Docker Commands

Contents

[Container & Image Management 4](#_Toc89431687)

[To create a container in interactive mode: 4](#_Toc89431688)

[To go inside the container: 4](#_Toc89431689)

[Docker History will give all the changes in images: 4](#_Toc89431690)

[To create new image with comment with latest changes in container: 4](#_Toc89431691)

[To take backup of Image: 4](#_Toc89431692)

[To take backup on particular path: 4](#_Toc89431693)

[To See All the Running containers: 5](#_Toc89431694)

[To See All the Running and stopped containers: 5](#_Toc89431695)

[To remove container: 5](#_Toc89431696)

[To see all the running and stopped container ID only: 5](#_Toc89431697)

[Remove all container forcefully: 5](#_Toc89431698)

[to see all images  : 5](#_Toc89431699)

[to see the Image ID only belong to all images : 5](#_Toc89431700)

[to remove any images  : 6](#_Toc89431701)

[To remove all images: 6](#_Toc89431702)

[To restore the backup images after removing all: 6](#_Toc89431703)

[Port Exposer: 6](#_Toc89431704)

[Docker files and image creation implementation 7](#_Toc89431705)

[Build Image using Docker file with automate changes: 8](#_Toc89431706)

[Tags : 8](#_Toc89431707)

[Port Exposed in container 9](#_Toc89431708)

[Update in docker file with CP command: 10](#_Toc89431709)

[**Volume Management or Volume Mapping** 10](#_Toc89431710)

[App1 : 10](#_Toc89431711)

[App2: 11](#_Toc89431712)

[App3: 11](#_Toc89431713)

[Folder Sharing 12](#_Toc89431714)

[Memory Management 14](#_Toc89431715)

[NETWORK ADMINISTRATION 15](#_Toc89431716)

[For Creating Network : 15](#_Toc89431717)

[Create Container using new network : 16](#_Toc89431718)

[Connect & Disconnect the Network in Existing Container : 16](#_Toc89431719)

[How to add specific IP to container from any of the network: 17](#_Toc89431720)

[Repository or Local/Private Registry Creation 19](#_Toc89431721)

[Steps for Insecure Registry 19](#_Toc89431722)

[Steps for Secure Registry 20](#_Toc89431723)

[Steps for AuthBased Registry 22](#_Toc89431724)

## Container & Image Management

### To create a container in interactive mode:

docker run –d –it –name <New containername> < imame ID>

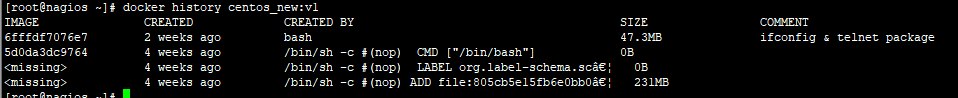
cid:image005.png@01D7DE73.EACF52E0

### To go inside the container:

docker exec -it <container ID> bash

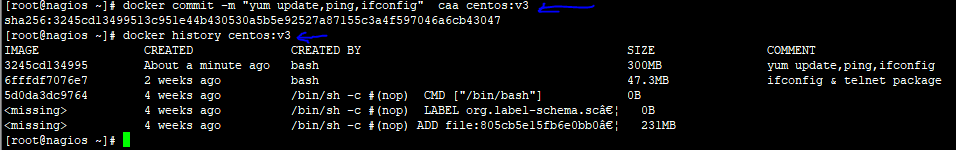
cid:image007.png@01D7DE74.6FDB3AD0

### Docker History will give all the changes in images:



### To create new image with comment with latest changes in container:

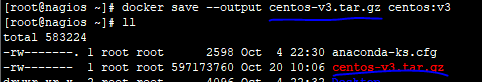
docker commit –m “message will give info” <NewImagename:tag> <Container ID>



### To take backup of Image:

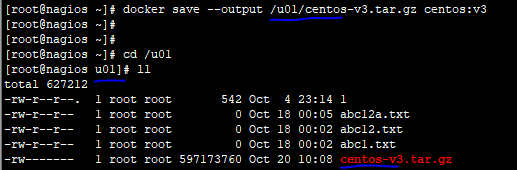
docker save –output imagename-tag.tar.gz <imagename:tag>

===================



### To take backup on particular path:

docker save –output  /path/imagename-tag.tar.gz <imagename:tag>



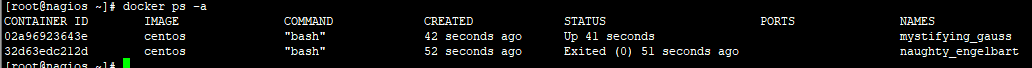
### To See All the Running containers:

docker ps

cid:image001.png@01D7DE73.3CEBCA50

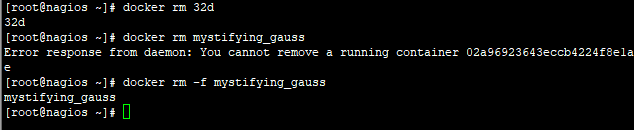
### To See All the Running and stopped containers:

docker ps –a



### To remove container:

docker rm <Container ID>



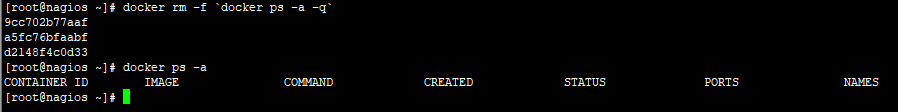
### To see all the running and stopped container ID only:

