**Hosting Angular-10-Spring-Boot-CRUD-Full-Stack-App on localhost**

1. **Installing all required softwares on your localhost**

sudo apt update -y && sudo apt upgrade -y

sudo curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash && source ~/.bashrc (This will install nvm and restart the terminal)

nvm install 22 (Using nvm to install node-22)

sudo apt install -y git maven openjdk-17-jre-headless

Installing Docker

Ref link: https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-22-04

sudo apt install apt-transport-https ca-certificates curl software-properties-common

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update

apt-cache policy docker-ce (will get output for Docker repo instead of the default Ubuntu repo)

sudo apt install docker-ce

sudo systemctl status docker

sudo usermod -aG docker ${USER} (Adding all users to group)

su - ${USER} (To apply the membership)

1. **Hosting mysql in localhost using docker on host network**

docker run -dit --name mysql-crud --network host -e MYSQL\_ROOT\_PASSWORD=root mysql:latest

After the container has been deployed successfully, create a database with name “employee\_db”

docker exec -it mysql-crud mysql -u root -p

login password: root

create database employee\_db;

show databases;

Confirm the database you have just created.

1. **Hosting and building App-backend**

Cloned the “CRUD-Full-Stack-App” in your working dir

https://github.com/sohail-ops/CRUD-Full-Stack-App.git

Navigate to ~/CRUD-Full-Stack-App/springboot-backend/

Change values in .env file

nano .env

DB\_URL=jdbc:mysql://localhost:3306/employee\_db

DB\_USERNAME=root

DB\_PASSWORD=root

CORS\_ORIGIN=http://localhost

Build the application

mvn clean install

Run the application

java -jar target/springboot-backend-1.0.jar

Once the application is build and running without any error, build Docker image for the same.

Creating “Dockerfile” and “.dockerignore” in root folder of his application.

Once both files are created, run the following

docker build --no-cache -t app-be:1.0 .

Run the image

docker run -dit --name app-be --network host app-be:1.0

1. **Hosting and building App-frontend**

Navigate to ~/CRUD-Full-Stack-App/angular-frontend/src/environments/

Change values in env files

environment.prod.ts and environment.ts

apiIp: 'localhost:8080'

Installing application required modules

npm install

Test the application

npm start

Once the application is running without any error, build Docker image for the same.

Creating “Dockerfile” and “.dockerignore” in root folder of his application.

We will be using nginx image on docker to run the application, nginx requires to make configuration changes in default.conf file to communicate with the backend server.

Once all files are created, run the following

docker build --no-cache -t app-fe:1.0 .

Run the image

docker run -dit --name app-fe --network host app-fe:1.0

Verify the application is running on browser: http://localhost

Add, Update and delete the entries for employees to verify if data is been getting stored in db.