

**Important points:**

1. Instance store is directly connected to the instance, and EBS instance is indirectly connected through a network to its volume.
2. So, instance store is slightly faster than EBS instance.
3. Instance Store Backed EC2 – means your EC2 is using Instance Store to store its data.
4. EBS backed EC2 – means EC2 is using EBS volume to store its data.
5. General limited storage limit for ‘Instance Store’ is 10GB per instance.
6. Ephemeral Storage is nothing but non-persistent storage. Data will be lost when EC2 instance is stopped or terminated.
7. A single EBS Volume is connected to one and only one EC2 instance at a time.
8. Both EBS volume and EC2 instance must be in the same AZ.
9. An EBS volume data is replicated by AWS across multiple servers in the same AZ. This is to prevent data loss occurred from any single AWS component failure.

**Revision of EBS Volume Types:**

* SSD Backed Volume – Solid State Drive
  + General Purpose SSD [gp2]
  + Provisioned IOPS SSD [io1]
* HDD Backed Volume – Hard Disk Drive
  + Throughput Optimized HDD [ST1]
  + Cold HDD [SC1]
* Magnetic Standard

**NOTE:**

* Before 2012, there was only one EBS volume in AWS called as ‘EBS standard’.
* This EBS standard volume was later converted to Magnetic Standard.
* After 2015, AWS introduced new EBS volumes called SSD and HDD.

1. Bootable drives means , we can install any OS and use it as a root volume or C drive.
2. General Purpose, Provisioned IOPS and Magnetic Standard disks are Bootable Drives.
3. And ‘Throughput Optimised and Cold HDD’ are Non-Bootable Drives.
4. The default volume of AWS instance is ‘gp2’.
5. IOPS – Number of input and output operations per second
6. Throughput – Amount of data travelled. Unit is MB/sec.

**General Purpose SSD(gp2):**

1. Default EBS volume for an EC2 instance.
2. gp2 volumes are backed by SSDs.
3. Balances both price and performance.
4. Ratio of 3IOPS/1GB with upto 10,000 IOPS.
5. Boot volume having low latency
   1. Volume size range: 1GB – 16TB
   2. Price: $0.10GB/month

**Provisioned IOPS(io1):**

1. These volumes are ideal for both IOPS intensive and throughput intensive workloads that require extremely low latency or for mission critical applications.
2. Designed for I/O intensive applications such as large RDBMS or NoSQL Databases.
3. Recommended to use if you need more than 10,000 IOPS.
4. Can provision upto 32.000 IOPS per volume.
5. In special cases like using nitro based instances, these provision can be reached upto 64000 IPOS.
6. Volume size 4GB – 16TB
7. Price: $0.125/GB/Month(costlier among all).

**Throughput Optimised HDD(st1):**

1. ST1 is backed by Hard Disk Drives and is ideal for frequently accessed , throughput intensive workloads with large datasets.
2. Ideal for day to day operations.
3. ST1 deliver performance in terms of throughput (but not in IOPS like in SSD), measured as MB/s.
4. Used in applications like BigData, Data warehousing, Log processing etc…
5. It cannot be a boot volume.
6. Can provision upto 500 IOPS per volume.
7. Volume size: 500GB – 16TB
8. Price: $0.45 GB/month

