Virtual Private Cloud(VPC)

Author : Debajyoti Sahani

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own datacenter, with the benefits of using the scalable infrastructure of AWS.

Major Concepts:

Subnets

Route Table

Internet Gateway

Nat Gateway

Security Groups

Network Access Control List(NACL)

Peering Connection

VPN

Subnets:

- Subnetting is the process of dividing a network into two or more subnets.
- An IP address has numbers that identify the network ID and the host ID.
- A subnet address borrows some of the bits from the host ID of the IP address.
- Subnetting is largely invisible to computer users who aren't also network administrators

Route Tables:

A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic is directed.

Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

Internet Gateway:

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

An internet gateway supports IPv4 and IPv6 traffic.

NAT Gateway:

You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

NAT gateways are not supported for IPv6 traffic—use an egress-only internet gateway instead.

You are charged for creating and using a NAT gateway in your account. NAT gateway hourly usage and data processing rates apply.

Security Groups:

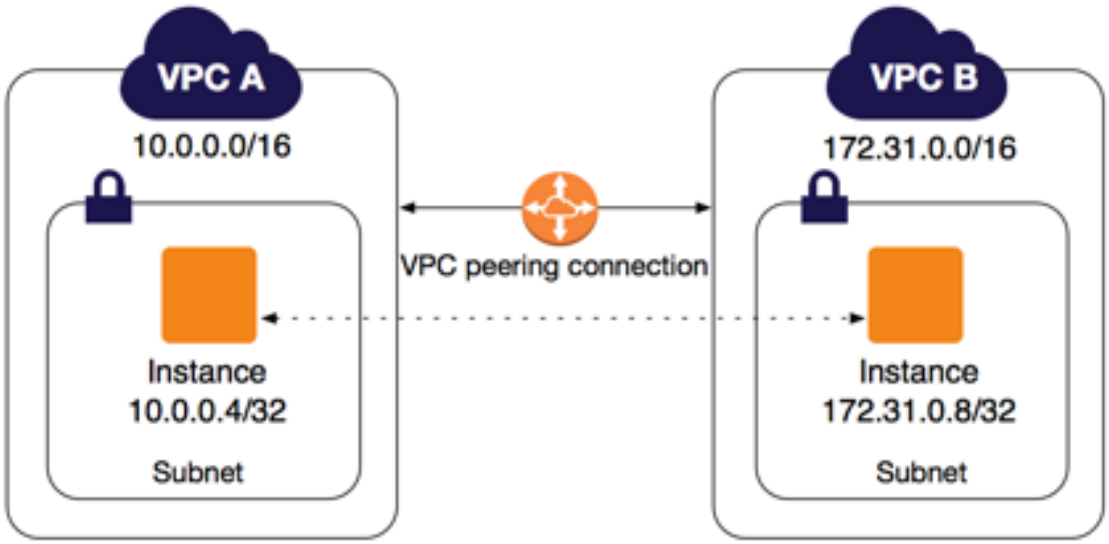
- A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.
- When you launch an instance in a VPC, you can assign up to five security groups to the instance.
- Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups.
- If you don't specify a particular group at launch time, the instance is automatically assigned to the default security group for the VPC.

Network Access Control Lists(NACL):

A *network access control list (ACL)* is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

VPC Peering:

- A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses.
- Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different regions (also known as an inter-region VPC peering connection).



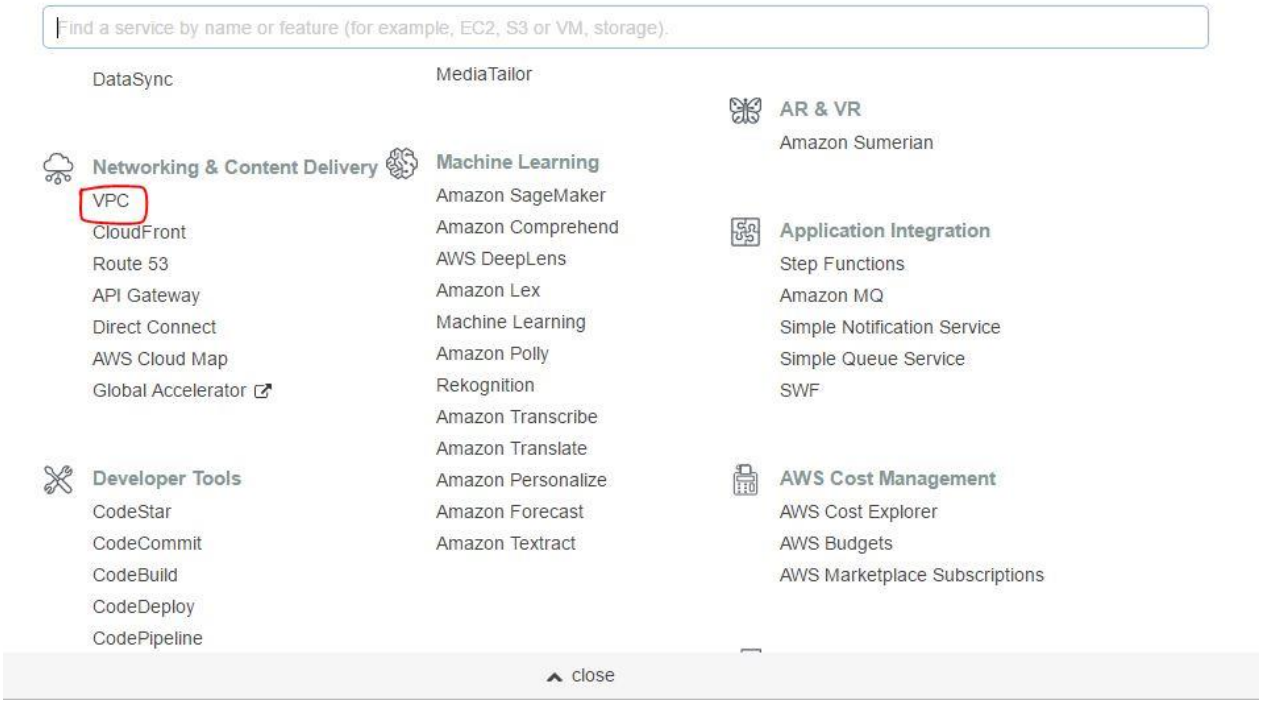
Virtual Private Network(VPN):

A VPN makes the private network (such as a company network) of an entity accessible through public infrastructure, primarily the internet. A VPN can allow users to exchange data efficiently across shared or public networks, as though they are directly linked to the private network.

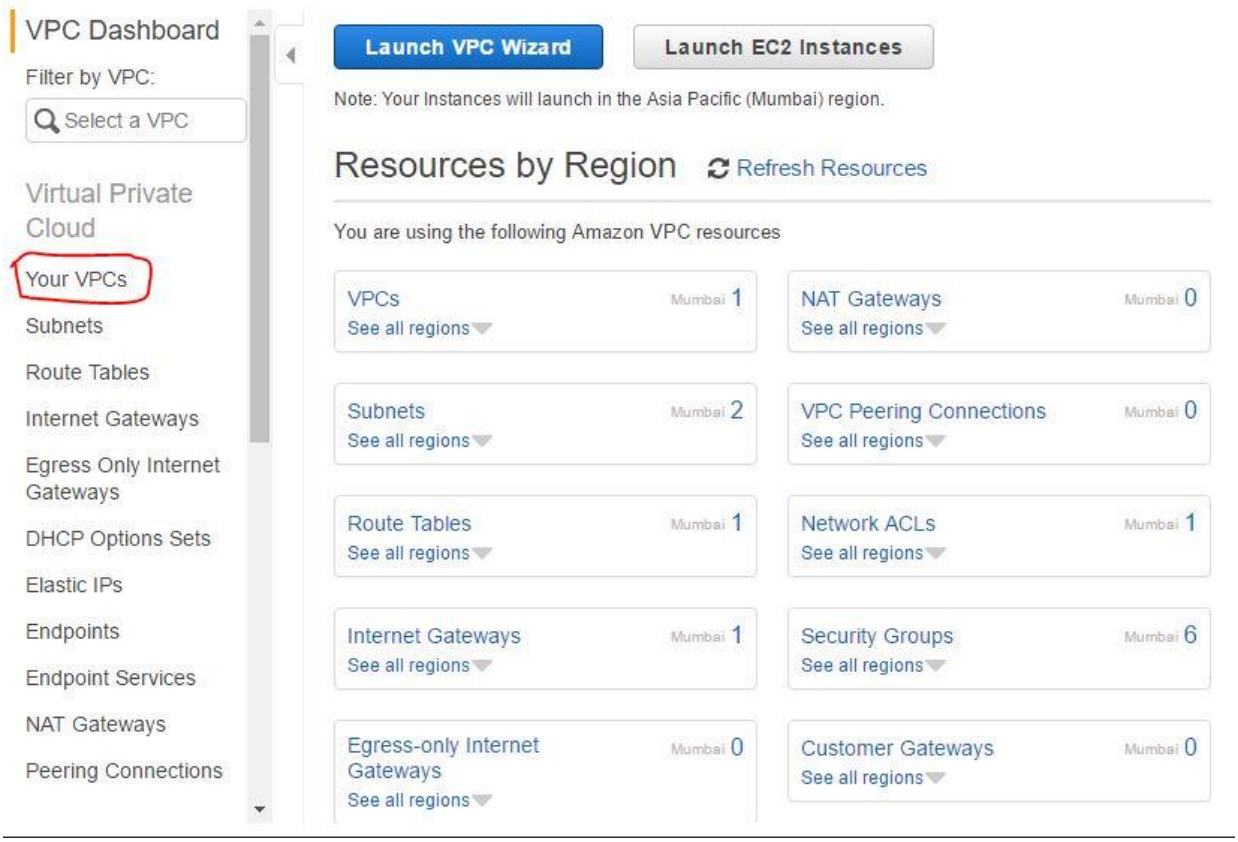
You can create an IPsec VPN connection between your VPC and your remote network. On the AWS side of the Site-to-Site VPN connection, a *virtual private gateway* provides two VPN endpoints (tunnels) for automatic failover. You configure your *customer gateway* on the remote side of the Site-to-Site VPN connection.

