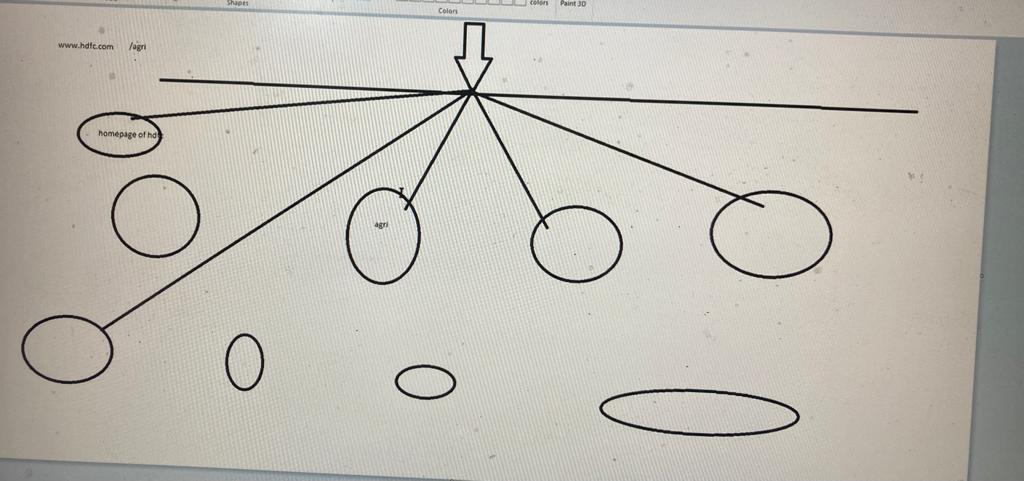
Docker storage: it is a volatile storage

Docker volume can be mounted to the container.



Each session will have a session id (sticky sessions will make the connections to be id will be saved and it will use the same session again and again. if connections are coming again it will redirect to the same application.)

Docker compose: it is a wrapper on docker.doing multiple things at a time.

We will compose a new file.

Docker compose file

Version:’3’

Service:

Sample1:

Image:httpd

Port:

-“80:80”

Sample2:

Image:nginx

Docker run –d –p 80:80 –name sample1 httpd

Docker run –d –name sample2 nginx

Deploying word press:

Version:’3.3’

Service:

Db:

Images: mysql5.7

Volumes:

-db\_data:/var/lib/mysql

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: somewordpress

MYSQL\_DATABASE: wordpress

MYSQL\_PASSWORD: wordpress

Wordpress:

Depends\_on:

-db

Image:wordpress:latest

Ports:

-“8000:80:

Restart: always

Environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: wordpress

WORDPRESS\_DB\_PASSWORD: wordpress

Volumes:

Db\_data:

Docker volume create db\_data

Docker run –d –mount source=db\_data,destion=/var/lib/mysql --e MYSQL\_ROOT\_PASSWORD: somewordpress –e –name db mysql

Docker run –a –link db –p 8000:80 –e wordpress\_db\_host=db:3306 –name wordpress wordpress:latest

Yaml:

Map

List

Maps: when we map a key to a value in yaml files, they are termed as maps.

<key>:<value>

For example:

Name:Intellipaat

Cource:Devops

Yaml list are the sequence of objects.

Args:

-arg1

-arg2

-arg3

-arg4

Example:

Args:

* Sleep
* “1000”
* message
* “Bring back Firefly”

