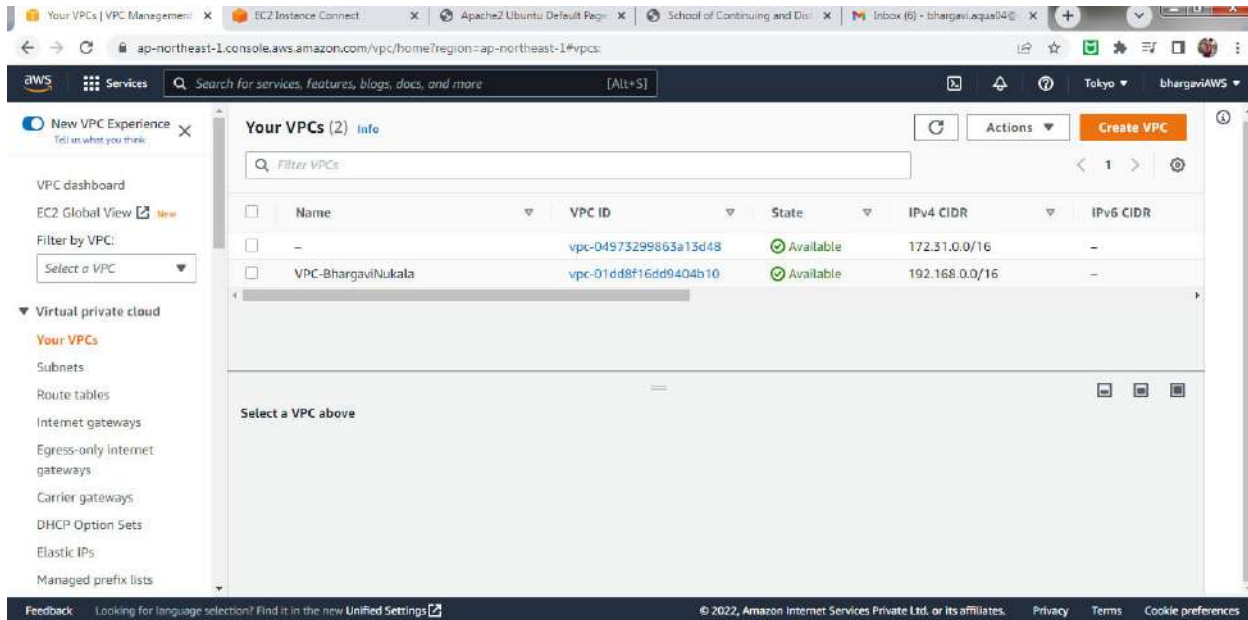
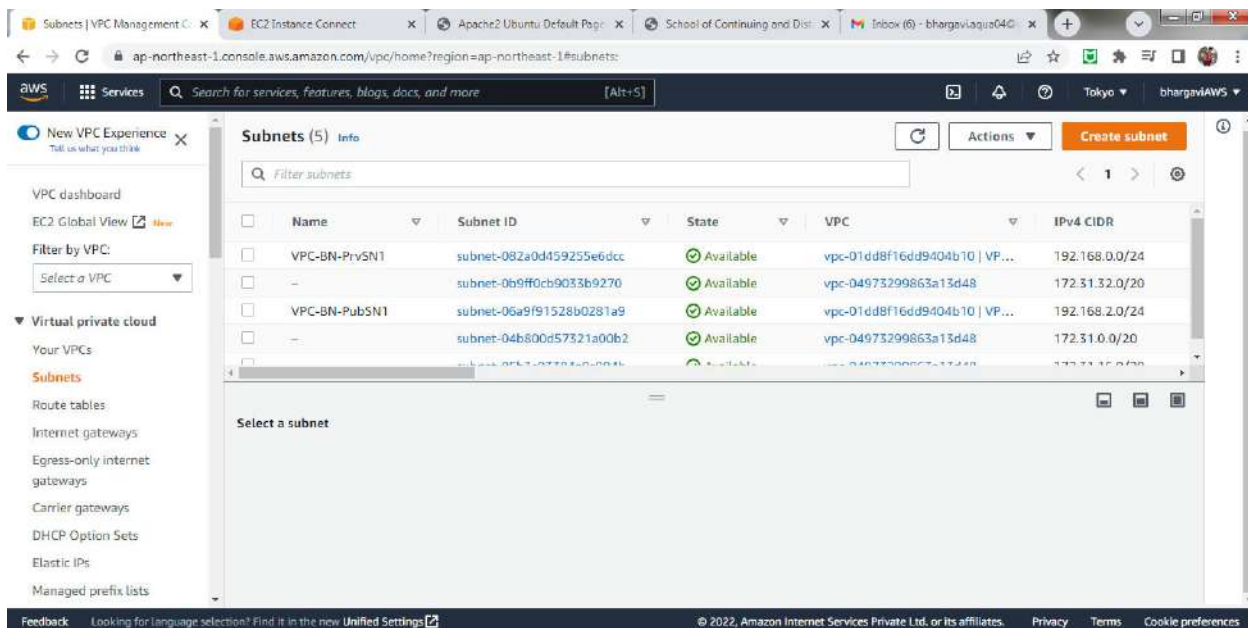


VPC:

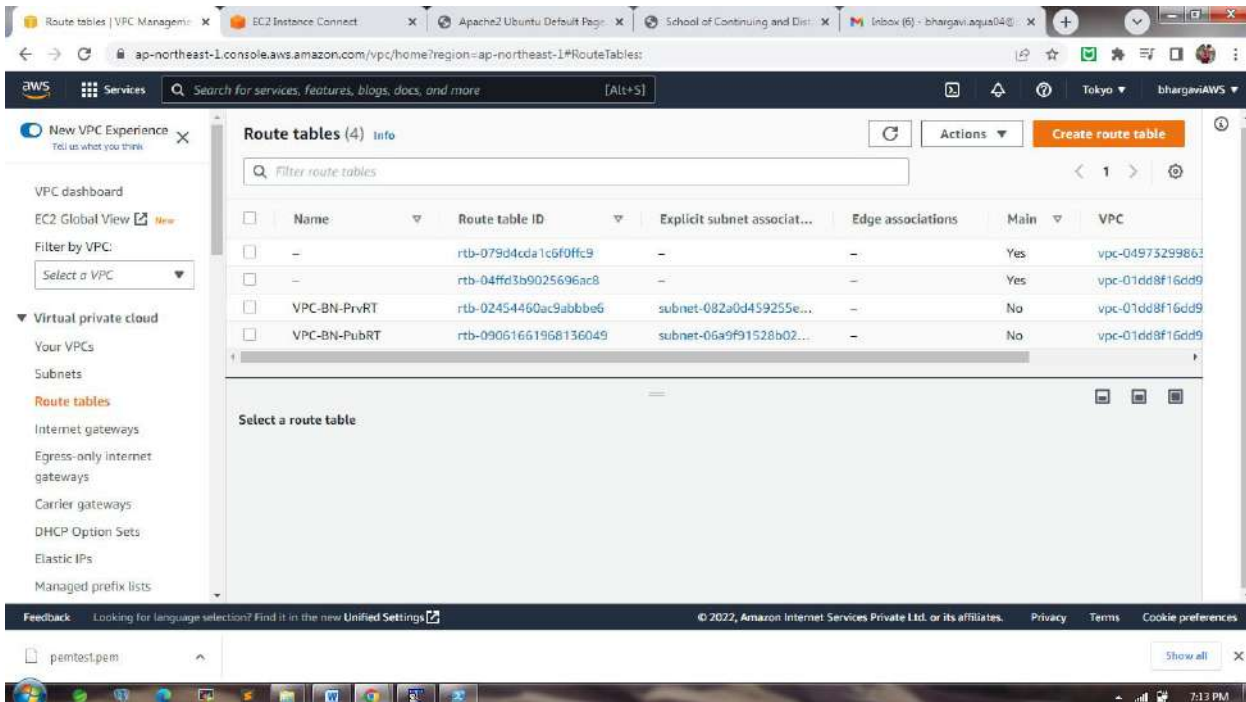
1. Create a new VPC and provided range of VPC as 192.168.0.0/16



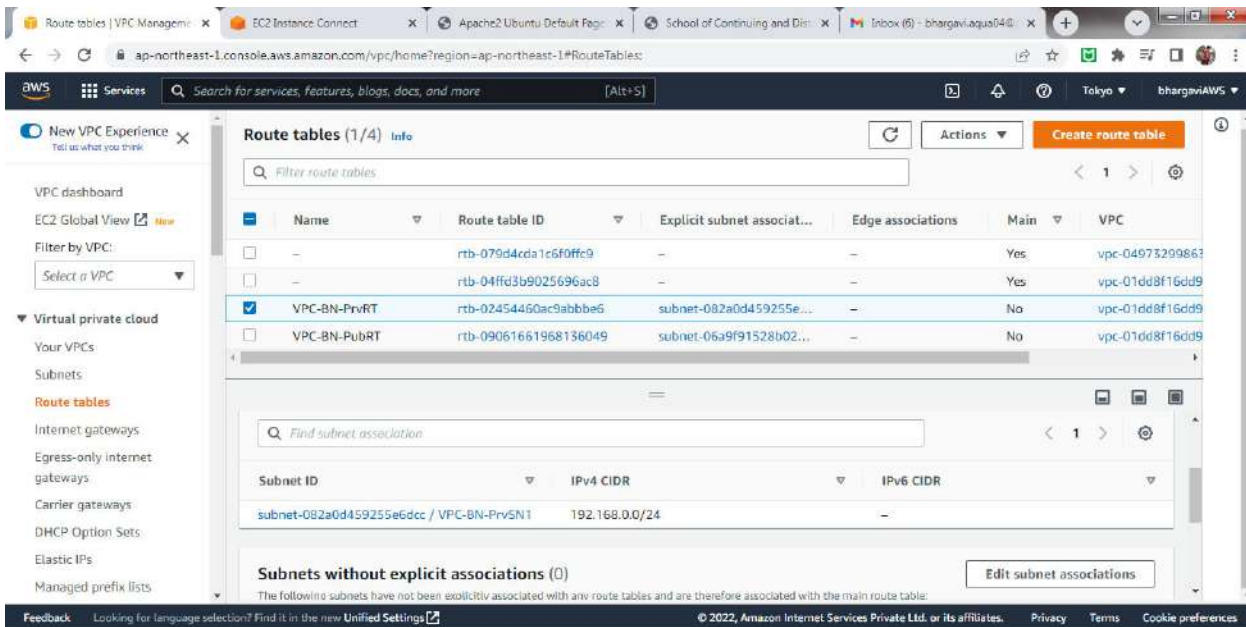
2. Create two subnets, one as Public subnet and one as Private Subnet in our VPC

