Deploy a Static Website on Amazon EC2 with Apache

JULEANNY NAVAS
AWS Cloud Computing

Introduction

In this project, I provisioned and configured an **Amazon EC2** instance within a custom **VPC** to serve a **static website** using the **Apache web server**. The deployment follows AWS best practices for **network design**, **security groups** configuration, and **automated setup** using EC2 user data. The goal was to host a personal static webpage in a secure, scalable environment without relying on CloudFormation templates.

The instance is launched with Apache pre-installed and configured through **user data**, and the HTML file is served from **/var/www/html**.

Project structure

Section 1: VPC + Networking Setup

Section 2: Create a Security Group

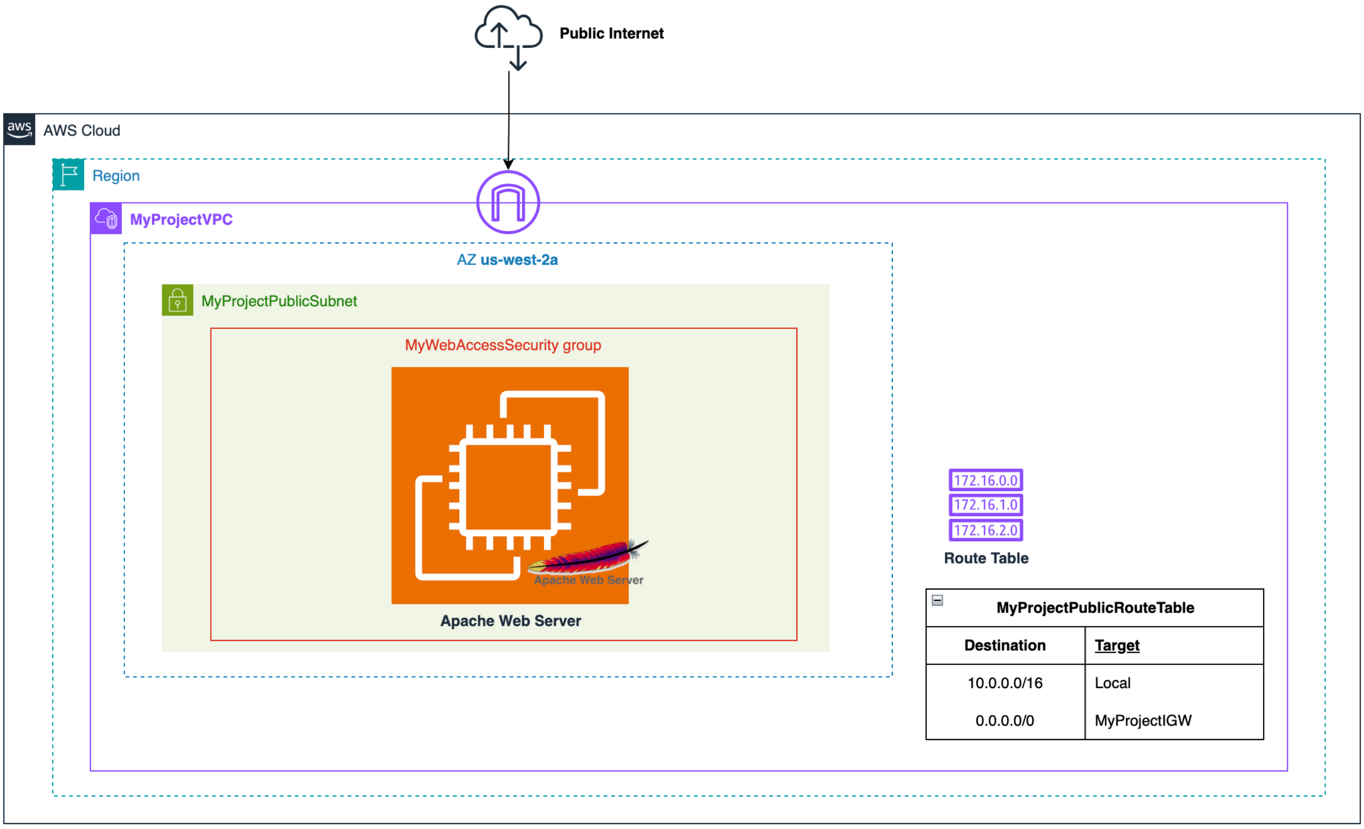
Section 3: EC2 Launch Configuration - Apache Setup via User Data

Section 4: Testing & Validation

Tools & Services used

- **Amazon EC2** – Virtual server to host and serve the web application
- **Amazon VPC** – Custom virtual network for secure resource isolation
- **Apache Web Server** – HTTP server used to host the static web page
- **AWS Security Groups** – Firewall rules to control traffic to the instance
- **EC2 instance access via SSH** – Secure command-line access to the virtual server for configuration and management
- **Bash & HTML** – Scripting and markup used for automation and the web page
- **Architecture Diagrams:** Visual representation of infrastructure components (**app.diagram.net)**

Architecture Overview



MyProjectPublicRouteTable

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	MyProjectIGW

Step-by-Step Implementation

Section 1: VPC + Networking setup

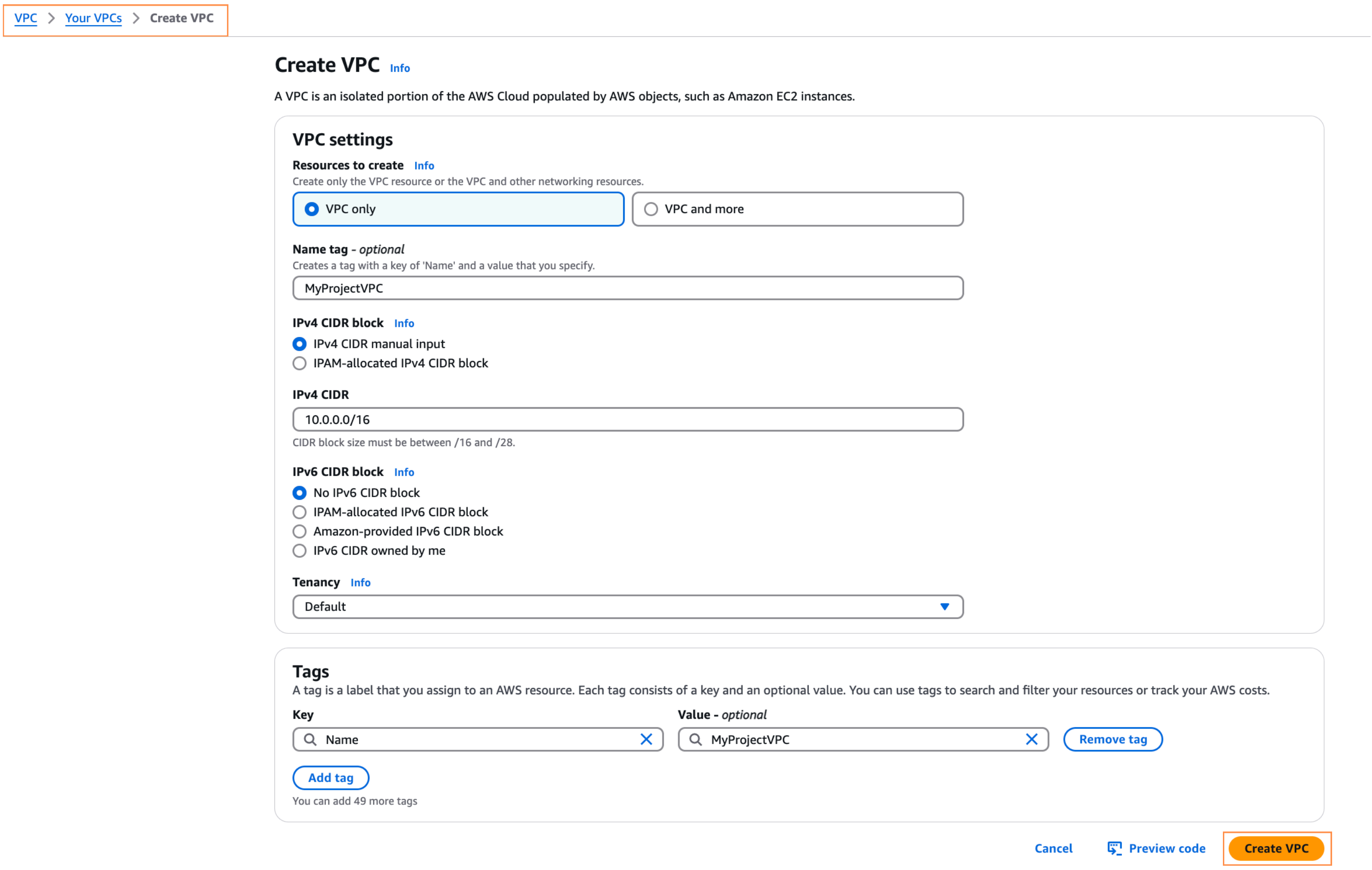
1 . Network configuration

- Created a **new VPC** with a **public subnet,** using a custom VPC is good AWS practice, it gives full control over security, networking, and scalability.
- Set up an **Internet Gateway** and **Route Table** for outbound access.
- Ensured auto-assign public IP is **enabled** for the subnet.

