

Lab Exercise 4- Building a Docker Image for an HTML App Using Nginx

Name- Misha

SAP ID-500119679

Batch-2

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.

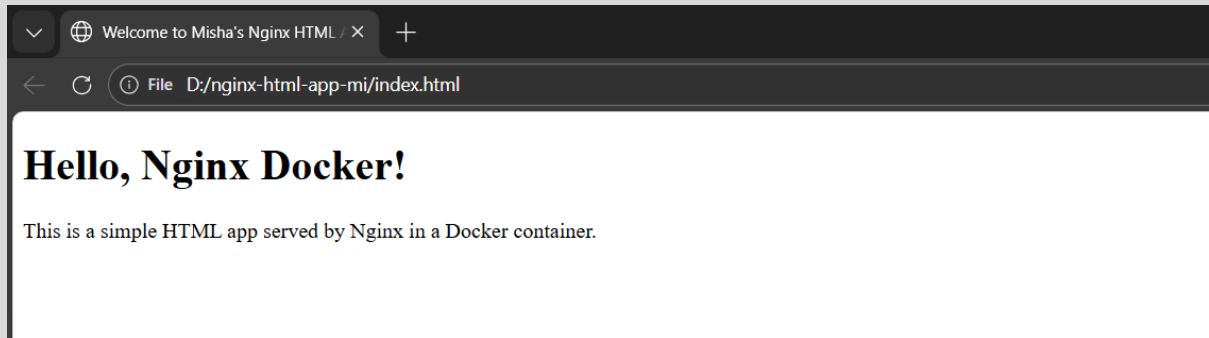
```
touch index.html
```

```
+ FullyQualifiedErrorId : CommandNotFoundException  
PS D:\nginx-html-app-mi> New-Item index.html  
  
Directory: D:\nginx-html-app-mi  
  
Mode                LastWriteTime         Length Name  
----                -  
-a-----         03-02-2026      11:20             0 index.html  
  
PS D:\nginx-html-app-mi> |
```

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 .
```

```
EXPOSE 80
PS D:\nginx-html-app-mi> docker build -t nginx-html-app-mi .
[+] Building 2.0s (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: 273B
=> [1/2] FROM docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf
=> => resolve docker.io/library/nginx:latest@sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf
=> [2/2] COPY index.html /usr/share/nginx/html/
=> => exporting to image
=> => exporting layers
=> => exporting manifest sha256:7a5126cc8d5febd8a6d2130d50b15de46ac32c19157c42238ed1d82954fac298
=> => exporting config sha256:505c0ba6bcb82e6a8794d6b8eef2752bb78caa5f717e5037d6383343a5a5c05e
=> => exporting attestation manifest sha256:b1b336ec44ececab20967ca94928f58cce88507d1ad9cfd9f218d0e9bfee5ffc
=> => exporting manifest list sha256:b184d0d81f70f644a89b4c40bbe9efc3d34ddc3161e45b19629bfb1f2eec3eac
=> => naming to docker.io/library/nginx-html-app-mi:latest
=> => unpacking to docker.io/library/nginx-html-app-mi:latest
PS D:\nginx-html-app-mi>
```

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:

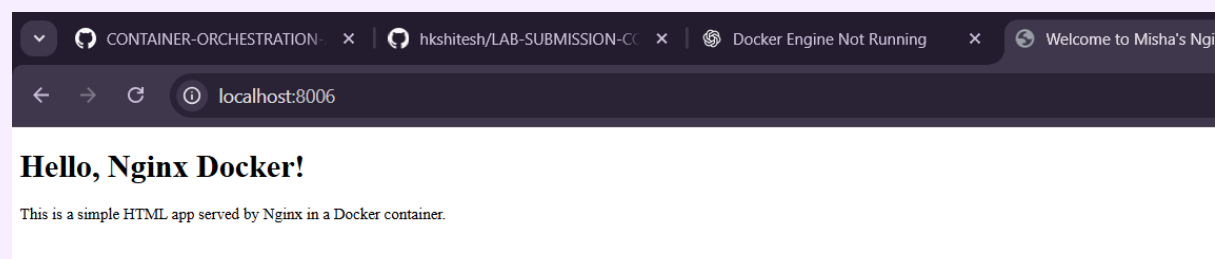
```
docker run -d -p 8006:80 nginx-html-app
```

```
PS D:\nginx-html-app-mi> docker run -d -p 8006:80 nginx-html-app-mi
c953bcfedcc9860738f455b6a4916f1f3e038ee82bac1fca1ccfec1ce9dd158
PS D:\nginx-html-app-mi>
```

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!”.

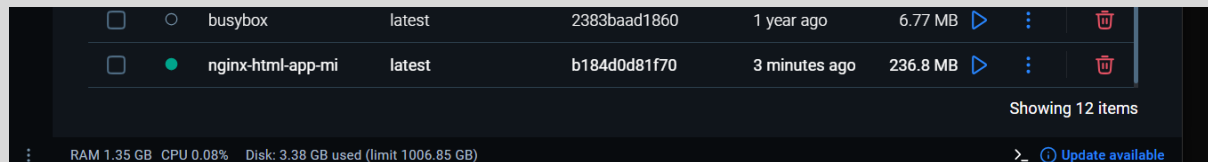


7. Step 6: Stop and Remove the Container

Once you're done, you can stop and remove the container:

`docker ps` # to see running containers

```
PS D:\nginx-html-app-mi> docker ps
CONTAINER ID   IMAGE          NAMES                  COMMAND                  CREATED          STATUS          PORTS
c953bcfedcc9   nginx-html-app-mi   "/docker-entrypoint..."   About a minute ago   Up About a minute   0.0.0.0:8006->80/tcp
PS D:\nginx-html-app-mi>
```



`docker stop <container-id>`

`docker rm <container-id>`

```
PS D:\nginx-html-app-mi> docker stop c953bcfedcc986
c953bcfedcc986
PS D:\nginx-html-app-mi> docker rm c953bcfedcc986
c953bcfedcc986
PS D:\nginx-html-app-mi>
```