

**PROJECT TITLE:DESIGN AND DEPLOYMENT OF CUSTOM VPC
WITH SUBNETS AND EC2 INSTANCE ON AWS**

NAME: ANNIE JOHN

USER ID:27739

Batch: 25VID0885_DC_Batch4

TABLE OF CONTENTS

Sl.no	Section Title	Page No
1	Introduction	3
2	Lab Objective	3
3	AWS VPC Concepts	3 - 4
4	Step-by-Step Guide	5-11
5	EC2 Instance Deployment	12-16
6	Conclusion	17

➤ INTRODUCTION

This project is based on designing and deploying a custom Virtual Private Cloud (VPC) using Amazon Web Services (AWS). It involves creating public and private subnets, setting up route tables, and attaching an internet gateway. An EC2 instance is launched in the public subnet to test connectivity. The main aim of this project is to understand the basic concepts of AWS networking and practice how to build a secure and organized cloud infrastructure.

➤ LAB OBJECTIVE:

To design a secure and scalable custom Virtual Private Cloud (VPC) in AWS, complete with subnets, route tables, an internet gateway, security groups, and deploy an EC2 instance.

➤ AWS VPC CONCEPTS

Amazon Virtual Private Cloud (VPC) serves as the networking layer for Amazon EC2 and other AWS services. Below are the core concepts and components essential to understanding and designing a VPC environment:

1. Virtual Private Cloud (VPC)

A logically isolated section of the AWS cloud dedicated to your AWS account. You define your own virtual network topology, including IP address ranges, subnets, route tables, and network gateways.

2. Subnet

A segment of your VPC's IP address range where you can place groups of isolated resources. Subnets can be:

- Public: Accessible from the internet.
- Private: Not directly accessible from the internet.

3. CIDR Block

CIDR (Classless Inter-Domain Routing) defines the IP address range of the VPC or subnet.

Example: 10.0.0.0/24 provides 256(2^8) IP addresses.

4. Route Table

A set of rules that determine how network traffic is directed. Each subnet must be associated with a route table to control outbound and inbound routes.

5. Internet Gateway (IGW)

A horizontally scaled, redundant, and highly available VPC component that enables communication between instances in your VPC and the internet.

6. NAT Gateway

A managed network address translation service that allows instances in a private subnet to access the internet while preventing unsolicited inbound connections.

7. Security Groups

Virtual firewalls attached to EC2 instances. They are stateful, meaning return traffic is automatically allowed, and they control traffic at the instance level.

➤ STEP-BY-STEP GUIDE

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>
2. On the **VPC Dashboard**, choose **Create VPC**. Under **VPC Settings** select VPC only and give your VPC name and **IP CIDR block 10.0.0.0/24(IPV4)**. Click on **Create VPC**.

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateVpccreateMode=vpcOnly

aws Search [Alt+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.

my_vpc1

IPv4 CIDR block Info
☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/24
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

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

3. Create and attach an **internet gateway** to a VPC.
 - i. **Navigate to Internet Gateways->select Create Internet Gateway->Name the Internet Gateway (Optional)->Create the Internet Gateway->Click Create Internet Gateway.**

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateInternetGateway

aws Search [Alt+S]

VPC > Internet gateways > Create internet gateway

Internet gateway successfully deleted - igw-087b2f69b3889e36e

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.

ING

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.

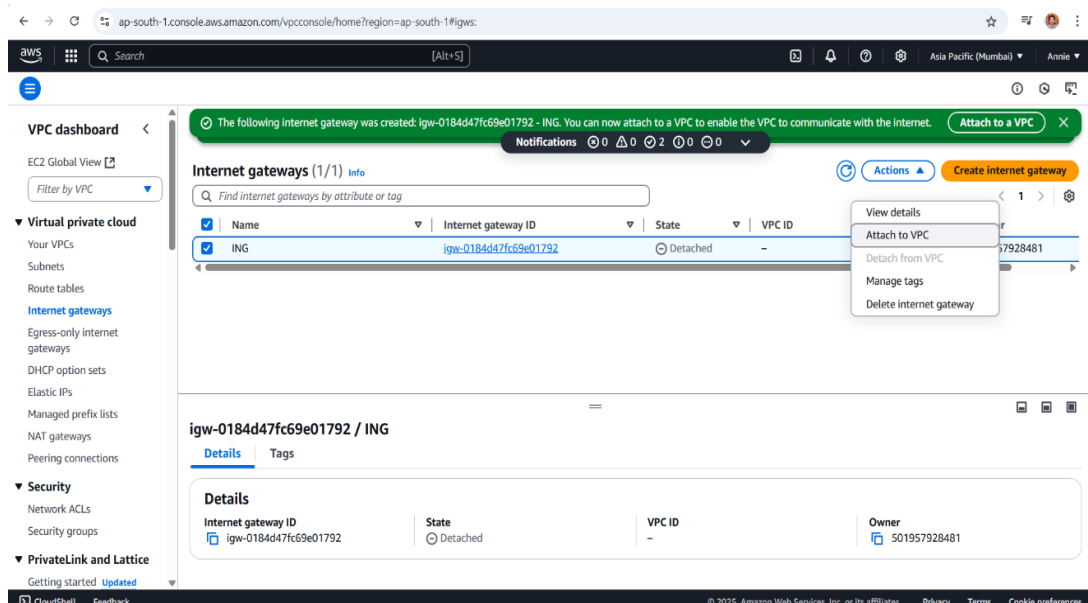
Key	Value - optional	
Q Name	Q ING	Remove

