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

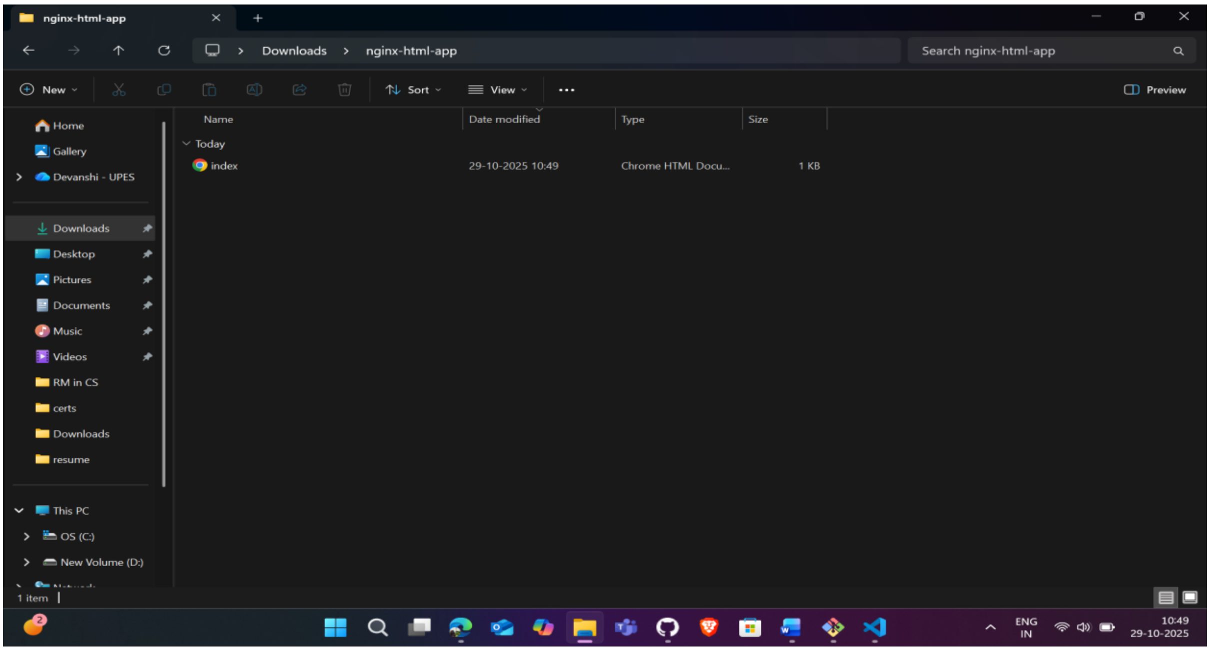
1. Setup

You will need:

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

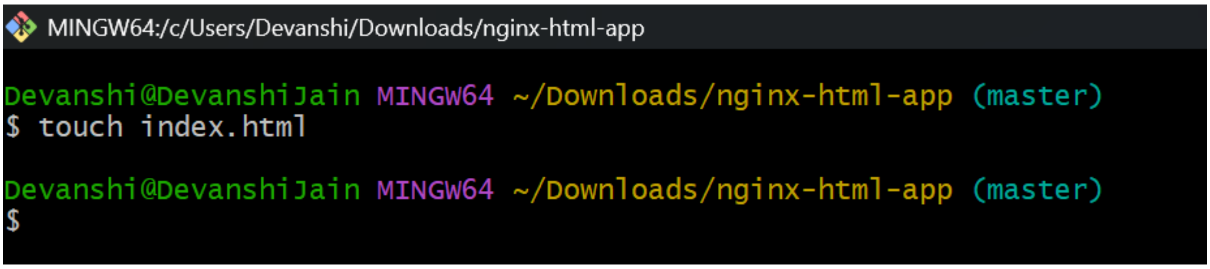
2. Step 1: Create the HTML File

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



Name	Date modified	Type	Size
Today			
index	29-10-2025 10:49	Chrome HTML Docu...	1 KB

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

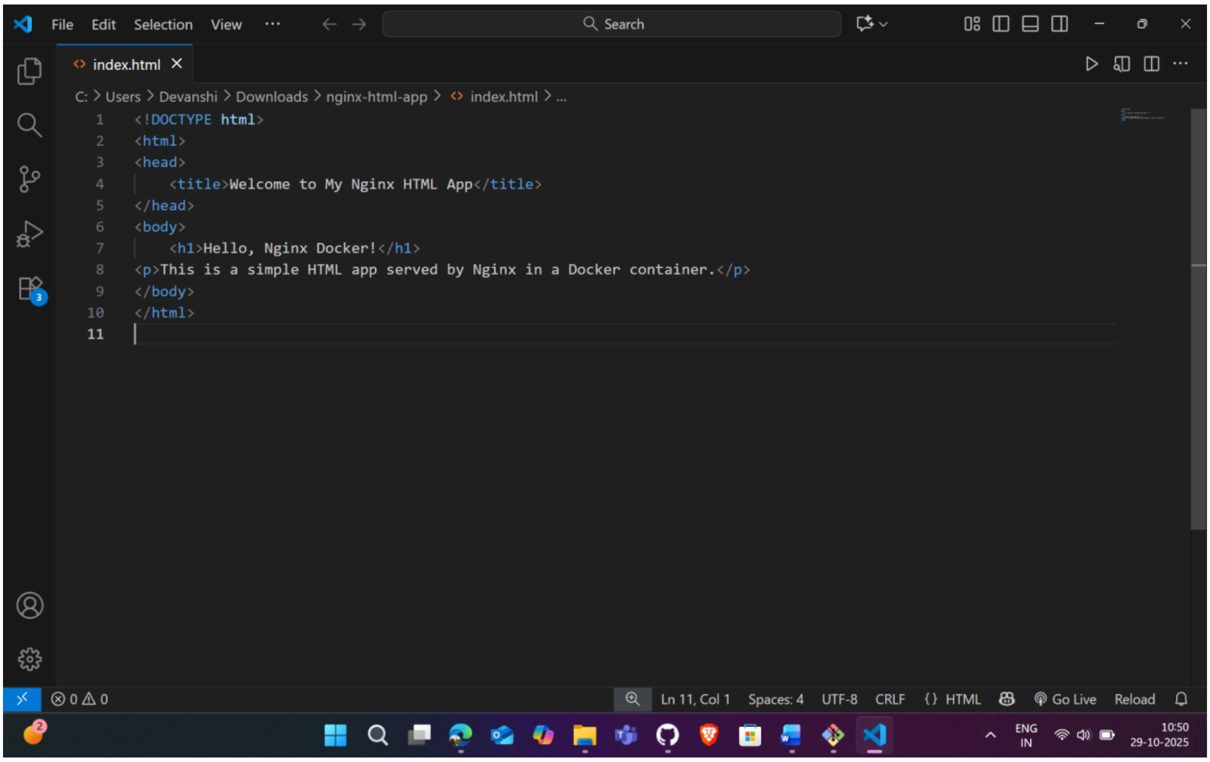


```
MINGW64:/c/Users/Devanshi/Downloads/nginx-html-app

Devanshi@DevanshiJain MINGW64 ~/Downloads/nginx-html-app (master)
$ touch index.html

Devanshi@DevanshiJain MINGW64 ~/Downloads/nginx-html-app (master)
$
```

