Project 01

Course: Creating a Custom VPC from scratch

Learning Outcomes

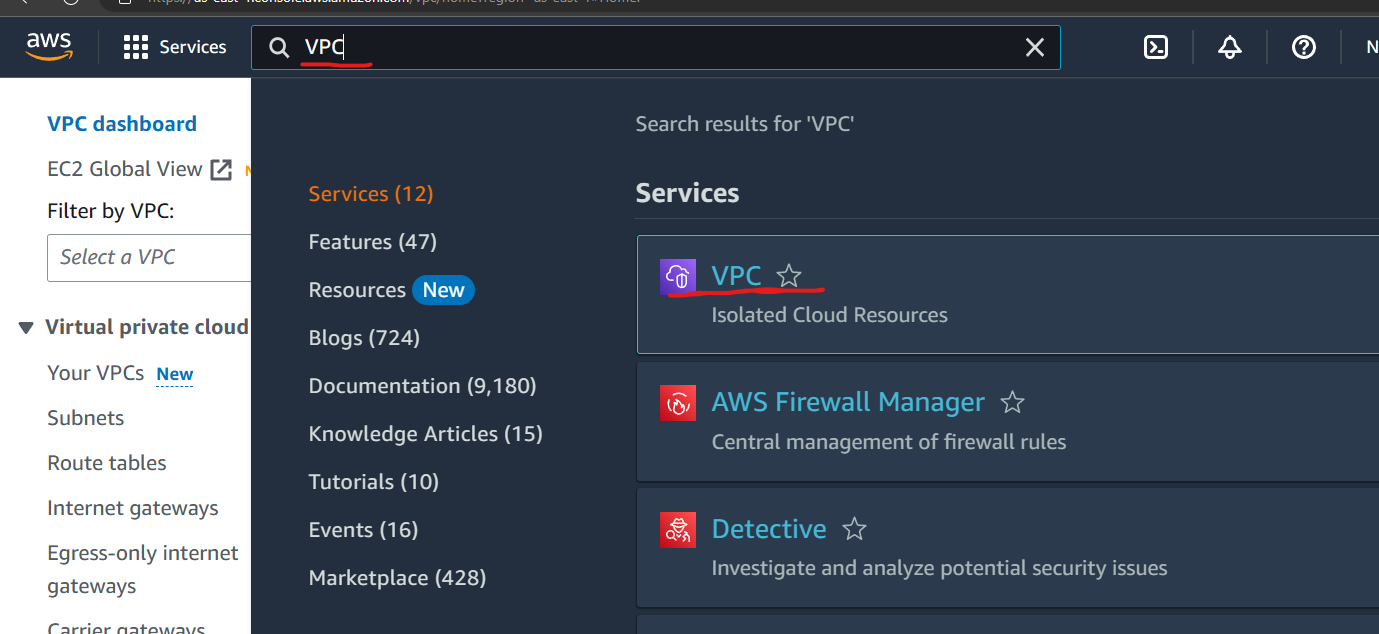
1. Be able to create Custom VPC and its components (Subnets, Route

tables, Internet Gateway..)

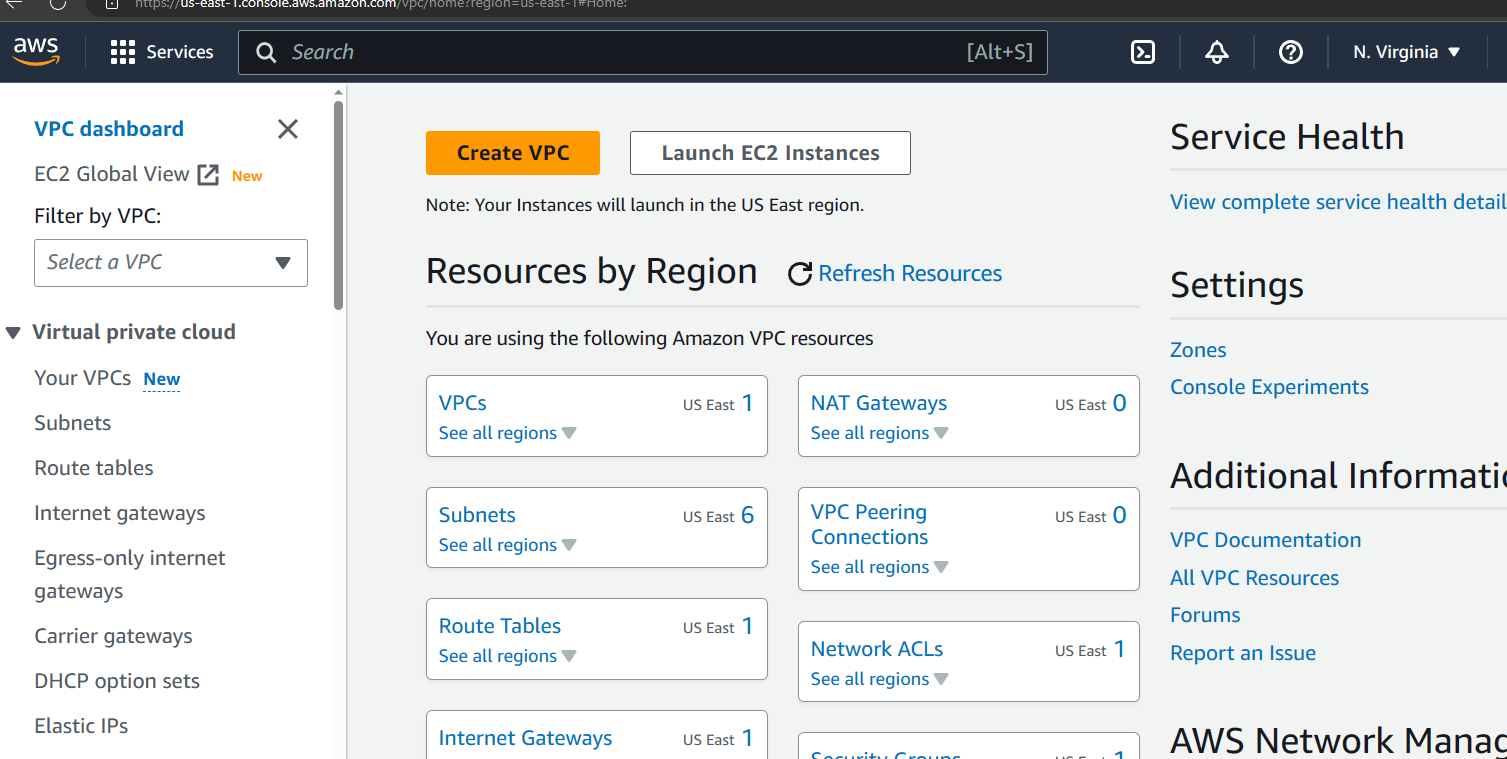
2. Connecting Private Subnet instance from public subnet instance

3. Understanding NAT

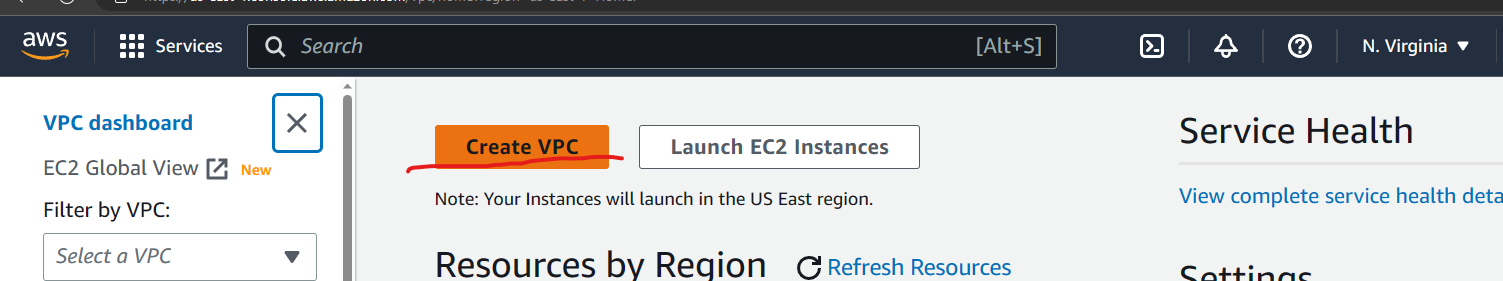
Create a VPC from the VPC services in AWS management console



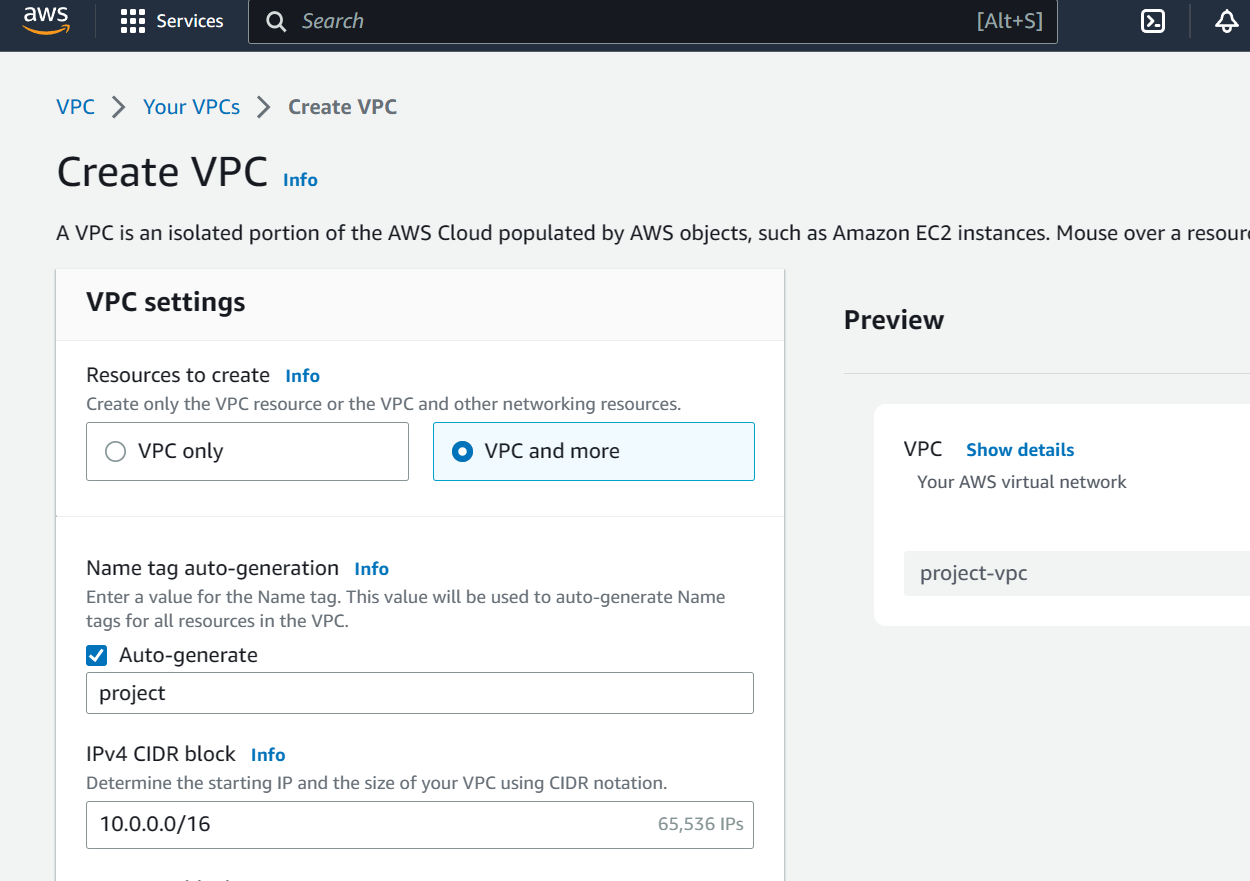
You will reach the below page.



Click on create VPC button

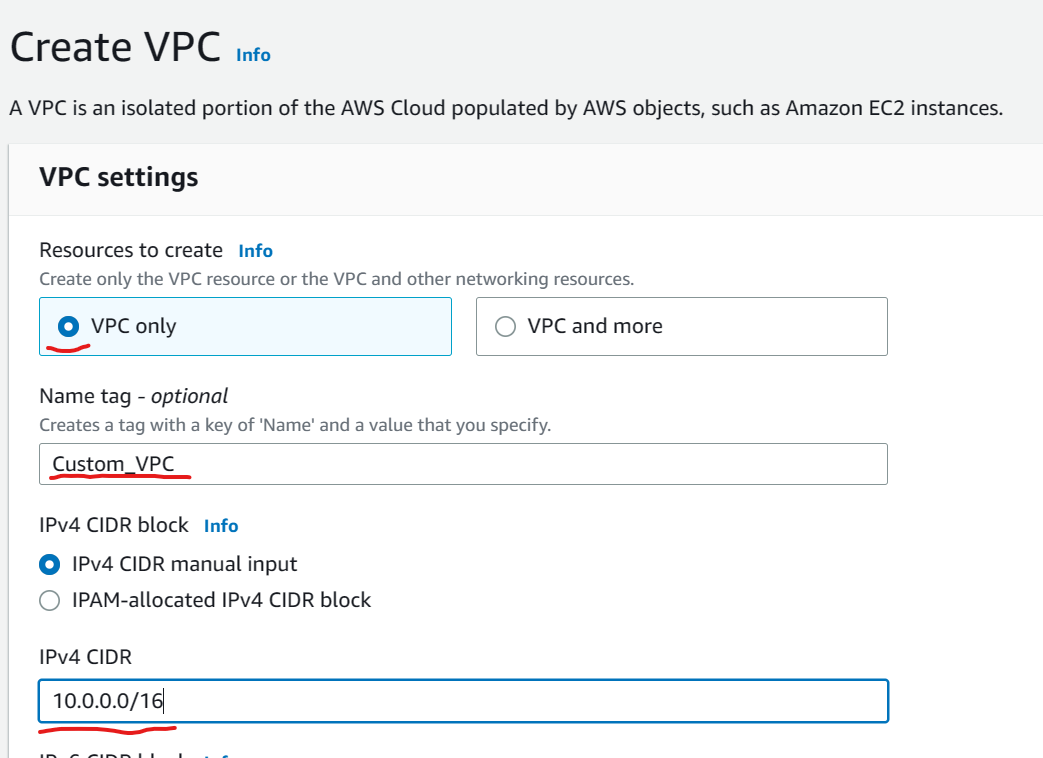


The following webpage will appear

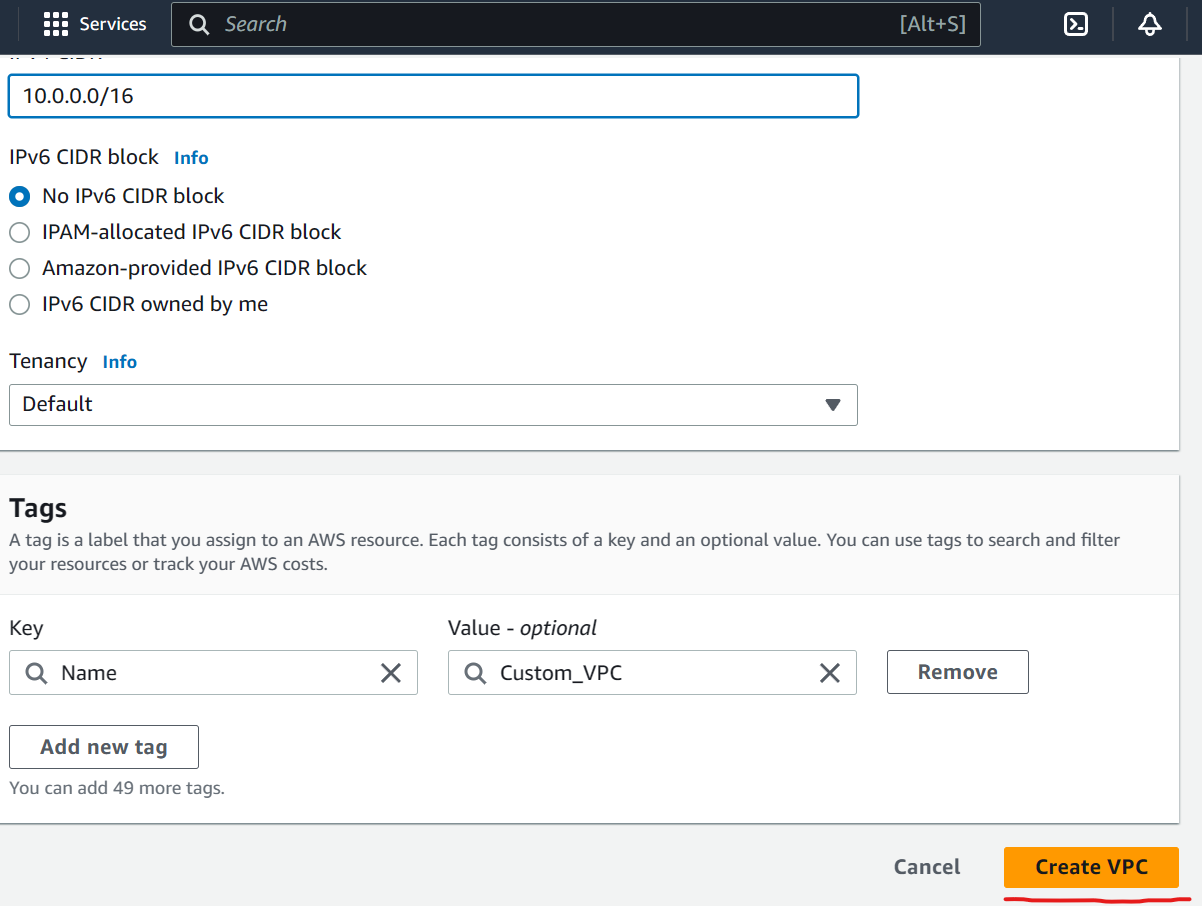


Select “VPC only” and Specify a desired name

Specify the CIDR as “10.0.0.0/16”