Toolos: intellij,notepad++

Notepad++language-🡪yaml

root@mani-VirtualBox:~# docker pull ubuntu

root@mani-VirtualBox:~# mkdir /data

root@mani-VirtualBox:~# df -h

Filesystem Size Used Avail Use% Mounted on

tmpfs 544M 1.6M 542M 1% /run

/dev/sda3 39G 11G 27G 29% /

tmpfs 2.7G 0 2.7G 0% /dev/shm

tmpfs 5.0M 4.0K 5.0M 1% /run/lock

/dev/sda2 512M 5.3M 507M 2% /boot/efi

tmpfs 544M 2.5M 541M 1% /run/user/1000

root@mani-VirtualBox:~# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

ubuntu latest 27941809078c 6 weeks ago 77.8MB

root@mani-VirtualBox:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

root@mani-VirtualBox:~# docker run -it -d -v /data:/tmp ubuntu /bin/bash

ecfee11ce9bcae8b692bb5e9830686bcc5684f752d65e6a37efadece051adbac

root@mani-VirtualBox:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

ecfee11ce9bc ubuntu "/bin/bash" 5 seconds ago Up 4 seconds hungry\_jang

root@mani-VirtualBox:~# cd /data

root@mani-VirtualBox:/data# ls

root@mani-VirtualBox:/data# ls -alrth

total 8.0K

drwxr-xr-x 21 root root 4.0K Jul 24 11:22 ..

drwxr-xr-x 2 root root 4.0K Jul 24 11:22 .

root@mani-VirtualBox:/data# touch 1 2 3 4 5 6 7

root@mani-VirtualBox:/data# ls -alrth

total 8.0K

drwxr-xr-x 21 root root 4.0K Jul 24 11:22 ..

-rw-r--r-- 1 root root 0 Jul 24 11:24 7

-rw-r--r-- 1 root root 0 Jul 24 11:24 6

-rw-r--r-- 1 root root 0 Jul 24 11:24 5

-rw-r--r-- 1 root root 0 Jul 24 11:24 4

-rw-r--r-- 1 root root 0 Jul 24 11:24 3

-rw-r--r-- 1 root root 0 Jul 24 11:24 2

-rw-r--r-- 1 root root 0 Jul 24 11:24 1

drwxr-xr-x 2 root root 4.0K Jul 24 11:24 .

root@ecfee11ce9bc:/# cd tmp

root@ecfee11ce9bc:/tmp# ls

1 2 3 4 5 6 7

root@ecfee11ce9bc:/tmp#

all the files created in data will store /tmp of the container. Because we have mounted it hear.

Same as vice versa if you create files in containers.

root@ecfee11ce9bc:/tmp# touch e r t y u

root@ecfee11ce9bc:/tmp# exit

exit

root@mani-VirtualBox:/data# ls

1 2 3 4 5 6 7 e r t u y

root@mani-VirtualBox:/data#

files created in the container is seen in this /data

root@mani-VirtualBox:/data# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

ecfee11ce9bc ubuntu "/bin/bash" 6 minutes ago Up 6 minutes hungry\_jang

root@mani-VirtualBox:/data# docker stop ecfee, docker rm ecfee

ecfee

root@mani-VirtualBox:/data# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

root@mani-VirtualBox:/data# ls

1 2 3 4 5 6 7 e r t u y

root@mani-VirtualBox:/data#

When we are stopping and deleting this container also files will be located internally.

/data ---🡪 is available on host machine (dependencies and disadvantages of using this).

We can use that /data to that specific host only.so we are moving to volumes

root@mani-VirtualBox:~# docker volume create myvol

myvol

root@mani-VirtualBox:~# docker volume ls

DRIVER VOLUME NAME

local myvol

root@mani-VirtualBox:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

root@mani-VirtualBox:~# docker run -it --mount source=myvol,destination=/tmp -d ubuntu

