Day 1 -

Compute, Network, Storage

US

Cloud(AWS/GCP/Azure)

us-Central(Region01)

AZ01/Datacenter

Baremetal Server 01

Virtual Machines(VM01)

(4GB mem+2vCPU+192.168.1.2)

Storage

Compute(mem+cpu)

Network

AZ02/ Datacenter

AZ03/Datacenter

AZ04/Datacenter

us-east(Region02)

us-west(Region03)

Laptop

C S + Assemble + OS + Apps

Laptop + OS + Apps

Laptop(OS) + Apps

Jenkins(container image)

OS(CI) + Java + Jenkins

Java(CI) + Jenkins

Jenkins

whatsapp = {java(platform) + package(jar)}

Container Image

OS - Ubuntu, CentOS

Platform - Java, Py, Go, Ruby

Tool - Jenkins, Redis, MySQL, Kafka

Application - metaverse, whatsapp, insta

Ports - extension number

+1 123456789 - Hospital

Department?

extn: 31323x1 - Pharmacy

extn: 31323x2 - Pharmacy

extn: 31323x3 - Pharmacy

extn: 31324 - Reports

NO extn - HR

VM IP: 10.0.0.3/localhost

LB(nginx-app)

VM Port: 80 -> ContainerA(nginx):80

VM Port: 81 -> ContainerB(nginx):80

VM Port: 82 -> ContainerC(nginx):80

ContainerD(nginx):80

whatsapp

code(github)

Dockerfile

container image

container

JAVA

code(app.java)

build(app.jar)

cp app.jar

install java

run app(java -jar app.jar)

NODE-APP

code(app.js)

Dockerfile

get node image version 6 —> FROM

create a dir(mkdir) & cd —> WORKDIR

copy app.js to dir —> COPY or ADD <internet-file> file

set port as 80

start the app - node app.js

Container image - docker build

Container

Python

code

Dockerfile

from python

\*install module/framework - Flask

cp app.py

expose

start app

prereq- create index.html

from nginx

cp index.html

expose

start the app

Multi-stage

Stage01:

create a file - index.html

Stage02:

from nginx

cp stage01/index.html .

expose

start the app

Java

Stage01:

mvn image

cp code

mvn package

jar

Stage02:

openjdk image

cp stage1/jar /app

expose 80

start app

Troubleshooting

start a container - python -dt

install git

image from container ?

# Networking

India

Maharastra

Mumbai

Street(2km~200)

Docker Apartment (Door No: 12)

A Block

1st Floor - A101… A110

A101(PG) - 2BHK - 4 Guest

VM01 - Small - 4 Containers

A102(PG) - 3BHK - 6 Guest

2nd Floor - B201… B210

B Block

C Block

AWS

Simplilearn

US-East

DCA-Lab(VPC-200ip’s)

PG-Batch-Apr23(SubnetA-100ip’s)

VM01(ip1) - 172.31.13.194/20

DockerNetwork(default)/bridge/docker0 -

ContainerA - 172.17.0.2/16

ContainerB - 172.17.0.3/16

web - 172.17.0.4/16 & 172.18.0.3/16

netA - 172.18.0.0/16

db - 172.18.0.2/16

ContainerD

VM02(ip2)

PG-Batch-Apr23(SubnetB-100ip’s)

# Docker Compose

3 Tier

User -> Web -> App -> DB

Deployment - DB -> App -> Web

master

- feature1 - Developer -

Use cases:

1) Developer end to end testing

2) Jenkins -> Integration testing

container - Name, port-fwd, vol, network, mode - d or i

docker-compose

services:

web:

build:

dockerfile: web-dockerfile

app:

build:

dockerfile: web-dockerfile

db:

Interview Question:

app/

Code

Dockerfile

docker-compose.yml

1- d-c build

2- d-c up -d

3- d-c down

4- modify the code

5- d-c up -d —build

# Day 5

