**Task : Create VPC setup in Linux Environment.**

**1.Login Private Subnet through Public Subnet / Private Instance within Public Instance**

**2.Internet connectivity in Private Subnet**

**Steps to Create and set up a VPC in AWS environment:**

**=============================================**

**1. Create a own VPC.**

**2. Create a Public and Private subnet for different Available AZs by assigning diferent CIDR blocks.**

**3. Create Internet Gateway & attach it to the VPC.**

**4. Create Routing table [RT], One as Public & One as Private by associating the appropirate subnets to it.**

**5. Edit the Public route table's Route alone and map the IGW, not the Private and leave it as it is.**

**6. Create Two Security Groups - One for Public [Edit the Inbound rules with RDP, HTTP/HTTPS, SSH and map 0.0.0.0/0 in the source] &**

**One for Private [Edit the inbound rules and map the Security Grop of Public in the source].**

**7. Create Two EC2s one in public and one in private subnets with proper Security Groups.**

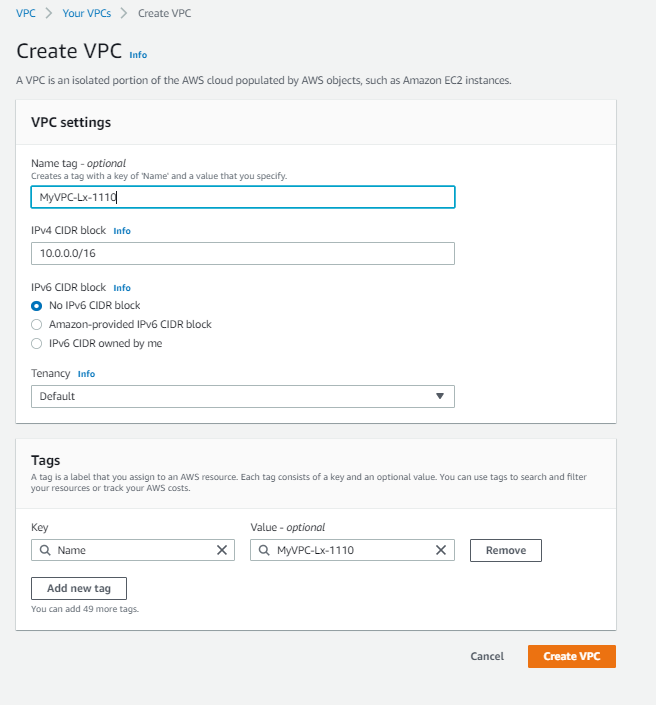
**8. Login into Public and check the internet connection.**

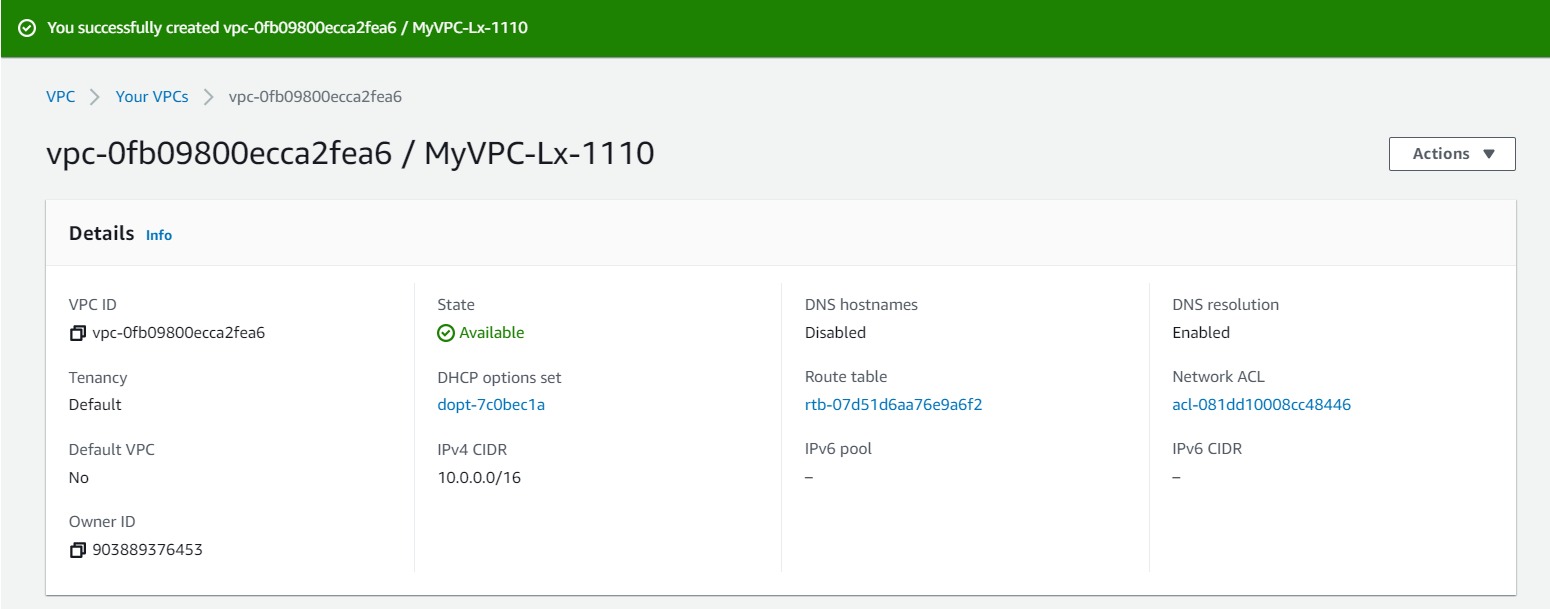
**9. Create NAT gateway with new Elastic IP for the internet connection in the Private Subnet. Map it to Private RT.**

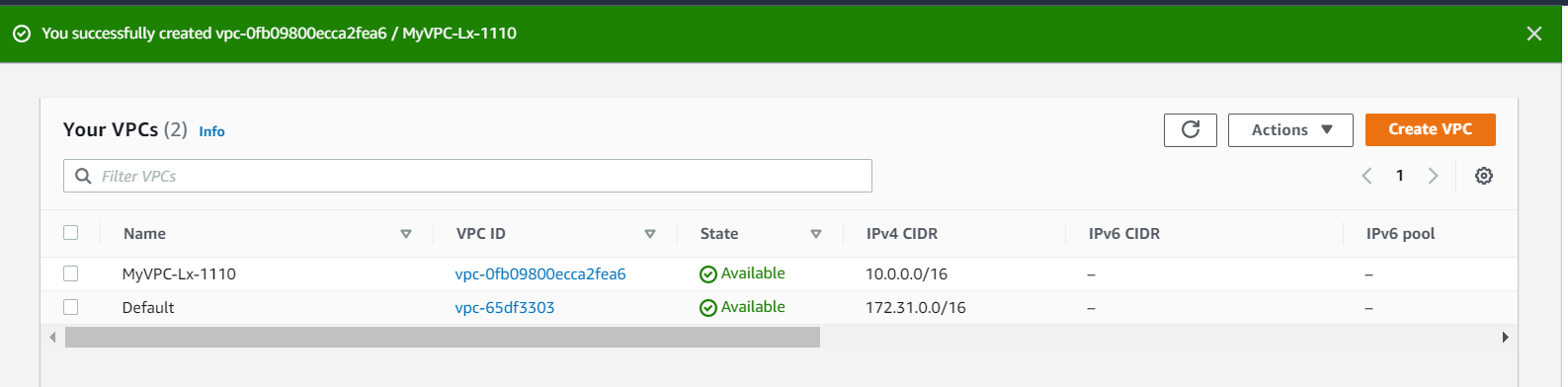
**10. Now login into the Private EC2 and verify the connectivity and Internet facility.**

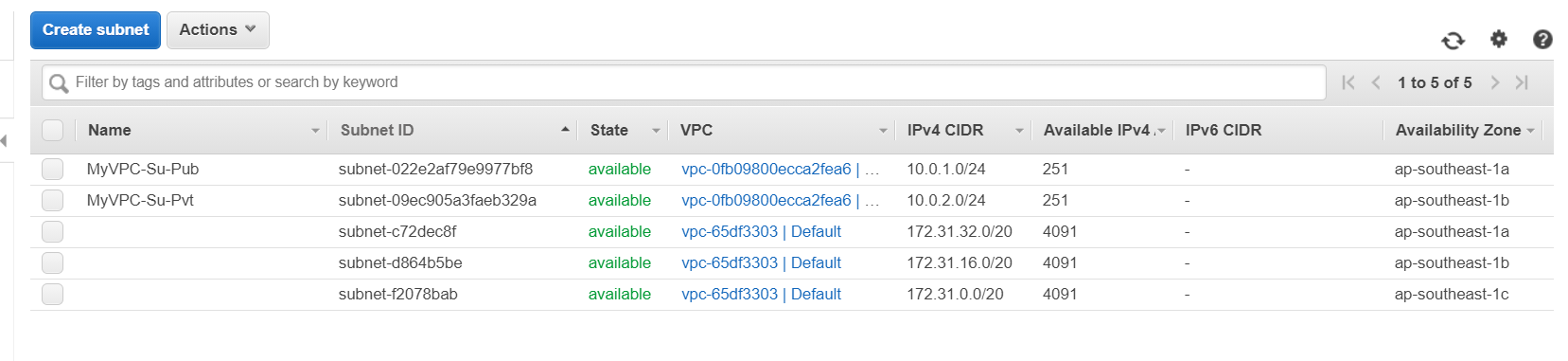
**Steps to Create and set up a VPC in AWS environment:**

