

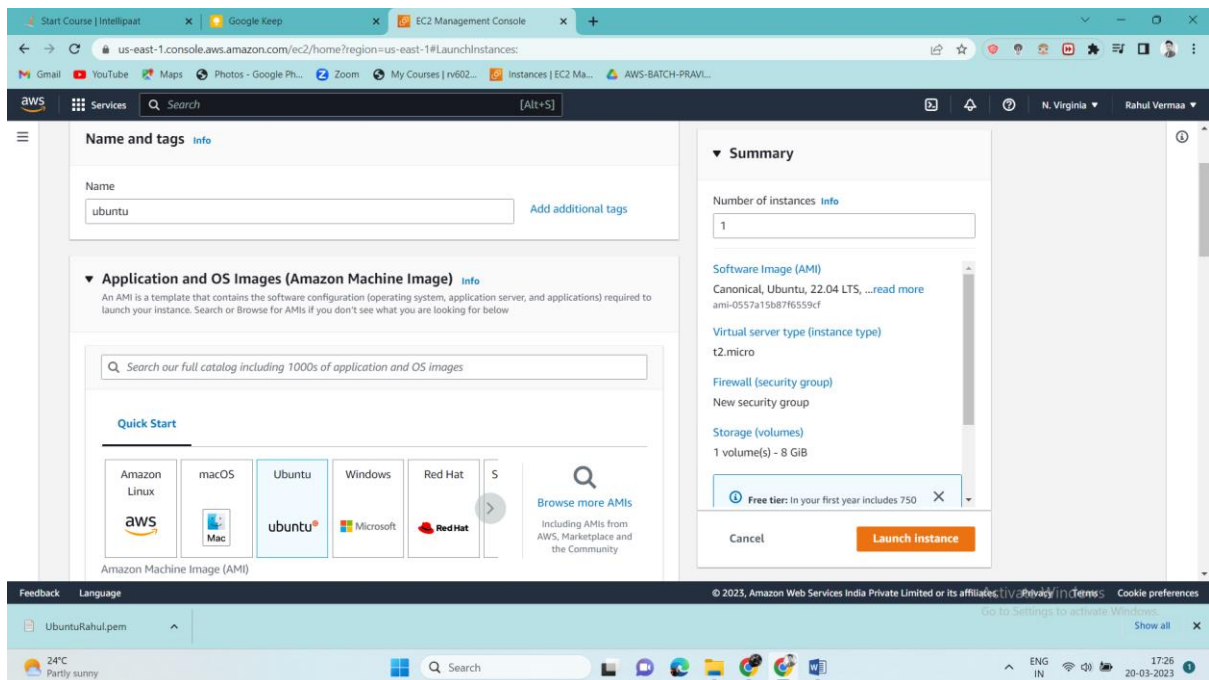
So in this Hands-on will create an EFS and connect it to 3 different EC2 instances and all instances have different operating system such as Ubuntu, red hat linux and amazon linux 2.

Prerequisite:

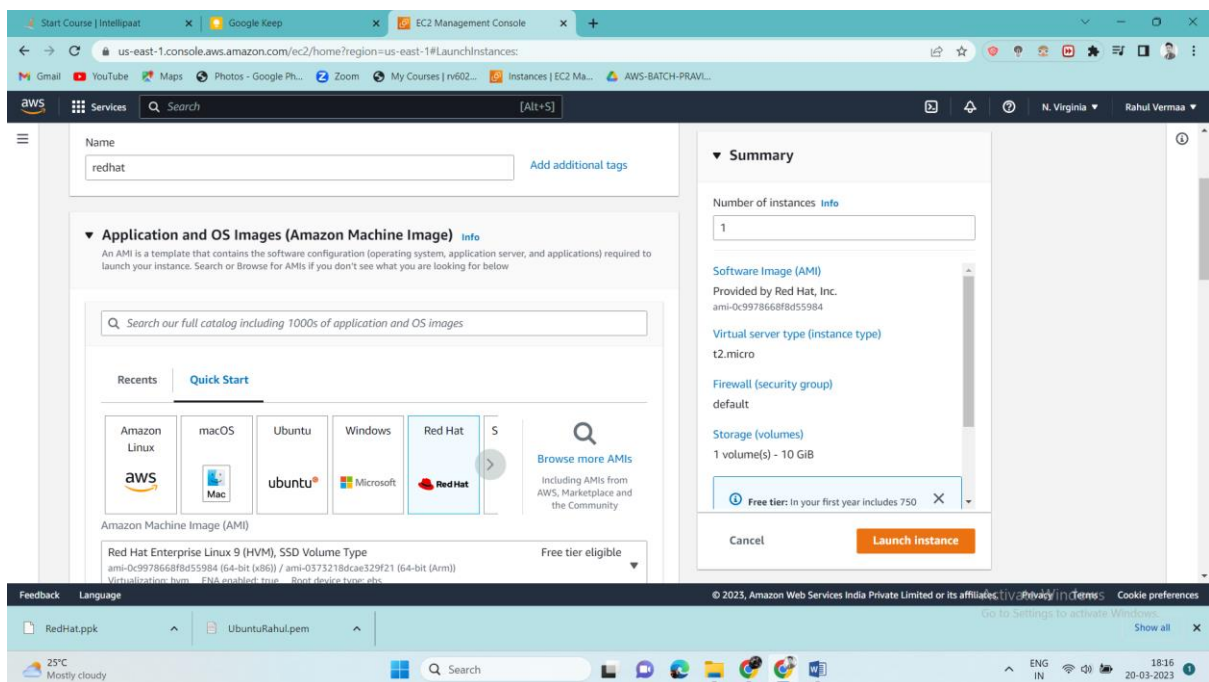
- 1. Create Ubuntu instance**
- 2. Create radhat Instance**
- 3. Create Amazon instance**
- 4. Create EBS volume of any size**

Prerequisite:

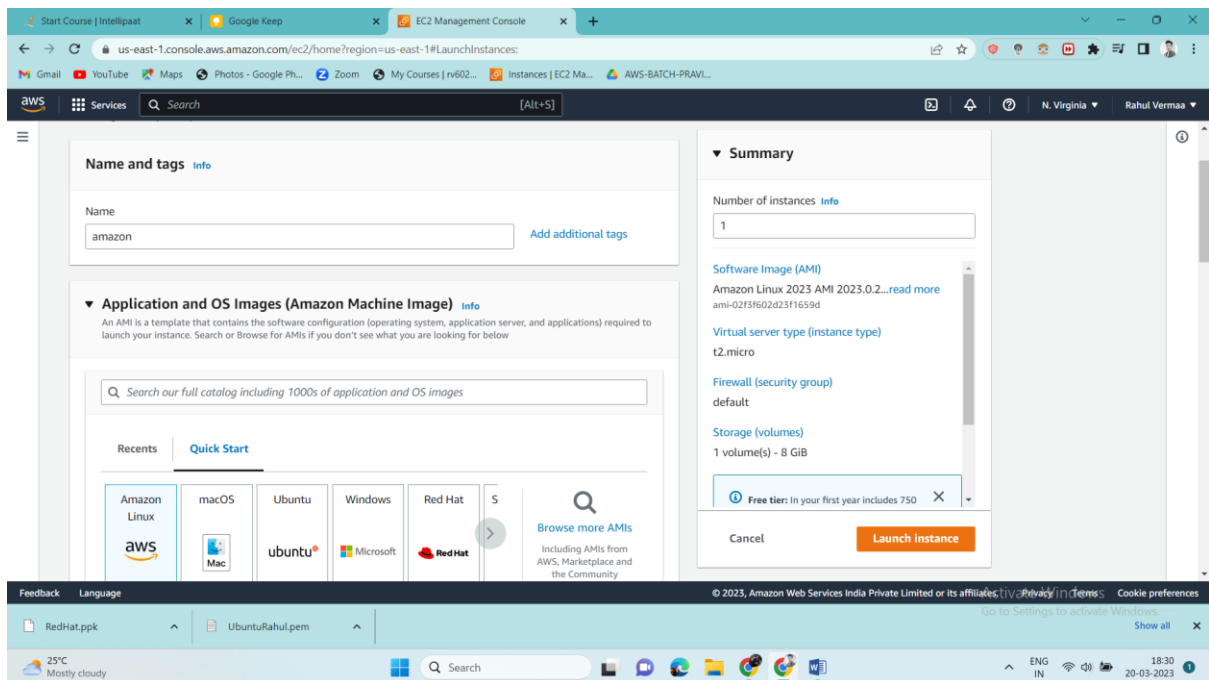
1) First instance OS type Ubuntu



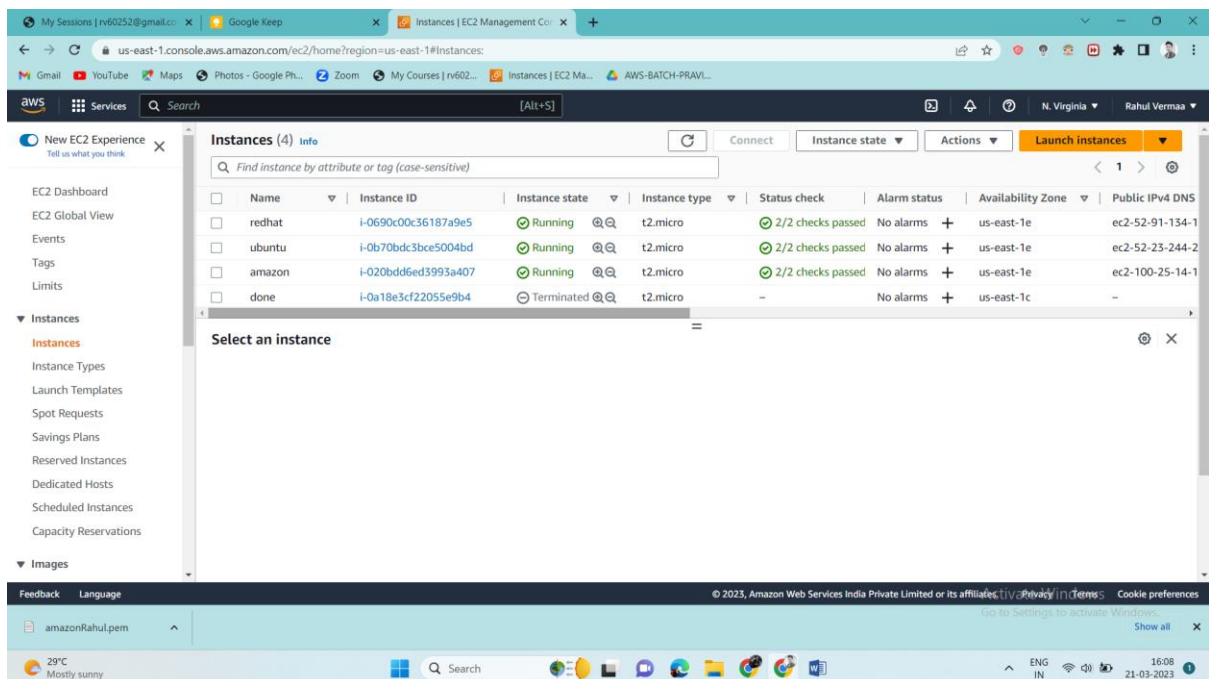
2) Second instance OS type Red hat



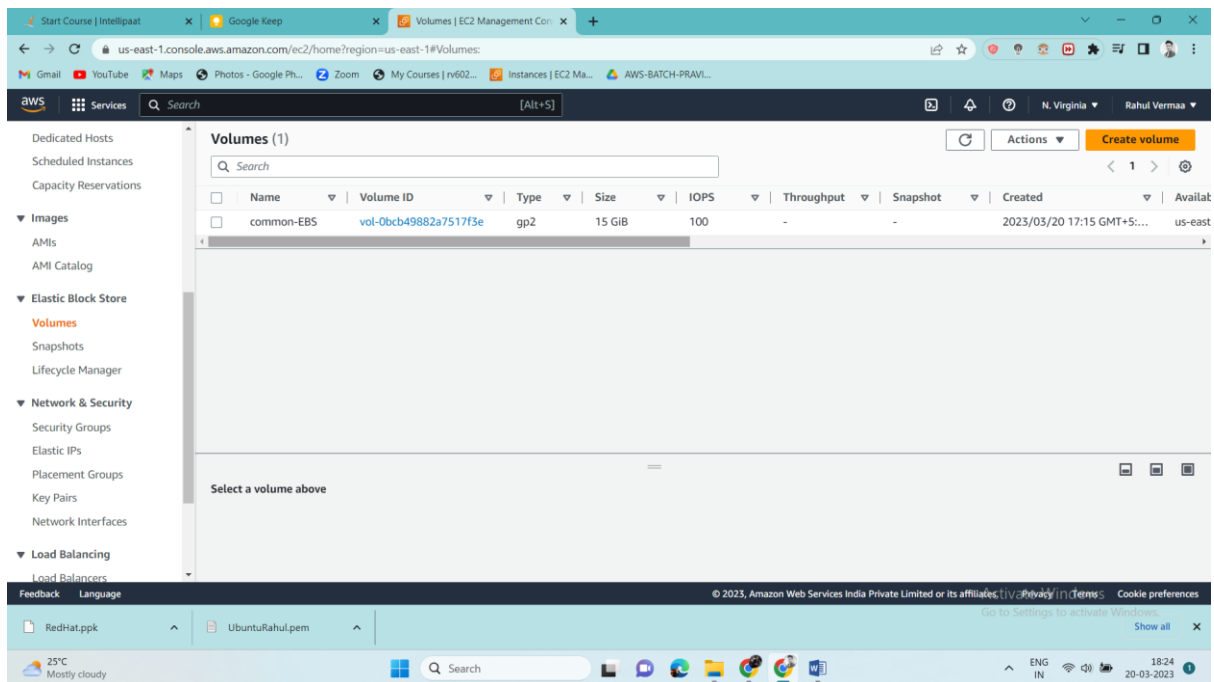
3) Third and last instance with Amazon Linux OS



You can see in the below pic we have created 3 instances with different OS (Ubuntu, red hat and Amazon linux)

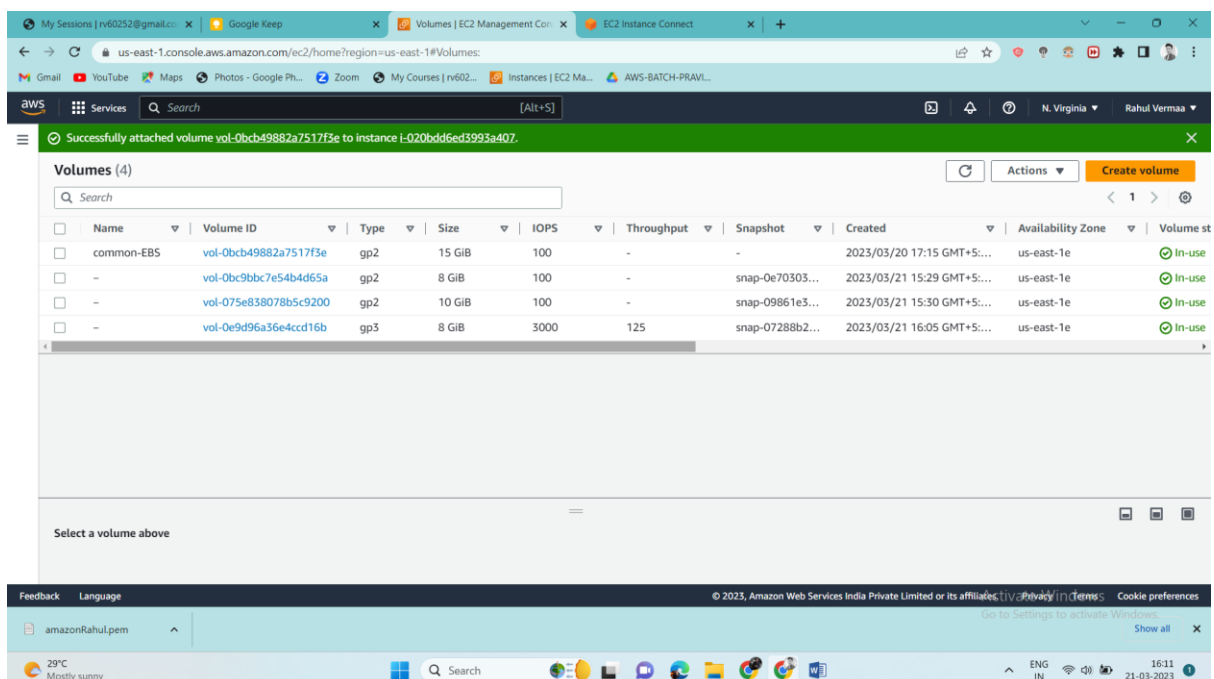


4) We have successfully created one EBS volume which is 15 GB in the below pic



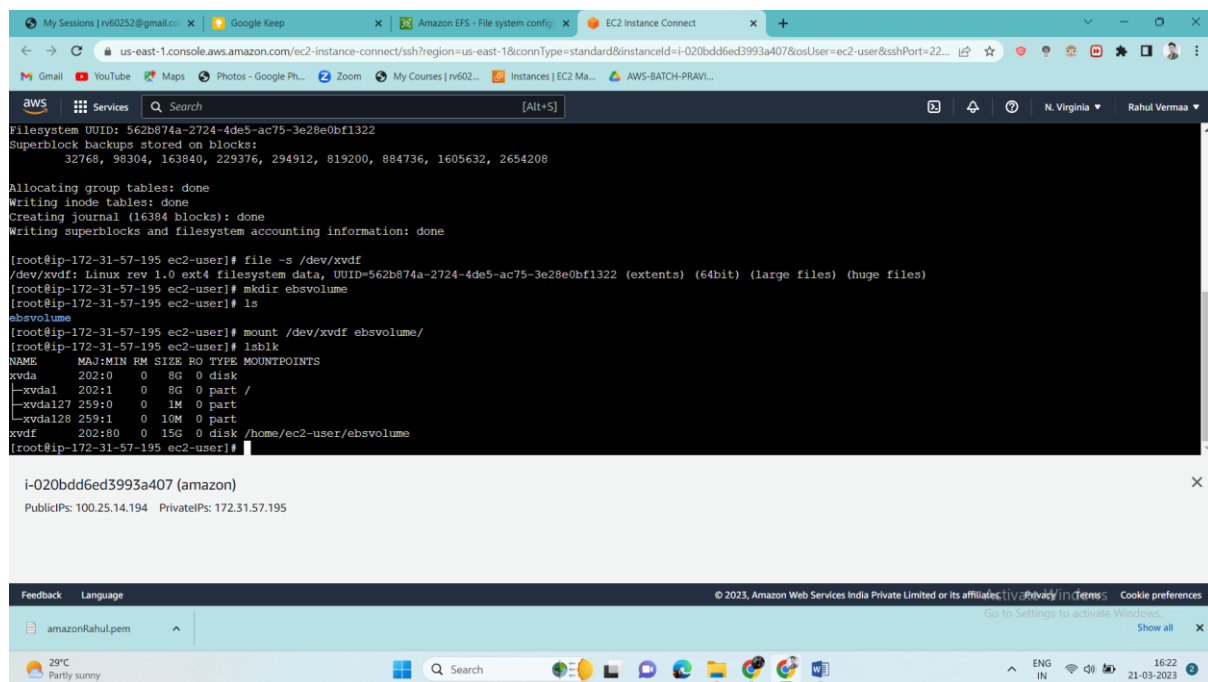
Now attach your EBS volume with any one of your instances here I'm attaching it with Amazon linux.