docker ps –a –q



### Remove all container forcefully:

docker rm `docker ps –a –q`

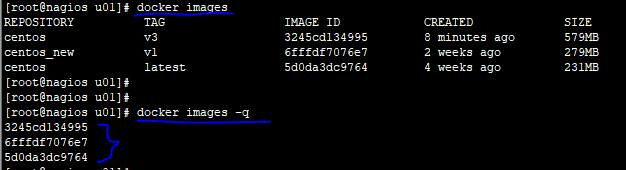


### to see all images  :

docker image

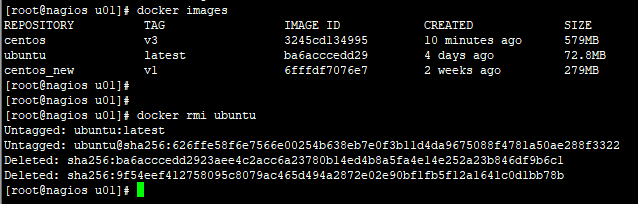
### to see the Image ID only belong to all images :

docker image –q



### to remove any images  :

docker rmi <image ID>

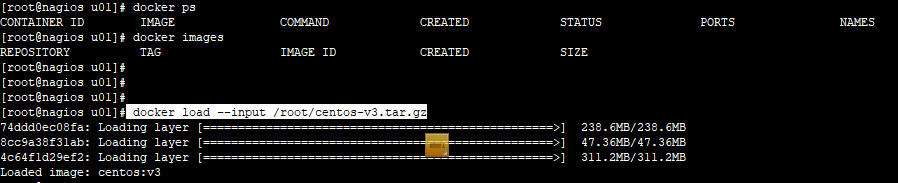


### To remove all images:

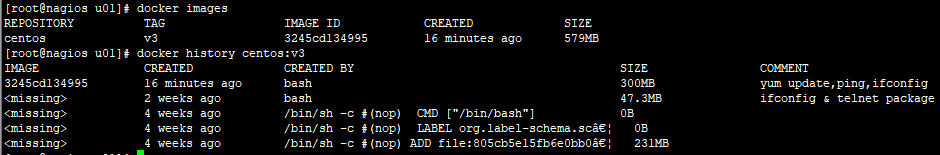
docker rmi `docker images –q`

### To restore the backup images after removing all:

docker load --input </path/backupofimage>



History of New Restored Images will be same:

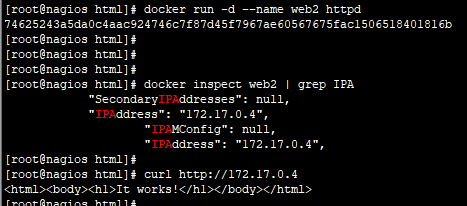


### Port Exposer:

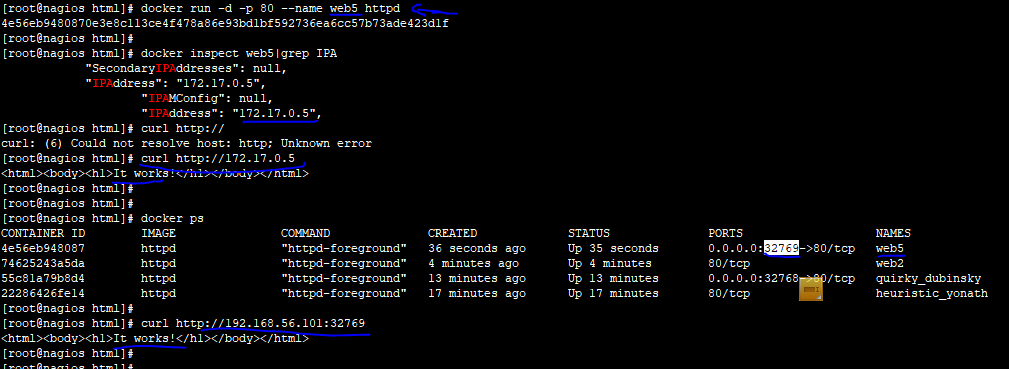
1. docker run httpd

2. docker run -it --name web1 httpd bash

3. docker run -d httpd : It open website only on container IP address , Not host system IP address



Through Port Expose Host machine will use the container website using random port  : docker run -d -p 80 httpd

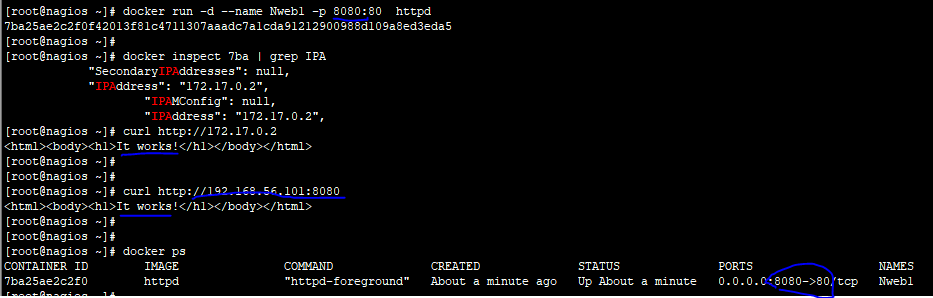


Host will use the container web using Specific port : docker run -d  --name web5 -p 8080:80 httpd

-p 8080:80 means

-p hostport:container port

It open website on container port 80 but assign specific port to host system Ip address



## Docker files and image creation implementation

=================================

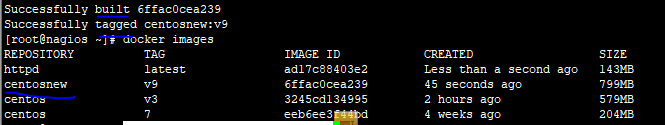
1. Docker file should have name “dockerfile”
2. For default path no need to do anything
3. For specific path use  “–f /path” to add configuration
4. All steps run at the time of image creation.



