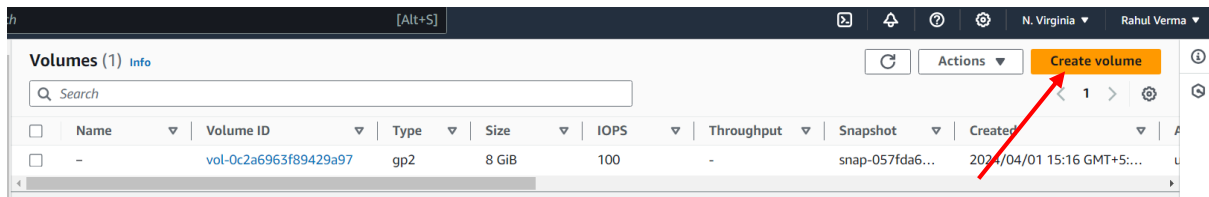


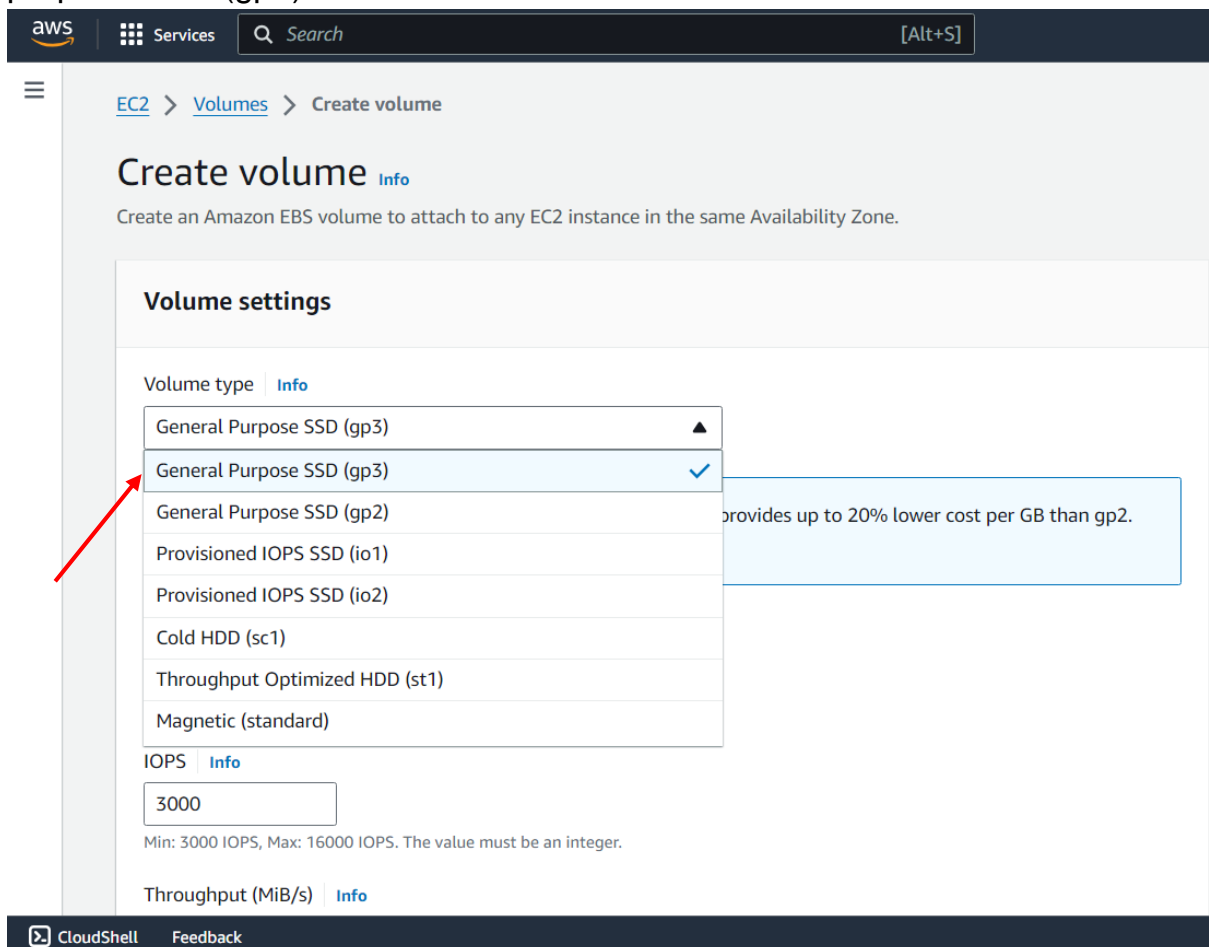
# Prerequisite: Create an Ec2 Instance in your AWS console

