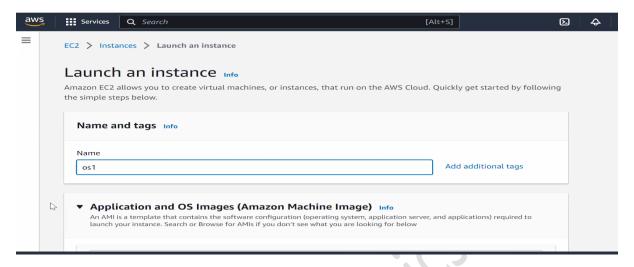**AWS Session**

**14-3-2023**

- If we want any data to be permanent then we use Storage Service.
- To need raw hard disk, we use block storage and for that in AWS we have EBS- Elastic Block Storage which is in STAAS- Storage as a service.
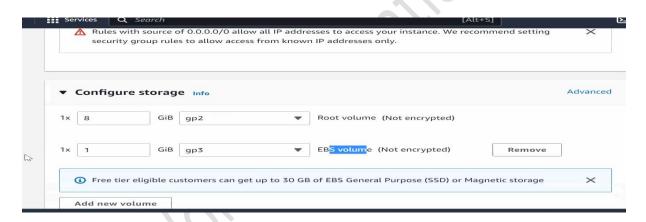  NOTE: Here Raw Disk includes pen-drive, external hard drive.
- Storage here is also known as Volume
- EBS- Elastic Block Storage is categorised in 3 storages:

  1. **Root Storage/ Root Volume**
  2. **EBS Volume/Network storage**
  3. **Instance Storage**

- In windows C-Drive is the root volume, so when the OS is deleted then our data will be loss.
- External Storage keeps the data persistent.
- Giving real Hard disk to OS, can be treated as external HD so we use instance store.
- EBS is comparatively slow and cheap whereas instance store is very fast. So if we have a requirement for high performance we use instance store but this will increase the cost also.
- In HD when we read and write then that is known as **I/O Operation** i.e., input/output operation.
- When we create any volume, internally hard disk is being created in EBS. It looks like it is a part of OS but internally everything is happening through network.
- Whenever we launch instance we select instance type like t2.micro that means behind the seen it has been given a dedicated hardware.

## PRACTICAL
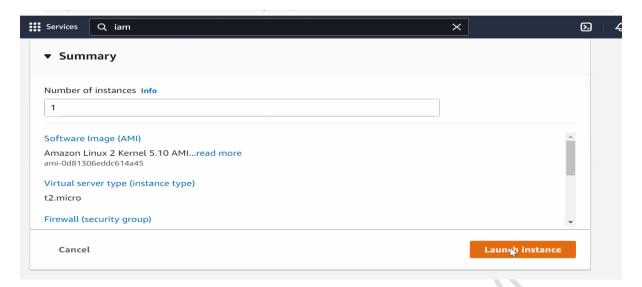
**Step-1:** Create EC2 instance





Configure aws through cli and see the instance types in ec2 so for that we used filter feature to find the instance types in ec2.
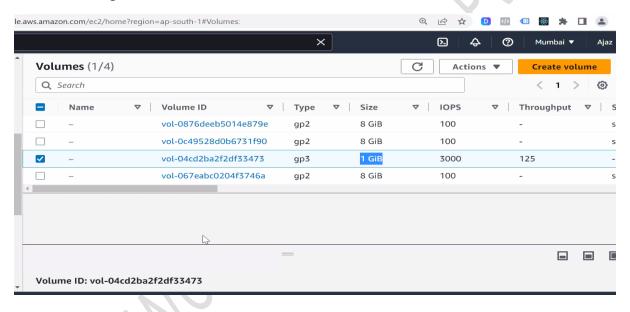
[AWS]





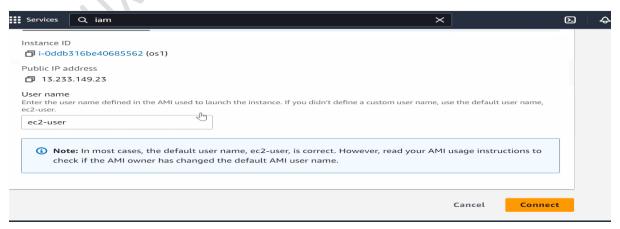If we want our output in tabular format then we used --output table in the command while filetring.

After creating instance volume will be created.



