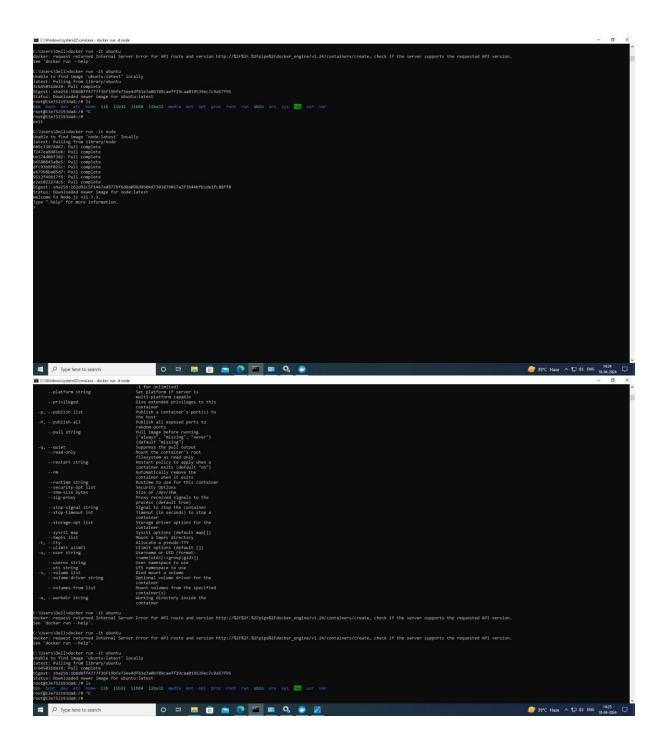
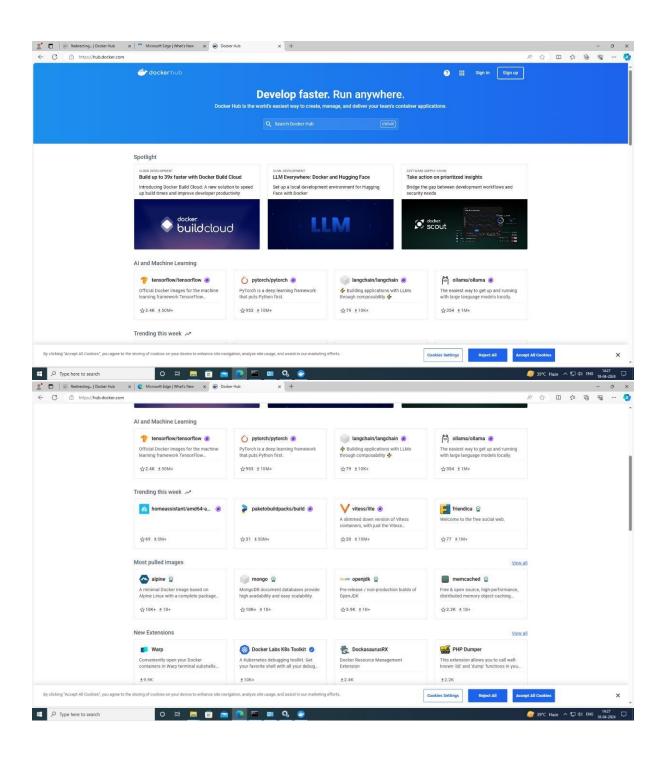
Name: Kaveri Vinod Shinde

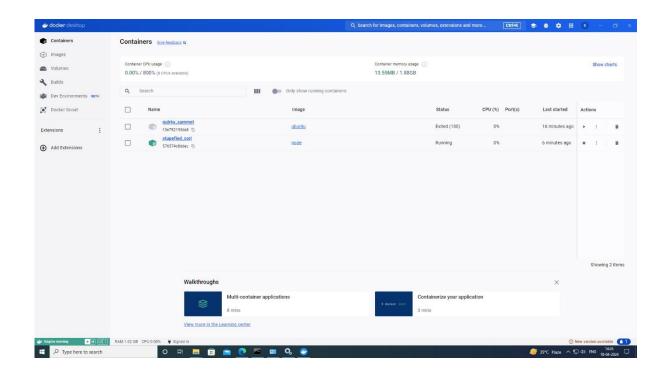
Roll No: 59

**Docker** 

## **Screenshots:**







## **Theory:**

Docker, an open-source software platform launched in 2013, has transformed the landscape of application development by popularizing software containers. These containers are standardized units that encapsulate an application's code and its dependencies, ensuring consistent and reliable performance across various computing environments. Unlike traditional virtual machines (VMs), which include a full operating system copy and abstract physical hardware, containers operate at the application layer. They share the host operating system's kernel, running as isolated processes, making them more portable and efficient.

At the core of Docker is the Docker Engine, initially introduced as an open-source tool and later expanded to support Microsoft Windows and Apple OS X. This engine comprises a server-side daemon responsible for managing images, containers, networks, and storage volumes. Additionally, it offers a client-side command-line interface (CLI) for users to interact with the daemon. Docker provides various components and tools to facilitate container creation, verification, and management. Docker files define container compositions, while Docker Compose files handle multi-container applications.

Despite its advantages, including streamlined development, portability, and scalability, Docker presents challenges, especially concerning security. Containers, while logically isolated, share the host's operating system. A vulnerability or attack on the underlying OS could potentially compromise all containers running on it. To address these concerns, Docker continuously integrates security enhancements into its platform, and numerous container security scanning tools have emerged to bolster protection.

In recent years, Kubernetes has emerged as the predominant container orchestration platform, often utilizing Docker as its container runtime. While Kubernetes enhances container management and scalability, security remains paramount. Organizations can mitigate risks by employing containers within VMs, opting for lower-profile VMs, or ensuring that container hosts remain isolated from the internet and only use images from trusted sources.