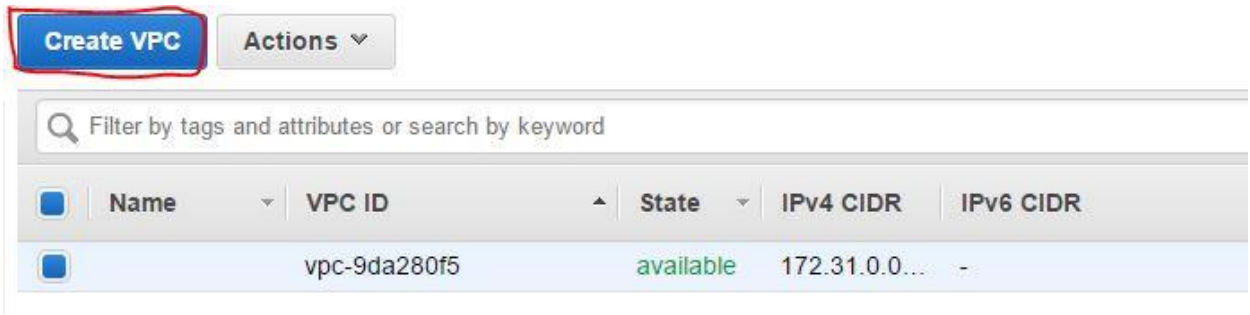
VPC Lab

1st step will be going into the **AWS Management Console > Services > Networking and Content Delivery > VPC**

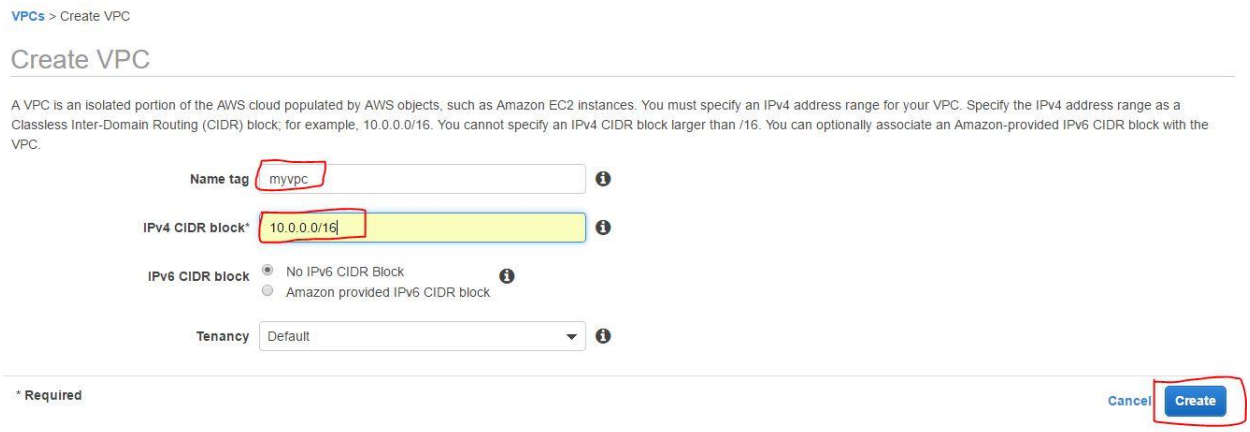


Now click on **Your VPCs > Create VPC**

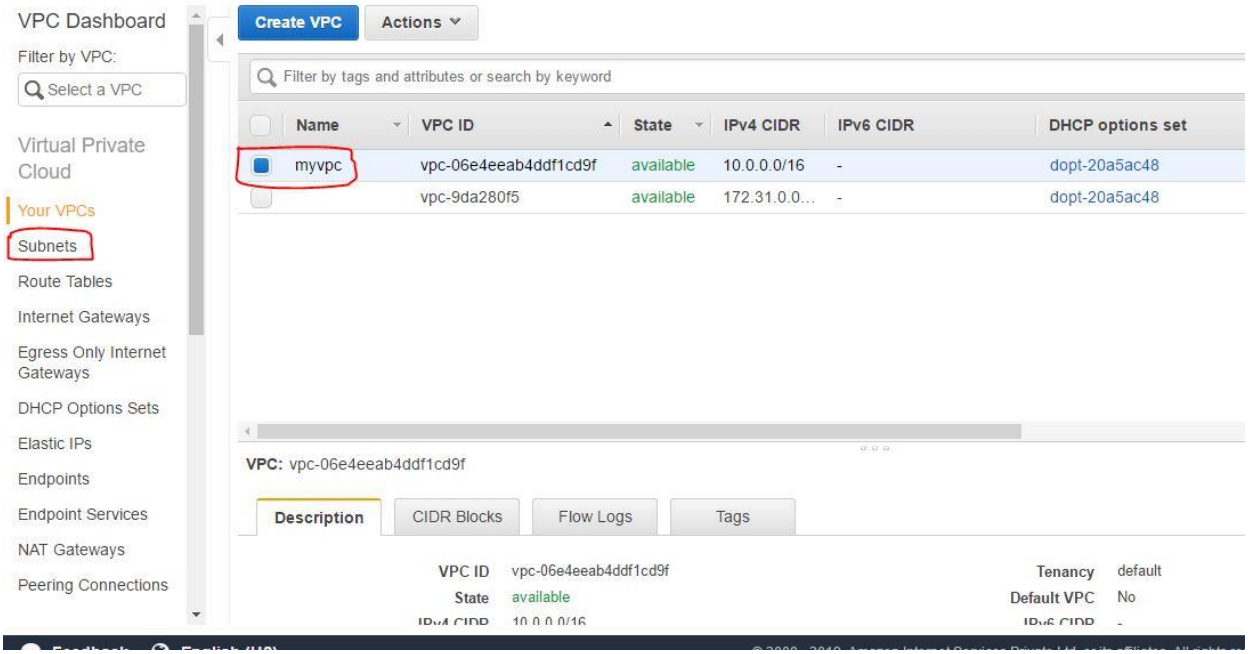




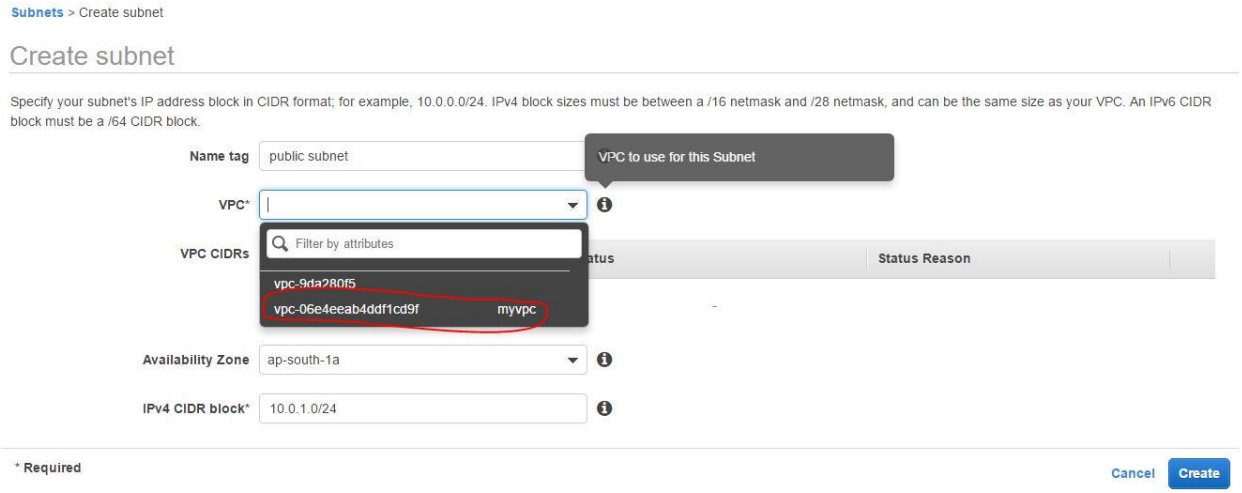
Next provide the **Name for your VPC > IP Address > Create**



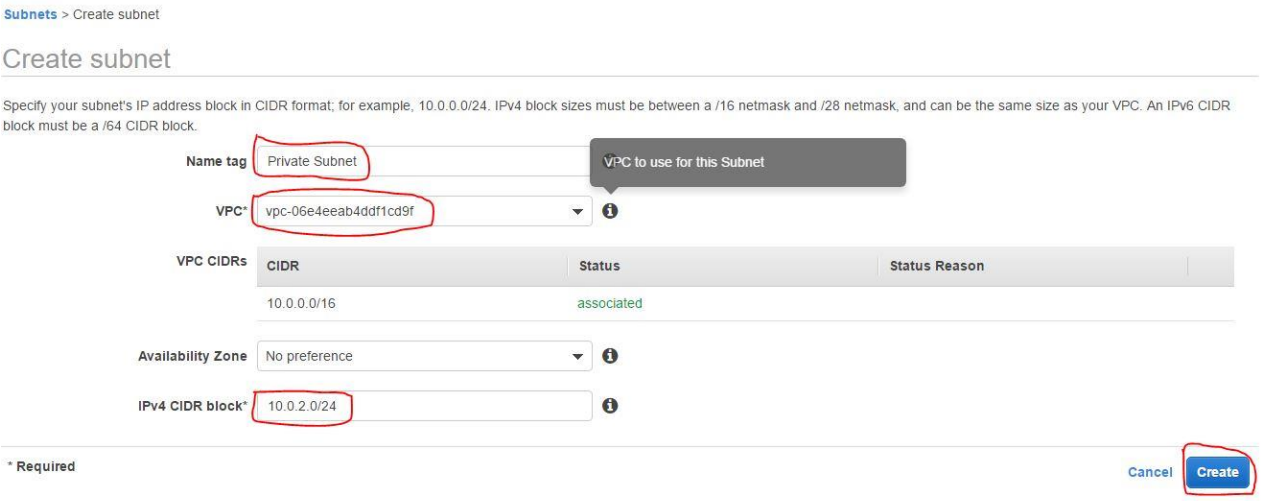
Next click on **Subnets from the Navigation pane >Create subnet**



