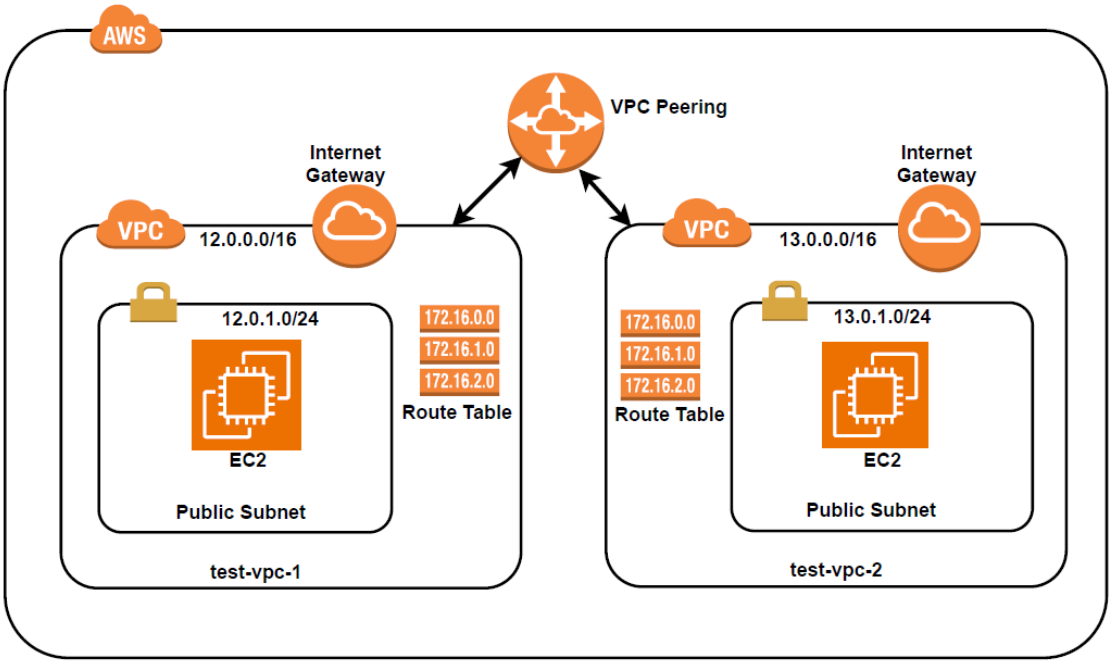
AWS VPC PEERING

- **VPC Peering** is a networking connection between two Virtual Private Clouds (VPCs) that allows you to route traffic between them using private IP addresses.
- Instances in either VPC can communicate with each other as if they are within the same network.
- VPC peering allows for seamless communication across different VPCs, whether they are in the same account, different accounts, or even different regions (inter-region VPC peering).

You can create a VPC peering connection between:

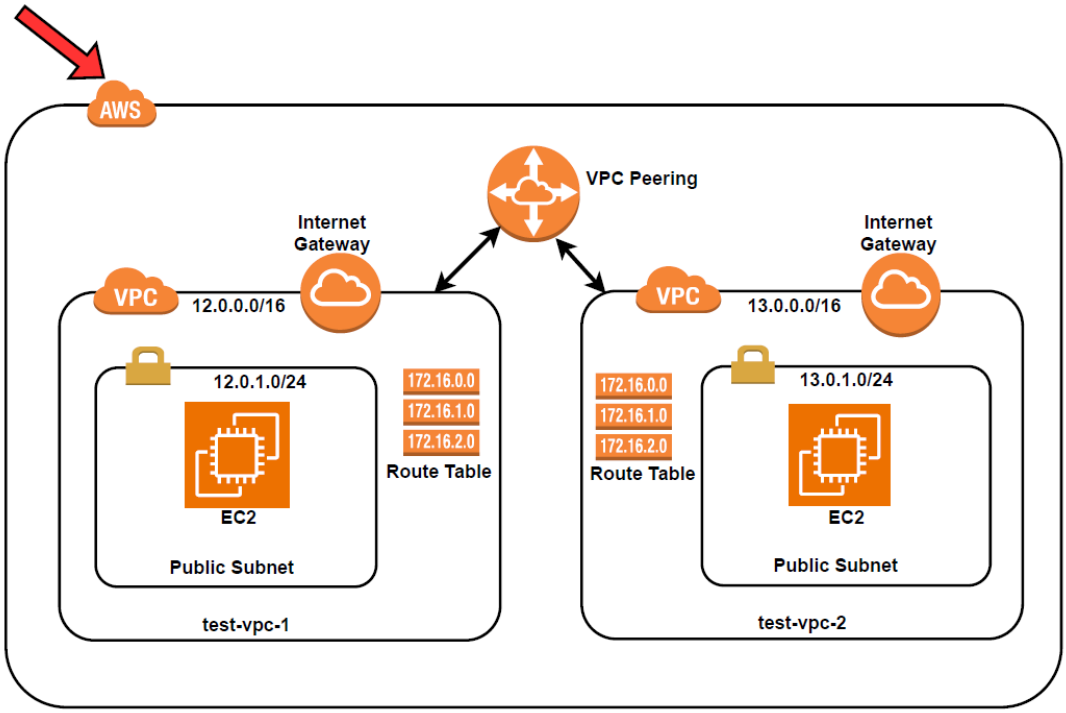
- Your own VPCs
- A VPC in another AWS Account
- A VPC in different AWS Region

In this tutorial, we will walk through the entire process of setting up a VPC peering connection between two VPCs: **test-vpc-1** and **test-vpc-2** that are in the same AWS Account.

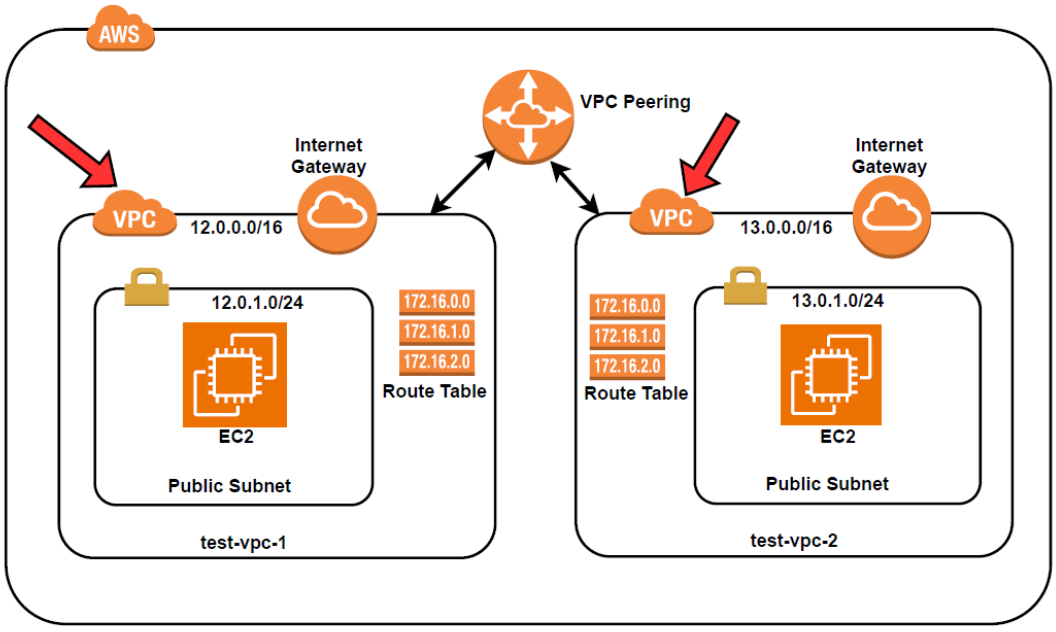


Part 1: Introduction

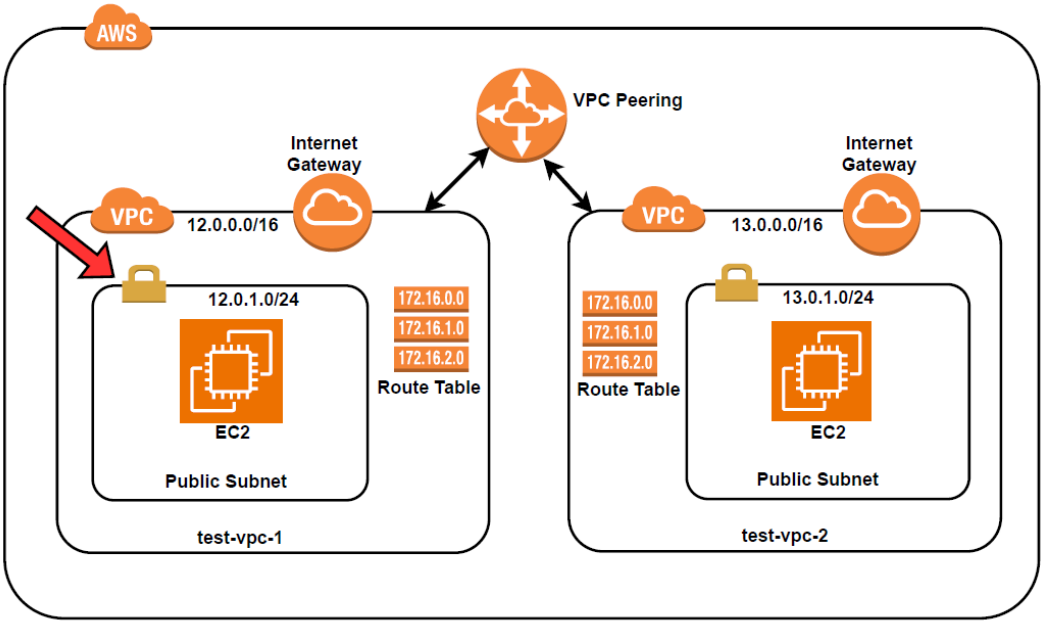
By the end of this session, you will know how to set up your VPC peering between the two VPCs and also you will know the benefits of using VPC peering. Let us start with it.



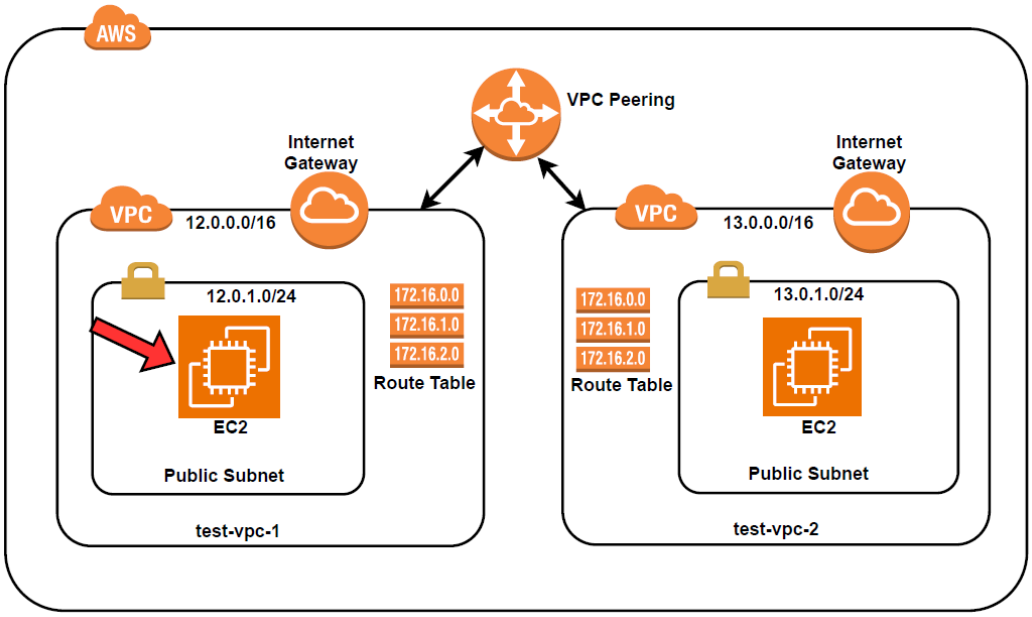
First of all, this is our AWS account on which we are going to set up this whole demo



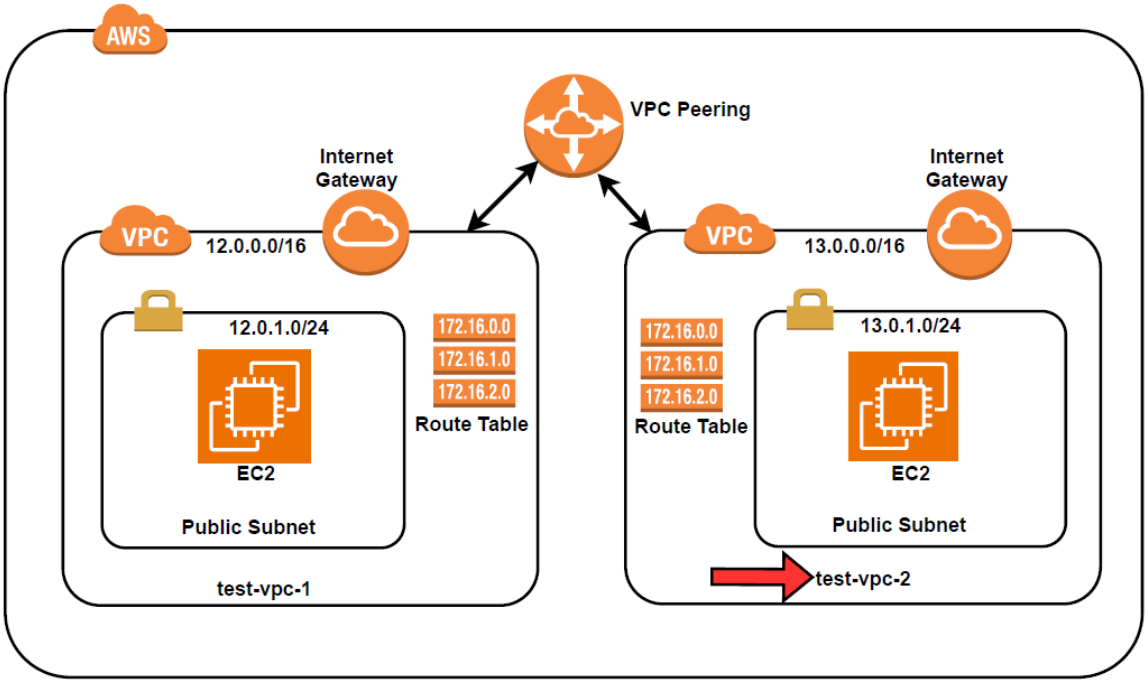
We are going to create two VPCs, on the left-hand side we have **test-vpc-1** and on the right-hand side you will find **test-vpc-2**. These two VPCs are separated from each other.



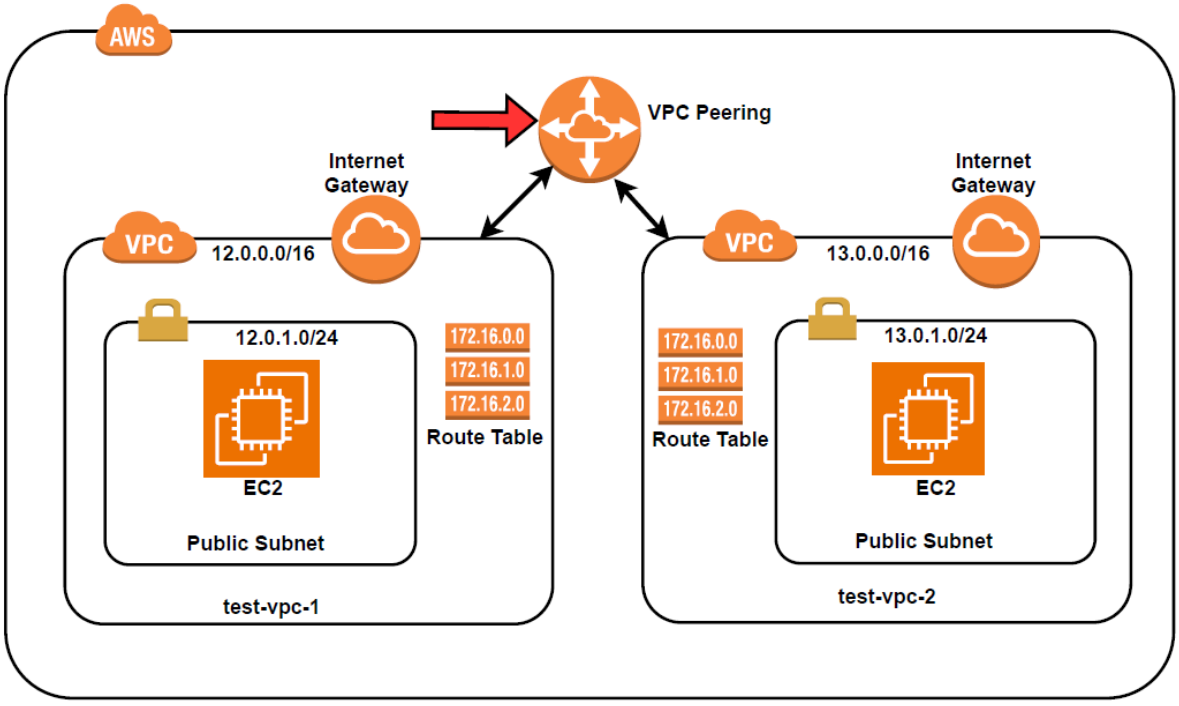
Inside the VPC, we are going to create a **subnet**.



Inside the subnet, we are going to provision **Amazon Linux EC2 instance**.



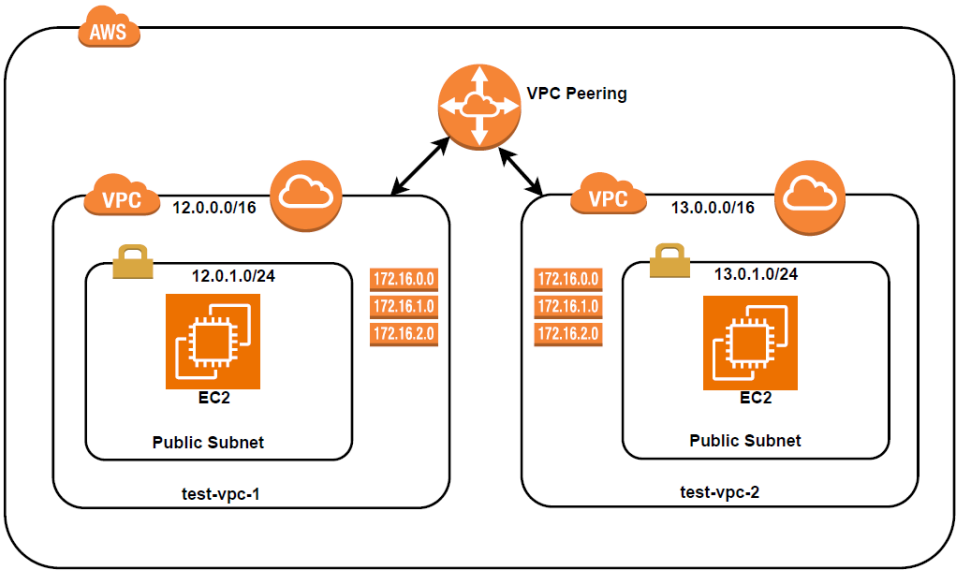
Similarly, on the other VPC (**test-vpc-2**), we are also going to create our subnet and EC2 instance.



After that we are going to set up a **VPC peering**. Once we set up the VPC peering, then the EC2 instance from **test-vpc-1** will be able to communicate with the EC2 instance in **test-vpc-2**.

STEPS

Let us start creating our AWS resources. We will follow these steps:



Step 1: Create the AWS VPC
Name: test-vpc-1 in US East (N. Virginia) us-east-1
CIDR: 12.0.0.0/16

Step 2: Create a public subnet with
CIDR 12.0.1.0/24
Name: test-subnet-vpc-1-1a

Step 3: Create an internet gateway and attach to test-vpc-1
Name: test-igw-vpc-1

Step 4: Create a route table and name it test-rt-vpc-1 and attach the public subnet to this Route Table.
Name: test-rt-vpc-1
Subnet Name: test-subnet-vpc-1-1a

Step 5: Create the AWS VPC
Name: test-vpc-2 in US East (N. Virginia) us-east-1
CIDR: 13.0.0.0/16

Step 6: Create a public subnet with
CIDR 13.0.1.0/24
Name: test-subnet-vpc-2-1a

Step 7: Create an internet gateway and attach to test-vpc-2
Name: test-igw-vpc-2

Step 8: Create a route table and name it test-rt-vpc-2 and attach the public subnet to this RouteTable.
`Name: test-rt-vpc-2`
`Subnet Name: test-subnet-vpc-2-1a`
Step 9: Create the VPC peering
`Name: peering-connection-between-vpc1-and-vpc2`

Step 10: Accept the Peering connection in both VPCs

Step 11: Update the route tables in both VPCs to allow traffic to the other VPC

- In test-rt-vpc-1 route table, add a route for test-vpc-2's CIDR block to the peering-connection-between-vpc1-and-vpc2 connection.
  `test-vpc-2 CIDR: 13.0.0.0/16`
- In test-rt-vpc-2 route table, add a route for test-vpc-1's CIDR block to the peering-connection-between-vpc1-and-vpc2 connection
  `test-vpc-1 CIDR: 12.0.0.0/16`

Step 12: Launch an EC2 instance in both VPC's public subnets
Launch two Ubuntu EC2 Instances
`Name First EC2: vpc-1-ec2-instance`
`Name Second EC2: vpc-2-ec2-instance`

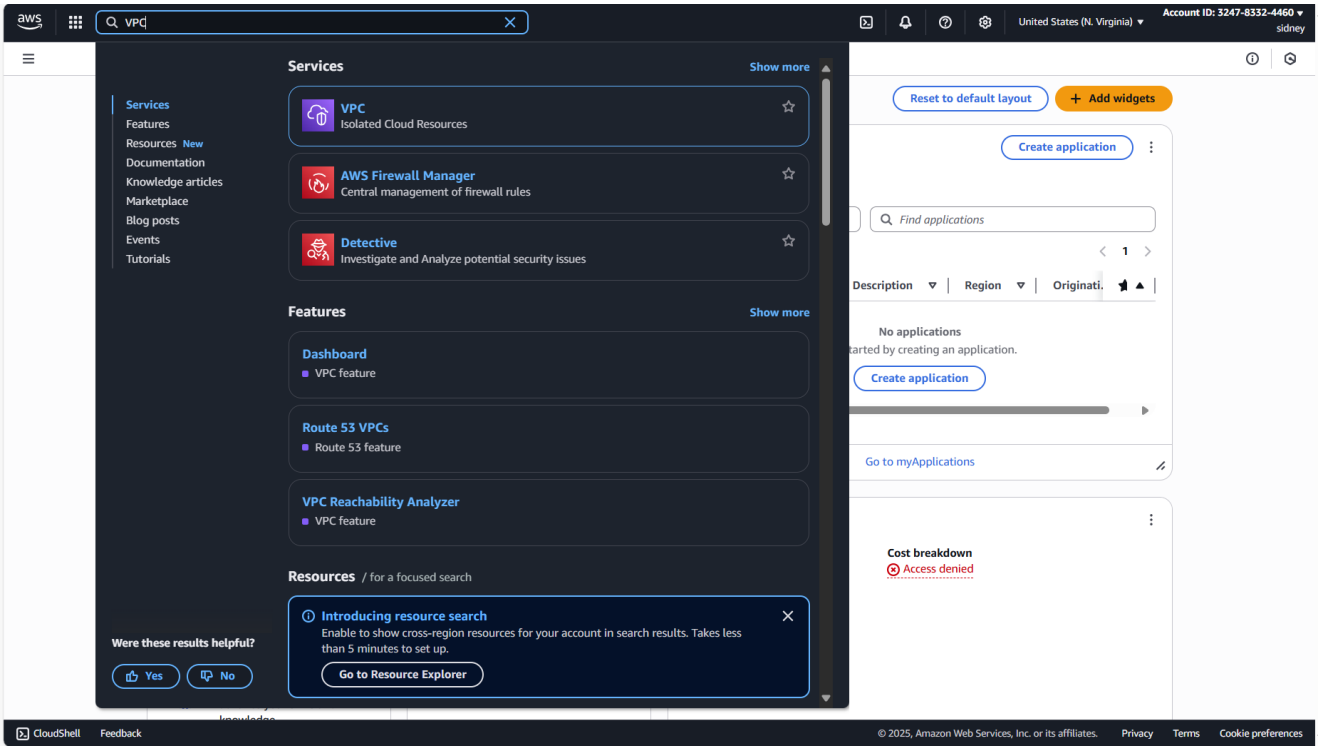
SSH connect to the EC2 instances

Step 13: Test the connectivity between the instances using the curl command.
You have successfully set up VPC peering between two VPCs and tested connectivity between instances in the public subnets of each VPC.

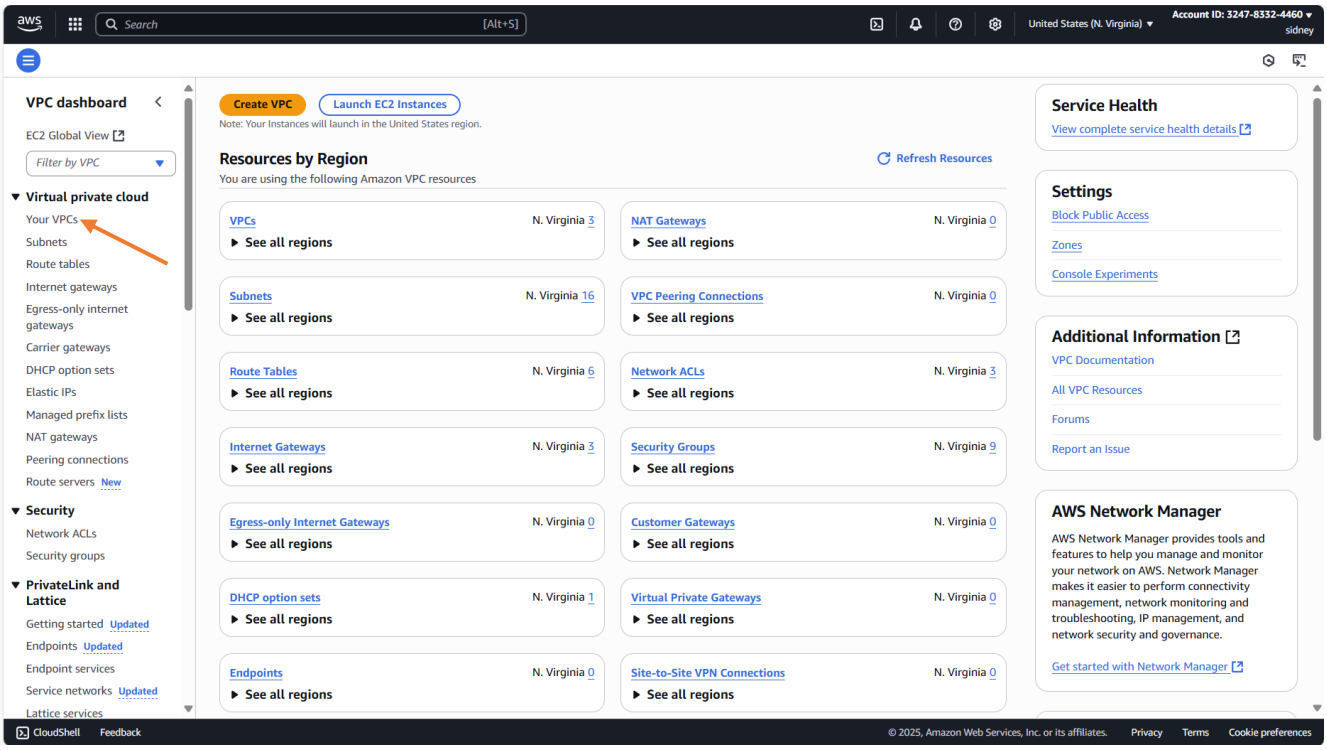
IMPLEMENTATION

Part 2: Creating test-vpc-1

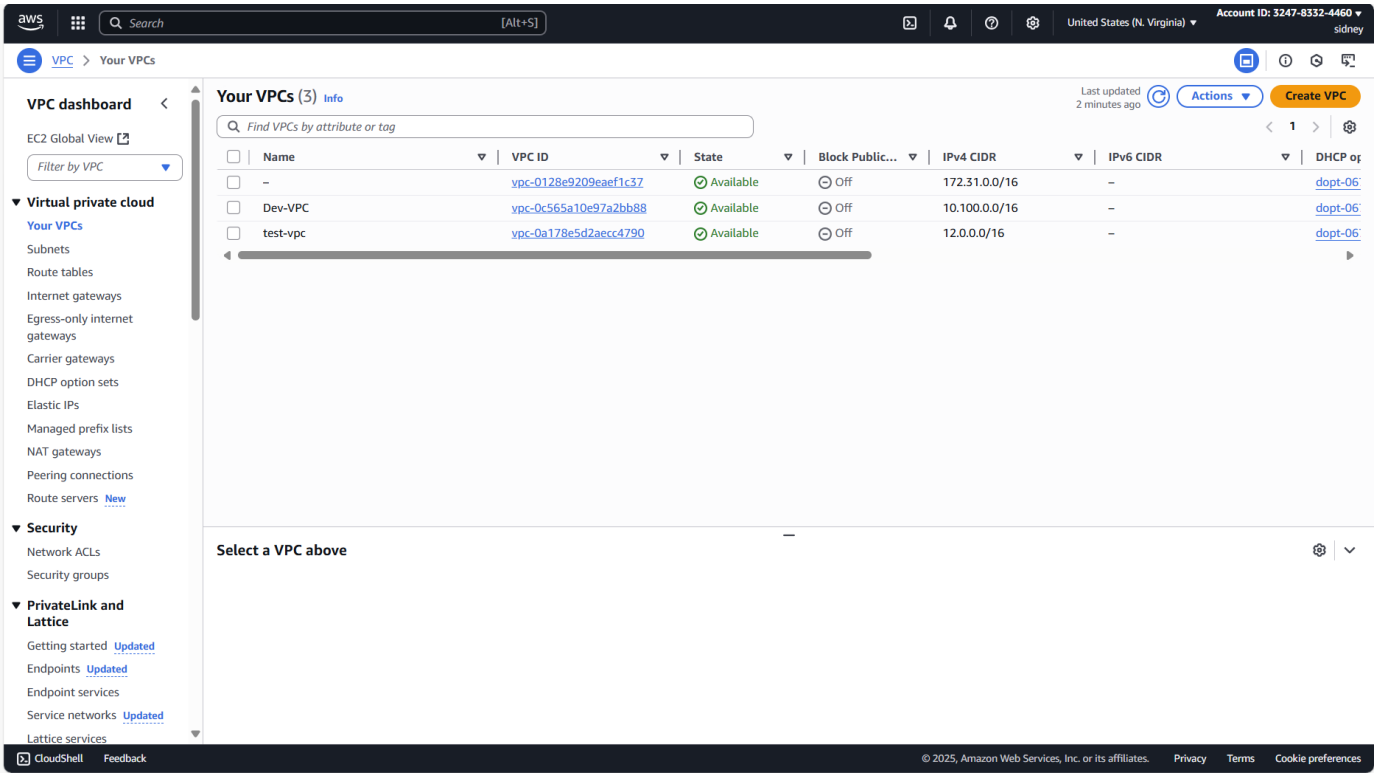
Let us start by creating the VPC, test-vpc-1. Go to AWS Management console and search for “VPC”



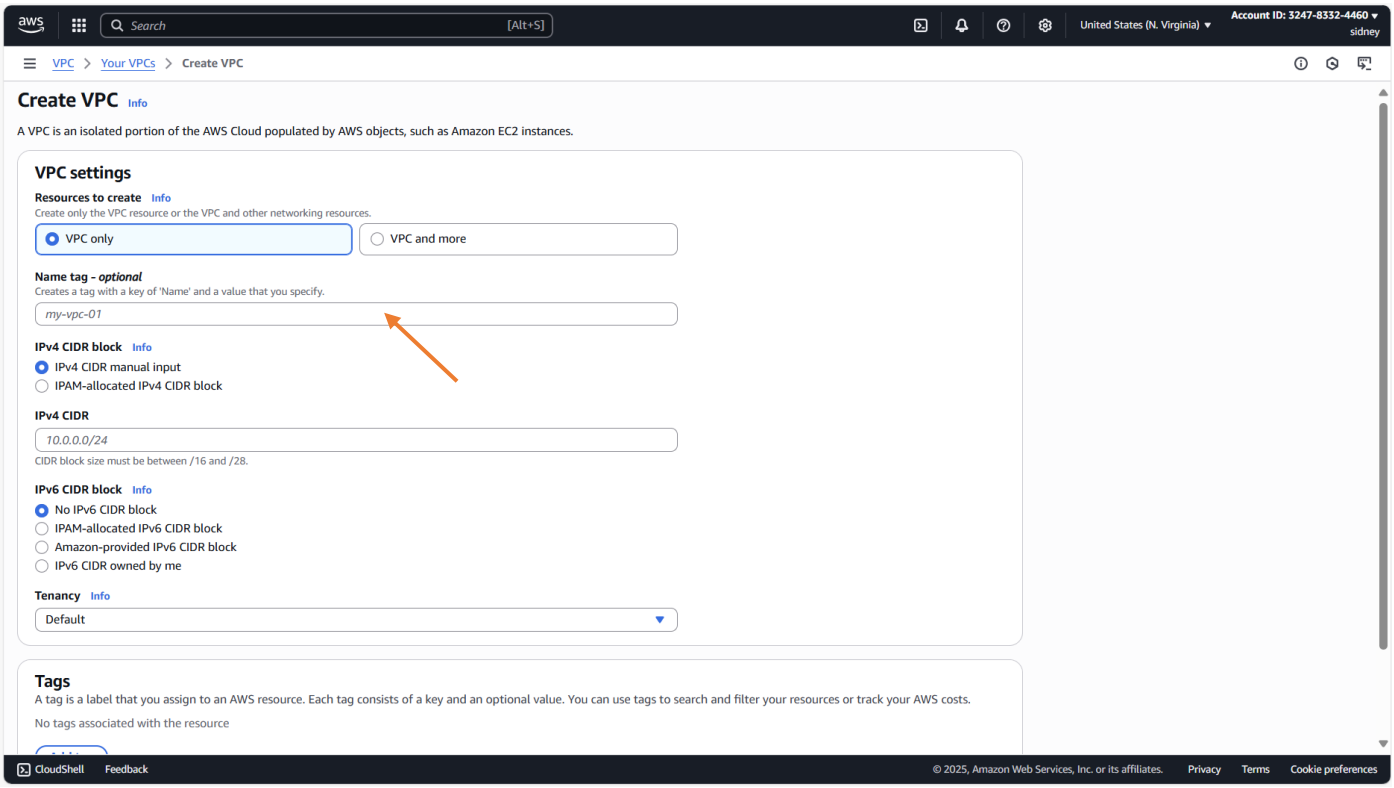
Click on “**VPC**” under services



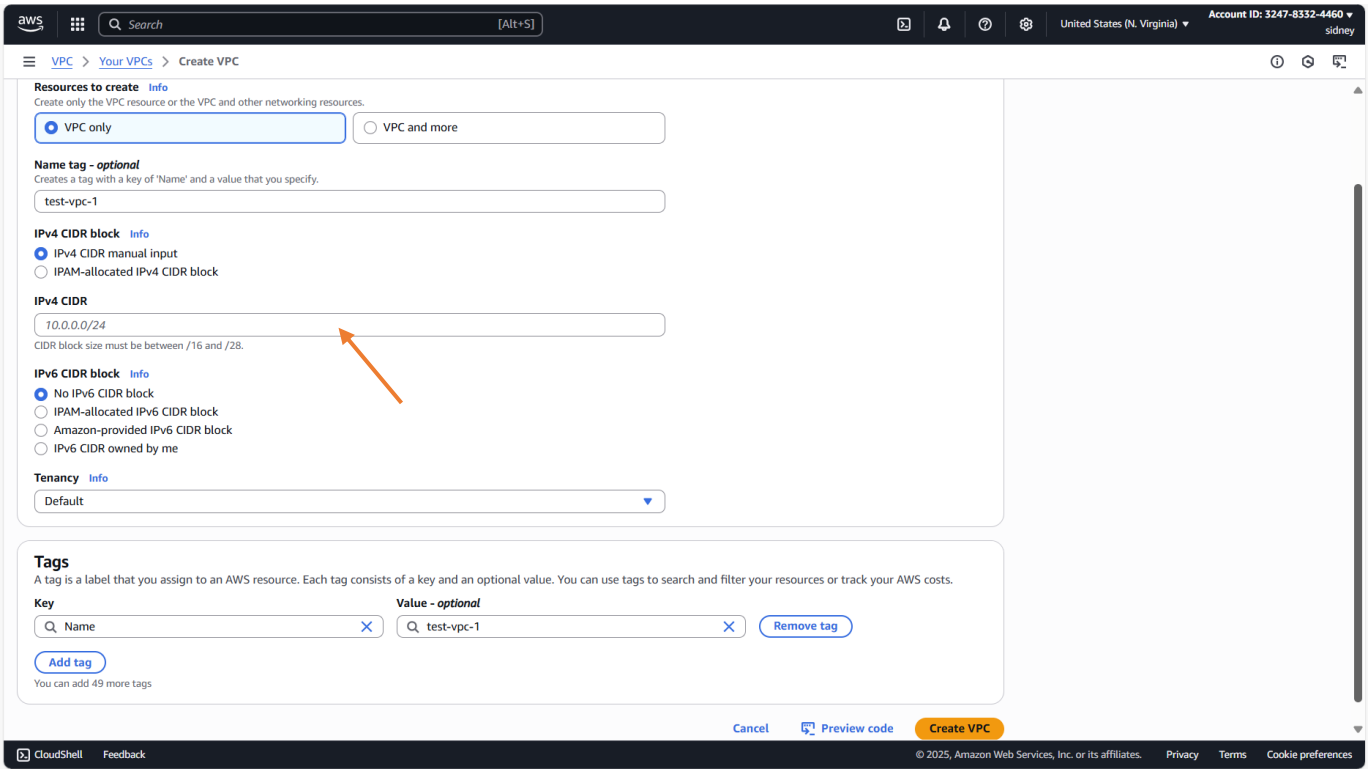
Click on “**Your VPCs**”



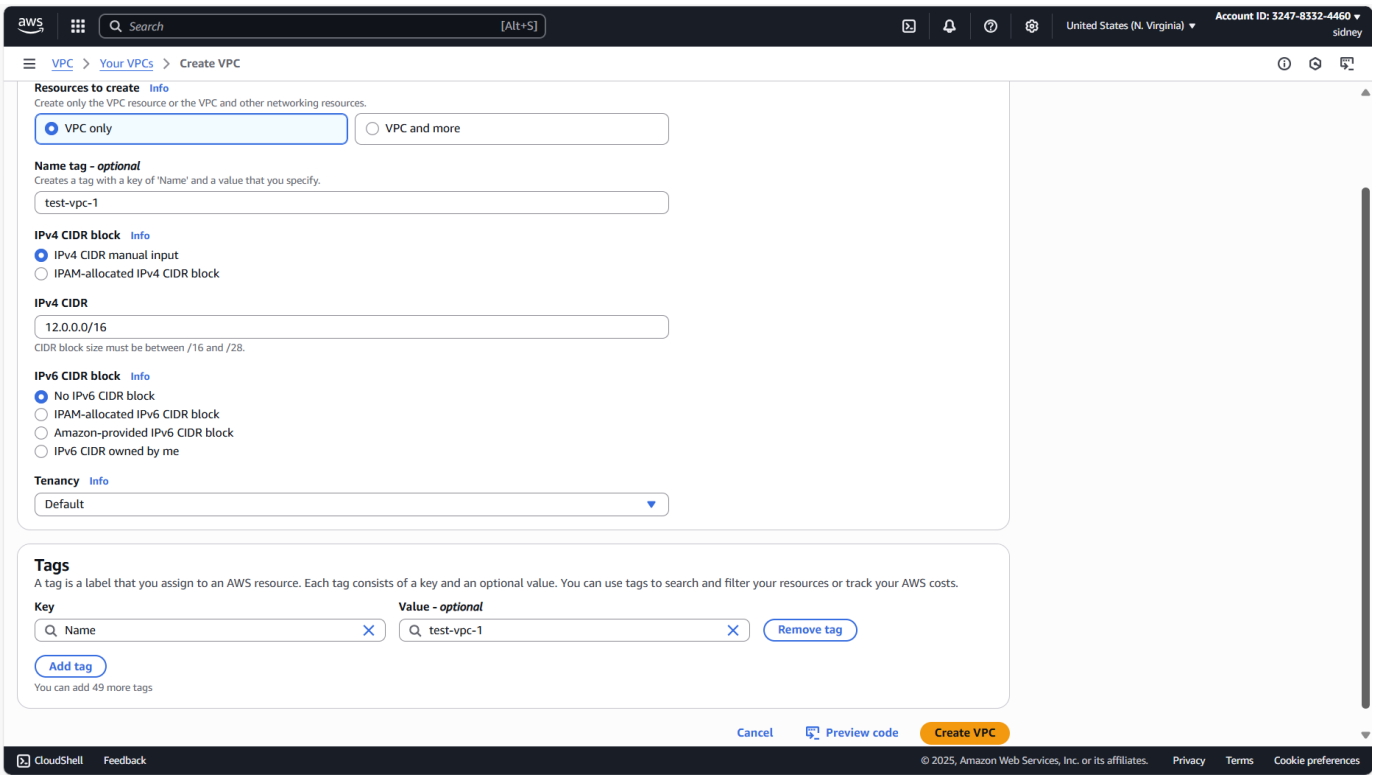
Click on "**Create VPC**"



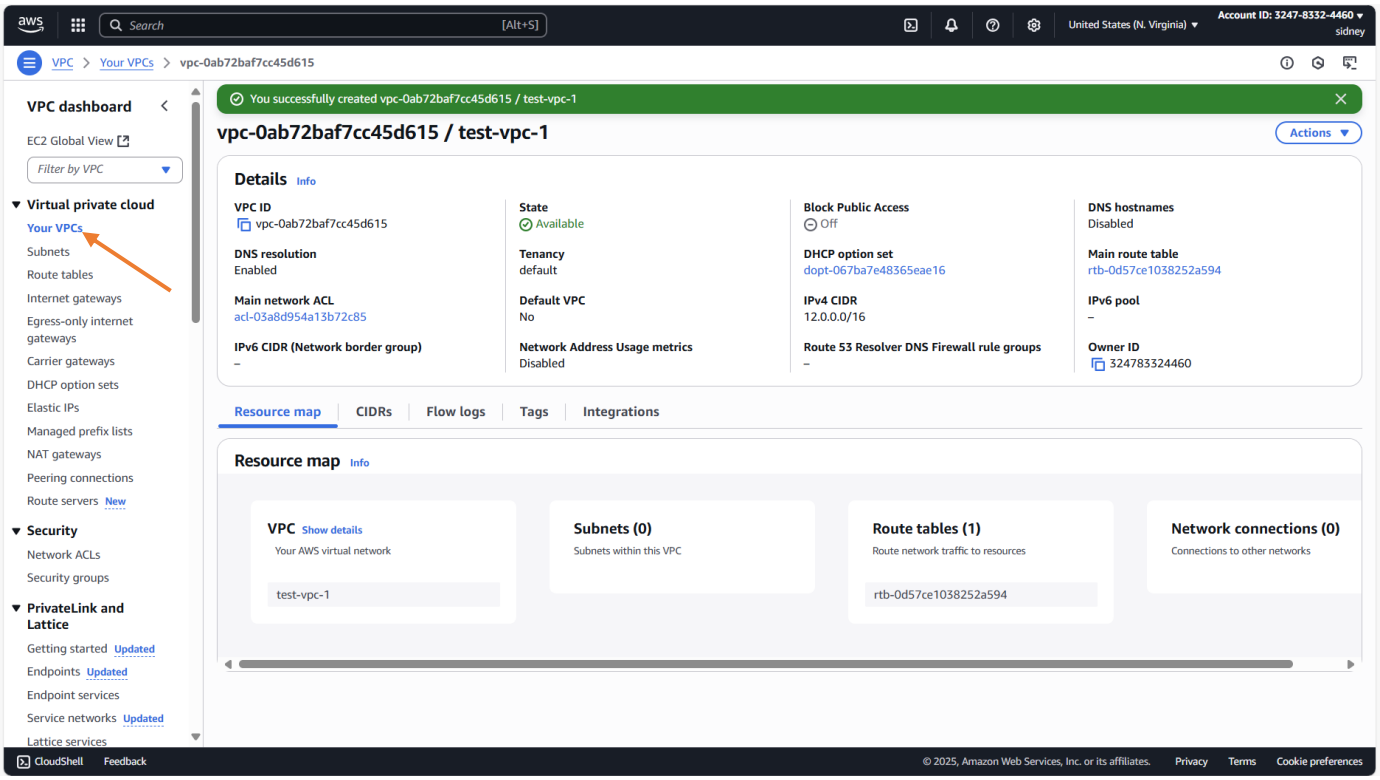
We will use "**VPC only**" as shown above. Then we will call our VPC "**test-vpc-1**"



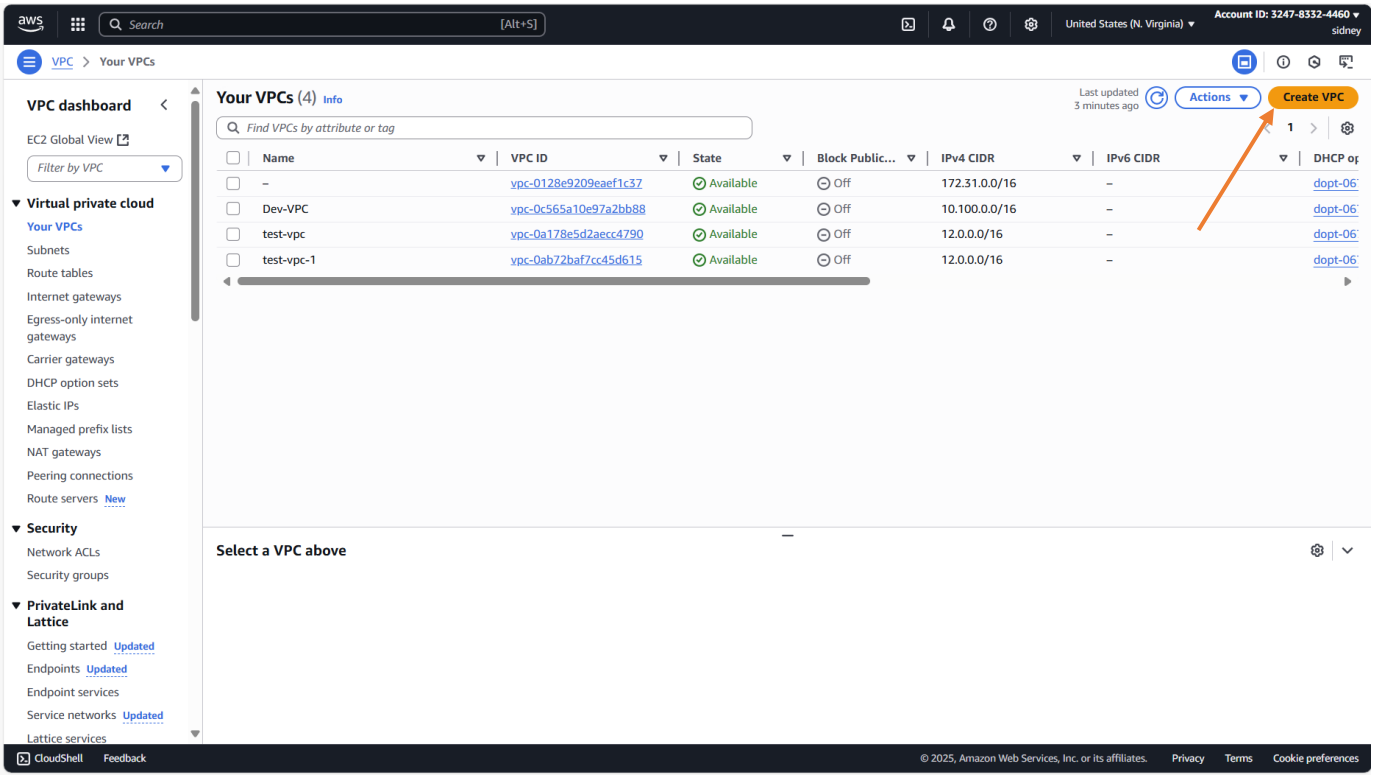
For the IPv4 CIDR, we will use "**12.0.0.0/16**"



Then click on "**create VPC**"

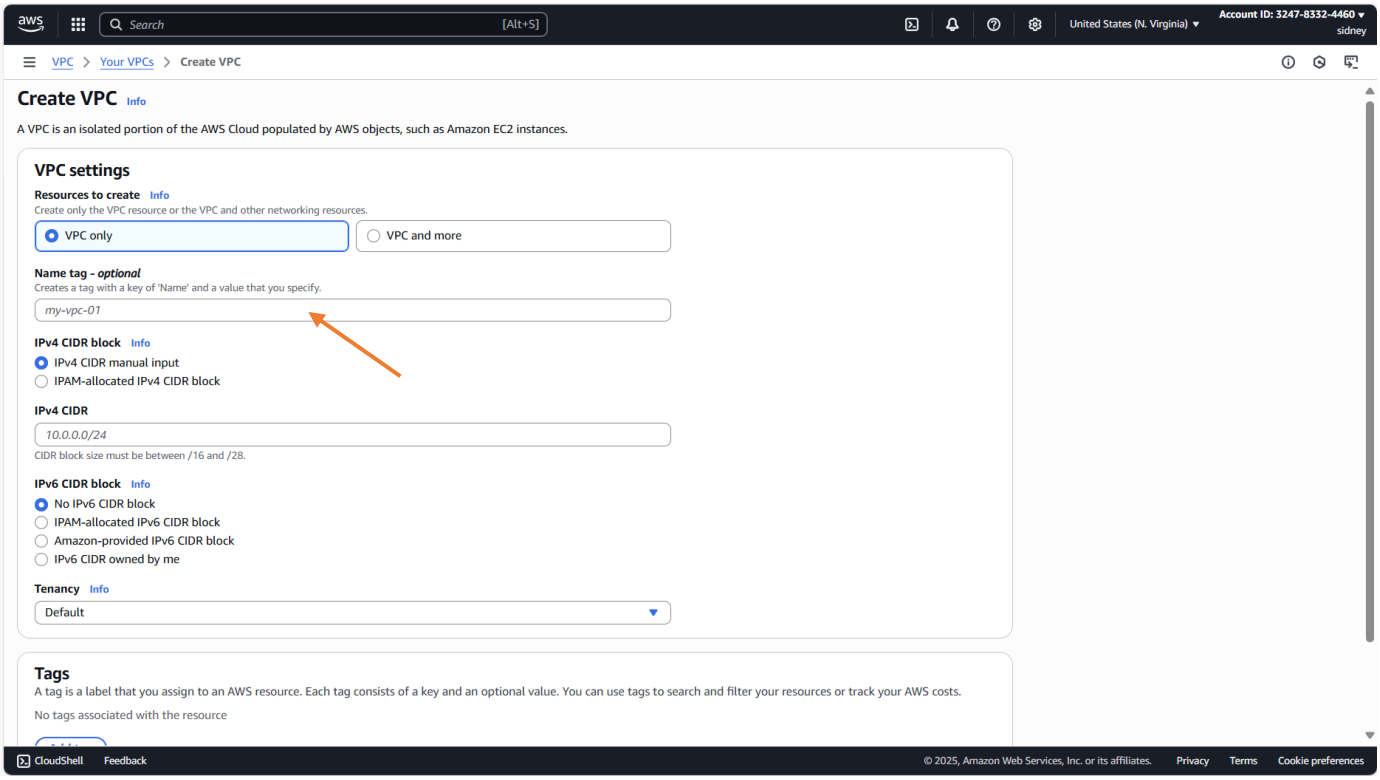


We have created the first VPC called "**test-vpc-1**", now let us create the second VPC called "**test-vpc-2**". Click on "**Your VPCs**"

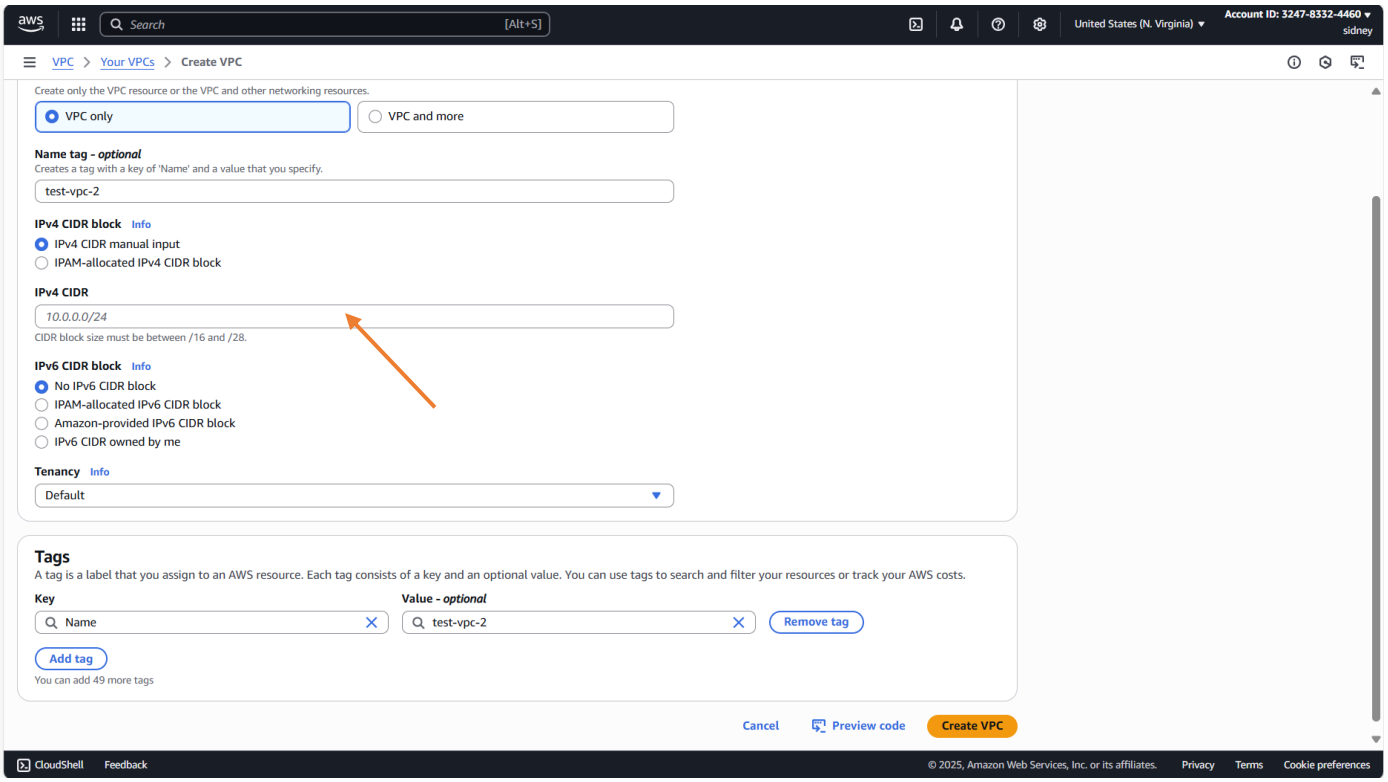


Part 3: Creating test-vpc-2

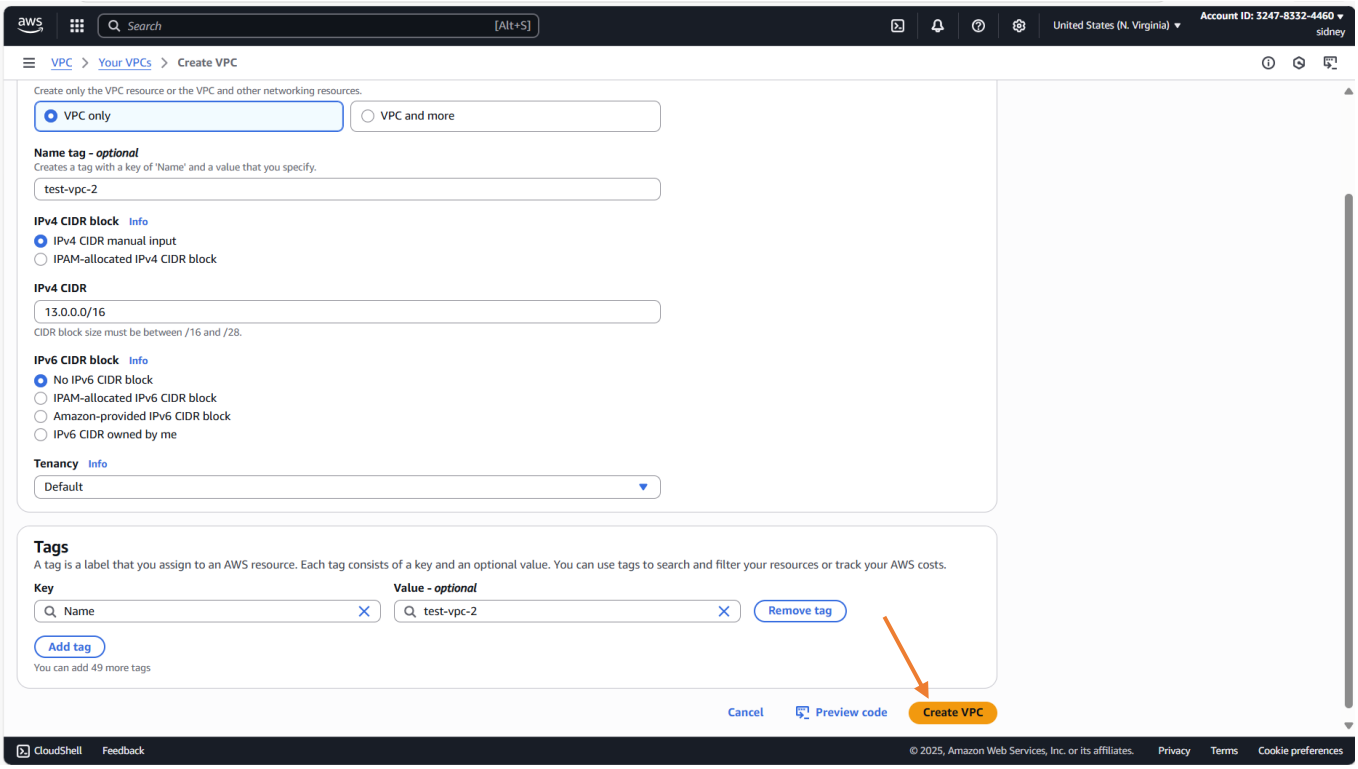
Click on “**create VPC**”



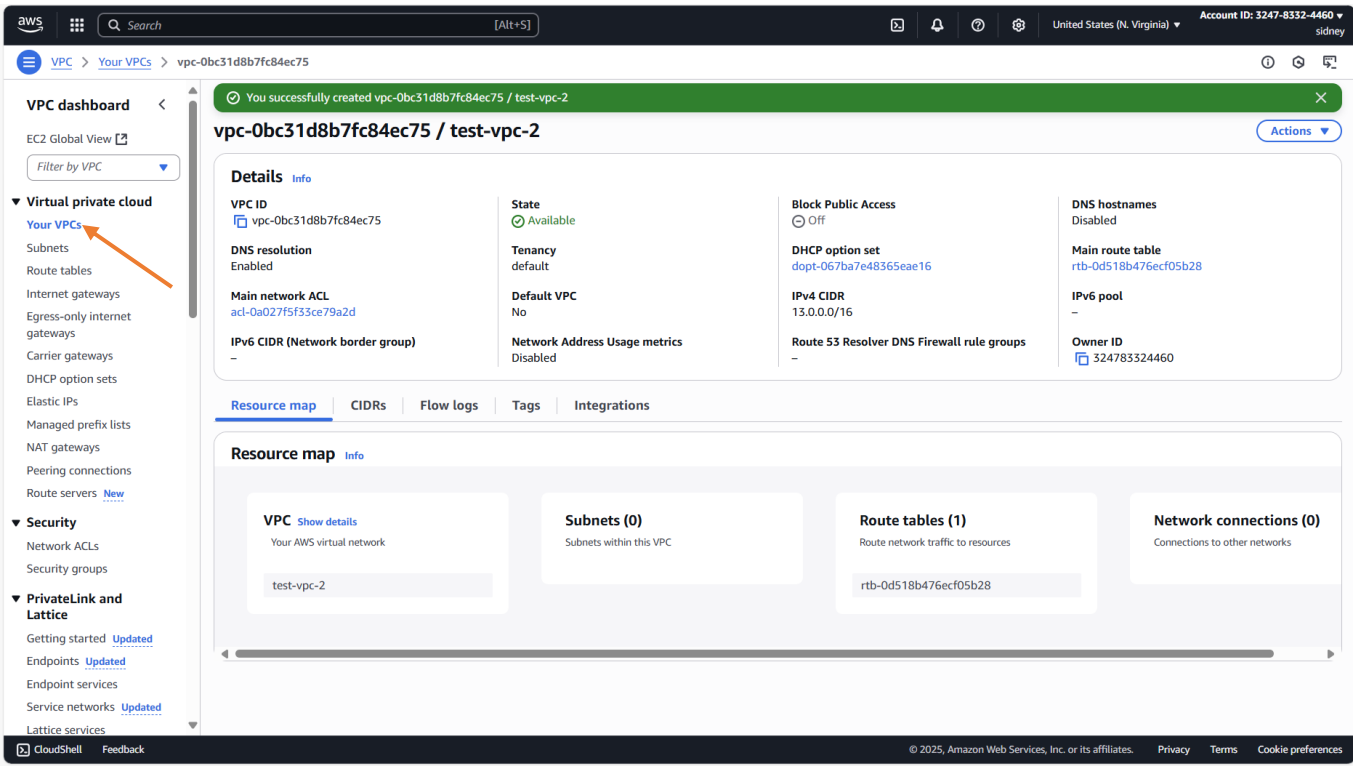
We will call this second VPC “**test-vpc-2**”



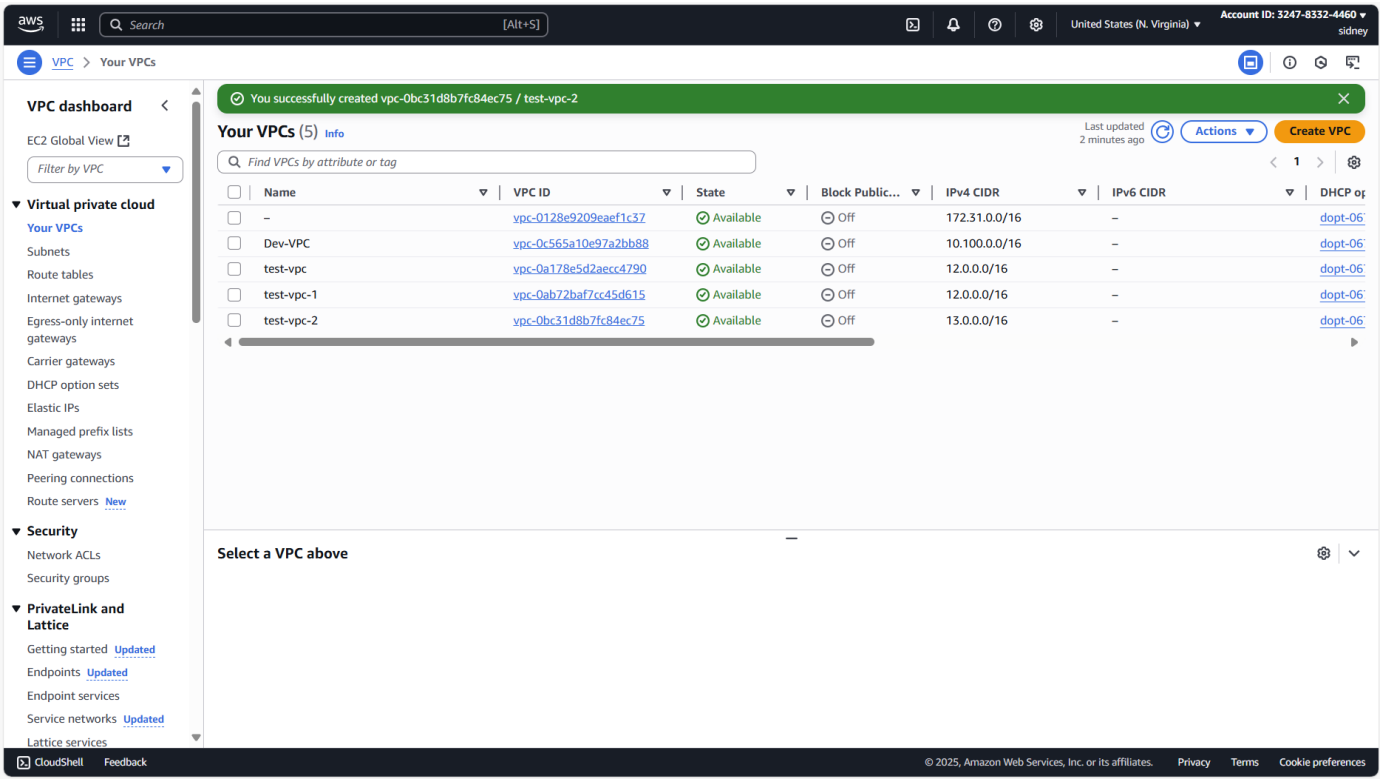
For this VPC, we will use the IPv4 CIDR “**13.0.0.0/16**”



Click on “**create VPC**”

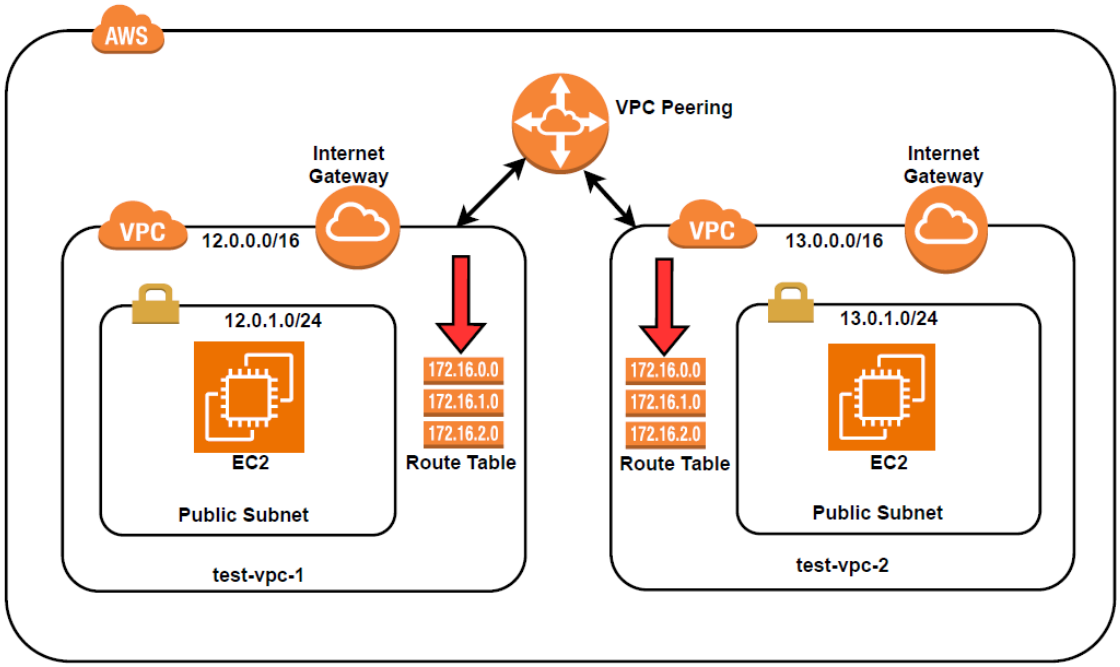


We have created the second VPC. Click on “**Your VPCs**”



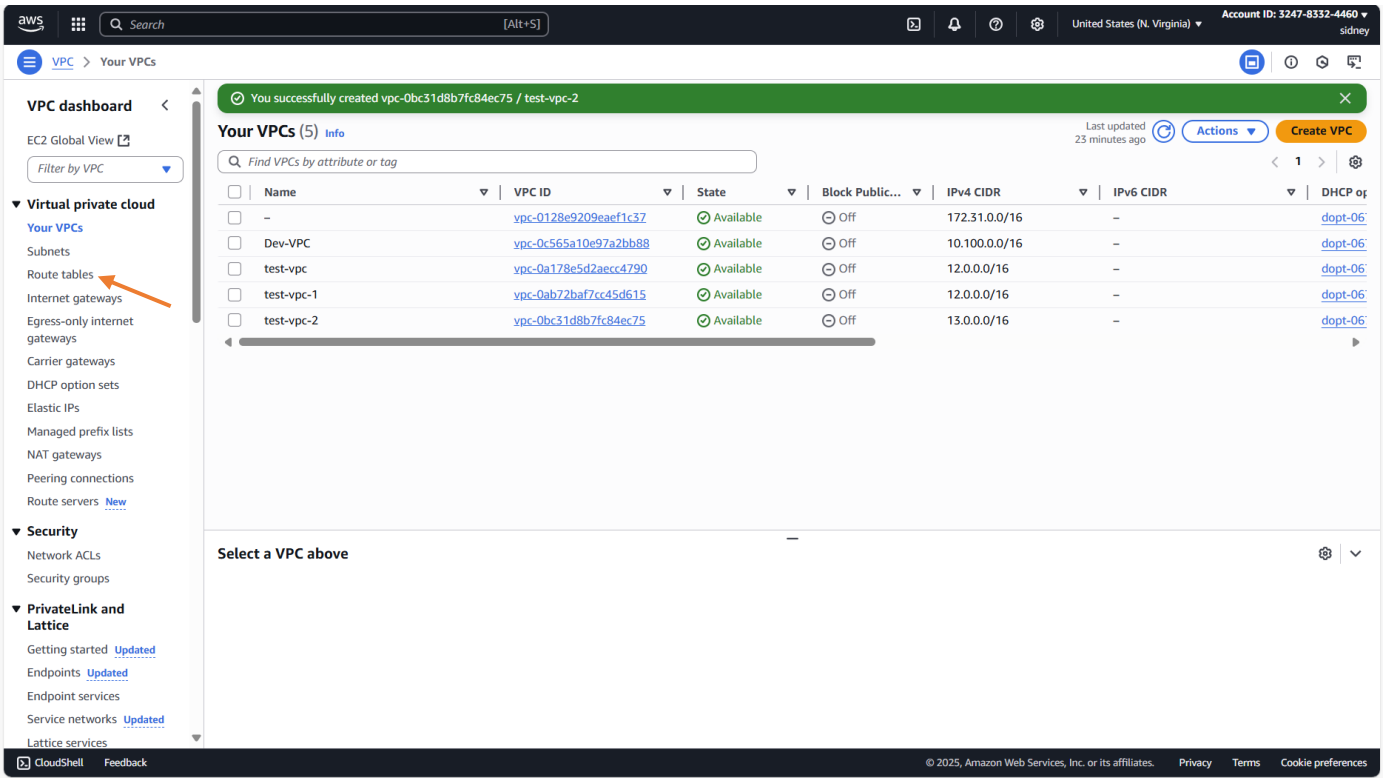
You can see our two VPC for this project: **test-vpc-1** and **test-vpc-2**.

