

## **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: Saravana Krishnan J

Department: IT

# 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.

# Importance

This POC is essential for anyone aiming to:

**1. Enhance Security Skills:** Learn the fundamentals of securing cloud-based architectures by isolating sensitive resources.

**2. Prepare for Real-World Scenarios:** Bastion hosts are frequently used in enterprise environments where private resources need secure access.

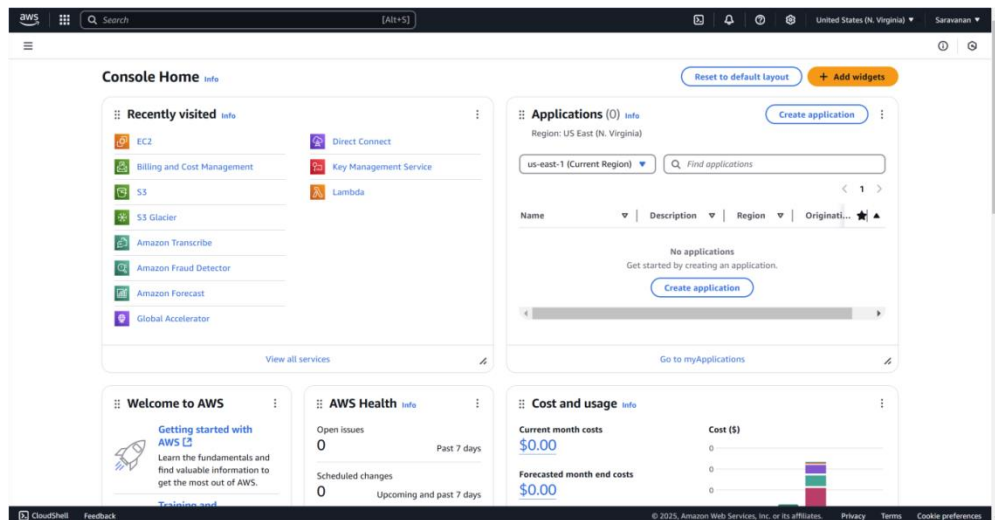
**3. Develop Cloud Expertise:** Gain hands-on experience with AWS services like EC2, VPC, and security groups.

**4. Build Foundational Knowledge:** This knowledge is crucial for advanced cloud topics, such as setting up VPNs, NAT gateways, or using AWS Systems Manager for access.

# Step-by-Step Overview

## Step 1:

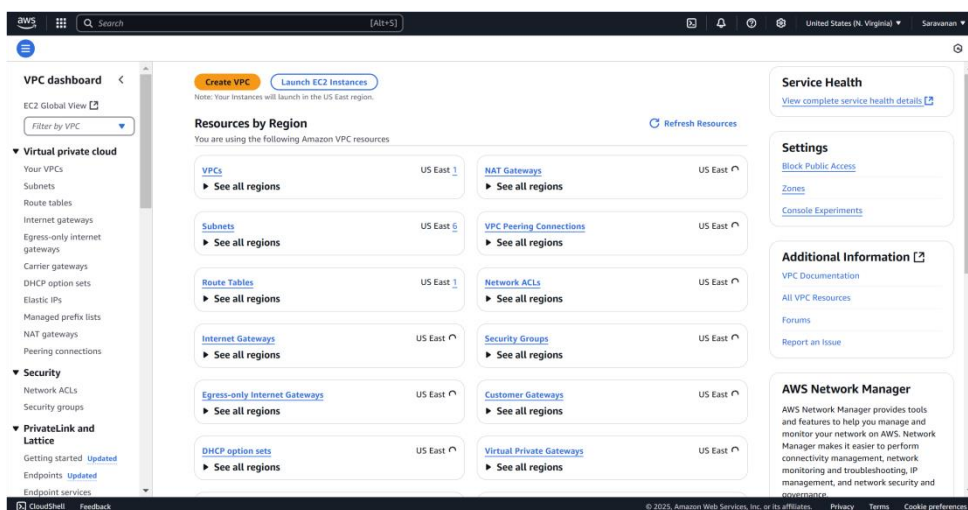
1. Go to [AWS Management Console](#).
2. Enter your username and password to log in.



