**Dockerization of zissa**

**Docker :** Docker is a containerization platform that packages your application and all its dependencies together in the form of Containers to ensure that your application works seamlessly in any environment.

**Dockerhub:**

Docker Hub is like GitHub for Docker Images.

Is a collection images with required environment setup.

you can develop your own image and upload on Docker Hub

## **Dockerfile, Docker Image And Docker Container:**

1. A Docker Image is created by the sequence of commands written in a file called as Dockerfile.
2. When this Dockerfile is executed using a docker command it results into a Docker Image with a name.
3. When this Image is executed by “docker run” command it will by itself start whatever application or service it must start on its execution.

## **Docker Compose:**

Docker Compose is basically used to run multiple Docker Containers as a single server.

**VMS vs Docker-containers**

1.size

2.startup

3.Integartion

Folder-structure :

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

---- zissa.sql and DBDockerfile-mysql

--- zissa-backend/---

---JAR, BEDockerfile and application.prperties (change ip of mysql connection with ip of mysql container in docker-compose file)

|

docker-zissa/------

(docker-compose.yml)

|

------zissa-frontend/---code, FEDockerfile

1.change the ip and port in appSettings.js file with ip and port of FE running ip:port of docker machine)

2.change the ip and port in angular.json file serve module with ip and port of FE container in docker-compose file

"serve": {

"builder": "@angular-devkit/build-angular:dev-server",

"options": {

"browserTarget": "zissa-frontend:build",

"port": 7070,

"host": "13.5.0.10"

},

docker-compose.yml

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

services:

app:

build:

context: zissa-backend

dockerfile: BEDockerfile

container\_name: BE

ports:

- "2020:2020”

networks:

dbn:

ipv4\_address: 13.5.0.14

depends\_on:

- db

db:

build:

context: zissa-backend

dockerfile: DBDockerfile

container\_name: DB

environment:

MYSQL\_ROOT\_PASSWORD: "root"

restart: always

ports:

- "3306:3306"

networks:

dbn:

ipv4\_address: 13.5.0.12

fe:

build:

context: zissa-front

dockerfile: FEDockerfile

container\_name: FE

ports:

- "7070:7070"

networks:

dbn:

ipv4\_address: 13.5.0.10

networks:

dbn:

driver: bridge

ipam:

config:

- subnet: 13.5.0.0/16

version: "2"

FEDockerfile

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

FROM node:8.12.0

COPY . /app/

#COPY appSettings.ts /app/src/app/pages/shared/appSettings.ts

WORKDIR /app/

RUN npm install

CMD [ "npm", "start" ]

BEDockerfile

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

FROM openjdk:10

RUN apt-get update

RUN mkdir -p /usr/local/app/bin/

COPY ZISSA-0.0.1-SNAPSHOT.jar /usr/local/app/bin/

COPY application.properties /usr/local/app/bin/

WORKDIR /usr/local/app/bin/

CMD ["java", "-Dspring.config.location=application.properties", "-jar", "ZISSA-0.0.1-SNAPSHOT.jar"]

DBDockerfile

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

FROM mysql:5.7

# Environment variables

ENV MYSQL\_ROOT\_PASSWORD=root

ENV MYSQL\_DATABASE=zissa

ENV MYSQL\_USER=zissa

ENV MYSQL\_PASSWORD=zissa

#RUN echo "lower\_case\_table\_names=1" >> /etc/mysql/mysql.conf.d/mysqld.cnf

#COPY zissa-backend/zissa.sql /docker-entrypoint-initdb.d/

EXPOSE 3306

**Commands**

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

Note: The 1&2&3 commands run from zissa folder as above

1. docker-compose build -----------> to build docker images

2. docker-compose up -d ------------> to run the docker builded images

3 docker-cmpose down --------------> to stop and remove the running containers

4. docker ps -a -------------> to see which containers are up and running

5. docker logs FE(container name) -------------> to see the logs of containers

6. docker exec -it FE(containername) bash ------> to enter into the container

**Docker Installation for LInux ubuntu**

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

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

**Docker Installation for Microsoft Windows**

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

<https://docs.docker.com/docker-for-windows/install/>

**To use docker-compose commands need to install docker-compose**

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

<https://docs.docker.com/compose/install/> -----> there are separate steps for linux and windows