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

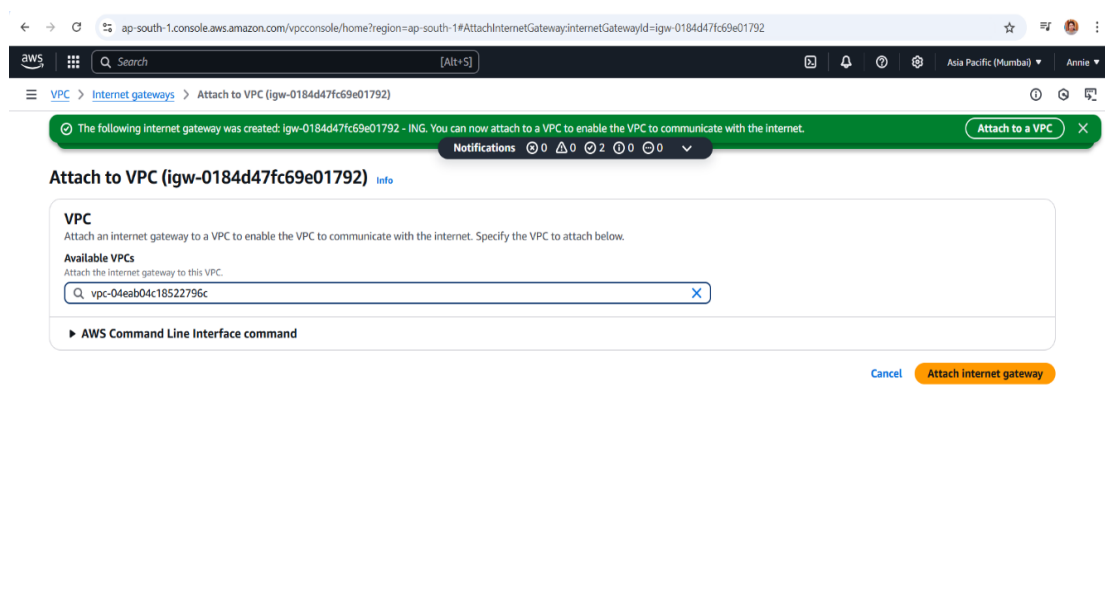
[Cancel](#) [Create internet gateway](#)

- ii. **Attach the Internet Gateway to a VPC**

- Select the internet gateway you just created.
- Click **Actions** → **Attach to VPC**.



- Choose your VPC from the dropdown list.
- Click **Attach Internet Gateway**.



4. Create Subnets

i. Steps to Create a Public Subnet

Go to Subnets ->Click Create Subnet->Enter Subnet Name->Choose VPC
Select the same VPC you created earlier->Choose a Availability Zone)->
Enter CIDR Block (eg:10.0.0.0/25 -dividing 256 ip to 128)->Click Create
Subnet.

The screenshot shows the 'Create Subnet' page in the AWS Management Console. The breadcrumb navigation is 'VPC > Subnets > Create subnet'. The page title is 'Subnet 1 of 1'. Under 'Subnet name', the text says 'Create a tag with a key of 'Name' and a value that you specify.' and the input field contains 'public subnet'. Below this, it says 'The name can be up to 256 characters long.' Under 'Availability Zone', the text says 'Choose the zone in which your subnet will reside, or let Amazon choose one for you.' and the dropdown menu shows 'Asia Pacific (Mumbai) / ap-south-1a'. Under 'IPv4 VPC CIDR block', the text says 'Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.' and the dropdown menu shows '10.0.0.0/24'. Under 'IPv4 subnet CIDR block', the input field contains '10.0.0.0/25' and a label '128 IPs' is shown to the right. Below this, there are navigation arrows. Under 'Tags - optional', there is a table with two columns: 'Key' and 'Value - optional'. The first row has 'Name' in the Key column and 'public subnet' in the Value column. There are buttons for 'Add new tag' and 'Remove'.