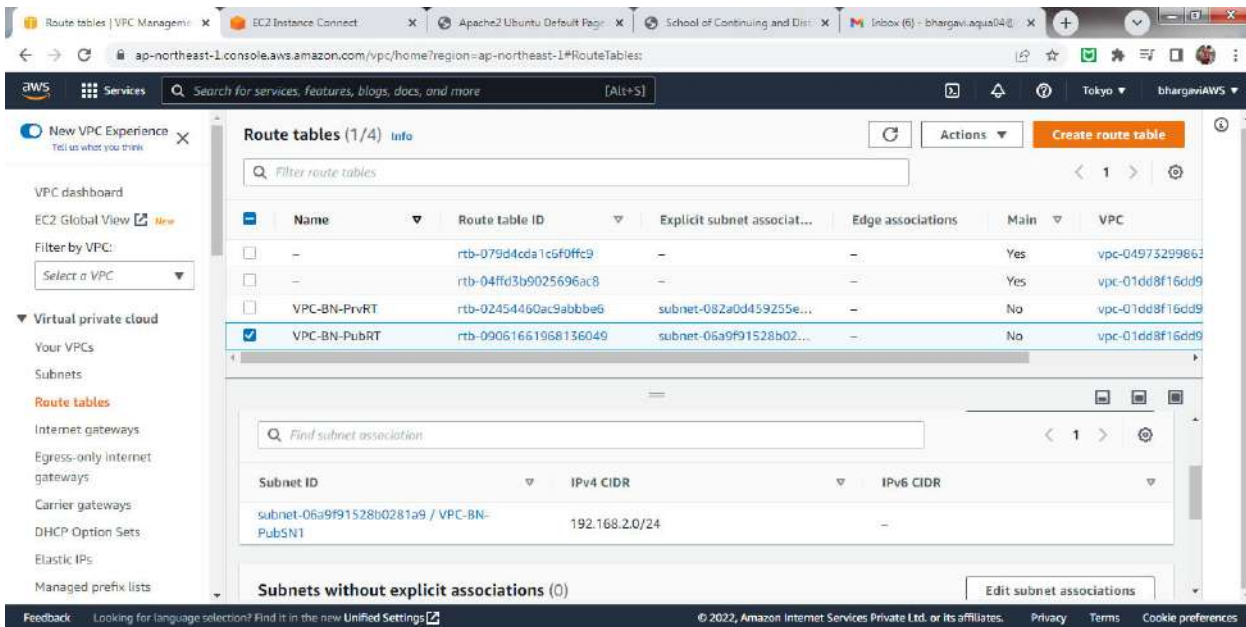


3. Create two Route Tables, one for Public Subnet and One for Private Subnet

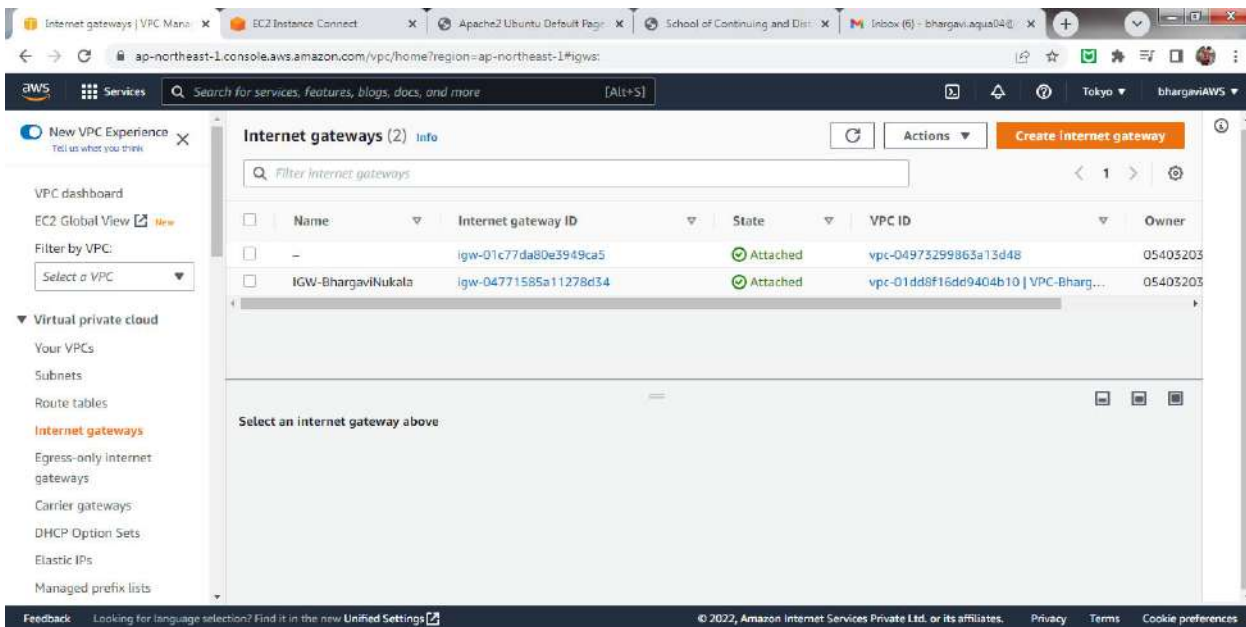


4. Associate Private Subnet to Private Route Table and Public subnet to Public Route Table





5. Create an Internet Gateway



6. Connect Public Route Table to Internet Gateway

The screenshot shows the AWS Route Tables console. The main table lists route tables, with 'VPC-BN-PubRT' selected. Below it, the routes table shows two entries: one for destination 0.0.0.0/0 pointing to Internet Gateway 'igw-04771585a11278d34' (Active), and another for destination 192.168.0.0/16 pointing to 'local' (Active).

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
-	rtb-079d4cda1c6f0fc9	-	-	Yes	vpc-04973299863
-	rtb-04ff3b9025696ac8	-	-	Yes	vpc-01dd8f16dd9
VPC-BN-PrivRT	rtb-02454460ac9abbbe6	subnet-082a0d459255e...	-	No	vpc-01dd8f16dd9
VPC-BN-PubRT	rtb-09061661968136049	subnet-06a9f91528b02...	-	No	vpc-01dd8f16dd9

Destination	Target	Status	Propagated
0.0.0.0/0	igw-04771585a11278d34	Active	No
192.168.0.0/16	local	Active	No

7. Created two instances in VPC, one in Private Subnet and one in Public subnet

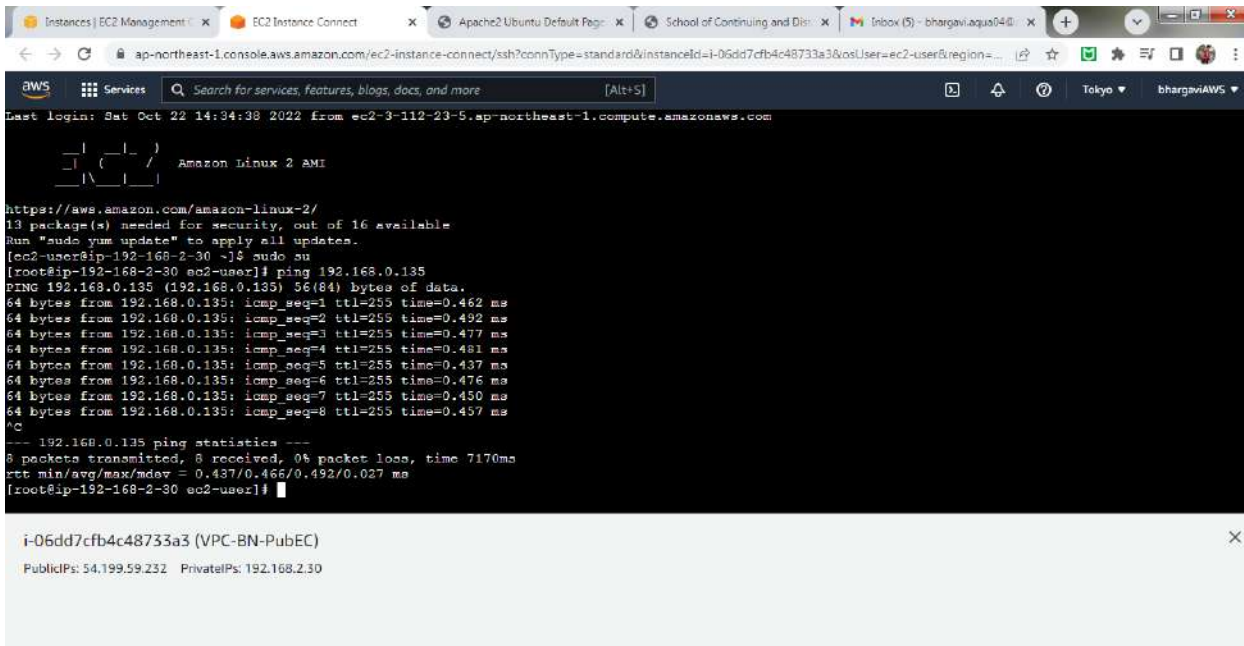
The screenshot shows the AWS EC2 console with two instances running. A monitoring dashboard is visible at the bottom, showing CPU utilization and status check failures.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
VPC-BN-PubEC	i-06dd7cfb4c48733a3	Running	t2.micro	2/2 checks passed	No alarms
VPC-BN-PrivEC	i-061eccdd7c87252768	Running	t2.micro	2/2 checks passed	No alarms

Monitoring dashboard for instances: i-0b012bbeb73e03088 (BhargaviNukala-AMI Instance), i-050a424927fc315f3 (Lakshmi Bhargavi Nukala)

Metric	Value
CPU utilization (%)	3.87
Status check failed (an...)	1
Status check failed (ins...)	1
Status check failed (sys...)	1