## 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.

**VPC settings**

**Resources to create** [Info](#)  
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

**Name tag - optional** [Info](#)  
Creates a tag with a key of 'Name' and a value that you specify.

MyBastionVPC

**IPv4 CIDR block** [Info](#)  
☒ IPv4 CIDR manual input  
☐ IPAM-allocated IPv4 CIDR block

**IPv4 CIDR**  
10.0.0.0/16  
CIDR block size must be between /16 and /28.

**IPv6 CIDR block** [Info](#)  
☐ No IPv6 CIDR block  
☐ IPAM-allocated IPv6 CIDR block  
☒ Amazon-provided IPv6 CIDR block  
☐ IPv6 CIDR owned by me

**Network border group** [Info](#)  
A network border group is a unique group of Zones from where IPv4 and IPv6 addresses are advertised. All Availability Zones in this VPC will use this network border group.

us-east-1

**Tenancy** [Info](#)

**Your VPCs (2)** [Info](#)

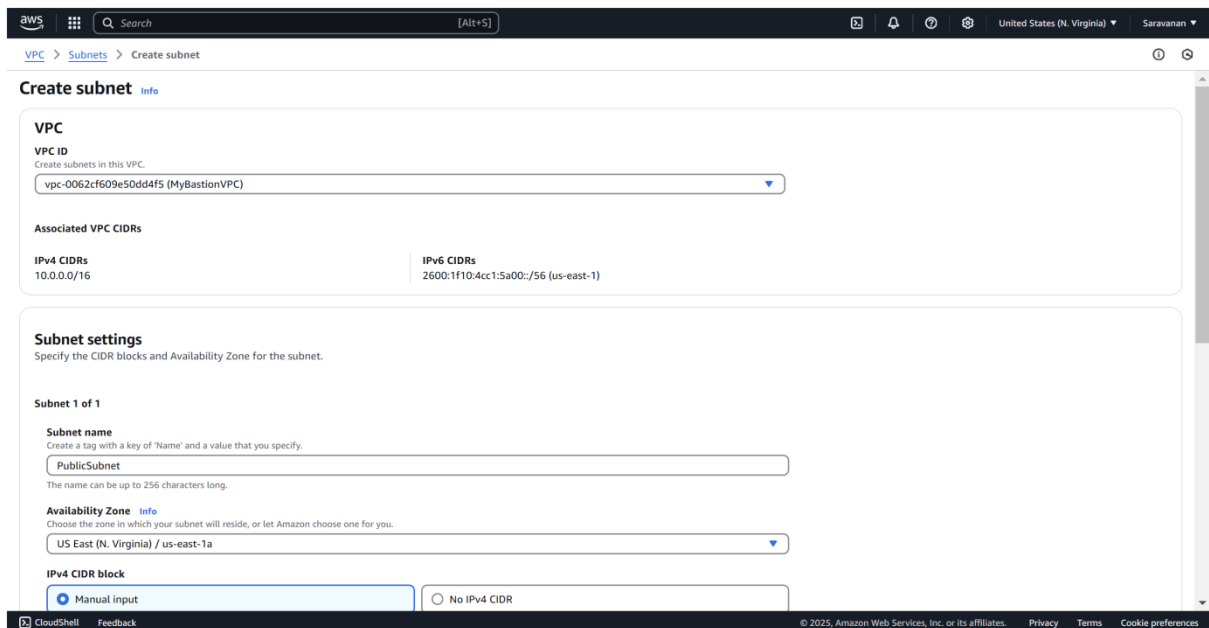
Last updated less than a minute ago [Actions](#) [Create VPC](#)

<input type="checkbox"/>	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	<a href="#">vpc-0f36f0944c17862e5</a>	Available	Off	172.31.0.0/16	-
<input type="checkbox"/>	MyBastionVPC	<a href="#">vpc-0062cf609e50dd4f5</a>	Available	Off	10.0.0.0/16	2600:1f10:4cc1:5a00::/56