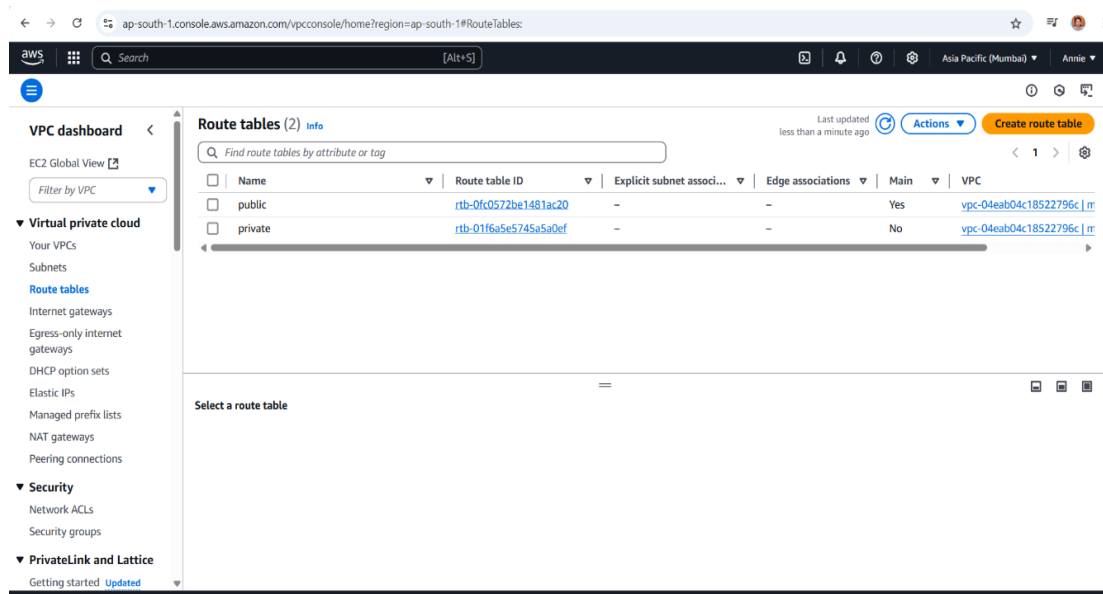
ii. Steps to Create a Private Subnet

Go to Subnets ->Click Create Subnet->Enter Subnet Name->Choose VPC
Select the same VPC you created earlier->Choose a Different Availability
Zone)->Enter CIDR Block (eg:10.0.0.128/25 -dividing 256 ip to 128)->Click
Create Subnet.

The screenshot shows the 'Create Subnet' page in the AWS Management Console. The breadcrumb navigation is 'VPC > Subnets > Create subnet'. The page title is 'Subnet settings'. The text says 'Specify the CIDR blocks and Availability Zone for the subnet.' Under 'Subnet 1 of 1', under 'Subnet name', the text says 'Create a tag with a key of 'Name' and a value that you specify.' and the input field contains 'private subnet'. Below this, it says 'The name can be up to 256 characters long.' Under 'Availability Zone', the text says 'Choose the zone in which your subnet will reside, or let Amazon choose one for you.' and the dropdown menu shows 'Asia Pacific (Mumbai) / ap-south-1b'. Under 'IPv4 VPC CIDR block', the text says 'Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.' and the dropdown menu shows '10.0.0.0/24'. Under 'IPv4 subnet CIDR block', the input field contains '10.0.0.128/25' and a label '128 IPs' is shown to the right. Below this, there are navigation arrows. Under 'Tags - optional', there is a table with two columns: 'Key' and 'Value - optional'. The first row has 'Name' in the Key column and 'private subnet' in the Value column. There are buttons for 'Add new tag' and 'Remove'.

5. Creating route table

- i. **Main route table**—The route table that automatically comes with your VPC. It controls the routing for all subnets that are not explicitly associated with any other route table. Named the **default route table** as public.



- ii. **Create a custom route table** - Go to Route Tables → Click Create Route Table → Enter a Name (e.g., Private Route Table) → Choose the custom VPC you created → Click Create.

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateRouteTable

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.

private

VPC
The VPC to use for this route table.

vpc-04eeb04c18522796c (my_vpc1)

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**

Q Name X Q private X Remove

Add new tag

You can add 49 more tags.

Cancel Create route table

6. Configure Route Tables.

- Select the Route Table you just created (choose public) → Go to Routes tab → Click Edit Routes → Click Add Route → Enter Destination: 0.0.0.0/0 → Target: Select Internet Gateway → Click Save Changes. (Helps in allow instances in the **public subnet to access the internet**).

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditRoutes?routeTableId=rtb-0fc0572be1481ac20

VPC > Route tables > rtb-0fc0572be1481ac20 > Edit routes

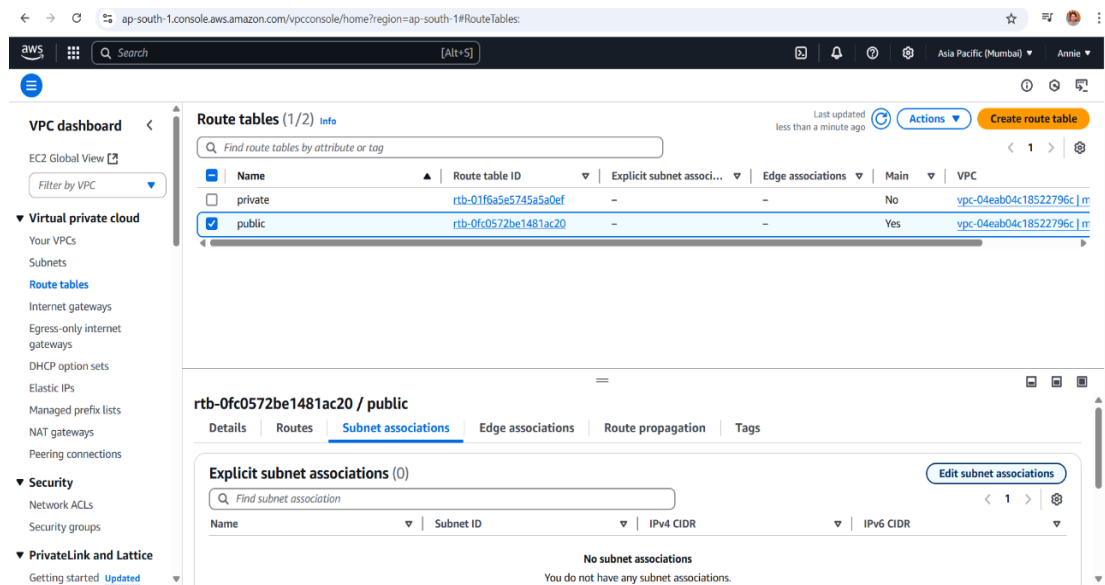
Edit routes

Destination	Target	Status	Propagated
10.0.0.0/24	local	Active	No
Q 0.0.0.0/0 X	Q local X Internet Gateway Q igw-0184d47fc69e01792 X	-	No

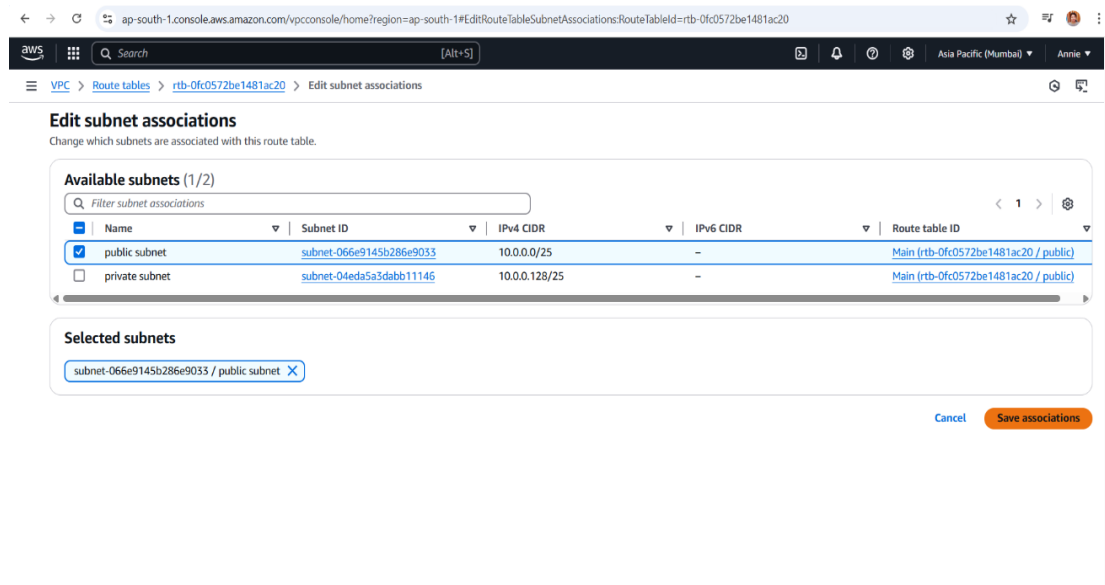
Add route

Cancel Preview Save changes

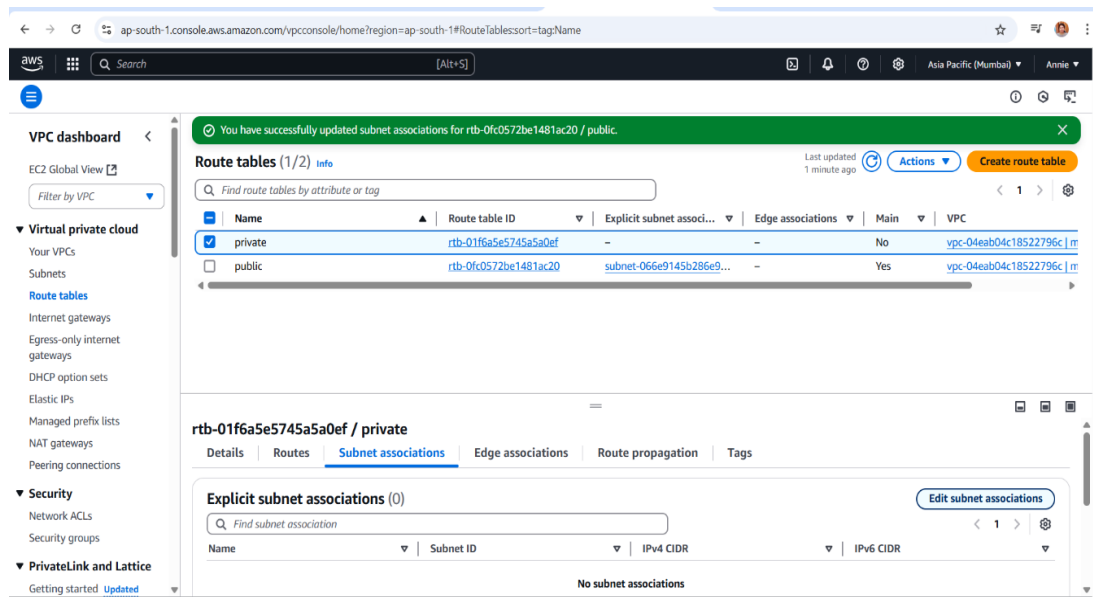
- Go to Subnet Associations tab → Click Edit Subnet Associations. (Subnet Associations - Each subnet must be associated with a route table. This determines how traffic flows in and out of that subnet.)



