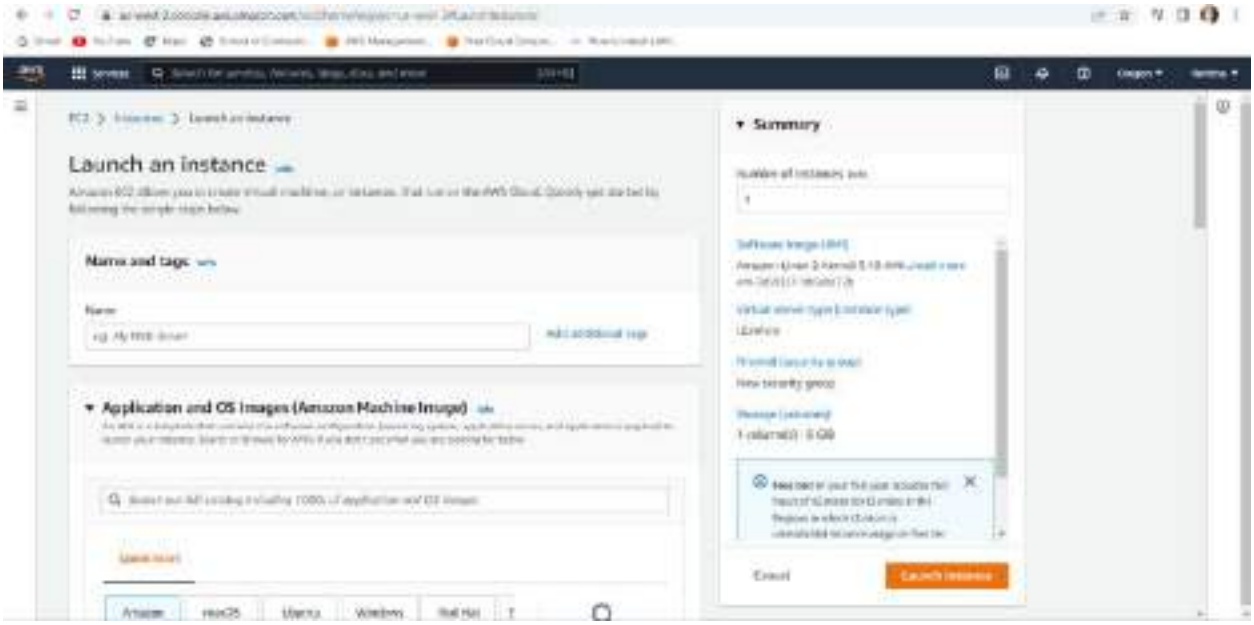
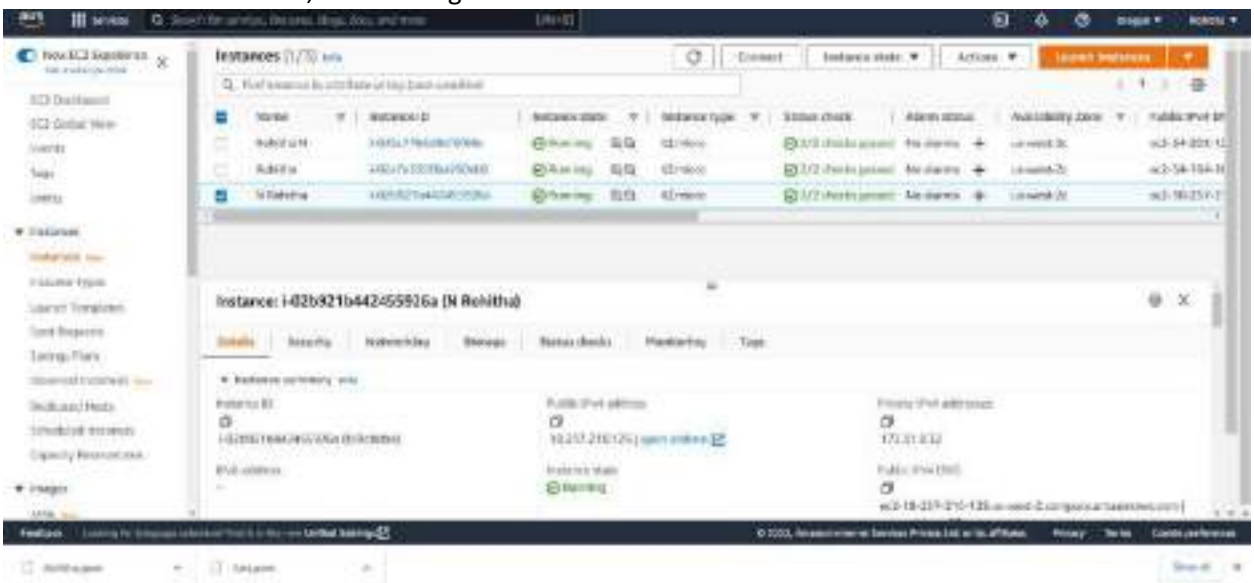
1. Create EC2 instance.

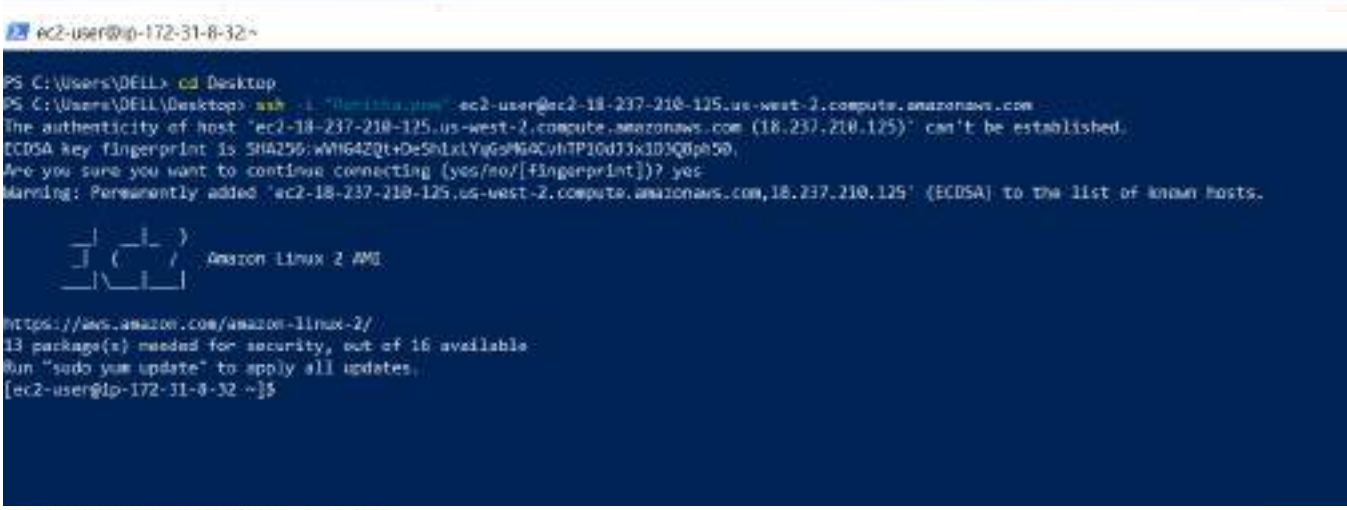
Go to instances and click on Launch instance.

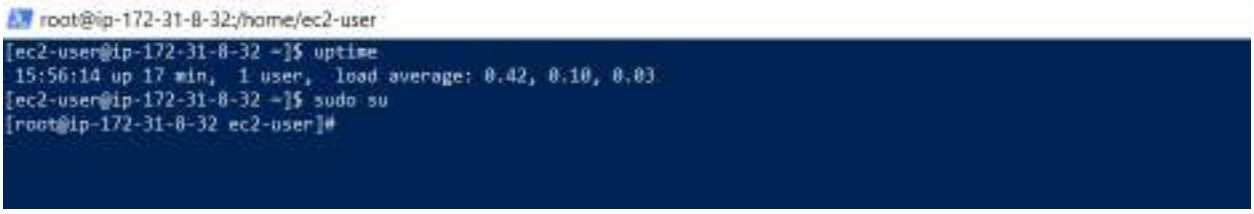


Once instance is created , select and go to connect.



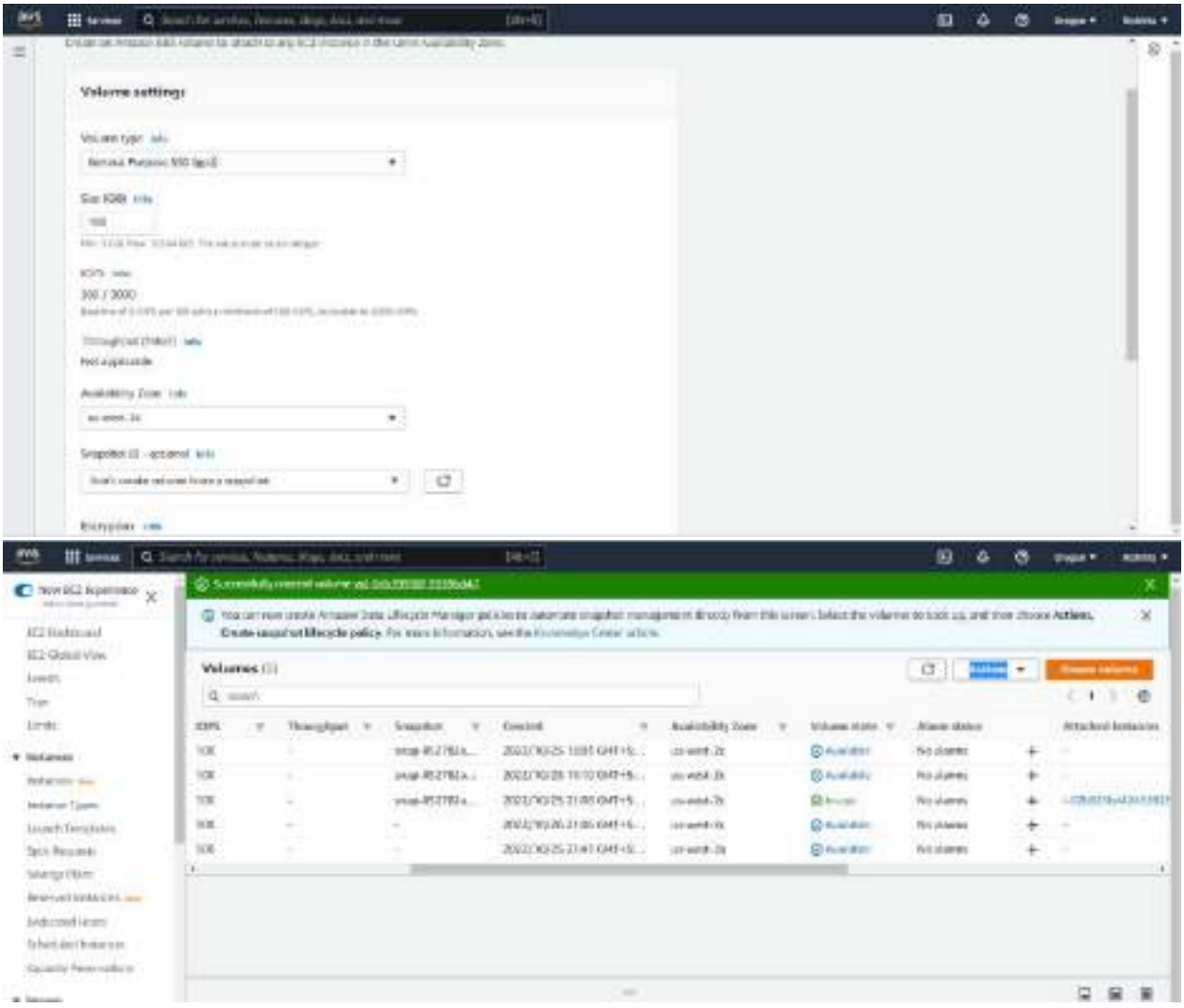
Go to SSH Client tab and copy SSH key and open windows powershell. Paste the SSH key and connect EC2 instance.

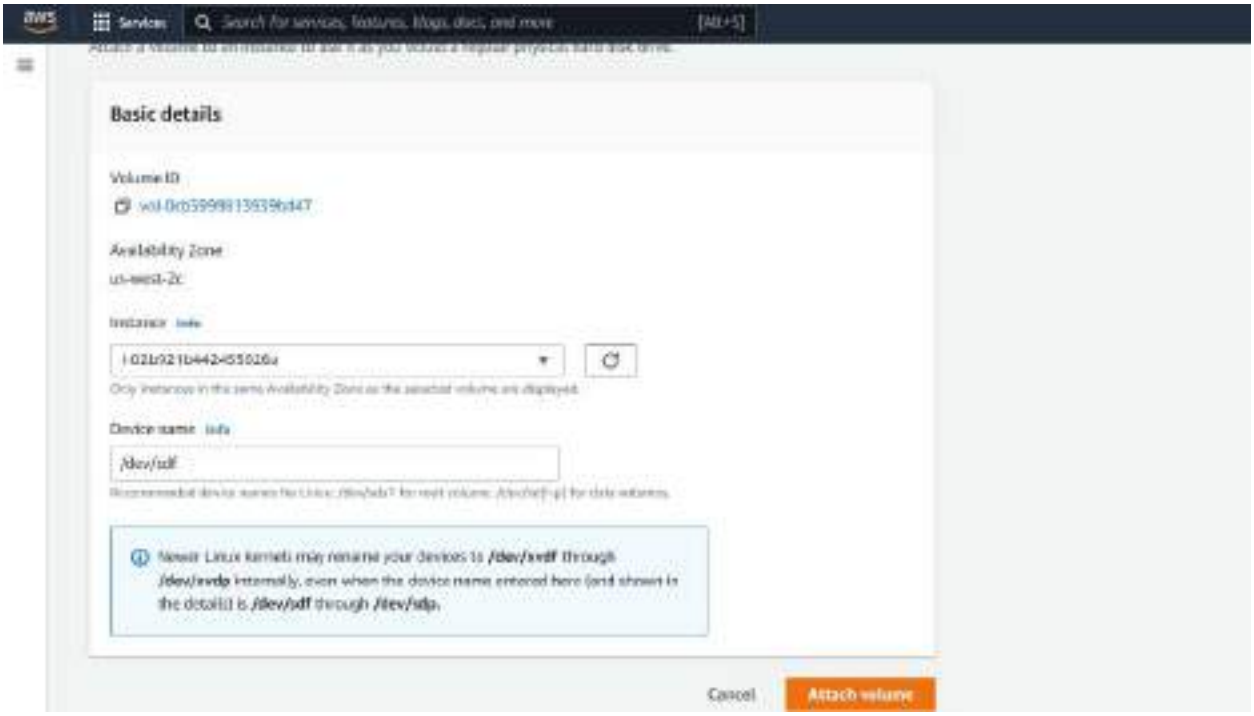




2. Create Elastic block storage:

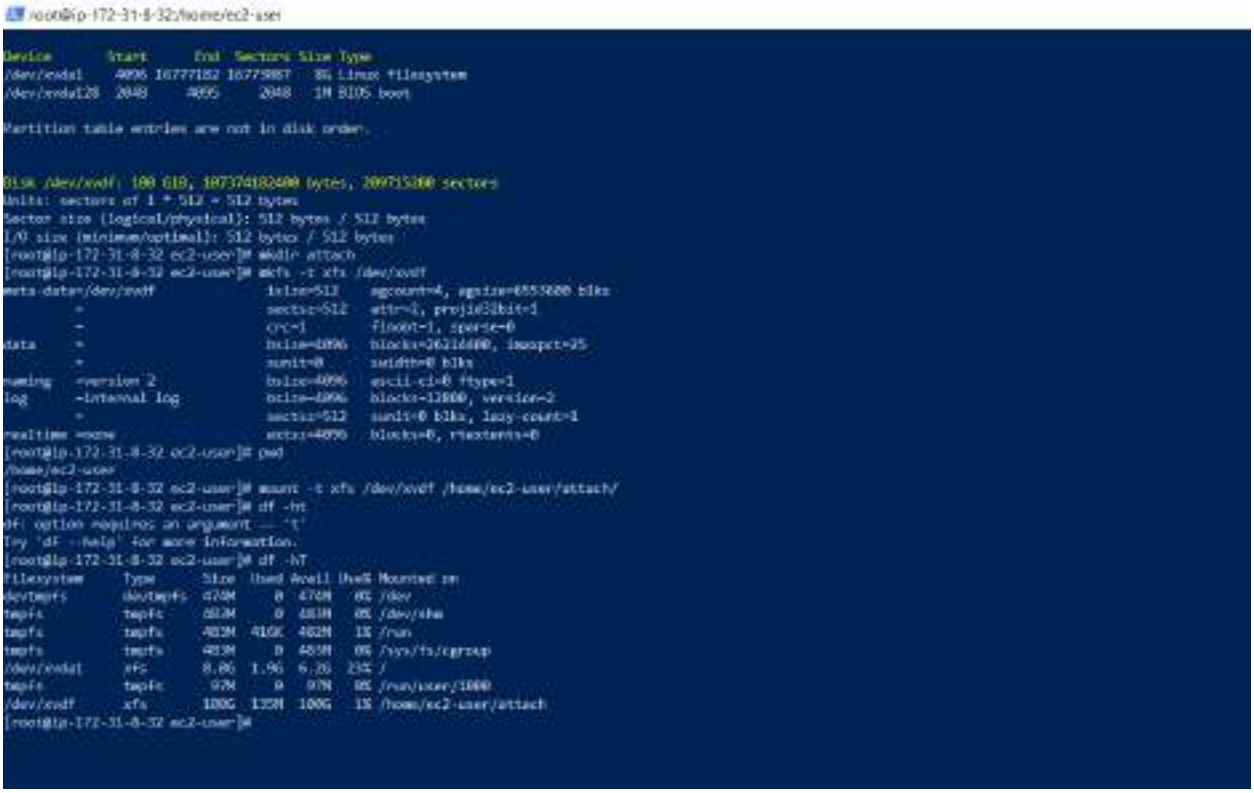
Create a volume storage and attach storage to machine.


