Amazon Virtual Private Cloud (VPC)

What is VPC?

- Amazon VPC enables you to provision a logically isolated section of the Amazon web services (AWS) cloud where you can launch AWS resources(EC2,DB etc) in a virtual network that you define.
- This virtual network closely resembles a traditional network that you'd operate in your own datacenter.

Why it is used?

- You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

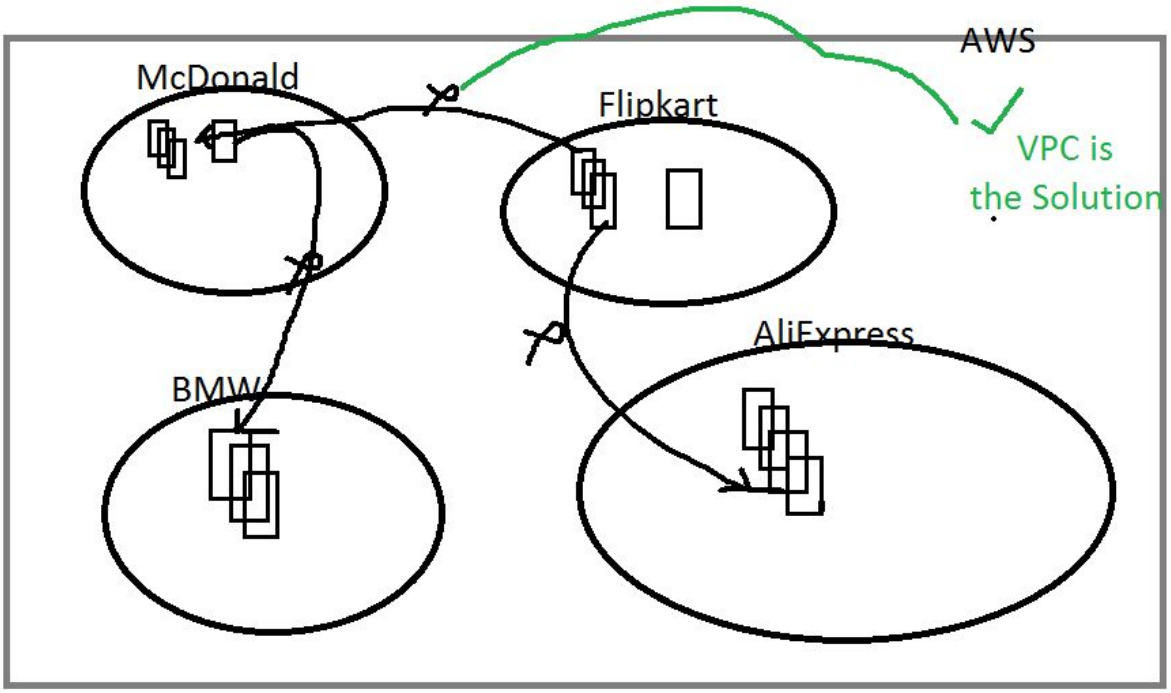
(This is like creating a Data center within our organization with the benefits of using the scalable infrastructure of AWS)

How ?

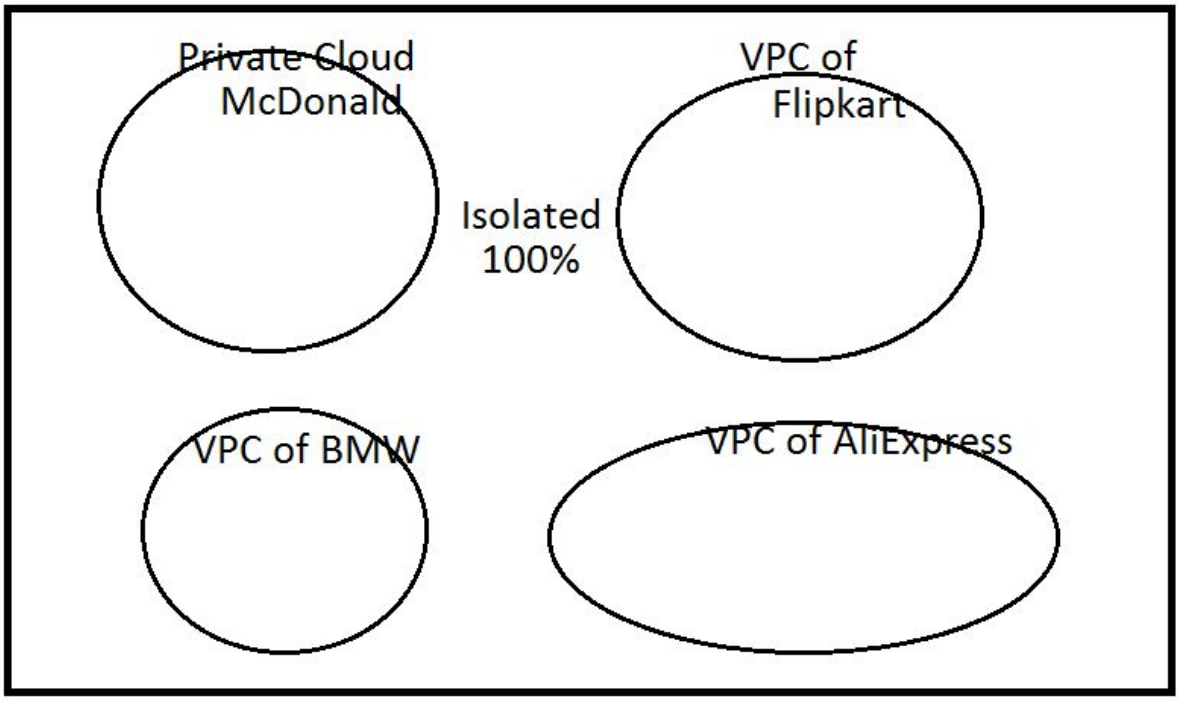
With few clicks if you know the basic concept of networking

Purpose

- Logically isolate network for AWS resources
- Logically secure yourself from other tenants
- Provides security
- Isolation



There was a possibility that other tenants can compromise the security of your network hence AWS given a solution of Virtual Private Cloud.
without VPC you can't use EC2 services



Key Concepts

VPC : A Virtual Private Cloud is a sub cloud inside the AWS public cloud. Sub-cloud means it is inside an isolated logical network.
→ Other servers can't see instances that are inside a VPC. you can launch your AWS resources, such as Amazon EC2 instances, into your VPC. You can configure your VPC, you can select its IP address range , create subnets, and configure route tables, network gateway and security settings.while doing LAb, we can use default VPC as amazon is providing that.
NOTE: By default, instances communicate between different subnets if it is in same VPC

Subnet :

- A subnet is a range of IP addresses in your VPC.
- You can launch AWS resources into the subnet that you select.
- Use a public subnet for resources that must be connected to the internet and a private subnet for resources that won't be connected to the internet.
- A subnet is a sub-network inside a VPC. An example of a subnet inside a VPC (10.123.X.Y) is 10.123.1.A/24. This means any instance that belongs to this subnet will have an IP 10.123.1.A where **A** can be anything between 2 and 254. These are also known as CIDR notations. (CIDR- Classless Inter-Domain Routing)
- CIDR is a method of assigning your IP address and defining your IP routings.
- A instance always belongs to a subnets. You cannot have an instance inside a VPC that does not belong to any subnets, while spawning instances inside AWS VPC , one must specify which subnet the instance should belong to.

Security Group :

- Act as a firewall for associated amazon EC2 instances, controlling both inbound and outbound traffic at the instance level.

Network Access Control List (ACLs)

- Act as a firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level.

Internet Gateway

- Internet gateway allows communications between instances in your VPC and the internet.
- It is horizontally scaled, redundant, and highly available VPC component
- It therefore imposes no availability risks or bandwidth constraints on your network traffic.
- It always connected to VPC

NAT Device:

- NAT device is to enable instances in a private subnet to connect to the internet

- For example, in the case of software updates or other AWS services, but prevent the internet from initiating connections with the instances.

Route Tables

- A route table contains a set of rules, called as **routes,** that are used to determine where 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.

- Network traffic of any instance inside a subnet is dictated by a routing table.
  An example of routing table is :

CIDR --	target
10.123.X.Y/16	local
0.0.0.0/0	igw

  Above example shows that any traffic destined for 10.123.X.Y IP (where X and Y can be anything from 2 to 254) will be sent directly. The rest of the traffic will be directed to igw

Network Access Control List

- It 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 setup network ACL with rules similar to your security groups in order to add an additional layer of security to your VPC.
- Apart from routing tables, each subnet also assigned a network ACL. Network ACLs specify what type of traffic is allowed inside the subnet. By default, it might have the following rules

Rule number	port	protocol	source	action
100	ALL	ALL	0.0.0.0/0	allow

This means that all traffic is allowed within this network. You can think of Network ACL as subnet-wide security groups. They are effective while isolating subnets from each other, reducing the collision of domains etc.

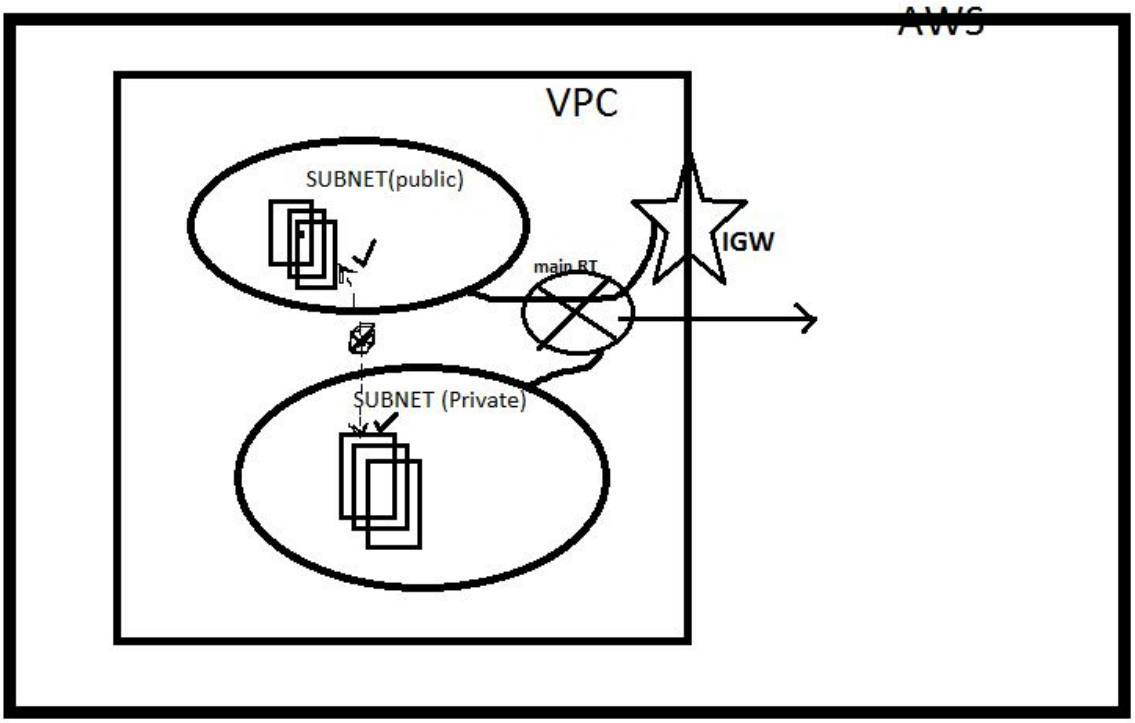
LAB

Requirement:

Some servers are publicly accessible

Some servers are on private subnet (Not reachable over internet) like Databases

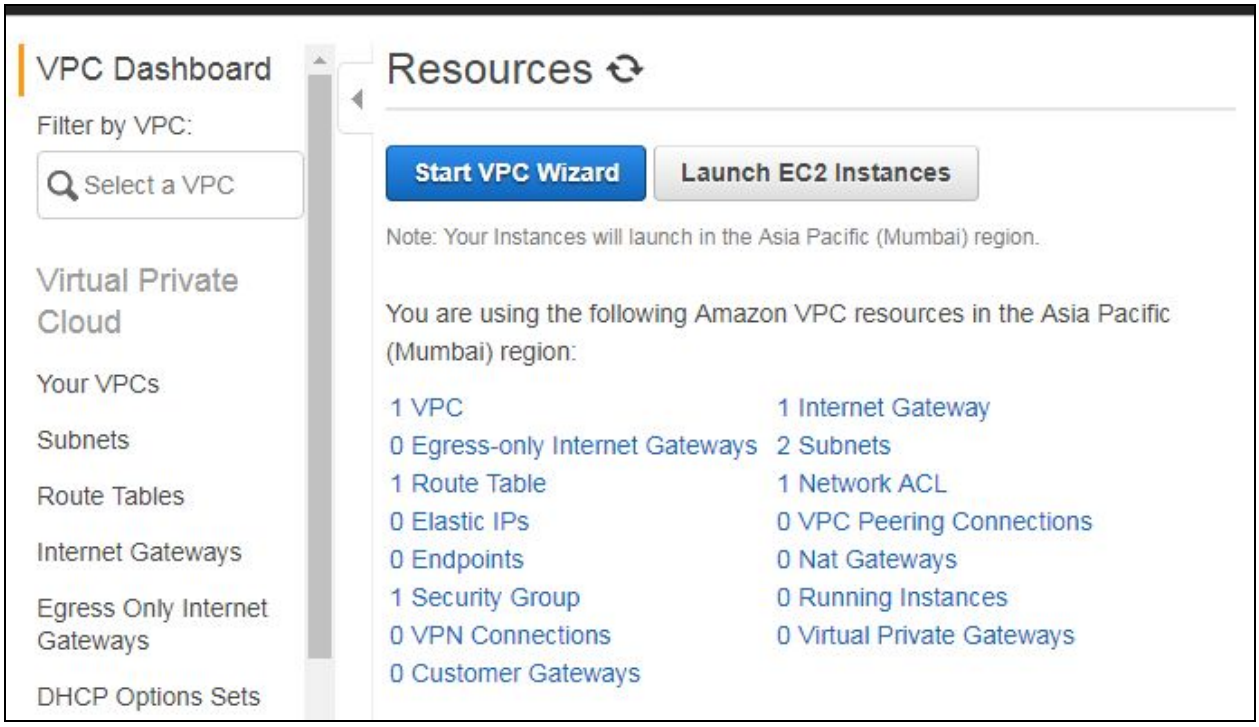
WEB (PUBLIC) → DATABASE (PRIVATE)

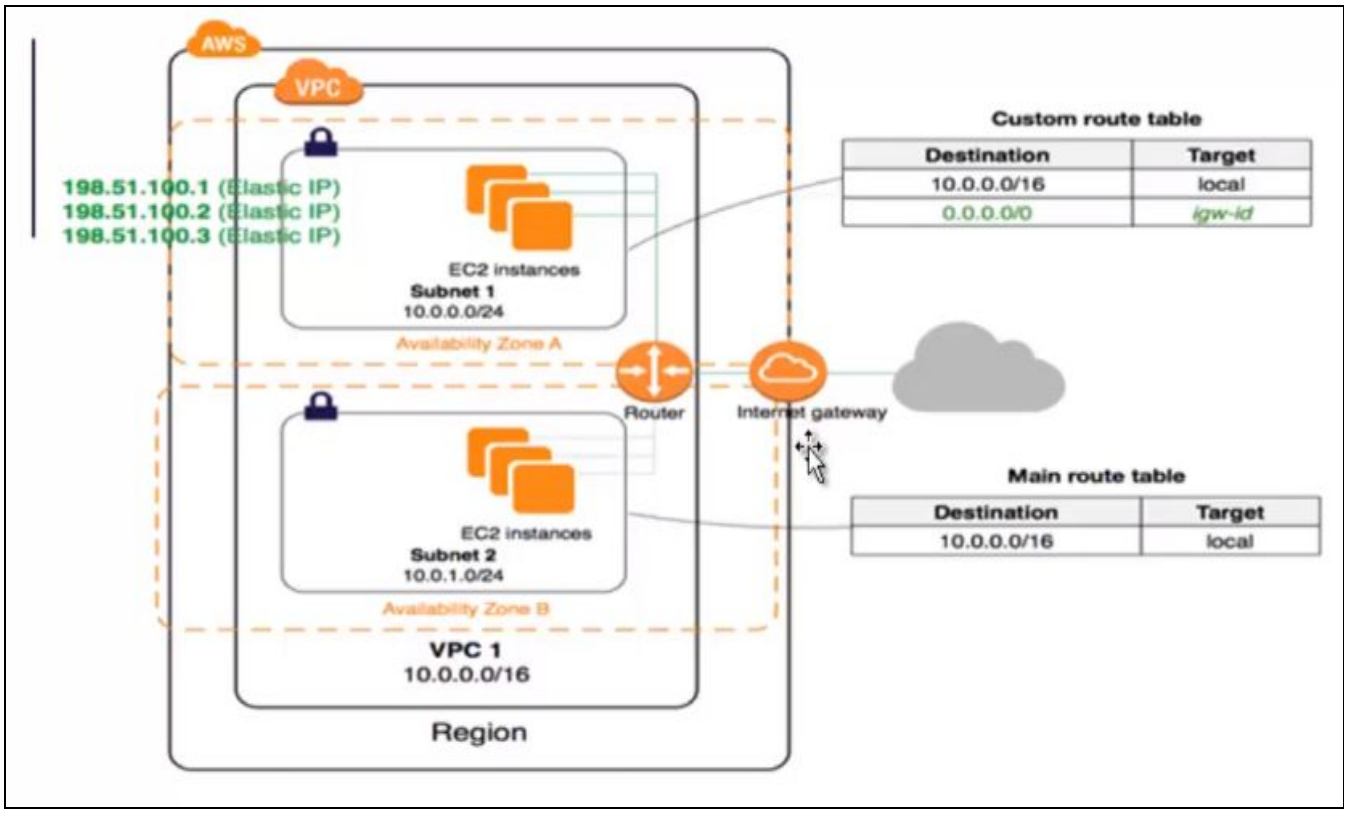


Login to management console and then check for VPC service in Networking

You will see 1 Default VPC , 1 route table , 1 security group , 1 network ACL, 1 IGW , 6 subnets (based on North Virginia Region)

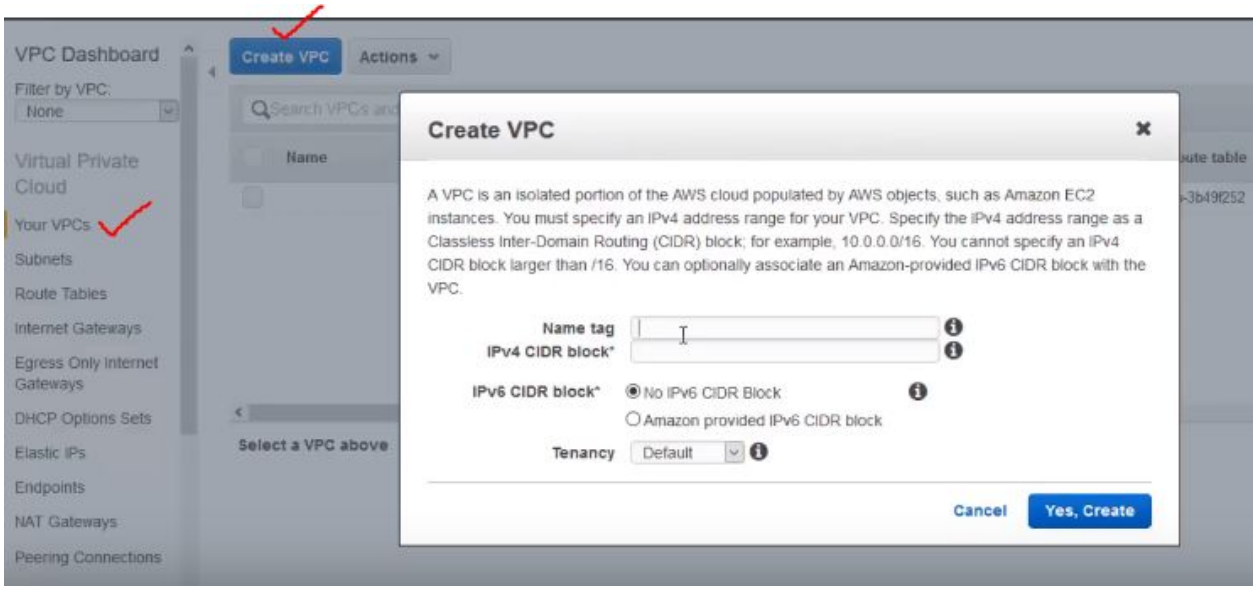
You will see 1 default VPC, 1 Route Table, 1 Security group , 1 network ACL, 1 IGW, 2 subnets (based on mumbai region)


VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Resources
Start VPC Wizard
Launch EC2 Instances
Note: Your Instances will launch in the Asia Pacific (Mumbai) region.
You are using the following Amazon VPC resources in the Asia Pacific (Mumbai) region:
1 VPC
0 Egress-only Internet Gateways
1 Route Table
0 Elastic IPs
0 Endpoints
1 Security Group
0 VPN Connections
0 Customer Gateways
1 Internet Gateway
2 Subnets
1 Network ACL
0 VPC Peering Connections
0 Nat Gateways
0 Running Instances
0 Virtual Private Gateways

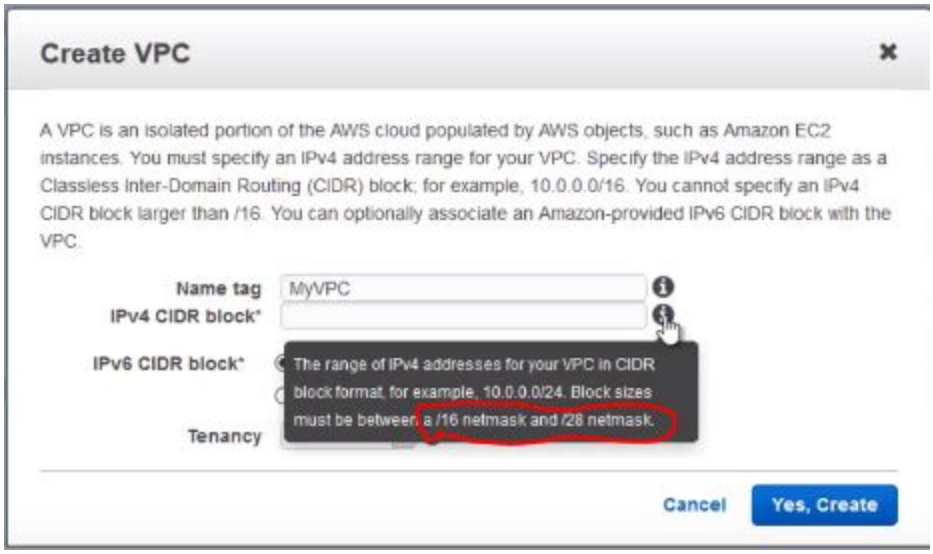

AWS
VPC
198.51.100.1 (Elastic IP)
198.51.100.2 (Elastic IP)
198.51.100.3 (Elastic IP)
EC2 instances
Subnet 1
10.0.0.0/24
Availability Zone A
Router
Internet gateway
EC2 instances
Subnet 2
10.0.1.0/24
Availability Zone B
VPC 1
10.0.0.0/16
Region
Custom route table
Destination
Target
10.0.0.0/16
local
0.0.0.0/0
igw-id
Main route table
Destination
Target
10.0.0.0/16
local

This is the architecture which we are going to implement in our AWS.
So we implement 1 VPC in a region and with 2 subnets we can give any IP range in these subnets based on our need. We make one subnet for private and other for public.
Public subnet will route out to internet but we have to place a IGW

