**Lab Exercise 5- Building a Docker Image for an HTML App Using Nginx**

**1. Setup**

You will need:

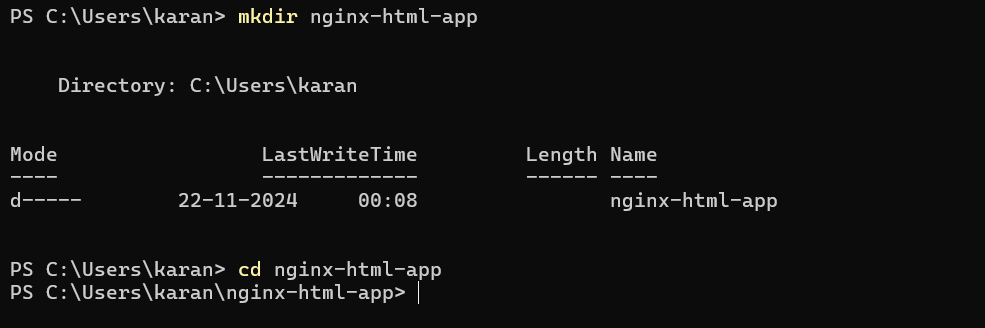
* Docker installed on your machine.
* A simple HTML file for the app.

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

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

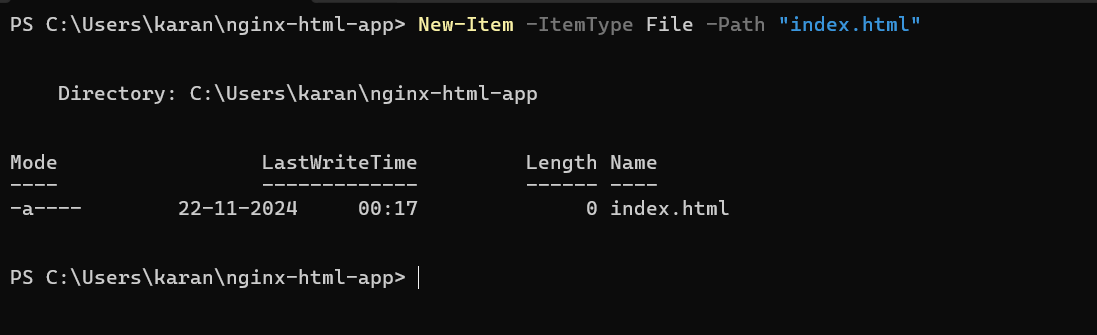
mkdir nginx-html-app

cd nginx-html-app



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>

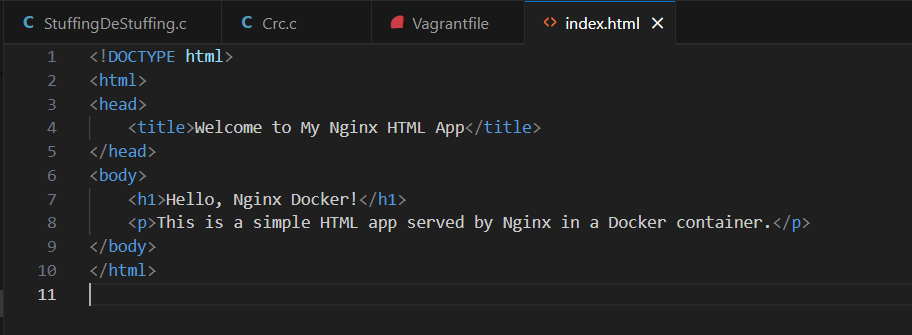
<body>

<h1>Hello, Nginx Docker!</h1>

<p>This is a simple HTML app served by Nginx in a Docker container.</p>

