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

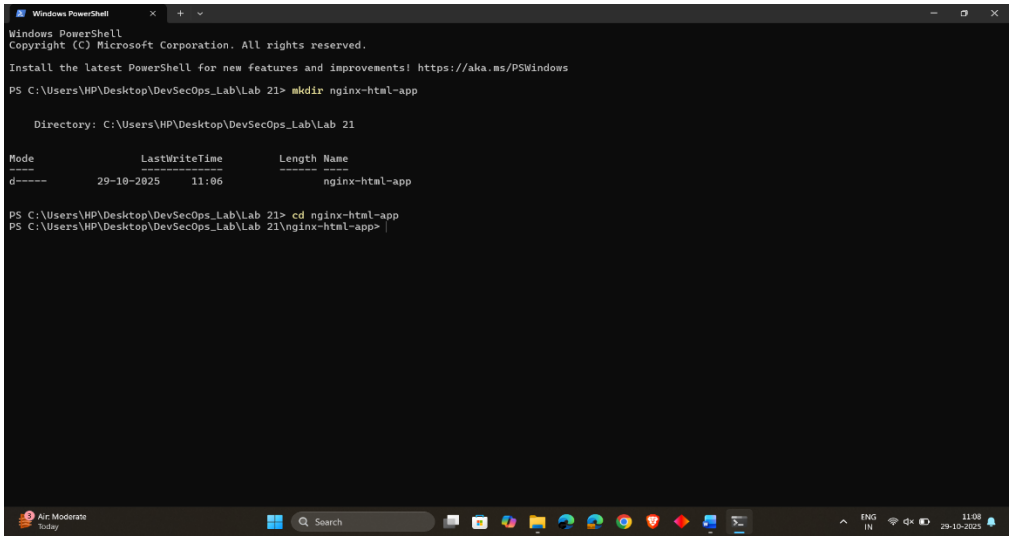
Create a directory for your HTML app and place an index.html file in it.

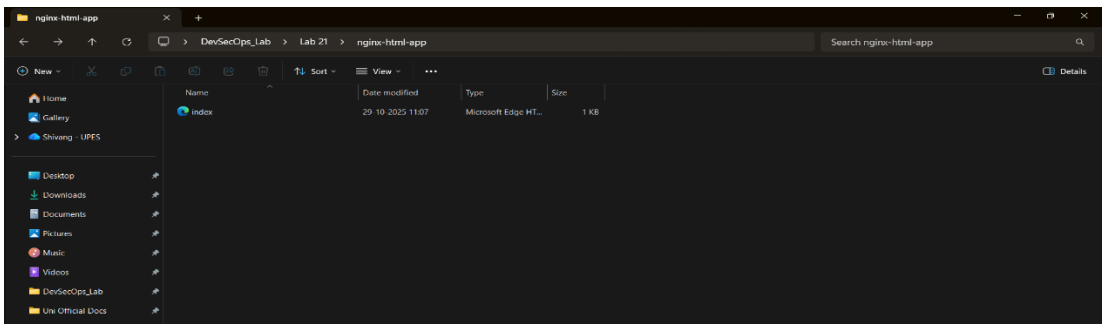
```
mkdir nginx-html-app

cd nginx-html-app
```