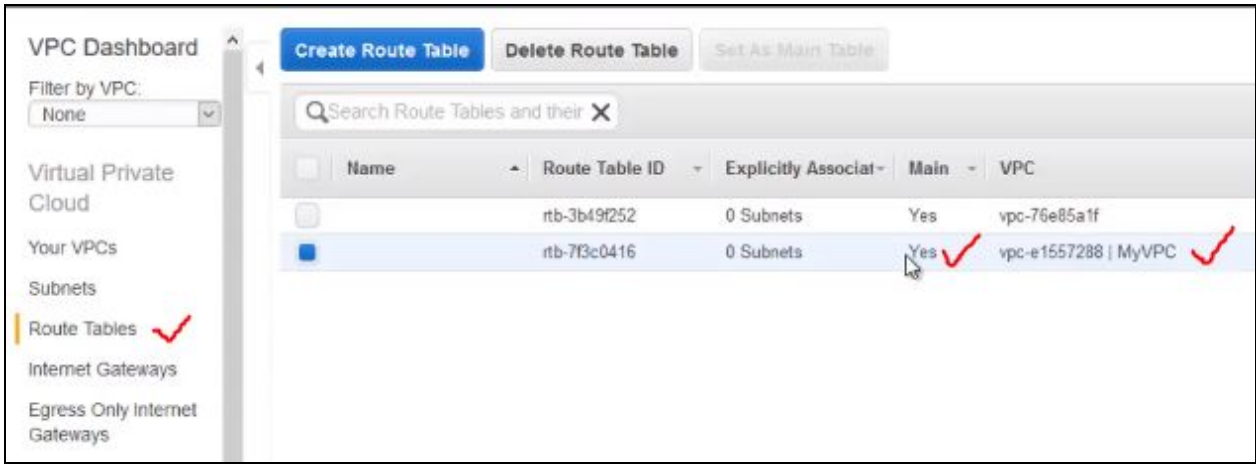
Go to your VPC tab
Click on create VPC



Fill out the fields
Name Tag : MyVPC
IPv4 CIDR block: 10.0.0.0/16 you cannot mention as 10.0.0.0/8

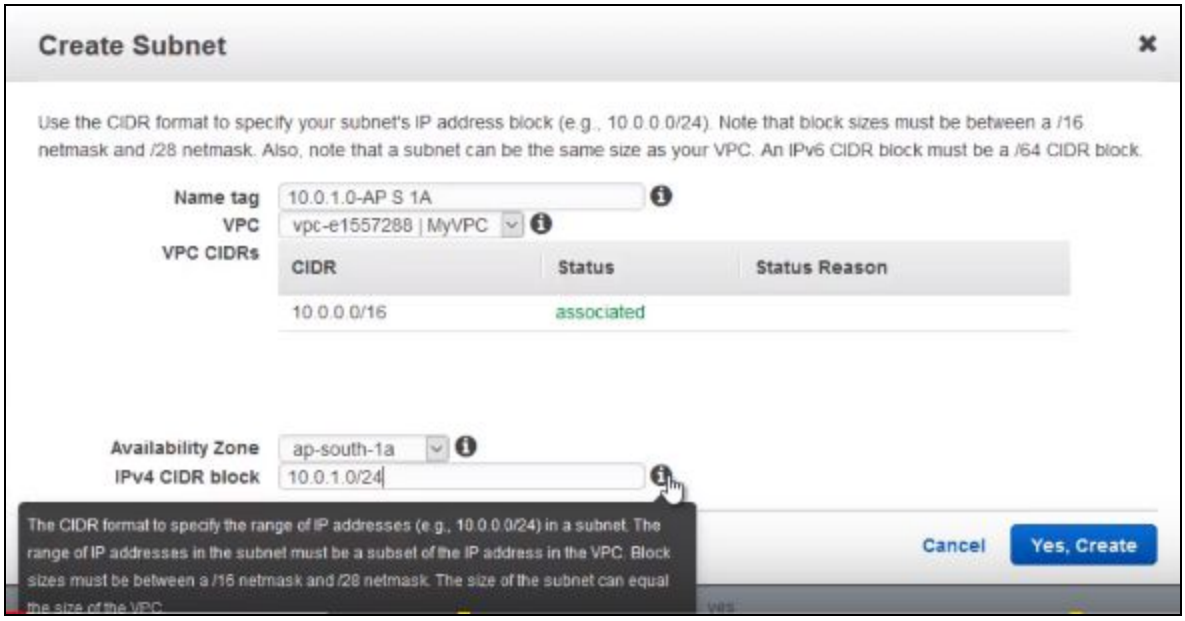


As a part of VPC , **main route table** and Network ACL and Security Group also got created. But subnet and IGW will not be created.

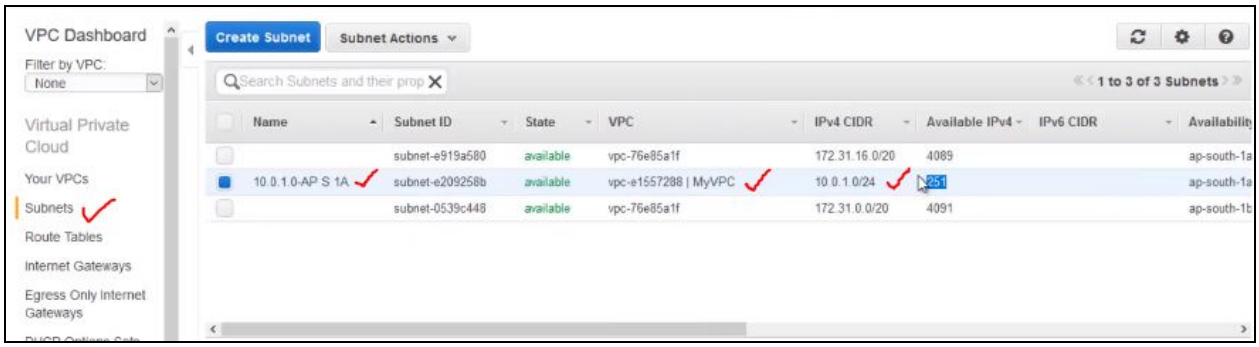


Since subnet is not created so we will create now.

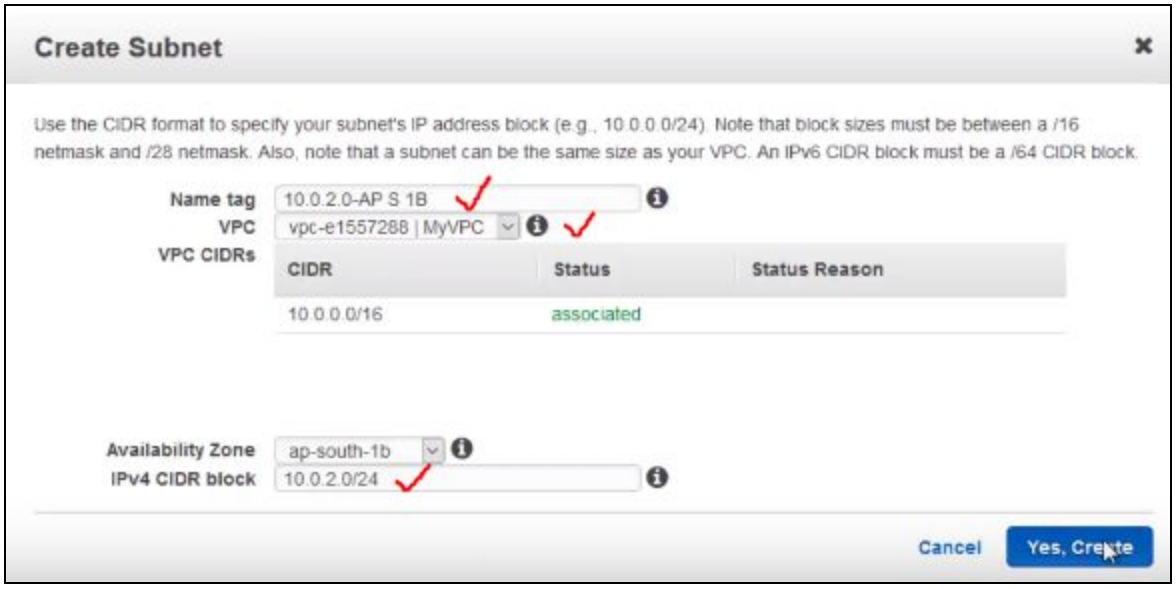
Go to subnet tab and click on create subnet



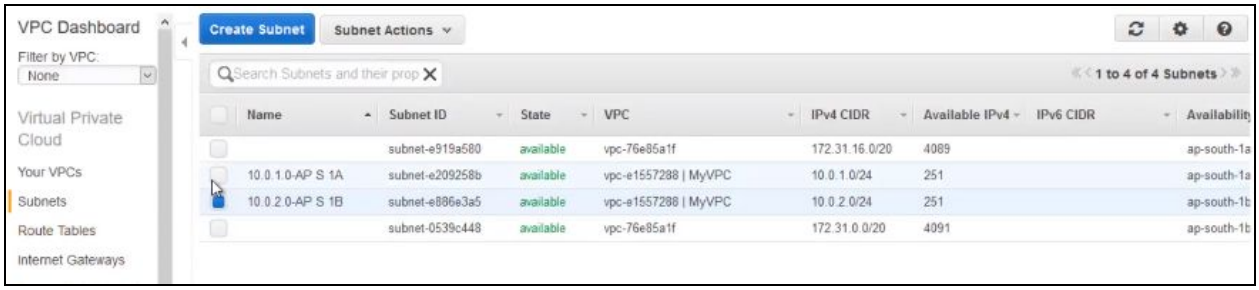
Verify



Create one more subnet in another AX

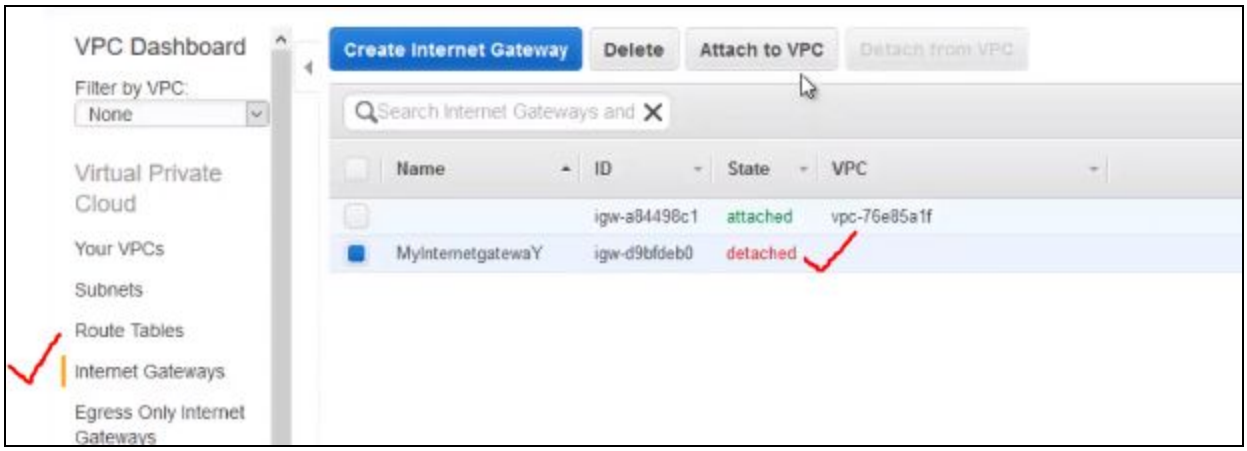


Verify :



Now I want to make sure 1A subnet should be public subnet. We can make it by defining IGW