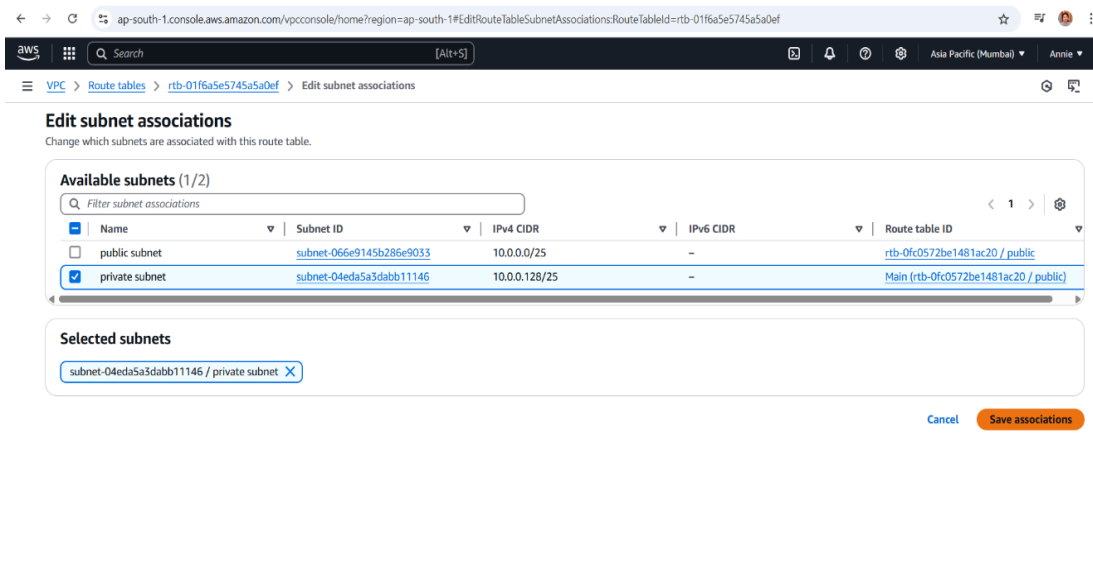
- Select the Public Subnet → Click Save Associations. (This allows **public subnet** to access the internet.)



- b) Since the **private route table** is meant to isolate the **private subnet** from direct internet access, we only perform **subnet association** for the private subnet.
We **do not edit the routes** in the private route table at this stage, because we are not enabling any external internet access (such as through an Internet Gateway or NAT Gateway).



- Select the private subnet->save changes.



7. Custom VPC Setup Completed

At this stage, the custom VPC has been successfully configured with:

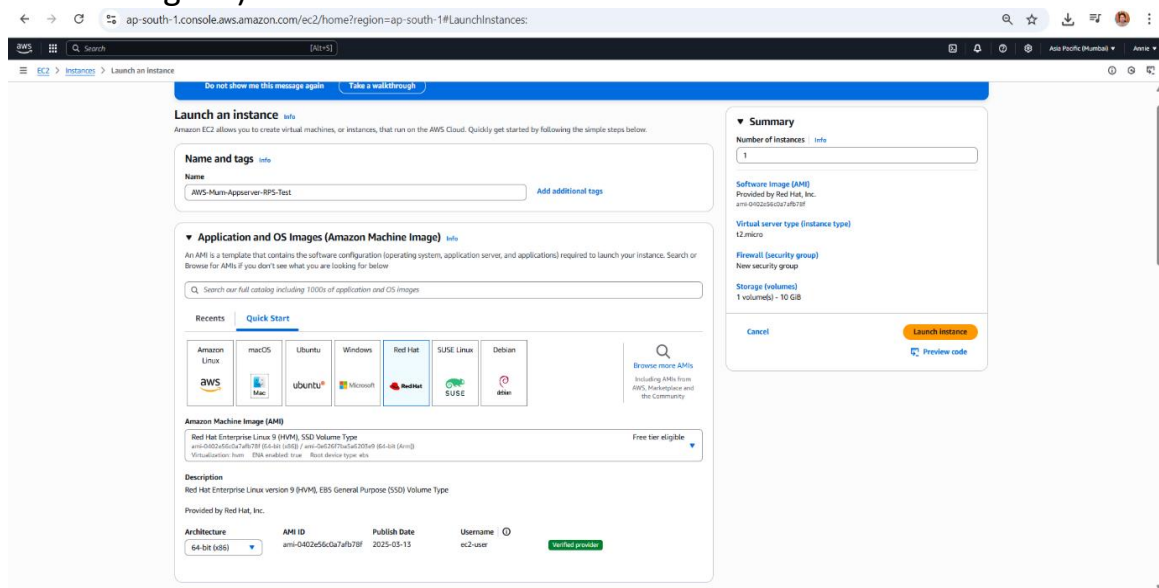
- One **public subnet** (with internet access via Internet Gateway)
- One **private subnet** (isolated from direct internet access)
- Properly configured **route tables** and **subnet associations**.