**Step-2)** Now connect with the instance.

After connecting the instance, we create partition, format it, and mount it.



```
                aws     ⠿⠿⠿ Services    Q  Search                                    [Alt+S]
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-32-52 ~]$ sudo su –
[root@ip-172-31-32-52 ~]# fdisk  -l
Disk /dev/xvda: 8 GiB, 8509934592 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: 740A12D4-3E3E-4A54-A989-6F6AB0688304

Device        Start      End  Sectors Size Type
/dev/xvda1     4096 16777182 16773087   8G Linux filesystem
/dev/xvda128   2048     4095     2048   1M BIOS boot

Partition table entries are not in disk order.


Disk /dev/xvdb: 1 GiB, 1073741824 bytes, 2097152 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
[root@ip-172-31-32-52 ~]#
```



```
[root@ip-172-31-32-52 ~]# df
Filesystem      1K-blocks    Used Available Use% Mounted on
devtmpfs           485168       0    485168   0% /dev
tmpfs              493956       0    493956   0% /dev/shm
tmpfs              493956     412    493544   1% /run
tmpfs              493956       0    493956   0% /sys/fs/cgroup
/dev/xvda1        8376300 1664884   6711416  20% /
tmpfs               98792       0     98792   0% /run/user/1000
[root@ip-172-31-32-52 ~]#
```



```
[root@ip-172-31-32-52 ~]#
[root@ip-172-31-32-52 ~]# lsblk
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda      202:0    0   8G  0 disk
└─xvda1   202:1    0   8G  0 part /
xvdb      202:16   0   1G  0 disk
[root@ip-172-31-32-52 ~]# fdisk  /dev/xvdb

Welcome to fdisk (util-linux 2.30.2).
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 0x078dff3d.

Command (m for help):
```

```
Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x078dff3d.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-2097151, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097151, default 2097151): +

Created a new partition 1 of type 'Linux' and of size 100 MiB.

Command (m for help):
```

```
Using default response p.
Partition number (1-4, default 1):
First sector (2048-2097151, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097151, default 2097151): +100M

Created a new partition 1 of type 'Linux' and of size 100 MiB.

Command (m for help): p
Disk /dev/xvdb: 1 GiB, 1073741824 bytes, 2097152 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: dos
Disk identifier: 0x078dff3d

Device     Boot Start    End Sectors  Size Id Type
/dev/xvdb1       2048 206847  204800  100M 83 Linux

Command (m for help):
```

```
Command (m for help): p
Disk /dev/xvdb: 1 GiB, 1073741824 bytes, 2097152 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: dos
Disk identifier: 0x078dff3d

Device     Boot Start    End Sectors  Size Id Type
/dev/xvdb1       2048 206847  204800  100M 83 Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[root@ip-172-31-32-52 ~]#
```

```
[root@ip-172-31-32-52 ~]# mkfs.ext4    /dev/xvdb1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
25688 inodes, 102400 blocks
5120 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=33685504
13 block groups
8192 blocks per group, 8192 fragments per group
1976 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

[root@ip-172-31-32-52 ~]#
```
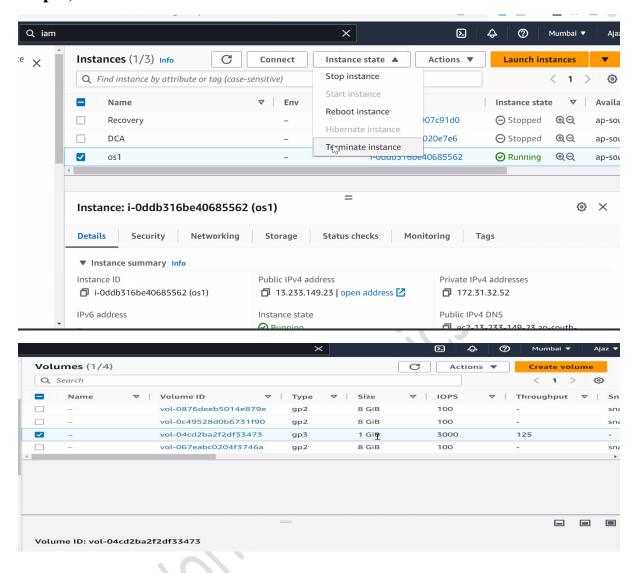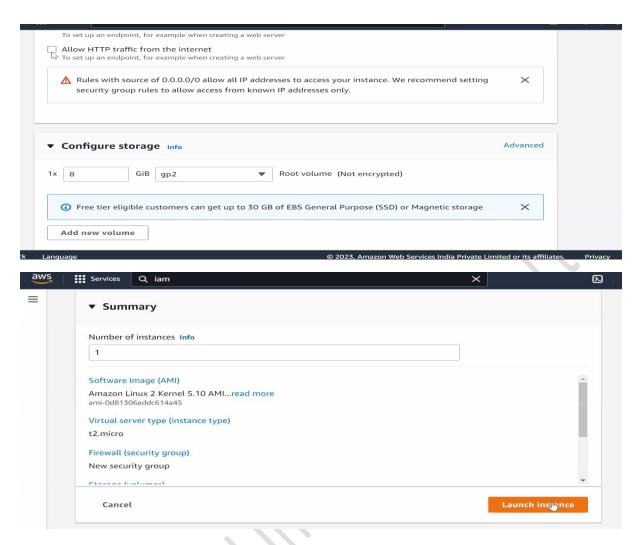
```
aws   Services    Q  Search                        [Alt+S]
[root@ip-172-31-32-52 ~]# mkdir /mydrive
[root@ip-172-31-32-52 ~]# mount /dev/xvdb1    /mydrive/
[root@ip-172-31-32-52 ~]#
[root@ip-172-31-32-52 ~]#
[root@ip-172-31-32-52 ~]# history
    1  fdisk   -l
    2  df
    3  lsblk
    4  fdisk  /dev/xvdb
    5  mkfs.ext4    /dev/xvdb1
    6  mkdir /mydrive
    7  mount /dev/xvdb1    /mydrive/
    8  history
[root@ip-172-31-32-52 ~]# lsblk
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda     202:0    0    8G  0 disk
└─xvda1 202:1    0    8G  0 part /
xvdb     202:16   0    1G  0 disk
└─xvdb1 202:17   0  100M  0 part /mydrive
[root@ip-172-31-32-52 ~]# cd /mydrive/
[root@ip-172-31-32-52 mydrive]# ls
lost+found
[root@ip-172-31-32-52 mydrive]# cat > linuxworld.txt
i m vimal , i located in EBS
[root@ip-172-31-32-52 mydrive]# p
```

```
aws   Services    Q  Search                        [Alt+S]
    2  df
    3  lsblk
    4  fdisk  /dev/xvdb
    5  mkfs.ext4    /dev/xvdb1
    6  mkdir /mydrive
    7  mount /dev/xvdb1    /mydrive/
    8  history
[root@ip-172-31-32-52 ~]# lsblk
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda     202:0    0    8G  0 disk
└─xvda1 202:1    0    8G  0 part /
xvdb     202:16   0    1G  0 disk
└─xvdb1 202:17   0  100M  0 part /mydrive
[root@ip-172-31-32-52 ~]# cd /mydrive/
[root@ip-172-31-32-52 mydrive]# ls
lost+found
[root@ip-172-31-32-52 mydrive]# cat > linuxworld.txt
i m vimal , i located in EBS
[root@ip-172-31-32-52 mydrive]# pwd
/mydrive
[root@ip-172-31-32-52 mydrive]# ls
linuxworld.txt   lost+found
[root@ip-172-31-32-52 mydrive]# cat linuxworld.txt
i m vimal , i located in EBS
[root@ip-172-31-32-52 mydrive]#
```
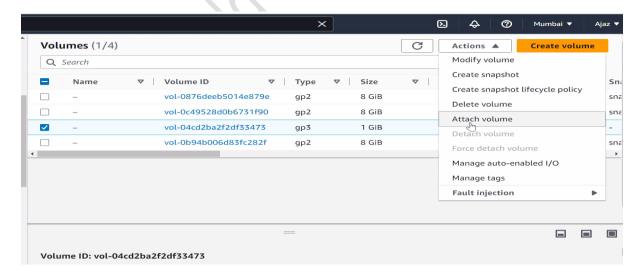
## Step-3) Now terminate that instance.





