DOCKER

Create Instance in AWS:

- Copy the link(.pem) file from the AWS and open it in .pem containing folder or file
- Open Git Bash here and paste the link in -----($)
  It will ask yes/no and type -----yes and Enter
- Then it will display like this -----ubuntu@ip-172-31-0-118:~$
- Then type command to change to root directory as----sudo –i and Enter
- Then it will display like this ------root@ip-172-31-0-118:~#

  Now we are in root directory or admin

  Type Command in root i.e apt-get update
- root@ip-172-31-0-118:~# apt-get update and Enter

Docker installation in two steps (using Docker Installation Script):

>>Type in Google Docker installation Script and click on first link and paste in git bash and then Enter

>>Next click on the second link and Enter

>>Docker will install automatically (It will display like -- sudo docker engine activate)

To create nginx in Docker :-

Type Command as: docker run --name mynginx -d -p 8080:80 nginx and Enter

After nginx will downloaded automatically and installed

To know the nginx container is created type command as ----- docker ps and Enter

To display all the containers type command as ----- docker ps –a and Enter

To display container id and name type command as---- docker images and Enter

(It will shows the images site and containers running)

To know the ngnix created or not follow the steps:

Assign port numbers in the particular instance

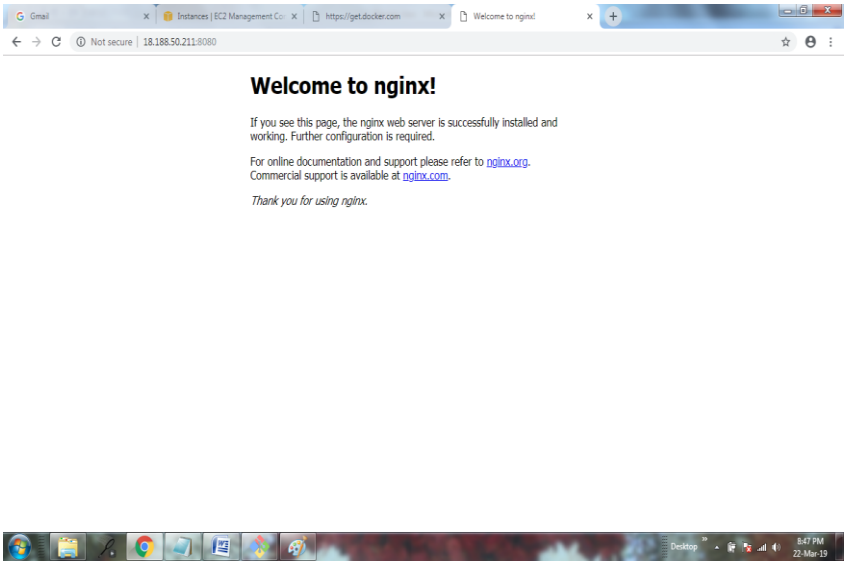
i. Select instance > go to security groups >click on that below link and

  >click on inbound > Edit >Assign port numbers ----Ex: 8080

ii. Copy the public ip address from description field > paste in web page

(I.e Ex: 18.188.50.211:8080)

Then it will display like this



To create Jenkins in Docker :-

Type Command as: docker run --name myjenkins -d -p 8080:8080 jenkins and Enter

After jenkins will downloaded automatically and installed

To know the Jenkins container is created type command as ----- docker ps and Enter

To display all the containers type command as ----- docker ps –a and Enter

To display container id and name type command as---- docker images and Enter

(It will shows the images site and containers running)

To know the Jenkins created or not follow the steps:

Assign port numbers in the particular instance

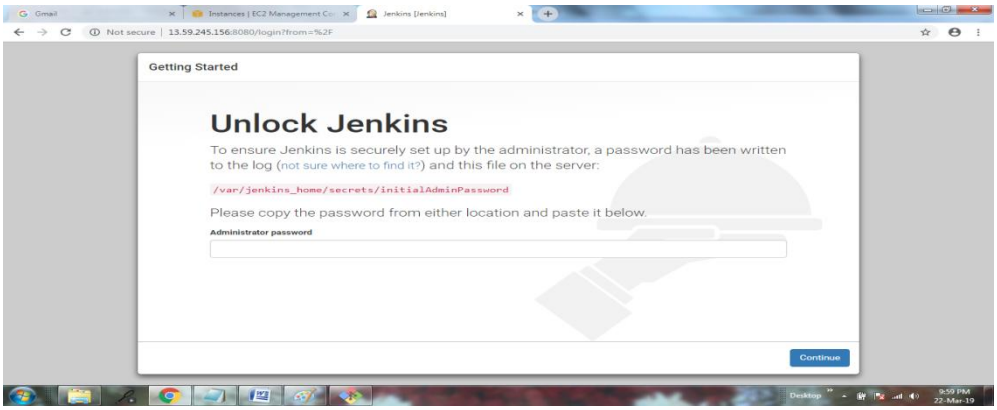
i. Select instance > go to security groups >click on that below link and

>click on inbound > Edit >Assign port numbers ----Ex: 8080

ii. Copy the public ip address from description field > paste in web page

(I.e Ex: 18.188.50.211:8080)

Then it will display like this



Next step:

We have to interact with Jenkins via this command:

>>docker exec -it myjenkins /bin/bash and Enter

Now the root changes to jenkins like this

>>jenkins@3038bcdb1b33

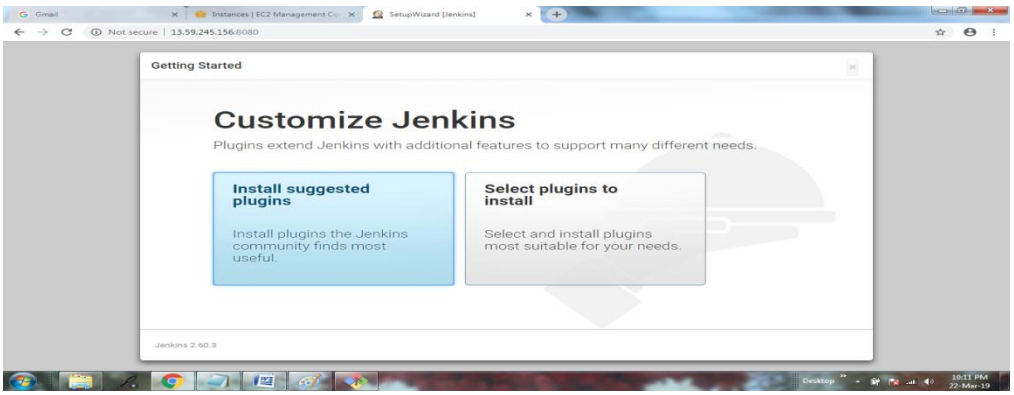
To Generated admin password we have follows this command

We need to copy the link from Jenkins page

>>jenkins@3038bcdb1b33:/$ cat (/var/jenkins_home/secrets/initialAdminPassword)

Password will be generated and copy the password and paste in jenkins page and continue or login

Now we entred to the Jenkins displayed below



<u>To come out from the jenkins to root directory:</u>

syntax exit <container name/id> Ex:- exit myjenkins

DOCKER LINK :-

>>SQL to WORDPRESS:-

Type command in git bash as follows:

>>docker run --name mydb -e MYSQL_ROOT_PASSWORD=sai -d mysql:5.7.25 Enter

Again type command in git bash as follows to link SQL to WORDPRESS:

>>docker run --name mywordpress --link mydb:mysql -d -p 8080:80 wordpress Enter

To see the link created or not follow the steps:

Assign port numbers in the particular instance

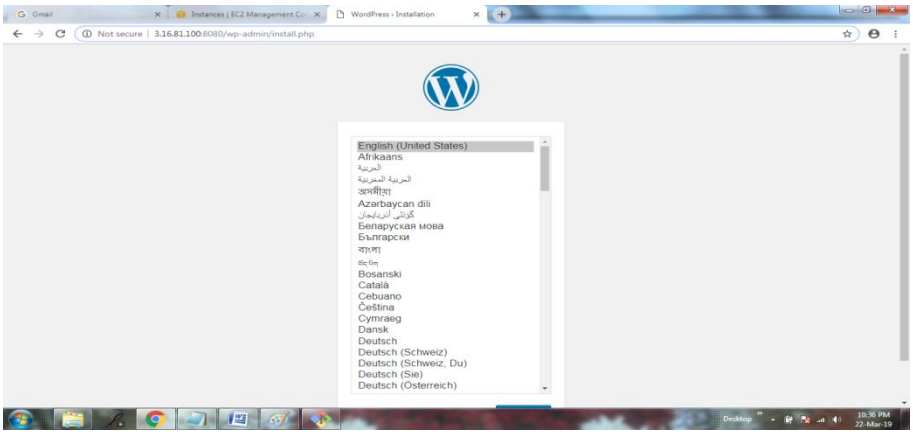
i. Select instance > go to security groups >click on that below link and

>click on inbound > Edit >Assign port numbers ----Ex: 8080

ii. Copy the public ip address from description field > paste in web page

(I.e Ex: http://3.16.81.100:8080)

