St. JOSEPH'S
GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

aws

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: Hema S
Department: ECE

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY

AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

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.

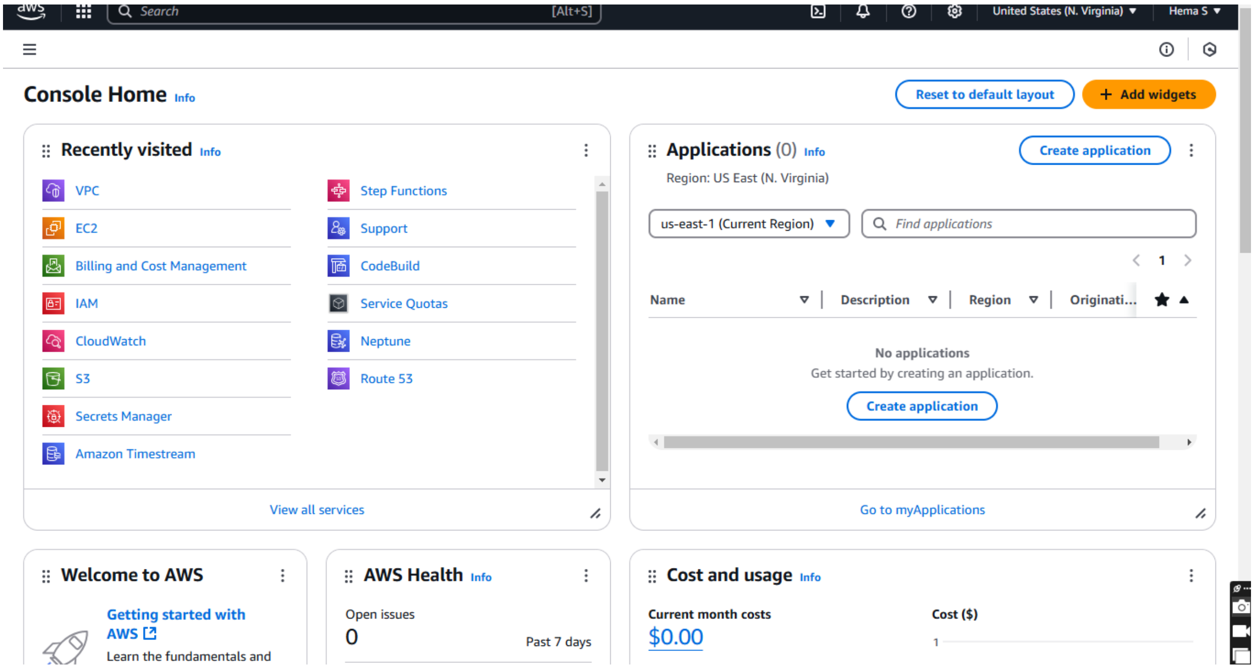
4. **Build Foundational Knowledge:** This knowledge is crucial for advanced cloud topics, such as setting up VPNs, NAT gateways, or using AWS Systems Manager for access.

Step-by-Step Overview

Step 1:

1. Go to AWS Management Console.

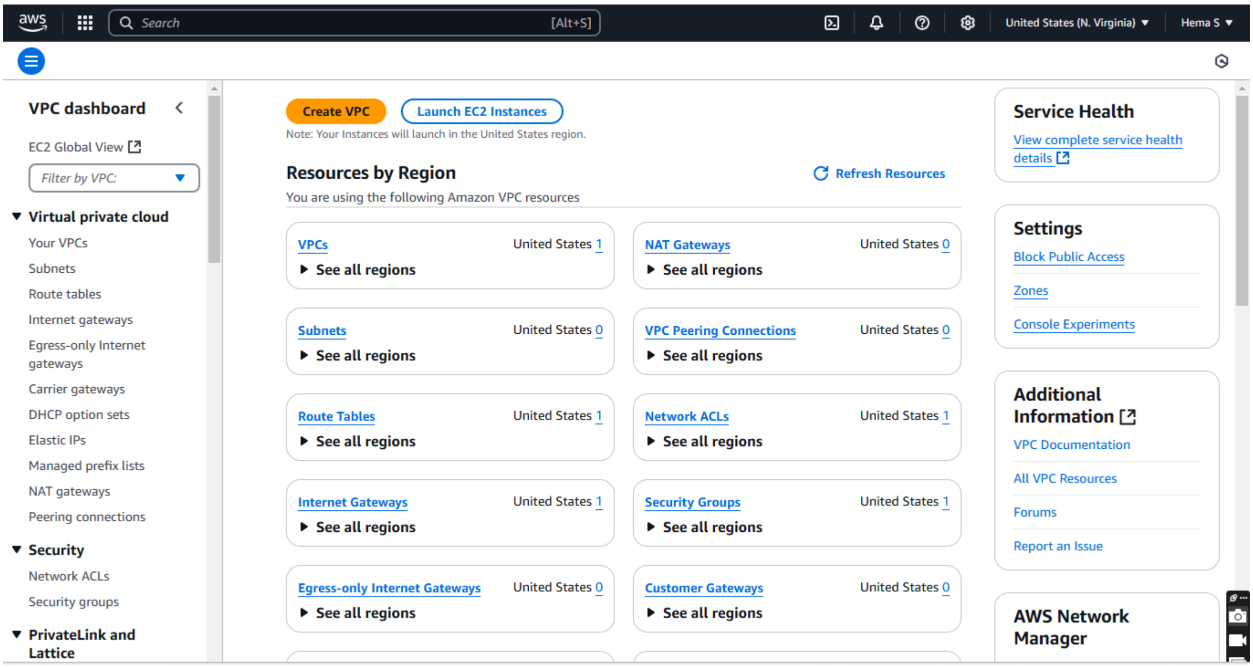
2. Enter your username and password to log in.



Step 2:

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

Click on **Create VPC**.



Search [Alt+S]

United States (N. Virginia) ▼ Hema S ▼

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

Create VPC | Launch EC2 Instances

Note: Your Instances will launch in the United States region.

