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: Akshaya S Department:CSE

Introduction

A bastion host is a secure server that acts as a bridge between public and private networks. In cloud environments, a bastion host is used to securely access instances in private subnets, as direct internet access is restricted for security reasons. This Proof of Concept (POC) demonstrates how to set up a bastion host in AWS to access private instances while ensuring robust network security.

Overview

In this POC, we design and implement a secure architecture using AWS services. The project involves:

1. Creating a custom Virtual Private Cloud (VPC) with public and private subnets.
2. Launching an EC2 instance (bastion host) in the public subnet and a private instance in the private subnet.
3. Configuring security groups to control network traffic and enable secure access.
4. Using the bastion host as an intermediary to SSH into the private instance without exposing it directly to the internet.

The POC verifies secure access by testing connectivity, verifying the private instance's setup, and ensuring proper configurations.

Objectives

The primary objectives of this POC are:

1. Learn Network Segmentation:
Understand how to segregate public and private resources within a VPC.

2. Secure Private Resources:
Enable access to private instances without exposing them to the internet.

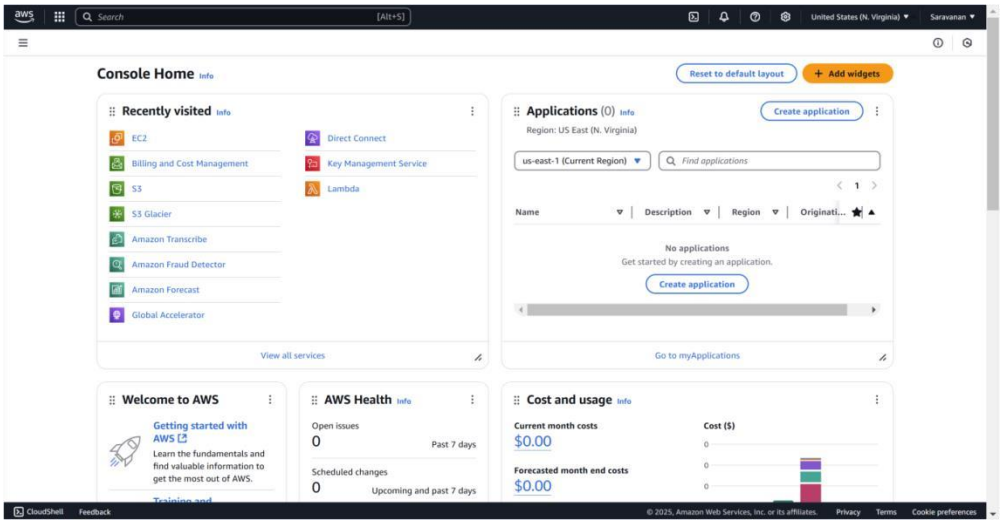
3. Practice Secure Access Techniques:
Use a bastion host to securely SSH into a private instance.

4. Apply Security Best Practices:
Use key-based authentication, restrict inbound traffic, and follow the principle of least privilege in security group configurations.

Step-by-Step Overview

Step 1:

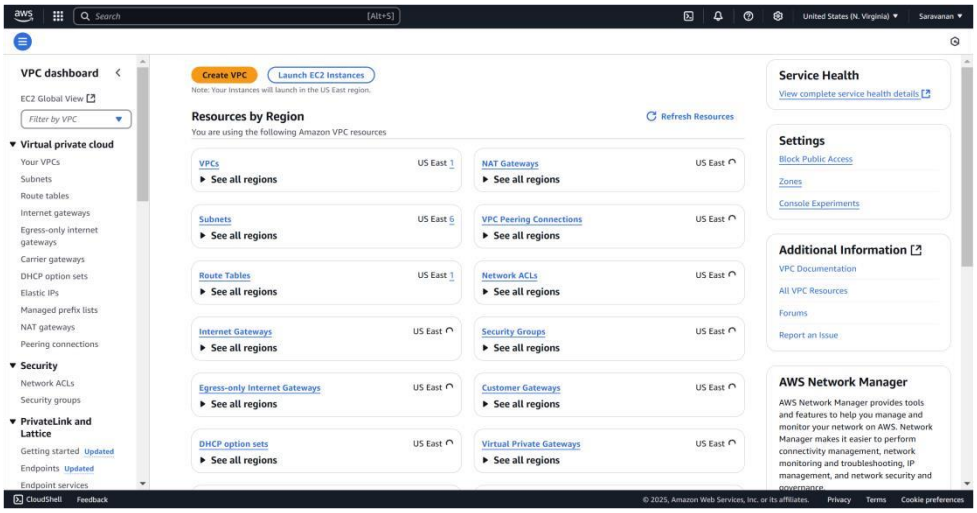
1. Go to AWS Management Console.
2. Enter your username and password to log in.