Create a IGW

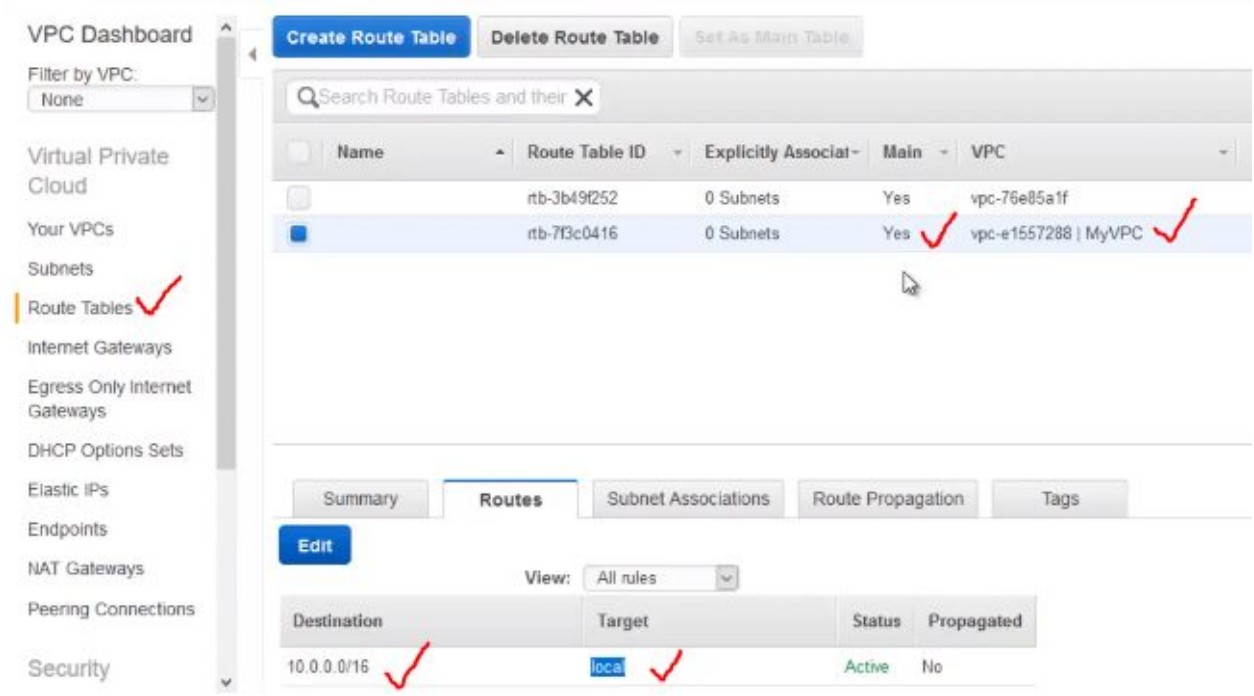


Now attach that IGW to VPC

Note: One VPC , it can only be attached to single IGW.
Route Table

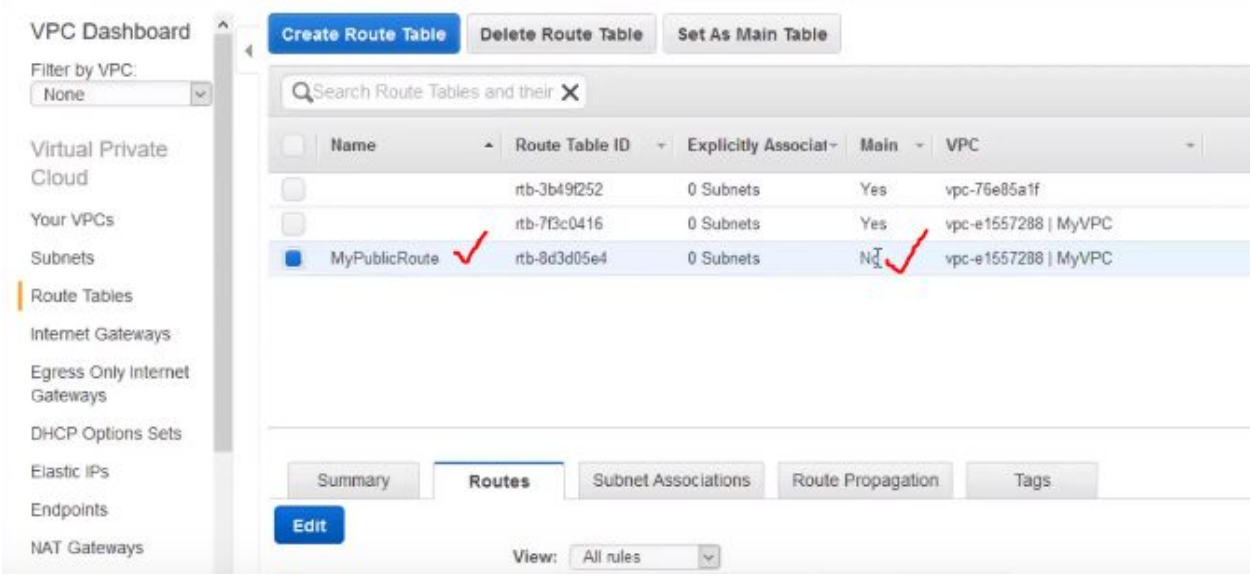
We have to make sure that there is a route out to internet so that instances launched into the public subnet , it should have access from internet

Don’t change the main route table of MyVPC



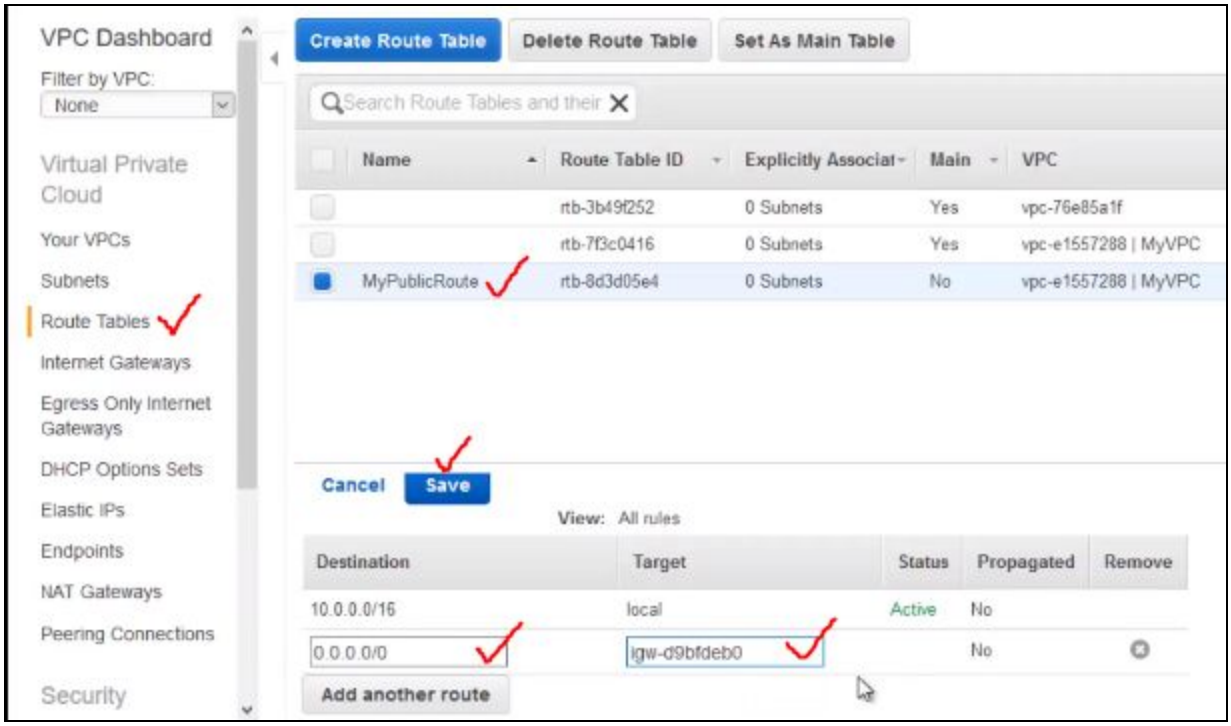
Create a new Route table

Verify



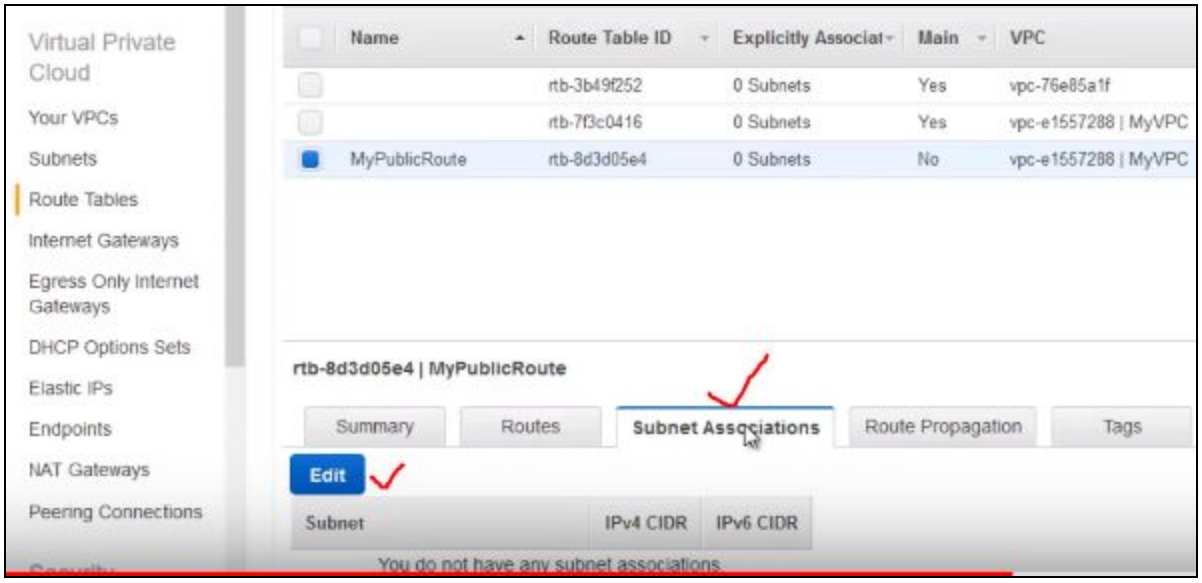
NOW CREATE A ROUTE OUT TO INTERNET SO THAT THE TARGET IS IGW FOR THIS PARTICULAR PUBLIC SUBNET

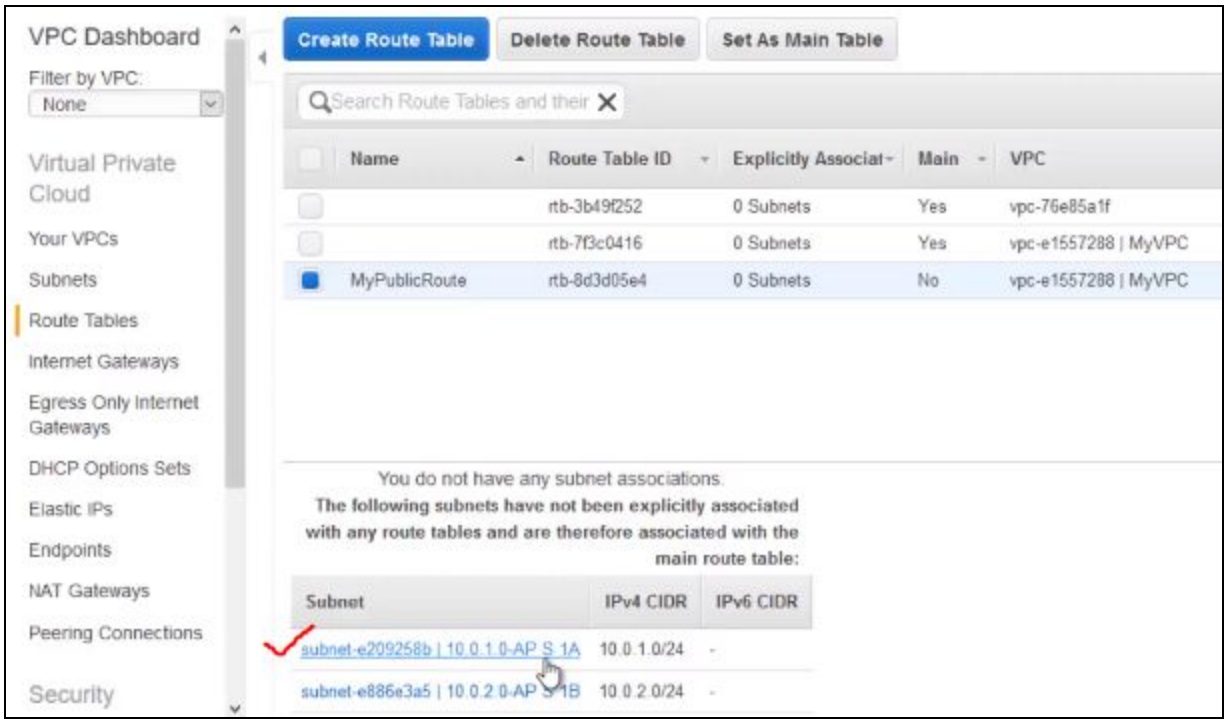
In Route Tab , click on Edit

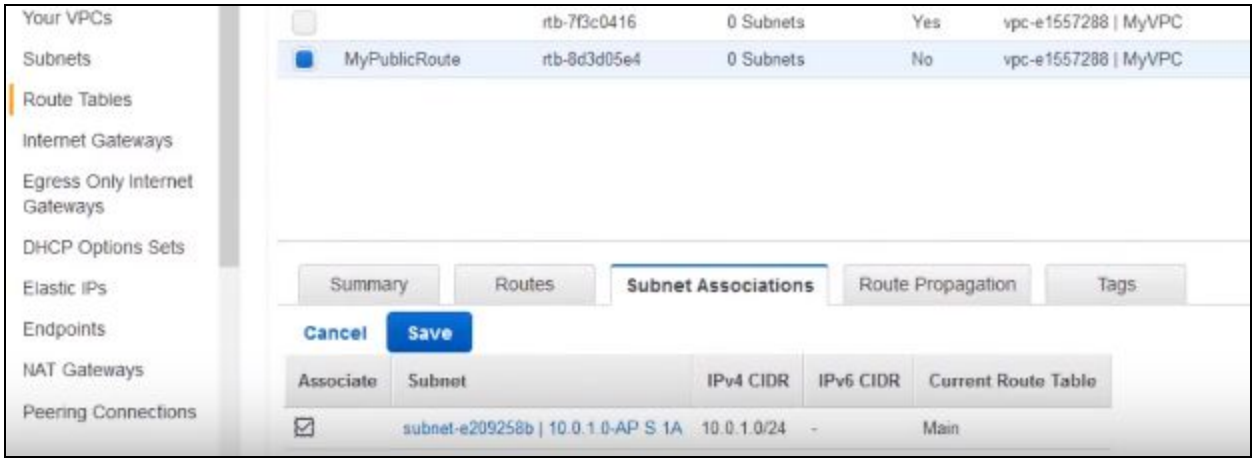


Now we define route
But still we did not associate to publicly created subnet.