Scroll down and click on “create” button

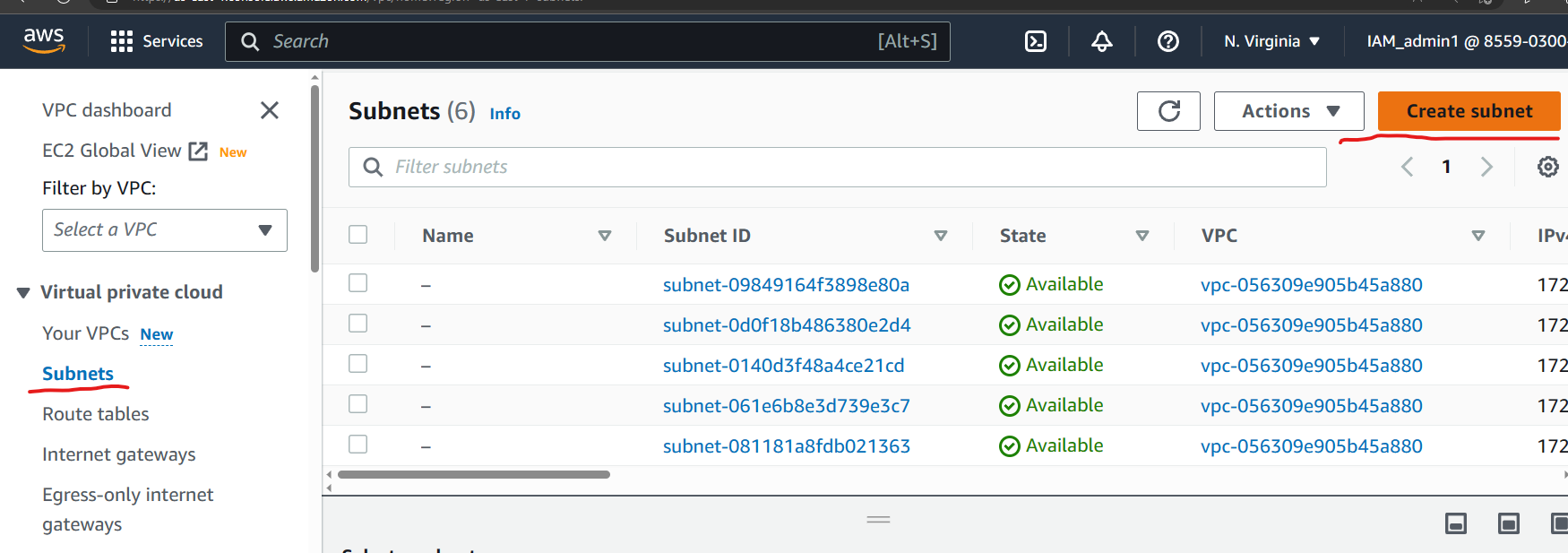


The following page will appear

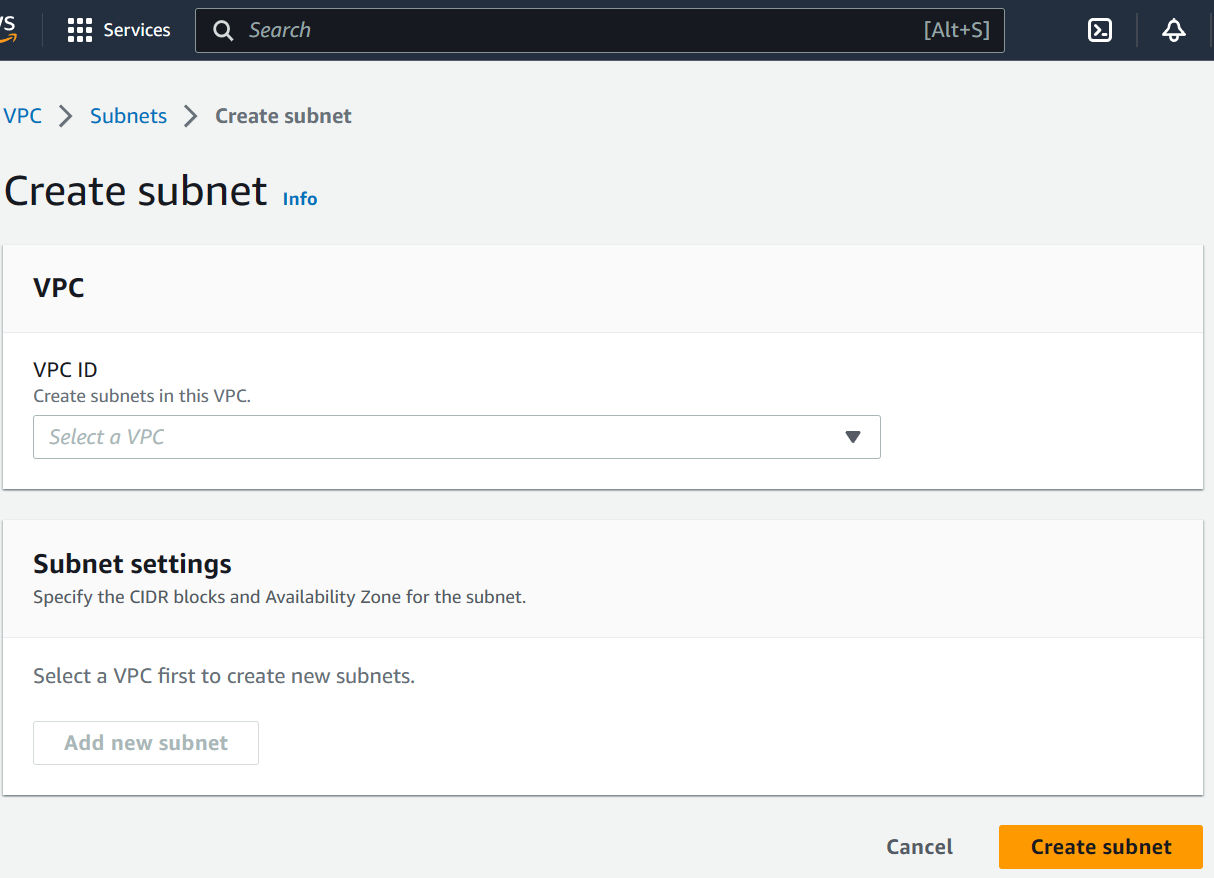


Congratulations you have created a VPC successfully

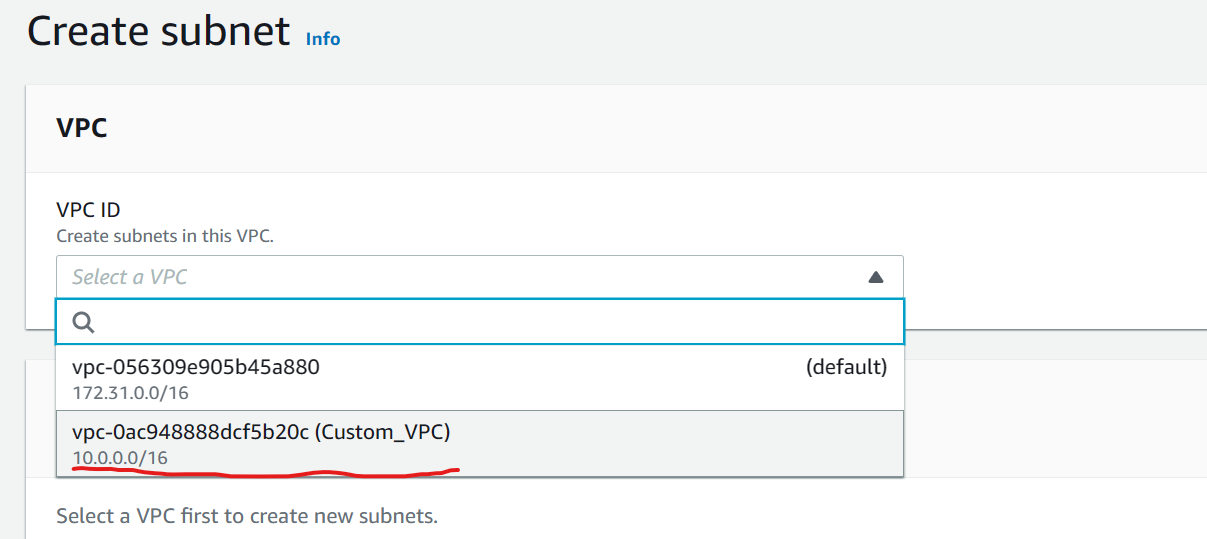
Visit the **Subnets** menu from the left panel and create 2 subnets



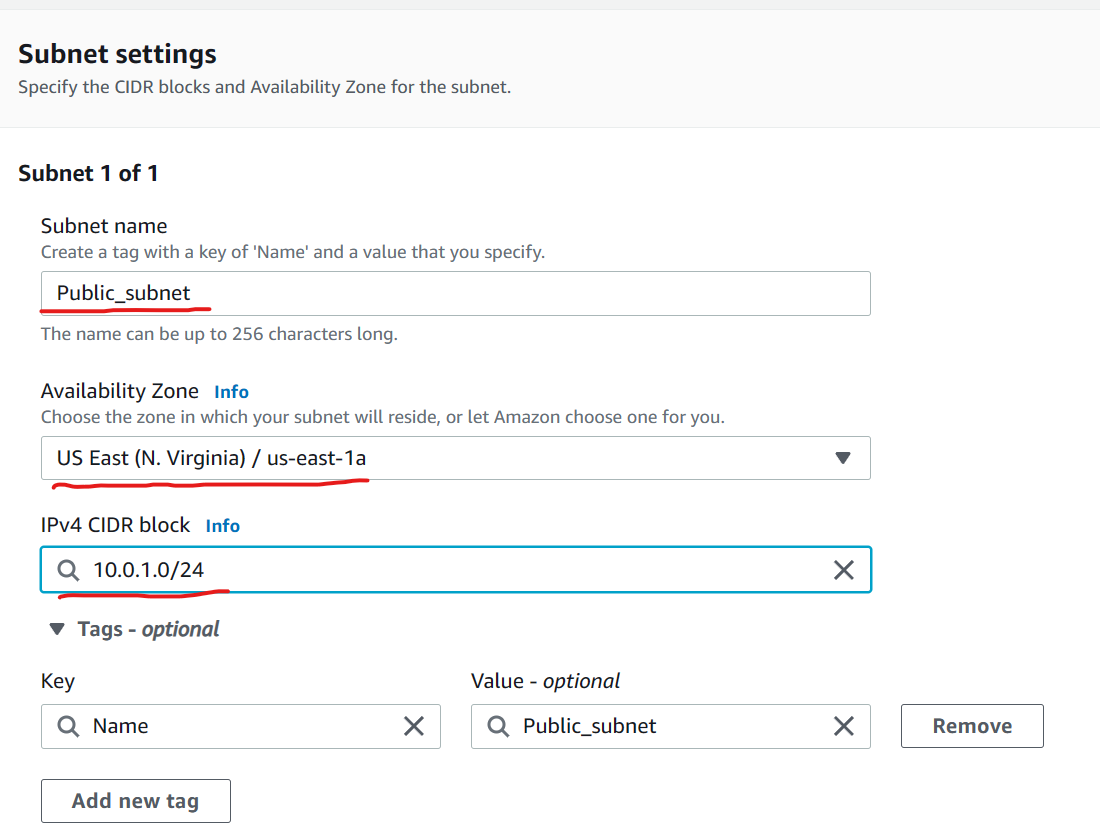
The page shown below would appear



Select the custom vpc that you had created earlier from the VPC dropdown menue

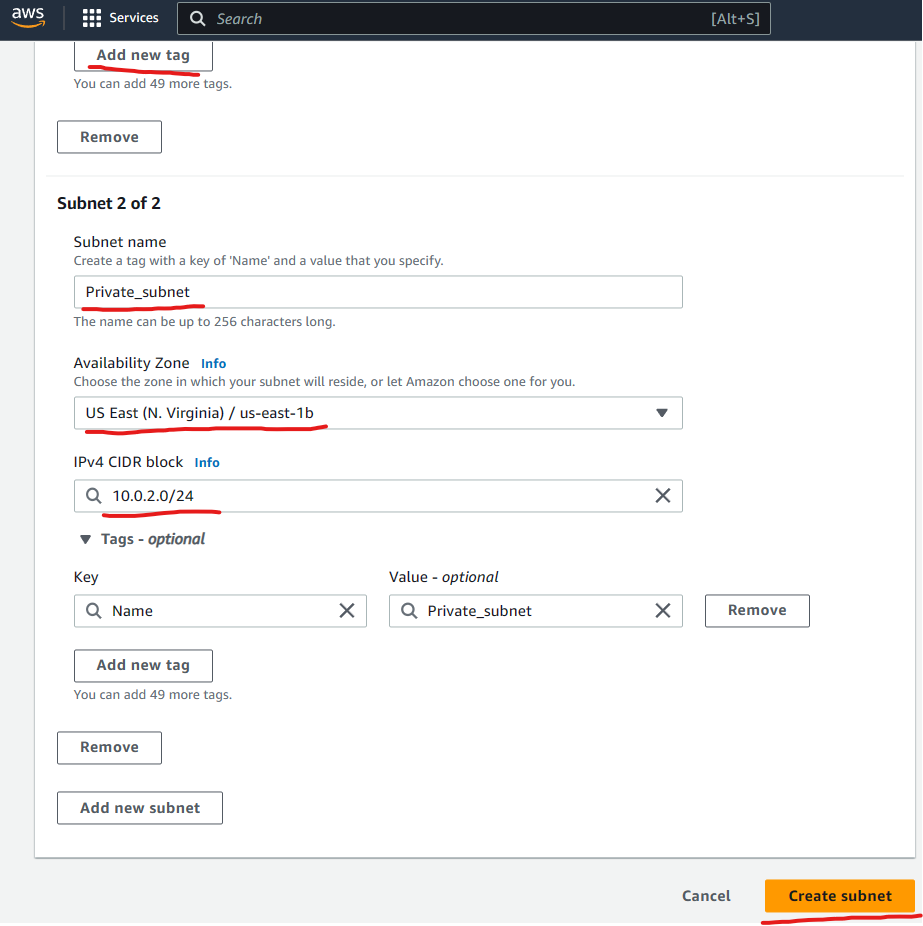


Scroll down and fill the subnet setting as

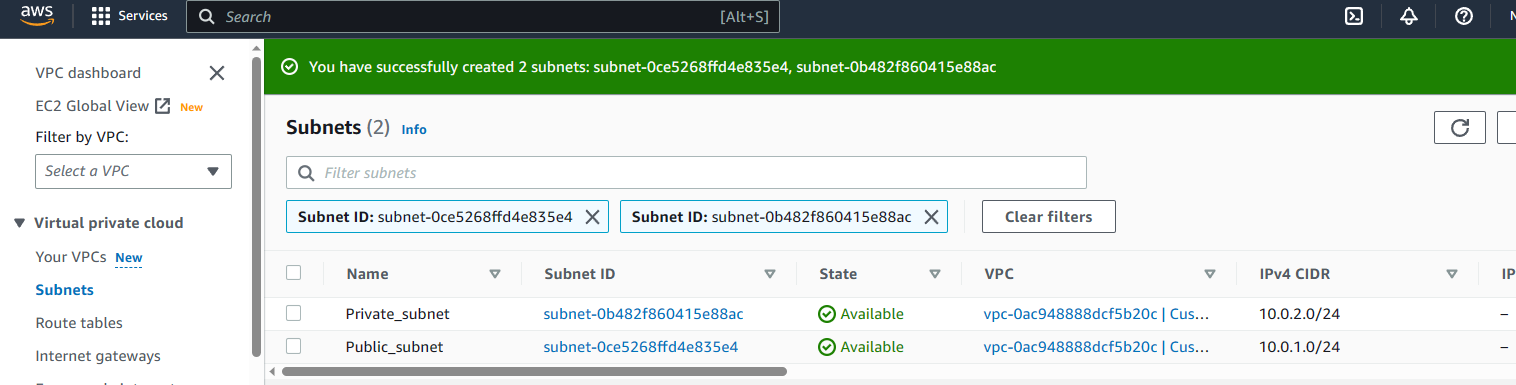
Name: Public\_subnet  
Availability zone: us-ease-1a  
ipv4 CIDR block : 10.0.1.0/24  
as shown below  


Scroll down, click on **add new subnet** button and fill subnet details for a private subnet as follows.

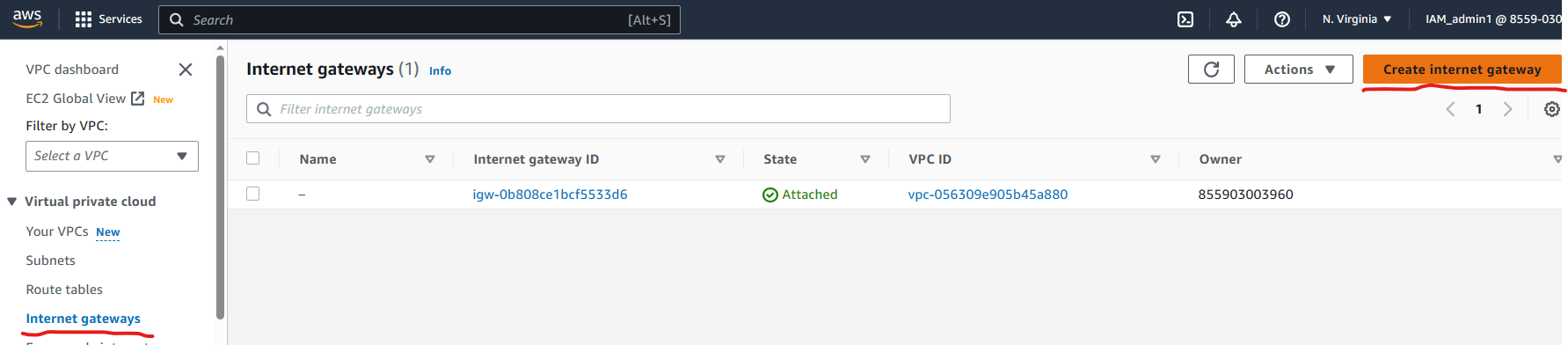
Name: Private\_subnet  
availability zone: us-east-1b  
ipv4 CIDR block: 10.0.2.0/24



Scroll down and click on create subnet button

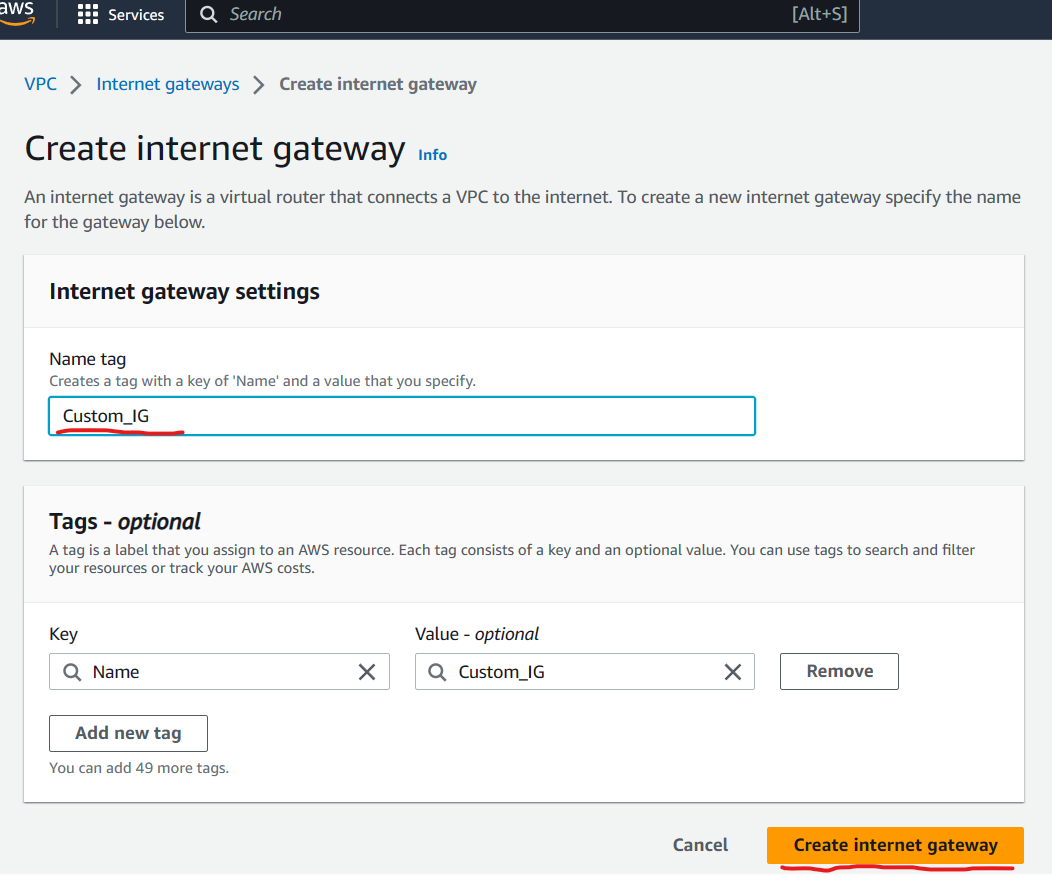
You would receive the below page   


To grant internet access to the VPC you would need to create an internet gateway.  
you can create an internet gateway from internet gateway menu on the left

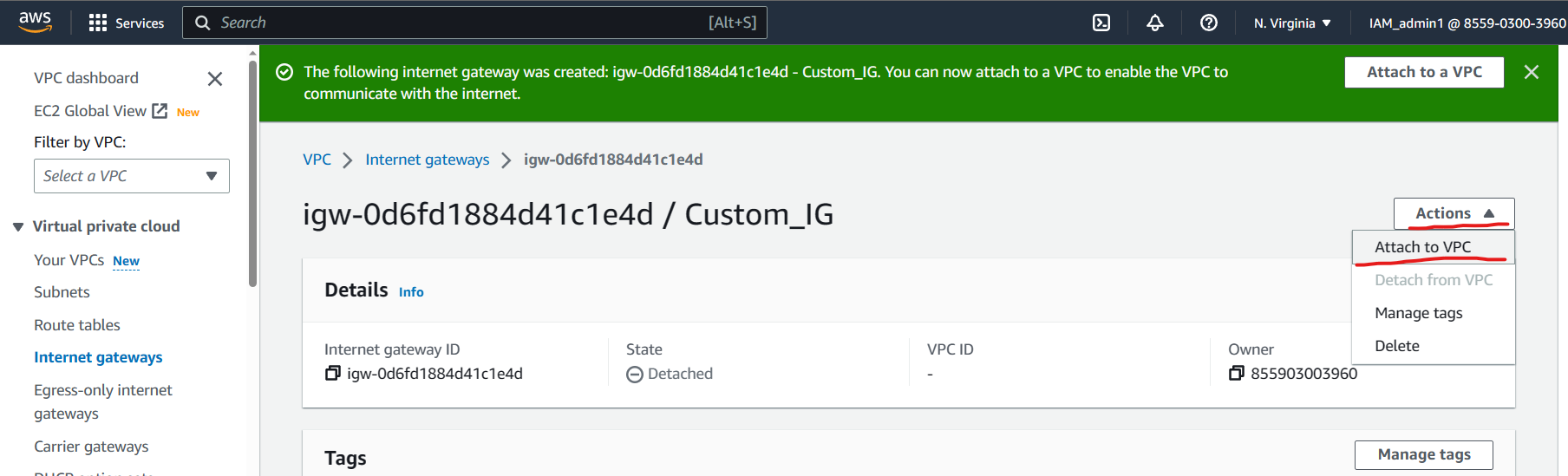


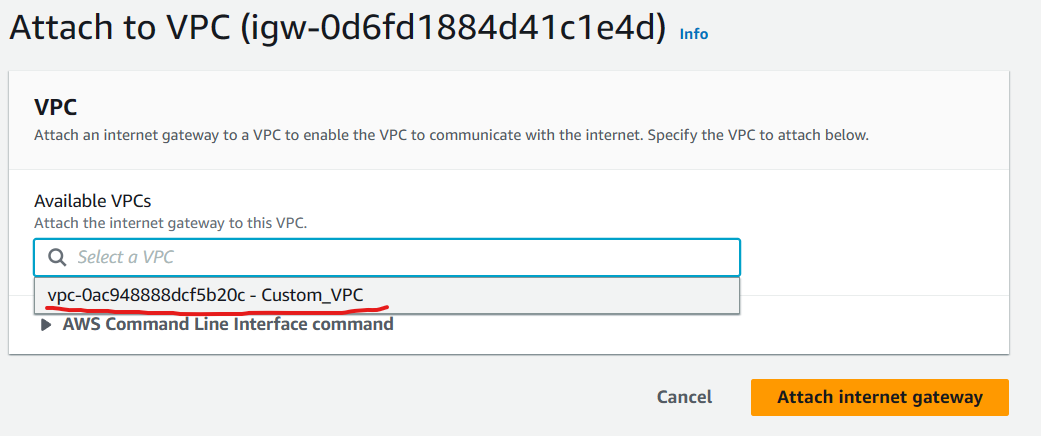
You would come across an existing internet gateway, Please ignore as that gateway is being used by the default VPC.  
Proceed to create a new Internet gateway for the custom VPC.

