Docker Commands

①

1) docker build -t image-name dockerfile-path
// build an image using a Dockerfile with the specified path.

fanculture [any name you can give]
↑
2) docker build -t image-name .
// build D-f, if D-f is in CD.

3) docker history image-name
// History of an image.

4) docker images
docker images ls
// List the images.

5) docker image image-name
// display The info about image.

6) docker rmi -f image-id / Image-name
// remove an image from local registry.

7) docker --version.
// Docker version & Version is =

8) docker info
// → Status of Docker running or not
→ Display Detailed info about docker service.

9) docker images -q

// Display ONLY the images ID's

10) docker rmi $(docker images -q)

(or)

docker rmi -f $(docker images -q)

// Remove all the images from docker engine

11) docker search image-name

// it will search all of the publicly available images on Docker-hub

[https://hub.docker.com)

12) docker pull ubuntu:22.04
push sugram/image-name:tag

// pull image from Docker hub.
push to

13) docker images --filter dangling=true.

// list dangling [No name & tag] only.

14) docker images --filter dangling=false

// list all images other than dangling. images

15) docker login (or)

docker login -u u-name -p pass

17) docker run -it image-name bash ③

// run /start create a container in interactive mode

i= interactive t= terminal.

18) docker run -d detached mode image-name.

// run in detached mode.

19) docker exec -it container_id bash

// get inside the running container.

20) docker logs -f container_name

// logs of specific container.

21) docker ps or docker container ls

// list only active container

22) docker ps -a

// list running + stopped = all

23) docker ps -a --filter ancestor = image-name

// list all containers of a particular image

24) docker ps -a -q

// list all the STOPPED container

25) docker stop $(sudo docker ps -q)

// Delete ALL running container

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

// delete running + stopped [stop won't work use rm only)

27) docker rm -f Containeid
// To delete the Container.

28) docker run --name "Hello" hellowrld
// Start Hello image world image with "hello" Container name

29) docker create "hello" helloworld
// It will create a container called hello from an image helloword and it won't start the container

30) docker start CID|CNAME
// Start the container

31) docker Stop CID|CNAME
// Stop the container

32) docker pause CID|CNAME
// pause the container

33) docker unpause CID|CNAME
// unpause the container

Container status
⇓

34) docker ps --filter .status = running
= paused
= created
= restarting
= removing

// Docker container status

35) docke
// log

36) docke
// ren

37) dock
// sto

38) doc
// k

39) doc
// de

40) doc
// d

41) doc
// S

42) do
// ge

43) d
//

44)
//

35) docker logs CNAME
// log for that container

36) docker rename CNAME_old CNAME_new
// rename the container.

37) docker stop CID/CNAME
// stop the container [grace period = 10 sec]

38) docker kill CID/CNAM
// kill the container

39) docker container prune
// delete all stopped container [container cache]

40) docker image prune
// delete all dangling & unused images

41) docker system prune
// remove all unused docker object

42) docker inspect container id | volume id
// get more info & details of container & volume

43) docker run -p host-port : cont_port
// map port of container to host machine

44) docker run -it image-name -P
// all ports of container is used to same port of the H-M.

45)
```
docker build -t imag-name --build-arg
        var-name = var-value
```
// arg override at build time.

46)
```
docker run -it -e var-name=var-value
        image-name bash
```
// env override at run-time.

47)
```
sudo DOCKER_BUILDKIT=0 docker
        build -t image-name .
```
// Build image & set build kit=0.

48)
```
docker run -d jenkins/jenkins:lts
```
// run jenkins container

49)
```
docker run -d -p 8070:8080
        jenkins/jenkins:lts
```
// map port.

50)
```
docker volume create volu-name
```
// To create a volume

51)
```
docker volume ls
```
// list the volume.

52) docke

-v

je

-v

Note :-

After

53) do

// re

54) d

// ima

52) docker run -d -p 8090:8080
-v /home/ubuntu/jen-volume : /var/jenkins_home
jenkins/jenkins : lts

- v volume : Container path

<u>Note</u> :- To access the appl^n in the container
After port mapping. [copy pub-ip : host-port]
[Browser]

53) docker system prune --all
// remove all stopped + n/w + cache + n/w
Container Not used build .

54) docker objects
// image, container, volume, n/w + bridge