Disk Sharing using EBS and EFS

Goal: Configure a EBS such that, it can be shared between different instances.

1. Create and set up an EBS such that it should be enabled by two ec2 instances
2. Verify the working
3. Study about elastic File system in AWS.
4. Configure an EFS shared between different ec2 instances.

EBS (Elastic Block Store) and EFS (Elastic File System) are both storage services provided by AWS (Amazon Web Services), but they serve different purposes:

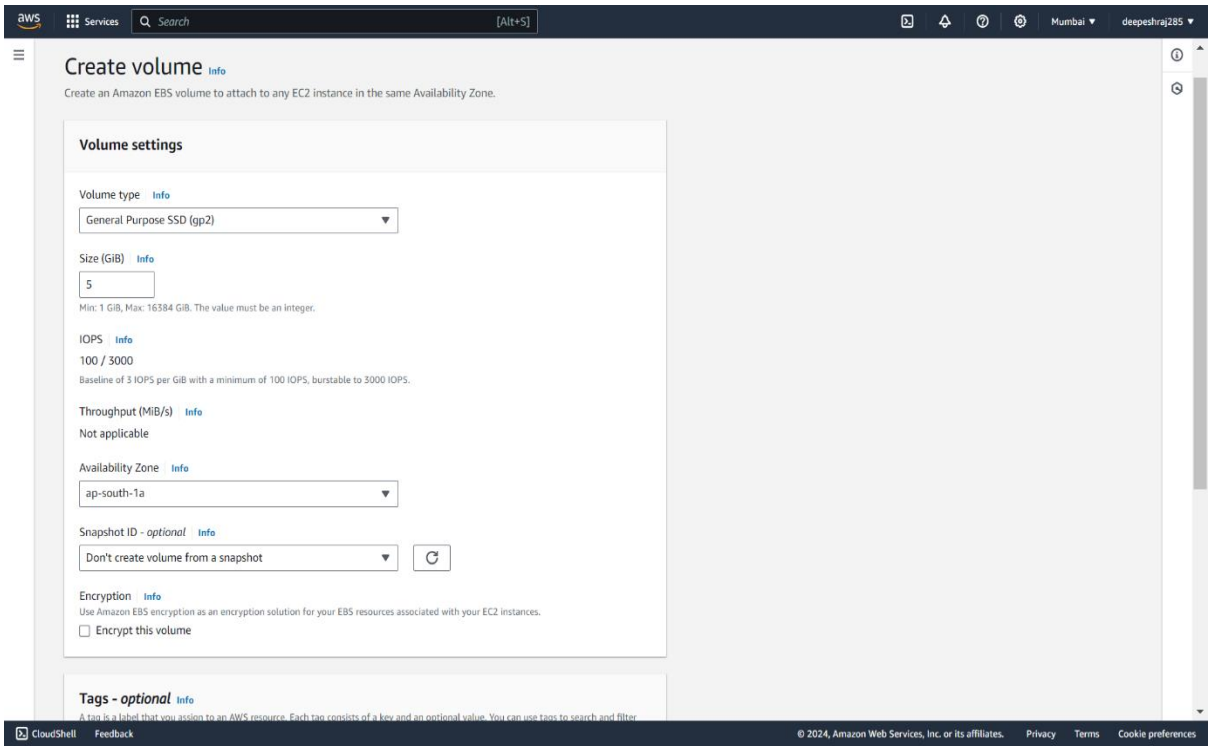
1. **EBS (Elastic Block Store)**:
    - **Type**: Provides block-level storage, meaning it is more akin to having a raw disk attached to your server.
    - **Usage**: Typically used with EC2 instances where you need low-latency and high-throughput storage that behaves like a physical hard drive.
    - **Performance**: Designed for workloads that require high IOPS (Input/Output Operations Per Second) and low-latency storage.
    - **Scalability**: Can be scaled vertically (by increasing volume size) but is tied to a single EC2 instance at a time.
    - **Use Cases**: Ideal for databases, applications that require frequent updates and access to individual files, and boot volumes for EC2 instances.
2. **EFS (Elastic File System)**:
    - **Type**: Provides file-level storage, offering a managed NFS (Network File System) that can be accessed by multiple EC2 instances concurrently.
    - **Usage**: Suitable for scenarios where multiple EC2 instances need to access the same data simultaneously, making it highly scalable and flexible.
    - **Performance**: Optimized for scalability rather than ultra-low latency, so it's great for workloads that need shared access to files across instances.
    - **Scalability**: Scales horizontally with the amount of data stored, and multiple EC2 instances can mount an EFS file system simultaneously.
    - **Use Cases**: Shared content repositories, development environments, content management systems, and applications that require shared access to data.

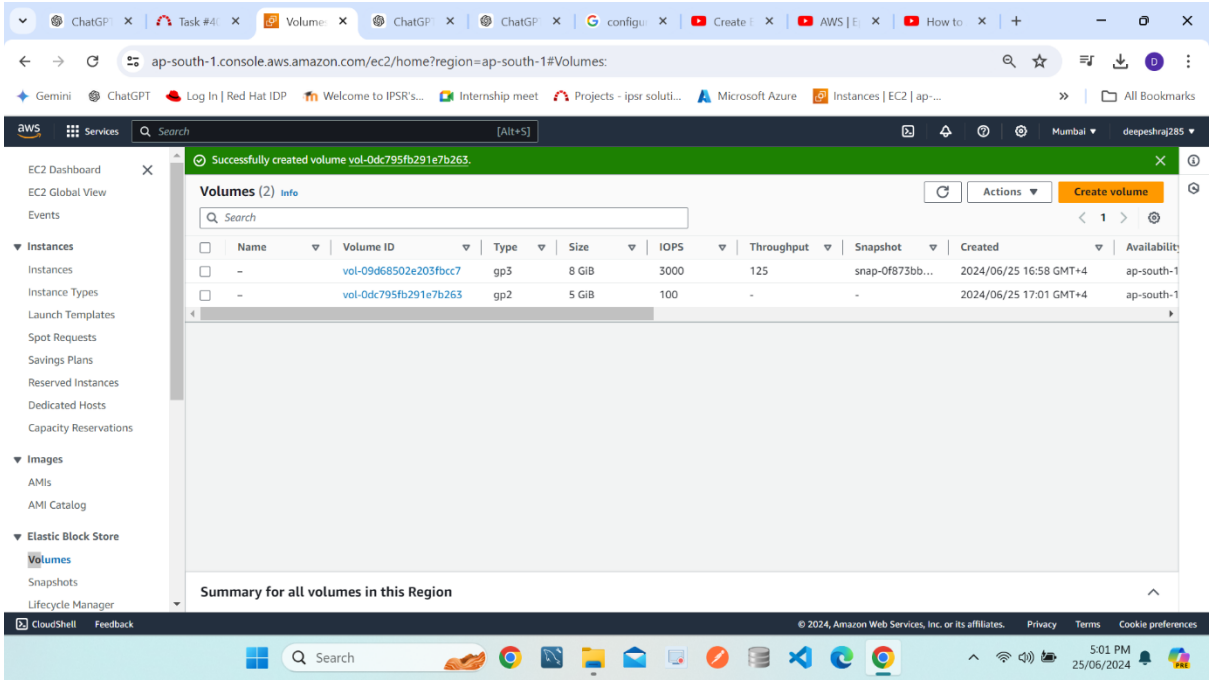
In summary, choose EBS for scenarios requiring high-performance, low-latency access to block storage that behaves like a physical disk attached to your instance. Choose EFS when

