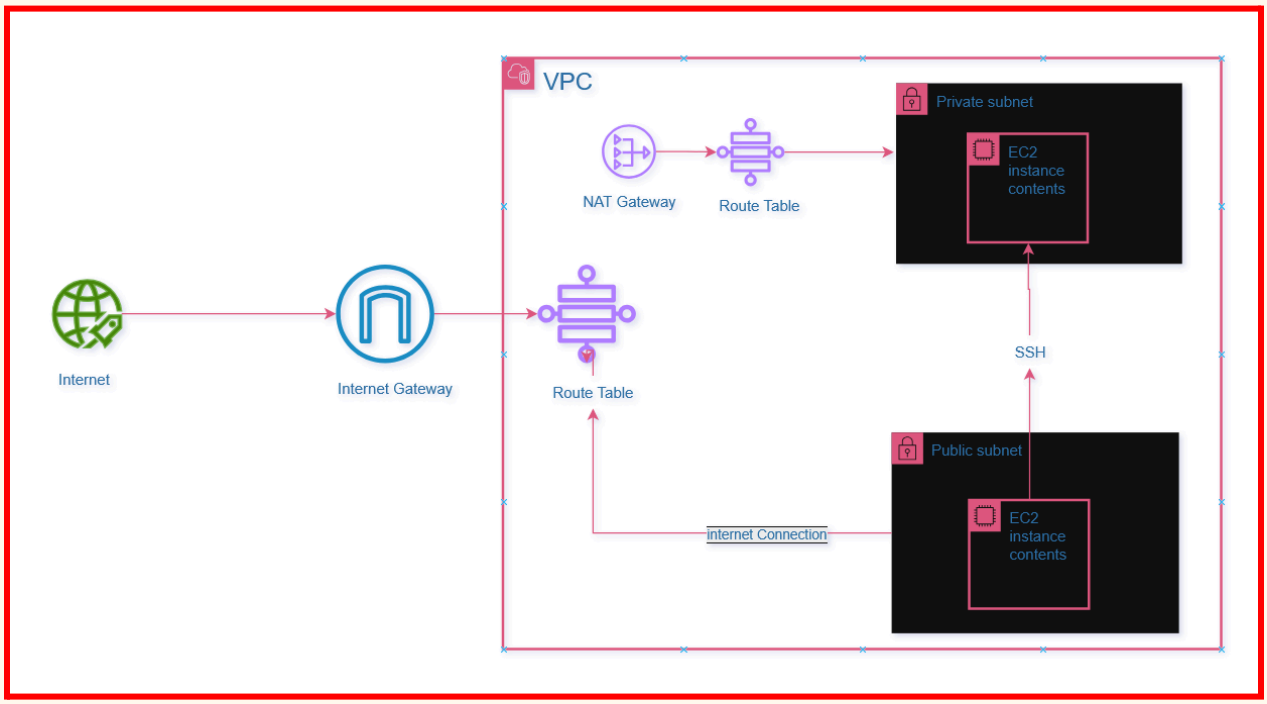


AWS VPC Networking with Public and Private Subnets

31-August

—

By:

Lalit Kumar
https://bento.me/lalit192977

Overview

In this project, I designed and implemented a **custom Virtual Private Cloud (VPC)** in AWS to demonstrate a secure and scalable networking architecture. The setup includes both **public and private subnets**, each hosting an EC2 instance, and proper routing to enable internet access where required.

- ## Key Highlights:

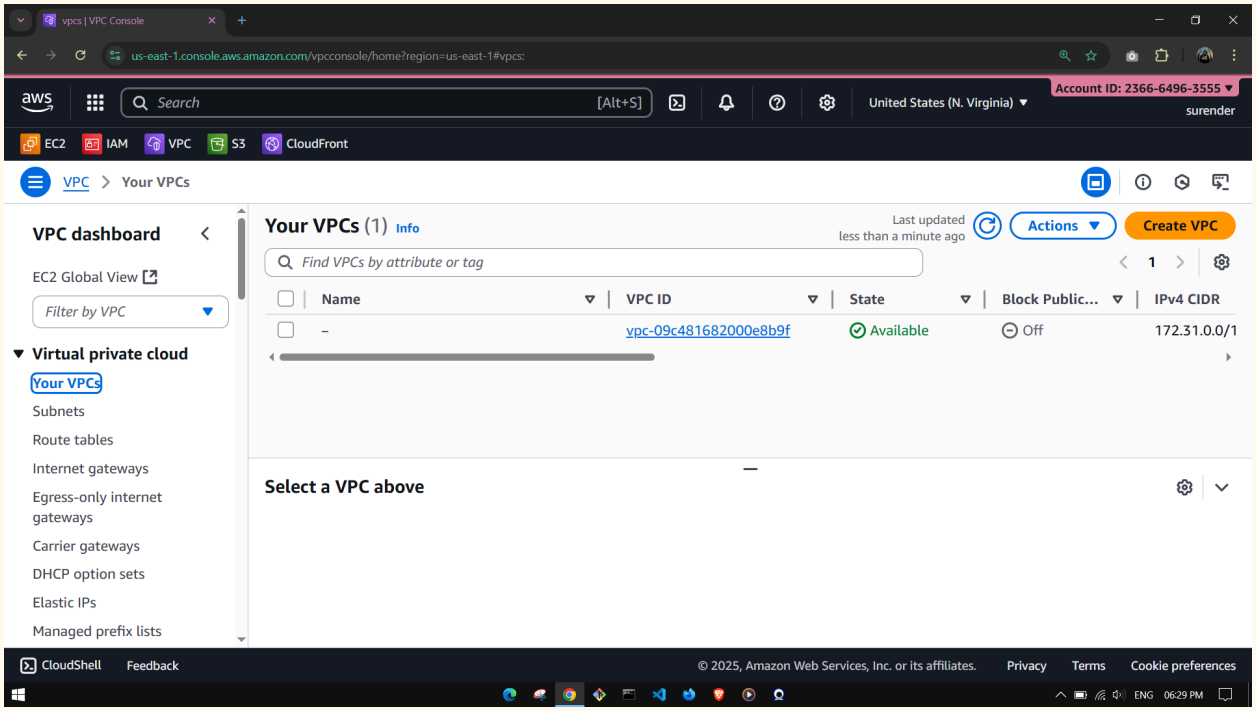
1. **Custom VPC Creation** – A dedicated VPC was created to logically isolate resources within AWS.
2. **Subnet Design** – Two subnets were deployed:
    - **Public Subnet**: Connected to the Internet Gateway for direct internet access.
    - **Private Subnet**: Isolated from direct internet exposure, but enabled outbound internet via a NAT Gateway.
3. **Routing Configuration** –
    - Public subnet route table mapped to the Internet Gateway.
    - Private subnet route table mapped to the NAT Gateway.
4. **EC2 Instances** –
    - A **public EC2 instance** was launched in the public subnet to act as a **bastion/ jump server**.
    - A **private EC2 instance** was launched in the private subnet without direct internet exposure.
5. **Secure Connectivity** – Using the public instance as a bastion host, SSH access was established to the private instance. The private instance successfully accessed the internet through the NAT Gateway.

- ## Purpose of the Project:

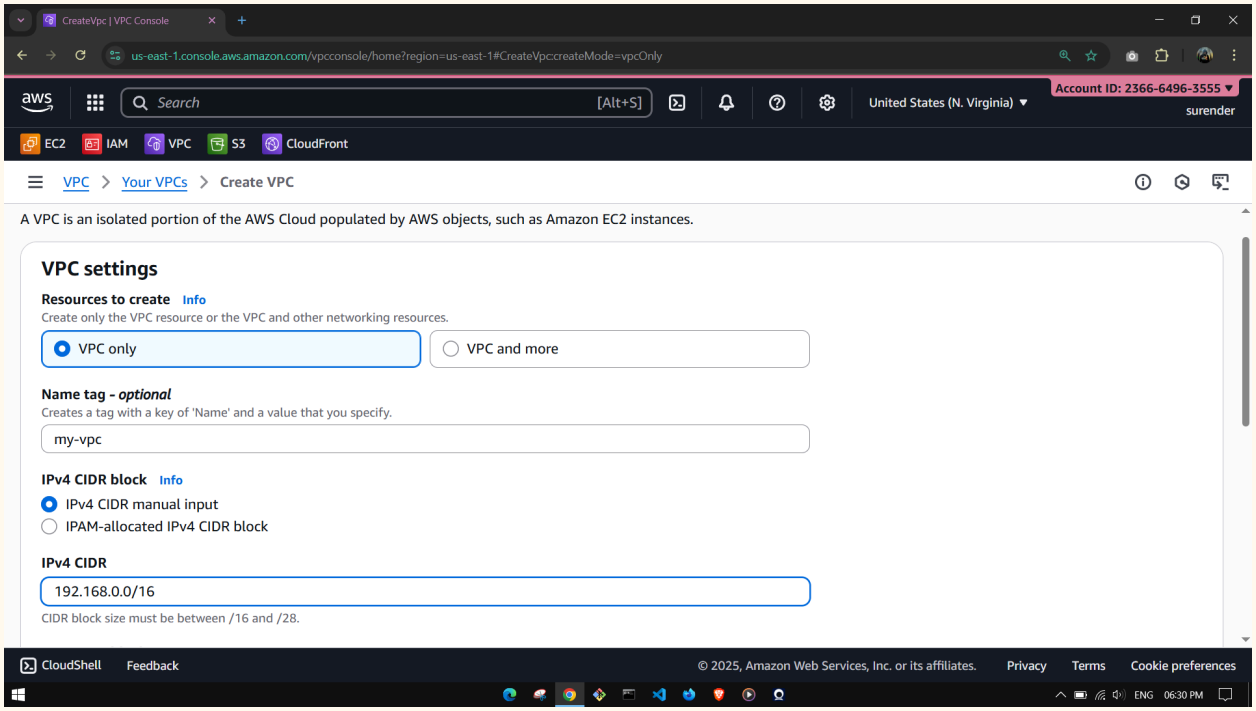
This project demonstrates how to build a **secure AWS network architecture** where sensitive workloads are placed in private subnets, and only controlled access is allowed through a bastion host. Such architectures are commonly used in **production environments** to ensure security, scalability, and controlled resource access.

Let's Start

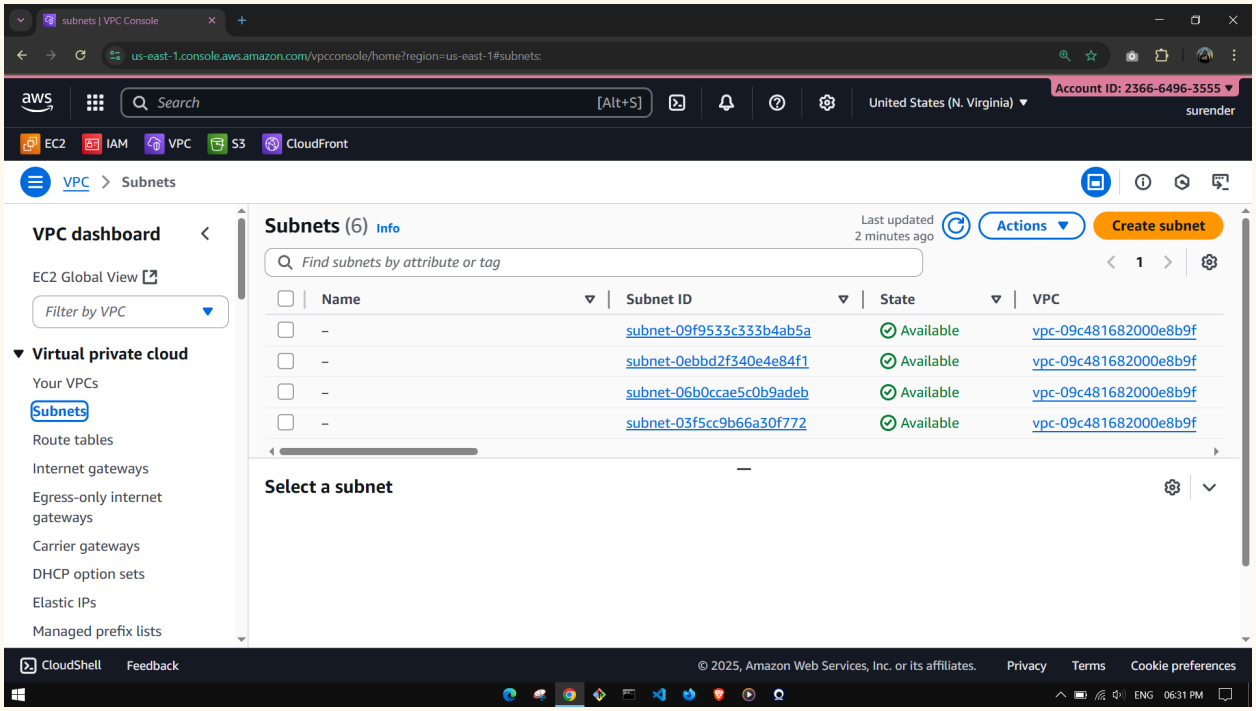
Go to vpc service and create a vpc from right top corner



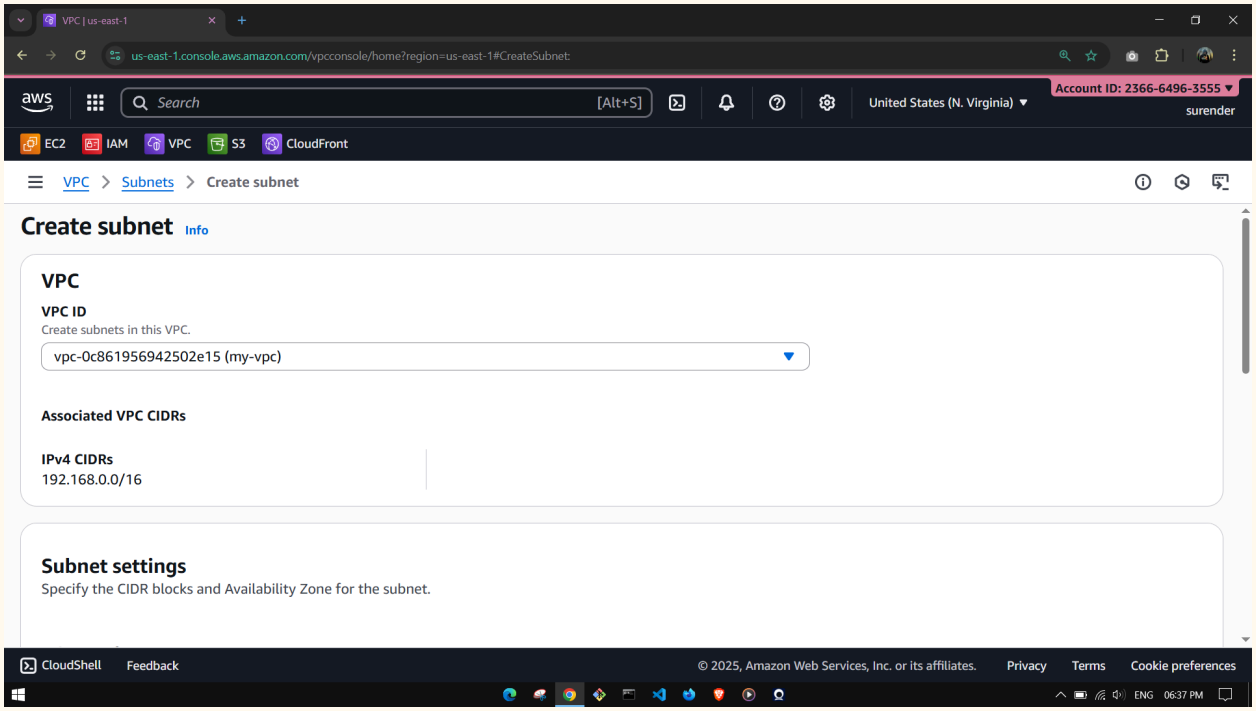
Now vpc name and CIDR range/block



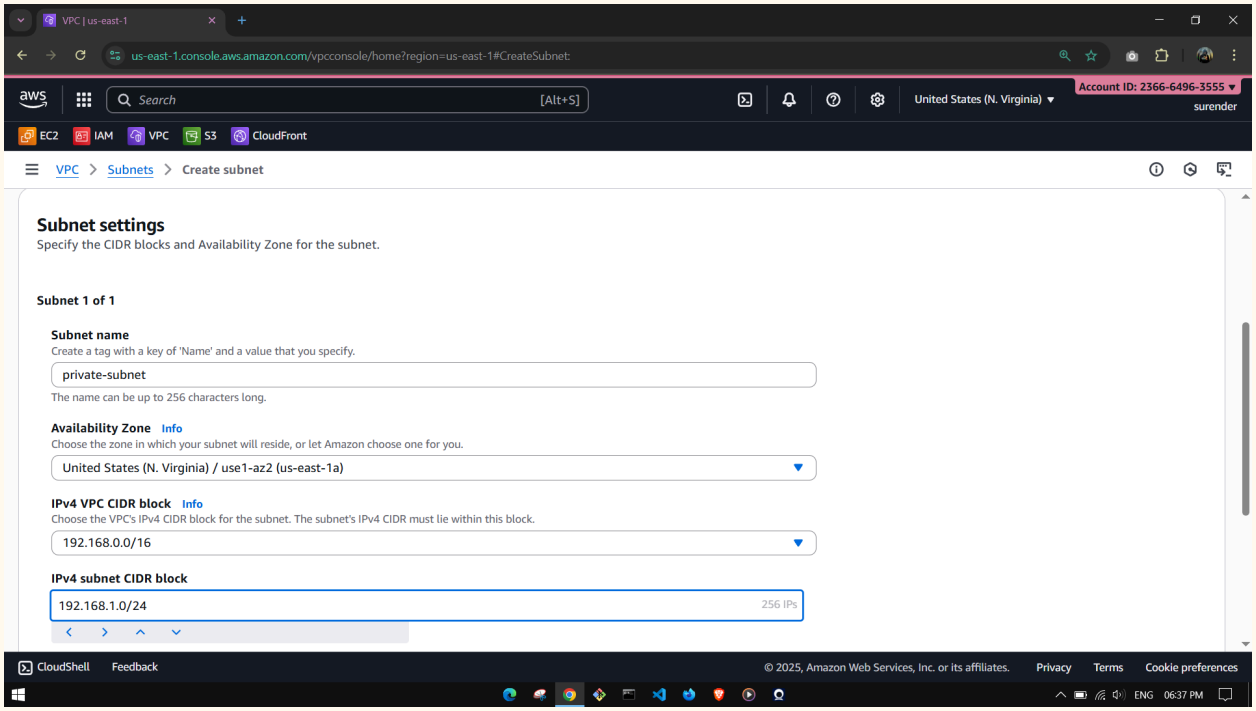
Now create a subnet from the subnet service of vpc



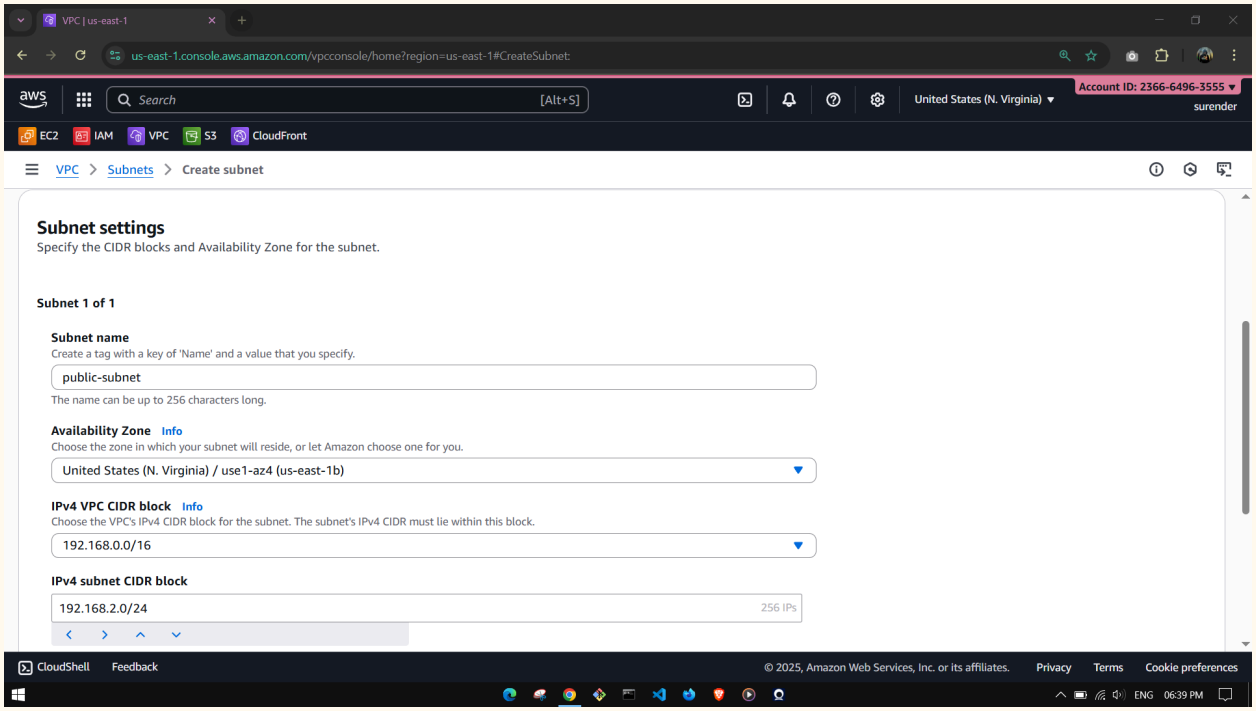
Select your vpc which you created before and create a private subnet



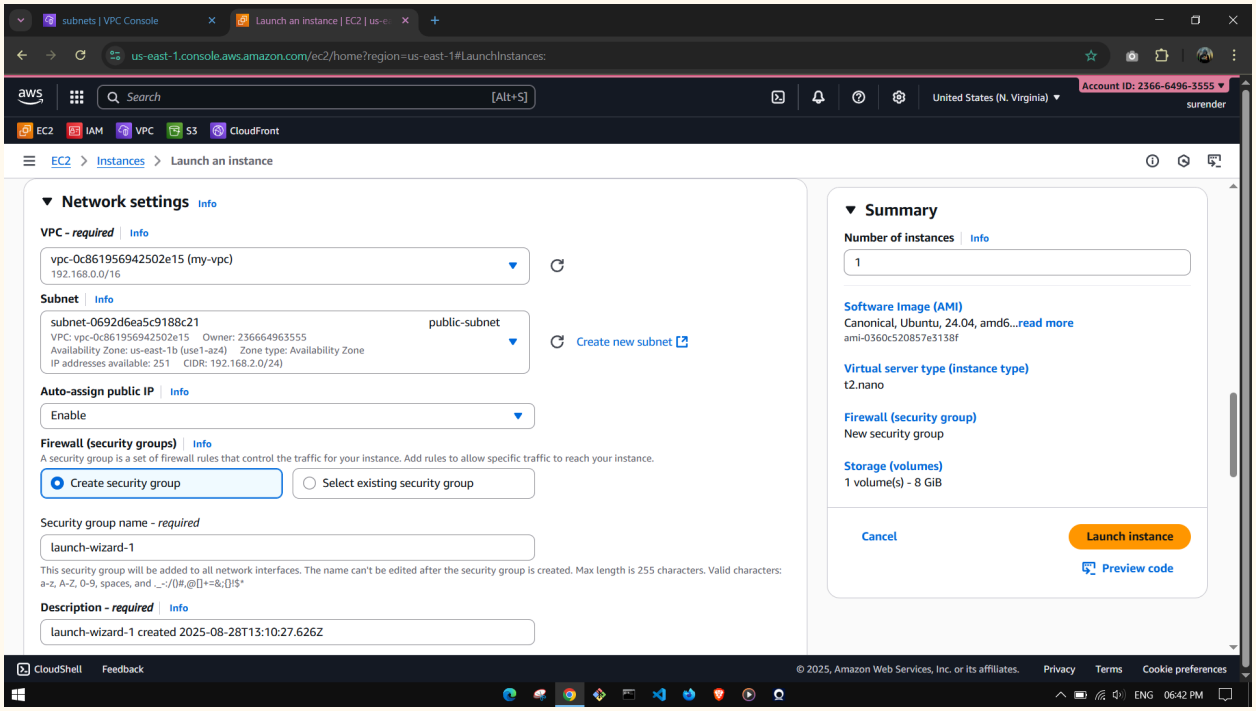
Now give a name and CIDR range



Now create another public subnet with a name and CIDR block

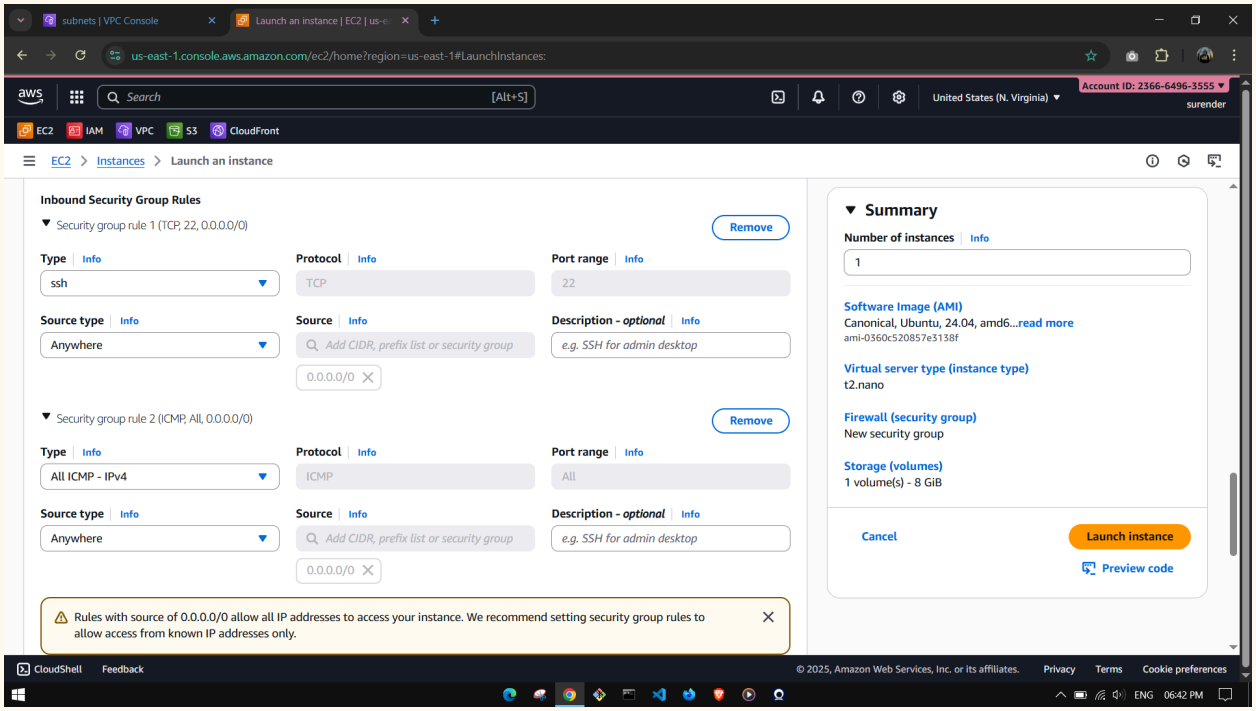


Now After creating public and private subnet create an instance in your public subnet

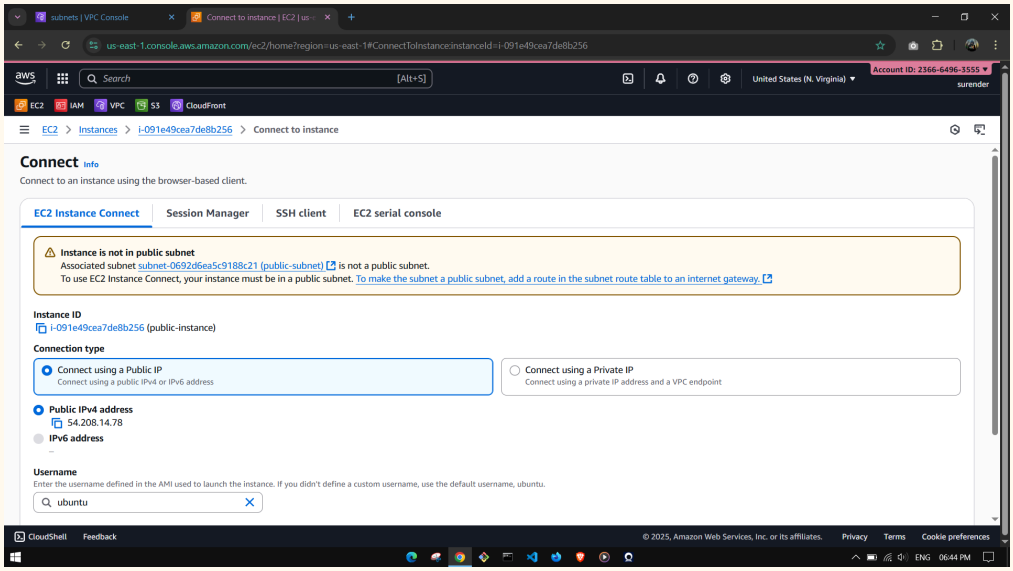


Select vpc which we created first, select public subnet and enable public ip to your instance

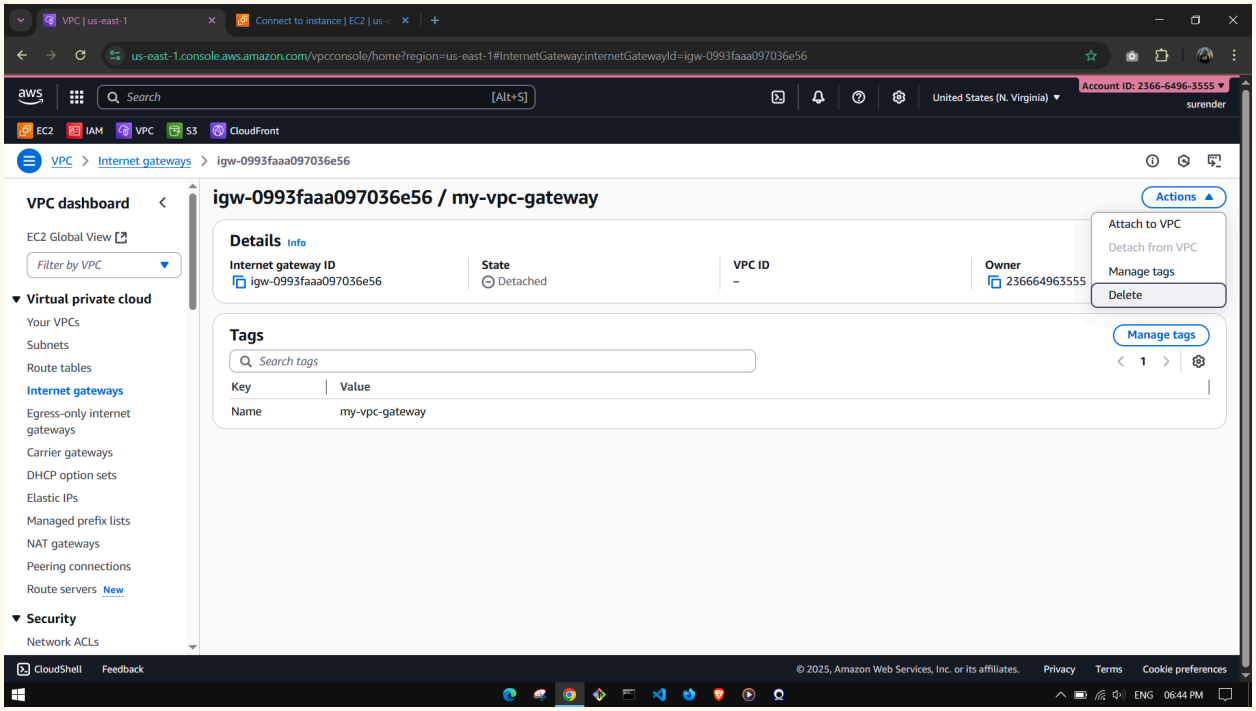
Allow icmp from anywhere in inbound rule of your security group and launch instance

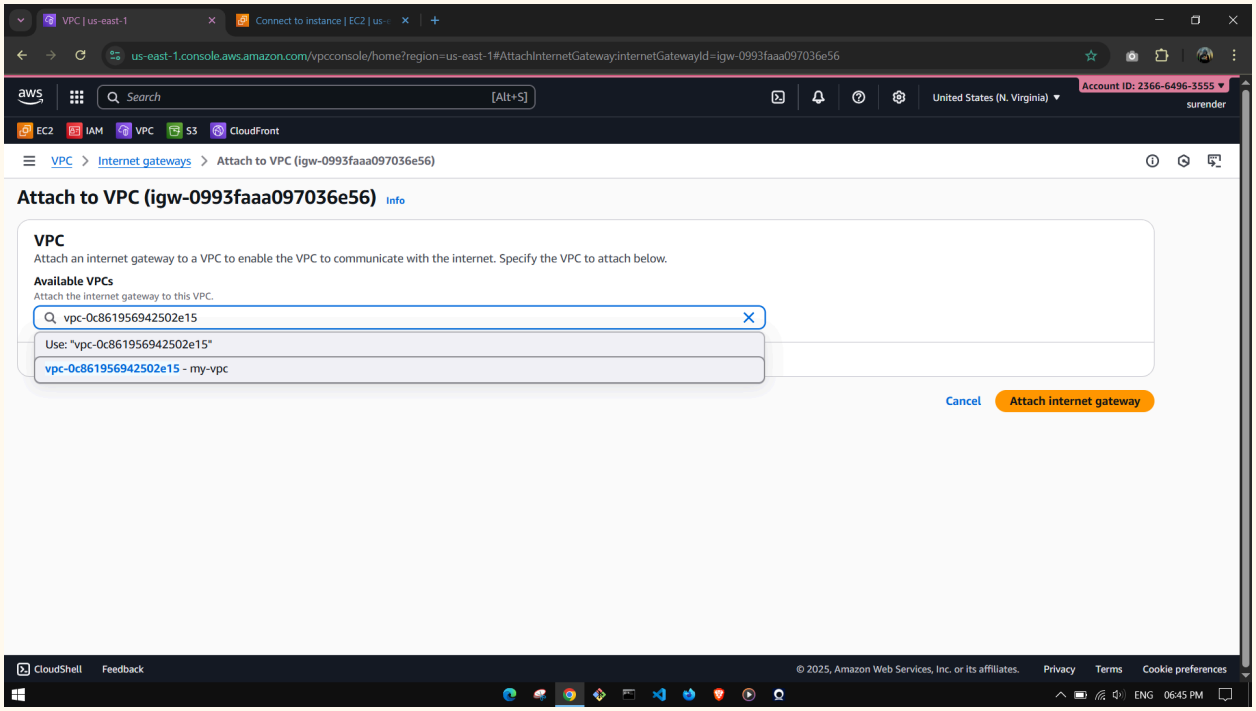


Try to connect your instance and you see you are not able to connect to your instance this happens because you are not provided because any traffic is not allowed to your instance and for that we have to attach **internet gateway** and assign a **route table.**

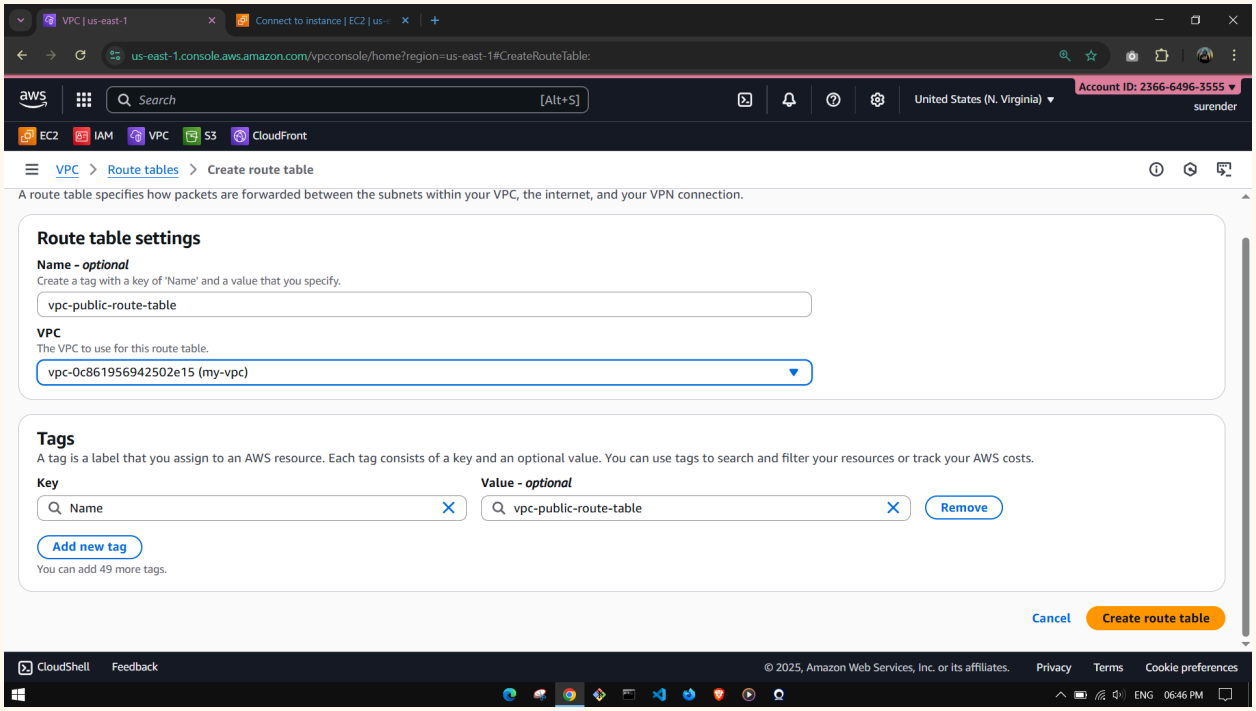


So first create **Internet Gateway** and attach to your vpc you created

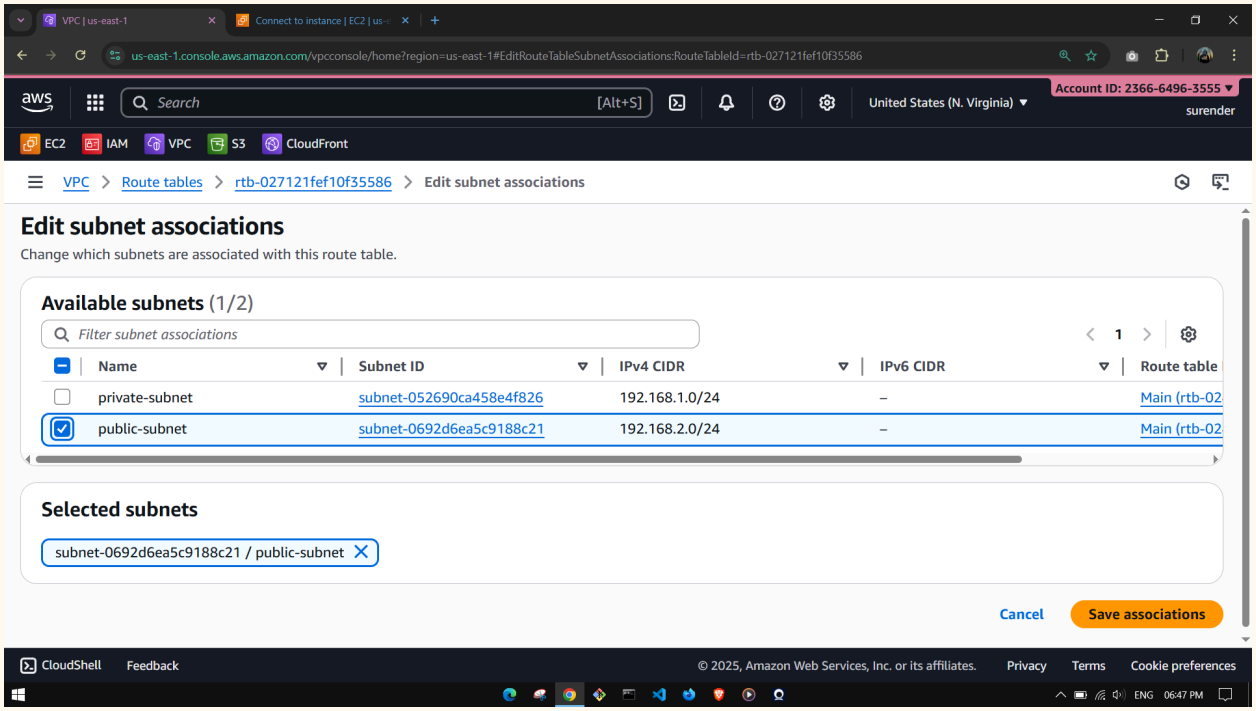




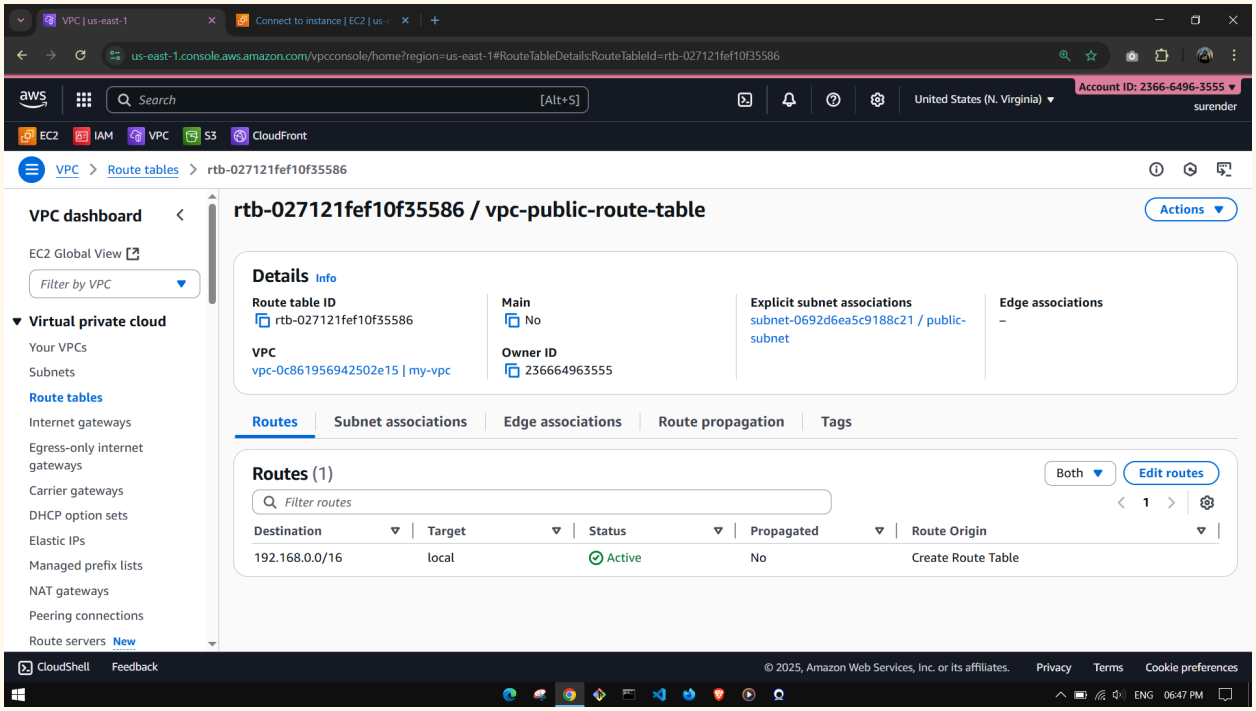
Now create a public **Route Table** for your vpc

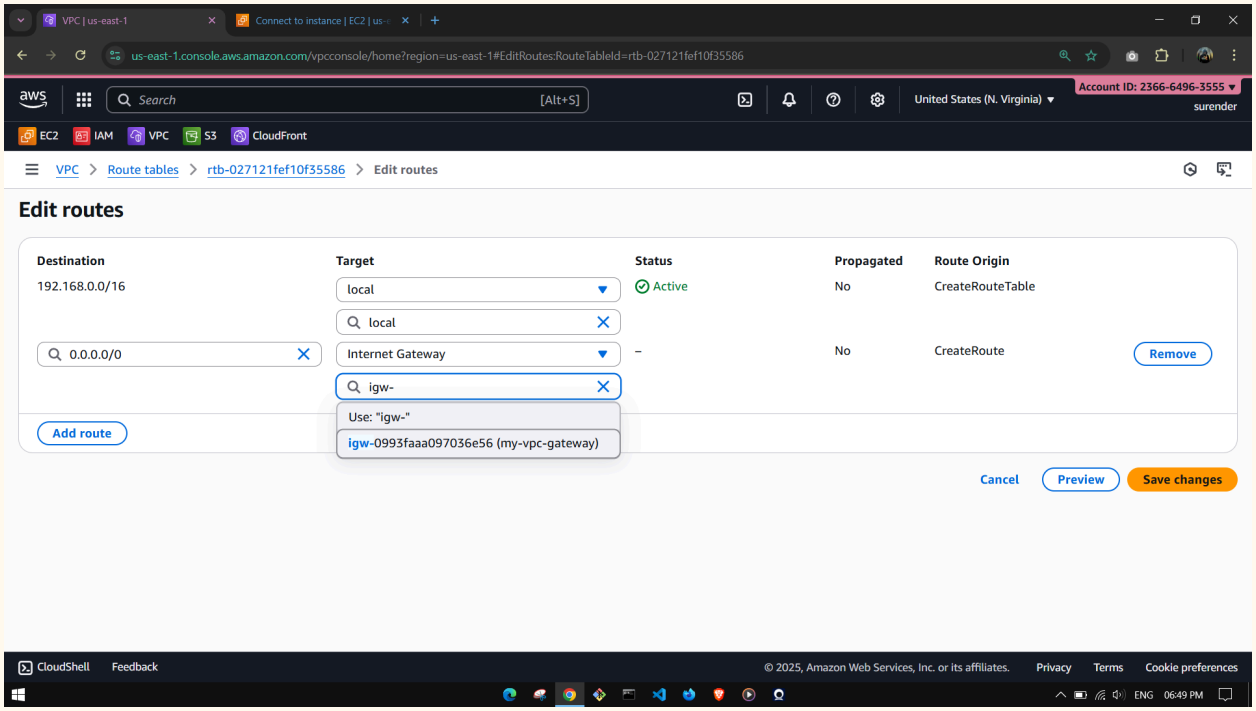


Associate your **public subnet** to your **public route table**

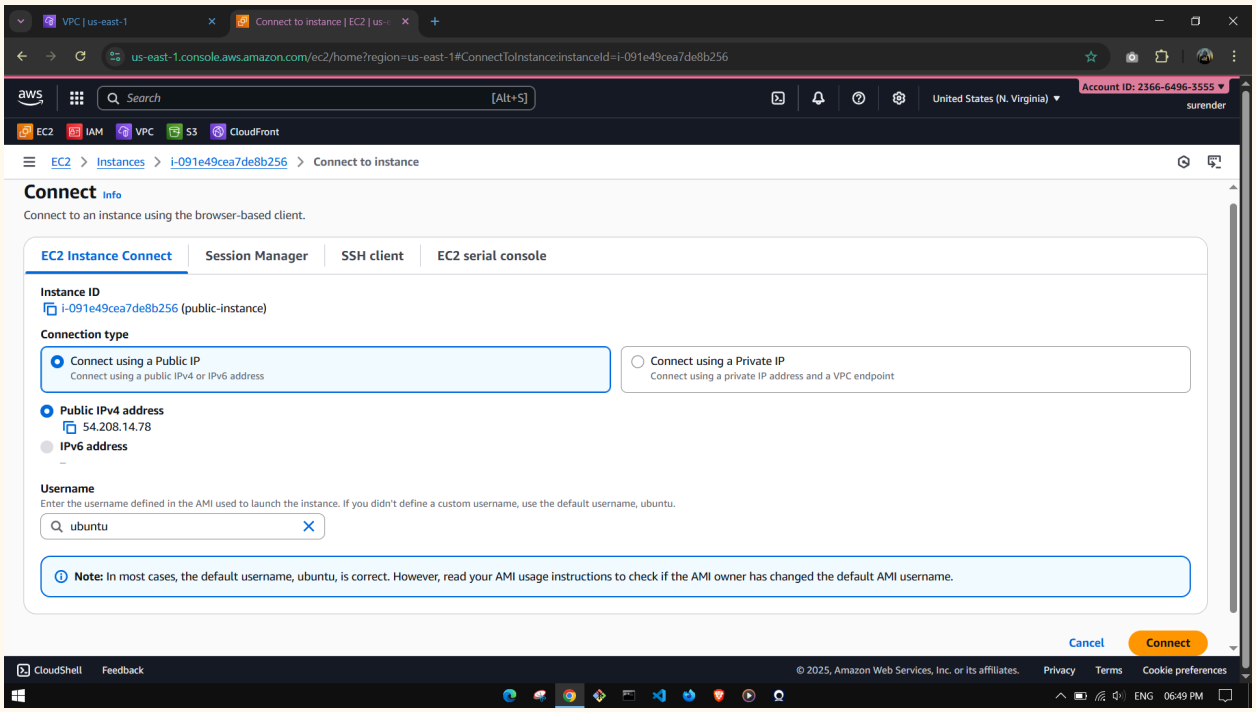


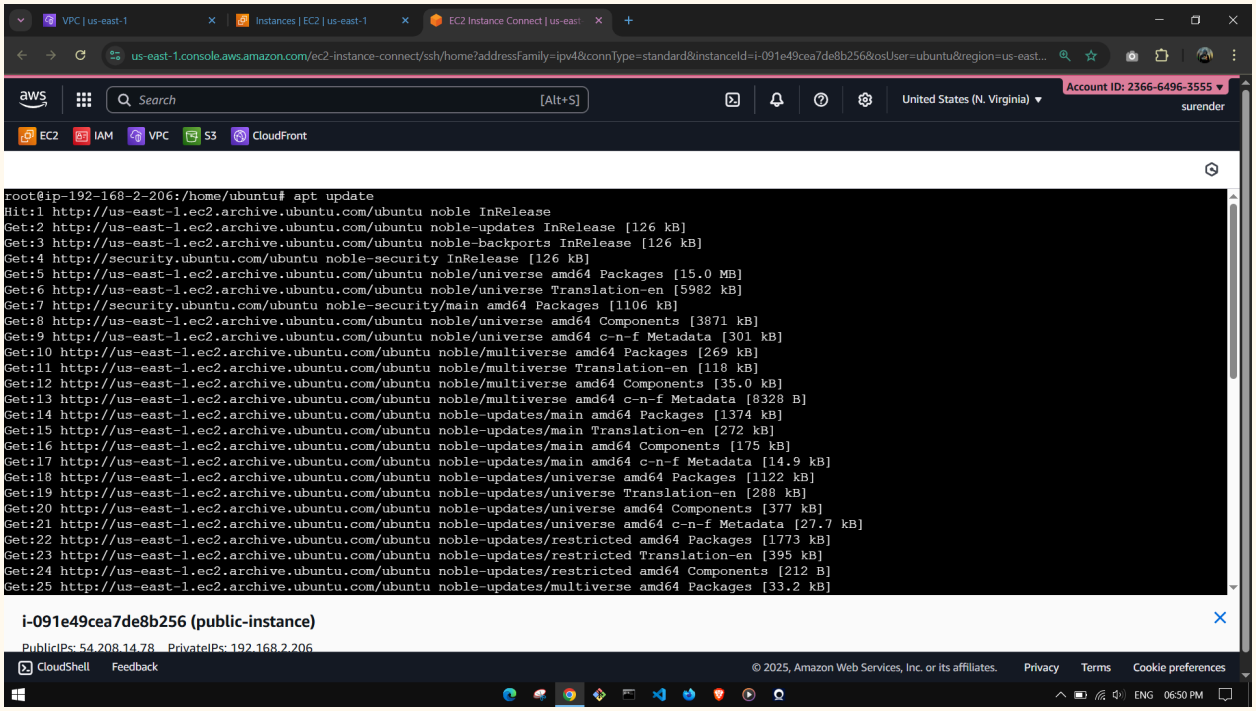
Create a route in your public route table for your internet gateway



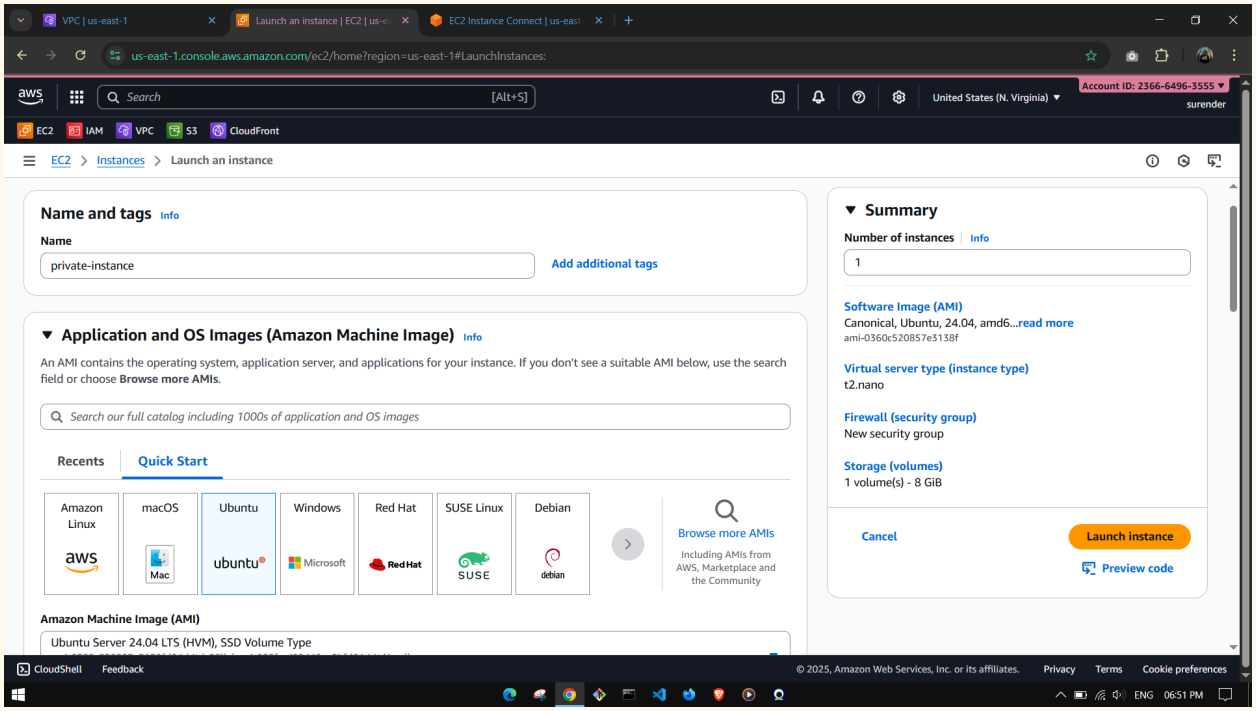


And now you are able to connect your public instance which you created in the public subnet of your vpc and you can run the internet in your instance

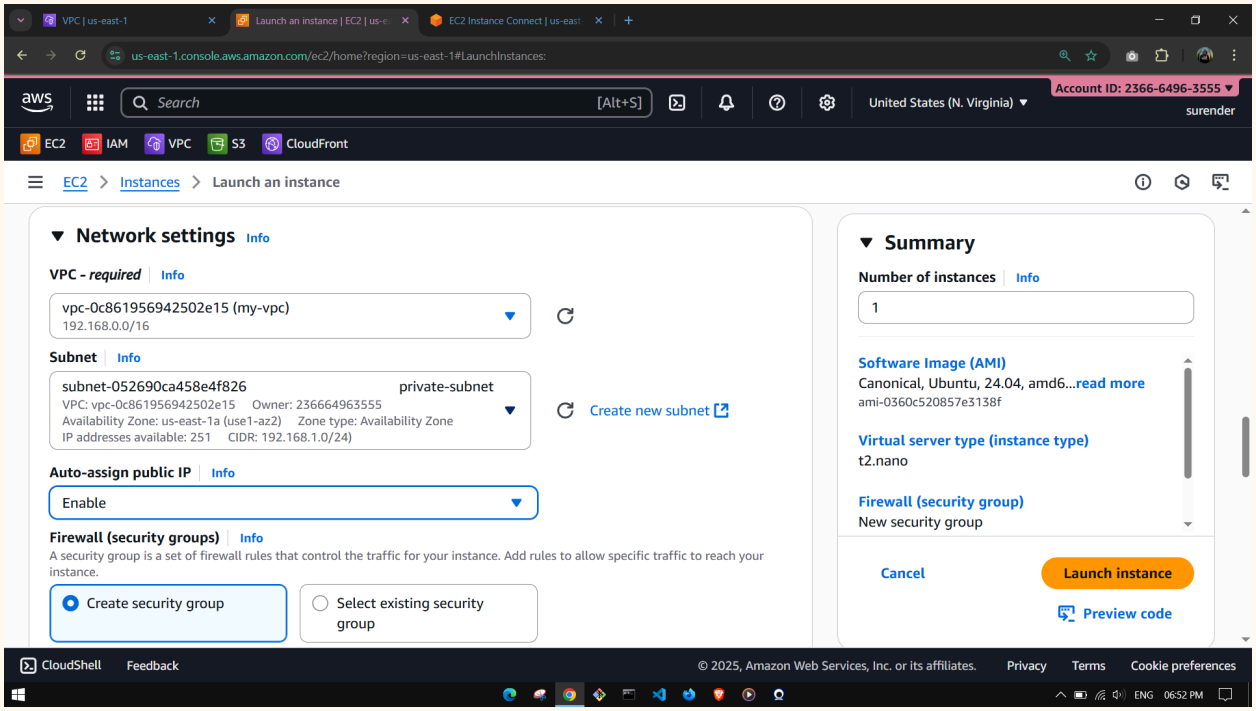




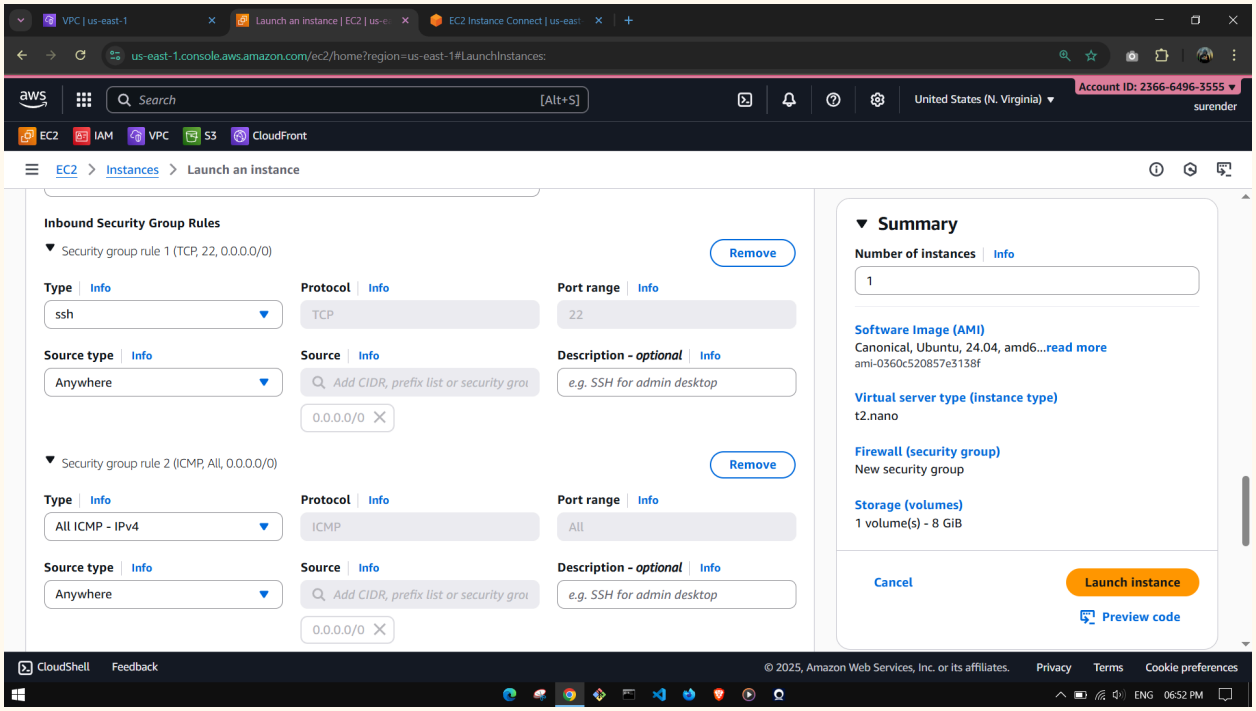
Now create another instance in your private subnet (i.e. private instance)



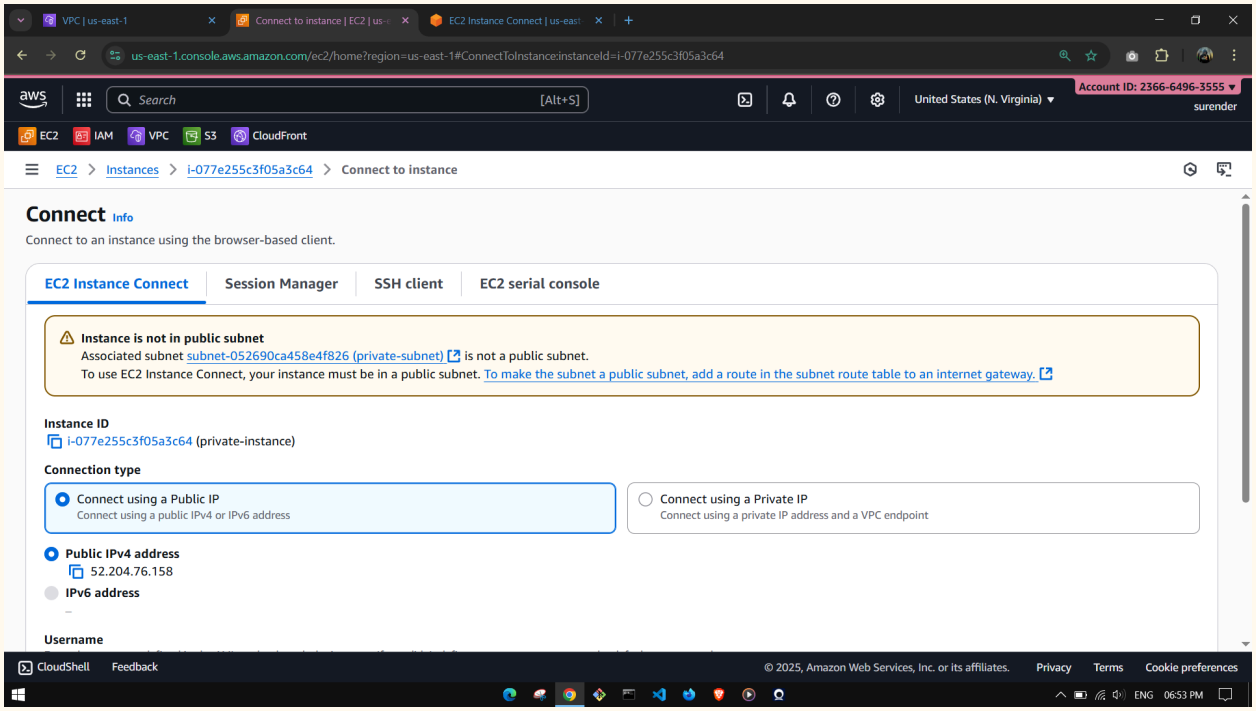
Select your vpc and private subnet and assign public ip to your instance



Allow icmp rule from anywhere and launch instance

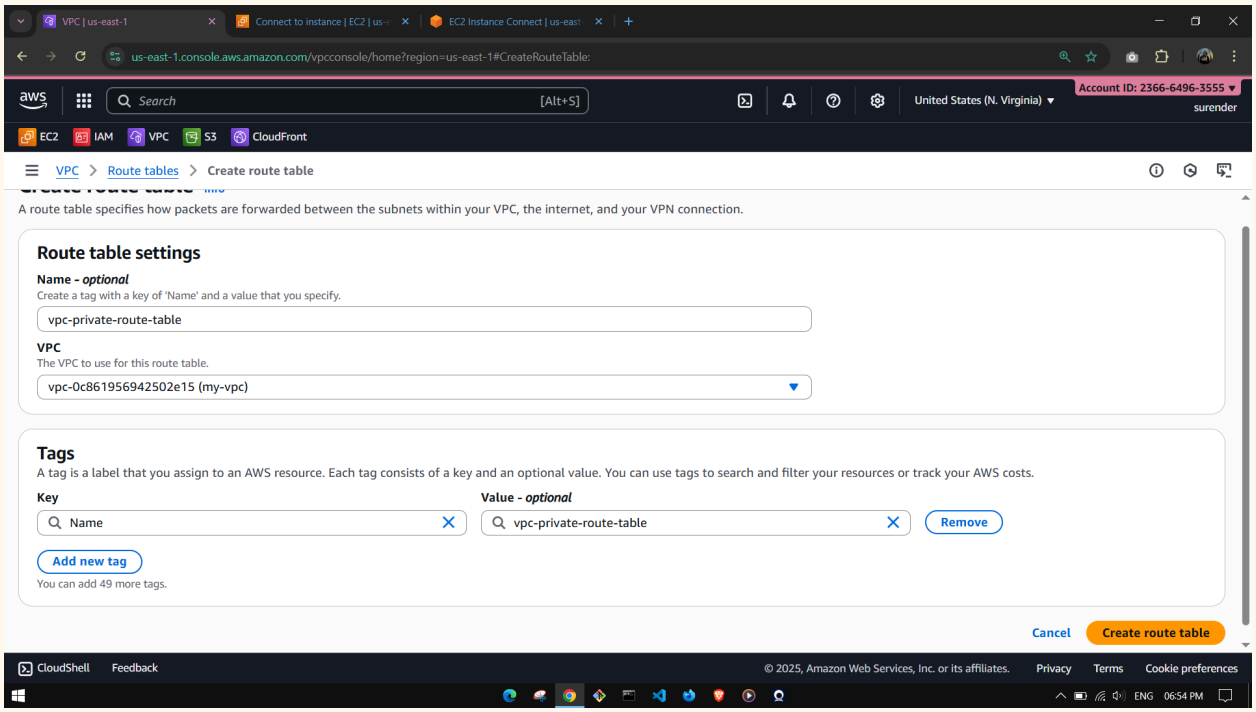


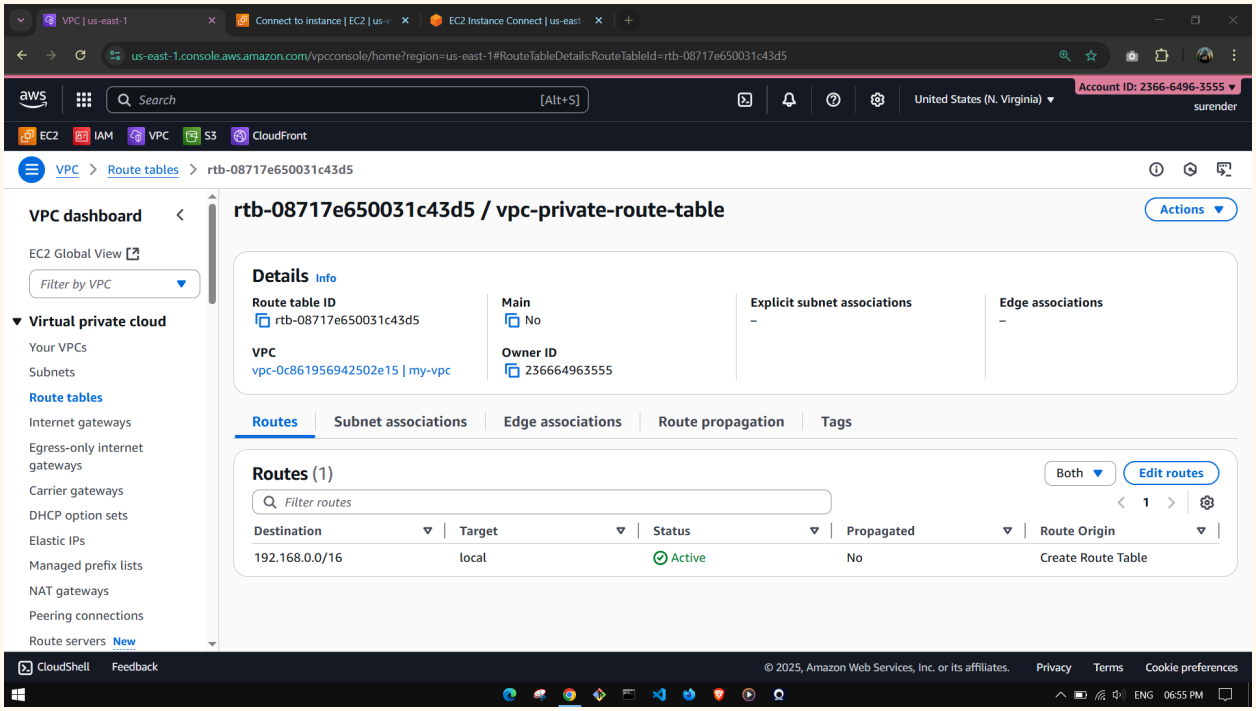
Try to connect your instance



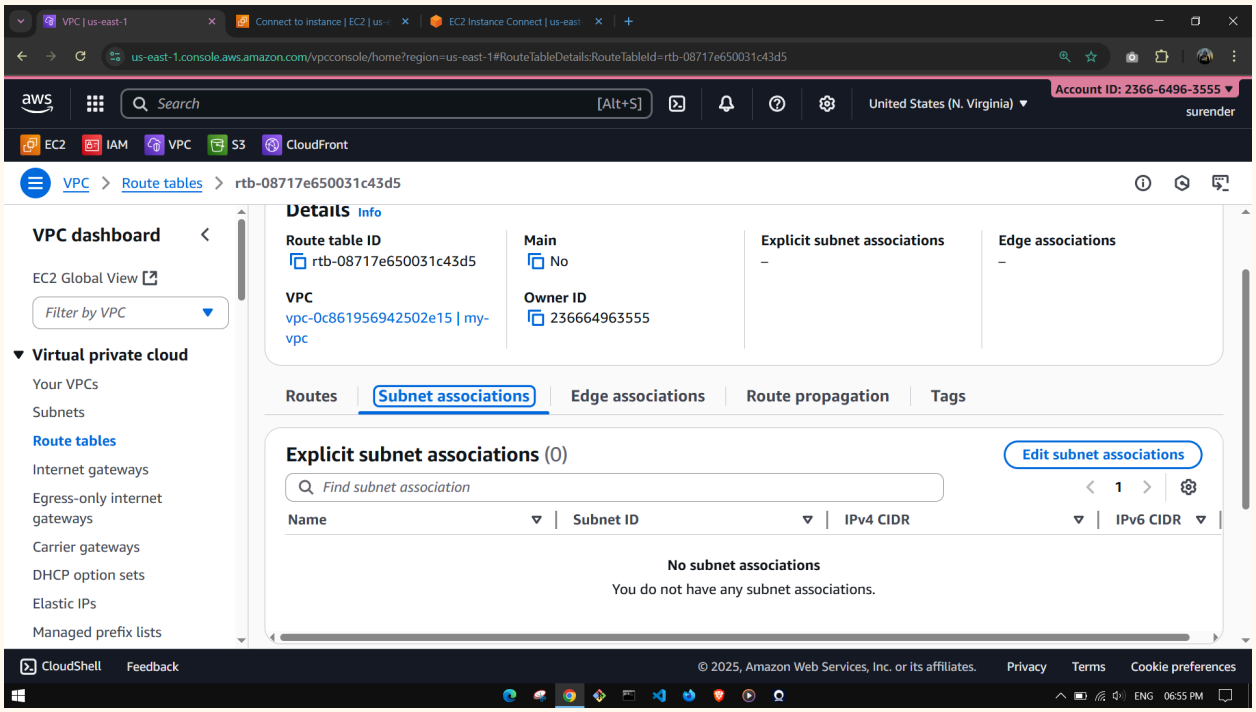
Your are not able to connect your instance directly because private subnet is not connected to the internet gateway and does not have any route table for this private subnet

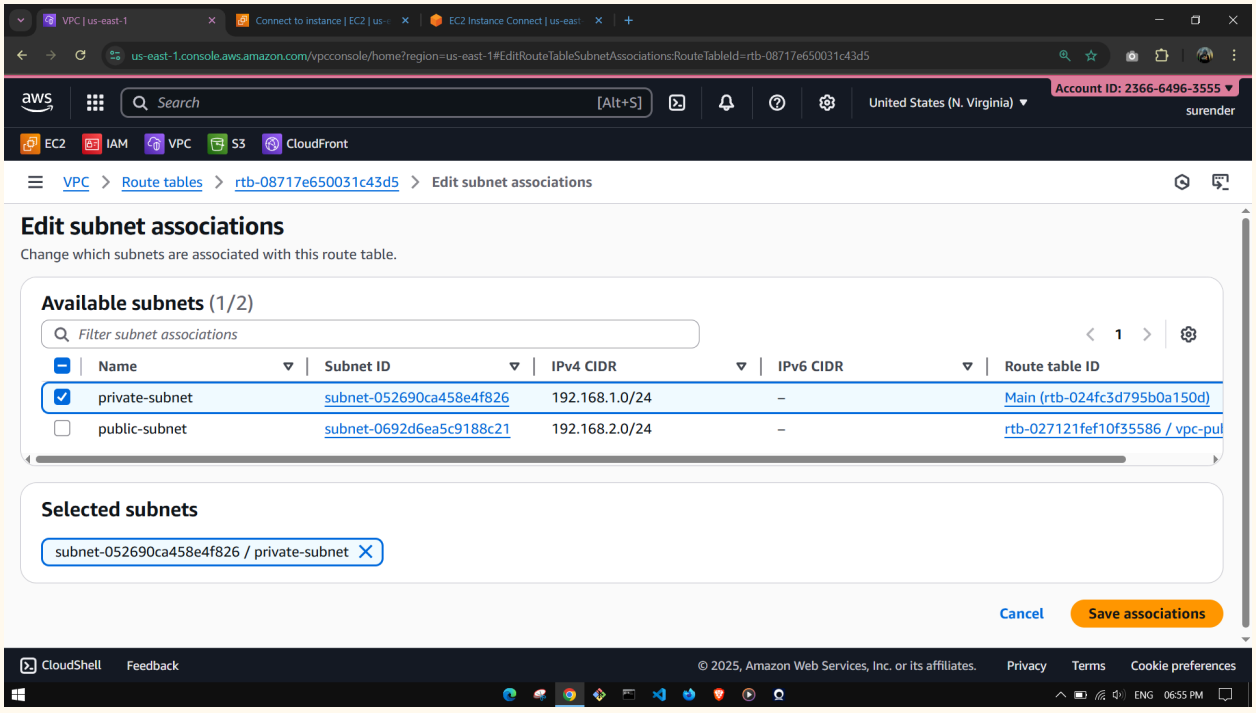
So create a route table for this private subnet



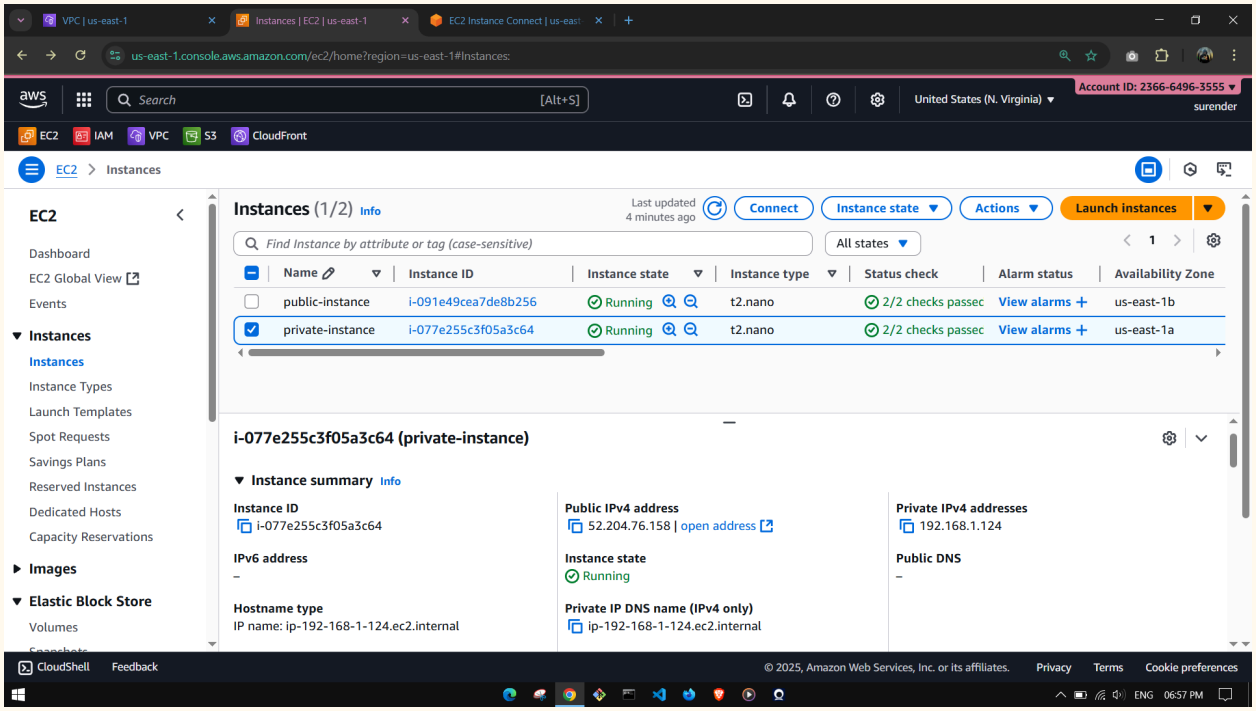


Form the subnet associations tab associate your private subnet to your private route table from the button **edit subnet associations**

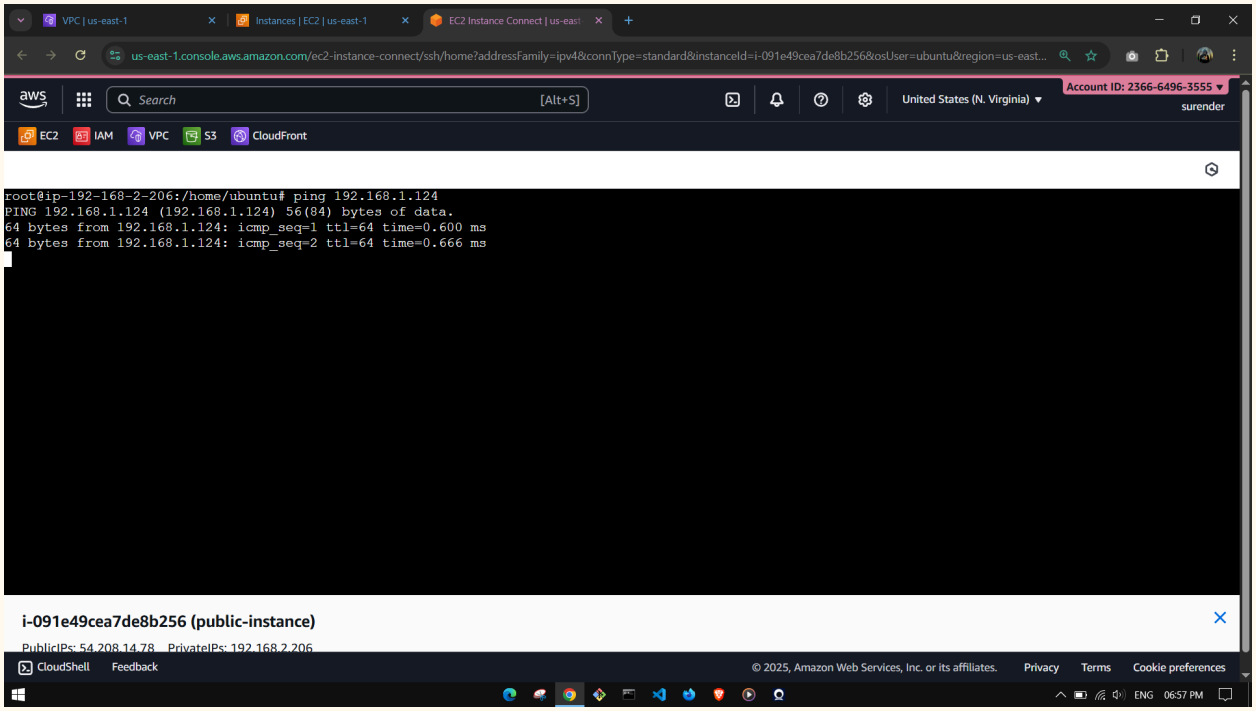




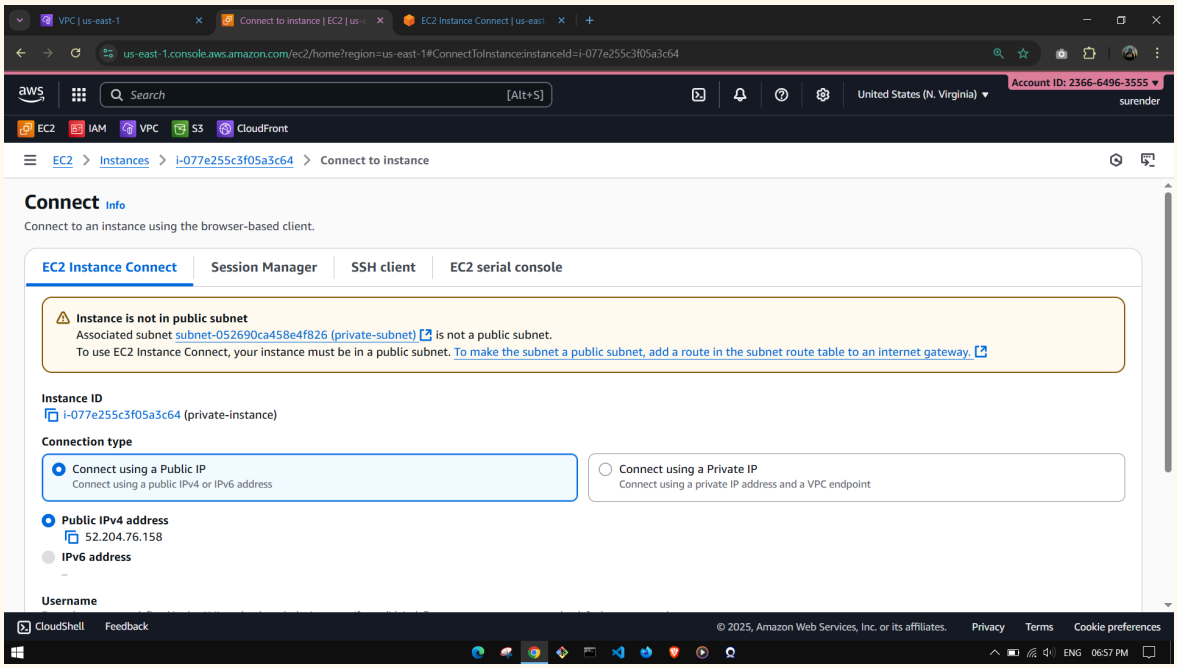
Select your private instance and try to ping it with your public instance



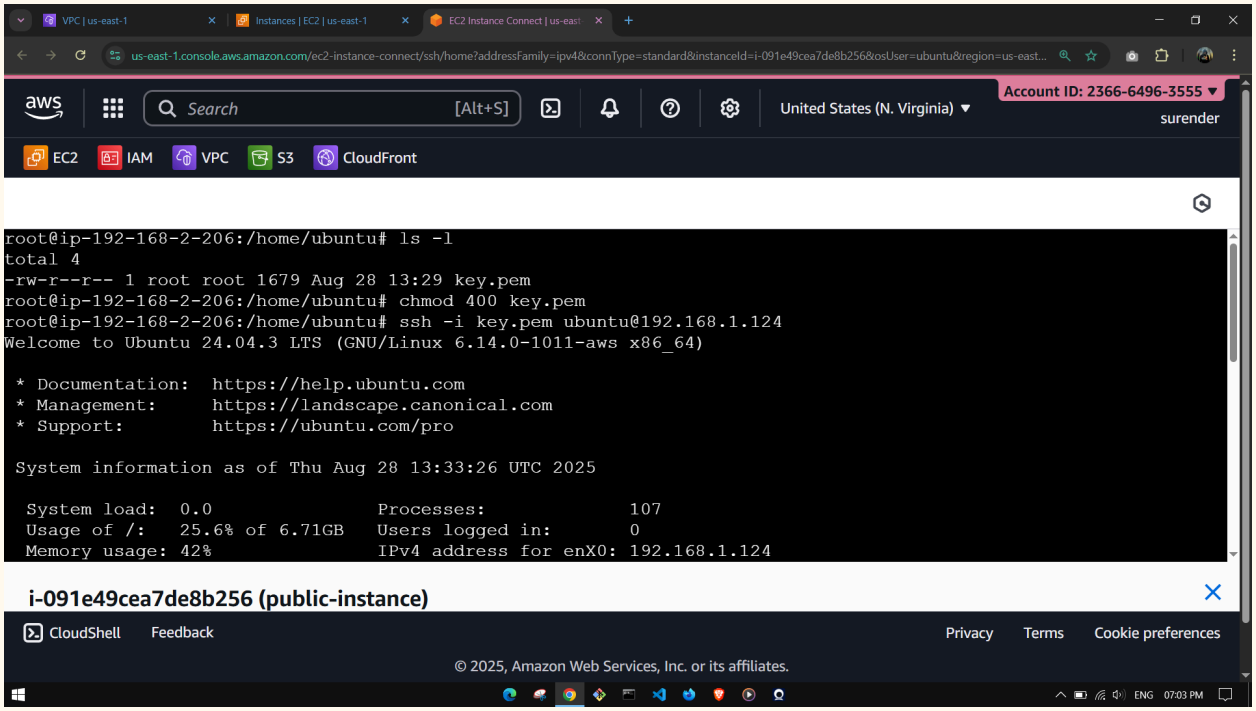
Here you can see instances can communicate internally with each other even if they are in different subnets and even if they have not internet connection.



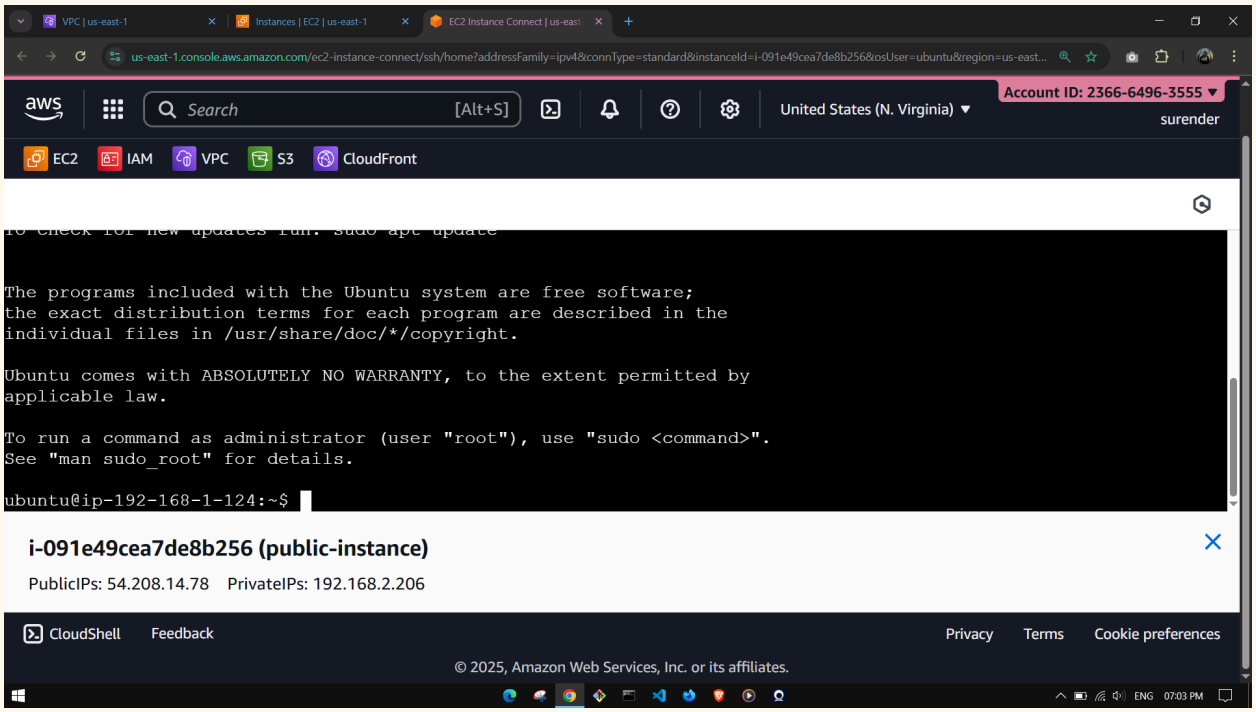
But you can't ssh your private instance because it has only internal network to communicate you can see clearly that you can not ssh directly because it dont have network connection with you



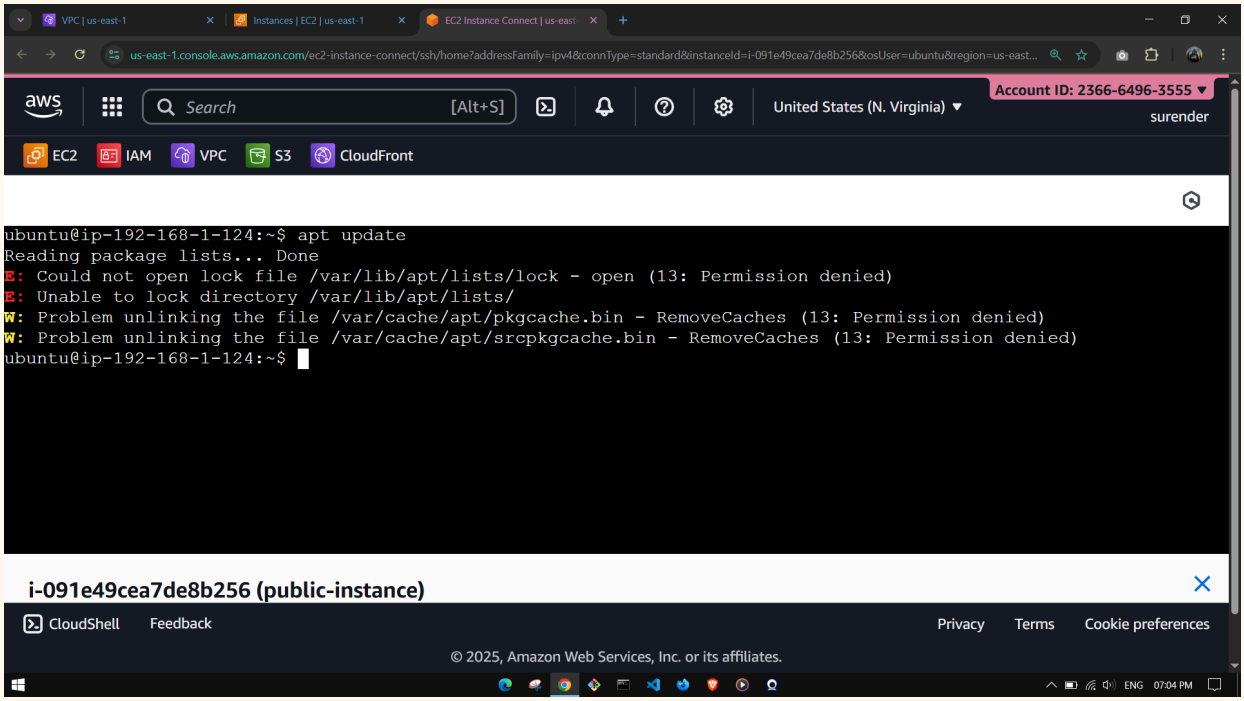
But there is a way to ssh your private instance, you can ssh your private instance with your public instance, firstly change the permission of your key as shown in the image and try to ssh it



You can clearly see, you have done ssh connection with the private instance with your public instance, now this public instance is called jump server or bastien host



Your are able to ssh your private instance from your public instance due to internal connectivity but it does not have internet connection as you can see in the below image



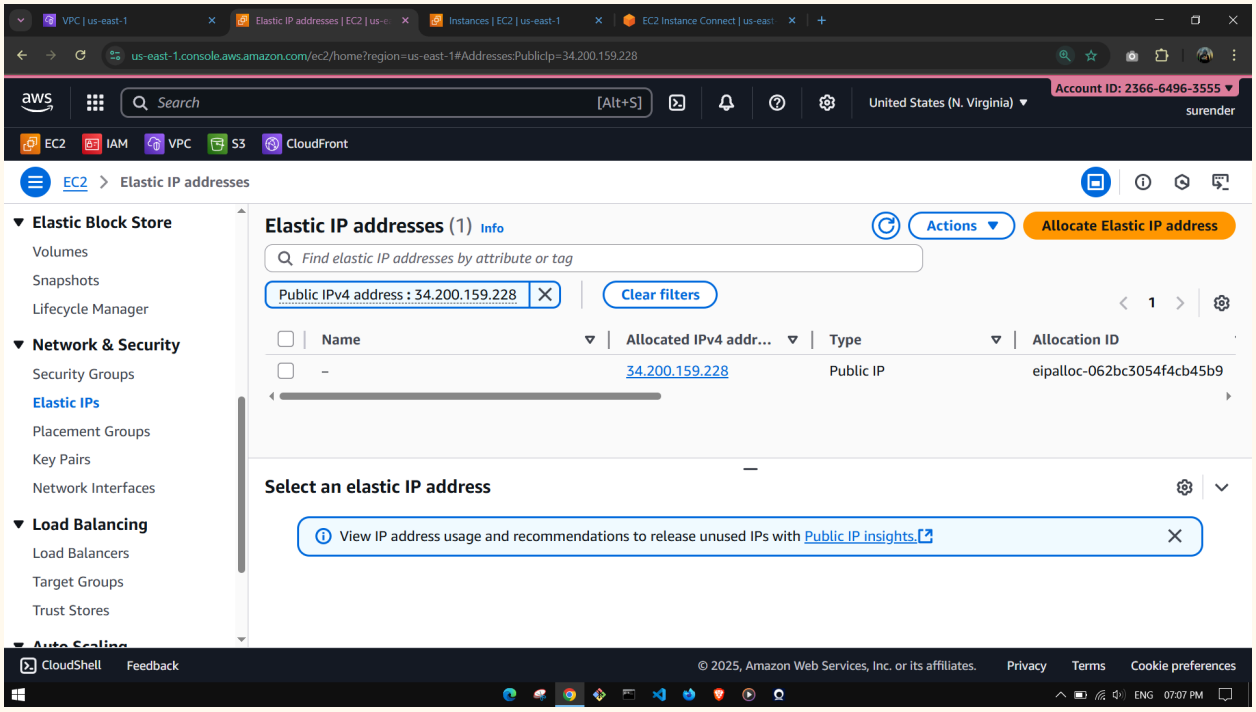
For the internet connection in your private instance you have two common option

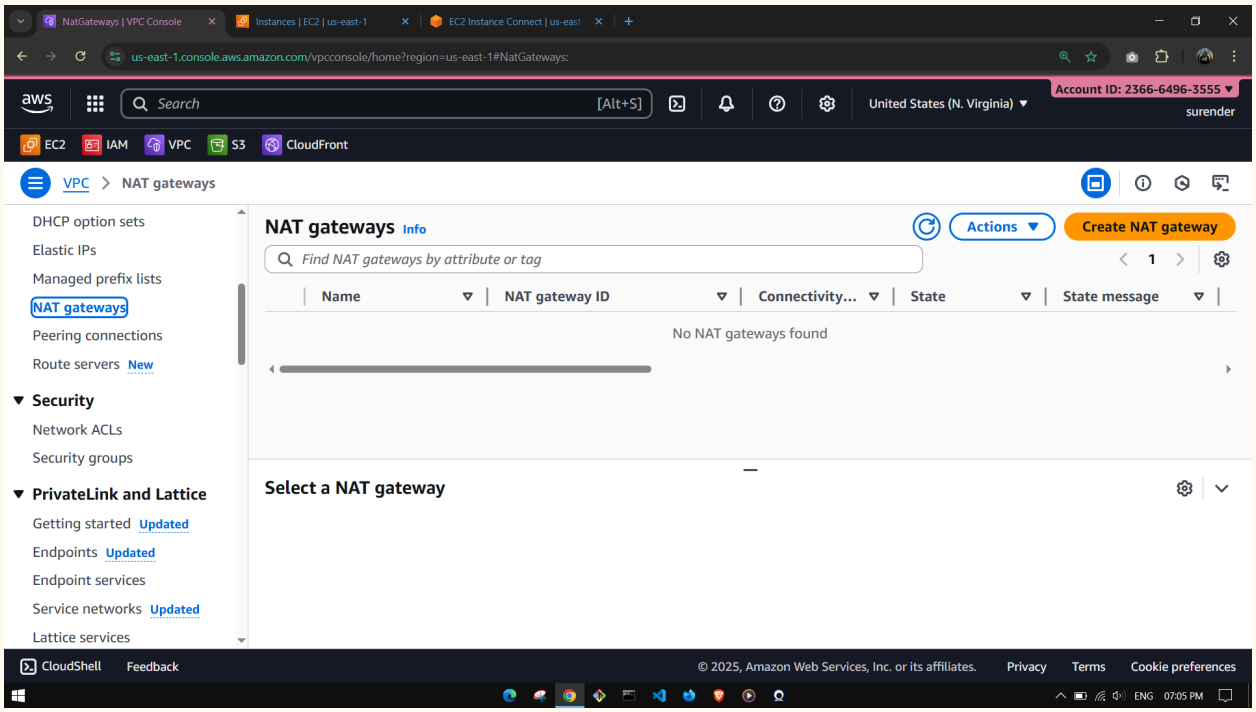
1. Through the NAT instance
2. Through the NAT gateway

We will use NAT gateway because it is simple and easy. AWS manages the NAT gateway other wise we have to manage the NAT instance which may be very serious task and AWS suggest to use NAT gateway because it is secure and managed by AWS itself. No need to take haptic.

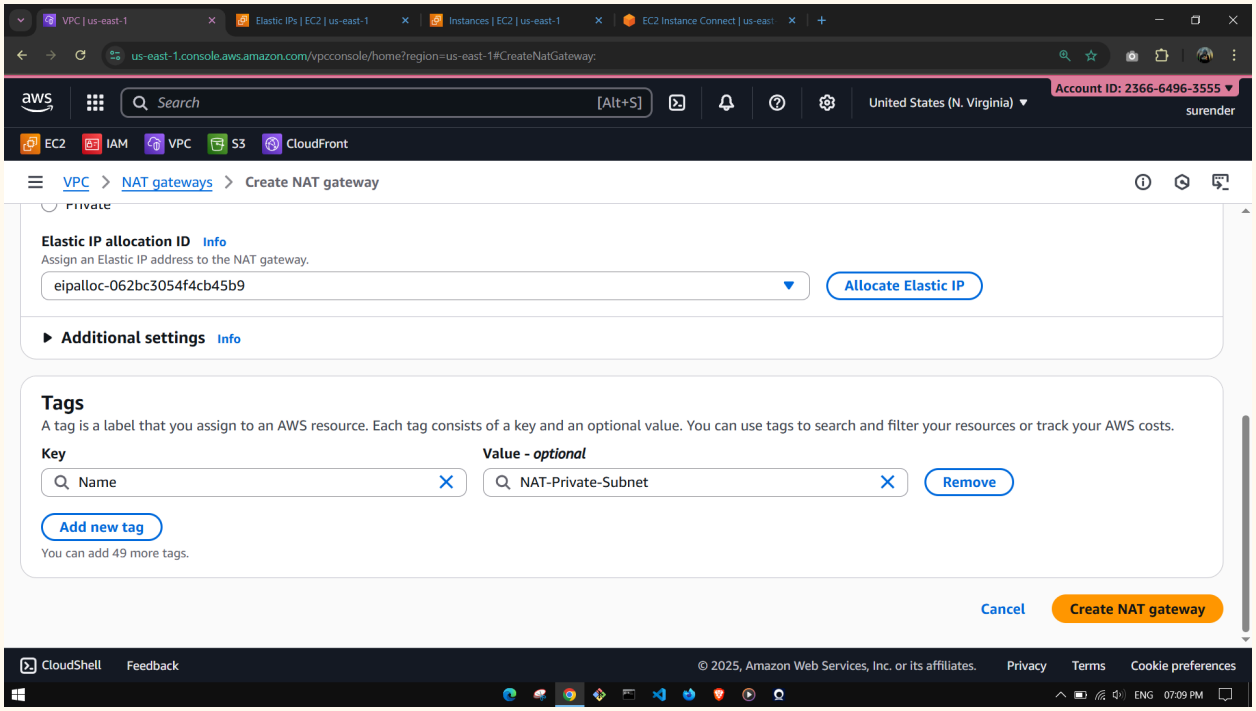
Go to NAT gateway and create one for your private instance for secure internet connection in your private instance.

But firstly we have to create an **Elastic IP** for your NAT Gateway. From the EC2 service create an Elastic ip

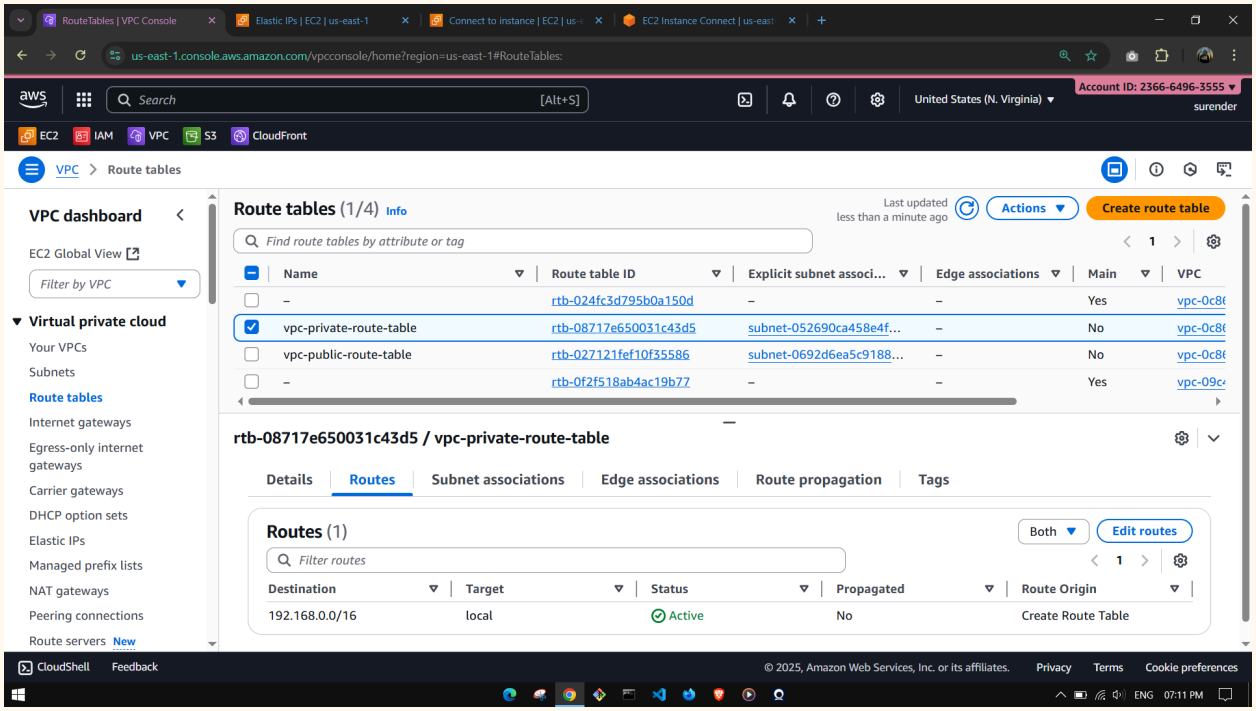




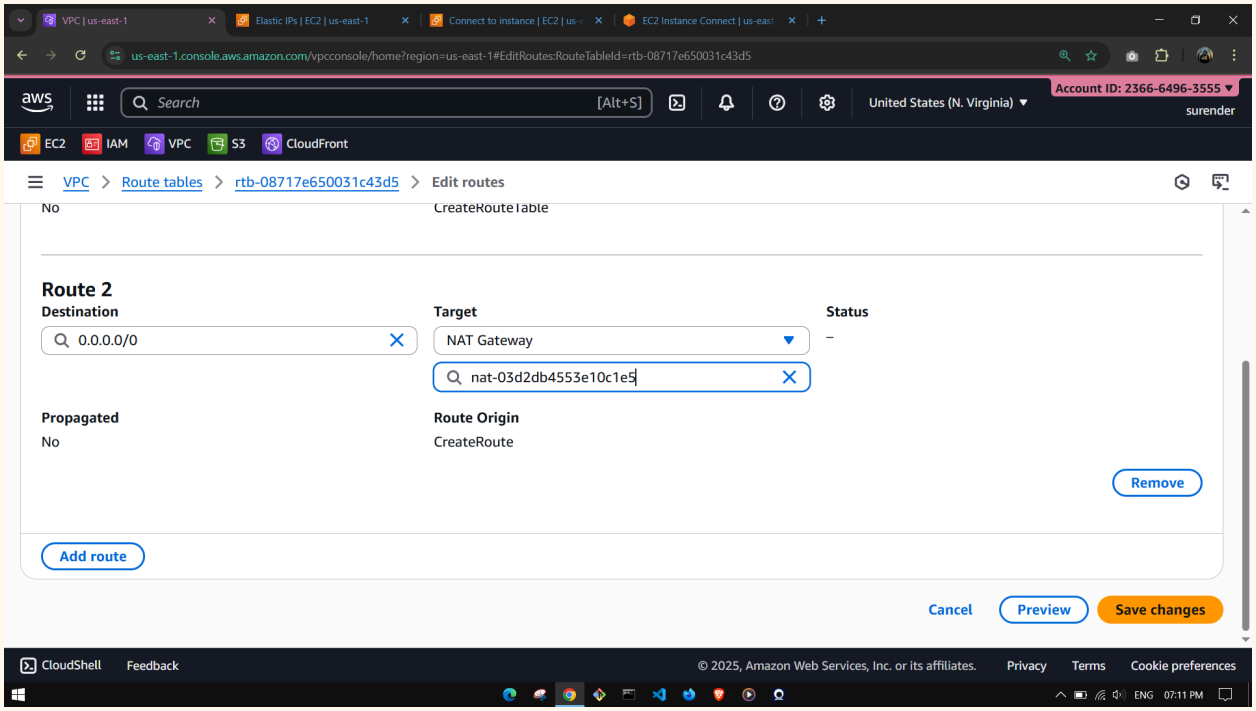
Attach your elastic ip which you created before creating you NAT Gateway



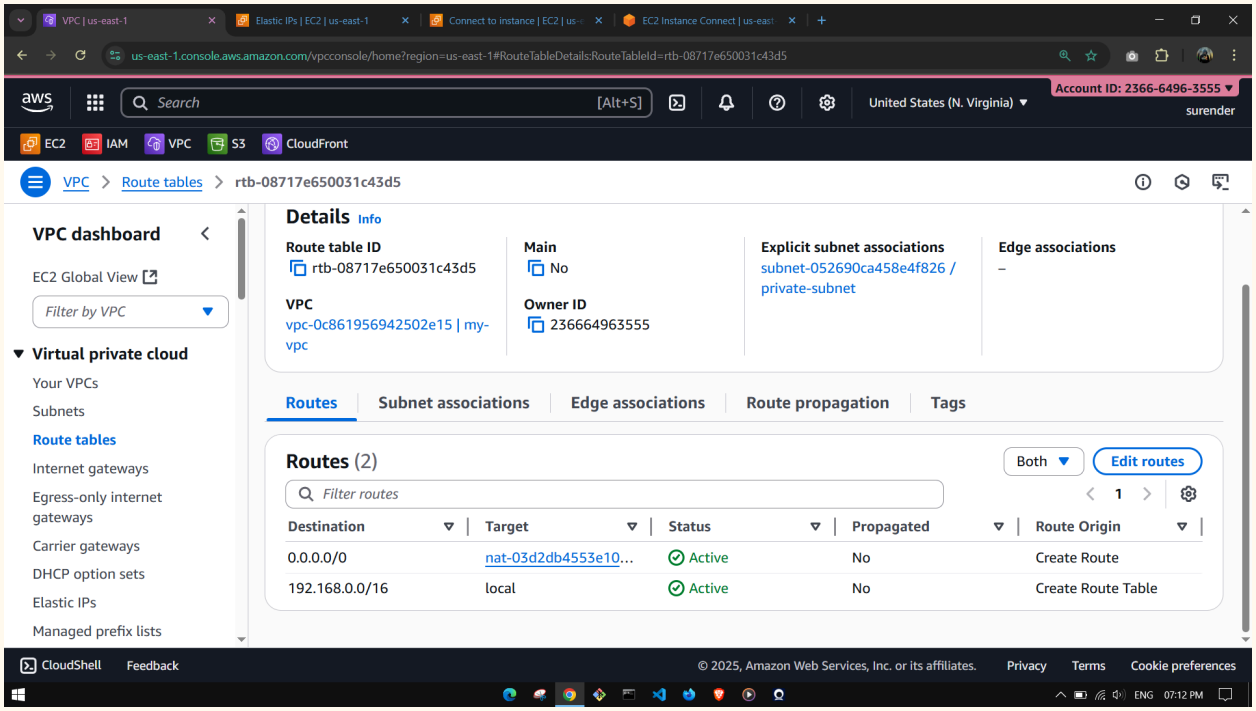
Now go your private route table and add one more route for your NAT Gateway



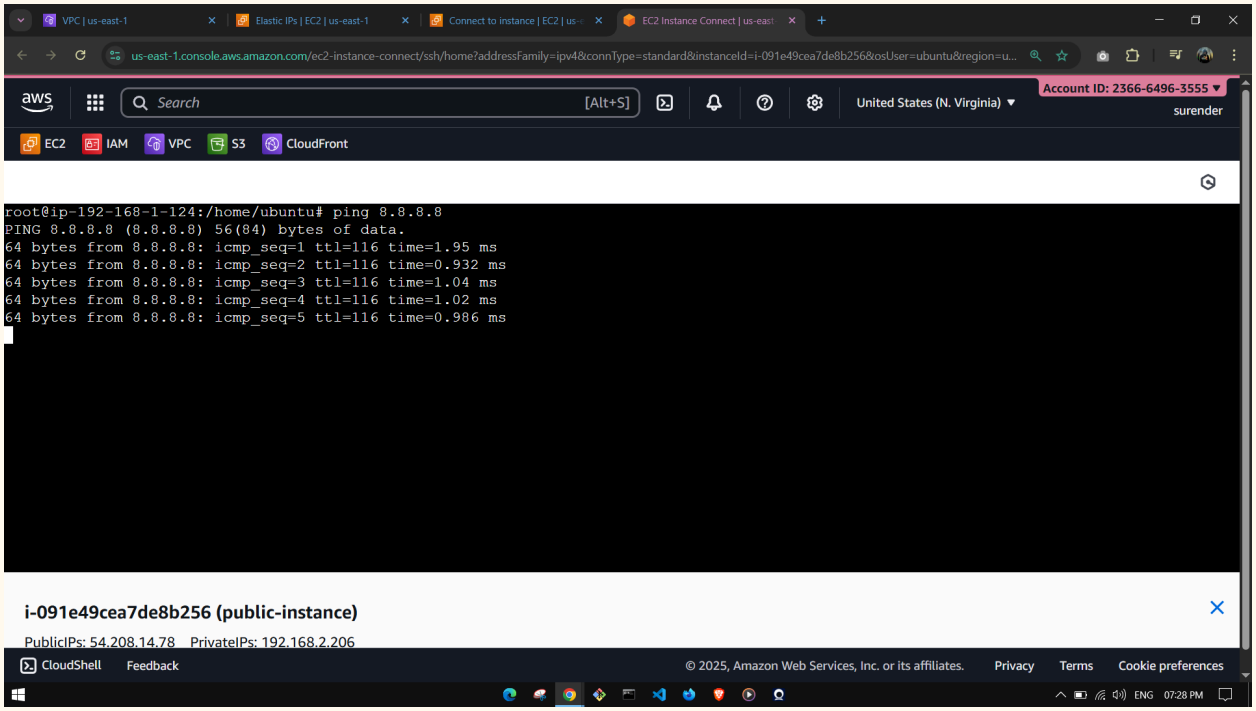
Target should be your NAT Gateway and select your NAT gateway and click to save changes.



This is your output after adding route for NAT gateway



Now you are able to run the internet in your private instance. For confirmation try to ping google as shown below.



Let's Connect :-

Lalit Kumar
Follow: LinkedIn
Profile: Portfolio