```
docker version
docker info
docker login --username vatadepalli


docker container run -it -p 80:80 nginx
                -it : interactive mode
                -p / --publish : publish port


docker pull nginx
                kinda like - git clone - for docker images
                happenens automatically in - container run
                        when the image is not present locally

docker container ls
                shows running containers

docker container ls -a
                shows all containers, whether running or not

docker stop containerID
                Sends SIGTERM signal to stop
                Waits for 10 secs, and issues kill signal

docker kill containerID
                Sends SIGKILL

docker start containerID
        -a      - logs

docker logs imageID



docker container rm 3bd
                3bd - is the 1st 3 chars of container ID
                removes the container

docker system prune
                Removes stopped containers, build cache (images
downloaded from hub)

docker images
                shows all local images

docker image rm f68
                3bd - is the 1st 3 chars of image ID
                deletes the image

docker rmi $(docker images -a -q)
        Docker Remove all Images

docker logs containerID
```

```
-------------- Detached & Name -------------------------------
docker container run -d -p 8080:80 --name mynginx nginx

                -d : detached, runs the container in background
                -p : port
                        8080 - external system port
                        80 - container internal port
                --name : name of the container

docker ps
docker ps -a
        same as - docker container ls

docker stop mynginx
        stops a container

docker container rm mynginx -f
        force delete a running container using name

docker container rm mynginx
        delete a stopped container using the name

docker rm $(docker ps -aq) -f
        deletes all the containers running or not

------------ ENV VARIABLES -----------------------------------

docker container run -d -p 3306:3306 --name mysql --env
MYSQL_ROOT_PASSWORD=Trachea@88 mysql

        --env : adds ENV VARIABLES

------------ SSH into Container ------------------------------

docker container exec -it mynginx bash

        you can do all linux stuff here
        you can ssh into - Running containers only

exit

        to get the hell out of the container

------- MOUNTING ------------------------------------------------

docker container run -d -p 8080:80 -v $(pwd):/usr/share/nginx/html
--name nginx-website nginx

                -v $(pwd):/usr/share/nginx/html
                        $(pwd) - Binds present local directory &
                        /usr/share/nginx/html - container
directory
```

```
------------ BUILDING IMAGE ---------------------------------

First wite a - Dockerfile
                FROM nginx:latest
                WORKDIR /usr/share/nginx/html
                COPY . .

docker image build -t vatadepalli/nginx-website .
                build image

docker images
                list images

docker push vatadepalli/nginx-website
                push to dockerhub

-----------  EXEC ------------------------------------------------

docker exec -it _containerID _command

                -i              attach terminal
                -t              make the text look good

----------- CMD LINE ---------------------------------------

docker exec -it _containerID sh

        CMD + D  Get the hell out
        exit

docker run -it _image sh

                docker run -it busybox sh

------------ BUILDING CUSTOM IMAGES ----------------------------

Dockerfile

docker build .

docker build -t vatadepalli/nginx:latest .

        -t              name of img

docker commit -c 'CMD ["redis-serger"]' _containerID

                        create image from a container
                        less ususal approach

------------ DOWNLOAD IMAGE FROM REPO ------------------------

docker pull _imageName
```

```
---------- DOCKER-COMPOSE -----------------------------------

docker-compose.yml / yaml

docker-compose up -d

docker-compose down

                          version 3 of docker-cli seems common

docker-compose ps

------------ PRODUCTION GRADE ENVIRONMENT ----------------------

docker build -f Dockerfile.dev . -t _myDockerImage

docker run _myDockerImage -p _ext_port:_int_port --name
_containterName

------------ DOCKER VOLUMES -------

docker run -p 80:5000 -v /app/node_modules -v $(pwd):/app
_imageID_name

                -v /app/node_modules              This folder is
inside container. Don't copy
                -v      _path_to_local:_path_container     Mapping
volume
```