LAB 3 working with EC2 and EFS

Objective

1. Create a security group with NFS protocol open

2. Create a EFS with above security group and make sure you don’t

enable encryption and Lifecycle management and automated

backups to avoid charges

3. Create an EC2 instance (Ubuntu 16.04 LTS) using the 7 step

workflow in 2 different availability zones(1a and 1b)

4. Login into 2 EC2 instances and install required packages

5. Mount to EFS and start creating files to check if the data is sharing

between 2 instances

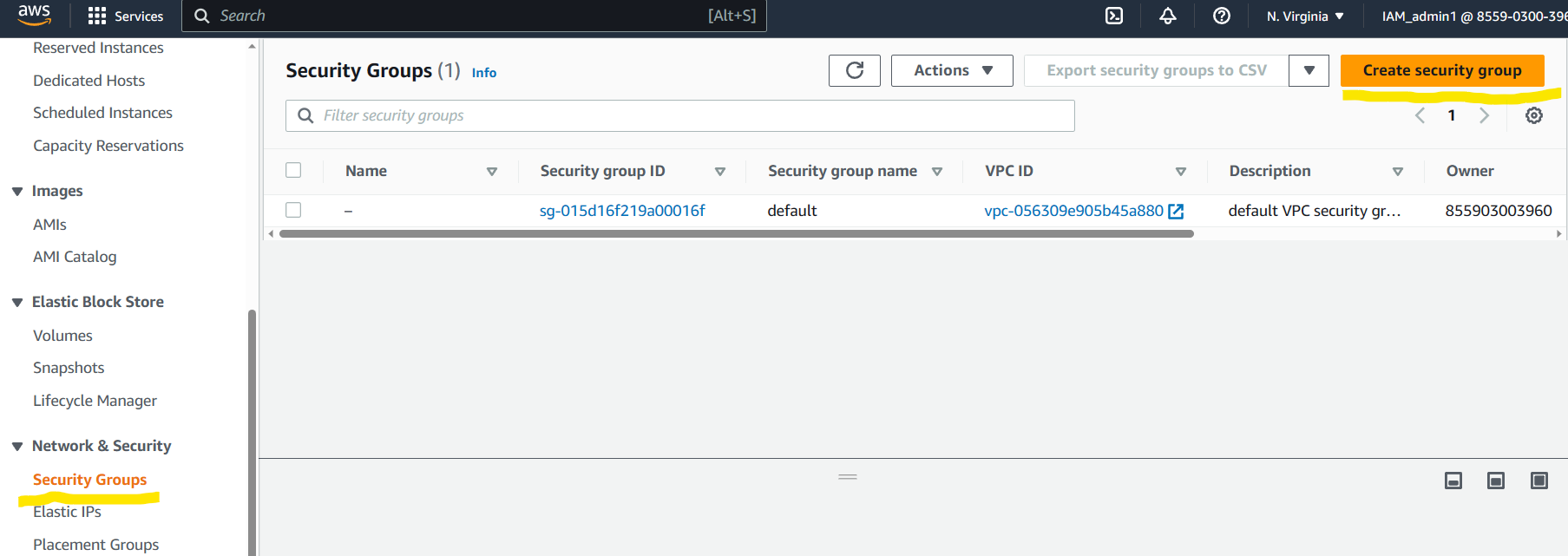
6. Cleanup

 Terminate 2 EC2 instances

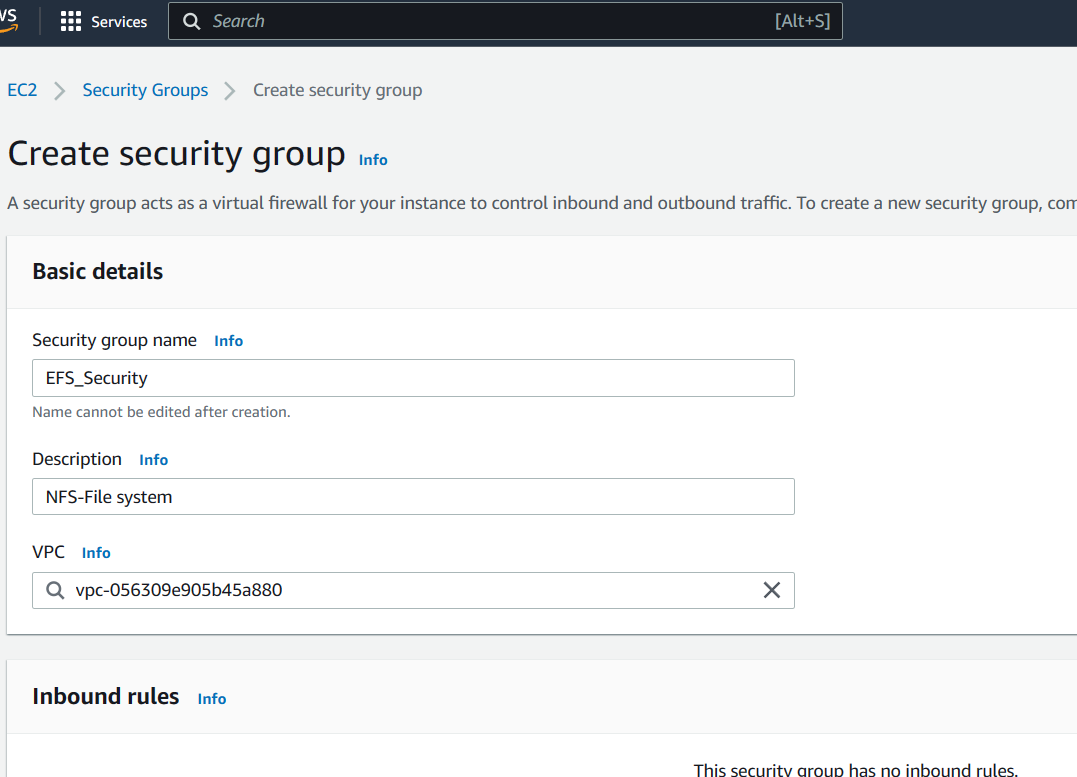
 Delete EFS

To create a security group visit the **Security group** menu in the AWS management console under **Network and security** and click on **Create Security group**

Note: ignore the default security group



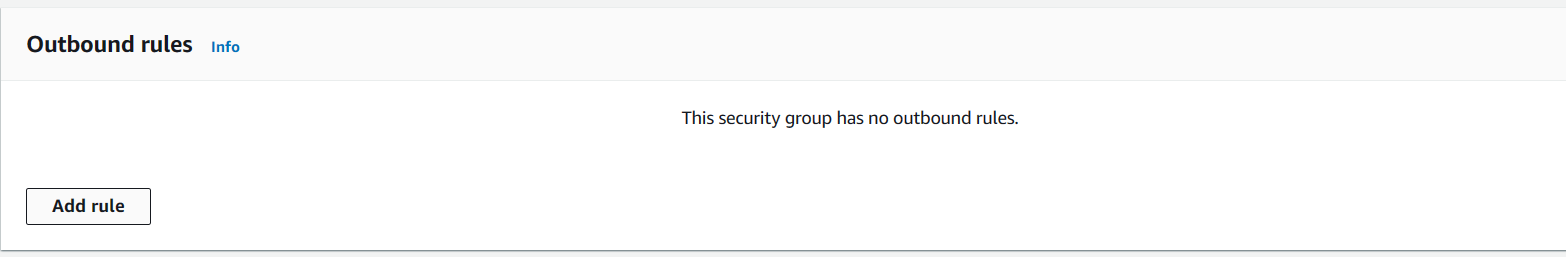
Fill the Security group name field and description



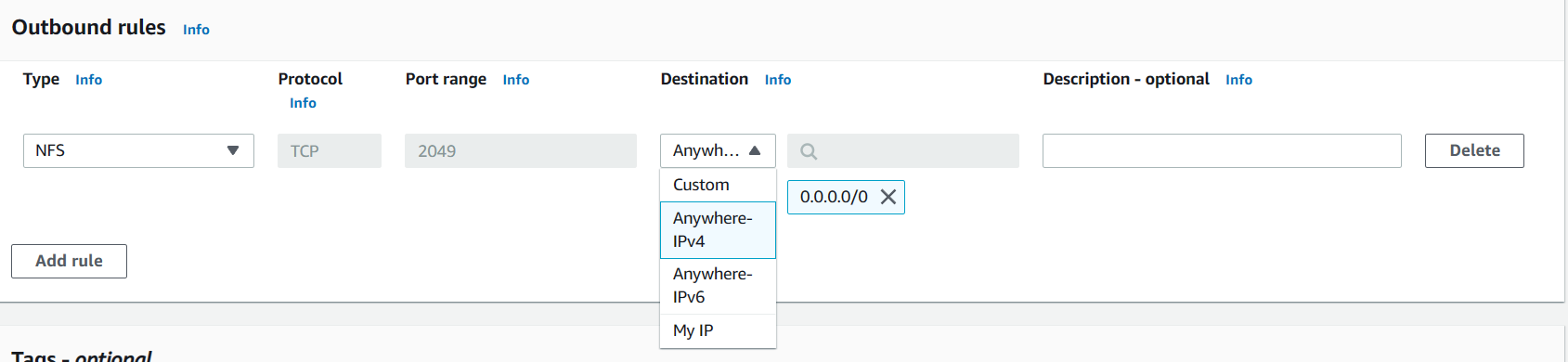
Scroll down to Outbound rules

Remove the existing rules and click on add rules button

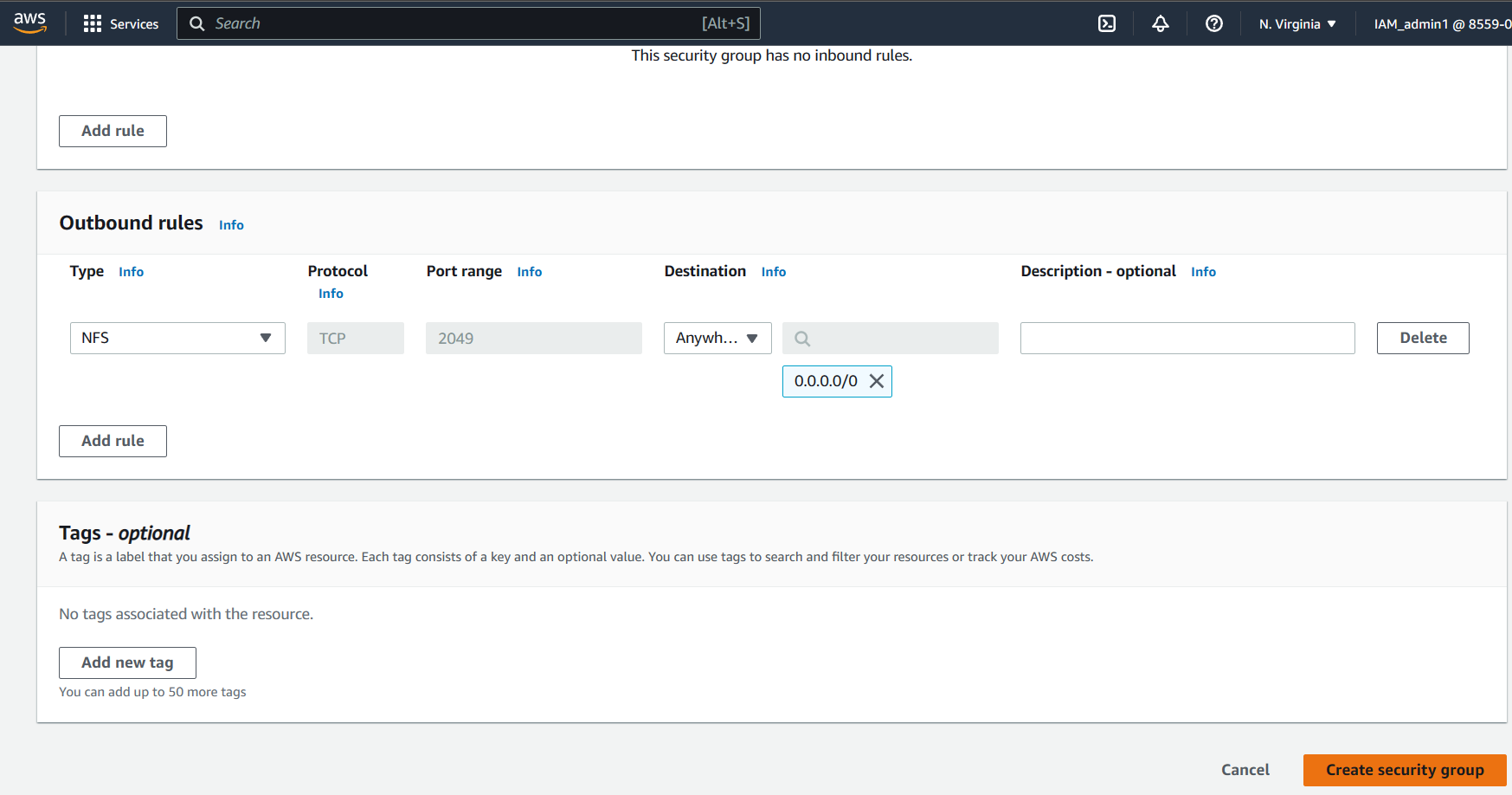
Click on add rules



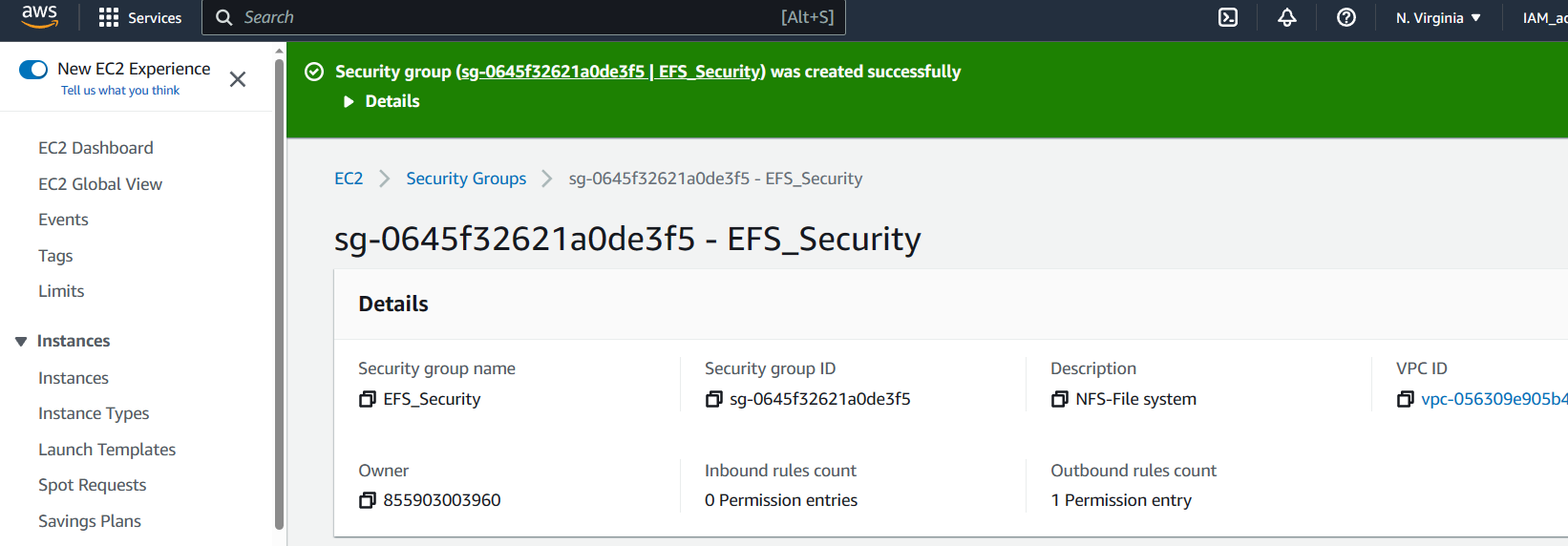
Set the type as “NFS” and destination as “Anywhere ipv4”