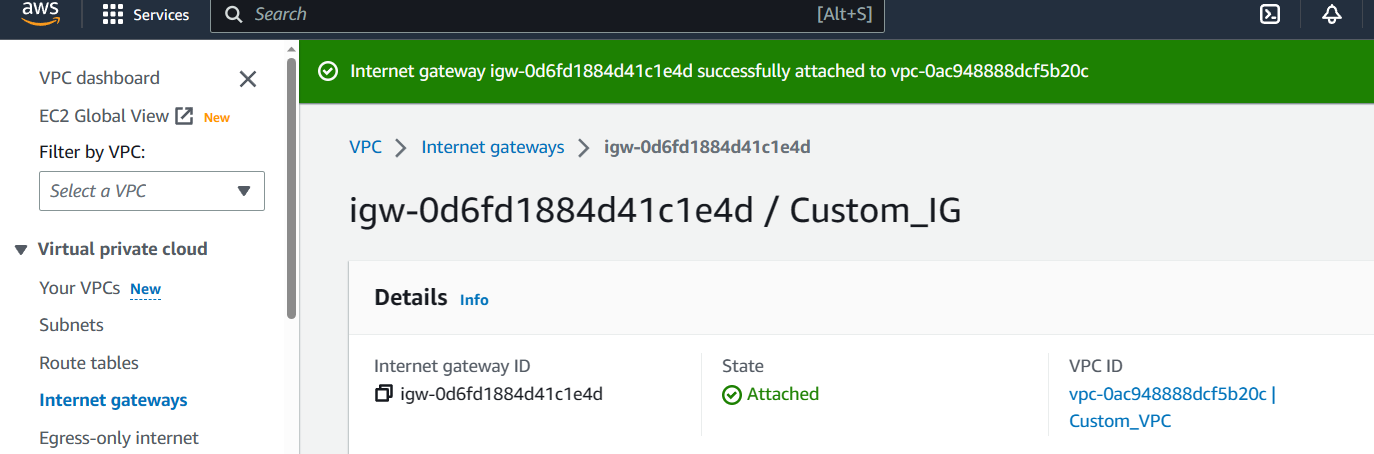
Specify a name in the name field and click on **create internet gateway**



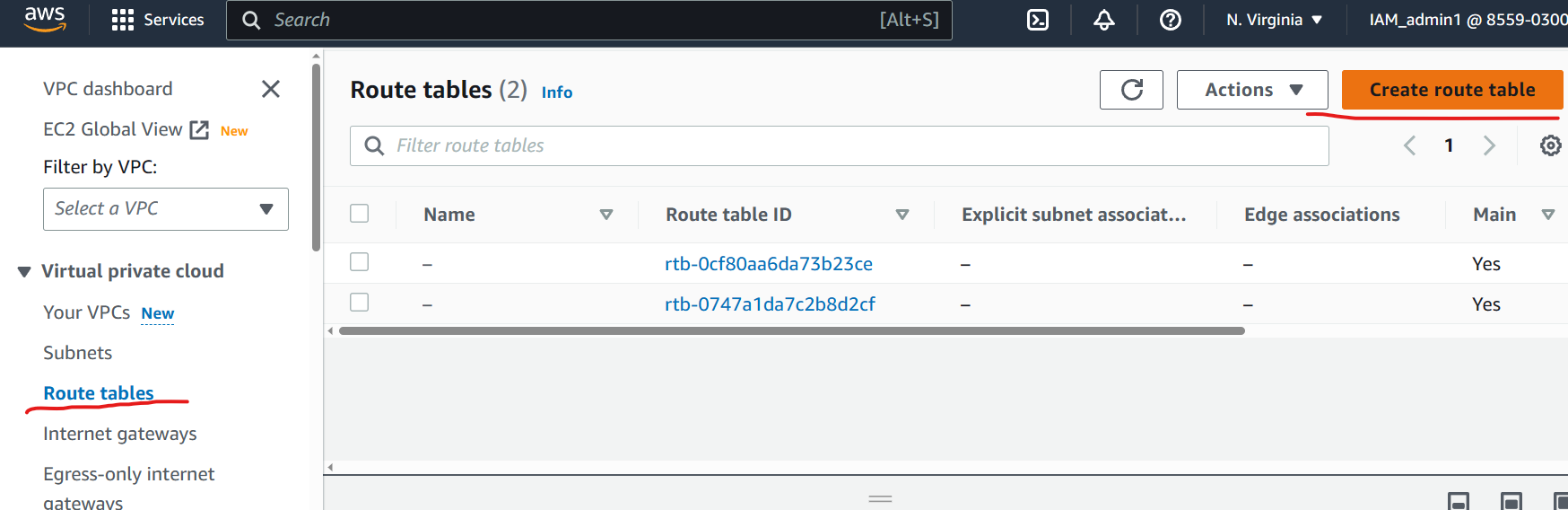
The following page would appear

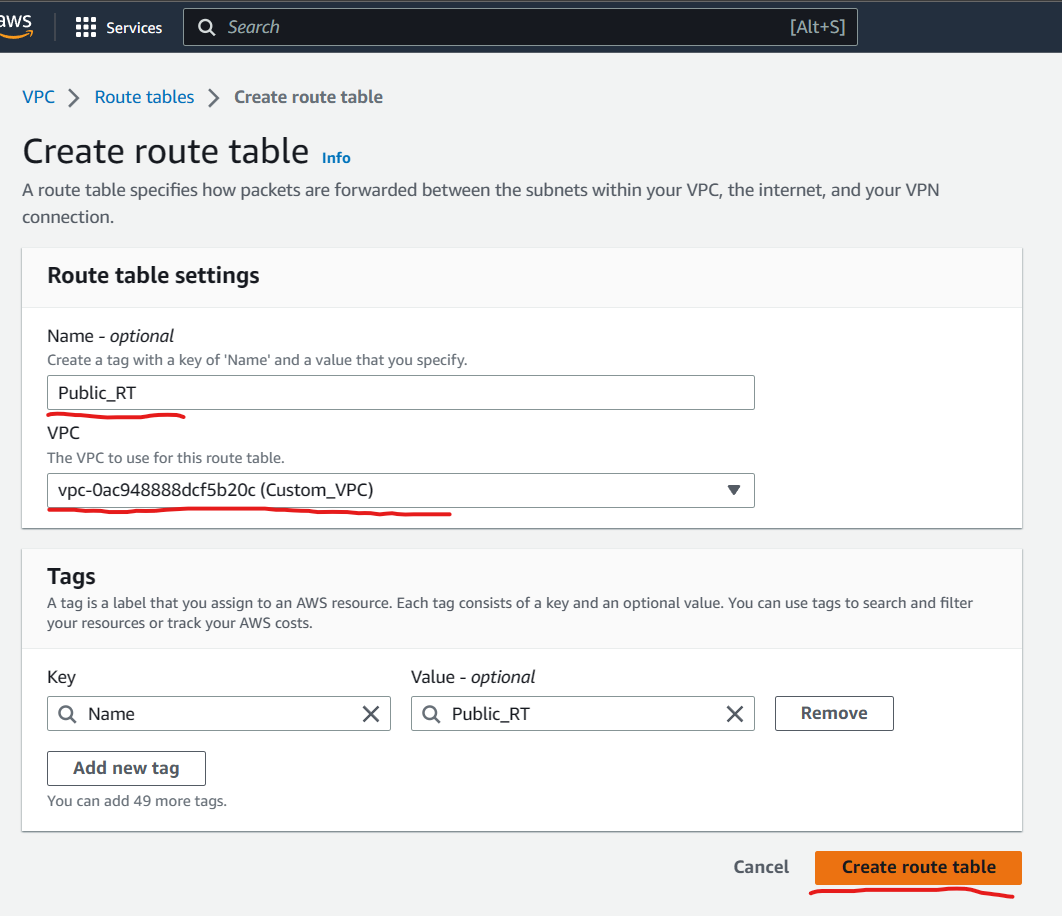


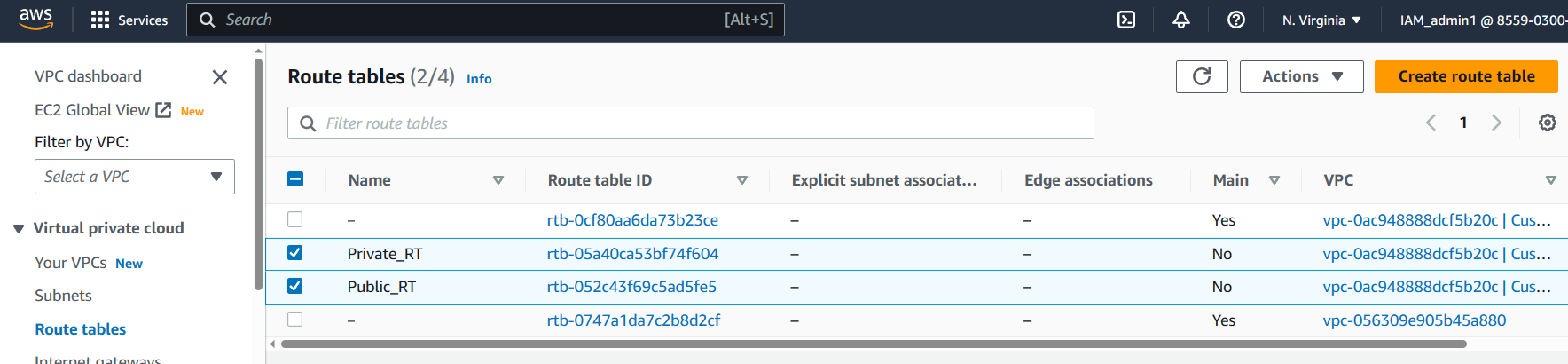
Click on actions button on the right and select **attach to VPC**  
The following page would appear  


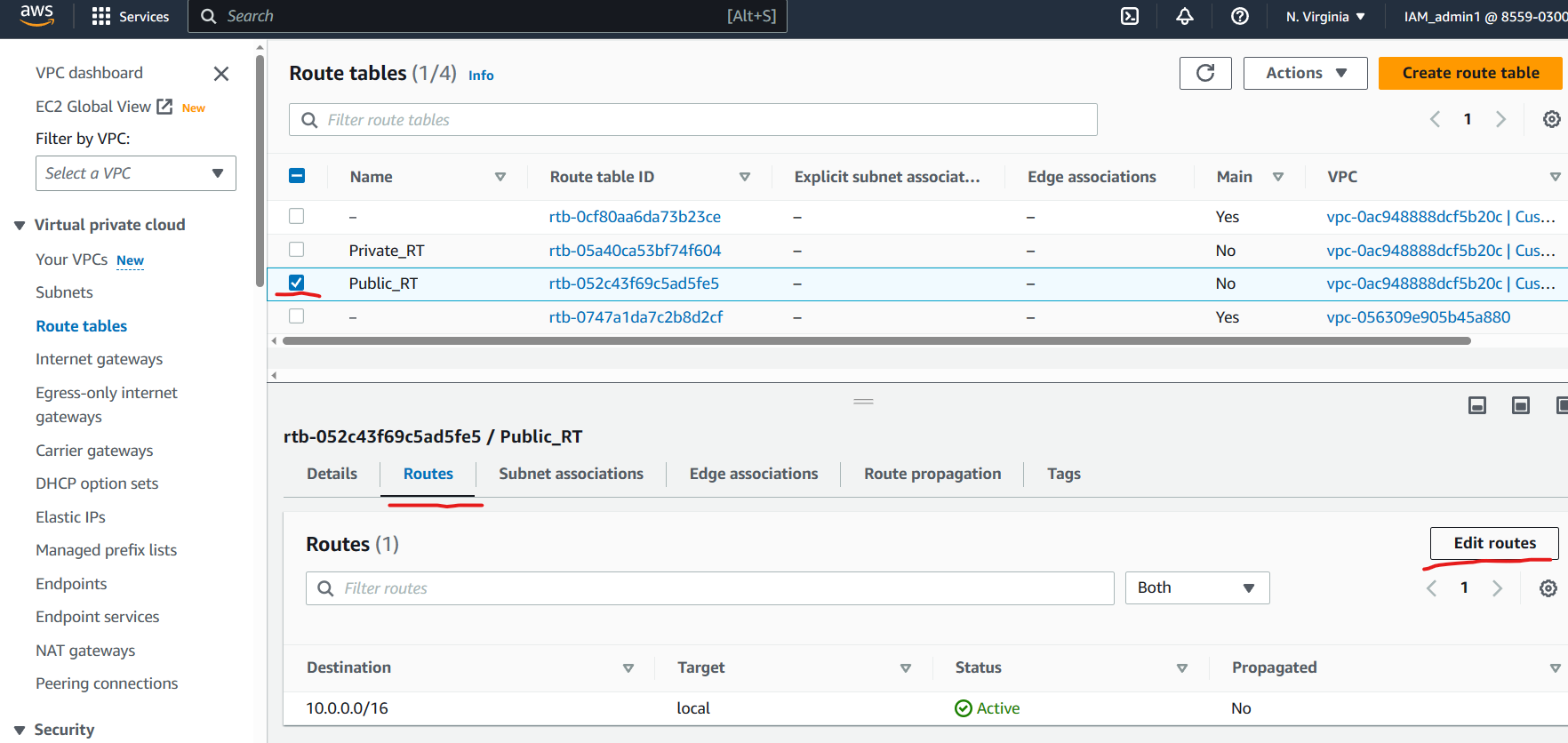
Select the custom VPC which was created earlier and click on **Attach Internet gateway** button at the bottom right.  
you will receive the below output page  


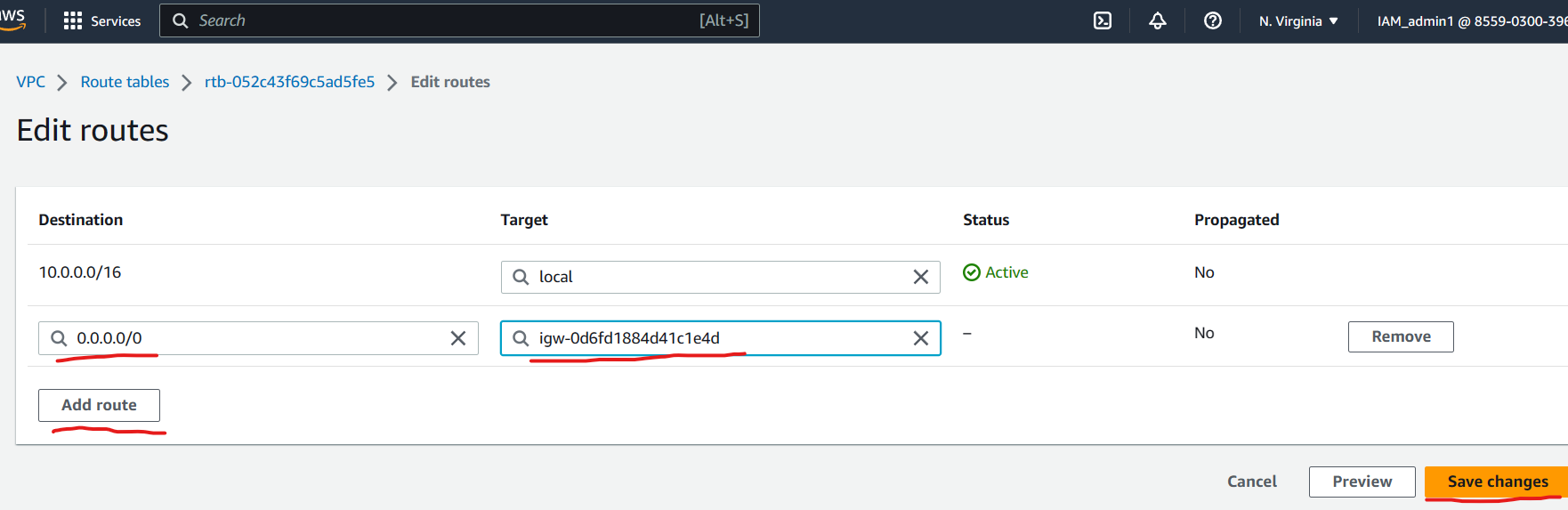
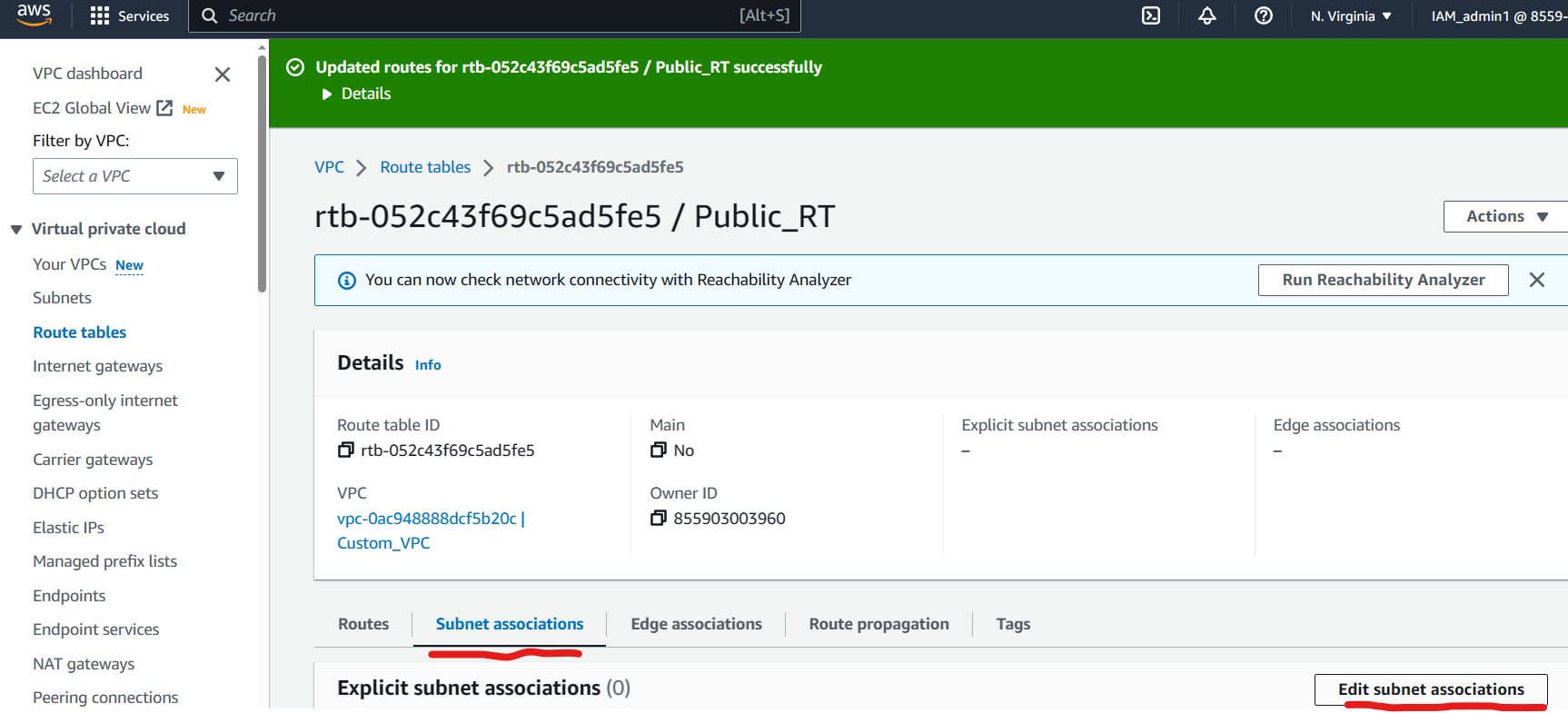
Now to create Public and private route tables.

To create route tables visit the **Route Table** menu on the left and clicking on **create route table** button as shown below  


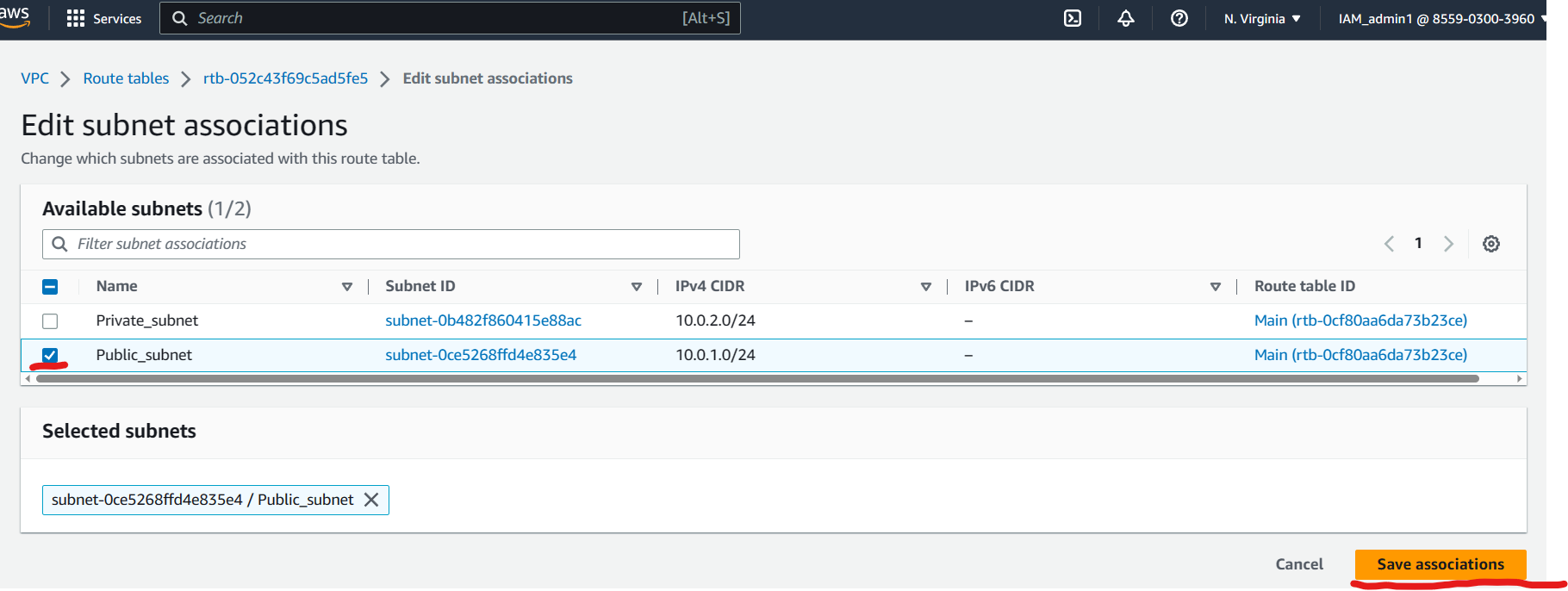
Fill up the route table name and select the custom VPC and click on create route table as shown below.  


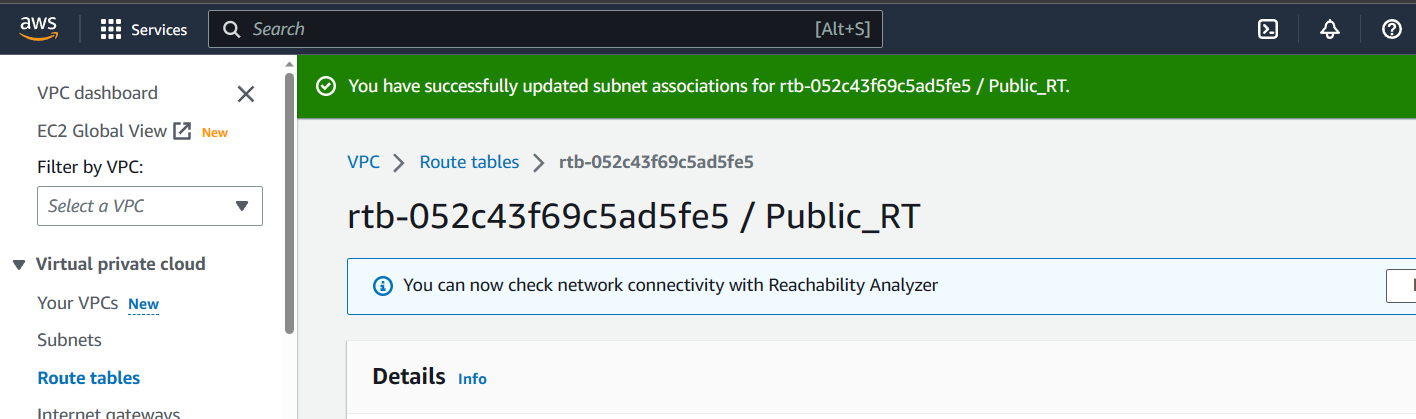
Similarly create another route table for Private route table..  


