* CB FSD - Integration and Deployment

Day 5 : 18 July 2024

docker-compose.yml

version: '3.8'

services:

  mysql-service:

    image: mysql:8.0

    container\_name: mysql\_container

    environment:

      - MYSQL\_ROOT\_PASSWORD=root

      - MYSQL\_DATABASE=capstonedb

    ports:

      - 3307:3306

    networks:

      - private

    volumes:

      - db:/var/lib/mysql

    restart: always

  eureka-server:

    image: akashkale/eureka-server:c2

    #build: ./backend/Eureka-Server/Eureka-Server/

    container\_name: eureka\_container

    ports:

      - 8761:8761

    networks:

      - public

    restart: always

  login-micro-service:

    image: akashkale/login-micro-service-app:c2

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

    container\_name: login\_micro\_service\_container

    environment:

      - EUREKA\_CLIENT\_SERVICEURL\_DEFAULTZONE=http://eureka-server:8761/eureka/

    ports:

      - 8181:8181

    depends\_on:

      - mysql-service

      - eureka-server

    networks:

      - private

      - public

    restart: always

  angular-service:

    image: akashkale/angular-framework-app:c2

    #build: ./frontend/shopping-frontend-app/

    container\_name: angular\_container

    ports:

      - 80:80

    networks:

      - public

networks:

  private :

    name : spring-boot-mysql-network

  public:

    name : angular-and-spring-boot-network

volumes:

  db:

   driver: local

docker-compose up --build

test you application running on port number

<http://localhost:80>

DevOps

Git, Maven or Gradle, Docker, CI/CD, Kubernetes etc.

CI and CD : Continuous Integration and Continuous Deployment or delivery

Dev1 pull

pull

Dev2 shared repository <----🡪 CI/CD -🡪 Team

Pull

Dev3

Push

Manager

Manager pull the code and integrate and re-build it (compile program, run program, test program, creating jar or war file)

CD and CD tool link with any shared repository. Whenever any developer or team push the code then CI/CD tool pull the code and re-build the application in their environment. If any error then it can send the notification to respective team else it pass this code to another team like testing server, or production server or any team.

CI and CD tools

1. Jenkin
2. Gitlab
3. Github with actions
4. Teamcity

Jenkin : Jenkin is a type of CI and CD tools. Which is also known as continuous automation server or tool. This tool is base upon java technologies. It is an open source and plugin base CI and CD tool.

Running Jenkin

1. Install software in your machine ie window or non window
2. Using war file (java software present)
3. Using Docker we can run Jenkin Image.

Running the Jenkin using Docker

docker run -p 8181:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:lts-jdk17

after up please open browser and hit the URL as

<http://localhost:8181>

it ask you password . please check password in console.

Installed suggested plugin

After downloaded plugin create the account and login to Jenkin dashboard

Jenkins Jobs

1. Creating Simple Jenkin job to display the message.
2. Creating Jenkin job to pull the project from github account and build the project (we compile and run the program).

If we want to compile java program using terminal

javac Demo.java compile the program

java Demo run the program

1. Creating another job responsible to build the project again and again using trigger options.
2. Creating job to build the project whenever any developer push the code in git hub account.

Trigger with poll SCM : whenever any changes happen in repository base upon time then only build it.