* To check docker status

Systemctl status docker

* To pull docker image

docker pull nginx

* To pull particular version of docker image using tag

docker pull nginx:1.24.0

If not given by default latest is the tag

* To list docker images

docker images (or)

docker image ls

* To inspect particular image

docker image inspect ubuntu or

docker image inspect <imageid>

* To search an image in respository

docker search nginx

* To delete an image

docker rmi nginx

If a container is associated with an image. It cannot be deleted unless using -f

* To delete dangling images (orphan images)

docker image prune

* To create a container

docker create ubuntu

(We haven’t given a name to the container so docker gives a random name)

* To create a container and giving a name

docker create --name server01 nginx

* To create a container from a particular version of image

docker create --name server02 nginx:1.24.0

* To list only running containers

docker ps (list all running processes)

* To list all created, running, paused, exited, stopped, removed, and dead containers

docker ps -a

* To list only IDs of all containers

Docker ps -a -q

* To start a container

docker start web01

* To stop a container

docker stop web01

* We can use run command instead of create and start
* To remove a container

Docker rm web01

OPERATIONAL COMMANDS

* Usually, clean OS containers start and exit immediately when we run them. Only application specific containers will be up and running. Because OS containers are not designed to be in foreground. We need to get them foreground.

Startup command (entry point) defined in OS images - /bin/bash

* To get OS containers foreground and running

Docker run -dit --name server02 ubuntu

Now the container server02 is up and running. We need to interact with the container

So, we need to use attach command

Docker attach server02 /bin/bash 🡪 goes to bash shell

Docker attach server02 🡪 goes to root

* To come out to docker we use ‘exit’ command which not only comes out but also kills the container and container will be exited state. That’s why we should not use attach command. We should use exec command.
* Docker exec -it server02 /bin/bash

Now we give exit command to come out to docker. Here the container still be up and running.

* Linux command to check the running process in a container

Ps -ef

* We can give a linux command along with exec without entering into container

Docker exec -it server02 ls

It gives the list of files and folders in server02 container without entering into it.

* We can use docker events to check live activity in another window.

Docker events

Live activity doesn’t record in a log file. It just streams the live activity.

* To check logs of a container

Docker logs web01

We can check logs alternatively by going into the container and use linux commands.

Docker exec -it web02 /bin/bash

Cd /var/log

In log folder enter following commands

Cat lastlog

Cat faillog

* To inspect a container

Docker inspect web01

* We can use Container ID instead of Name in all docker commands
* We can pass a command result as input to another command by using $()

Docker start $(docker -ps -a -q)

Docker starts all containers in loop

Docker stop $(docker -ps -a -q)

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

* To display system wide information of the docker server

Docker info