Project Documentation

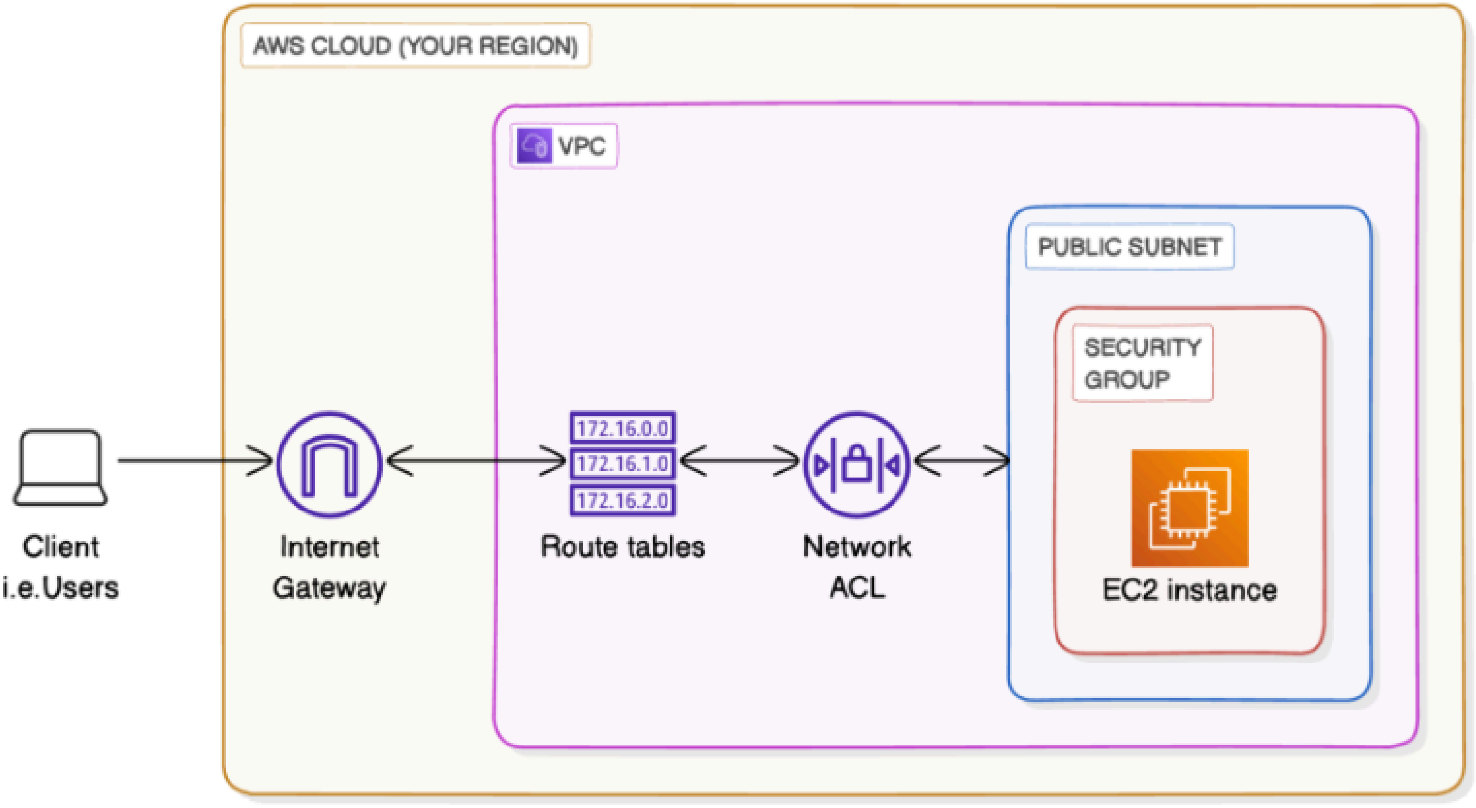
Overview

Project Name	Isolated Network Architecture on AWS with Apache Server
Project Manager	Mridul Gharami
Project Dates	Start Date: Jun 29, 2025 End Date: Jul 2, 2025
Background	With the growing reliance on cloud infrastructure, the ability to design and deploy secure, scalable, and isolated environments has become an essential skill. Amazon Web Services (AWS), being a leading cloud provider, offers core services such as Virtual Private Cloud (VPC) and Elastic Compute Cloud (EC2) to support such deployments. This project was undertaken to understand and implement the foundational components of a secure cloud network. By creating a custom VPC, configuring subnets, Internet Gateway, routing, and setting up EC2 with a web server, the project simulates real-world scenarios where infrastructure security and controlled access are critical. Through this hands-on exercise, the goal was to gain deeper insight into how AWS networking components interact, and how to apply best practices in securing public-facing services in the cloud.
Objectives	• **Design a custom Virtual Private Cloud (VPC)** to create an isolated and controlled cloud network. • **Configure a public subnet** to host externally accessible resources like web servers. • **Attach and route traffic through an Internet Gateway (IGW)** to enable internet access. • **Create and associate a route table** to direct traffic correctly within the VPC. • **Implement a security group** to allow only specific traffic (SSH, HTTP, HTTPS) while maintaining security.

- **Deploy a Free Tier EC2 instance** inside the VPC using a secure key pair.
- **Connect to the instance via SSH** using the .pem file to enable remote configuration.
- **Install and configure Apache web server** on the EC2 instance to serve web content.
- **Access the web server via browser** using the public IP of the EC2 instance.
- **Validate the entire infrastructure setup** and document each

