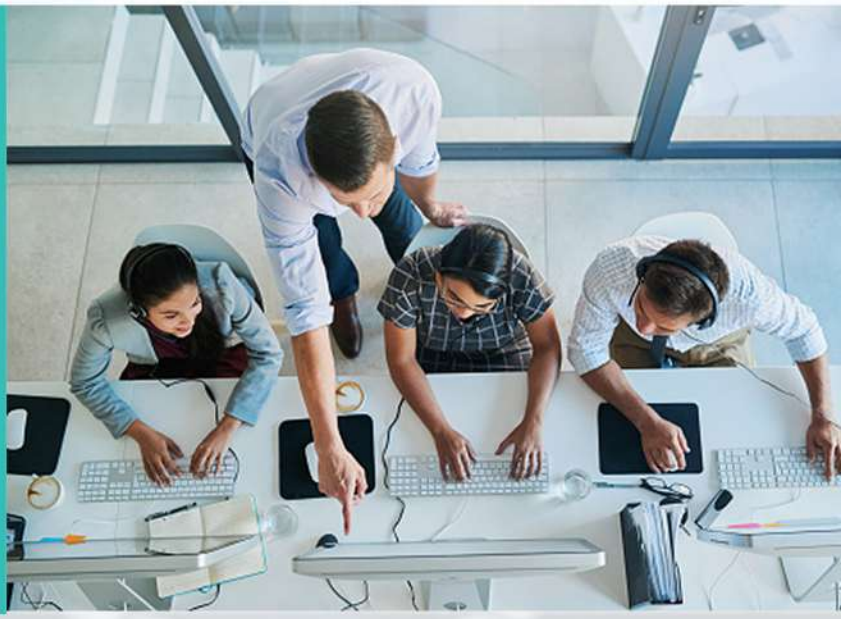
Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

e Skill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation



AWS CLOUD COMPUTING

NATGATE WAYS

Configuration of NAT Gateways

NAT Gateways

NAT is a networking technique commonly used to give an entire private network access to the internet without assigning each host a public IPv4 address. The hosts can initiate connections to the internet and receive responses, but not receive inbound connections initiated from the internet.

When a host in the private network initiates an internet-bound connection, the NAT device's public IP address becomes the source IP address for the outbound traffic. The response traffic from the internet therefore uses that public IP address as the destination IP address. The NAT device then routes the response to the host in the private network that initiated the connection.

The Networking service offers a reliable and highly available NAT solution for your VPC in the form of a NAT gateway.

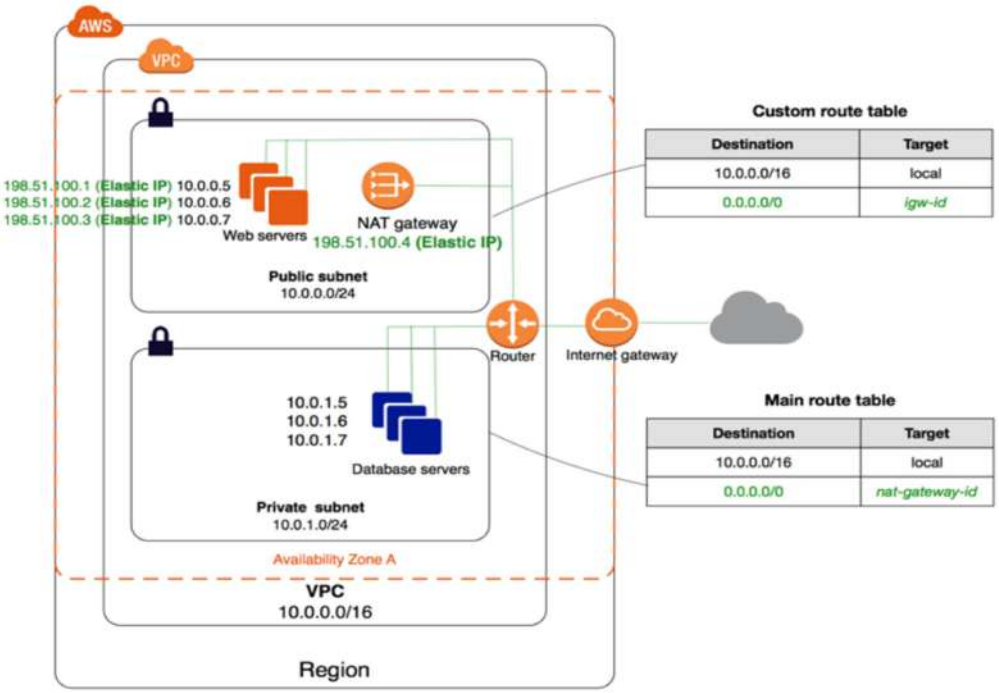
Example scenario: Imagine you have resources that need to receive inbound traffic from the internet (for example, web servers). You also have private resources that need to be protected from inbound traffic from the internet. All of these resources need to initiate connections to the internet to request software updates from sites on the internet.

You set up a VPC and add a public subnet to hold the web servers. When launching the instances, you assign public IP addresses to them so they can receive inbound internet traffic. You also add a private subnet to hold the private instances. They cannot have public IP addresses because they are in a private subnet.

You add an internet gateway to the VPC. You also add a route rule in the public subnet's route table that directs internet-bound traffic to the internet gateway. The public subnet's instances can now initiate connections to the internet and also receive inbound connections initiated from the internet. Remember that you can use security rules to control the types of traffic that are allowed in and out of the instances at the packet level.

You add a NAT gateway to the VPC. You also add a route rule in the private subnet's route table that directs internet-bound traffic to the NAT gateway. The private subnet's instances can now initiate connections to the internet. The NAT gateway allows responses, but it does not allow connections that are initiated from the internet. Without that NAT gateway, the private instances would instead need to be in the public subnet and have public IP addresses to get their software updates.

The following diagram represents the NAT Gateway connection in AWS.

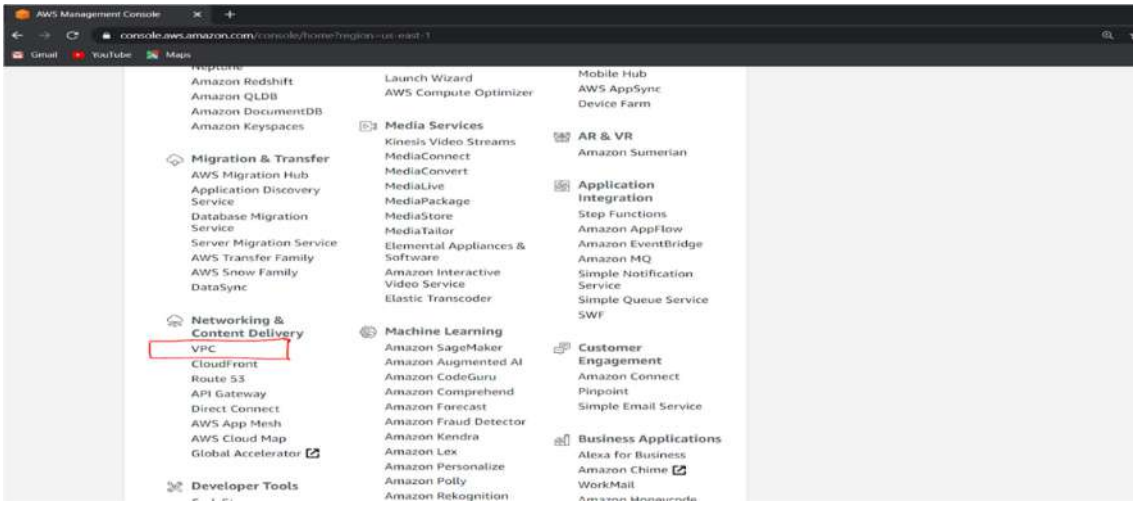


Practical Steps:

To Create your own VPC

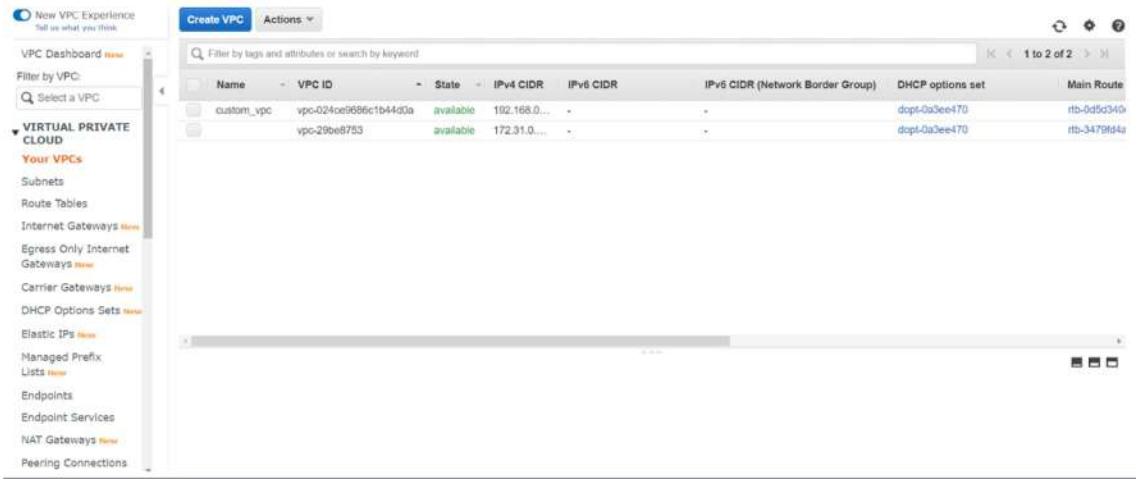
Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. If you have a default VPC, you can skip this section and move to the next task,

Open AWS console, Click on Services. Select Networking and Content Delivery and click on VPC