Resources by Region — Refresh Resources

You are using the following Amazon VPC resources

VPCs	United States 1	NAT Gateways	United States 0
Subnets	United States 0	VPC Peering Connections	United States 0
Route Tables	United States 1	Network ACLs	United States 1
Internet Gateways	United States 1	Security Groups	United States 1
Egress-only Internet Gateways	United States 0	Customer Gateways	United States 0

▶ See all regions

Service Health

View complete service health details

Settings

- Block Public Access
- Zones
- Console Experiments

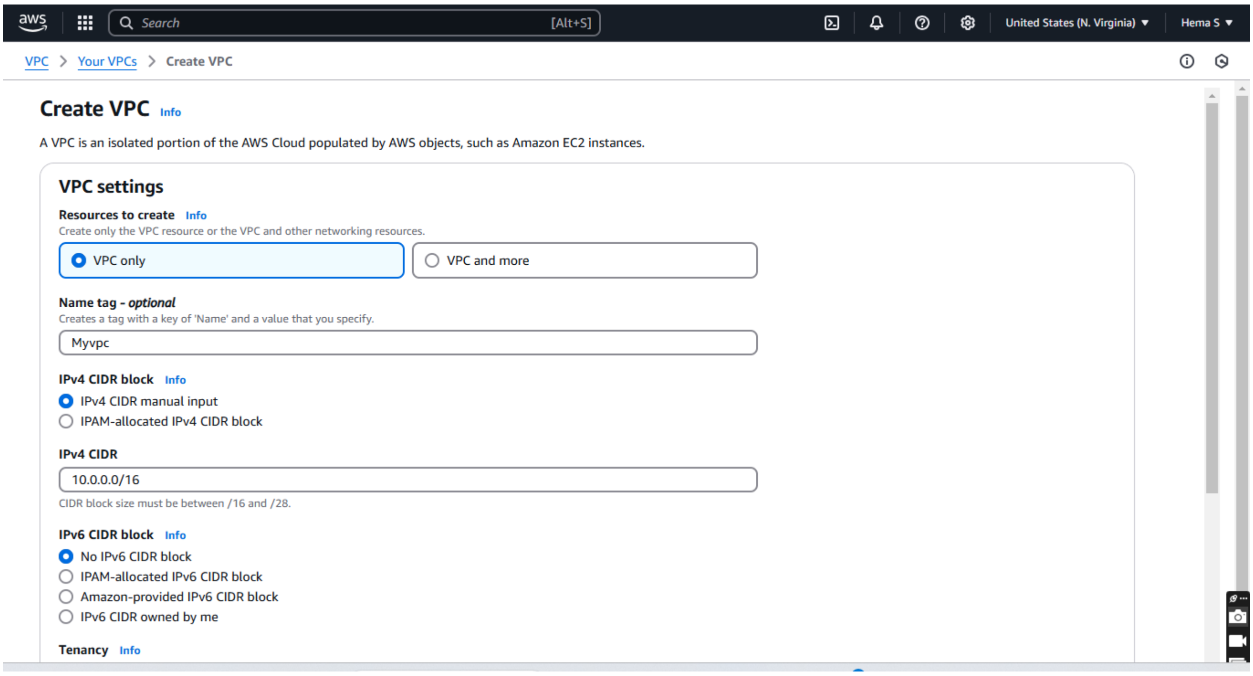
Additional Information

- VPC Documentation
- All VPC Resources
- Forums
- Report an Issue

AWS Network Manager

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.



Search [Alt+S]

United States (N. Virginia) ▼ Hema S ▼

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

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*

Creates a tag with a key of 'Name' and a value that you specify.

