Lab Exercise 21- Building a Docker Image for an HTML App Using Nginx

1. Setup

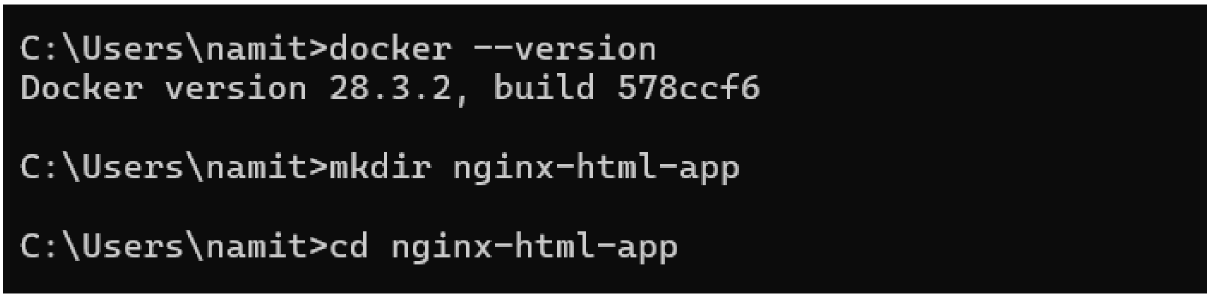
You will need:

- 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
```



```
C:\Users\namit>docker --version
Docker version 28.3.2, build 578ccf6

C:\Users\namit>mkdir nginx-html-app

C:\Users\namit>cd nginx-html-app
```

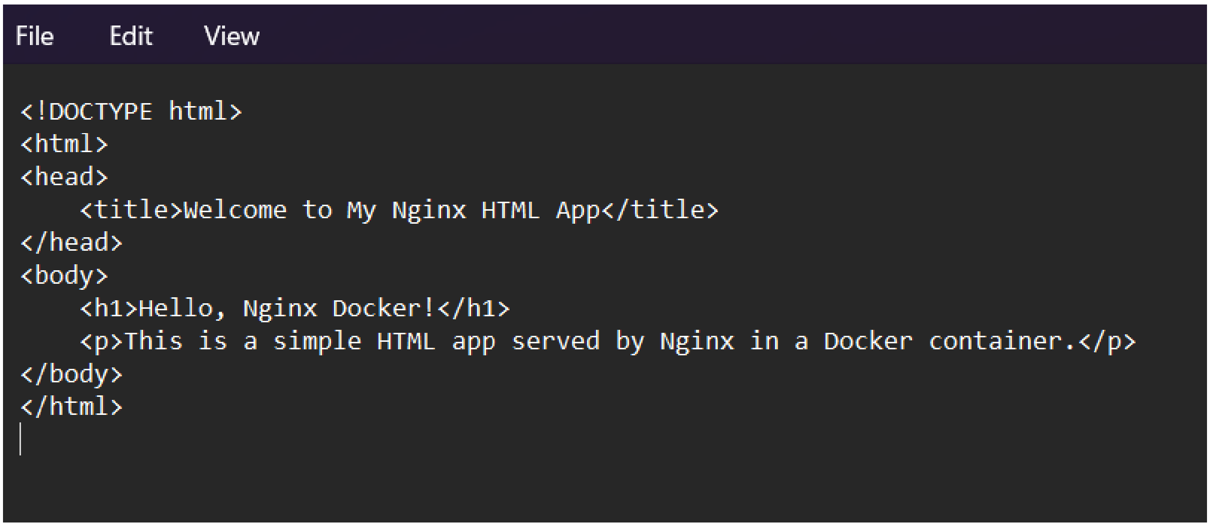
Inside the nginx-html-app directory, create the HTML file.