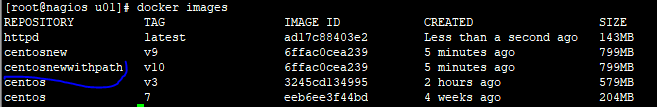
### Build Image using Docker file with automate changes:

docker build  -t <image:version> <path of docker file>

docker build -t centosnew:v9 .



docker build -t centosnewwithpath:v10 -f dockerfilewithpath .



### Tags :

CMD: it will execute at the time of Container creationm

     if you define multiple CMD than it will execute last Command only.

                if you define command during container creation than last CMD will be overwrite

ENTRYPOINT : it will execute at the time of Container creationm

     if you define multiple CMD than it will execute last Command only.

                but you define command during container creation than last CMD will not be overwrite and last line will be executed

USER: it will used to login container through particular user

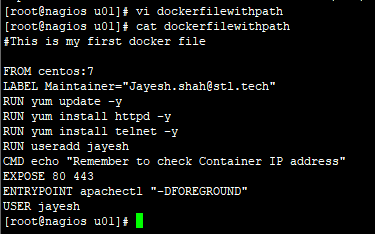
LABEL : to maintain the change name

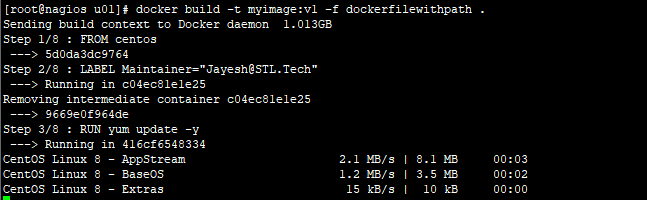
FROM : It is used to get image from docker hub.

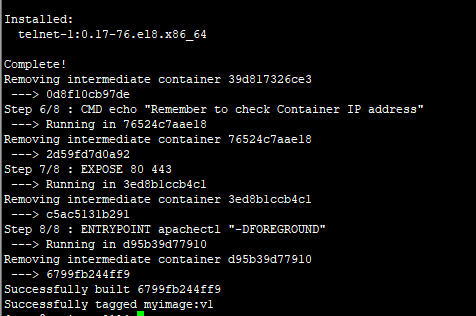
## Port Exposed in container

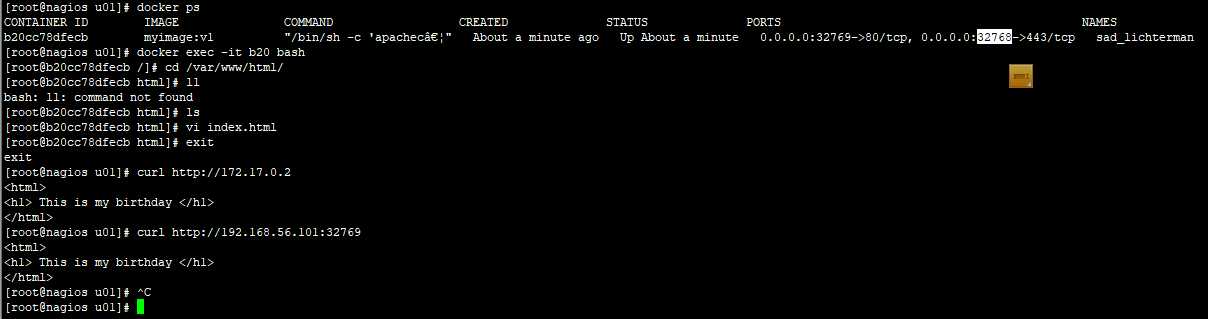
====================

1. Expose port is used to identify the port used in container



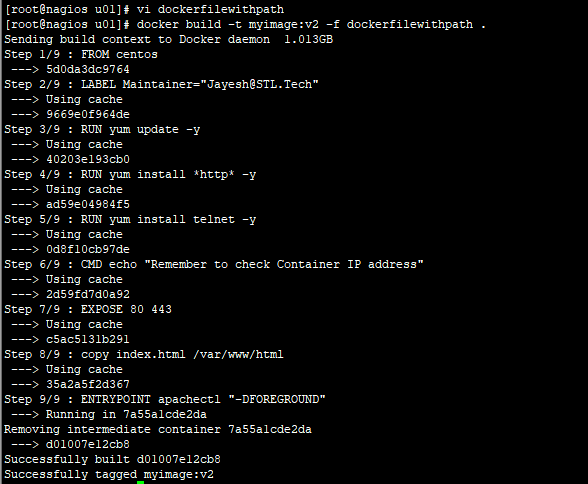






Update in docker file with CP command:

copy index.html /var/www/html



## **Volume Management or Volume Mapping**

=================================

Approach 1 : only “–v <container path>”

Approach 2:  map  “<host path > -v <container path>”

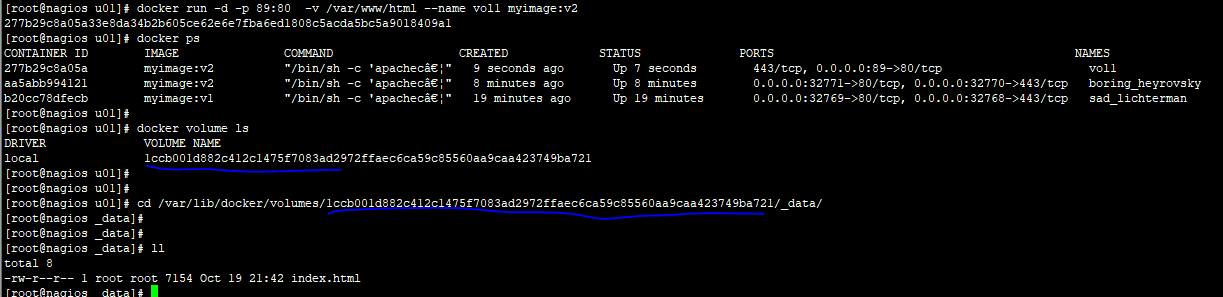
Approach 3: map Volume ID  “VolumeName –v <container path>”

