AWS EC2 Web Server Deployment Using Apache

Project Objective

The objective of this project is to launch an Amazon EC2 instance, configure secure access, install and run the Apache2 web server, and deploy a simple HTML web application that is accessible over the internet using HTTP.

Task 1: Create an EC2 Instance

□ Open a web browser and log in to the AWS Management Console using your AWS credentials.

□ From the AWS services menu, select **EC2** to open the EC2 Dashboard.

□ Click on Launch Instance to begin creating a new virtual server.

□ Choose an Amazon Machine Image (AMI)**:**

- Select Ubuntu **Server 22.04 LTS** as the operating system.

□ Select an instance type:

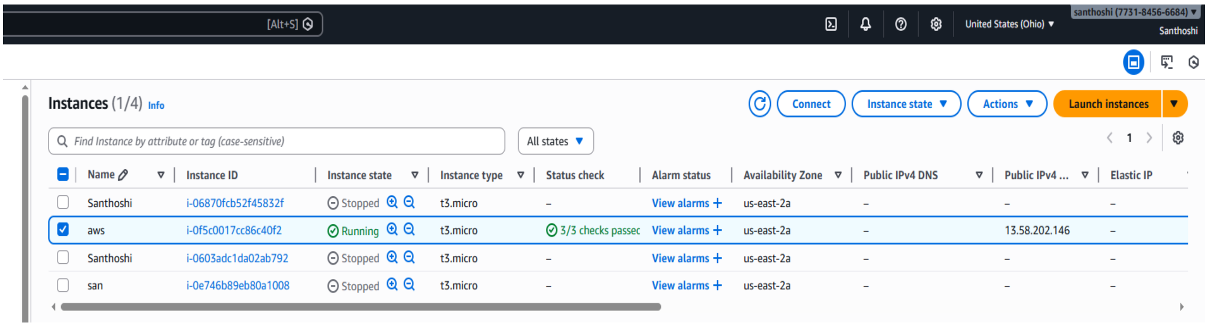
- Choose **t2.micro**, which is eligible for the AWS Free Tier.

□ Configure the **key pair**:

- Create a new key pair or select an existing one.
- Download the .pem file securely, as it is required for SSH access.

□ Review the instance configuration and click **Launch Instance**.

□ Verify that the instance state changes to **Running** in the EC2 dashboard.



Instances (1/4) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☐	Santhoshi	i-06870fcb52f45832f	Stopped	t3.micro	–	View alarms +	us-east-2a	–	–	–
☑	aws	i-0f5c0017cc86c40f2	Running	t3.micro	3/3 checks passec	View alarms +	us-east-2a	–	13.58.202.146	–
☐	Santhoshi	i-0603adc1da02ab792	Stopped	t3.micro	–	View alarms +	us-east-2a	–	–	–
☐	san	i-0e746b89eb80a1008	Stopped	t3.micro	–	View alarms +	us-east-2a	–	–	–

Task 2: Configure Security Groups (Firewall Rules)

□ Open **EC2 Dashboard** and select your instance.

□ Click the attached **Security Group** under the *Security* tab.