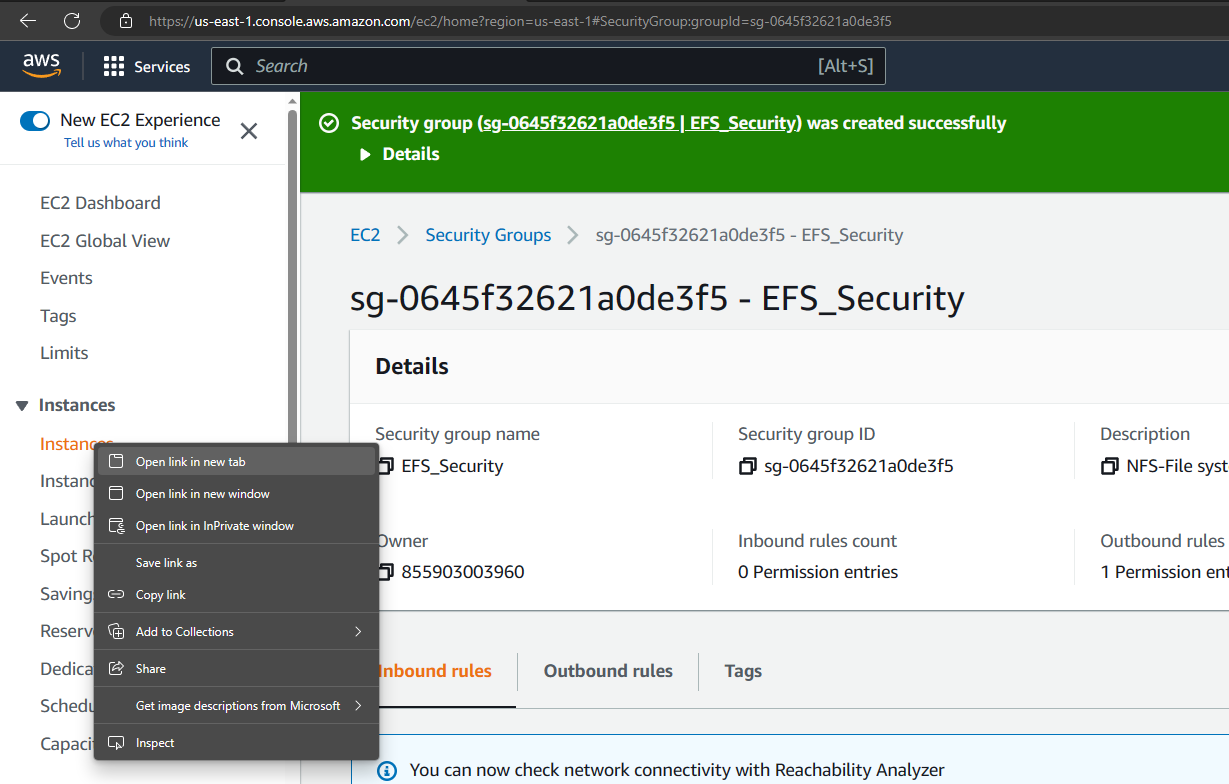
Scroll down and click on create security group



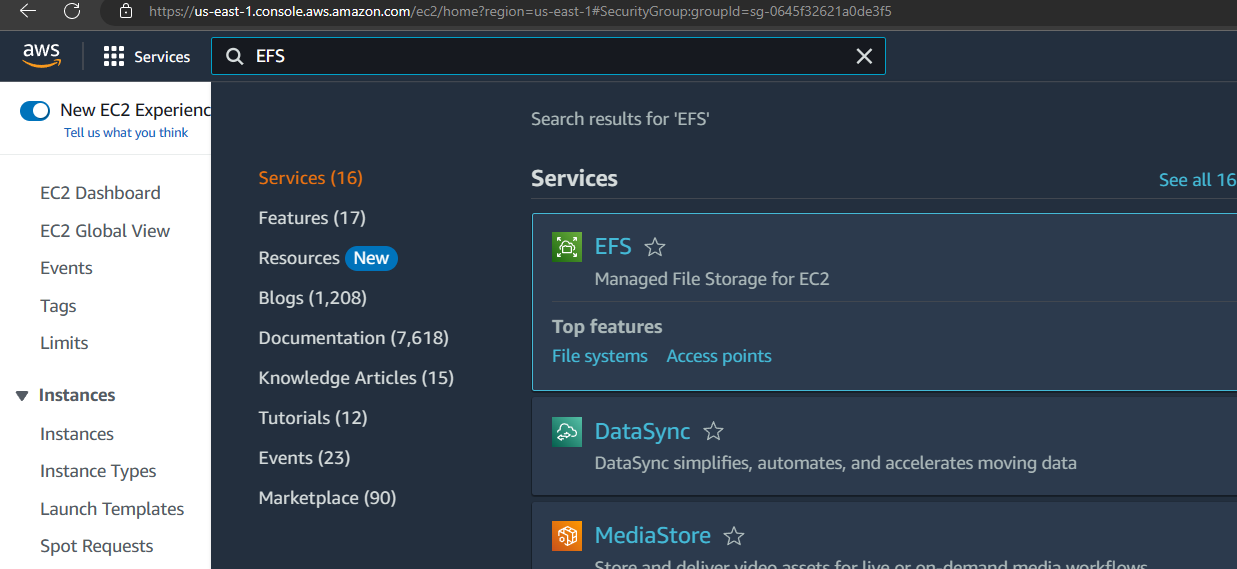
Congratulations you have successfully created a security group



Open a new tab with Instance page loaded in it.



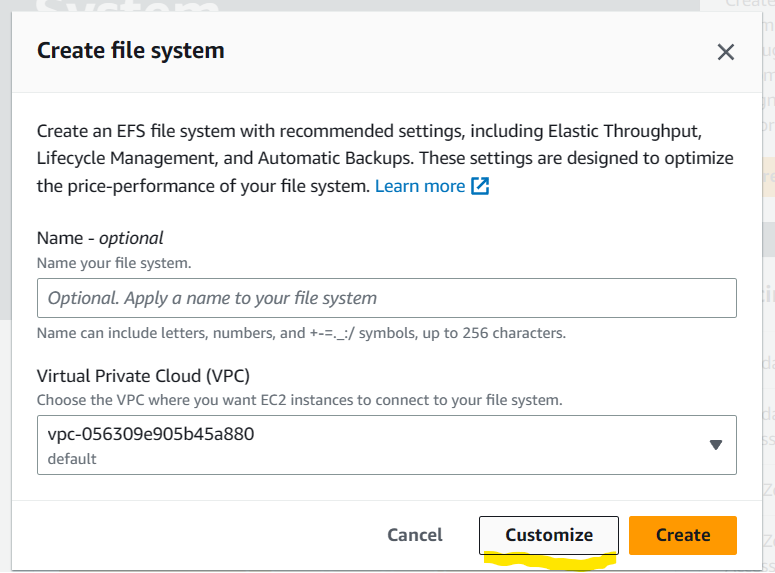
Go back to the previous tab in which the security group page was loaded and navigate to Elastic file storage services



After the File system services page is loaded click on the **create file system** button shown below



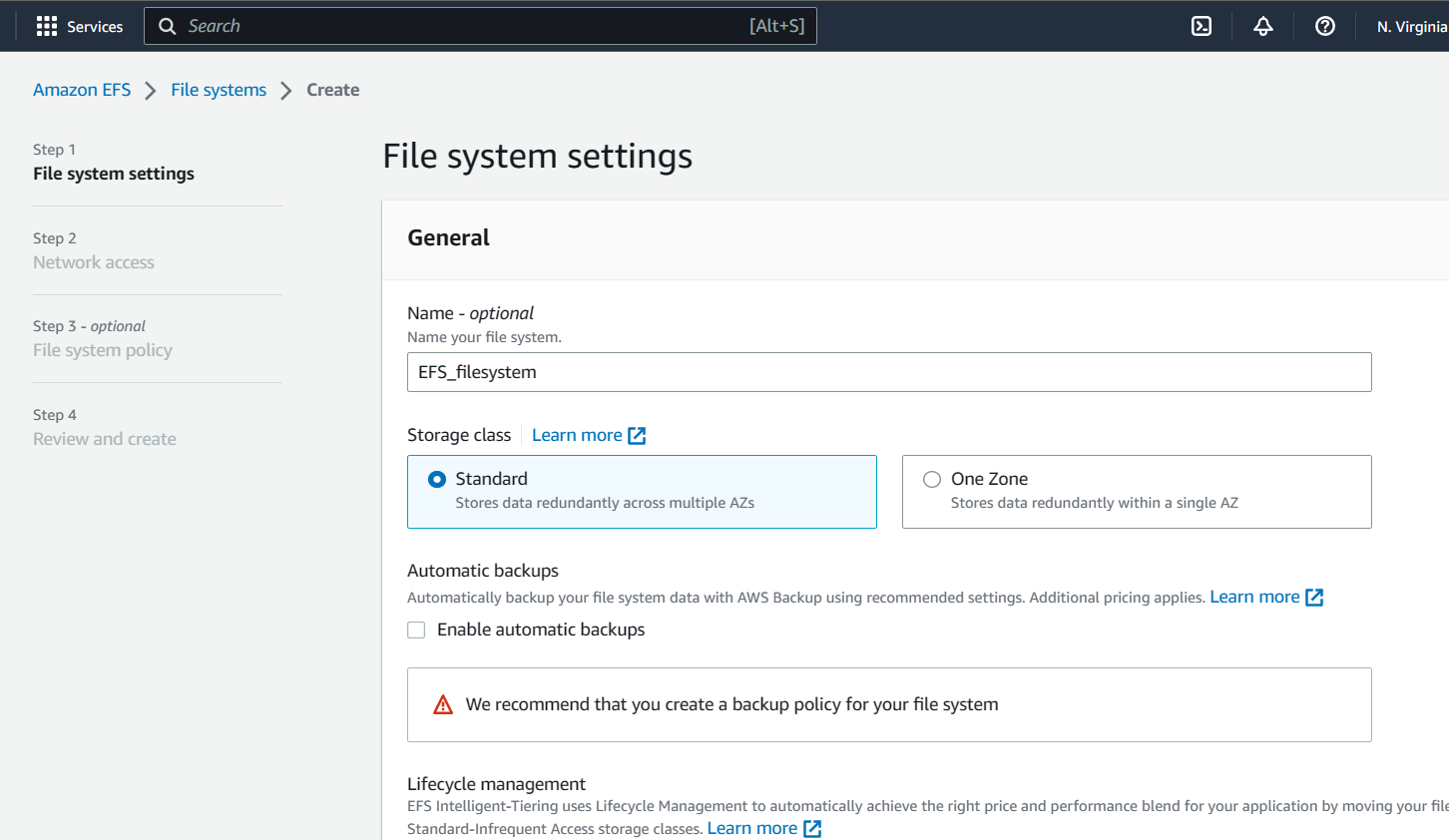
Click on the **customize** button when the pop up appears



A file system settings page would load on the browser

Fill in the File system with a desired name.

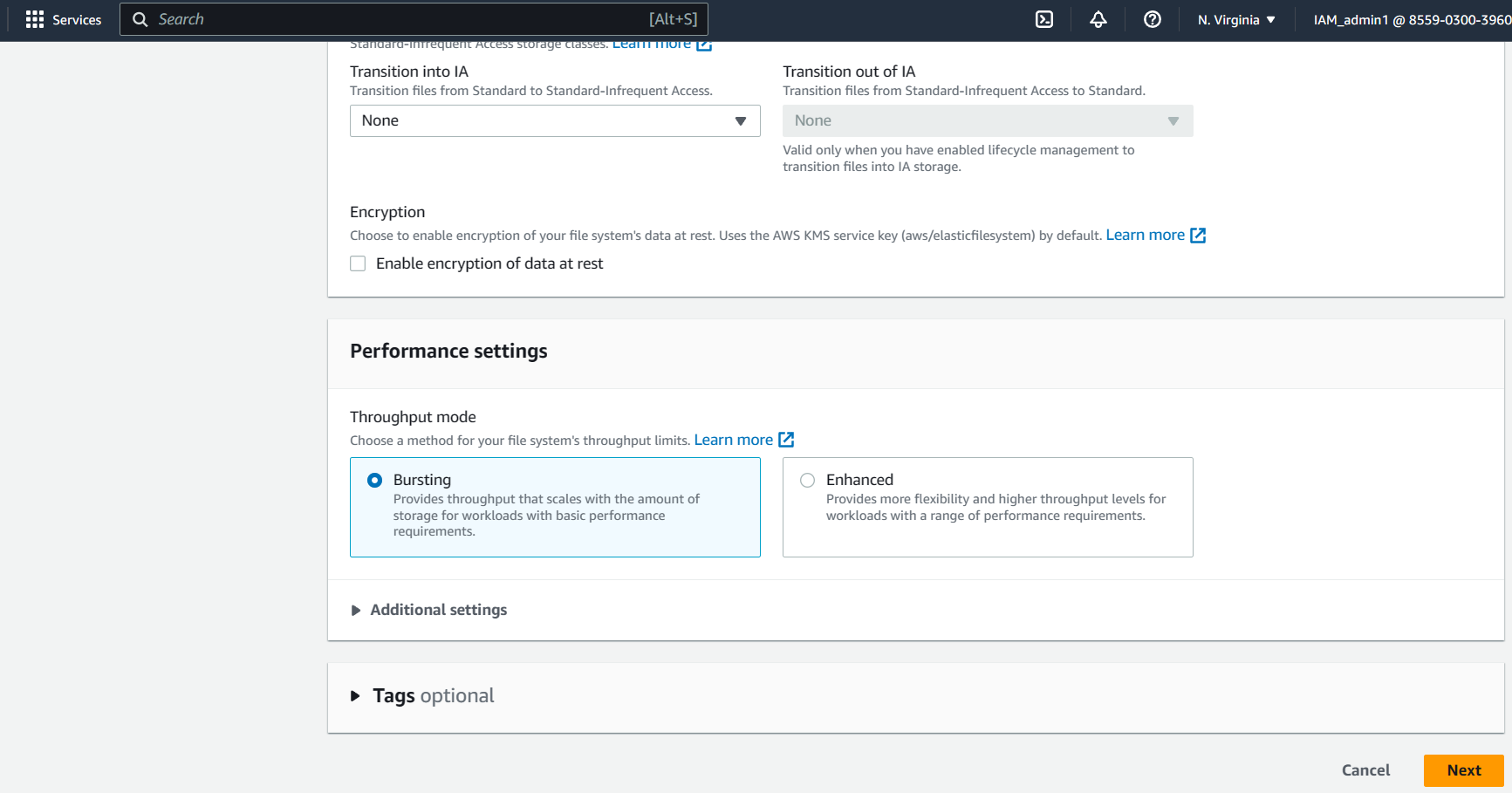
Uncheck the “enable automatic backup” check box.



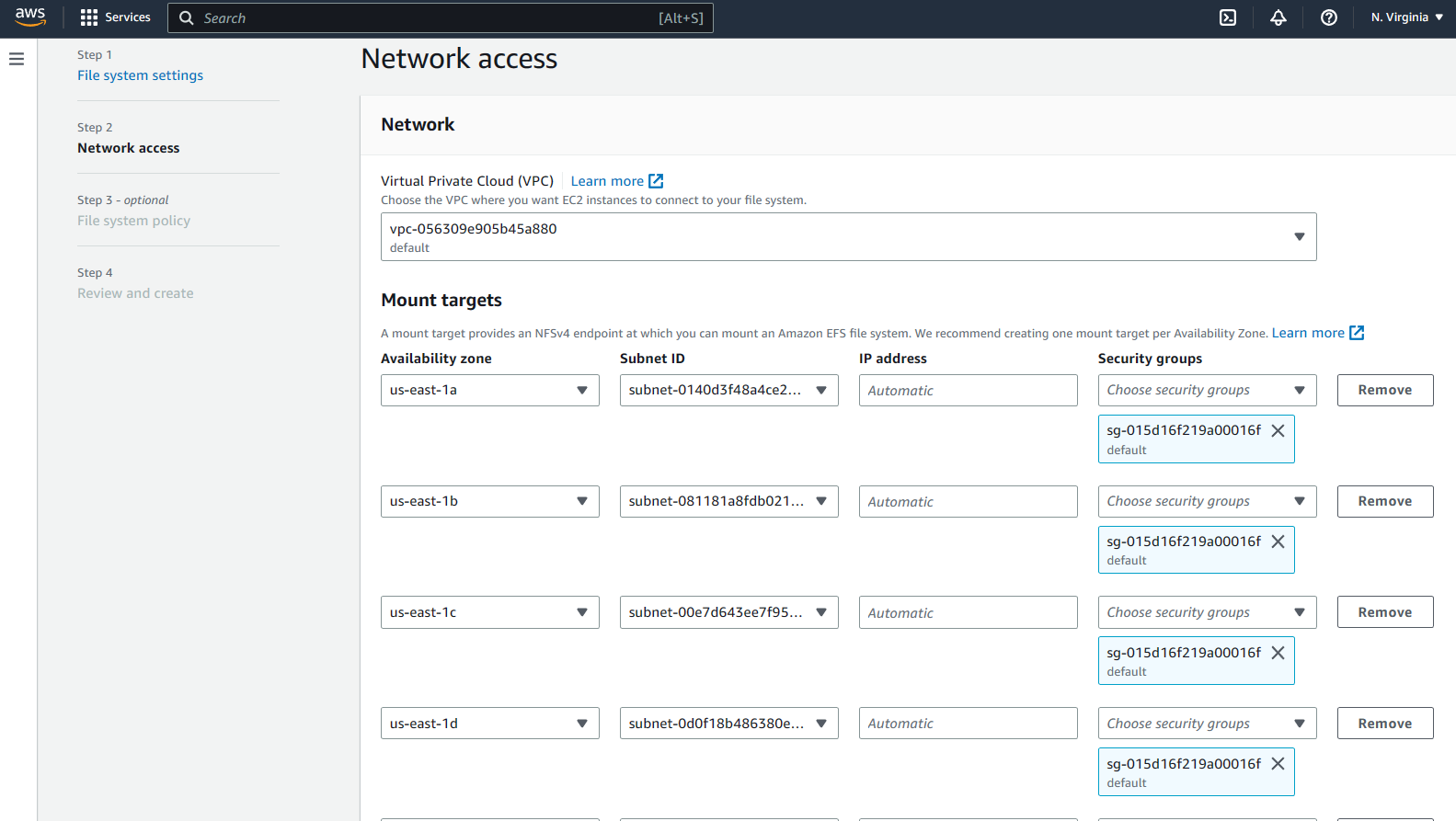
Set the “transition IA” to none

Uncheck “enable encryption of data at rest” check box.