Part 4: Create route tables

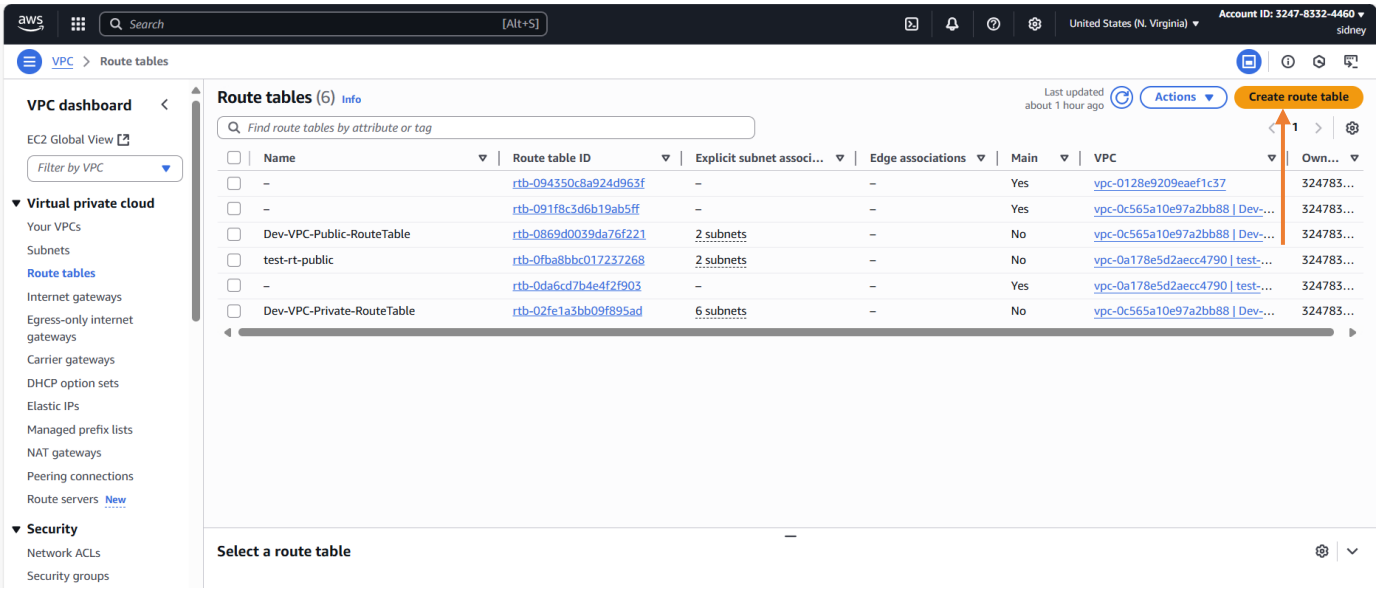


We are going to create two Route Tables one for test-vpc-1 and the other for **test-vpc-2**, so that it can route the traffic to our particular subnet.

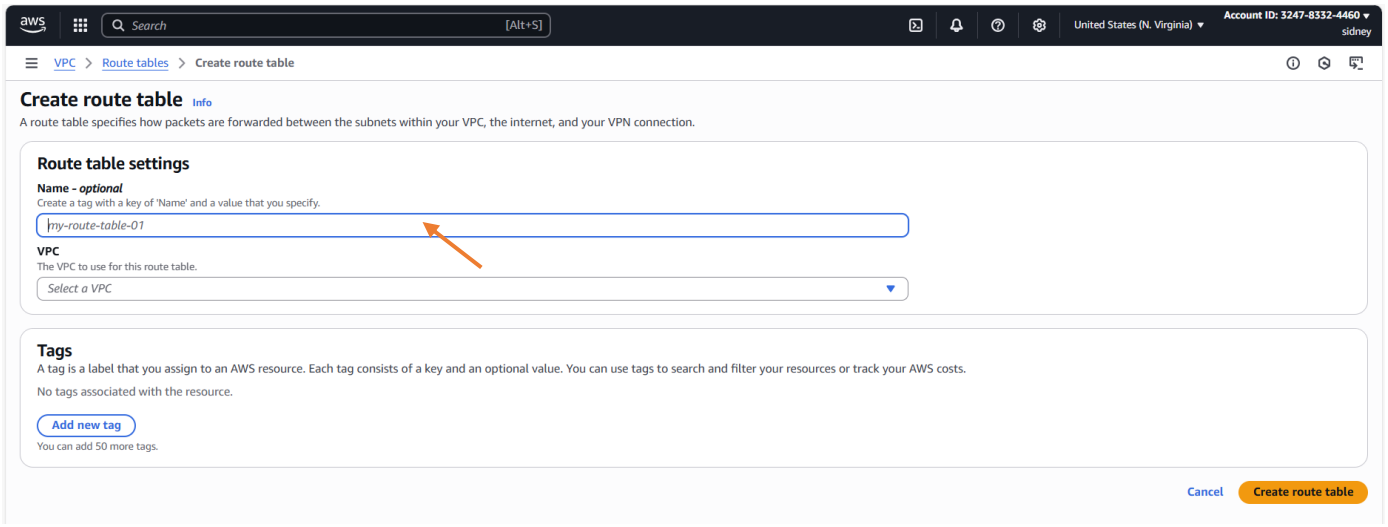
Let us go back to the VPC dashboard on our AWS Management Console



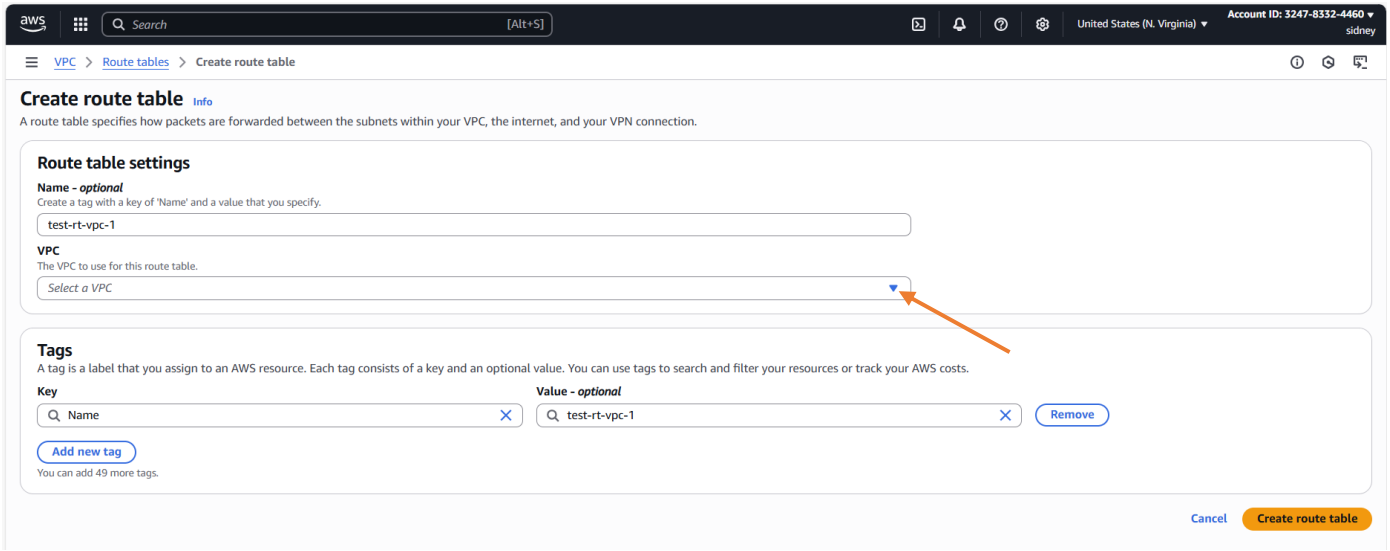
Click on “**Route Tables**”



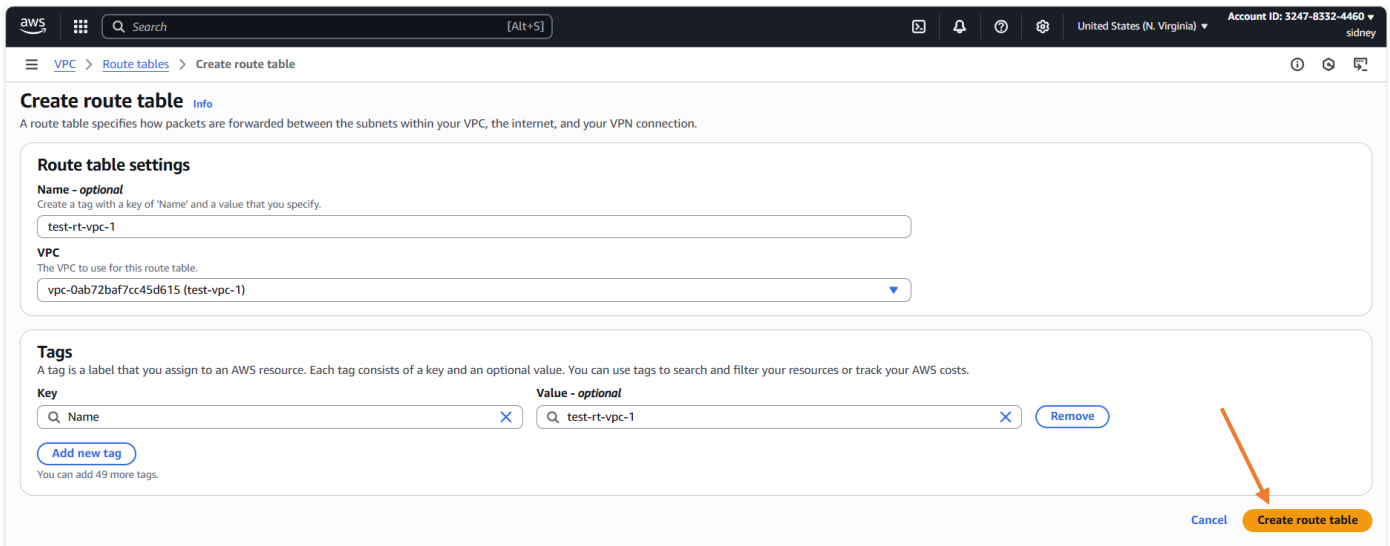
Click on “**create Route Table**”



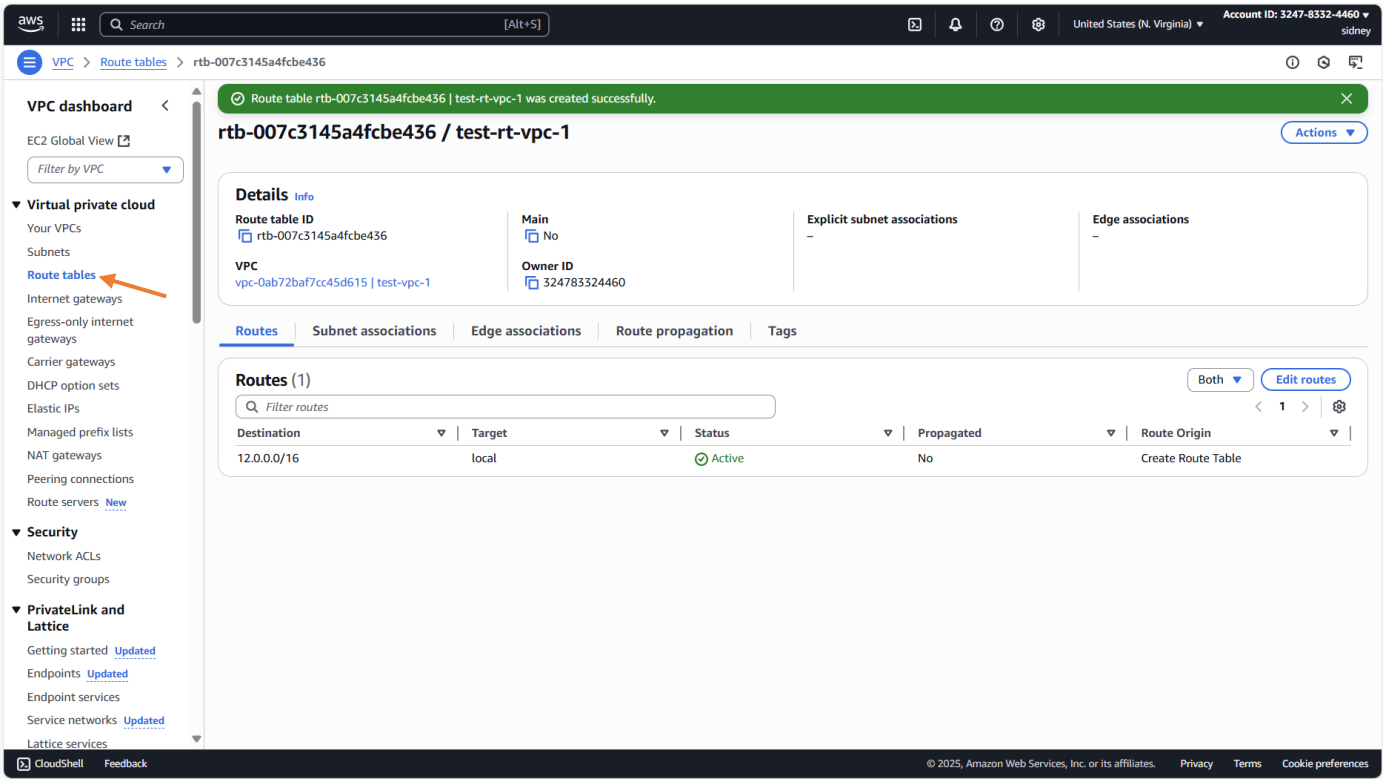
Give the Route Table a name, we will call it "**test-rt-vpc-1**".



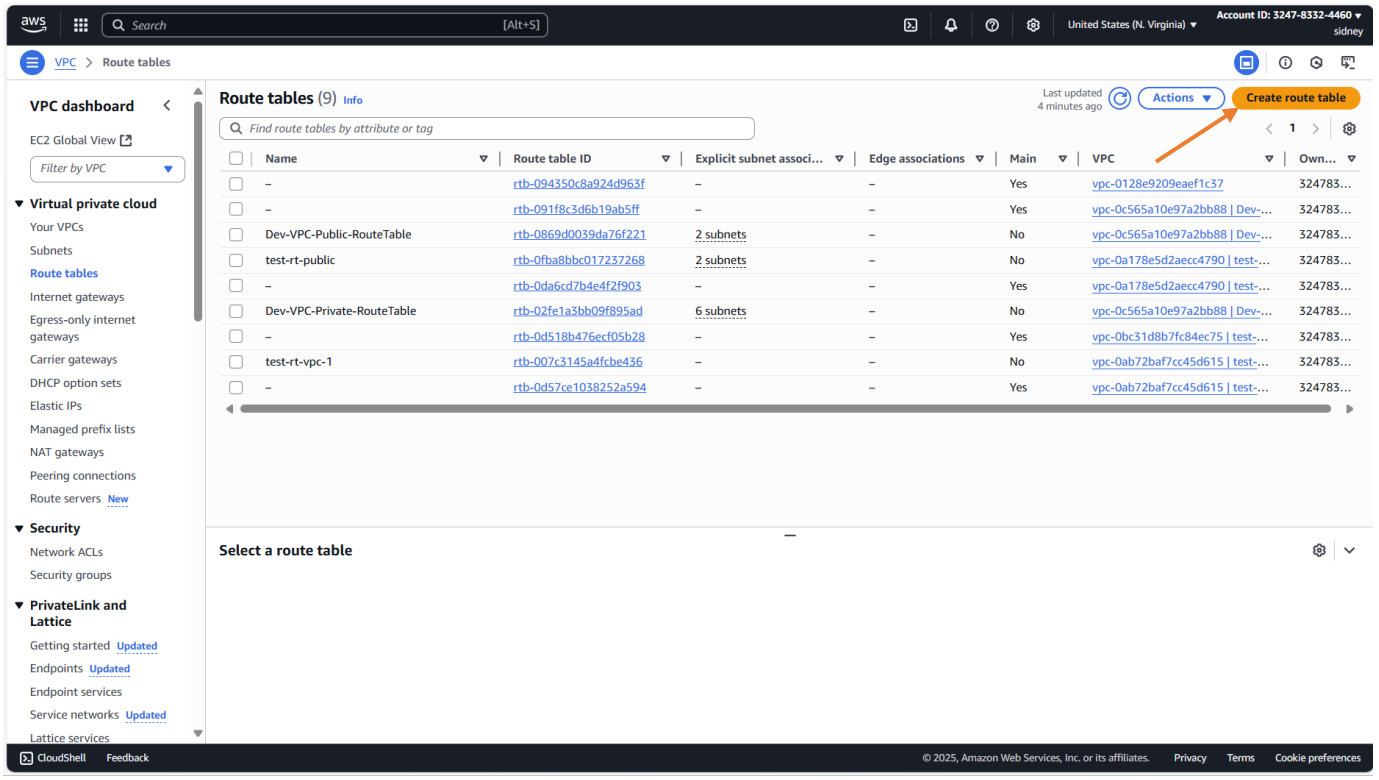
Click on the drop down and select our first VPC "**test-vpc-1**"



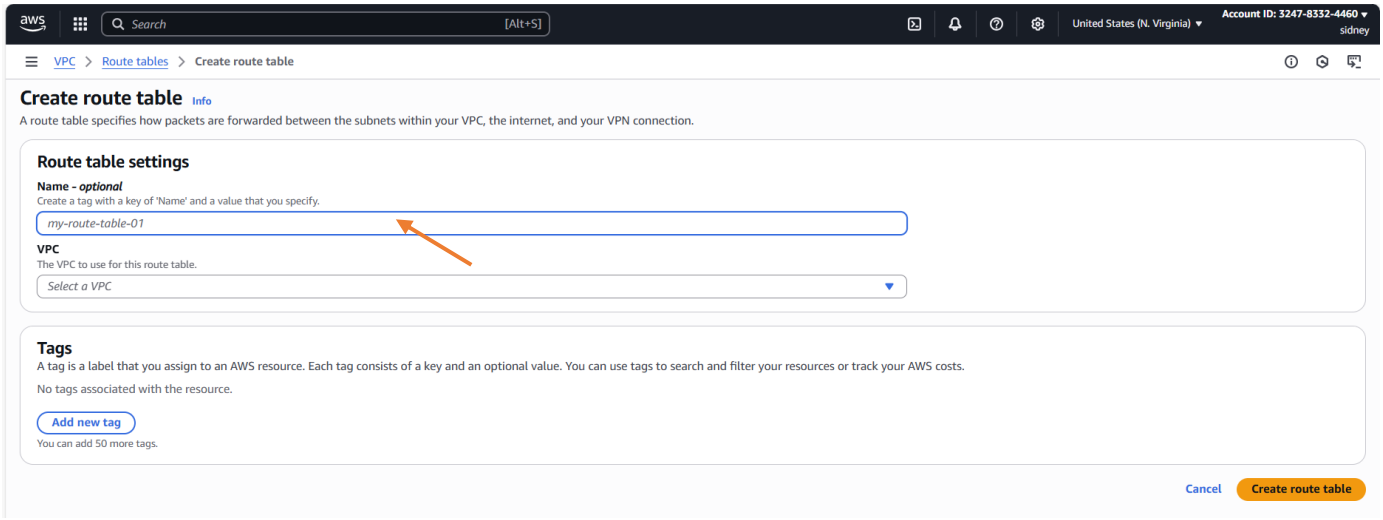
Click on “**create route table**”



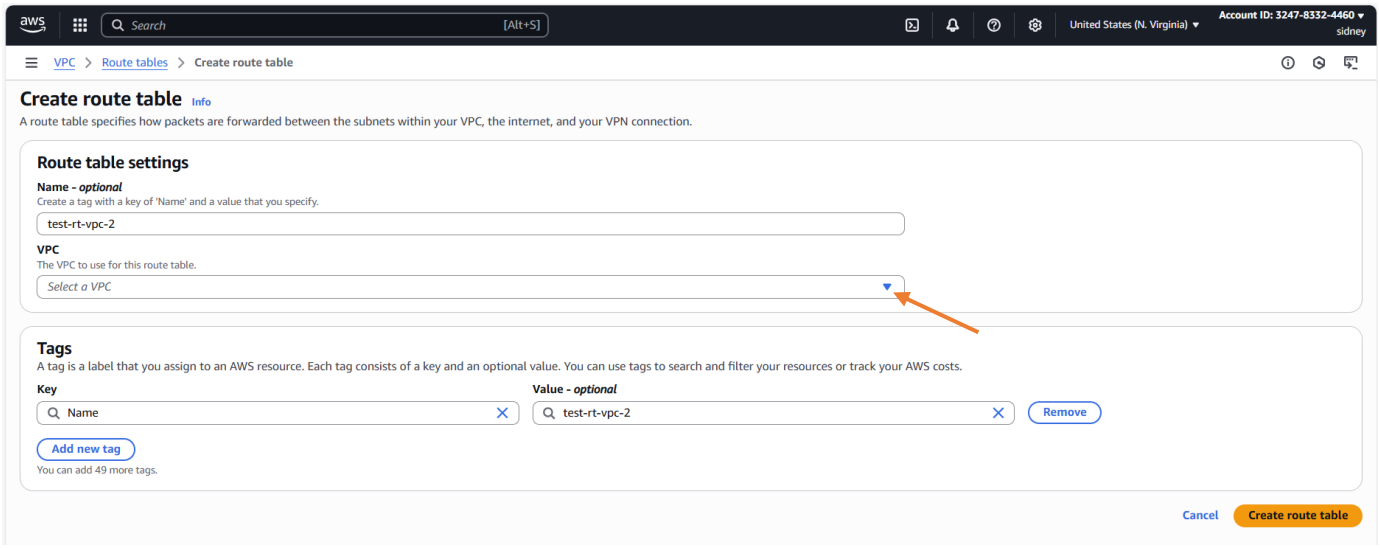
We have created the route table for “**test-vpc-1**”. Now, let us create a Route Table for “**test-vpc-2**”. Click on “**Route Tables**”



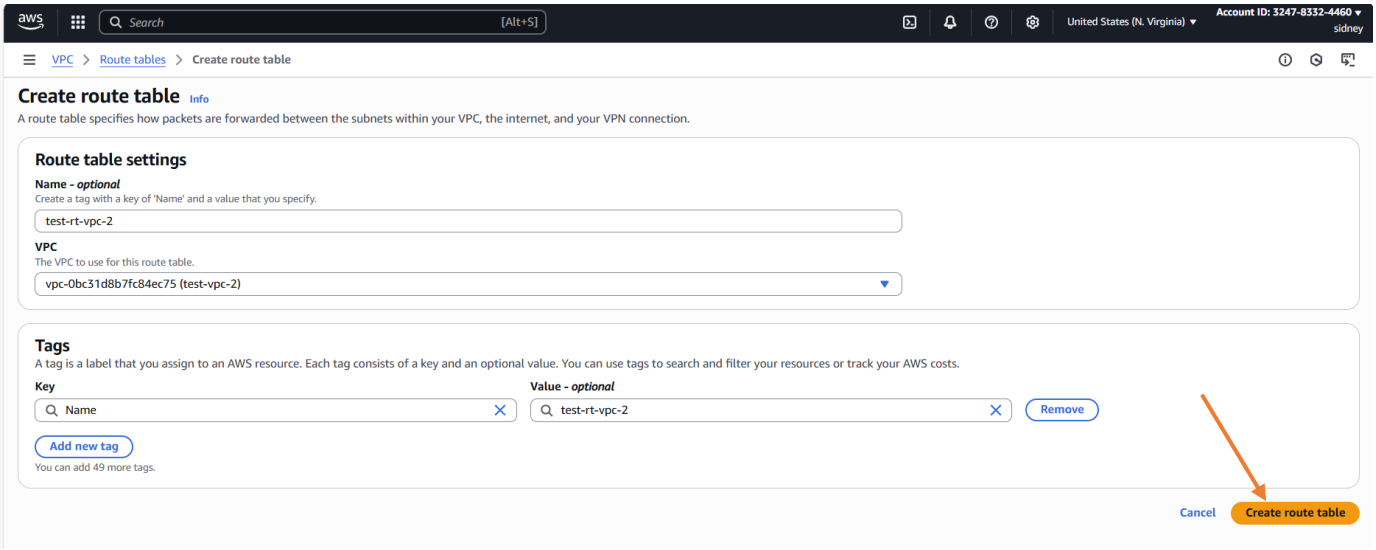
Click on “**Create Route Table**”



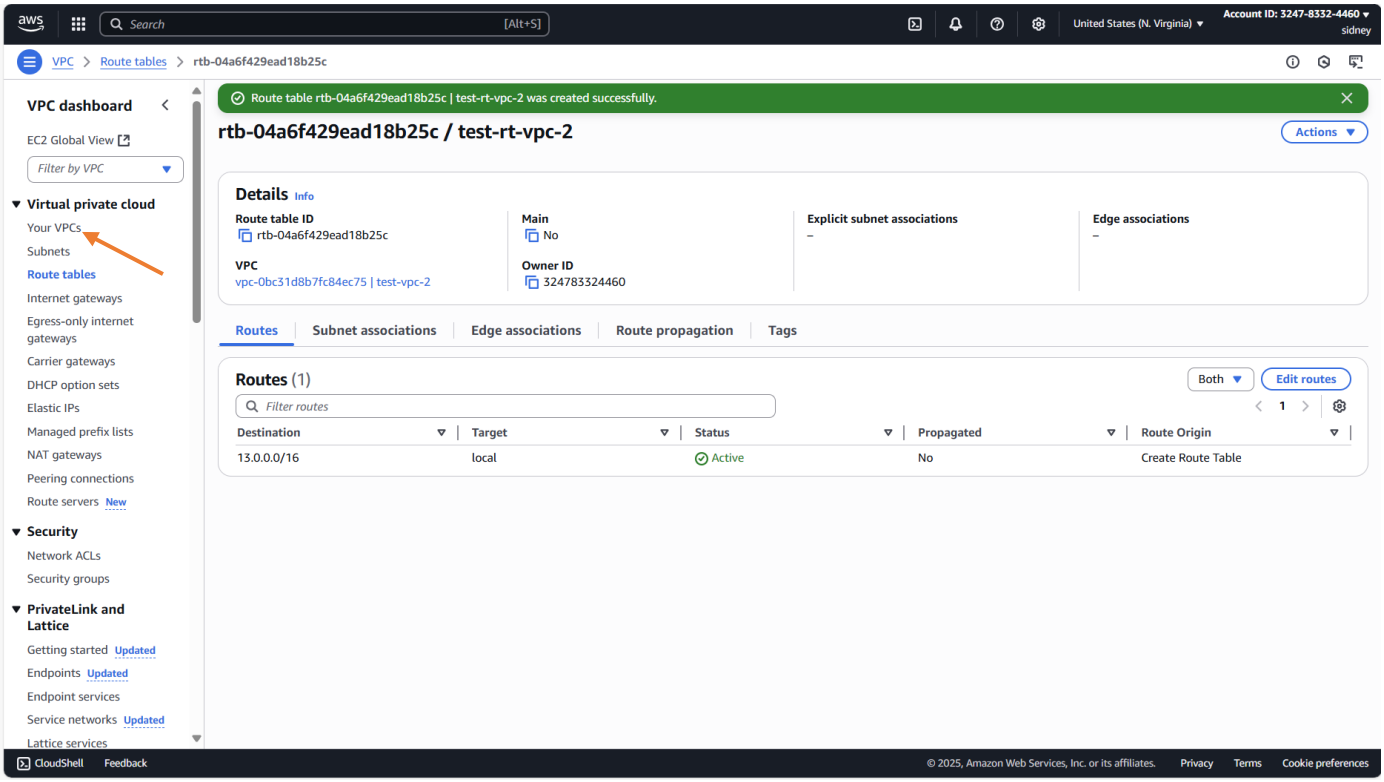
We will call our Route Table "**test-rt-vpc-2**"



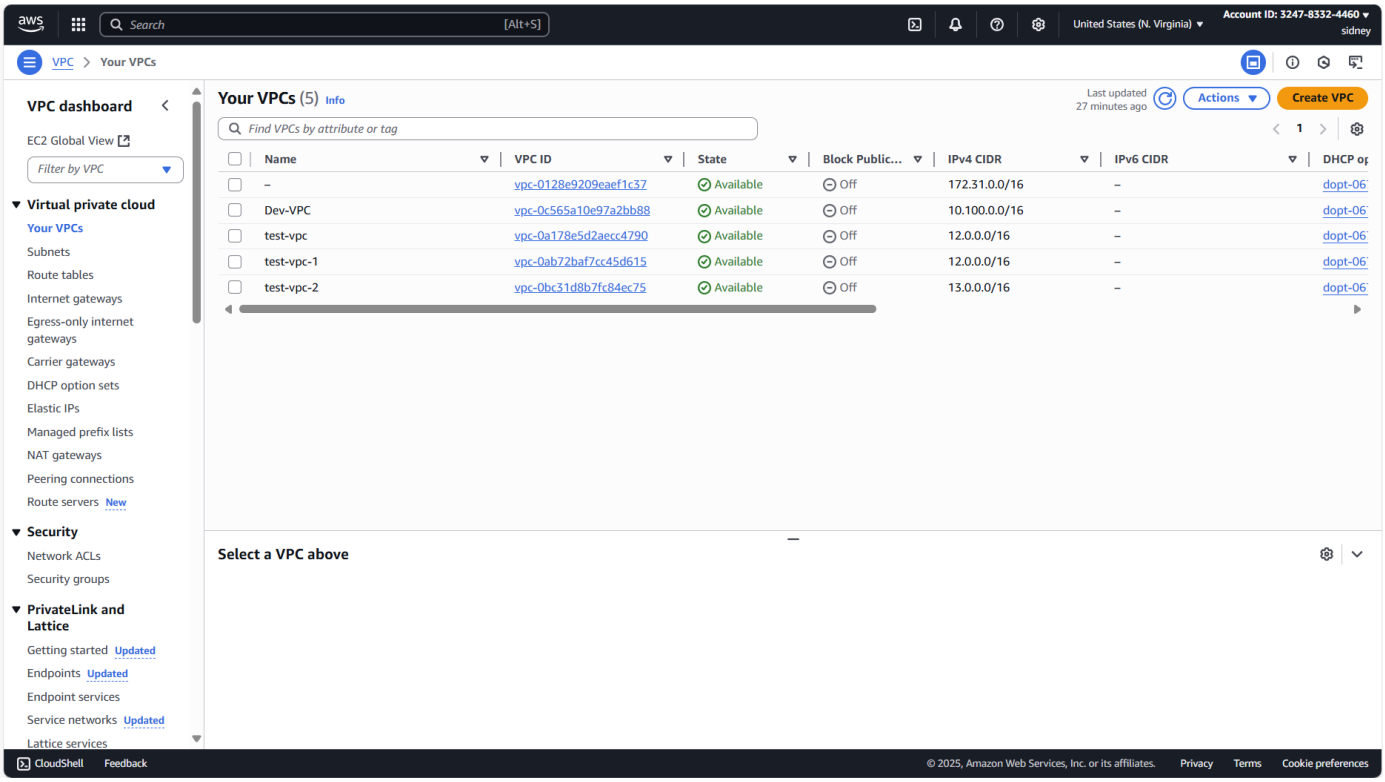
Click on the drop down and select our second VPC, "**test-vpc-2**"



Click on “**create route table**”

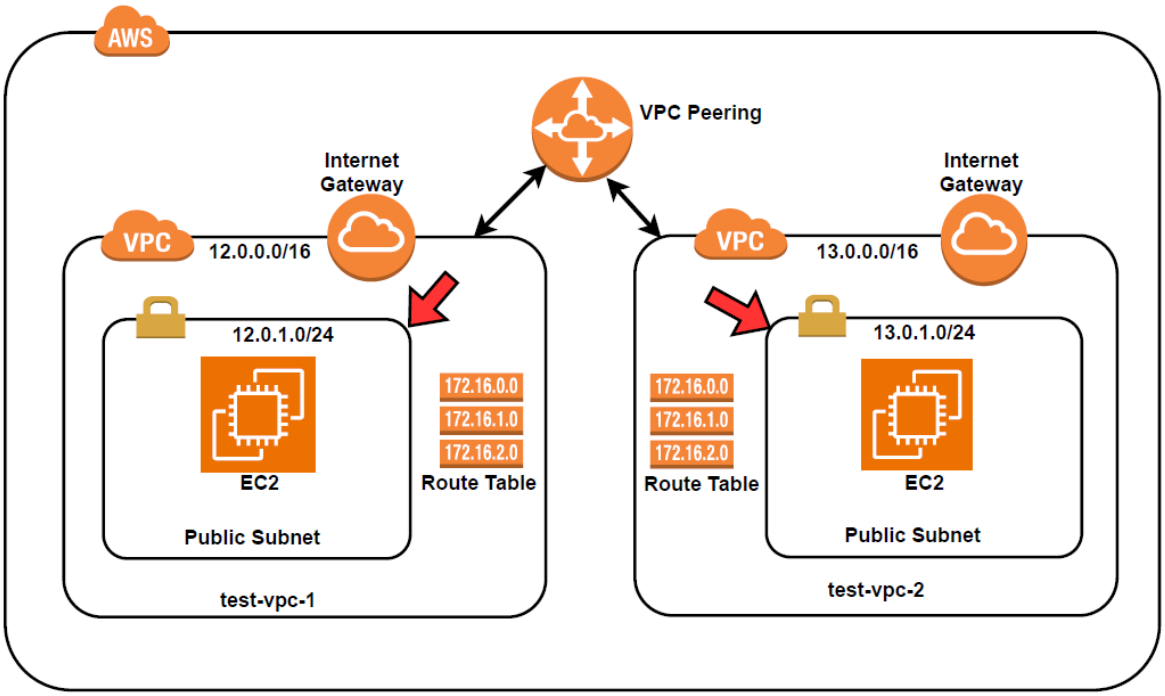


We have created the Route Table for “**test-vpc-2**”. Click on “**Your VPCs**”

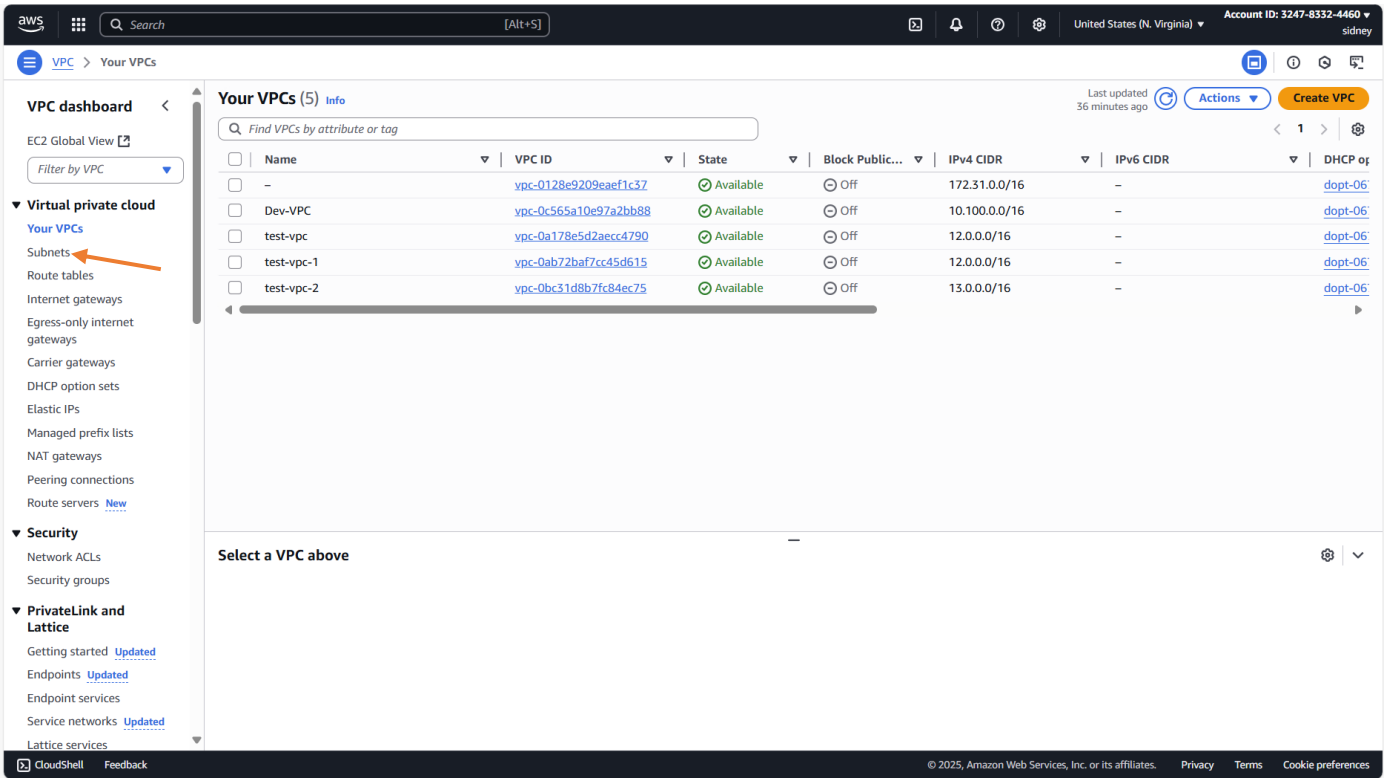


You can see the two Route Tables we have created.

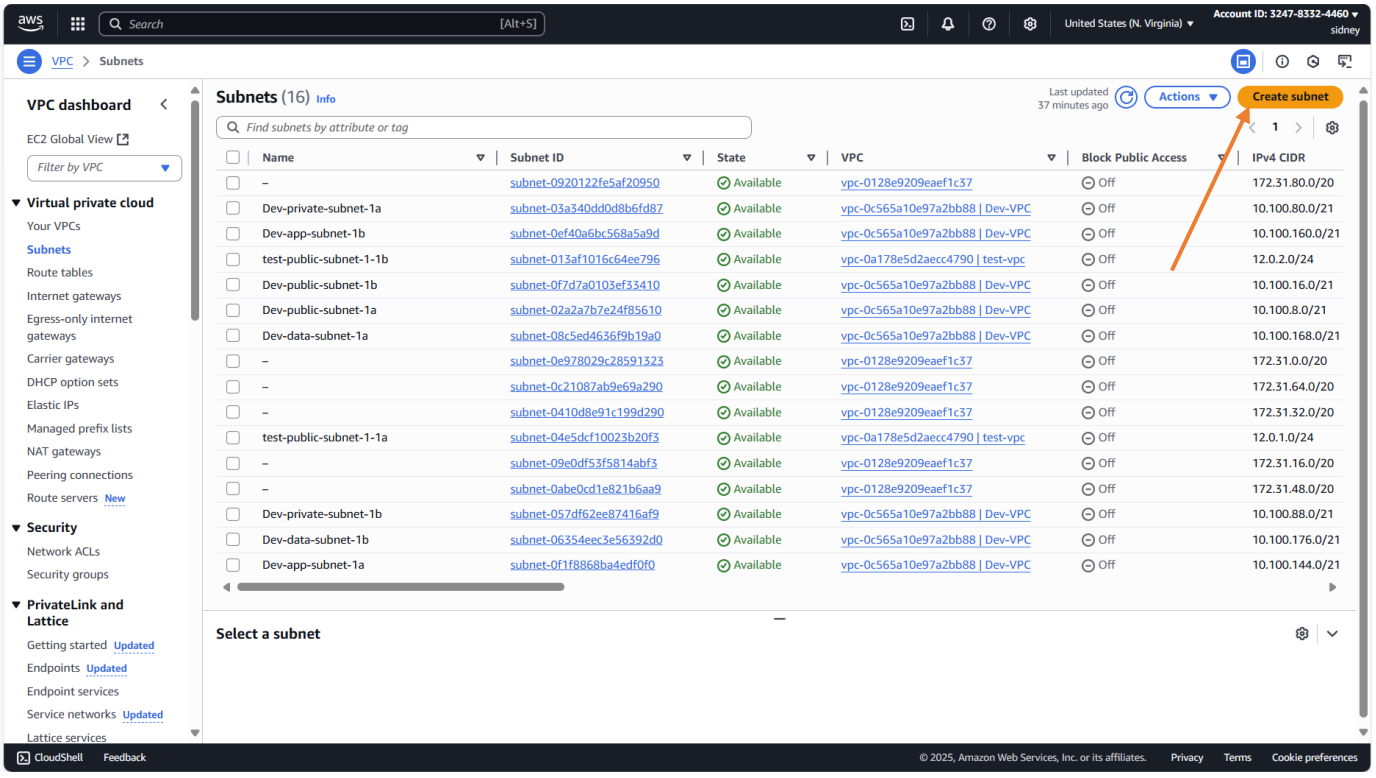
Part 5: Create Subnet



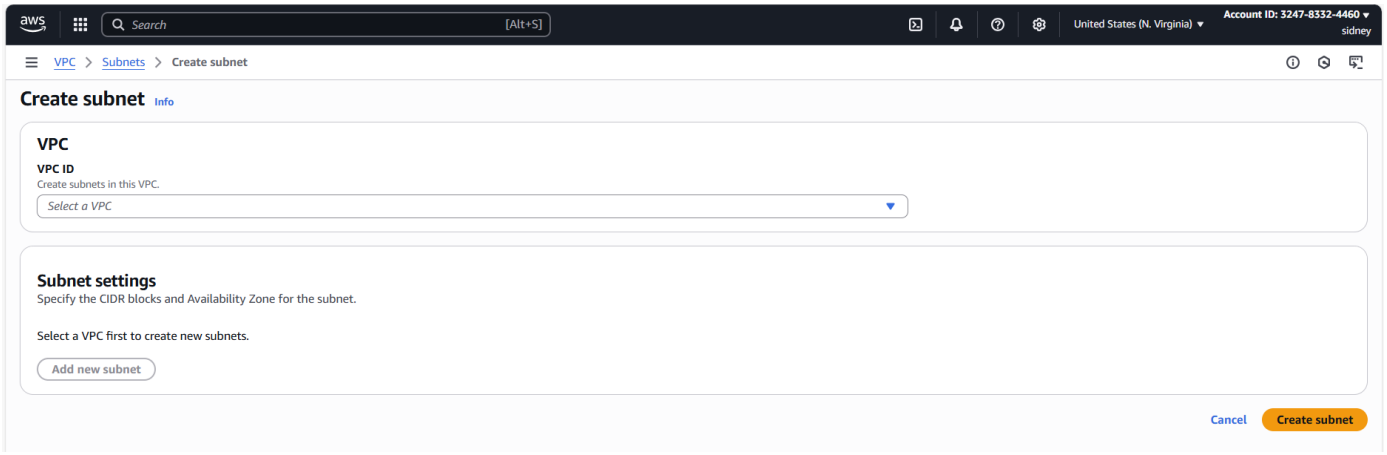
After creating the route tables, the next thing we have to do is to create the subnets. To do that, go to the VPC dashboard on AWS Management Console



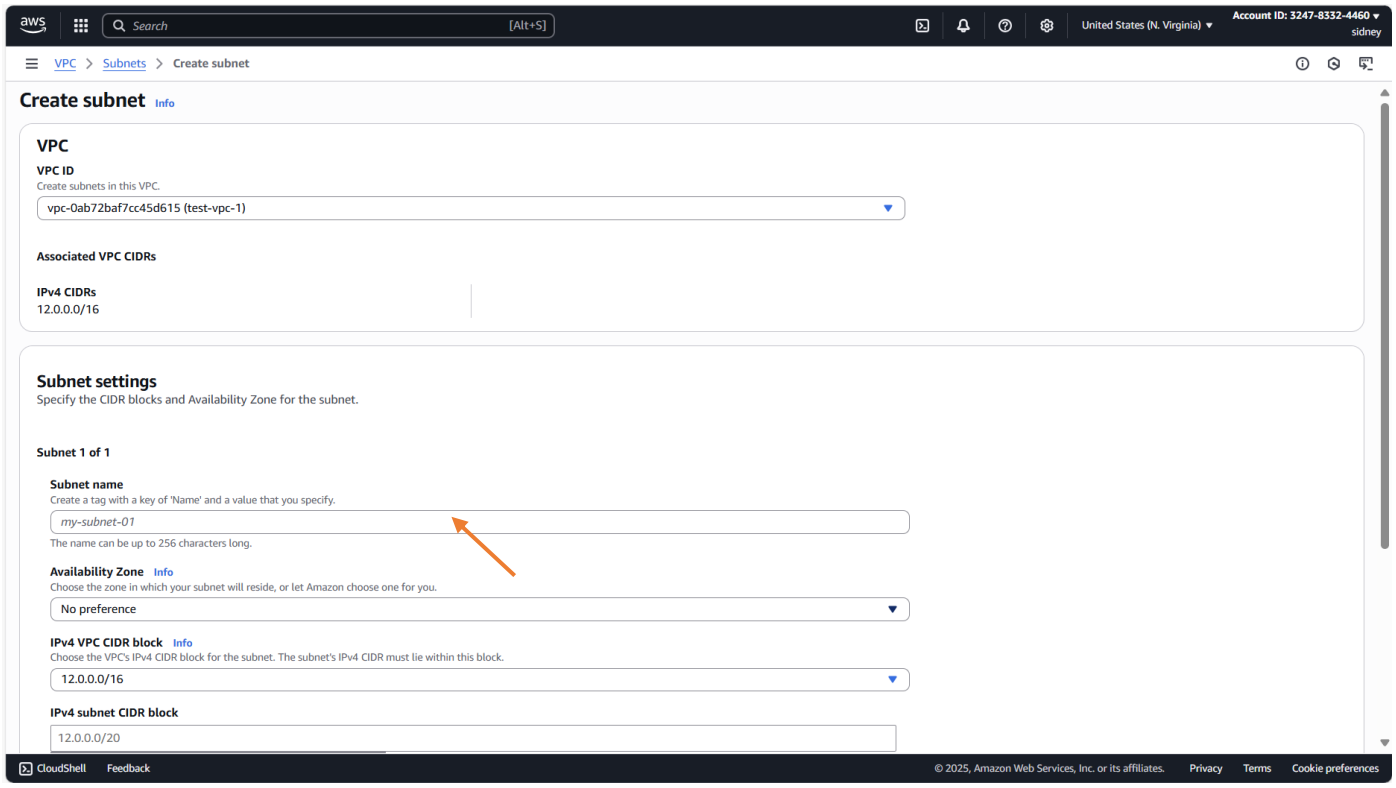
Click on “**Subnets**”



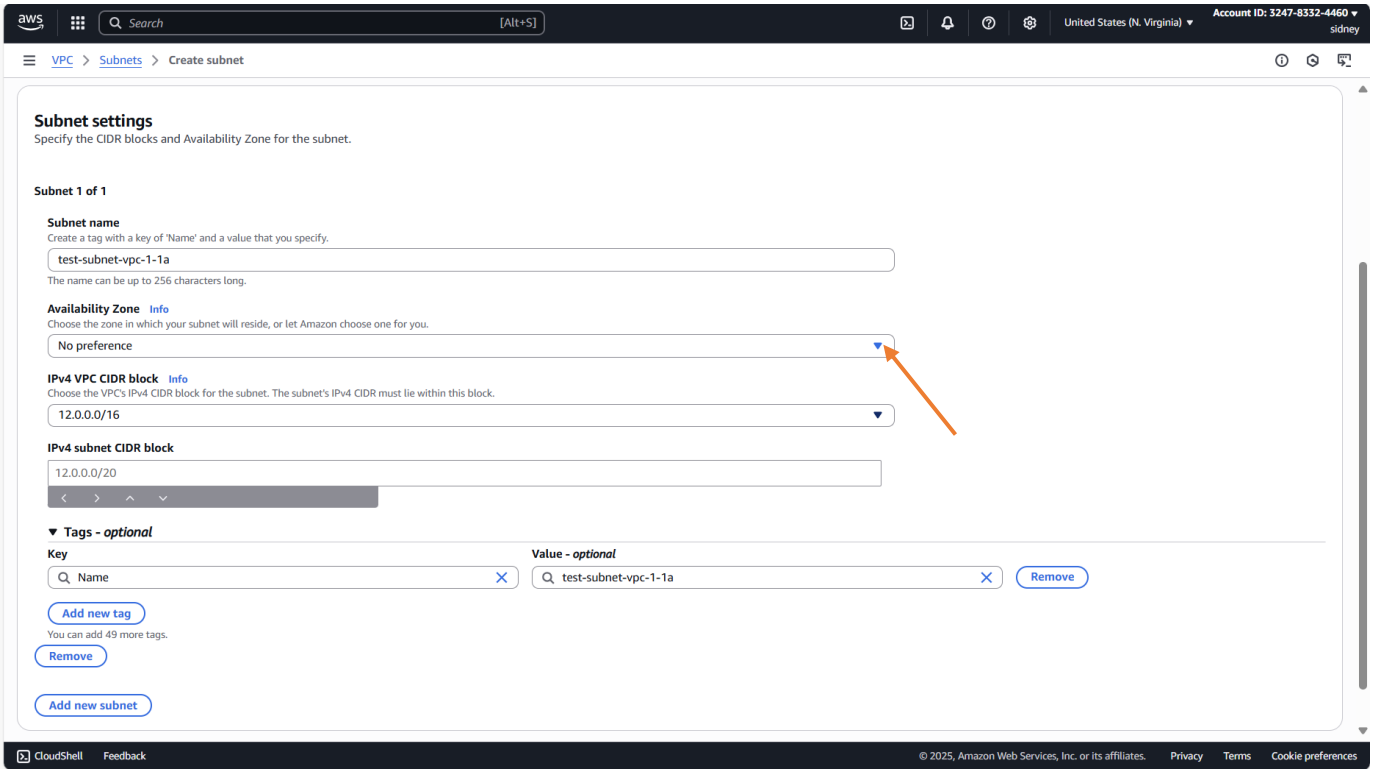
Click on “**create subnet**”



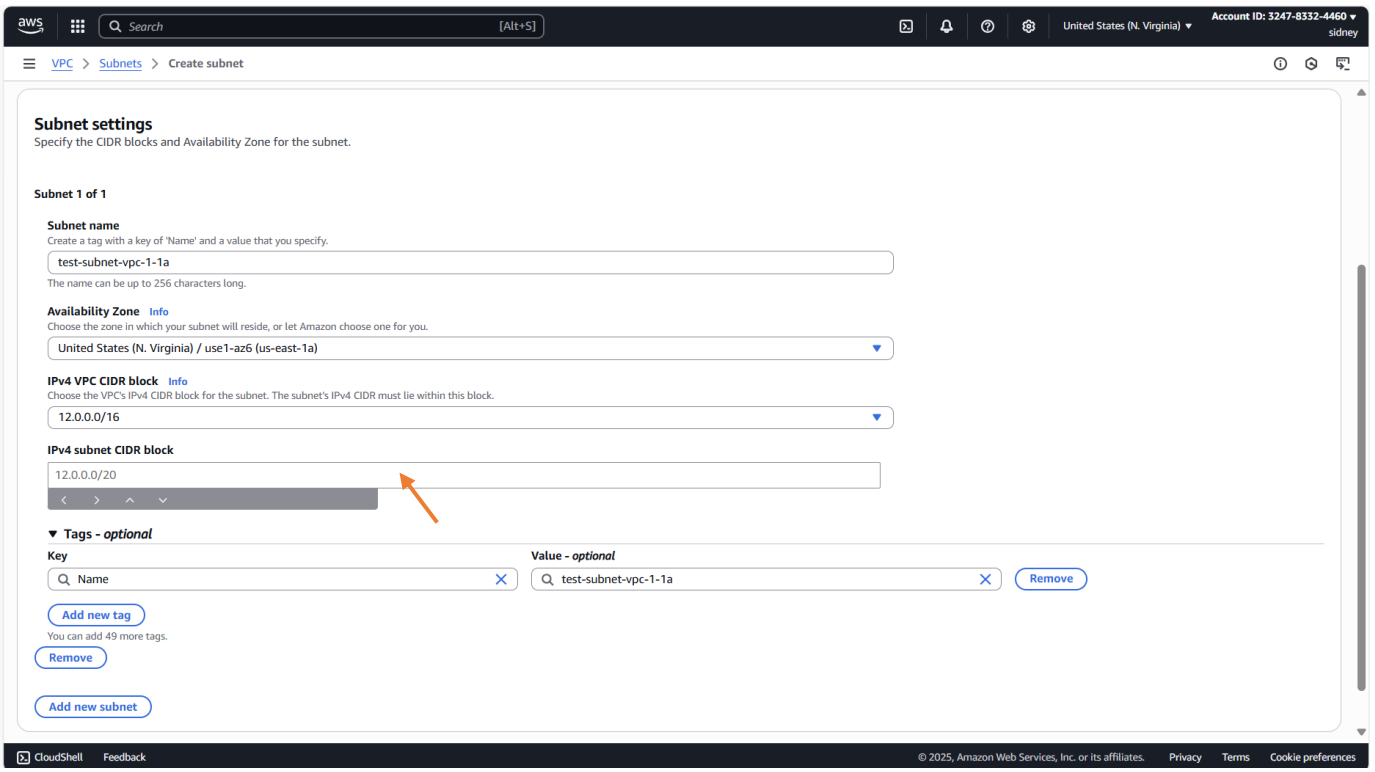
Click on the drop down and select the first VPC "**test-vpc-1**"



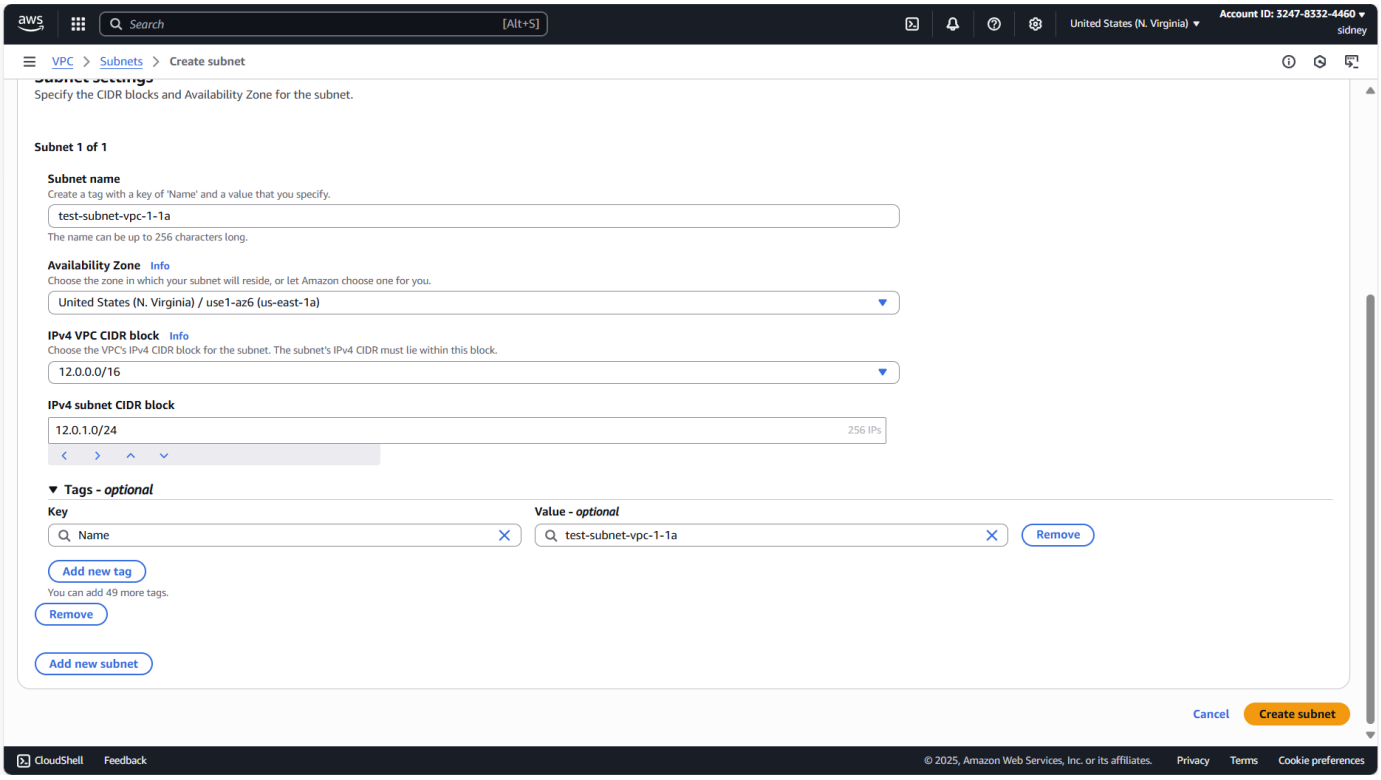
Then give the subnet a name, I will call it "**test-subnet-vpc-1-1a**"



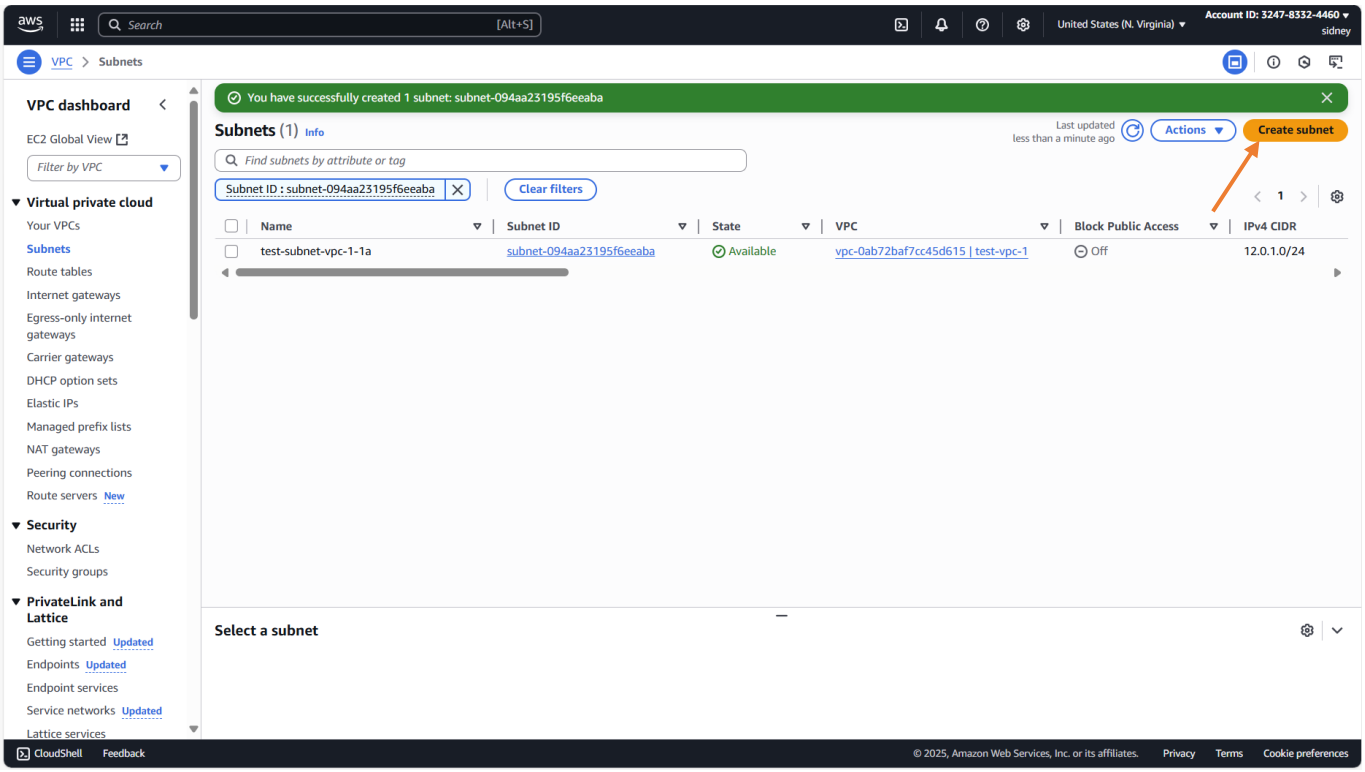
Then click on the drop down on availability zone, and select "**us-east-1a**"



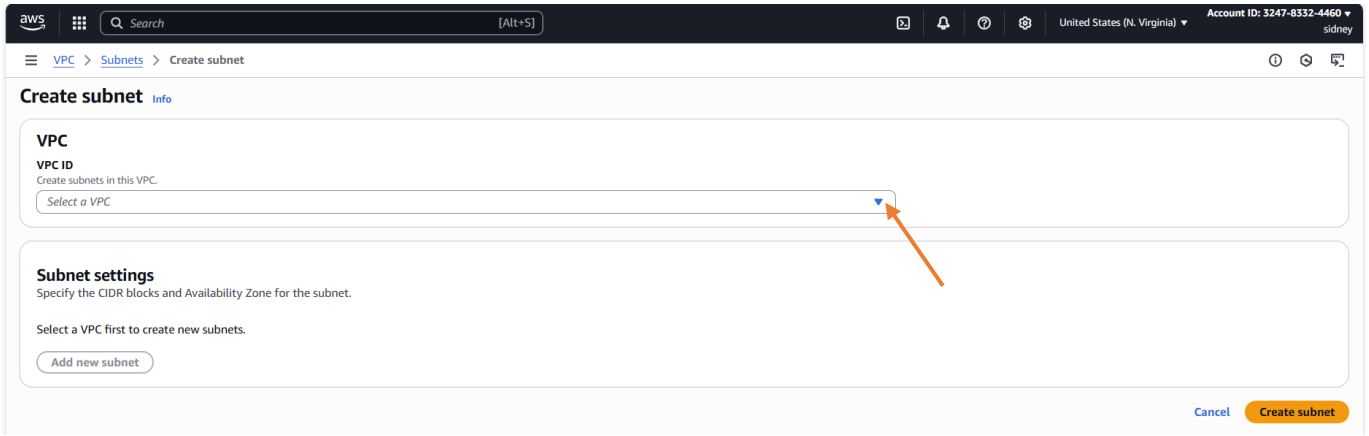
On "**IPv4 subnet CIDR block**", enter "**12.0.1.0/24**"