Select a VPC above

## Step 4:

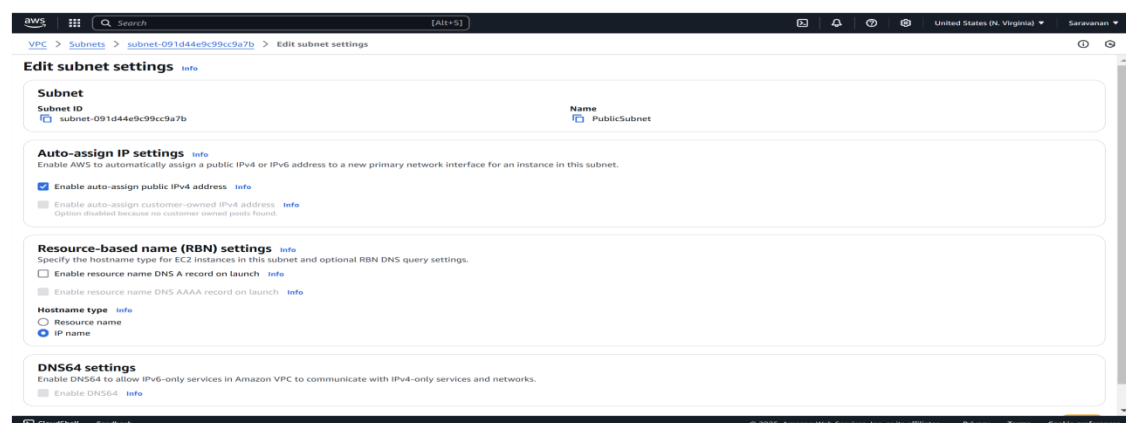
In the **VPC Dashboard**, go to **Subnets** and click **Create Subnet**. Select the **VPC ID** of the VPC you created earlier (*MyBastionVPC*). Enter the **Subnet Name** as *PublicSubnet*, choose an **Availability Zone** (e.g., *us-east-1a*), and set the **IPv4 CIDR Block** as *10.0.1.0/24*. Click **Create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



The screenshot shows the 'Create subnet' page in the AWS Management Console. The 'VPC ID' is set to 'vpc-0062cf609e50dd4f5 (MyBastionVPC)'. Under 'Associated VPC CIDRs', the IPv4 CIDR is '10.0.0.0/16' and the IPv6 CIDR is '2600:1f10:4cc1:5a00::/56 (us-east-1)'. In the 'Subnet settings' section, 'Subnet 1 of 1' is shown with the 'Subnet name' set to 'PublicSubnet'. The 'Availability Zone' is 'US East (N. Virginia) / us-east-1a'. Under 'IPv4 CIDR block', the 'Manual input' radio button is selected.

## Step 5:

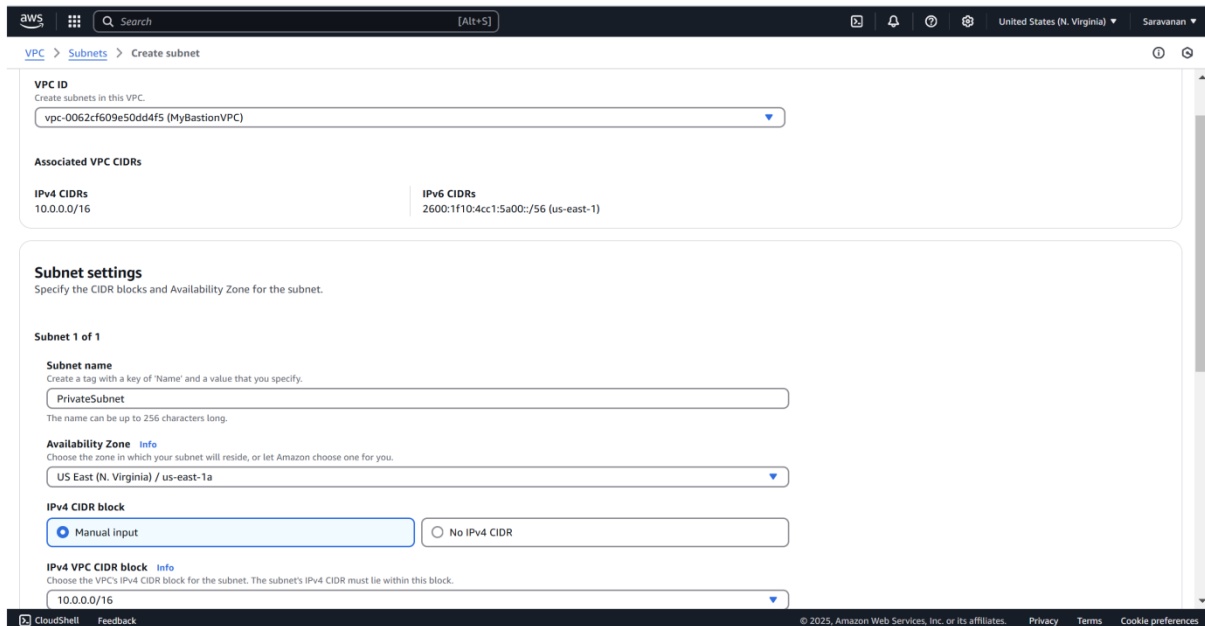
Select your **PublicSubnet** from the list, click **Actions** → **Modify auto-assign IP settings**, check **Enable auto-assign public IPv4 address**, and click **Save**.



