**Docker Notes -25-08-2023**

Docker installation command 👍

<https://docs.docker.com/engine/install/ubuntu/> -> follow this website

Replacement command installation -> apt install docker.io -y

sudo systemctl status docker ->Checking the docker status 👍

Docker images -> see the images in dockers

Docker ps -> running container

Docker ps -a -> seeing all container

docker run -it --name my-first-containner ubuntu:latest /bin/bash

(run the docker image and bin bash will open)

Control pq -> without exiting inside the docker

Exit -> out of container

Docker start container-id -> start the container after exiting the container

Docker attach container-id -> going inside the container

sudo -i -> root user

visudo -> edit privilege

%Jenkins ALL=(ALL) NOPASSWD:ALL

mkdir -m 777 mydocker -> creating new folder

cd mydocker/ -> inside the docker folder

vi Dockerfile -> creating new file for docker image

FROM ubuntu

MAINTAINER kirubha

RUN apt-get update -y

RUN apt-get -y install apache2

ADD . /var/www/html

ENV devops tutorial

docker build -t kirub .

docker images

docker run -it --name kirubh-container kirub:latest /bin/bash

p-> port binding

P-> port forwarding -> automatic assign port number assigned by docker proxy

Command 👍

docker run -itd -P --name jenkins2 jenkins:2.60.3

Docker inspect container-id -> will see the all details about container

Create own network 👍

step1-> docker network create production --subnet=172.30.0.0/16 -> creating network

step2->docker run -itd -p 2026:8080 --network=production --ip=172.34.0.0 --name jenkins jenkins:2.60.3

docker network ls -> list the created networks

**04-09-2023 - monday**

**Creating local repo**

**docker images**

**docker search registry**

**docker pull registry:2**

**Default port number :5000**

**docker run -itd -p 9000:5000 registry:2**

curl -i <http://localhost:9000/v2/>

docker tag container\_id [localhost:9000/](http://localhost:9000/v2/)jenkins:jenkins\_version2

Eg: docker tag 0030ba3d620c [localhost:9000/](http://localhost:9000/v2/)jenkins:jenkins\_version2

Note 👍

Creating docker image , after create the container using below command

docker run -itd -p 5050:8080 --name mycontainer app:v1 /bin/bash -> changing port number after creating image

5050: host port number

8080: container port number

-p : port binding

Any one who is pinging 5050 , then traffic will routed to tomcat port 8080 which is reserved for ip address belongs to class B ip range

**29-08-20223**

docker rmi -f container-id -> remove the container

docker pause container-id -> pause the container

docker unpause container-id ->

docker start container-id

docker tag image\_id username/reponame:myimage

docker login -> first time need to give username and password

docker push username/reponamemy:image

Homework : go to dockerhub pull the image named as myimage have to pull it and make it an container

**31-08-2023**

**Docker network**

| **Docker docker docker ->vnic card** |
| --- |
| **Docker demon** |
| **Guest os** |
| **Hypervisor -os level virtualization** |
| **Hardware -> pnic card** |

Each docker need network for communication

Eg: default bridge network

PNIC card :host portno:contaier privateip :container port no

Any container wants to connect to the network will connect through bridge network

Vnic card : each vnic card should be connect to one pnic card

We conot change the port number , but we can change the host

By using port forwarding and port binding

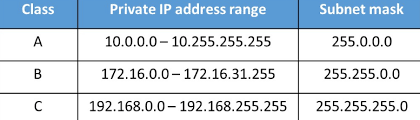
| Bridge network |
| --- |

| Port forwarding | Port binding |
| --- | --- |
| Docker proxy -> will define port no | User can define port no = 1 to 65535 |

Note : user cannot define port number of port forwarding , only docker proxy play of it

Docker will always class B private IP

If u want to connect port number bind container private ip to the container



root@jenkins:/# docker network ls

NETWORK ID NAME DRIVER SCOPE

1426b046cd4e bridge bridge local

e86d8a57ca87 host host local

af2c118bc56b none null local

Bridge network

These are the network to bind two or more containers

Makes the connectivity two containers

None network

Kind of isolated or private container

-p its represent port binding

-P represent port forwarding

**28-08-2023**

FROM = to choose an container based os

Maintainer- Author Name

RUN- while build the image what are the package want to

install with an image

ADD - any url package to download

copy - copy file b/w source and destination to ensure source and destination

should be on same path

Expose - define the port number for an image

Entrypoint -- during the rnning state of container if we want

to append any version

cmd -- during running state of container if we want o

replace any package version

label - tagging the name of the image

ENV --store credentials

**30-08-2023**

#create a directory

mkdir -m 777 app71

#get into directory

cd app71

#copy the war file jobs workspace to docker file location

cp /var/lib/jenkins/workspace/package/target/addressbook.war .

#create a dockerfile

touch dockerfile

#get into inside the Dockerfile echo will pass value in docker file

echo "

#choose container based images

FROM ubuntu:latest

#update the images

RUN apt-get update -y

#install java

RUN apt install openjdk-11-jdk -y

RUN apt-get install -y && apt-get install vim -y && apt-get install nano -y

RUN apt-get update -y

#download tomcat

Add https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.80/bin/apache-tomcat-9.0.80.tar.gz /opt

#Extract tomcat

RUN tar -zxvf /opt/apache-tomcat-9.0.80.tar.gz

#deploy war on tomcat

COPY addressbook.war /apache-tomcat-9.0.80/webapps/

#create port number

EXPOSE 8080" >> dockerfile

#build image this because we have changed the visudo

sudo docker build -t app:v1 .

#create a directory

#mkdir -m 777 mydockerfile5

#get into directory

cd mydockerfile5

#copy the war file jobs workspace to docker file location

cp /var/lib/jenkins/workspace/Package/target/addressbook.war .

#create a dockerfile

touch dockerfile

#get into inside the Dockerfile echo will pass value in docker file

sudo cat > dockerfile << EOF

#choose container based images

FROM ubuntu:latest

#update the images

RUN apt-get update -y

RUN apt update -y

#install java

#RUN apt-get install openjdk-11-jdk -y

#RUN apt-get install default-jdk -y

RUN apt install openjdk-8-jdk -y

RUN apt-get install -y && apt-get install vim -y && apt-get install nano -y

RUN apt-get update -y

#download tomcat

Add https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.80/bin/apache-tomcat-9.0.80.tar.gz /opt

#Extract tomcat

RUN tar -zxvf /opt/apache-tomcat-9.0.80.tar.gz

#deploy war on tomcat

COPY addressbook.war /apache-tomcat-9.0.80/webapps/

#create port number

EXPOSE 8080

EOF

#build image this because we have changed the visudo

sudo docker build -t app:v1 .