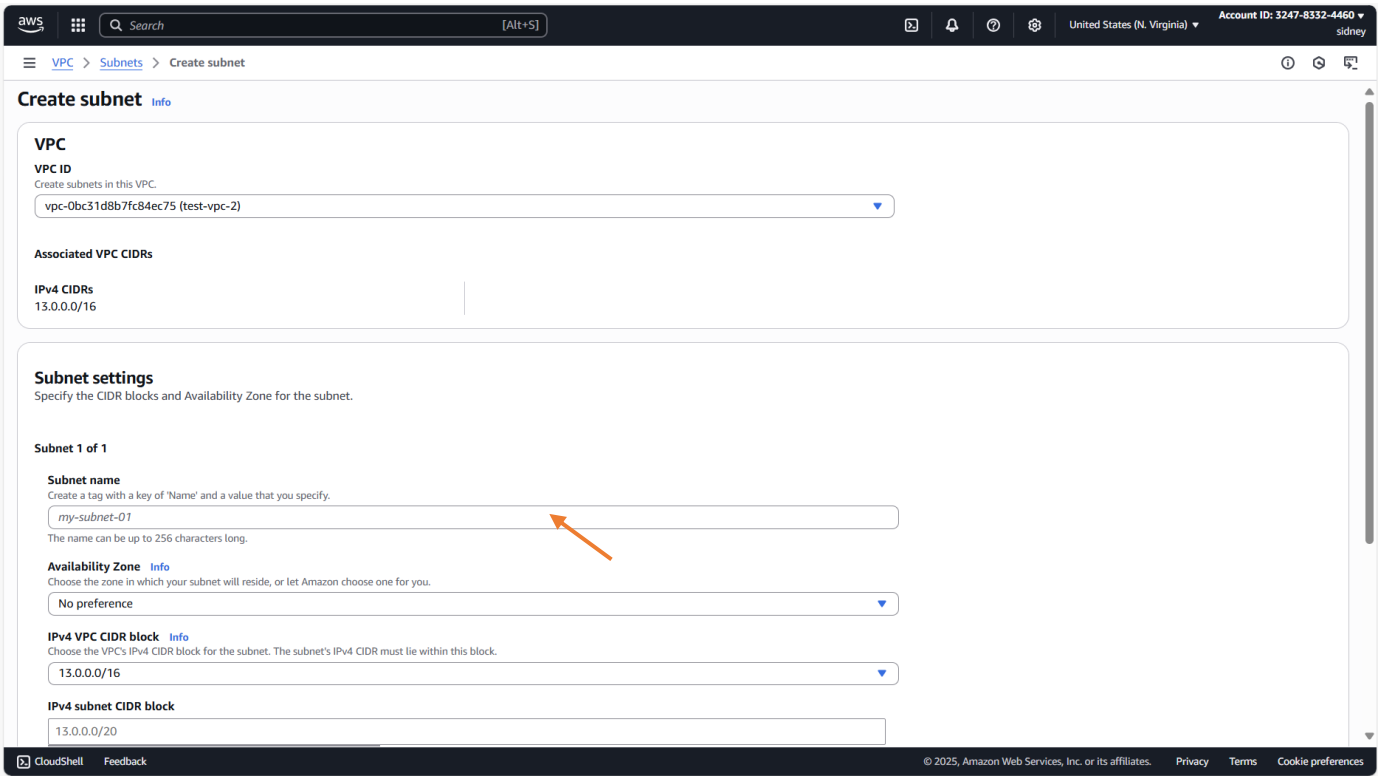
Click on "**create subnet**"



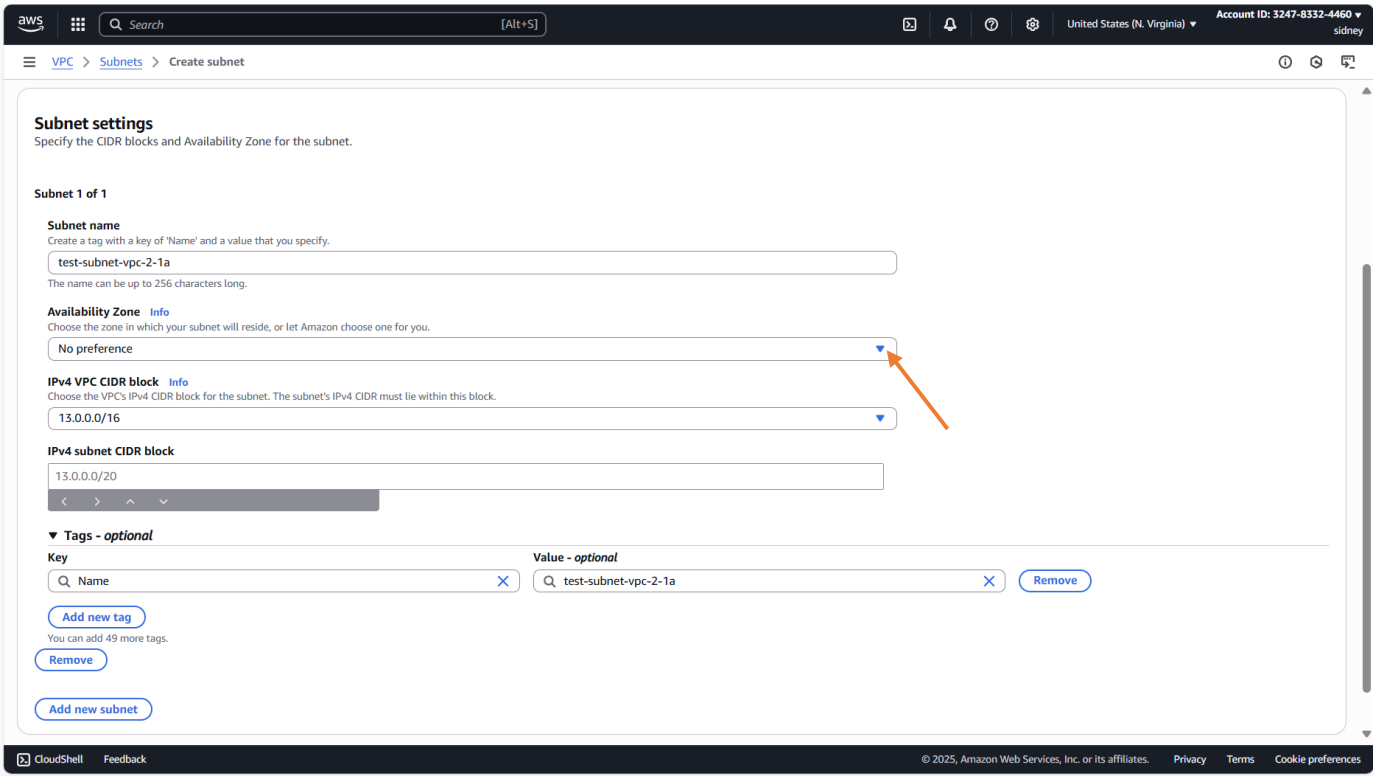
The first subnet for **test-vpc-1** has been created. Let us create the subnet for **test-vpc-2**. Click on "**create subnet**"



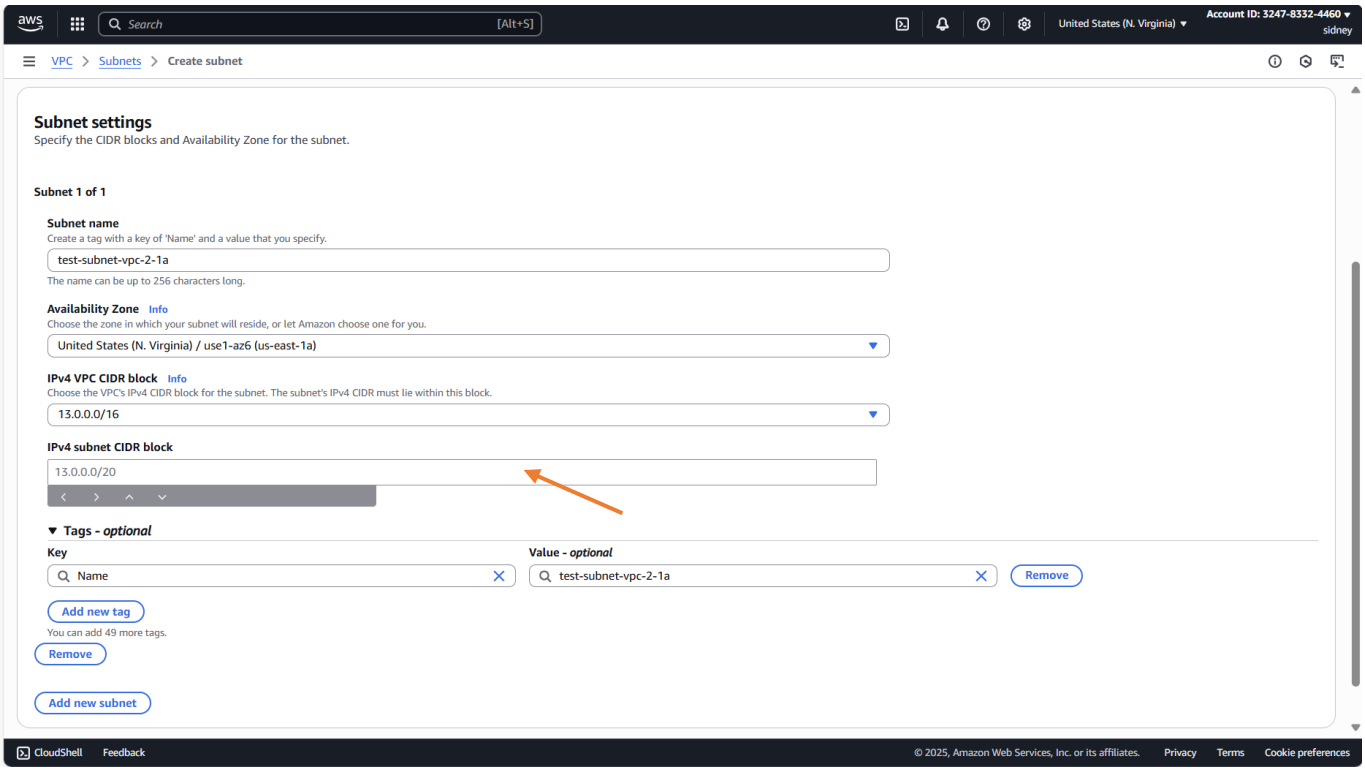
Select our VPC “**test-vpc-2**”



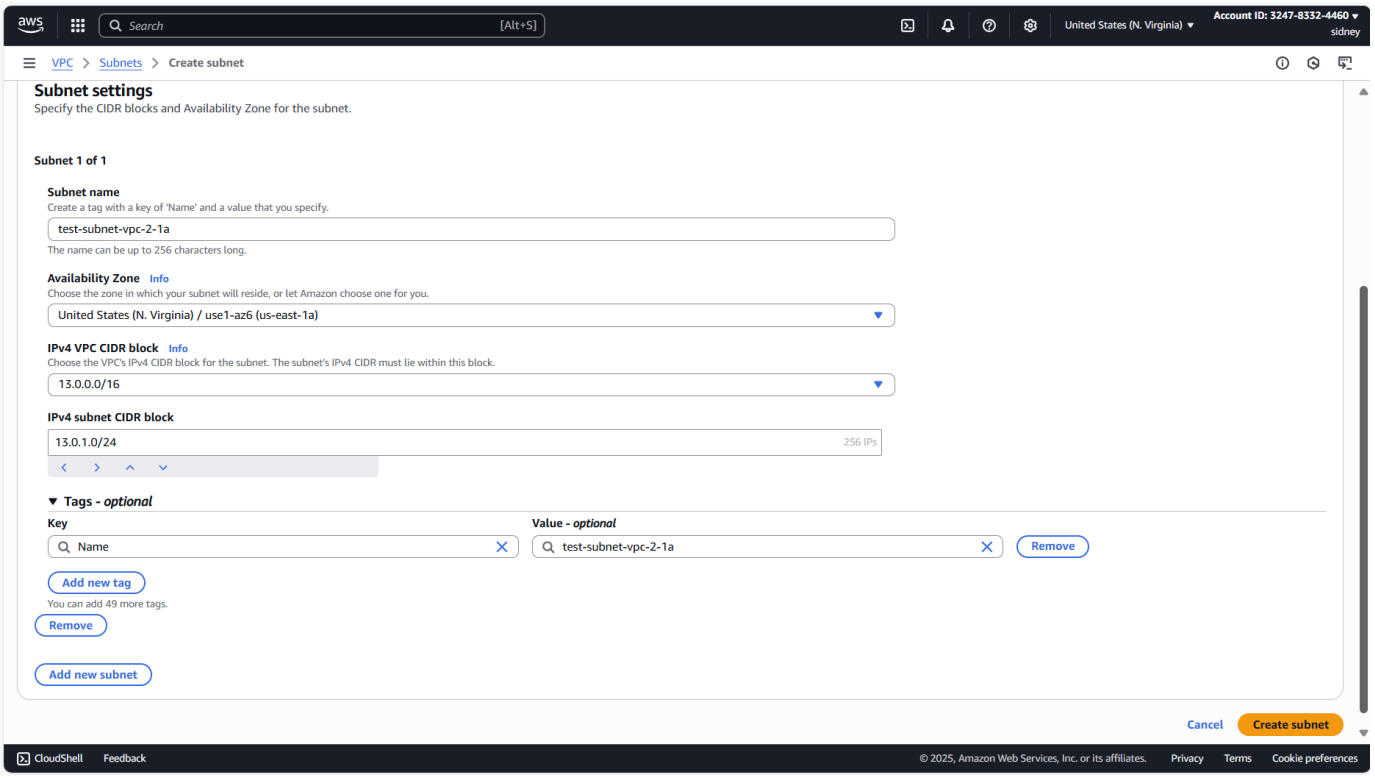
Let us give the VPC a name, I will call it “**test-subnet-vpc-2-1a**”



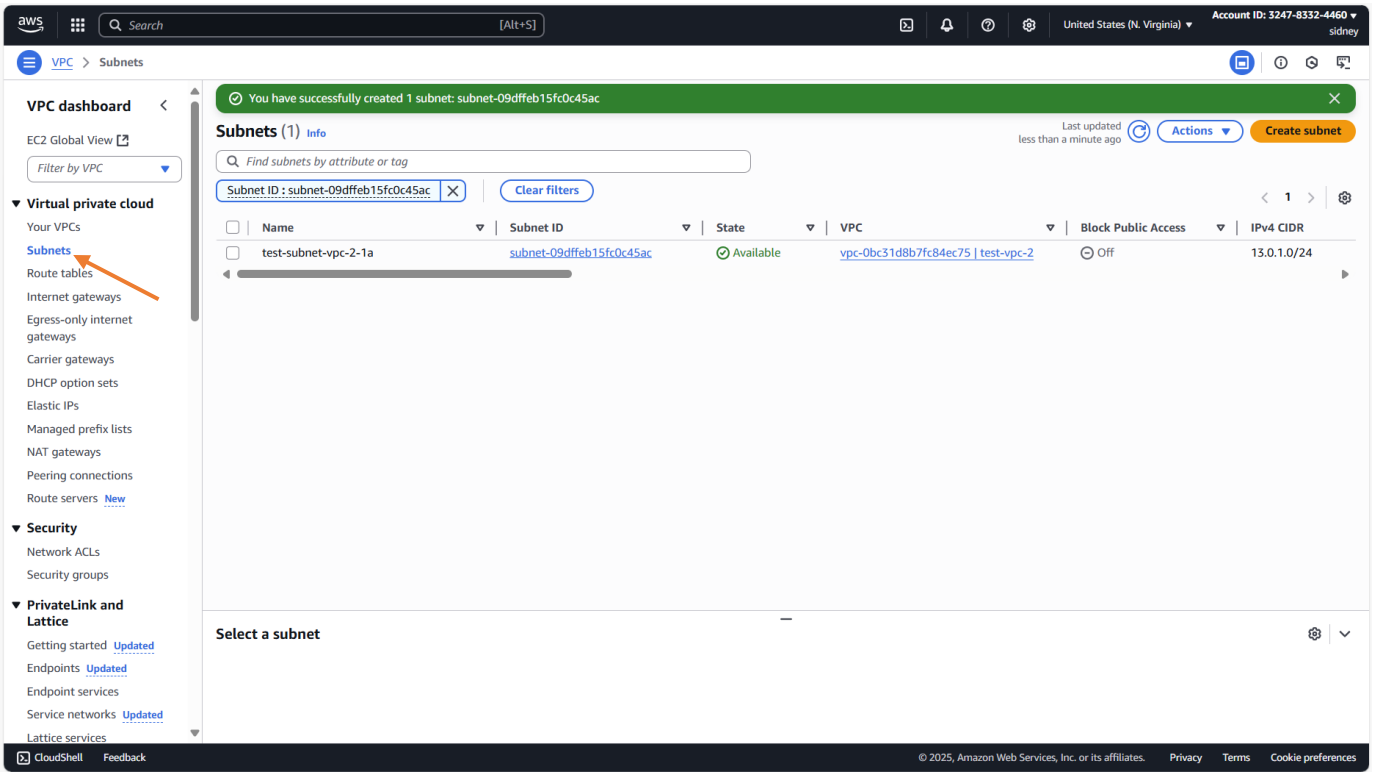
Then on "**availability zone**", click on the drop down and select "**us-east-1a**"



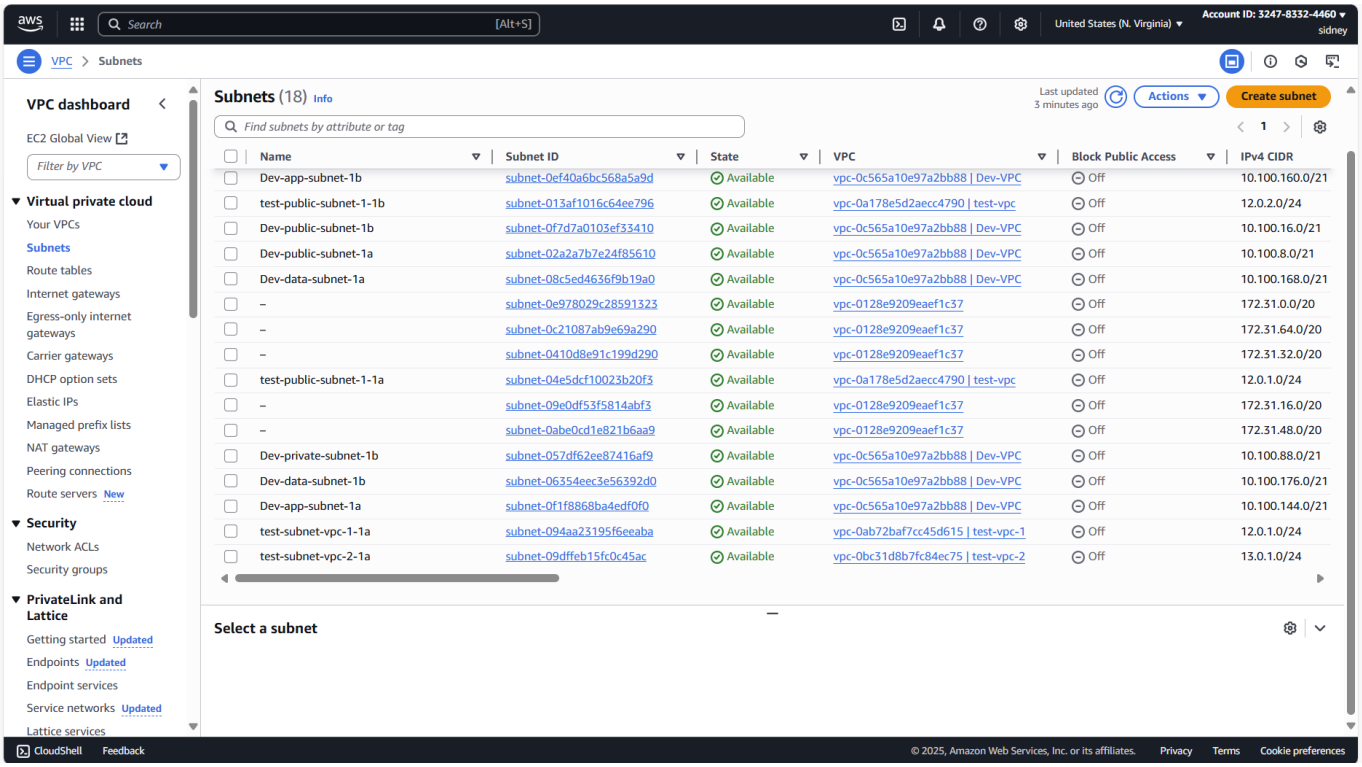
On "**IPv4 subnet CIDR block**", enter "**13.0.1.0/24**"



Click on "**Create subnet**"

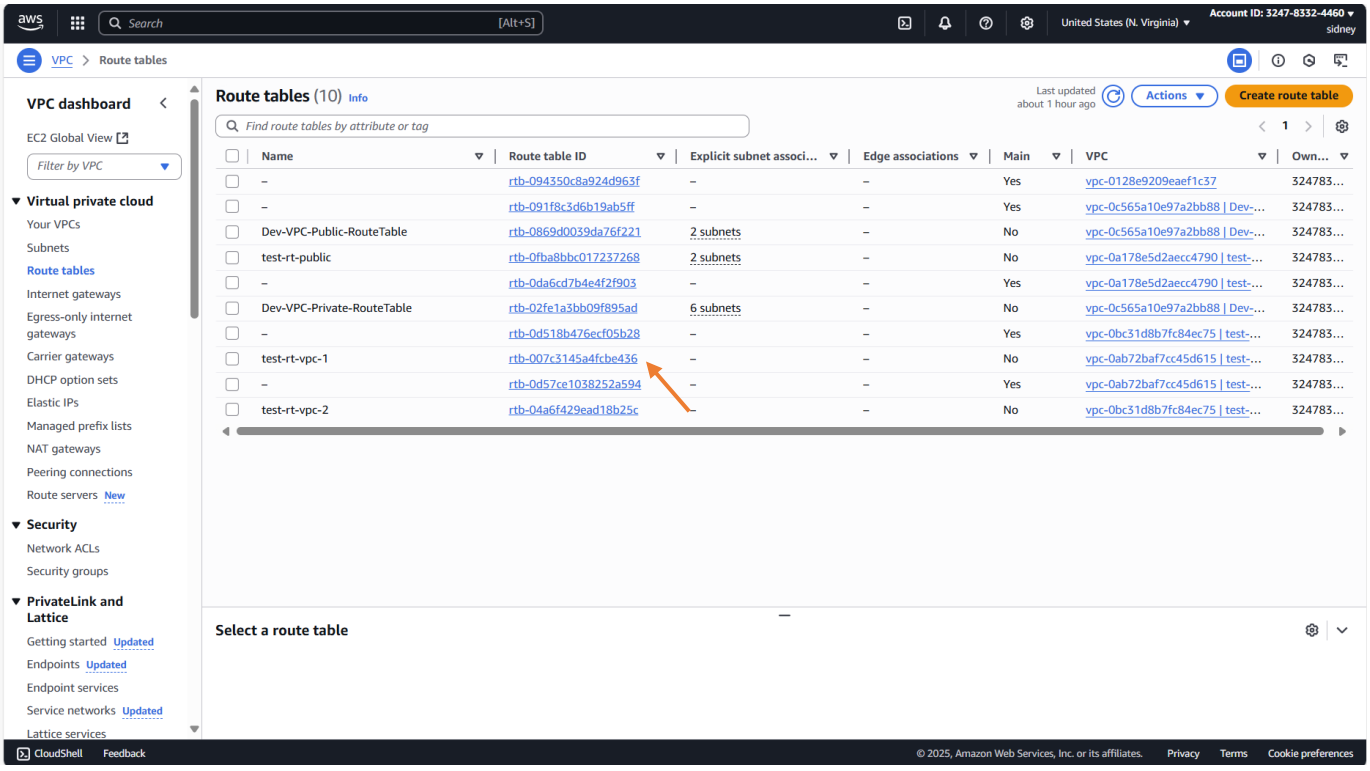


The subnet has been created. Click on "**Subnets**"

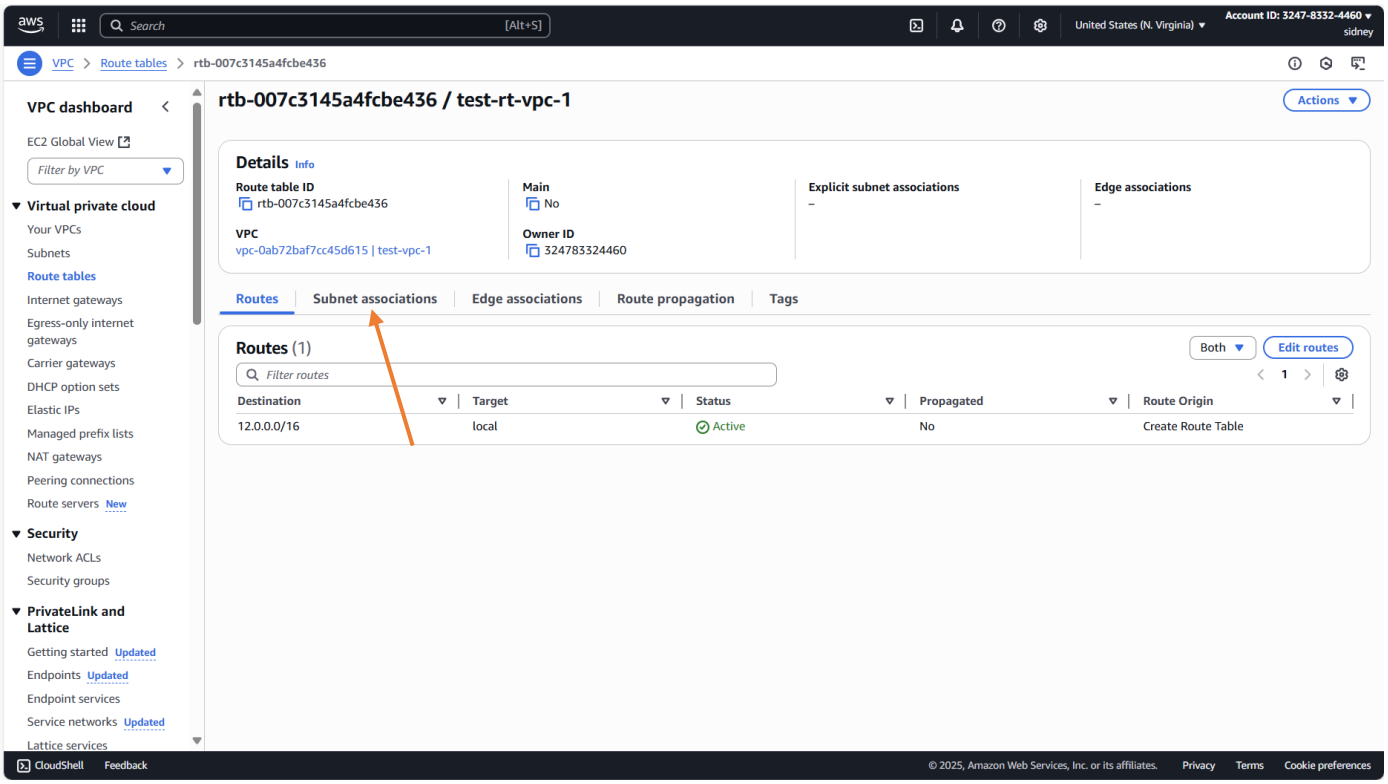


You can see the two subnets we have created.

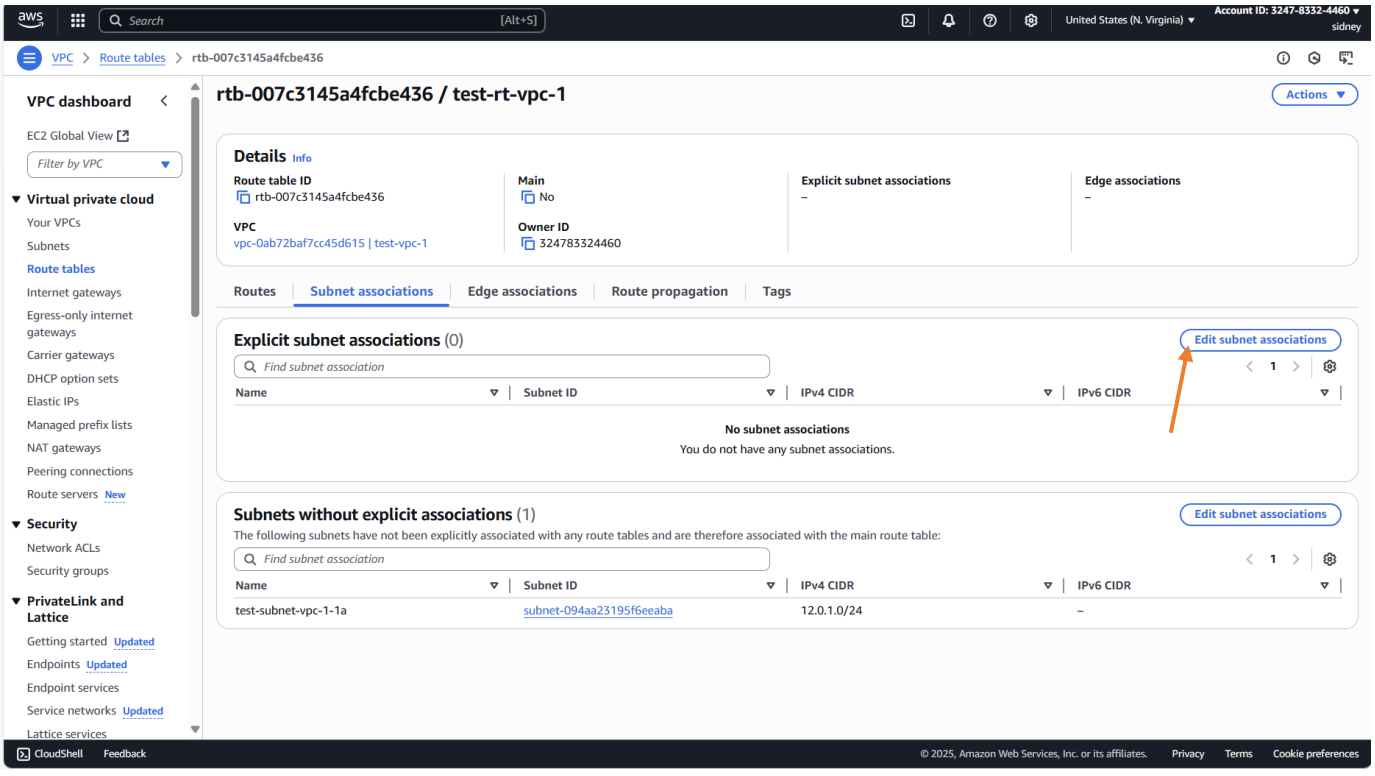
The next thing we have to do is to associate our route tables with our subnets. Go to the Route tables



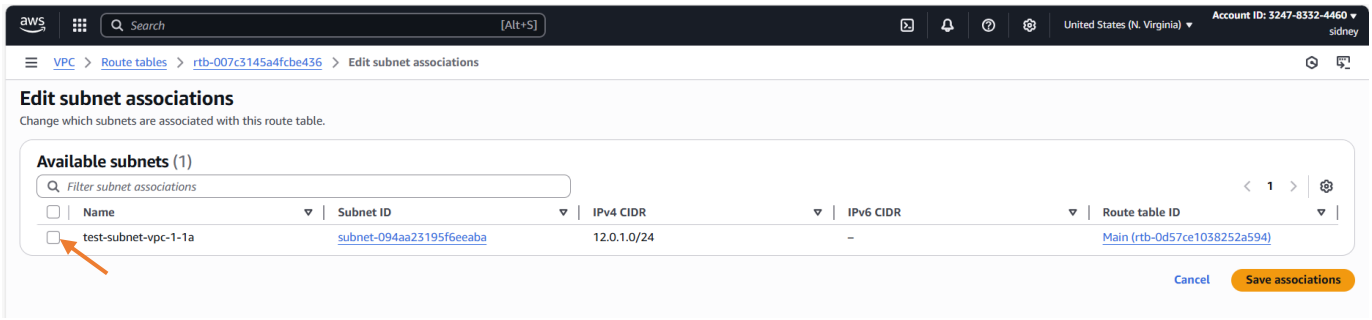
Click on the first route table “**test-rt-vpc-1**”



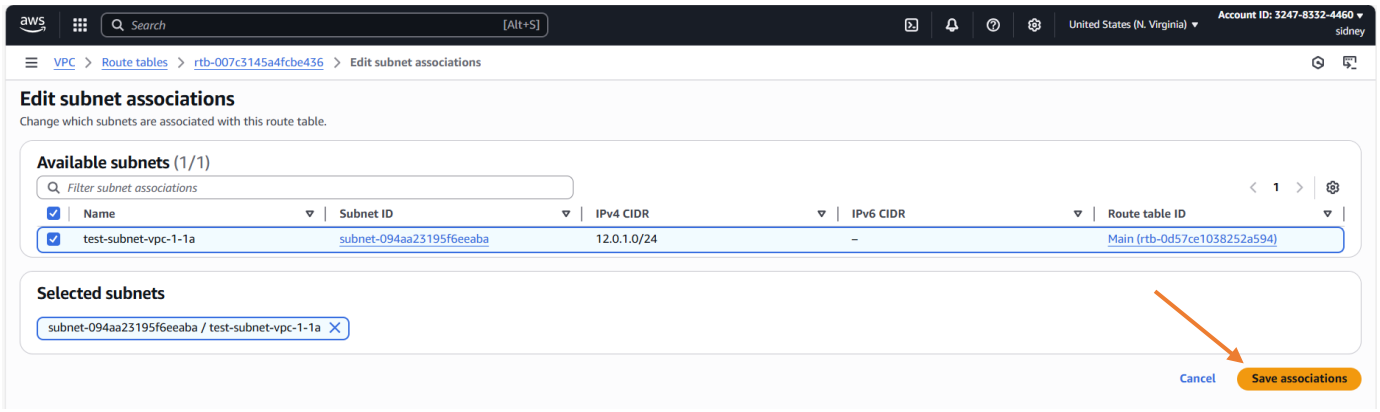
Click on the "**subnet Association**" tab



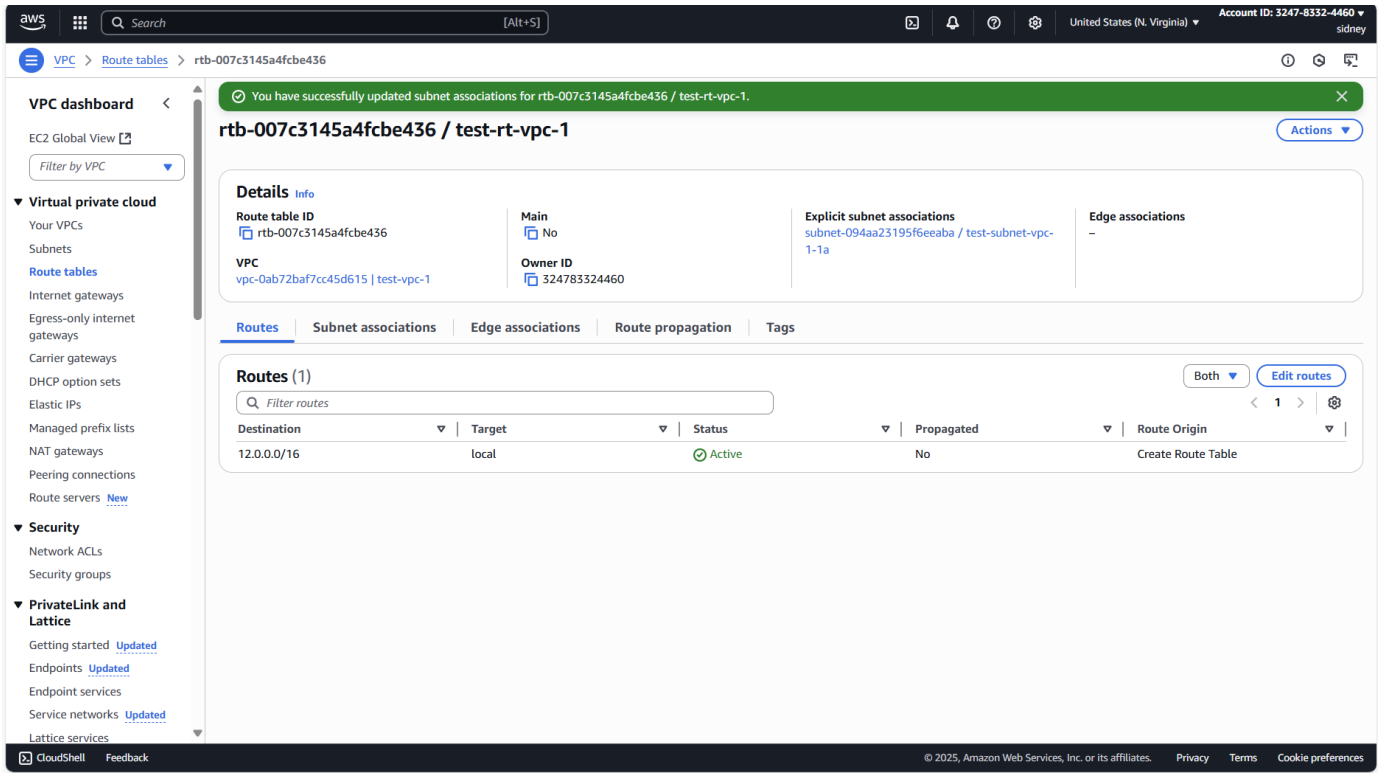
Click on "**Edit Subnet Associations**"



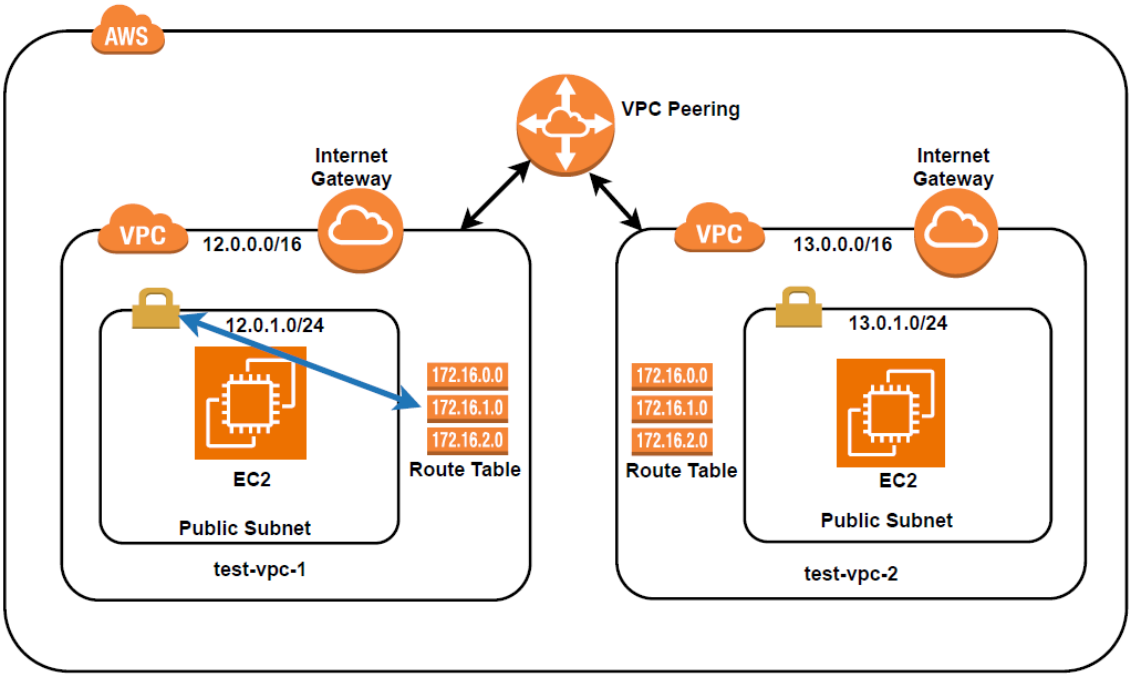
Select the subnet



Click on “**Save Associations**”

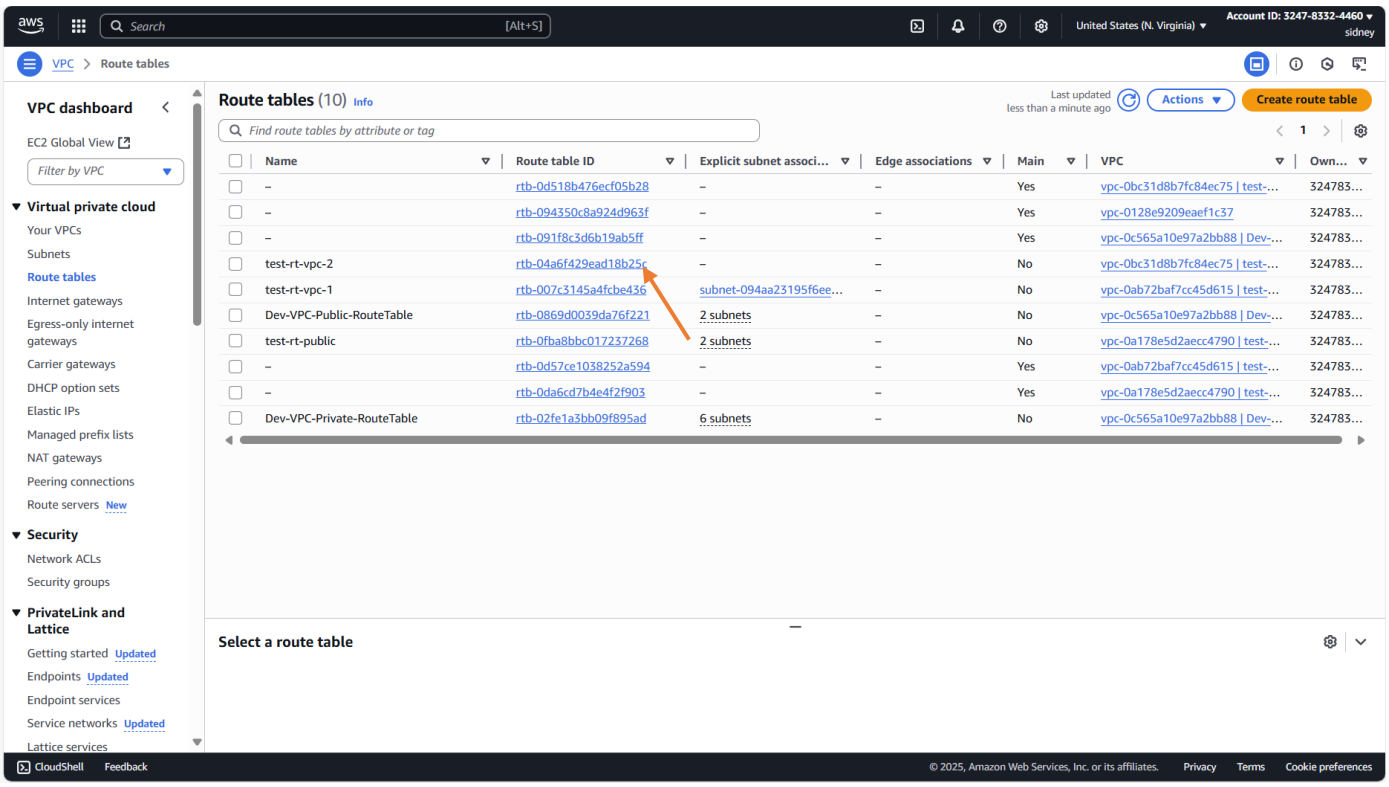


The route table has been associated with the subnet

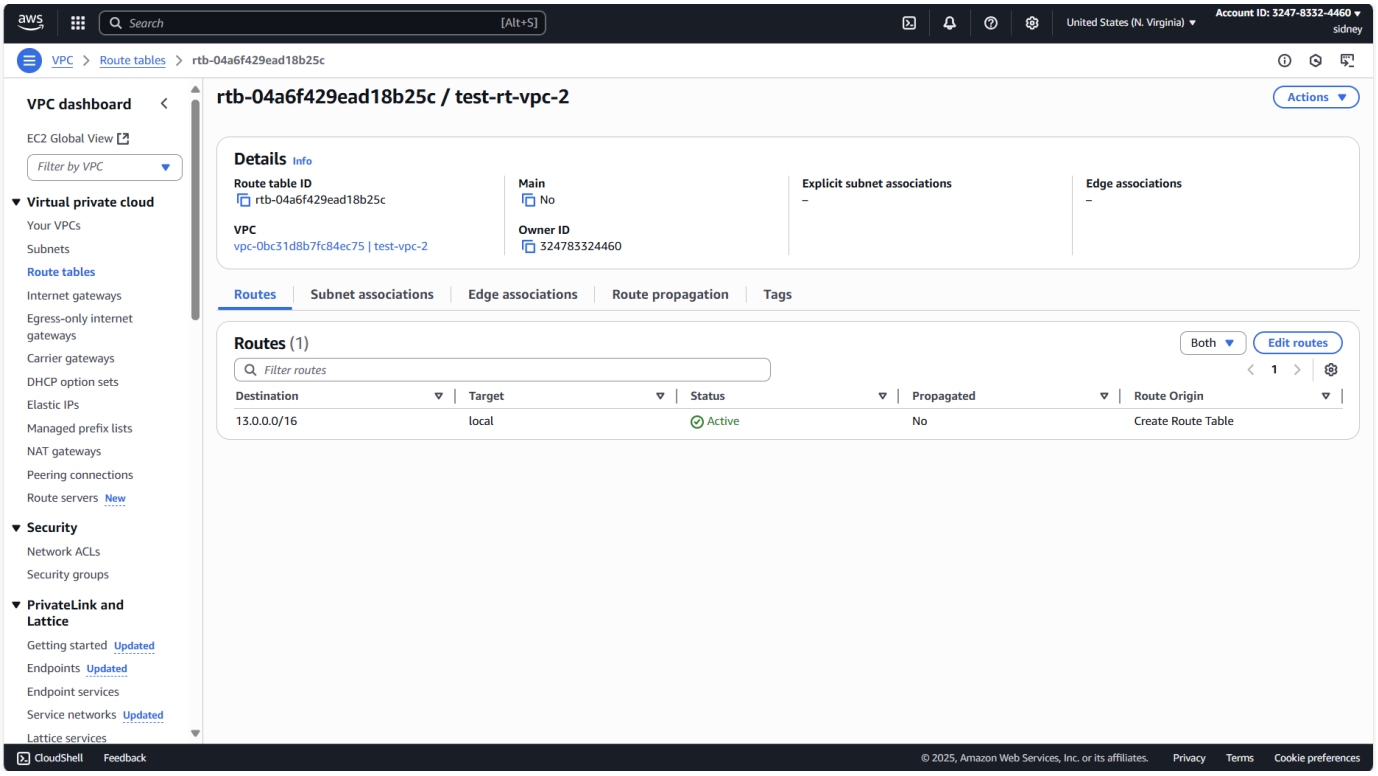


Our route table has been associated with our subnet in **test-vpc-1**.

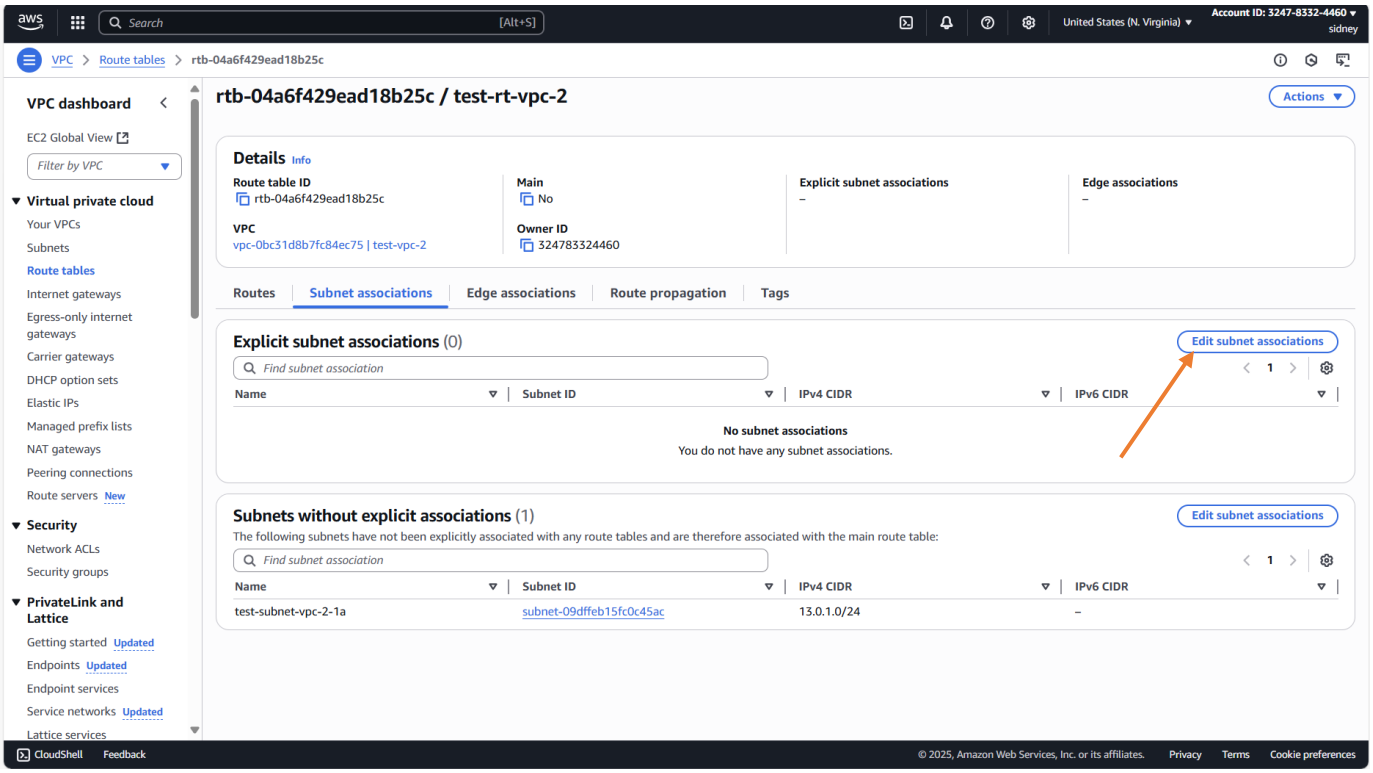
Let us associate the second route table to our subnet in test-vpc-2. Click on "**Route Tables**"



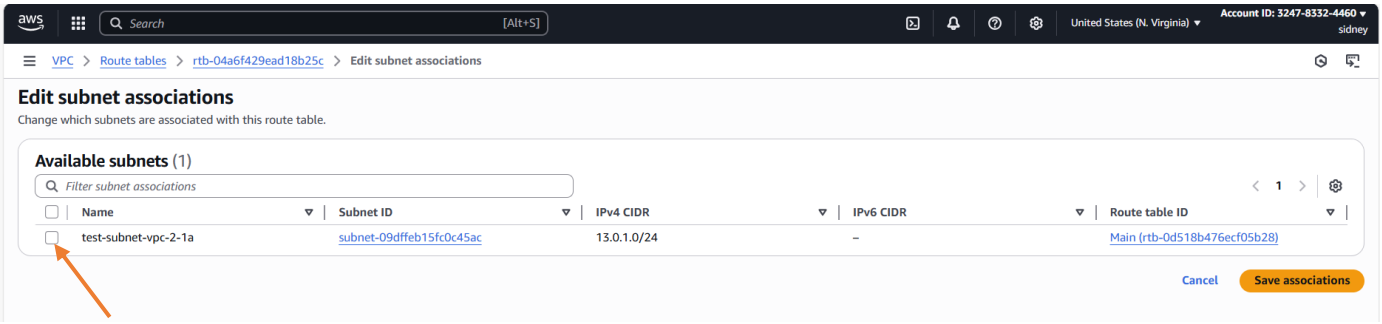
Click on the "**Route Table ID**" of the VPC "**test-vpc-2**", that is "**test-rt-vpc-2**"



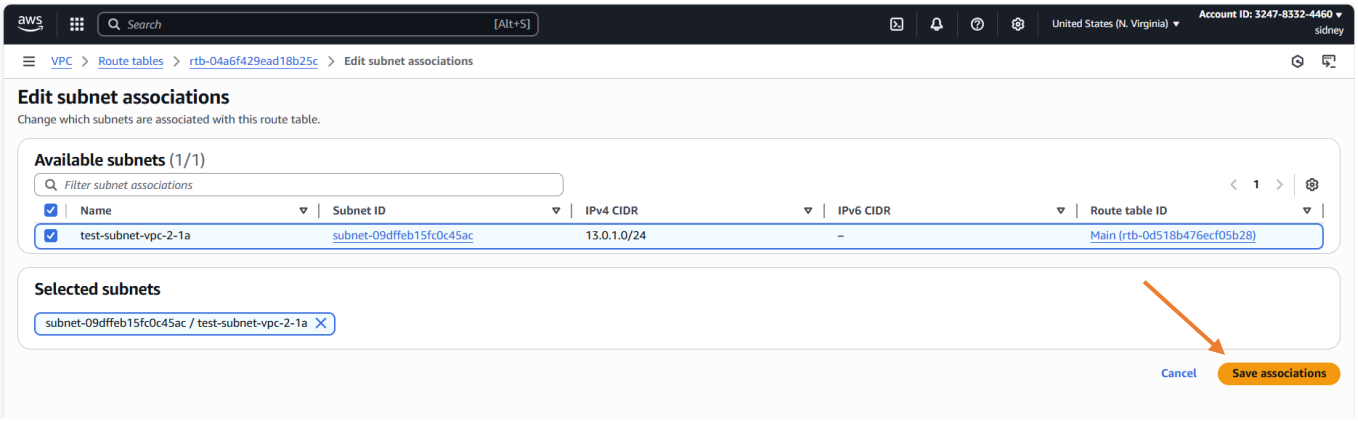
Click on the "**Subnet Associations**" tab



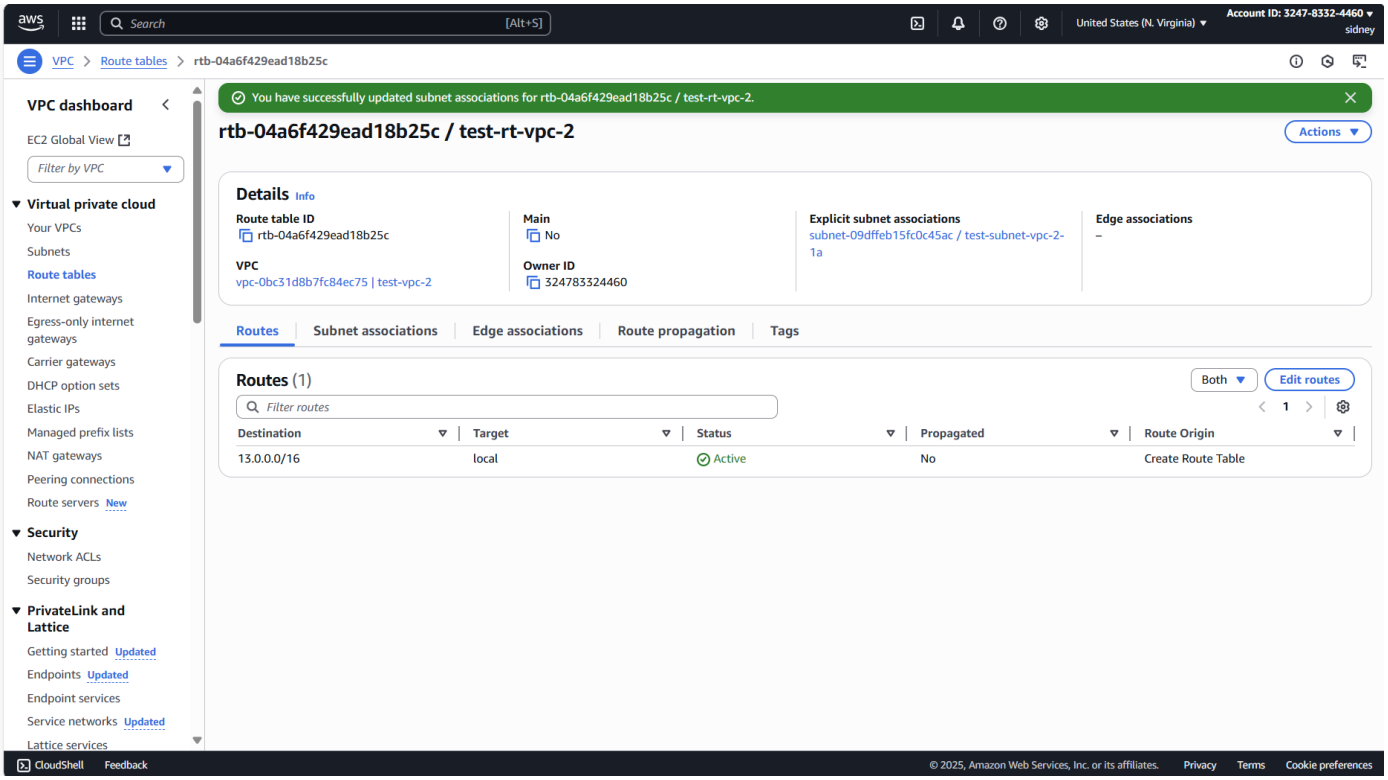
Click on "**Edit Subnet Associations**"



Select the subnet



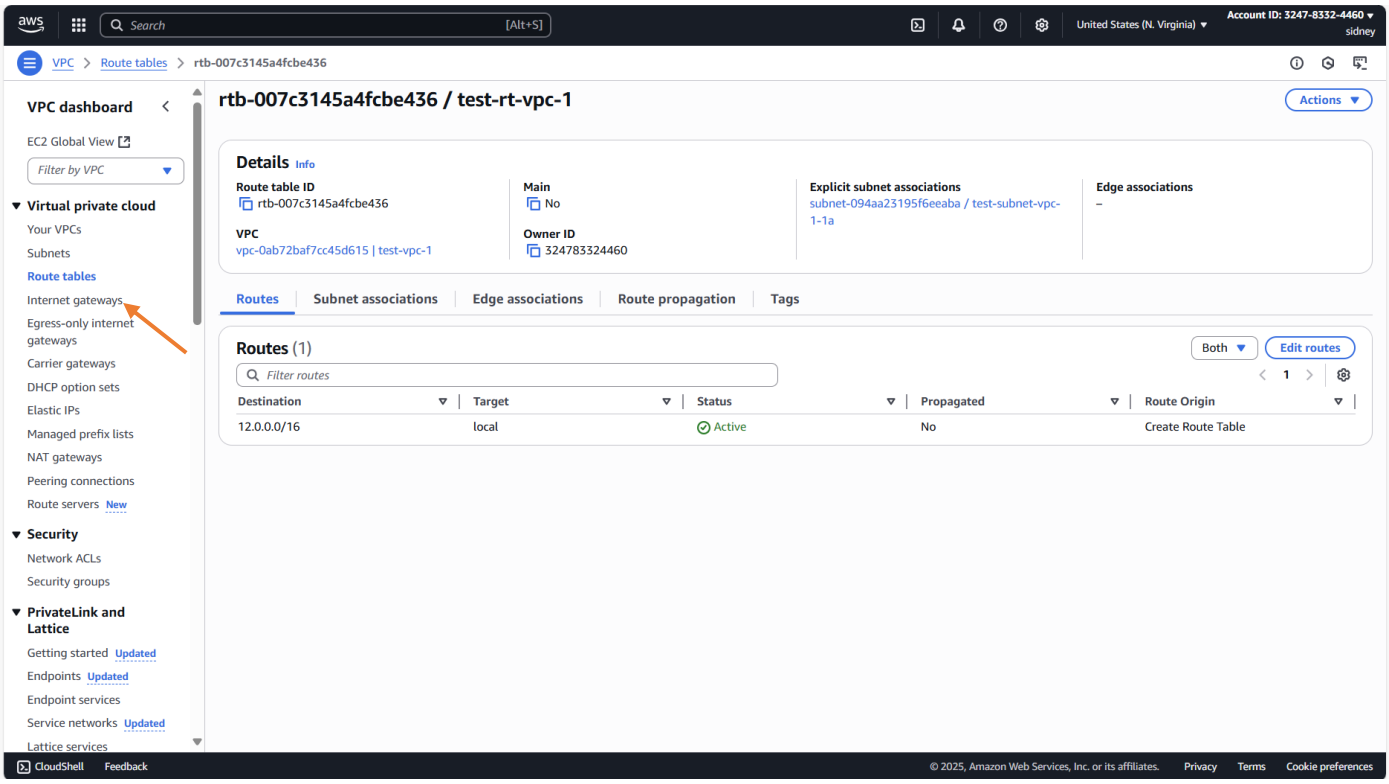
Click on “**Save Associations**”



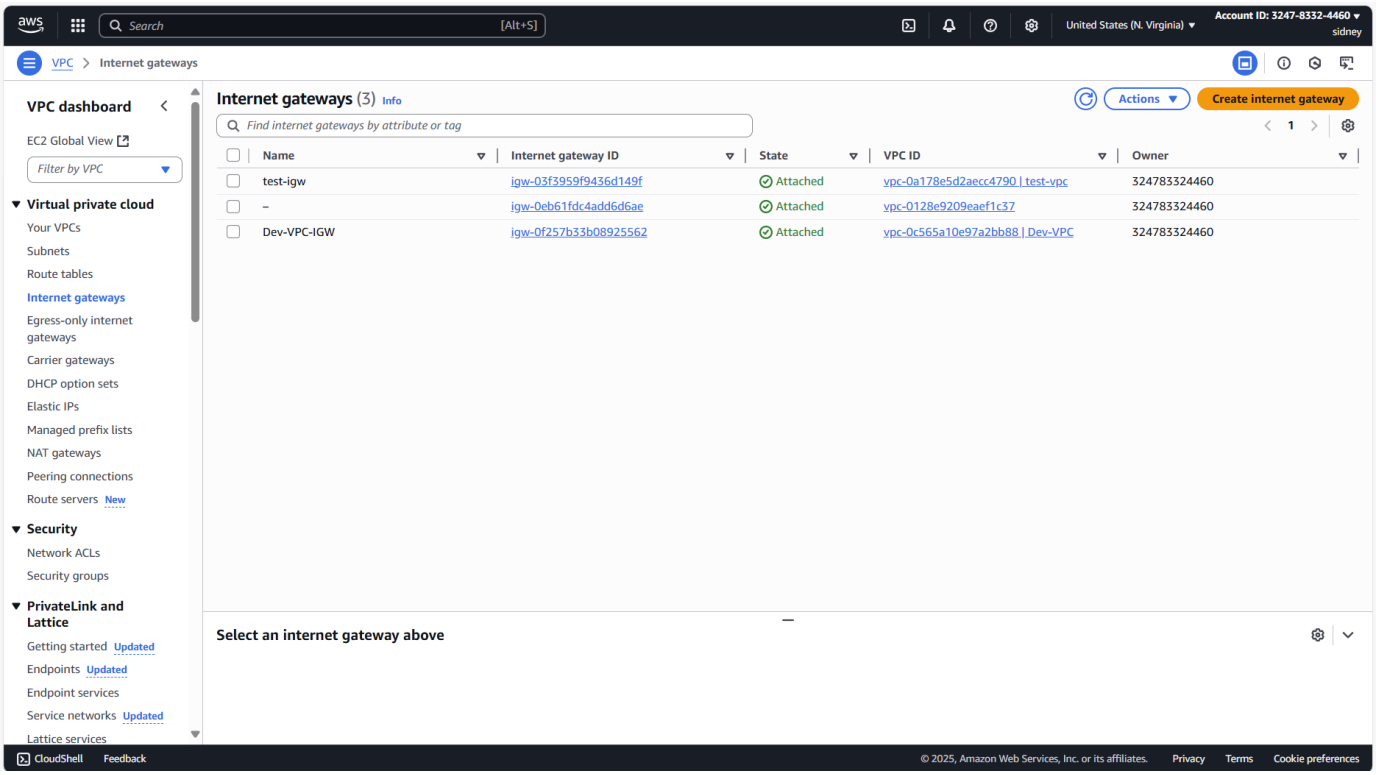
We have created two Route tables and associated them with their respective VPCs. But also in the route tables, we need to grant the internet access. To do that we need the **Internet Gateway**.

Part 6: Create Internet Gateway

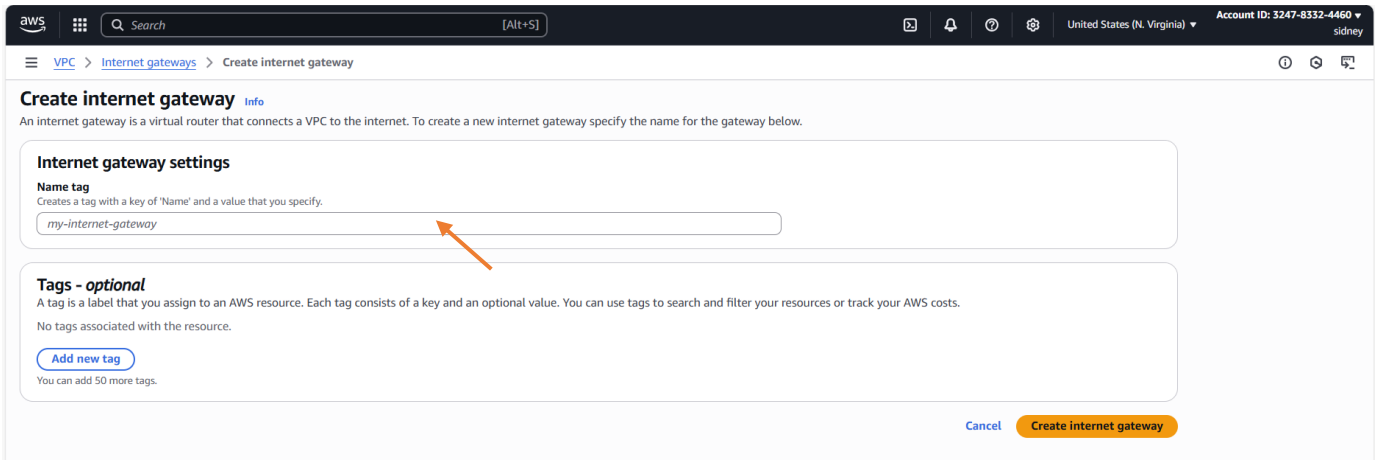
We have to create internet gateway for both VPCs. Let us create the Internet Gateway for the first VPC.



