Name: Kaushal Kotkar Department: Cyber Security

Div: BE-15 Roll No: 54

Subject: DSO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment No. – 6** | | | | |
| **Date of Performance:** | 19/08/2024 | | | |
| **Date of Submission:** | 26/08/2024 | | | |
| Program Execution/ formation/ correction/  ethical practices (06) | Timely Submission  (01) | Viva (03) | Experiment Total (10) | Sign with Date |
|  |  |  |  |  |

**Experiment No. 6**

**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.

**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.

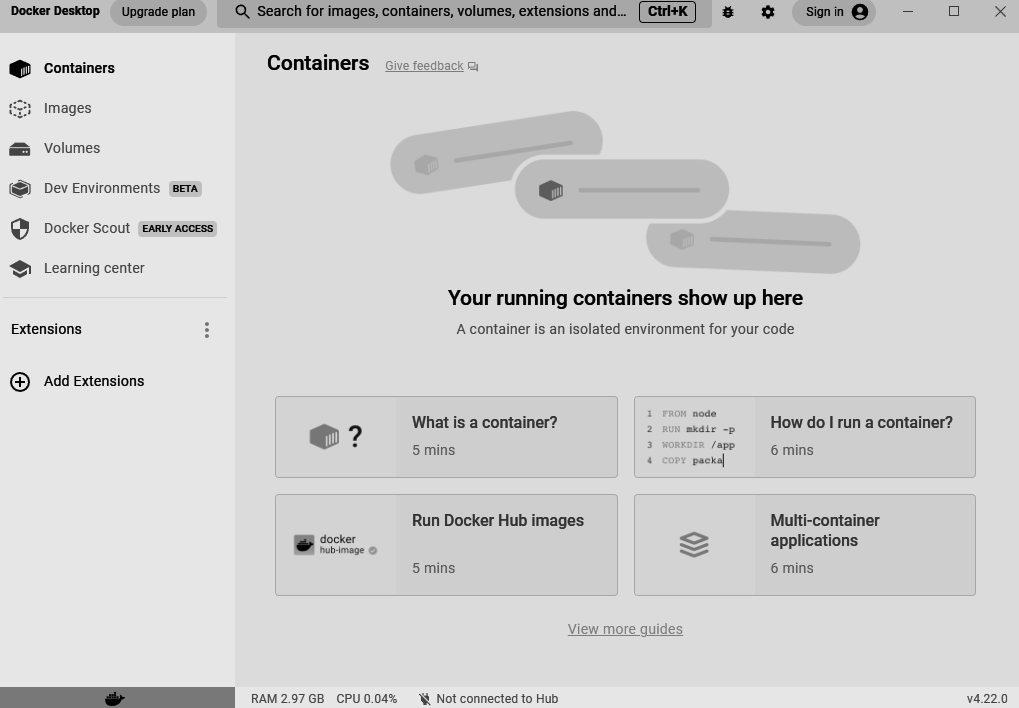
**Container:**

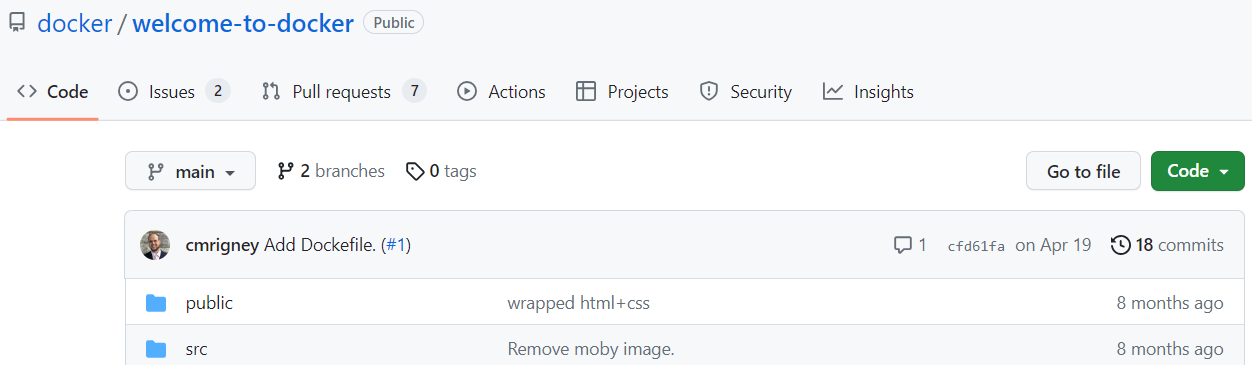
A container is a lightweight, standalone, and executable package that includes everything needed to run a piece of software, including the code, runtime, libraries, and system tools. Containers provide consistency and portability, making it easy to deploy applications across different environments.