**Cold HDD (SC1):**

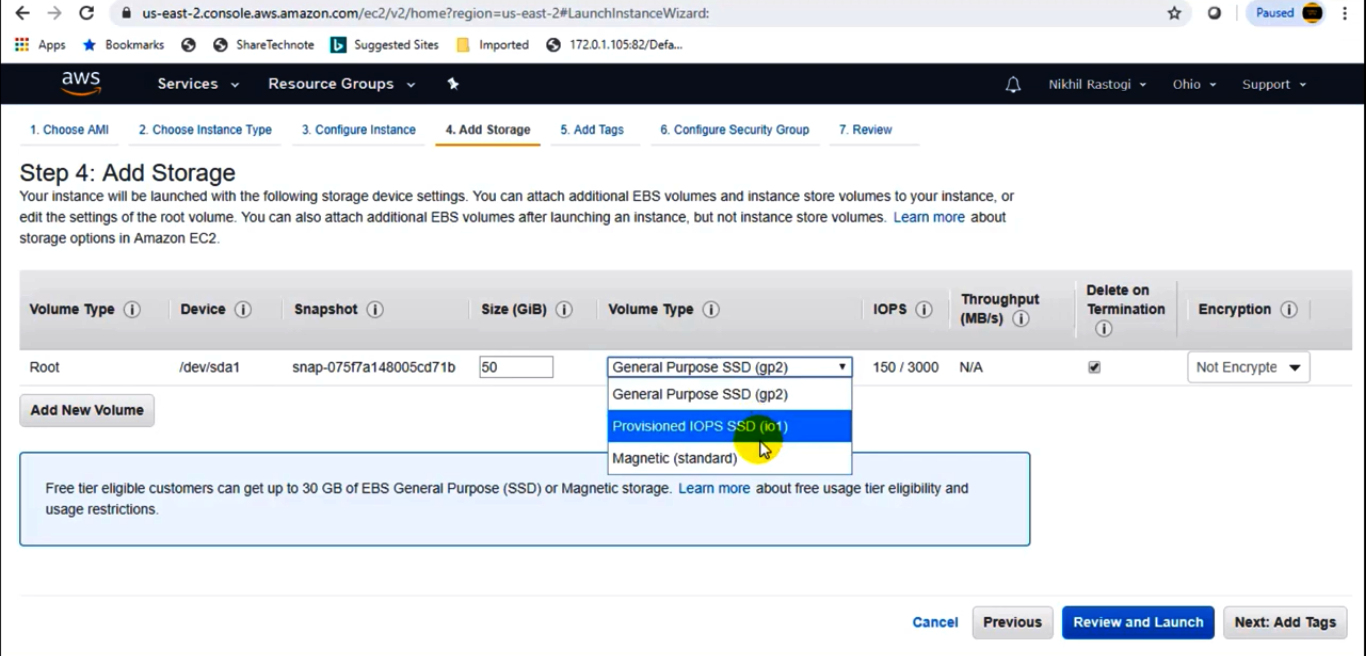
1. Backed by HDD and provides the lowest cost per GB among all EBS volume types.
2. Lowest cost storage for infrequent access workloads.
3. Used in file servers.
4. Cannot used as a boot volume.
5. Can provision up-to 250IOPS per volume.
6. Volume size: 500GB – 16TB
7. Price: $0.025/GB/month

