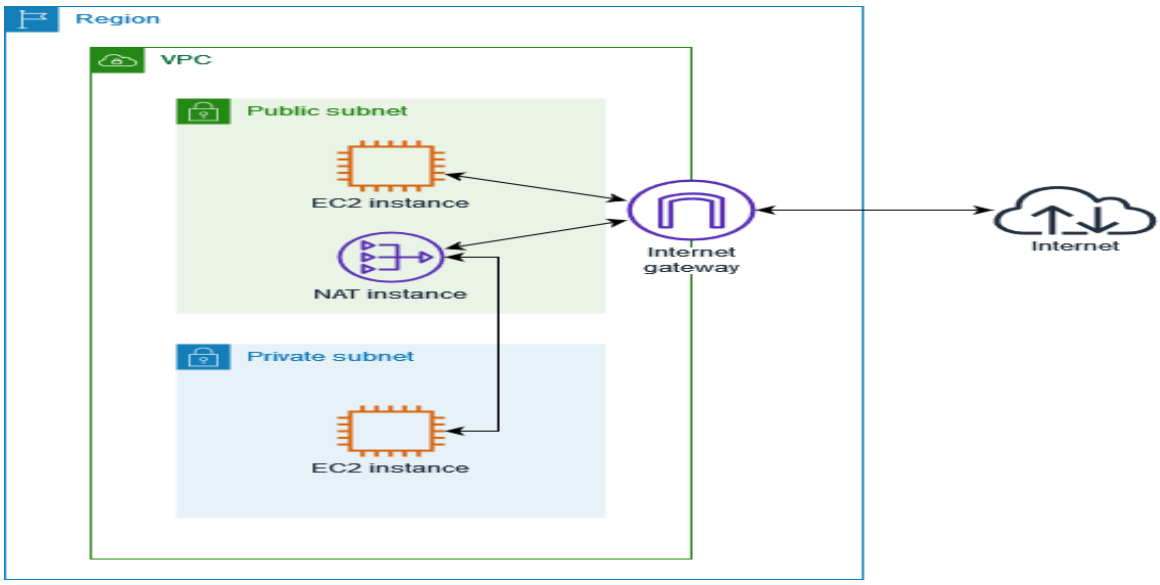
# WEEK :10
# AIM: Create and Configure Amazon Virtual Private Cloud (VPC).




**TASK**

Create your own VPC

Create Public subnet

Create Private subnet

Create Internet Gateway

Attach Internet Gateway to your VPC

Create Public Routing Table, associate subnet and add routing rules

Create Private Routing table, associate subnet and add routing rules

Launch an instance in Public network

Launch an instance in Private network

Create Nat Gateway

Connect to public instance and check internet connectivity

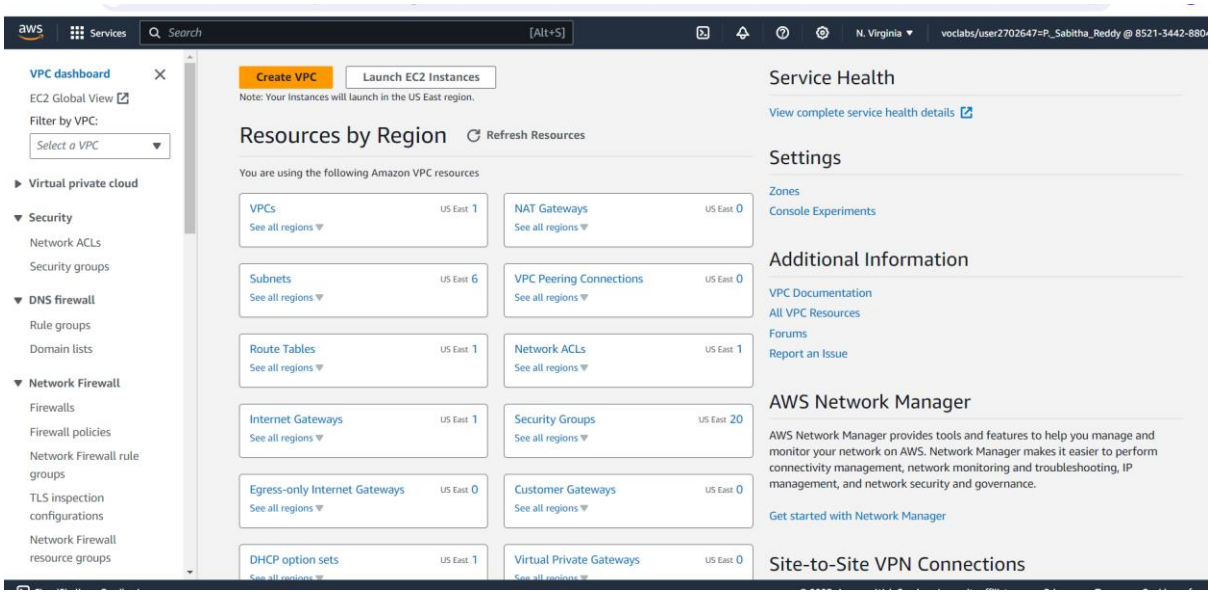
Connect to private instance and check internet connectivity

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

**Step1:** Open AWS console,

Search for VPC in Search Bar,

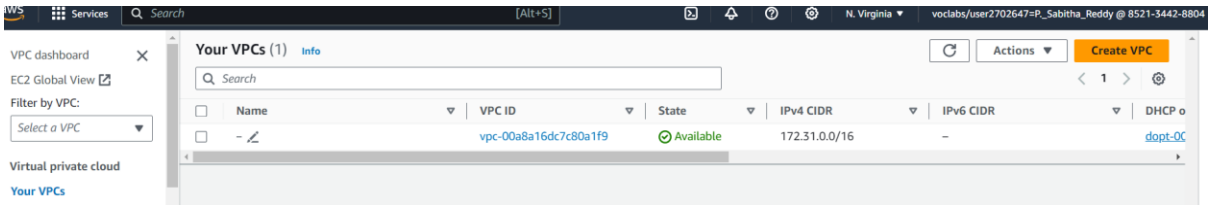
Click on **VPC**




On **VPC Dashboard** Panel,

Click on **YOUR VPC,**

Click on **CREATE VPC** Button




On Create VPC page,

For Name Tag→lab_vpc,

For IPv4CIDR Block→10.0.0.0/16

Leave remaining fields as default,

Click on CREATE VPC Button.

aws | Services | Search [Alt+S]

## VPC settings

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

- (●) VPC only
- ( ) VPC and more

Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

lab_vpc

IPv4 CIDR block Info

- (●) IPv4 CIDR manual input
- ( ) IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info

- (●) No IPv6 CIDR block
- ( ) IPAM-allocated IPv6 CIDR block
- ( ) Amazon-provided IPv6 CIDR block
- ( ) IPv6 CIDR owned by me

Tenancy Info

Default

---

IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info

- (●) No IPv6 CIDR block
- ( ) IPAM-allocated IPv6 CIDR block
- ( ) Amazon-provided IPv6 CIDR block
- ( ) IPv6 CIDR owned by me

Tenancy Info

Default

## Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

| Key | Value - *optional* | |
|---|---|---|
| Name | lab_vpc | Remove tag |

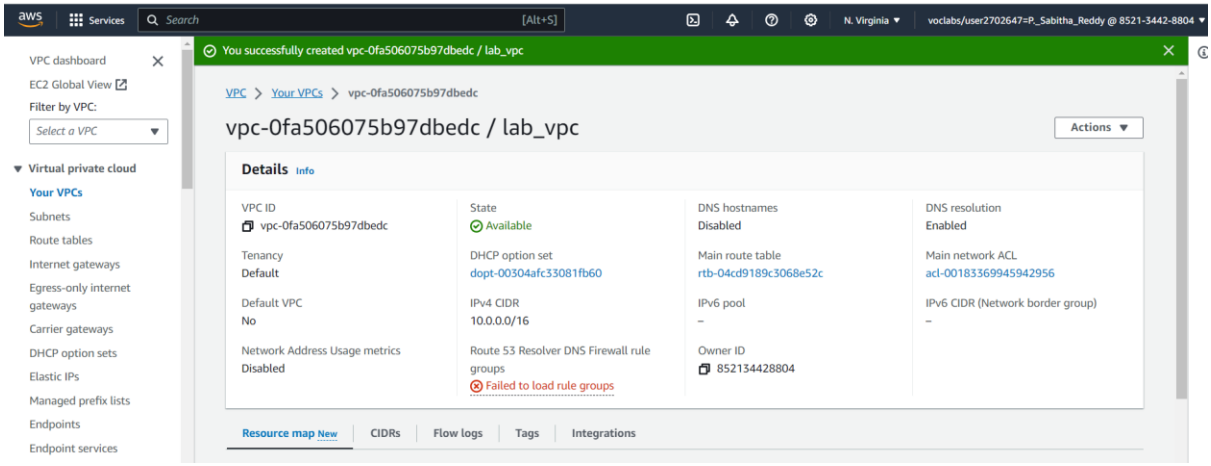
Add tag

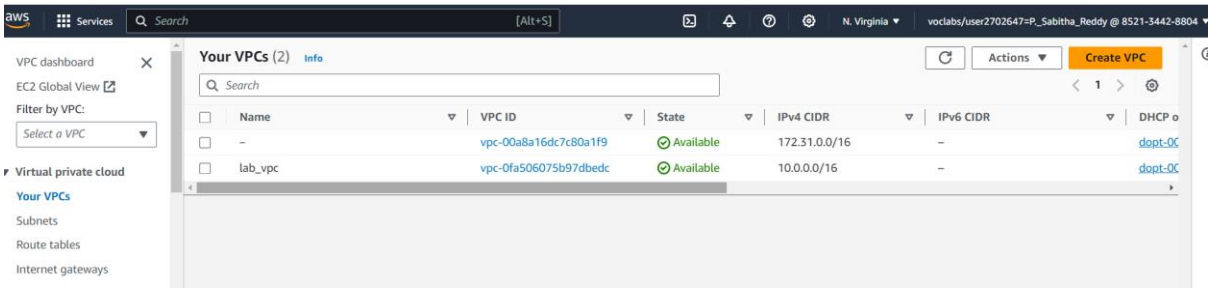
You can add 49 more tags

Cancel | Create VPC

## Verify

Lab_vpc is Created.



