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: CHANDRU S

Department: INFORMATION TECHNOLOGY

Introduction

In cloud environments, securing access to private instances is essential. A **Bastion Host** (also known as a **Jump Box**) acts as a secure gateway to access **EC2 instances in a private subnet**. Instead of exposing private instances directly to the internet, users first connect to the Bastion Host, which then allows access to the private instances.

This setup enhances security by **restricting direct SSH access**, enforcing **strict security controls**, and reducing exposure to potential threats.

Overview

In this guide, we will set up a Bastion Host in a public subnet to provide controlled SSH access to instances inside a private subnet.

What We Will Do

1. Create a VPC with Public and Private subnets.
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 (SSM) as an alternative.

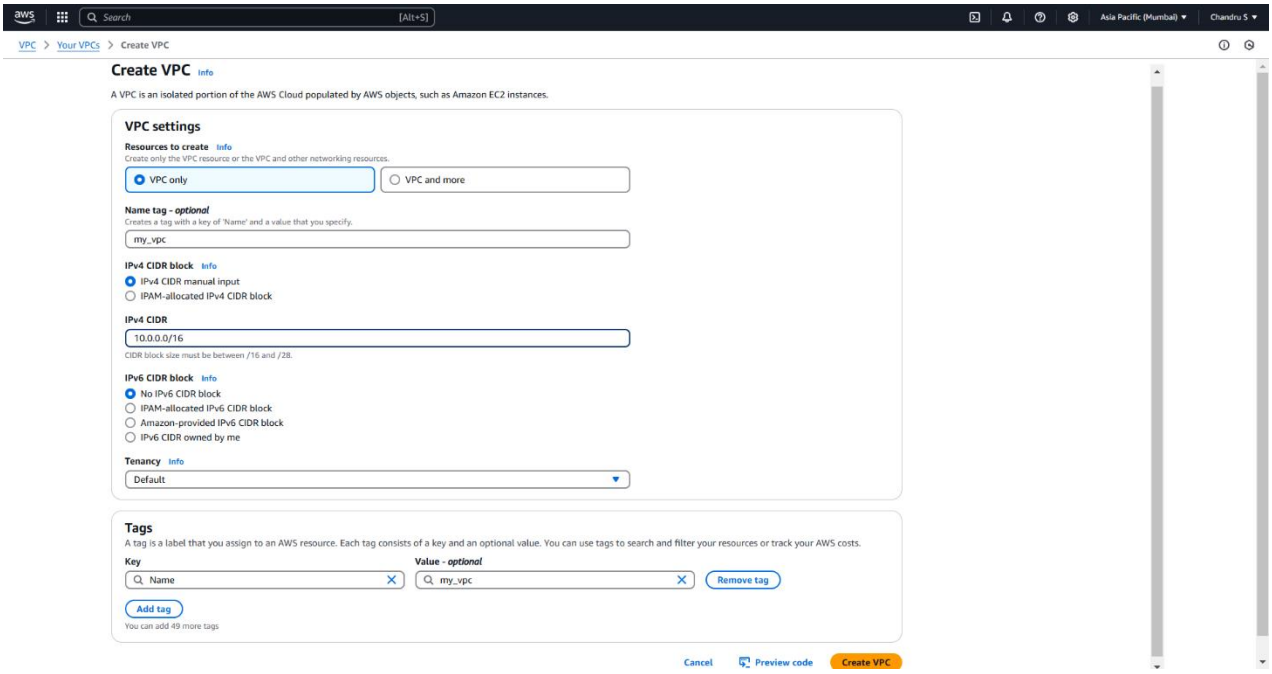
Step-by-Step Overview

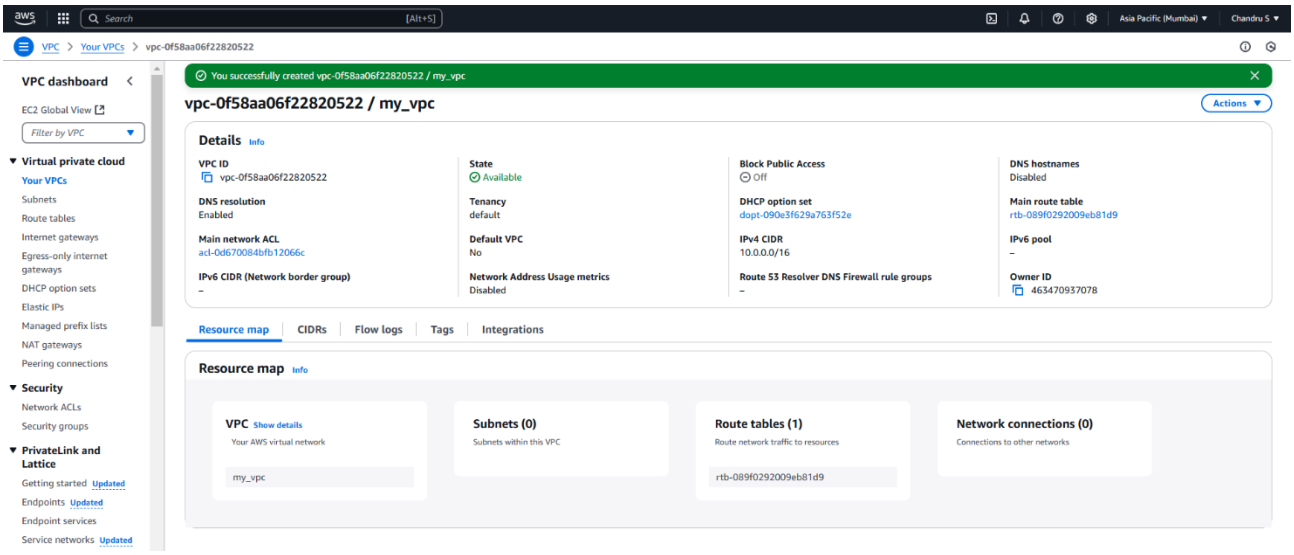
Step 1: Create a VPC with Public and Private Subnets

1.1 Create a VPC

- Open the AWS Management Console → Navigate to **VPC Dashboard**.

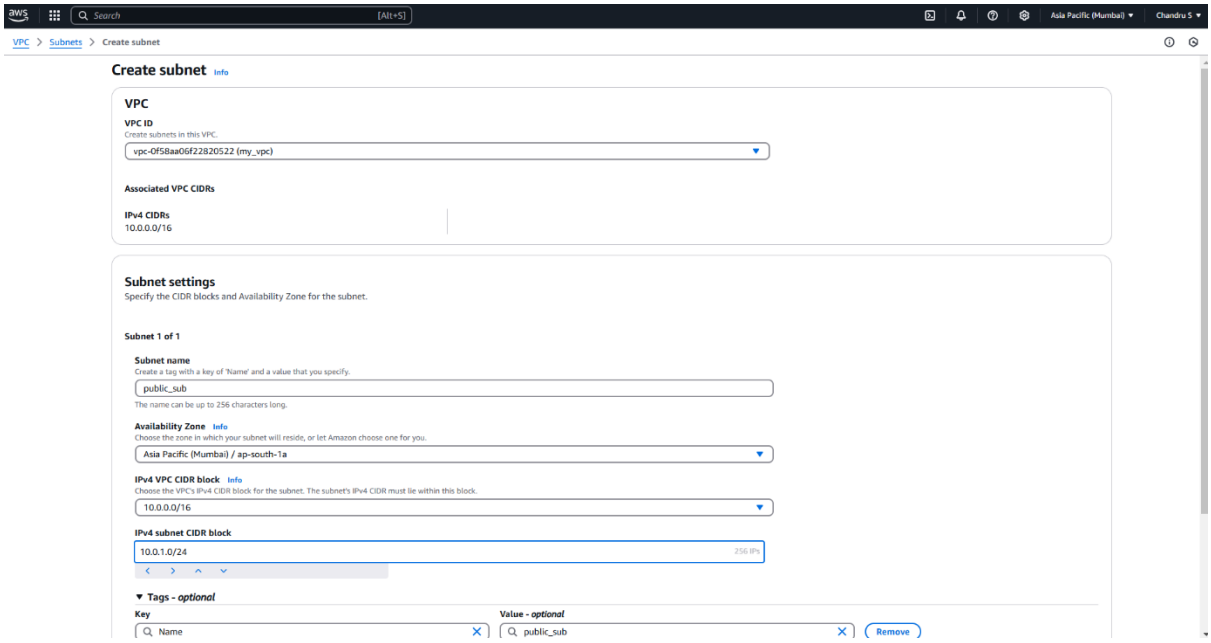
- Click Create VPC and name it **my_vpc**.
- Set **IPv4 CIDR Block**: 10.0.0.0/16.
- Click **Create VPC**.

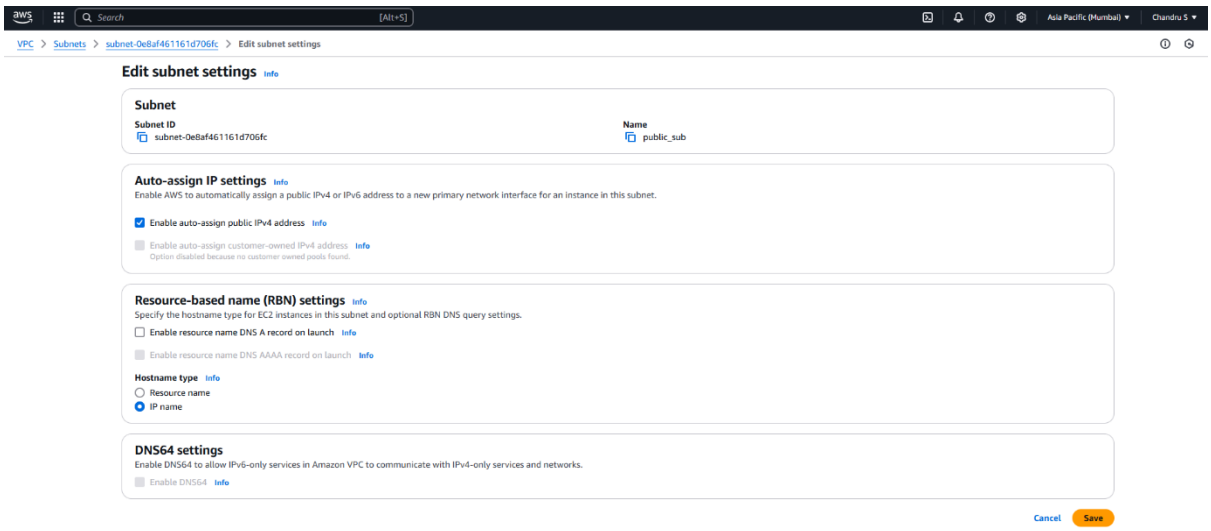




1.2 Create a Public Subnet

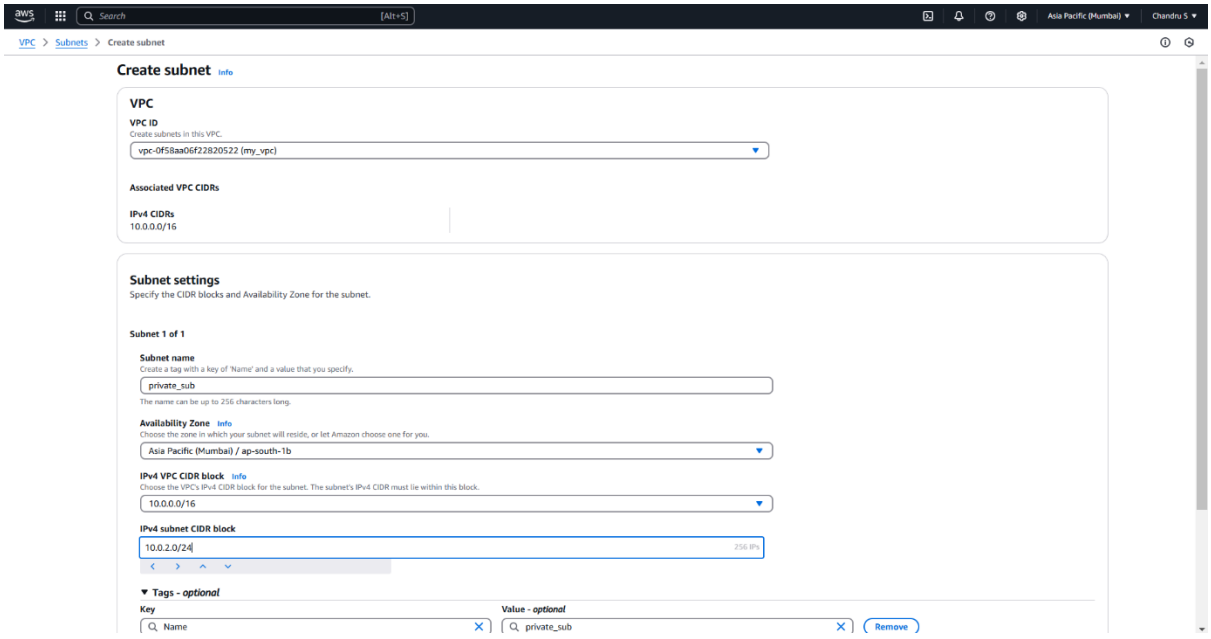
- Go to Subnets → Create Subnet.
- Select **my_vpc** and set **CIDR block:** 10.0.1.0/24.
- Enable **Auto-Assign Public IP**.

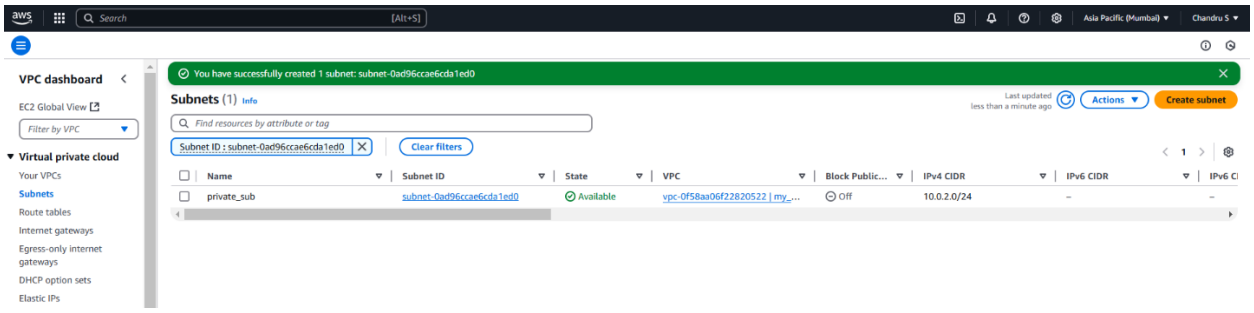




1.3 Create a Private Subnet

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

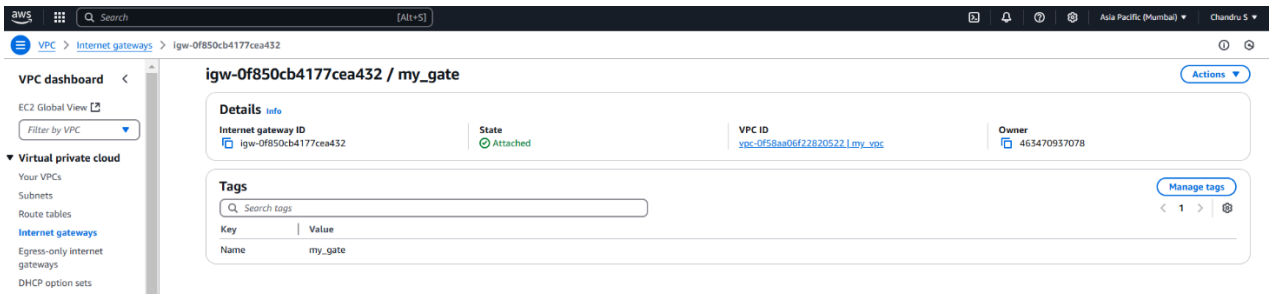




Step 2: Configure Public Subnet for Internet Access

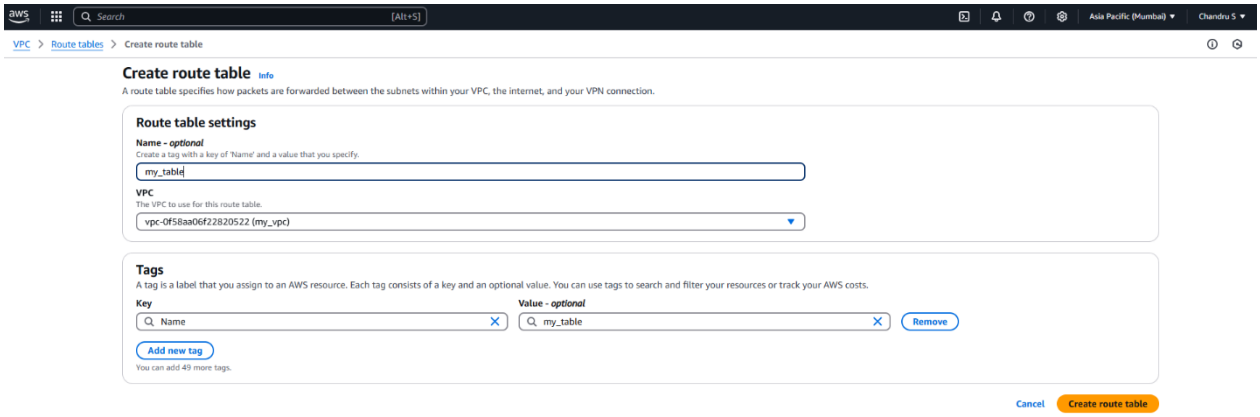
2.1 Create an Internet Gateway (IGW)

- Go to Internet Gateways → Click Create Internet Gateway.
- Name it **my_gate**, attach it to **my_vpc.**

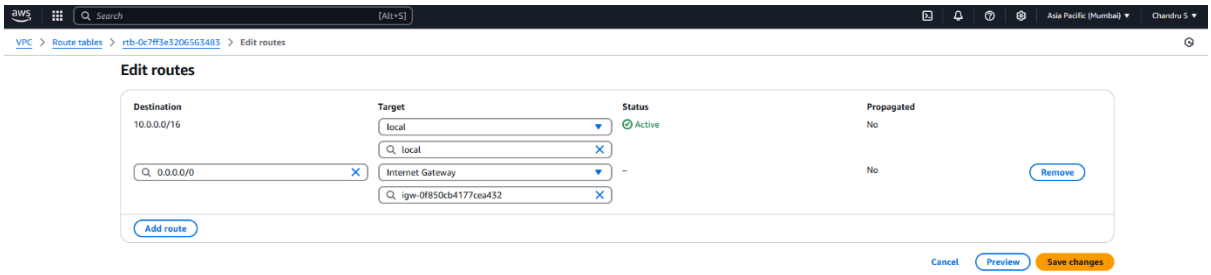


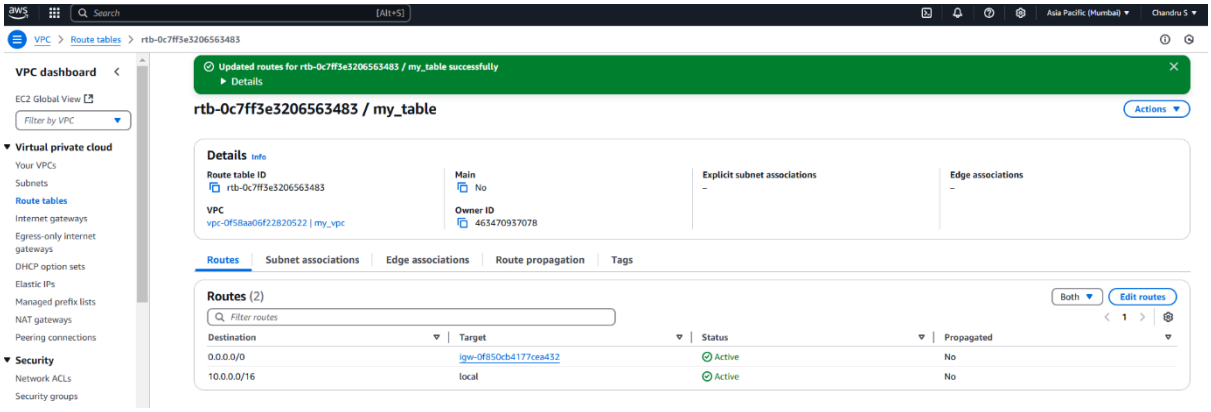
2.2 Update Public Route Table

- Go to Route Tables → Create Route Table → Name it **my_table.**



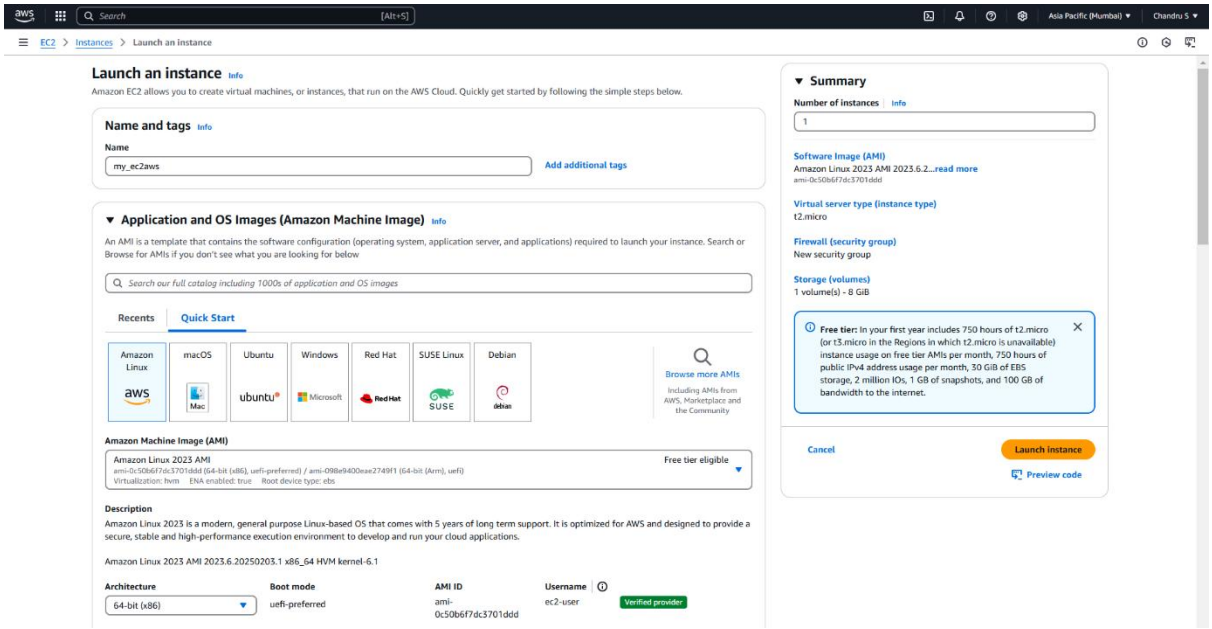
- Associate it with **PublicSubnet**.
- Add a route:
  - Destination: **0.0.0.0/0**
  - Target: Internet Gateway (**my_gate**)



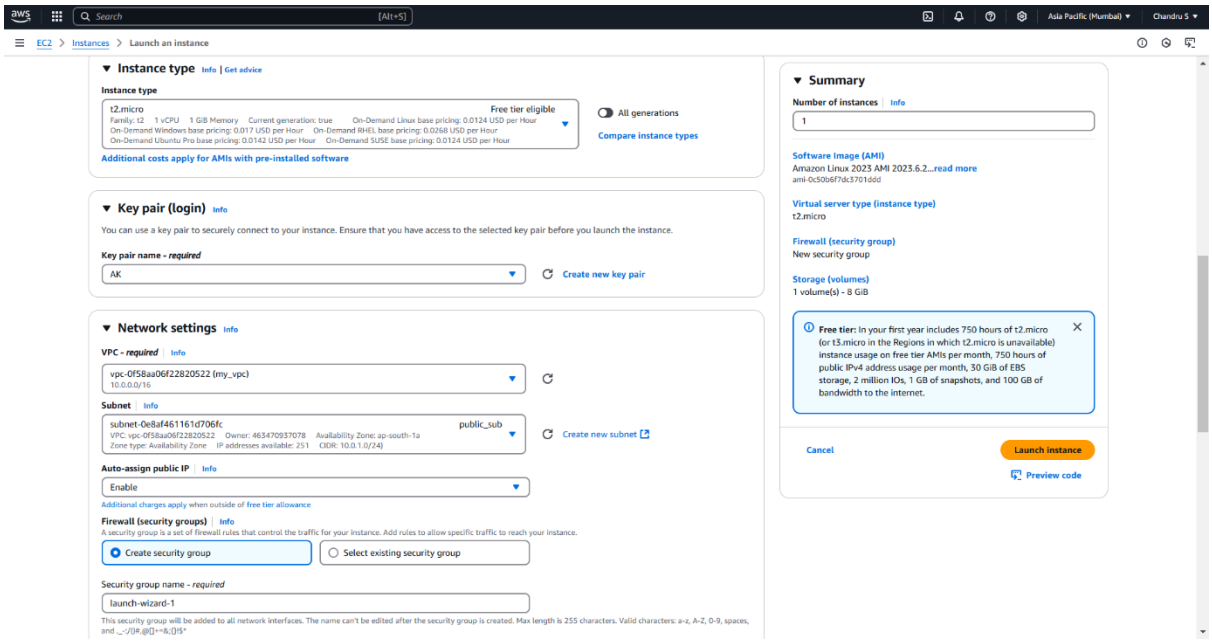


Step 3: Launch a Bastion Host in the Public Subnet

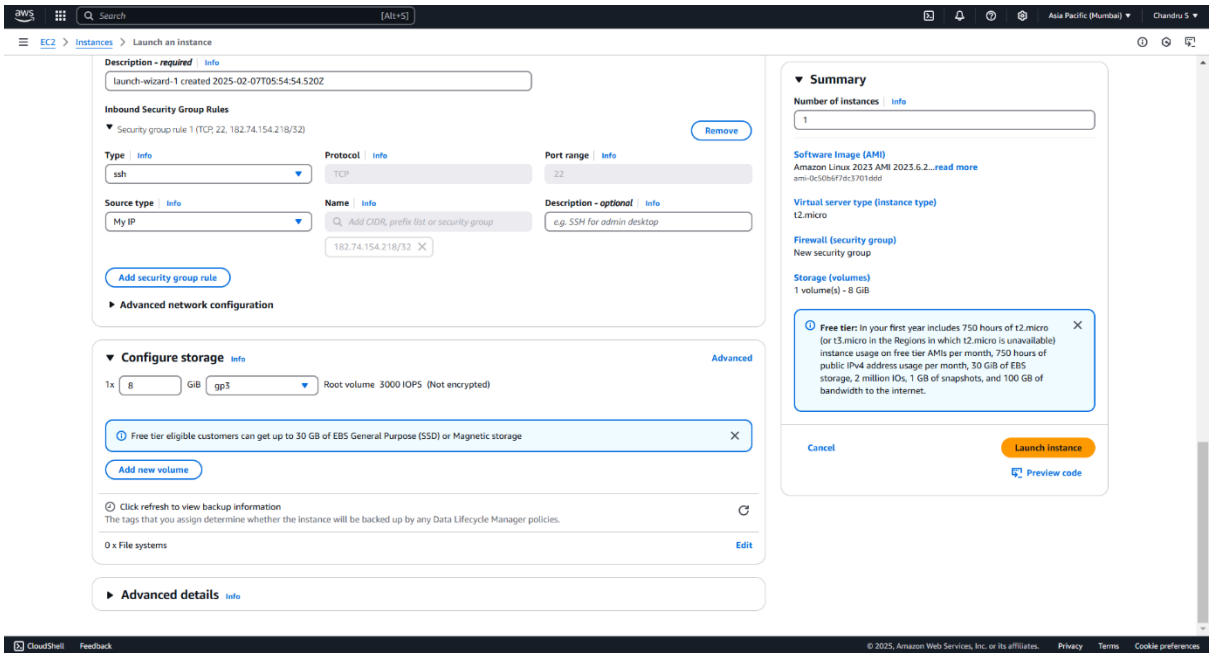
1. Go to **EC2 Dashboard** → Launch Instance.
2. Select **Amazon Linux 2** (or **Ubuntu**).



3. Choose **t2.micro** (Free Tier Eligible).
4. Place it in **PublicSubnet** with Auto-Assign Public IP **enabled**.

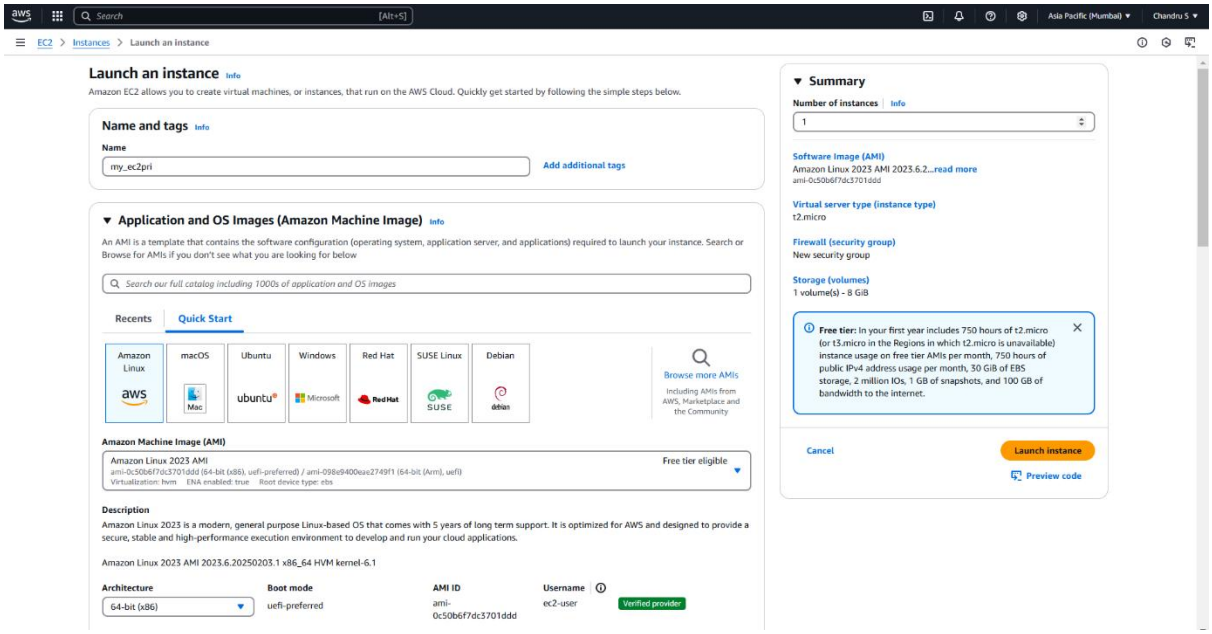


5. Create a Security Group (**BastionSG**):
    - Allow **SSH** (Port 22) from **Your IP** (xx.xx.xx.xx/32).
6. Create or use an existing key pair (e.g., **bastion-key.pem**).
7. Click **Launch**.

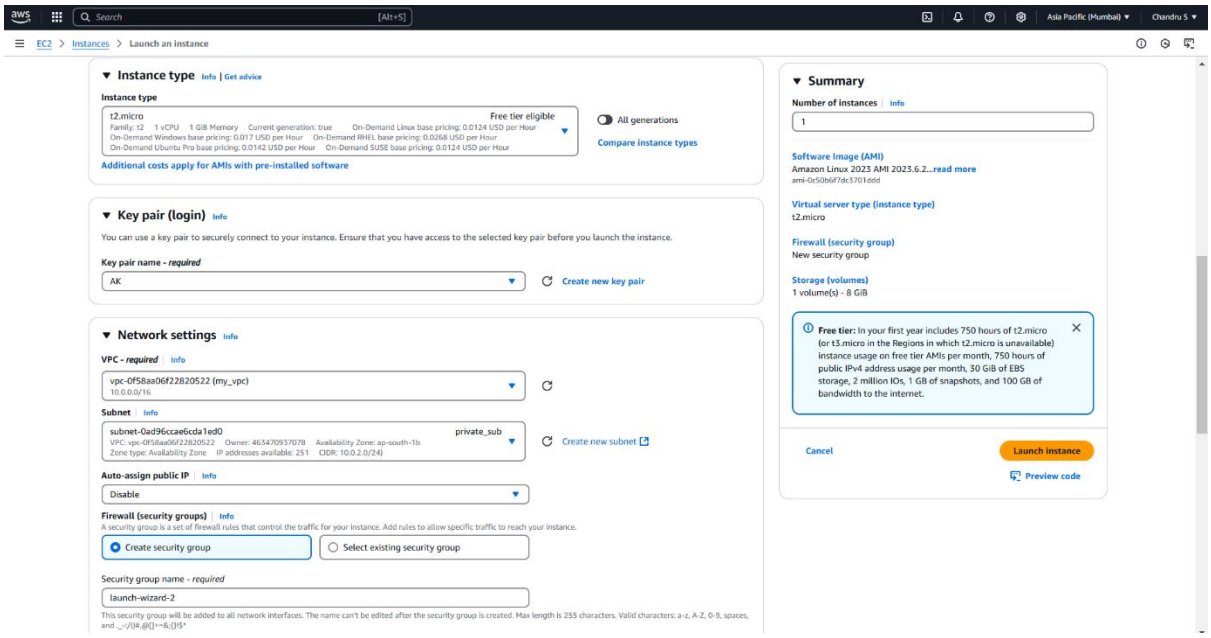


Step 4: Launch a Private EC2 Instance

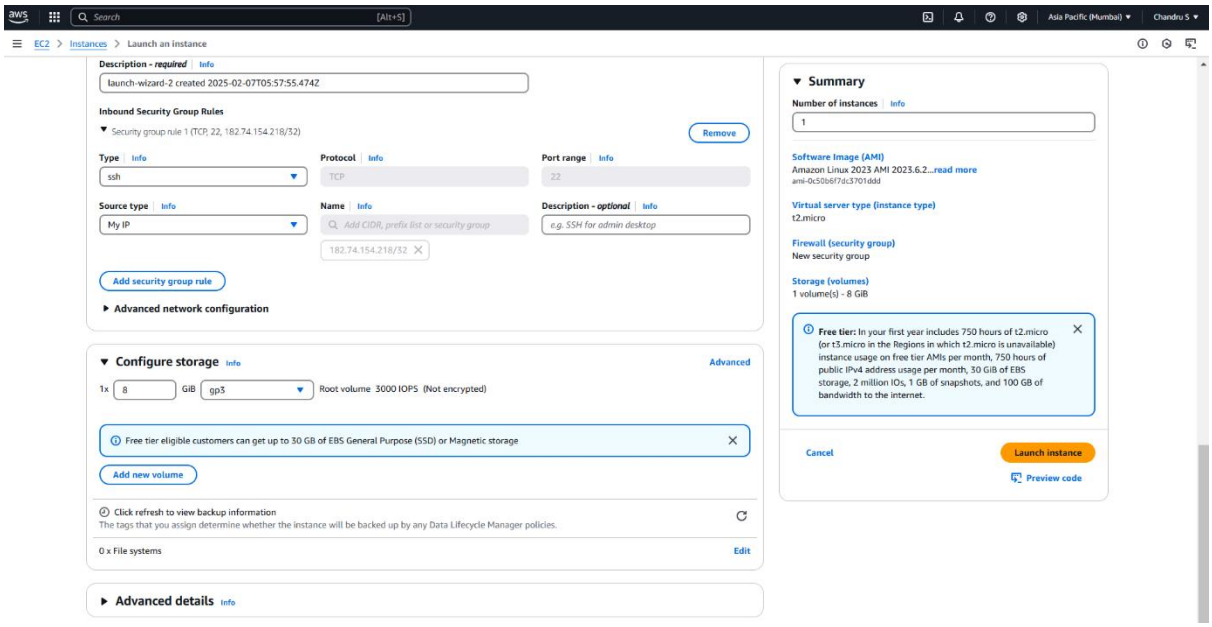
1. Go to EC2 Dashboard → Launch Instance.
2. Choose Amazon Linux 2 (or Ubuntu).



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).



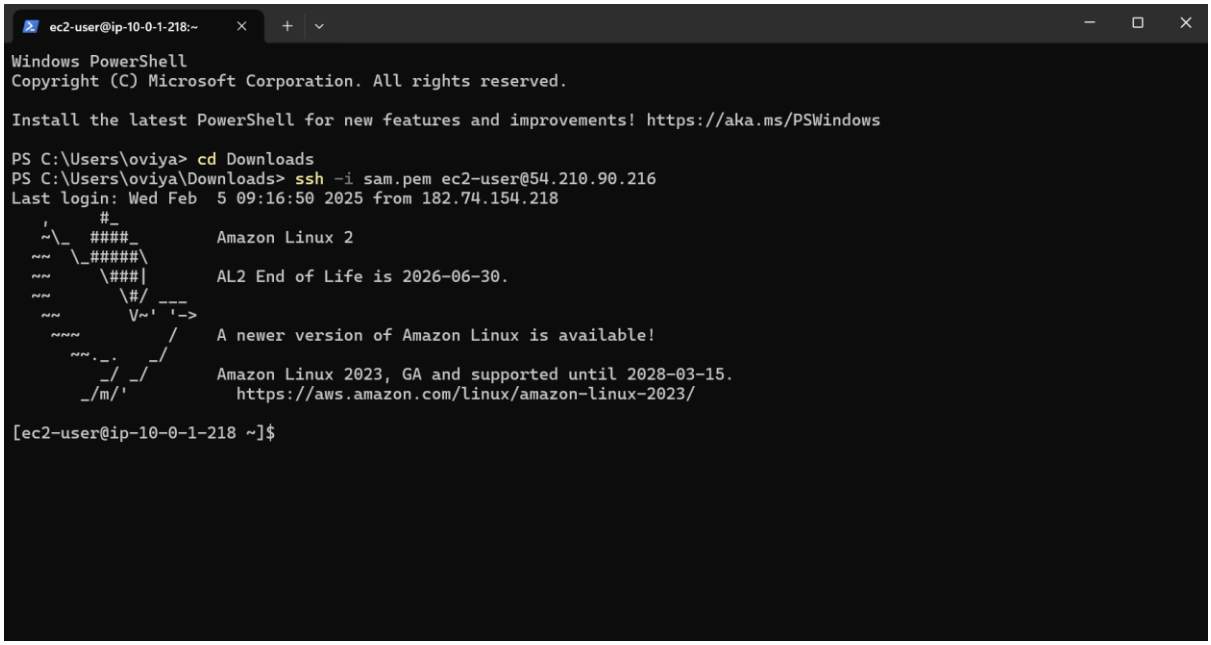
7. Click Launch.

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

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>:~/

2. Connect to the Bastion Host:

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

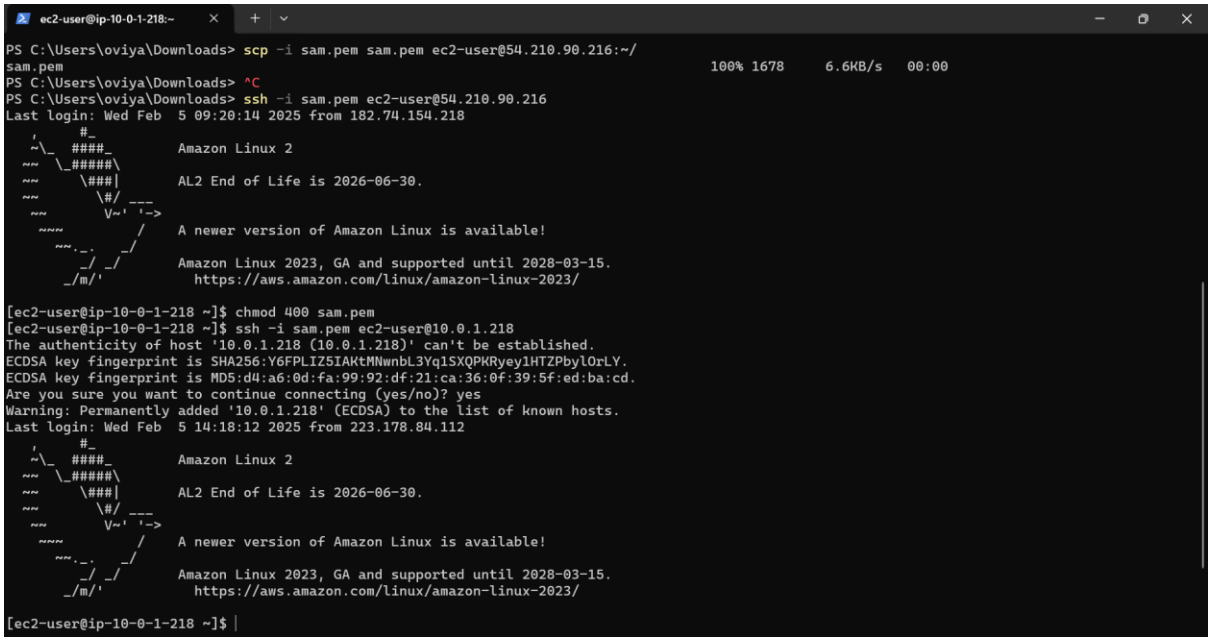
3. Change permissions for the key file:

chmod 400 bastion-key.pem

4. SSH into the Private Instance from the Bastion Host:

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

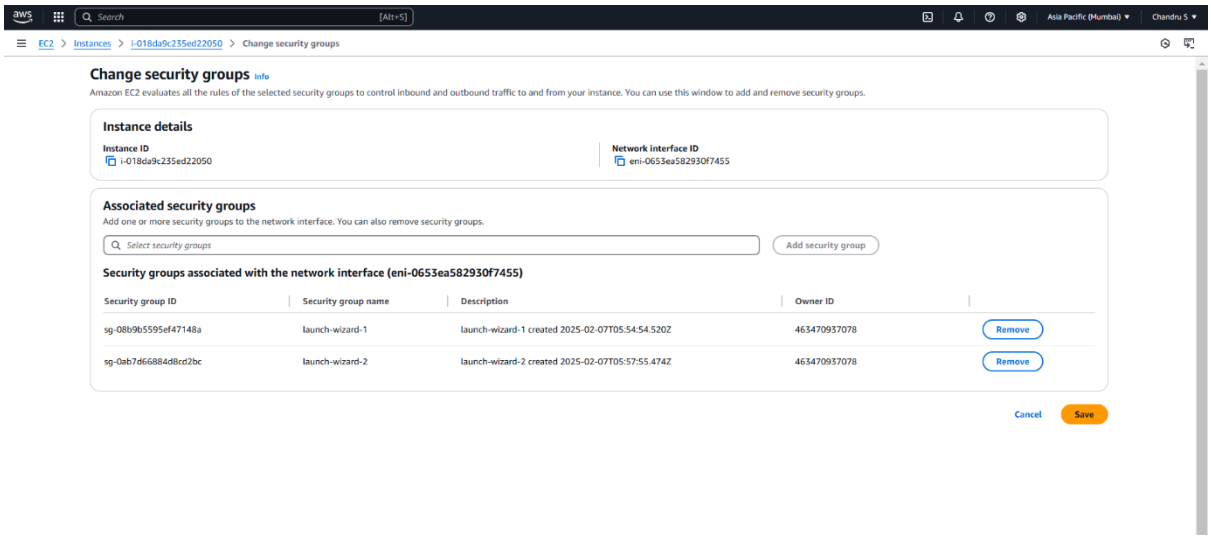
(Replace <private-instance-ip> with the private IP of your instance.)



Step 6: Secure Your Bastion Host

6.1 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)

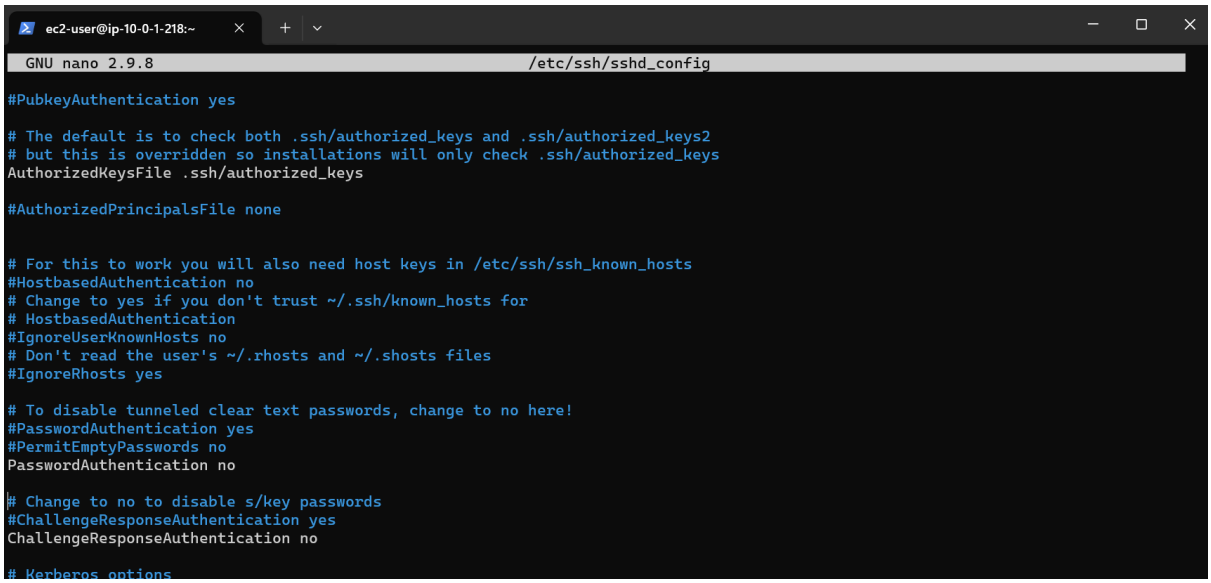


6.2 Disable Password Authentication

1. Edit SSH config:
   sudo nano /etc/ssh/sshd_config
2. Find and update these lines:

PasswordAuthentication no
PermitRootLogin no

3. Restart SSH service:
   sudo systemctl restart sshd



Step 7: Alternative - Use AWS Systems Manager (SSM) Instead of SSH

- Attach **AmazonSSMManagedInstanceCore** IAM policy to the EC2 instance role.
- Ensure the **SSM Agent** is enabled (pre-installed on Amazon Linux & Ubuntu).
- Use **AWS Systems Manager > Session Manager** to connect without SSH.

Conclusion

Using a Bastion Host improves security by acting as a controlled access point for private instances. It prevents direct internet exposure, enforces security group rules, and allows monitoring/logging of access attempts.

For even stronger security, consider eliminating SSH access entirely and using AWS Systems Manager (SSM) Session Manager instead.