## NOW TO ADD ELASTIC BLOCK STORE

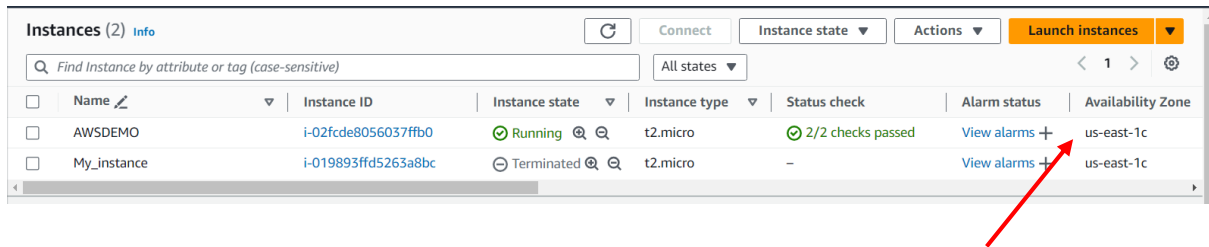
**Step 1:** Go to Volumes under Elastic Block Store and click on Create Volume button



**Step 2:** Select Volume type for demo will go with default one which is general purpose SSD (gp3)

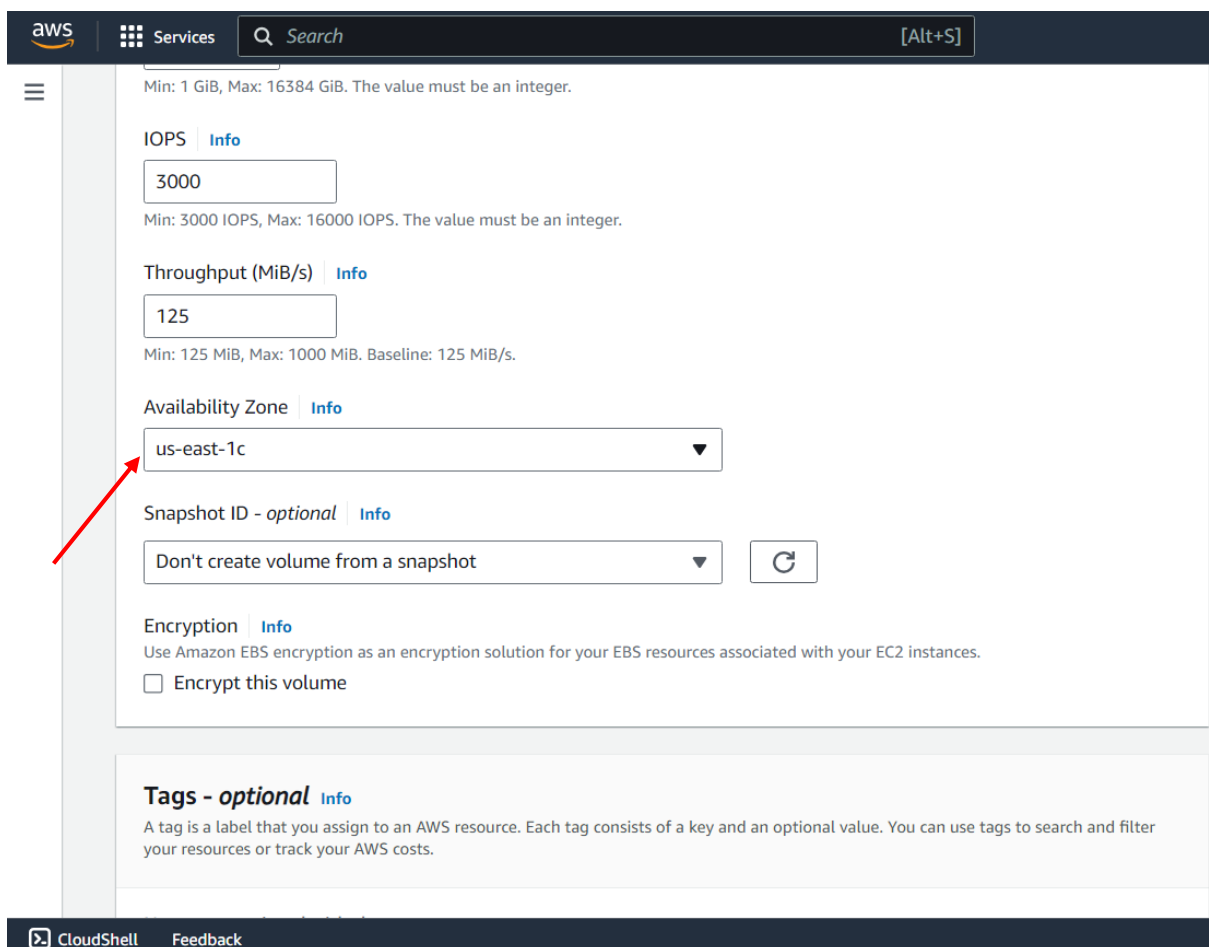


Now you need to check in which availability zone your instances are created on that Az only we have to create our EBS volume so that we can attach it.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	AWSDemo	i-02fcde8056037ffb0	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
<input type="checkbox"/>	My_instance	i-019893ffd5263a8bc	Terminated	t2.micro	-	<a href="#">View alarms</a>	us-east-1c

Our instance is created under **us-east-1c** availability zone so will select the same.



Min: 1 GiB, Max: 16384 GiB. The value must be an integer.

**IOPS** [Info](#)

3000

Min: 3000 IOPS, Max: 16000 IOPS. The value must be an integer.

**Throughput (MiB/s)** [Info](#)

125

Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s.