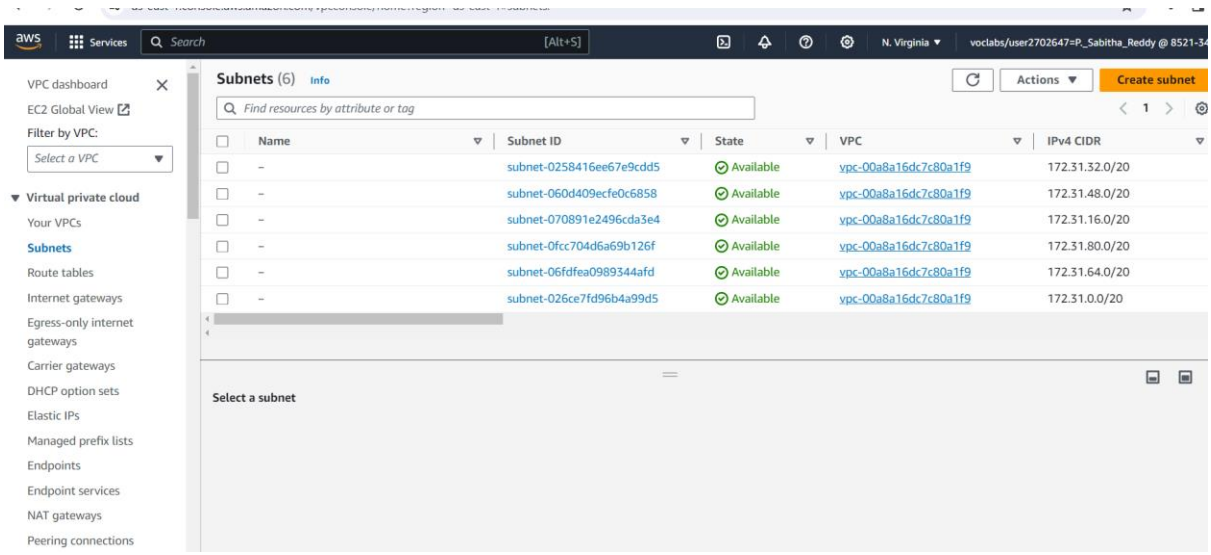




## Step2:

To Create Public Subnet

Click on Subnet

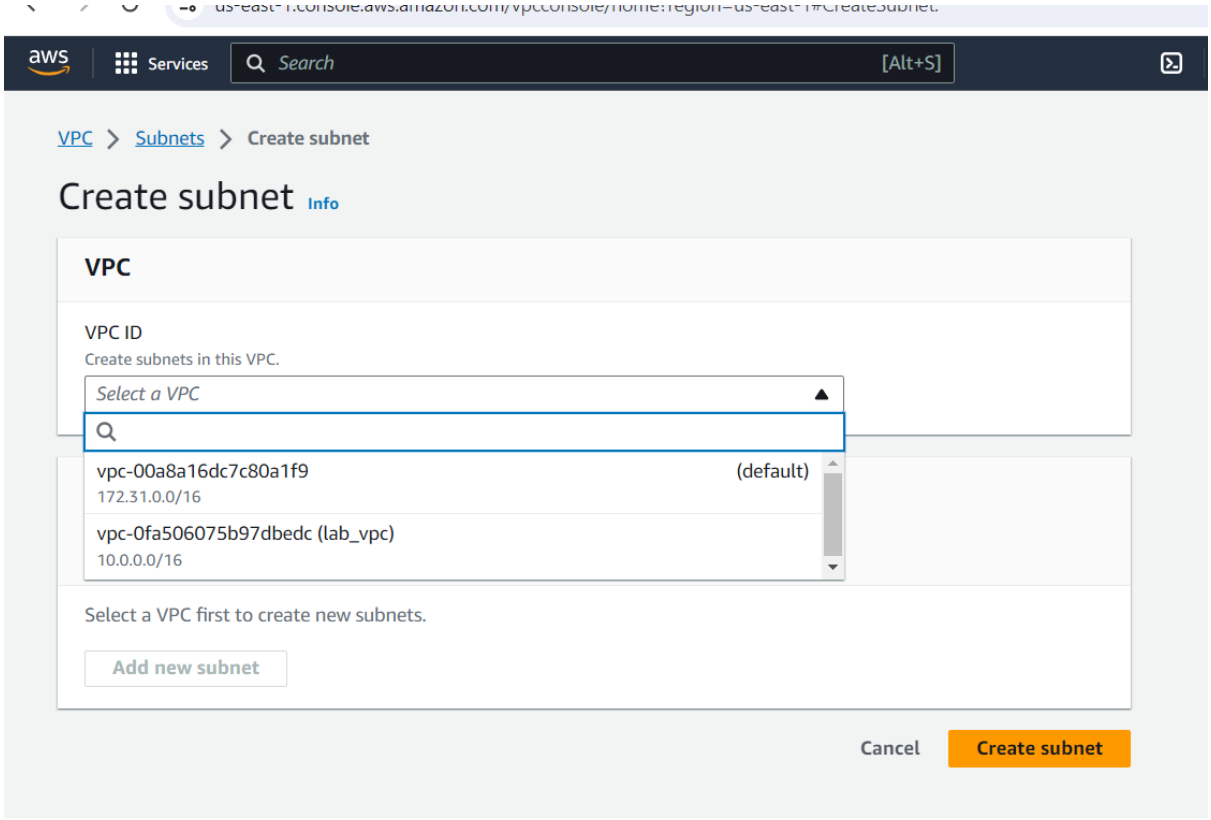
Click on Create Subnet button




On Create Subnet page

For VPC Id:  lab_vpc

Click on Create Subnet button



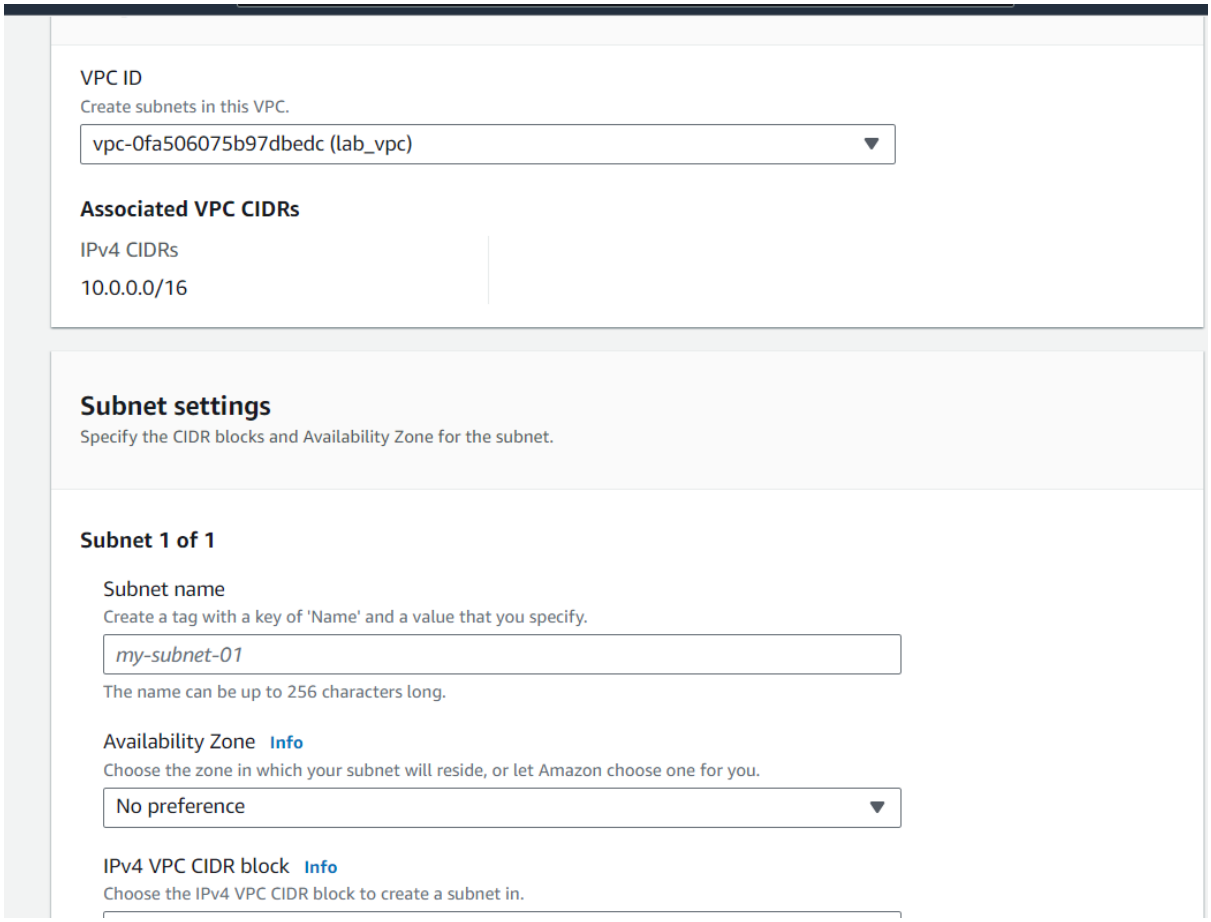
For Subnet Name→ public_subnet

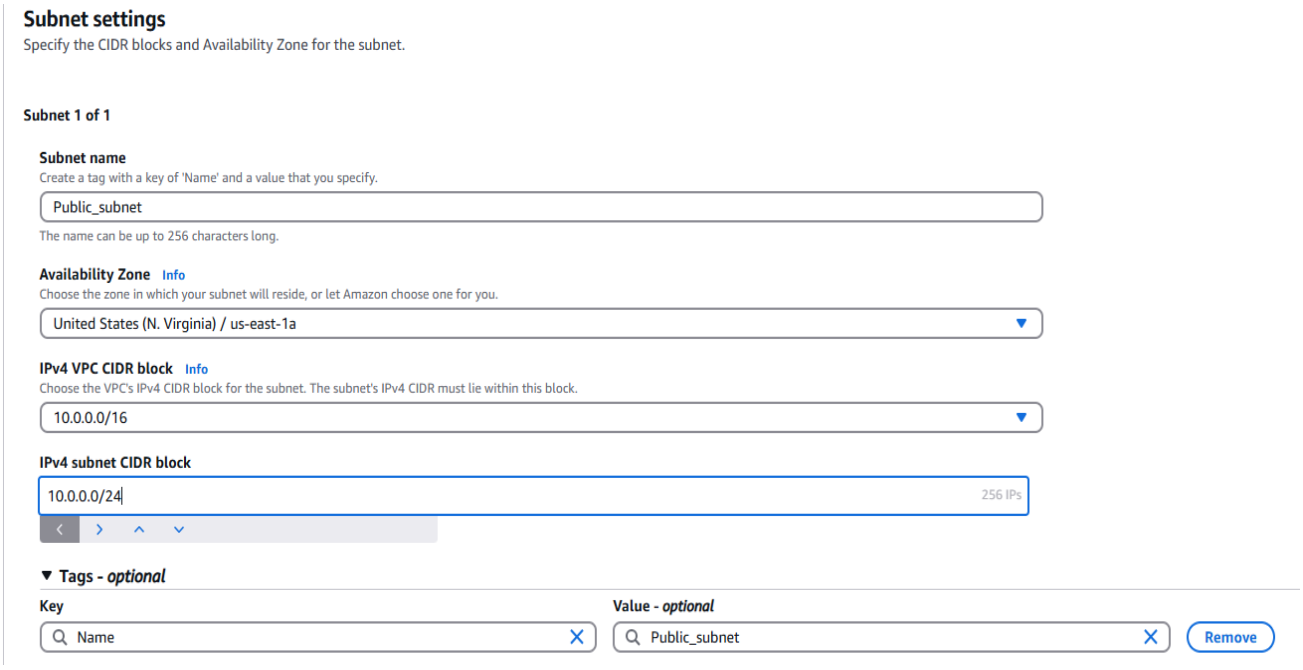
Availability Zone→US East(N.Virginia)/us-east-1a