**1. Create a own VPC.-10.0.0.0/16 (65536 ips)**



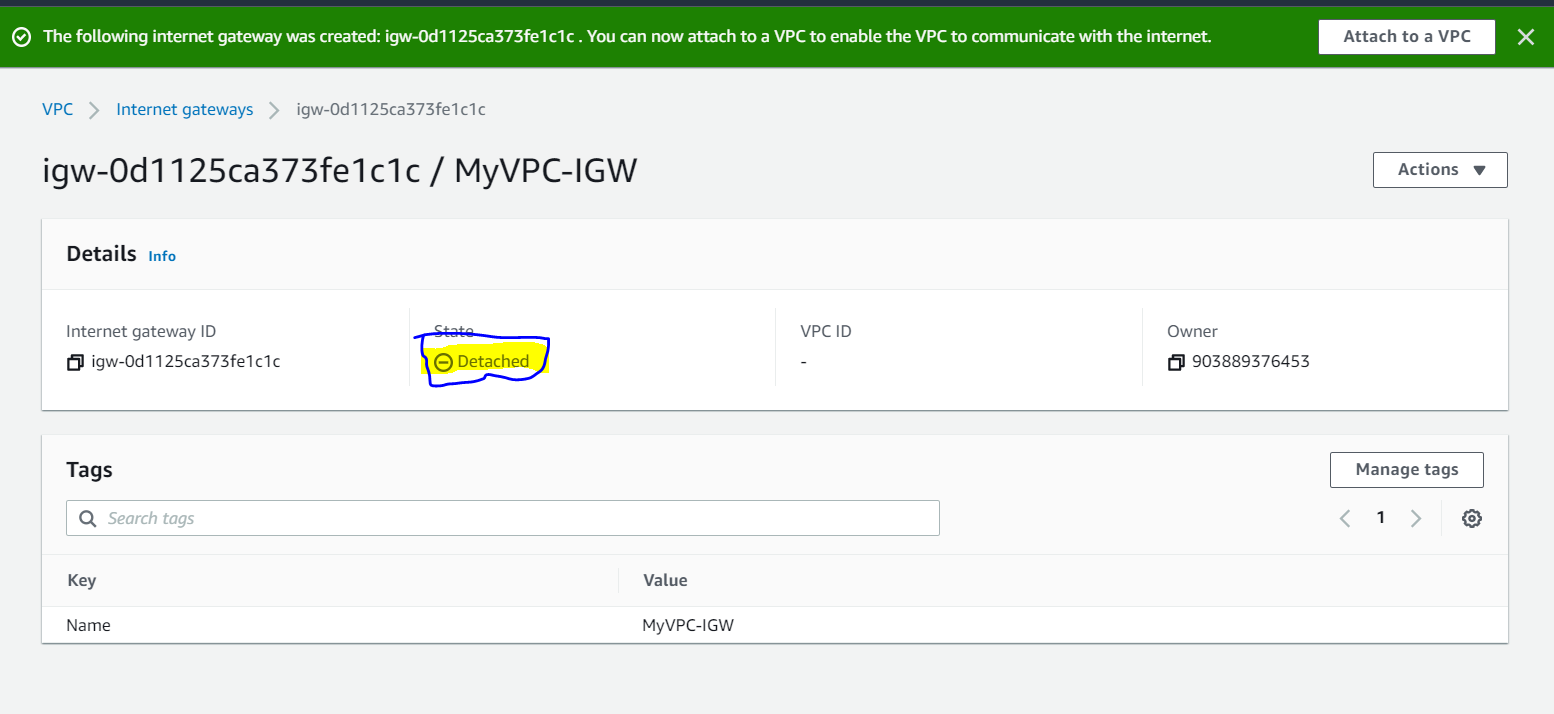


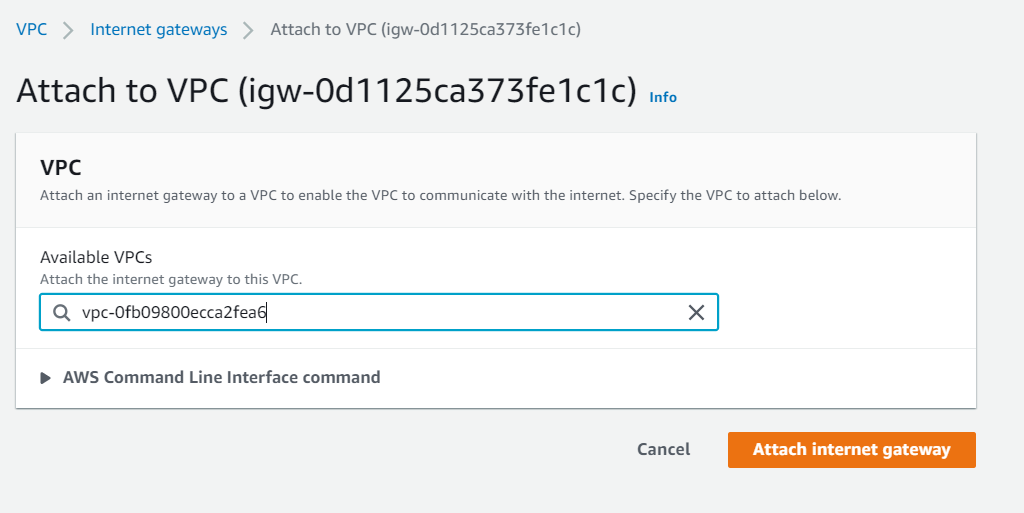
  
  
2. Create a Public and Private subnet(s) for different Available Zones by assigning different CIDR (Classless inter-domain routing ) blocks.

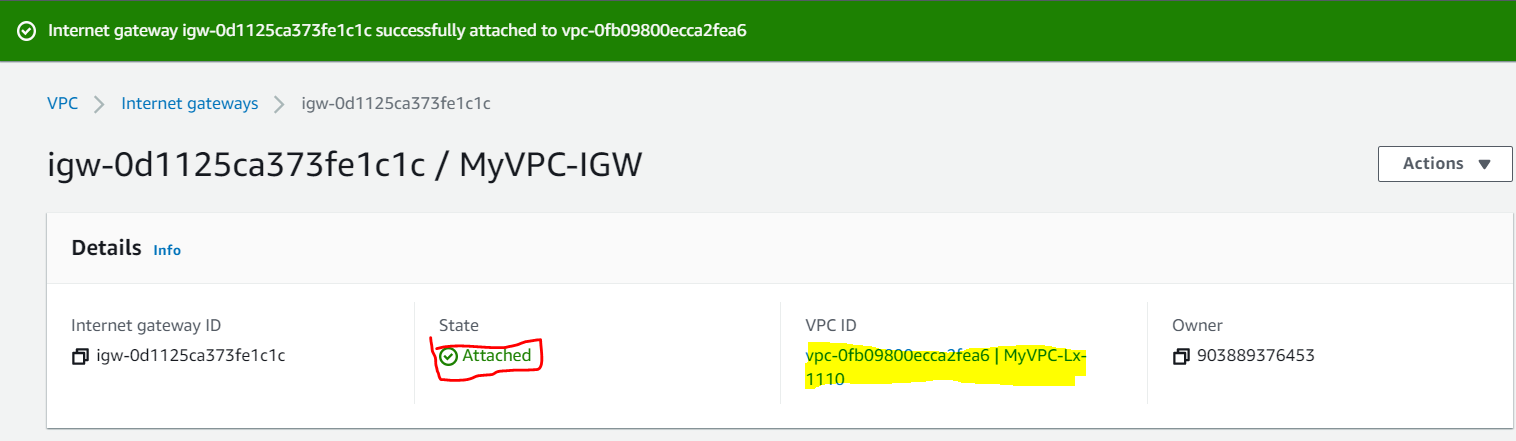
Public – 10.0.1.0/24 – Public zone (256 ips)  
Private – 10.0.2.0/24 – Private zone (256 ips)  
  


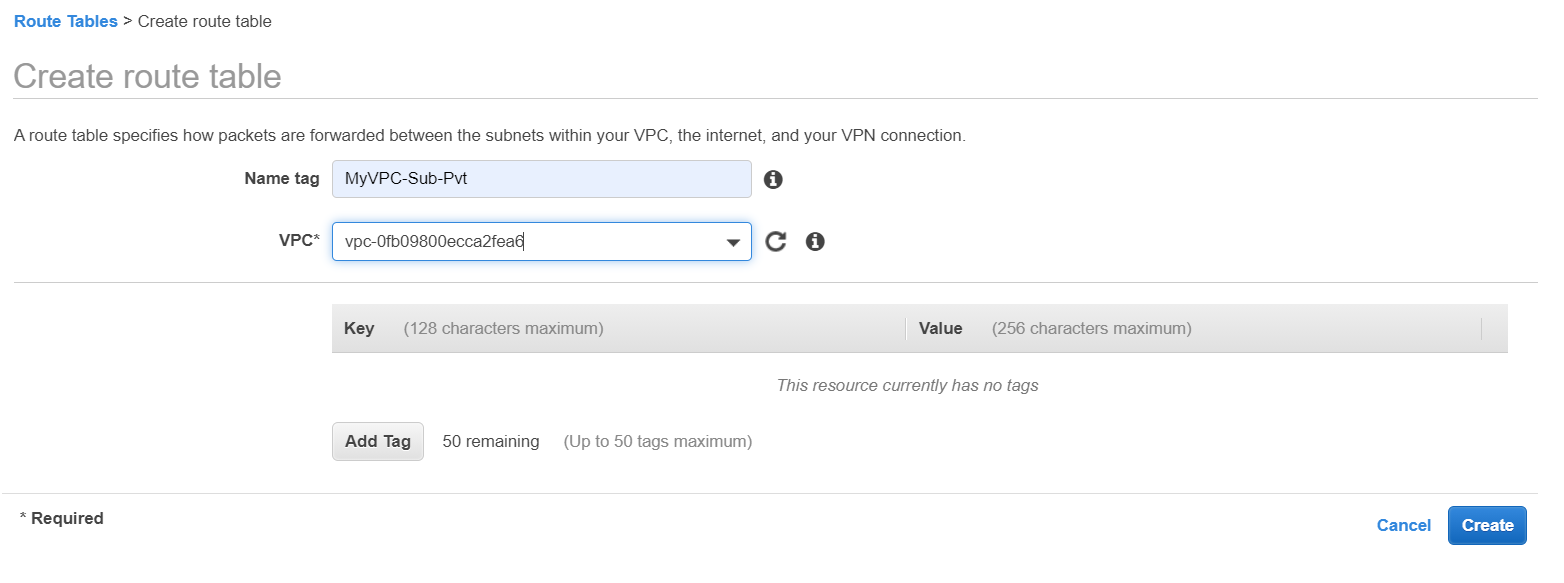
3. Create Internet Gateway & attach it to the VPC.(Own VPC)

