* CB FSD - Integration and Deployment

Day 3 : 12 July 2024

If we want to run more than one container and those container running independently or interact with each other to share the data

Angular container Spring boot container mysql container

OS Os Os

http protocol TCP/IP

docker compose : docker compose provide toolkit which help to run more than one container and those container work independently as well as they can interact with each others. To write these all container and image information we need to use docker-compose.yml or docker-compose.yaml. yaml (yet another markup language).

Inside this file we write all image and container information details.

Below command remove all images, all stopped container, cache and network.

docker system prune -a

if docker compose command not present please run below commands.

Download the current stable release of Docker Compose

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

Give the permission

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

docker-compose version

**docker-compose.yml**

version: '3.6'

services:

  hello-service:

    image: hello-world

  ng-service:

    image: nginx:latest

    ports:

      - 80:80

  akash-service:

    image: akashkale/my-angular-test:a2

    ports:

      - 81:80

Then open the command prompt

docker-compose up

open another terminal and check all images and running container

docker images

docker ps

docker-compose up -d created and run in background

docker-compose down remove all container

docker-compose start start all container

docker-compose stop stop all container

creating docker-compose file to run three container ie angular, spring boot and mysql

docker-compose.yml only run mysql container

version: '3.6'

services:

  mysql-service:

    image: mysql:8.0

    container\_name: mysql\_container

    environment:

      - MYSQL\_ROOT\_PASSWORD=root

      - MYSQL\_DATABASE=capstonedb

    ports:

      - 3307:3306

Open the command prompt then run the command as

docker-compose up

once service up

open another terminal and check all images and container status

docker images

docker ps

to open mysql terminal

docker exec -it mysql\_container bash

then connect mysql database

mysql -u root -p

🡪root

Then run the command as show databases.

As of now please disable eureka client and testing starter from pom.xml file and disable related classes as well as annotation.

Do the changes in application.properties file

spring.application.name=Login\_app\_micro\_service

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

spring.datasource.url=jdbc:mysql://mysql-service/capstonedb

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

server.port=8181

now we update docker-compose.yml file to run two containers.

docker-compose.yml file

version: '3.6'

services:

  mysql-service:

    image: mysql:8.0

    container\_name: mysql\_container

    environment:

      - MYSQL\_ROOT\_PASSWORD=root

      - MYSQL\_DATABASE=capstonedb

    ports:

      - 3307:3306

    restart: always

  spring-boot-service:

    #image: imageName

    build: ./backend/Login-app-micro-service/Login-app-micro-service/

    container\_name: spring\_boot\_container

    ports:

      - 8181:8181

    depends\_on:

      - mysql-service

    restart: always

then using docker-compose please start the container

docker-compose up

wait to up both the containers

then verify

docker images

docker ps

to check table and record present inside table please connect to mysql container.

docker exec -it mysql\_container bash

mysql -u root -p

password 🡪 root

use capstonedb

select \* from login;