➤ EC2 INSTANCE DEPLOYMENT

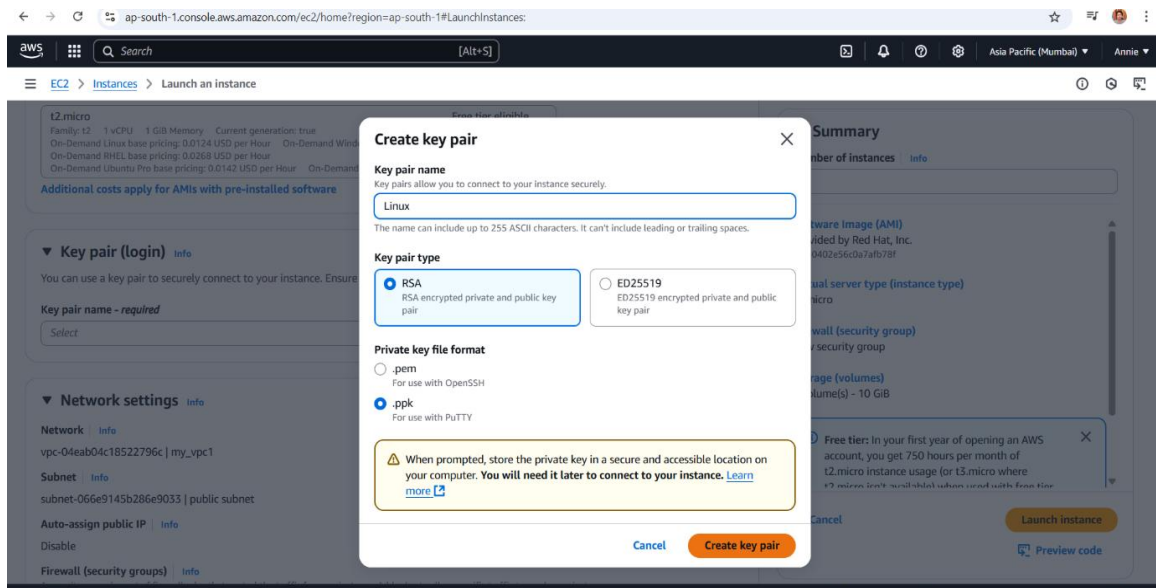
8. EC2 Instance Deployment in the Custom VPC

After successfully designing and deploying the custom VPC with appropriate subnets, route tables, and internet connectivity, the next step is to **deploy an EC2 instance** within the VPC. This instance will help verify that the network is functioning correctly and can be used for hosting applications or services as required.

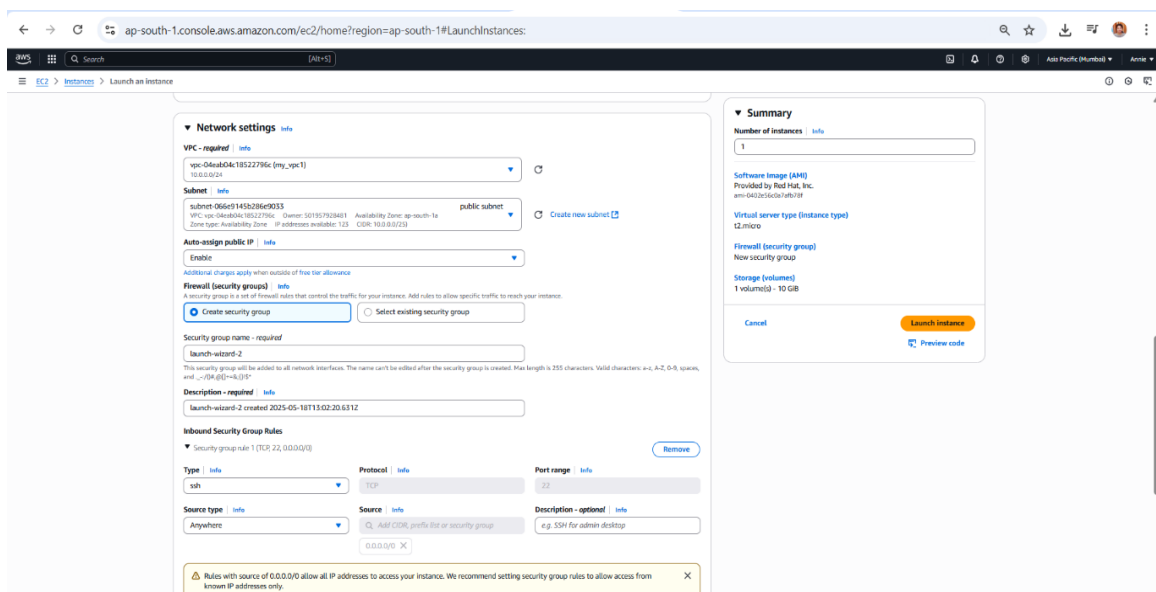
- **Step 1:** Open the Amazon EC2 console by navigating to <https://console.aws.amazon.com/ec2/>
- **Step 2:** Click on **Launch instance** for creating an instance->give name -> choose AMI (select Red Hat)-> instance type (t2. micro, free tier eligible).