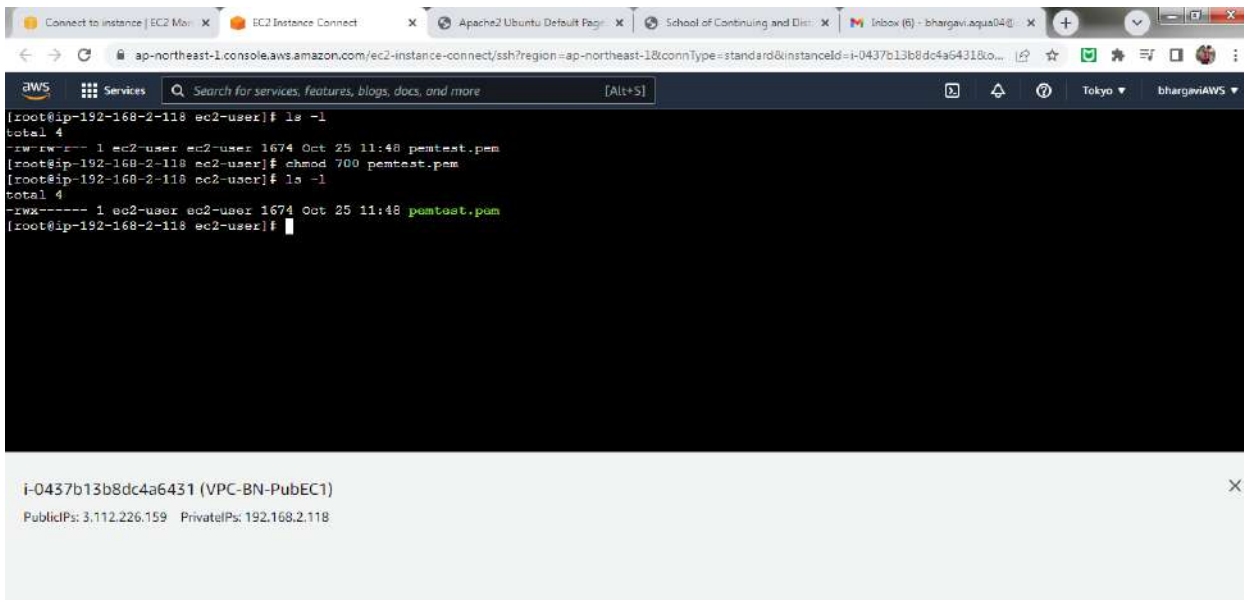
8. Login to Instance in Public subnet and try to Ping instance in Private subnet from Instance in Public subnet



```
ap-northeast-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-06dd7cfb4c48733a3&osUser=ec2-user&region=...
Last login: Sat Oct 22 14:34:38 2022 from ec2-3-112-23-5.ap-northeast-1.compute.amazonaws.com
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
13 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-2-30 ~]$ sudo su
[root@ip-192-168-2-30 ec2-user]# ping 192.168.0.135
PING 192.168.0.135 (192.168.0.135) 56(84) bytes of data:
64 bytes from 192.168.0.135: icmp_seq=1 ttl=255 time=0.462 ms
64 bytes from 192.168.0.135: icmp_seq=2 ttl=255 time=0.492 ms
64 bytes from 192.168.0.135: icmp_seq=3 ttl=255 time=0.477 ms
64 bytes from 192.168.0.135: icmp_seq=4 ttl=255 time=0.481 ms
64 bytes from 192.168.0.135: icmp_seq=5 ttl=255 time=0.437 ms
64 bytes from 192.168.0.135: icmp_seq=6 ttl=255 time=0.476 ms
64 bytes from 192.168.0.135: icmp_seq=7 ttl=255 time=0.450 ms
64 bytes from 192.168.0.135: icmp_seq=8 ttl=255 time=0.457 ms
^C
--- 192.168.0.135 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7170ms
rtt min/avg/max/mdev = 0.437/0.466/0.492/0.027 ms
[root@ip-192-168-2-30 ec2-user]#
```

i-06dd7cfb4c48733a3 (VPC-BN-PubEC)
PublicIps: 54.199.59.232 PrivateIps: 192.168.2.30

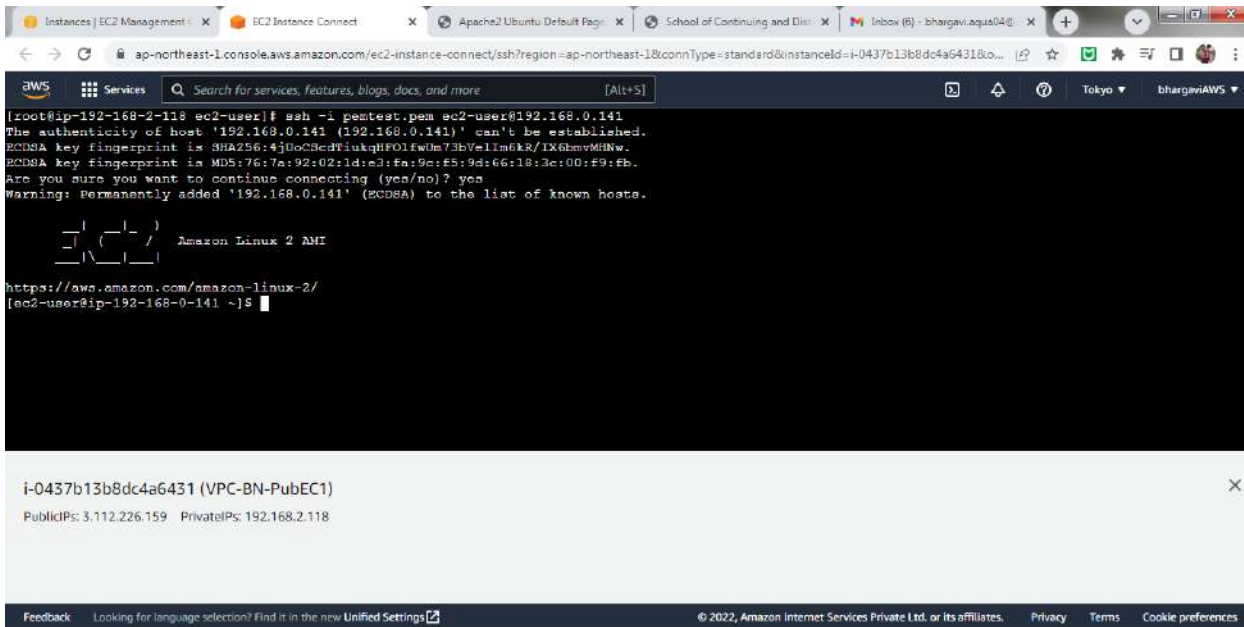
9. Ssh to private instance will fail, so Copied the pemtest.pem file onto Public EC2 instance and changed the permissions



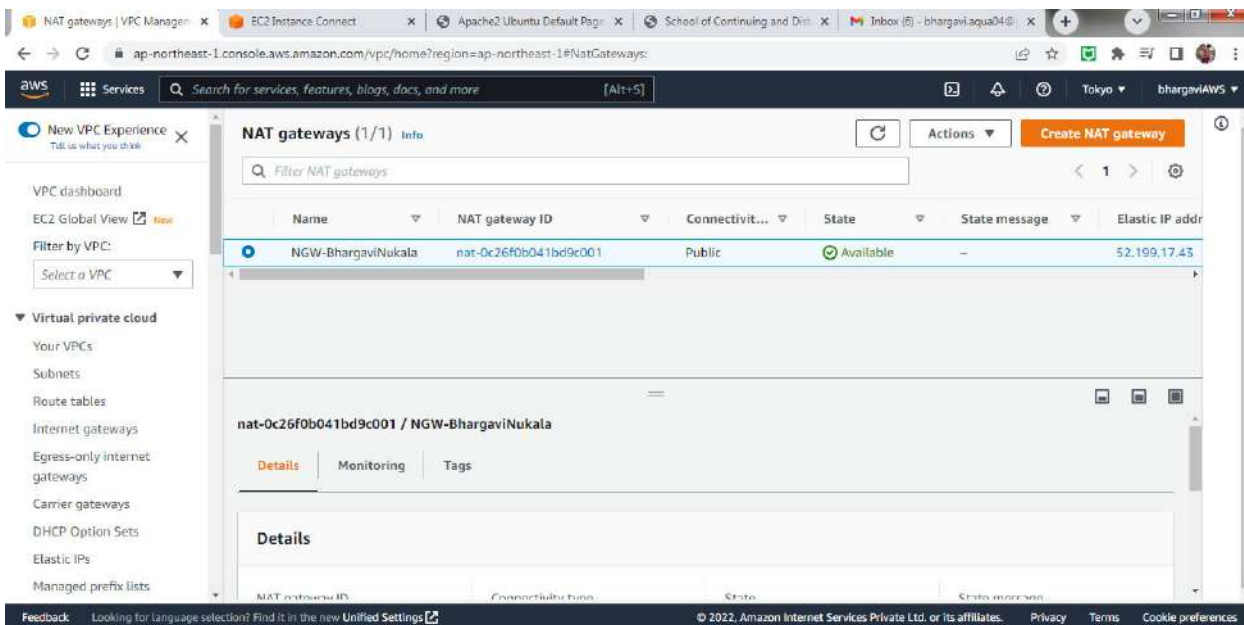
```
Connect to instance | EC2 Ma... x EC2 Instance Connect x Apache2 Ubuntu Default Page x School of Continuing and Dis... x Inbo... (5) - bhargavi.aqua04@... x
ap-northeast-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-northeast-1&connType=standard&instanceId=i-0437b13b8dc4a6431&o...
[ec2-user@ip-192-168-2-118 ec2-user]# ls -l
total 4
-rw-rw-r-- 1 ec2-user ec2-user 1674 Oct 25 11:48 pemtest.pem
[ec2-user@ip-192-168-2-118 ec2-user]# chmod 700 pemtest.pem
[ec2-user@ip-192-168-2-118 ec2-user]# ls -l
total 4
-rwx----- 1 ec2-user ec2-user 1674 Oct 25 11:48 pumtoat.pem
[ec2-user@ip-192-168-2-118 ec2-user]#
```

i-0437b13b8dc4a6431 (VPC-BN-PubEC1)
PublicIps: 3.112.226.159 PrivateIps: 192.168.2.118

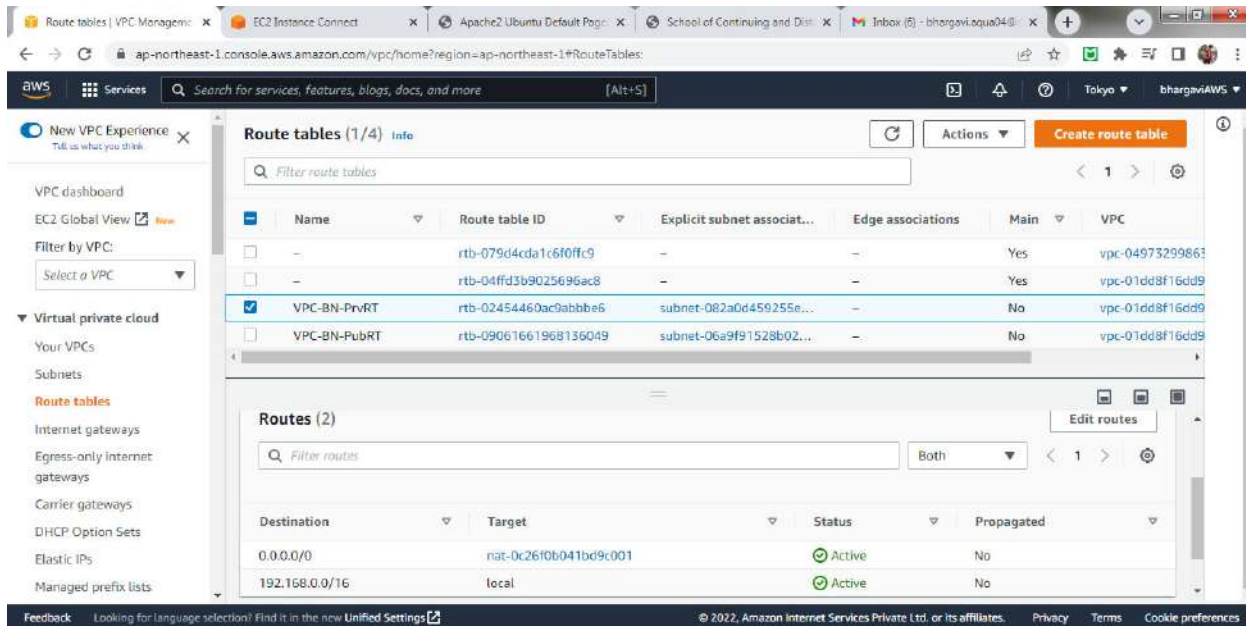
10. Now when we should be able to Connect to a private EC2 instance from a public EC2 instance using ssh



11. Create NAT GateWay to get internet connection to instances on Private Subnet



12. Attaching Private RouteTable to NAT Gateway

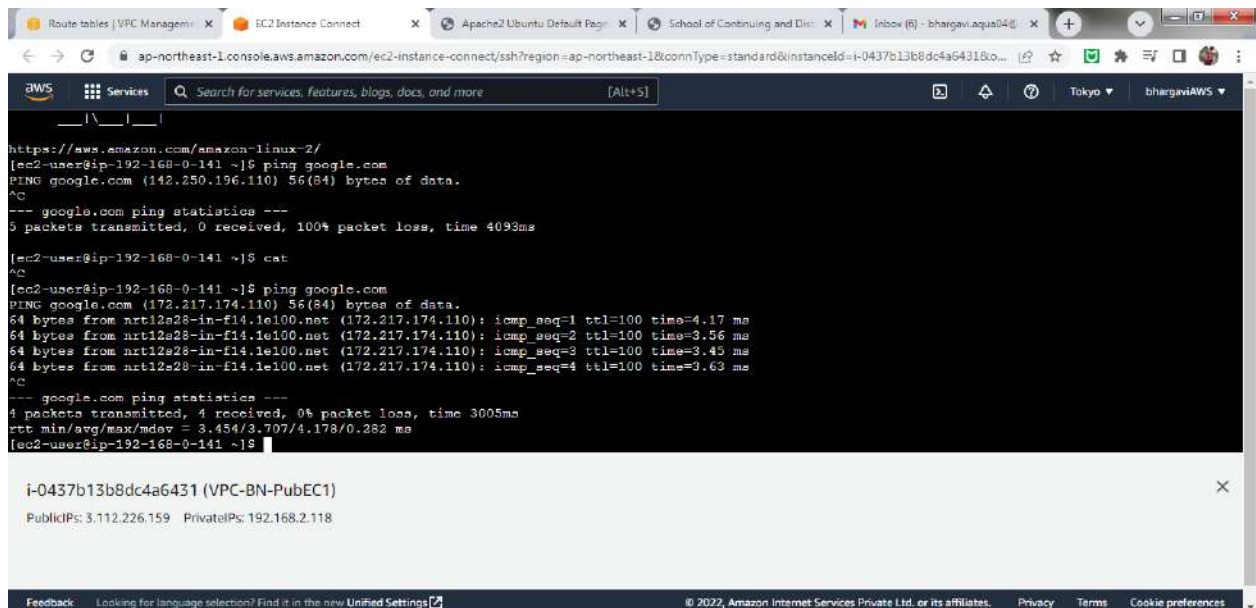


The screenshot shows the AWS Management Console interface for Route Tables. The main content area displays a list of route tables for a specific VPC. The selected route table, 'VPC-BN-PrivRT', is highlighted in blue. Below the list, the 'Routes (2)' section shows two routes: one for destination '0.0.0.0/0' targeting 'nat-0c26f0b041bd9c001' (Active), and another for destination '192.168.0.0/16' targeting 'local' (Active).

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-079d4cda1c6f0fc9	-	-	Yes	vpc-04973299863
-	rtb-04ffd3b9025696ac8	-	-	Yes	vpc-01dd8f16dd9
VPC-BN-PrivRT	rtb-02454460ac9abbb6	subnet-082a0d459255e...	-	No	vpc-01dd8f16dd9
VPC-BN-PubRT	rtb-09061661968136049	subnet-06a9f91528b02...	-	No	vpc-01dd8f16dd9

Destination	Target	Status	Propagated
0.0.0.0/0	nat-0c26f0b041bd9c001	Active	No
192.168.0.0/16	local	Active	No

13. Pinging google.com from instance on Private Subnet



The screenshot shows a terminal window within the AWS Management Console. The terminal output displays the results of a ping test to google.com from an EC2 instance. The first ping attempt from IP 192.168.0-141 fails with 100% packet loss. The second ping attempt from IP 192.168.0-141 succeeds, showing 4 packets received and 0% packet loss.

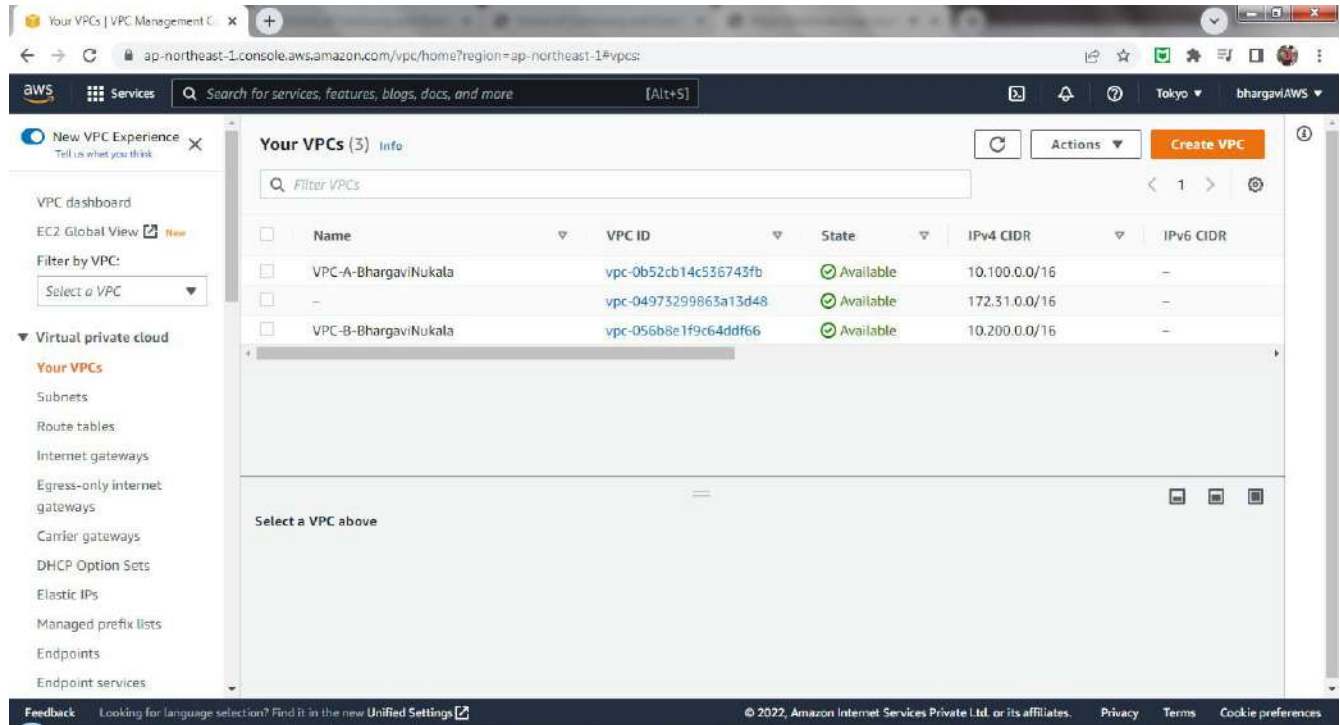
```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-192-168-0-141 ~]$ ping google.com
PING google.com (142.250.196.110) 56(84) bytes of data:
^C
--- google.com ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4093ms

[ec2-user@ip-192-168-0-141 ~]$ cat
^C
[ec2-user@ip-192-168-0-141 ~]$ ping google.com
PING google.com (172.217.174.110) 56(84) bytes of data:
64 bytes from rrt12a28-in-f14.1e100.net (172.217.174.110): icmp_seq=1 ttl=100 time=4.17 ms
64 bytes from rrt12a28-in-f14.1e100.net (172.217.174.110): icmp_seq=2 ttl=100 time=3.56 ms
64 bytes from rrt12a28-in-f14.1e100.net (172.217.174.110): icmp_seq=3 ttl=100 time=3.45 ms
64 bytes from rrt12a28-in-f14.1e100.net (172.217.174.110): icmp_seq=4 ttl=100 time=3.63 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 3.454/3.707/4.178/0.282 ms
[ec2-user@ip-192-168-0-141 ~]$
```

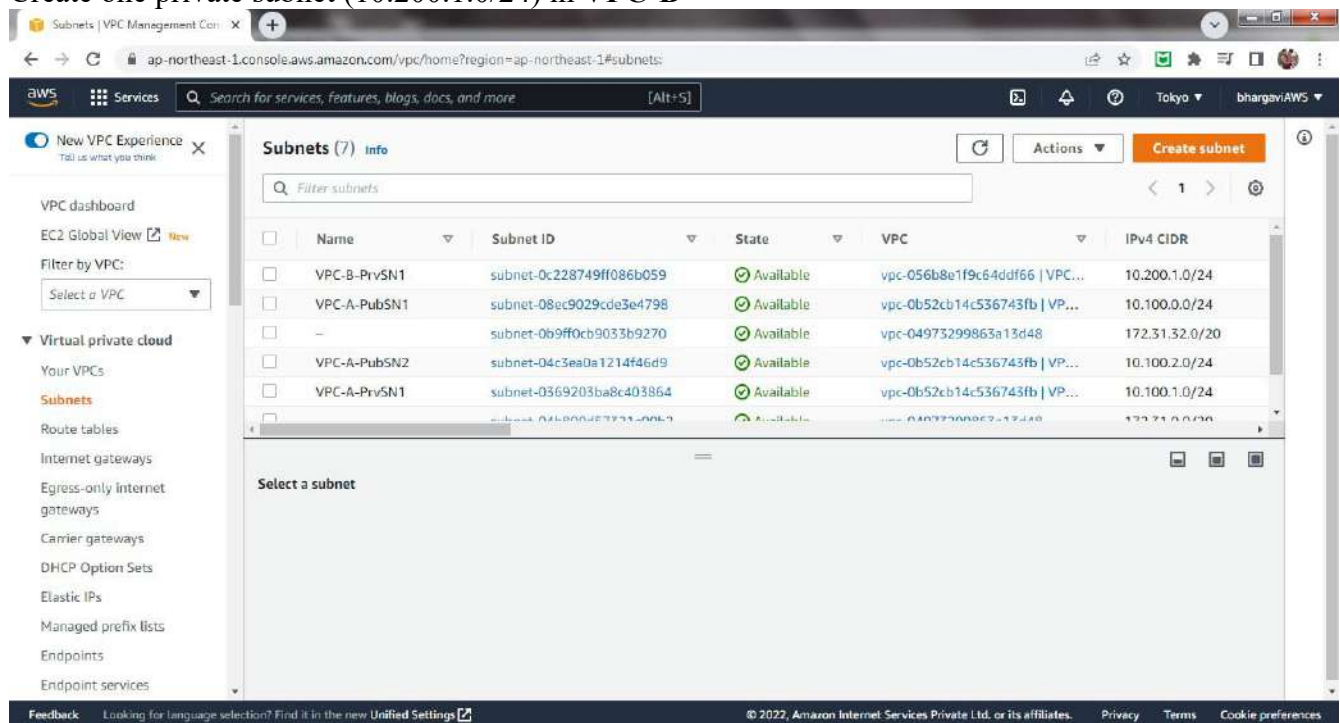
i-0437b13b8dc4a6431 (VPC-BN-PubEC1)
PublicIPs: 3.112.226.159 PrivateIPs: 192.168.2.118

