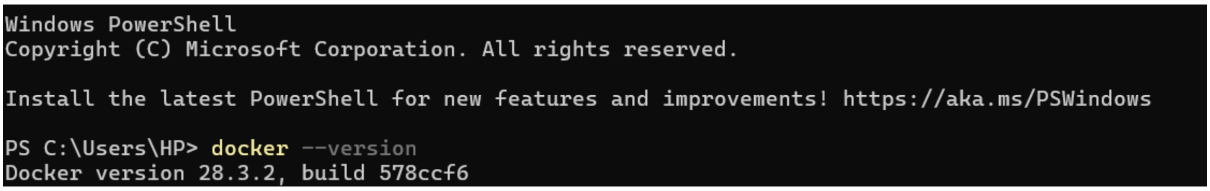
Lab Exercise 4- Building a Docker Image for an HTML App Using Nginx

1. Setup

You will need:

- Docker installed on your machine.
- A simple HTML file for the app.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

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

2. Step 1: Create the HTML File

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

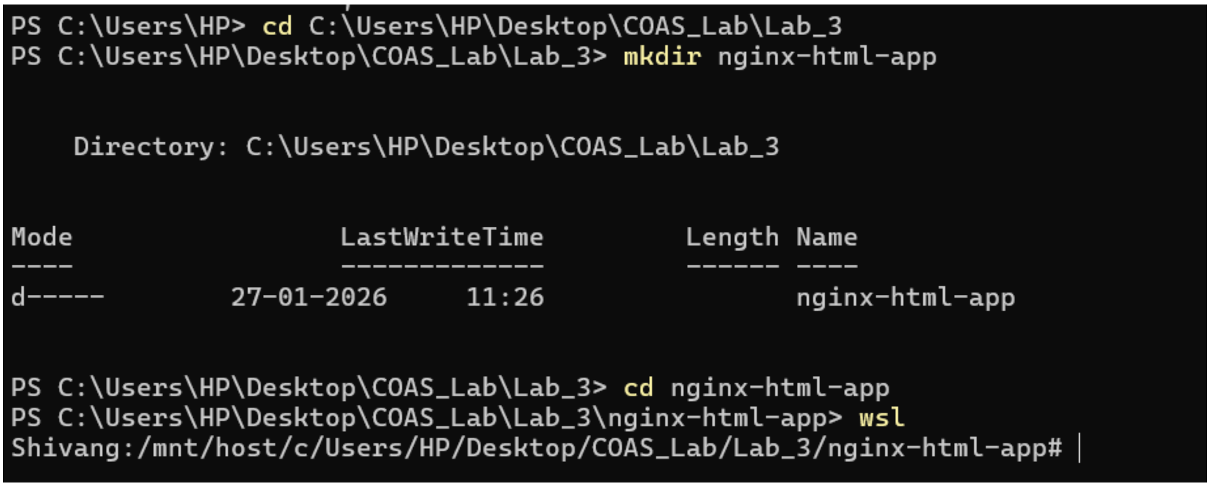
```
mkdir nginx-html-app

cd nginx-html-app
```

