Docker Continue

Day12

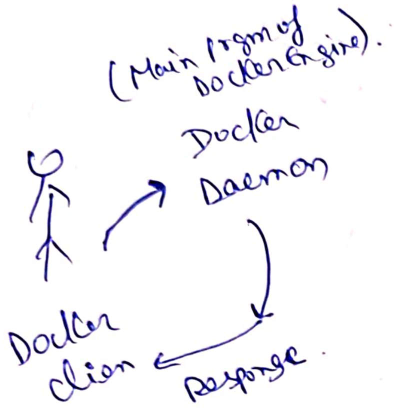
⇒ Launch an Instance (Default Security Group).

⇓

Docker installation (Engine)

⇓

Docker Daemon. (-d detach Mode).



⇓

Start Docker.

⇓

Create a Container

⇓

docker pull httpd

⇓

docker attach --name. (Modifications).

Base command was bash command

⇓

docker exec -it OS1 bash.

which command

Not Base command

docker run -it --name OS1 httpd.

⇓

docker run -dit --name OS2 httpd.

detach mode

Docker is very fast, launch, delete etc Containers very fast

Container with same replica

(Image can be same but Replica/Image can be same).

Shut down gracefully

whatever command we are running in the container is, the httpd-foreground whenever we attach command they always attach to the running command in the container. Currently running command is httpd and it does not provide any terminal

One more container with amazon linux

⇓

Exit command. will always exit the bash main program, program will Not exit.

docker run -it amazonlinux.

⇓

Base command is bash by default.

⇓

docker attach OS2 (fail → Base command).

docker attach -- (work). (exit)

⇓

docker exec -it funy-yalow bash

⇓

exit

Container life cycle ≃ Container

New Command ⇒ docker exec -it funy-yalow date

Create a new session

If we do or run 2 Bash commands then only lastest Bash will be exit & container will not be stopped

Pblm whenever need to launch a container, Manually go to the container & Make changes.

⇓

Base System

touch index.html.

⇓

docker ps

*

Copy file from Base System to Container

docker cp index.html OS3:/usr/local/[redacted] apach2/htdocs/

source — destination. → Directory path.

from Host to Container

Copy from containers to Base System

docker cp funny yalow:/test.txt ⊙ → current location Base System.

```
docker cp <container_name_id>:<source_path> <destination_path>
```

(from Container to Host)

Mount a particular Dir to the container (Not everytime cp commad).

Base System ⇒ mkdir. data
⇓
index.html

-v /data/:

```
⇒ docker run -dit --name webserver -p 81:80
    -v data/:/usr/share/nginx/html/ nginx.
```

-v → Volume

data/: → Mounting

81 → Port of Base System

80 → Port 80 of Container

nginx

webserver

```
docker rm -f --
```

forcefully

⇓

Again run above same commad

Can create multiple containers mounting to same directory.

Vice-Versa as well ⇒ Launch one more instance from Amazonlinux whatever will do in the container, will be mounted to base system Directory as well.

⇒ docker volume ls

⇓

* docker volumes create --help → Can be Google Drive

docker volume create --driver local gfgvol

⇓

docker volume ls ↓

⇓

docker run -dit -v gfgvol:/htdocs code httpd.

⇓

cd /var/lib/docker/volumes/gfgvol

/data was Managed by me

Instead of creating Manual Dir, use Docker volumes

* Images* (Custom Image).

⇒ docker ps
⇒ go to one container.

⇓ ls ⇓

ifconfig ↓ Not present.

⇓

System level Softwares/ other config

yum whatprovides ifconfig.

⇓

yum install nettools

⇓

yum install httpd.

Server (Code)

Pip libraries.

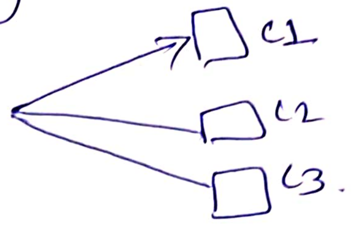
Python runtim

Sytem⇒ nettools, httpd, pip3, vim.

Adding Multiple layers.

amazon linux

(Base Image) | Recipe



⇓

ifconfig I

⇓

rpm -q httpd I

⇓

yum install python3 I

⇓

yum install pip3.

⇓

pip install pandas.

⇓

vi a.py I

⇓

yum install vim

⇓

pip3 list.

⇓

Exit out of this containe → from container

will try to create image ← docker commit -m " ________ " webserve2

Container Name

gfg image : V1

version

docker run -dit --name test 1 gfgimage :V 1

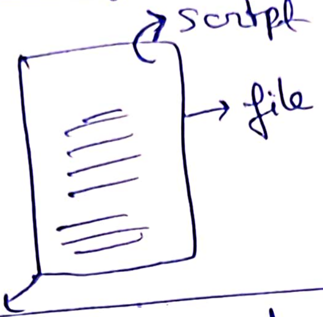
⇓

docker attach.

```
rpm -q httpd.
  ⇓
ifconfig
```

] Custom Recipes.

New way of creating Image from Script



for this we have Dockerfile

```
FROM amazonlinux

RUN yum install -y httpd net-tools vim
                                 pytho3-pip.

LABEL maintainer= gfg.com.
LABEL os= linux.

ENV x=100

ENV url=gfg
WORKDIR /var/www/html.
COPY index.html . ~~/var/www/html~~ .

WORKDIR

EXPOSE 80

CMD ["date"] ⟹ ]
```

maintainer of this Image.

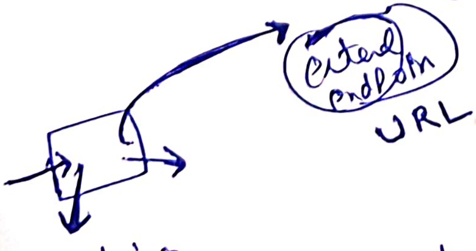
Build Time / Run time.

. → in my current location.

Very Imp

EXPOSE 80 → Anyone from outer world can connect at Port No 80.

CMD ↓ Command which this image would execute.



URL

⇓

docker.

container.

Read Dockerfile and build Image

docker build -t gfgwebimage .]   Enter

↓
tag

RUN
↓
Build Time
will be
executed

⇓

docker run gfgwebimage]

CMD → can be override
↓
at Run time

Commands running in the Container should be running in foreground

ENTRYPOINT → Cannot be override.

```
CMD ["-D", " FOREGROUND"].
```

Push Image to Docker Hub

from CMD

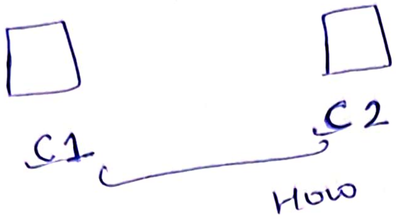
```
docker login

docker tag gfgwebimage : V1    jimmy1/ gfgdevops33sample : V1

docker push ______ :V1
```

```
docker cp <source-path> <contain-name-or-id>: <destination-path>
```

Docker Networking



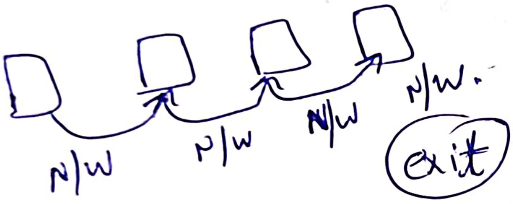
How they can communicate to each other.

Create 2 containers

yum install inputils. → for ping command

⇓ yum install net-tools. > /dev/null.

Ping IP of C2 (They Have N/W Connections using Docker).



Exit

⇒ docker network ls (bridge where containers got created).

Custom N/W

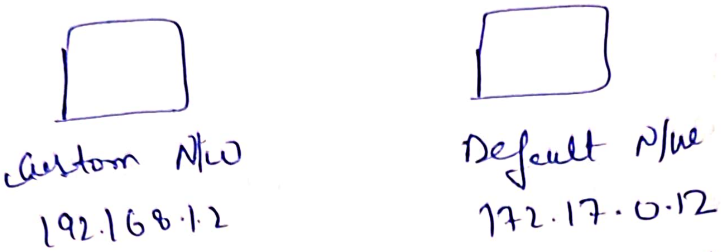
⇒ docker network create --help.

⇒ docker network create --driver= bridge --subnet= 192.168.1.0/24 → first 24 bits N/W name g-fgnet.

⇓ docker network ls (for security purpose)

172.17.0.8

Base System communicate with both the Containers.



Website would be very fast and less latency.

Container to have Base O/S Networks

⇓

Add this in host N/W's.

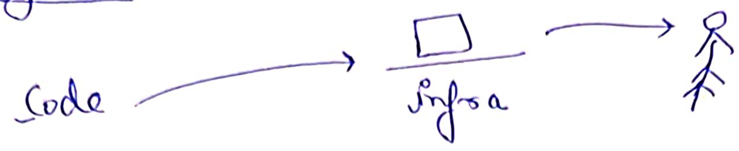
Good idea to store data outside the Container

⇛ docker ps -a -q] All containers Id's.

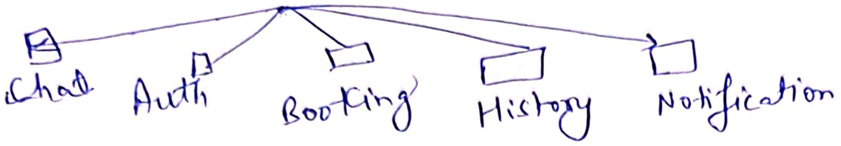
echo `date`

V.Imp ☆ docker rm -f `docker ps -a -q`

flow Diagram.



. Microservices Architecture.



OS → Virtualization.

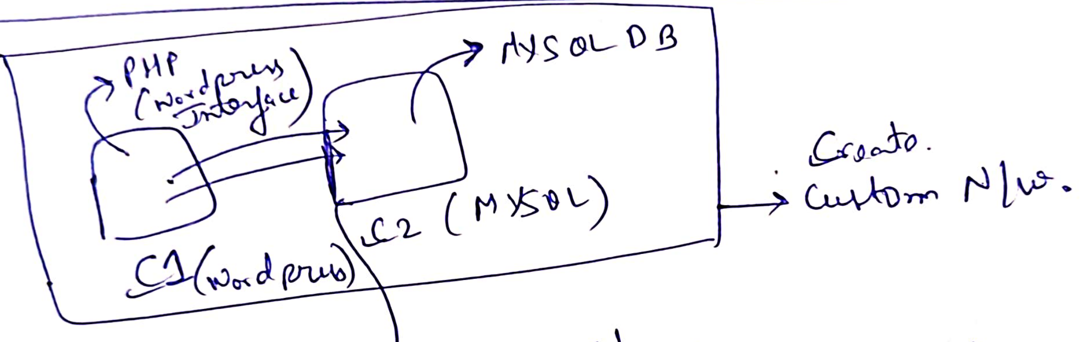
Hardware } AWS

Launch New Instance, (t2.medium)

Wordpress Image pull

Custom N/w , Custom Volume

Host Wordpress Server In EC2 Instance



Username | Host.
password |
(Can set custom credentials).

⇒ Need to create a volume and Mount that volume to MySQL Container. (/var/lib/mysql).

⇒ docker ~~create~~ volume create abhivol.

⇒ docker network create --driver bridge --Subnet=CIDR. gfgnet.

⇒ docker run -dit --name db -e ~ -e ~ = ~. -e ~ = ~ -e ~ = ~ -e ~ -e — --network gfgnet -v abhivol : /var/lib/mysql