Name: Kaushal Kotkar Department: Cyber Security

Div: BE-15 Roll No: 54

Subject: DSO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment No. – 5** | | | | |
| **Date of Performance:** | 12/08/2024 | | | |
| **Date of Submission:** | 19/08/2024 | | | |
| Program Execution/ formation/ correction/  ethical practices (06) | Timely Submission  (01) | Viva (03) | Experiment Total (10) | Sign with Date |
|  |  |  |  |  |

**Experiment No. 5**

**Aim:** To create a custom Docker image using Docker files and upload it to the Docker hub.

**Lab Outcome:** Analyze &amp; Illustrate the Containerization of OS images and deployment of applications over Docker

**Theory:**

**Docker:**

Docker is a platform that enables you to develop, ship, and run applications within containers, offering a lightweight and consistent environment that can host various applications and even different operating systems.

**Why Use Docker:**

Docker simplifies application deployment by packaging software with all its dependencies into a single container. It provides consistency, isolation, and portability across various environments.

**Features:**

Containerization: Encapsulates applications and their dependencies.

Portability: Containers run consistently across various environments.

Efficiency: Shares host OS resources, minimizing overhead.

Isolation: Ensures applications don't interfere with each other.

Scalability: Easily scale applications by launching multiple containers.

Version Control: Containers can be versioned for reproducibility.

**Why Use WSL:**

WSL integration enhances performance and compatibility by providing a Linux-compatible environment on Windows.

It allows you to work with Docker commands directly from a Linux terminal, improving the Docker experience on Windows.

**Steps to Create a Container:**

**Step 1: Install Docker Desktop**

Download and install Docker Desktop from the official Docker website.

Follow the installation instructions for your operating system.

**Step 2: Setup WSL**

1. Install and Update WSL:

Open the command prompt (CMD) or PowerShell as an administrator.

Run the following command to install WSL 2: **wsl --install**

After the installation, reboot your system.

Once your system reboots, open the command prompt (CMD) or WSL terminal again.

1. Update WSL:

To ensure you have the latest version of WSL, run: **wsl –update**

1. Enable WSL Integration:

Open Docker Desktop settings.

Navigate to the "Resources" section and select "WSL Integration."

Enable the WSL integration for the desired WSL distribution.

**Step 3: Pull Image**

Open the command prompt (CMD) or WSL terminal.

Run the following command to pull the "alpine" image from Docker Hub: docker pull alpine

**Step 4: Check Images**

To verify that the "alpine" image has been downloaded, run: **docker images**

**Step 5: Run Container**

Create and enter an "alpine" container using the following command: **docker run -it alpine**

The -it flag allows you to interact with the container using the terminal.

This command starts a new instance of the "alpine" image and opens a terminal within the container.