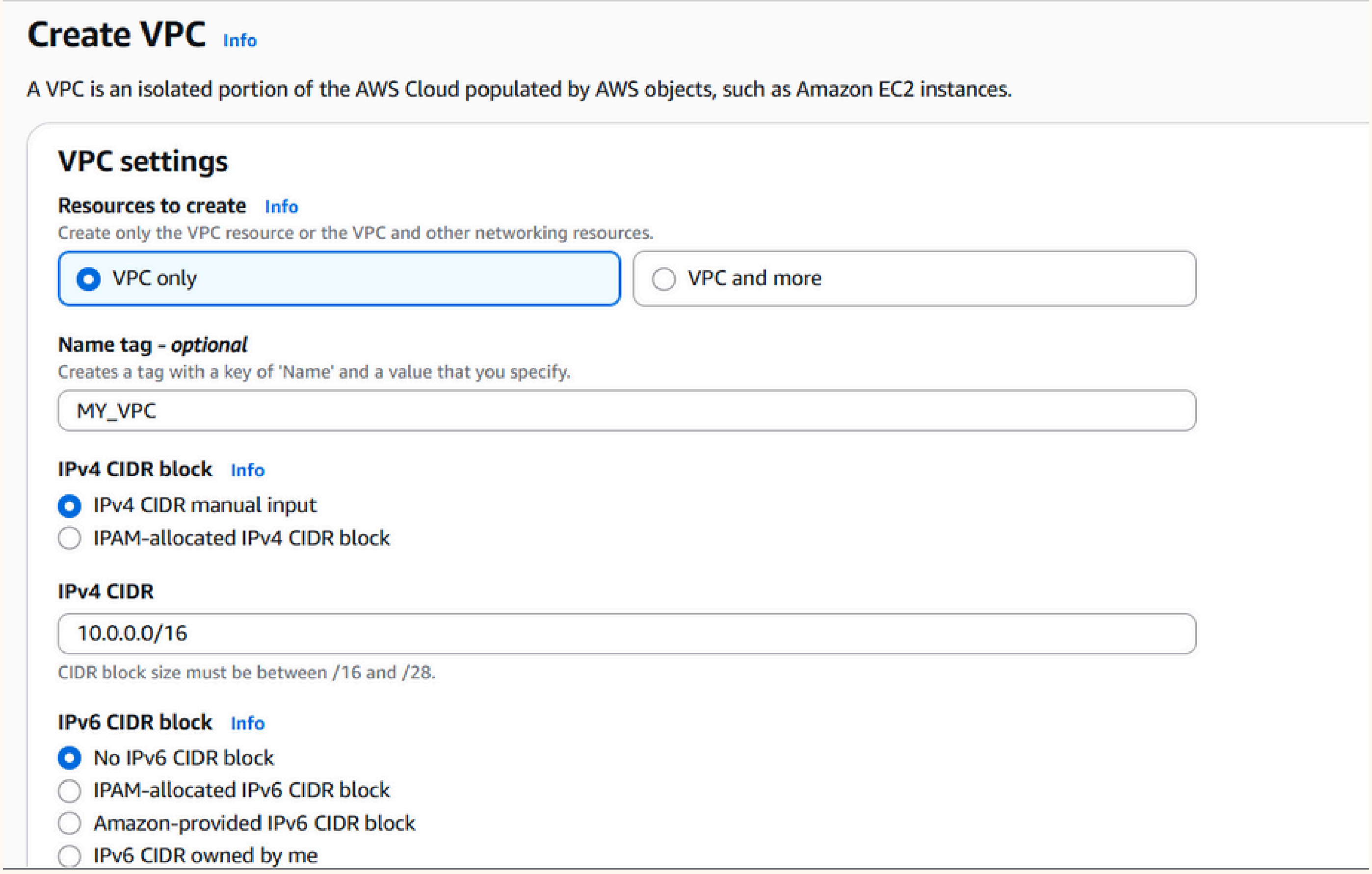
Project Architecture

The architecture consists of a custom VPC containing a public subnet, with an Internet Gateway attached for external access. A route table directs outbound traffic, while a security group protects the EC2 instance. The instance is launched with a public IP and configured with a basic web server (Apache) accessible via HTTP and SSH.



Create a Custom VPC

Description:	A Virtual Private Cloud (VPC) provides an isolated network environment in AWS. Creating a custom VPC allows full control over IP ranges, subnets, gateways, and access control. This forms the foundational layer for all networking in this project.
Configuration Details:	**VPC Name**: MY_VPC **IPv4 CIDR Block**: 172.16.0.0/16 **Tenancy**: Default **Enable DNS Hostnames**: Yes *(optional but recommended for EC2 access)*



