AWS VPC

VPC (Virtual Private Cloud):

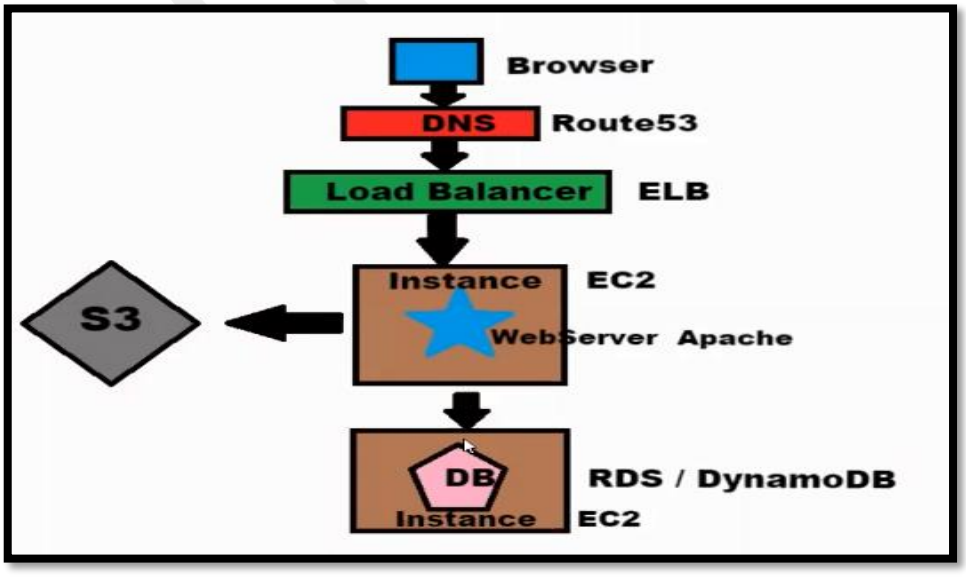
- VPC comes under Network Engineering
- Ip ranges from 0 to 255
- CIDR (Classless Inter-Domain Routing) – Representation of Ip address.
  https://www.ipaddressguide.com/cidr
- Understand the Difference between IPv4 and IPv6 :
  https://byjus.com/free-ias-prep/difference-between-ipv4-and-ipv6/?utm_source=Google&utm_medium=CPC&utm_campaign=IAS_Dynamic_Traffic_Chennai_April30&utm_term=&gclid=CjwKCAjw9r-DBhBxEiwA9qYUpTsn9BITHLlN5qXby5QYV-rLezM5a78stKWNrcdaTulb3Ydq-0Fs6xoCjakQAvD_BwE

Refer below to understand IP, CIDR, IPv4 and Need of Private IP
VPC - Overview.txt

Key Concepts:

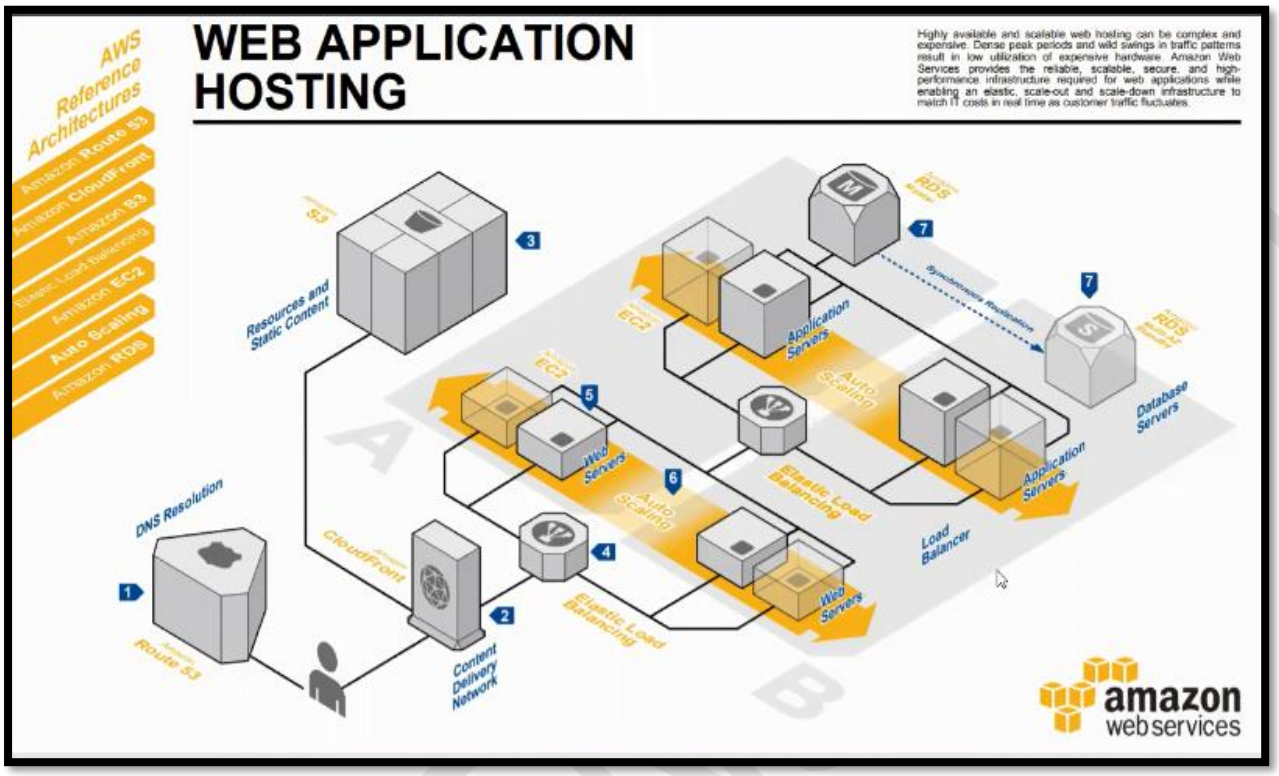
- **VPC** - a Virtual Network dedicated to aws user on internet
- **IGW** - Enables communication between instances on your VPC over net
- **SUBNET:**
  - A logical subdivision of a Network
  - One or more sub network can creates based on business requirement
  - IP representation and range can be set using CIDR
- **CIDR:** Classless Inter-Domain Routing helps to set the range and ip addresses representation for the subnets.
- **IP:** Essential to access the Internet. Which has set of rules
- **Public IP**: assigning from Amazon pool and able to access our instance only through this dynamic and static
- **Private IP:** Each instance will have a default network interface eth0...from here the private ip generate for each instances static
- **Route tables:** This directs subnets and control traffics through routes (set of rules)
- **NAT gateways:** Enables communication between instances (Public and Private)
- **Elastic ip on Nat gateways:** Static IP

Amazon Management Console (AMC):



- Browser – Way to access the server
- DNS (Route53) – Since It is easy to remember we have to host domain to Public Ip or the server
- ELB – Balances Clients Application Request with the help of Autoscalling and ensure better server performance
- Webserver – Application server Engineering
- Database Engineering (Private and it contains confidential data's)
- Understand the differences between two-tire and three-tier architecture

Example Architecture from AWS



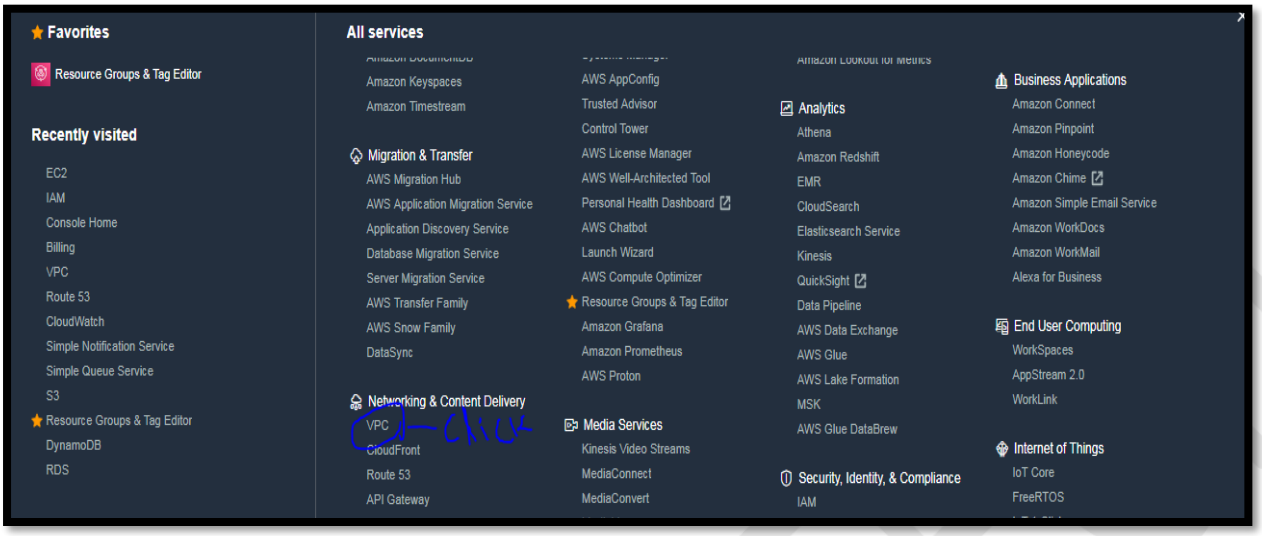
- 2-tier => Client Application + Webserver + DB (Static Application)
- 3-Tier=> Client Application + Webserver+Application Server +DB (dynamic application)

VPC Creation:

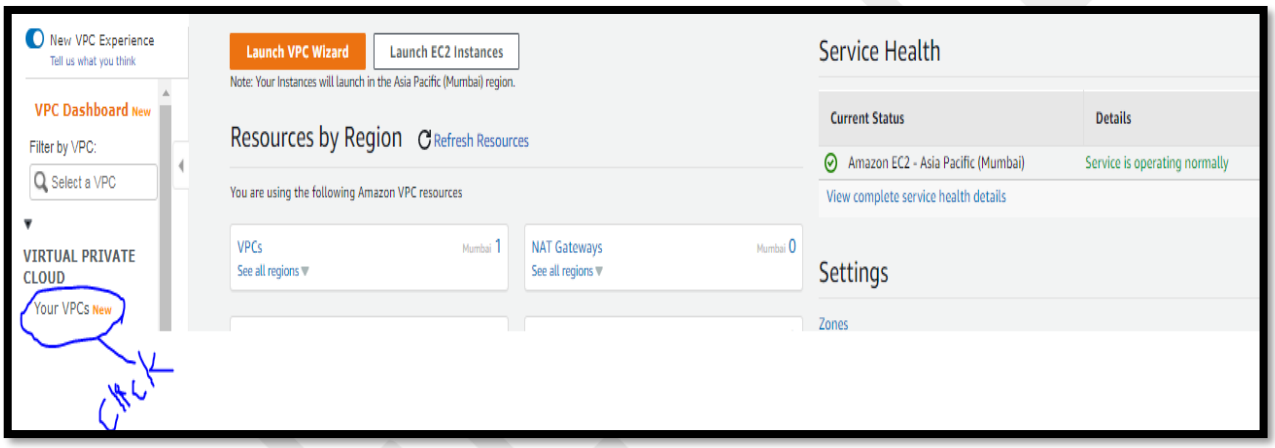
```
Steps to Create and set up a VPC in AWS environment:
============================================

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 SG
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.
```

Step1: Create a VPC

Choose the Service from AWS Console and single click→

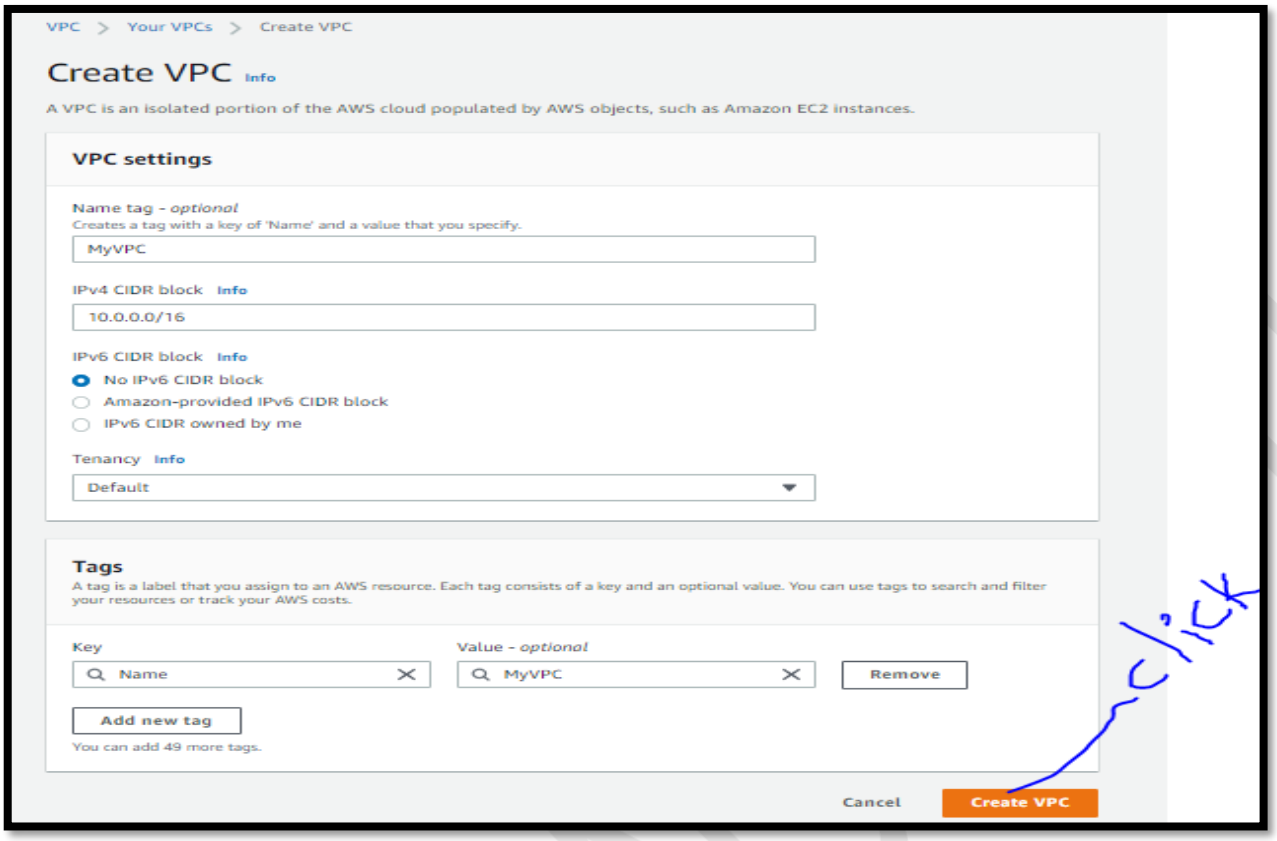


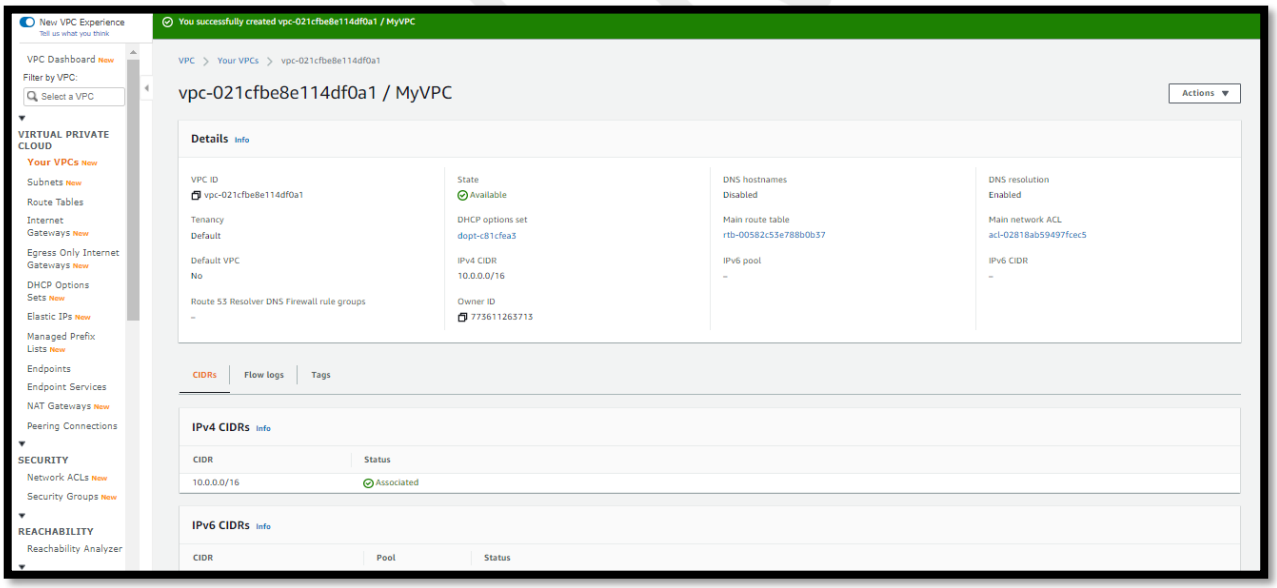
Click on Your VPC



Click on Create VPC

Name the VPC / Set CIDR as 10.0.0.0/16 (65536 Hosts created)

