Aws\_Storage\_Service:

Concept: 1 "Amazon Elastic Block Store"

Amazon Elastic Block Store (Amazon EBS) is a cloud-based block storage service offered by Amazon Web Services (AWS).It provides persistent, high-performance storage volumes that can be attached to Amazon EC2 (Elastic Compute Cloud) instances.These volumes function similarly to traditional hard drives, allowing you to store files, run databases, and install applications.

Key Features of Amazon EBS:

* Persistent Storage: Data stored in EBS volumes remains intact even if the associated EC2 instance is stopped or terminated.
* High Availability: EBS volumes are automatically replicated within their Availability Zone to protect against hardware failures.
* Scalability: You can dynamically scale EBS volumes in terms of size and performance to meet your application needs without downtime.
* Snapshots: EBS supports point-in-time snapshots, which are incremental backups stored in Amazon S3. These snapshots can be used for data recovery, migration across regions, or creating new volumes.
* Encryption: EBS offers encryption at rest and in transit, managed by AWS Key Management Service.

EBS Volume Types:

Amazon EBS provides various volume types to cater to different performance and cost requirements:

* General Purpose SSD (gp2/gp3): Balanced performance for a wide range of workloads.
* Provisioned IOPS SSD (io1/io2): Designed for I/O-intensive applications like large databases.
* Throughput Optimized HDD (st1): Ideal for frequently accessed, throughput-intensive workloads.
* Cold HDD (sc1): Suitable for infrequently accessed data with lower cost requirements.

Common Use Cases:

Databases: Hosting relational and NoSQL databases that require consistent and low-latency storage

Enterprise Applications: Running applications like SAP, Oracle, and Microsoft products that demand high availability and performance

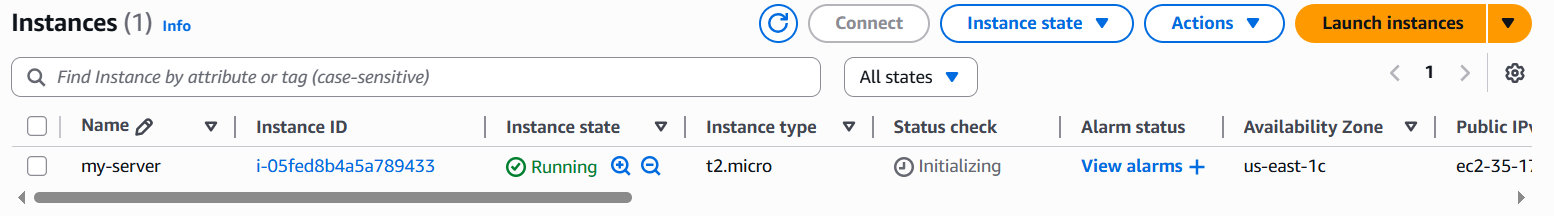
Backup and Recovery: Creating snapshots for disaster recovery and data migration purposes

Big Data Analytics: Storing and processing large datasets for analytics workloads.

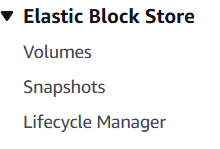
Ebs\_setup :

Step:1

   We have to launch an instance



Step:2

  Go to Elastic Block Storage Inside we have volumes and create a new volume  

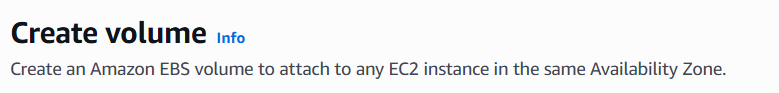
Step:3

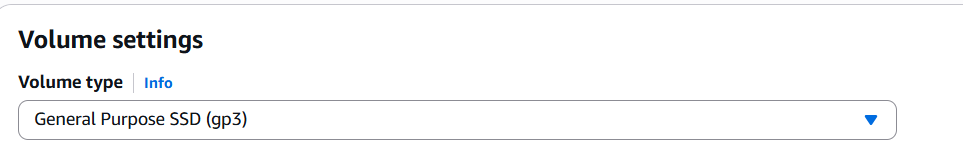
 We have default volume and we have to create a customized volume like hard disk because if we stop an instance volume will delete to overcome these problem we are create an volume