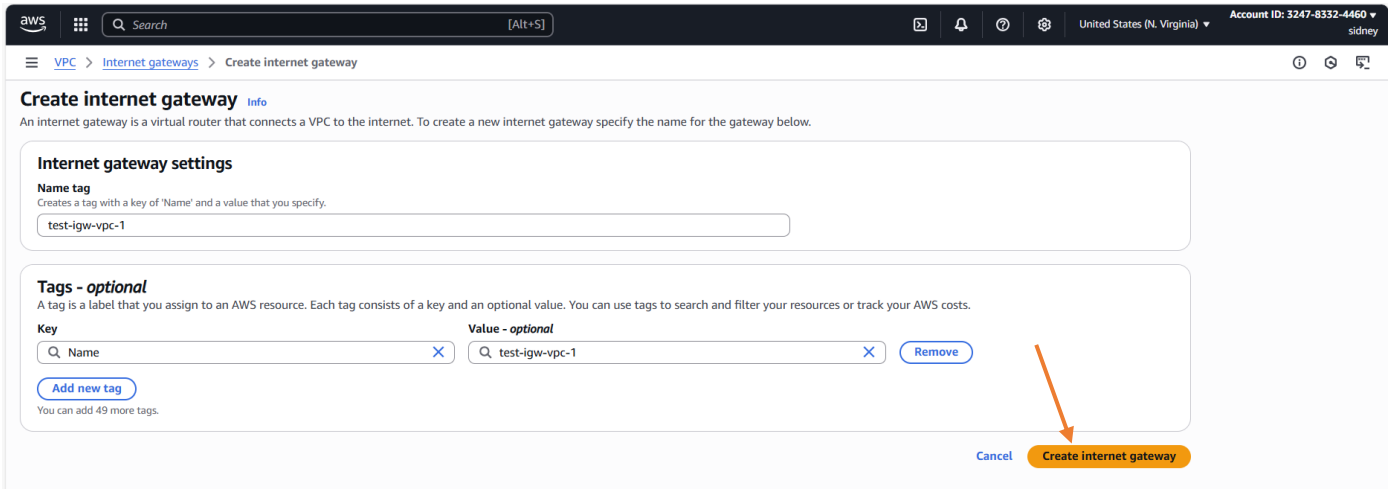
Click on "**Internet Gateways**"



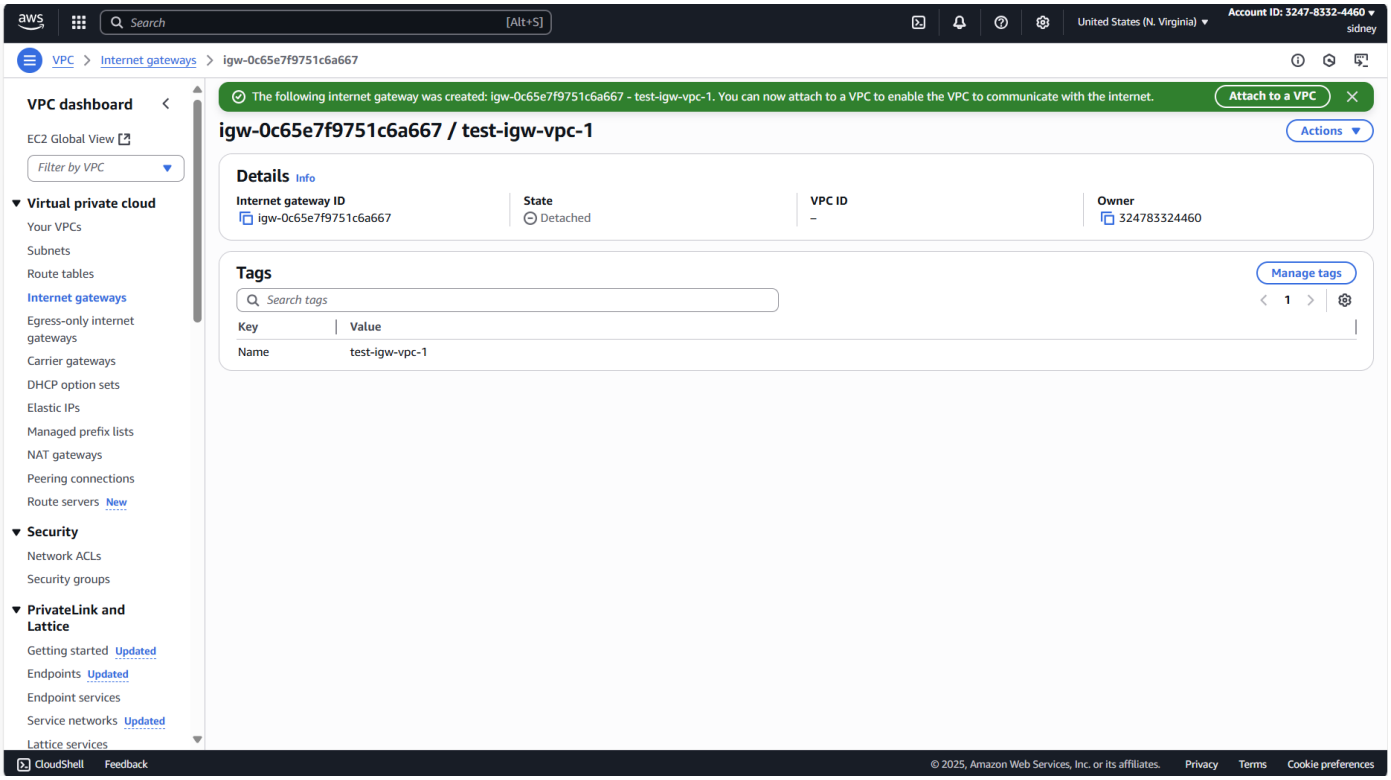
Click on “**Create Internet Gateway**”



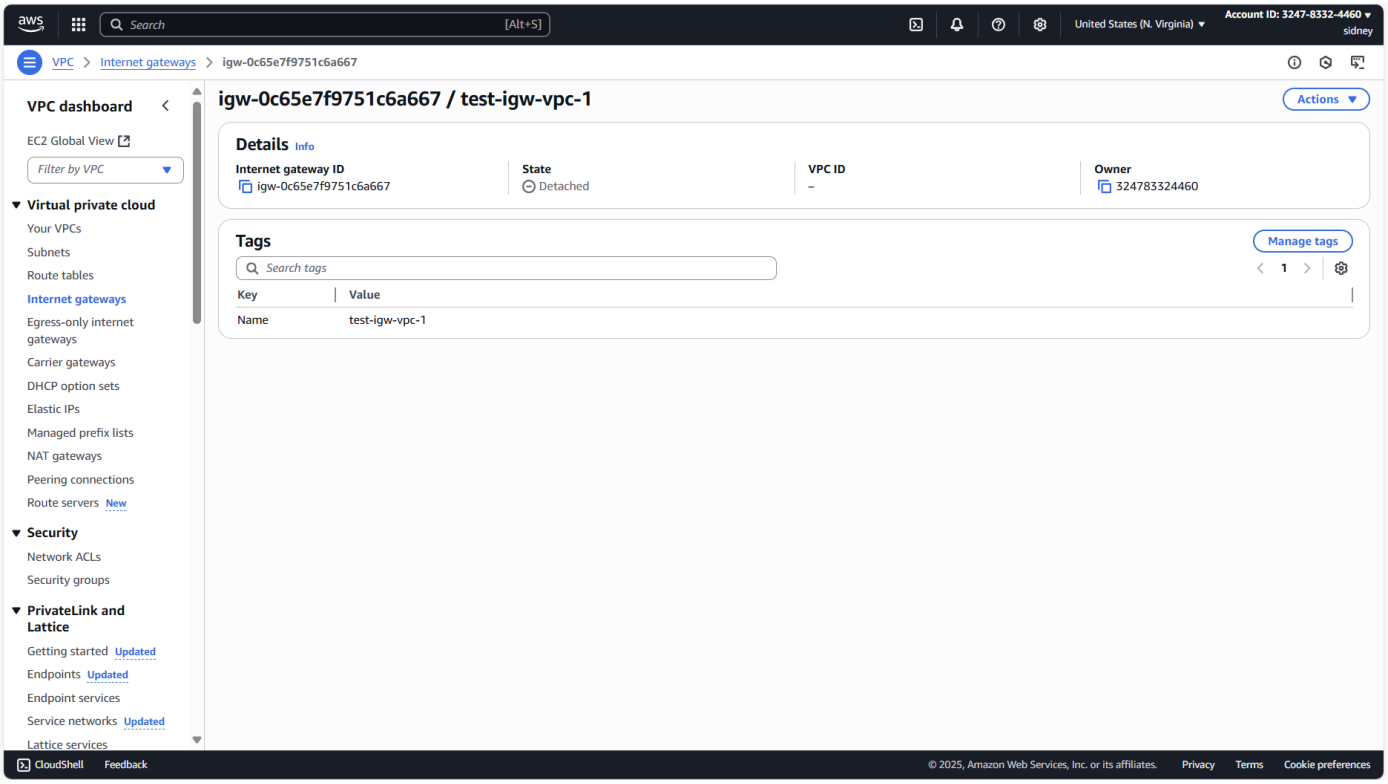
Give the Internet Gateway a name, I will call it “**test-igw-vpc-1**”



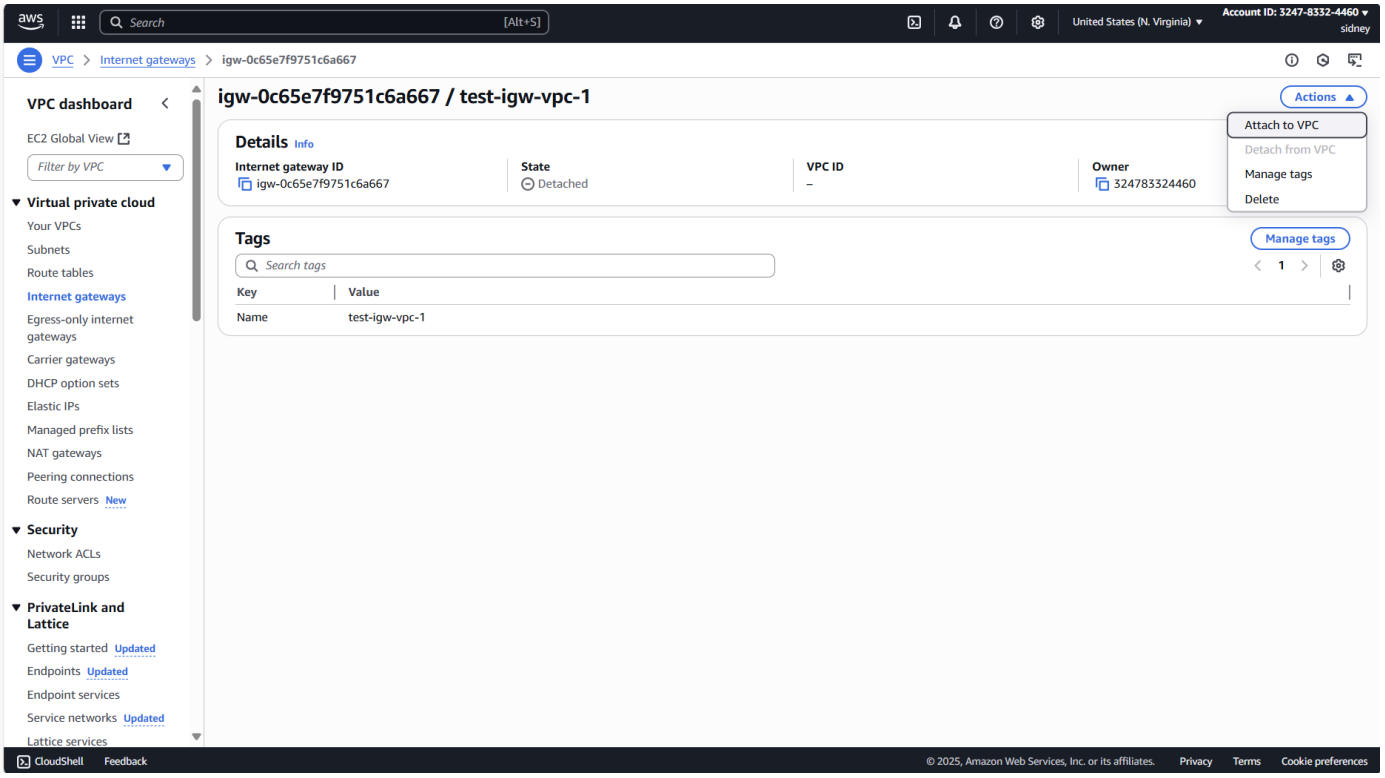
Click on “**create Internet Gateway**”



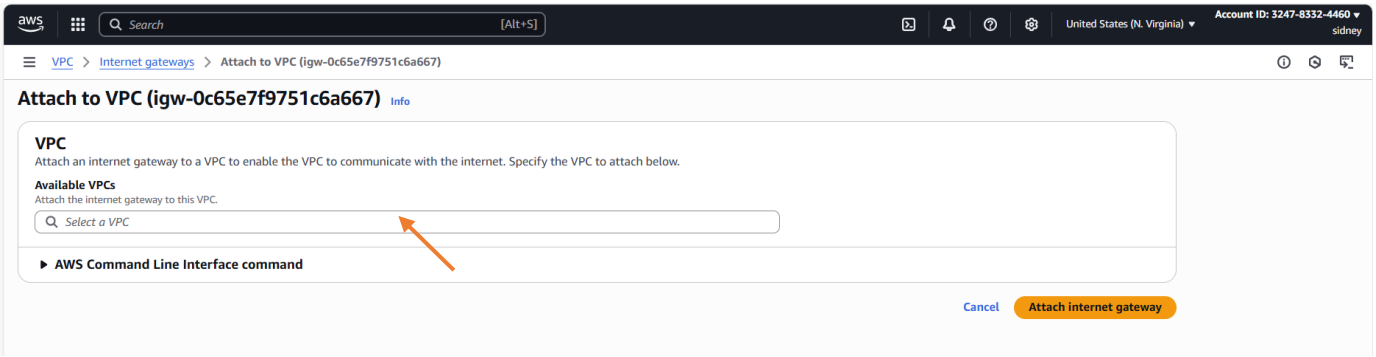
The internet gateway has been created but it is not yet associated with our VPC. Let us associate the Internet Gateway with our VPC.



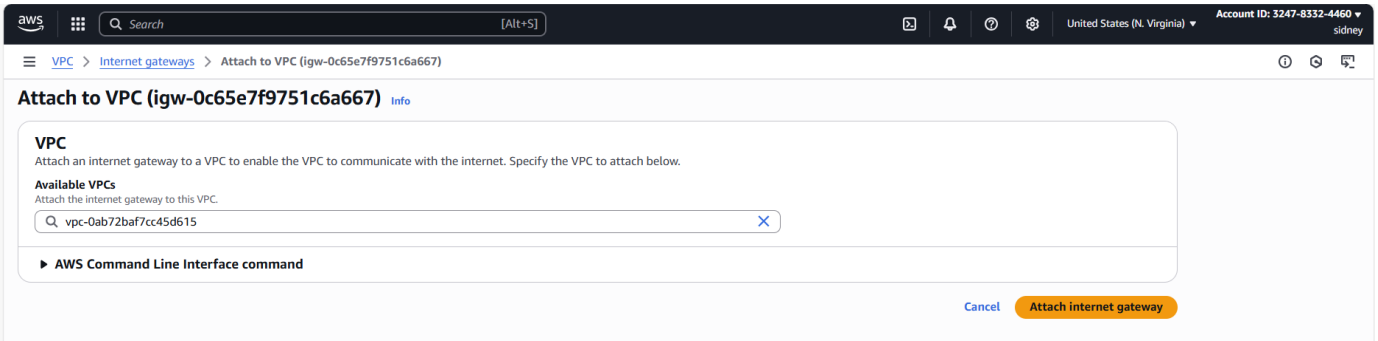
Now, we have to attach the Internet Gateway to the VPC. Click on the drop down on "**Actions**"



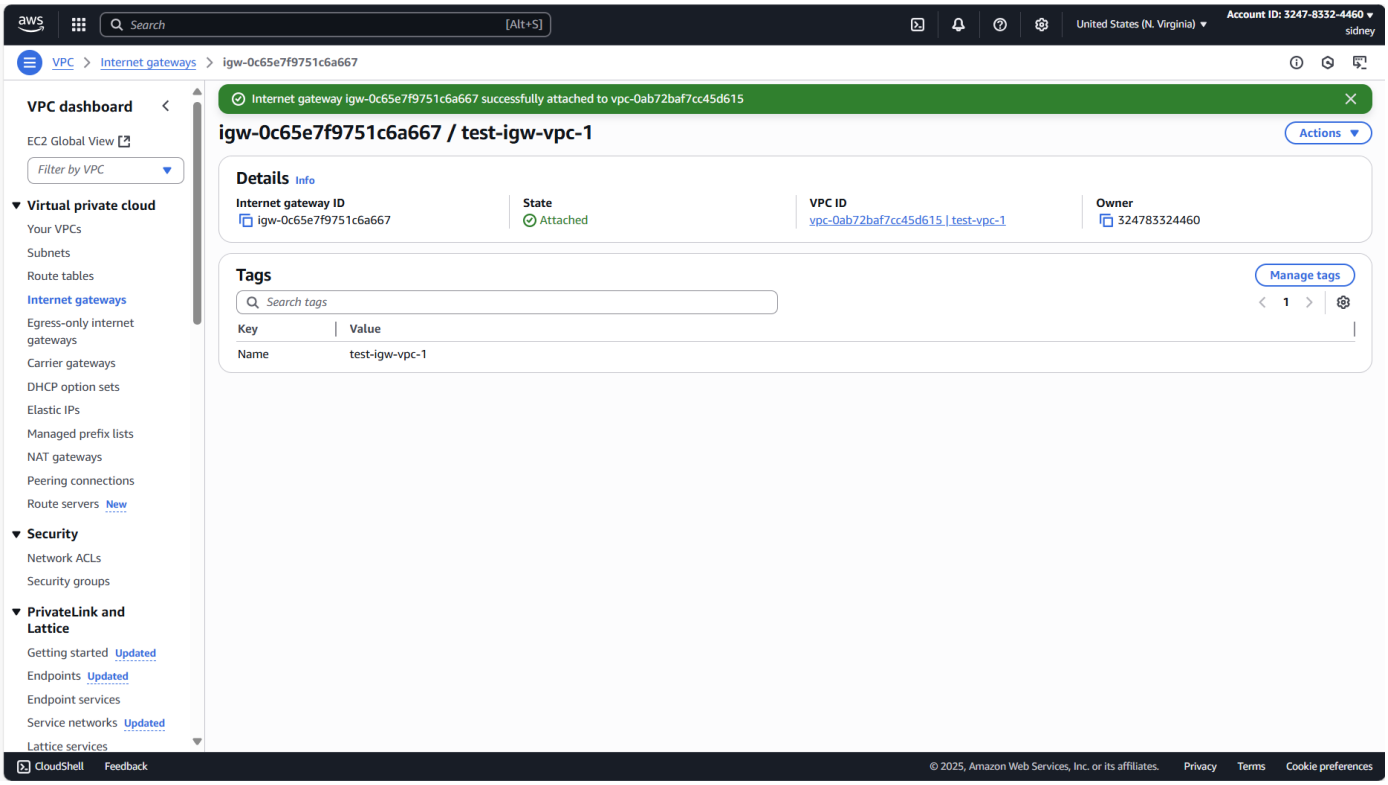
Select “**Attach to VPC**”



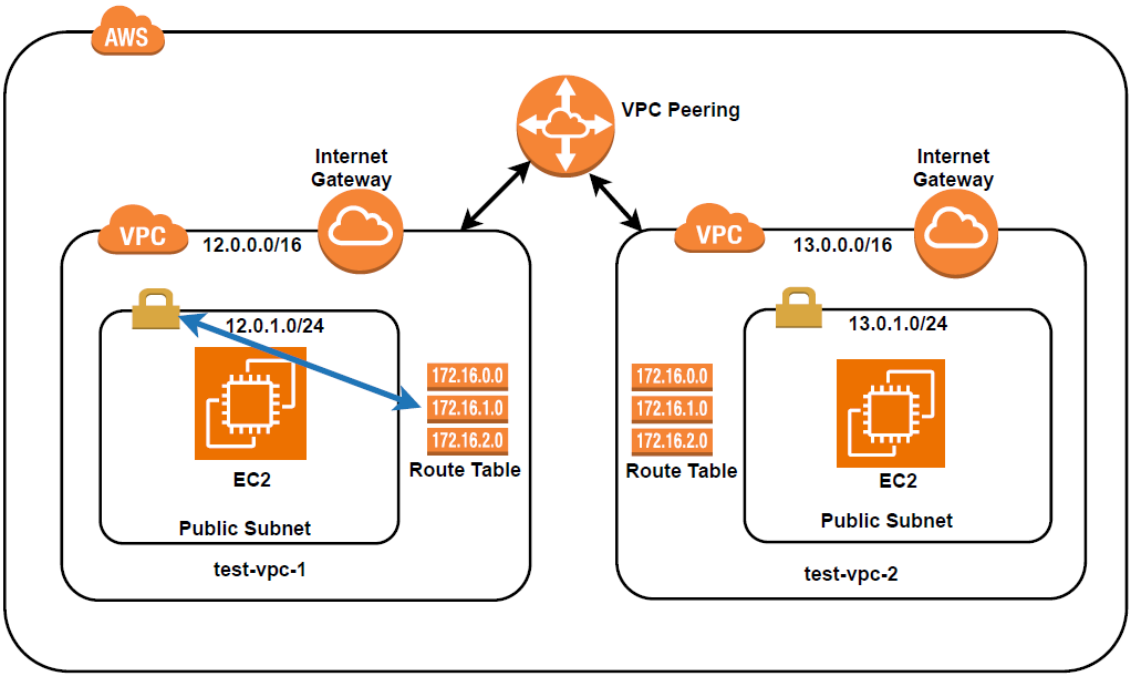
Select our VPC “**test-vpc-1**”



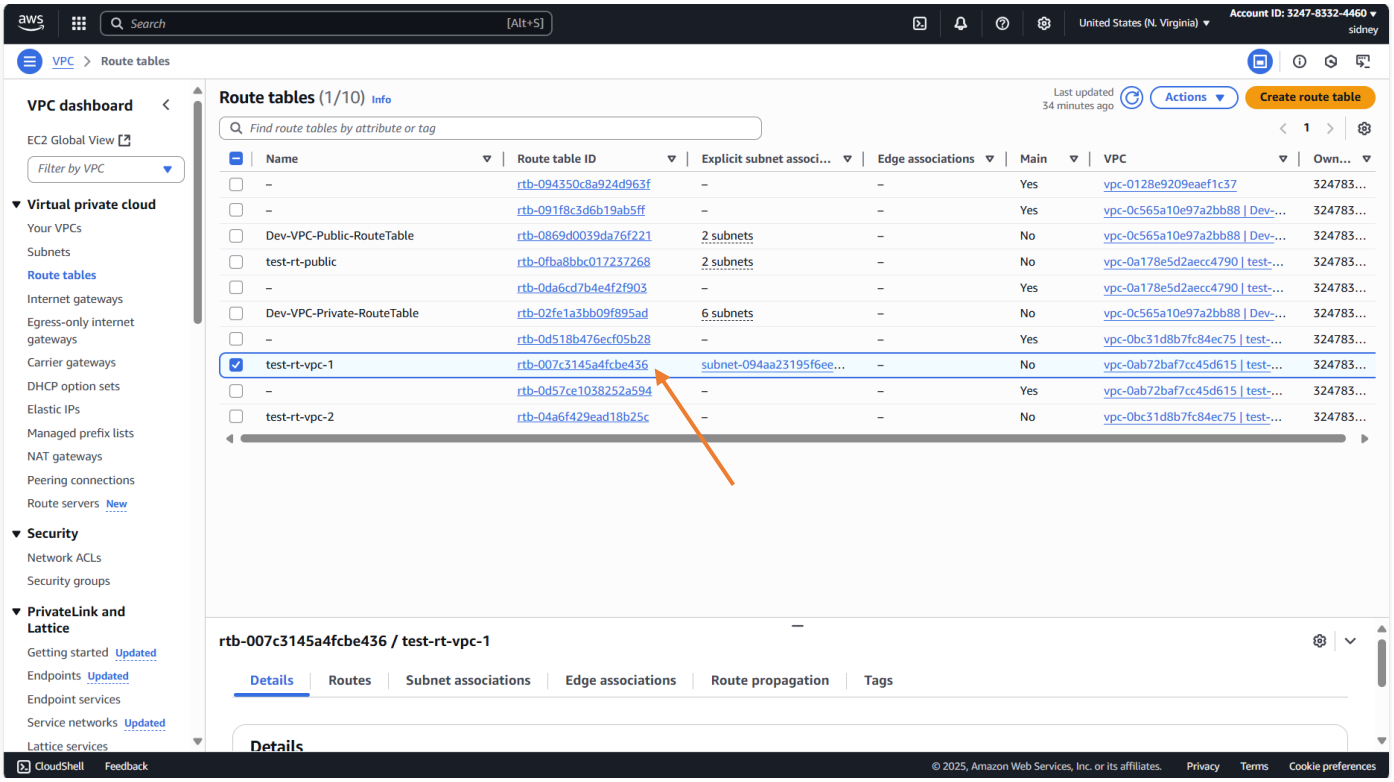
Click on “**Attach Internet Gateway**”



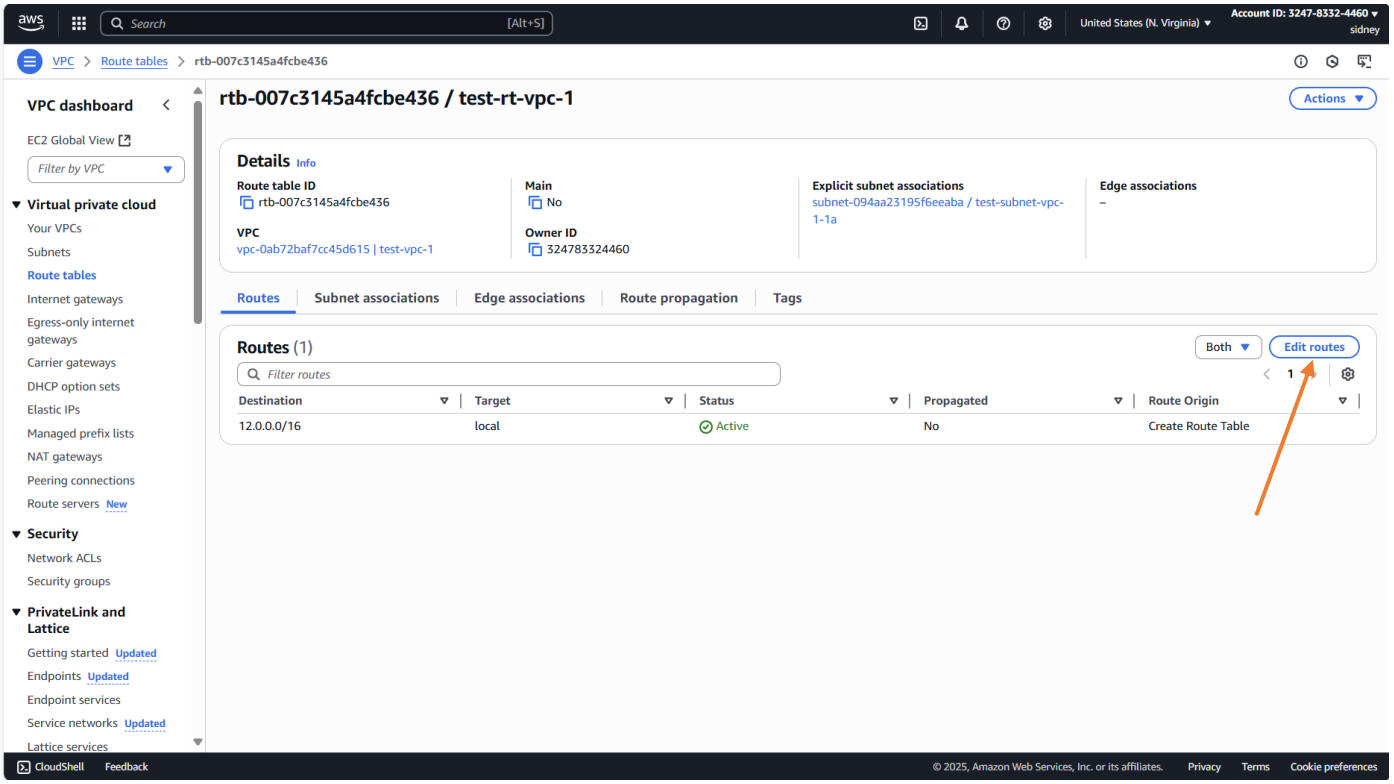
We have created the internet gateway for VPC "**test-vpc-1**".



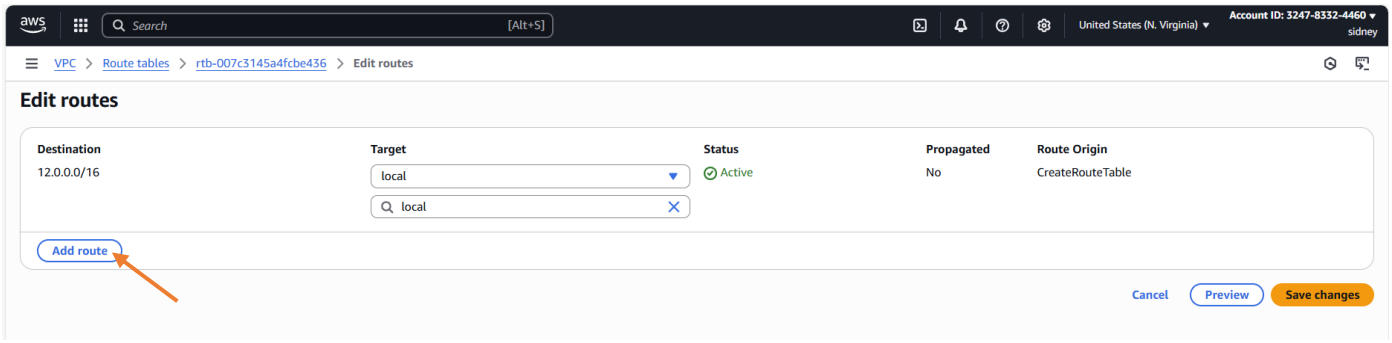
We have to go to the Route Table and create a route for internet using the internet gateway so that any resources in the subnet can be accessed from the internet. To do this, go back to our route tables



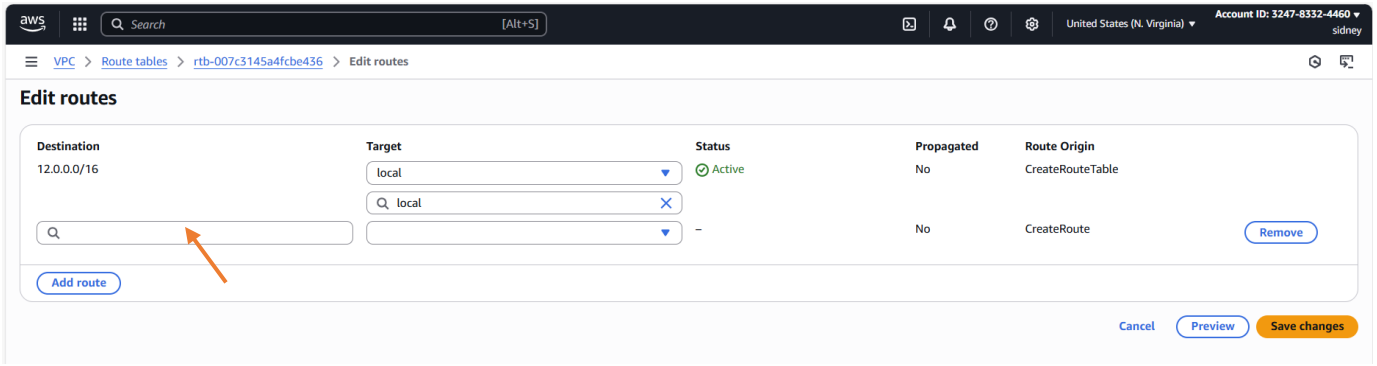
Click on the Route Table ID for test-vpc-1



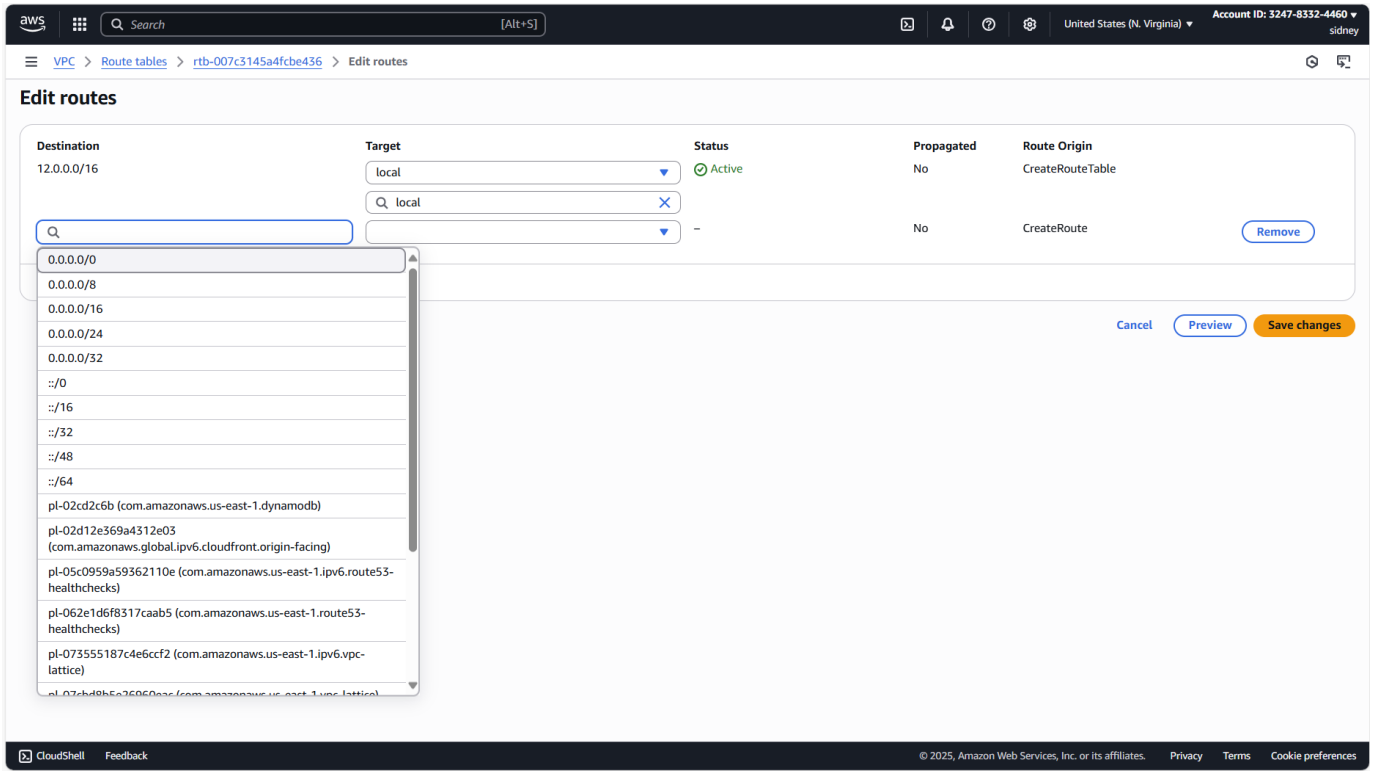
Click on “**Edit Routes**”



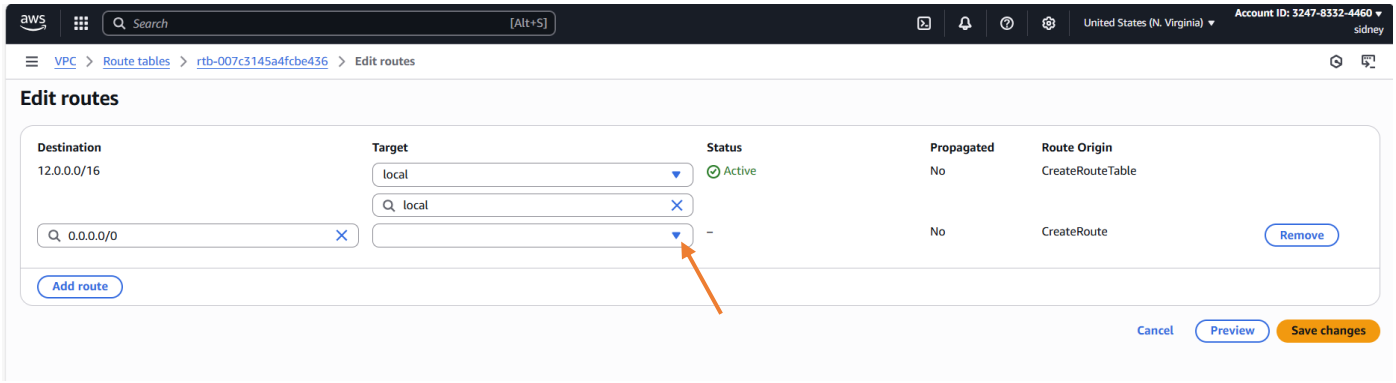
Click on “**Add Route**”



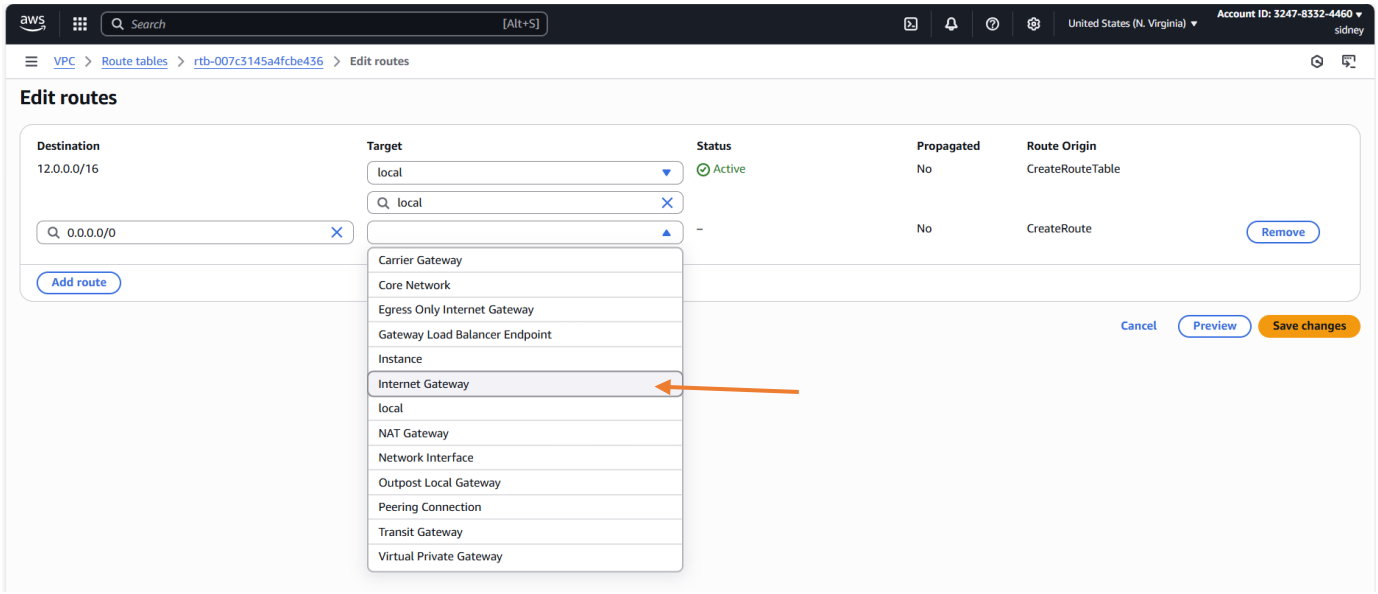
Click on the destination



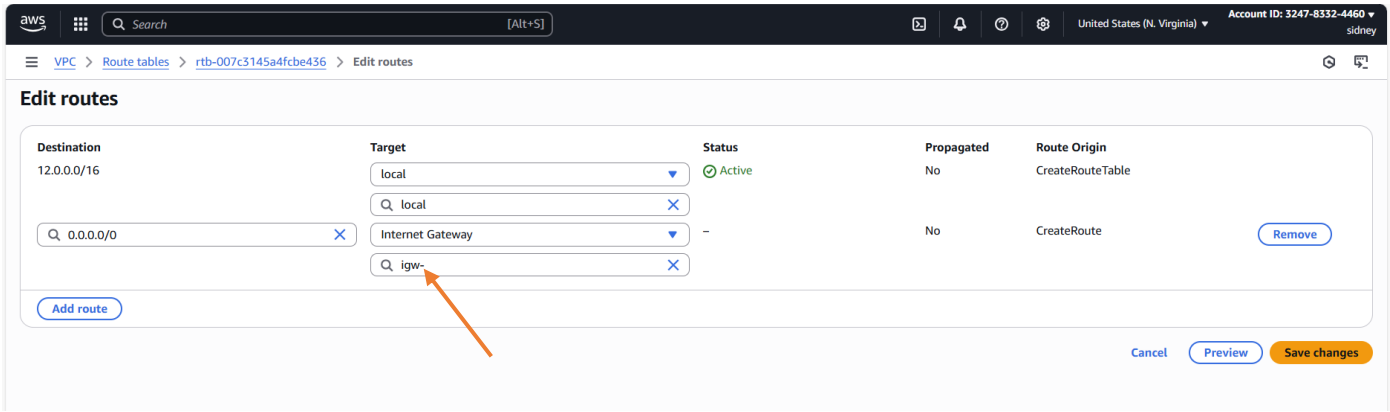
Select “**0.0.0.0/0**”



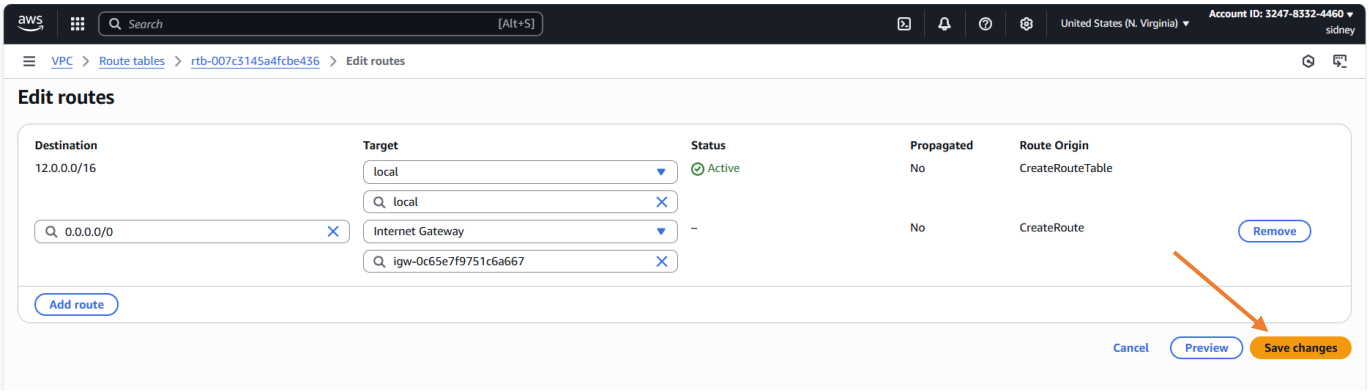
Click on “**Target**”



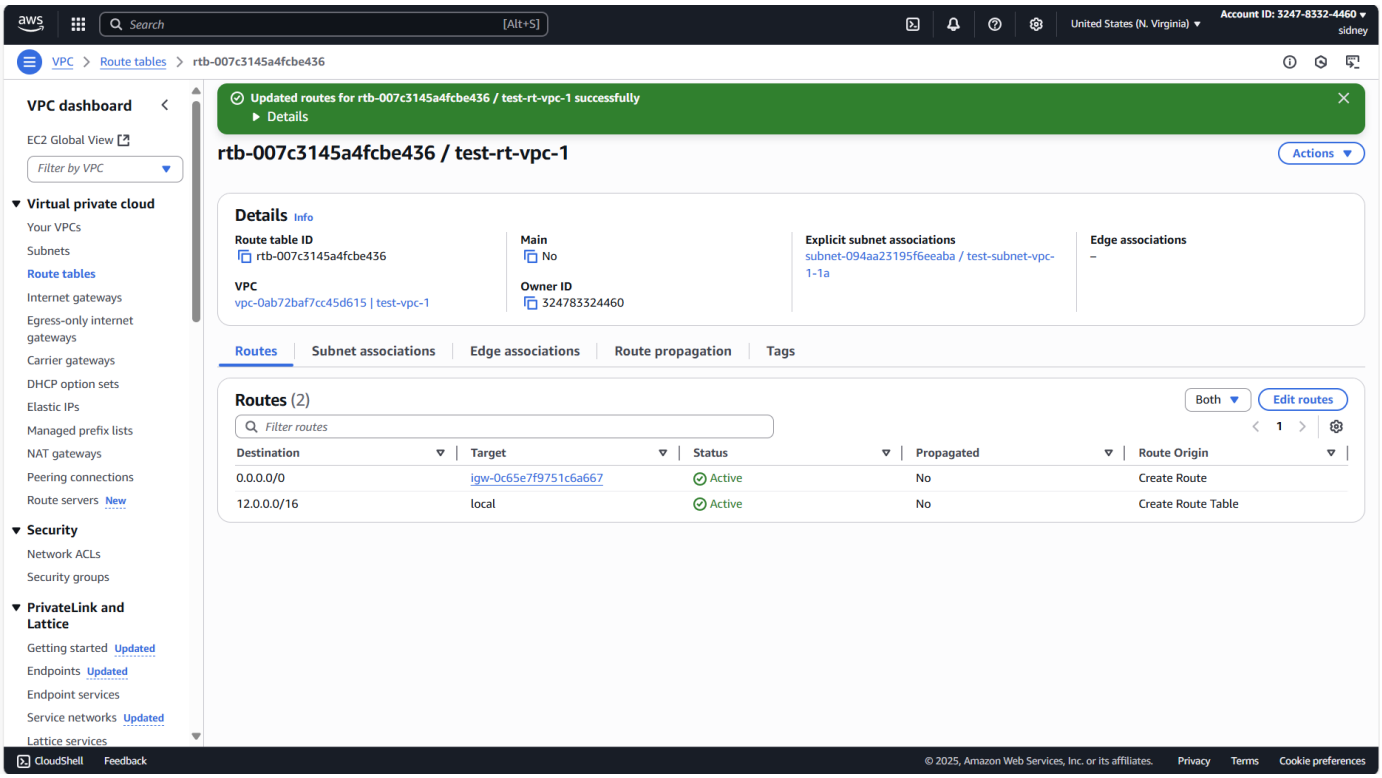
Select “**Internet Gateway**”



Click on “**igw**” and select our internet gateway

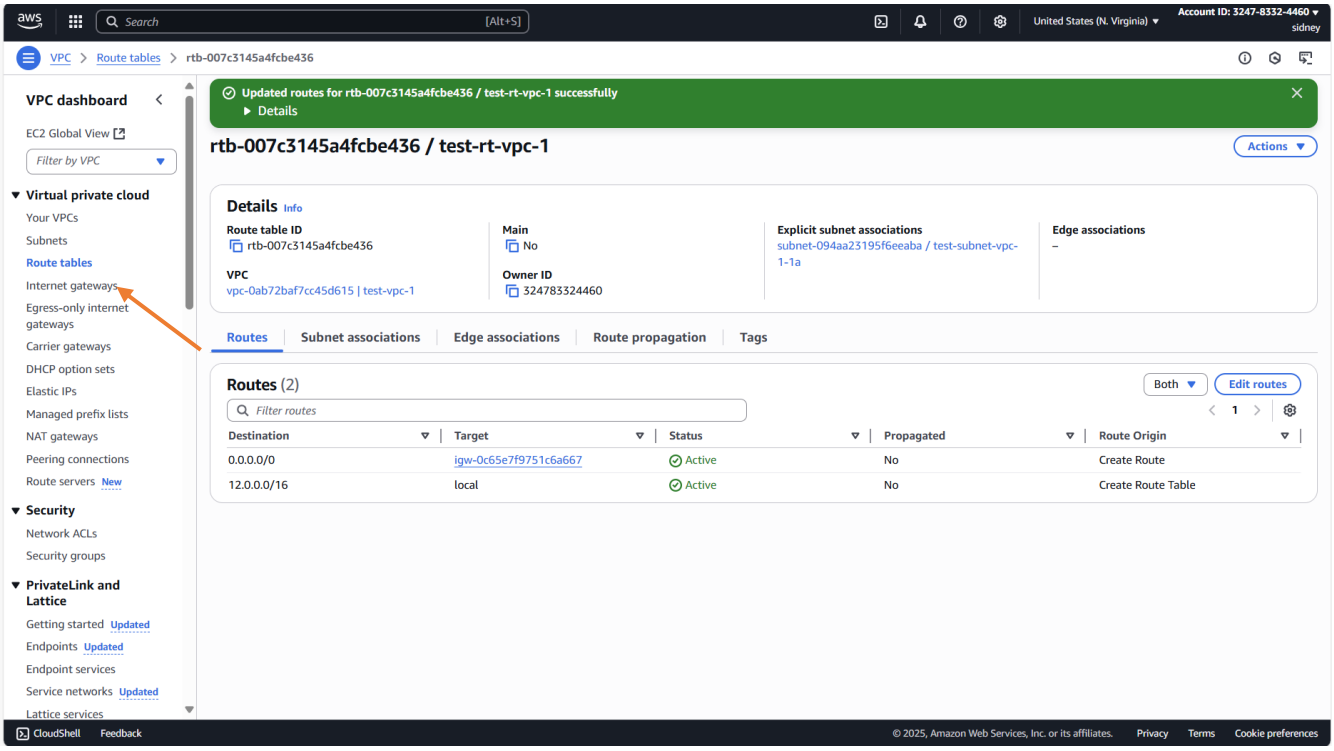


Click on "**Save Changes**"

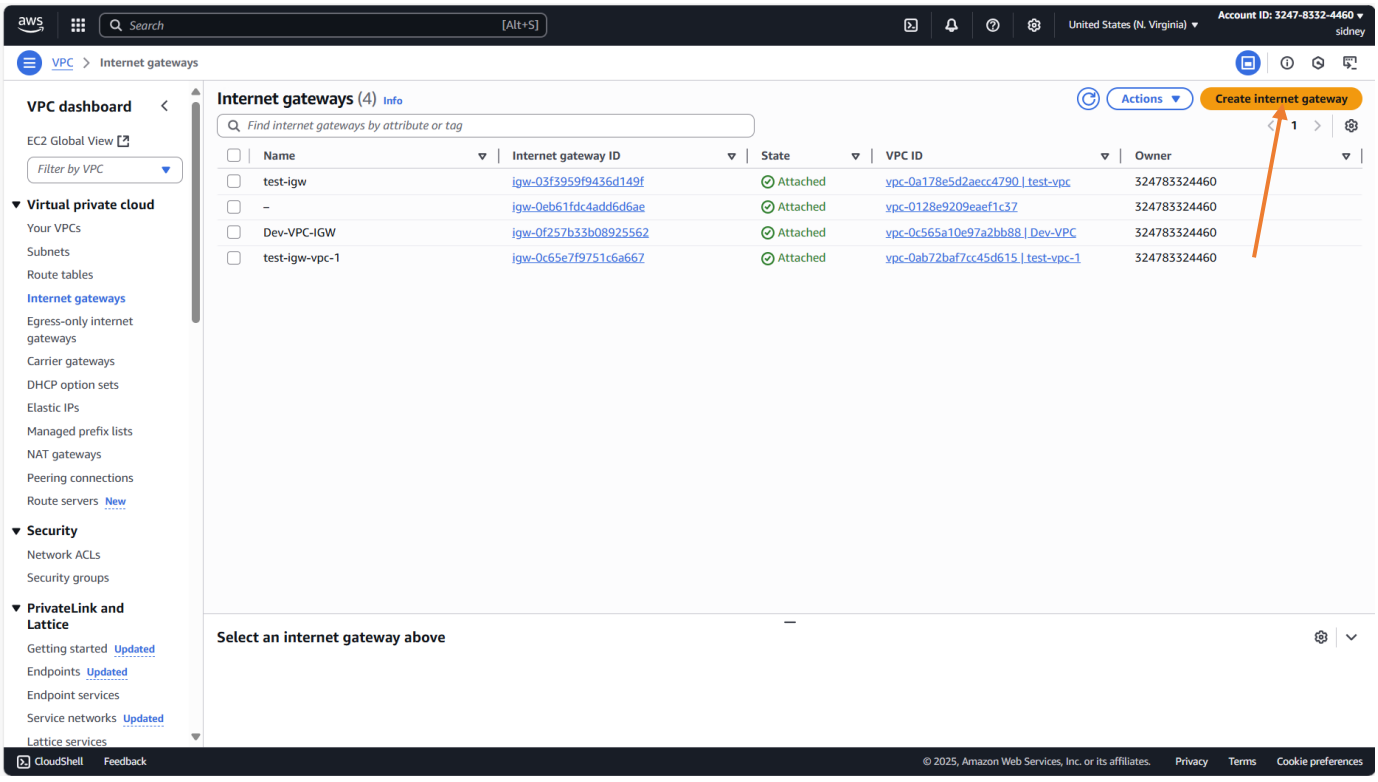


We have created the internet gateway for the first VPC and associate it with the VPC. Now, we have to create the Internet Gateway for the second VPC.

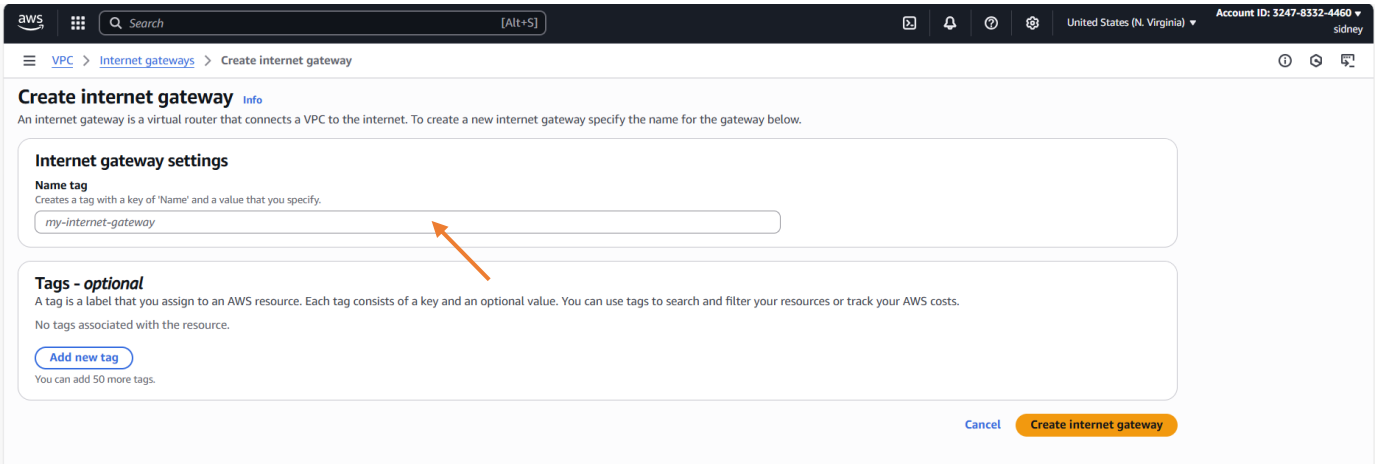
Go to VPC dashboard



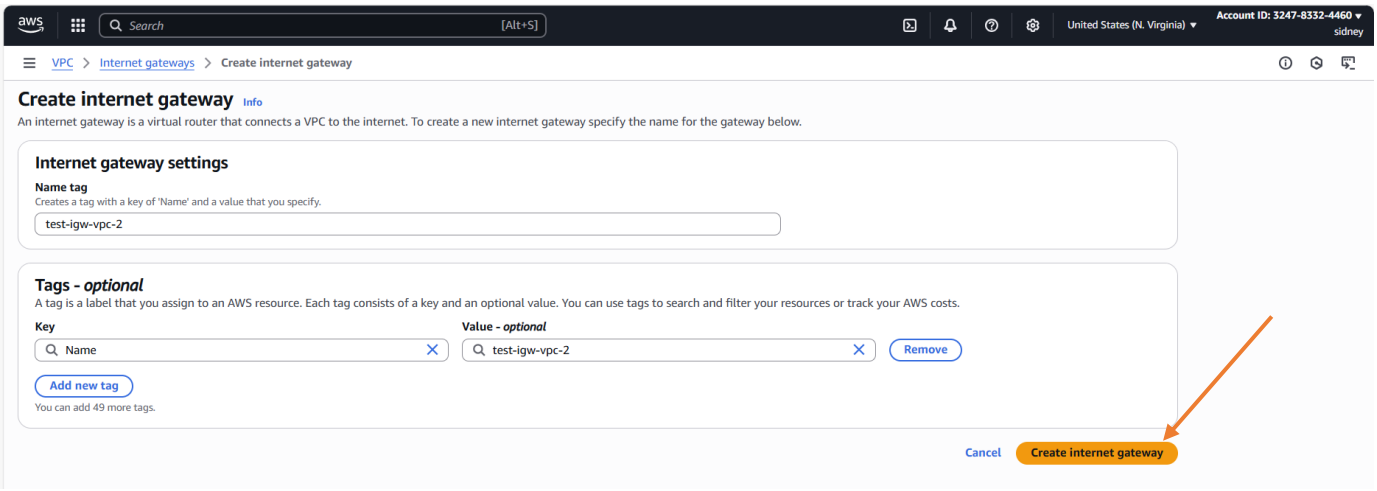
Click on “**Internet Gateways**”



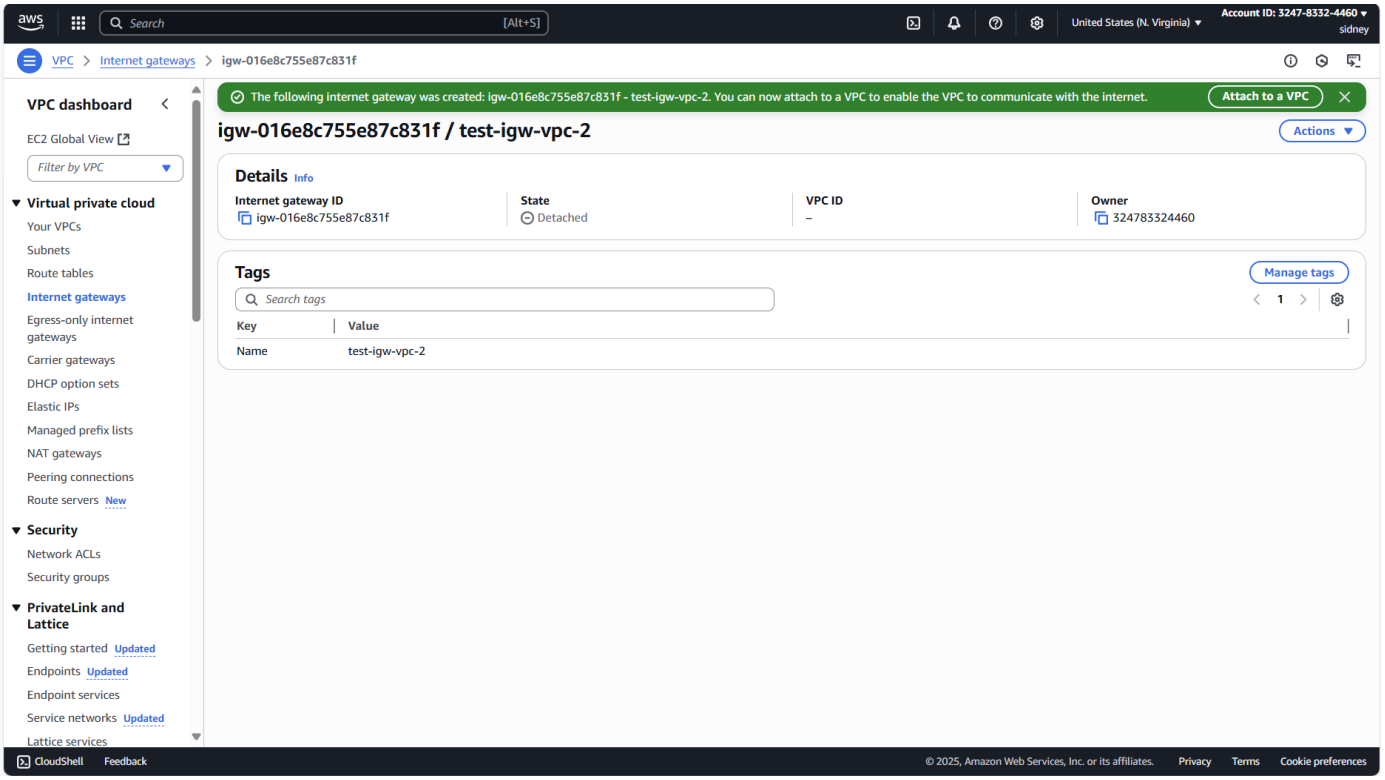
Click on “**Create Internet Gateway**”



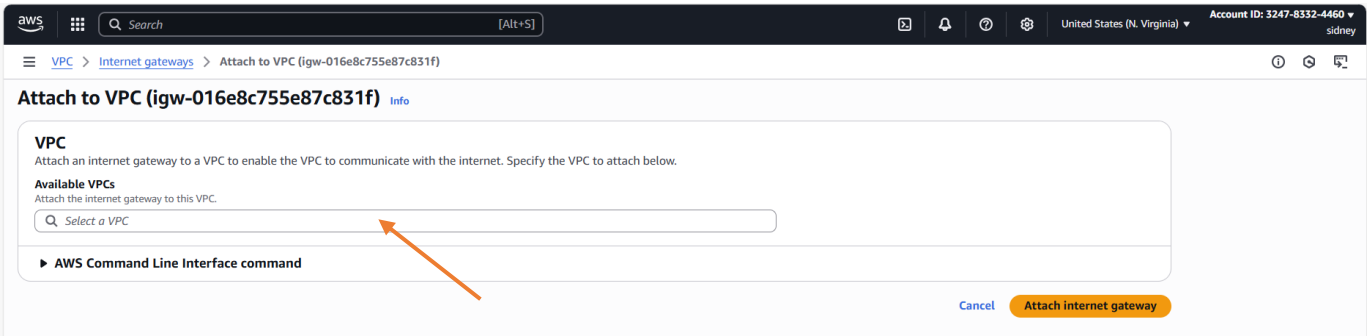
We will give the Internet Gateway the name "**test-igw-vpc-2**"



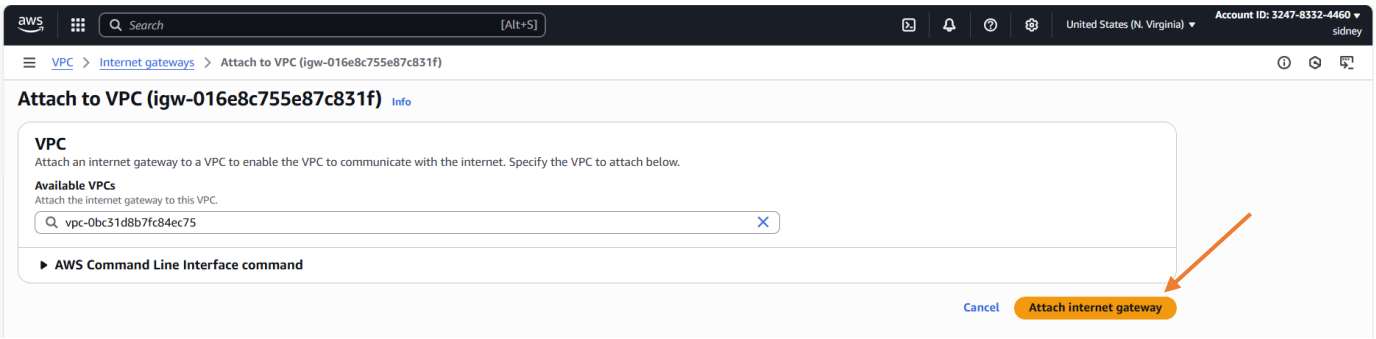
Click on "**Create Internet Gateway**"



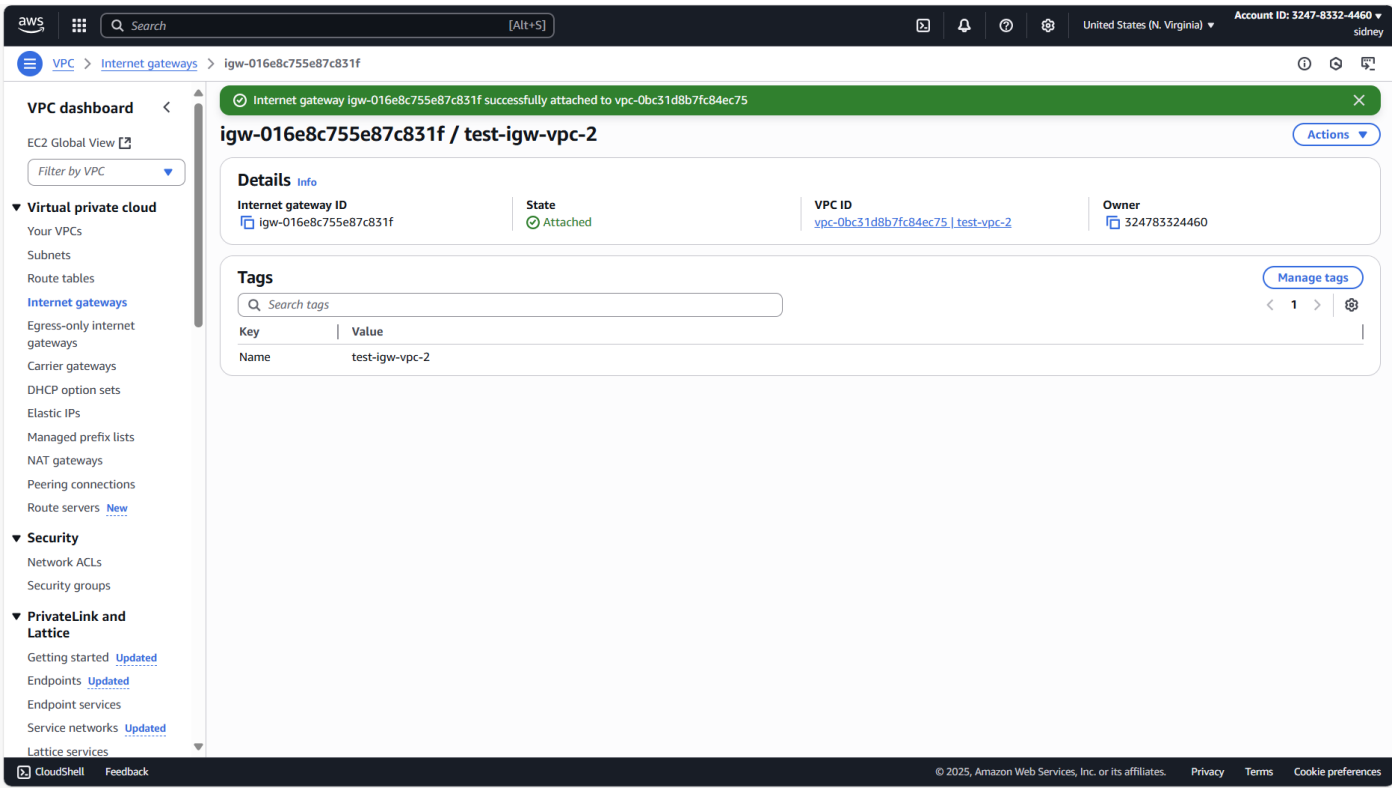
Let us now attach the created Internet Gateway to our VPC "**test-vpc-2**". Click on the drop down on "**Action**" and select "**Attach to VPC**".