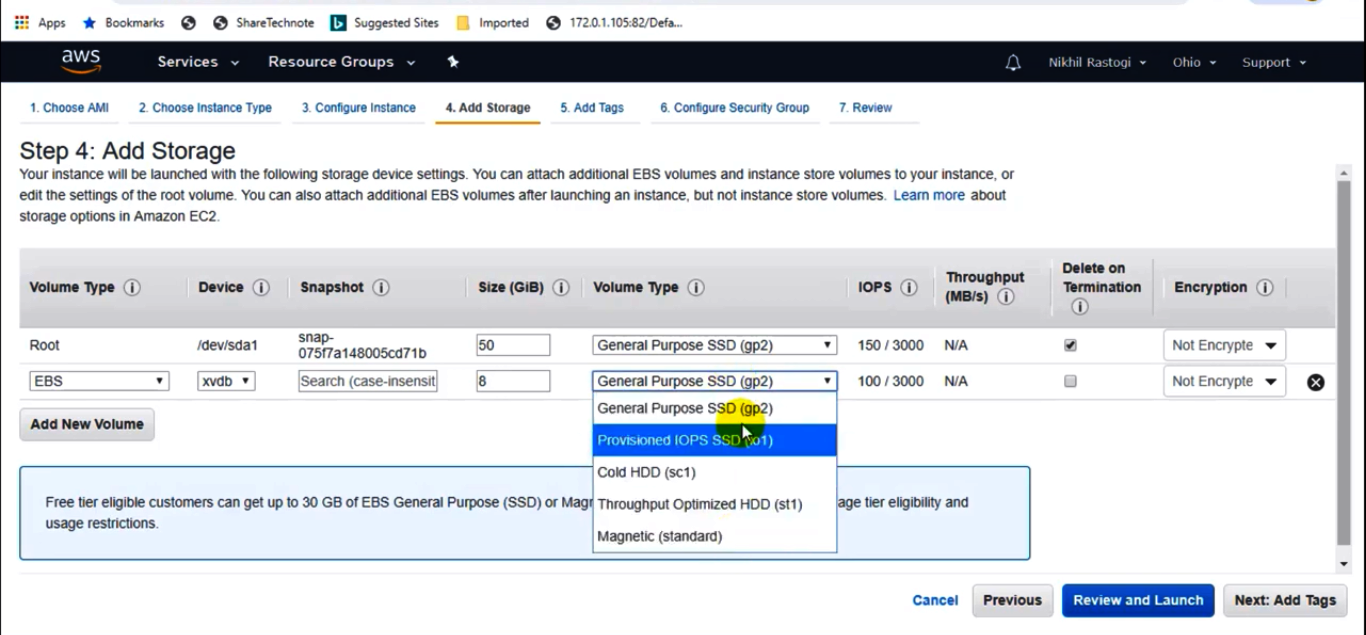
**Magnetic Standard:**

1. Lowest cost per GB of all EBS volume type that is bootable
2. Before 2012 this is the only bootable EC2 type available.
3. Ideal for workloads where data is accessed infrequently and applications where the lowest storage cost is important.
4. Price: $0.05 per gb/month

Volume: 1GB – 1TB

Max IOPS/volume: 40-200 IOPS





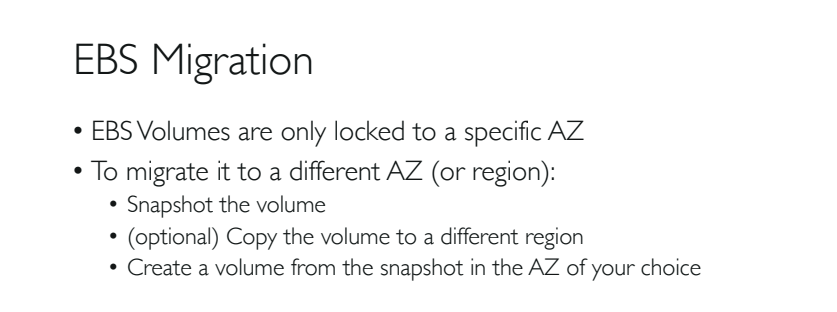
Make an Amazon EBS volume available for use on Linux:

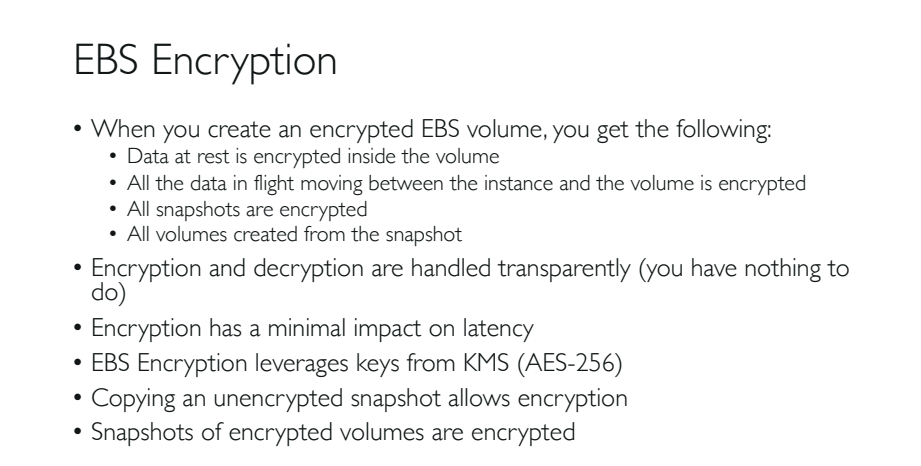
After you attach an Amazon EBS volume to your instance, it is exposed as a block device. You can format the volume with any file system and then mount it. After you make the EBS volume available for use, you can access it in the same ways that you access any other volume. Any data written to this file system is written to the EBS volume and is transparent to applications using the device.

You can take snapshots of your EBS volume for backup purposes or to use as a baseline when you create another volume. For more information, see [Amazon EBS snapshots](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html).

For more information on the notes just follow the following aws documentation link:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html





**Extra Points on EBS Encryption:**

1. EBS encryption type is supported on all EBS volume types and all EC2 Instance families.
2. Snapshots of encrypted volumes are also encrypted.
3. Creating an EBS volume from an encrypted snapshot will result an encrypted volume.
4. Data at rest: Encryption of data when the data is at rest (in volume).
5. Data in transit: Encryption of data when the data is transferring from EC2 instance to EBS volume.
6. There are many ways you can encrypt data on an EBS volume at rest, while the volume is attached to an EC2 instance.
   1. Use 3rd party EBS volume.
   2. Encryption tools.
   3. Use encrypted EBS volume.
   4. Use encryption at the OS level.
   5. Use encrypted file system on top of the EBS volume.
7. You can connect an encrypted and an un-encrypted EBS volumes to a single instance at the same time.
8. Encryption is always performed on EC2 instance side( but not in EBS).
9. There is no direct way to change the encryption state of a volume.
10. If you still want to change the encryption state of the volume follow either of the following 2 ways:

**Way-1:**

* 1. Create an extra encrypted volume for the existing instance.
  2. Mount the new volume to the EC2 instance.
  3. Copy the data from unencrypted volume to encrypted volume via EC2 instance.
  4. By default EC2 instance will encrypt the data before storing the data into an encrypted volume.
  5. Note that both volumes must be attached to the same instance.