VPC Peering:

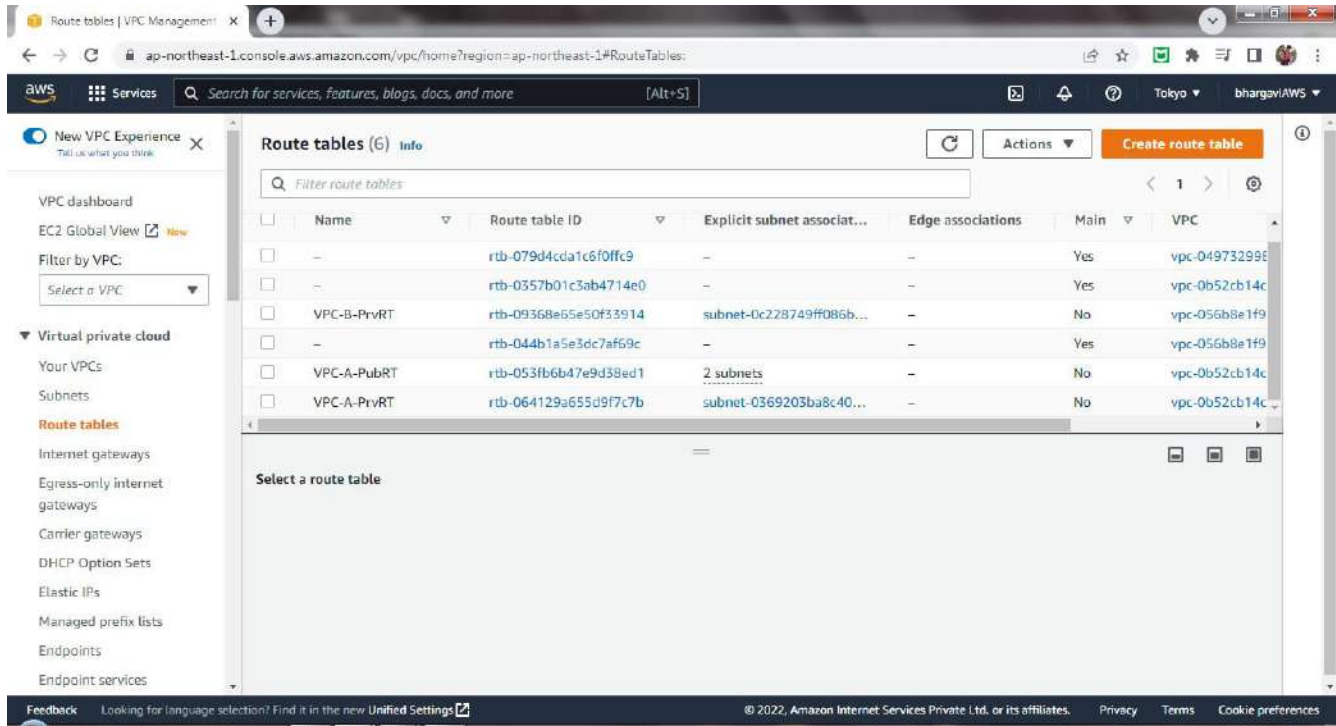
1. Create two VPCs, VPC-A and VPC-B



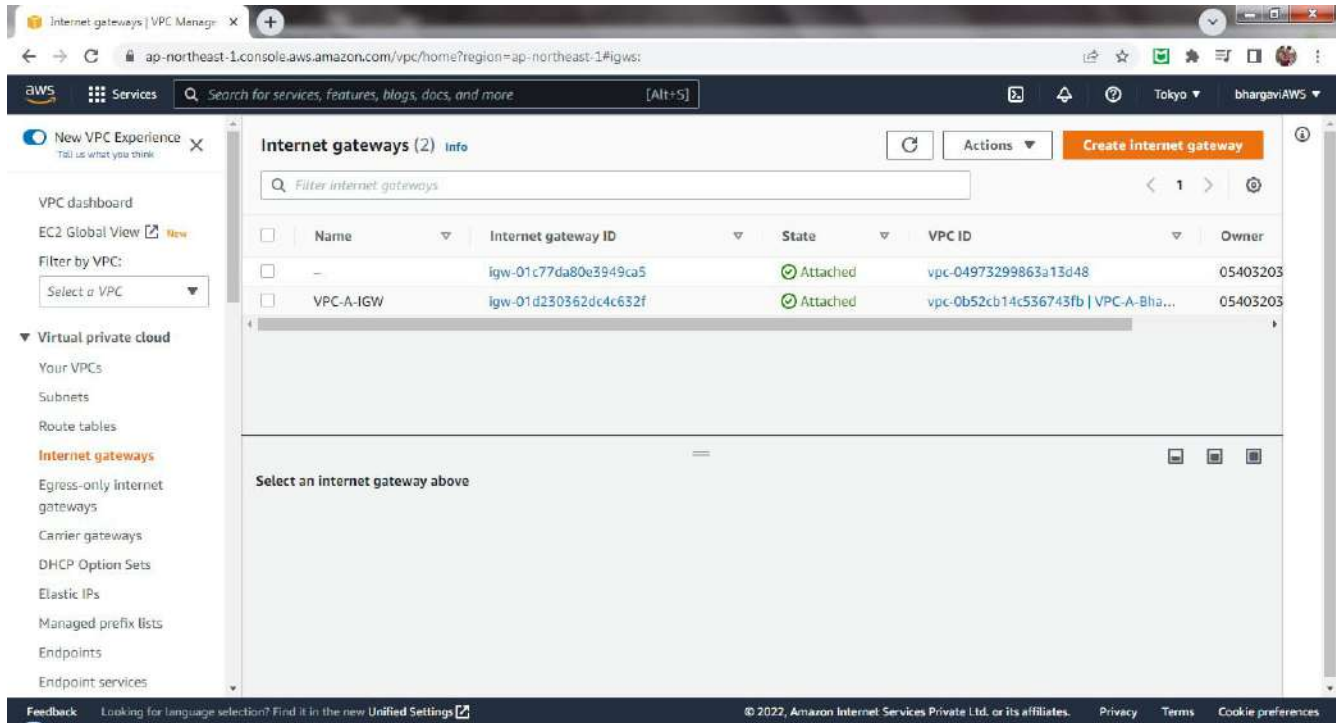
2. Create two public subnets with ranges (10.100.0.0/24) & (10.100.2.0/24) and one private subnet with range (10.100.1.0/24) in VPC-A Create one private subnet (10.200.1.0/24) in VPC-B

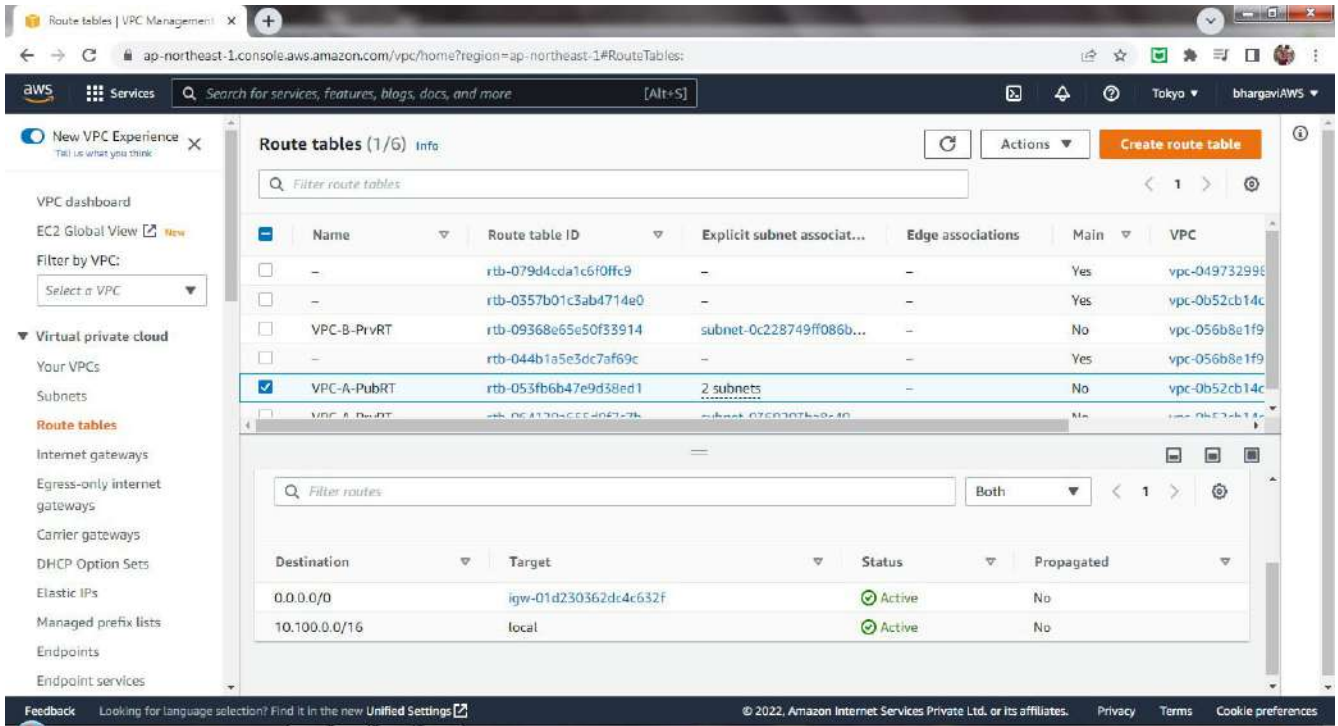


3. Create a private RouteTable for private subnet and public route table for 2 public subnets in **VPC-A**
 Create a private route table for private subnet in **VPC-B**
 Associate the concerned subnets with the route tables.

