Session 01: Explore Docker Hub using pre-built Docker images

Pre-Lab:

1. Name any pre-built Docker image available on Docker Hub.

Examples of pre-built Docker images:

- nginx
- httpd
- node
- python
- mysql

2. Which command is used to download (pull) an image from Docker Hub?

docker pull <image-name>

Example: docker pull httpd

3. What is the purpose of Docker Hub?

Docker Hub is a **cloud-based container registry** that allows users to:

- Store Docker images
- Share images publicly or privately
- Download official and community images
- Collaborate on containerized applications

In-Lab Task:

1) Install Docker, Docker Desktop and Creating an Account in Docker Hub

Steps

1. Install **Docker Desktop** from official Docker website
2. Verify installation:

docker --version

docker info

3. Create an account on **Docker Hub**
4. Login from terminal:

docker login

2) Browse Docker Hub for appropriate images to host a website. Download these images to your development environment and experiment with them. Deploy a website in a container using the httpd (Apache HTTP Server) image

Open Docker Hub website

Search for:

- httpd
- nginx

Observe:

- Image description
- Number of downloads
- Official image badge

Deploy Website using Apache (httpd)

Apache HTTP Server Docker image name: httpd

Step 1: Pull the Image

```
docker pull httpd
```

Step 2: Create a Custom HTML File

```
<!-- index.html -->
<html>
<head><title>Apache in Docker</title></head>
<body>
<h1>Welcome to Apache in Docker!</h1>
</body>
</html>
```

Step 3: Create Dockerfile

```
FROM httpd
COPY index.html /usr/local/apache2/htdocs/
EXPOSE 80
```

Step 4: Build & Run Container

docker build -t apache-web .

docker run -d -p 8080:80 apache-web

Step 5: Output Verification

Open browser:

http://localhost:8080

Expected Output:

Welcome to Apache in Docker!

Post Lab Task:

1) Browse Docker Hub for images that can host a website and pull them into your development setup for experimentation. Deploy a website using the 'Nginx' image in a container

Nginx is a high-performance, lightweight web server commonly used for hosting static websites.

Docker Hub provides an **official Nginx image** that allows instant deployment without manual installation.

Docker Hub is a cloud-based registry used to store and distribute Docker images.

Step 1: Browse Docker Hub

1. Open Docker Hub website
2. Search for **nginx**
3. Select the **official Nginx image**
4. Observe:
    - Image description
    - Pull command
    - Tags and versions

Step 2: Pull Nginx Image

docker pull nginx

This command downloads the Nginx image from Docker Hub to the local Docker environment.

Step 3: Verify Image Download

docker images

The nginx image should appear in the image list.

Step 4: Create Project Directory

```
mkdir nginx-website
cd nginx-website
```

Step 5: Create Website File

Create a file named **index.html**

```
<!DOCTYPE html>
<html>
<head>
   <title>Nginx Docker Website</title>
</head>
<body>
   <h1>Website Hosted Using Nginx Docker Container</h1>
   <p>This page is served from a Docker container.</p>
</body>
</html>
```

Step 6: Run Nginx Container

```
docker run -d -p 8081:80 -v C:\Users\NAGAMANI\nginx-website\index.html:/usr/share/nginx/html/index.html --name nginx-web nginx
```

Step 7: Verify Running Container

```
docker ps
```

nginx-web container should be in **running** state.

Step 8: Access Website

Open browser and enter:

http://localhost:8081

Output