We have attached our EBS with amazon linux



Command to mount EBS volume-

sudo su (To become root or super user)
lsblk (To list list of block)
{file -s /dev/"Ebs drive name" (ESB volume name}
file -s /dev/xvdbz (In my case my EBS drive name is xvdbz)
{mkfs -t ext4 /dev/"Ebs drive name"
mkfs -t ext4 /dev/xvdbz (To make file system)
mkdir ebs (To create a directory name as Ebs)
ls (To list out)
{mount /dev/"ebs drive name" ebs/}
mount /dev/xvdbz ebs/ (xvdbz mount to ebs directory)
lsblk (now write this to verify where it is mounted)



The screenshot shows a terminal window connected to an AWS EC2 instance. The terminal output displays the following commands and their results:

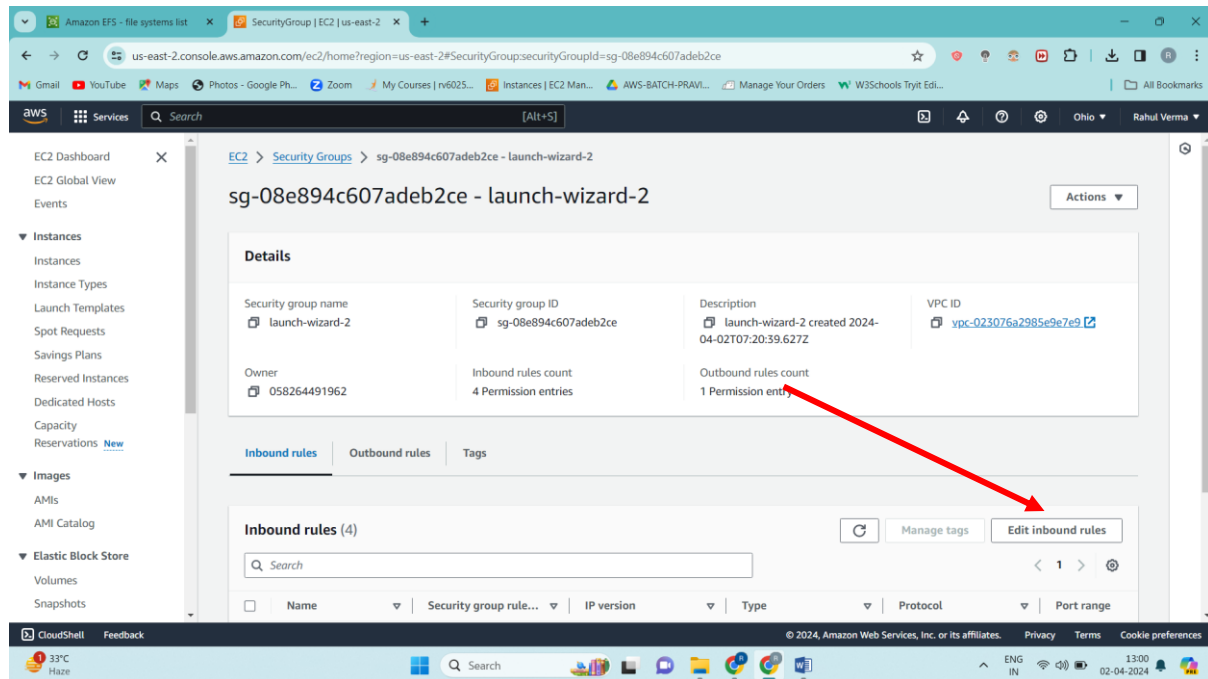
```
Filesystem UUID: 562b874a-2724-4de5-ac75-3e28e0bf1322
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

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

[root@ip-172-31-57-195 ec2-user]# file -s /dev/xvdf
/dev/xvdf: Linux rev 1.0 ext4 filesystem data, UUID=562b874a-2724-4de5-ac75-3e28e0bf1322 (extents) (64bit) (large files) (huge files)
[root@ip-172-31-57-195 ec2-user]# mkdir ebsvolume
[root@ip-172-31-57-195 ec2-user]# ls
ebsvolume
[root@ip-172-31-57-195 ec2-user]# mount /dev/xvdf ebsvolume/
[root@ip-172-31-57-195 ec2-user]# 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 
xvdf         202:80    0   15G  0 disk /home/ec2-user/ebsvolume
[root@ip-172-31-57-195 ec2-user]#
```

Below the terminal window, a metadata box for the instance `i-020bdd6ed3993a407` (amazon) is visible, showing Public IPs: 100.25.14.194 and Private IPs: 172.31.57.195.

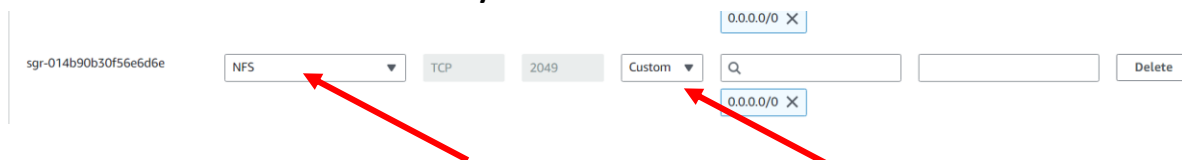
Now go to security group of all the instances one by one and Allow NFS. let's do it for Amazon linux first. So after going to security group click on edit inbound rules.



Now click on add rule button



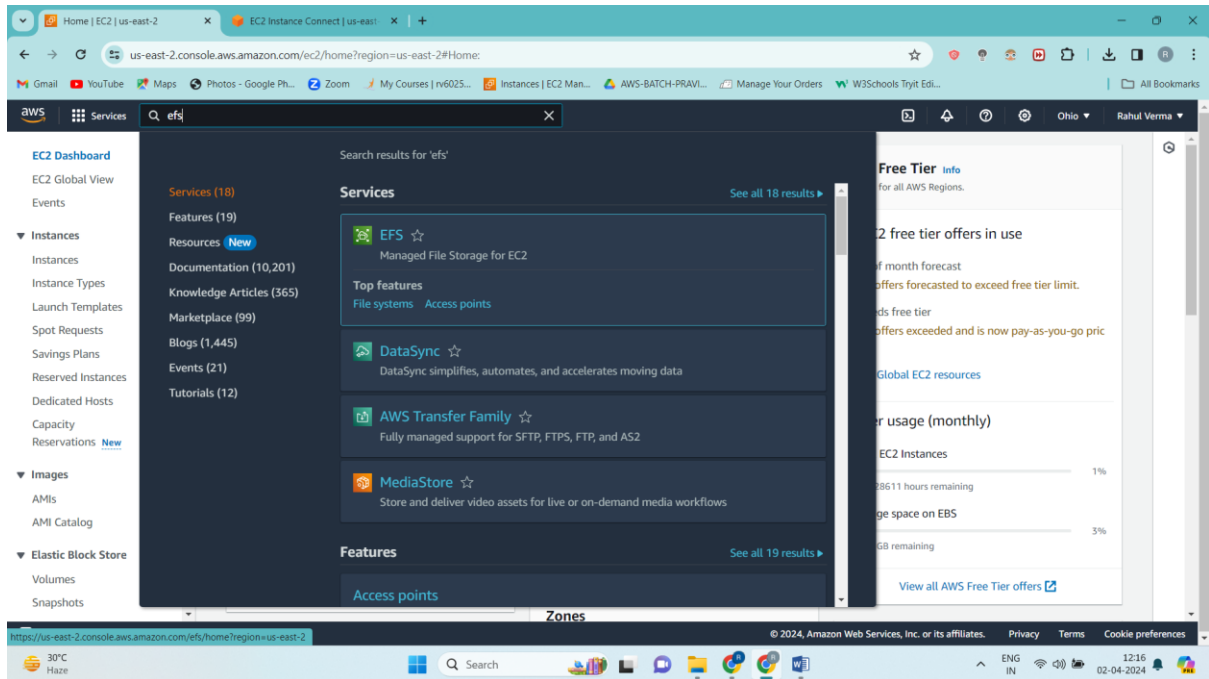
Allow NFS traffic it to anywhere and save it



