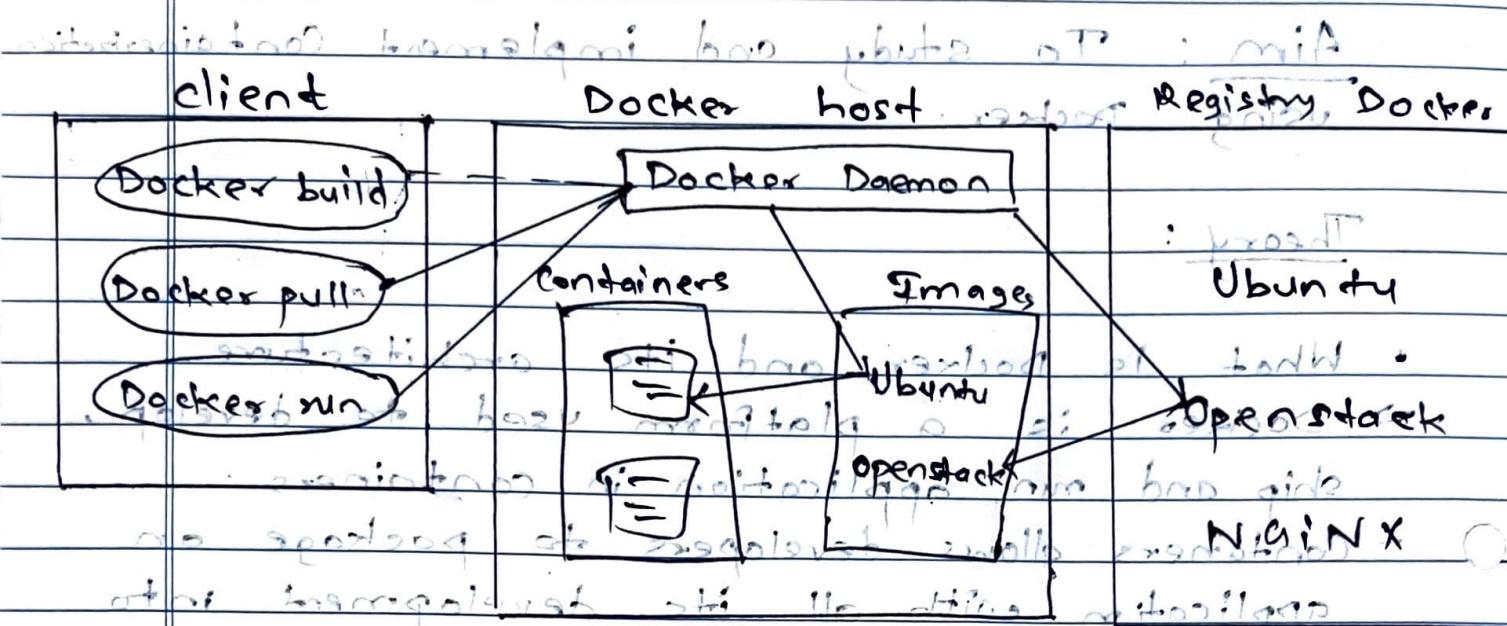


## Experiment No: 9

Aim: To study and implement Containerization using Docker.

### Theory:

- What is Docker, and its architecture
- Docker is a platform used to develop, ship and run application in containers.
- Containers allows developers to package an application with all its development into a standardized unit for software development.
- Docker makes use of client-server architecture. The Docker client talks with the docker daemon which helps in building, running, and distributing the docker containers.
- The Docker client runs with the daemon on the same system or we can connect the Docker client with the Docker daemon remotely.
- With the help of REST API over a UNIX socket or a network, the docker client and daemon interact with each other.
- Docker daemon manages all the services by communication with other daemon.
- It manages docker objects such as images, containers, networks and volumes with the help of the API requests of Docker.



Build → pull → run

- Benefits of Containerization

⇒ Containerization offers several advantages in

- Portability: Containers encapsulate an application and its dependencies, making them portable across different environments. From the development to production, the application does not need to change its environment.

b) Isolation → application cannot interact with other applications or dependencies outside the container.

⇒ Containers provide isolated environments for applications, ensuring that changes or dependencies outside the container do not affect others.

3) Scalability: Containers can be easily scaled up or down based on the demand, allowing for efficient resource utilization.

• Explain the following with Docker:

1) Containers: A container is a lightweight, standalone, executable package that includes everything needed to run a piece of software, including the code, runtime, system tools, system libraries and settings.

2) Images: Images are read-only templates used to create containers. They contain the application code, dependencies, and other necessary files and metadata required to run the app.

3) Dockerfile: A Dockerfile is a text file that contains instructions for building a Docker image. It specifies the base image, environment variables, commands to run during the image build process and other configuration settings necessary to create the image.