**Availability Zone** [Info](#)

us-east-1c

**Snapshot ID - optional** [Info](#)

Don't create volume from a snapshot

**Encryption** [Info](#)

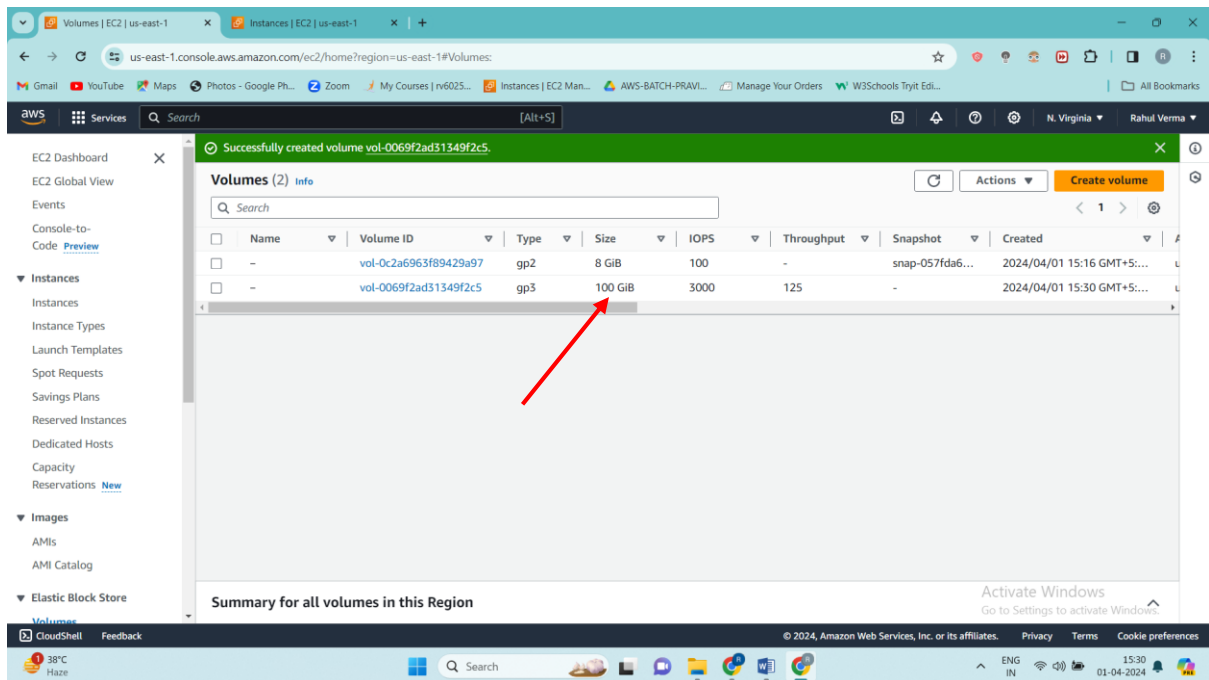
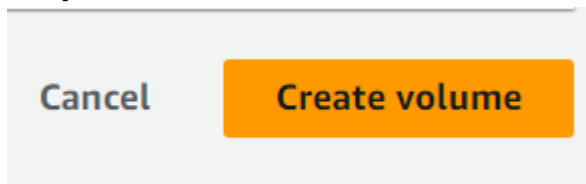
Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.

☐ Encrypt this volume

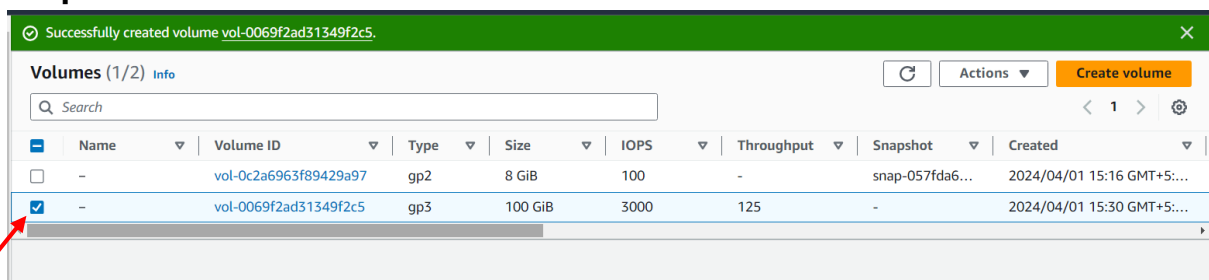
**Tags - optional** [Info](#)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