96653951cb70902c6c1d99ad24962d49e96230956cfb591fab3347e3423ff6ad

root@mani-VirtualBox:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

96653951cb70 ubuntu "bash" 10 minutes ago Up 10 minutes stupefied\_tharp

root@mani-VirtualBox:~# docker exec -it 96653951cb70 /bin/bash

root@96653951cb70:/# ls

bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var

root@96653951cb70:/# cd /tmp

root@96653951cb70:/tmp# ls

root@96653951cb70:/tmp# touch aa bb cc dd ee ff

root@96653951cb70:/tmp# exit

exit

root@mani-VirtualBox:~# cd /var/lib/docker/

root@mani-VirtualBox:/var/lib/docker# ls -l

total 44

drwx--x--x 4 root root 4096 Jul 24 11:02 buildkit

drwx--x--- 4 root root 4096 Jul 24 12:10 containers

drwx------ 3 root root 4096 Jul 24 11:02 image

drwxr-x--- 3 root root 4096 Jul 24 11:02 network

drwx--x--- 8 root root 4096 Jul 24 12:10 overlay2

drwx------ 4 root root 4096 Jul 24 11:02 plugins

drwx------ 2 root root 4096 Jul 24 11:02 runtimes

drwx------ 2 root root 4096 Jul 24 11:02 swarm

drwx------ 2 root root 4096 Jul 24 11:17 tmp

drwx------ 2 root root 4096 Jul 24 11:02 trust

drwx-----x 3 root root 4096 Jul 24 12:08 volumes

root@mani-VirtualBox:/var/lib/docker# cd volumes/

root@mani-VirtualBox:/var/lib/docker/volumes# ls -l

total 28

brw------- 1 root root 8, 3 Jul 24 11:02 backingFsBlockDev

-rw------- 1 root root 32768 Jul 24 12:08 metadata.db

drwx-----x 3 root root 4096 Jul 24 12:08 myvol

root@mani-VirtualBox:/var/lib/docker/volumes# cd myvol/

root@mani-VirtualBox:/var/lib/docker/volumes/myvol# ls -l

total 4

drwxrwxrwt 2 root root 4096 Jul 24 12:21 \_data

root@mani-VirtualBox:/var/lib/docker/volumes/myvol# cd \_data

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data# ls

aa bb cc dd ee ff

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data#

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data# docker stop 96653951cb70, docker rm 96653951cb70

96653951cb70

Error response from daemon: No such container: 96653951cb70,

Error response from daemon: No such container: docker

Error response from daemon: No such container: rm

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data# ls

aa bb cc dd ee ff

See after deleting this container also we are still seeing my data and network volume.

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data# docker run -it --mount source=myvol,destination=/tmp -d ubuntu

23115b47689fe07da45417631971addfc52c1a19cbe01a5a1a9d8bc4e327b688

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data# docker run -it --mount source=myvol,destination=/tmp -d ubuntu

ef408d78fa9474448fd0f6606279b5c7fcf4400ccc348f1842b8a41d6bab0c9f

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data# docker run -it --mount source=myvol,destination=/tmp -d ubuntu

9387489a123545e3728d5c580fc965c61edc6119fd17e8526affb7c7c53965bf

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data# docker run -it --mount source=myvol,destination=/tmp -d ubuntu

5751e6da795c17b8d060e6b7c3834e037e5f31b6b1da5bc7eff28da976fa7079

Network storage We can create multiple volumes. And we can mount these values for multiple values.

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data# docker info

Client:

Context: default

Debug Mode: false

Plugins:

app: Docker App (Docker Inc., v0.9.1-beta3)

buildx: Docker Buildx (Docker Inc., v0.8.2-docker)

compose: Docker Compose (Docker Inc., v2.6.0)

scan: Docker Scan (Docker Inc., v0.17.0)

Server:

Containers: 6

Running: 4

Paused: 0

Stopped: 2

Images: 1

Server Version: 20.10.17

Storage Driver: overlay2

Backing Filesystem: extfs

Supports d\_type: true

Native Overlay Diff: true

userxattr: false

Logging Driver: json-file

Cgroup Driver: systemd

Cgroup Version: 2

Plugins:

Volume: local

Network: bridge host ipvlan macvlan null overlay

Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog

Swarm: inactive

Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc

Default Runtime: runc

Init Binary: docker-init

containerd version: 10c12954828e7c7c9b6e0ea9b0c02b01407d3ae1

runc version: v1.1.2-0-ga916309

init version: de40ad0

Security Options:

apparmor

seccomp

Profile: default

cgroupns

Kernel Version: 5.15.0-41-generic

Operating System: Ubuntu 22.04 LTS

OSType: linux

Architecture: x86\_64

CPUs: 1

Total Memory: 5.305GiB

Name: mani-VirtualBox

ID: EFCR:4IWB:NIEZ:CWXU:PYF2:CQFR:H752:LN7Z:GBXD:KJDB:XWNM:K3HH

Docker Root Dir: /var/lib/docker

Debug Mode: false

Registry: https://index.docker.io/v1/

Labels:

Experimental: false

Insecure Registries:

127.0.0.0/8

Live Restore Enabled: false

Linking containers:

In the world of containers we never use host name and containers.

root@mani-VirtualBox:/var/lib/docker/volumes/myvol/\_data# cat /etc/hosts

127.0.0.1 localhost

127.0.1.1 mani-VirtualBox

# The following lines are desirable for IPv6 capable hosts

::1 ip6-localhost ip6-loopback

fe00::0 ip6-localnet

ff00::0 ip6-mcastprefix

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

linking containers:naming to the container identification to a container

root@mani-VirtualBox:~# docker run -it --name container -d ubuntu

acfd012d167135e9c44854796567af312935b97c41cad6a6b4f6b27627c94561

root@mani-VirtualBox:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

acfd012d1671 ubuntu "bash" 5 seconds ago Up 4 seconds container

5751e6da795c ubuntu "bash" 49 minutes ago Up 49 minutes cool\_heisenberg

9387489a1235 ubuntu "bash" 49 minutes ago Up 49 minutes infallible\_kare

ef408d78fa94 ubuntu "bash" 49 minutes ago Up 49 minutes mystifying\_neumann

23115b47689f ubuntu "bash" 49 minutes ago Up 49 minutes nervous\_hertz

Note: All Dockers are immutable. we can’t change the config files.

root@mani-VirtualBox:~# docker run -it --name container2 --link container -d ubuntu /bin/bash

13abf11f78d141c085389faea366b58b74f3e6d58abbc8cafc7f00ed86d0b1de

root@mani-VirtualBox:~# docker run -it --name container2 --link container -d ubuntu /bin/bash

13abf11f78d141c085389faea366b58b74f3e6d58abbc8cafc7f00ed86d0b1de

root@mani-VirtualBox:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

13abf11f78d1 ubuntu "/bin/bash" 36 seconds ago Up 34 seconds container2

acfd012d1671 ubuntu "bash" 4 hours ago Up 4 hours container

5751e6da795c ubuntu "bash" 5 hours ago Up 5 hours cool\_heisenberg

9387489a1235 ubuntu "bash" 5 hours ago Up 5 hours infallible\_kare

ef408d78fa94 ubuntu "bash" 5 hours ago Up 5 hours mystifying\_neumann

23115b47689f ubuntu "bash" 5 hours ago Up 5 hours nervous\_hertz

root@mani-VirtualBox:~# docker exec -it container2 /bin/bash

root@13abf11f78d1:/# apt-get update

root@13abf11f78d1:/# apt install iputils-ping

root@13abf11f78d1:/# ping container

PING container (172.17.0.6) 56(84) bytes of data.

