A white and blue logo

AI-generated content may be incorrect. A logo with a smile

AI-generated content may be incorrect.

# **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: Kirthika S Department: CSE

A black and white logo

AI-generated content may be incorrect.

**Introduction**

In cloud environments, securing access to private instances is crucial. A **Bastion Host** (or Jump Box) is a special-purpose instance that acts as a secure gateway to access EC2 instances in a private subnet. Instead of exposing private instances directly to the internet, users connect to the Bastion Host first and then access the private instances from there.

This setup **enhances security** by limiting direct SSH access to private instances and applying strict security controls.

**Overview**

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

**What We Will Do?**

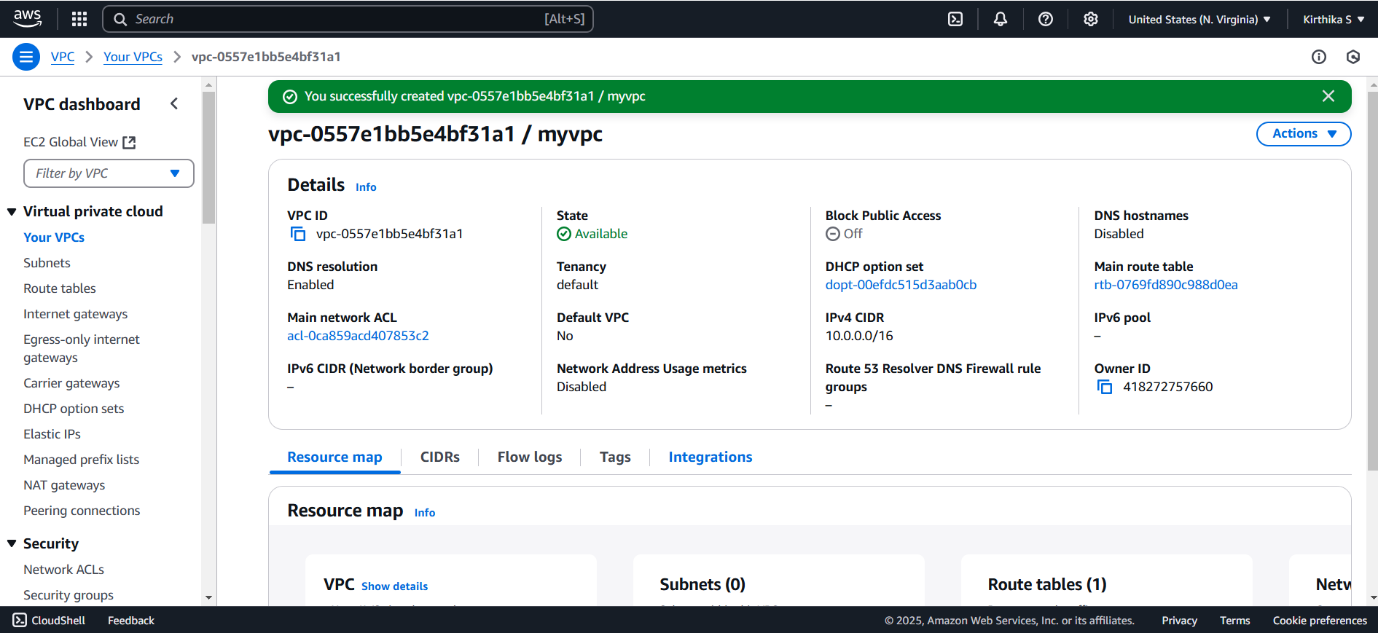
1. **Create a VPC** with a **Public and Private Subnet**.
2. **Set Up a Bastion Host** in the Public Subnet.
3. **Launch a Private EC2 Instance** in the Private Subnet.
4. **Configure Secure SSH Access** via the Bastion Host.
5. **Enhance Security** by restricting SSH access and considering AWS Systems Manager as an alternative.

**Step 1:**

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

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.

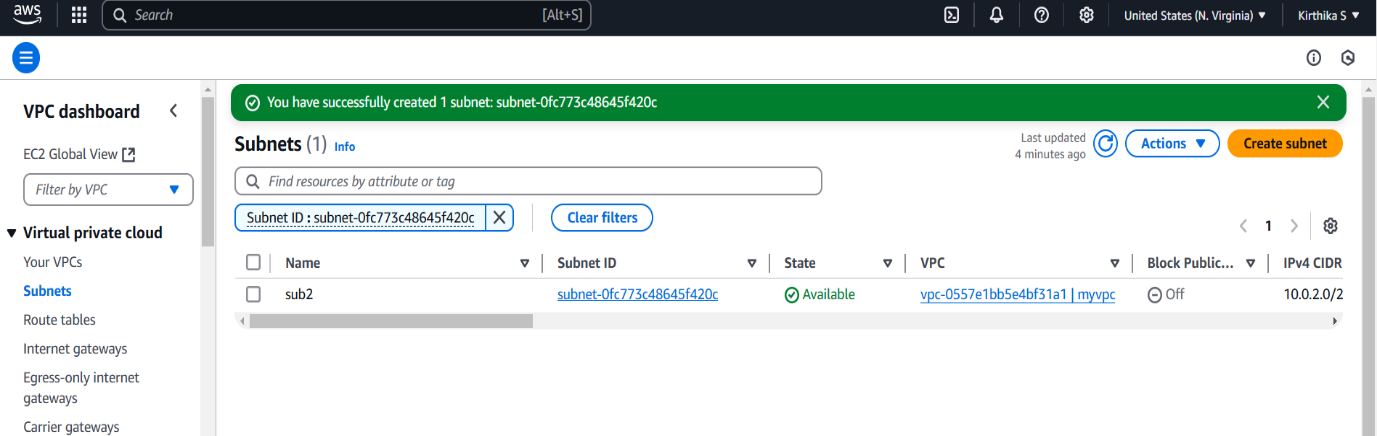


**Create a Public Subnet**

* Go to **Subnets** → **Create Subnet**.
* Select **MyVPC** and set CIDR block 10.0.1.0/24.
* Enable **Auto-Assign Public IP**.

**Create a Private Subnet**

* Repeat the same process, but use CIDR block 10.0.2.0/24.
* **Do not enable** Auto-Assign Public IP.

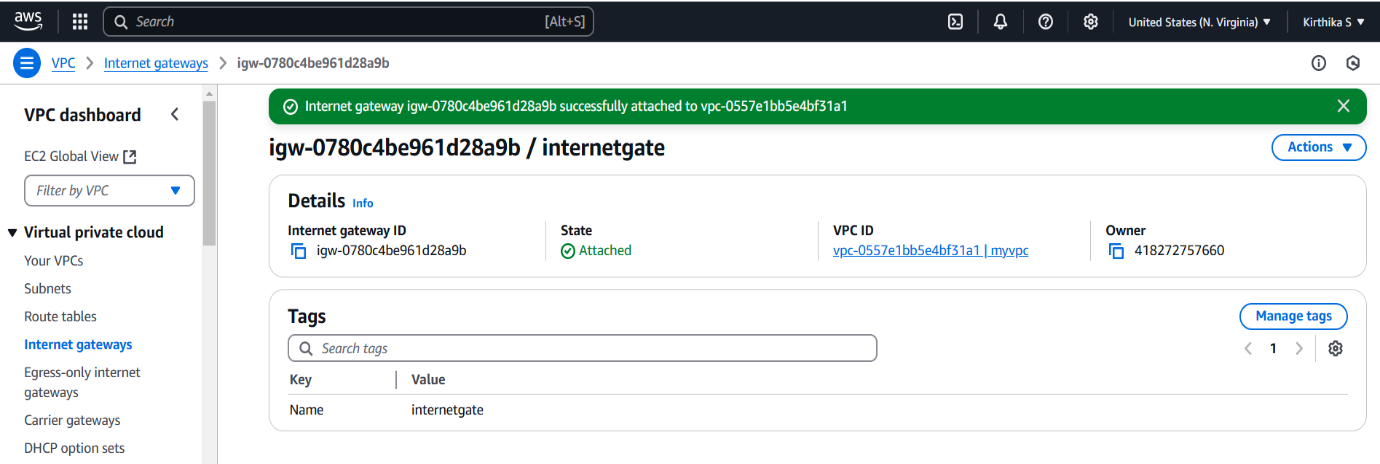


**Step 2:**

**Configure Public Subnet for Internet Access**

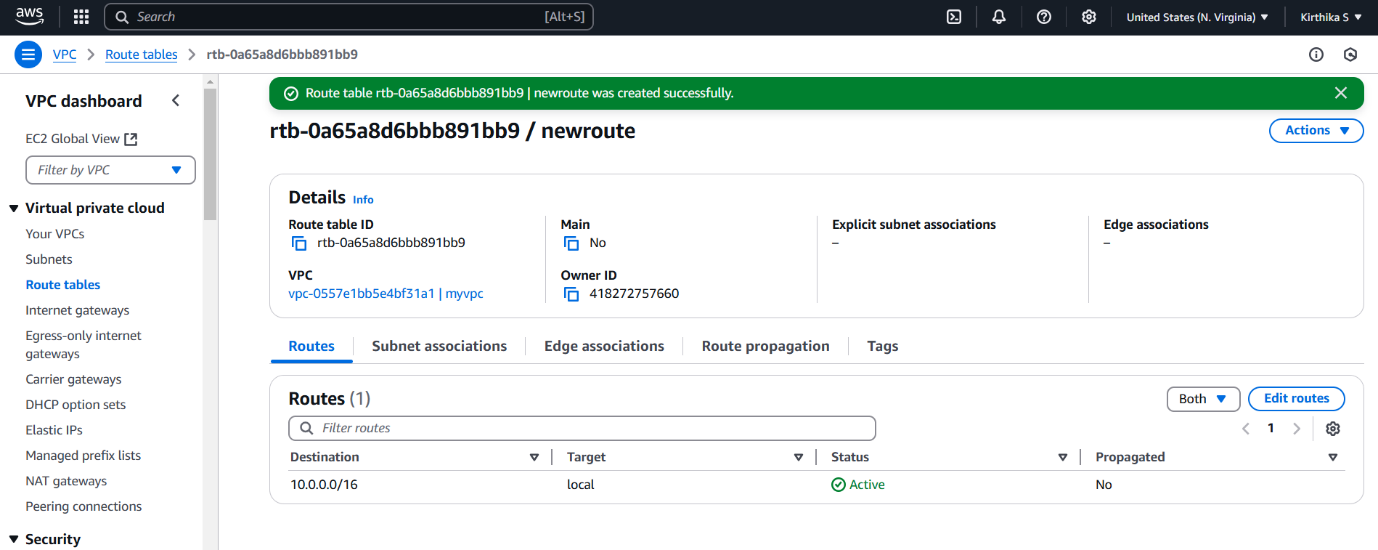
**Create an Internet Gateway (IGW)**

* Go to **Internet Gateways** → Click **Create Internet Gateway**.
* Name it **MyIGW**, attach it to **MyVPC**.



**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:**

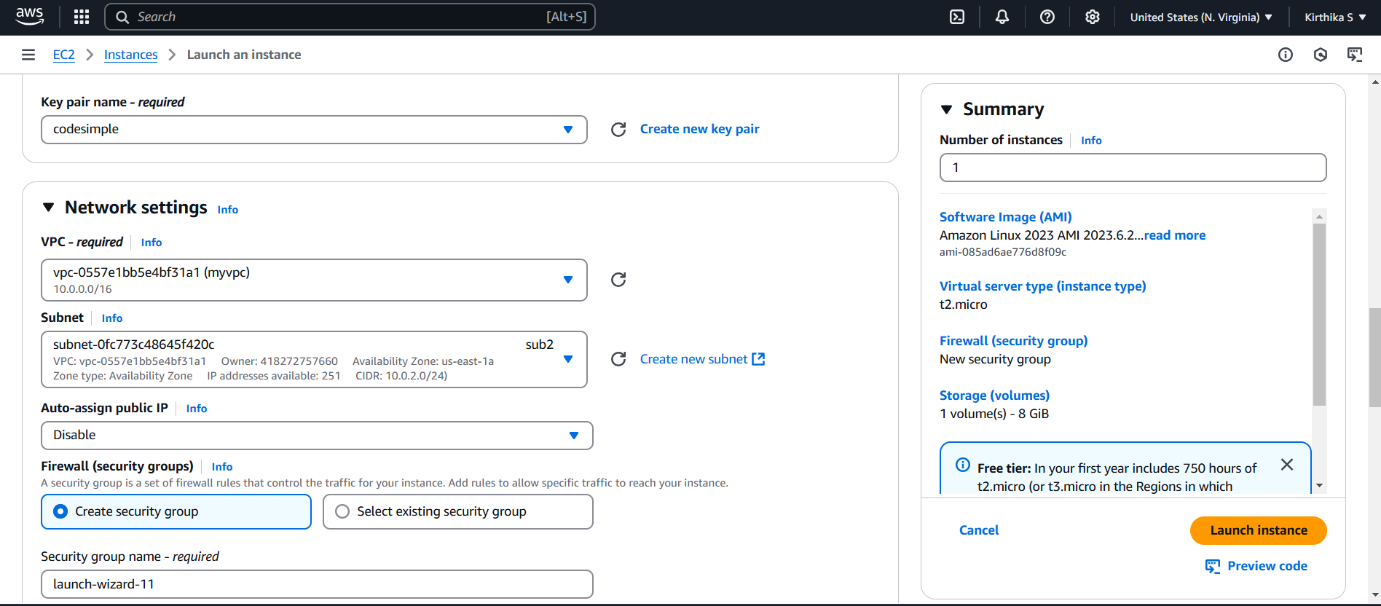
**Launch a Bastion Host (Public Subnet)**

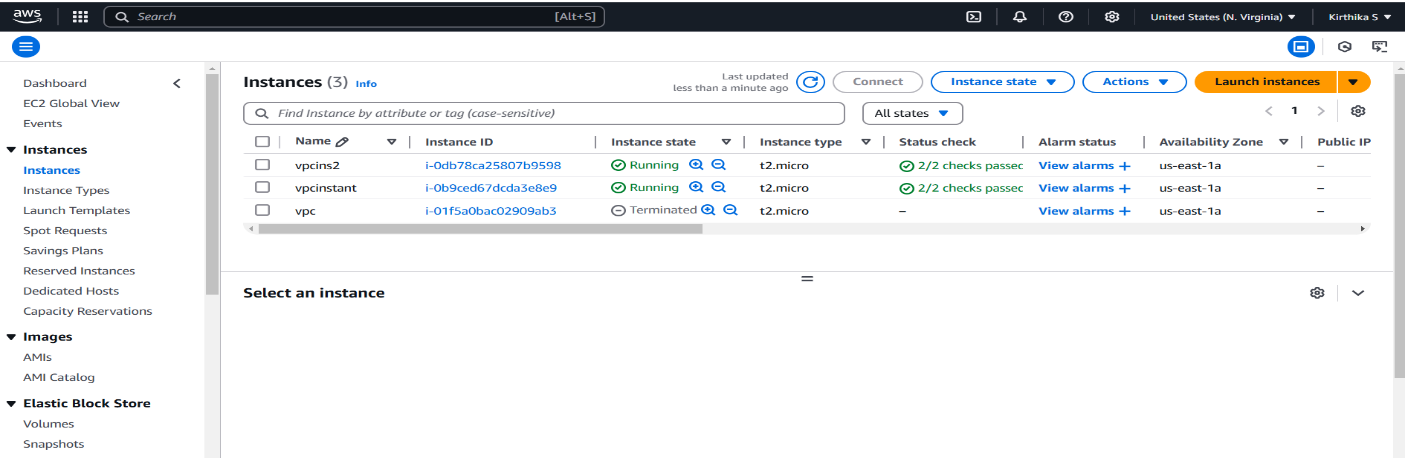
1. Go to **EC2 Dashboard** → **Launch Instance**.
2. Select **Amazon Linux 2** (or **Ubuntu**).
3. Choose **t2.micro (Free Tier Eligible)**.
4. Place it in **PublicSubnet** with **Auto-Assign Public IP enabled**.
5. Create a **Security Group (BastionSG)**:
   * Allow **SSH (Port 22) from Your IP** (xx.xx.xx.xx/32).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



