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 : AAKASH RAMADURAI B Department : MECHANINCAL ENG

A black and white logo

AI-generated content may be incorrect.

**Introduction**

In cloud systems, keeping private servers safe is very important. A Bastion Host (or Jump Box) is a special server that works as a secure entry point to reach private EC2 servers. Instead of allowing direct internet access to these private servers, users first connect to the Bastion Host and then access the private servers from there.

This setup improves security by preventing direct SSH access to private servers and adding strict security rules.

**Overview**

We will deploy a Bastion Host in a public subnet to securely manage SSH access to instances located in 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. Allow secure SSH access through the Bastion Host.

5. Improve security by limiting SSH access and using AWS Systems Manager as an option.

**Step 1:**

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

**1.1 Create a VPC**

* Open the AWS Console and go to the **VPC Dashboard**.
* Click **Create VPC** and name it **MyVPC**.
* Set **IPv4 CIDR Block** to **10.0.0.0/16**.
* Click **Create VPC**.

A screenshot of a computer

Description automatically generated

**1.2 Create a Public Subnet**

* Go to **Subnets** , then click **Create Subnet**.
* Choose MyVPC and set CIDR block to 10.0.1.0/24.
* Enable **Auto-Assign Public IP**.

**1.3 Create a Private Subnet**

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

A screenshot of a computer

Description automatically generated

**Step 2:**

**Configure the Public Subnet for Internet Access**

2.1 Create an Internet **Gateway (IGW)**

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

A screenshot of a computer

Description automatically generated

**2.2 Update the Public Route Table**

* Go to **Route Tables,** Create **Route Table** and then 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)**

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

A screenshot of a computer

AI-generated content may be incorrect.

**Step 4:**

**Launch a Private EC2 Instance**

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

A screenshot of a computer

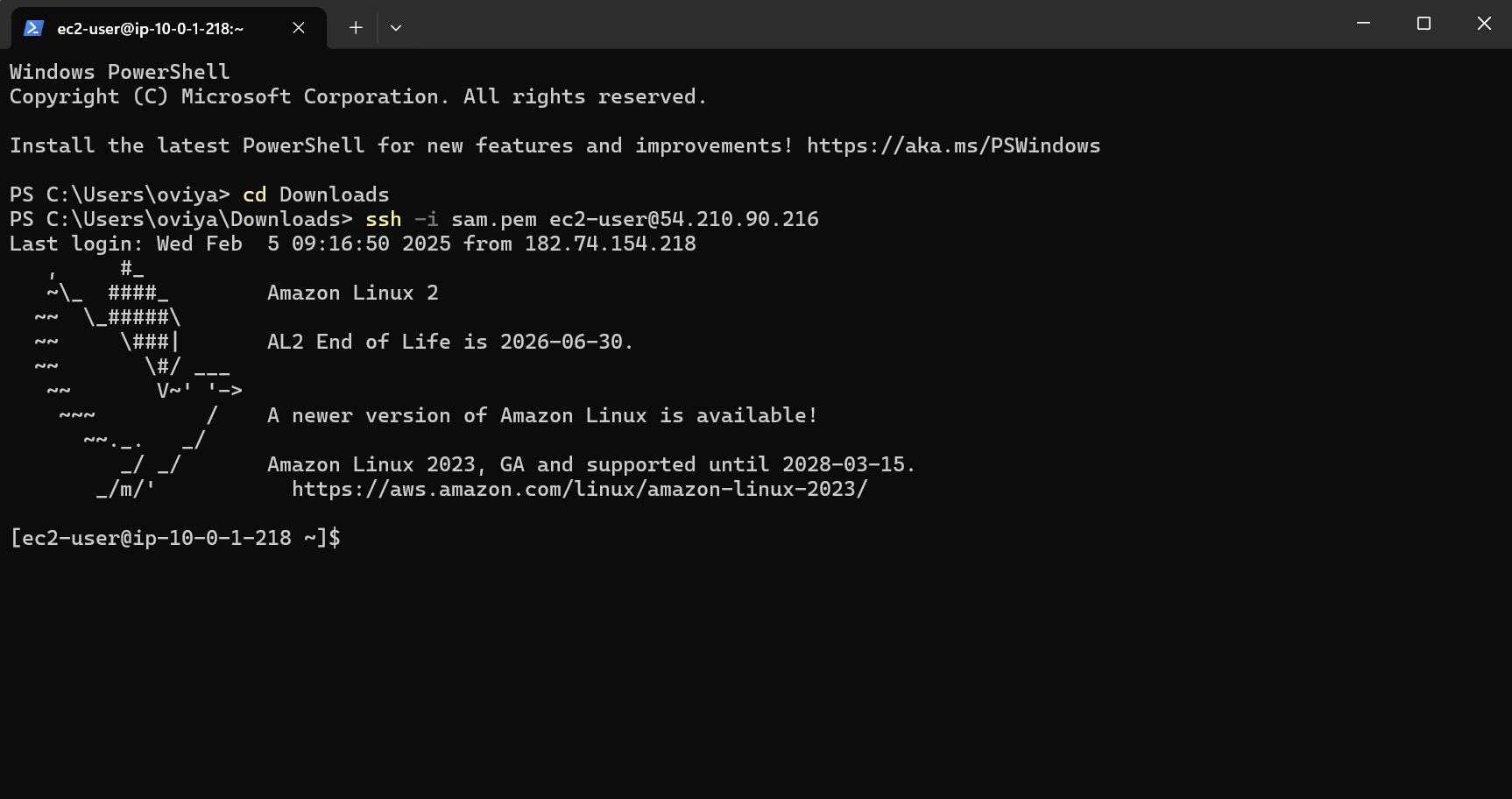
Description automatically generated

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

**5.1 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.)



**5.2 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**

**6.1 Restrict SSH Access**

* **Go to Security Group** (BastionSG) , and then Edit Inbound Rules.
* **Allow SSH only from your IP address (xx.xx.xx.xx/32)** instead of allowing all (0.0.0.0/0)

**6.2 Disable Password Authentication**

* Edit SSH config:

sudo nano /etc/ssh/sshd\_config

* Find and update these lines:

PasswordAuthentication no

PermitRootLogin no

* Restart SSH service:

sudo systemctl restart sshd

A screenshot of a computer

AI-generated content may be incorrect.

**Step 7:**

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

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

**Conclusion:**

A Bastion Host improves security by serving as a controlled access point to private instances. It blocks direct internet access, enforces security rules, and enables access monitoring and logging.

For even stronger security, you can remove SSH access entirely and use AWS Systems Manager (SSM) Session Manager instead.

THANK YOU!