**Docker Document**

**Docker properties:**

1. Docker file –image,dependency,commands,port
2. Docker image – build
3. Docker container – run docker image
4. Docker hub – central repository

Launch instance

apt-get update

apt-get install docker.io

go docker hub

copy the httpd

paste the server

docker pull httpd

how to identify:

docker images

docker run –itd --name apachecontainer –p “8090:80” httpd

docker ps

copy server ip paste google: ip:8090

docker version

docker stop (ID)

docker start(ID)

docker ps –a

docker rm (ID)

docker ps –a

docker rm –f (ID)

docker rmi (ID)

docker exec –it (ID) /bin/bash

ls

exit

docker logs (ID)

docker top (ID) – a process

docker stats (ID) – memory and cpu

docker inspect (ID) – complete details (docker container details)

docker info – running,paused,stopped (docker machine details)

**if developer asking for wget and vim packages:**

mkdir fileforvim

cd fileforvim

vi Dockerfile

FROM ubuntu

MAINTAINER buvan

RUN apt-get update

RUN apt-get install vim -y

:wq!

docker build –t vimimage .

docker images

docker run –itd --name vimcontainer vimimage

docker ps

docker exec –it (ID) /bin/bash

vim

:wq!

**If the developer asks git:**

vi Dockerfile

FROM vimimage

MAINTAINER damo

RUN yum install git –y

:wq!

docker build –t vimimage .

docker ps

docker rm –f (ID)

docker run –itd --name vimcontainer vimimage

docker ps

docker exec –it (ID) /bin/bash

git

vim

wget

**how to restrict the container access to the developer:**

cd ..

mkdir sshd

cd sshd

vi Dockerfile

FROM ubuntu

MAINTAINER buvanesh

RUN apt-get update

RUN apt-get install vim git wget openssh-server -y

RUN sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd\_config

RUN echo 'root:root123' | chpasswd

RUN mkdir /var/run/sshd

CMD ["/usr/sbin/sshd", "-D"]

EXPOSE 22

:wq!

docker build -t sshdimage:v1.0 .

docker images

docker run -itd --name sshdconatiner -p "2323:22" sshdimage:v1.0

docker ps

**login instance developer:**

paste IP port:2323

lagin as: root

password:root123

exit

mkdir httpd

cd httpd

vi Dockerfile

FROM ubuntu

MAINTAINER buvan

ENV DEBIAN\_FRONTEND=noninteractive

RUN apt-get update && apt-get install -y apache2

RUN mkdir /var/run/apache2

ENV APACHE\_RUN\_USER www-data

ENV APACHE\_RUN\_GROUP www-data

ENV APACHE\_LOG\_DIR /var/log/apache2

ENV APACHE\_RUN\_DIR /var/log/apache2

EXPOSE 80

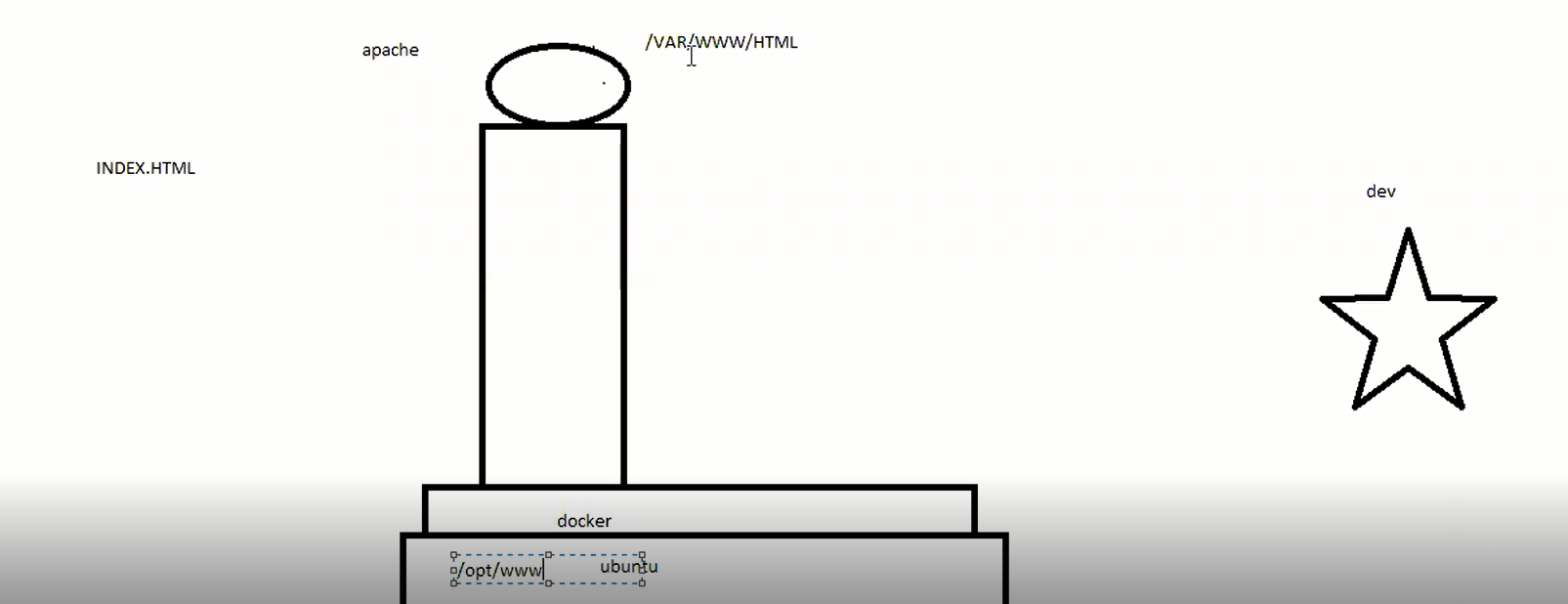
CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]