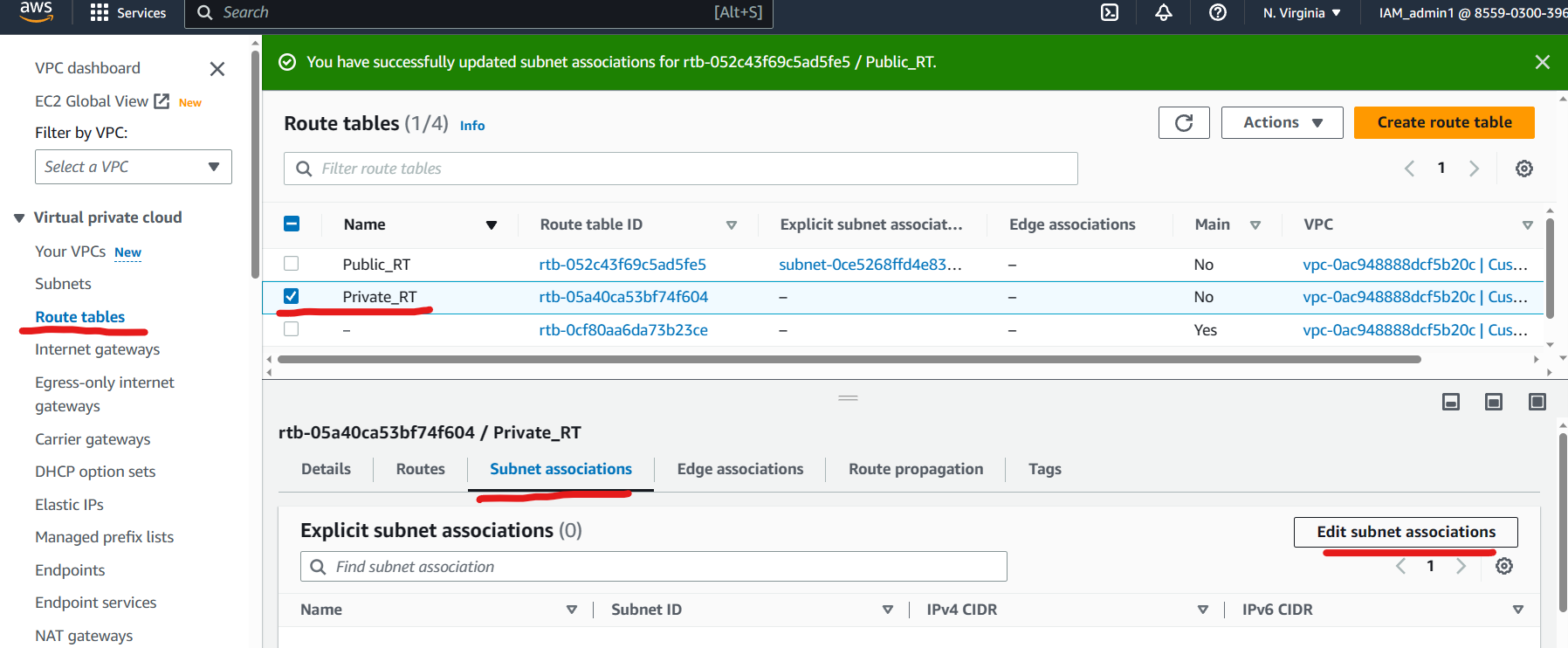
Select the Public route table and go to routs tab as shown below and click on edit route  


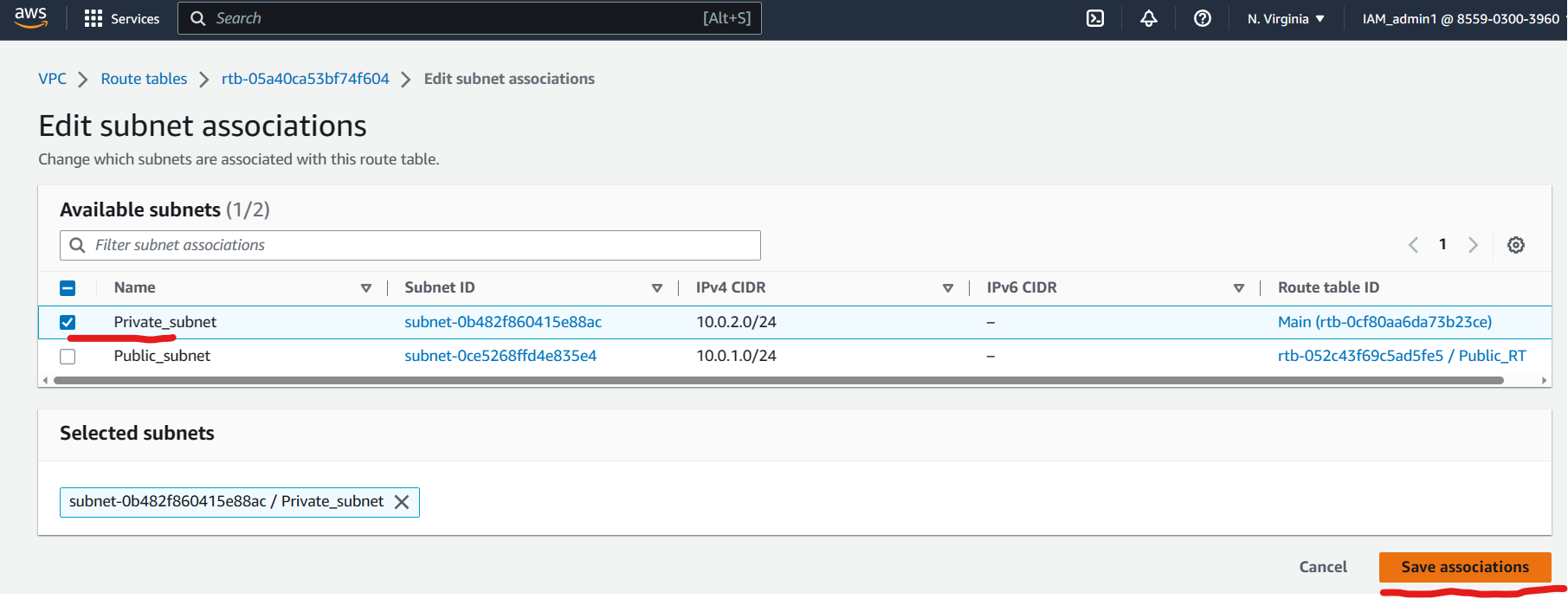
When the below page appears click on **Add** button and enter   
Destination as 0.0.0.0/0  
Target as “Internet gateway” and select your custom Internet gateway option and click on save changes button  
  
After the changes are made visit the **Subnet Association** tab and click on **Edit Subnet association** button as shown below.  


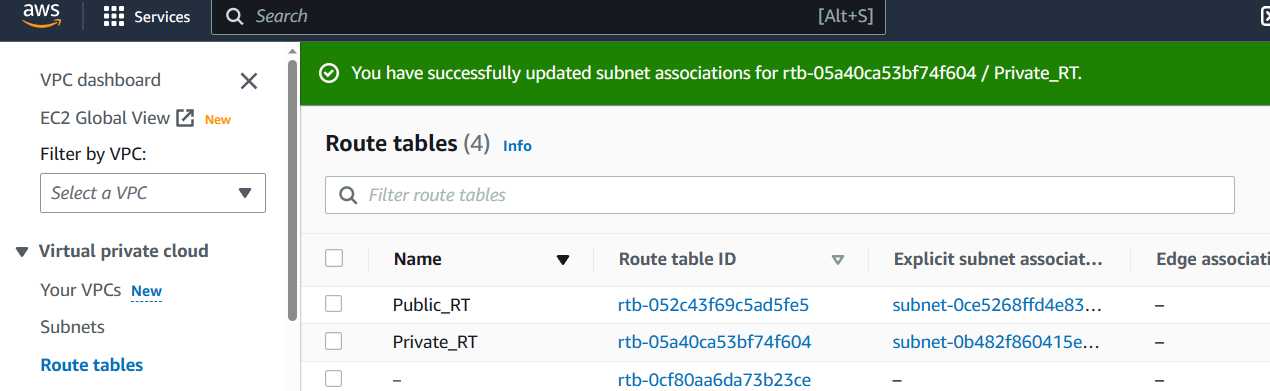
Select the Public Subnet and click on save changes as shown below.



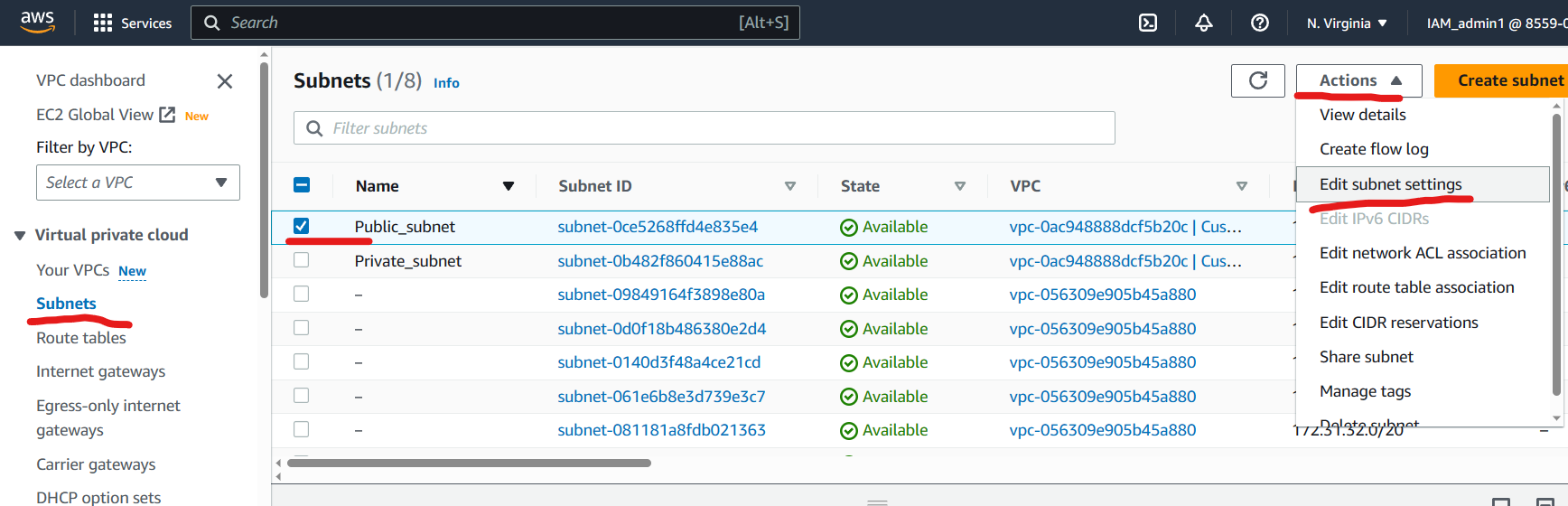
You will receive the below output.  


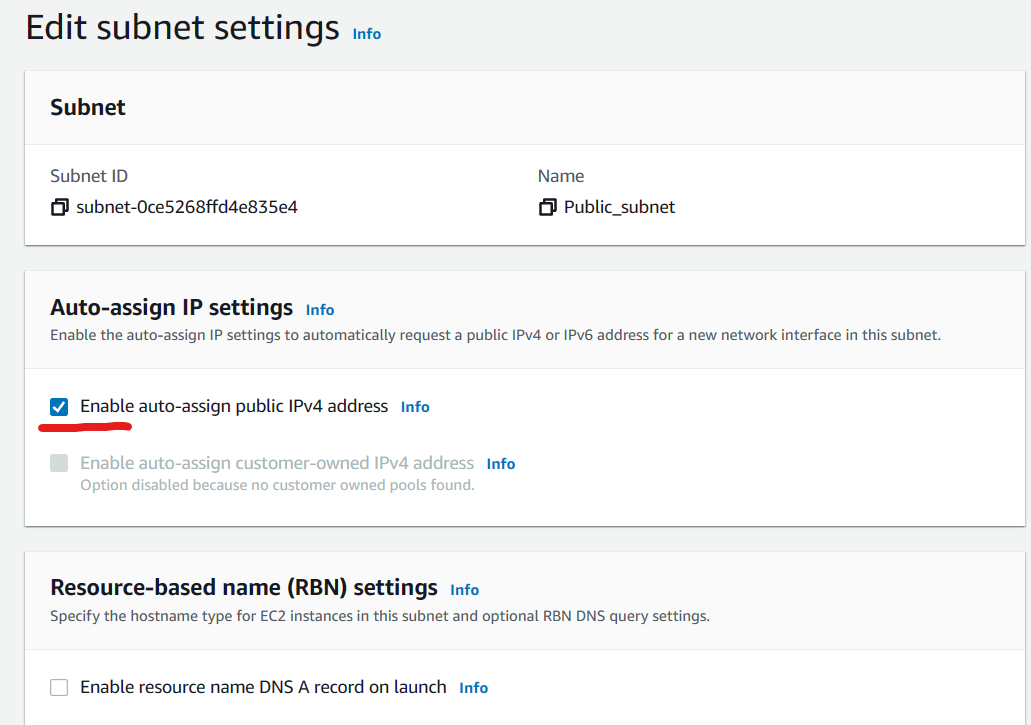
Perform similar steps for the private Route table.  
Visit the Route Table menu and select the Private table that you recently created as shown below.  


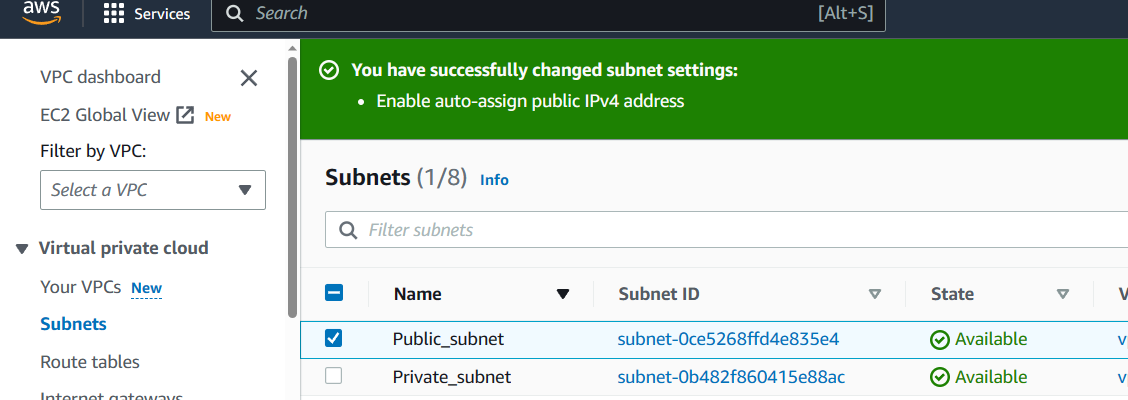
Select the private Subnet and click on save changes as shown below  


You will receive the below output.  


Click on Subnets menu , select the public subnet and click on **Actions** button to edit the subnet settings as shown below.

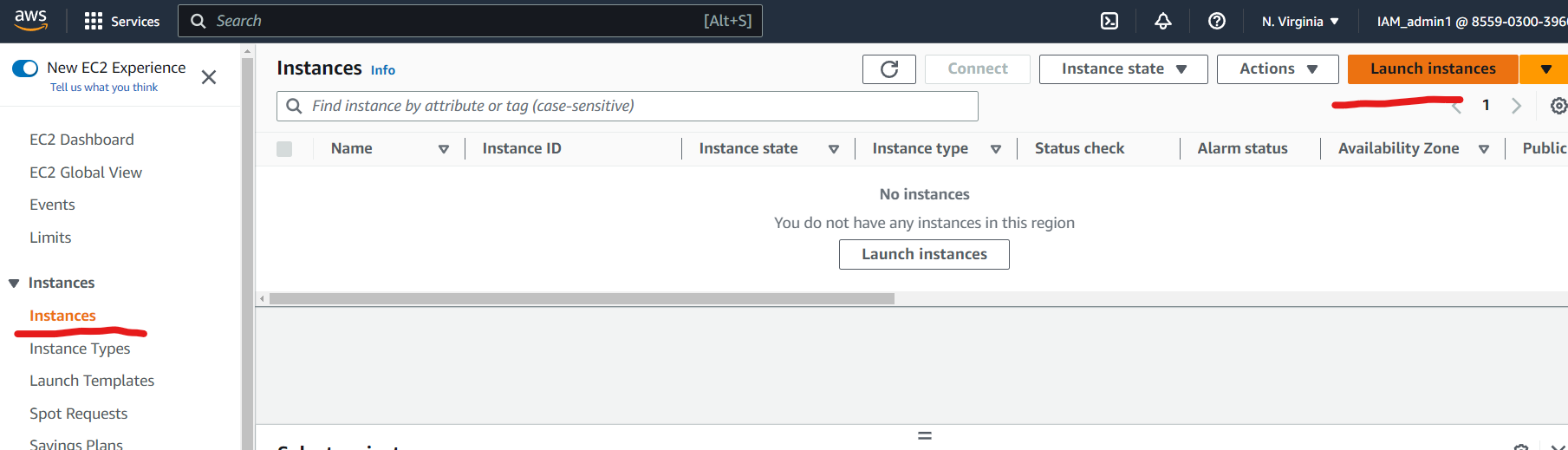


Enable the auto assign public IP address and click on save changes below.  


The below out put would appear  


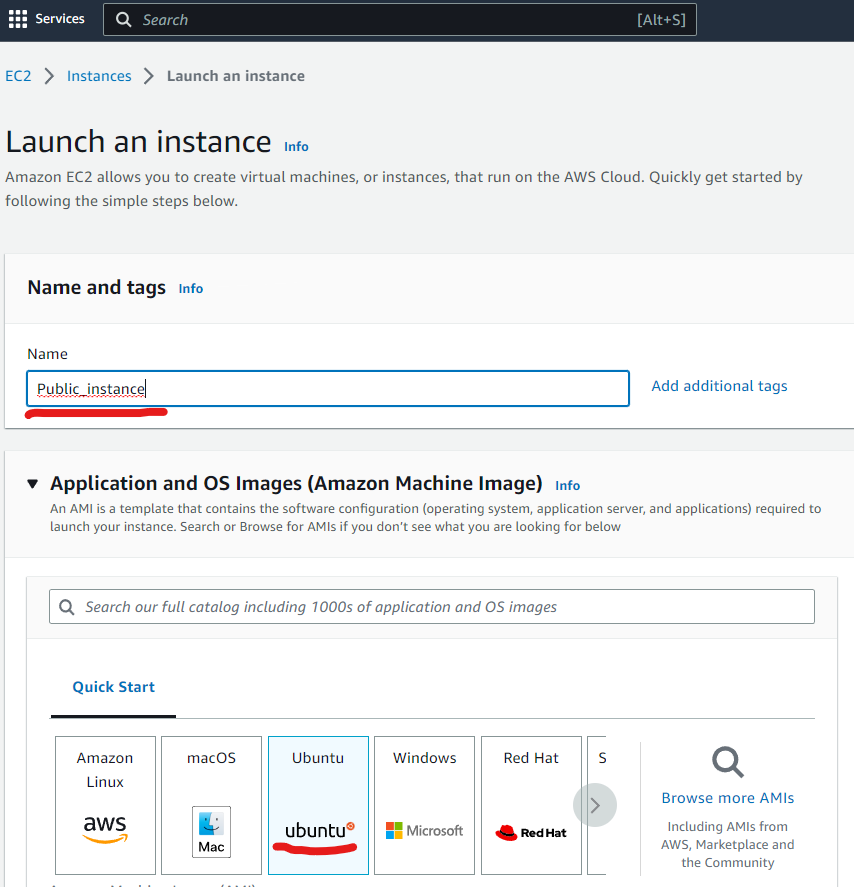
Now to launch 2 EC2 instances in two different subnets.

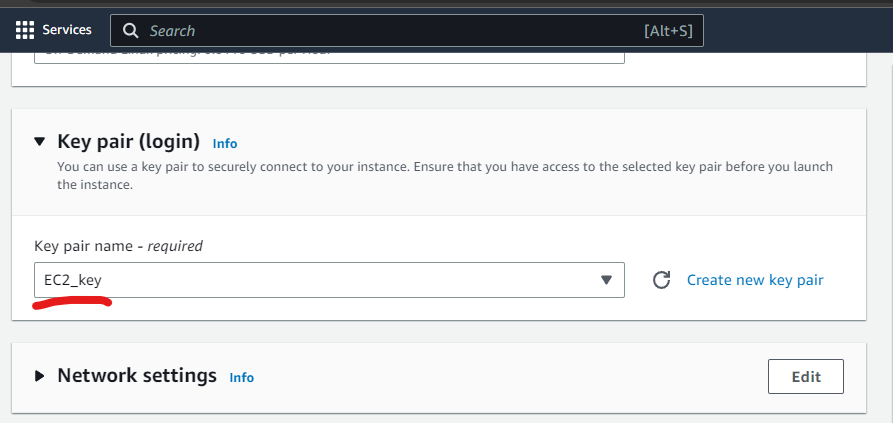
Visit the **EC2** Menu and create 2 instances as shown below.

