How to Encrypt an EBS Volume

Keeping data and applications safe in the cloud is one the most visible challenges facing cloud teams in 2018. Cloud storage services where data resides are frequently a target for hackers, not because the services are inherently weak, but because they are often improperly configured.

Encryption—to protect data at rest and in-flight—should be an organization's number-one priority when using any storage service. On AWS, its Elastic Block Store (EBS) service provides persistent block-level storage volumes for Amazon EC2 instances. EBS volumes can be attached to your instances and primarily used for data that is rapidly changing or that requires specific input/output operations per second, (IOPS). Because they provide persistent level storage to your instances, EBS volumes are ideally suited for retaining important data and can be used to store personally identifiable information (PII). In any environment where this is the case, it's essential that data on the volume is encrypted to protect it from malicious activity.

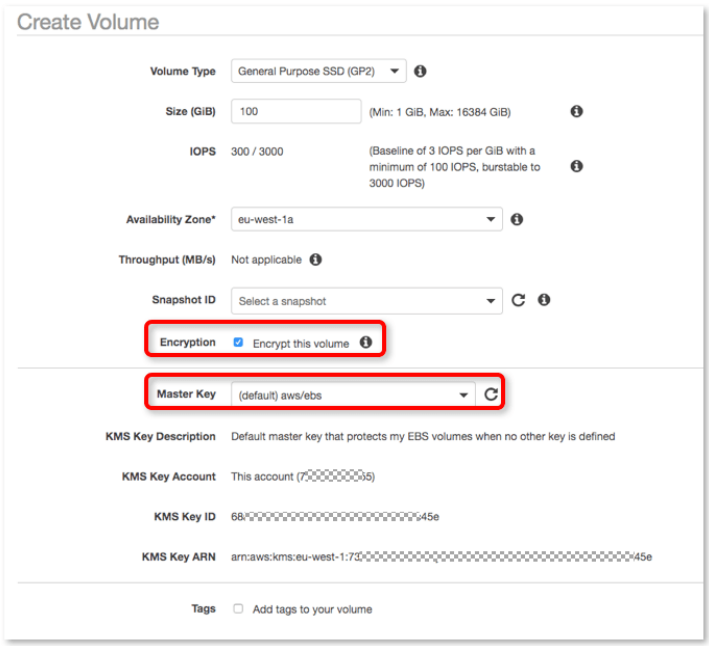
AWS makes encryption easy. In this post, we'll step through three processes using AWS Elastic Block Store to help you make sure that your encryption is configured correctly.

How to Encrypt an EBS Volume

With the EBS encryption mechanism, you don't have to worry about managing keys to perform encryption yourself—it's all managed and implemented by EBS. However, there are a couple of different ways that encryption can be applied depending on how and when you are creating your new EBS volumes. I will walk you through each process.

Encryption for a new EBS Volume

1. From within the AWS Management Console, select EC2
2. Under 'Elastic Block Store' select 'Volumes'
3. Select 'Create Volume'
4. Enter the required configuration for your Volume
5. Select the checkbox for 'Encrypt this volume'
6. Select the KMS Customer Master Key (CMK) to be used under 'Master Key'
7. Select 'Create Volume'

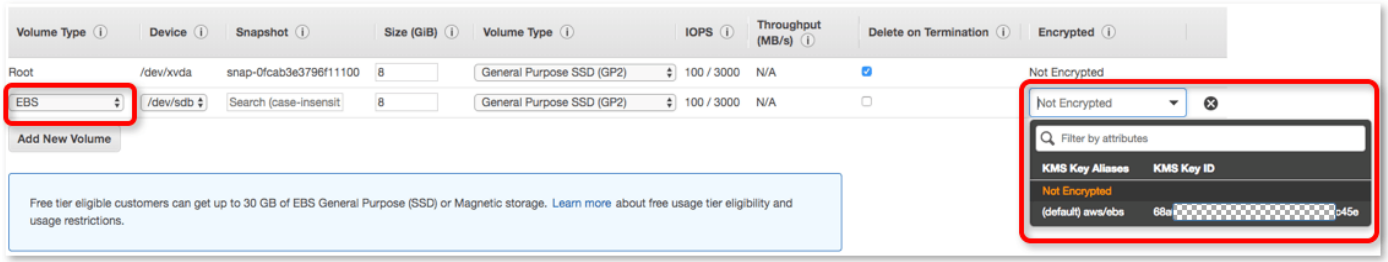


How to Encrypt an EBS Volume

Once your volume has been created, all data saved to this volume will be encrypted when attached to an EC2 instance.

Encrypt a new EBS volume when launching an EC2 instance

1. From within the AWS Management Console, select EC2
2. Select 'Launch Instance'
3. Select your AMI type
4. Select your Instance Type
5. Click 'Next: Configure Instance Details'
6. Configure your instance as required
7. Select 'Next: Add Storage'
8. Select 'Add New Volume'
9. Ensure your volume type is 'EBS' and configure your storage requirements
10. Select the drop-down list under 'Encryption' and select the KMS CMK key to be used
11. Continue with your EC2 instance launch process



The EBS volume attached to that instance will now be encrypted. It's also worth noting that any snapshots created from these encrypted volumes (and any volumes created from these snapshots) will also be encrypted.

EBS with KMS

The EBS service interacts with another AWS service, the Key Management Service (KMS) to perform encryption. KMS uses Customer Master Keys (CMK) to create Data Encryption Keys (DEK), which enables data encryption across EBS and a range of AWS services.

When a volume is defined as an encrypted volume, EBS sends a request to KMS asking for a Data Encryption Key. The DEK is generated AND encrypted by the Customer Master Key, which by default will be a unique, regional CMK provided by AWS unless otherwise specified. The encrypted DEK is then stored with the metadata on the EBS volume.

It's important to point out that no data has been encrypted up to this point. So far only the Data Encryption Key has been encrypted. The data encryption process is driven from the EC2 instance, not the EBS volume, so your data will be encrypted when it is connected to an associated EC2 instance.

When the volume is attached to an EC2 instance, the instance sends a 'decrypt' request to KMS along with the encrypted DEK from the EBS volume. KMS then responds with a plaintext version of the DEK. EC2 will store this within its hypervisor memory, allowing the instance itself to perform encryption on any read/writes to the EBS volume using the plaintext version of the DEK.

The DEK uses the AES-256 (Advanced Encryption Standard – 256 bit) algorithm to encrypt any data written. The process is managed from the EC2 instance, which ensures that the data is also encrypted when in transit to the EBS volume.

How to Encrypt an EBS Volume

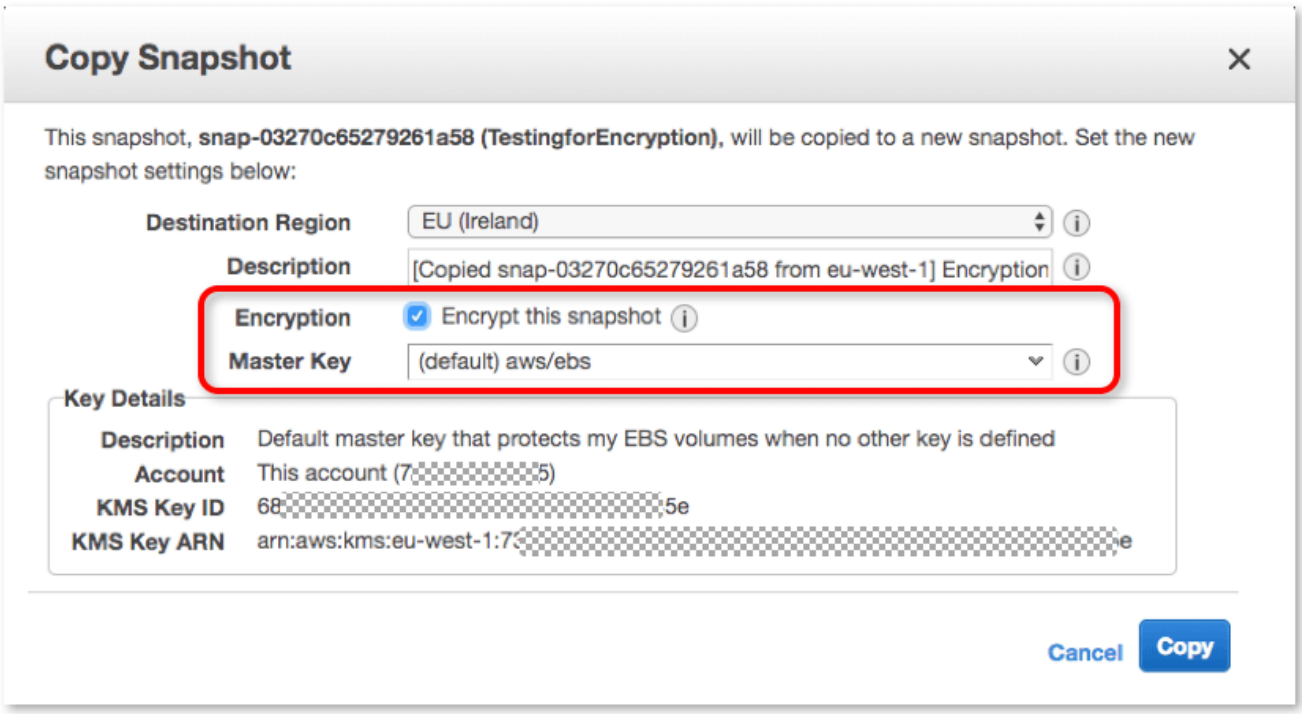
Let's look at the full process:

1. A volume is defined as 'encrypted' in EBS
2. EBS calls KMS to request a Data Encryption Key
3. KMS generates a DEK from the specified Customer Master Key
4. The CMK encrypts the DEK
5. The DEK is then stored on the encrypted EBS volume as metadata
6. The EBS volume is then attached to an EC2 instance
7. EC2 sends a 'decrypt' request to KMS with the encrypted DEK from the volume
8. KMS decrypts the DEK into a plaintext DEK and sends it back to the EC2 instance
9. EC2 stores the plaintext DEK in its hypervisor memory for as long as the EBS volume is attached to the instance
10. EC2 uses the DEK to perform I/O encryption to the volume using the AES-256 algorithm

How to Encrypt an Existing EBS Volume

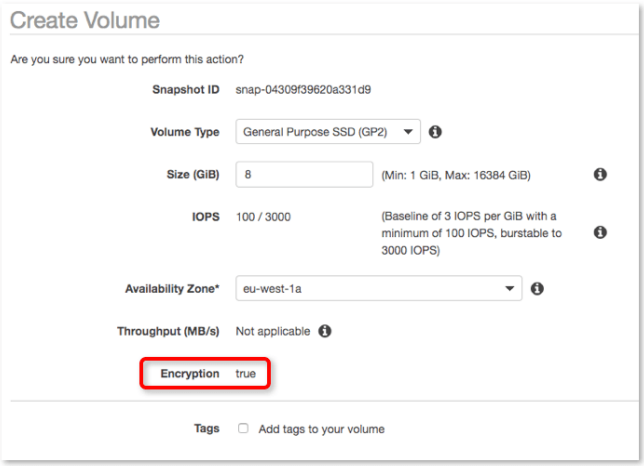
An existing unencrypted volume and the data it contains may not be encrypted. Instead, you'll need to follow another process, outlined below.

1. Select your unencrypted volume
2. Select 'Actions' – 'Create Snapshot'
3. When the snapshot is complete, select 'Snapshots' under 'Elastic Block Store' Select your newly created snapshot
4. Select 'Actions' – 'Copy'
5. Check the box for 'Encryption'
6. Select the CMK for KMS to use as required
7. Click 'Copy'



8. Select the newly created snapshot
9. Select 'Actions' – 'Create Volume'
10. You will notice that the normal 'Encryption' option is set to 'True.' Because the snapshot is itself encrypted, this cannot be modified. The volume now created from this snapshot will be encrypted

How to Encrypt an EBS Volume



Supported Instance Types for Encryption

Although all EBS volume types support encryption, not all instance types are supported.

The following AWS instance types are supported for EBS encryption:

Instance family	Instance types that support Amazon EBS encryption
General purpose	`t2.nano` | `t2.micro` | `t2.small` | `t2.medium` | `t2.large` | `t2.xlarge` | `t2.2xlarge` | `m4.large` | `m4.xlarge` | `m4.2xlarge` | `m4.4xlarge` | `m4.10xlarge` | `m4.16xlarge` | `m5.large` | `m5.xlarge` | `m5.2xlarge` | `m5.4xlarge` | `m5.12xlarge` | `m5.24xlarge`
Compute optimized	`c4.large` | `c4.xlarge` | `c4.2xlarge` | `c4.4xlarge` | `c4.8xlarge` | `c5.large` | `c5.xlarge` | `c5.2xlarge` | `c5.4xlarge` | `c5.9xlarge` | `c5.18xlarge`
Memory optimized	`r4.large` | `r4.xlarge` | `r4.2xlarge` | `r4.4xlarge` | `r4.8xlarge` | `r4.16xlarge` | `x1.16xlarge` | `x1.32xlarge` | `x1e.xlarge` | `x1e.2xlarge` | `x1e.4xlarge` | `x1e.8xlarge` | `x1e.16xlarge` | `x1e.32xlarge` | `cr1.8xlarge`
Storage optimized	`d2.xlarge` | `d2.2xlarge` | `d2.4xlarge` | `d2.8xlarge` | `h1.2xlarge` | `h1.4xlarge` | `h1.8xlarge` | `h1.16xlarge` | `i3.large` | `i3.xlarge` | `i3.2xlarge` | `i3.4xlarge` | `i3.8xlarge` | `i3.16xlarge`
Accelerated computing	`f1.2xlarge` | `f1.16xlarge` | `g3.4xlarge` | `g3.8xlarge` | `g3.16xlarge` | `p2.xlarge` | `p2.8xlarge` | `p2.16xlarge` | `p3.2xlarge` | `p3.8xlarge` | `p3.16xlarge`

Image Source: Encryption Supported Instances

Consult the official AWS documentation to stay up to date with the supported instance types.

Stuart Scott
AWS Content Lead

@Stuart_A_Scott
https://uk.linkedin.com/in/stuartanthonyscott