**Docker**

**Installation Steps:**

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "${UBUNTU\_CODENAME:-$VERSION\_CODENAME}") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

Verify that the installation is successful by running the hello-world image:

$ sudo docker run hello-world

**What is Docker**

Have you ever wondered about the reason for creating Docker Containers in the market? Before Docker, there was a big issue faced by most developers whenever they created any code that code was working on that developer computer, but when they try to run that particular code on the server, that code was not working. This happened because apps need the right environment to run (like the right OS, libraries, and settings). If something was different on your computer vs. the server, things would break.

To solve this issue, Docker container comes. **Docker is an open-source containerization platform** by which you can pack your application and all its dependencies into a standardized unit called a container. Containers are light in weight, which makes them portable, and they are isolated from the underlying infrastructure and from each other’s container. You can run the docker image as a docker container in any machine where docker is installed without depending on the operating system.

There are two big pieces to Docker: The **Docker Engine**, which is the Docker binary that’s running on your local machine and servers and does the work to run your software. The **Docker Hub** is a website and cloud service that makes it easy for everyone to share their docker images.

## **Key Components of Docker**

The following are the some of the key components of Docker:

* **Docker Engine:** [Docker Engine](https://www.geeksforgeeks.org/what-is-docker-engine/) is a core part of docker, that handles the creation and management of containers.
* **Docker Image:** [Docker Image](https://www.geeksforgeeks.org/what-is-docker-image/) is a read-only template that is used for creating containers, containing the application code and dependencies.
* **Docker Hub:** It is a cloud based repository that is used for finding and sharing the container images.
* **Dockerfile:** It is a file that describes the steps to create an image quickly.
* **Docker Registry** : It is a storage distribution system for docker images, where you can store the images in both public and private modes

## **What is Docker Container?**

Docker container is a runtime instance of an image. Allows developers to package applications with all parts needed such as libraries and other dependencies. Docker Containers are runtime instances of Docker images. Containers contain the whole kit required for an application, so the application can be run in an isolated way.

## **Docker Commands**

Through introducing the essential docker commands, docker became a powerful software in streamlining the container management process. It helps in ensuring a seamless development and deployment workflows. The following are the some of docker commands that are used commonly:

* **Docker Run:** It used for launching the containers from images, with specifying the runtime options and commands.
* **Docker Pull:** It fetches the container images from the container registry like Docker Hub to the local machine.
* **Docker ps** : It helps in displaying the running containers along with their important information like container ID, image used and status.
* **Docker Stop** : It helps in halting the running containers gracefully shutting down the processes within them.
* **Docker Start:** It helps in restarting the stopped containers, resuming their operations from the previous state.
* **Docker Login:** It helps to login in to the docker registry enabling the access to private repositories.

sudo docker run hello-world

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.

(amd64)

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/

**sudo docker images**

REPOSITORY TAG IMAGE ID CREATED SIZE

hello-world latest 74cc54e27dc4 3 months ago 10.1kB

**sudo docker pull ubuntu**

Using default tag: latest

latest: Pulling from library/ubuntu

2726e237d1a3: Pull complete

Digest: sha256:1e622c5f073b4f6bfad6632f2616c7f59ef256e96fe78bf6a595d1dc4376ac02

Status: Downloaded newer image for ubuntu:latest

docker.io/library/ubuntu:latest

**sudo docker images**

REPOSITORY TAG IMAGE ID CREATED SIZE

ubuntu latest 602eb6fb314b 2 weeks ago 78.1MB

hello-world latest 74cc54e27dc4 3 months ago 10.1kB

**sudo docker run -it ubuntu**

root@7a2400281564:/# pwd

/

root@7a2400281564:/#

root@7a2400281564:/# ls

bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

root@7a2400281564:/# cd home

root@7a2400281564:/home# ls

ubuntu

root@7a2400281564:/home# cd ubuntu

root@7a2400281564:/home/ubuntu# ls

root@7a2400281564:/home/ubuntu# exit

exit

**sudo docker ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

**sudo docker ps -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

7a2400281564 ubuntu "/bin/bash" 5 minutes ago Exited (0) 58 seconds ago sweet\_shamir

