Docker

* Docker was first released in march 2013 it is developed by Solomon hykes and sebastianpahl.
* Docker is an open-source centralized platform designed to create, deploy and run the application
* Docker uses a container on the host O.S to run applications .it allows the application to use the same Linux kernel as a system on the host computer, rather than creating a whole virtual O.S
* We can install docker on any O.S but the docker engine runs natively on the Linux distribution.
* Docker is written in ‘go’ language
* Docker is a tool that performs O.S level virtualization, also known as containerization
* Before docker, many users face the problem that a particular code is running In the developer’s System but not in the user’s system.
* Docker is a set of platforms as a service that uses O.S level virtualization wheres VMware uses hardware-level virtualization.

Ques 1:- Does Container contain any OS?

Ans:- Yes, the container contains an operating file in it but that file is very small compared to the actual OS

For example, one OS file of 2GB, and the container file is 60 MB. That’s why the container is called lightweight it is negligible.in the container, there is 5% of the operating file is in image form, the supported file we can say that’s why we say the container doesn’t contain any operating file.

When the image is in the running condition is called a container {running stage of the image is called a container and when the container is in stop condition is called image.

Image will be moved from the development server to another side, the container will not move.

Advantaged of docker:-

* No pre-allocation of RAM
* CI efficiency:- docker enable you to build a container image and use that same image across every step of the deployment process.

Meaning:- this code is running my side you have ishu, and you don’t know how to do, in short, blame game is over with the help of this.

* Less cost
* It is light in weight {the application which needs to use few resources to run is called light weight}
* It can run on a physical H/W virtual machine or on cloud
* You can re-use the image.
* It took very less time to create a container

Disadvantages of docker:-

* Docker is not a good solution for applications that require rich GUI.
* Difficult to manage a large amount of containers.
* Docker does not provide cross-platform compatibility means if an application is

Designed to run in a docker container on windows, then it cant run on Linux or vice versa.

* Docker is suitable when the development OS and testing OS are same if the OS is different

We should use VM.

* NO solution for data recovery & backup.
* Component of docker

Docker daemon/engine

* Docker daemon run on the host OS.
* It is responsible for running containers to manage docker service.
* Docker daemon can communicate with other daemons.

Docker clints

* Docker user can interact with docker daemon through a clint.
* Docker clint uses command and rest to communicate with the docker daemon.
* When a clint runs any server command on the docker clint terminal, the clint terminal

Sends these docker command to the docker daemon

* It is possible for docker client to communicate with more then one daemon.

Docker host:-

* Docker host is used to provide an environment to execute and run application. It contains the docker daemon, images, containers ,network and storage…….

Docker hub/registry

* Docker registry manages and store the dicker images.
* There are two types of registry in docker

1)PUBLIC REGISTRY:-

Public registry is also called a docker hub.

2)private registry :-

it is used to share images within the enterprise.

Docker images

* Docker images are the read only binary templates used to create docker container’s

Or singlr file with all dependencies and configuration required to run a program.

Ways to create an images:-

* Take image from docker hub.
* Create image from docker file.
* Create image from existing docker containers.

Docker container

* Container hold the entire package that is needed to run the application

Or in other words, we can say that, the image is a template and the container is a

Copy of that temple,

* Container is like a virtual machine
* Images become container when they run on docker engine.

Basic command in docker:-

* To see all images present in your local machine

[ ]# docker imanges

* To find out imsges in docker hub

[ ]# docker search Jenkins

* To download image from docker hub to local machine

[ ]# docker pull Jenkins

* To give name to container

[ ]# docker run -it –name Bhupinder ubuntu /bin /bash

Internel mode terminal

* To check service start or not

[ ]# service docker status

* To start container

[ ]# docker start Bhupinder

* To go inside container

[ ]# docker attach Bhupinder

* To see all container

[ ]#docker ps -a

* To only running containers

[ ]# docker ps {process status}

* To stop container

[ ]# doker stop Bhupinder

* To delete container

[ ]# docker rm bhupinder

Docker file creation

-dockerfile is a basically a text file.it contains some set of instruction.

Automation of Docker image creation.

Docker component

FROM:- For base image this command must be on top of the docker file

RUN:- to execute command it will create a layer in image

MAINTAINER:-author/owner/description

Diff in copy and add:-

If we want to copy something then it must be available in local machine other wise we have to create first.

COPY:- Copy file from local system{docker VM}we need to provide source ,destination

We cant download file from internet and only remort repo.

ADD:-similar to copy but ,it provides a feature to download file from internet,also we

Extract file at docker image side.

EXPOSE:-to expose ports such as port 8080 for tomcat ,port 80 for nginx etc

WORKDIR:-To set working directory for a container.

CMD:-execute command but during comtainer creation.

ENTRYPOINT:-similar to CMD but has higher priority over CMD first command will be

Executed by ENTRYPIONT only.

