**Task: Host a CSS Template on Nginx and Apache (HTTPD) Simultaneously**

**Name:** Abhishek Hengade

**Batch:** B-56

This guide demonstrates how to run two different web servers, **Nginx** and **Apache (HTTPD)**, on the same server to host the exact same website content. This is achieved by configuring the servers to listen on different network ports to avoid conflicts.

**1. Understanding the Process**

By default, both Nginx and Apache attempt to listen for web traffic on the standard HTTP port (port 80). Since two applications cannot listen on the same port simultaneously, this would cause a conflict.

The solution is to configure them to use different ports. In this guide, we will:

1. Configure **Apache** to run on the default **port 80**.
2. Configure **Nginx** to run on a non-standard port, **8080**.
3. Point both servers to the same directory containing the website files.

**2. Step-by-Step Guide**

**Prerequisites:** An Amazon EC2 instance (e.g., Amazon Linux 2) with SSH access.

**Step 1: Install Nginx and Apache** First, connect to your EC2 instance via SSH and install both web server packages using the package manager.

Bash

# This command is for Amazon Linux 2 / RHEL / CentOS

sudo yum install httpd nginx -y

**Step 2: Prepare the Website Directory** Create a single directory that will hold your website's files. Both web servers will be configured to serve content from this location.

Bash

# Create the directory

sudo mkdir -p /var/www/my-template

# Place your website files here. For this example, we'll create a simple index file.

echo "<h1>Hosted by BOTH Nginx and Apache!</h1>" | sudo tee /var/www/my-template/index.html

**Step 3: Configure Apache to Use Port 80** Apache is typically configured for port 80 by default. We just need to ensure its configuration points to our new website directory. We can do this by editing its main configuration file.

Bash

# Open the Apache configuration file

sudo nano /etc/httpd/conf/httpd.conf

Inside the file, find the DocumentRoot directive and change its path to your new directory.

Apache

# Find this line:

DocumentRoot "/var/www/html"

# Change it to:

DocumentRoot "/var/www/my-template"

# Also find the <Directory> block for "/var/www/html" and change its path

<Directory "/var/www/my-template">

AllowOverride None

Require all granted

</Directory>

Save and close the file.

**Step 4: Configure Nginx to Use Port 8080** Next, we'll edit the Nginx configuration to change its listening port and document root.

Bash

# Open the Nginx configuration file

sudo nano /etc/nginx/nginx.conf

Inside the file, find the server block. Modify the listen and root directives as shown below.

Nginx

# Find the default server block

server {

# Change the port from 80 to 8080

listen 8080;

listen [::]:8080;

server\_name \_;

# Change the root directory to point to your template

root /var/www/my-template;

# Include server block directives;

include /etc/nginx/default.d/\*.conf;

error\_page 404 /404.html;

location = /404.html {

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

}

}

Save and close the file.

**Step 5: Update the EC2 Security Group** Navigate to your EC2 instance in the AWS Console. Select the **Security** tab and click on its security group. Edit the **inbound rules** to allow traffic on both ports.

* **Rule 1:** Type: HTTP, Port: 80, Source: Anywhere-IPv4 (0.0.0.0/0)
* **Rule 2:** Type: Custom TCP, Port: 8080, Source: Anywhere-IPv4 (0.0.0.0/0)

**Step 6: Start Both Web Servers** Now, start both services and enable them to launch on boot.

Bash

# Start and enable Apache (httpd)

sudo systemctl start httpd

sudo systemctl enable httpd

# Start and enable Nginx

sudo systemctl start nginx

sudo systemctl enable nginx

**Step 7: Verify the Setup** You can now access your website through both servers. Open your web browser and navigate to:

* **Apache:** http://<Your-EC2-Instance-IP>
* **Nginx:** http://<Your-EC2-Instance-IP>:8080

Both URLs should display the same index.html page you created in Step 2.