you need scalable, shared file storage accessible from multiple instances, where data consistency and concurrent access are important.

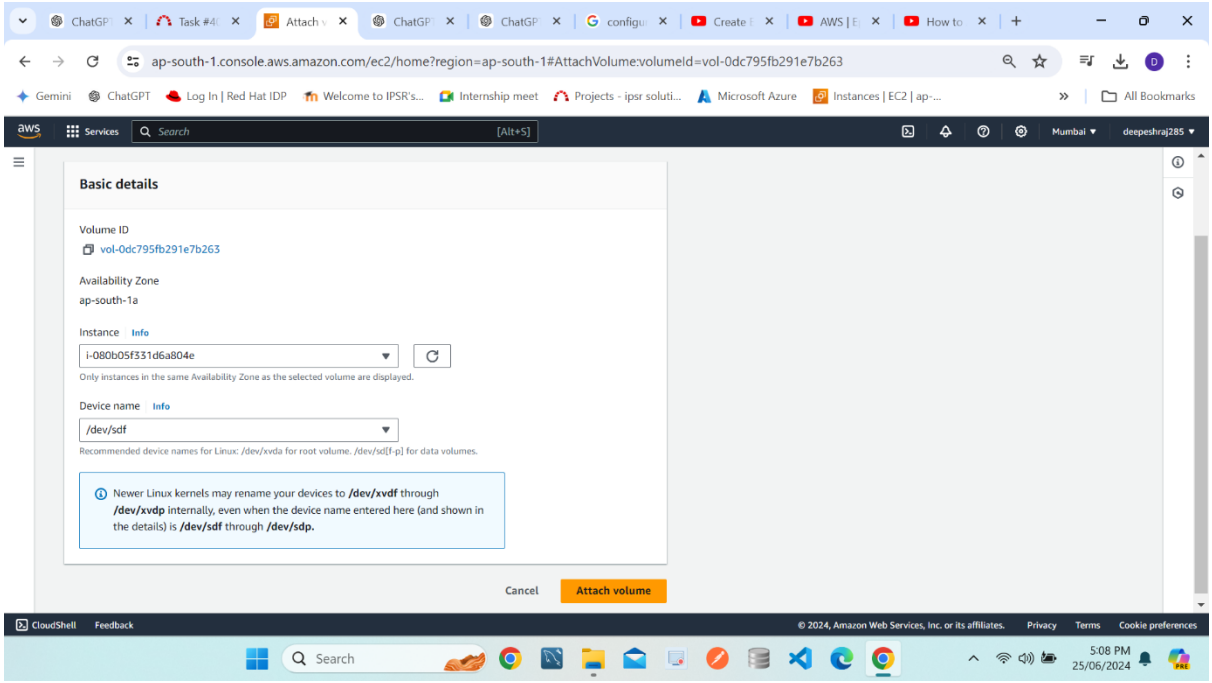
<u>EBS</u>

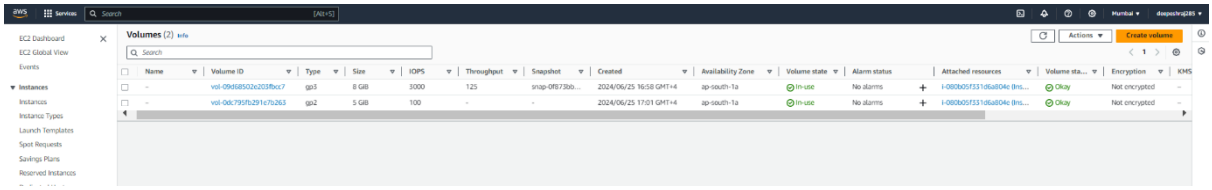
1. Create first AWS Ec2 instances on Linux machine
2. Create an EBS Volume
    a. Navigate to EC2 Dashboard.
    b. On the left-hand side, click on Volumes under the Elastic Block Store section.
    c. Click Create Volume.
    d. Configure the volume:
        i. Size: Choose an appropriate size (e.g., 5 GiB for the free tier).
        ii. Availability Zone: Choose the same availability zone as your EC2 instances.
        iii. Other settings can be left as default.
    e. Click Create Volume.





3. Attach EBS Volume to the First Instance
    a. Navigate to Instances in the EC2 Dashboard.
    b. Go back to Volumes, select the volume you created, and click Actions > Attach Volume.
    c. Choose your instance and click Attach.





4. Mount the EBS Volume on the First Instance
    a. Connect to the instance via SSH.
    b. List the available disk devices using lsblk to find your new volume (Here it is /dev/xvdf).
    c. Create a filesystem on the volume

       # sudo mkfs -t ext4 /dev/xvdf

```
[root@ip-172-31-40-147 ~]#
[root@ip-172-31-40-147 ~]# 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   5G  0 disk
[root@ip-172-31-40-147 ~]#
[root@ip-172-31-40-147 ~]#
[root@ip-172-31-40-147 ~]# mkfs -t ext4 /dev/xvdf
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 1310720 4k blocks and 327680 inodes
Filesystem UUID: 825e25e8-52d1-473a-b896-292f4d056176
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[root@ip-172-31-40-147 ~]#
```

    d. Create a mount point and mount the volume

       # mkdir /mnt/mydata

       # mount /dev/xvdf /mnt/mydata

    e. Write some data to the volume to test

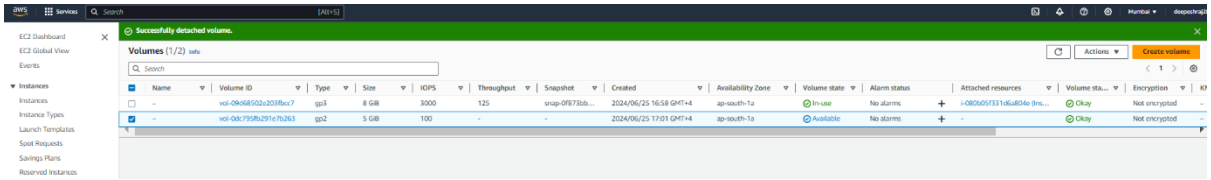
       # echo "Hello, This is the Instance1 EBS!" | sudo tee /mnt/mydata/test.txt

    f. Unmount the volume and detach it

       # umount /mnt/mydata

```
[root@ip-172-31-40-147 ~]# mkdir /mnt/mydata
[root@ip-172-31-40-147 ~]# mount /dev/xvdf /mnt/mydata
[root@ip-172-31-40-147 ~]#
[root@ip-172-31-40-147 ~]#
[root@ip-172-31-40-147 ~]#
[root@ip-172-31-40-147 ~]# echo "Hello,This is the Instance1 EBS!" | sudo tee /mnt/mydata/test.txt
Hello,This is the Instance1 EBS!
[root@ip-172-31-40-147 ~]#
[root@ip-172-31-40-147 ~]#
[root@ip-172-31-40-147 ~]#
[root@ip-172-31-40-147 ~]# umount /mnt/mydata
[root@ip-172-31-40-147 ~]#
```

    g. Stop the instance, go to Volumes, select your volume, and click Detach Volume.



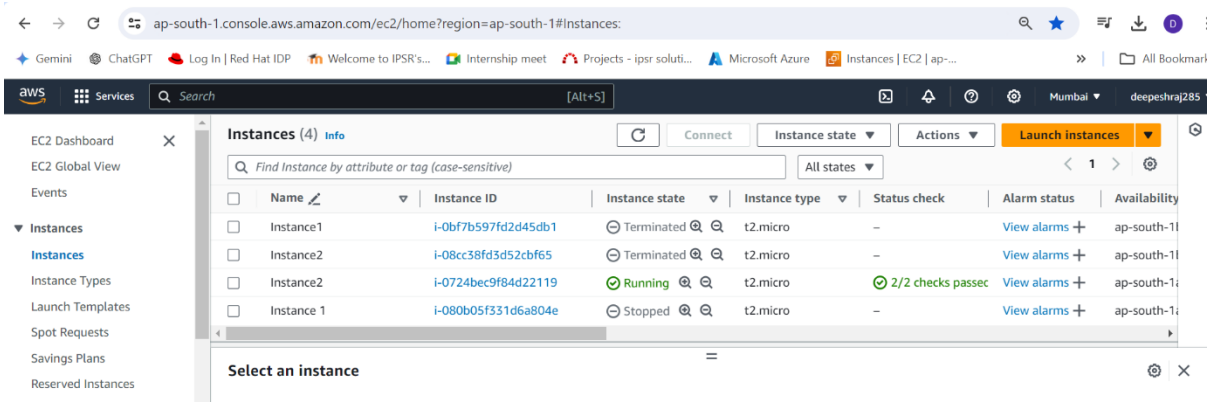
```
[root@ip-172-31-40-147 ~]# lsblk
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda      202:0    0   8G  0 disk
├─xvda1   202:1    0   8G  0 part /
├─xvda127 259:0    0   1M  0 part
└─xvda128 259:1    0  10M  0 part /boot/efi
[root@ip-172-31-40-147 ~]#
```

5. Create Second AWS EC2 Instance
6. Attach and Verify EBS Volume on the Second Instance
   a. Attach the EBS volume to the second instance following the steps in Step 3
   b. Start the second instance and connect via SSH.
   c. Mount the EBS volume on the second instance

