Lesson-End Project

Customizing EC2 Instance for Web Instance

Project agenda: To create a VPC and launch a web instance for hosting your application

Description: As a cloud engineer at a tech firm, you need to set up a custom network infrastructure. This involves creating a custom VPC, attaching a subnet, setting up an internet gateway and route table, and launching an EC2 instance. This setup will ensure a secure and scalable environment for your application.

Tools required: AWS Management Console

Prerequisites: AWS account

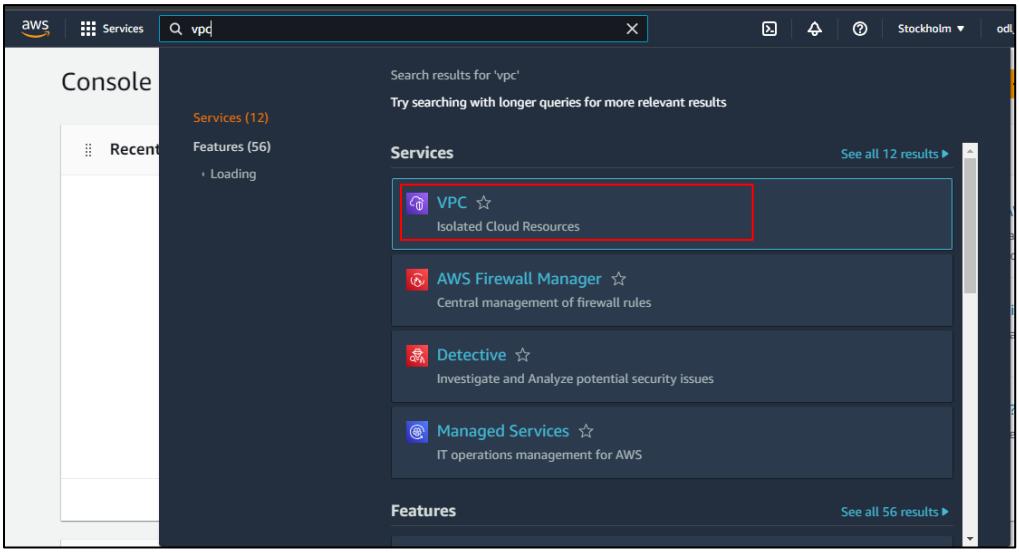
Expected deliverables: EC2 web instance

Steps to be followed:

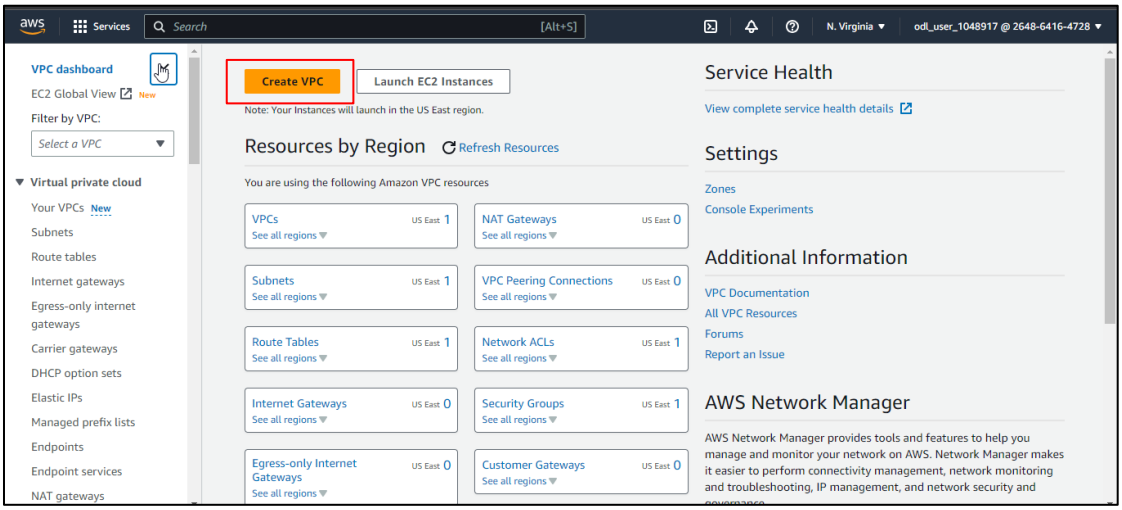
1. Create a custom VPC and enable DNS hostname
2. Create an internet gateway and attach it to the VPC
3. Create a subnet and a route table
4. Configure a route table
5. Launch the EC2 instance

Step 1: Create a custom VPC and enable DNS hostname

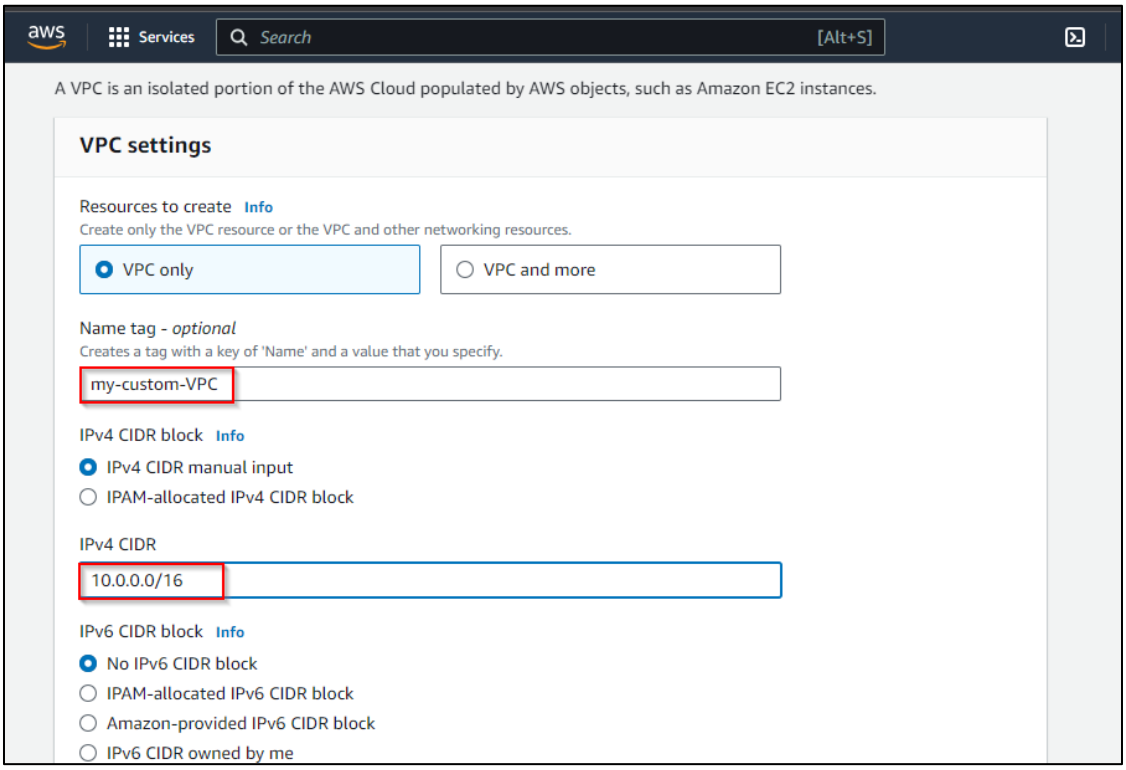
1.1 Open the AWS Management Console and search for **VPC**