Step 2:

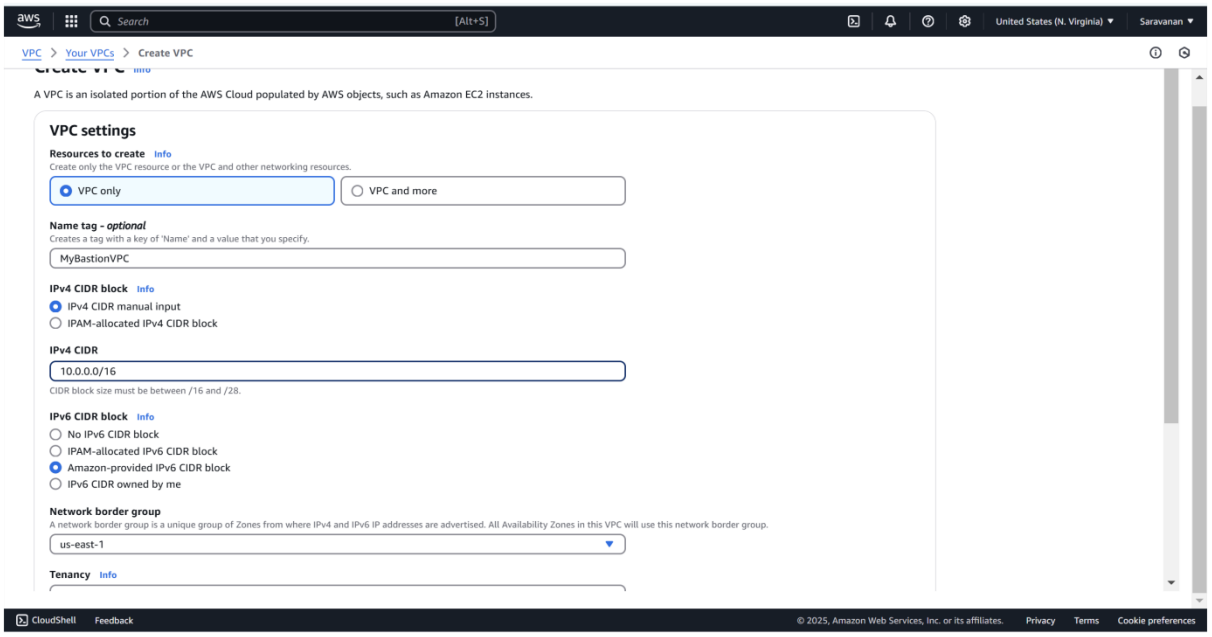
Search for **VPC** in the AWS search bar and click on it.

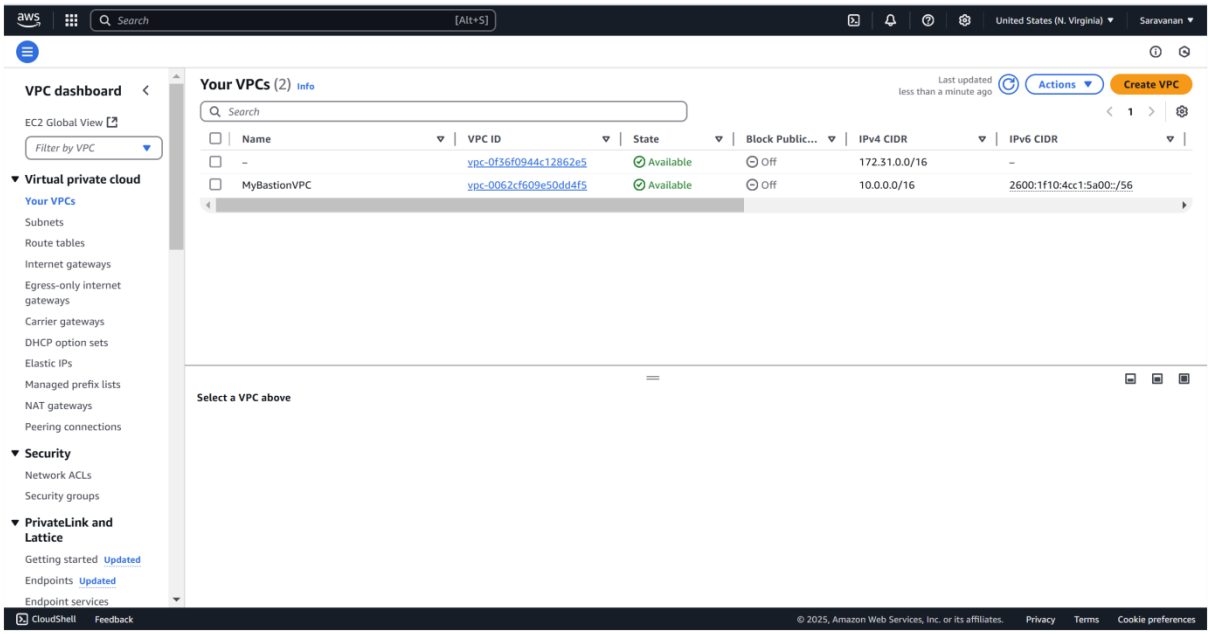
Click on **Create VPC**.



Step 3:

