Ubuntu 14.04

1)Installing Docker

apt-key adv --keyserver hkp://pgp.mit.edu:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D

echo "deb https://apt.dockerproject.org/repo ubuntu-trusty main" > /etc/apt/sources.list.d/docker.list

apt-get update

apt-get purge lxc-docker\*

apt-cache policy docker-engine

apt-get install docker-engine

2)Creating Docker Group and adding User(jithinr)

//groupadd <groupname>

groupadd docker

//sudo gpasswd -a <username> <groupname>

sudo gpasswd -a jithinr docker

//Permission to "docker" group

// chown root:<groupname> /var/run/docker.sock

chown root:docker /var/run/docker.sock

reboot

//Cross-checking the permission for the group

cat /etc/group | grep -i docker

3)Resolving network issue for ssh installation and installing open ssh-server on Ubuntu image

//Resolving n/w issue for ssh installation

vi /etc/resolv.conf

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)

# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN

#nameserver 127.0.1.1

nameserver 172.27.172.10 //Cybage Network

search cybage.com

vi /etc/default/docker

//Installing ssh-sserver on Ubuntu Image

sudo apt-get install openssh-server

sudo service ssh restart

4)docker basic commands

docker -v

docker version

docker info

5 a) To create an image

docker build -it ubuntu /bin/bash

5 b)To start a container

//ubuntu :- image name

// -it :- make it interactive and assign TTY

/bin/bash :- inside the container we are running bash process

docker run -it ubuntu /bin/bash

6)//Using docker unit socket

sudo netstat -tlp //netstat :- network status , tlp :- tcp listening program

//Using network

docker -H <ip\_address>:port -d & // -d :- daemon mode , & - for giving cmd back

docker -H 172.27.59.190:2375 -d &

//For checking docker connection

ps -aux | grep docker

7)To check container status

//Active Container

docker ps

//Conatiners that have ever run on the machine

docker ps -a

//Location of the container on local machine

/var/lib/docker/<storage\_driver>

/var/lib/docker/aufs/diff

8)For seeing docker images

docker images

9)Docker container is still running and we are on normal host machine(Detaching our container)

Control + P+Q

docker ps // For checking current running container

10)Stopping container from outside of the container

docker stop {<container \_id>/<container\_name>}

11)//Last container that have run

docker ps -l

//For starting and opening container

docker start <container\_id> // we won't go inside container

docker attach <container\_id> // we will go inside container

//Removnig an inactive container

docker rm {<container\_id>/<container\_name>}

//Removing a runnig/active container

docker rm -f {<container\_id/<contanier\_name>}

//Creating alias

alias <alias\_name>="<command>"

alias dps="docker ps" //Example

12)Exact command inside a container for executing bash terminal

a)nsenter(entering name space)

#Need Pid(Process Id) of the container

docker inspect <container\_id> | grep Pid

nsenter -m -u -n -p -i -t <container's\_Pid> /bin/bash

b)docker-enter <container\_id>

c)docker exec -it <container\_id> /bin/bash

13)Pushing images to repository

Before pushing the images,we need to tag it first

a)docker images

b)docker tag <image\_id> <repo\_name where it is to be pushed:version>

c)docker push <repo\_name where it is to be pushed:version>

14)Removing Containers/Images

//Removing Containers

docker ps -a

docker rm <container\_id>

//Removing Images

docker images

docker rmi <image\_id>

15)Difference b/w CMD and RUN

CMD RUN

a)run-time a)build-time

b)run-commands in container in launch time b)add layers to images

c)equivalent to c)used to install apps

docker run <args> <command>

docker run <args> /bin/bash

d)one per Dockerfile

CMD takes two forms/styles

(i)shell form

commands are expressed the same way as shell

commands

(ii)exec form

JSON array style(ie., ["command","arg1"])

Container does not need a shell

Avoid string munging by the shell

No shell features:-

No variable expansion,special characters

ENTRYPOINT

a)Can't be overridden at the runtime with the normal commands

docker run <command>

Any command at run-time is used as an argument to ENTRYPOINT

Ex :- docker build -t = "hw2" .

docker run -it hw2 /bin/bash

instead of considering /bin/bash as a command

it is interpretted as an argument to the ENTRYPOINT

If ENTRYPOINT is specified,CMD in Dockerfile also get interpretted as arguments

CMD instructions in the docker gets overridden at runtime art options at the docker run command

16) ENV command

ENV var = value

17)VOLUME

a)Decoupling of data from the container

docker run -it -v /test-vol --name=voltainer ubuntu14.04 /bin/bash

where

-v //volume, /test-vol //foldername, voltainer //container name

b)Second container can access the data of first container and share its contents

docker run -it --volumes-from=voltainer ubuntu14.04 /bin/bash

<1st container name>

c)Mounting a directory from docker host to container(host\_mount)

docker run -v /data:/data

//(docker host):(container)

d)Deleting volumes

Delete the volume within the container

docker rm -v <container\_name>

if we delete the container without deleting volume,volume will be still remain

18)

a)By default each container get one interface automatically attached to the docker0

b)By default resolv.conf of every container is same as resolv.conf of docker host

c)Exposing port at the runtime

We need to launch the docker container with -p option when doing docker run

d)Port forwarding

docker run -p 5001:80

5001 port on docker host; 80 is the port on the container

docker run -d -p 172.27.59.96:5003:8080 --name=web3 tomcat7

-P to expose all the ports

docker run -d -P --name=web4 tomcat7

e)Container Linking

Pros:- More secure than exposing port

Cons:- Only container-container communication

docker run --name=src -t tomcat7

where src is the name of the container, tomcat7 is the name of the image

docker run --name=rcvr --link=src:ali-src -it for\_demo /bin/bash

where rcvr is the name of the container, src is the name of the previous container, ali-src is the alias-name of the current container

for\_demo is the name of the image from which rcvr container is getting created

19)docker commands

docker run -it --name=tomcat7container tomcat7

docker run -it --name=mysqlcontainer mysql /bin/bash

/var/lib/tomcat7/webapps

To delete all the containers

docker rm `docker ps -qa`