Placement Empowerment Program

Cloud Computing and DevOps Centre

Build and Run a Custom Docker ImageCreate a Dockerfile to package your static website into a Docker container and run it locally.

Name: Namrata choudhary Department: CSE

Introduction :

Docker is a powerful tool that allows developers to package applications and their dependencies into containers. In this task, we will create a custom Docker image for a static website, package it into a container, and run it locally using Docker. This approach ensures consistency, portability, and ease of deployment across different environments.

Overview :

This task involves setting up a simple static website, writing a **Dockerfile** to package it into a Docker image, and then running the image as a container. We will be using **Nginx**, a lightweight and high-performance web server, as the base image to serve our website. Once the container is running, we can access the website via a web browser on our local machine.

Objectives :

The key objectives of this task are:

- To understand the fundamentals of Docker and containerization.
- To create a static website and package it into a Docker container.
- To build a custom Docker image using a **Dockerfile**.
- To run the website locally in a Docker container using **Nginx**.
- To expose the website on a specific port for local access.

Importance

1. Foundation for Cloud and DevOps: Learning Docker is essential for modern cloud-based and DevOps workflows.

2. Portability & Efficiency: Containers eliminate the "works on my machine" problem by ensuring consistency across different environments.

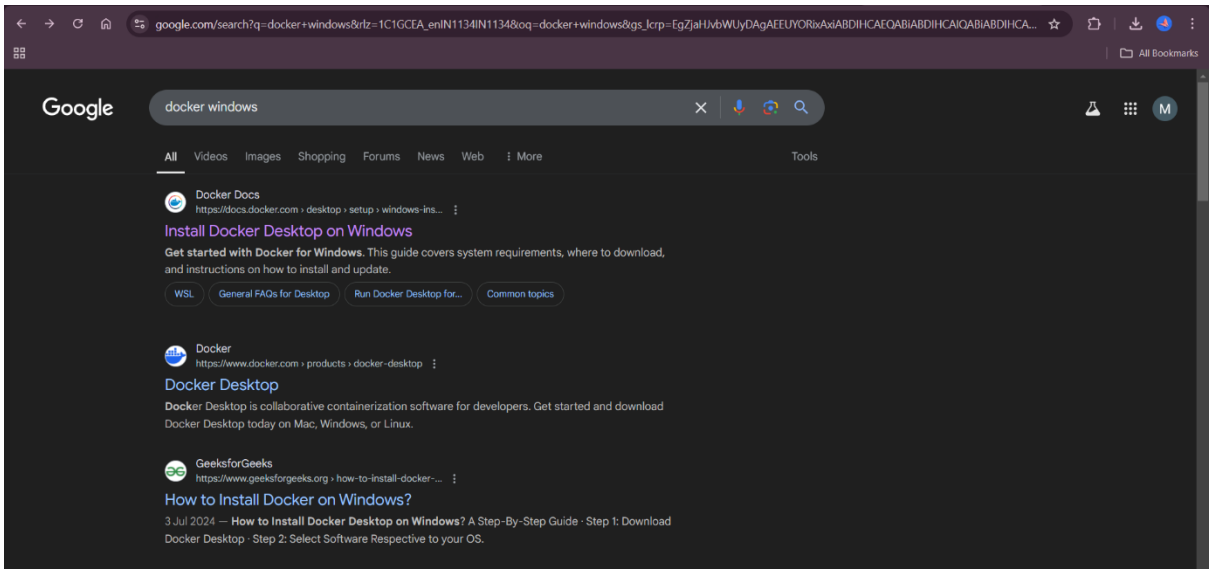
3. Scalability & Deployment: Running Nginx in a container helps in understanding web server deployments, a key aspect of cloud infrastructure.

4. Hands-on Experience: This POC provides practical knowledge applicable in real-world cloud projects, CI/CD pipelines, and Kubernetes deployments.