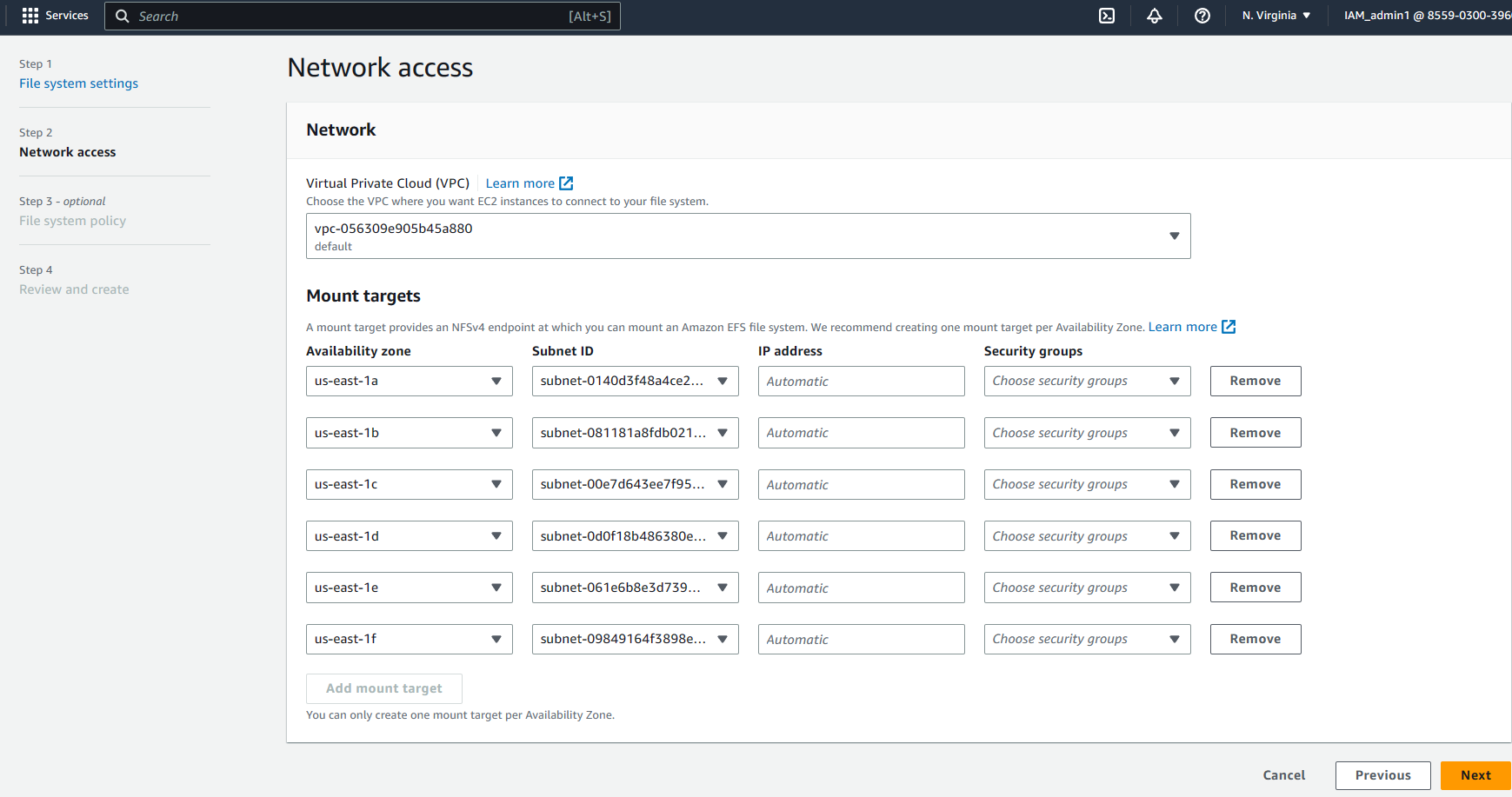
Select **throughput mode** to Bursting and click on **next** button



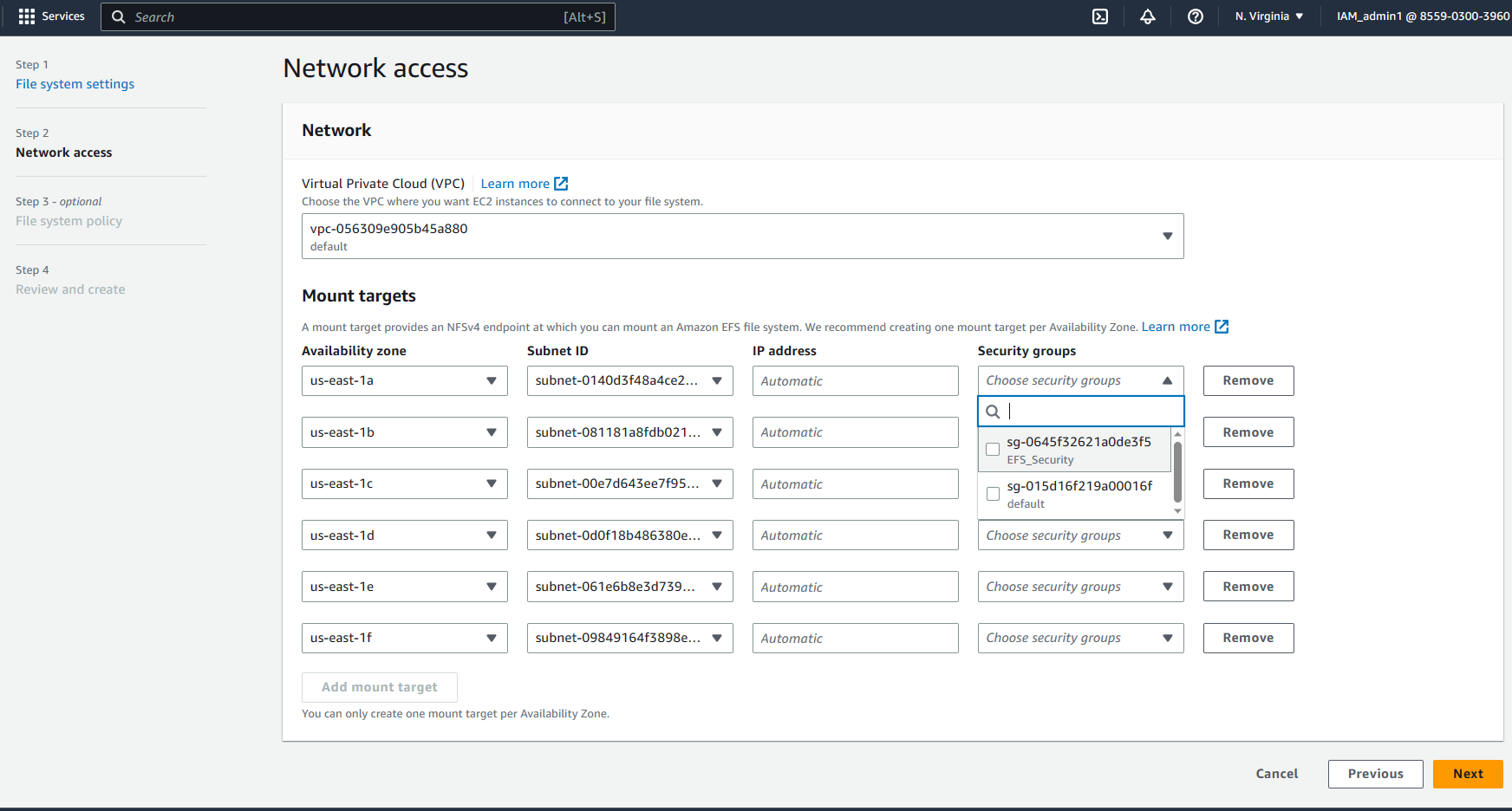
A network access page would appear



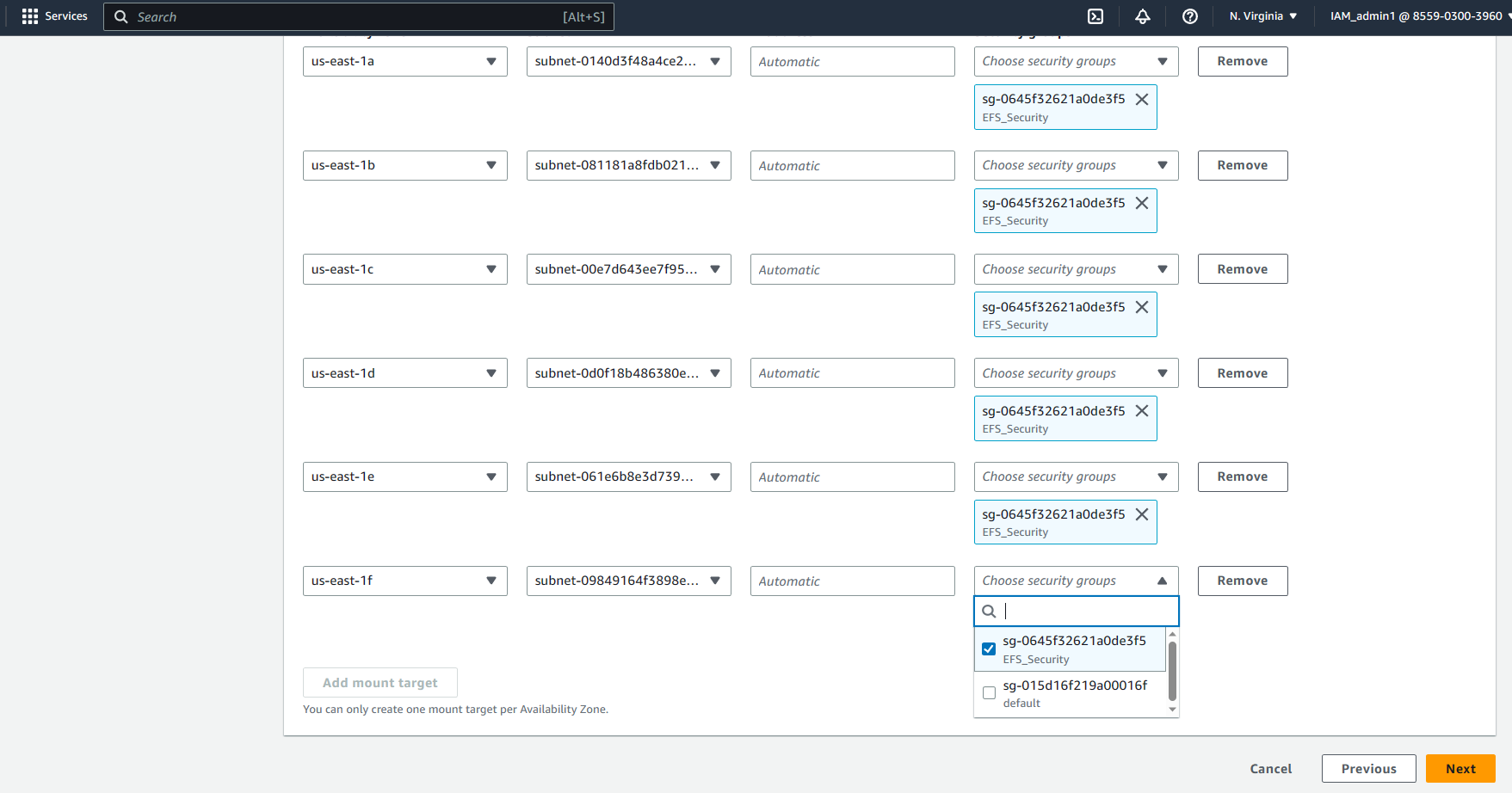
Clear the security group of the Availability zones by click on the “X” on the security group for for each Availability zone.



Select the security group which you had created earlier in the security group drop down