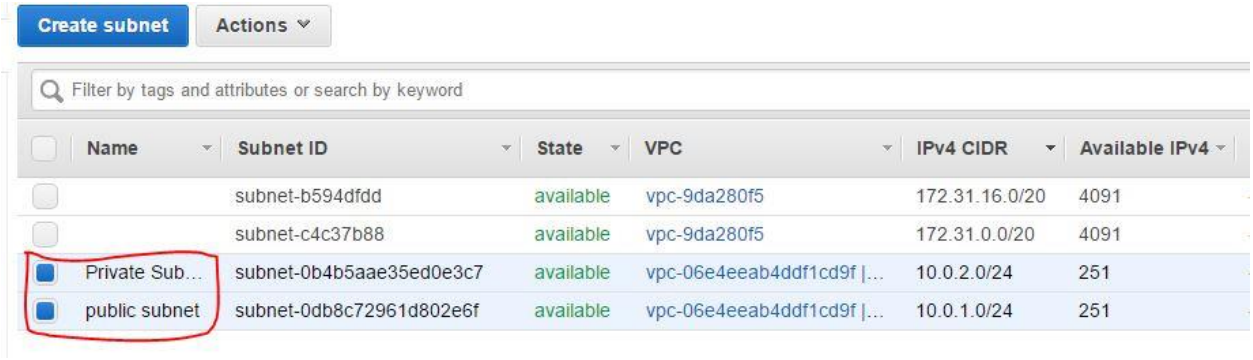
Next In the Create Subnet page **provide the name of your Subnet > IP Address > Availability Zone**



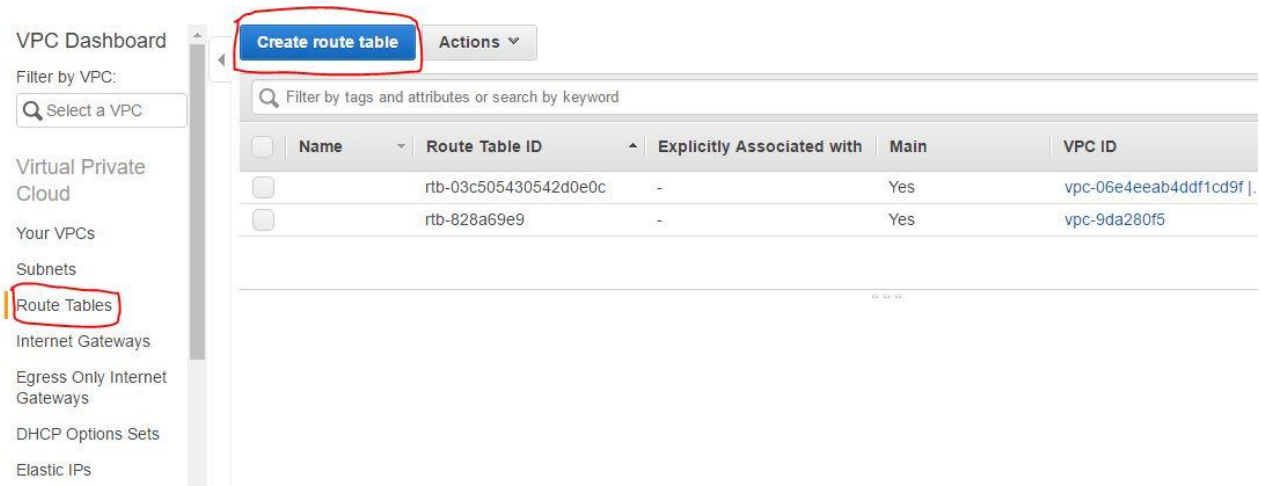
Again click on **Create Subnet > Now provide the name(as private) > IP Address > Availability Zone(2nd One) > Create**



Now you can see your both subnets has been created (Public as well as Private) subnets.



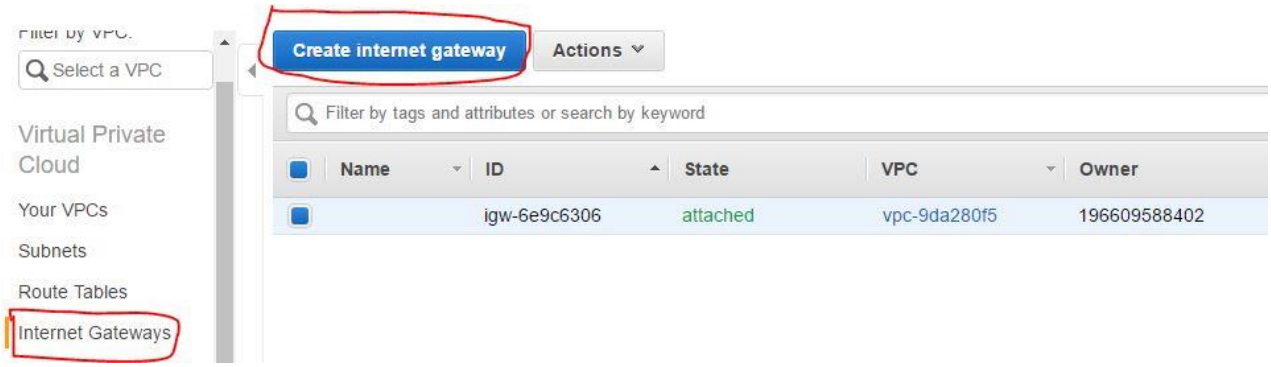
Next click on the Route Table from the navigation pane > click on Create Route Table



In Create Route Table page provide the name of the Route table as publicRT and choose the VPC which you have created and click on Create.

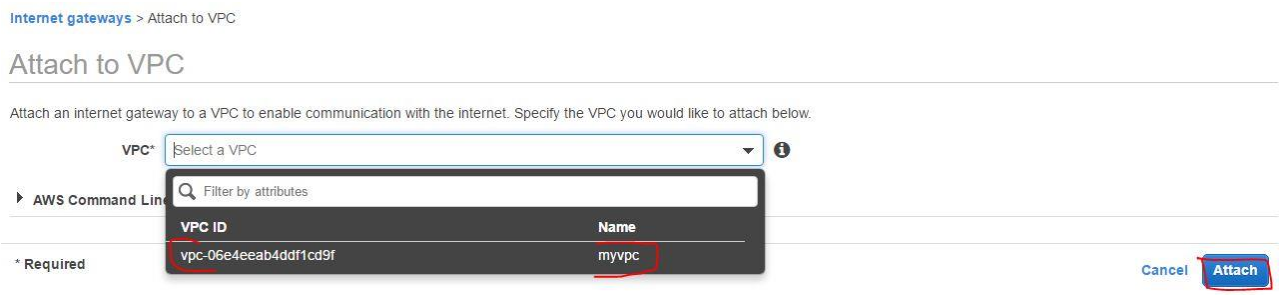
Repeat this same thing again by click on Create Route Table and providing **the name as privateRT > VPC created > Create**

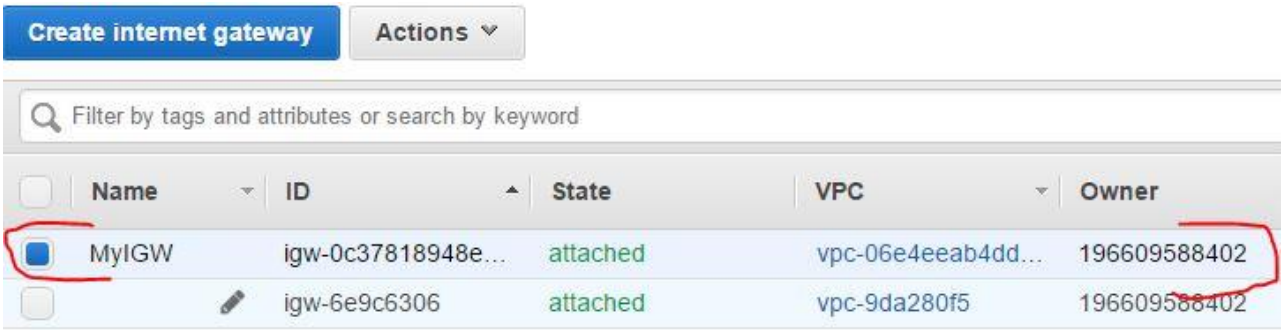
Next choose Internet Gateway from the navigation pane > Create internet gateway



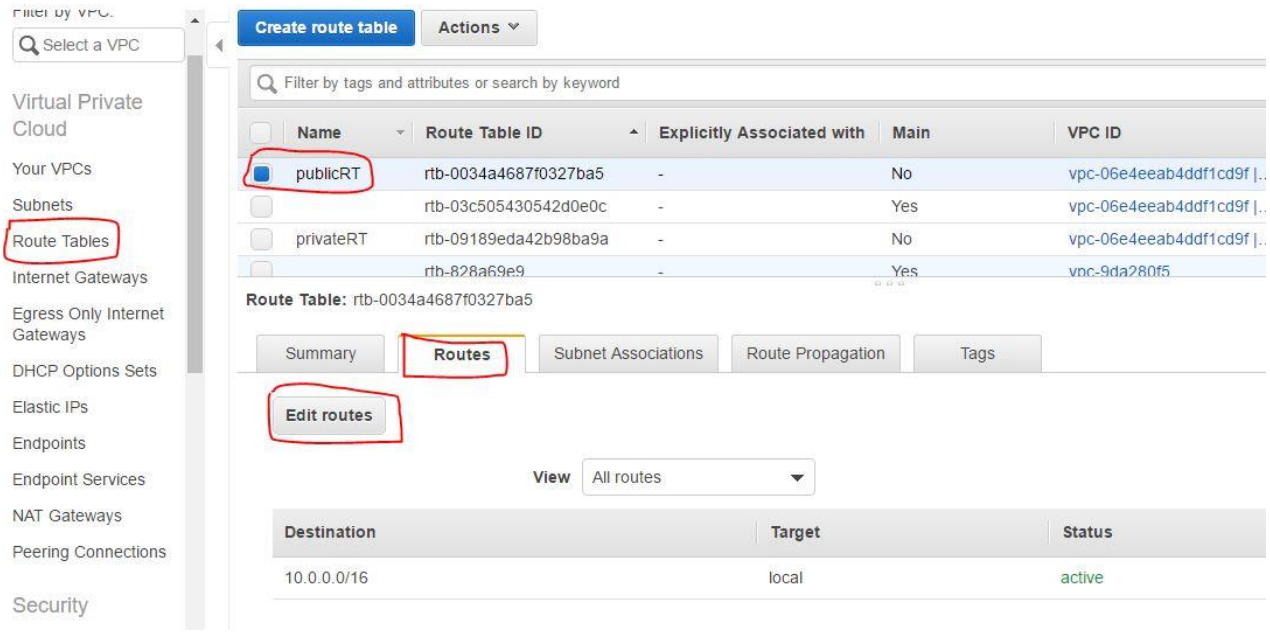
Provide the name of the Internet gateway > Create

Now as you have created your Internet gateway, it needs to be attached to some VPC, for that **click on the newly created Internet gateway > Actions > Attach to VPC > choose the newly created VPC > Attach**

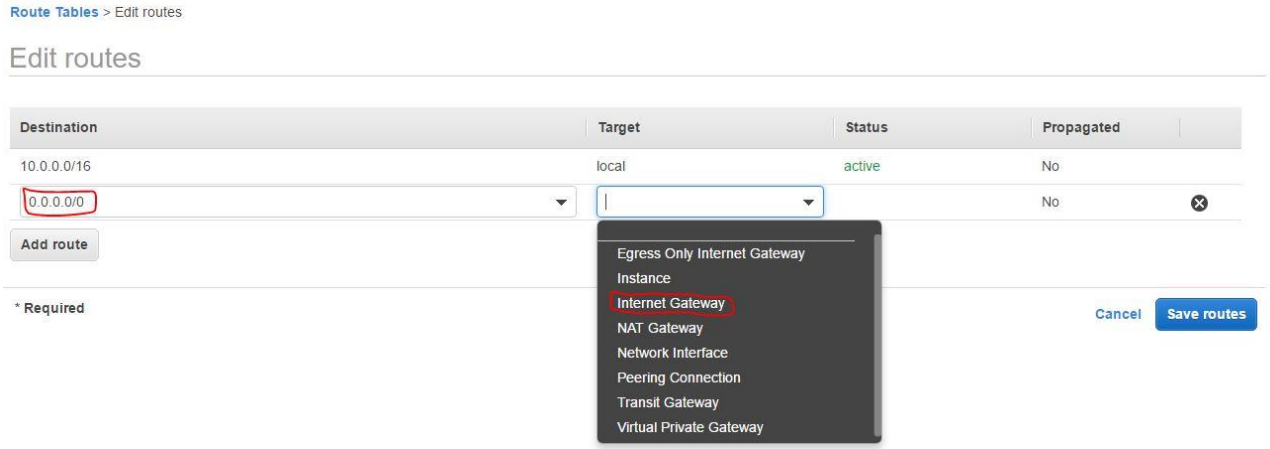


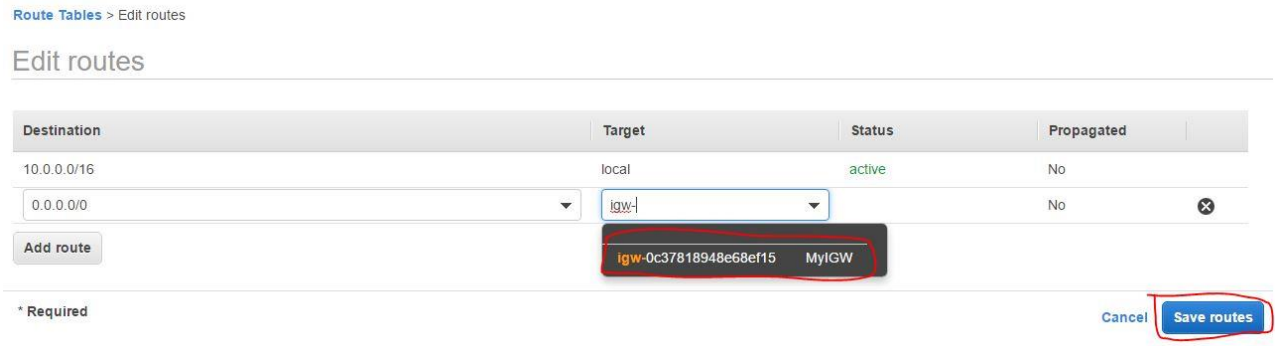


Next move to **Route Tables > choose publicRT > Routes > Edit Routes**

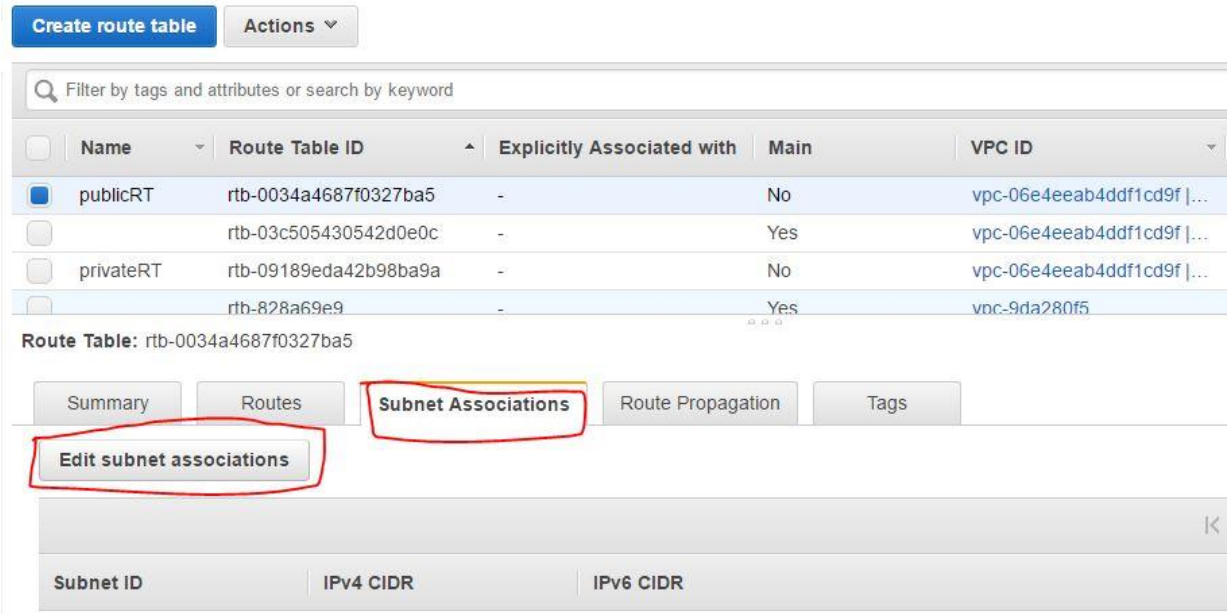


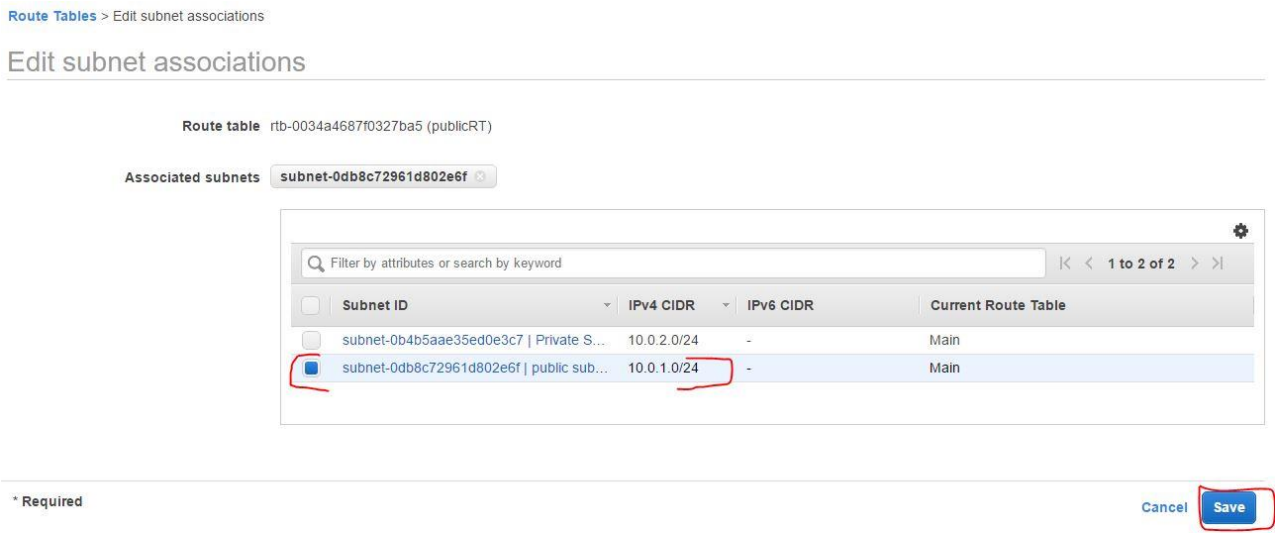
Now in that click on **Add route > In Destination provide value 0.0.0.0/0 > In target choose Internet Gateway and choose the one you created > Save Routes**