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



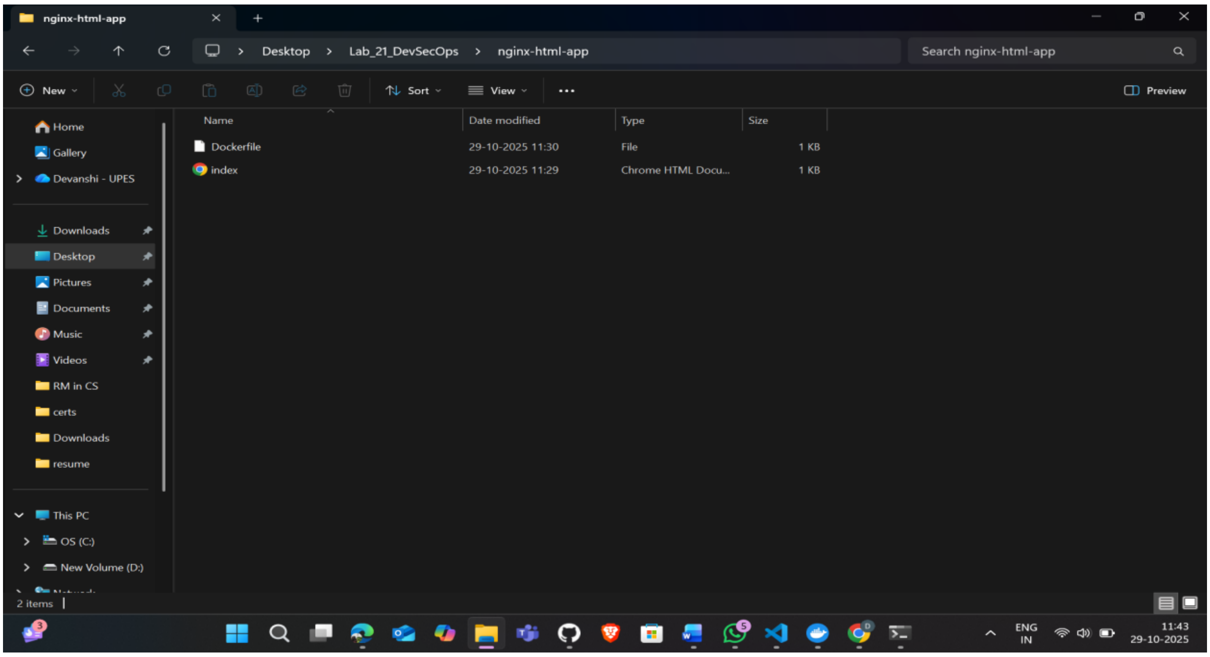
C: > Users > Devanshi > Downloads > nginx-html-app > index.html > ...

```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

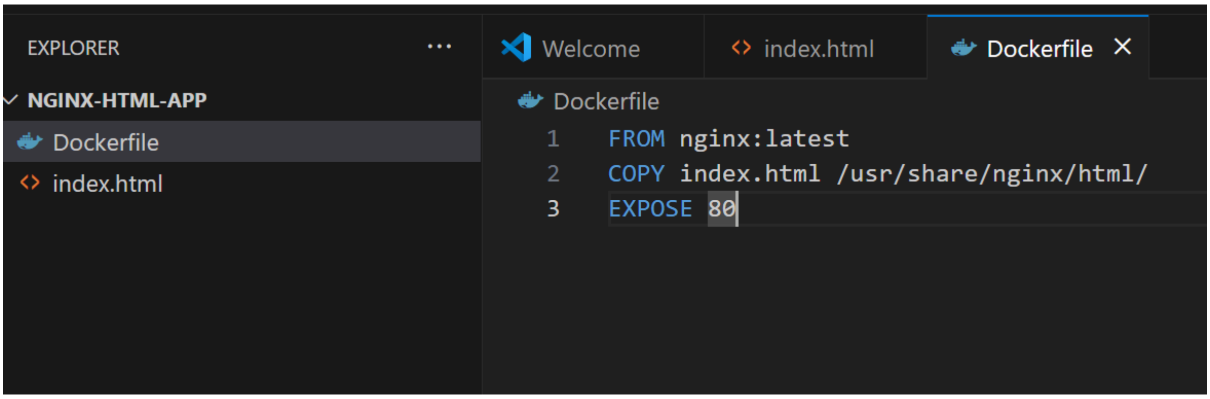
In the same directory, create a Dockerfile. This file will define how to build the Docker image using Nginx as the base image.



Desktop > Lab_21_DevSecOps > nginx-html-app

Name	Date modified	Type	Size
Dockerfile	29-10-2025 11:30	File	1 KB
index	29-10-2025 11:29	Chrome HTML Docu...	1 KB

Edit the Dockerfile and add the following content:

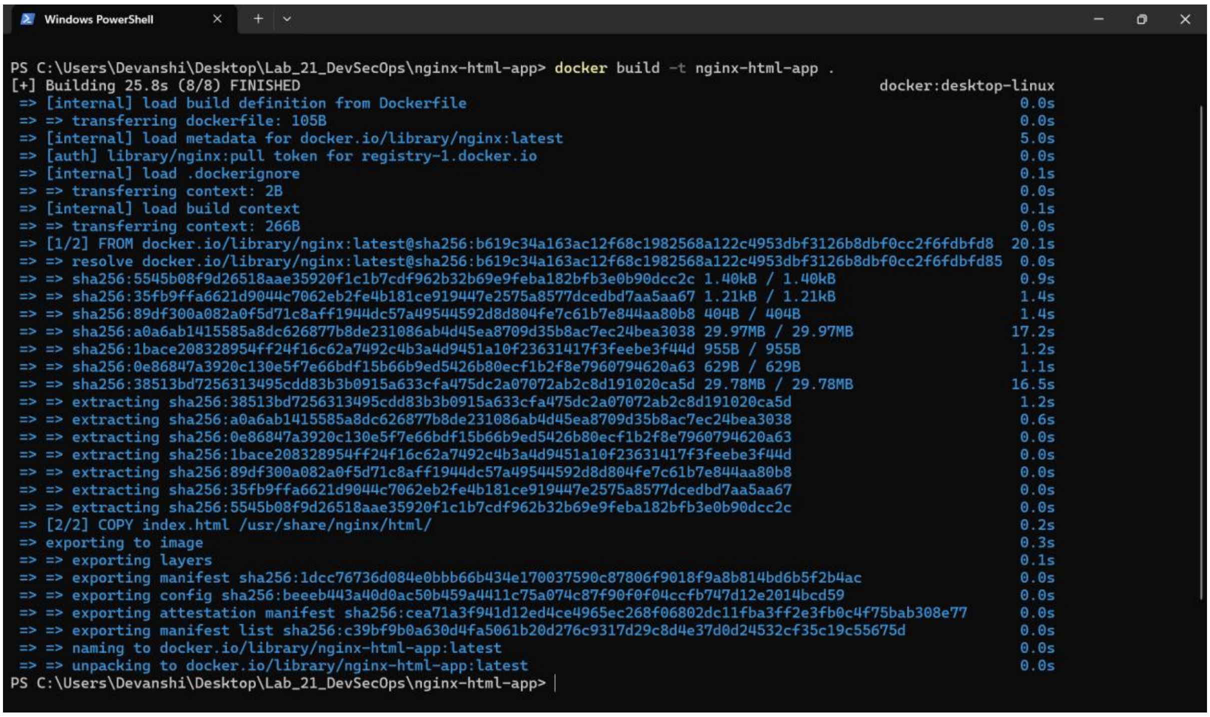


