

**Pratik Agrawal**

**500123601**

**Devops B2**

## **Lab Exercise 4- Building a Docker Image for an HTML App Using Nginx**

### **1. Setup**

You will need:

- Docker installed on your machine.
- A simple HTML file for the app.

### **2. Step 1: Create the HTML File**

Create a directory for your HTML app and place an index.html file in it.

```
mkdir nginx-html-app
```

```
cd nginx-html-app
```

```
C:\Users\prati>cd nginx-html-app
```

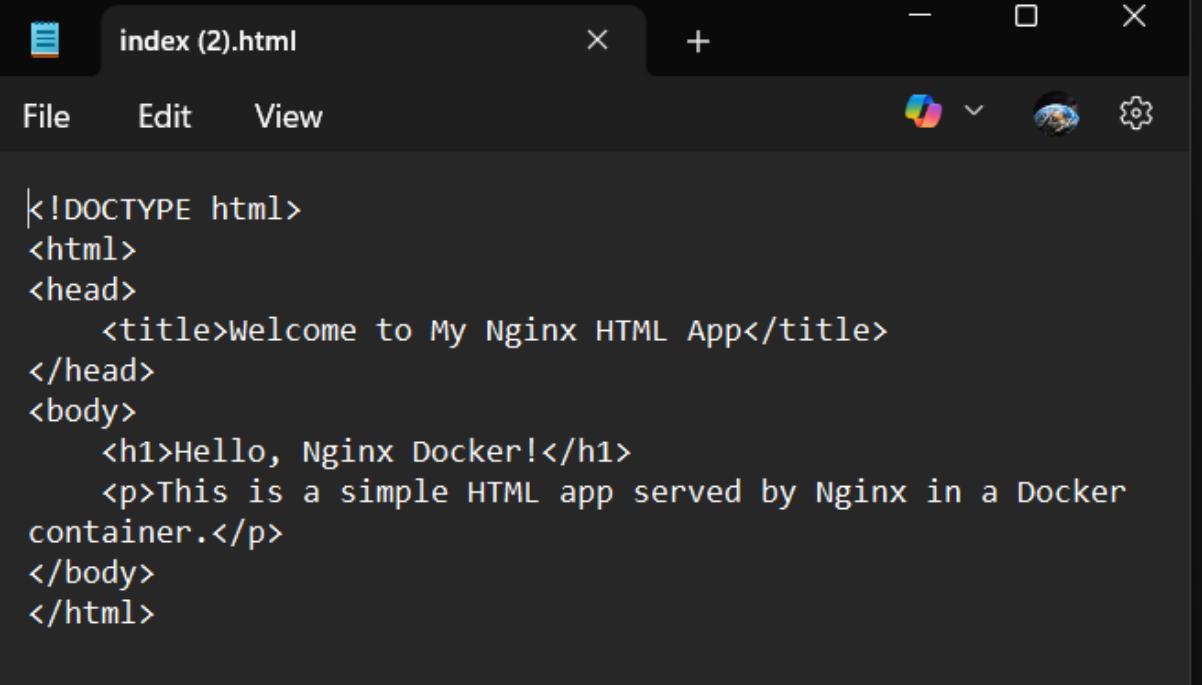
Inside the nginx-html-app directory, create the HTML file.

```
touch index.html
```

Edit the index.html file with the following content (or any custom HTML content you want):

```
<!DOCTYPE html>
<html>
<head>
  <title>Welcome to My Nginx HTML App</title>
</head>
<body>
  <h1>Hello, Nginx Docker!</h1>
  <p>This is a simple HTML app served by Nginx in a Docker container.</p>
</body>
```

```
</html>
```



```
index (2).html
```

```
File Edit View
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Welcome to My Nginx HTML App</title>
</head>
<body>
    <h1>Hello, Nginx Docker!</h1>
    <p>This is a simple HTML app served by Nginx in a Docker container.</p>
</body>
</html>
```