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

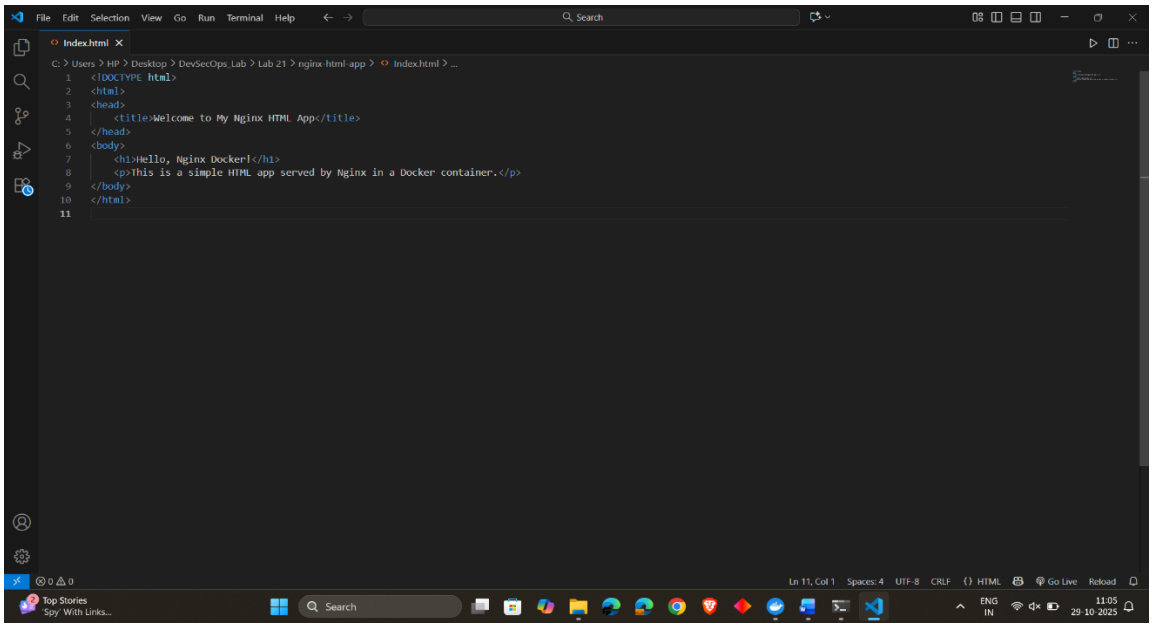
```
touch index.html
```





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>
```



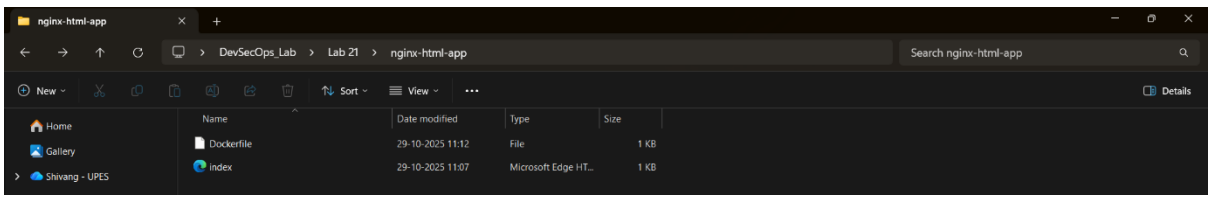
3. Step 2: Create a Dockerfile

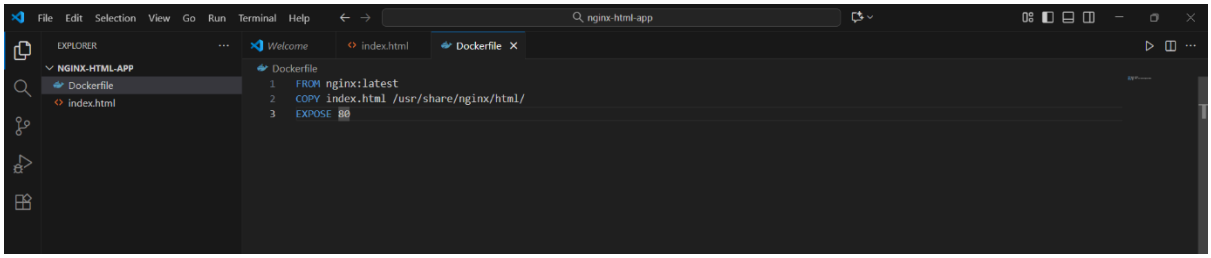
In the same directory, create a Dockerfile. This file will define how to build the Docker image using Nginx as the base image.

```
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 .
```

Docker will use the Nginx base image, copy your index.html into the appropriate directory,

and build the image.