```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```

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):

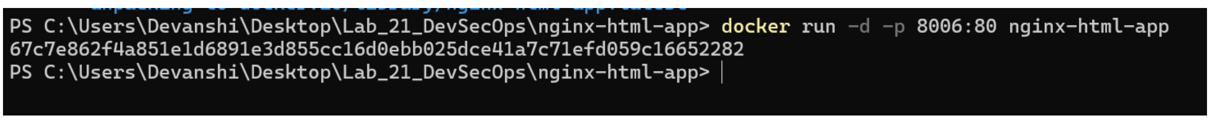


```
PS C:\Users\Devanshi\Desktop\Lab_21_DevSecOps\nginx-html-app> docker build -t nginx-html-app .
[+] Building 25.8s (8/8) FINISHED                                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                               0.0s
 => => transferring dockerfile: 105B                                                                               0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                                    5.0s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                       0.0s
 => [internal] load .dockerignore                                                                                  0.1s
 => => transferring context: 2B                                                                                    0.0s
 => [internal] load build context                                                                                  0.1s
 => => transferring context: 266B                                                                                  0.0s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:b619c34a163ac12f68c1982568a122c4953dbf3126b8dbf0cc2f6fdbfd8  20.1s
 => => resolve docker.io/library/nginx:latest@sha256:b619c34a163ac12f68c1982568a122c4953dbf3126b8dbf0cc2f6fdbfd85  0.0s
 => => sha256:5545b08f9d26518aae35920f1c1b7cdf962b32b69e9feba182bfb3e0b90dcc2c 1.40kB / 1.40kB                     0.9s
 => => sha256:35fb9ffa6621d9044c7062eb2fe4b181ce919447e2575a8577dcedbd7aa5aa67 1.21kB / 1.21kB                     1.4s
 => => sha256:89df300a082a0f5d71c8aff1944dc57a49544592d8d804fe7c61b7e844aa80b8 404B / 404B                         1.4s
 => => sha256:a0a6ab1415585a8dc626877b8de231086ab4d45ea8709d35b8ac7ec24bea3038 29.97MB / 29.97MB                  17.2s
 => => sha256:1bace208328954ff24f16c62a7492c4b3a4d9451a10f23631417f3feebe3f44d 955B / 955B                         1.2s
 => => sha256:0e86847a3920c130e5f7e66bdf15b66b9ed5426b80ecf1b2f8e7960794620a63 629B / 629B                         1.1s
 => => sha256:38513bd7256313495cdd83b3b0915a633cfa475dc2a07072ab2c8d191020ca5d 29.78MB / 29.78MB                  16.5s
 => => extracting sha256:38513bd7256313495cdd83b3b0915a633cfa475dc2a07072ab2c8d191020ca5d                          1.2s
 => => extracting sha256:a0a6ab1415585a8dc626877b8de231086ab4d45ea8709d35b8ac7ec24bea3038                          0.6s
 => => 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/                                                                   0.2s
 => exporting to image                                                                                             0.3s
 => => exporting layers                                                                                            0.1s
 => => exporting manifest sha256:1dcc76736d084e0bbb66b434e170037590c87806f9018f9a8b814bd6b5f2b4ac                  0.0s
 => => exporting config sha256:beeeb443a40d0ac50b459a4411c75a074c87f90f0f04ccfb747d12e2014bcd59                    0.0s
 => => exporting attestation manifest sha256:cea71a3f941d12ed4ce4965ec268f06802dc11fba3ff2e3fb0c4f75bab308e77      0.0s
 => => exporting manifest list sha256:c39bf9b0a630d4fa5061b20d276c9317d29c8d4e37d0d24532cf35c19c55675d             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\Devanshi\Desktop\Lab_21_DevSecOps\nginx-html-app> |
```

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:

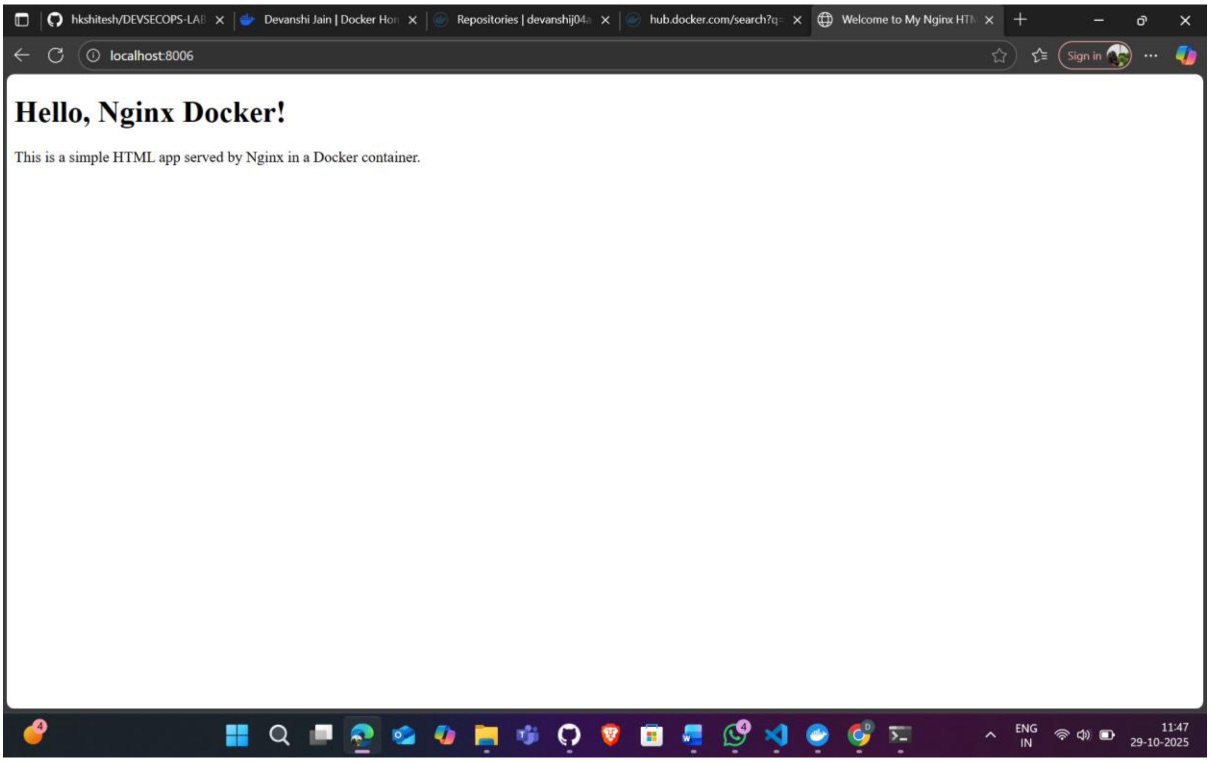


```
PS C:\Users\Devanshi\Desktop\Lab_21_DevSecOps\nginx-html-app> docker run -d -p 8006:80 nginx-html-app
67c7e862f4a851e1d6891e3d855cc16d0ebb025dce41a7c71efd059c16652282
PS C:\Users\Devanshi\Desktop\Lab_21_DevSecOps\nginx-html-app> |
```

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!”.

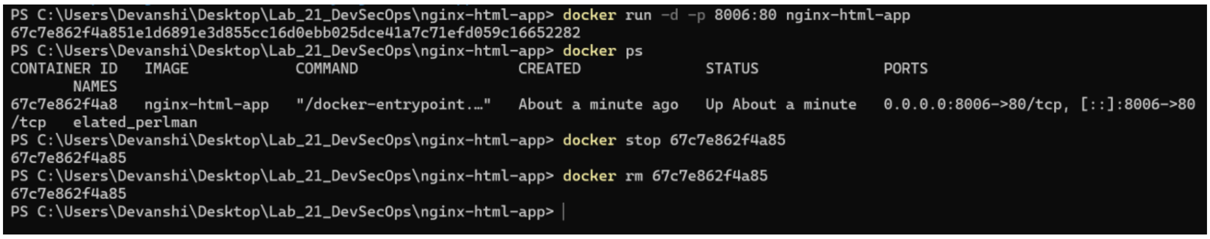


Hello, Nginx Docker!

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:



```
PS C:\Users\Devanshi\Desktop\Lab_21_DevSecOps\nginx-html-app> docker run -d -p 8006:80 nginx-html-app
67c7e862f4a851e1d6891e3d855cc16d0ebb025dce41a7c71efd059c16652282
PS C:\Users\Devanshi\Desktop\Lab_21_DevSecOps\nginx-html-app> docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED              STATUS              PORTS
      NAMES
67c7e862f4a8   nginx-html-app   "/docker-entrypoint.…"   About a minute ago   Up About a minute   0.0.0.0:8006->80/tcp, [::]:8006->80
/tcp   elated_perlman
PS C:\Users\Devanshi\Desktop\Lab_21_DevSecOps\nginx-html-app> docker stop 67c7e862f4a85
67c7e862f4a85
PS C:\Users\Devanshi\Desktop\Lab_21_DevSecOps\nginx-html-app> docker rm 67c7e862f4a85
67c7e862f4a85
PS C:\Users\Devanshi\Desktop\Lab_21_DevSecOps\nginx-html-app> |
```