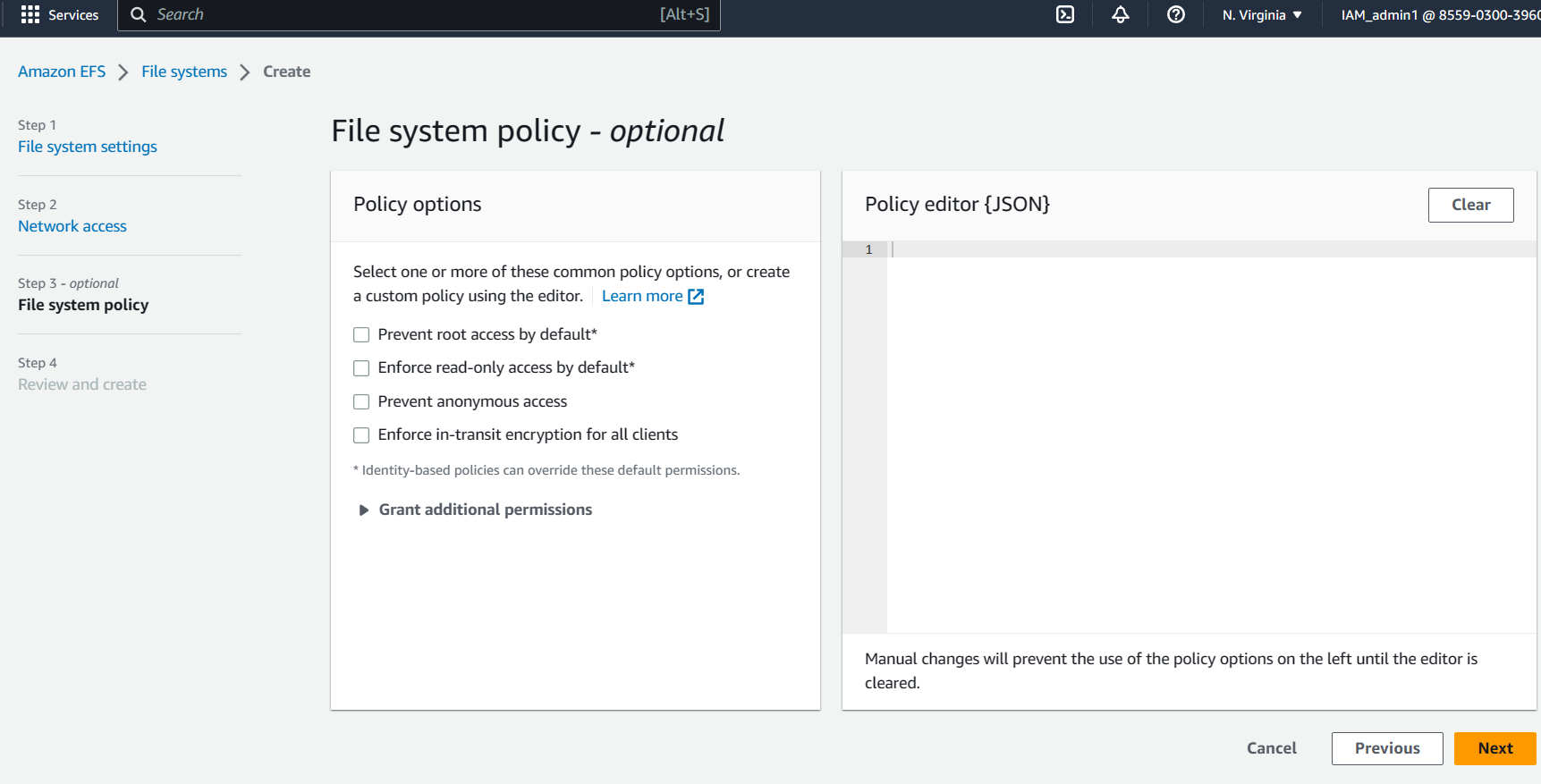


Select your security group for all the availability zone



Click on next

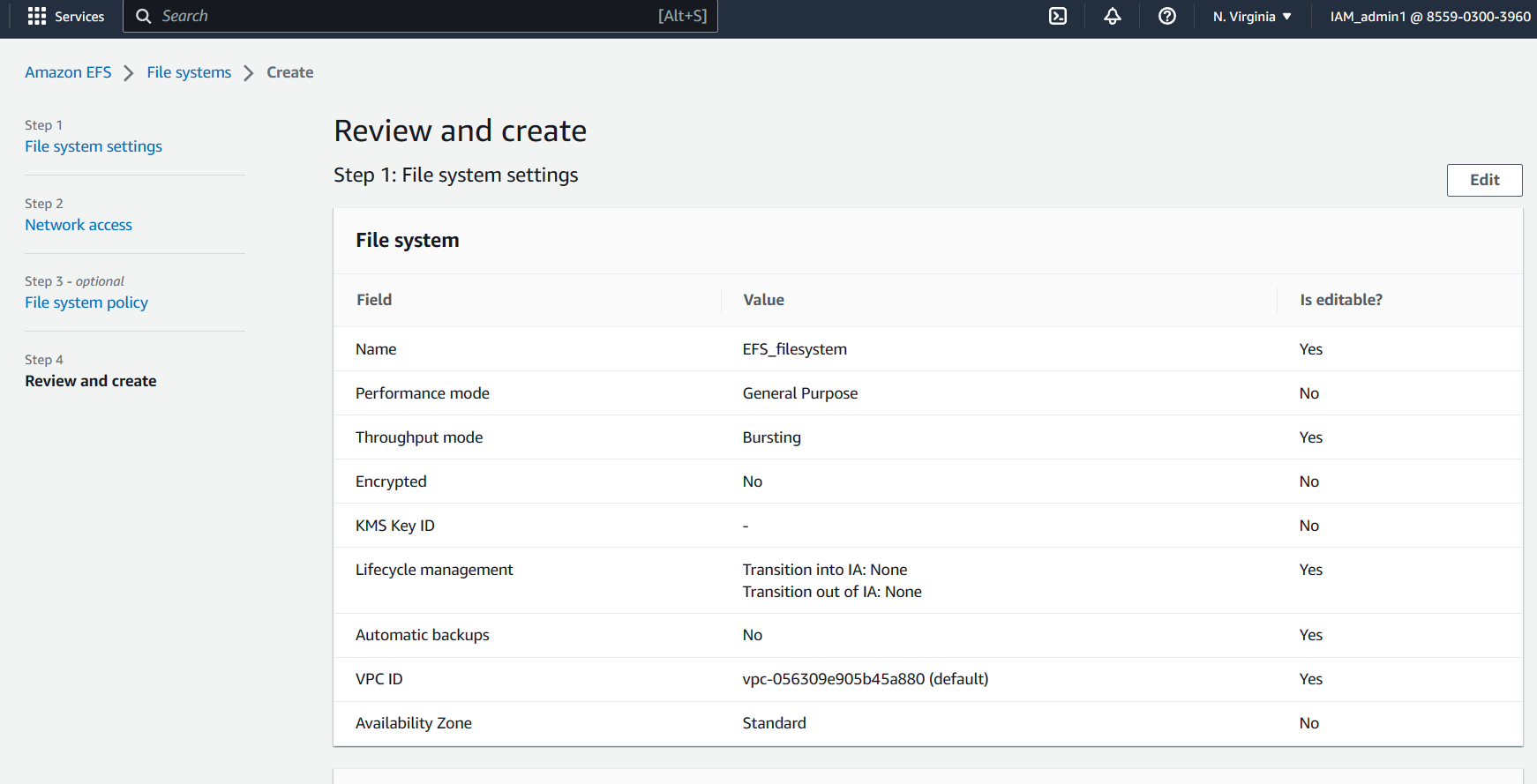
A file system policy page would appear.



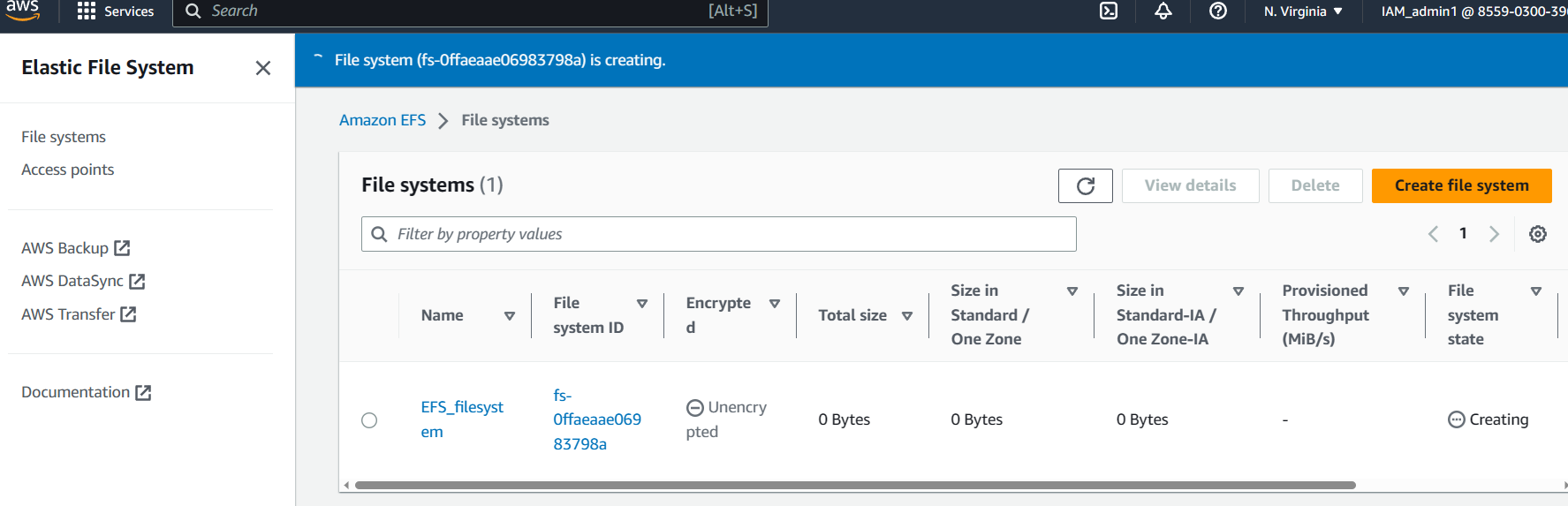
Click on next

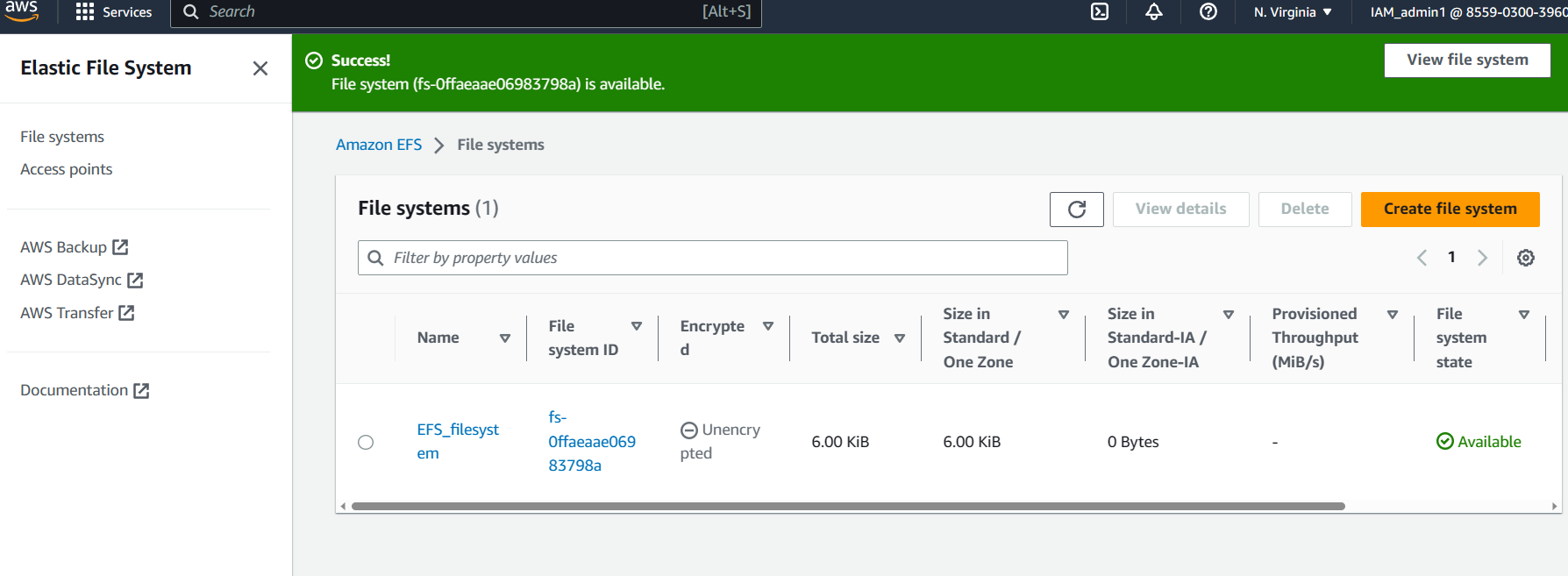
Review and create page would appear.

Scroll down and click on **create**

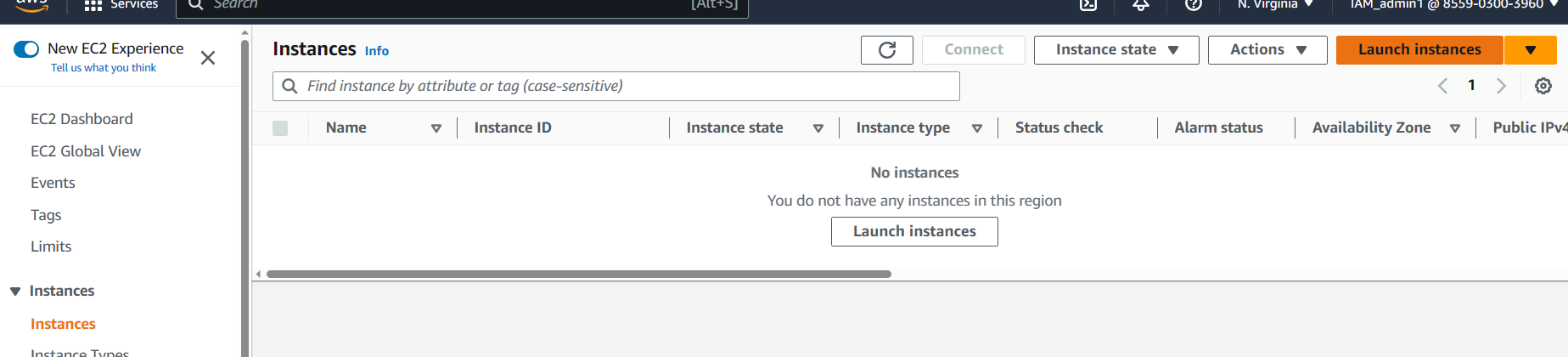


Congratulations you have successfully created a file system



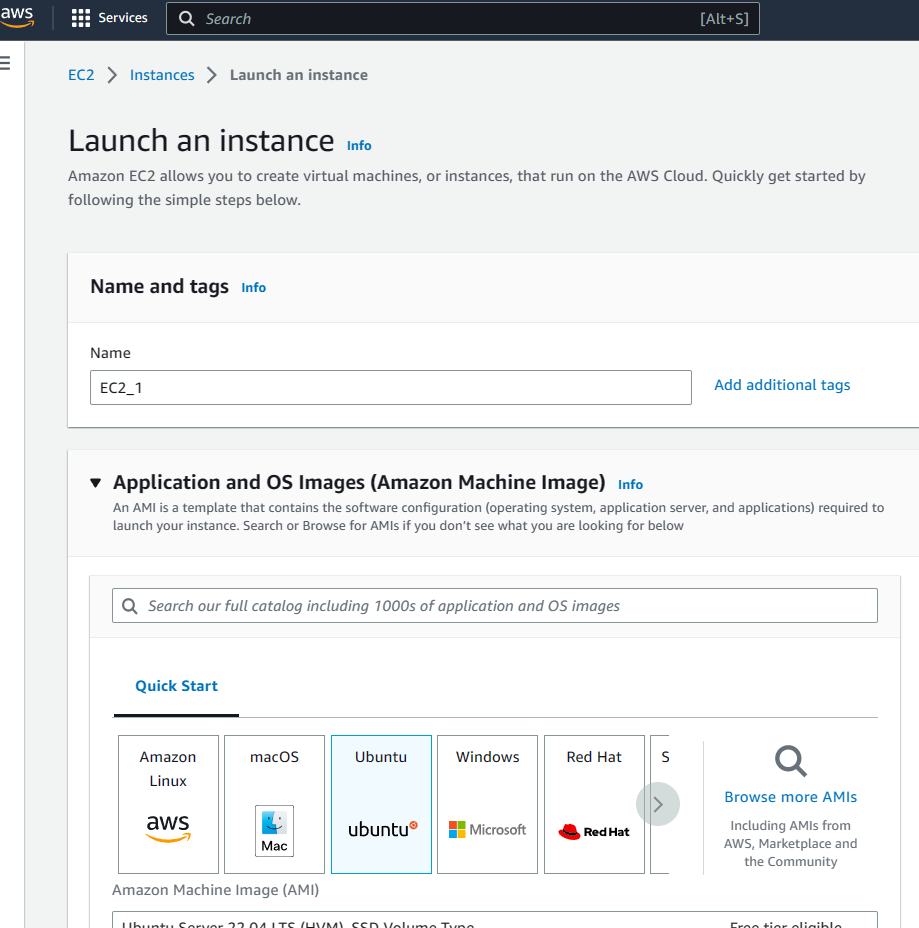


Switch to the tab in which the Instance menu was loaded and click on **Launch instance**

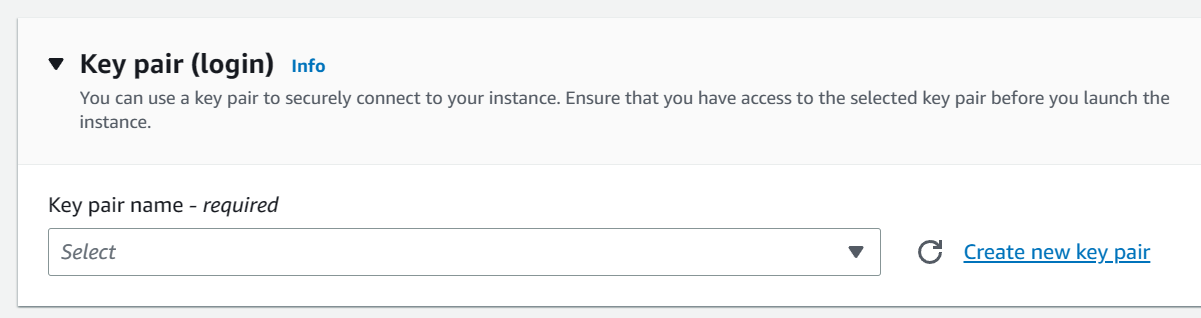


A launch instance page would appear

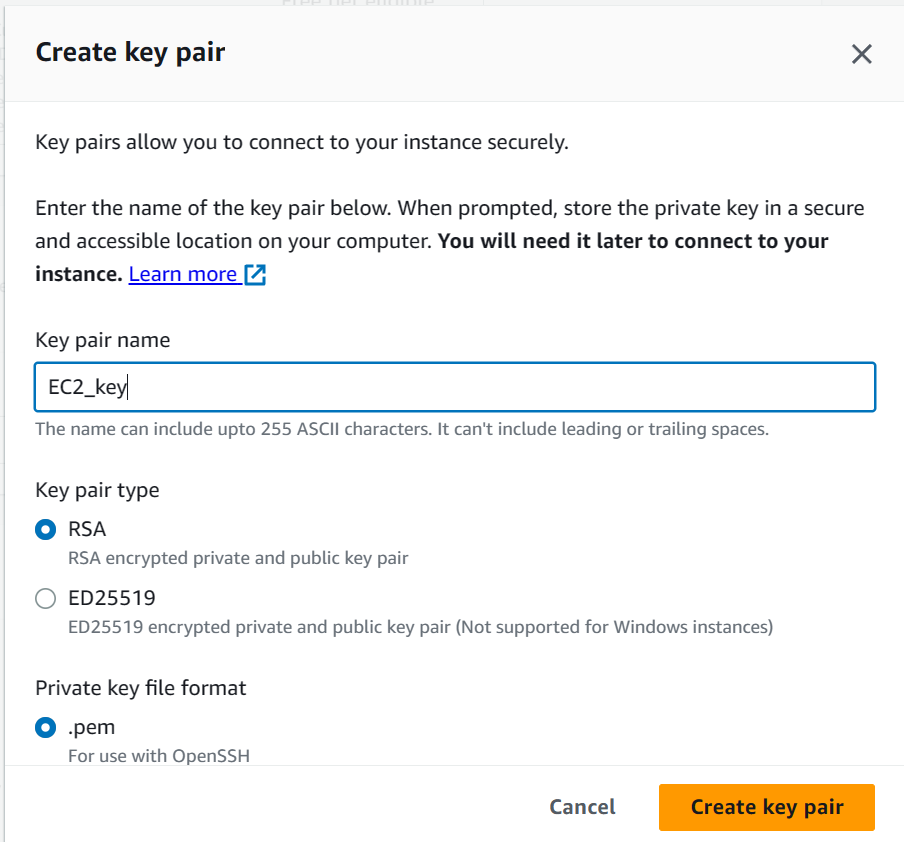
Fill in a desired name in the name field and select ubuntu in the quick start section