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_VPC

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

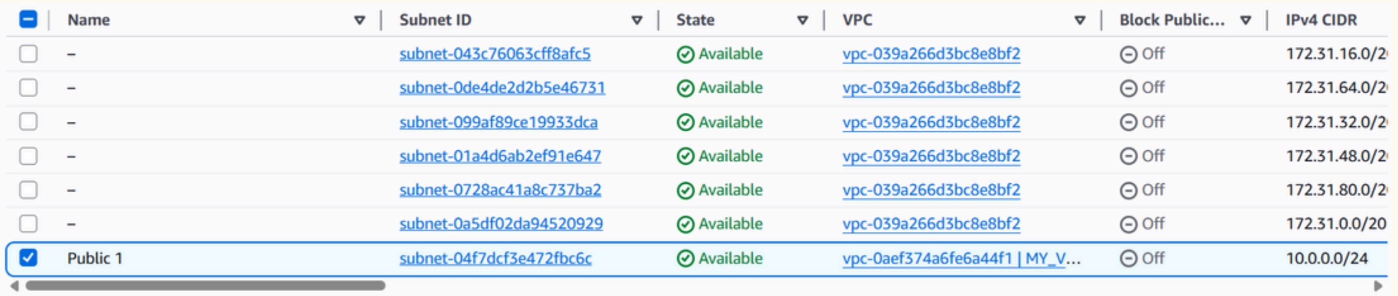
○ IPAM-allocated IPv6 CIDR block

○ Amazon-provided IPv6 CIDR block

○ IPv6 CIDR owned by me

Create a Public Subnet

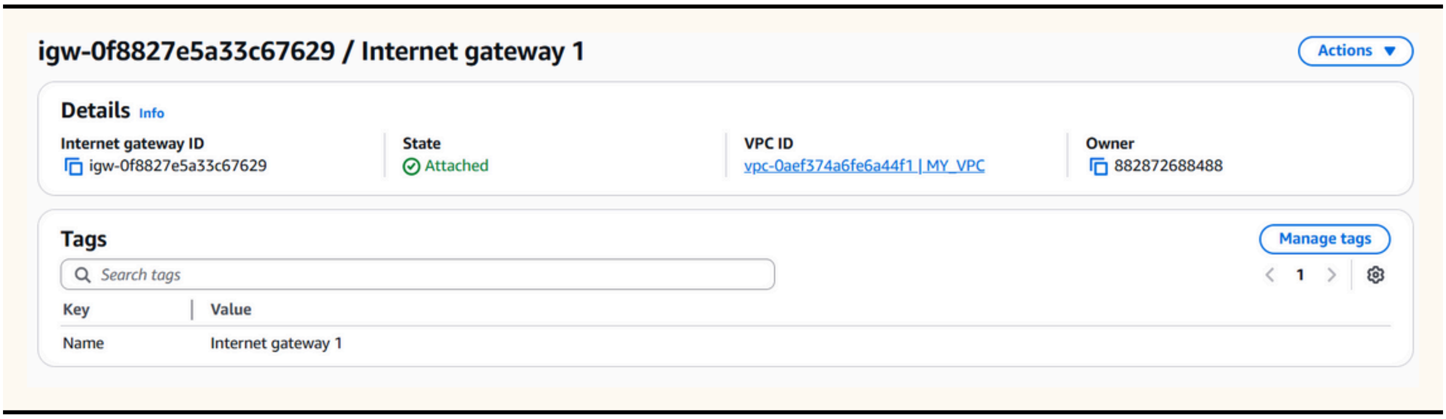
Description:	A **subnet** is a range of IP addresses within your VPC. Public subnets are used to host resources (like EC2 instances) that need direct access to the internet. In this step, we define a subnet within our custom VPC and assign a CIDR block to it.
Configuration Details:	**Subnet Name**: Public 1 **VPC**: MY_VPC **Availability Zone**: Choose one (e.g., ap-south-1a) **IPv4 CIDR Block**: 172.16.1.0/24 **Auto-assign public IPv4 address**: Enabled