**Way 2:**

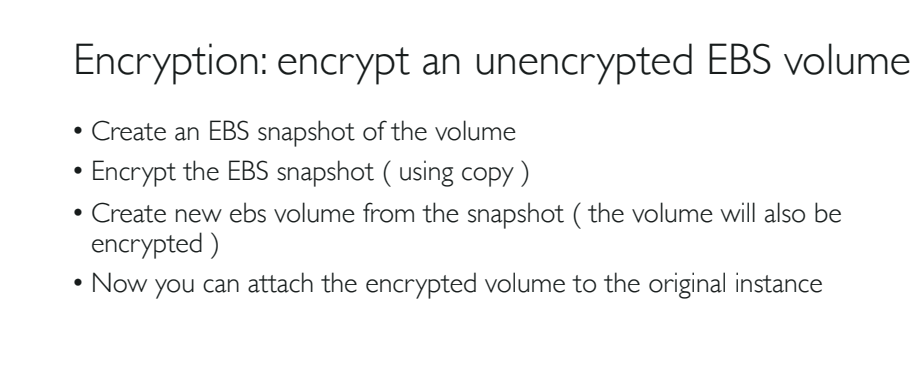
1. Create a snapshot of the unencrypted volume.
2. Copy the snapshot and choose encryption for the new copy. This will create an encrypted copy of the snapshot.
3. Use this new copy to create an EBS volume, which will be encrypted too.
4. Attach the new encrypted EBS volume to the EC2 instance.

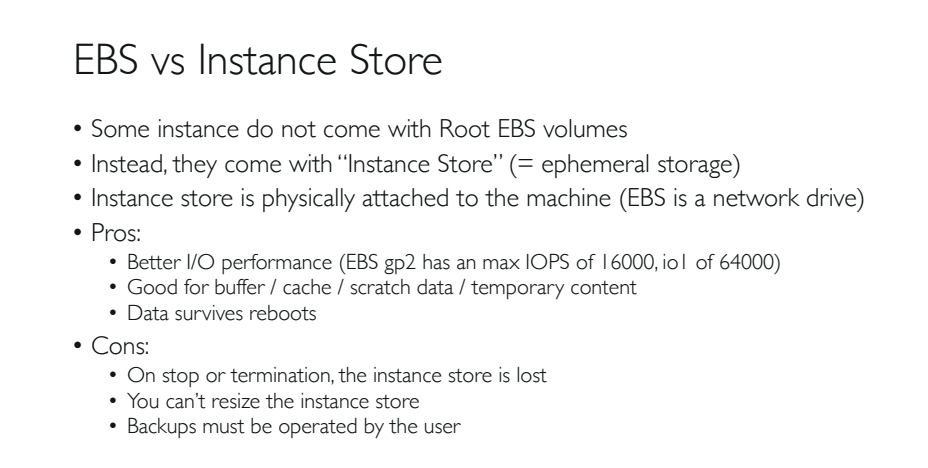
**Root EBS volume Encryption:**

1. There is no direct way to change the encryption state of a root volume.
2. We can achieve this type of encryption indirectly.
   1. Launch the instance with the EBS volume required.
   2. Do the required installations, updates, patching etc…
   3. Create and AMI from the EC2 instance.
   4. Copy the AMI and choose encryption while copying.
   5. You will get an encrypted AMI which is private (you cannot share this AMI).
   6. Use this encrypted AMI to create a new Instance, which will have the EBS root volume encrypted.

**Sharing EBS snapshot (or) EBS encryption key:**

1. To encrypt a volume or a snapshot you need a CMK key( Customer Master Key).
2. These keys are managed my AWS KMS service(Key Management Service).
3. When encrypting the first EBS volume AWS KMS creates a default CMK key.
4. After that each newly encrypted volume is encrypted with a unique AES-256 bit encryption key. This same key will be used for the subsequent snapshots, volumes etc.





Hands On: