S3 Bucket Administration from EC2 Instance using AWS CLI

In this Lab, we are going to access and administer S3 Buckets from EC2 Instance using AWS CLI. We are also going to learn how to mount S3 Buckets to an EC2 Instance using S3fs file system. This is a best practice in industries if you want to archive data from local server or an EC2 Instance to S3 Buckets.

Below is the list of Tasks:

Task 1: Create IAM Role

Task 2: Launch & Configure EC2 Instance

Task 3: Connect to the EC2 Instance

Task 4: S3 Bucket Administration using AWS CLI

Task 5: Mount S3 Buckets to EC2 Instance using AWS CLI

Task 1: Create IAM Role

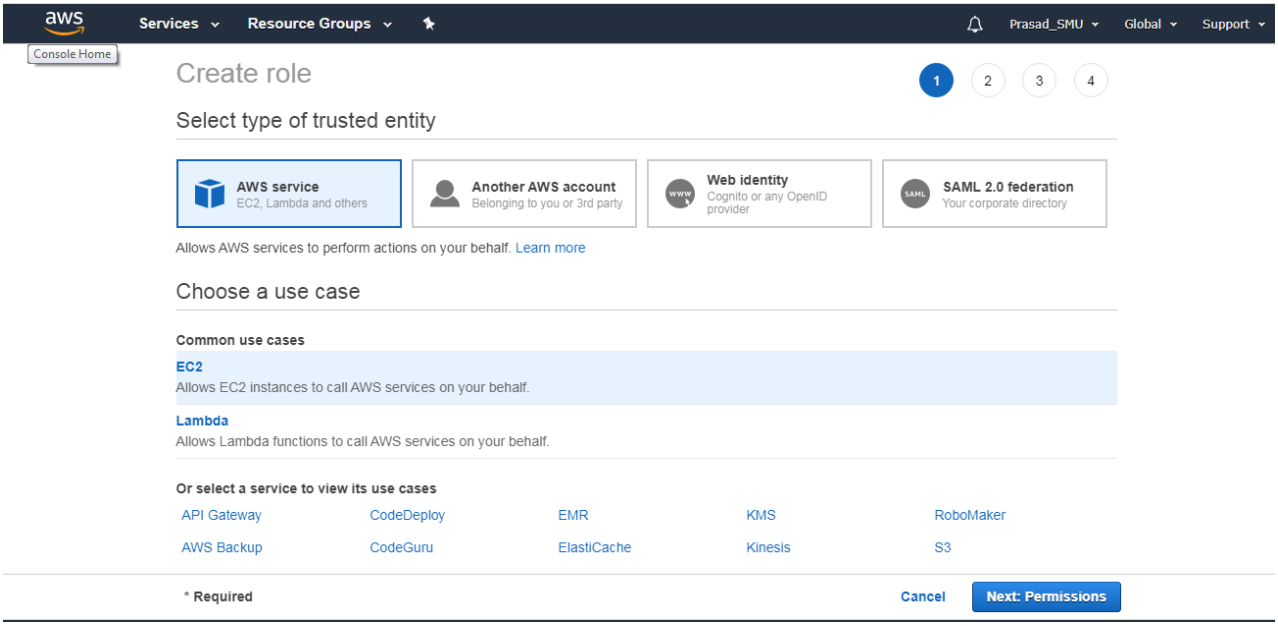
Login to the AWS Management Console.

Navigate to IAM Service and click on Roles.

Click on Create Role.

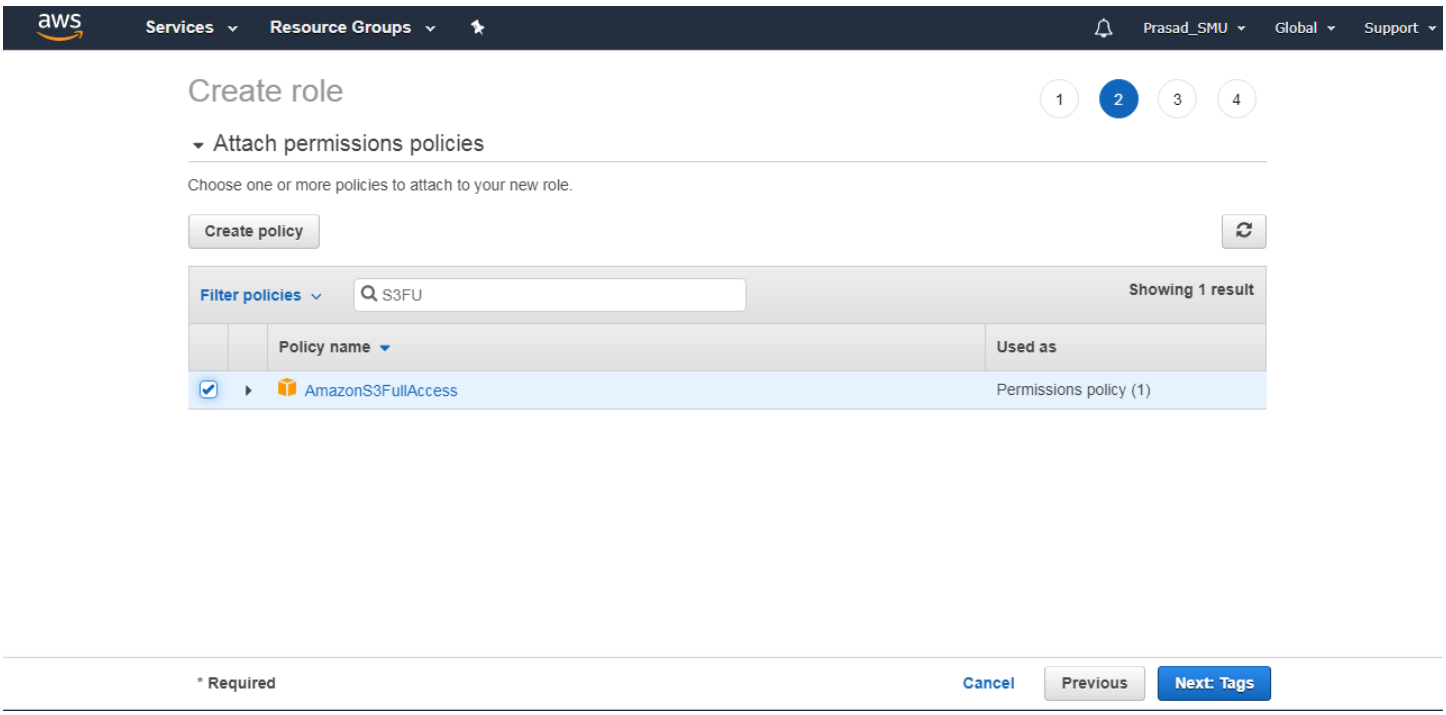
Make sure to select the Use Case as **EC2**.

Click Next: Permissions.

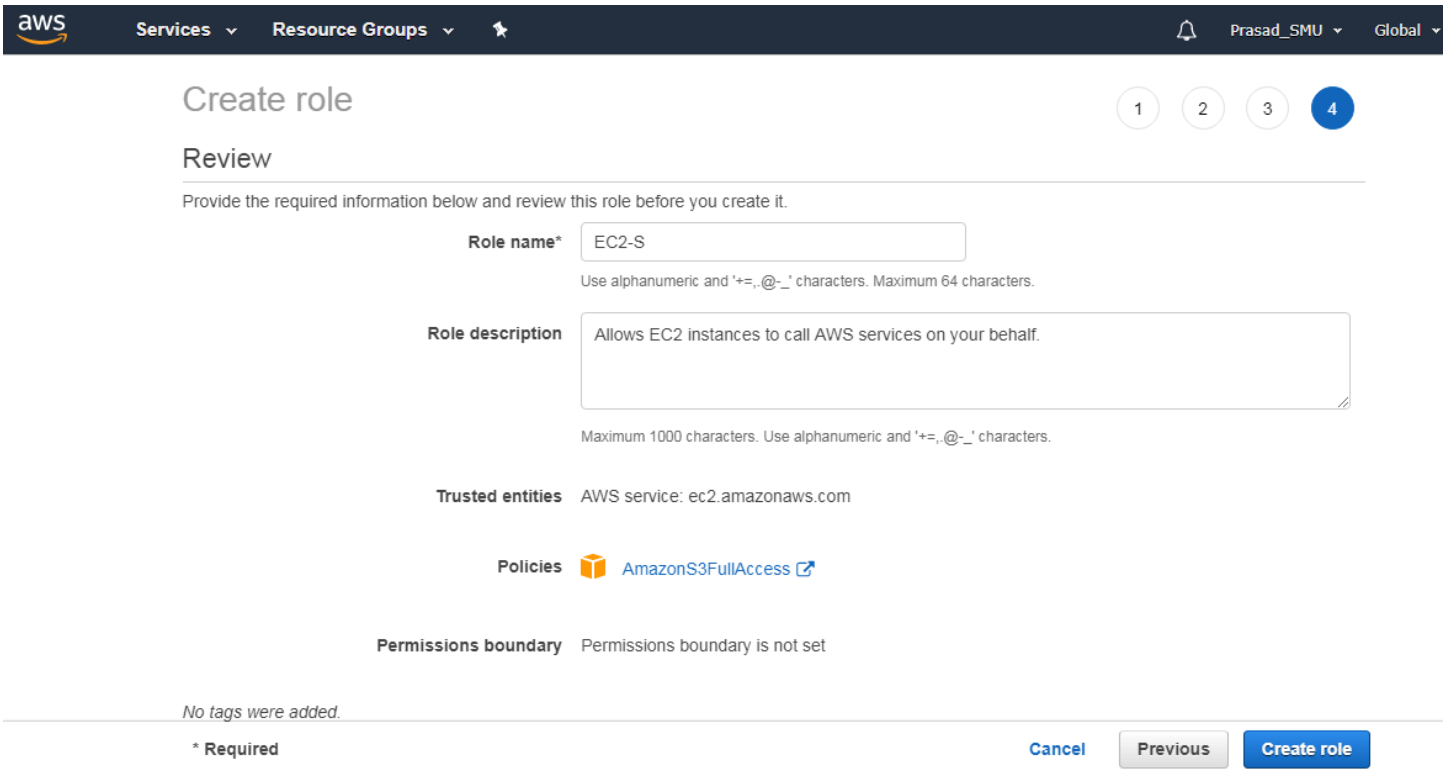


Select the below Default IAM Policies:

AmazonS3FullAccess



Give the Role Name as per your Choice and click on Create Role.

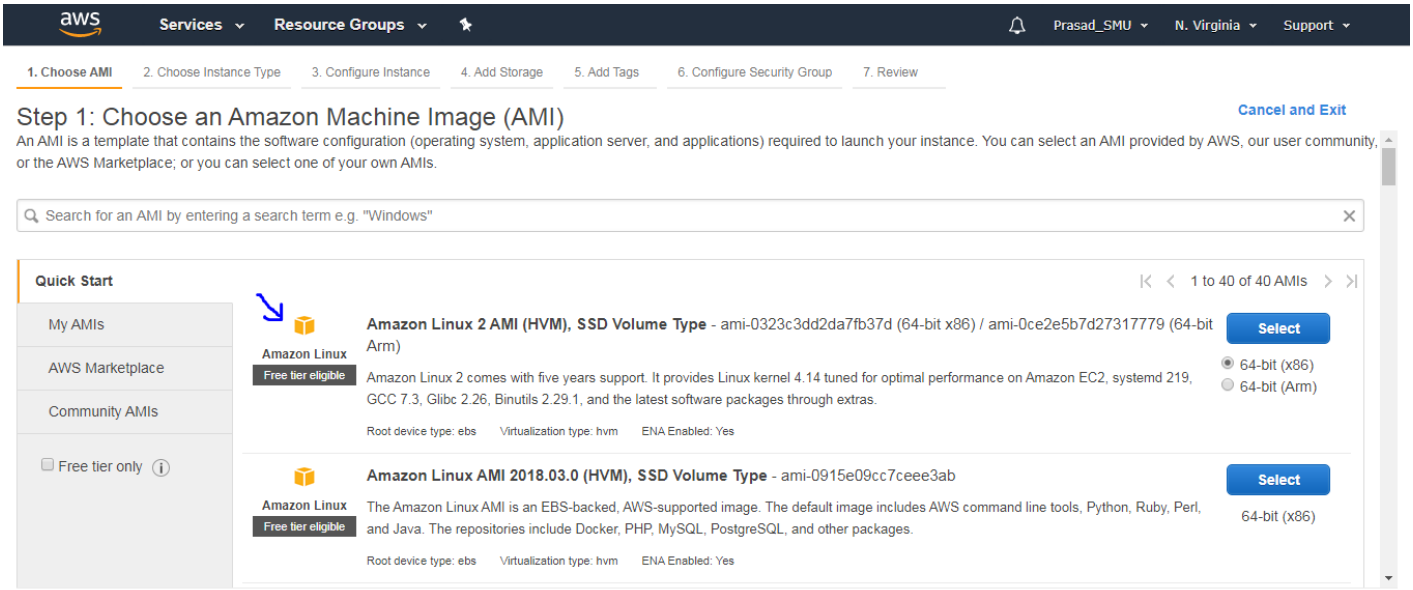


Task 2: Launch & Configure EC2 Instance

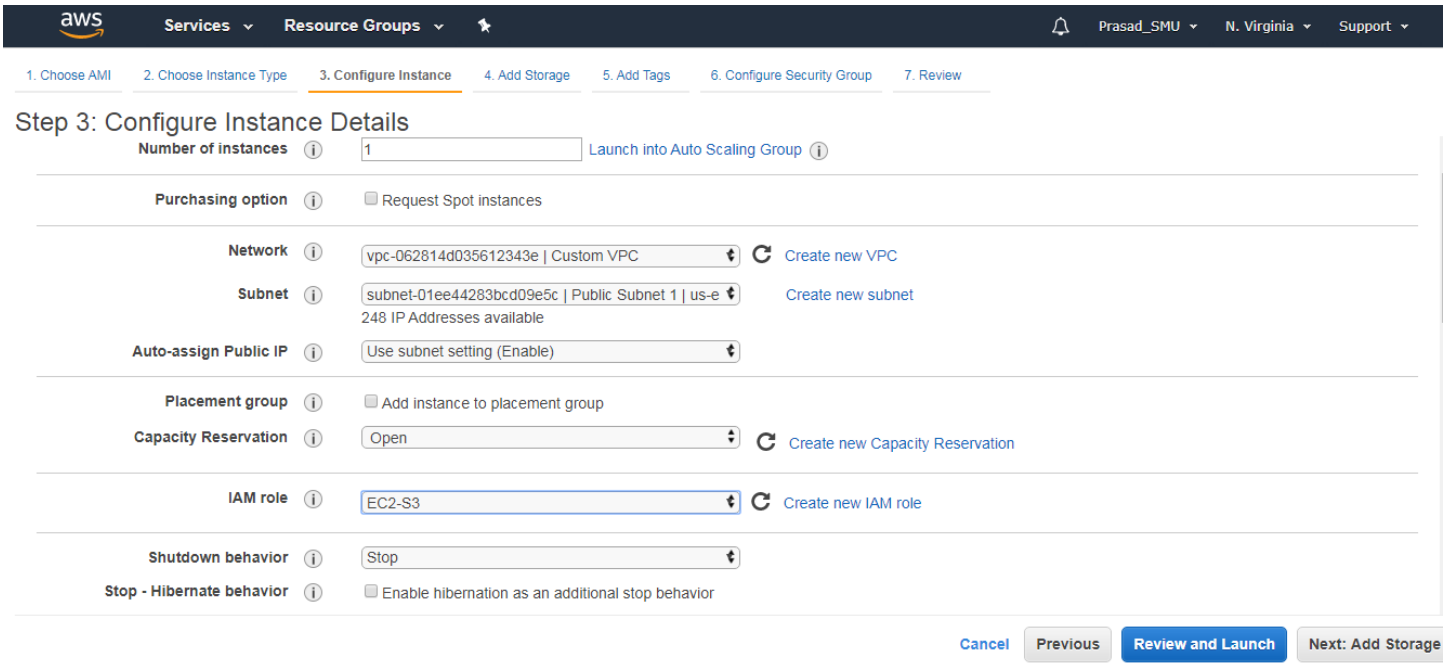
Login to the AWS Management Console.

Navigate to EC2 Service and click on Launch Instance.

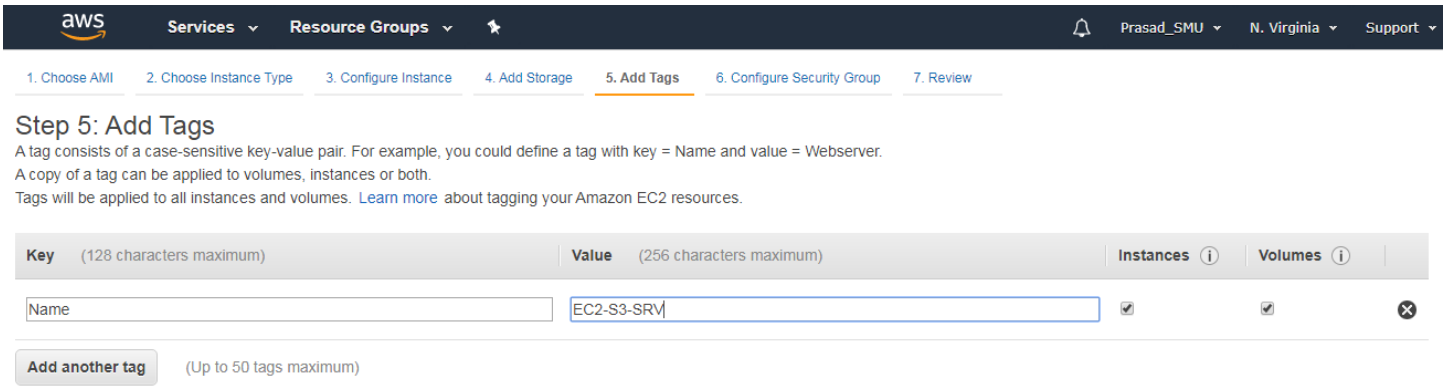
Select the **Amazon Linux AMI**.



Select the Number of Instances as 1, select the Network as our Custom VPC, Select Subnet as Public Subnet 1 and select the IAM Role which you configured in the Task 1.

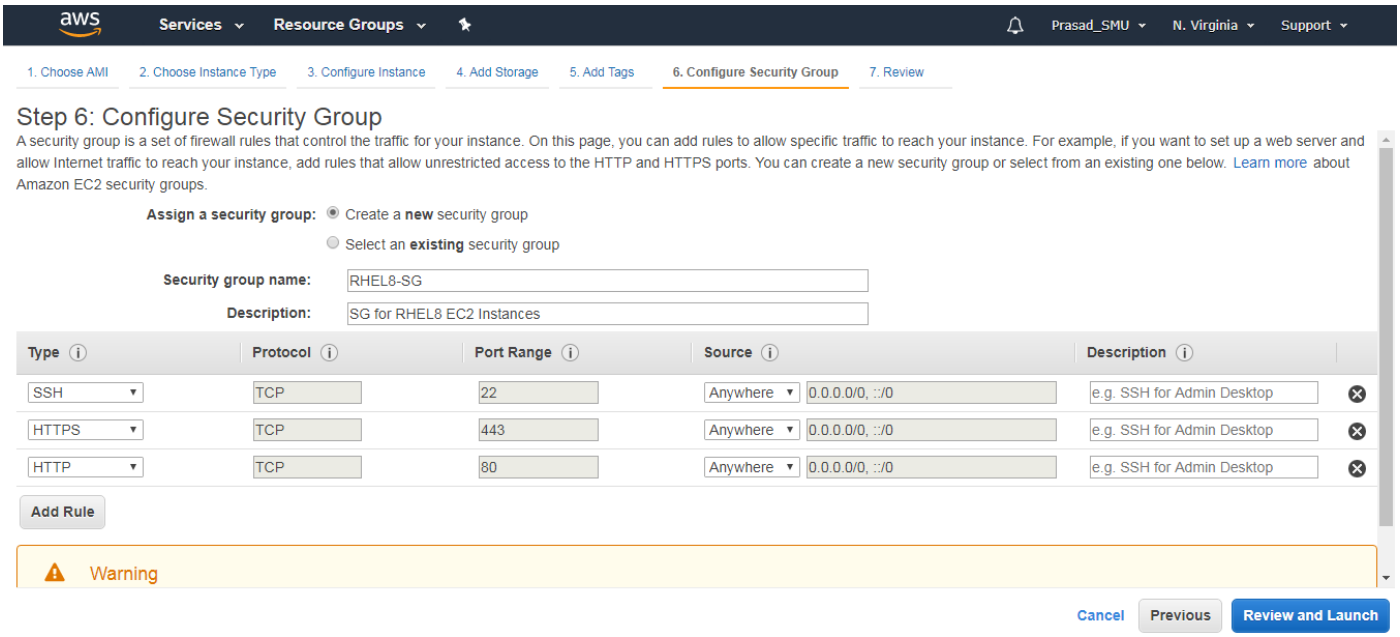


You can mention Tags as per your choice.



Click Next: Security Groups.

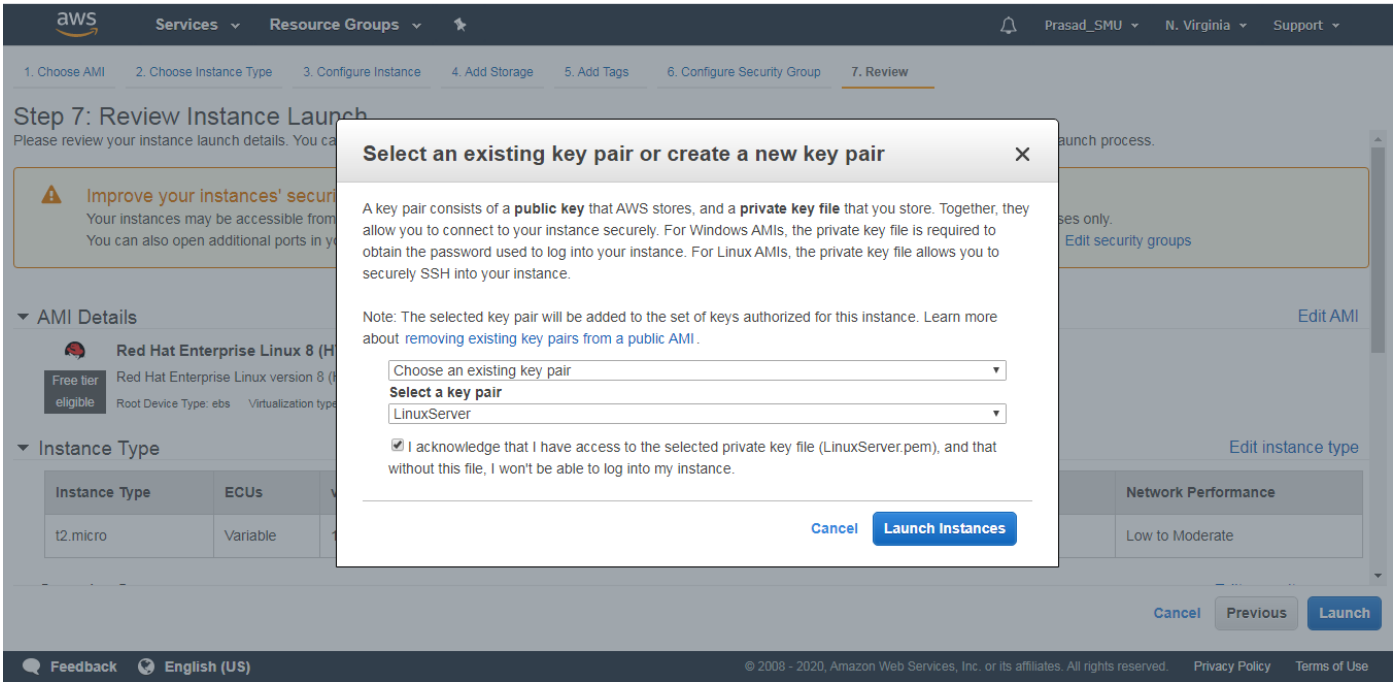
Create a new Security Group. Give the Name & Discription as per your choice. Allow SSH, HTTPS, HTTP Inbound traffic from Anywhere.



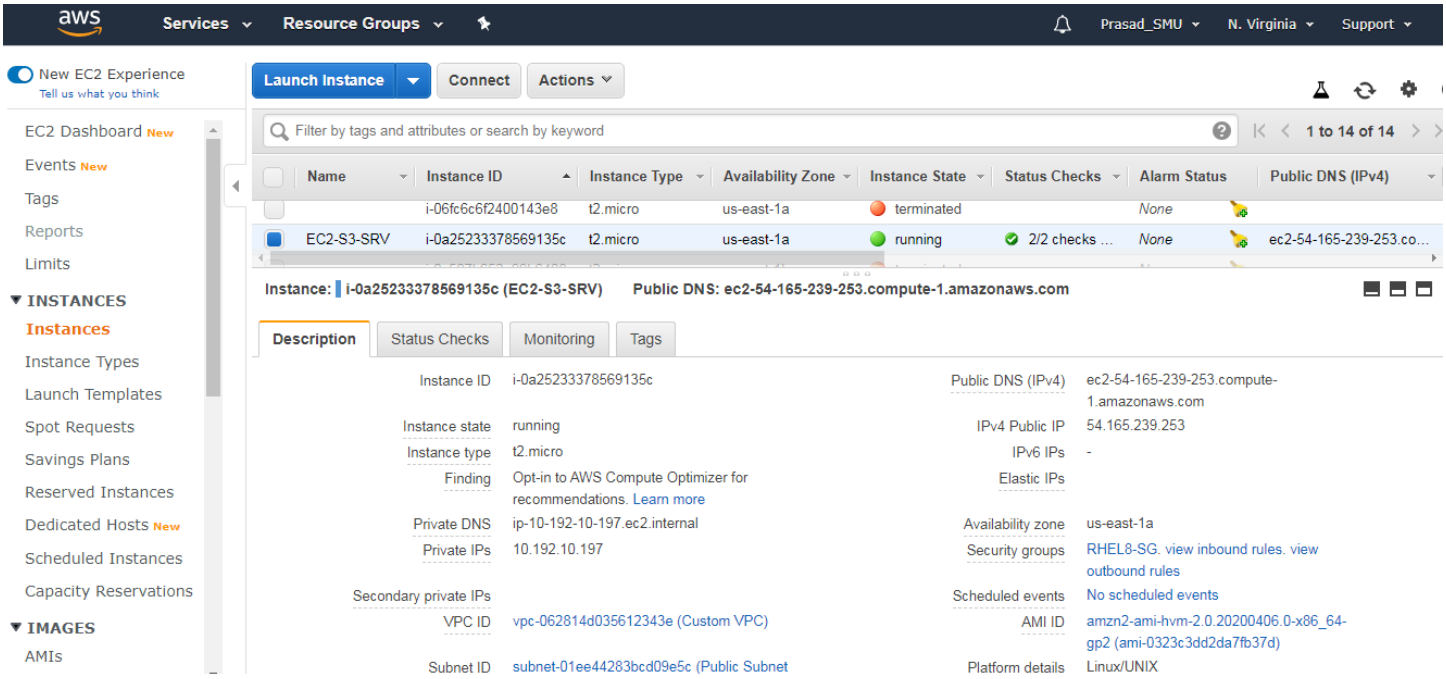
Click on Review and Launch.

Select the existing Key Pair which you've using for previous labs.

Click on Launch Instances.



You can see that the Highlighted Instance has been launched Successfully!!!!!



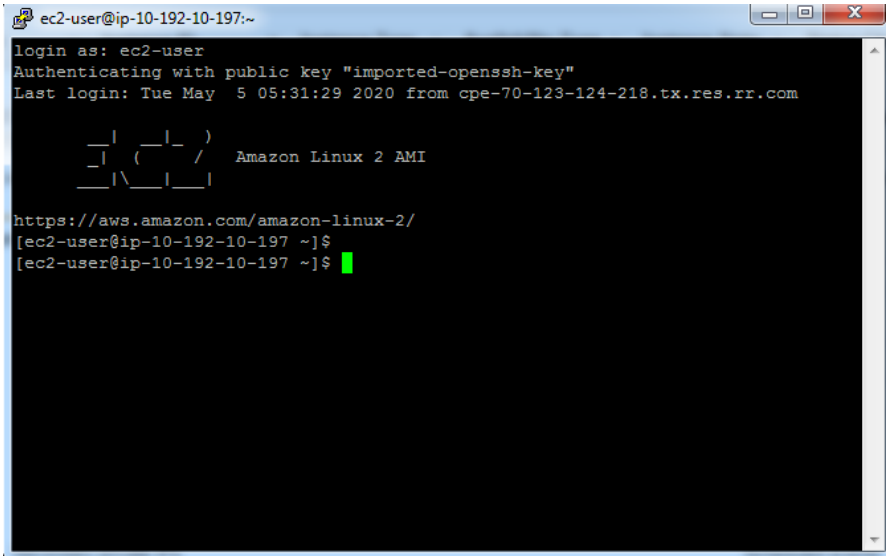
Task 3: Connect to the EC2 Instance

First, using the PuTTYgen, convert the downloaded. pem key to. ppk format.

- Open puttygen.exe
- In the PuTTY Key Generator panel, choose File > Load private key.
- At the bottom of the Load private key panel, click on the drop-down menu that displays *PuTTY Private Key Files (.ppk) and choose All Files**.
- Still in same panel, browse to the directory where you downloaded the .pem file (for example the Downloads directory).
- Select. pem and click Open.
- A PuTTYgen Notice screen should display, indicating that the key was successfully imported. Click OK.
- Click Save private key, then click Yes to save it without a passphrase.
- Give it the filename and click Save.
- Click the X at the top right of the PuTTY Key Generator to close it.

To connect to your Linux EC2 Instance, follow the below steps.

- Open PuTTY software.
- Give the Hostname as the Public IP Address of the Linux EC2 Instance.
- Click on Connections, then click on SSH, then click on Auth.
- Browse the .ppk file and hit Open.
- For certificates validation, click on YES.
- Provide the username as “ec2-user”.



Task 4: S3 Bucket Administration using AWS CLI

By default, AWS packages are included in Linux based EC2 Instance.

Command: aws (double tab)

```
[root@ip-10-192-10-197 ~]# aws
aws            aws_completer
[root@ip-10-192-10-197 ~]# aws
```

Switch to ROOT user.

Command: sudo su –

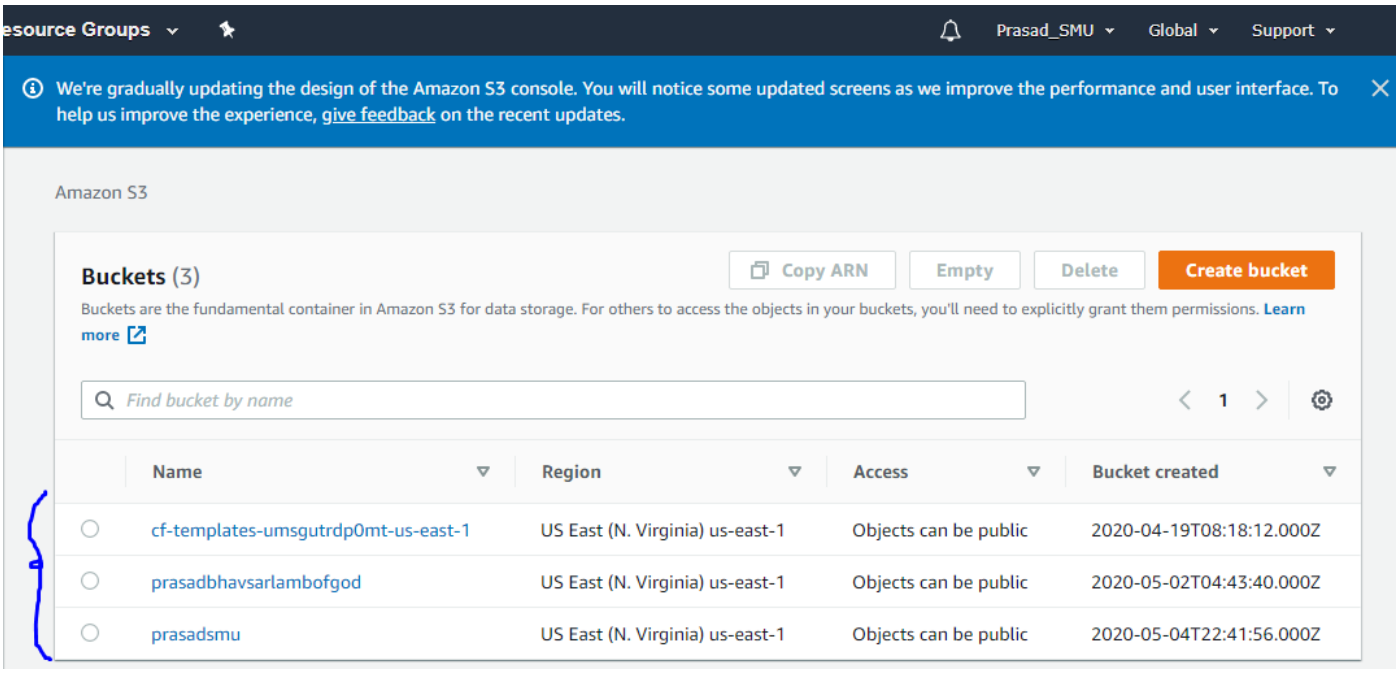
```
[ec2-user@ip-10-192-10-197 ~]$ sudo su -
Last login: Tue May  5 05:31:56 UTC 2020 on pts/0
[root@ip-10-192-10-197 ~]#
```

To list the existing S3 Buckets in your account, run the below command.

Command: aws s3 ls

```
[root@ip-10-192-10-197 ~]# aws s3 ls
2020-04-19 08:18:12 cf-templates-umsgutrdp0mt-us-east-1
2020-05-02 04:43:40 prasadbhavsarlambofgod
2020-05-04 22:41:56 prasadsmu
[root@ip-10-192-10-197 ~]#
```

You can verify the S3 Buckets from AWS Management Console.

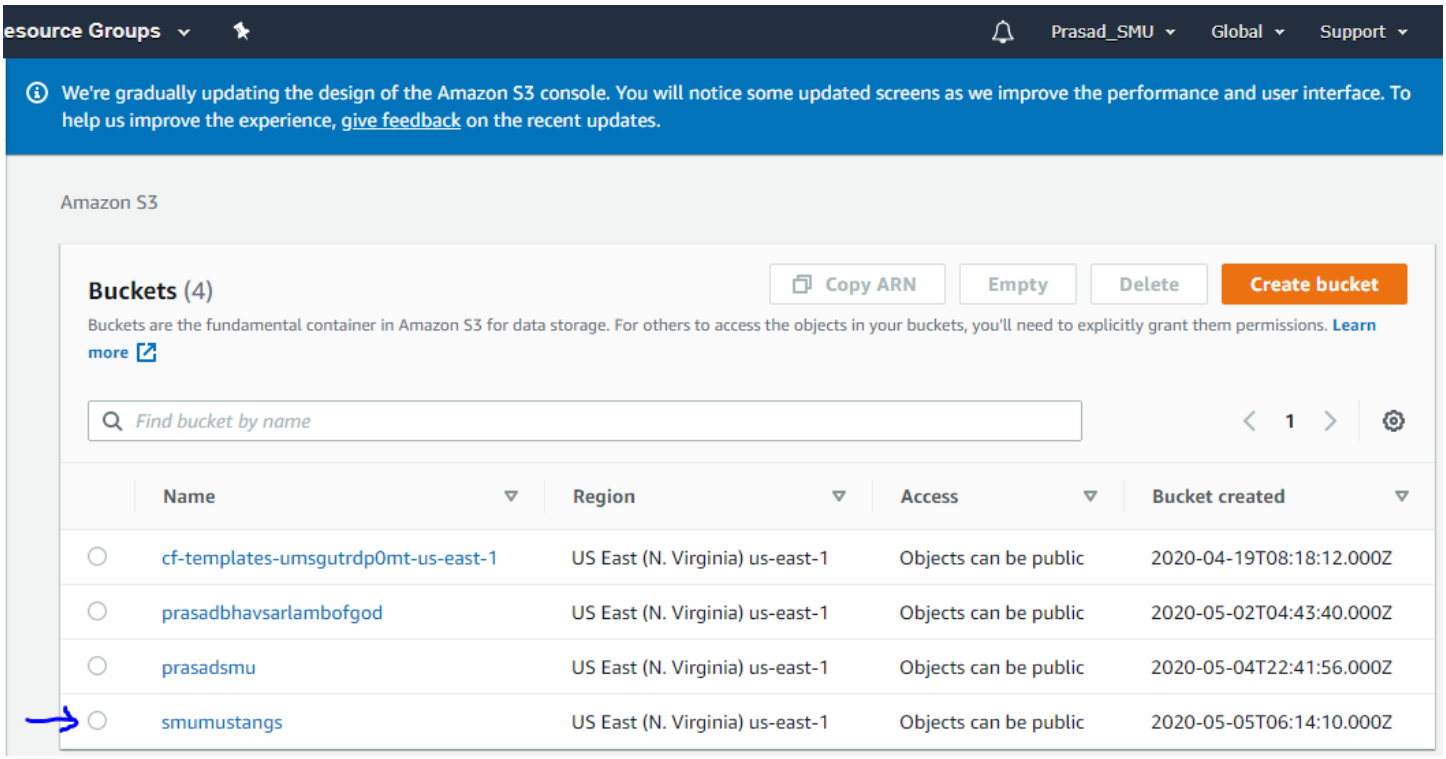


To create a new Bucket, run the below commands.

Command: aws s3 mb s://smumustangs

```
[root@ip-10-192-10-197 ~]# aws s3 mb s3://smumustangs
make_bucket: smumustangs
[root@ip-10-192-10-197 ~]#
```

You can verify the new S3 Bucket from AWS Management Console.