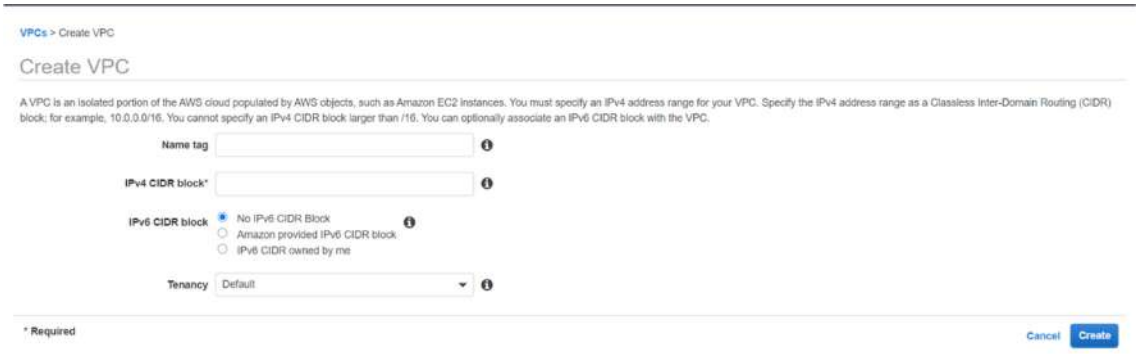


On VPC Dashboard panel

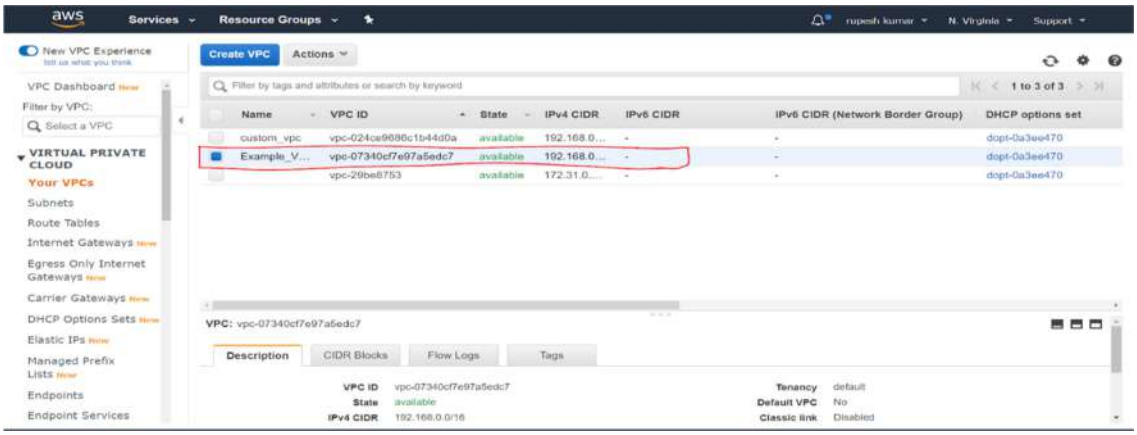
Click on Your VPC, Click on Create VPC button



On “Create VPC”, page
For Name tag → Example_VPC
For IPv4 CIDR block → 192.168.0.0/16
Click on “Create” button

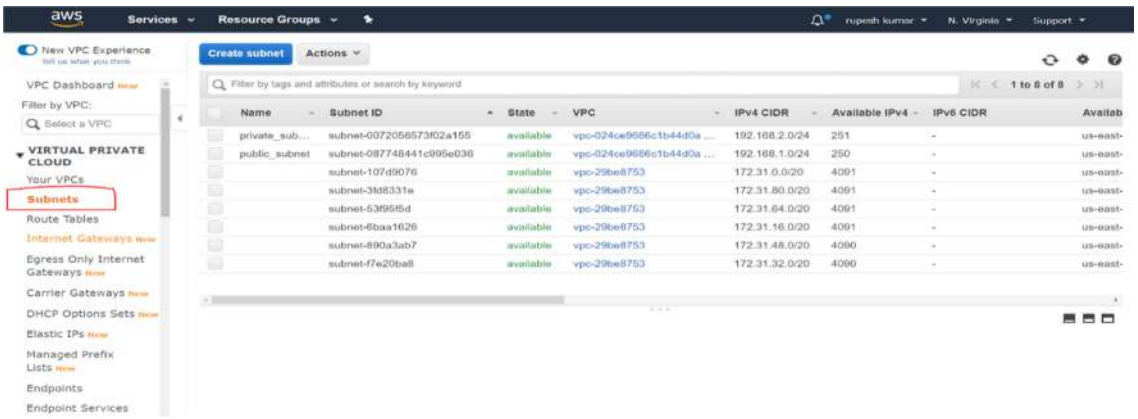


Verify Example_VPC was created

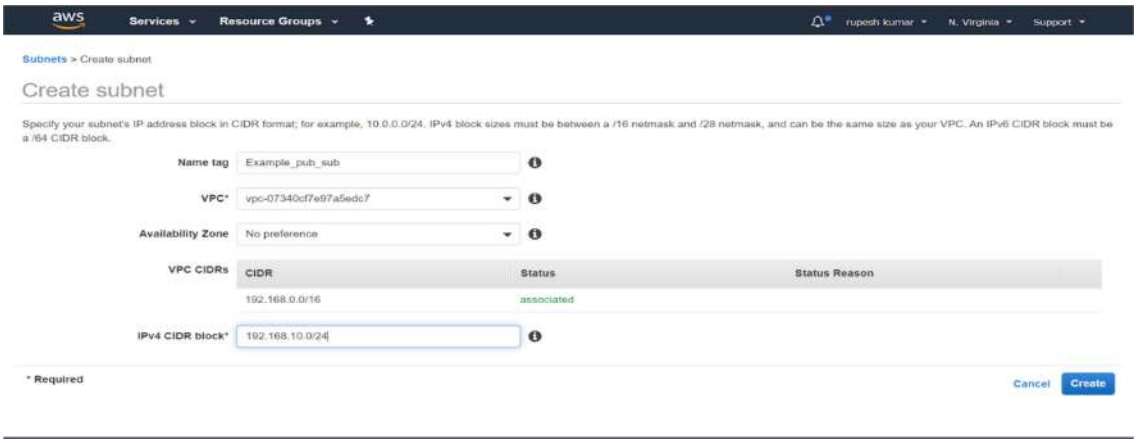


To create public subnet

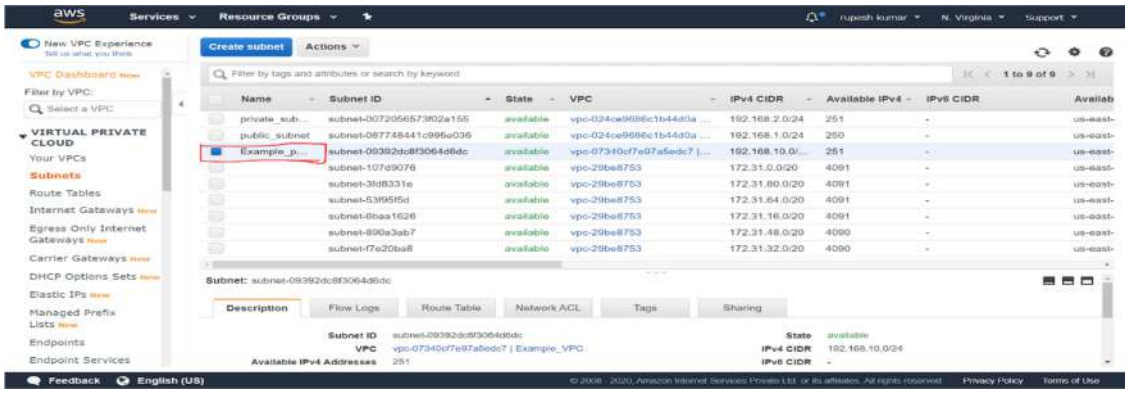
Click on Subnet. Click on Create Subnet button



On Create Subnet, page
For Name tag → Example_pub_sub
For VPC → Example_VPC
For IPv4 CIDR block → 192.168.10.0/24
Click on Create button