</body>

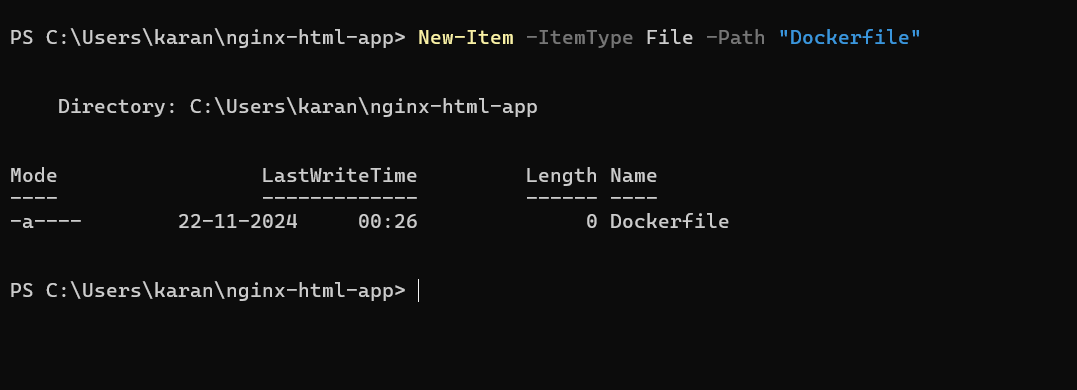
</html>

****

**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

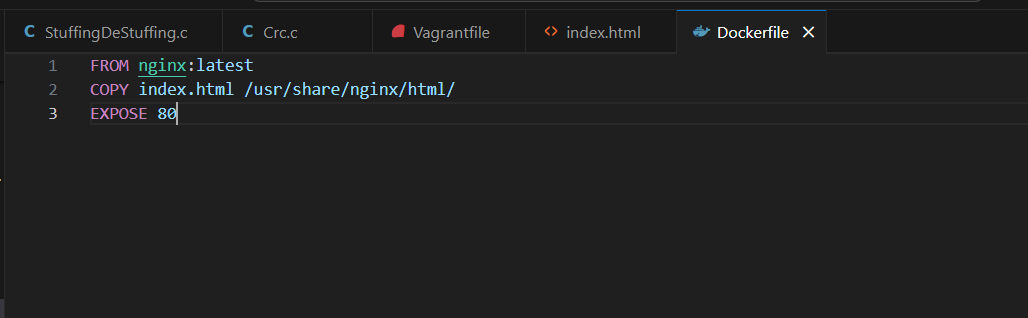


Edit the Dockerfile and add the following content:

FROM nginx:latest

COPY index.html /usr/share/nginx/html/

EXPOSE 80

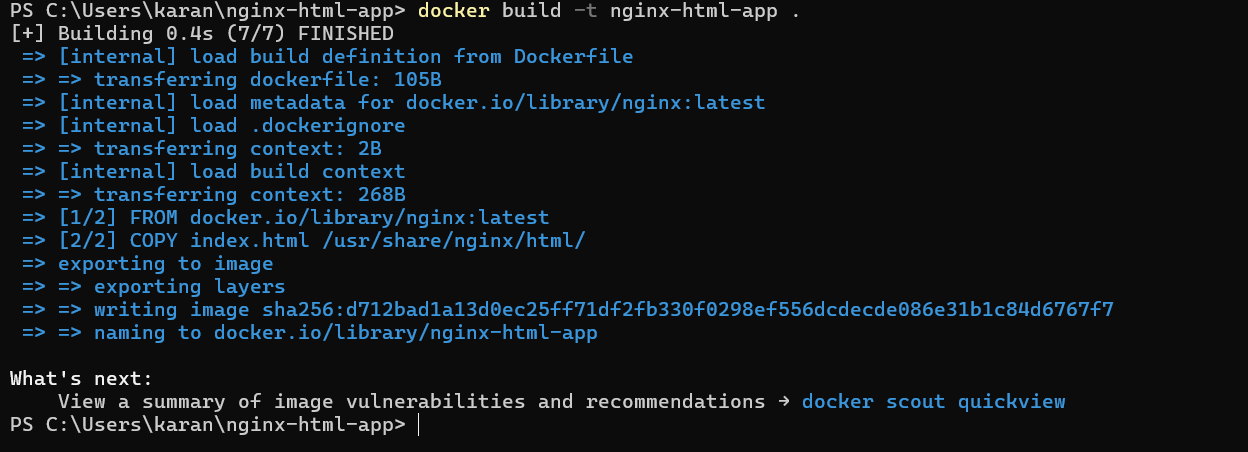
****

**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 .

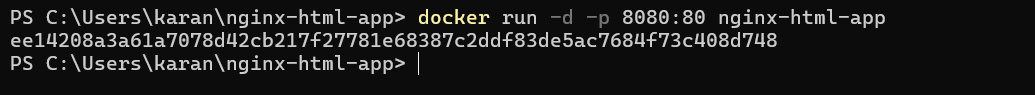


Docker will use the Nginx base image, copy your index.html into the appropriate directory, and build the image.