Steps:

- Open the **AWS Management Console.**
- Go to **VPC > Create VPC:**
  - **> Name**: MyProjectVPC
  - **> IPv4 CIDR:** 10.0.0.0/16: /16 provides 65.536 private IP addresses, therefore allows future scalability, creating multiple subnets within this VPC without overlapping ranges.
  - **> Tenancy:** Default is selected to avoid the extra cost associated with dedicated instances, it shares the underlying hardware with other AWS accounts but still maintains

isolation at the hypervisor level. It's recommended unless you have compliance or licensing needs requiring dedicated hardware.



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.
MyProjectVPC

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
Default

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.

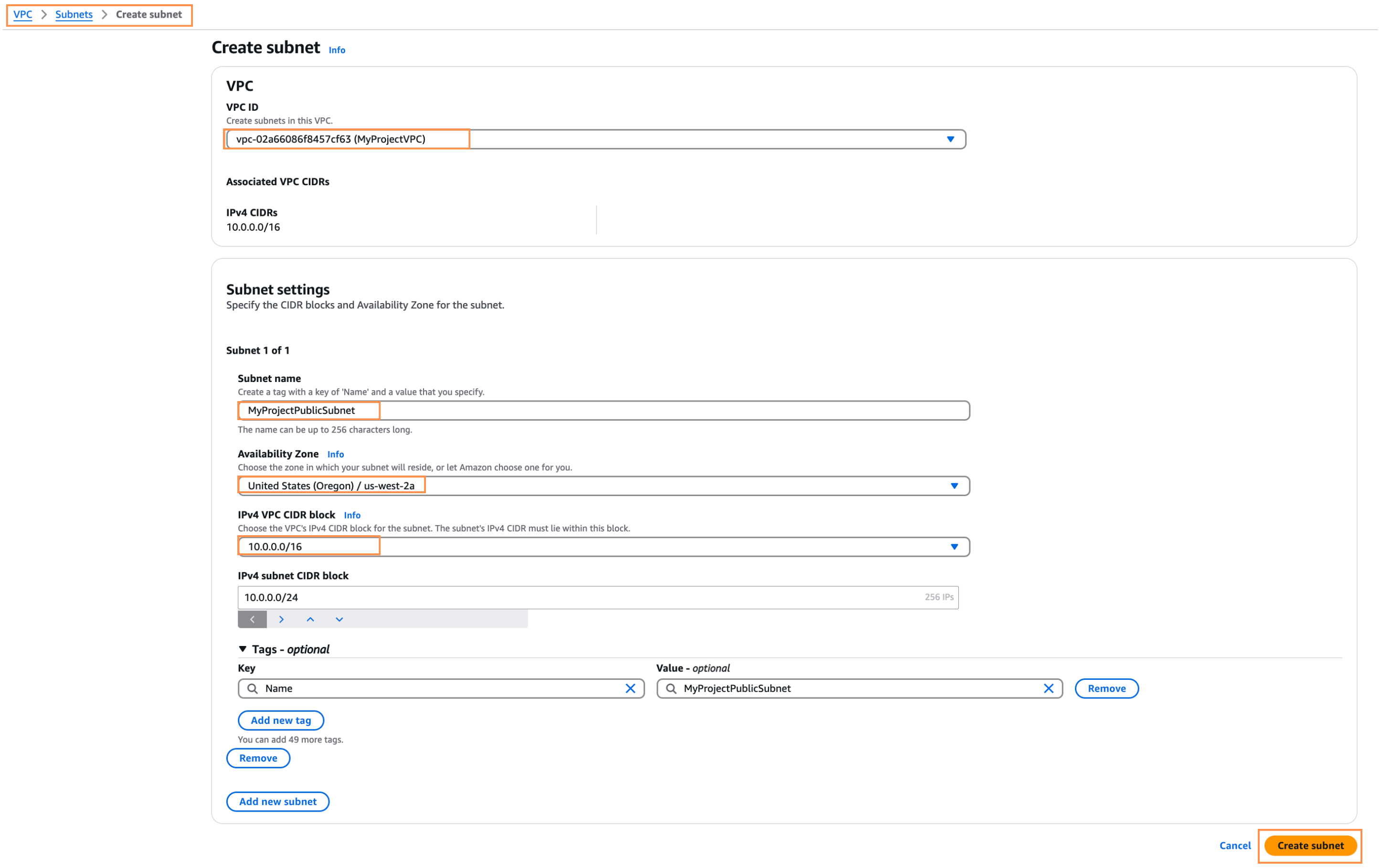
Key	Value - *optional*
Name	MyProjectVPC

Remove tag

Add tag
You can add 49 more tags

Cancel Preview code Create VPC

- Go to **Subnets > Create Subnet:**
  - **> Name**: **MyProjectPublicSubnet**
  - **>** Attach to: **MyProjectVPC**
  - **> AZ:** Select an Availability zone
  - **> IPv4 CIDR:** 10.0.0.0/24: /24 allocates 256 IP addresses, it is suitable for a small public-facing workload such as a web server. It helps to keep IP allocation granular and organised.



VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-02a66086f8457cf63 (MyProjectVPC)

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.
MyProjectPublicSubnet
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 (Oregon) / us-west-2a

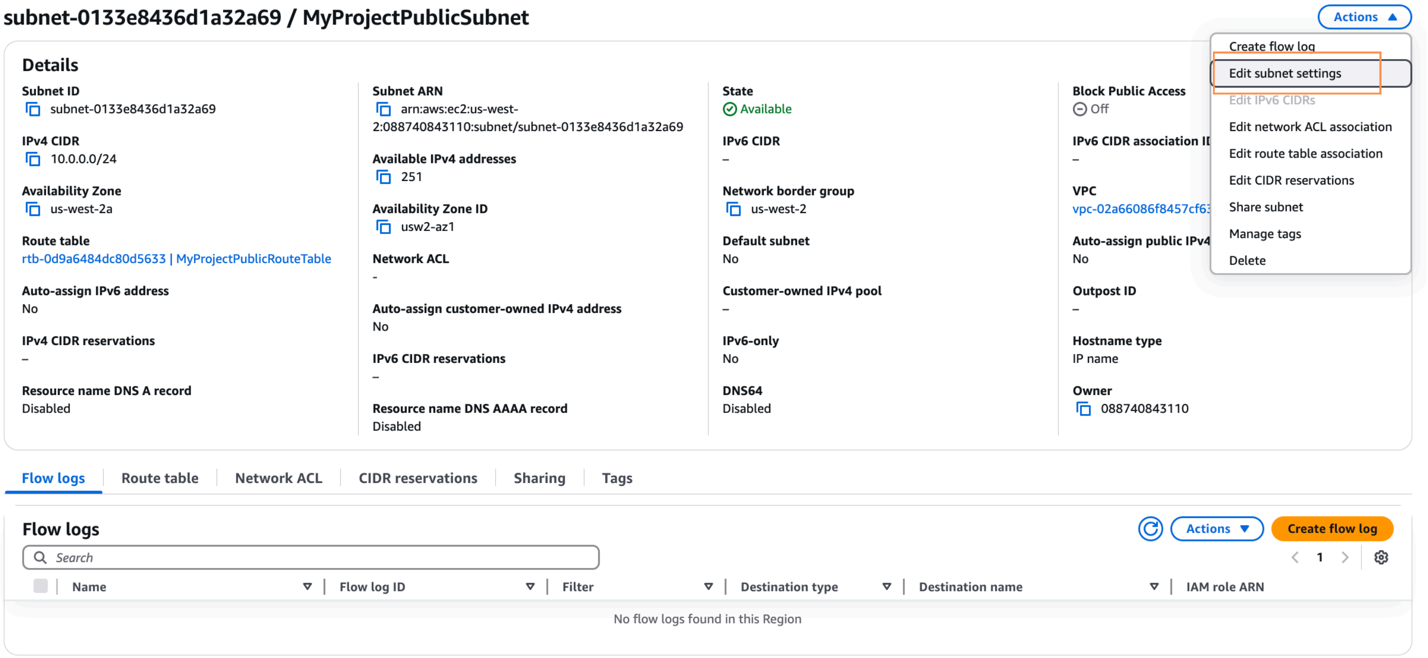
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.0.0/24 256 IPs

Tags - *optional*

Key	Value - *optional*
Name	MyProjectPublicSubnet

- **>** Enable a Public IP for **MyProjectPublicSubnet**, it allows the EC2 instance in this subnet to be accessible over the internet, an essential configuration for serving a static web page.



