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

Name: Aditya Tomar

Sap: 500106015

R.no: R2142221060

Batch: 2(DevOps)

1. Setup

You will need:

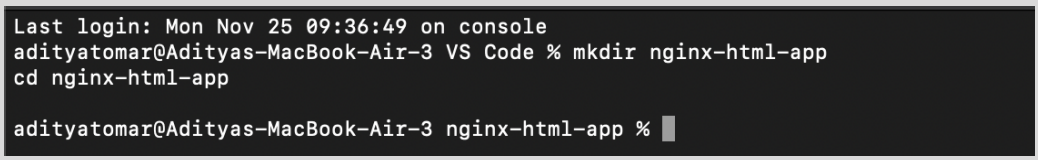
- Docker installed on your machine.
- A simple HTML file for the app.

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
```



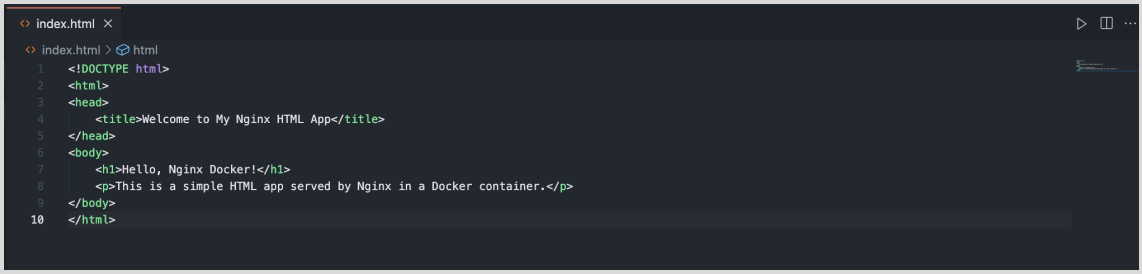
Inside the nginx-html-app directory, create the HTML file.