The screenshot shows the 'Edit subnet settings' page for the 'PublicSubnet' (subnet-091d44e9c99cc9a7b). Under 'Auto-assign IP settings', the checkbox 'Enable auto-assign public IPv4 address' is checked. Under 'Resource-based name (RBN) settings', the 'IP name' radio button is selected. Under 'DNS64 settings', the checkbox 'Enable DNS64' is checked.

## 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**.



The screenshot shows the AWS Management Console 'Create subnet' page. The 'VPC ID' is set to 'vpc-0062cf609e50dd4f5 (MyBastionVPC)'. The 'Associated VPC CIDRs' section shows 'IPv4 CIDRs' as '10.0.0.0/16' and 'IPv6 CIDRs' as '2600:1f10:4cc1:5a00::/56 (us-east-1)'. Under 'Subnet settings', 'Subnet 1 of 1' is shown. The 'Subnet name' is 'PrivateSubnet'. The 'Availability Zone' is 'US East (N. Virginia) / us-east-1a'. The 'IPv4 CIDR block' is set to 'Manual input' with a value of '10.0.0.0/16'. The 'IPv4 VPC CIDR block' is also set to '10.0.0.0/16'. The 'auto-assign public IP' checkbox is unchecked.

## 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

Value - optional

Q NameX

Q MyInternetGatewayX

Remove

Add new tag

You can add 49 more tags.

Cancel

Create internet gateway

Search

[Alt+S]

United States (N. Virginia)Saravanan

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

Endpoints

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

Actions

Create internet gateway

Q Search

Name	Internet gateway ID	State	VPC ID
-	igw-07cca87f427f8c08f	Attached	vpc-0f16f0944c12862a3
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

Search

[Alt+S]

United States (N. Virginia)Saravanan

VPC

Internet gateways

Attach to VPC (igw-0f2d01304813527f5)

Attach to VPC (igw-0f2d01304813527f5)

info

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

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.

Q vpc-0062cf609e50dd4f5X

AWS Command Line Interface command

Cancel

Attach internet gateway

CloudShellFeedback

© 2025, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie 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**.

The screenshot shows the 'Create route table' page in the AWS Management Console. The 'Route table settings' section has 'Name' set to 'PublicRouteTable' and 'VPC' set to 'vpc-0062cf609e50dd4f5 (MyBastionVPC)'. The 'Tags' section shows a tag with key 'Name' and value 'PublicRouteTable'. At the bottom right, there are 'Cancel' and 'Create route table' buttons.

**Create route table** [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.

PublicRouteTable

**VPC**  
The VPC to use for this route table.

vpc-0062cf609e50dd4f5 (MyBastionVPC)

**Tags**  
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**

Q Name X

**Value - optional**

Q PublicRouteTable X Remove

Add new tag

You can add 49 more tags.

Cancel Create route table

The screenshot shows the 'Route tables' page in the AWS Management Console. A green banner at the top indicates that the route table 'rtb-03e764b95de7259c7 | PublicRouteTable' was created successfully. The 'Route tables (1/3)' table lists three route tables, with 'PublicRouteTable' selected. Below the table, the 'Routes (2)' section shows two routes: one for destination '2600:1f10:4cc1:5a00::/56' and another for '10.0.0.0/16', both with a 'local' target and 'Active' status.

**Route table rtb-03e764b95de7259c7 | PublicRouteTable** was created successfully.

Last updated less than a minute ago [Actions](#) [Create route table](#)

**Route tables (1/3)** [Info](#)

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	
-	rtb-09a4c4f7007b1ed8a	-	-	Yes	vpc-0f36f0944c12862e5	34
<input checked="" type="checkbox"/> PublicRouteTable	rtb-03e764b95de7259c7	-	-	No	vpc-0062cf609e50dd4f5   MyB...	34
-	rtb-013655e0739d83668	-	-	Yes	vpc-0062cf609e50dd4f5   MyB...	34

**rtb-03e764b95de7259c7 / PublicRouteTable**

[Details](#) [Routes](#) [Subnet associations](#) [Edge associations](#) [Route propagation](#) [Tags](#)

**Routes (2)**

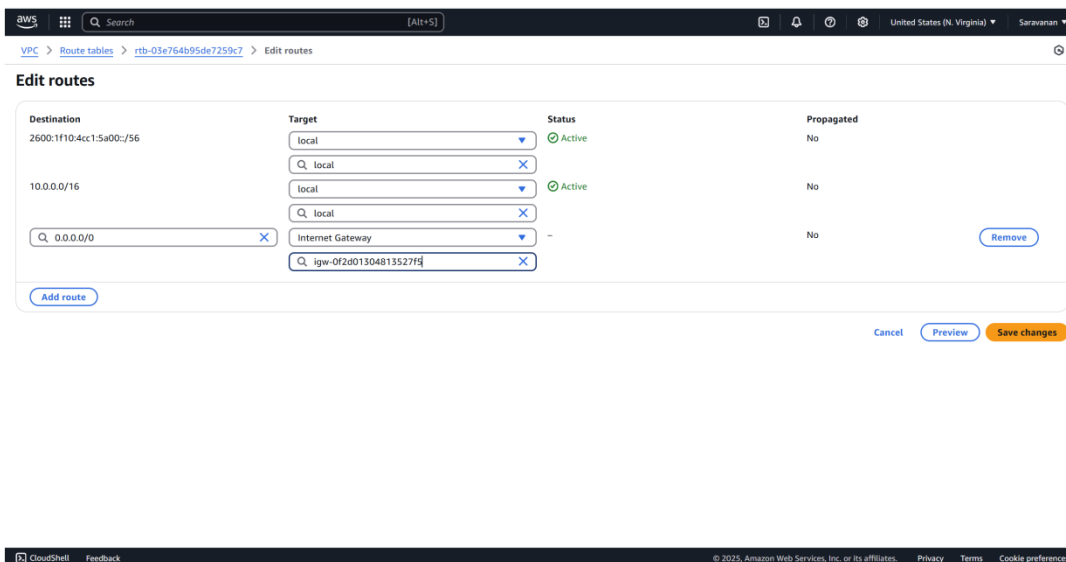
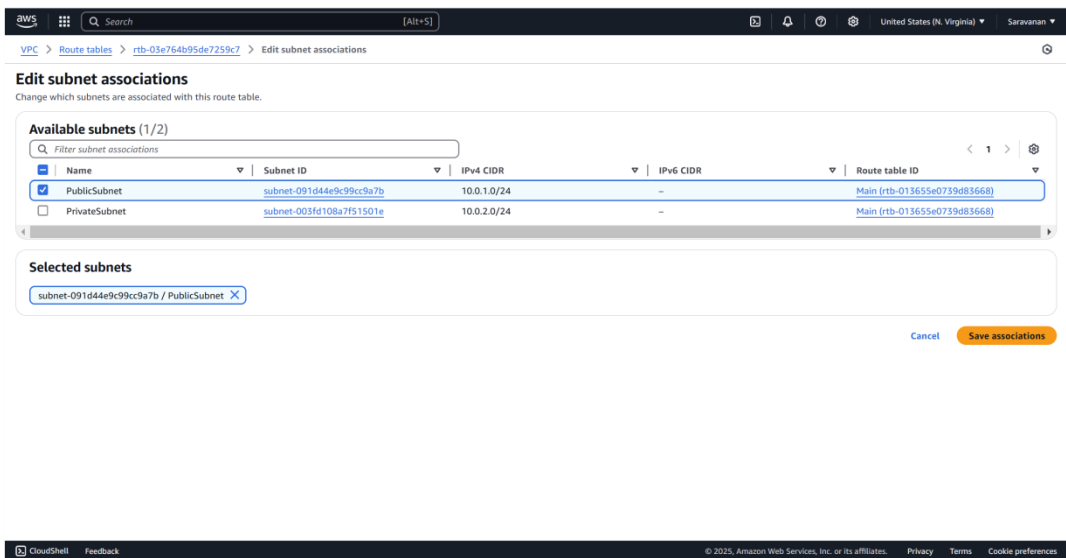
Filter routes

Both Edit routes

Destination	Target	Status	Propagated
2600:1f10:4cc1:5a00::/56	local	Active	No
10.0.0.0/16	local	Active	No

# Step 9:

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



## Step 10:

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**, 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.