IPv4 VPC CIDR block→10.0.0.0/16

IPv4 subnet CIDR block→10.0.0.0/24

Click on Create Subnet button






## Step 3:

Click on ADD NEW SUBNET BUTTON

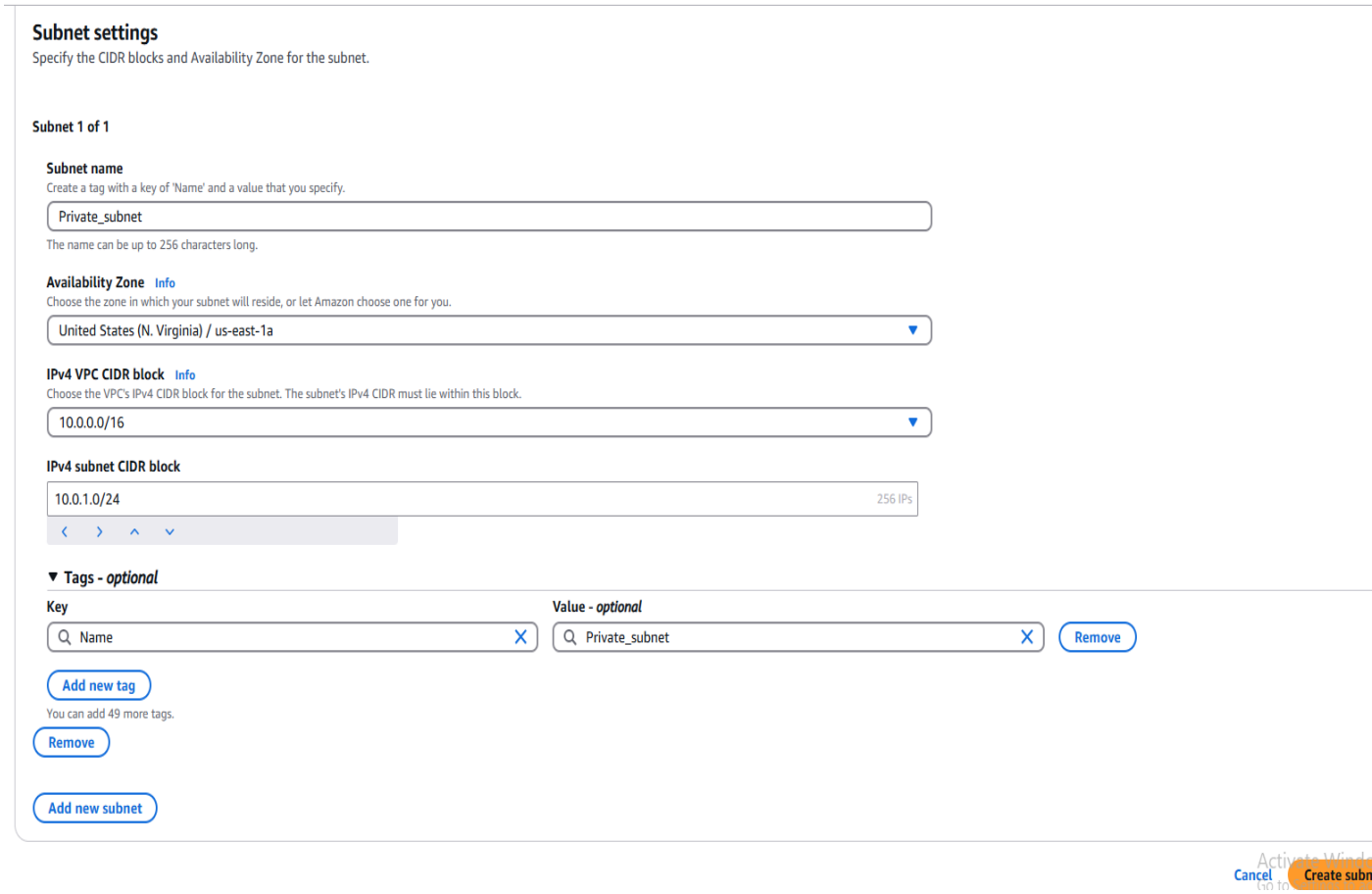
For Subnet Name→ private_subnet

Availability Zone→US East(N.Virginia)/us-east-1a

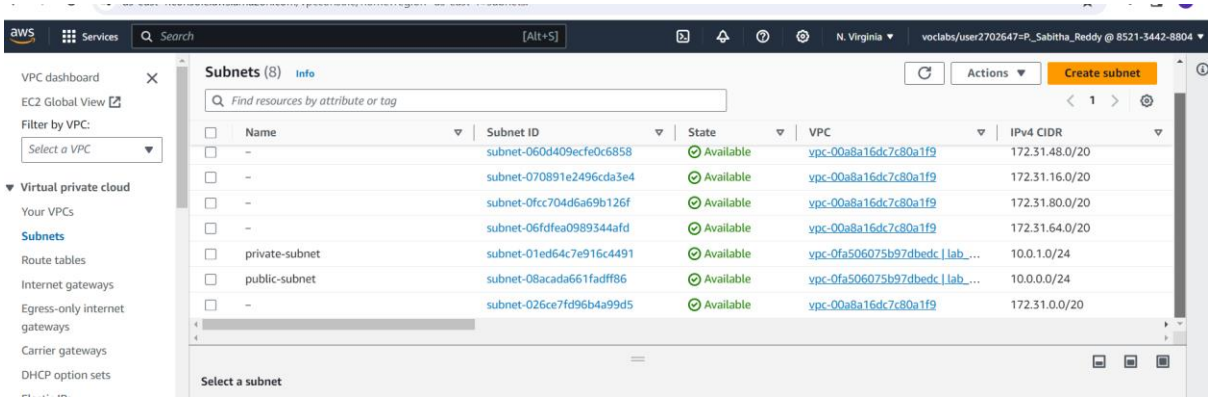
IPv4 VPC CIDR block→10.0.0.0/16

IPv4 subnet CIDR block→10.0.1.0/24

Click on Create Subnet button




Verify public, private subnets are created.

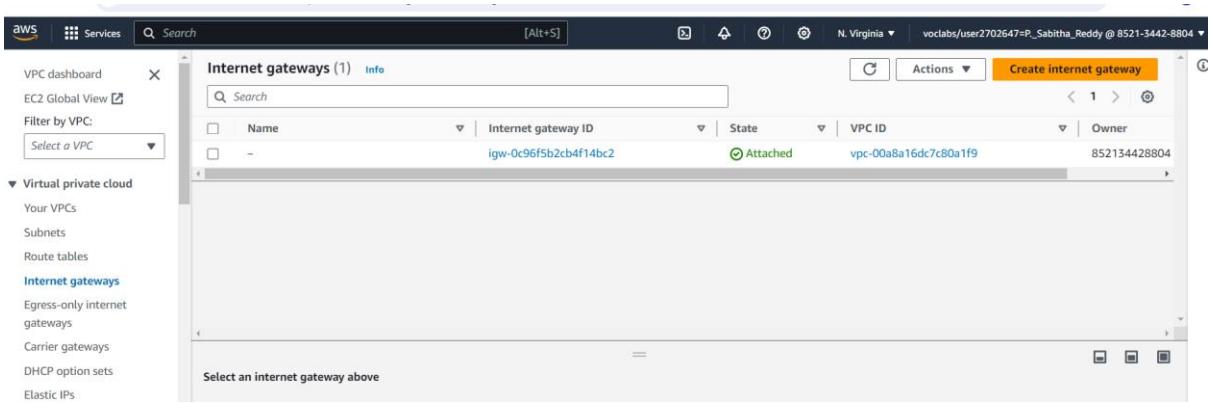



**Step 4:** Create INTERNET GATEWAY and ATTACH TO VPC

In VPC Dashboard Panel

Click on Internet Gateways

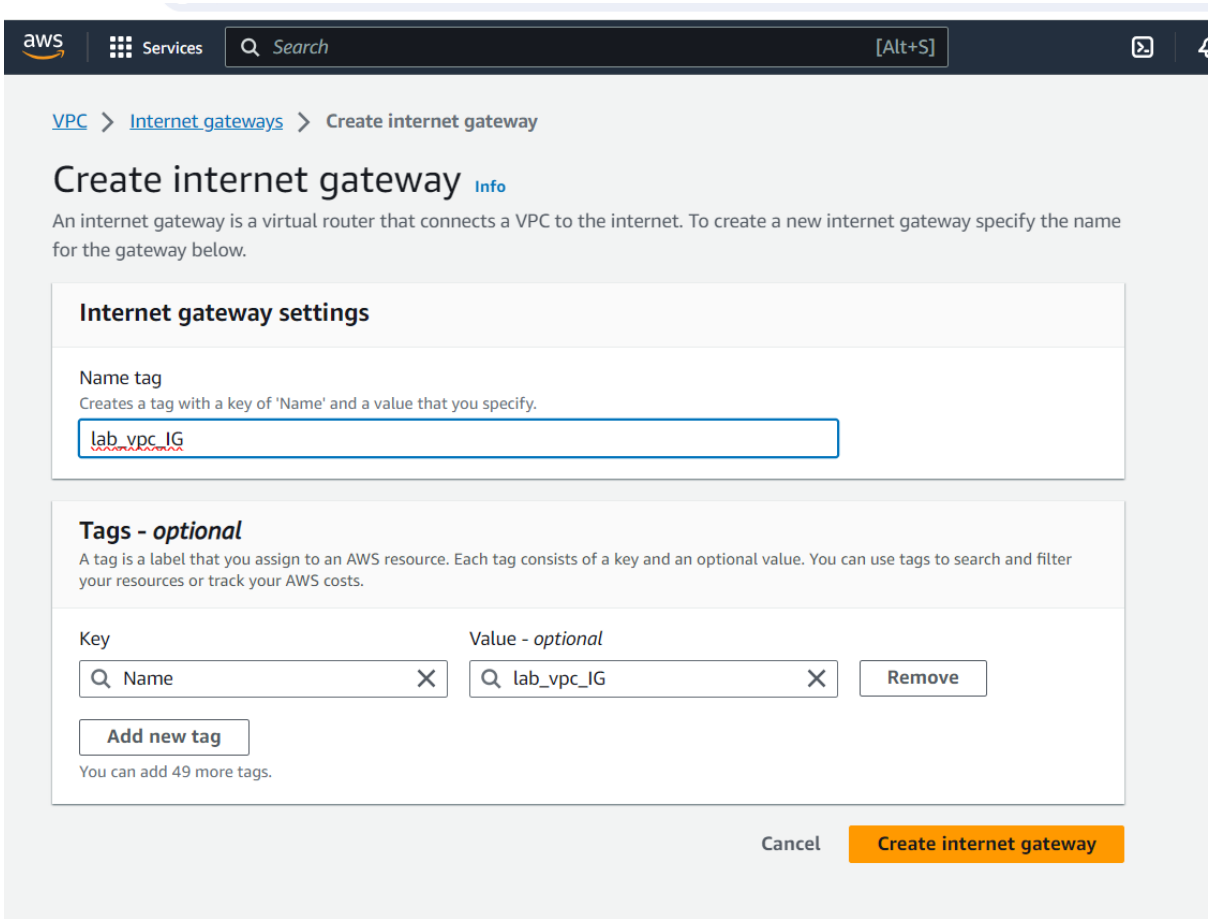
Click on Create  Internet Gateway button




In Create  Internet Gateway page

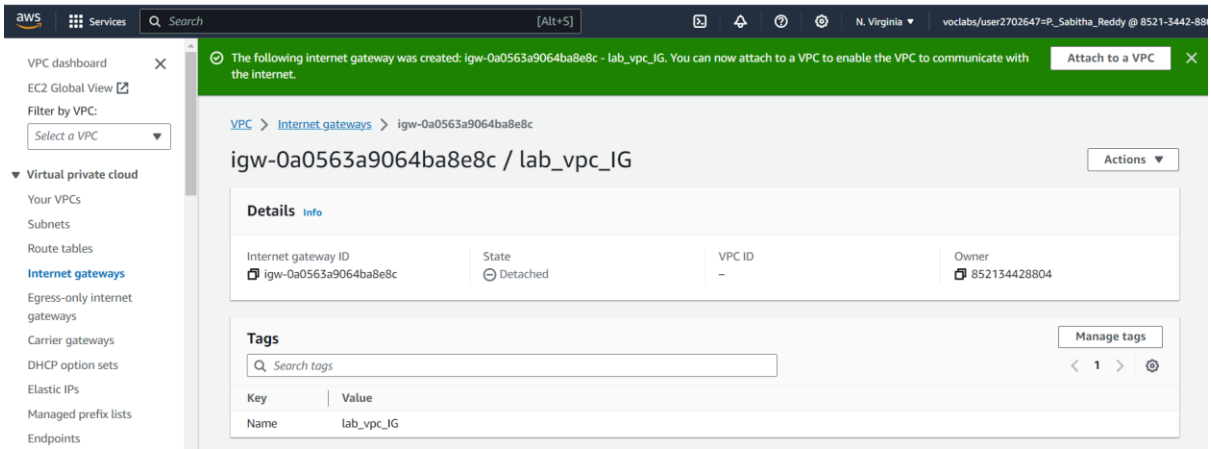
For Name Tag→ lab_vpc_IG

Click on Create  Internet Gateway button




**Verify**

Internet Gateway created.




## Step 5:

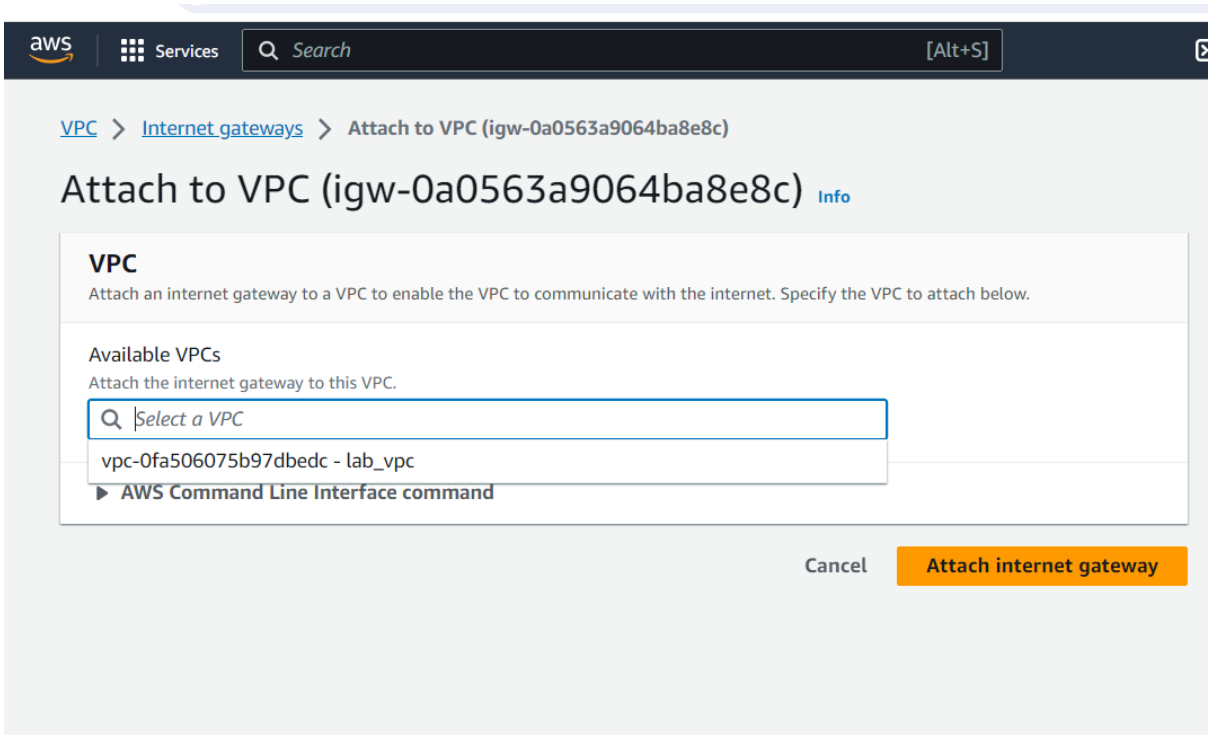
Select lab_vpc_IG

Click on ATTACH to VPC

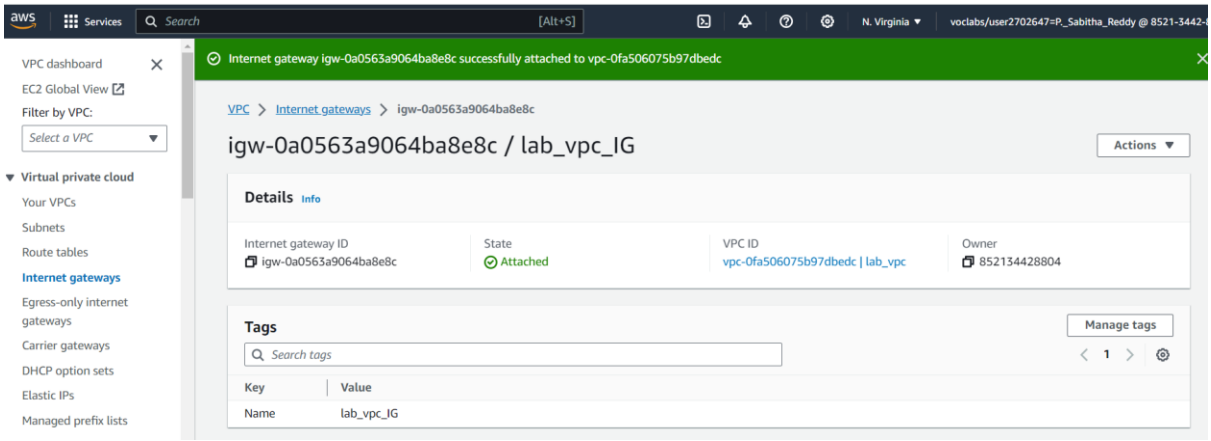
In ATTACH to VPC box

For VPC→lab_vpc

Click on attach internet gateway button.