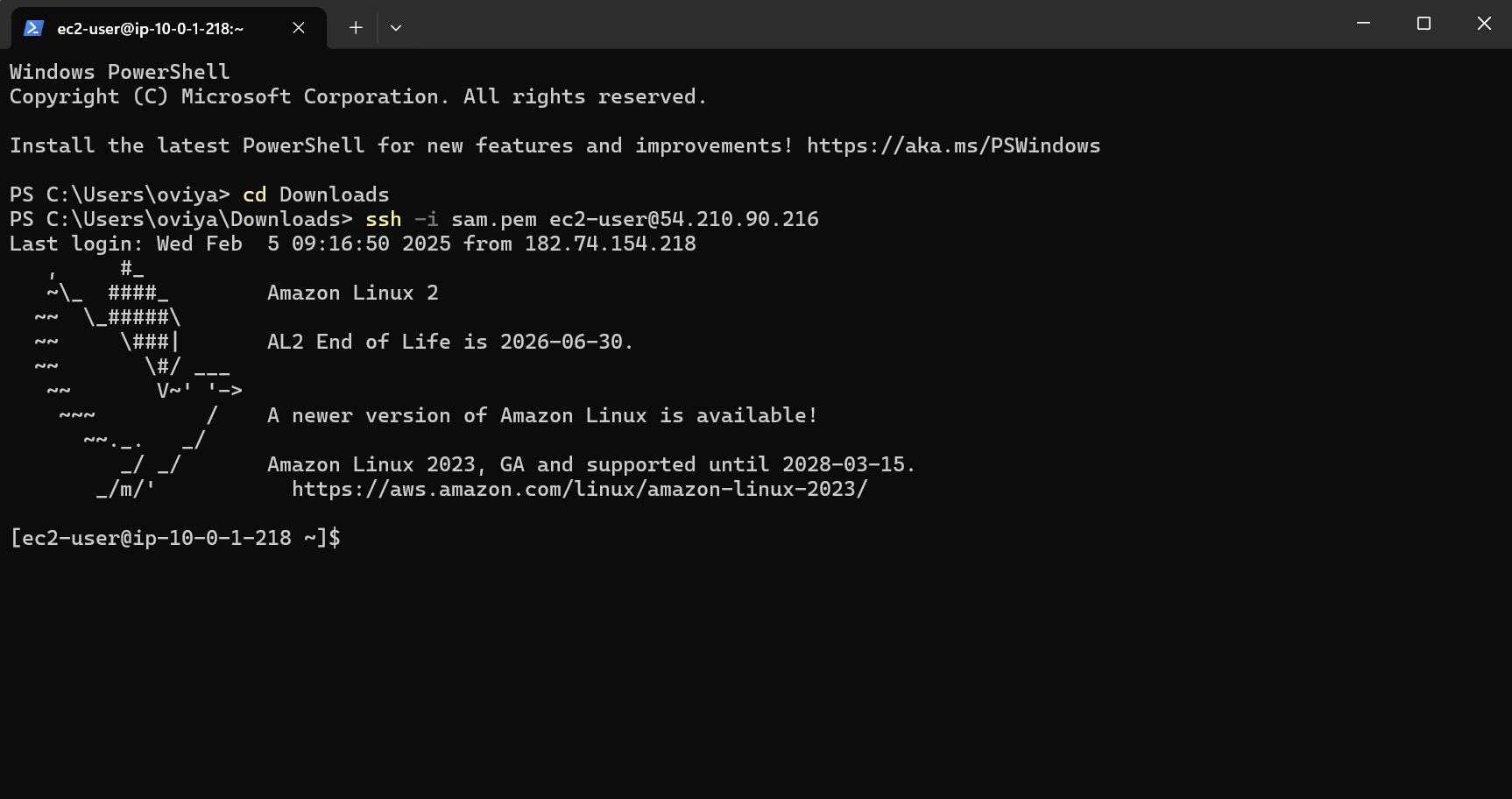


**Step 5: Connect to the Private Instance Using the Bastion Host**

**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.)*



**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

1. 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.)*

A screenshot of a computer

AI-generated content may be incorrect.

**Step 6: Secure Your Bastion Host**

**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)

**Disable Password Authentication**

1. Edit SSH config:

sudo nano /etc/ssh/sshd\_config

1. Find and update these lines:

PasswordAuthentication no

PermitRootLogin no

1. Restart SSH service:

sudo systemctl restart ssh

A screenshot of a computer

AI-generated content may be incorrect.

**Step 7:**

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

1. **Attach SSM Managed Policy to EC2 IAM Role** (AmazonSSMManagedInstanceCore).
2. **Enable SSM Agent** (Pre-installed on Amazon Linux & Ubuntu).
3. Use **AWS Systems Manager > Session Manager** to connect to instances without SSH.

**Outcome**

By completing this POC of setting up a Bastion Host in AWS, you will:

1. Deploy a bastion host in a public subnet and a private instance in a private subnet for secure access.

2. Enable SSH access to the private instance through the bastion host, ensuring the private instance remains isolated from the internet.

3. Configure security groups to restrict network traffic and enforce access control based on best practices.

4. Verify connectivity and communication between the bastion host and private instance within the VPC.