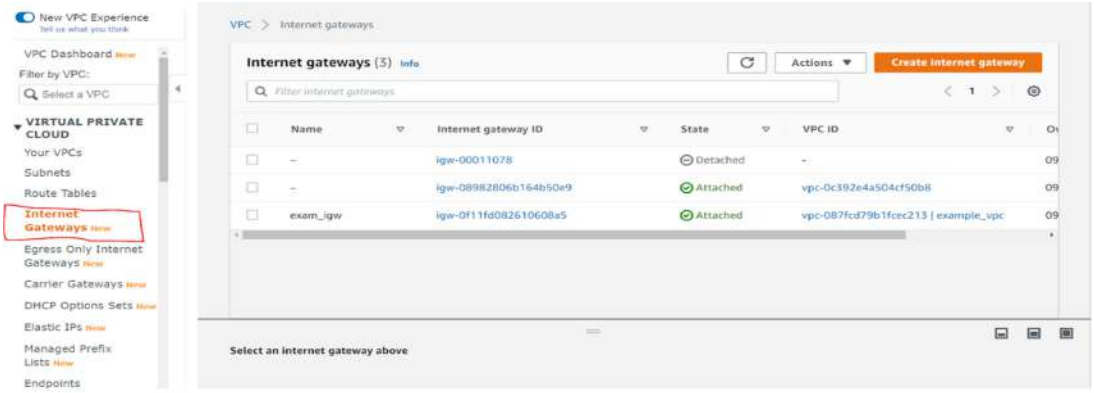


Verify Example_pub_subnet got created

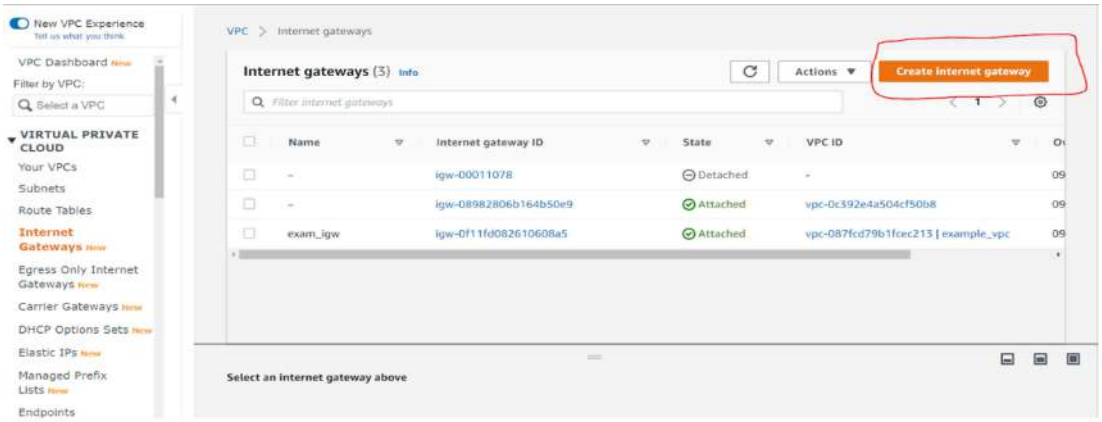


Create Internet gateway and attach to your VPC

In VPC Dashboard panel, Click on Internet Gateways

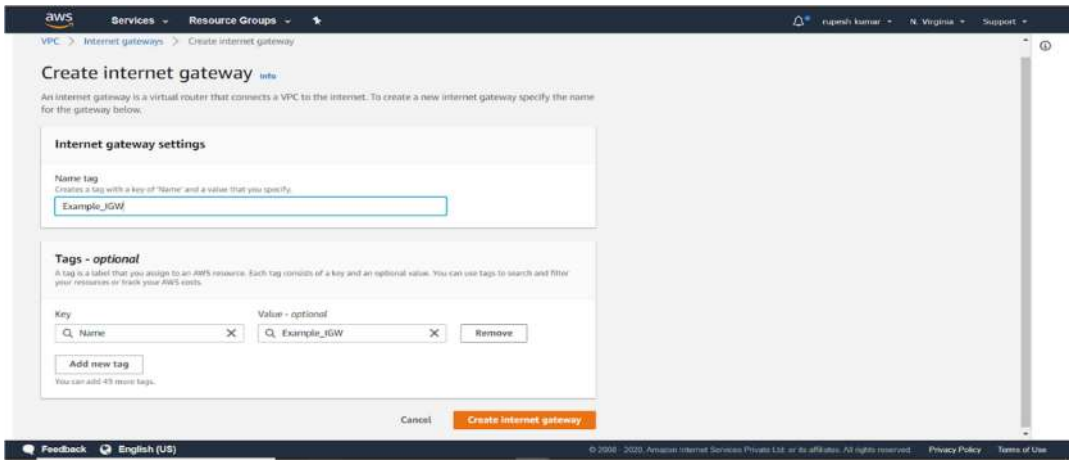


Click on Create Internet Gateway button

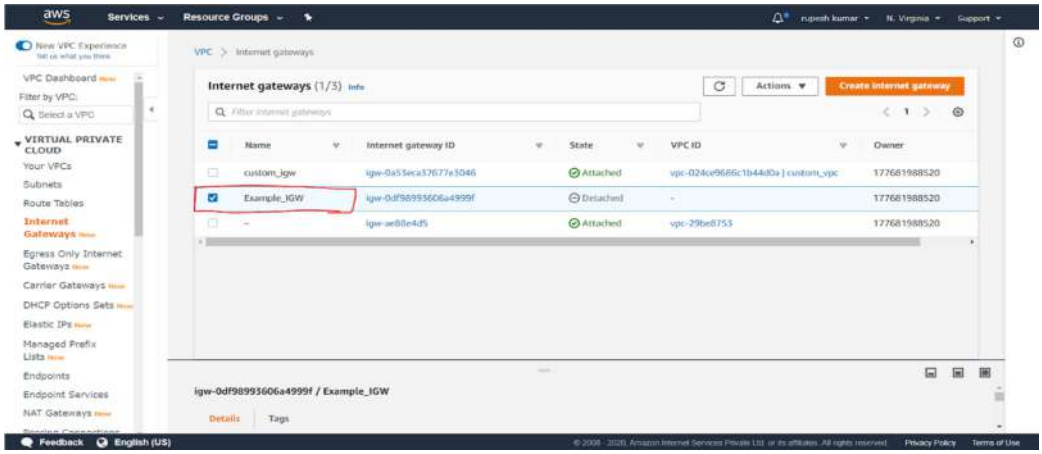


In Create Internet Gateway, box

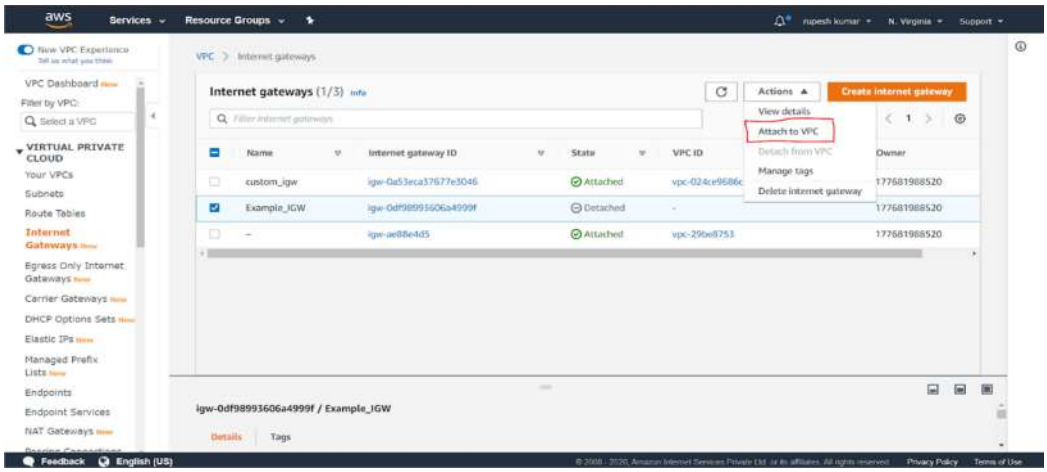
For Name tag → Example_IGW, Click on "Create internet gateway" button



Verify Internet gateway is created

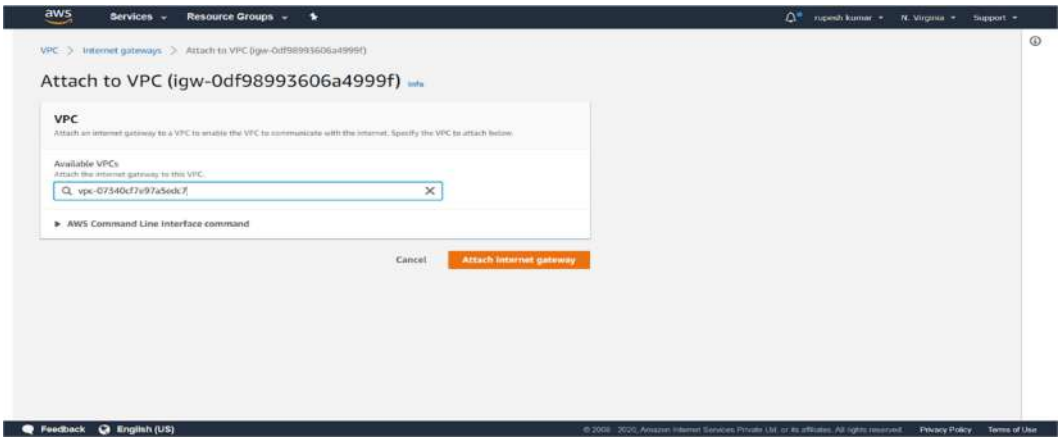


Select Example_IGW
Click on actions and select option “Attach to VPC”

