**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**Secure Access with a Bastion HostSet up a bastion host in a public subnet to securely access instances in a private subnet.**

Name: Paripurani S P Department: IT



# Introduction

A bastion host, deployed in a public subnet of your VPC, serves as a secure gateway to access instances in a private subnet. By launching an EC2 instance in the public subnet and assigning it a public IP or Elastic IP, you can create a security group to allow SSH access only from trusted IP addresses. Once set up, use the bastion host to connect securely to instances in the private subnet, ensuring controlled and protected access.

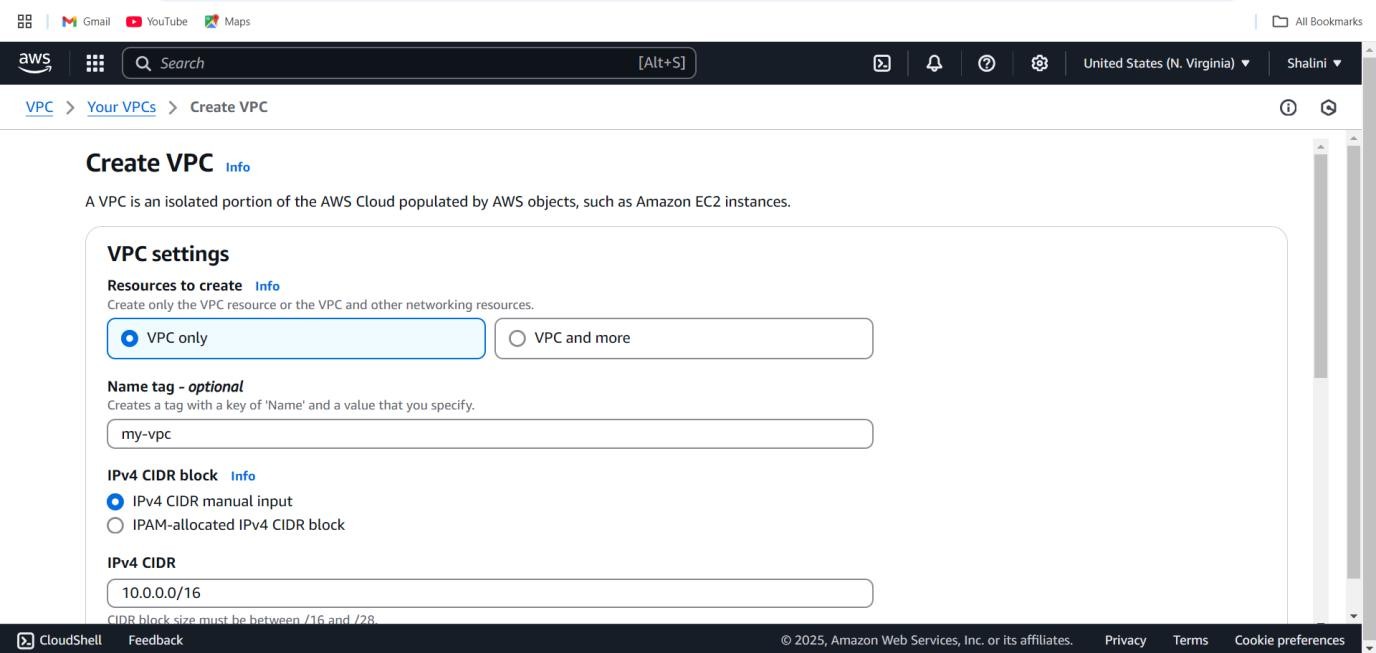
# Overview

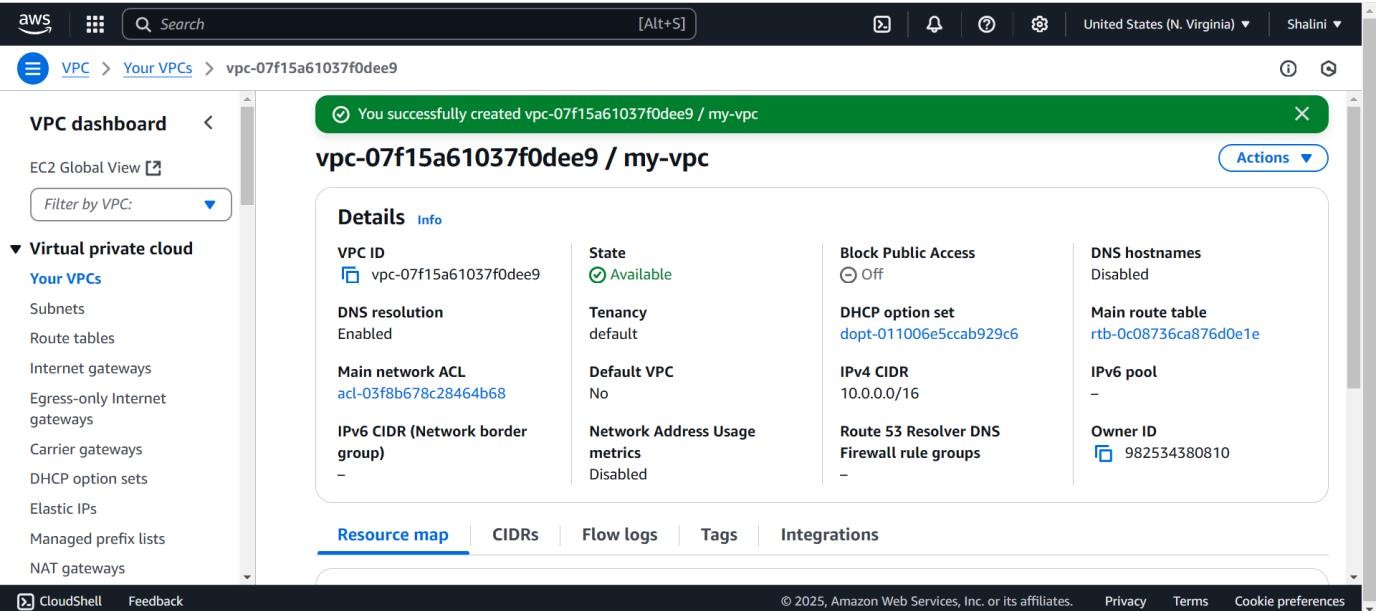
We will set up a **Bastion Host** in a **public subnet** that provides controlled SSH access to instances inside a **private subnet**.

# Step 1:

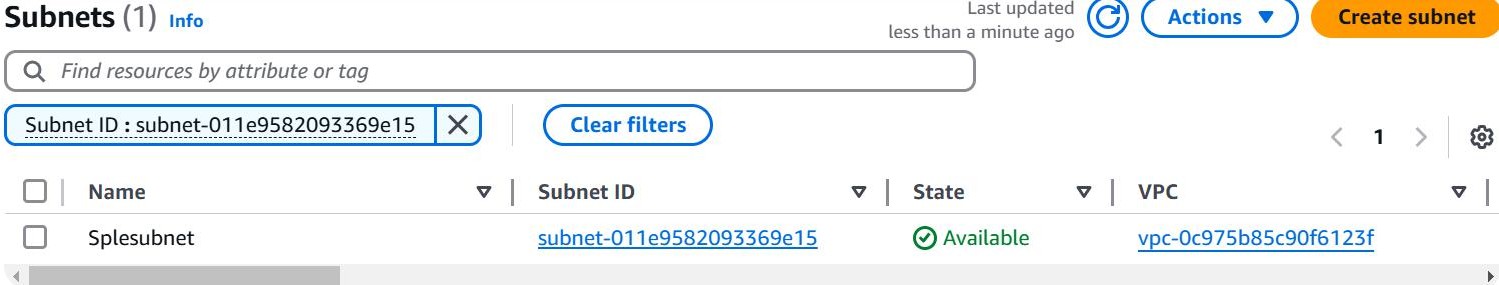
**Create a VPC with Public and Private Subnets**

* 1. Create a VPC
     + Go to AWS Console → VPC Dashboard.
     + Click Create VPC and name it MyVPC.
     + Set IPv4 CIDR Block: 10.0.0.0/16.
     + Click Create VPC.



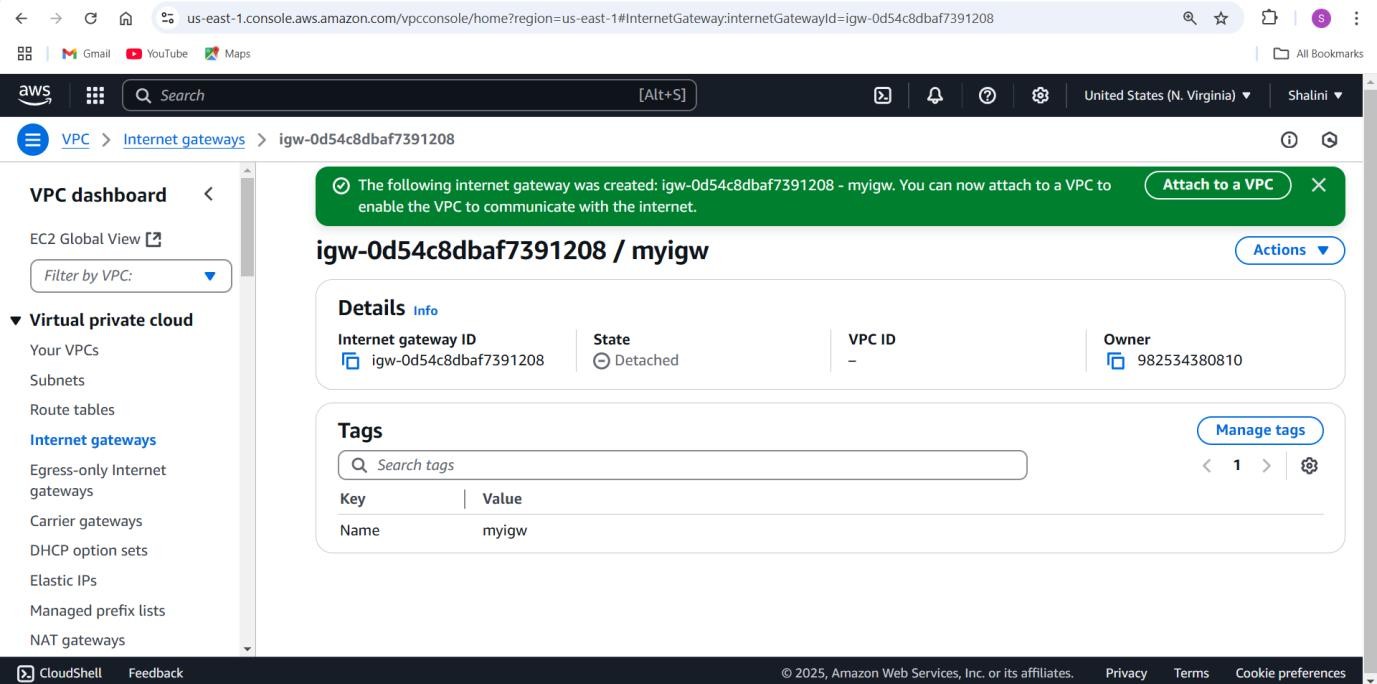


* 1. **Create a Public Subnet**
     + Go to **Subnets** → **Create Subnet**.
     + Select **MyVPC** and set CIDR block 10.0.1.0/24.
     + Enable **Auto-Assign Public IP**.
  2. **Create a Private Subnet**
     + Repeat the same process, but use CIDR block 10.0.2.0/24.
     + **Do not enable** Auto-Assign Public IP.

**Step 2:**

**Configure Public Subnet for Internet Access**

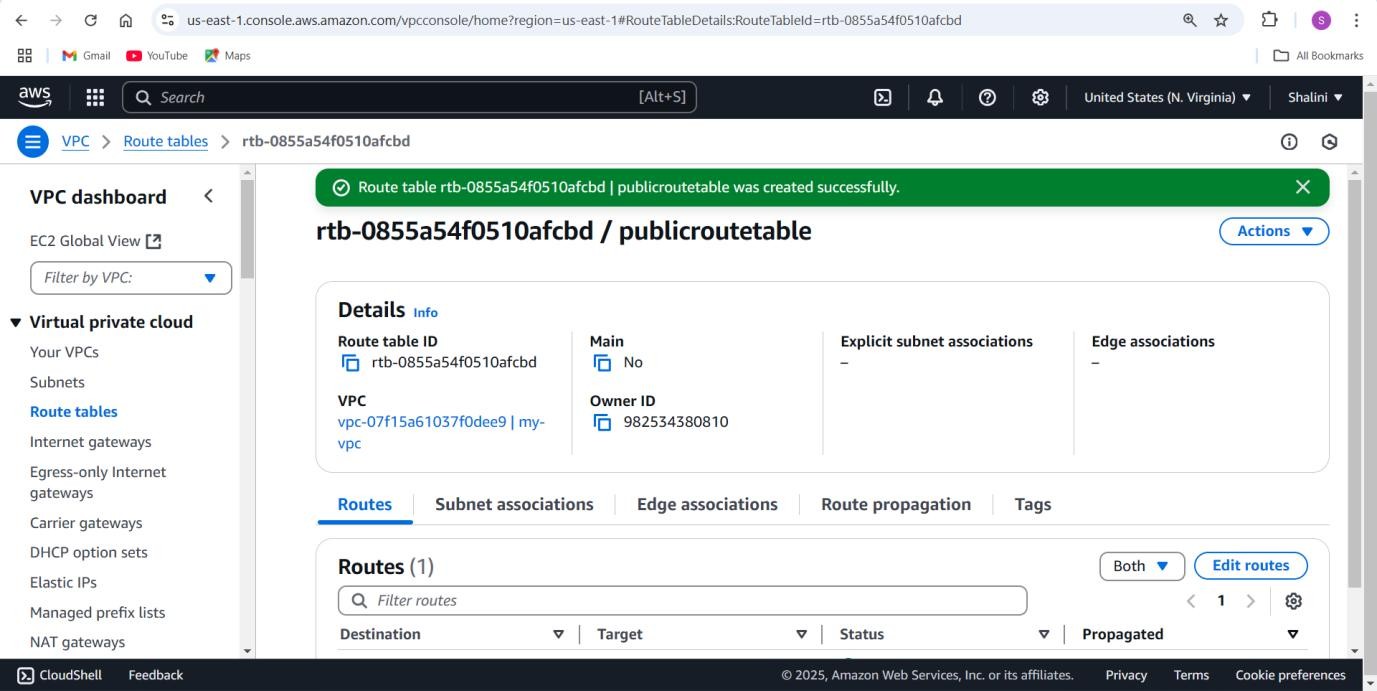
* 1. **Create an Internet Gateway (IGW)**
     + Go to **Internet Gateways** → Click **Create Internet Gateway**.
     + Name it **MyIGW**, attach it to **MyVPC**.



* 1. **Update Public Route Table**
     + Go to **Route Tables** → **Create Route Table** → Name it

**PublicRouteTable**.

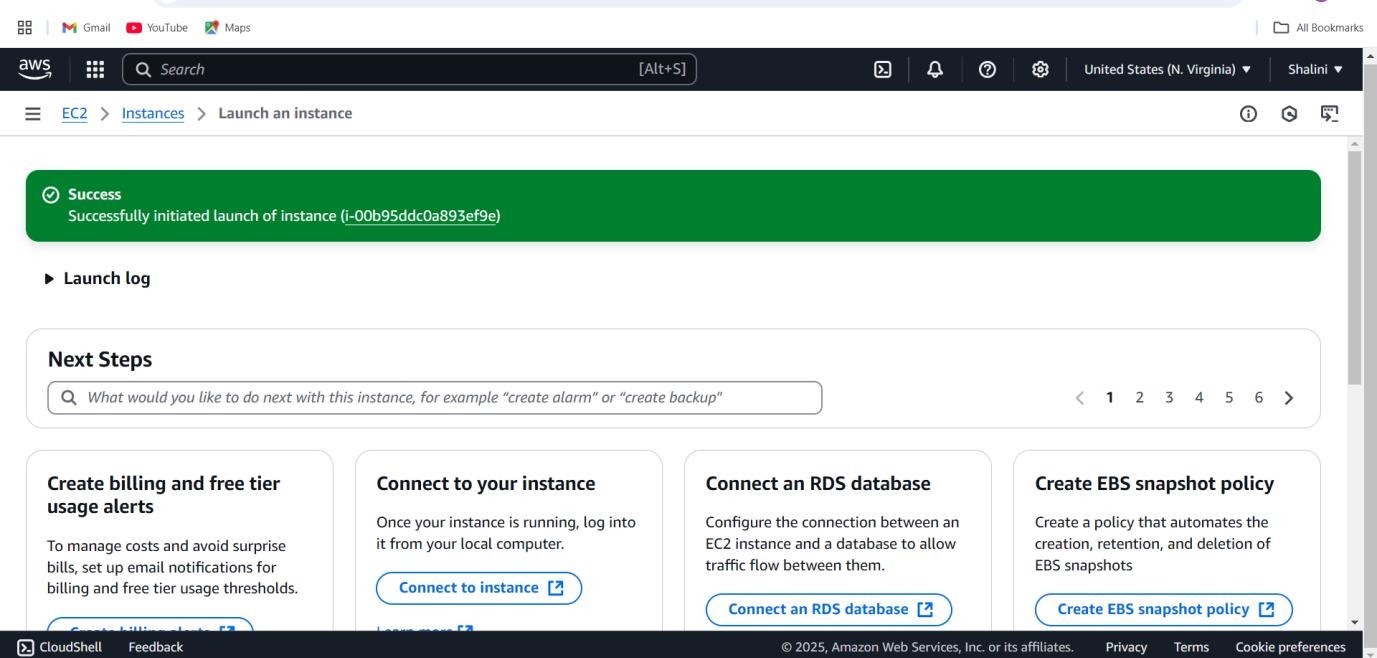
* + - Associate it with **PublicSubnet**.
    - Add a route:
      * **Destination:** 0.0.0.0/0
      * **Target: Internet Gateway (MyIGW)**



**Step 3:**

**Launch a Bastion Host (Public Subnet)**

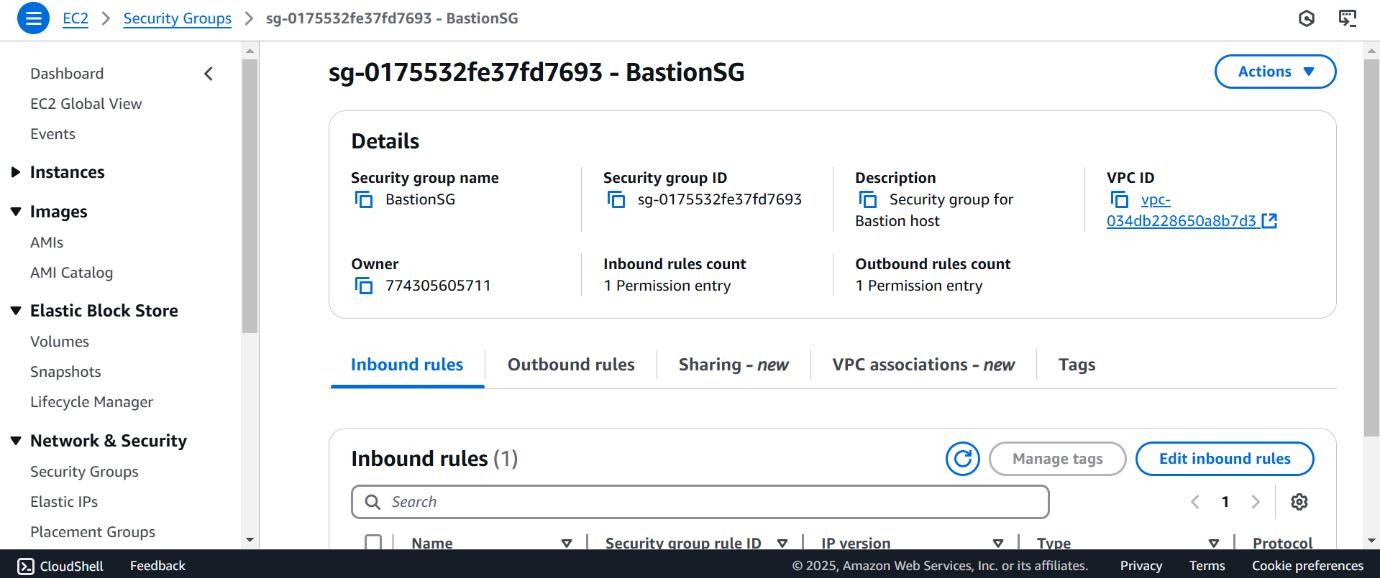
1. Go to **EC2 Dashboard** → **Launch Instance**.
2. Select **Amazon Linux 2** (or **Ubuntu**).
3. Choose **t2.micro (Free Tier Eligible)**.
4. Place it in **PublicSubnet** with **Auto-Assign Public IP enabled**.
5. Create a **Security Group (BastionSG)**:
   * Allow **SSH (Port 22) from Your IP** (xx.xx.xx.xx/32).
6. Create or use an **existing key pair** (e.g., bastion-key.pem).
7. Click **Launch**.



**Step 4:**

**Launch a Private EC2 Instance**

1. Go to **EC2 Dashboard** → **Launch Instance**.
2. Choose **Amazon Linux 2** (or **Ubuntu**).
3. Choose **t2.micro** and place it in **PrivateSubnet**.
4. **Disable Auto-Assign Public IP**.
5. Create a **Security Group (PrivateSG)**:
   * Allow **SSH (Port 22) only from Bastion Host’s Security Group**.
6. Use the same **key pair** (bastion-key.pem).
7. Click **Launch**.

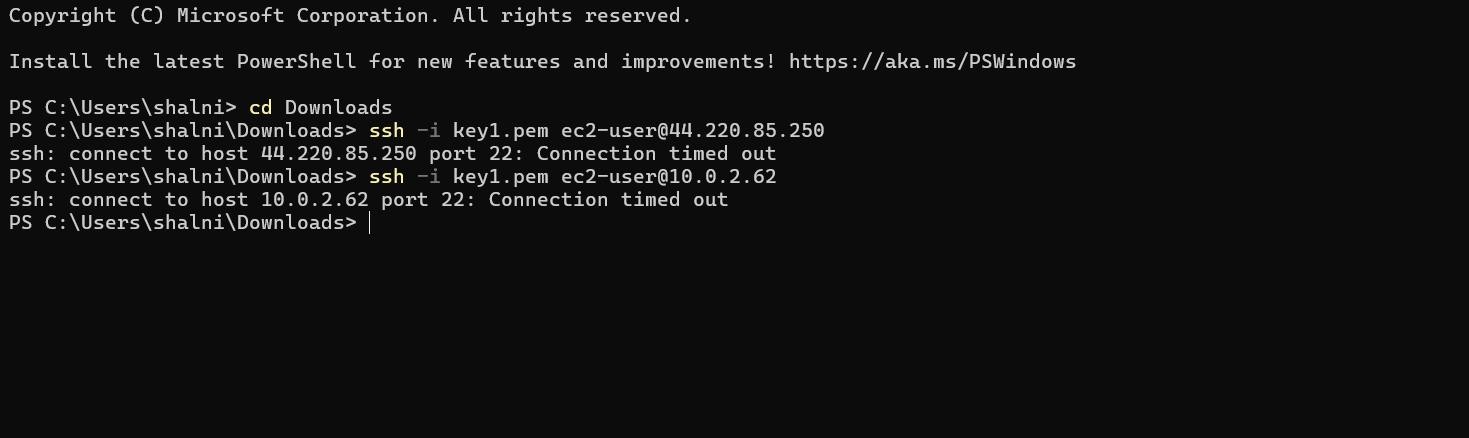


**Step 5: Connect to the Private Instance Using the Bastion Host**

* 1. **Connect to the Bastion Host**

ssh -i bastion-key.pem ec2-user@<bastion-public-ip>

*(Replace <bastion-public-ip> with the actual Bastion Host public IP.)*



* 1. **SSH from Bastion to Private Instance**
     1. Copy the bastion-key.pem file to the Bastion Host:

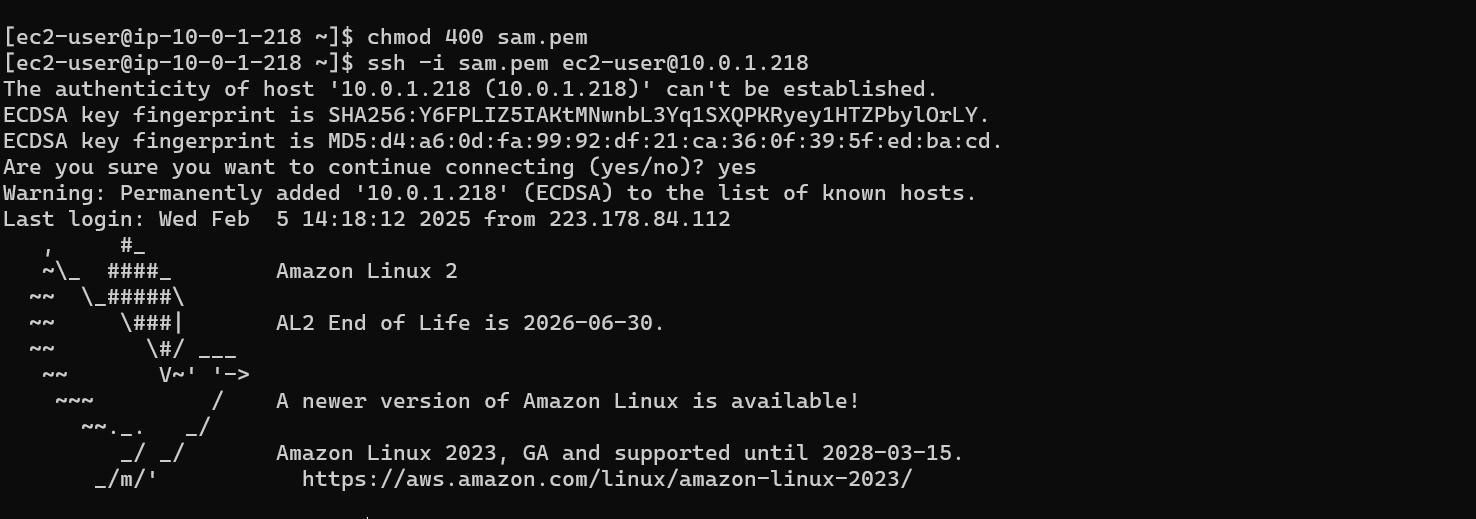
scp -i bastion-key.pem bastion-key.pem ec2-user@<bastion-public- ip>:~/

* + 1. Connect to the Bastion Host:

ssh -i bastion-key.pem ec2-user@<bastion-public-ip>

* + 1. Change permissions for the key file: chmod 400 bastion-key.pem
    2. SSH into the Private Instance from the Bastion Host: ssh -i bastion-key.pem ec2-user@<private-instance-ip>

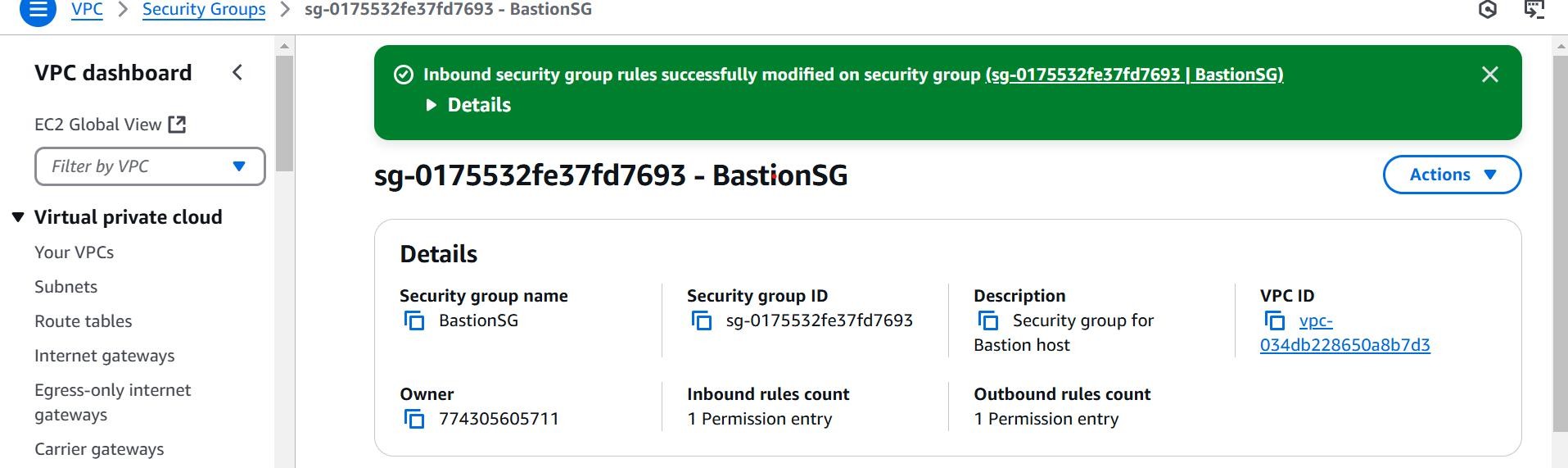
*(Replace <private-instance-ip> with the private IP of your instance.)*



**Step 6: Secure Your Bastion Host**

* 1. **Restrict SSH Access**
     + **Go to Security Group (BastionSG)** → Edit Inbound Rules.
     + **Allow SSH only from your IP address (xx.xx.xx.xx/32)**

instead of allowing all (0.0.0.0/0)

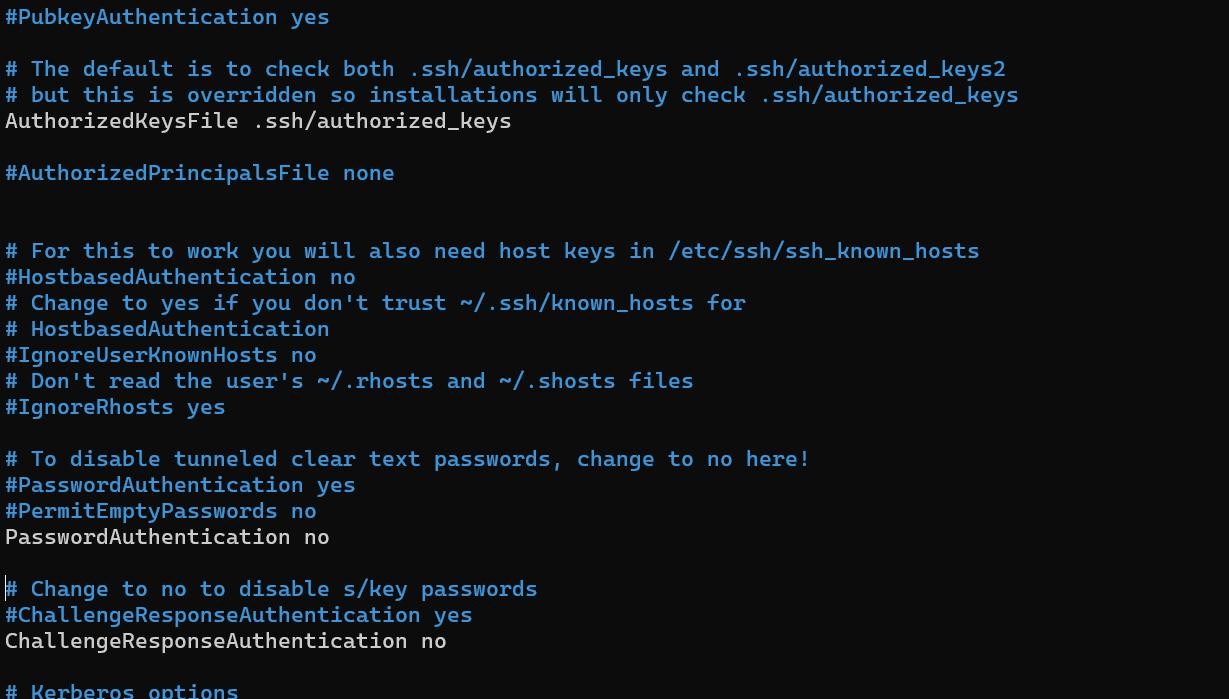


* 1. **Disable Password Authentication**

1. Edit SSH config:

sudo nano /etc/ssh/sshd\_config

1. Find and update these lines: PasswordAuthentication no PermitRootLogin no
2. Restart SSH service: sudo systemctl restart sshd



**Step 7:**

**Alternative - Use AWS Systems Manager (SSM) Instead of SSH**

1. **Attach SSM Managed Policy to EC2 IAM Role**

(AmazonSSMManagedInstanceCore).

1. **Enable SSM Agent** (Pre-installed on Amazon Linux & Ubuntu).
2. Use **AWS Systems Manager > Session Manager** to connect to instances without SSH.

**Conclusion**

Setting up a bastion host in a public subnet to securely access instances in a private subnet is an effective way to enhance security within your network infrastructure. By funneling all administrative access through a single, secure entry point, you ensure that your private instances remain shielded from direct exposure to the internet, reducing the risk of unauthorized access. The bastion host acts as a gatekeeper, allowing you to manage your resources in a controlled and monitored manner, thereby fortifying your overall security posture.