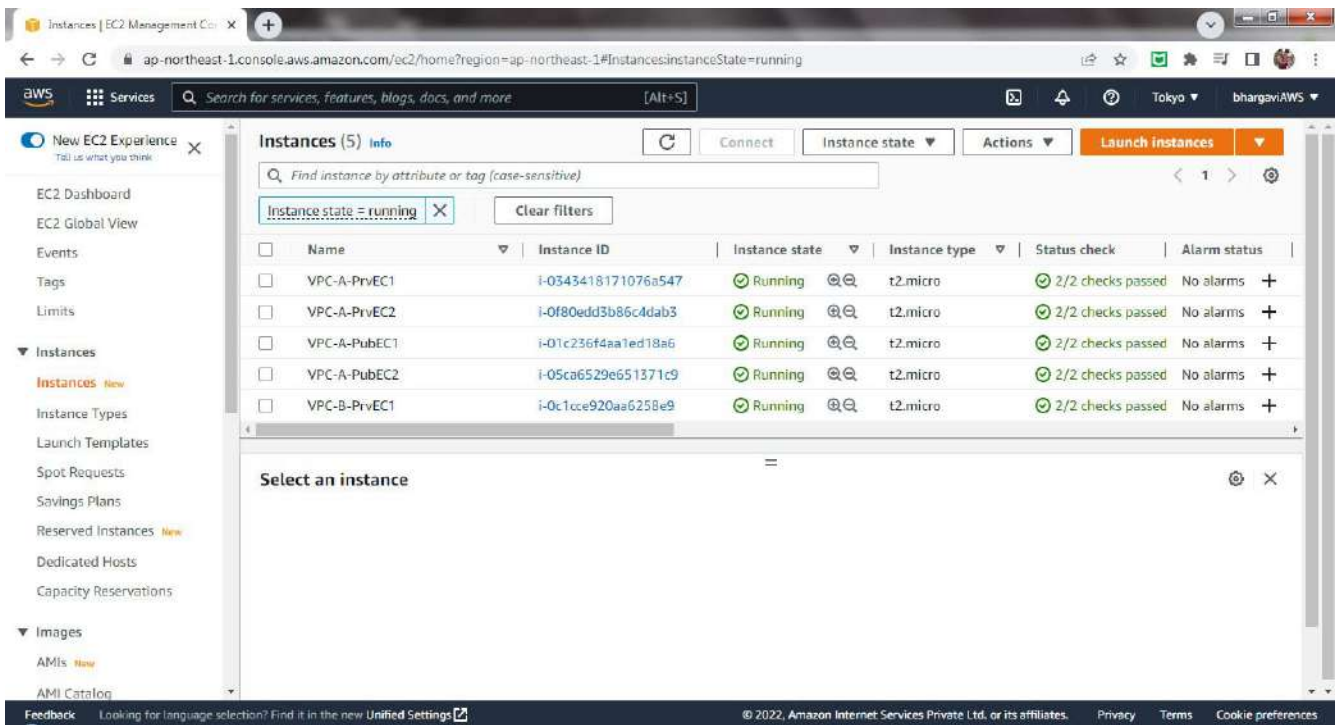


4. Create an Internet Gateway and associate public Route Table of **VPC-A** to that Internet Gateway.

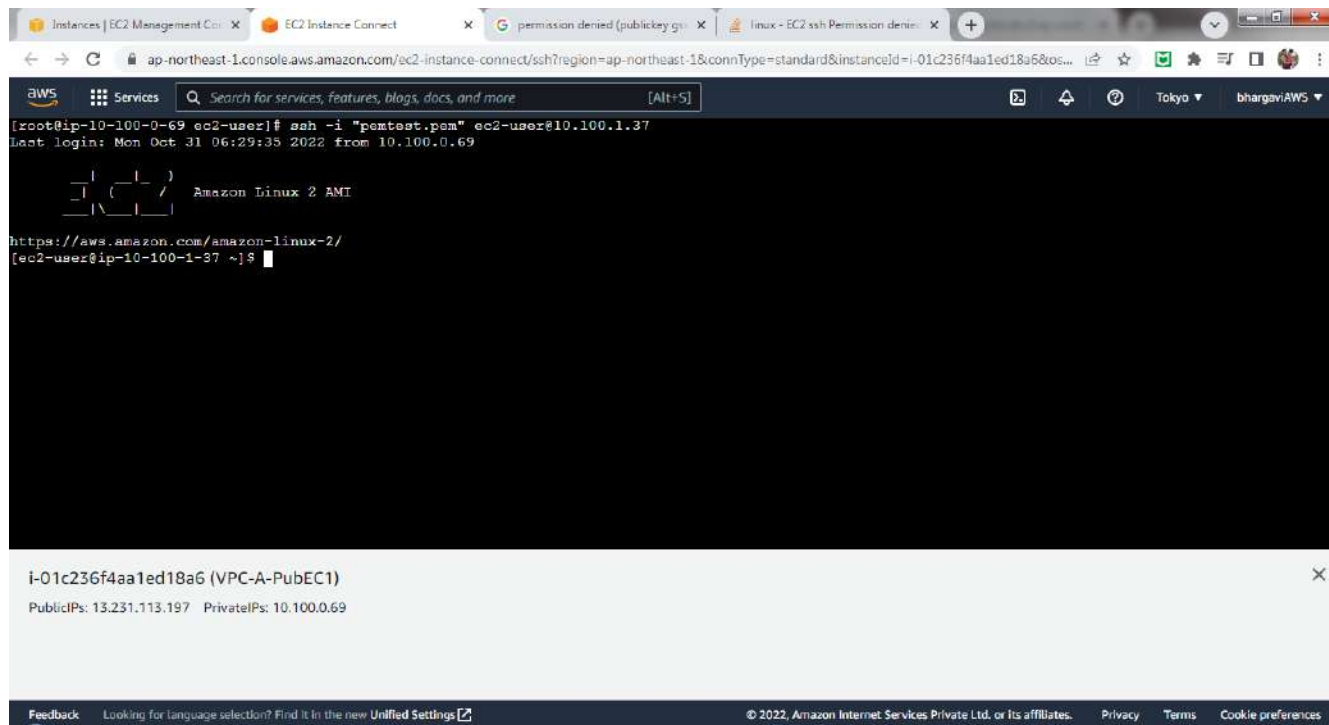




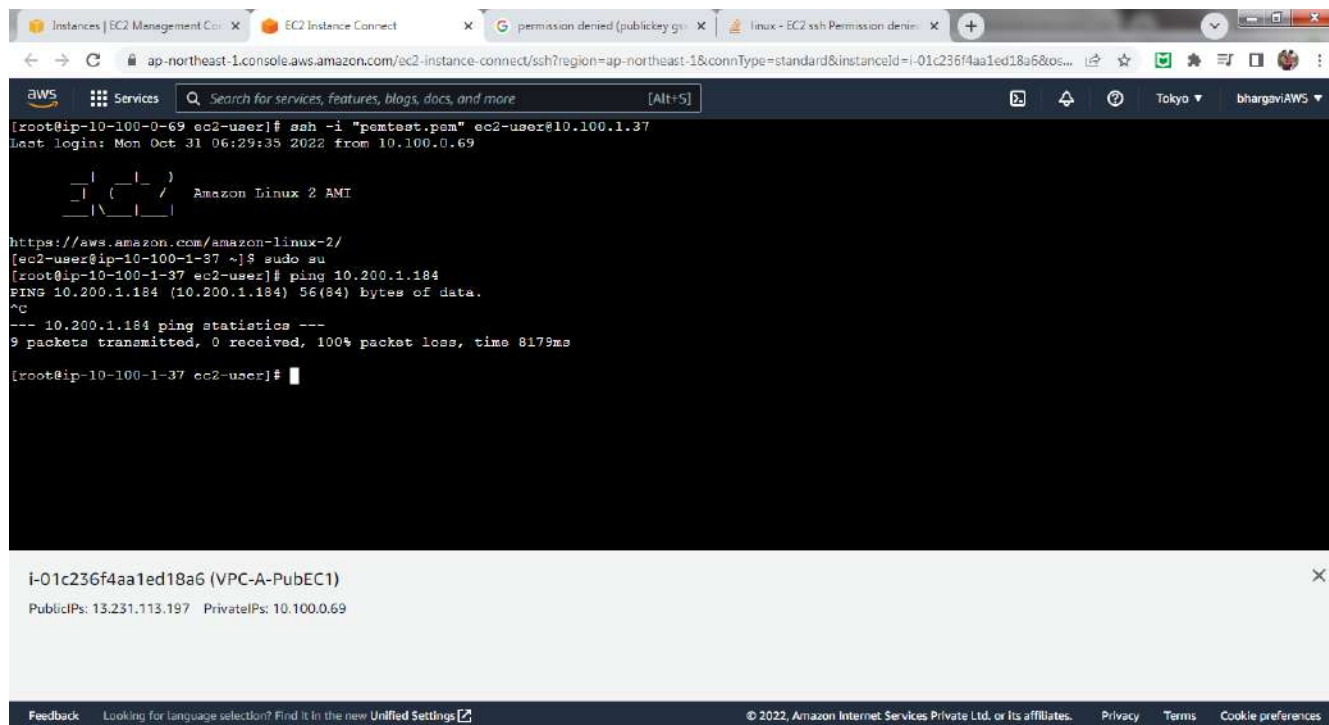
5. Create a EC2 machine in each public subnet & private subnet should have 2 EC2 instance for VPC-A
 A
 create one EC2 instance in private subnetof VPC-B



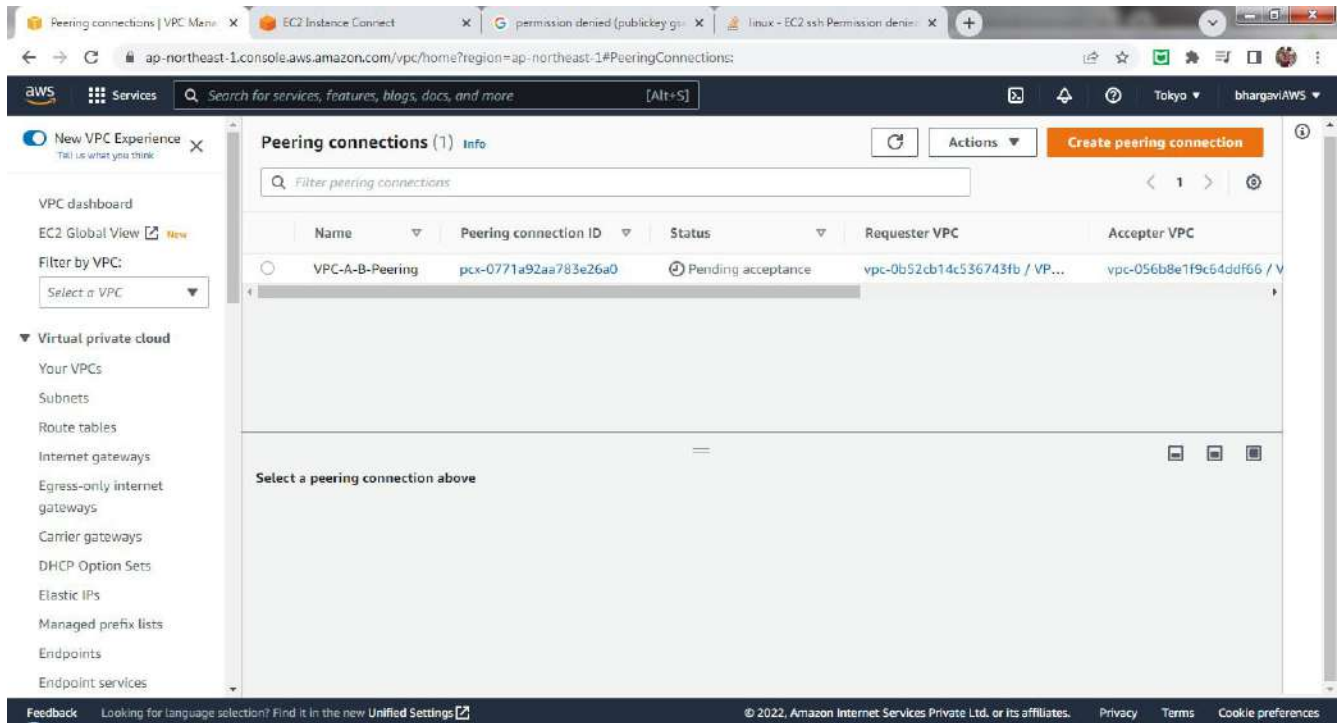
6. Connect to private EC2 instance of VPC-A from public EC2 instance of VPC-A



