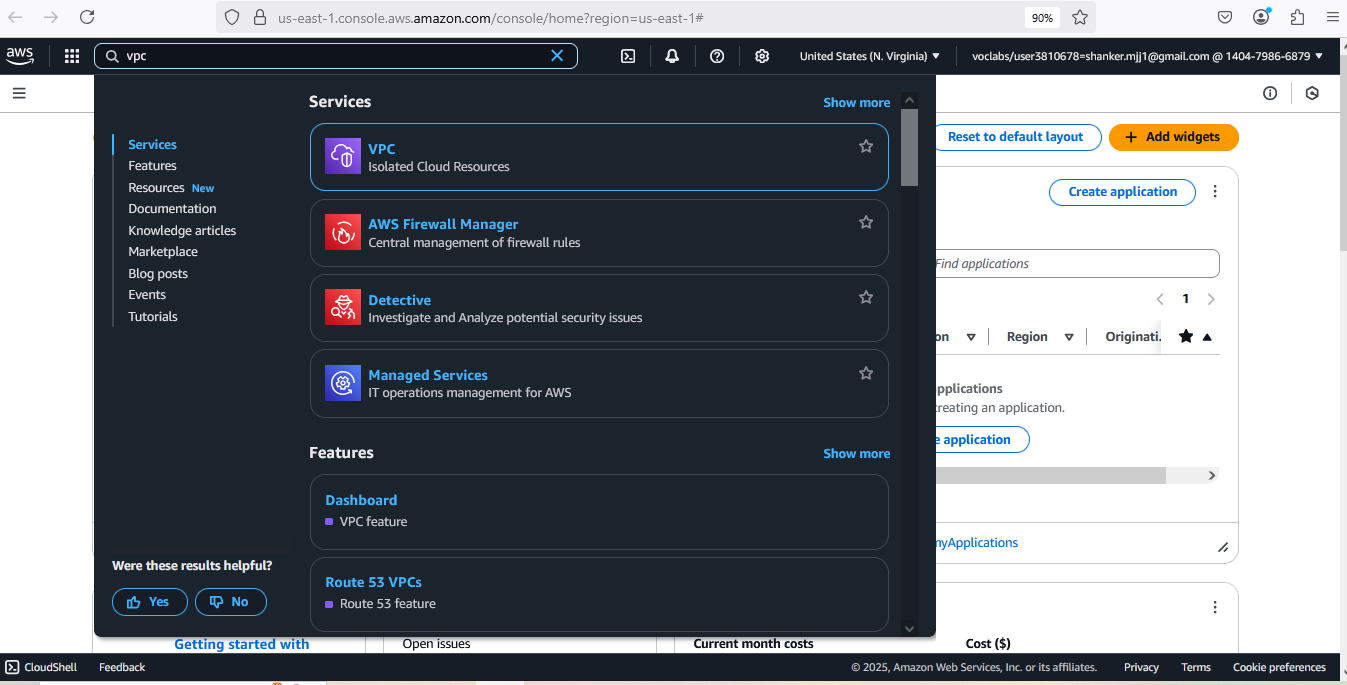
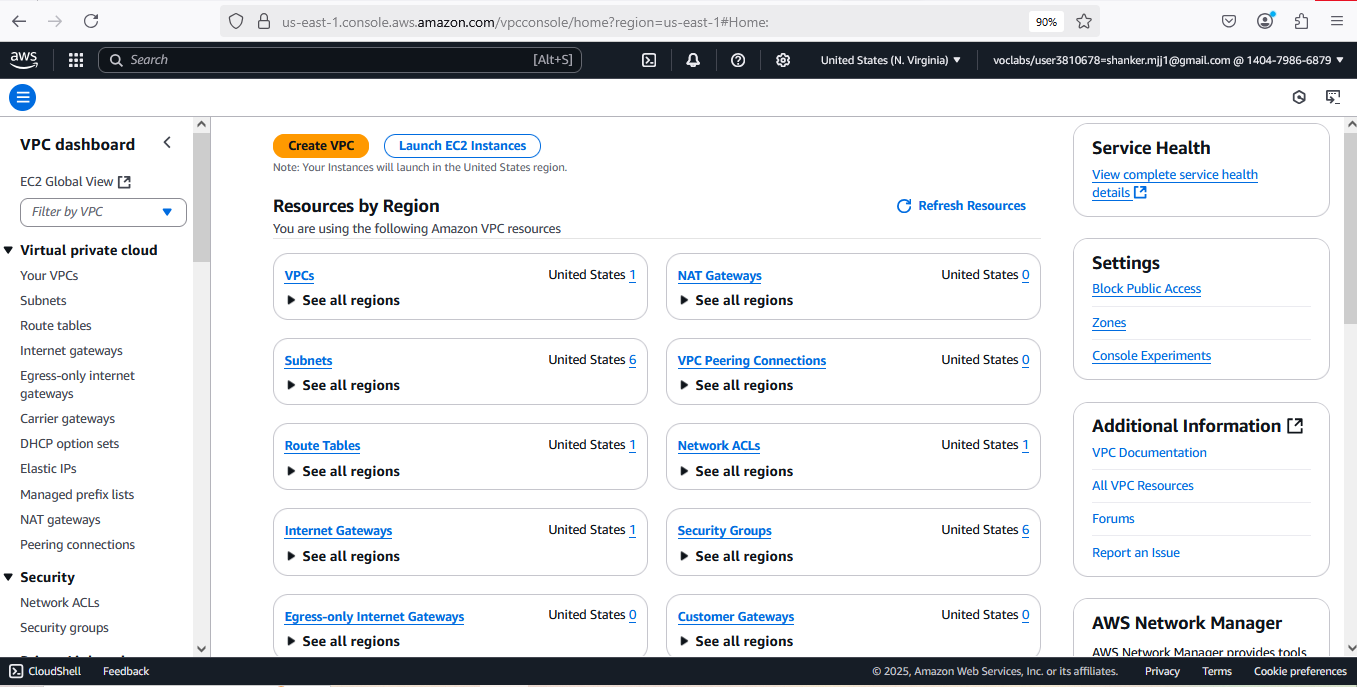
**Creating a Amazon VPC**

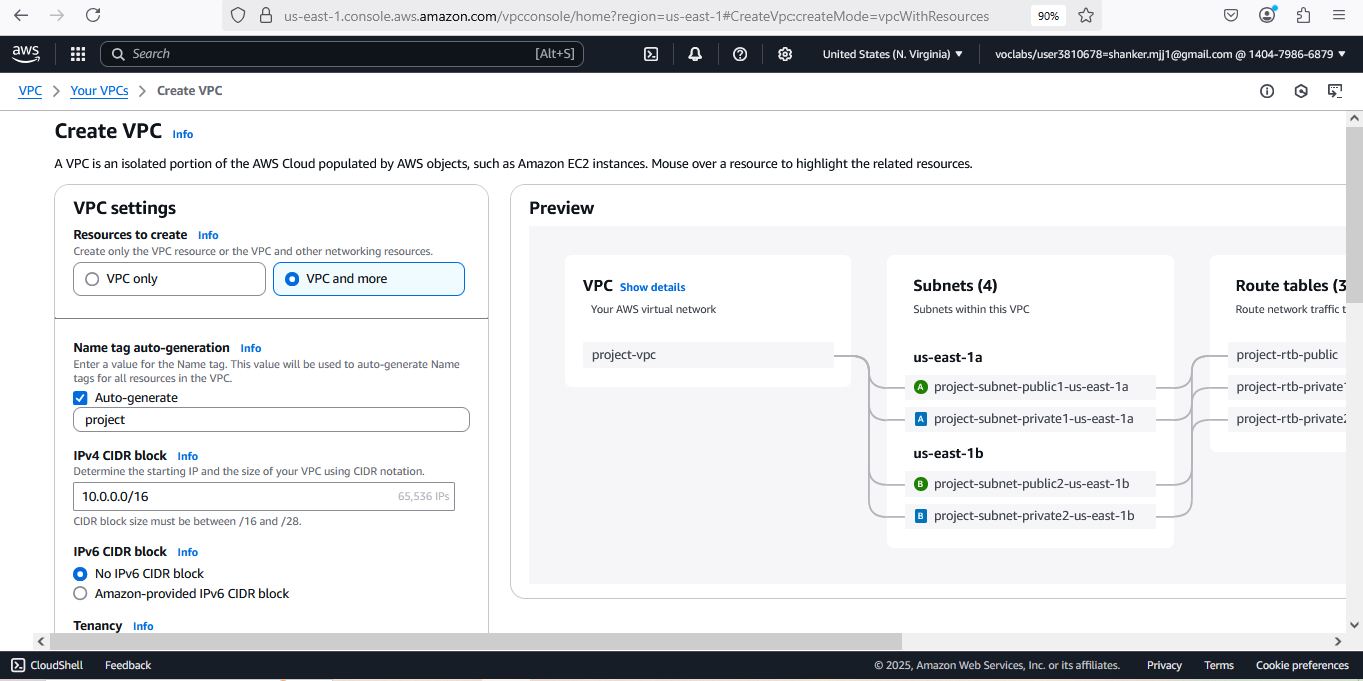
**Step 1:** Start Lab→ click AWS → Search bar (Search VPC) & select it