### ▼ Network settings [Info](#)

VPC - *required* | [Info](#)

vpc-0062cf609e50dd4f5 (MyBastionVPC)  
10.0.0.0/16 2600:1f10:4cc1:5a00::/56



Subnet | [Info](#)

subnet-091d44e9c99cc9a7b PublicSubnet  
VPC: vpc-0062cf609e50dd4f5 Owner: 343218194491  
Availability Zone: us-east-1a Zone type: Availability Zone  
IP addresses available: 251 CIDR: 10.0.1.0/24  
subnet-091d44e9c99cc9a7b



[Create new subnet](#)

Auto-assign public IP | [Info](#)

Enable

[Additional charges apply](#) when outside of [free tier allowance](#)

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance

☒ Create security group

☐ Select existing security group

Security group name - *required*

launch-wizard-27

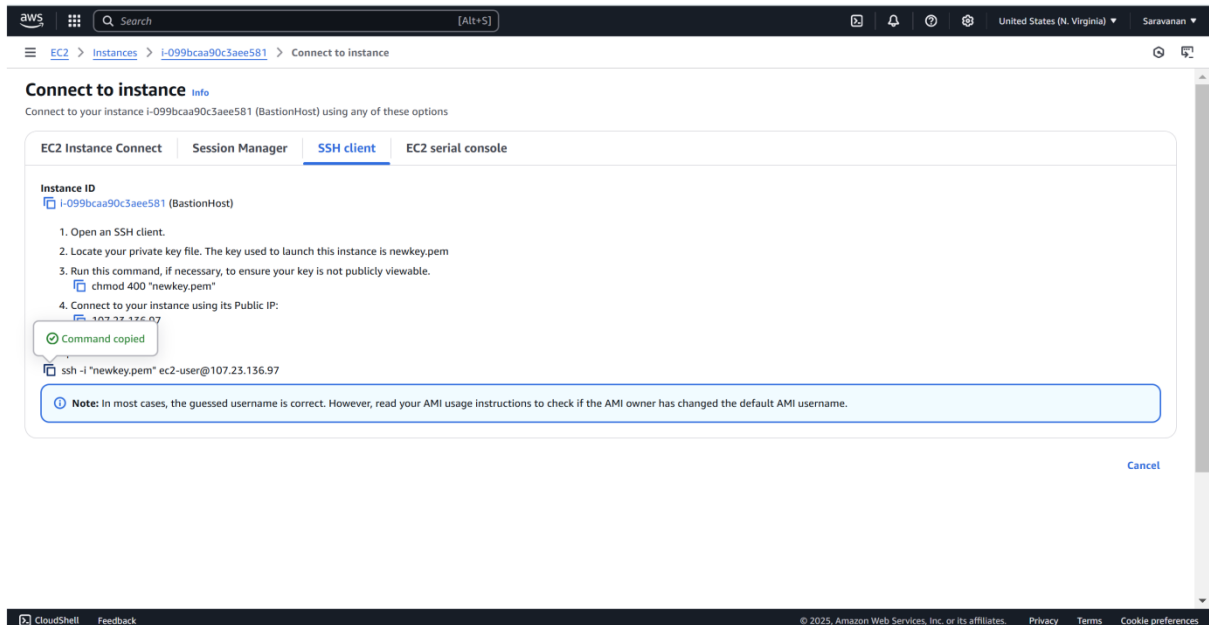
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and . \_ - / ( ) # , @ [ ] + = & ; { } ! \$ \*

Description - *required* | [Info](#)

launch-wizard-27 created 2025-02-06T05:34:23.118Z

## 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
```

## Step 12:

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

```
scp -i /path/to/your-key.pem /path/to/your-key.pem ec2-user@<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-Private-IP>**

```
[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:MGRZMakTZuL8b0oaK307T50//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
```

# 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.
5. Gain a practical understanding of secure cloud networking and foundational AWS services like EC2, VPC, and key-based authentication.