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

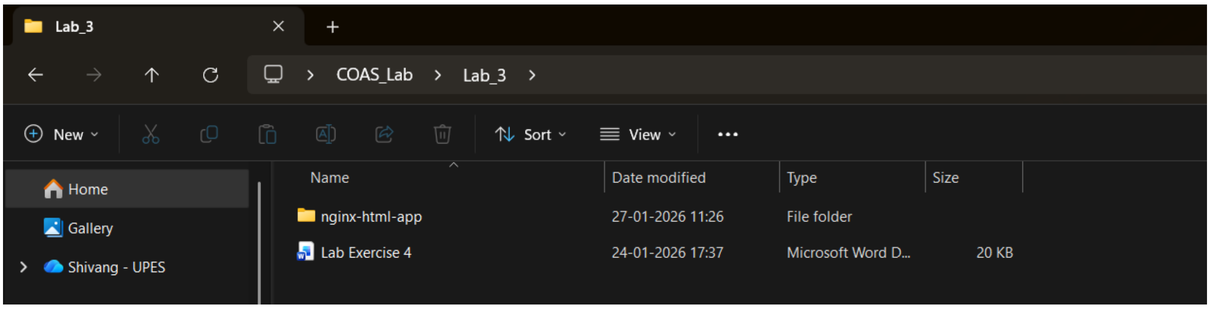
```
touch index.html
```

Edit the index.html file with the following content (or any custom HTML content you want):

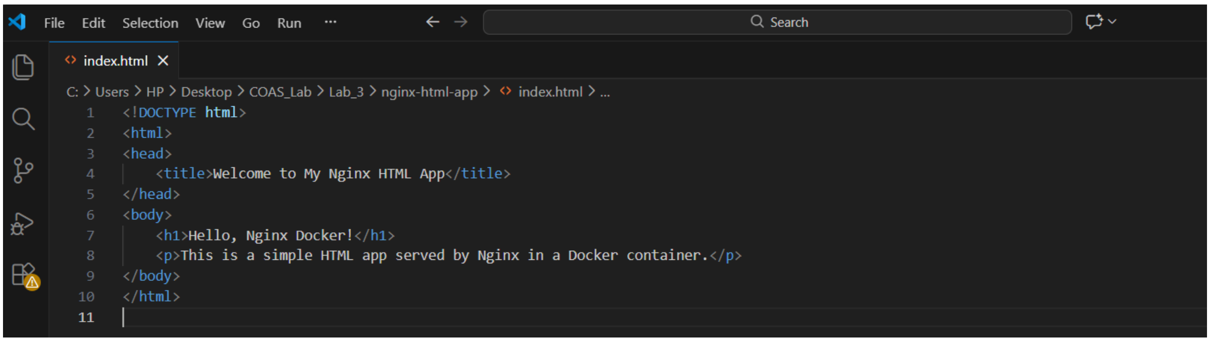
```
<!DOCTYPE html>
<html>
<head>
   <title>Welcome to My Nginx HTML App</title>
</head>
<body>
   <h1>Hello, Nginx Docker!</h1>
   <p>This is a simple HTML app served by Nginx in a Docker container.</p>
</body>
</html>
```



```
PS C:\Users\HP> cd C:\Users\HP\Desktop\COAS_Lab\Lab_3
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3> mkdir nginx-html-app


    Directory: C:\Users\HP\Desktop\COAS_Lab\Lab_3


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        27-01-2026     11:26                nginx-html-app


PS C:\Users\HP\Desktop\COAS_Lab\Lab_3> cd nginx-html-app
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\nginx-html-app> wsl
Shivang:/mnt/host/c/Users/HP/Desktop/COAS_Lab/Lab_3/nginx-html-app# |
```



Name	Date modified	Type	Size
nginx-html-app	27-01-2026 11:26	File folder	
Lab Exercise 4	24-01-2026 17:37	Microsoft Word D...	20 KB



File
Edit
Selection
View
Go
Run
Search
index.html
C: > Users > HP > Desktop > COAS_Lab > Lab_3 > nginx-html-app > index.html > ...

```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>
```