subnet-0133e8436d1a32a69 / MyProjectPublicSubnet

Details

Subnet ID
subnet-0133e8436d1a32a69

IPv4 CIDR
10.0.0.0/24

Availability Zone
us-west-2a

Route table
rtb-0d9a6484dc80d5633 | MyProjectPublicRouteTable

Auto-assign IPv6 address
No

IPv4 CIDR reservations
–

Resource name DNS A record
Disabled

Subnet ARN
arn:aws:ec2:us-west-2:088740843110:subnet/subnet-0133e8436d1a32a69

Available IPv4 addresses
251

Availability Zone ID
usw2-az1

Network ACL
-

Auto-assign customer-owned IPv4 address
No

IPv6 CIDR reservations
–

Resource name DNS AAAA record
Disabled

State
Available

IPv6 CIDR
–

Network border group
us-west-2

Default subnet
No

Customer-owned IPv4 pool
–

IPv6-only
No

DNS64
Disabled

Block Public Access
Off

IPv6 CIDR association ID
–

VPC
vpc-02a66086f8457cf63

Auto-assign public IPv4
No

Outpost ID
–

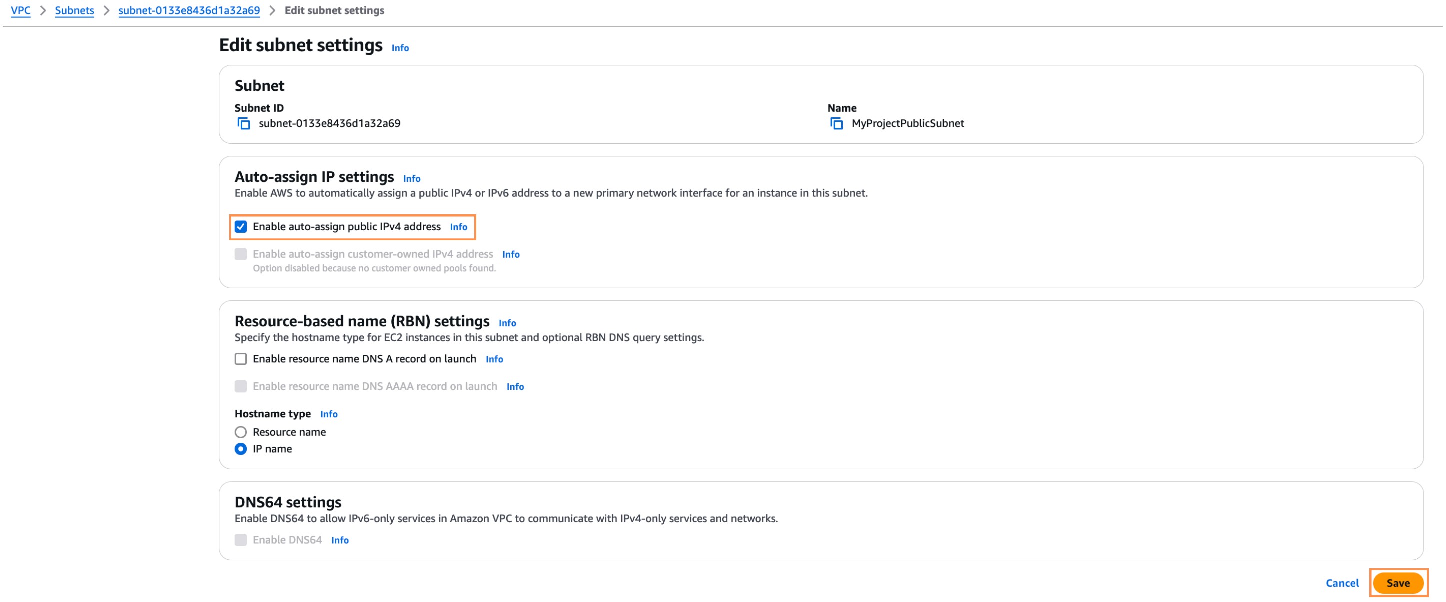
Hostname type
IP name

Owner
088740843110

Flow logs

Name	Flow log ID	Filter	Destination type	Destination name	IAM role ARN

No flow logs found in this Region



VPC > Subnets > subnet-0133e8436d1a32a69 > Edit subnet settings

Edit subnet settings Info

Subnet

Subnet ID: subnet-0133e8436d1a32a69

Name: MyProjectPublicSubnet

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

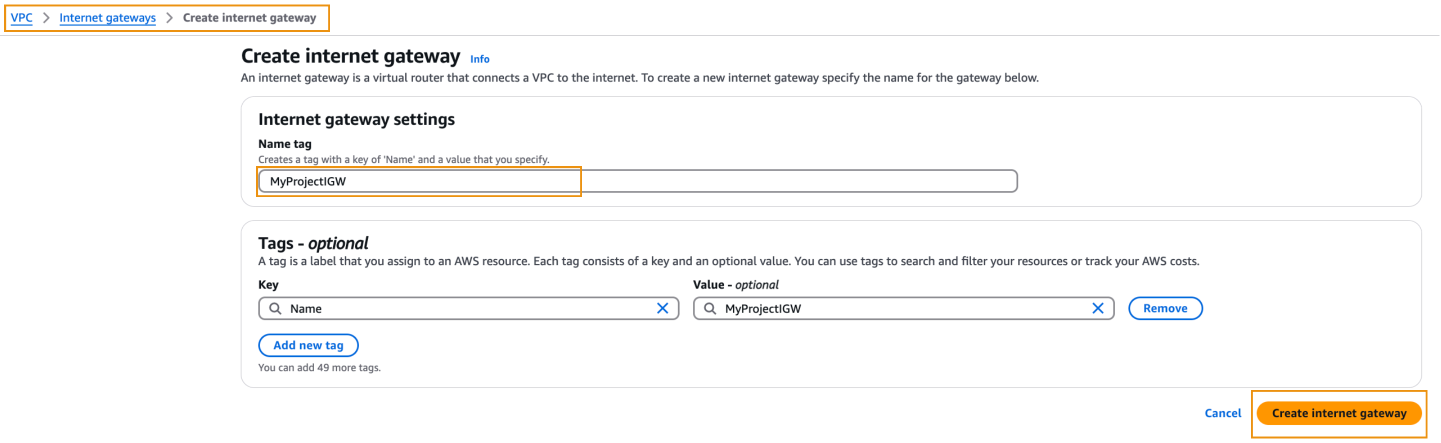
○ Resource name
● IP name

DNS64 settings

Enable DNS64 to allow IPv6-only services in Amazon VPC to communicate with IPv4-only services and networks.

□ Enable DNS64 Info

- Go to **Internet Gateways > Create IGW:**
  - **> Name:** MyProjectIGW



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.
MyProjectIGW

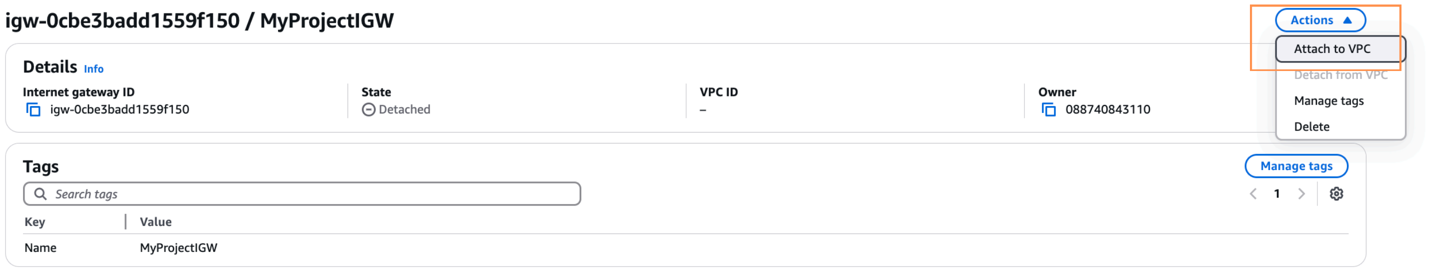
Tags - *optional*

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	MyProjectIGW

You can add 49 more tags.

  - **>** Attach to: **MyProjectVPC**



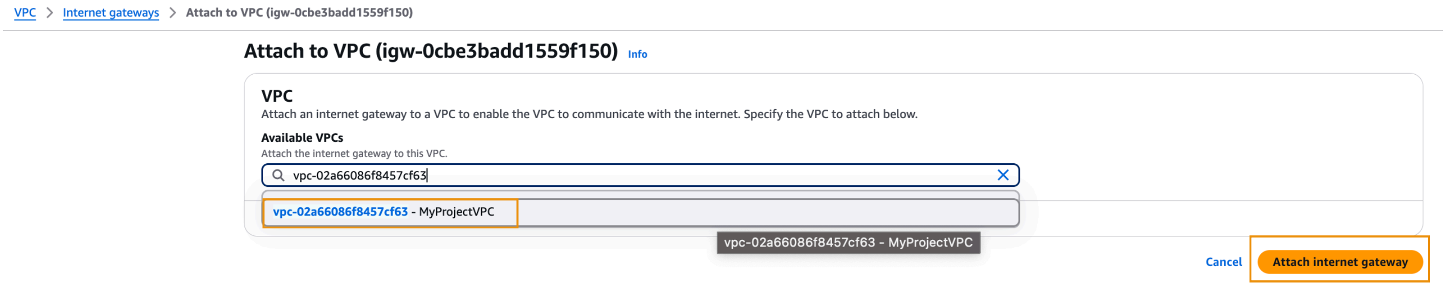
igw-0cbe3badd1559f150 / MyProjectIGW

Details Info

Internet gateway ID	State	VPC ID	Owner
igw-0cbe3badd1559f150	⊖ Detached	–	088740843110

Tags

Key	Value
Name	MyProjectIGW



VPC > Internet gateways > Attach to VPC (igw-0cbe3badd1559f150)