☑	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
□	–	subnet-043c76063cff8afc5	✓ Available	vpc-039a266d3bc8e8bf2	⊖ Off	172.31.16.0/2
□	–	subnet-0de4de2d2b5e46731	✓ Available	vpc-039a266d3bc8e8bf2	⊖ Off	172.31.64.0/2
□	–	subnet-099af89ce19933dca	✓ Available	vpc-039a266d3bc8e8bf2	⊖ Off	172.31.32.0/2
□	–	subnet-01a4d6ab2ef91e647	✓ Available	vpc-039a266d3bc8e8bf2	⊖ Off	172.31.48.0/2
□	–	subnet-0728ac41a8c737ba2	✓ Available	vpc-039a266d3bc8e8bf2	⊖ Off	172.31.80.0/2
□	–	subnet-0a5df02da94520929	✓ Available	vpc-039a266d3bc8e8bf2	⊖ Off	172.31.0.0/20
☑	Public 1	subnet-04f7dcf3e472fbc6c	✓ Available	vpc-0aef374a6fe6a44f1 | MY_V...	⊖ Off	10.0.0.0/24

Attach an Internet Gateway

Description:	An **Internet Gateway (IGW)** is a horizontally scaled, redundant component that allows instances in your VPC to connect to the internet. Attaching an IGW to your custom VPC is required for any public-facing EC2 instances to send or receive traffic from outside AWS.
Configuration Details:	**Internet Gateway Name**: Internet gateway 1 **Attached to VPC**: MY_VPC



igw-0f8827e5a33c67629 / Internet gateway 1

Details

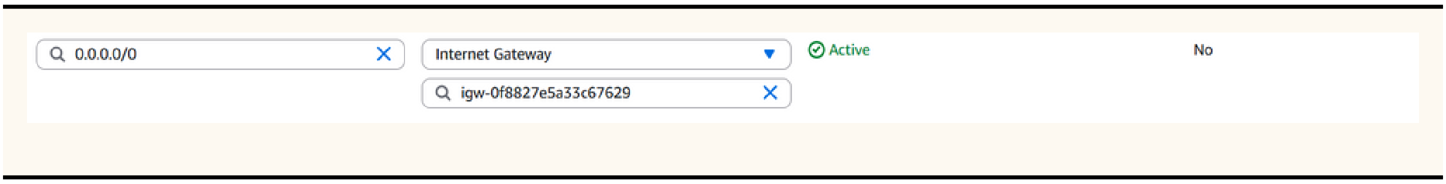
Internet gateway ID	State	VPC ID	Owner
igw-0f8827e5a33c67629	Attached	vpc-0aef374a6fe6a44f1 | MY_VPC	882872688488

Tags

Key	Value
Name	Internet gateway 1

Configure a Route Table

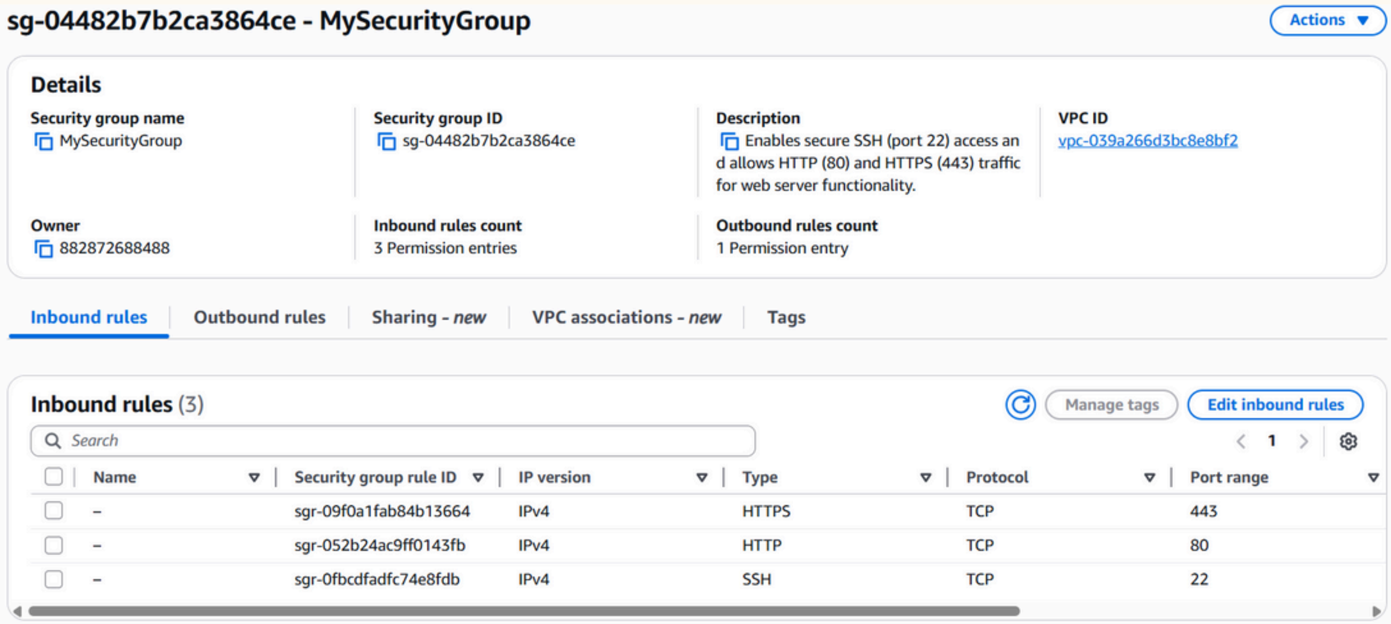
Description:	A **Route Table** in AWS defines how traffic is directed within your VPC. To allow internet access from your public subnet, you need to add a route that sends outbound traffic to the Internet Gateway.
Configuration Details:	**Route Table Name**: Public 1 **Associated VPC**: MY_VPC **Destination**: 0.0.0.0/0 **Target**: Internet gateway 1



0.0.0.0/0 | Internet Gateway | igw-0f8827e5a33c67629 | Active | No

Create a Security Group

Description:	A **Security Group** acts as a virtual firewall that controls inbound and outbound traffic for your EC2 instance. In this project, we create a security group that allows SSH and web traffic from specific sources while blocking all other traffic by default.
Configuration Details:	**Security Group Name**: MySecurityGroup **Description**: Allows SSH, HTTP, and HTTPS access **VPC**: MY_VPC



sg-04482b7b2ca3864ce - MySecurityGroup

Details

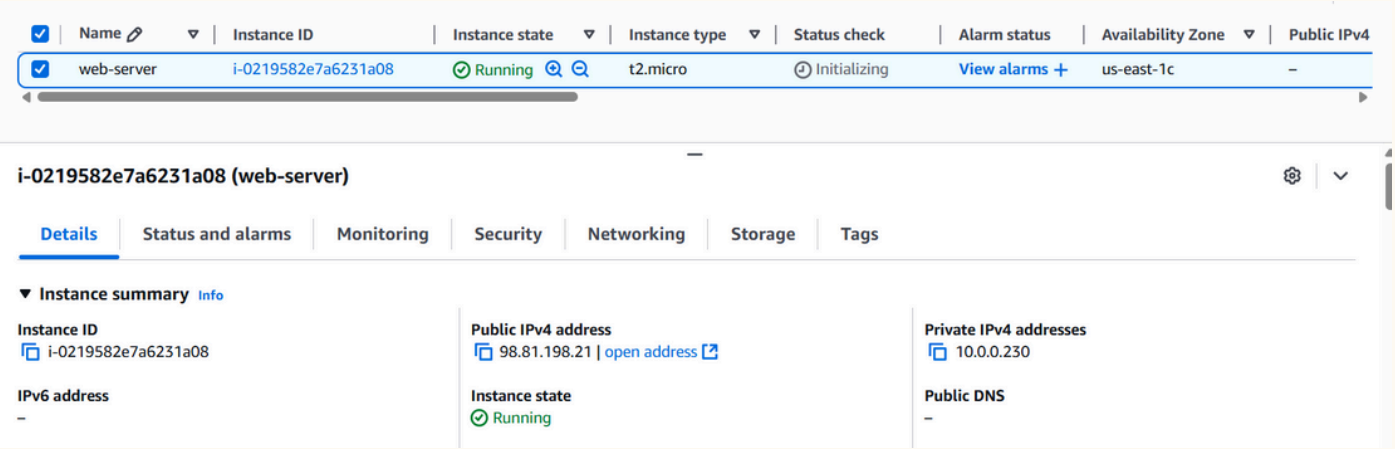
Security group name	Security group ID	Description	VPC ID
MySecurityGroup	sg-04482b7b2ca3864ce	Enables secure SSH (port 22) access an d allows HTTP (80) and HTTPS (443) traffic for web server functionality.	vpc-039a266d3bc8e8bf2
Owner	**Inbound rules count**	**Outbound rules count**	
882872688488	3 Permission entries	1 Permission entry	