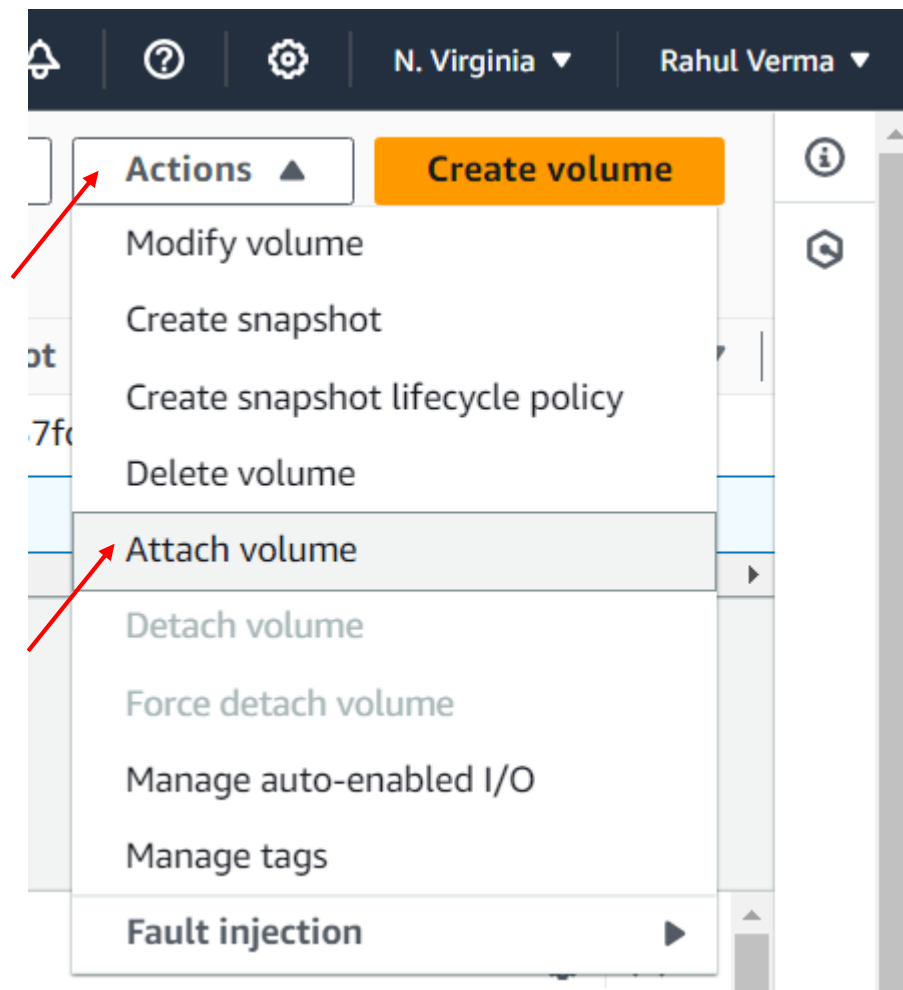
**Step 3:** And click on create volume and it will create our 100gb EBS volume



**Step 4:** Now select that volume



Go to actions and click on Attach volume



**Step 5:** Attach it to your instance im selecting “AWSDEMO” Instance which is available in useast-1c and define drive name such as- /dev/xvdbz or you can select from recommendation also.

**Attach volume** [Info](#)

Attach a volume to an instance to use it as you would a regular physical hard disk drive.

**Basic details**

Volume ID  
vol-0069f2ad31349f2c5

Availability Zone  
us-east-1c

Instance [Info](#)

Search instance ID or name tag

i-02fcde8056037ffb0  
(AWSDEMO) (running)

Select a device name

i-02fcde8056037ffb0

Cancel **Attach volume**

**Basic details**

Volume ID  
vol-0069f2ad31349f2c5

Availability Zone  
us-east-1c

Instance [Info](#)

i-02fcde8056037ffb0

Only instances in the same Availability Zone as the selected volume are displayed.

Device name [Info](#)

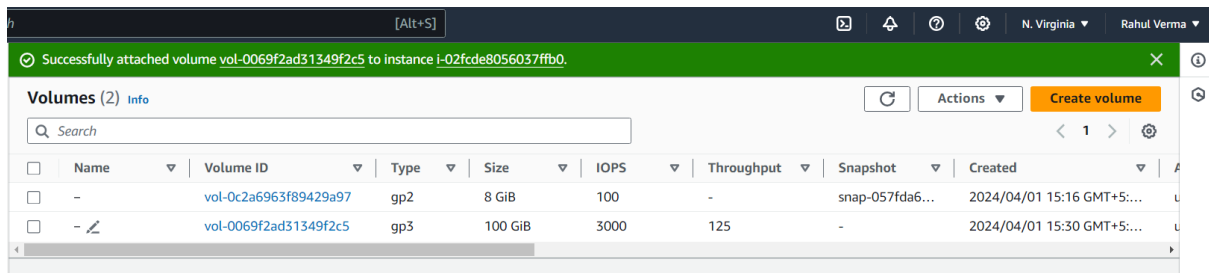
/dev/xvdbz

Recommended device names for Linux: /dev/sda1 for root volume. /dev/sd[f-p] for data volumes.

