------------------------------------DOCKER-----------------------------

Lec-24 What is Docker Architecture & Container

Docker was first Release in March 2013 It is developed by Solomon Hykes and Sebastian Pahl

→Docker is a Set of Platform as a Service that uses Os Level Virtualization Whereas VMware Virtualisation Hardware level

→Docker is an open-source Centralised Platform designed to Create, deploy and Run applications.

→Docker uses Container on the host Os to run applications It allows applications to we the Same linux Kernel as a system on the host Computer, rathen System than Creating a Whole Virtual OS .

→ We Can install docker on any OS but Docker engine runs natively on Linux distribution .

→Docker written in go language

→ Docker is a tool that performs OS Level Virtualization, also Known as Containerization .

→ Before Docker, many that users faces the particular

Code is problem running in the developer's system but not in the User's System

Lec-25-Advantages, Dieadvantages & Architecture of Docker

Advantages of Docker

→No pre-allocation of RAM

→CI Effaiency = Docter enables you to build a Container image and we that same image across every step of the deployment process

→Leas Cost

→It is light in Weight

→It can run on physical H/w / Vertual H/w or on Cloud

→You Can re-use the image

→It took Very lees time to Create Container

Disadvantages of Docker

Disadvantages of Docker

→Docker is not a good Solution for application that requires rich GUI

→Difficult to manage large amount of Containers

→Docker does not provide Cross-platform Compatibility means if on application is designed to run in a docker

Container on Windows, then It Can't run on linux or vice-versa

→Docker is suitable when the development Os and testing 05 are Same If the OS is diffrent, we should use VM

→No solution for Data Recovery & Backup

DOCKER ECO-SYSTEM

Set of s/w or Packages

1)Docker Client

2)Docker Hub

3)Docker Deamon or server or Docker engine

4)Docker hub or Registry

5)Docker images (template)

6)Docker Compose

Components of Docker

Docker Daemon

→ Docker daemon runs on the Host OS

→ It is responsible for running Containers to manages docker Services

→ Docker Daemon Can Communicate with Other demons

Docker Client

Docker users Can interact with docker through a Client

→Docker Client uses Commands and Rest API to Communicate with the docker daemon

→ when a Client runs any Server Command on the docker Client terminal, the Client terminal Sends these docker commands to the docker daemon

→It is possible for docker Client to Communicate with more than one daemon

Docker Host

Docker Host is used to provide environment an to execute and run applications It Contains the docker daemon, images, Containers, networks and Storages.

Docker Hub/Registry

Docker registry manages and Stores (Private

) the docker images.

There docker are two types of registries in the

1)Public Registry→ Public registry is also called as docker hub .

2)Private Registry→ It is used to share images within the enterprise.

Docker images

→ Docker images are the read only binary templates docker Containers used to Create docker containers.

or

Single file with all dependencies and Configuration required to run a program

Ways to Create an Images

1)Take image from docker hub

2)Create image from docker file.

3)Create image from existing docker Containers.

Docka Container

→ Containers hold the entire packages that application 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 template.

→Container is like a Virtual Machine.

→ Images becomes Container when they run on docker engine.

Lec-26 Basic Commands in Docker

→To see all images present in your local machine

```
[root@ip-172-31-0-45 ec2-user]# docker images
```

1)To find out images. in docker tub

[root@ip-172-31-0-45 ec2-user]# **docker Search (image name Jenkins)**

2)To download image from dockerhub to local machine

[root@ip-172-31-0-45 ec2-user]# **docker pull jenkins**

3)To give name to Container

[root@ip-172-31-0-45 ec2-user]# **docker run -it-name bhupinder ubuntu /bin/bash**

I=Interactive mode , t=Terminal

4)To Check, Service is start or not

[root@ip-172-31-0-45 ec2-user]# **Service docker status**

5)To Start Container

[root@ip-172-31-0-45 ec2-user]# **docker Start bhupinder**

6)To go inside container

[root@ip-172-31-0-45 ec2-user]# **docker attach bhupinder**

7)To See all Containers

[root@ip-172-31-0-45 ec2-user]# **docker ps -a**

8)To See only running Containers

[root@ip-172-31-0-45 ec2-user]# docker ps (Process status)

9)To Stop Container

[root@ip-172-31-0-45 ec2-user]# **docker stop bhupinder**

10)To delete Container

[root@ip-172-31-0-45 ec2-user]# **docker rm bhupinder**

Lec-27 Dockerfile Components & diff Command

→ Login into AWS account and Start your EC2 instance Access it from putty!

→ Now we have to Create Container Our Own image from

Therefore, Create one Container first

→ **docker ubuntu run -it --name bhupicontainer /bin/bash**

→ **cd tmp/**

Now Create One file inside this timp directory

→ **touch myfile**

Now if you want to see the difference between the base image & changes on it then

→ **docker diff bhupicontainer updateimage**

O/P

C /root

A /root/bash-history

C/tmp

A /tmp/myfile

Now, Create image of this Container

→**docker Commit newcontainer update image**

 docker images Now Create Container from this image

→**docker run -it--name rajcontainer updateimage/bin/bash**

root@cid# ls

# cd tmp/

tmp# ls

O/P →myfile   { you will back get all files}

Dockerfile

→Dockerfile is basically a text file It Contains Some set of instruction

→Automation of Docker image Creation

Docker Components

FROM → For base image This Command must be on top of the dockerfile

RUN →To execute Commands, it will Create a layer in image

MAINTAINER → Author/ Owner / Description

COPY Copy files from local system (docker VM) We need to provide Source, destination (We Can't downbad file from internet and any remote repo)

ADD →Similar to COPY but, it provides a feative to download files from internet, also we extract file at docker image side.

EXPOSE →To expose ports such as 8080 for tomcat, port 80 for nginx etc

WORKDIR →To set working clirectory for a Container

CMD →Execute Commands but during Container Creation

ENTRYPOINT → Similar to CMD, but has higher prionty over CMD, first commands will be executed by ENTRYPOINT only

ENV → Environment Variables

Dockerfile

1) → Create a file named Dockerfile(Capital)

2) → Add instructions in Dockerfile

3)→Build dockerfile to Create image

4)→ Run image to Create Container

[vi Dockerfile

FROM ubuntu

RUN echo " Technical guftgu" >/tmp/testfile **]**

To Create image out of dockerfile

docker build –t(tag) myimg

docker ps -a

docker images

Now, Create Container from the above image

docker run -it --name mycontainer mying/bin/bash

→Cat /tmp/testfile

[vi Dockerfile

FROM ubunte Run echo Subscribe Technical giftgu">

ENV myname bhupinderrayput

COPY bestflet /tmp

ADD ted torge Emp

]