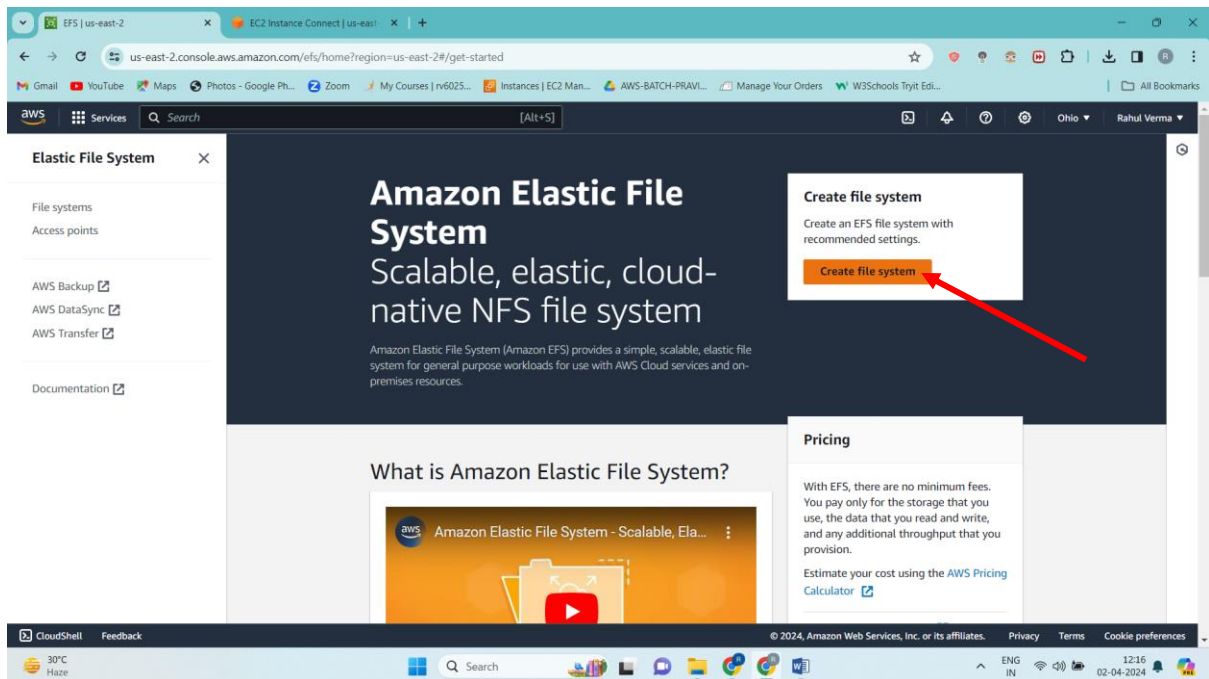
EFS Steps



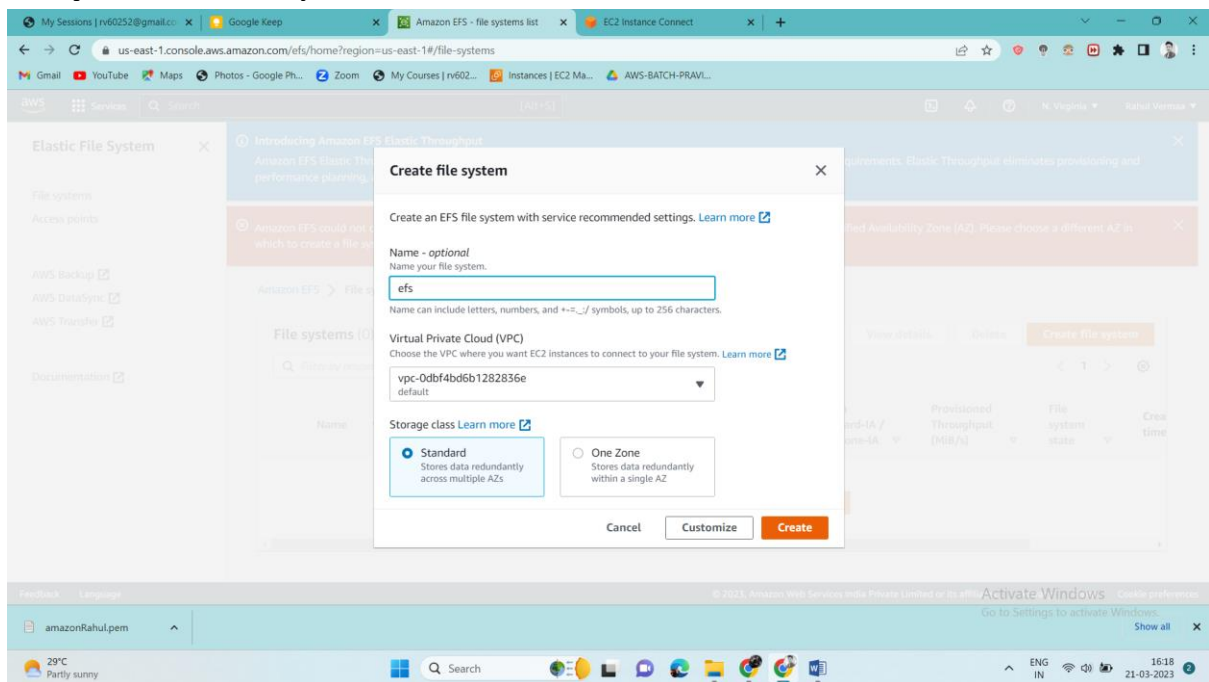
Step 1: Create an EFS from amazon console



