

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: Vishali V

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.

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.

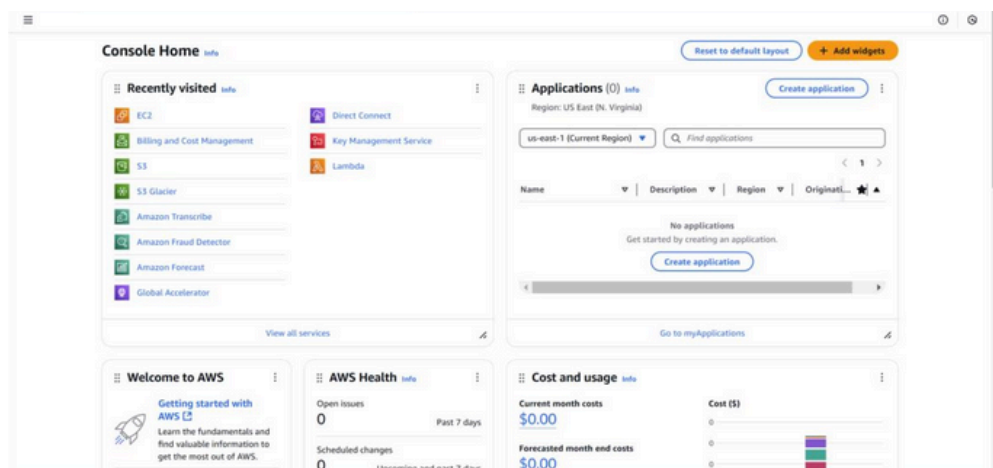
3. Develop Cloud Expertise: Gain hands-on experience with AWS services like EC2, VPC, and security groups.

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:

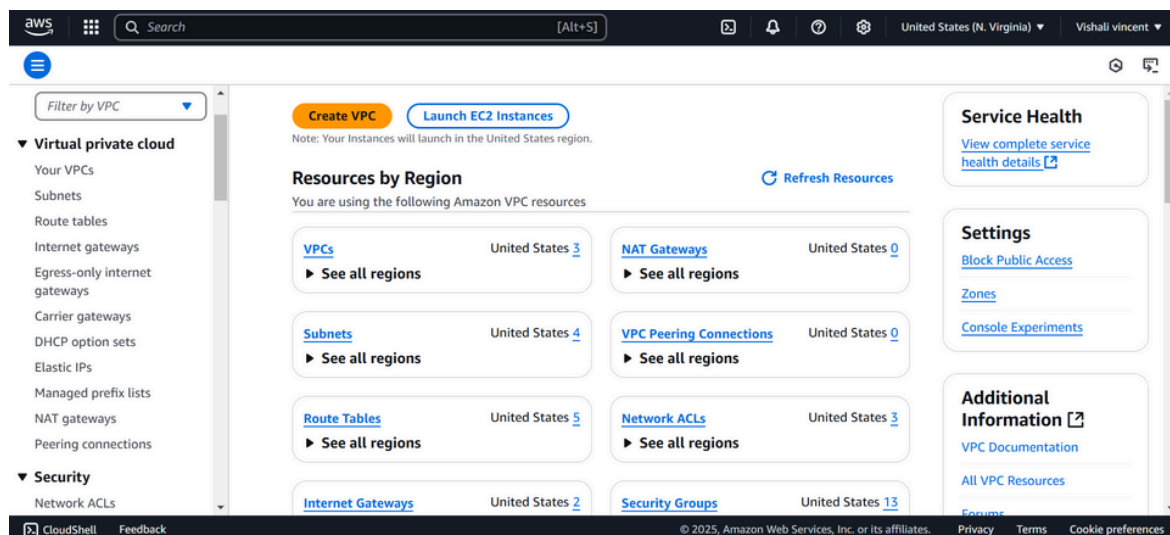
1. Go to [AWS Management Console](#).
2. Enter your username and password to log in.



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.

The screenshot shows the 'Create VPC' page in the AWS Management Console. The page title is 'Create VPC' with an 'Info' link. Below the title, a note states: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.' The 'VPC settings' section contains the following fields:

- Resources to create:** Two radio buttons are present. 'VPC only' is selected, and 'VPC and more' is unselected.
- Name tag - optional:** A text input field contains the value 'my-bastion-vpc'. A small note below the field says: 'Creates a tag with a key of 'Name' and a value that you specify.'
- IPv4 CIDR block:** Two radio buttons are present. 'IPv4 CIDR manual input' is selected, and 'IPAM-allocated IPv4 CIDR block' is unselected.
- IPv4 CIDR:** A text input field contains the value '10.0.0.0/16'. A note below the field states: 'CIDR block size must be between /16 and /28.'