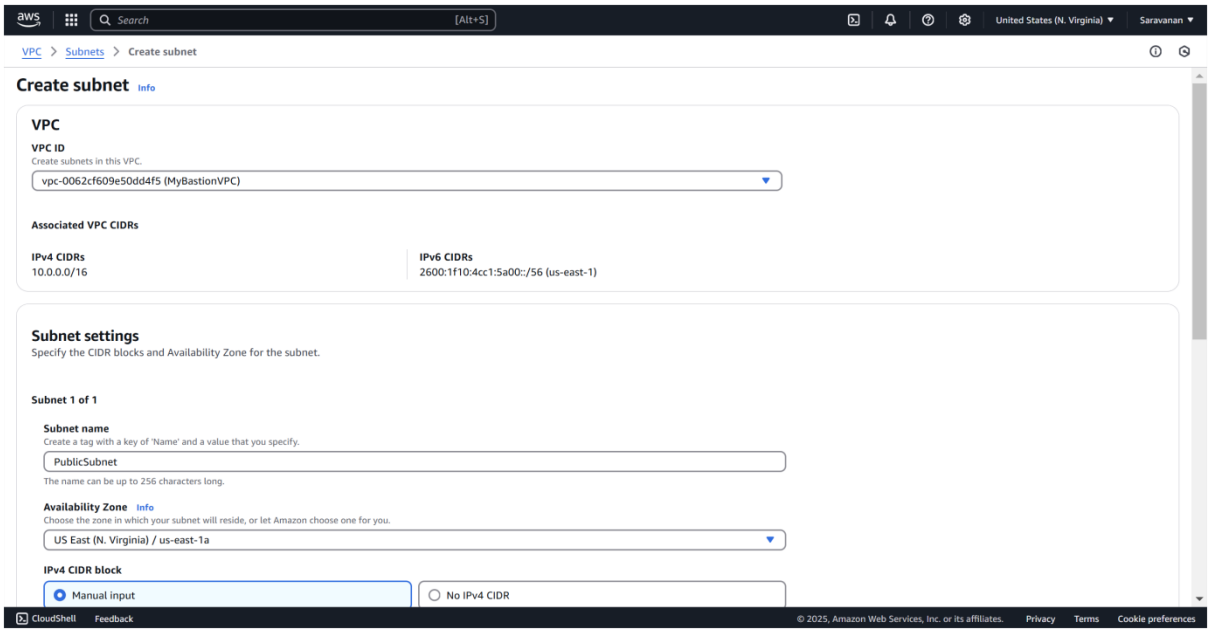
Create a new VPC by selecting **VPC only** and filling in the following details: set the **Name Tag** as *MyBastionVPC* and the **IPv4 CIDR Block** as *10.0.0.0/16*. Leave all other settings as default, then click **Create VPC**. Once created, the new VPC will appear in the VPC list.





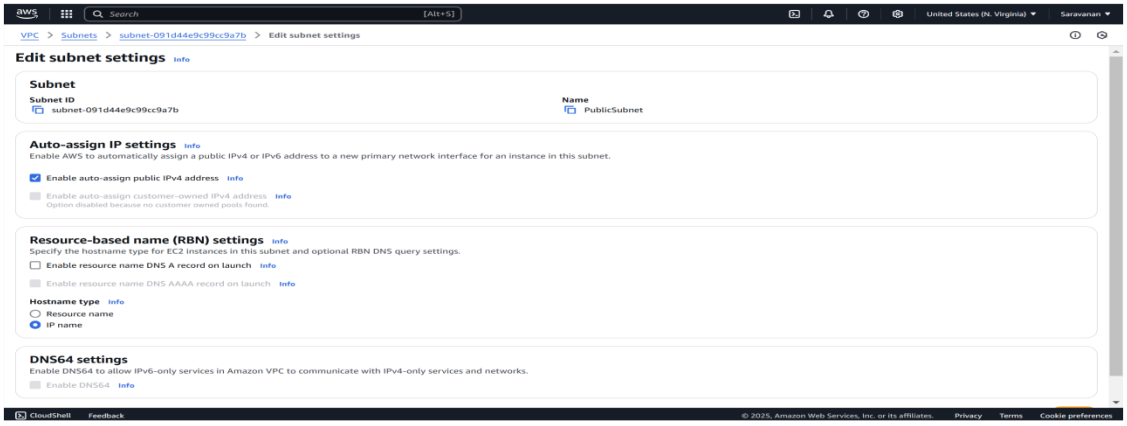
Step 4:

In the **VPC Dashboard**, go to **Subnets** and click **Create Subnet**. Select the **VPC ID** of the VPC you created earlier (*MyBastionVPC*). Enter the **Subnet Name** as *PublicSubnet*, choose an **Availability Zone** (e.g., *us-east-1a*), and set the **IPv4 CIDR Block** as *10.0.1.0/24*. Click **Create Subnet**.



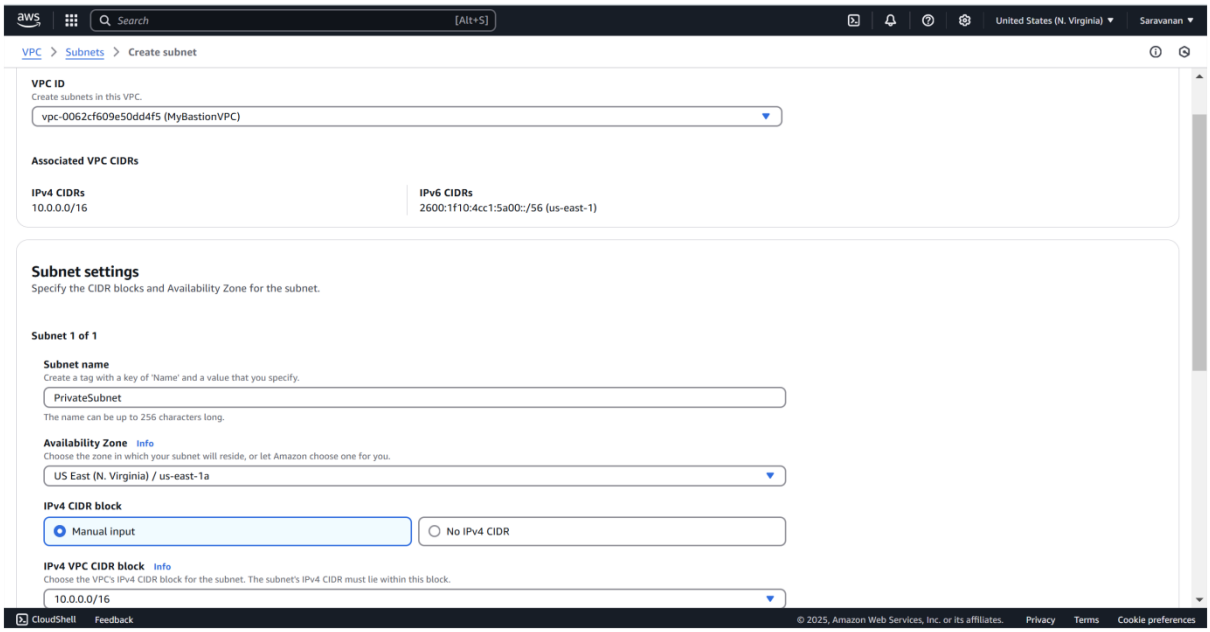
Step 5:

Select your **PublicSubnet** from the list, click **Actions → Modify auto-assign IP settings**, check **Enable auto-assign public IPv4 address**, and click **Save**.