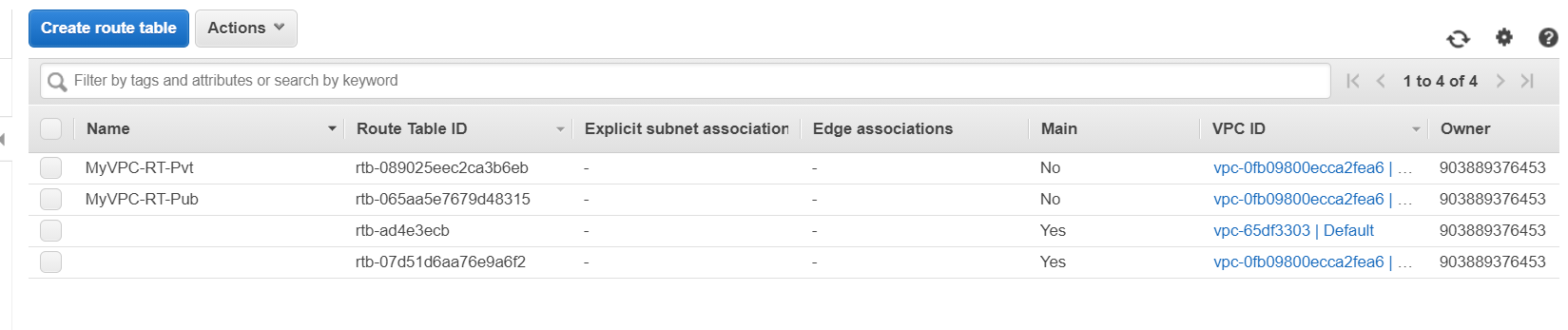
- Internet Supply from AWS(open internet)



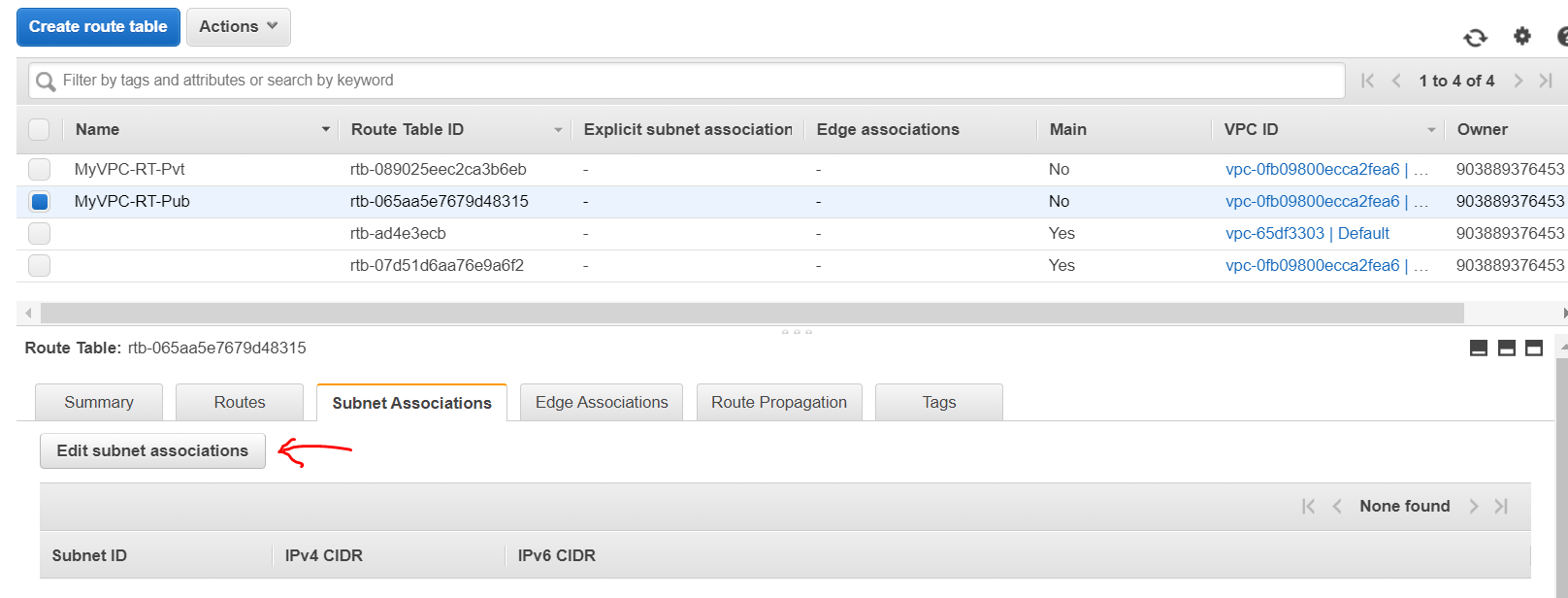


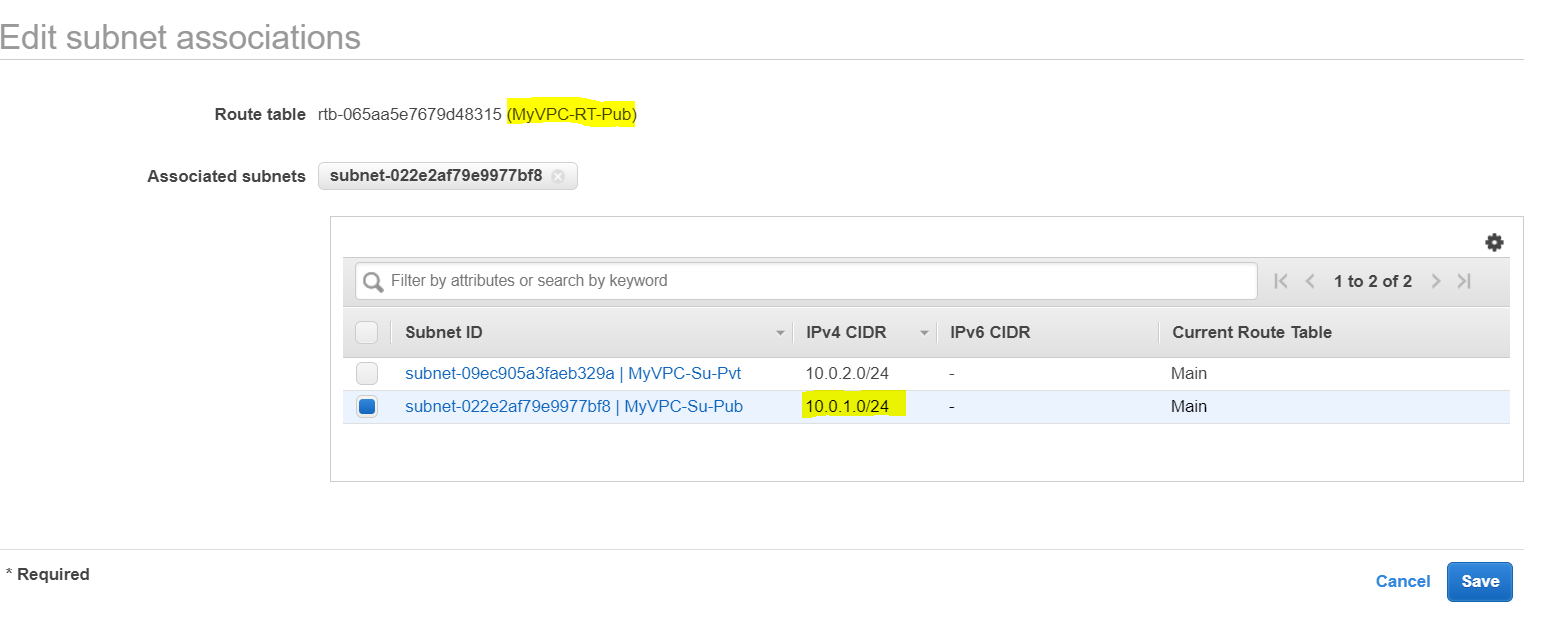


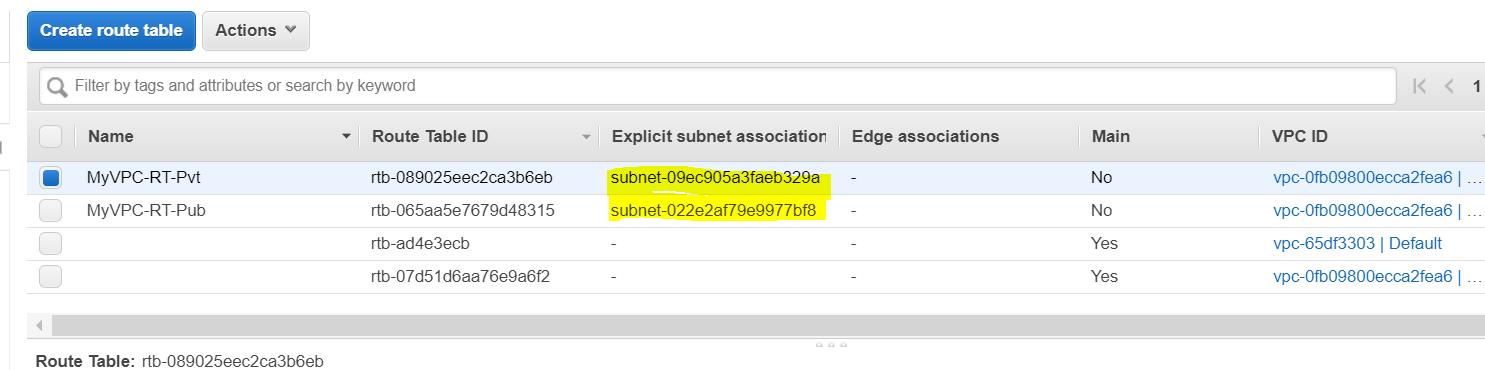
**4. Create Routing table [RT], One as Public & One as Private by associating the appropriate subnets to it.**-Assign Appropriate Subnets  
Pubic RT for Open Internet  
**Private Routing Table**



**Subnet Associations**Select Public Routing Table – Assign the Public Subnet(10.0.1.0/24)  
Select Private Routing Table – Assign the Private Subnet(10.0.2.0/24)