```
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>
```



```
<!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

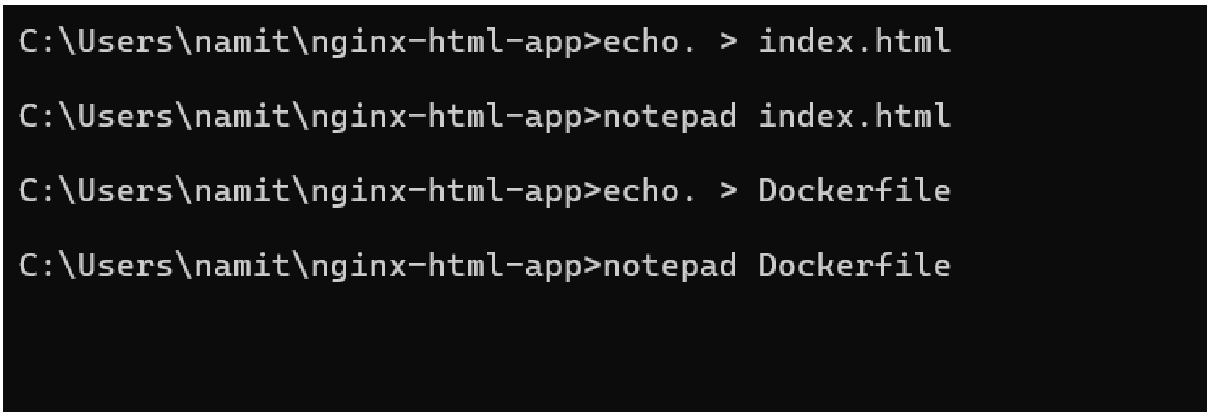
In the same directory, create a Dockerfile. This file will define how to build the Docker image using Nginx as the base image.

```
touch Dockerfile
```

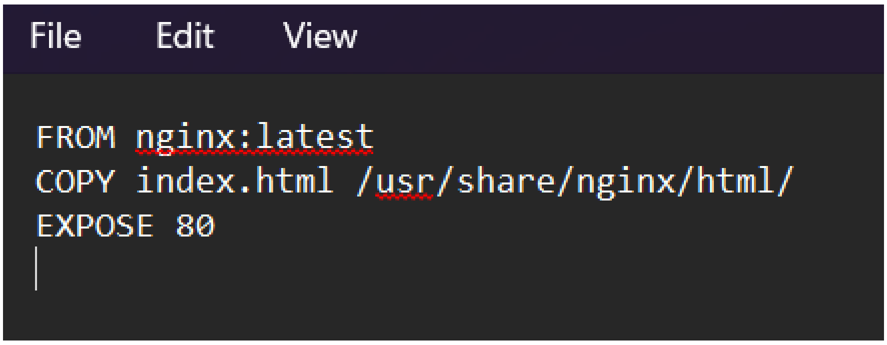
Edit the Dockerfile and add the following content:

```
FROM nginx:latest

COPY index.html /usr/share/nginx/html/
```

```
EXPOSE 80
```



```
C:\Users\namit\nginx-html-app>echo. > index.html

C:\Users\namit\nginx-html-app>notepad index.html

C:\Users\namit\nginx-html-app>echo. > Dockerfile

C:\Users\namit\nginx-html-app>notepad Dockerfile
```



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

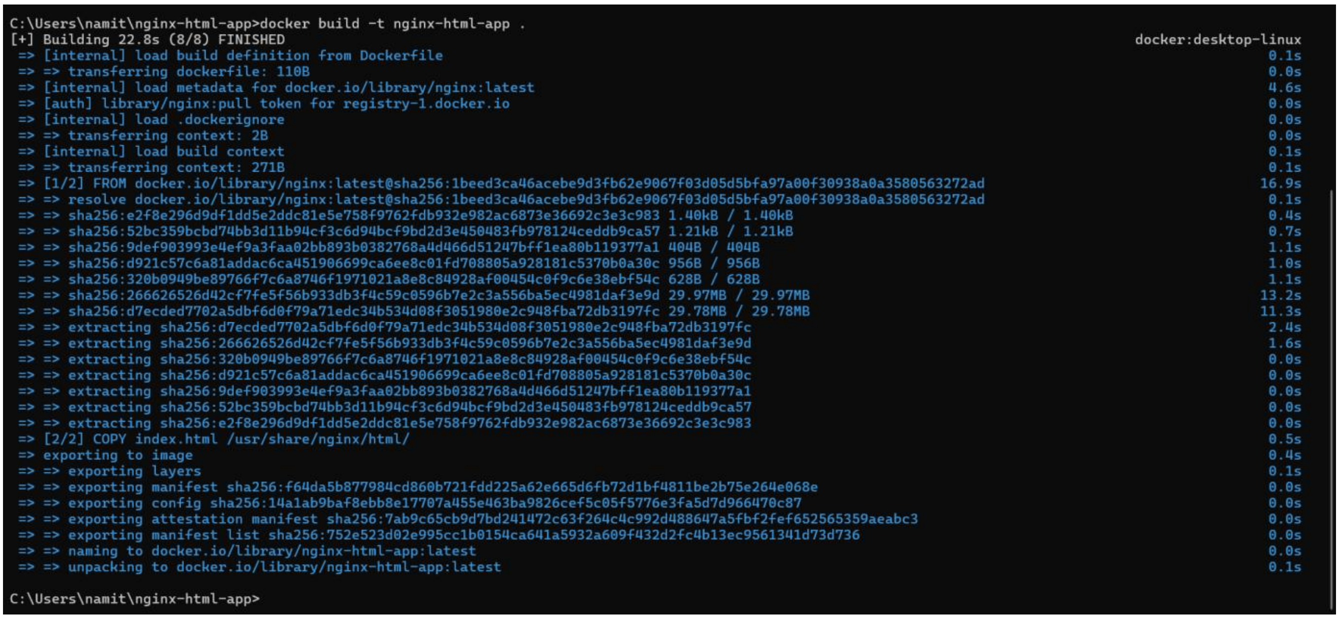
4. Step 3: Build the Docker Image

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.



