**EBS**

* IT is a Root volume in which the OS will be run, It is also called a Boot Volume.

**BLOCK STORAGE TYPES:**

1. EBS

* Persistence: If you stop or reboot then data will not be deleted.
* If you terminate then it will be deleted.
* N/w Attached Virtual Drive.
* N/w attached is slow than physical attached.

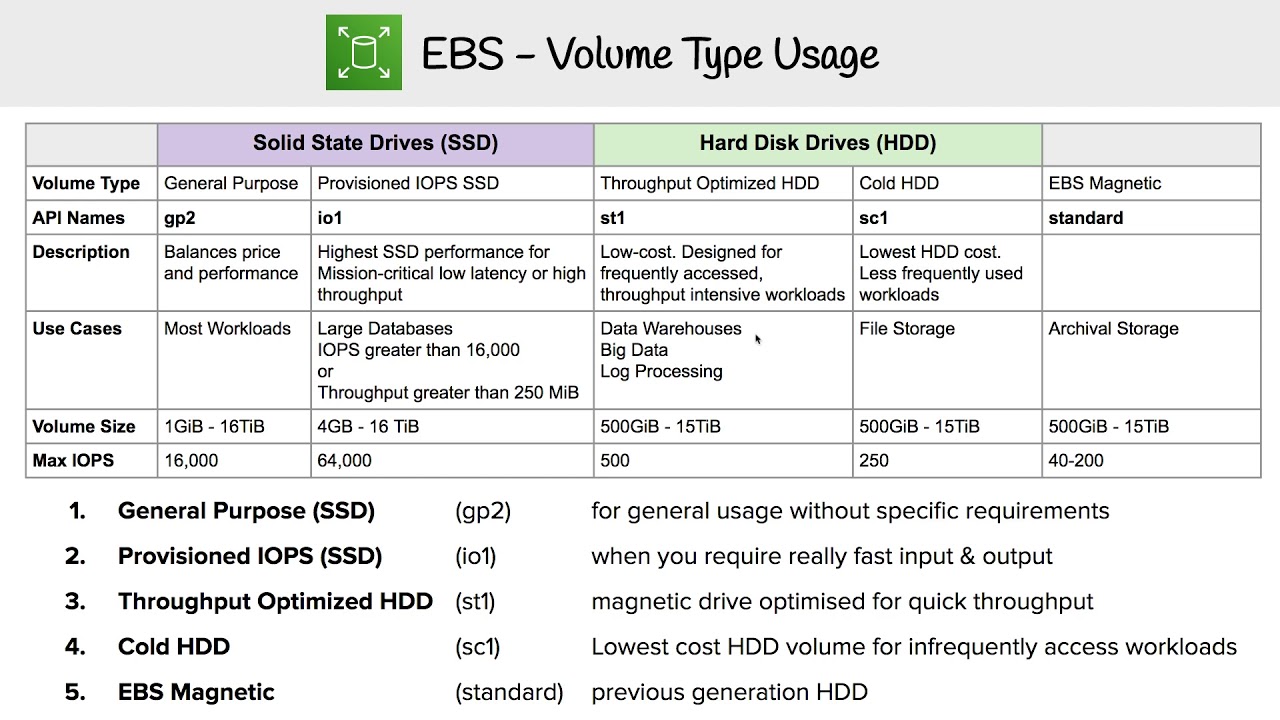
1. INSTANCE STORE BACKED EC2:

* The virtual Hard Drive on the host is allocated to EC2 Instances.
* It is limited to some instances.
* Limited to 10 GB Per Device.
* Non-Persistence (Ephemeral)
* Can only be rebooted, if you stop or terminate data will be lost.

**EBS:**

* It is a Raw, Un-formatted, external block storage that you can attach to EC2 instances.
* Ebs attached to EC2 through the AWS Network, like a virtual Hard drive.
* Multiple EBS Volumes can be attached to a single EC2 instance.
* But you can attach a single EBS Volume to an instance once at a time.
* EBS Volumes and instances must be in the same Availability Zone.
* The Data in EBS Volume will be replicated in multiple copies in the same AZ to prevent Data loss.

