Ebs is created when the instance is created it is the root volume

Ebs has two volumes one is root volume and one is additional volume

Additional volume per instance 16 tb can be given

.

1. Provisioned IOPS SSD (io1 & io2)

1 io1

- You **choose** the number of IOPS.
- Min: 100 IOPS, Max: 64,000 IOPS (for most instances)
- But there's a rule:
 - You can provision up to 50 IOPS per GiB.

Example:

- 100 GiB volume → Max = 100 × 50 = **5,000 IOPS**
- 1,280 GiB \rightarrow 1,280 × 50 = **64,000 IOPS** (maximum allowed)

✓ io2

- Up to 256,000 IOPS much higher than io1
- You can provision up to 1,000 IOPS per GiB

Example:

- 100 GiB volume → Max = 100 × 1,000 = **100,000 IOPS**
- To get **256,000 IOPS**, volume must be ≥ 256 GiB

Proof: Use io2 only when you need:

- High durability
- Very high and consistent IOPS for mission-critical apps (e.g., Oracle, SAP HANA)

2. General Purpose SSD (gp2)

- IOPS = 3 × volume size in GiB
- Minimum: 100 IOPS
- Maximum: 16,000 IOPS (Reached when size ≥ 5,334 GiB)

Example:

- 10 GiB → 3 × 10 = 30 IOPS
 But since minimum is 100 → You get 100 IOPS
- $500 \text{ GiB} \rightarrow 3 \times 500 = 1,500 \text{ IOPS}$
- 5,500 GiB → You hit **max 16,000 IOPS**

3. General Purpose SSD (gp3)

- Default (baseline): 3,000 IOPS and 125 MiB/s
- You can increase IOPS up to 16,000, regardless of volume size.
- Rule: You can set up to **500 IOPS per GiB**, up to 16,000 IOPS.

Example:

- 5 GiB volume → Max allowed IOPS = 5 × 500 = 2,500
 Even though the system defaults to 3,000 IOPS, the limit is 2,500 (based on size), so you'd need to increase the volume to set 3,000+
- 32 GiB volume → Max IOPS = 32 × 500 = **16,000** (limit reached)

♣ Unlike gp2, gp3 lets you customize IOPS and throughput separately, but both are limited by volume size × 500.

4. Magnetic (Standard)

- Legacy type
- About 100 IOPS on average
- Can burst to hundreds of IOPS temporarily

Good for infrequent access or small archives. Not suitable for high performance workloads.

5. st1 (Throughput-Optimized HDD) & sc1 (Cold HDD)

- These are HDD-based, used for large, sequential workloads
- Performance measured in throughput (MB/s), not IOPS

Typ e	Performance	Use Case	
st1	40 MiB/s per TiB (burst to 500 MiB/s)	Big data, logs, ETL	
sc1	12 MiB/s per TiB (burst to 250 MiB/s)	Cold storage, backups	

Example:

- st1, 2 TiB volume → 2 × 40 = 80 MiB/s baseline
- sc1, 4 TiB volume → 4 × 12 = 48 MiB/s baseline

These are **cheap** and good for sequential read/write workloads (not random DB-like traffic).

Final Summary Table

Volume	IOPS Range	IOPS per GiB	Throughput (MB/s)	Best For
gp2	100–16,000	3 × size	≤ 250	General workloads
gp3	3,000-16,000	Up to 500 × GiB	125–1,000	Web/app servers, custom config
io1	100–64,000	50 × GiB	Up to 1,000-4,000	Critical DBs
io2	100–256,000	1,000 × GiB	Up to 4,000	High-IO, high-durability DBs
st1	N/A	N/A	40 MiB/s per TiB	Big data, logs
sc1	N/A	N/A	12 MiB/s per TiB	Archive, cold data

Would you like a visual chart to summarize this information too?

SSH into EC2

ssh -i "mad_max.pem"
ec2-user@ec2-15-206-195-208.ap-south-1.compute.amazonaws.com

- This connects you to your EC2 instance using the private key mad_max.pem.
- ec2-user is the default user for Amazon Linux.
- The IP is your instance's public IP (from AWS console).

Check Block Devices

lsblk

- Lists all attached block devices (disks).
- Output:
 - o xvda: Root EBS volume (8 GB)
 - o xvdb: New EBS volume (100 GB) not yet mounted

Format the New Volume

sudo mkfs -t ext4 /dev/xvdb

- Formats the /dev/xvdb disk with the **ext4 filesystem**.
- Prepares the volume to store files.
- sudo is required because formatting disks needs root permissions.

Create a Directory for Mounting

mkdir psa

- Creates a folder named psa in your home directory.
- This will be the **mount point** for the new volume.

Mount the New Volume

sudo mount /dev/xvdb psa

- Mounts the 100 GB EBS volume (/dev/xvdb) to the psa folder.
- Now, anything you store in ~/psa will actually be saved to /dev/xvdb.

Navigate into the Mounted Directory

cd psa

• Changes your current working directory to psa.

Create Files in the Mounted Volume

sudo touch a.txt sudo touch b.txt

- Creates two empty text files: a.txt and b.txt inside psa.
- sudo ensures permission is not denied (though ec2-user often owns the mount point).

Current State Summary:

- You connected to EC2, formatted and mounted a 100 GB EBS volume, and created files on it.
- These files are now stored physically on your 100 GB EBS disk (/dev/xvdb).

Next Recommended Step (Optional but Important)

To make the mount **persist after reboot**, add this to /etc/fstab:

sudo blkid /dev/xvdb

Copy the UUID and add a line like this in /etc/fstab:

UUID=your-volume-uuid /home/ec2-user/psa ext4 defaults, nofail 2

Would you like help making this mount persistent or setting permissions for non-root access?

Absolutely! Let's walk through everything that just happened — clearly and step by step.

Your Actions and What They Mean

1. List block devices

lsblk

- Shows your instance has:
 - xvda = Root disk (8GB)
 - xvdb = Extra EBS disk (100GB) not yet mounted

2. Create a mount directory

mkdir demo

 Creates a folder called demo in your home directory. This will be the mount point for the disk.

3. X Incorrect mount command

sudo mount /dev/xvdb/demo

X This failed because /dev/xvdb/demo does not exist.

- You tried to mount a non-existent device path.
- Fix: You just needed to separate the device and the mount point:

sudo mount /dev/xvdb demo

4. Correct mount command

sudo mount /dev/xvdb demo

- This mounts the 100GB volume to the demo folder.
- Now you can use demo/ to read/write files on /dev/xvdb.

5. Listing files in the mounted folder

ls -1 demo

You saw:

- a.txt and b.txt test files you created earlier.
- lost+found automatically created by the ext4 filesystem for recovery purposes.

Broadcast message from root@localhost: The system will power off now!

- This was likely triggered manually or via an auto-shutdown script, causing:
 - o All users to be logged out.
 - Your terminal session to close.

IMPORTANT: What Happens Now?

X Your mount is lost on reboot!

Because you used mount manually, the OS doesn't remember it after shutdown. You'll need to re-mount or configure it to mount automatically.