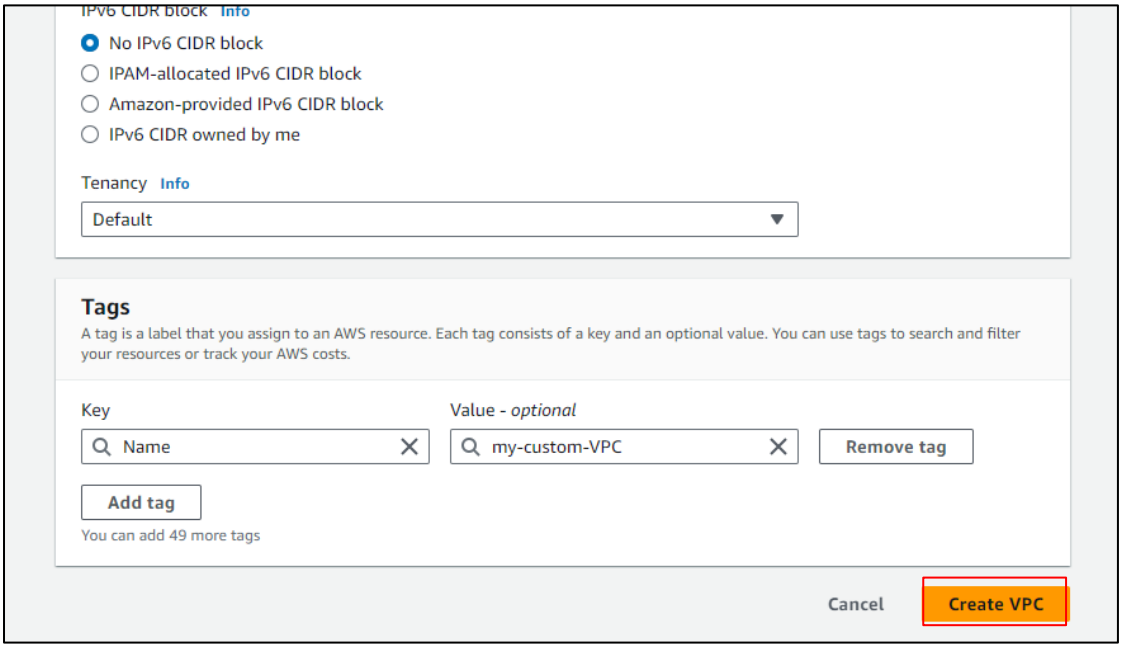
1.2 Click on **Create VPC**

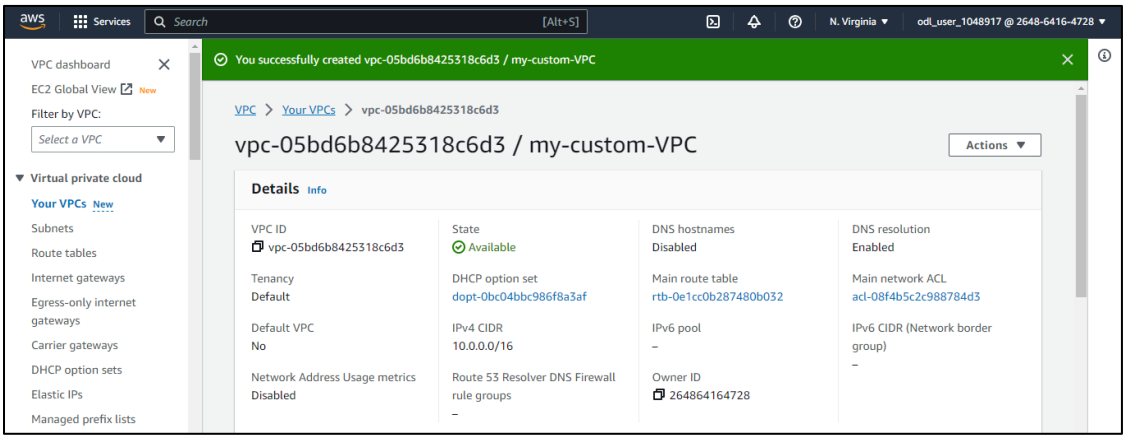


1.3 Enter **my-custom-VPC** as the VPC Name and **10.0.0.0/16** as the **IPv4 CIDR**



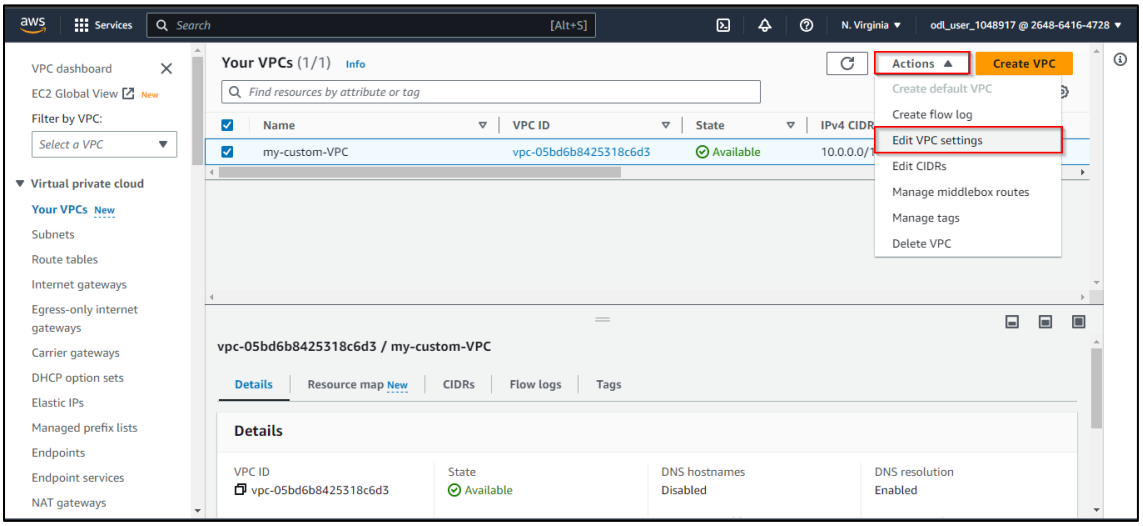
1.4 Click on **Create VPC**



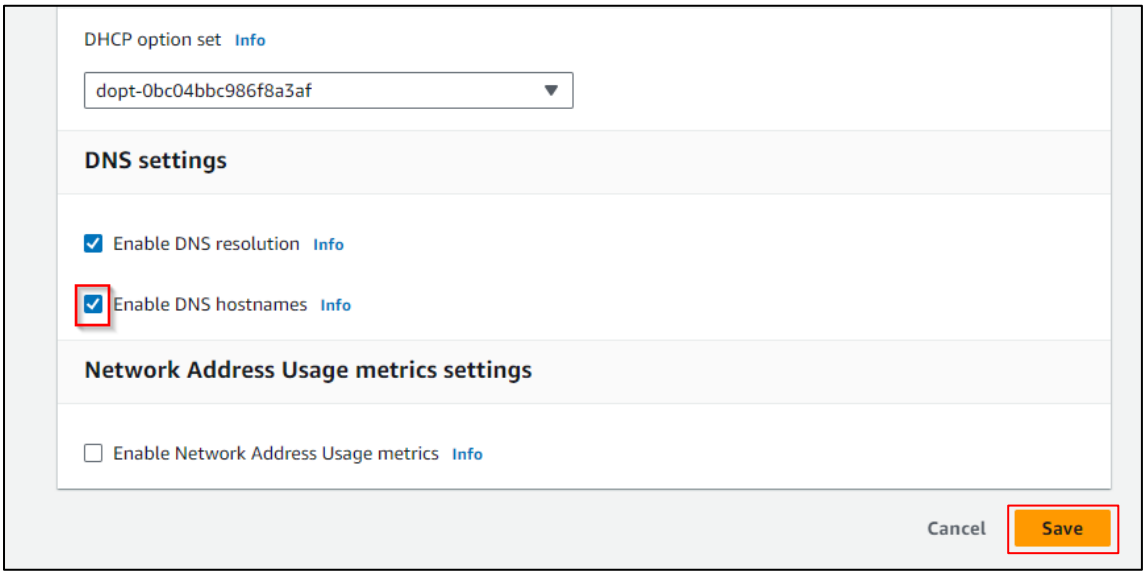


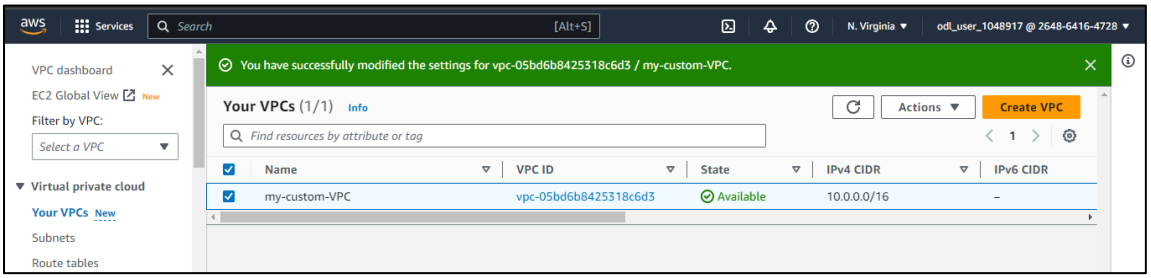
The VPC has been created successfully.

1.5 Select the created VPC and click on **Edit VPC settings** under the **Actions** tab



1.6 Select **Enable DNS hostnames** under **DNS settings**, and then click on **Save**

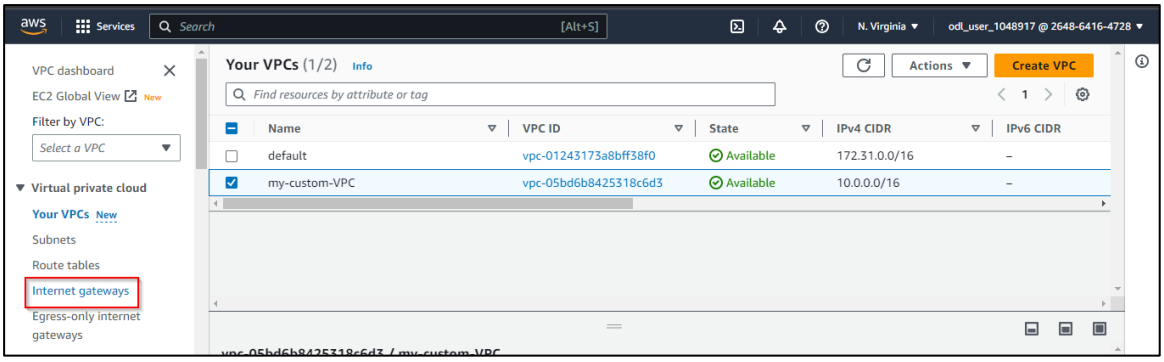




DNS has been successfully enabled.

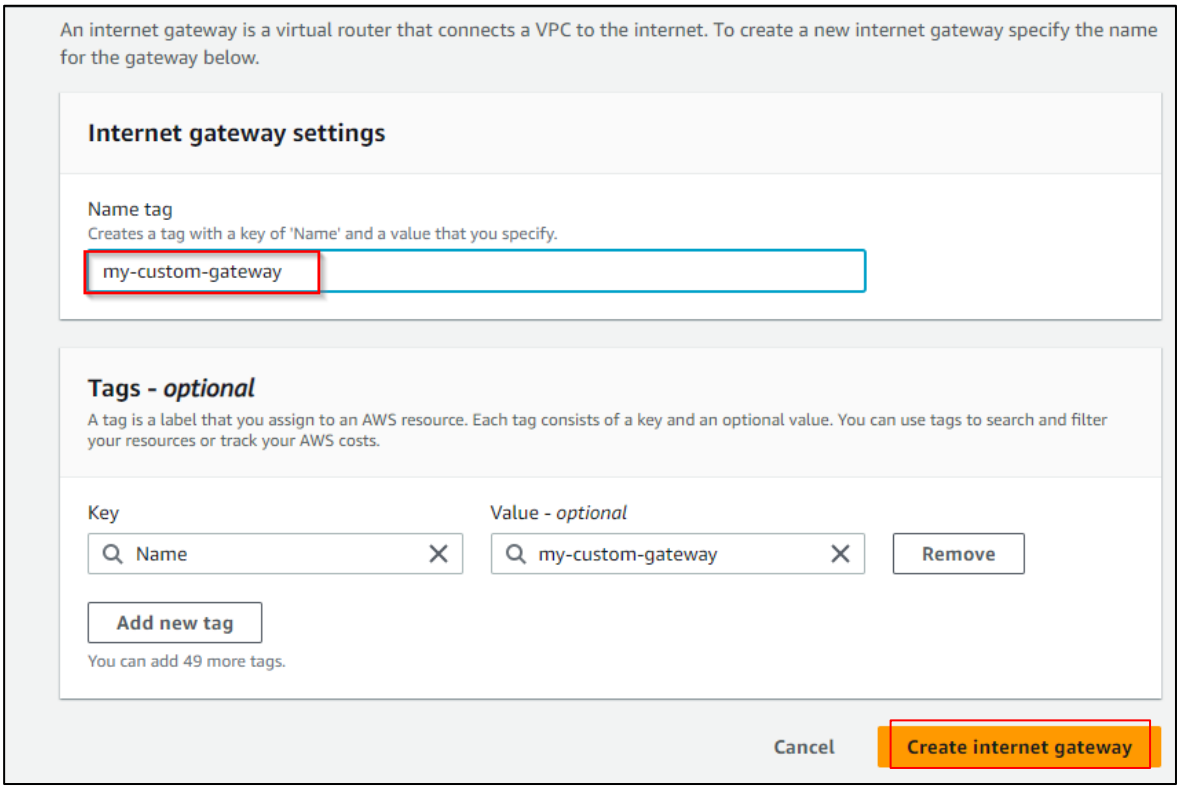
Step 2: Create an internet gateway and attach it to the VPC