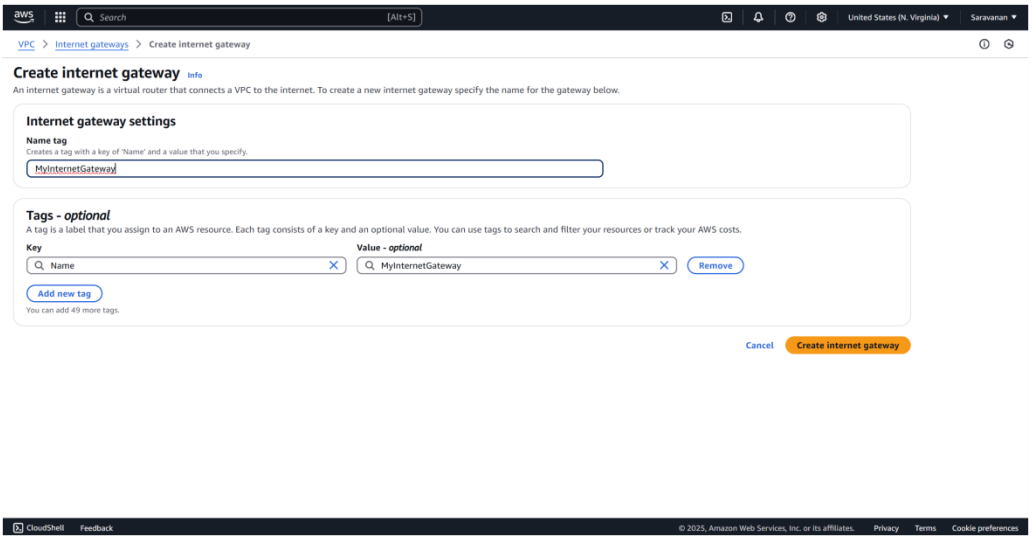
Step 6:

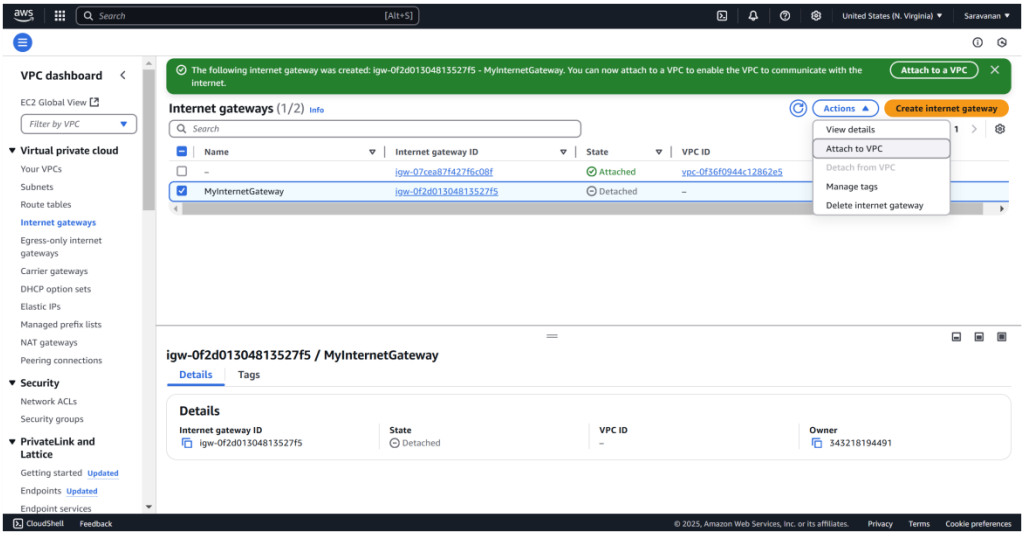
Click **Create Subnet** again and fill in the details: select the same **VPC ID** (*MyBastionVPC*), set **Subnet Name** to *PrivateSubnet*, use the same **Availability Zone** as the public subnet (e.g., *us-east-1a*), and set the **IPv4 CIDR Block** to *10.0.2.0/24*. Leave **auto-assign public IP** disabled and click **Create Subnet**.