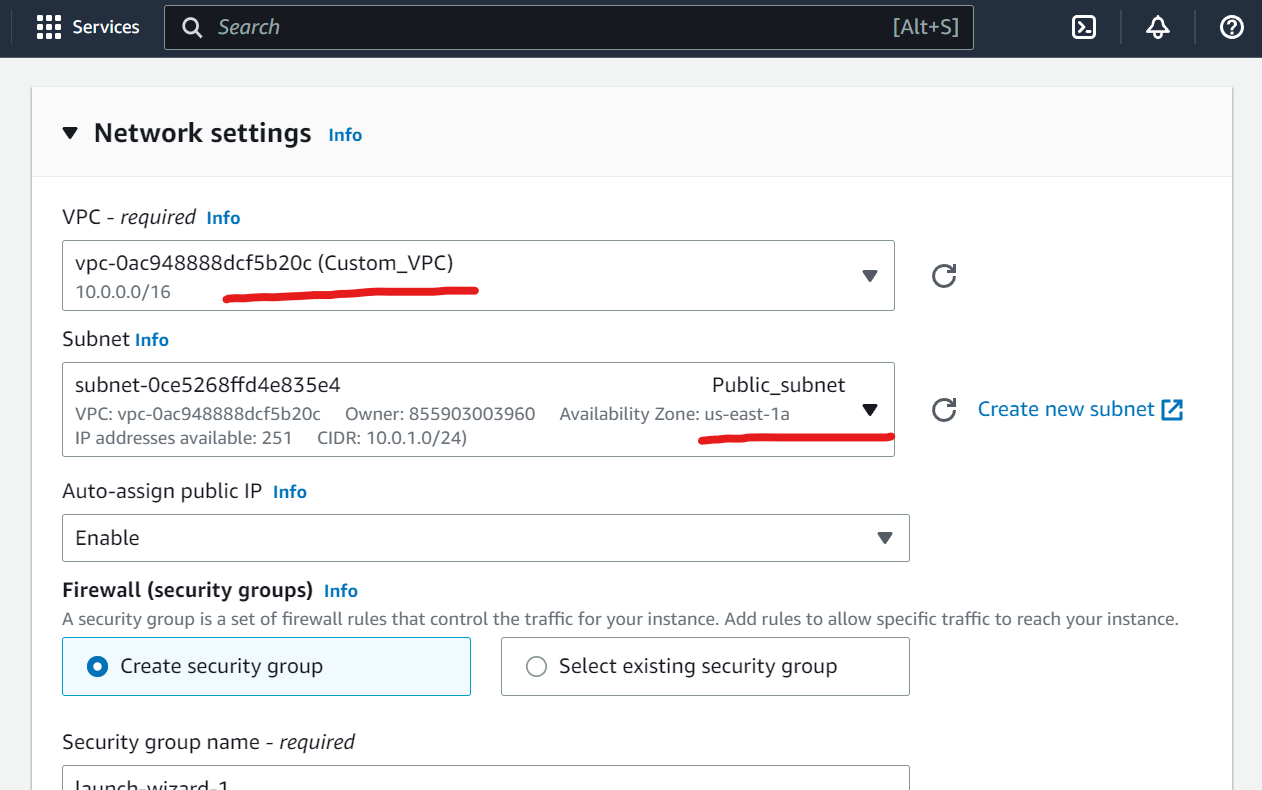


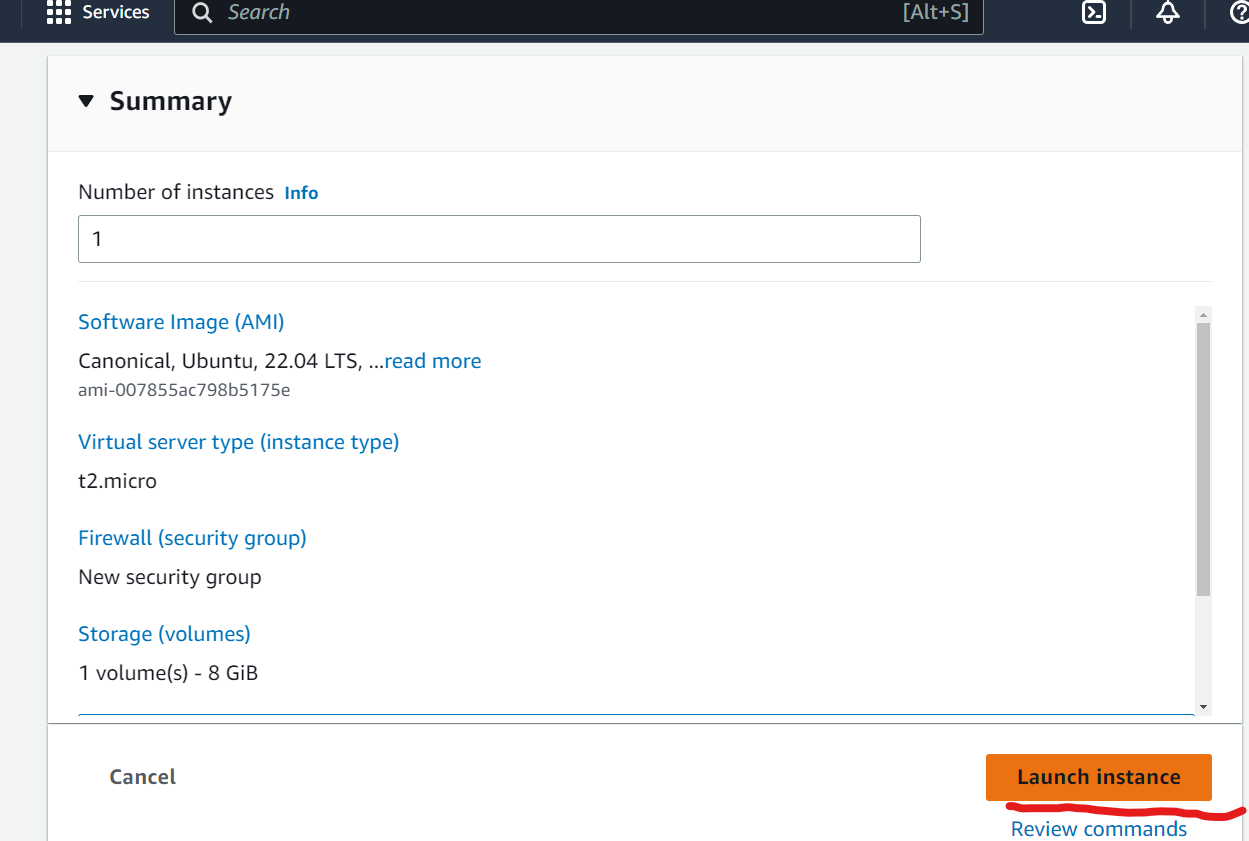
Fill “name” field with your desired name.  
Select “ubuntu” under quick start options.

Select an existing key pair or create new key pair and download the .pem file.

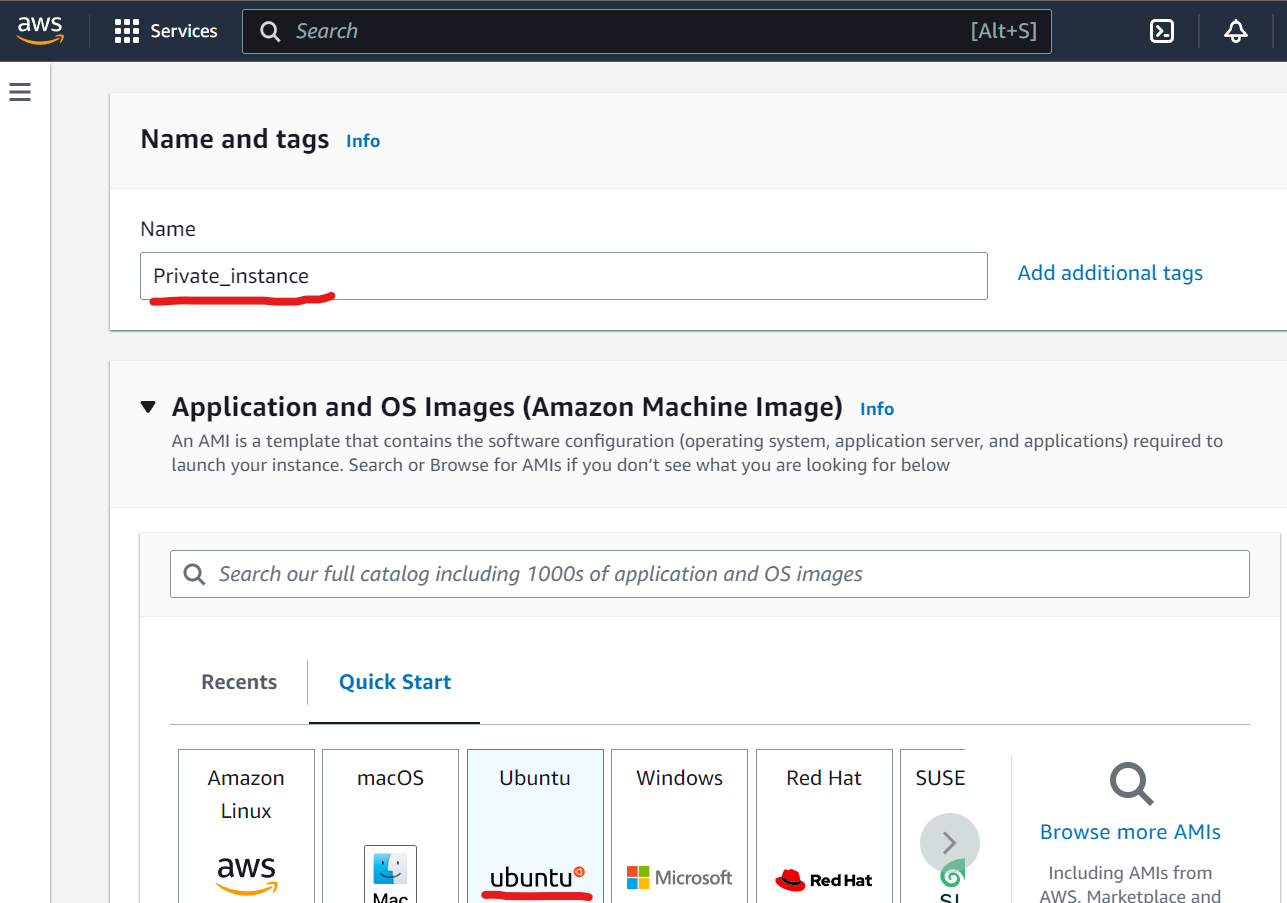




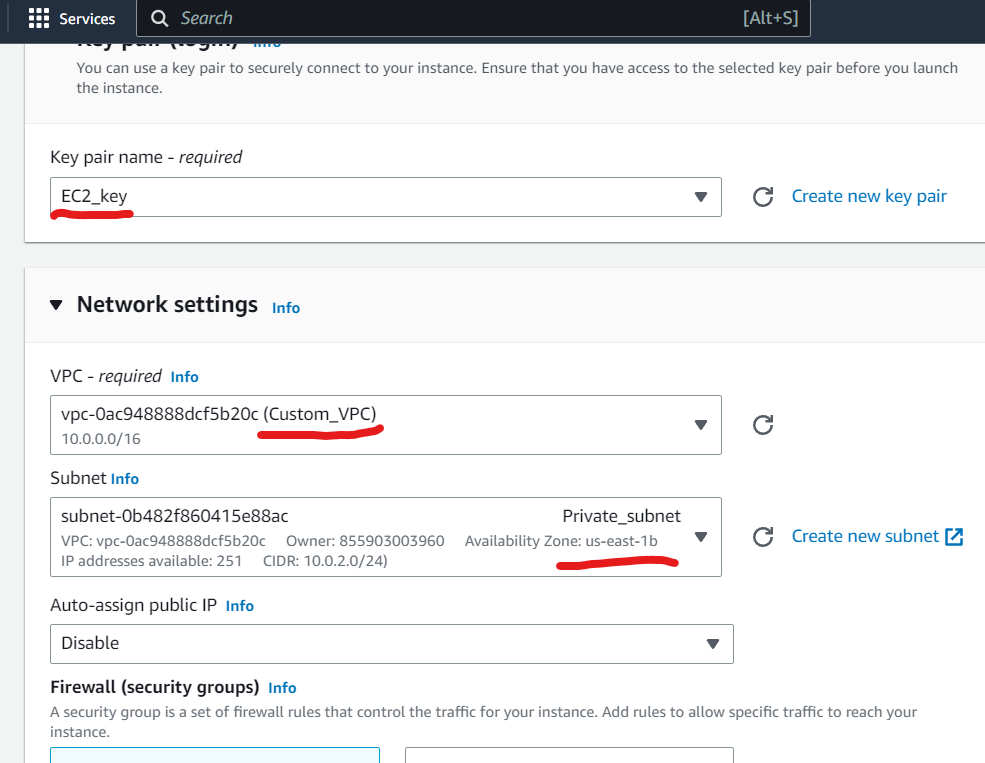
Click on “**Edit** button near Network settings.  
Select VPC as “Custom VPC” which you have created earlier.  
Select subnet as “public\_subnet” which was created earelir.  


Scroll down and click on **Launch Instance**  


Simlarly create another EC2 instance with different name and different Subnet as shown below

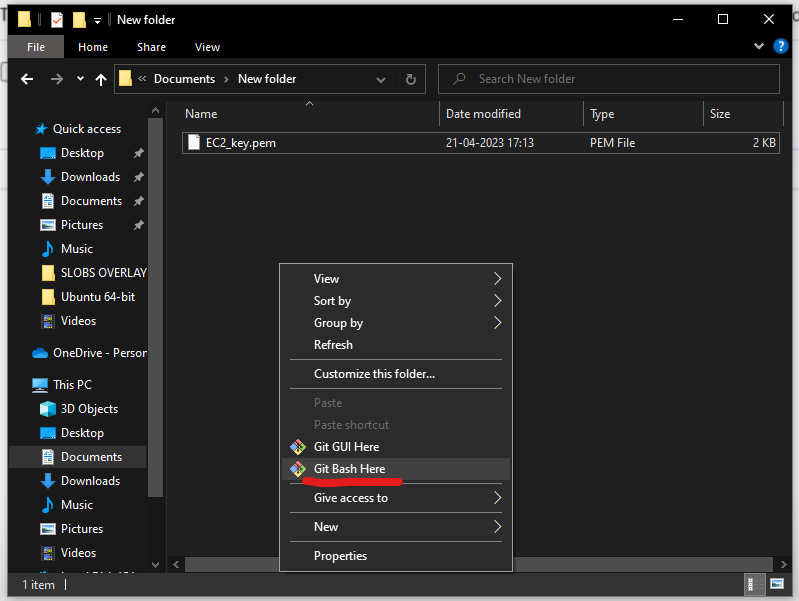


Select the same key pair.  
select the same custom VPC and select the **Private** **subnet** as shown below.

  
Scroll down and click on **launch Instance** like earlier.

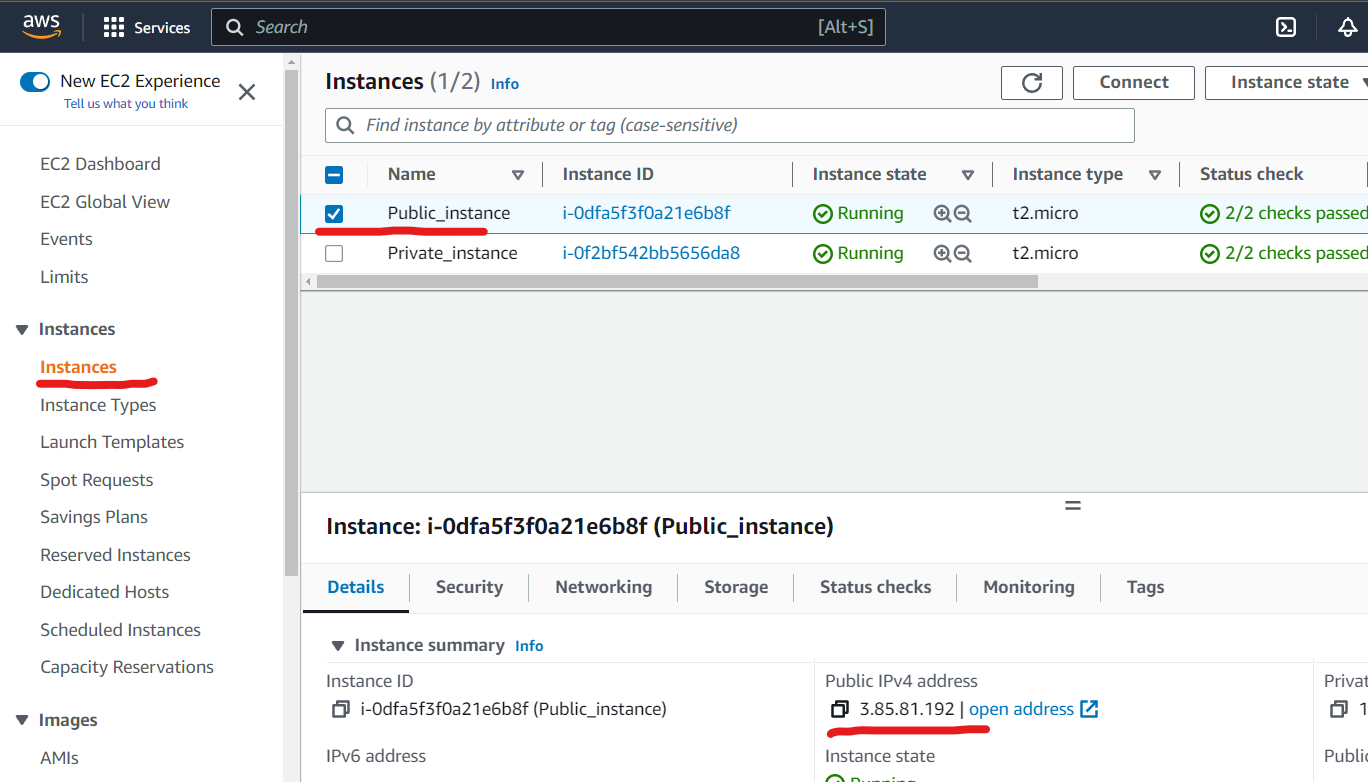
Connect to Public instance with the help of the key pair pem file and the public instances Public IP address.

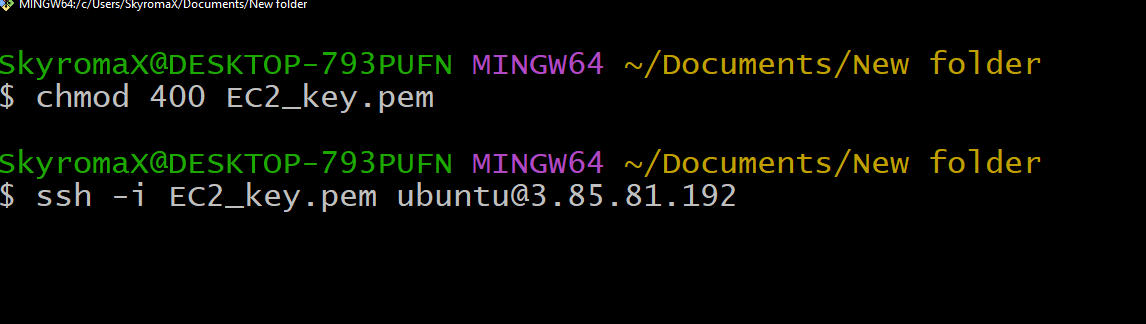
Browser to the folder in your computer/laptop where you have recently downloaded the key pair .pem file and open a Git bash terminal in the same directory.

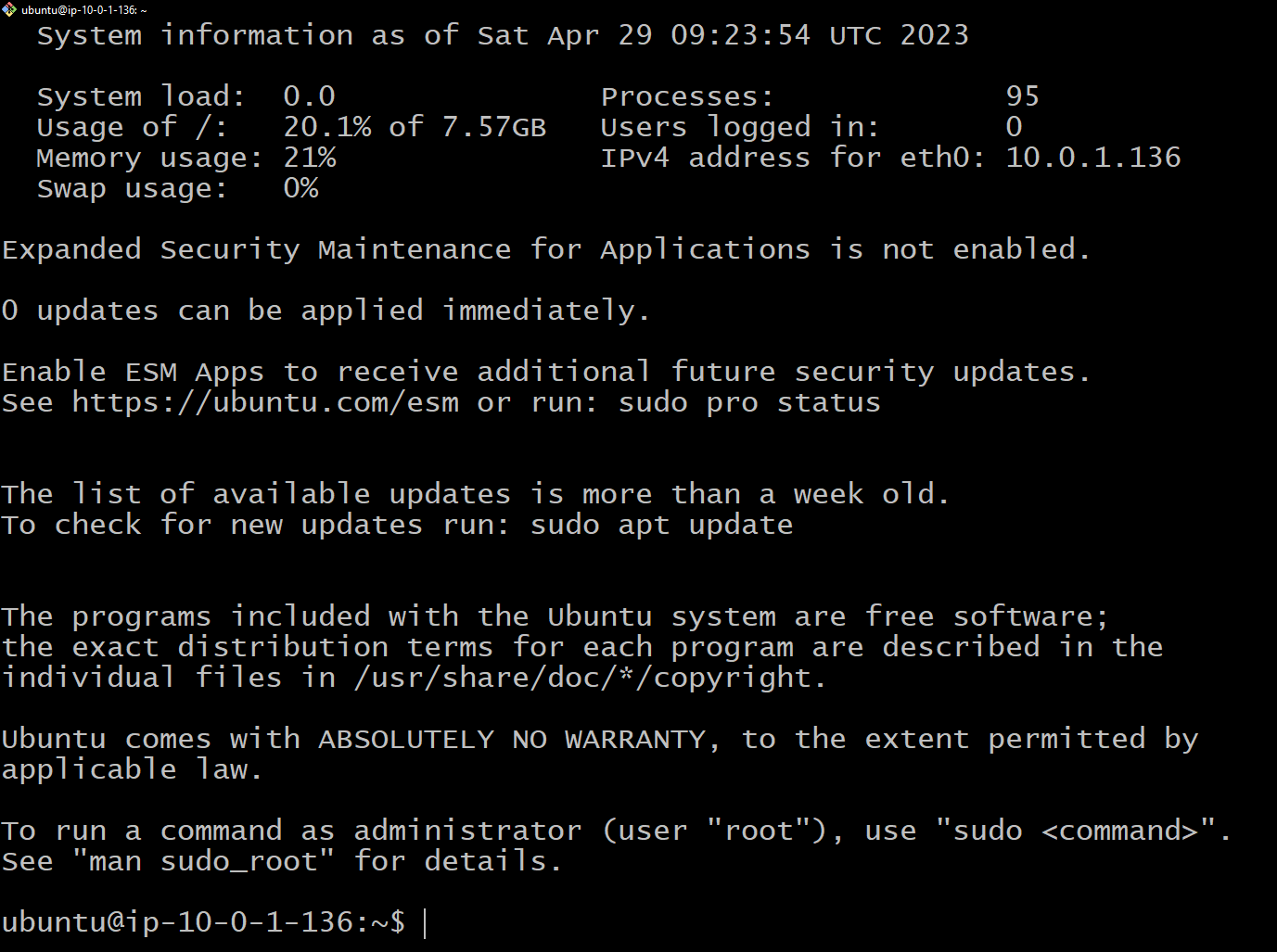


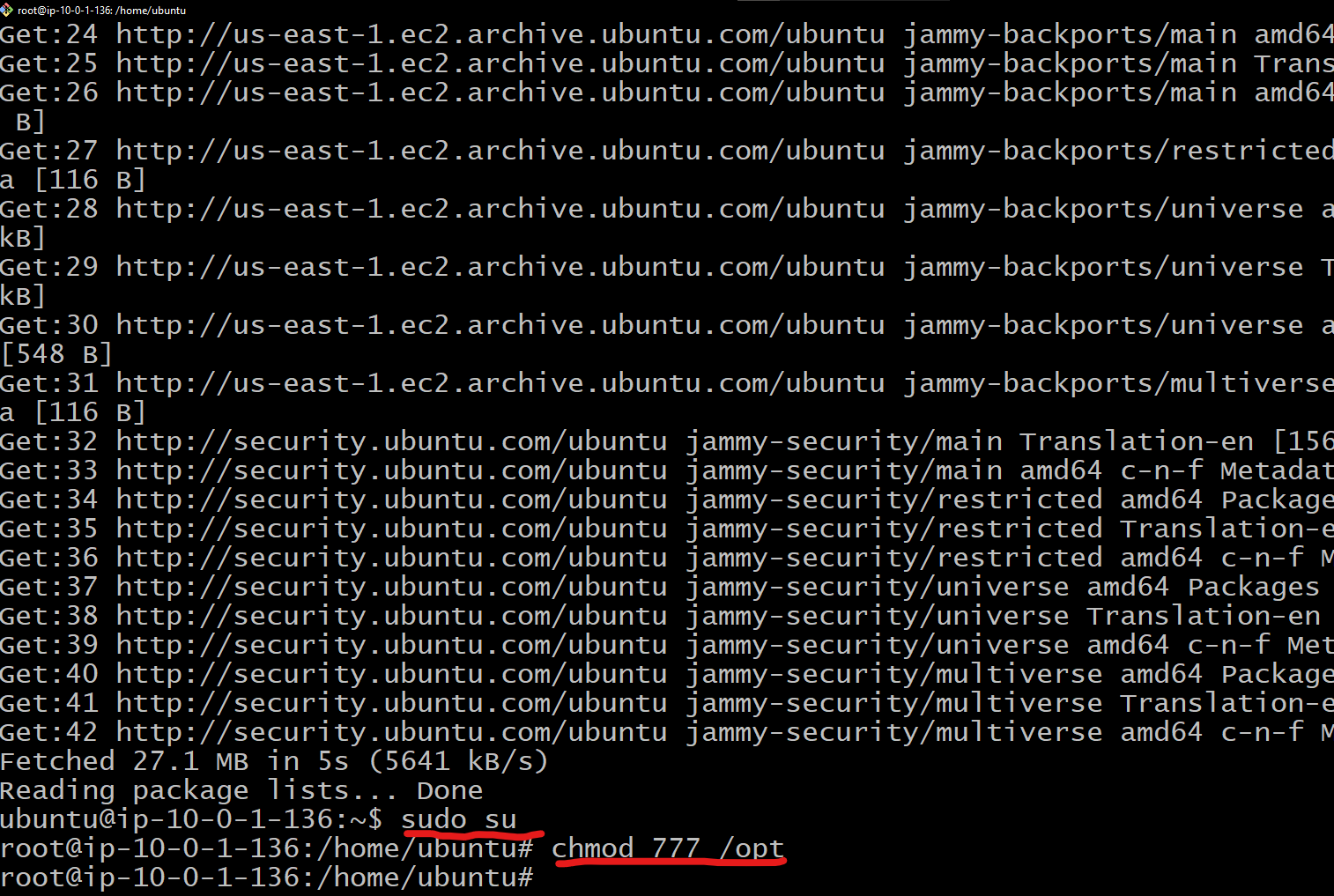
Open a git bash terminal as shown above  
Enter the below command to change permissions on the .pem file.  
>chmod 400 EC2\_key.pem

