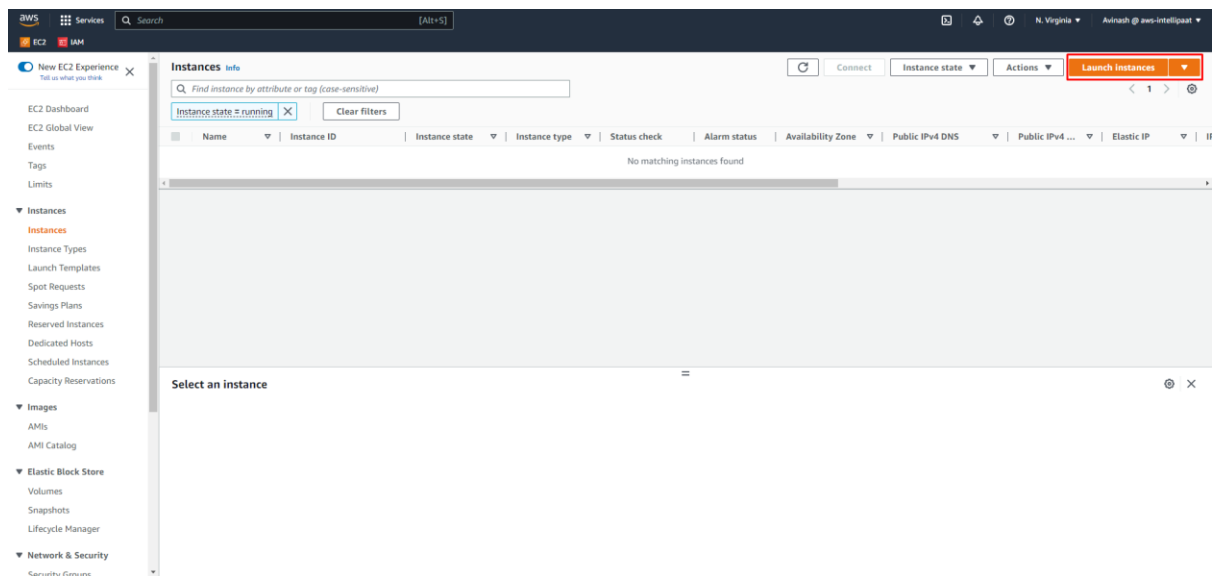


Module-2: EC2 and EFS Assignment - 3

1. Create an EFS and connect it to 3 different EC2 instances. Make sure the all instances have different Operating System. For instance, Ubuntu, Red Hat Linux and Amazon Linux 2

Ans:

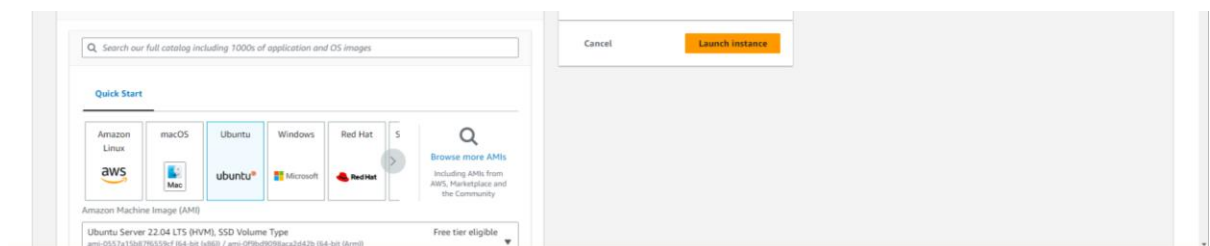
1. Open up the AWS Management Console
2. Check for the region [us-east-1(N. Virginia)]
3. Search for EC2 in the search box
4. Click on instances to go to the EC2 console
5. Click on Launch Instance



6. Setup the 1st instance using following configurations:

- a. Name : AvinashAWS
- b. AMI : Quickstart >> UbuntuOS [Any version which is free tier eligible]
- c. Instance type : t2.micro [free tier eligible]
- d. Key-pair : Create a key pair [rsa and .pem]
- e. Security group : Default

7. Click on Launch Instance.



8. Setup the 2nd instance using following configurations:

- a. Name : AvinashAWS
 - b. AMI : Quickstart >> Red Hat Linux [Any version which is free tier eligible]
 - c. Instance type : t2.micro [free tier eligible]
 - d. Key-pair : Create a key pair [rsa and .pem]
 - e. Security group : Default
9. Click on Launch Instance.

Launch an instance [info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [info](#)

Name
Avinash-RHEL-efs-assignment [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat S [Browse more AMIs](#)

Amazon Machine Image (AMI)

Red Hat Enterprise Linux 9 (HVM), SSD Volume Type
ami-0c979668f8d553954 (64-bit x86) / ami-037527180a5329f21 (64-bit ARM)

Free tier eligible

Number of instances [info](#)

1

Software image (AMI)
Provided by Red Hat, Inc.
ami-0c979668f8d553954

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 10 GiB

Cancel [Launch instance](#)

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

10. Setup the 3rd instance using following configurations:

- a. Name : AvinashAWS
 - b. AMI : Quickstart >> Amazon Linux Machine [Any version which is free tier eligible]
 - c. Instance type : t2.micro [free tier eligible]
 - d. Key-pair : Create a key pair [rsa and .pem]
 - e. Security group : Default
11. Click on Launch Instance.

Launch an instance [info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [info](#)

Name
Avinash-AmazonLinux-efs-assignment [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat S [Browse more AMIs](#)

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-006dcf34c09c50022 (64-bit x86) / ami-0f254afbc55dad58 (64-bit ARM)

Free tier eligible

Number of instances [info](#)

1

Software image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-006dcf34c09c50022

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

Cancel [Launch instance](#)

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Now we will be creating an EFS Volume(Elastic File System) in AWS.

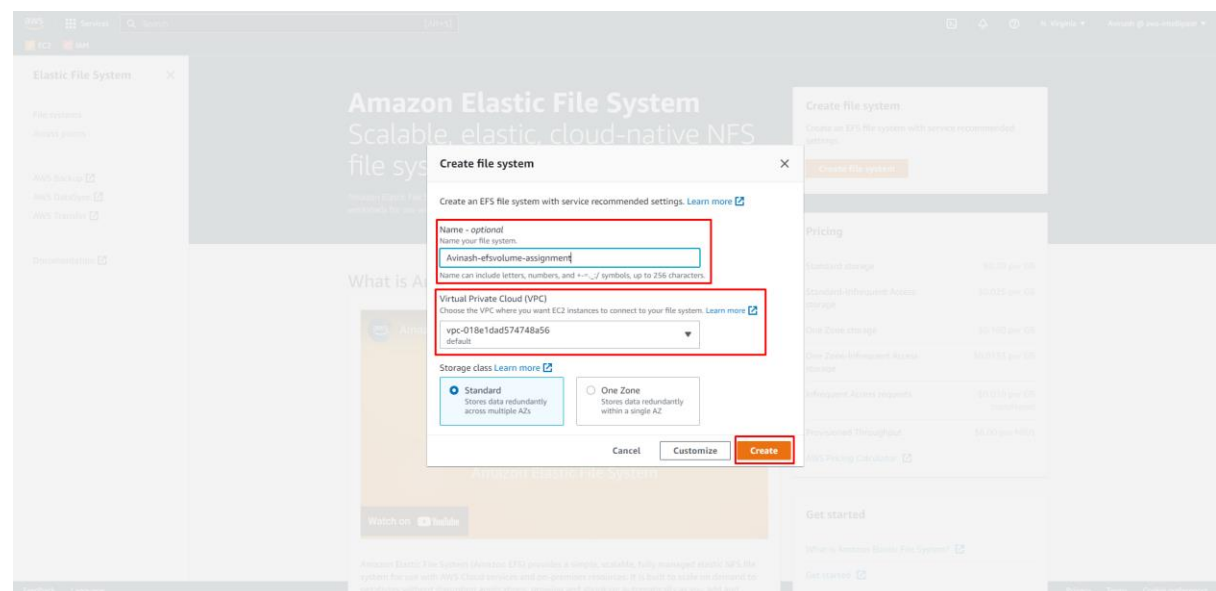
12. Search for EFS on the Search Bar

13. Click on Create a File System

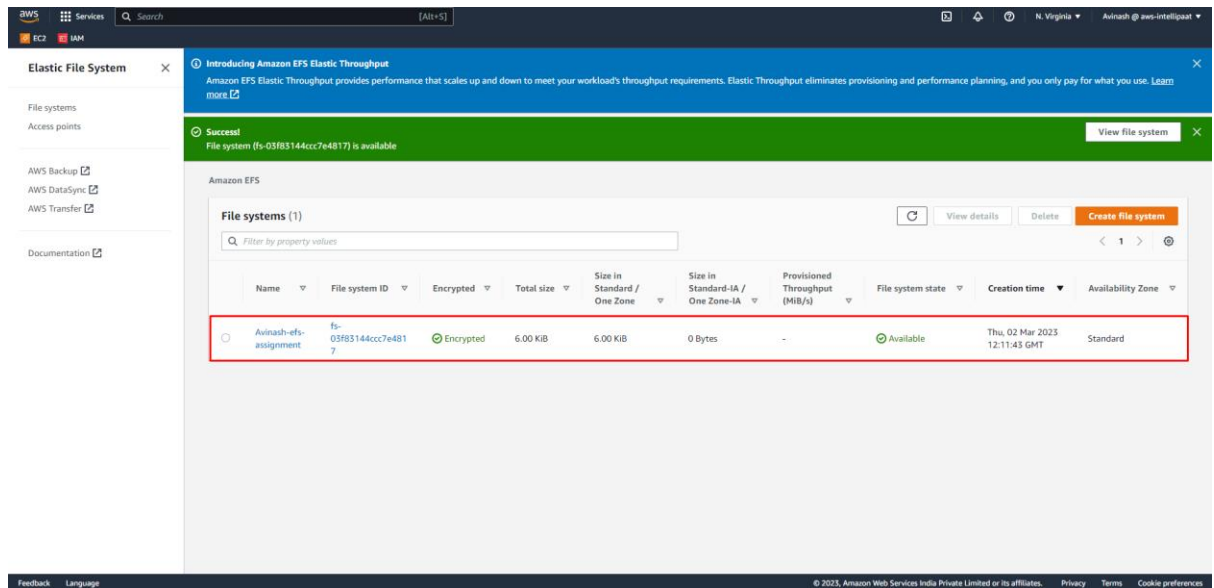


14. Create a EFS Volume with the following characteristics:

- a. Name: Any name
- b. VPC : should be same that you have used while creating the instance.
- c. Storage Class: Standard



15. We will be seeing the EFS Volume created and is Available



Now we will attach this EFS to three EC2 instance that we have made previously.

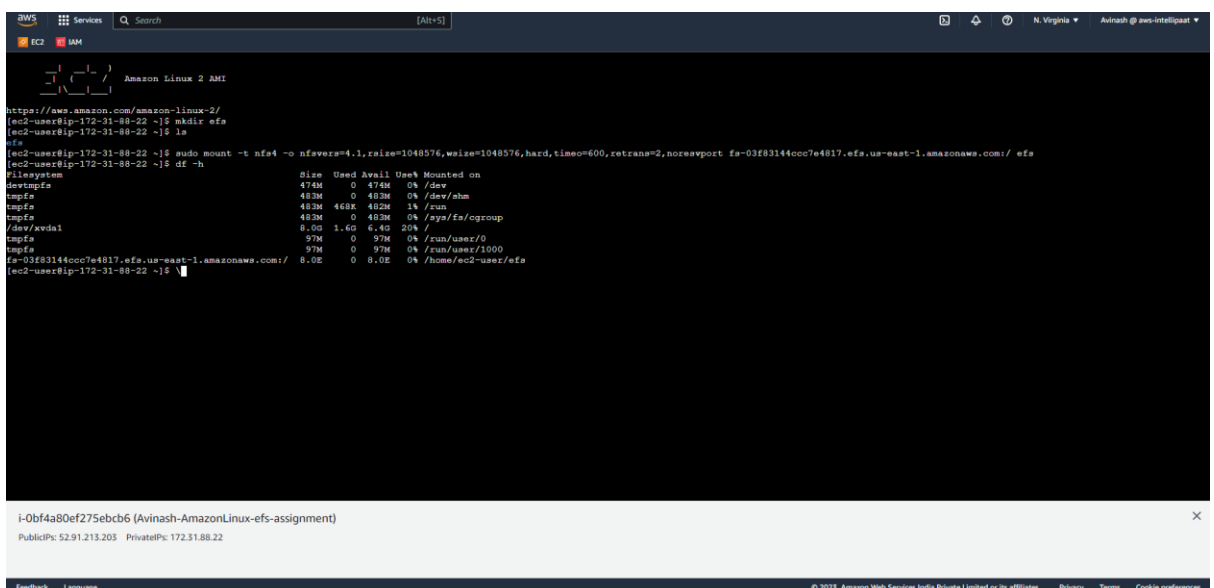
Amazon-Linux -> Already has amazon-efs-utils so directly make a directory and mount it

1. Make a directory named efs using the following command:

```
$ sudo mkdir efs
$ ls
```

2. Click on Attach button on EFS dashboard
3. Copy the command from Mount with DNS
4. Run the command in the machine.
5. Check the storage on the machine.

```
$ df -h
```



RHEL -> Import repositories from Github then install packages from that repo and then create a directory and mount it.