2.1 Navigate to the **Internet gateways** page

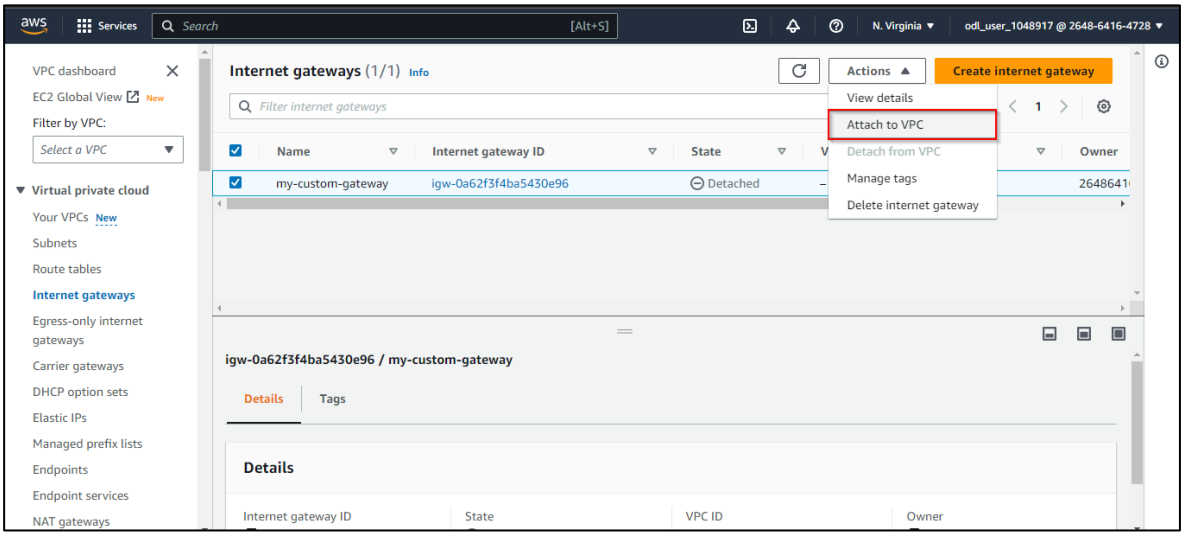


2.2 Click on **Create internet gateway**

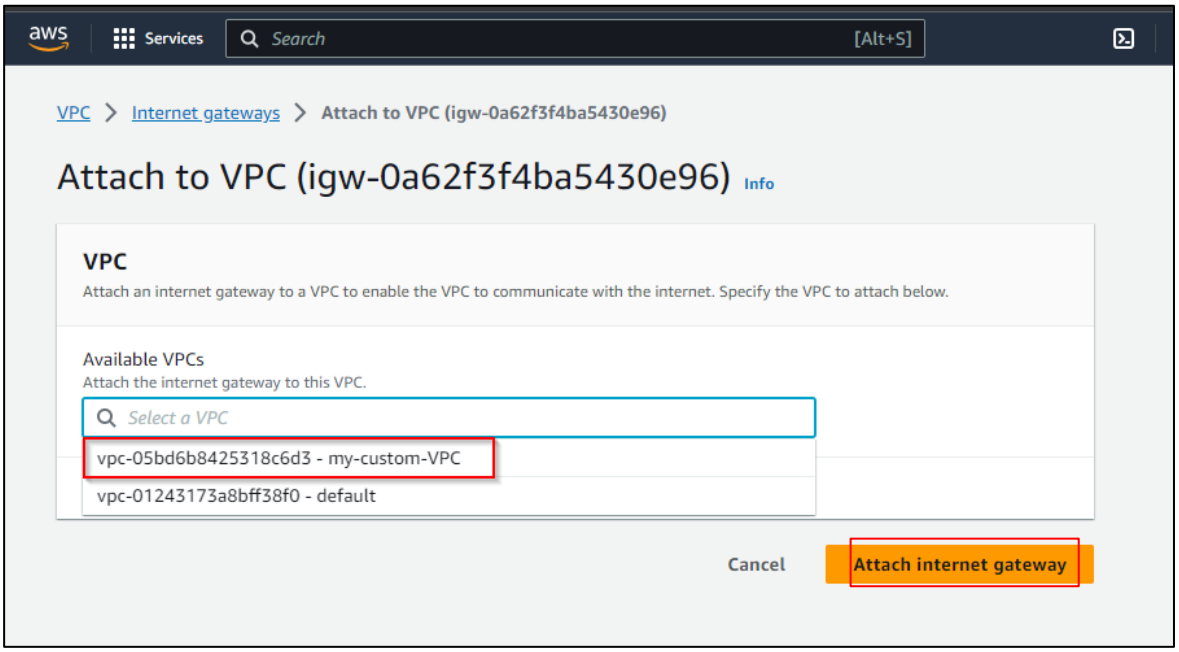
2.3 Enter the name **my-custom-gateway** and click on **Create internet gateway**

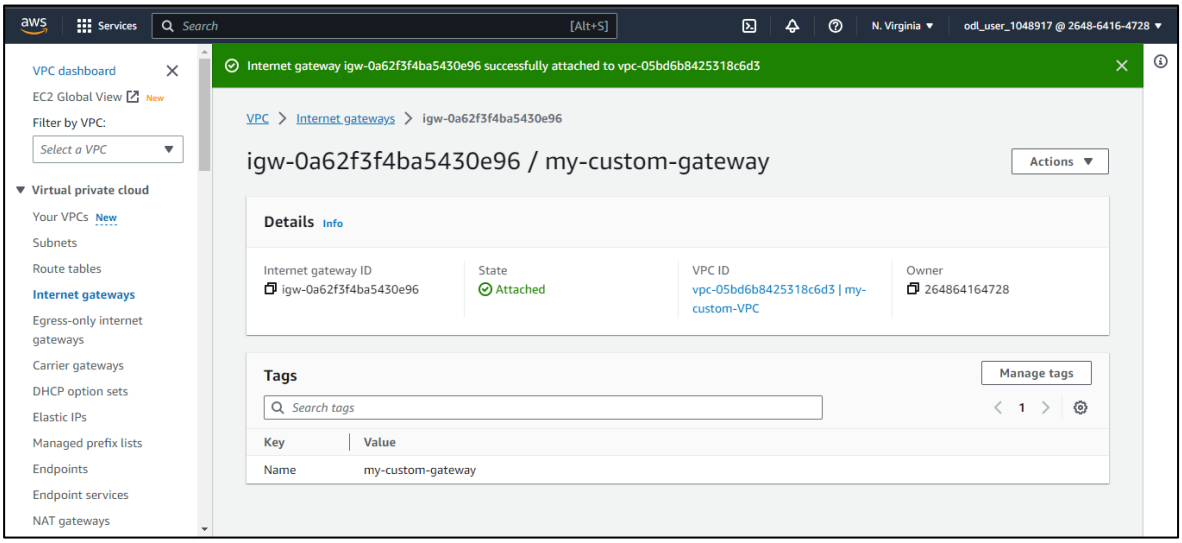


2.4 Click on **Attach to VPC** under the **Actions** tab



2.5 Select **my-custom-VPC** and click on **Attach internet gateway**

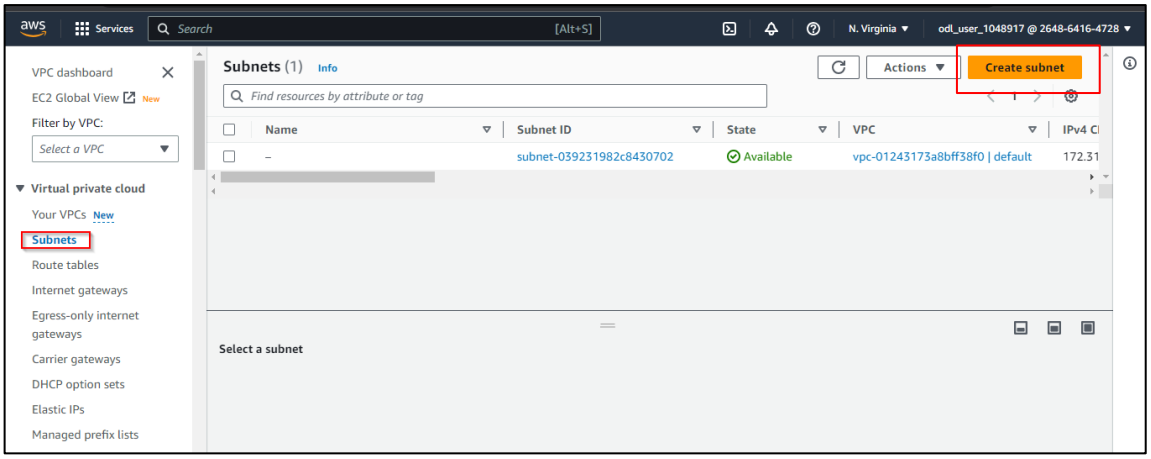




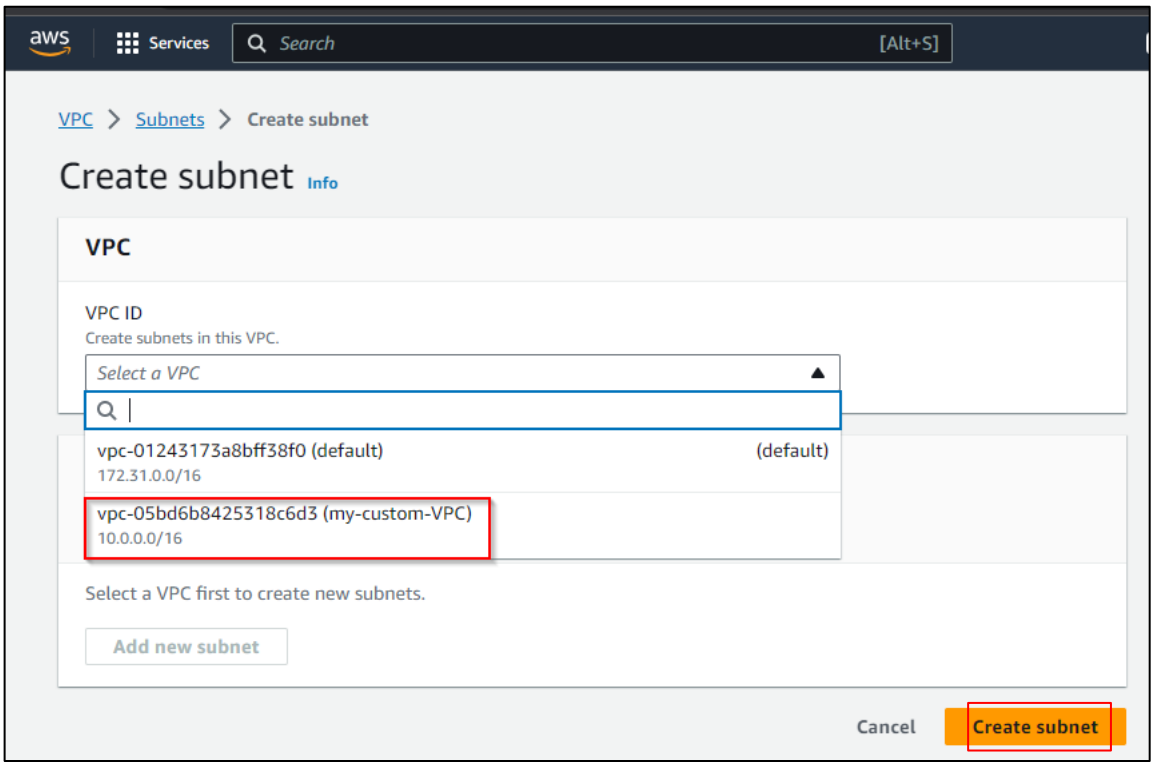
The internet gateway has been attached successfully.