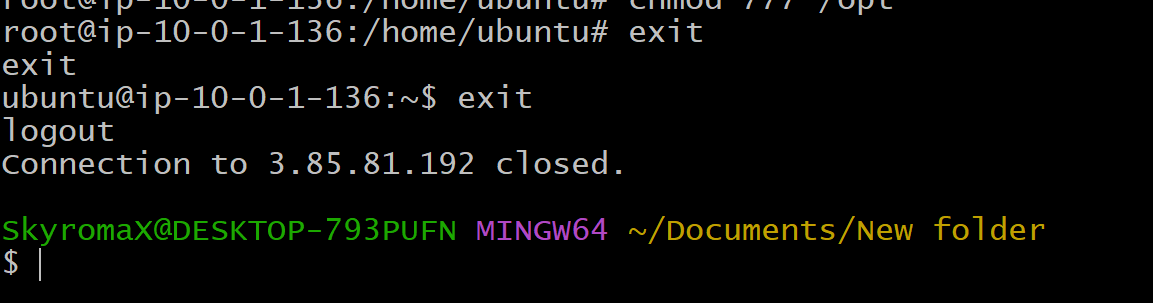
Copy the public IP address from the EC2 management console as shown below.

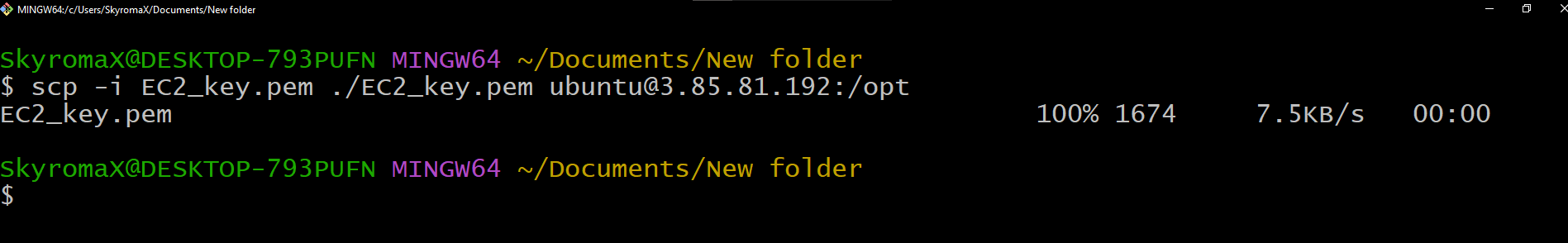


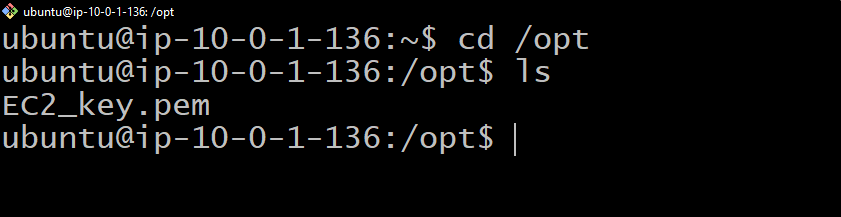
Enter the bellow command to connect to the public Instance.  
>ssh -i EC2\_key.pem [ubuntu@3.85.81.192](mailto:ubuntu@3.85.81.192)  


Type “yes” when the prompt asks for a yes/no question.  


Enter the mandatory command  
>Sudu apt-get update  
Switch to the super user by entering the below command.  
>sudo su  
Enter the below command to grant read,write, execute permissions to all users in opt folder  
>chmod 777 /opt  


Exit from super user mode.  
>exit  
exit the public instance but entering the below command  
>exit  


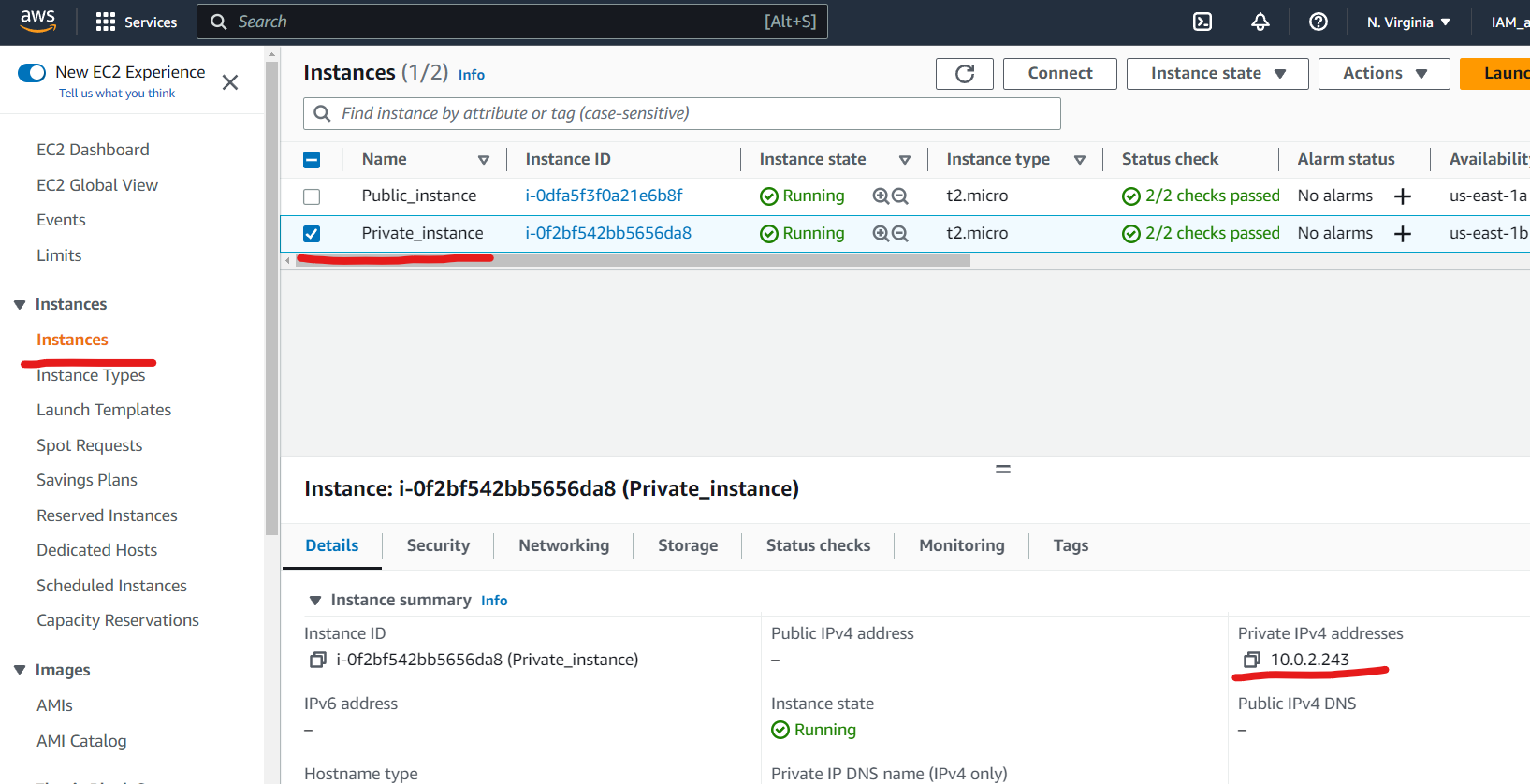
Now to copy the .pem file from the local machine into the public instance.  
to copy the file enter the below command.  
>scp -I EC2.pem ./EC2.pem ubuntu@3.85.81.192:/opt  


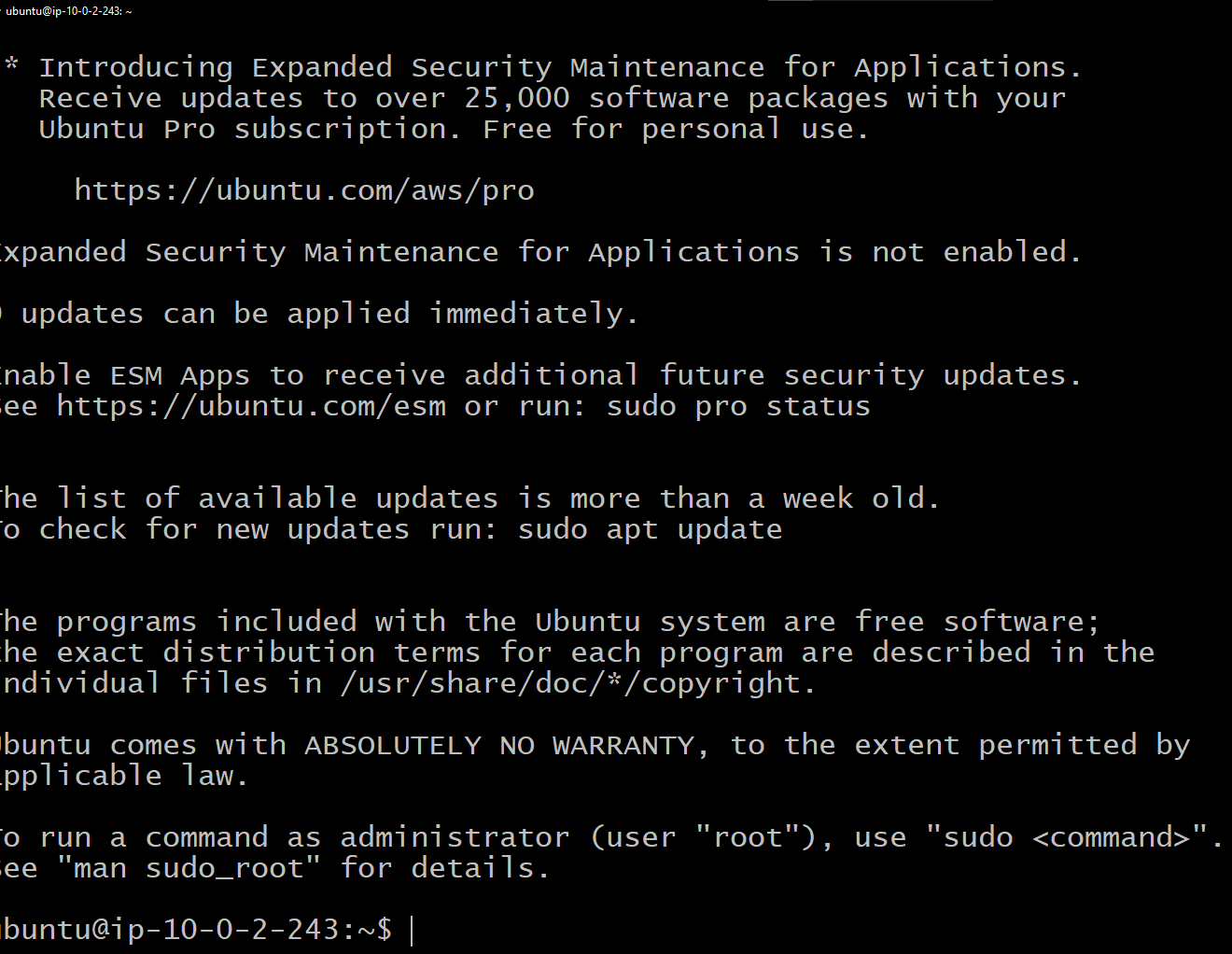
Reconnect back to the Public instance by entering the below command  
>ssh -i EC2\_key.pem [ubuntu@3.85.81.192](mailto:ubuntu@3.85.81.192)  
check whether you have successfully copied the .pem file by changing directory to the /opt folder  


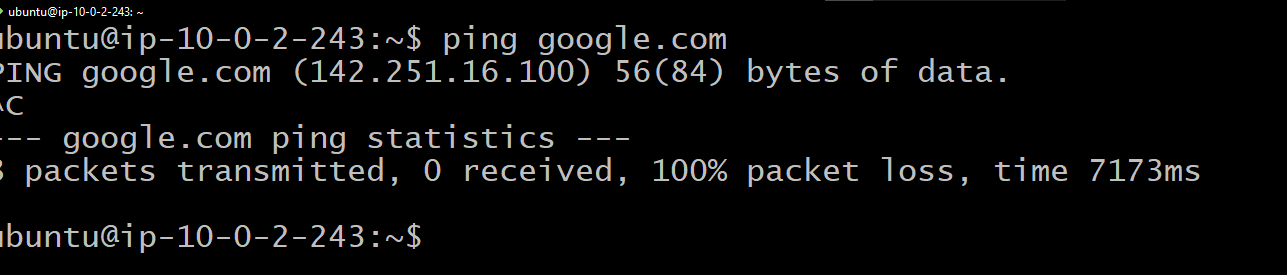
If the EC2\_key.pem file reflects , you have successfully copied the .pem file.

Change the permissions of the .pem file in the opt folder by entering the following command  
> chmod 400 EC2\_key.pem

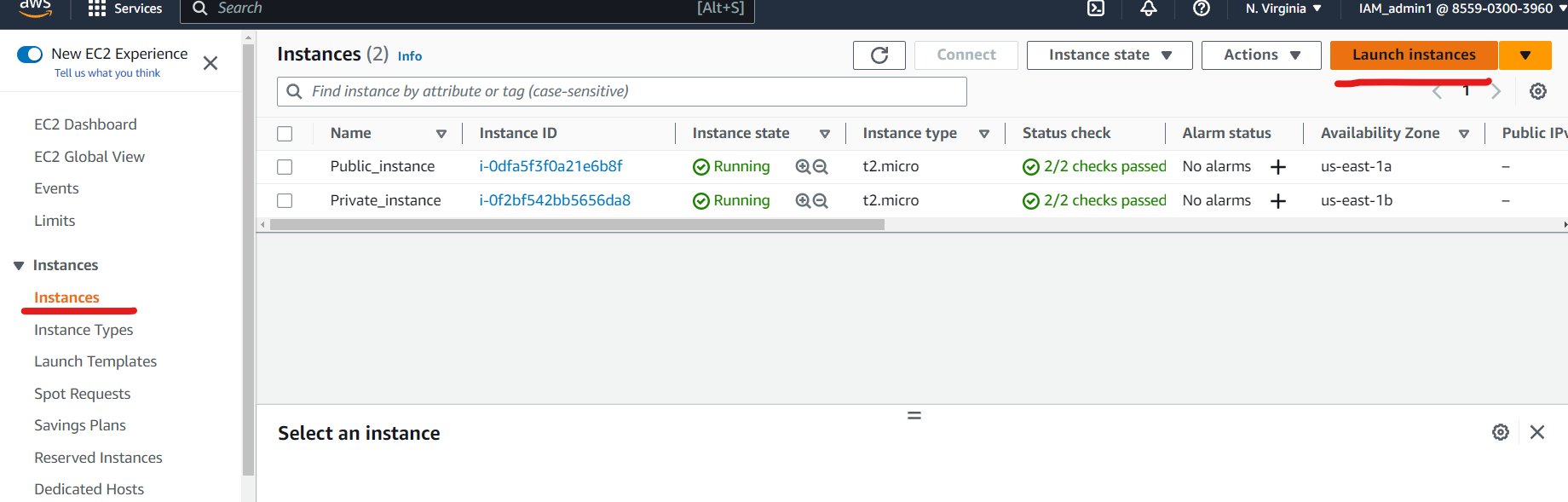
Optain the Private IP address of the Private instance from the EC2 menu as shown below.

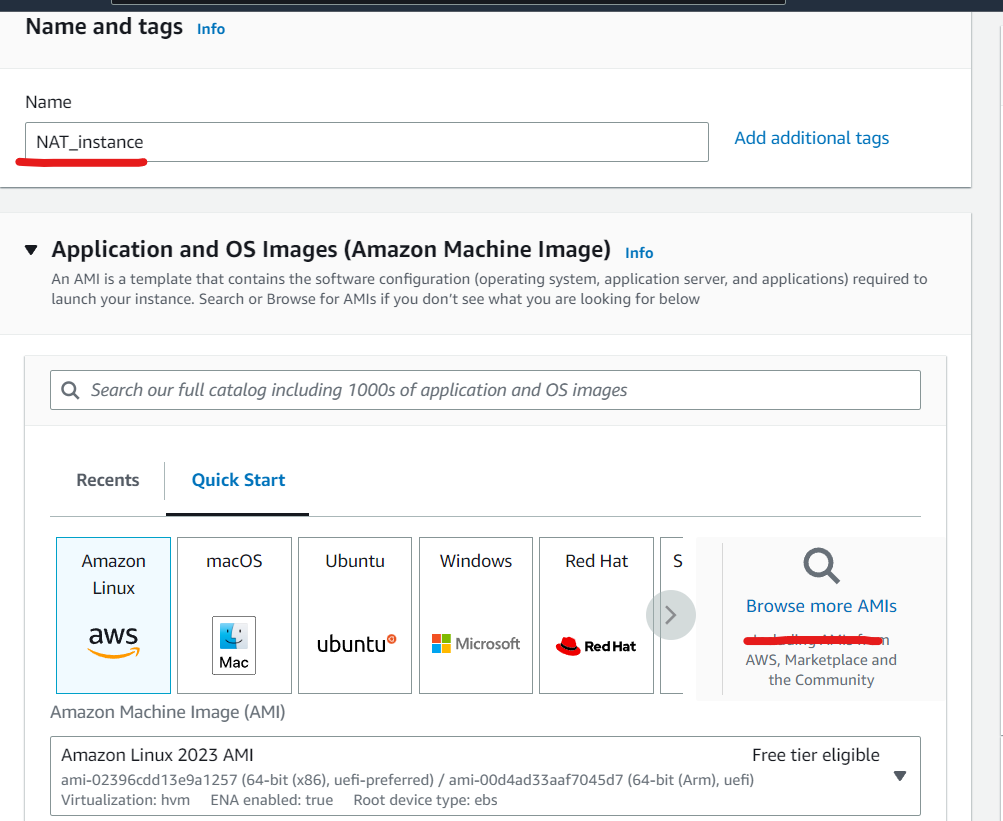
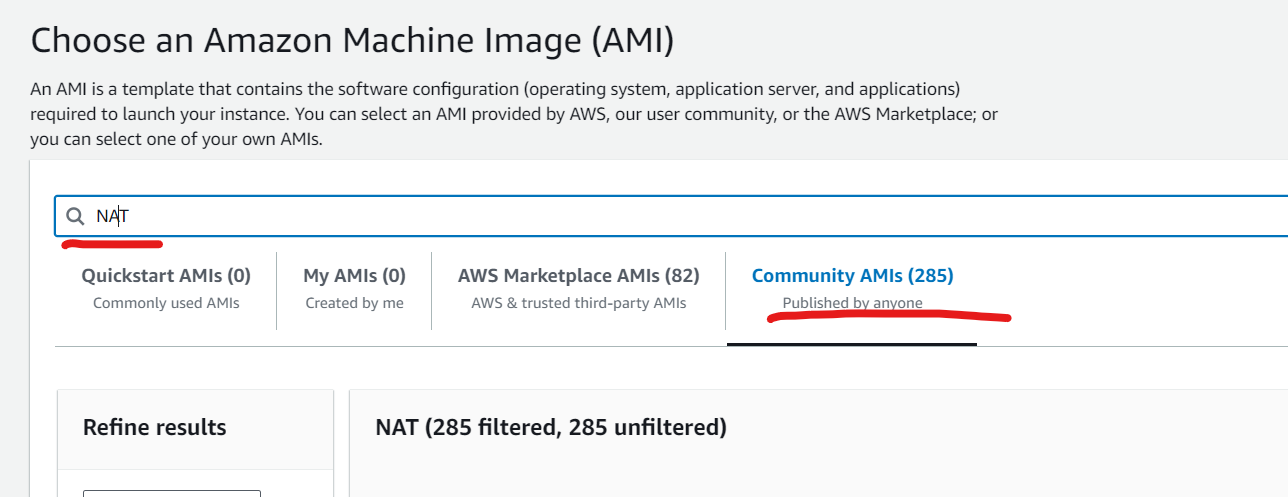