Then it will display like this



Docker Compose:-

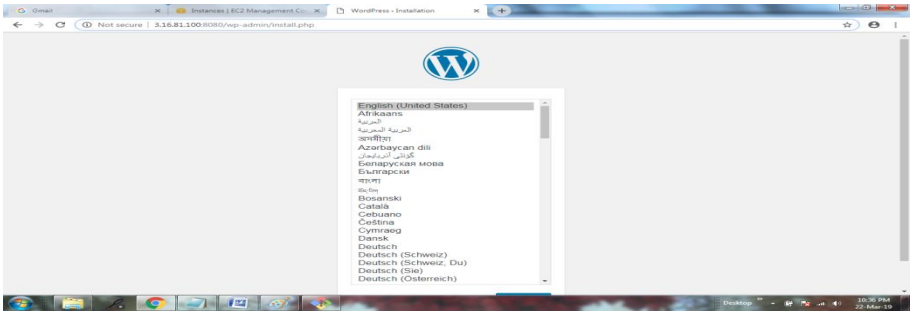
We have to install Docker Compose.
>>Type in Google as:---- github.com/devopsvc
>>Go to repositories
>>My docker
>>Docker compose install
Copy the links

```
version: '3'
services:
  db:
    image: mysql:5.7.22
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: test
  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - 8080:80
    restart: always
```

to run the above follow the command:
docker-compose up –d
to stop:
docker-compose down

To see the link created or not follow the steps:

Assign port numbers in the particular instance
i. Select instance > go to security groups >click on that below link and
  >click on inbound > Edit >Assign port numbers ----Ex: 8080
ii. Copy the public ip address from description field > paste in web page
(I.e Ex: http://3.16.81.100:8080)
Then it will display like this



Docker file:-

We can create our own image by docker file.

Type command as follows:

Syntax: mkdir <file name> Ex: mkdir mydocker

>>cd mydocker ---- it will move from root to directory

>>vi **D**ockerfile

>>> it will open window and type as follows:-

Press I button and it will go to insert mode

FROM ubuntu:16.04

MAINTAINER <Any name or Number>

RUN apt-get update

RUN apt-get install git –y

CMD [“echo”, ”This is sai”]

>>> after writing press escape and save **:wq!** and Enter

We have to build type as command

>>docker build –t <any name>:1 . and Enter

Ex: docker build –t <myimage>:1 . and Enter

We have to run the above

>>docker run myimage:1 and Enter

Then it will display the data which you entered.

We need to tag the above build to the docker hub account username

>>docker tag myimage:1 <username of the docker hub>/myimage:1

Ex: docker tag myimage:1 sai218/myimage:1 and Enter

Then type command as

>>docker login

Then it will ask username and password in git bash

Enter Username :

Password:

Then we need to push the above image to the docker hub account

Type command as

>>docker push sai218/myimage:1

Then login in to docker hub account and click on repositories.:

We can see image created in the name like ……ex: sai218/myimage:1

Q.I WANT TO CREATE A IMAGE FOR DEPLOYEE APPLICATION (JENKINS):

Tomcat→application server

Type in google as : https://hub.docker.com/

Search word **tomcat** in search bar and copy docker file version as : 7.0.93-jre8

Then follows the commands

root@ip-172-31-20-67:~# mkdir rahul

root@ip-172-31-20-67:~# cd rahul

root@ip-172-31-20-67:~/rahul# vi Dockerfile

>>> it will open window and type as follows:-

Press I button and it will go to insert mode

FROM tomcat:7.0.93-jre8

MAINTAINER 7794808362

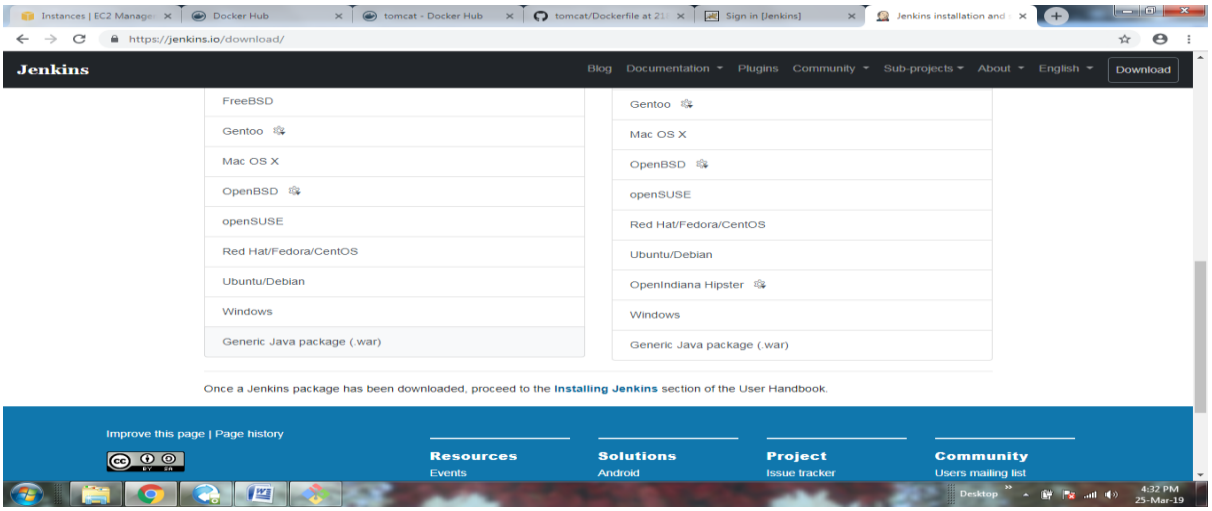
RUN apt-get update

COPY

for **COPY** save and quit from window

- root@ip-172-31-20-67:~# ls

- Then it will display Dockerfile
- Now we have to go to google page and type Jenkins.io and click on download and it will display like th



is

- Then copy the generic link address and paste in root.

- **root@ip-172-31-20-67:~ copy the link and Enter**

- then it will downloaded automatically then open

- root@ip-172-31-20-67:~vi Dockerfile

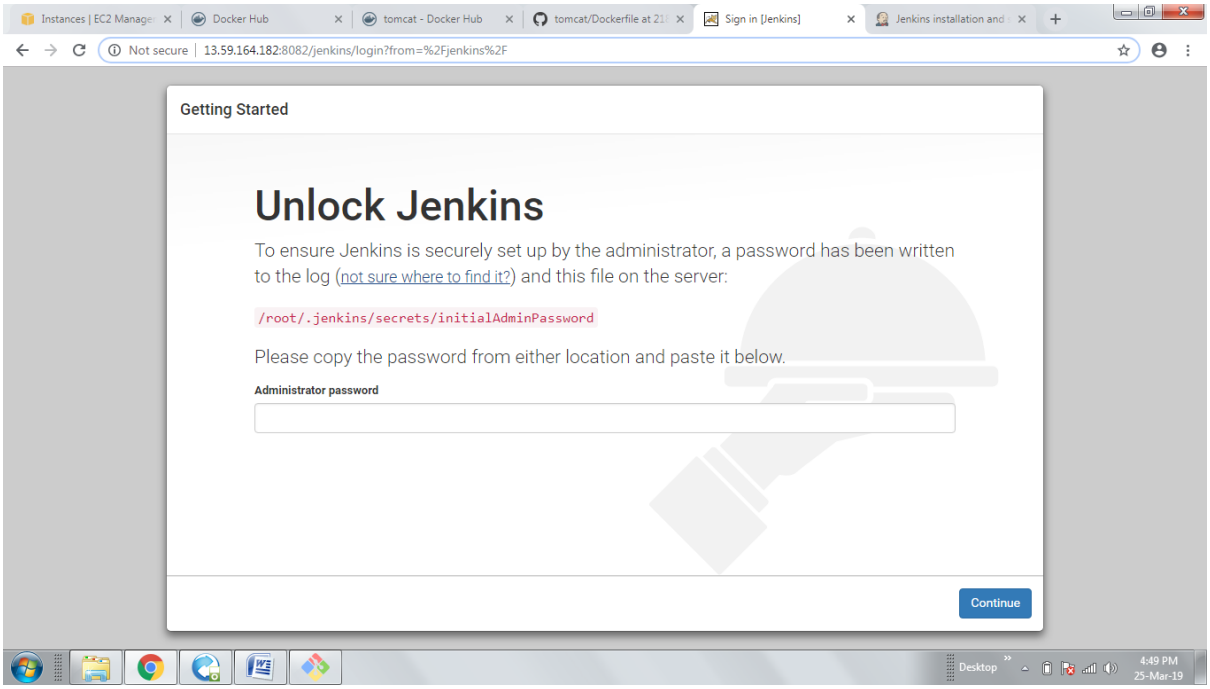
- Then changes like this

  COPY jenkins.war /usr/local/tomcat/webapps/
  EXPOSE 8080
  CMD ["catalina.sh", "run"]
  Save and quit from the window
  We have to build the image as follows:-

- root@ip-172-31-20-67:~docker build –t <name of the image already created>:1 . and Enter
- E x: root@ip-172-31-20-67:~docker build –t <uma:1 . and Enter
- check whether it I created or not
- docker images
- then it will shows whether it is created or not
- then run the image as follows
- root@ip-172-31-20-67:~docker run –name <container name> -d –p 8080:8080 uma:1 and Enter
- Ex: root@ip-172-31-20-67:~docker run --name <tm1> -d –p 8080:8080 uma:1 and Enter

To see the link created or not follow the steps:

Assign port numbers in the particular instance
i. Select instance > go to security groups >click on that below link and >click on inbound > Edit >Assign port numbers ----Ex: 8080
ii. Copy the public ip address from description field > paste in web page (I.e Ex: http://3.16.81.100:8080)
Then it will display like this
Tomcat page will displayed
For jenkins : type after 8080/jenkins



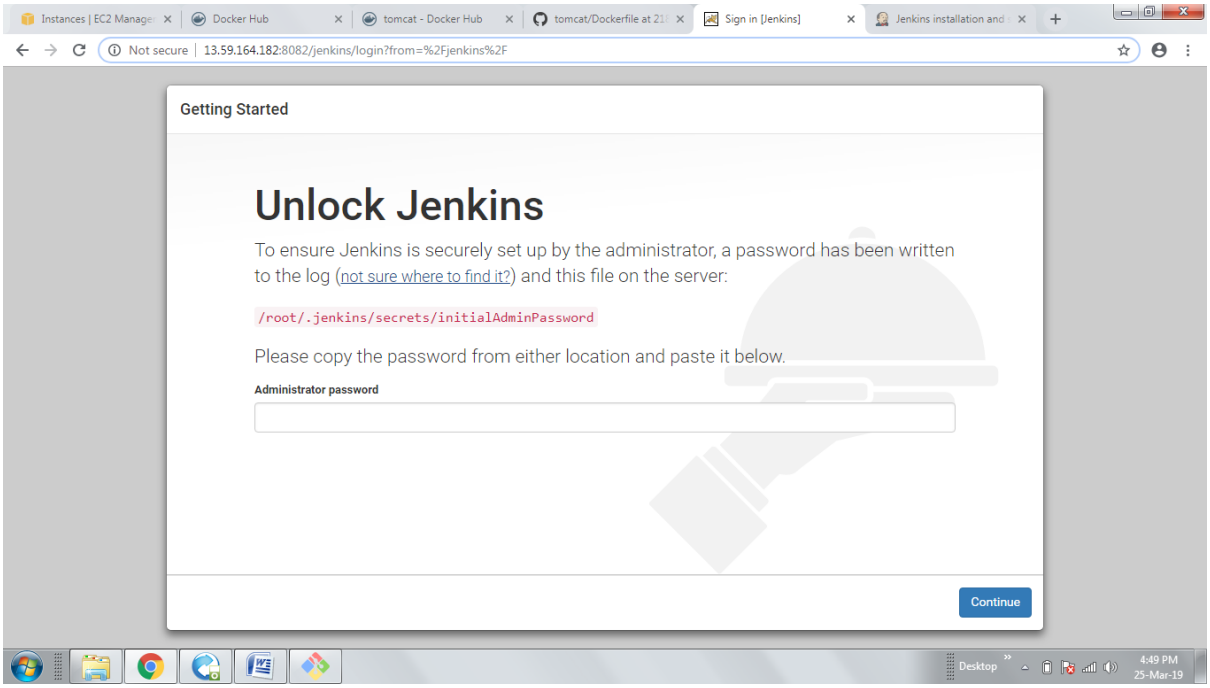
By using **ADD** command instead **of COPY** we can do run image tomcat and jenkins**:**

FROM tomcat:7.0.93-jre8
MAINTAINER 7794808362
RUN apt-get update
ADD <copy the link address from Jenkins.io web site war file **>**
Ex: ADD http://mirrors.jenkins.io/war-stable/latest/jenkins.war /usr/local/tomcat/webapps/
EXPOSE 8080
CMD ["catalina.sh", "run"]

Save and quit from the window

We have to build the image as follows:-

- **root@ip-172-31-20-67:~docker build –t <name of the image already created>:1 . and Enter**
- **Ex: root@ip-172-31-20-67:~docker build –t <uma:1 . and Enter**
- **check whether it is created or not**
- **docker images**
- **then it will shows whether it is created or not**
- **then run the image as follows**
- **root@ip-172-31-20-67:~docker run –name <container name> -d –p 8080:8080 uma:1 and Enter**
- **Ex: root@ip-172-31-20-67:~docker run --name <tm1> -d –p 8080:8080 uma:1 and Enter**
- **<u>To see the link created or not follow the steps:</u>**
-
- **Assign port numbers in the particular instance**
- **i. Select instance > go to security groups >click on that below link and**
- **>click on inbound > Edit >Assign port numbers ----Ex: 8080**
- **ii. Copy the public ip address from description field > paste in web page**
- **(I.e Ex: http://3.16.81.100:8080)**
- **Then it will display like this**
- **Tomcat page will displayed**
- **For jenkins : type after 8080/jenkins**
-


Instances | EC2 Manage
Docker Hub
tomcat - Docker Hub
tomcat/Dockerfile at 21
Sign in [Jenkins]
Jenkins installation and
Not secure | 13.59.164.182:8082/jenkins/login?from=%2Fjenkins%2F
Getting Started
Unlock Jenkins
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:
/root/.jenkins/secrets/initialAdminPassword
Please copy the password from either location and paste it below.
Administrator password
Continue
Desktop
4:49 PM
25-Mar-19

DOCKER VOLUMES:

To know the volumes structure type

>> root@ip-172-31-20-67:~docker volume

It will display like below

Usage: docker volume COMMAND

Manage volumes

Commands:
  create   Create a volume
  inspect   Display detailed information on one or more volumes
  ls   List volumes
  prune   Remove all unused local volumes
  rm   Remove one or more volumes