Step 3: Create a subnet and a route table

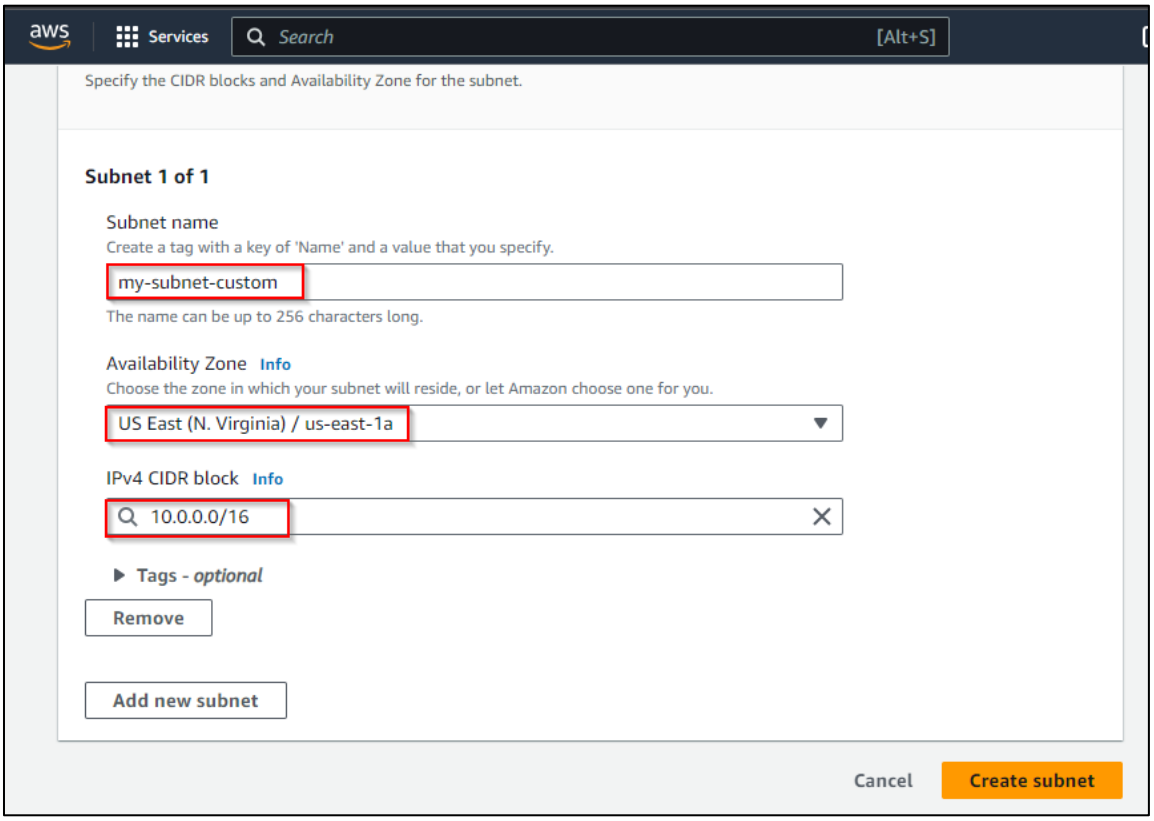
3.1 Navigate to **Subnets** and click on **Create Subnet**

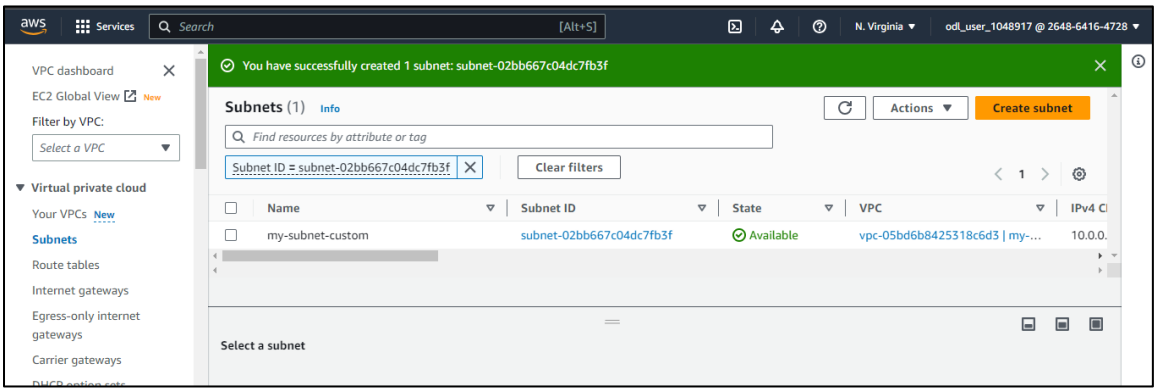


3.2 Choose **my-custom-VPC** and click on **Create subnet**



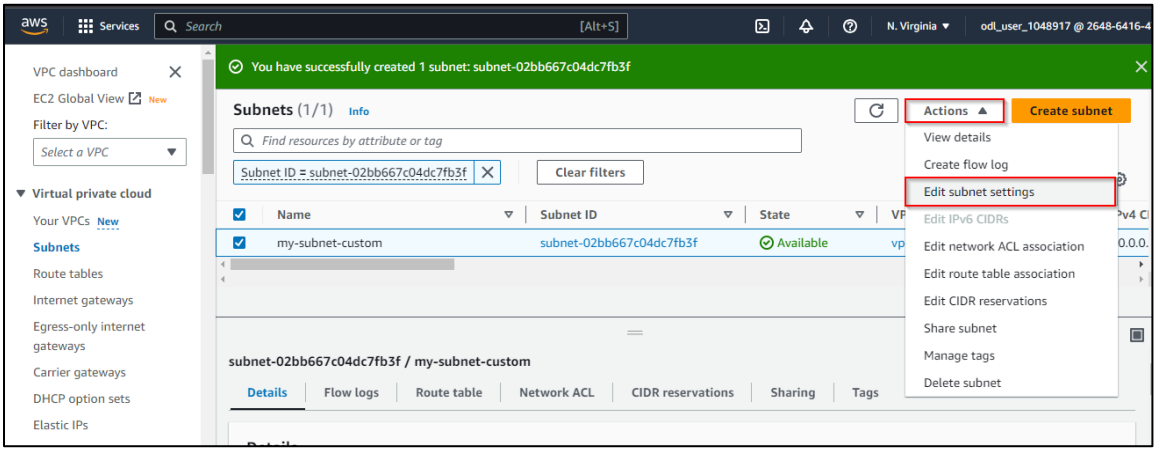
3.3 Enter **my-subnet-custom** as the **Subnet name**, choose the **Availability Zone**, set **10.0.0.0/16** as **the IPv4 CIDR block**, and click on **Create subnet**



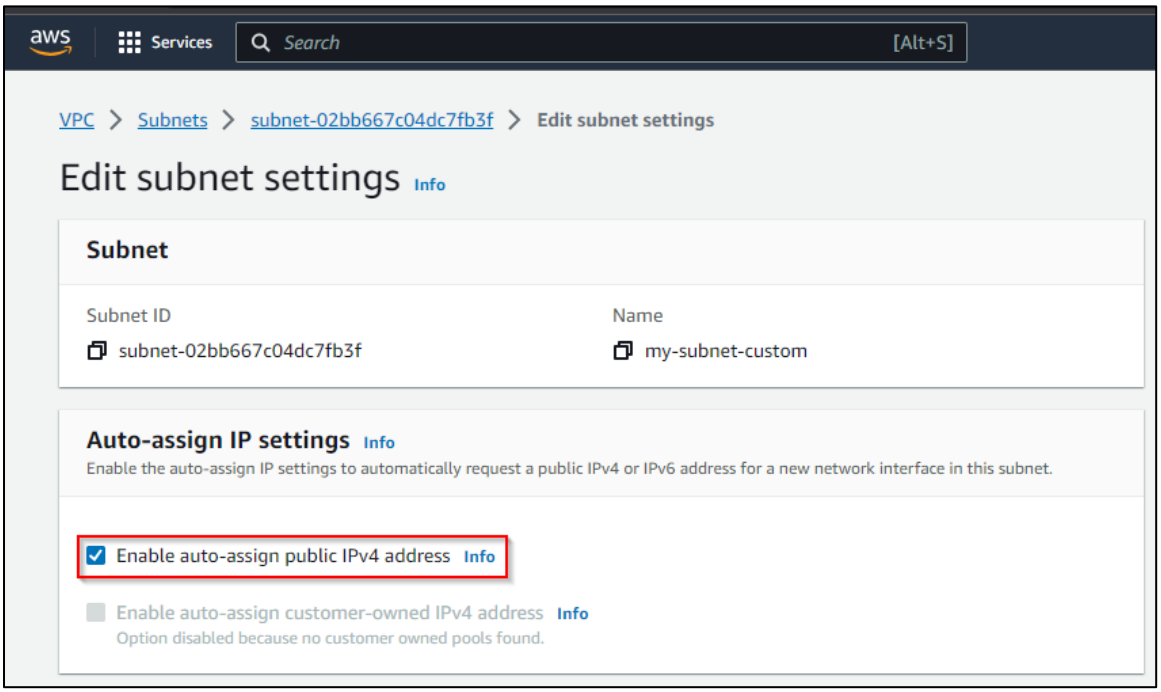


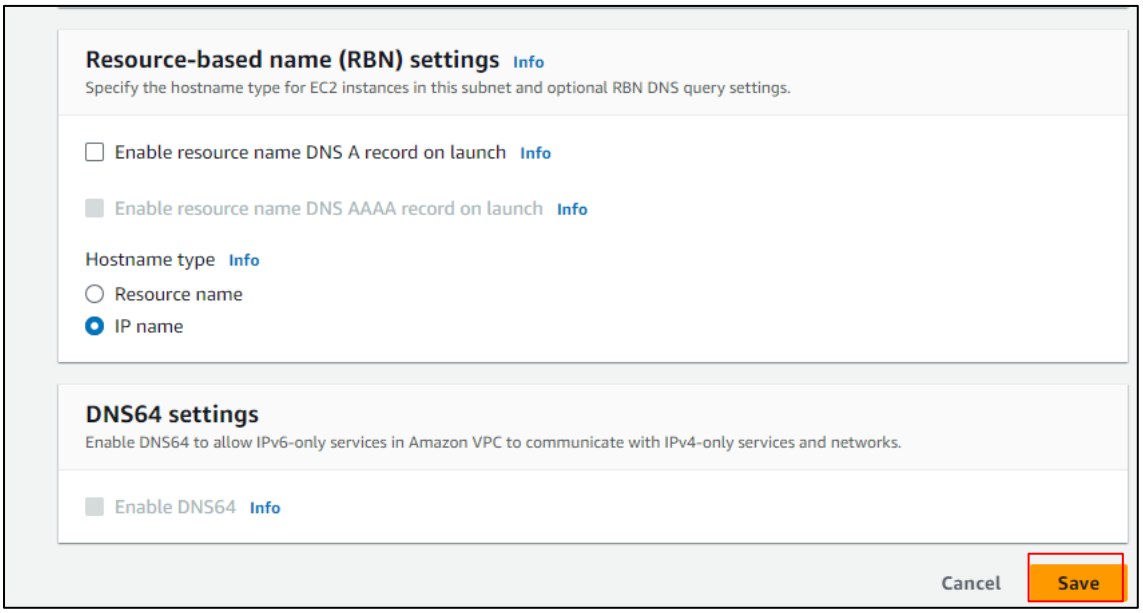
The subnet has been created successfully.