### App1 :

docker run -d -p 89:80  -v /var/www/html --name vol1 myimage:v2

This will map container path “/var/www/html” to default path with random volume ID, So all the data present in container path will be stored at below path

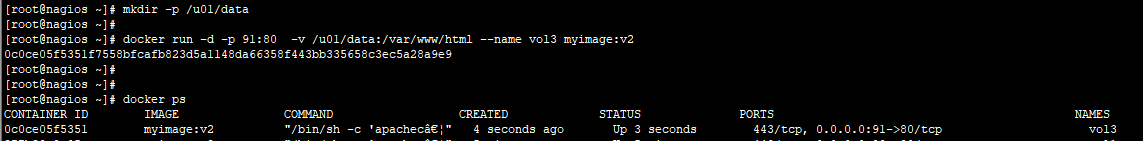
               /var/lib/docker/volumes/**<Random Volume ID>**/\_data/



### App2:

 mkdir -p /u01/data; docker run -d -p 91:80  -v /u01/data:/var/www/html --name vol3 myimage:v2

1. Mapped the host path “/u01/data” to “/var/www/html” So all the data belong to container at container path will be saved at given host path but no volume ID generated for the same.
2. We can map same path with multiple container to shared path



### App3:

docker run -d -p 92:80  -v web\_vol3:/var/www/html --name vol4 myimage:v2

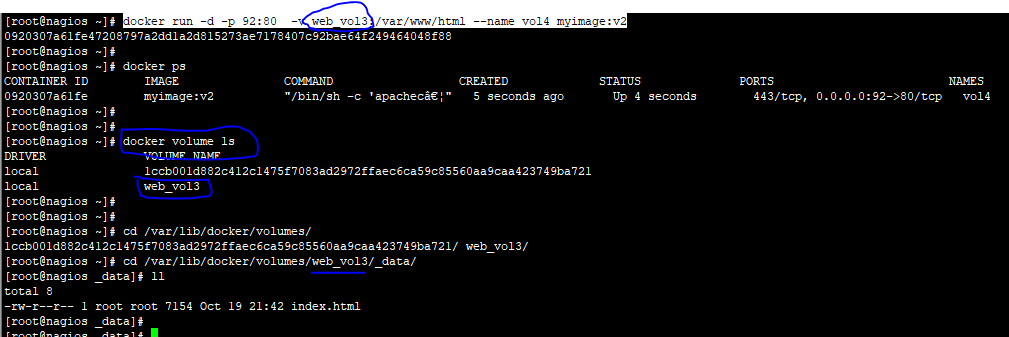
1. Here mapped the docker default volume with fixed Volume ID to Container path  So all the data belong to container will be stored in docker default with defined Volume ID generated.

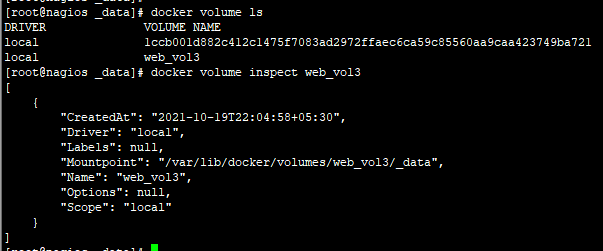
/var/lib/docker/volumes/<Volume ID>/\_data/

1. We can check with volume ID with below command :

docker volume ls

docker volume inspect <volumeID>





### Folder Sharing

=============

Task  : We have to share particular volume in host system “/u01/data” to 3 container and in sync to all of the container

docker run -d -p 92:80  -v /u01/data:/var/www/html --name vol3 myimage:v2

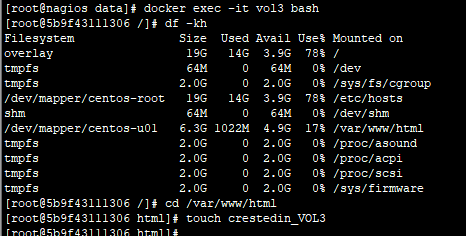
docker run -d -p 93:80  -v /u01/data:/var/www/html --name vol4 myimage:v2

docker run -d -p 94:80  -v /u01/data:/var/www/html --name vol5 myimage:v2

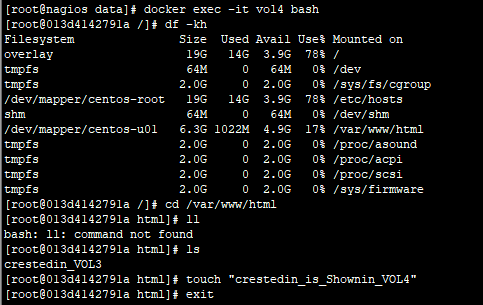
Three Container **vol3,vol4,vol5**  created using same volume **/u01/data** mapped :



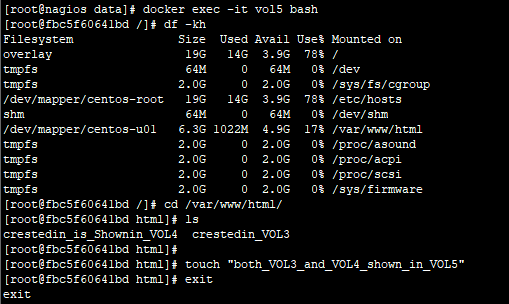
Checked Vol3 and created one file in shared partition