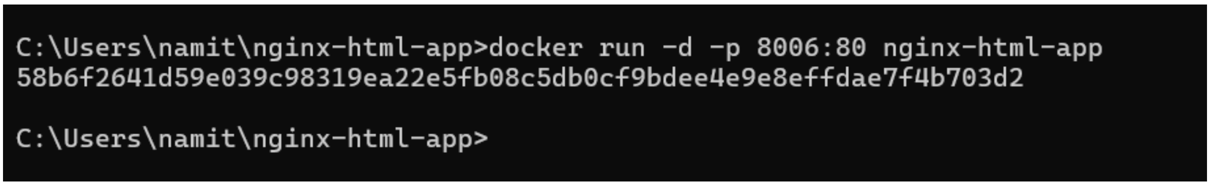
```
C:\Users\namit\nginx-html-app>docker build -t nginx-html-app .
[+] Building 22.8s (8/8) FINISHED                                                                                       docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                                    0.1s
 => => transferring dockerfile: 110B                                                                                                    0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                                                         4.6s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                                            0.0s
 => [internal] load .dockerignore                                                                                                       0.0s
 => => transferring context: 2B                                                                                                         0.0s
 => [internal] load build context                                                                                                       0.1s
 => => transferring context: 271B                                                                                                       0.1s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:1beed3ca46acebe9d3fb62e9067f03d05d5bfa97a00f30938a0a3580563272ad                  16.9s
 => => resolve docker.io/library/nginx:latest@sha256:1beed3ca46acebe9d3fb62e9067f03d05d5bfa97a00f30938a0a3580563272ad                   0.1s
 => => sha256:e2f8e296d9df1dd5e2ddc81e5e758f9762fdb932e982ac6873e36692c3e3c983 1.40kB / 1.40kB                                          0.4s
 => => sha256:52bc359bcbd74bb3d11b94cf3c6d94bcf9bd2d3e450483fb978124ceddb9ca57 1.21kB / 1.21kB                                          0.7s
 => => sha256:9def903993e4ef9a3faa02bb893b0382768a4d466d51247bff1ea80b119377a1 404B / 404B                                              1.1s
 => => sha256:d921c57c6a81addac6ca451906699ca6ee8c01fd708805a928181c5370b0a30c 956B / 956B                                              1.0s
 => => sha256:320b0949be89766f7c6a8746f1971021a8e8c84928af00454c0f9c6e38ebf54c 628B / 628B                                              1.1s
 => => sha256:266626526d42cf7fe5f56b933db3f4c59c0596b7e2c3a556ba5ec4981daf3e9d 29.97MB / 29.97MB                                       13.2s
 => => sha256:d7ecded7702a5dbf6d0f79a71edc34b534d08f3051980e2c948fba72db3197fc 29.78MB / 29.78MB                                       11.3s
 => => extracting sha256:d7ecded7702a5dbf6d0f79a71edc34b534d08f3051980e2c948fba72db3197fc                                               2.4s
 => => extracting sha256:266626526d42cf7fe5f56b933db3f4c59c0596b7e2c3a556ba5ec4981daf3e9d                                               1.6s
 => => extracting sha256:320b0949be89766f7c6a8746f1971021a8e8c84928af00454c0f9c6e38ebf54c                                               0.0s
 => => extracting sha256:d921c57c6a81addac6ca451906699ca6ee8c01fd708805a928181c5370b0a30c                                               0.0s
 => => extracting sha256:9def903993e4ef9a3faa02bb893b0382768a4d466d51247bff1ea80b119377a1                                               0.0s
 => => extracting sha256:52bc359bcbd74bb3d11b94cf3c6d94bcf9bd2d3e450483fb978124ceddb9ca57                                               0.0s
 => => extracting sha256:e2f8e296d9df1dd5e2ddc81e5e758f9762fdb932e982ac6873e36692c3e3c983                                               0.0s
 => [2/2] COPY index.html /usr/share/nginx/html/                                                                                        0.5s
 => exporting to image                                                                                                                  0.4s
 => => exporting layers                                                                                                                 0.1s
 => => exporting manifest sha256:f64da5b877984cd860b721fdd225a62e665d6fb72d1bf4811be2b75e264e068e                                       0.0s
 => => exporting config sha256:14a1ab9baf8ebb8e17707a455e463ba9826cef5c05f5776e3fa5d7d966470c87                                         0.0s
 => => exporting attestation manifest sha256:7ab9c65cb9d7bd241472c63f264c4c992d488647a5fbf2fef652565359aeabc3                           0.0s
 => => exporting manifest list sha256:752e523d02e995cc1b0154ca641a5932a609f432d2fc4b13ec9561341d73d736                                  0.0s
 => => naming to docker.io/library/nginx-html-app:latest                                                                                0.0s
 => => unpacking to docker.io/library/nginx-html-app:latest                                                                             0.1s

C:\Users\namit\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
```