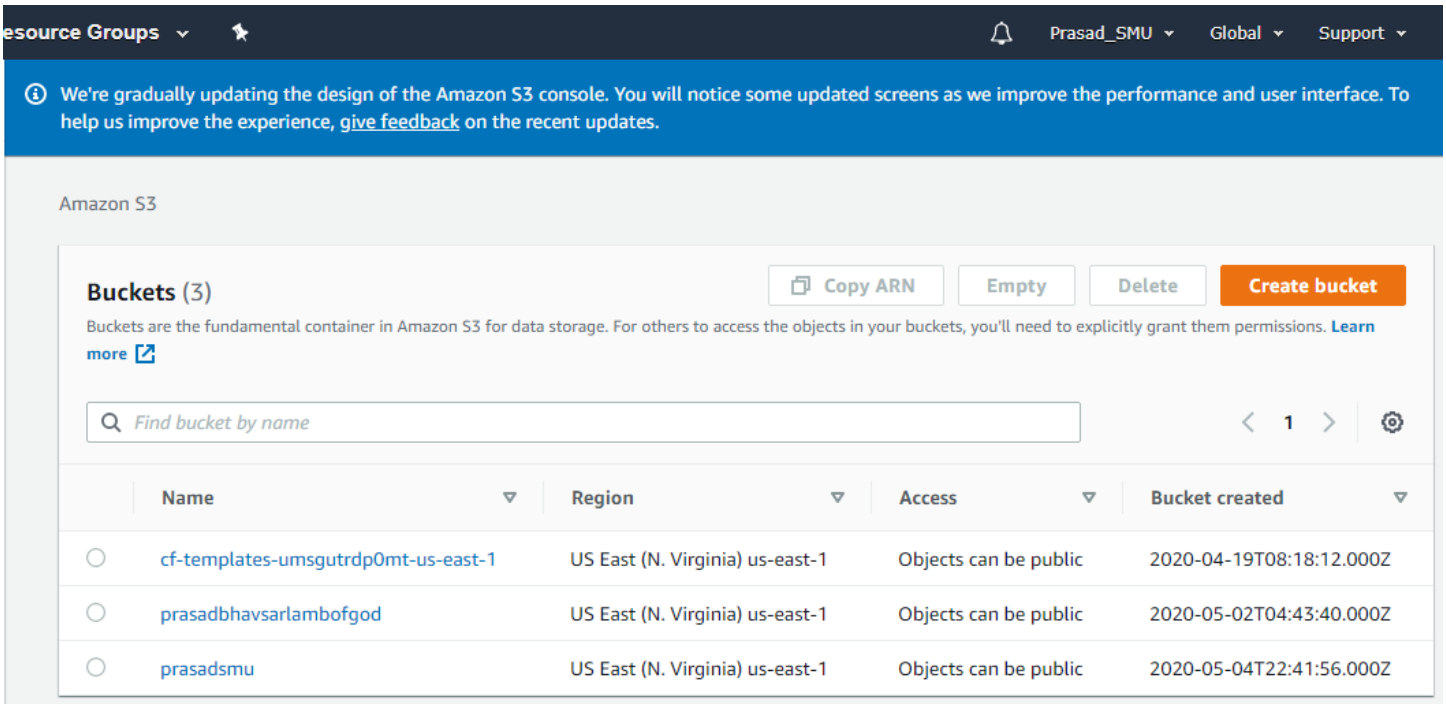


To remove a S3 Bucket, run the below commands.

Command: aws s3 rb s://smumustangs

```
[root@ip-10-192-10-197 ~]# aws s3 rb s3://smumustangs
remove_bucket: smumustangs
[root@ip-10-192-10-197 ~]#
[root@ip-10-192-10-197 ~]# aws s3 ls
2020-04-19 08:18:12 cf-templates-umsgutrdp0mt-us-east-1
2020-05-02 04:43:40 prasadbhavsarlambofgod
2020-05-04 22:41:56 prasadsmu
[root@ip-10-192-10-197 ~]#
```

You can verify the S3 Bucket from AWS Management Console.



Task 5: Mount S3 Buckets to EC2 Instance using AWS CLI

Let's Install necessary packages to mount s3 Bucket.

First, list all the Mount Points.

Command: df -h

```
[root@ip-10-192-10-197 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        475M     0  475M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  436K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.7G  6.4G  21% /
tmpfs            99M     0   99M   0% /run/user/1000
[root@ip-10-192-10-197 ~]#
```

To install FUSE packages, run the below set of commands one-by-one.

Command: yum -y install automate fuse fuse-devel gcc-c++ libcurl-devel libxml12-devel make openssl-devel

```
[root@ip-10-192-10-197 ~]# yum -y install automate fuse fuse-devel gcc-c++ libcurl-devel libxml12-
devel make openssl-devel
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                               | 2.4 kB  00:00:00
amzn2extra-docker                                                        | 1.8 kB  00:00:00
epel/x86_64/metalink                                                     |  19 kB  00:00:00
```

Command: git clone https://github.com/s3fs-fuse/s3fs-fuse

```
[root@ip-10-192-10-197 ~]# git clone https://github.com/s3fs-fuse/s3fs-fuse
Cloning into 's3fs-fuse'...
remote: Enumerating objects: 50, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 5995 (delta 24), reused 29 (delta 12), pack-reused 5945
```

Commands:

cd s3fs-fuse/

./autogen.sh

```
[root@ip-10-192-10-197 ~]# cd s3fs-fuse/
[root@ip-10-192-10-197 s3fs-fuse]# ./autogen.sh
--- Make commit hash file -------
--- Finished commit hash file ---
--- Start autotools -------------
configure.ac:26: installing './config.guess'
configure.ac:26: installing './config.sub'
configure.ac:27: installing './install-sh'
configure.ac:27: installing './missing'
src/Makefile.am: installing './depcomp'
parallel-tests: installing './test-driver'
```

Command: **.**/configure

```
[root@ip-10-192-10-197 s3fs-fuse]# ./configure
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
```

Command: make

```
[root@ip-10-192-10-197 s3fs-fuse]# make
make  all-recursive
make[1]: Entering directory `/root/s3fs-fuse'
Making all in src
make[2]: Entering directory `/root/s3fs-fuse/src'
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2
   -g -O2 -Wall -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=2 -MT s3fs.o -MD -MP -MF .deps/s3fs.Tpo -
c -o s3fs.o s3fs.cpp
mv -f .deps/s3fs.Tpo .deps/s3fs.Po
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2
   -g -O2 -Wall -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=2 -MT curl.o -MD -MP -MF .deps/curl.Tpo -
c -o curl.o curl.cpp
```

Command: make install

```
[root@ip-10-192-10-197 s3fs-fuse]# make install
Making install in src
make[1]: Entering directory `/root/s3fs-fuse/src'
make[2]: Entering directory `/root/s3fs-fuse/src'
 /usr/bin/mkdir -p '/usr/local/bin'
  /usr/bin/install -c s3fs '/usr/local/bin'
```

FUSE Package has been successfully installed.

Now to create a Mountpoint run the below command.

Command: mkdir -p /var/s3fs-demo-fs

```
[root@ip-10-192-10-197 s3fs-fuse]# mkdir -p /var/s3fs-demo-fs
[root@ip-10-192-10-197 s3fs-fuse]#
```

Let's now again create a S3 Bucket with the name "smumustangs".

```
[root@ip-10-192-10-197 s3fs-fuse]#
[root@ip-10-192-10-197 s3fs-fuse]# aws s3 mb s3://smumustangs
make_bucket: smumustangs
[root@ip-10-192-10-197 s3fs-fuse]#
```

Mount the S3 Bucket "smumustangs" on the mount point of EC2 Instance "s3fs-demo-fs" by specifying the IAM Role.

Command: s3fs smumustangs /var/s3fs-demo-fs -o iam_role=EC2-S3

```
[root@ip-10-192-10-197 s3fs-fuse]# s3fs smumustangs /var/s3fs-demo-fs -o iam_role=EC2-S3
[root@ip-10-192-10-197 s3fs-fuse]#
[root@ip-10-192-10-197 s3fs-fuse]#
```

To verify the Mount Points, run the below command. You'll notice that the S3 Bucket has been mounted on the EC2 Instance's mount point.

