* **What is Docker?**

Docker is an **open-source platform designed** to create, deploy, and run applications. Docker uses **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. Containers ensure that our application works in any environment like development, test, or production.

* **Why Docker?**

Developers can create containers without Docker, but **the platform makes it easier, simpler, and safer to build, deploy and manage containers**. Docker is essentially a toolkit that enables developers to build, deploy, run, update, and stop containers using simple commands and work-saving automation through a single API.

* **Features Of Docker?**

**Easy and Faster Configuration:**

This is a key feature of docker that helps us to configure the system easily and faster.

We can deploy our code in less time and effort. As Docker can be used in a wide variety of environments.

**Increase Productivity:**

By easing technical configuration and rapid deployment of application. No doubt it has increase productivity. Docker not only helps to execute the application in isolated environment but also it has reduced the resources.

**Application Isolation:**

It provides containers that are used to run applications in isolation environment. Each container is independent to another and allows us to execute any kind of application.

**Routing Mesh:**

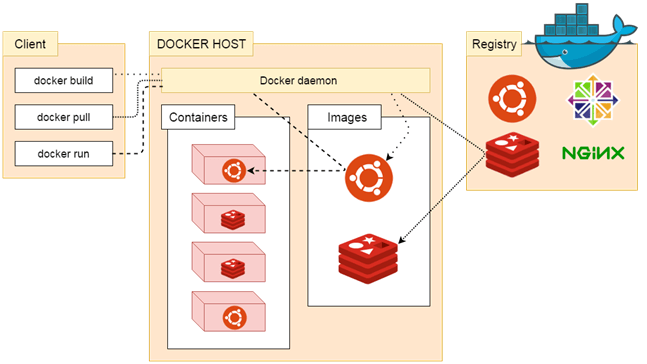
It routes the incoming requests for published ports on available nodes to an active container. This feature enables the connection even if there is no task is running on the node.

**Security Management:**

It allows us to save secrets into the swarm itself and then choose to give services access to certain secrets.

* **Docker Architecture:**

Docker follows Client-Server architecture, which includes the three main components that are **Docker Client**, **Docker Host**, and **Docker Registry**.



**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.

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 -

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

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

**Docker Objects:**

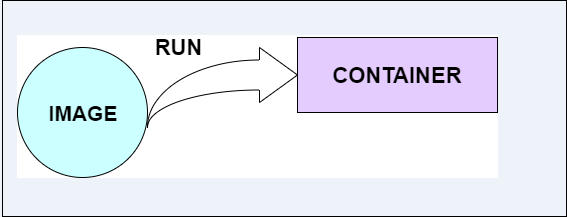
There are the following Docker Objects –

* **Docker Image:**

Docker images are the **read-only binary templates** used to create Docker Containers. It uses a private container registry to share container images within the enterprise and also uses public container registry to share container images within the whole world. Metadata is also used by docket images to describe the container's abilities.

* **Docker Container:**

Containers are the structural units of Docker, which is used to hold the entire package that is needed to run the application. The advantage of containers is that it requires very less resources.



### Docker Networking

Using Docker Networking, an isolated package can be communicated. Docker contains the following network drivers -

* **Bridge -** Bridge is a default network driver for the container. It is used when multiple docker communicates with the same docker host.
* **Host -** It is used when we don't need for network isolation between the container and the host.
* **None -** It disables all the networking.
* **Overlay -** Overlay offers Swarm services to communicate with each other. It enables containers to run on the different docker host.
* **Macvlan -** Macvlan is used when we want to assign MAC addresses to the containers.

### Docker Storage:

Docker Storage is used to store data on the container. Docker offers the following options for the Storage -

* **Data Volume -** Data Volume provides the ability to create persistence storage. It also allows us to name volumes, list volumes, and containers associates with the volumes.
* **Directory Mounts -** It is one of the best options for docker storage. It mounts a host's directory into a container.
* **Storage Plugins -** It provides an ability to connect to external storage platforms.

**Docker Installation:**

