Setting Up a Web Server Using Apache2 on AWS Steps

- **Create an EC2 Instance:**
  - Log in to your AWS console.
  - Navigate to the EC2 service.
  - Launch a new instance.
  - Configure security groups to allow HTTP (port 80) traffic to your instance.

- **Connect to the Instance:**
  - Once the instance is running, use your SSH client to connect to it. You'll need the public IP address of the instance and the key pair you used during launch.

- **Install Apache2:**
  - Use the following command to install Apache2:

```
sudo apt install apache2 -y
```

- **Verify Apache2 Installation:**
  - Check the Apache2 version:

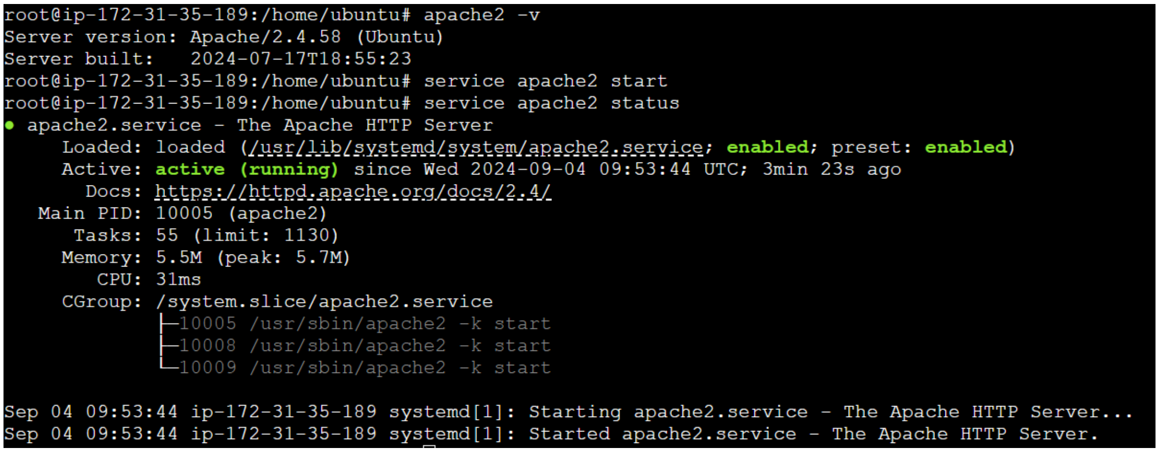
```
apache2 -v
```

- **Start Apache2 Service:**
  - Start the Apache2 service:

```
sudo service apache2 start
```

- **Check Service Status:**
  - Verify that the service is running:

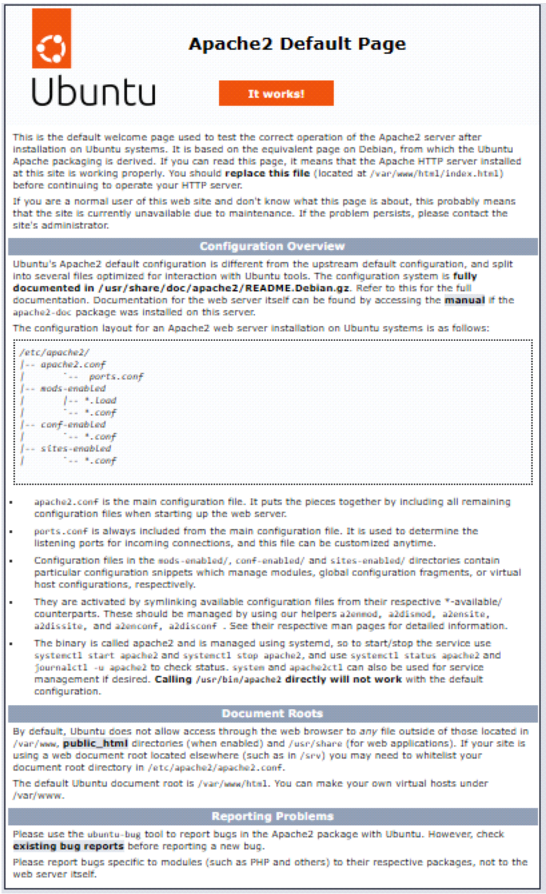
```
sudo service apache2 status
```



```
root@ip-172-31-35-189:/home/ubuntu# apache2 -v
Server version: Apache/2.4.58 (Ubuntu)
Server built:   2024-07-17T18:55:23
root@ip-172-31-35-189:/home/ubuntu# service apache2 start
root@ip-172-31-35-189:/home/ubuntu# service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-09-04 09:53:44 UTC; 3min 23s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 10005 (apache2)
      Tasks: 55 (limit: 1130)
     Memory: 5.5M (peak: 5.7M)
        CPU: 31ms
     CGroup: /system.slice/apache2.service
             ├─10005 /usr/sbin/apache2 -k start
             ├─10008 /usr/sbin/apache2 -k start
             └─10009 /usr/sbin/apache2 -k start

Sep 04 09:53:44 ip-172-31-35-189 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Sep 04 09:53:44 ip-172-31-35-189 systemd[1]: Started apache2.service - The Apache HTTP Server.
```

- **Access the Web Server:**
  - Open a web browser and enter the public IP address of your EC2 instance. You should see a default Apache2 welcome page.



Apache2 Default Page

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 `systemctl start apache2` and `systemctl stop apache2`, and use `systemctl status apache2` and `journalctl -u apache2` to check status. `system` and `apache2ctl` can also be used for service management if desired. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file outside of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under /var/www.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to their respective packages, not to the web server itself.

- **Create a Custom HTML File:**
  - Navigate to the web server's document root directory:
    ```
    cd /var/www/html/
    ```
  - Remove the existing index.html file:
    ```
    sudo rm index.html
    ```
  - Create a new index.html file:
    ```
    sudo vim index.html
    ```
  - Paste the following HTML code into the file:
    ```
    <!DOCTYPE html>
    <html>

    <head>
        <title>Congratulations, Chaitanya!</title>
        <style>
            body {
                background-color: #f0f0f0;
                font-family: Arial, sans-serif;
                text-align: center;
            }
    ```

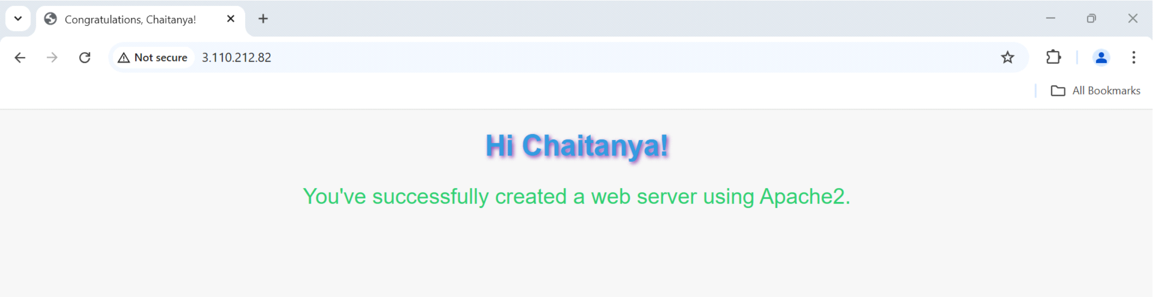
```
        h1 {
            color: #3498db;
            text-shadow: 2px 2px 4px #8e44ad;
        }
        p {
            font-size: 24px;
            color: #2ecc71;
        }
    </style>
</head>
<body>
    <h1>Hi Chaitanya!</h1>
    <p>You've successfully created a web server using
Apache2.</p>
</body>
</html>
```

  - Save the file and exit Vim.

- **Restart Apache2:**
  - Restart the Apache2 service to apply the changes:

```
sudo service apache2 restart
```

- **Access the Custom Page:**
  - Refresh the web browser page pointing to your instance's public IP. You should now see the custom HTML content.



Hi Chaitanya!

You've successfully created a web server using Apache2.