

Name: Aarushi
SAP ID: 500105028
Roll no.: R2142220004
Batch: DevSecOps B1:H

Lab Exercise 5- 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
```

```
AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security
$ mkdir nginx-html-app

AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security
$ cd nginx-html-app/

AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app
```

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

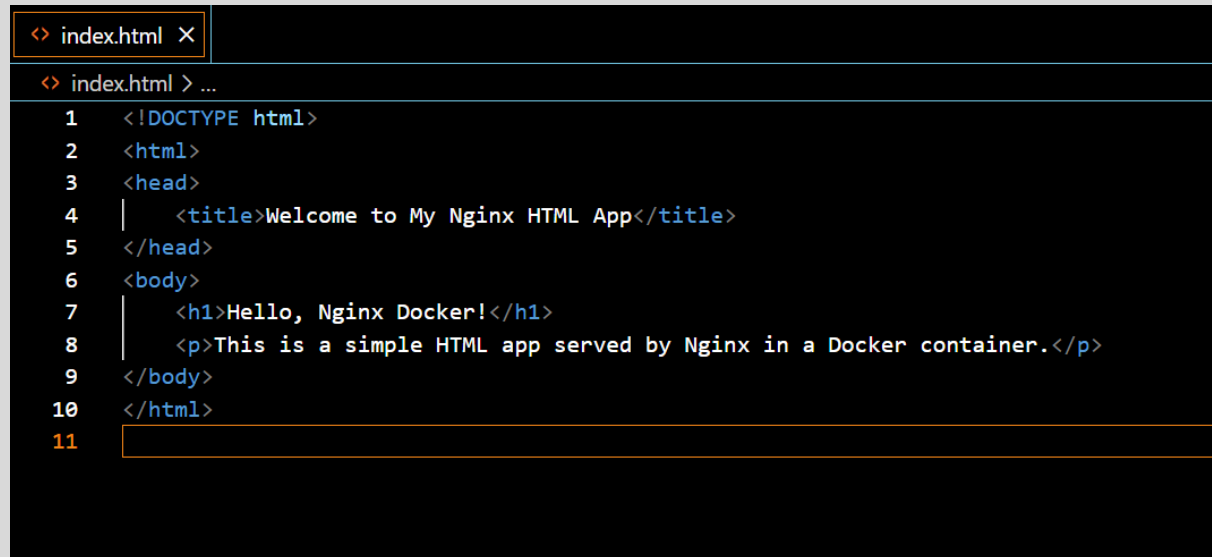
```
touch index.html
```

```
AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app
$ 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>
```

A screenshot of a code editor with a dark theme. The top tab is labeled 'index.html' with a close button. The editor shows the following HTML code:

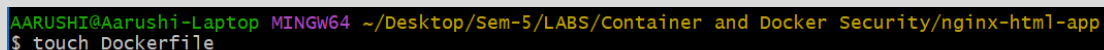
```
<!DOCTYPE html>
<html>
<head>
  <title>Welcome to My Nginx HTML App</title>
</head>
<body>
  <h1>Hello, Nginx Docker!</h1>
  <p>This is a simple HTML app served by Nginx in a Docker container.</p>
</body>
</html>
```

The line numbers 1 through 11 are visible on the left side of the editor.

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

A terminal window screenshot showing the command to create a Dockerfile. The prompt is 'AARUSHI@Aarushi-Laptop MINGW64' and the command is 'touch Dockerfile'. The path shown is '~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app'.

```
AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app
$ touch Dockerfile
```

Edit the Dockerfile and add the following content:

```
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 .
```

```
AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app
$ docker build -t nginx-html-app .
[+] Building 0.3s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 105B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 268B
=> [1/2] FROM docker.io/library/nginx:latest
=> [2/2] COPY index.html /usr/share/nginx/html/
=> exporting to image
=> exporting layers
=> writing image sha256:c50648c20a4bdc64deb085c29b8a0baba8156aa8d26509e749f932f69d9684f5
=> naming to docker.io/library/nginx-html-app
What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview
```

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 8080:80 nginx-html-app
```

```
AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app
$ docker run -d -p 8080:80 nginx-html-app
06559263ea956166f1b551baf979827276a3cc24b27993d54ec7cd41c3b75562
```

This command runs the container in detached mode (-d) and maps port 8080 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:8080>. You should see your HTML page with the message “Hello, Nginx Docker!”.

← ↻ ⓘ localhost:8080

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:

`docker ps` # to see running containers

```
AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app
$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
06559263ea95   nginx-html-app  "/docker-entrypoint..."  2 minutes ago  Up 2 minutes  0.0.0.0:8080->80/tcp      trusting_ganguly
```

`docker stop <container-id>`

```
AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app
$ docker stop 06559263ea95
06559263ea95
```

`docker rm <container-id>`

```
AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app
$ docker rm 06559263ea95
06559263ea95
```

```
AARUSHI@Aarushi-Laptop MINGW64 ~/Desktop/Sem-5/LABS/Container and Docker Security/nginx-html-app
$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
```