

Lab Exercise 4- 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 .
```

```
[mohdanas@Mohds-MacBook-Air nginx-html-app % docker build -t nginx-html-app .]
[+] Building 0.2s (7/7) FINISHED
  => [internal] load build definition from Dockerfile                               docker:desktop-linux
  => => transferring dockerfile: 104B                                         0.0s
  => [internal] load metadata for docker.io/library/nginx:latest                 0.1s
  => [internal] load .dockerignore                                              0.0s
  => => transferring context: 2B                                              0.0s
  => [internal] load build context                                              0.0s
  => => transferring context: 259B                                             0.0s
  => [1/2] FROM docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457 0.0s
  => => resolve docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457 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:b0e496a60411b836d11cd7689a447a99c199b4a11a128f071f482b87b86a31 0.0s
  => => exporting config sha256:46c5fa6a457e7a9c8afc8e3e0514b9f7ebc0ac675358b9e5beb699b530a9cf 0.0s
  => => exporting attestation manifest sha256:959ae455e9e32a209cfa853f0d692fe9807e90f9c6b121fa15 0.0s
  => => exporting manifest list sha256:dd441558eb7ed10dc41922d73dbe9a528d9faa21cb9445a487435097 0.0s
  => => naming to docker.io/library/nginx-html-app:latest                      0.0s
  => => unpacking to docker.io/library/nginx-html-app:latest                   0.0s
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/x0t6j5qy5vpxe07fr7380
m0ak
```

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 8006:80 nginx-html-app
```

```
[mohdanas@Mohds-MacBook-Air nginx-html-app % docker run -d -p 8006:80 nginx-html-app
4470dbbba17d6ce7672f46a2e4e8573fd78961e388c45bbc4d34e07f3c37cb1e
```

This command runs the container in detached mode (-d) and maps port 8006 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:8006>. 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>
```

```
[mohdanas@Mohds-MacBook-Air nginx-html-app % docker ps
CONTAINER ID   IMAGE       COMMAND           CREATED          STATUS          PORTS
NAMES
4470dbbba17d   nginx-html-app   "/docker-entrypoint...."  58 seconds ago   Up 58 seconds   0.0.0.0:8006
->80/tcp, [::]:8006->80/tcp   quirky_hypatia
[mohdanas@Mohds-MacBook-Air nginx-html-app % docker stop 4470dbbba17d
4470dbbba17d
[mohdanas@Mohds-MacBook-Air nginx-html-app % docker rm 4470dbbba17d
4470dbbba17d
```