Attach to VPC (igw-0cbe3badd1559f150) Info

VPC

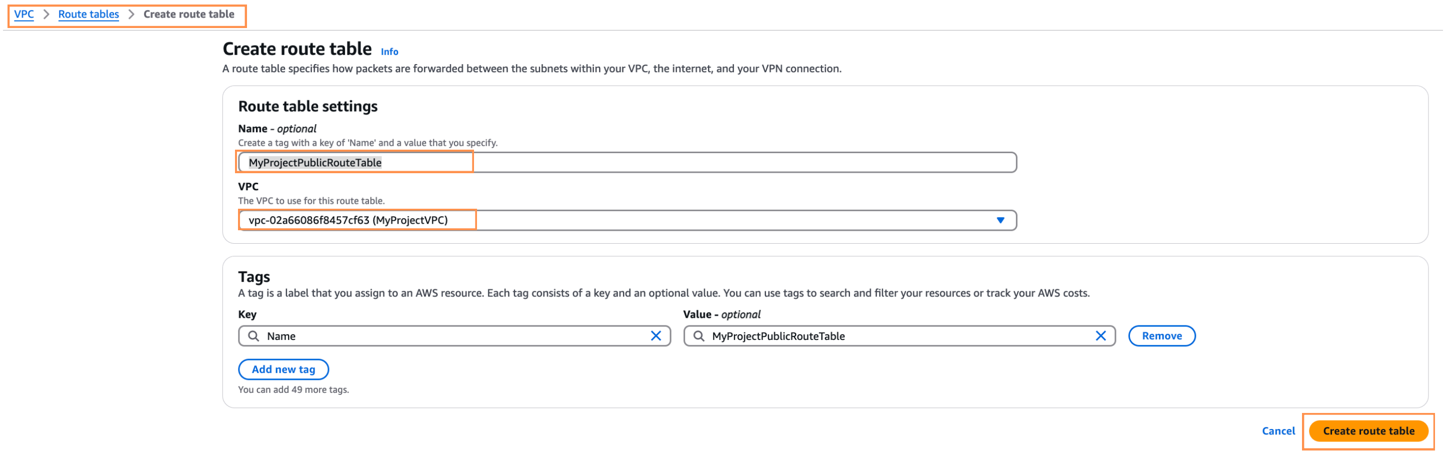
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.
vpc-02a66086f8457cf63

vpc-02a66086f8457cf63 - MyProjectVPC

vpc-02a66086f8457cf63 - MyProjectVPC

- Go to **Route Tables > Create route tables:**
    - **> Name:** MyProjectPublicRouteTable
    - **>** Select the VPC: MyProjectVPC



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.

MyProjectPublicRouteTable

VPC

The VPC to use for this route table.

vpc-02a66086f8457cf63 (MyProjectVPC)

Tags

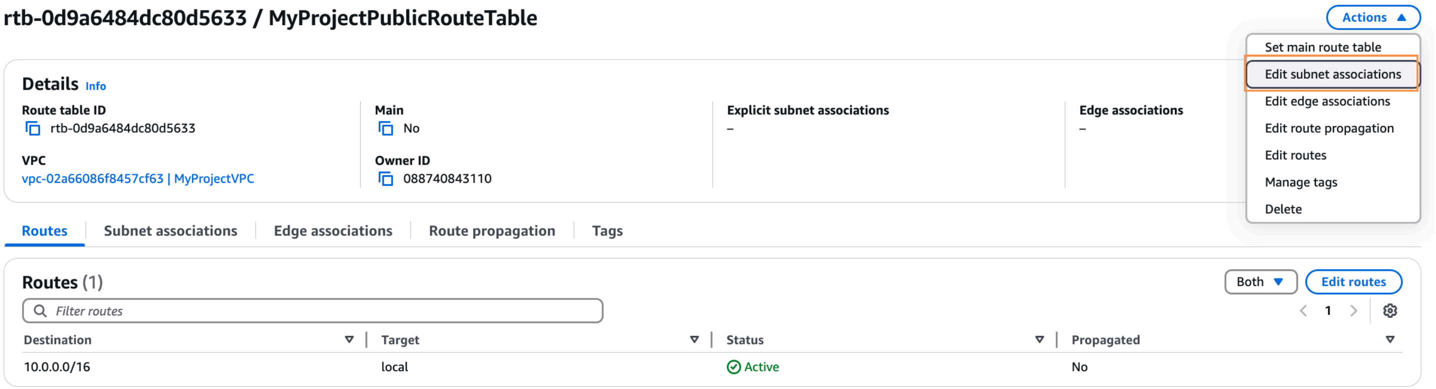
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	MyProjectPublicRouteTable	Remove

Add new tag

You can add 49 more tags.

- **>** Associate with: **MyProjectPublicSubnet**



rtb-0d9a6484dc80d5633 / MyProjectPublicRouteTable

Details Info

Route table ID: rtb-0d9a6484dc80d5633

Main: No

Explicit subnet associations: –

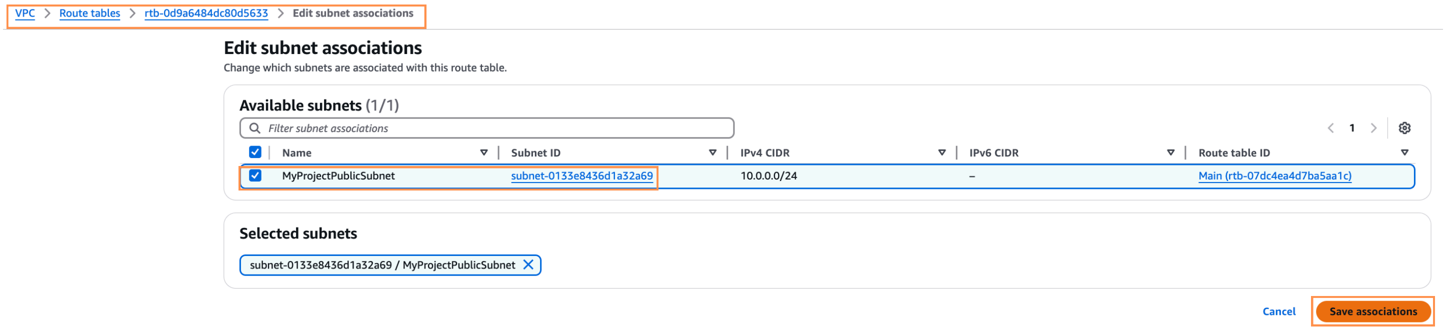
Edge associations: –

VPC: vpc-02a66086f8457cf63 | MyProjectVPC

Owner ID: 088740843110

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/16	local	✓ Active	No



VPC > Route tables > rtb-0d9a6484dc80d5633 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

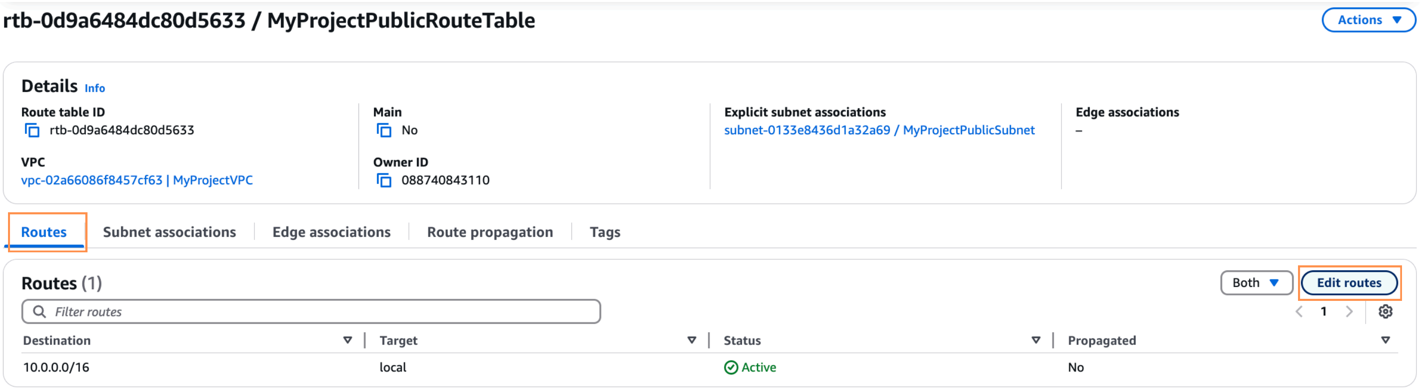
Available subnets (1/1)

☑	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
☑	MyProjectPublicSubnet	subnet-0133e8436d1a32a69	10.0.0.0/24	–	Main (rtb-07dc4ea4d7ba5aa1c)

Selected subnets

subnet-0133e8436d1a32a69 / MyProjectPublicSubnet

- **>** Add route: **>** Destination: **0.0.0.0/0** **>** Target: **MyProjectIGW**



rtb-0d9a6484dc80d5633 / MyProjectPublicRouteTable

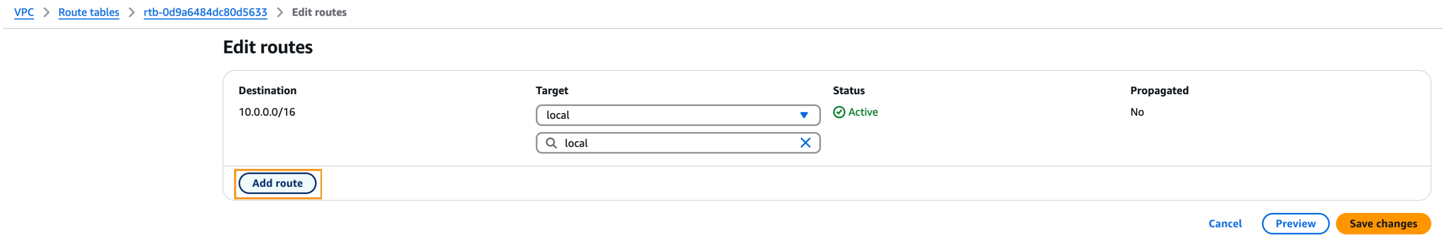
Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0d9a6484dc80d5633	No	subnet-0133e8436d1a32a69 / MyProjectPublicSubnet	–

VPC	Owner ID
vpc-02a66086f8457cf63 | MyProjectVPC	088740843110

Routes (1)

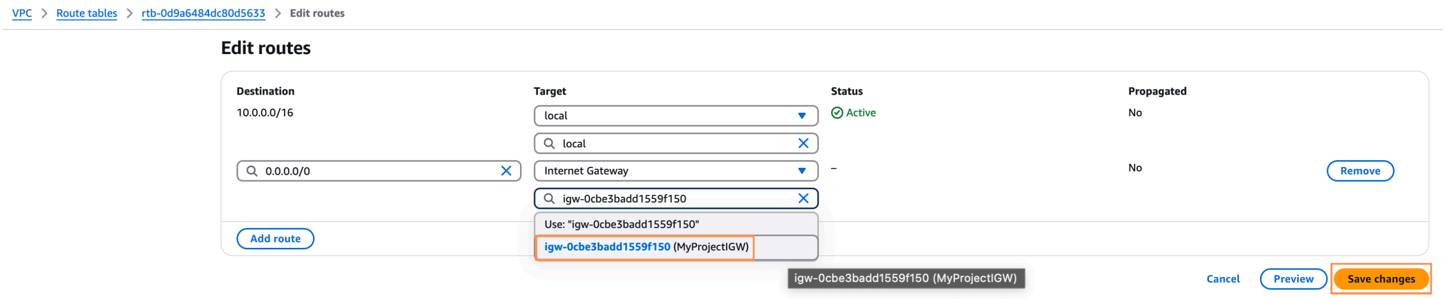
Destination	Target	Status	Propagated
10.0.0.0/16	local	✓ Active	No



VPC > Route tables > rtb-0d9a6484dc80d5633 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local ▼ local	✓ Active	No



VPC > Route tables > rtb-0d9a6484dc80d5633 > 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-0cbe3badd1559f150	–	No

Use: "igw-0cbe3badd1559f150"

igw-0cbe3badd1559f150 (MyProjectIGW)

igw-0cbe3badd1559f150 (MyProjectIGW)

Section 2: Create a Security Group

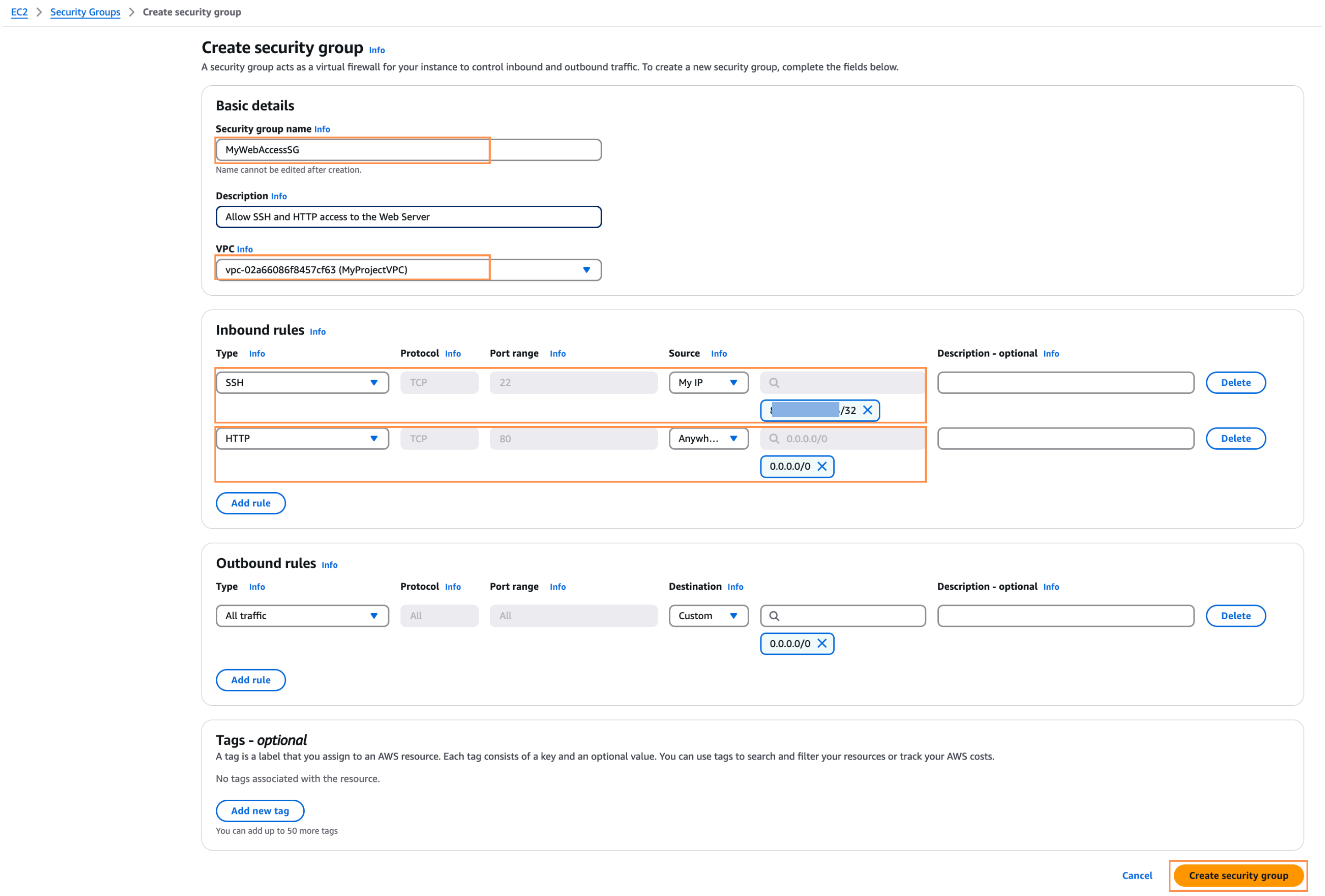
2 . Configuring the EC2 Security Group:

The Security Group is a virtual firewall used to defined the network rules for the EC2 instance, since I am setting up a basic web server (Apache), I opened two ports:

- ○ **SSH (port 22):** This allows to connect to the EC2 instance remotely to manage and configure it.
- ○ **HTTP (port 80):** This is the port the web server (Apache) will use to serve the static web page to the internet.

Steps:

- Go to **EC2** > **Security Groups → Create Security Group**
  - ○ **>** Name**: MyWebAccessSG**
  - ○ **>** VPC: **MyProjectVPC**
  - ○ **>** Inbound Rules:
    - ▪ SSH (port 22) – Source: My IP (or 0.0.0.0/0 if needed)
    - ▪ HTTP (port 80) – Source: 0.0.0.0/0
  - ○ **>** Outbound rules: leave default



EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info: MyWebAccessSG

Name cannot be edited after creation.

Description Info: Allow SSH and HTTP access to the Web Server

VPC Info: vpc-02a66086f8457cf63 (MyProjectVPC)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	My IP [redacted]/32	
HTTP	TCP	80	Anywh... 0.0.0.0/0	

Outbound rules Info

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom 0.0.0.0/0	

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.

No tags associated with the resource.

You can add up to 50 more tags

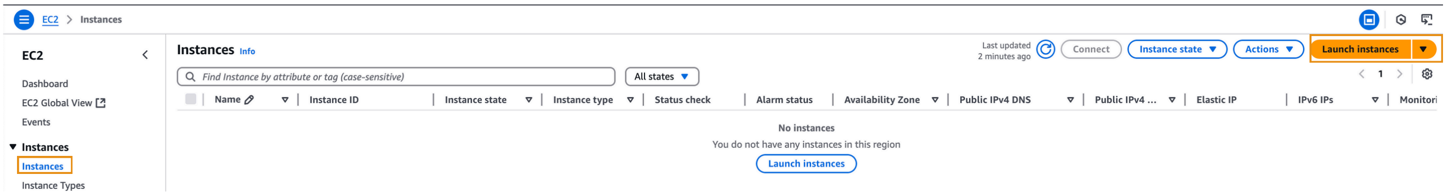
Section 3: Launch EC2 instance

5 . Launch EC2 Instance:

The goal is to automate the web server setup using **user data**, demonstrating **automation** and **basic Infrastructure as Code (IaC)** practices.

