Elastic block store (EBS):-

Types of EBS volumes:-

1) SSD (gp2, gp3, io1, io2)
    - **gp2 :- (General Purpose SSD)**
      Good for everyday use.
      Works well for small to medium databases and general applications.

    - **gp3 :- (Next Gen General Purpose SSD)**
      Better for bigger, more frequent data tasks.
      Useful for databases and big data applications.

    - **Io1:- (Provisioned IOPS SSD)**
      High-performance storage.
      Best for critical business apps and databases that need a lot of speed.

    - **Io2 :- (Next gen Provisioned IOPS SSD):**
      Even higher performance and durability than io1.
      Great for demanding, fast application

2) HDD (st1, sc1)
    - **st1 (Throughput Optimized HDD):**
      Good for big data and large-scale tasks.
      Works well with data warehouses and streaming tasks.

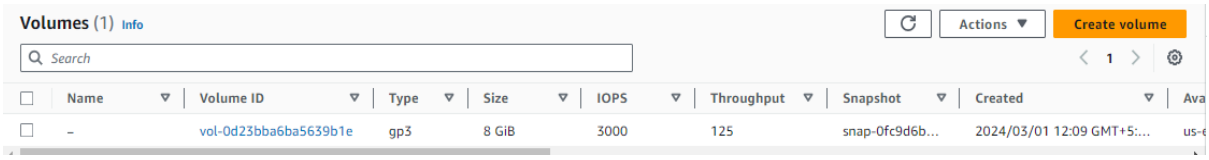
    - **sc1 (Cold HDD):**
      Budget-friendly storage for less frequently used data.
      Useful for backups and archives.

3) Previous generation volumes
    - **magnetic (Standard):**
      Basic storage with bursts of speed.
      Suitable for older systems or less demanding workloads.

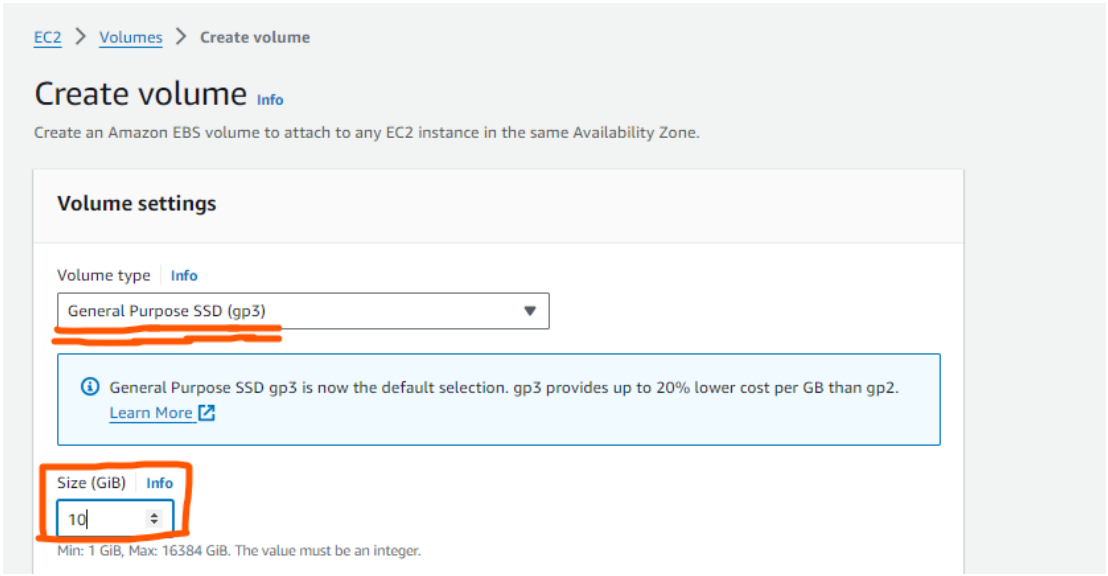
Assigning new EBS volume to Instance :-

1. create new instance (after creating new instance the default EBS volume also created)