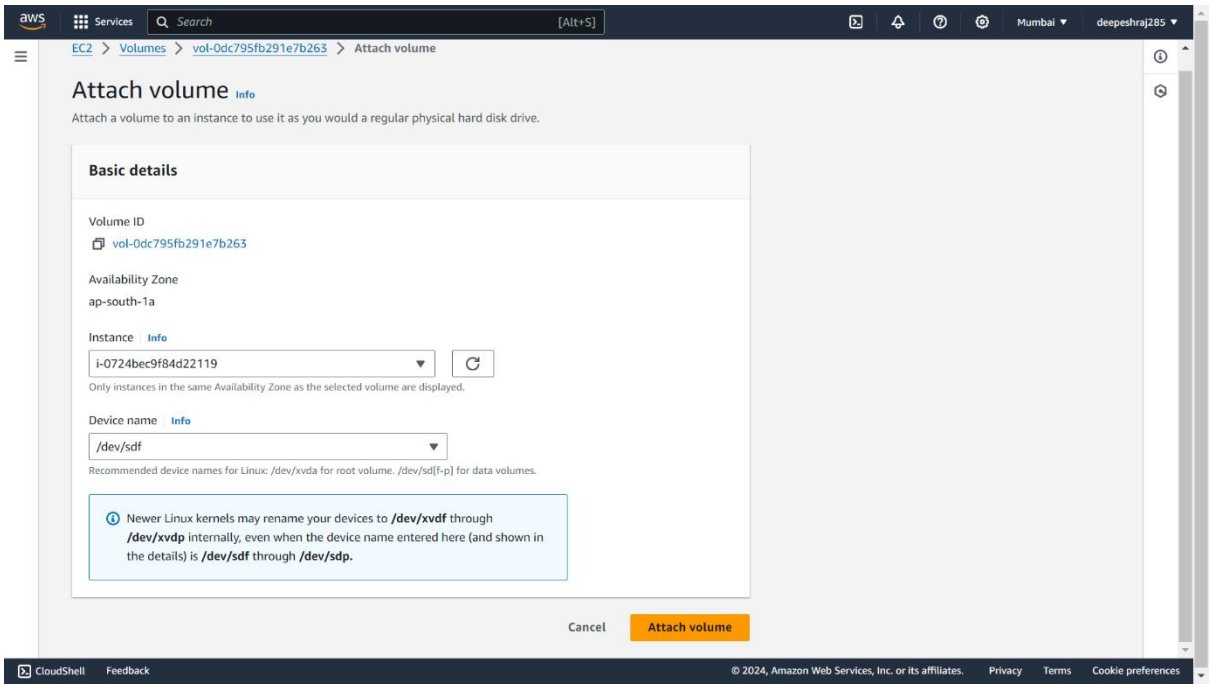
      # mkdir /mnt/mydata

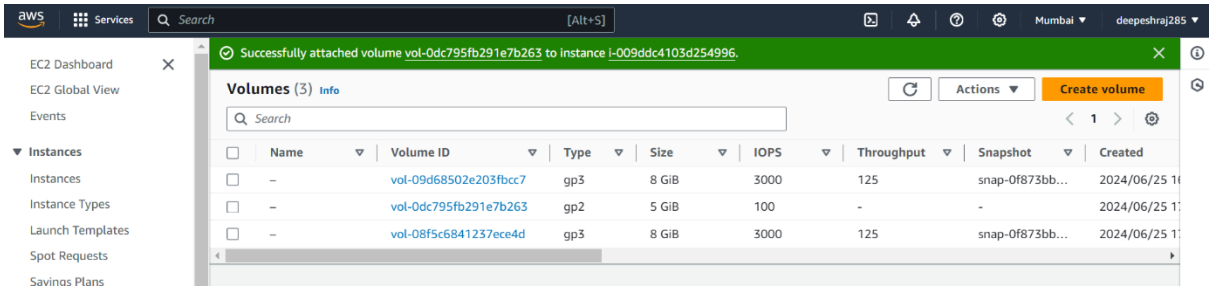
      # mount /dev/xvdf /mnt/mydata

Second EC2 Instance



Attach Volume





Mount EBS Volume to Second Instance

```
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]# mkdir /mnt/mydata
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]# 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   5G  0 disk
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]# mount /dev/xvdf /mnt/mydata
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
```

