# 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

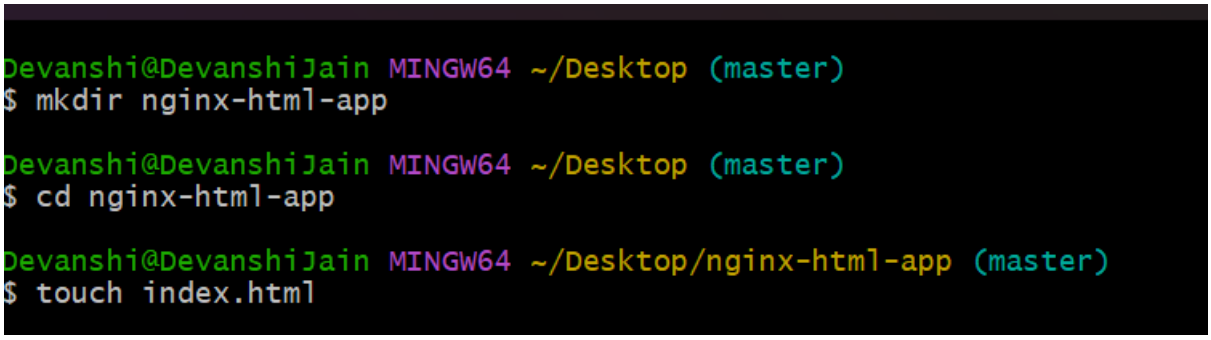
Create a directory for your HTML app and place an index.html file in it.

```
mkdir nginx-html-app

cd nginx-html-app
```

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

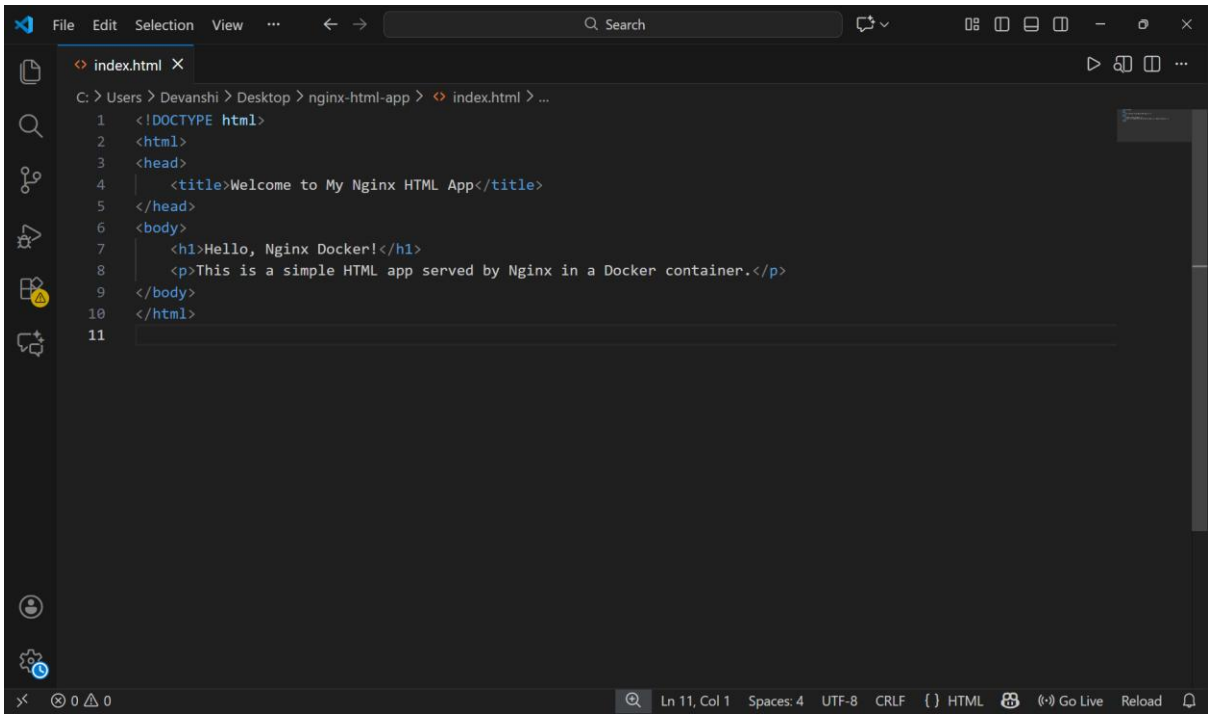
```
touch index.html
```




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 .

```
Devanshi@DevanshiJain MINGW64 ~/Desktop/nginx-html-app (master)
$ docker build -t nginx-html-app .
[+] Building 0.9s (7/7) FINISHED                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                      0.0s
 => => transferring dockerfile: 107B                                      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: 268B                                         0.0s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1  0.2s
 => => resolve docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1  0.0s
 => [2/2] COPY index.html /usr/share/nginx/html/                          0.0s
 => exporting to image                                                    0.3s
 => => exporting layers                                                   0.1s
 => => exporting manifest sha256:b6749f3ad06f704a40479350ac063e8a409c6bbe  0.0s
 => => exporting config sha256:589e122d6d62d7135f66e315c85df585a52abc6b46  0.0s
 => => exporting attestation manifest sha256:7fbd3900311f1608ce03543eed02  0.0s
 => => exporting manifest list sha256:5e84d56148aa9ac15f3bce3e8ffb6f6e50f  0.0s
 => => naming to docker.io/library/nginx-html-app:latest                  0.0s
 => => unpacking to docker.io/library/nginx-html-app:latest               0.1s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux
/343x9w5u8tfhbk6zrspzosuwo
```

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:

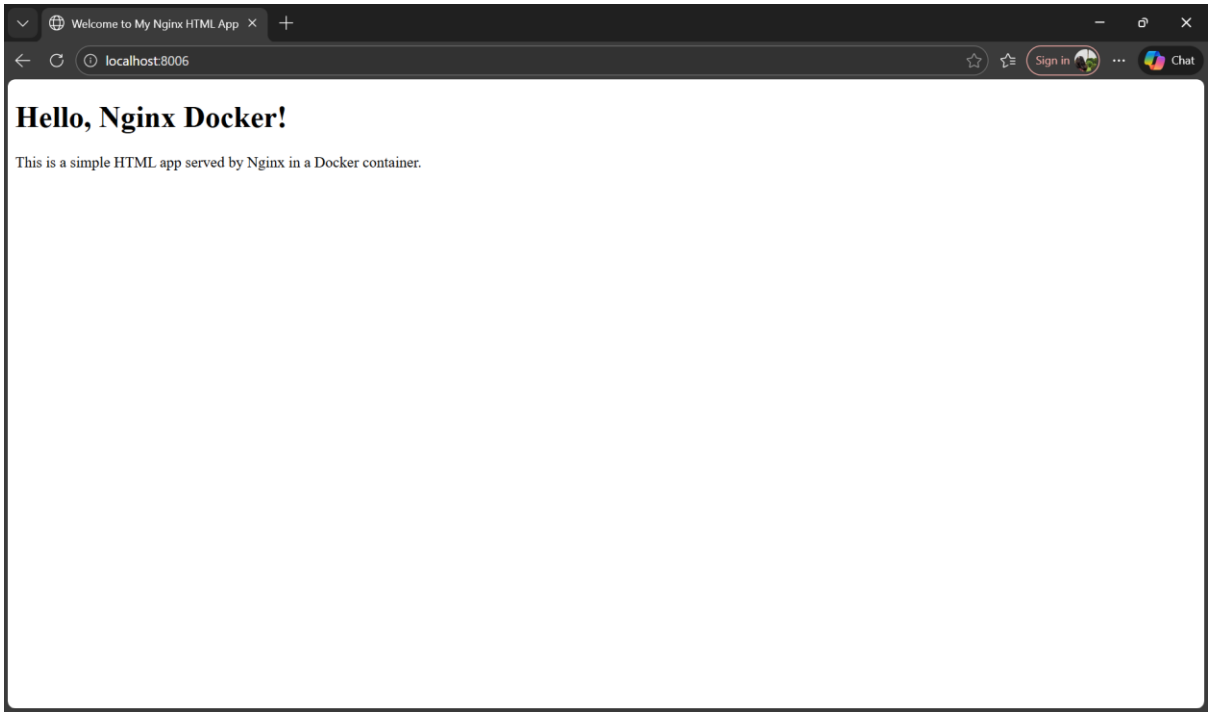
docker run -d -p 8006:80 nginx-html-app

```
Devanshi@DevanshiJain MINGW64 ~/Desktop/nginx-html-app (master)
$ docker run -d -p 8006:80 nginx-html-app
b70bb73dca2c57bfcd9e23a9bad1564cde2059ac660aa01643fb4581717282d9
```

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!”.




## 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>

```
Devanshi@DevanshiJain MINGW64 ~/Desktop/nginx-html-app (master)
$ docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED         STATUS
        PORTS                          NAMES
b70bb73dca2c   nginx-html-app   "/docker-entrypoint.…"   4 minutes ago   Up 4 mi
nutes   0.0.0.0:8006->80/tcp, [::]:8006->80/tcp   vibrant_shirley

Devanshi@DevanshiJain MINGW64 ~/Desktop/nginx-html-app (master)
$ docker stop b70bb73dca2c
b70bb73dca2c

Devanshi@DevanshiJain MINGW64 ~/Desktop/nginx-html-app (master)
$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

Devanshi@DevanshiJain MINGW64 ~/Desktop/nginx-html-app (master)
$ docker rm b70bb73dca2c
b70bb73dca2c

Devanshi@DevanshiJain MINGW64 ~/Desktop/nginx-html-app (master)
$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED        STATUS                     PORTS     NAMES
268bcc1cf464   hello-world   "/hello"   2 months ago   Exited (0) 2 months ago              nervous_ride
```