jenkins web ... procedure to create

①

Docker

installation

>sudo apt-get install docker.io ↵

below commands ensure docker starts after reboot

>sudo systemctl start docker ↵
>sudo systemctl enable docker.

>we need to add our user to ~~current~~ docker group.
sudo usermod -a G docker $USER.

>groups ↵ → this will show the usergroups available.

we need to create a docker group as well.
>sudo groupadd docker ↵ (it will create docker usergroup)

group change may require a restart to reflect changes

Done we are ready to ~~run~~ use docker

1. docker run hello-world ↵
2. docker pull busybox → it will pull busybox image.
3. docker run busybox ↵ } these will run the
4. docker run busybox echo hello ↵ } command & exit
5. to run docker in interactive mode run below
   docker run -it busybox ↵ it will give busybox prompt to us.
6. docker ps → will show current running ~~images~~ containers
   docker ps -a → will show all containers that are running
   docker ps -aq and exited
7. docker images ↵ gives all local images.
8. docker run -d image-name ↵

it will run our image in container & provide the container id as output ~~container~~ image will be running in that container in background.
-d is option for daemon process.

②

To stop a container

>docker stop contain...
>docker rm contain...

Image - is collectio...
→ images are having ...
→ an image is re...
→ container is a ...
running in read-...
→ docker run sto...

→ images ...
→ layers are ...
one layer con...
→ containers ...
i.e. instance ...

Searching an ...
>docker sear...

docker ps -l

To commit ch...
>docker commi...
or
run your fo...
more chan...
then run ...
docker ...
>~~docker commit~~ ...
or ~~docker~~ ...
docker com...
~~or~~ ~~do~~ ...
>docker t...

(7)

To stop a container:-

> docker stop container-id ↵ it first executes terminate & then kill.
> docker kill container id ↵ kill is faster than stop.

Image:- is collection of file and metadata.

→ images are having layers.
→ an image is readonly filesystem.
→ container is a encapsulated set of processes running in read-write copy of image.
→ docker run starts a container from image.

* → * images are conceptually same to class.
→ layers are same to inheritance where one layer can obtain features from other.
→ containers are same to instances of class. i.e. instance is in memory at runtime.

Searching an image.

> docker sear zookeeper ↵ → it will search on specified repo or docker hub.
imagename

· docker ps -l > gives last run image -> container.

* To commit changes to original image:-

> docker commit id ↵ (id is image id).

or

run your favourate image of your app / and make changes. & exit.
then run above command.
docker ps -l → gives last ~~const~~ executed container.

> ~~docker commit container-id ↵~~

or ~~docker commit container id~~ new-image-name.

docker commit image-id ↵ (imageid is new ~~or~~ ~~do~~ imageid we will get it with docker images)

> docker tag IMAGE-ID New-Name-for-image.

(8) * To remove container from list
> docker rm container-id ↵ (after this docker ps -a) will not able to find that container.

* To remove image from list.
> docker rmi image-id ↵

* building a new Image from Dockerfile.

(1) create a directory,

(2) create a file "Dockerfile"
e.g. content.

FROM Ubuntu (image name).
RUN get-update.
RUN ~~get~~-get install wget -y
save & exit.

then run.
> docker build -t new name-for-image . (done).

> docker history image-name ↵
this will give us history of updates on that images.

* publishing our docker container service.
> docker images ↵
> docker run image-name -P
> docker run image-name -publish-all
> docker run -d -P 88:8000 image-name.
> ~~docker port~~ ↳ container port.
> docker port
local port

> docker -rmi $(docker images) } removing all containers
> docker -rm $(docker ps -a) } & images.

* uploading Docker image:- create a/c on docker hub.
> docker login (enter username & password & email
> docker tag image username/image-name.
> docker push ~~image~~ username/imagename.

- add dependency sp
- add mysql depndcy -
- what is use of Entity

- spring-boot-starter-data-jpa
- add dependency spring boot data jpa
- add mysql depndy - mysql → mysql-connector-java.
- what is use of EntityManagerFactory ?