**Docker**

\***What is Docker**?

Its an open-source centralized platform designed to create, deploy, and run applications. Docker uses container on the host's operating system to run applications.

\***So what is Docker container**?

Docker containers are the lightweight alternatives of the virtual machine. It allows developers to package up the application with all its libraries and dependencies, and ship it as a single package.

Advantages of docker container:

* 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.

Disadvantages of Docker:

* It increases complexity due to an additional layer
* In Docker, it is difficult to manage large amount of containers

\***Features of docker**

* Easy and Faster Configuration
* Increase productivity
* Application Isolation
* Swarm
* Routing Mesh
* Services
* Security Management

\***Docker container & 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. 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, which has a simple, well-defined syntax. An image does not have states and never changes. Docker Engine provides the core Docker technology that enables images and containers.

\***Useful Commands**

🡪 To build Docker Image from a Dockerfile:

$ docker build -t image-name docker-file-location

-t: it is used to tag Docker image with the provided name.

🡪 To Run Docker Image

$ docker run -d image-name

-d: It is used to create a daemon process.

🡪 To Check available Docker images

$ docker images

🡪 To Check for latest running container

$ docker ps -l

-l: it is used to show latest available container.

🡪 To Check all running containers

$ docker ps -a

-a: It is used to show all available containers.

🡪 Stop running container

$ docker stop container\_id

container\_id: It is an Id assigned by the Docker to the container.

🡪 Delete an image

$ docker rmi image-name

🡪 Delete all images

$ docker rmi $(docker images -q)

🡪 Delete all images forcefully

$ docker rmi -r $(docker images -q)

-r: It is used to delete image forcefully.

🡪 Delete all containers

$ docker rm $(docker ps -a -q)

🡪 Enter into Docker container

$ docker exec -it container-id bash