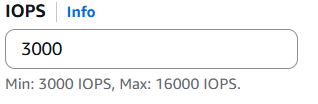


Step:4

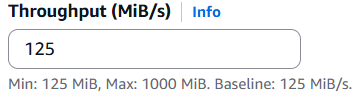
   Create a volume





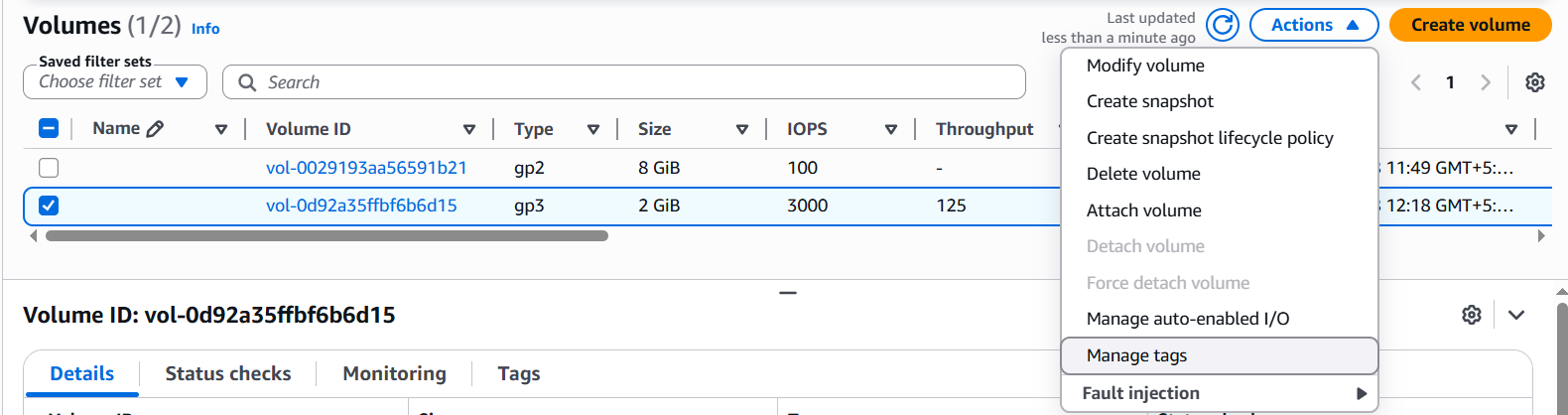


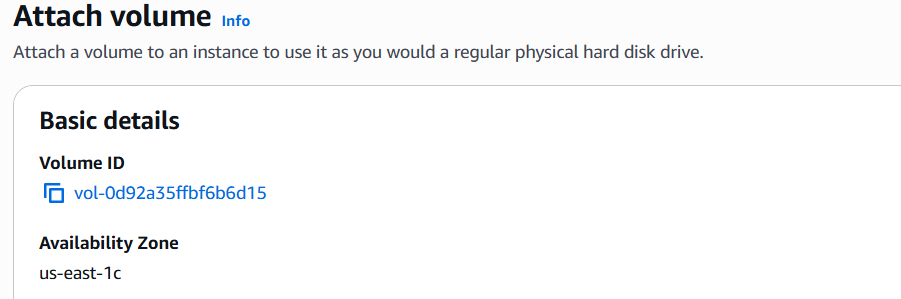
Input output per second : searching speed



