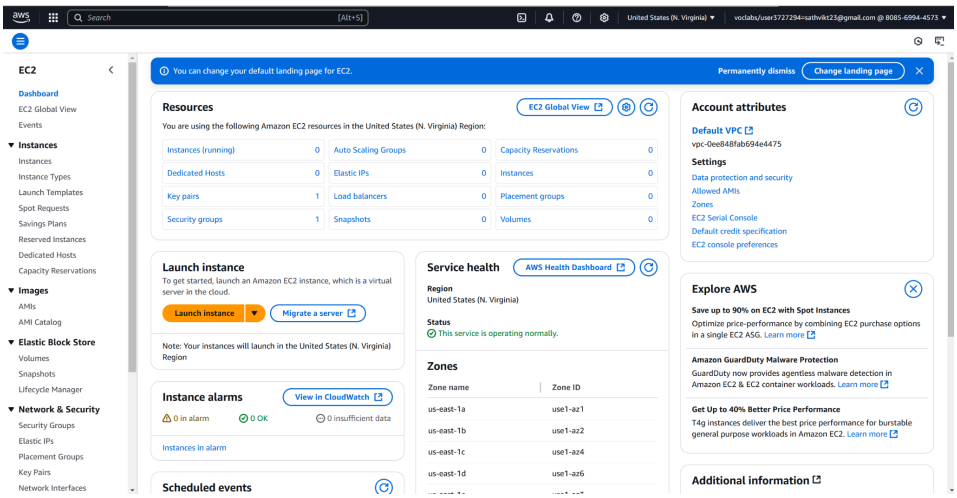
Experiment 4

Create and configure storage services and upload files and objects using Amazon EBS, Amazon EFS and Amazon S3

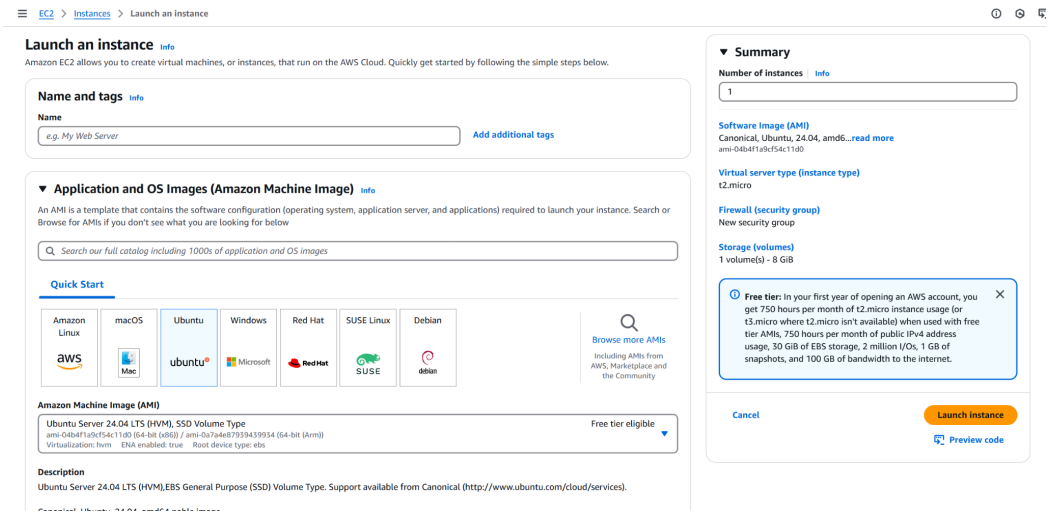
Part-1: Attach and Mount Extra EBS(Amazon Elastic Block Store) Volume to Linux EC2 in AWS