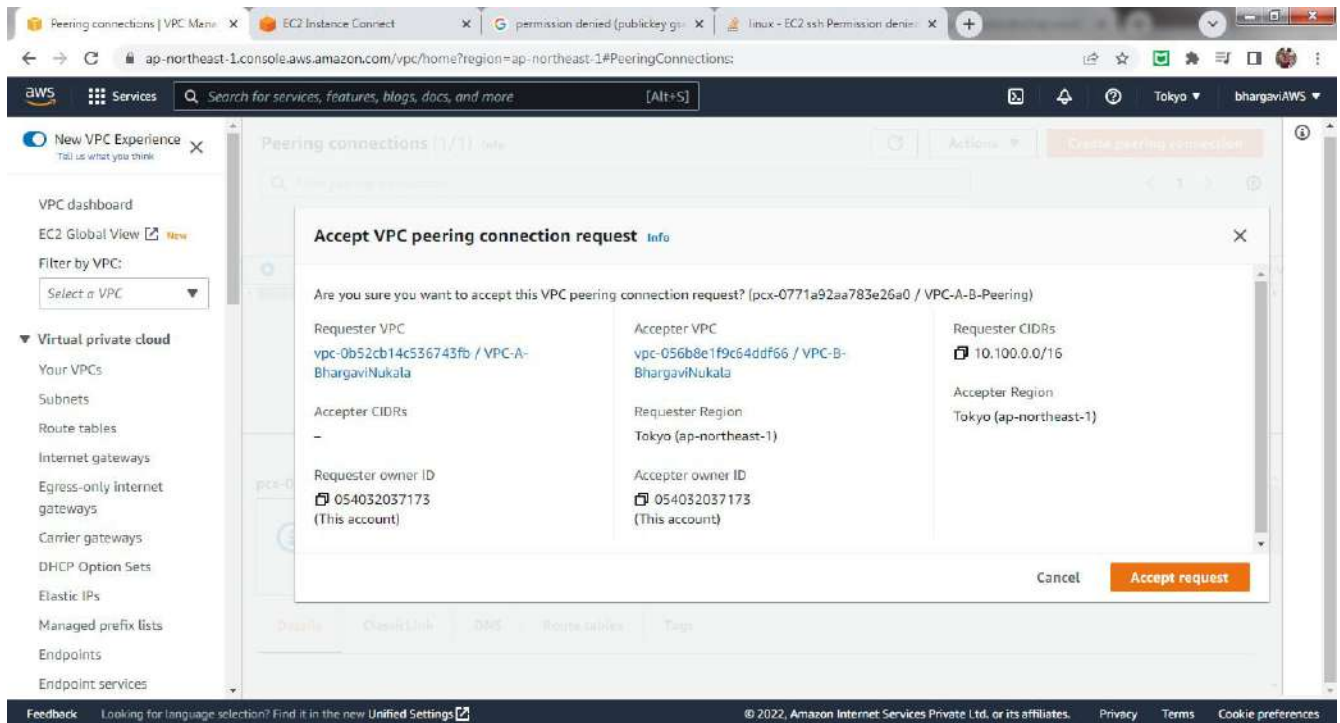
7. From this private instance of VPC-A, try to ping private instance of VPC-B. It will fail as there is no internet connection to VPC-B



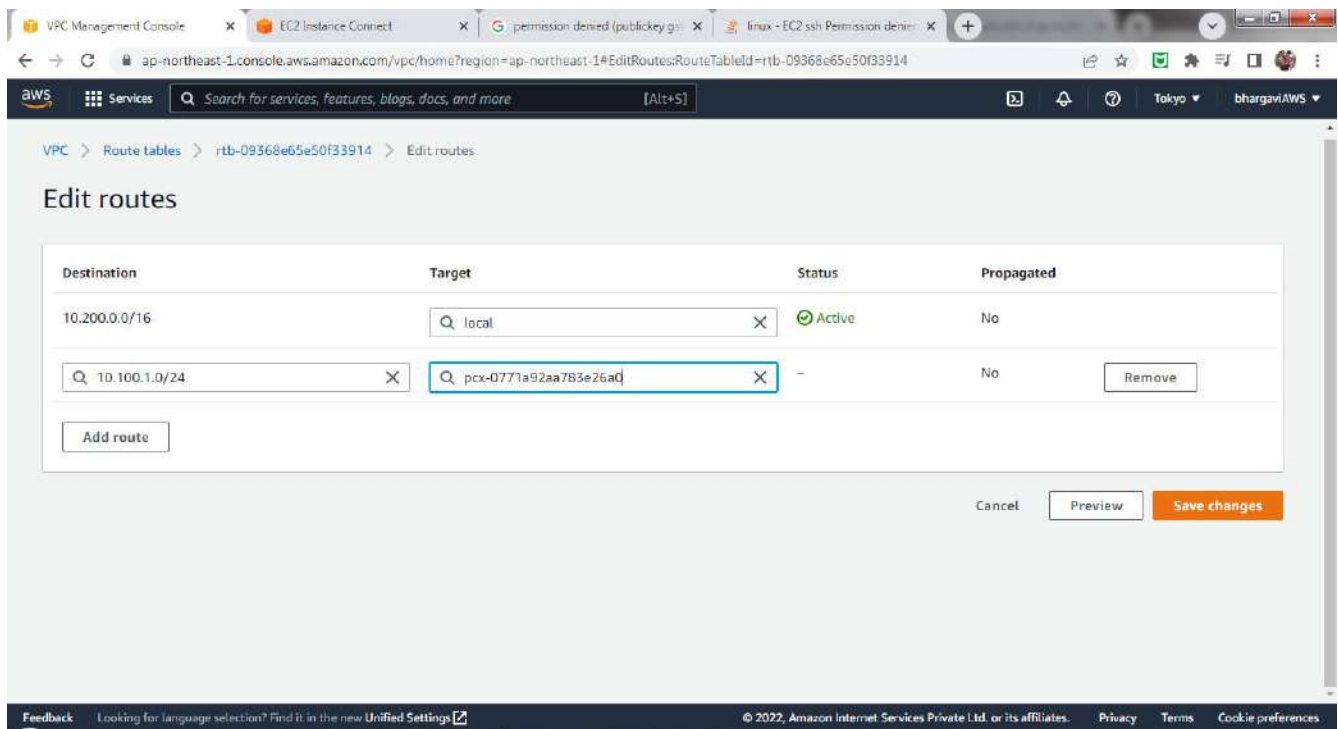
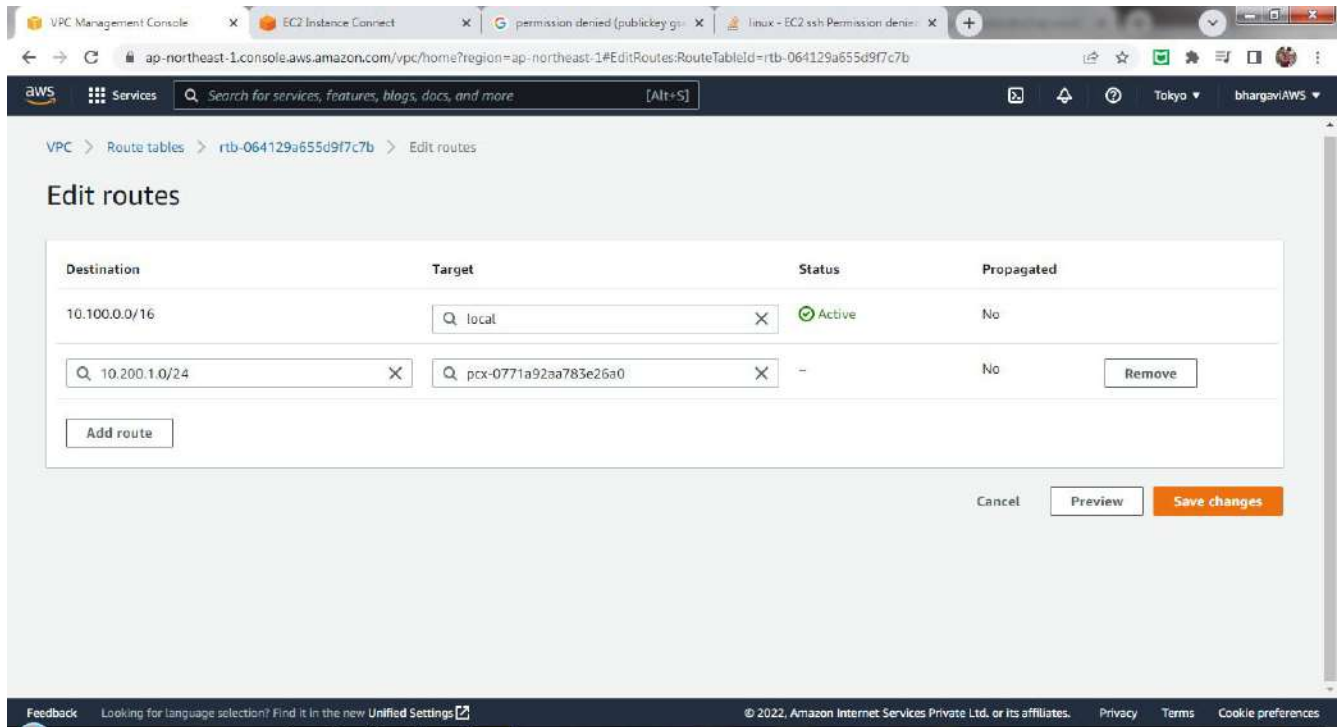
8. So for communication between two private instances of VPC-A and VPC-B, create a peer connection.



