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.

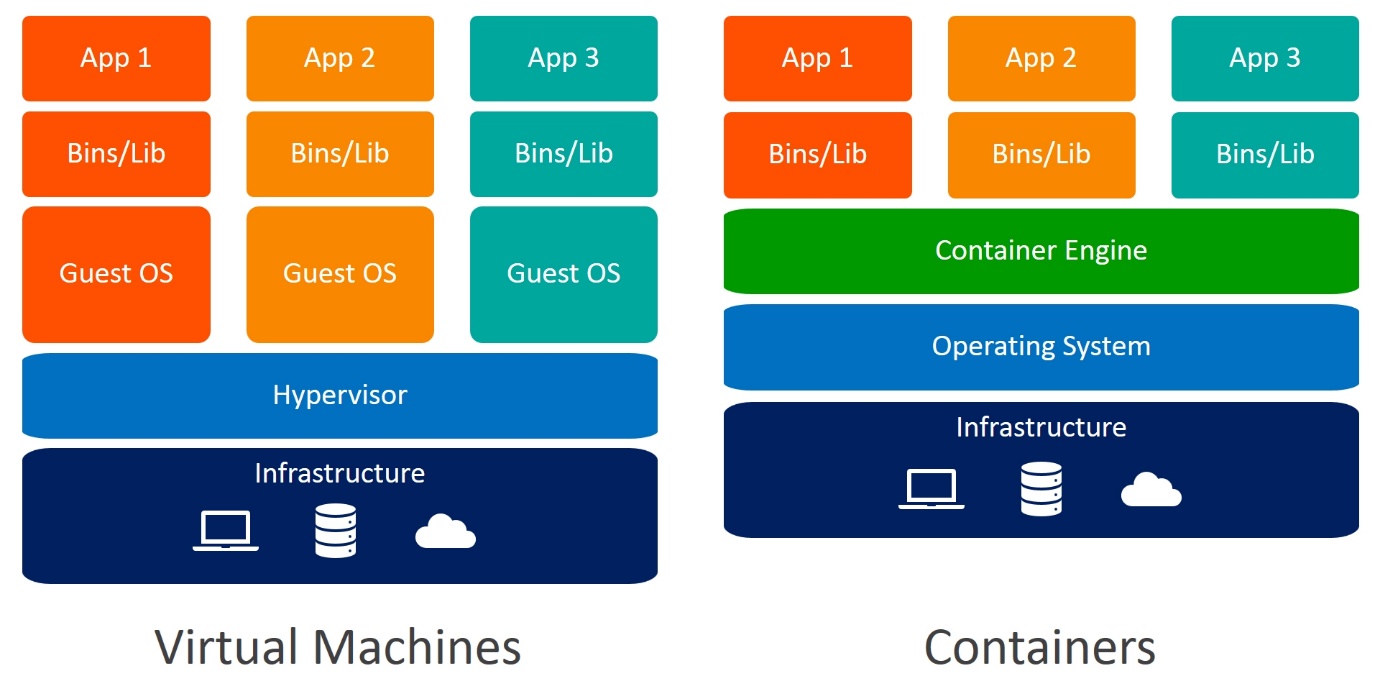
Virtualization lets you divide a system into a series of separate section, each one acting as distinct individual system or machine.

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

Day 3

08/12/2023

docker-compose build To build more than one image present in

docker-compose file

docker-compose up start all container containers

docker-compose up -d : run all those container in detached mode

docker-compose down : stop all containers

after up

please open browser and hit URL as <http://localhost:80>

80 : angular

8080: spring boot

3307 : mysql container are running

CI And CD tools

Manager

Organization repository

I shared project structure

Day 4

08/13/2023

git branch : This command is use to display all the branches present in local

machine.

default branch name is master/main

branch : it is like a pointer which hold more than one commit details.

git branch branchName

git branch A

to switch from one branch to another branch

git checkout branchname

Baron clone Employee.java create baron

Main/master

Neha clone Java\_programs Demo.java neha

Master/main Customer.java

Medic clone

Master/main Login.java medic

git branch neha

git checkout neha

Then please write the code

git add .

git commit -m “file created”

git push -u origin HEAD

HEAD always represent to last commit.

git clone URL

or

git init

git add .

git commit -m “done”

git remote add origin URL

git push -u origin HEAD

git checkout main

Demo.java

git pull

git branch -D baron (clean up activity)

git branch -D neha

git

Demo.java

Employee.java

Customer.java

Manager.java

Login.java

Demo.java Baron 2 line code int a,

String name;

Neha

float age;

Johson

int a;

Medic

String name;

git checkout main

git branch -D

Day 5

08/19/2023

Jenkin Pipe line

Jenkin pipeline is a collection of events or jobs which are interconnected with each others.

Simple Jenkin Pipe Line Job

pipeline {

agent any

stages {

stage('Verify all version of software') {

steps {

echo 'Verification of all necesary software'

sh 'java --version'

}

}

stage('Compile all the files') {

steps {

echo "Compile the project"

}

}

stage('Build the project') {

steps {

echo 'Project going to build it'

}

}

stage('Test the code') {

steps {

echo 'Testing the code and generat the test report'

}

}

}

}

Cloud Computing

Cloud always refer to network.

Rather than installing any software or tools or database. Those tools or software provided by cloud

We need to access through command prompt or gui base.

Cloud computing mainly divided into two types of models

1. Deployment model

Deployment model defines the type of access to the cloud.

It mainly divided into 4 types

1. Public cloud : the public cloud allow all resource by general public with

Login details. AWS and Azure etc

1. Private cloud : it is type of cloud maintain by organization. These all modules can be access with their private network.
2. Hybrid cloud : it is a combination of public and private cloud.
3. Community cloud :open source technologies use community cloud.

Python or javascript library.

1. Service model

This type of model define what type of service they are providing

1. IaaS : Infrastructure as a service : it provides to access to fundamental resoures

Such as physical machine, virtual machine, storages, database etc.

1. PaaS : Platform as a service : it provides the run time environment for application, development and deployments. :AWS and Azure etc.
2. SaaS : Software as a service : this cloud provide application base upon our requirement.

Sales force , Share point ,

Aws and Azure , google , oracle etc.

A to z services.

AWS S3 :

Simple Storage Service : S3 it a type of service provided by aws which help to share any type of data

Text file, doc file, folder, jar, war or ear file etc.

S3 Bucket : it is a like a container which help to share the data of any types.

First download the jar file

Then open the command prompt in that location where jar file present

java --version

java -jar filename.jar

then open the browser and hit the URL as

<http://localhost:8080>

AWS

EC2 instance Elastic Compute Cloud

It is use to create virtual server machine. Using this module we can create virtual server machine

While creating server machine we can chose OS ie window, linux or mac.

The machine configuration ie RAM and memory.

We can start that machine

Using SSH or putty client we can connect that machine.

And base upon our requirement we can install the required software to deploy the project.

Once we can any application in that instance that instance provide us unique IP address.

So we can access that application using Id address like

<http://187.1.45.89:8080>

<https://akash121234543456.s3.amazonaws.com/spring-boot-with-aws-0.0.1-SNAPSHOT.jar>

Day 6

08/20/2023

To install Java software in EC2 instance

sudo yum install java (it install current version java ie 17)

to download jar file from S3 bucket to Ec2

wget URL

<http://localhost:8080/URL>

if we run any spring boot or angular project on EC2 instance we can view that

running application using ip address provided by EC2 instance.

<http://54.84.72.27:8080>

by default Ec2 instance open only 22 port number which help to connect EC2 instance

through SSH client.

SSH client internally use 22 port number to connect any application.

If we run our spring boot or angular or any other application we need to open that application port number.

Using Java We can connect any database

MySQL

Oracle

Db2

Mongo db

1st option

AWS provide as RDS module which provided different database

In RDS we can create database ie mysql or oracle or db2

Then in spring boot application in place of localhost:3306

We need to use IP Address or domain name provided by AWS to access that database.

Or

We can use other cloud provider databases etc

2nd option

We are planning to deploy this application using docker.

So docker provide mysql image to store the data.

So using docker compose we can run three container ie

Angular

Spring boot

Mysql database

Deploy angular application in EC2 instance.

If we are planning to deploy angular project in AWS ec2 instance.

1. Create angular project in local machine
2. After development using ng build. Build the project
3. Then using docker create the image
4. And then publish this image in our docker hub account.
5. Create EC2 instance
6. Then in EC2 instance using few command please install docker
7. Then pull this image in EC2 instance from docker hub
8. Then run this image EC2.
9. If we run that application we can access using IP Address provided by EC2 instance.