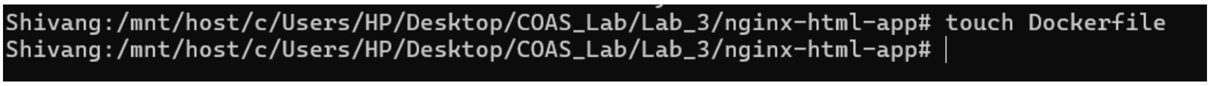
3. Step 2: Create a Dockerfile

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

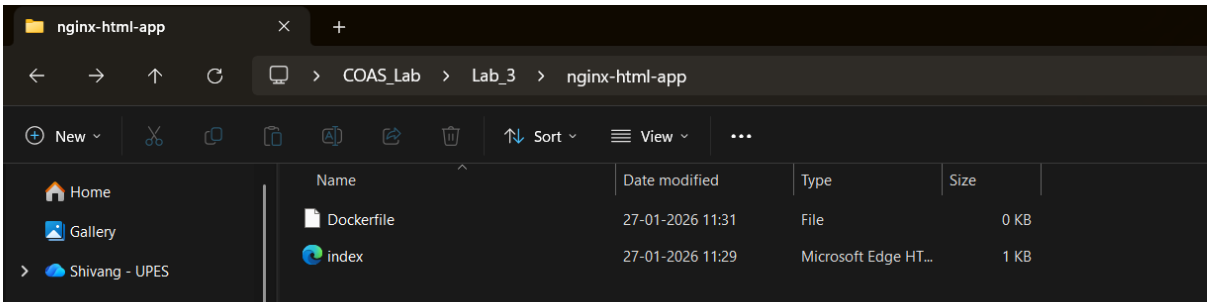
```
touch Dockerfile
```

Edit the Dockerfile and add the following content:

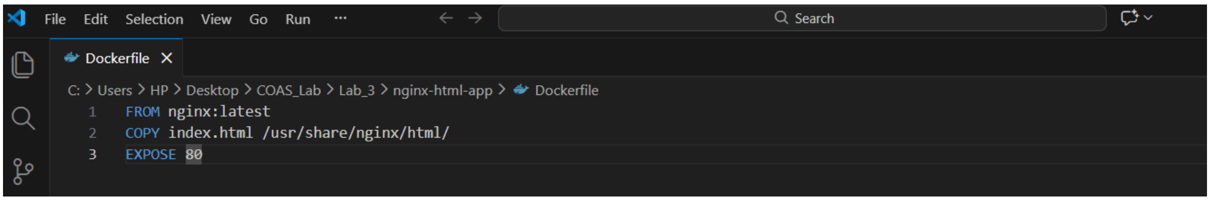
```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```