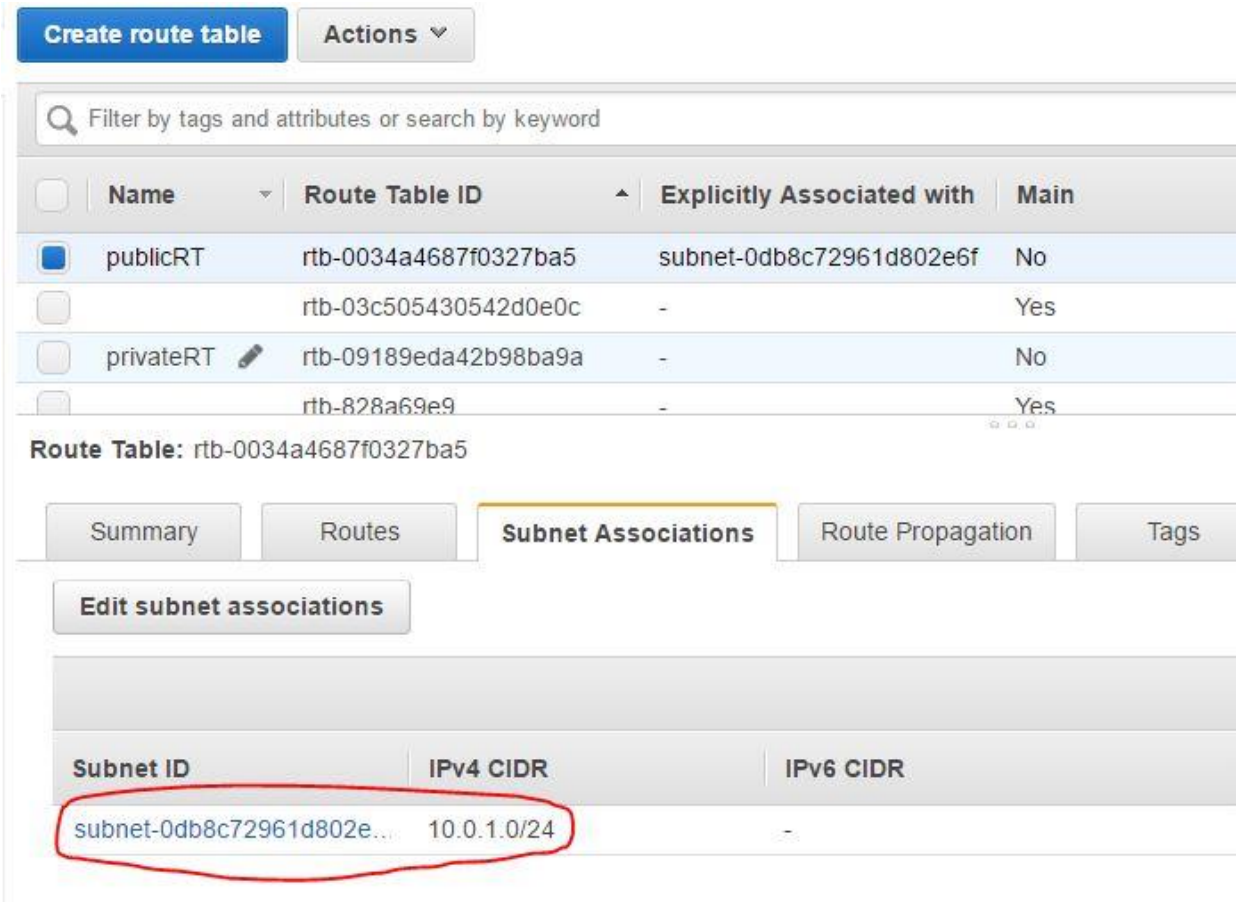




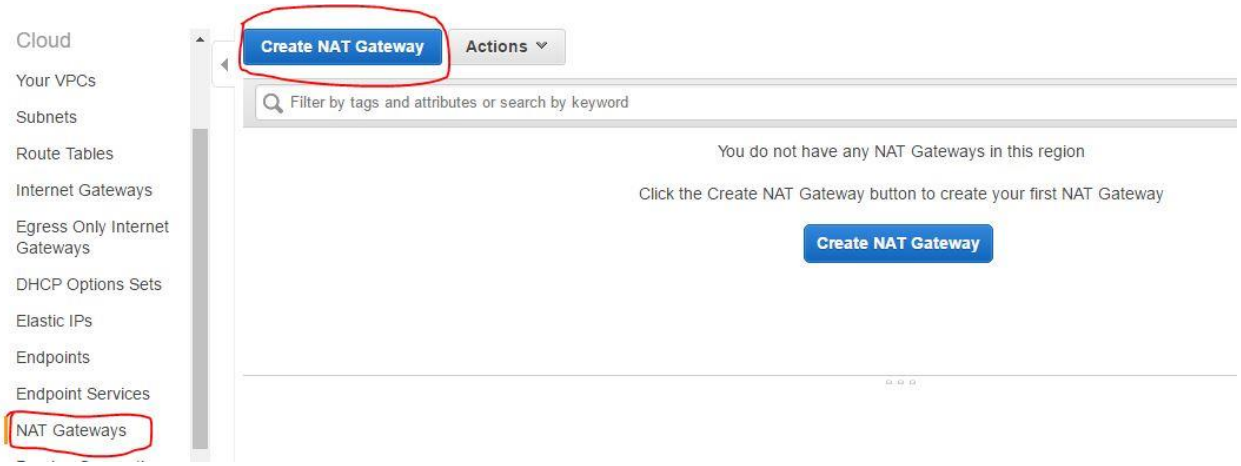
Next in same **publicRT click on Subnet Associations > Edit subnet associations > choose the public Subnet > Save**

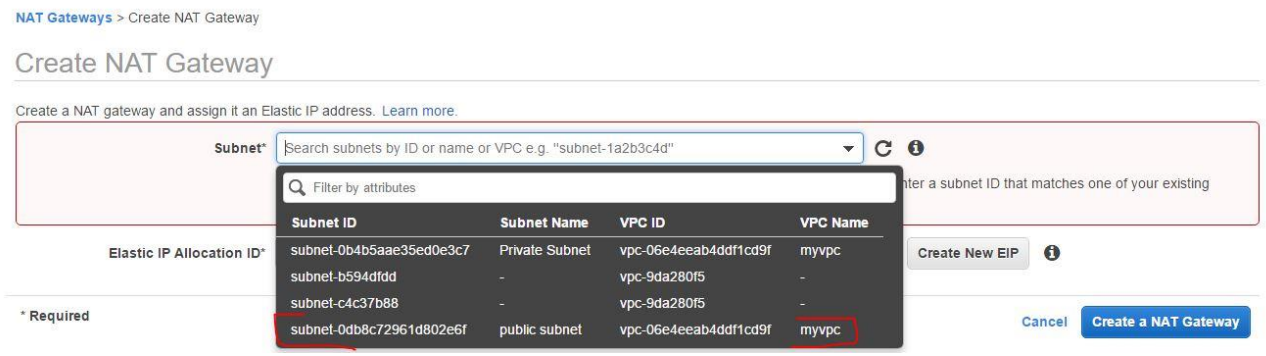




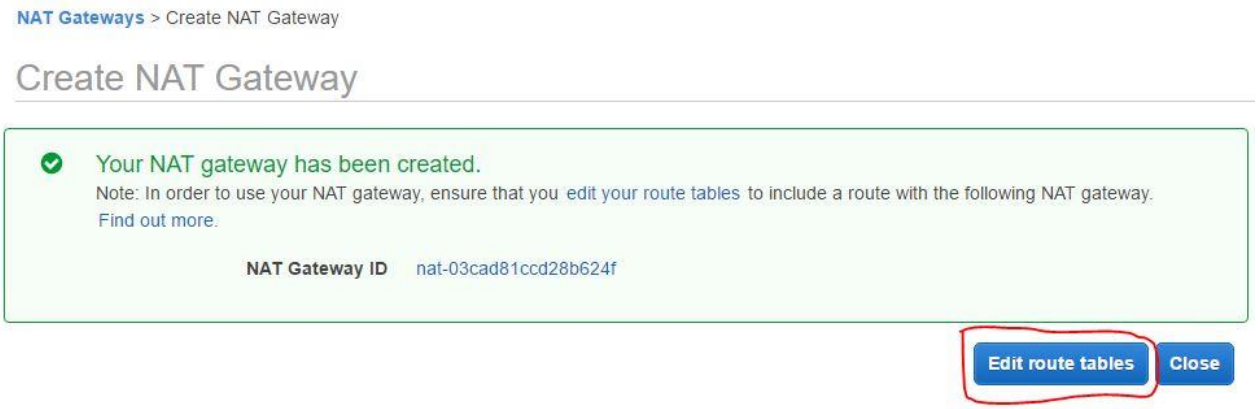


Next click on Nat Gateways from the navigation pane > Create NAT Gateway > choose the public subnet > Create Elastic IP > click on Create a Nat Gateway