**Verify** Internet gateway is connected to your VPC

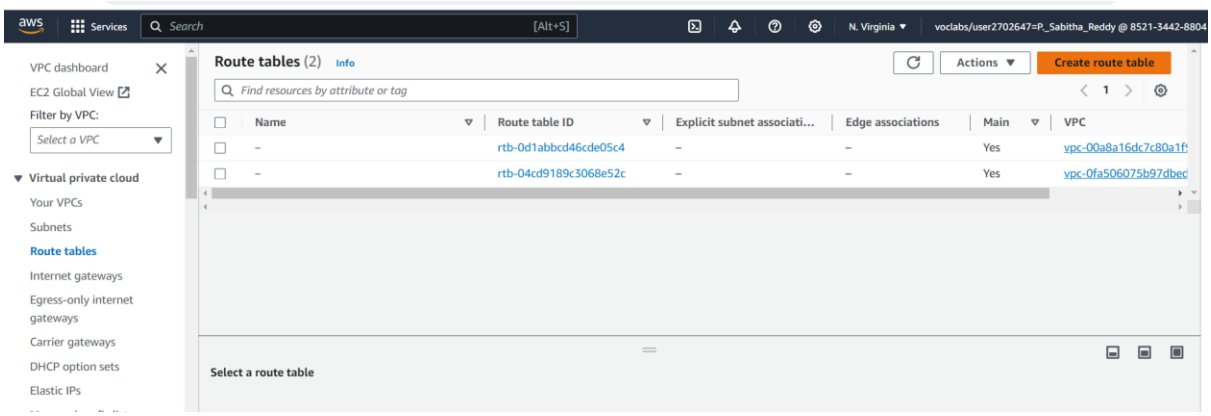



**Step 6:** Create Public Routing Table, associate subnet and add routing rules

On VPC Dashboard panel

Click on Route Table

Click on Create route table button

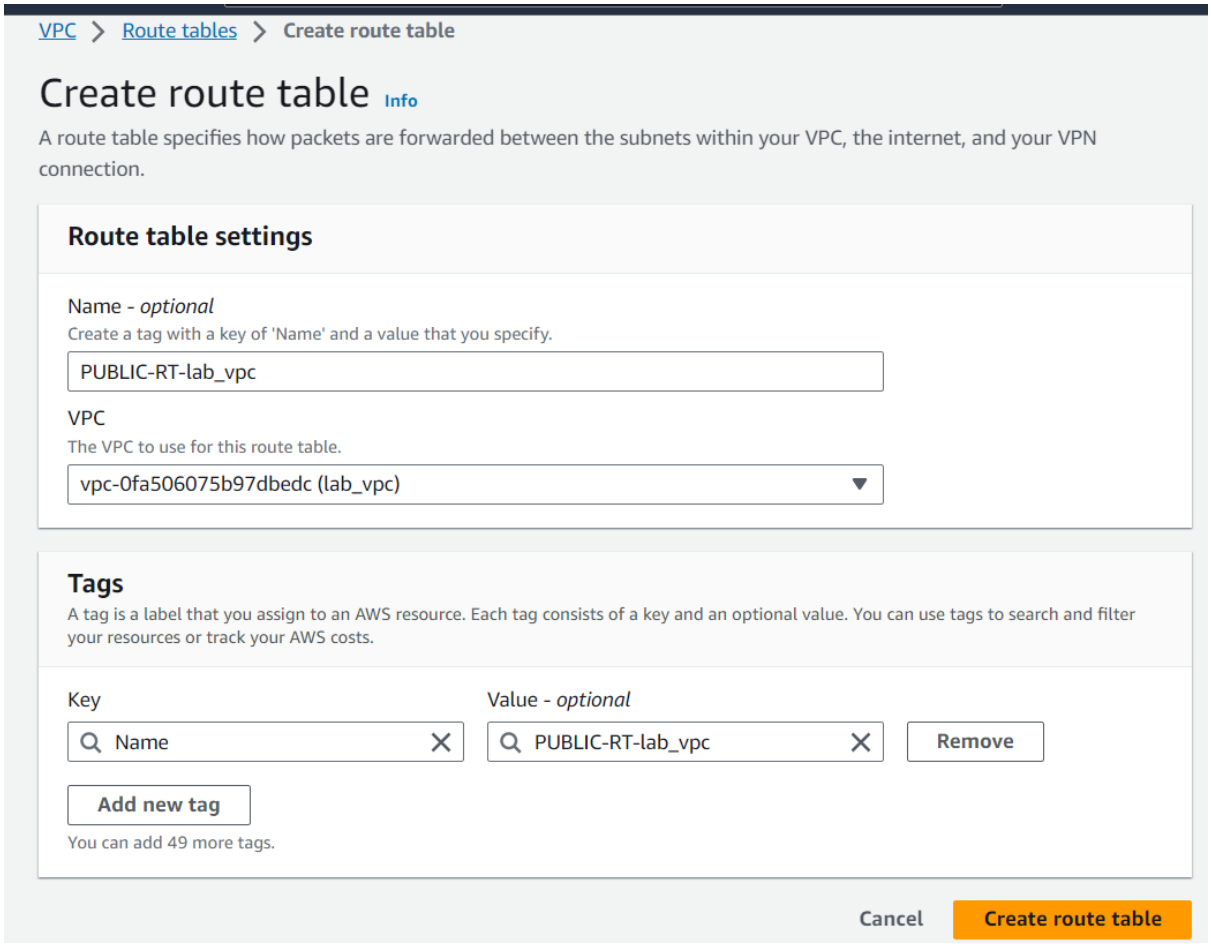



On  route table box

For Name Tag→PUBLIC-RT-lab_vpc

For VPC→ lab_vpc

Click on Create route table button




**Verify ,**

PUBLIC-RT-lab_vpc  Is created.

**Step 7:**

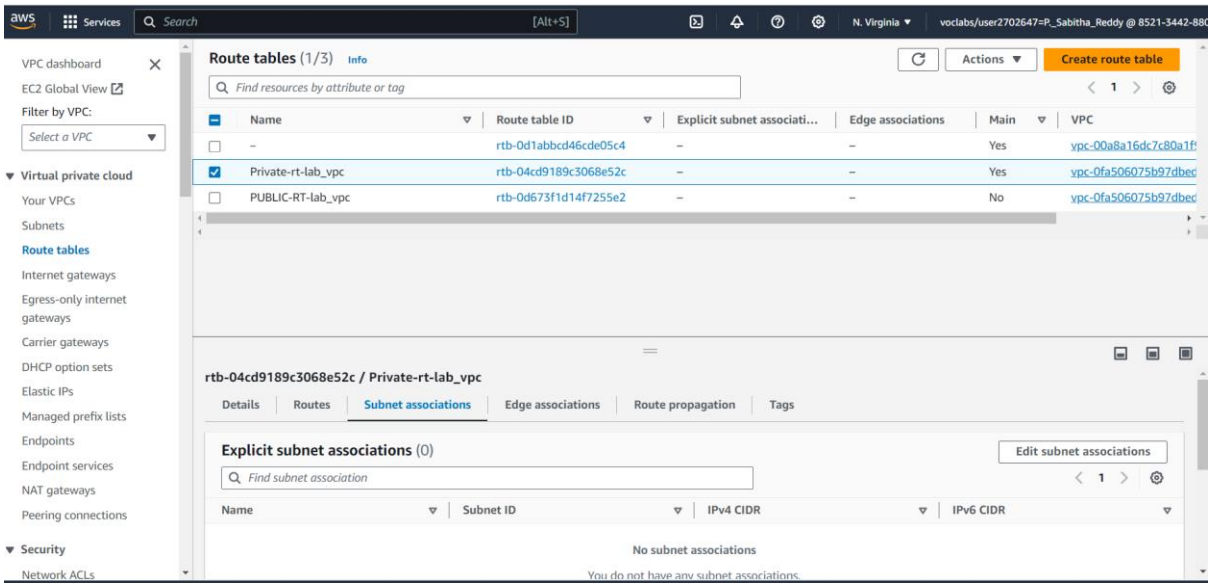
For Name Tag→PRIVATE-RT-lab_vpc

For VPC→ lab_vpc

Click on Create route table button

Select  Private-RT-lab_vpc  →Actions

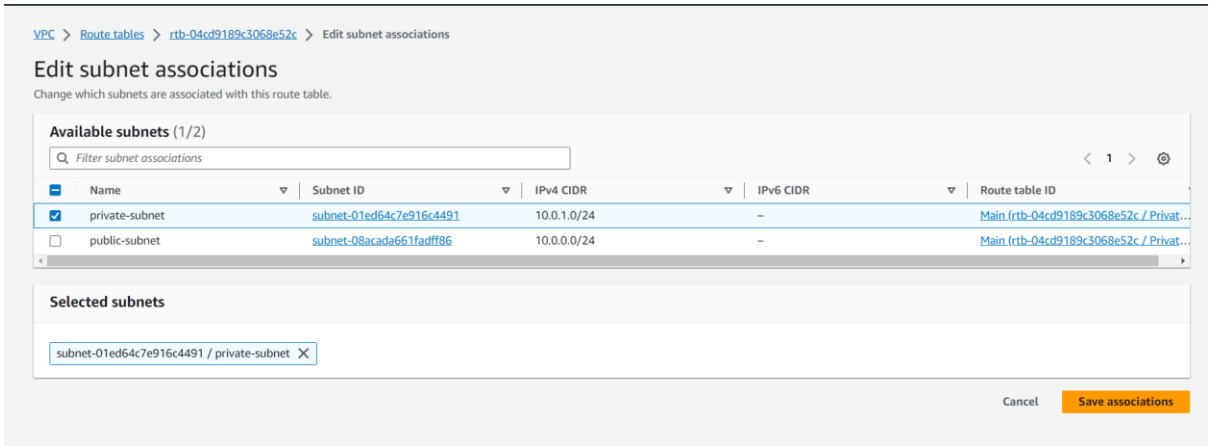
Click on subnet associations




Click on  Edit subnet associations

Select check box of private_subnet→10.0.1.0/24

Click on save associations

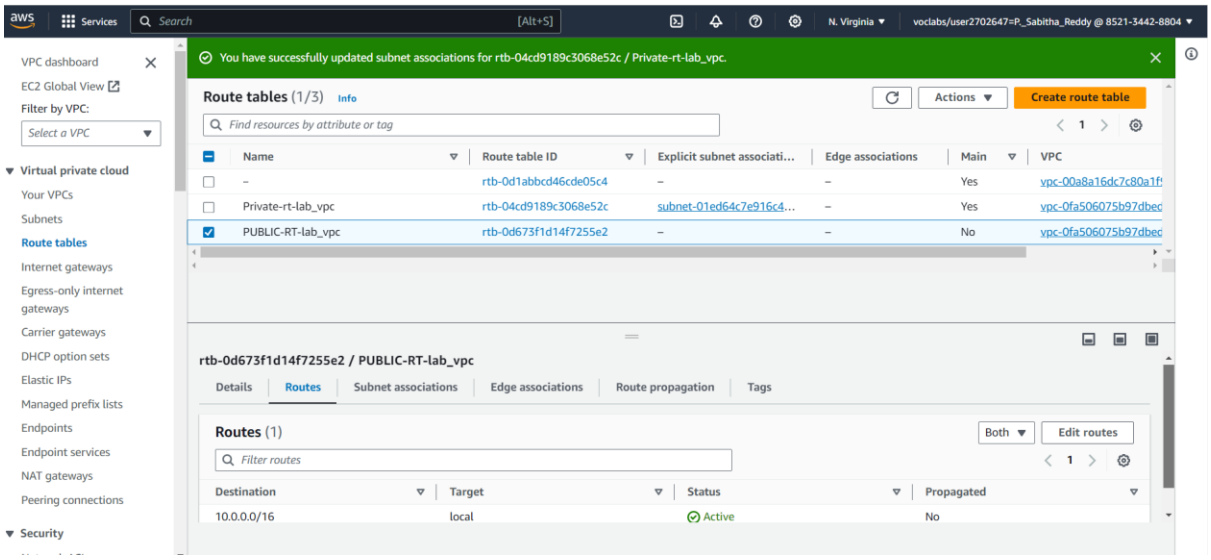



Verify private_subnet is associated with routing table .