**Step 2:** click VPC, you will navigate below UI



**Step 3:** Click on create VPC



**Step 4:**

VPC Setting:

Resources to create: select Resources to create

Name tag *- optional* : Write ur Rollno-VPC

IPv4 CIDR block : IPv4 CIDR manual input

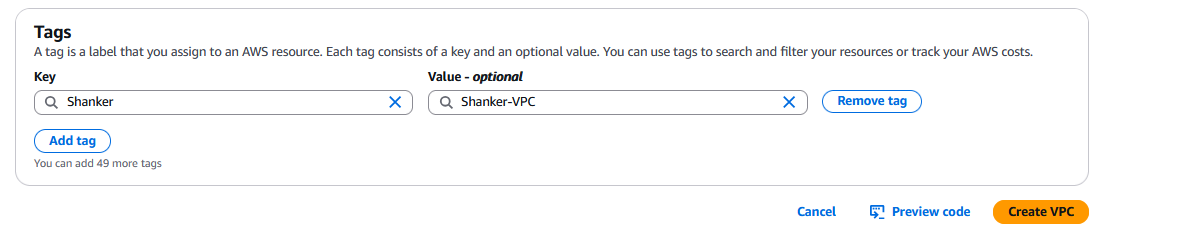
IPv4 CIDR: 12.0.0.0/16

IPv6 CIDR block: No IPv6 CIDR block

Tenancy: Default

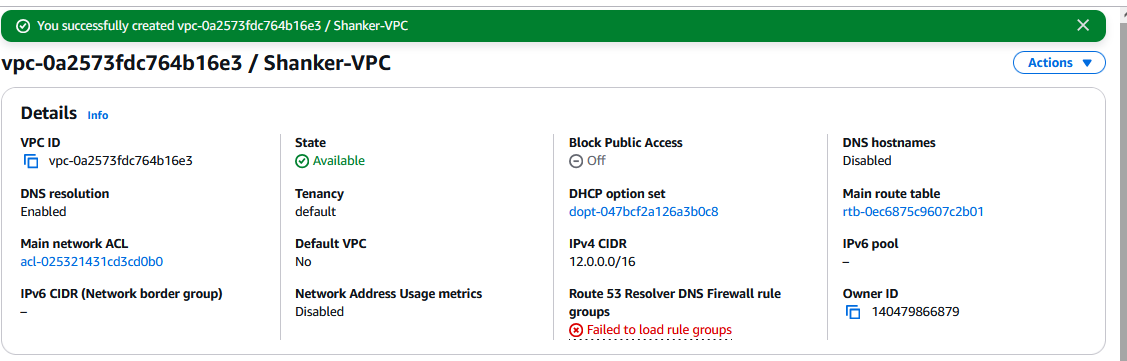
Tags: Key : Your Name Value- Rollno-VPC

Finally Click Create VPC



NOTE:

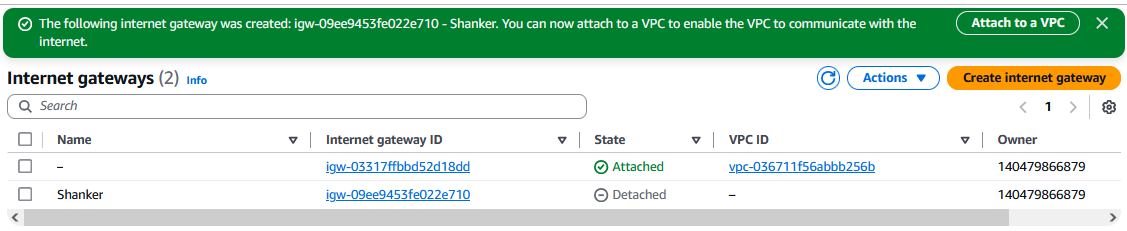
Subnets: A subnet is a smaller, segmented part of a larger network that isolates and organizes devices within a specific IP address range.



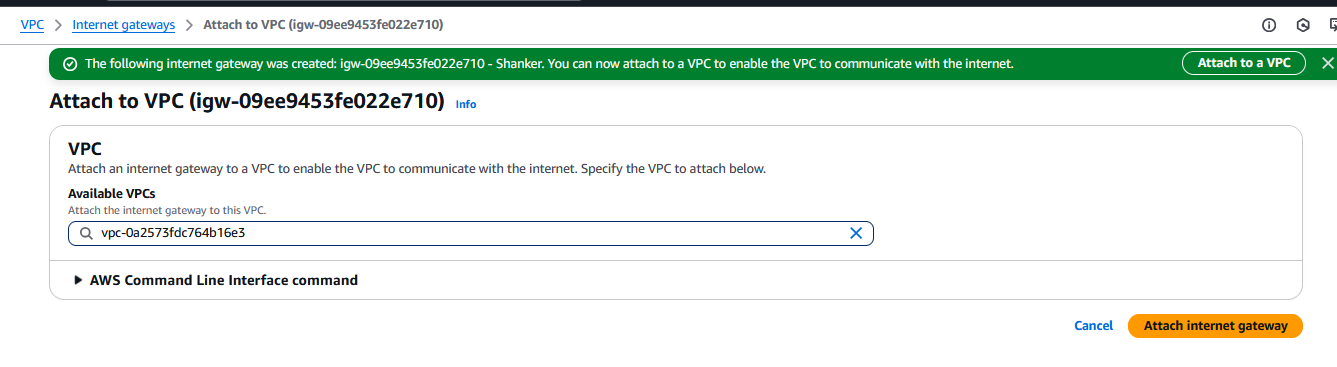
**Step 5:** **Creating Internet Gateways**

Click on [Internet gateways](https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#igws:) at left side pallet

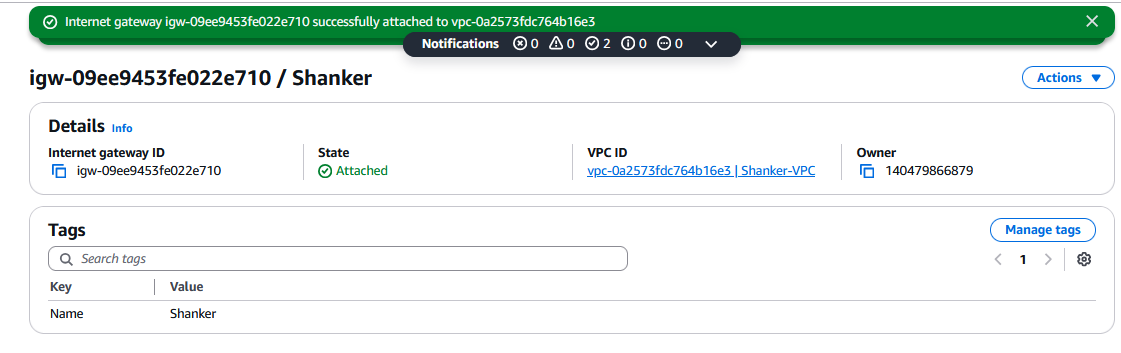
1. Click on create **Create internet gateway**
2. **Internet gateway settings:**
   1. **Name tag : Rollno-IG**
   2. **Tags - optional (Default)**

****

* 1. Select your **internet gateway** Which is created by you
  2. Click on Action → Attach to VPC → Select your VPC created above Steps



* 1. Click on Attach internet gateway
  2. Now you can see your attached successfully



**Step 6: Creating Subnets:**

Click on subnets(Left Panel):