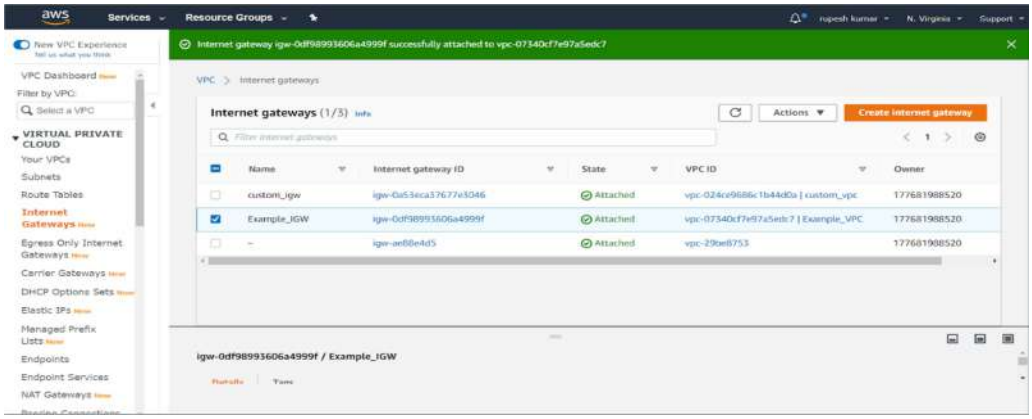


In “Attach to VPC” box

For VPC Example_VPC
Click on “Attach internet gateway” button

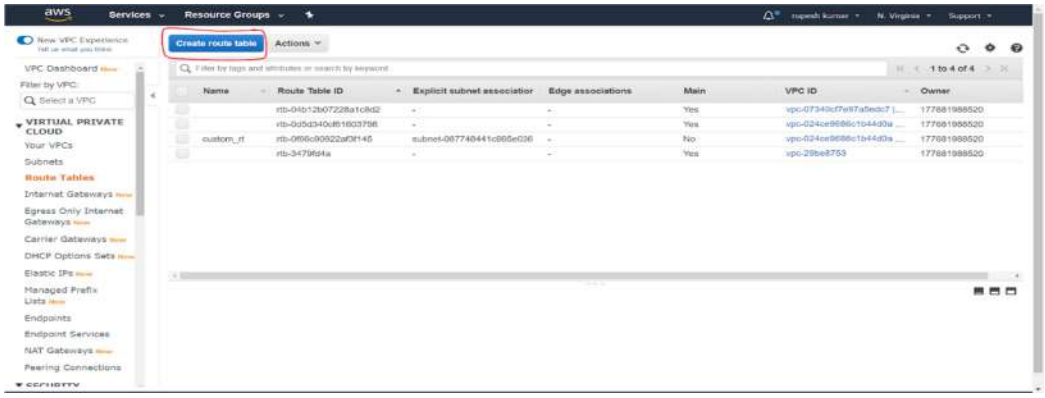


Verify Internet gateway is connected to your VPC

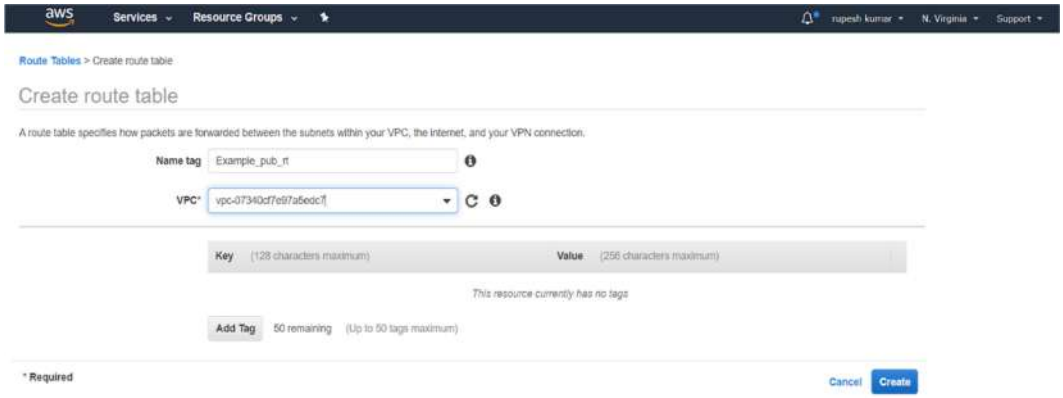


Create Pubic Routing Table, associate subnet and add routing rules

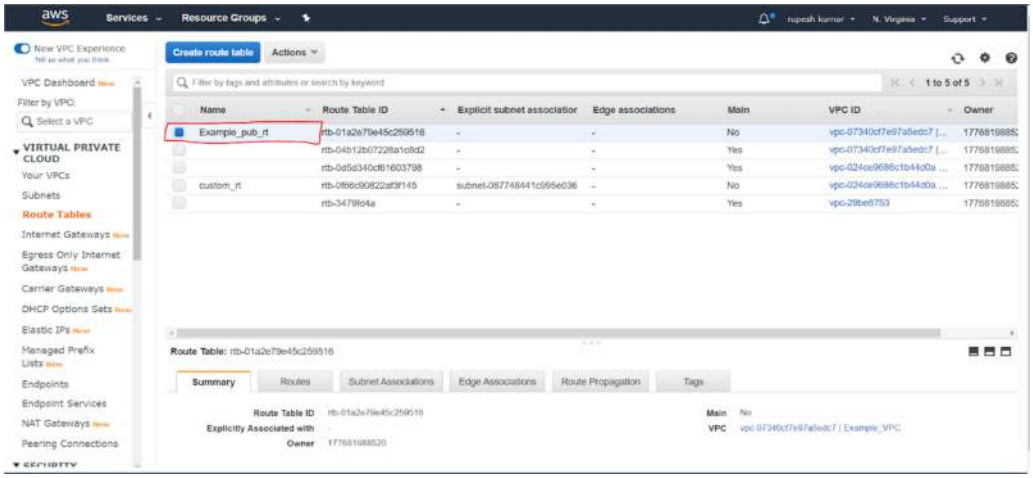
On VPC Dashboard panel, Click on Route Table then Click on “Create Route Table” button



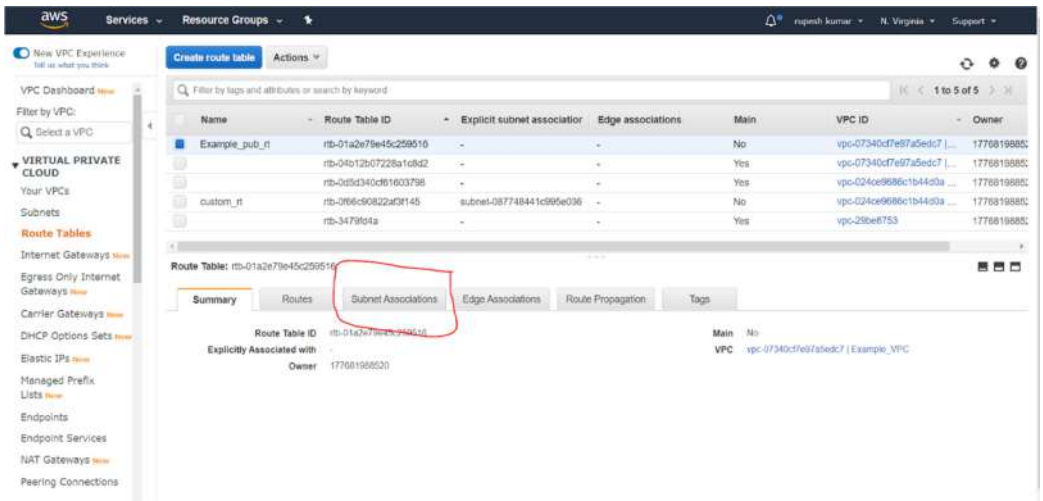
On “Create Route Table” box
For Name tag → Example_pub_rt
For VPC → Example_VPC
Click on “Create” button