1. Install the package **nfs-common**

```
$ sudo yum install nfs-utils -y
```

2. Make a directory named **efs**

```
$ sudo mkdir efs
```

3. Click on Attach button on EFS dashboard
4. Copy the command from Mount with DNS
5. Run the command in the machine.
6. Check the storage on the machine.

```
$ df -h
```

```
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : libtirpc-1.3.3-0.el9.x86_64 1/1
Installing : libnfs-idmap-1:2.5.4-15.el9.x86_64 1/11
Installing : rpcbind-1:2.6-5.el9.x86_64 2/11
Running scriptlet: rpcbind-1:2.6-5.el9.x86_64 3/11
Installing : rpcbind-1:2.6-5.el9.x86_64 3/11
Running scriptlet: rpcbind-1:2.6-5.el9.x86_64 3/11
Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service -> /usr/lib/systemd/system/rpcbind.service.
Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket -> /usr/lib/systemd/system/rpcbind.socket.
Installing : keyutils-1.6.1-4.el9.x86_64 4/11
Installing : libev-4.33-5.el9.x86_64 5/11
Installing : libev-4.33-5.el9.x86_64 5/11
Installing : gssproxy-0.8.4-4.el9.x86_64 6/11
Running scriptlet: gssproxy-0.8.4-4.el9.x86_64 7/11
Installing : quota-nls-1:4.06-6.el9.noarch 8/11
Installing : quota-1:4.06-6.el9.x86_64 9/11
Running scriptlet: nfs-utils-1:2.5.4-15.el9.x86_64 10/11
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-2.6.2-4.el9_0.1.x86_64 11/11
Running scriptlet: sssd-nfs-idmap-2.6.2-4.el9_0.1.x86_64 11/11
Verifying : quota-nls-1:4.06-6.el9.noarch 1/11
Verifying : libverto-libev-0.3.2-3.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 : sssd-nfs-idmap-2.6.2-4.el9_0.1.x86_64 7/11
Verifying : rpcbind-1:2.6-5.el9.x86_64 8/11
Verifying : libnfs-idmap-1:2.5.4-15.el9.x86_64 9/11
Verifying : libtirpc-1.3.3-0.el9.x86_64 10/11
Verifying : nfs-utils-1:2.5.4-15.el9.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      libnfs-idmap-1:2.5.4-15.el9.x86_64      libtirpc-1.3.3-0.el9.x86_64      libverto-libev-0.3.2-3.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.noarch  rpcbind-1:2.6-5.el9.x86_64      sssd-nfs-idmap-2.6.2-4.el9_0.1.x86_64

Complete!
[ec2-user@ip-172-31-27-238 ~]$ sudo mkdir efs
[ec2-user@ip-172-31-27-238 ~]$ ls
efs
[ec2-user@ip-172-31-27-238 ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-03f83144ccc7e4817.efs.us-east-1.amazonaws.com:/ efs
[ec2-user@ip-172-31-27-238 ~]$ df -h
Filesystem                                Size  Used Avail Use% Mounted on
devtmpfs                                 440M   0  440M   0% /dev
tmpfs                                     478M   0  478M   0% /dev/shm
tmpfs                                     192M  5.3M  186M   3% /run
/dev/xvda4                               9.4G  1.2G  8.2G  12% /
/dev/xvda3                               495M  108M  388M  22% /boot
/dev/xvda2                               200M  8.0K  200M   1% /boot/efi
tmpfs                                     95M   0   95M   0% /run/user/1000
fs-03f83144ccc7e4817.efs.us-east-1.amazonaws.com:/ 8.0E   0  8.0E   0% /home/ec2-user/efs
```

Ubuntu -> install nfs-commons and then make a directory and mount it.

