Containers Commands

Once we make sure docker is installed successfully on our system. Lets dive in containers.

Commands:

1. **docker version** - this will return the version of docker for CLI and server.
2. **docker info** : this will return the detailed information about the docker , like number of containers we are running , stoped containers , network settings, downloaded docker images and what not.
3. **docker** : this command will return list of all commands we can use in docker segregated in categories
   1. Management command
   2. commands

that brings us to docker command format

**\*\*\*\* docker <Management\_Command> Sub\_command options \*\*\*\*\***

For ex: **1.docker container start [Container\_id | Container\_name] (Will start the stopped containers )**

**2. docker container ls (will list all active containers)**

**Docker Commands to start the nginx server:**

Before that difference between docker image and container,

Docker image is the set of binaries and source of thet application , and container can be considered as running instance of the docker image.

We can have many containers based of same image.

We get out images from registries that is (hub.docker.com)

1. **docker container run --publish 80:80 nginx**

Explanation: the above command will run the nginx image in an container , where port exposed is 80.

Concepts of ports , if we want to run the container on port of host 8080 which and exposing virtual port 80 of container , then we can mention like

--publish 8080:80

a. First port is port of host

B. second port is port of docker container

So background process of above command:

\*\*Will download the image nginx from hub.docker.com and run the image in the container exposing port 80 of host engine and virtual port 80 of docker container

Once we enter this command the container will start running and we can see all logs 0n the screen if any request is given to local host port 80:80.

To exit from the container in above scenario press (ctrl+c)

1. **docker container run --publish 80:80 --detach nginx**

This command will execute in same way the above command only difference is “--detach” will run the nginx in background , that is all logs are written in log file and are not visible in terminal.

1. **docker container ls :**

This command will list all the running containers.

1. **docker container ls -a:**

This command will list all containers which are running and which has been stopped from running.

1. **docker container stop [Container\_id | conatiner\_name ]:**

This command will stop the running container.

1. **docker container --name webhost --publish 80:80 --detach nginx:**

This command is same as second command in the list , except we have explicitly provided name to container using --name option. We can view the name by doing coker container ls

1. **docker container logs [container\_id | container\_name]:**

This will return the logs of the running mentioned container id or container\_name.

1. **docker container top [Container\_id | Container\_name]:**

This will list processes running inside the containers.

1. **docker container rm [container\_id... | container\_name...]:**

This will remove all mentioned containers id or name , rm command accepts multiple containers\_id or names to remove also , it will throw and error if we try to remove a running container.

1. **docker container rm -f [container\_iod | container\_name]:**

this will remove all mentioned container name or id irrespective of it is running or not.

Just added “-f ” which is for force

**What happens when we run a container .?**

1. Look for the image which we specify locally in image cache., if doesn’t find there then it will search on docker hub download it and store it in image cache
2. If version is not specified , it will by default it will choose latest.
3. Creates a new container and starts it , once the container is started , networks are specified and virtual IP address are given to container.
4. Opens up the port at host and route all the traffic of that port of host to port of container.
5. And finally executes the command present in docker image file.

So wrapping all in one.

docker container run --name [container\_name] --publish 80:80 --detach nginx:version nginx -T

**Running a linux commands in container:**

1. **Docker container run -it --name webhost nginx bash**

Above command -it will start a container interactively and inside a container it will start a bash

Where t is for creating TTy and i is for starting in interactive mode.s

1. **Docker container exec -it [Conatiner\_id | Container\_name] bash:**

Above command will run the bash command inside already running container.

1. **Docker container inspect [Conatiner\_id | Container\_name]**

Above command will return a json file , which have all configuration of the container. Link network it is dusing , IP address, volumes which ios using.

1. **Docker container start -ai [container\_name |container\_id]**

Above command start the container by executing default command mentioned..

Where -a is to attach the stopped container

-i is for interactive mode.