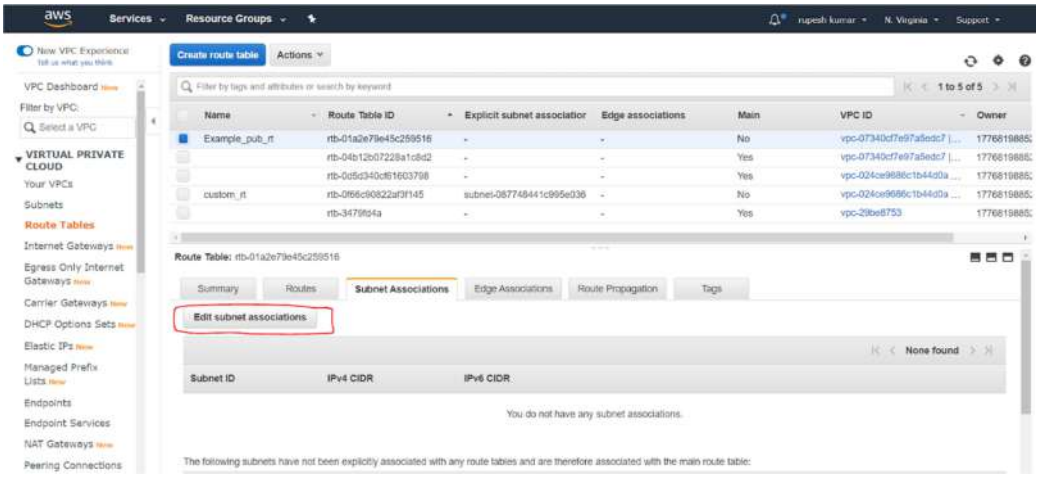
Verify Example_pub_rt table is created



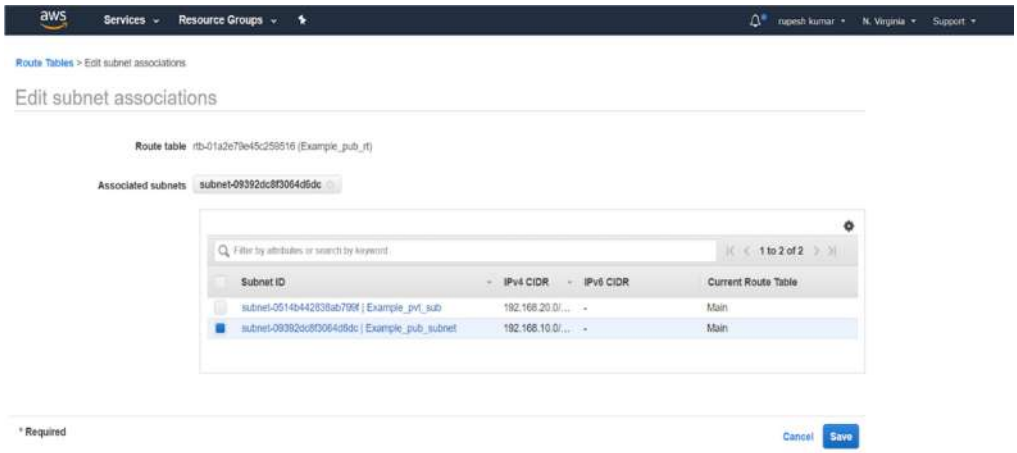
Click on "Subnet Association" button



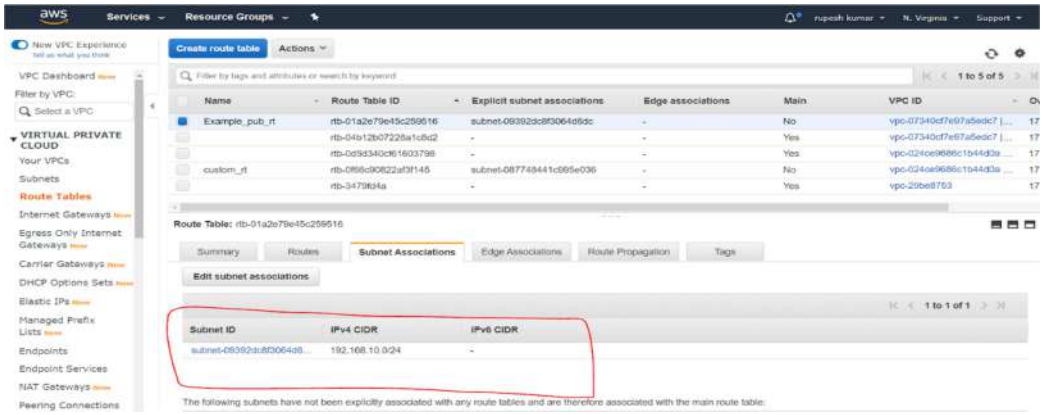
Click on Edit subnet association button



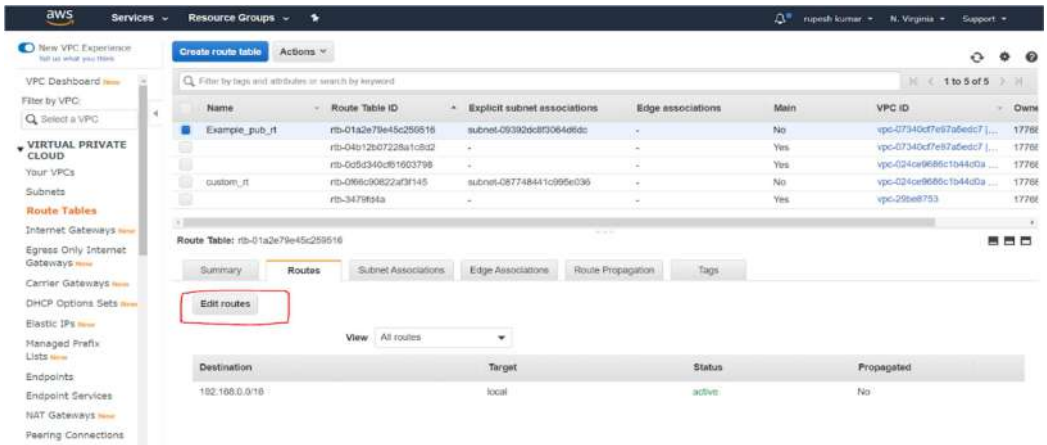
Select checkbox of Example_pub_sub → 192.168.10.0/24
Click on save button



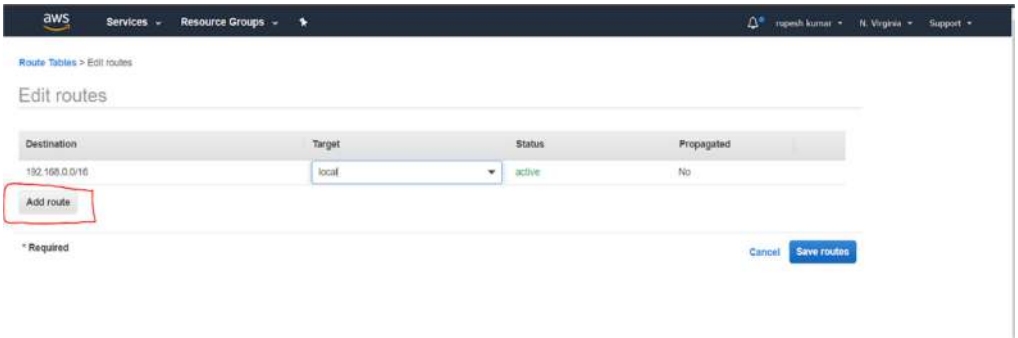
Verify Example_pub_subnet is associated with routing table



Click on Route button and Click on Edit button



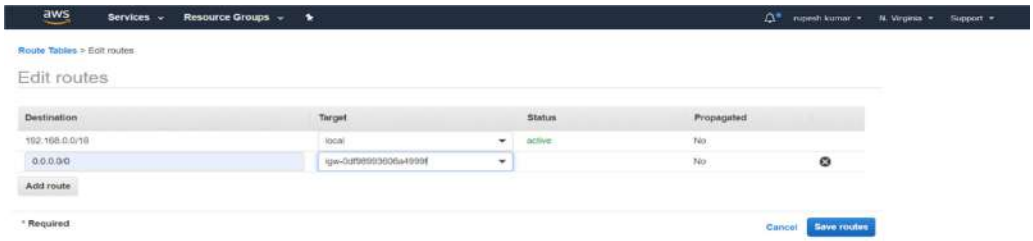
Click on “Add route” button



For Destination → 0.0.0.0/0

For Target → select Example_IGW

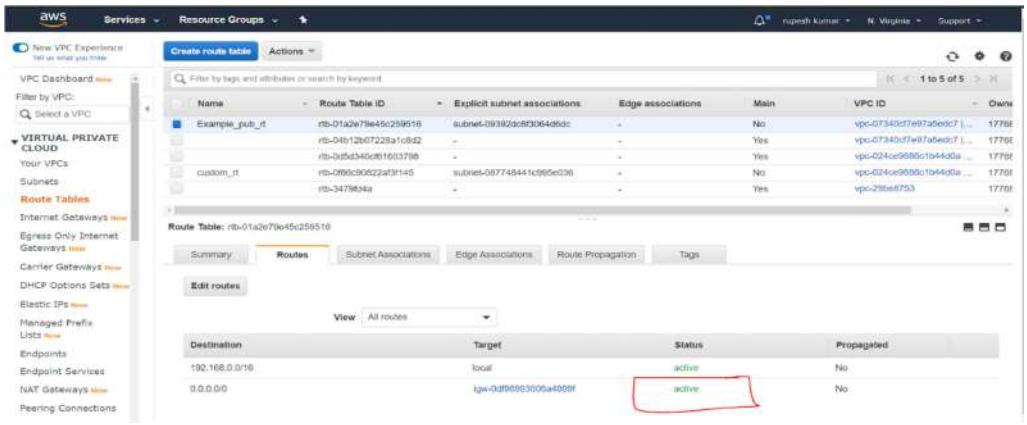
Click on Save button



Verification

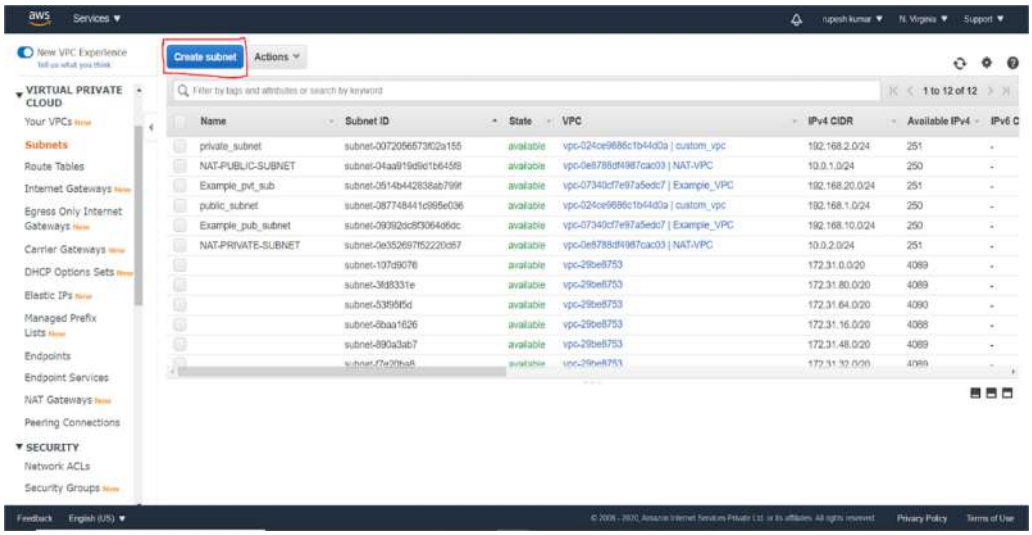
Public route is added through internet gateway

Verify Status column show Active



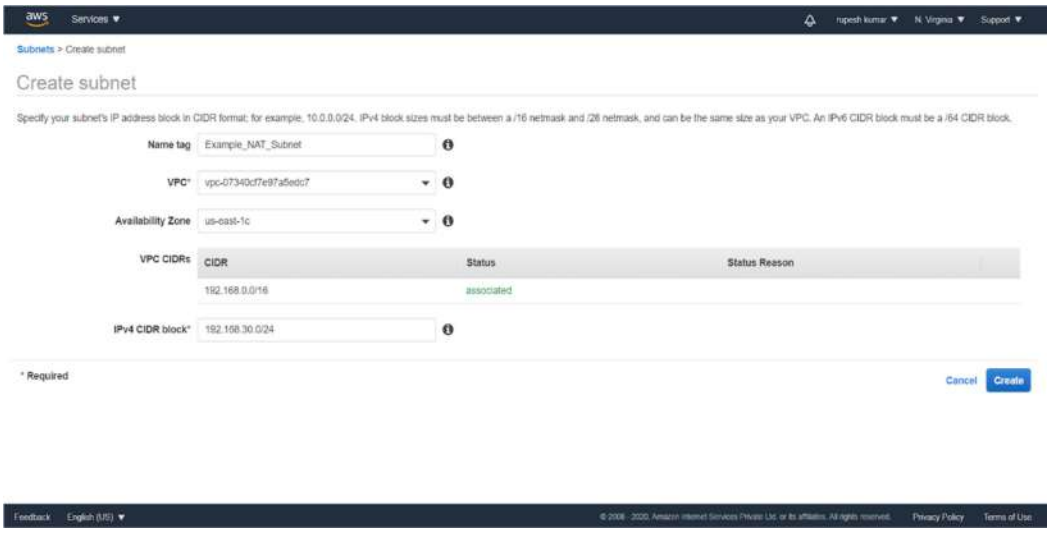
Create NAT Subnet

Click on create subnet



For the subnet details:

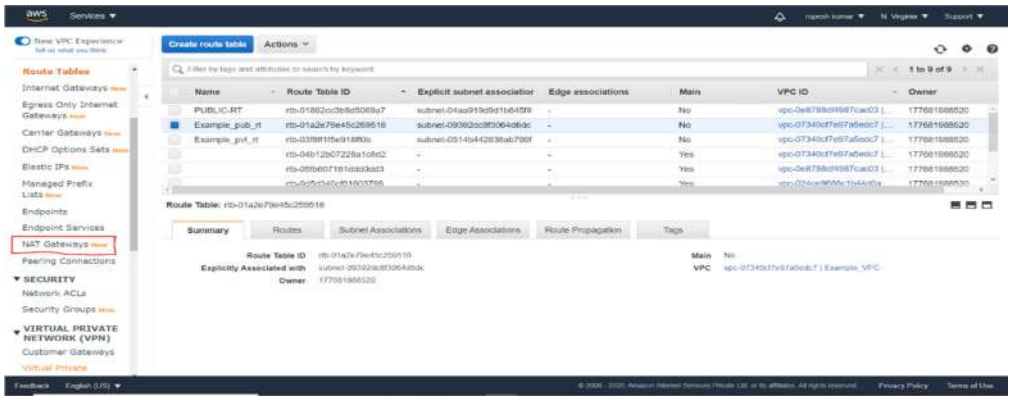
Name tag → Example_NAT_subnet
VPC → Example_VPC
IPV4 CIDR Block →192.168.30/24



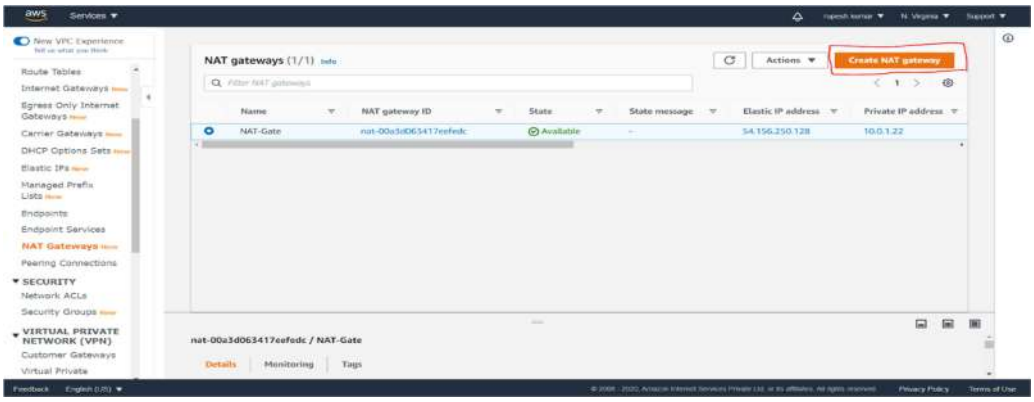
Now click on create and then NAT subnet was created

Create NAT Gateway

Click on NAT Gateway in the VPC dashboard

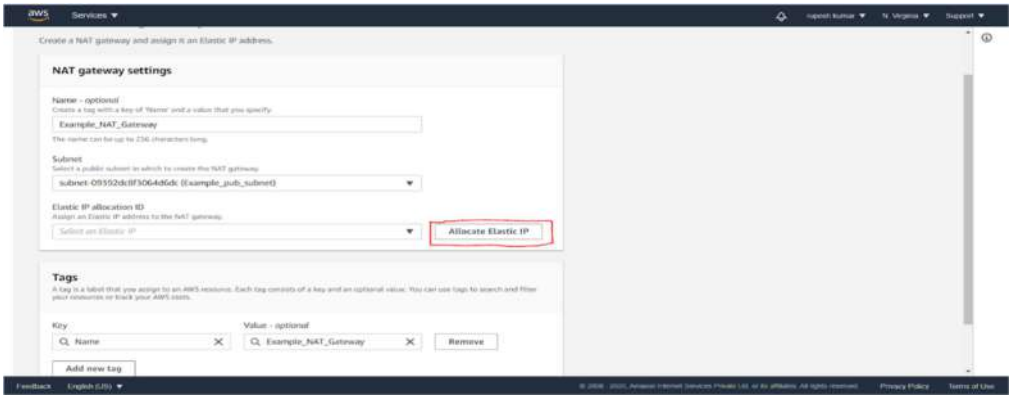


Now Click on create NAT Gateway

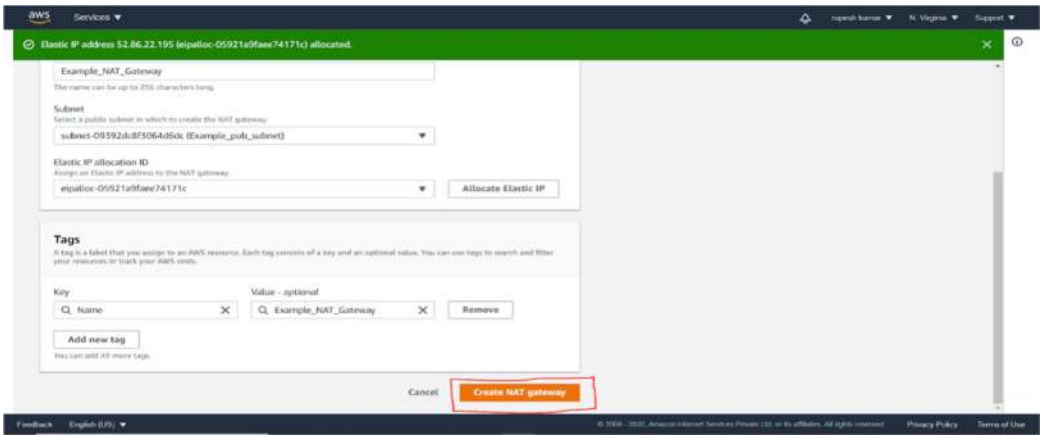


In the NAT Gateway settings

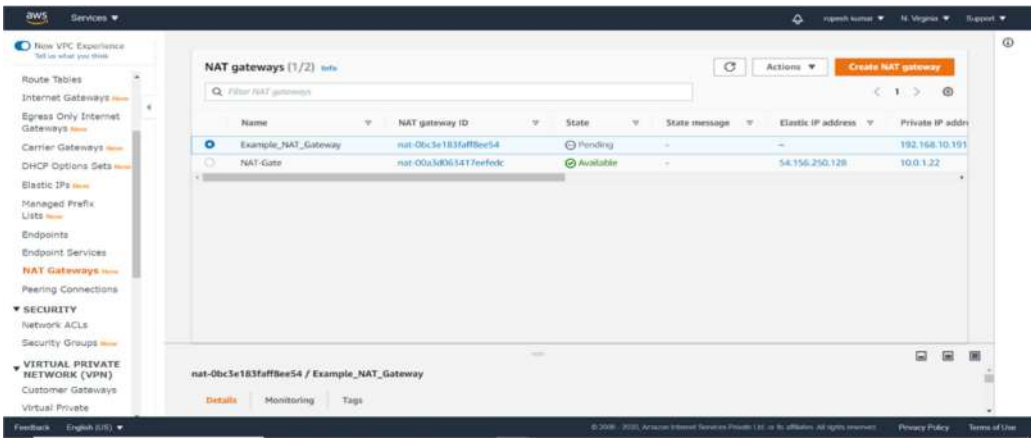
Name →Example_NAT_Gateway
Subnet → select the public subnet from the Example_VPC
For the Elastic IP → Click on Allocate Elastic IP