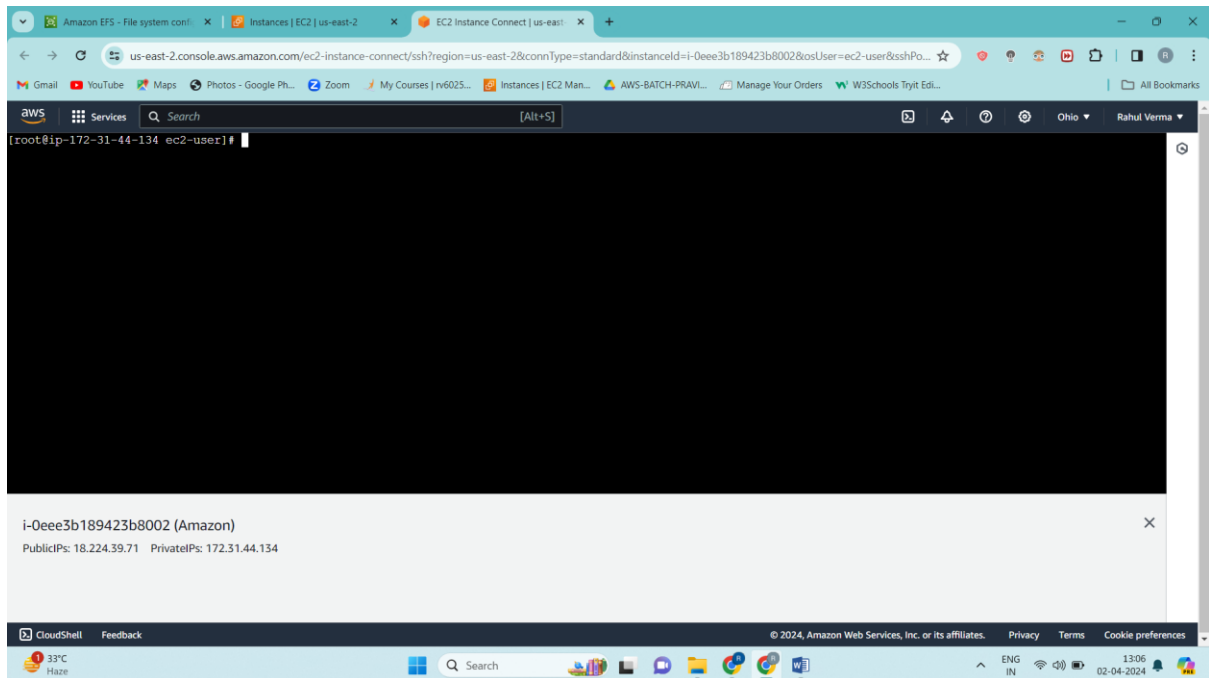
Now click on create file system



Step 2: Define your efs name



Once your EFS is created now go back to your Amazon linux



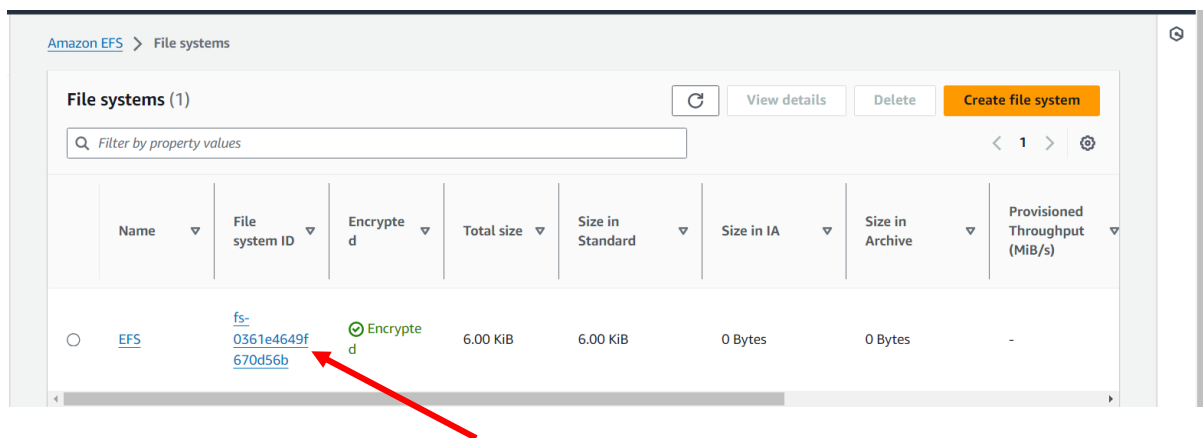
Write following Commands-

`sudo su`

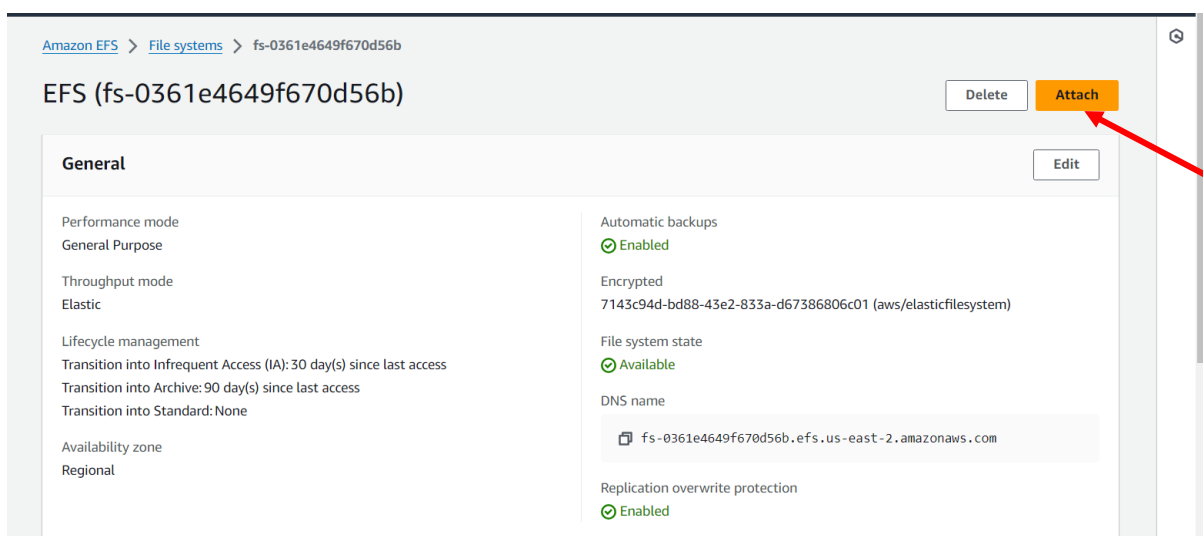
`yum install amazon-efs-utils -y` (to Install efs dependencies)

`mkdir efs` (to create efs directory)

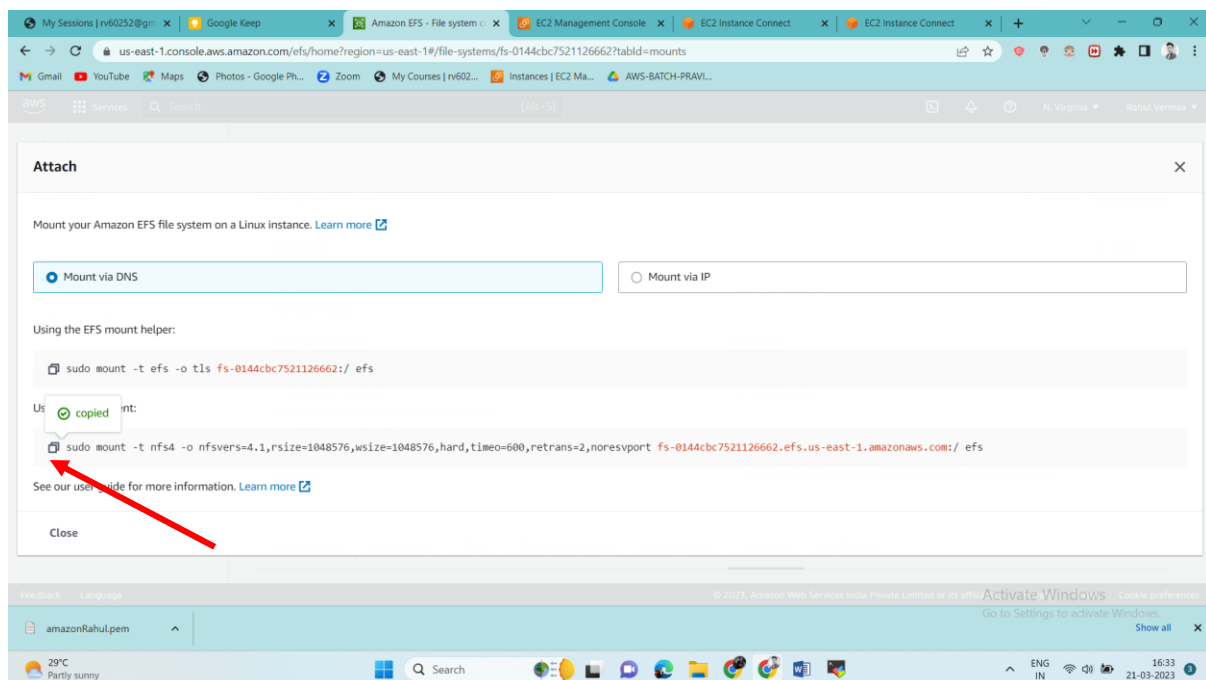
Step 3: Now go to EFS click on your EFS which you have created



