Lab Exercise 21- 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:

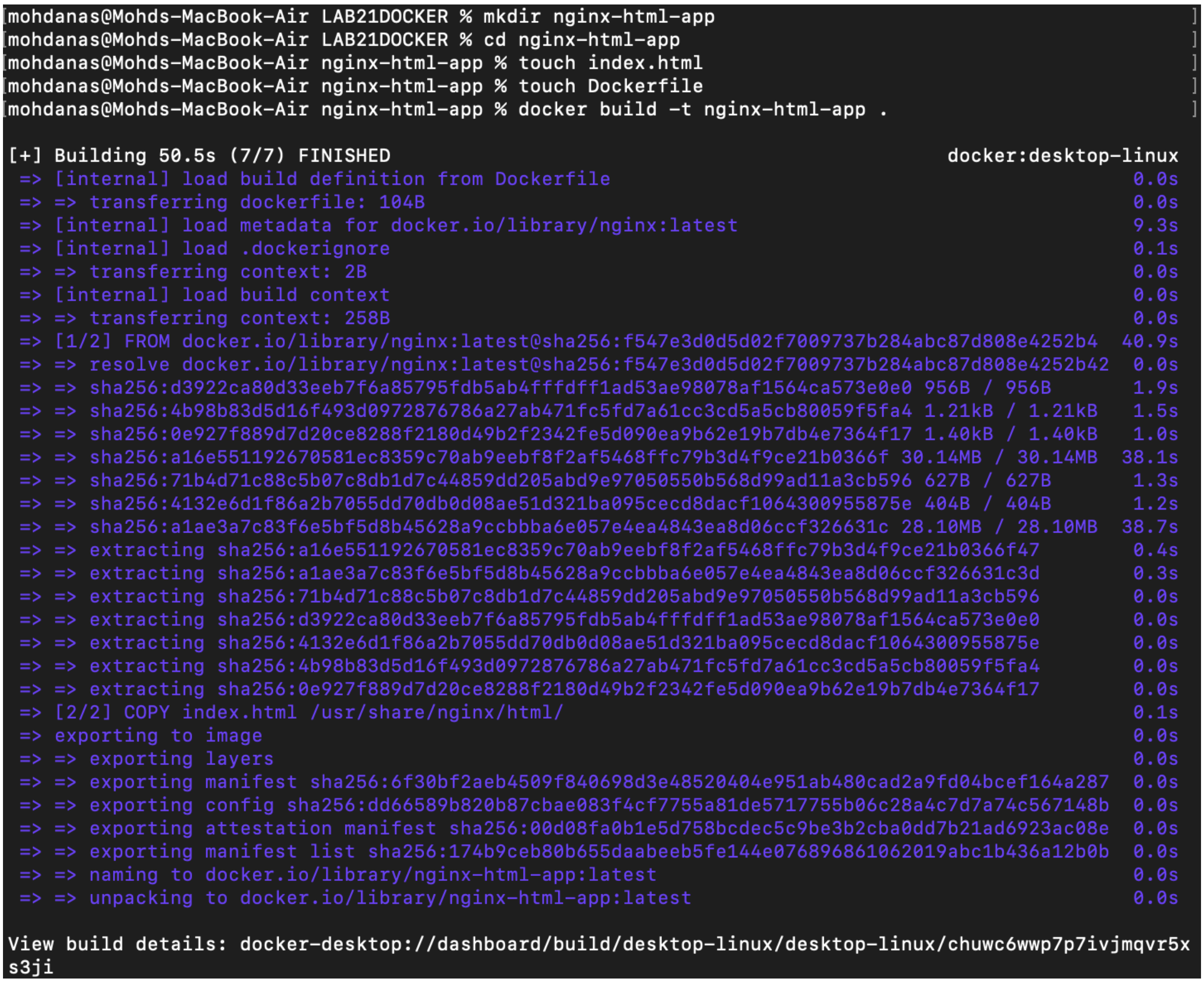
```
FROM nginx:latest

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.



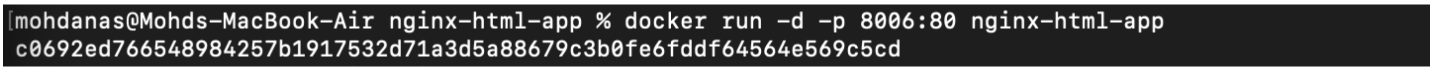
```
[mohdanas@Mohds-MacBook-Air LAB21DOCKER % mkdir nginx-html-app
[mohdanas@Mohds-MacBook-Air LAB21DOCKER % cd nginx-html-app
[mohdanas@Mohds-MacBook-Air nginx-html-app % touch index.html
[mohdanas@Mohds-MacBook-Air nginx-html-app % touch Dockerfile
[mohdanas@Mohds-MacBook-Air nginx-html-app % docker build -t nginx-html-app .

[+] Building 50.5s (7/7) FINISHED                                                docker:desktop-linux
 => [internal] load build definition from Dockerfile                                             0.0s
 => => transferring dockerfile: 104B                                                             0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                  9.3s
 => [internal] load .dockerignore                                                                0.1s
 => => transferring context: 2B                                                                  0.0s
 => [internal] load build context                                                                0.0s
 => => transferring context: 258B                                                                0.0s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:f547e3d0d5d02f7009737b284abc87d808e4252b4  40.9s
 => => resolve docker.io/library/nginx:latest@sha256:f547e3d0d5d02f7009737b284abc87d808e4252b42  0.0s
 => => sha256:d3922ca80d33eeb7f6a85795fdb5ab4fffdff1ad53ae98078af1564ca573e0e0 956B / 956B       1.9s
 => => sha256:4b98b83d5d16f493d0972876786a27ab471fc5fd7a61cc3cd5a5cb80059f5fa4 1.21kB / 1.21kB   1.5s
 => => sha256:0e927f889d7d20ce8288f2180d49b2f2342fe5d090ea9b62e19b7db4e7364f17 1.40kB / 1.40kB   1.0s
 => => sha256:a16e551192670581ec8359c70ab9eebf8f2af5468ffc79b3d4f9ce21b0366f 30.14MB / 30.14MB  38.1s
 => => sha256:71b4d71c88c5b07c8db1d7c44859dd205abd9e97050550b568d99ad11a3cb596 627B / 627B       1.3s
 => => sha256:4132e6d1f86a2b7055dd70db0d08ae51d321ba095cecd8dacf1064300955875e 404B / 404B       1.2s
 => => sha256:a1ae3a7c83f6e5bf5d8b45628a9ccbbba6e057e4ea4843ea8d06ccf326631c 28.10MB / 28.10MB  38.7s
 => => extracting sha256:a16e551192670581ec8359c70ab9eebf8f2af5468ffc79b3d4f9ce21b0366f47        0.4s
 => => extracting sha256:a1ae3a7c83f6e5bf5d8b45628a9ccbbba6e057e4ea4843ea8d06ccf326631c3d        0.3s
 => => extracting sha256:71b4d71c88c5b07c8db1d7c44859dd205abd9e97050550b568d99ad11a3cb596        0.0s
 => => extracting sha256:d3922ca80d33eeb7f6a85795fdb5ab4fffdff1ad53ae98078af1564ca573e0e0        0.0s
 => => extracting sha256:4132e6d1f86a2b7055dd70db0d08ae51d321ba095cecd8dacf1064300955875e        0.0s
 => => extracting sha256:4b98b83d5d16f493d0972876786a27ab471fc5fd7a61cc3cd5a5cb80059f5fa4        0.0s
 => => extracting sha256:0e927f889d7d20ce8288f2180d49b2f2342fe5d090ea9b62e19b7db4e7364f17        0.0s
 => [2/2] COPY index.html /usr/share/nginx/html/                                                 0.1s
 => exporting to image                                                                           0.0s
 => => exporting layers                                                                          0.0s
 => => exporting manifest sha256:6f30bf2aeb4509f840698d3e48520404e951ab480cad2a9fd04bcef164a287  0.0s
 => => exporting config sha256:dd66589b820b87cbae083f4cf7755a81de5717755b06c28a4c7d7a74c567148b  0.0s
 => => exporting attestation manifest sha256:00d08fa0b1e5d758bcdec5c9be3b2cba0dd7b21ad6923ac08e  0.0s
 => => exporting manifest list sha256:174b9ceb80b655daabeeb5fe144e076896861062019abc1b436a12b0b  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/chuwc6wwp7p7ivjmqvr5x
s3ji
```

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
```

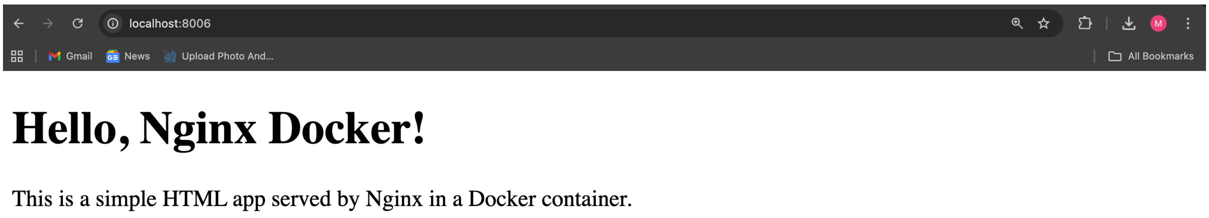
This command runs the container in detached mode (-d) and maps port 8006 on your host machine to port 80 inside the container, where Nginx is serving your HTML app.



```
[mohdanas@Mohds-MacBook-Air nginx-html-app % docker run -d -p 8006:80 nginx-html-app
c0692ed766548984257b1917532d71a3d5a88679c3b0fe6fddf64564e569c5cd
```

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!”.



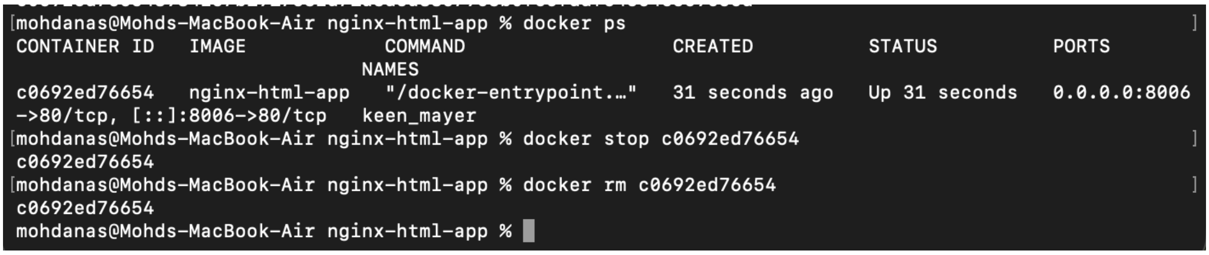
Hello, Nginx Docker!

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
 c0692ed76654   nginx-html-app   "/docker-entrypoint.…"   31 seconds ago   Up 31 seconds   0.0.0.0:8006
 ->80/tcp, [::]:8006->80/tcp   keen_mayer
[mohdanas@Mohds-MacBook-Air nginx-html-app % docker stop c0692ed76654                                     ]
 c0692ed76654
[mohdanas@Mohds-MacBook-Air nginx-html-app % docker rm c0692ed76654                                       ]
 c0692ed76654
 mohdanas@Mohds-MacBook-Air nginx-html-app % 
```