Command: df -h

```
[root@ip-10-192-10-197 s3fs-fuse]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        475M     0  475M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  440K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.9G  6.2G  24% /
tmpfs            99M     0   99M   0% /run/user/1000
s3fs            256T     0  256T   0% /var/s3fs-demo-fs
[root@ip-10-192-10-197 s3fs-fuse]#
[root@ip-10-192-10-197 s3fs-fuse]#
```

Go to the Mount Point.

Command: cd /var/s3fs-demo-fs

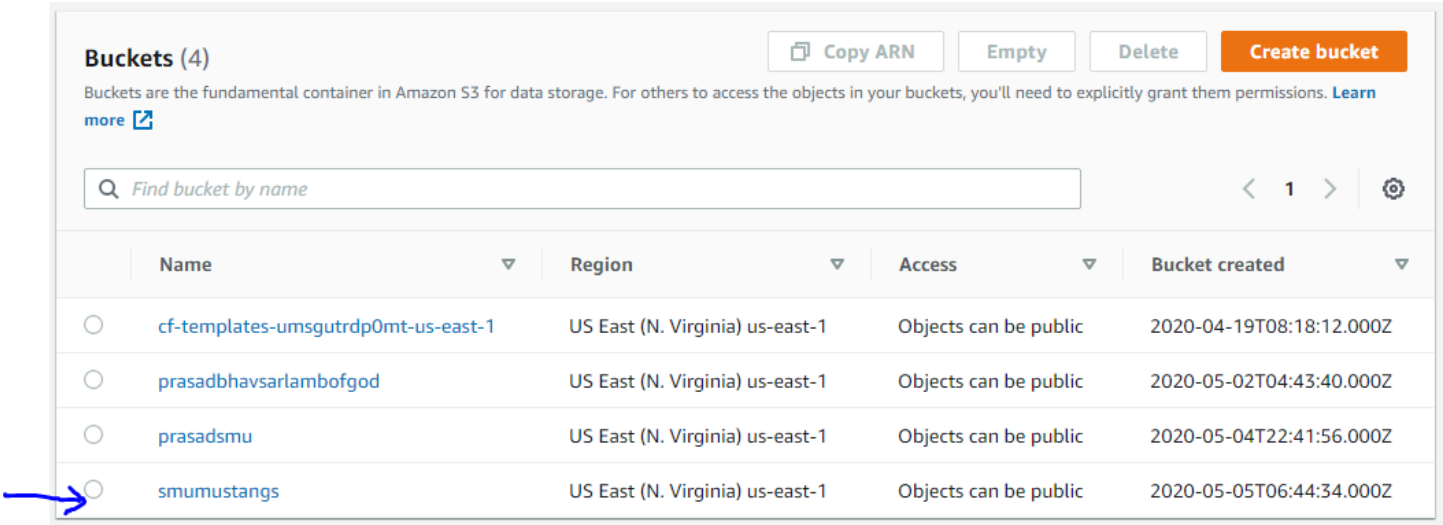
```
[root@ip-10-192-10-197 s3fs-fuse]# cd /var/s3fs-demo-fs
[root@ip-10-192-10-197 s3fs-demo-fs]#
```

Create a dummy file inside the Directory "/var/s3fs-demo-fs".

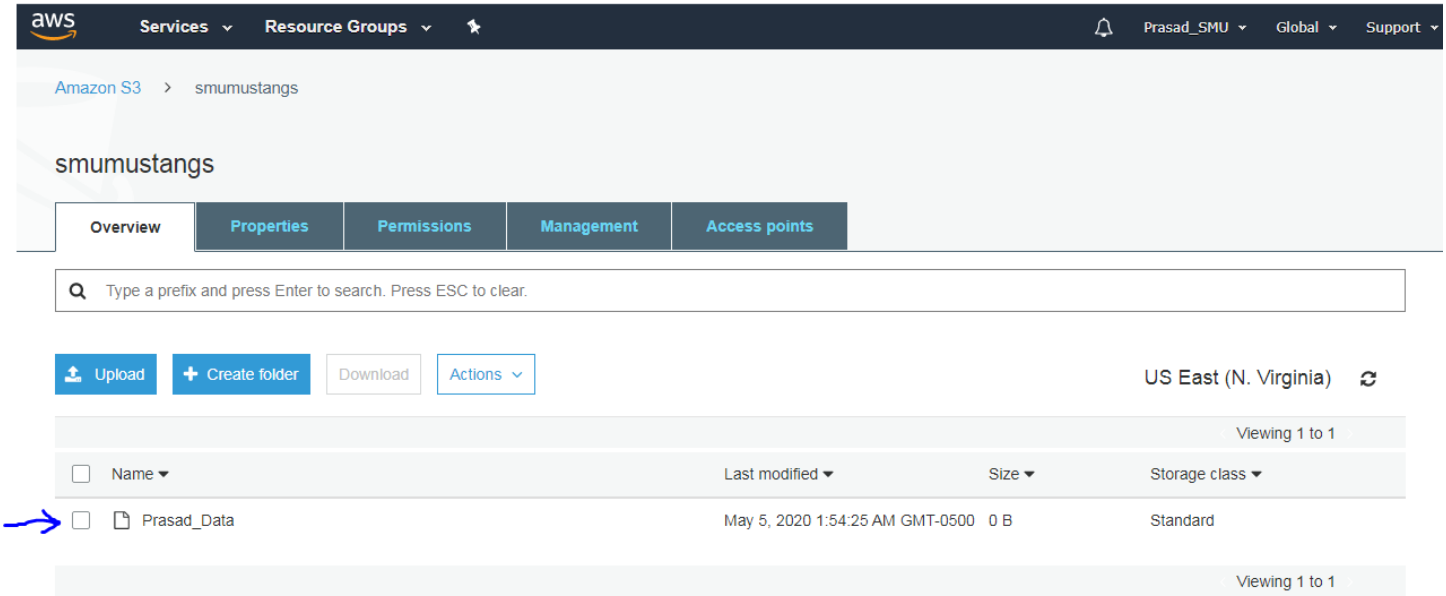
Command: touch Prasad_Data

```
[root@ip-10-192-10-197 s3fs-demo-fs]#
[root@ip-10-192-10-197 s3fs-demo-fs]# touch Prasad_Data
```

Now go to AWS Management Console and go to S3 Service. You'll notice the Bucket is created.



Get inside the Bucket and you'll notice the dummy file which we created is stored in S3 Bucket as S3 Object.



This completes the Lab on S3 Bucket Administration from EC2 Instance using AWS CLI.

For Questions, contact me one pbhavsar@smu.edu .