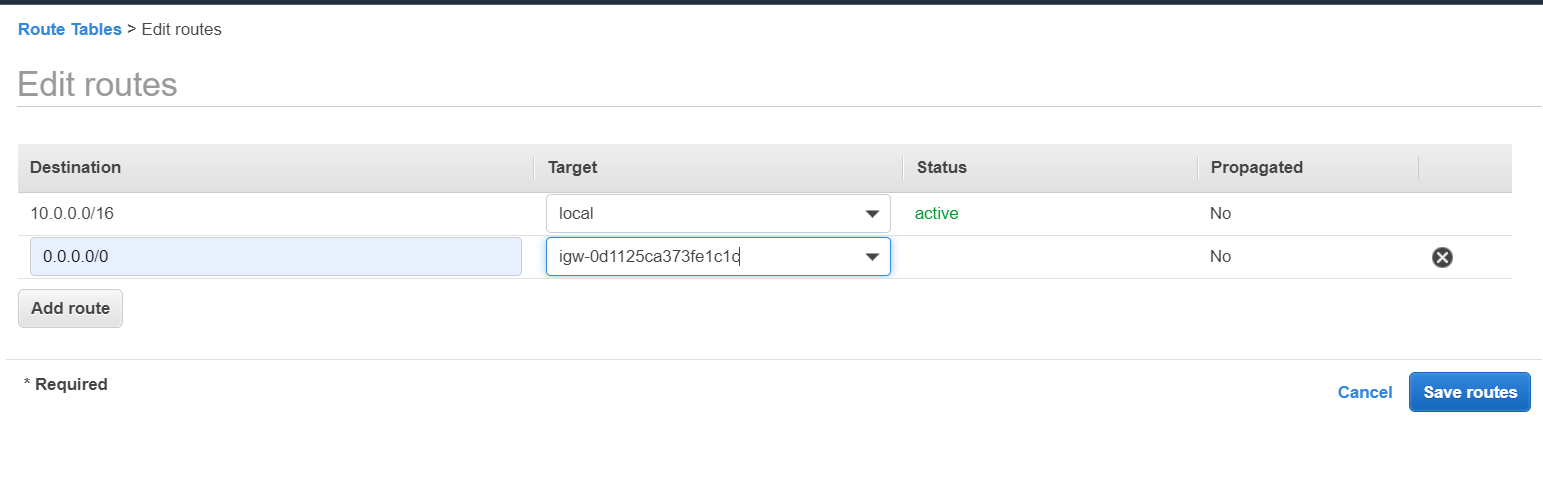




**\*\*\* Yet There is no difference between subnets (Private, Public) & Routing Tables**

**5. Edit the Public route table's Route alone and map the IGW, not the Private and leave it as it is.  
Connect Routing Table With Internet Gateway**

Connect RT(Public)-wit IGW

*   
  Private RT is not connected with open Internet Connectivity

Difference 1 :Public **Routing Table** alone map with Internet Gateway**(IGW**)  
 Private **Routing Table** leave as it is**.(**not connected with IGW**)**

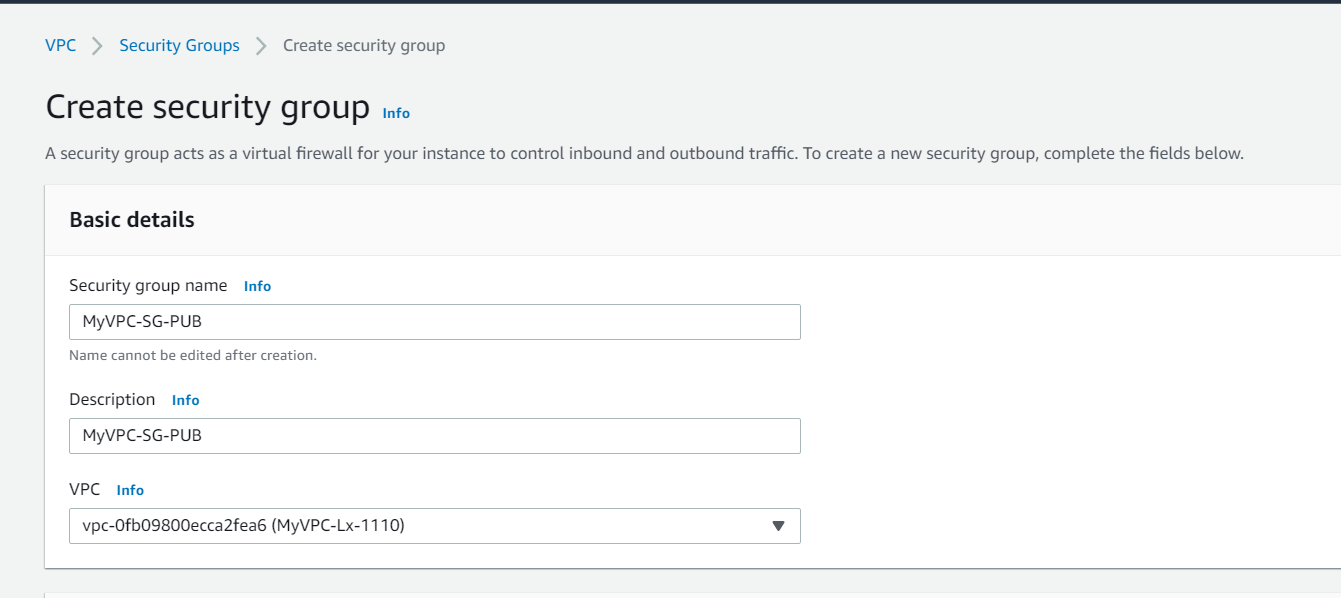
**6. Create Two Security Groups -**   
One for Public [Edit the Inbound rules with **RDP(3389), HTTP(80),HTTPS(443), SSH(22)** and map 0.0.0.0/0 in the source] &   
One for Private inboud rules map with the Security Group of Public in the source  
 [Edit the inbound rules and map the Security Group of Public in the source].

Now – The Private subnet will not access publically, it will access only through Public only  
 Copy group id of public RT’s group id  
 Inbound Rules -> ALL TCP 🡪<source> Paste Group id of publc RT

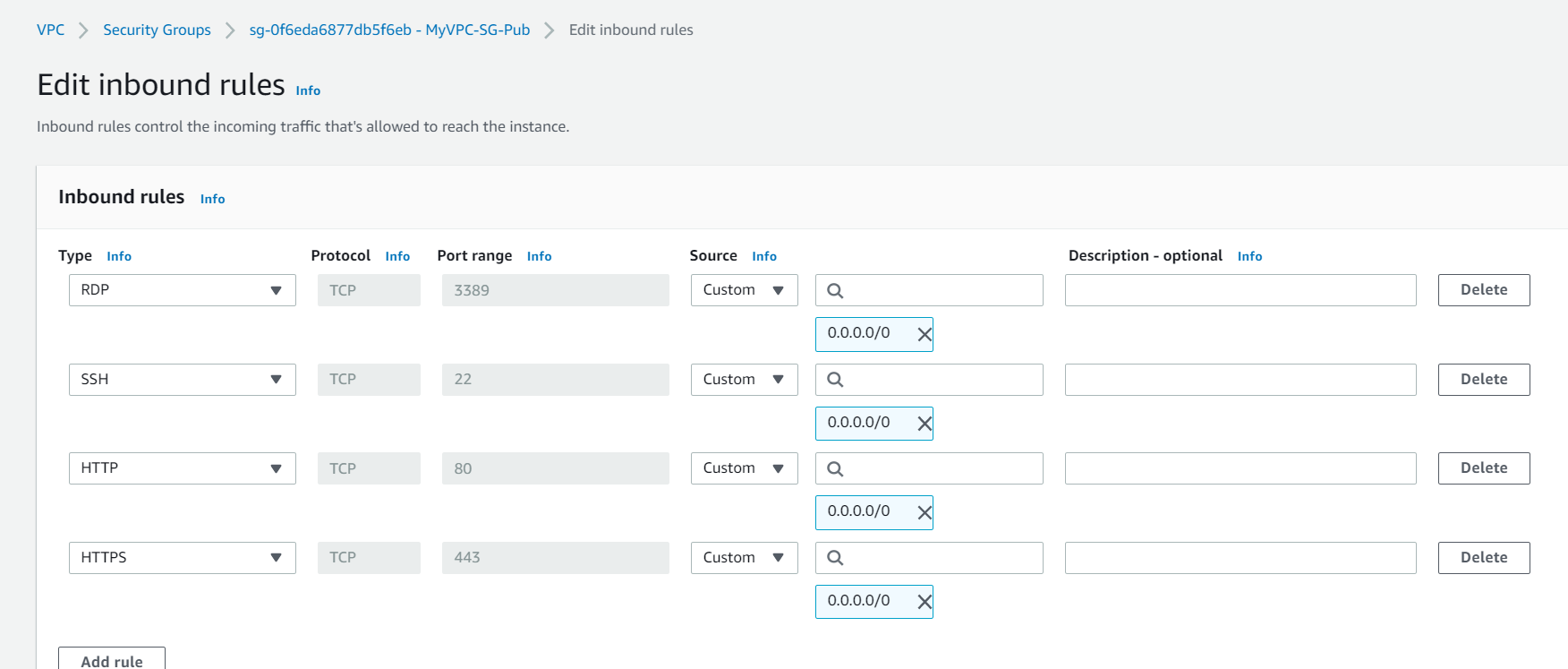
Difference 2- Public subnet canbe access by anyone. But

The private will be access who come through the public gateway / subnet

Security group

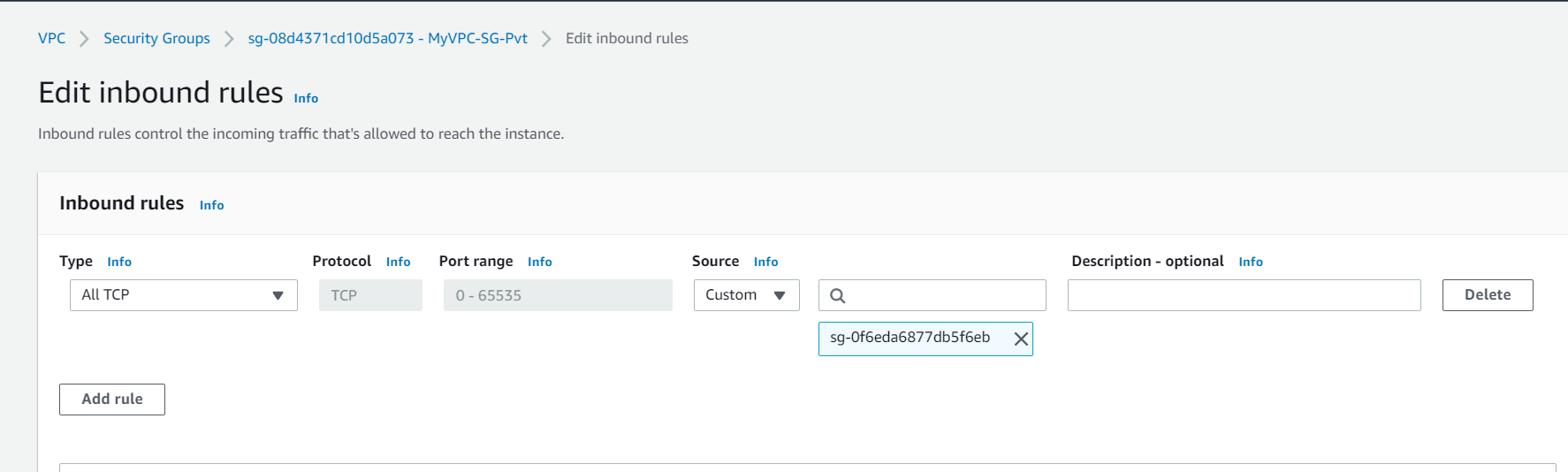


FOR PUBLIC SECURITY GROUP



FOR PRIVATE SECURITY GROUP

Copy **Security Group of Public** and apply into the inbound rules and map the Security Group of Public in the source