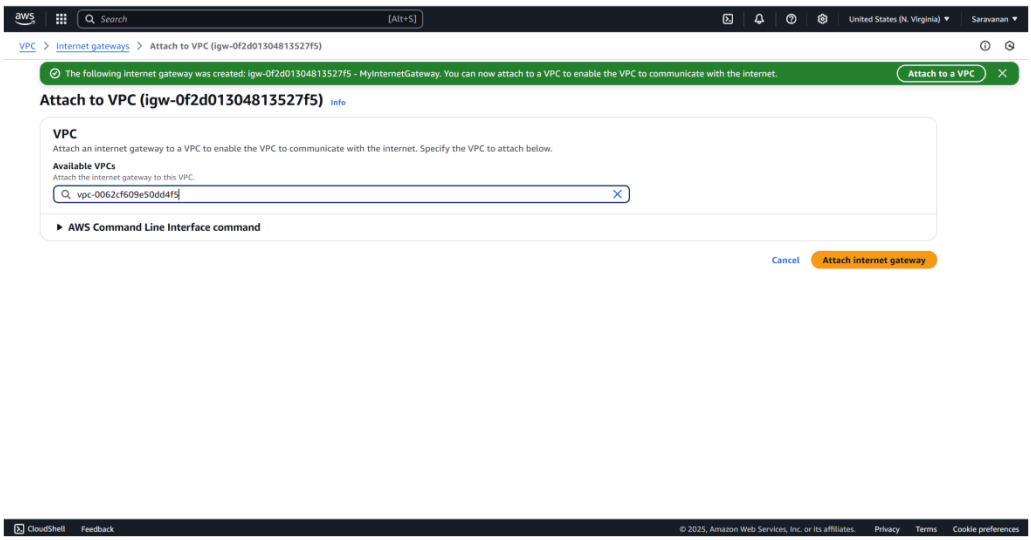


Step 7:

In the **VPC Dashboard**, go to **Internet Gateways** and click **Create Internet Gateway**. Name it *MyInternetGateway* and click **Create Internet Gateway**. Select your new gateway, click **Actions → Attach to VPC**, choose your VPC (*MyBastionVPC*), and click **Attach Internet Gateway**.

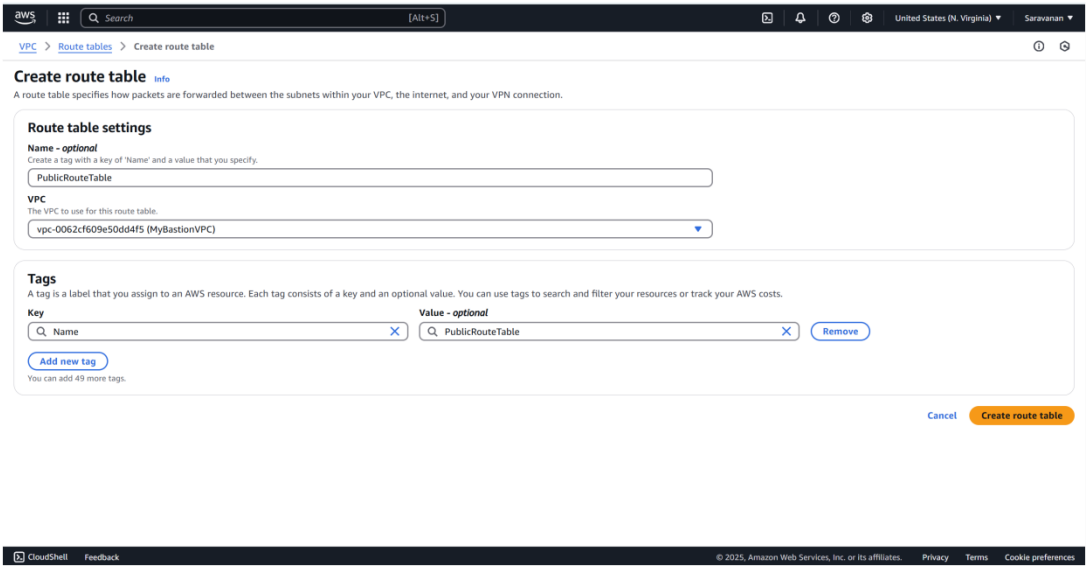

Search [Alt+S]
United States (N. Virginia)
Saravanan
VPC > Internet gateways > Create internet gateway
Create internet gateway Info
An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.
Internet gateway settings
Name tag
Creates a tag with a key of 'Name' and a value that you specify.
MyInternetGateway
Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
Key
Name
Value - optional
MyInternetGateway
Remove
Add new tag
You can add 49 more tags.
Cancel
Create internet gateway
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

