EBS Volumes/Snapshots

# EC2 Elastic Block Storage – EBS Overview

* Amazon EBS provides highly available, reliable, durable, block-level storage volumes that can be attached to a running instance
* EBS as a primary storage device is recommended for data that requires frequent and granular updates for e.g. running a database or filesystems
* An EBS volume behaves like a raw, unformatted, external block device that can be attached to a single EC2 instance at a time
* EBS volume persists independently from the running life of an instance.
* An EBS volume can be attached to any instance within the **same Availability Zone**, and can be used like any other physical hard drive.
* EBS volumes allows encryption using the EBS encryption feature. All data stored at rest, disk I/O, and snapshots created from the volume are encrypted. Encryption occurs on the EC2 instance, providing encryption of data-in-transit from EC2 to the EBS volume
* **EBS volumes can be backed up by creating a snapshot of the volume, which is stored in S3.  EBS volumes can be created from a snapshot can be attached to an another instance within the same region**
* **EBS volumes are created in a specific Availability Zone, and can then be attached to any instances in that same Availability Zone. To make a volume available outside of the Availability Zone, create a snapshot and restore that snapshot to a new volume anywhere in that region**
* **Snapshots can also be copied to other regions and then restored to new volumes, making it easier to leverage multiple AWS regions for geographical expansion, data center migration, and disaster recovery.**
* General Purpose (SSD) volumes support up to  16000 IOPS and 250 MB/s of throughput and Provisioned IOPS (SSD) volumes support up to  80,000 IOPS and  1000 MB/s of throughput.
* **EBS Volumes can be re-sized to a larger size while its is attached to a running instance.**

Diagram

Description automatically generated

## EBS Benefits

* Data Availability
  + EBS volume is automatically replicated in an Availability Zone to prevent data loss due to failure of any single hardware component.
* Data Persistence
  + persists independently of the running life of an EC2 instance
  + persists when an instance is stopped and started or rebooted
  + Root EBS volume is deleted, by default, on Instance termination but can be modified by changing the DeleteOnTermination flag
  + All attached volumes persist, by default, on instance termination
* Data Encryption
  + can be encrypted by EBS encryption feature
  + EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256) and an Amazon-managed key infrastructure.
  + Encryption occurs on the server that hosts the EC2 instance, providing encryption of data-in-transit from the EC2 instance to EBS storage
  + Snapshots of encrypted EBS volumes are automatically encrypted.
* Snapshots
  + EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones
  + Snapshots can be used to create new volumes, increase the size of the volumes or replicate data across Availability Zones or regions
  + Snapshots are incremental backups and store only the data that was changed from the time the last snapshot was taken.
  + Snapshots size can probably be smaller then the volume size as the data is compressed before being saved to S3
  + Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

### EBS Volume Creation

* EBS volume can be created either
  + Creating New volumes
    - Completely new from console or command line tools and can then be attached to an EC2 instance in the same Availability Zone
  + Restore volume from Snapshots
    - EBS volumes can also be restored from a previously created snapshots
    - **New volumes created from existing EBS snapshots load lazily in the background.**
    - There is no need to wait for all of the data to transfer from S3 to the EBS volume before the attached instance can start accessing the volume and all its data.
    - If the instance accesses the data that hasn’t yet been loaded, the volume immediately downloads the requested data from S3, and continues loading the rest of the data in the background.
    - EBS volumes restored from encrypted snapshots are encrypted, by default
  + **EBS volumes can be created and attached to a running EC2 instance by specifying a block device mapping**

### EBS Volume Detachment

* EBS volumes can be detached from an instance explicitly or by terminating the instance
* EBS root volumes can be detached by stopping the instance
* **EBS data volumes, attached to a running instance, can be detached by unmounting the volume from the instance first.**
* **If the volume is detached without being unmounted, it might result the volume being stuck in the busy state and could possibly damaged the file system or the data it contains**
* EBS volume can be force detached from an instance, using the Force Detach option, but it might lead to data loss or a corrupted file system as the instance does not get an opportunity to flush file system caches or file system metadata
* **Charges are still incurred for the volume after its detachment**

### EBS Volume Deletion

* EBS volume deletion would wipe out its data and the volume can’t be attached to any instance. However, it can be backed up before deletion using EBS snapshots

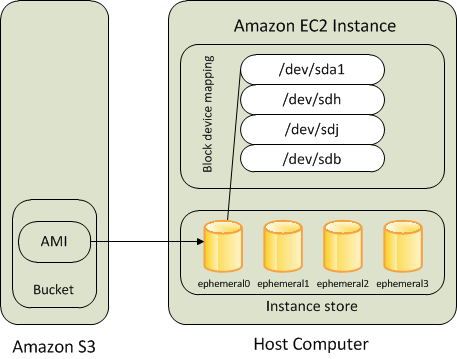
## EBS Encryption

* EBS volumes can be encrypted and attached to a supported instance type, and supports following types of data
  + Data at rest
  + All disk I/O i.e All data moving between the volume and the instance
  + All snapshots created from the volume
  + All volumes created from those snapshots
* Encryption occurs on the servers that host EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage.
* EBS encryption **is supported with all EBS volume types** (gp2, io1, st1 and sc1), and has the same IOPS performance on encrypted volumes as with unencrypted volumes, with a minimal effect on latency
* EBS encryption is **only available on select instance types**
* **Snapshots of encrypted volumes and volumes created from encrypted snapshots are automatically encrypted using the same volume encryption key**
* EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and any snapshots created from the encrypted volumes.
* EBS volumes can be encrypted using either
  + a default CMK is created for you automatically.
  + a CMK that you created separately using AWS KMS, giving you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data.
* Public or shared snapshots of encrypted volumes are not supported, because other accounts would be able to decrypt your data and needs to be migrated to an unencrypted status before sharing.
* **Existing unencrypted volumes cannot be encrypted** **directly**, but can be migrated by
  + create a unencrypted snapshot from the volume
  + create an encrypted copy of unencrypted snapshot
  + create an encrypted volume from the encrypted snapshot
* Encrypted snapshot can be created from a unencrypted snapshot by create an encrypted copy of the unencrypted snapshot
* Unencrypted volume cannot be created from an encrypted volume directly but needs to be migrated

AWS EBS vs Instance Store

* EC2 instances support two types for block level storage
  + Elastic Block Store (EBS)
  + Instance Store (Ephemeral store)
* EC2 Instances can be launched using either Elastic Block Store (EBS) or Instance Store volume as root volumes and additional volumes.
* **EC2 instances can be launched by choosing between AMIs backed by EC2 instance store and AMIs backed by EBS. However, AWS recommends use of EBS backed AMIs, because they launch faster and use persistent storage**

Instance Store (Ephemeral storage)

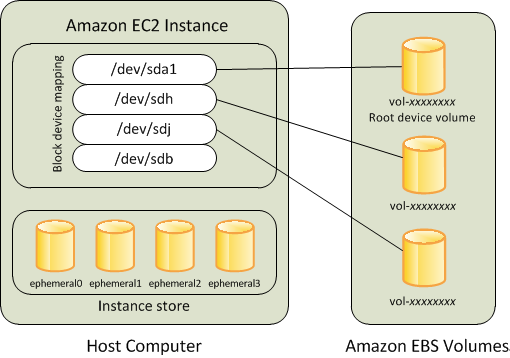


* **An Instance store backed instance is an EC2 instance using an Instance store as root device volume created from a template stored in S3.**
* Instance store volumes accesses storage from disks that are physically attached to the host computer.
* When an Instance stored instance is launched, the image that is used to boot the instance is copied to the root volume (typically sda1).
* Instance store provides temporary block-level storage for instances.
* **Data on an instance store volume persists only during the life of the associated instance; if an instance is stopped or terminated, any data on instance store volumes is lost.**

Key points for Instance store backed Instance

1. Boot time is slower then EBS backed volumes and usually less then 5 min
2. Can be selected as Root Volume and attached as additional volumes
3. **Instance store backed Instances can be of maximum 10GiB volume size**
4. Instance store volume can be attached as additional volumes only when the instance is being launched and cannot be attached once the Instance is up and running
5. Instance store backed Instances **cannot be stopped**, as when stopped and started AWS does not guarantee the instance would be launched in the same host and hence the data is lost
6. Data on Instance store volume is **LOST** in following scenarios:-
   * Failure of an underlying drive
   * Stopping an EBS-backed instance where instance store are attached as additional volumes
   * Termination of the Instance
7. Data on Instance store volume is **NOT LOST** when the instance is rebooted
8. For EC2 instance store-backed instances AWS recommends to
   * 1. distribute the data on the instance stores across multiple AZs
     2. back up critical data from the instance store volumes to persistent storage on a regular basis.
9. AMI creation requires usage on AMI tools and needs to be executed from within the running instance
10. Instance store backed Instances cannot be upgraded

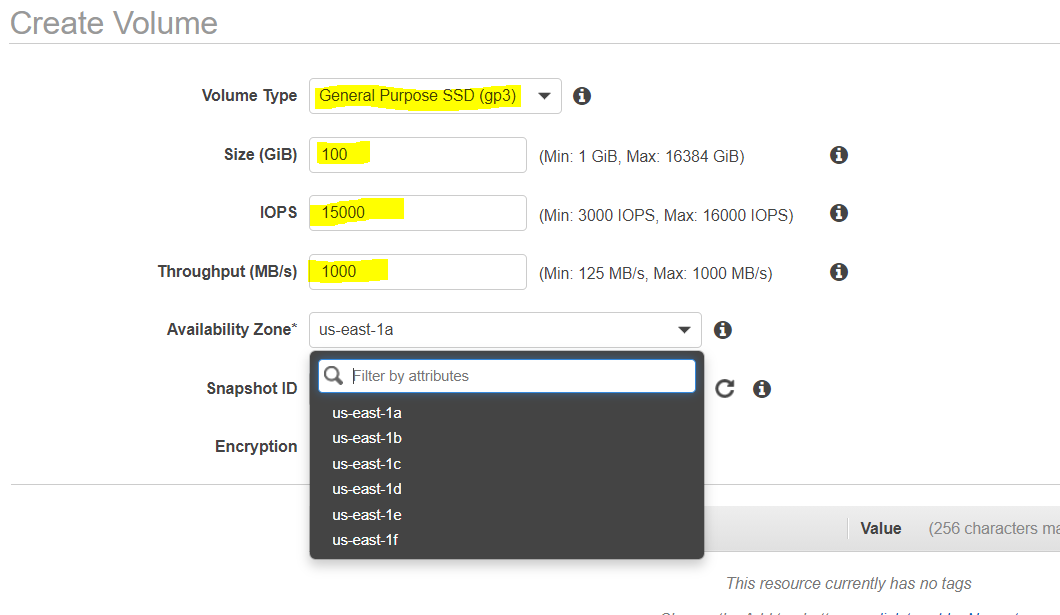
Amazon Elastic Block Store (EBS)



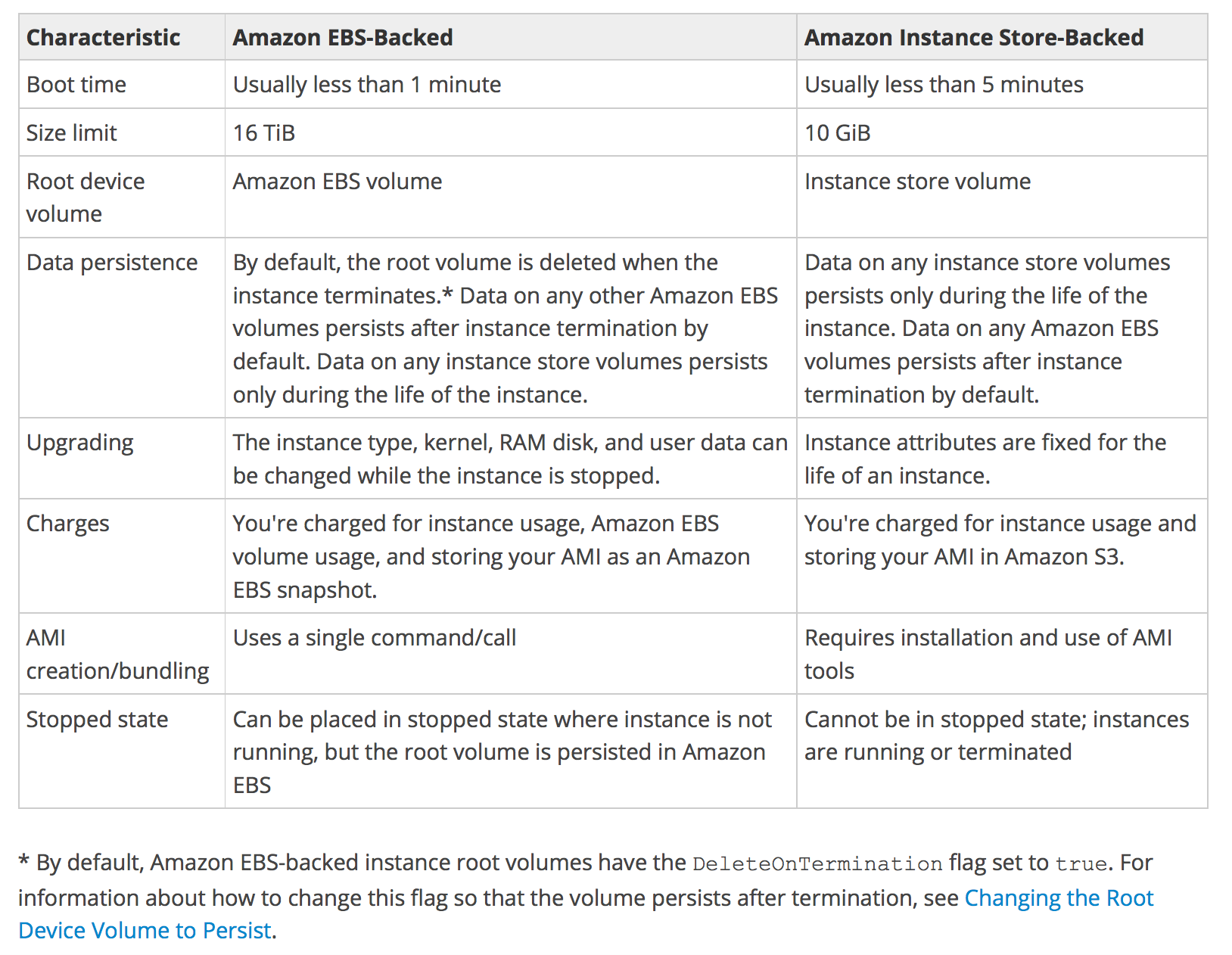
* An “EBS-backed” instance means that the root device for an instance launched from the AMI is an EBS volume created from an EBS snapshot
* An EBS volume behaves like a raw, unformatted, external block device that can be **attached to a single instance** and are not physically attached to the Instance host computer (more like a network attached storage).
* Volume persists independently from the running life of an instance.
* After an EBS volume is attached to an instance, you can use it like any other physical hard drive.
* EBS volume can be detached from one instance and attached to another instance
* EBS volumes can be created as encrypted volumes using the EBS encryption feature

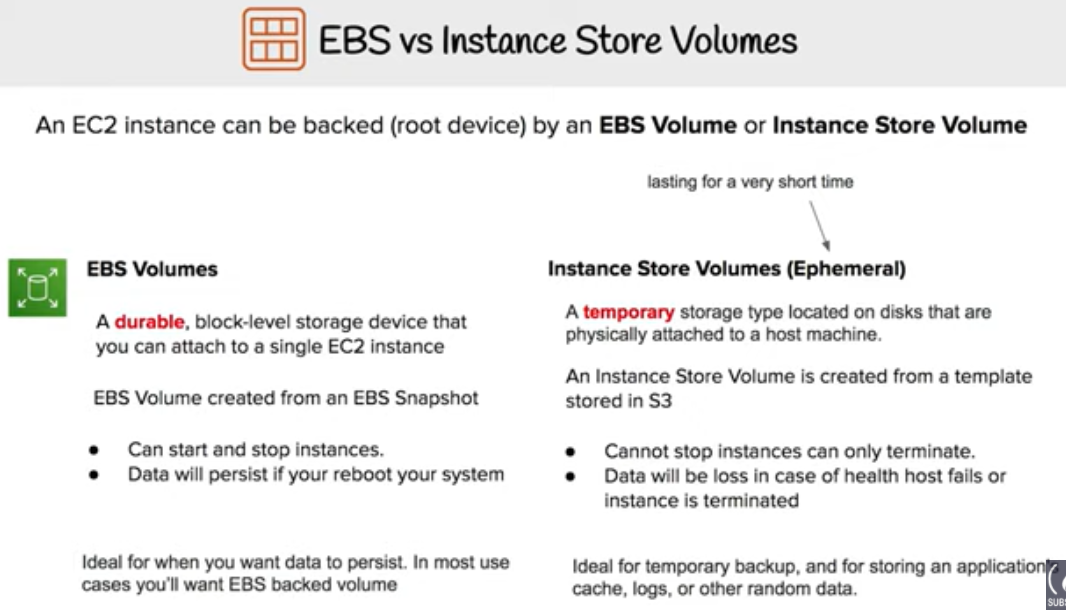
Key points for EBS backed Instance

1. Boot time is very fast usually less then a min
2. Can be selected as Root Volume and attached as additional volumes
3. **EBS backed Instances can be of maximum 16TiB volume size depending upon the OS**
4. EBS volume can be attached as additional volumes when the Instance is launched and even when the Instance is up and running
5. Data on the EBS volume is **LOST** for  
   1. EBS Root volume, if Delete On Termination flag is disabled (enabled, by default)
   2. attached EBS volumes, if the Delete On Termination flag is disabled, which is the default.
6. Data on EBS volume is **NOT LOST** in following scenarios :-
   1. Reboot on the Instance
   2. Stopping an EBS-backed instance
   3. Termination of the Instance for the additional EBS volumes. Additional EBS volumes are detached with their data intact
7. When EBS-backed instance is in a stopped state, various instance– and volume-related tasks can be done *for e.g. you can modify the properties of the instance, you can change the size of your instance or update the kernel it is using, or you can attach your root volume to a different running instance for debugging or any other purpose*
8. **EBS volumes are AZ scoped and tied to a single AZ  in which created**



1. EBS volumes are automatically replicated within that zone to prevent data loss due to failure of any single hardware component
2. AMI creation is easy using a Single command
3. EBS backed Instances can be upgraded for instance type, Kernel, RAM disk and user data



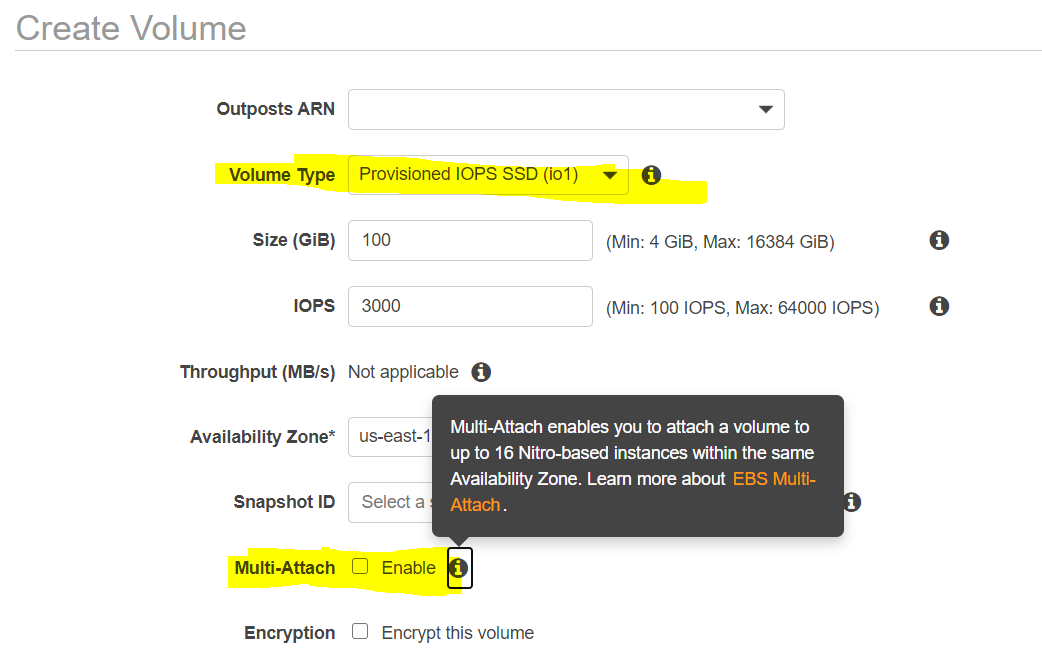


Boot Times

* EBS-backed AMIs launch faster than EC2 instance store-backed AMIs.
* When an EC2 instance store-backed AMI is launched, all the parts have to be retrieved from S3 before the instance is available.
* With an EBS-backed AMI is launched, parts are lazily loaded and only the parts required to boot the instance need to be retrieved from the snapshot before the instance is available.
* However, the performance of an instance that uses an EBS volume for its root device is slower for a short time while the remaining parts are retrieved from the snapshot and loaded into the volume.
* When you stop and restart the instance, it launches quickly, because the state is stored in an EBS volume.

**Exam tip**

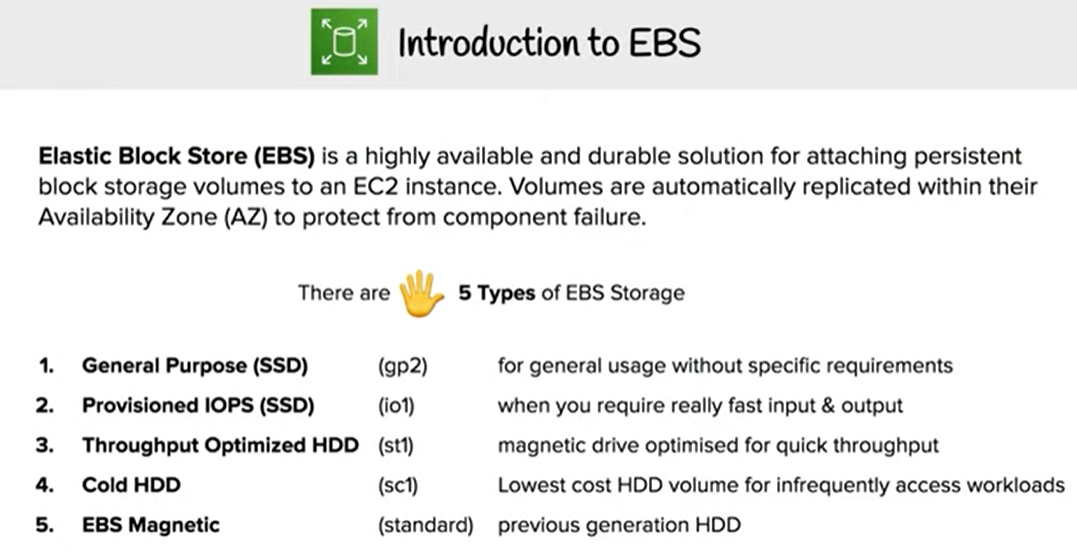
1. **Both EC2 instance & EBS volume must be in the same availability zone (AZ) for attachment.**
2. **An EBS volume data is replicated by AWS into multiple servers in same AZ to prevent data loss from any single AWS component failure**.
3. **In general, An EBS volume can be attached to only a single EC2 instance at a time. It doesn’t allow to attach another EC2 instance unless it is detached from current instance.**
4. **Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1) volume to up to 16 Nitro-based instances that are in the same Availability Zone. You can attach multiple Multi-Attach enabled volumes to an instance or set of instances. Each instance to which the volume is attached has full read and write permission to the shared volume. Multi-Attach makes it easier for you to achieve higher application availability in clustered Linux applications that manage concurrent write operations.**

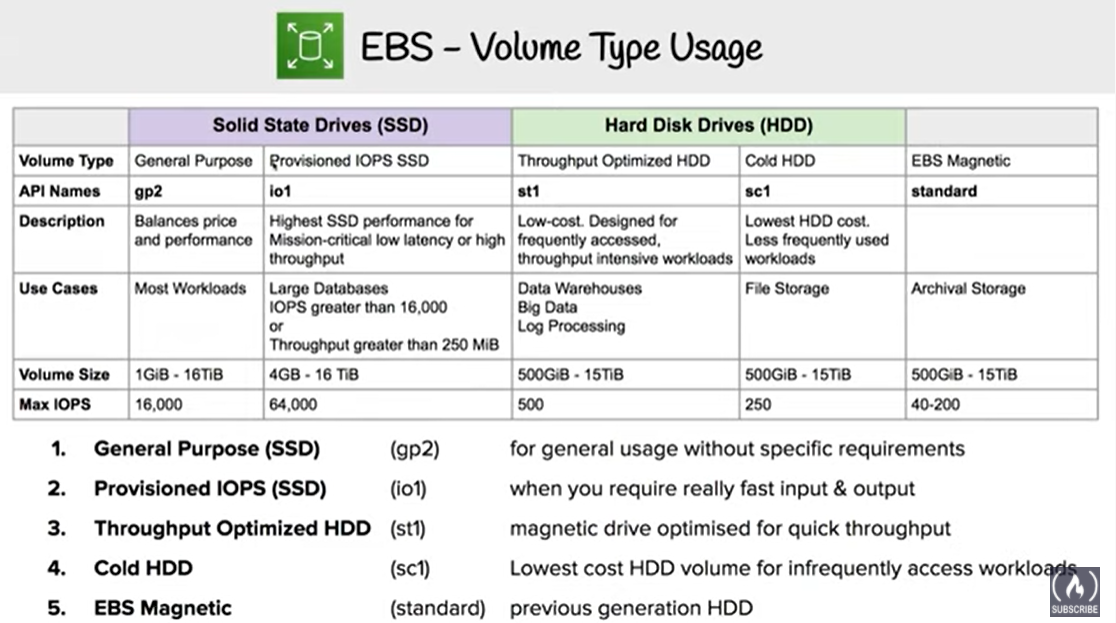


Diagram

Description automatically generated

# 





Graphical user interface, table

Description automatically generated with medium confidence

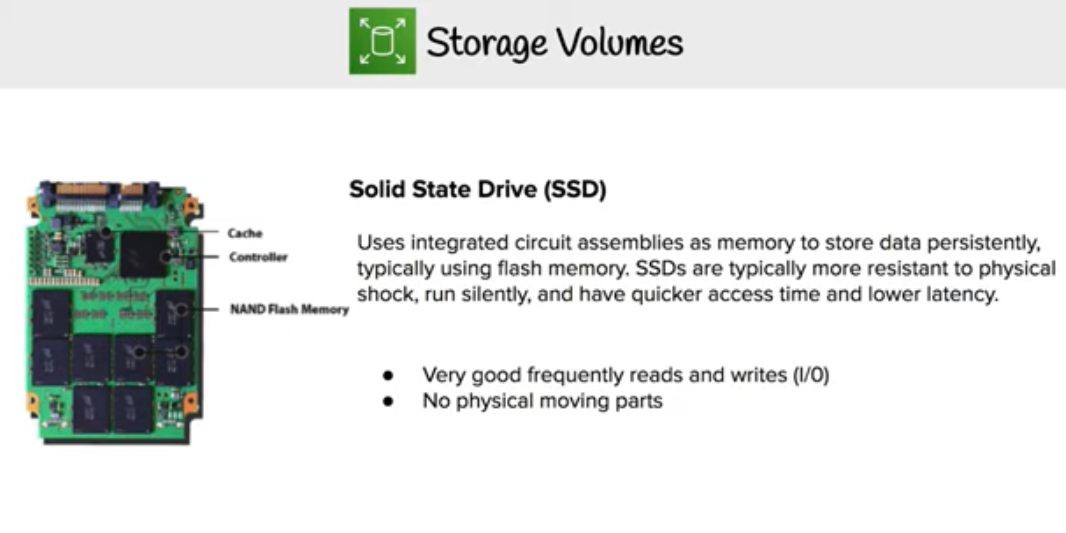
**IOPS:** Stands for Input & Output per second. it’s a standard to measure read/write operations**.**

**Throughput:** refers to how much data can be transferred from source to destination within a given timeframe

**Types of EBS Volumes**

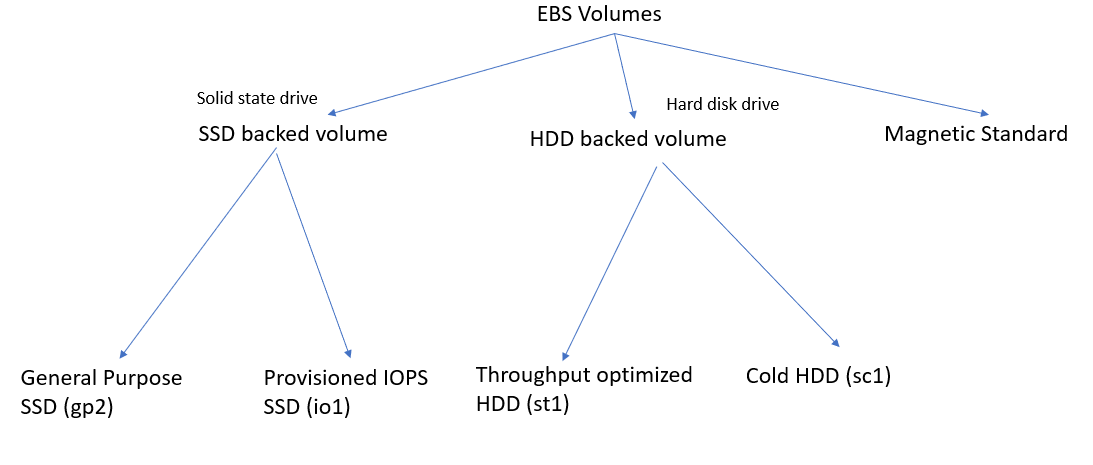
AWS provides the following EBS volume types, which differ in performance characteristics and price which can be tailored for storage performance and cost to the needs of the applications:

* **SSD-backed** volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS
  + General Purpose SSD (gp2)
  + Provisioned IOPS SSD (io1)
  + Provisioned IOPS SSD (io2)



* **HDD-backed** volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS
  + Throughput Optimized HDD (st1)
  + Cold HDD (sc1)
  + (Previous Generation)







EBS Volume Types (New Generation)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Solid State Drive (SSD) | | | Hard Disk Drives (HDD) | |
| Volume Type | General Purpose SSD (gp2) | Provisioned IOPS SSD (io1) | Provisioned IOPS SSD (io2) | Throughput Optimized HDD (st1) | Cold HDD (sc1) |
| Description | General purpose SSD volume that balances price & performance for wide variety of workloads | High performance SSD volume for mission critical low latency or high throughput workloads | Highest performance SSD volume for mission critical low latency or high throughput workloads | Low cost HDD volume designed for frequently accessed, throughput intensive workloads. | Lowest cost HDD volume designed for less frequency accessed workloads. |
| Use case | * Recommended for most of workloads * System boot volumes * Virtual desktops * Low-latency interactive apps * Development & test environments. | * Business critical application that require sustained IOPS performance, or more than 10,000 IOPS with high throughput per volume. * Large database workloads such as Mongo, Cassesndra, MSSQL, Oracle, etc | * Business critical application that require sustained IOPS performance, or more than 10,000 IOPS or 160 MBps throughput per volume. * Large database workloads such as Mongo, Cassesndra, MSSQL, Oracle, etc | * Streaming workload requiring consistent, fast throughput at low cost. * Big Data * Data warehouse * Log processing | * Throughput oriented storage for large volume of data which is in-frequently accessed. * Scenarios where the lowest storage cost is important. |
| Boot Volume | Can be used as boot volume | Can be used as boot volume | Can be used as boot volume | Cannot be used as boot volume | Cannot be used as boot volume |
| Volume Size | 1 GB -16 TB | 4 GB -16 TB |  |  |  |
| Baseline IOPS ration | 1:3  For General Purpose SSD volumes, baseline performance is 3 IOPS per GiB, with a minimum of 100 IOPS and a maximum of 16000 IOPS. General Purpose SSD volumes under 1000 GiB can burst upto 3000 IOPS | 1:50  Provisioned IOPS SSD volumes, you can provision up to 50 IOPS per GiB for io1 | 1:500  Provisioned IOPS SSD volumes, you can provision up to 500 IOPS per GiB for io2 |  |  |
| Max IOPS per volume | 16,000 | 64,000 \*\* | 64,000 \*\* |  |  |
| Max Throughput per volume | 160 Mbps | 500 Mbps |  |  |  |
| Max IOPS per Instance | 80,000 |  |  |  |  |
| Max Throughput per Instance | 1750 Mbps |  |  |  |  |
| Dominant performance attribute | IOPS | IOPS | IOPS | MiB/s | MiB/s |
| **SSD :**    **HDD :** | | | | | |

**Linux:** /dev/xvda

**Root volume mapping**

**Windows:** /dev/sda1

**Linux:** /dev/xvda

Table

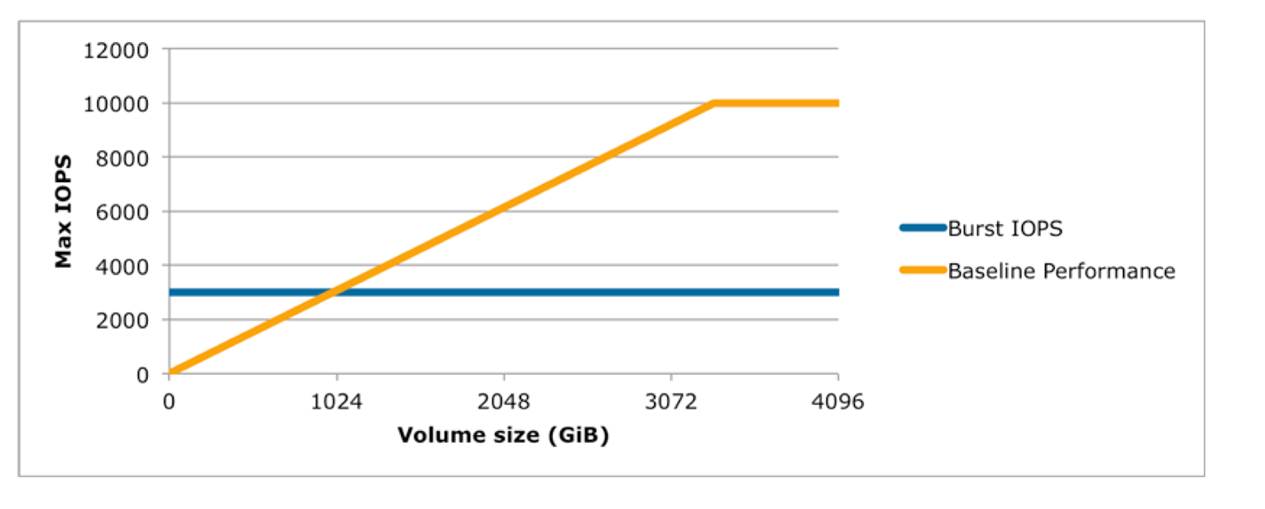
Description automatically generated

**General purpose SSD (gp2)**

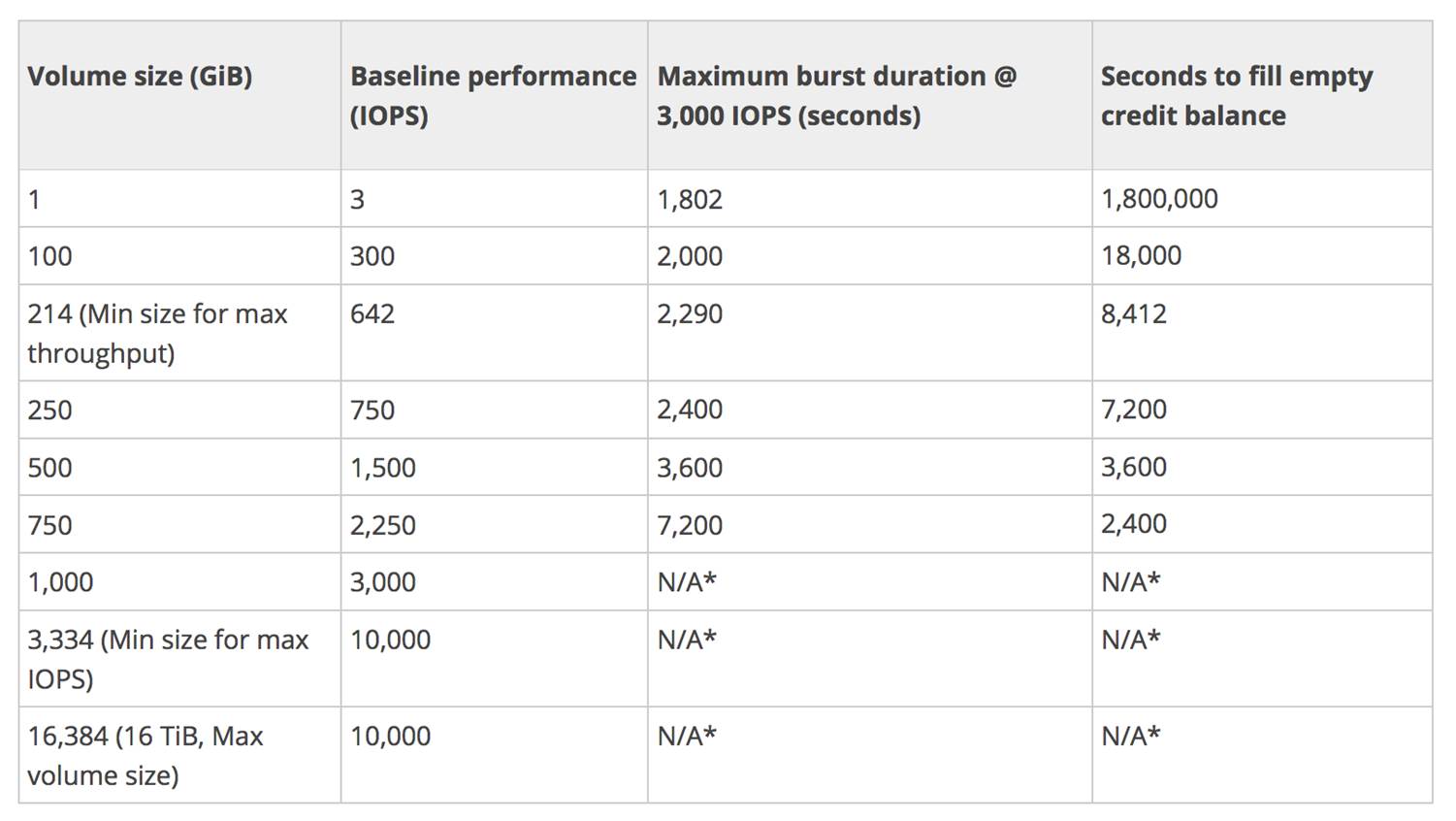
1. gp2 is the default volume type for AWS EC2 instance.
2. gp2 volumes are backed by SSD.
3. General purpose SSD balances both price & performance.
4. gp2 provides a baseline performance of 3 IOPS/GiB & it can go max till ~~10,000~~ 16,000 IOPS.
5. GP2 provides the ability to burst to 3,000 IOPS for extended periods of time for volume size less than 1 TiB and up to a maximum of ~~10,000~~ 16,000 IOPS (at 5.33 TB).
6. If the volume performance is frequently limited to the baseline level (due to an empty I/O credit balance), consider using a larger General Purpose SSD volume (with a higher baseline performance level) or switching to a Provisioned IOPS SSD volume for workloads that require sustained IOPS performance greater than ~~10,000~~ 16,000 IOPS.
7. General Purpose SSD volumes has a maximum throughput of 160 MiB/s (at 214 GiB and larger).
8. Can be used as Boot volume since it has low latency (i.e. less than 10 ms).
9. Volume size can be in range of **1GB to 16 TB**.
10. Price: $0.10 per GB/month in US. It differs region to region

I/O Credits and Burst Performance

* I/O credits represent the available credit that your General Purpose SSD volume can use to burst large amounts of I/O when more than the baseline performance is needed.
* General Purpose SSD volume performance is governed by volume size, which dictates the baseline performance level of the volume for e.g. 100 GiB volume has a 300 IOPS @ 3 IOPS/GiB
* General Purpose SSD volume size also determines how quickly it accumulates I/O credits for e.g. 100 GiB with a performance of 300 IOPS can accumulate 180K IOPS/10 mins (300 \* 60 \* 10).
* Larger volumes have higher baseline performance levels and accumulate I/O credits faster for e.g. 1 TiB has a baseline performance of 3000 IOPS
* More credits the volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed for e.g. 300 GiB volume with 180K I/O credit can burst @ 3000 IOPS for 1 minute (180K/3000)
* Each volume receives an initial I/O credit balance of 5,400,000 I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes.
* Initial credit balance is designed to provide a fast initial boot cycle for boot volumes and a good bootstrapping experience for other applications.
* Each volume can accumulate I/O credits over a period of time which can be to burst to the required performance level, up to a max of 3,000 IOPS
* Unused I/O credit cannot go beyond 54,00,000 I/O credits.



* Volumes till 1 TiB can burst up to 3000 IOPS over an above its baseline performance
* Volumes larger than 1 TiB have a baseline performance that is already equal or greater than the maximum burst performance, and their I/O credit balance never depletes.
* Baseline performance cannot be beyond 10000 IOPS for General Purpose SSD volumes and this limit is reached @ 3333 GiB



Baseline Performance

* Formula – 3 IOPS i.e. GiB \* 3
* Calculation example
  + 1 GiB volume size =  3 IOPS (1 \* 3 IOPS)
  + 250 GiB volume size = 750 IOPS (250\* 3 IOPS)

Maximum burst duration @ 3000 IOPS

* How much time can 5400000 IO credit be sustained @ the burst performance of 3000 IOPS. Subtract the baseline performance from 3000 IOPS which would be contributed by the volume size
* Formula – 5400000/(3000 – Baseline performance)
* Calculation example
  + 1 GiB volume size @ 3000 IOPS with 5400000 the burst performance can be maintained for 5400000/(3000-3) = 1802 secs
  + 250 GiB volume size @ 3000 IOPS with 5400000 the burst performance can be maintained for 5400000/(3000-3\*250) = 2400 secs

Time to fill the 5400000 I/O credit balance

* Formula – 5400000/Baseline performance
* Calculation
  + 1 GiB volume size @ 3 IOPS would require 5400000/3 = 1800000 secs
  + 250 GiB volume size @ 750 IOPS would require 5400000/750 = 7200 secs

**Provisioned IOPS SSD (io1)**

1. These volumes are ideal of IOPS intensive & throughput intensive workloads that requires extremely low latency or for mission critical applications.
2. Designed for I/O intensive applications such as large relational or NOSQL databases.
3. Use if you need more than 3000 IOPS.
4. **Can provision up-to ~~32,000~~ 64,000 IOPS per volume. (Nitro based ec2 instances can go up-to 80,000 IOPS).**
5. **Volumes with greater than 32,000 IOPS requires Nitro Instances as other instance types can limit performa**
6. Can be used as Boot volume since it has very low latency.
7. Volume size can range from **4GB to 16 TB**.
8. Price: 0.125 per GB/month

**Throughput Optimized HDD (st1)**

1. Throughput optimized HDD (st1) is backed by hard disk drive & it is ideal for throughput intensive workloads with large datasets.
2. st1 volumes delivers performance in terms in throughput, measures in MBs.
   1. BigData
   2. Data Warehouse
   3. Log processing
3. It cannot be a boot/root volume.
4. Can provision up-to **500 IOPS per volume**.
5. Volume size can range from **500 GB to 16 TB**.
6. Price: $0.045 per GB/month

**Cold HDD (sc1)**

1. Cold HDD (sc1) is also backed by hard disk drive & provides lowest cost per GB of among SSD & HDD volume types.
2. Lowest cost storage for infrequent access workloads.
3. Cannot be a boot/root volume.
4. Used in file servers.
5. Can provision up-to **250 IOPS per volume**.
6. Volume size can range from **500 GB to 16 TB**.
7. Price: 0.025 per GB/month.

**Magnetic Standard**

1. Before 2012, this is the only supported volume.
2. Backed by magnetic disk hence provides lowest cost per GB of all the volume types.
3. Can be used as root/boot volume.
4. Magnetic volumes are idea for workloads where data is accessed in-frequently. Also, it used in application where storage cost is important.
5. Can provision IOPS between 40-200 IOPS.
6. Volume size can range from 1 GB to 1 TB.
7. Price: 0.05 per GB/month.

**General:**

**Empty EBS volumes receive their maximum performance the moment that they are created and do not require initialization (formerly known as pre-warming).**

For volumes that were created from snapshots, the storage blocks must be pulled down from Amazon S3 and written to the volume before you can access them. This preliminary action takes time and can cause a significant increase in the latency of I/O operations the first time each block is accessed. Volume performance is achieved after all blocks have been downloaded and written to the volume.

# EBS Snapshot

* EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to S3, where it is stored redundantly in multiple Availability Zones
* Snapshots can be used to create new volumes, increase the size of the volumes or replicate data across Availability Zones
* Snapshots are incremental backups and store only the data that was changed from the time the last snapshot was taken.
* Snapshots size can probably be smaller than the volume size as the data is compressed before being saved to S3
* Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.
* EBS Snapshots can be used to migrate or create EBS Volumes in different AZs or regions.
* Any data written to EBS volume after snapshot process is initiated will not be included in snapshots. It will be included in future incremental update.
* EBS snapshots are stored in S3 however you cannot access them directly. It can be accessed via EC2 API
* **While EBS Volumes are AZ specific, Snapshots are region specific**.
* **To migrate an EBS volume from AZ-1 to AZ-2, first create a snapshot (region specific) & then use it to create a new volume in AZ-2.**
* Any AZ in a specific region can use snapshots to create EBS volumes.
* EBS snapshots are stored incrementally. That means only first snapshot is taken as full copy & further snapshots would be incremental.
* While in pending state or In-Progress, snapshots will not include data from on-going read & writes to the volume.
* By-default only account owner can create volume from account snapshots.
* Per AWS account up to 5000 EBS volumes can be created.
* Per AWS account up to 10,000 snapshots can be created.

## Multi-Volume Snapshots

* Snapshots can be used to create a backup of critical workloads, such as a large database or a file system that spans across multiple EBS volumes.
* Multi-volume snapshots helps take exact point-in-time, data coordinated, and crash-consistent snapshots across multiple EBS volumes attached to an EC2 instance.
* It is no longer required to stop the instance or to coordinate between volumes to ensure crash consistency, because snapshots are automatically taken across multiple EBS volumes.

## EBS Snapshot creation

* Snapshots can be created from EBS volumes periodically and are point-in-time snapshots.
* Snapshots are **incremental** and only store the blocks on the device that changed since the last snapshot was taken
* Snapshots occur **asynchronously**; the point-in-time snapshot is created immediately while it takes time to upload the modified blocks to S3
* Snapshots can be taken from in-use volumes. However, snapshots will only capture the data that was written to the EBS volumes at the time snapshot command is issued excluding the data which is cached by any applications of OS
* To take complete snapshot of your non-root EBS volume, either stop EC2 instance OR Unmount the volume from instance.
  + Pause all file writes to the volume
  + Unmount the Volume -> Take Snapshot -> Remount the Volume
* To create snapshot of root volume, you must stop instance & then take snapshot.
* You can take snapshot of non-root EBS volume while it is in use by running EC2 instance. That means EBS volume is still accessible when snapshot is in-progress.
* Recommended ways to create a Snapshot from an EBS volume are

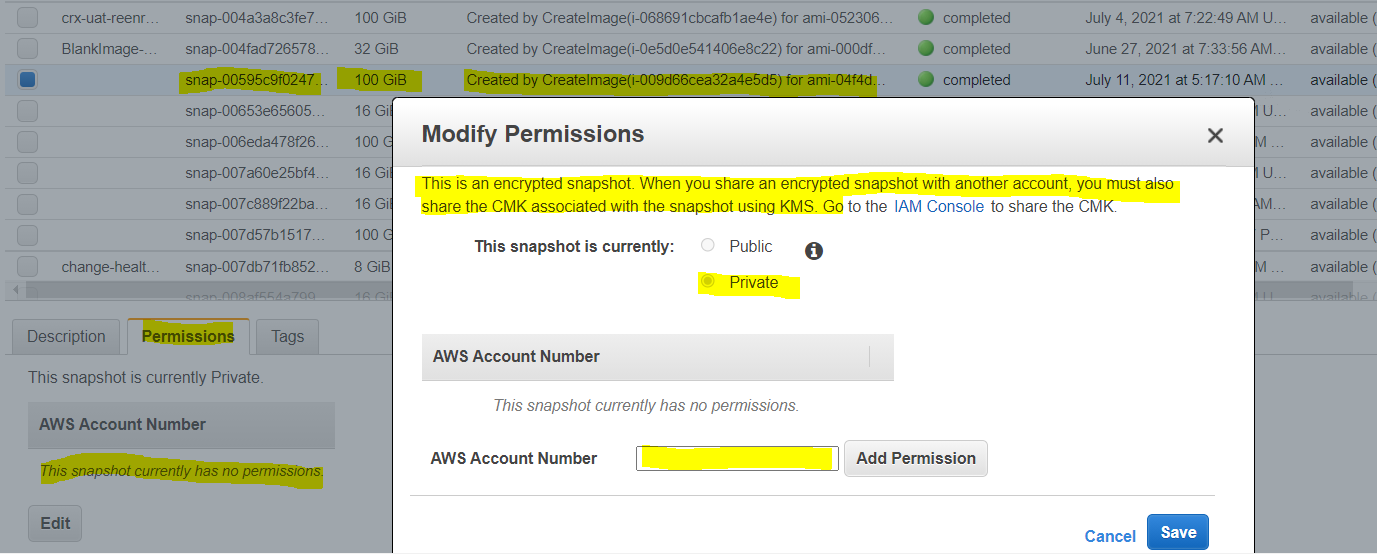
## EBS Snapshot Deletion

* When a snapshot is deleted only the data exclusive to that snapshot is removed.
* Deleting previous snapshots of a volume do not affect your ability to restore volumes from later snapshots of that volume.
* Active snapshots contain all of the information needed to restore your data (from the time the snapshot was taken) to a new EBS volume.
* Even though snapshots are saved incrementally, **the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume**.
* Snapshot of the root device of an EBS volume used by a registered AMI can’t be deleted. AMI needs to be deregistered to be able to delete the snapshot

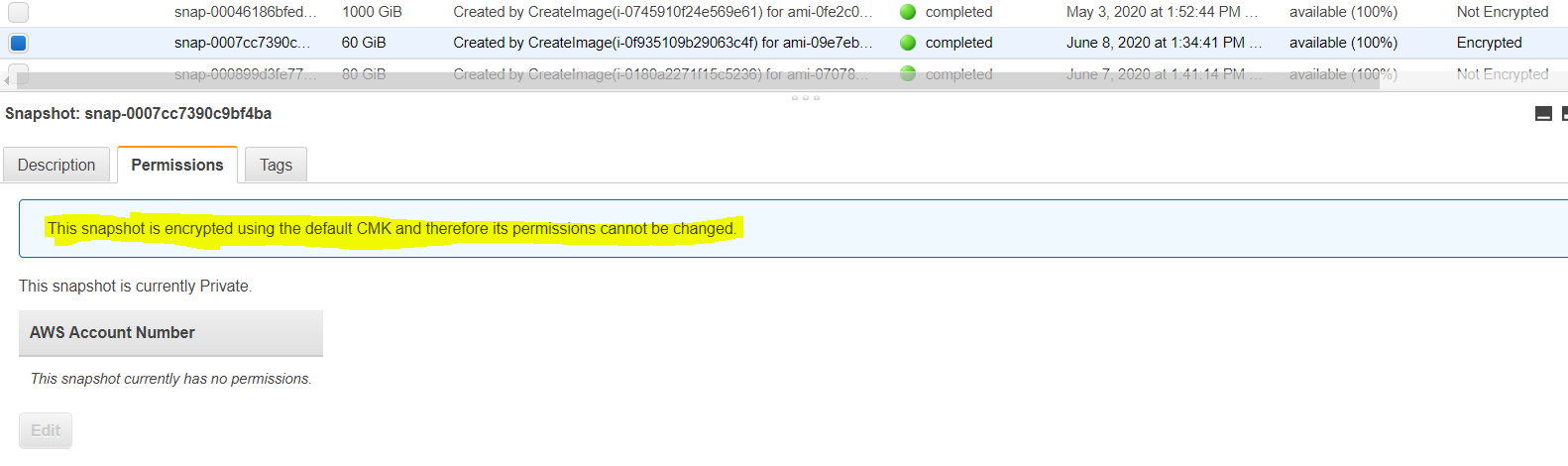
## EBS Snapshot Copy

* **Snapshots are constrained to the region** in which they are created and can be used to launch EBS volumes within the same region only
* Snapshots can be **copied across regions** to make it easier to leverage multiple regions for geographical expansion, data center migration, and disaster recovery
* Snapshots are copied with S3 server-side encryption (256-bit Advanced Encryption Standard) to encrypt the data and the snapshot copy receives a snapshot ID that’s different from the original snapshot’s ID.
* User-defined tags are not copied from the source to the new snapshot.
* First Snapshot copy to another region is always a full copy, while the rest are always **incremental**.
* When a snapshot is copied,
  + it can be encrypted if currently unencrypted or
  + can be encrypted using a different encryption key. Changing the encryption status of a snapshot or using a non-default EBS CMK during a copy operation always results in a full copy (not incremental)

## EBS Snapshot Sharing



* Snapshots can be shared by making them **public** or with specific AWS accounts by modifying the access permissions of the snapshots
* **Encrypted snapshots cannot be made available publicly.**
* **Encrypted snapshot can be shared with specific AWS accounts by sharing the custom CMK key used must also be shared to encrypt it**
* Cross-account permissions may be applied to a custom key either when it is created or at a later time.
* **Users, with access to snapshots, can copy the snapshot and create their own EBS volumes based on the snapshot while the original snapshot remains unaffected**
* **AWS prevents you from sharing snapshots that were encrypted with the default CMK.**



* **You can share your unencrypted snapshots with AWS community by making them public**.
* You can also share unencrypted snapshots with selected account by making them private & granting permission to those account.
* **You cannot make your encrypted snapshots public.**
* **You can share encrypted snapshots to selected account as follow**
  1. **Make sure you use non-default/custom CMK key to encrypt snapshot, not the default CMK key as AWS doesn’t allow it share with other account**
  2. **In the source account update custom kms access policy to grant cross account permission to the destination account.**
  3. **Without Cross account kms access, destination account will not be able to copy the snapshots nor will be able to create volume from the snapshot.**

## EBS Snapshot Encryption

* EBS snapshots fully support EBS encryption.
* **Snapshots of encrypted volumes are automatically encrypted**
* **Volumes created from encrypted snapshots are automatically encrypted**
* All data in flight between the instance and the volume is encrypted
* Volumes created from an unencrypted snapshot owned or have access to can be encrypted on-the-fly.
* **Unencrypted snapshot you own, can be encrypted during the copy process**
* Encrypted snapshot that you own or have access to, can be encrypted with a different key during the copy process.
* **First snapshot of an encrypted volume that has been created from an unencrypted snapshot is always a full snapshot**.
* **First snapshot of a re-encrypted volume, which has a different CMK compared to the source snapshot, is always a full snapshot.**
* **There is no direct way to change encryption key(CMK) used to encrypt an existing encrypted snapshots or encrypted EBS volume.**
* **If you want to change the encryption key(CMK), create a copy of snapshot & while creating snapshot copy specify that you want to re-encrypt with different encryption key.**
* **This comes in handy, when you have a snapshot that was encrypted with default CMK key(Cannot be shared with other accounts) & you want to change the key in order to share snapshot with other accounts.**
* There is no direct way to change encryption state of a volume.
* To change the state(in-directly), you need to follow either of two ways. Let’s assume you want to convert unencrypted volume to an encrypted volume
  1. **Copy content to new encrypted volume**
     + Create a new encrypted volume.
     + Attach it to EC2 instance.
     + Move data from un-encrypted volume to encrypted volume.
  2. **Snapshot Copy**
     + Create a snapshot of the unencrypted volume.
     + Copy the snapshot & choose encryption for the new copy. This will create an encrypted copy of snapshot.
     + Use this new encrypted snapshot copy to create new volume which will be encrypted too.
     + Attach new encrypted volume to EC2 instance

# 

# 

Graphical user interface

Description automatically generated

Graphical user interface, diagram

Description automatically generated

Graphical user interface

Description automatically generated

A picture containing graphical user interface

Description automatically generated

# EBS Snapshot Lifecycle Automation

* **Amazon Data Lifecycle Manager** can be used to automate the creation, retention, and deletion of snapshots taken to back up the EBS volumes.
* Automating snapshot management helps you to:
  + Protect valuable data by enforcing a regular backup schedule.
  + Retain backups as required by auditors or internal compliance.
  + Reduce storage costs by deleting outdated backups.

## 

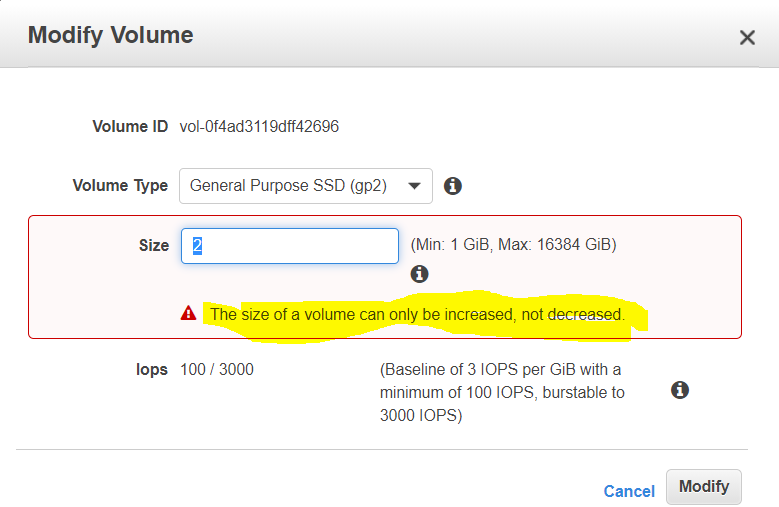
## AWS Certification Exam Practice Questions

1. \_\_\_\_\_ is a durable, block-level storage volume that you can attach to a single, running Amazon EC2 instance.
   1. Amazon S3
   2. **Amazon EBS**
   3. None of these
   4. All of these
2. Which Amazon storage do you think is the best for my database-style applications that frequently encounter many random reads and writes across the dataset?
   1. None of these.
   2. Amazon Instance Storage
   3. Any of these
   4. **Amazon EBS**
3. What does Amazon EBS stand for?
   1. Elastic Block Storage
   2. Elastic Business Server
   3. Elastic Blade Server
   4. **Elastic Block Store**
4. Which Amazon Storage behaves like raw, unformatted, external block devices that you can attach to your instances?
   1. None of these.
   2. Amazon Instance Storage
   3. **Amazon EBS**
   4. All of these
5. A user has created numerous EBS volumes. What is the general limit for each AWS account for the maximum number of EBS volumes that can be created?
   1. 10000
   2. **5000**
   3. 100
   4. 1000
6. Select the correct set of steps for exposing the snapshot only to specific AWS accounts
   1. Select Public for all the accounts and check mark those accounts with whom you want to expose the snapshots and click save.
   2. **Select Private and enter the IDs of those AWS accounts, and click Save.**
   3. Select Public, enter the IDs of those AWS accounts, and click Save.
   4. Select Public, mark the IDs of those AWS accounts as private, and click Save.
7. If an Amazon EBS volume is the root device of an instance, can I detach it without stopping the instance?
   1. Yes but only if Windows instance
   2. **No**
   3. Yes
   4. Yes but only if a Linux instance
8. Can we attach an EBS volume to more than one EC2 instance at the same time?
   1. Yes
   2. **No**
   3. Only EC2-optimized EBS volumes.
   4. Only in read mode.
9. Do the Amazon EBS volumes persist independently from the running life of an Amazon EC2 instance?
   1. **Only if instructed to when created**
   2. Yes
   3. No
10. Can I delete a snapshot of the root device of an EBS volume used by a registered AMI?
    1. Only via API
    2. Only via Console
    3. Yes
    4. **No**
11. By default, EBS volumes that are created and attached to an instance at launch are deleted when that instance is terminated. You can modify this behavior by changing the value of the flag\_\_\_\_\_ to false when you launch the instance
    1. **DeleteOnTermination**
    2. RemoveOnDeletion
    3. RemoveOnTermination
    4. TerminateOnDeletion
12. Your company policies require encryption of sensitive data at rest. You are considering the possible options for protecting data while storing it at rest on an EBS data volume, attached to an EC2 instance. Which of these options would allow you to encrypt your data at rest? (Choose 3 answers)
    1. **Implement third party volume encryption tools**
    2. Do nothing as EBS volumes are encrypted by default
    3. **Encrypt data inside your applications before storing it on EBS**
    4. **Encrypt data using native data encryption drivers at the file system level**
    5. Implement SSL/TLS for all services running on the server
13. Which of the following are true regarding encrypted Amazon Elastic Block Store (EBS) volumes? Choose 2 answers
    1. **Supported on all Amazon EBS volume types**
    2. **Snapshots are automatically encrypted**
    3. Available to all instance types
    4. Existing volumes can be encrypted
    5. Shared volumes can be encrypted
14. How can you secure data at rest on an EBS volume?
    1. Encrypt the volume using the S3 server-side encryption service
    2. Attach the volume to an instance using EC2’s SSL interface.
    3. Create an IAM policy that restricts read and write access to the volume.
    4. Write the data randomly instead of sequentially.
    5. **Use an encrypted file system on top of the EBS volume**
15. A user has deployed an application on an EBS backed EC2 instance. For a better performance of application, it requires dedicated EC2 to EBS traffic. How can the user achieve this?
    1. Launch the EC2 instance as EBS dedicated with PIOPS EBS
    2. Launch the EC2 instance as EBS enhanced with PIOPS EBS
    3. Launch the EC2 instance as EBS dedicated with PIOPS EBS
    4. **Launch the EC2 instance as EBS optimized with PIOPS EBS**
16. A user is trying to launch an EBS backed EC2 instance under free usage. The user wants to achieve encryption of the EBS volume. How can the user encrypt the data at rest?
    1. **Use AWS EBS encryption to encrypt the data at rest (**Encryption is allowed on micro instances**)**
    2. User cannot use EBS encryption and has to encrypt the data manually or using a third party tool (Encryption was not allowed on micro instances before)
    3. The user has to select the encryption enabled flag while launching the EC2 instance
    4. Encryption of volume is not available as a part of the free usage tier
17. A user is planning to schedule a backup for an EBS volume. The user wants security of the snapshot data. How can the user achieve data encryption with a snapshot?
    1. **Use encrypted EBS volumes so that the snapshot will be encrypted by AWS**
    2. While creating a snapshot select the snapshot with encryption

## By default the snapshot is encrypted by AWS

* 1. Enable server side encryption for the snapshot using S3

1. A user has launched an EBS backed EC2 instance. The user has rebooted the instance. Which of the below mentioned statements is not true with respect to the reboot action?
   1. The private and public address remains the same
   2. The Elastic IP remains associated with the instance
   3. The volume is preserved
   4. **The instance runs on a new host computer**
2. A user has launched an EBS backed EC2 instance. What will be the difference while performing the restart or stop/start options on that instance?
   1. **For restart it does not charge for an extra hour, while every stop/start it will be charged as a separate hour (**A fresh billing hour is started for the instance when you start it again. E.g., if you start a new instance and then stop/start it 3 times within the first 60 minutes, you’ll get charged for 4 hours instead of 1. When you reboot, it’s a simple reboot at the OS level and the instance stays running on the same hardware, with the same private and public IP addresses, keeps the same Elastic IP address (if associated), and keeps the same ephemeral storage without getting wiped. No new billing hour is started on a reboot and you do not give up the instance hardware.**)**
   2. Every restart is charged by AWS as a separate hour, while multiple start/stop actions during a single hour will be counted as a single hour
   3. For every restart or start/stop it will be charged as a separate hour
   4. For restart it charges extra only once, while for every stop/start it will be charged as a separate hour
3. A user has launched an EBS backed instance. The user started the instance at 9 AM in the morning. Between 9 AM to 10 AM, the user is testing some script. Thus, he stopped the instance twice and restarted it. In the same hour the user rebooted the instance once. For how many instance hours will AWS charge the user?
   1. **3 hours**
   2. 4 hours
   3. 2 hours
   4. 1 hour
4. You are running a database on an EC2 instance, with the data stored on Elastic Block Store (EBS) for persistence At times throughout the day, you are seeing large variance in the response times of the database queries Looking into the instance with the isolate command you see a lot of wait time on the disk volume that the database’s data is stored on. What two ways can you improve the performance of the database’s storage while maintaining the current persistence of the data? Choose 2 answers
   1. Move to an SSD backed instance
   2. **Move the database to an EBS-Optimized Instance**
   3. **Use Provisioned IOPs EBS**
   4. Use the ephemeral storage on an m2.4xLarge Instance Instead
5. An organization wants to move to Cloud. They are looking for a secure encrypted database storage option. Which of the below mentioned AWS functionalities helps them to achieve this?
   1. AWS MFA with EBS
   2. **AWS EBS encryption**
   3. Multi-tier encryption with Redshift
   4. AWS S3 server-side storage
6. A user has stored data on an encrypted EBS volume. The user wants to share the data with his friend’s AWS account. How can user achieve this?
   1. Create an AMI from the volume and share the AMI
   2. **Copy the data to an unencrypted volume and then share**
   3. Take a snapshot and share the snapshot with a friend
   4. If both the accounts are using the same encryption key then the user can share the volume directly
7. A user is using an EBS backed instance. Which of the below mentioned statements is true?
   1. The user will be charged for volume and instance only when the instance is running
   2. **The user will be charged for the volume even if the instance is stopped**
   3. The user will be charged only for the instance running cost
   4. The user will not be charged for the volume if the instance is stopped
8. A user is planning to use EBS for his DB requirement. The user already has an EC2 instance running in the VPC private subnet. How can the user attach the EBS volume to a running instance?
   1. The user must create EBS within the same VPC and then attach it to a running instance.
   2. **The user can create EBS in the same zone as the subnet of instance and attach that EBS to instance.**(Should be in the same AZ)
   3. It is not possible to attach an EBS to an instance running in VPC until the instance is stopped.
   4. The user can specify the same subnet while creating EBS and then attach it to a running instance.
9. A user is creating an EBS volume. He asks for your advice. Which advice mentioned below should you not give to the user for creating an EBS volume?
   1. Take the snapshot of the volume when the instance is stopped
   2. Stripe multiple volumes attached to the same instance
   3. **Create an AMI from the attached volume** (AMI is created from the snapshot)
   4. Attach multiple volumes to the same instance
10. An EC2 instance has one additional EBS volume attached to it. How can a user attach the same volume to another running instance in the same AZ?
    1. Terminate the first instance and only then attach to the new instance
    2. Attach the volume as read only to the second instance
    3. **Detach the volume first and attach to new instance**
    4. No need to detach. Just select the volume and attach it to the new instance, it will take care of mapping internally
11. What is the scope of an EBS volume?
    1. VPC
    2. Region
    3. Placement Group
    4. **Availability Zone**
12. EC2 EBS-backed (EBS root) instance is stopped, what happens to the data on any ephemeral store volumes?
    1. Data is automatically saved in an EBS volume.
    2. Data is unavailable until the instance is restarted.
    3. **Data will be deleted and will no longer be accessible.**
    4. Data is automatically saved as an EBS snapshot.
13. When an EC2 instance that is backed by an S3-based AMI is terminated, what happens to the data on the root volume?
    1. Data is automatically saved as an EBS snapshot.
    2. Data is automatically saved as an EBS volume.
    3. Data is unavailable until the instance is restarted.
    4. **Data is automatically deleted.**
14. Which of the following will occur when an EC2 instance in a VPC (Virtual Private Cloud) with an associated Elastic IP is stopped and started? (Choose 2 answers)
    1. The Elastic IP will be dissociated from the instance
    2. **All data on instance-store devices will be lost**
    3. All data on EBS (Elastic Block Store) devices will be lost
    4. The ENI (Elastic Network Interface) is detached
    5. **The underlying host for the instance is changed**
15. Which of the following provides the fastest storage medium?
    1. Amazon S3
    2. Amazon EBS using Provisioned IOPS (PIOPS)
    3. **SSD Instance (ephemeral) store**(SSD Instance Storage provides 100,000 IOPS on some instance types, much faster than any network-attached storage)
    4. AWS Storage Gateway
16. An existing application stores sensitive information on a non-boot Amazon EBS data volume attached to an Amazon Elastic Compute Cloud instance. Which of the following approaches would protect the sensitive data on an Amazon EBS volume?
    1. Upload your customer keys to AWS CloudHSM. Associate the Amazon EBS volume with AWS CloudHSM. Remount the Amazon EBS volume.
    2. **Create and mount a new, encrypted Amazon EBS volume. Move the data to the new volume. Delete the old Amazon EBS volume.**
    3. Unmount the EBS volume. Toggle the encryption attribute to True. Re-mount the Amazon EBS volume.
    4. Snapshot the current Amazon EBS volume. Restore the snapshot to a new, encrypted Amazon EBS volume. Mount the Amazon EBS volume (Need to create a snapshot, create an encrypted copy of snapshot and then create a EBS volume and mount it)
17. Is it possible to access your EBS snapshots?
    1. Yes, through the Amazon S3 APIs.
    2. **Yes, through the Amazon EC2 APIs**
    3. No, EBS snapshots cannot be accessed; they can only be used to create a new EBS volume.
    4. EBS doesn’t provide snapshots.
18. Which of the following approaches provides the lowest cost for Amazon Elastic Block Store snapshots while giving you the ability to fully restore data?
    1. Maintain two snapshots: the original snapshot and the latest incremental snapshot
    2. Maintain a volume snapshot; subsequent snapshots will overwrite one another
    3. **Maintain a single snapshot the latest snapshot is both Incremental and complete**
    4. Maintain the most current snapshot, archive the original and incremental to Amazon Glacier.
19. Which procedure for backing up a relational database on EC2 that is using a set of RAIDed EBS volumes for storage minimizes the time during which the database cannot be written to and results in a consistent backup?
    1. Detach EBS volumes, 2. Start EBS snapshot of volumes, 3. Re-attach EBS volumes
    2. Stop the EC2 Instance. 2. Snapshot the EBS volumes
    3. Suspend disk I/O, 2. Create an image of the EC2 Instance, 3. Resume disk I/O
    4. Suspend disk I/O, 2. Start EBS snapshot of volumes, 3. Resume disk I/O
    5. **Suspend disk I/O, 2. Start EBS snapshot of volumes, 3. Wait for snapshots to complete, 4. Resume disk I/O**
20. How can an EBS volume that is currently attached to an EC2 instance be migrated from one Availability Zone to another?
    1. Detach the volume and attach it to another EC2 instance in the other AZ.
    2. Simply create a new volume in the other AZ and specify the original volume as the source.
    3. **Create a snapshot of the volume, and create a new volume from the snapshot in the other AZ**
    4. Detach the volume, then use the ec2-migrate-volume command to move it to another AZ.
21. How are the EBS snapshots saved on Amazon S3?
    1. Exponentially
    2. **Incrementally**
    3. EBS snapshots are not stored in the Amazon S3
    4. Decrementally
22. EBS Snapshots occur \_\_\_\_\_
    1. **Asynchronously**
    2. Synchronously
    3. Weekly
23. What will be the status of the snapshot until the snapshot is complete?
    1. Running
    2. Working
    3. Progressing
    4. **Pending**
24. Before I delete an EBS volume, what can I do if I want to recreate the volume later?
    1. Create a copy of the EBS volume (not a snapshot)
    2. **Create and Store a snapshot of the volume**
    3. Download the content to an EC2 instance
    4. Back up the data in to a physical disk
25. Which of the following are true regarding encrypted Amazon Elastic Block Store (EBS) volumes? Choose 2 answers
    1. **Supported on all Amazon EBS volume types**
    2. **Snapshots are automatically encrypted**
    3. Available to all instance types
    4. Existing volumes can be encrypted
    5. Shared volumes can be encrypted
26. Amazon EBS snapshots have which of the following two characteristics? (Choose 2.) Choose 2 answers
    1. **EBS snapshots only save incremental changes from snapshot to snapshot**
    2. **EBS snapshots can be created in real-time without stopping an EC2 instance** (the snapshot can be taken real time however it will not be consistent and the recommended way is to stop or freeze the IO)
    3. EBS snapshots can only be restored to an EBS volume of the same size or smaller (EBS volume restored from snapshots need to be of the same size of larger size)
    4. EBS snapshots can only be restored and mounted to an instance in the same Availability Zone as the original EBS volume**(**Snapshots are specific to Region and can be used to create a volume in any AZ and does not depend on the original EBS volume AZ**)**
27. A user is planning to schedule a backup for an EBS volume. The user wants security of the snapshot data. How can the user achieve data encryption with a snapshot?
    1. **Use encrypted EBS volumes so that the snapshot will be encrypted by AWS**(Refer [link](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html))
    2. While creating a snapshot select the snapshot with encryption
    3. By default the snapshot is encrypted by AWS
    4. Enable server side encryption for the snapshot using S3
28. A sys admin is trying to understand EBS snapshots. Which of the below mentioned statements will not be useful to the admin to understand the concepts about a snapshot?
    1. **Snapshot is synchronous**
    2. It is recommended to stop the instance before taking a snapshot for consistent data
    3. Snapshot is incremental
    4. Snapshot captures the data that has been written to the hard disk when the snapshot command was executed
29. When creation of an EBS snapshot is initiated but not completed, the EBS volume
    1. Cannot be detached or attached to an EC2 instance until me snapshot completes
    2. Can be used in read-only mode while me snapshot is in progress
    3. **Can be used while the snapshot is in progress**
    4. Cannot be used until the snapshot completes
30. You have a server with a 5O0GB Amazon EBS data volume. The volume is 80% full. You need to back up the volume at regular intervals and be able to re-create the volume in a new Availability Zone in the shortest time possible. All applications using the volume can be paused for a period of a few minutes with no discernible user impact. Which of the following backup methods will best fulfill your requirements?
    1. **Take periodic snapshots of the EBS volume**
    2. Use a third-party Incremental backup application to back up to Amazon Glacier
    3. Periodically back up all data to a single compressed archive and archive to Amazon S3 using a parallelized multi-part upload
    4. Create another EBS volume in the second Availability Zone attach it to the Amazon EC2 instance, and use a disk manager to mirror me two disks
31. A user is creating a snapshot of an EBS volume. Which of the below statements is incorrect in relation to the creation of an EBS snapshot?
    1. Its incremental
    2. It can be used to launch a new instance
    3. **It is stored in the same AZ as the volume** (stored in the same region)
    4. It is a point in time backup of the EBS volume
32. A user has created a snapshot of an EBS volume. Which of the below mentioned usage cases is not possible with respect to a snapshot?
    1. Mirroring the volume from one AZ to another AZ
    2. Launch an instance
    3. **Decrease the volume size**
    4. Increase the size of the volume



1. What is true of the way that encryption works with EBS?
   1. Snapshotting an encrypted volume makes an encrypted snapshot; restoring an encrypted snapshot creates an encrypted volume when specified / requested.
   2. Snapshotting an encrypted volume makes an encrypted snapshot when specified / requested; restoring an encrypted snapshot creates an encrypted volume when specified / requested.
   3. **Snapshotting an encrypted volume makes an encrypted snapshot; restoring an encrypted snapshot always creates an encrypted volume.**
   4. Snapshotting an encrypted volume makes an encrypted snapshot when specified / requested; restoring an encrypted snapshot always creates an encrypted volume.
2. Why are more frequent snapshots of EBS Volumes faster?
   1. Blocks in EBS Volumes are allocated lazily, since while logically separated from other EBS Volumes, Volumes often share the same physical hardware. Snapshotting the first time forces full block range allocation, so the second snapshot doesn’t need to perform the allocation phase and is faster.
   2. **The snapshots are incremental so that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot.**
   3. AWS provisions more disk throughput for burst capacity during snapshots if the drive has been pre-warmed by snapshotting and reading all blocks.
   4. The drive is pre-warmed, so block access is more rapid for volumes when every block on the device has already been read at least one time.
3. Which is not a restriction on AWS EBS Snapshots?
   1. **Snapshots which are shared cannot be used as a basis for other snapshots**(Snapshots shared with other users are usable in full by the recipient, including but limited to the ability to base modified volumes and snapshots)
   2. You cannot share a snapshot containing an AWS Access Key ID or AWS Secret Access Key
   3. You cannot share encrypted snapshots (NOTE: this has be updated partially where you can share a encrypted snapshot with other accounts)
   4. Snapshot restorations are restricted to the region in which the snapshots are created
4. There is a very serious outage at AWS. EC2 is not affected, but your EC2 instance deployment scripts stopped working in the region with the outage. What might be the issue?
   1. The AWS Console is down, so your CLI commands do not work.
   2. **S3 is unavailable, so you can’t create EBS volumes from a snapshot you use to deploy new volumes.**(EBS volume snapshots are stored in S3. If S3 is unavailable, snapshots are unavailable)
   3. AWS turns off the <code>DeployCode</code> API call when there are major outages, to protect from system floods.
   4. None of the other answers make sense. If EC2 is not affected, it must be some other issue.

**Question-1**

I want to access my Amazon Elastic Block Store (Amazon EBS) volume from more than one Amazon Elastic Compute Cloud (Amazon EC2) instance. Can I use [Amazon EBS Multi-Attach](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volumes-multi.html) to enable multiple EC2 instances to simultaneously access a standard file system?

**Resolution**

Standard file systems aren't supported with EBS Multi-Attach. File systems such as XFS, EXT3, EXT4, and NTFS aren't designed to be simultaneously accessed by multiple servers or EC2 instances. Therefore, these file systems don't have built-in mechanisms to manage the coordination and control of writes, reads, locks, caches, mounts, fencing, and so on.

Enabling multiple servers to simultaneously access a standard file system can result in data corruption or loss. The operation of standard file systems on EBS Multi-Attach volumes isn't a supported configuration.

EBS Multi-Attach allows the attachment of a single io1 Provisioned IOPS volume to up to 16 Nitro-based instances in the same Availability Zone. EBS Multi-Attach volumes can be used as a block-level subcomponent of an overall shared storage solution. Configuration and operation of shared storage systems should be attempted only with a deep understanding of the potential pitfalls and configuration requirements. For more information on using EBS Multi-Attach, refer to [Attaching a Volume to Multiple Instances with Amazon EBS Multi-Attach - Considerations and Limitations](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volumes-multi.html#considerations).

**Question-2** What happens when you are using gp-2 & your workload exceeds IOPS limit of the gp2 volume you had provisioned?

**Resolution**

* You will start to get your IO requests queuing
* Depending on your application’s sensitivity to IOPS & latency, your application starts slowing down.

Two approaches to address IOPS limits

1. For gp2, you can increase size of your volume, but if your volume is already 5.2TB or more, you will have already reached 16,000 IOPS (i.e. max limit) limits for gp2.
2. If you need more than 16,000 IOPS, you will need to change your storage class to provisioned IOPS.