



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: SUBASHINI P

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:

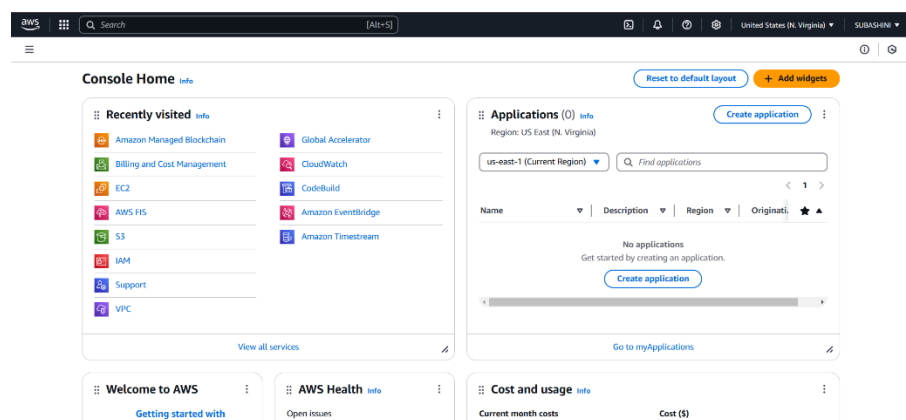
- 1. Enhance Security Skills:** Learn the fundamentals of securing cloud-based architectures by isolating sensitive resources.

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:

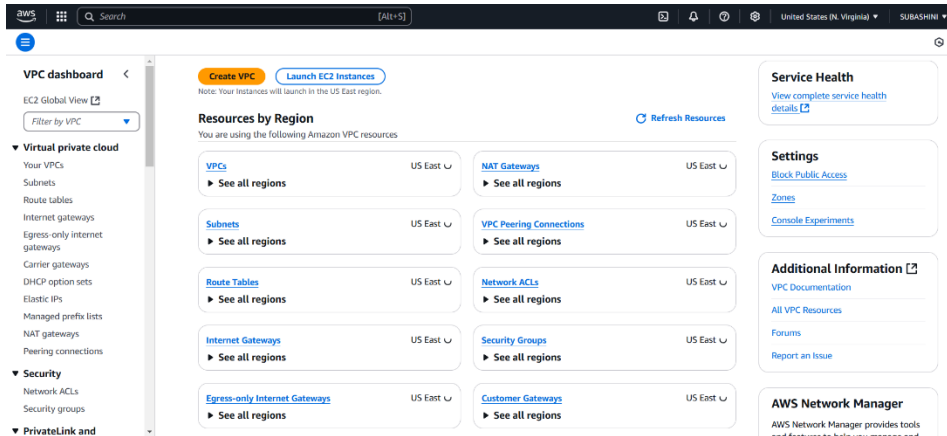
1. Go to [AWS Management Console](#).
2. Enter your username and password to log in.



Step 2:

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

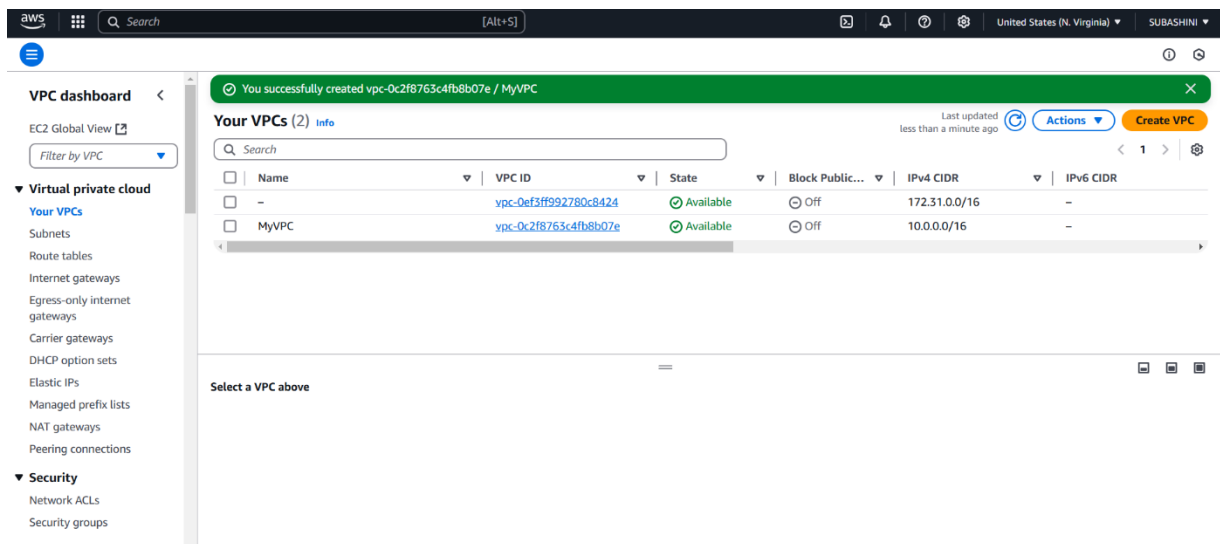
Click on **Create VPC**.