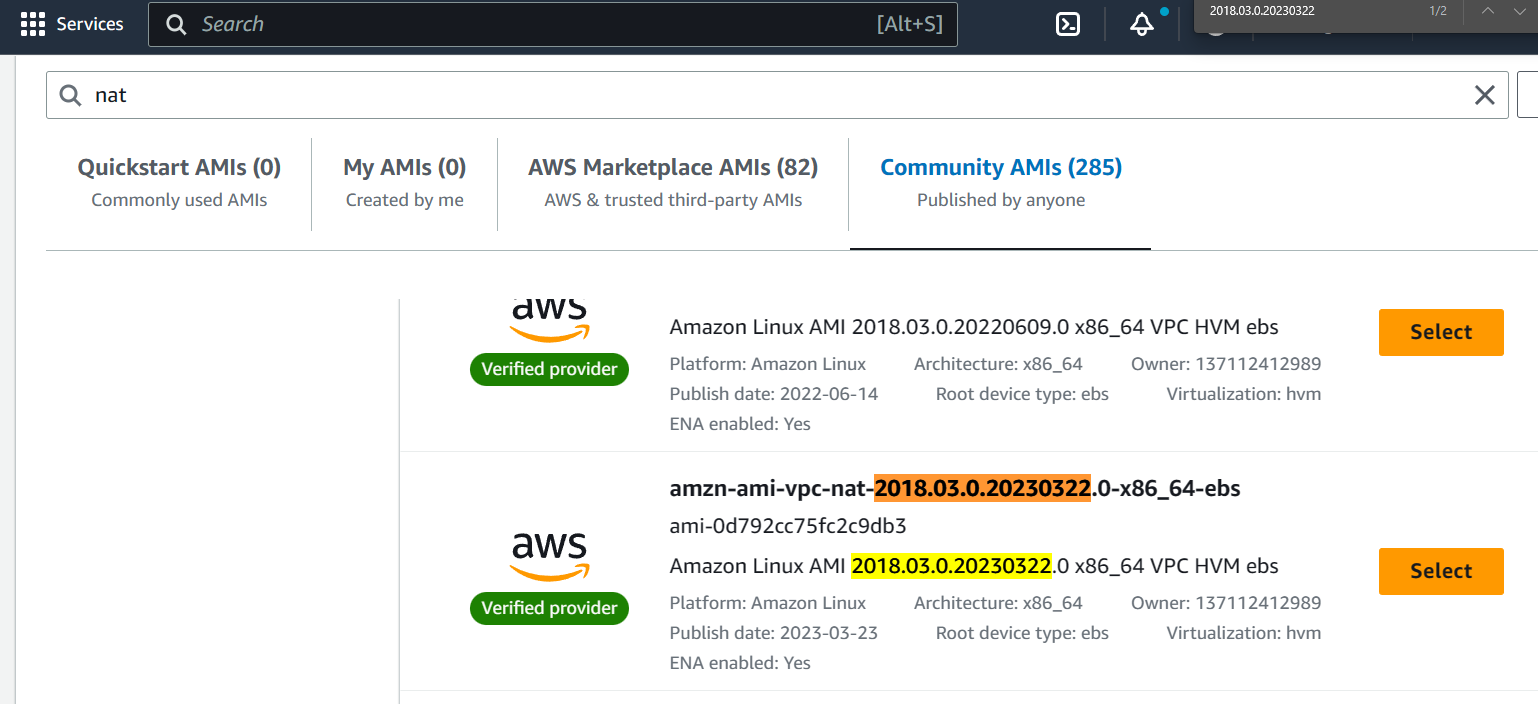
Enter the following command to connect to the private instance from the public instance.  
>ssh -I EC2\_key.pem ubuntu@ 10.0.2.243  
type “yes” when prompted with a yes/no option  


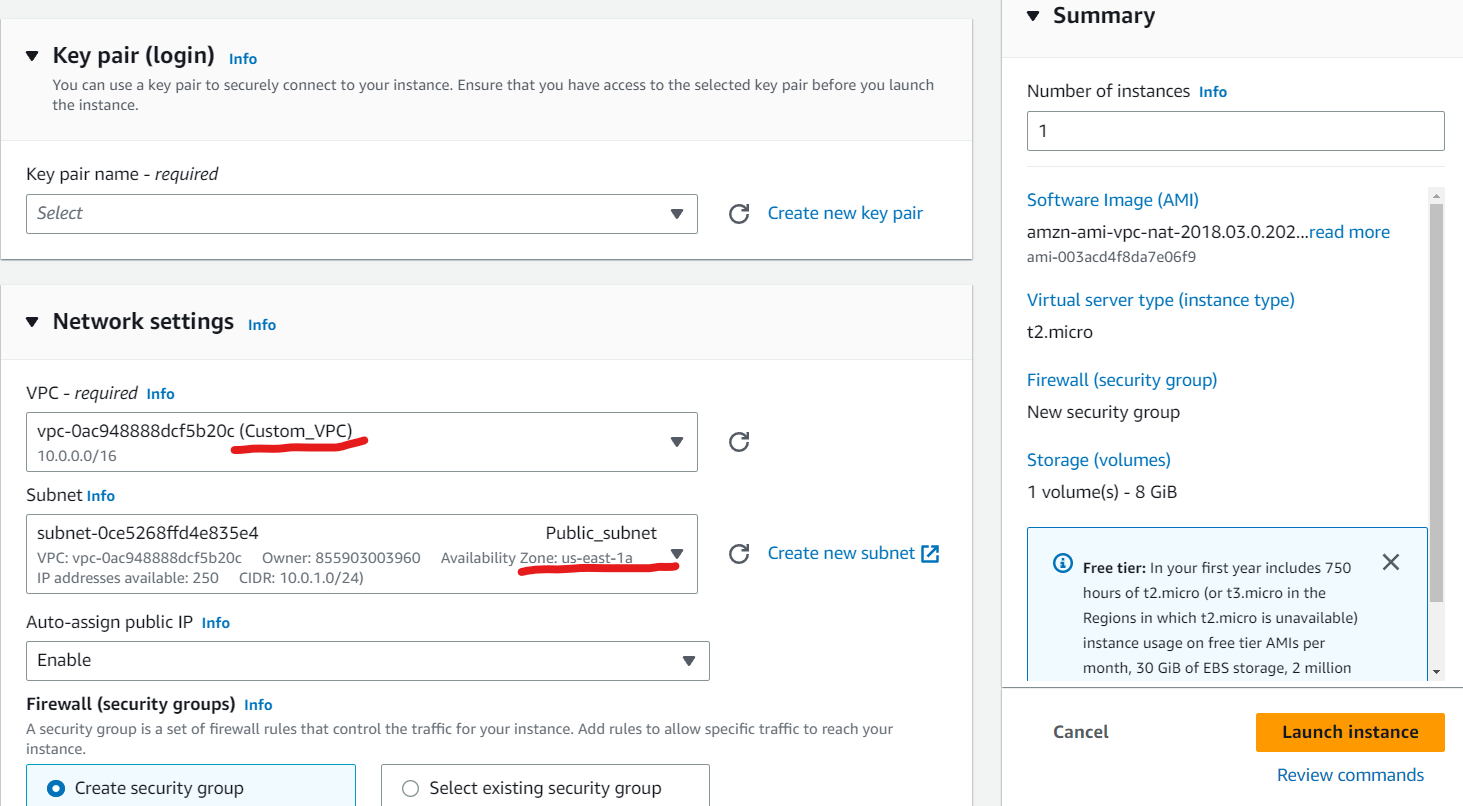


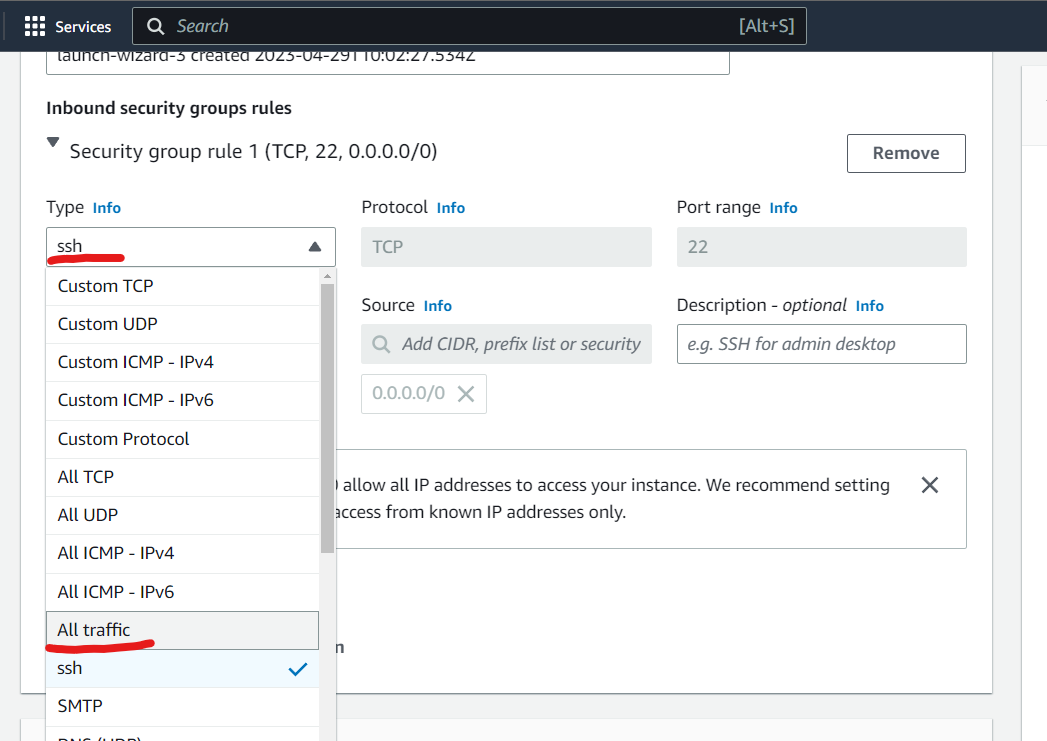
You would not be able to connect internet as the Private instance does not have access to the internet.  
you need to setup a NAT instance in the network to access internet in the private instance.

Visit the **EC2** menu and click on **Launch Instance**   


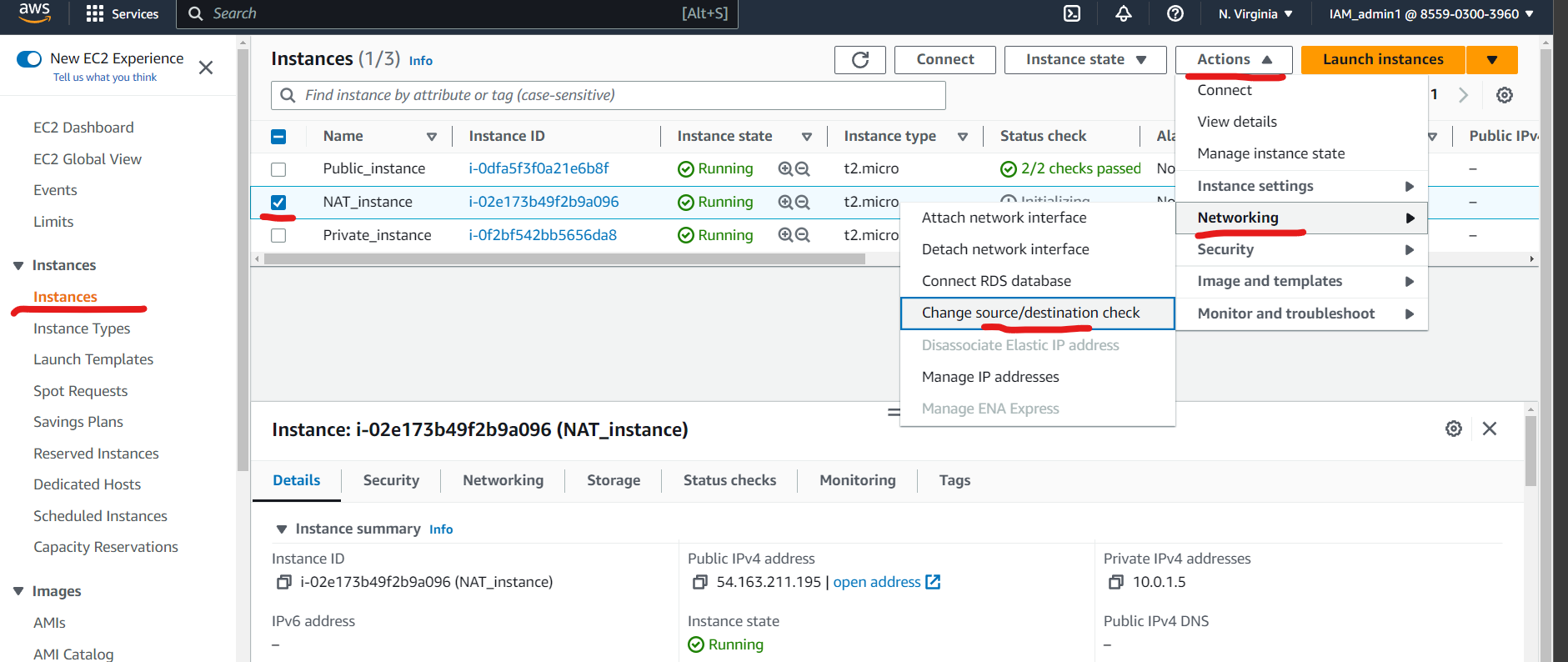
Enter name as NAT instance  
click on **browse more AMI** as shown below  
  
switch to **community AMI** and search for “NAT” as shown below  


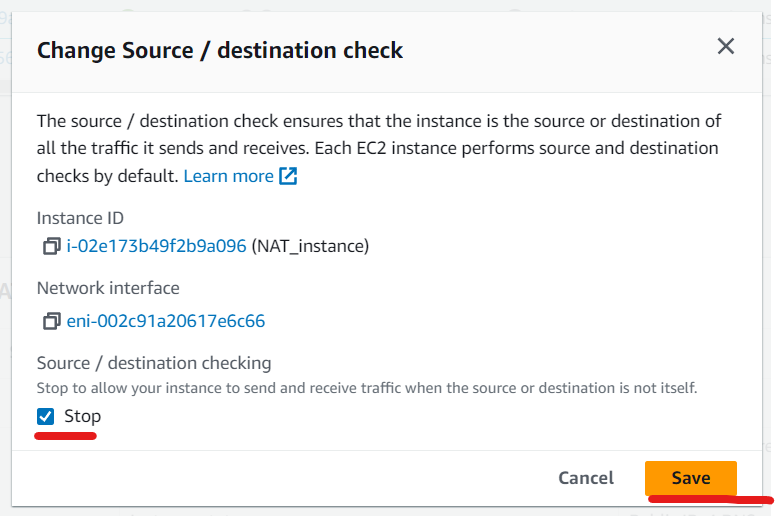
Scroll to the AMI 2018.03.0.20230322 and click on **Select** button as shown below  


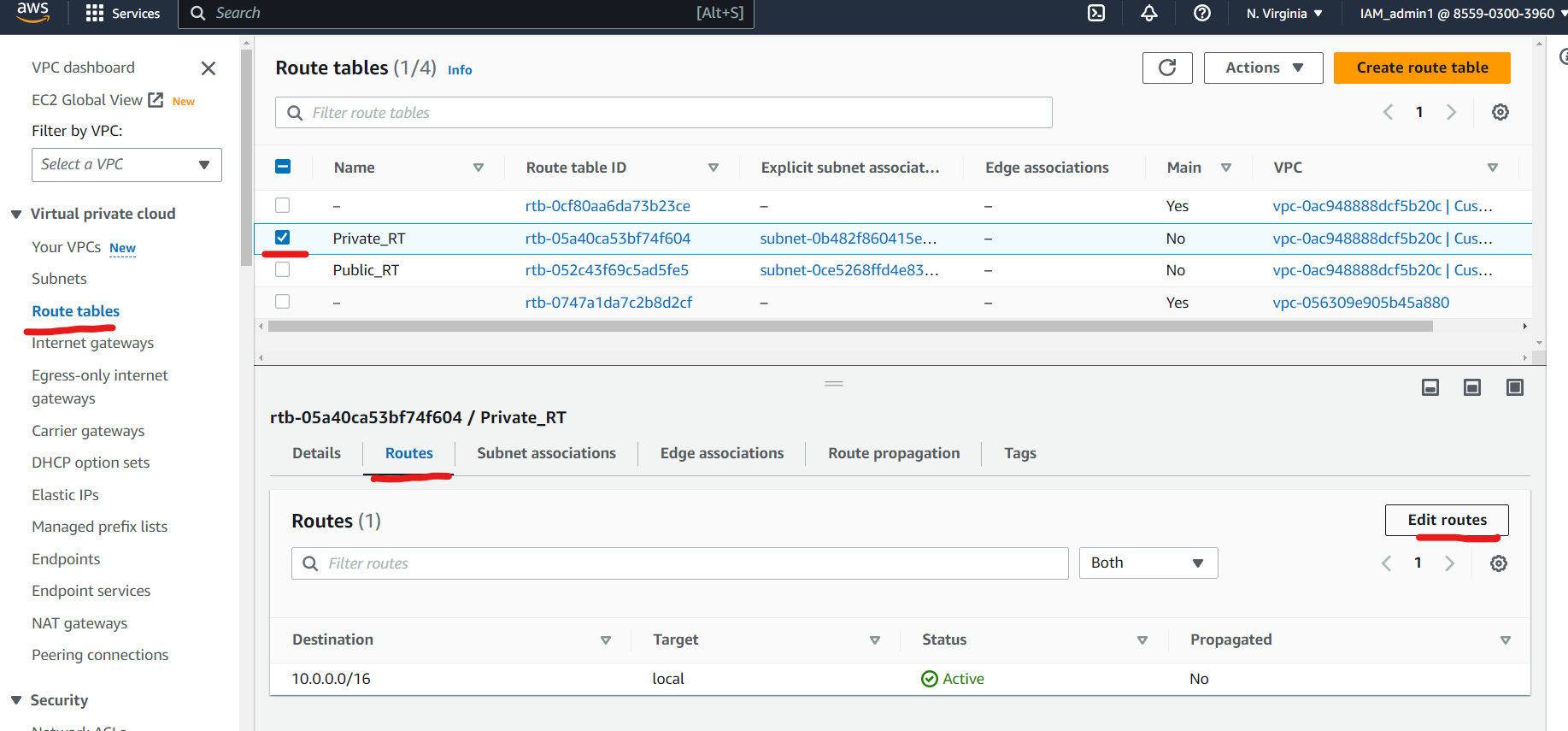
In the create instance page scroll down to the Network setting and click on edit.  
Select the Custom VPC and select the Public Subnet as shown below.  
select key pair as proceed without a keypair  


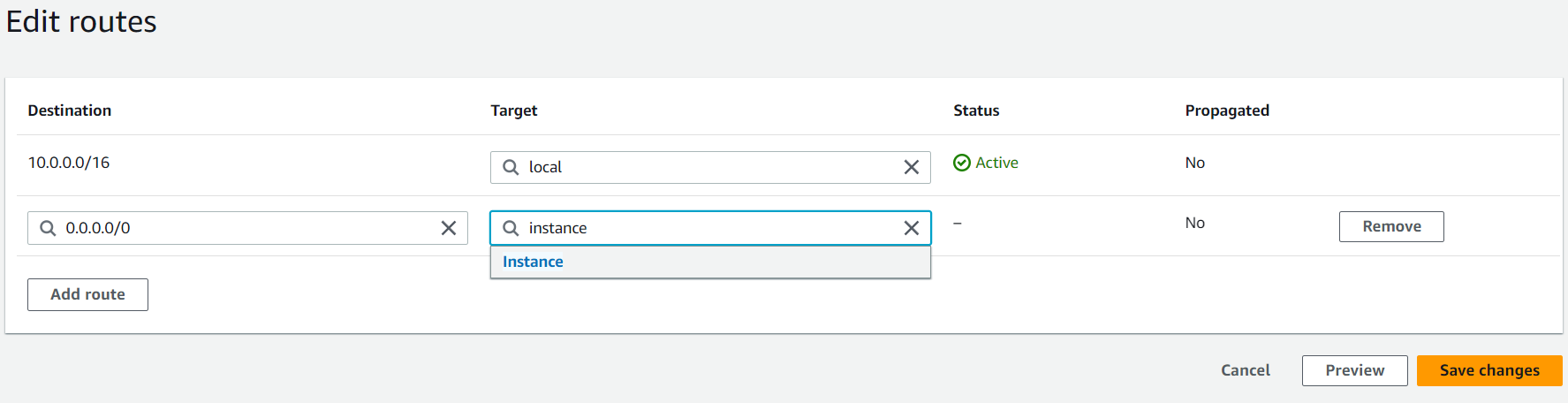
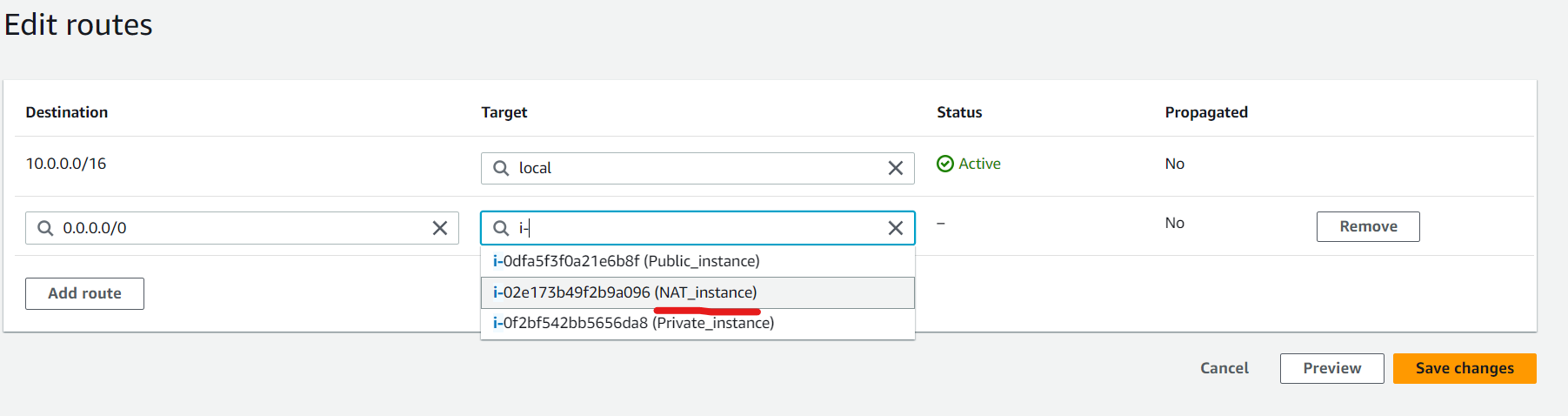
Change SSH to All traffic in security group setting as shown below  