Myvpc

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

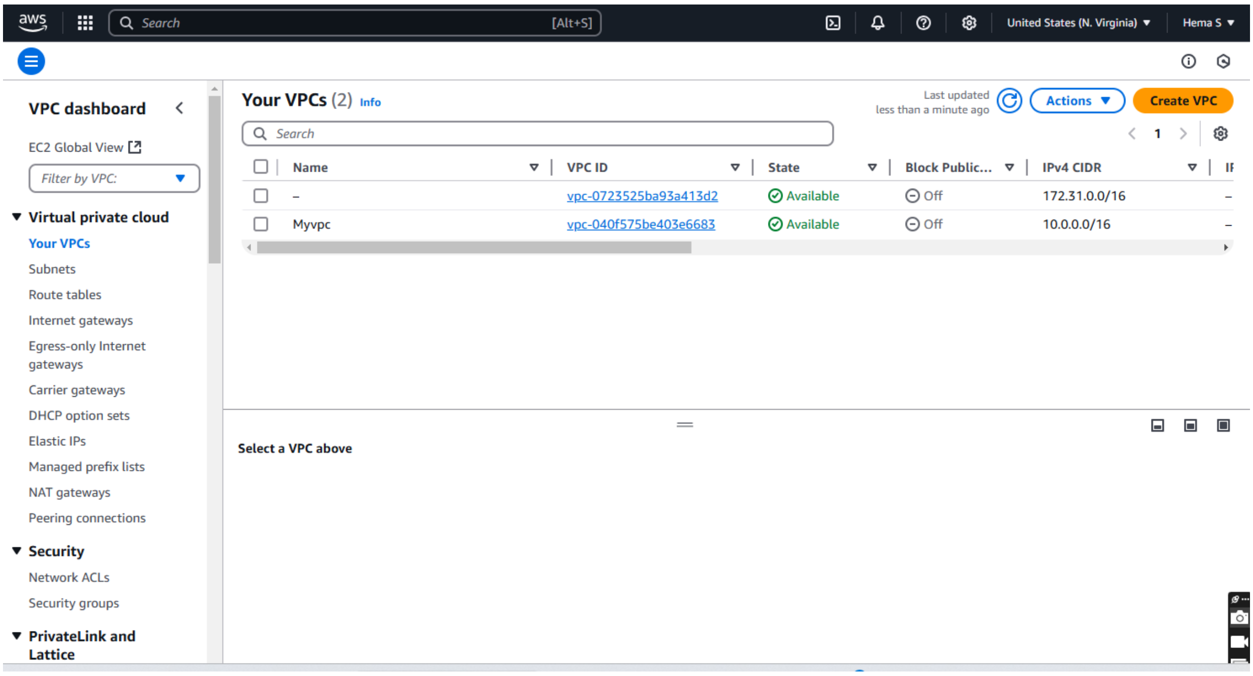
● No IPv6 CIDR block

○ IPAM-allocated IPv6 CIDR block

○ Amazon-provided IPv6 CIDR block

○ IPv6 CIDR owned by me

Tenancy Info



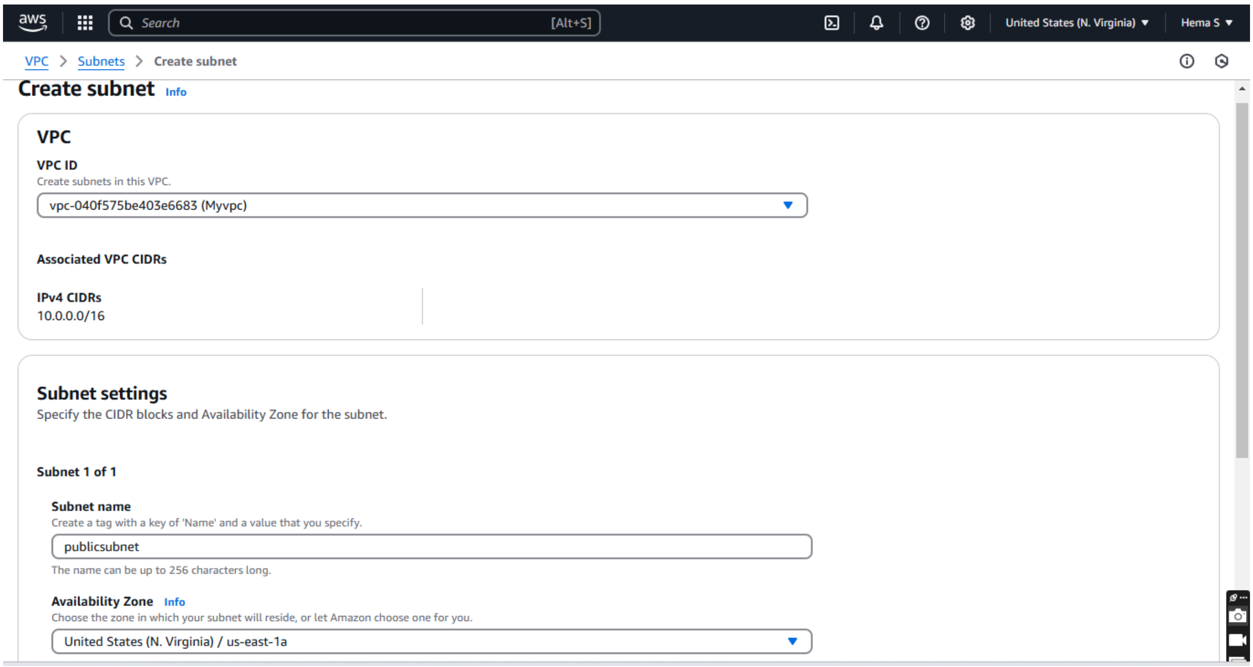
Your VPCs (2) Info

Name	VPC ID	State	Block Public...	IPv4 CIDR	IF
–	vpc-0723525ba93a413d2	Available	Off	172.31.0.0/16	–
Myvpc	vpc-040f575be403e6683	Available	Off	10.0.0.0/16	–

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 (*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**.



Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-040f575be403e6683 (Myvpc)

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

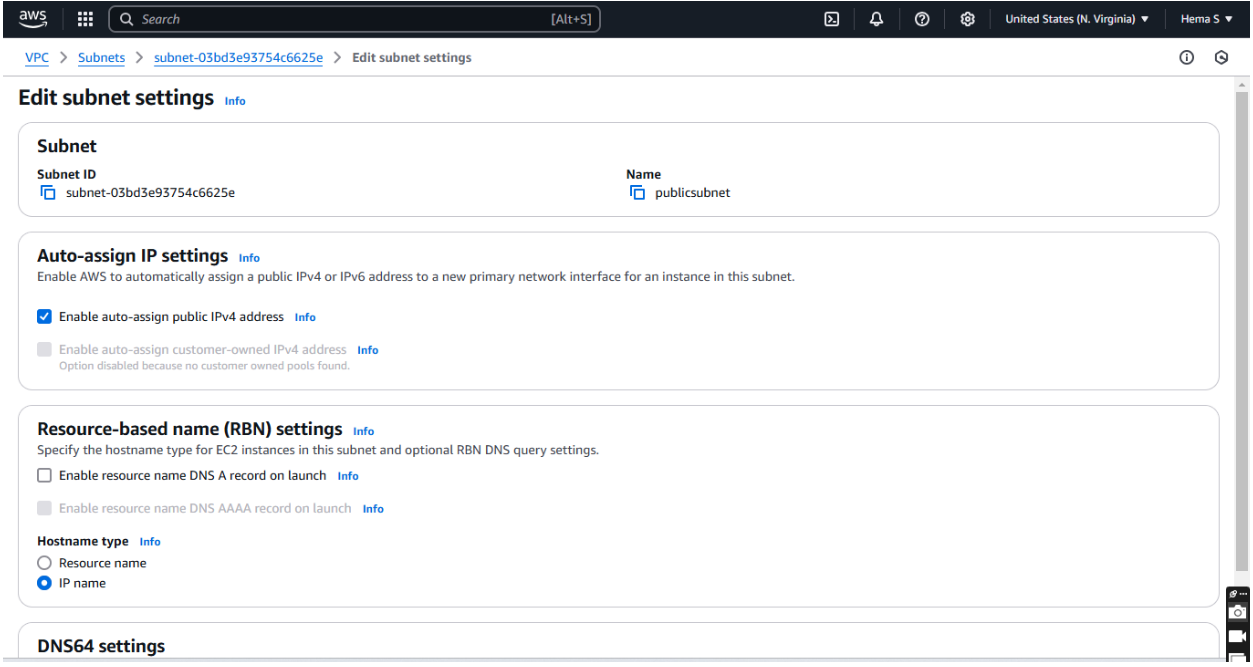
Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
publicsubnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. Virginia) / us-east-1a

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**.



aws
Search [Alt+S]
United States (N. Virginia) ▼
Hema S ▼
VPC > Subnets > subnet-03bd3e93754c6625e > Edit subnet settings

Edit subnet settings Info

Subnet

Subnet ID: subnet-03bd3e93754c6625e

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

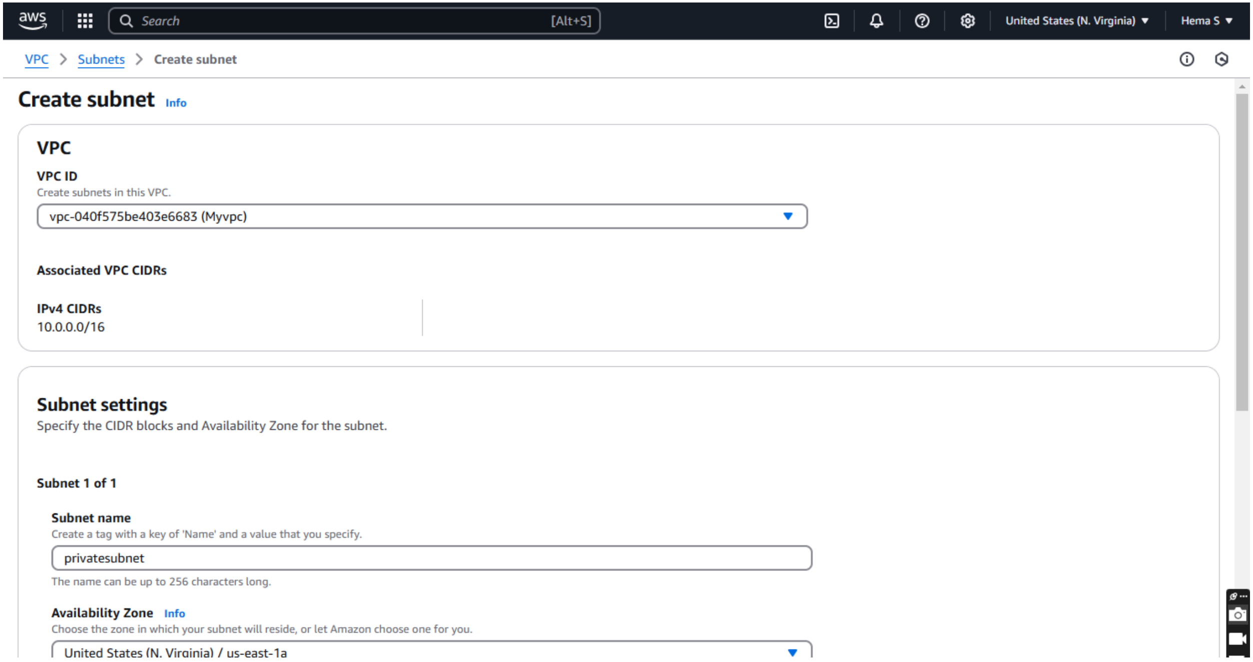
□ Enable resource name DNS AAAA record on launch Info

Hostname type Info
○ Resource name
● IP name

DNS64 settings

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**.



Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-040f575be403e6683 (Myvpc)

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

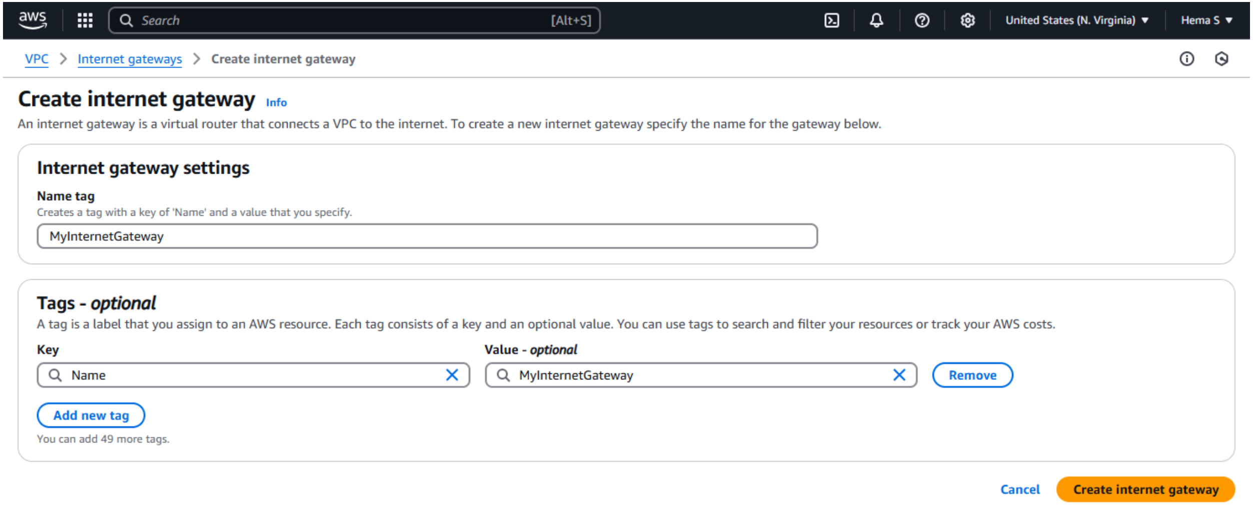
Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
privatesubnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. Virginia) / us-east-1a

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**.



aws
Search [Alt+S]
United States (N. Virginia)
Hema S
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.

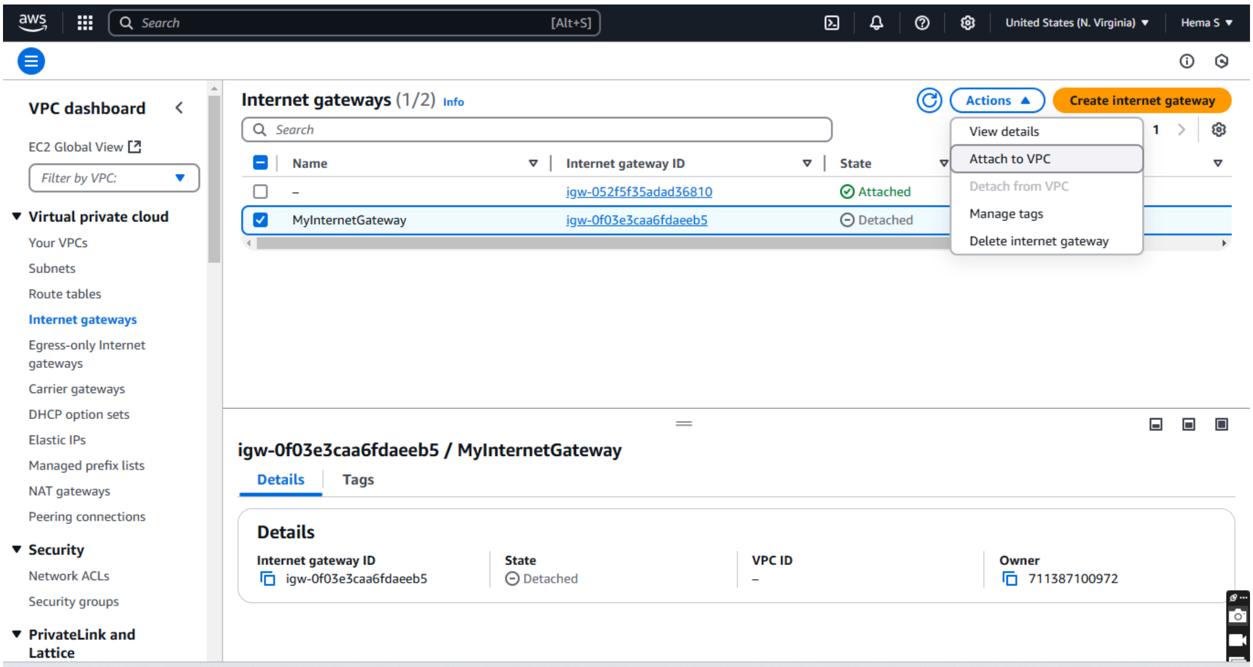
Internet gateway settings

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*
Name	MyInternetGateway

You can add 49 more tags.