**VOLUME TYPES:**



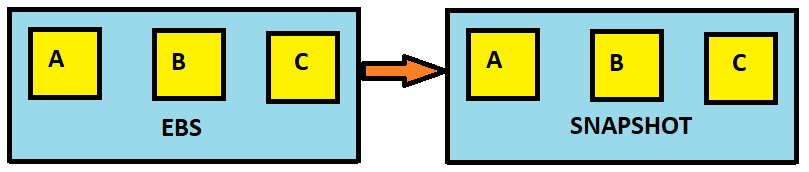
Throughput: How to speed the data transferred in MB per Second

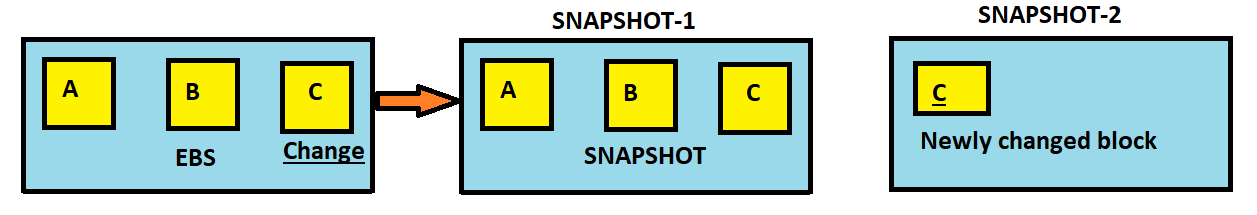
IOPS: How speed the data read and write.

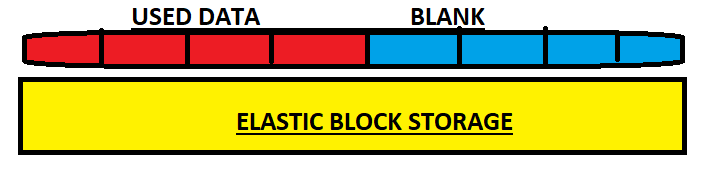
**SNAPSHOT:**

* EBS Snapshots are point-in-time images/copies of your EBS Volumes.
* Any data written to that volume after the snapshots process is initiated, will not be included in the resulting snapshot(But will be included in future, incremental updates).
* We can create up to 5000 EBS Volumes in a single account.
* We can create up to 10000 EBS snapshots in a single account.
* Snapshots can be stored on S3 (not ours) but can only be accessed through EC2 API’S.
* Snapshots are Region Specific only not AZ specific.
* To migrate EBS Volume from one region to another we can create a snapshot and create EBS volume from the snapshot in the intended AZ.
* You can create a snapshot to an EBS Volume of the same or larger size than the original volume size, from which the snapshot was initially created.
* You can take a snapshot of non-root volume while the volume Is used in running the EC2 instance.
* That means you can still access it while the snapshot is being processed.
* However, the snapshot will include data that is already written to your volume.
* A snapshot will be created immediately but will be pending until the full snapshot is completed.
* While the snapshot is in pending status you can access non-root volume, but I/O might be slower because of the snapshot activity.
* To take a complete snapshot of your non-root Ebs volume stop or unmount the Volume.
* To create a snapshot of a root EBS volume, you must stop the EC2 Instance.

**INCREMENTAL VOLUME:**







* For example, if you change the content in only one block and if you want to take a snapshot then it will not take whole blocks, whatever u made changes it will be taken as a snapshot.
* If you delete snapshot-1 then the blocks in snapshot-1 (a & b) will be shifted to snapshot-2 automatically then you get all the data by creating volume from that snapshot.
* The AWS will take a snapshot of user data but will not take a backup of unused data.

**EBS ENCRYPTION:**

* EBS Encryption will be supported by all EBS Volumes and all Instance family types.
* Snapshots of encrypted volumes are also encrypted.
* Creating EBS Volumes from Encrypted Snapshots will result in encryption Volume.
* When data is moving from EBS to EC2 that is called data in transit.
* Data encryption in rest means encryption of data while it is stored on the data storage device.
* We can attach both encrypted and unencrypted EBS Volumes to an EC2 Instance.

**EBS ENCRYPTION WAYS:**

* Use 3rd part EBS Volumes.
* Encryption Tools.
* Use Encrypted EBS Volumes.
* Use Encryption at O.S Level

LAB: TAKING A SNAPSHOT OF ROOT VOLUMES

**1. create a server on server-a and add files**

**2. create snapshot from server-a**

**3. create ebs volume from snapshot**

**4. stop server-b and remove existing volume**

**5. add the ebs volume created from snaphost and start server**