Day 1

08/05/2023

Query retrieve join

CustomerName

OrderDate

ProductName

List<Object>

List<JoinWrapper>

Class JoinWrapper {

CustomerName

OrderDate

ProductName

}

CName orderDate ProductName

Steven 08/05/2023 TV

Class JoinWrapper {

Constructor(CustomerName:string

OrderDate:Date

ProductName:String ){}

}

VM ware software :

Virtual Machine

Base Machine with 16 RAM

1TB hard disk

Linux or Unix OS.

4GB

50GB

10OS

Docker : Docker is an advanced OS virtualization software platform

That makes it easier to create, deploy and run the application in Docker container.

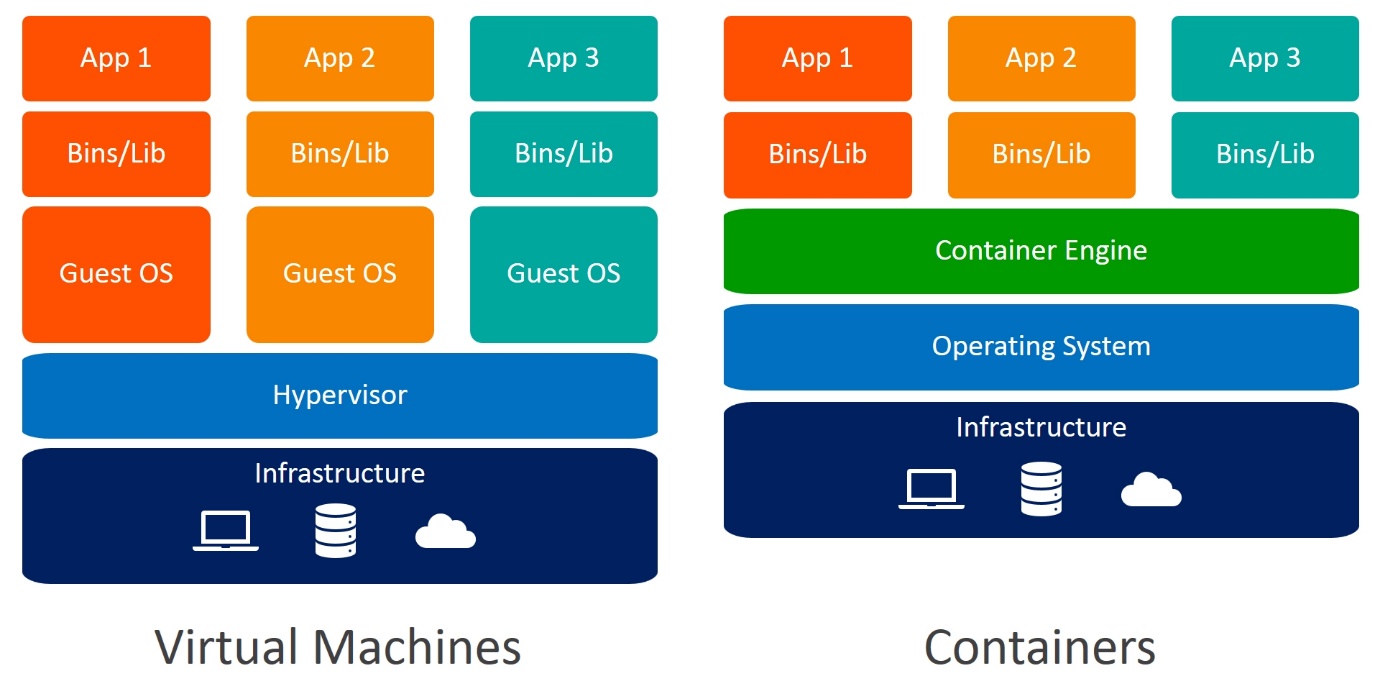
Virtualization lets you divide a system into a series of separate section, each one acting as distinct individual system or machine.

Using Docker we can create containerization application.

Virtualization is an abstract version of physical machine.

Containerization is an abstract version of an application or tool or software.

Docker container is responsible to run the application. Docker container is a part of docker engine.



@OneToMany(cascade=CascadeType.ALL)

List<Student> listofStd;

docker --version it is use to check the version of docker

docker images : it is use to check all images details present in local machine.

docker pull imagename this command is use to pull the image

docker pull hello-world

docker run imagename/imageId this command is use to run the image

docker hub account :

Docker hub is a like a git hub which help to publish our own images as well as

We can pull pre defined and user defined images in local machine.

In git hub we can push any type of data in Docker hub we need to publish or push

Docker images.

Docker Container : this is a running process or / instance of a images.

Docker Images : This file system and configuration of our application which are used to create a container.

Or

Docker images are the source code for our container.

Once we run the Docker images container become up and Container is responsible to execute the specific application mention in Docker file.

Docker file A docker file is a blue print or set of instruction that defines how our images is built.

.java

Docker file to display echo message through busybox images

Dockerfile

FROM busybox

CMD ["echo","Welcome to Docker Training"]

docker build -t imageName . -f Dockerfile

docker build -t cbfsd-busybox . -f Dockerfile

docker images

docker run cbfsd-busybox

Docker images to run simple java program

Demo.java

public class Demo {

    public static void main(String[] args) {

        System.out.println("Welcome to Java through Docker");

    }

}

Dockerfile

FROM openjdk:11

COPY Demo.java .

RUN javac Demo.java

CMD ["java","Demo"]

docker build -t cbfsd-java . -f Dockerfile

Creating docker image to run spring boot project

First create spring boot project and then

Create more than one rest api

Then using mvc package command create jar or war file.

Then create image with the help of

**Dockerfile**

FROM openjdk:11

COPY ./target/spring-boot-with-docker-0.0.1-SNAPSHOT.jar .

CMD ["java","-jar","spring-boot-with-docker-0.0.1-SNAPSHOT.jar"]

Command to create the images

**docker build -t cbfsd-springboot . -f Dockerfile**

To run the image if image is responsible to execute or run web app

**docker run -d -p 8080:8080 imageName/imageId**

docker run -d -p 8080:8080 cbfsd-springboot

docker run -d -p 8181:8080 --name my-spring-container cbfsd-springboot

docker ps This command is use to check all running container

docker ps -a This command display all container present in local machine (it may be running or stopped)

docker stop containerId/containerName : to suspend the running container

docker start containerId/containerName : to resumes the container

docker rm containerId –f

docker rmi imageName/imageId

creating the images for static web page

index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        div{background-color: burlywood;}

    </style>

</head>

<body>

    <div>

        Welcome to simple web page running throug Docker

    </div>

</body>

</html>

Dockerfile

FROM nginx

COPY index.html /usr/share/nginx/html

**docker build -t cbfsd-web-app . -f Dockerfile**

by default nginx server run on default port number 80.

Day 2

08/06/2023

docker system prune -a : this command is use to remove all stopped container,

images, network and cache memory.

Creating the image to run the angular application

So create the angular project

ng build

Now create dockerFile

FROM nginx

COPY ./dist/angular-with-docker/ /usr/share/nginx/html

docker build -t cbfsd-angular . -f dockerFile : creating image for angular project

docker images : checking images

docker run -d -p 80:80 cbfsd-angular : run the angular image

docker ps : checking running container

<http://locahost:80> : run the application on browser.

docker login This command is use to connect

local machine terminal with Docker hub account.

Before push this image into docker hub account we need to create the tag for that image

Tag is like a identity or version for that image.

docker tag imageName dockerhubaccount/imageName:versionNumber

docker tag cbfsd-angular akashkale/cbfsd-angular:1.0

After created tag for that image now you can push this image in docker hub account

docker push dockerhubaccount/imageName:version

docker push akashkale/cbfsd-angular:1.0

docker pull akashkale/cbfsd-angular:1.0

docker run -d -p 84:80 akashkale/cbfsd-angular:1.0

Spring boot container and mysql container

angular-app imaegname

angular-container which run with nginx server on port number 80

198.1.67.89:80 if we run angular on ec2

http://localhost:8080/product/findAll

<http://localhost:8080/product/store>

168.76.34.76:8080

mysql:8.0 imageName

springboot-container mysql-container container name

spring-boot-mysql-network

Spring boot MySQL

Different OS Different OS

Docker network : Docker network is a collection of a more than one container running together to communicate to each others.

1. If we want to run both container through command prompt we need to create the network and run both the container.
2. Docker compose : Docker compose is tool which provide us yml file. Inside this yml file we can write all container details and with the help of docker compose command we can run more than one container.

.yml

.yaml

docker network ls it will display all network present in local machine.

docker network create spring-boot-mysql-network This command is use to create the network

docker network ls

Now we will pull mysql image from docker hub account

docker pull mysql:8.0

docker run --name mysql-container --network spring-boot-mysql-network -d -p 3307:3306 -e MYSQL\_ROOT\_PASSWORD=root mysql:8.0

--name : container name

--network : network name

-d : detached mode

-p : published port number

Now open mysql container terminal

docker exec -it mysql-container bash

to connect mysql container

mysql -u root -p

password : root

mvn compile

mvn test

mvn package (this command is use to create jar or war file)

please remove testing starter from pom.xml file as well as sample testing file

in application.properties file we need to provide mysql container details

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://mysql-container/mydb

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

we will create the jar file using mvn package

Now we will create the image Dockerfile

FROM openjdk:11

COPY ./target/spring-boot-0.0.1-SNAPSHOT.jar .

CMD ["java","-java","spring-boot-0.0.1-SNAPSHOT.jar"]

docker build -t cbfsd-spring-boot . -f Dockerfile

docker run –-name springboot-container –-network spring-boot-mysql-network -d -p 8080:8080 cbfsd-spring-boot

now create the angular project

ng new angular-product-app

ng g c product this is component

ng g class product this is model class

ng g s product this is service class

develop the application then build the project

Dockerfile

FROM nginx

COPY ./dist/angular-product-app/ /usr/share/nginx/html

Then create the image

docker build -t angular-app . -f Dockerfile

docker run --name angular-container -d -p 88:80 angular-app

Docker compose

Docker compose is a tool kit which provided set of commands which help to run more than one container as a single service. Each container can execute independently or they can communicate with each others.

Docker compose use one of the scripting language ie yaml. (yet another markup language) which is base upon xml language