3.4 Select the subnet, click on **Actions**, and choose **Edit subnet settings**

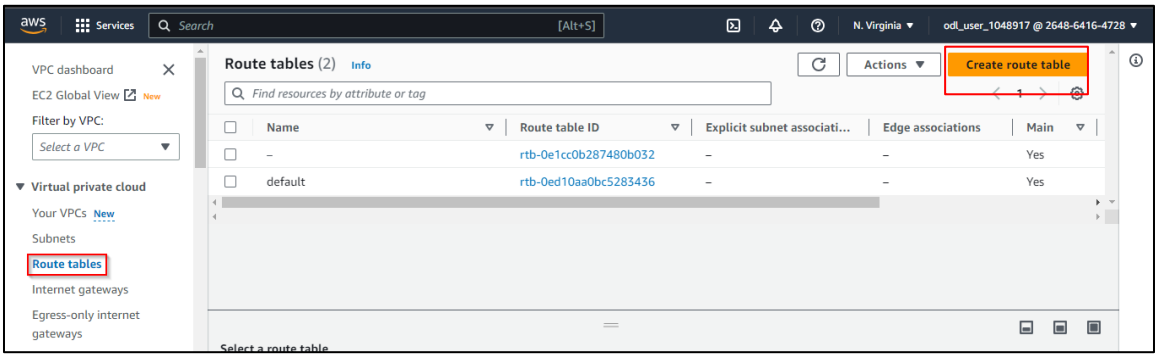


3.5 Select **the Enable auto-assign public IPv4 addresses** option and click **Save**

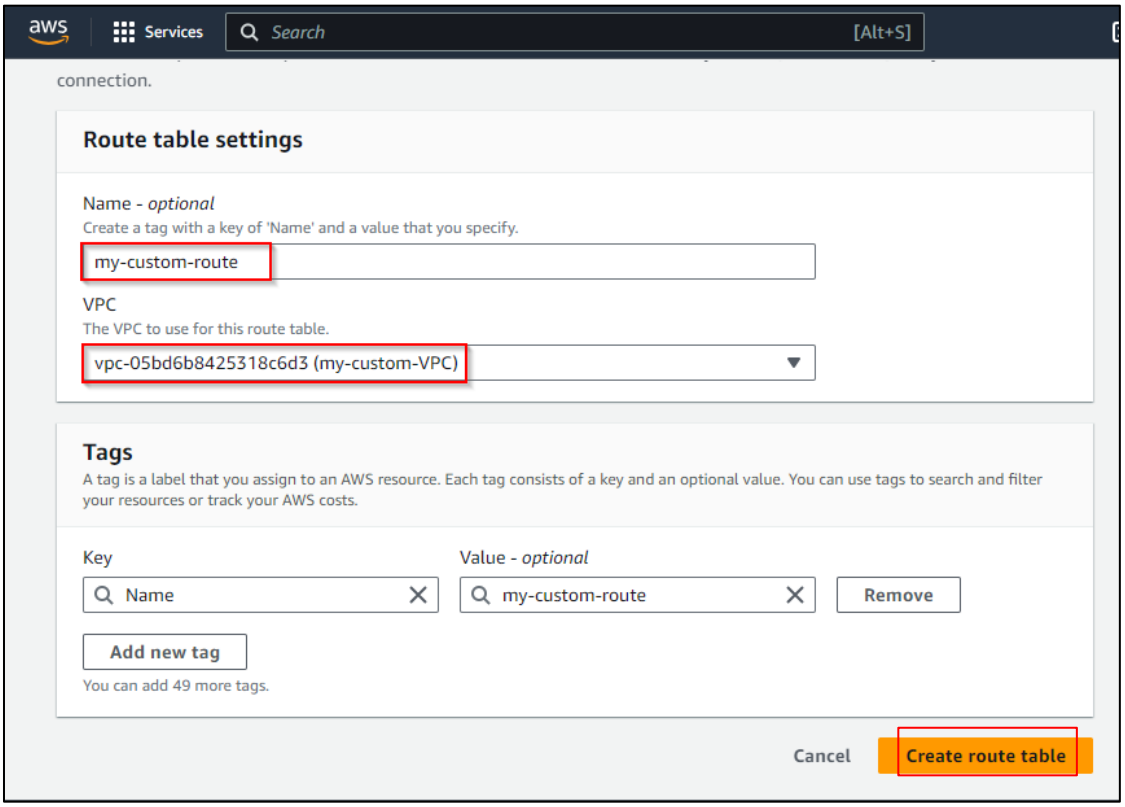




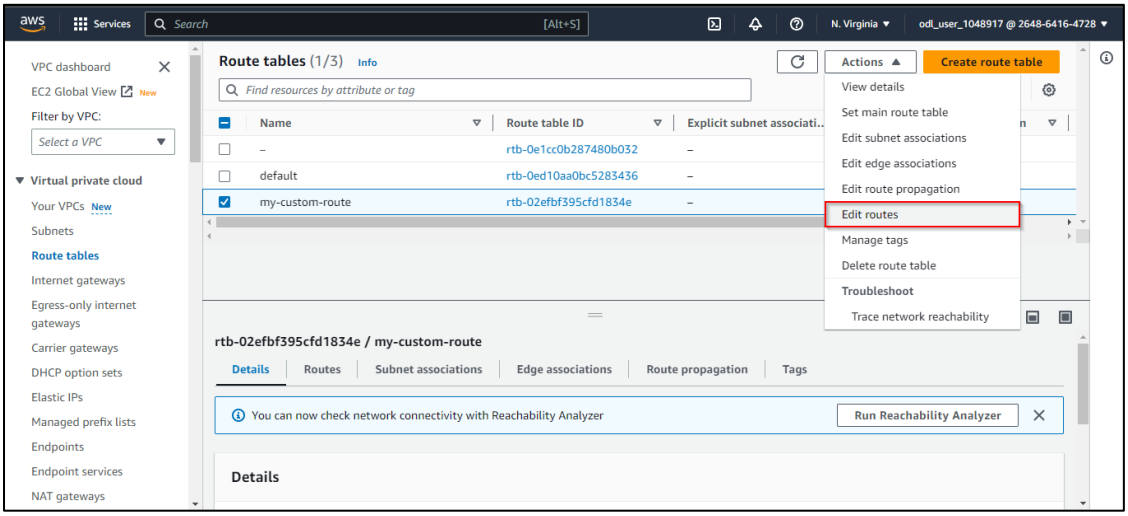
3.6 Navigate to the **Route tables** page and click on **Create route table**



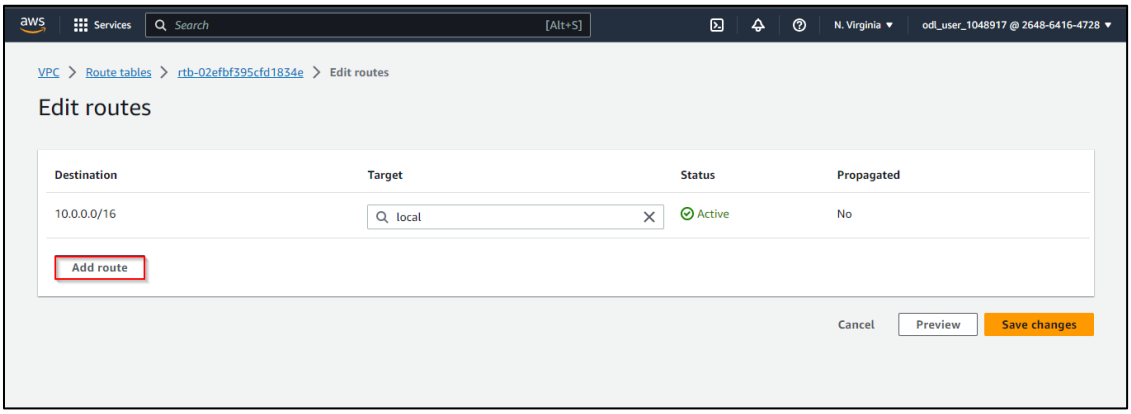
3.7 Enter **my-custom-route** as the **Name**, select **my-custom-VPC** in the VPC field, and click one **Create the route table**



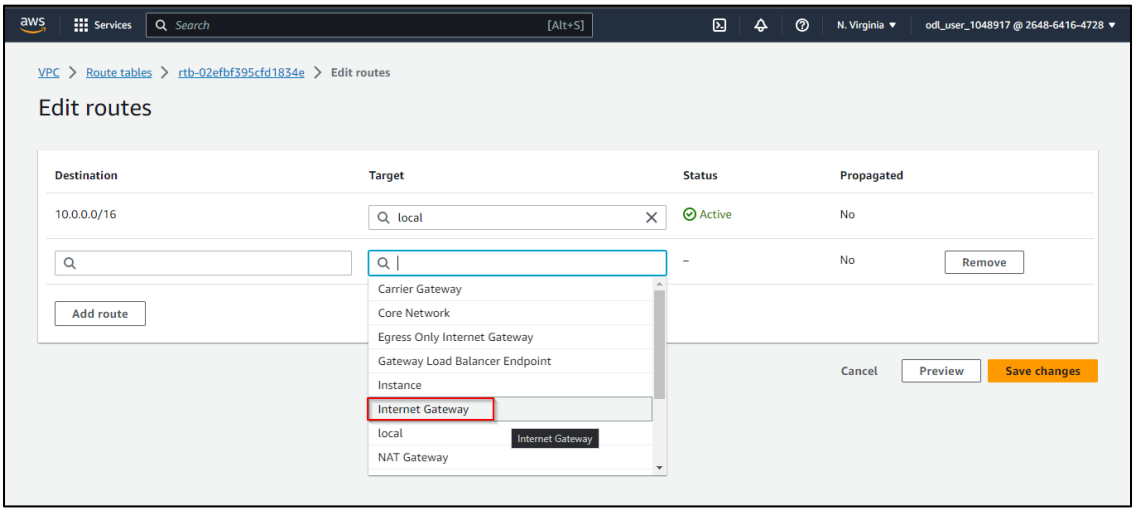
3.8 Select the route table, click on **Actions**, and choose **Edit routes**



