Containerisation Tool Docker :

--image
--container
--dockerhub
--push and pulling images

--what is Image ?

Linux Commands ------> Kernal

-->docker image is light weight component
-->image with no kernal

container-1 with no kernal

-->docker Architecture :
---

1)Launch ubuntu EC2 on aws

2)Install Docker on ubuntu

3)Login to Ubuntu

4)Commands

sudo su -

apt update -y

apt install docker.io -y

docker --version

docker pull ubuntu

docker images

docker run -it -p 81:80 --name web1 -d ubuntu
docker run -it -p 82:80 --name web2 -d ubuntu
.

.

docker container ps

docker stop <container-id/container-name>

docker container ps -a

docker pause web3

docker container ps

docker rm <container-id/container-name>

docker rm -f <container-id/container-name>

docker start <container-id/container-name>

docker unpause <container-id/container-name>

#login to container

docker exec -it <container-id/container-name> bash

exit

docker container ps

#execute on browser

http://43.205.143.147:83/readme.html

hosting a website into docker container

down load and install winscp

connect to linux machine using winscp

download a HTML template on windows Folder

copy template to linux machine

docker cp . web3:/var/www/html

---

7)docker create --help
8)docker inspect <container-id>/<container-name>
9)docker exec -it <container-id> pwd (present working directory)
10)docker run -dit -w /shashi ubuntu (container with working directory)
11)docker exec -it <container-id> pwd
12)docker exec -it <container-id> ls /
13)docker exec -it <container-id> env
14)docker run -dit --env Java_Home=/usr/share/java/jdk-1.8 --env
JRE_HOME=/usr/share/java/jre ubuntu
15)docker exec -it <container-id> env

vi env.list
JAVA_HOME=/usr/share/java/jdk-1.8
MAVEN_HOME=/usr/lib/maven

docker logs <container_id> --details
docker logs <container_id> -t

>To remove all running contaniers
 docker container rm -f $(docker container ps -aq)
>To remove all images
docker rmi $(docker images -a -q)

---

setting memry limit for container

docker container run --rm -it -d --name mem-limit-demo --memory=256m nginx:alpine

docker stats mem-limit-demo --no-stream --format "{{ json . }}" | python3 -m json.tool
-------->
docker container run --rm -it -d --name soft-mem-limit-demo --memory=1g
--memory-reservation=512m nginx:alpine

docker stats soft-mem-limit-demo --no-stream --format "{{ json . }}" | python3 -m json.tool

---------->
cpu limit

$ docker container run --rm -it -d --name cpu-limit-demo --cpus=1 nginx:alpine
$ docker container run --rm -it -d --name cpu-sets-demo --cpus=1 --cpuset-cpus=2 nginx:alpine
---

docker restart <container name> Restart a container
docker stats Show running container stats
docker system df Check docker daemon disk space usage
docker system prune -af Remove images, networks, containers, and volumes

docker pause <container>
docker unpause <container>