

E25 Internship Assignment (Front-End development)

Anupa Prabhasara

Explain how you would create a virtual host for the domain `http://eight25-internship-assessment.test` for your website in both Apache and Nginx web servers, providing the necessary configuration code for each.

Optional

Create a Dockerfile to set up an Nginx container to serve the website you've created in the above steps. The container should expose port 80 and map it to port 3000 on the host. Additionally, configure a volume to sync the website's public files from the host to the container.

Apache Virtual Host on macOS

Step 1: Add Local Domain to Hosts File

I use macOS as my primary operating system, so I will continue using macOS when answering this question.

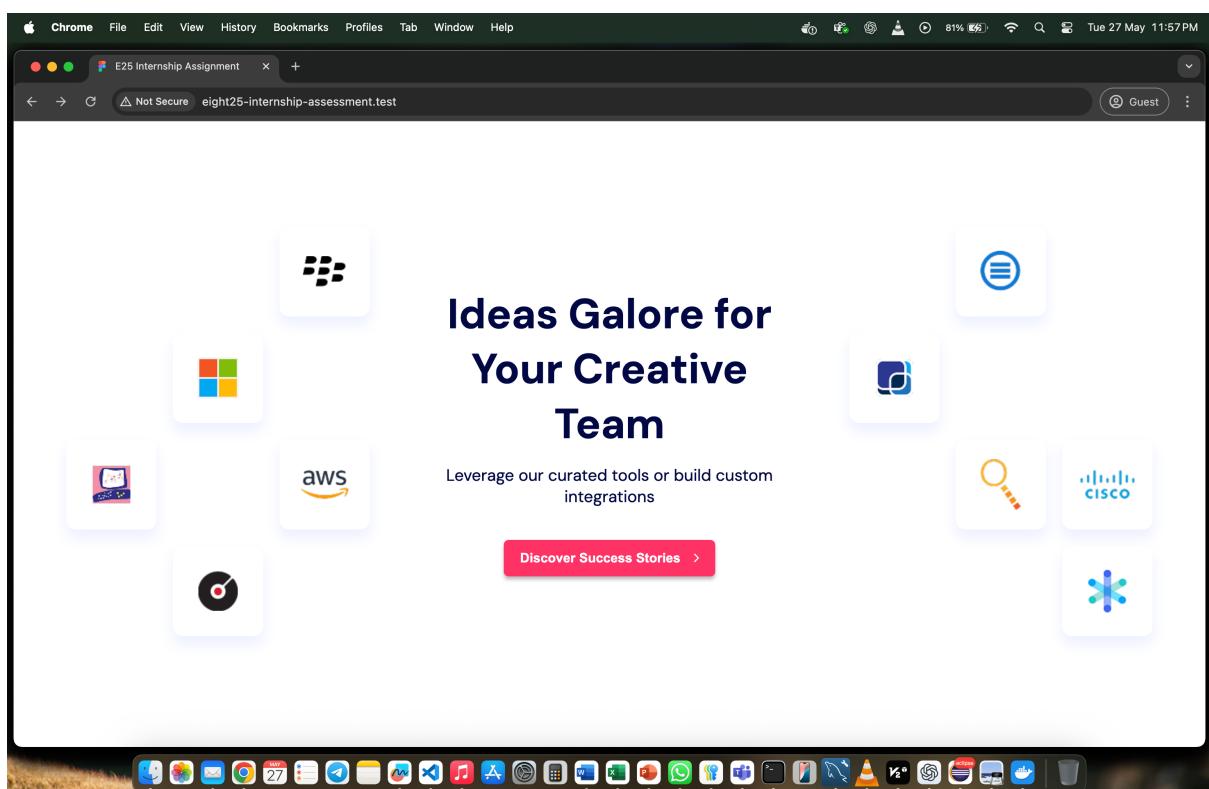
On macOS, edit `/etc/hosts` to map your custom domain to localhost.

```
sudo nano /etc/hosts
```

Add this line:

```
127.0.0.1 eight25-internship-assessment.test
```

Save and exit.



Step 2: Prepare Your Project Directory

Make sure your HTML files are in the specified directory:

```
mkdir -p /Users/anuprabhasara/Documents/HTML
```

Step 3: Enable Apache and Edit Virtual Hosts Config

Start Apache if not already running:

```
sudo apachectl start
```

Edit Apache's virtual host config:

```
sudo nano /etc/apache2/extra/httpd-vhosts.conf
```

Add this block:

```
<VirtualHost *:80>
```

```
ServerName eight25-internship-assessment.test
DocumentRoot "/Users/anupaprabhasara/Documents/HTML"
ErrorLog "/private/var/log/apache2/eight25-error.log"
CustomLog "/private/var/log/apache2/eight25-access.log" common
```

```
<Directory "/Users/anupaprabhasara/Documents/HTML">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
</VirtualHost>
```

Step 4: Enable Virtual Hosts in Main Config

Edit the main Apache config file:

```
sudo nano /etc/apache2/httpd.conf
```

Uncomment this line (remove the #):

```
# Include /private/etc/apache2/extra/httpd-vhosts.conf
```