```
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app> docker build -t nginx-html-app .
[+] Building 277.6s (8/8) FINISHED                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                  0.0s
 => => transferring dockerfile: 105B                                                                                  0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                                       9.4s
 => [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                                                                                     0.0s
 => => transferring context: 272B                                                                                     0.0s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:b619c34a163ac12f68c1982568a122c4953dbf3126b8dbf0cc2f6fdbfd85de27  267.8s
 => => resolve docker.io/library/nginx:latest@sha256:b619c34a163ac12f68c1982568a122c4953dbf3126b8dbf0cc2f6fdbfd85de27  0.0s
 => => sha256:5545b08f9d26518aae35920f1c1b7cdf962b32b69e9feba182bfb3e0b90dcc2c 1.40kB / 1.40kB                         0.9s
 => => sha256:35fb9ffa6621d9044c7062eb2fe4b181ce919447e2575a8577dcedbd7aa5aa67 1.21kB / 1.21kB                         1.5s
 => => sha256:89df300a082a0f5d71c8aff1944dc57a49544592d8d804fe7c61b7e844aa80b8 404B / 404B                             2.5s
 => => sha256:1bace208328954ff24f16c62a7492c4b3a4d9451a10f23631417f3feebe3f44d 955B / 955B                             2.4s
 => => sha256:0e86847a3920c130e5f7e66bdf15b66b9ed5426b80ecf1b2f8e7960794620a63 629B / 629B                             0.9s
 => => sha256:a0a6ab1415585a8dc626877b8de231086ab4d45ea8709d35b8ac7ec24bea3038 29.97MB / 29.97MB                     255.4s
 => => sha256:38513bd7256313495cdd83b3b0915a633cfa475dc2a07072ab2c8d191020ca5d 29.78MB / 29.78MB                     264.8s
 => => extracting sha256:38513bd7256313495cdd83b3b0915a633cfa475dc2a07072ab2c8d191020ca5d                             0.6s
 => => extracting sha256:a0a6ab1415585a8dc626877b8de231086ab4d45ea8709d35b8ac7ec24bea3038                             0.5s
 => => extracting sha256:0e86847a3920c130e5f7e66bdf15b66b9ed5426b80ecf1b2f8e7960794620a63                             0.0s
 => => extracting sha256:1bace208328954ff24f16c62a7492c4b3a4d9451a10f23631417f3feebe3f44d                             0.0s
 => => extracting sha256:89df300a082a0f5d71c8aff1944dc57a49544592d8d804fe7c61b7e844aa80b8                             0.0s
 => => extracting sha256:35fb9ffa6621d9044c7062eb2fe4b181ce919447e2575a8577dcedbd7aa5aa67                             0.0s
 => => extracting sha256:5545b08f9d26518aae35920f1c1b7cdf962b32b69e9feba182bfb3e0b90dcc2c                             0.0s
 => [2/2] COPY index.html /usr/share/nginx/html/                                                                      0.1s
 => exporting to image                                                                                                0.1s
 => => exporting layers                                                                                               0.1s
 => => exporting manifest sha256:19ec18b4cdd354476bf4b90482d7cf6b46993fc1178d0f5360ae6e6ecb630585                     0.0s
 => => exporting config sha256:d179db5794308ae7351b135c1aca750af85367af68813bdfa36b7b60b54e71e5                       0.0s
 => => exporting attestation manifest sha256:d4a8ee25ab2aec1fdfa471e14f6b0e11ae7d070c61c4b758803aacf7b96c08b8         0.0s
 => => exporting manifest list sha256:d09021b70462174f1b7e08374b8639518dd4a27065bc84c550a6bdaf926eee7f                0.0s
 => => naming to docker.io/library/nginx-html-app:latest                                                              0.0s
 => => unpacking to docker.io/library/nginx-html-app:latest                                                           0.0s
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app>
```

```
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app> docker images
REPOSITORY        TAG       IMAGE ID       CREATED          SIZE
nginx-html-app    latest    d09021b70462   2 minutes ago    225MB
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app>
```

5. Step 4: Run the Docker Container

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

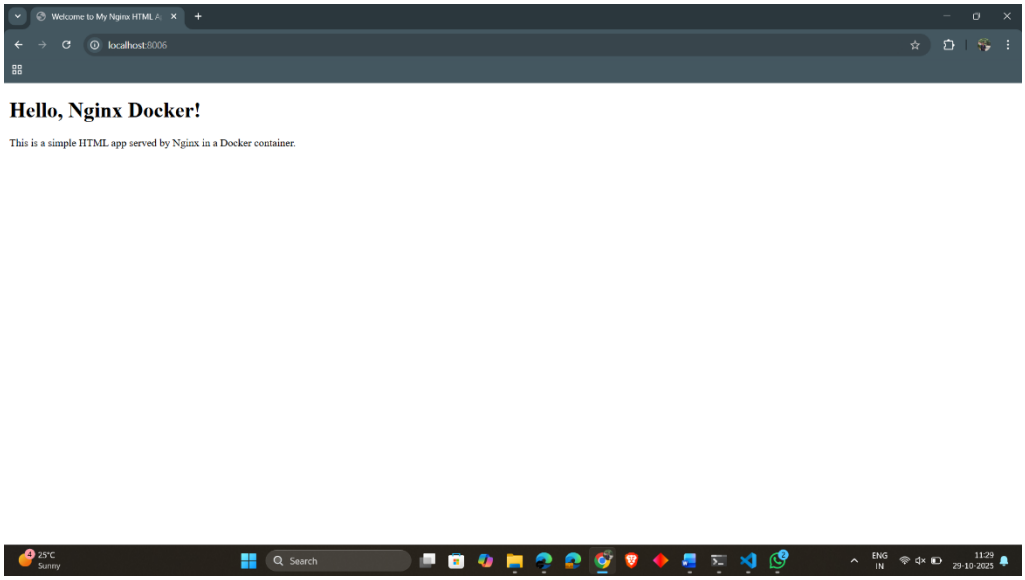
```
docker run -d -p 8006:80 nginx-html-app
```

```
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app> docker run -d -p 8006:80 nginx-html-app
7413b5d01c73dd8b1d6cb2415a6217dbad5191561f567f8eb49198cec838ef32
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\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.

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 C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app> docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED         STATUS         PORTS                                       NAMES
7413b5d01c73   nginx-html-app   "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:8006->80/tcp, [::]:8006->80/tcp   romantic_dhawan
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app> docker stop 7413b5d01c73
7413b5d01c73
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app>
```

```
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app> docker rm 7413b5d01c
7413b5d01c
PS C:\Users\HP\Desktop\DevSecOps_Lab\Lab 21\nginx-html-app>
```