Click on the field under "**Available VPCs**" and select the VPC "**test-vpc-2**"

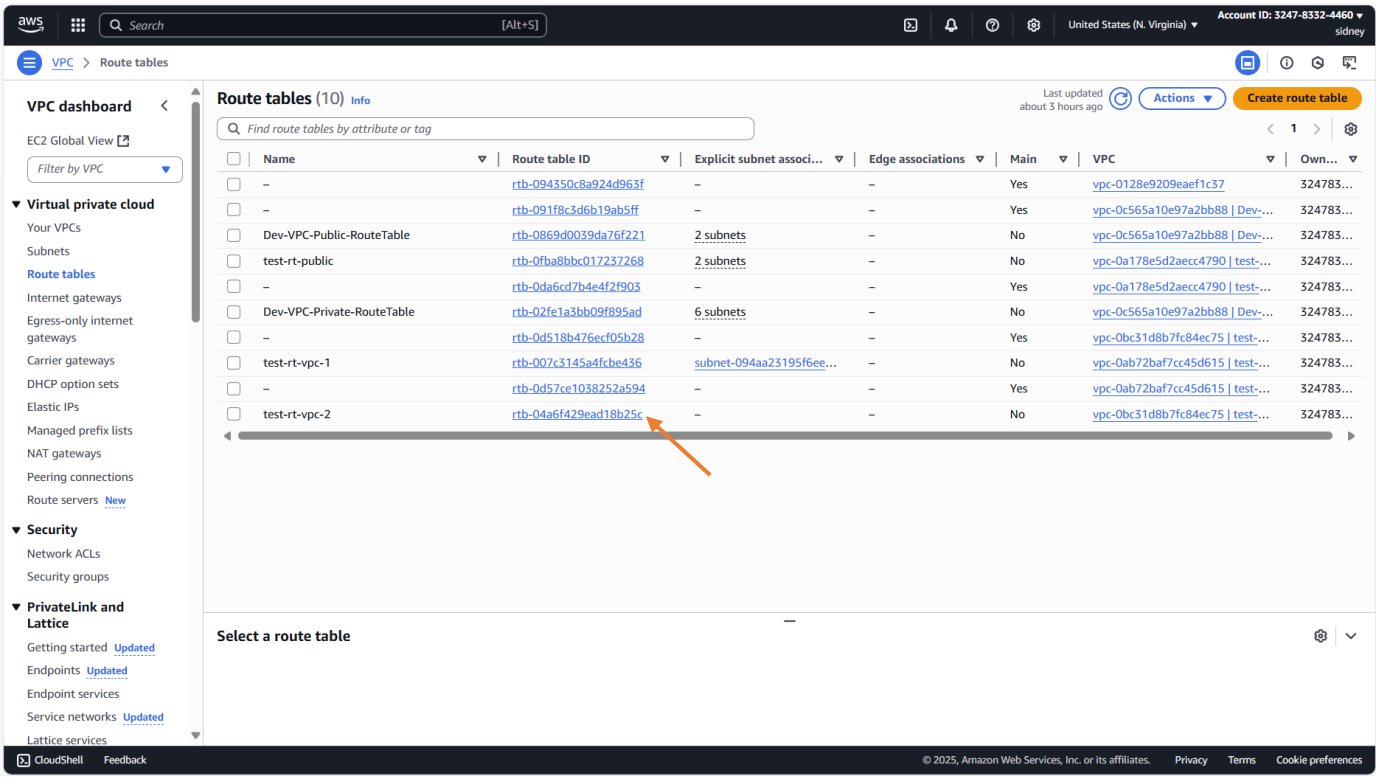


Click on "**Attach Internet Gateway**"

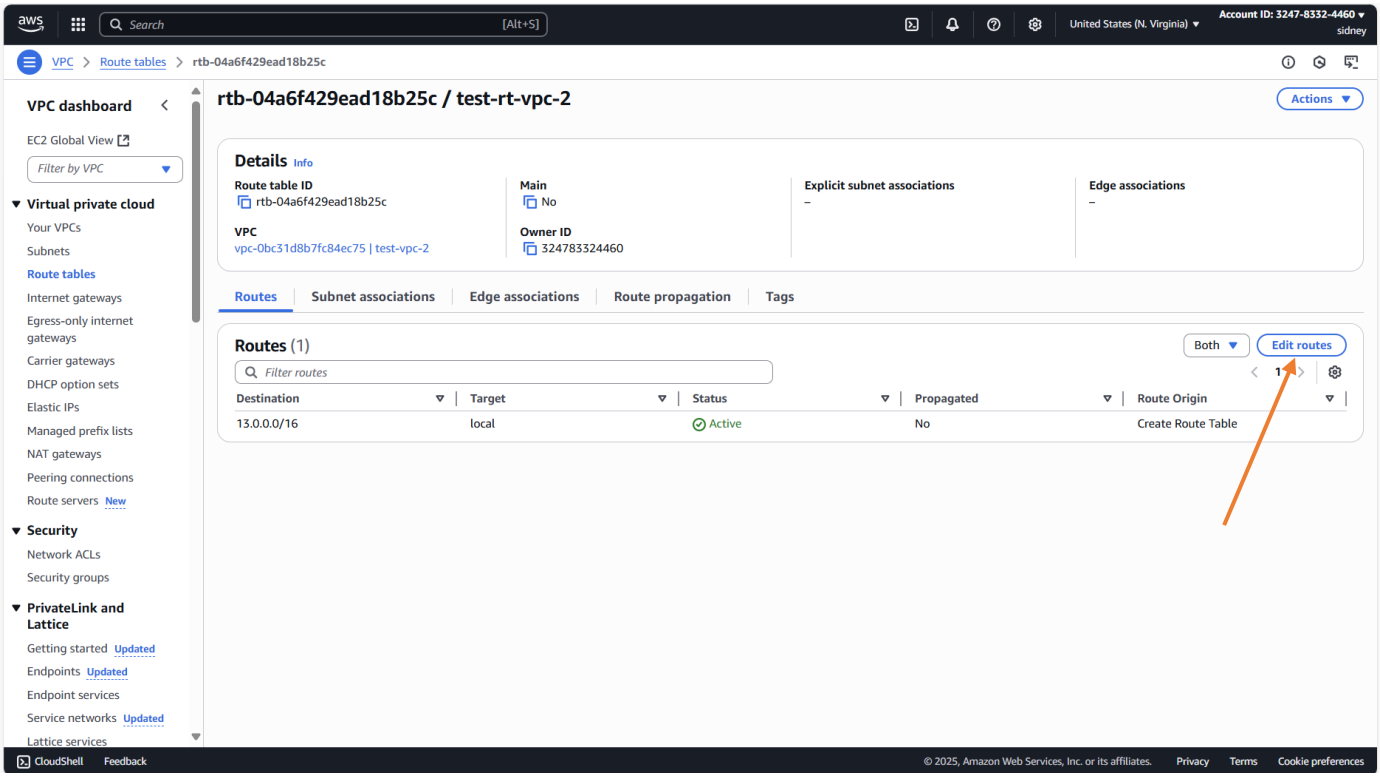


The internet gateway has been attached to our VPC "**test-vpc-2**". Now, let us to go to the Route Table and create a route for internet using the internet gateway so that any resources in the subnet can be accessed from the internet.

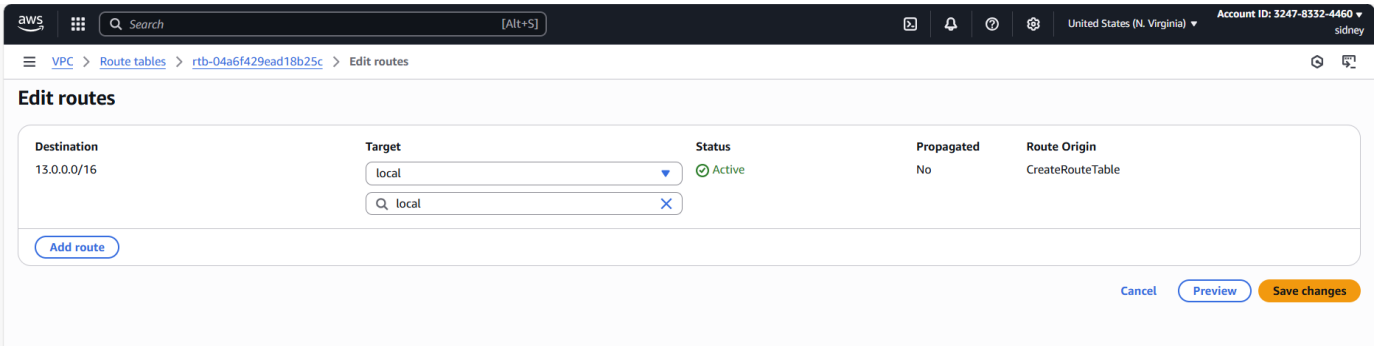
To do this, go back to our route tables



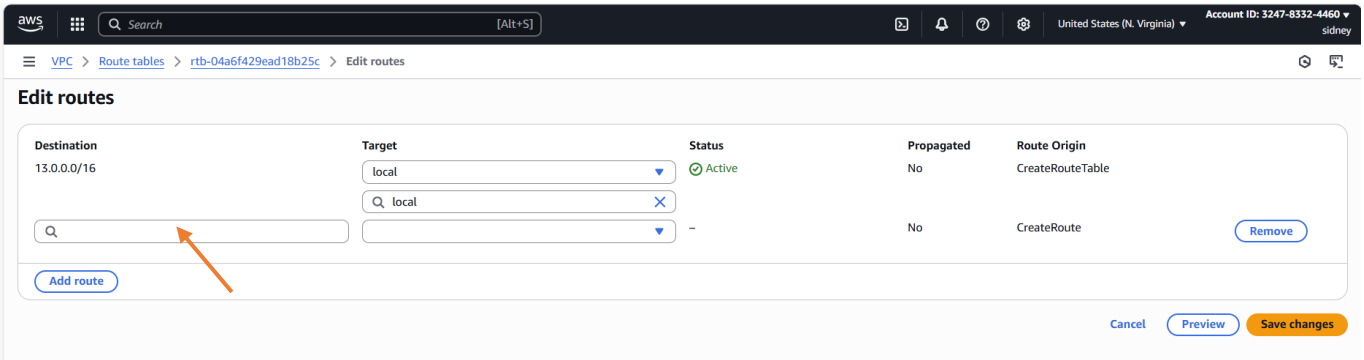
Click on the “**Route Table ID**” of the “**test-rt-vpc-2**”



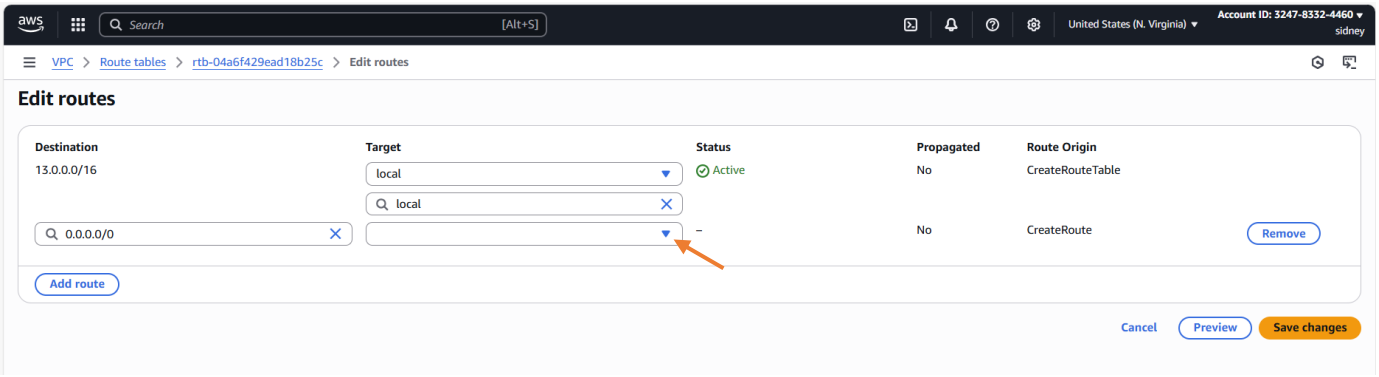
Click on “**Edit Routes**”



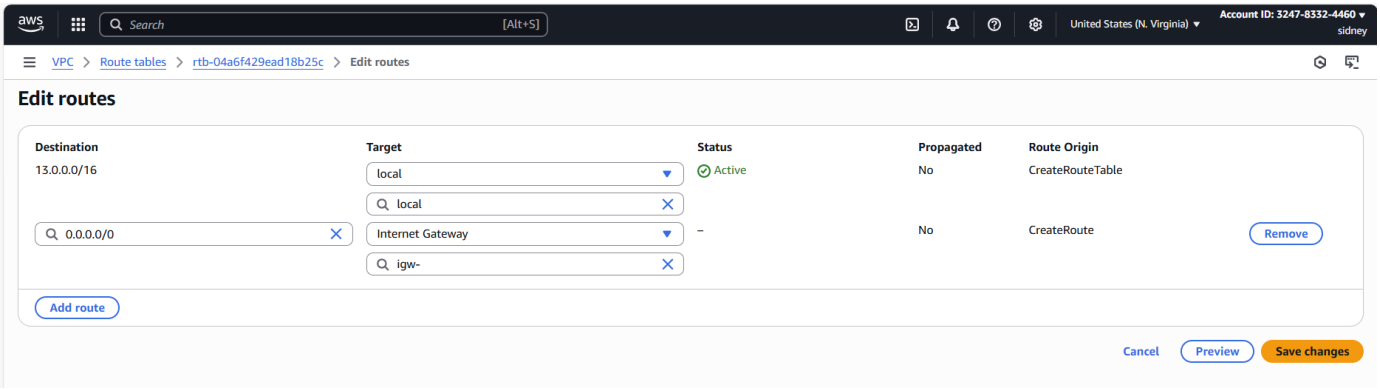
Click on “**Add Route**”



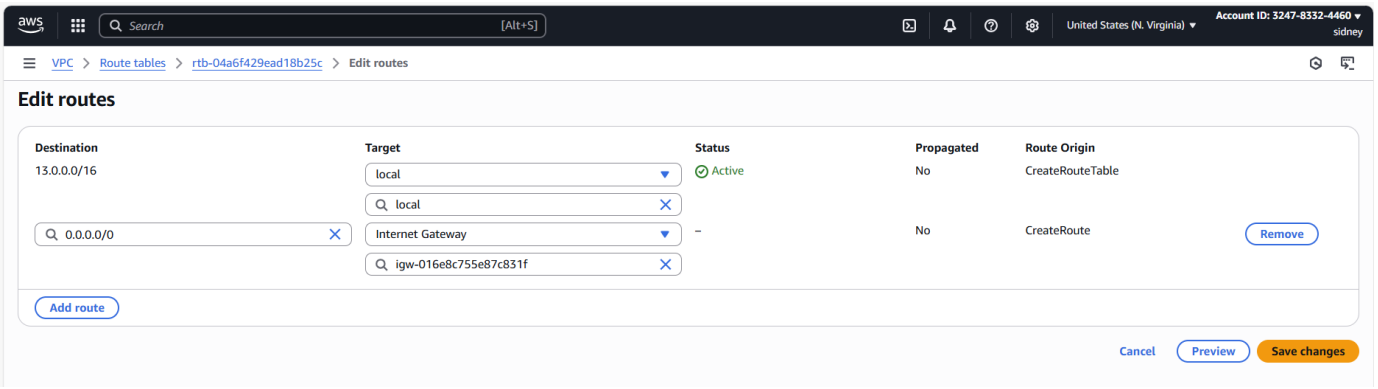
Click on the “**destination**” and choose anywhere (**0.0.0.0/0**)



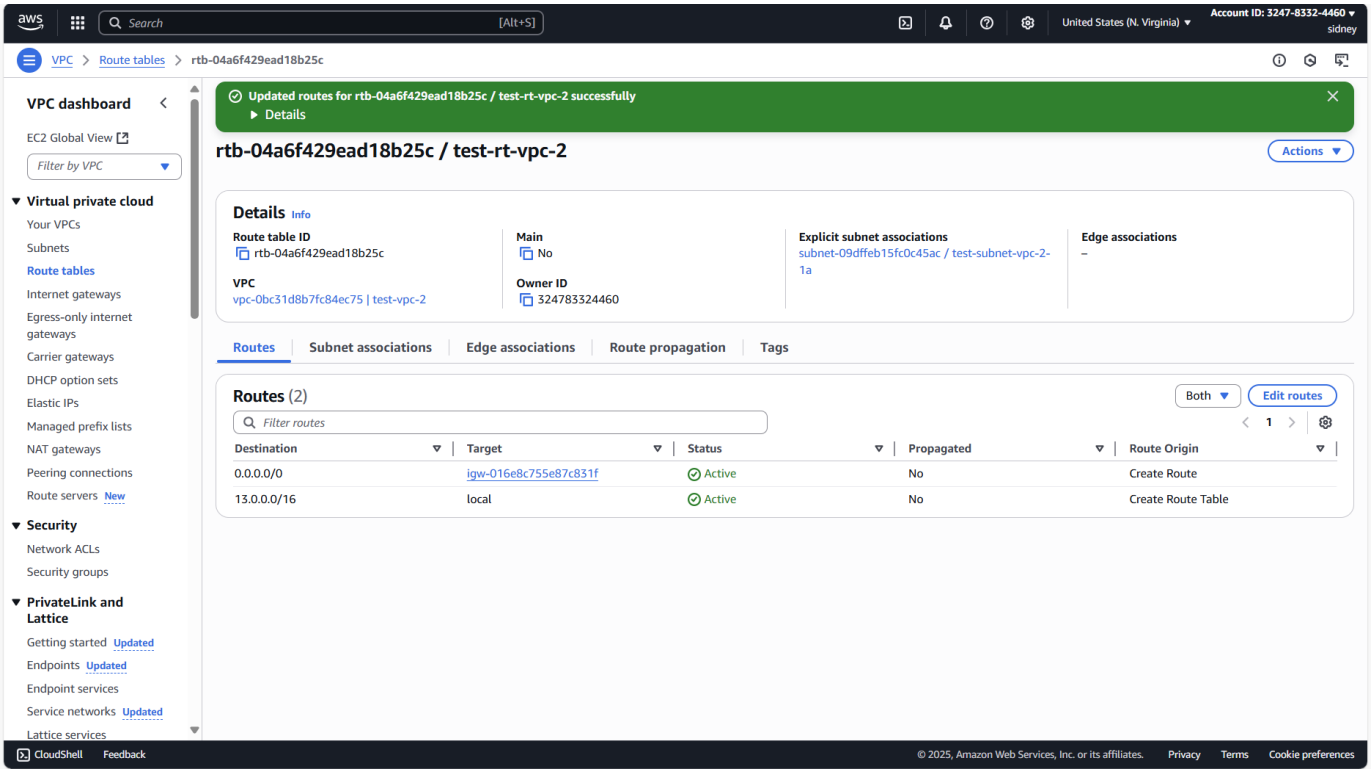
Click on the drop down on "**Target**" and select "**Internet Gateway**"



Click on "**igw**" and select our Internet Gateway



Click on "**Save Changes**"

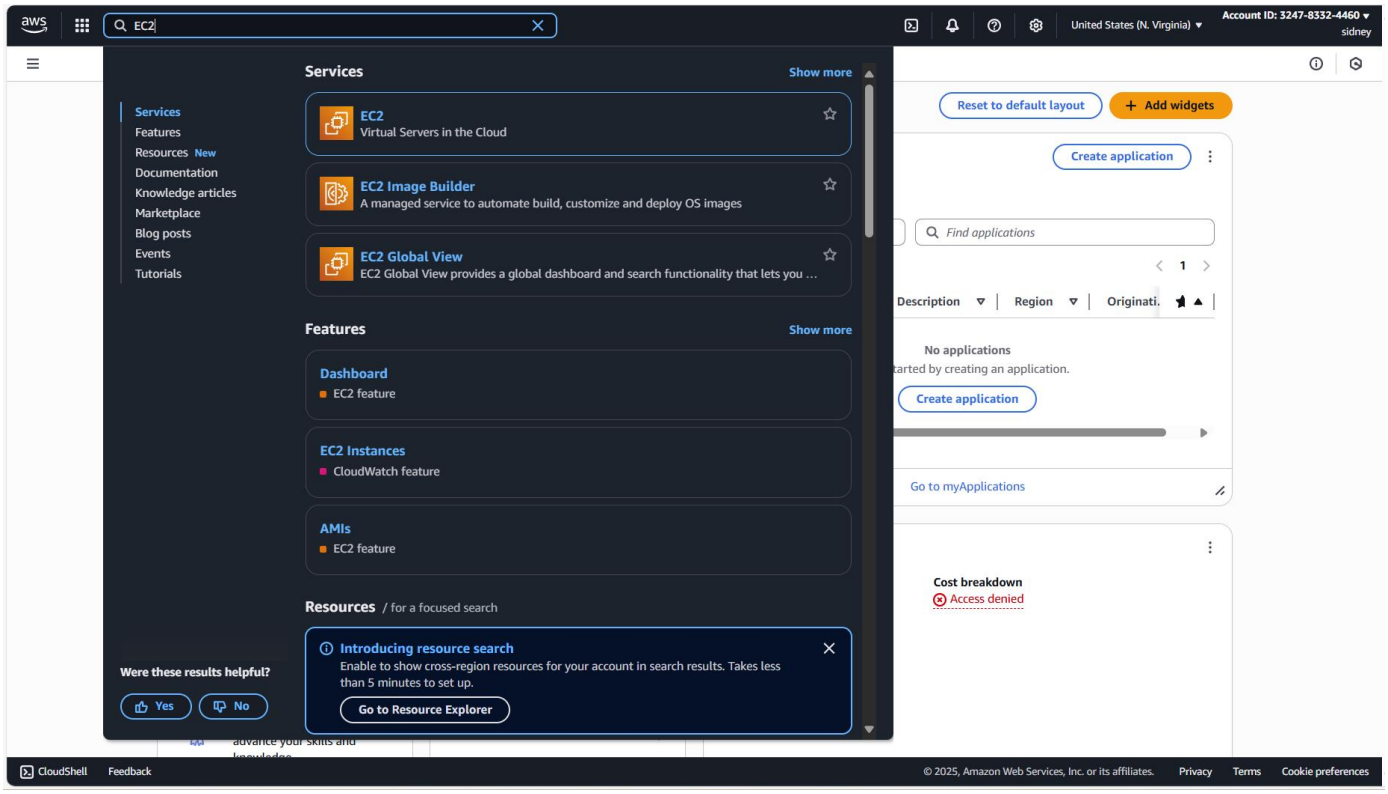


We have created the internet gateway for the second VPC and associate it with the VPC

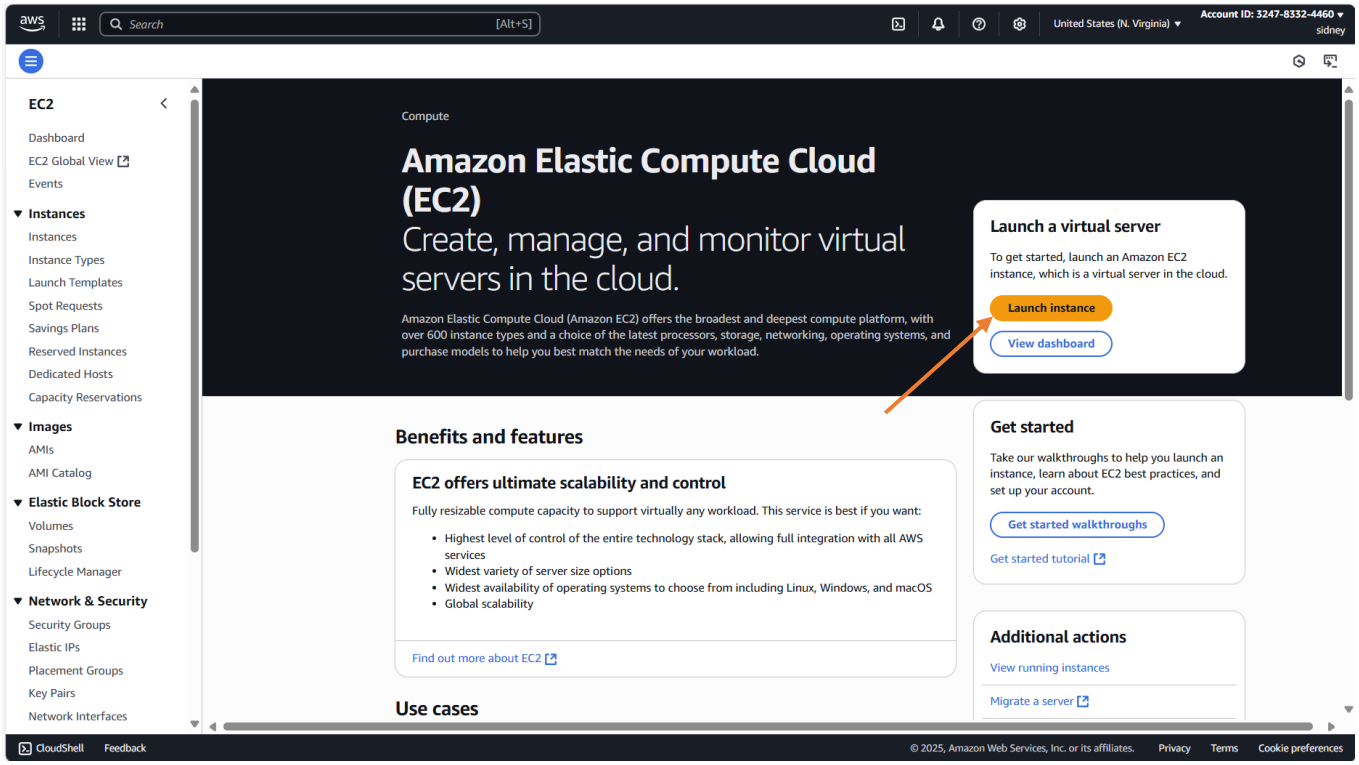
Part 7: Launch EC2 instances in test-vpc-1 and test-vpc-2

We are almost done with our networking part. Let us now provision our EC2 instances.

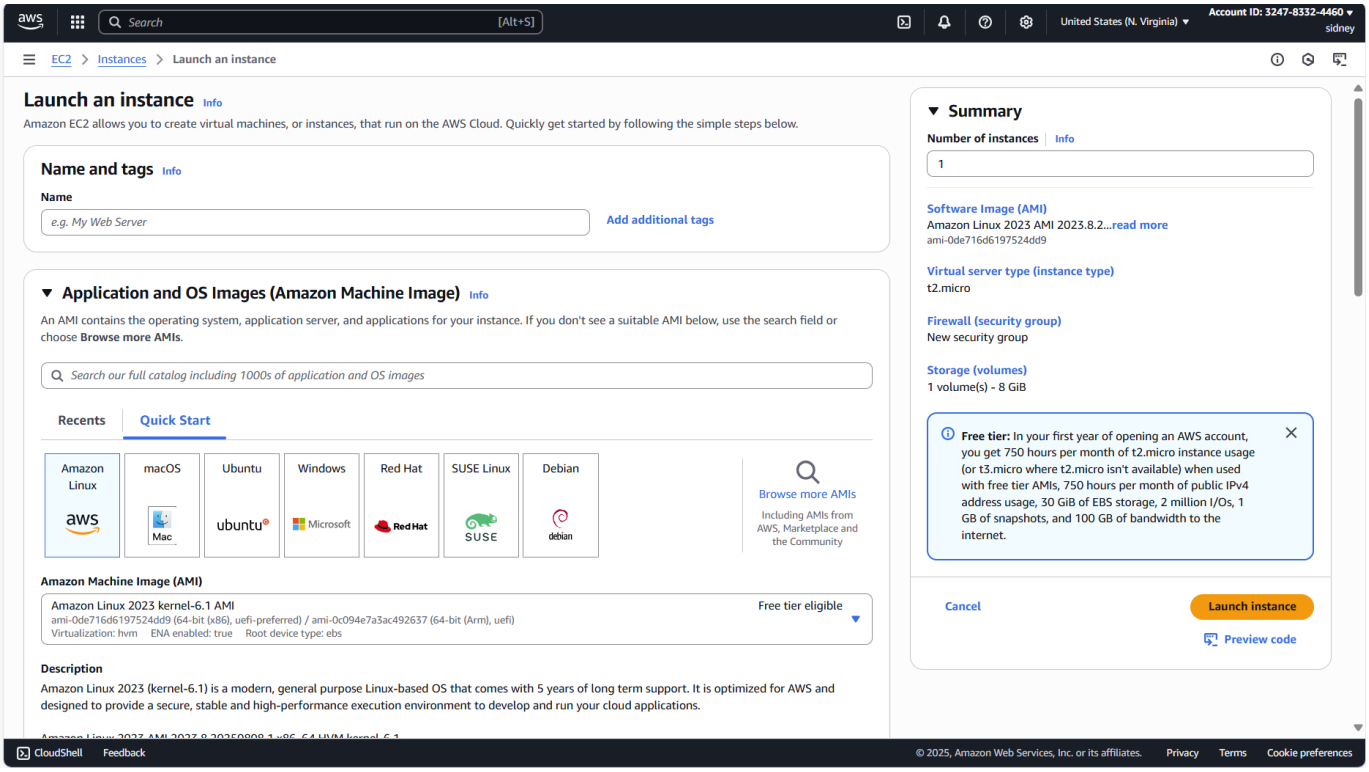
Go to AWS Management console and search for EC2



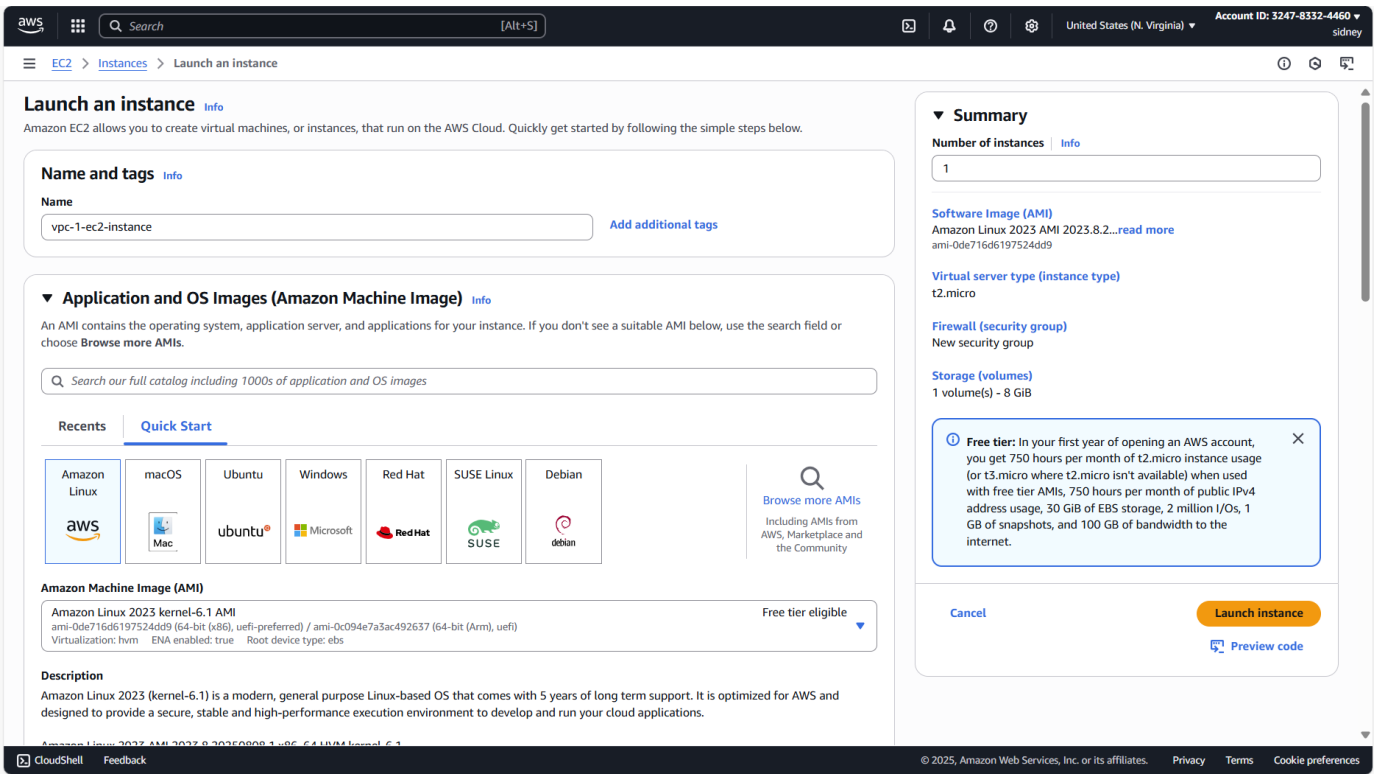
Click on "**EC2**"



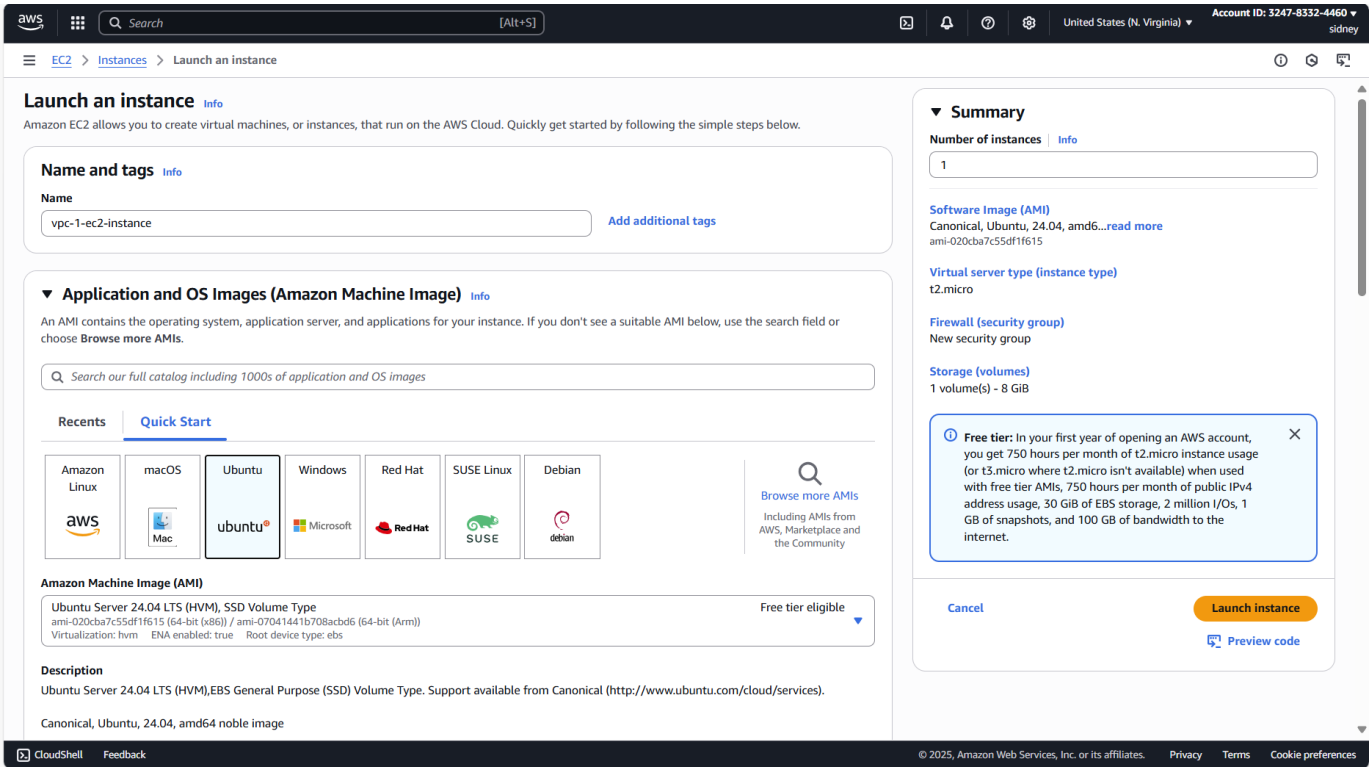
Click on "**Launch Instance**"



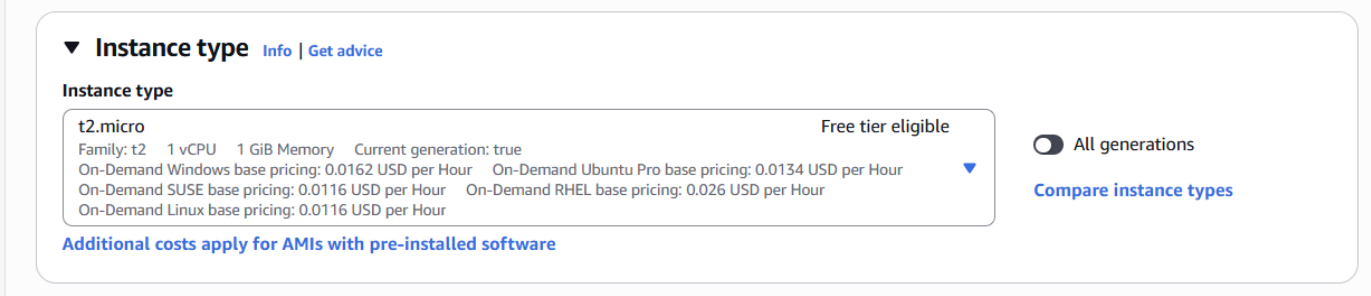
Give the EC2 instance a name, I will call it "**vpc-1-ec2-instance**"



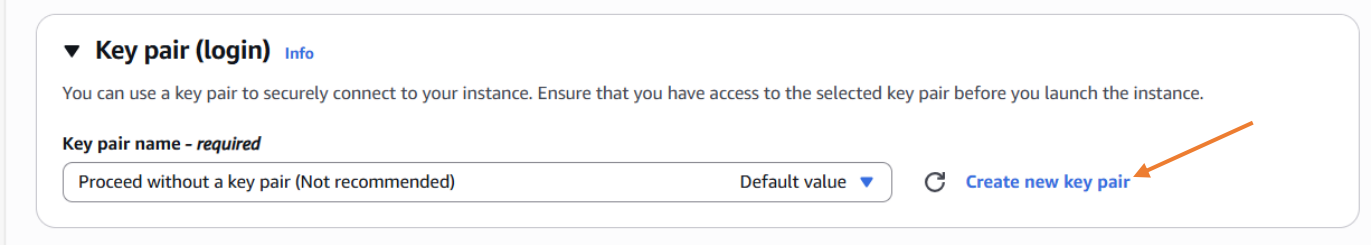
For AMI, we will choose "**Ubuntu**"



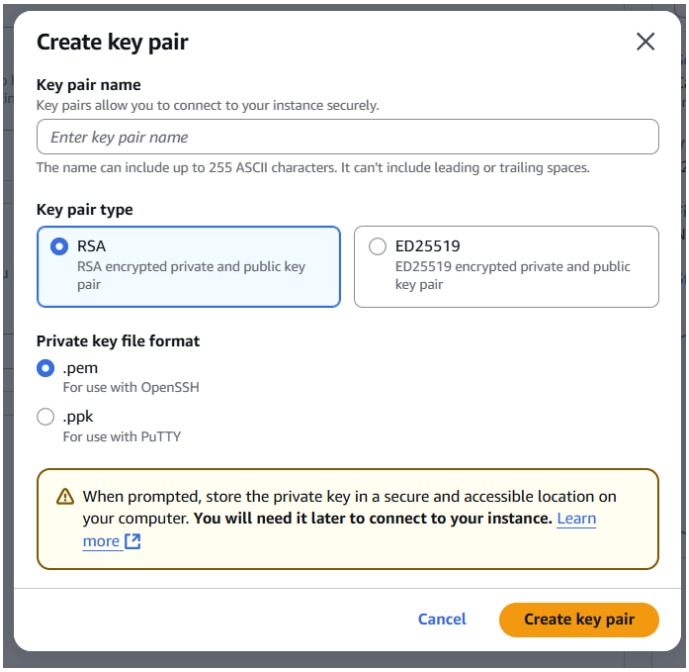
Scroll down to “**Instance Type**” and select “**t2.micro**”



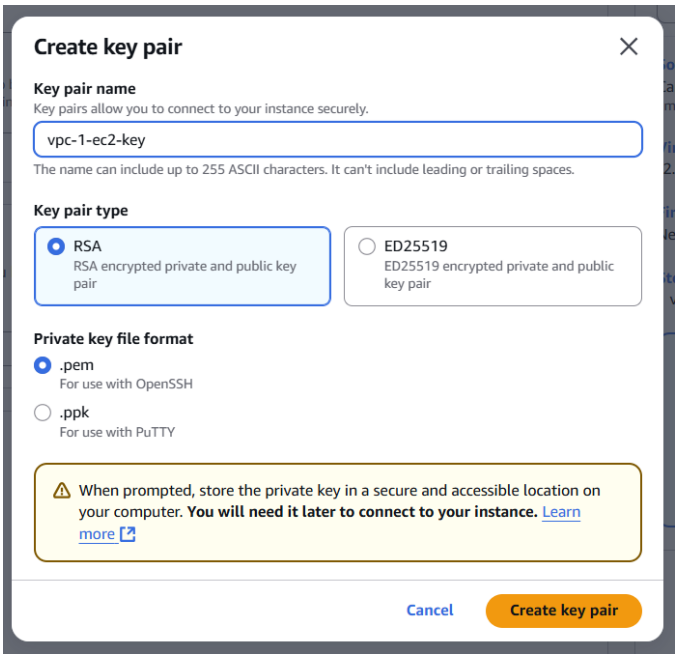
Scroll down to “Key Pair” and click on “create new key pair”



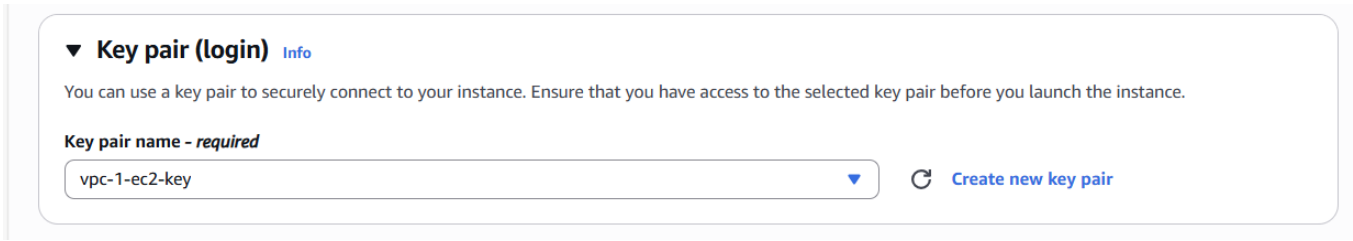
Click on “create new key pair”



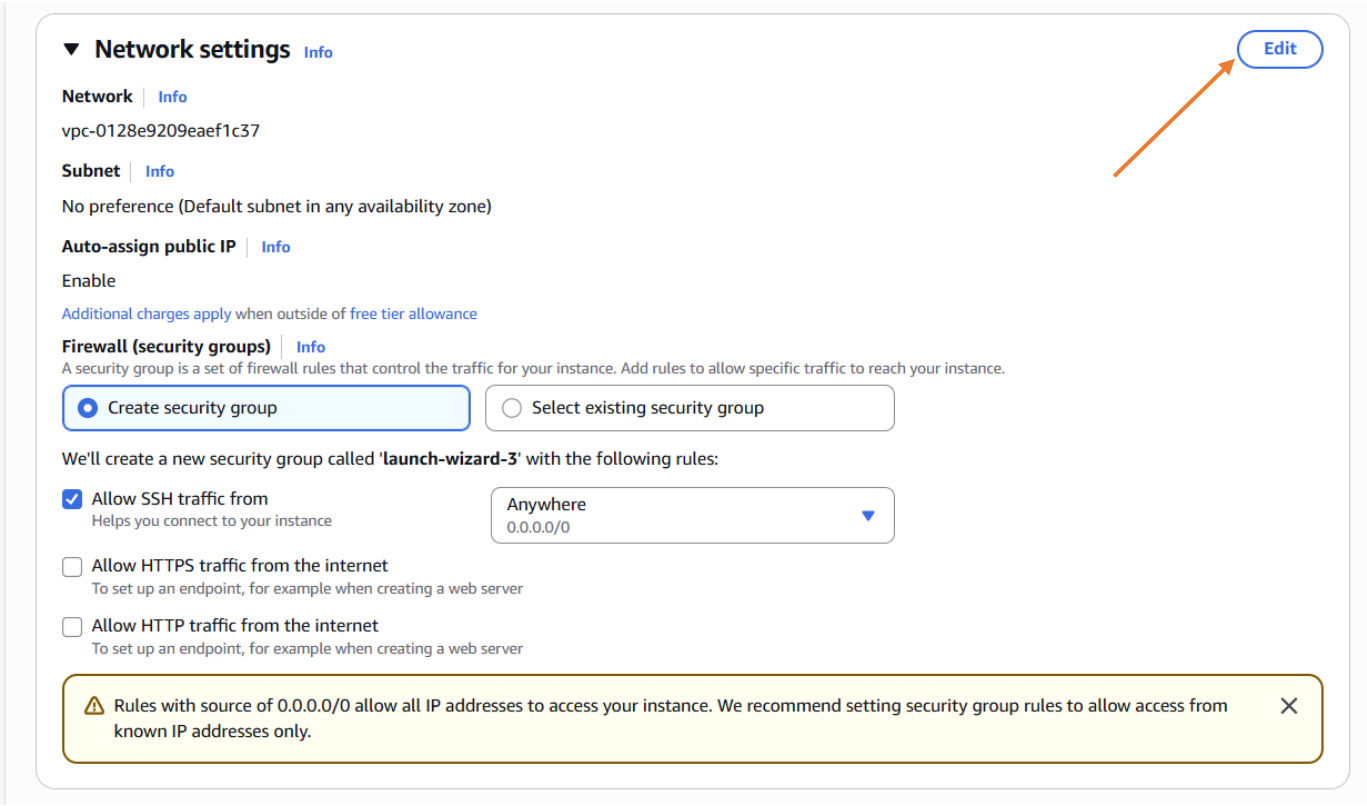
We will name the key "**vpc-1-ec2-key**"



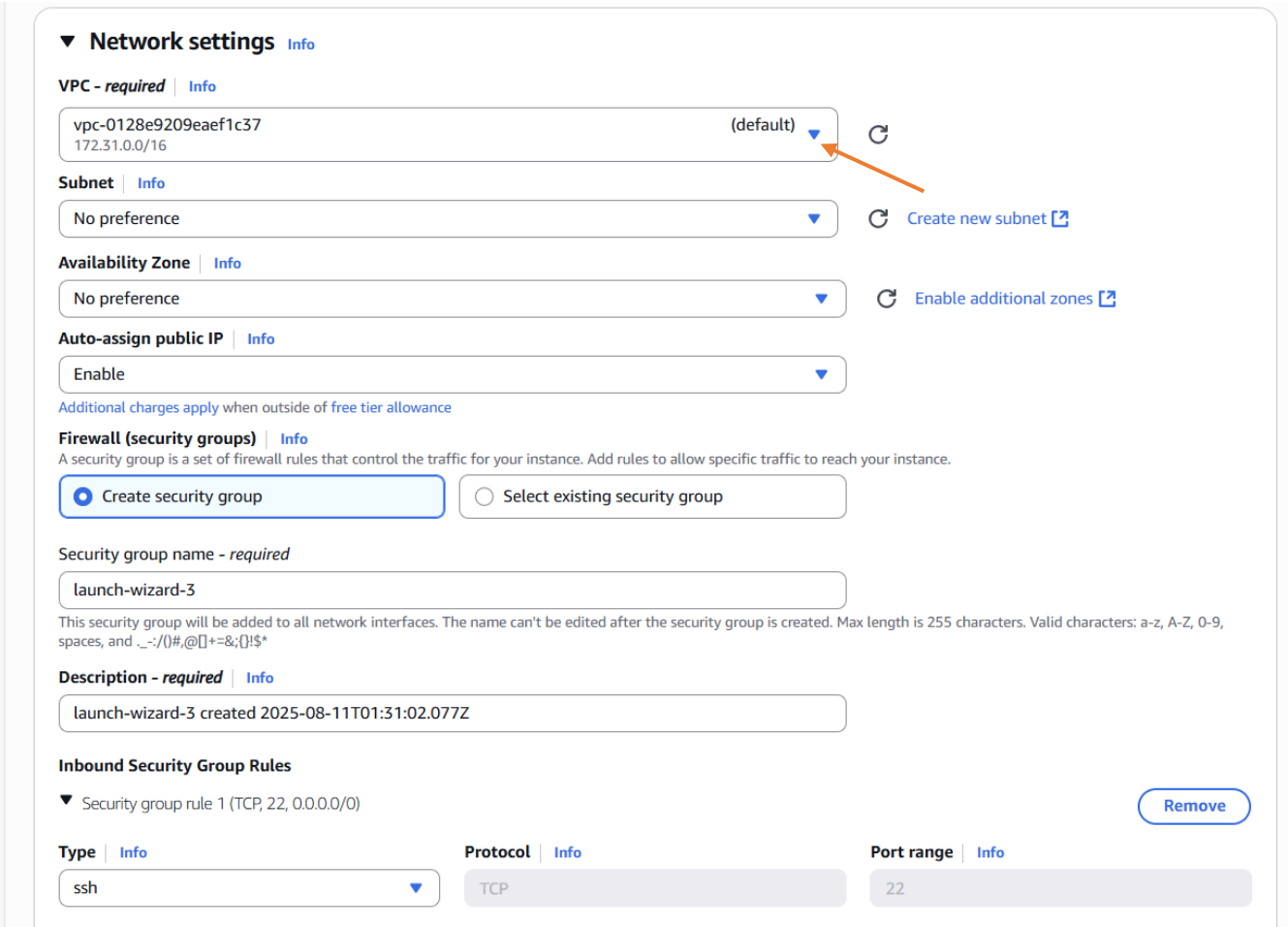
Click on "**create key pair**"



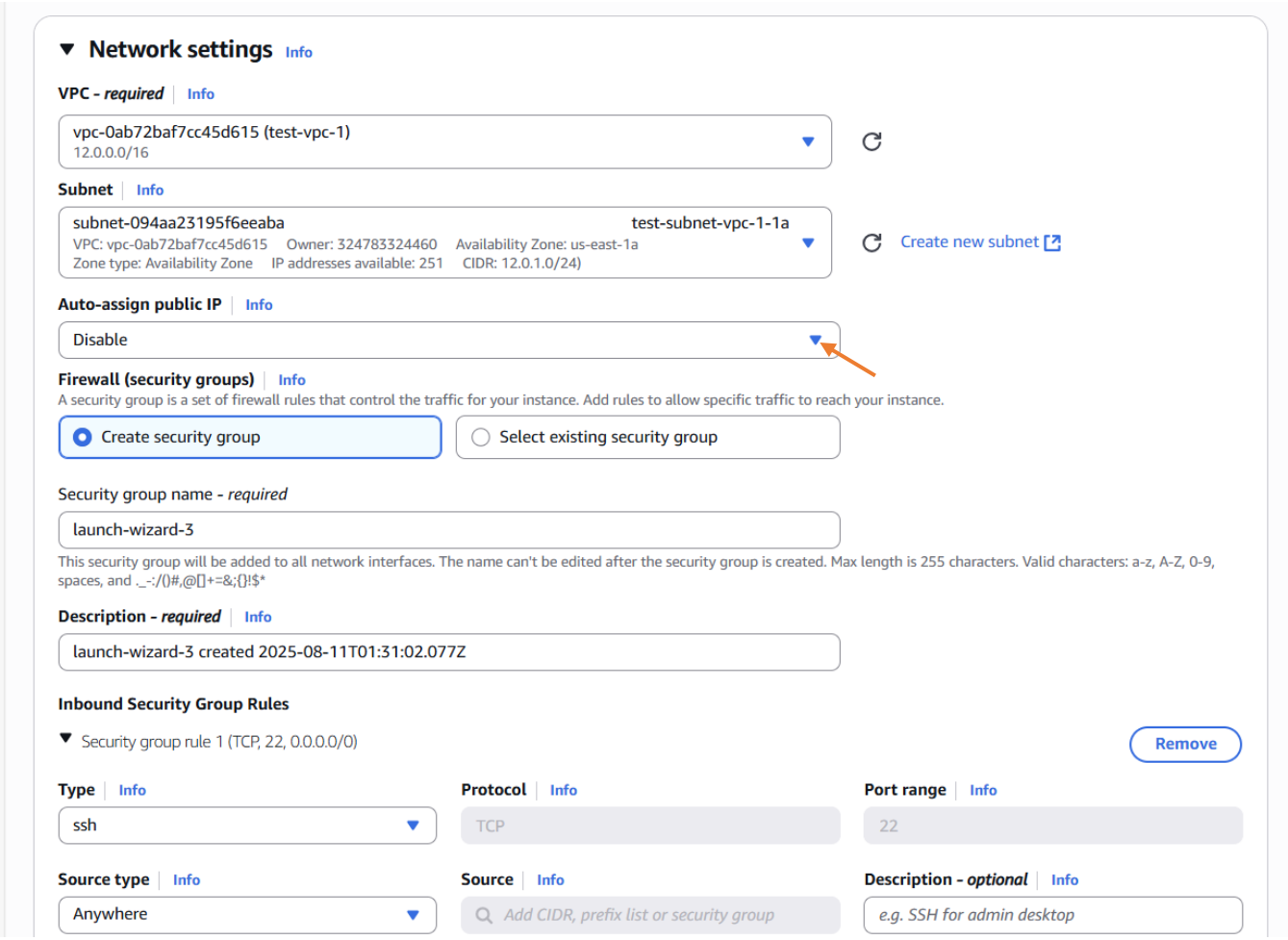
Scroll down to "**Network Settings**"



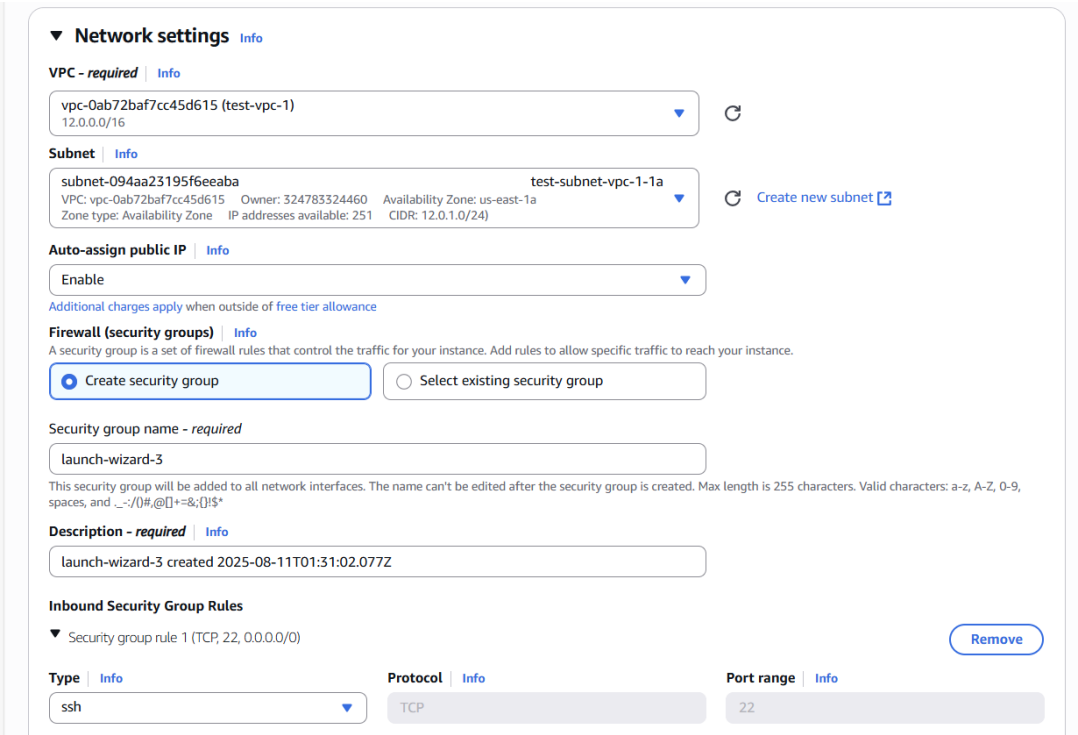
Click on "**Edit**"



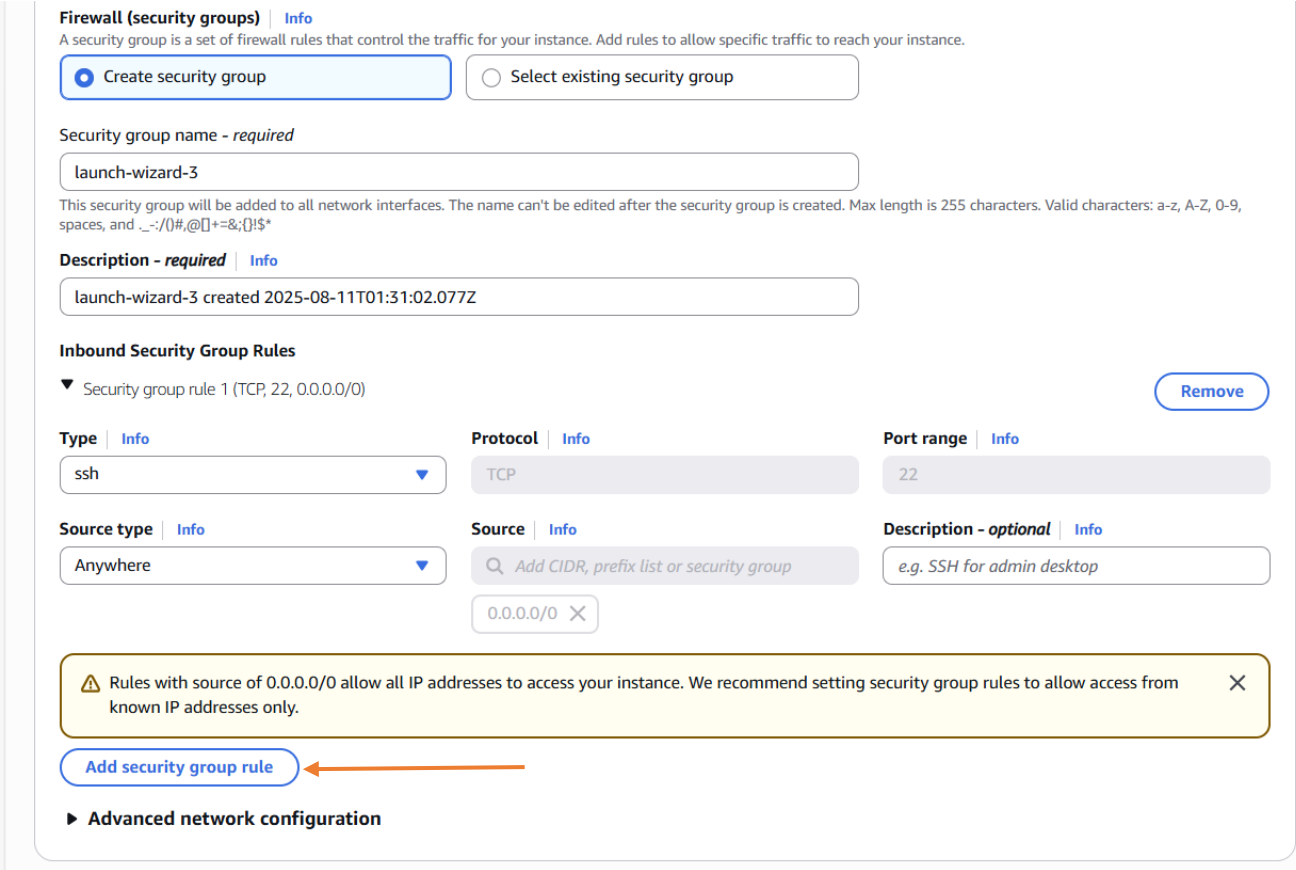
Click on the drop down on “**VPC**” and select our VPC “**test-vpc-1**”



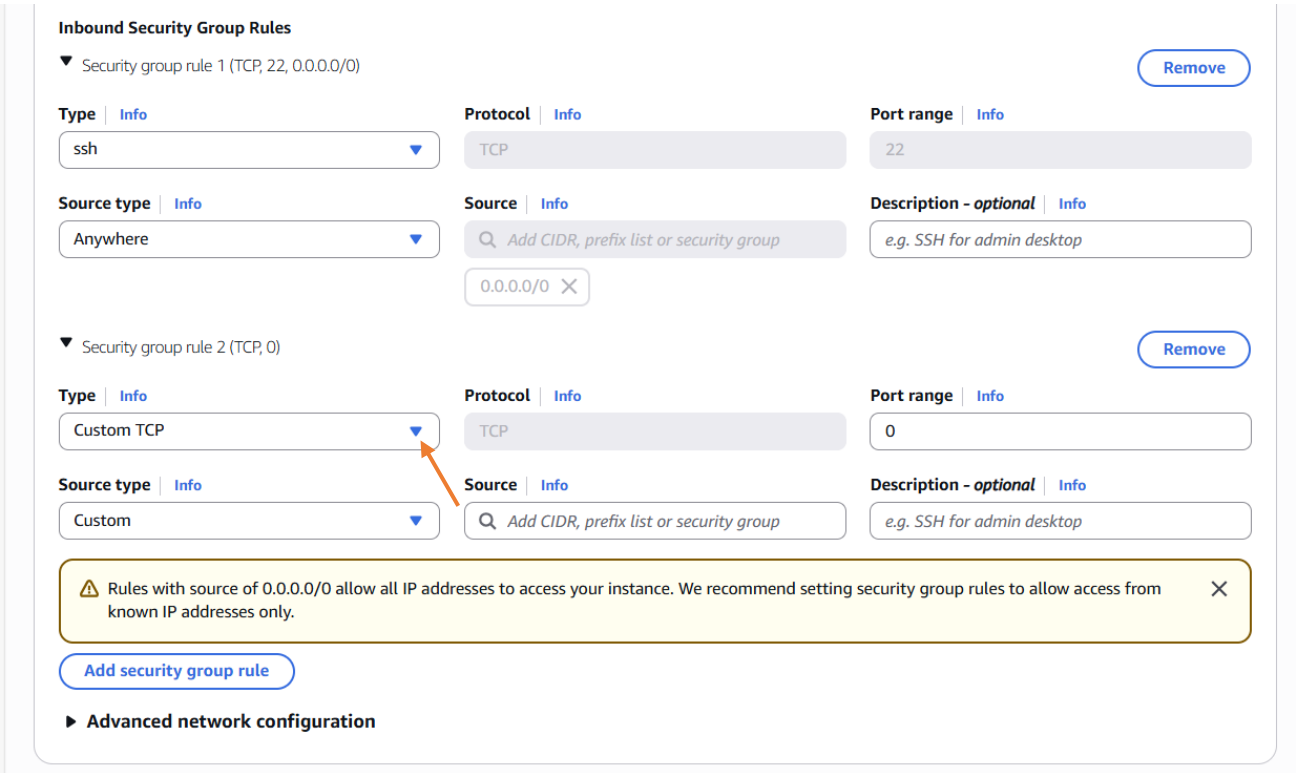
The subnet is already selected. On "**auto-sign public IP**", select "**Enable**"



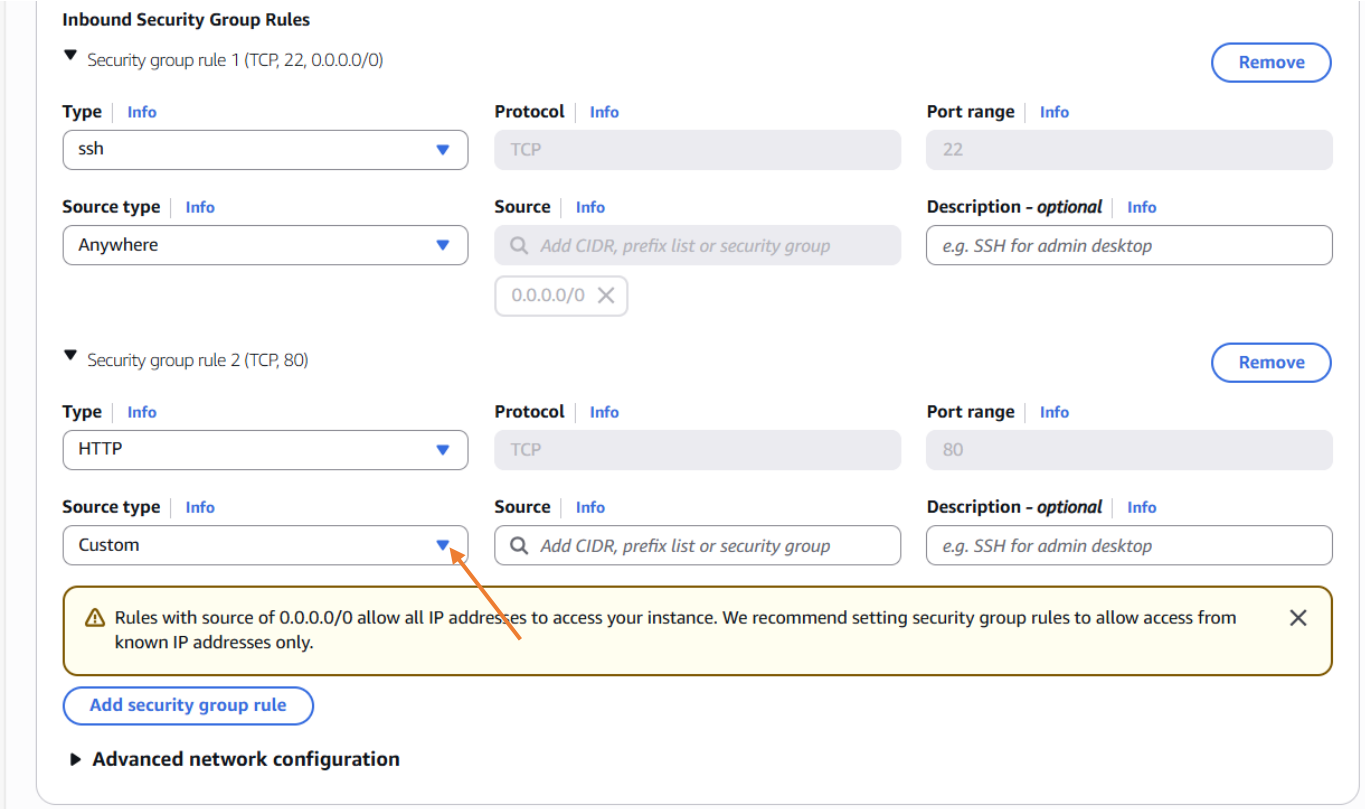
Scroll down to “**Inbound Security Group Rules**”