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

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

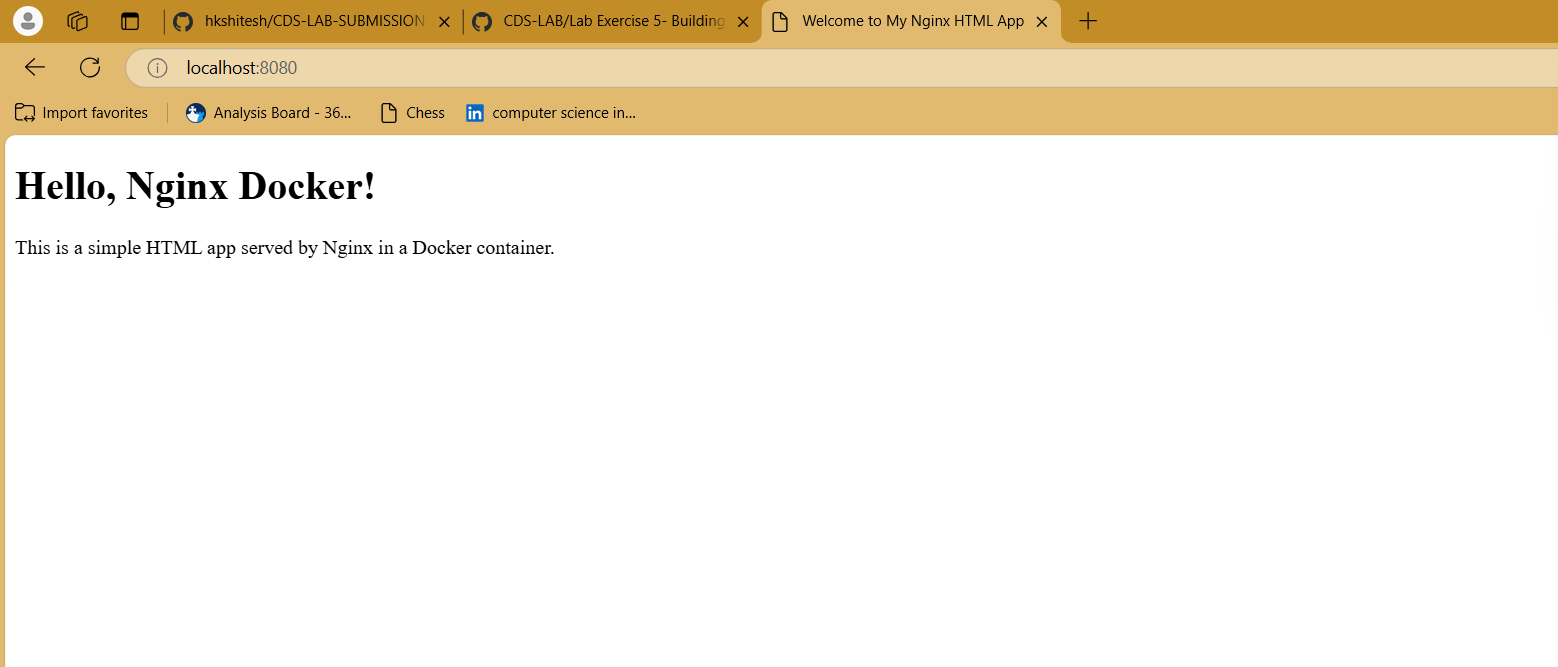
docker run -d -p 8080:80 nginx-html-app



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!”.



**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

docker stop <container-id>

docker rm <container-id>