2a11e09ea3b1 hello-world "/hello" 17 minutes ago Exited (0) 17 minutes ago jovial\_shtern

5c7a74d8e149 hello-world "/hello" 20 minutes ago Exited (0) 20 minutes ago bold\_jackson

**sudo docker run -d ubuntu**

13bf0f7eab1c5d357cd71bef1a580e57c41b34741c726dedba3f899622edb958

**sudo docker ps -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

13bf0f7eab1c ubuntu "/bin/bash" 10 seconds ago Exited (0) 9 seconds ago eager\_stonebraker

7a2400281564 ubuntu "/bin/bash" 7 minutes ago Exited (0) 3 minutes ago sweet\_shamir

2a11e09ea3b1 hello-world "/hello" 19 minutes ago Exited (0) 19 minutes ago jovial\_shtern

5c7a74d8e149 hello-world "/hello" 22 minutes ago Exited (0) 22 minutes ago bold\_jackson

**sudo docker run -d ubuntu sleep 7000**

D415de7d830a077d5750eb2a2a734fd2a780492573292d8c1b5605391f185c69

**sudo docker ps -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

d415de7d830a ubuntu "sleep 7000" 2 seconds ago Up 1 second cranky\_benz

13bf0f7eab1c ubuntu "/bin/bash" About a minute ago Exited (0) About a minute ago eager\_stonebraker

7a2400281564 ubuntu "/bin/bash" 8 minutes ago Exited (0) 4 minutes ago sweet\_shamir

2a11e09ea3b1 hello-world "/hello" 20 minutes ago Exited (0) 20 minutes ago jovial\_shtern

5c7a74d8e149 hello-world "/hello" 23 minutes ago Exited (0) 23 minutes ago bold\_jackson

**sudo docker ps -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

d415de7d830a ubuntu "sleep 7000" 38 seconds ago Up 37 seconds cranky\_benz

13bf0f7eab1c ubuntu "/bin/bash" About a minute ago Exited (0) About a minute ago eager\_stonebraker

7a2400281564 ubuntu "/bin/bash" 8 minutes ago Exited (0) 4 minutes ago sweet\_shamir

2a11e09ea3b1 hello-world "/hello" 21 minutes ago Exited (0) 21 minutes ago jovial\_shtern

5c7a74d8e149 hello-world "/hello" 23 minutes ago Exited (0) 23 minutes ago bold\_jackson

**sudo docker ps -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

d415de7d830a ubuntu "sleep 7000" About a minute ago Up About a minute cranky\_benz

13bf0f7eab1c ubuntu "/bin/bash" 2 minutes ago Exited (0) 2 minutes ago eager\_stonebraker

7a2400281564 ubuntu "/bin/bash" 9 minutes ago Exited (0) 5 minutes ago sweet\_shamir

2a11e09ea3b1 hello-world "/hello" 21 minutes ago Exited (0) 21 minutes ago jovial\_shtern

5c7a74d8e149 hello-world "/hello" 24 minutes ago Exited (0) 24 minutes ago bold\_jackson

**sudo docker stop cranky\_benz**

Cranky\_benz

**sudo docker ps -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

d415de7d830a ubuntu "sleep 7000" 2 minutes ago Exited (137) 4 seconds ago cranky\_benz

13bf0f7eab1c ubuntu "/bin/bash" 3 minutes ago Exited (0) 3 minutes ago eager\_stonebraker

7a2400281564 ubuntu "/bin/bash" 10 minutes ago Exited (0) 6 minutes ago sweet\_shamir

2a11e09ea3b1 hello-world "/hello" 23 minutes ago Exited (0) 23 minutes ago jovial\_shtern

5c7a74d8e149 hello-world "/hello" 25 minutes ago Exited (0) 25 minutes ago bold\_jackson