2. click on **Create volume** select the volume type

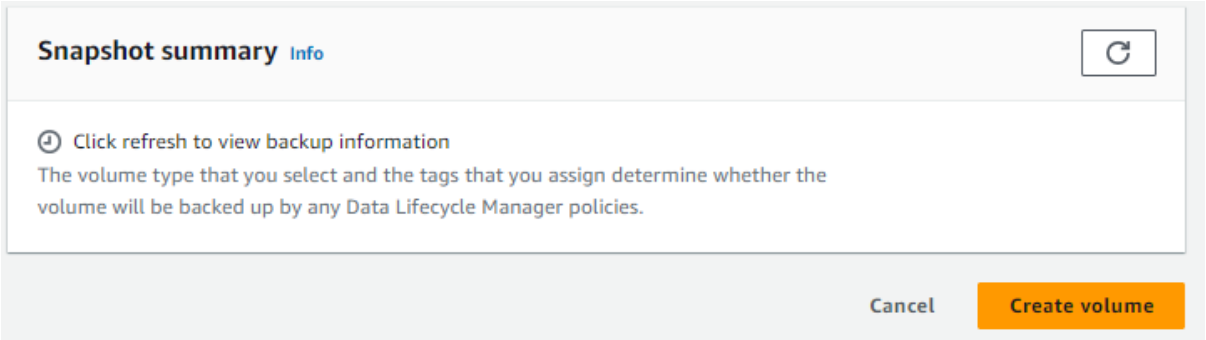


3. select **volume type** and assign **size** of volume



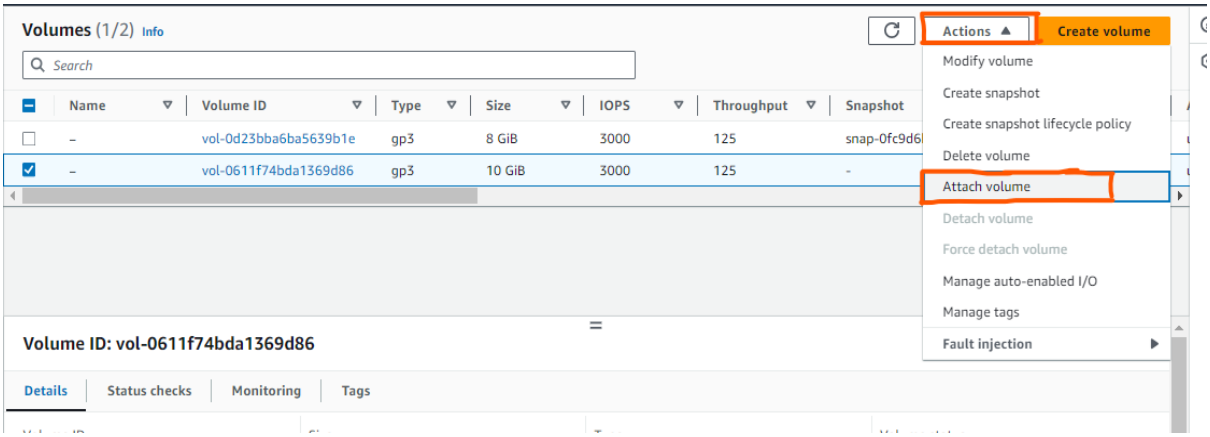
4. select the availability zone
   (note:- EBS service is a **availability zone specific service** as a reason for assigning new volume to instance, we required our instance and EBS volume in same availability zone)