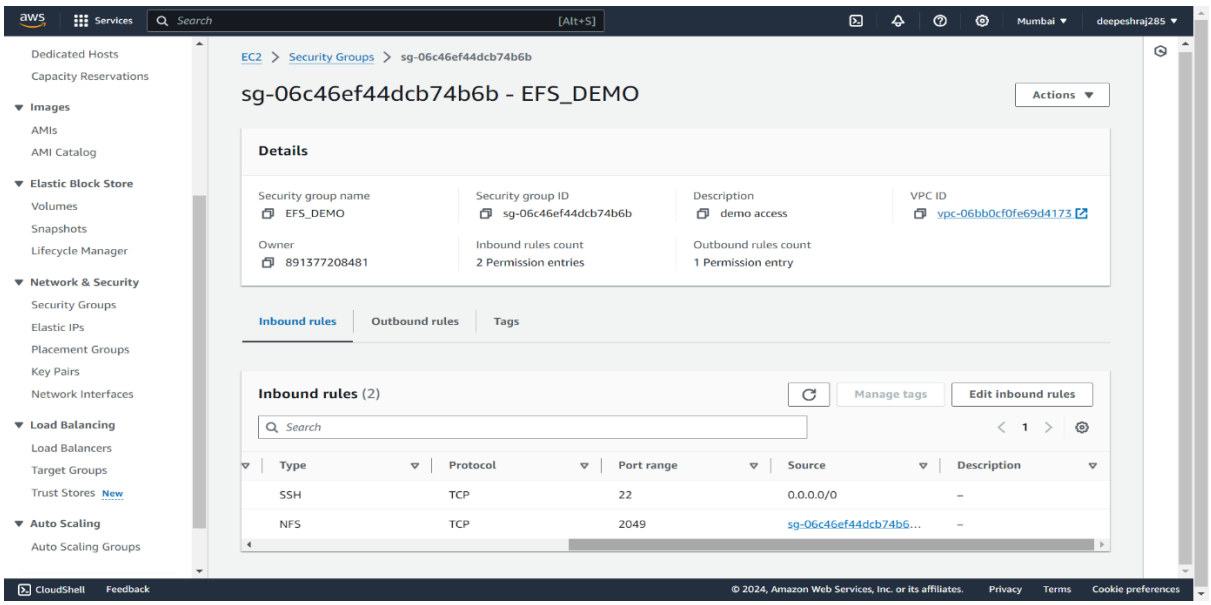
d. Verify the data

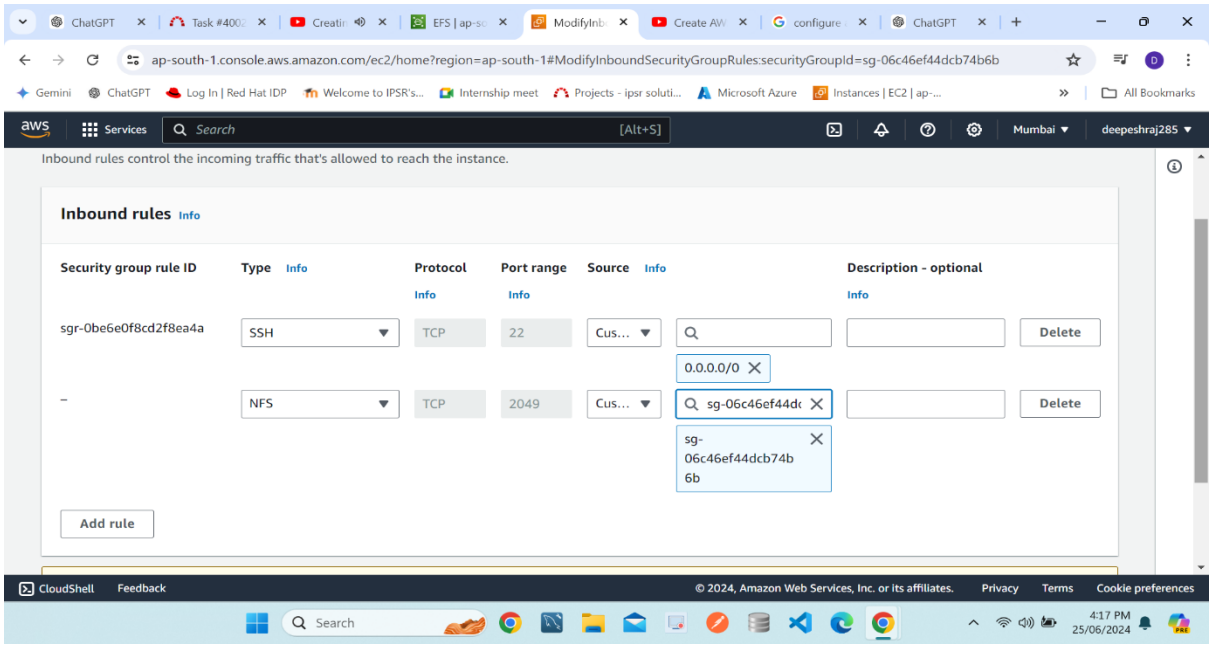
cat /mnt/mydata/test.txt

```
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]# df -Th
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  444K  190M   1% /run
/dev/xvda1     xfs       8.0G  1.6G  6.5G  20% /
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/xvdf      ext4      4.9G   28K  4.6G   1% /mnt/mydata
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]#
[root@ip-172-31-34-147 ~]# cd /mnt/mydata
[root@ip-172-31-34-147 mydata]# ls
lost+found  test.txt
[root@ip-172-31-34-147 mydata]# cat test.txt
Hello,This is the Instance1 EBS!
[root@ip-172-31-34-147 mydata]#
[root@ip-172-31-34-147 mydata]#
[root@ip-172-31-34-147 mydata]#
```