Afterwards click on Attach



Now copy NFS client command-



Step 4: Paste that nfs client command in your amazon instance after this mounting will be done.

Now connect to your Ubuntu instance

Write following commands to setup-

`sudo su -`

`apt-get update`

`apt-get install nfs-common -y`

`mkdir efs`

copy paste efs client commad

Similarly will connect to our rad hat linux instance

```
sudo su
```

```
yum update -y
```

```
yum install nfs-utils -y
```

```
mkdir efs
```

copy paste efs client command

Step 5: Now let test it we are connected to our ubuntu instance.

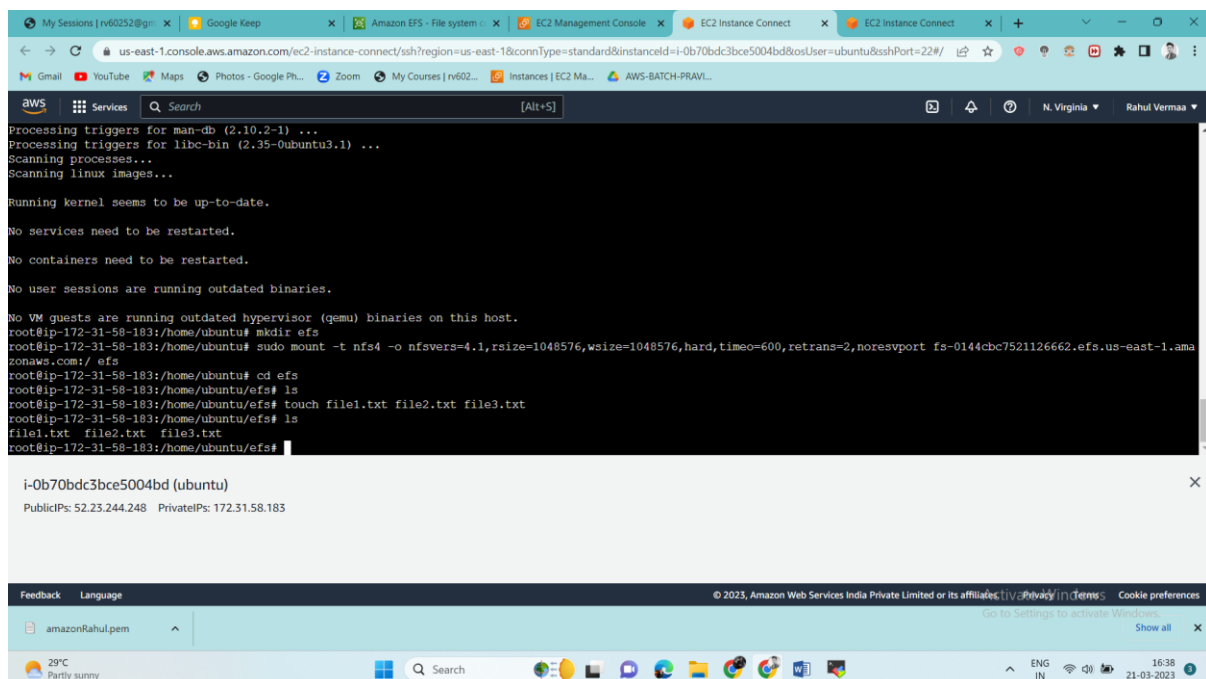
```
cd efs
```

 (to go into efs directory)

```
touch file1.txt file2.txt file3.txt
```

 (this will create 3 files)

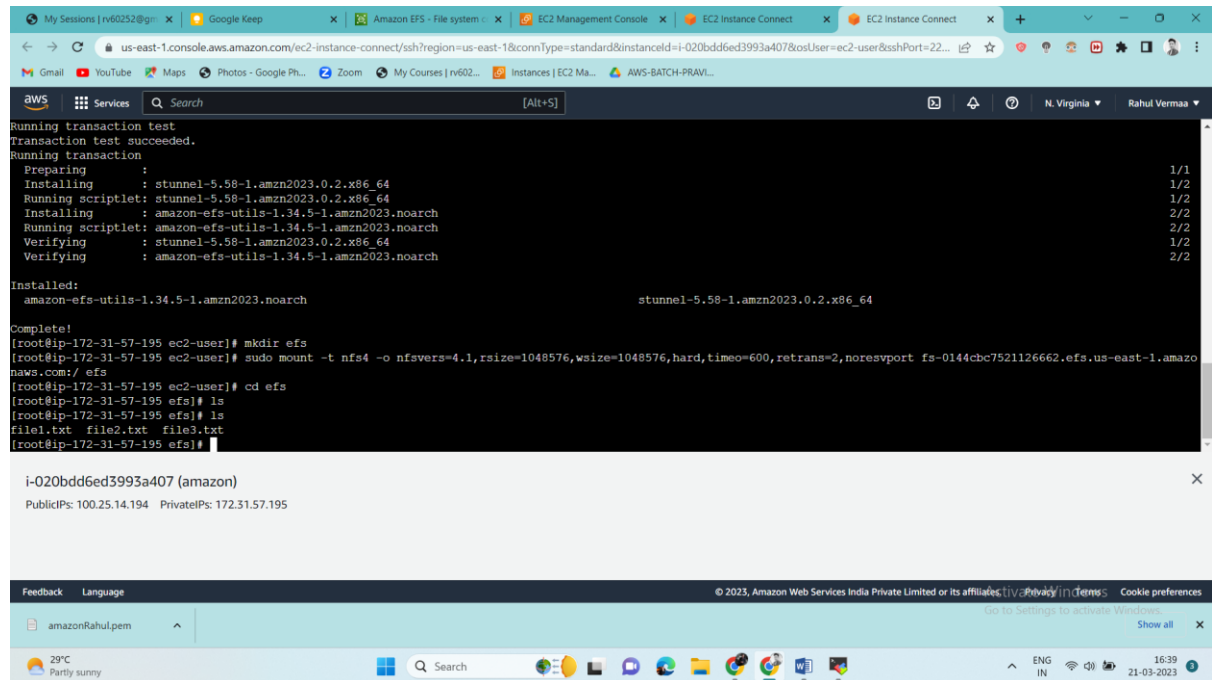
```
ls
```

 (to list out these files)