5. scroll down and click on create volume

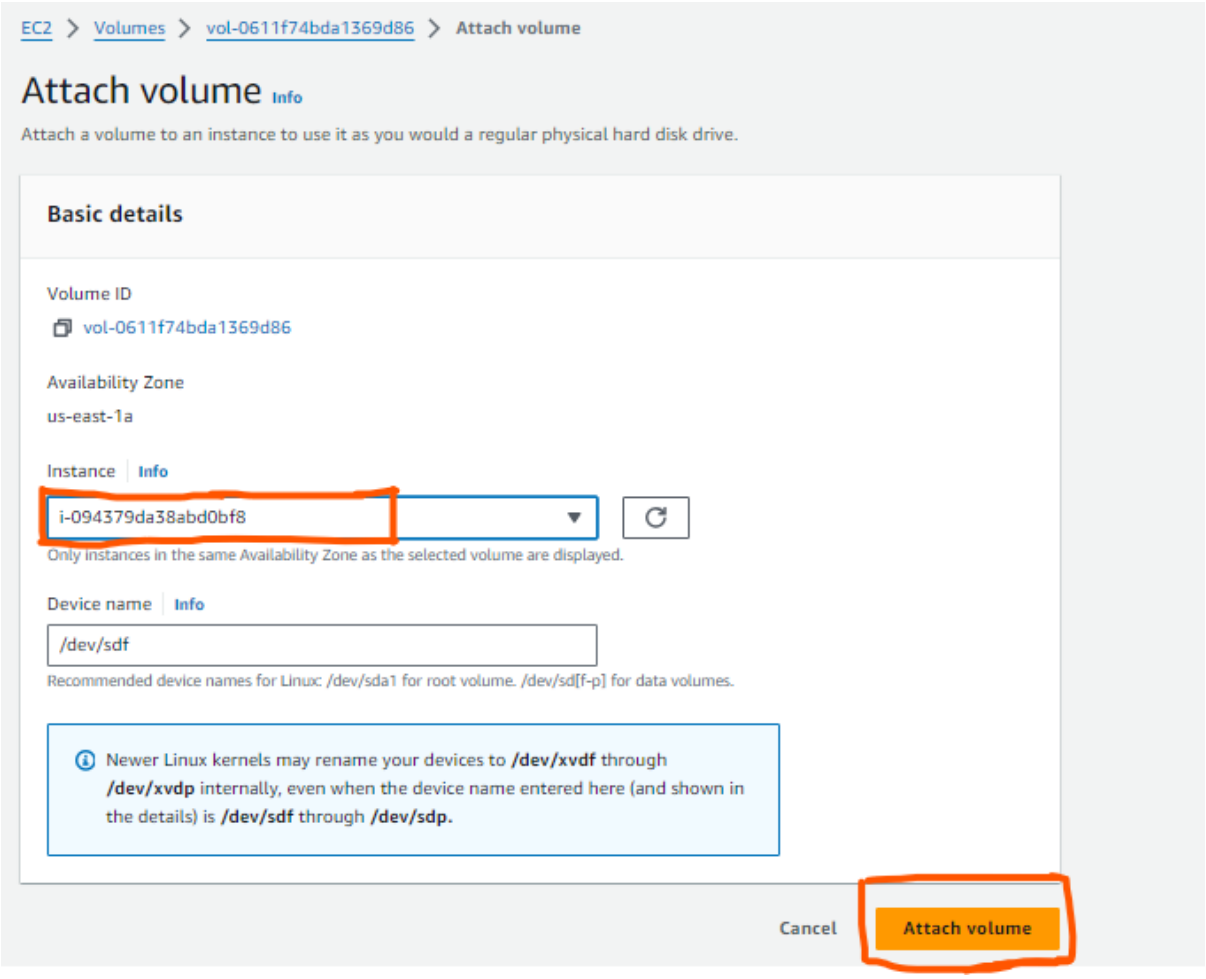


6. **new EBS volume is created…..**

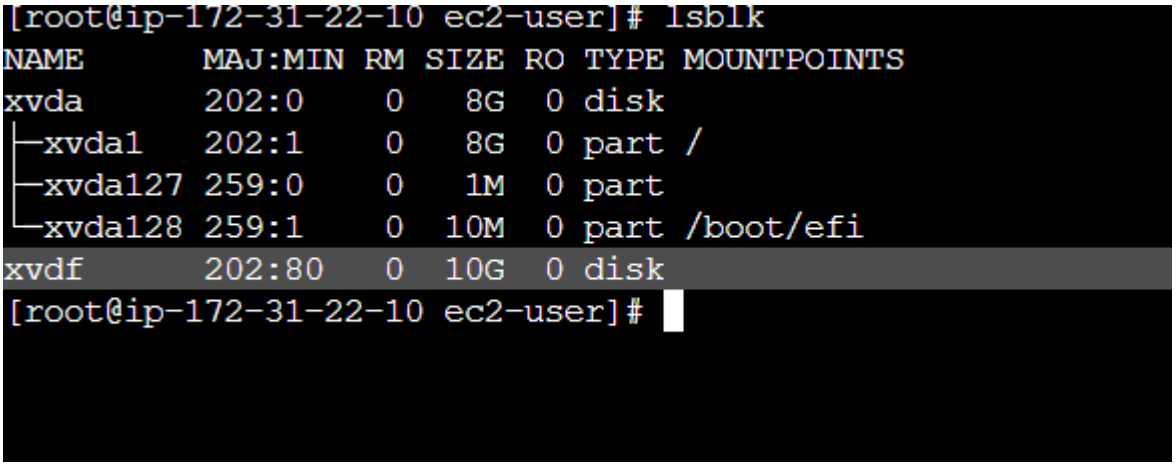
7. For attaching new EBS volume to existing instance select EBS volume and **click on Action** Button and also **click on attach** option