Scroll down to **Key pair** section and click on **Create new key pair**

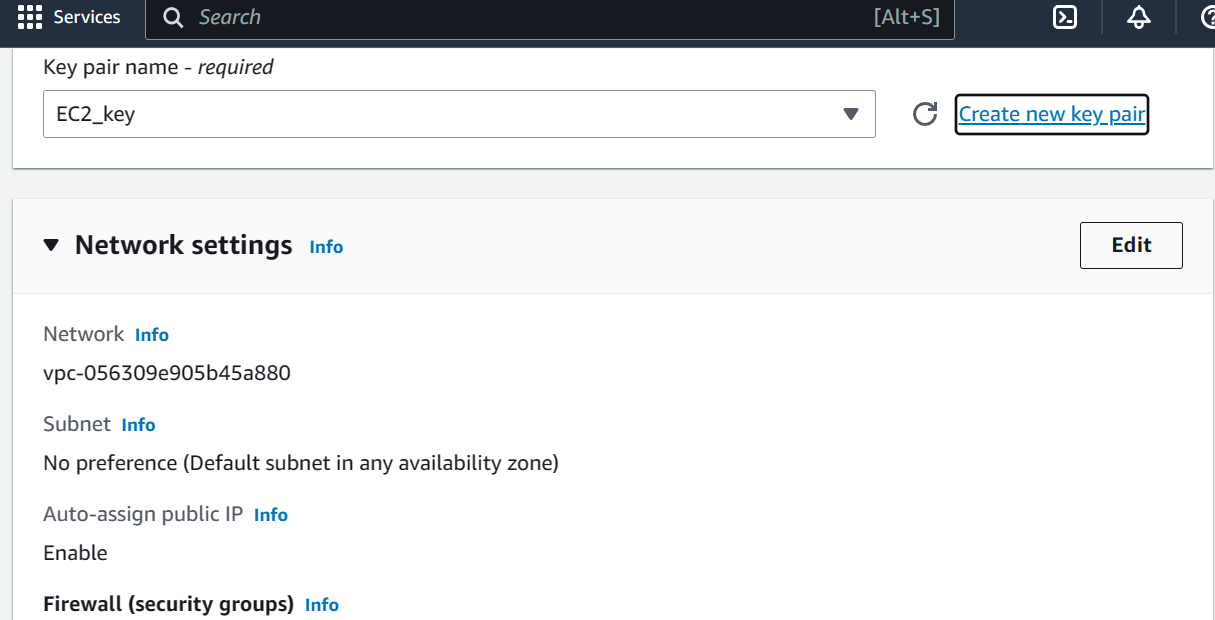


Specify a name in the name field and click on **create key pair**

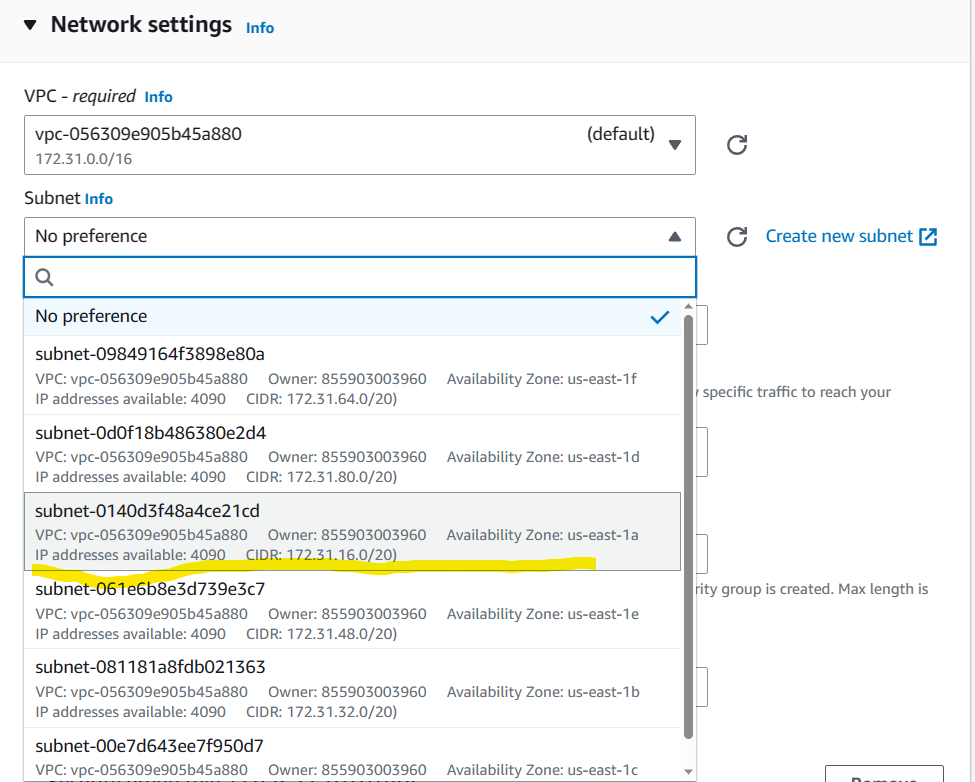


After click on create key pair a pem file would be downloaded place the pem file in a safe location

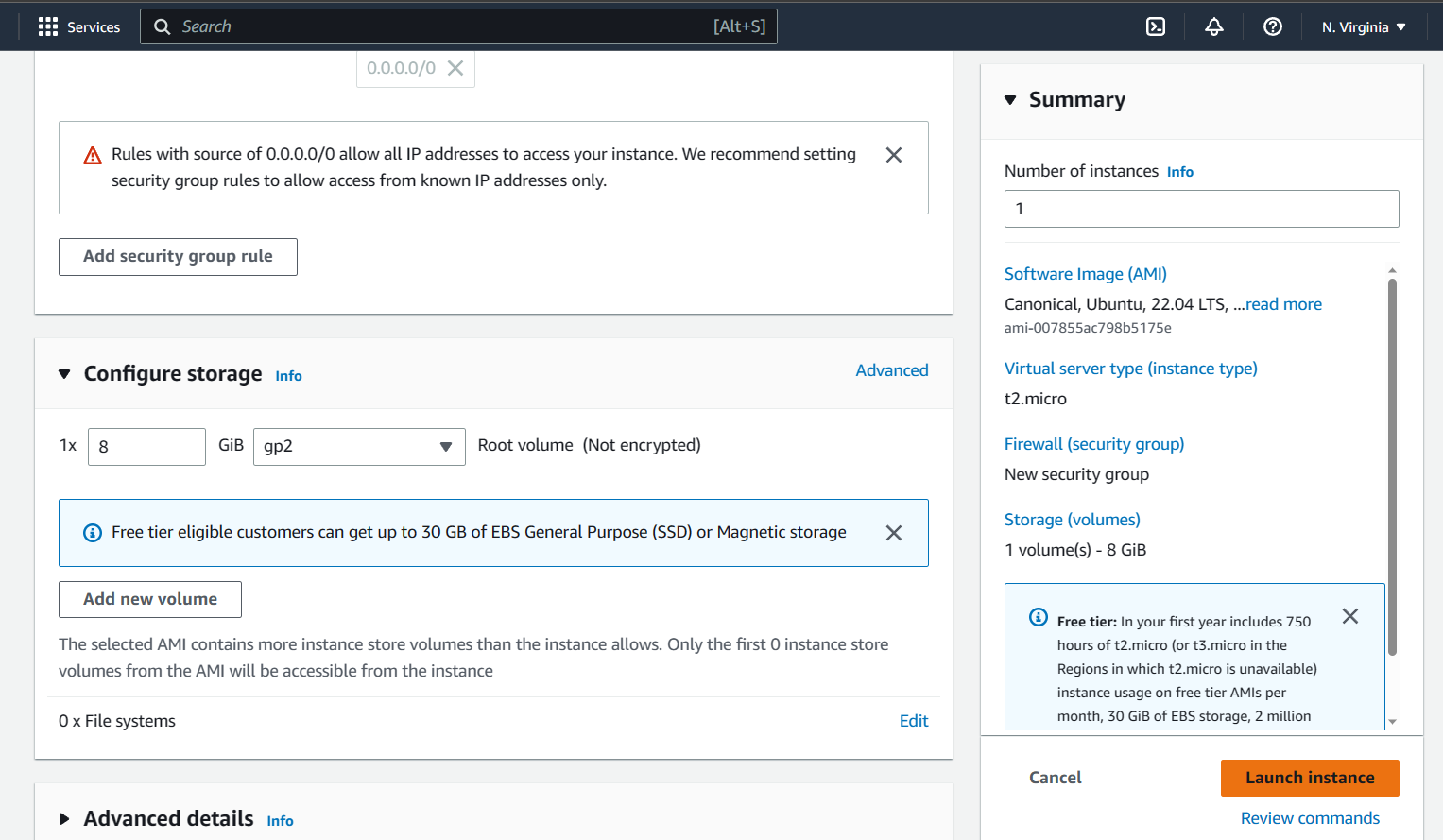
Scroll to network settings and click on edit

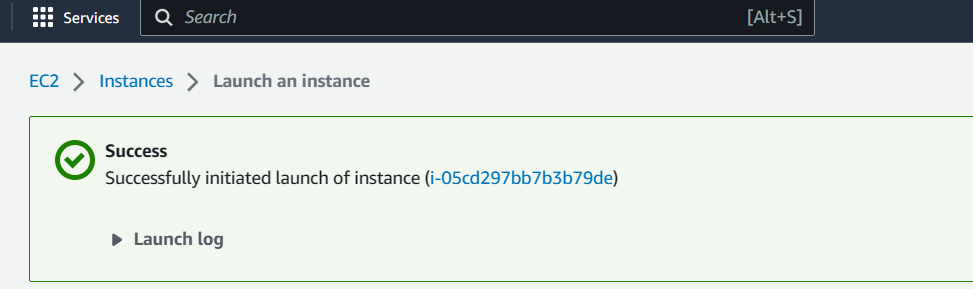


Under subnet preference select us-east-1a as shown below



Scroll down and click on **Launch instance**

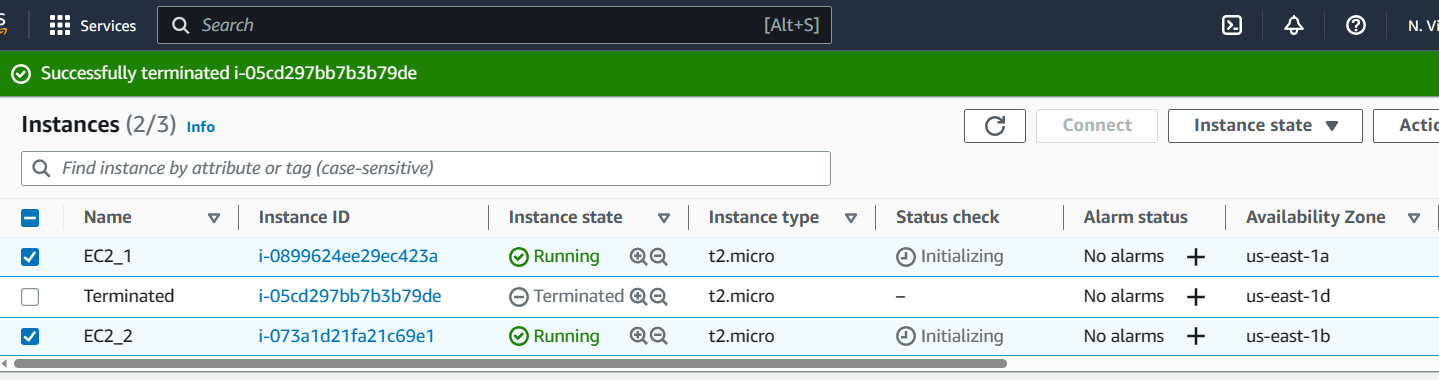


Congratulations you have successfully created an instance

Follow the same steps to create another instance with different Availability zone by selecting the subnet preference as us-east-1b

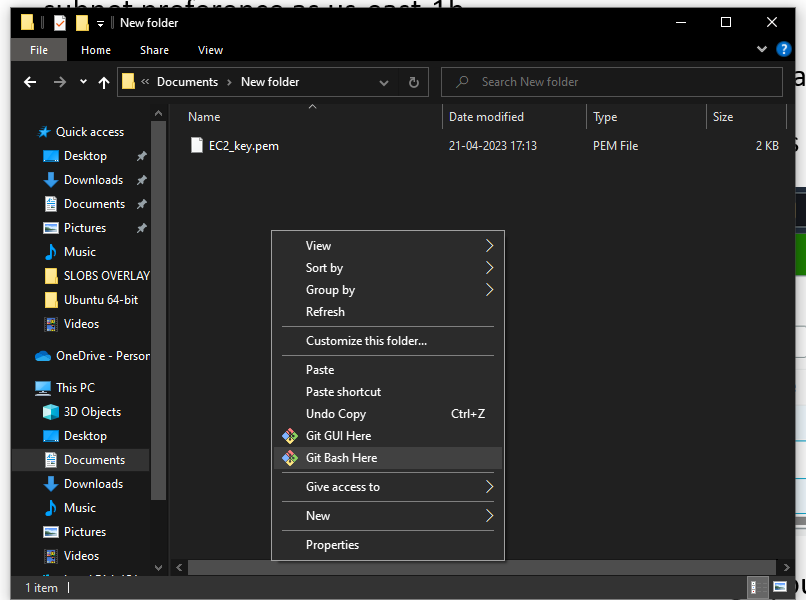
Make sure to select the new created key pair in the key pair section

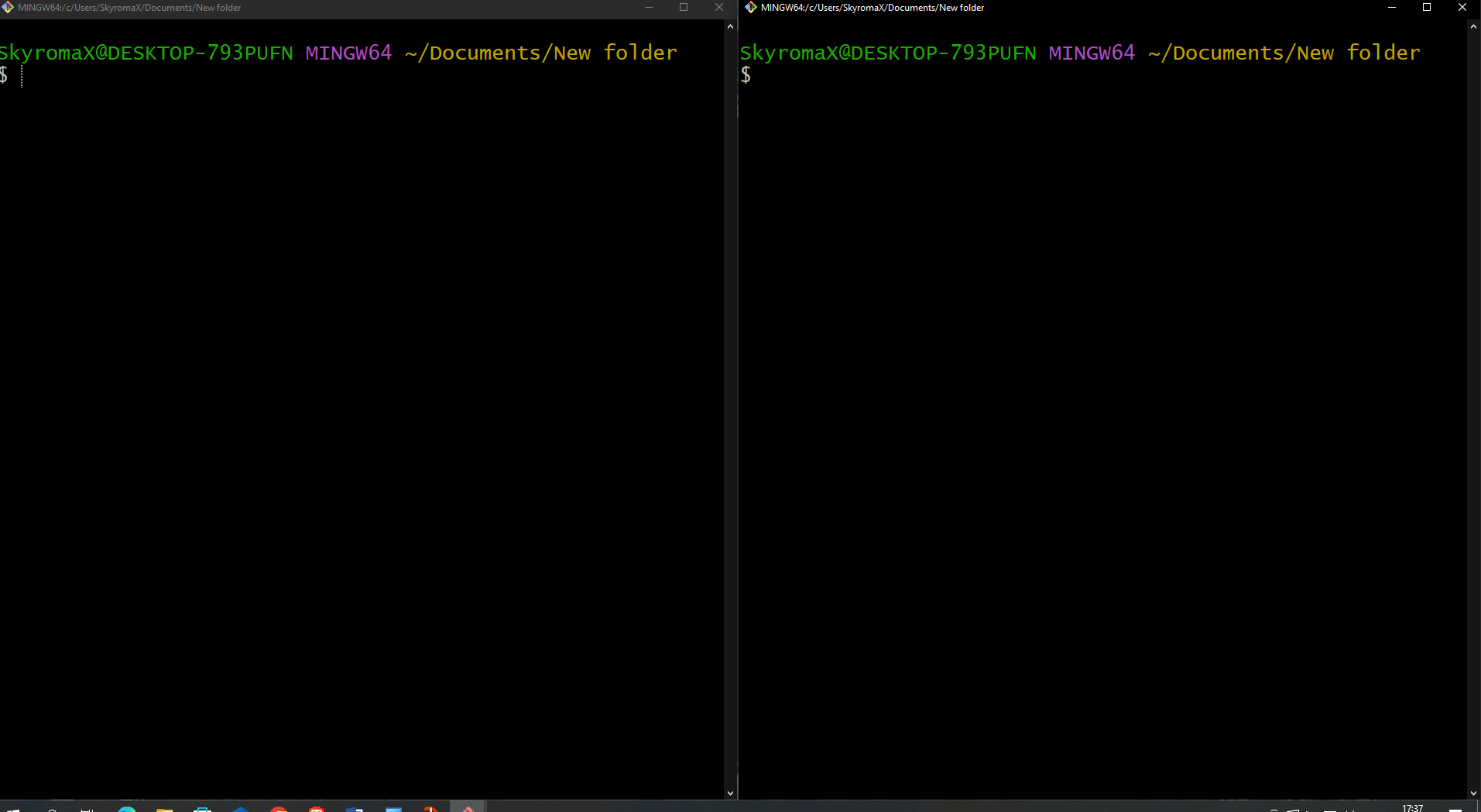
Congratulations you have successfully created 2 instances in two different availability zone



Now Establish a connection to those instances through your desktop/laptop computer through git bash terminal / linux terminal