The bottom of the page shows the AWS footer with 'CloudShell', 'Feedback', and copyright information for 2025.

The screenshot shows the 'Your VPCs' page in the AWS Management Console. The page title is 'Your VPCs (1/4)' with an 'Info' link. A search bar is at the top. Below the search bar is a table listing VPCs. The table has columns: Name, VPC ID, State, Block Public..., and IPv4 CIDR. The 'my-bastion-vpc' entry is selected with a blue checkmark. Below the table, the details for 'vpc-03f7b7fd67427962d / my-bastion-vpc' are shown, including tabs for Details, Resource map, CIDRs, Flow logs, Tags, and Integrations. The 'Details' tab is active, showing the VPC ID, State (Available), Block Public Access (Off), and DNS hostnames (Disabled).

| Name | VPC ID | State | Block Public... | IPv4 CIDR |
|----------------|---------------------------------------|-----------|-----------------|---------------|
| vpc | vpc-073a8453f212e8475 | Available | Off | 10.0.0.0/16 |
| - | vpc-09f8b583f34e30082 | Available | Off | 172.31.0.0/16 |
| myvpc | vpc-029a3880b63d3d88b | Available | Off | 10.0.0.0/16 |
| my-bastion-vpc | vpc-03f7b7fd67427962d | Available | Off | 10.0.0.0/16 |

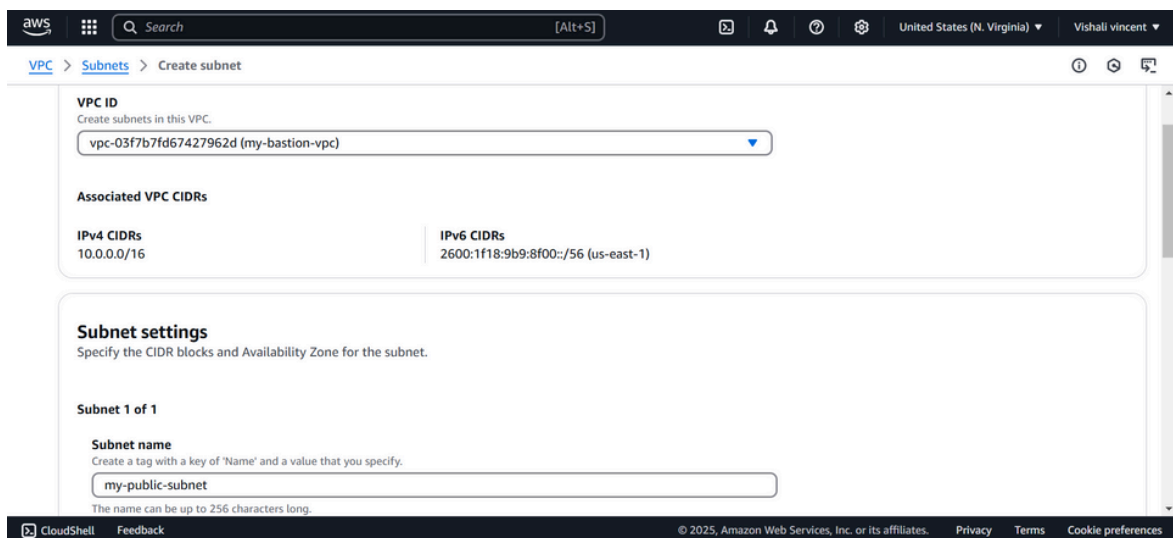
vpc-03f7b7fd67427962d / my-bastion-vpc

Details

| VPC ID | State | Block Public Access | DNS hostnames |
|-----------------------|-----------|---------------------|---------------|
| vpc-03f7b7fd67427962d | Available | Off | Disabled |

