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
```

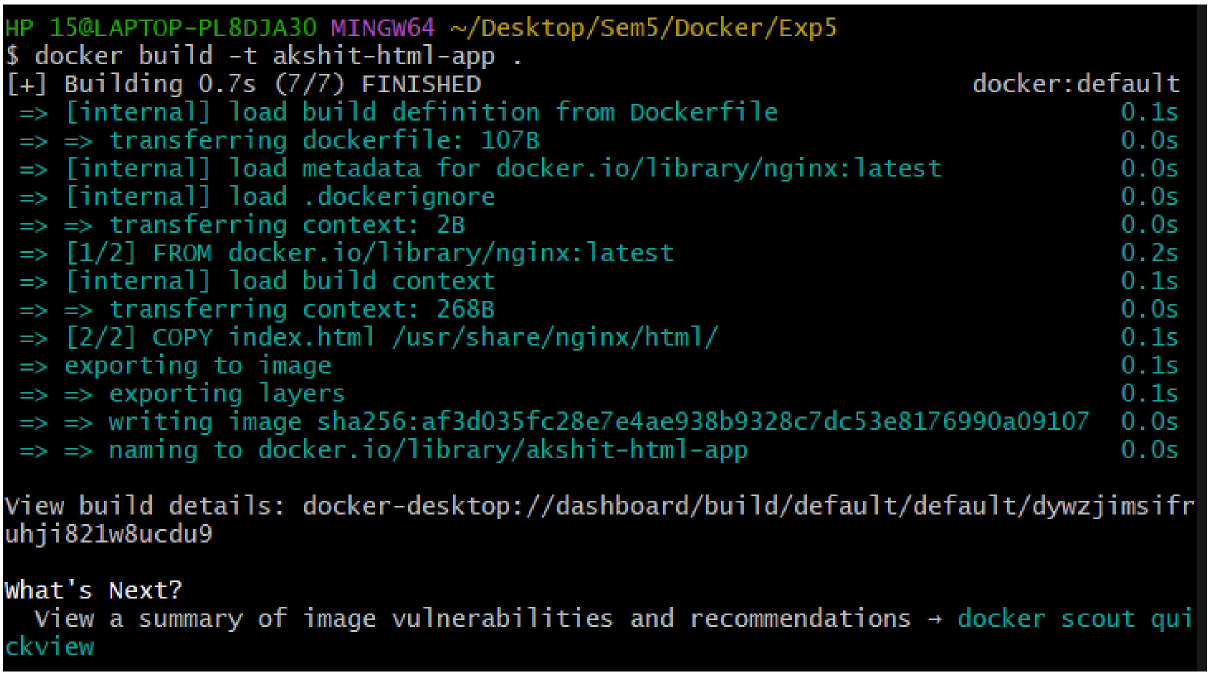
Edit the Dockerfile and add the following content:

```
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 .
```



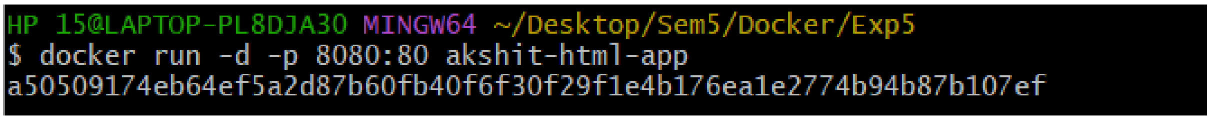
```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp5
$ docker build -t akshit-html-app .
[+] Building 0.7s (7/7) FINISHED                                   docker:default
 => [internal] load build definition from Dockerfile                         0.1s
 => => transferring dockerfile: 107B                                         0.0s
 => [internal] load metadata for docker.io/library/nginx:latest              0.0s
 => [internal] load .dockerignore                                            0.0s
 => => transferring context: 2B                                              0.0s
 => [1/2] FROM docker.io/library/nginx:latest                                0.2s
 => [internal] load build context                                            0.1s
 => => transferring context: 268B                                            0.0s
 => [2/2] COPY index.html /usr/share/nginx/html/                             0.1s
 => exporting to image                                                       0.1s
 => => exporting layers                                                      0.1s
 => => writing image sha256:af3d035fc28e7e4ae938b9328c7dc53e8176990a09107  0.0s
 => => naming to docker.io/library/akshit-html-app                           0.0s

View build details: docker-desktop://dashboard/build/default/default/dywzjimsifr
uhji821w8ucdu9

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout qui
ckview
```

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
```

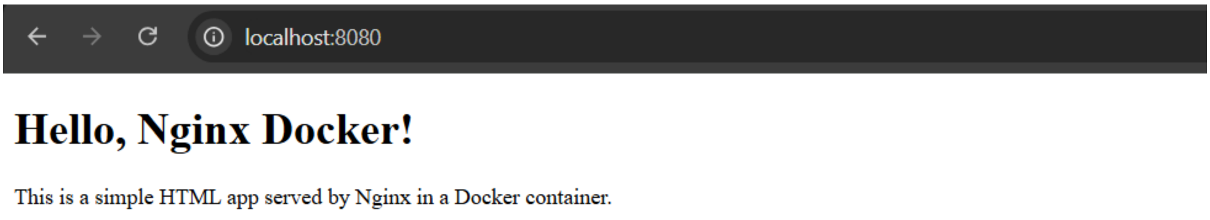


```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp5
$ docker run -d -p 8080:80 akshit-html-app
a50509174eb64ef5a2d87b60fb40f6f30f29f1e4b176ea1e2774b94b87b107ef
```

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!”.



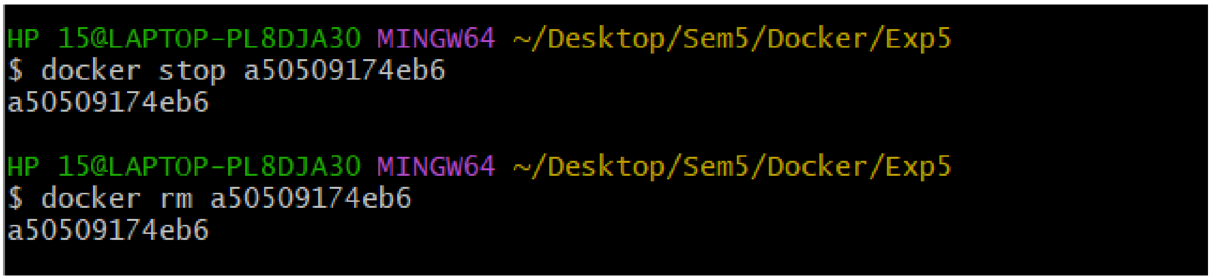
Hello, Nginx Docker!

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

docker stop <container-id>

docker rm <container-id>
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp5
$ docker stop a50509174eb6
a50509174eb6

HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp5
$ docker rm a50509174eb6
a50509174eb6
```