Open git bash terminal in the folder/location where the pem file is placed



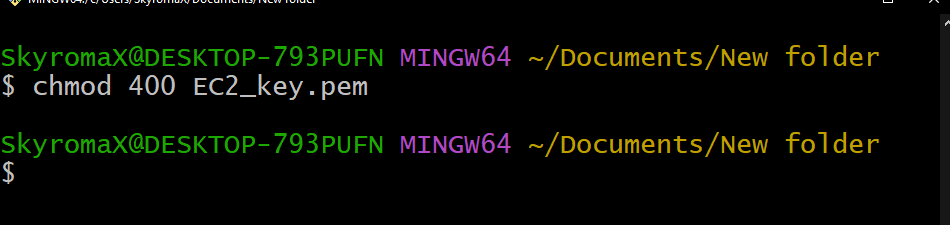
Open 2 git bash terminals as you have created two instances

In the 1st git terminal run the following command

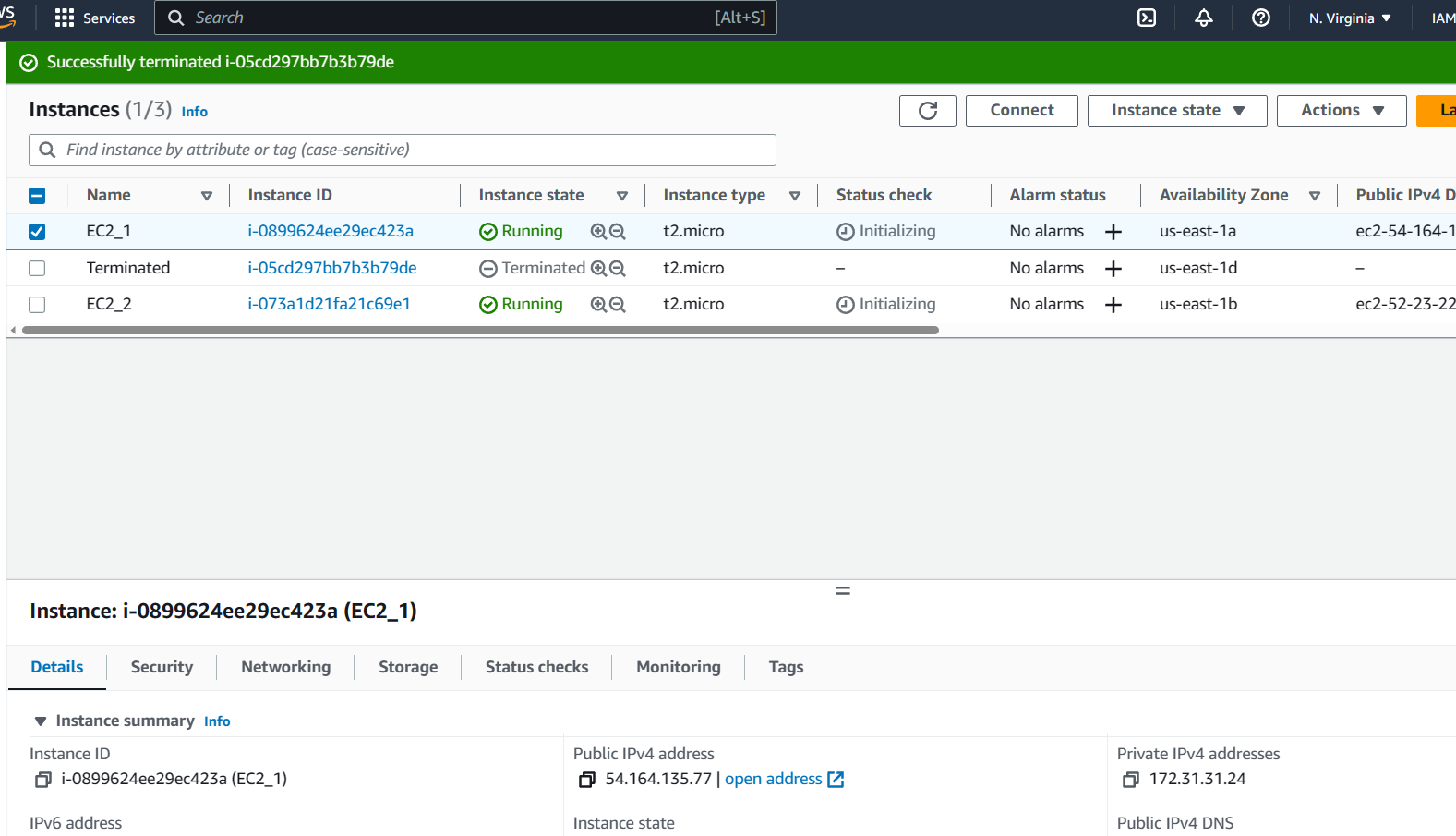
>chmod 400 [file.pem]  
#Replace the “file.pem” with the appropriate file name. in the current scenario it is EC2\_key.pem

>chmod 400 EC2\_key.pem

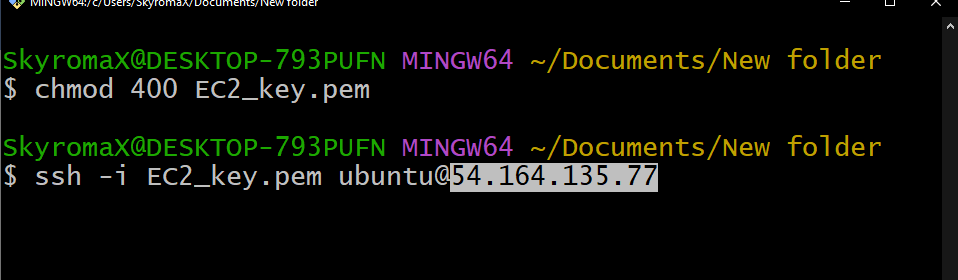
Click enter

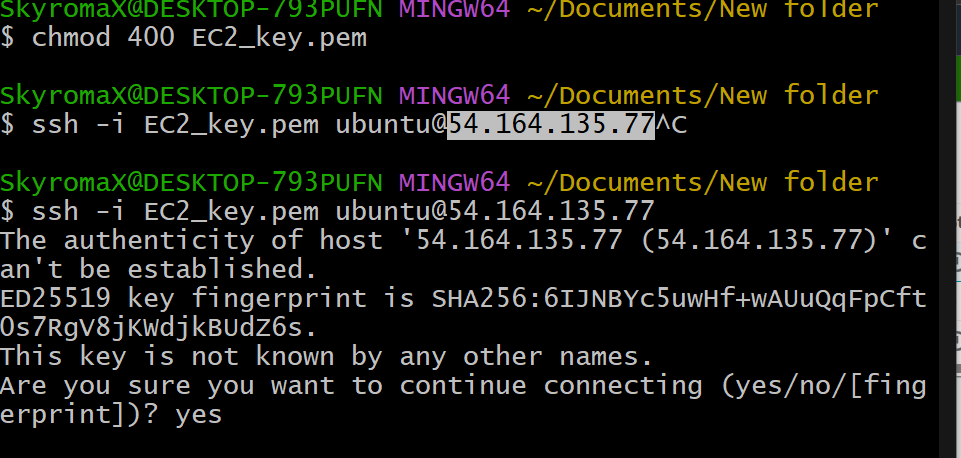


In order to connect to the 1st instance, visit the instance details and copy the public address in the

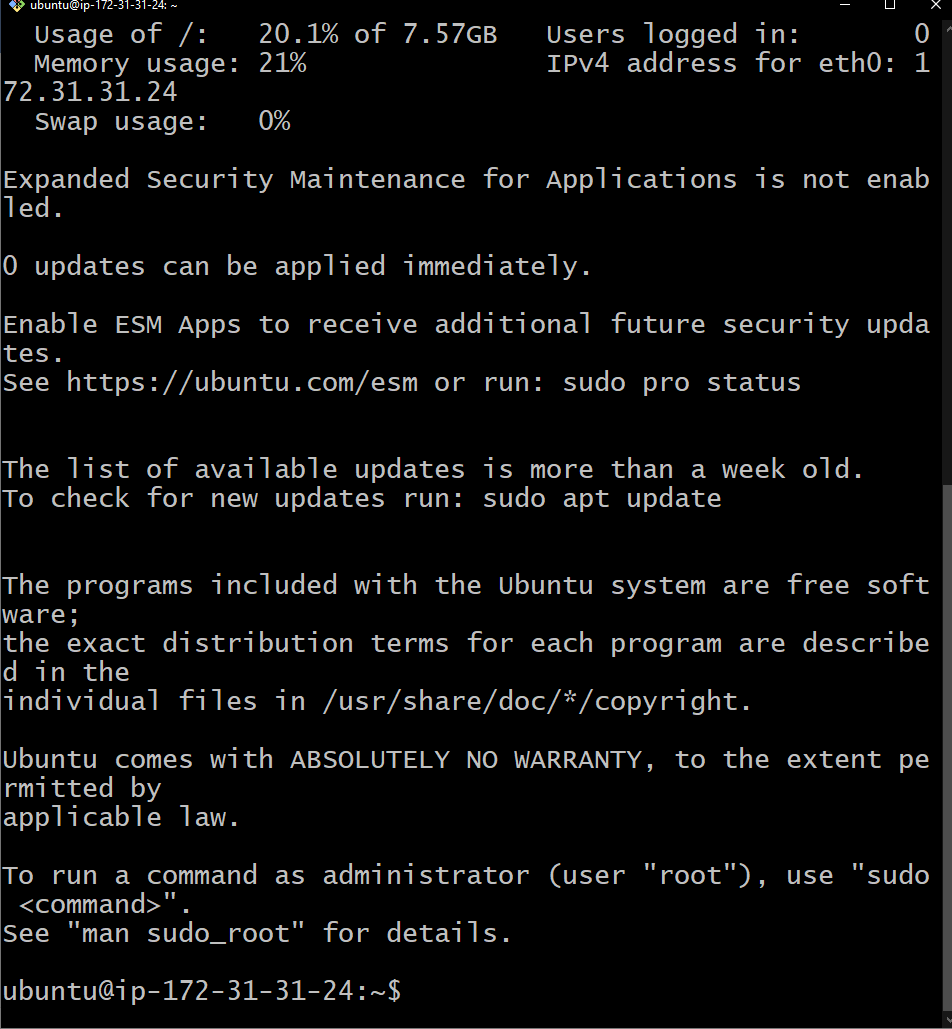


After coping the public address run the following command in the git terminal  
>ssh -I [file.pem] ubuntu@public address

In the current scenario the code would look like this

Type “yes” when the below prompt appears and click on enter

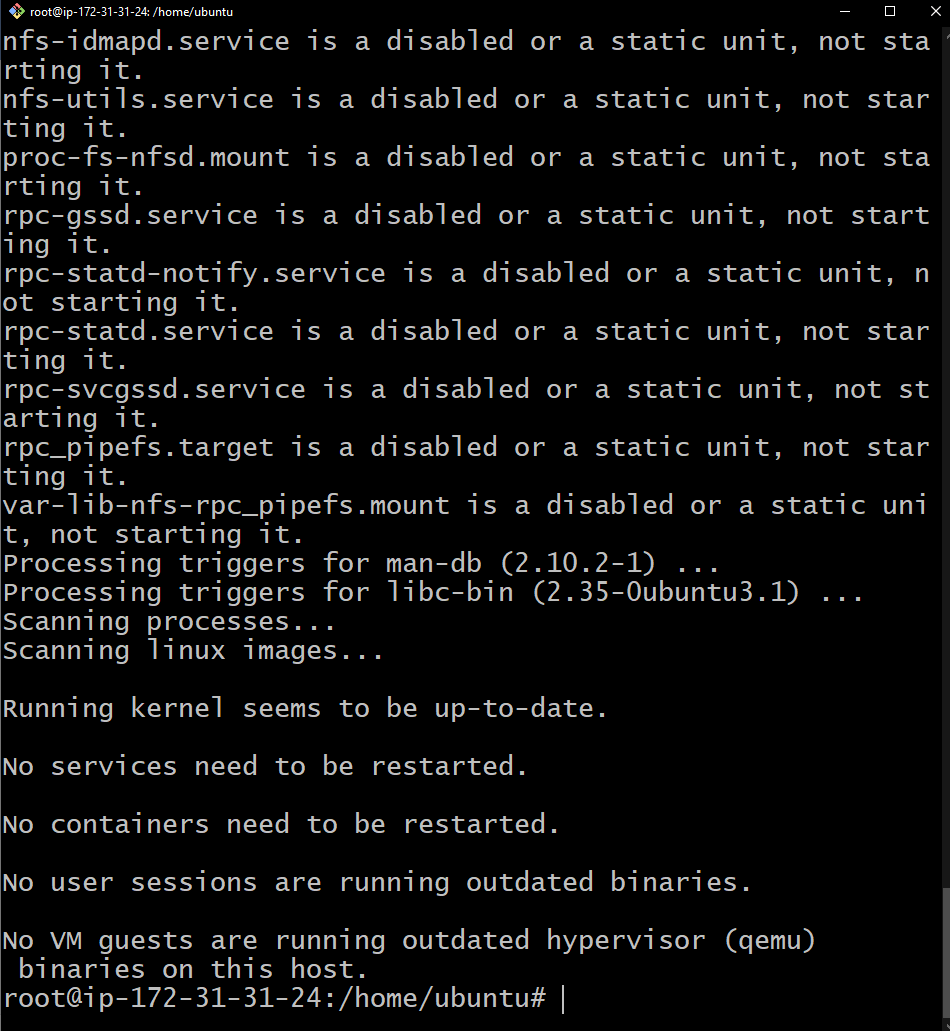
Congratulations you have successfully connected to the 1st instance



Now run the following mandatory command

>sudo apt-get update  
>sudo su  
>sudo apt-get install nfs-common

Click on y when a “(y/n)” prompt appears and click on enter

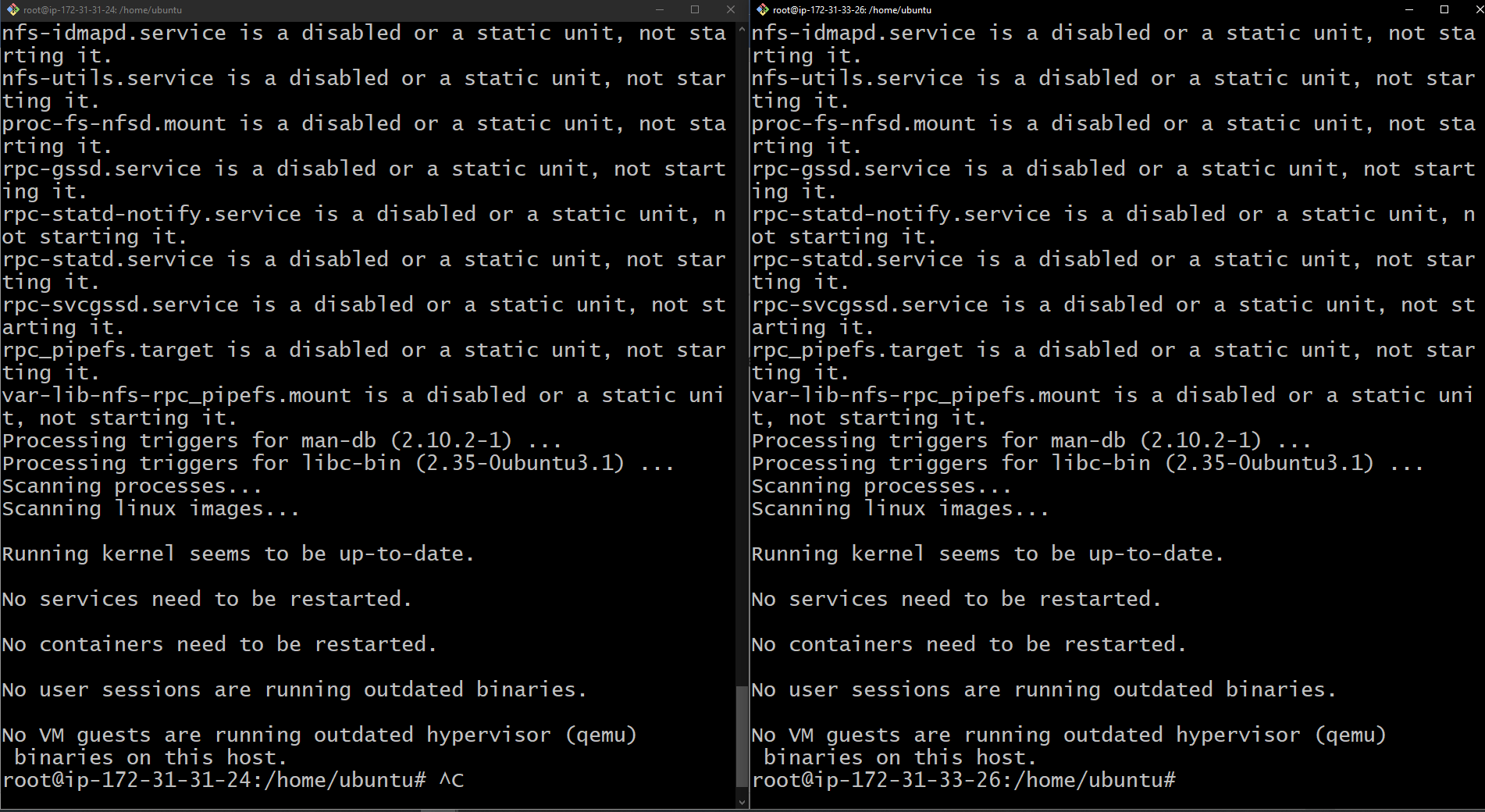


Great your 1st instance is ready to mount.

