you to enter the AWS KMS ARN ID of the master key for encryption.

UI Configuration

To enable the encryption on UI, navigate to Infrastructure > Clouds, and select the AWS cloud to enable encryption for. Click on the edit icon.

In the AWS User Credentials section, enable the checkbox for Use Encryption for SE S3 Bucket to encrypt S3 bucket used during the Service Engine image upload and enable the checkbox for Use Encryption for SE AMI/EBS volumes to encrypt Service Engine AMI, snapshot, or volume.

Select *SSE KMS* from the dropdown list for Encryption Mode. For the AWS KMS Master Key ARN ID field, choose one to the relevant options: 1. If the given credentials or Controller role has sufficient permissions to read the list of the keys, then they will be displayed in a dropdown. Choose the displayed option. 2. The key ARN can be entered manually in the Customer Master Key (CMK) format `arn:aws:kms:AWS-Region:AWS-Account-ID:key/CMK-key-ID`.
If left blank, the default KMS CMK of the service will be used.



Note:
1. Most instance types are supported for EBS encryption. For complete information, refer to Amazon EBS Encryption. 2. The S3 bucket encryption feature requires VMimport.

As a part of cloud orchestration, Avi Controller will upload and manage either an unencrypted or encrypted Service Engine AMI based on the Use Encryption for SE AMI/EBS volumes option.

Operator Errors on KMS CMK deletion

- If the KMS CMK for EBS is deleted in AWS, then
  - The existing running Controllers and Service Engines will continue to function.
  - Starting or restarting the Controller or Service Engine may not work as expected.
  - New Service Engine virtual machines cannot be launched.
  - The current encrypted AMI will not be effective and will be manually deleted.
  - The cloud configuration needs to be updates to provide a new KMS CMK.
- If the KMS CMK for S3 is deleted in AWS during an Avi Service Engine image upload, then a new Service Engine upload will fail.

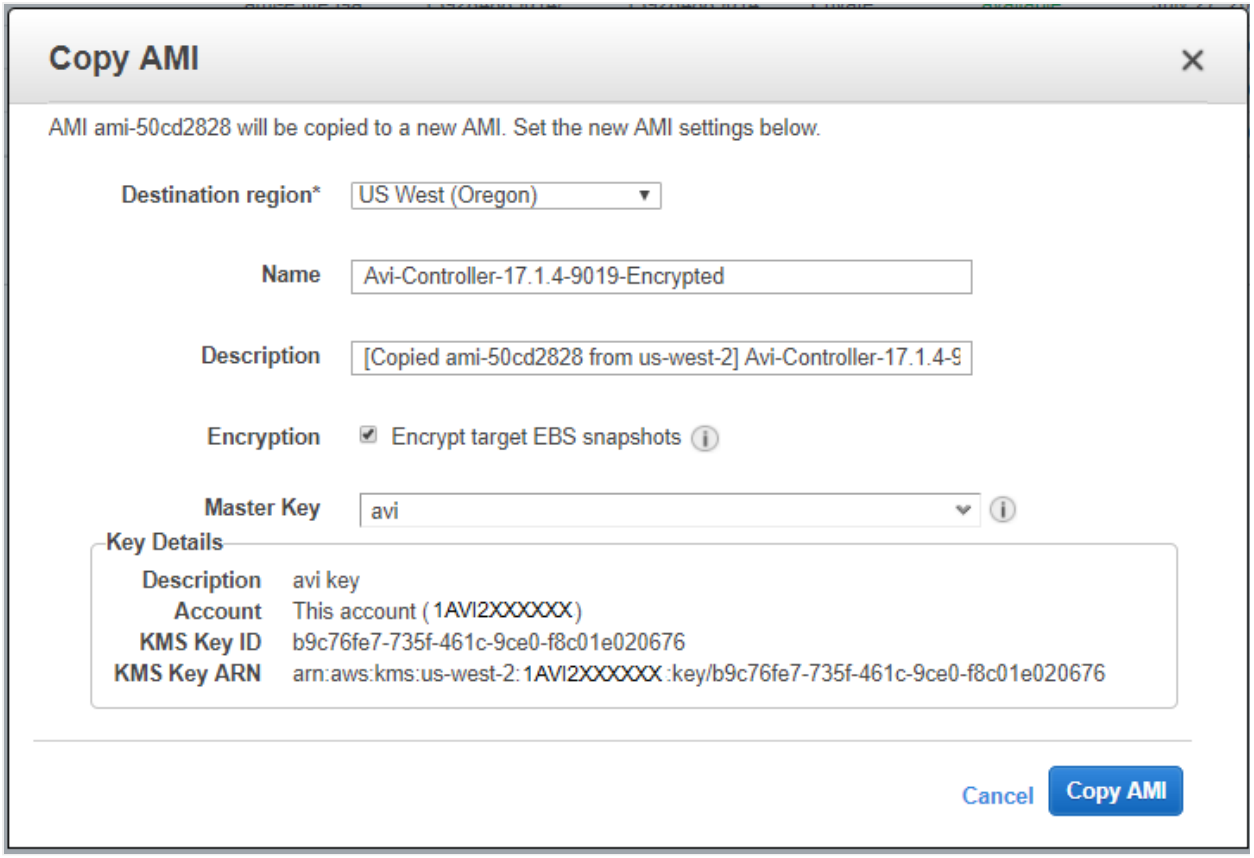
Using EBS Encryption for Controller AMI

This section discusses creating your own copy of the encrypted Avi Controller AMI.