Steps:

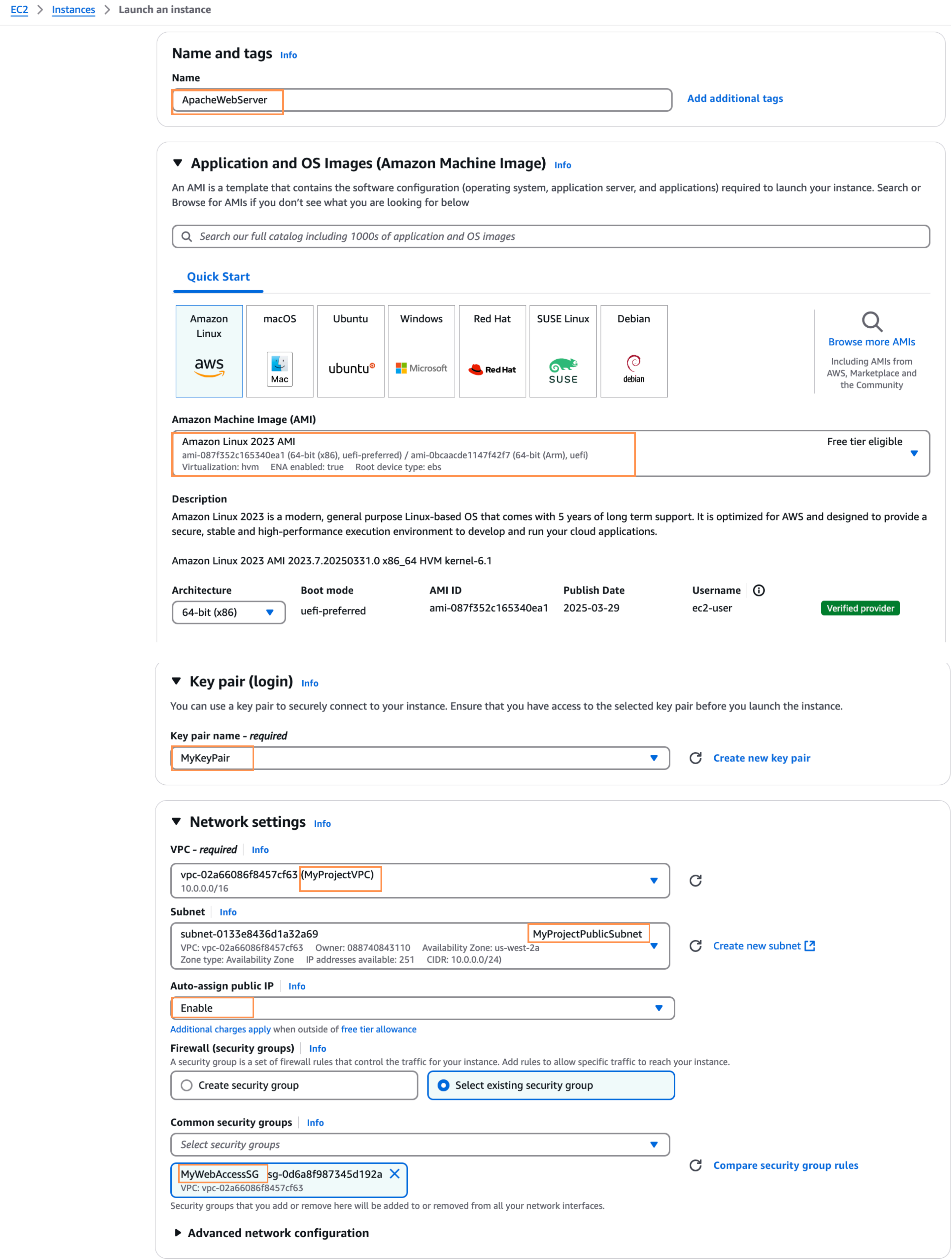
- Go to **EC2** > **Launch Instance**



No instances

You do not have any instances in this region

- **Instance specifications:**
    - **Instance Name**: **ApacheWebServer**
    - **AMI**: Amazon Linux 2023
    - **Instance Type**: t3.micro (ideal for lightweight projects)
    - **Network**: Select **MyProjectVPC**
    - **Subnet**: Choose **MyProjectPublicSubnet**
    - **Auto-assign Public IP**: Enabled (to ensure your instance is publicly accessible)
    - **Security Group**: Attach **WebAccessSG** (the previously created security group)
    - **Key Pair**: Select or create a new key pair (it will be needed for SSH access)



EC2 > Instances > Launch an instance

Name and tags Info

Name

ApacheWebServer

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-087f352c165340ea1 (64-bit (x86), uefi-preferred) / ami-0bcaacde1147f42f7 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250331.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username ⓘ	
64-bit (x86) ▼	uefi-preferred	ami-087f352c165340ea1	2025-03-29	ec2-user	Verified provider

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

MyKeyPair

Create new key pair

▼ Network settings Info

VPC - *required* | Info

vpc-02a66086f8457cf63 (MyProjectVPC)
10.0.0.0/16

Subnet | Info

subnet-0133e8436d1a32a69 MyProjectPublicSubnet
VPC: vpc-02a66086f8457cf63 Owner: 088740843110 Availability Zone: us-west-2a
Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.0.0/24)

Create new subnet

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

Common security groups | Info

Select security groups

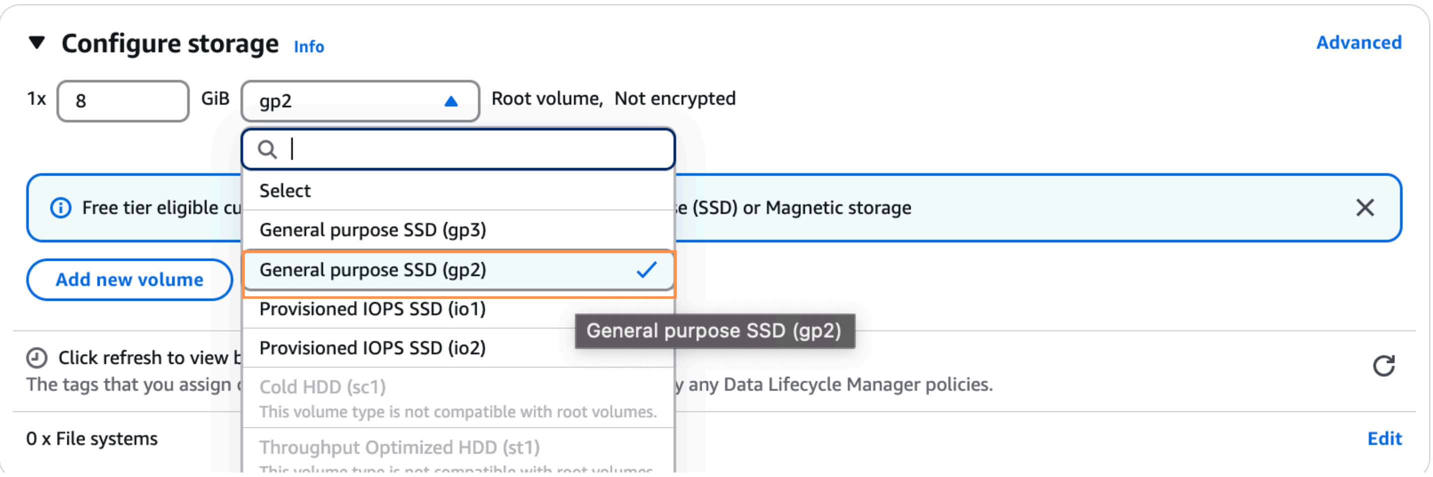
MyWebAccessSG sg-0d6a8f987345d192a ✕
VPC: vpc-02a66086f8457cf63

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▶ Advanced network configuration

- **Storage**: **General Purpose SSD (gp2)** with **8 GB** (is usually sufficient for small web projects).



▼ Configure storage Info

Advanced

1x 8 GiB gp2 ▲ Root volume, Not encrypted

Select

General purpose SSD (gp3)

General purpose SSD (gp2) ✓

Provisioned IOPS SSD (io1)

Provisioned IOPS SSD (io2)

Cold HDD (sc1)

This volume type is not compatible with root volumes.