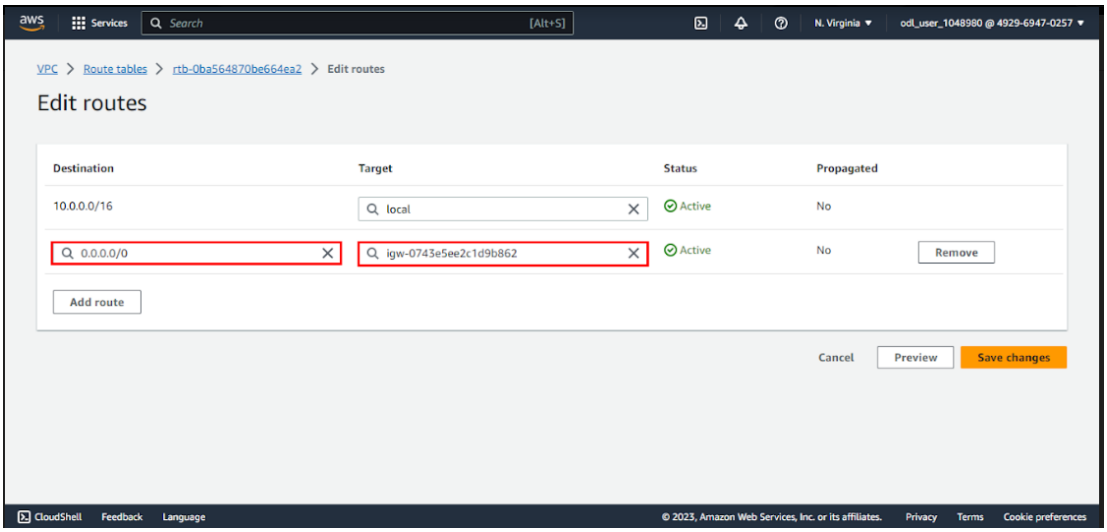
3.9 Click on **Add route**

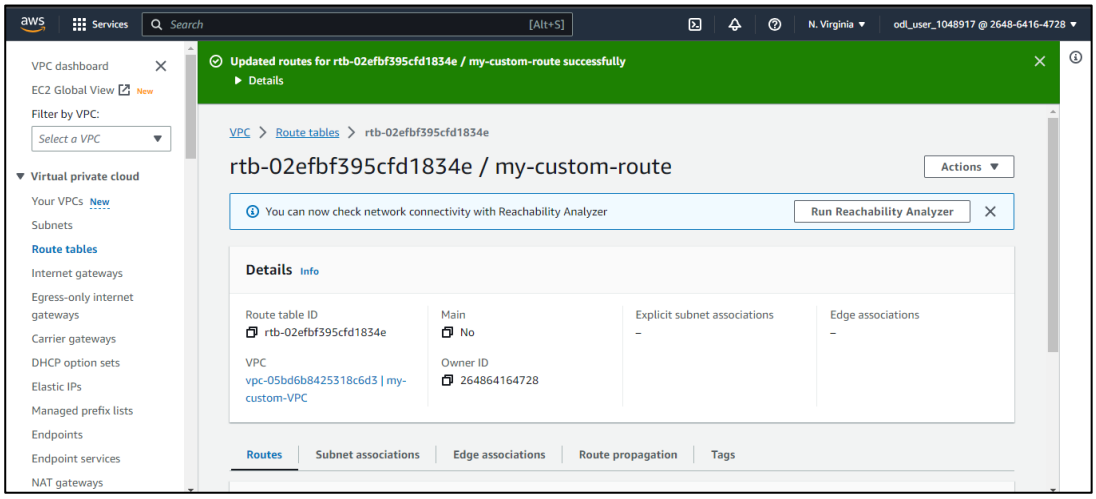


3.10 Select **Internet Gateway** in the **Target** field



3.11 Click on **Save changes**

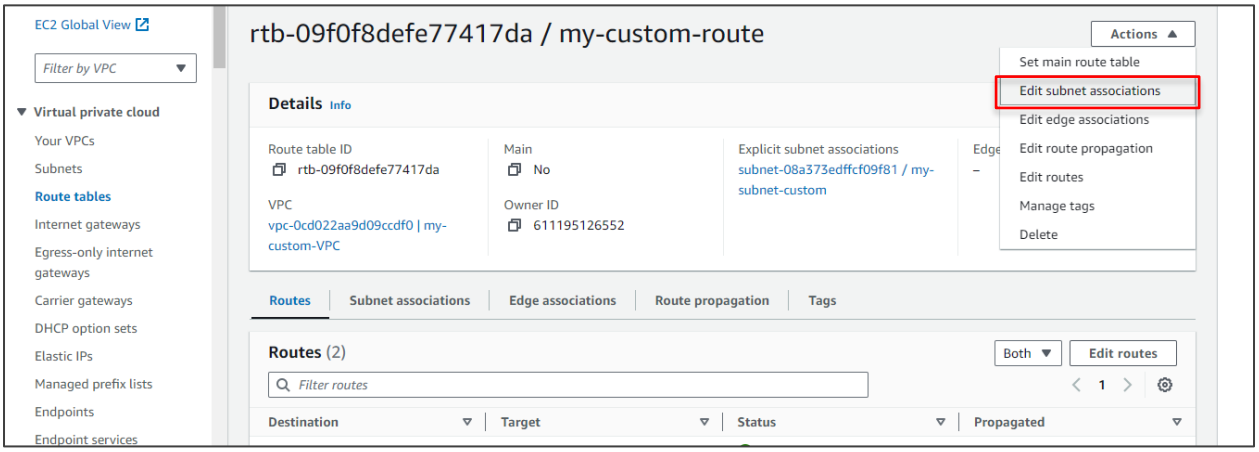




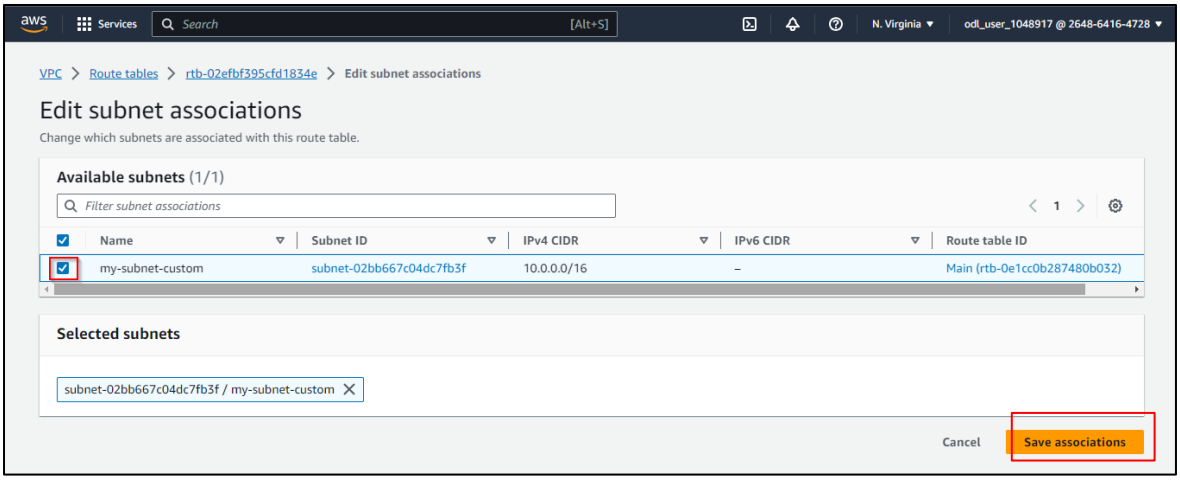
The edit route table has been successfully created.

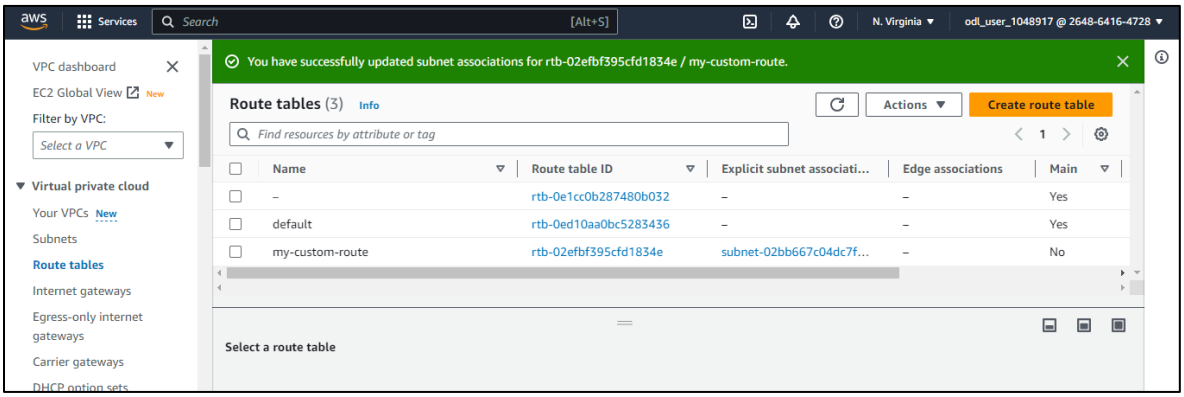
Step 4: Configure a route table

4.1 In the **Route tables** dashboard, click on **Edit subnet ociations** under **Actions**



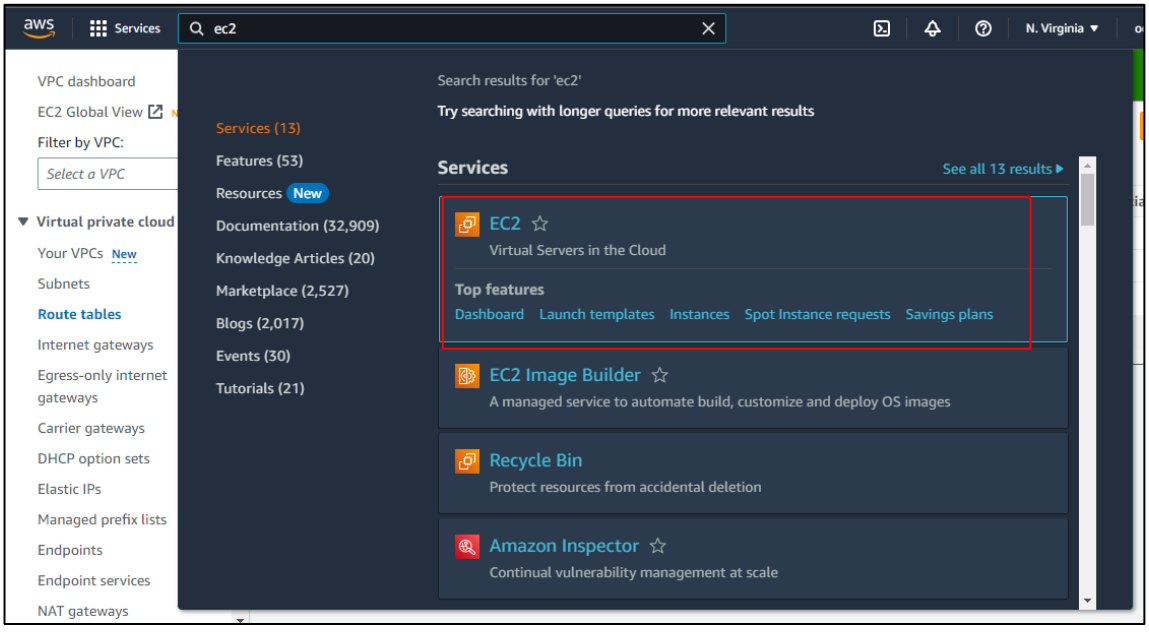
4.2 Select **my-subnet-custom** and click on **Save associations**