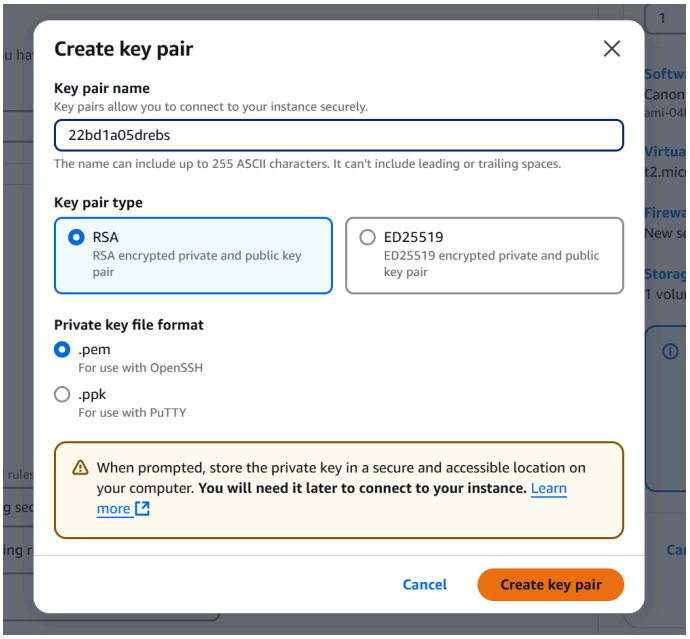
Step 1 : Go to the **EC2 section** and click on launch instances .



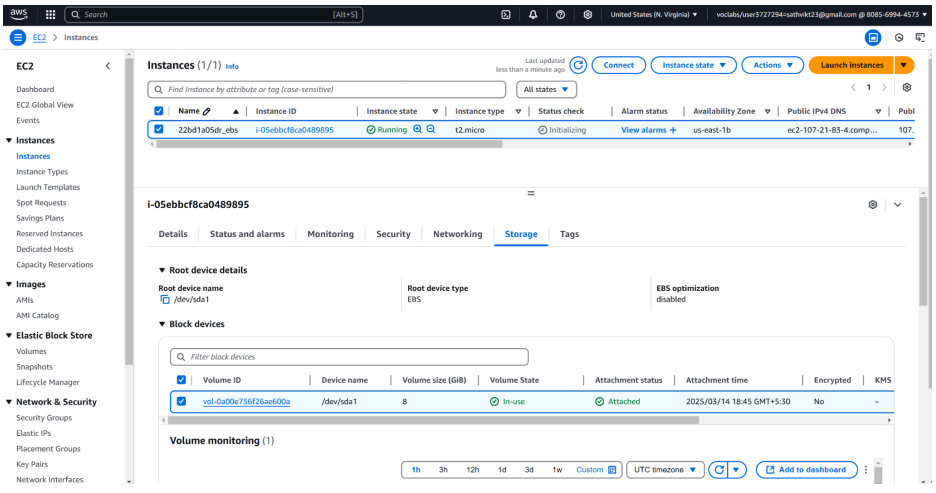
Step 2 : **Launch an instance**

- Select the OS as Ubuntu , type as t2.micro
- Allow all the Network permissions
- Create key value pair (.perm)
- Click on launch instance