>>To create a volume by using 2 containers (Ex:c1, c2)follow the below commands:

i) For creation of 1st container i.e c2

>>root@ip-172-31-20-67:~# docker run --name c2 -v/myvolume -d -p 8081:8080 uma:1

It will display like below : it is created successfully
f5b20b23fd6667cbe83b09256704a22715f893e1ee37efb6ca4a7f782637dd80

to check whether it is created or not :

>>root@ip-172-31-20-67:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f5b20b23fd66 uma:1 "catalina.sh run" 15 seconds ago Up 13 seconds 0.0.0.0:8081**->8080/tcp c2**

Now we have to execute the created container follow the commands

root@ip-172-31-20-67:~# docker exec -it c2 /bin/bash

Now we entered to tomcat it will display like below:

root@f5b20b23fd66:/usr/local/tomcat#

To know whether **my volume** is created or not follow the below command

root@f5b20b23fd66:/usr/local/tomcat# df –h

it will show like below and we can check whether it is created or not:

Filesystem Size Used Avail Use% Mounted on

overlay 7.7G 2.9G 4.9G 38% /

tmpfs 64M 0 64M 0% /dev

tmpfs 496M 0 496M 0% /sys/fs/cgroup

/dev/xvda1 7.7G 2.9G 4.9G 38% **/myvolume**

shm 64M 0 64M 0% /dev/shm

tmpfs 496M 0 496M 0% /proc/acpi

tmpfs 496M 0 496M 0% /proc/scsi

tmpfs 496M 0 496M 0% /sys/firmware

Now we have to move into myvolume follow below command

>>root@f5b20b23fd66:/usr/local/tomcat# cd /myvolume

Then we entered root to my volume

>>root@f5b20b23fd66:/myvolume#

Now create folder or file in that my volume follow the below command:

root@f5b20b23fd66:/myvolume# cat >devops and Enter

Now you can write the data as follows and after writing the data save by pressing Ctrl+c

this is devops volume class

^C -- Then it will save

To know the file or folder and date is created or not follow the commands

>>root@f5b20b23fd66:/myvolume# ls

Devops --- it will show folder name

>>root@f5b20b23fd66:/myvolume# cat devops

this is devops volume class -- it will show the data created

Now we have to come to the root directory from /myvolume by using follow commands below:

>>root@f5b20b23fd66:/myvolume#

>>root@f5b20b23fd66:/myvolume# cd

>>root@f5b20b23fd66:~# exit

>>exit

root@ip-172-31-20-67:~# -- Now we are in root directory

ii) For creation of 2nd container i.e c3

>>root@ip-172-31-20-67:~# docker run --name c3 -v/myvolume -d -p 8082:8080 uma:1

It will display like below : it is created successfully

f5b20b23fd6667cbe83b09256704a22715f893e1ee37efb6ca4a7f782637dd80

to check whether it is created or not :

>>root@ip-172-31-20-67:~# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

f5b20b23fd66 uma:1 "catalina.sh run" 15 seconds ago Up 13 seconds 0.0.0.0:8081**->8082/tcp c3**

Now we have to execute the created container follow the commands

root@ip-172-31-20-67:~# docker exec -it c3 /bin/bash

Now we entered to tomcat it will display like below:

root@f5b20b23fd66:/usr/local/tomcat#

To know whether **my volume** is created or not follow the below command

root@f5b20b23fd66:/usr/local/tomcat# df –h

it will show like below and we can check whether it is created or not:

Filesystem Size Used Avail Use% Mounted on

overlay 7.7G 2.9G 4.9G 38% /

tmpfs 64M 0 64M 0% /dev

tmpfs 496M 0 496M 0% /sys/fs/cgroup

/dev/xvda1 7.7G 2.9G 4.9G 38% **/myvolume**

shm 64M 0 64M 0% /dev/shm

tmpfs 496M 0 496M 0% /proc/acpi

tmpfs 496M 0 496M 0% /proc/scsi

tmpfs 496M 0 496M 0% /sys/firmware

Now we have to move into myvolume follow below command

>>root@f5b20b23fd66:/usr/local/tomcat# cd /myvolume

Then we entered root to my volume

>>root@f5b20b23fd66:/myvolume#

Now create folder or file in that my volume follow the below command:

root@f5b20b23fd66:/myvolume# cat >aws and Enter

Now you can write the data as follows and after writing the data save by pressing Ctrl+c

this is aws class

^C -- Then it will save

To know the file or folder and date is created or not follow the commands

>>root@f5b20b23fd66:/myvolume# ls

aws --- it will show folder name

>>root@f5b20b23fd66:/myvolume# cat aws

this is aws class -- it will show the data created

Now we have to come to the root directory from /myvolume by using follow commands below:

>>root@f5b20b23fd66:/myvolume#

>>root@f5b20b23fd66:/myvolume# cd

>>root@f5b20b23fd66:~# exit

>>exit

root@f5b20b23fd66 #--- now we are at root directory

docker inspect: command is used to see whether myvolume is created or not follow the command**.**

root@f5b20b23fd66 #:docker inspect c2

To see the created container and data is existed in <u>myvolume</u> follow the commands :

root@ip-172-31-20-67:~# cd /var/lib/docker/volumes/

Then we entered to volume:

root@ip-172-31-20-67:/var/lib/docker/volumes#

To see the containers existed in the volumes follow the commands:

>>root@ip-172-31-20-67:/var/lib/docker/volumes# ls

>>6fb14cd6269dd6809e0d00905ec2f5d3e436911f1ba77e4da0ad1efe6ed747f3 metadata.db

>>9e989605efefaface5b53a522caf6162060c96644d880446a4566b8accb29988

The above two are the id's of containers

Then we have to move to the one of the container follow the command below and type first two letters of any id and press tab then automatically it will take the complete id as below

>>root@ip-172-31-20-67:/var/lib/docker/volumes# **cd-------**it will come back to volume like below:

>>root@ip-172-31-20-67:/var/lib/docker/volumes/--------now type 1st 2 letters of container id

>>ex: /var/lib/docker/volumes/6fb**--------press TAB button now it will display like below**

>>root@ip-172-31-20-67:/var/lib/docker/volumes/6fb14cd6269dd6809e0d00905ec2f5d3e436911f1ba77e4da0ad1efe6ed747f3/-----**press enter and it move to that container**

>>root@ip-172-31-20-67:/var/lib/docker/volumes/6fb14cd6269dd6809e0d00905ec2f5d3e436911f1ba77e4da0ad1efe6ed747f3# ls ----then it will display the existed data like below

_data

>>root@ip-172-31-20-67:/var/lib/docker/volumes/6fb14cd6269dd6809e0d00905ec2f5d3e436911f1ba77e4da0ad1efe6ed747f3# cd>> **we have to come back to the data** cd /var/lib/docker/volumes/6fb14cd6269dd6809e0d00905ec2f5d3e436911f1ba77e4da0ad1efe6ed747f3/_data

Aftter entering to the data press ls command then it will show the file or folder

>>root@ip-172-31-20-67:/var/lib/docker/volumes/6fb14cd6269dd6809e0d00905ec2f5d3e436911f1ba77e4da0ad1efe6ed747f3/_data# ls

Aws >> our file

Then to display that data existed in that folder or file type below command

>>root@ip-172-31-20-67:/var/lib/docker/volumes/6fb14cd6269dd6809e0d00905ec2f5d3e436911f1ba77e4da0ad1efe6ed747f3/_data# cat aws

This is aws class >> then it shows the data.

Then repeat the above steps for displaying or data of the container 2 follw the steps

>>root@ip-172-31-20-67:/var/lib/docker/volumes/6fb14cd6269dd6809e0d00905ec2f5d3e436911f1ba77e4da0ad1efe6ed747f3/_data# cd

>>root@ip-172-31-20-67:~# cd /var/lib/docker/volumes/

>>root@ip-172-31-20-67:/var/lib/docker/volumes# ls

6fb14cd6269dd6809e0d00905ec2f5d3e436911f1ba77e4da0ad1efe6ed747f3
metadata.db

9e989605efefaface5b53a522caf6162060c96644d880446a4566b8accb29988

>>root@ip-172-31-20-67:/var/lib/docker/volumes# cd /var/lib/docker/volumes/9e989605efefaface5b53a522caf6162060c96644d880446a4566b8accb29988/

>>root@ip-172-31-20-67:/var/lib/docker/volumes/9e989605efefaface5b53a522caf6162060c96644d880446a4566b8accb29988# ls

_data

>>root@ip-172-31-20-67:/var/lib/docker/volumes/9e989605efefaface5b53a522caf6162060c96644d880446

a4566b8accb29988# cd /var/lib/docker/volumes/9e989605efefaface5b53a522caf6162060c96644d880446a4566b8accb29988/_data

>>root@ip-172-31-20-67:/var/lib/docker/volumes/9e989605efefaface5b53a522caf6162060c96644d880446a4566b8accb29988/_data# ls

>> devops our file

>>root@ip-172-31-20-67:/var/lib/docker/volumes/9e989605efefaface5b53a522caf6162060c96644d880446a4566b8accb29988/_data# cat devops

this is devops volume class >> then it shows the data.

Docker Storage (Persistence Storage)

To create a storage follow the below commands

>>root@ip-172-31-20-67:~# mkdir /home/saistore

>>root@ip-172-31-20-67:~# cd /home/saistore

>>root@ip-172-31-20-67:/home/saistore#

To create a persistence volume and container c4 follow below commands

>>root@ip-172-31-20-67:/home/saistore# docker run --name h1 -v /home/saistore:/myvolume -d -p 8083:8080 uma:1

2c59882dea66b74270a5f3eb4e0db2dd3b5dca9ee8b8dabe210e3b2822518cc6

It will display above

The execute the above created command follow the command

>>root@ip-172-31-20-67:/home/saistore# docker exec -it h1 /bin/bash

Then we have to check whether ?myvolume is created or not follow the command

root@2c59882dea66:/usr/local/tomcat# **df -h**

Filesystem Size Used Avail Use% Mounted on

overlay 7.7G 3.1G 4.7G 40% /

tmpfs 64M 0 64M 0% /dev

tmpfs 496M 0 496M 0% /sys/fs/cgroup

/dev/xvda1 7.7G 3.1G 4.7G 40% /myvolume

shm 64M 0 64M 0% /dev/shm

tmpfs 496M 0 496M 0% /proc/acpi

tmpfs 496M 0 496M 0% /proc/scsi

tmpfs 496M 0 496M 0% /sys/firmware

we have to go to the /myvolume directory by following the below command

>>root@2c59882dea66:/usr/local/tomcat# cd /myvolume

>>root@2c59882dea66:/myvolume# -- we entered to /myvolume

We have to create the file follow the command

>>root@2c59882dea66:/myvolume# cat > sql

this is sql server class

^C -- press Ctrl+c

>>root@2c59882dea66:/myvolume# **ls**

sql

>>root@2c59882dea66:/myvolume# **cat sql**

this is sql server class

>>root@2c59882dea66:/myvolume# **cd**

>>root@2c59882dea66:~# **exit**

exit

>>root@ip-172-31-20-67:/home/saistore# --- we are in home storage (i.e persistence storage)

Now we have to create another container in the same volume Follow the commands :

>>root@ip-172-31-20-67:/home/saistore# **docker run --name h2 -v /home/saistore:/myvolume -d -p 8084:8080 uma:1**

3edbc4460c90793f13b950d918ac2be2738e5f6ebcab2d4860ed4e23fbcbae50

>>root@ip-172-31-20-67:/home/saistore# **docker exec -it h2 /bin/bash**

>>root@3edbc4460c90:/usr/local/tomcat# **cd /myvolume**

>>root@3edbc4460c90:/myvolume# **cat > mydb**

my data base class

^C

>>root@3edbc4460c90:**/myvolume# ls**

mydata mydb sql ---- showing the files

>>root@3edbc4460c90:/myvolume# **cat mydb**

my data base class

>>root@3edbc4460c90:/myvolume# **cat sql**

this is sql server class

>>root@3edbc4460c90:/myvolume# cd

>>root@3edbc4460c90:~# cd

>>root@3edbc4460c90:~# exit

exit

>>root@ip-172-31-20-67:/home/saistore# ls

mydata mydb sql

Now if we remove one container (i.e h2)

>>root@ip-172-31-20-67:/home/saistore# **docker rm h2 -f**

h2 ---- **will be removed**

>>root@ip-172-31-20-67:**/home/saistore# ls**

mydata mydb sql --- Data

>>root@ip-172-31-20-67:/home/saistore# **docker run --name h3 -v /home/saistore:/myvolume -d -p 8085:8080 uma:1**

e234b2338a2d6447df6ed1b835e6d01a67c2ed4f17f28d3df86cdf8a976c1a14

>>root@ip-172-31-20-67:/home/saistore# **docker exec -it h3 /bin/bash**

>>root@e234b2338a2d:/usr/local/tomcat# **cd /myvolume**

>>root@e234b2338a2d:/myvolume# **cat > sap**

this is sap class

^C

>>root@e234b2338a2d:/myvolume# **ls**

mydata mydb sap sql

>>root@e234b2338a2d:/myvolume# **cd**

>>root@e234b2338a2d:~# **exit**

exit

>>root@ip-172-31-20-67:/home/saistore# ls

mydata mydb sap sql

>>root@ip-172-31-20-67:/home/saistore#

Short cuts used in Docker:

>>To clear the Screen Short Cut (Press Ctrl+L) Enter

To start the images/container:

>>Syntax:docker:- start <container id/name>

To stop the images/container:

>>Syntax:docker:- stop <container id/name>

To know the running containers and images in Docker:

>>docker images

docker ps-- running Images

To know the running and stopping status of images created:

syntax:docker ps -a

To delete the container:

root@ip-172-31-0-118:~# docker rm $(docker ps -a -q) –f

To delete the images:

root@ip-172-31-0-118:~# docker rmi $(docker images -q) -f

to delete the images

syntax:

>>dokcer rm = container delete

>>docker -rmi = images deletion

21.03.2019 SQL Server:

Docker hub:

>>docker run --name myadminmysql -e MYSQL_ROOT_PASSWORD=admin -d mysql:8.0.15