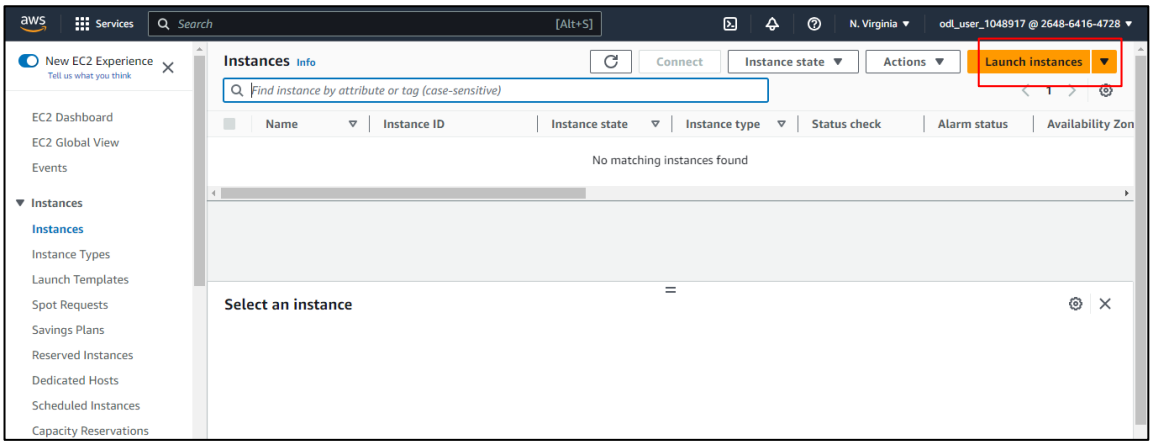


Step 5: Launch the EC2 instance

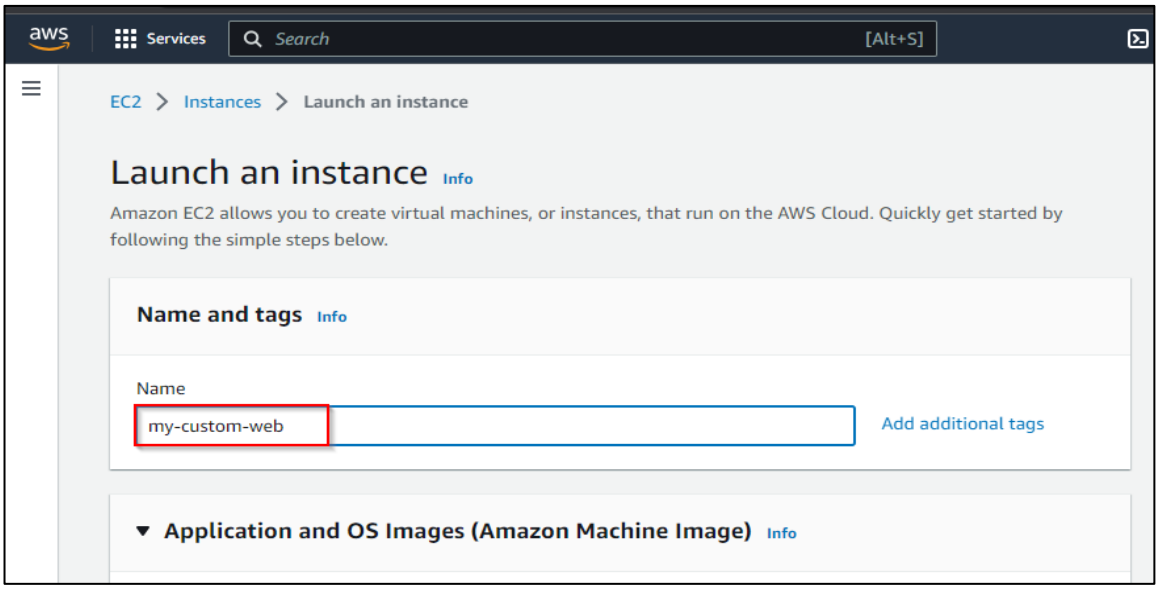
5.1 Navigate back to the EC2 console



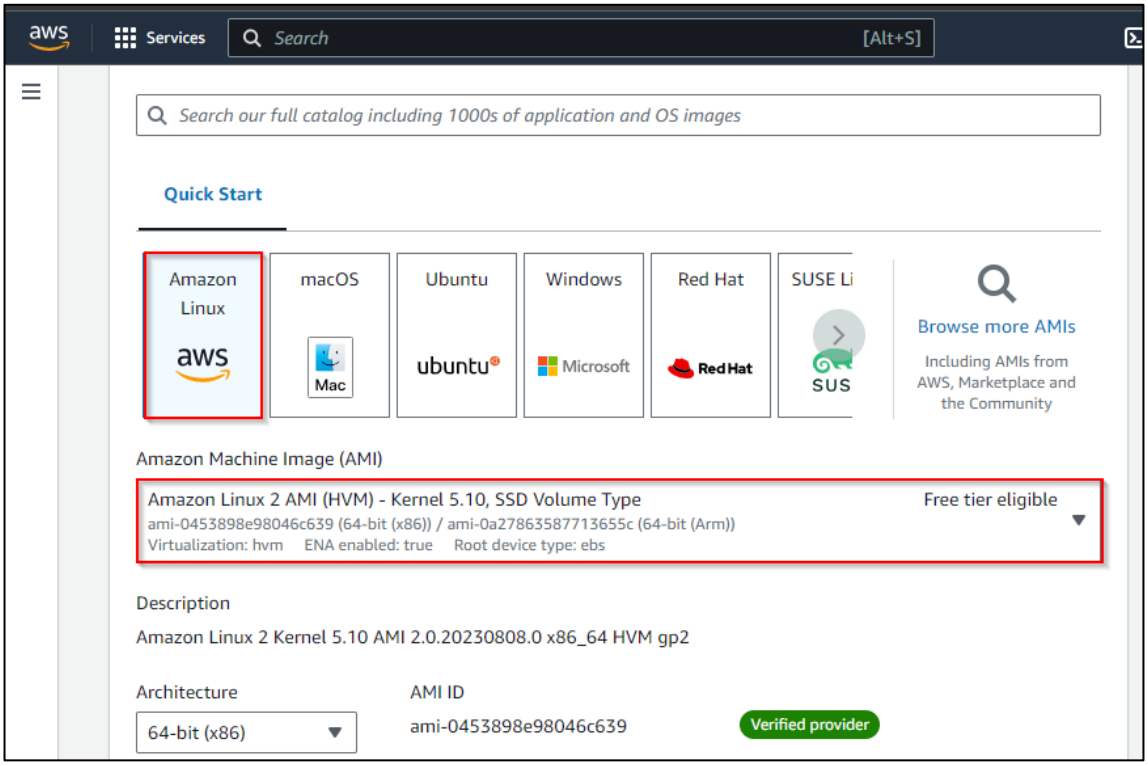
5.2 Click on **Launch instances**



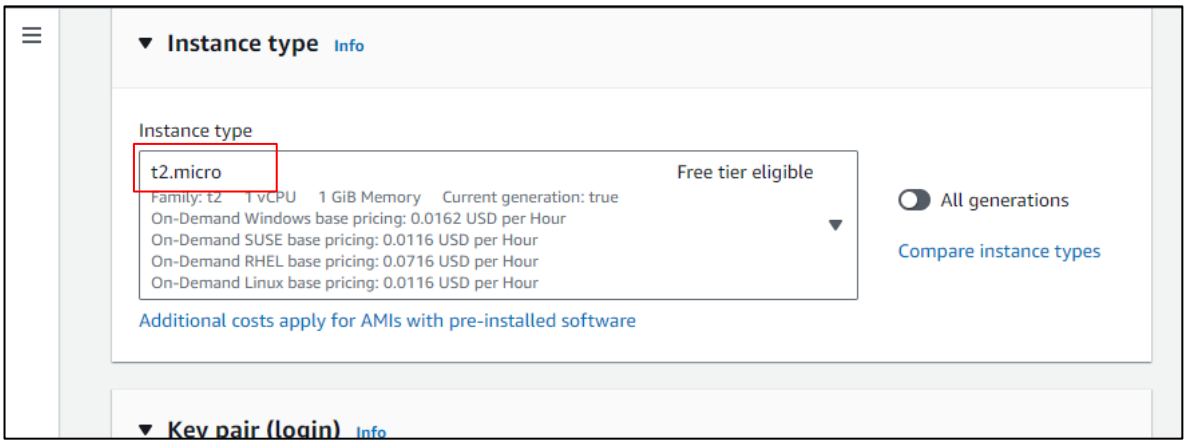
5.3 Enter the name **my-custom-web** in the field



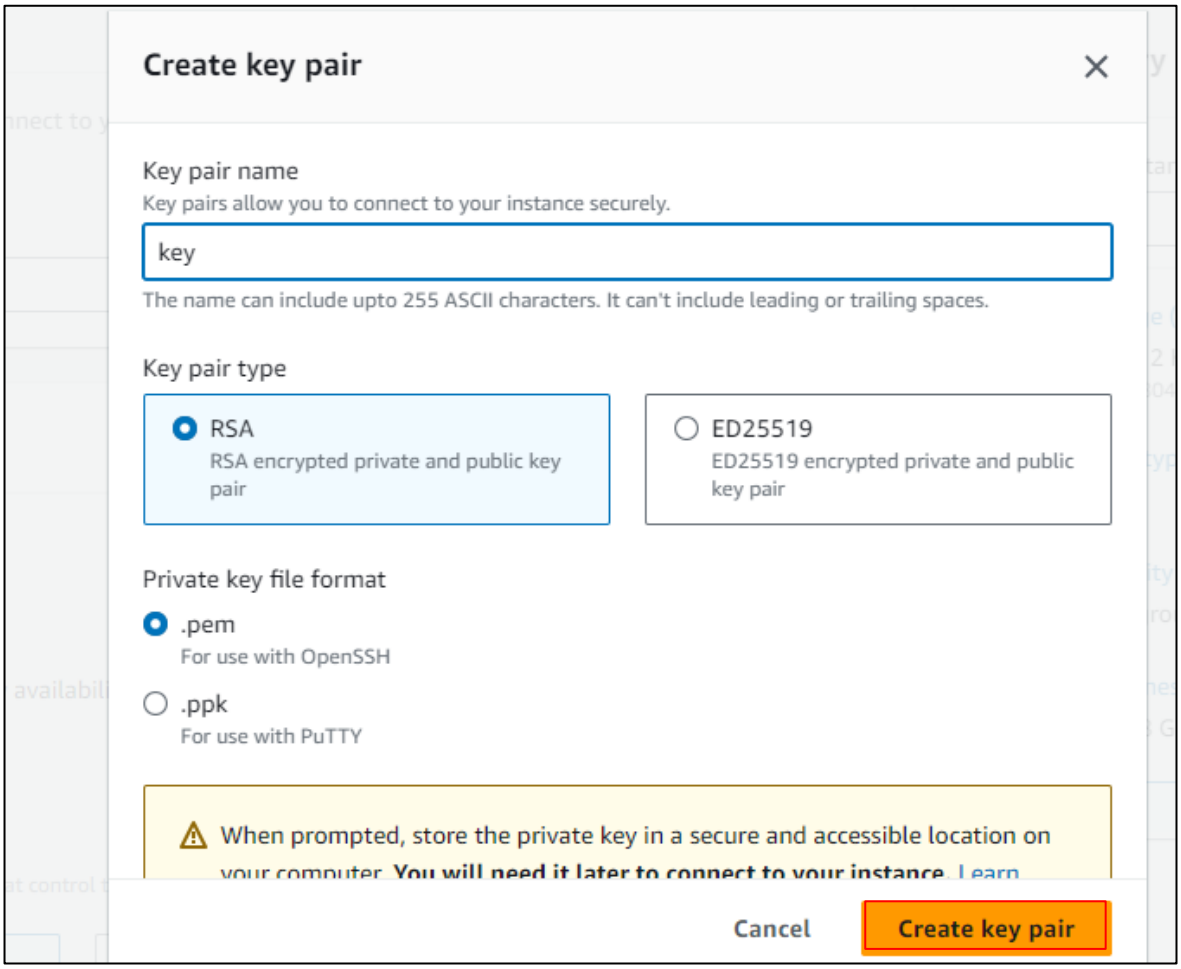
5.4 Click on the **Amazon Linux** option and select the **SSD Volume Type** as **AMI**