Step-by-Step Overview

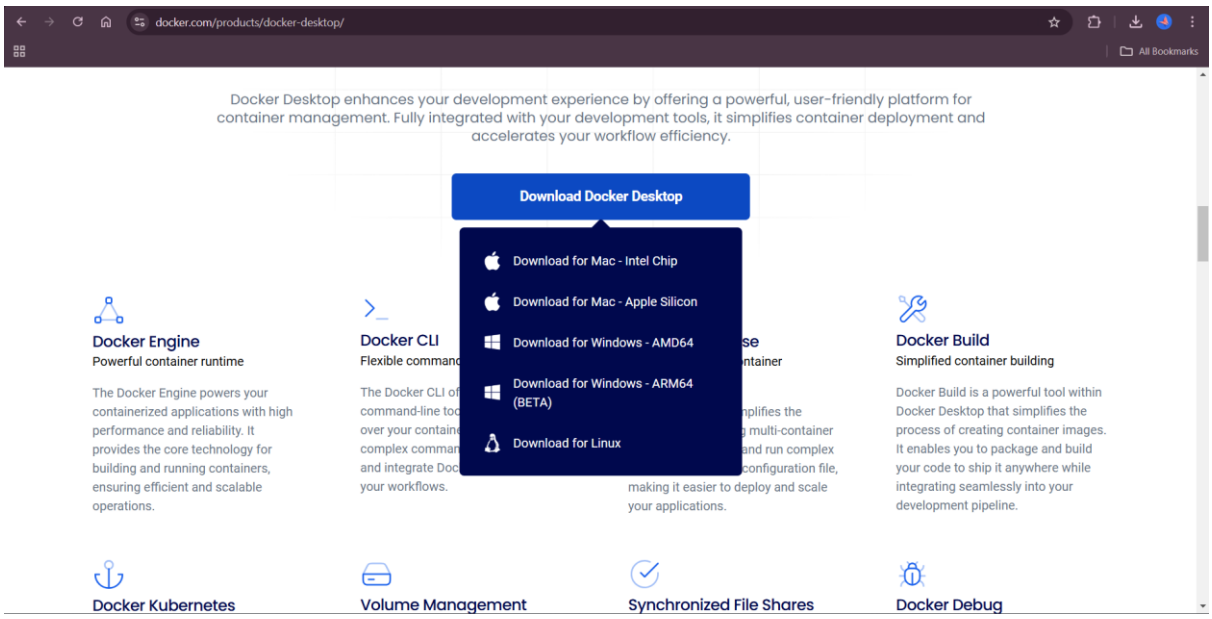
Step 1:

In Google Search for **Docker windows Download**.



Step 2:

Scroll Down and **Download Docker Desktop**. Complete the installation process.



Step 3:

Open Command Prompt and run the commands:

cd C:\Users\YourUsername\Documents
mkdir my-static-website
cd my-static-website

Once inside the my-static-website directory, you can proceed with creating your **index.html** and **Dockerfile**.

```
C:\Windows\system32\cmd.e:

Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd C:\Users\user\Documents

C:\Users\user\Documents>mkdir my-static-website

C:\Users\user\Documents>cd my-static-website

C:\Users\user\Documents\my-static-website>
```

Step 4:

Create the file

You can create it manually or use the following command:

echo "" > index.html and code .

```
C:\Users\user\Documents\my-static-website>echo "" > index.html

C:\Users\user\Documents\my-static-website>code .
```

Step 5 :

Write the following html code in index.html

Copy and paste the following HTML code inside index.html:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Dockerized Website</title>
</head>
<body>
    <h1>Welcome to My Website!</h1>
    <p>Running inside a Docker container.</p>
</body>
</html>
```

Step 6:

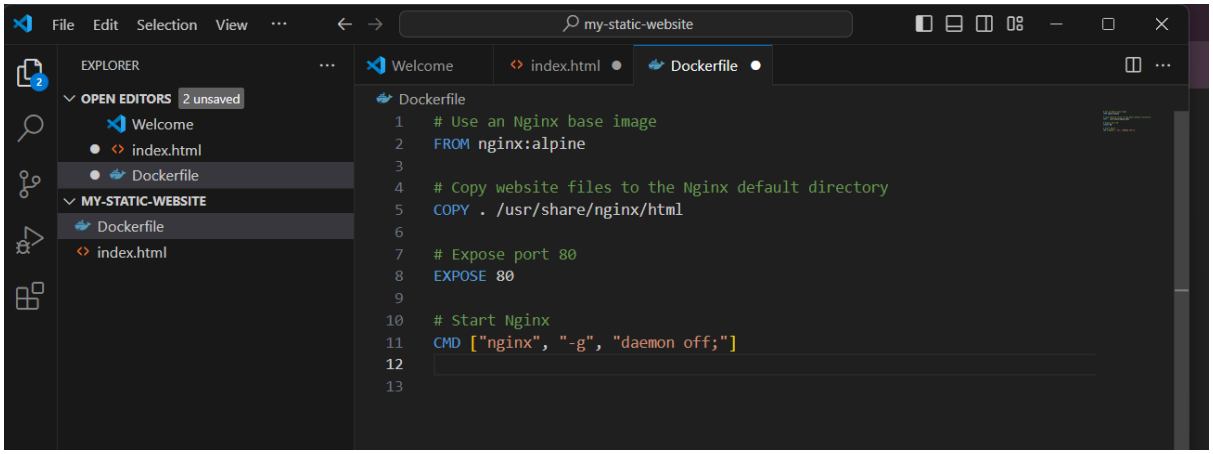
Create a new file named Dockerfile and run these commands in docker:

echo. > Dockerfile

code Dockerfile

