**DOCKER**

**What is Docker?**

Docker is an open-source containerization platform by which you can pack your application and all its dependencies into a standardized unit called a container. Containers are light in weight which makes them portable and they are isolated from the underlying infrastructure and from each other container.

Docker is popular because of the following:

1. Portability.
2. Reproducibility.
3. Efficiency.
4. Scalability.

**How Docker Works**

Docker makes use of a client-server architecture. The Docker client talks with the docker daemon which helps in building, running, and distributing the docker containers. The Docker client runs with the daemon on the same system or we can connect the Docker client with the Docker daemon remotely. With the help of REST API over a UNIX socket or a network, the docker client and daemon interact with each other.

**What is Container ?**  
A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

What is Image?

A Docker image is simply a read-only file that’s used to run code within a Docker container. Think of it as a template that contains all the instructions needed to run the code. All of the code and dependencies are packaged in one file.

**command to check status and start docker:**

: service docker status

: service docker start

**docker info (to check information about docker)**

sudo docker run <containerID>

sudo docker start <containerID>

sudo docker stop <containerID>

**command shows existing containers :-**

sudo docker ps -a

**command to checking current running container :-**

sudo docker ps

sudo docker attach <containerID> **(to log in container)**

**When we log in container and exit from container we press "ctrl+c" / exit , here we get out from container but the container also get stop/exited.**

**Then exit the container while the container press "ctrl+p+q". here the container will still run in background.**

sudo docker container stats  **(It shows the running container status like cpu & memory utilization and more)**

sudo docker container top <containerID> **(shows all process of container)**

**( dmidecode shows all info of our pc h/w and s/w )**

**docker kill 'psID'**

**Port mapping:**

sudo docker container run -d -p 3000:80 --name test1 nginx

**-p : for port forward 80 to 3000**

**--name : Name for new container 🡺 test1**

**Nginx 🡺 old container name**

sudo docker diff <container ID>

**c change**

**A append**

**d delete**

**The below command is used when we wait for the container to go in stop state :**

Sudo docker wait <containerID>

**To pause the running container:**

Sudo docker container pause <containerID>

**To unpause the pause container:**

Sudo docker container unpause <containerID>

**To export container :**

sudo docker container export 874d0a23c205 > my\_nginx.tar **(compressed file name)**

**Container ID { ID of container to which we want to export}**

**The container will be export in the tar format.**

**To Import container:**

sudo docker image import my\_nginx.tar my-g-t

**my\_nginx.tar = compressed file name**

**my-g-t = new image name which you can define**

**To make container from existing image:**

sudo docker container run -it my-g-t /bin/bash

**my-g-t ==> image name from which we have to make container**

**To make Image of a container :**

sudo docker container commit -m "This is change" b43a11c3d74f my\_test\_image

**New image name Container ID**

**To make container from image:**

sudo docker container run -it my\_test\_image /bin/bash

**Existing Image name to make container**

**# IMAGE {inspect, history, remove and list}**

**To format image i.e format according to ID ,REPOSITORY ,etc.**

sudo docker images --format '{{.ID}} , {{.Repository}} - {{.Tag}}'

**it will list only ID and Repository**

**To remove image :**

sudo docker image rmi python:3.8

sudo docker image rm -f nginx ubuntu

**History of image:**

sudo docker image history hello-world

Image name