Move to the second terminal and connect to the second instance using the second instance public address and the pem file  
after connecting to the second instance through the 2nd terminal run the following commands similar to the 1st terminal

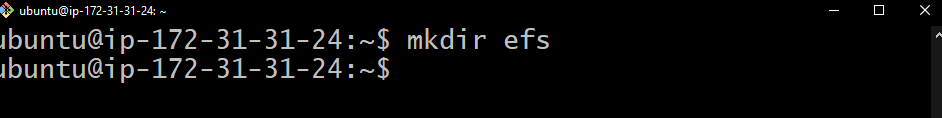
>sudo apt-get update  
>sudo su  
>sudo apt-get install nfs-common

Click on y when a “(y/n)” prompt appears and click on enter

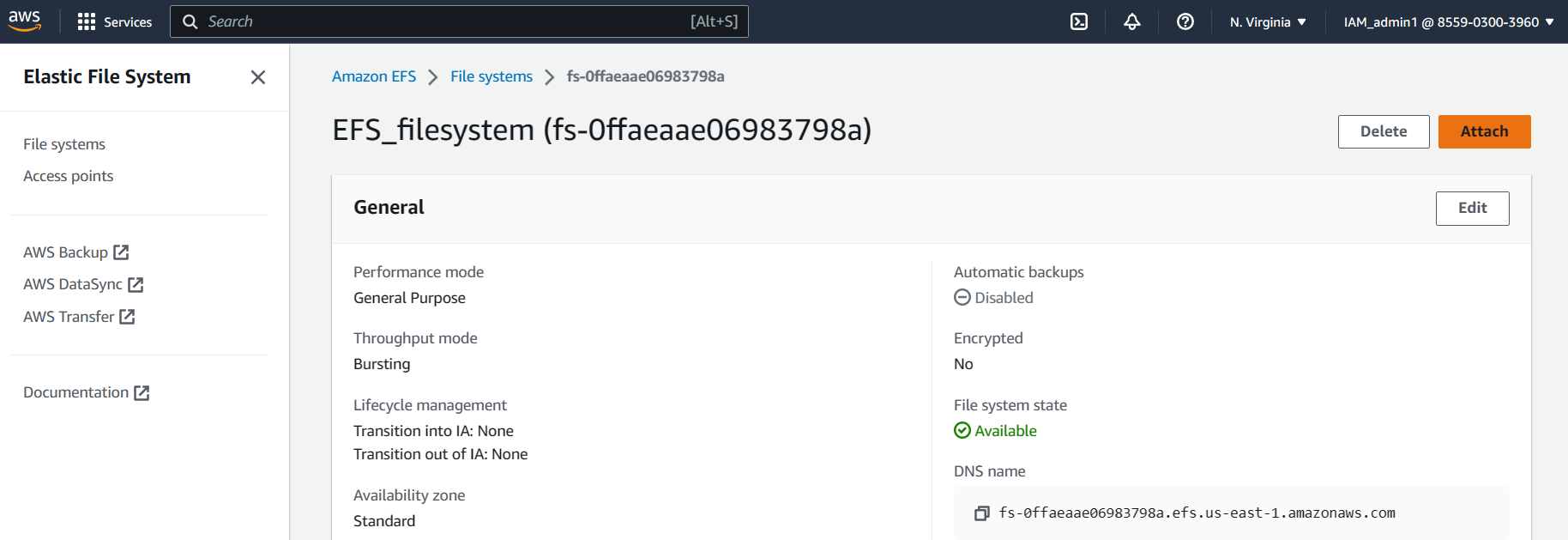


Both your terminals are uptodate and ready to mount the EFS

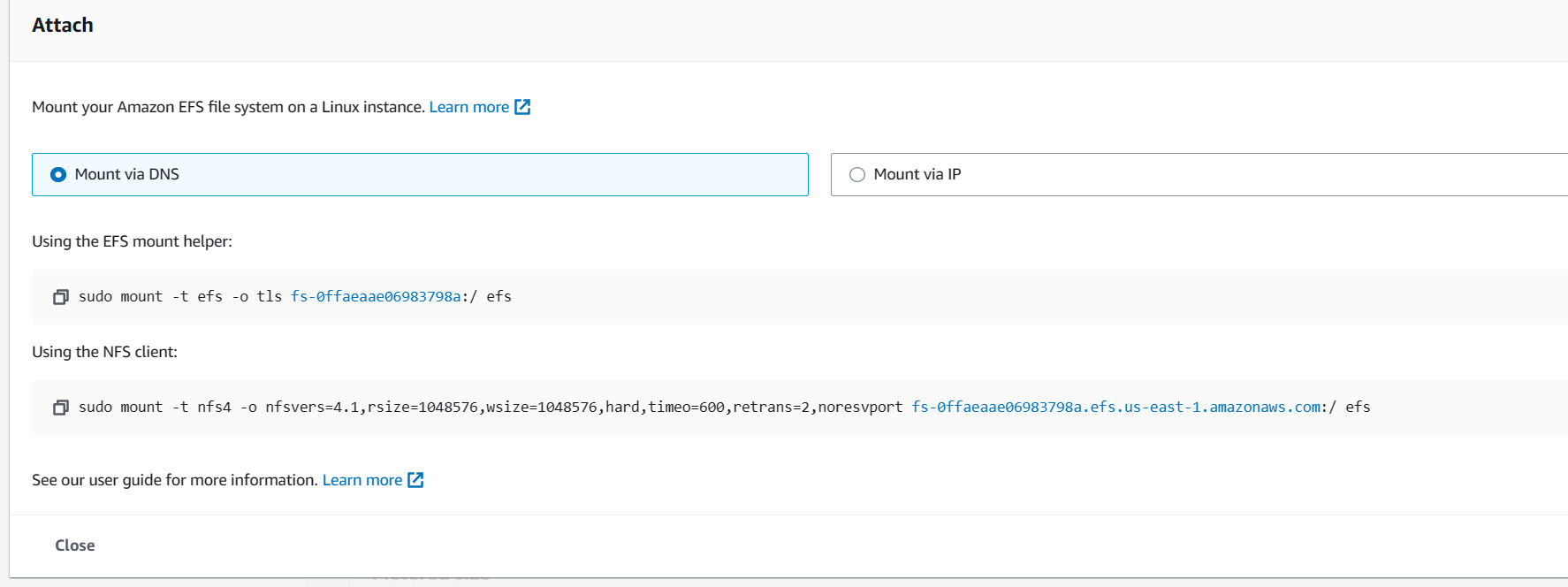
Switch to the 1st terminal and create a folder named “efs” by entering the following command

>mkdir efs  


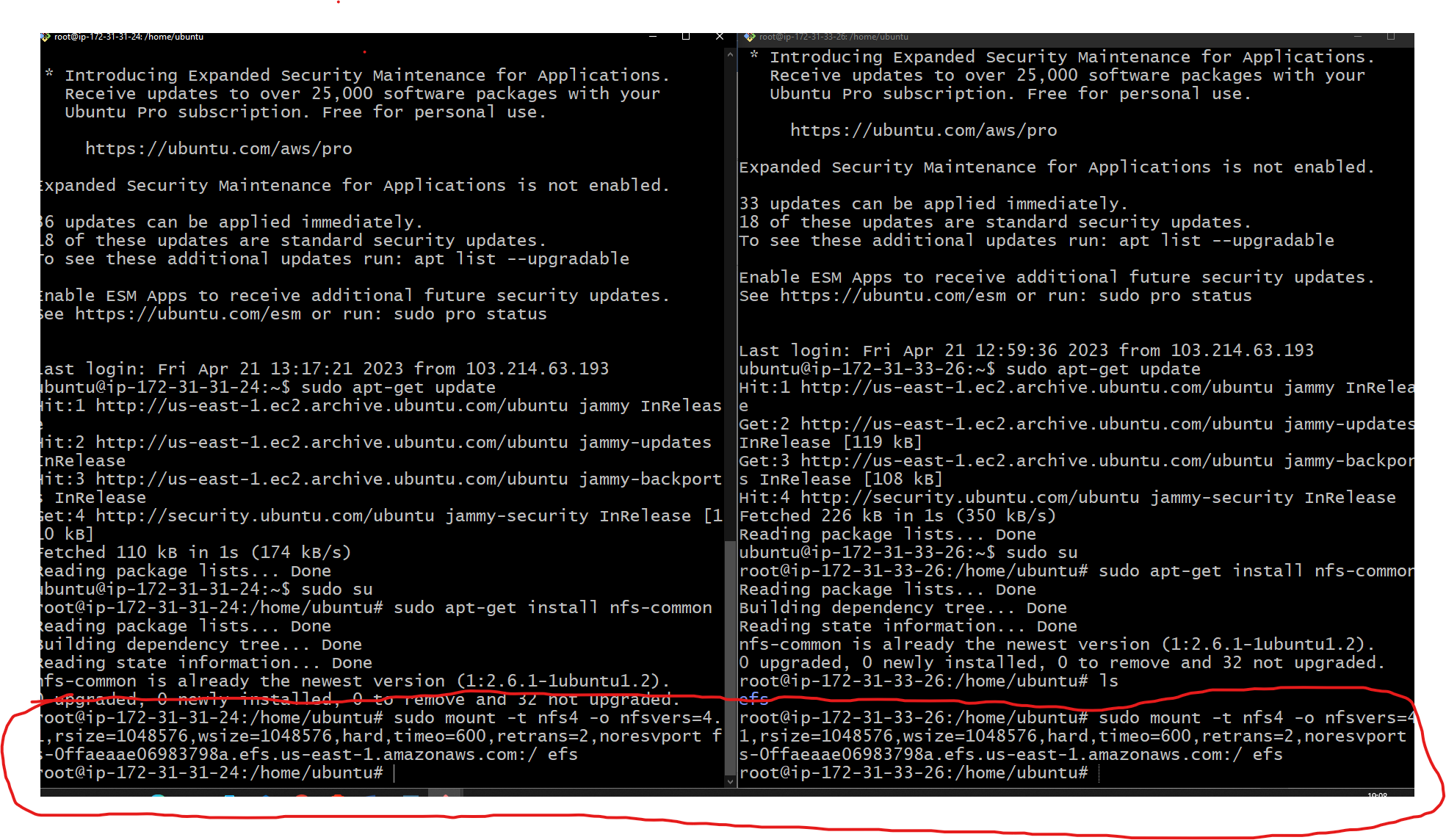
To mount the EFS on the instance visit the EFS browser tab, select the EFS unit and click on mount



A pop up with mount command would appear as shown below



Copy the “using the NFS Client” command, paste in the 1st terminal and click on enter.



Run the same set of mount commands in the second terminal as well

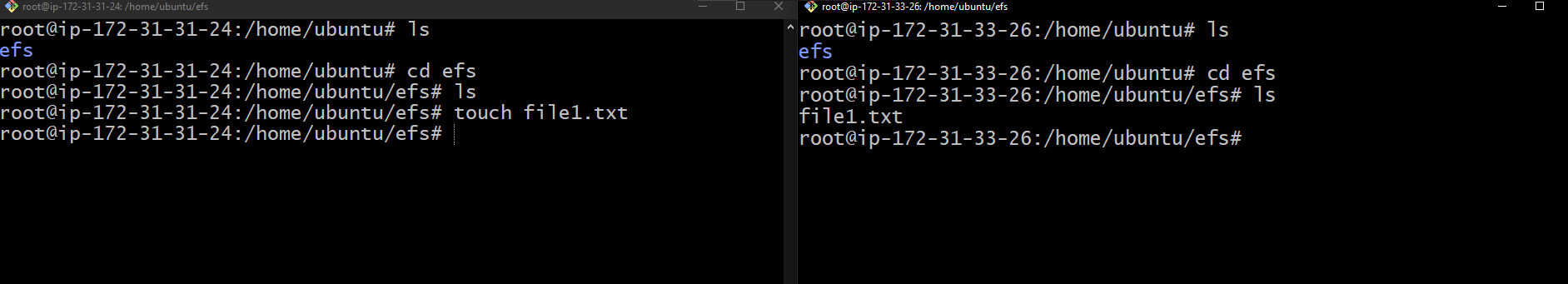
To check whether the mount is functional try creating a file in 1st terminal and checking whether it is reflecting in the second terminal

In the 1st terminal open the mounted folder (in this scenario is efs folder)

>cd efs  
>ls  
>touch file1.txt

Switch to the 2nd terminal and open the mounted folder  
>cd efs  
>ls

By following the above steps the new file created in the 1st terminal will reflect ias an output in the 2nd terminla’s ls command as shown below

  
Left terminal = 1st terminal right = 2nd terminal

Similarly create a file2.txt in the 2nd terminal and check in the ls output in the 1st terminal

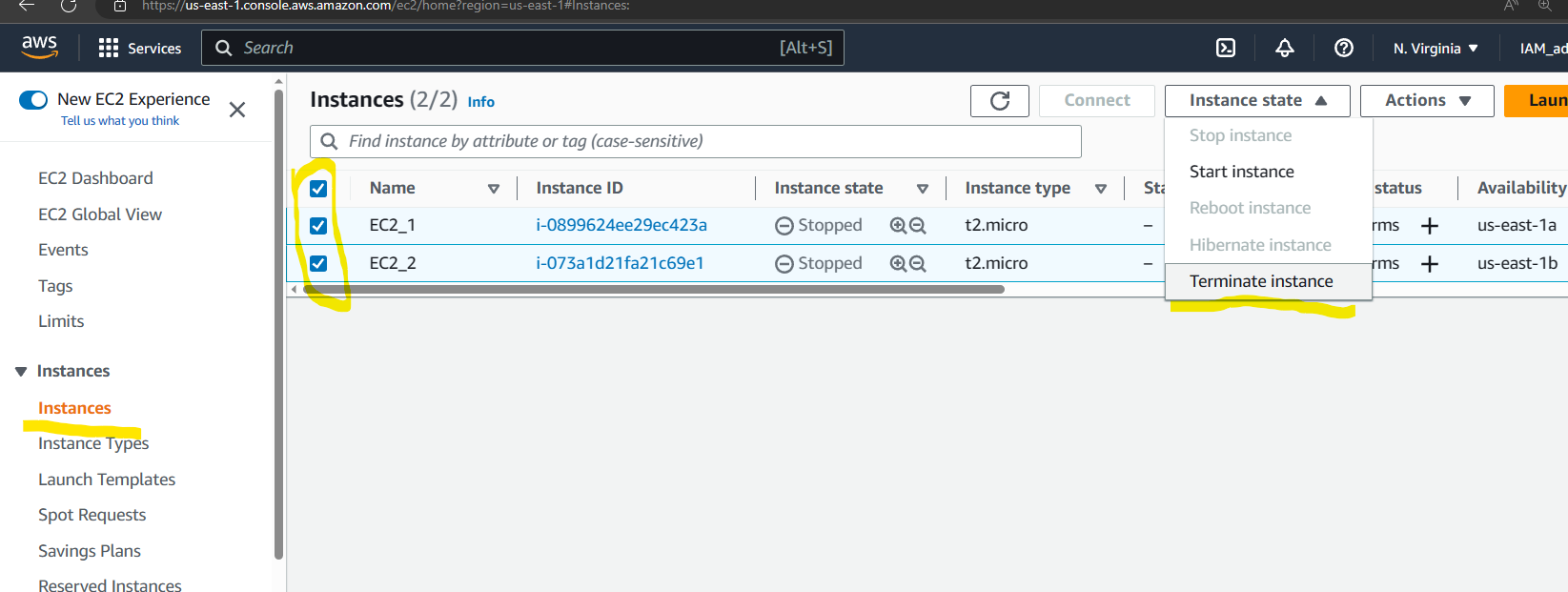


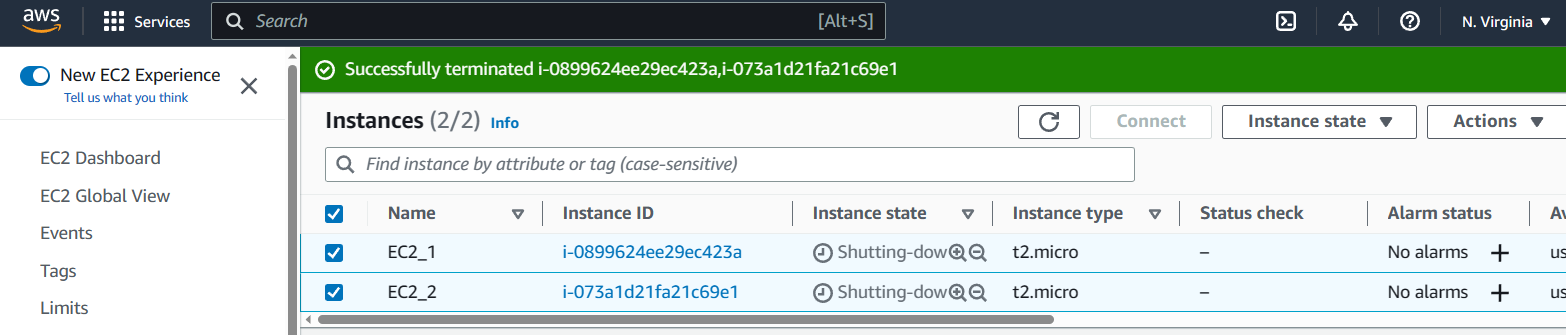
Congratulations you have successfully created a file system that sync files between 2 instances.

