Docker commands

docker image ls is same as docker images

docker container ls is same as docker ps

docker pa -a to view all containers (including hidden)

these commands were used to list the images and containers.

Now, how to remove an image?

docker rmi <<image id>>

How to remove a container?

docker rm <<container id>>

docker pull

docker push

docker run

docker tag

docker build

docker kill

docker stop

docker exec

docker login

docker network ls

docker network create

You should be able to

Create a docker image of spring boot restful webservice

After creating build and test an application or service, you should be able to create a docker image of that project.

ECS

We deploy and run docker images

Elastic Container Service allows running our docker images in AWS

EKS Elastic Kubernetes Service

-----------------------

Kubernetes

Activity

Understand what is Kubernetes. What is the need for Kubernetes

Demo:

Create a rest api and deploy it in hub.docker.com

18-jul-employee

Migrate from h2 to mysql:

1. In pom.xml, comment or remove h2 dependency
2. In pom.xml, add the mysql dependency
3. In application.properties

Change the datasource and dialect configurations to mysql

We have changed the project configuration to mysql and it is working fine locally. Table created in mysql by jpa

Now, we created a docker image of this “18-jul-employee-mysql” and when we run this docker image, we get error: Caused by: com.mysql.cj.jdbc.exceptions.CommunicationsException: Communications link failure

This is because, docker image runs in a separate virtual machine and cannot access localhost mysql.

So we need to run our docker image and another mysql docker image together using: either

Docker compose (or)

Docker network

Steps to run using docker network:

1. Create a docker network

docker network create employee-mysql

1. Check if network is create:

Docker network ls

1. Run mysql image:

docker container run --name mysqldb --network employee-mysql -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=ust2 -d mysql:8

1. In our application.properties: we need to change localhost to mysqldb (container name)

And also the root password if different

1. Build image by skipping test

mvn spring-boot:build-image -Dmaven.test.skip=true

or in right click project run as maven build tick that skip test option

1. Run this image in the same network

docker container run --network employee-mysql --name emp-service -p 8080:8080 -d 18-jul-employee-h2:0.0.1-SNAPSHOT

1. Check logs

docker container logs -f emp-service

Before lunch, we run our spring boot app docker image along with mysql docker image using same docker network.

Now, we do the same using docker compose.

YAML?

Yet Another Markup Language

1. application.properties

server.port=5000

spring.datasource.url=jdbc:mysql://mysql\_db:3306/ust2

spring.datasource.username=root

spring.datasource.password=root

spring.datasource.hikari.connection-timeout=20000

spring.datasource.hikari.minimum-idle=5

spring.datasource.hikari.maximum-pool-size=12

spring.datasource.hikari.idle-timeout=300000

spring.datasource.hikari.max-lifetime=1200000

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect

spring.jpa.properties.hibernate.id.new\_generator\_mappings = false

spring.jpa.properties.hibernate.format\_sql = false

spring.jpa.hibernate.ddl-auto=update

spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

logging.level.org.hibernate.SQL=DEBUG

logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE

spring.datasource.initialization-mode=always

1. Dockerfile

FROM openjdk:8

VOLUME /main-app

ADD target/22-jul-employee-mysql-dc-0.0.1-SNAPSHOT.jar app.jar

EXPOSE 5000

ENTRYPOINT ["java", "-jar","/app.jar"]

1. docker-compose.yml

version: "3.7"

services:

api\_service:

build: .

restart: always

ports:

- 5000:5000

depends\_on:

- mysql\_db

command: sh -c './wait-for mysql\_db:3306 -- npm start'

mysql\_db:

image: "mysql:8.0"

restart: always

ports:

- 3306:3306

environment:

MYSQL\_DATABASE: ust2

MYSQL\_ROOT\_PASSWORD: root

1. docker-compose build
2. docker-compose up