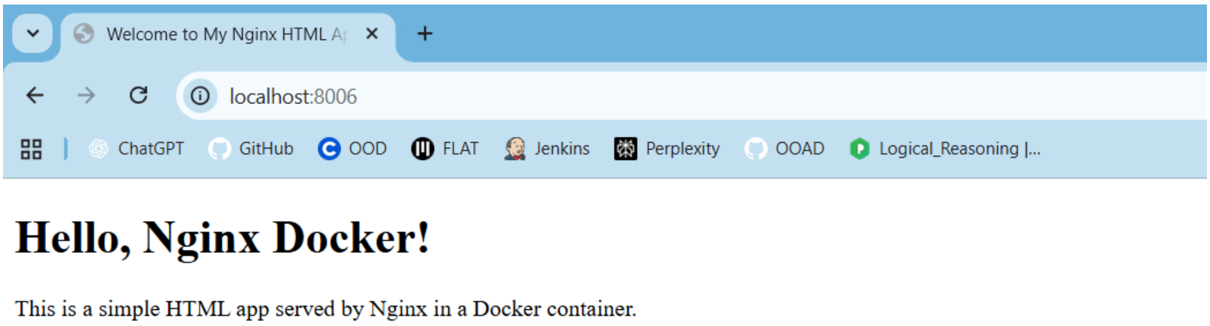
This command runs the container in detached mode (-d) and maps port 8006 on your host machine to port 80 inside the container, where Nginx is serving your HTML app.



```
C:\Users\namit\nginx-html-app>docker run -d -p 8006:80 nginx-html-app
58b6f2641d59e039c98319ea22e5fb08c5db0cf9bdee4e9e8effdae7f4b703d2

C:\Users\namit\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!”.



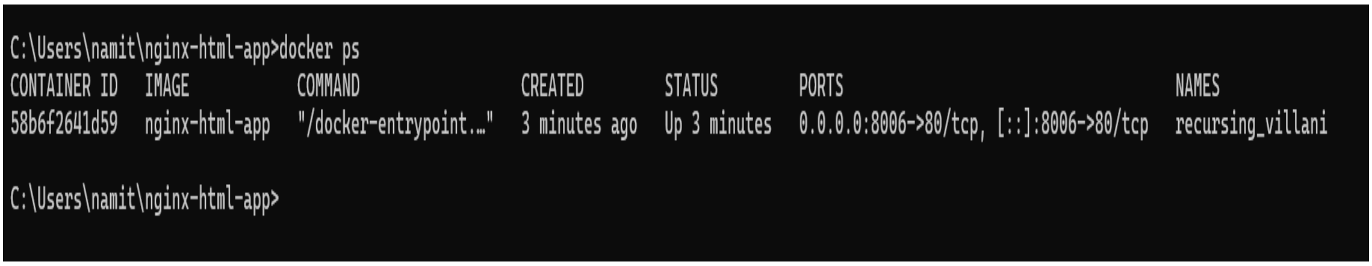
Hello, Nginx Docker!

This is a simple HTML app served by Nginx in a Docker container.

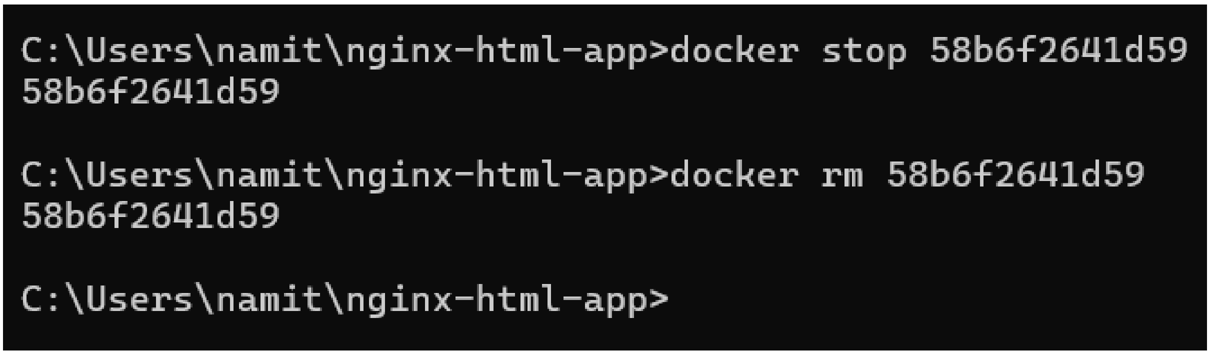
7. Step 6: Stop and Remove the Container

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>
```



```
C:\Users\namit\nginx-html-app>docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED         STATUS         PORTS                                  NAMES
58b6f2641d59   nginx-html-app  "/docker-entrypoint.…"   3 minutes ago   Up 3 minutes   0.0.0.0:8006->80/tcp, [::]:8006->80/tcp   recursing_villani

C:\Users\namit\nginx-html-app>
```



```
C:\Users\namit\nginx-html-app>docker stop 58b6f2641d59
58b6f2641d59

C:\Users\namit\nginx-html-app>docker rm 58b6f2641d59
58b6f2641d59

C:\Users\namit\nginx-html-app>
```