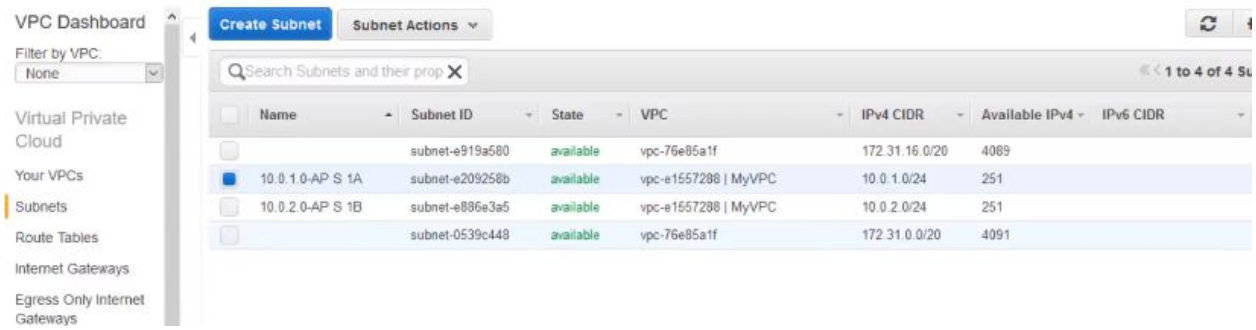
Click on Subnet Association

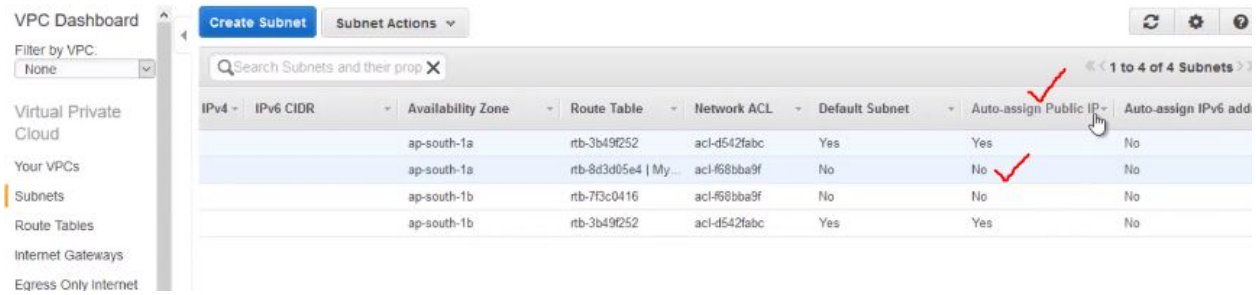




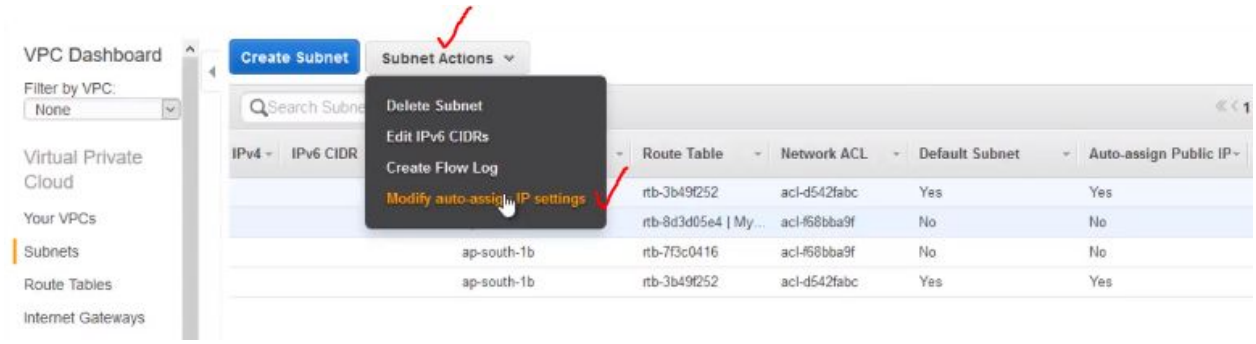


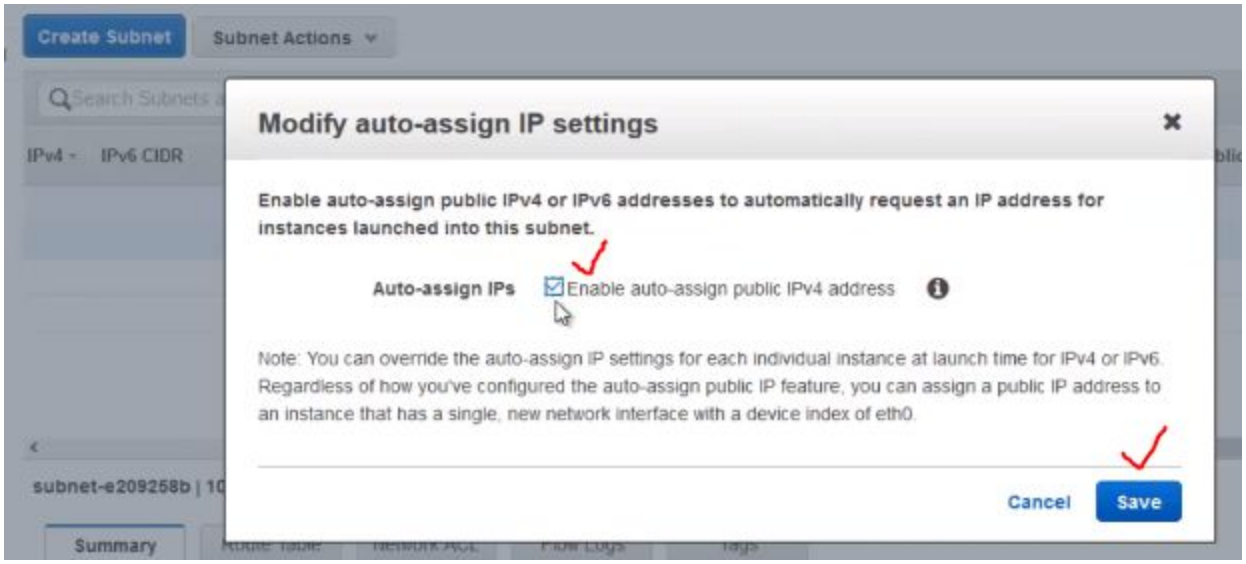
Click on Edit , select 1A and Save it



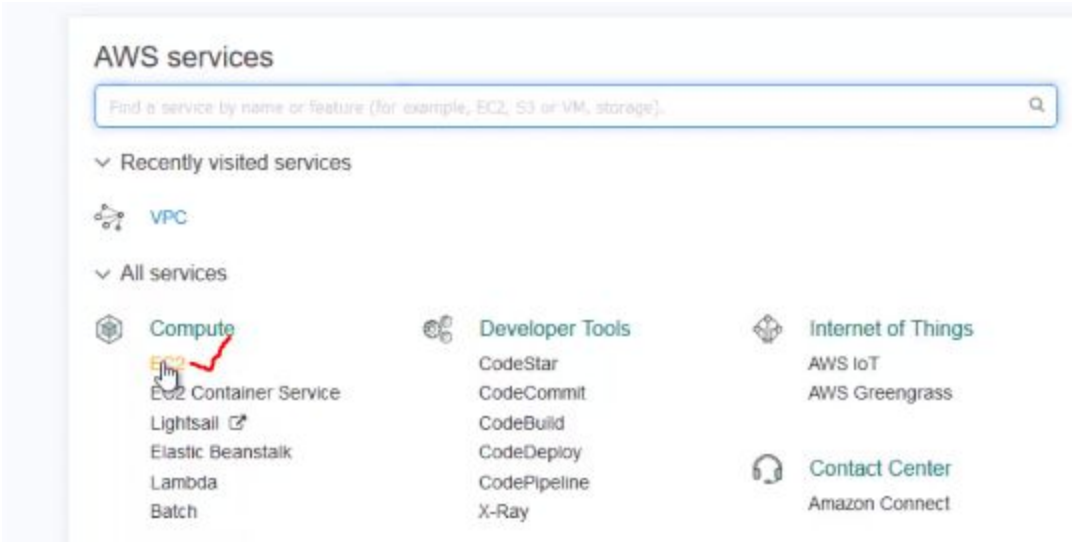


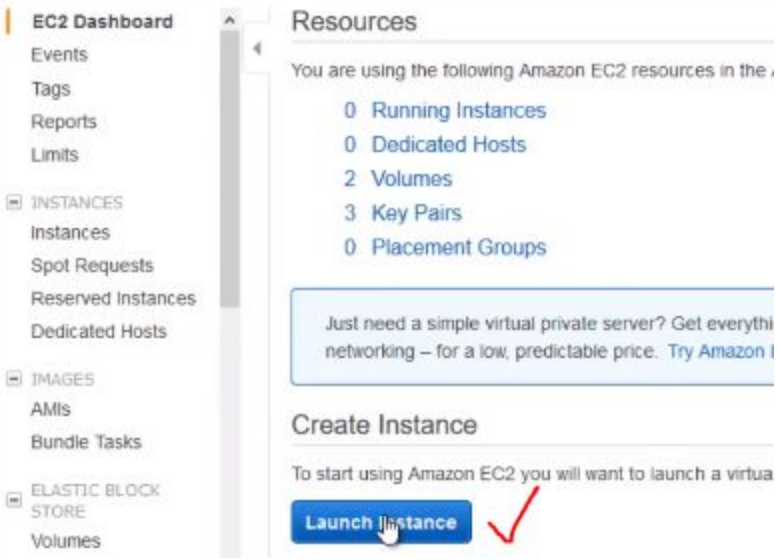
We can change that option

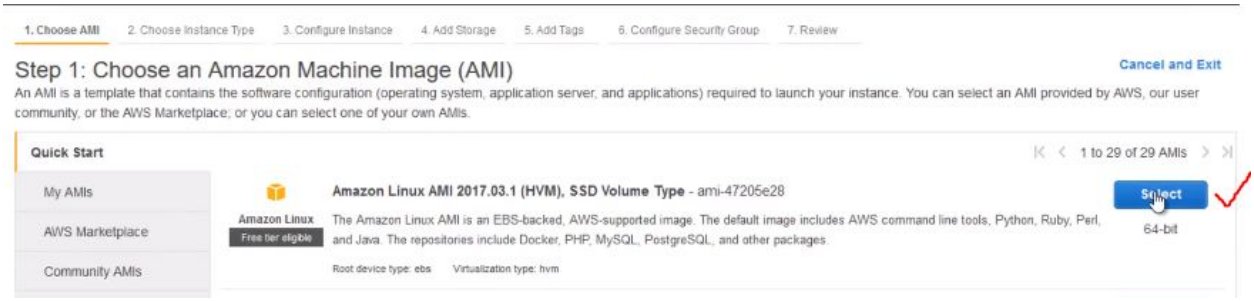

VPC Dashboard
Filter by VPC:
None
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Create Subnet
Subnet Actions
Delete Subnet
Edit IPv6 CIDRs
Create Flow Log
Modify auto-assign IP settings
IPv4
IPv6 CIDR
Route Table
Network ACL
Default Subnet
Auto-assign Public IP
rtb-3b49f252
acl-d542fabc
Yes
Yes
rtb-8d3d05e4 | My...
acl-f68bba9f
No
No
ap-south-1b
rtb-7f3c0416
acl-f68bba9f
No
No
ap-south-1b
rtb-3b49f252
acl-d542fabc
Yes
Yes

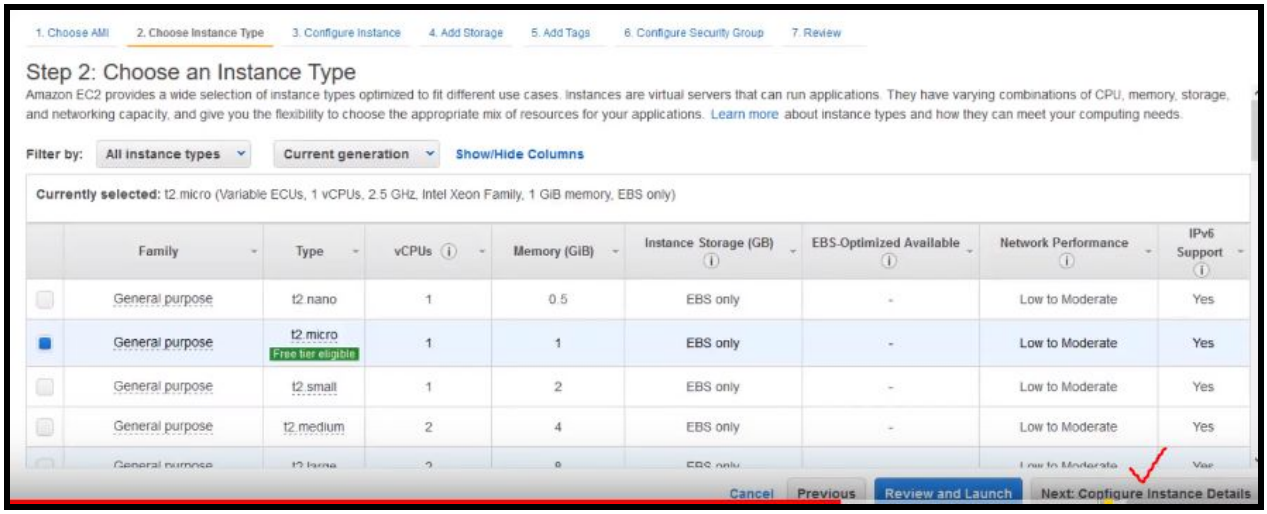

Create Subnet
Subnet Actions
Modify auto-assign IP settings
Enable auto-assign public IPv4 address and IPv6 addresses to automatically request an IP address for instances launched into this subnet.
Auto-assign IPs
Enable auto-assign public IPv4 address
Note: You can override the auto-assign IP settings for each individual instance at launch time for IPv4 or IPv6. Regardless of how you've configured the auto-assign public IP feature, you can assign a public IP address to an instance that has a single, new network interface with a device index of eth0.
Cancel
Save
Summary