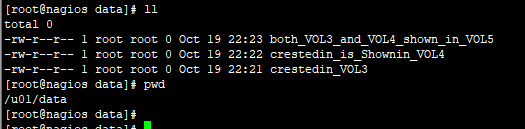
Checked vol4 and found files created under vol3 is visible and new file created in vol4



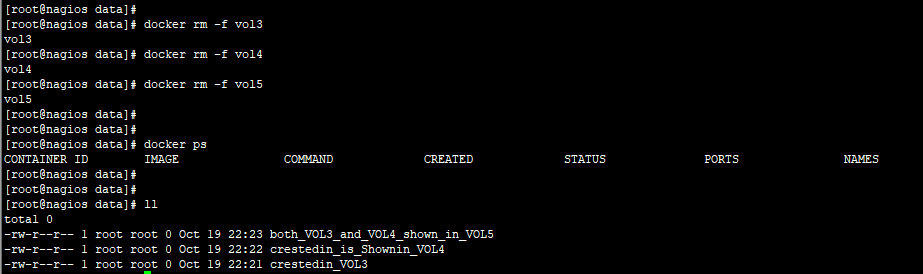
Checked in vol5 and found files created under vol3 and vol4  is visible and new file created in vol5



Check the host partition and found all the files present in this :



Removed all three container vol3 and vol4 and vol5 but data remain present at host shared partition Conclusion: after deleting the container we can save the configuration in it



Same thing we can map the Volume ID with multiple container for file sharing

docker run -d -p 92:80  -v web\_vol3:/var/www/html --name vol3 mypage:v2

docker run -d -p 93:80  -v web\_vol3:/var/www/html --name vol4 mypage:v2

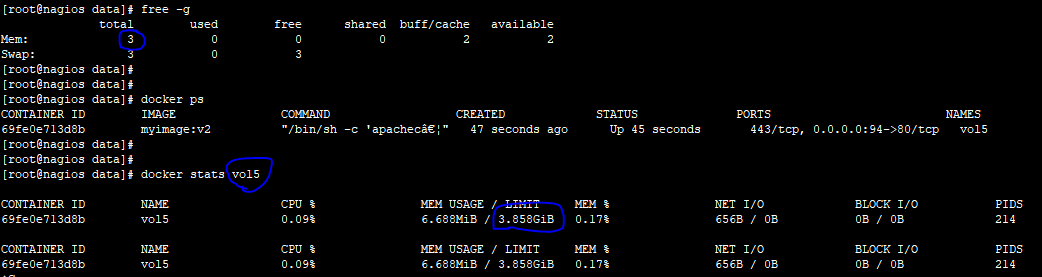
docker run -d -p 94:80  -v web\_vol3:/var/www/html --name vol5 mypage:v2

## Memory Management

==================

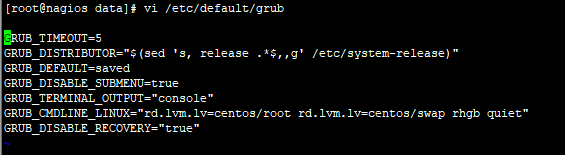
docker stats <container ID> :

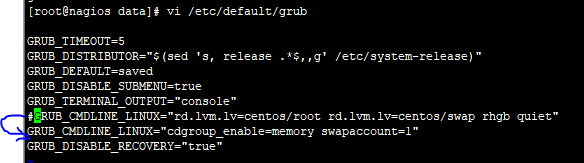
1. it is used to check the utilization in the container
2. by default all the memory of the host system is assigned to the container
3. docker to <container ID> give the processes running in container



1. For Setting the specific memory/CPU  to one container

vi /etc/default/grub : add GRUB\_CMDLINE\_LINUX="cdgroup\_enable=memory swapaccount=1"





1. Save the changes and exit the file
2. Update the grub file using below command

              Update-grub

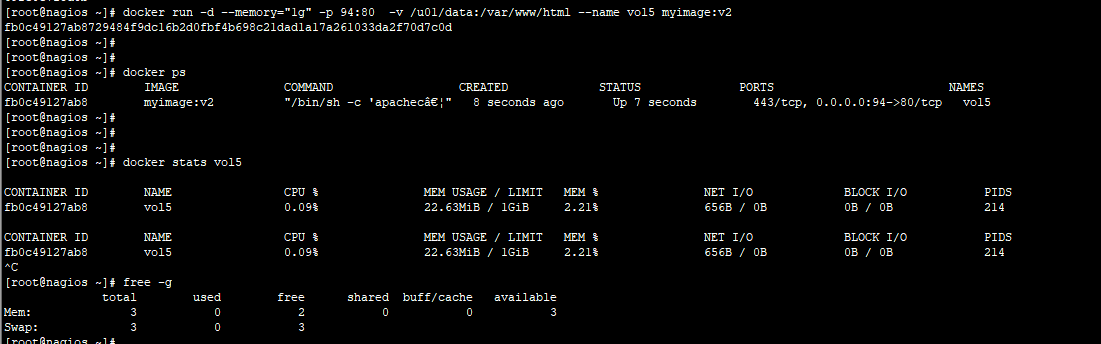
1. Finally, reboot your machine for the changes to take place

              sudo docker run -it --memory="[memory\_limit]" [docker\_image]

(The value of memory\_limit should be a positive integer followed by the suffix b, k, m, or g (short for bytes, kilobytes, megabytes, or gigabytes).

               docker run -d --memory="1g" -p 94:80  -v /u01/data:/var/www/html --name vol5 myimage:v2

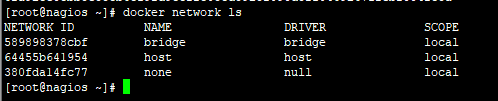
               docker run -d --memory="1g" –cpu =”.5” -p 95:81  -v /u01/data:/var/www/html --name vol6 myimage:v2



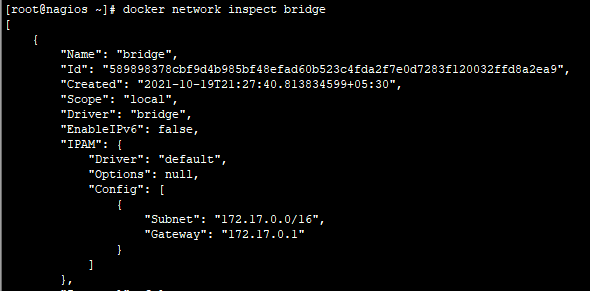
## NETWORK ADMINISTRATION

=======================

1. There is three type of Network Type : Bridge,host & none

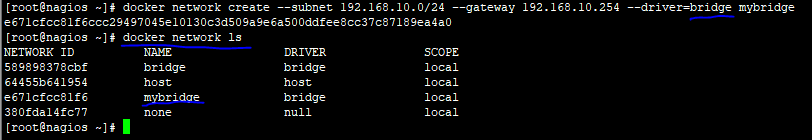


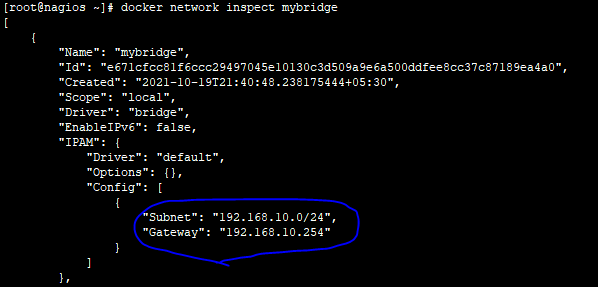
Bridge Default details :



For Creating Network :

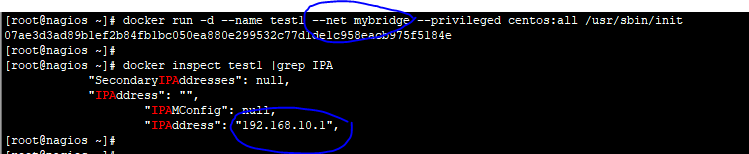
docker network create --subnet <Subnet Range> --gateway <Gateway IP> --driver=<Network Type bridge/host/null> <network Name>





### Create Container using new network :

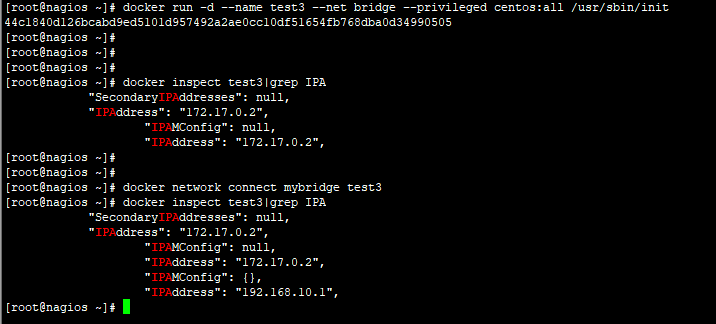
docker run –d –it –name <containername> --net <networkname> <image:tag>

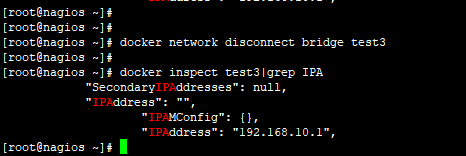


### Connect & Disconnect the Network in Existing Container :

                docker network connect <Network name> <container name/id>

                docker network disconnect <Network name> <container name/id>





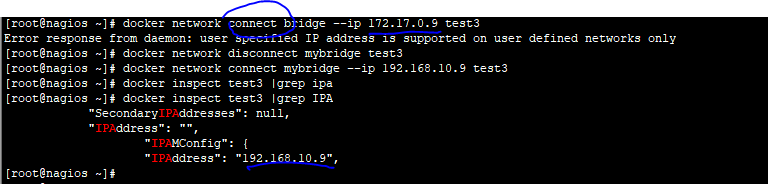
### How to add specific IP to container from any of the network:

note : Specific IP can be assigned in personalize network only not on public network

docker run -d --name <container name> --net <network name>  --ip <IP address>  <Image:tag>

we can also add the IP during container is already running

docker connect <networkname> --ip <IPaddress belong to network> <container ID/name>

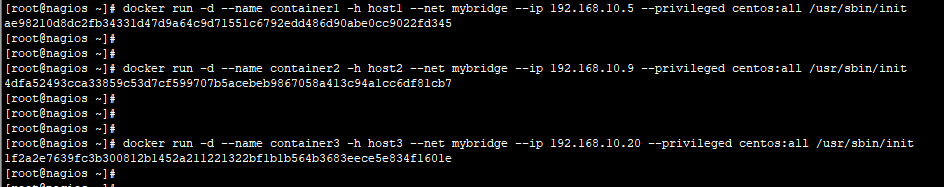


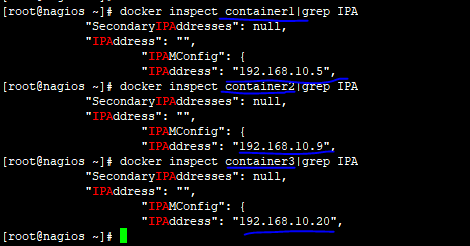
We can give the hostname while creating the container and if the container is running using personalize network then all container can communicate using container name itself