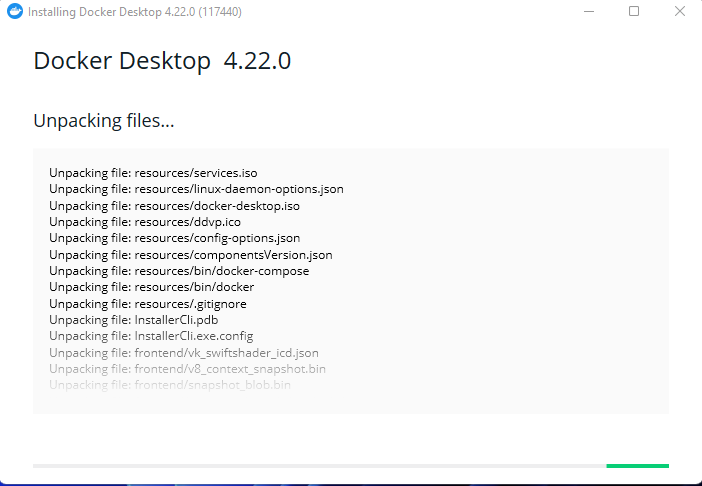
**Step 6: Explore the Container**

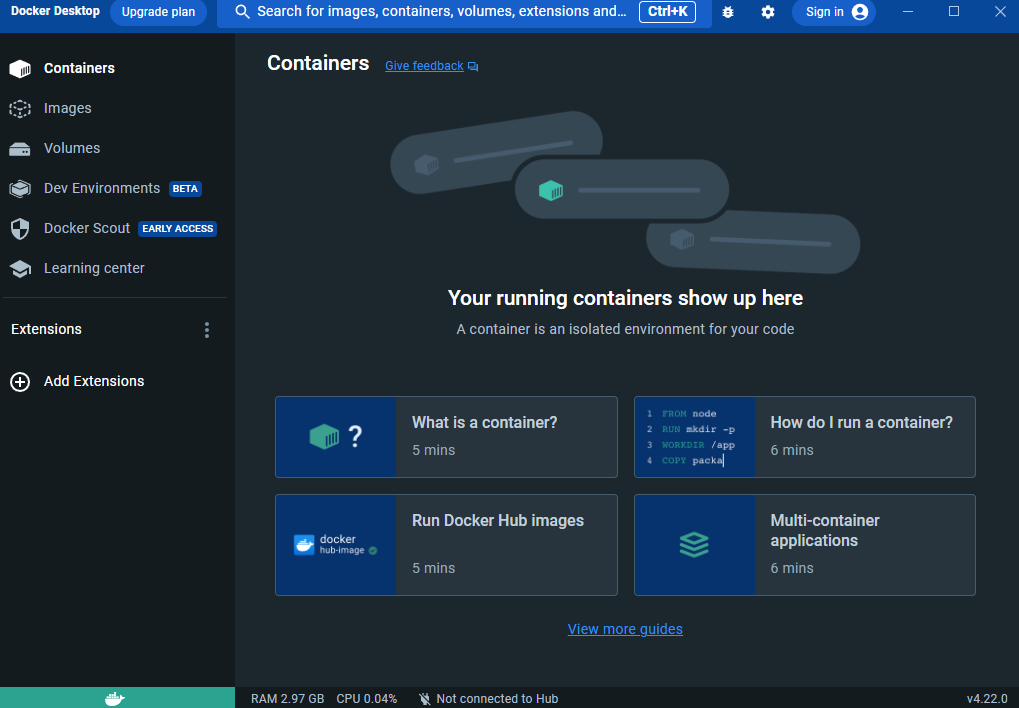
You'll now be inside the "alpine" container's terminal.

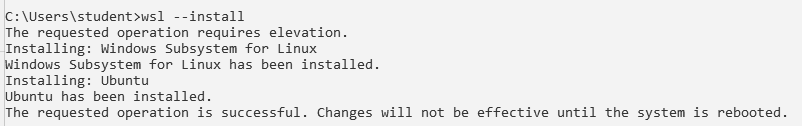
You can explore the container, run commands, and interact with the isolated environment.

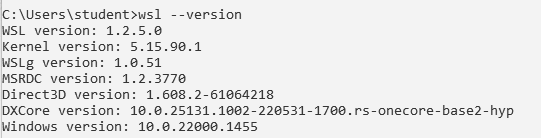
For example, you can try running commands like ls, echo, etc.

