Pandit Deendayal Energy University (PDEU)

Cloud Computing Lab (20CP322P)

B.Tech-Computer Science & Engineering (Sem-VI)

PATEL VEDANT H.

19BCP138

DIVISION – 2 (G4)

Experiment – 2

Docker

❖ **Aim:** To Explore **Docker Container**.

❖ Introduction:

Docker is a set of platforms as a service product that uses OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their software, libraries, and configuration files; they can communicate with each other through well-defined channels.

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

Docker works by providing a standard way to run your code. Docker is an operating system for containers. Similar to how a virtual machine virtualizes (removes the need to directly manage) server hardware, containers virtualize the operating system of a server. Docker is installed on each server and provides simple commands you can use to build, start, or stop containers.

Using Docker lets you ship code faster, standardize application operations, seamlessly move code, and save money by improving resource utilization. With Docker, you get a single object that can reliably run anywhere. Docker's straightforward syntax gives you full control. Wide adoption means there's a robust ecosystem of tools and off-the-shelf applications that are ready to use with Docker.

You can use Docker containers as a core building block creating modern applications and platforms. Docker makes it easy to build and run distributed micro services architectures, deploy your code with standardized continuous integration and delivery pipelines, build highly-scalable data processing systems, and create fully-managed platforms for your developers.

❖ Procedure:

Step 1: Go to (https://www.docker.com/get-started) and download Docker for your Operating System.
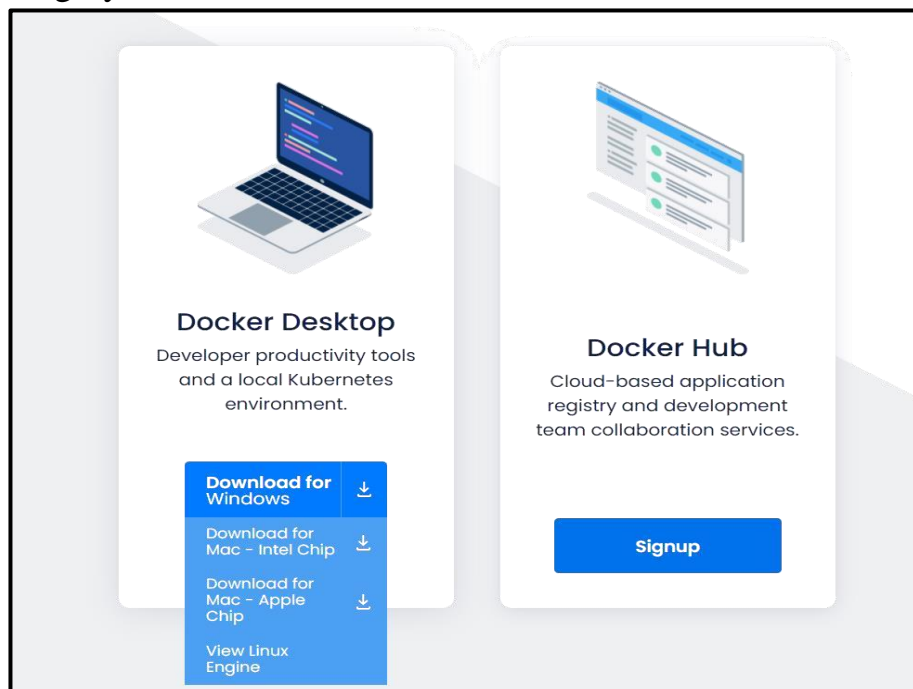
Step 2: Click on the downloaded .exe file to install Docker in your system. Follow the normal procedure to install app. As it's already installed in my system, it is showing this.
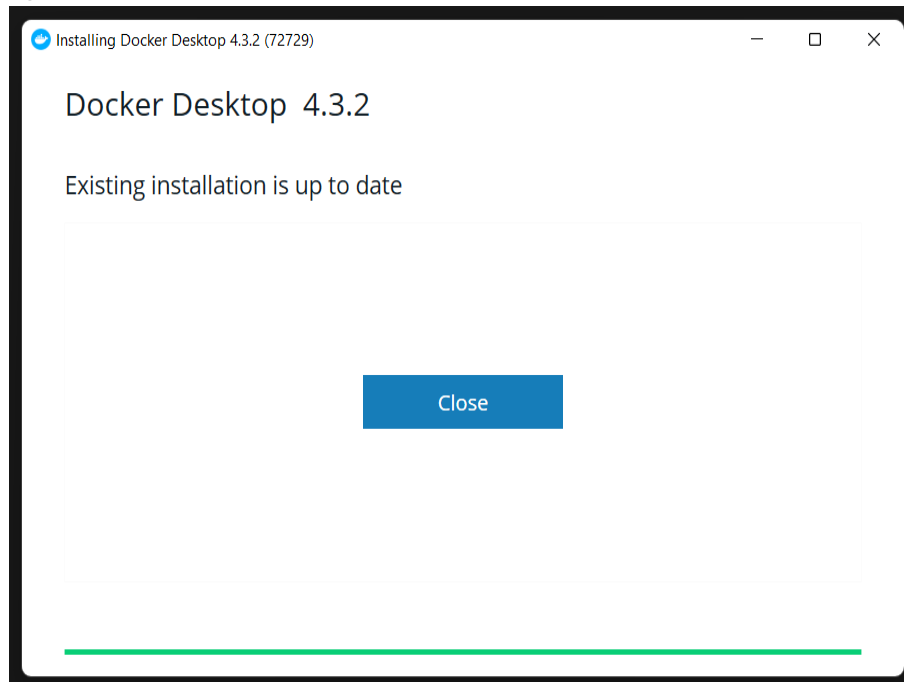
Step 3: Go to Terminal/Command and run `docker –version` to verify whether the Docker is installed properly or not.
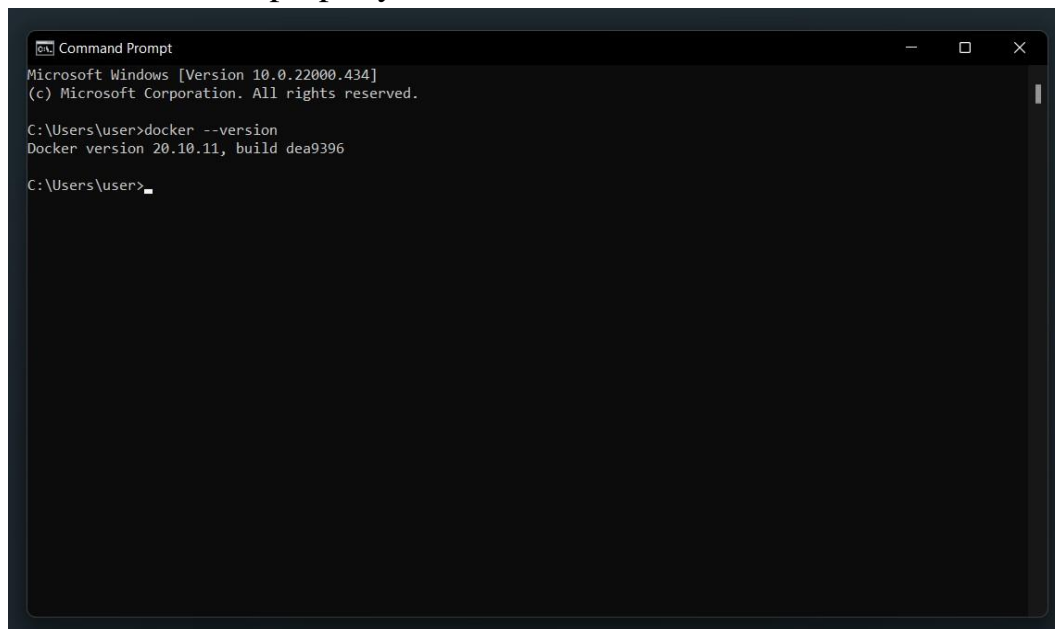
Step 5: Open a command-line terminal like Power Shell, and try out some Docker commands!

Run **docker run hello-world** to verify that Docker can pull and run images. It didn't find `hello-world` in local so it downloaded from Docker Hub.