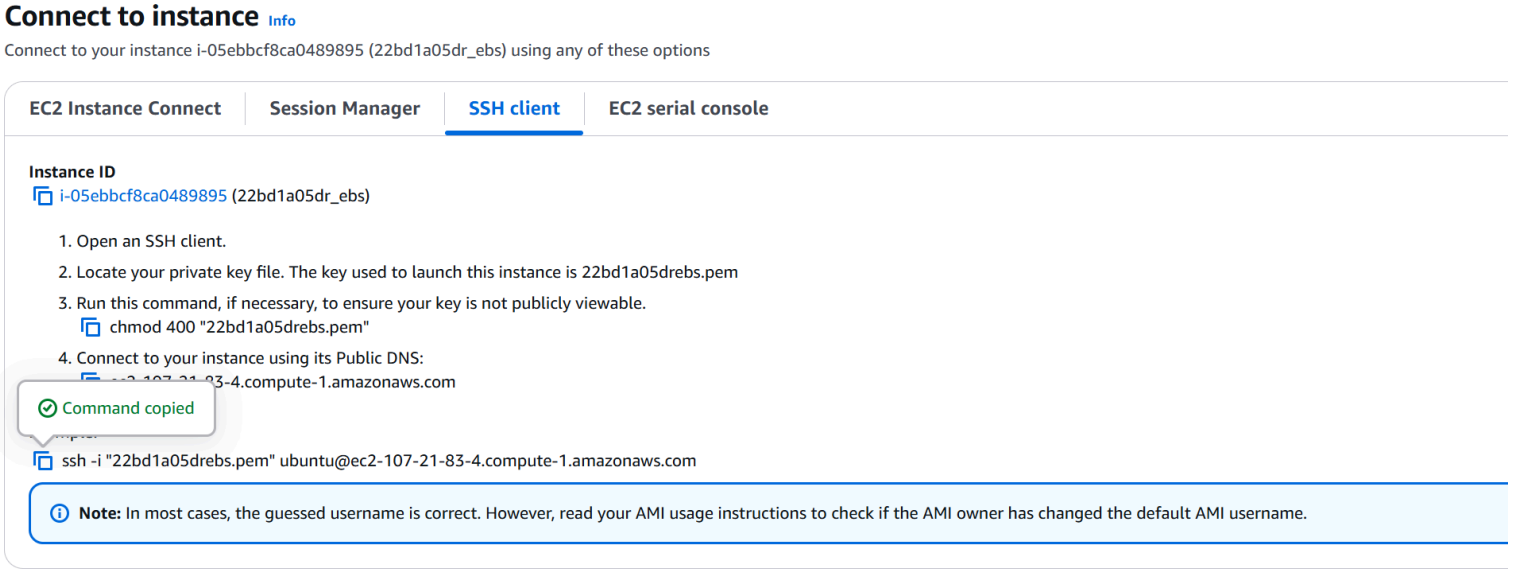




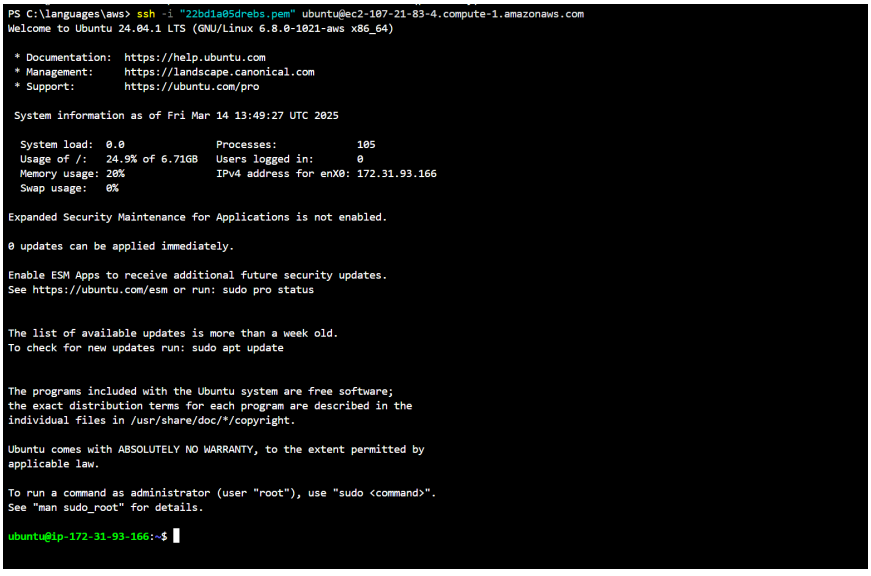
Step 3 Redirect to Instances dashboard



Step 4 : Connect to the instance by **SSH client**

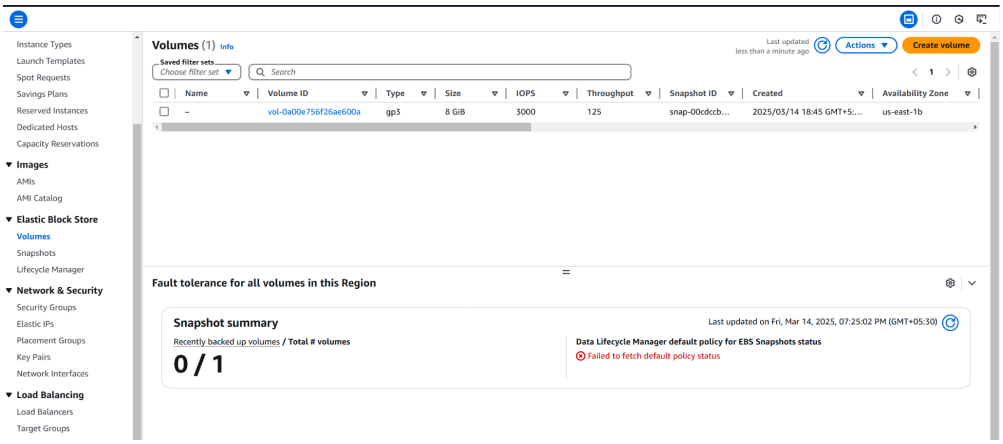


Step 5 Connect to **SSH client** in your terminal and run command **$lsblk** to verify the attached volumes



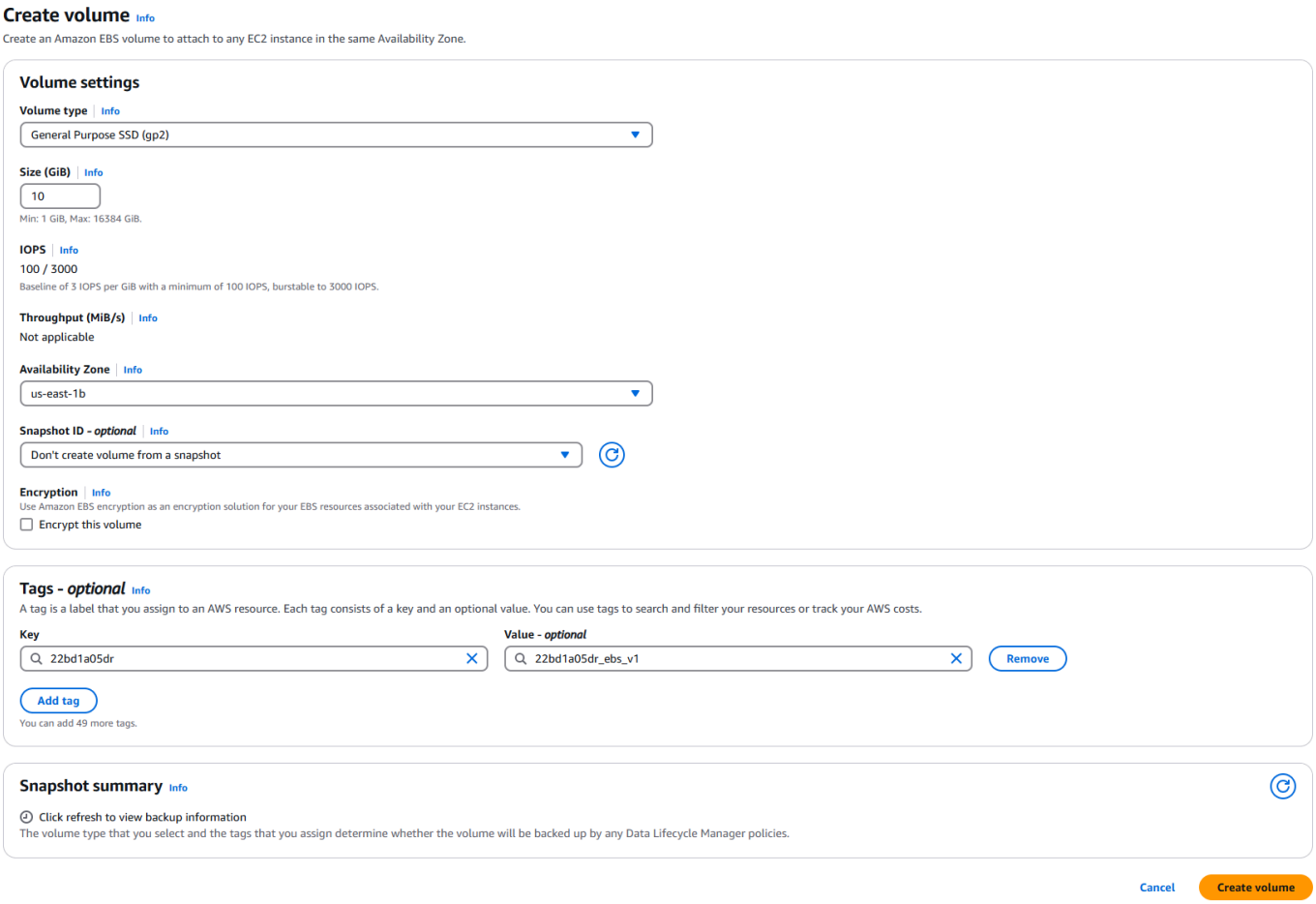
```
ubuntu@ip-172-31-93-166:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0       7:0    0 26.3M  1 loop /snap/amazon-ssm-agent/9881
loop1       7:1    0 44.4M  1 loop /snap/snapd/23545
loop2       7:2    0 73.9M  1 loop /snap/core22/1722
xvda      202:0    0    8G  0 disk
├─xvda1   202:1    0    7G  0 part /
├─xvda14  202:14   0    4M  0 part
├─xvda15  202:15   0  106M  0 part /boot/efi
└─xvda16  259:0    0  913M  0 part /boot
ubuntu@ip-172-31-93-166:~$
```