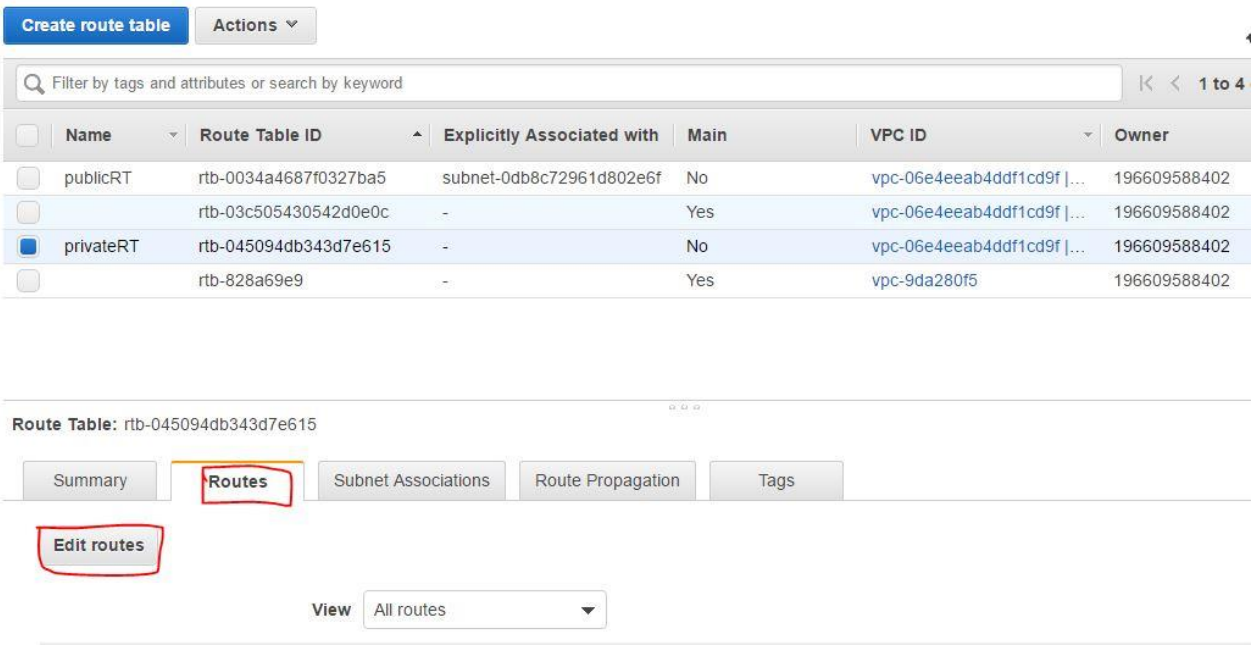




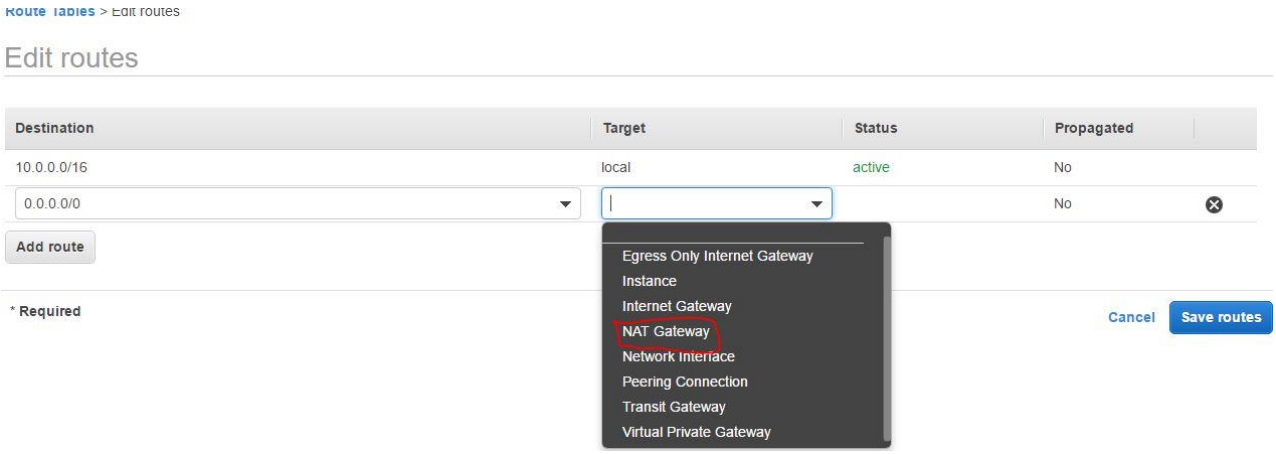
Once its created instead of Close click on Edit route tables

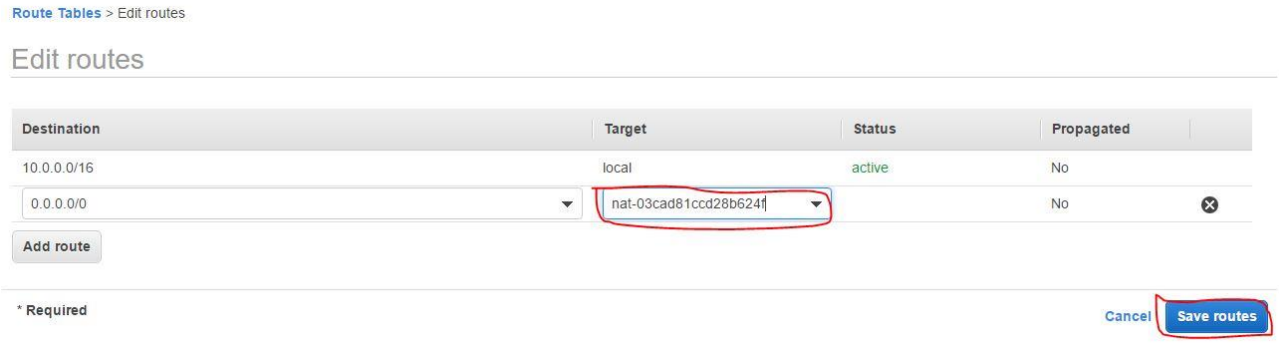


In Route tables click on privateRT > Routes > Edit Routes

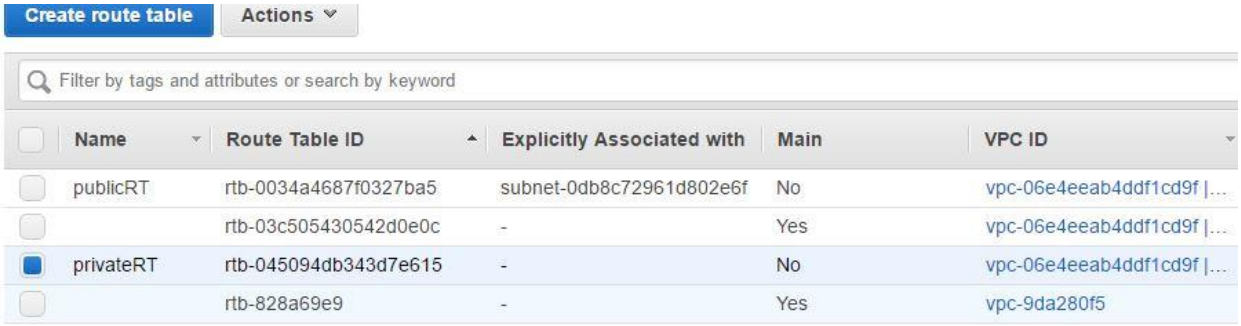


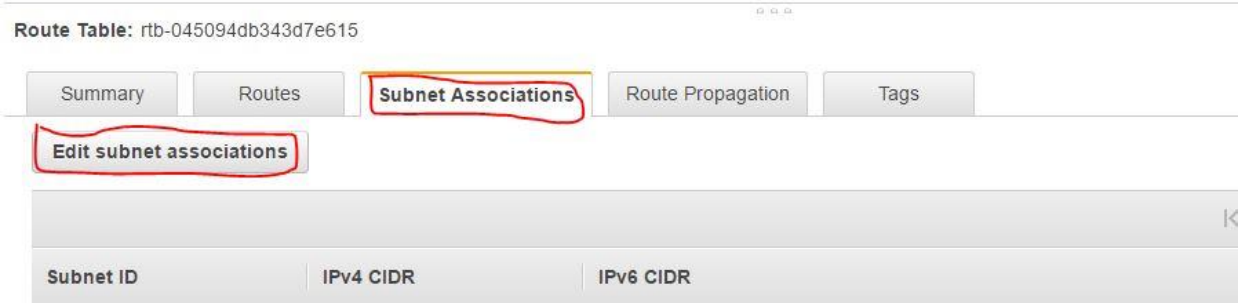
Click on Add Route > In Destination provide the value 0.0.0.0/0 > In target provide Nat Gateway > Select the newly created one > Save routes

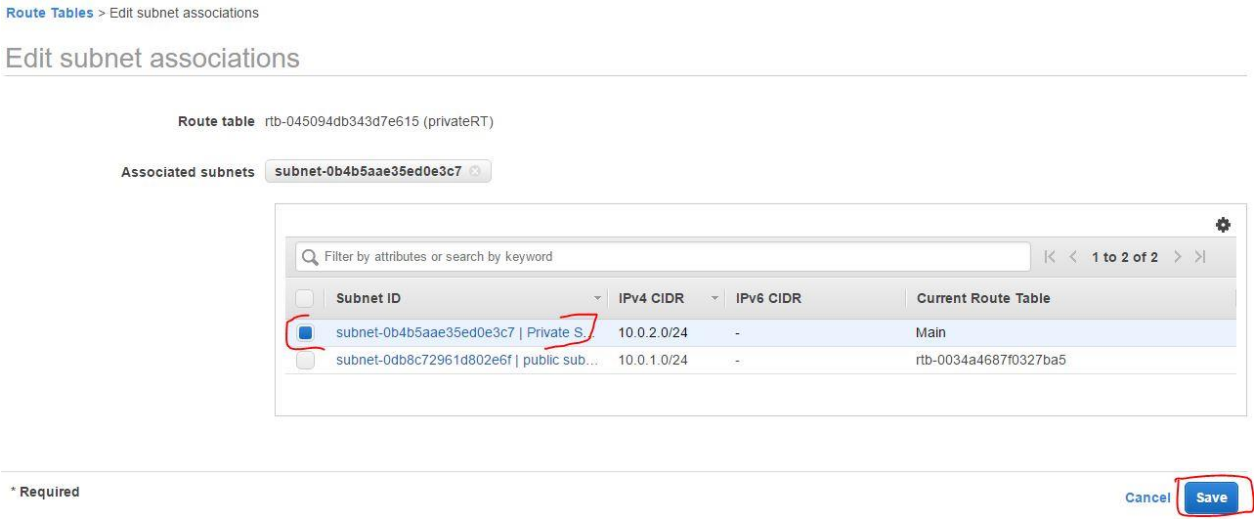




Next click on the Subnet Associations > Edit subnet associations > choose the private subnet > Save



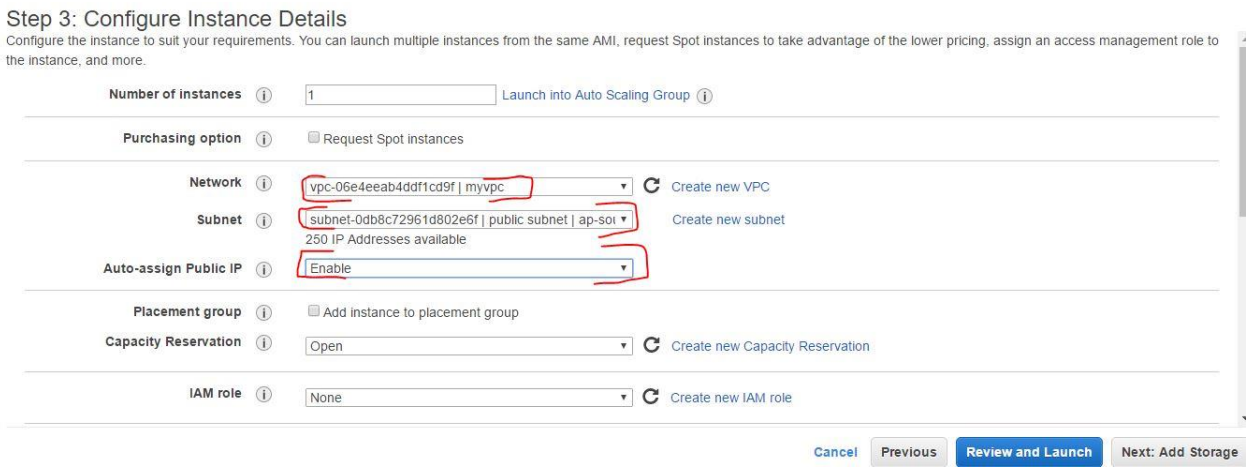




Now we have covered our VPC configuration, our main objective here is to create 2 instances where in 1 Instance we will be attaching it to Public Subnet and the other one to the Private Subnet, checking the Internet Connection for the Public Instance which has been connected to the public Subnet.