1. Click on Create Subnets

Create subnet: Select your VPC created

Subnet settings:

**Subnet 1 of 1**

**I. Subnet name :** Rollno-public-SB1

II. Availability Zone : Select US(N.Virgina) -us-east 1a

III. IPv4 VPC CIDR block : 12.0.0.0/16

iV. IPv4 subnet CIDR block: 12.0.0.0/20

Keep all Default Tags

Click on Add new Subnet

**Subnet 2 of 2**

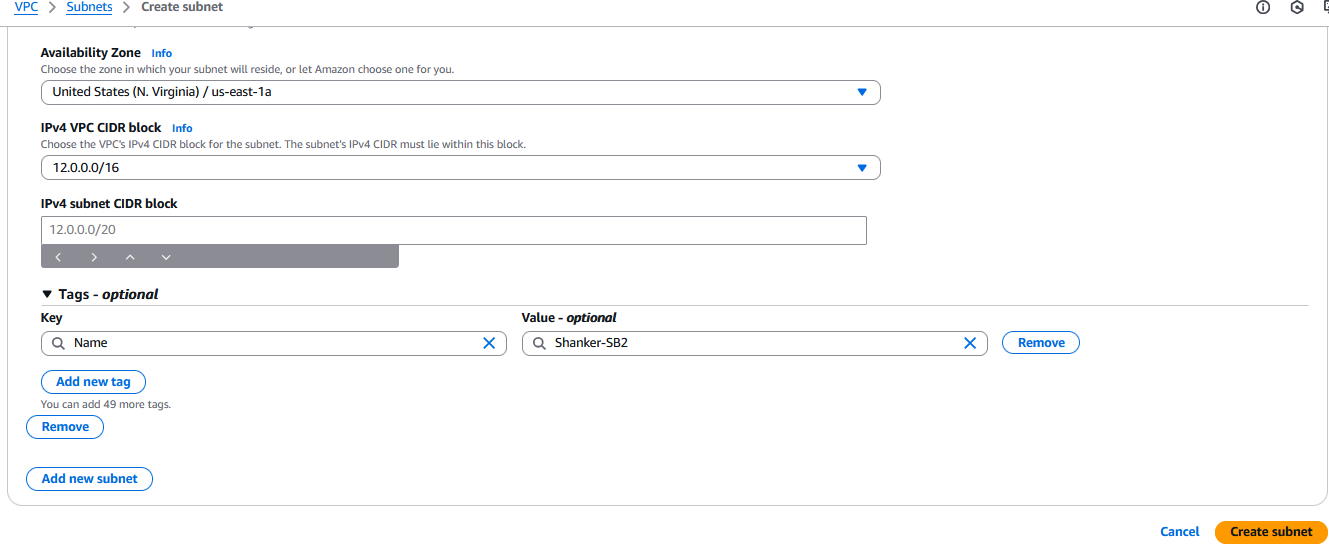
**I. Subnet name :** Rollno-private-SB2

II. Availability Zone : Select US(N.Virgina) -us-east 1a

III. IPv4 VPC CIDR block : 12.0.2.0/24

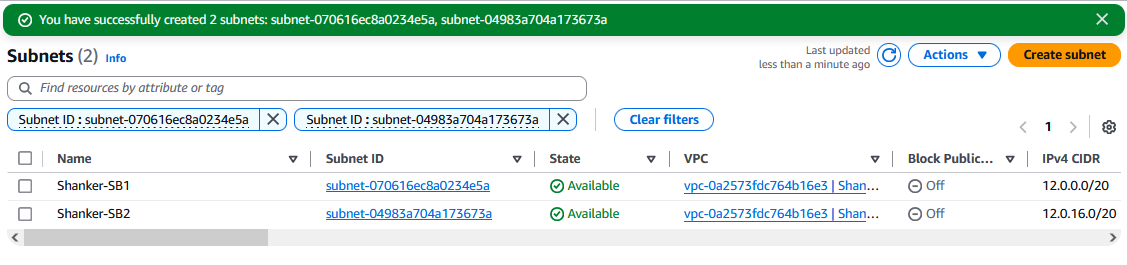
iV. IPv4 subnet CIDR block: 12.0.16.0/20