Now above created files should reflect in other 2 instances also. Amazon instance-

cd efs

ls



```
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : stunnel-5.58-1.amzn2023.0.2.x86_64 1/1
Running scriptlet: stunnel-5.58-1.amzn2023.0.2.x86_64 1/2
Installing : amazon-efs-utils-1.34.5-1.amzn2023.noarch 2/2
Running scriptlet: amazon-efs-utils-1.34.5-1.amzn2023.noarch 2/2
Verifying : stunnel-5.58-1.amzn2023.0.2.x86_64 1/2
Verifying : amazon-efs-utils-1.34.5-1.amzn2023.noarch 2/2

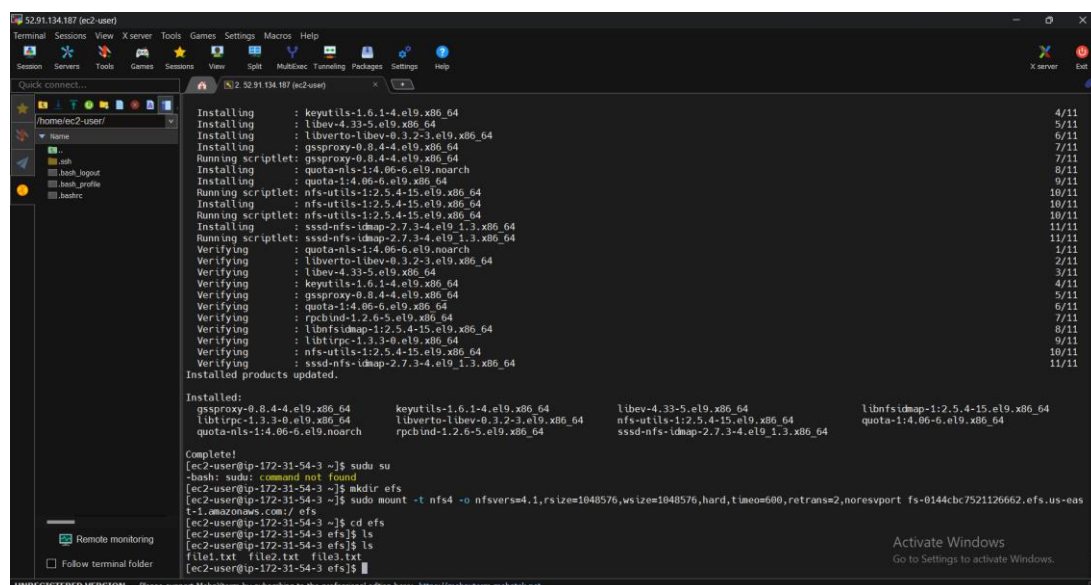
Installed:
  amazon-efs-utils-1.34.5-1.amzn2023.noarch                                stunnel-5.58-1.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-57-195 ec2-user]# mkdir efs
[root@ip-172-31-57-195 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport fs-0144cbc7521126662.efs.us-east-1.amazonaws.com:/ efs
[root@ip-172-31-57-195 ec2-user]# cd efs
[root@ip-172-31-57-195 efs]# ls
[root@ip-172-31-57-195 efs]# ls
file1.txt file2.txt file3.txt
[root@ip-172-31-57-195 efs]#
```

Redhat instance

cd efs

ls



```
Installing : keyutils-1.6.1-4.el9.x86_64 4/11
Installing : libev-4.33-5.el9.x86_64 5/11
Installing : libvirt-10.2.2-3.el9.x86_64 6/11
Installing : gssproxy-0.8.4-4.el9.x86_64 7/11
Running scriptlet: gssproxy-0.8.4-4.el9.x86_64 7/11
Installing : quota-nls-1:4.06-6.el9.x86_64 8/11
Running scriptlet: nfs-utils-1:2.5.4-15.el9.x86_64 9/11
Installing : nfs-utils-1:2.5.4-15.el9.x86_64 10/11
Running scriptlet: nfs-utils-1:2.5.4-15.el9.x86_64 10/11
Installing : sssd-nfs-idmap-2.7.3-4.el9_1.3.x86_64 11/11
Running scriptlet: sssd-nfs-idmap-2.7.3-4.el9_1.3.x86_64 11/11
Verifying : quota-nls-1:4.06-6.el9.x86_64 1/11
Verifying : libvirt-10.2.2-3.el9.x86_64 2/11
Verifying : libev-4.33-5.el9.x86_64 3/11
Verifying : keyutils-1.6.1-4.el9.x86_64 4/11
Verifying : gssproxy-0.8.4-4.el9.x86_64 5/11
Verifying : quota-1:4.06-6.el9.x86_64 6/11
Verifying : rpcbind-1.2.6-5.el9.x86_64 7/11
Verifying : libfsidmap-1:2.5.4-15.el9.x86_64 8/11
Verifying : libtirpc-1.3.3-0.el9.x86_64 9/11
Verifying : nfs-utils-1:2.5.4-15.el9.x86_64 10/11
Verifying : sssd-nfs-idmap-2.7.3-4.el9_1.3.x86_64 11/11
Installed products updated.

Installed:
  gssproxy-0.8.4-4.el9.x86_64      keyutils-1.6.1-4.el9.x86_64      libev-4.33-5.el9.x86_64      libfsidmap-1:2.5.4-15.el9.x86_64
  libtirpc-1.3.3-0.el9.x86_64      libvirt-10.2.2-3.el9.x86_64      nfs-utils-1:2.5.4-15.el9.x86_64      quota-1:4.06-6.el9.x86_64
  quota-nls-1:4.06-6.el9.x86_64      rpcbind-1.2.6-5.el9.x86_64      sssd-nfs-idmap-2.7.3-4.el9_1.3.x86_64

Complete!
[ec2-user@ip-172-31-54-3 ~]$ sudo su
-bash: sudo: command not found
[ec2-user@ip-172-31-54-3 ~]$ mkdir efs
[ec2-user@ip-172-31-54-3 ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport fs-0144cbc7521126662.efs.us-east-1.amazonaws.com:/ efs
[ec2-user@ip-172-31-54-3 ~]$ cd efs
[ec2-user@ip-172-31-54-3 efs]$ ls
[ec2-user@ip-172-31-54-3 efs]$ ls
file1.txt file2.txt file3.txt
[ec2-user@ip-172-31-54-3 efs]$
```

Note: To avoid charges from AWS, terminate all the instances
After your hands-on is done.

Thank You