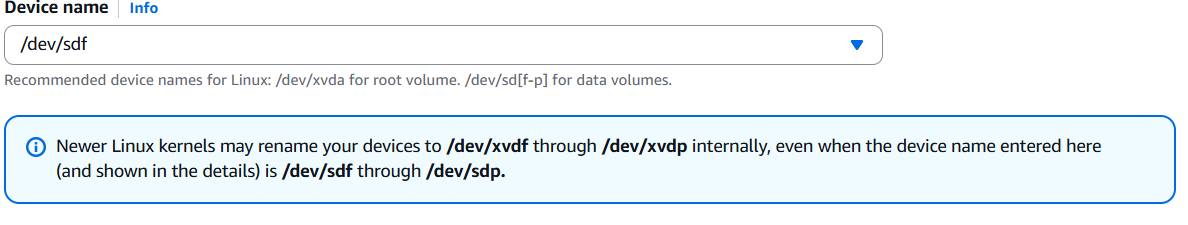
Step:5

We have to select and go to action and click attach volume

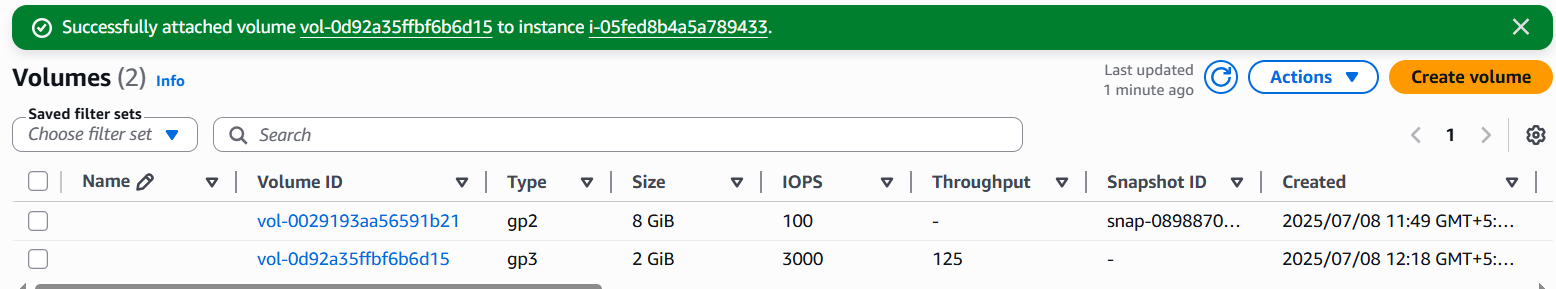








Click Attach Volume



In use

We have to check data will store in 2gb or not

Go to server and check

* Connect to instance :  sudo su
* [root@ip-172-31-80-53 ec2-user]# df -h

Filesystem      Size  Used Avail Use% Mounted on

devtmpfs        468M     0  468M   0% /dev

tmpfs           477M     0  477M   0% /dev/shm

tmpfs           477M  408K  476M   1% /run

tmpfs           477M     0  477M   0% /sys/fs/cgroup

/dev/xvda1      8.0G  1.9G  6.2G  24% /

tmpfs            96M     0   96M   0% /run/user/1000

* [root@ip-172-31-80-53 ec2-user]# lsblk : list of block device

NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

xvda    202:0    0   8G  0 disk

└─xvda1 202:1    0   8G  0 part /

xvdf    202:80   0   2G  0 disk

* To check xvdf have file system or not

             [root@ip-172-31-80-53 ec2-user]# file -s /dev/xvdf

             /dev/xvdf: data

* We should create a file system

           [root@ip-172-31-80-53 ec2-user]# mkfs -t xfs /dev/xvdf

meta-data=/dev/xvdf              isize=512    agcount=4, agsize=131072 blks

         =                       sectsz=512   attr=2, projid32bit=1

         =                       crc=1        finobt=1, sparse=1, rmapbt=0

         =                       reflink=1    bigtime=0 inobtcount=0

data     =                       bsize=4096   blocks=524288, imaxpct=25

         =                       sunit=0      swidth=0 blks

naming   =version 2              bsize=4096   ascii-ci=0, ftype=1

log      =internal log           bsize=4096   blocks=2560, version=2

         =                       sectsz=512   sunit=0 blks, lazy-count=1

realtime =none                   extsz=4096   blocks=0, rtextents=0

* To check xvdf have file system or not

             [root@ip-172-31-80-53 ec2-user]# file -s /dev/xvdf

/dev/xvdf: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)