Volume settings
Volumes

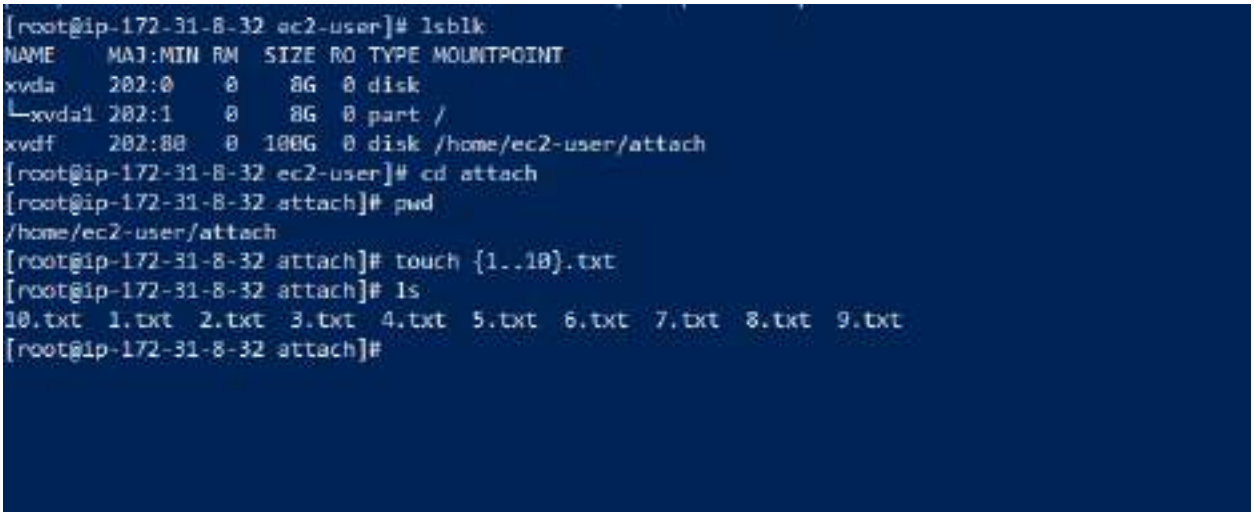


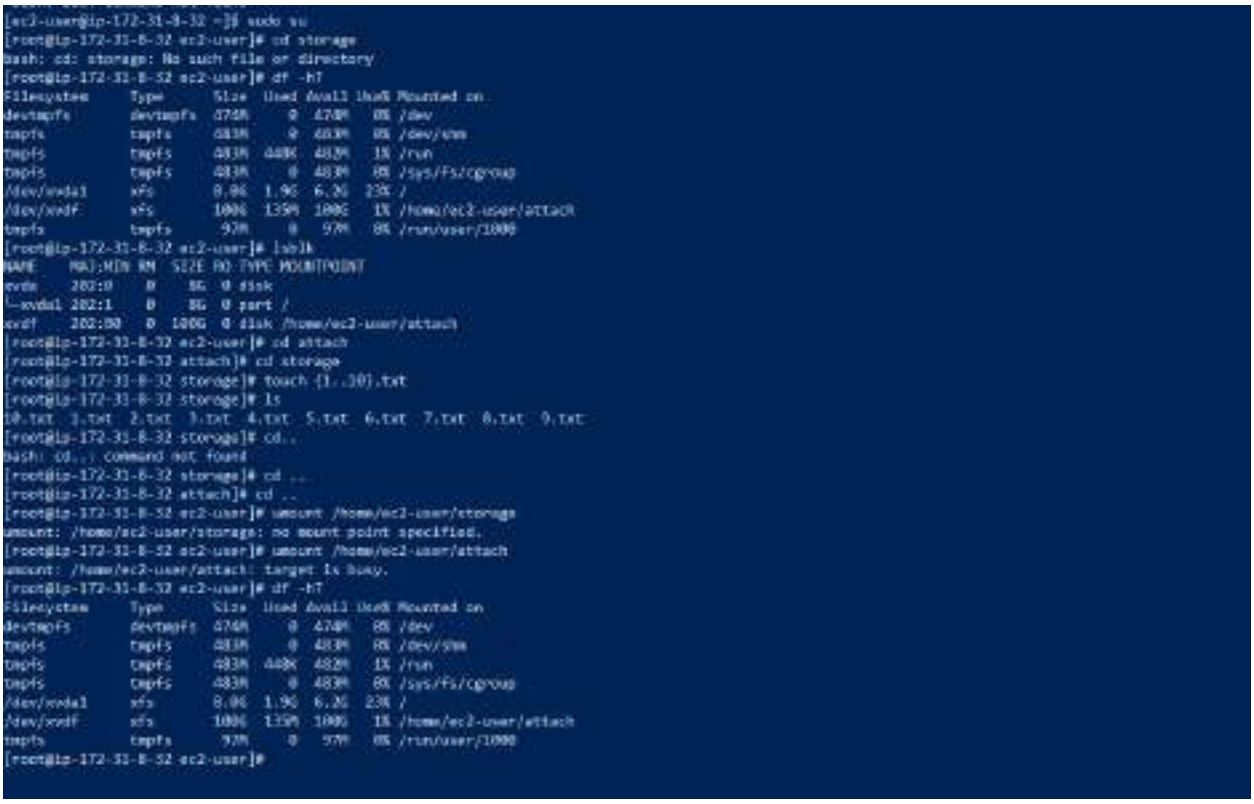
Make a file system and mount the file system.

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-8-32 ~]$ sudo su
[root@ip-172-31-8-32 ec2-user]# lsdlk
bash: lsdlk: command not found
[root@ip-172-31-8-32 ec2-user]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0    8G  0 disk
└─xvda1 202:1    0    8G  0 part /
xvdf    202:80   0  100G  0 disk
[root@ip-172-31-8-32 ec2-user]# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
[root@ip-172-31-8-32 ec2-user]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0    8G  0 disk
└─xvda1 202:1    0    8G  0 part /
xvdf    202:80   0  100G  0 disk
```

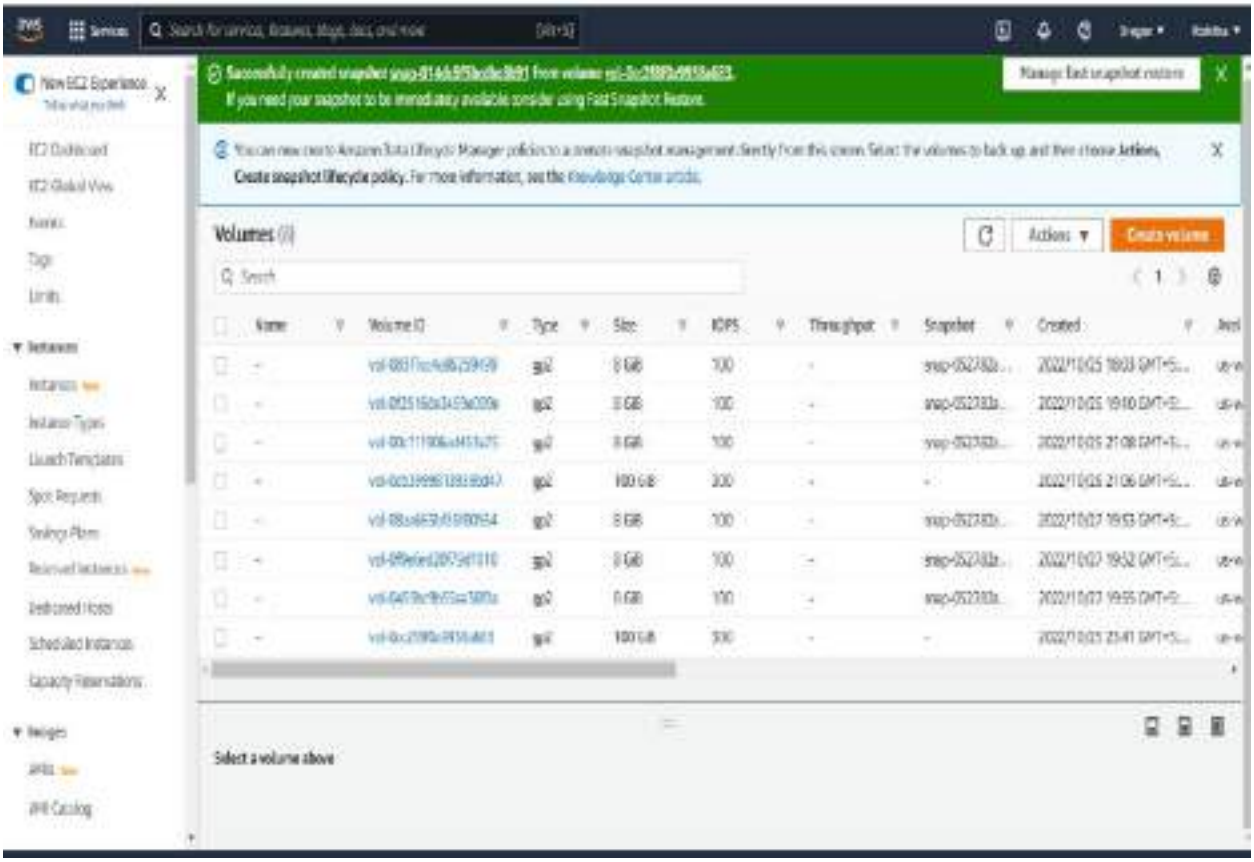


Create some files in the file system and unmount the file system.

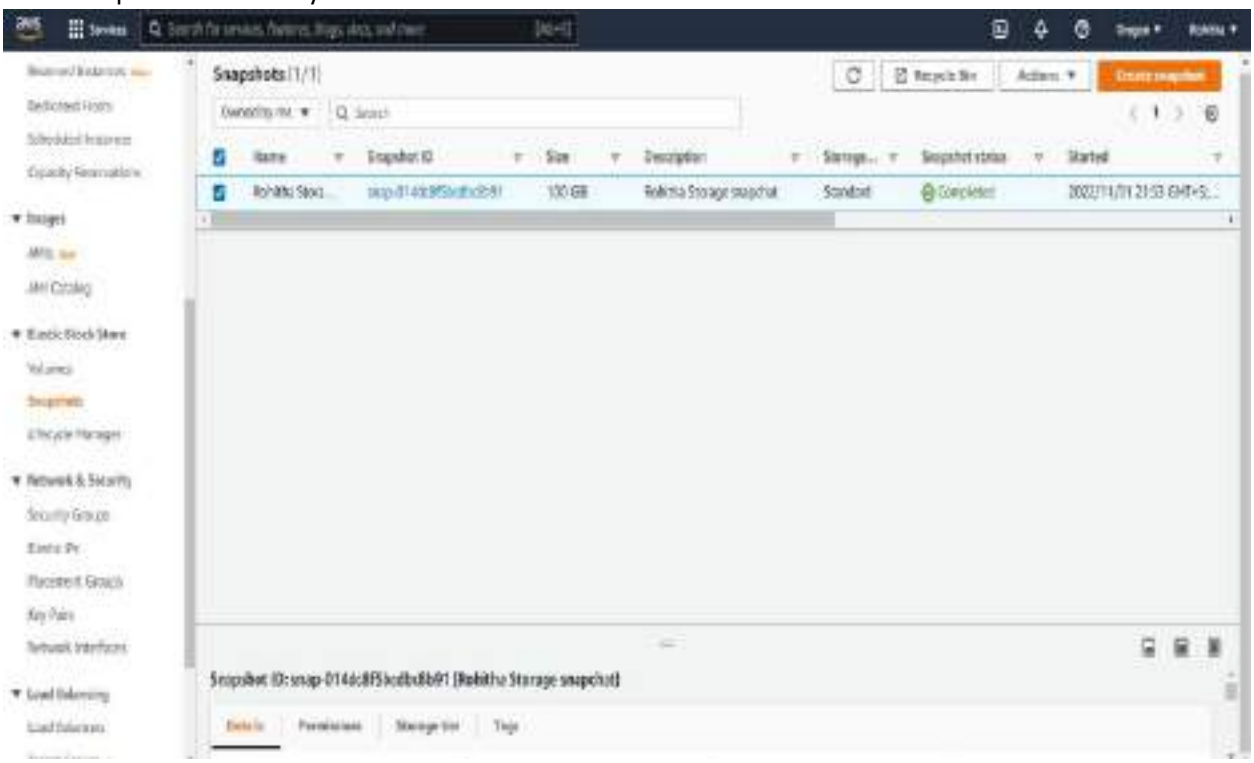




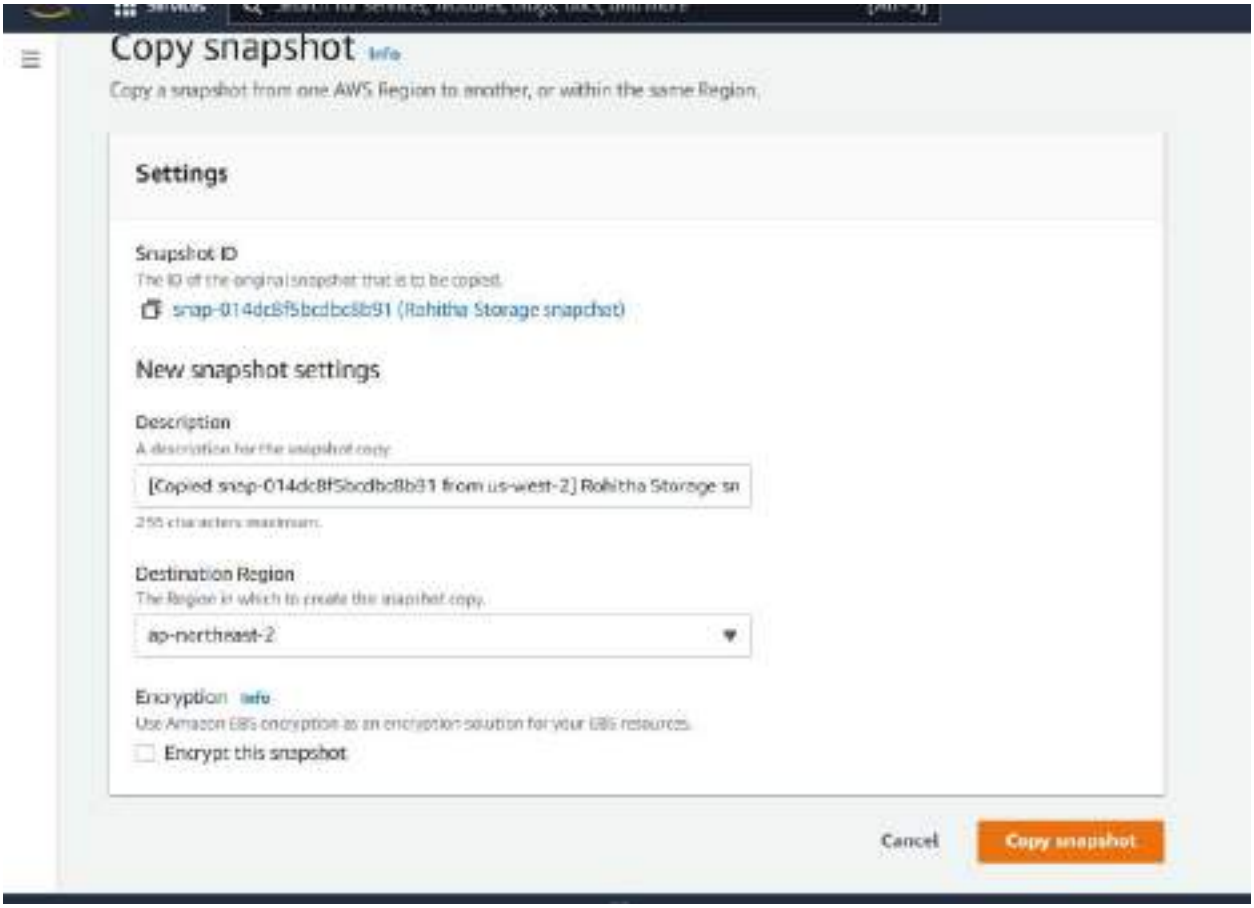
3. **Snapshot Screenshot creation:**
   Create Snapshot from any volume.



Go to snapshots and verify.

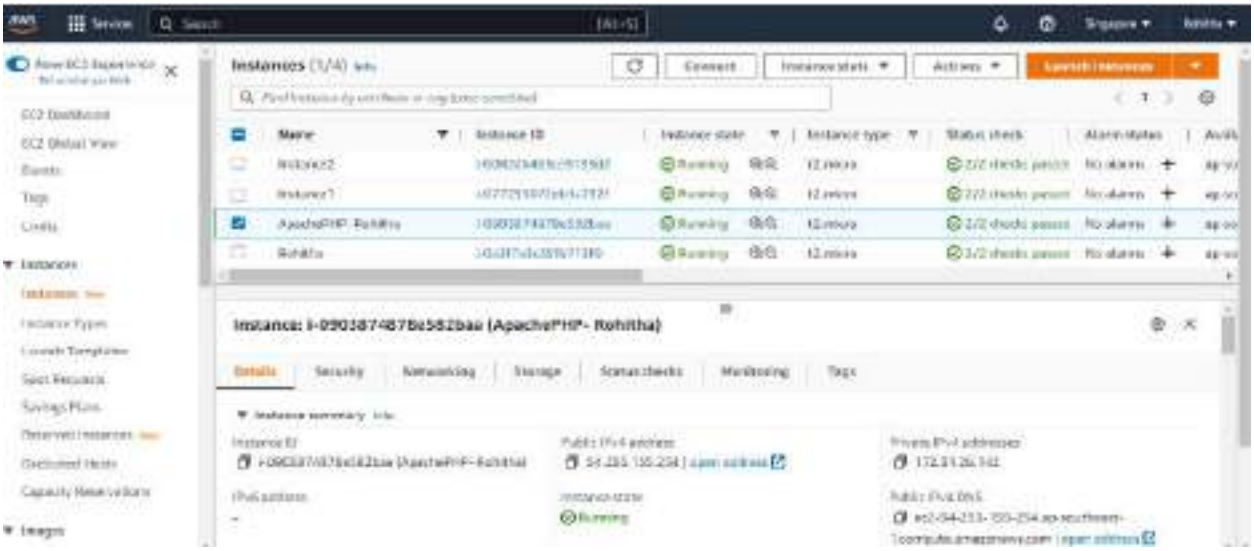


Create a volume from snapshot.