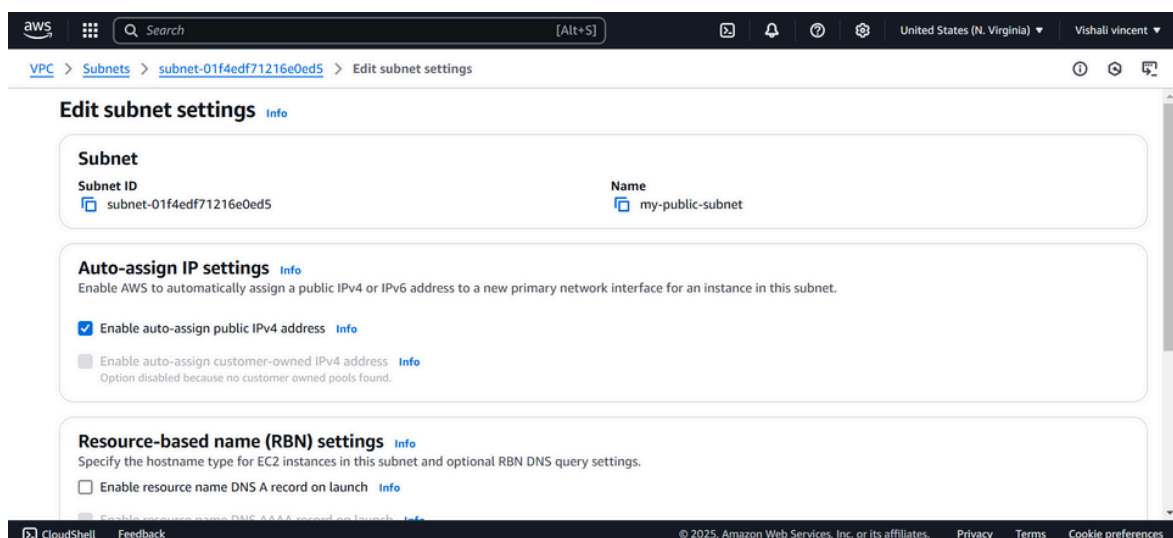
Step 4:

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.



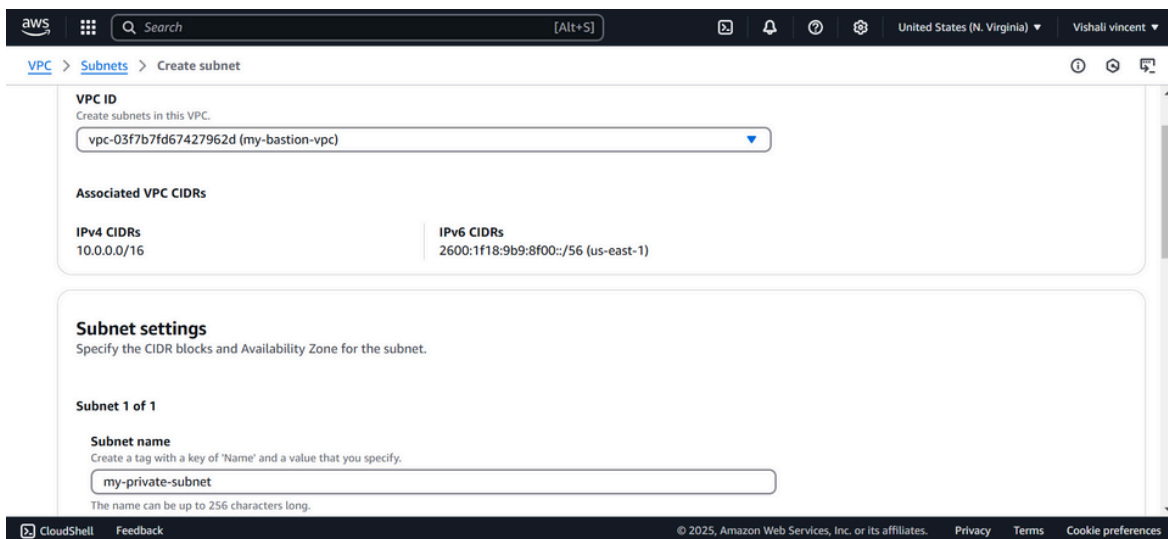
Step 5:

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.



The screenshot shows the AWS Management Console 'Create subnet' page. The breadcrumb navigation at the top indicates the path: VPC > Subnets > Create subnet. The page is divided into several sections:

- VPC ID:** A dropdown menu showing 'vpc-03f7b7fd67427962d (my-bastion-vpc)'.
- Associated VPC CIDRs:** A table with two columns: 'IPv4 CIDRs' and 'IPv6 CIDRs'. The IPv4 CIDR is '10.0.0.0/16' and the IPv6 CIDR is '2600:1f18:9b9:8f00::/56 (us-east-1)'.
- Subnet settings:** A section with the heading 'Subnet settings' and the instruction 'Specify the CIDR blocks and Availability Zone for the subnet.'
- Subnet 1 of 1:** A section for configuring the first subnet.
- Subnet name:** A text input field containing 'my-private-subnet'. Below the field, a note states: 'The name can be up to 256 characters long.'

The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

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.

aws

Search

[Alt+S]

United States (N. Virginia)

Vishali vincent

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.

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

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

aws

Search

[Alt+S]

United States (N. Virginia)

Vishali vincent

VPC > Internet gateways > igw-0be8c08e2e6d91051

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

The following internet gateway was created: igw-0be8c08e2e6d91051 - my-internet-gateway. You can now attach to a VPC to enable the VPC to communicate with the internet.

Attach to a VPC

igw-0be8c08e2e6d91051 / my-internet-gateway

Actions

Attach to VPC

Detach from VPC

Manage tags

Delete

Manage tags

1

Details Info

Internet gateway ID

igw-0be8c08e2e6d91051

State

Detached

VPC ID

-

Owner

970

Tags

Search tags

Key

Name

Value

my-internet-gateway

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws

Search

[Alt+S]

United States (N. Virginia)

Vishali vincent

VPC > Internet gateways > Attach to VPC (igw-0be8c08e2e6d91051)

Attach to VPC (igw-0be8c08e2e6d91051) 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.

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.

The screenshot shows the 'Create route table' page in the AWS Management Console. The page is titled 'Route table settings' and includes the following sections:

- Name - optional:** A text input field containing 'my-public-route-table'.
- VPC:** A dropdown menu showing 'vpc-03f7b7fd67427962d (my-bastion-vpc)'.
- Tags:** A section for adding tags. It shows a key 'Name' and a value 'my-public-route-table'. There is a 'Remove' button and an 'Add new tag' button.

At the bottom right, there are 'Cancel' and 'Create route table' buttons.

The screenshot shows the 'Route table details' page for the route table 'rtb-0ba0496913719a98e / my-public-route-table'. The page is divided into several sections:

- Details:** A section containing key information about the route table, including the Route table ID, VPC, Main status, Owner ID, Explicit subnet associations, and Edge associations.
- Routes:** A section showing the list of routes. It includes a search bar, a table with columns for Destination, Target, Status, and Propagation, and an 'Edit routes' button.

The 'Routes' table shows one route with the following details:

| Destination | Target | Status | Propagation |
|-------------------------|--------|--------|-------------|
| 2600:1f18:9b9:8f00::/56 | 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.

The screenshot shows the 'Edit subnet associations' page in the AWS Management Console. The breadcrumb trail is 'VPC > Route tables > rtb-0ba0496913719a98e > Edit subnet associations'. The page title is 'Edit subnet associations' with a subtitle 'Change which subnets are associated with this route table.' Below this, there are two sections: 'Available subnets (1/2)' and 'Selected subnets'. The 'Available subnets' section contains a table with columns: Name, Subnet ID, IPv4 CIDR, IPv6 CIDR, and Route table ID. Two subnets are listed: 'my-public-subnet' (subnet-01f4edf71216e0ed5, 10.0.1.0/24) and 'my-private-subnet' (subnet-018ab0599ca8216c0, 10.0.2.0/24). The 'my-public-subnet' row is selected with a checked checkbox. The 'Selected subnets' section shows 'subnet-01f4edf71216e0ed5 / my-public-subnet' with a close button. At the bottom right, there are 'Cancel' and 'Save associations' buttons.

| Name | Subnet ID | IPv4 CIDR | IPv6 CIDR | Route table ID |
|--|--------------------------|-------------|-----------|------------------------------|
| <input checked="" type="checkbox"/> my-public-subnet | subnet-01f4edf71216e0ed5 | 10.0.1.0/24 | - | Main (rtb-0f72fef841249890a) |
| <input type="checkbox"/> my-private-subnet | subnet-018ab0599ca8216c0 | 10.0.2.0/24 | - | Main (rtb-0f72fef841249890a) |

The screenshot shows the 'Subnets' page in the AWS Management Console for the subnet 'subnet-01f4edf71216e0ed5'. The breadcrumb trail is 'VPC > Subnets > subnet-01f4edf71216e0ed5'. The left sidebar shows the 'VPC dashboard' with a 'Subnets' link. The main content area has tabs for 'Flow logs', 'Route table', 'Network ACL', 'CIDR reservations', 'Sharing', and 'Tags'. The 'Route table' tab is selected, showing 'Route table: rtb-0ba0496913719a98e / my-public-route-table' with an 'Edit route table association' button. Below this, there is a 'Routes (3)' section with a table showing three routes. The first route has destination '10.0.0.0/16' and target 'local'. The second route has destination '0.0.0.0/0' and target 'igw-0be8c08e2e6d91051'. The third route has destination '2600:1f18:9b9:8f00::/56' and target 'local'.

| Destination | Target |
|-------------------------|-----------------------|
| 10.0.0.0/16 | local |
| 0.0.0.0/0 | igw-0be8c08e2e6d91051 |
| 2600:1f18:9b9:8f00::/56 | local |

Step 10:

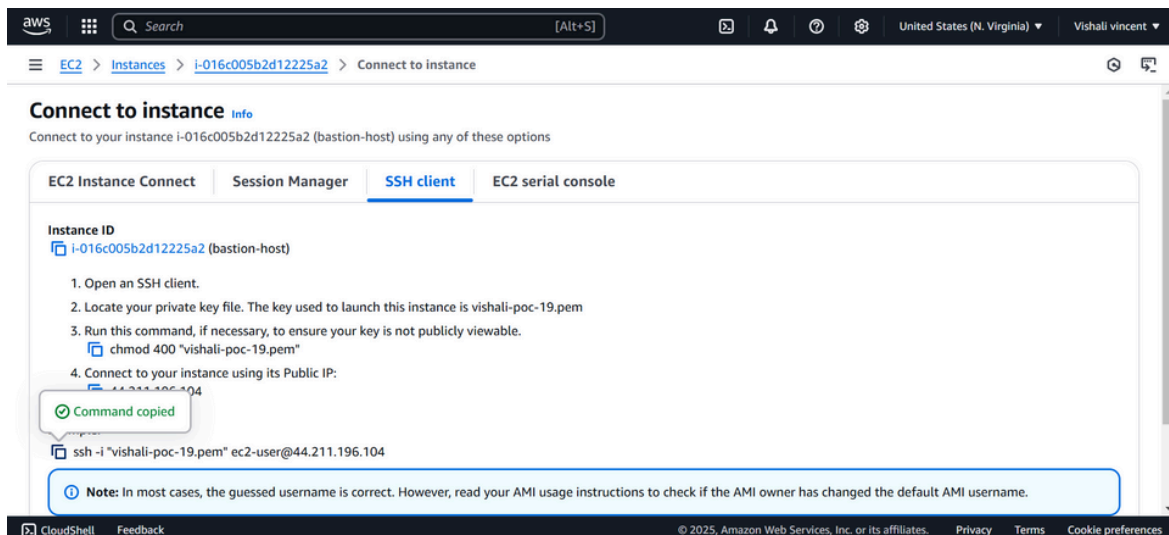
In the EC2 Dashboard, click Launch Instance and configure: set Name as BastionHost, select Amazon Linux 2 AMI (HVM) - Free Tier eligible, and choose t2.micro as the Instance Type. For Key Pair, create or select one, downloading the .pem file if creating. Under Network Settings, select MyBastionVPC for 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.

The screenshot shows the 'Launch an instance' page in the AWS Management Console, specifically the 'Network settings' tab. The 'VPC' is set to 'vpc-03f7b7fd67427962d (my-bastion-vpc)' and the 'Subnet' is 'subnet-01f4edf71216e0ed5 my-public-subnet'. The 'Auto-assign public IP' option is set to 'Enable'. Under 'Firewall (security groups)', the 'Create security group' button is selected. The 'Summary' panel on the right shows 'Number of instances' as 1, 'Software Image (AMI)' as 'Amazon Linux 2023 AMI 2023.6.2...', 'Virtual server type (instance type)' as 't2.micro', and 'Firewall (security group)' as 'New security group'. The 'Launch instance' button is visible at the bottom right.

The screenshot shows the 'Launch an instance' page in the AWS Management Console, specifically the 'Inbound Security Group Rules' tab. A new security group rule is being added with the following details: 'Type' is 'ssh', 'Protocol' is 'TCP', 'Port range' is '22', 'Source type' is 'My IP', and 'Name' is '49.37.221.187/32'. The 'Description - optional' field contains 'e.g. SSH for admin desktop'. The 'Summary' panel on the right is visible, showing the same configuration as the previous screenshot. The 'Launch instance' button is visible at the bottom right.

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:
 - o hostname to check the instance hostname.
 - o 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.