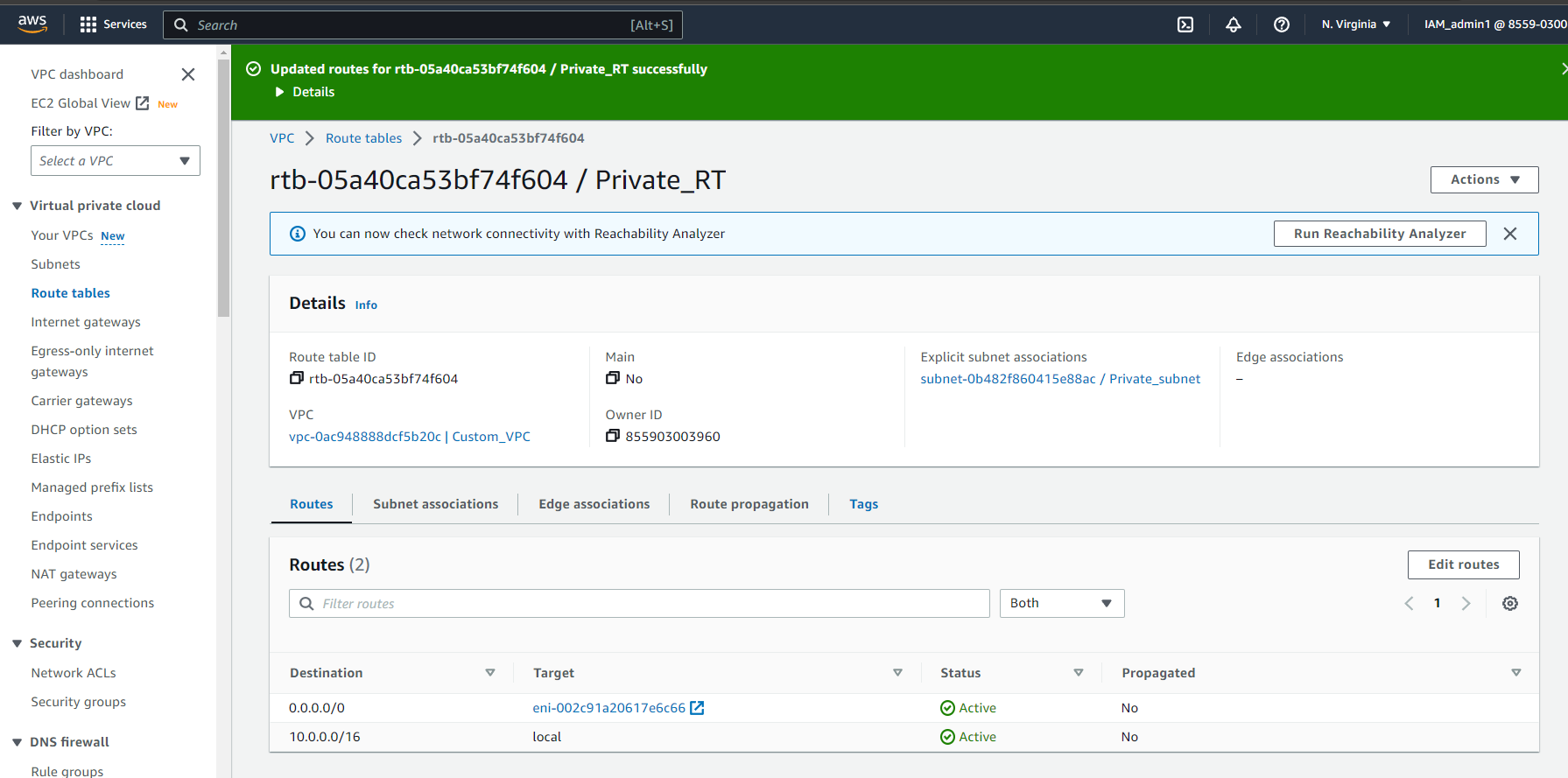
Scroll down and click on launch instance.

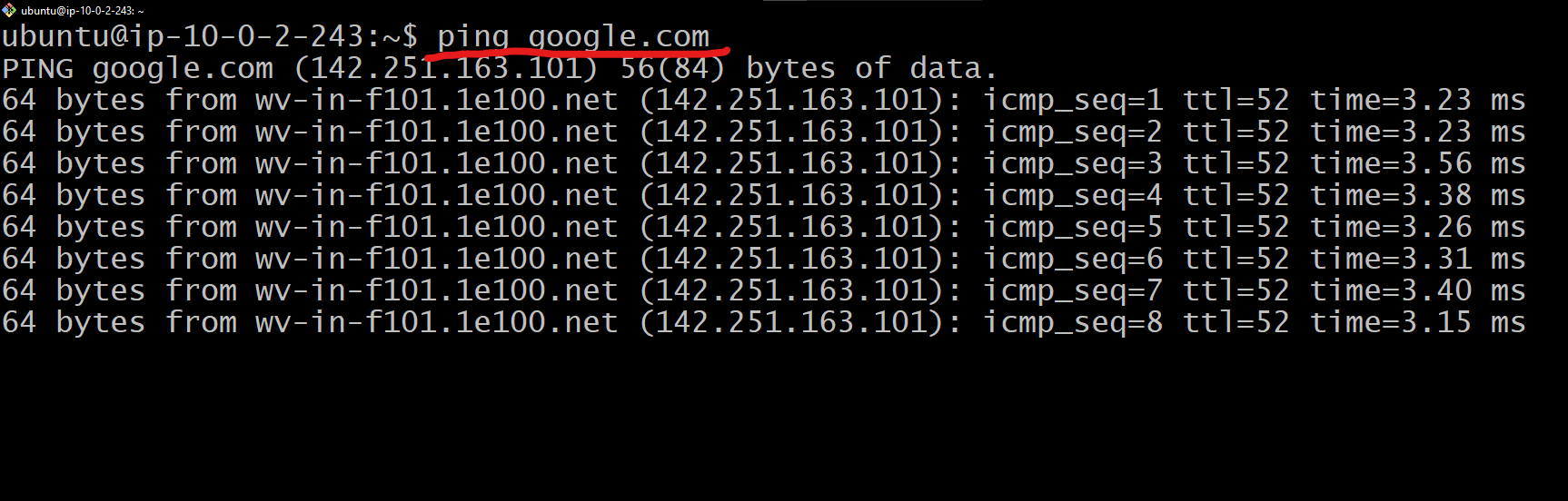
After the NAT instance has been successfully created   
select the NAT instance in the Instance menue  
click on **Actions** button  
select **networking** option.  
select **change Source/destination check** as shown below  


Tick mark the “Stop” checkbox and click on **save** as shown below.  


Visit the **Route Tables** menu in VPC serices.  
Select the **private route** table.  
Switch to **route** tab and click on **edit route** as shown below>  


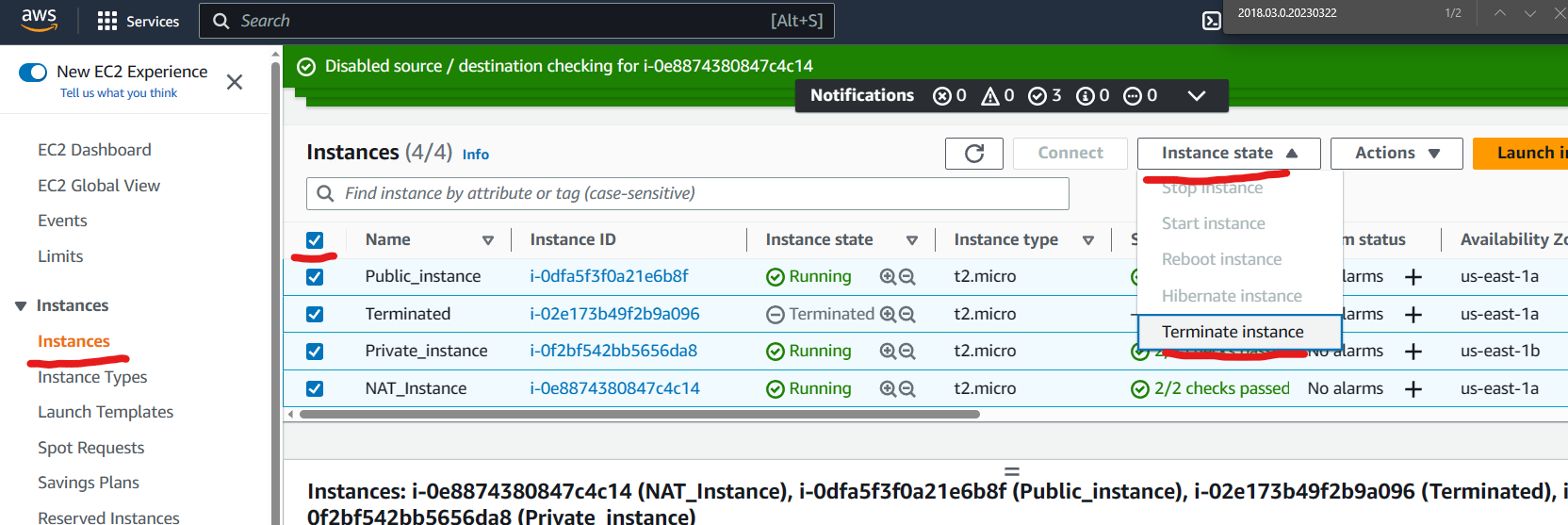
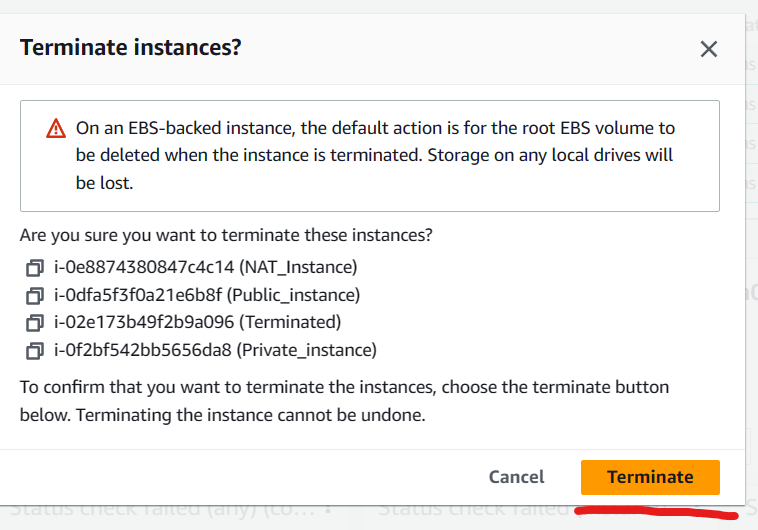
Click on **Add route**   
Set Destination = 0.0.0.0/0  
Search “instance” in the target field and click on “instance” as shown below.  
  
Select the NAT instance that you have recently created as shown below  
  
Click on **Save changes**

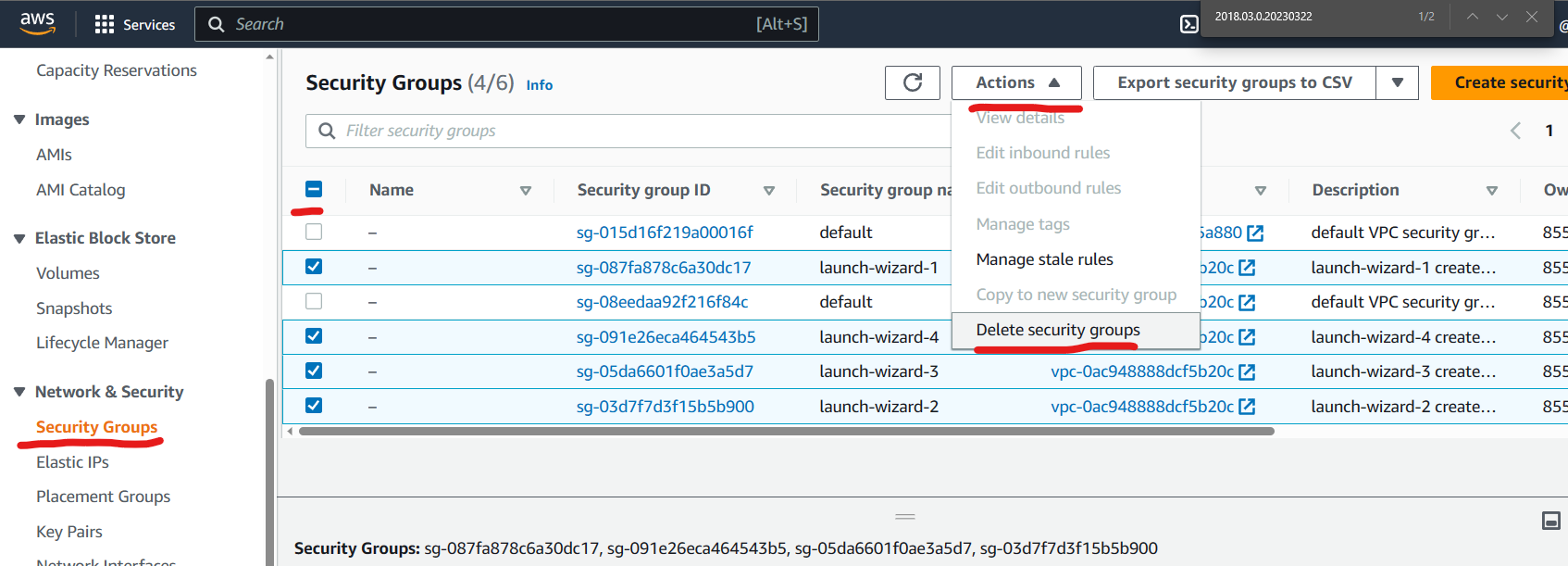


Go back to the terminal and try to ping google.com with the below command.  
>ping google.com  


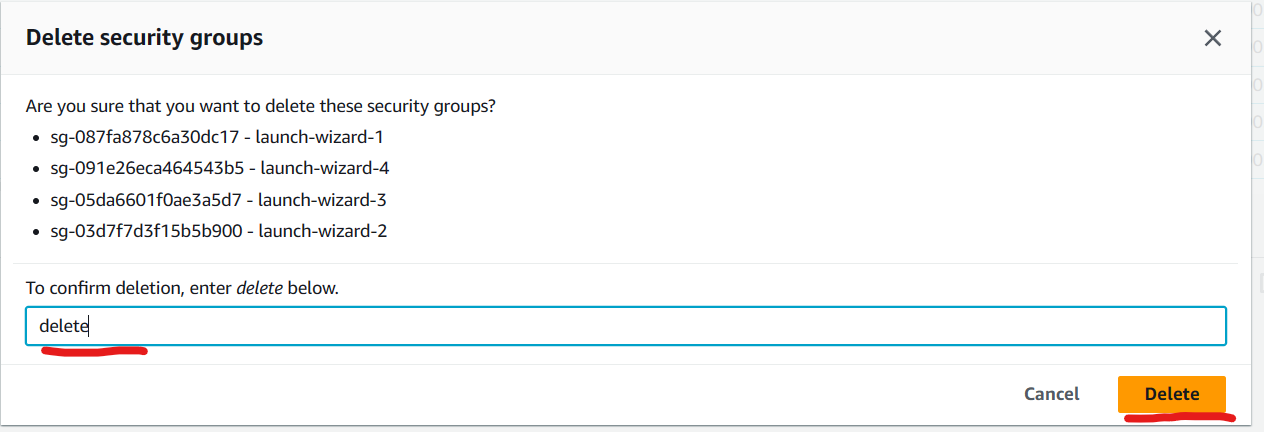
You have successfully connected to the internet in a private instance through NAT instance setup.

Clean up  
Break the ping out by clicking Ctrl+C on the keyboard to end the Ping google output.  
Exit the the instance session by entering the command “exit” and close the terminal.

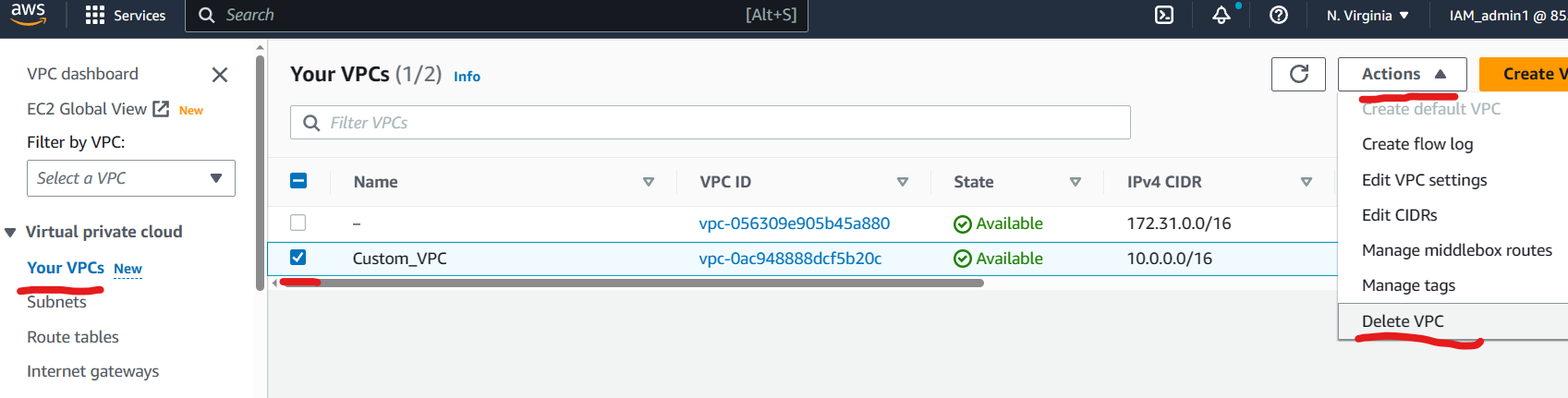
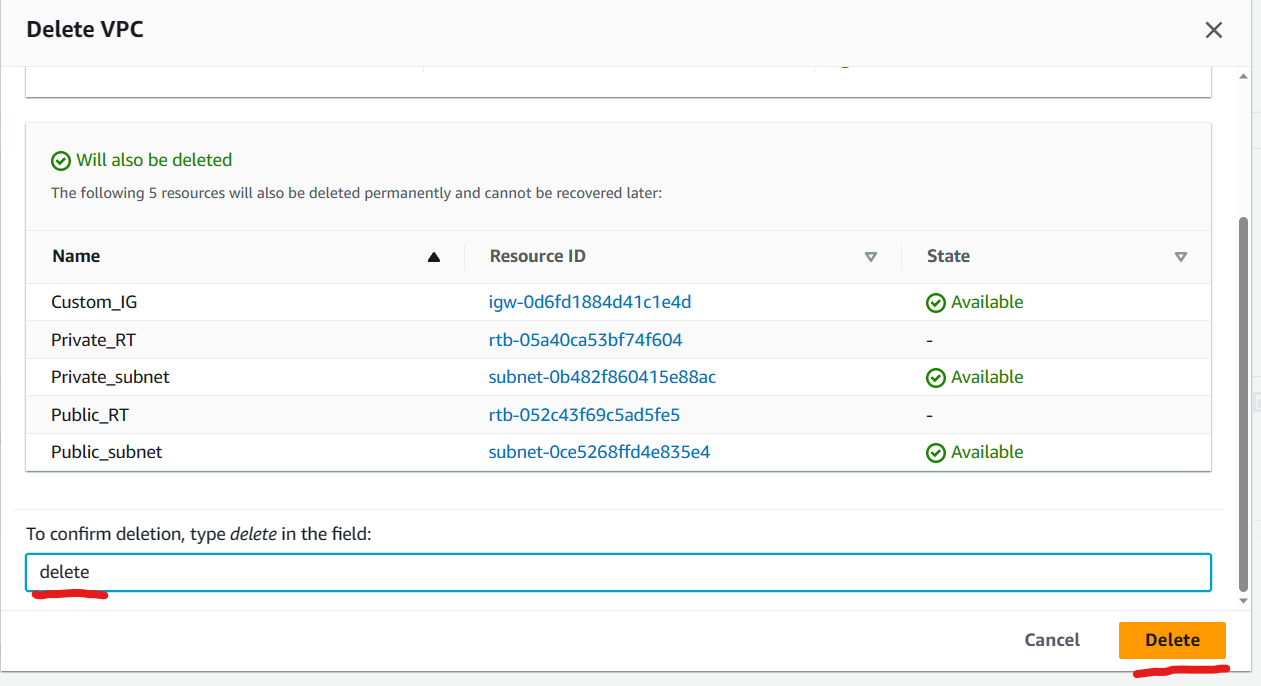
Terminate the instance that you have created above.  
visit the **EC2** Menu   
Select all the instance you have created for the above project  
click on **Instance state** button and select **Terminate instance** option as shown below.  
  
  
Click on terminate button as shown above.

Clear the Security Groups  
visit the **Security Groups** menu  
Select the Security groups which were created in association to the Instance created earlier.  


Unselect the Default Security group as they are dependent on other essential services.

  
Type “delete” and click on delete as shown above.

Delete VPC

Visit the **Your VPC** menu  
select your custom VPC and click on **Actions** button  
click on **Delete VPC** as shown below  
  
Type “delete” and click on **Delete** button as shown below  


Visit the **VPC Dashboard** and **EC2 Dashboard** to confirm that you have cleared all the resources that you have used above are deleted/dregistered in the system.

Q. What is an Egress only Internet gateway?  
Ans. A form of AWS (Amazon Web Services) gateway called an egress-only Internet Gateway permits traffic to flow from an Amazon Virtual Private Cloud (VPC) to the internet but forbids inbound traffic from the internet from establishing connections with instances in the VPC. In other words, it protects resources in a VPC from incoming unwanted traffic while providing a secure and controlled method for them to access the internet. When instances within a VPC must connect with services outside the VPC but inbound traffic from the internet is neither necessary or preferred, the egress-only Internet Gateway is commonly employed. If a web server in a VPC needs to make API calls to a third-party service, for instance, it shouldn't be directly