# docker ps: List the running containers

# docker build -t <name of docker image> <location of dockerfile>: Used to build docker image

# docker ps -a: List all the containers (both running and stopped)

# docker version: Show the version of installed docker

# docker info: Display information related to the Docker installation

# docker images: Show all locally saved docker images

# docker pull <docker-image-name>: Pull/download docker image from docker hub

# docker create <docker-image-name>: Create docker container without starting it

# docker start <container-name>: Start the docker container

# docker run -it --rm ubuntu /bin/bash: Run the docker container in terminal interactive mode

# docker run -d --name <container-name> -p <hostport>:<containerport> <image-name>: Map a port Example: docker run -d --name nginx-base -p 80:80 nginx:latest {D stands for running container background}

# docker tag local\_image/dockerhub\_repo: tag local image with dockerhub image

# docker push dockerhub\_repo: Push local docker image to dockerhub

# docker run -P nginx

# docker logs <container-name>: Show the logs of containers

# docker export <container-name> -o <filename>.tar: Export a container's filesystem as a tar archive

# docker pause <container-name>: Pause all processes within one or more containers

# docker unpause <container-name>: unpause all processes within one or more containers

# docker restart <container-name>: Restart one or more containers

# docker port <container-name>: Display port mapping of container

# docker kill <container-name>: Kill container

# docker stop <container-name>: Stop containers (different from kill)

# docker events: Capture the real time events from server

# Commit changes

apt update && apt upgrade -y

apt install python3

STOP RUNNING THE CONTAINER

docker ps -a

docker commit <container-id> <new-docker-image>

# docker inspect <container/image/network/volume>: Get detailed (configuration) info about container or image

# docker diff <container-name>: Show all modified files in a container Status- A(added), C(changed), D(deleted) file/dir

# docker rename <old-container-name> <new-container-name>

# docker stats: Show stats of running containers

# docker save <image-name> <file.tar>: Save docker image to tar file

# docker load -i <tarfile>: Load docker image from tar file

# docker rmi <image-name>: Delete docker image

# docker top <container-name>: Show process of containers

# docker rm <container-name>: Delete a container

# docker rm -f <container-name>: Delete a running container

# docker stop <container-name>: Stop a container

# docker exec -it <running-container-name> /bin/bash

# docker attach --sig-proxy=false <container-id> and to detach hold ctrl, then press P, then Q and then release ctrl

docker attach --detach-keys="ctrl-x" <container-id>

# docker cp <container-name>:source target ---> Copy a file from container to host Example: docker cp kind\_golick:/home/pranjal.txt p.txt

# docker cp <container-name>:target ---> Copy a file from host to container Example: docker cp p.txt kind\_golick:/home/hw.txt

# docker history <image-name>: Show the history of a docker image

# docker update --cpu-shares 512 -m 300M <container-name/id>: To update the container configuration ---> check the container configuration using docker inspect <container-name>

# Setup a Docker Nginx reverse proxy server

docker run -d --name nginx-base -p 80:80 nginx:latest

docker ps -a --filter "ancestor=27625053029c"

docker volume create psvolume1

docker volume create psvolume2

docker run --mount source=psvolume1,target=/app/data1 --mount source=psvolume2,target=/app/data2 ubuntu

docker run -d --name flaskapp--mount source=psvolume1,target=/app/data1 flaskapp

Bind Volume

docker run -v %cd%:/home/pranjaldata ubuntu