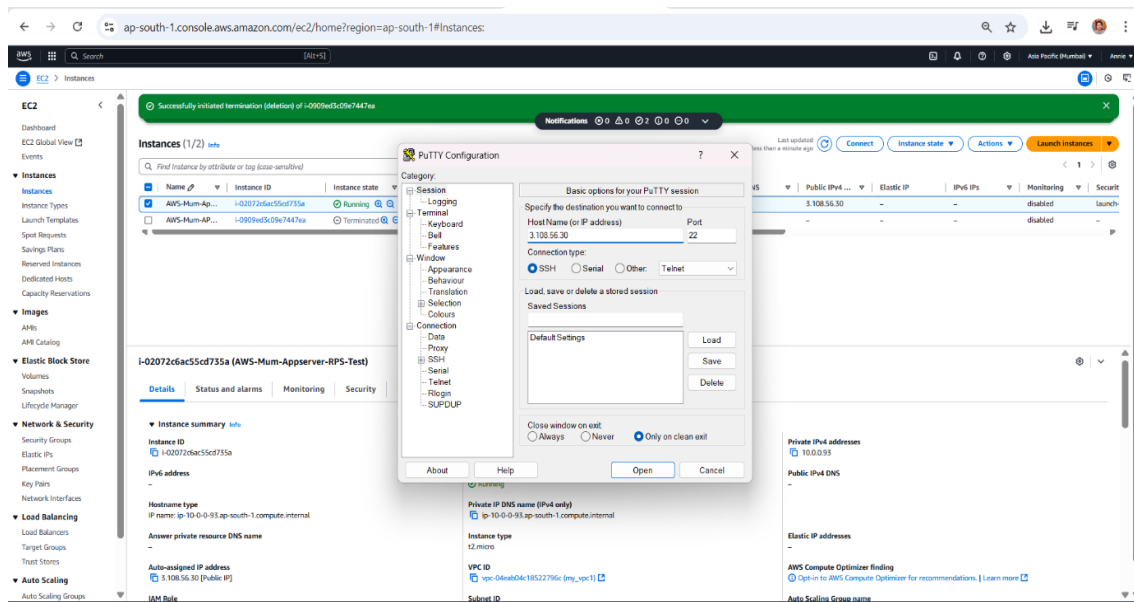
- **Step 3:** Create a **key pair** choose RSA, give name for **key pair** (since Linux is chosen select file format '.ppk').



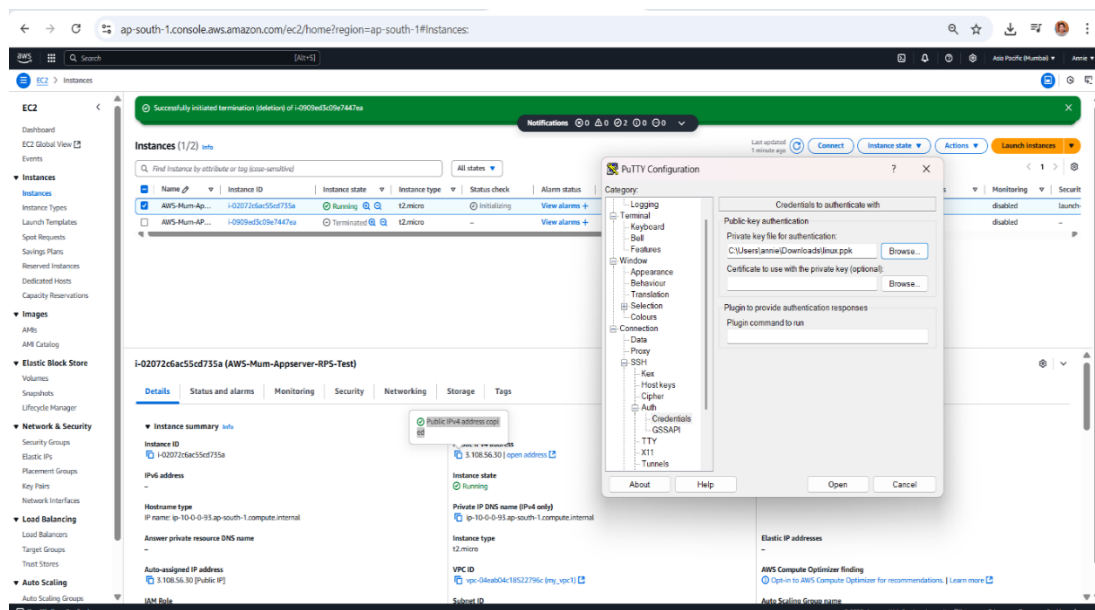
- **Step 4: Edit Network Settings->Enable auto-assign public IP**
->create security group use ssh.