Now elastic ip got created and then click on the create NAT Gateway

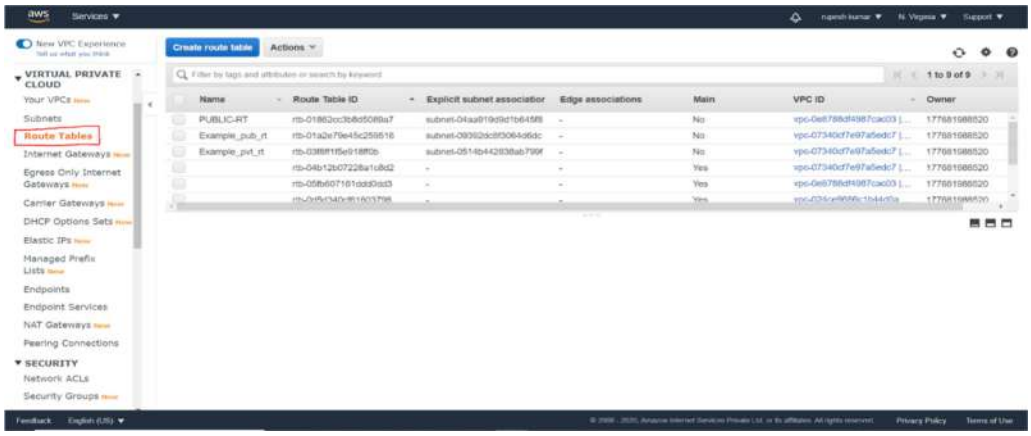


Now NAT Gateway was created

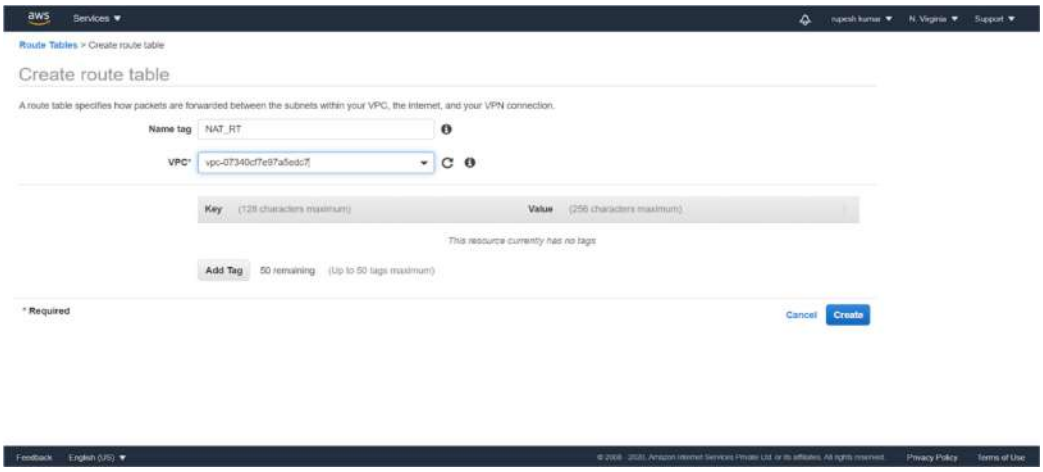


Create NAT Route Table

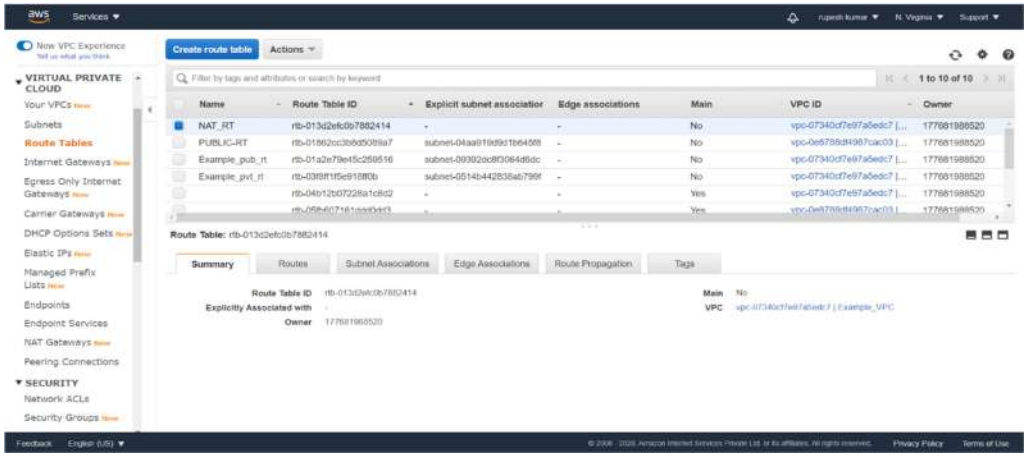
From the VPC dashboard console click on the route table



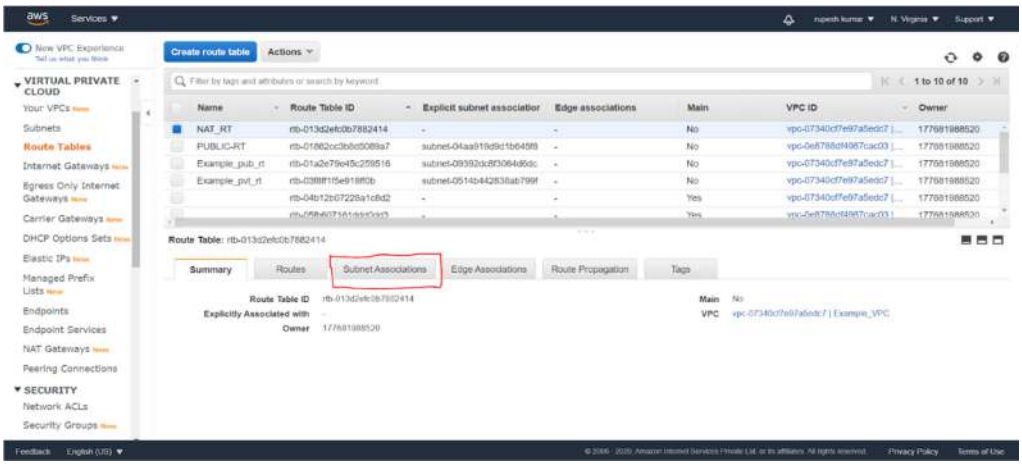
Now click on the create route table
For the route table details:
Name tag →NAT_RT
VPC →Example_VPC



Click on create, route table was created

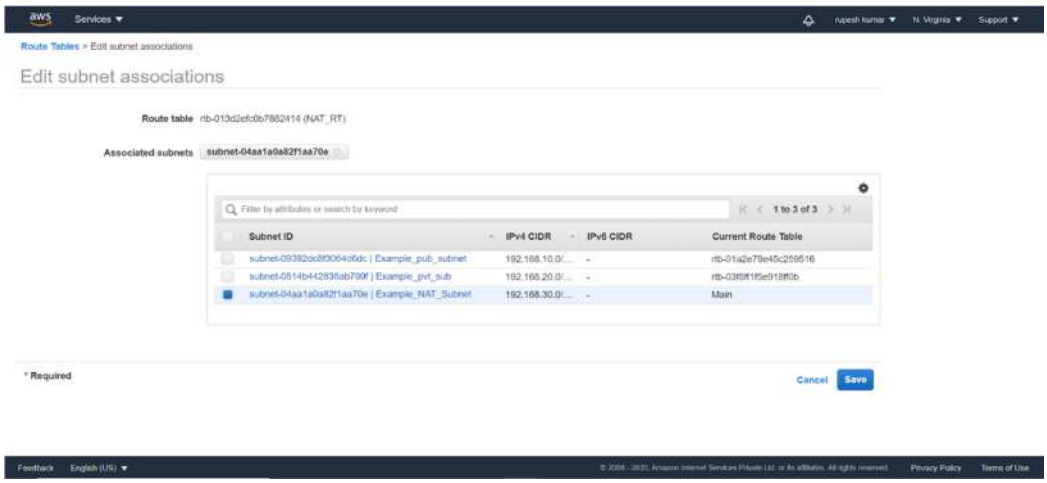


Click on subnet association

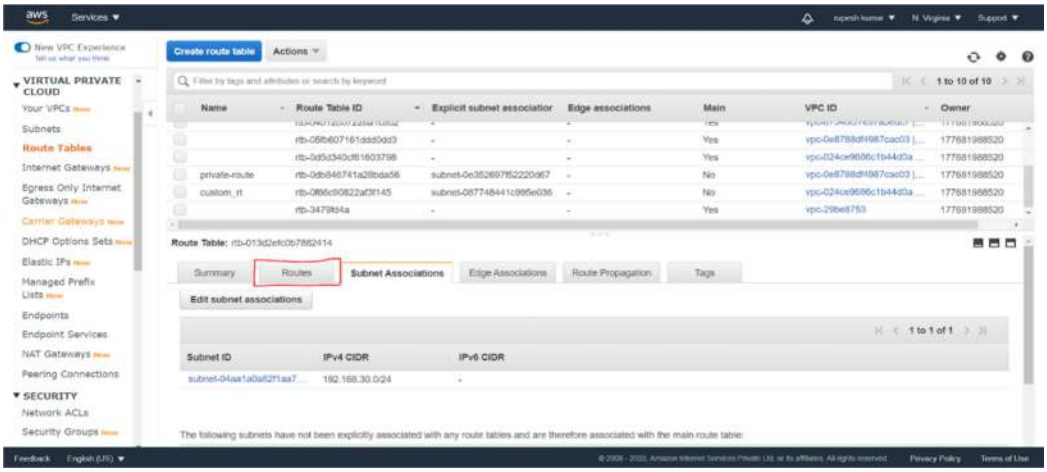


Click on edit subnet association

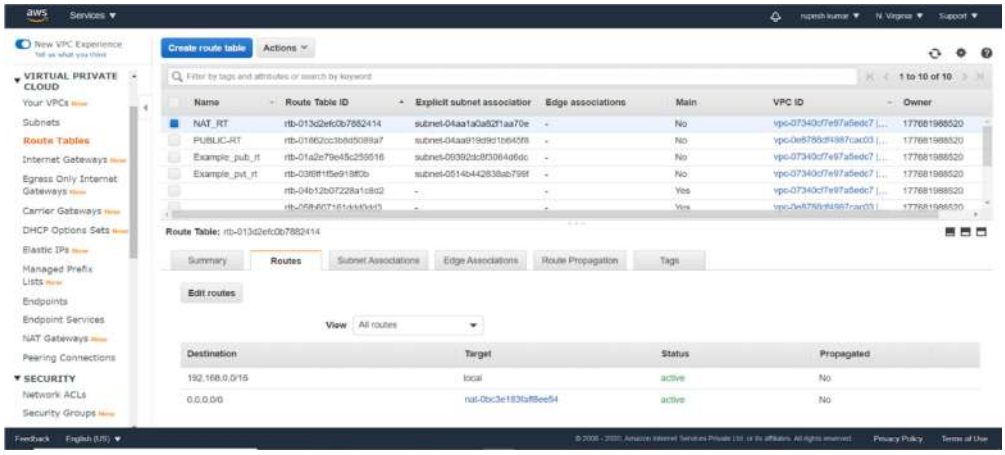
Select the Example_NAT_subnet and click on Save



Now click on routes and add the NAT Gateway

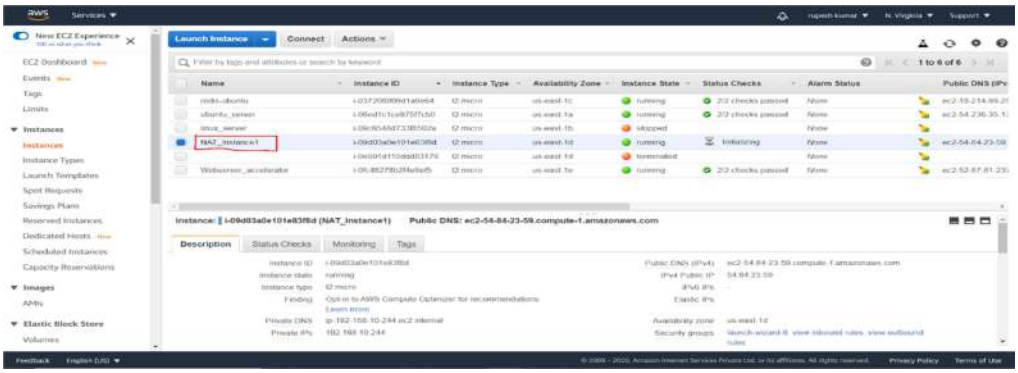


Click on edit routes and add the route
For the routes details:
Destination → 0.0.0.0./0
Target → NAT Gateway and select the Example_NAT_Gateway
Now the routes were added successfully.