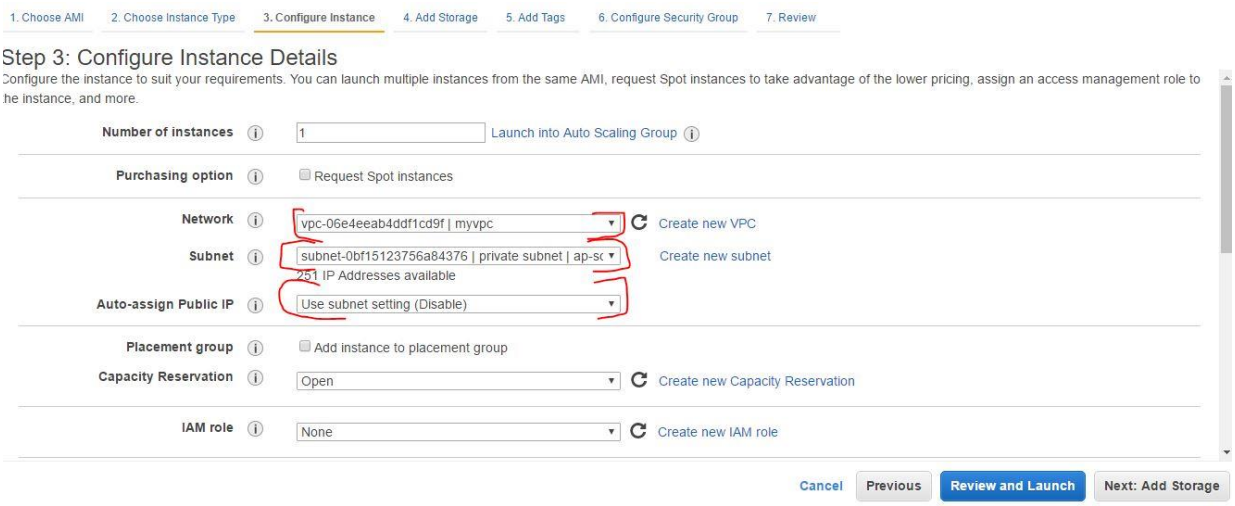
Further for private Instance which have been attached to the Private Subnet, will not be providing Public IP to it, so we have to enter through our public Instance and also to check if Internet connection is available to it or not.

For this lets create the 1st instance attaching the VPC you created > choose the Public Subnet > Enable Auto assign public Ip



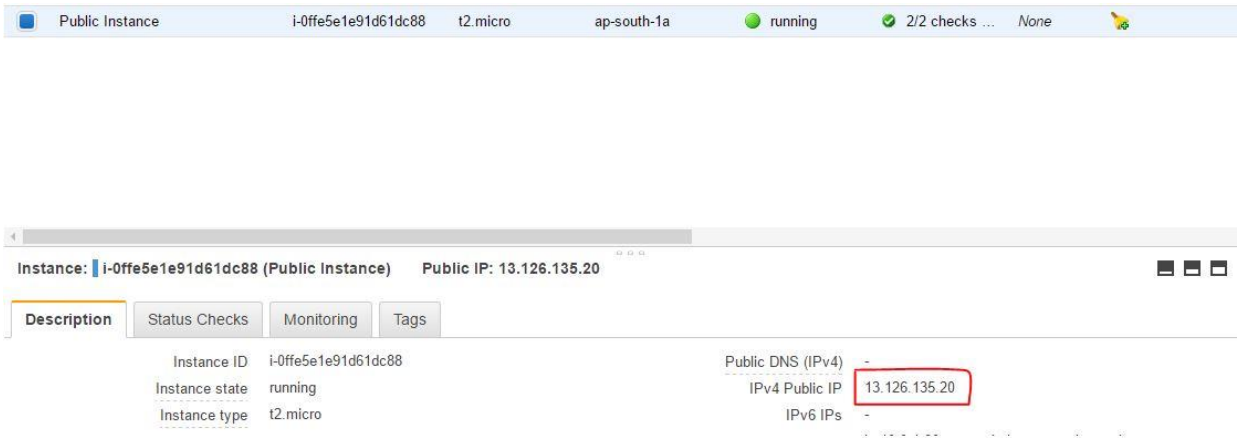
Create a new security key for the instance and launch the usual way.

Next create 2nd Instance, in Configure Instance Details provide Network as the VPC created > Subnet > Private Subnet > Auto-assign Public IP > Subnet setting-Disable



Next go to your Public Instance > copy the Public IP > enter into the Console > Use command "yum update -y"

Here you will find out internet is available to this instance.



```
root@ip-10-0-1-90:~
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating   : 1:openssl-libs-1.0.2k-16.amzn2.0.3.x86_64                    1/5
  Updating   : 1:openssl-1.0.2k-16.amzn2.0.3.x86_64                         2/5
  Installing : kernel-4.14.101-91.76.amzn2.x86_64                           3/5
  Cleanup    : 1:openssl-1.0.2k-16.amzn2.0.2.x86_64                         4/5
  Cleanup    : 1:openssl-libs-1.0.2k-16.amzn2.0.2.x86_64                    5/5
  Verifying  : 1:openssl-libs-1.0.2k-16.amzn2.0.3.x86_64                    1/5
  Verifying  : kernel-4.14.101-91.76.amzn2.x86_64                           2/5
  Verifying  : 1:openssl-1.0.2k-16.amzn2.0.3.x86_64                         3/5
  Verifying  : 1:openssl-libs-1.0.2k-16.amzn2.0.2.x86_64                    4/5
  Verifying  : 1:openssl-1.0.2k-16.amzn2.0.2.x86_64                         5/5

Installed:
  kernel.x86_64 0:4.14.101-91.76.amzn2

Updated:
  openssl.x86_64 1:1.0.2k-16.amzn2.0.3
  openssl-libs.x86_64 1:1.0.2k-16.amzn2.0.3

Complete!
[root@ip-10-0-1-90 ~]#
```

Now we need to enter into the private instance for which we don't have any public IP assigned, for this you need to go the pem file of your key attached to the private Instance > Open the file > copy the details > In console of public Instance > create on file with the same name > paste the content of the pem file > save

testlinux1.pem - Notepad

File Edit Format View Help


```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAg1EVBv7shKCH7Q9nH6OFhYK7tgxUNEayJ7thPzytI+SzIz7k4P9bWhg+Bi8/
6L+e95HDqJR3NwIf6EnjzQ2hoDSPRfiD7a+BdS+0o0J31ZXm0F861wKOCvQHVVfnRTFN6tj1DqKH
W36rGBnSVVIqoDLF5v34sv3U0WhCOCm149XRfcLbLoy7qnyDkocs/stkY2MoPB/hdYm5/A/BPW1G
IaRrGr0VnhQZNUJxxK8jOCp4nhHeEz8G/Z11M9X8eysK/GcssguBKF6UxnbOGrhj1oySIdYy9dAM
7c+uukXLBsnyP3EhrQKIW2hzGV/RuesdCqeCfBHwRN+B1I/20XvotwIDAQABAoIBAA6GXG4Oqnit
DPvitURhoPY41Ky5RmMV4sTUTBhjjL8on8vpPAXLGP6pKcsjVM1YzKuTmvG9okYaXQ8jhSf8NwFy
btqNd4JF5Gri4Pch9gFQ+FvZ/LclmDKL9XeyJmmX34d3gfuhbJlpNGGREZvQX9iiZRN8yhJ6Uxzq
jFCMBItzxYuvvc1fOv4jRDsd4mdZ8M3q2q2cv593G9Ltf/79+Lnp6tbr8rFl3lCsAxTtiGUbRAB/
nHo8gMEpMHcyF1AEYlEO3ALfsefNbL+Gg4EW2vGp8x3/83ICrum0GdnW0UVNIXfnH2UIcHRFbQbn
/UHhKV956AotyTUTruv7BoplxkkCgYEAuvZf1NoFNPkER+IsfWjLxWcUF5oGdQahYohGdk7pVIJa
c9yj/8LvyEW9FxuKLtJLHk18VePkV3nbDDy8p3UFTUnHoVymmYQ6LsH+esjgJ5S2pNOqb2Y6SYm+
mDBkqqia06yYGnVDU0AHn4Hcw2IxMFqIWOq73aQZFUP1qy20bbsCgYEAs86AY8UvDHRLZ7ai4xo9
6V1L5qB+vA++GCqH9lWaBpp/p1R1xB/YbxNjviVThigVNGUCjQdOts0qdej9D9987CgubVxBdMdN
8fAnjvsrHnfWHHz/HBj057dyC0ZwZXnAReng/7m2d6aqabSBUd64twWKFCDdCKyJ9qgLiHAIAzUC
gYAjHKoau2dG741SQrPASbIbPKSVWiPkTn54lsdE6MUBTgnWwJU8L0wA/LfqYw2shWn+Iuc66jW0
B/qPbciwWlXIkhafw1q2chLLPTuhq6ANTuyOQAI+MaC4ttnIky4Urt2E62EVdvhR5jCIUEc5mnNK
dXYt3MezdGIRkFXO9CJyiQKBgQCqS2xCEiFLVGGOauW+3b3BZgFa/qrJhsdtr+pxngVxN6arXkzj
YZ3lJaIUFtoSd6gNpZVS6Jbef1Y1smLrQ6nXKP8/yF6eMZ1bBS8SNkRSaQE6nhAoFmLti0rR7vbI
xRUbojTdPpDYpdtEnhbU6XJctTxfb/ioHHHWULYLlL+h+QKBgGonXaBEGH3IUYuE87eSiwPnudqb
8XsLTiDsLeLxbwaHxrmJfbYaRr7GzNHU34BEPK4n0cAbulcyMOSCqc1afBGGUJCWMeVYs6giqVBY
o5WHmJnygs7v4SzD5b33IZjV3S6a7G3PJjbS5Tx6M37j0I1iKgCzSdsk6C0Y0DXCZ4Sy
-----END RSA PRIVATE KEY-----
```