Internet gateways (1/2) Info

Actions ▲ | Create internet gateway

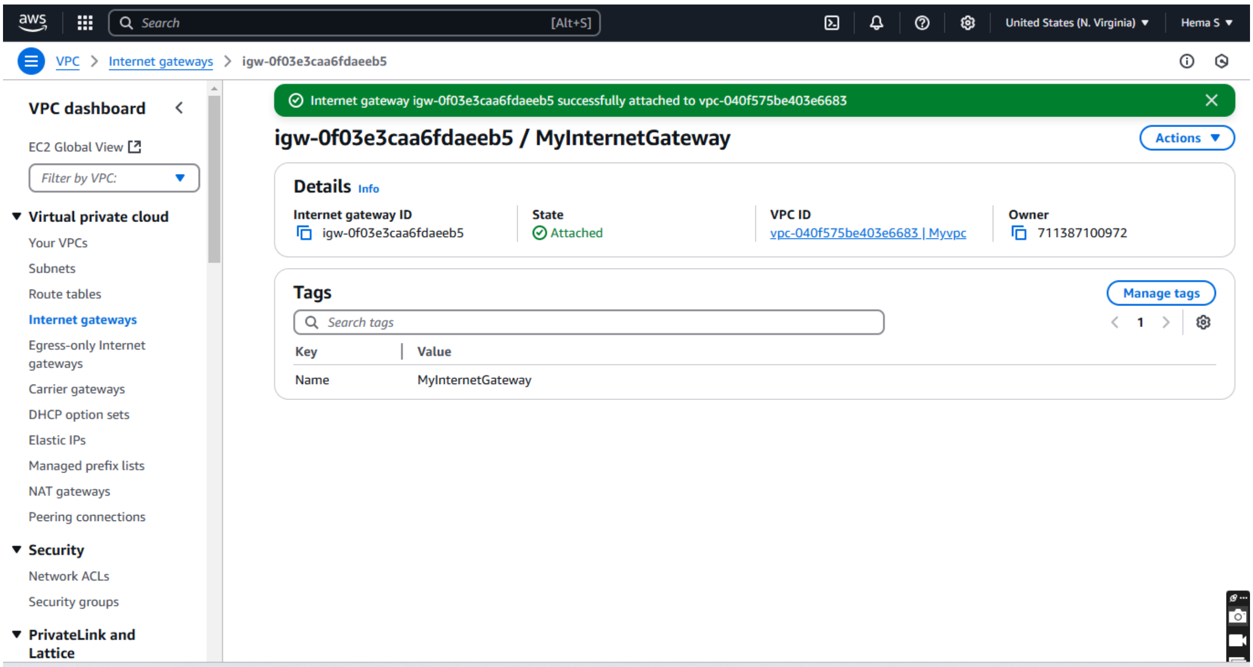
	Name	Internet gateway ID	State
□	–	igw-052f5f35adad36810	Attached
☑	MyInternetGateway	igw-0f03e3caa6fdaeeb5	Detached

igw-0f03e3caa6fdaeeb5 / MyInternetGateway

Details | Tags

Details

Internet gateway ID	State	VPC ID	Owner
igw-0f03e3caa6fdaeeb5	Detached	–	711387100972



VPC > Internet gateways > igw-0f03e3caa6fdaeeb5

Internet gateway igw-0f03e3caa6fdaeeb5 successfully attached to vpc-040f575be403e6683

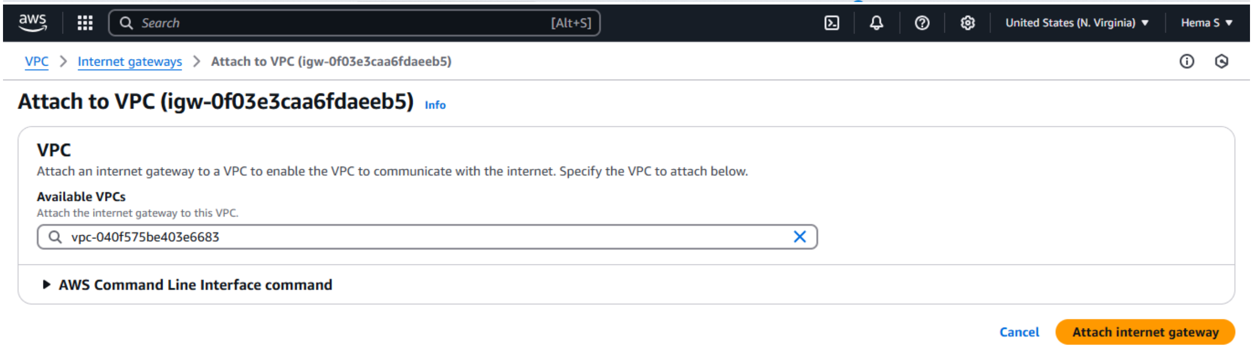
igw-0f03e3caa6fdaeeb5 / MyInternetGateway

Details Info

Internet gateway ID	State	VPC ID	Owner
igw-0f03e3caa6fdaeeb5	Attached	vpc-040f575be403e6683 | Myvpc	711387100972

Tags

Key	Value
Name	MyInternetGateway



VPC > Internet gateways > Attach to VPC (igw-0f03e3caa6fdaeeb5)

Attach to VPC (igw-0f03e3caa6fdaeeb5) 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

