

Placement Empowerment Program

Cloud Computing and DevOps Centre

BUILD AND RUN A CUSTOM DOCKER IMAGE

(Create a Docker file to package your static website into a Docker container and run it locally)

NAME: NIDHISHA A DHAS

DEPARTMENT: AML

INTRODUCTION:

Containerization has revolutionized the way applications are developed, deployed, and managed. Docker provides a lightweight and efficient way to package applications with all dependencies, ensuring consistency across different environments. In this POC, we will build and run a custom Docker image that serves a simple HTML web page using NGINX.

OBJECTIVE:

This POC demonstrates how to:

- Create a Docker static website using NGINX.
- Build a custom Docker image with an HTML file.
- Run a Docker container to serve the HTML content.
- Access the webpage via `http://localhost:1802`.

By the end of this exercise, you will have a functional web server running inside a Docker container.

STEP BY STEP OVERVIEW:

Step 1: INSTALL DOCKER DESKTOP

- Download Docker Desktop for Windows from the official website.
- Check if docker is installed using the following command.

```
C:\Users\Arulldhas>docker --version  
Docker version 27.5.1, build 9f9e405
```

Step 2: CREATE PROJECT DIRECTORY

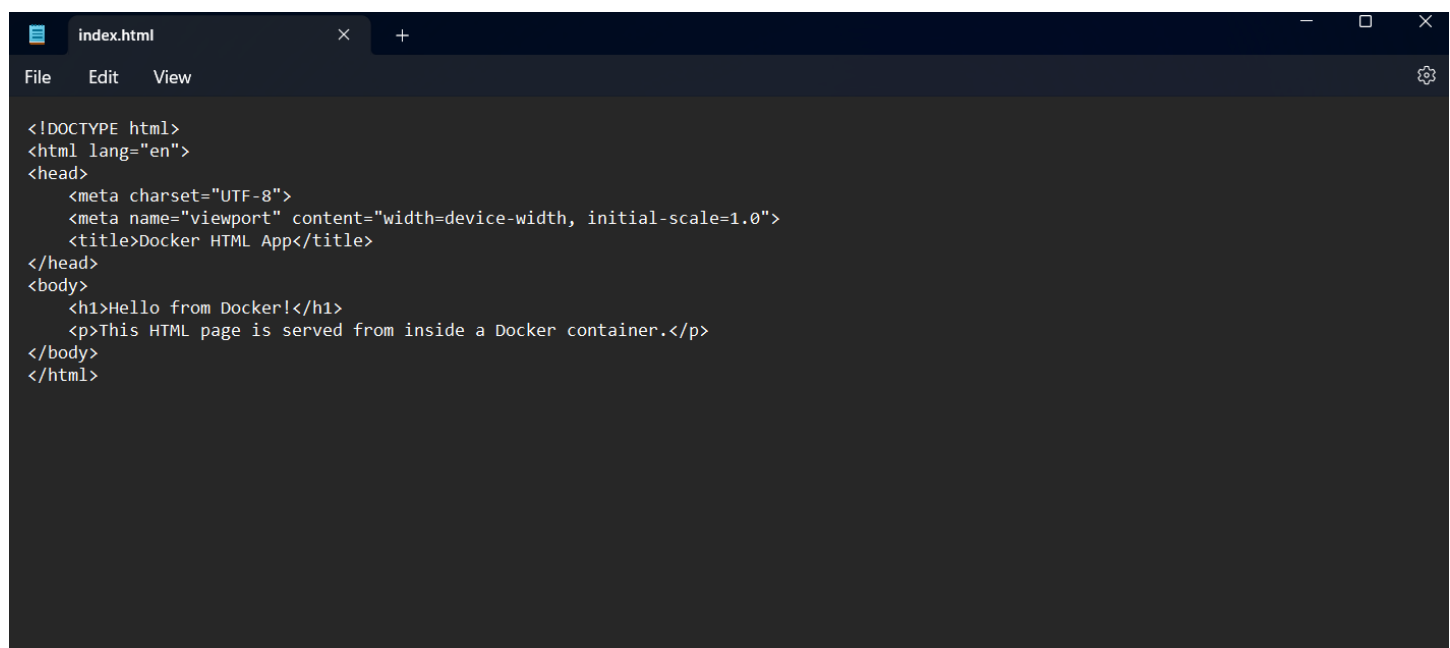
- Open Command Prompt and run the following command to create a Project Directory.
- The command 'mkdir my-docker-html' creates a new directory and 'cd my-docker-html' moves it inside the directory.

```
C:\Users\Arulldhas>mkdir my-docker-html && cd my-docker-html  
C:\Users\Arulldhas\my-docker-html>|
```