Inbound rules (3)

□	Name	Security group rule ID	IP version	Type	Protocol	Port range
□	–	sgr-09f0a1fab84b13664	IPv4	HTTPS	TCP	443
□	–	sgr-052b24ac9ff0143fb	IPv4	HTTP	TCP	80
□	–	sgr-0fbcdfadfc74e8fdb	IPv4	SSH	TCP	22

Launch an EC2 Instance

Description:	Amazon EC2 (Elastic Compute Cloud) provides scalable virtual servers in the cloud. In this step, we launch a lightweight, free-tier eligible EC2 instance inside the custom VPC and public subnet. This instance will later host a web server and be accessible from the internet
Configuration Details:	**Instance Name**: web-server **AMI (OS)**: Ubuntu Server 20.04 LTS (Free tier eligible) **Instance Type**: t2.micro **Key Pair**: Existing or new .pem key (used for SSH) **Network**: MY_VPC **Subnet**: Public 1 **Auto-assign Public IP: Enable** **Security Group**: MySecurityGroup



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	web-server	i-0219582e7a6231a08	Running	t2.micro	Initializing	View alarms +	us-east-1c	–

i-0219582e7a6231a08 (web-server)

▼ Instance summary Info

Instance ID: i-0219582e7a6231a08

Public IPv4 address: 98.81.198.21 | open address

Private IPv4 addresses: 10.0.0.230

IPv6 address: –

Instance state: Running

Public DNS: –

Connect to EC2 via SSH

Description:

After launching the EC2 instance, you need to connect to it via **SSH (Secure Shell)** using your .pem key file. This allows you to access the server terminal and install software such as Apache.

Configuration Details:

You must have the .pem key file downloaded when you created the key pair.

Your instance should have a public IPv4 address assigned.

Port 22 (SSH) must be open in your security group.

```
chmod 400 your-key.pem

ssh -i your-key.pem ubuntu@10.0.0.0
```

Note:

SSH login screenshots are not included to protect sensitive details such as public IP and user information.

Install and Configure Apache Web Server

Description:

With SSH access, install **Apache** to serve web content over HTTP. Apache is a widely used, open-source web server.

Configuration Details:

```
sudo apt update

sudo apt install apache2 -y

sudo systemctl start apache2

sudo systemctl enable apache2
```

Note:

Browser screenshots are not included for privacy. You can verify functionality by visiting your EC2's public IP in a browser.

Validate Access and Final Summary

Description:	This step ensures that your EC2 instance is correctly configured, reachable over the internet, and hosting a working web server. You'll also verify that your network setup and security rules are functioning as expected.
Validation Checklist:	**EC2 instance is in running state** **Security Group allows HTTP (port 80) and SSH (port 22)** **Apache is installed and running** **Visiting the EC2 Public IP in a browser loads the Apache default page** **You can SSH into the instance using your .pem key**

Note:

Security Reminder:

After validation:

- *Consider stopping or terminating the EC2 instance to avoid exceeding Free Tier limits*
- *Remove any unnecessary open ports*
- *Keep your key file secure and never share it*

Final Note:

This completes the deployment of a secure, browser-accessible web server on AWS using a custom VPC, public subnet, and properly configured networking components.