Finally Create a subnet



1. Or Else It creates defaults subnets by AWS (We can using it, Instead of new).

Finally you can see the Subnets:



**Step 7:** Creating a Route Table:

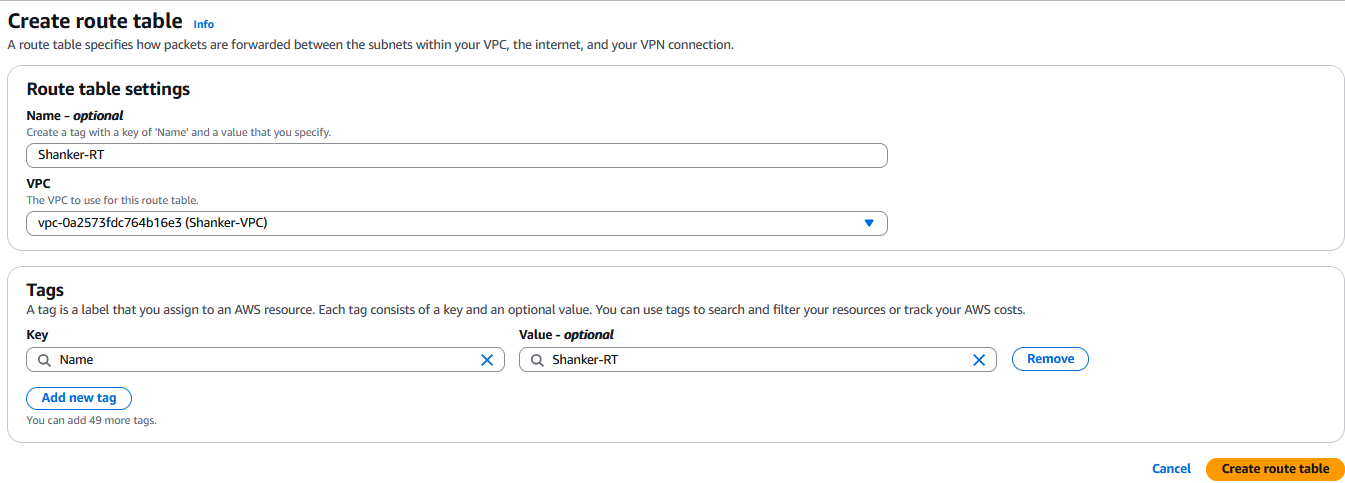
Click on [Route tables](https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTables:) left side of pallet

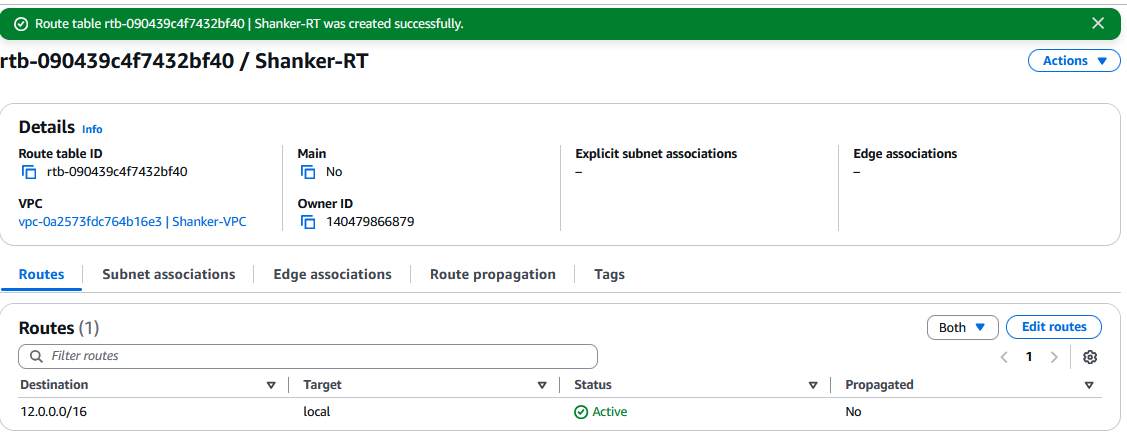
1. Click on Create route table

Route table settings :

