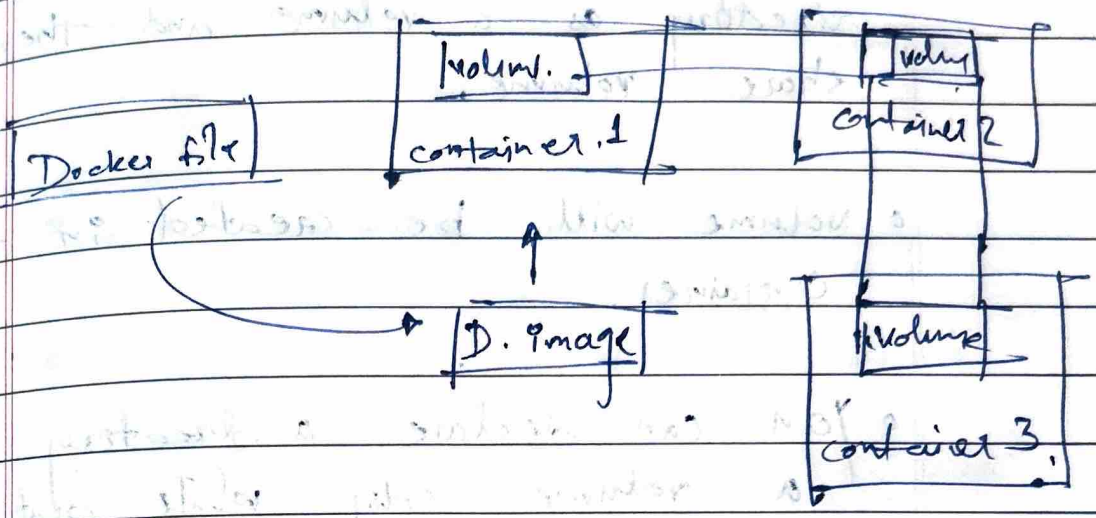
Lecture 06

Volume is a directory that can be shared with others.



Volume can be shared from container to container.

& also from host to container.

PAGE NO.
DATE :

- Volume is simply a directory inside our container.
- Firstly, we have to declare this directory as a volume and then share volume.
- Volume will be created in one container.
- You can declare a directory as a volume only while creating container.
- You can't create volume from existing container.
- You can share one volume across number of containers
- Volume will not be included when you update an image

You can mapp volume in two ways.
① Container → Container
② Host → Container.

Benefits of Volume

- Decoupling Containes from storage
- Share Volume among different containers
- Attach Volumes to Containes
- On deleting Containes volume does not delete.

Dockerfile

```
FROM ubuntu
VOLUME ["/file path"]
```

sharing volume.

#docker run -it --name containername 2 --privileged=true --volume-from containername 1 ubuntu /bin/bash

PAGE NO.
DATE :

→ lab C ←→ C.

[With docker file]

yum install -y docker
service docker start
touch file1 file2
vi Dockerfile
FROM ubuntu
VOLUME ["/myvolume"]
Docker build -t myimage .
docker images.
docker run -it --name mycont1 myimage /bin/bash
ls
cd myvolume
touch 1 2 3 4
ls.
exit.
docker run -it --name mycont2 --privilege=true --volume-from mycont1 ubuntu /bin/bash:
ls.
cd myvolume.
ls.

With Command line

--name

docker run -it 1 container3 -v /volume2 ubuntu /bin/bash
ls
cd /volume2
touch file1 file2 file3 file4
exit
docker run -it --name container4 --privileged=true --volume-from container3 ubuntu /bin/bash
ls
cd /volume2
ls.