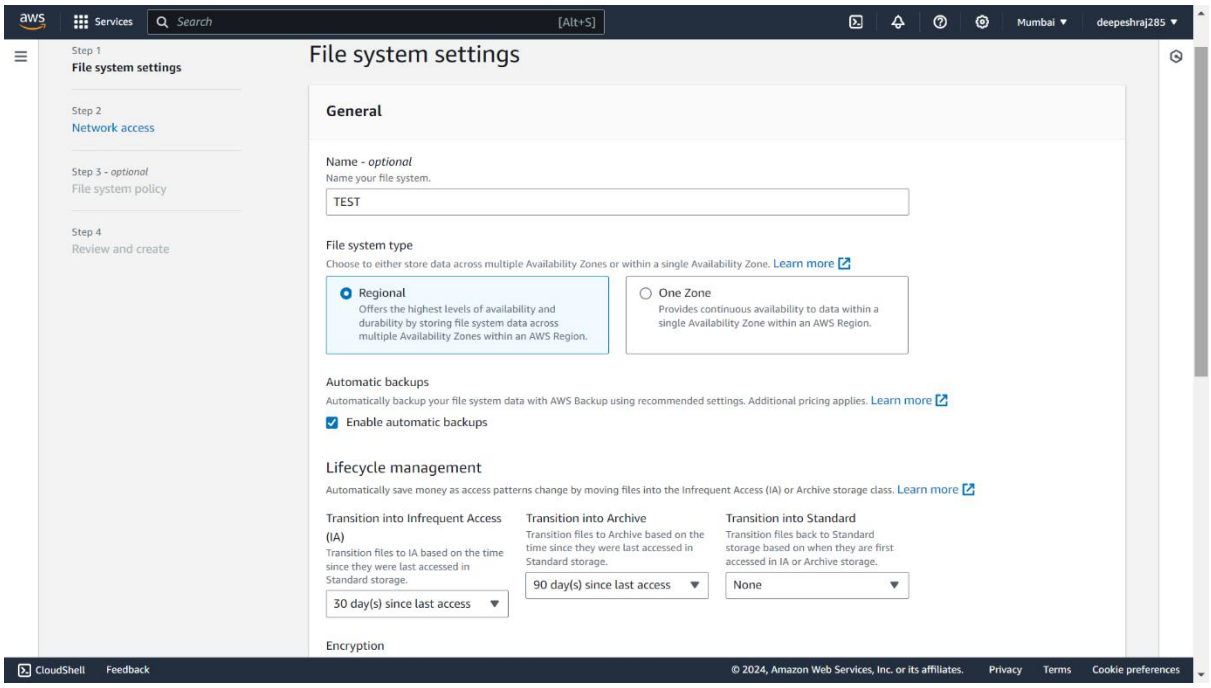
EFS

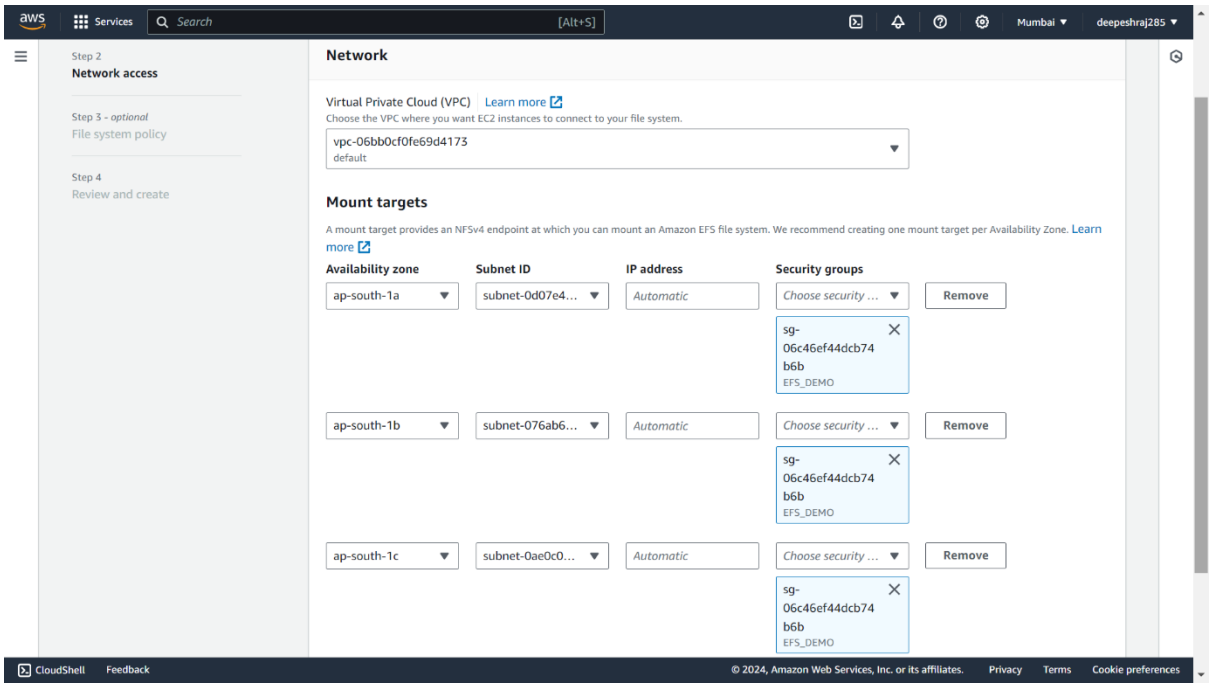
1. Create a Security Group for EFS
    a. Navigate to VPC Dashboard and click on Security Groups.
    b. Click Create Security Group and configure:
        i. Name: EFS-DEMO
        ii. Description: Security group for EFS
        iii. VPC: Choose the VPC your instances are in.
    c. Add inbound rules
        Type: NFS, Protocol: TCP, Port Range: 2049, Source: Custom, add the security groups of your EC2 instances.

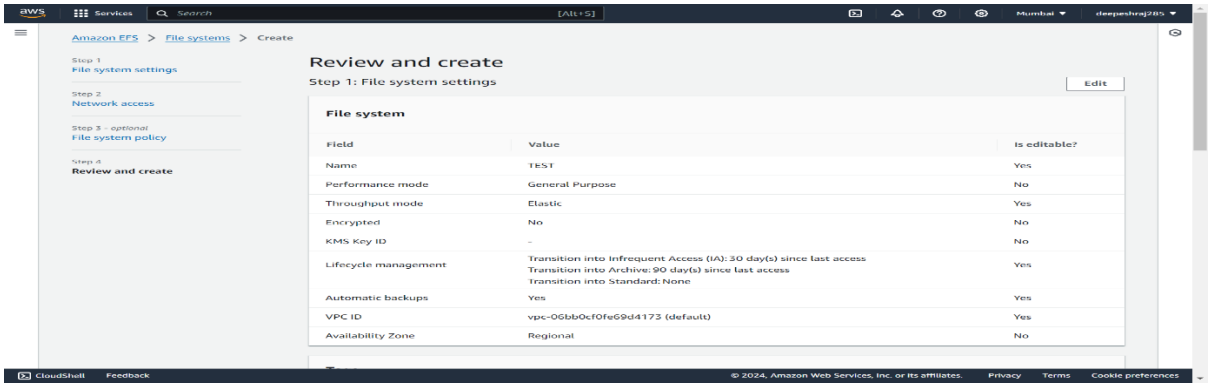


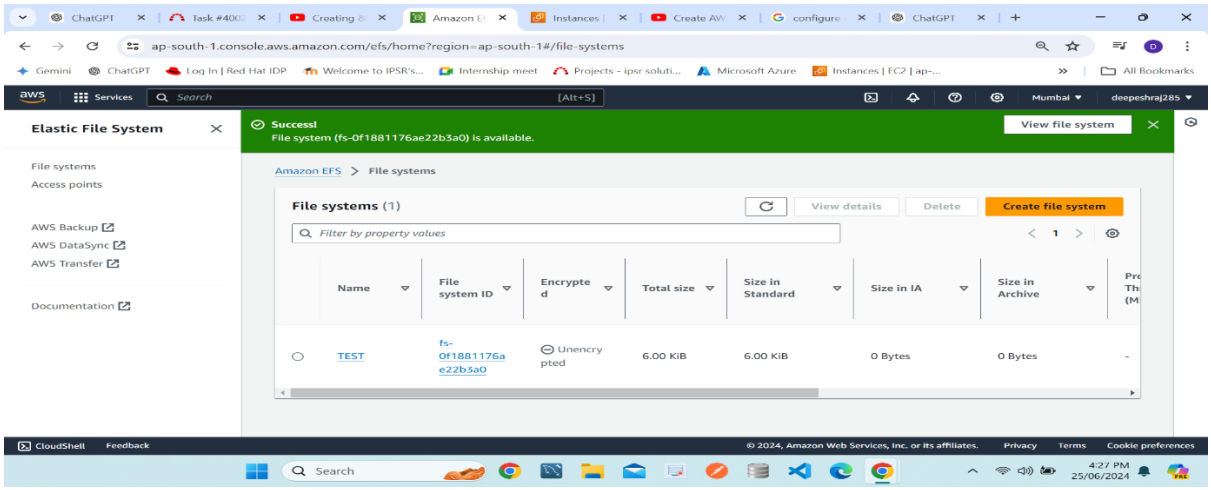


2. Create an EFS File System
    a. Navigate to EFS Dashboard.
    b. Click Create file system.
    c. Configure the file system
        - VPC: Choose your VPC.
        - Security Groups: Select the EFS-DEMO security group.
    d. Click Next and configure the mount targets.
    e. Click Next and review settings, then click Create.