* 1. Name: Rollno-Public-RT
  2. VPC: Select ur VPC created one
  3. Tags : Default

1. Finally click on Create route Table



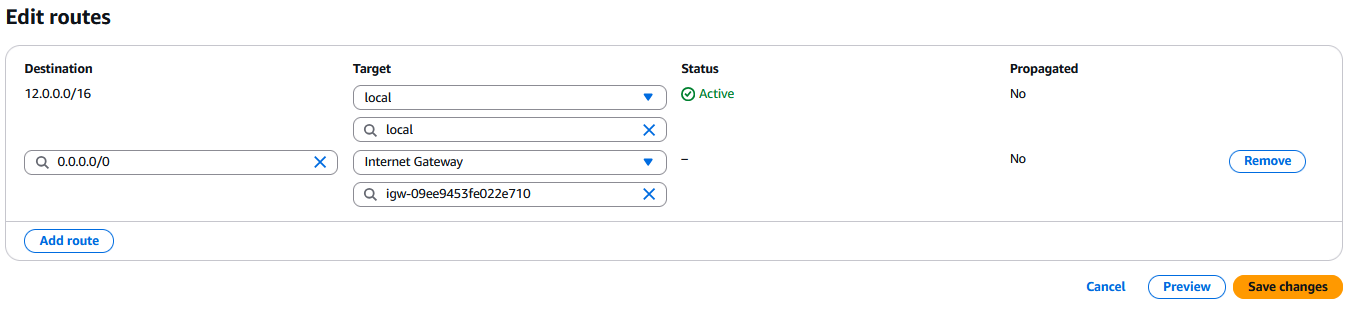


1. We need to provide Internet to Route

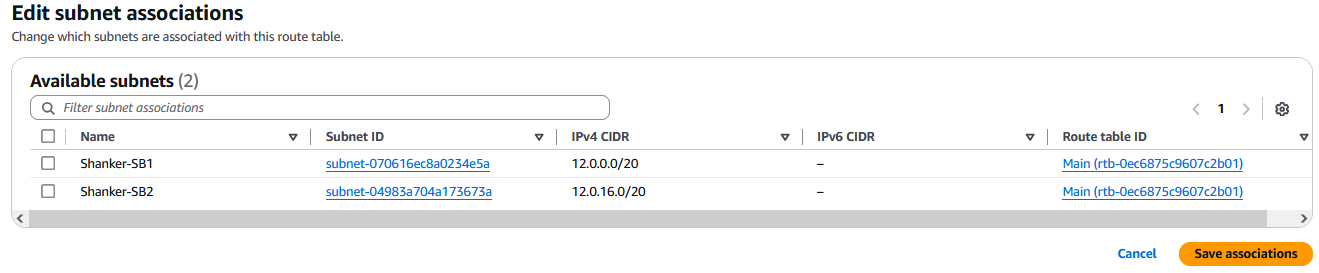
I. Click on edit Route

II. click Add Route

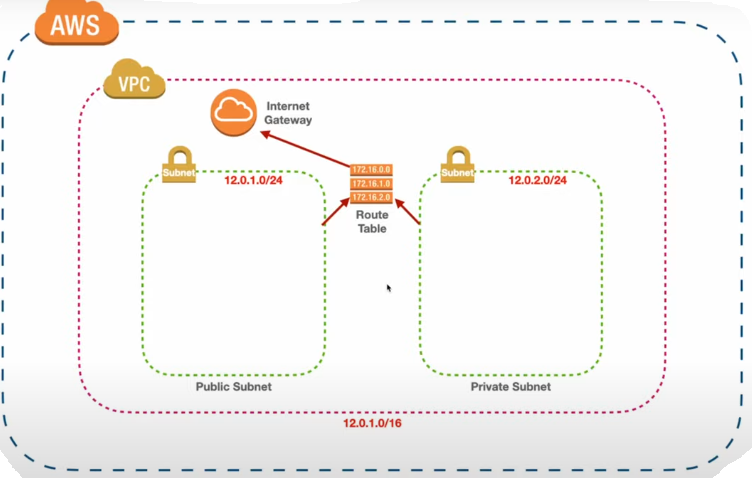
0.0.0.0/0 (Internet) Target (Internet Gateway)

