

## Lab Exercise 21

# Building a Docker Image for an HTML App Using Nginx

Name: Ayush Bhardwaj

Sap id: 500124917

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.

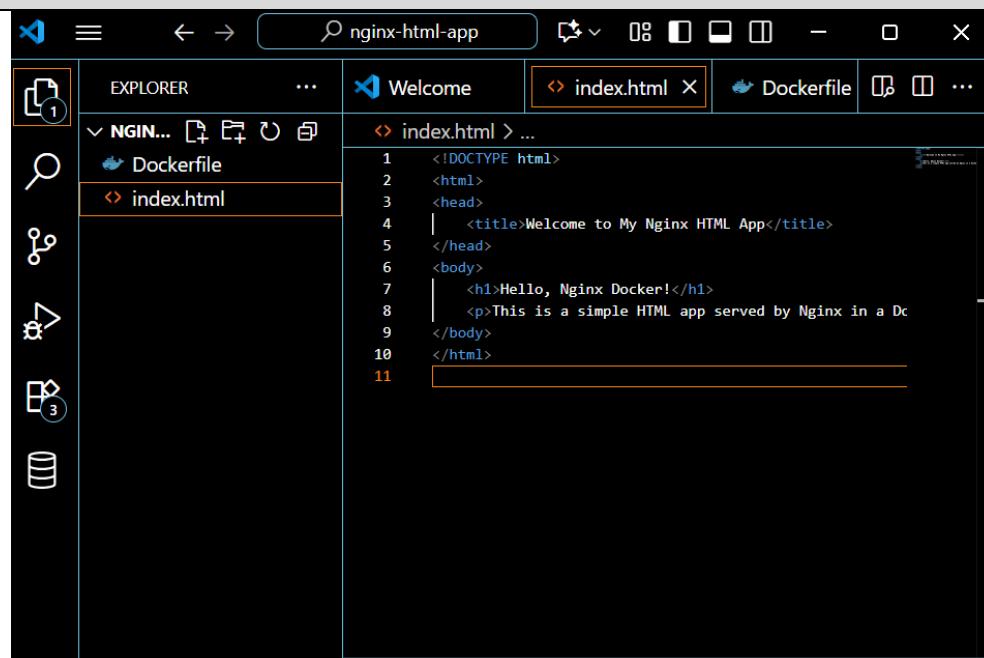
```
touch Dockerfile
```

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 .
```

```
PS C:\Users\ASUS> cd C:\Users\ASUS\nginx-html-app
PS C:\Users\ASUS\nginx-html-app> docker build -t nginx-html-app .
[+] Building 22.8s (7/7) FINISHED          docker:desktop-linux
=> [internal] load build definition from Dockerfile      0.1s
=> => transferring dockerfile: 223B                      0.1s
=> [internal] load metadata for docker.io/library/nginx: 5.1s
=> [internal] load .dockerignore                   0.1s
=> => transferring context: 2B                      0.0s
=> [1/2] FROM docker.io/library/nginx:latest@sha256:1be 16.8s
=> => resolve docker.io/library/nginx:latest@sha256:1bee 0.1s
=> => sha256:9def903993e4ef9a3faa02bb893b038 404B / 404B 0.8s
=> => sha256:d921c57c6a81addac6ca451906699ca 956B / 956B 1.2s
=> => sha256:52bc359bcd74bb3d11b94cf3c6 1.21kB / 1.21kB 1.5s
=> => sha256:266626526d42cf7fe5f56b93 29.97MB / 29.97MB 12.0s
=> => sha256:320b0949be89766f7c6a8746f197102 628B / 628B 0.8s
=> => sha256:e2f8e296d9df1dd5e2ddc81e5e7 1.40kB / 1.40kB 0.7s
=> => sha256:d7ecded7702a5dbf6d0f79a7 29.78MB / 29.78MB 12.8s
=> => extracting sha256:d7ecded7702a5dbf6d0f79a71edc34b5 1.1s
=> => extracting sha256:266626526d42cf7fe5f56b933db3f4c5 0.9s
=> => extracting sha256:320b0949be89766f7c6a8746f1971021 0.0s
=> => extracting sha256:d921c57c6a81addac6ca451906699ca6 0.0s
=> => extracting sha256:9def903993e4ef9a3faa02bb893b0382 0.0s
=> => extracting sha256:52bc359bcd74bb3d11b94cf3c6d94bc 0.0s
=> => extracting sha256:e2f8e296d9df1dd5e2ddc81e5e758f97 0.0s
=> [internal] load build context           0.1s
=> => transferring context: 268B        0.1s
=> [2/2] COPY index.html /usr/share/nginx/html/    0.2s
=> exporting to image                    0.3s
=> => exporting layers                  0.1s
=> => exporting manifest sha256:7bb2caac20a984b5f08ac463 0.0s
```

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.

```
=> => unpacking to docker.io/library/nginx-html-app:late 0.0s
PS C:\Users\ASUS\nginx-html-app> docker run -d --name my-nginx
-p 8006:80 nginx-html-app
deb432f3ff339d7577d27d0ad0fa99987c73ef65f2501c6b8443352dc74c296c
PS C:\Users\ASUS\nginx-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>
```

```
PS C:\Users\ASUS\nginx-html-app> docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
deb432f3ff33      nginx-html-app    "/docker-entrypoint..."   4 minutes ago     Up 4 minutes       0.0.0.0:8006->80/tcp, [::]:8006->80/tcp
my-nginx
PS C:\Users\ASUS\nginx-html-app> docker stop my-nginx
my-nginx
PS C:\Users\ASUS\nginx-html-app> docker rm my-nginx
my-nginx
PS C:\Users\ASUS\nginx-html-app>
```