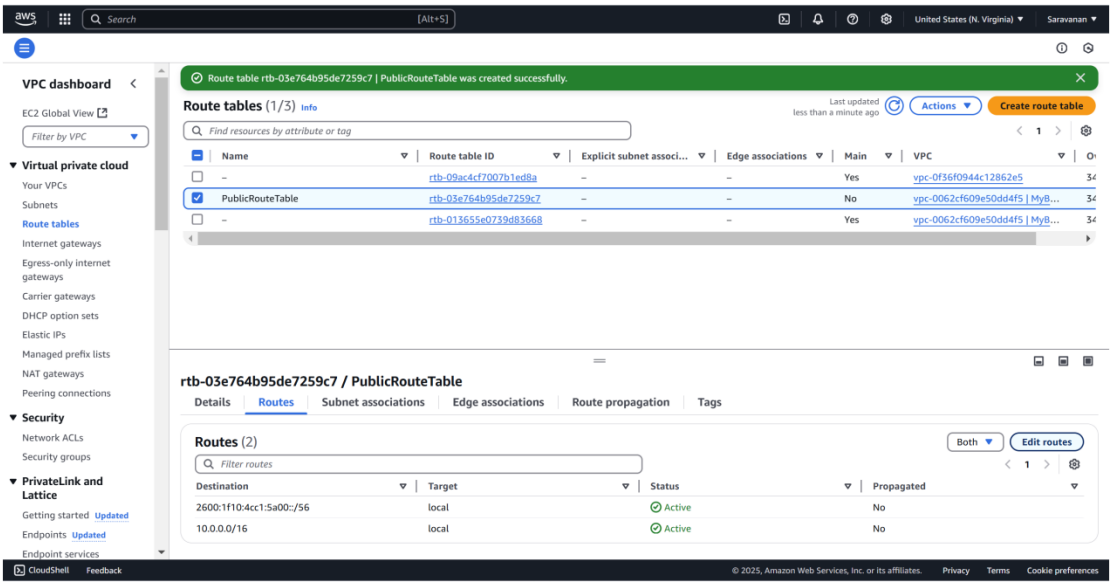

VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Security
Network ACLs
Security groups
PrivateLink and Lattice
Getting started Updated
Endpoints Updated
Endpoint services
The following internet gateway was created: igw-0f2d01304813527f5 - MyInternetGateway. You can now attach to a VPC to enable the VPC to communicate with the internet.
Attach to a VPC
Internet gateways (1/2) Info
Search
Actions
Create internet gateway
Name
Internet gateway ID
State
VPC ID
igw-07cea87f427f6c08f
Attached
vpc-0f36f0944c12862e5
MyInternetGateway
igw-0f2d01304813527f5
Detached
View details
Attach to VPC
Detach from VPC
Manage tags
Delete internet gateway
igw-0f2d01304813527f5 / MyInternetGateway
Details
Tags
Details
Internet gateway ID
igw-0f2d01304813527f5
State
Detached
VPC ID
Owner
343218194491
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
United States (N. Virginia)
Saravanan
VPC > Internet gateways > Attach to VPC (igw-0f2d01304813527f5)
The following internet gateway was created: igw-0f2d01304813527f5 - MyInternetGateway. You can now attach to a VPC to enable the VPC to communicate with the internet.
Attach to a VPC
Attach to VPC (igw-0f2d01304813527f5) Info
VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.
Available VPCs
Attach the internet gateway to this VPC.
vpc-0062cf609e50dd4f5
AWS Command Line Interface command
Cancel
Attach internet gateway
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Step 8:

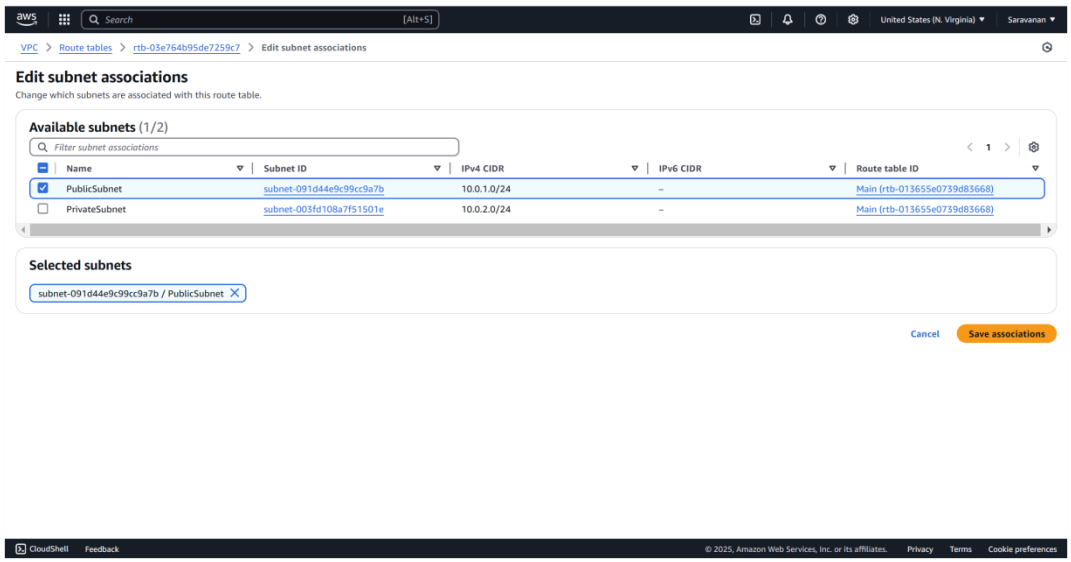
In the **VPC Dashboard**, go to **Route Tables** and click **Create Route Table**. Name it *PublicRouteTable*, select your VPC (*MyBastionVPC*), and click **Create Route Table**. Then, select *PublicRouteTable*, go to the **Routes** tab, click **Edit routes**, and add a route with **Destination** as *0.0.0.0/0* and **Target** as *MyInternetGateway*. Click **Save changes**.

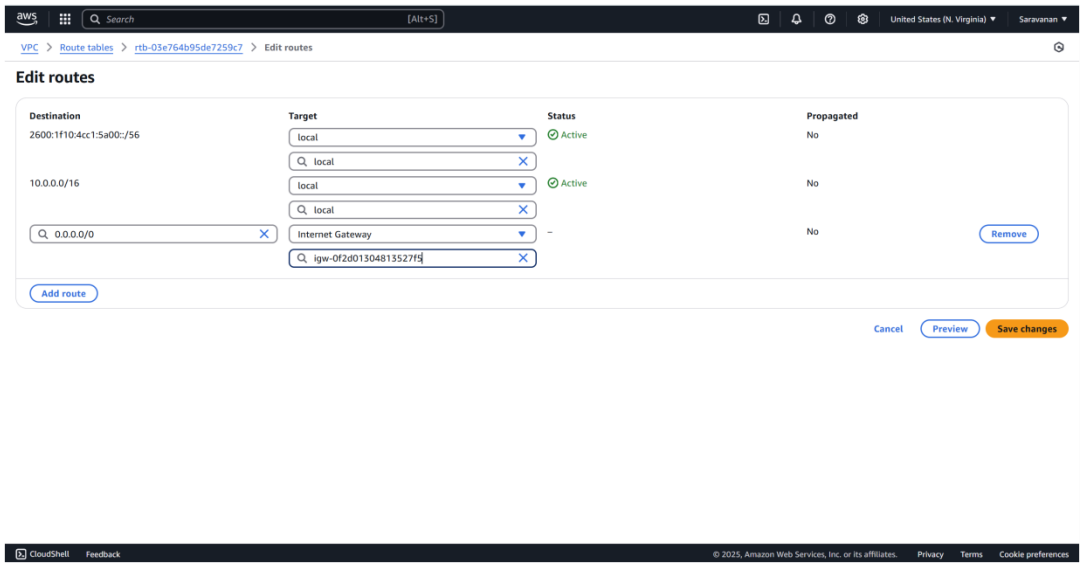




Step 9:

Next, go to the **Subnet associations** tab of *PublicRouteTable*, click **Edit subnet associations**, check the box for *PublicSubnet*, and click **Save associations**.

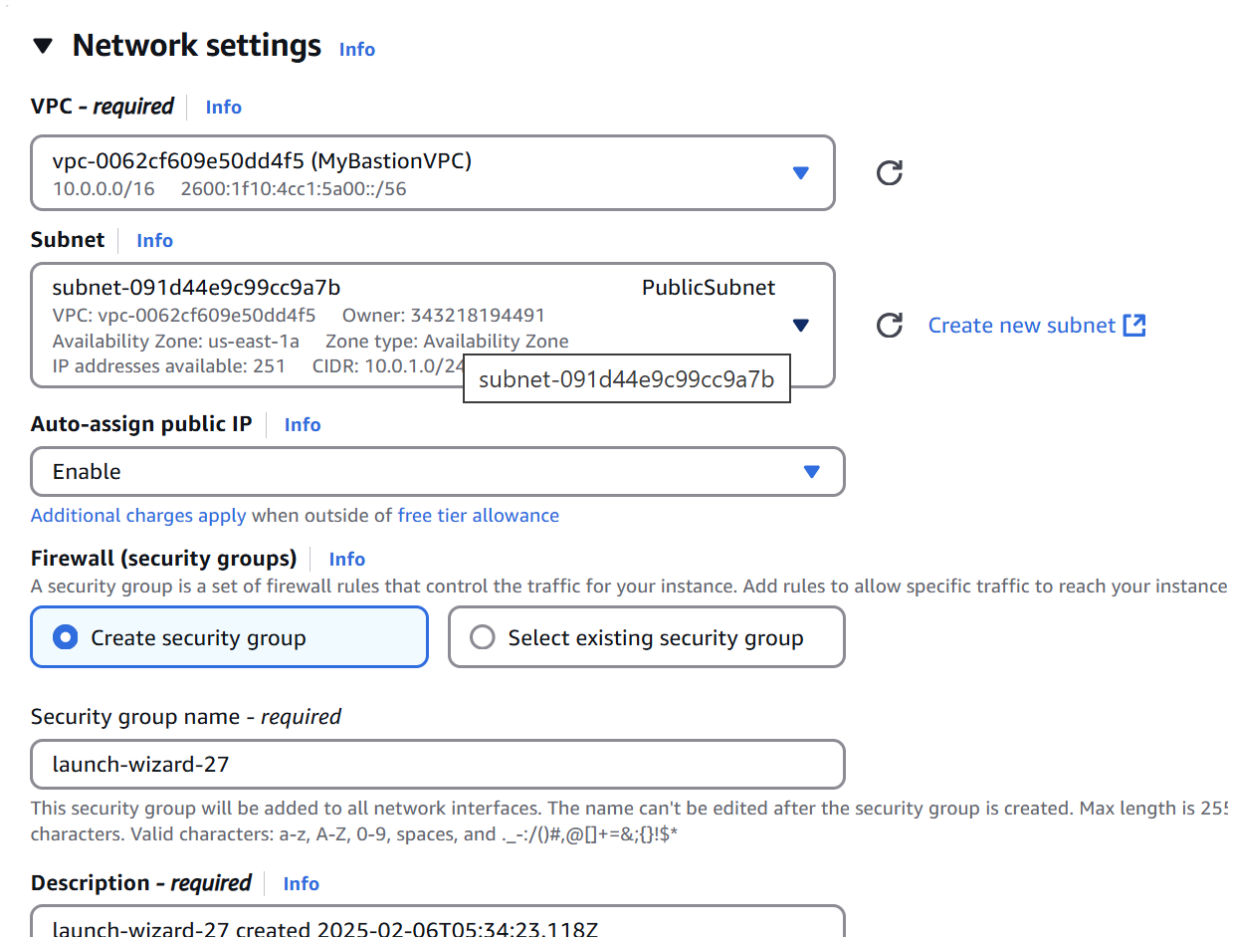




In the **EC2 Dashboard**, click **Launch Instance** and configure: set **Name** as *BastionHost*, select *Amazon Linux 2 AMI (HVM)* - Free Tier eligible, and choose **t2.micro** as the **Instance Type**. For **Key Pair**,

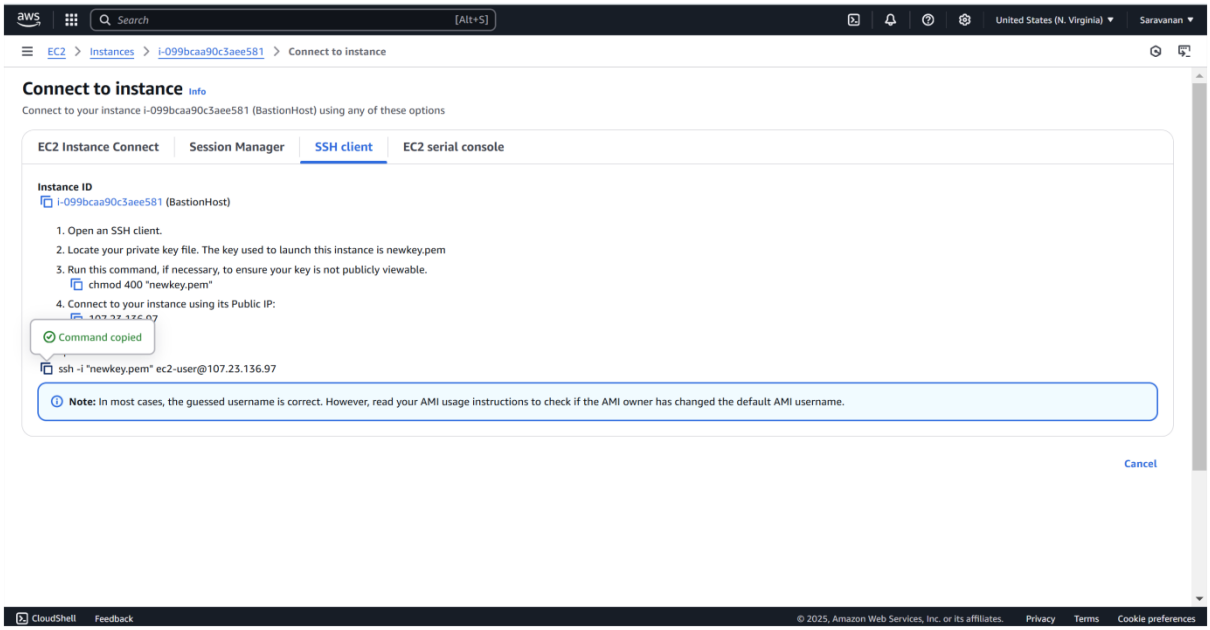
Step 10:

create or select one, downloading the .pem file if creating. Under **Network Settings**, select *MyBastionVPC* for the **VPC**, *PublicSubnet* for the **Subnet**, and ensure **Auto-assign Public IP** is enabled. Create a **Security Group** to allow SSH (port 22) access, setting **Source** to *MyIP*. Use the default storage of 8 GiB, click **Launch Instance**, and wait for it to initialize.



Step 10:

Connect with your PowerShell terminal by copying the ssh command in the SSH client of the *BastionHost(Ec2)*.



Step 11:

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

```
PS C:\Users\Hi> cd Downloads
PS C:\Users\Hi\Downloads> ssh -i "newkey.pem" ec2-user@44.212.36.24
The authenticity of host '44.212.36.24 (44.212.36.24)' can't be established.
ED25519 key fingerprint is SHA256:G5t53dqZ4PoDFHzgf/SJYBIc509HxQC7ROVSqDKom/Y.
This host key is known by the following other names/addresses:
    C:\Users\Hi/.ssh/known_hosts:28: 107.23.136.97
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

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

```
[ec2-user@ip-10-0-1-208 ~]$ mkdir -p ~/.ssh
```

Step 13:

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

Step 12:

scp -i /path/to/your-key.pem /path/to/your-key.pem ec2user@<BastionHost-Public-IP>:~/.ssh/

```
PS C:\Users\Hi> scp -i "C:\Users\Hi\Downloads\newkey.pem" "C:\Users\Hi\Downloads\newkey.pem" ec2-user@44.212.36.24:~/.ssh/
newkey.pem                                                                                  100% 1678     4.0KB/s   00:00
```

Step 14:

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

```
[ec2-user@ip-10-0-1-208 ~]$ chmod 400 ~/.ssh/newkey.pem
```

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-PrivateIP>**

```
[ec2-user@ip-10-0-1-208 ~]$ ssh -i ~/.ssh/newkey.pem ec2-user@10.0.2.68
The authenticity of host '10.0.2.68 (10.0.2.68)' can't be established.
ED25519 key fingerprint is SHA256:MGRZMakTZuL8b0oaK3O7T50//sj23zJJQJn+Zl9lzc4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.68' (ED25519) to the list of known hosts.
```

Step 16:

To verify network access and security, follow these steps:

1. **Check Internet Connectivity (Optional)**: If your private instance has internet access via a NAT gateway or instance, verify by running ping google.com. If there's no internet, it's fine as long as the private instance can communicate with the bastion host.
2. **Inspect Instance Details**: Connect to your private instance and run:
    - hostname to check the instance hostname.
    - ifconfig to verify the private IP address.

```
[ec2-user@ip-10-0-2-68 ~]$ ping google.com
PING google.com (172.253.62.102) 56(84) bytes of data.
^C
--- google.com ping statistics ---
37 packets transmitted, 0 received, 100% packet loss, time 37458ms

[ec2-user@ip-10-0-2-68 ~]$ ^C
[ec2-user@ip-10-0-2-68 ~]$ hostname
ip-10-0-2-68.ec2.internal
[ec2-user@ip-10-0-2-68 ~]$ ifconfig
enX0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.0.2.68  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::1019:f0ff:fe5e:c45b  prefixlen 64  scopeid 0x20<link>
        ether 12:19:f0:5e:c4:5b  txqueuelen 1000  (Ethernet)
        RX packets 1223  bytes 142227 (138.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1531  bytes 159827 (156.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12  bytes 1020 (1020.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 1020 (1020.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```