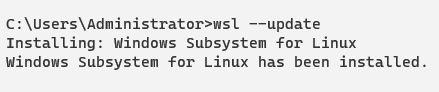
**Output:**

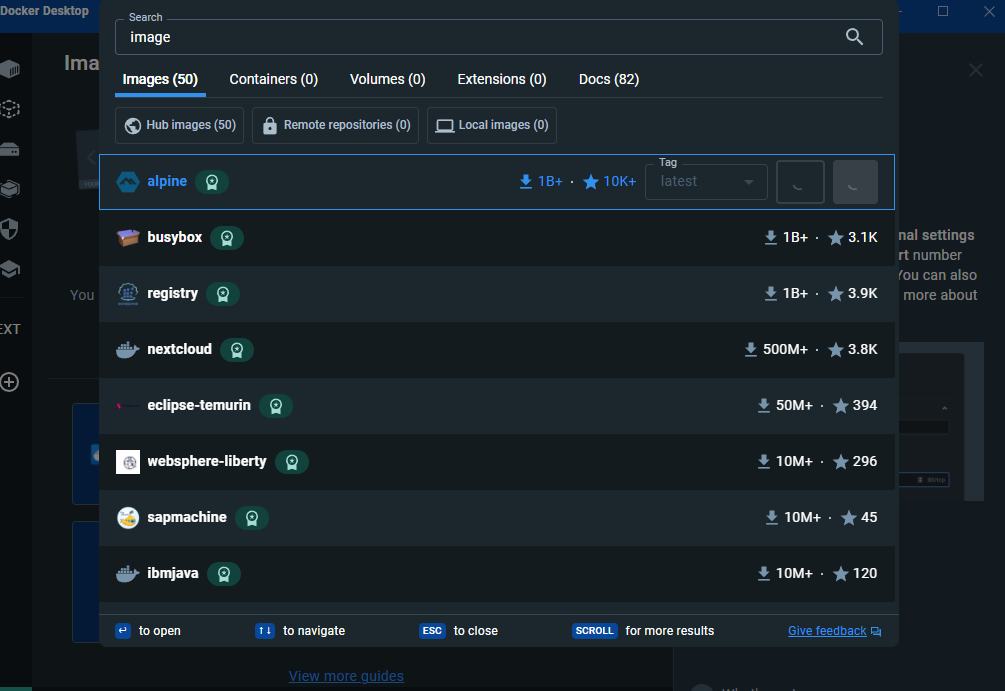


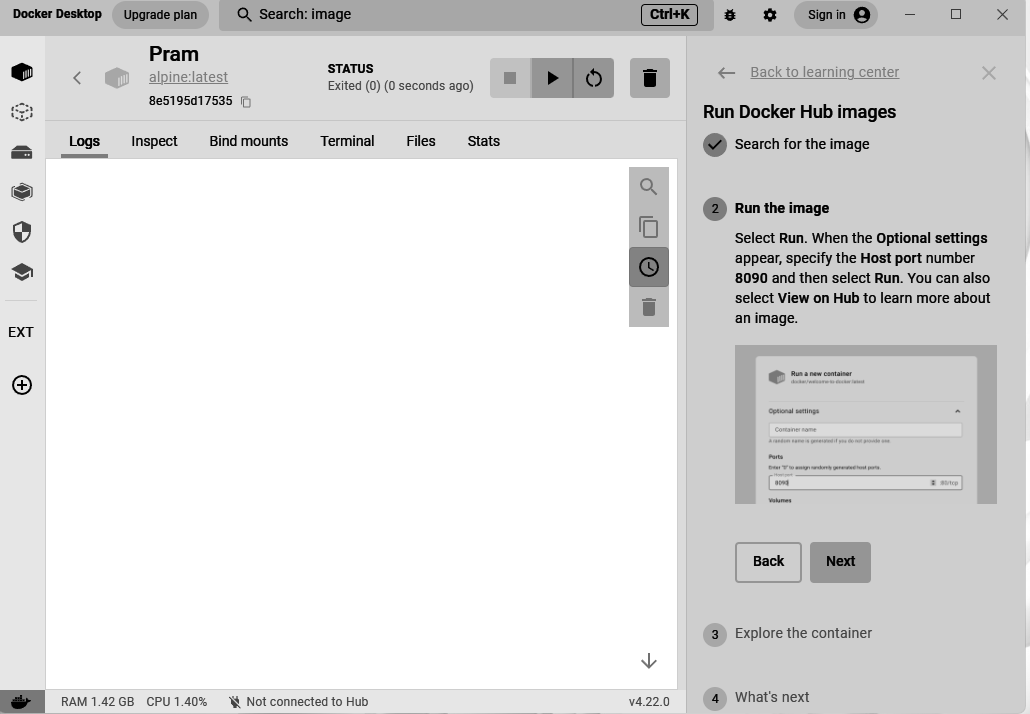


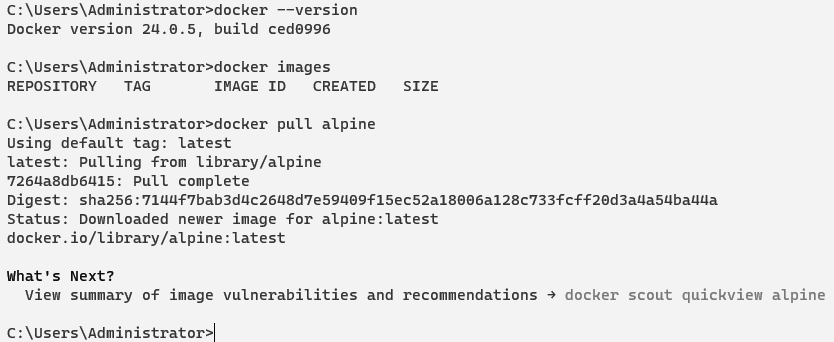


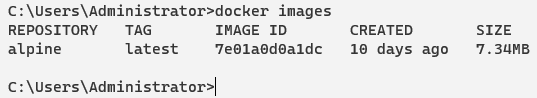


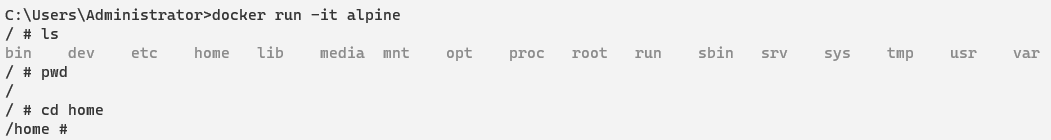












**Conclusion:**

Leveraging Docker to run containers for different applications and OS offers a flexible and efficient way to manage and deploy software, enhancing development workflows and cross-platform compatibility.