```
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

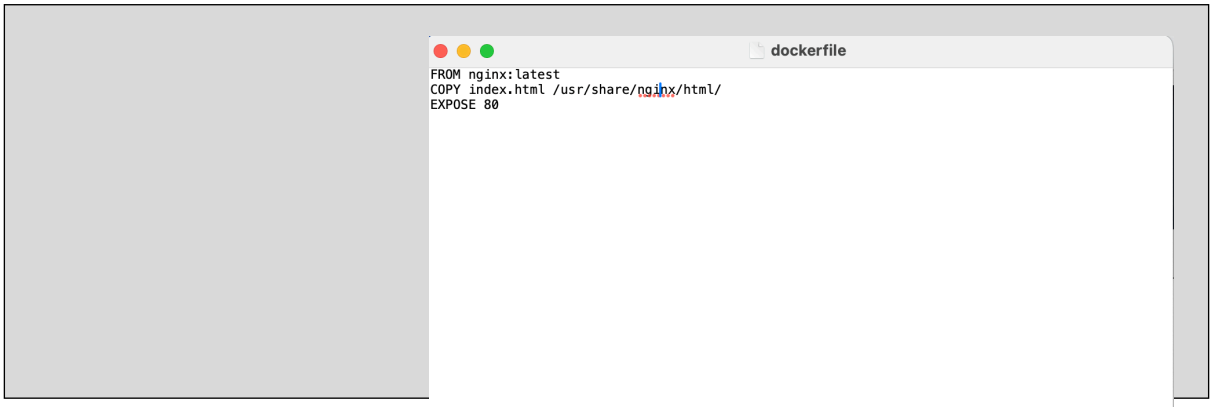
In the same directory, create a Dockerfile. This file will define how to build the Docker image using Nginx as the base image.

```
touch Dockerfile
```

```
adityatomar@Adityas-MacBook-Air-3 nginx-html-app % touch dockerfile
adityatomar@Adityas-MacBook-Air-3 nginx-html-app % open 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 .
```

```
adityatomar@Adityas-MacBook-Air-3 nginx-html-app % docker build -t ngnix-html-app .
[+] Building 2.6s (8/8) FINISHED                                                                                    docker:desktop-linux
 => [internal] load build definition from dockerfile                                                                                0.0s
 => => transferring dockerfile: 377B                                                                                                0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                                                     0.0s
 => [internal] load .dockerignore                                                                                                   0.0s
 => => transferring context: 2B                                                                                                     0.0s
 => [internal] load build context                                                                                                   0.0s
 => => transferring context: 257B                                                                                                   0.0s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:28402db69fec7c17e179ea87882667f1e054391138f77ffaf0c3eb388efc3ffb               2.5s
 => => resolve docker.io/library/nginx:latest@sha256:28402db69fec7c17e179ea87882667f1e054391138f77ffaf0c3eb388efc3ffb               2.4s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                                        0.0s
 => [2/2] COPY index.html /usr/share/nginx/html/                                                                                    0.0s
 => exporting to image                                                                                                              0.1s
 => => exporting layers                                                                                                             0.0s
 => => exporting manifest sha256:912a4361ccd6916fdcbce53ff79837cefe9b227ddd2786ba50173943c1ea4270                                   0.0s
 => => exporting config sha256:a65842434d886d46bff5899c3d87dad54d7980e8feb2cb0444dd17d450a784dc                                     0.0s
 => => exporting attestation manifest sha256:0d0361ecef6399d41c7a2e9060c8c5b58c3164b89bcffde8eb6437f0edad0595                       0.0s
 => => exporting manifest list sha256:3233ca131a6040b5d63dc39d70bc4ba0893c9b39e931916f38b3ac8e8347534d                              0.0s
 => => naming to docker.io/library/ngnix-html-app:latest                                                                            0.0s
 => => unpacking to docker.io/library/ngnix-html-app:latest                                                                         0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/qbsrtkma6k7axt8465rspd59a

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
adityatomar@Adityas-MacBook-Air-3 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:

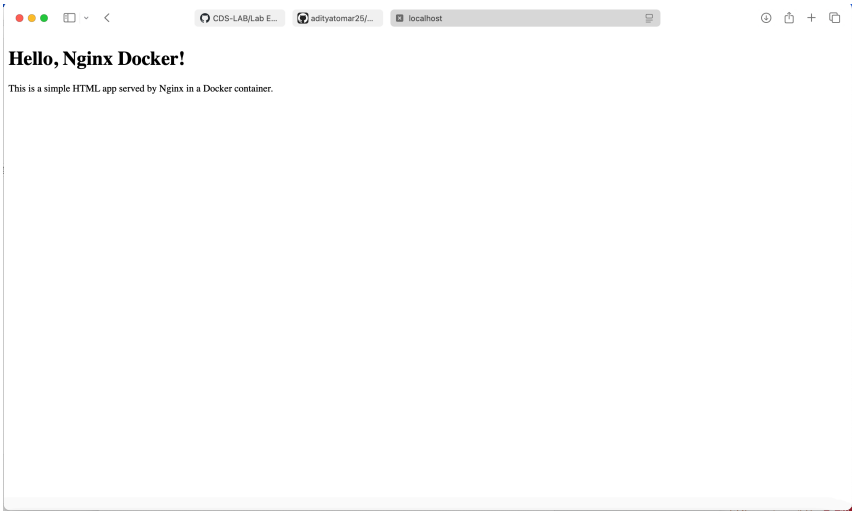
```
docker run -d -p 8080:80 nginx-html-app
```

```
adityatomar@Adityas-MacBook-Air-3 nginx-html-app % docker run -d -p 8080:80 ngnix-html-app
6c0445629b7fa2ef1e34858ffbde4c394897b54f8928b2d3b15d18cd78addc5a
```

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>
```

```
[adityatomar@Adityas-MacBook-Air-3 nginx-html-app % docker stop 6c0445629b7f
6c0445629b7f
[adityatomar@Adityas-MacBook-Air-3 nginx-html-app % docker rm 6c0445629b7f
6c0445629b7f
[adityatomar@Adityas-MacBook-Air-3 nginx-html-app % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
adityatomar@Adityas-MacBook-Air-3 nginx-html-app %
```