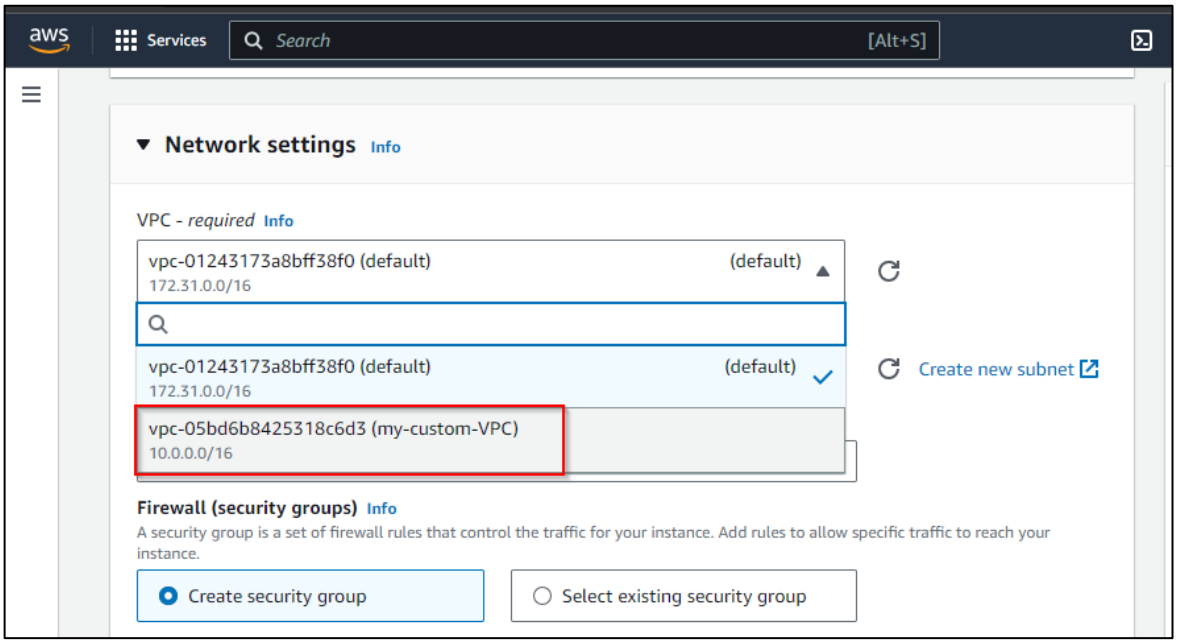
5.5 Select the instance type as **t2.micro**



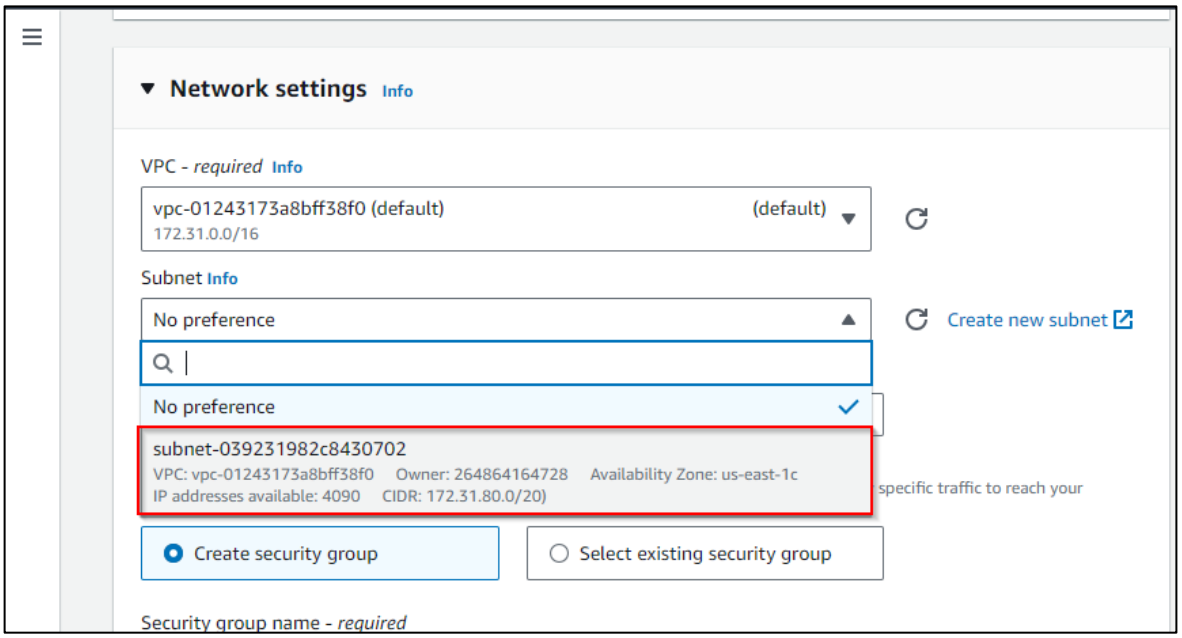
5.6 Enter the **Key pair name** and click on the **Create key pair** button



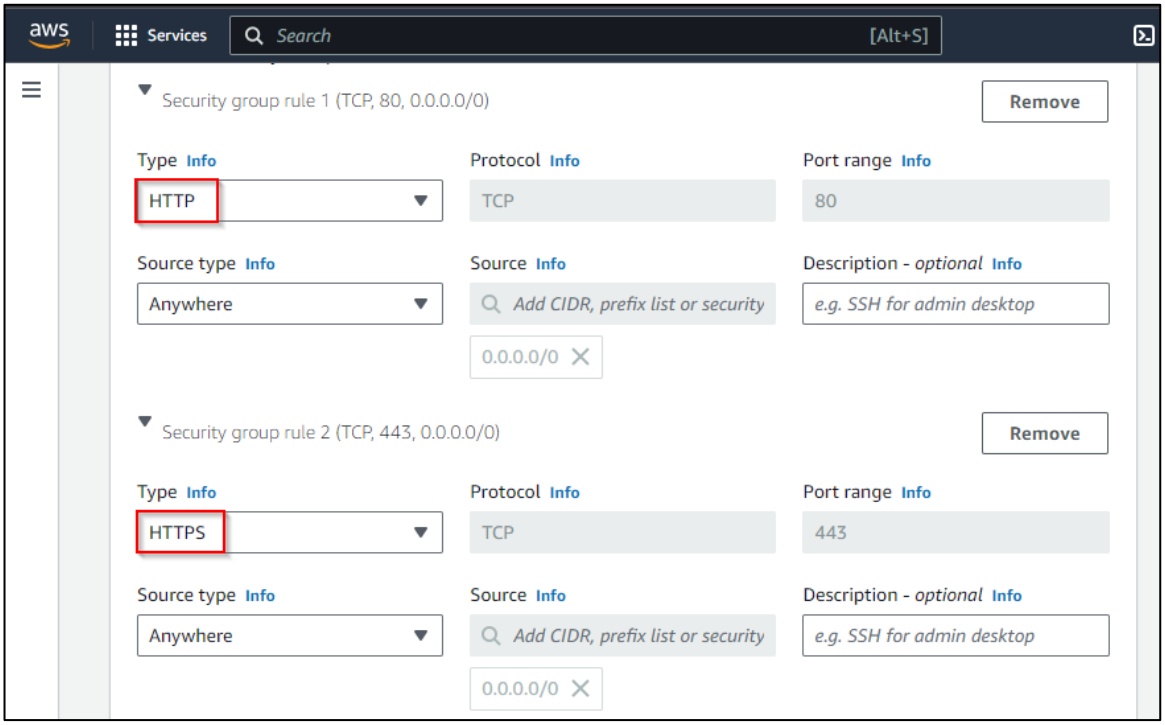
5.7 Configure **Network settings** by selecting your **VPC** and an availability zone



5.8 Select the **availability zone** in the **Subnet**

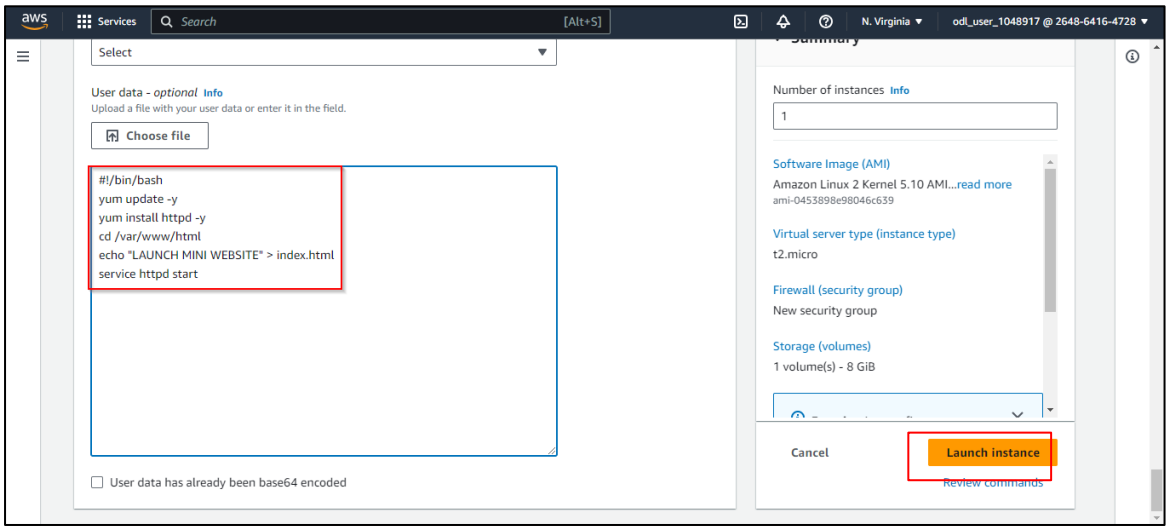


5.9 Add inbound rules for **HTTP** and **HTTPS**

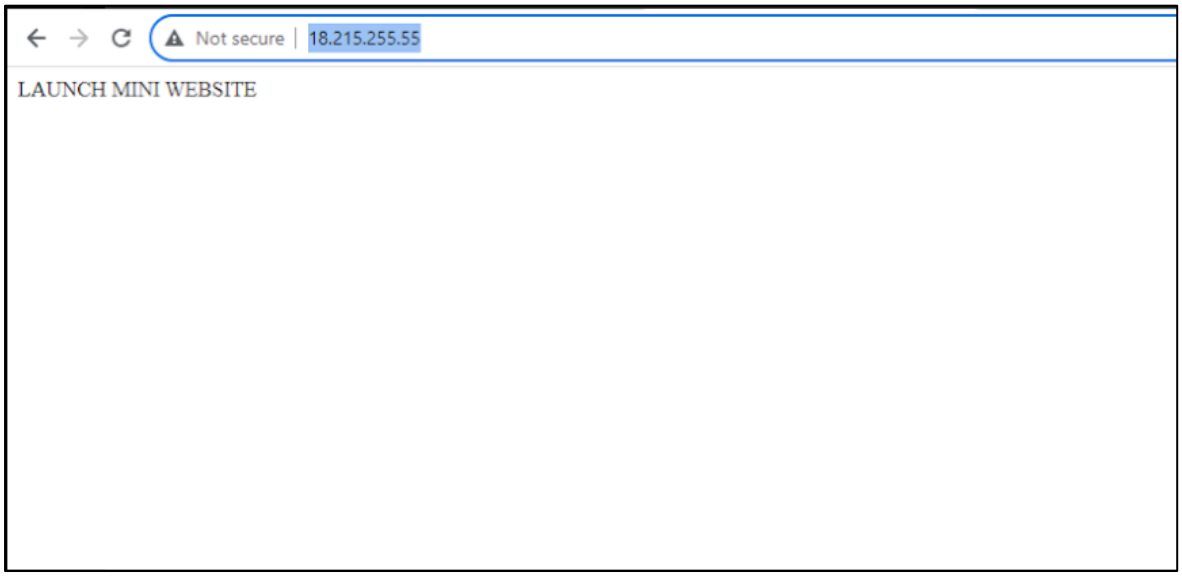


5.10 Click on **Advanced details,** enter the code in **User data,** and click on **Launch instance**

#!/bin/bash
yum update -y
yum install httpd -y
cd /var/www/html
echo "LAUNCH MINI WEBSITE" > index.html
service httpd start



5.11 After the instance is running, copy the IPv4 address and paste it into a browser to view the mini website



By following these steps, you have successfully customized an EC2 instance for web deployment within an Amazon Web Services (AWS) environment.