Select  PUBLIC-RT-lab_vpc  →Actions

Click on subnet associations

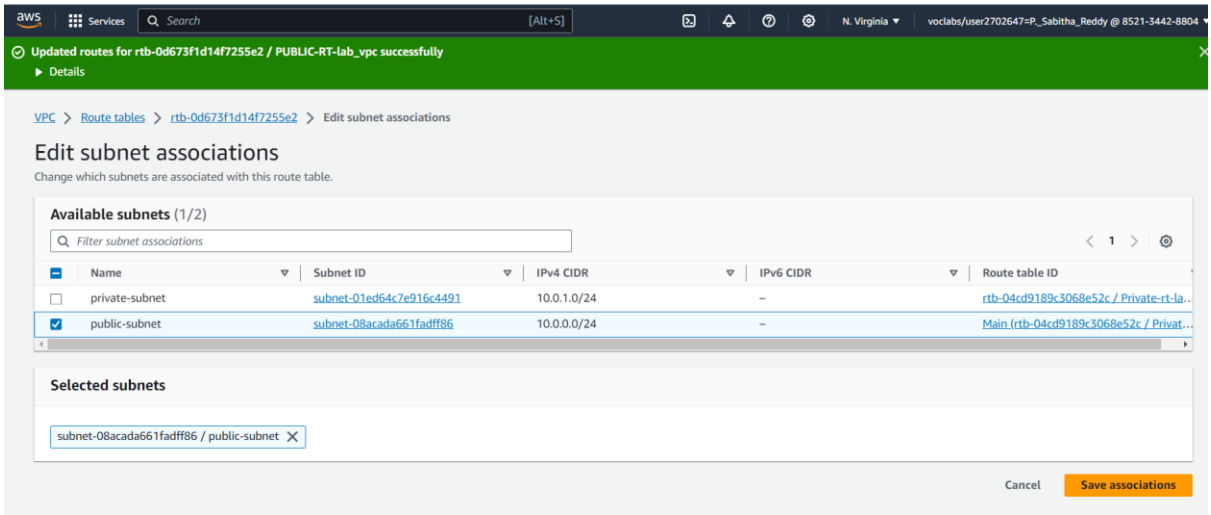
Click on  Edit subnet associations




Select check box of public_subnet→10.0.0.0/24

Click on save associations

Verify public_subnet is associated with routing table




Select PUBLIC-RT-lab_vpc→ Actions

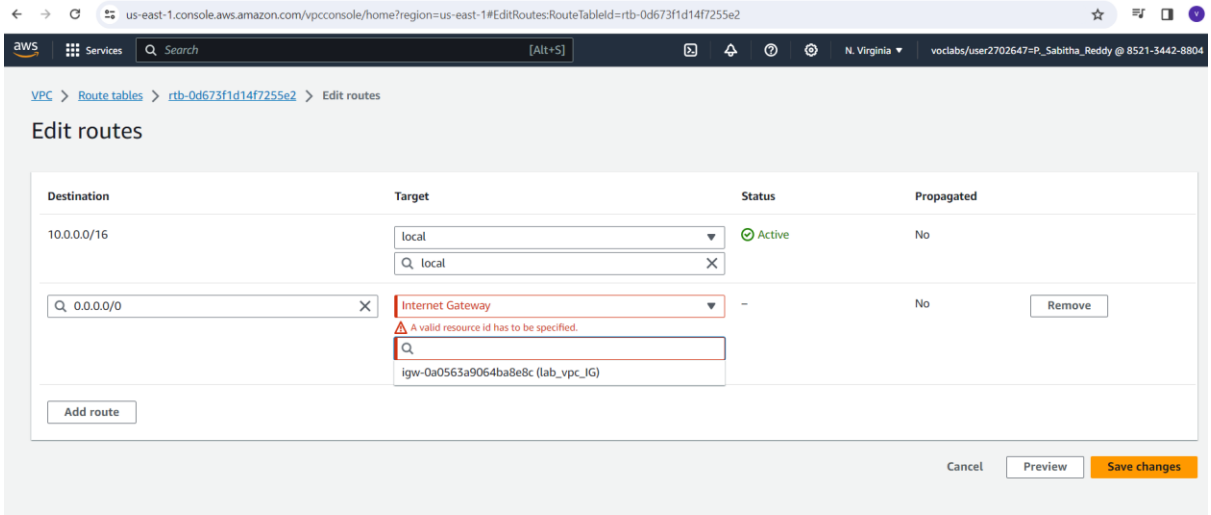
Click on Edit routes button,

Click on add route button,

For Destination→ 0.0.0.0/0

Target→internet gateway→ igw-0a0563a9064ba8e8c(lab_vpc_IG)

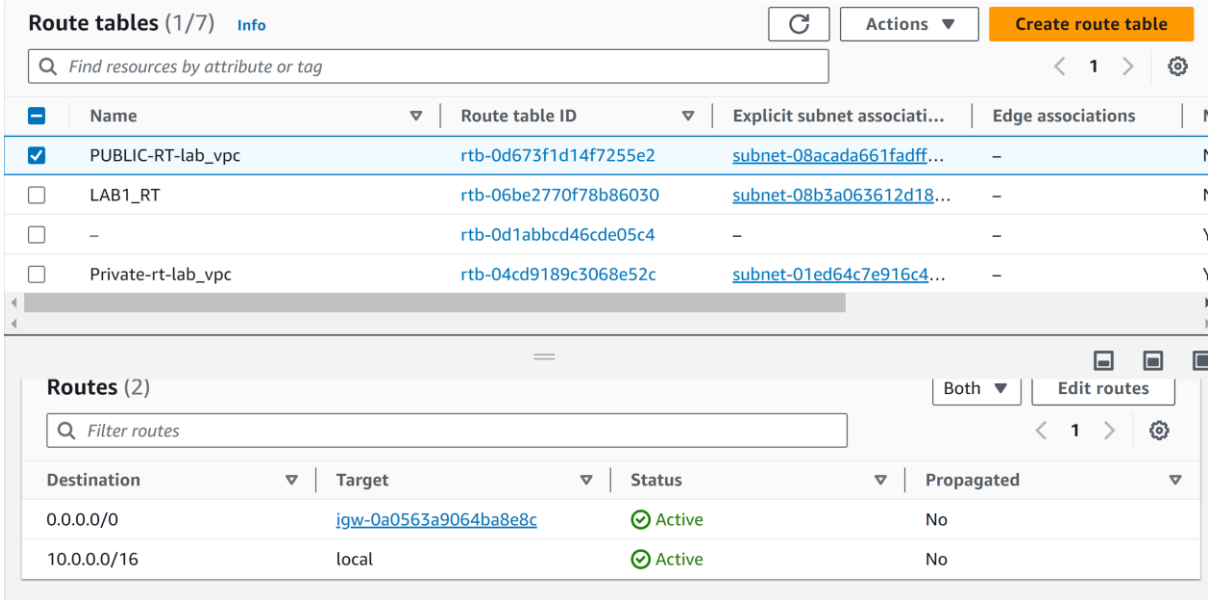
Click on Save changes button




Verification

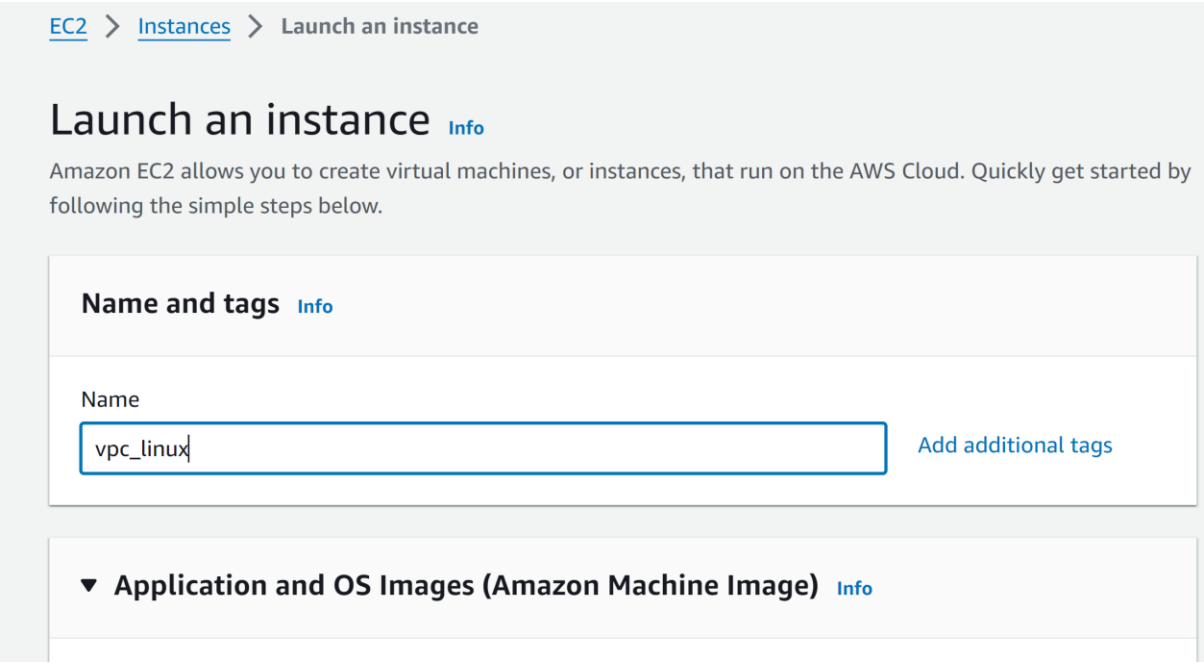
PUBLIC-RT-lab_vpc is added through Internet Gateway

Verify Status column is Active.