```
C:\Users\user\Documents\my-static-website>echo. > Dockerfile

C:\Users\user\Documents\my-static-website>code Dockerfile
```



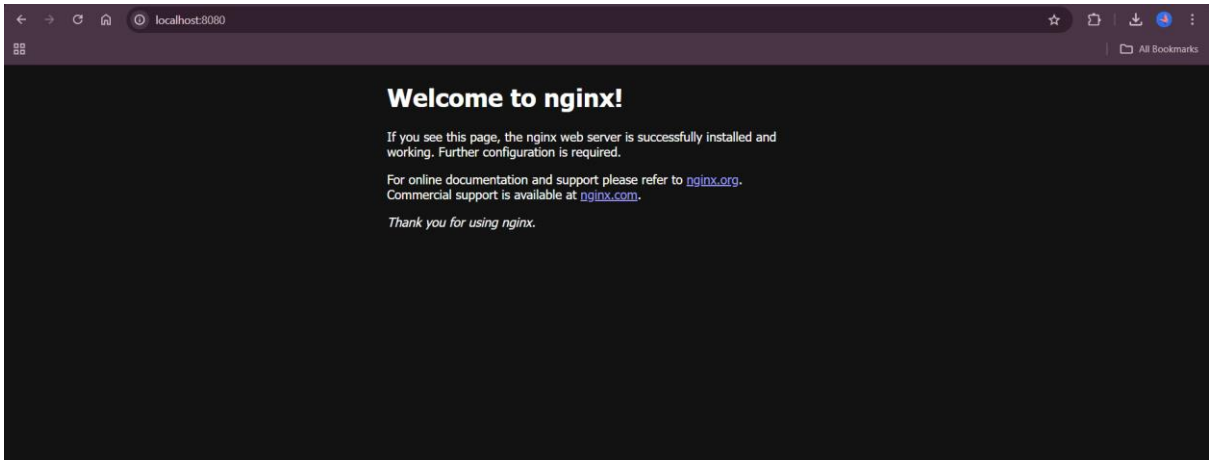
Step 7:

Test Accessing Nginx:

1. Open a browser and go to:

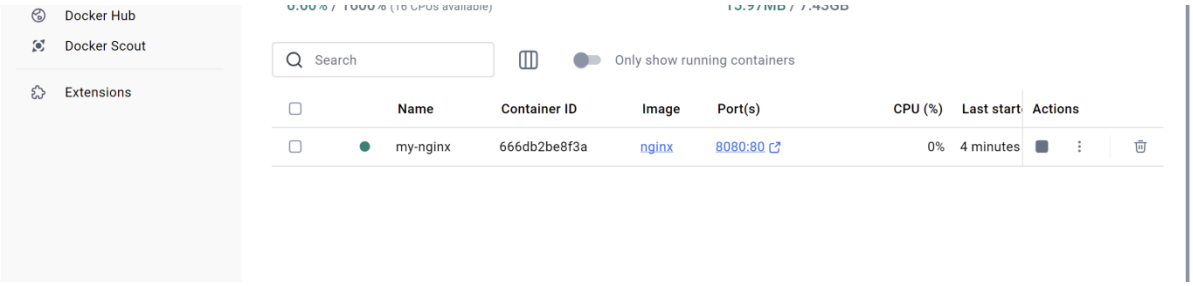
   http://localhost:8080

2. You should see the default Nginx welcome page.



Step 8:

By Opening Docker Desktop App We can see our container running.



Step 9:

Stop and Remove the Container

1. Stop the container:

   docker stop my-nginx

2.Remove the container:

docker rm my-nginx

```
C:\Users\Hi>docker stop my-nginx
my-nginx

C:\Users\Hi>docker rm my-nginx
my-nginx
```

You have successfully installed Docker, run your first Nginx container, and tested it!

Outcomes

By completing this POC, you will:

1. **Install and Configure Docker:** Learn to set up Docker on Windows and prepare the environment for containerized applications.

2. **Pull and Run an Nginx Container:** Gain hands-on experience in downloading and deploying a web server using Docker.

3. **Expose and Access the Web Server:** Understand how to map ports and access the running Nginx container via a browser.

4. **Manage and Monitor Containers:** Learn essential Docker commands to start, stop, inspect, and remove containers efficiently.

5. **Understand the Benefits of Containerization:** Explore how Docker simplifies application deployment, enhances scalability, and streamlines DevOps workflows.