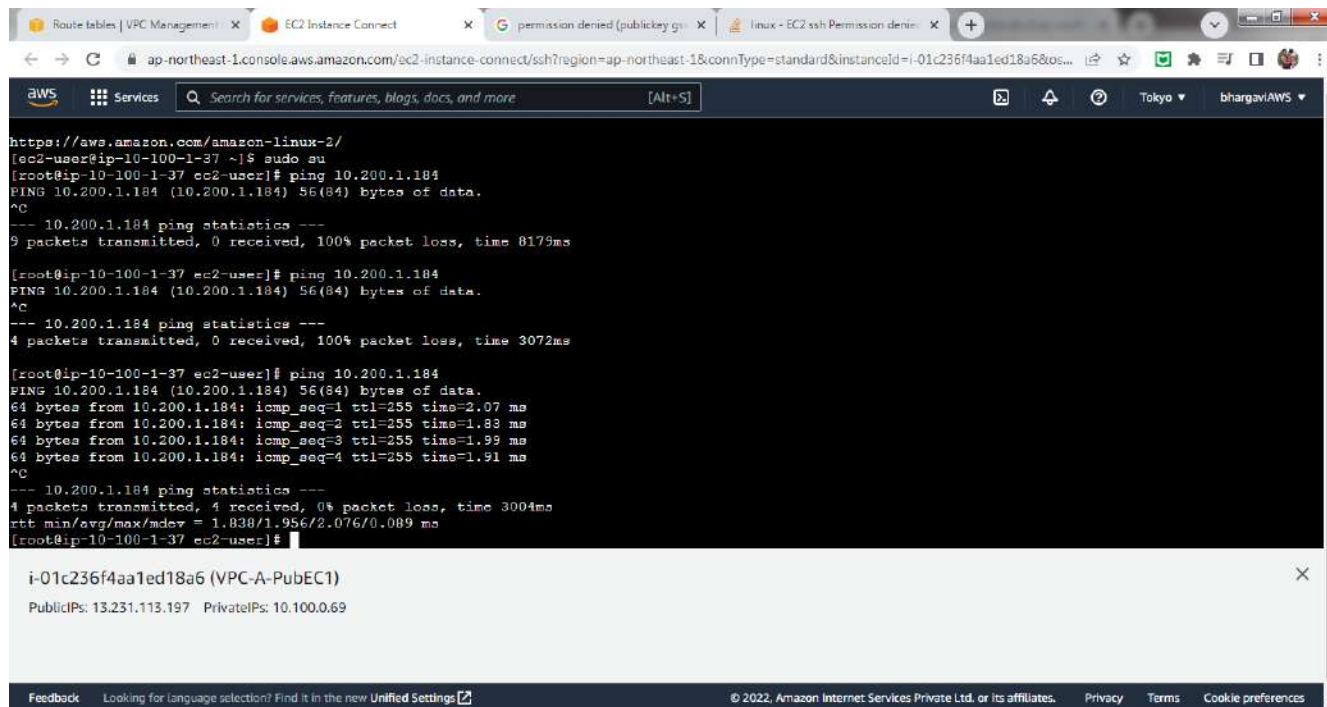
9. Accept the peering connection request



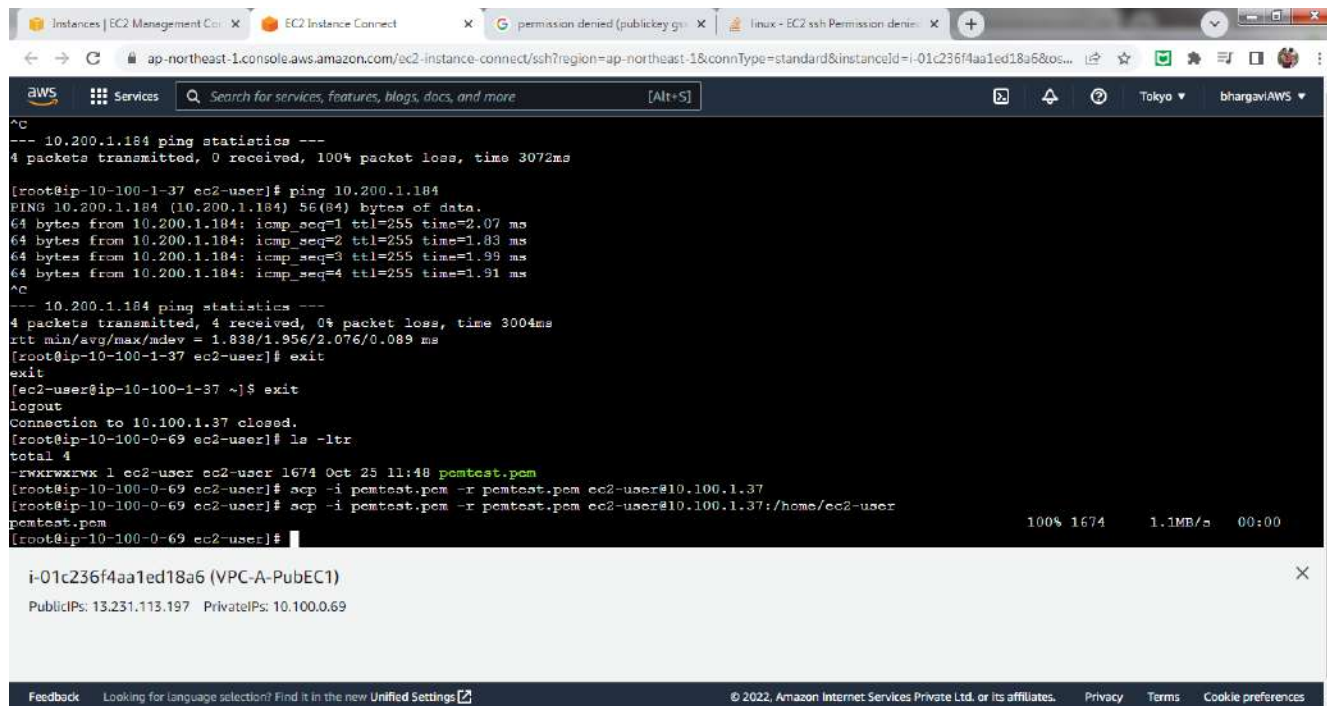
10. Update the private route tables of VPC-A and VPC-B with the newly created peering connection.



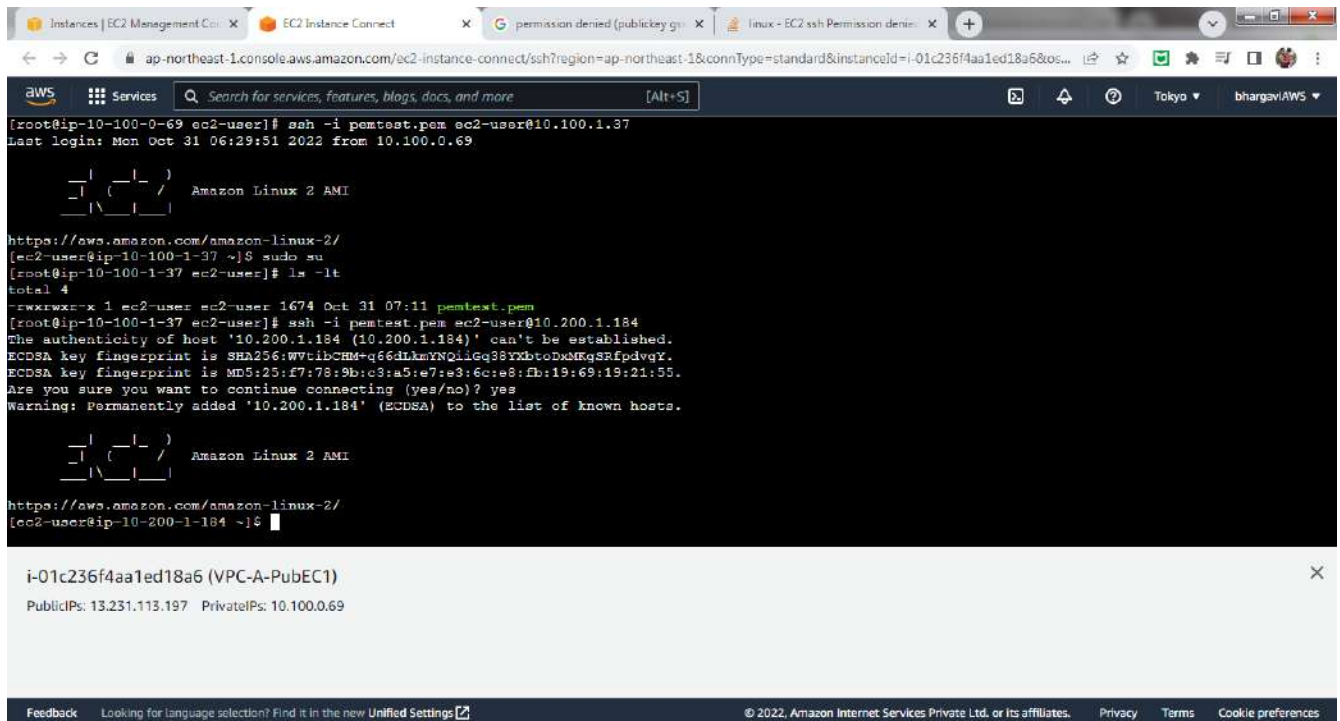
11. Now if we ping from private EC2 instance of VPC-A to private EC2 instance of VPC-B, ping will be successful



12. To connect to private EC2 instance of VPC-B, we need to copy the .pem file from public EC2 instance of VPC-A to private EC2 instance of VPC-A and then ssh to private EC2 instance of VPC-B



13. Now when we try to connect to private EC2 instance of VPC-B from private EC2 instance of VPC-A, both ping and ssh will succeed. We can also observe that .pem file is available in private EC2 instance of VPC-A



The screenshot shows the AWS Management Console interface. The main window displays a terminal session for an EC2 instance in VPC-A. The user is in the root shell and attempts to connect to another EC2 instance in VPC-B using SSH. The terminal output shows the following sequence of events:

```
[root@ip-10-100-0-69 ec2-user]# ssh -i pemtest.pem ec2-user@10.100.1.37
Last login: Mon Oct 31 06:29:51 2022 from 10.100.0.69

  _   _   _   _   _
 _/   \_ /   \_ /   \_ /   \_ /
/_/___\/_/___\/_/___\/_/___\/_/___\
                                     Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-100-1-37 ~]$ sudo su
[root@ip-10-100-1-37 ec2-user]# ls -lt
total 4
-rwxrwxr-x 1 ec2-user ec2-user 1674 Oct 31 07:11 pemtest.pem
[root@ip-10-100-1-37 ec2-user]# ssh -i pemtest.pem ec2-user@10.200.1.184
The authenticity of host '10.200.1.184 (10.200.1.184)' can't be established.
ECDSA key fingerprint is SHA256:WV7ibCHM+q66dLkMYN0iGq38YXbtoDxMKqSRfodvqY.
ECDSA key fingerprint is MD5:25:E7:78:9b:c3:a5:e7:e3:6c:e8:fb:19:69:19:21:55.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.200.1.184' (ECDSA) to the list of known hosts.

  _   _   _   _   _
 _/   \_ /   \_ /   \_ /   \_ /
/_/___\/_/___\/_/___\/_/___\/_/___\
                                     Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-200-1-184 ~]$
```

Below the terminal window, a metadata box for the instance 'i-01c236f4aa1ed18a6 (VPC-A-PubEC1)' is visible, showing PublicIPs: 13.231.113.197 and PrivateIPs: 10.100.0.69.

So peering connection is done and we are able to connect the private EC2 instances of both VPC-A and VPC-B.