

1. What is docker?

* Docker is a centralized platform for packaging, deploying, and running applications.
* Before Docker, many users face the problem that a particular code is running in the developer's system but not in the user's system. So, the main reason to develop docker is to help developers to develop applications easily, ship them into containers, and deploy them anywhere.
* Docker uses a container on the host's operating system to run applications. It allows applications to use the same Linux kernel as a system on the host computer, rather than creating a whole virtual operating system.

2. Docker Containers

* Docker containers are lightweight alternatives to the virtual machine.
* It allows developers to package up the application with all its libraries and dependencies, and ship it as a single package.
* The advantage of using a docker container is that you don't need to allocate any RAM and disk space for the applications. It automatically generates storage and space according to the application requirement.

3. Virtual Machine

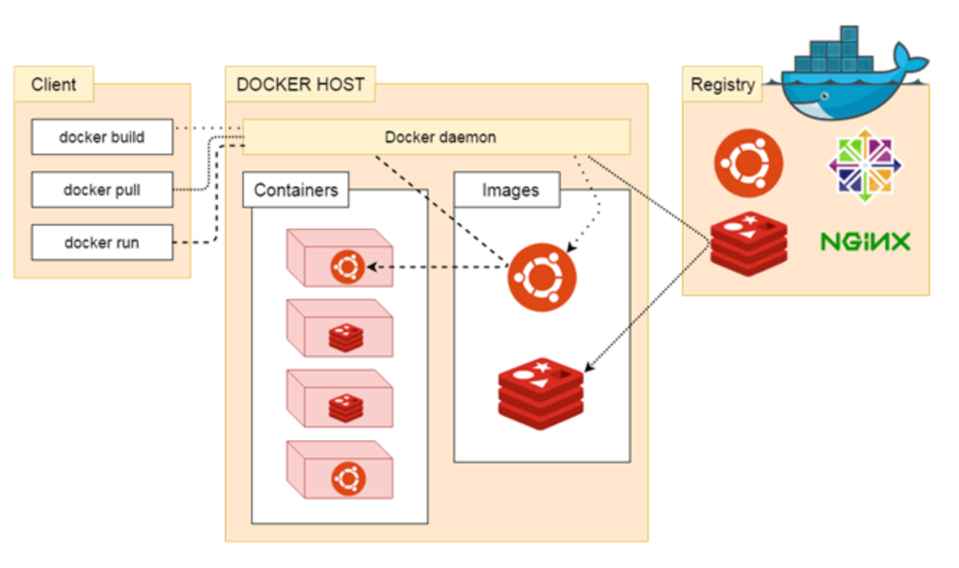
* A virtual machine is a software that allows us to install and use other operating systems (Windows, Linux, and Debian) simultaneously on our machine.
* The operating system in which virtual machine runs are called virtualized operating systems.
* These virtualized operating systems can run programs and performs tasks that we perform in a real operating system.

4. Containers Vs. Virtual Machine

|  |  |
| --- | --- |
| **Containers** | **Virtual Machine** |
| Integration in a container is faster and cheap. | Integration in virtual is slow and costly. |
| No wastage of memory. | Wastage of memory. |
| It uses the same kernel, but different distribution. | It uses multiple independent operating systems. |

5. Docker daemon?

* Docker daemon runs on the host operating system.
* It is responsible for running containers to manage docker services.
* Docker daemon communicates with other daemons. It offers various Docker objects such as images, containers, networking, and storage.



6. Docker Architecture

Docker Client

Docker client uses **commands** and **REST APIs** to communicate with the Docker Daemon (Server). When a client runs any docker command on the docker client terminal, the client terminal sends these docker commands to the Docker daemon. Docker daemon receives these commands from the docker client in the form of command and REST API's request.

#### Note: Docker Client has the ability to communicate with more than one docker daemon.

Docker Client uses Command Line Interface (CLI) to run the following commands -

docker build

docker pull

docker run

### Docker Host

Docker Host is used to provide an environment to execute and run applications. It contains the docker daemon, images, containers, networks, and storage.

Docker Registry

Docker Registry manages and stores the Docker images.

There are two types of registries in the Docker -

**Pubic Registry -** Public Registry is also called as **Docker hub**.

**Private Registry -** It is used to share images within the enterprise.

Docker Objects

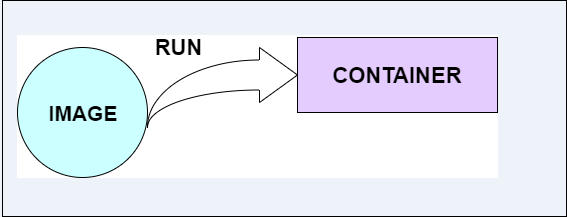
There are the following Docker Objects -

# Docker Container and Image

Docker container is a running instance of an image. You can use Command Line Interface (CLI) commands to run, start, stop, move, or delete a container. Docker container is an isolated and secure application platform, but it can share and access to resources **running in a different host or container**.

An image is a read-only template with instructions for creating a Docker container. A docker image is described in text file called a **Dockerfile**. An image does not have states and never changes. Docker Engine provides the core Docker technology that enables images and containers.

AD



Eg - $ docker run hello-world

The above command **docker run hello-world** has three parts.

1) **docker:** It is docker engine and used to run docker program. It tells to the operating system that you are running docker program.

2) **run:** This subcommand is used to create and run a docker container.

3) **hello-world:** It is a name of an image. You need to specify the name of an image which is to load into the container.