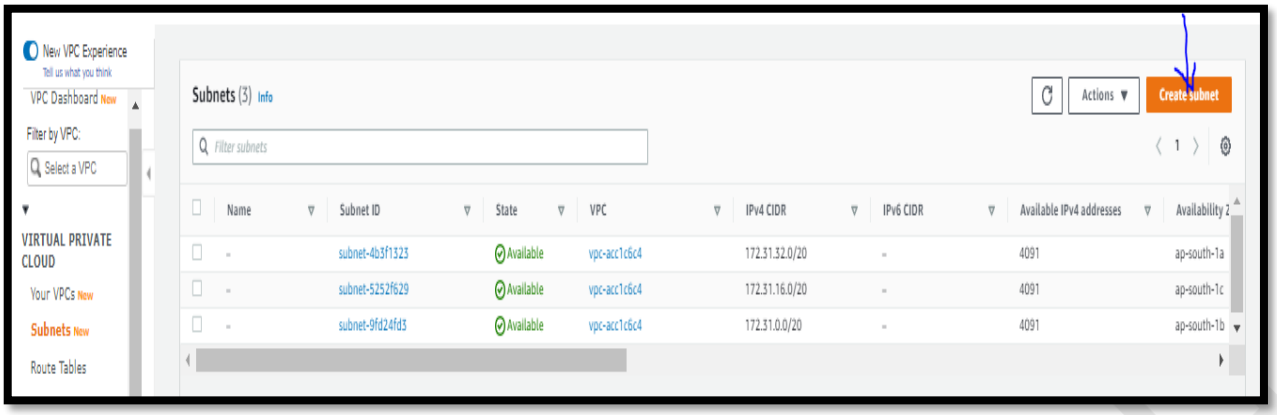


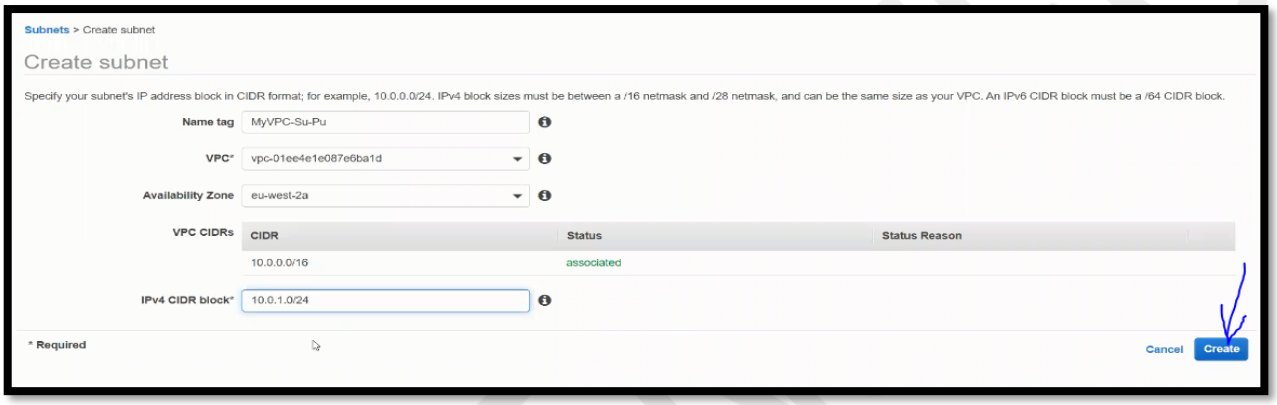


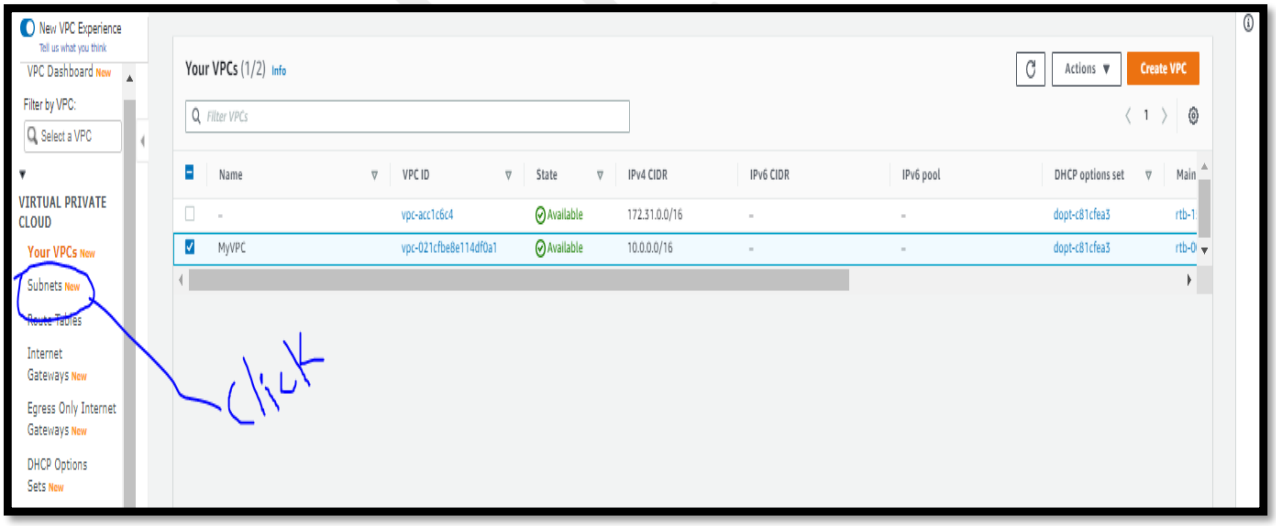
VPC Created

Step2: Create a Public and Private Subnets in different AZ's

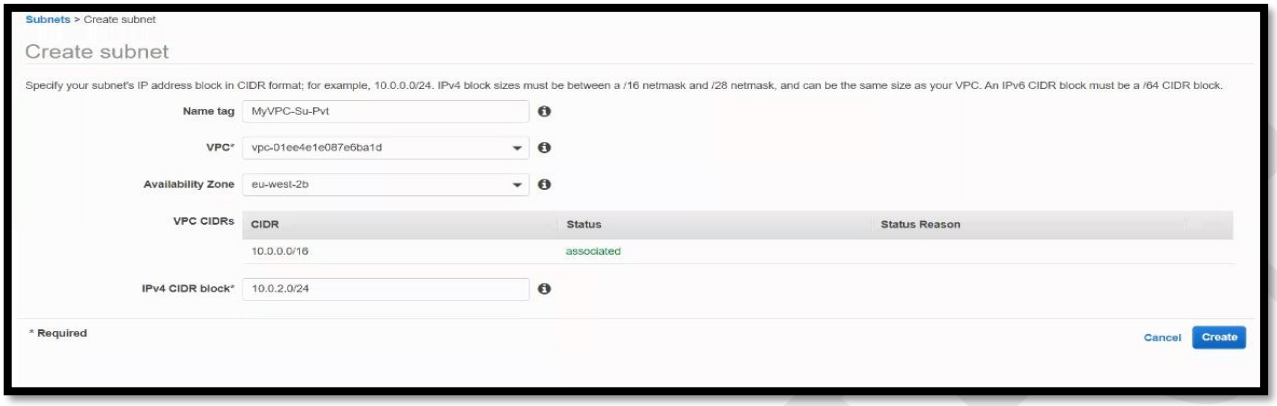
Create a Public Subnet:



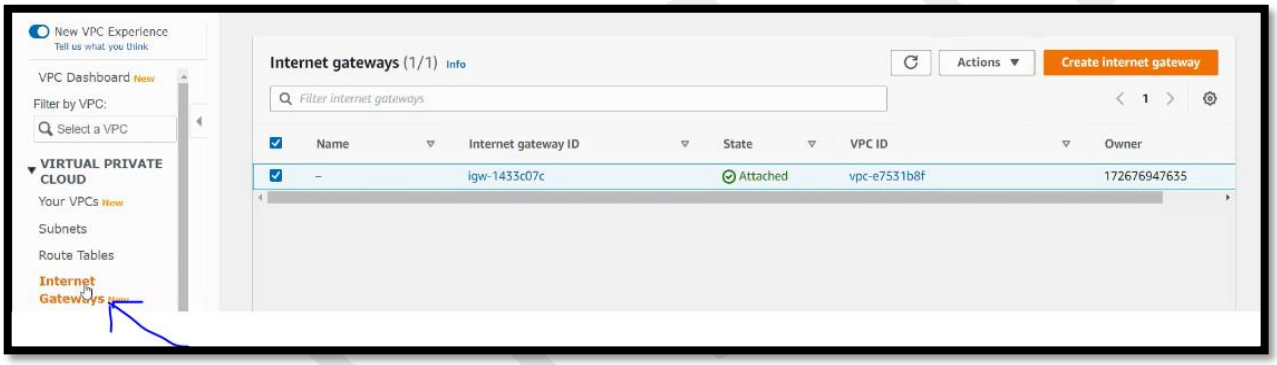


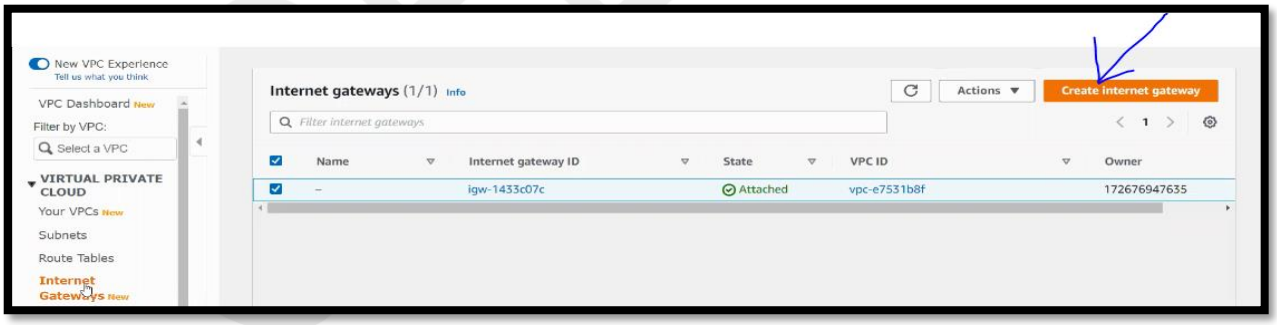


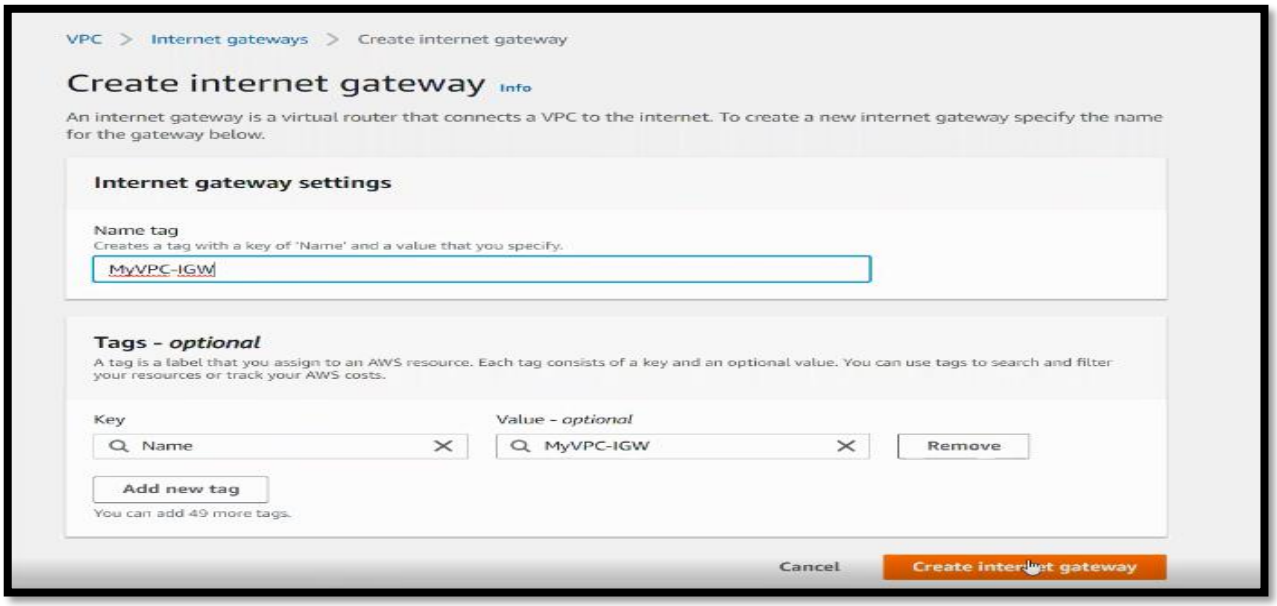
Create a Private Subnet:



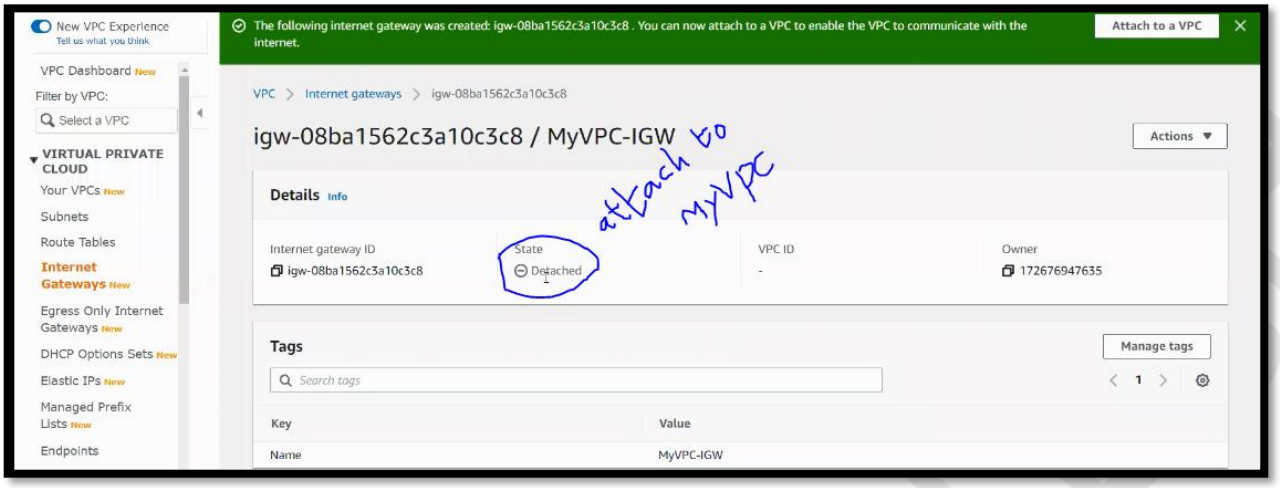
Step3: Create an IGW & attach to VPC (IGW supplies n/w to VPC)





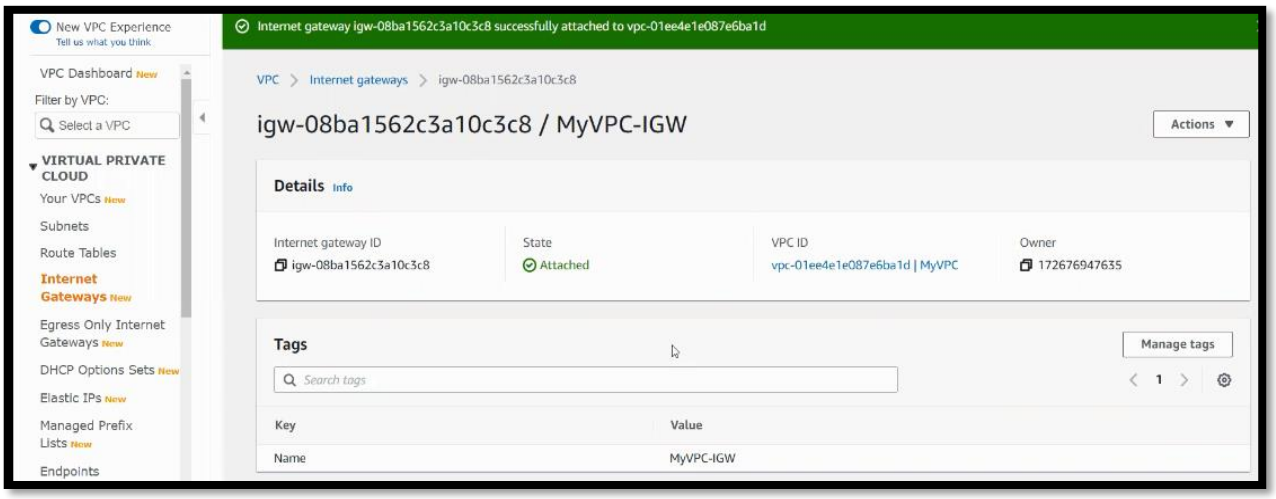


Attach the IGW to MyVPC – Click attach to a VPC

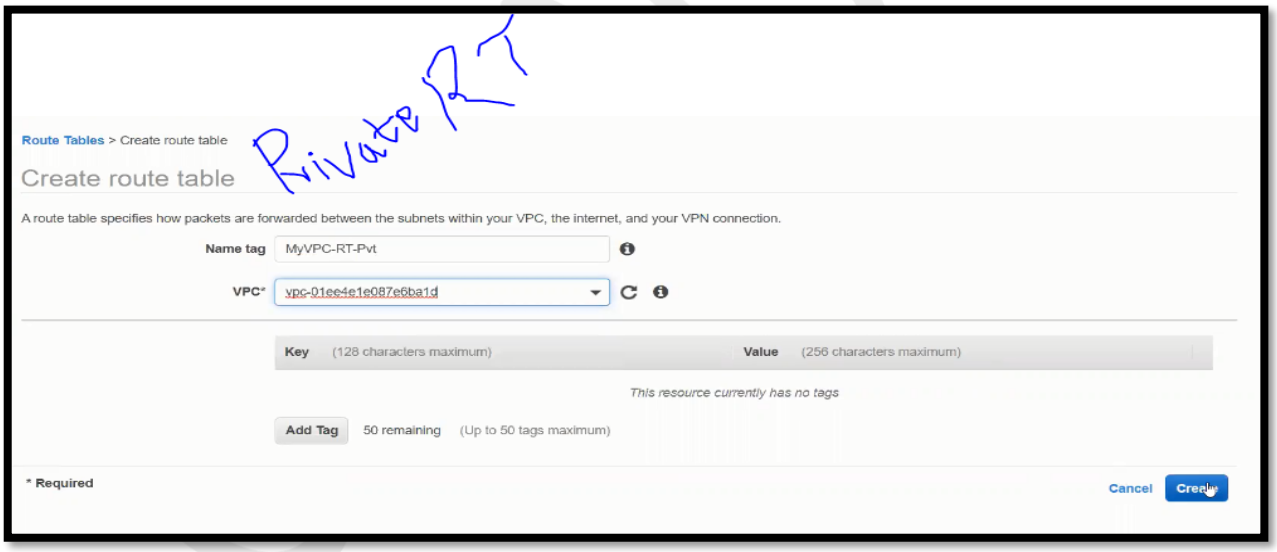


Choose MyVPC and attach

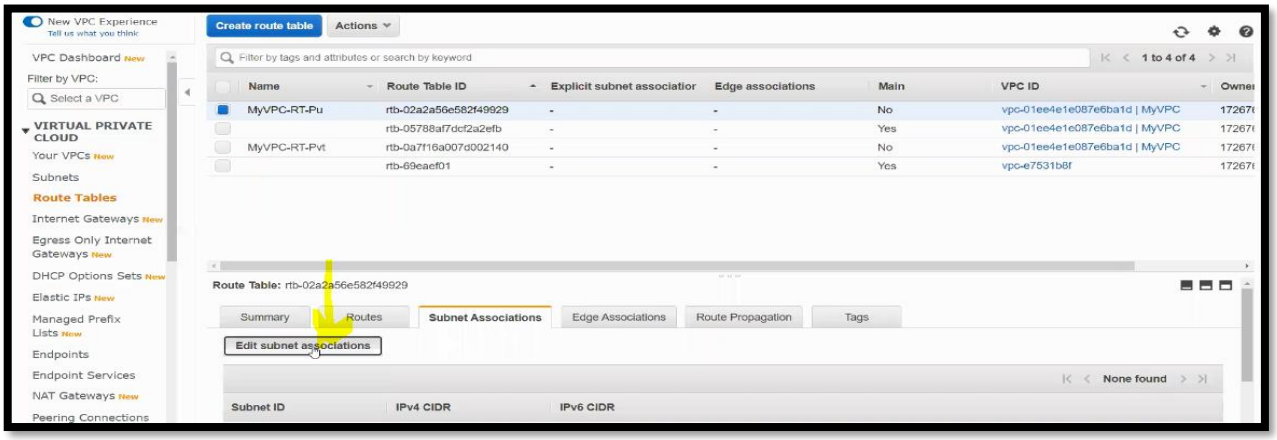
Attachment Completed



Step4: Create a Public & private Route Tables & associate with appropriate Subnets.