❖ Basic Commands:

docker run: The docker run command first creates a writeable container layer over the specified image, and then starts it using the specified command.

docker pull: Can be used to pull images from docker hub.

docker images: The default docker images will show all top-level images, their repository and tags, and their size.
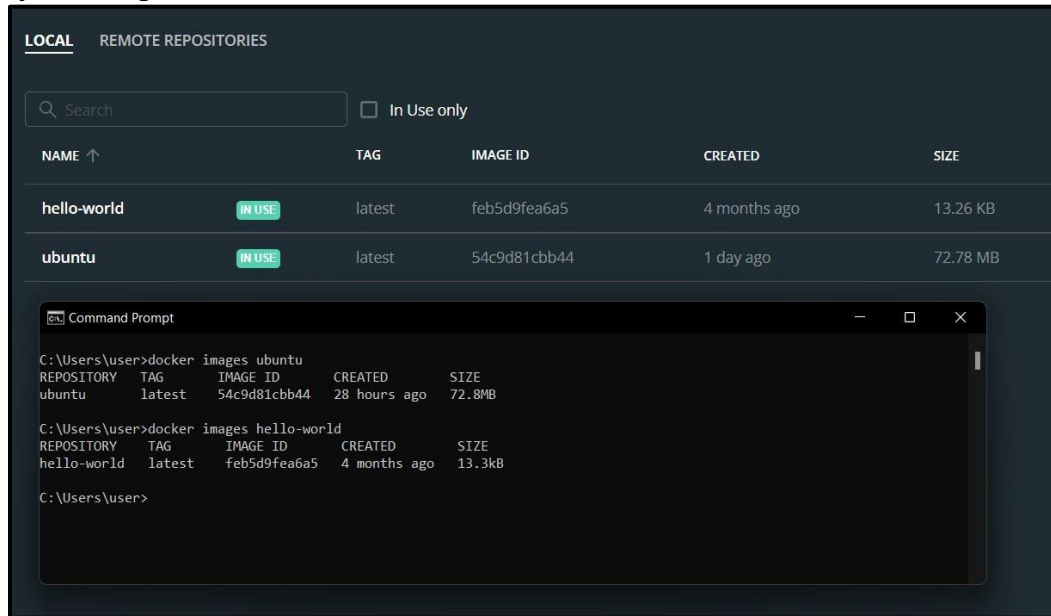
docker start: A stopped container can be restarted with all its previous changes intact using docker start.

docker save: Produces a tarred repository to the standard output stream.

docker rename: The docker rename command renames a container.
$ docker rename CONTAINER NEW_NAME

docker rm: Remove one or more containers. docker rmi: Remove one or more images.

❖ Observation and Learning:

It was found that using Docker lets you ship code faster, standardize application operations, seamlessly move code, and save money by improving resource utilization. With Docker, you get a single object that can reliably run anywhere. Docker's simple and straightforward syntax gives you full control. Wide adoption means there's a robust ecosystem of tools and off-the-shelf applications that are ready to use with Docker.

With this experiment, I got to know about the Docker, how it works, why to use Docker and when to use Docker.

❖ Conclusion:

Docker is a set of platforms as a service product that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries, and configuration files; they can communicate with each other through well- defined channels.

Docker is a software platform that allows you to build, test, and deploy applications quickly.

❖ Questions:

Question: 1 →What is the difference between Docker and VMware?

- VMware emulates machine hardware whereas Docker emulates the operating system in which your application runs. Docker is a much more lightweight virtualization technology since it does not have to emulate server hardware resources. The focus is on abstracting the environment required by the app, rather than the physical server.
- VMware, just like actual machine hardware, lets you install operating systems and other tasks that require a full server.

Question: 2 →Where is docker images located?

- The heaviest contents are usually images. If you use the default storage driver overlay2, then your Docker images are stored in /var/lib/docker/overlay2. There, you can find different files that represent read-only layers of a Docker image and a layer on top of it that contains your changes.

Question: 3 →What do you mean by the term container?

- A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. ... Secure: Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry.