Docker

What is Docker?

Docker is a container management service. The keywords of Docker are developing**, ship**, and **run** anywhere. The main idea of Docker is to help developers quickly improve applications, ship them into containers, which can then be, deployed anywhere.

Docker helps developers bring their ideas to life by conquering the complexity of app development. We simplify and accelerate development workflows with an integrated dev pipeline and through the consolidation of application components. Actively used by millions of developers around the world, Docker Desktop and Docker Hub provide unmatched simplicity, agility, and choice.

The initial release of Docker was in March 2013; it has become the buzzword for modern world development, especially in the face of Agile-based projects.

Features of Docker

- Docker can reduce the size of development by providing a smaller footprint of the operating system via containers.
- Through containers, it becomes easier for teams across different units, such as development, QA, and Operations to work seamlessly across applications.
- You can deploy Docker containers anywhere, on any physical and virtual machines, and even on the cloud.
- Since Docker containers are lightweight, they are very easily scalable.

Components of Docker

Software: The Docker daemon, called Dockerd, is a steadfast process that manages Docker containers and handles container objects. The daemon listens for requests sent via the Docker Engine API. The Docker client program, called Docker, provides a command-line interface, CLI, which allows users to interact with Docker daemons

Objects: Docker objects are various entities used to assemble an application in Docker. The main classes of Docker objects are images, containers, and services.

- **A Docker** container is a standardized, encapsulated environment that runs applications. A container supervised using the Docker API or CLI.
- **A Docker** image is a read-only template used to build containers. Images marketed to store and ship applications.
- **A Docker** service allows containers to peel across multiple Docker daemons. The result is known as a swarm, a set of cooperating daemons that communicate through the Docker API.

Registries: A Docker registry is a repository for Docker images. Docker clients connect to registries to download ("pull") images for use or upload ("push") images that they have built. Registries can be public or private. Two chief public registries are Docker Hub and Docker Cloud.

- **Docker for Mac:** It enables the developer to run the Docker containers within Mac OS.
- **Docker for Linux:** It allows the developer to run the Docker containers within Linux OS.

- **Docker for Windows:** It enables the developer to run the Docker containers within Windows OS.
- **Docker Engine:** It managed for building Docker images and creating Docker containers.
- **Docker Hub:** This is the registry, which used to host various Docker images.
- **Docker Compose:** This handled to define applications using multiple Docker containers.

Tools

Docker Compose: is a tool for defining and running multi-container Docker applications. It uses YAML files to configure the application's services and performs the creation and start-up process of all the containers with a single command. The Docker-compose CLI utility allows users to run commands on multiple containers at once. For example, building images, scaling containers, running containers that had stopped, and more. Commands related to image manipulation, or user-interactive options, are not relevant in Docker Compose because they address one container. The Docker-compose.yml file is used to define an application's services and includes various configuration options. For example, the build option defines configuration options such as the Docker-file path. The command option allows one to override default Docker commands, and more.

Docker Swarm: provides native clustering functionality for Docker containers, which turns a group of Docker engines into a single virtual Docker Engine. In Docker 1.12 and higher, Swarm mode mixed with Docker Engine. The Docker swarm CLI utility allows users to run Swarm containers, create discovery tokens, list nodes in the cluster, and more. The Docker node CLI utility enables users to run various commands to manage nodes in a swarm, for example, listing the nodes in a cloud, updating nodes, and removing nodes from the cloud. Docker handles Troops using the Raft consensus algorithm. According to Raft, for an update that has been performed, the majority of Swarm nodes need to agree on the update.