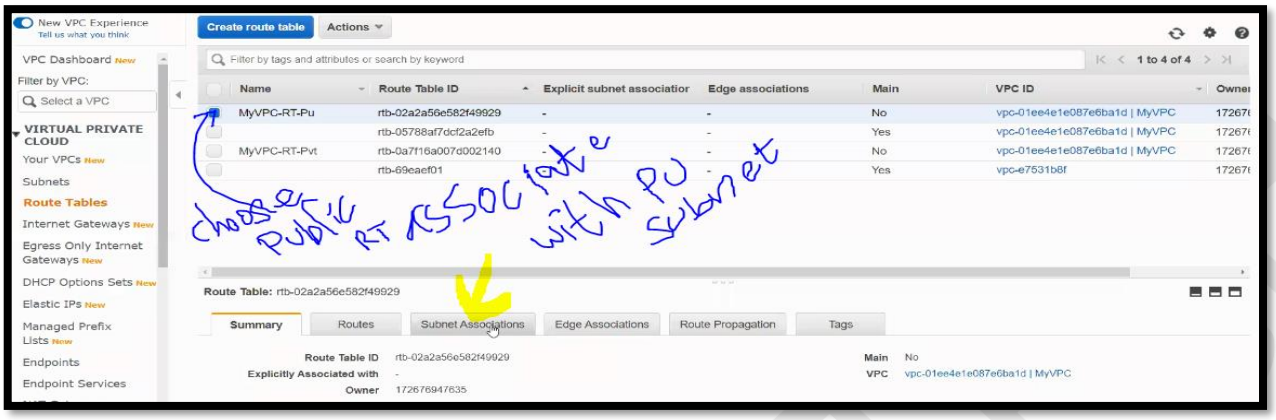


Ensure to Two Route Tables Created

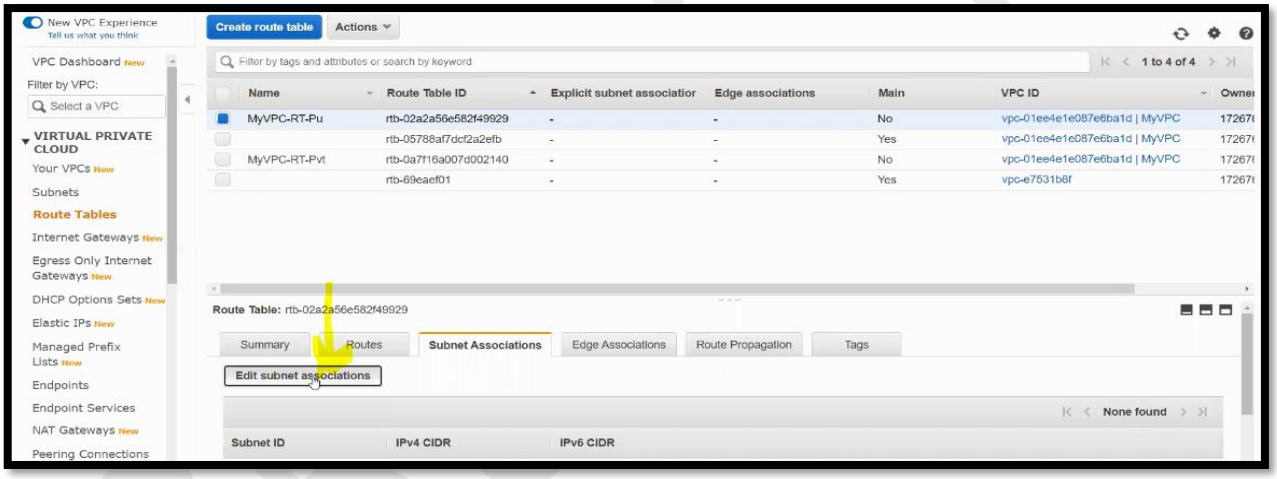


Associate Public RT with Public Subnet

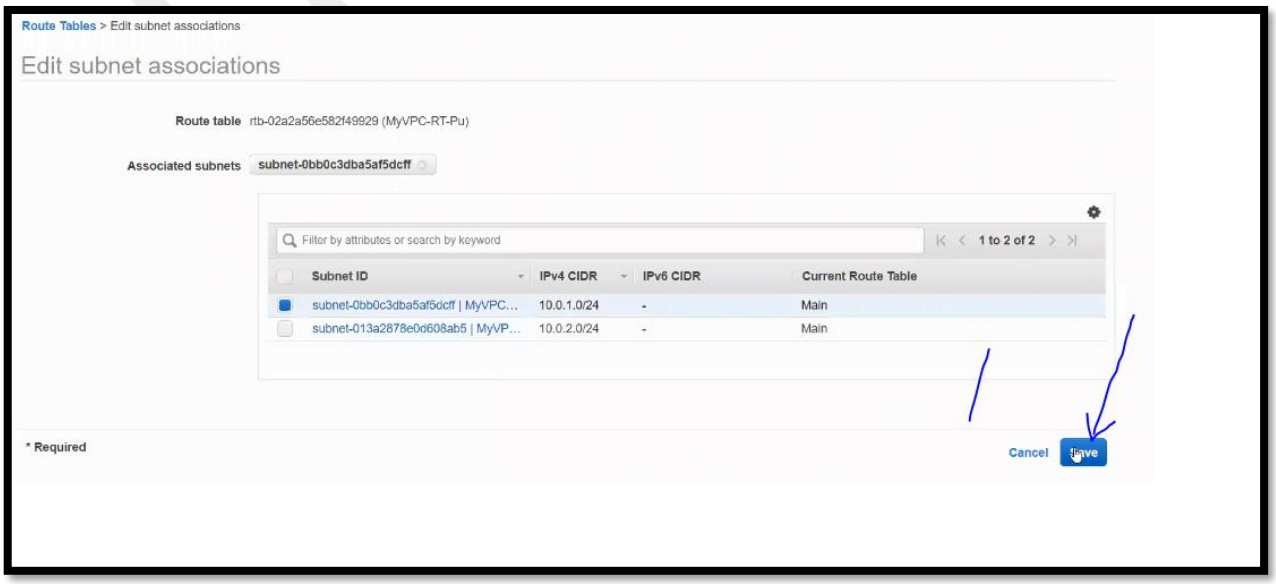
Click Subnet Associations



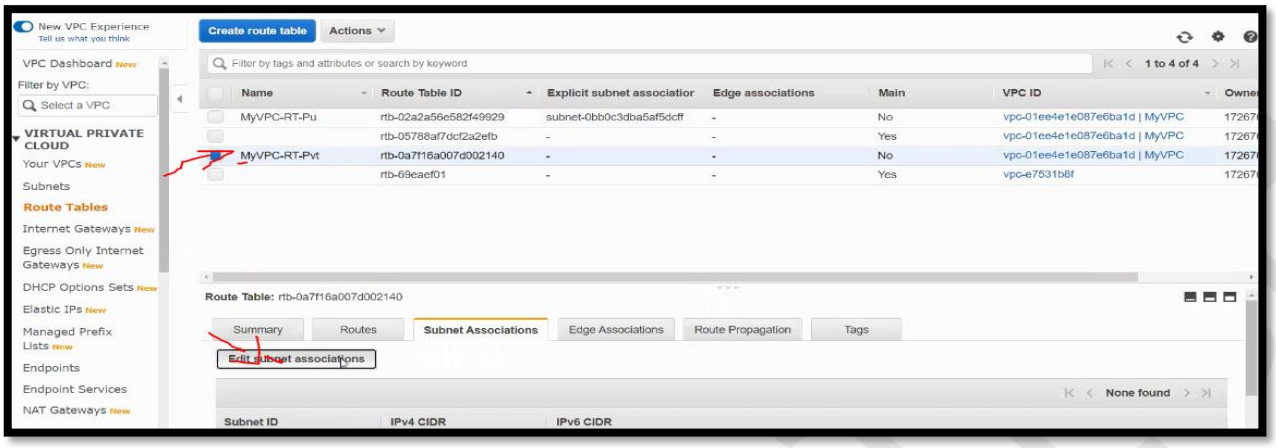
Click Edit Subnet Associations

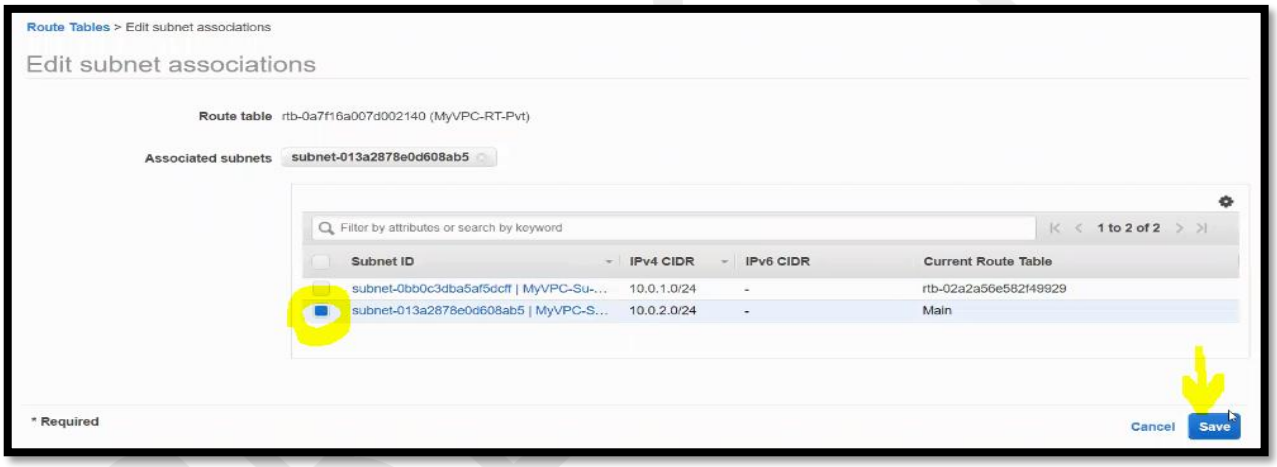


Select Public Subnet and save

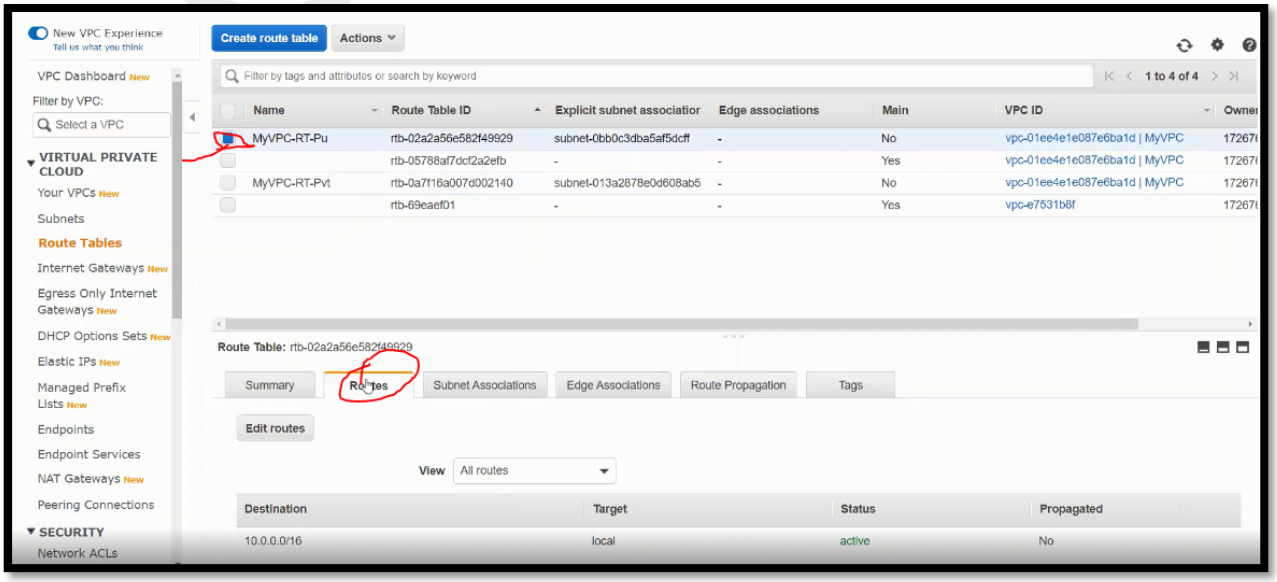


Associate Private RT with Private Subnet
Click Subnet Associations

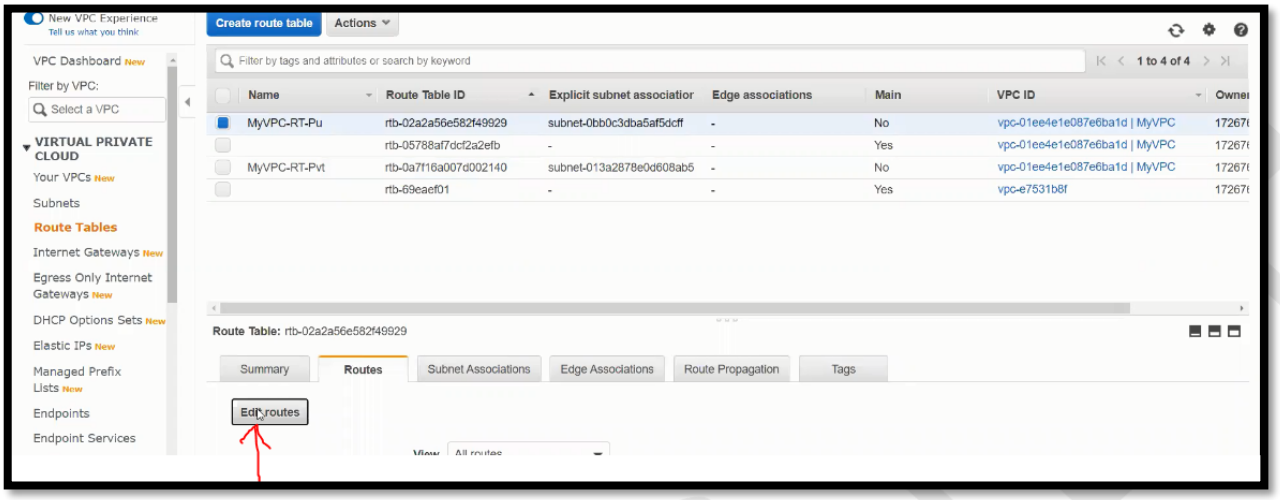




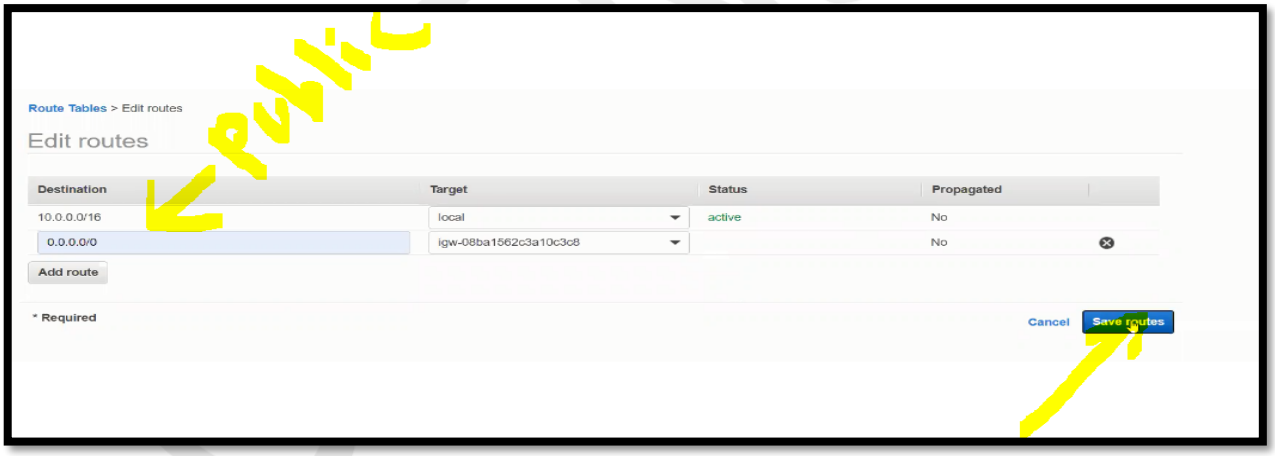
Step5: Set the Routes to the Public Subnet
Select Public Subnet and click on routes



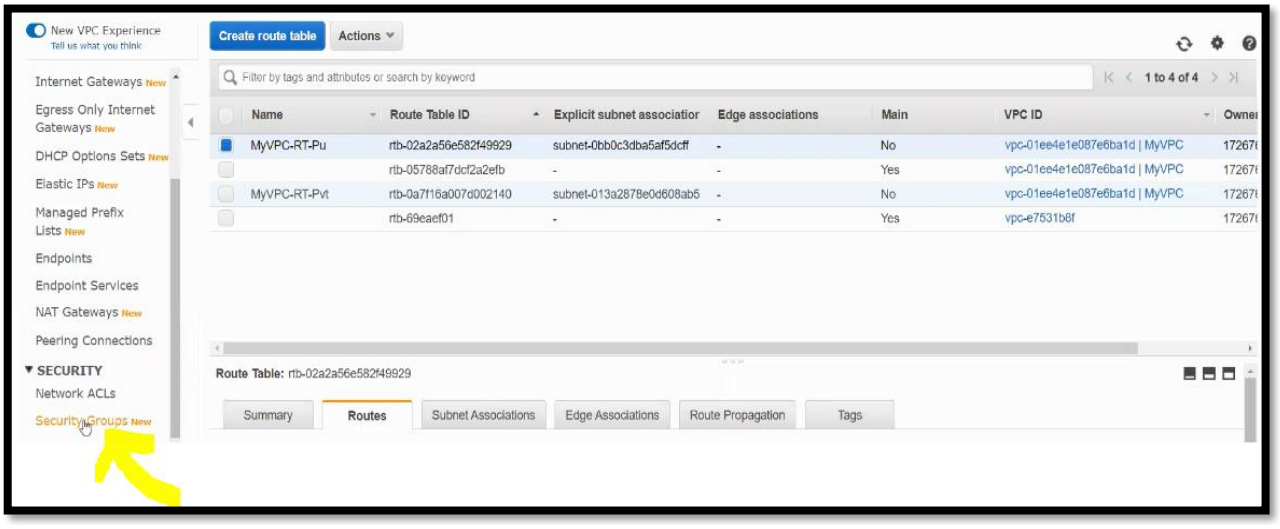
Click Edit Routes



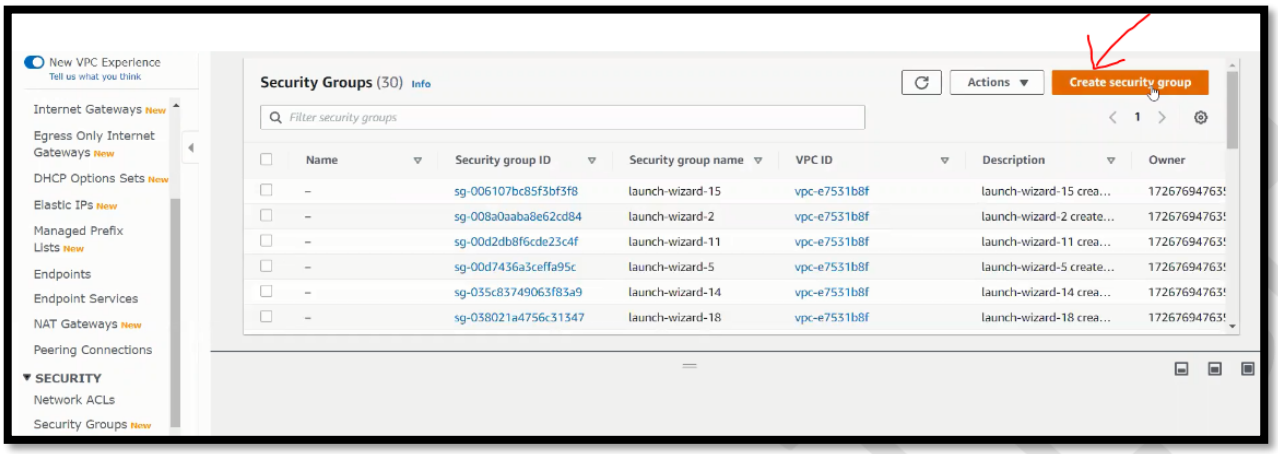
Destination 0.0.0.0/0 indicates can access from anywhere – Public
Target – IGW needs to be selected because we are using our own VPC
Then click Save routes



Step6: Create an Public and Private Security Groups
Single click Security Group module highlighted in Yellow mark