Step 3: CREATE AN HTML FILE

- To create an empty html file using the command prompt use the following commands:
- 'type nul > index.html' – creates a html file named 'index.html' and 'notepad index.html' opens the file in notepad.

```
C:\Users\Arulldhas\my-docker-html>type nul > index.html  
C:\Users\Arulldhas\my-docker-html>notepad index.html
```

A screenshot of a Notepad application window titled 'index.html'. The window has a dark theme and shows the following HTML code:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Docker HTML App</title>  
</head>  
<body>  
  <h1>Hello from Docker!</h1>  
  <p>This HTML page is served from inside a Docker container.</p>  
</body>  
</html>
```

Step 4: CREATE A DOCKER FILE

- Create a docker file via command prompt using the following command:
- ‘type nul> Dockerfile’. This creates an empty docker file.
- Now, use the command ‘notepad Dockerfile’ to open the file created in notepad.

```
C:\Users\Arulldhas\my-docker-html>type nul >Dockerfile  
C:\Users\Arulldhas\my-docker-html>notepad Dockerfile
```

```
File Edit View  
  
FROM nginx:latest  
  
COPY index.html /usr/share/nginx/html/index.html  
  
CMD ["nginx", "-g", "daemon off;"]
```

Step 5: BUILD DOCKER IMAGE

- To build a Docker image via command prompt, use the command ‘build -t my-html-image’.

```
C:\Users\Arulldhas\my-docker-html>docker build -t my-html-image .  
[+] Building 4.4s (8/8) FINISHED  
=> [internal] load build definition from Dockerfile  
=> => transferring dockerfile: 144B  
=> [internal] load metadata for docker.io/library/nginx:latest  
=> [internal] load .dockerignore  
=> => transferring context: 2B  
=> [internal] load build context  
=> => transferring context: 357B  
=> [1/2] FROM docker.io/library/nginx:latest@sha256:9d6b58feebd2dbd3c56ab585333d627cc6e281011cfd6050fa4bcf2072c9496  
=> => resolve docker.io/library/nginx:latest@sha256:9d6b58feebd2dbd3c56ab585333d627cc6e281011cfd6050fa4bcf2072c9496  
=> [auth] library/nginx:pull token for registry-1.docker.io  
=> [2/2] COPY index.html /usr/share/nginx/html/index.html  
=> exporting to image  
=> => exporting layers  
=> => exporting manifest sha256:150638ba559859053a07f3c585265a978442f605c3bdd3be2e86638a9fd23e0a  
=> => exporting config sha256:aac5af08de5bd38aa2de0f5a463d7dda8e722290b0ebd0e5c6a8877e30dc977  
=> => exporting attestation manifest sha256:e1c9404e2a095b5d5e48aa87b38c9c5cc0c1c631eb91d164d4b86b9b8dccb58  
=> => exporting manifest list sha256:6103103033b37d608c3e9f9eb174d1c49e50566d331382e73e642c17f6988057  
=> => naming to docker.io/library/my-html-image:latest  
=> => unpacking to docker.io/library/my-html-image:latest  
  
C:\Users\Arulldhas\my-docker-html>
```

Step 6: RUN THE DOCKER CONTAINER

- Run the following command to start the container from the image and expose it on port 1802.

```
C:\Users\Arulldhas\my-docker-html>docker run -d -p 1802:80 my-html-image  
9cc9db09b819b7ed7ee8250e25ebd40cfa808522daf439c2cf2227c73f79fc5e
```

Step 7: VIEW THE WEBPAGE

- Open any web browser and search for localhost:1802 and you will see your webpage.



EXPECTED OUTCOME

From this POC, we have learnt:

- Successful installation of Docker Desktop on Windows.
- Verification that Docker is running correctly through PowerShell commands.
- Build a docker image via command prompt and run the container.
- Accessing the Nginx default welcome page in a web browser at 'http://localhost:1802'
- Understanding basic Docker commands.