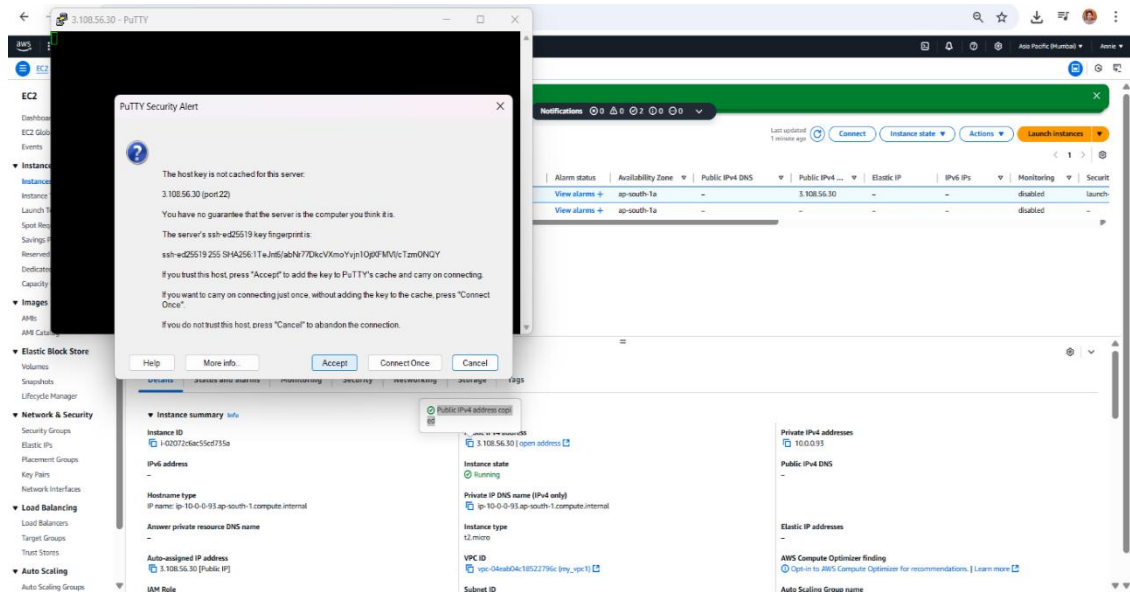
- **Step 5: Created instance->open PuTTY->give public IP.**



- **Step 6:** In SSH->Auth->Credentials->Browse the downloaded key pair.



- **Step 7:** Click Open->accept.



- **Step 8:** Give your username: ec2-user. Completed the step of connecting to your EC2 instance via SSH using your key pair.

```
ec2-user@ip-10-0-0-93:~$
login as: ec2-user
Authenticating with public key "linux"
Register this system with Red Hat Insights: rhc connect

Example:
# rhc connect --activation-key <key> --organization <org>

The rhc client and Red Hat Insights will enable analytics and additional
management capabilities on your system.
View your connected systems at https://console.redhat.com/insights

You can learn more about how to register your system
using rhc at https://red.ht/registration
[ec2-user@ip-10-0-0-93 ~]$
```

- **Step 9:** Verify Internet Connectivity from EC2 Instance. After deploying the EC2 instance in the public subnet and establishing SSH access using PuTTY, internet connectivity was tested to ensure the instance could reach external networks, by ping google.com.

```
ec2-user@ip-10-0-0-85:~$ ssh
login as: ec2-user
Authenticating with public key "linux3"
Register this system with Red Hat Insights: rhc connect

Example:
# rhc connect --activation-key <key> --organization <org>

The rhc client and Red Hat Insights will enable analytics and additional
management capabilities on your system.
View your connected systems at https://console.redhat.com/insights

You can learn more about how to register your system
using rhc at https://red.ht/registration
Last login: Wed May 21 14:59:22 2025 from 116.68.73.199
[ec2-user@ip-10-0-0-85 ~]$ ping google.com
PING google.com (142.250.183.206) 56(84) bytes of data:
64 bytes from bom07a33-in-f14.1e100.net (142.250.183.206): icmp_seq=1 ttl=114 time=1.66 ms
64 bytes from bom07a33-in-f14.1e100.net (142.250.183.206): icmp_seq=2 ttl=114 time=1.79 ms
64 bytes from bom07a33-in-f14.1e100.net (142.250.183.206): icmp_seq=3 ttl=114 time=1.77 ms
64 bytes from bom07a33-in-f14.1e100.net (142.250.183.206): icmp_seq=4 ttl=114 time=1.69 ms
```


➤ CONCLUSION

This project "**Design and Deployment of Custom VPC with Subnets and EC2 Instance on AWS**" successfully demonstrates the fundamental process of creating a secure and scalable cloud network using AWS services.

A custom VPC was designed and configured with both public and private IP addressing. Within this VPC, subnets were created and associated with route tables and an Internet Gateway to allow internet connectivity. An EC2 instance was deployed in the public subnet, and connectivity was verified using PuTTY for SSH access. Internet access was confirmed through successful ping tests to external domains such as google.com, validating the proper configuration of route tables and security groups.

This implementation showcases a hands-on understanding of AWS networking components including:

- VPC
- Subnets
- Route Tables
- Internet Gateway
- EC2 Instances
- Security Groups

In conclusion, this project reinforces essential AWS networking and compute skills that are critical for deploying cloud-based solutions in real-world environments.