****

1. **Finally save changes**
2. In [Route tables](https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTables:) click on Subnet Association → Click edit Subnet Association



1. Public RouteTable: To provide access to public route internet access SB-1 Select (Checkbox) & Save association.
2. Private RouteTable: To provide access to Private route internet access SB-2
   1. Goto RouteTable
   2. Click & create a route Table
   3. Name : Rollno-private-RT
   4. VPC: Select your VPC
   5. Click create Route Table
   6. Select your Rollno-private-RT, Goto → subnet association → Edit subnet association → select Rollno-Private-SB2→click save association



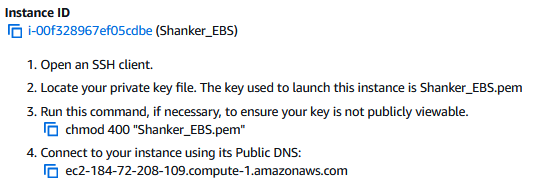
**Step 8:** Create a Resource inside the public subnet (EC2)

Navigate to EC2 → click on Instances

1. Create a Instance
   1. **Name**: EC1\_Rollno\_Public\_subnet-Instance
   2. **Select** : Ubuntu OS
   3. **AMI**: Free Tier Eligible, 64 Bit,
   4. **Instance type**: t2.micro
   5. **KeyPair**: PPK File(Already Exist\_One)
   6. **Network Settings**: Edit (Previous VPC Setup as to do Here..)
      1. VPC: Select which was created Earlier (Rollno-VPC)
      2. Subnet : Choose public subnet (Rollno-public-SB1)
      3. Auto-Assign public IP: Enable (To assign public IP)
      4. Firewall (Security Group) :
         1. Security Group Name: Public-Rollno-ec2-SG
         2. Description: Same
         3. Type: ssh, Protocol : TCP, PORT :22,
         4. Source Type: Anywhere
   7. **Configuration Storage**: Default
   8. **Advance Details:** Default
2. Launch the Instance
3. Copy the Public Address

**Step 9:** Connect with SSH Client with Putty to connect public subnet using **Bastion Host** (Go to Powershell & Connect through Putty with using PPK & URL)

1. You need to follow all the steps (1 to 4) in your Terminal
2. Here point 2 (Need to Identify the Private Key)



1. Once connected/logged through putty you can check that we are connected with Public Address subnet (12.0.0.0)

**Step 10:** Connect Private subnet with **Bastion Host**

1. Create a Instance
   1. Name: EC2\_Rollno\_Private\_subnet-Instance
   2. Select : Ubuntu OS
   3. AMI: Free Tier Eligible, 64 Bit,
   4. Instance type: t2.micro
   5. KeyPair: PPK File(Already Exist\_One)
   6. Network Settings: Edit (Previous VPC Setup as to do Here..)
      1. VPC: Select which was created Earlier (Rollno-VPC)
      2. Subnet : Choose public subnet (Rollno-**private**-SB1)
      3. Auto-Assign public IP: Disable (No need public IP)
      4. Firewall (Security Group) :
         1. Security Group Name: Private-Rollno-ec2-SG
         2. Description: Same as Private-Rollno-ec2-SG
         3. Type: ssh, Protocol : TCP, PORT :22,
         4. Source Type: Custom: Source: 12.0.2.0/24 (Private IP)
   7. Configuration Storage: Default
   8. Advance Details: Default
2. Launch the Instance

**Step 11:** Connect Public Subnet (Bastion Host) to Private subnet

1. Goto Ec2 Instance (EC1\_Rollno\_Public\_subnet-Instance)
2. Already we are connected through putty in step 9
3. $ ls -lart
4. Create a File called aws\_privatekey\_ec2
5. Copy the Private Key of your PPK file Generated (Secret Key)
6. Paste in aws\_privatekey\_ec2.pem & save
7. chmod 400 aws\_privatekey\_ec2
8. Go to Private Instance (EC2\_Rollno\_Private\_subnet-Instance) **Copy the Private address** do not have public address
9. Connecting public with Private
   1. $ ssh -i “aws\_privatekey\_ec2” ubuntu@**Private IP address**
   2. $ yes
10. Finally you can see that you can enter into private EC2 with the help of **Bastion Host**

**Final Agenda**