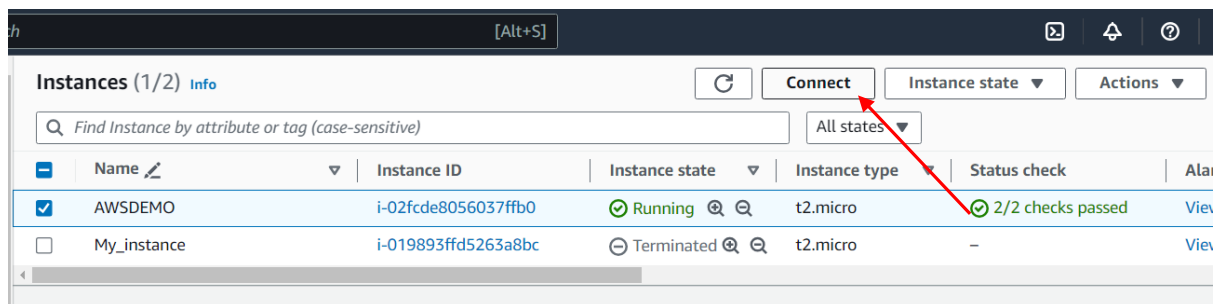
**Info** Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Cancel **Attach volume**

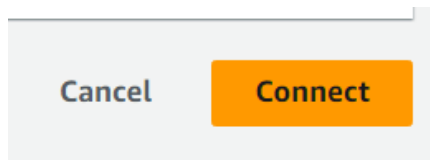
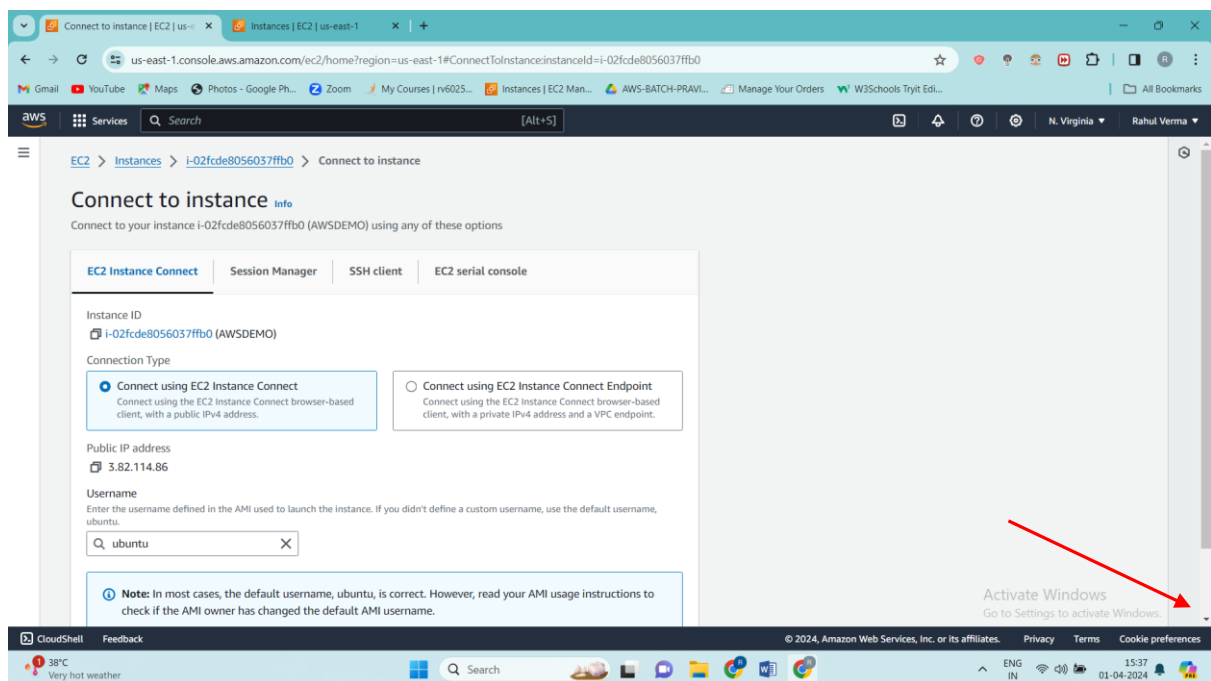
And it's attached now