Step 6 : Go to volume section and Click on **Create volume**



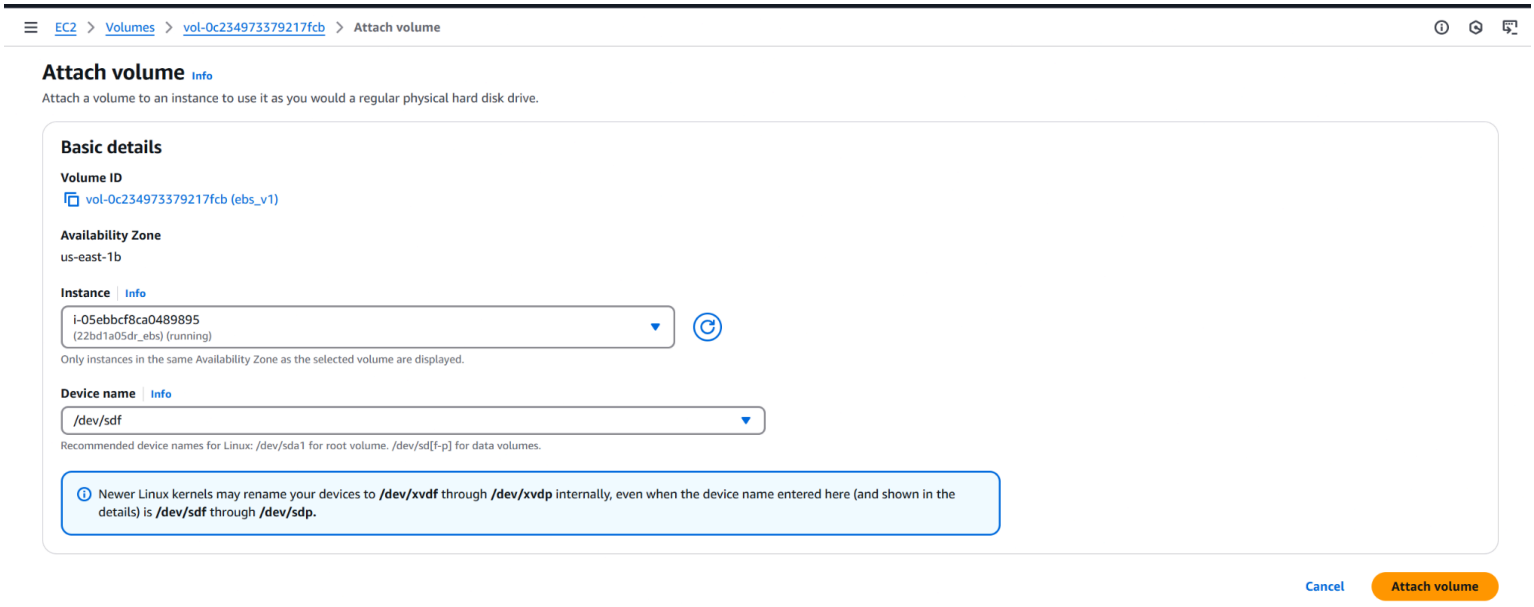
Step 7 : Enter the **Volume configurations**

