

# **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 .
```

```
C:\Users\shwet\nginx-html-app>docker build -t nginx-html-app .
[+] Building 21.3s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 107B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 268B
=> [1/2] FROM docker.io/library/nginx:latest@sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c 16.0s
=> => resolve docker.io/library/nginx:latest@sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c5 0.1s
=> => sha256:de57a609c9d5148f10b38f5c920d276e9e38b2856fe16c0aae1450613dc12051 1.40kB / 1.40kB 0.7s
=> => sha256:192e2451f8751fb74549c932e26a9bcfd7b669fe2f5bd8381ea5ac65f09b256b 1.21kB / 1.21kB 1.0s
=> => sha256:77fa2eb0631772679b0e48eca04f4906fba5fe9437fe01618873a4a1171107ce 404B / 404B 1.3s
=> => sha256:53d743880af45ad9f9f141eec1fe3a413087e528075a5d8884d6215ddfd2b806 954B / 954B 1.3s
=> => sha256:108ab82928207dabd9abffdbc960dd842364037563fc560b8f6304e4a91454fe 628B / 628B 0.8s
=> => sha256:b5feb73171bf1bcf29fdd1ba642c3d30cdf4c6329b19d89be14d209d778c89ba 29.97MB / 29.97MB 9.3s
=> => sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db 29.78MB / 29.78MB 12.2s
=> => extracting sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db 1.1s
=> => extracting sha256:b5feb73171bf1bcf29fdd1ba642c3d30cdf4c6329b19d89be14d209d778c89ba 0.8s
=> => extracting sha256:108ab82928207dabd9abffdbc960dd842364037563fc560b8f6304e4a91454fe 0.0s
=> => extracting sha256:77fa2eb0631772679b0e48eca04f4906fba5fe94377e01618873a4a1171107ce 0.0s
=> => extracting sha256:192e2451f8751fb74549c932e26a9bcfd7b669fe2f5bd8381ea5ac65f09b256b 0.0s
=> => extracting sha256:de57a609c9d5148f10b38f5c920d276e9e38b2856fe16c0aae1450613dc12051 0.0s
=> [2/2] COPY index.html /usr/share/nginx/html/
=> exporting to image
=> => exporting layers
0.3s
0.3s
0.1s
```

ChatGPT can make mistakes. Check important info. See [Cookie Preferences](#).

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
```

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
```

```
C:\Users\shwet\nginx-html-app>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
2a650f780157 nginx-html-app "/docker-entrypoint..." 5 minutes ago Up 5 minutes 0.0.0.0:8006->80/tcp, [::]:8006->80/tcp nginx-demo
```

```
docker stop 2a650f780157
```

```
C:\Users\shwet\nginx-html-app>docker stop 2a650f780157
2a650f780157
```

```
docker rm 2a650f780157
```

```
C:\Users\shwet\nginx-html-app>
C:\Users\shwet\nginx-html-app>docker rm 2a650f780157
2a650f780157
```