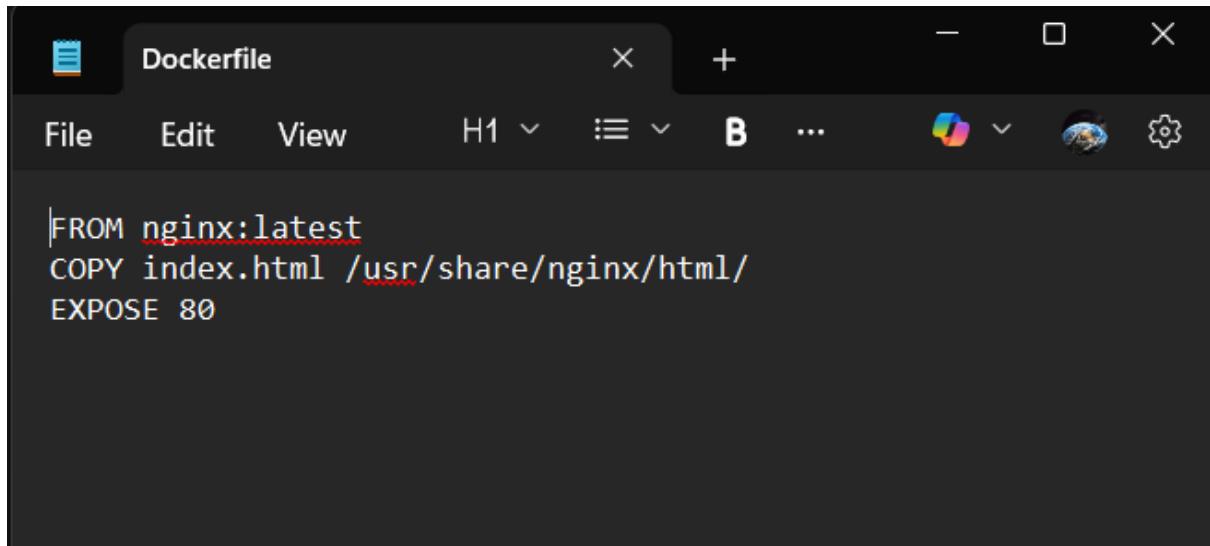
### 3. Step 2: Create a Dockerfile

In the same directory, create a Dockerfile. This file will define how to build the Docker image using Nginx as the base image.

```
touch Dockerfile
```

Edit the Dockerfile and add the following content:

```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```



```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```

#### 4. Step 3: Build the Docker Image

Now that you have the Dockerfile and index.html, it's time to build the Docker image.

Run the following command to build the image, giving it a tag (e.g., nginx-html-app):

```
docker build -t nginx-html-app .
```

```
C:\Users\prati\nginx-html-app>docker build -t nginx-html-app .
[+] Building 0.9s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerrcignore
=> [internal] load context: 2B
=> [internal] load build context
=> [internal] transfer context: 31B
=> [1/2] FROM docker.io/library/nginx:latest@sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d31
=> => resolve docker.io/library/nginx:latest@sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d31
=> [2/2] COPY index.html /usr/share/nginx/html/
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:ad2ea8d1b0f3cb6c1ff5707a308185047beb19007f2867e7e4c14196702e6e24
=> => exporting config sha256:f798eb9b99d59a9fe9ac3bd74dcadbc90519c156198b19f5286ca2b2885fdb1
=> => exporting attestation manifest sha256:a07031e219724ceec970876034c6ccbc3903a22d61373dc33b4fc01752387538
=> => exporting manifest list sha256:6f593b122ff5e225fe9beb9d8f99f6ea1884883b0b0b81310b9ee4d5d40f320f
=> => naming to docker.io/library/nginx-html-app:latest
=> => unpacking to docker.io/library/nginx-html-app:latest
  View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/6mktb7bymhjs20u5u4clo1ih5
```

Docker will use the Nginx base image, copy your index.html into the appropriate directory, and build the image.

#### 5. Step 4: Run the Docker Container

After building the image, you can run the container with the following command:

```
docker run -d -p 8006:80 nginx-html-app
```

```
C:\Users\prati\nginx-html-app>docker run -d -p 8006:80 nginx-html-app
86680e450cd809e0a27811a5b606f89daee15f0096649aa42ea9448d63eec086
```

This command runs the container in detached mode (-d) and maps port 8006 on your host machine to port 80 inside the container, where Nginx is serving your HTML app.

## 6. Step 5: Verify

Open a browser and go to <http://localhost:8006>. You should see your HTML page with the message “Hello, Nginx Docker!”.



## Hello, Nginx Docker!

This is a simple HTML app served by Nginx in a Docker container.

## 7. Step 6: Stop and Remove the Container

Once you're done, you can stop and remove the container:

```
C:\Users\prati\nginx-html-app>docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
86680e450cd8      nginx-html-app     "/docker-entrypoint..."   6 minutes ago      Up
, [::]:8006->80/tcp   ecstatic_cohen
d73989e7af93      pratikragrawal/500123601  "nginx -g 'daemon of..."  40 minutes ago    Up
, [::]:8080->80/tcp   mycontainer

C:\Users\prati\nginx-html-app>docker stop 86680e450cd8
86680e450cd8

C:\Users\prati\nginx-html-app>docker rm 86680e450cd8
86680e450cd8

C:\Users\prati\nginx-html-app>
```

```
docker ps # to see running containers
```

```
docker stop <container-id>
```

```
docker rm <container-id>
```