64 bytes from container (172.17.0.6): icmp\_seq=1 ttl=64 time=0.147 ms

64 bytes from container (172.17.0.6): icmp\_seq=2 ttl=64 time=0.048 ms

64 bytes from container (172.17.0.6): icmp\_seq=3 ttl=64 time=0.048 ms

64 bytes from container (172.17.0.6): icmp\_seq=4 ttl=64 time=0.049 ms

64 bytes from container (172.17.0.6): icmp\_seq=5 ttl=64 time=0.048 ms

64 bytes from container (172.17.0.6): icmp\_seq=6 ttl=64 time=0.049 ms

^C

--- container ping statistics ---

6 packets transmitted, 6 received, 0% packet loss, time 5125ms

rtt min/avg/max/mdev = 0.048/0.064/0.147/0.036 ms

root@13abf11f78d1:/# cat /etc/hosts

127.0.0.1 localhost

::1 localhost ip6-localhost ip6-loopback

fe00::0 ip6-localnet

ff00::0 ip6-mcastprefix

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

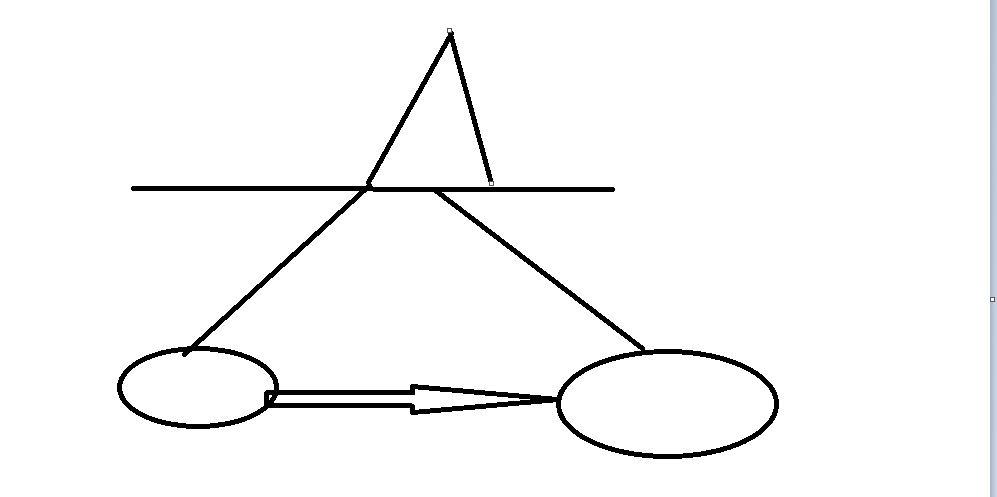
172.17.0.6 Container acfd012d1671

172.17.0.7 13abf11f78d1

This is adding entry of the container in container1 /etc/hosts file.

Above is the one way communication not two way communication.

We can communicate the Dockers using host machines we need to expose the application to hostmechine by this this two containers will be communicating with each other in a two way communication.



While creating the container only we can do it.once done we cant link them.

root@mani-VirtualBox:~# cat docker-compose.yml

VERSION:'3'

SERVICES:

apache:

image: httpd

ports:

- 80:80

ngnix:

image: nginx

ports:

- 8080:80

jenkins:

image: jenkins:2.60.3

ports:

- 8081:8080

- 50000:50000

root@mani-VirtualBox:~# sudo apt-get update

root@mani-VirtualBox:~# sudo apt-get install docker-compose-plugin

root@mani-VirtualBox:~# apt-cache madison docker-compose-plugin

docker-compose-plugin | 2.6.0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages

docker-compose-plugin | 2.5.0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages

docker-compose-plugin | 2.3.3~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages

sudo apt-get install docker-compose-plugin=<VERSION\_STRING>

root@mani-VirtualBox:~# sudo apt-get install docker-compose-plugin=2.6.0~ubuntu-jammy

DOCKER\_CONFIG=${DOCKER\_CONFIG:-$HOME/.docker}

$ mkdir -p $DOCKER\_CONFIG/cli-plugins

$ curl -SL https://github.com/docker/compose/releases/download/v2.7.0/docker-compose-linux-x86\_64 -o $DOCKER\_CONFIG/cli-plugins/docker-compose

Docker compose installation:

root@mani-VirtualBox:~# apt install docker-compose

root@mani-VirtualBox:~# cat docker-compose.yml

services:

apache:

image: httpd

ports:

- "80:80"

ngnix:

image: nginx

ports:

- "8080:80"

jenkins:

image: jenkins:2.60.3

ports:

- "8081:8080"

- "50000:50000"

Docker-compose up –d-----🡪 to run this in a demon mode.

root@mani-VirtualBox:~# docker-compose up -d

Starting root\_jenkins\_1 ... done

Starting root\_ngnix\_1 ... done

Starting root\_apache\_1 ... done

Docker-compose up-----🡪 to run this in a demon mode.

root@mani-VirtualBox:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

9c11854e3bc4 nginx "/docker-entrypoint.…" 4 minutes ago Up 2 minutes 0.0.0.0:8080->80/tcp, :::8080->80/tcp root\_ngnix\_1

2c7e32e1cb5a httpd "httpd-foreground" 4 minutes ago Up 2 minutes 0.0.0.0:80->80/tcp, :::80->80/tcp root\_apache\_1

e24a36230089 jenkins:2.60.3 "/bin/tini -- /usr/l…" 4 minutes ago Up 2 minutes 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp, 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp root\_jenkins\_1

root@mani-VirtualBox:~# docker-compose down

Stopping root\_ngnix\_1 ... done

Stopping root\_apache\_1 ... done

Stopping root\_jenkins\_1 ... done

Removing root\_ngnix\_1 ... done

Removing root\_apache\_1 ... done

Removing root\_jenkins\_1 ... done

Removing network root\_default

Linking containers:

root@mani-VirtualBox:~# cat docker-compose.yml

services:

apache:

image: httpd

ports:

- "80:80"

hostname: apache

ngnix:

image: nginx

ports:

- "8080:80"

jenkins:

image: jenkins:2.60.3

ports:

- "8081:8080"

- "50000:50000"

root@mani-VirtualBox:~# docker-compose down

Stopping root\_jenkins\_1 ... done

Stopping root\_ngnix\_1 ... done

Stopping root\_apache\_1 ... done

Removing root\_jenkins\_1 ... done

Removing root\_ngnix\_1 ... done

Removing root\_apache\_1 ... done

Removing network root\_default

root@mani-VirtualBox:~# docker-compose up -d

Creating network "root\_default" with the default driver

Creating root\_apache\_1 ... done

Creating root\_ngnix\_1 ... done

Creating root\_jenkins\_1 ... done

root@mani-VirtualBox:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

12869202a771 jenkins:2.60.3 "/bin/tini -- /usr/l…" 46 seconds ago Up 45 seconds 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp, 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp root\_jenkins\_1

bdb333fb3c0c nginx "/docker-entrypoint.…" 47 seconds ago Up 45 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp root\_ngnix\_1

9c1713532a52 httpd "httpd-foreground" 47 seconds ago Up 45 seconds 0.0.0.0:80->80/tcp, :::80->80/tcp root\_apache\_1

root@mani-VirtualBox:~# docker exec -it 12869202a771 /bin/bash

jenkins@12869202a771:/$ ping root\_ngnix\_1

PING root\_ngnix\_1 (172.20.0.3) 56(84) bytes of data.

64 bytes from root\_ngnix\_1.root\_default (172.20.0.3): icmp\_seq=1 ttl=64 time=0.074 ms

64 bytes from root\_ngnix\_1.root\_default (172.20.0.3): icmp\_seq=2 ttl=64 time=0.063 ms

64 bytes from root\_ngnix\_1.root\_default (172.20.0.3): icmp\_seq=3 ttl=64 time=0.064 ms

^C

--- root\_ngnix\_1 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2056ms

rtt min/avg/max/mdev = 0.063/0.067/0.074/0.005 ms

jenkins@12869202a771:/$ cat /etc/hosts

127.0.0.1 localhost

::1 localhost ip6-localhost ip6-loopback

fe00::0 ip6-localnet

ff00::0 ip6-mcastprefix

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

172.20.0.4 12869202a771

By default they can ping each other.

Note:in docker compose version 3 no need to ping in specific they will ping by default.with other containers.