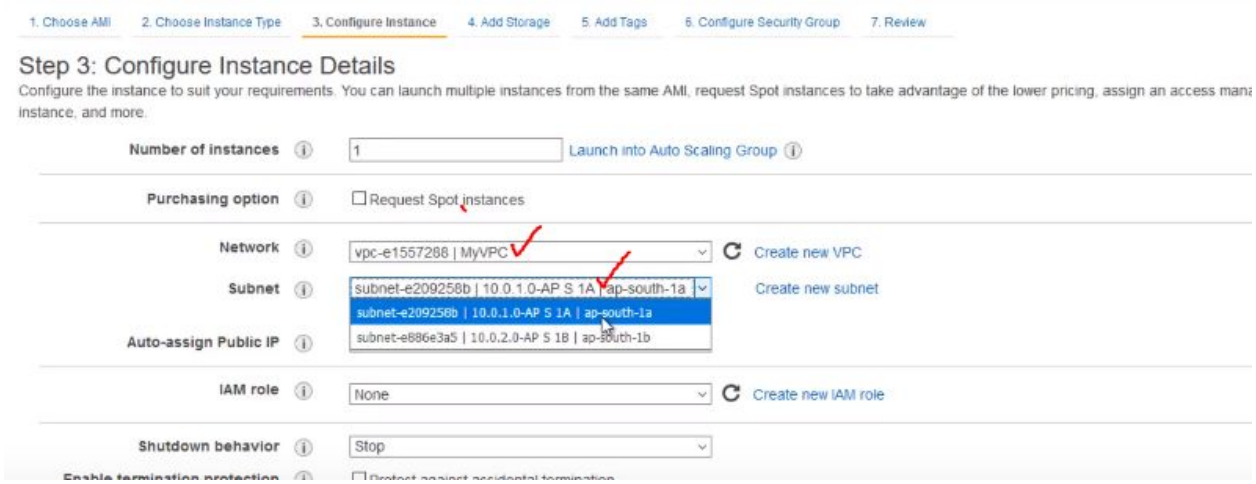
NOW Launch EC2 Instances



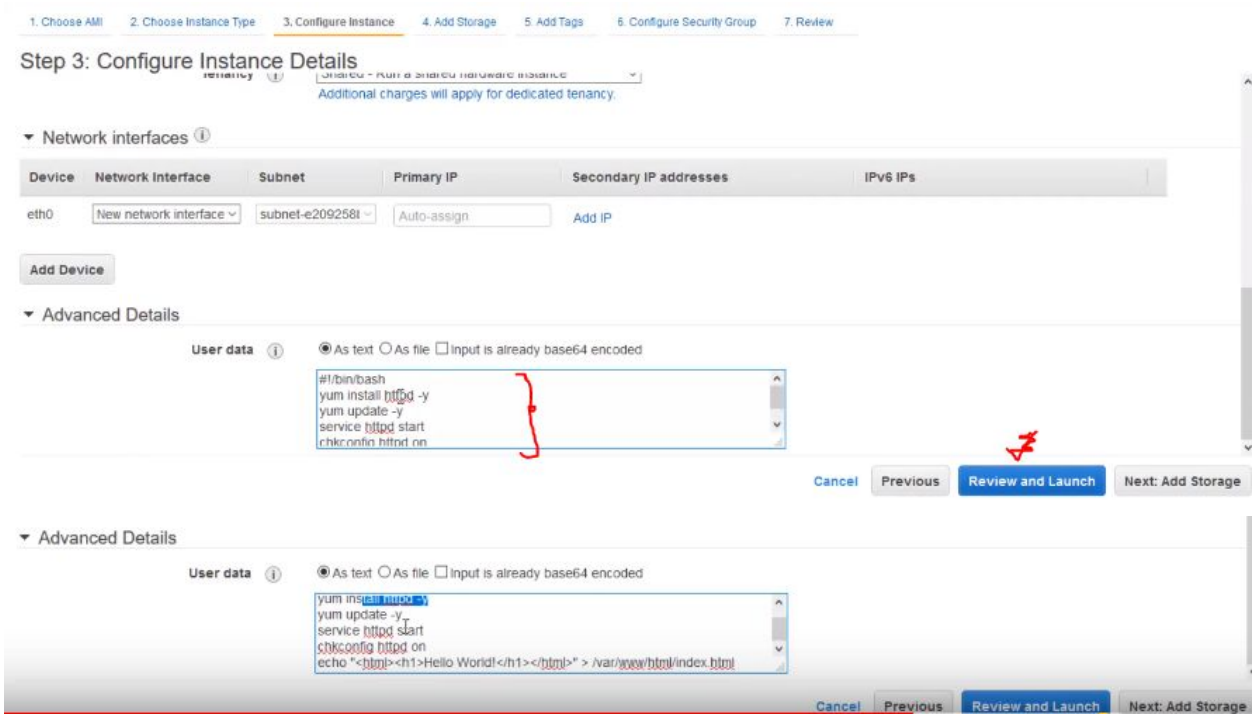




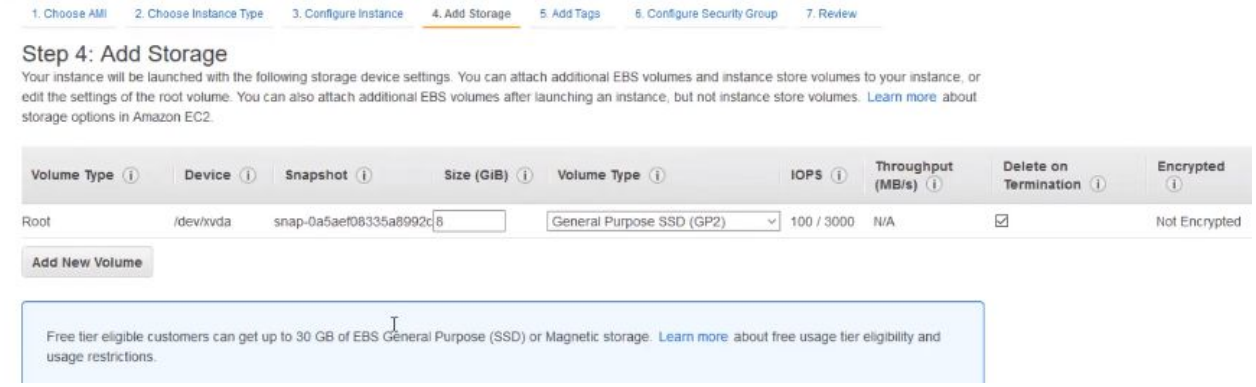


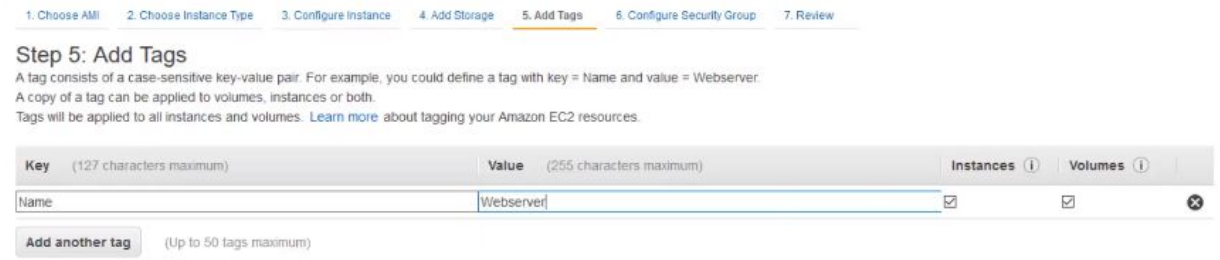


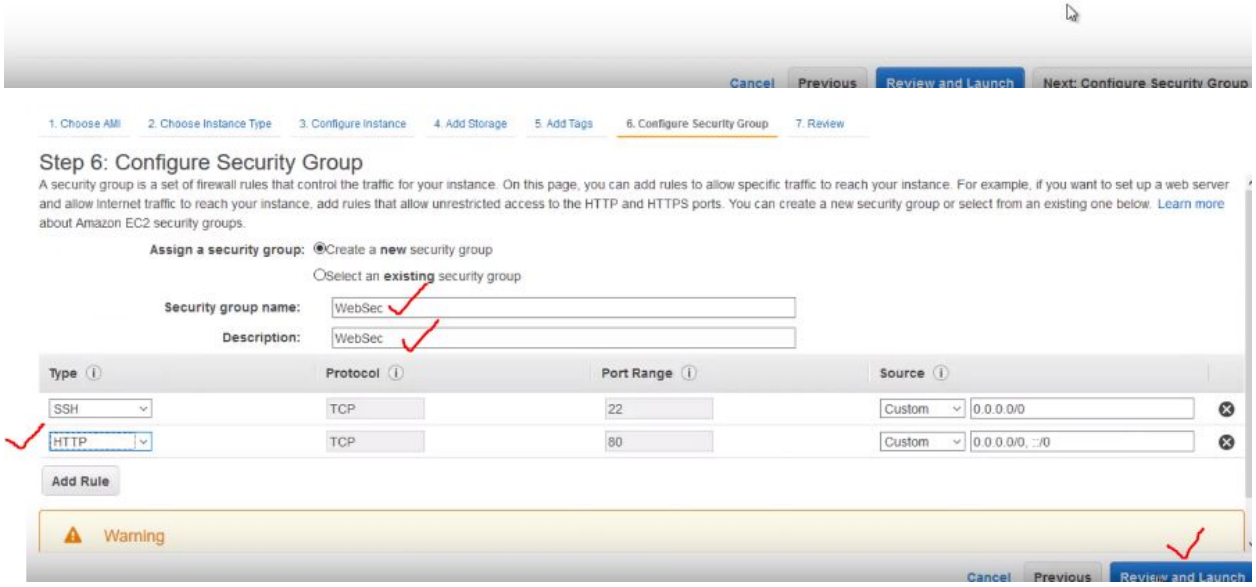
Select MyVPC and public subnet


1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
Additional charges will apply for dedicated tenancy.
Network interfaces
Device
Network Interface
Subnet
Primary IP
Secondary IP addresses
IPv6 IPs
eth0
New network interface
subnet-e209258t
Auto-assign
Add IP
Add Device
Advanced Details
User data
As text
As file
Input is already base64 encoded
#!/bin/bash
yum install httpd -y
yum update -y
service httpd start
chkconfig httpd on
Cancel
Previous
Review and Launch
Next: Add Storage
Advanced Details
User data
As text
As file
Input is already base64 encoded
yum install httpd -y
yum update -y
service httpd start
chkconfig httpd on
echo "<html><h1>Hello World!</h1></html>" > /var/www/html/index.html
Cancel
Previous
Review and Launch
Next: Add Storage

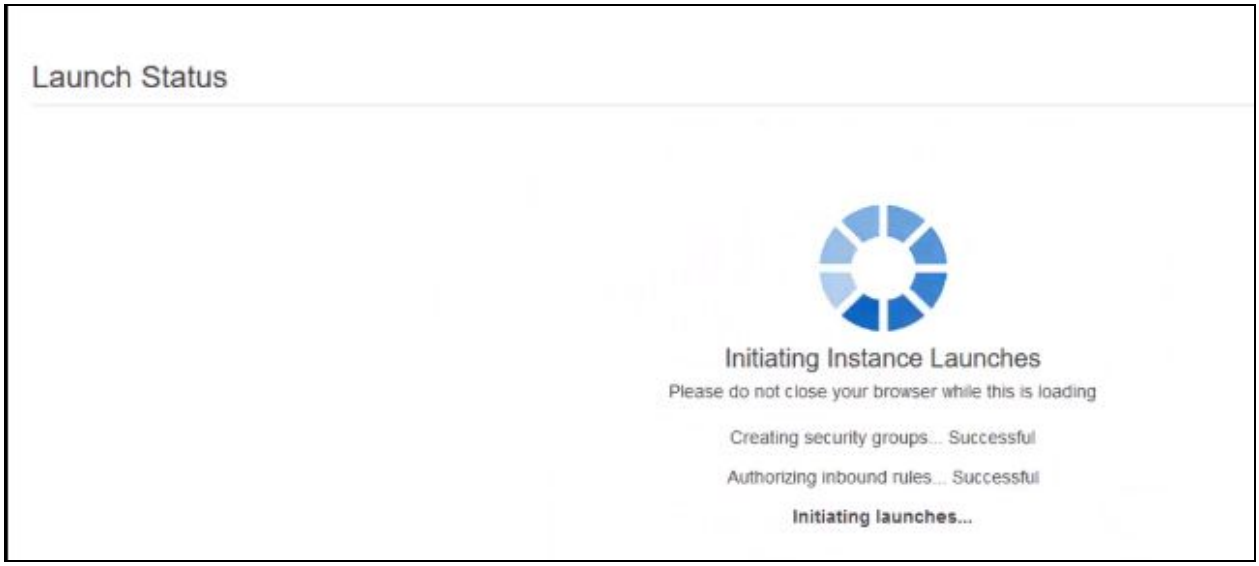
Add storage


1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 4: Add Storage
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.
Volume Type
Device
Snapshot
Size (GiB)
Volume Type
IOPS
Throughput (MB/s)
Delete on Termination
Encrypted
Root
/dev/xvda
snap-0a5aef08335a8992c
8
General Purpose SSD (GP2)
100 / 3000
N/A
Not Encrypted
Add New Volume
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

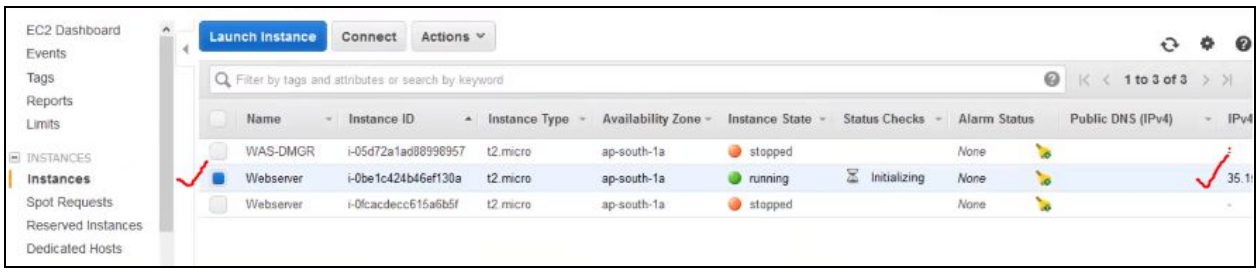

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 5: Add Tags
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.
Key (127 characters maximum)
Value (255 characters maximum)
Instances
Volumes
Name
Webserver
Add another tag
(Up to 50 tags maximum)


Cancel
Previous
Review and Launch
Next: Configure Security Group
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.
Assign a security group: Create a new security group
Select an existing security group
Security group name: WebSec
Description: WebSec
Type
Protocol
Port Range
Source
SSH
TCP
22
Custom
0.0.0.0/0
HTTP
TCP
80
Custom
0.0.0.0/0, ::/0
Add Rule
Warning
Cancel
Previous
Review and Launch

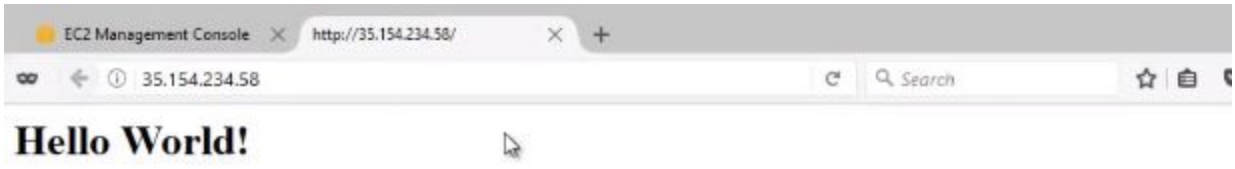

Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Create a new key pair
Key pair name
WebSec
Download Key Pair
You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.



Now check EC2 is ready



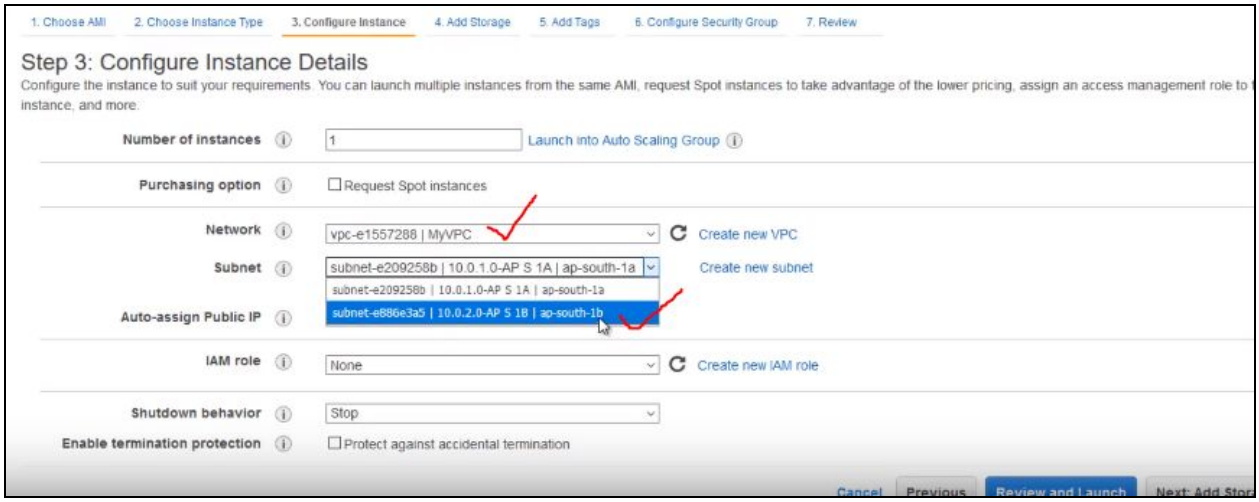
Now try to open in browser



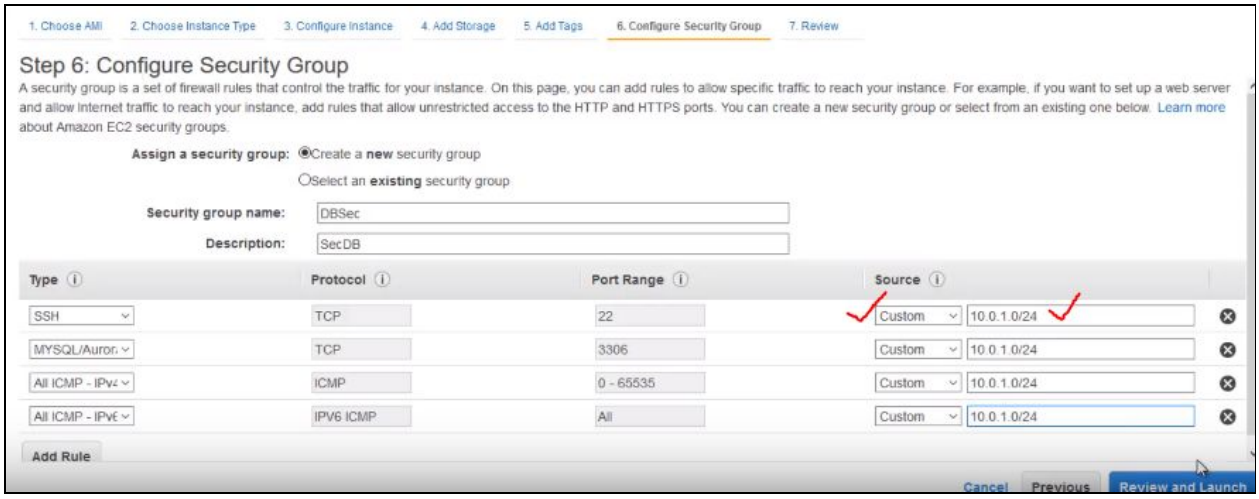
You will get the response as it is under public subnet
I.e from the internet, you can access ec2 instance which we place in public subnet