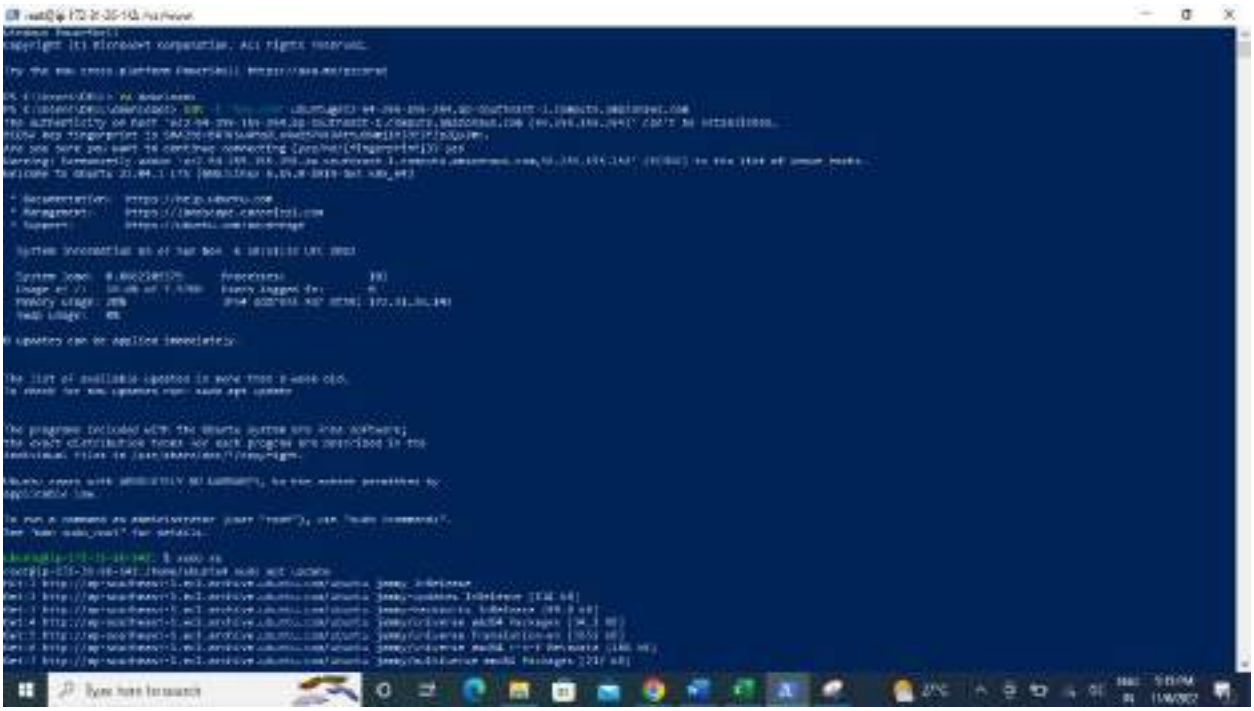


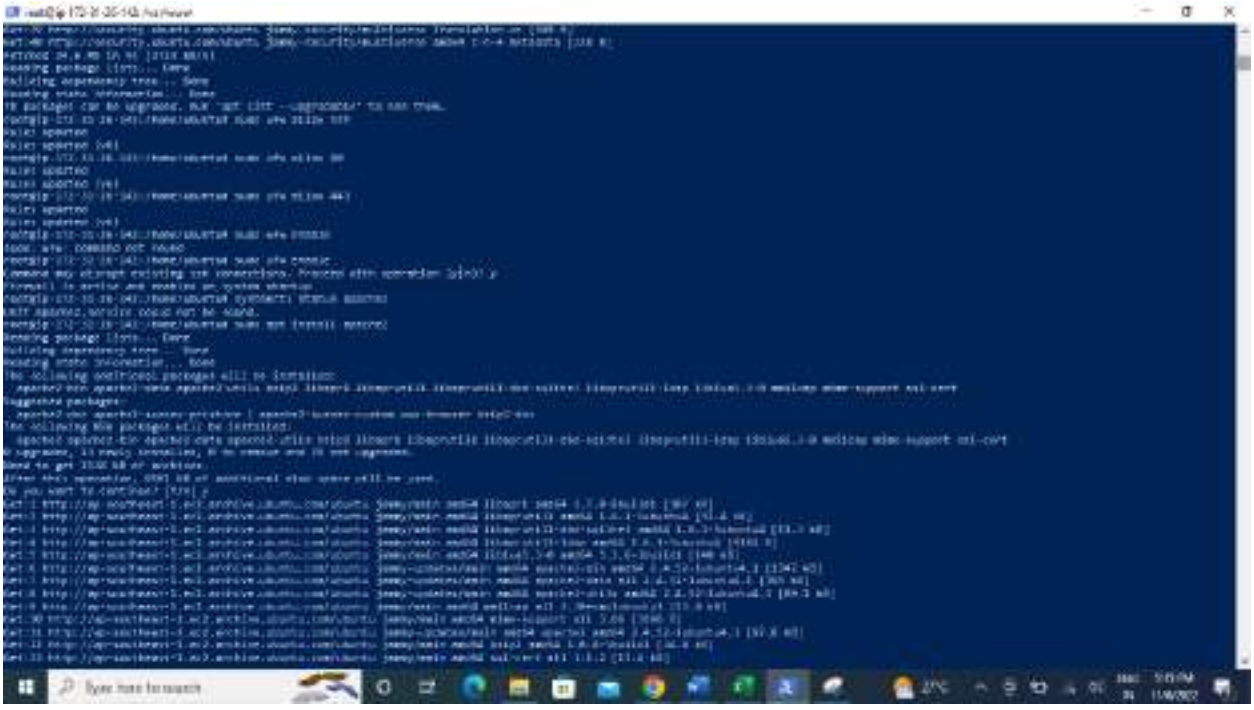
4. **AMI**:

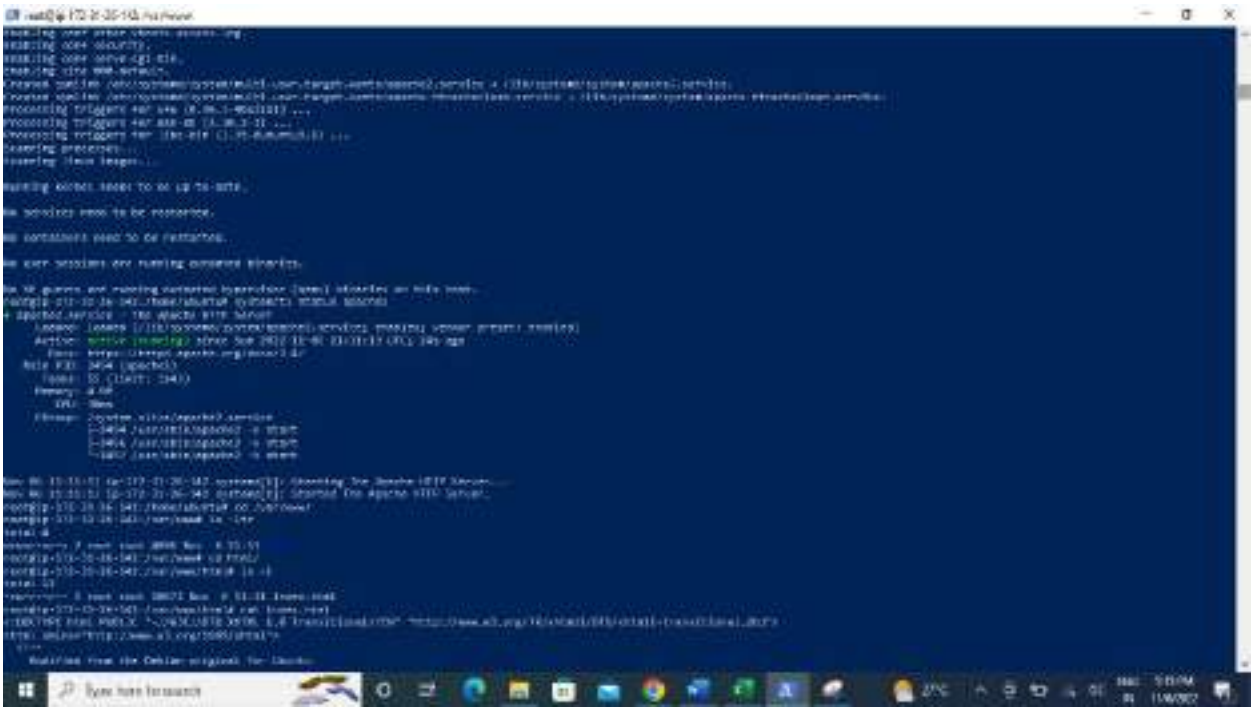
Create an instance and connect to Ubuntu server.

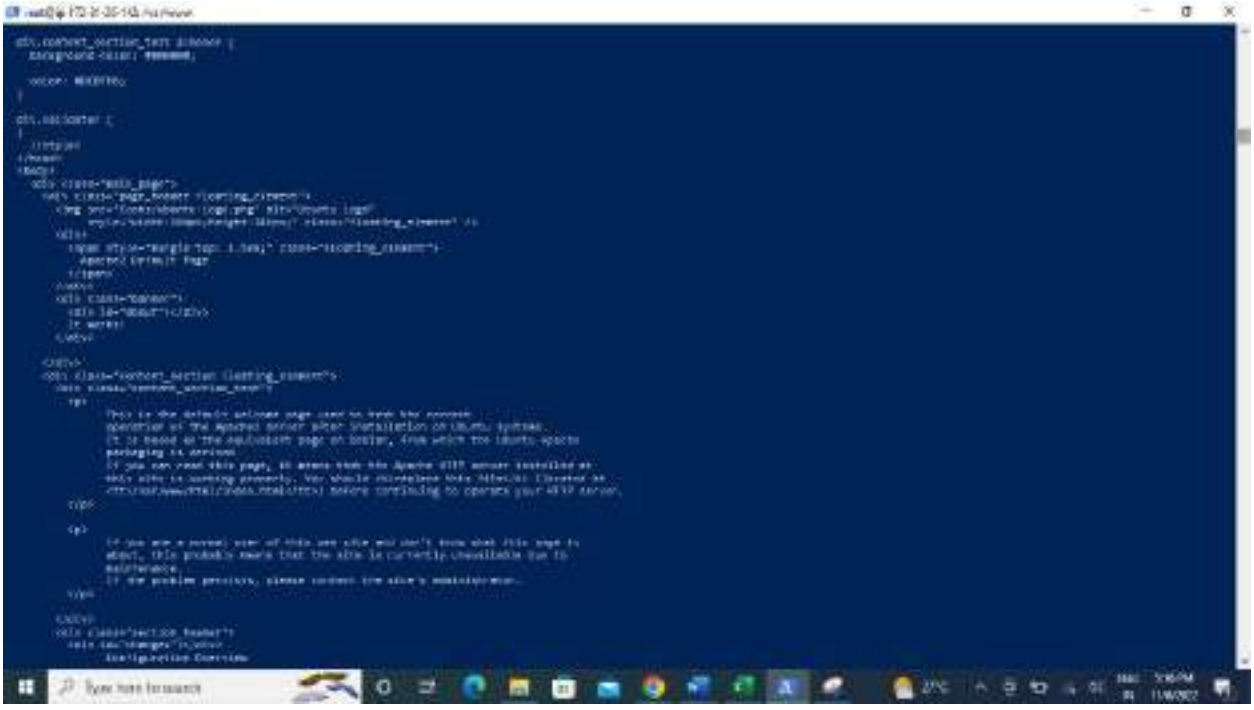


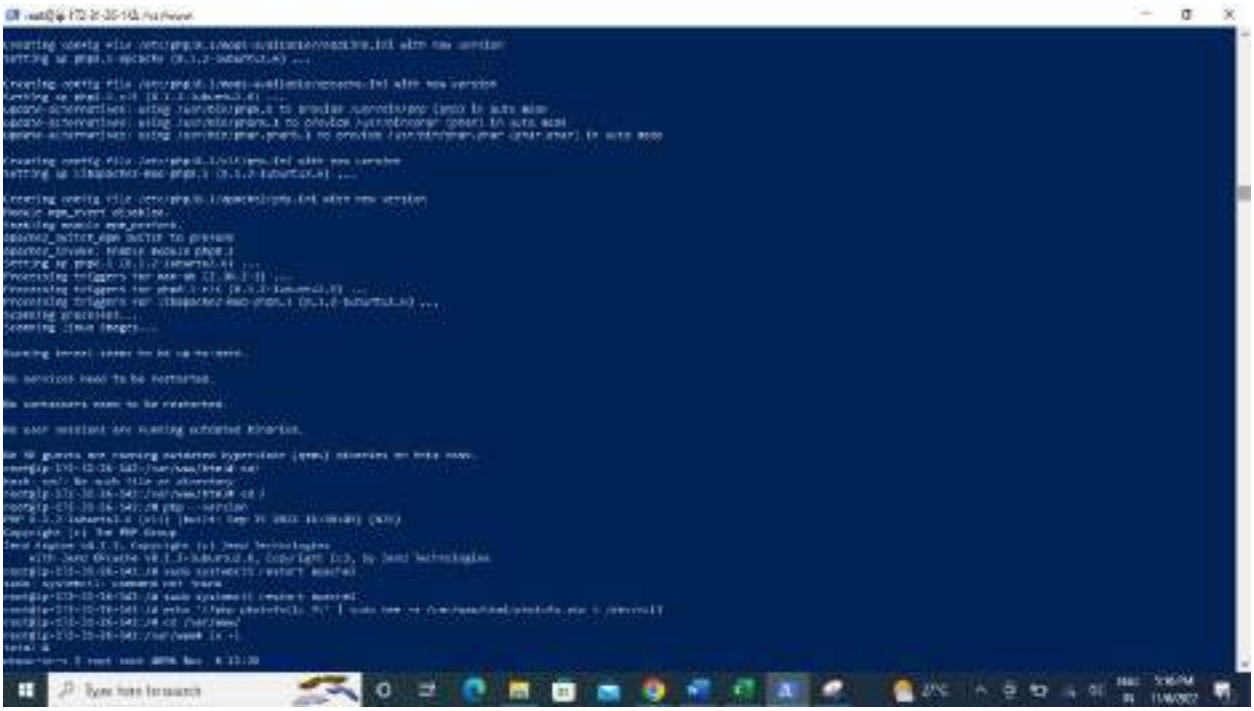
Connect to Apache and PHP server and launch a web application.



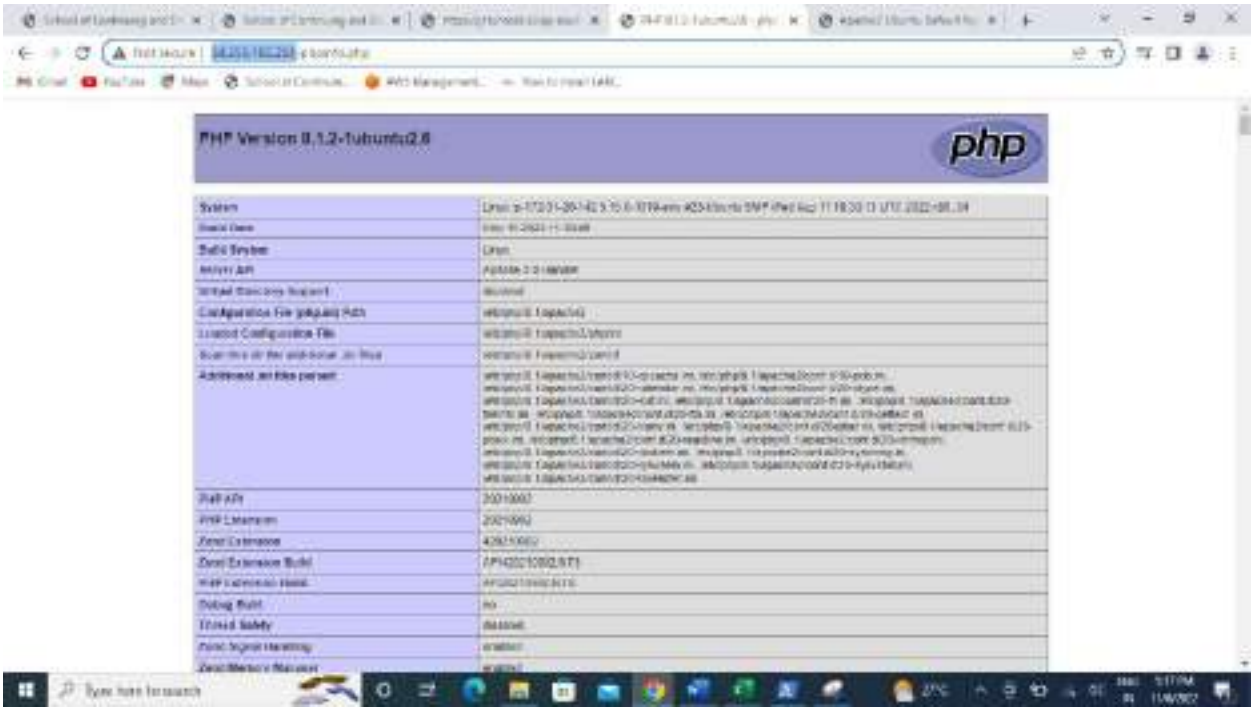




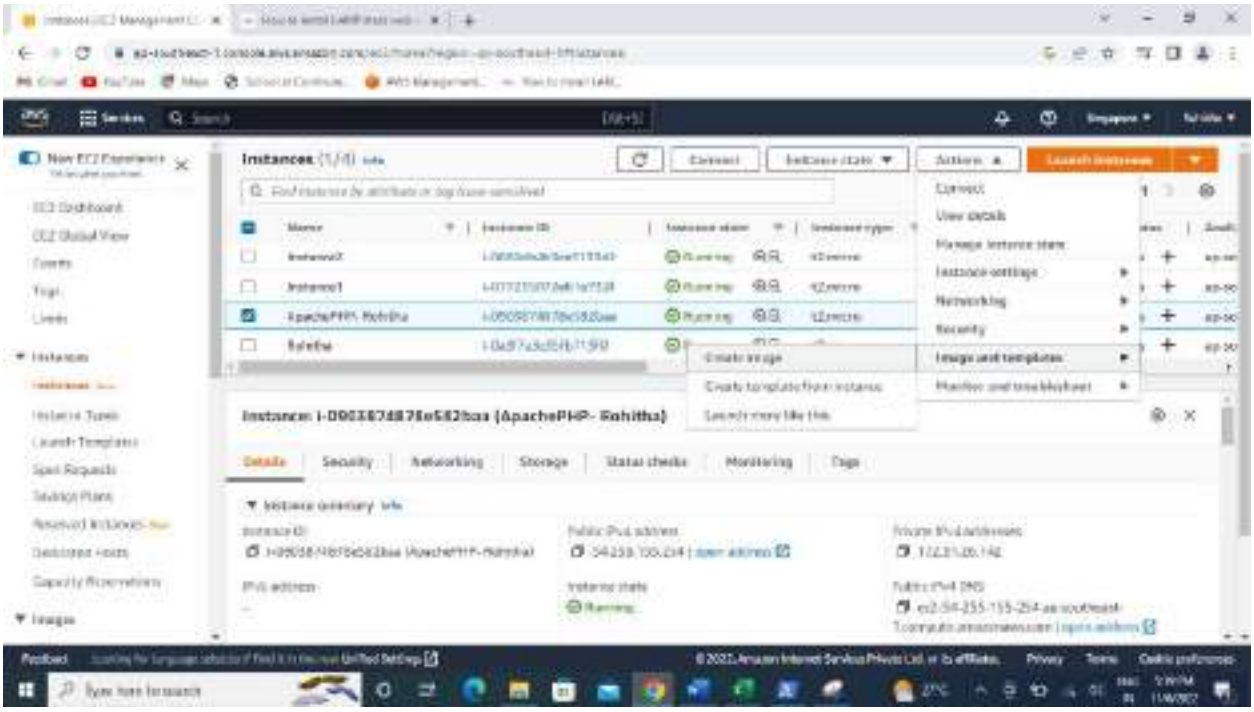


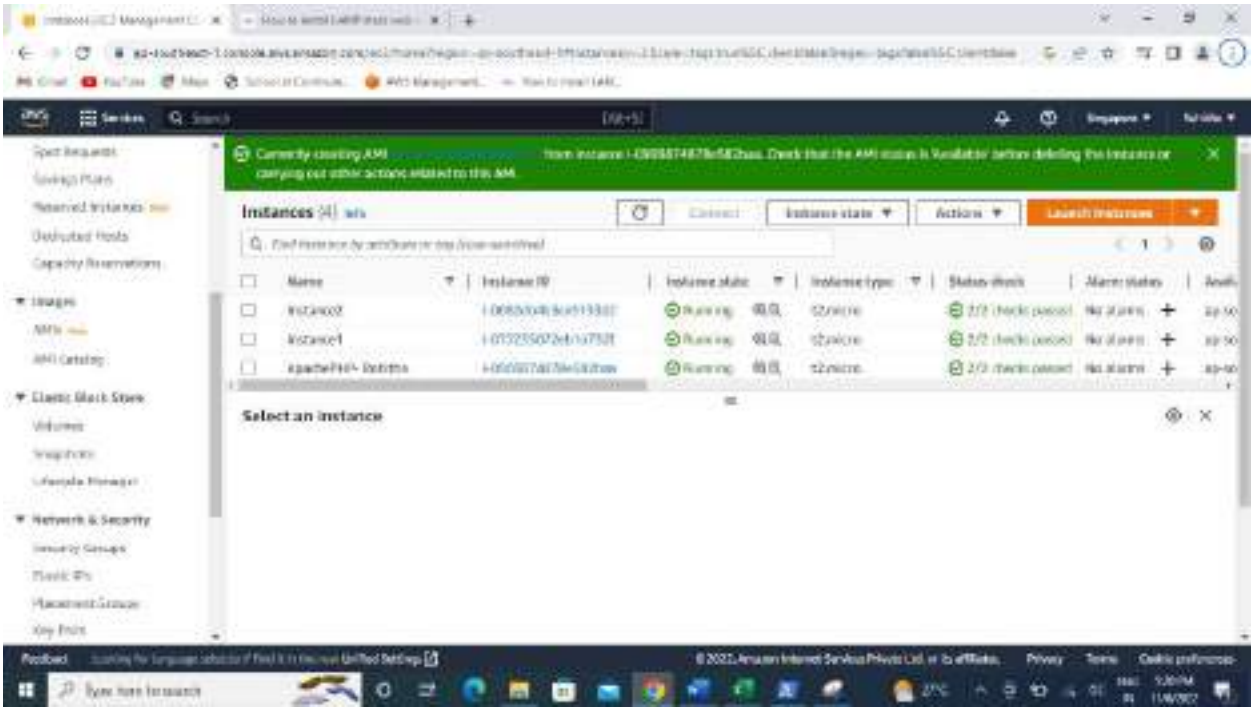


Launch an application using public ip address.