root@ip-10-0-1-90:~


```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAg1EVBv7shKCH7Q9nH6OFhYK7tgxUNEayJ7thPzytI+SzIz7k4P9bWhg+Bi8/
6L+e95HDqJR3NwIf6EnjzQ2hoDSPRfiD7a+BdS+0o0J31ZXm0F861wKOCvQHVVfnRTFN6tj1DqKH
W36rGBnSVVIqoDLF5v34sv3U0WhCOCm149XRfcLbLoy7qnyDkocs/stkY2MoPB/hdYm5/A/BPW1G
IaRrGr0VnhQZNUJxxK8jOCp4nhHeEz8G/Z11M9X8eysK/GcssguBKF6UxnbOGrhj1oySIdYy9dAM
7c+uukXLBsnyP3EhrQKIW2hzGV/RuesdCqeCfBHwRN+B1I/20XvotwIDAQABAoIBAA6GXG4Oqnit
DPvitURhoPY41Ky5RmMV4sTUTBhjjL8on8vpPAXLGP6pKcsjVM1YzKuTmvG9okYaXQ8jhSf8NwFy
btqNd4JF5Gri4Pch9gFQ+FvZ/LclmDKL9XeyJmmX34d3gfuhbJlpNGGREZvQX9iiZRN8yhJ6Uxzq
jFCMBItzxYuvvc1fOv4jRDsd4mdZ8M3q2q2cv593G9Ltf/79+Lnp6tbr8rFl3lCsAxTtiGUbRAB/
nHo8gMEpMHcyF1AEYlEO3ALfsefNbL+Gg4EW2vGp8x3/83ICrum0GdnW0UVNIXfnH2UIcHRFbQbn
/UHhKV956AotyTUTruv7BoplxkkCgYEAuvZf1NoFNPkER+IsfWjLxWcUF5oGdQahYohGdk7pVIJa
c9yj/8LvyEW9FxuKLtJLHk18VePkV3nbDDy8p3UFTUnHoVymmYQ6LsH+esjgJ5S2pNOqb2Y6SYm+
mDBkqqia06yYGnVDU0AHn4Hcw2IxMFqIWOq73aQZFUP1qy20bbsCgYEAs86AY8UvDHRLZ7ai4xo9
6V1L5qB+vA++GCqH9lWaBpp/p1R1xB/YbxNjviVThigVNGUCjQdOts0qdej9D9987CgubVxBdMdN
8fAnjvsrHnfWHHz/HBj057dyC0ZwZXnAReng/7m2d6aqabSBUd64twWKFCDdCKyJ9qgLiHAIAzUC
gYAjHKoau2dG741SQrPASbIbPKSVWiPkTn54lsdE6MUBTgnWwJU8L0wA/LfqYw2shWn+Iuc66jW0
B/qPbciwWlXIkhafw1q2chLLPTuhq6ANTuyOQAI+MaC4ttnIky4Urt2E62EVdvhR5jCIUEc5mnNK
dXYt3MezdGIRkFXO9CJyiQKBgQCqS2xCEiFLVGGOauW+3b3BZgFa/qrJhsdtr+pxngVxN6arXkzj
YZ3lJaIUFtoSd6gNpZVS6Jbef1Y1smLrQ6nXKP8/yF6eMZ1bBS8SNkRSaQE6nhAoFmLti0rR7vbI
xRUbojTdPpDYpdtEnhbU6XJctTxfb/ioHHHWULYLlL+h+QKBgGonXaBEGH3IUYuE87eSiwPnudqb
8XsLTiDsLeLxbwaHxrmJfbYaRr7GzNHU34BEPK4n0cAbulcyMOSCqc1afBGGUJCWMeVYs6giqVBY
o5WHmJnygs7v4SzD5b33IZjV3S6a7G3PJjbS5Tx6M37j0I1iKgCzSdsk6C0Y0DXCZ4Sy
-----END RSA PRIVATE KEY-----
```

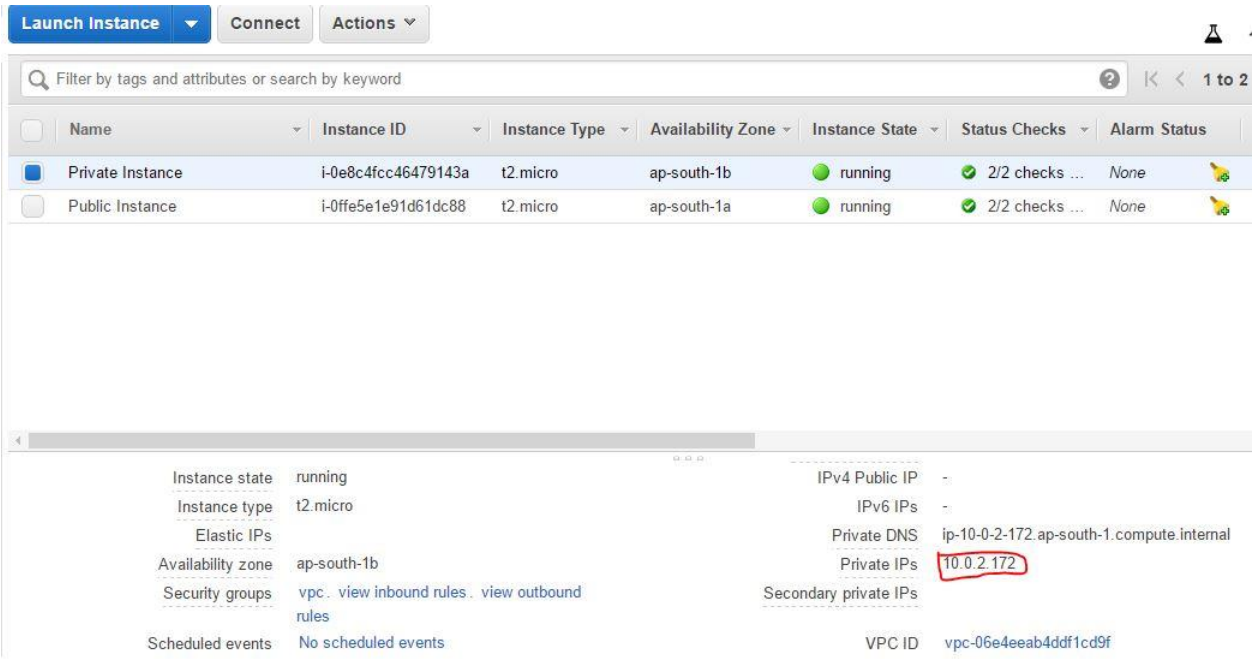

```
:wq!
```

Give permissions to the file created using the following command

chmod 400 testlinux1.pem

```
[root@ip-10-0-1-90 ~]# vi testlinux1.pem
[root@ip-10-0-1-90 ~]# chmod 400 testlinux1.pem
[root@ip-10-0-1-90 ~]#
```

Now copy the Private Instance Private IP Address



Next go to your console of Public Instance, type the command

ssh -i testlinux1.pem ec2-user@10.0.2.172

Note : testlinux1.pem is the IP Address which I used for my demonstration, you can provide your created file name and also the IP Address for the private Instance may be difference.

```
ec2-user@ip-10-0-2-172:~
  kernel.x86_64 0:4.14.101-91.76.amzn2

Updated:
  openssl.x86_64 1:1.0.2k-16.amzn2.0.3
  openssl-libs.x86_64 1:1.0.2k-16.amzn2.0.3

Complete!
[root@ip-10-0-1-90 ~]# vi testlinux1.pem
[root@ip-10-0-1-90 ~]# chmod 400 testlinux1.pem
[root@ip-10-0-1-90 ~]# ssh -i testlinux1.pem ec2-user@10.0.2.172
The authenticity of host '10.0.2.172 (10.0.2.172)' can't be established.
ECDSA key fingerprint is SHA256:BKY+5fjZKBdpmqjm0eqTTlLoy7PKOoR2YoCPHUOEObo.
ECDSA key fingerprint is MD5:45:ab:cb:bb:03:09:27:d0:0f:92:85:3e:de:e8:fb:9e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.172' (ECDSA) to the list of known hosts.

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

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 3 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-2-172 ~]$
```

Next check if internet is available to this instance or not by using update command or you can also use ping command, follow the image below

```
[ec2-user@ip-10-0-2-172 ~]$ ping www.google.com
PING www.google.com (172.217.160.164) 56(84) bytes of data.
64 bytes from bom05s12-in-f4.1e100.net (172.217.160.164): icmp_seq=1 ttl=52 time
=2.12 ms
64 bytes from bom05s12-in-f4.1e100.net (172.217.160.164): icmp_seq=2 ttl=52 time
=1.81 ms
64 bytes from bom05s12-in-f4.1e100.net (172.217.160.164): icmp_seq=3 ttl=52 time
=1.81 ms
64 bytes from bom05s12-in-f4.1e100.net (172.217.160.164): icmp_seq=4 ttl=52 time
=1.84 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.813/1.898/2.122/0.136 ms
```

Here we can see internet is available for the private instance.