```
Shivang:/mnt/host/c/Users/HP/Desktop/COAS_Lab/Lab_3/nginx-html-app# touch Dockerfile
Shivang:/mnt/host/c/Users/HP/Desktop/COAS_Lab/Lab_3/nginx-html-app# |
```



Name	Date modified	Type	Size
Dockerfile	27-01-2026 11:31	File	0 KB
index	27-01-2026 11:29	Microsoft Edge HT...	1 KB



File Edit Selection View Go Run ...
Search
Dockerfile
C: > Users > HP > Desktop > COAS_Lab > Lab_3 > nginx-html-app > Dockerfile

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

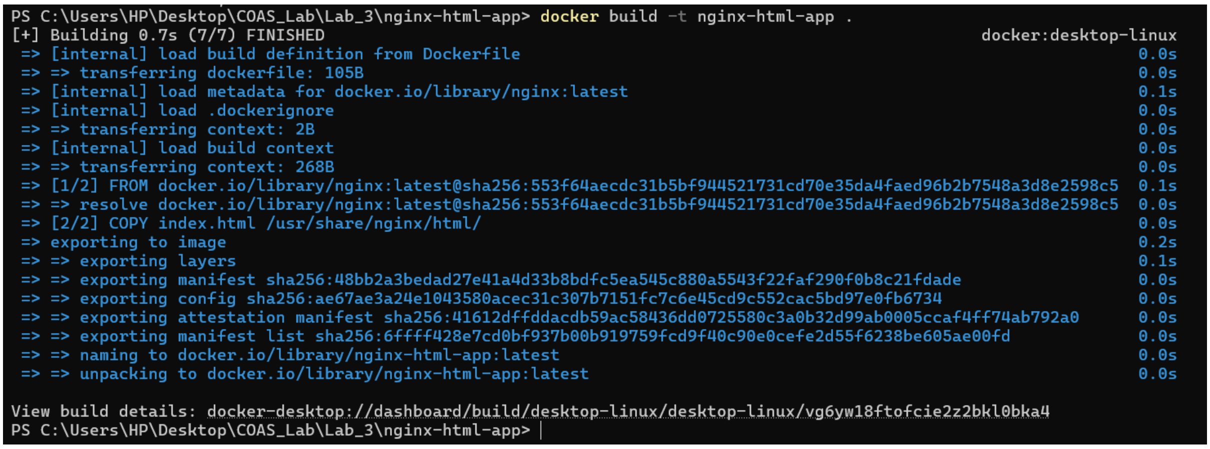
4. Step 3: Build the Docker Image

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.



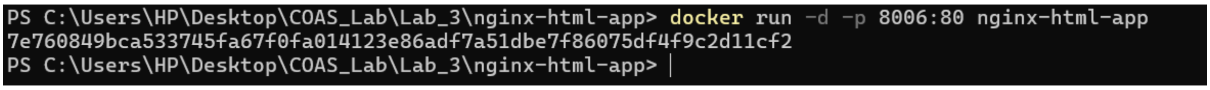
```
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\nginx-html-app> docker build -t nginx-html-app .
[+] Building 0.7s (7/7) FINISHED                                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                             0.0s
 => => transferring dockerfile: 105B                                                                             0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                                  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@sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c5  0.1s
 => => resolve docker.io/library/nginx:latest@sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c5  0.0s
 => [2/2] COPY index.html /usr/share/nginx/html/                                                                 0.0s
 => exporting to image                                                                                           0.2s
 => => exporting layers                                                                                          0.1s
 => => exporting manifest sha256:48bb2a3bedad27e41a4d33b8bdfc5ea545c880a5543f22faf290f0b8c21fdade                 0.0s
 => => exporting config sha256:ae67ae3a24e1043580acec31c307b7151fc7c6e45cd9c552cac5bd97e0fb6734                   0.0s
 => => exporting attestation manifest sha256:41612dffddacdb59ac58436dd0725580c3a0b32d99ab0005ccaf4ff74ab792a0     0.0s
 => => exporting manifest list sha256:6ffff428e7cd0bf937b00b919759fcd9f40c90e0cefe2d55f6238be605ae00fd            0.0s
 => => naming to docker.io/library/nginx-html-app:latest                                                         0.0s
 => => unpacking to docker.io/library/nginx-html-app:latest                                                      0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/vg6yw18ftofcie2z2bkl0bka4
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\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
```

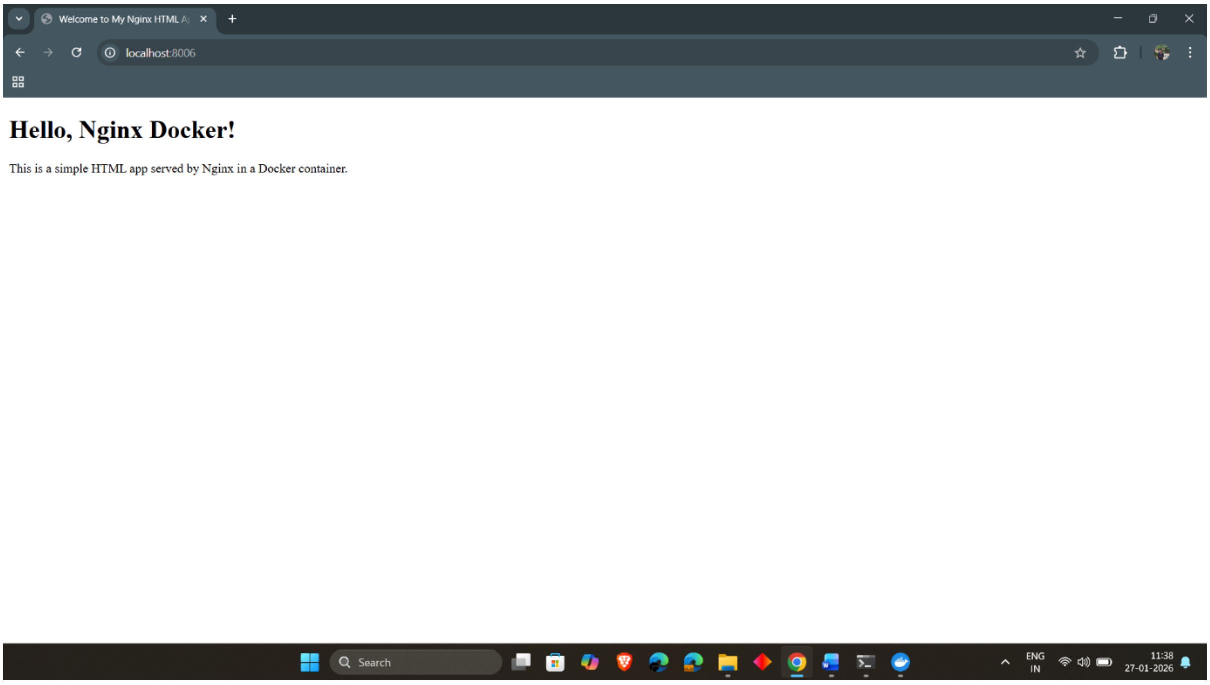
This command runs the container in detached mode (-d) and maps port 8006 on your host machine to port 80 inside the container, where Nginx is serving your HTML app.