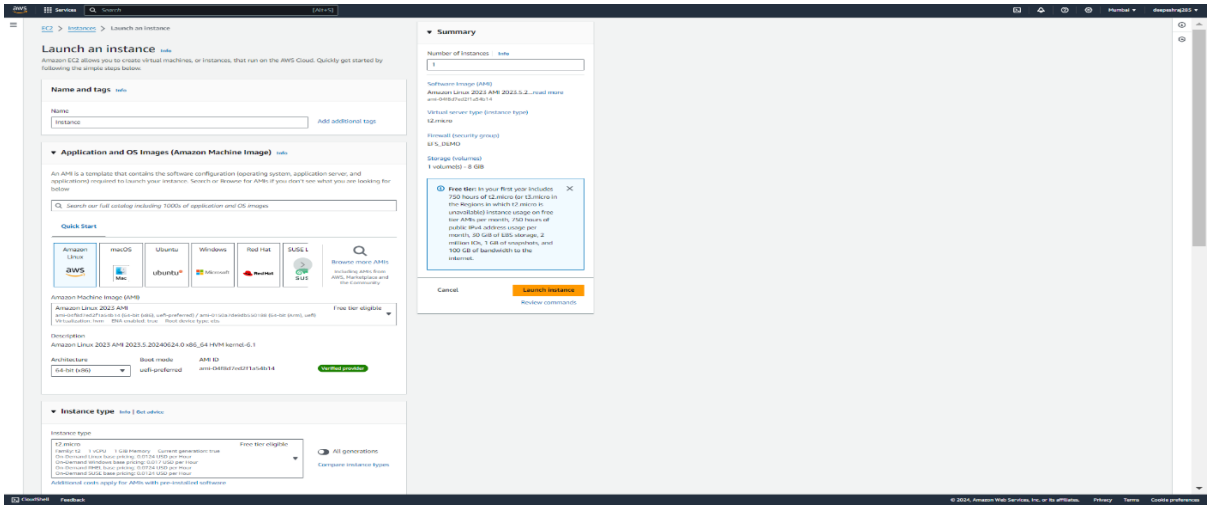


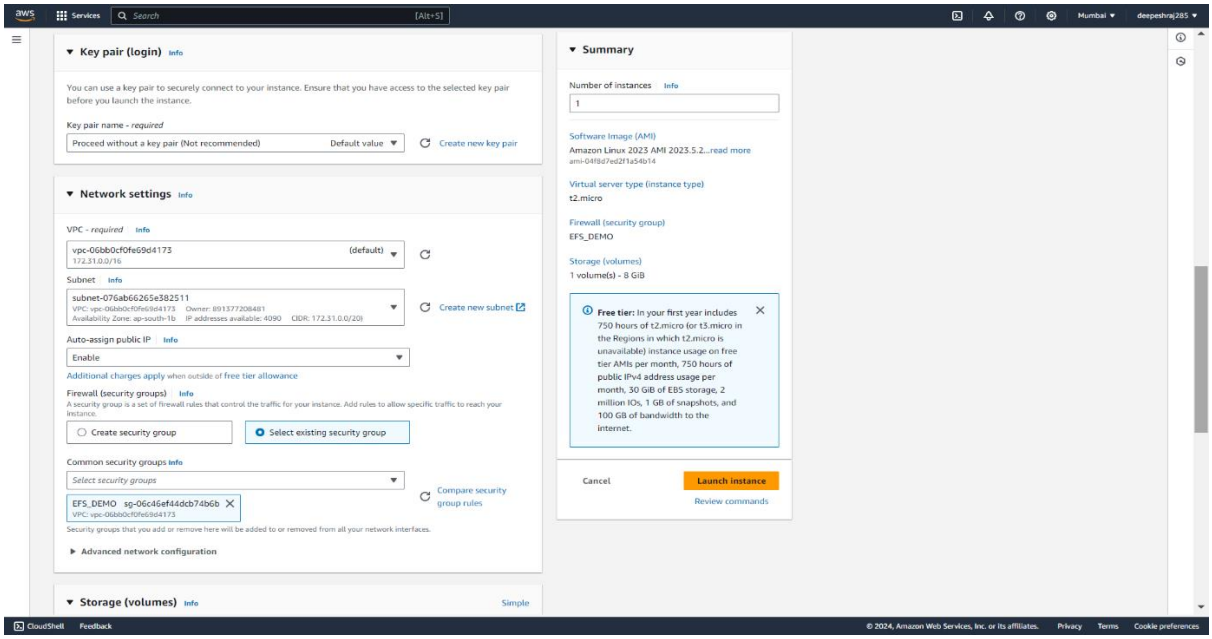


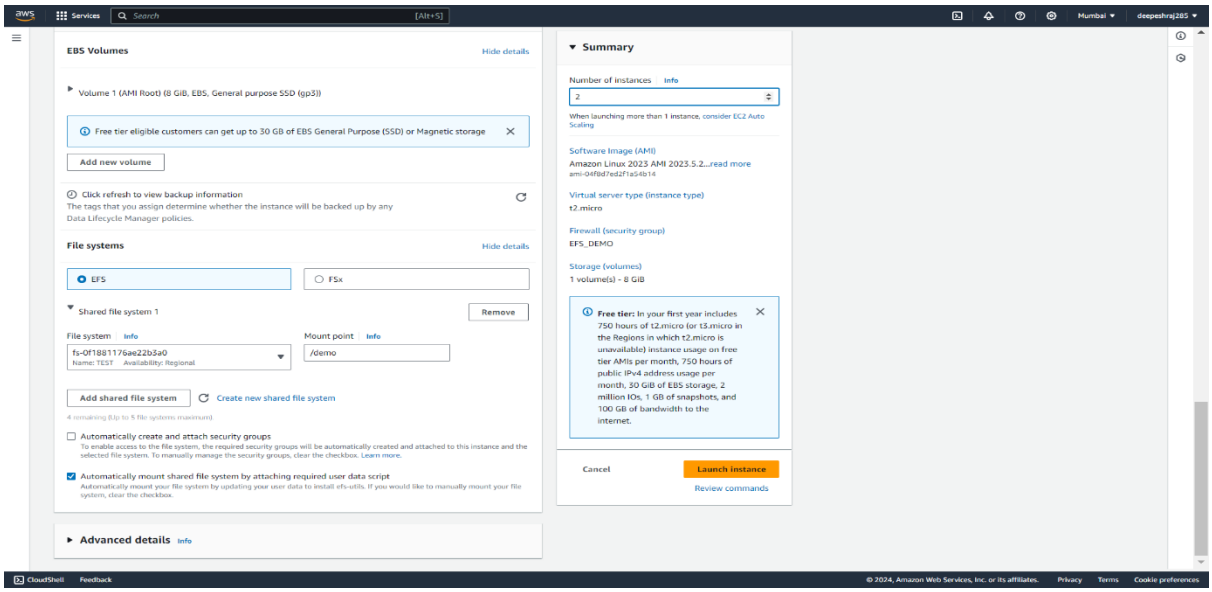




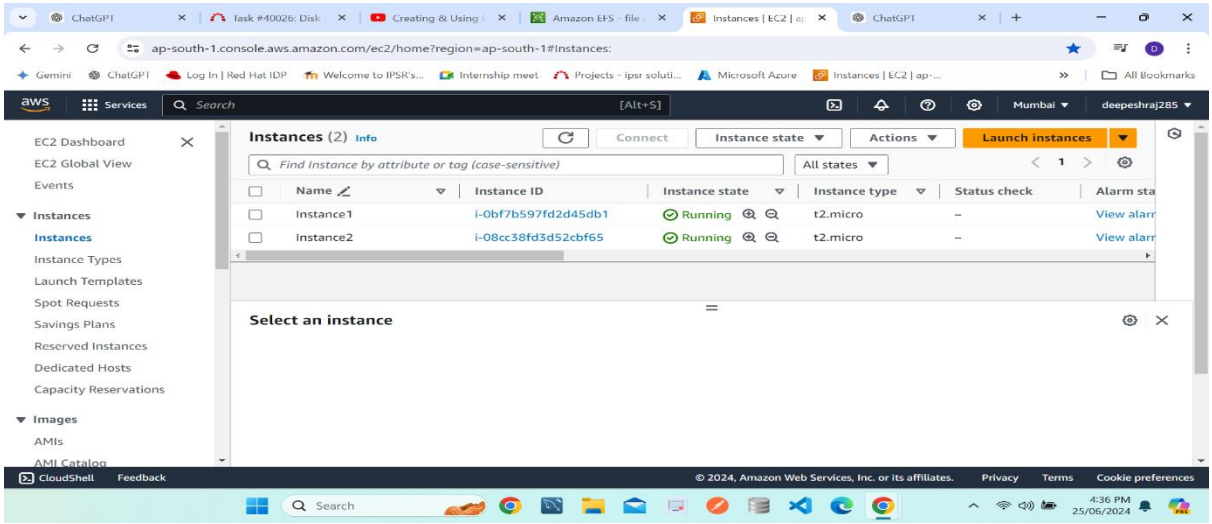
3. Create Two EC2 Instance
    a. Name the Ec2 Instance
    b. Choose the key pair
    c. Provide the network settings here choose the security group that we created first