* We create a folder because we should mount the file system to create folder

             mkdir /ebsdemo

             mount /dev/xvdf /ebsdemo

             [root@ip-172-31-80-53 ec2-user]# df -h

Filesystem      Size  Used Avail Use% Mounted on

devtmpfs        468M     0  468M   0% /dev

tmpfs           477M     0  477M   0% /dev/shm

tmpfs           477M  464K  476M   1% /run

tmpfs           477M     0  477M   0% /sys/fs/cgroup

/dev/xvda1      8.0G  1.9G  6.2G  24% /

tmpfs            96M     0   96M   0% /run/user/1000

tmpfs            96M     0   96M   0% /run/user/0

/dev/xvdf       2.0G   47M  2.0G   3% /ebsdemo

* We have to set-up these settings otherwise data will store in default volume

      [root@ip-172-31-80-53 /]# blkid

/dev/xvda1: LABEL="/" UUID="6e43975a-1916-4f3f-b40f-cfed25542d46" TYPE="xfs" PARTLABEL="Linux" PARTUUID="8642b54c-0495-4673-bd96-dfabd01ffa3a"

/dev/xvdf: UUID="6fa68314-91a7-46cb-89fc-d3d4d39b9363" TYPE="xfs"

vim  /etc/fstab

#

UUID=6e43975a-1916-4f3f-b40f-cfed25542d46     /           xfs    defaults,noatime  1   1

UUID=6fa68314-91a7-46cb-89fc-d3d4d39b9363  /ebsdemo  xfs defaults,nofail     0   0

 mount -a to check weather it mount or not

Mainly stop instance and detach volume and after we create a snapshot and we can create a number of volumes and we can change az also

This snapshot is for volume

Create a snapshot —---- go to snapshot and select and go to action and create a volume

* [ec2-user@ip-172-31-85-90 ~]$ sudo su
* [root@ip-172-31-85-90 ec2-user]# df -h

Filesystem      Size  Used Avail Use% Mounted on

devtmpfs        468M     0  468M   0% /dev

tmpfs           477M     0  477M   0% /dev/shm

tmpfs           477M  400K  476M   1% /run

tmpfs           477M     0  477M   0% /sys/fs/cgroup

/dev/xvda1      8.0G  1.9G  6.2G  24% /

tmpfs            96M     0   96M   0% /run/user/1000

* [root@ip-172-31-85-90 ec2-user]# df -h

Filesystem      Size  Used Avail Use% Mounted on

devtmpfs        468M     0  468M   0% /dev

tmpfs           477M     0  477M   0% /dev/shm

tmpfs           477M  404K  476M   1% /run

tmpfs           477M     0  477M   0% /sys/fs/cgroup

/dev/xvda1      8.0G  1.9G  6.2G  24% /

tmpfs            96M     0   96M   0% /run/user/1000

* [root@ip-172-31-85-90 ec2-user]# lsblk

NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

xvda    202:0    0   8G  0 disk

└─xvda1 202:1    0   8G  0 part /

xvdf    202:80   0   2G  0 disk

* [root@ip-172-31-85-90 ec2-user]# file -s /dev/xvdf

/dev/xvdf: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)

[root@ip-172-31-85-90 ec2-user]# mkdir /ebsdemo

[root@ip-172-31-85-90 ec2-user]# mount /dev/xvdf /ebsdemo

[root@ip-172-31-85-90 ec2-user]# cd /ebsdemo/

* [root@ip-172-31-85-90 ebsdemo]# ll

total 4

-rw-r--r-- 1 root root 22 Jul  8 07:25 file1

-rw-r--r-- 1 root root  0 Jul  8 07:24 file2

-rw-r--r-- 1 root root  0 Jul  8 07:24 file3

-rw-r--r-- 1 root root  0 Jul  8 07:24 file4

-rw-r--r-- 1 root root  0 Jul  8 07:24 file5

* [root@ip-172-31-85-90 ebsdemo]# cat file1

this is done by file1

Region wise means we take snapshot copy