# School of Computer Science

# UNIVERSITY OF PETROLEUM AND ENERGY STUDIES DEHRADUN, UTTARAKHAND



## Containers & Docker Security

## Lab File (2022-2026)
## 5th Semester

*Submitted To:*

**Dr. Hitesh Kumar**

**Sharma**

*Submitted By:*

*Akshat Pandey*
*(500101788)*
*B Tech  CSE*
*DevOps[5th Semester]*
*R2142220306*
*Batch - 1*

# EXPERIMENT 5

**AIM: 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
```

```
akshat-2004@UBUNTU:~$ mkdir nginx-html-app
akshat-2004@UBUNTU:~$ cd nginx-html-app
akshat-2004@UBUNTU:~/nginx-html-app$ touch index.html
akshat-2004@UBUNTU:~/nginx-html-app$
```

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>
```

```
  GNU nano 7.2                                    index.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
```

```
akshat-2004@UBUNTU:~/nginx-html-app$ touch Dockerfile
akshat-2004@UBUNTU:~/nginx-html-app$ nano Dockerfile
```

Edit the Dockerfile and add the following content:

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

```
  GNU nano 7.2                                    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.

```
\aksha\

C:\Users\aksha\nginx-html-app>docker build -t nginx-html-app .
[+] Building 0.2s (7/7) FINISHED                                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                                      0.0s
 => => transferring dockerfile: 104B                                                      0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                           0.0s
 => [internal] load .dockerignore                                                         0.0s
 => => transferring context: 2B                                                           0.0s
 => [internal] load build context                                                         0.0s
 => => transferring context: 266B                                                         0.0s
 => [1/2] FROM docker.io/library/nginx:latest                                             0.1s
 => [2/2] COPY index.html /usr/share/nginx/html/                                          0.0s
 => exporting to image                                                                    0.0s
 => => exporting layers                                                                   0.0s
 => => writing image sha256:3fc3bee893974e8c7238db38ddf3f224d406809b66b04809fb1c90705c0e1606  0.0s
 => => naming to docker.io/library/nginx-html-app                                         0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/t9kloyqvwpjtaf1w5yz1qdsk9

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
```

## 5. Step 4: Run the Docker Container

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

```
docker run -d -p 8080:80 nginx-html-app
```

This command runs the container in detached mode (-d) and maps port 8080 on your host machine to port 80 inside the container, where Nginx is serving your HTML app.

```
C:\Users\aksha\nginx-html-app>docker run -d -p 8080:80 nginx-html-app
38f6c88d364a261c3173de88aff3a58c4f137fac92e6e411514ccc07f6092365
```

## 6. Step 5: Verify

Open a browser and go to http://localhost:8080. You should see your HTML page with the message "Hello, Nginx Docker!".
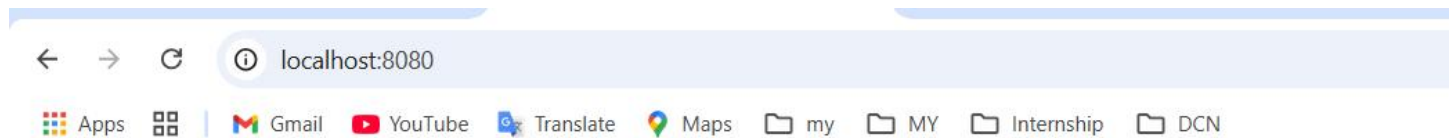
# Hello, Nginx Docker!

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\aksha\nginx-html-app>docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED          STATUS          PORTS
38f6c88d364a   nginx-html-app   "/docker-entrypoint.…"   57 seconds ago   Up 57 seconds   0.0.0.0:808
_shirley

C:\Users\aksha\nginx-html-app>docker stop38f6c88d364a
docker: 'stop38f6c88d364a' is not a docker command.
See 'docker --help'

C:\Users\aksha\nginx-html-app>docker stop 38f6c88d364a
38f6c88d364a

C:\Users\aksha\nginx-html-app>docker rm 38f6c88d364a
38f6c88d364a
```