Save and exit. Then restart Apache:

```
sudo apachectl restart
```

Nginx Virtual Host on macOS

Step 1: Install Nginx via Homebrew

```
brew install nginx
```

Step 2: Add Your Site Directory

Ensure this exists:

```
mkdir -p /Users/anupaprabhasara/Documents/HTML
```

Step 3: Edit Nginx Config

```
nano /opt/homebrew/etc/nginx/nginx.conf
```

Add this block inside the http {} section:

```
server {
    listen 80;
    server_name eight25-internship-assessment.test;

    root /Users/anupaprabhasara/Documents/HTML;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

```
access_log /usr/local/var/log/nginx/eight25-access.log;  
error_log /usr/local/var/log/nginx/eight25-error.log;  
}
```

Step 4: Restart Nginx

```
brew services restart nginx
```

Docker + Nginx + Volume + Port Mapping

Folder Structure

```
/Users/anupaprabhasara/Documents/HTML/
```


```
|— index.html  
|— default.conf  
|— Dockerfile
```

Step 1: Create default.conf

```
server {  
    listen 80;  
    server_name localhost;  
  
    root /usr/share/nginx/html;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Step 2: Create Dockerfile

```
FROM nginx:latest  
  
COPY default.conf /etc/nginx/conf.d/default.conf  
  
EXPOSE 80
```

A screenshot of a code editor window titled "Dockerfile" with a close button. The editor shows the following content:

```
1 FROM nginx:latest  
2  
3 COPY default.conf /etc/nginx/conf.d/default.conf  
4  
5 EXPOSE 80
```

Step 3: Build and Run the Docker Container

```
cd /Users/anupaprabhasara/Documents/HTML
```

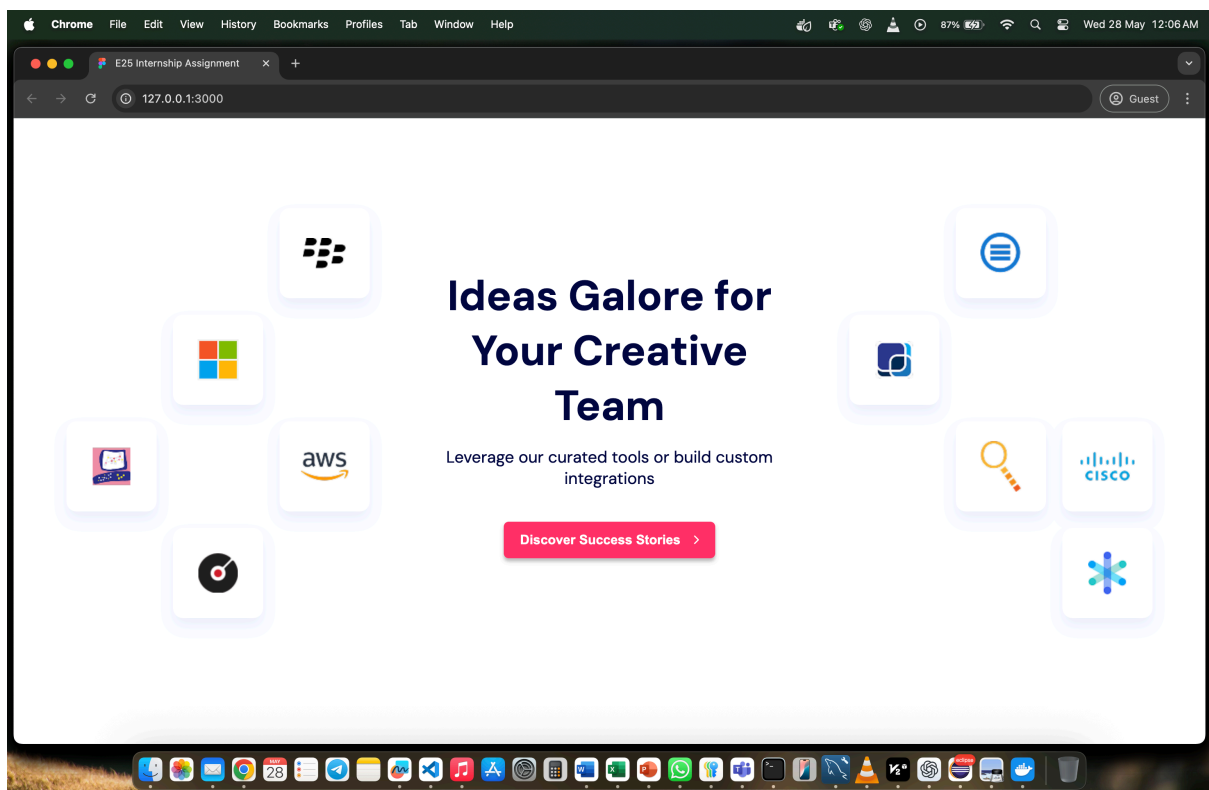
```
docker build -t eight25-nginx .
```

```
docker run -d \  
--name eight25-nginx-container \  
-p 3000:80 \  
-v "$PWD":/usr/share/nginx/html \  
eight25-nginx
```

Step 5: Test in Browser

Visit: <http://localhost:3000>

The browser should display the content defined in index.html.



Note: I have practical experience working with Apache and Nginx web servers on Windows, Ubuntu Server, and macOS environments. This assignment marks my first hands-on experience with Docker, which I explored and successfully implemented by learning through community forums and guidance from ChatGPT.