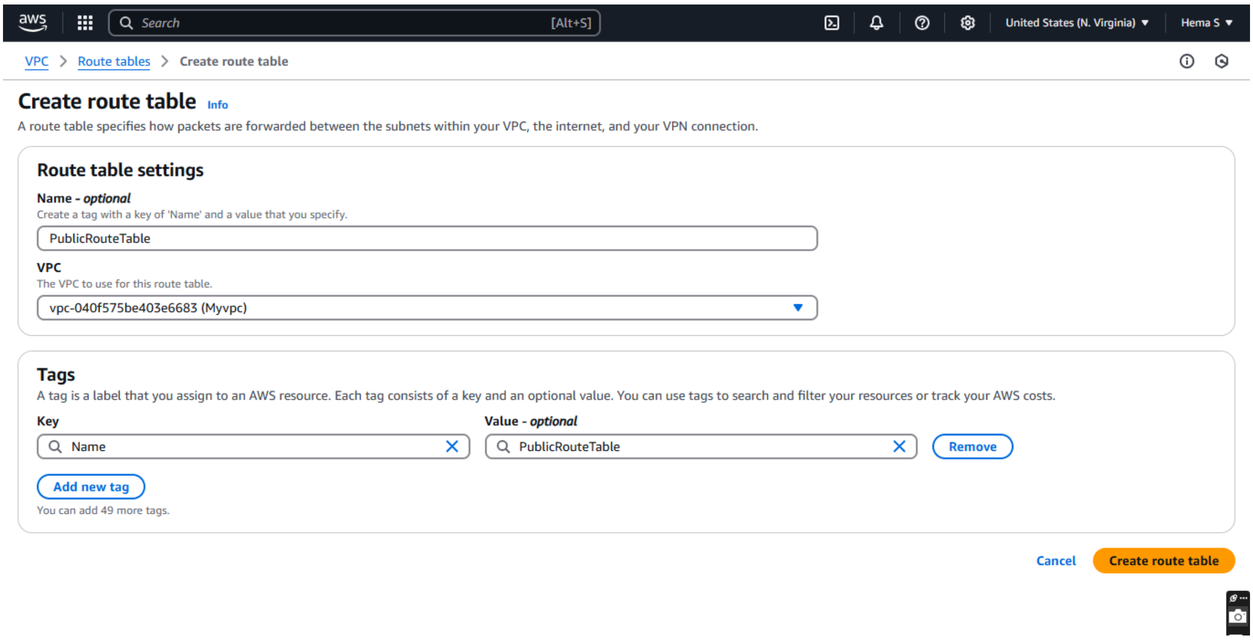
Attach the internet gateway to this VPC.

vpc-040f575be403e6683

▶ AWS Command Line Interface command

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**.


aws
Search [Alt+S]
United States (N. Virginia) ▼
Hema S ▼
VPC > Route tables > Create route table

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

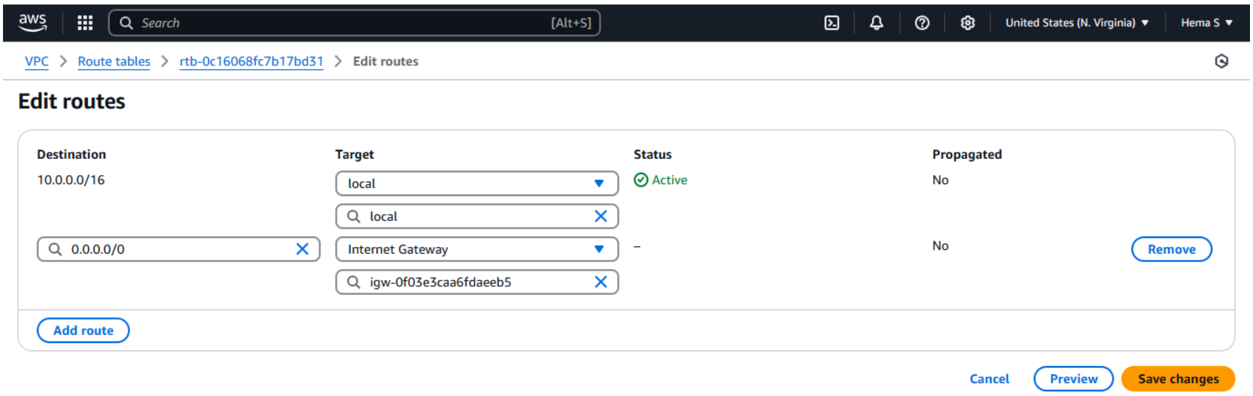
VPC
The VPC to use for this route table.
vpc-040f575be403e6683 (Myvpc)

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.

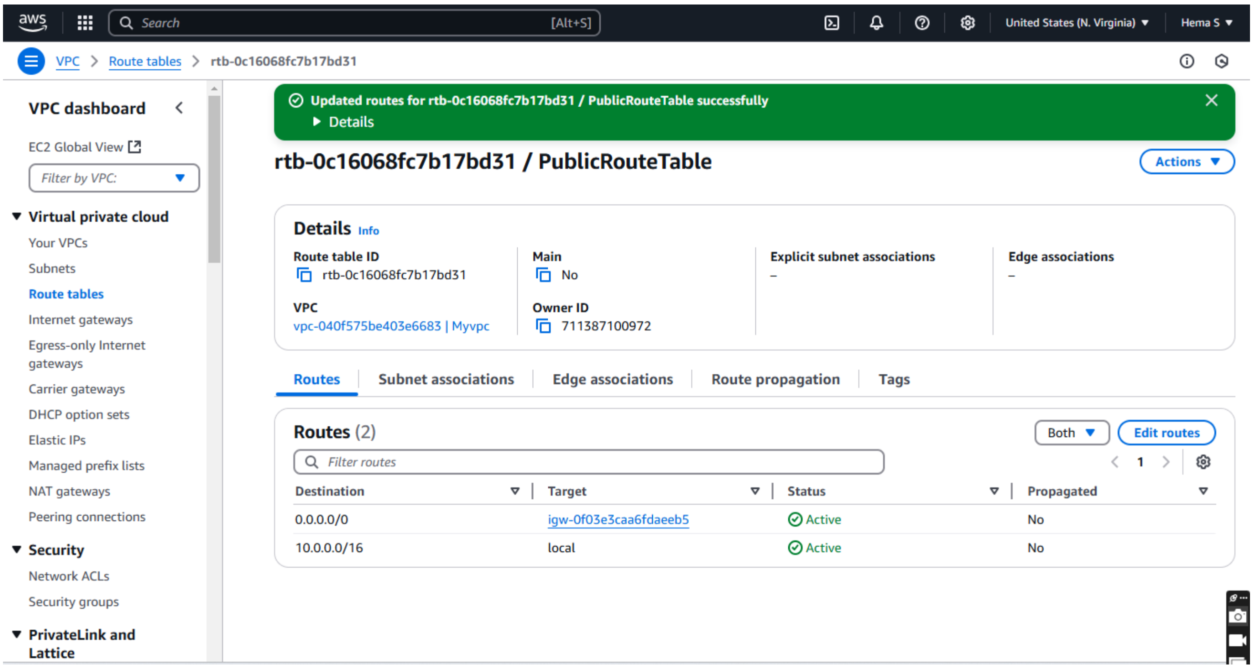
Key	Value - optional
Name	PublicRouteTable