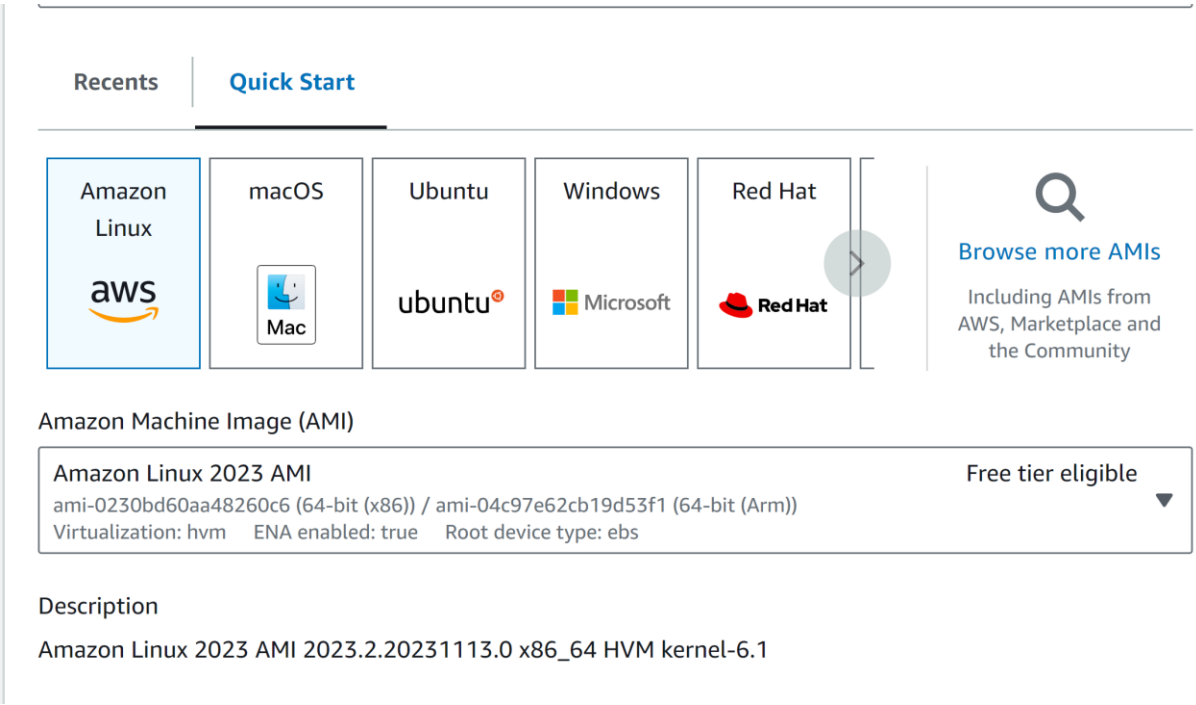



## Step 8:

Create Amazon Linux Instance with lab_vpc



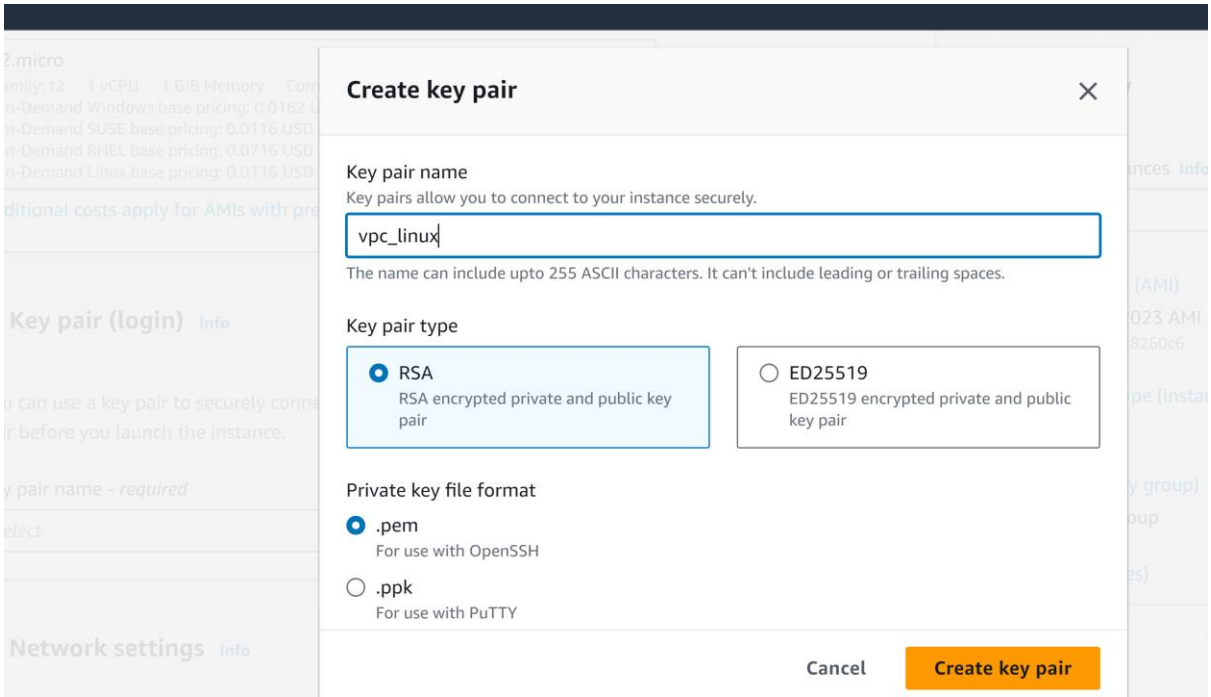

Select AMI →amazon linux, instance type→t2.micro,



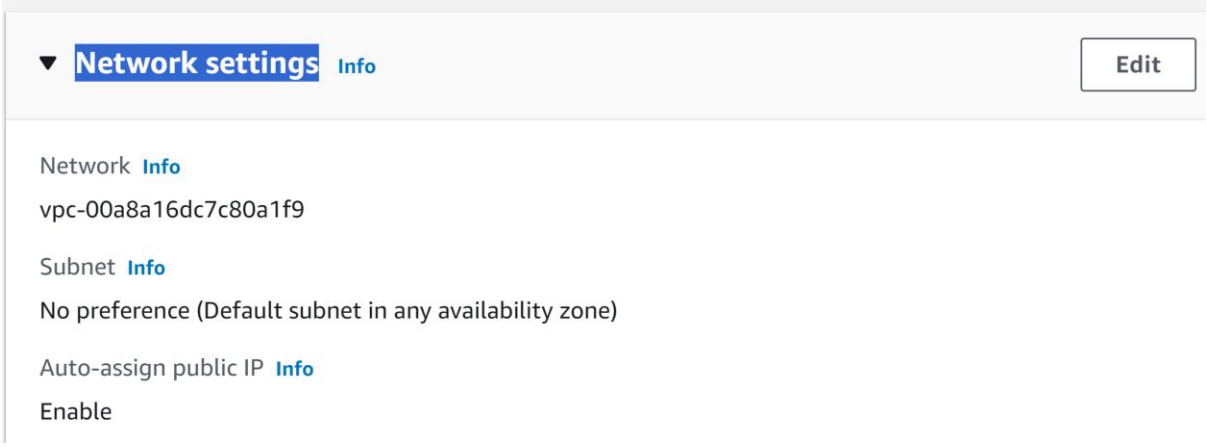

Click on create new key pair,

Key pair name→ vpc_linux

Click on create  key pair



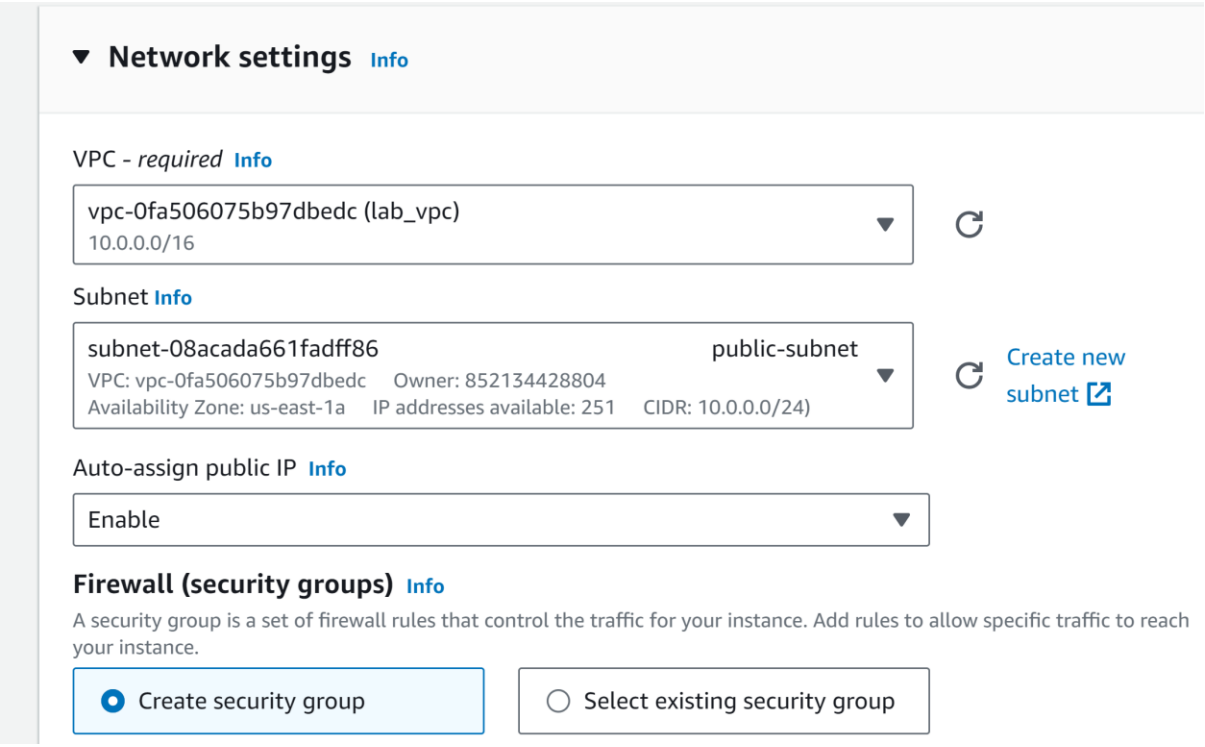

Click on Edit on Network settings




For VPC →lab_vpc,

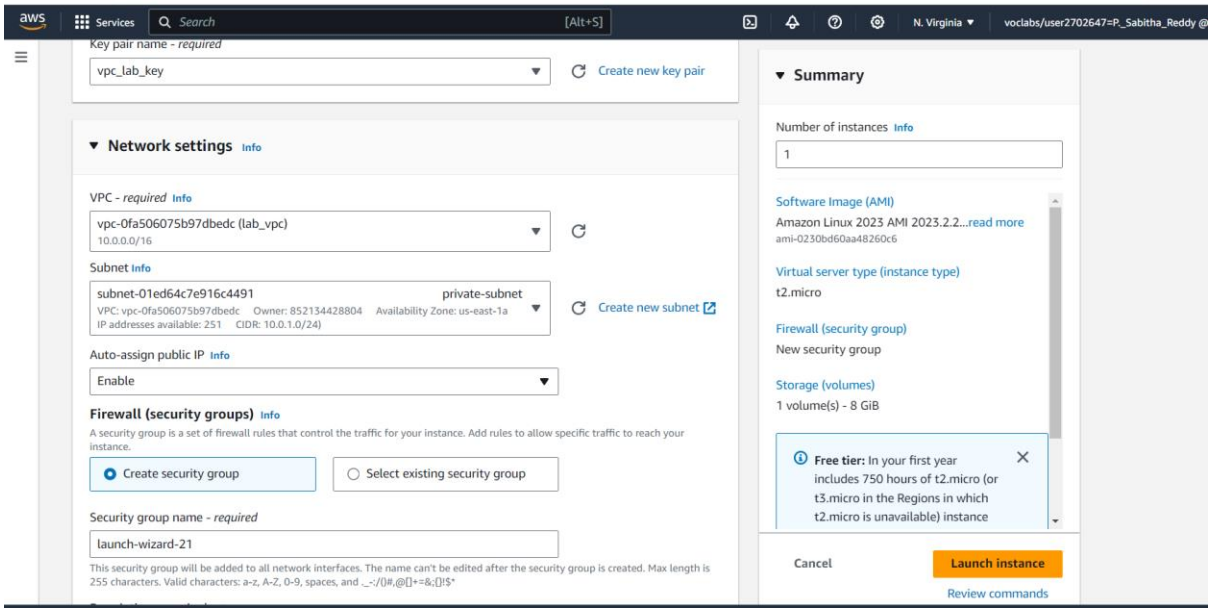
Subnet →public-subnet

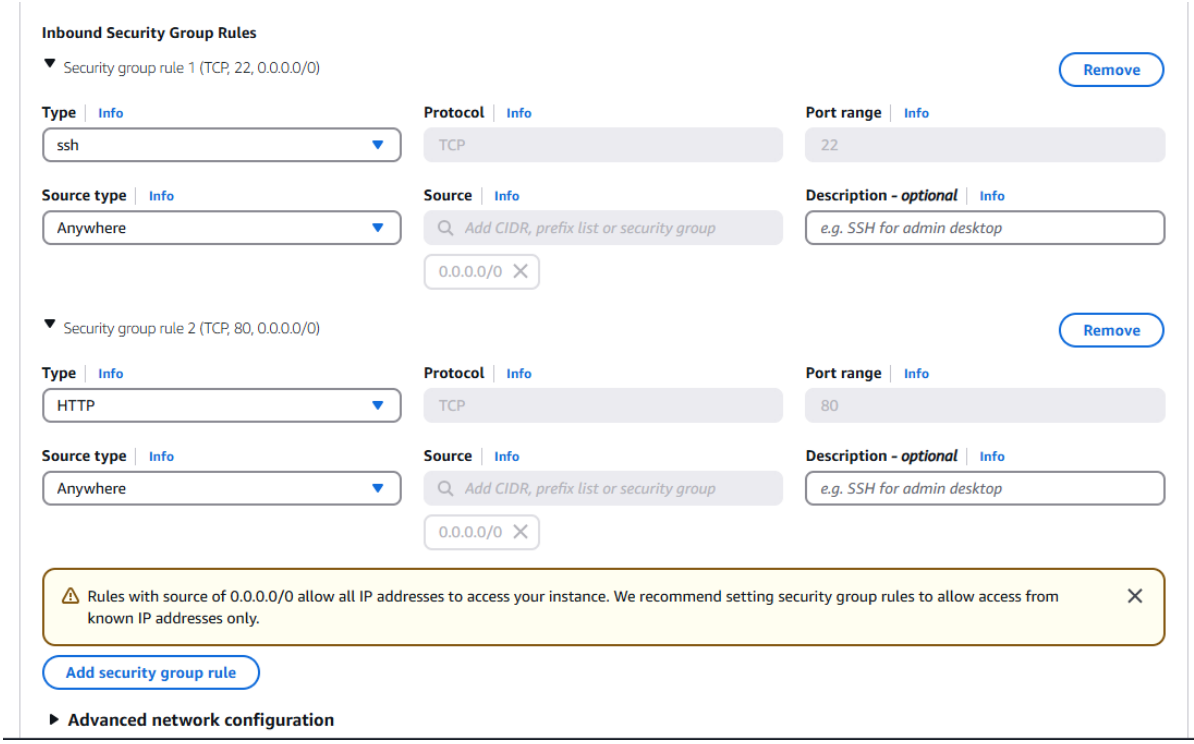
Auto-assign public IP → enable




Configure storage settings are default.

