**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.

\* Docker is created by Solomon Hykes

\* Released as open source in March 2013

\* Docker library is written in GO programming language

**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

**Basic Commands:**

- docker version

- docker -v

- docker info

- docker –help (For eg: docker images --help)

- docker login (login through docker hub)

**System Commands:**

**\*** docker stats (shows the memory usage)

\* docker system df (to check disc usage of docker)

\* docker system prune (to remove unused data)

\***Docker Image**

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.

\***Useful Commands:**

🡪 docker images --help

🡪 docker image -a (show all images)

🡪 docker image -q (only shows numeric ID’s)

🡪 docker image -f “dangling=false” -q (shows us the images which are not associated with the running containers & only the image ID)

\* -f (used to filter the output based on the conditions provided)

\* Dangling images are the images which are not associated with the running container

🡪 docker inspect ubuntu (shows all the details of the image)

**Containers:**

Containers are the running instances of docker images

(whenever you run an image it creates a container)

\***Useful Commands:**

🡪 docker ps (used to list the currently running Docker containers

🡪 docker ps -a (used to list all containers, including running containers and stopped containers)

🡪 docker run (Create a container with the image)

🡪 docker run --name <container\_name> <image\_name> (to give name to the container)

🡪 docker kill (to kill one or more running containers)

🡪 docker history

**Dockerfile:**

\*Dockerfile is a text file with instructions to buid an image

\*Dockerfile is an automation of docker image creation

**Steps to create a dockerfile**:

Step1: Create a file named dockerfile

Step2: Add instructions in dockerfile

Step3: Build dockerfile to create an image

Step4: Run image to create container

RUN- gets executed during the building of the image

CMD- gets executed only after created container out of the image

**DockerCompose:**

🡪tool for defining & running multi-container docker aplications

🡪Use yaml files to configure application services

(docker-compose.yml)

\*Can start all the services with a single command

: docker compose up

\*Can stop all the services with a single command

: docker compose down

\*Can scale up selected services when required

To know how to scale the services use the command

: docker-compose --help