Launch EC2 Instance private subnet

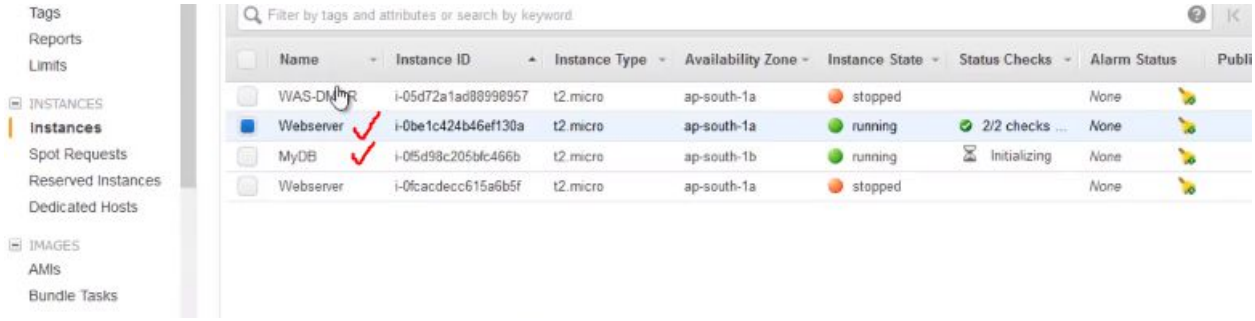
Eg: DB instances normally inside private subnet
Select the private subnet in configuration

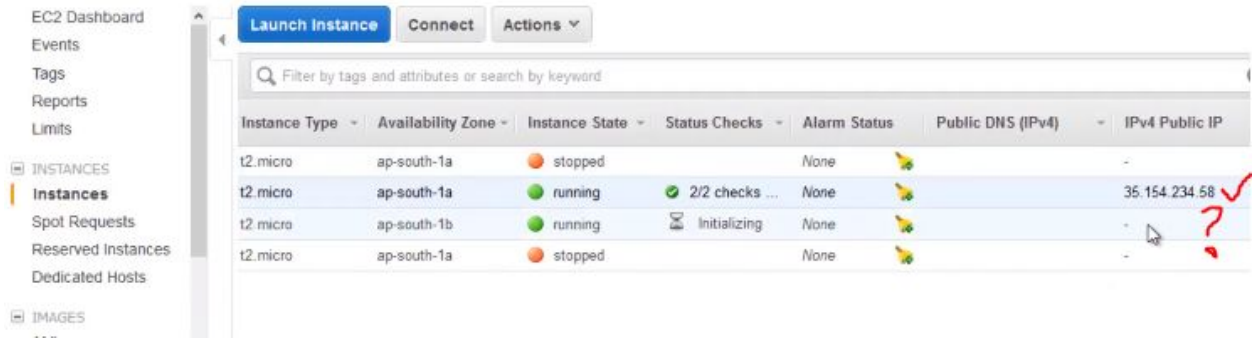


In security Group, we are locking down to only public subnet



Verify

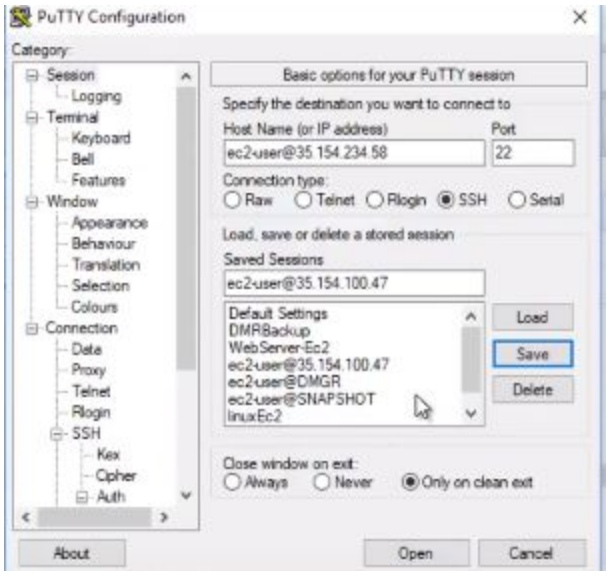




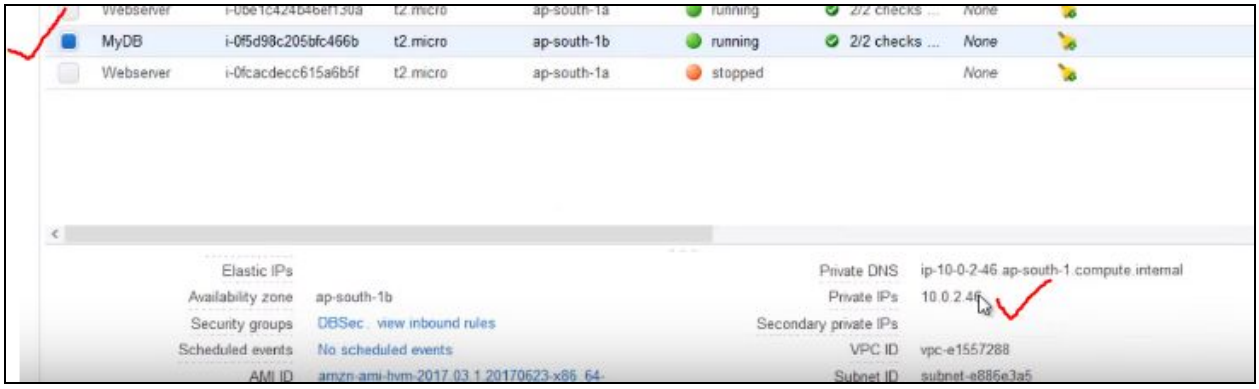
Db instance will not have any public IP address. As we did not give any public ip addres enable.

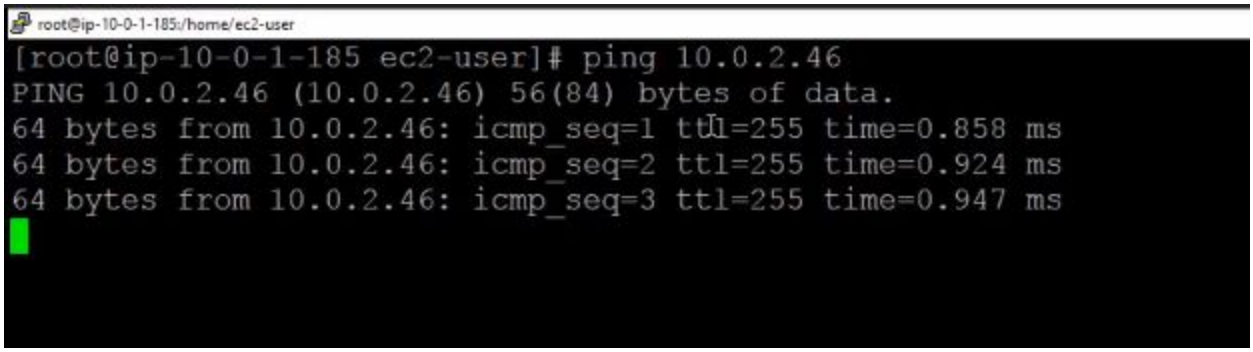
We can access instances in private subnet from instances in public subnet

So first login to EC2 in public



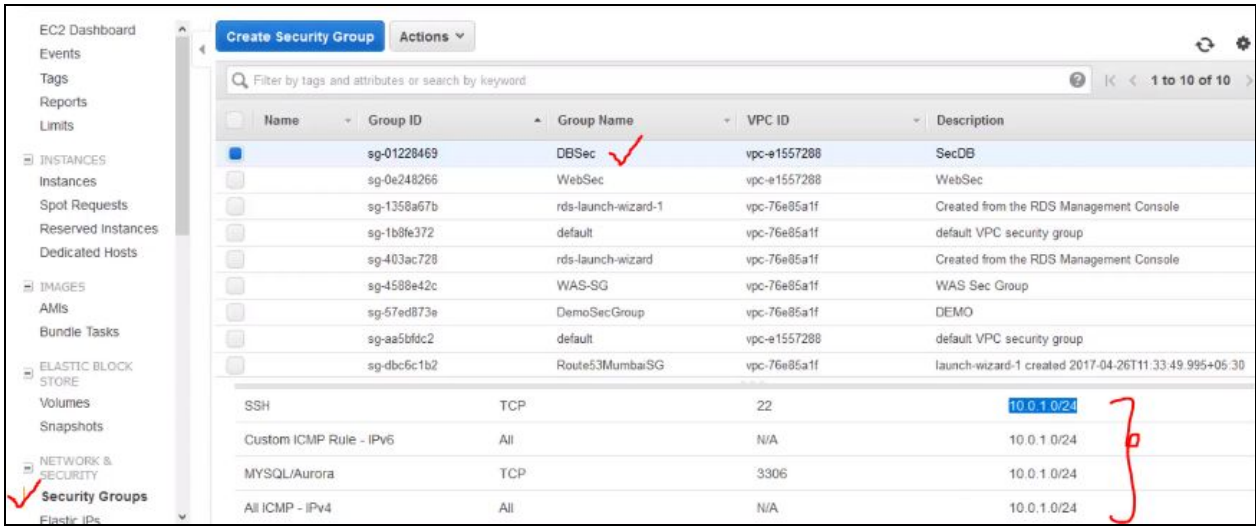
Now try to hit the DB instance ip address





Done, we are able to see the communication happen from public EC2 to private EC2 Application to DB

It happen because of Security group of DB instance (for troubleshooting purpose)



NAT Instances

- If you want to go from private servers to internet then you need to do translation from priv to pub
- Suppose for patching in private servers, you need to have access to internet
- You can use NAT instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound IPv4 traffic to the internet or other AWS services, but prevent the instances from receiving inbound traffic initiated by someone on the internet.
- So outbound traffic allowed - for patching or may be GIT repository connection from DB engine is the best use case for NAT instance

NAT Gateway

- Its a service offering from AWS and no need to maintain patching NAT instance, scaling up and on this things are not our headache
- We have to place the NAT gateway on public subnet

LAB:
Create a NAT gateway
Select a subnet (must be public subnet)
Create Elastic IP address
This completes the NAT gateway creation

Now goto RouteTable
Create a new private RT
---->privatename
---->vpc name selected
Now Edit RT
Already main table line is there , add new line(to tell where to get internet)
0.0.0.0/0 ngw
Because of this it will go to the internet
Ngw already know which subnet it has to take internet
Subnet willask routetable to go internet

So very imp now is make sure in subnet association , add private subnet
Now priv subnet will get internet

Now you will be able to ping internet google