- Volume type as gp2
- Enter tag details as required
- Then click on Create volume



Step 8 : **Click On Attach volume**

- Select the instance type and device name



Step 9 : Verify the Disk allocation and volume attachment by running the commands

- **$sudo fdisk -l**
- **$ sudo -s /dev/allocated_deviceName**

```
ubuntu@ip-172-31-93-166:~$ sudo fdisk -l
Disk /dev/loop0: 26.32 MiB, 27602944 bytes, 53912 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop1: 44.44 MiB, 46596096 bytes, 91008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop2: 73.87 MiB, 77459456 bytes, 151288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: D7B2CFD6-F96A-42FB-B8E1-13FF6555EB07

Device       Start      End  Sectors  Size Type
/dev/xvda1  2099200 16777182 14677983    7G Linux filesystem
/dev/xvda14    2048    10239     8192    4M BIOS boot
/dev/xvda15   10240   227327   217088  106M EFI System
/dev/xvda16  227328  2097152  1869825  913M Linux extended boot

Partition table entries are not in disk order.


Disk /dev/xvdf: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
ubuntu@ip-172-31-93-166:~$
```

```
ubuntu@ip-172-31-93-166:~$ sudo file -s /dev/xvdf
/dev/xvdf: data
ubuntu@ip-172-31-93-166:~$
```

Step 10 : Create a directory by running the command **$sudo mkdir dir_name**

```
ubuntu@ip-172-31-93-166:~$ sudo file -s /dev/xvdf
/dev/xvdf: data
ubuntu@ip-172-31-93-166:~$ sudo mkdir /myebsvol
```

Step 11 : Create a file system by running the command

$sudo mkfs -t /dev/allocated_name

```
ubuntu@ip-172-31-93-166:~$ sudo mkfs -t xfs /dev/xvdf
meta-data=/dev/xvdf              isize=512    agcount=4, agsize=655360 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=1
         =                       reflink=1    bigtime=1 inobtcount=1 nrext64=0
data     =                       bsize=4096   blocks=2621440, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=16384, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
ubuntu@ip-172-31-93-166:~$ sudo file -s /dev/xvdf
/dev/xvdf: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
```

Step 12 : Mount the volume to the created directory by running the command

$sudo mount /dev/allocated_name dir_name

```
ubuntu@ip-172-31-93-166:~$ sudo mount /dev/xvdf /myebsvol
```

Step 13 : Run the command **$ sudo df -h** to check the mounting status

```
ubuntu@ip-172-31-93-166:~$ sudo df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  1.7G  5.1G  26% /
tmpfs           479M     0  479M   0% /dev/shm
tmpfs           192M  880K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M   76M  744M  10% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
/dev/xvdf        10G  228M  9.8G   3% /myebsvol
ubuntu@ip-172-31-93-166:~$
```