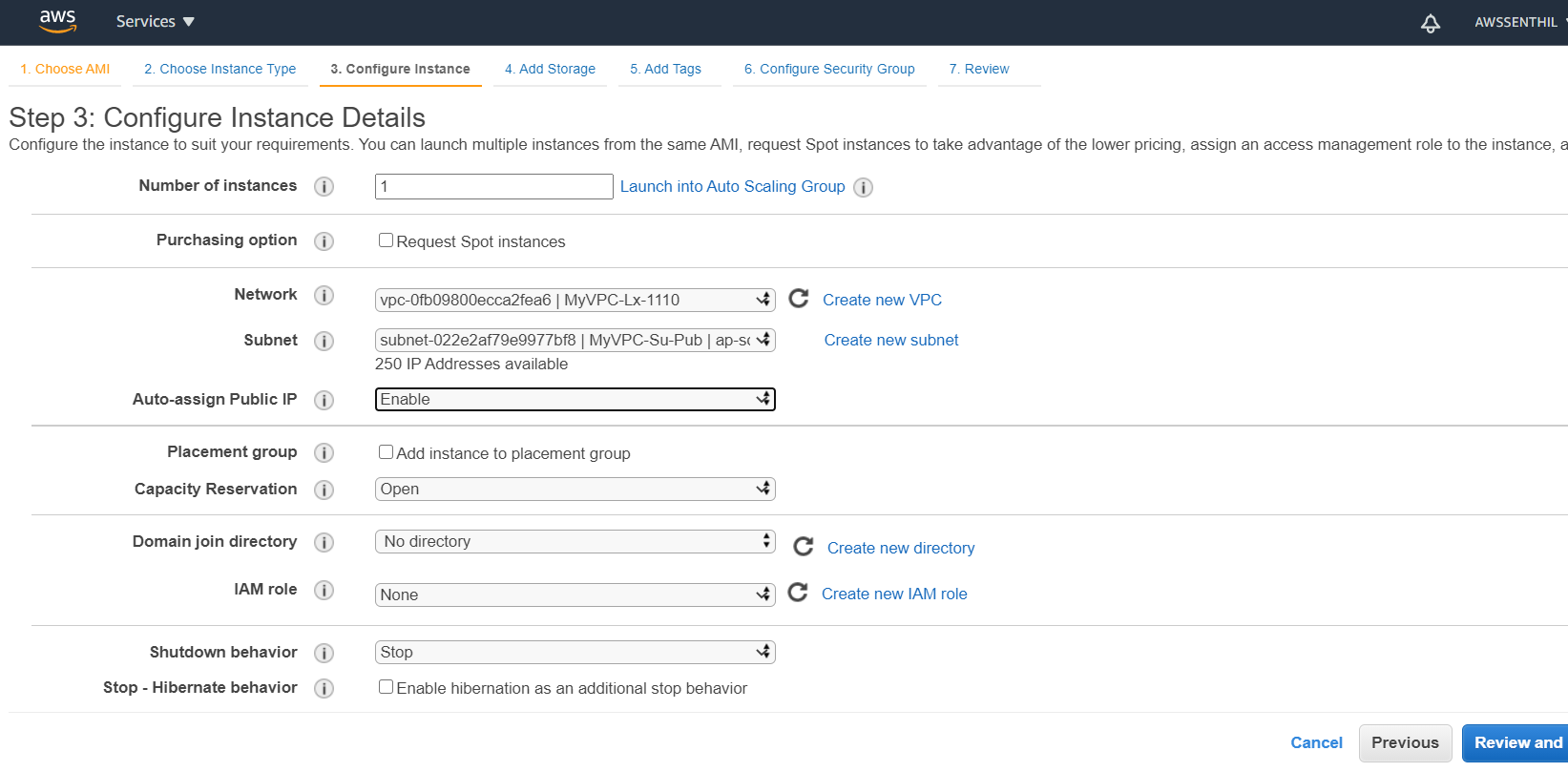


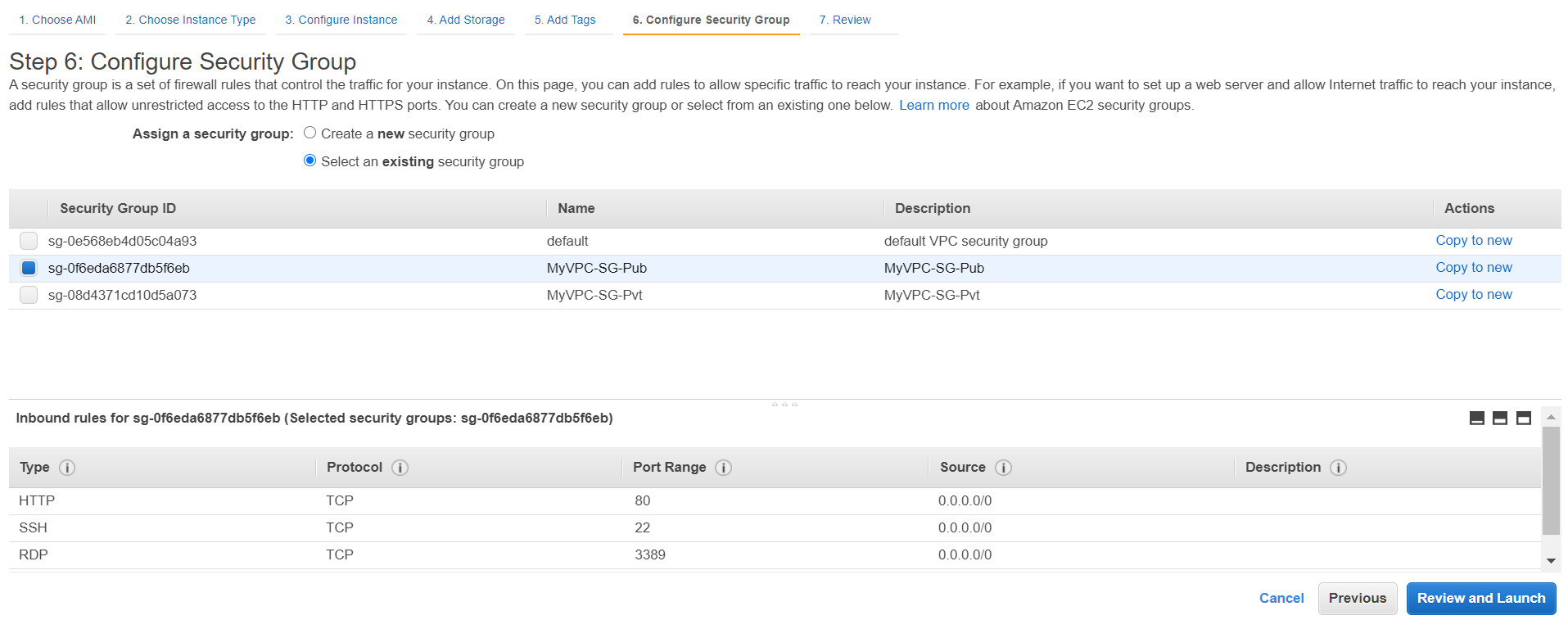
**7. Create Two EC2s instances one in public and one in private subnets with proper Security Groups.**

**7.1 Create EC2 – Linux AMI for Public Instance  
EC2s instances – Auto Assign public – Enable  
EC2s instances – Auto Assign public –Disable**

Difference 3- Select Own VPC

select Private subnet/ **EC2s instances  
 – Auto Assign private-Disable  
- Auto Assign public – Enable**



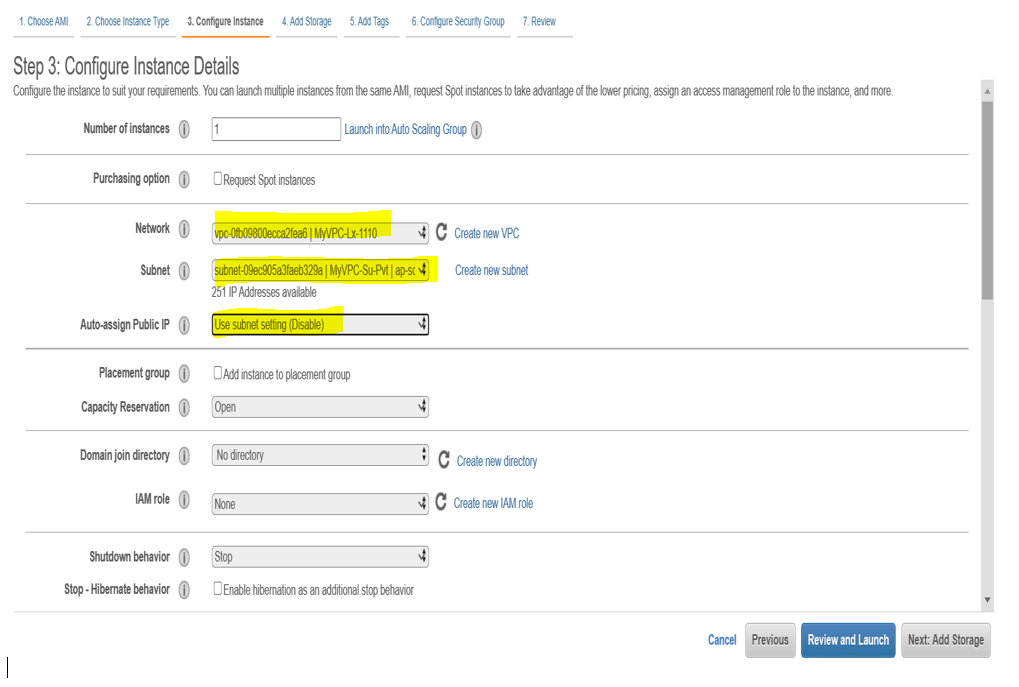


**Create EC2 – Linux AMI for Public & Pivate Instances  
Public Instance:**

**Select Network as Own VPC  
Subnet :Select Public Subnet (created)ex:MyVPC-Su-Pub  
Auto Assign Public IP: Enabled**

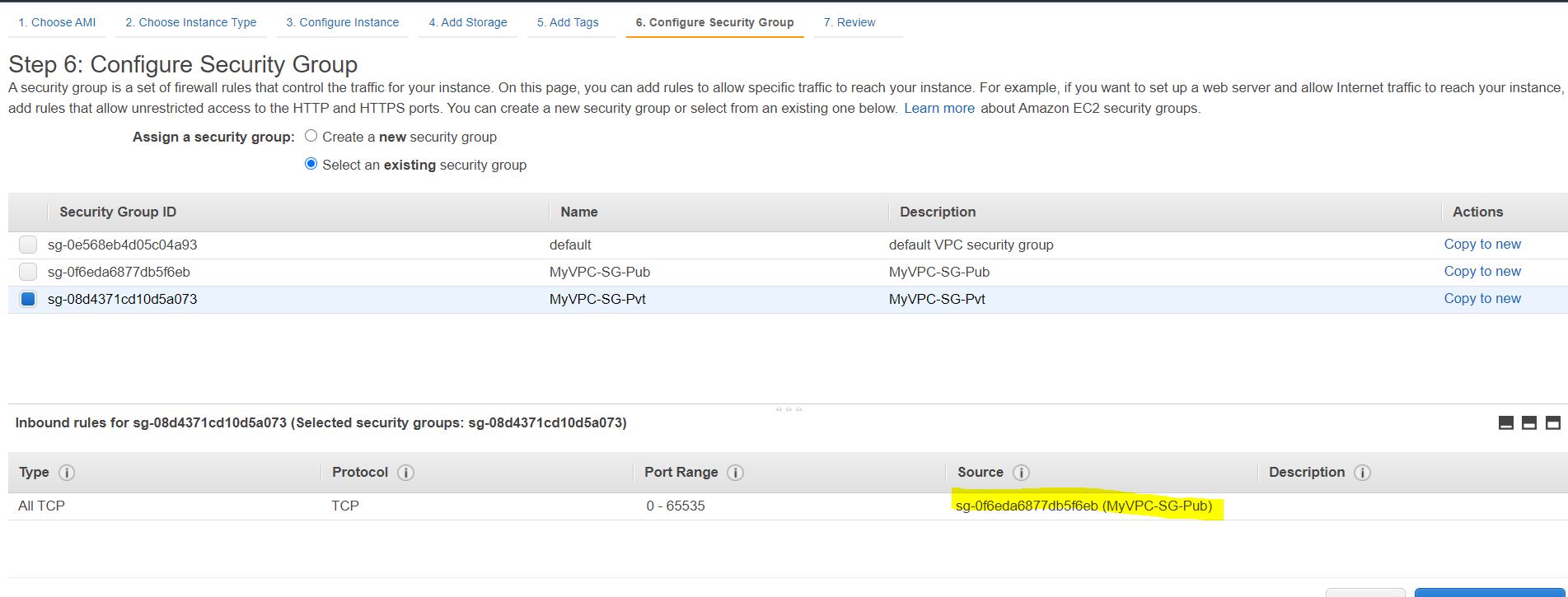
**Pivate Instance:**

**Select Network as Own VPC  
Subnet :Select Private Subnet (created)ex:MyVPC-Su-Pvt  
Auto Assign Public IP: Default / DisabledSE**

  
Select Security Group as Existing.

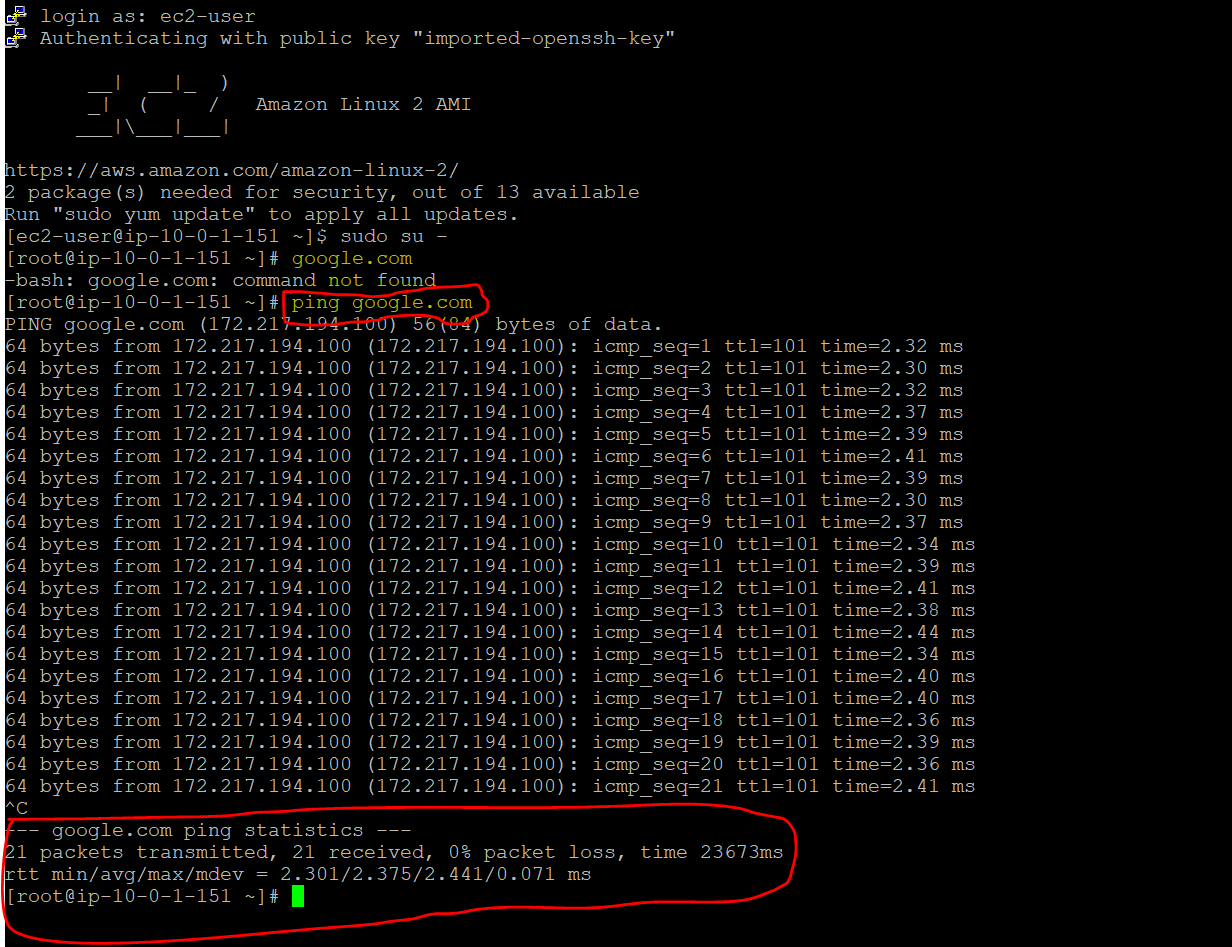
For Public EC2 Instance : Select Public Seciruty Group (**MyVPC-SG-Pub)**

For Private EC2 Instance : Select Private Seciruty Group (**MyVPC-SG-Pvt)**

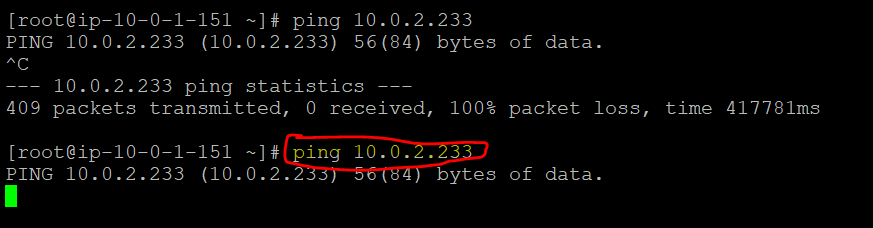


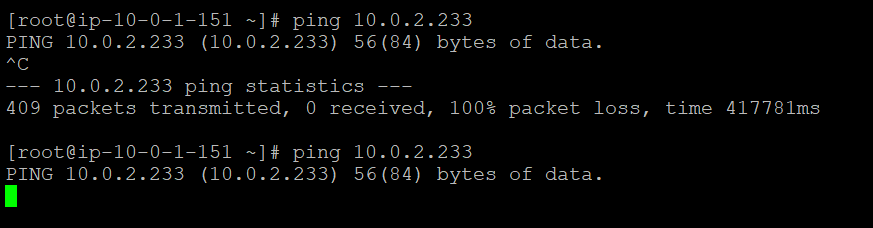
**8. Login into Public EC2 Instance and check the internet connection.   
Connect with Public Ec2 / Public Subnet.**