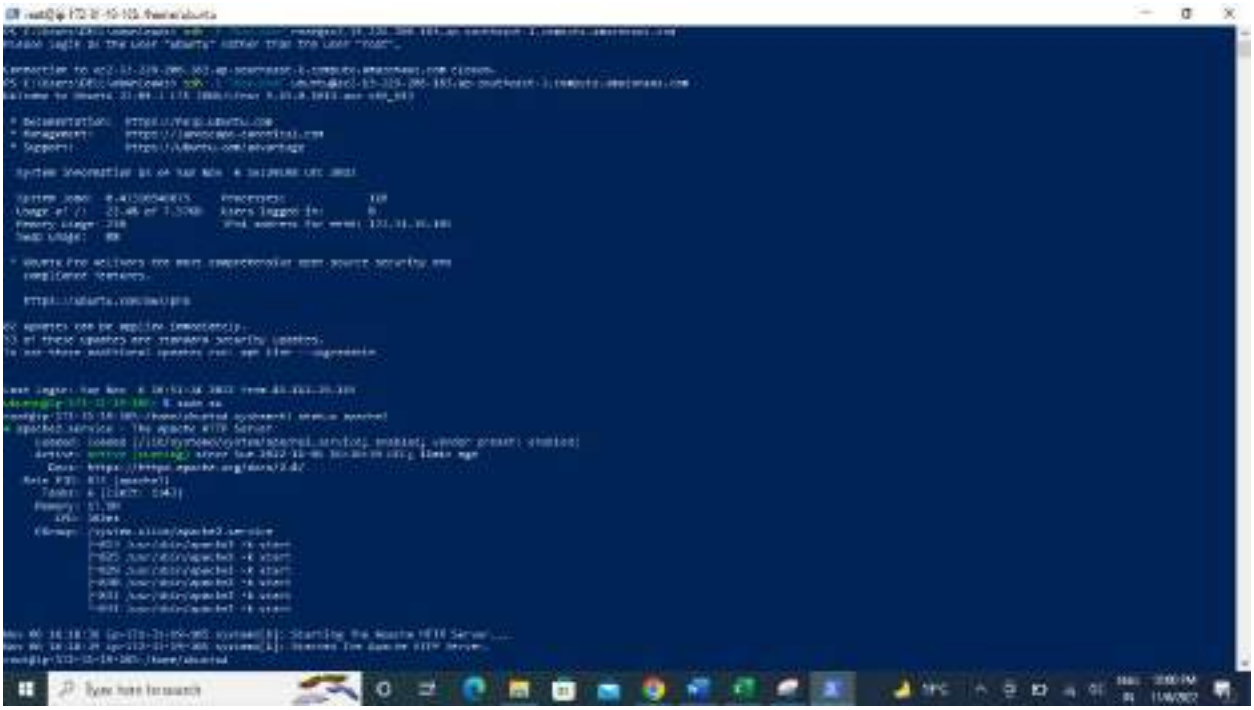
Now create an image from the instance.





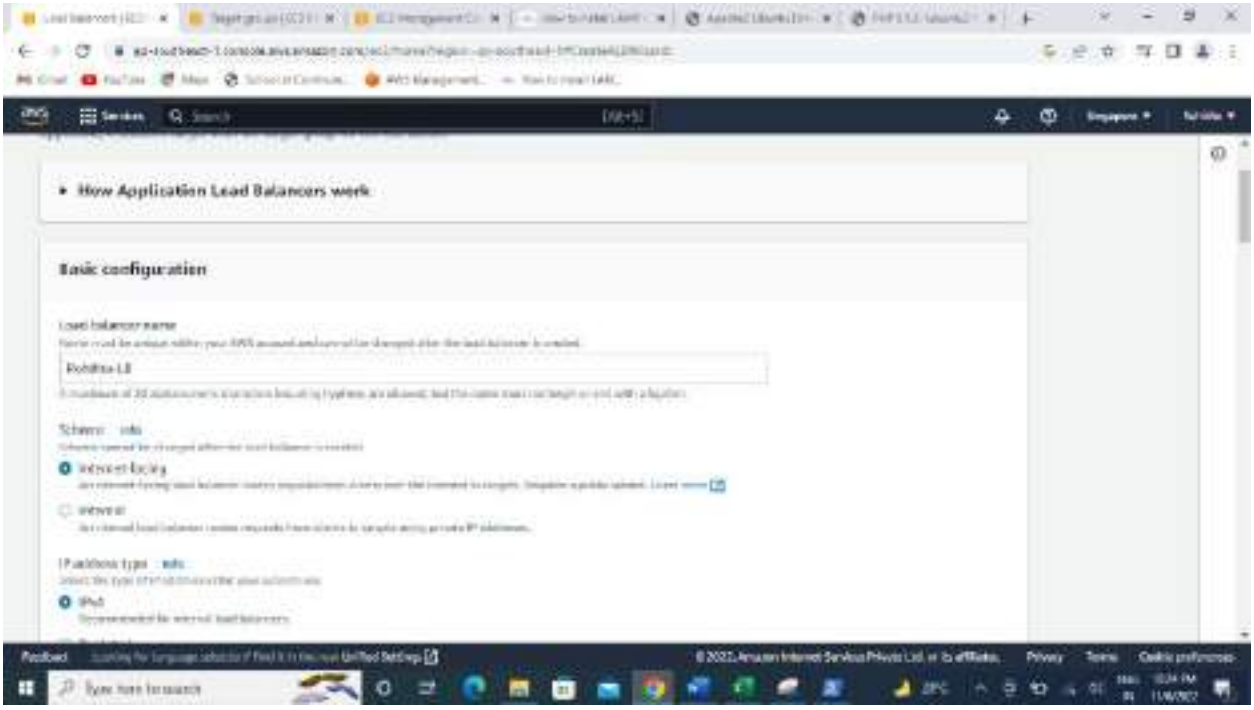
Go to AMI and select image and go to Launch instance EMI option.

Create an instance and connect to ubuntu machine.

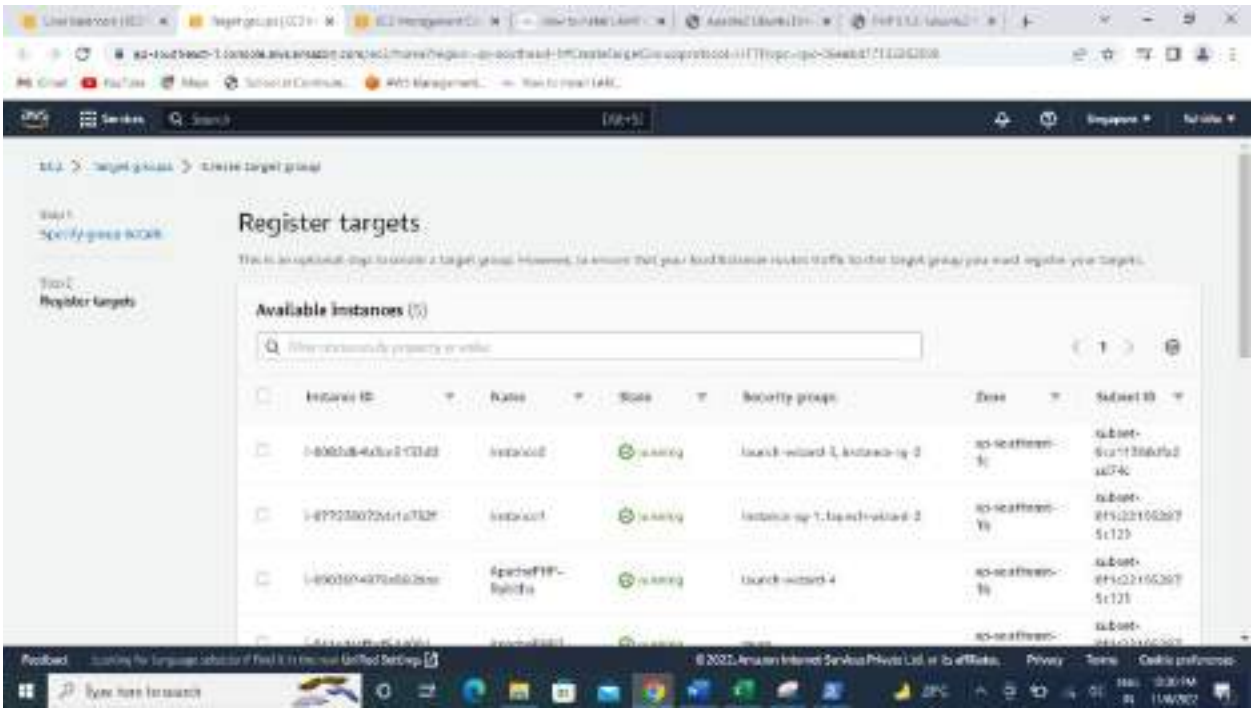


5. Load balancer:

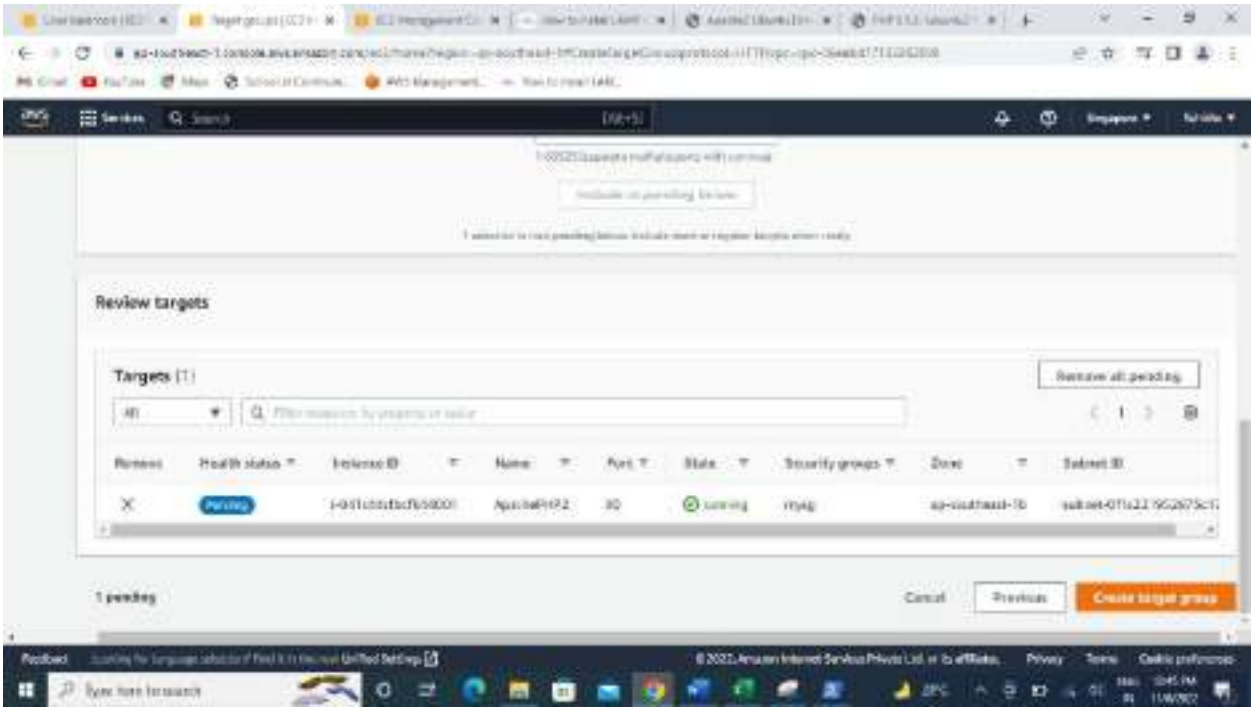
Create Application load balancer

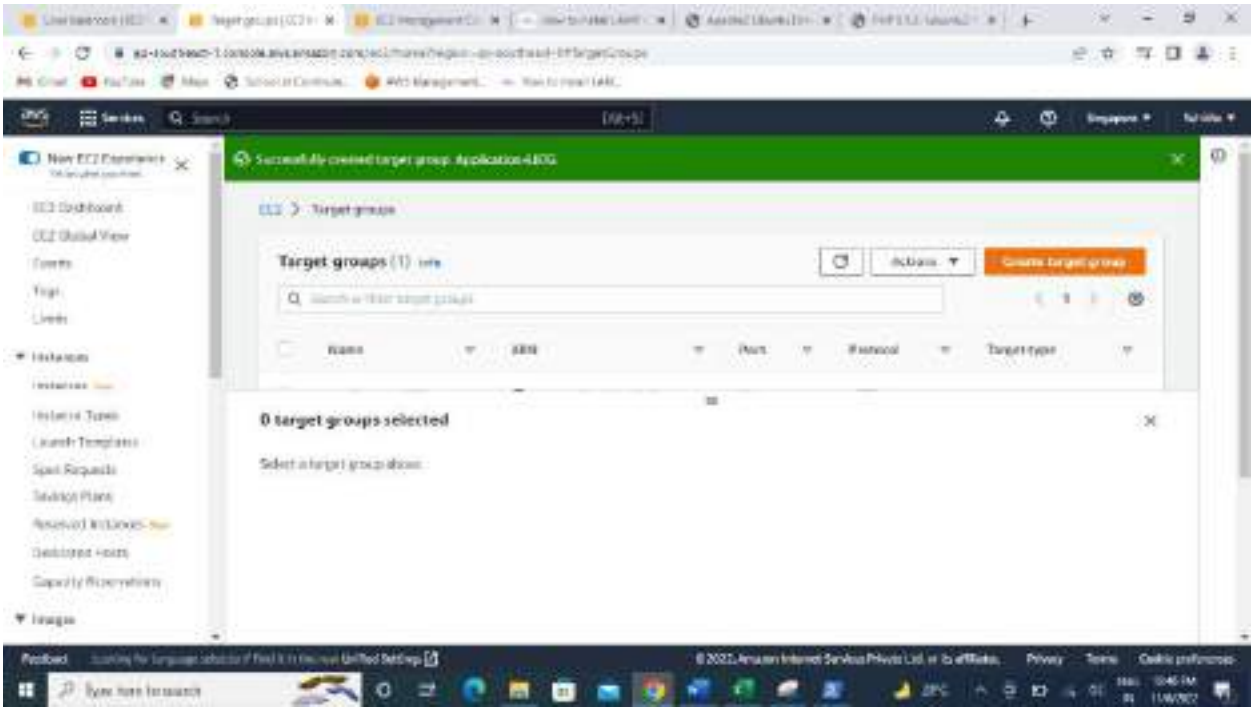


Create target groups under Listeners and routing.