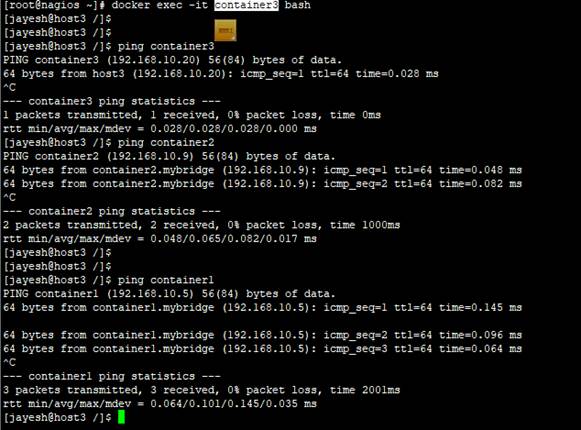
docker run -d --name container1 -h host1 --net mybridge --ip 192.168.10.5 --privileged centos:all /usr/sbin/init

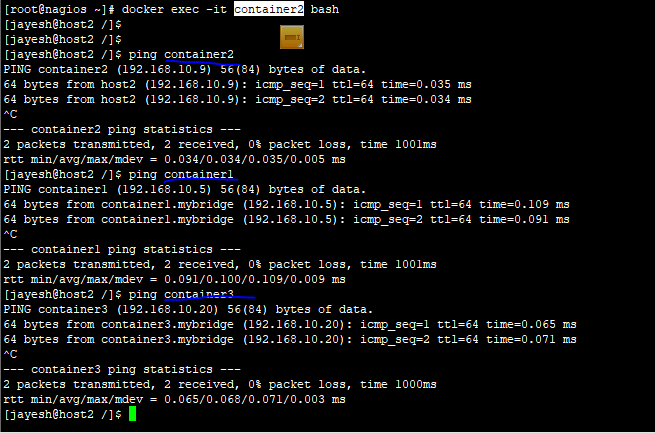
docker run -d --name container2 -h host2 --net mybridge --ip 192.168.10.9 --privileged centos:all /usr/sbin/init

docker run -d --name container3 -h host3 --net mybridge --ip 192.168.10.20 --privileged centos:all /usr/sbin/init







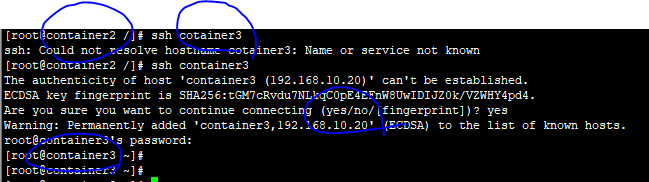


So we can give container name and hostname same :

docker run -d --name container1 -h container1 --net mybridge --ip 192.168.10.5 --privileged centos:all /usr/sbin/init

docker run -d --name container2 -h container2 --net mybridge --ip 192.168.10.9 --privileged centos:all /usr/sbin/init

docker run -d --name container3 -h container3 --net mybridge --ip 192.168.10.20 --privileged centos:all /usr/sbin/init



# Repository or Local/Private Registry Creation

## Steps for Insecure Registry

Step1 :

docker pull registry

[root@nagios v2]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

registry latest b8604a3fe854 Less than a second ago 26.2MB

Step2 :

host entry :

[root@nagios v2]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.56.101 dockerrepo.com dockerrepo

Step 3.

docker insecure entry in /etc/docker/daemon.json

[root@nagios v2]# cat /etc/docker/daemon.json

{

"insecure-registries" : ["dockerrepo.com:5000"]

}

Step 4.

mkdir /data/registry/

Step 5.

docker run -d -p 5000:5000 --restart=always -v /data/registry/:/var/lib/registry --name insecurereg registry

Step 6.

systemctl daemon-reload

systemctl restart docker

Step 7.

http://192.168.56.101:5000/v2/\_catalog in browser

Step 8.

docker tag registry:latest dockerrepo.com:5000/registry:latest

Step 9.

docker push dockerrepo.com:5000/registry:latest

## Steps for Secure Registry

Step 1 :

docker pull registry

[root@nagios v2]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

registry latest b8604a3fe854 Less than a second ago 26.2MB

Step 2:

host entry :

[root@nagios v2]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.56.101 dockerrepo.com dockerrepo

Step 3.

Generate certificate :

1. mkdir certs
2. openssl req -newkey rsa:4096 -nodes -sha256 -keyout certs/domain.key -x509 -days 365 -out certs/domain.crt

[root@nagios ~]# openssl req -newkey rsa:4096 -nodes -sha256 -keyout certs/domain.key -x509 -days 365 -out certs/domain.crt

1. Generating a 4096 bit RSA private key

.........++

...................................................................................................................................................................................................................................................................................................................................................++

writing new private key to 'certs/domain.key'

-----

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [XX]:

State or Province Name (full name) []:

Locality Name (eg, city) [Default City]:

Organization Name (eg, company) [Default Company Ltd]:

Organizational Unit Name (eg, section) []:

Common Name (eg, your name or your server's hostname) []:dockerrepo.com

Email Address []:

[root@nagios certs]# ll

total 8

-rw-r--r-- 1 root root 1980 Oct 22 05:12 domain.crt

-rw-r--r-- 1 root root 3272 Oct 22 05:12 domain.key

Step 4:

cd /etc/docker/

Step 5:

mkdir -p certs.d/dockerrepo.com:5000/

Step 6.