7. Install the package **nfs-common**

```
$ sudo apt-get install nfs-common -y
```

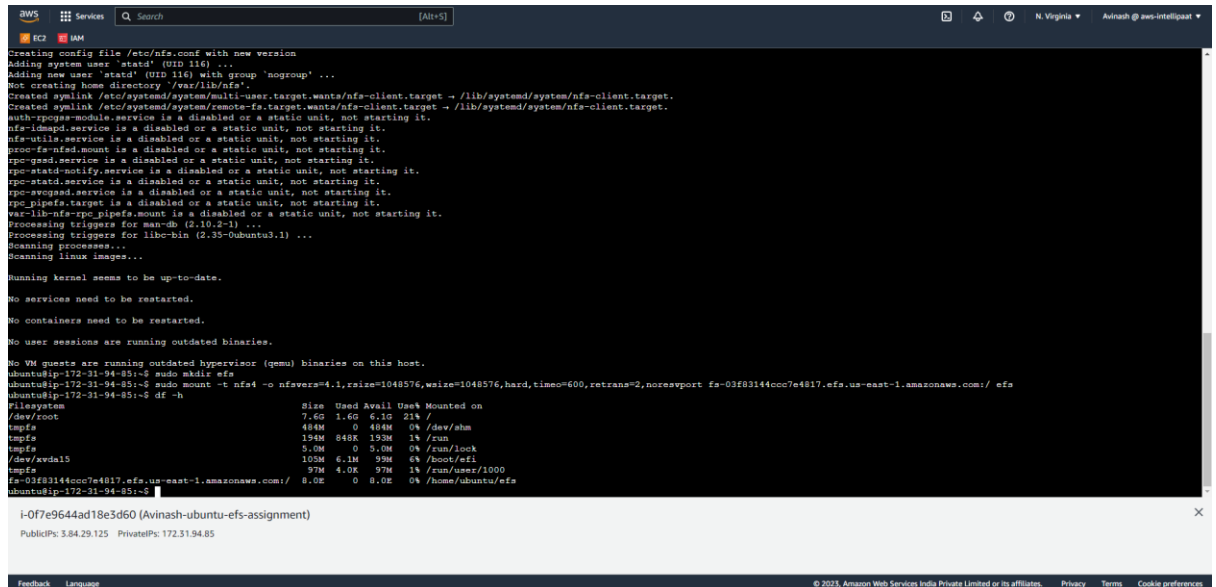
8. Make a directory named **efs**

```
$ sudo mkdir efs
```

9. Click on Attach button on EFS dashboard
10. Copy the command from Mount with DNS
11. Run the command in the machine.

12. Check the storage on the machine.

```
$ df -h
```



```
Creating config file /etc/nfs.conf with new version
Adding system user 'nfsd' (UID 116) ...
Adding new user 'nfsd' (UID 116) with group 'nogroup' ...
Not creating home directory '/var/lib/nfs/'.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-client.target -> /lib/systemd/system/nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target -> /lib/systemd/system/nfs-client.target.
auth-rpcgss-module.service is a disabled or a static unit, not starting it.
nfs-fsmapd.service is a disabled or a static unit, not starting it.
nfs-tilde.service is a disabled or a static unit, not starting it.
proc-fs-nfsd.mount is a disabled or a static unit, not starting it.
rpc-rpcd.service is a disabled or a static unit, not starting it.
rpc-statd.notify.service is a disabled or a static unit, not starting it.
rpc-statd.service is a disabled or a static unit, not starting it.
rpc-svcgssd.service is a disabled or a static unit, not starting it.
rpc-pipefs.target is a disabled or a static unit, not starting it.
var-lib-nfs-rpc-pipefs.mount is a disabled or a static unit, not starting it.
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-94-85:~$ sudo mkdir efs
ubuntu@ip-172-31-94-85:~$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-03f83144ccc7e4817.efs.us-east-1.amazonaws.com:/ efs
ubuntu@ip-172-31-94-85:~$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/root	7.6G	1.6G	6.1G	21%	/
tmpfs	484M	0	484M	0%	/dev/shm
tmpfs	194M	848K	193M	1%	/run
tmpfs	3.0M	0	3.0M	0%	/run/lock
/dev/xvda15	105M	6.1M	99M	6%	/boot/efi
tmpfs	97M	4.0K	97M	1%	/run/user/1000
fs-03f83144ccc7e4817.efs.us-east-1.amazonaws.com:/	8.0E	0	8.0E	0%	/home/ubuntu/efs

i-07e9644ad18e3d60 (Avinash-ubuntu-efs-assignment)
PublicIPs: 3.84.29.125 PrivateIPs: 172.31.94.85

To detach the EFS Volume use the following command:

```
$ sudo umount efs
```