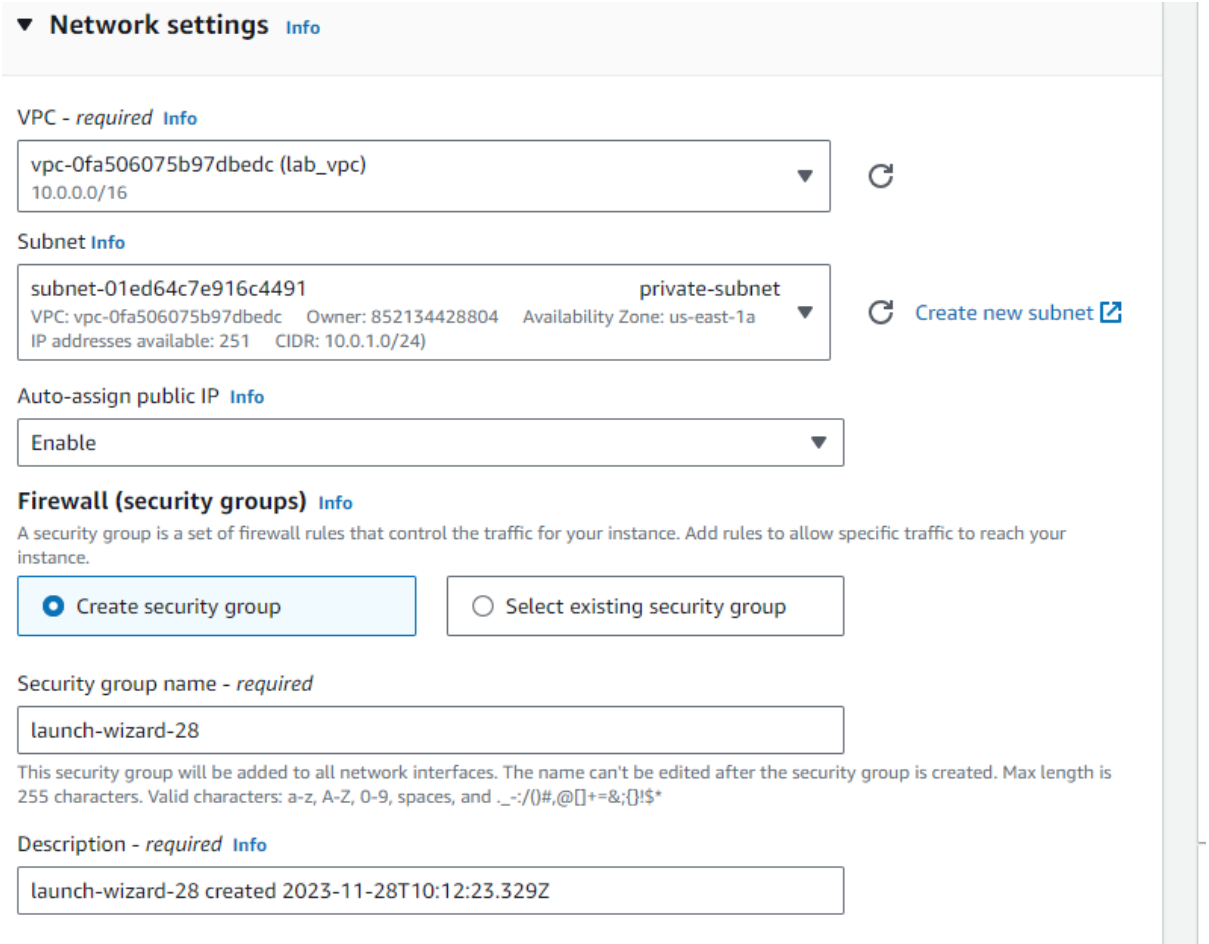
Click on launch Instance






# Step 9:

Create one more instance in same vpc and  subnet is Private-subnet




Configure storage settings are default.

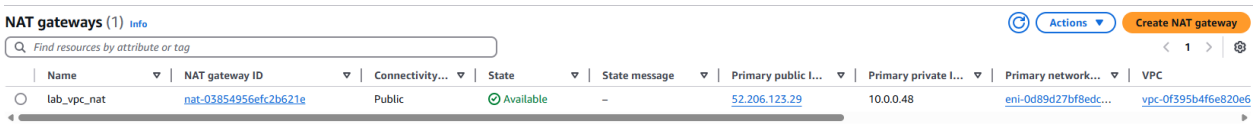
Click on launch Instance

So here we created two amazon linux instances with same VPC and SUBNETS are PRIVATE & PUBLIC subnets.

**Step 10:**

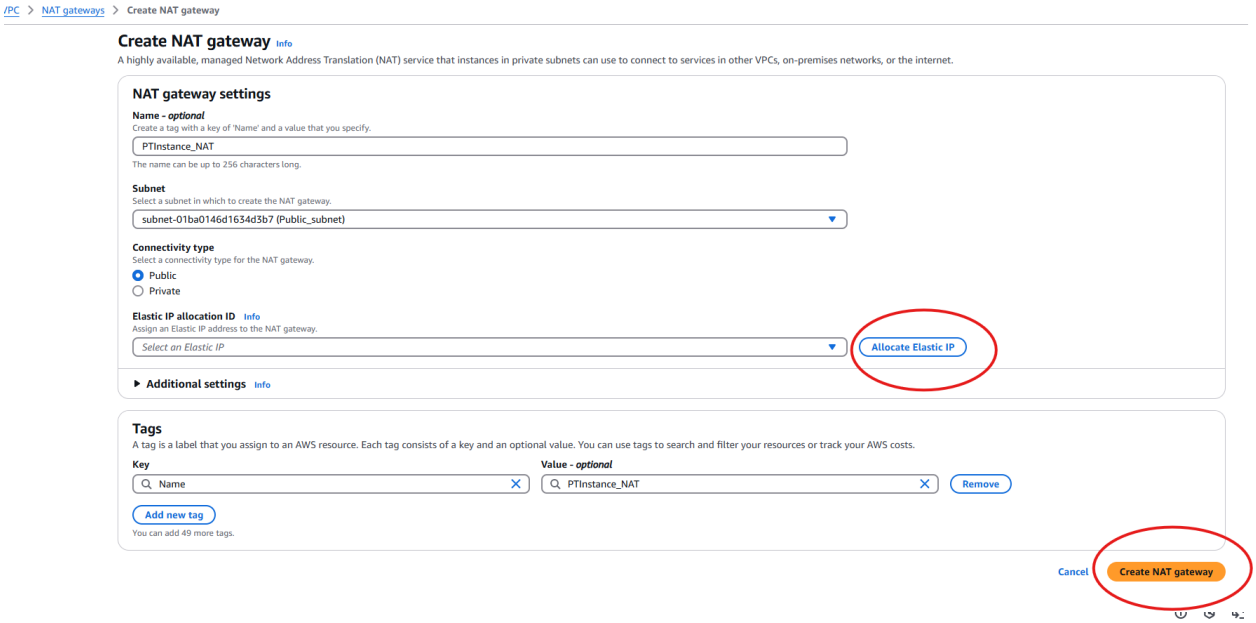
**Create NAT Gateway :**

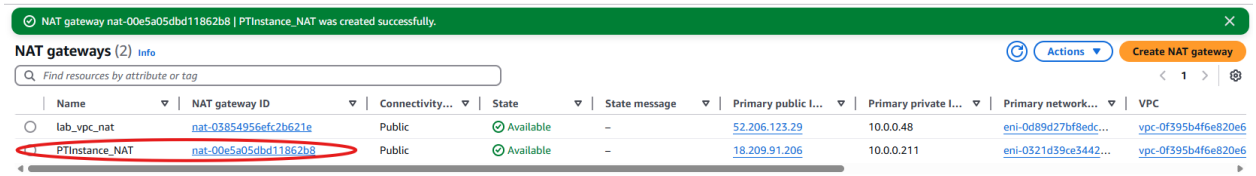
On VPC Dashboard panel→ NAT Gateways



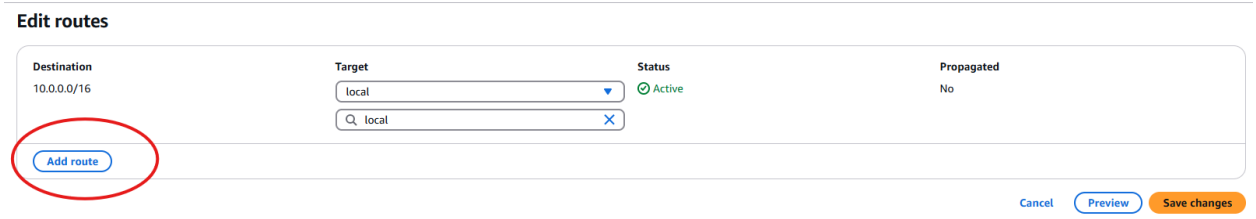
Click on Create NAT Gateway button→**Name tag**:PTInstance_NAT

**Subnet :** Public_subnet,
**Connectivity type:** Public,
**Elastic IP allocation ID:** Click on allocate Elastic IP button

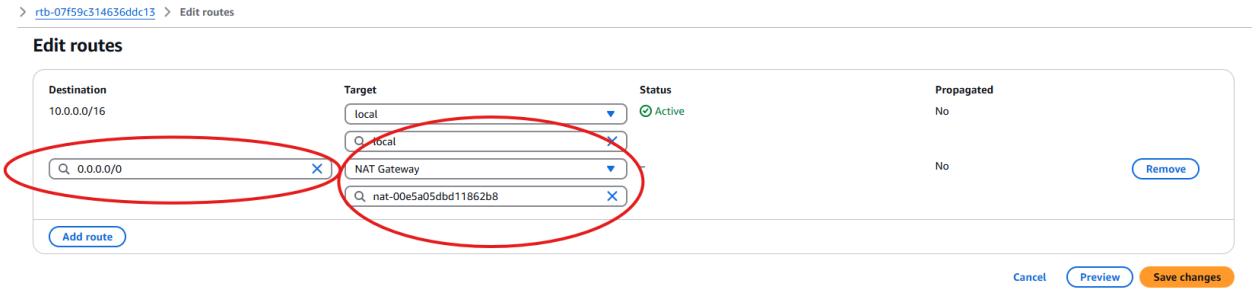


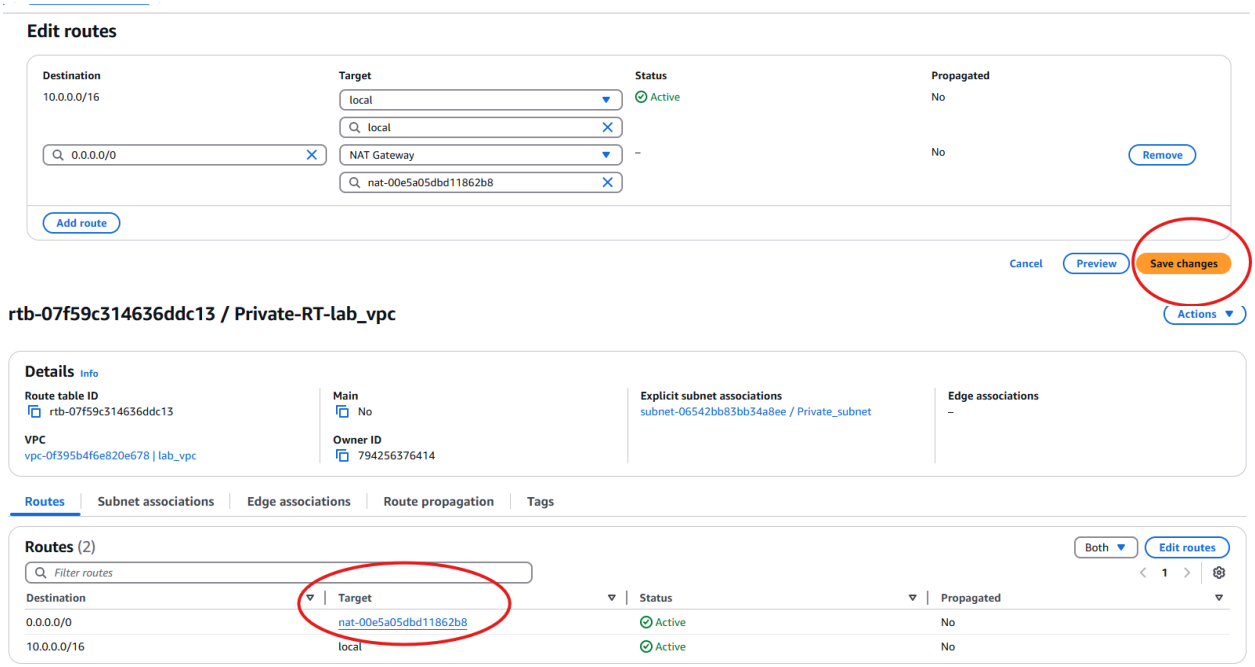

Then Attach this NAT Gateway to Private Route Table,
Click on **Route Tables→select** Private-RT-lab_vpc →Actions→edit Routes→add route→




**Destination** is Public IP address,0.0.0.0/0,**Target :** NAT Gateway, select PTInstance_NAT




Click on Save Changes.




**Step 11:**
**Connect to public instance and check internet connectivity**.

Connect the Public Linux instance with **SSH client** connection or EC2 **Instance Connect**.

If connecting with **SSH Client** ,open command prompt in local system and change path for folder where your Linux instance key pair available in local system.
Then paste the instance path.