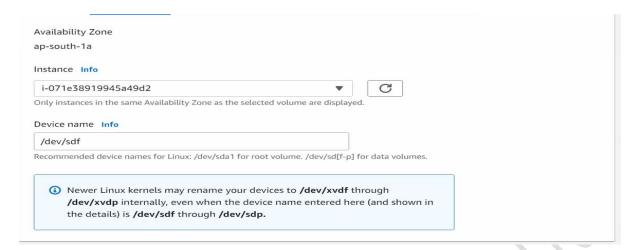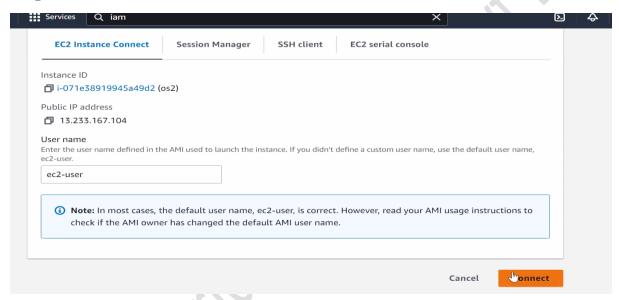## Step-4) Now launch one new instance.

Now attach that volume to the new created instance.

**Step-5)** Now connect to the instance.



**Step-6)** After connecting that instance and mount it to a folder and in that folder we will able to see the data which we added.