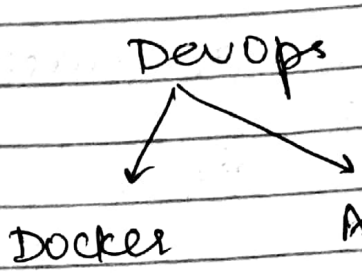
DATE 27 Oct
PAGE NO.:

SESSION : 30



- Elasticity :: LVM

TASK. (i) Hadoop → LVM (Elasticity)
(ii) Linux → change the size of static partition.

- configured yum

wget url → (available on drive)
↳ downloads the url from webpage.

yum repolist
yum install docker-ce

#. vim Docker
docker.repo

[docker]
baseurl = https://download.docker.com/linux/centos/7/x86_64/stable
gpgcheck = 0

DATE : / /
PAGE NO. :

yum install docker-ce --nobest
dnf clean packages
↳ clears all the packages.

systemctl start docker

docker info

- Platform for OS
  (i) physical (Bare Metal)
  (ii) virtual machine (Oracle V.B)
  (iii) cloud computing (AWS)
  (iv) container (Docker)

docker pull ubuntu:14.04
docker run -it ubuntu:14.04
docker ps
docker stop kind-brattain
docker start

DATE 5 NOV.
PAGE NO.:

SESSION: 36

docker search centos
docker run -it --name web1 centos

yum whatprovides ifconfig

docker cp z.py web1:/root
↳ copies the file z.py from base os to docker container (web1).

docker cp web1:/root/aa.txt /mnt/
↳ copies file (aa.txt) from docker to base os.

docker logs web1
↳ from base os ; ~~docker~~ gives the exact screen of what is happening inside the container.

TASK. launch an GUI program in docker container.
launch jupyter inside d.c.

yum install httpd

netstat -tnlp
↳ shows all the running ports.

DATE : / /
PAGE NO. :

/usr/sbin/httpd

↳ Instead of systemctl we can use it to start httpd services inside a docker container.

xox

DOCKER

- commit ; we can create a customised image.

yum install httpd
cd /var/www/html
vim index.html
/usr/sbin/httpd
  & starts httpd services
yum install net-tools

commit

base OS # docker commit os8 myweb:v1

↳ clones os8 to myweb:v1

DATE : 4/Dec
PAGE NO. :

SESSION : 46 DOCKER

ps -aux | grep httpd
↳ checks if httpd service is start/stop

- commit will only clone the storage. Then we manually need to start the services.

vi /root/.bashrc
↳ makes the cmd permanent.

killall httpd → stops httpd service.
yum provides killall
#

<<last line>>
/usr/sbin/httpd
↳ inside /root/.bashrc

DATE: 24 Dec 20
PAGE NO.:

SESSION: 60 ANSIBLE DOCKER

TASK: Blog, Used cases of Kubernetes, Companies using, the growth

docker images
docker run -it --name w1 centos:latest

- SDN - Software Defined Networking
- SNAT is only available in docker. i.e docker can connect to him but him can't connect to docker.
- If we want to visit a web page then we have to use the browser of base os.

docker network ls
↳ Docker creates a isolated network of its own, known as bridge.

- Modem = Router + Switch
↳ Also known as L3 switch. (support)
Devices which have ip ^

- bridge.enable_ip_masqueeding
↳ means SNAT enabled

DATE: / /
PAGE NO.:

07

- PAT: Port address Translation or Port mapping
  - ↳ Done to enable DNATing.
- We do PATing while launching the OS.

docker run -it --name <<-->> -p 2222:80
Centos: latest

↑ PATing

✗ http://172.17.0.2:2222/index.html

http://192.168.0.129:2222/index.html
↳ we have to go to the base OS (ip) & the registered port.

PAGE NO.:
DATE: 20 Jan 21

SESSION 16 DOCKER

- we can either commit a docker container or can create a dockerfile.

docker info

TASK: Python menu. for all the technologies
- (Voice Recognition)
- configuration : Ansible

docker run -it centos:latest

docker cont. # yum install httpd -y
yum install net-tools -y
<copy files in /var/www/html/>
docker cp index.html <doc_id>:/var/www/html
/usr/sbin/httpd
↳ Not permanent
vi /root/.bashrc
↳ /usr/sbin/httpd

* **echo** is a non-interactive program that can be used for automation
↳ echo "hi" >> /root/.bashrc
echo /usr/sbin/httpd >> /root/.bashrc

docker commit <doc-id> web:v1

PAGE NO. :
DATE : / /

mkdir /dws
cd /dws
vim Dockerfile ← compulsory name for dockfile

~~ESC~~

```
FROM centos:latest
RUN yum install httpd -y
RUN yum install net-tools -y
  # create a file in same dir (index.html)
COPY index.html /var/www/html
RUN /usr/sbin/httpd
RUN echo /usr/sbin/httpd >> /root/.bashrc
```

:wq

docker build -t newweb:v1 /dws/
↳ image is build

docker images
↳ newweb:v1

docker history newweb:v1
↳ gives all the steps for creating the image.

docker run -it newweb:v1
rpm -q net-tools
ls /var/www/html/

- Now, we can also give our img to K8s for launching a pod.

curl <<ip>>
↳ Checks service

PAGE NO. :
DATE : / /

- .bashrc file is only in the OS with bash scripting. eg: CENTOS7HA
- ~~For~~ In Dockerfile. If we want to run a cmd everytime we launch an OS/container we use the keyword **CMD**. (runtime)

vim Dockerfile

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

↓

["/usr/sbin/httpd", "-D", "FOREGROUND"]

↳ In CMD every option is written in list format

docker run -itd newweb:v1

PAGE NO.:
DATE: 22/Jan/21

SESSION: 77 DOCKER

- Build Time : Time required to build a ~~Code~~ - On keyword eg. at build. Eg RUN, COPY etc execute
- Run Time : Some keywords run/ at run time. Eg: CMD.

- History cmd on an image gives the metadata or how it is created.

docker run -it --name z1 --rm centos:latest
date

↳ It will run a container only for the date cmd. As soon as it is completed they terminate the container. without --rm option they only stop the container.

- It is compulsory to pass a cmd at the run time. But till now we have always run without any cmd.
- The ~~cmd~~ by default cmd is /bin/bash. Dockerfile always run this cmd. 99.99% of images come with this option. We can check by:
  # docker history centos:latest
- As soon as the program/app is live/running state, the container will run. And the life of /bin/bash

is infinite.

- Exit is the end of bash. So technically we close the program (bash), not a container. So in turn, the container also stops (bec container can't run without any program).

```
# mkdir Dockerfile
  FROM centos:latest
  CMD date

# docker build -t date:v1 .
                          ↑
                  This means current dir

# docker history date:v1
# docker run date:v1
    ↳ Run date cmd and stop the container.
# Dockerfile
  FROM centos:latest
  RUN yum install net-tools -y
  RUN mkdir /zz        } Persistent data
  RUN touch /f.txt     } on HDD
  RUN echo hello  —— volatile data stored in layer
  CMD ifconfig
```

writable to see the changes

layer is not created

```
  ENTRYPOINT ["ls"]
```

↳ It is similar to CMD. But the benefit is it also allows arguments. But CMD don't.

PAGE NO. :
DATE : / /

docker run date:v1 -a
↳ will show list with -a option. But CMD will show error. Or replace it with the cmd (-a).

Dockerfile
ENTRYPOINT ["httpd"]
CMD ["-DFOREGROUND"]
EXPOSE 80

docker run -d --name b1 -p 2222:80 date:v1
↳ Now we have exposed 80 so we don't need this cmd.

* docker run -d --name b1 -P date:v1
↳ This will give the exposed port as target port (80) and assign a random port for service port (2222).

PAGE NO.:
DATE 12 Feb 21

SESSION: 92 Docker

- API : Application programming Interface

docker info
docker login
  ↳ we can link dockerhub.
docker run -it --name c1 centos:latest
#
ssh <docker>
  ↳ connection refused

- ssh works on port no 22.

#yum install net-tools -y
netstat -tulp
  ↳ NO ssh (22 port)
yum install openssh-server -y
/usr/sbin/sshd
  ↳ systemctl start sshd
  ↳ failed! no pvt key avail
ssh-keygen -A
ls /etc/ssh
  ↳ confirms we created pvt/pub key
/usr/sbin/sshd
  ↳ worked.
netstat -tulp
  ↳ port 22 enabled!!
vi /etc/shadow
  ↳ contains all users & passwd
  ↳ root:!!:locked:0:

PAGE NO. :
DATE : / /

yum install passwd -y

- If the password is locked and we change the password, it automatically unlocked.

password root
↳ redhat

docker commit c1 centos_ssh:v1

↳ we won't be able to see the ssh service not started from the ~~committed~~ image. To make it permanent copy the cmd in ~~root~~ /root/.bashrc

~~vim~~ mkdir /dussh
vim Dockerfile
FROM centos:latest
RUN yum install net-tools -y
RUN yum install openssh-server -y
RUN ssh-keygen -A
RUN ~~password root~~ yum install passwd -sddn
RUN echo redhat | passwd root --stdin
CMD /usr/sbin/sshd

docker build . -t vimal13/centos_ssh:v1
↳ creates and pushes it on dockerhub

PAGE NO. :
DATE : / /

#docker run -it --name c4 vimal13/centos_ssh:v1
↳ will be terminated

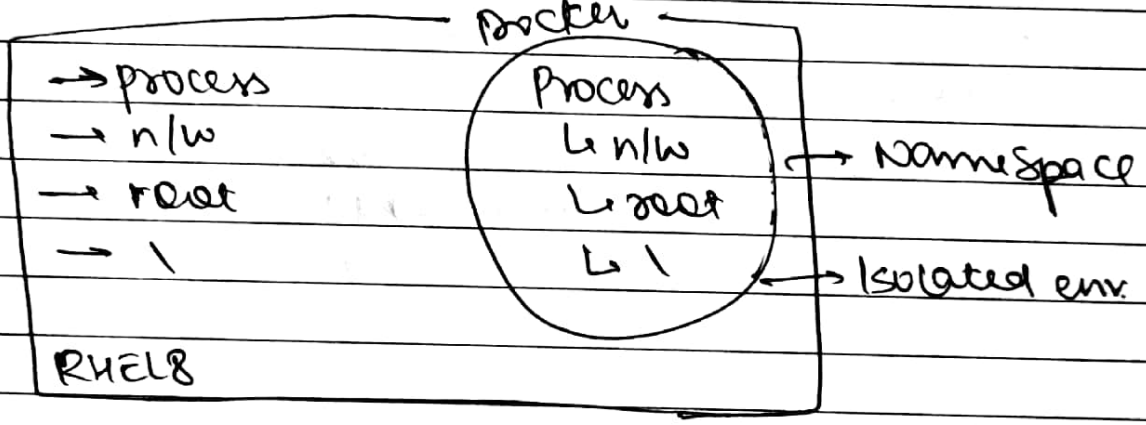
*. Find ssh option to make it ~~as~~ in background ~~ed~~

SESSION: ~~18~~ CRI-O / PODMAN

PAGE NO.:
DATE: 31/Mar/21

- Docker never launches the container. It will only send the ~~cmd~~ instruction
container run-time is the program that starts a container.
↳ Runc (exp.)

- Container == Process
↳ Containers run inside ~~the~~ a process (bash)



- Container Image : Contains all the data we have to mount & on the / drive.

- Docker has the capability to manage images, unar it, create images, mount data on it, etc.

- Docker is only a container management tool. It neither is container runtime nor container engine.

PAGE NO. :
DATE : / /

- Docker server program
  - ~~Containerd~~ containerd
    - It is the one that goes to runc

aws

yum install runc -y
runc spec
  - Creates a config file
  - config.json

vi lw.go ⟶ # yum install go -y

```
package main
func main(){
    println("hi")
}
```

go build -o lw

vi lw.go

```
package main
import "os"
import "fmt"
func main(){
    fmt.Println(os.Hostname())
}
```

go build -o lw

vi config.json

```
"args": [
        "/lw"
],
```

PAGE NO. :
DATE : / /

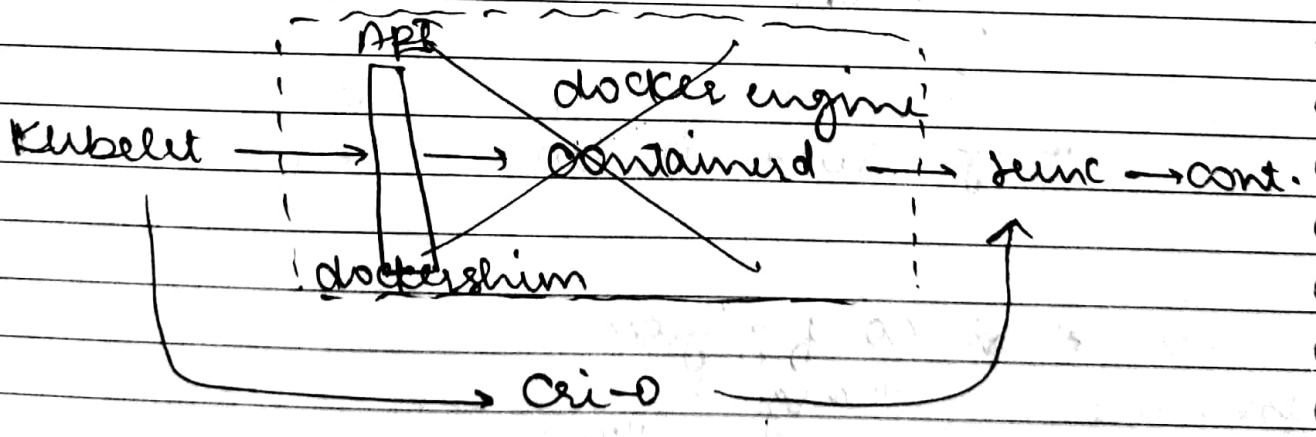
mkdir rootfs
tar -xf ubuntu.tar -C rootfs/

runc create myctl

nsenter -t 16841
↳ pid of runc
↳ This will enter us inside
- u → hostname
- n → network
- t → target

↳ somehow similar to docker attach cmd.

runc list
runc start myctl
↳ cmd will run inside the container and o/p will be displayed.



PAGE NO. :
DATE : / /

- OCI → Open container Initiative
  ↳ a standard is created by which we can switch in-bet products.
  ↳ ex: we can run docker image from podman
- podman is service less program that directly goes to runc and launches the container
  - ~~Especially~~ podman = = docker