EC2 > Instances > i-0a567ad3fdafd3827 > Connect to instance

## Connect to instance Info

Connect to your instance i-0a567ad3fdafd3827 (serverinstance) using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

**Instance ID**
i-0a567ad3fdafd3827 (serverinstance)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is serverkey.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "serverkey.pem"
4. Connect to your instance using its Public DNS:
   ec2-3-83-217-134.compute-1.amazonaws.com

Example:

```
ssh -i "serverkey.pem" ec2-user@ec2-3-83-217-134.compute-1.amazonaws.com
```

**Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

ec2-user@ip-10-0-0-100:~

```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SOE III\Downloads>ssh -i "serverkey.pem" ec2-user@44.200.243.70
The authenticity of host '44.200.243.70 (44.200.243.70)' can't be established.
ED25519 key fingerprint is SHA256:hFiB68JZAc9TCJhsEkYjZx6dE4pHKYhec5s/8C2rKxs.
This host key is known by the following other names/addresses:
    C:\Users\SOE III/.ssh/known_hosts:8: 44.214.91.172
    C:\Users\SOE III/.ssh/known_hosts:10: 44.222.243.177
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.200.243.70' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Mar 18 07:28:47 2025 from 119.235.52.196
[ec2-user@ip-10-0-0-100 ~]$
```

To check internet Connectivity in Linux instance
Command: ping www.google.com

ec2-user@ip-10-0-0-100:~

```
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Mar 18 07:28:47 2025 from 119.235.52.196
[ec2-user@ip-10-0-0-100 ~]$ ping www.google.com
PING www.google.com (142.250.31.99) 56(84) bytes of data.
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=1 ttl=58 time=2.66 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=2 ttl=58 time=2.64 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=3 ttl=58 time=2.65 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=4 ttl=58 time=2.42 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=5 ttl=58 time=2.64 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=6 ttl=58 time=2.41 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=7 ttl=58 time=2.65 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=8 ttl=58 time=2.61 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=9 ttl=58 time=2.96 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=10 ttl=58 time=2.79 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=11 ttl=58 time=2.61 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=12 ttl=58 time=2.34 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=13 ttl=58 time=2.53 ms
^C
--- www.google.com ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12021ms
rtt min/avg/max/mdev = 2.337/2.608/2.957/0.157 ms
[ec2-user@ip-10-0-0-100 ~]$
```

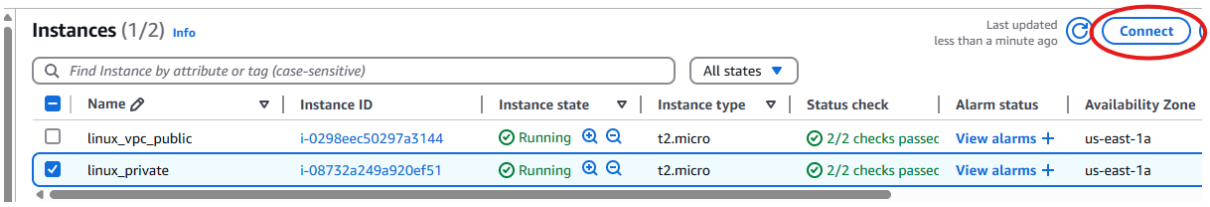
If data packet's are transmitted without any loss then can understood that instance have internet connection.
with this we can say that Public Instance have internet connectivity.

**Step 12:**

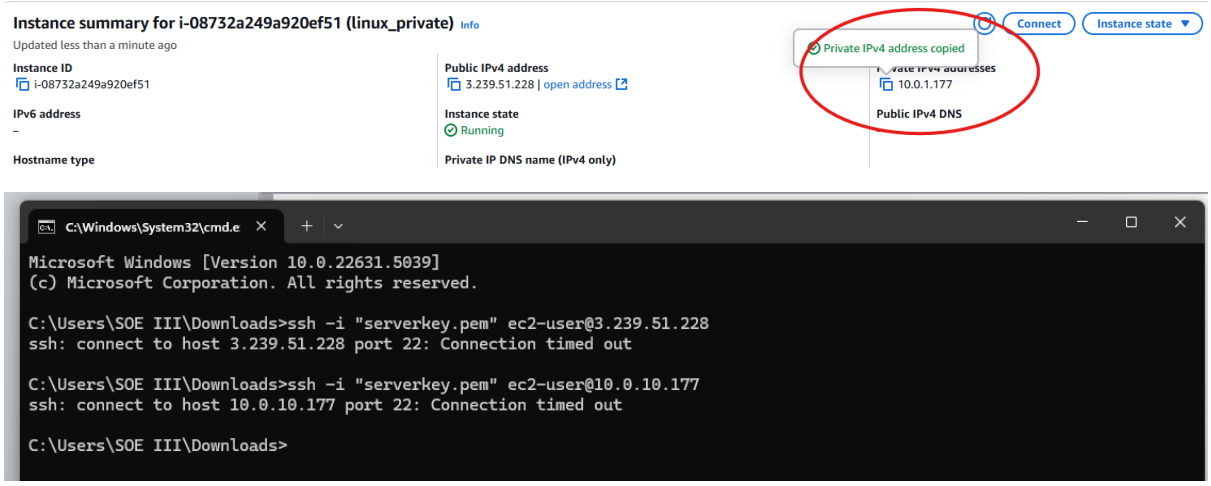
**Connect to private instance and check internet connectivity**.

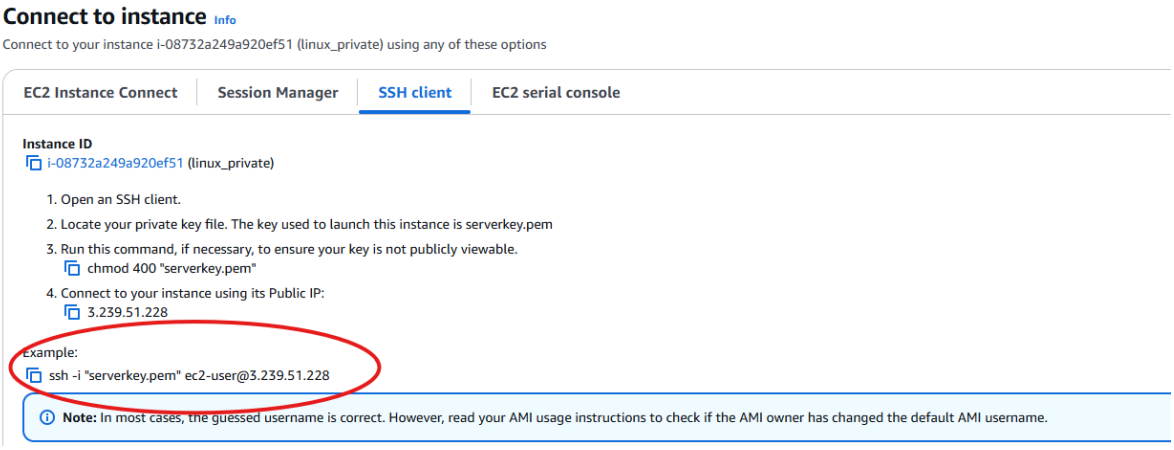
Connect the Private Linux instance with **SSH client** connection or EC2 **Instance Connect**.

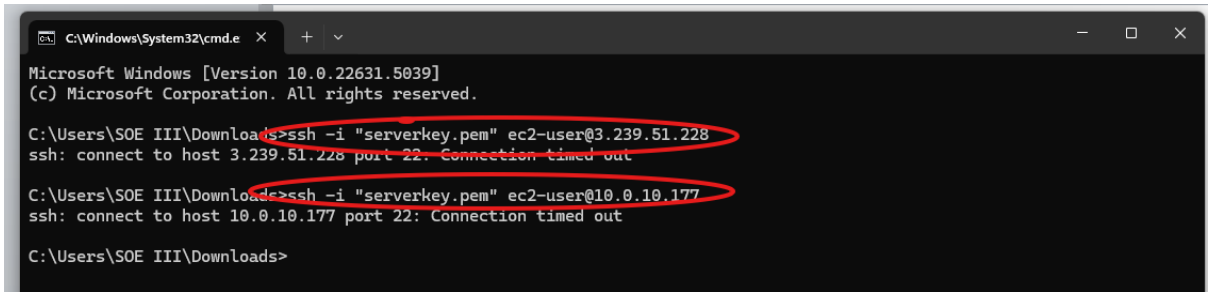
If connecting with **SSH Client** ,open command prompt in local system and change path for folder where your Linux instance key pair available in local system.
Then paste the instance path.




To connect with the private linux instance need to use Private ip Address






With Both Public and Private IP addresses also we can't connect the instance from outside because it is launched in the Private Subnet.

**To connect the private instance**

From **public instance** we can connect the private instance.

For that Key pair file should available in the Public instance.

To copy files from local system to Public Linux instance need to use **SCP** command:

## Steps to connect private instance

### In the public Linux Instance CMD:

Execute below commands:

- **Change user from ec2-user to root:**

**Sudo Su**

- **Open the html folder in public linux instace:**

## Cd /var/www/html

- **Give owner permission to ec2-user**

## Chown ec2-user .

```
^C
--- www.google.com ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12021ms
rtt min/avg/max/mdev = 2.337/2.608/2.957/0.157 ms
[ec2-user@ip-10-0-0-100 ~]$ sudo su
[root@ip-10-0-0-100 ec2-user]# cd /var/www/html
[root@ip-10-0-0-100 html]# chown ec2-user
chown: missing operand after 'ec2-user'
Try 'chown --help' for more information.
[root@ip-10-0-0-100 html]# chown ec2-user .
[root@ip-10-0-0-100 html]#
```

## Open Cmd in Local System:

- ✓ Open path where key pair file available in the command prompt
- ✓ Scp –i ./serverkey.pem ./serverkey.pem ec2-user@ 44.200.243.70 :/var/www/html

```
C:\Users\SOE III\Downloads>scp -i ./serverkey.pem ./serverkey.pem ec2-user@44.200.243.70:/var/www/html
serverkey.pem                                                                 100% 1678     6.9KB/s   00:00

C:\Users\SOE III\Downloads>
```

Successfully pem file transferred top public Linux instance

Check the file is available in public Linux instance

Use command in linux instance : **ls**

```
root@ip-10-0-0-100:/var/www
       _/m/'
Last login: Tue Mar 18 07:28:47 2025 from 119.235.52.196
[ec2-user@ip-10-0-0-100 ~]$ ping www.google.com
PING www.google.com (142.250.31.99) 56(84) bytes of data.
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=1 ttl=58 time=2.66 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=2 ttl=58 time=2.64 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=3 ttl=58 time=2.65 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=4 ttl=58 time=2.42 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=5 ttl=58 time=2.64 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=6 ttl=58 time=2.41 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=7 ttl=58 time=2.65 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=8 ttl=58 time=2.61 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=9 ttl=58 time=2.96 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=10 ttl=58 time=2.79 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=11 ttl=58 time=2.61 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=12 ttl=58 time=2.34 ms
64 bytes from bj-in-f99.1e100.net (142.250.31.99): icmp_seq=13 ttl=58 time=2.53 ms
^C
--- www.google.com ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12021ms
rtt min/avg/max/mdev = 2.337/2.608/2.957/0.157 ms
[ec2-user@ip-10-0-0-100 ~]$ sudo su
[root@ip-10-0-0-100 ec2-user]# cd /var/www/html
[root@ip-10-0-0-100 html]# chown ec2-user
chown: missing operand after 'ec2-user'
Try 'chown --help' for more information.
[root@ip-10-0-0-100 html]# chown ec2-user .
[root@ip-10-0-0-100 html]# ls
index.html  serverkey.pem
```

Connecting private linux instance from public linux instance using private IPaddress of private linux instance.

ssh –i “serverkey.pem” ec2-user@10.0.1.177

```
[root@ip-10-0-0-100 html]# chown ec2-user .
[root@ip-10-0-0-100 html]# ls
index.html  serverkey.pem
[root@ip-10-0-0-100 html]# ssh -i "serverkey.pem" ec2-user@10.0.1.177
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Mar 18 08:43:54 2025 from 10.0.0.100
[ec2-user@ip-10-0-1-177 ~]$
```

Successfully connected private instance from the public Linux instance.

Check the internet connectivity in private linux instance

**Command:** ping www.google.com

```
ec2-user@ip-10-0-1-177:~

  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Mar 18 08:43:54 2025 from 10.0.0.100
[ec2-user@ip-10-0-1-177 ~]$ ping www.google.com
PING www.google.com (142.251.16.106) 56(84) bytes of data.
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=1 ttl=54 time=2.83 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=2 ttl=54 time=2.45 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=3 ttl=54 time=2.33 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=4 ttl=54 time=2.35 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=5 ttl=54 time=2.48 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=6 ttl=54 time=2.81 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=7 ttl=54 time=2.67 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=8 ttl=54 time=2.34 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=9 ttl=54 time=3.06 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=10 ttl=54 time=2.47 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=11 ttl=54 time=2.46 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=12 ttl=54 time=2.33 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=13 ttl=54 time=2.48 ms
64 bytes from bl-in-f106.1e100.net (142.251.16.106): icmp_seq=14 ttl=54 time=2.35 ms
^C
--- www.google.com ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13022ms
rtt min/avg/max/mdev = 2.325/2.528/3.064/0.221 ms
[ec2-user@ip-10-0-1-177 ~]$
```

If data packets are transmitted without any loss then can understood that instance have internet connection.
With this we can say that Public Instance have internet connectivity.