PAGE NO.
DATE :

With Command line

Host ↔ Cont

```
# pwd
  /home/ec2-user
# ls
# mkdir /volume5
# docker run
# touch /volume5/filexyz
# docker run -it --name cont
  -v /home/ec2-user/volume5:/rajput
  --privileged=true ubuntu /bin/bash
# ls
# cd /rajput
# ls.
```

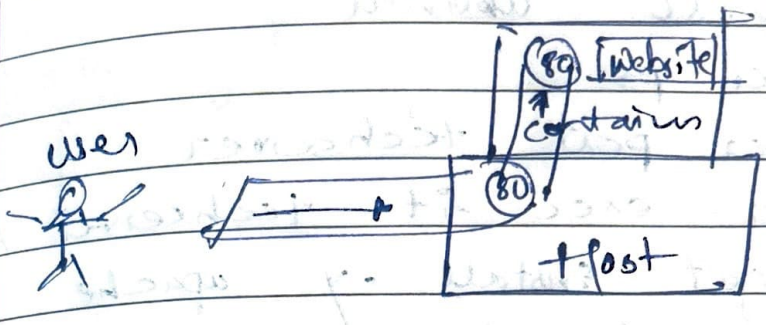
local m/c → Container m/c

Commands

```
# docker volume ls
# docker volume create <volumename>
# docker volume rm <volumename>
# docker volume prune
    → Removes all unused volumes.
# docker volume inspect <volumename>
# docker Container inspect <volumename>
```

Lecture 07

Logical ports : 0 to 65535



expose : Cont to Cont in same host
publish : Cont to outside too (like internet)
" -p includes expose"

docker run -td --name techserver
-p 80:80 ubuntu.
↑ daemon
publish /port

docker port techserver
- Available : ports

docker exec -it techserver . /bin/bash
↑ take inside Cont. with new process (attach uses old process)

Port of local machine can be changed
But not of containers ???

PAGE NO.
DATE :

→ lab

yum install dockey -y
service docker start
docker run -td --name tech
-p 80:80 ubuntu
docker ps
docker port techserver
docker exec -it techserver /bin/bash
apt-get install -y apache
cd /var/www/html
vim index.html
"Hello everyone"
exit # service apache start

Copy IP to browser. :80

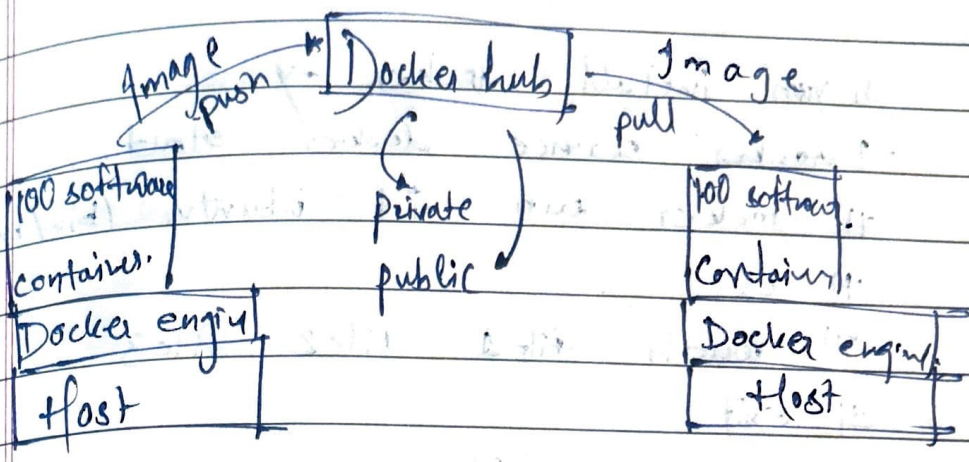
exit
docker run -td --name jenkins
-p 8080:8080 jenkins
docker port jenkins.

Copy IP to browser :8080

#

If Container is stopped or deleted all the ports are unmapped.

Lecture 08



① stopping all running containers :

docker stop $(docker ps -a -q)

runs in loop — quit

② Delete all stopped Containers :

docker rm $(docker ps -a -q)

③ Delete all images :

docker rmi -f $(docker images -q)

forcefully

PAGE NO.
DATE :

→ Lab

yum install docker -y
docker service docker start
docker run -it ubuntu /bin/bash
ls
touch file1 file2 file3
exit
docker commit
docker ps
docker commit cont.name image1
docker images
#

go to google
hub.docker.com
sign-up

docker login
username
password.

docker tag image1 technicalguftgu/project1
username tag

docker push technicalguftgu/project1

going to another instance & starting docker service.

docker pull technicalguftgu/project1
docker images.
docker run -it id ——————
ls.

You can change image into private by going into docker hub settings.
Then while pulling image you have to run first "# docker login" command & log in into docker hub.