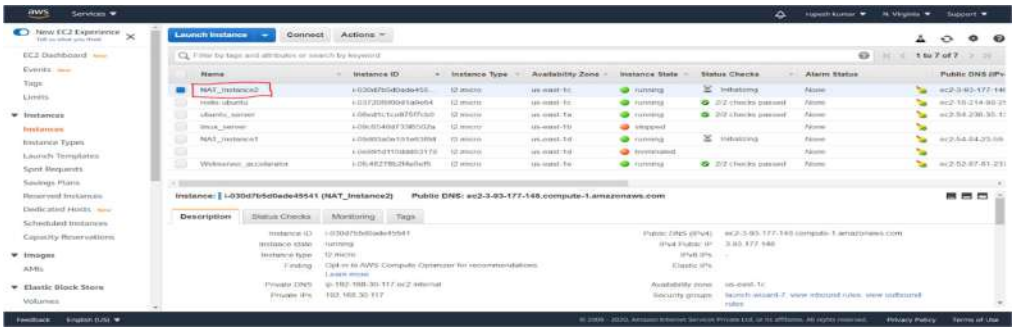
Verify NAT Gateway Configuration

To check the NAT Gateway connection, create two instances.
Create the first instance (NAT_Instance1):
VPC → Example_VPC
Subnet → Example_pub_subnet
Name Tag → NAT_Instance1

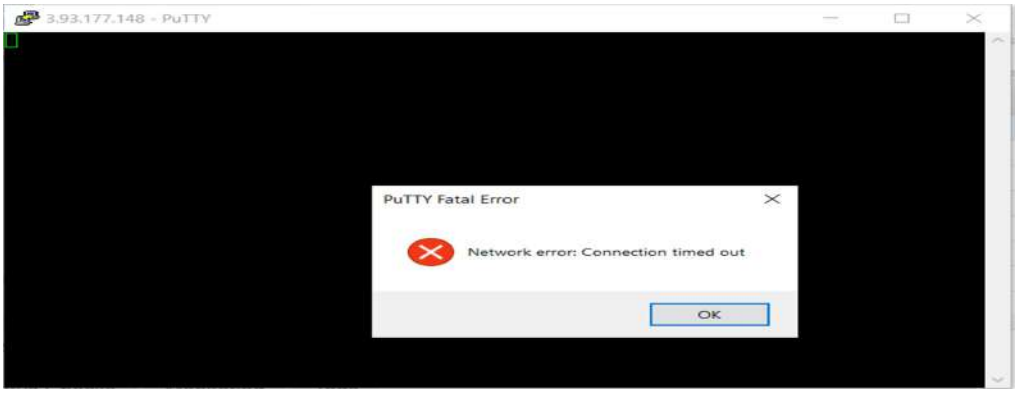


Create the second instance (NAT_Instance2):

VPC → Example_VPC
Subnet → Example_NAT_subnet
Name Tag → NAT_Instance2



Now connect through NAT_Instance2 using putty, it should be failed.
Here the NAT Gateway shouldn't allow the direct connection to the instance.



If you want to connect to NAT_Instance2, first connect to the public instance and then connect to the instance which is in NAT Gateway.

Now connect to the NAT_Instance1 using putty.
Instance1 was connected successfully

```
ec2-user@ip-192-168-10-244:~
login as: ec2-user
Authenticating with public key "imported-openssh-key"

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

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-10-244 ~]$
```

Now check if the instance is working or not using the private ip address of the instance2.

```
root@ip-192-168-10-244:/home/ec2-user
       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-10-244 ~]$ sudo su
[root@ip-192-168-10-244 ec2-user]# ping 192.168.30.117
PING 192.168.30.117 (192.168.30.117) 56(84) bytes of data.
64 bytes from 192.168.30.117: icmp_seq=1 ttl=255 time=1.51 ms
64 bytes from 192.168.30.117: icmp_seq=2 ttl=255 time=1.55 ms
64 bytes from 192.168.30.117: icmp_seq=3 ttl=255 time=1.54 ms
64 bytes from 192.168.30.117: icmp_seq=4 ttl=255 time=1.54 ms
64 bytes from 192.168.30.117: icmp_seq=5 ttl=255 time=1.54 ms
64 bytes from 192.168.30.117: icmp_seq=6 ttl=255 time=1.54 ms
64 bytes from 192.168.30.117: icmp_seq=7 ttl=255 time=1.47 ms
64 bytes from 192.168.30.117: icmp_seq=8 ttl=255 time=1.52 ms
64 bytes from 192.168.30.117: icmp_seq=9 ttl=255 time=1.53 ms
64 bytes from 192.168.30.117: icmp_seq=10 ttl=255 time=1.54 ms
64 bytes from 192.168.30.117: icmp_seq=11 ttl=255 time=1.50 ms
64 bytes from 192.168.30.117: icmp_seq=12 ttl=255 time=1.54 ms
64 bytes from 192.168.30.117: icmp_seq=13 ttl=255 time=1.50 ms
```

The results show that the instance2 is working properly.
Now connect to the instance2 in NAT subnet using the private ip, the command is
Ssh ec2-user@192.168.30.117.

```
root@ip-192-168-10-244:/home/ec2-user
[root@ip-192-168-10-244 ec2-user]# ssh ec2-user@192.168.30.117
The authenticity of host '192.168.30.117 (192.168.30.117)' can't be established.
ECDSA key fingerprint is SHA256:XLXSkORrs2nSKwcko3GjaWtIoICopGXX5GSgUTNlc/Y.
ECDSA key fingerprint is MD5:30:d9:ab:ff:5b:77:e5:0c:ab:ef:9f:f6:80:c3:8e:25.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.30.117' (ECDSA) to the list of known hosts.
Permission denied (publickey).
[root@ip-192-168-10-244 ec2-user]#
```

It gives the error message, now add the key pair related to the NAT subnet instance and add the content of the keypair.
nano example.pem

```
root@ip-192-168-10-244:/home/ec2-user
  GNU nano 2.5.3              File: example.pem                    Modified

DrqDz3PJejLo9qm8u0CQnab0LqkCgYEA6NppDG2vYoL++uDjEObwFA1G0qIOEv7QdEOJ/MMv+vdm
lkJd3Dh9kdYkl7wjAvWlnznVp2/VeIFUREzDGXlke4tLyjkz8TLL90lUUQ2KuFssuxDRAbtPpkYq
CQKs4UhDRYjtlwxHFGy+cvmy8J+Ergptuwrz7hIIkTAnbNH7jgUCgYEAnvYTjiVQJwgGVoH+a+ly
j8YZoQBbX7vUn4p6YSpwXRGRKefSqVQVTCQjXpb8NtRTHlBy6gYonzw2sn+gpBNP83P7RLm+clMn
+81MlYpmijZxc2633NUiegsegmeStZgk9/zRs2+j/Cv95Cd80Y2NM5KmSjMN/KSuPklV9ozPfsMC
gYAFtalTv7DIQpwL/M20kWUbqMOu0Ih10Me9whY1G3gmuEBMuO3iQ4RYuh6FlfhpzyozgFCL3YMn
hExTrGowSri7Csxd9g8e//beZm7p2eIn6RxsrdniJ/ypABlxxR4GHCaCIvxocsWfa5cz7ImFuvOe
7OSJ7JfUo56QKNUepvt7HQKBgE+m0F+utL4KGFrqoljiTiRayGbTbfXHgaA73FypzknkTo/n+Lge
SNmxVgPEtjanJ8sN1rcLRPmdS28f4pe0kaBU4ngqdpKEtolG8vQPuJfS9gA4X2sJZVAwGCaRzYKu
7P9SMKWv8ESgqoTTgIjkW/eWqTqO4JhS0HineKqlHifPAoGAE4GjkOhkrm16Ajtoq8yFeifPWA3S
Ry2ITBnPyKCz7bn9DcsFxj8YfD1xb4dQmBsdU11wUU4OG7zoR269bplVYBuuJi8LZaj8hkHmeA03
h/zvnab6SfnbZ3SNnFYPHD3uhVi8yQ7aFZBjaVd2aSR2KYwqGqoa2t5Oc94Toh91Nak=
-----END RSA PRIVATE KEY-----

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Now run the command to connect to the instance in the NAT subnet

```
root@ip-192-168-10-244:/home/ec2-user
[root@ip-192-168-10-244 ec2-user]# nano example.pem
[root@ip-192-168-10-244 ec2-user]# ssh -i example.pem ec2-user@192.168.30.117
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'example.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "example.pem": bad permissions
Permission denied (publickey).
[root@ip-192-168-10-244 ec2-user]#
```

Now change the permissions using the command
Chmod 400 example.pem
And again, try to connect the NAT instance

```
ec2-user@ip-192-168-30-117:~
[root@ip-192-168-10-244 ec2-user]# chmod 400 example.pem
[root@ip-192-168-10-244 ec2-user]# ssh -i example.pem ec2-user@192.168.30.117

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

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-30-117 ~]$
```

After running the command immediately, the prompt was changed from instance1 to instance2.

```
ec2-user@ip-192-168-30-117:~
[root@ip-192-168-10-244 ec2-user]# chmod 400 example.pem
[root@ip-192-168-10-244 ec2-user]# ssh -i example.pem ec2-user@192.168.30.117

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

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-30-117 ~]$
```

Check if the instance2 is working properly.

```
ec2-user@ip-192-168-30-117:~
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-30-117 ~]$ ping google.com
PING google.com (172.217.164.142) 56(84) bytes of data.
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=1 ttl=111 ti
me=2.62 ms
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=2 ttl=111 ti
me=2.18 ms
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=3 ttl=111 ti
me=2.24 ms
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=4 ttl=111 ti
me=2.15 ms
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=5 ttl=111 ti
me=2.14 ms
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=6 ttl=111 ti
me=6.01 ms
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=7 ttl=111 ti
me=2.20 ms
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=8 ttl=111 ti
me=2.26 ms
```

The results show that the NAT Gateway connections were working properly.