**Ping with Internet Connectivity**



**Ping with Private Machine Connectivity**



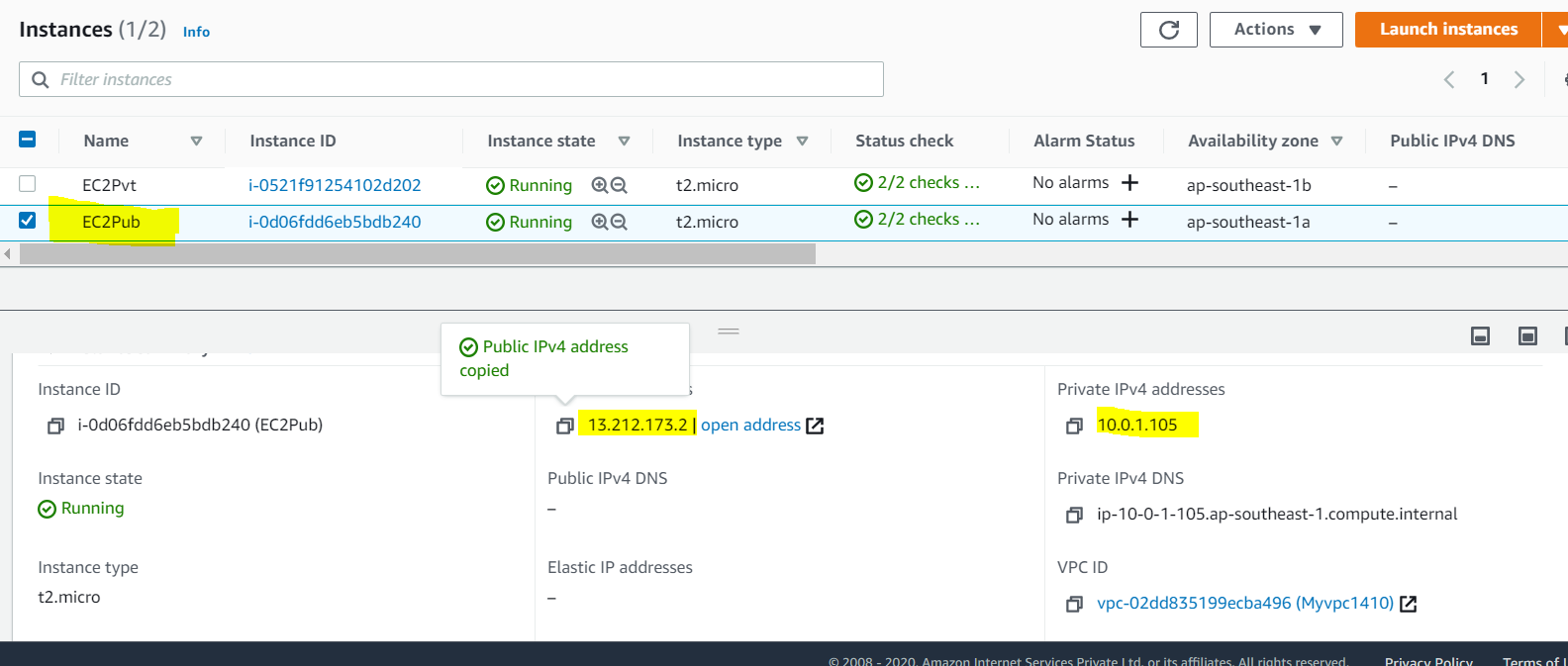


------------------------------------------------------------

**Date : 14.10.2020 ….. continue**

Public EC2 Instance…..

Public subnet



[ec2-user@**ip-10-0-1-105** ~]$ ls -la

total 20

drwx------ 3 ec2-user ec2-user 117 Oct 14 14:48 .

drwxr-xr-x 3 root root 22 Oct 14 13:28 ..

-rw------- 1 ec2-user ec2-user 1700 Oct 11 07:55 AWSVPC1110.pem

-rw------- 1 ec2-user ec2-user 1406 Oct 14 16:49 .bash\_history

-rw-r--r-- 1 ec2-user ec2-user 18 Jan 16 2020 .bash\_logout

-rw-r--r-- 1 ec2-user ec2-user 193 Jan 16 2020 .bash\_profile

-rw-r--r-- 1 ec2-user ec2-user 231 Jan 16 2020 .bashrc

drwx------ 2 ec2-user ec2-user 48 Oct 14 14:52 .ssh

[ec2-user@ip-10-0-1-105 ~]$ -**Private IP of EC2Public Instance**

[ec2-user@ip-10-0-1-105 ~]$ dig +short myip.opendns.com @resolver1.opendns.com

13.212.173.2 – **Public IP of EC2Public Instance**

**Ping with Internet Connectivity(Public EC2 Instance)**

[ec2-user@ip-10-0-1-105 ~]$ **ping google.com**

PING google.com (172.217.194.102) 56(84) bytes of data.

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=1 ttl=52 time=2.05 ms

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=2 ttl=52 time=2.20 ms

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=3 ttl=52 time=2.18 ms

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=4 ttl=52 time=2.12 ms

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=5 ttl=52 time=2.14 ms

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=6 ttl=52 time=2.19 ms

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=7 ttl=52 time=2.06 ms

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=8 ttl=52 time=2.12 ms

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=9 ttl=52 time=2.15 ms

64 bytes from 172.217.194.102 (172.217.194.102): icmp\_seq=10 ttl=52 time=2.23 ms

^C

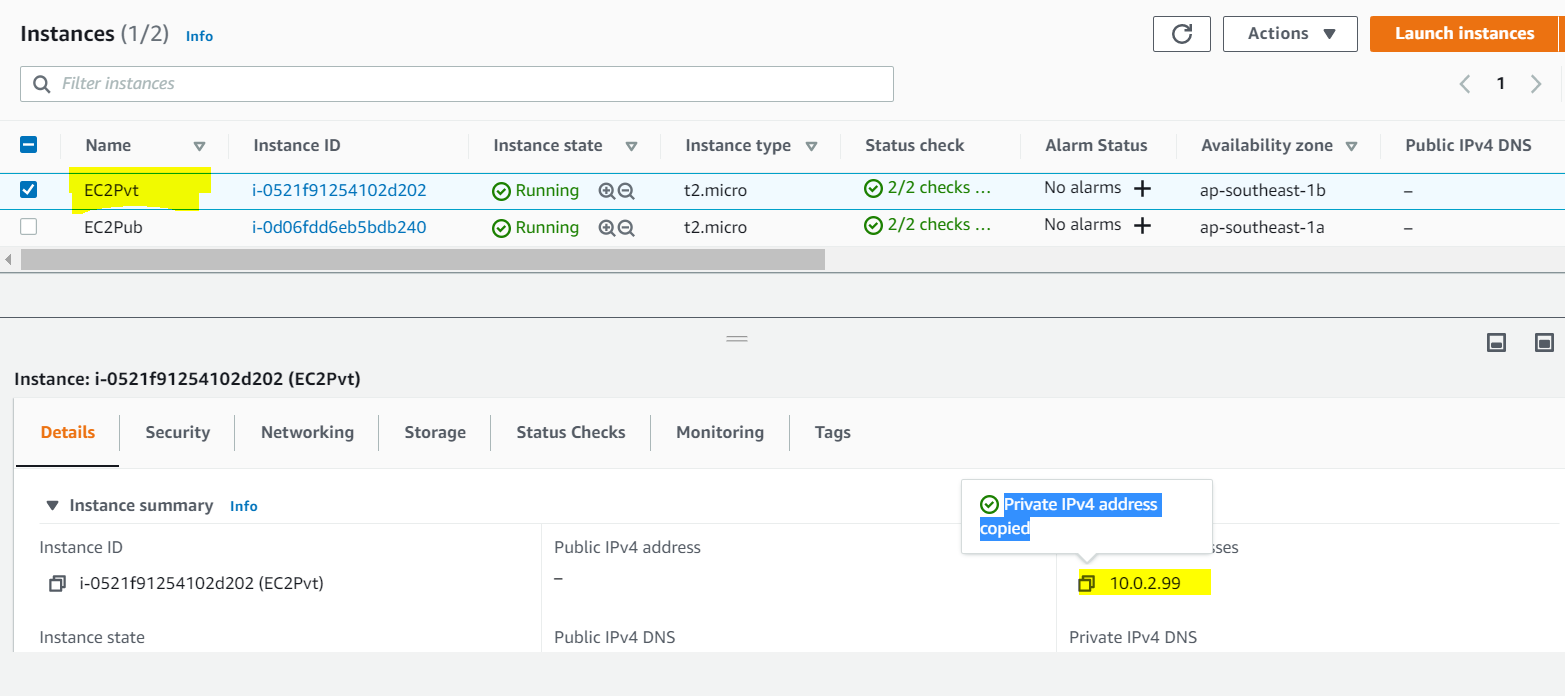
**--- google.com ping statistics ---**

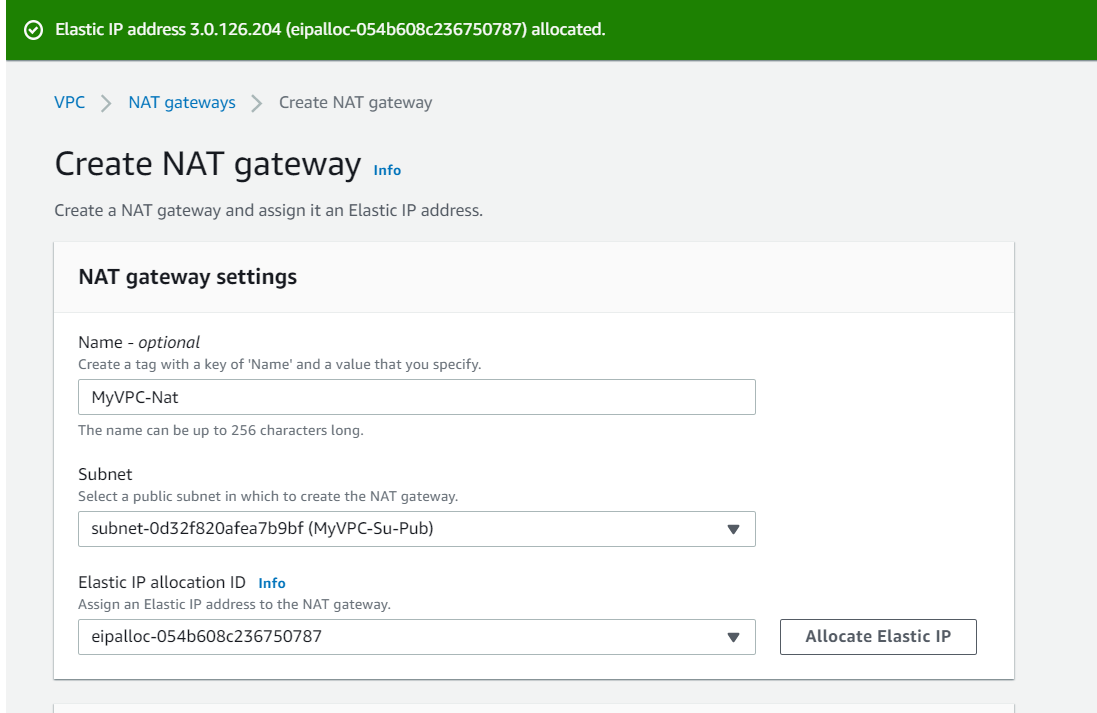
10 packets transmitted, 10 received, 0% packet loss, time 9539ms