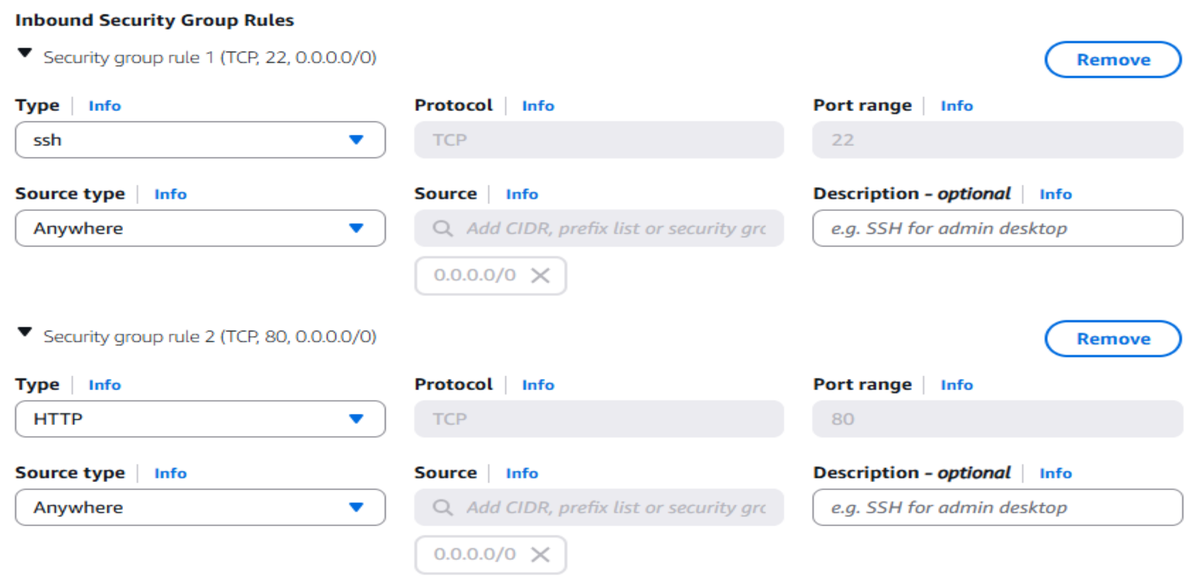
□ Edit **Inbound rules**.

□ Allow **SSH (TCP, Port 22)** from **My IP**.

□ Allow **HTTP (TCP, Port 80)** from **Anywhere (0.0.0.0/0)**.

□ Save the rules.

Protocol	Port	Source
SSH	22	My IP
HTTP	80	Anywhere (0.0.0.0/0)



Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type: ssh

Protocol: TCP

Port range: 22

Source type: Anywhere

Source: Add CIDR, prefix list or security grc

0.0.0.0/0 ✕

Description - *optional*: *e.g. SSH for admin desktop*

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type: HTTP

Protocol: TCP

Port range: 80

Source type: Anywhere

Source: Add CIDR, prefix list or security grc

0.0.0.0/0 ✕

Description - *optional*: *e.g. SSH for admin desktop*

Task 3: Connect to EC2 Using SSH

□ From the EC2 console, copy the **Public IPv4 address** of the running instance.

□ Open a terminal (Command Prompt / PowerShell / Linux terminal).

□ Navigate to the folder containing your **key pair (.pem)** file.

 Connect to the instance using the SSH command

```
ubuntu@ip-10-0-14-124:~$ ls
ubuntu@ip-10-0-14-124:~$
ubuntu@ip-10-0-14-124:~$
ubuntu@ip-10-0-14-124:~$
```

Task 4: Install and Start Apache2 Web Server

After logging in to the EC2 instance, update the system package repository.

sudo apt update

1. Install the **Apache2 web server** using the package manager.

   sudo apt install apache2 -y

2. Start the Apache2 service to run the web server.

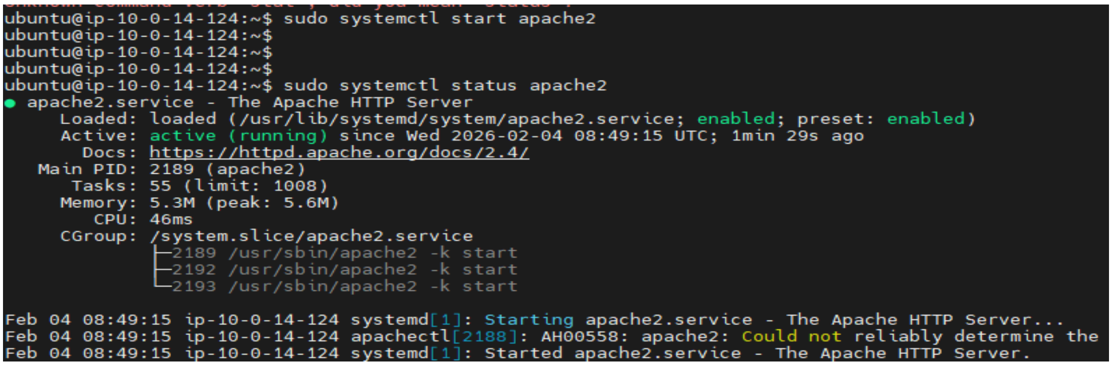
   sudo systemctl start apache2

3. Enable Apache2 to start automatically on system reboot.

   sudo systemctl enable apache2

4. Check the service status to confirm that Apache2 is running successfully.

   sudo systemctl status apache2



```
ubuntu@ip-10-0-14-124:~$ sudo systemctl start apache2
ubuntu@ip-10-0-14-124:~$
ubuntu@ip-10-0-14-124:~$
ubuntu@ip-10-0-14-124:~$
ubuntu@ip-10-0-14-124:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Wed 2026-02-04 08:49:15 UTC; 1min 29s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2189 (apache2)
      Tasks: 55 (limit: 1008)
     Memory: 5.3M (peak: 5.6M)
        CPU: 46ms
     CGroup: /system.slice/apache2.service
             ├─2189 /usr/sbin/apache2 -k start
             ├─2192 /usr/sbin/apache2 -k start
             └─2193 /usr/sbin/apache2 -k start

Feb 04 08:49:15 ip-10-0-14-124 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Feb 04 08:49:15 ip-10-0-14-124 apachectl[2188]: AH00558: apache2: Could not reliably determine the
Feb 04 08:49:15 ip-10-0-14-124 systemd[1]: Started apache2.service - The Apache HTTP Server.
```

Task 5: Test HTTP Access

- Go to the **AWS EC2 console** and copy the **Public IPv4 address** of your instance.

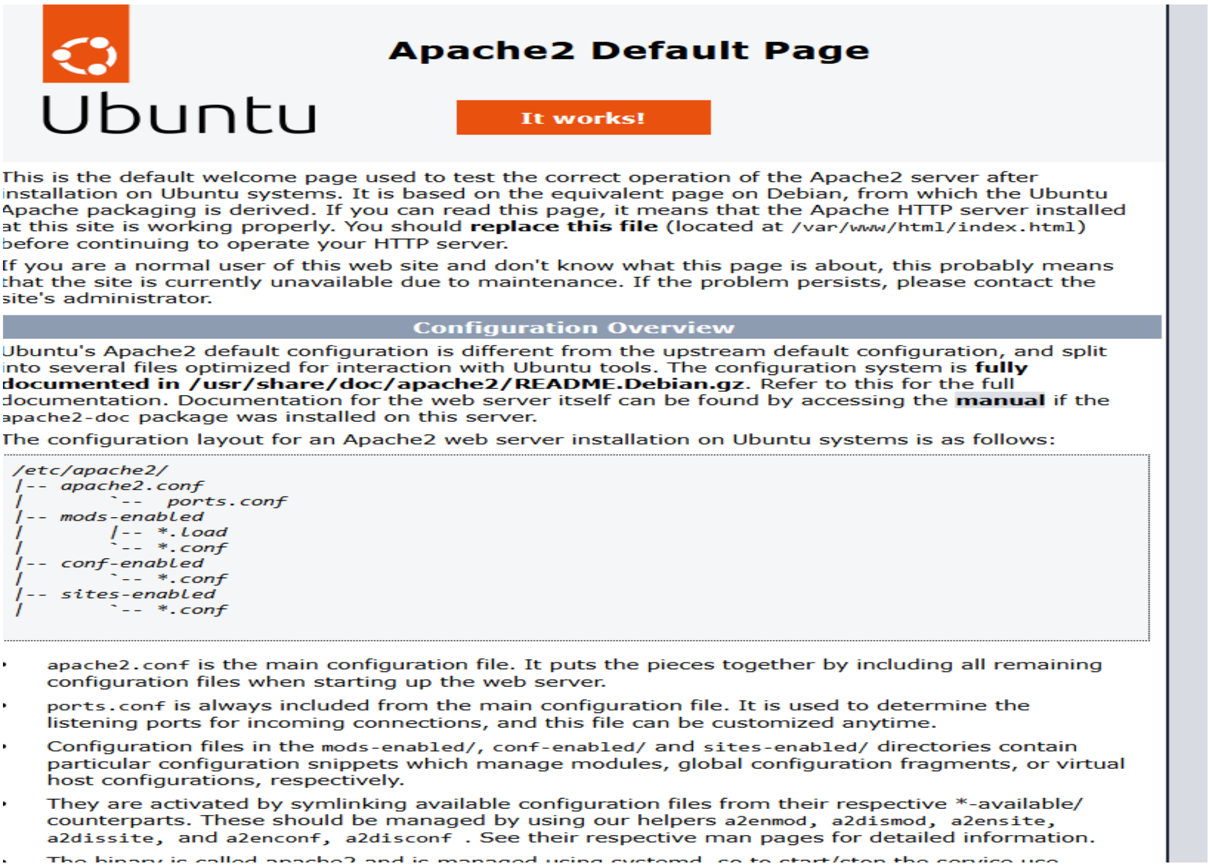
 Open a **web browser** on your system.

Type the public IP address in the address bar:

13.58.202.146

□ Press **Enter** to load the page.

□ Verify that the **Apache2 default page** is displayed, confirming HTTP access.



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2 and is managed using systemd, so to start/stop the service use

Task 6: Deploy a Sample Web Application

Navigate to the Apache web root directory:

```
cd /var/www/html
```

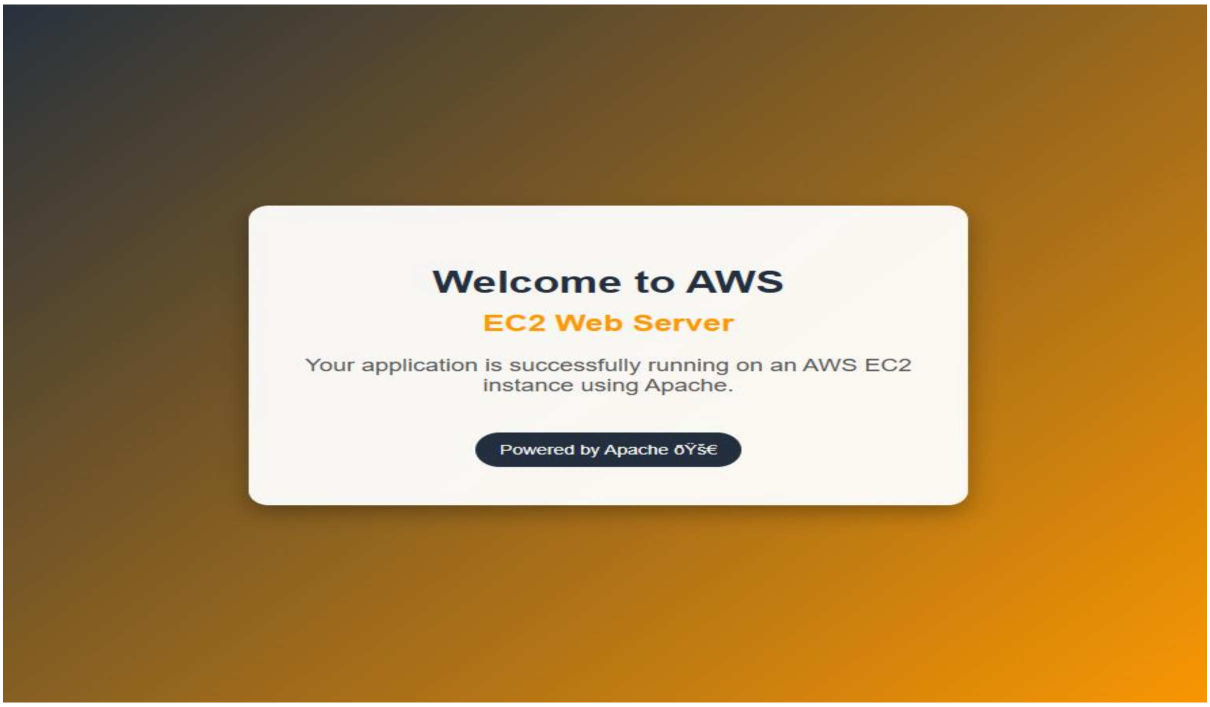
Create and open the main HTML file:

sudo nano index.html

□ Add the required HTML content to the file.

□ Save the file (CTRL +WQ!)

□ Refresh the browser to view the deployed web application.



Welcome to AWS

EC2 Web Server

Your application is successfully running on an AWS EC2 instance using Apache.

Conclusion

This project involved launching an AWS EC2 instance and deploying a web application using Apache2. It provided hands-on experience with EC2, security groups, and SSH access. The task helped build a strong foundation in cloud computing and web server deployment.