# Placement Empowerment Program

***Cloud Computing and DevOps Centre***

Secure Access with a Bastion Host : Set up a bastion host in a public subnet to securely access instances in a private subnet.

Name: Shahana.M.S Department:ADS



Introduction:

A **bastion host** is a special-purpose server designed to provide **secure access** to instances located in a **private subnet** of a cloud environment. It acts as an **intermediary** between external users and private instances, reducing the risk of direct exposure to the internet. By placing the bastion host in a **public subnet**, users can securely connect to private instances using SSH or RDP, ensuring better security and access control.

**Overview**

In a **cloud networking environment**, private instances should not be exposed to the public internet for security reasons. Instead, a **bastion host** (also called a **jump server**) is deployed in a **public subnet** with a public IP, allowing controlled access to instances in the **private subnet**. The process involves:

* **Creating a bastion host in the public subnet** – A VM with SSH or RDP access.
* **Configuring security groups and firewall rules** – Restricting access to only trusted IP addresses.
* **Connecting to private instances through the bastion host** – Using SSH agent forwarding or proxy commands.

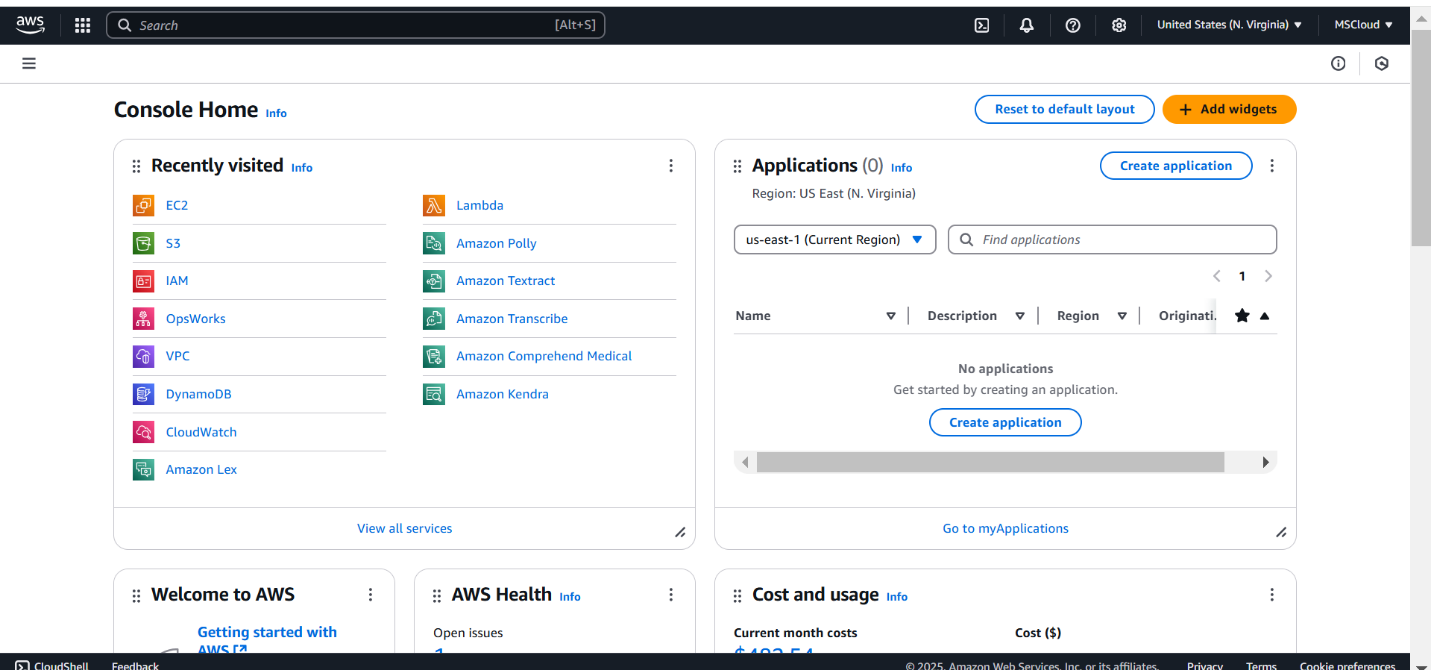
This setup enhances security while maintaining administrative access to critical cloud resources.

**Objective**

* To deploy a **bastion host** in a **public subnet** for controlled access.
* To configure **security groups** and **firewall rules** for secure connections.
* To access **private instances** via the bastion host using SSH or RDP.
* To ensure **better security and compliance** by limiting direct exposure to the internet.

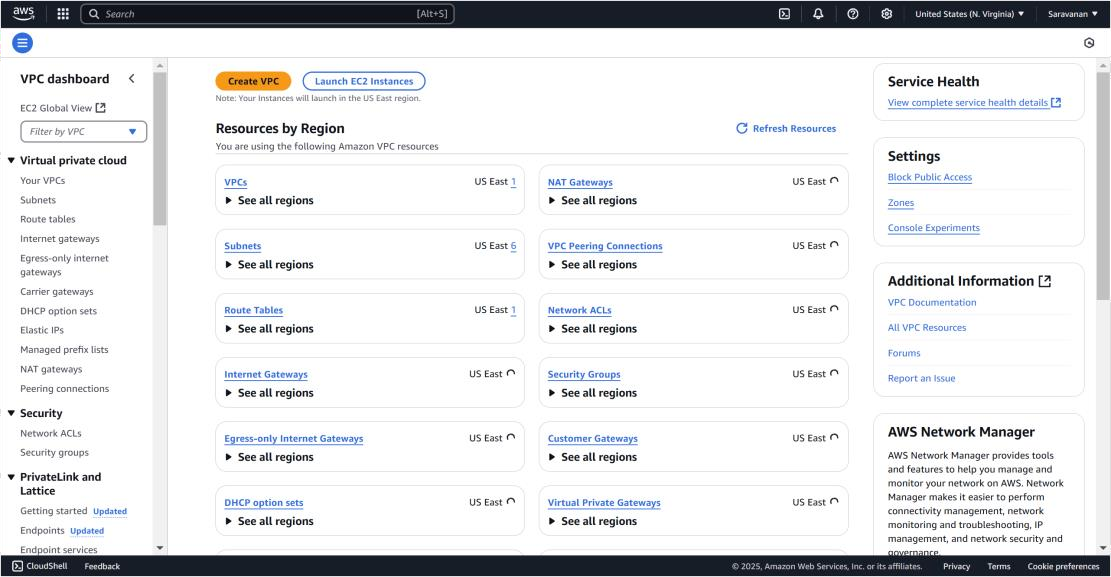
## Step 1:

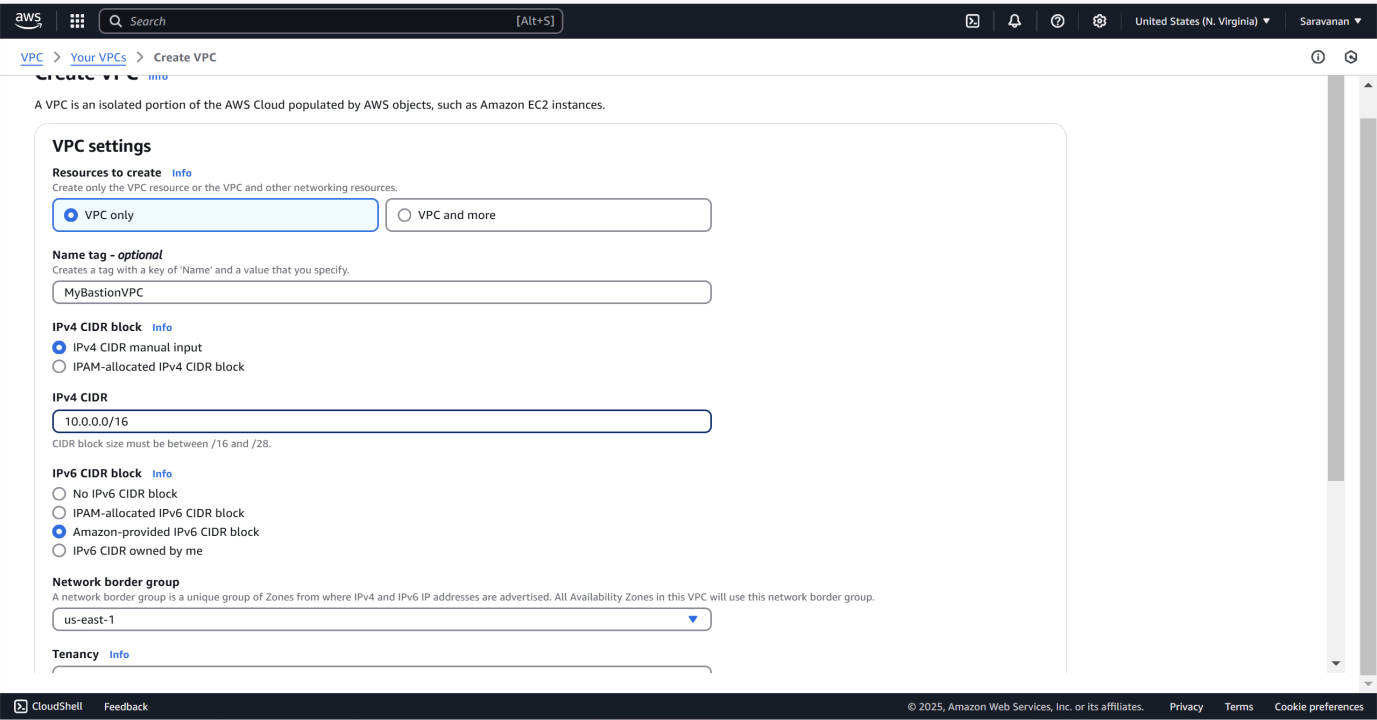
1. Go to [AWS Management Console](https://aws.amazon.com/console/).
2. Enter your username and password to log in.



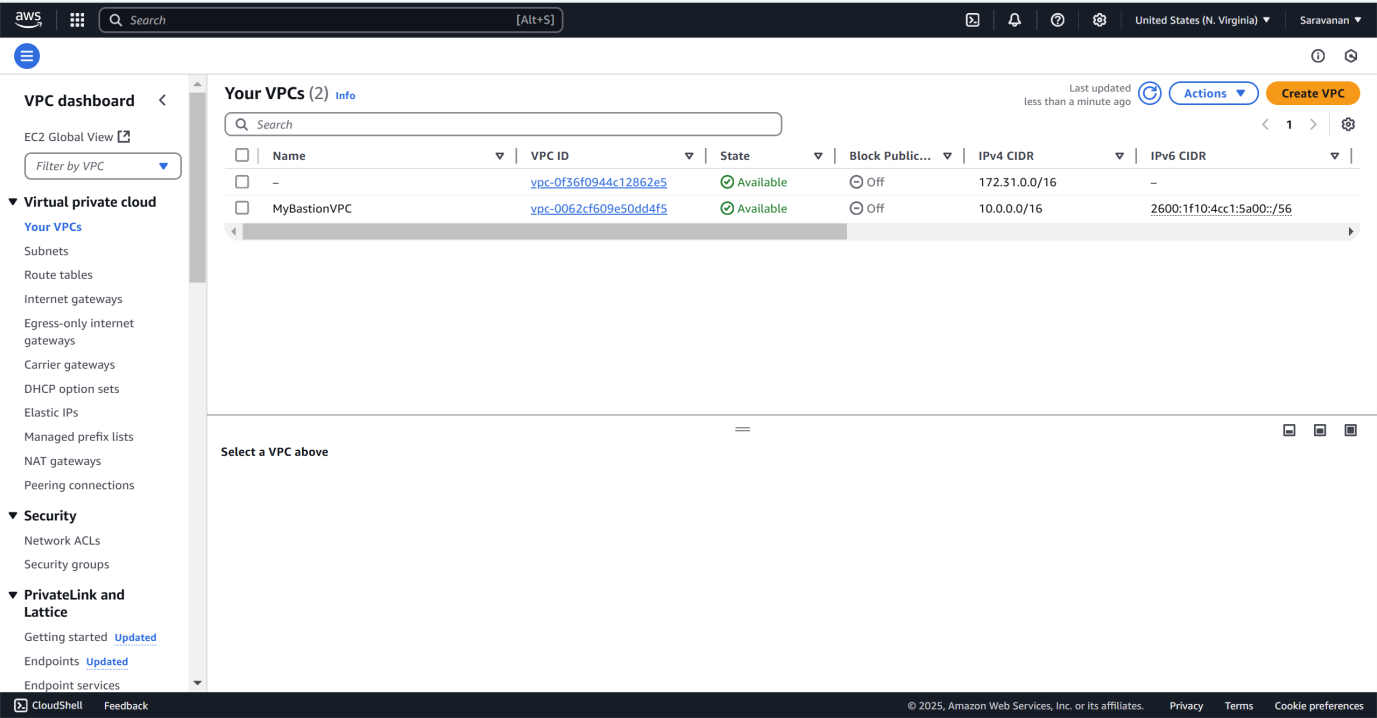
## Step 2:

On the VPC dashboard , click on create VPC

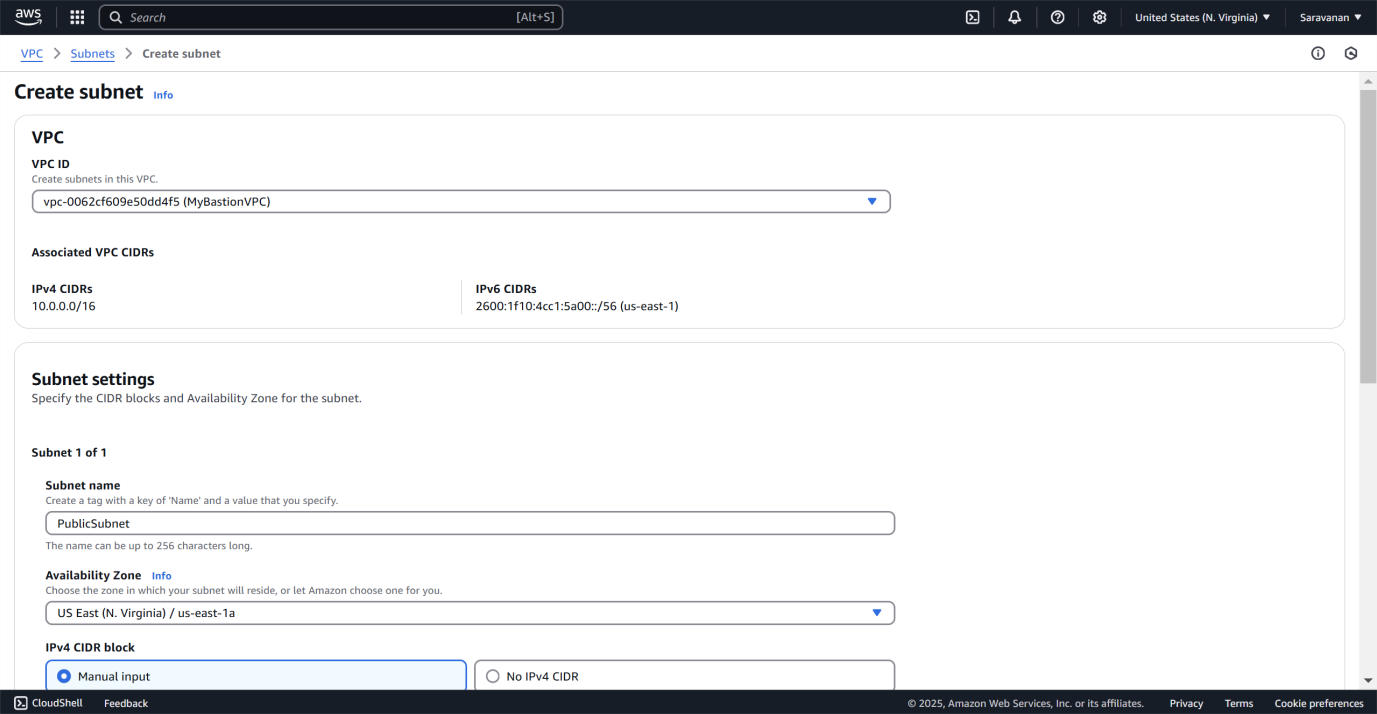


.

IN the VPC setting change VPC only,name it, then IPv4 as usual and change IPv6 by choosing the third radio button,then click create VPC.

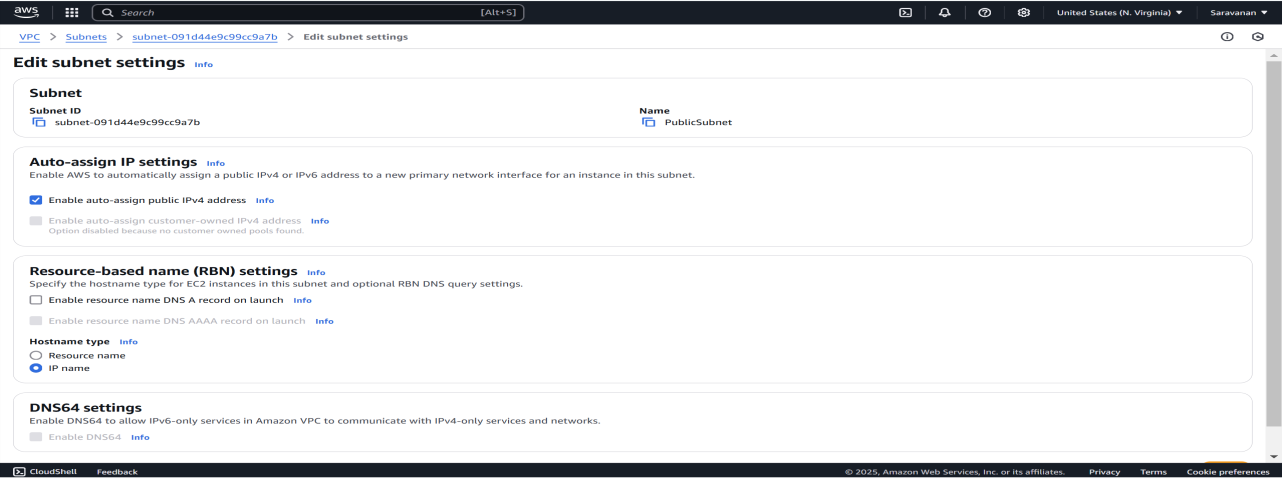


This is same like creating VPC but we crete subnet with IPv4 as 10.0.0.0/24 and the more prior thing is to match to the created VPC.



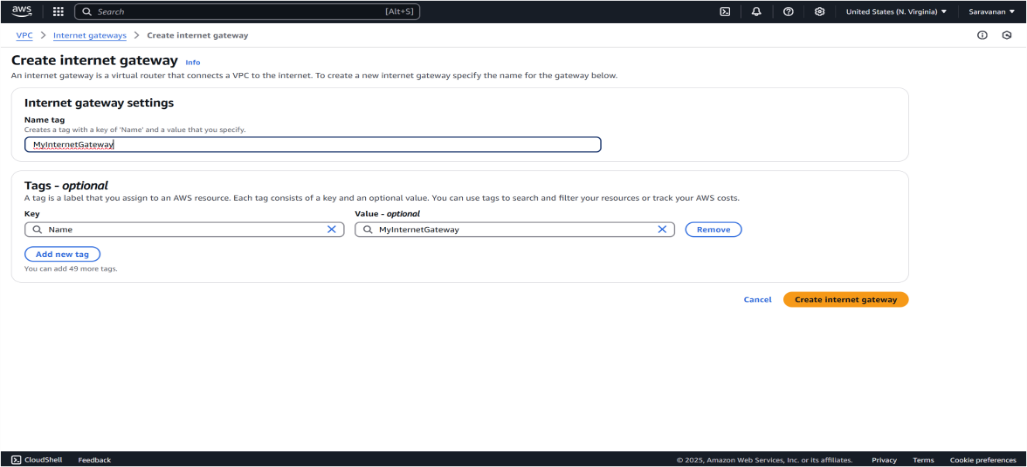
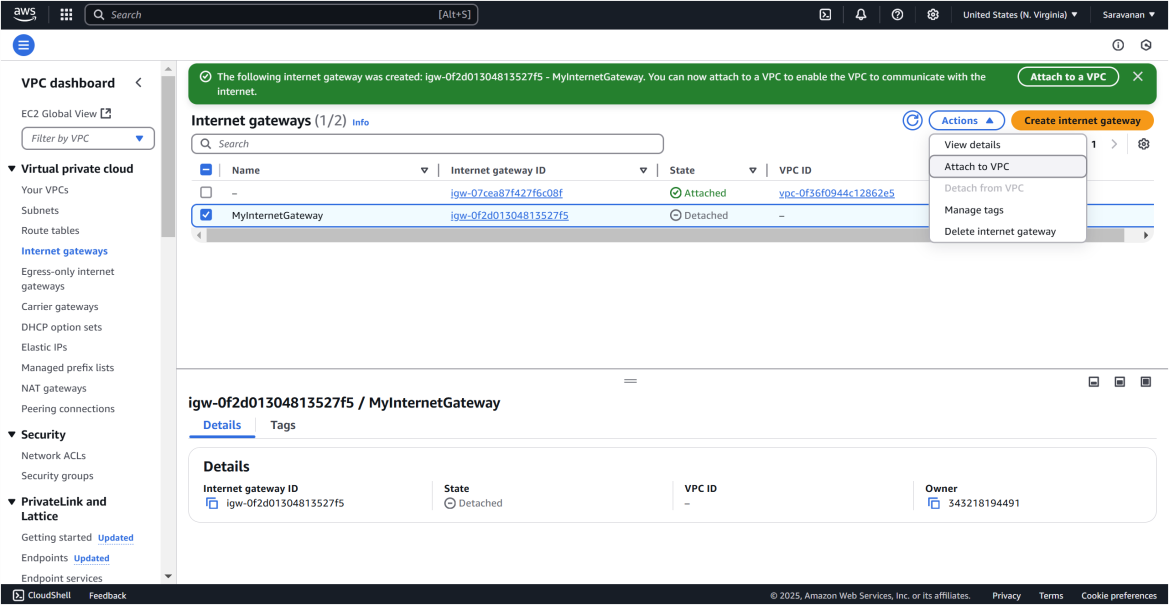
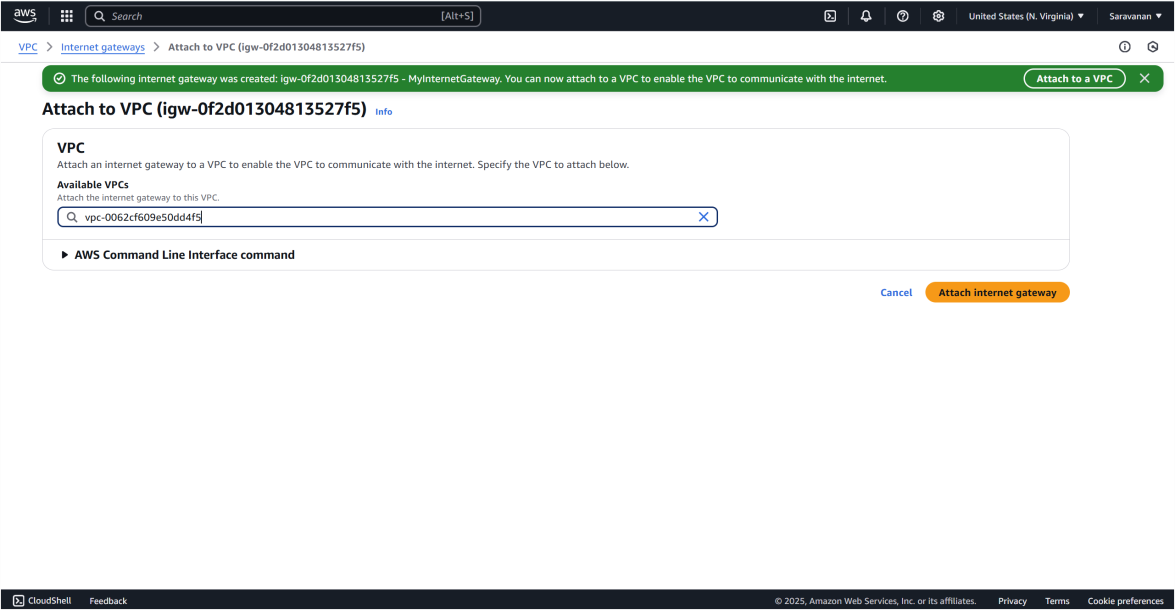
## Step 5:

click **Actions → Modify**, check **Enable auto-assign public IPv4 address**, and click **Save**.

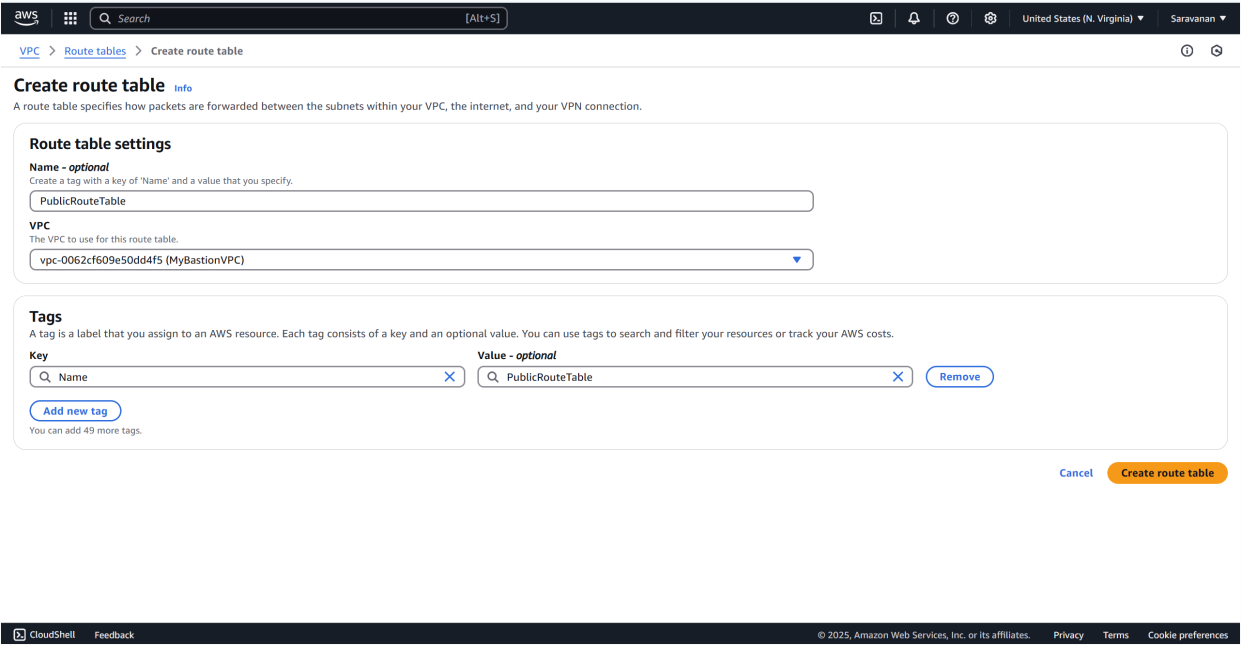
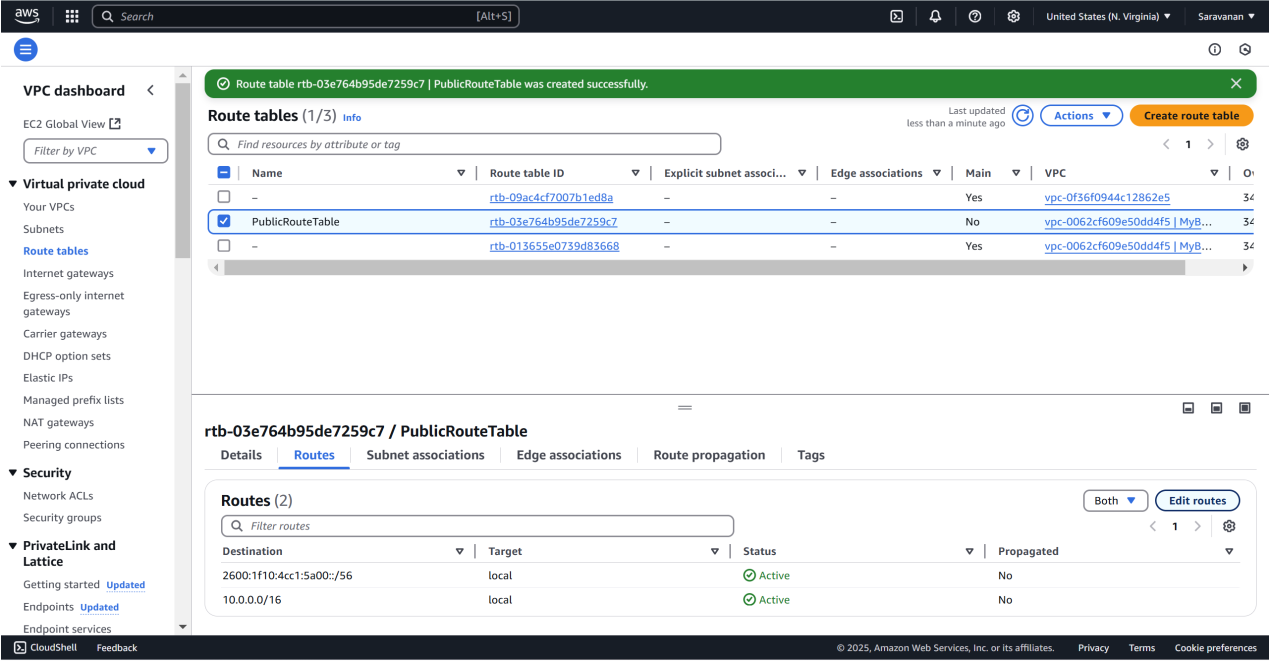


## Step 7:

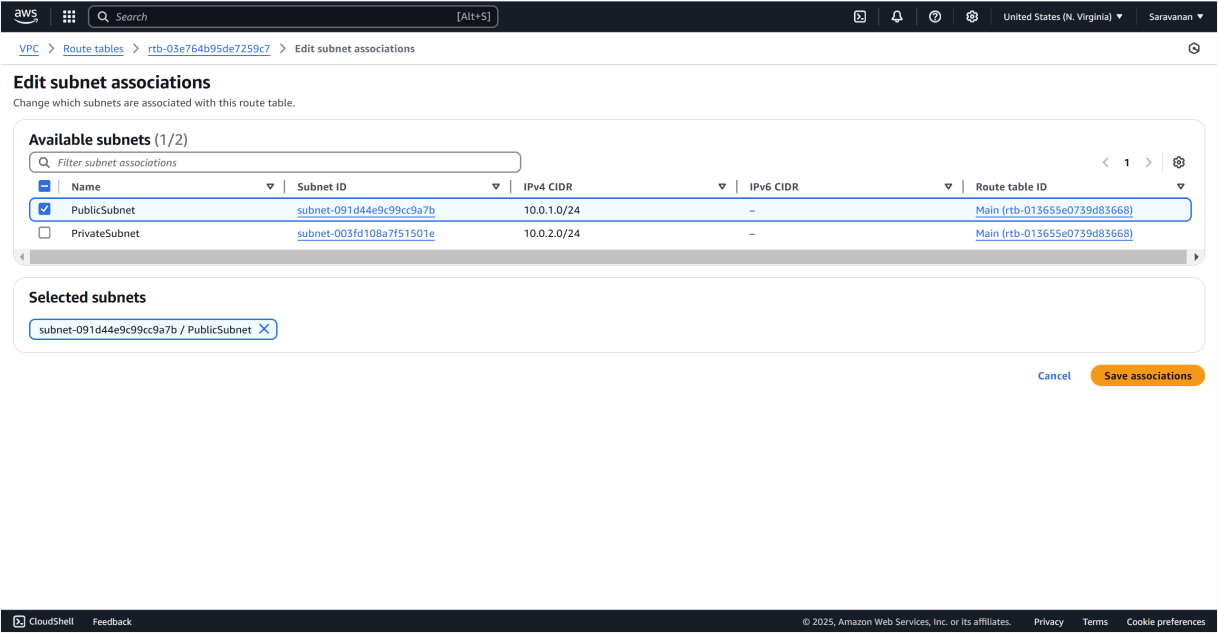
Select internet gateway in the VPC dashboard and and click on create internet gateway.

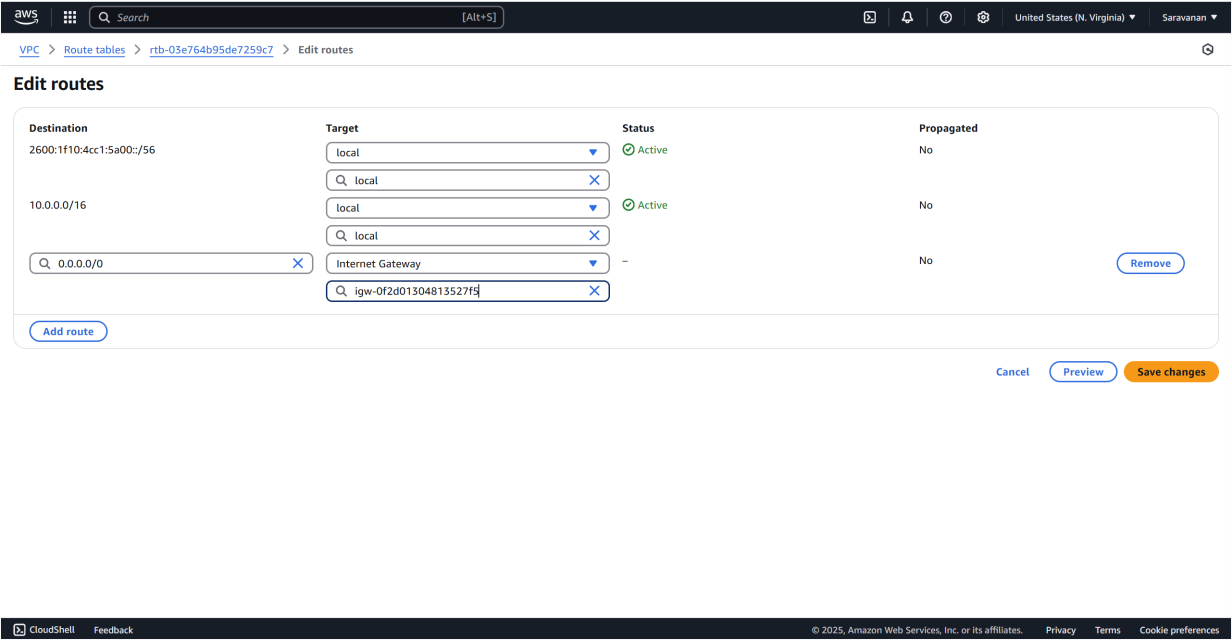


For Route table just enter the name of the routr table and in the VPC box , click or select the vpc you created.

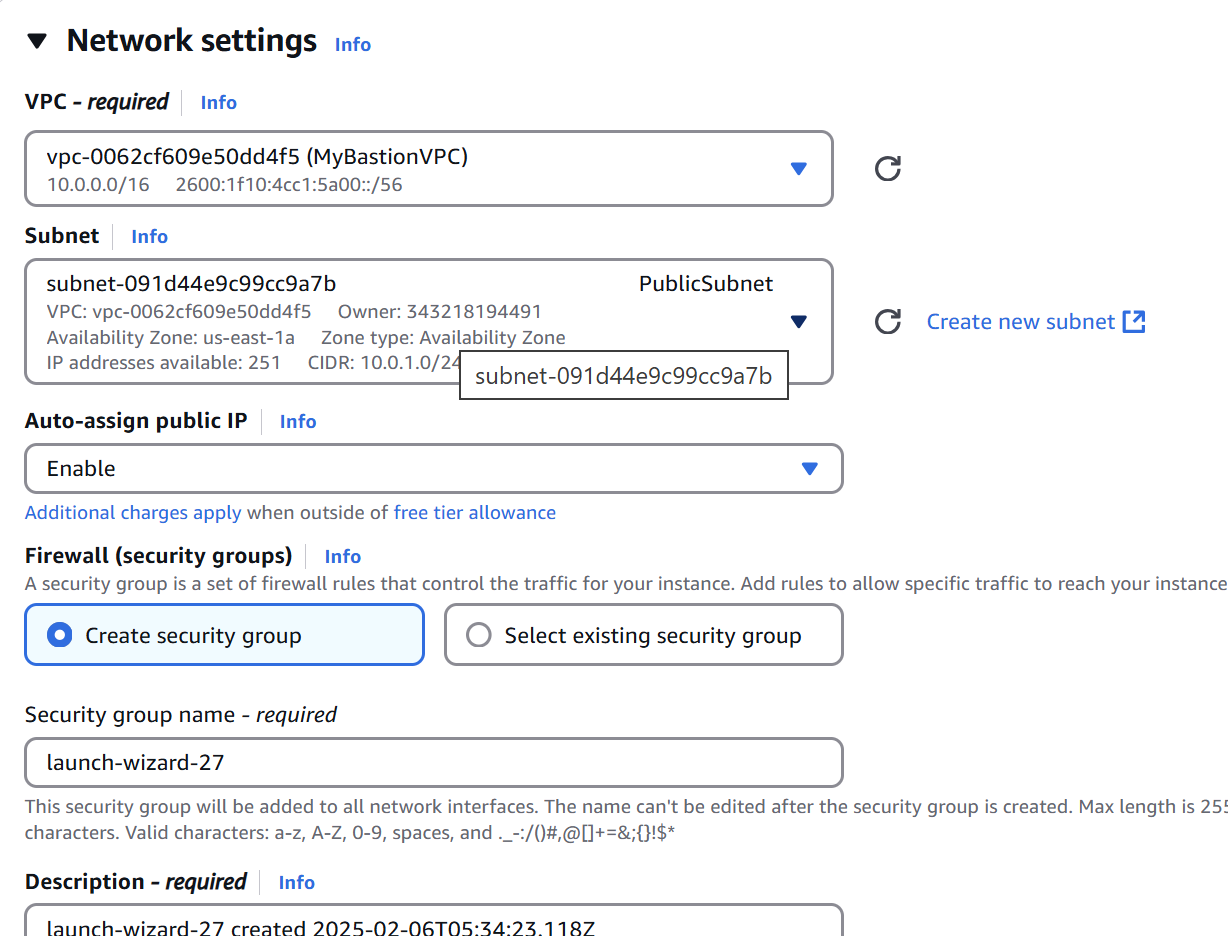


Next, go to the **Subnet associations** tab of *c*, click **Edit created route table** , check the box for *created subnet*, and click **Save associations**.

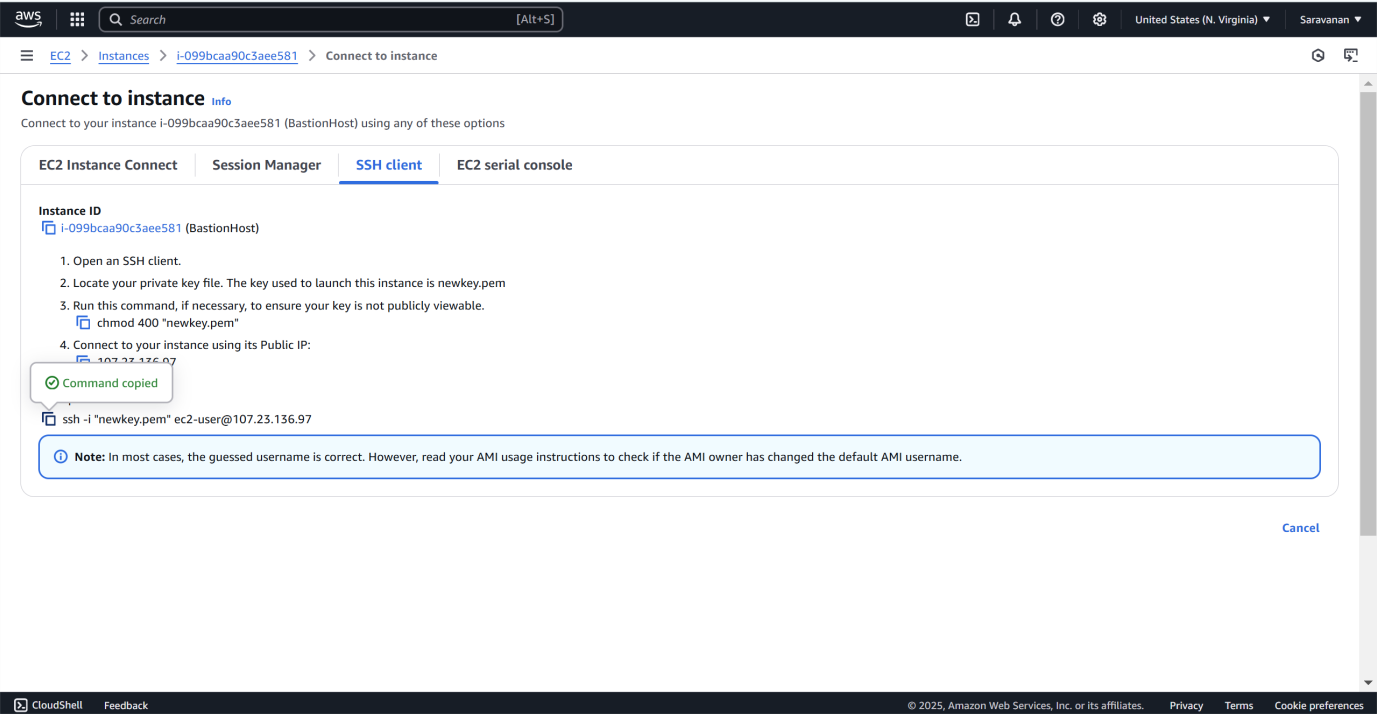




This is simply launching EC2 instance and get modifying or editing the VPc and Subnet you created

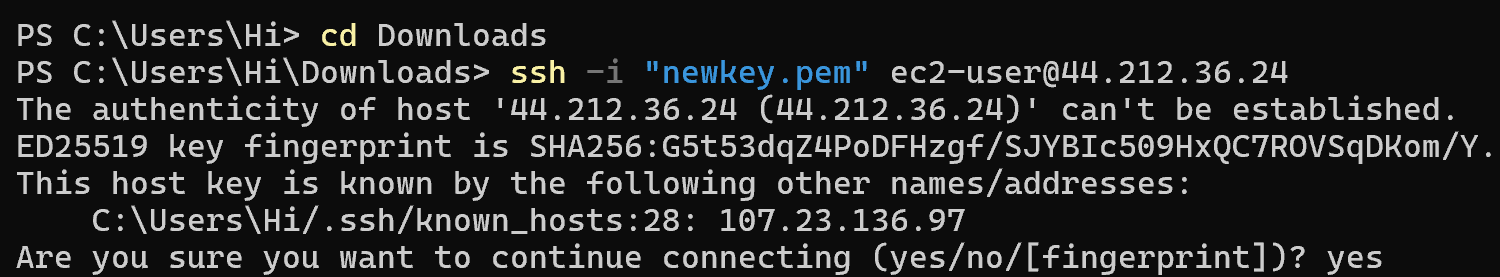


Connect with your PowerShell terminal by copying the ssh command in the SSH client of the *created VPC*

**

## Step 11:

Paste the command copied in the SSH client and connect it by using your key pair.



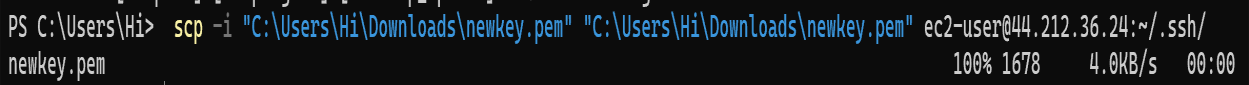
While connected to the bastion host, run this command to create a .ssh folder:

Screenshot 2025-02-06 140922.png

## Step 13:

On your local machine, upload the key file to the bastion host

**scp -i /path/to/your-key.pem /path/to/your-key.pem ec2- user@<BastionHost-Public-IP>:~/.ssh/**

****

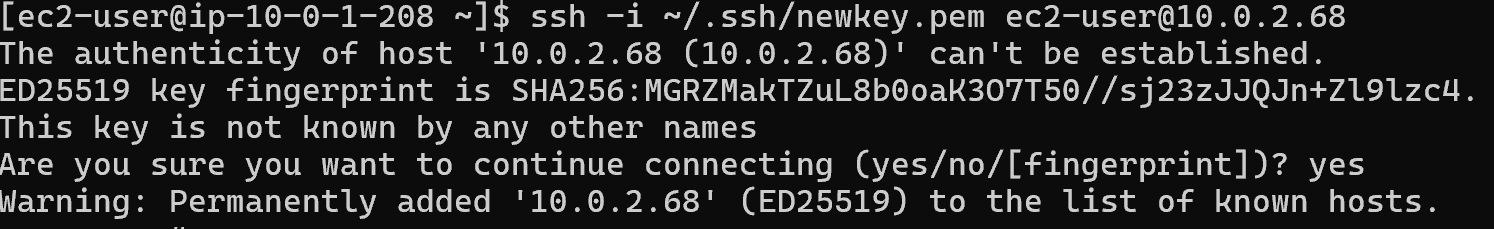
## Step 14:

On the bastion host, run the following command to secure the key:

Screenshot 2025-02-06 141004.png

## Step 15:

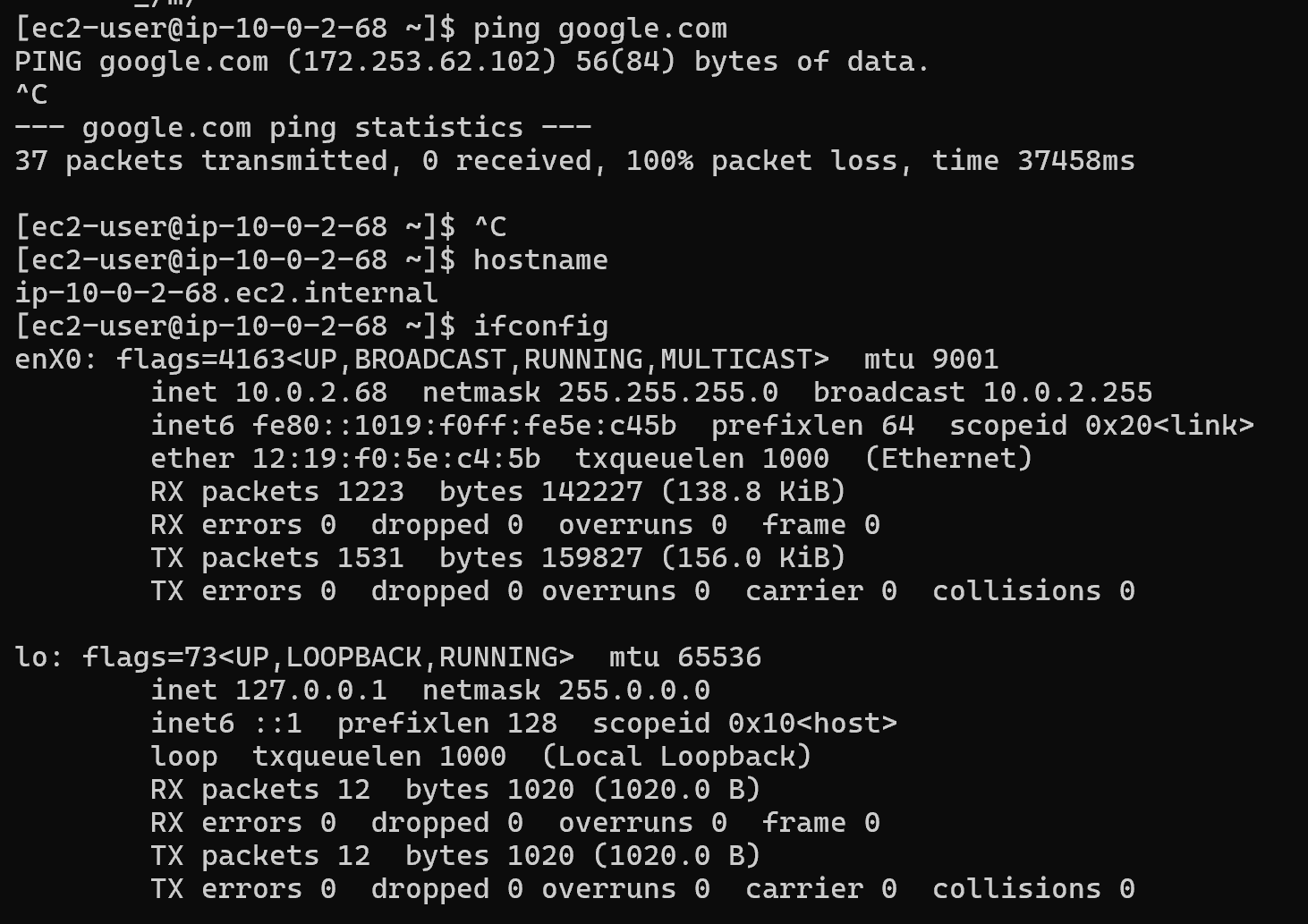
Use the private IP of the private instance (e.g., 10.0.2.x) and run: **ssh -i ~/.ssh/your-key.pem ec2-user@<PrivateInstance-Private- IP>**



## Step 16:

To verify network access and security, follow these steps:

1. **Inspect Instance Details**: Connect to your private instance and run:
   * hostname to check the instance hostname.
   * ifconfig to verify the private IP address.



**Outcome**

* A **functional bastion host** enabling secure access to private instances.
* Improved **security by preventing direct internet access** to private resources.
* Properly **configured security rules** allowing only authorized users to connect.
* **Seamless and efficient management** of private instances through the bastion host.