**Practice on 27-04-2023**

Capex,opex,physicalserver,vm,roi

Evlution1..directly install servers 222,,,,install on vms ,,,,3)

Docker : docker is packaging technology

Way of packaging image1..devlop on tech and devlop on server

Expectations: containariized,,secure,,manage data,,

Installation apache in ubuntu commands sudo apt update/sudo apt install net-tools,,,

Docker client …docker engine///orchestraion,,demon.,,,runtime,,,-----manual help or cheat sheet

Docker install manual using dockmentation by docker and script based installation

Need to create docker group command **sudo usermod -aG docker (user name)**

run hello world by using command **docker container run hello-world**

install ubntu in docker play ground on ubuntu installed java and tomcat9

command used **$ docker container run -it ubuntu /bin/bash ---ubuntu**

**sudo apt update ///sudo apt install net-tools/// sudo apt install openjdk-11-jdk tomcat9**

**to check net work command 1) ifconfig** check java version ---**2) java -version** check tomcat9 status **3) Sudo systemctl status tomcat9**

**installing spring pet clinic**

**1)creating ubuntu vm ----2)sudo apt update 3) sudo apt install net-tools**

**4)install java—sudo apt install openjdk-17-jdk maven -y 5) pull file from git hub**

**By using command –git close** [**https://github.com/spring-projects/spring-petclinic.git**](https://github.com/spring-projects/spring-petclinic.git)

**6)cd file location( cd spc) 7) command used mvn package**

**8)Run application command Java -jar target /spring-petclinic-3.0.0-SNAPSHOT.jar**

Docker way of working: docker image ---docker file—push image docker hub---create image anywhere

* Understanding docker file /docker image/create container
* Dockerfile

FROM amazoncorretto:17-alpine-jdk

LABEL author=somanath

ADD target/spring-petclinic-3.0.0-SNAPSHOT.jar /springpetclinic.jar

EXPOSE 8080

CMD ["java", "-jar", "/springpetclinic.jar"]

* Create docker image

docker image build -t spc:1.0 .

* to create container

docker container run -d -P spc:1.0

docker container run -d -P spc:1.0

docke next steps’’’ -how to give ip//container architecture\\\genration 1,2,3 //understatnding image//container//life cycle//containraization//networking//networking//storage aspects/

play with docker :docker command –help

image naming

[username]/[repository]:[<tag>]

Pull images from docker hub -----docker image pull nginx/ alpine /(jenkins/Jenkins)/

Remove images ---- docker image rm imageid/ image name

Remove all images at a time – docker image rm $(docker image ls -q)

ubuntu@ip-172-31-42-165:~$ docker container create nginx –create container

docker container run -d nginx:1.22—to run container&create

docker container rm -f container id

giving name

ubuntu@ip-172-31-42-165:~$ docker container run --name somanath1 -d nginx:1.22

remove container //ubuntu@ip-172-31-42-165:~$ docker container rm -f somanath1

remove all running containers //ubuntu@ip-172-31-42-165:~$ docker container rm -f $(docker container ls -q)

stop container //ubuntu@ip-172-31-42-165:~$ docker container stop somanath2

remove all container //ubuntu@ip-172-31-42-165:~$ docker container rm -f $(docker container ls -a -q) (which are in stopped state also

* Docker lifecycle states
  + Created/Running/Paused/Stopped

COMMAND : Docker container inspect --- inspect container image ip ..etc

Port forwarding : ubuntu@ip-172-31-42-165:~$ docker container run -d -p 30000:80 --name somanth nginx

ubuntu@ip-172-31-42-165:~$ docker container run -d -p 30001:8080 --name jenkins jenkins/Jenkins

Random port forwarding

ubuntu@ip-172-31-42-165:~$ docker container run -d -P --name jenkins jenkins/jenkins

spc required

java -=---openjpdk -amazon correto command: docker image pull amazoncorreto:11

docker container run -it -p 30000:8080 amazoncorreto:11

log in the container --- java-verion check

then pull spc in to java curl --curl https://referenceapplicationskhaja.s3.us-west-2.amazonaws.com/spring-petclinic-2.4.2.jar -o spring-petclinic-2.4.2.jar

Run the application java -jar spring-petclinic-2.4.2.jar

Create image from container---- docker image commit amazoncorreto:11