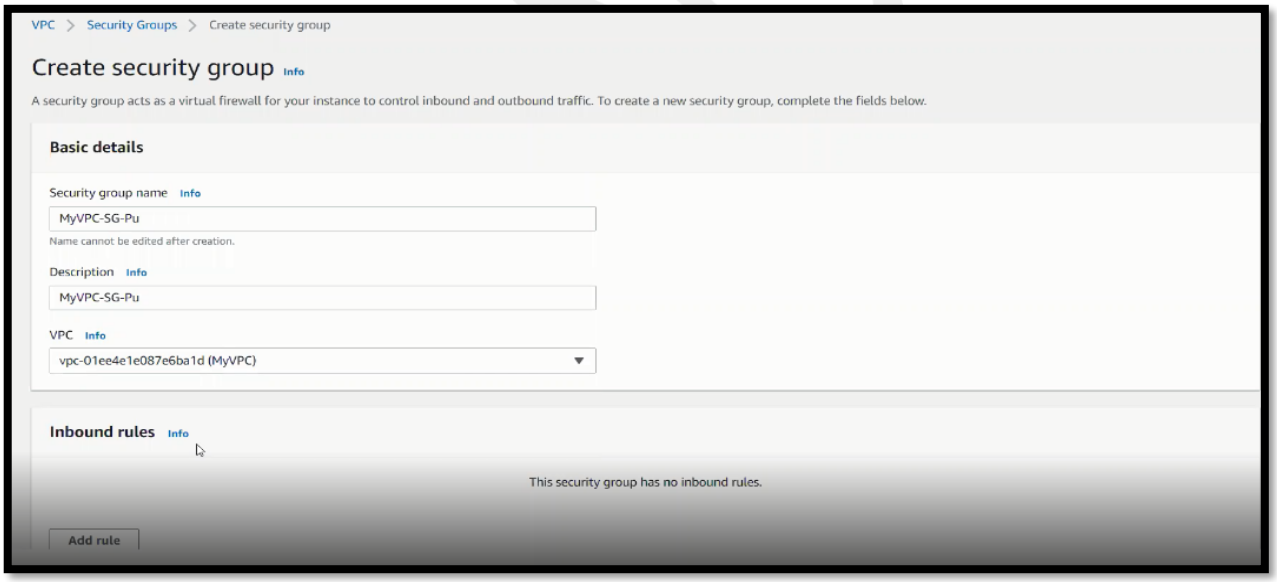
Click Create Security Group

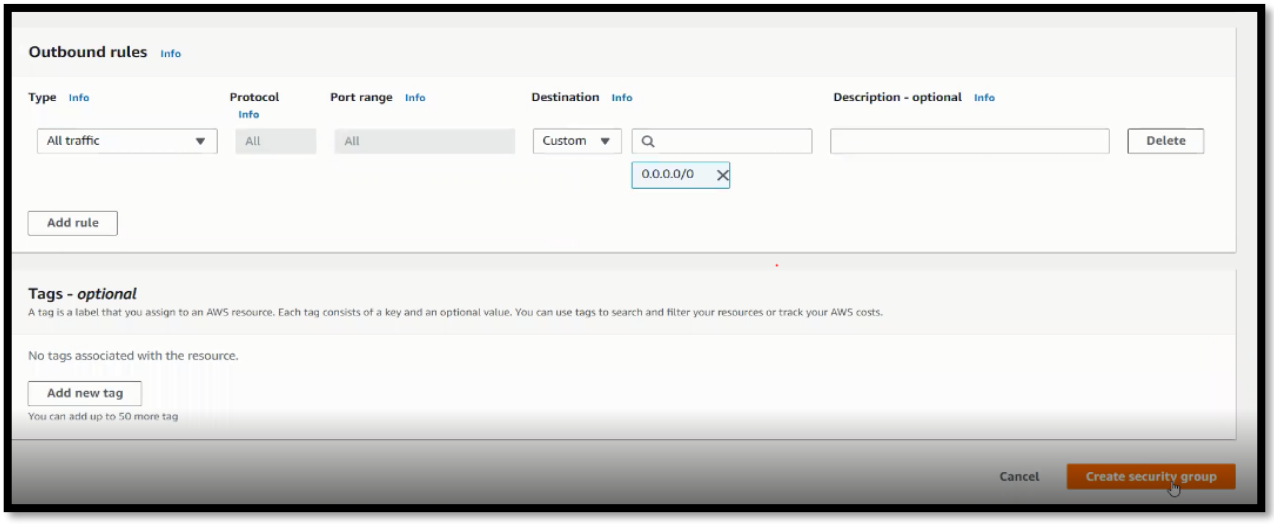


Create Public Security Group:

Tag Understandable Public SG name, keep the same in Description

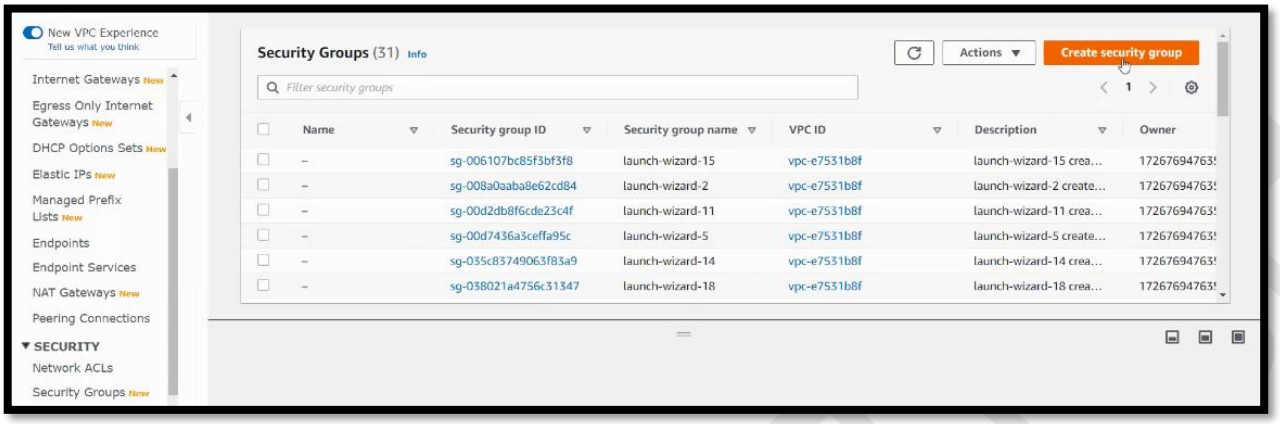
Choose our own VPC



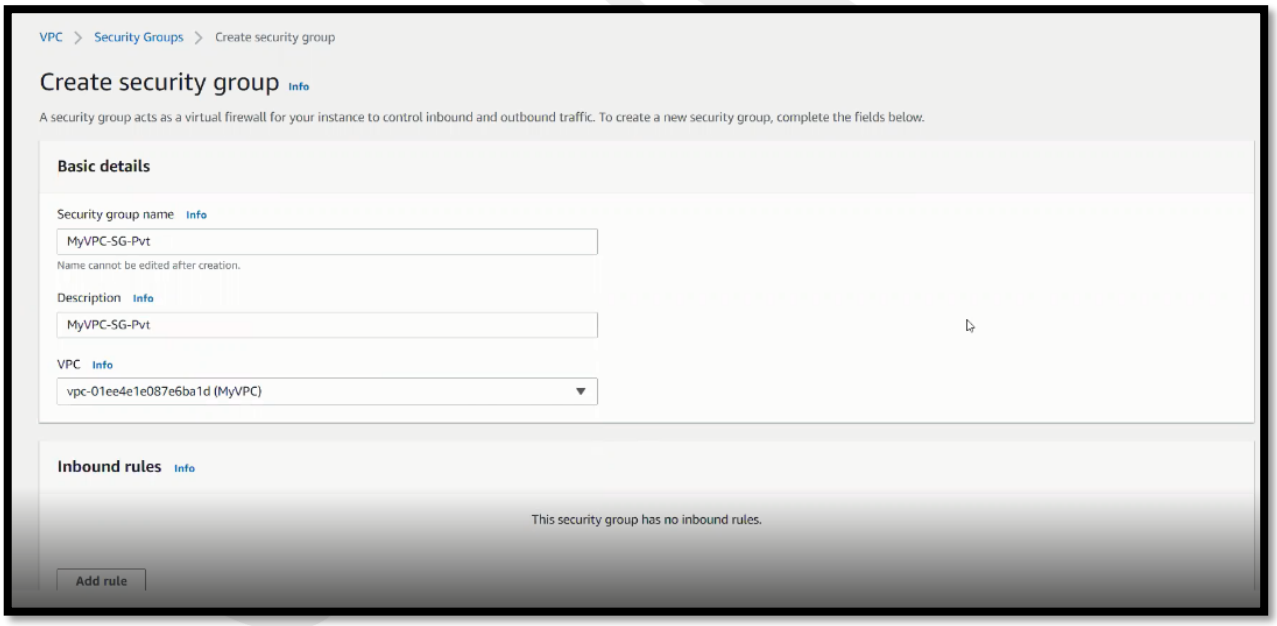


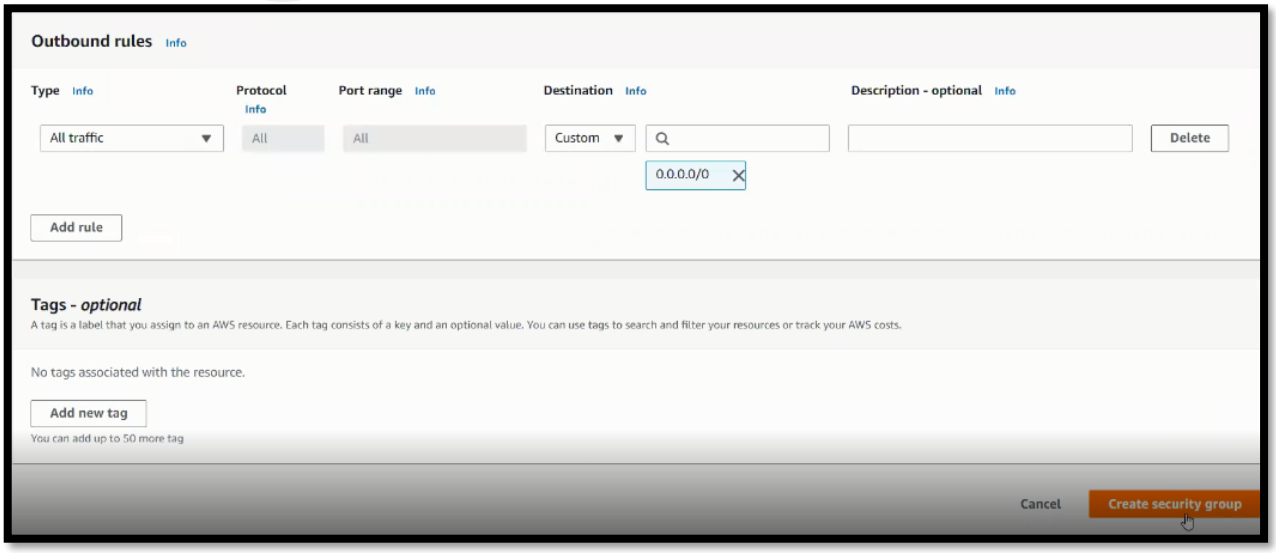
Note: Once Click on Create Security Group, Public SG will be created

Create a Private Security Group



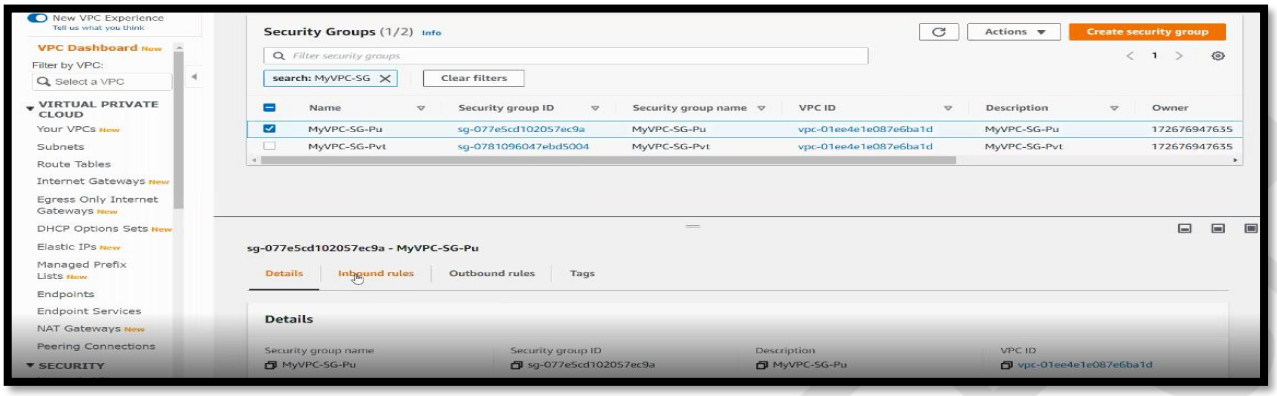
Name the Understandable Private SG name, keep the same in Description
Choose our own VPC



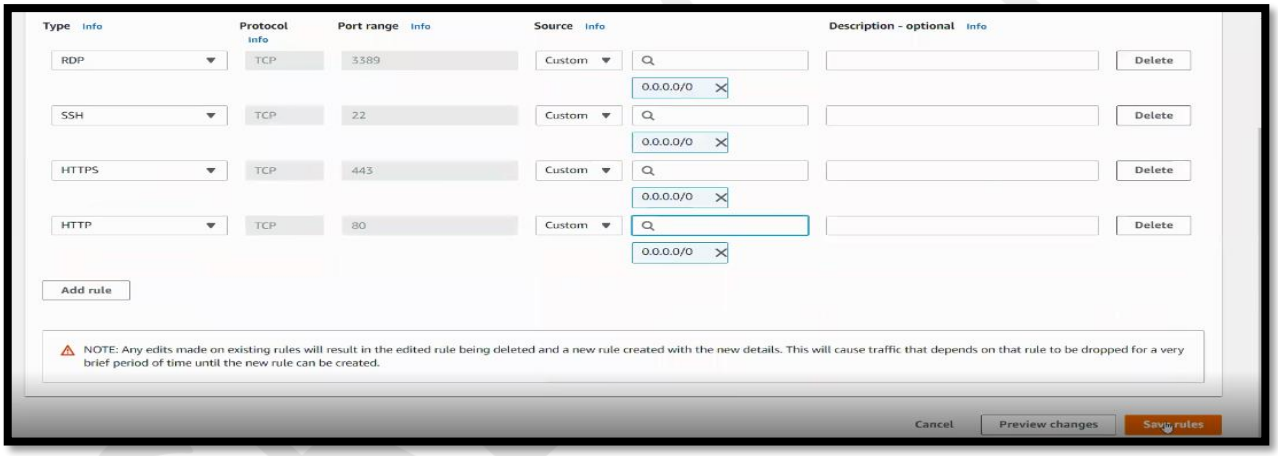


Note: Once Click on Create Security Group, Private SG will be created

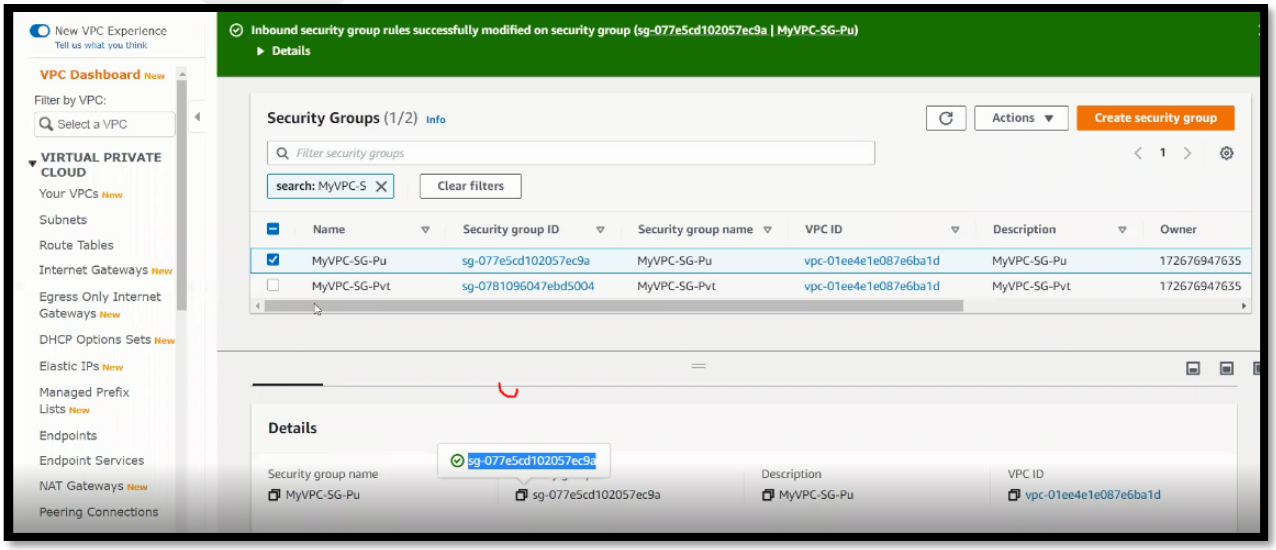
Choose Public Security Group and Edit Inbound Rules



