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
```

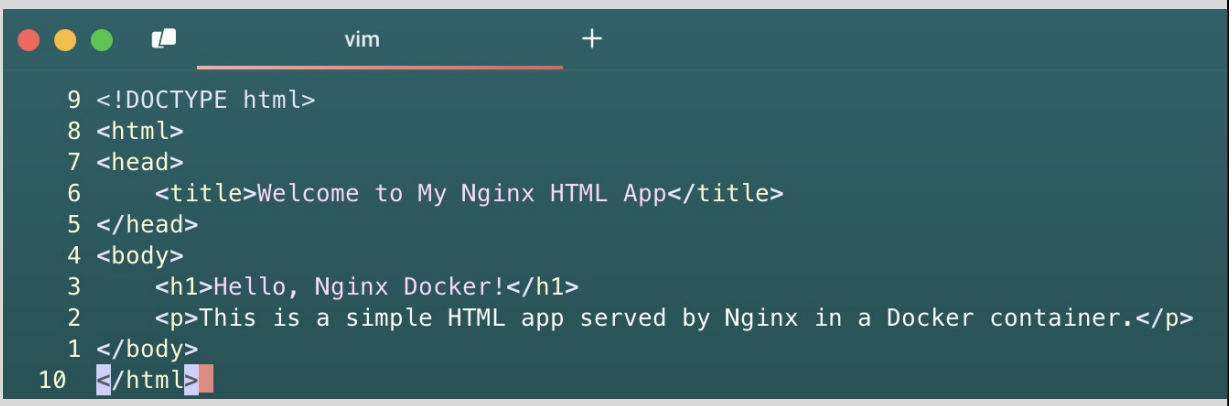
Inside the nginx-html-app directory, create the HTML file.

```
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>
```



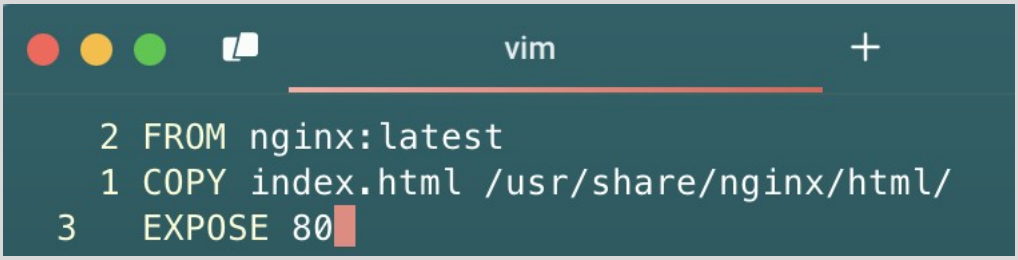
3. Step 2: Create a Dockerfile

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 .
```

```
~/nginx-html-app
❯ docker build -t nginx-html-app .

[+] Building 0.1s (7/7) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 104B
 => [internal] load metadata for docker.io/library/nginx:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build context
 => => transferring context: 258B
 => [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:f5771badcbb618c7faf2174f73d3db10d1a397a8ddecfa44ca774187b8b65d72
 => => naming to docker.io/library/nginx-html-app

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/atz81d9x2l4tso6h172uu1wgj

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
```

```
~/nginx-html-app
❯ docker run -d -p 8080:80 nginx-html-app
b91f5c06c76e62d6a13d8eceedcb1902263db6fd9a67f58b0db8493c64dedbe8
```

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.

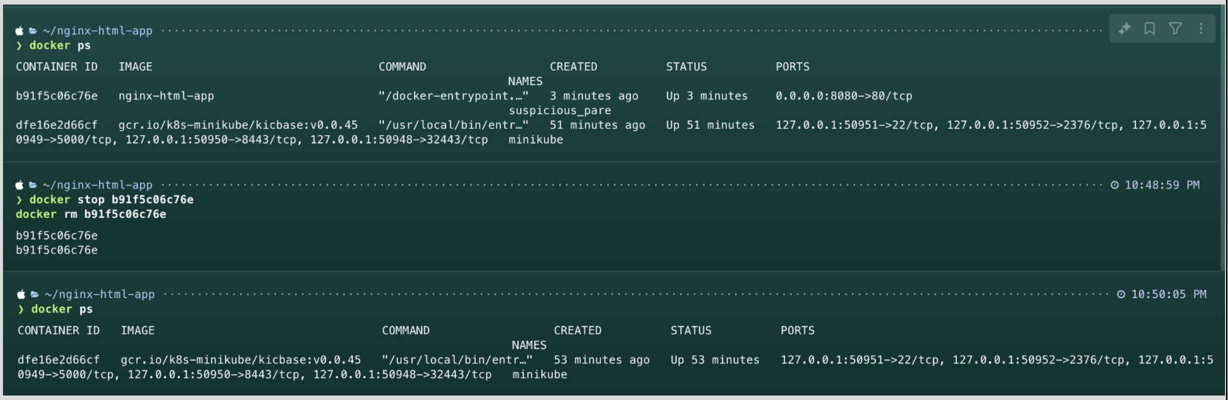
7. Step 6: Stop and Remove the Container

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>
```



```
 ~/nginx-html-app ·····································································
❯ docker ps
CONTAINER ID   IMAGE                                 COMMAND                   CREATED          STATUS          PORTS
                                                                             NAMES
b91f5c06c76e   nginx-html-app                        "/docker-entrypoint.…"   3 minutes ago    Up 3 minutes    0.0.0.0:8080->80/tcp
                                                                             suspicious_pare
dfe16e2d66cf   gcr.io/k8s-minikube/kicbase:v0.0.45   "/usr/local/bin/entr…"   51 minutes ago   Up 51 minutes   127.0.0.1:50951->22/tcp, 127.0.0.1:50952->2376/tcp, 127.0.0.1:5
0949->5000/tcp, 127.0.0.1:50950->8443/tcp, 127.0.0.1:50948->32443/tcp   minikube

 ~/nginx-html-app ····································································· 10:48:59 PM
❯ docker stop b91f5c06c76e
docker rm b91f5c06c76e
b91f5c06c76e
b91f5c06c76e

 ~/nginx-html-app ····································································· 10:50:05 PM
❯ docker ps
CONTAINER ID   IMAGE                                 COMMAND                   CREATED          STATUS          PORTS
                                                                             NAMES
dfe16e2d66cf   gcr.io/k8s-minikube/kicbase:v0.0.45   "/usr/local/bin/entr…"   53 minutes ago   Up 53 minutes   127.0.0.1:50951->22/tcp, 127.0.0.1:50952->2376/tcp, 127.0.0.1:5
0949->5000/tcp, 127.0.0.1:50950->8443/tcp, 127.0.0.1:50948->32443/tcp   minikube
```