rtt min/avg/max/mdev = 2.050/2.148/2.230/0.060 ms

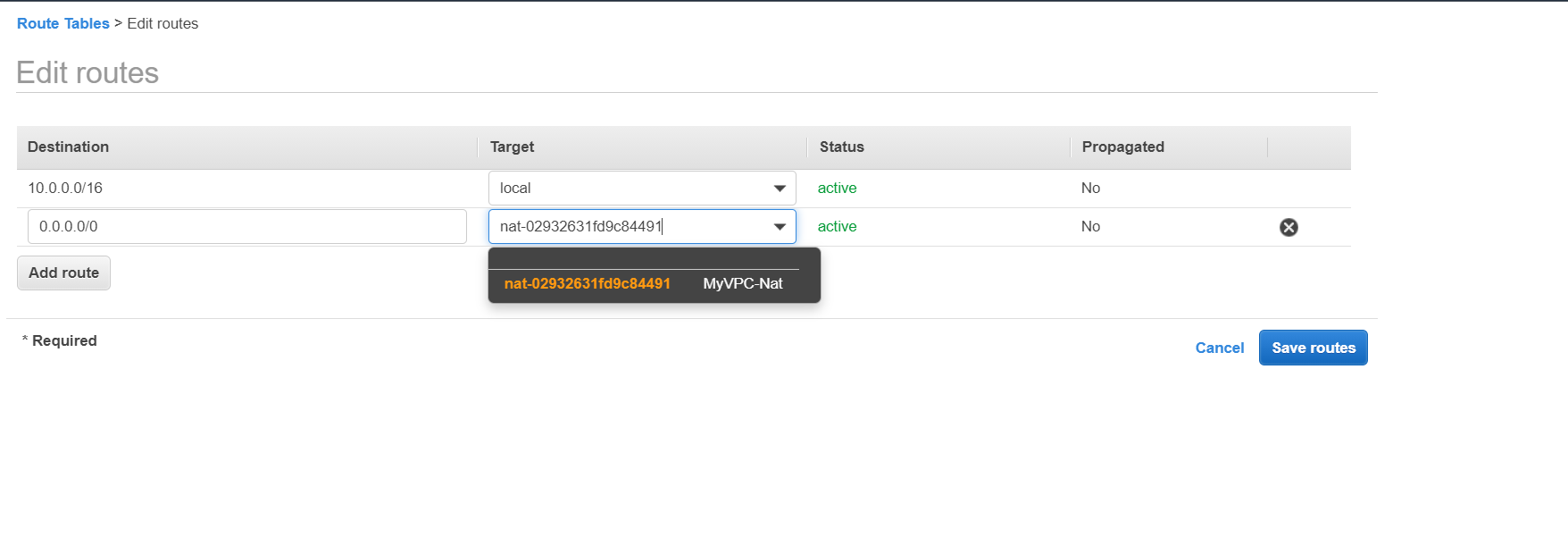
[ec2-user@ip-10-0-1-105 ~]$

**--------------------------Private subnet / EC2 Instance Login from Public**



**9. Create NAT gateway with new Elastic IP for the internet connection in the Private Subnet. Map it to Private RT  
Assign NAT Gateway Subnet as Public Subnet**  


Create NAT gateway with new Elastic IP for the internet connection in the Private Subnet.  
 **Map it to Private RT.**

**10. Now login into the Private EC2 and verify the connectivity and Internet facility**

**Login into Private EC2 Instance from Public Instance**

[ec2-user@ip-10-0-1-105 ~]$ (-**Private IP of EC2Public Instance)**

**[ec2-user@ip-10-0-1-105** ~]$ ssh -i AWSVPC1110.pem ec2-user@10.0.2.99(Private IPv4 address )

(**-i identity\_file** A file from which the [**identity key**](https://www.ssh.com/ssh/identity-key) (private key) for [**public key authentication**](https://www.ssh.com/ssh/public-key-authentication) is read.)

**Last login: Wed Oct 14 15:42:00 2020 from 10.0.1.105**

\_\_| \_\_|\_ )

\_| ( / Amazon Linux 2 AMI

\_\_\_|\\_\_\_|\_\_\_|

https://aws.amazon.com/amazon-linux-2/

**[ec2-user@ip-10-0-2-99 ~]$** whoami

**[ec2-user@ip-10-0-2-99 ~]$** whoami

ec2-user

[ec2-user@ip-10-0-2-99 ~]$ who am i

ec2-user pts/0 2020-10-14 17:06 (10.0.1.105)

**[ec2-user@ip-10-0-2-99 ~]$** ls -la

total 16

drwx------ 3 ec2-user ec2-user 95 Oct 14 15:45 .

drwxr-xr-x 3 root root 22 Oct 14 13:29 ..

-rw------- 1 ec2-user ec2-user 77 Oct 14 15:45 .bash history

-rw-r--r-- 1 ec2-user ec2-user 18 Jan 16 2020 .bash logout

-rw-r--r-- 1 ec2-user ec2-user 193 Jan 16 2020 .bash profile

-rw-r--r-- 1 ec2-user ec2-user 231 Jan 16 2020 .basher

drwx------ 2 ec2-user ec2-user 29 Oct 14 13:29 .sash

**[ec2-user@ip-10-0-2-99 ~]$**

**--Communication with Public----**

**PING 13.212.173.2** (13.212.173.2) 56(84) bytes of data.

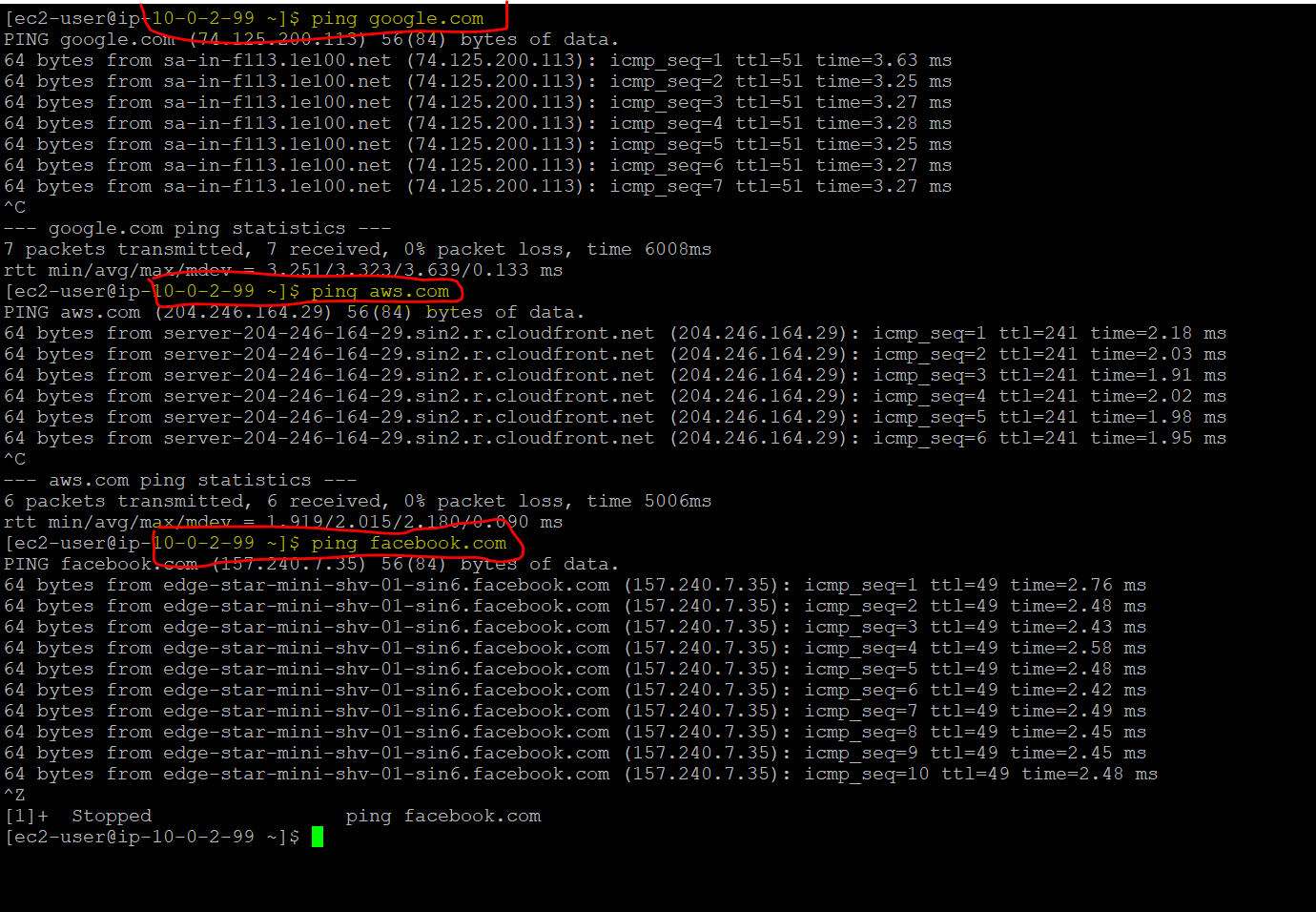
^C

--- 13.212.173.2 ping statistics --- Public IP of EC2Public Instance

9 packets transmitted, 0 received, 100% packet loss, time 8183ms

* The Private Ec2 Instance/ Subnet is **Login within** Public Ec2 Instance/ Subnet
* Now Checking with the **Internet Connectivity** with Private Ec2 Instance/ Subnet

**.**



**Note : Issues Faced**

#1.10.10.20 & 11.10.20 – Upto Private Ec2 Instance Login & internet Connectivity … No issues Faced.  
#2. Facing issues with Login to Private Subnet from Public Subnet. Try to Many way resolve the issues.  
(with reference to many articles)  
#3.14.10.20 – Start again from scratch, same issues.  
#4.The ssh.Pem file is not loaded / available in public instance  
#5.Permissions are denied to copy the .pem to usig ssh.  
#6.Try to use Winscp to file transfer from laptop to Public EC2  
#7.using ssh -i <Filename>.pem ec2-user@YOUR-IP(Private Ec2)  
#8.Permissions are denied .Not Login  
#9.Remove the key file(lock) from SSH folder  
#10.now the command is executed., file login to Private EC2, and Internet Executed.

At last fine.