You can add 49 more tags.


aws
Search [Alt+S]
United States (N. Virginia) ▼
Hema S ▼
VPC > Route tables > rtb-0c16068fc7b17bd31 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local local	✓ Active	No
0.0.0.0/0	Internet Gateway igw-0f03e3caa6fdaeeb5	–	No



Updated routes for rtb-0c16068fc7b17bd31 / PublicRouteTable successfully

▶ Details

rtb-0c16068fc7b17bd31 / PublicRouteTable

Actions ▼

Details Info

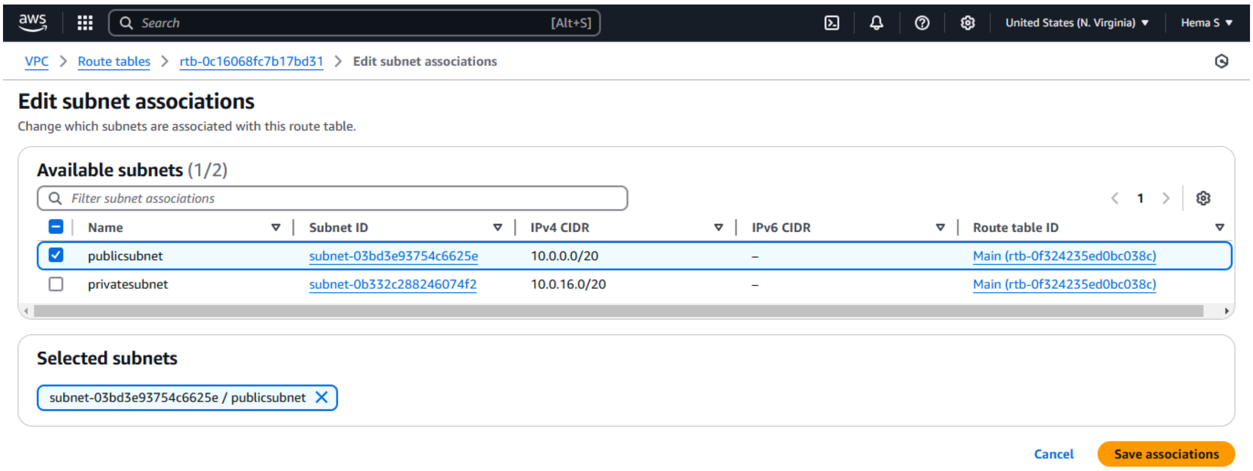
Route table ID rtb-0c16068fc7b17bd31	Main No	Explicit subnet associations –	Edge associations –
VPC vpc-040f575be403e6683 | Myvpc	Owner ID 711387100972		

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0f03e3caa6fdaeeb5	✓ 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**.


aws
Search
[Alt+S]
United States (N. Virginia) ▼
Hema S ▼
VPC > Route tables > rtb-0c16068fc7b17bd31 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)

Filter subnet associations

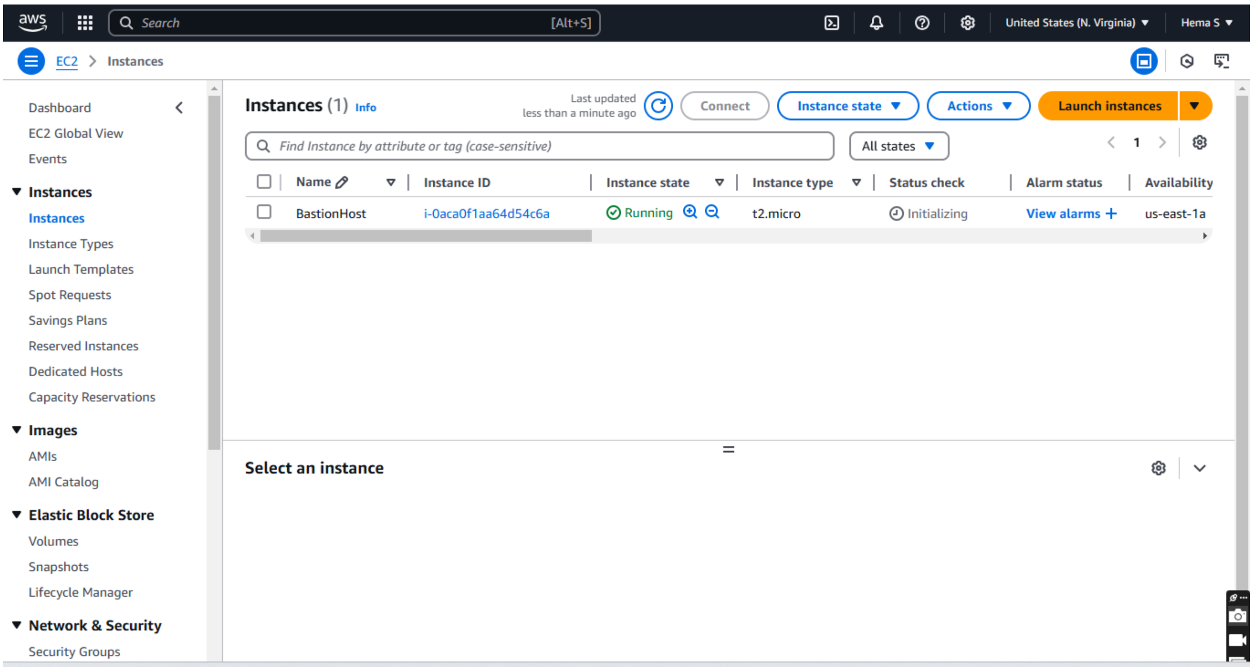
	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
☑	publicsubnet	subnet-03bd3e93754c6625e	10.0.0.0/20	–	Main (rtb-0f324235ed0bc038c)
□	privatesubnet	subnet-0b332c288246074f2	10.0.16.0/20	–	Main (rtb-0f324235ed0bc038c)