First create the angular project

ng new angular-aws

routing 🡪 yes

styling 🡪 css

once project created using cd command move inside an angular project

then develop the application.

Then check application running or not using ng serve

To build the project we need to run ng build command

Once build file is ready now we need to create docker image using build file.

**Dockerfile**

FROM nginx

COPY ./dist/angular-aws/ /usr/share/nginx/html

Now we need to create docker image

Below command is use to create the image

docker build -t angular-fsd-aws . -f Dockerfile

docker image please verify image created or not.

Now we need to push this image in docker hub account.

Before push we need to create tag for that image.

docker tag angular-fsd-aws akashkale/angular-fsd-aws:1.0

now we can push this image in docker hub account

docker push akashkale/angular-fsd-aws:1.0

Now we need to install the docker in EC2 instance

sudo yum install docker

sudo service docker start

sudo docker info

sudo docker –version

sudo docker images

sudo docker pull akashkale/angular-fsd-aws:1.0 it is use to pull the image

sudo docker images this command to verify image present or not

sudo docker run -d -p 80:80 imageName/imageId this command is use to run the image

docker-compose is a tool kit which help to run more than one container.

Using Docker compose we already run angular --🡪 spring boot 🡪 mysql container.

Now we install docker-compose tool kit in EC2 instance using below commands.

install docker compose

Download the current stable release of Docker Compose

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

Apply executable permissions to the binary

sudo chmod +x /usr/local/bin/docker-compose

sudo reboot

Verify it’s installed

docker-compose version

Docker-compose file

version: '3.8'

services:

mysql-container:

image: mysql:8

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: mydb

ports:

- 3306:3306

restart: always

springboot-container:

build: ./backend/spring-boot/spring-boot/

depends\_on:

- mysql-container

ports:

- 8181:8080

restart: always

angular-container:

build: ./frontend/angular-product-app/

ports:

- 81:80

MySQL port number : 3306

Angular port number 81

Spring boot port number 8181

Angular service calling spring boot application using ip address not localhost.

Please install the git in EC2 instance using command as

**sudo yum install git -y**

after git instal in EC2 instance please clone sample project with backend and frontend folder with docker compose file

git clone <https://github.com/Kaleakash/docker-compose-repository.git>

once you clone please

move inside repository folder which you downloaded using docker clone command.

sudo docker-compose build this command is use to build the angular and spring images

sudo doker-composer up -d : this command use to start angular, spring boot and mysql container in background

to verify all container running or not we need to run the command as

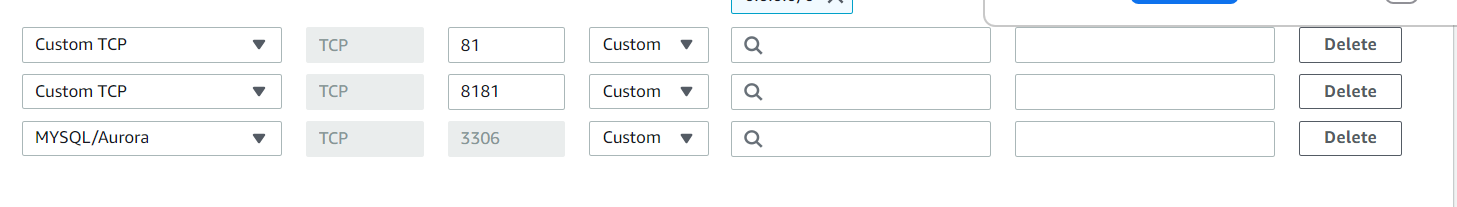
sudo docker ps

now in security group wizard we need to open three port number

3306 for mysql

8181 for spring boot

81 for angular



8/26/23

We will create EC2 instance in AWS

Then we will install git, docker, Jenkin

In Jenkin We will create Jenkin pipe line job

That Jenkin job pull the project from git

<https://github.com/Kaleakash/docker-compose-repository.git>

and they build the project with trigger.

If we Run Jenkin job in EC2 instance once the project run using docker-compose

We can access that application with help of IP Address provided by EC2 instance.

In future if we need to do any changes in angular or spring boot we will do in local machine

And push this code to git. And Jenkin pull the code from git and run once again

Docker-compose

In same example we will use RDS aws database.

Jenkin default port number 8080