Difference-----entrypoint has higher priority we can set which command execute first and others

ENV:-environment variables

ARG:-

HOW TO CREATE DOCKER IMAGE AND CONTAINER:-

DOCKER FILE

* 1. Create a file named Dockerfile
* Add instructions in Dokerfile
* Build Docker file to create an image
* Run the image to create a container

1. Vi Dockerfile

2. FROM ubuntu

RUN echo “technical guftugu”>/tmp/testfile

* TO CREATE IMAGE FROM DOCKER

Docker build -t myimg. ----🡪 -t =tag [name] . = current directory

Docker ps -a

Docker images

NOW,CREATE CONTAINER FROM THE ABOVE IMAGES

Docker run -it - - name mycontainer myimg /bin/bash

Now check Cat /tmp/testfile

DOCKER VOLUME

* Volume is simply a directory inside our container
* Firstly,we have to declare this directory as a volume and then share volume
* Even if we stop container,still we can access volume
* Volume will be created in one container
* You can declare a directory as a volume only while creating container .
* You cant create volume from existing container.
* You can share one volume across any number of containers.
* Volume will not be included when you update an imag
* We can mapped volume in two ways :-

Container <- --------- >container

Host <- -------------- ->container..

BENEFITS OF VOLUME

* Decoupling container from storeg
* Share volume among different container
* Attach volume to container
* On deleting containers volume does not delete.

CREATING VOLUME FROM DOCKERFILE

1 create a dockerfile and write

FROM ubuntu

VOLUME [“myvolume1”] :wq

2 Then create image from this dockerfile

Docker build -t myimage

3 Now create a container from this image & run

Docker run -it -- --name container1 myimage /bin/bash

4 Now do ls,you can see myvolume1

* Now share volume with another container

Container1< -------------------- > container2

* Docker run -it -- --name container2

--privileged=true –volumefrom container1 /bin/bash

🡪 Now after creating container2,myvolume1 is visible whatever you do in one volume,

Can see from other volume.

Touch /myvolume1/samplefile

Docker start container1

Docker attach container1

Ls

We can see samplefile here

exit

Q.how can we send volume?

🡪 with the help of command and docker file we can create volume and share with

CONTAINER TO CONTAINER and

HOST TO CONTAINER {create volume on ec2 and container }

Benefits of volume :-

* Decoupling conainer from storage
* Share volume among different container
* Attach volukme to container
* On deleting container volume does not delete.

Now,try to create volume by using command

🡪docker run -it - - name container3 -v /volume2 ubuntu /bin/bash

Do ls ---------------🡪 cd /volume2

Now,create one file cont3file and exit

Now create one more container ,and share volum2

🡪docker run -it - - name container4

- - privileged =true - -volume-from container3 ubuntu /bin/bash

Now we are inside container do ls, we can see volume2

Now create one file inside this volume and then check in container3,we can see that file.

VOLUME (HOST-CONTAINER)

* Verify files in /home/ec2-user
* Docker run -it - - name hostcont -v /home /ec2 -user:/rajput - - privileged=true ubuntu /bin/bash
* Cd /Rajput

Do ls, now you can see all files of the host machine

* Touch Rajputfile (in a container )

Exit

Now check in EC2 machine,you can see that file

SOME OTHER COMMANDS

* Docker volume ls

All local volume will display

* Docker volume create < volumename>

For creating volume on Local machine

* Docker volume rm <volume =name >

For deleting specific volume

* Docker volume prune

it removed(delete) all unused docker volumes

* Docker volume inspect <volumename>

For detail about volume

* Docker container inspect<containername>

For detail about container.

DOCKER PORT EXPOSE

Login into aws account create one linux instance

Now go to putty 🡪login as🡪 ec2 user

Sudo su

Yum update -y

Yum install docker -y

Service docker start

Docker run -td - - name techserver -p 80:80 ubuntu

Docker ps

Docker port techserver o/p🡪80/tcp--🡪0.0.0.0/80n

Docker exec -it techserver /bin/bash

Apt-get update

Apt-get install apache2

Cd /var/www/html

Echo “subscribe techguftgu”

Q:difference between docker attach and docker exec?

Ans: docker exec create a new process in the containers environment while docker attach just connect the standard input/output of the main process inside the container to corresponding standard input/output error of current terminal.

Docker exec is specially for running new thing in a already started container, be it a shell or same other process..

Q: WHAT IS THE DIFFERENCE BETWEEN EXPOSE AND PUBLISH A DOCKER ?

ANS: basically we have three option

1)neither specify expose nor -p

2)only specify expose

3)specify expose and -p

1)if we secify neither expose nor -p ,the service in the container will only be accessible from inside the container itself..

2)if you expose a port ,the service in the container is not accessible from outside docker,but from inside other docker containers so this is good for inter-container communication.

3)if you expose and -p a port, the service in the container is accessible from anywhere, even outside docker.

HOW TO PUSH DOCKER IMAGE IN DOCKERHUB

GO to aws account🡪select amazon machine

Now go to putty 🡪login as ec2-user

* Sudo su
* Yum update -y
* Yum install docker -y
* Service docker start
* Docker run -it ubuntu /bin/bash

Now create some file inside the container

Now create image of this container

* Docker commit container1 image1

{Docker commit and then container name which containers image we want and then name of image }

* Now create account in hub.docker.com

Now go to ec2 instance

* Docker login

Enter your username and password

Now give tag to your image

* Docker tag image1 docker id/new image
* Docker push docker id/new image

Now we can see this image in docker hub account

Now create one instance in Tokyo region and pull image from hub

* Dockerer pull docker id/new image
* Docker run -it - - name mycon docker id/new image /bin/bash

Some important command

1]stop all running containers: docker stop $(docker ps -a -q)

2]delete all stop container: docker rm $(docker ps -a -q)

3]Delete all images: docker rmi -f $(docker images -q)