

## 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
```

```
PS C:\Users\namit> cd D:\
PS D:\> cd D:\folder
PS D:\folder> mkdir nginx-html-app

Directory: D:\folder


Mode                LastWriteTime         Length Name
----                -
d-----          09-02-2026  12:47 AM                nginx-html-app

PS D:\folder> cd nginx-html-app
PS D:\folder\nginx-html-app>
```

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

```
touch index.html
```

```
PS D:\folder\nginx-html-app> New-Item index.html
```

```
Directory: D:\folder\nginx-html-app
```

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	09-02-2026 12:50 AM	0	index.html

```
PS D:\folder\nginx-html-app>
```

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>
```

```
PS D:\folder\nginx-html-app> notepad index.html
PS D:\folder\nginx-html-app>
```

```
File Edit View

<!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
```

```
PS D:\folder\nginx-html-app> New-Item Dockerfile

Directory: D:\folder\nginx-html-app

Mode                LastWriteTime         Length Name
----                -
-a-----          09-02-2026  12:53 AM             0 Dockerfile

PS D:\folder\nginx-html-app>
```

Edit the Dockerfile and add the following content:

```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```

```
PS D:\folder\nginx-html-app> notepad Dockerfile
PS D:\folder\nginx-html-app>
```

File Edit View

```
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.

```
PS D:\folder\nginx-html-app> docker build -t nginx-html-app .
[+] Building 1.5s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 107B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 268B
=> [1/2] FROM docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf4aec
=> => resolve docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf4aec
=> [2/2] COPY index.html /usr/share/nginx/html/
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:5aa8ff56a96e597c17c48acf361a9f61bd074929e565174854213d177914437a
=> => exporting config sha256:59259c00a65abd76bef61468009236bc53b8d7e2caea2a7de8e1f69cdeeb96af
=> => exporting attestation manifest sha256:65339ff3147b0c6a2e2a4474e55a79470467ac4bf070c30618f3b64bc2474634
=> => exporting manifest list sha256:0af0c443ce10a4e6dc5314b3da2bc948ff02afd85af64e6d1818334f644e180e
=> => naming to docker.io/library/nginx-html-app:latest
=> => unpacking to docker.io/library/nginx-html-app:latest
PS D:\folder\nginx-html-app>
```

#### 5. Step 4: Run the Docker Container

After building the image, you can run the container with the following command:

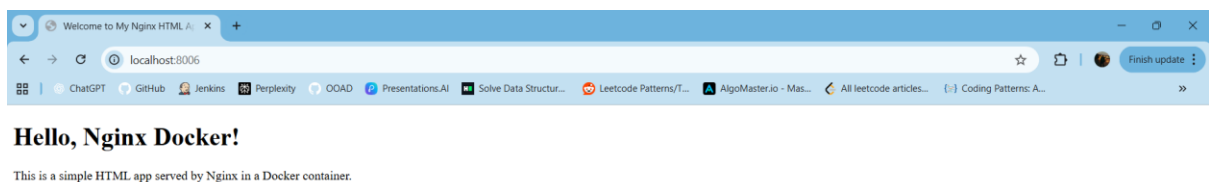
```
docker run -d -p 8006:80 nginx-html-app
```

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.

```
PS D:\folder\nnginx-html-app> docker run -d -p 8006:80 nginx-html-app
0f09c55db0c76d2d2a1e73906afa49c4453f66bf2dc72a1bb4f887bee3b8f924
PS D:\folder\nnginx-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>
```

```
PS D:\folder\nginx-html-app> docker ps # to see running containers
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
0f09c55db0c7   nginx-html-app "/docker-entrypoint..." 9 minutes ago  Up 9 minutes  0.0.0.0:8006->80/tcp, [::]:8006->80/tcp  quirky_feynman
PS D:\folder\nginx-html-app> docker stop 0f09c55db0c7
0f09c55db0c7
```

```
PS D:\folder\nginx-html-app> docker rm 0f09c55db0c7
0f09c55db0c7
PS D:\folder\nginx-html-app>
```