**Steps Followed:**

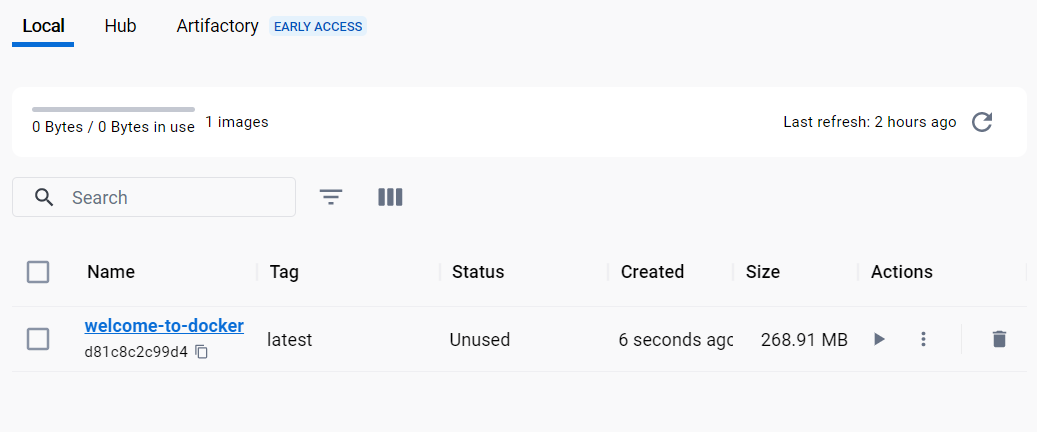
1. **Install Docker:**
   * Download and install Docker Desktop from the official Docker website based on your operating system.
   * Follow the installation instructions to set up Docker.
2. **Download a Project from GitHub:**
   * Choose a project from GitHub that contains a Dockerfile, which defines the container environment.
3. **Build a Docker Image:**
   * Open a terminal and navigate to the directory containing the Dockerfile.
   * Run the command to build a Docker image, e.g., **docker build -t welcome-to-docker.**
   * This command creates an image based on the instructions in the Dockerfile.
4. **Check the Built Image:**
   * Go to the Docker Desktop's Image tab to verify that the image has been successfully built.
5. **Run the Container:**
   * Start a container based on the image by specifying a container name and port.
   * Runs the application in the container and maps port 3000 on your local machine to port 80 in the container.
6. **Access the Application:**
   * Open a web browser and go to **localhost:3000**. You should see your application running in the container.
7. **Remove the Container:**
   * Once you're done, you can stop and remove the container by going to the Docker Desktop's Container tab.