Selected subnets

subnet-03bd3e93754c6625e / publicsubnet ✕

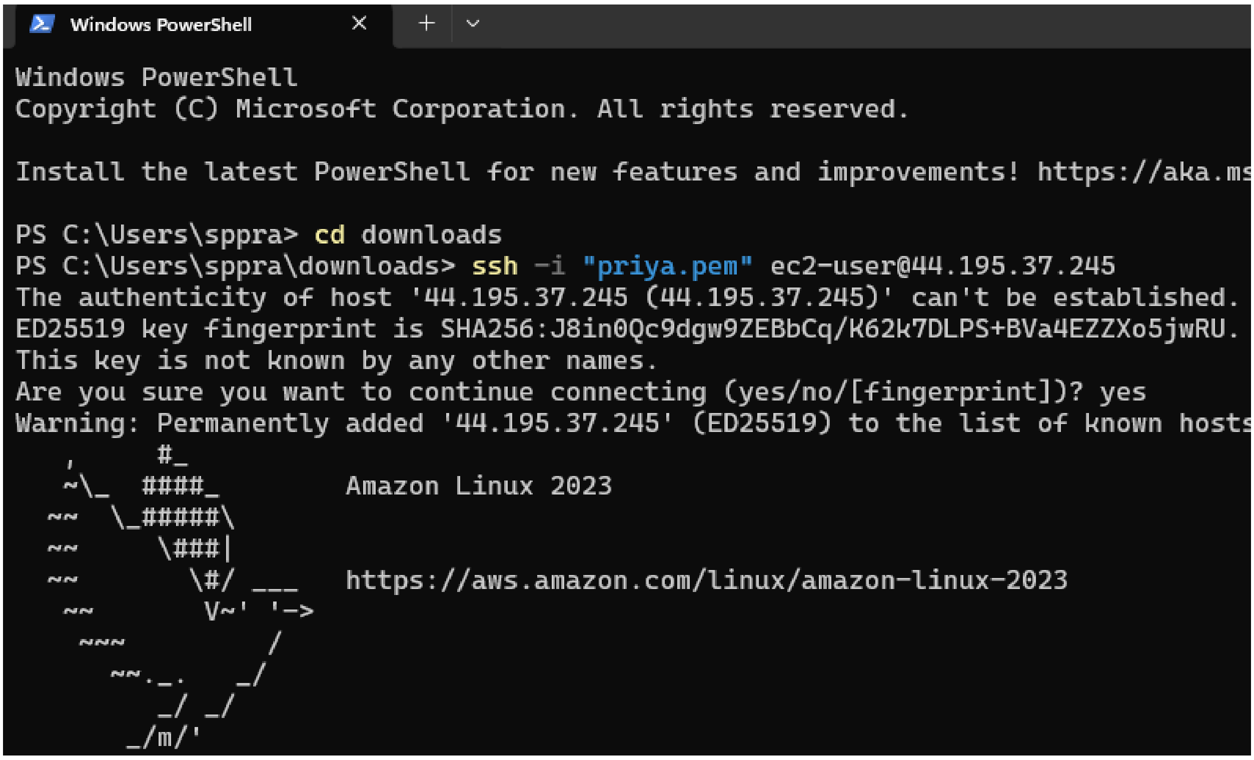
Step 10:

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 *MyVPC* 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.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
BastionHost	i-0aca0f1aa64d54c6a	Running	t2.micro	Initializing	View alarms +	us-east-1a

Step 11: Paste the command copied in the SSH client and connect it by using your key pair.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms

PS C:\Users\sppra> cd downloads
PS C:\Users\sppra\downloads> ssh -i "priya.pem" ec2-user@44.195.37.245
The authenticity of host '44.195.37.245 (44.195.37.245)' can't be established.
ED25519 key fingerprint is SHA256:J8in0Qc9dgw9ZEBbCq/K62k7DLPS+BVa4EZZXo5jwRU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.195.37.245' (ED25519) to the list of known hosts
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```

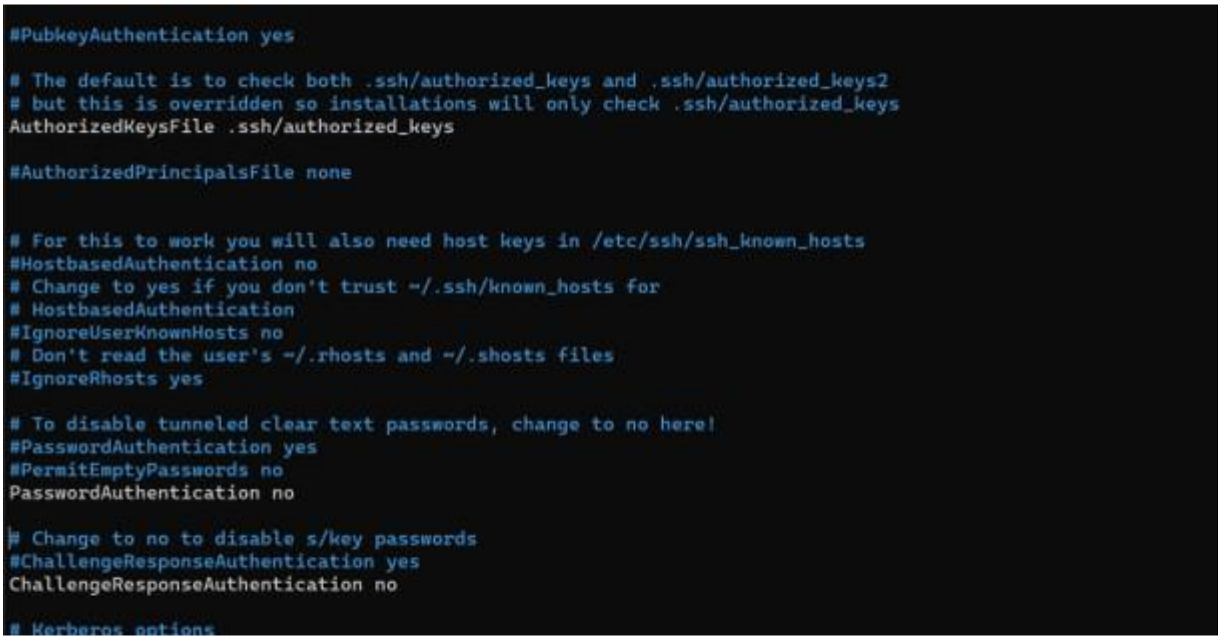
Step 12:

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.