The Avi Controller's AMI is generally unencrypted. This AMI is either publicly available or explicitly shared.

1. On EC2 portal, you can view the AMI by switching to *public images* or *private images* filter in the AMI tab.

2. Create an encrypted copy of the AMI in the destination region using the preferred master key.



3. The AMI will appear with the encrypted snapshot. The Snapshots tab will show the details of the KMS key used.

```
 AMI: 1AVI2XXXXXX/Avi-Controller-17.1.4-9019
Block Devices: /dev/sda1=snap-091b3e473091b8a0d:64:true:standard:encrypted

Snapshot: snap-091b3e473091b8a0d
Encrypted: Encrypted
KMS Key ID: b9c76fe7-735f-461c-9ce0-f8c01e020676
```

Notes:
1. Copied snapshots (encrypted or unencrypted) will have a volume ID of vol-ffffffff.
2. AMIs with encrypted snapshots cannot be shared publicly or with other accounts. 3. As per the recent changes on AWS, an encrypted Amazon Elastic Block Store (EBS) backed Amazon Elastic Compute Cloud (EC2) instance can now be launched from any unencrypted Amazon Machine Image (AMI), such as an AWS community or marketplace AMI with a single API call. For more information, refer to Launch encrypted EBS backed EC2 instances from unencrypted AMIs in a single step.

Migrating from Unencrypted to Encrypted AMI

The AMIs for the Avi Controller and Service Engines can be migrated from unencrypted to encrypted. You can either opt for a migration activity with downtime or without downtime. This section discusses the detailed steps required for migrating from unencrypted to encrypted AMI.

Controller 3-node cluster using New Controller VMs

Use this approach if you have a 3-node cluster and do not need to preserve the Controller private IP address. This approach involves no downtime and the nodes are migrated one at a time starting with the cluster followers.

1. From the AMIs tab, launch the Controller cluster using the encrypted AMI.
2. The virtual machines root device volume are encrypted. In the Volumes tab, the volumes will show the details of the Snapshot and the KMS key used.

   {%cli%}Volume ID: Vol-0de3660075df2bb90 Snapshot: snap-091b3e473091b8a0d Encrypted: Encrypted KMS Key ID: b9c76fe7-735f-461c-9ce0-f8c01e020676 {%endcli%}
3. Modify cluster membership to remove one existing unencrypted node and add this newly created encrypted node.
4. Wait for cluster READY state and then repeat the steps for other nodes.
5. Re-associate any elastic-IPs that were associated directly with the old VMs. Cluster-VIP and elastic-IP association to cluster-VIP are handled by Avi Controller orchestration.
6. After all cluster nodes have been replaced, stop or terminate the old unencrypted controller VMs.

Controller 1-node or 3-node cluster using Existing Controller VMs

Use this approach to preserve Controller private IP and migrate nodes of the cluster one at a time starting with the cluster followers. Account for the downtime involved in this approach.

1. In the Snapshots tab, locate the snapshot of the encrypted AMI and create a volume from it in the AZ of the Controller VM. The new volume will be encrypted as well.

   {%cli%}Volume ID: Vol-0f261557a638f5628 Snapshot: snap-091b3e473091b8a0d Encrypted: Encrypted KMS Key ID: b9c76fe7-735f-461c-9ce0-f8c01e020676 {%endcli%}
2. For a 1-node cluster, backup the current Controller's configuration. Refer to Backup and Restore of Avi Vantage Configuration for backup instructions.
3. In the Instances tab, shutdown the Avi Controller and note the Block devices value, for instance, */dev/sda1*.
4. In the Volumes tab, locate the volume associated with the Controller VM and detach from it.
5. In the Volumes tab, locate the newly created volume and associate it with the Controller VM. Provide the same device name as noted for the old volume, as in Step 2.
6. From the Instances tab, start the Avi Controller.
7. For a 1-node cluster, restore the configuration on the new Controller. Refer to Backup and Restore of Avi Vantage Configuration for configuration restore instructions.
8. For a 3-node cluster, edit and save the cluster configuration. Wait for the modified Controller to re-join the cluster and attain the cluster READY state. Repeat the steps for other nodes.
9. After all controller nodes' volume have been replaced, delete the old unencrypted snapshots and volumes.?

Service Engines with Zero Downtime

This approach involves scaling out the existing virtual services, so that the new Service Engine with encrypted AMI are spun out and then it is scaled back in.

1. Ensure that the SE-Group and AWS account has sufficient capacity (for VMs and management-IPs) to launch more SE VMs.
2. Enable the option Disable-for-VS-Placement for all Service Engines in the group. Make a note of the current unencrypted list.
3. Disable one of the Service Engine. This will launch a new SE VM using the encrypted SE AMI and will migrate all the existing virtual service VIPs to the new Service Engine.
4. The Service Engine dashboard will display the value of active virtual services under # Virtual Services. Once all the virtual service VIPs have migrated away from the disabled SEs, repeat Step 3 for other unencrypted Service Engines.
5. When all old unencrypted Service Engines have been disabled and show zero virtual service count, delete all the old Service Engines, that are in the disabled state.

Service Engines with Downtime

This approach involves detaching just the EBS volume component, encrypting it, and placing it back on the Service Engine. To migrate a Service Engine, perform the following steps for every SE group:

1. Using Service Engine dashboard, delete all the existing Service Engines in the SE groups.
2. The new Service Engines will be launched using the new encrypted SE AMI.