docker build –t apache2 .

docker run –itd --name apache –p “8090:80” apache2

docker ps

**if the developer asks to give permission to the /var/www/html path for the index file:**



cd ..

mkdir /opt/www

echo "welcome to buvanesh" > /opt/www/index.html

docker run -d -p "8085:80" --name=webserver5 -v "/opt/www/html" apache2

**docker volume:**

cd ..

docker volume create my-vol

docker volume ls

docker inspect my-vol (identify volume path)

cd /var/lib/docker/volume/my-vol/\_data

ls -lrt

docker run -d -it --name=damo --monut source=my-vol,destination=/usr/local/apache2/htdocs httpd:2.4

touch damo.txt

docker exec -t 72127be32a99 /bin/bash

cd /usr/local/apache2

cd htdocs/

ls –lrt

docker ps

docker rm –f (ID)

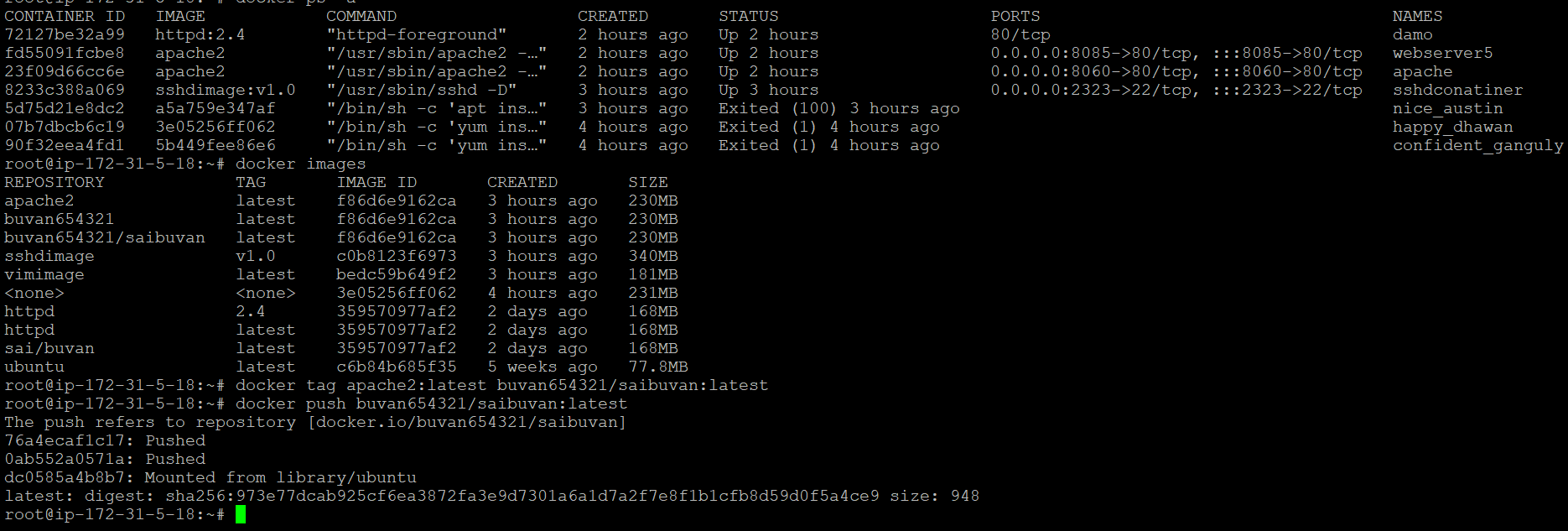
docker ps

docker Hub:

docker login

username:

password:



docker tag centos sai/Buvanesh

docker push sai/Buvanesh

Container backup:

docker commit (ID) backupcontainer

docker images

docker save -o ~/container1.tar backupcontainer (image backup as a file)

ls –lrt

docker rmi backupcontainer

docker load -i container.tar

docker images

docker run -itd --name apache1 -p "8070:80" apache2

docker exec -it (ID) /bin/bash

apt-get install vim -y

apt-get install git -y

docker commit (ID) apachevimgit - Container change as a image

docker images

**launching database container**

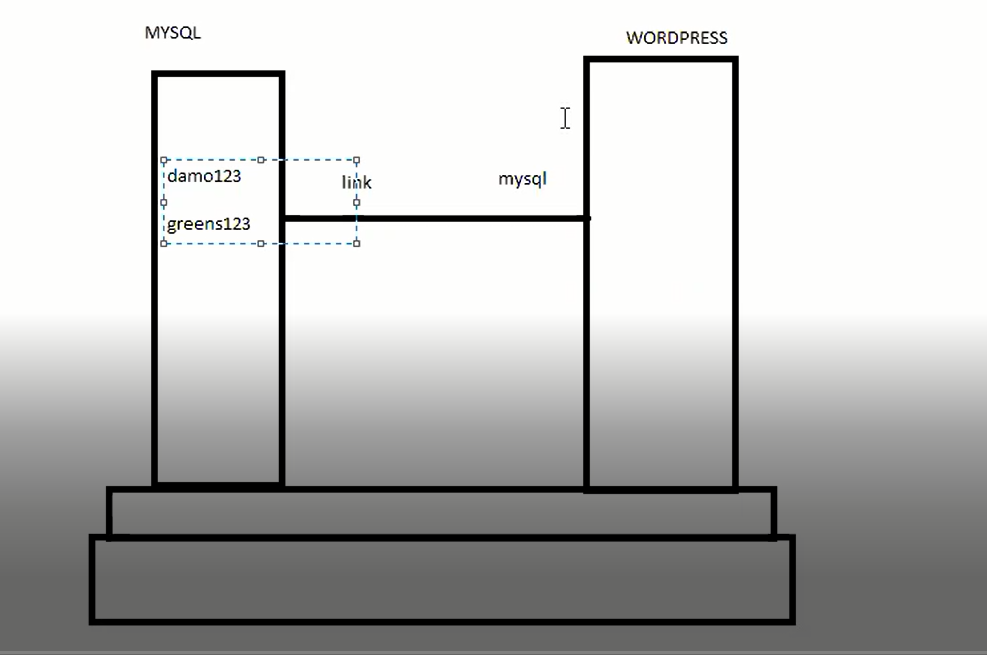
docker run -d --name db mysql:5.5

docker log (ID)

docker container prune (stopped container removing purpose)

docker run -d --name db -e MYSQL\_ROOT\_PASSWORD=sai123 mysql:5.5

container to container speak



docker run -d --name myweb-server --link db:mysql -p "8060:80" wordpress

docker ps

public image customizes docker file:

mkdir mynginx

cd mynginx

vi Dockerfile

FROM nginx

MAINTAINER sai

WORKDIR /usr/share/nginx/html

COPY index.html /uer/share/nginx/html

:wq!

vi index.html

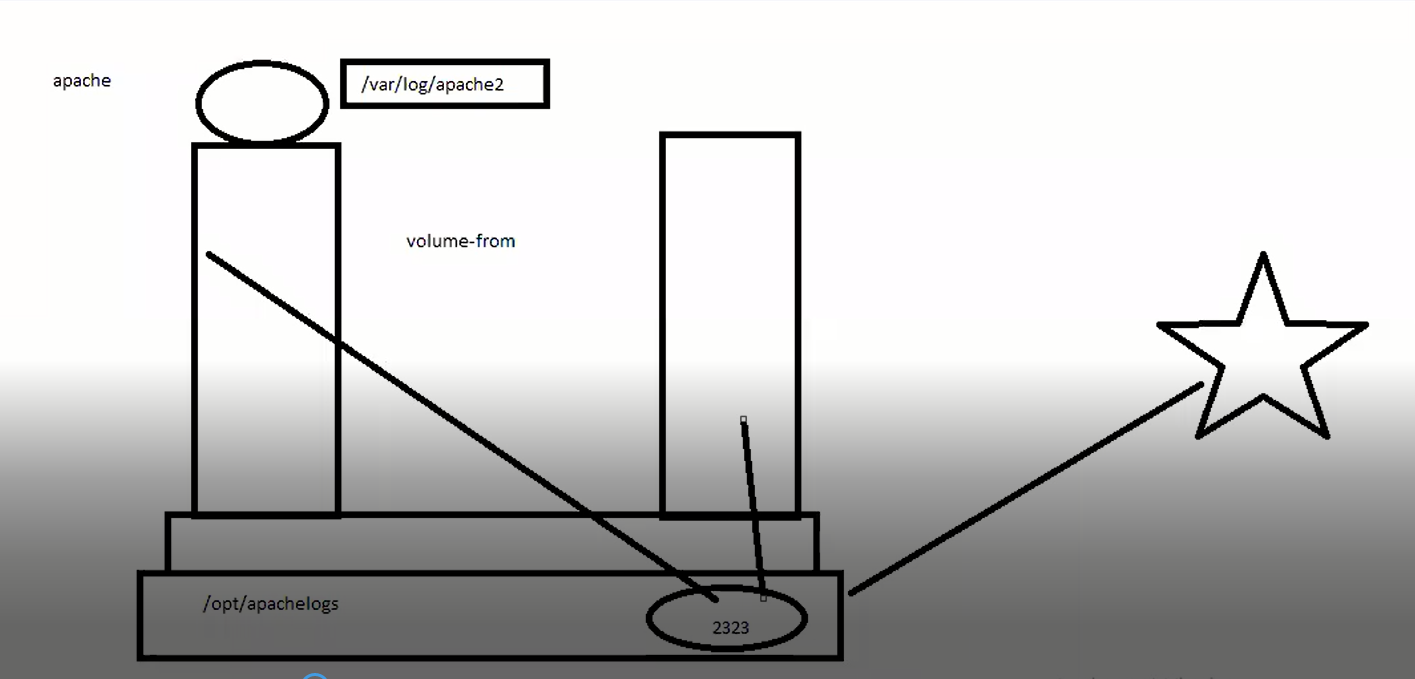
this my page

docker build -t mynginx .

docker run -itd --name mynginx -p "8040:80" mynginx

copy IP past Google

**restrict the container access:**



docker run -d -p "8090:80" -v /var/log/apache2 --name=webserver2 apache2

docker run -d -p "2222:22" --volumes-from webserver1 sshdimage:v1.0

login ec2 developer:

ip:2222

username: root

password: root123

cd /var/log/apache2

ls -lrt

tail -1000f access.log

how to find container IP:

docker inspect (ID)

docker network ls

docker network inspect (bridge netwok ID)

docker run -it -d --name c1 alpine

docker run -it -d --name c2 alpine

docker exec -it c1 sh

ping 172.17.0.8

docker network create --driver bridge demo\_net

docker run -it -d --name A1 --network damo\_net alpine

docker run -it -d --name A2 --network damo\_net alpine

docker exec -it A1 /bin/bash

ping A2

docker compose yml:

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

vi docker-compose.yaml

vi docker-compose.txt

mv docker-compose.txt docker-compose.yaml

version: '3.9'

services:

mysql:

image: mysql:8.0

volumes:

- mysql\_data:/var/lib/mysql

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: wordpress

MYSQL\_USER: wordpress

MYSQL\_PASSWORD: wordpress

volumes:

- ./mysql:/var/lib/mysql

phpmyadmin:

image: phpmyadmin/phpmyadmin

depends\_on:

- mysql

environment:

PMA\_HOST: mysql

PMA\_PORT: 3306

PMA\_ARBITRARY: 1

restart: always

ports:

- 8183:80

wordpress:

depends\_on:

- mysql

image: wordpress:latest

ports:

- "8020:80"

restart: always

environment:

WORDPRESS\_DB\_HOST: mysql:3306

WORDPRESS\_DB\_USER: wordpress

WORDPRESS\_DB\_PASSWORD: wordpress

WORDPRESS\_DB\_NAME: wordpress

volumes:

- ./wordpress:/var/www/html

volumes:

mysql\_data: {}

docker-compose up -d

docker-compose ps