Step 3:

Create a new VPC by selecting **VPC only** and filling in the following details: set the **Name Tag** as *MyVPC* 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.

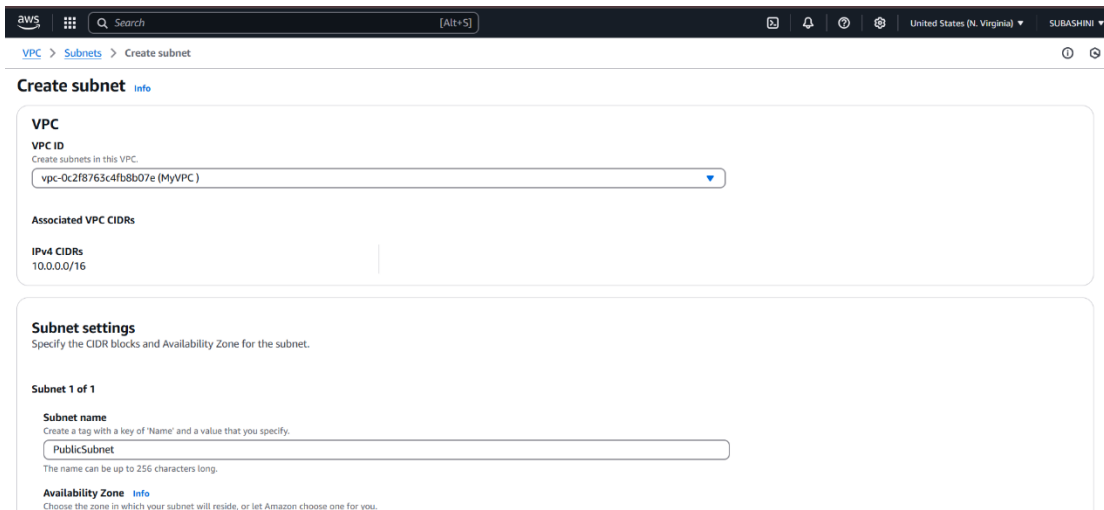
The screenshot shows the 'Create VPC' wizard in the AWS console. The breadcrumb navigation at the top is 'VPC > Your VPCs > Create VPC'. The main heading is 'Create VPC' with an 'Info' link. Below the heading is a descriptive sentence: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.' The 'VPC settings' section is expanded, showing 'Resources to create' with two radio buttons: 'VPC only' (selected) and 'VPC and more'. Under 'Name tag - optional', there is a text input field containing 'MyVPC'. The 'IPv4 CIDR block' section has two radio buttons: 'IPv4 CIDR manual input' (selected) and 'IPAM-allocated IPv4 CIDR block'. Below this, the 'IPv4 CIDR' text input field contains '10.0.0.0/16'. A note states: 'CIDR block size must be between /16 and /28.' The 'IPv6 CIDR block' section has four radio buttons: 'No IPv6 CIDR block' (selected), 'IPAM-allocated IPv6 CIDR block', 'Amazon-provided IPv6 CIDR block', and 'IPv6 CIDR owned by me'.



Step 4:

In the **VPC Dashboard**, go to **Subnets** and click **Create Subnet**.

Select the **VPC ID** of the VPC you created earlier (*MyVPC*). 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**.

The screenshot shows the AWS Management Console interface for editing subnet settings. The breadcrumb trail at the top indicates the path: VPC > Subnets > subnet-0dfe225979493b356 > Edit subnet settings. The page title is 'Edit subnet settings' with an 'Info' link. The main content is divided into three sections: 1. 'Subnet' section showing 'Subnet ID' as subnet-0dfe225979493b356 and 'Name' as PublicSubnet. 2. 'Auto-assign IP settings' section with the instruction 'Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.' It contains two checkboxes: 'Enable auto-assign public IPv4 address' (checked) and 'Enable auto-assign customer-owned IPv4 address' (unchecked, with a note 'Option disabled because no customer owned pools found.'). 3. 'Resource-based name (RBN) settings' section with the instruction 'Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.' It contains two checkboxes: 'Enable resource name DNS A record on launch' (unchecked) and 'Enable resource name DNS AAAA record on launch' (unchecked). At the bottom, the 'Hostname type' is set to 'IP name' (selected) over 'Resource name'.

Edit subnet settings [Info](#)

Subnet

Subnet ID: subnet-0dfe225979493b356

Name: PublicSubnet

Auto-assign IP settings [Info](#)

Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

☒ Enable auto-assign public IPv4 address [Info](#)

☐ Enable auto-assign customer-owned IPv4 address [Info](#)
Option disabled because no customer owned pools found.

Resource-based name (RBN) settings [Info](#)

Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

☐ Enable resource name DNS A record on launch [Info](#)

☐ Enable resource name DNS AAAA record on launch [Info](#)

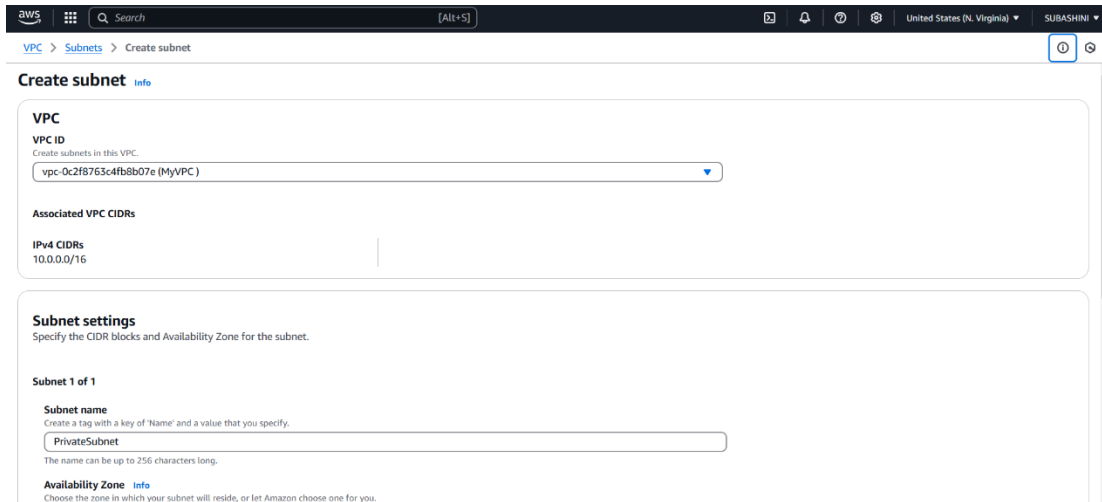
Hostname type [Info](#)

☐ Resource name

☒ IP name

Step 6:

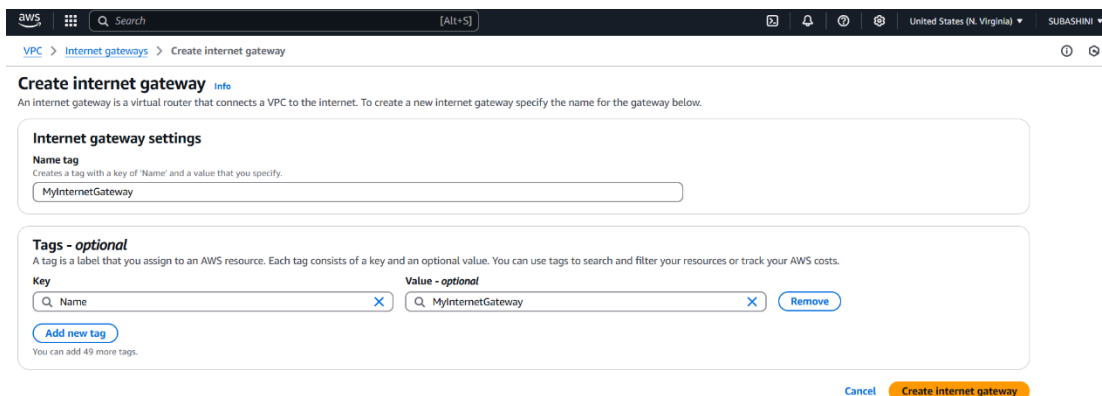
Click **Create Subnet** again and fill in the details: select the same **VPC ID** (*MyVPC*), set **Subnet Name** to *PrivateSubnet*, use the same **Availability Zone** as the public subnet (e.g., *us-east-1a*), and set the **IPv4 CIDR Block** to *10.0.2.0/24*. Leave **auto-assign public IP** disabled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



The screenshot shows the 'Create subnet' page in the AWS Management Console. The page is titled 'Create subnet' with a blue 'info' link. It is divided into two main sections: 'VPC' and 'Subnet settings'. In the 'VPC' section, the 'VPC ID' is set to 'vpc-0c2f8763c4fb8b07e (MyVPC)'. Below it, 'Associated VPC CIDRs' shows 'IPv4 CIDRs' as '10.0.0.0/16'. The 'Subnet settings' section has a subtitle 'Specify the CIDR blocks and Availability Zone for the subnet.' and indicates 'Subnet 1 of 1'. Under 'Subnet name', the name 'PrivateSubnet' is entered. The 'Availability Zone' section is partially visible, showing a dropdown menu.

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 (*MyVPC*), and click **Attach Internet Gateway**.



The screenshot shows the 'Create internet gateway' page in the AWS Management Console. The page is titled 'Create internet gateway' with a blue 'info' link. It includes a subtitle: '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.' The 'Internet gateway settings' section has a 'Name tag' field with the value 'MyInternetGateway'. The 'Tags - optional' section shows a table with one tag: 'Name' as 'MyInternetGateway' and 'Value - optional' as 'MyInternetGateway'. There are 'Add new tag' and 'Remove' buttons. At the bottom right, there are 'Cancel' and 'Create internet gateway' buttons.

aws

Search

[Alt+S]

United States (N. Virginia)

SUBASHINI

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

Internet gateways (1/2)

Info

Search

Name

Internet gateway ID

State

VPC ID

☐

-

igw-0395583b0e542d10b

Attached

vpc-0ef3ff992780c84

☒

MyInternetGateway

igw-02da9bbb09ed3e23f

Detached

-

Actions

Create internet gateway

View details

Attach to VPC

Detach from VPC

Manage tags

Delete internet gateway

igw-02da9bbb09ed3e23f / MyInternetGateway

Details

Tags

Details

Internet gateway ID

igw-02da9bbb09ed3e23f

State

Detached

VPC ID

-

Owner

253490765722

aws

Search

[Alt+S]

United States (N. Virginia)

SUBASHINI

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

VPC

Internet gateways

igw-02da9bbb09ed3e23f

Internet gateway igw-02da9bbb09ed3e23f successfully attached to vpc-0c2f8763c4fb8b07e

igw-02da9bbb09ed3e23f / MyInternetGateway

Actions

Details

Info

Internet gateway ID

igw-02da9bbb09ed3e23f

State

Attached

VPC ID

vpc-0c2f8763c4fb8b07e | MyVPC

Owner

253490765722

Tags

Search tags

Key

Value

Name

MyInternetGateway

Manage tags

Step 8:

In the **VPC Dashboard**, go to **Route Tables** and click **Create Route Table**. Name it *PublicRouteTable*, select your VPC (*MyVPC*), 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**.

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.

VPC
The VPC to use for this route table.

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 **Value - optional**

[Remove](#)

[Add new tag](#)
You can add 49 more tags.

[Cancel](#) [Create route table](#)

Updated routes for rtb-0454e793a9ac820e9 / PublicRouteTable successfully

[Details](#)

rtb-0454e793a9ac820e9 / PublicRouteTable [Actions](#)

Details [info](#)

Route table ID

VPC

Owner ID

Main

Explicit subnet associations
-

Edge associations
-

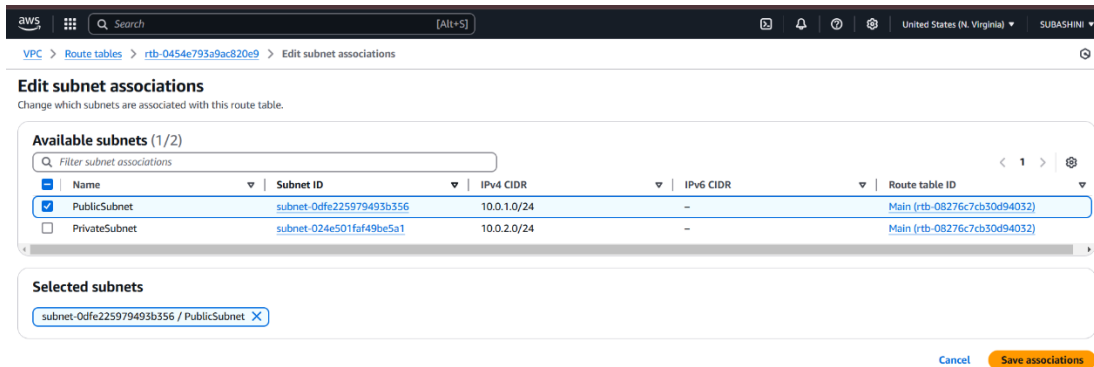
Routes [Subnet associations](#) [Edge associations](#) [Route propagation](#) [Tags](#)

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-02da9bbb09ed3e23f	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

EC2

Instances

Launch an instance

▼ Network settings

VPC - required

vpc-0c2f8765c4fb8b07e (MyVPC)

Subnet

subnet-0dfe225979493b356

PublicSubnet

Auto-assign public IP

Enable

Firewall (security groups)

Create security group

Select existing security group

Security group name - required

launch-wizard-13

Description - required

launch-wizard-13 created 2025-02-12T14:01:04.604Z

▼ Summary

Number of instances

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.6.2...read more

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier

In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month. 750 hours of public IP add

Cancel

Launch instance

Preview code

```
Last login: Wed Feb  5 09:20:14 2025 from 182.74.154.218

_#_
#\   ####_
### \#####\
###  \###|
###   \|
###    V~'  --->
      /
     / 
    /  
   /   
  /    
 /      
/_/____/_____
/_/____/_____
/_/____/_____
/_/____/_____

Amazon Linux 2

AL2 End of Life is 2026-06-30.

A newer version of Amazon Linux is available!

Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-1-218 ~]$ chmod 400 sam.pem
[ec2-user@ip-10-0-1-218 ~]$ ssh -i sam.pem ec2-user@10.0.1.218
The authenticity of host '10.0.1.218 (10.0.1.218)' can't be established.
ECDSA key fingerprint is SHA256:Y6FPLIZ5IAKtMNwnbL3Yq1SXQPKRyey1HTZPbylOrLY.
ECDSA key fingerprint is MD5:d4:a6:0d:fa:99:92:df:21:ca:36:0f:39:5f:ed:ba:cd.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.218' (ECDSA) to the list of known hosts.
Last login: Wed Feb  5 14:18:12 2025 from 223.178.84.112

_#_
#\   ####_
### \#####\
###  \###|
###   \|
###    V~'  --->
      /
     / 
    /  
   /   
  /    
 /      
/_/____/_____
/_/____/_____
/_/____/_____
/_/____/_____

Amazon Linux 2

AL2 End of Life is 2026-06-30.

A newer version of Amazon Linux is available!

Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-1-218 ~]$ |
```

Step 12:

Disable Password Authentication

1. Edit SSH config
2. find and update these lines: passwordAuthentication no
PermitRootLogin no
3. Restart SSH service

```
#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no

# Kerberos options
```

Step 13:

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.

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.