

DOCKER COMMANDS

MOST USED DOCKER COMMANDS:-

- `docker --version`
- `docker --help`
- `docker pull`
- `docker run`
- `docker build`
- `docker login`
- `docker push`
- `docker ps`(`ps` stands for docker processes, this command used to see active container currently)
- `docker images`
- `docker stop`
- `docker kill`
- `docker rm`(`rm`-stands for docker remove)
- `docker rmi`(`rmi`-stands for remove images)
- `docker exec`
- `docker commit`
- `docker import`
- `docker export`
- `docker container`
- `docker compose`
- `docker swarm`
- `docker service`

BASIC DOCKER COMMANDS

* `docker --version`

This command returns the version of Docker which is installed.

* **docker --help**

1. This command return a list of commands available in Docker along with the possible flags(options) and also with discription.

```
|-----|
|  HELP  |
|  ME    |
|-----|
```

- If we have any doubt respect to uses of any command in docker, than use the help
- Help will bascially tell us different commands that are basically there along with discription.
- It also explain what each and every command does.

docker pull \$ docker pull ubuntu

This command pulls a new Docker image from the Docker Hub.

- --- what we do first
 - We dont have any image in local repository
 - We have to pull it from Docker Hub
- -- When we pull it from Docker Hub, It store in our local repository
- -- Once it is their in local repository we can run docker images command and check all the different images

docker images \$ docker images

This command lists down all the images in your local repo

- After running this command all the images list down

docker run \$ docker run ubuntu(ubuntu- Is our image name)

This command executes a Docker image in our local repository and create a running Container out of it.

---> Now i have to apply this three commands to show what will happen

- `docker pull ubuntu` (By running this command i am pulling the latest ubuntu image from my Docker Hub)
- To check the image is pull or not we run command
- `docker images`
- -- When we hit enter, We have list of images available in our repository

docker build \$ docker build -t MyUbuntuImage

This command is used to compile the Dockerfile,for building custom Docker image

- `pwd` --> to check present working directory
- `mkdir docker`
- `cd docker`
- `sudo gedit Dockerfile`

---> **Dockerfile**:- It is the most important file where we want to build our own custom images, because whatever we want for the application run those dependency are specified in this file.

FROM ubuntu(Base image that we have to download from the Docker Hub)

RUN echo Hii, this is krishna!!!!!!

- ➔ Save this
- ➔ Close Docker file
- ➔ Execute this particular Dockerfile

`docker build -t "mycustomubuntu"`

we done two step so it

step1:- pulling image from docker hub

step2:- RUN echo Hii , this is krishna

docker container

This command is used to perform various operations on the container. Refer to www.docs.docker.com for more info.

SOME OTHER COMMAND USED WITH DOCKER CONTAINER AS A PARENT COMMAND

- **\$ docker container logs**
- **\$ docker container kill**
- **\$ docker container rm**
- **\$ docker container run**
- **\$ docker container start**

1. docker container logs id(to basically find out the different logs that are associated with this id container)
2. docker rm id(to remove this id container)
3. docker container kill(to kill one particular container)
4. docker container run(to start container which is temporarily stop or inactive)
5. docker container start(Again start the container)

docker login \$ docker login

This command is used to login to Docker Hub repo from the CLI

--> **Region why we need to login the region is that...**

If we want to push any of our image that we created locally, so that when we are working with the team who all are using docker, than we just pull the docker image or create docker image from scratch and build a container and if we want to share that container with other people, than we uploaded to Docker Hub.

How to Uploaded to docker hub.

- ❖ **To uploaded we dont have any workaround, we have to do it through terminal to do it**
- ❖ **docker login**
- ❖ **once we have to login using docker container credentials, than we simply start pushing our docker image to the Docker Hub**
- ❖ **Terminal:-**
- ❖ **docker login-- if we dont have Docker ID than create it**
- ❖ **Username:**
- ❖ **password:**
 - **Once login start pushing docker images which we worked locally to our Docker Hub.**

docker push \$ docker push docker id->krishna/imagename- >myubuntuimage

This command pushes a Docker image on our local repo to the Docker Hub

Terminal:-

-- First look the images that we want to upload to my docker hub

docker images

-- I found mycustomubuntu image that i have created

-- pushing this image to Docker hub

-- Copy this

It has name mycustomubuntu and i cannot upload it to Docker Hub with this name, since i have to tag it with my name.

■ Tag this image with my docker id

- docker tag mycustomubuntu krishna/mycustomubuntu
- docker images --> we find mycustomubuntu
 - krishna/mycustomubuntu (this is what i have to upload)
 - docker push krishna/mycustomubuntu(image would be uploaded to Docker Hub)

➔ After this command executed successfully we can go to our docker hub and check that our image which we create locally uploaded to docker hub, It can be accessed by other people.

docker ps \$ docker ps

\$ docker ps -a

This command lists all the running container in the host if '-a' flag is specified, shutdown container are also displayed.

✓ Go to terminal

- docker ps (no any active container)
- ✓ docker ps -a (it show all the container weather they are active or not active)

docker stop \$ docker stop id

This command shuts down the container whose Container ID is specified in arguments. Container is shut down gracefully by waiting for the dependency to shut

docker kill \$ docker kill id

This command kills the container by stoping its execution immediately. its similar to force kill

docker rm \$ docker rm id

This command removes the container whose Container ID is specified in arguments

If we want to remove any container from our host, we have to first stop it by above two commands

- Kill it gracefully
- force kill it

docker rmi \$ docker rmi MyUbuntuImage

This command removes the image whose name has been specified in arguments

Difference between rm and rmi

|

|

It remove container

It remove image from our repository

docker exec \$ docker exec -it id bash

This command is used to access an already running container and perform operation inside the container.

docker commit \$ docker commit id krishna/MyModifiedImage

This command creates a new image of an edited container on the local repository Copy & paste

docker export \$ docker export --output="latest.tar" mycontainer

This command is used to export a Docker image into a tar file in our local system

Export is the second way if we dont want to upload on Docker Hub because the image is very heavy. so we use docker export from one machine and we save that image as a tar file, this tar file imported inside the system , over there it can excess again.

- example:-
- docker images
- docker export --output "mydockerfile"containerid
- ls

docker import \$ docker import /path of that tar file

This command is used to import the contents of a tar file (usually a docker image)into our local repository.

