Assignment: 7

Name: Divya Shah Branch: IT/V Roll No.: 115 Date:26/08/2023

Aim: To understand Docker architecture and container life cycle, install dockers, deploy container in Docker.

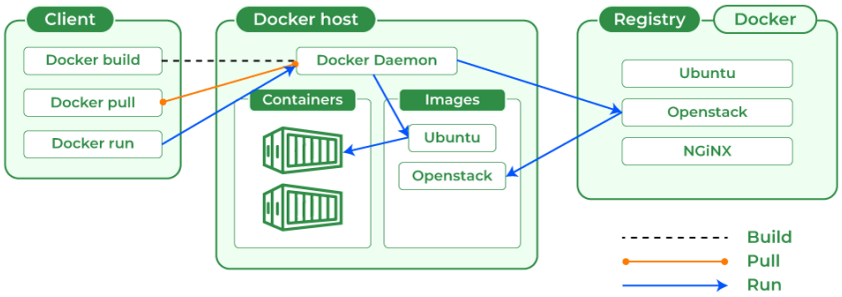
LO mapped: LO1, LO5

Theory:

- **Introduction to Docker-**

In the dynamic realm of software development and deployment, Docker has emerged as a game-changing technology. Docker's containerization approach has streamlined the way applications are packaged, distributed, and executed. It eliminates compatibility issues by encapsulating an application and its dependencies into a single, portable unit called a container. This assignment explores Docker's architecture, provides installation guidance, and guides you through the deployment of your own containers. By the end, you'll have a solid understanding of Docker's core concepts and practical skills for efficient application deployment.

- **Docker Architecture**



Architecture of Docker

Docker's architecture is the backbone of its containerization technology, enabling the efficient creation and management of containers. Understanding Docker's architecture is crucial for harnessing its full potential. Let's delve into the key components that make up Docker's architecture:

1. Docker Engine:
   - At the core of Docker is the Docker Engine, which is responsible for creating and running containers. It includes:
   - Docker Daemon: This background service manages containers. It listens for Docker API requests and takes care of container operations.

- Docker Client: The command-line tool that allows users to interact with the Docker Daemon. Users issue commands to the Docker Client, which in turn communicates with the Docker Daemon.

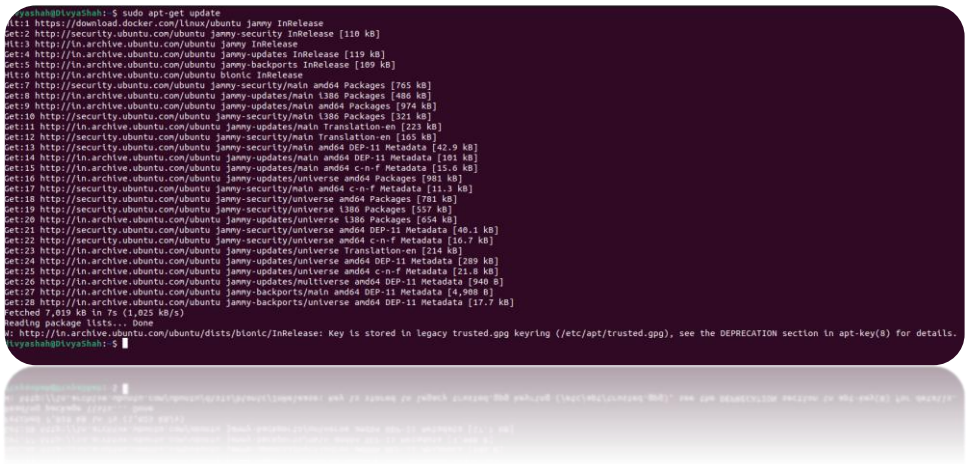
2. Docker Images:
   - Docker containers are based on Docker Images. These images are read-only templates that contain everything needed to run an application, including the code, runtime, libraries, and environment variables. Images are the building blocks of containers and are often shared via Docker registries like Docker Hub.
3. Docker Containers:
   - Docker Containers are instances of Docker Images. They are lightweight, isolated environments where applications run. Containers can be started, stopped, paused, and deleted, providing a consistent and portable environment for applications.
4. Docker Registry:
   - Docker Registries are repositories for Docker Images. The most commonly used registry is Docker Hub, a public registry that hosts a vast collection of Docker Images. Organizations often set up private registries to store and share their custom images securely.

Understanding this architecture is essential as it forms the basis for working with Docker. In the following sections of this assignment, we will explore how to install Docker, interact with Docker containers, and deploy applications within these containers. This hands-on experience will solidify your grasp of Docker's architecture and its practical applications.

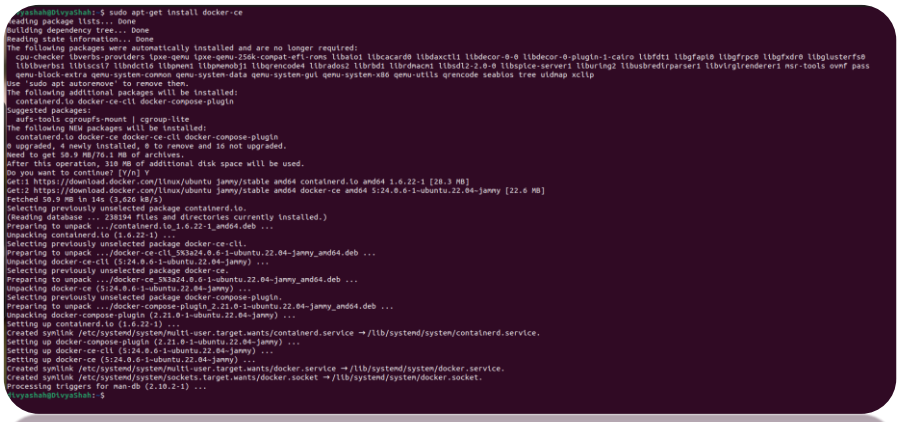
- **Installation of Docker**

  To install Docker on Windows, you can use Docker Desktop, which provides an easy way to set up and manage Docker containers on Windows 10 and Windows 11. Follow these steps to install Docker Desktop on Windows:

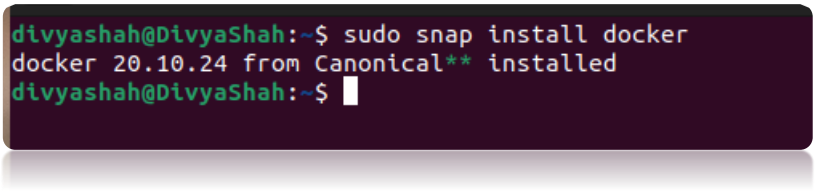
$$sudo\ apt - get\ update$$



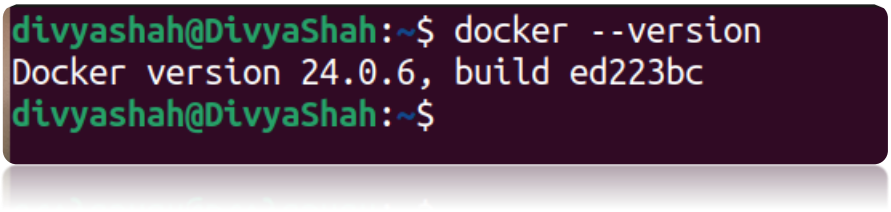
sudo apt – get install docker – ce



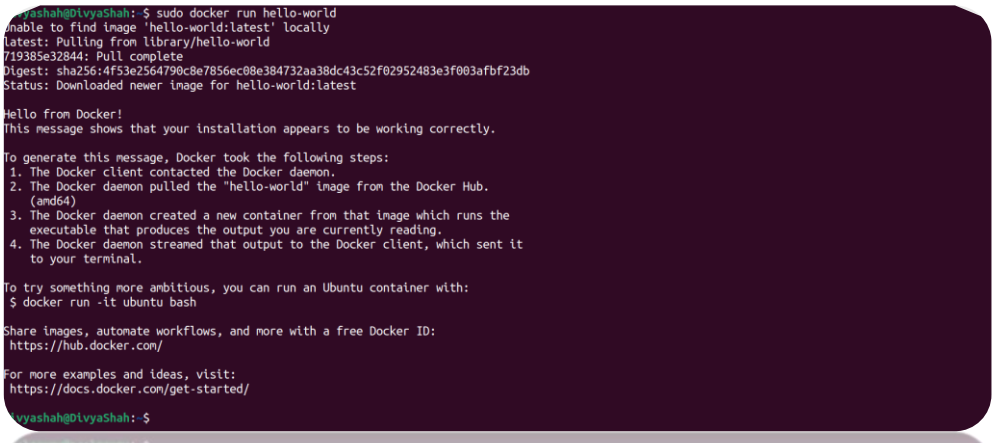
sudo snap install docker



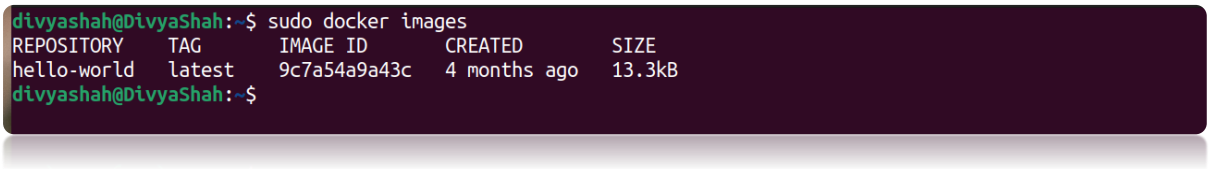
For checking docker version installed you can use



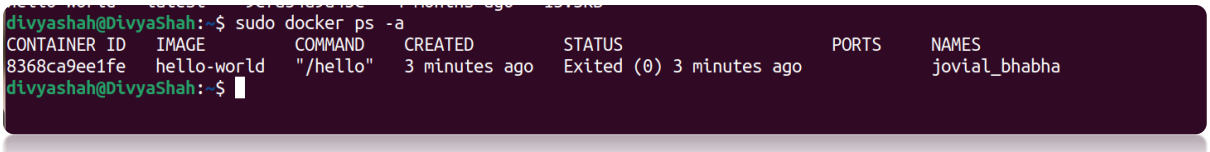
sudo docker run hello – world



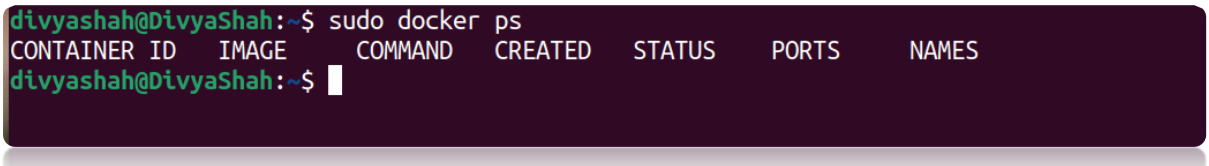
sudo docker images



sudo docker ps − a



sudo docker ps



Conclusion: By this assignment we understand Docker architecture and container life cycle, install dockers, deploy container in Docker.