```
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\nginx-html-app> docker run -d -p 8006:80 nginx-html-app
7e760849bca533745fa67f0fa014123e86adf7a51dbe7f86075df4f9c2d11cf2
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\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!”.



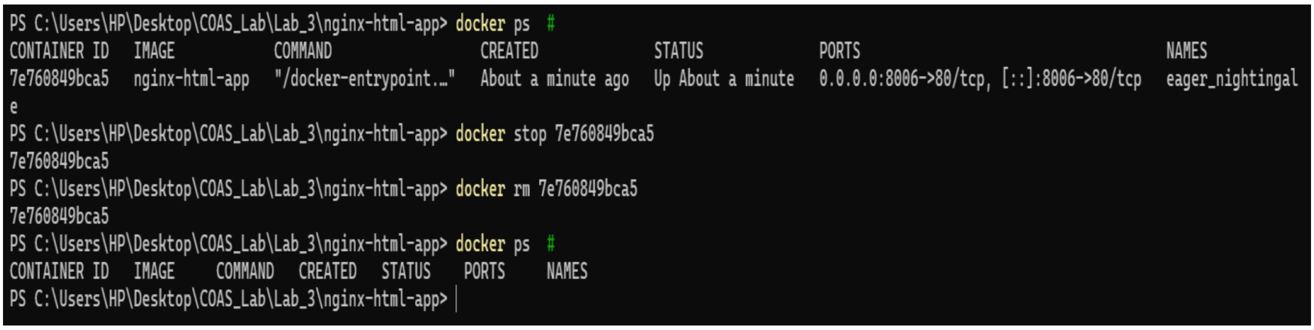
Hello, Nginx Docker!

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>
```



```
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\nginx-html-app> docker ps  #
CONTAINER ID   IMAGE            COMMAND                  CREATED              STATUS              PORTS                                   NAMES
7e760849bca5   nginx-html-app   "/docker-entrypoint.…"   About a minute ago   Up About a minute   0.0.0.0:8006->80/tcp, [::]:8006->80/tcp   eager_nightingal
e
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\nginx-html-app> docker stop 7e760849bca5
7e760849bca5
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\nginx-html-app> docker rm 7e760849bca5
7e760849bca5
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\nginx-html-app> docker ps  #
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\HP\Desktop\COAS_Lab\Lab_3\nginx-html-app> |
```