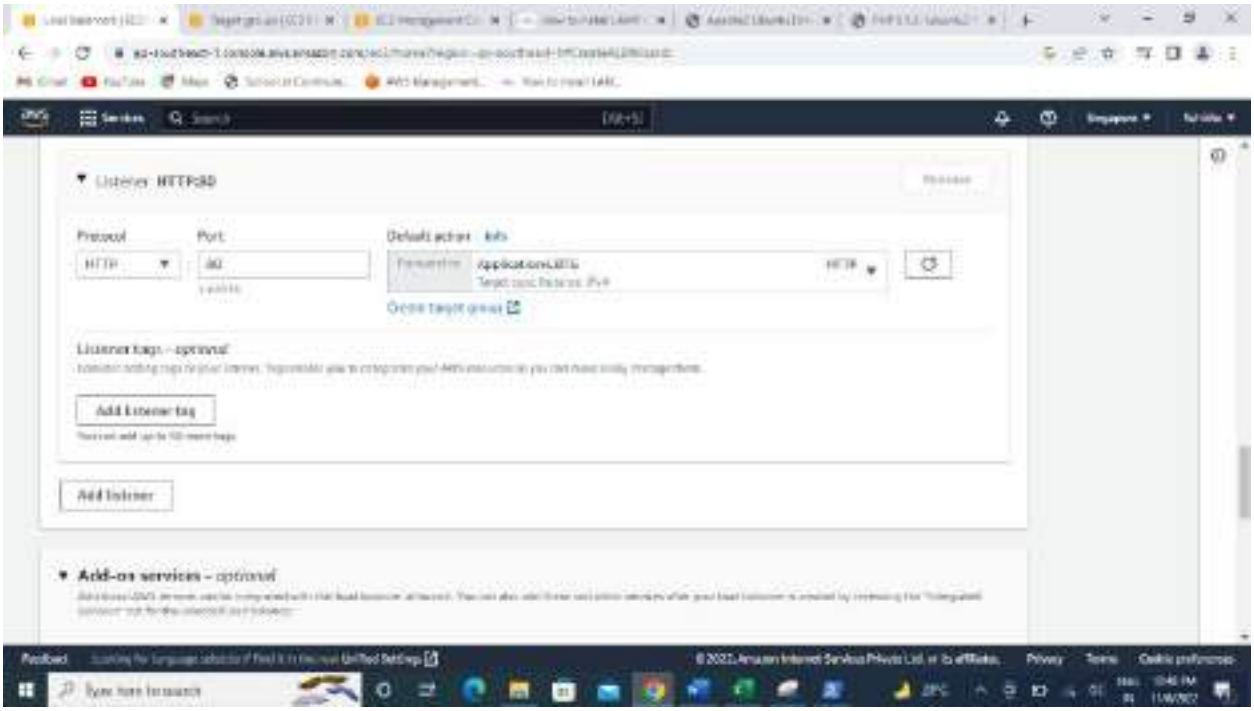


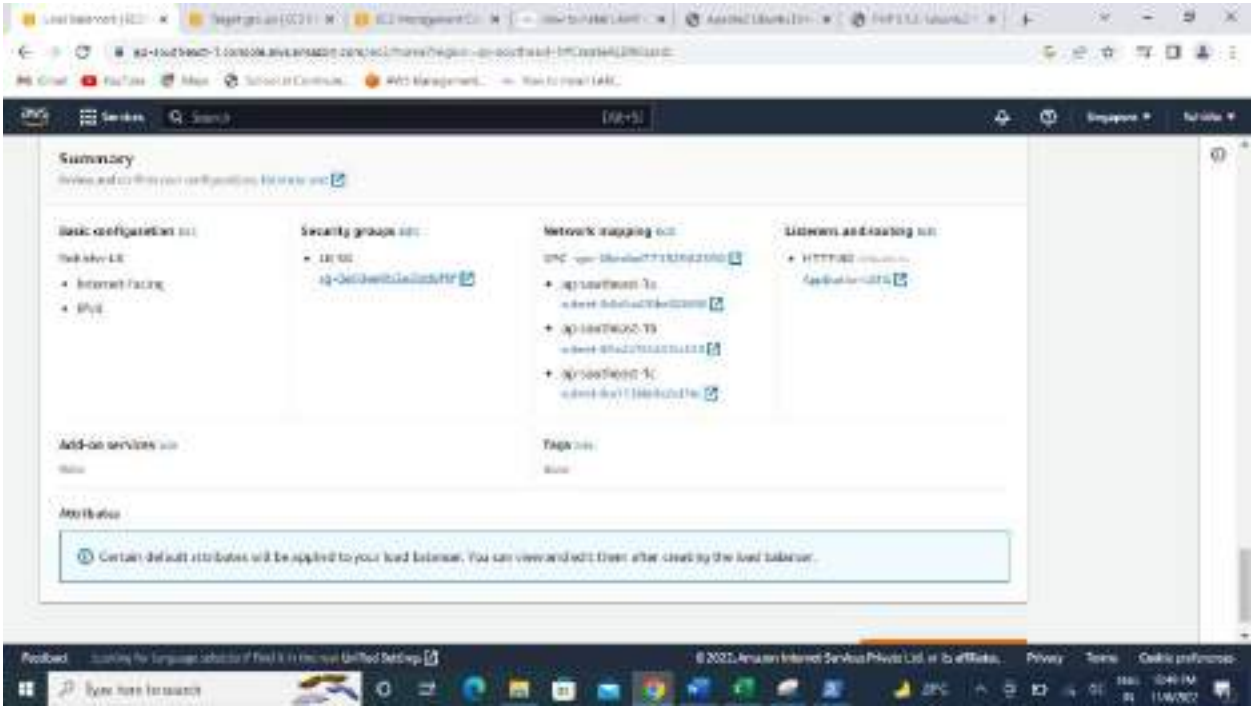
Select AMI instance machine and click on Include as pending below option

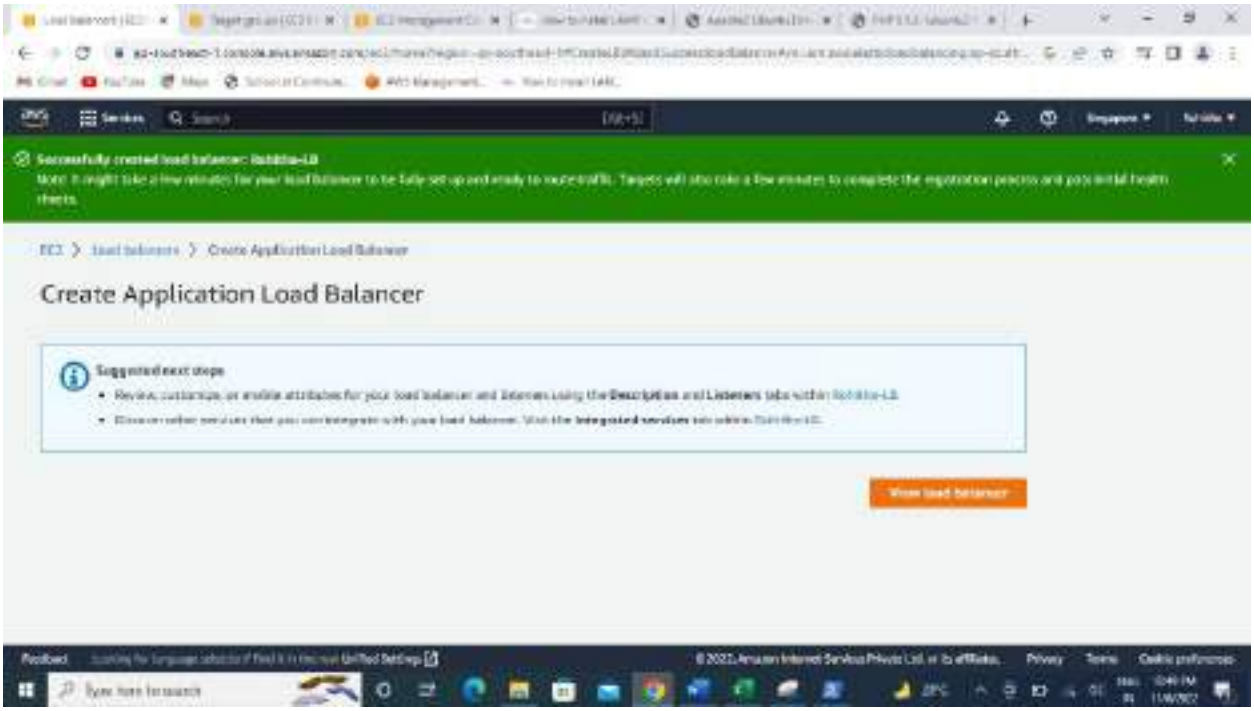


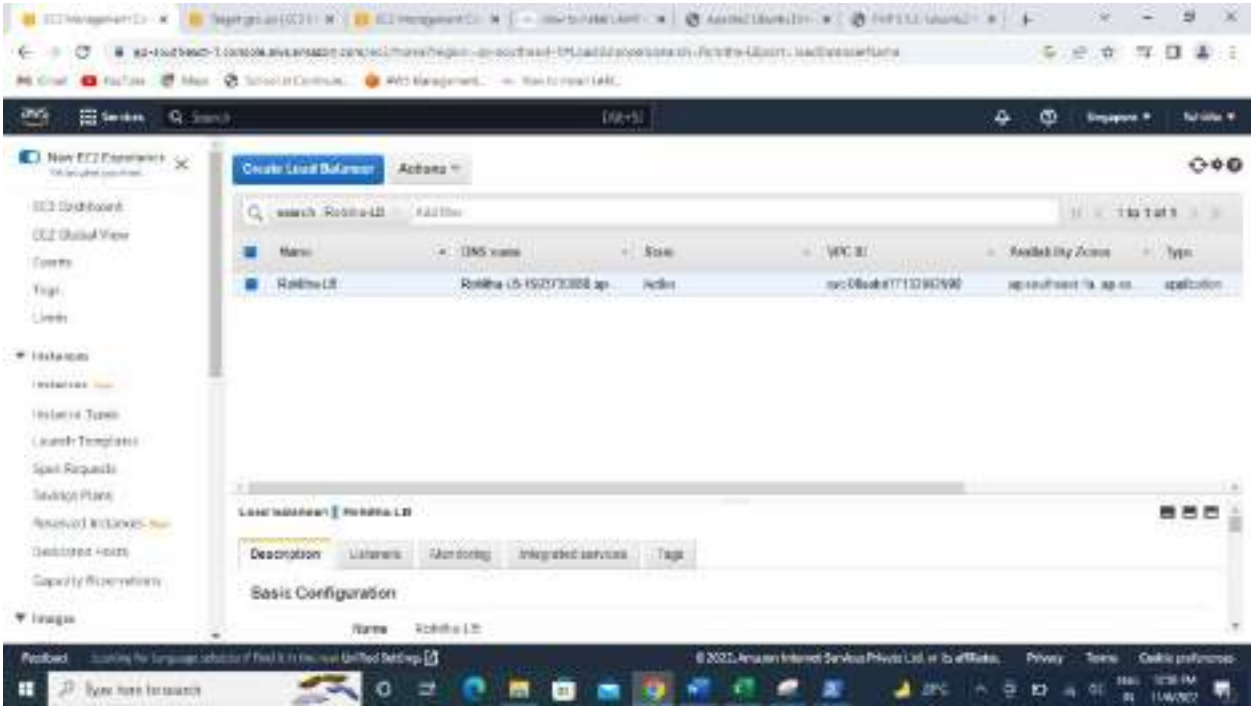


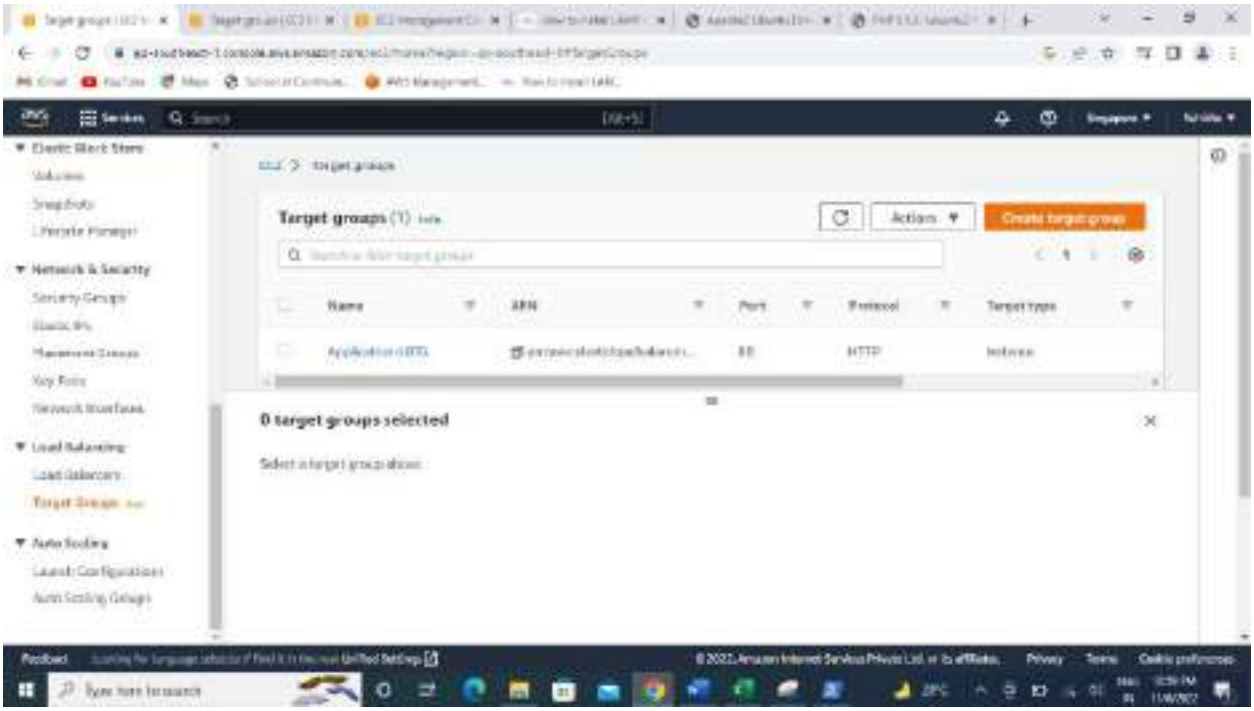
Once target group is created , select TG while creating Application load balancer as below:

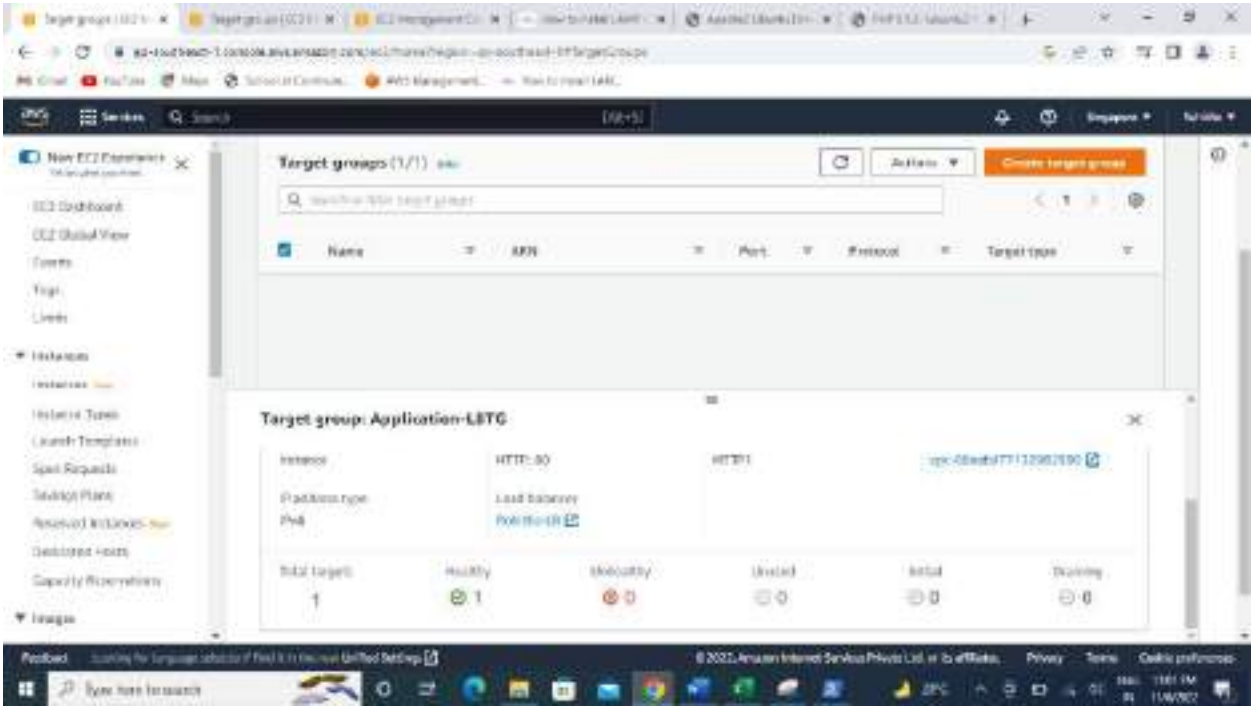

Listener HTTP:80
Protocol
Port
Default actions
Add listener tag
Add listener
Add-on services – optional


Summary
Basic configuration
Security groups
Network mapping
Listeners and routing
Add-on services
Tags
Attributes

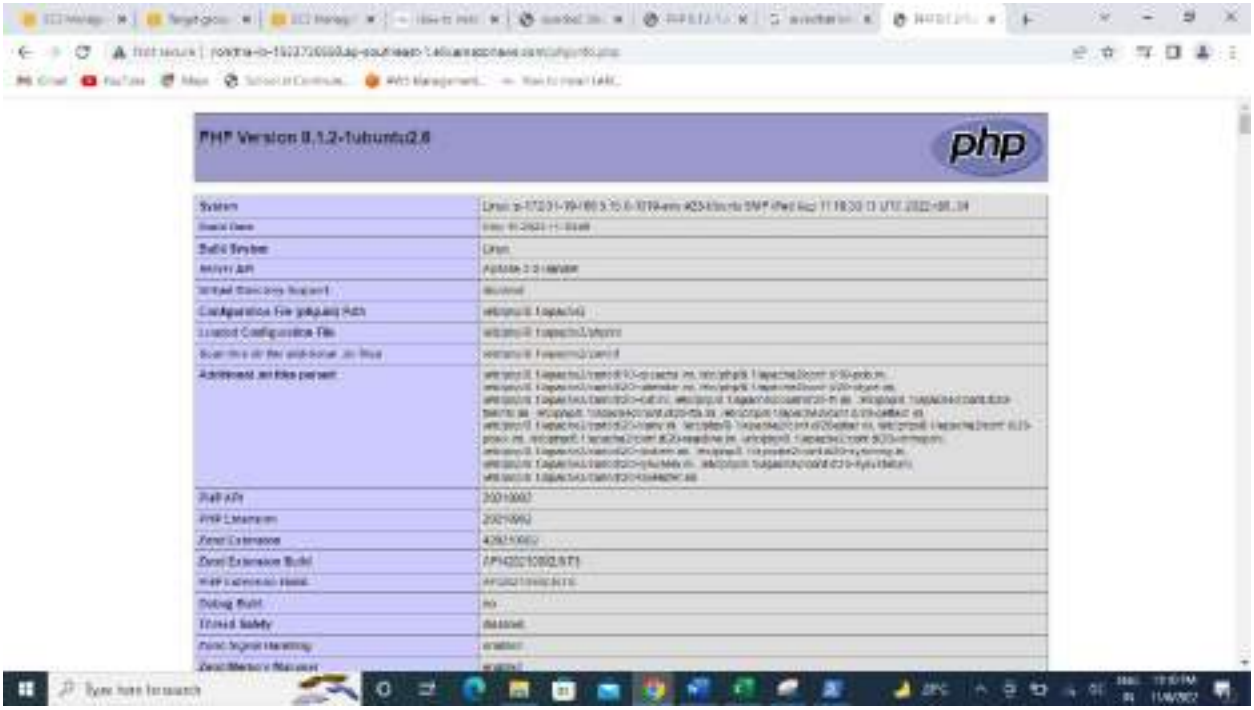

Create Application Load Balancer





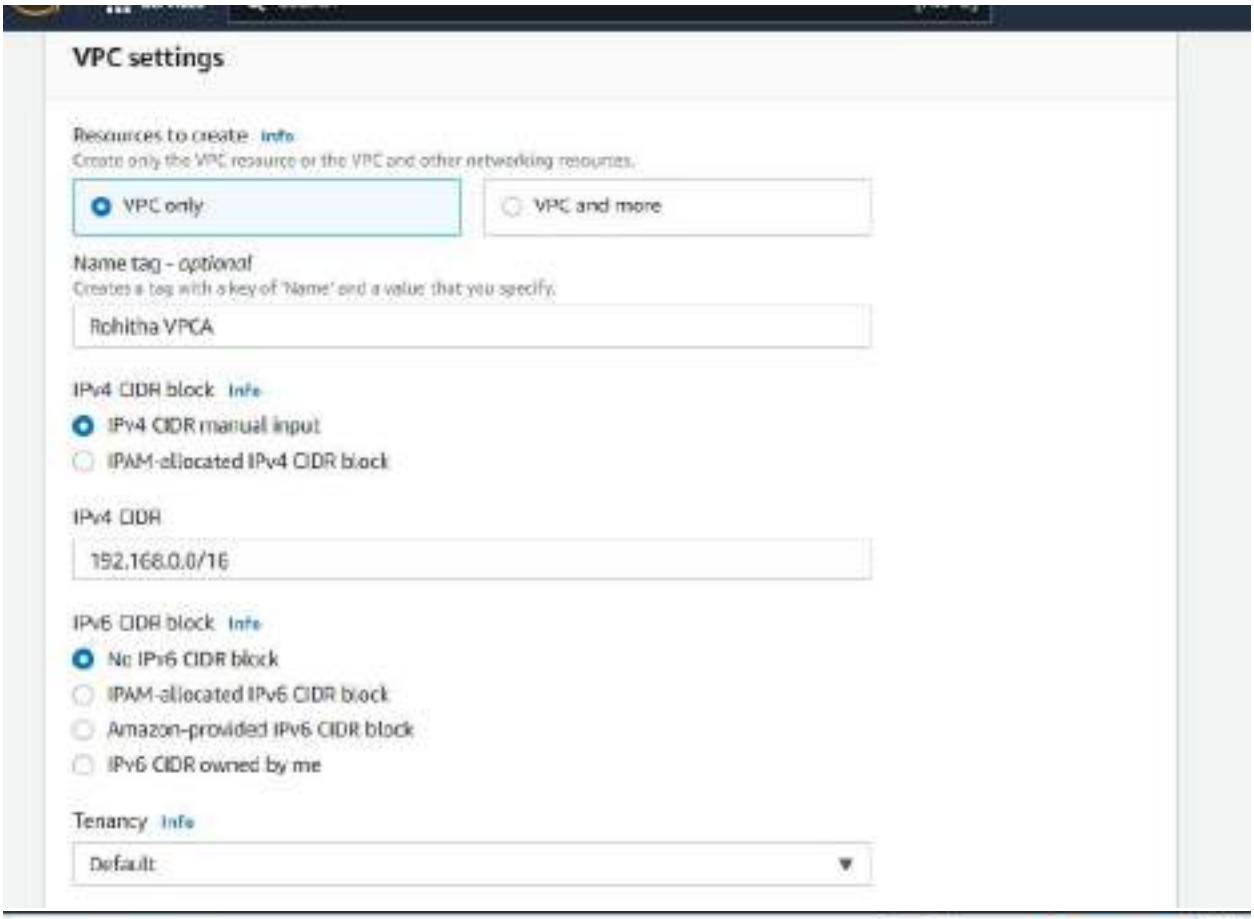


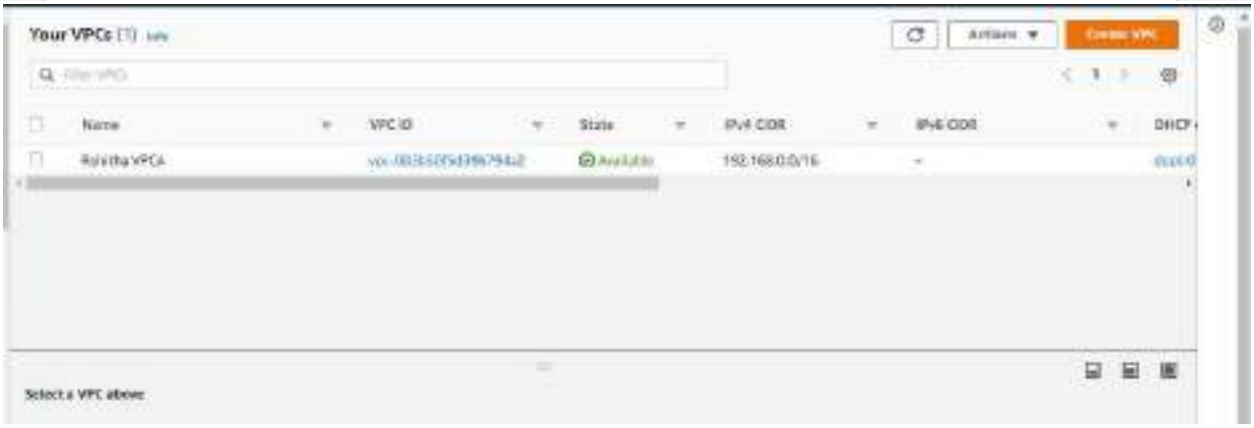
Select DNS from load balancer.