Q) Compare the following : Virtual machine & Container

Ans) Container vs virtual machine

### Virtual Machine vs Container

- |   |   |
|---|---|
| 1) Application running on VM system or hypervisor, are in a different OS. | 1) While application running in a container environment shares a single OS. |
| 2) VM virtualizes the computer system, meaning its hardware.              | 2) While a container virtualizes the OS or software only.                   |
| 3) VM size is very large, generally in GB.                                | 3) While the size of container is very light.                               |
| 4) E.g., Type 1 hypervisor bare KVM, Xen, etc.                            | 4) E.g., Rancher, OS, PhotonOS, etc.  |

Q) Image and container. Ans)

Ans) Both are the basic building blocks.

### Image

- 1) It is a blueprint of the container.
- 2) Image is a logical identity, no physical.
- 3) Images are created only once no matter how many times it is used.
- 4) Sharing of docker image is possible.

### Container

- 1) It is the instance of the image.
- 2) The container is real world identity.
- 3) The containers are created any kind of time using an image.
- 4) Sharing of containers is not exactly possible.

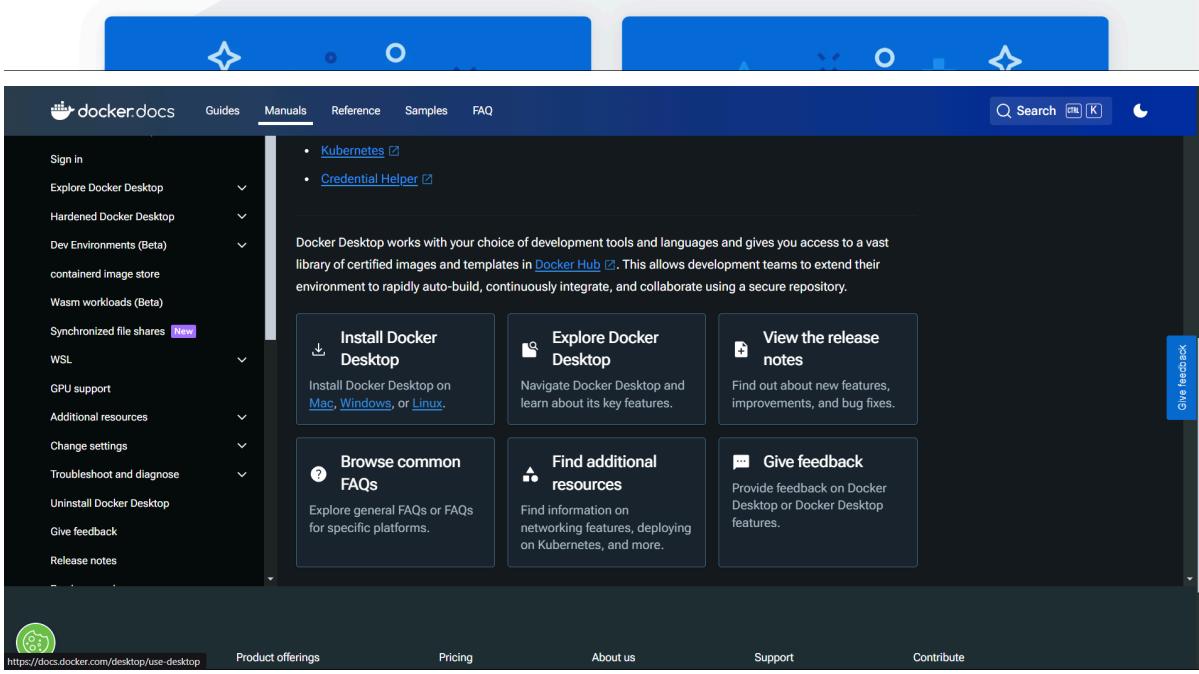
✓ (A)

SP  
18/3/2024

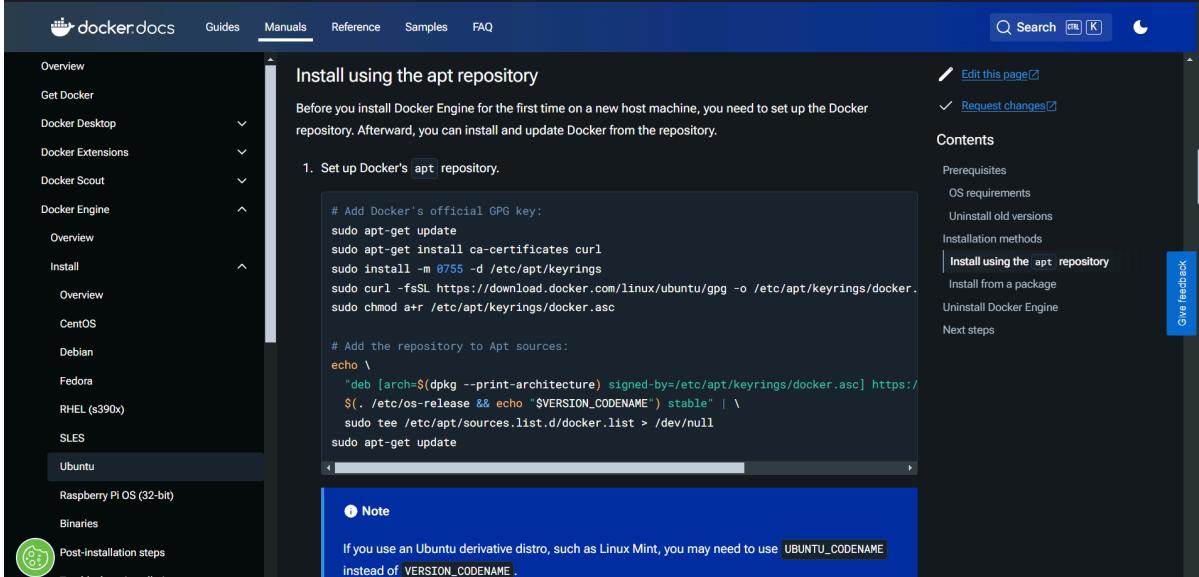
## Output:



The screenshot shows the Docker homepage with the title "Get Started with Docker" and a subtitle "Build applications faster and more securely with Docker for developers". It features two main buttons: "Learn how to install Docker" and "Download for Windows".

This screenshot shows the "Docker Desktop" documentation page under the "Manuals" tab. It includes sections for "Kubernetes" and "Credential Helper". Below these, there's a summary of Docker Desktop's capabilities and links to "Install Docker Desktop", "Explore Docker Desktop", "View the release notes", "Browse common FAQs", "Find additional resources", and "Give feedback".

This screenshot shows the "Ubuntu" documentation page for Docker Engine. It starts with an "Install using the apt repository" section containing a command-line script for setting up the Docker GPG key and repository. A note at the bottom states: "If you use an Ubuntu derivative distro, such as Linux Mint, you may need to use `UBUNTU_CODENAME` instead of `VERSION_CODENAME`". To the right, there are links for "Edit this page", "Request changes", and a "Contents" sidebar.

The screenshot shows the Docker documentation page for Ubuntu. The left sidebar lists various operating systems: Overview, Get Docker, Docker Desktop, Docker Extensions, Docker Scout, Docker Engine, Overview, Install, Overview, CentOS, Debian, Fedora, RHEL (x86\_64), SLES, Ubuntu, Raspberry Pi OS (32-bit), and Binaries. The 'Ubuntu' link is highlighted. Below the sidebar, the main content area has a blue header bar with the text: "If you use an Ubuntu derivative distro, such as Linux Mint, you may need to use `UBUNTU_CODENAME` instead of `VERSION_CODENAME`". The main content is divided into sections: "2. Install the Docker packages." (with "Latest" selected), "To install the latest version, run:" (containing a terminal command: `$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-`), "3. Verify that the Docker Engine installation is successful by running the `hello-world` image." (containing a terminal command: `$ sudo docker run hello-world`), and a "Tip" section with the text: "Receiving errors when trying to run without root?". On the right side, there is a "Contents" sidebar with links to Prerequisites, OS requirements, Uninstall old versions, Installation methods (highlighted), Install using the `apt` repository, Install from a package, Uninstall Docker Engine, and Next steps. There are also "Edit this page" and "Request changes" buttons at the top right.

The screenshot shows the Docker Desktop application window. The left sidebar includes icons for Containers, Images, Volumes, Builds, Dev Environments (BETA), Docker Scout, and Extensions (with a plus sign). The main area is titled "Containers" and features a central image of three overlapping containers. Below it, the text "Your running containers show up here" and "A container is an isolated environment for your code". At the bottom of this section are two cards: "What is a container?" (5 mins) and "How do I run a container?" (6 mins). A "View more in the Learning center" link is located between them. The bottom status bar shows "Engine running", system resources (RAM 1.64 GB, CPU 0.25%), and a signed-in user. The version "v4.28.0" is also visible.

The screenshot shows the Docker desktop application's main interface. On the left is a sidebar with icons for Containers, Images, Volumes, Builds, Dev Environments (BETA), Docker Scout, Extensions (with a plus icon and 'Add Extensions'), and a three-dot menu. The main area is titled 'Images' with a 'Give feedback' link. It has tabs for 'Local' and 'Hub'. Below that is a search bar and a table with columns: Name, Tag, Status, Created, Size, and Actions. One item is listed: 'ubuntu' with tag 'latest', status 'In use', created '17 days ago', size '77.86 MB', and actions buttons. A note says 'Last refresh: 10 minutes ago'. At the bottom, it says 'Showing 1 item'. Below the table is a 'walkthroughs' section with two cards: 'How do I run a container?' (with steps 1-4) and 'Run Docker Hub images' (with a 'docker hub image' icon). A link 'View more in the Learning center' is at the bottom.

The terminal window shows the following session:

```
student@LAB301PC05:~$ docker run ubuntu
student@LAB301PC05:~$ sudo docker pull ubuntu:20.04-
invalid reference format
student@LAB301PC05:~$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
^Cstudent@LAB301PC05:~$ sudo docker pull ubuntu:latest
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

**What's Next?**

1. Sign in to your Docker account → [docker login](#)
2. View a summary of image vulnerabilities and recommendations → [docker scout quickview ubuntu:latest](#)

```
student@LAB301PC05:~$ sudo docker run -ti --rm ubuntu
root@e56cf5d99e5:/#
```