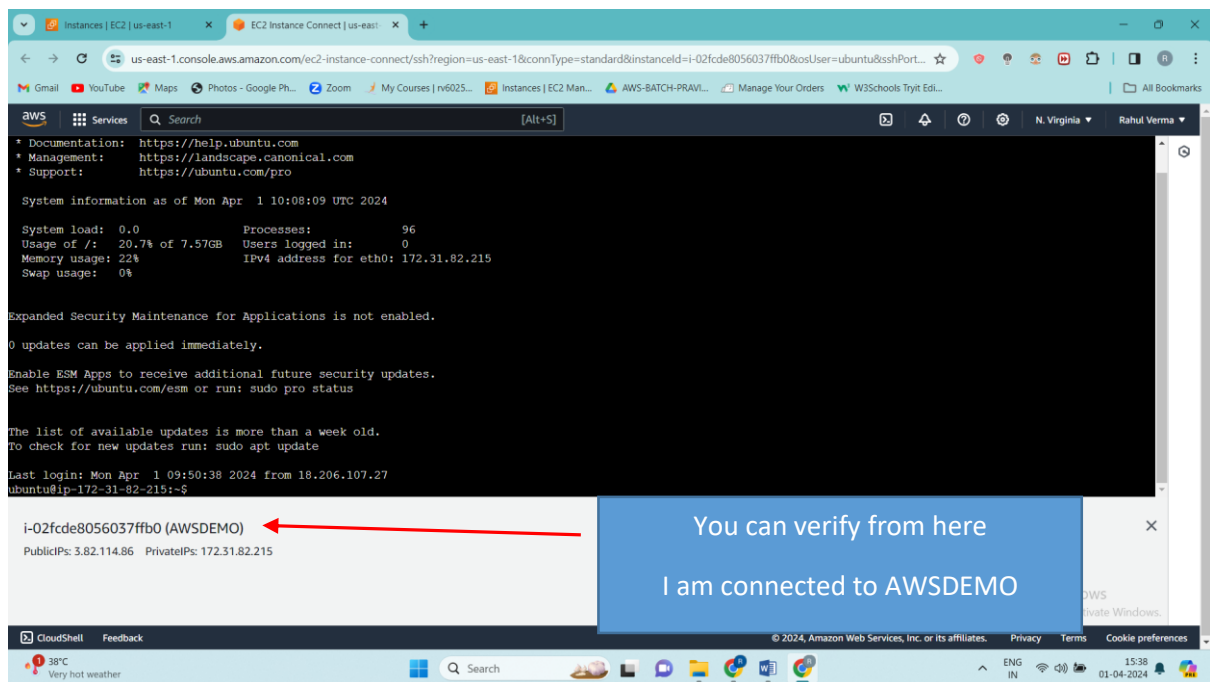
**Step 6:** Now to check if it's attached or not let's connect to our EC2 Instance. So go back to EC2 and select the instance in which you attached you EBS volume. And click on connect



Scroll down and click on connect



We are connected to our instance now



### Step 7:

Please **NOTE**: whatever is written in **Bold** and marked with **Yellow** colour are commands you can copy that

Now by using simple commands will check

**sudo su -** (to become root user)

**df -h** (It will display information about file system disk space usage on the mounted file system)

You can see our EBS volume is not there in the below pic

```
ubuntu@ip-172-31-82-215:~$ sudo su -
root@ip-172-31-82-215:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        7.6G  1.6G   6.0G  21% /
tmpfs            475M    0   475M   0% /dev/shm
tmpfs            190M  848K   190M   1% /run
tmpfs            5.0M    0    5.0M   0% /run/lock
/dev/xvda15     105M   6.1M   99M    6% /boot/efi
tmpfs            95M   4.0K   95M    1% /run/user/1000
root@ip-172-31-82-215:~#
```

We write the below command also

**lsblk** (to see list of blocks it will show where it is mounted)

```
root@ip-172-31-82-215:~# lsblk
NAME        MAJ:MIN     RM  SIZE  RO  TYPE MOUNTPOINTS
loop0        7:0         0 24.9M  1 loop /snap/amazon-ssm-agent/7628
loop1        7:1         0 55.7M  1 loop /snap/core18/2812
loop2        7:2         0 63.9M  1 loop /snap/core20/2182
loop3        7:3         0  87M   1 loop /snap/lxd/27037
loop4        7:4         0 40.4M  1 loop /snap/snapd/20671
xvda         202:0       0   8G   0 disk
├─xvda1      202:1       0  7.9G  0 part /
├─xvda14     202:14      0   4M   0 part
└─xvda15     202:15      0 106M  0 part /boot/efi
xvdbz        202:19712   0 100G   0 disk
root@ip-172-31-82-215:~#
```

So in above pic you can see it is showing xvdbz: 100 Gib

So “**xvdbz**” is our disk name

Disk is not mounted that's why it's not working

So to mount the disk let's check if any file system is present or not

file -s /dev/"drivename" my drive name is xvdbz which is 100 Gib

so command would be-

**file -s /dev/xvdbz**

```
root@ip-172-31-82-215:~# file -s /dev/xvdbz
/dev/xvdbz: data
root@ip-172-31-82-215:~#
```

So there is no file system, now let's make the file system

**mkfs -t ext4 /dev/xvdbz** “xvdbz is my EBS drive name”

```
root@ip-172-31-82-215:~# mkfs -t ext4 /dev/xvdbz
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 26214400 4k blocks and 6553600 inodes
Filesystem UUID: a643298b-a3ca-4ab4-a4ce-6550ca4c0356
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks):
```

Now we have to make a folder where it will mount or we can say it will be our mounting point



**mkdir ebs** (it will make a file directory named as ebs)

**ls** ( to list our )

```
root@ip-172-31-82-215:~# mkdir ebs
root@ip-172-31-82-215:~# ls
ebs  snap
root@ip-172-31-82-215:~#
```

You can see in the above image “ebs” is created

Command to mount

**mount /dev/xvdbz ebs/** (xvdbz is my EBS drive name)

and it's done

```
root@ip-172-31-82-215:~# mount /dev/xvdbz ebs/
root@ip-172-31-82-215:~#
```

now you can check it with

**lsblk** ( it will be mounted to ebs)

our EBS volume xvdbz is mounted

```
root@ip-172-31-82-215:~# lsblk
NAME        MAJ:MIN     RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0         0 24.9M  1 loop /snap/amazon-ssm-agent/7628
loop1        7:1         0 55.7M  1 loop /snap/core18/2812
loop2        7:2         0 63.9M  1 loop /snap/core20/2182
loop3        7:3         0  87M   1 loop /snap/lxd/27037
loop4        7:4         0 40.4M  1 loop /snap/snapd/20671
xvda         202:0       0   8G   0 disk
├─xvda1      202:1       0   7.9G  0 part /
├─xvda14     202:14      0    4M   0 part
└─xvda15     202:15      0 106M   0 part /boot/efi
xvdbz        202:19712   0 100G   0 disk /root/ebs
root@ip-172-31-82-215:~#
```

**Note:** Please do not forget to delete all the Instances and resources after your practice or else it might charge you.

# Thank You