    d. In EBS volume choose the file system EFS and choose the file system id we created already
    e. Type the mount point of the instance
    f. Choose two ec2 instances and launch it.



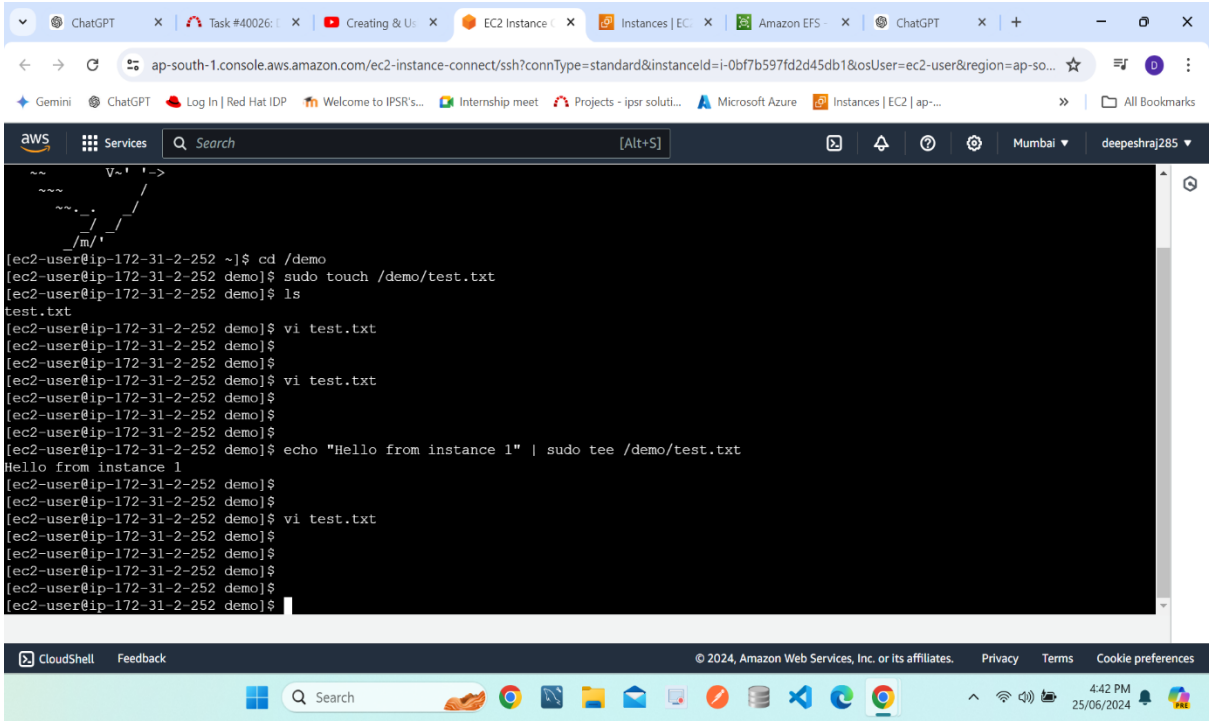




Lanunch the EC2 Instances



4. Connect the first instance and create a file in the mount point and enter data to the file



5. Connect the second instance and check the file which is in the mount point