3 tier web hosting project using docker (spring boot, java, mysql)

**create ec2 instance**

Os - ubuntu

processor - T2.medium

allow - ssh, http, https,all tcp ipv4..anywhere

storage 20gb

**connect ssh**

using WinSCP app copy 2 project folder and paste on terminal

ls - file terminal var copy Zali ka check karnyasathi

mkdir project - project navach folder banavane

sudo mv (project file name) (jithe move karaychay te folder name)

java chya project file name la rename karne ‘Quantumsoft Technologies pvt ltd’ cha Quantumsoft

**install docker**

sudo apt-get update

sudo apt-get install docker.io -y        – docker architecture download karnyasathi

sudo systemctl status docker             – docker chi service chalu ahe ka te check karnyasathi ---ctrl+c=exit

sudo usermod -aG docker $USER      - docker group la add karnyasathi

newgrp docker                                        – group refresh karnyasathi

docker --version                                      – docker cha version check karnyasathi

sudo apt-get install docker-compose -y – docker compose file run honyasathi

docker login -u [username] [password]

sudo apt-get install docker-compose-plugin

sudo apt-get install docker-compose -y

sudo curl -SL <https://github.com/docker/compose/releases/download/v2.32.0/docker-compose-linux-x86_64> -o /usr/local/bin/docker-compose

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

docker-compose --version

go frontend project file

cd config

vim Config.js

change public ip in Config.js

go frontend project file

**vim default.conf – frontend chi docker file jithe ahe tithe create karane ………**

server {

  listen 80;

  server\_name [server public ip];

   # Serve Angular application

   root /usr/share/nginx/html;

   index index.html;

  location / {

      try\_files $uri $uri/ /index.html;

   }

   # Proxy API requests to Spring Boot service

  location /api/ {

      proxy\_pass http://backend:8080;

      proxy\_set\_header Host $host;

      proxy\_set\_header X-Real-IP $remote\_addr;

      proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

      proxy\_set\_header X-Forwarded-Proto $scheme;

   }

   # Error handling for Angular routes

  error\_page 404 /index.html;

   location = /index.html {

      allow all;

   }

}

…………………………………………………………………………………………………..

sudo apt-get install unzip -y

unzip [zipfile]

mv [unzippedfolder] dist

**vim Dockerfile - create Dockerfile for frontend**

FROM nginx:latest

COPY dist/ /usr/share/nginx/html

RUN chmod -R 755 /usr/share/nginx/html && chown -R nginx:nginx /usr/share/nginx/html

COPY default.conf /etc/nginx/conf.d/default.conf

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

…………………………………………………………………………………………………………………………

docker build -t spquantum/frontend:v . - image change zalyas compose madhe pn image name change karne

docker push spquantum/qfrontend:v1

……………………………………………………………………………………………………………….

cd ..

cd backend

ls

target file asel tar delete karne

pom file javal ya command fire karne

sudo apt install maven

java -version

sudo apt update

sudo apt install openjdk-17-jdk

sudo update-alternatives --config java

choose 1

ls

sudo apt install maven

mvn clean install -Dmaven.test.skip=true

go target folder

chmod +x QuantomSoft-0.0.1-SNAPSHOT.jar

cd ..

…………………………………………………………………………………..

**vim Dockerfile ….create dockerfile near target file**

# Use an official Java runtime as a parent image

FROM openjdk:17-jdk-slim

# Set the

WORKDIR /app

# Copy the jar file into the container

COPY target/QuantomSoft-0.0.1-SNAPSHOT.jar QuantomSoft-0.0.1-SNAPSHOT.jar

# Make the port available to the outside world

EXPOSE 8080

# Run the jar file

CMD ["java", "-jar", "QuantomSoft-0.0.1-SNAPSHOT.jar"]

…………………………………………………………………………………………………………………………………………..

docker build -t spquantum/backend:v1 .

docker push spquantum/backend:v1

go application.property file backend-src-main-resources

sudo nano application.properties

4*th* line local host cha jaagi mysqldb

……………………………………………………………..

sudo nano docker-compose.yml                      – docker compose file banavine project file madhe

version: '3.4'

services:

 backend:

   build:

     context: ./backend  # Path to your backend directory

   environment:

    SPRING\_DATASOURCE\_URL: jdbc:mysql://mysqldb:3306/quantumsoft

    SPRING\_DATASOURCE\_USERNAME: root

    SPRING\_DATASOURCE\_PASSWORD: Neha@123

   ports:

     - "8080:8080"

   depends\_on:

     mysqldb:

       condition: service\_healthy  # Wait for MySQL to be ready

   networks:

     - mynetwork

 frontend:

   build:

     context: ./frontend  # Path to your frontend directory

   ports:

     - "5502:80"

   depends\_on:

     - backend

   networks:

     - mynetwork

mysqldb:

   image: mysql:8.0

   environment:

    MYSQL\_ROOT\_PASSWORD: "Neha@123"

     MYSQL\_DATABASE: "quantumsoft"

     MYSQL\_USER: "root"

     MYSQL\_PASSWORD: "Neha@123"

   ports:

     - "3306:3306"

   volumes:

     - mysql-data:/var/lib/mysql

   networks:

     - mynetwork

   healthcheck:

     test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]

     interval: 10s

     retries: 5

     start\_period: 30s

networks:

 mynetwork:

volumes:

 mysql-data:

……………………………………………………………………………………………………………

docker-compose up -d – compose file build karnyasathi

docker-compose down – compose file madhe correction kelyavar

edit inbound rule-  all tcp-anywhere ipv4

copy paste pb ip :5502 in new tab

……………………………………………………………………………………………………………..

docker ps

docker exec -it (container id) bash – docker exec -it c2f0adff3d1d bash

mysql -u root -p –

show databases; -

use (database name); -

show tables; -

select \* from (table name) – database madhil content pahnyasathi