copy the domain.crt to /etc/docker/certs.d/dockerrepo.com:5000/

cp domain.crt /etc/docker/certs.d/dockerrepo.com

Step 7:

mkdir /data/registry/

Step 8:

docker run -d \

--name securereg \

-v /data/registry/:/var/lib/registry \

-v /root/certs:/certs \

-e REGISTRY\_HTTP\_TLS\_CERTIFICATE=/certs/domain.crt \

-e REGISTRY\_HTTP\_TLS\_KEY=/certs/domain.key \

-p 5000:5000 \

registry

[root@nagios certs]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

f29b87ac26a4 registry "/entrypoint.sh /etcâ€¦" 3 seconds ago Up 2 seconds 0.0.0.0:5000->5000/tcp securereg

Step 9:

systemctl daemon-reload

systemctl restart docker

Step 10:

http://192.168.56.101:5000/v2/\_catalog in browser

Step 11:

[root@nagios certs]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

dockerrepo.com:5000/httpd v2 ad17c88403e2 Less than a second ago 143MB

httpd 2 ad17c88403e2 Less than a second ago 143MB

nginx latest ea335eea17ab Less than a second ago 141MB

registry 2 b8604a3fe854 Less than a second ago 26.2MB

registry latest b8604a3fe854 Less than a second ago 26.2MB

dockerrepo.com:5000/registry latest b8604a3fe854 Less than a second ago 26.2MB

ubuntu latest ba6acccedd29 5 days ago 72.8MB

[root@nagios certs]# docker tag ubuntu:latest dockerrepo.com:5000/ubuntu:latest

[root@nagios certs]# docker push dockerrepo.com:5000/ubuntu:latest

[root@nagios certs]# docker push dockerrepo.com:5000/ubuntu:latest

The push refers to repository [dockerrepo.com:5000/ubuntu]

9f54eef41275: Pushed

latest: digest: sha256:7cc0576c7c0ec2384de5cbf245f41567e922aab1b075f3e8ad565f508032df17 size: 529

Step 11:

[root@nagios certs]# cd /data/registry/docker/registry/v2/repositories/

[root@nagios repositories]# ls

httpd myfirstimage registry ubuntu

## Steps for AuthBased Registry

Step 1.

docker pull registry

[root@nagios v2]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

registry latest b8604a3fe854 Less than a second ago 26.2MB

Step 2.

host entry :

[root@nagios v2]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.56.101 dockerrepo.com dockerrepo

Step 3.

generate user credetials :

docker container run --entrypoint htpasswd registry:2.7.0 -bnB jayesh J@yesh2711 > /root/auth/htpasswd

Step 4. cd /root/auth

Step 5. mkdir /data/registry/

Step 6.

docker run -d \

--name authreg \

-v /data/registry/:/var/lib/registry \

-v /root/auth:/auth \

-v /root/certs:/certs \

-e "REGISTRY\_AUTH=htpasswd" \

-e "REGISTRY\_AUTH\_HTPASSWD\_REALM=Registry Realm" \

-e REGISTRY\_AUTH\_HTPASSWD\_PATH=/auth/htpasswd \

-e REGISTRY\_HTTP\_TLS\_CERTIFICATE=/certs/domain.crt \

-e REGISTRY\_HTTP\_TLS\_KEY=/certs/domain.key \

-p 5000:5000 \

registry

[root@nagios certs]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

f29b87ac26a4 registry "/entrypoint.sh /etcâ€¦" 3 seconds ago Up 2 seconds 0.0.0.0:5000->5000/tcp securereg

Step 7. systemctl daemon-reload

systemctl restart docker

Step 8. http://192.168.56.101:5000/v2/\_catalog in browser

Step 9.

[root@nagios auth]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

httpd 2 ad17c88403e2 Less than a second ago 143MB

dockerrepo.com:5000/httpd v2 ad17c88403e2 Less than a second ago 143MB

nginx latest ea335eea17ab Less than a second ago 141MB

registry 2 b8604a3fe854 Less than a second ago 26.2MB

registry latest b8604a3fe854 Less than a second ago 26.2MB

dockerrepo.com:5000/registry latest b8604a3fe854 Less than a second ago 26.2MB

ubuntu latest ba6acccedd29 6 days ago 72.8MB

dockerrepo.com:5000/ubuntu latest ba6acccedd29 6 days ago 72.8MB

registry 2.7.0 33fbbf4a24e5 2 years ago 24.2MB

Step 10.

[root@nagios certs]# docker tag registry:2.7.0 dockerrepo.com:5000/registry:2.7.0

Step 11.

[root@nagios auth]# docker login dockerrepo.com:5000

Authenticating with existing credentials...

Login did not succeed, error: Error response from daemon: login attempt to https://dockerrepo.com:5000/v2/ failed with status: 401 Unauthorized

Username (jayesh): jayesh

Password:

WARNING! Your password will be stored unencrypted in /root/.docker/config.json.

Configure a credential helper to remove this warning. See

https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

Step 12.

[root@nagios auth]# docker push dockerrepo.com:5000/registry:2.7.0

The push refers to repository [dockerrepo.com:5000/registry]

e0ce598fab8f: Pushed

85384d42542a: Pushed

d2e29d15f6a6: Pushed

6b6e0aba7201: Pushed

7bff100f35cb: Pushed

2.7.0: digest: sha256:d6df97c93a0b8db7b355e08d54fbf38e46667eaa251cd5a90ddf0e53c35375b1 size: 1363

Step 13.

[root@nagios certs]# cd /data/registry/docker/registry/v2/repositories/

[root@nagios repositories]# ls

httpd myfirstimage registry ubuntu