Throughput Optimized HDD (st1)

General purpose SSD (gp2)

ⓘ Free tier eligible cu… …e (SSD) or Magnetic storage ✕

Add new volume

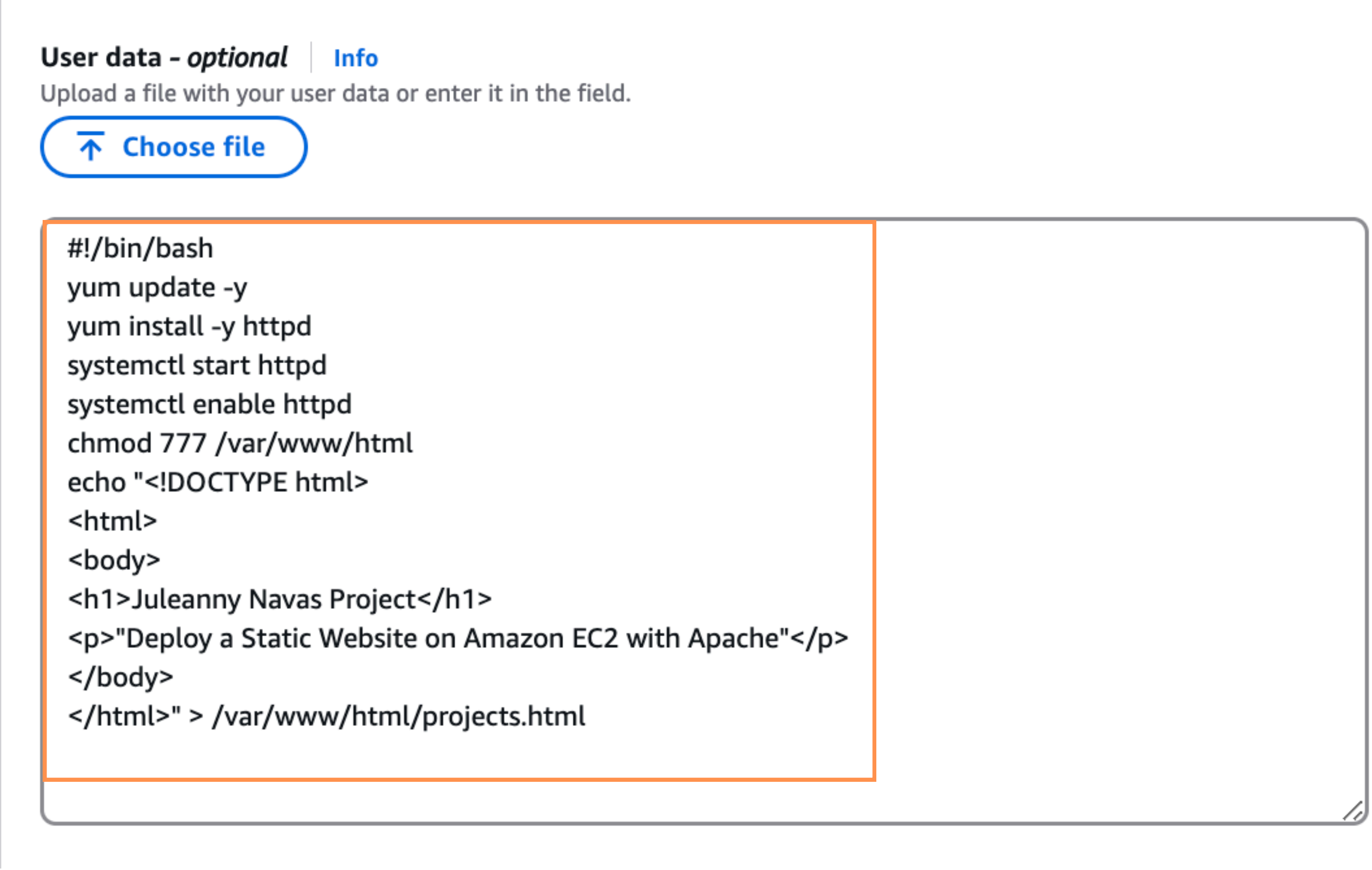
Click refresh to view b…

The tags that you assign … …y any Data Lifecycle Manager policies.

0 x File systems

Edit

- **User Data**: Add the following script:
  - This script updates the instance, **installs Apache (httpd)**, **starts the Apache** service, enables it to **run on boot**, and **creates a basic HTML file** (projects.html) in the **/var/www/html** directory. This ensures the web server is ready with content as soon as the instance is launched.
- **Launch**



User data - *optional* Info

Upload a file with your user data or enter it in the field.

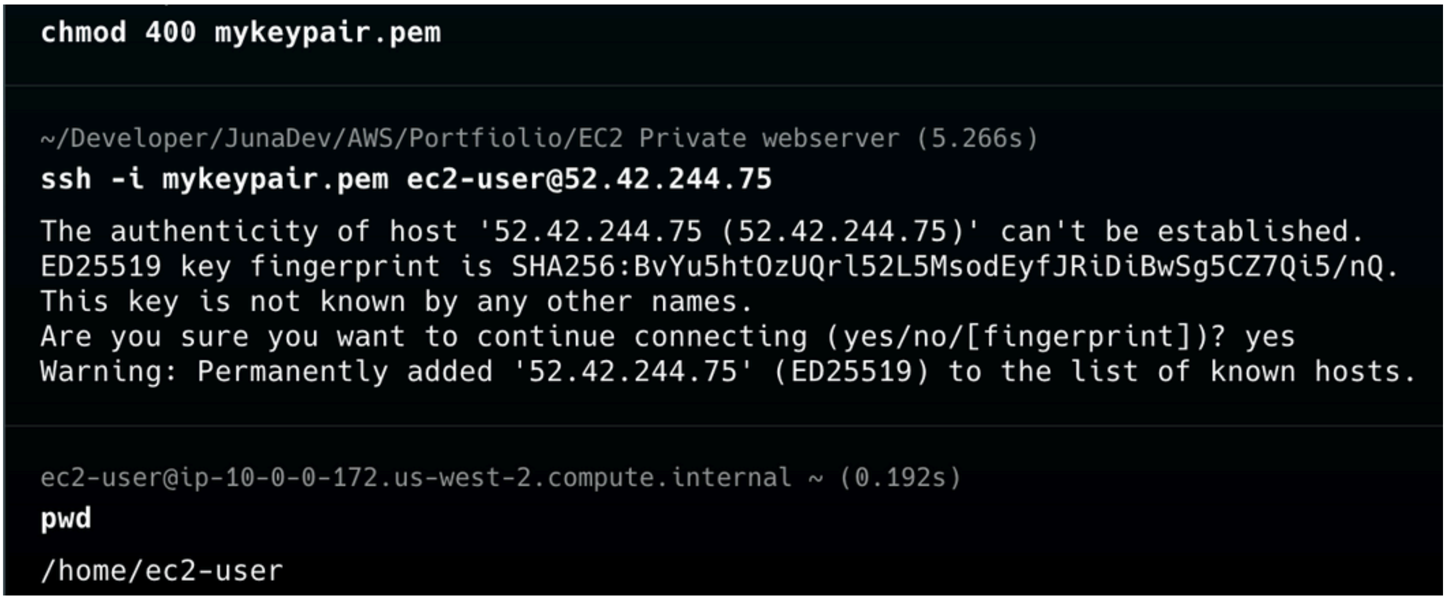
```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
chmod 777 /var/www/html
echo "<!DOCTYPE html>
<html>
<body>
<h1>Juleanny Navas Project</h1>
<p>"Deploy a Static Website on Amazon EC2 with Apache"</p>
</body>
</html>" > /var/www/html/projects.html
```

Section 4: Testing & Validation

4 . Connecting to the instance using SSH:

Steps:

- After download the .pem file to the local machine **> run chmod 400** command **>** SSH into the EC2 instance: **ssh -i** mykeypair.pem **ec2-user@**52.42.244.75

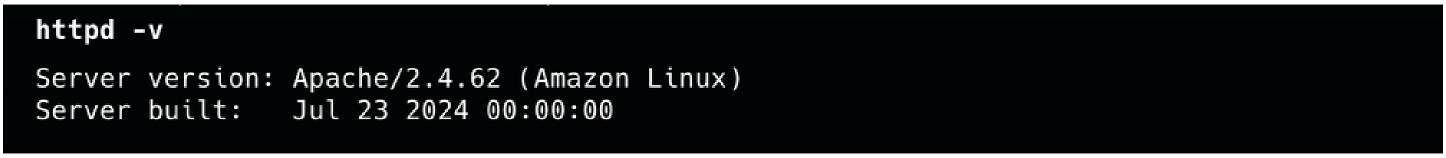


```
chmod 400 mykeypair.pem

~/Developer/JunaDev/AWS/Portfiolio/EC2 Private webserver (5.266s)
ssh -i mykeypair.pem ec2-user@52.42.244.75
The authenticity of host '52.42.244.75 (52.42.244.75)' can't be established.
ED25519 key fingerprint is SHA256:BvYu5htOzUQrl52L5MsodEyfJRiDiBwSg5CZ7Qi5/nQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.42.244.75' (ED25519) to the list of known hosts.

ec2-user@ip-10-0-0-172.us-west-2.compute.internal ~ (0.192s)
pwd
/home/ec2-user
```

5 . Verify Apache was installed successfully:

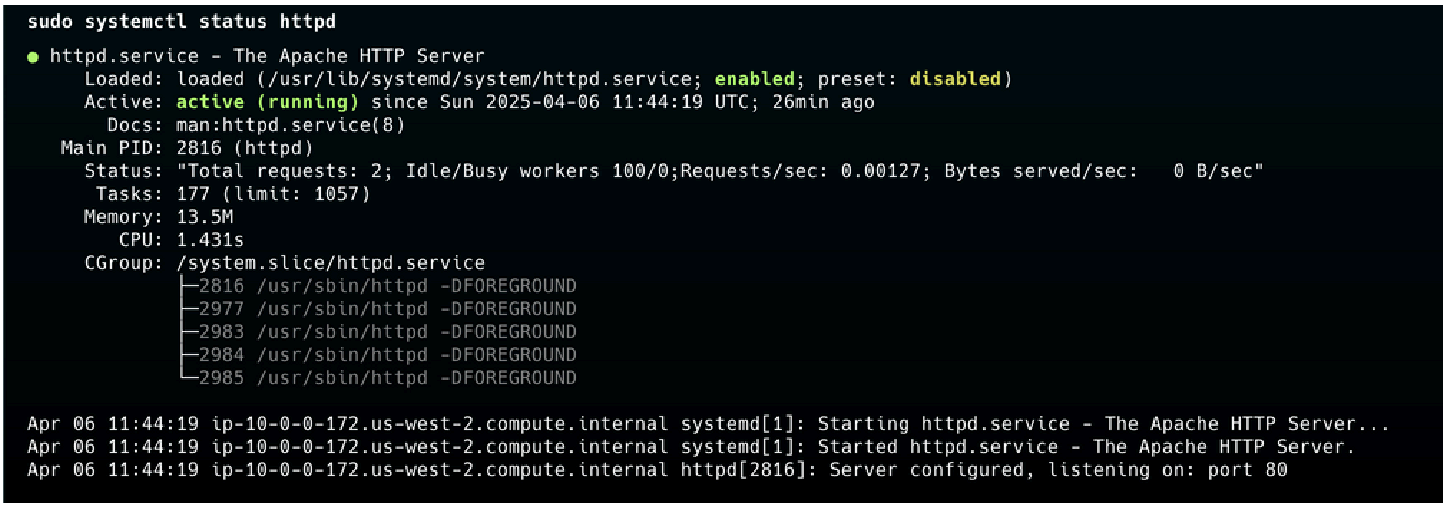
Steps:

- **Check the Apache version** to ensure it's installed: run **httpd -v**



```
httpd -v
Server version: Apache/2.4.62 (Amazon Linux)
Server built:   Jul 23 2024 00:00:00
```

- **Check the Apache service status**: run **sudo systemctl status httpd**

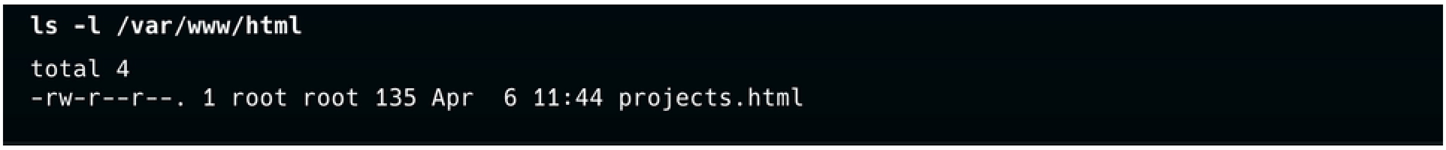


```
sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-04-06 11:44:19 UTC; 26min ago
       Docs: man:httpd.service(8)
   Main PID: 2816 (httpd)
     Status: "Total requests: 2; Idle/Busy workers 100/0;Requests/sec: 0.00127; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1057)
     Memory: 13.5M
        CPU: 1.431s
     CGroup: /system.slice/httpd.service
             ├─2816 /usr/sbin/httpd -DFOREGROUND
             ├─2977 /usr/sbin/httpd -DFOREGROUND
             ├─2983 /usr/sbin/httpd -DFOREGROUND
             ├─2984 /usr/sbin/httpd -DFOREGROUND
             └─2985 /usr/sbin/httpd -DFOREGROUND

Apr 06 11:44:19 ip-10-0-0-172.us-west-2.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Apr 06 11:44:19 ip-10-0-0-172.us-west-2.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Apr 06 11:44:19 ip-10-0-0-172.us-west-2.compute.internal httpd[2816]: Server configured, listening on: port 80
```

6 . List the content of the /var/www/html directory in a long format and display the content of projects.html file:

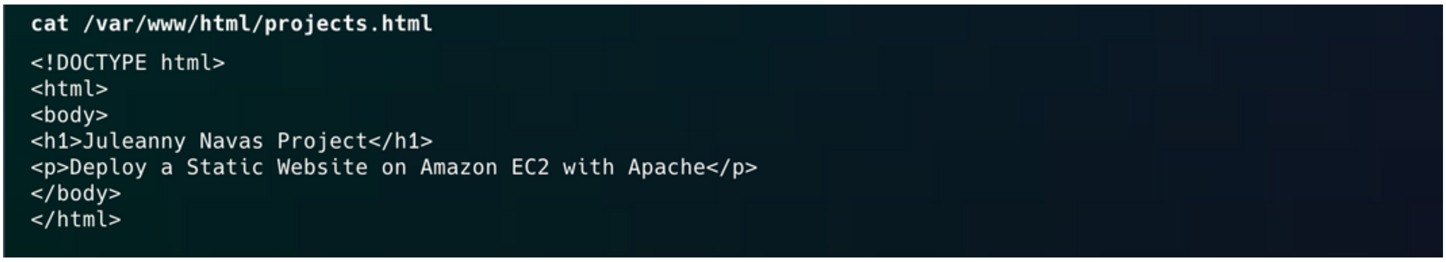
Steps:

- Run **ls -l /var/www/html:** This examine the content of the directory where web file are typically stored in an Apache web server (**/var/www/html**).



```
ls -l /var/www/html
total 4
-rw-r--r--. 1 root root 135 Apr  6 11:44 projects.html
```

- Run **cat /var/www/html/projects.html:**

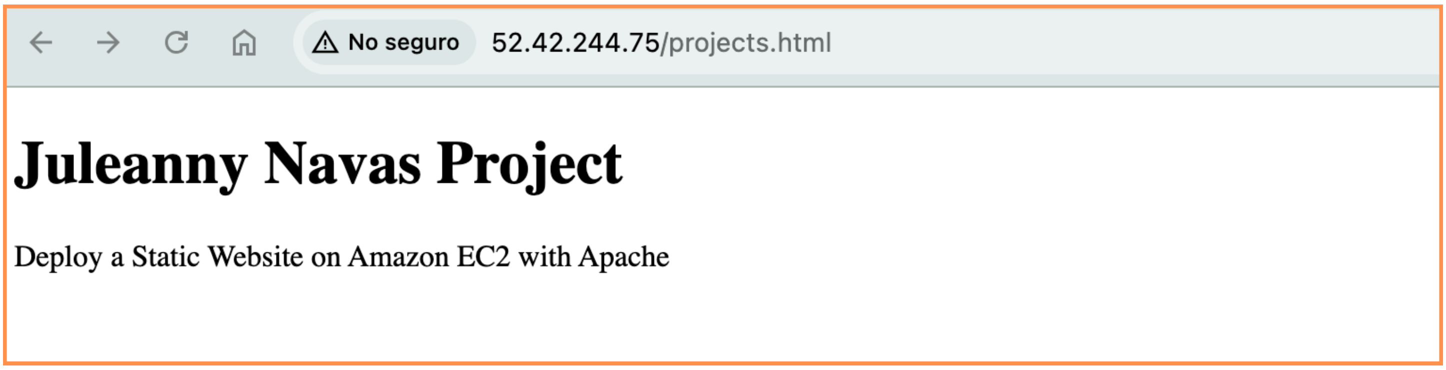


```
cat /var/www/html/projects.html
<!DOCTYPE html>
<html>
<body>
<h1>Juleanny Navas Project</h1>
<p>Deploy a Static Website on Amazon EC2 with Apache</p>
</body>
</html>
```

7 . Verify the Web page:

After the instance is running, I went to its **Public IPv4 address** in a browser:

http://<your-ec2-public-ip>/projects.html



Juleanny Navas Project

Deploy a Static Website on Amazon EC2 with Apache

Conclusions & Lessons Learned

Amazon EC2 Provides Flexible and Scalable Compute Resources:

- EC2 instances offer full control over the hosting environment, ideal for deploying custom websites and applications.
- Selecting the right instance type and security group settings is critical for performance and access control.

Apache Web Server is Reliable for Serving Static Content:

- Apache is easy to install and configure for hosting static websites.
- The default web directory /var/www/html allows quick deployment of HTML files with minimal setup.

Security Groups and Key Pairs are Essential for Secure Access:

- EC2 key pairs provide secure SSH access to the server, replacing passwords with encrypted credentials.
- Configuring security groups correctly ensures web traffic (HTTP/HTTPS) is allowed, while protecting the instance from unauthorized access.

Linux Command Line Skills are Crucial for Server Management:

- Commands like **ls, cat,** and **nano** were used to inspect, view, and edit website files.
- Understanding file permissions helped ensure that web content was readable by the web server.

Static Website Deployment is Straightforward but Teaches Core Concepts:

- Hosting a simple HTML page showed how web servers deliver content to users via public IP or domain.

Final Thoughts

This project provided valuable hands-on experience with **deploying a web server** and hosting static content in a cloud environment using **Amazon EC2 and Apache**. It served as a foundation for learning how to manage compute resources, work with **Linux systems**, and securely access cloud instances through **SSH**.

Successfully configuring the **Apache server** and serving the projects.html file allowed me to grasp the essentials of cloud hosting. It also reinforced the importance of security, proper file management, and understanding basic networking in cloud environments.

Overall, this project has laid the groundwork for more advanced topics like dynamic web applications, server automation, and full-stack deployment pipelines in AWS.