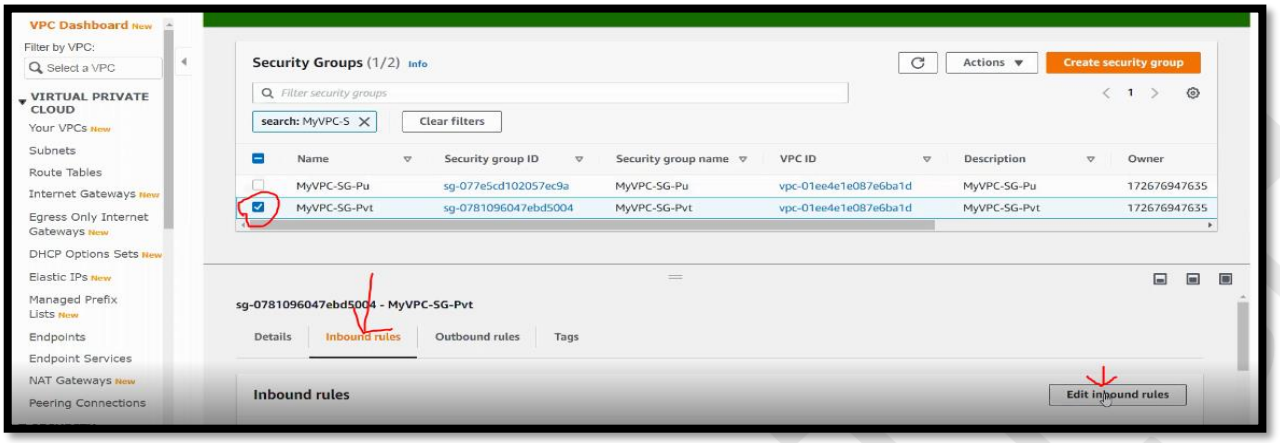
Add Firewalls RDP, SSH, HTTP and HTTPS; Source can be Public hence choose 0.0.0.0/0
Finally click on save rules



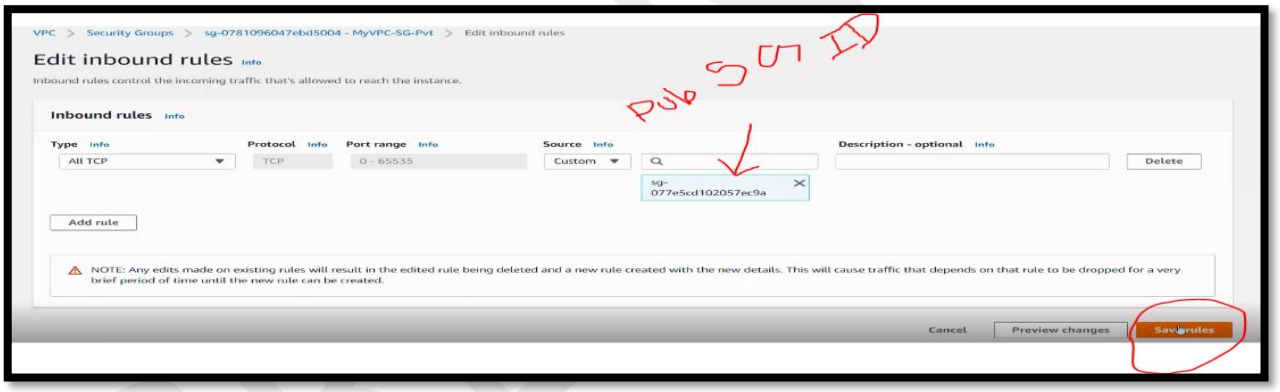
Choose private Security Group and Edit Inbound Rules
First Copy the Public Security Group ID



Choose Private Security Group and Edit Inbound Rules

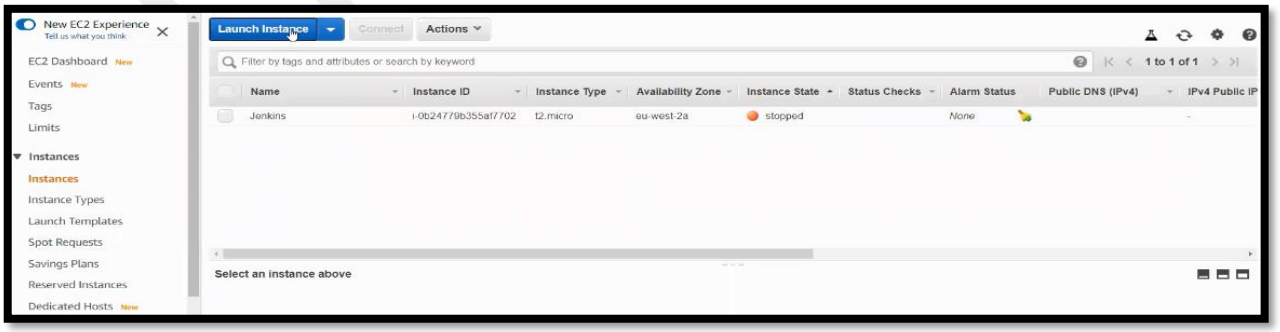


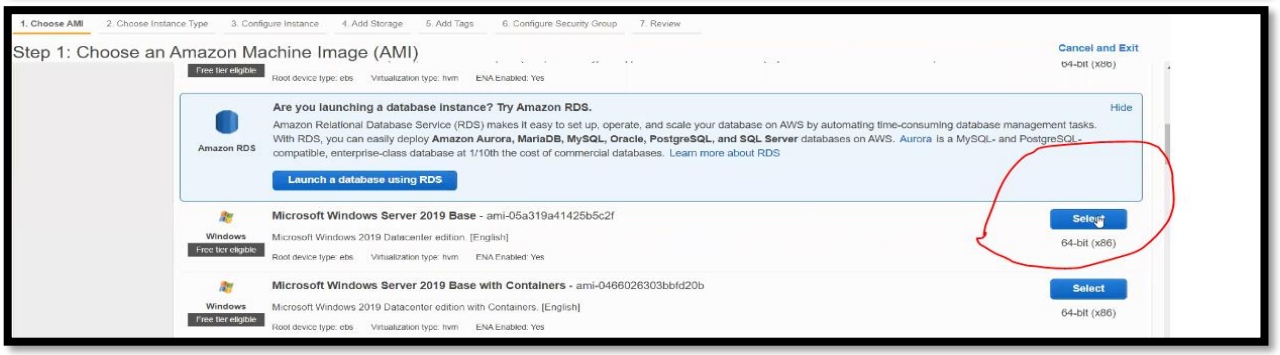
Choose All TCP (0-65535) ports; Paste Public Security Group ID in Source then Save Rules

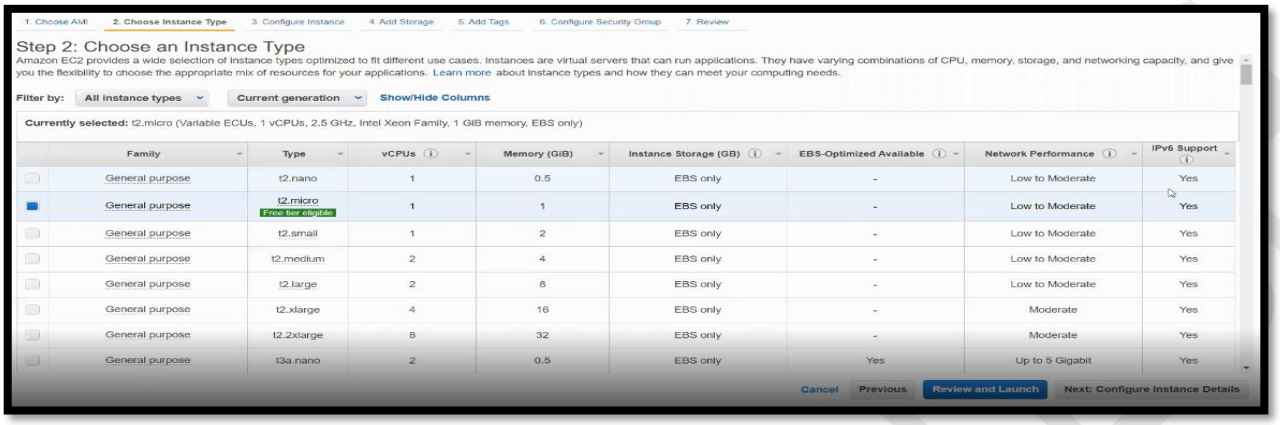


Step7: Create an Public and Private Windows Instances

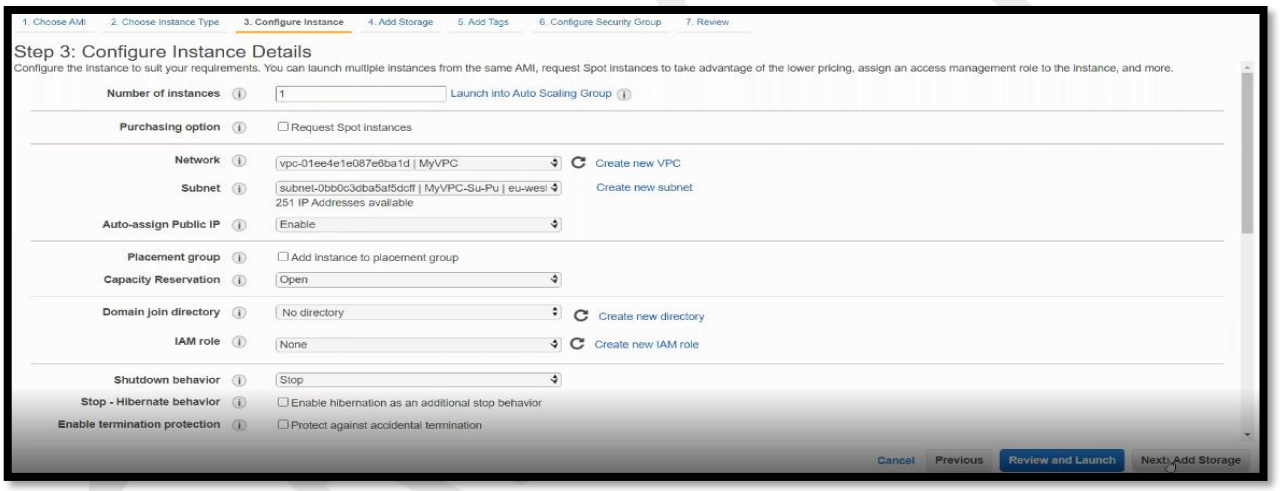
Create Public:

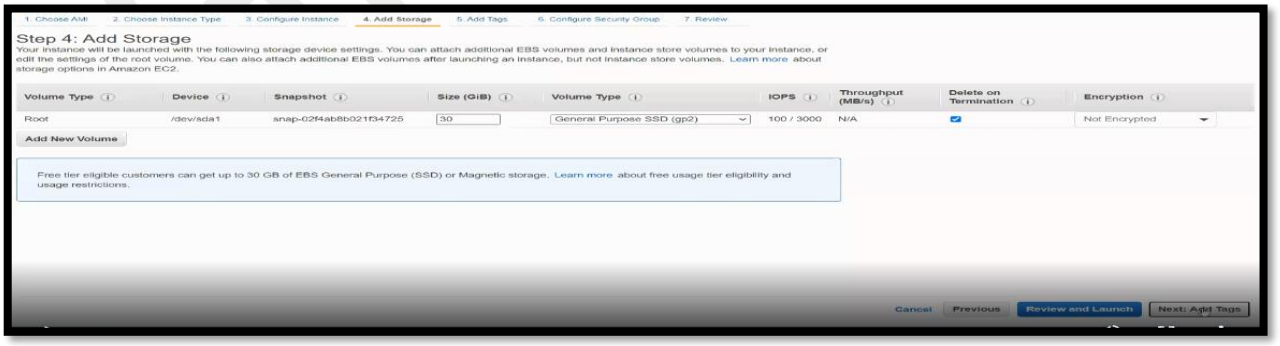


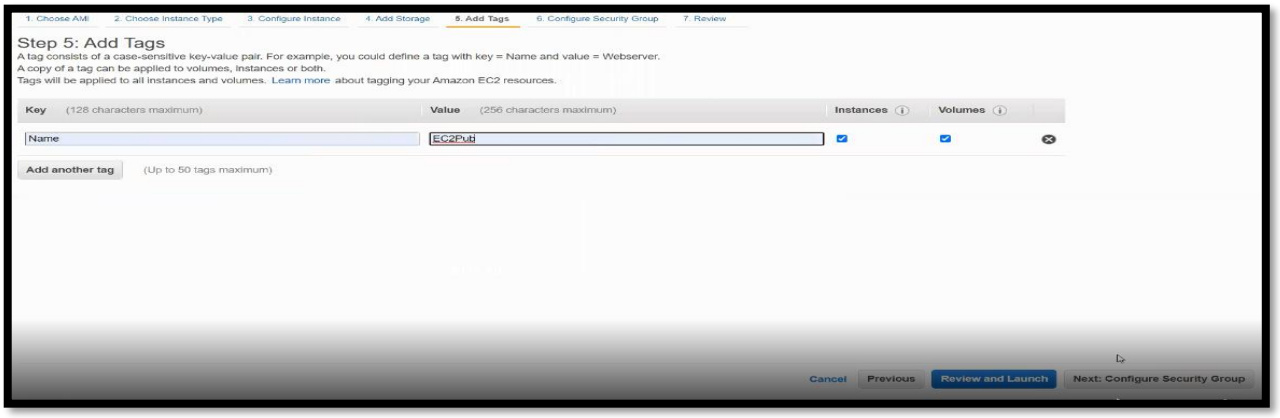




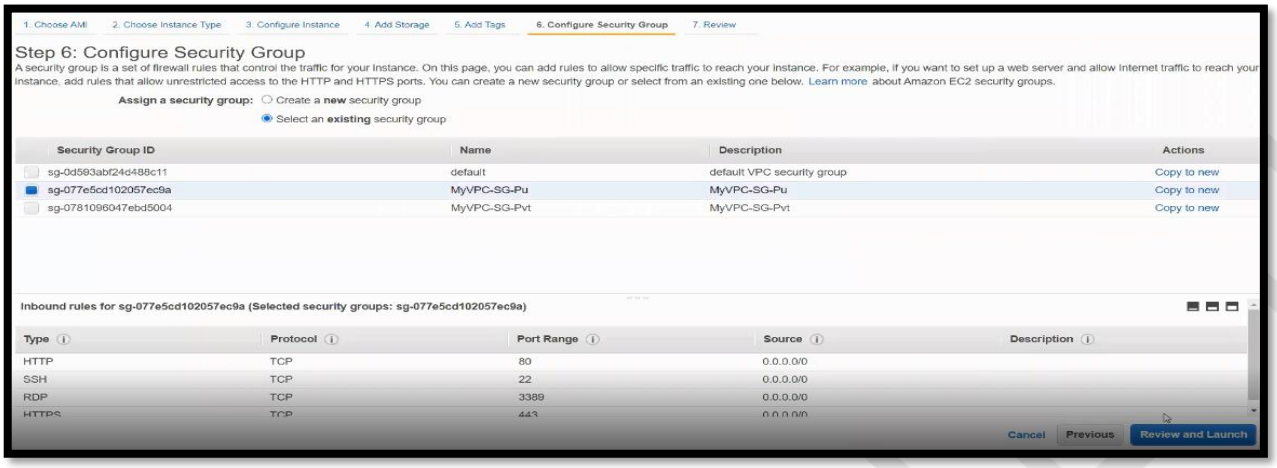
Select Own VPC, Public Subnet and Enable Auto-assign Public Ip