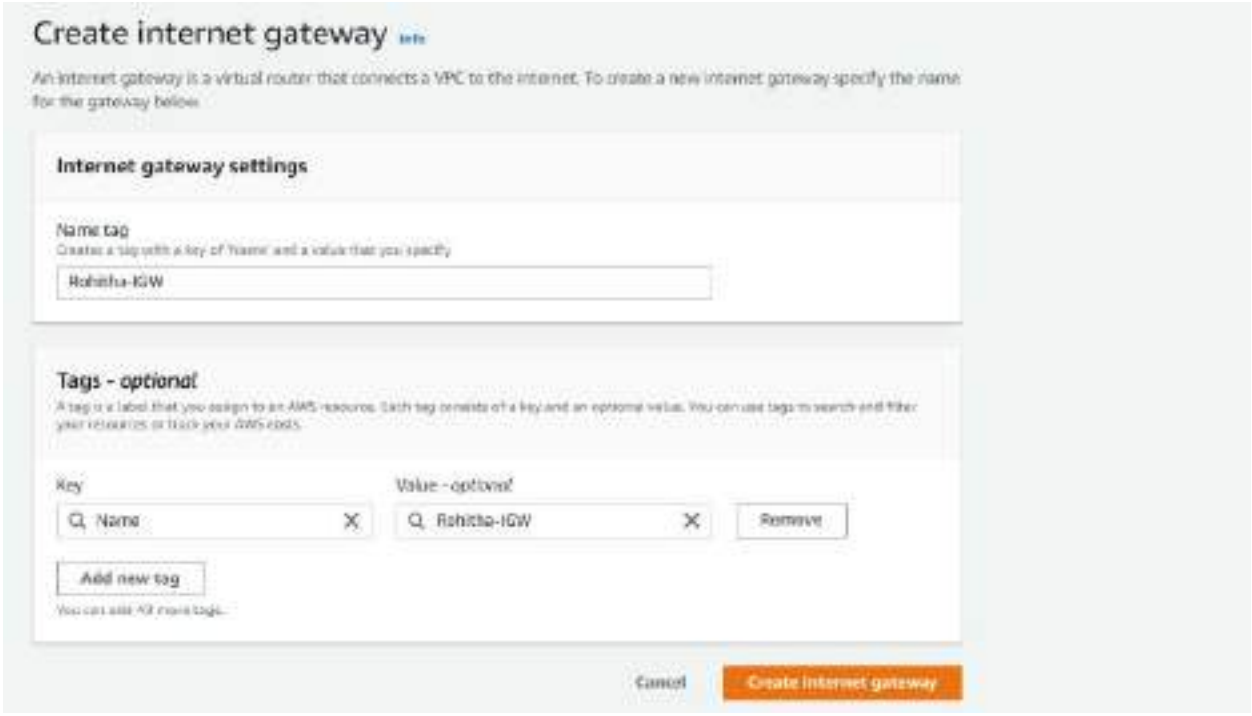
6. VPC :

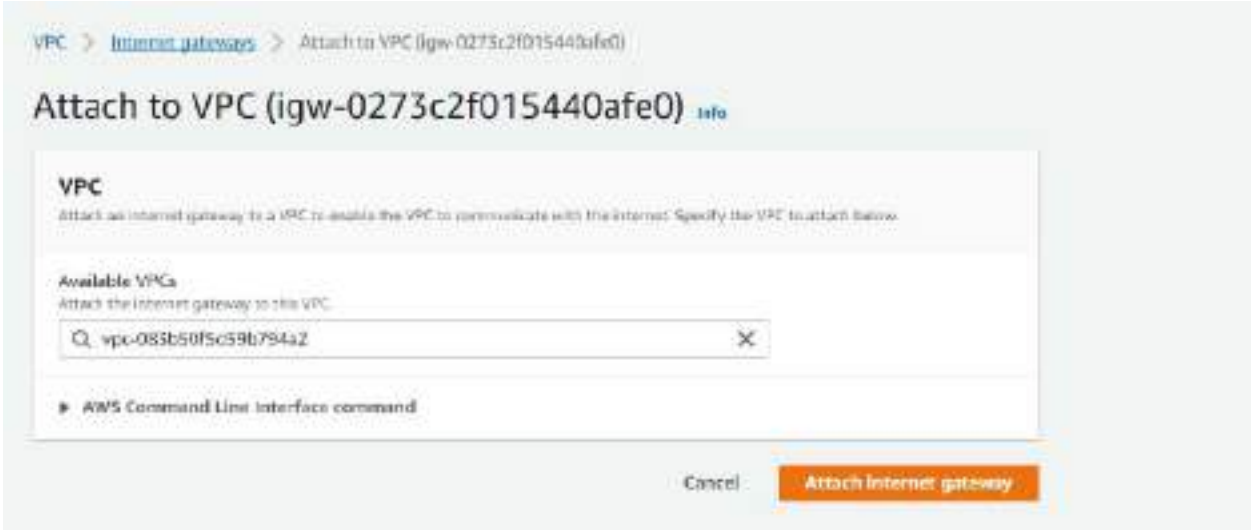
1. Create VPC and provide range of VPC as 192.168.0.0/16



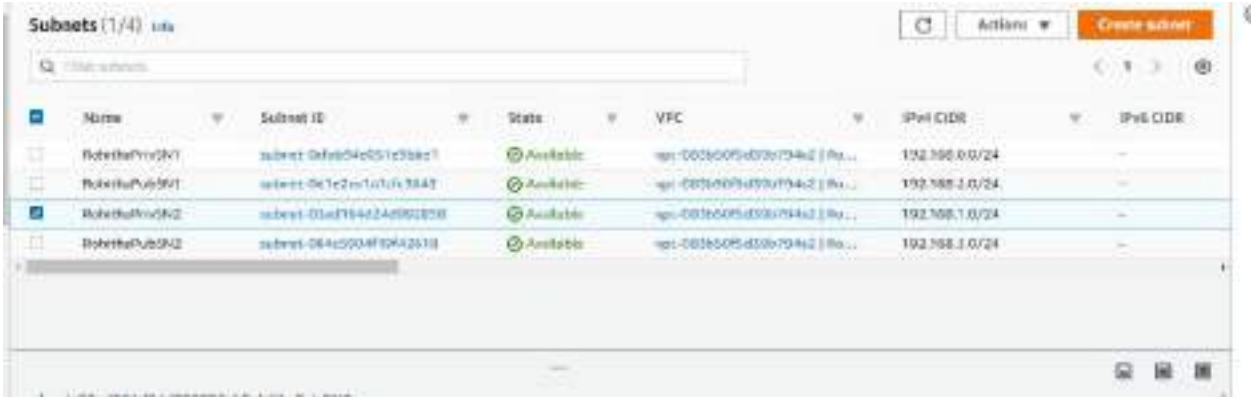


2. Create an internet gateway and attach it to VPC.



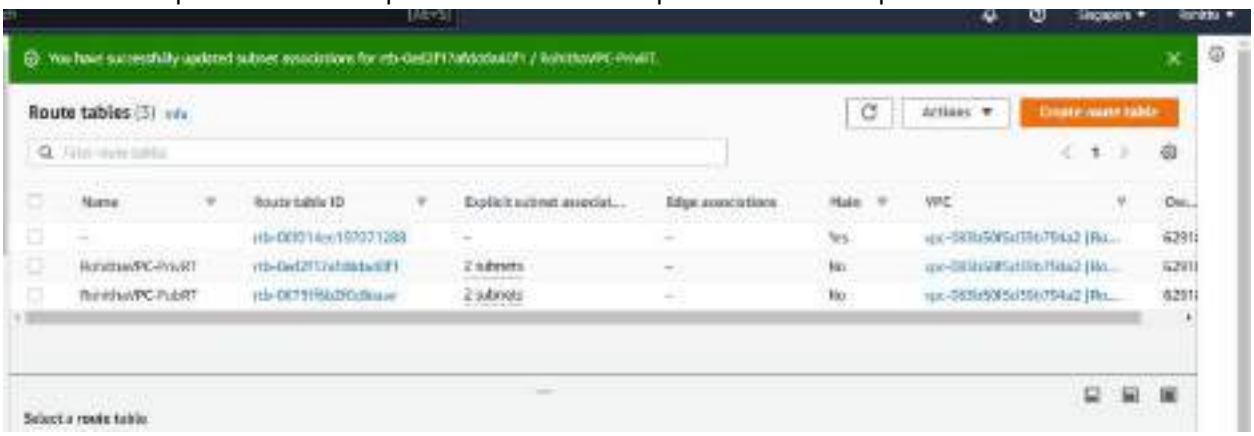


3. Create 4 subnets (2 public and 2 private)

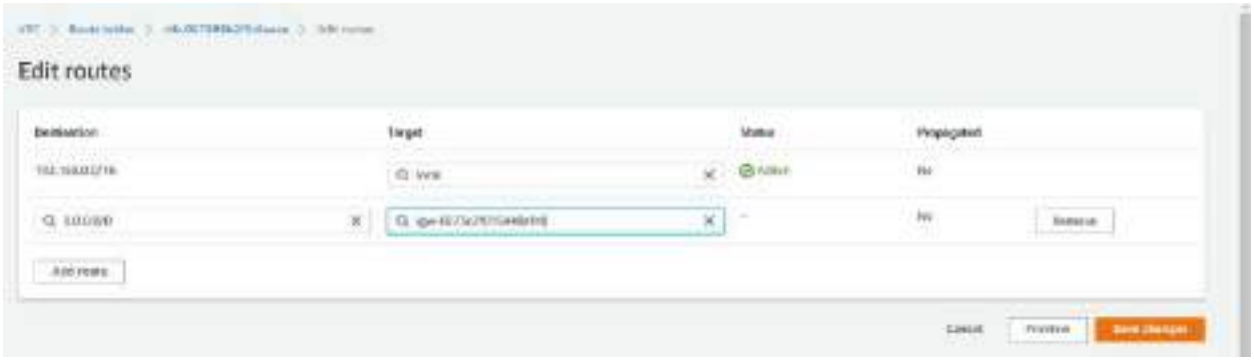


4. Create 2 route tables (public and private)

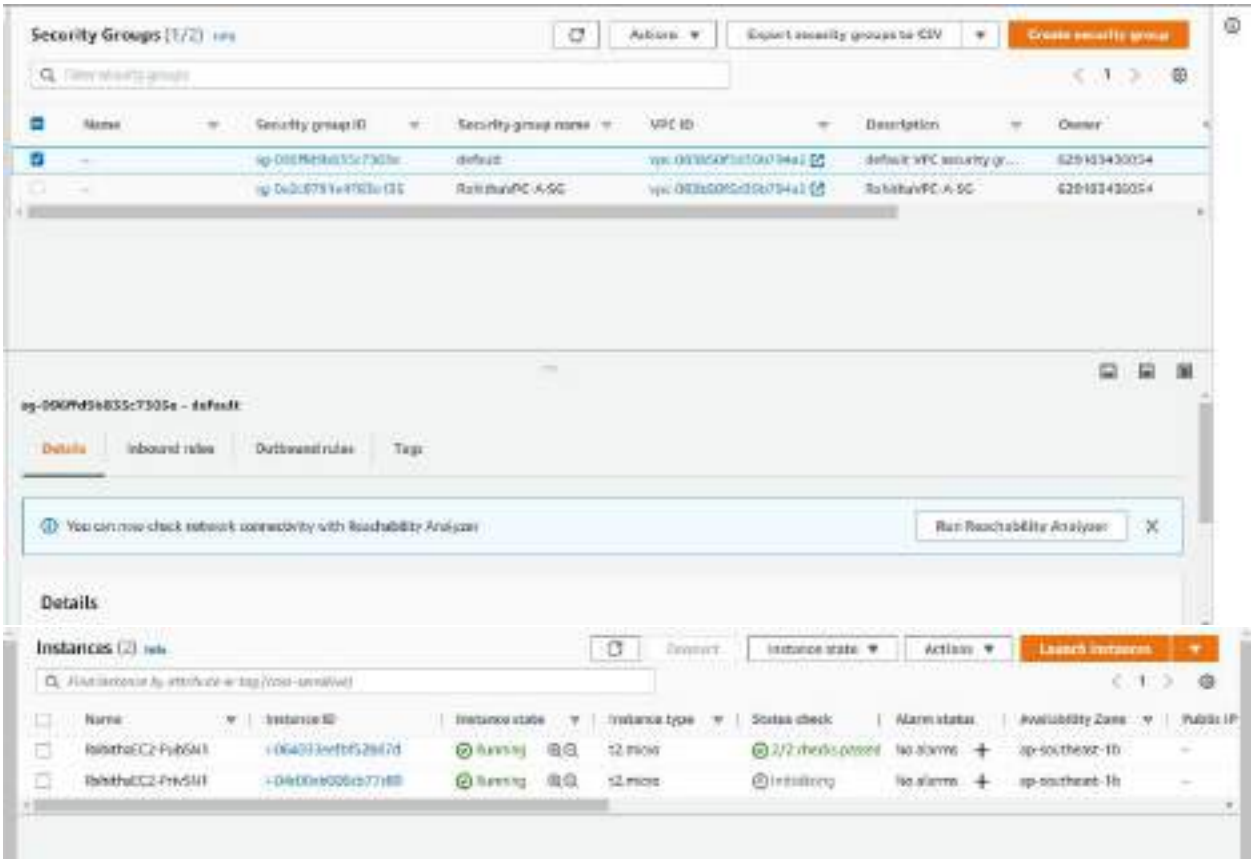
5. Associate the public subnets to public route table and private subnets to private route table.



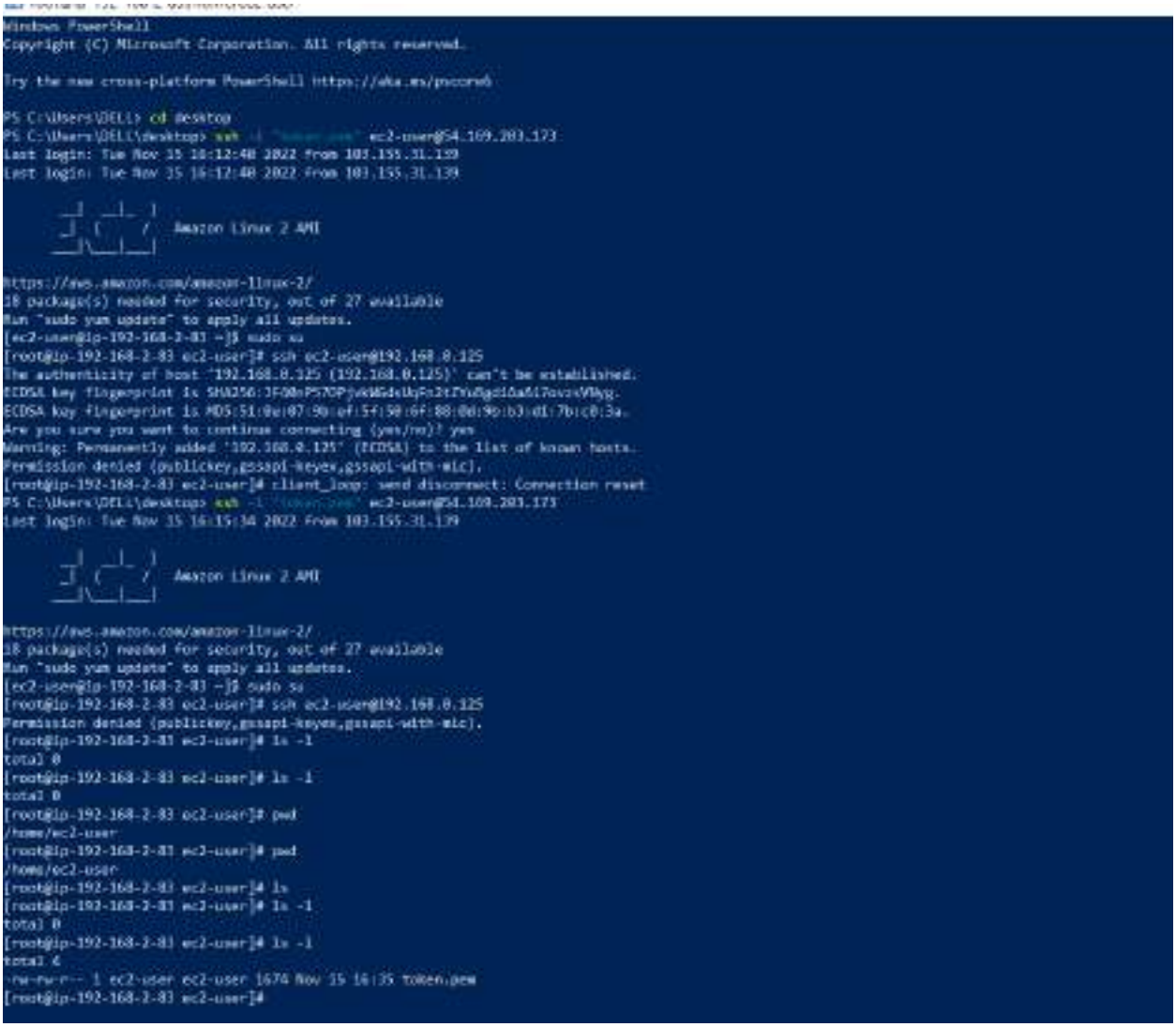
6. Connect public route table to the internet gateway.



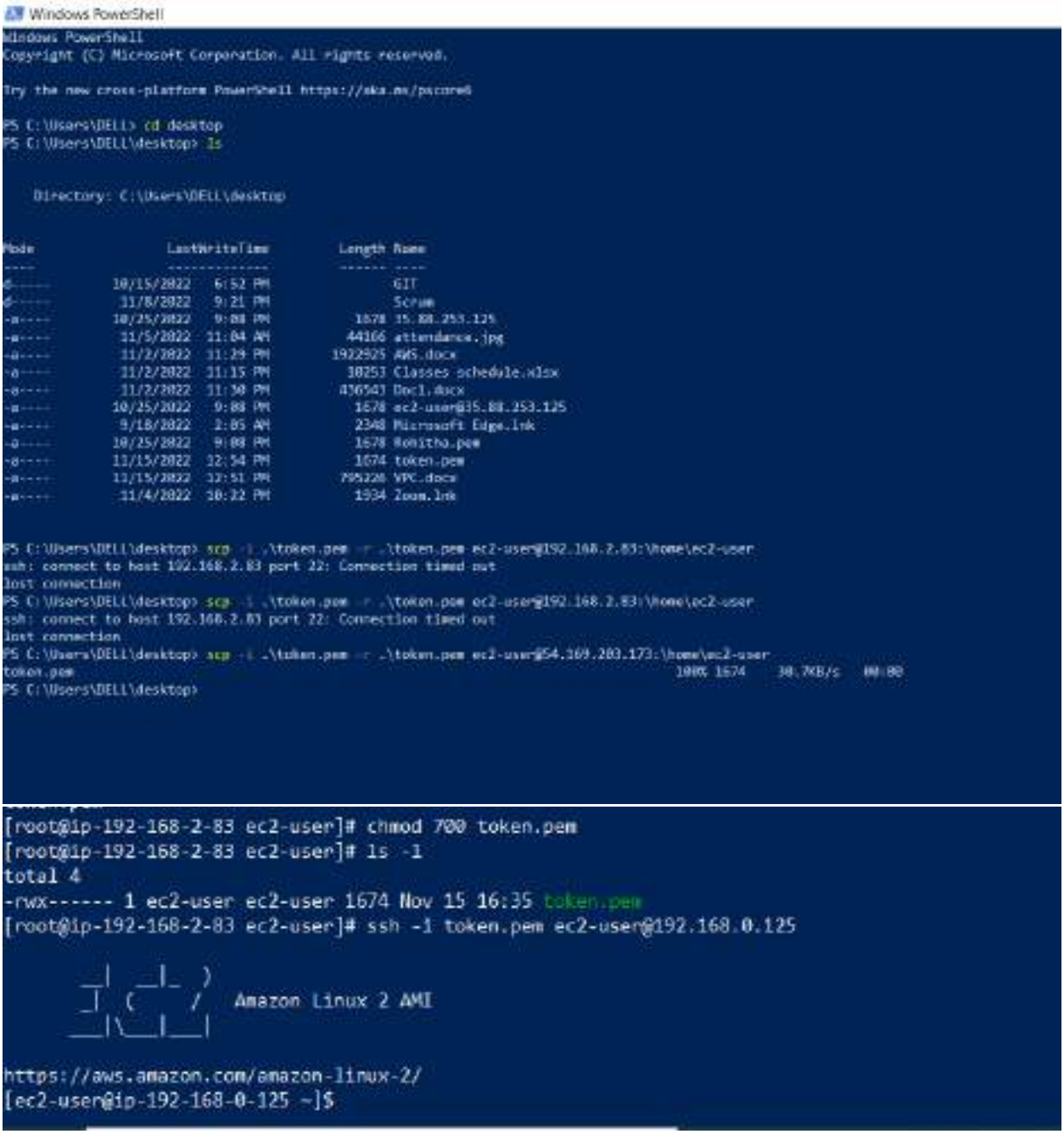
7. Create 2 instances (public subnet 1, private subnet 1)
   Also create new security group by allowing all traffic.



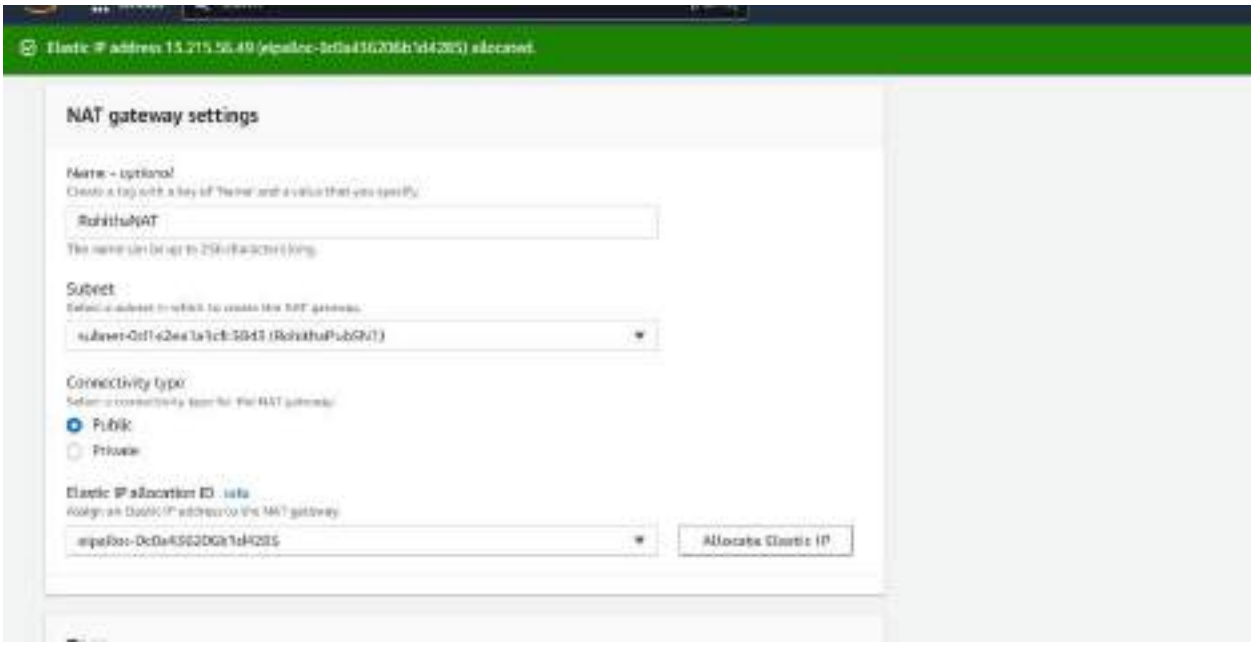
8. Login to Public subnet1 machine and try to ping and ssh (ping will get success and where ssh will fail as there is no key)



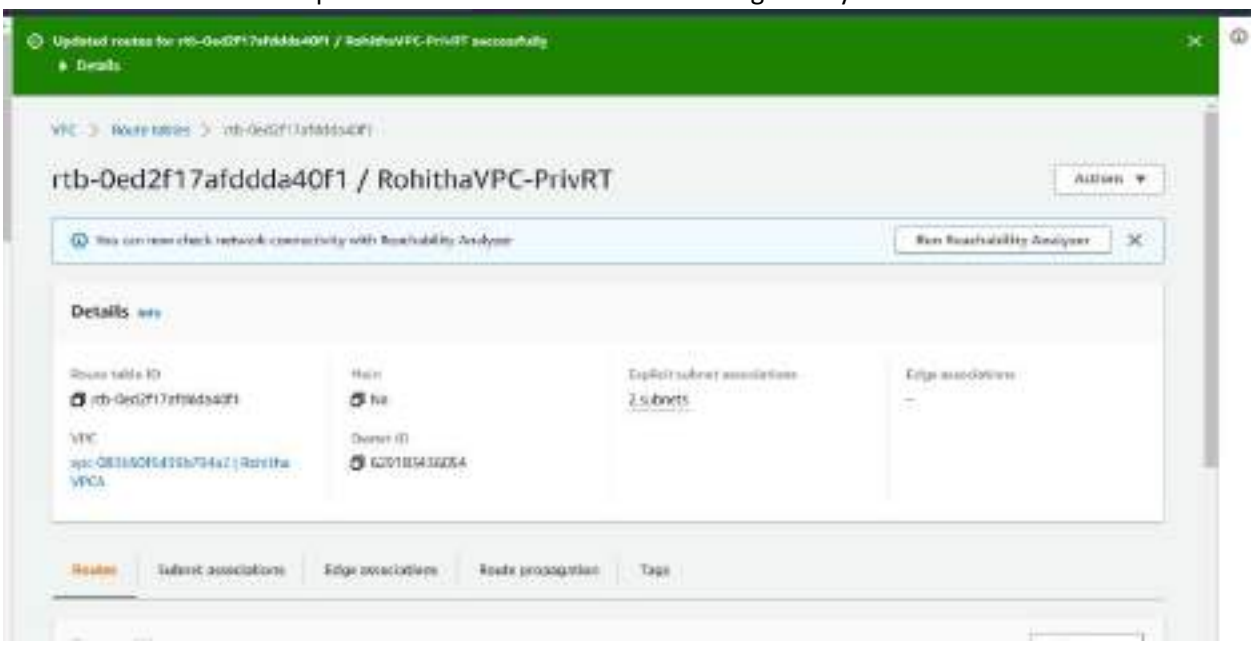
9. Copy the key pair on to Public subnet machine1 and then from Public subnet machine try to connect private subnet machine with SSH.



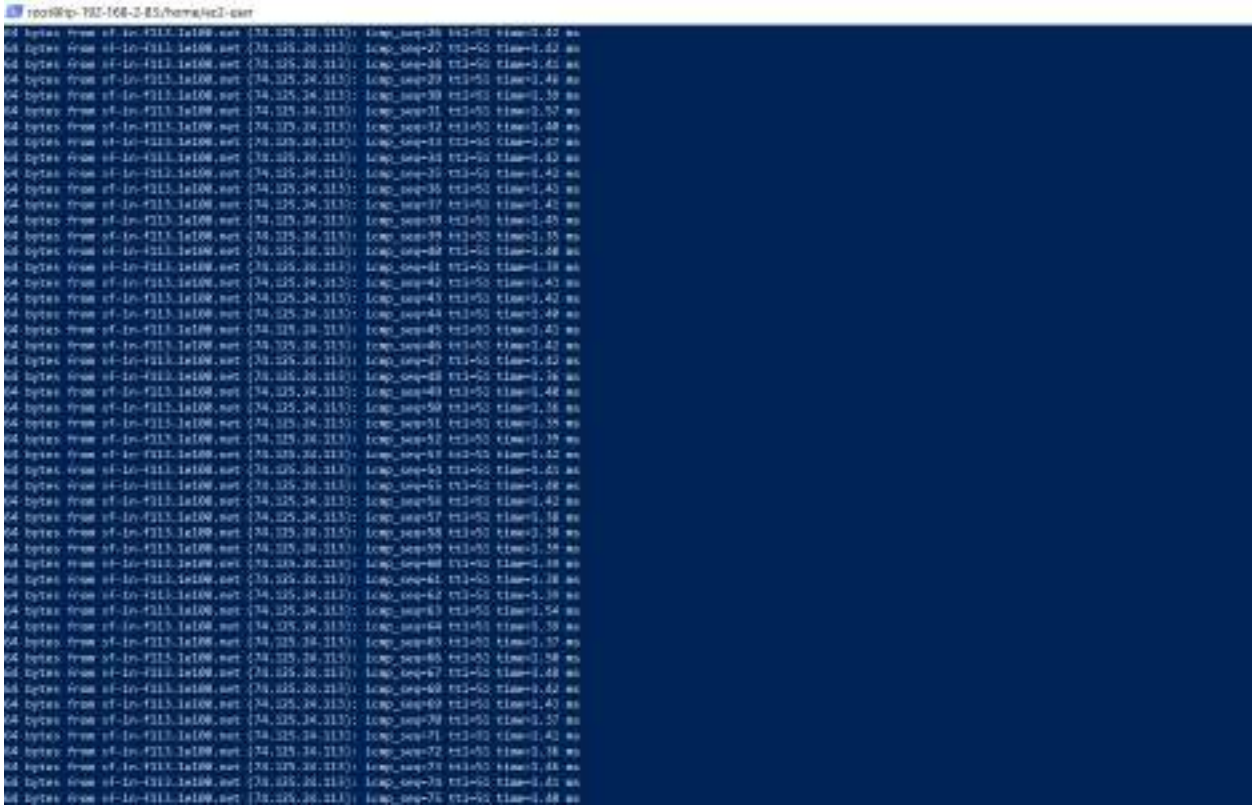
10. In order to get internet on your Private subnet machine then we have to Create a NAT gateway in Public subnet1.



11. Edit the route table of private subnet1 with a route to NAT gateway.

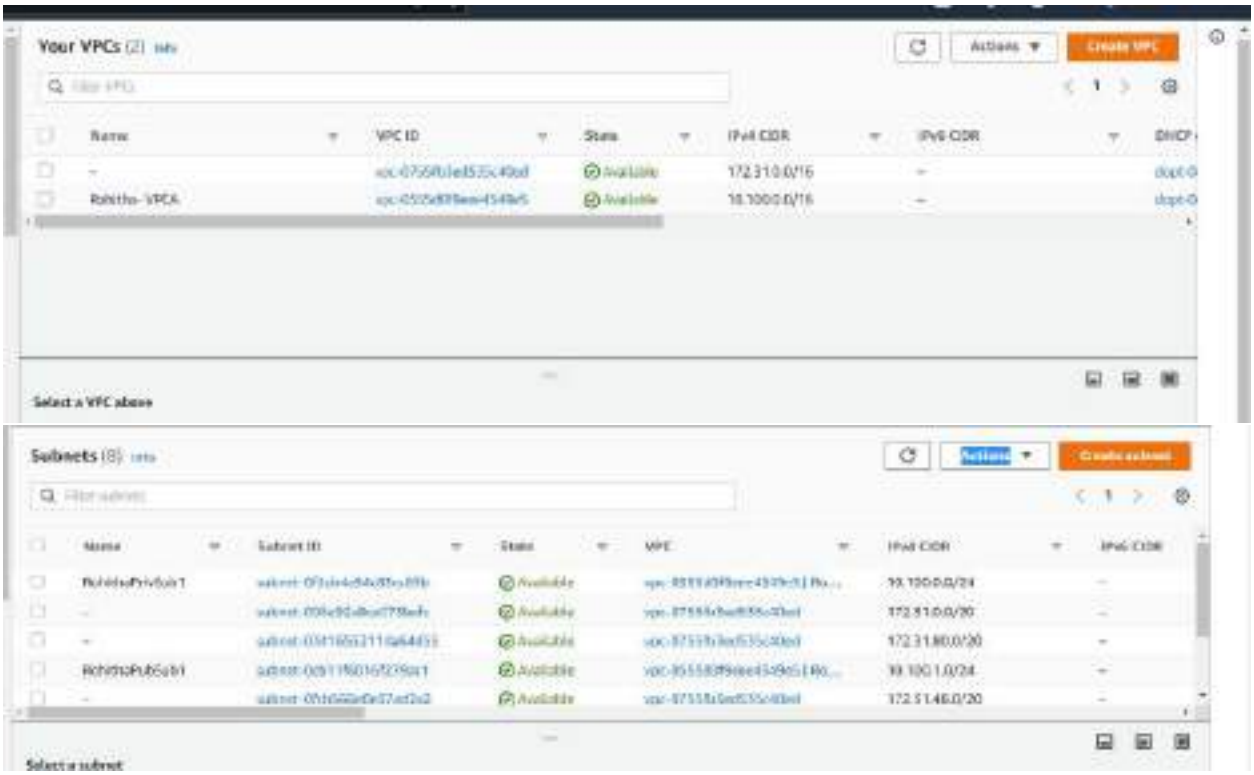


12. Login to private subnet machine and try to ping google.com

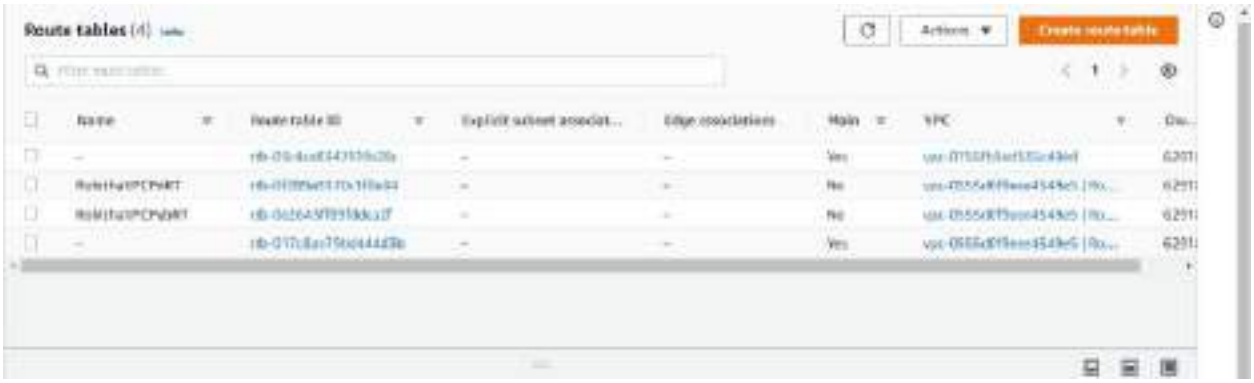


7. VPC Peering :

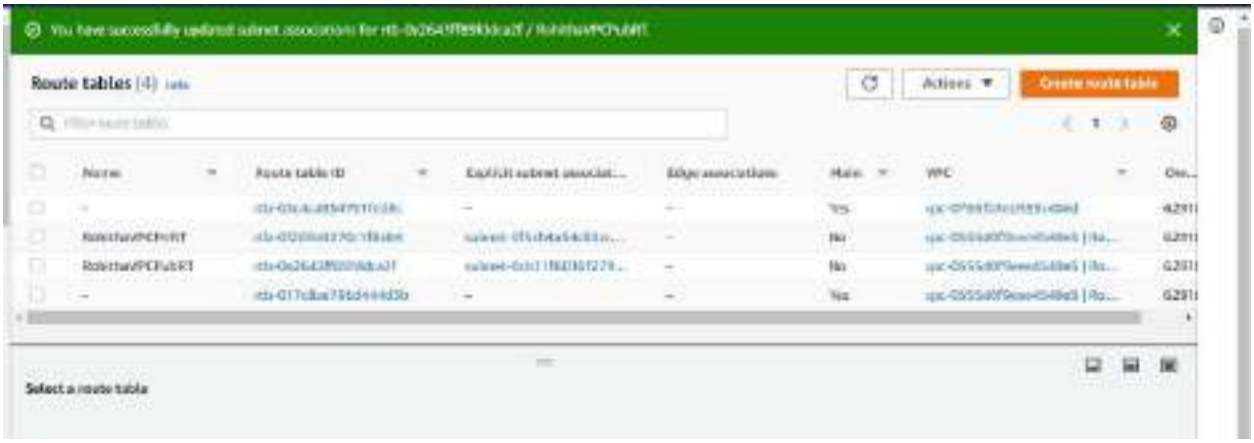
1. Create VPC A and 2 subnets (one is public and one is private)



2. Create 2 route tables (one is public and one is private)

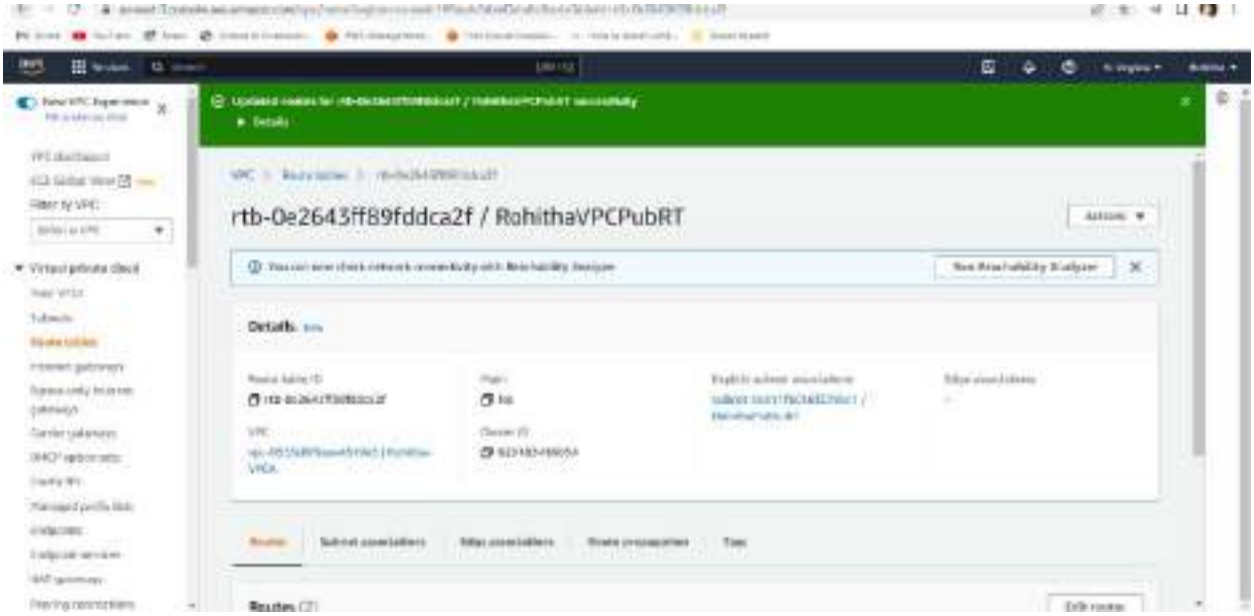


3.Associate public route table with public subnet and private route table with private subnet.

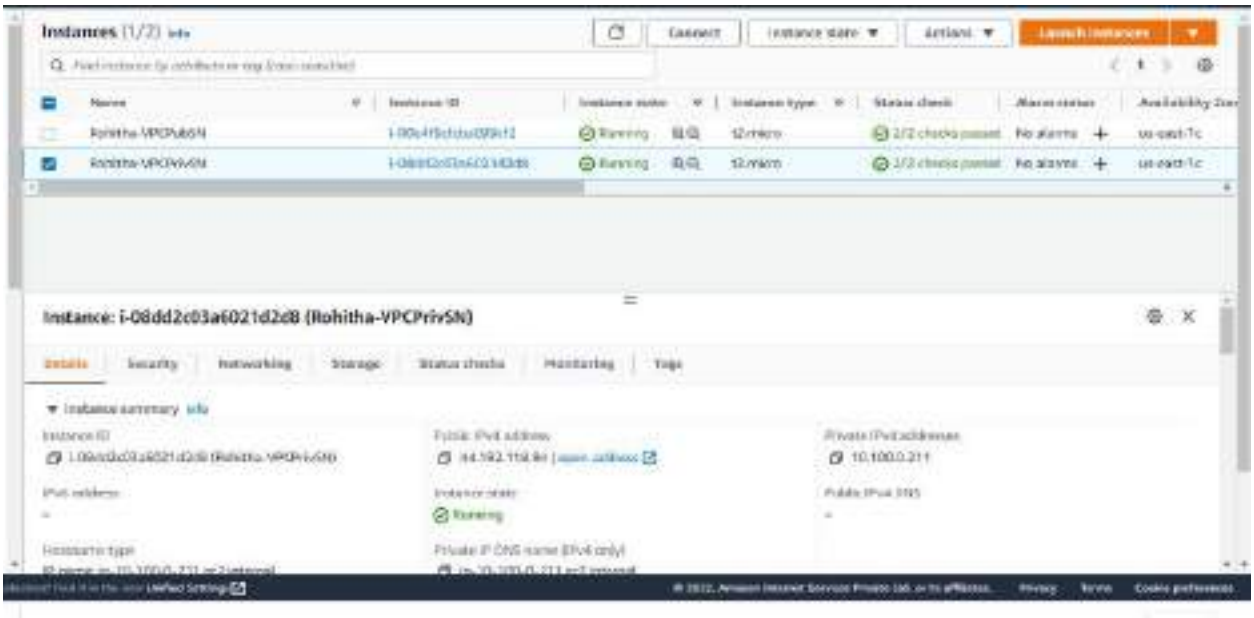


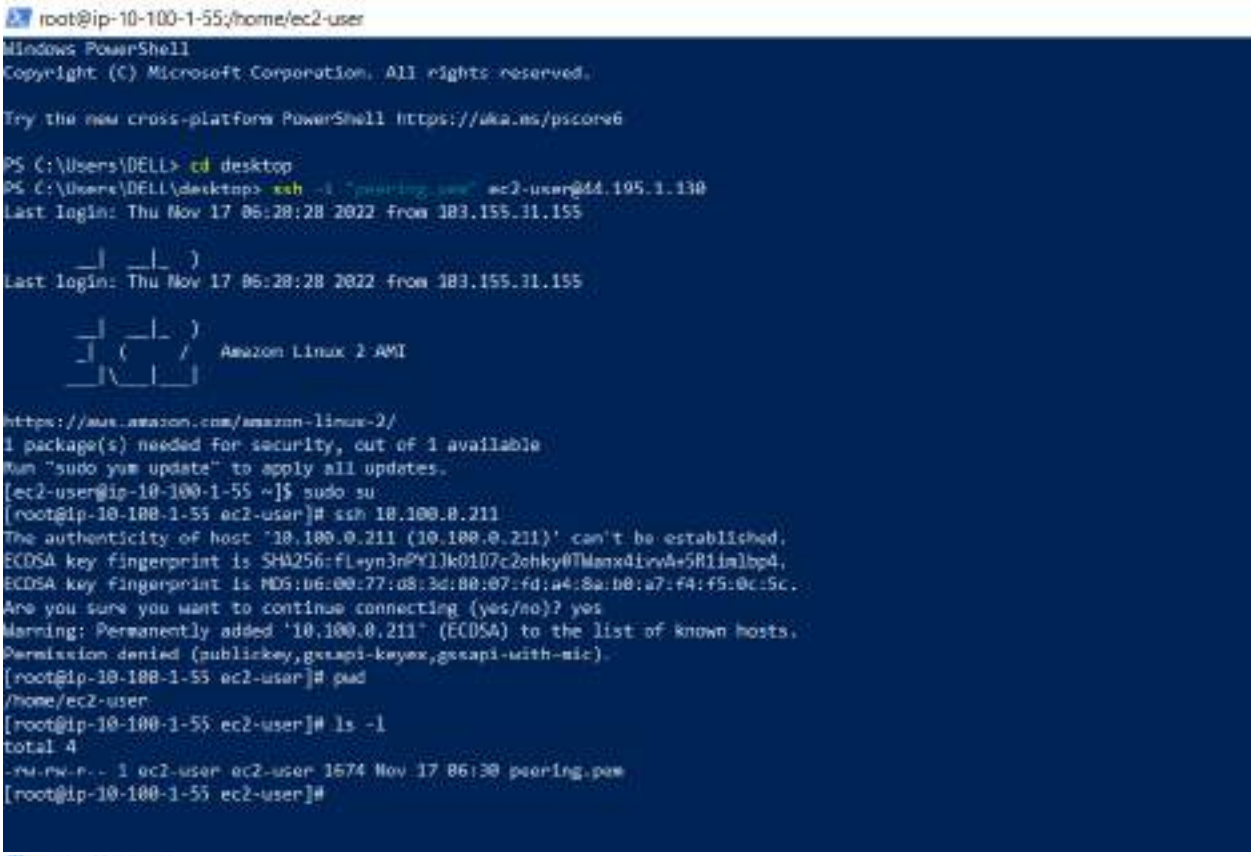
4. Create internet gateway and attach it to VPC.

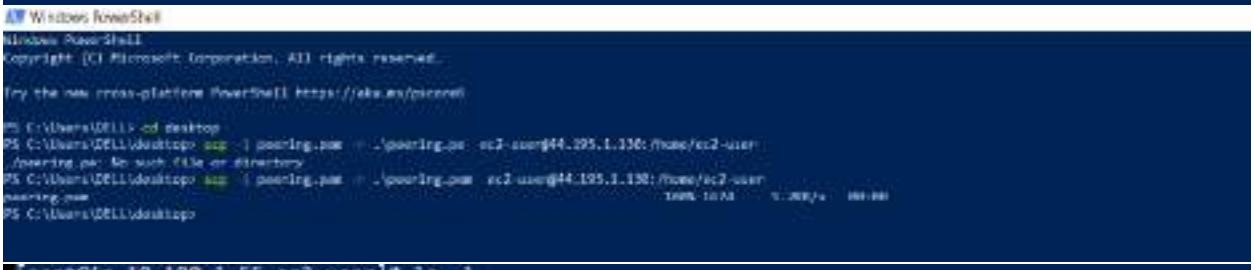
5. Add public subnet to internet gateway.

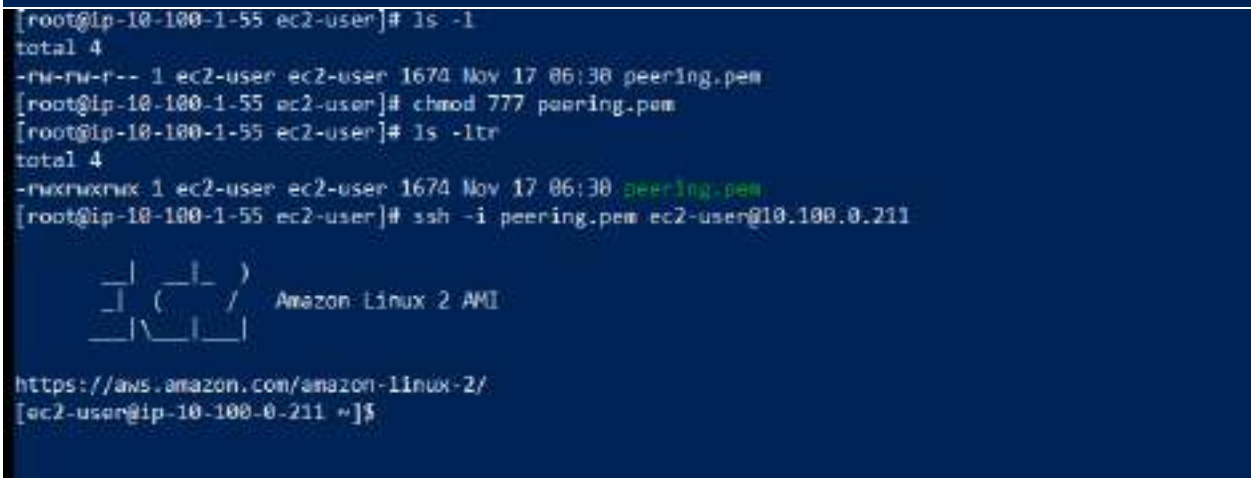


6. Create 2 EC2 machines.

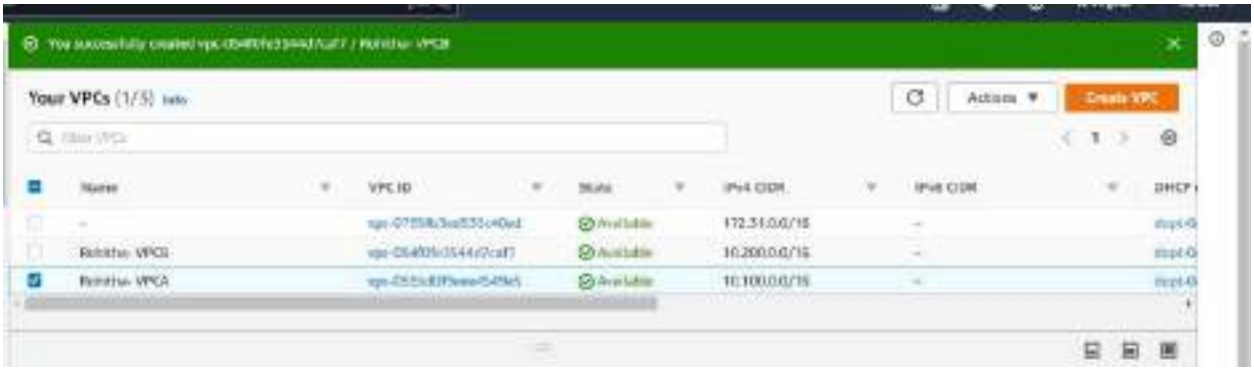




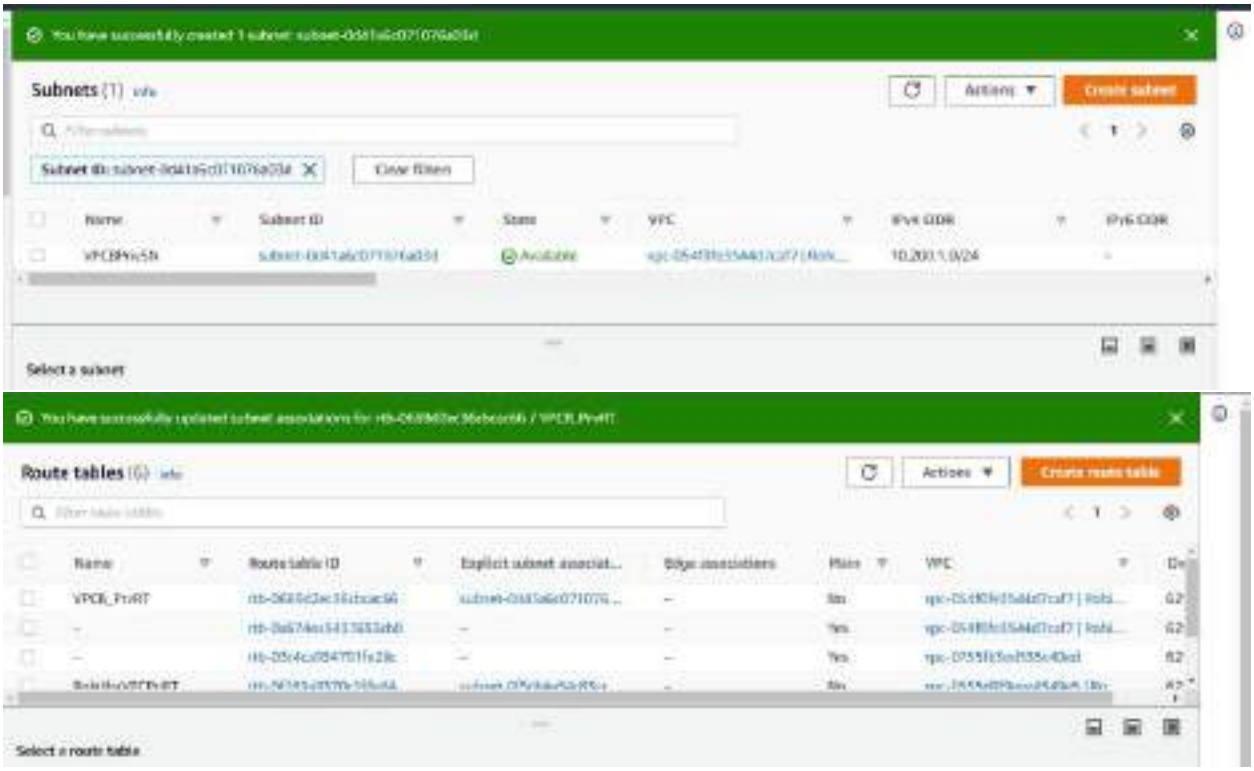




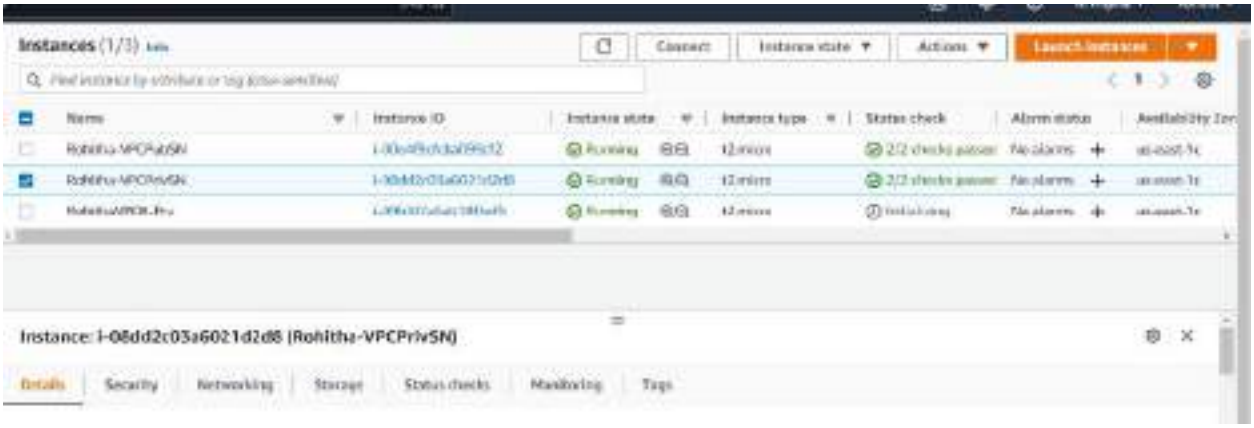
7. Create VPC B machine.



8. Create private subnet and route table.



9. Create EC2 instance.



10. Create peering connection.


```
[root@ip-10-100-0-211 ec2-user]# ssh ec2-user@10.200.1.174
ssh: connect to host 10.200.1.174 port 22: Connection timed out
[root@ip-10-100-0-211 ec2-user]# exit
exit
[ec2-user@ip-10-100-0-211 ~]$ exit
logout
Connection to 10.100.0.211 closed.
[root@ip-10-100-1-55 ec2-user]# ls -ltr
total 4
-rwxrwxrwx 1 ec2-user ec2-user 1674 Nov 17 06:30 peering.pem
[root@ip-10-100-1-55 ec2-user]# scp -i peering.pem -r peering.pem ec2-user@10.100.0.211:/home/ec2-user
peering.pem
[root@ip-10-100-1-55 ec2-user]# ssh -i peering.pem ec2-user@10.100.0.211
Last login: Thu Nov 17 16:26:50 2022 from 10.100.1.55

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-100-0-211 ~]$ sudo su
[root@ip-10-100-0-211 ec2-user]# ls -ltr
total 4
-rwxrwxr-x 1 ec2-user ec2-user 1674 Nov 17 16:58 peering.pem
[root@ip-10-100-0-211 ec2-user]# ssh -i peering.pem ec2-user@10.200.1.174
```