8. Select the instance and click on attach volume



9. New volume is attached (checked using **lsblk** command)



10.For using this volume we need to **create partition** first
(for that we have **fdisk** command)

```
[root@ip-172-31-22-10 ec2-user]# fdisk  /dev/xvdf

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xfa17224b.

Command (m for help):
```

Click on :- n (new partition) ➔ p (primary partition) ➔ 1 (partition number) ➔ default (first sector) ➔ 5g (last sector) ➔ w (for save)

11. Two partitions are created **(xvdf1, xvdf2)**

```
[root@ip-172-31-22-10 ec2-user]# partprobe
[root@ip-172-31-22-10 ec2-user]# lsblk
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda      202:0    0   8G  0 disk
├─xvda1   202:1    0   8G  0 part /
├─xvda127 259:0    0   1M  0 part
└─xvda128 259:1    0  10M  0 part /boot/efi
xvdf      202:80   0  10G  0 disk
├─xvdf1   202:81   0   5G  0 part
└─xvdf2   202:82   0   2G  0 part
[root@ip-172-31-22-10 ec2-user]#
```

12.Next we need to assign file system to created partitions

```
[root@ip-172-31-22-10 ec2-user]# mkfs.ext4  /dev/xvdf1 ; mkfs.ext4 /dev/xvdf2
mke2fs 1.46.5 (30-Dec-2021)
/dev/xvdf1 contains a ext4 file system
        created on Fri Mar  1 07:45:02 2024
Proceed anyway? (y,N) y
Creating filesystem with 1310720 4k blocks and 327680 inodes
Filesystem UUID: 005b9106-f86b-4943-9c47-a26423e99f9c
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
```

13.Next we need to mount the partitions for

- Temporary mount

```
[root@ip-172-31-22-10 ec2-user]# mount  /dev/xvdf1   /mnt
[root@ip-172-31-22-10 ec2-user]#
```

```
[root@ip-172-31-22-10 mnt]# df  -hT
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  4.0M     0  4.0M   0% /dev
tmpfs          tmpfs     475M     0  475M   0% /dev/shm
tmpfs          tmpfs     190M  2.9M  188M   2% /run
/dev/xvda1     xfs       8.0G  1.6G  6.5G  19% /
tmpfs          tmpfs     475M     0  475M   0% /tmp
/dev/xvda128   vfat       10M  1.3M  8.7M  13% /boot/efi
tmpfs          tmpfs      95M     0   95M   0% /run/user/1000
/dev/xvdf1     ext4      4.9G   24K  4.6G   1% /mnt
```

- Permanent mount

```
[root@ip-172-31-22-10 mnt]# vim  /etc/fstab
```

```
odified
UUID=81e4e009-191b-464c-8cc3-22de217d1136     /           xfs
UUID=EA7D-FA7D          /boot/efi       vfat    defaults,noatim
/dev/xvdf1  /mnt    ext4  default 0  0
```

```
[root@ip-172-31-22-10 mnt]# ls
lost+found
[root@ip-172-31-22-10 mnt]# mount -a
[root@ip-172-31-22-10 mnt]#
```

14.We successfully attached the new volume to existing instance…..

15.Now we can put our data into **/mnt** folder means the data we are putting is stored in another volume……

```
[root@ip-172-31-22-10 mnt]# touch  mayur{1..100}.txt
[root@ip-172-31-22-10 mnt]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M     0  4.0M   0% /dev
tmpfs           475M     0  475M   0% /dev/shm
tmpfs           190M  2.9M  188M   2% /run
/dev/xvda1      8.0G  1.6G  6.5G  19% /
tmpfs           475M     0  475M   0% /tmp
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi
tmpfs            95M     0   95M   0% /run/user/1000
/dev/xvdf1      4.9G   24K  4.6G   1% /mnt
[root@ip-172-31-22-10 mnt]#
```

(24kb is used in /dev/xvdf1 volume)