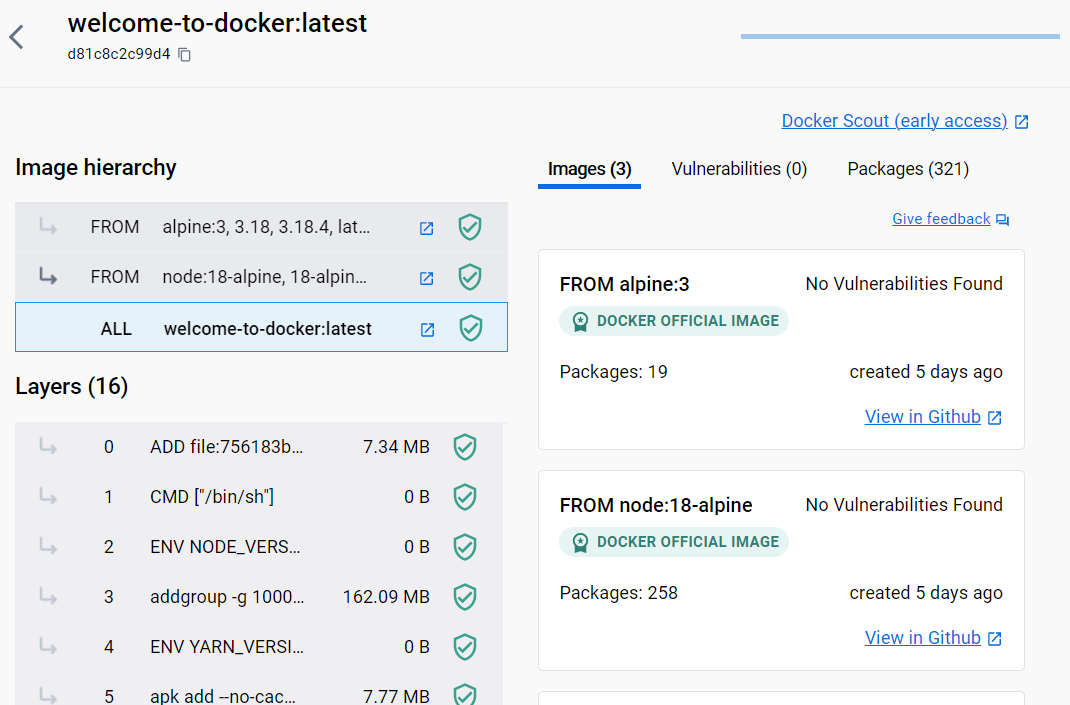
**Output:**

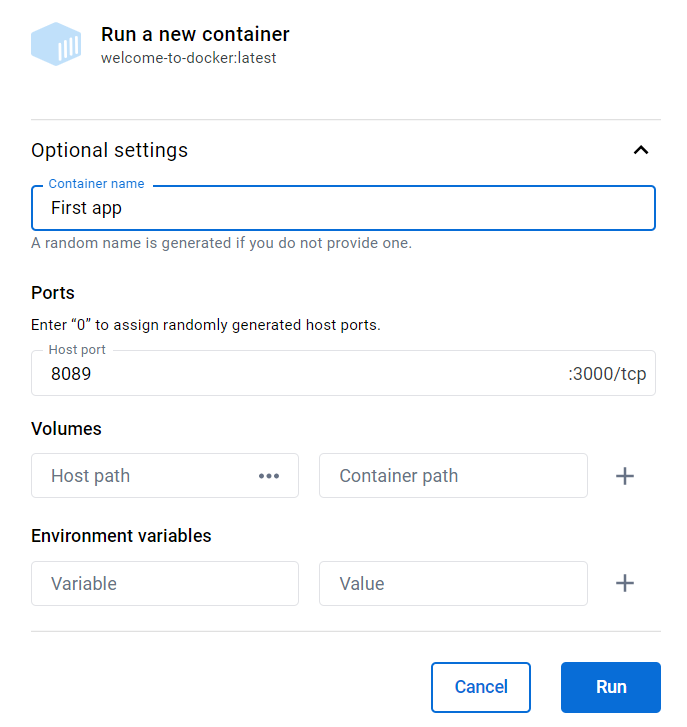


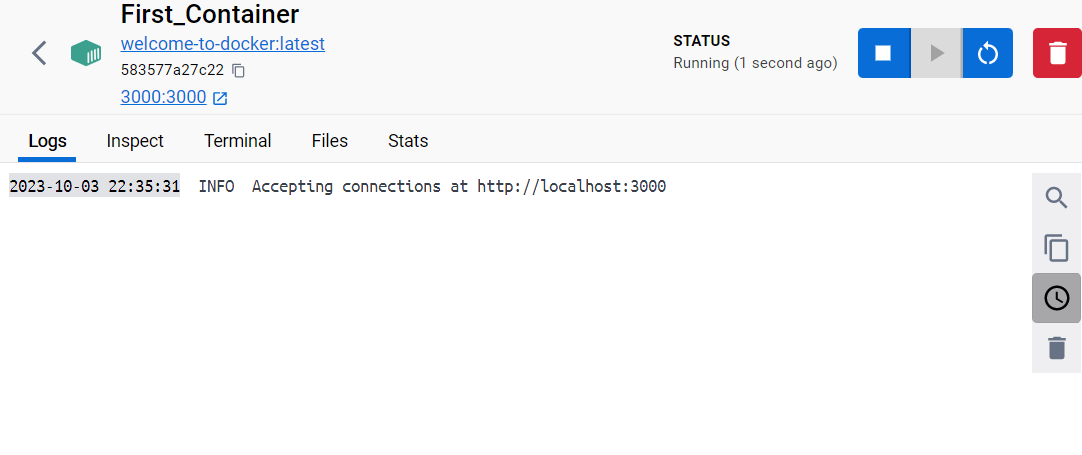


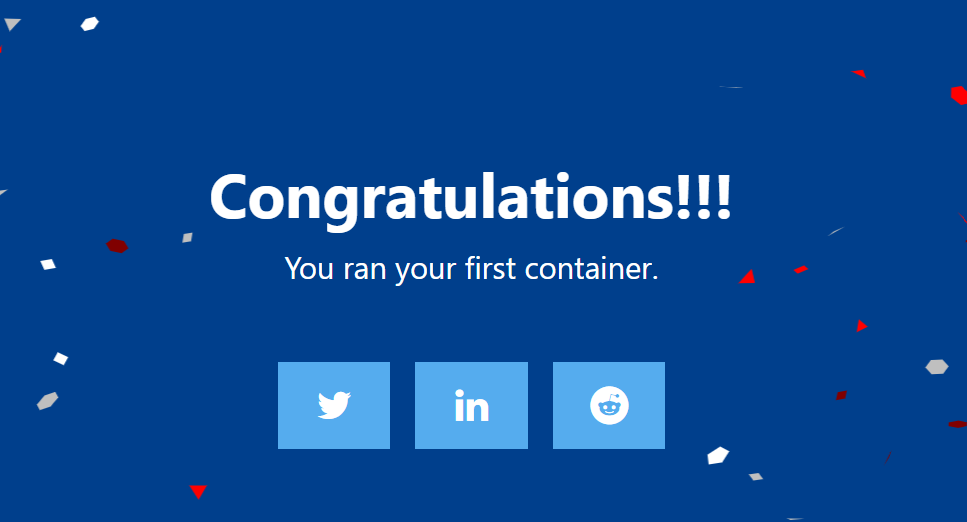


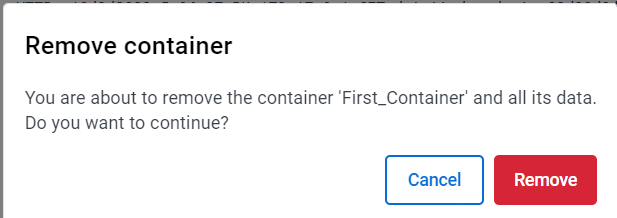












**Conclusion:**

In this experiment, we successfully utilized Docker to deploy containers running various applications and operating systems. Docker's containerization technology proved invaluable in achieving consistency and portability across different environments. By following the steps outlined, we gained practical experience in efficiently managing and running applications within containers, highlighting the advantages of this approach in modern software development and deployment.