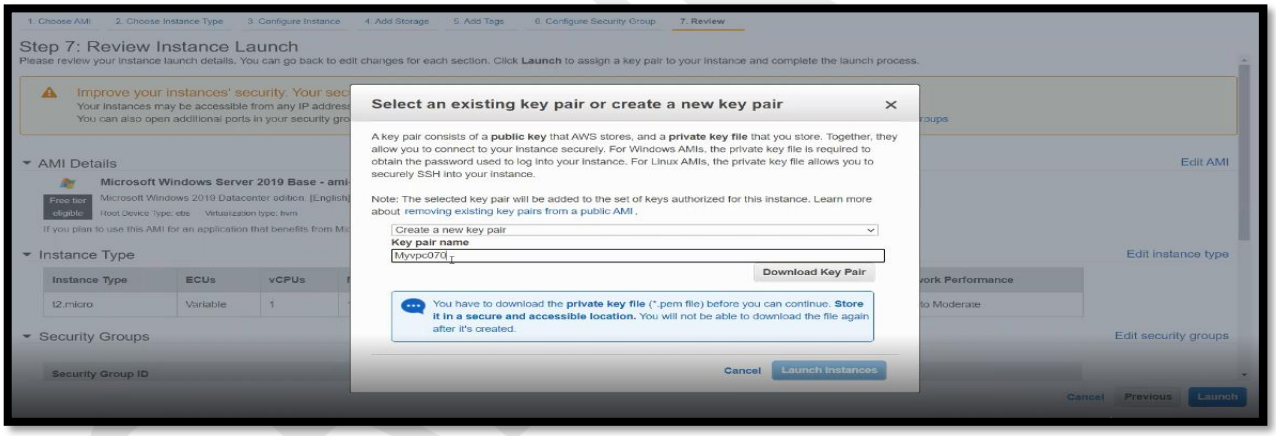




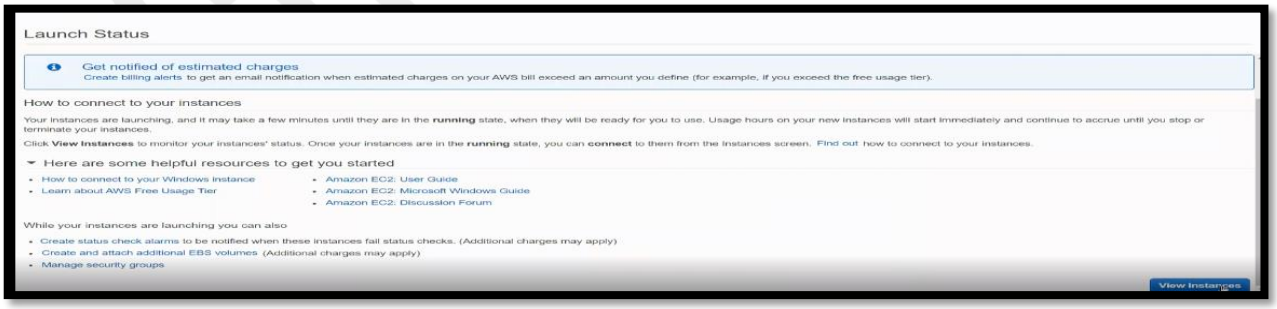
Select existing Public Security Group

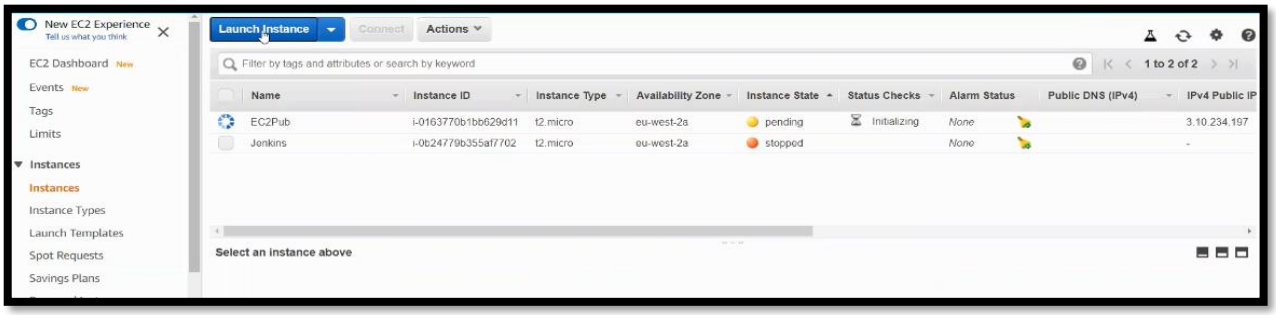


Create a pem file – Encrypted Private Key file in .pem format

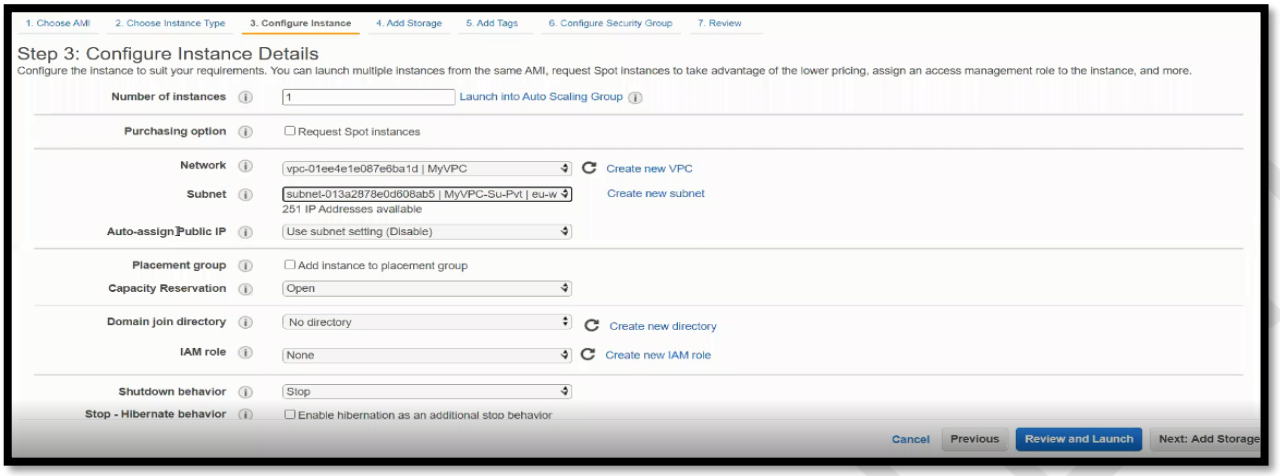


Instance Creation started



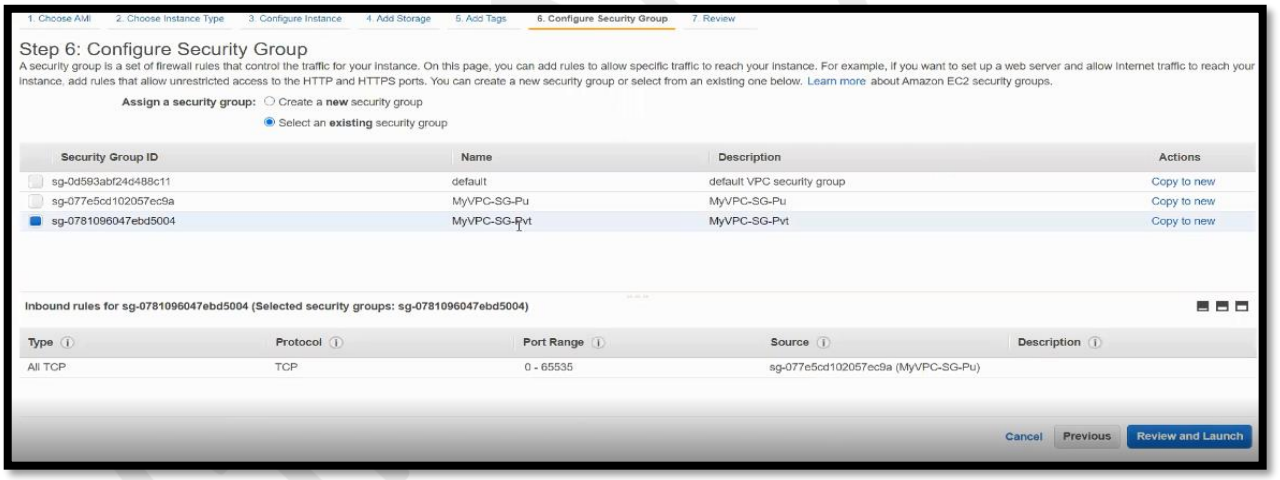


Except Step3, 4 and 6 rest all are similar where we created the Public instance
Select Own VPC, Private Subnet but Auto-assign Public IP should be disabled
Then only we can give rights to access Private Machine only through Public

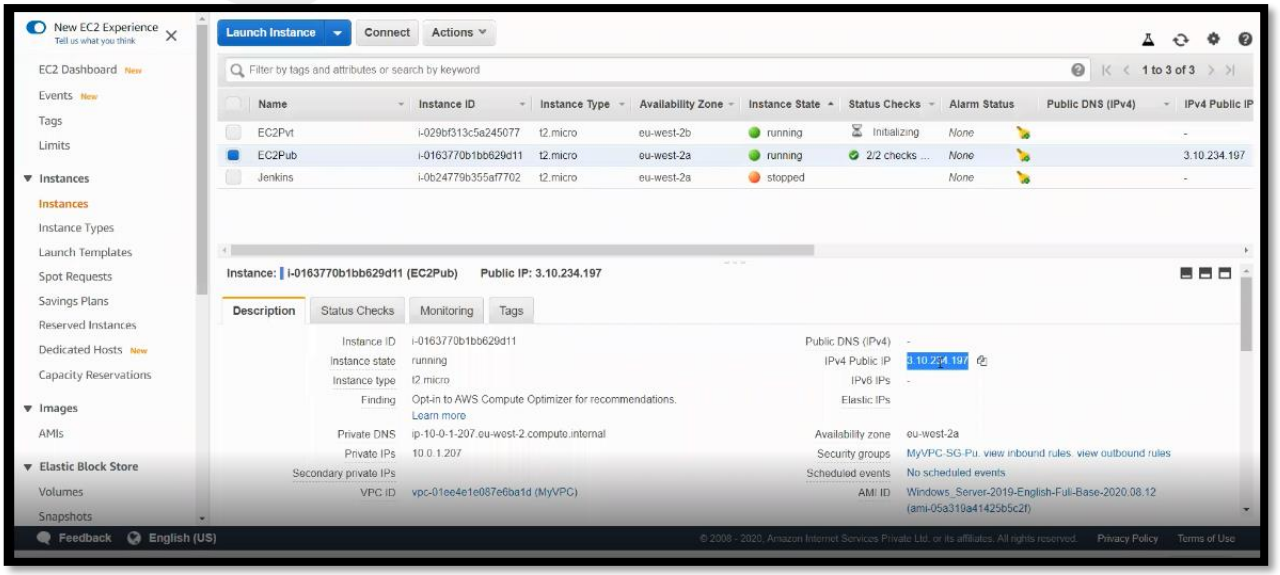


Step4 Tag Name could be different

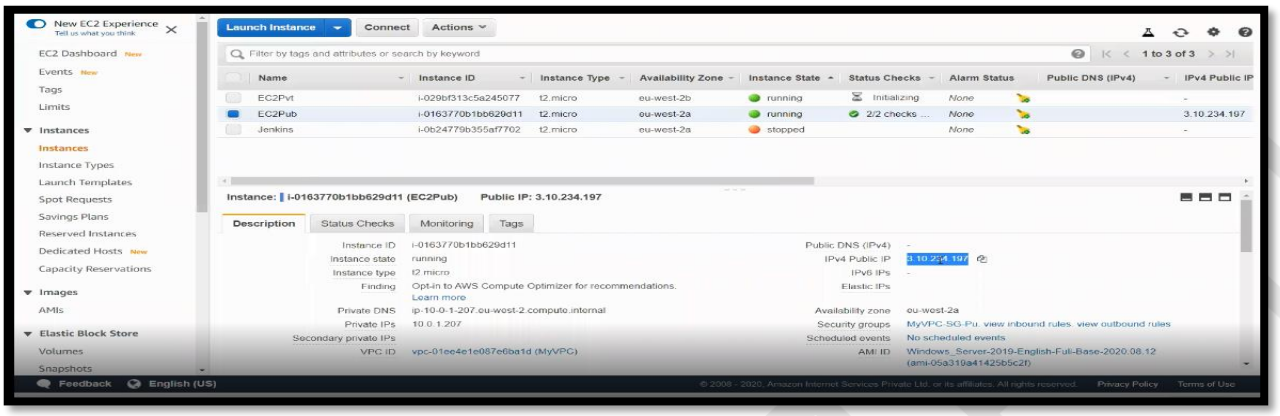
Select Private Security Group and create instance with existing pem



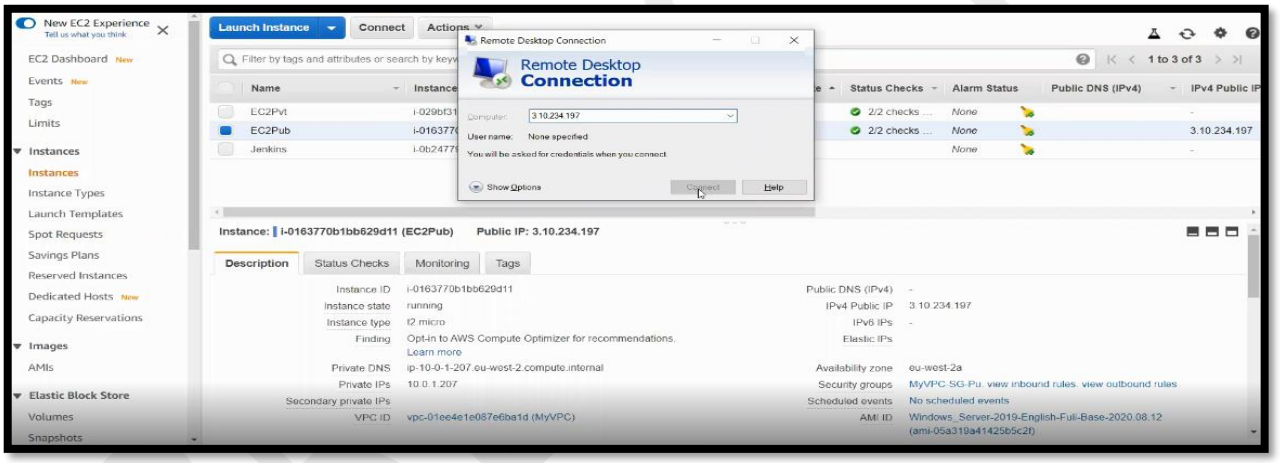
Two instances has been created successfully



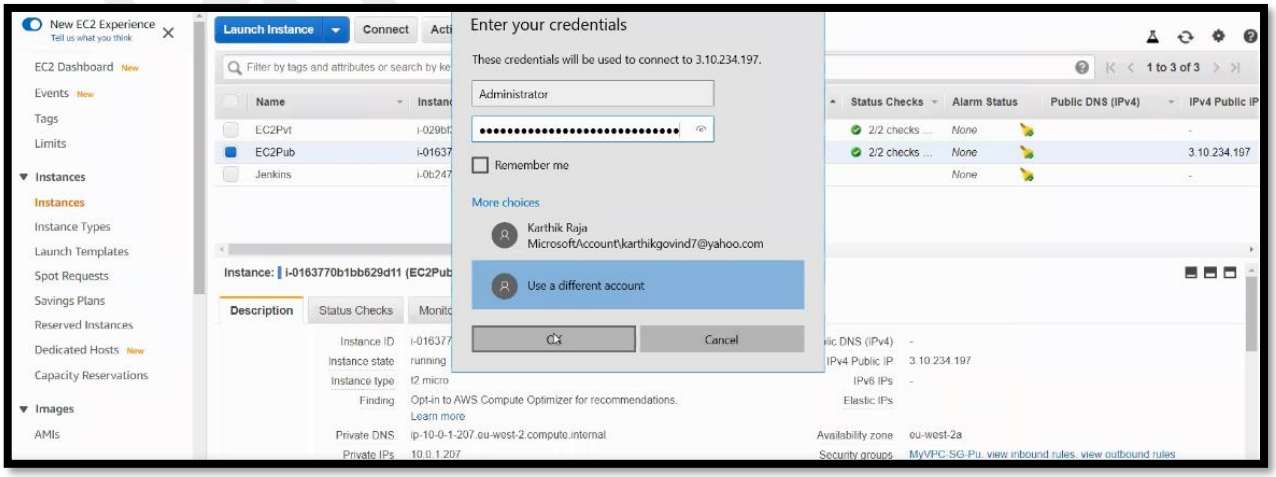
Login Public Machine and Test Internet Availability
Get the Public Ip



In the Window start, Select Remote Desktop Connection
Pass on Public IP as a Parameter

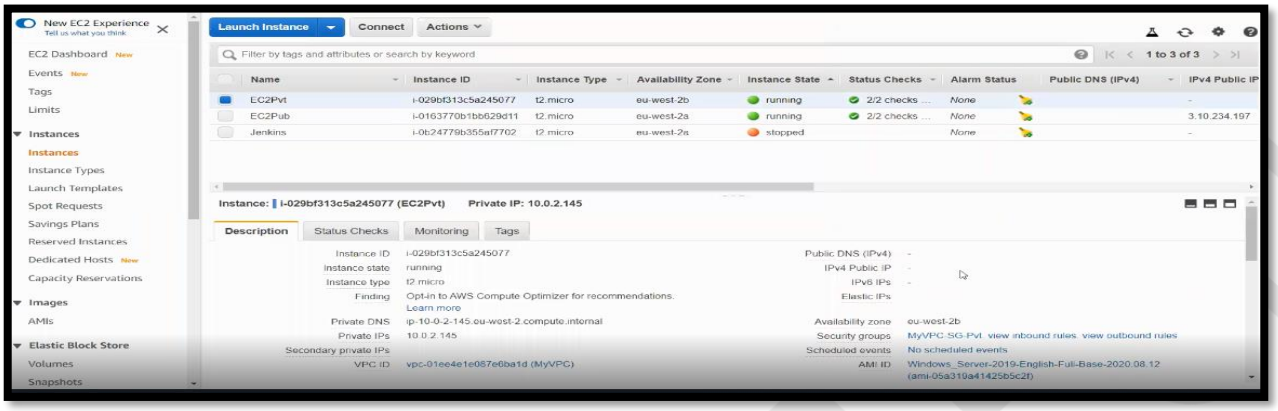


Get the User name and Password and pass like below

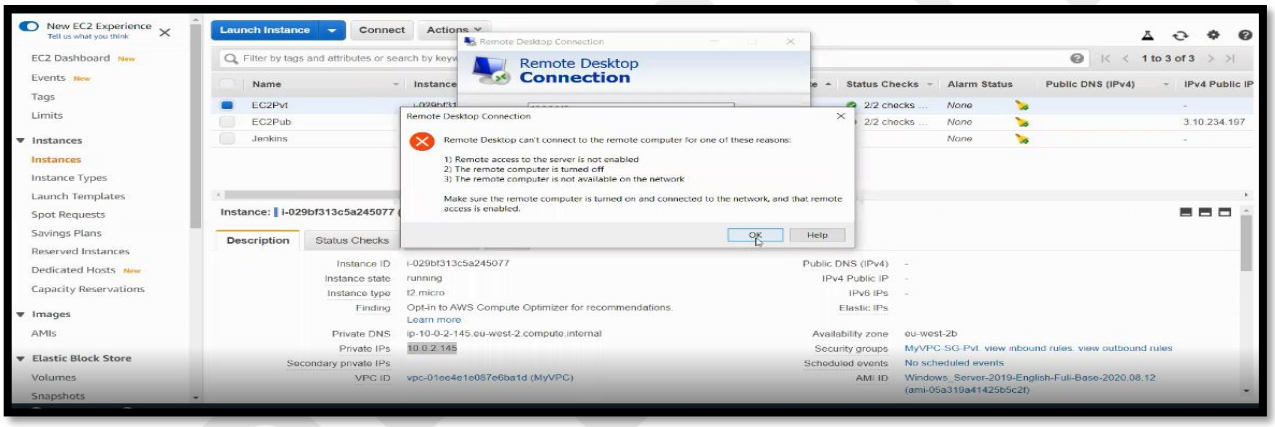


Once VM loaded check the Internet availability at the right bottom cornor

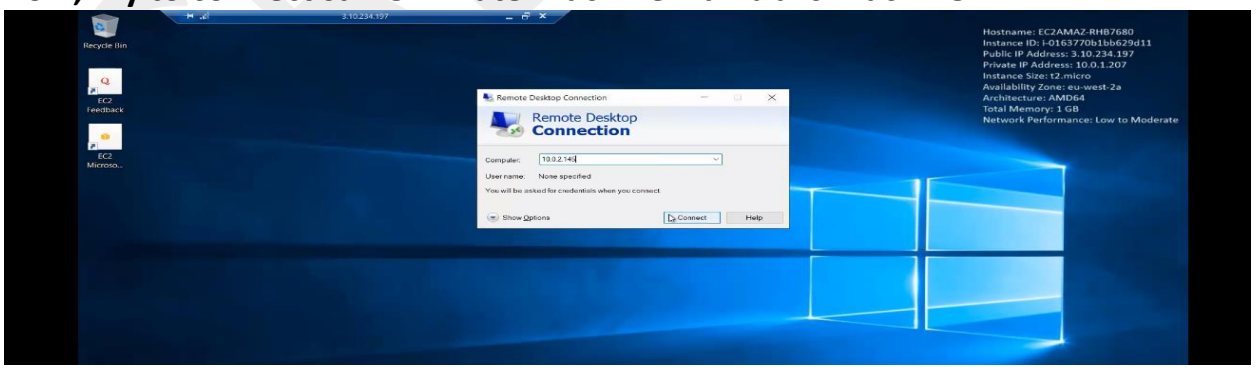
Private machine Doesn’t have Public IP

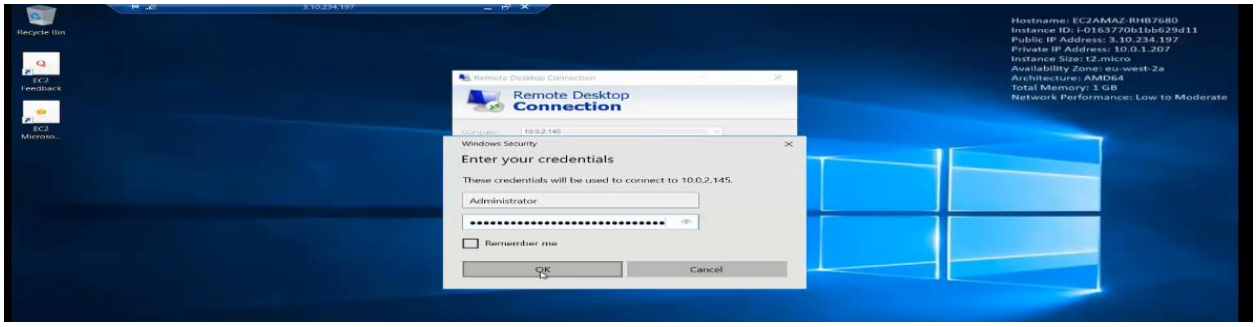


Try to connect Private Machine from Local PC



Now, Try to connect same Private Machine via Public Machine

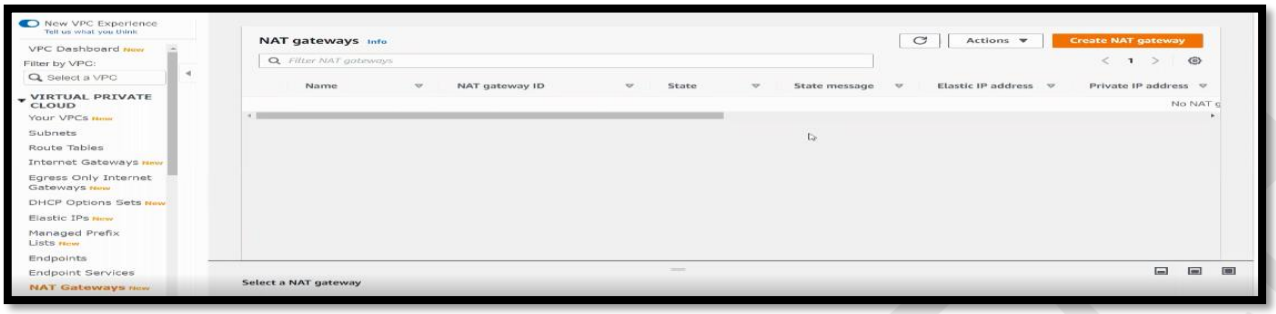




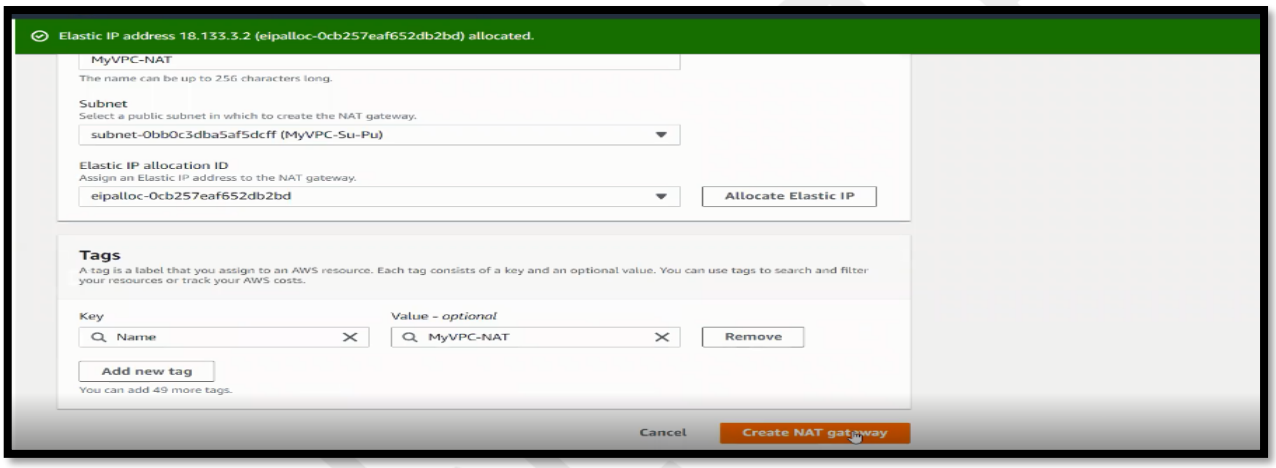
Note: In the Private Machine Internet will not be available

Solution: Need to Create NAT Gateways, Connect to Public Subnet also connect with RT

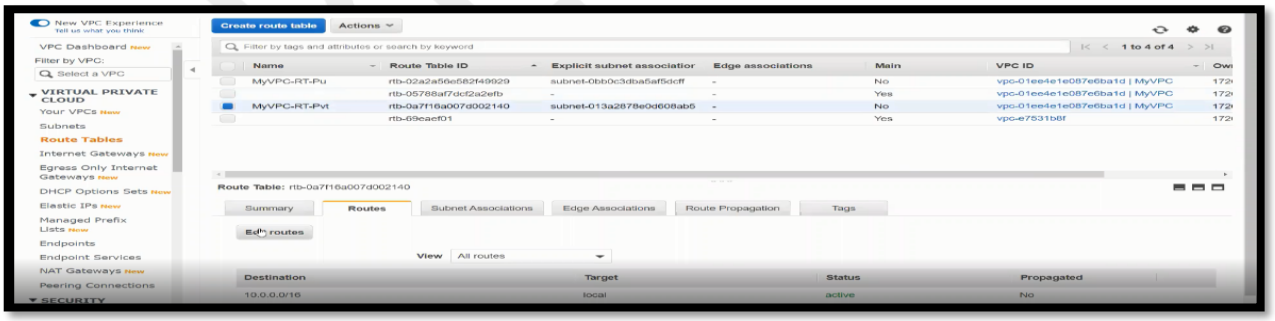
Create an NAT Gateway

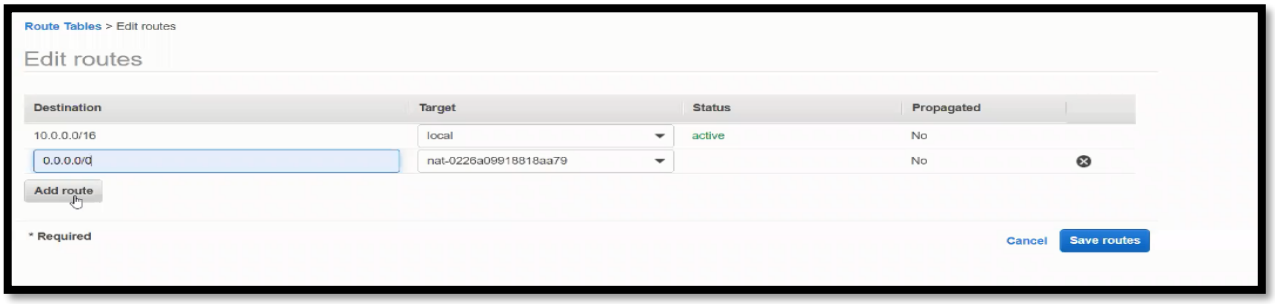


Pass all parameters properly, Create Static ip



In Private RT, associate with NAT Gateways





NOTE: Private machine now has Internet facility.

Demolish:

- **Delete EC2 Machines**
- **Delete NAT Gateway**
- **Delete VPC**
- **Delete Elastic IP**

Assignment: Create an Amazon Linux instances and check the same.