Cleanup

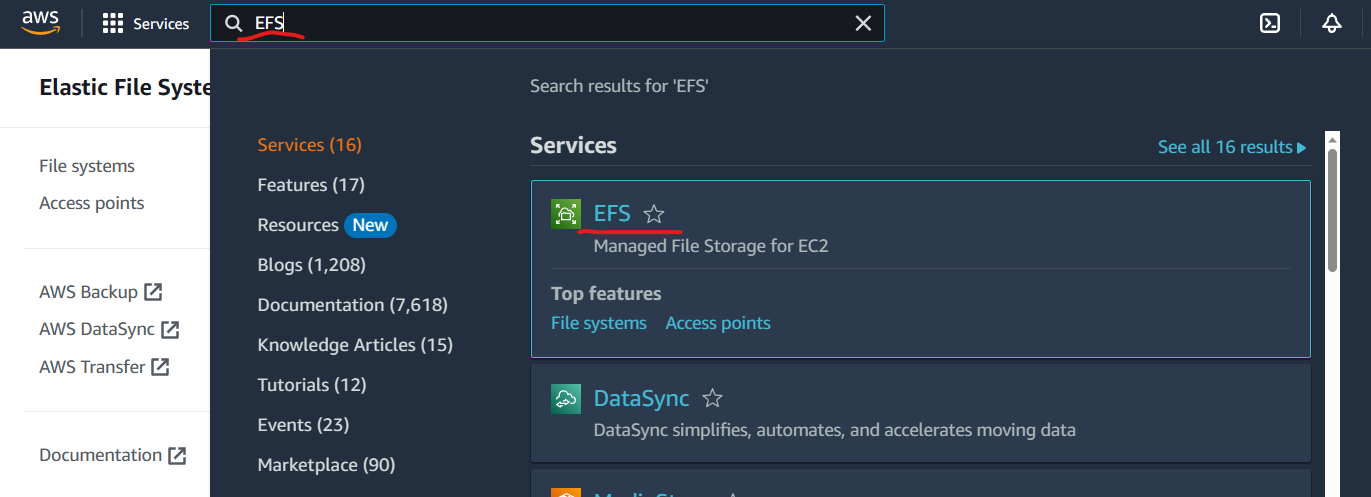
To perform the clean up make sure to delete/dregister all the services that ve been used in the above practice.

Visit the **Instance** menu and terminate the 2 instances as shown below

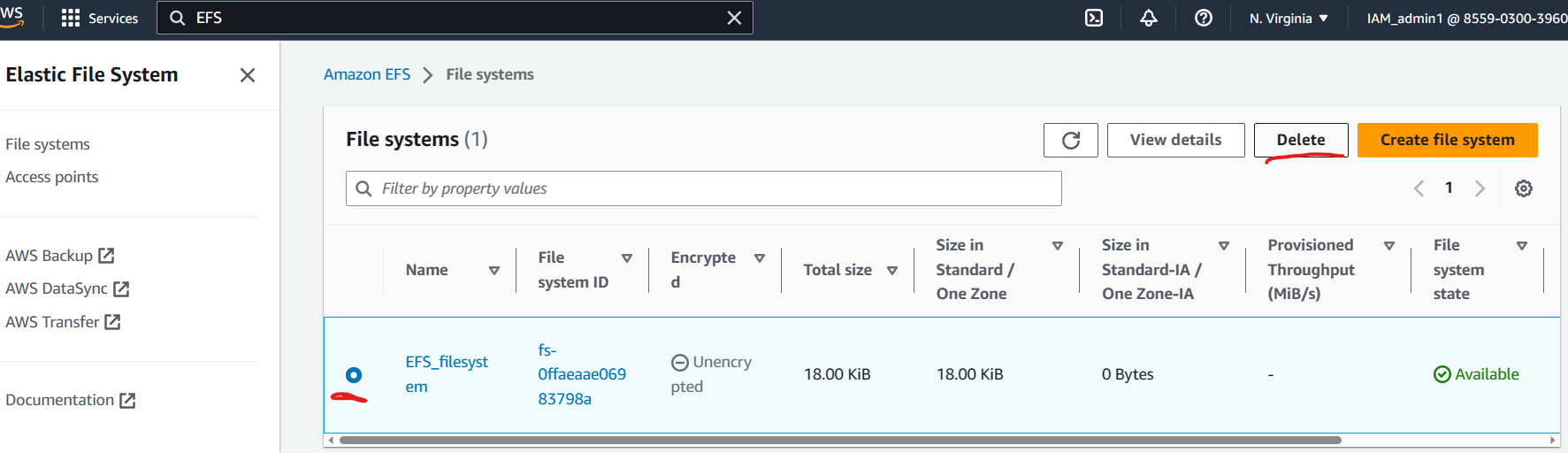




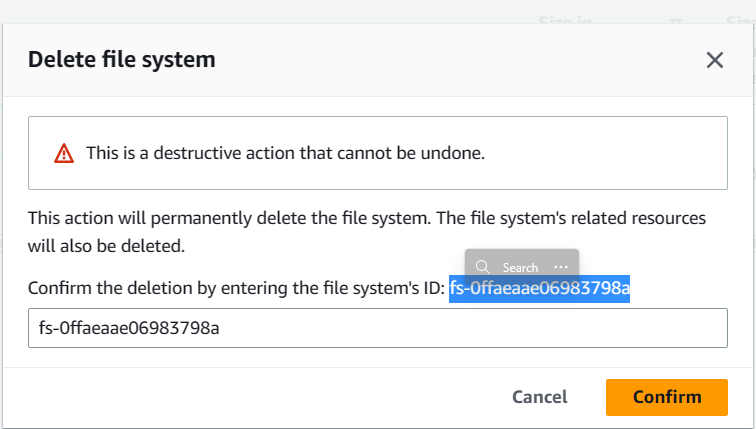
Visit the EFS services menu



Select the EFS which was used in the above practice and click on delete as shown below

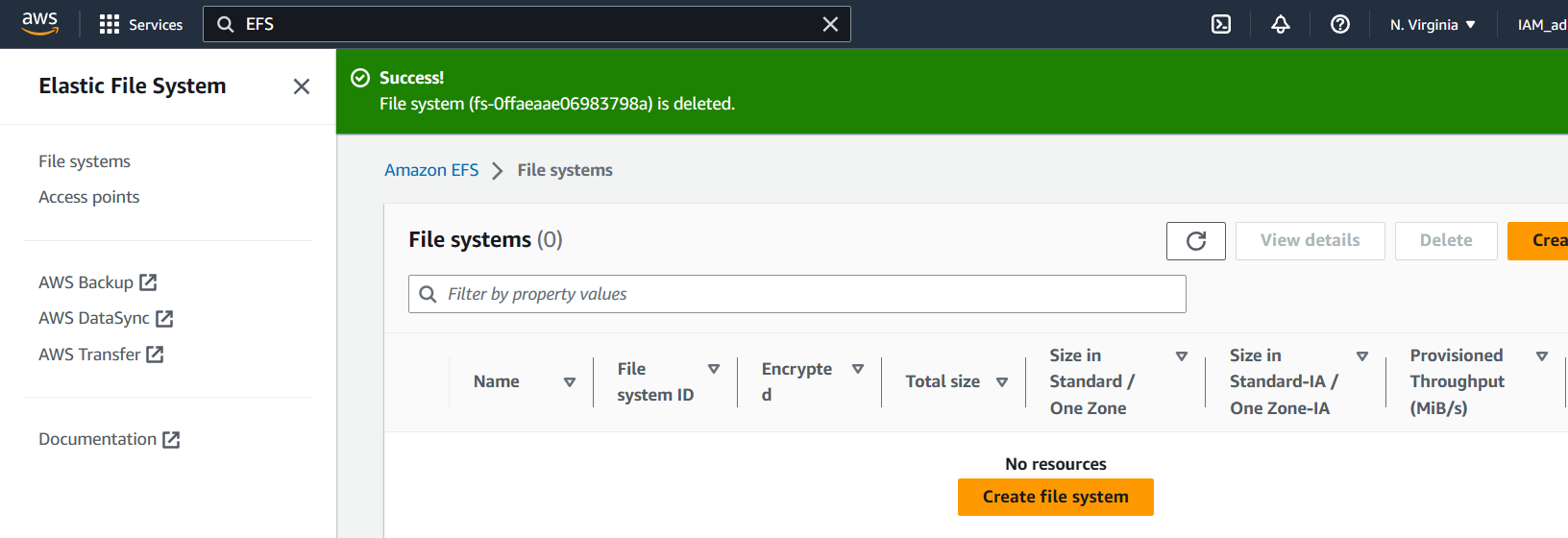


Copy the name from the pop up and paste in the field as shown below

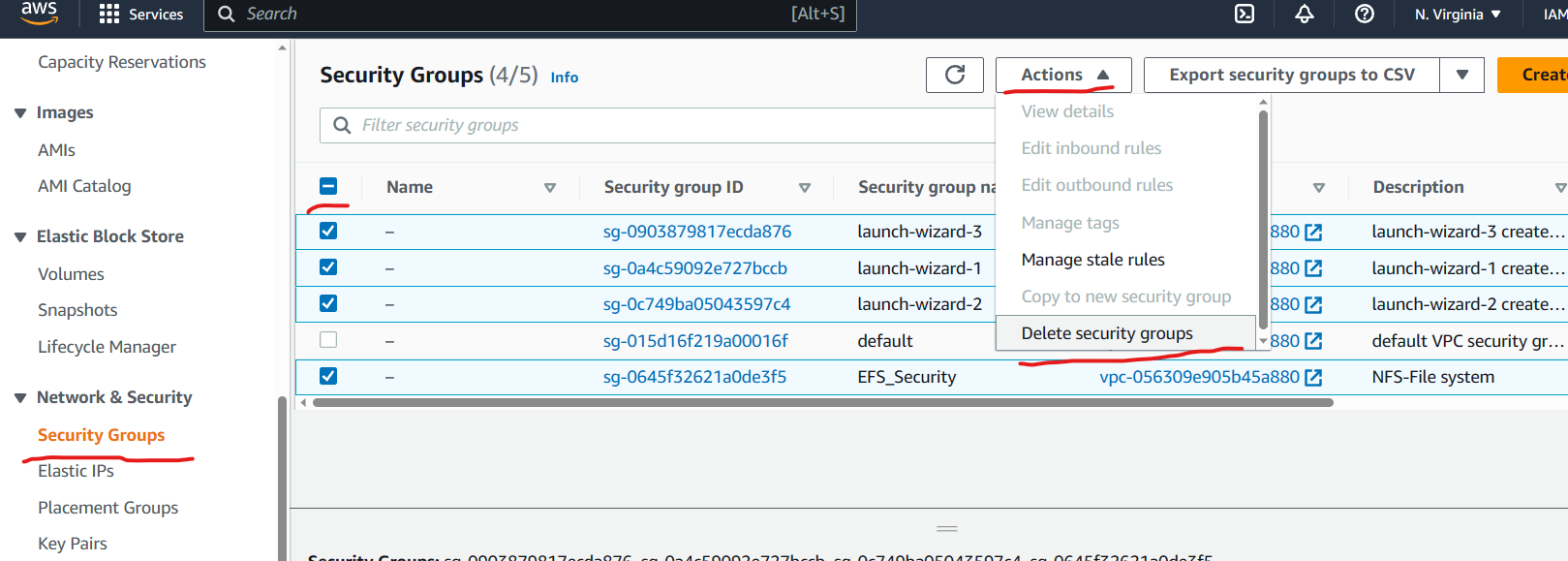


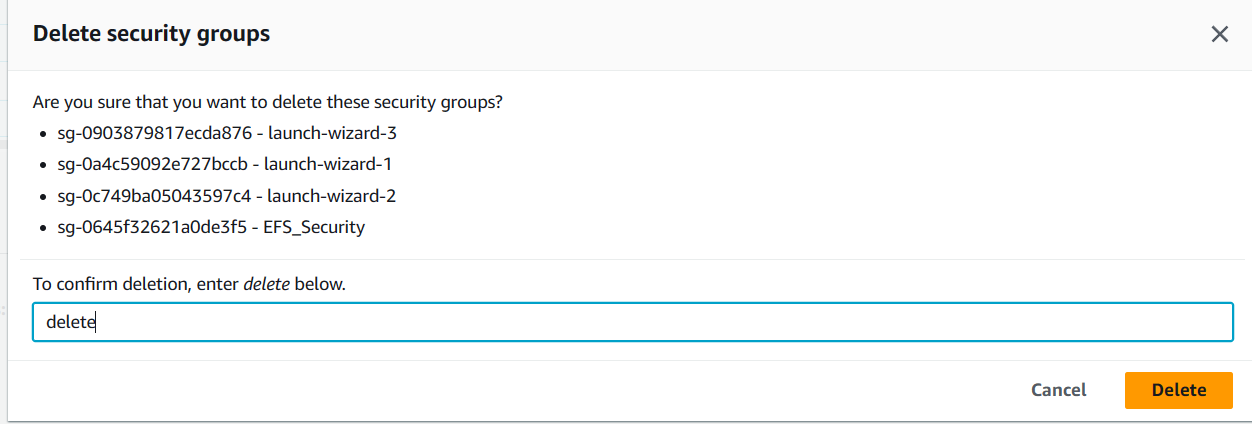
Click on **confirm**

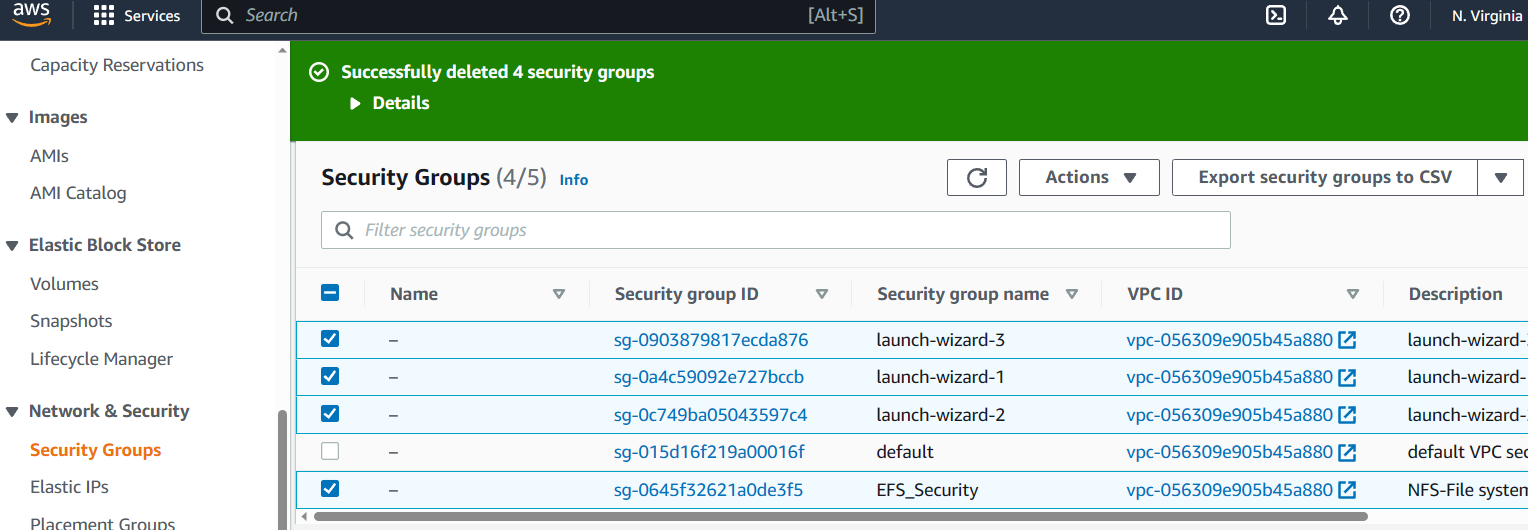
You would see the below output after few seconds



In the EC2 Services, visit the **Security group** menu under **Network and security**  
Select all the security groups that were used in the above practice (except Default security group)   
Click on actions and click on **delete security group** as shown below



Enter “Delete” in the pop up and click on delete button  




When should I use Amazon EFS vs. Amazon Elastic Block

Store (EBS)?

One instance may access an EBS volume at a time and is the maximum number allowed. Contrarily, EFS enables tens of thousands of instances to access the file system.

EFS is the best option if you want to swiftly transition your enterprise apps to the cloud without affecting the architecture. EBS volume is probably the best option if you need to operate a NoSQL database (like MongoDB and Cassandra) on your instance because to its low latency and consistent nature.