

# 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.

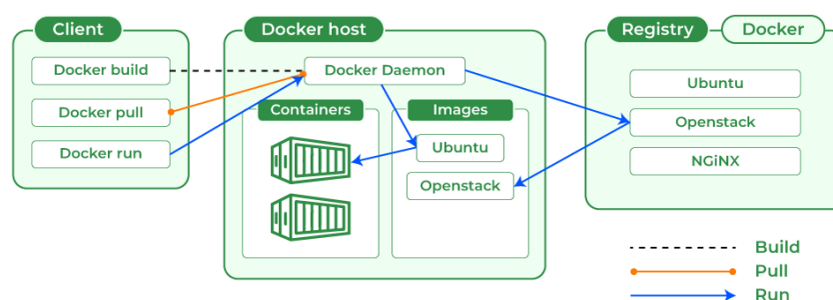
**LO mapped:** LO1, LO5

**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*

```

vishah@divyashah:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Hit:6 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [765 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [486 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [574 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [321 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [223 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [165 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [42.9 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [181 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 C-n-f Metadata [15.6 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [981 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 C-n-f Metadata [11.9 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [781 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [557 kB]
Get:20 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [554 kB]
Get:21 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [40.1 kB]
Get:22 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 C-n-f Metadata [16.7 kB]
Get:23 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [214 kB]
Get:24 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [289 kB]
Get:25 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 C-n-f Metadata [21.8 kB]
Get:26 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:27 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [4,988 B]
Get:28 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.7 kB]
Fetched 7,019 kB in 7s (1,025 kB/s)
Reading package lists... Done
vishah@divyashah:~$

```

```
sudo apt-get install docker-ce
```

```

@ubuntu:~$ sudo apt-get install docker-ce
Installing package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
cni-plugins iproute2 libfido1 libfido2 libidn2-0 libltdl7 libncurses6 libnettle8_0 libopenblas-base libopenblas-dev libopenblas-pthread-compat libopenblas-threaded libopenblas0 libopenblas0-lapack libopenblas0-openblas libopenblas0-threads libopenmpi-bin libopenmpi-dev libopenmpi3 libpython3.9-minimal libpython3.9-stdlib libseccomp2 libsystemd-shared libsystemd-sysv libsystemd0 libtinfo6 libunistring2 libusb-1.0-0 libzstd1 python3-gpgme python3-idna python3-jinja2 python3-markupsafe python3-netaddr python3-paramiko python3-pyasn1 python3-pyserial python3-pyudev python3-pyyaml python3-zipp python3.9 python3.9-minimal xauth xfs-progs xkb-data xkb-tables
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
docker-ce-cli docker-compose-plugin
Suggested packages:
aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
containerd.io docker-ce docker-ce-cli docker-compose-plugin
0 upgraded, 4 newly installed, 0 to remove and 16 not upgraded.
Need to get 36.9 MB/36.9 MB of archives.
After this operation, 354 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.22-1 [28.0 MB]
Get:2 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:20.10.8~3-1ubuntu2.04-jammy [22.6 MB]
Get:3 amd64 io_uring in 34s (3.6 kB/s)
Selecting previously unselected package containerd.io.
Unpacking database ... 28896 files and directories currently installed.)
Preparing to unpack .../containerd.io_1.6.22-1_amd64.deb ...
Unpacking containerd.io (1.6.22-1) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../docker-ce-cli_5:20.10.8~3-1ubuntu2.04-jammy_amd64.deb ...
Unpacking docker-ce-cli (5:20.10.8~3-1ubuntu2.04-jammy) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../docker-ce_5:20.10.8~3-1ubuntu2.04-jammy_amd64.deb ...
Unpacking docker-ce (5:20.10.8~3-1ubuntu2.04-jammy) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../docker-compose-plugin_2.21.8-1ubuntu2.04-jammy_amd64.deb ...
Unpacking docker-compose-plugin (2.21.8-1ubuntu2.04-jammy) ...
Setting up containerd.io (1.6.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.21.8-1ubuntu2.04-jammy) ...
Setting up docker-ce-cli (5:20.10.8~3-1ubuntu2.04-jammy) ...
Setting up docker-ce (5:20.10.8~3-1ubuntu2.04-jammy) ...
Created symlink /etc/systemd/system/docker.socket → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up triggerman-man-db (2.20-2.1) ...
@ubuntu:~$
```

```
sudo snap install docker
```

```
divyashah@DivyaShah:~$ sudo snap install docker
docker 20.10.24 from Canonical** installed
divyashah@DivyaShah:~$
```

*For checking docker version installed you can use*

```
divyashah@DivyaShah:~$ docker --version
Docker version 24.0.6, build ed223bc
divyashah@DivyaShah:~$
```

```
sudo docker run hello -- world
```

```

ayyashah@DivaShah:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e3284a: Pull complete
Digest: sha256:4f53e2564790c8e7856c08e384732aa38dc43c52f09252483ef003afb2dbd
Status: Downloaded newer image for hello-world:latest

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (and64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a Free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

ayyashah@DivaShah:~$

```

*sudo docker images*

```
divyashah@DivyaShah:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world    latest    9c7a54a9a43c   4 months ago   13.3kB
divyashah@DivyaShah:~$
```

*sudo docker ps -a*

```
divyashah@DivyaShah:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
8368ca9ee1fe   hello-world   "/hello"   3 minutes ago   Exited (0) 3 minutes ago           jovial_bhabha
divyashah@DivyaShah:~$
```

*sudo docker ps*

```
divyashah@DivyaShah:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
divyashah@DivyaShah:~$
```

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