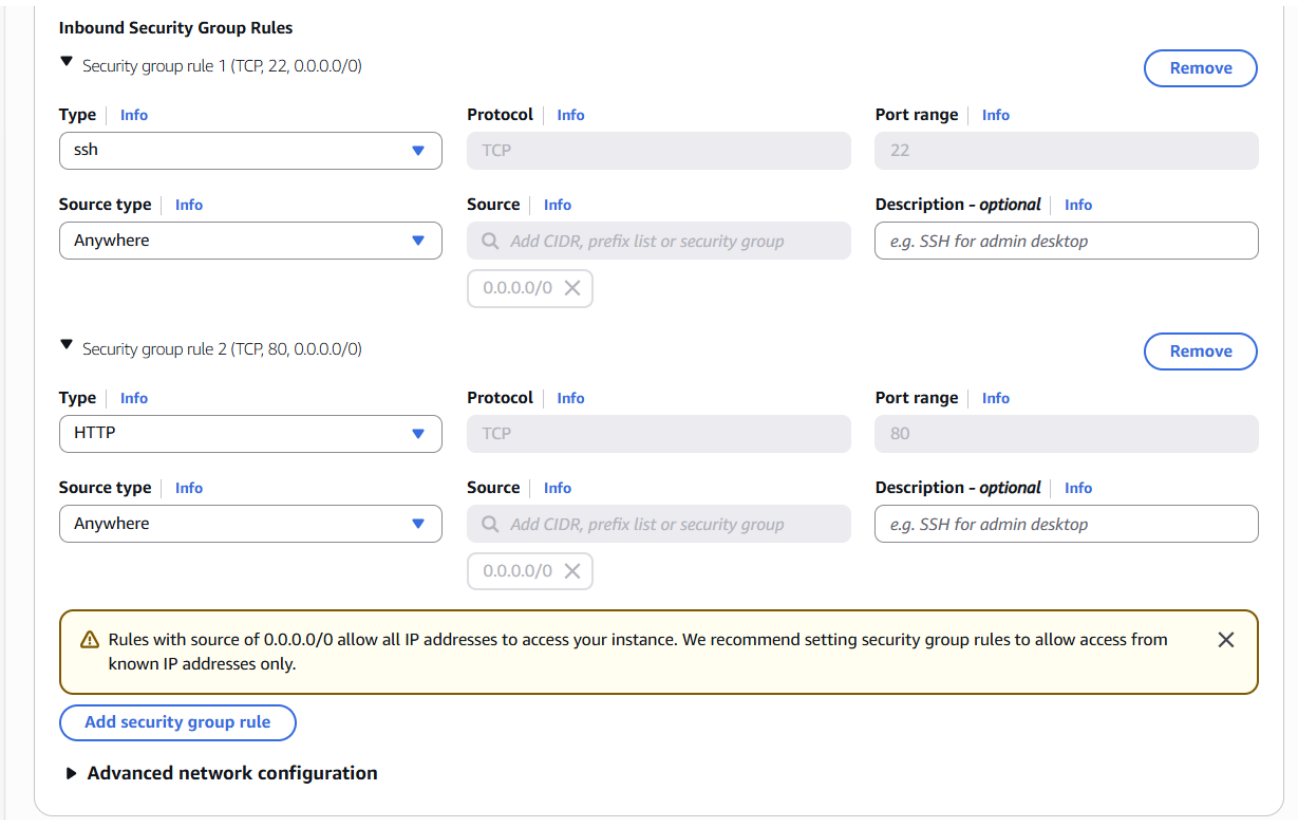
Click on “Add security group rule”



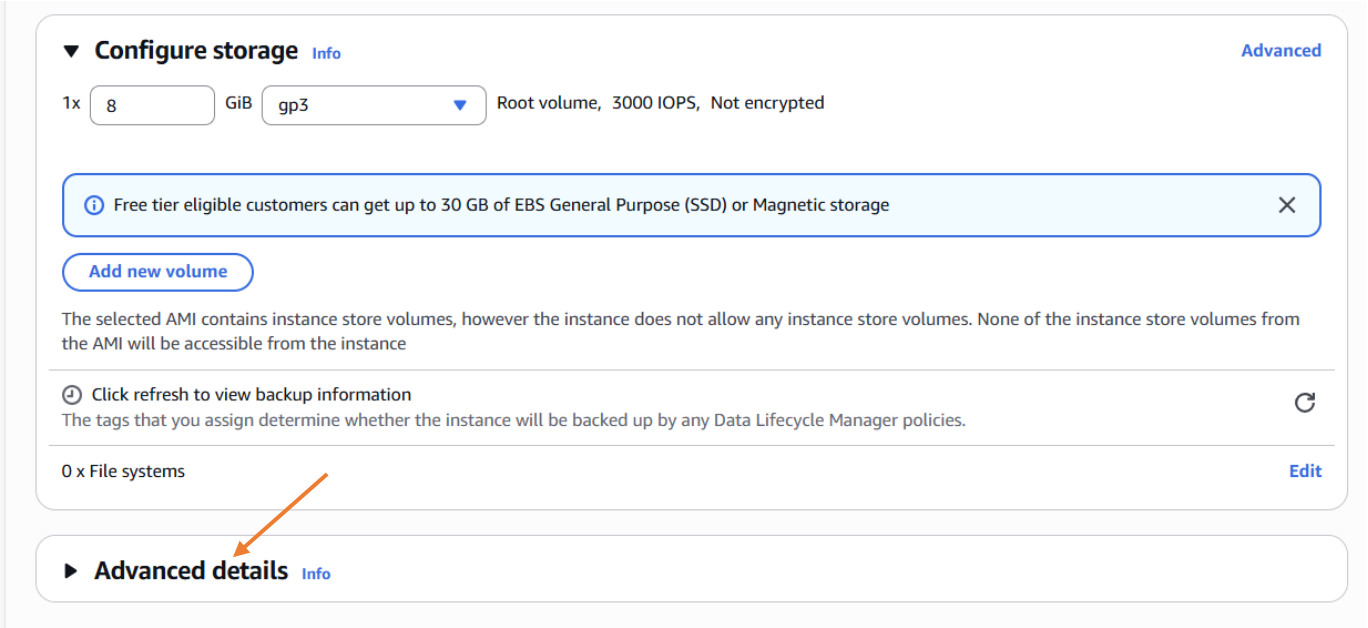
Click on the drop down on “**type**” and select “**HTTP**”



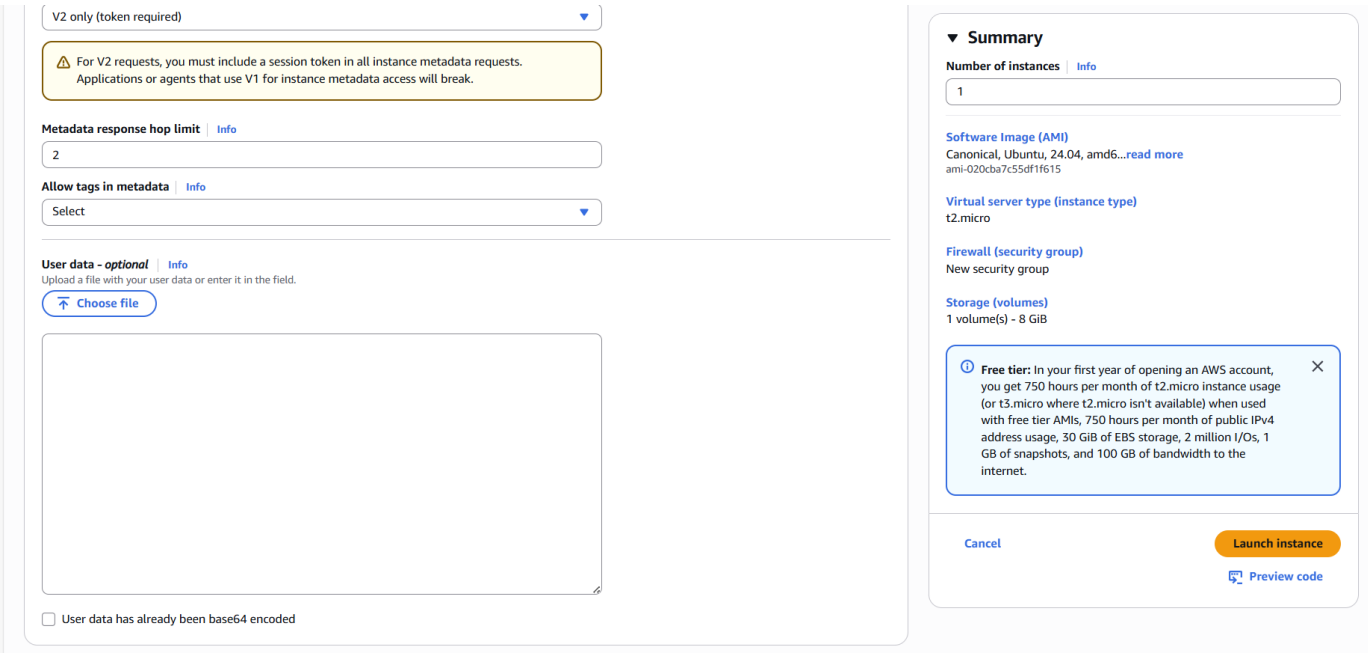
Click on “**Source Type**” and select “**anywhere**”



Scroll down to “Advanced Details”

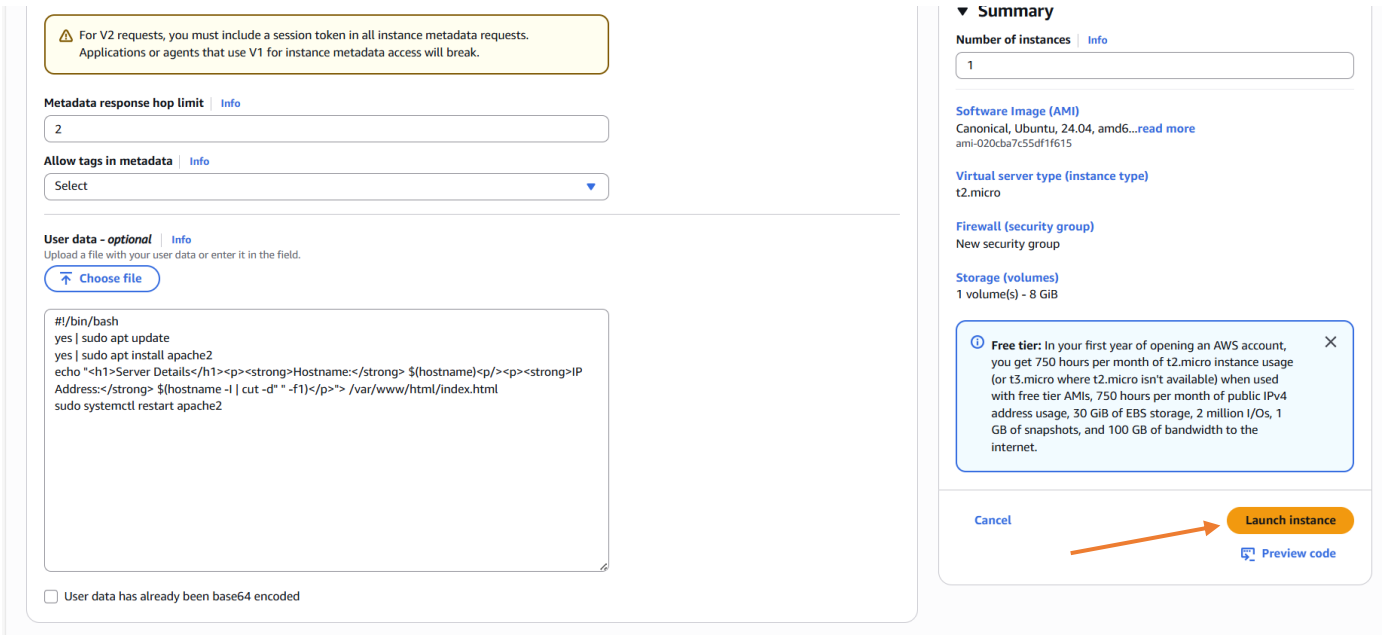


Click on “**Advanced Details**” and scroll down to “**User Data**”

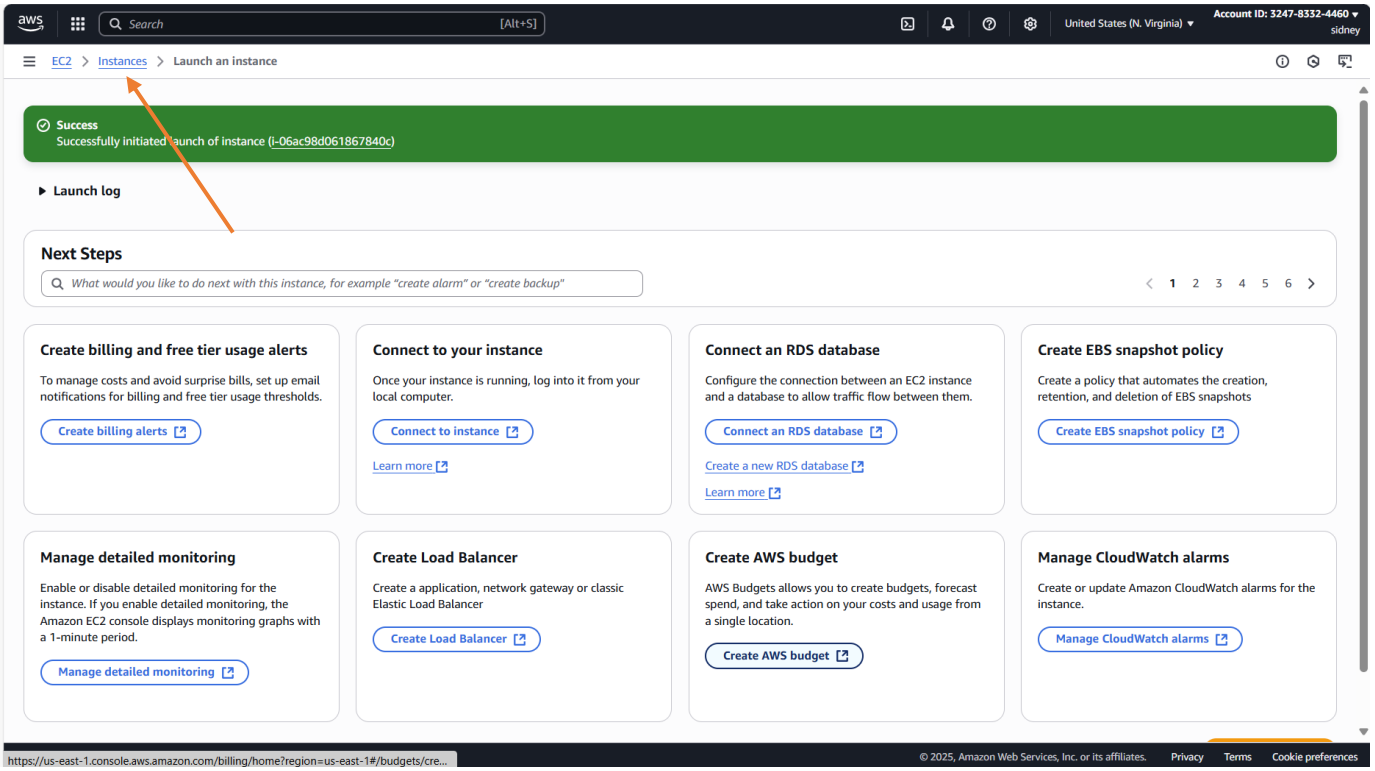


Paste the code below in the “**User Data**”

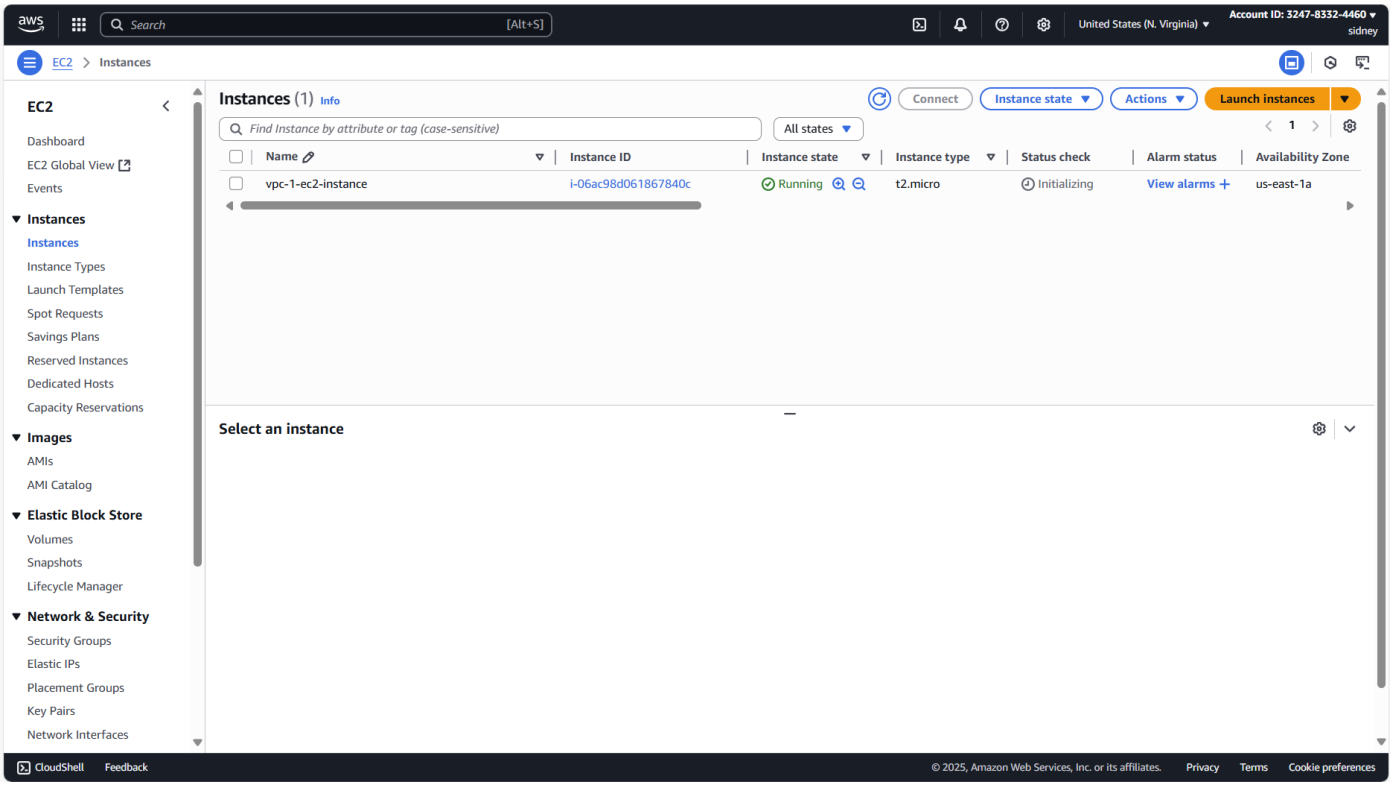
```
#!/bin/bash
yes | sudo apt update
yes | sudo apt install apache2
echo "<h1>Server Details</h1><p><strong>Hostname:</strong> $(hostname)<p/><p><strong>IP
Address:</strong> $(hostname -I | cut -d" " -f1)</p>"> /var/www/html/index.html
sudo systemctl restart apache2
```



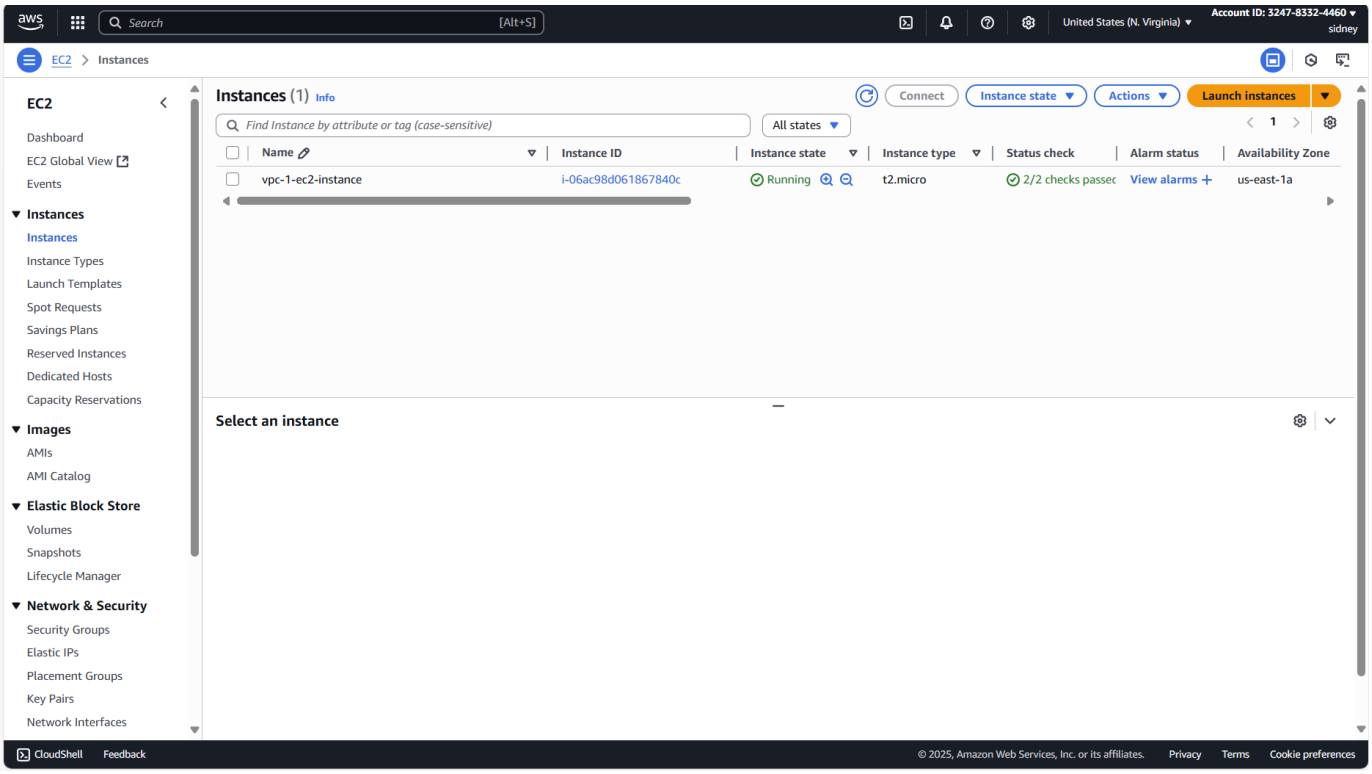
Click on “**Launch Instance**”



Click on “**Instances**”

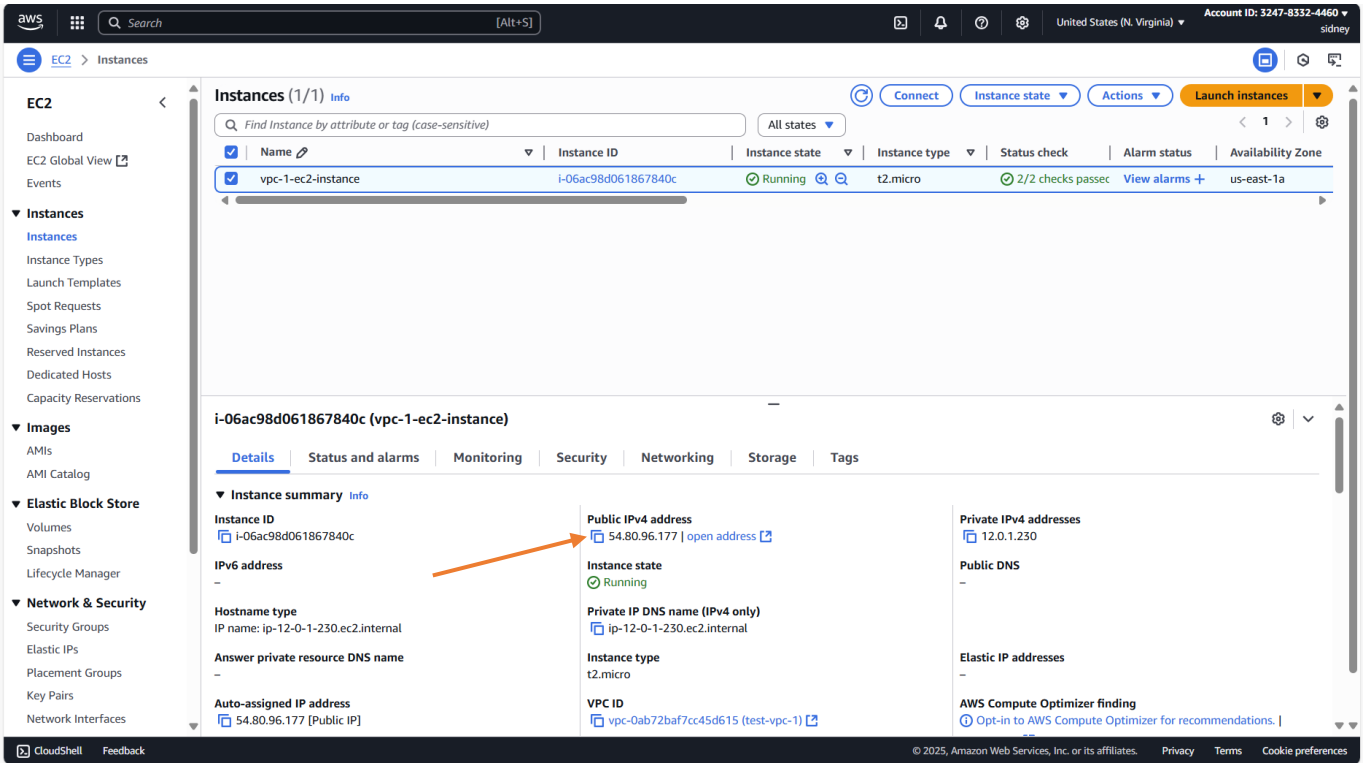


Wait for the instance to pass the "**2/2 check**"

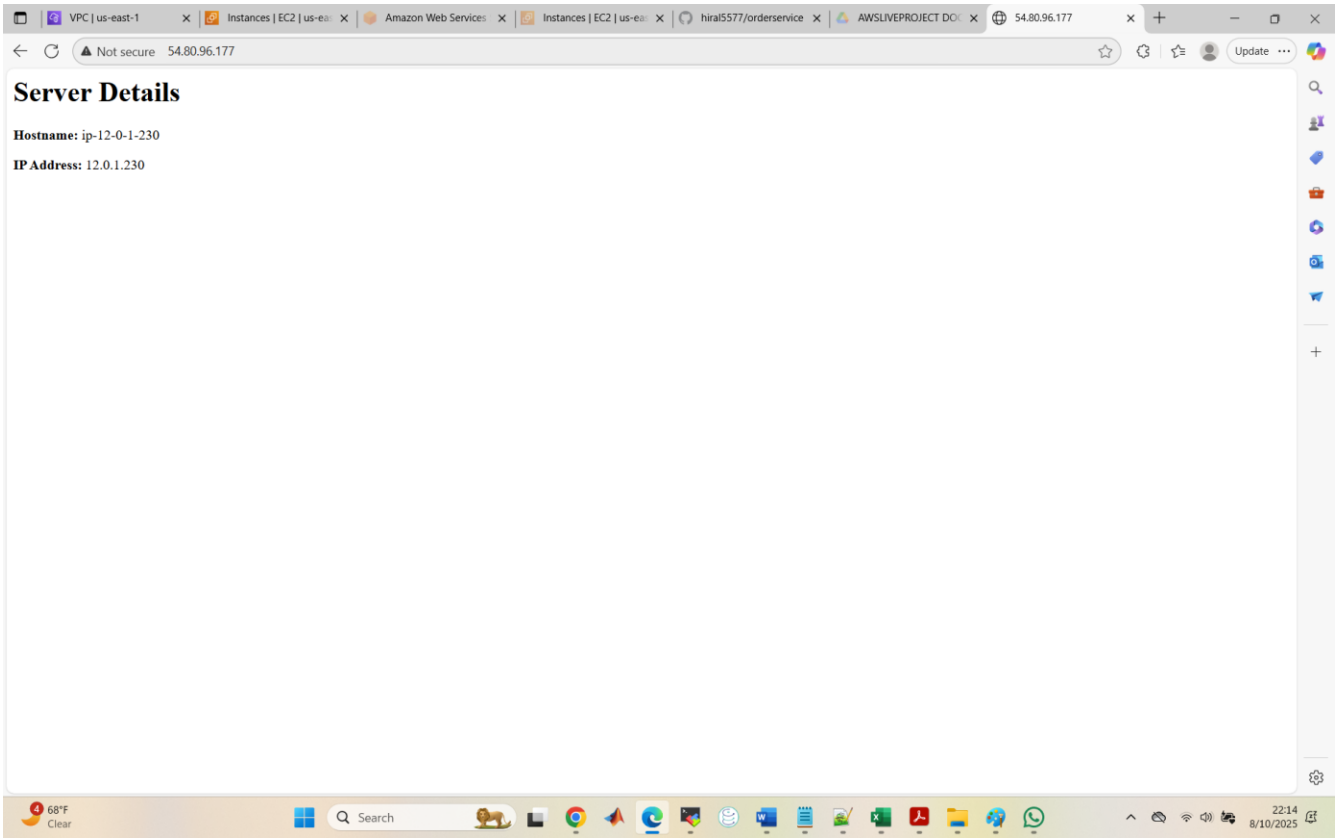


The "**2/2 check**" has passed. Let us check if the Apache has been installed successfully.

Select the EC2 instance



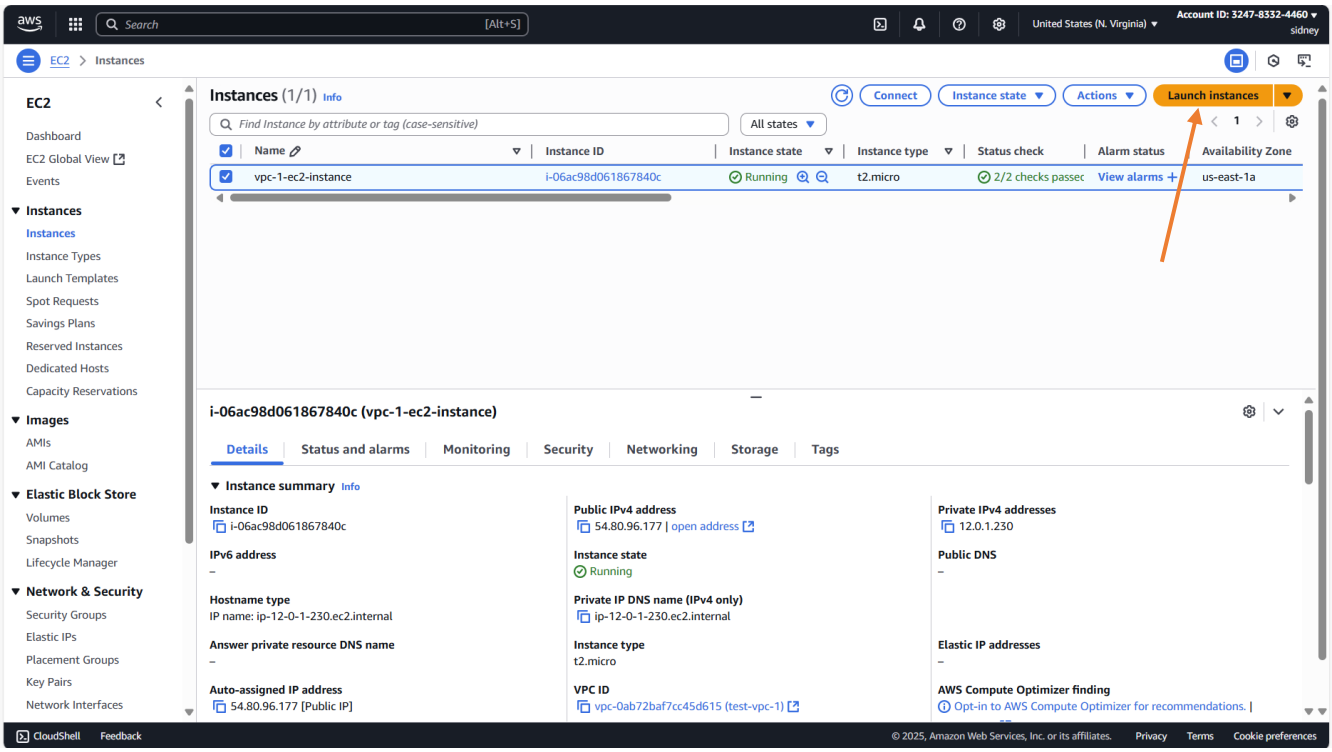
Copy the "**Public IPv4 address**" and test on your browser



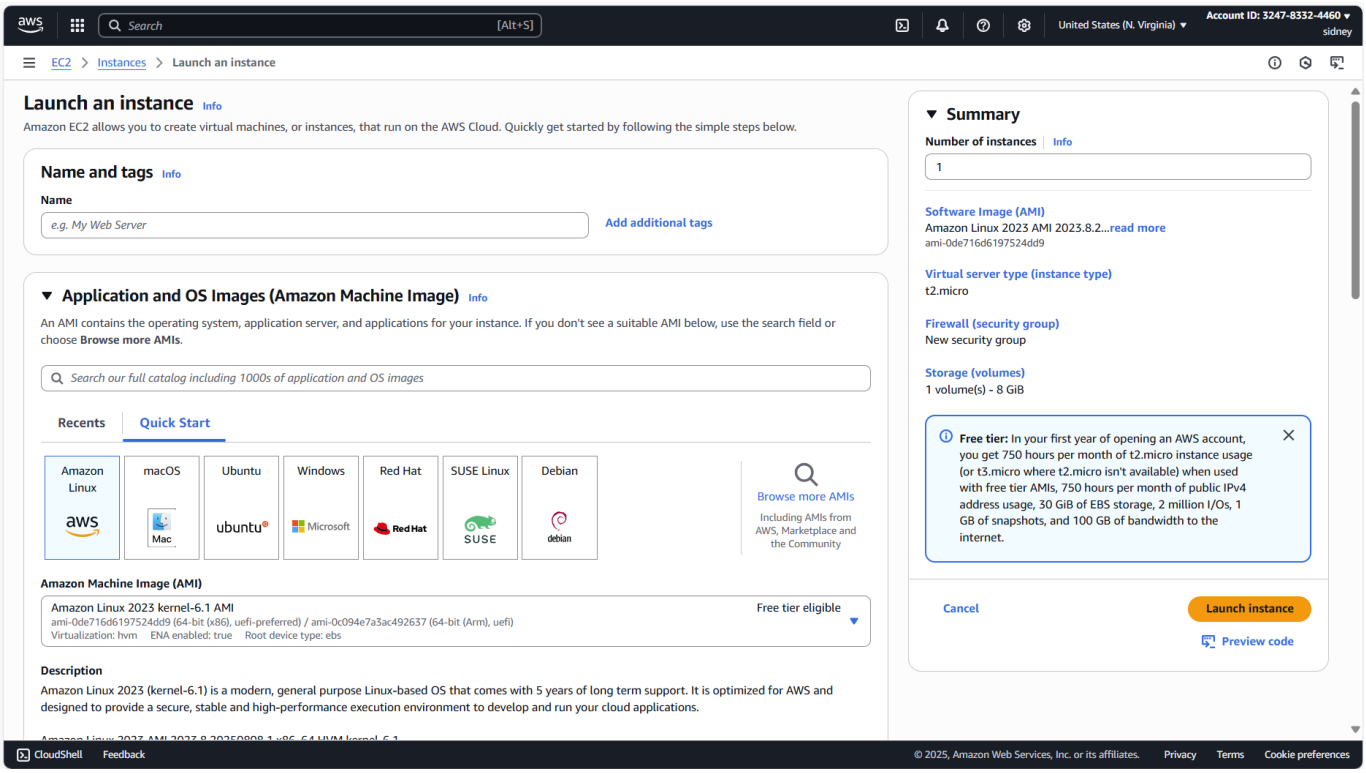
You can see that the Apache has been installed successfully.

For VPC peering, we are going to do the SSH and then we are going to check a few more things later.

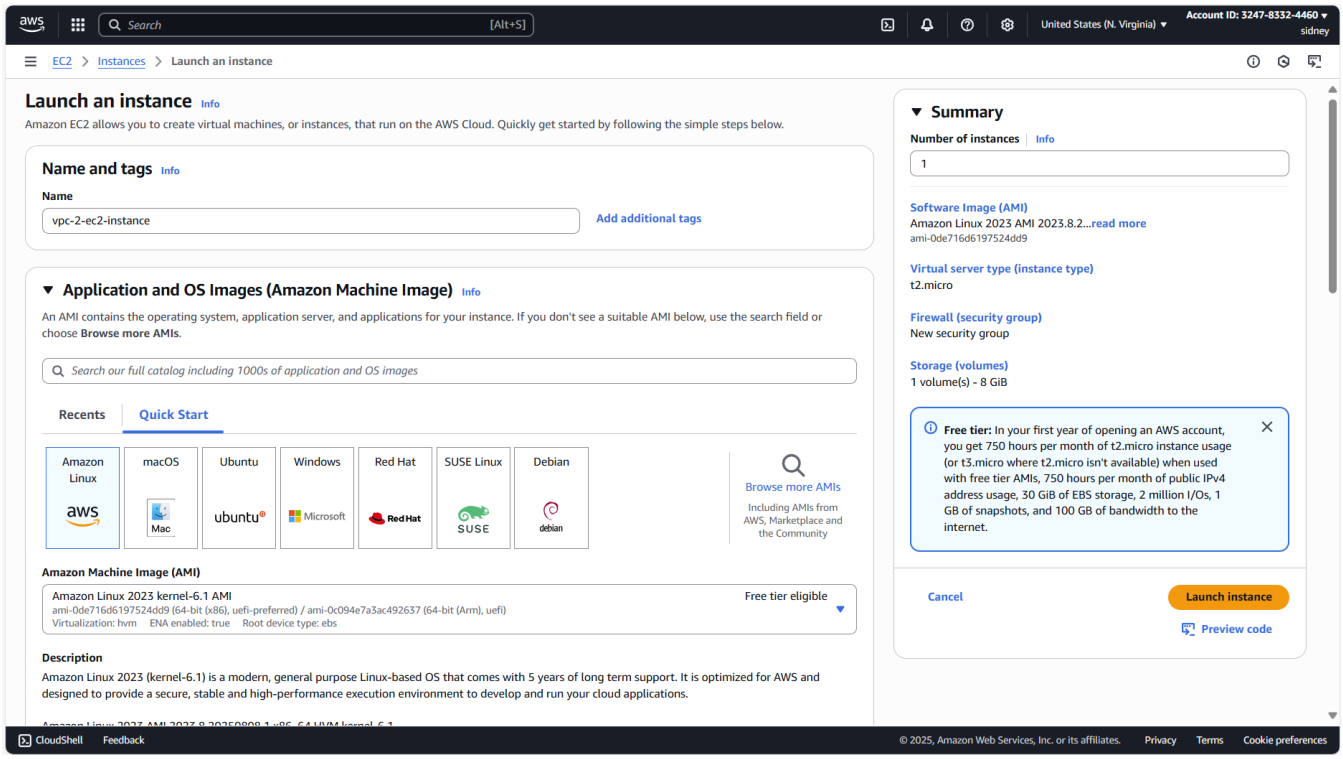
Now, we have to create another EC2 instance in **test-vpc-2**.



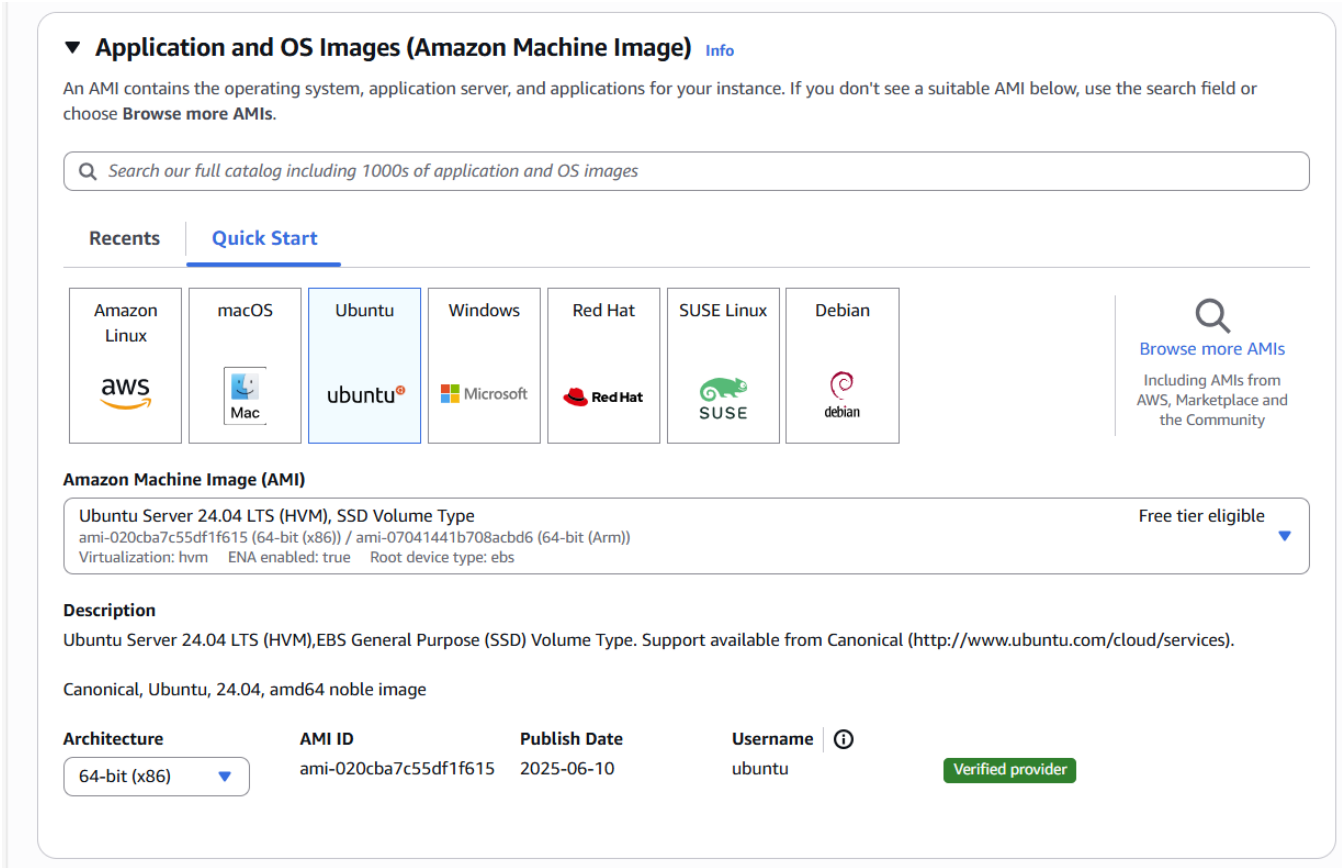
Click on "**Launch Instance**"

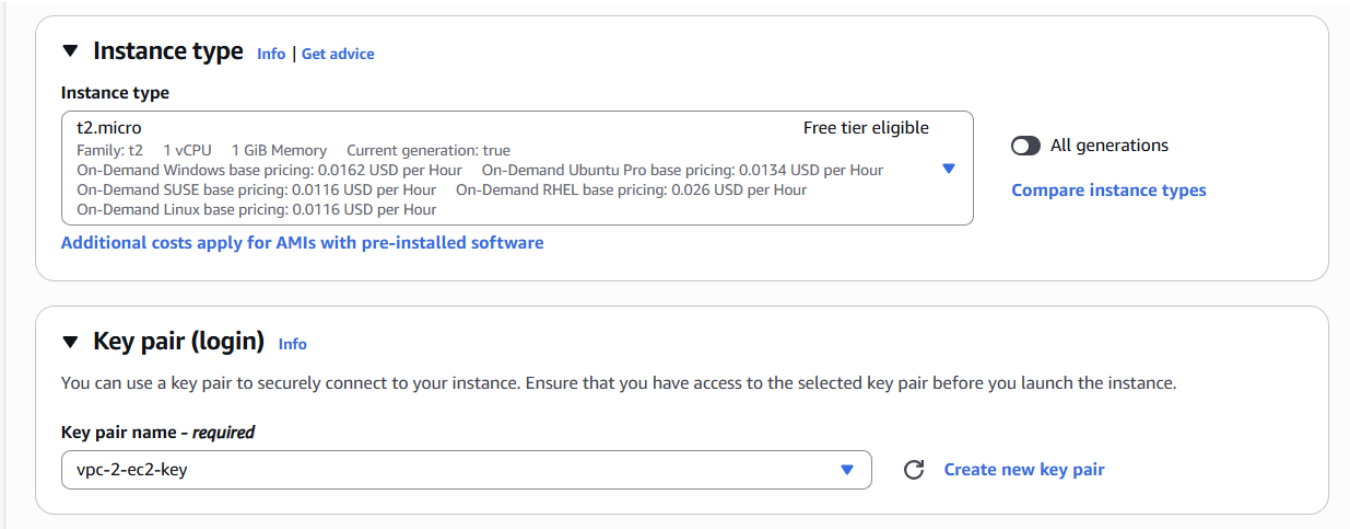


We will call the EC2 instance "**vpc-2-ec2-instance**"

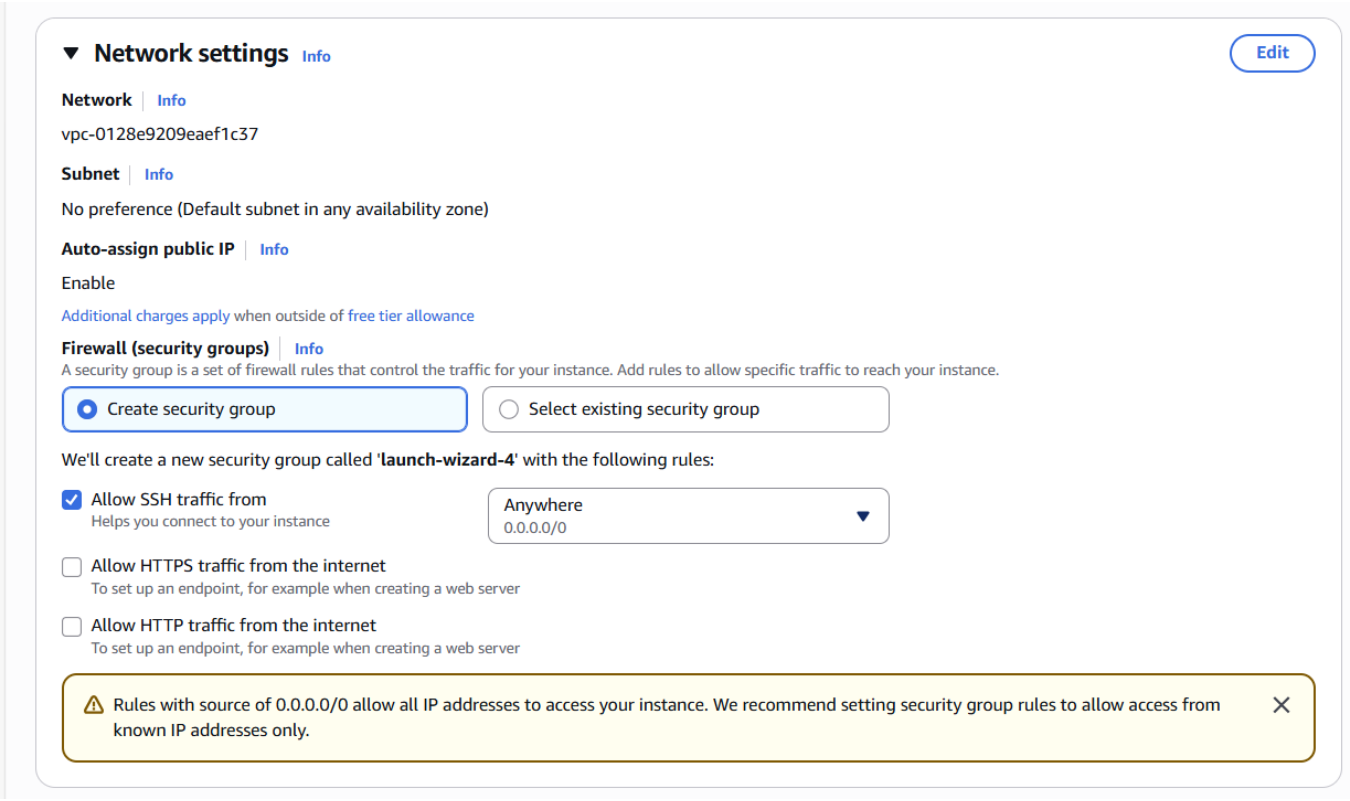


Then follow the step as we did in the other EC2 instance.

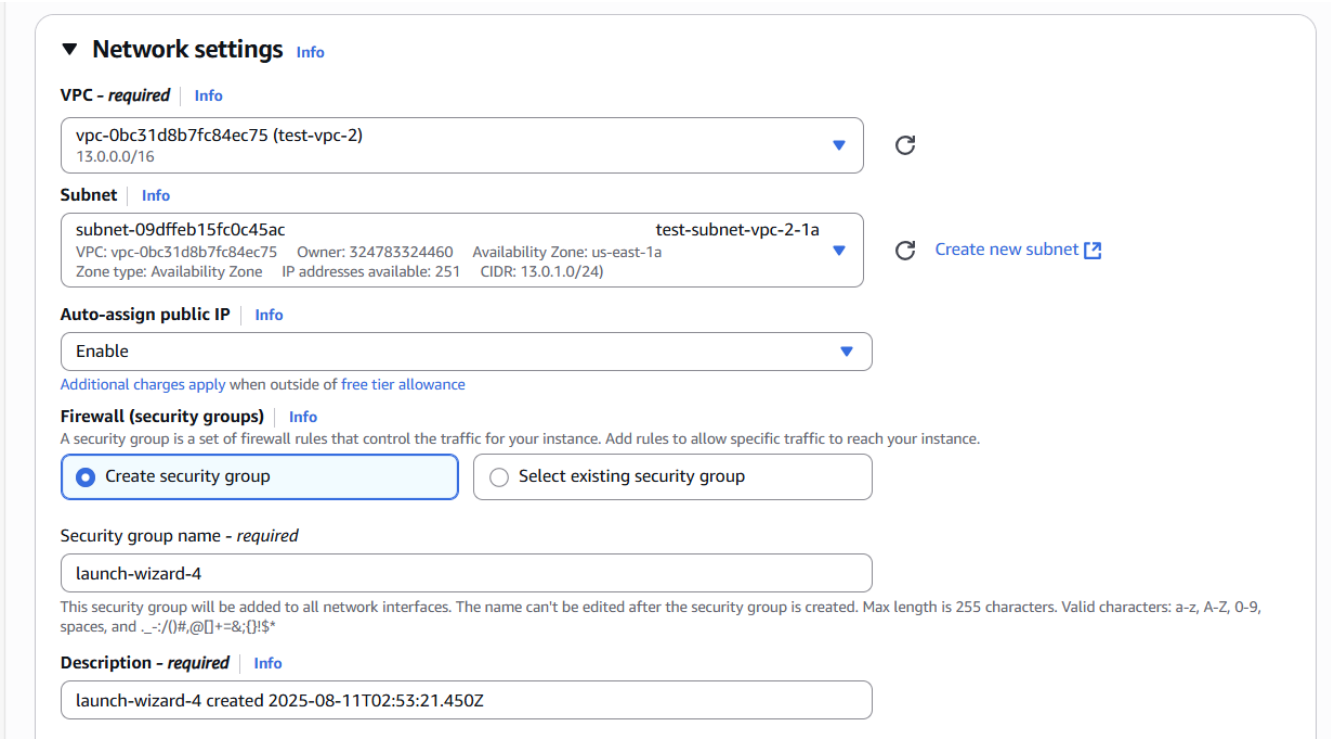




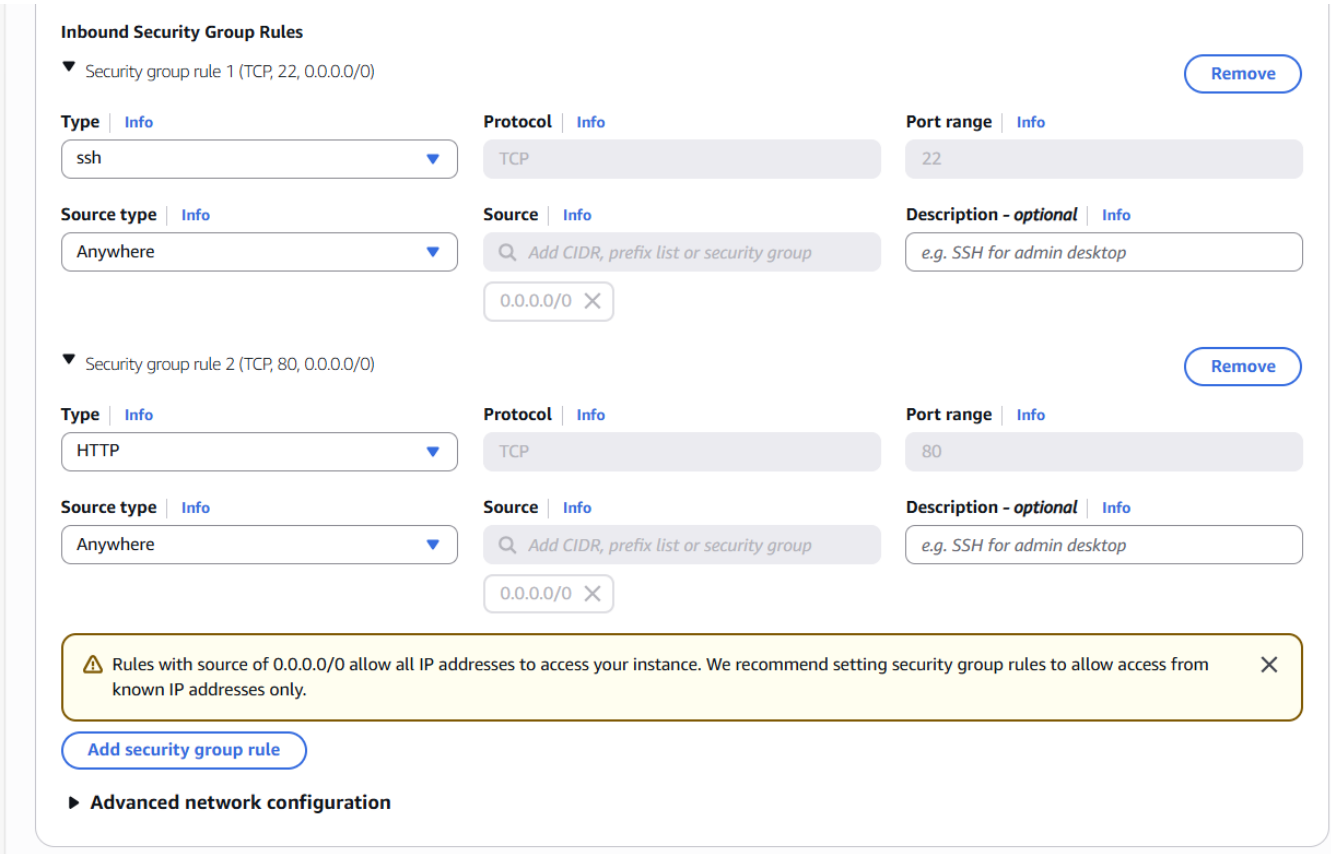
Scroll down to "**Network Settings**"



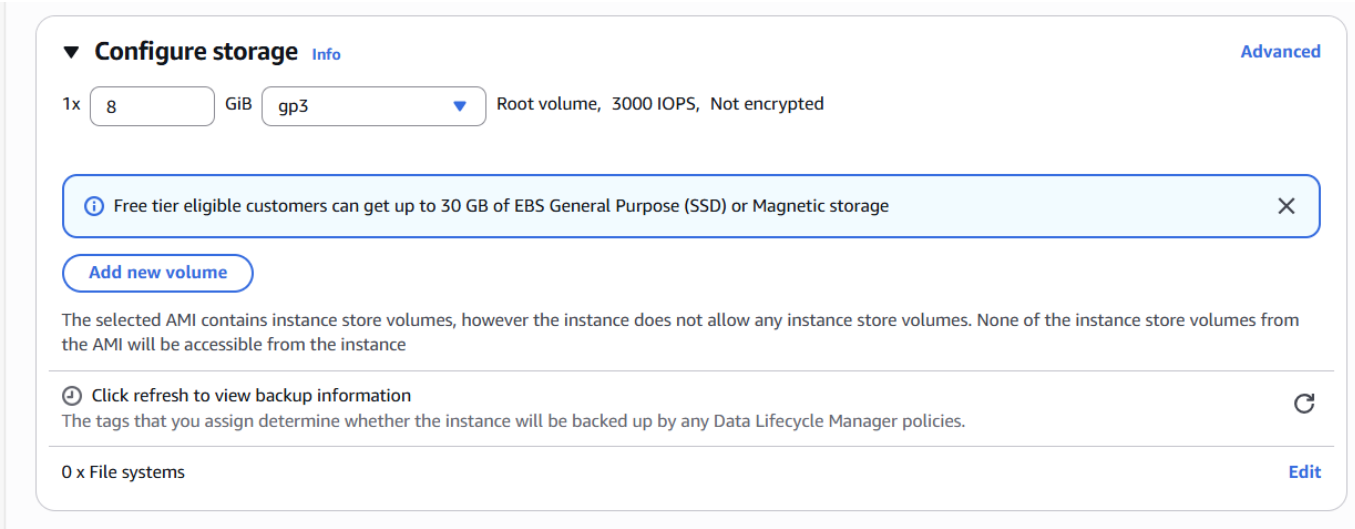
Click on "**Edit**"



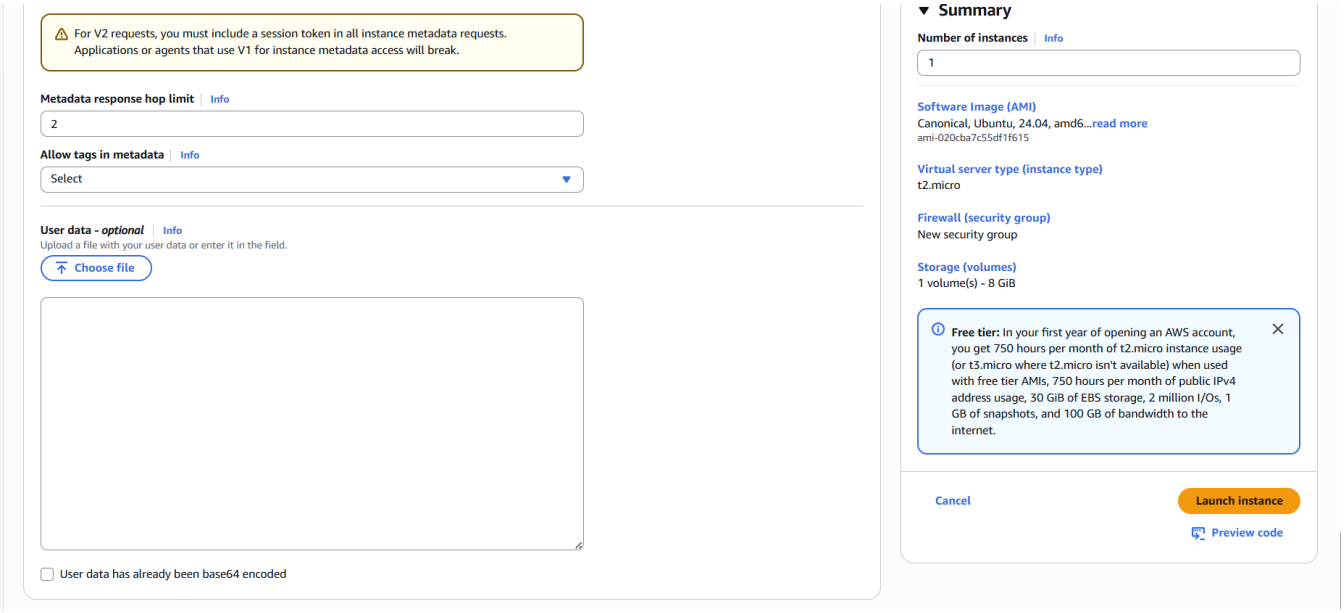
Scroll down



Scroll down

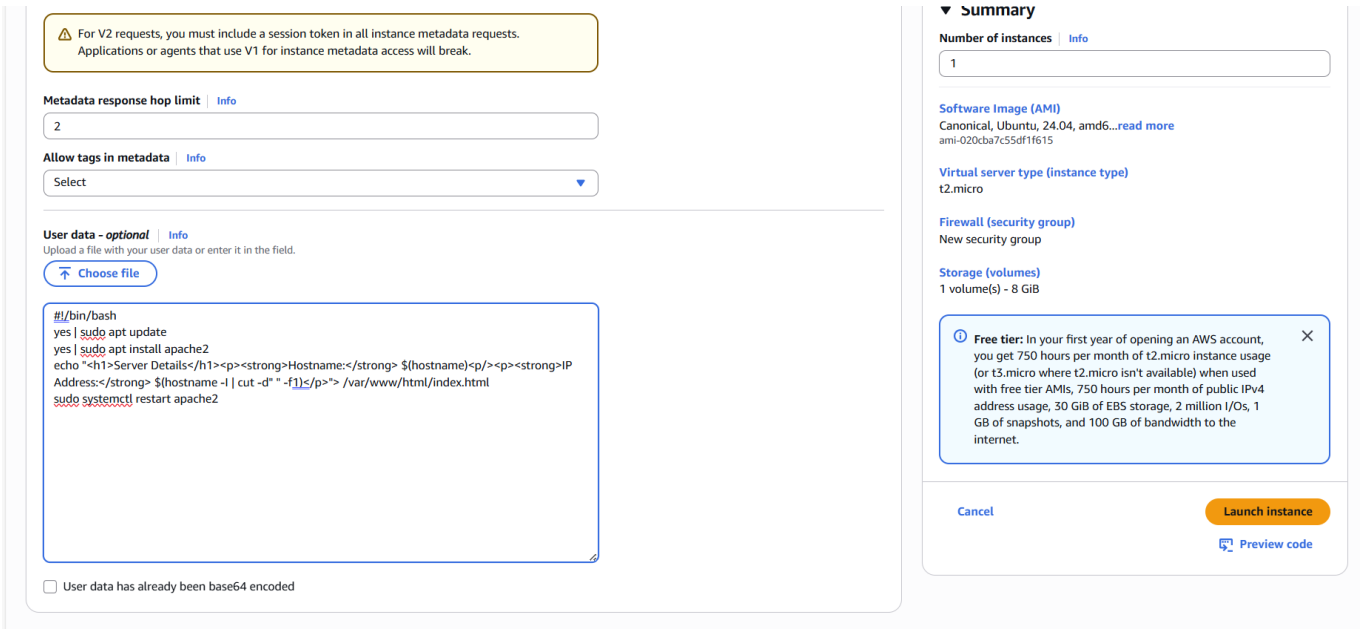


Scroll down to "**Advanced Details**" and click on it

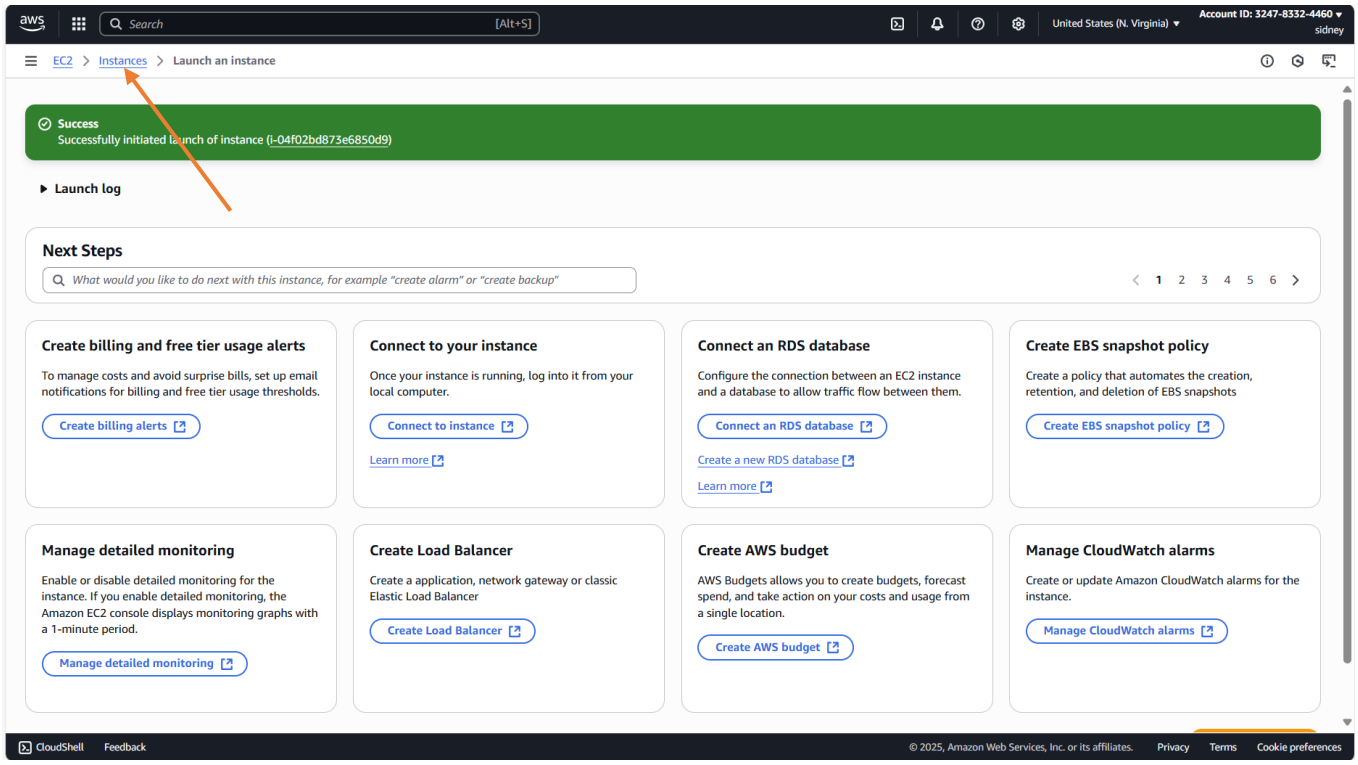


Paste the code below in the "**user data**"

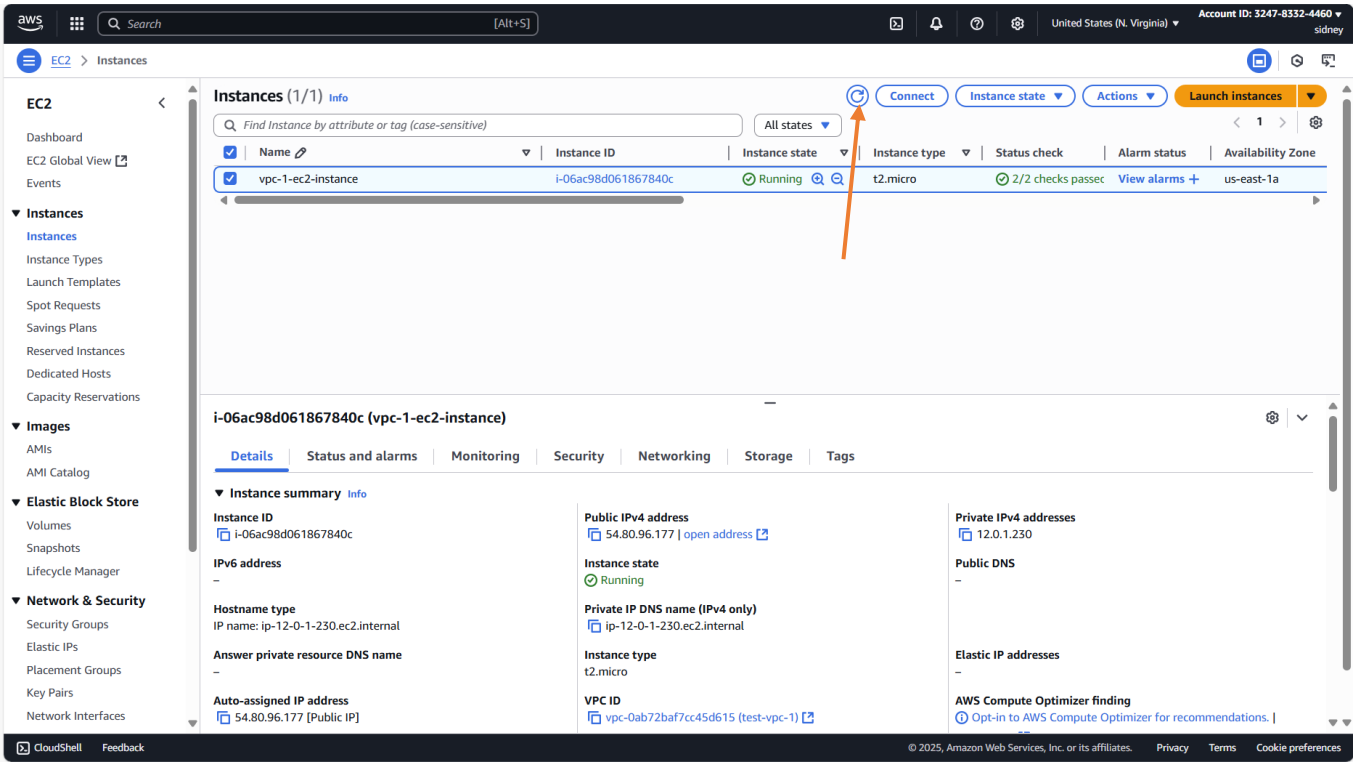
```
#!/bin/bash
yes | sudo apt update
yes | sudo apt install apache2
echo "<h1>Server Details</h1><p><strong>Hostname:</strong>
$(hostname)<p/><p><strong>IP Address:</strong> $(hostname -I | cut -d" " -f1)</p>">
/var/www/html/index.html
sudo systemctl restart apache2
```



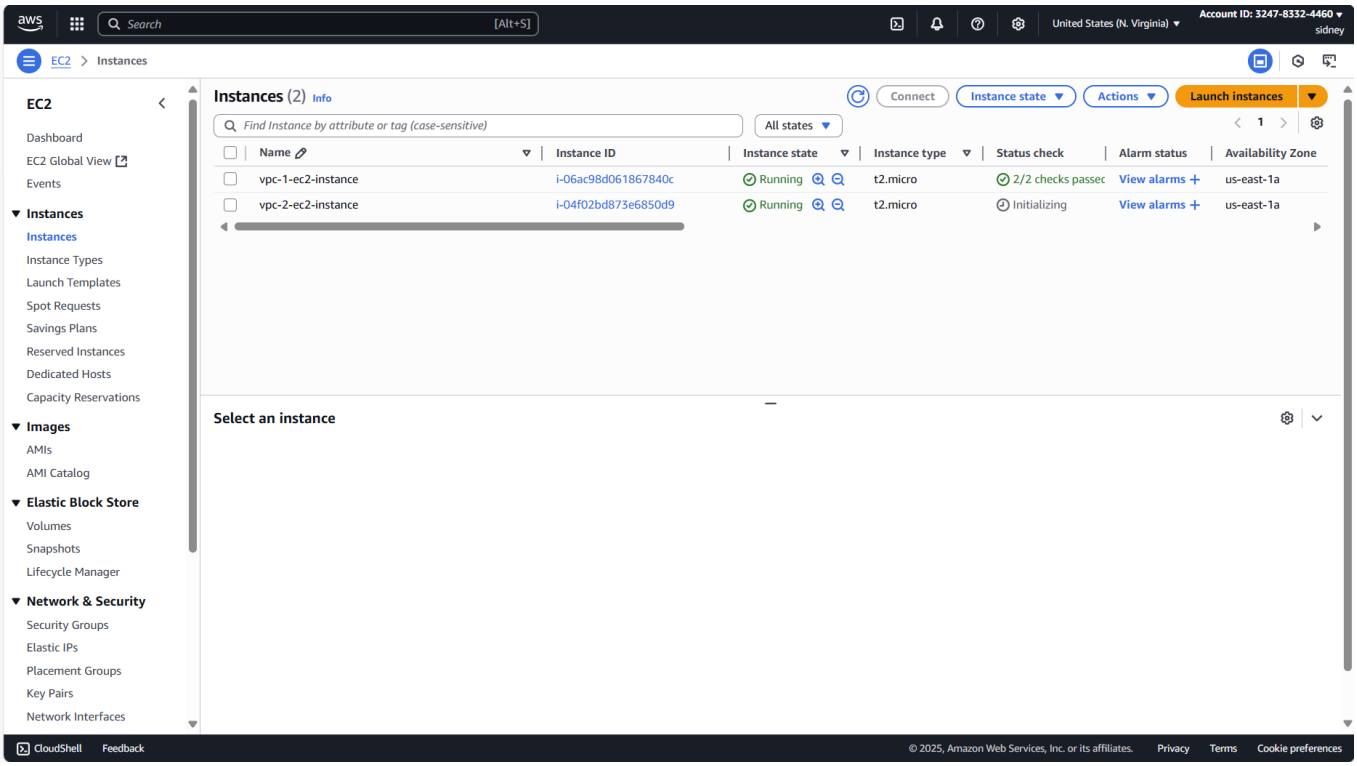
Click on “**Launch Instance**”



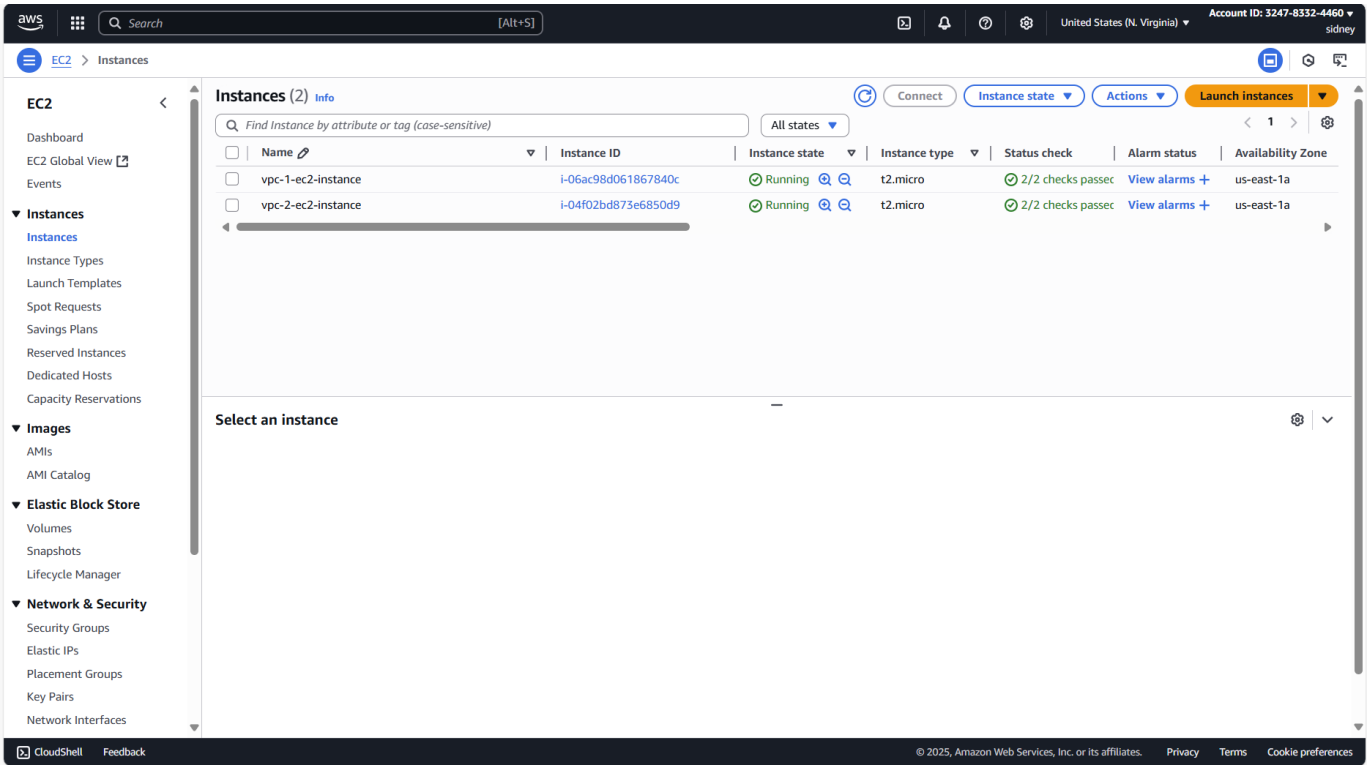
Click on “**Instances**”



Click on "**Refresh**"

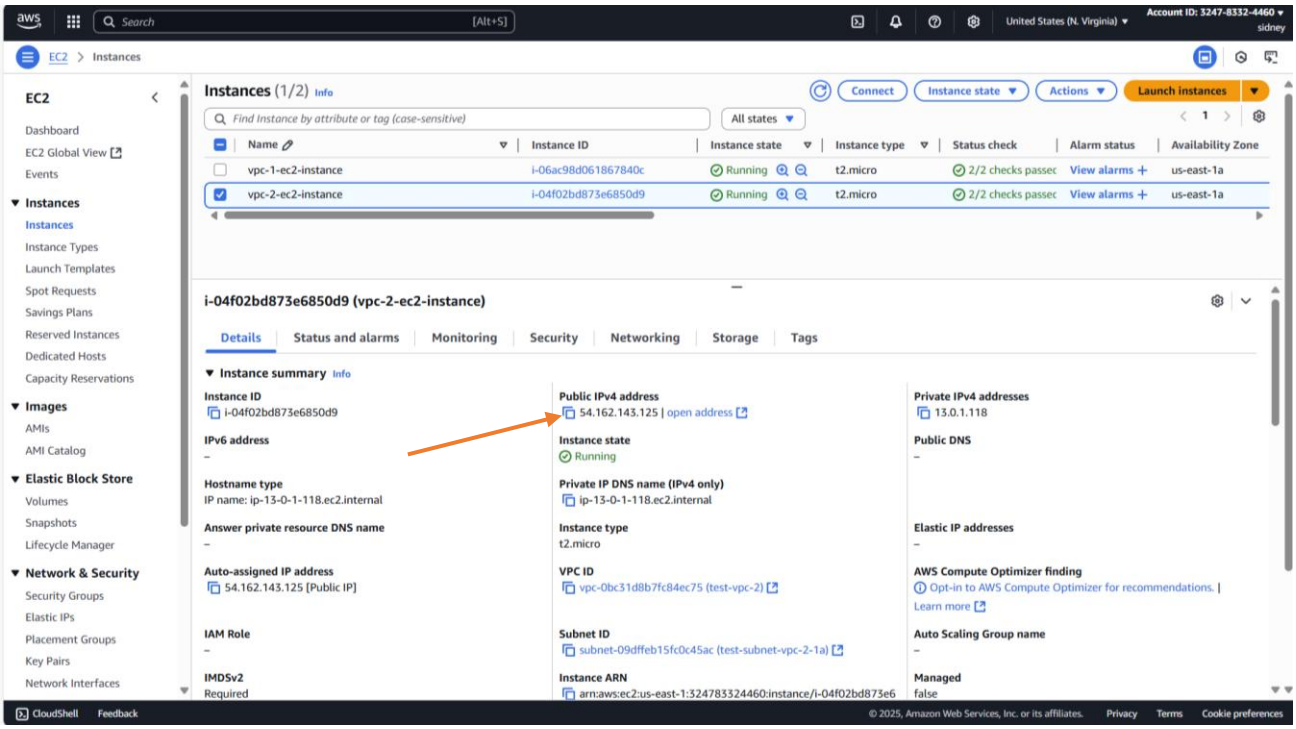


You can see that the second instance has been created. Let us wait for it to pass the "**2/2 check**"

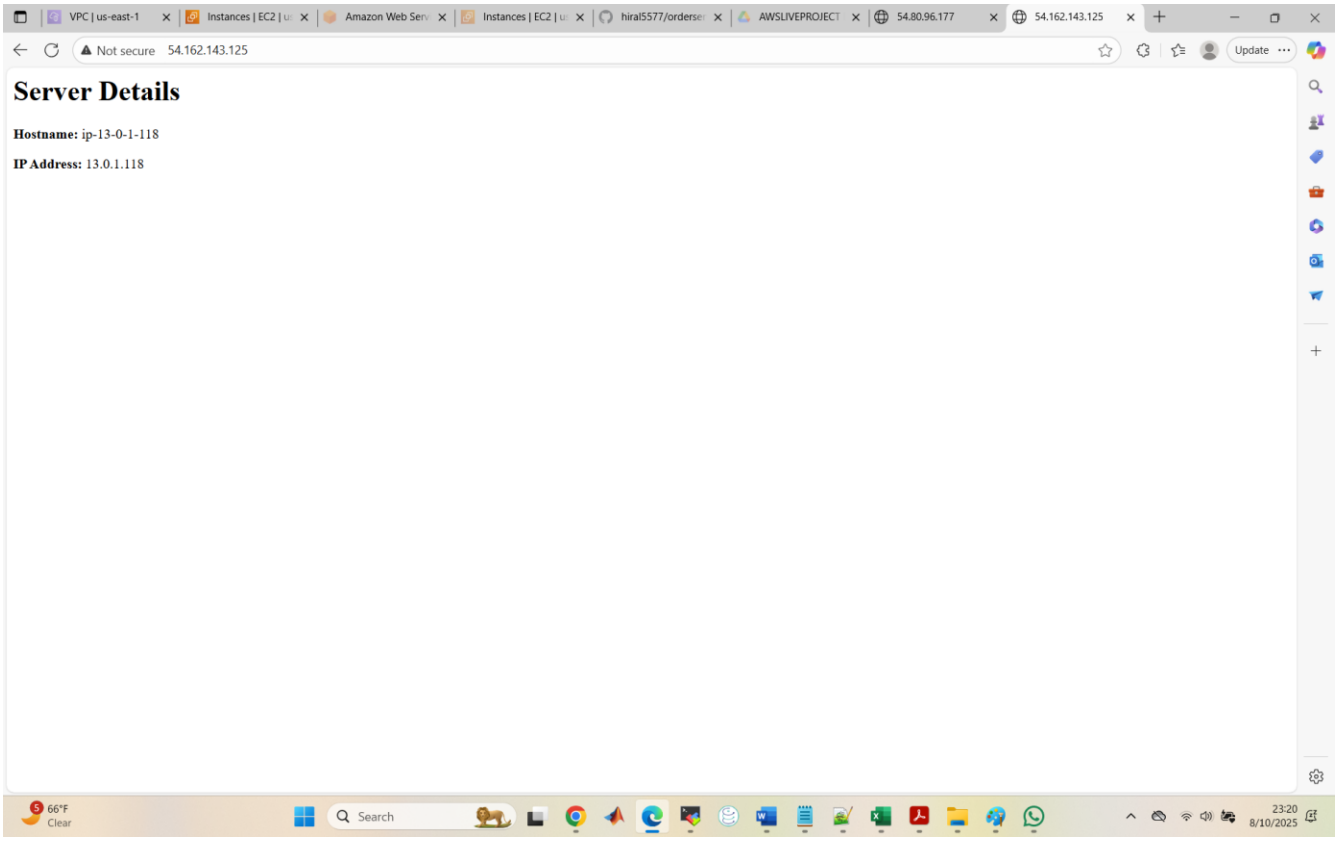


It has passed the "**2/2 check**". Let us check if the Apache has been installed successfully.

Select the EC2 instance

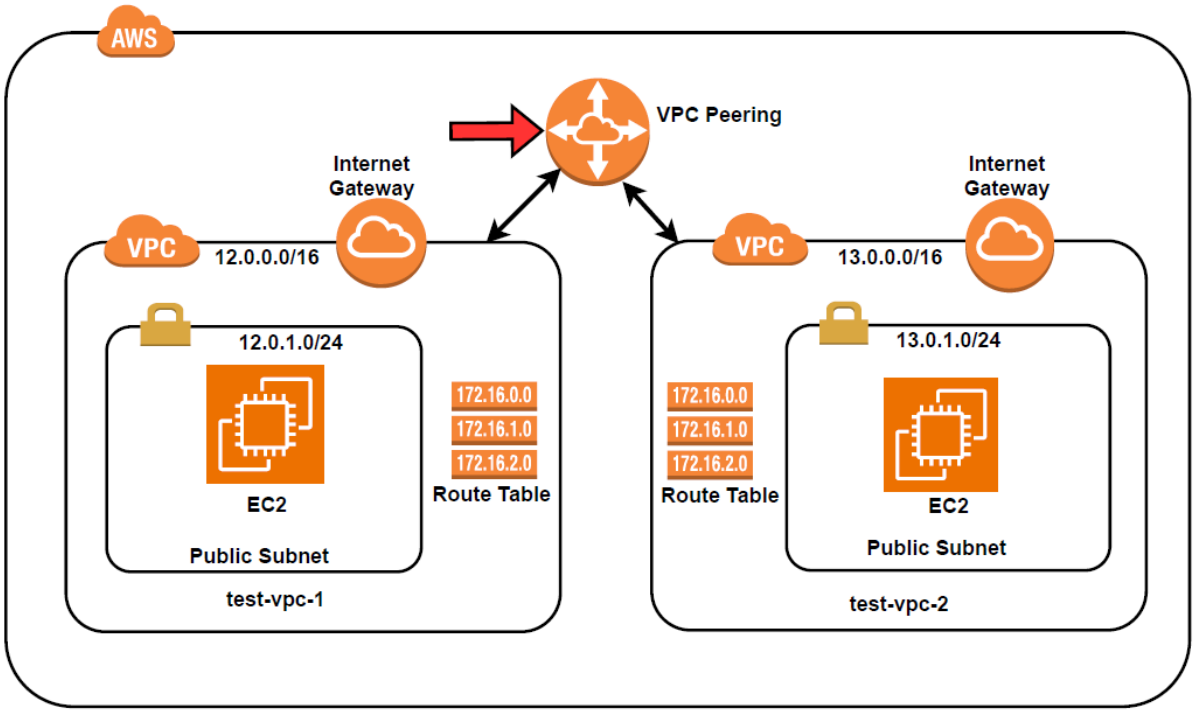


Copy the "**Public IPv4 address**" and test on your browser



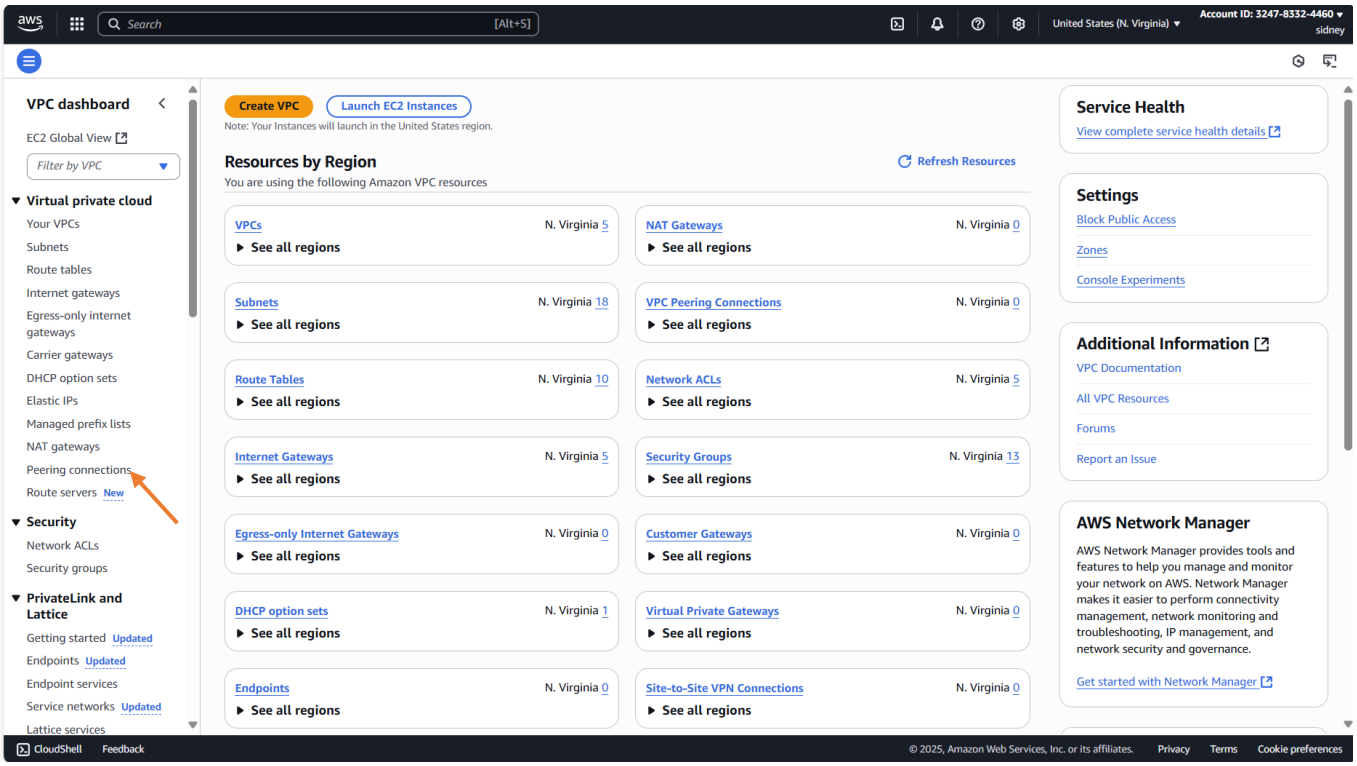
You can see that the Apache has been installed successfully.

Part 8: Setting up VPC Peering

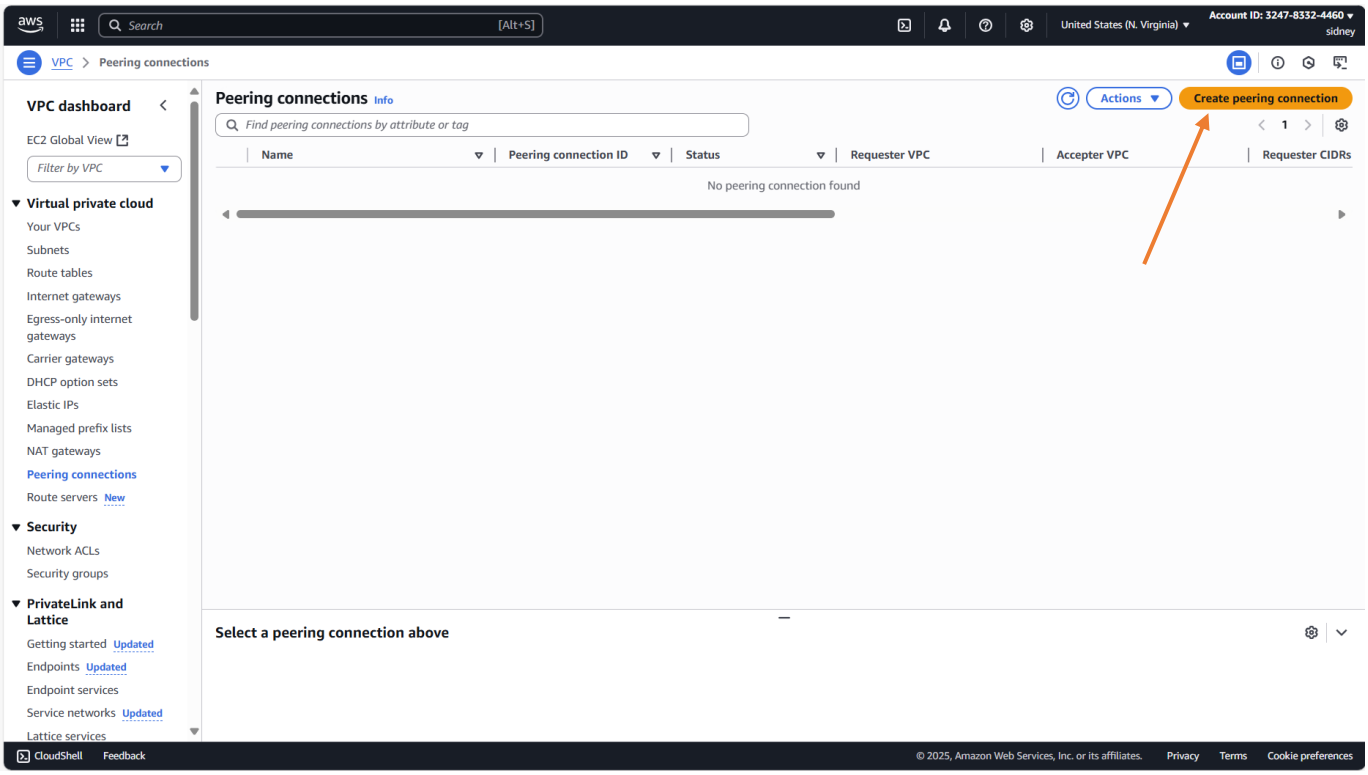


Now, we are going to create a VPC peering between the two VPCs.

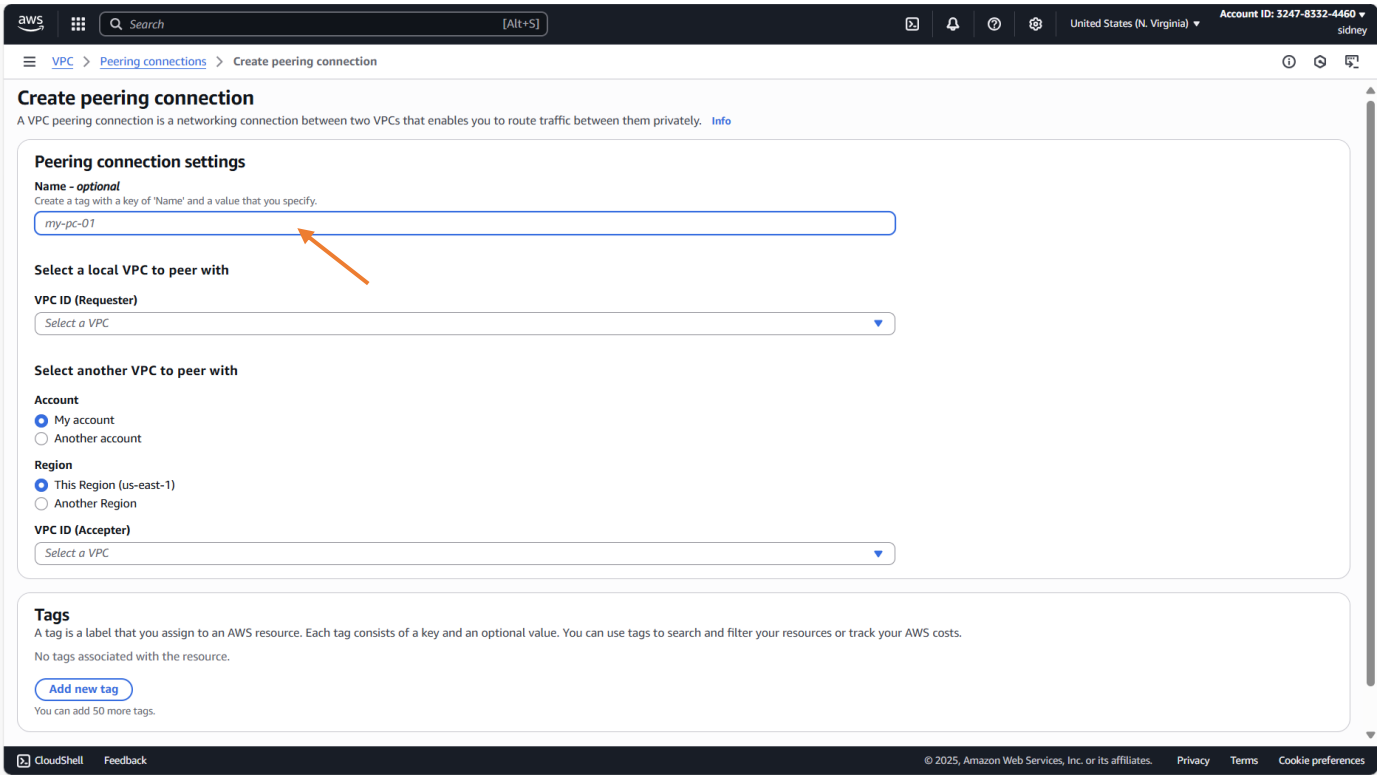
Go to VPC dashboard on AWS Management Console



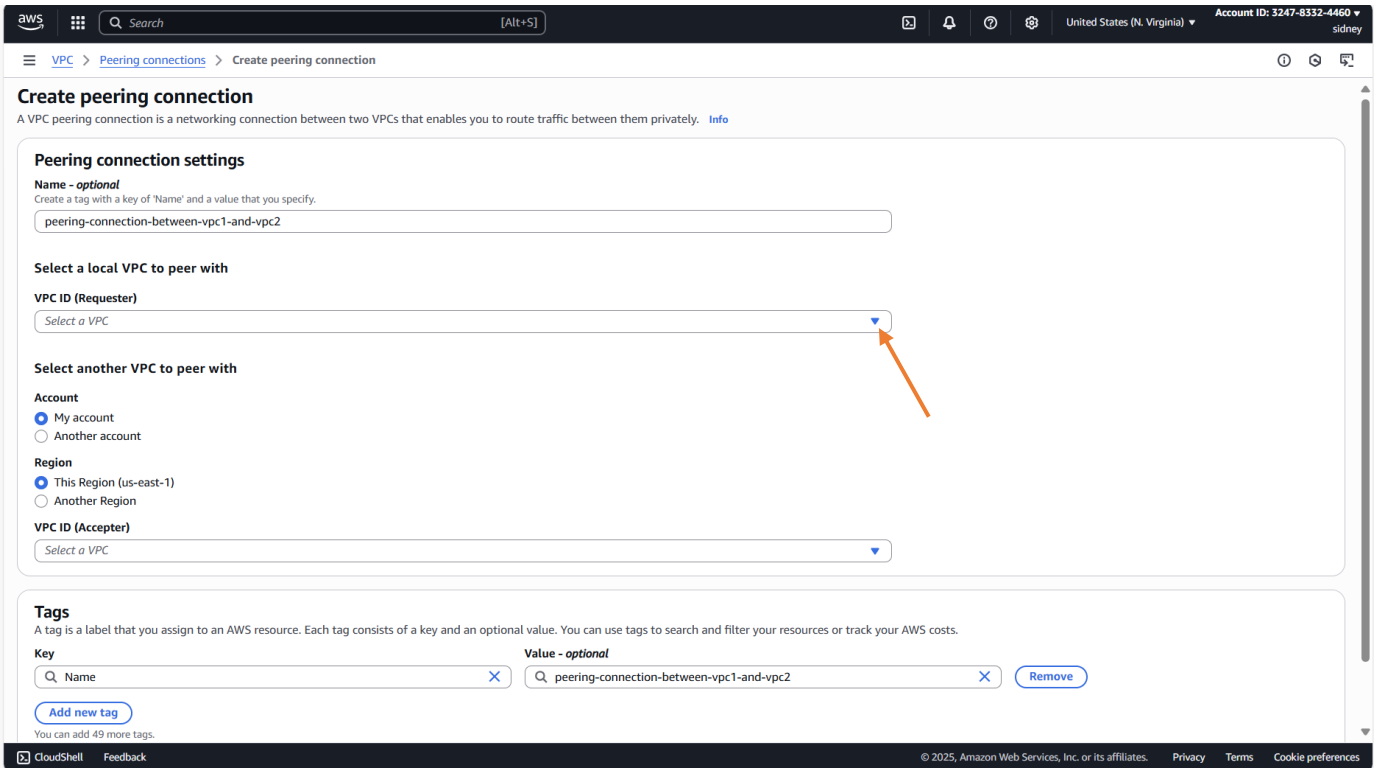
Click on “**Peering Connections**”



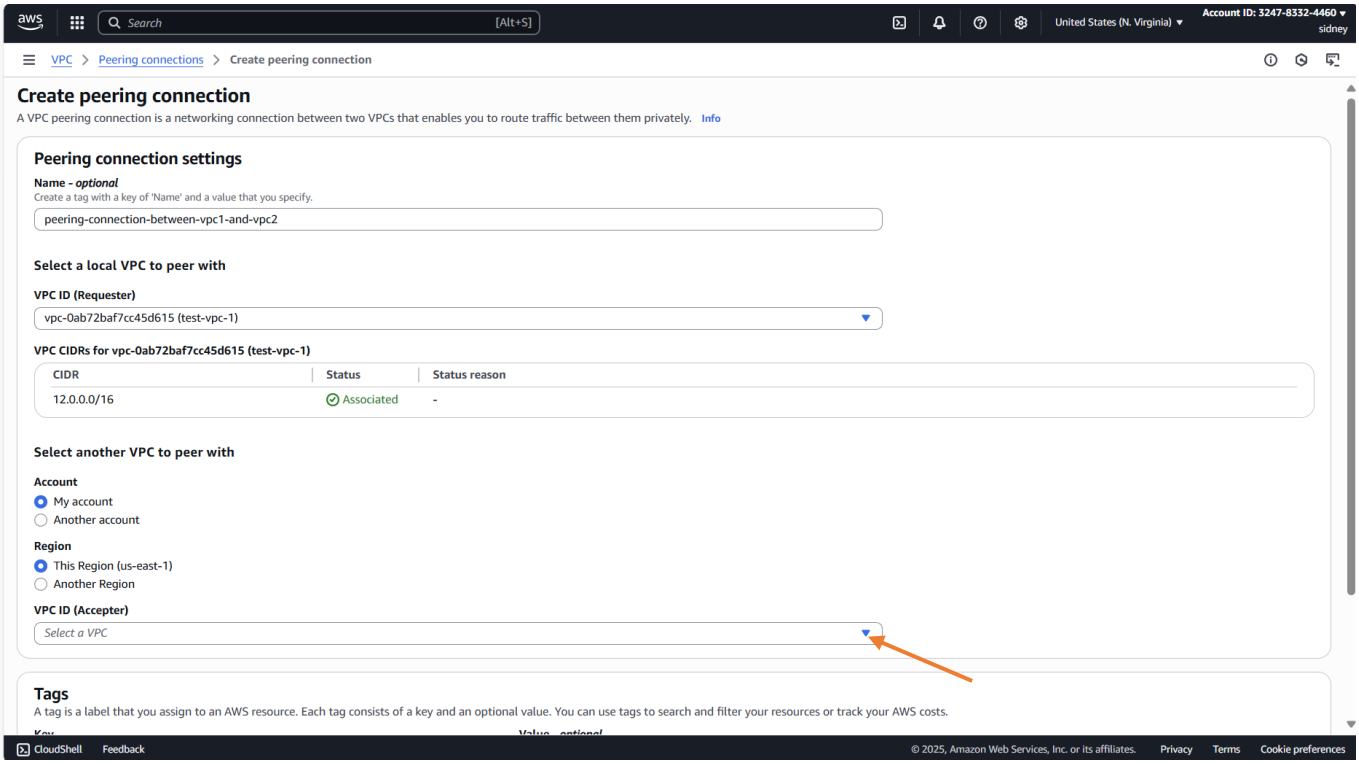
Click on “**create peering connection**”



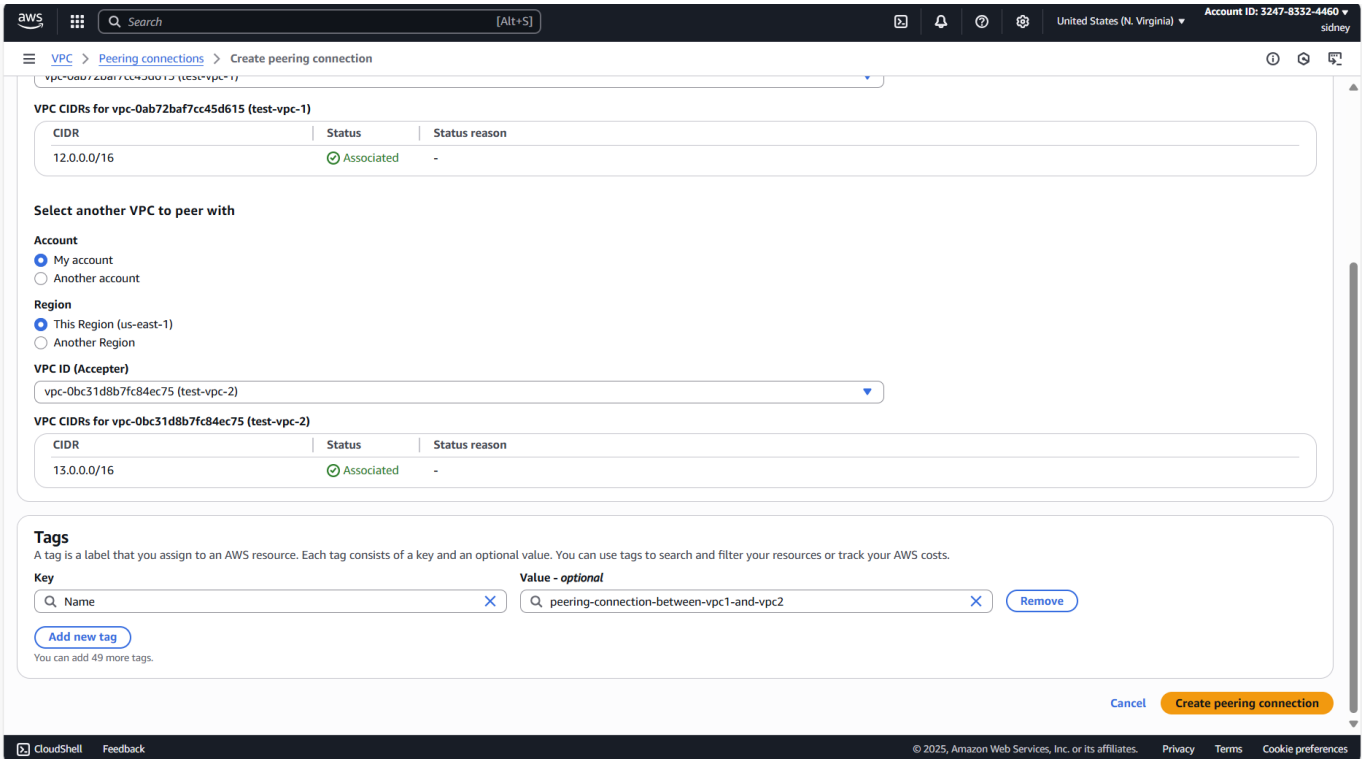
For the name, we will call it "**peering-connection-between-vpc1-and-vpc2**"



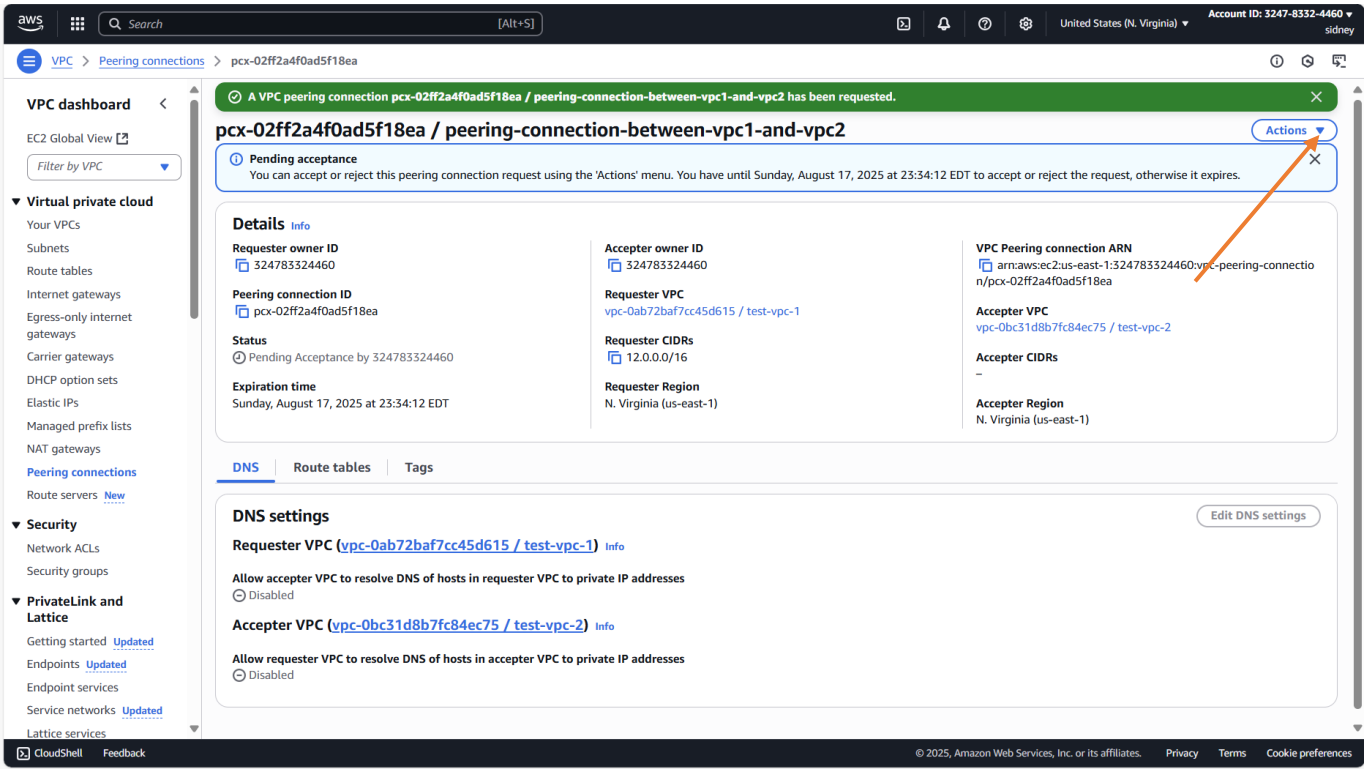
Click on the drop down and select "**test-vpc-1**"



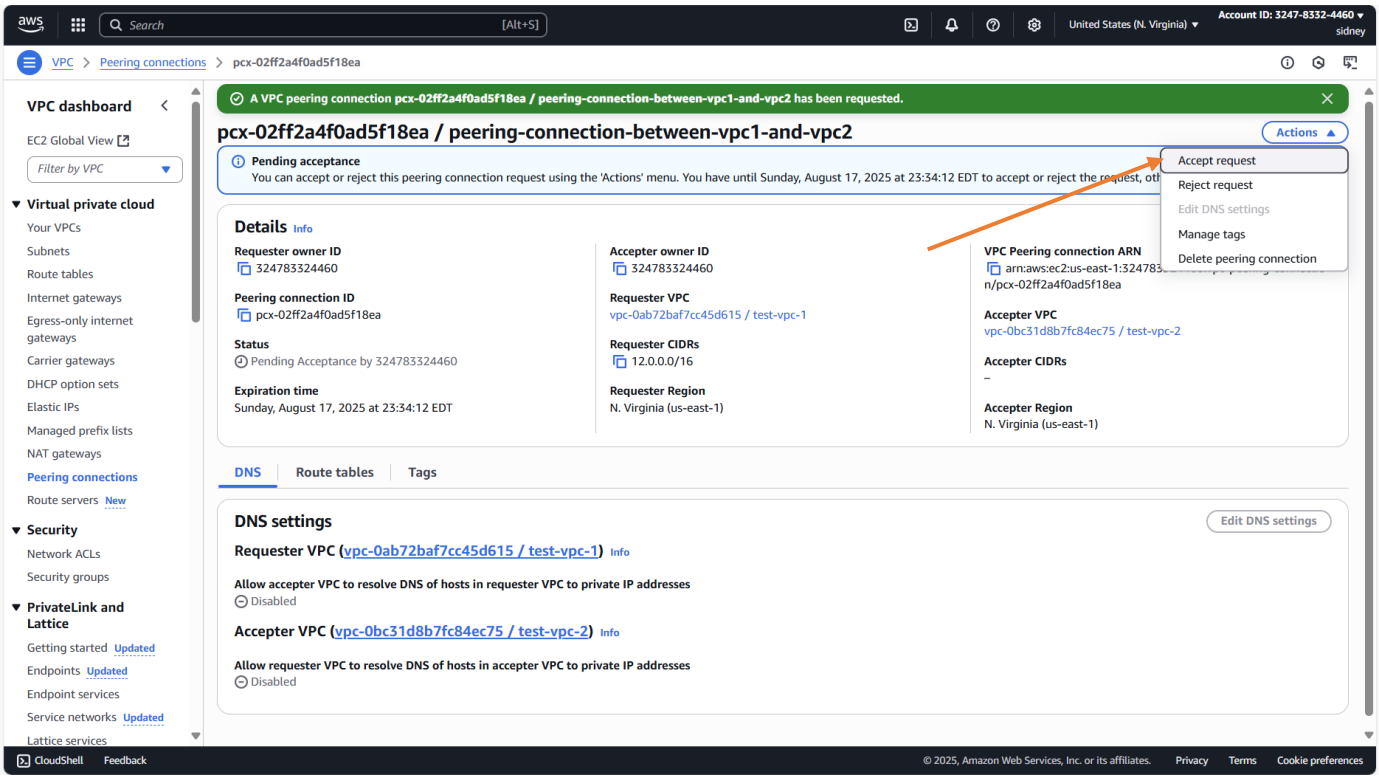
Click on the drop down and select "**test-vpc-2**"



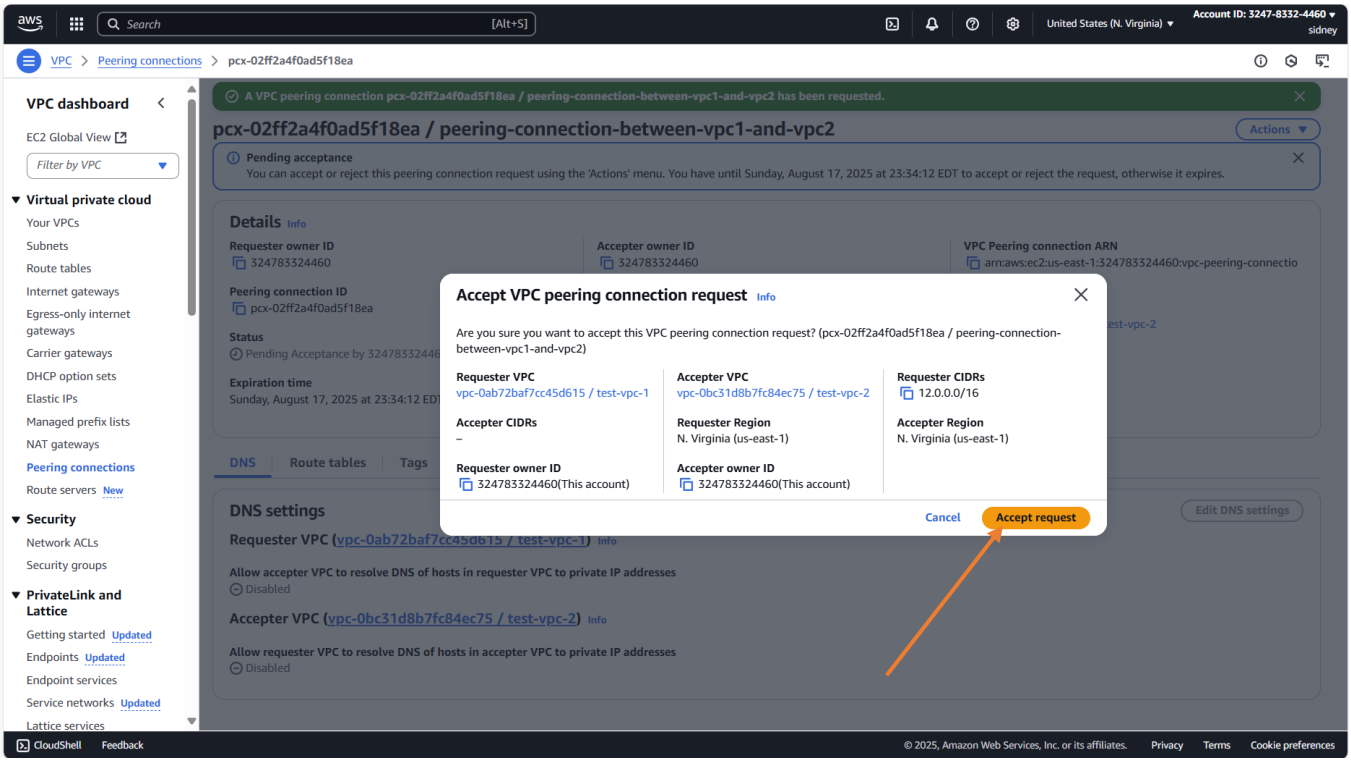
Click on "**create peering connection**"



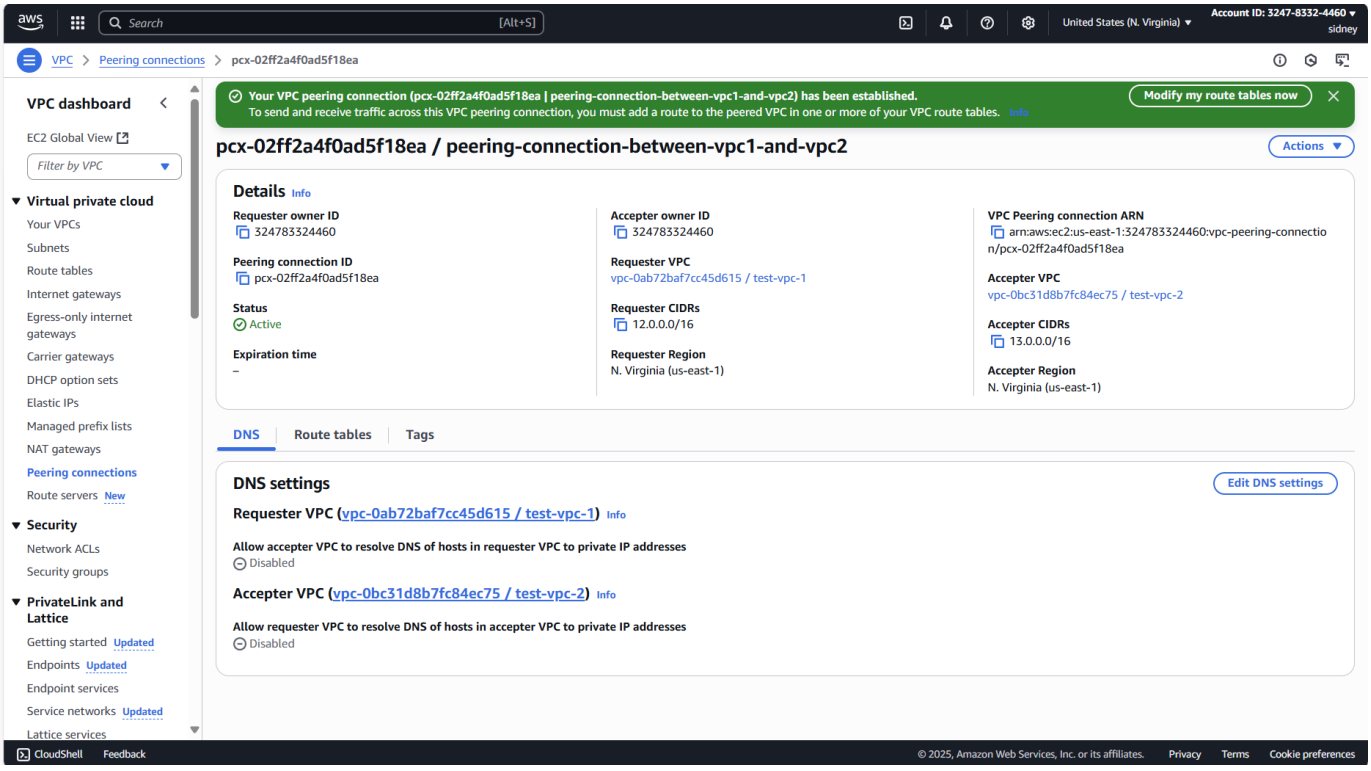
Click on the drop down on “Action”



Select “**Accept Request**”



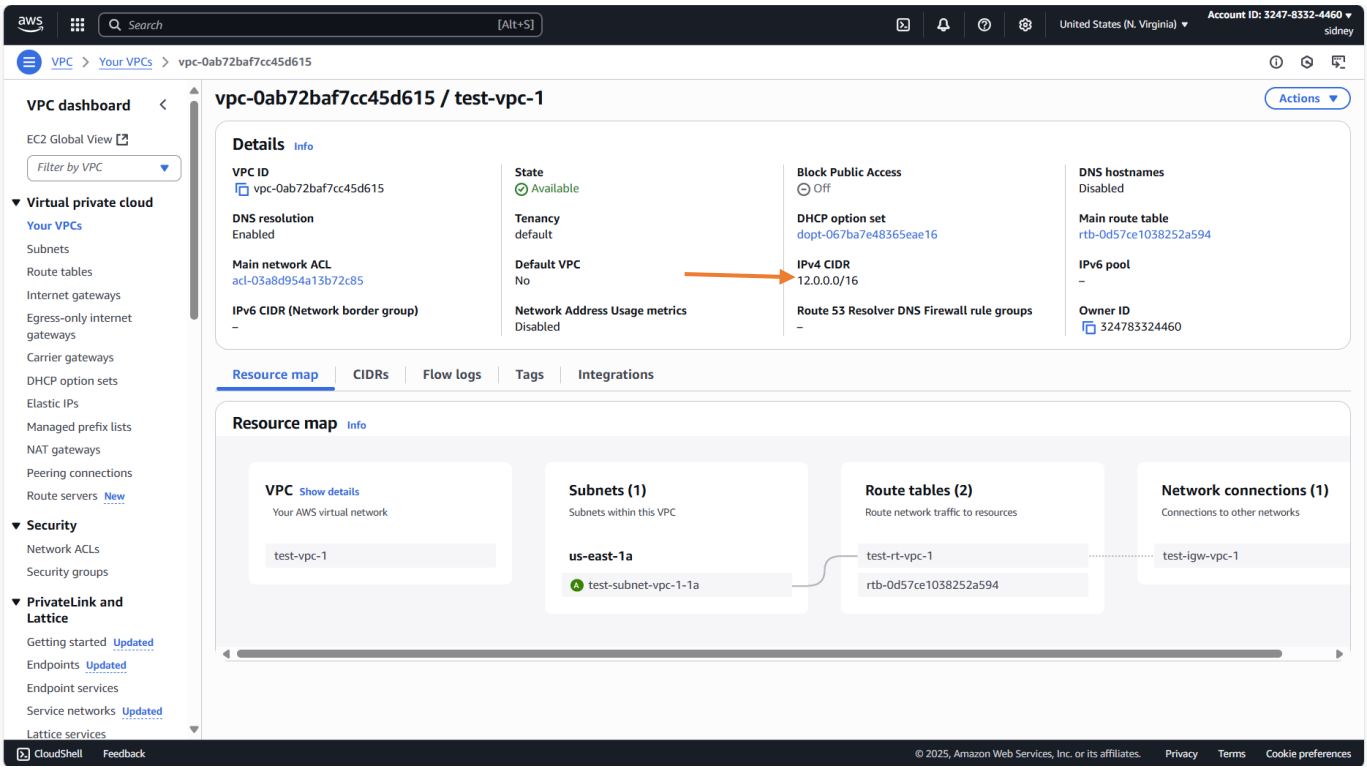
Click on “**Accept Request**”

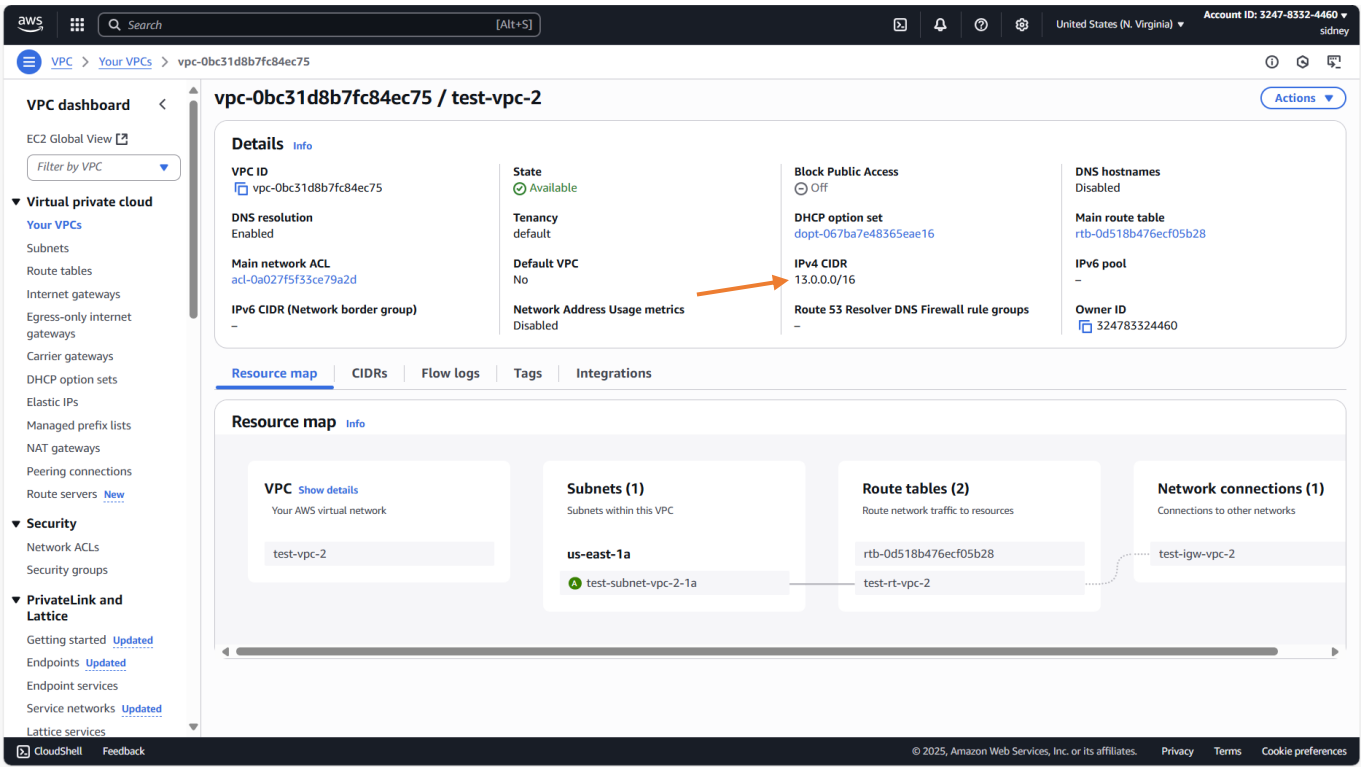


You can see our peering connection has been established

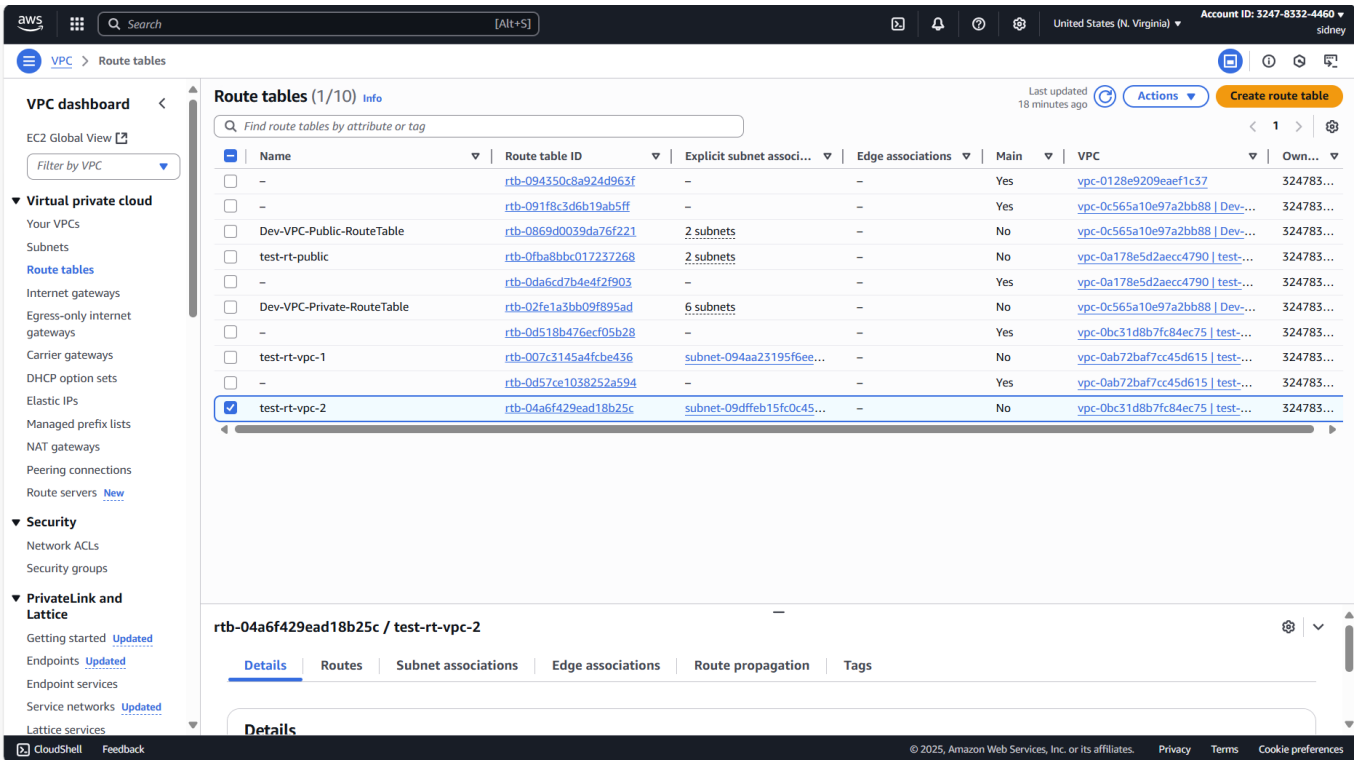
There is one more thing we need to set up before this connection works and we are able to communicate between the two EC2 instances.

The next thing we need to set is the Route Table. First go to our two VPCs and open them on separate tabs.

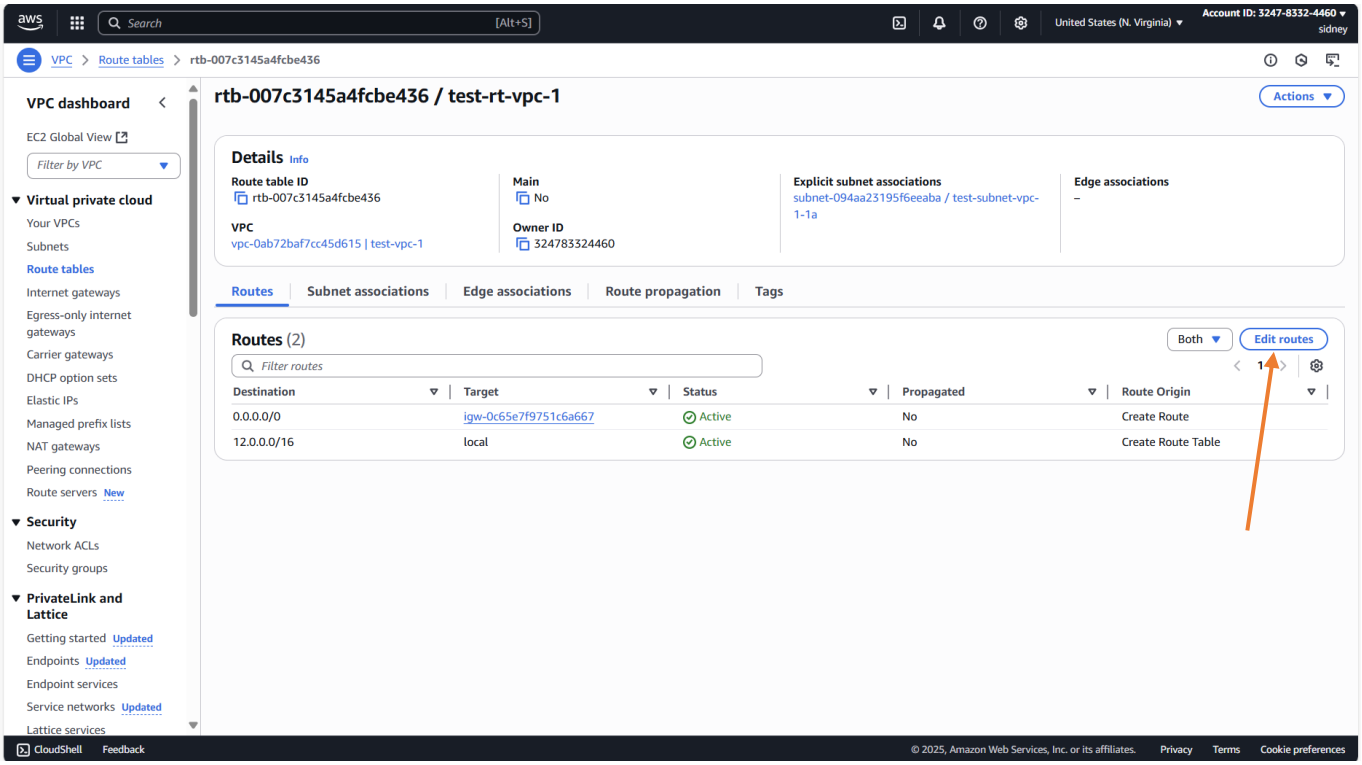




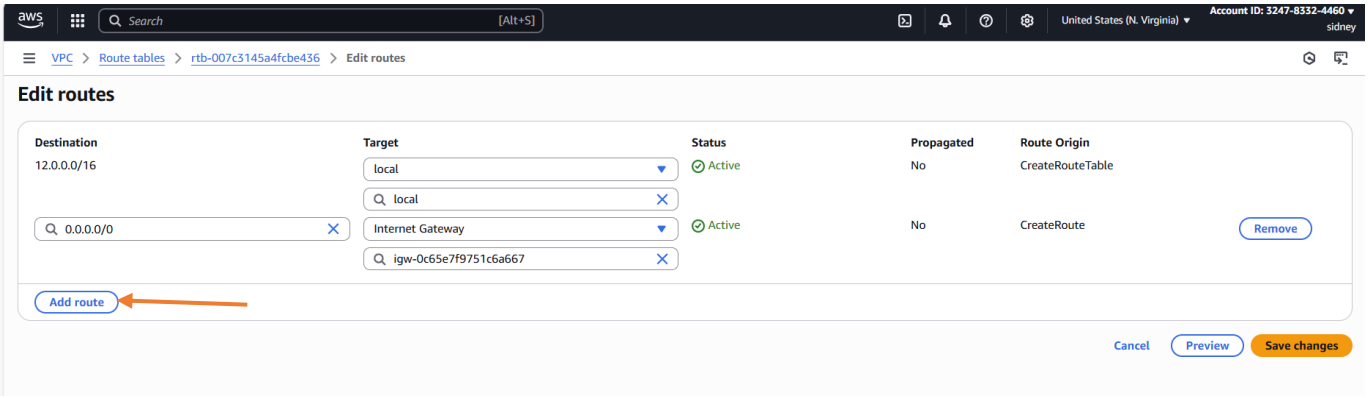
Go to the Route Tables and modify the route tables.



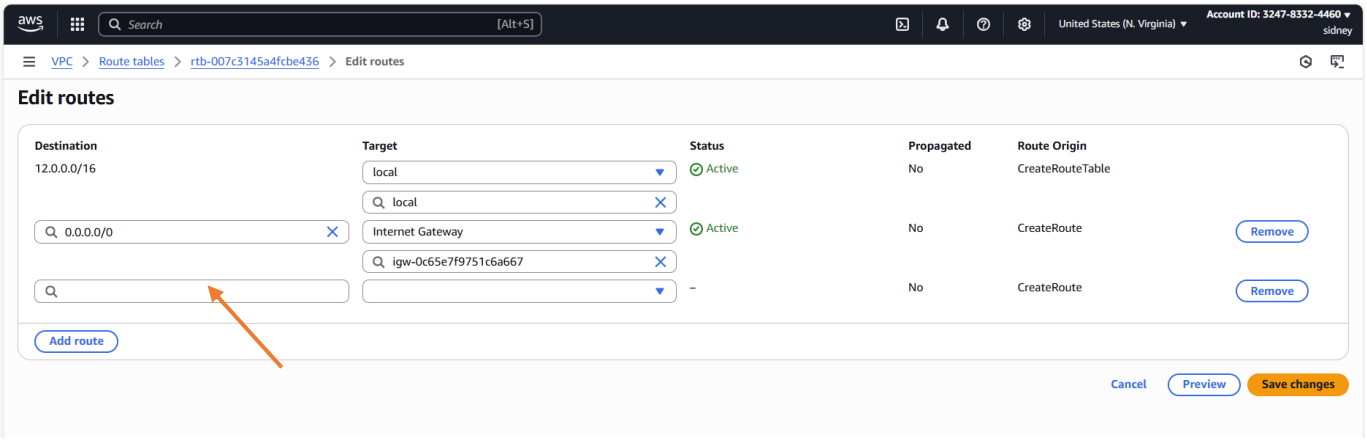
Open the route table of **test-vpc-1**.



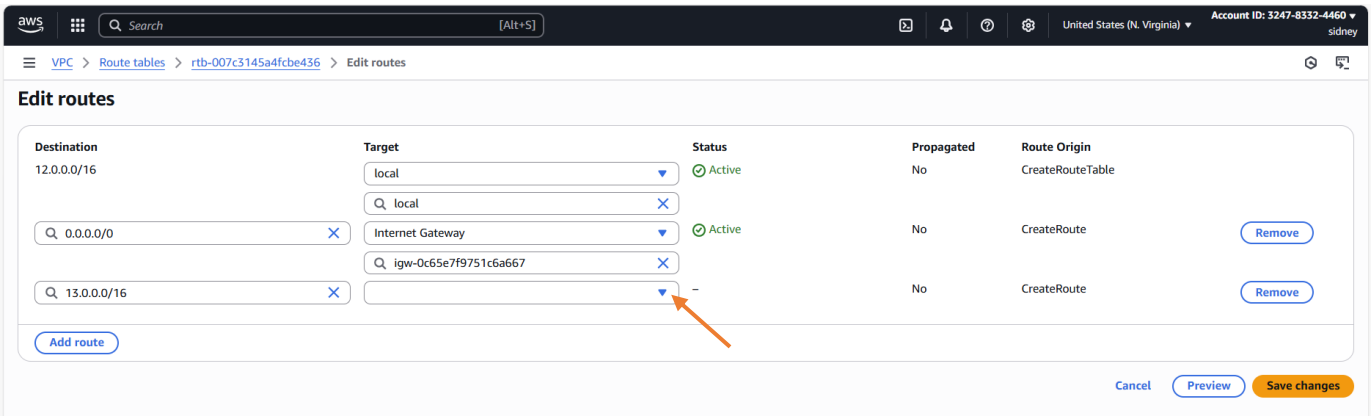
Click on "**Edit Route**"



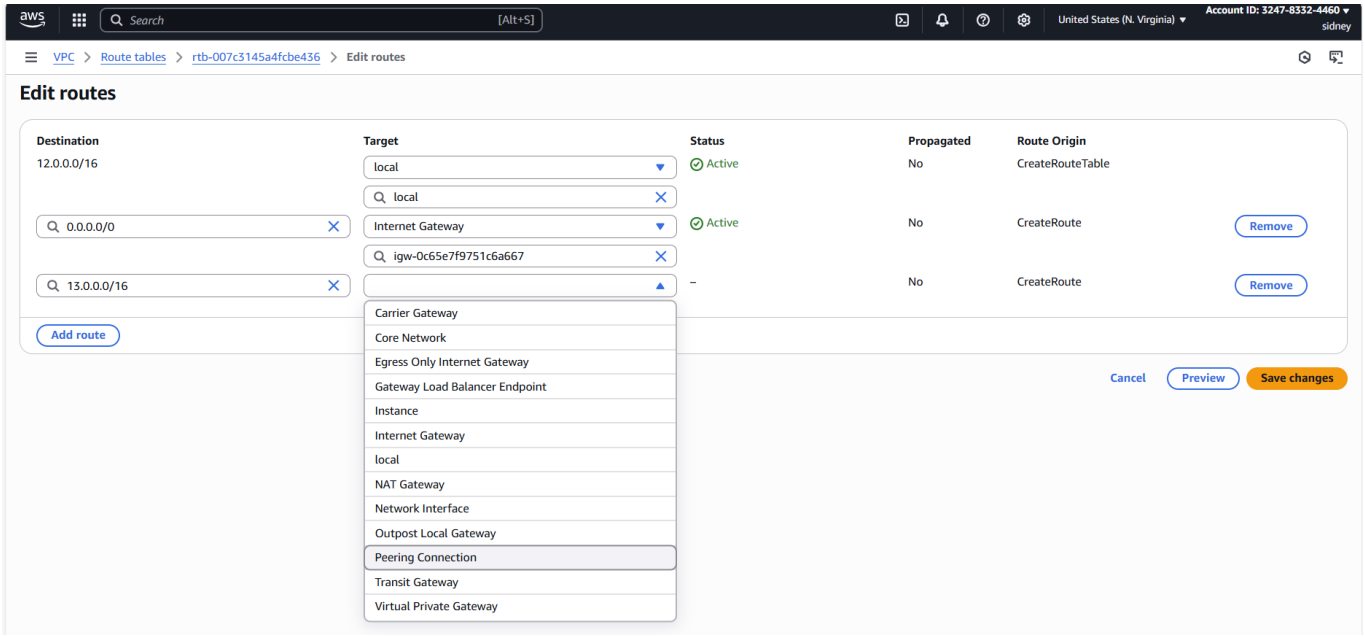
Click on "**Add Route**"



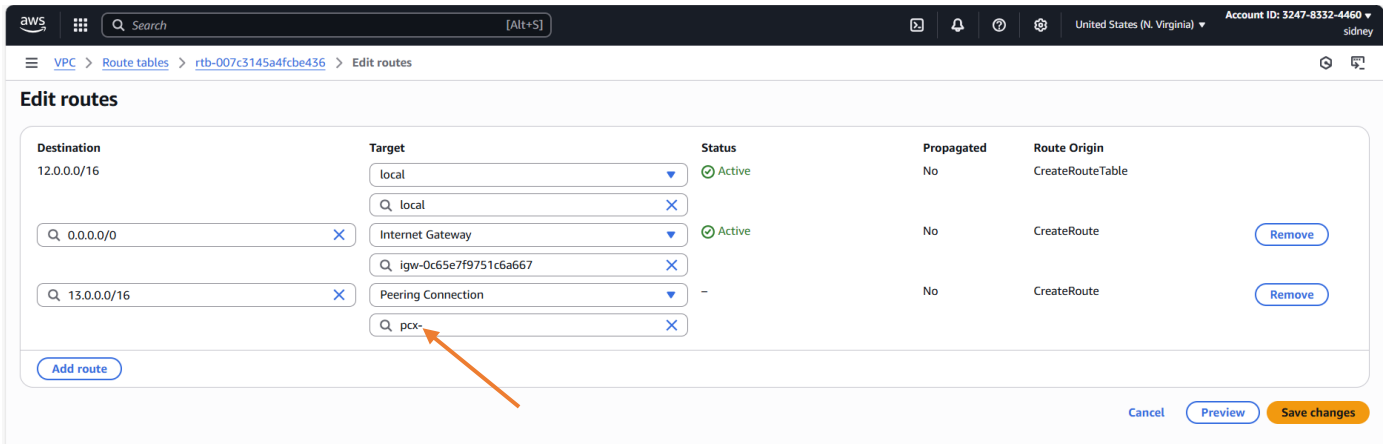
Copy the IPv4 CIDR of test-vpc-2 and paste on the "**destination**"



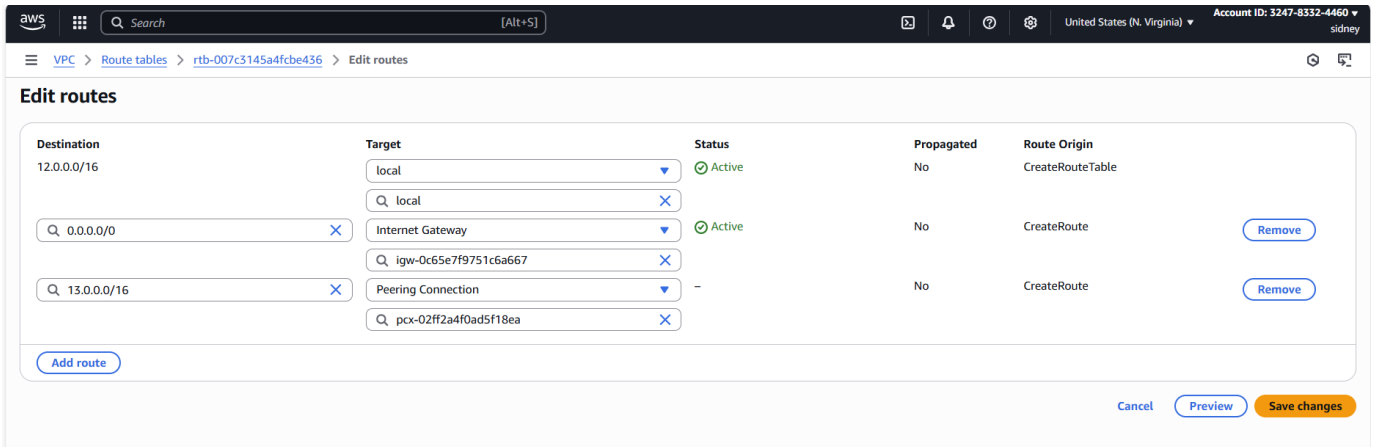
Click on the drop down on "**Target**"



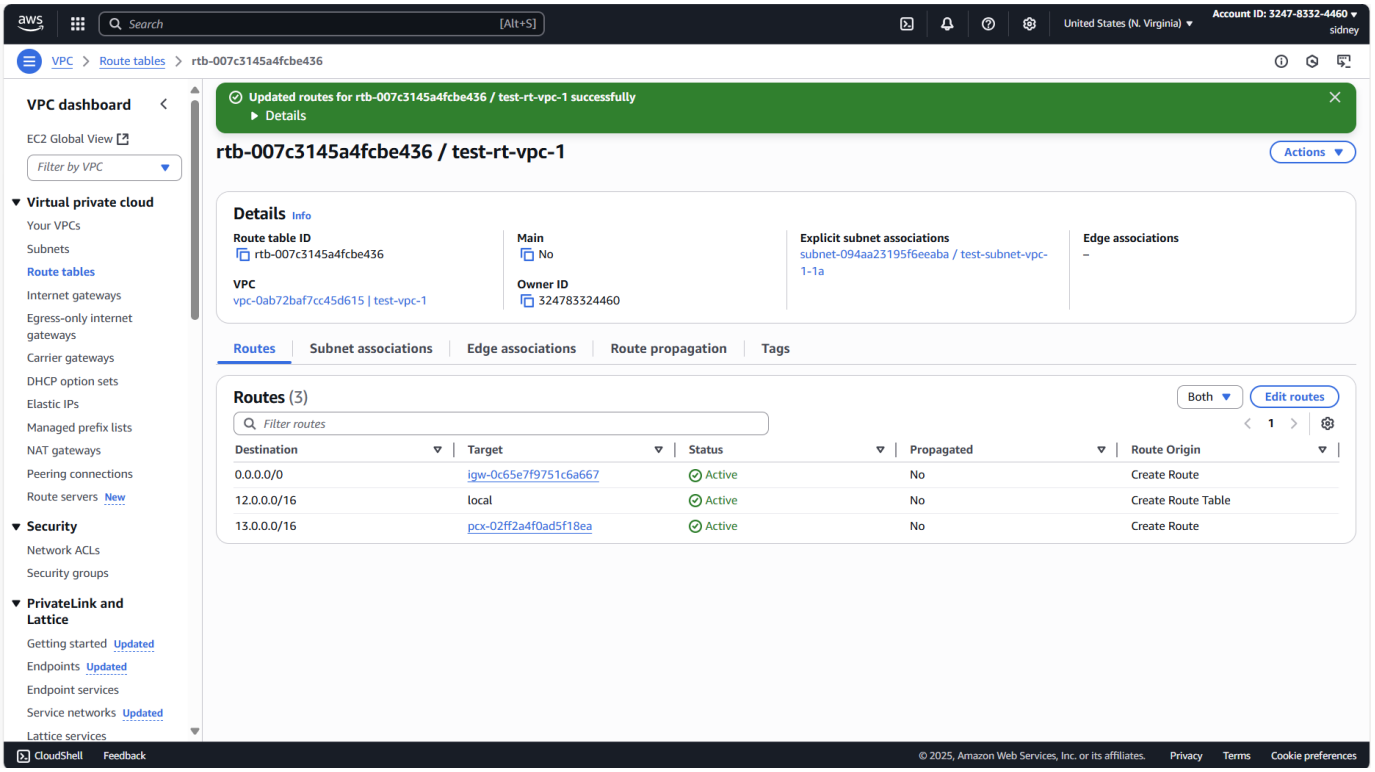
Select “**Peering Connection**”



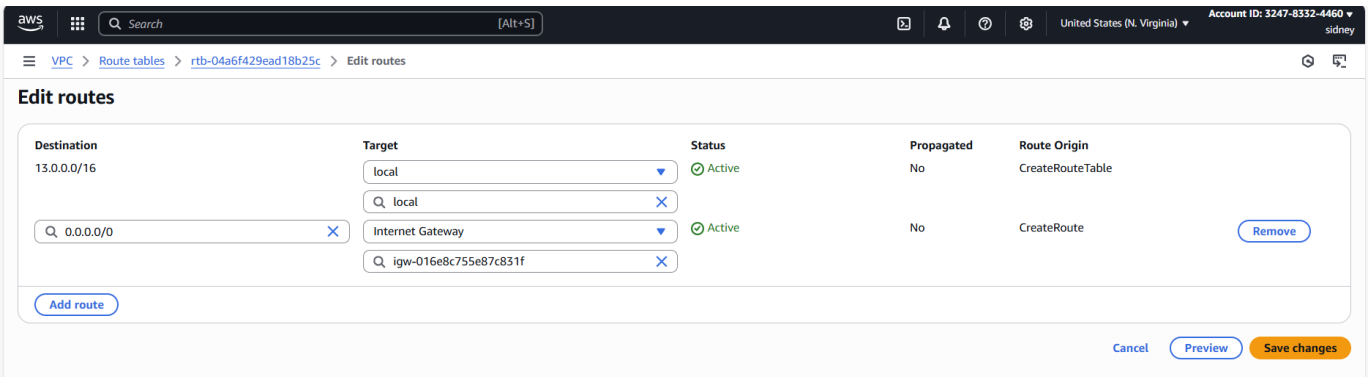
Click on “**pcx-**“ and select our connection



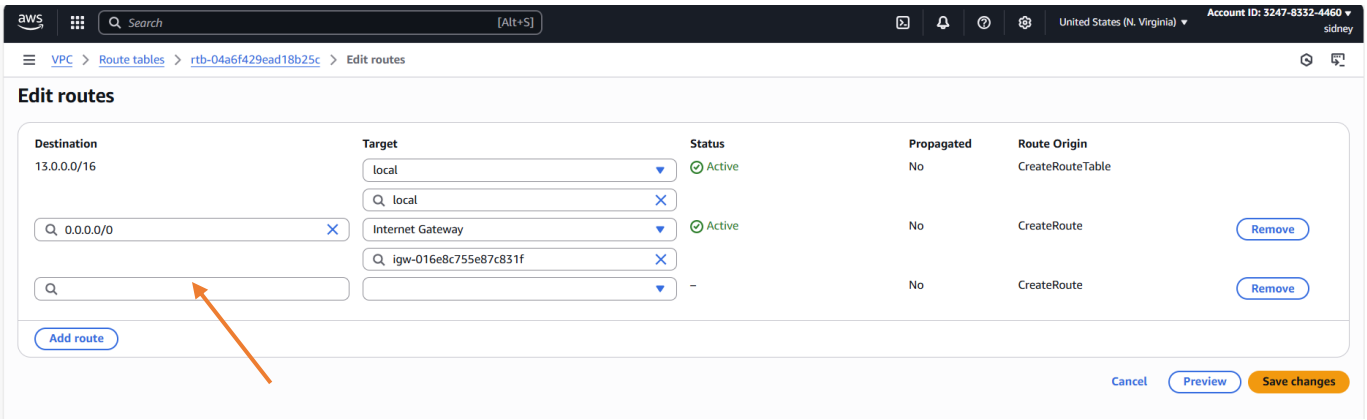
Click on “**Save Changes**”



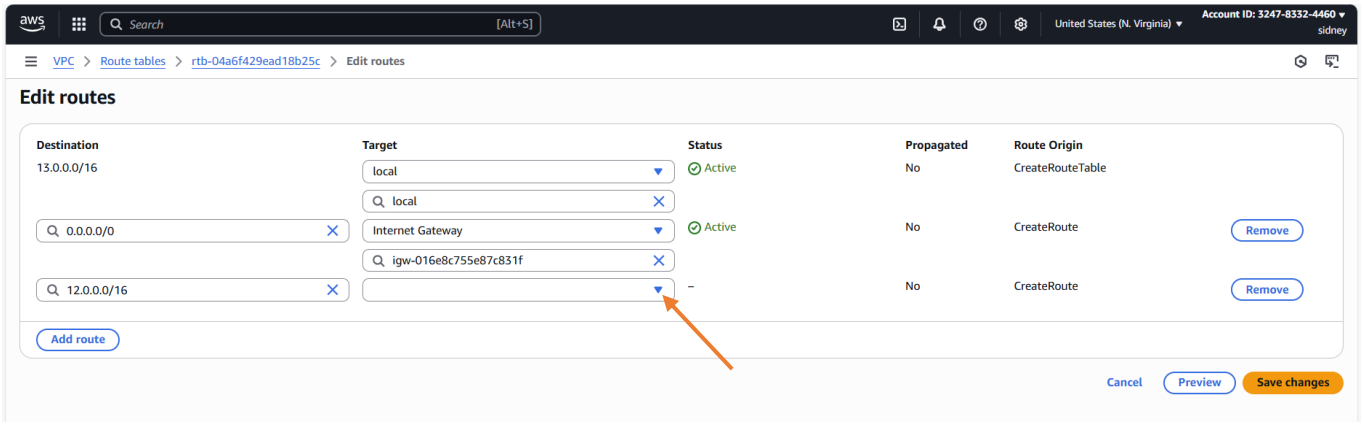
Now we do same for the other VPC. Click on "**Edit Route**" on the Route table of "**test-vpc-2**"



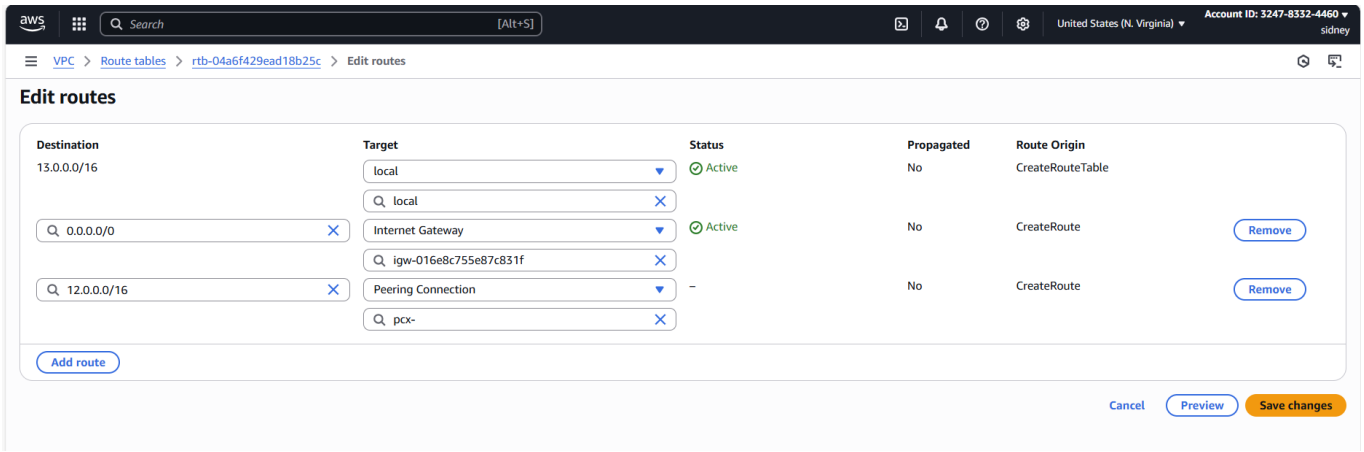
Click on "**Add Route**"



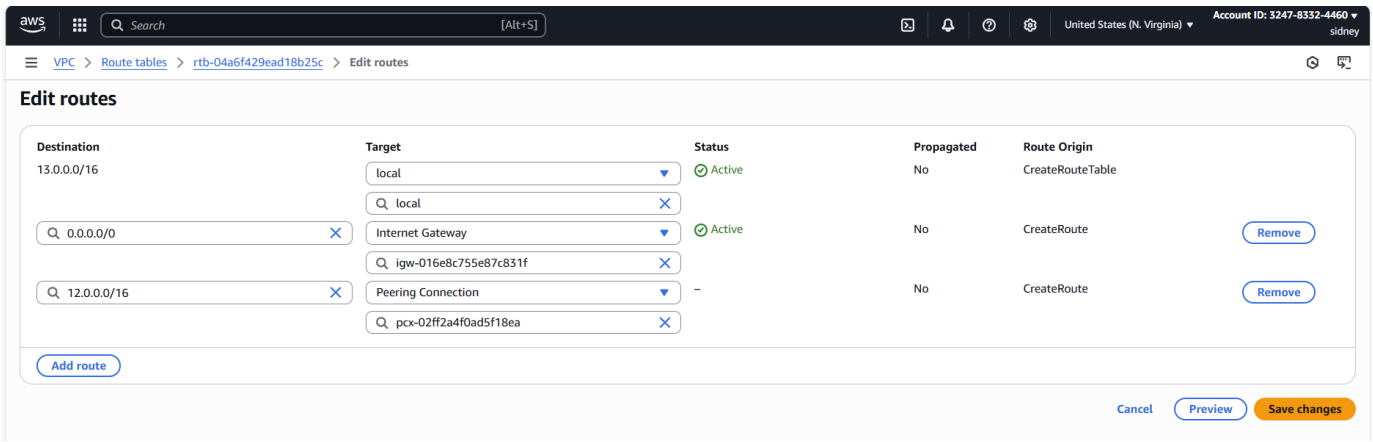
Add the IPv4 CIDR of "**test-vpc-1**" here



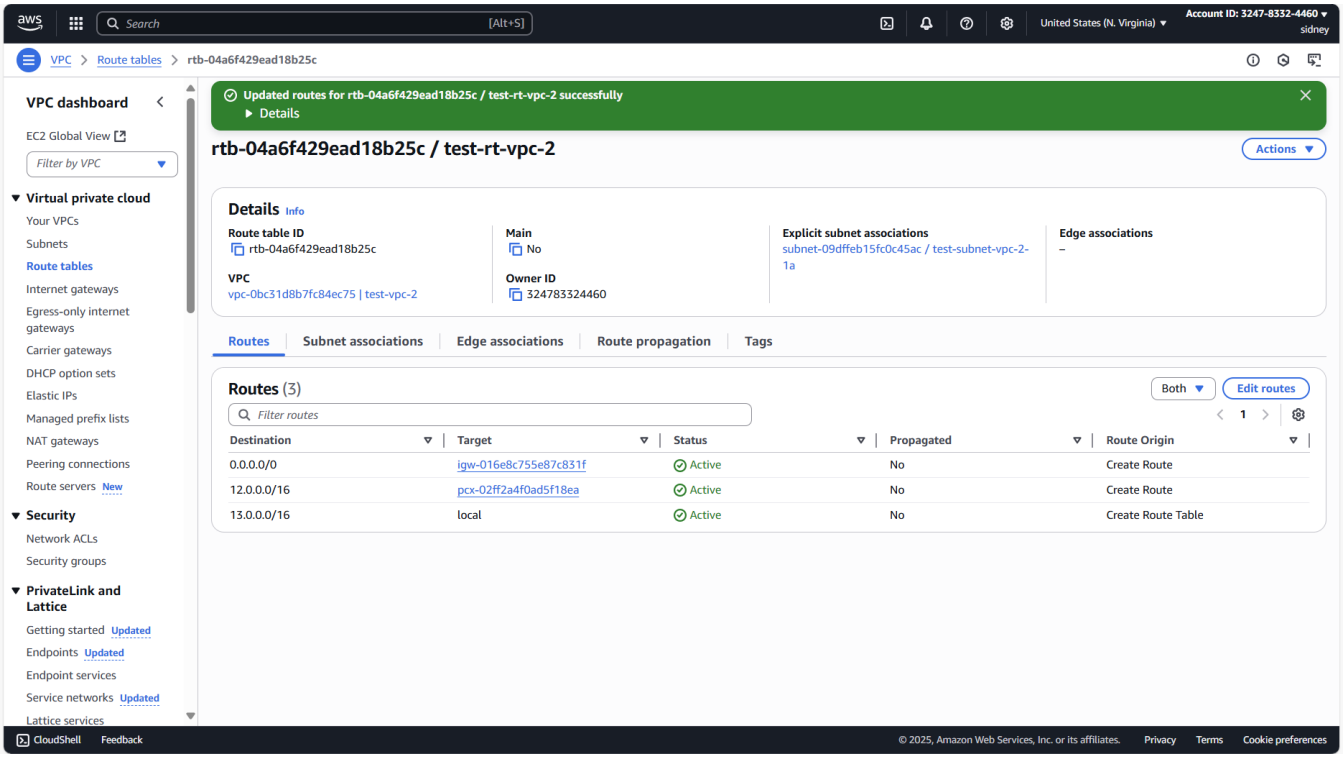
Click on the drop down on "**Target**" and select "**Peering Connection**"



Click on "**pcx-**" and select our peering connection



Click on "**Save Changes**"



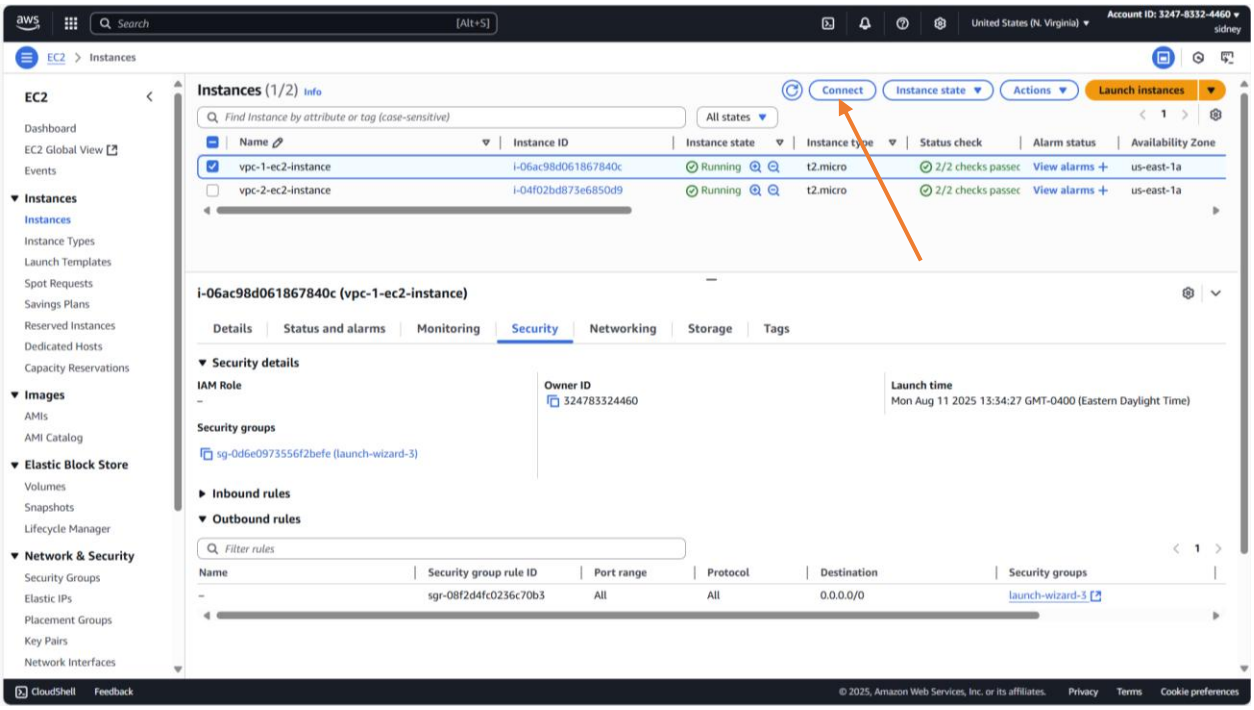
You can see our Route has been modified.

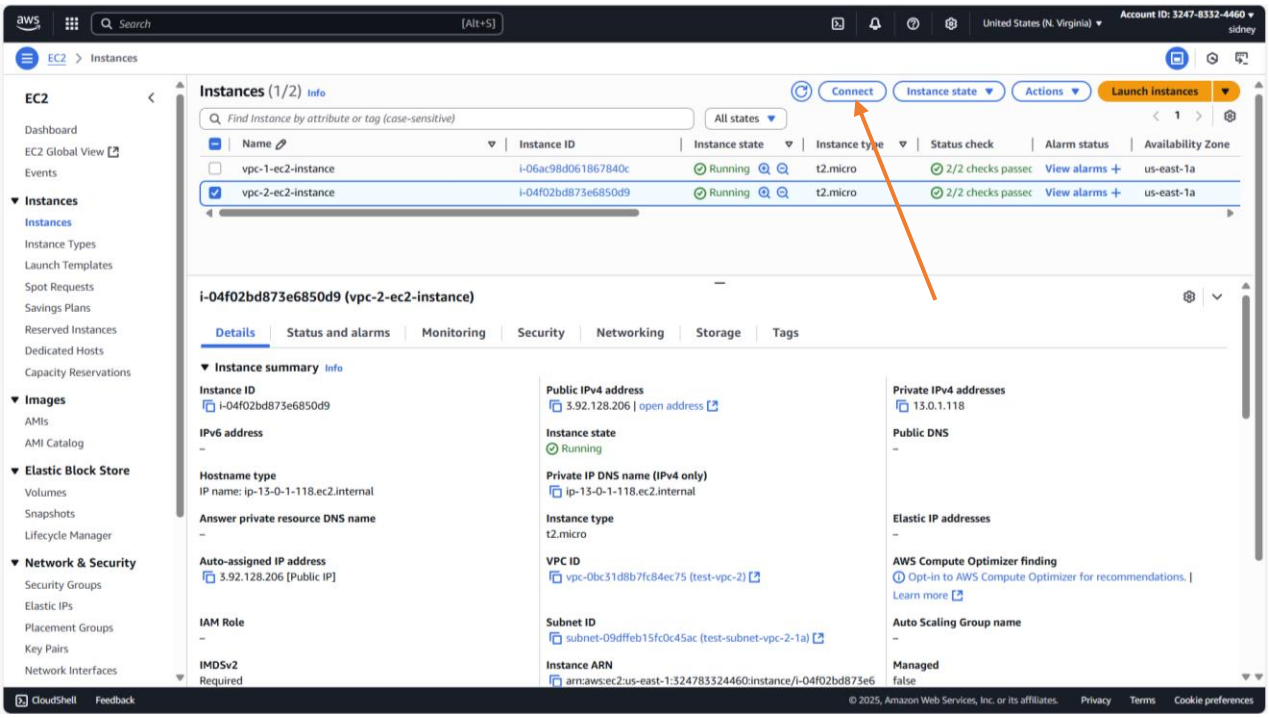
Part 9: Testing the Connection

The next thing we will do is to SSH into the EC2 instance and try to call each other from each EC2 instances. In testing the communication, we will SSH connect to the EC2 instance in test-vpc-1. Then run the command

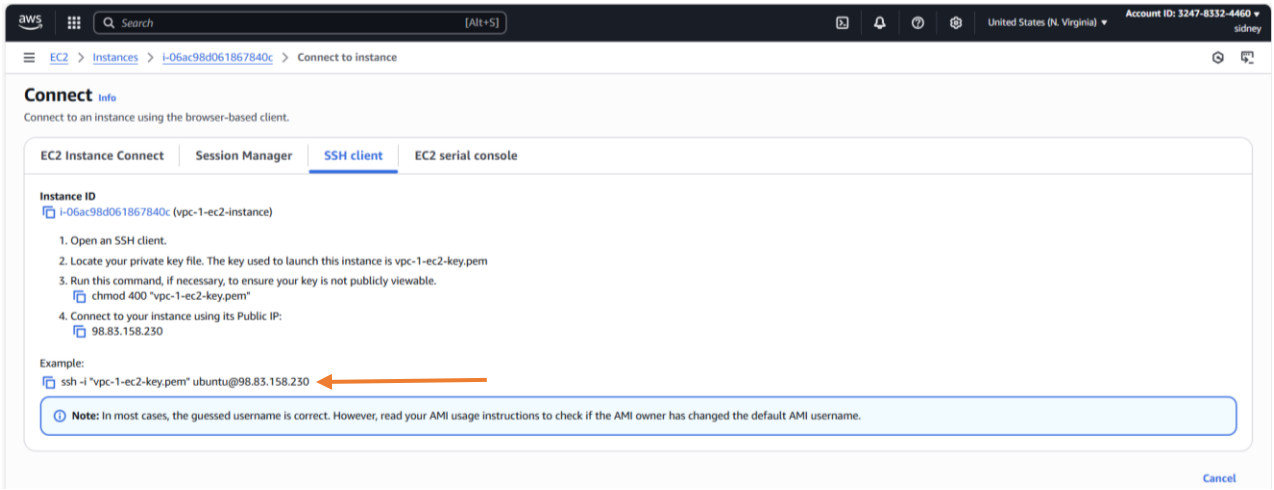
curl <Private IP of EC2 instance in test-vpc-2>. And vice versa.

Go to EC2 dashboard and select the two EC2 instances on separate tabs



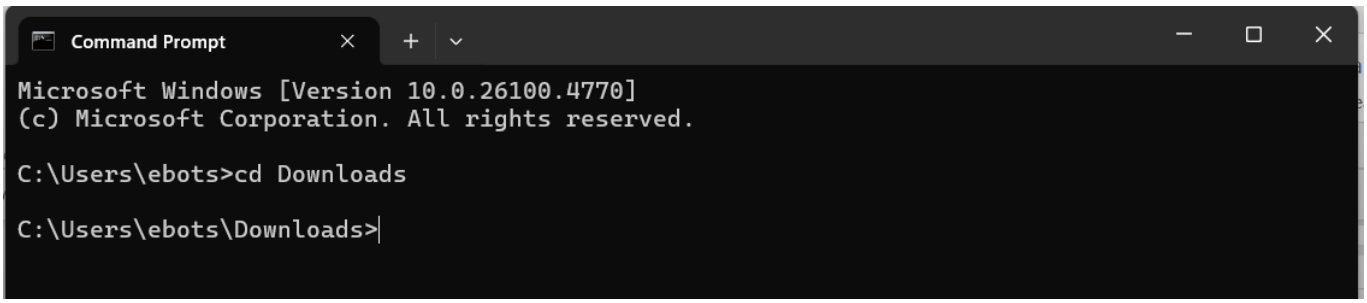


Click on “**Connect**” in the first EC2 instance

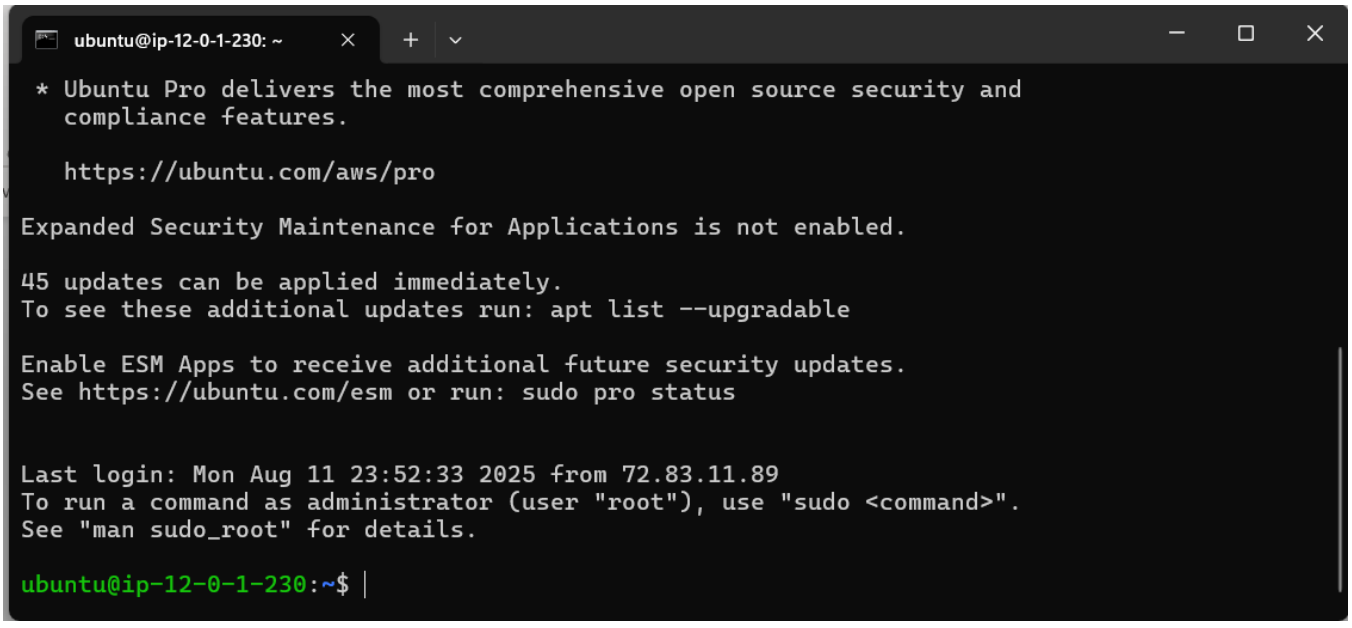


Copy the command: **ssh -i "vpc-1-ec2-key.pem" ubuntu@98.83.158.230**

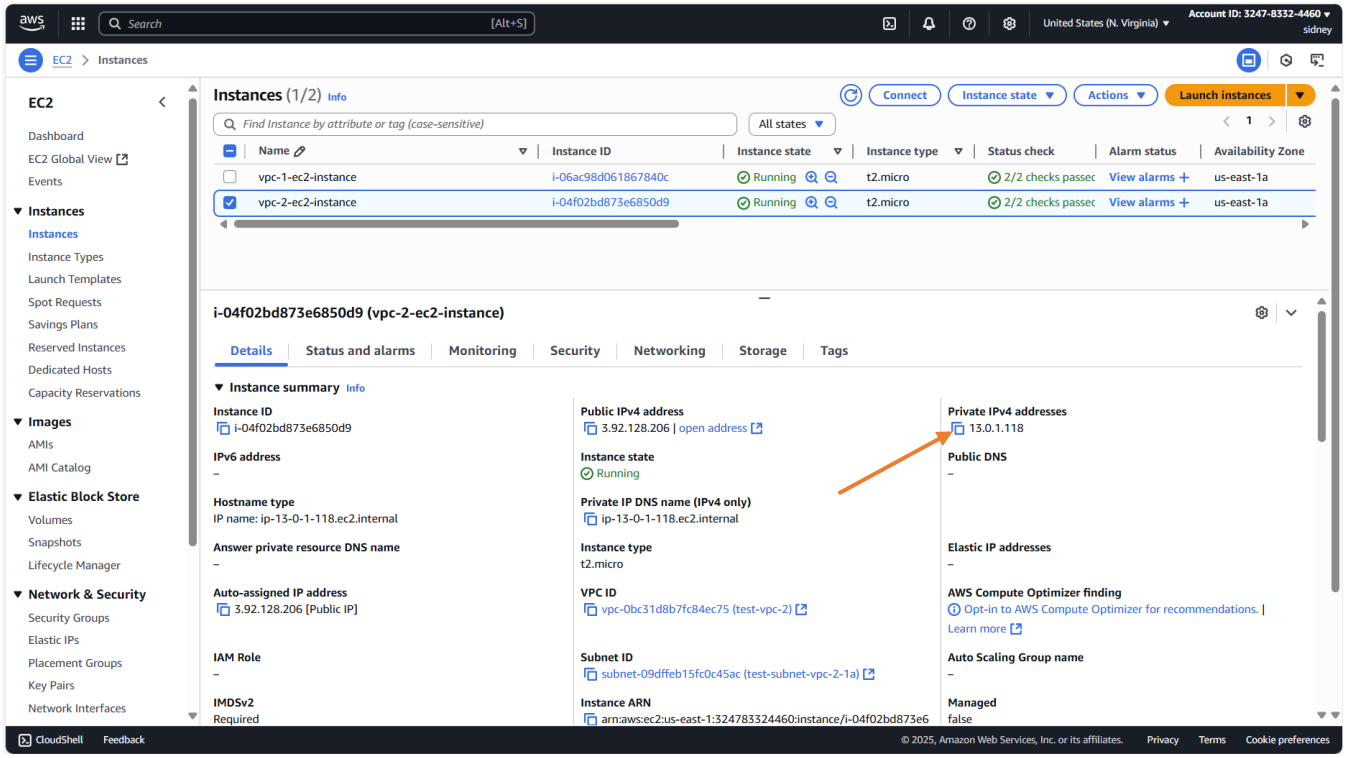
Then open command prompt and navigate to where the key is stored, that is our “**Downloads**” folder



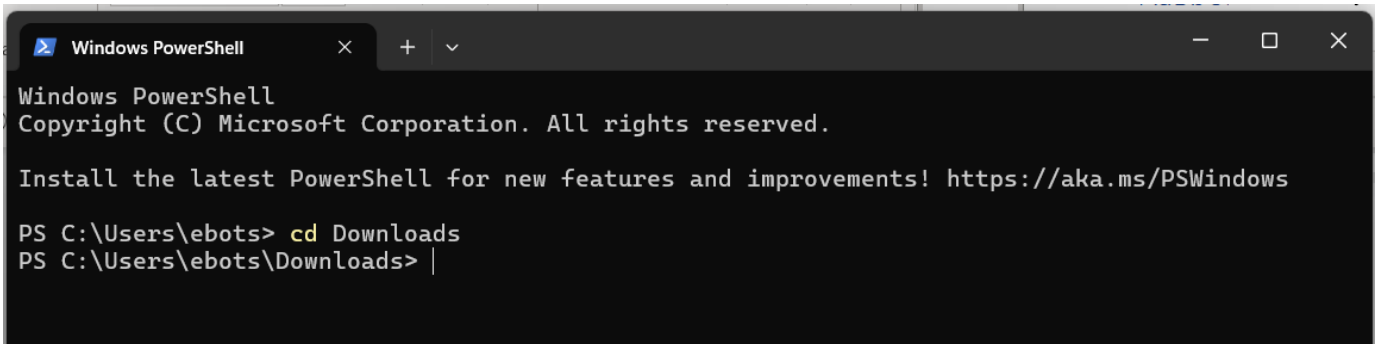
Then run the command: **ssh -i "vpc-1-ec2-key.pem" ubuntu@98.83.158.230**



We are now connected to our EC2 instance in the “**test-vpc-1**”. Now let us copy the “**Private IPv4 address**”

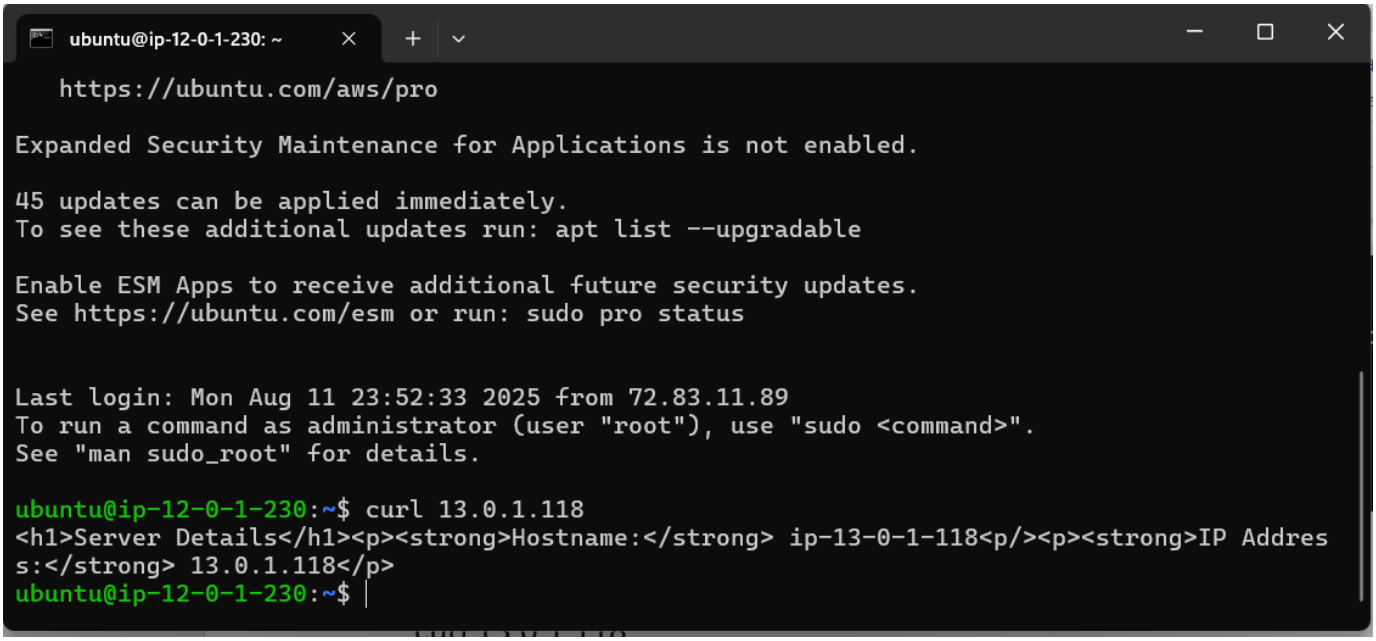


Open Command prompt and navigate to the "**Downloads**" folder where our keys are saved.



You should get a response by running the command:

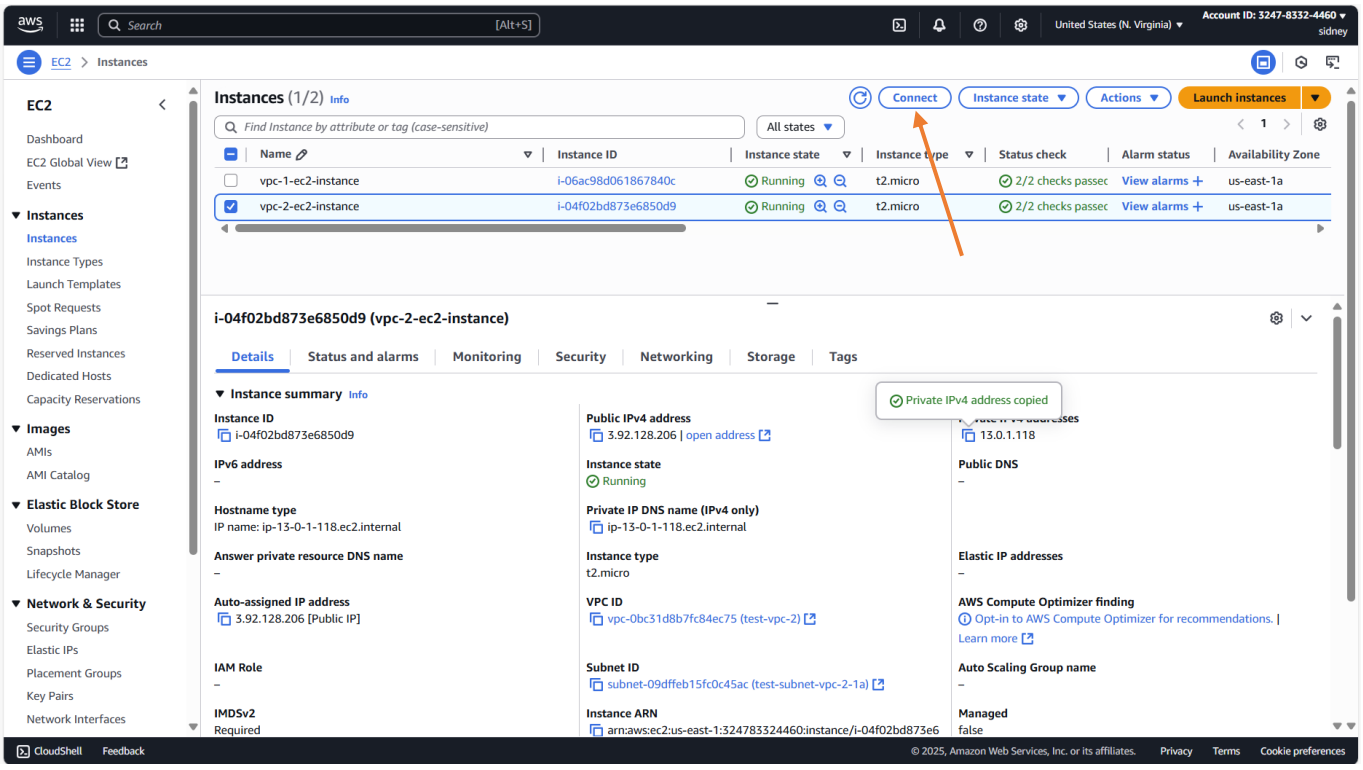
```
curl 13.0.1.118
```



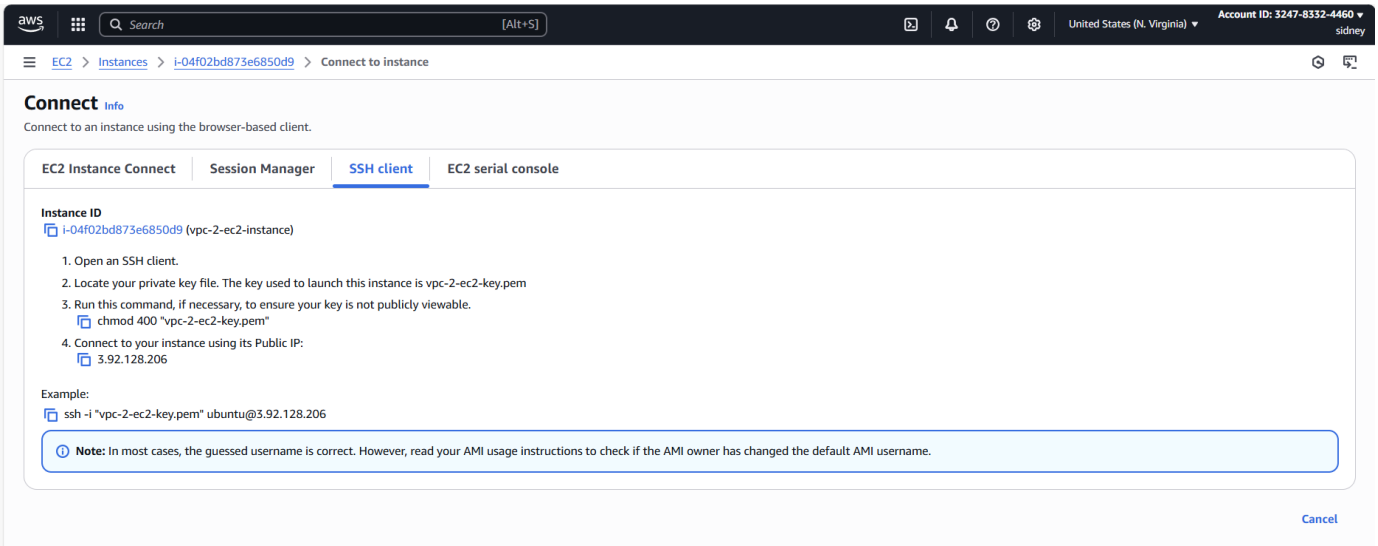
As you can see, we are getting a response from the EC2 instance in "**test-vpc-2**". We are in IP address "**12-0-1-230**" which is my VPC "**test-vpc-1**" IP range and we are accessing the IP address "**13-0-11-118**".

Now, let us try the other way. We will SSH connect to the EC2 instance in "**test-vpc-2**" and run the command:

`curl <Private IPv4 of the EC2 instance in test-vpc-1>`

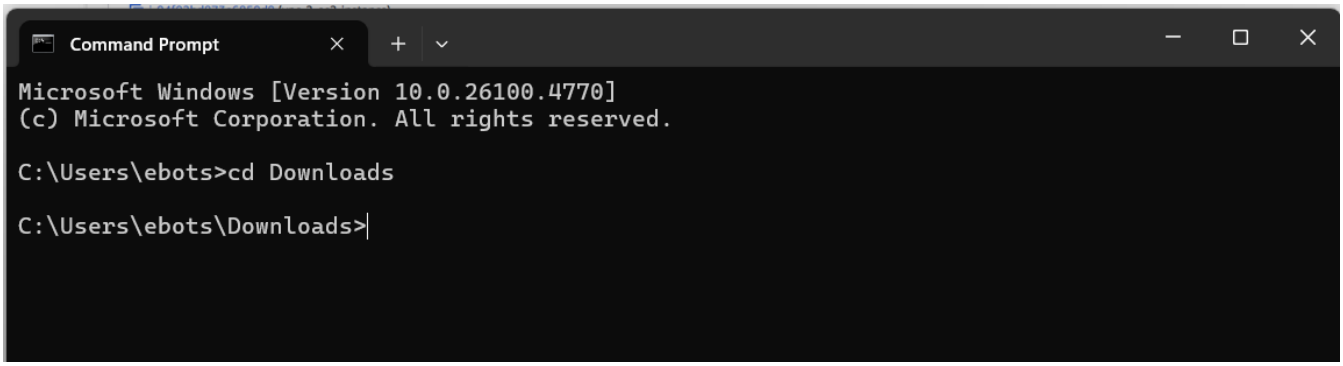


Click on "**Connect**"

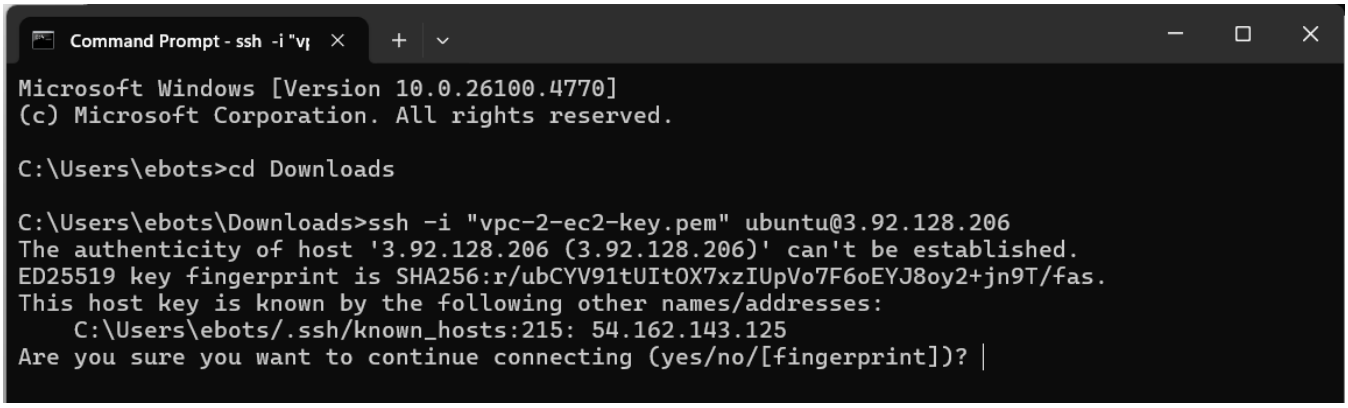


Copy the command: ssh -i "vpc-2-ec2-key.pem" ubuntu@3.92.128.206

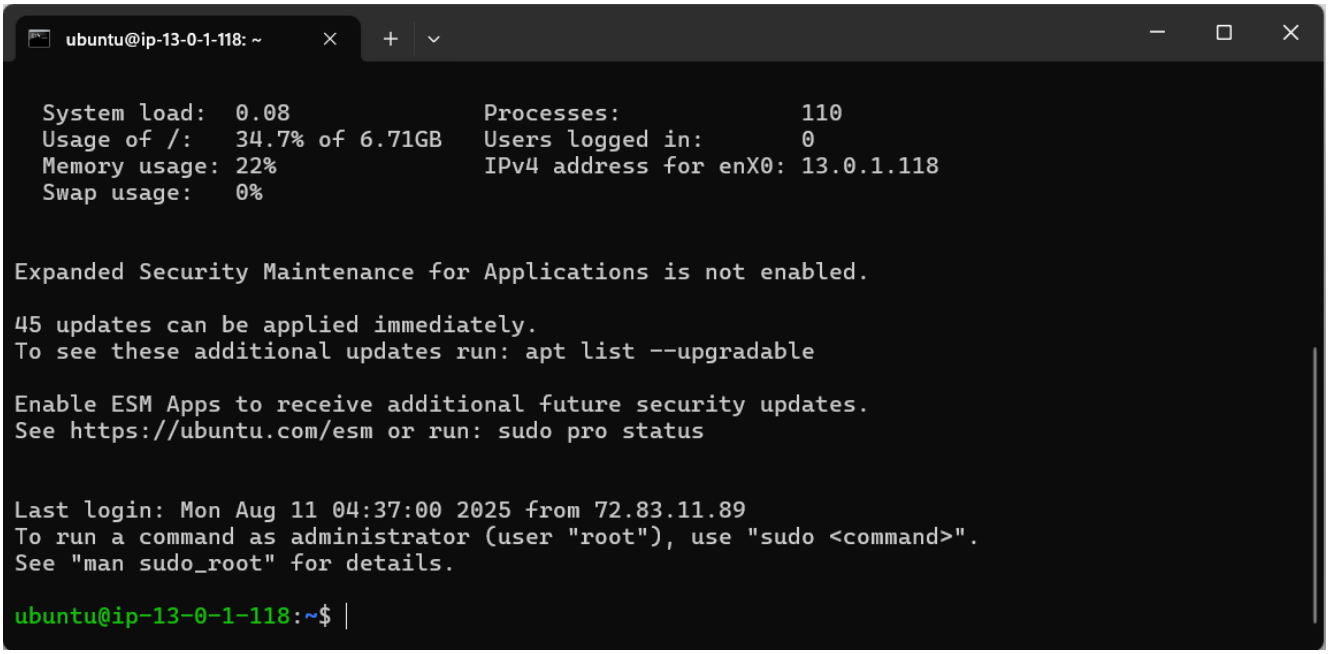
Open command prompt and navigate to the "**Downloads**" folder



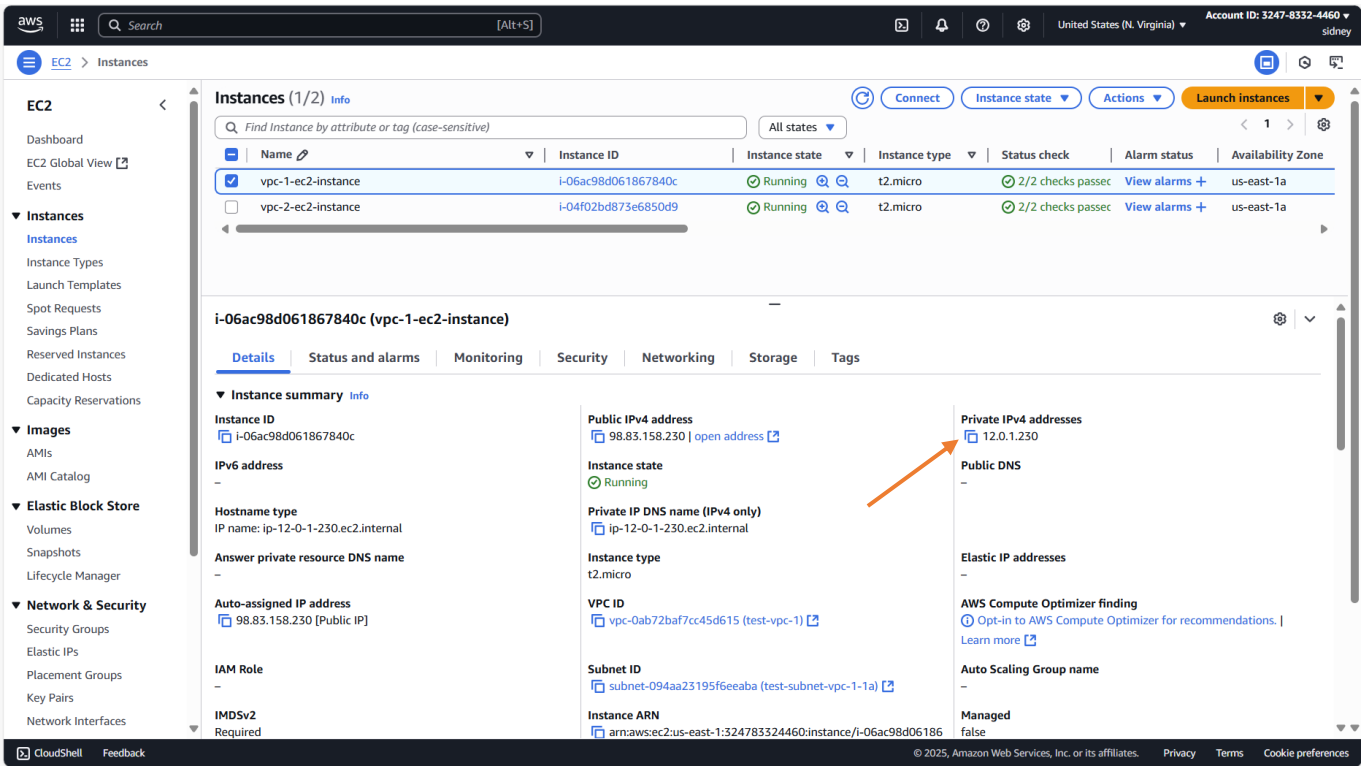
Run the command: **ssh -i "vpc-2-ec2-key.pem" ubuntu@3.92.128.206**



Type "**yes**" and press **ENTER**

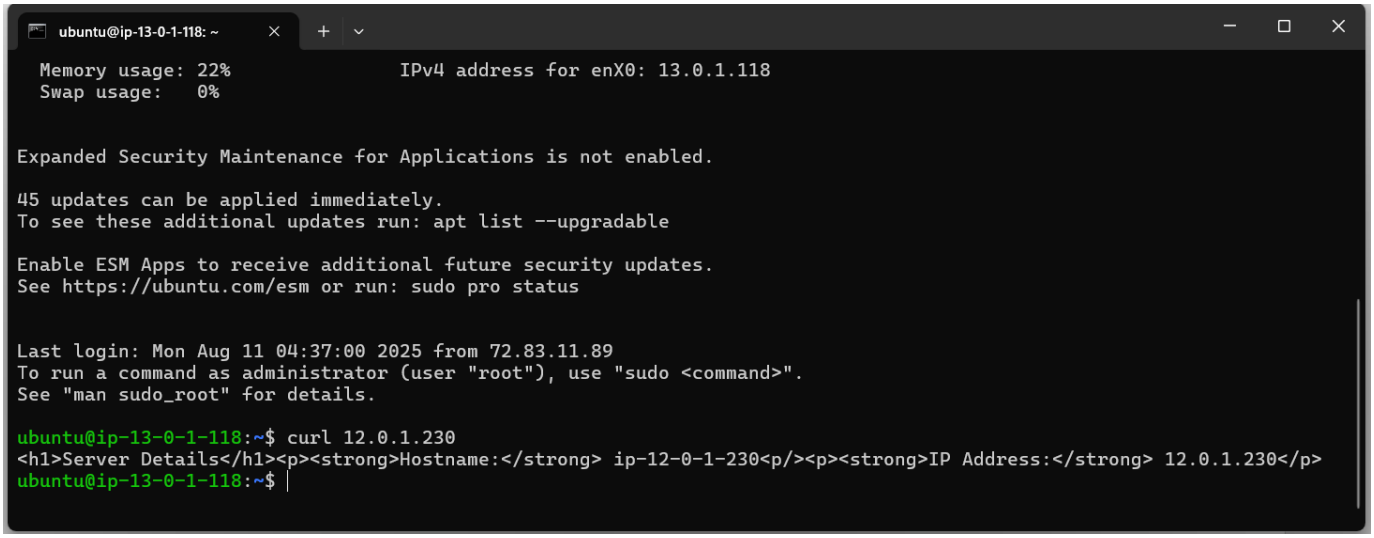


Then copy the "**Private IPv4 address**" of the EC2 instance in "**test-vpc-1**"



Then run the command:

```
curl 12.0.1.230
```



You can see that we are getting a response. You can see I am in IP address "**13-0-1-118**" and I can access the IP address "**12-0-1-230**".

Part 10: Conclusion

We are able to communicate within the VPCs. So, we are able to access the EC2 instance running in another VPC.