- sudo dnf update
- mkdir three-tier-app
- cd three-tier-app
- mkdir frontend
- mkdir backend
- sudo dnf install docker
- sudo usermod -aG docker ec2-user
- newgrp docker
- mkdir -p $DOCKER_CONFIG/cli-plugins
- curl -SL https://github.com/docker/compose/releases/download/v2.24.2/docker-compose-linux-x86_64 \ -o $DOCKER_CONFIG/cli-plugins/docker-compose
- docker pull mysql:8.4
- cd frontend
- vim nginx.conf –

```
events {}

http {
  server {
    listen 80;
    server_name _;

    #root /var/www/html;
    root /usr/share/nginx/html;

    index index.html;

    location / {
      try_files $uri $uri/ /index.html;
    }

    location /api/ {
      proxy_pass http://backend:8080/api/;
      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;
    }
  }
}
```

- Vim Dockerfile –
    1. FROM nginx:alpine
    2. COPY quantumsoft/ /usr/share/nginx/html
    3. RUN chmod -R 755 /usr/share/nginx/html
    4. COPY nginx.conf /etc/nginx/nginx.conf
    5. EXPOSE 80
    6. CMD ["nginx", "-g", "daemon off;"]

- Cd frontend > unzip ”……zip file” > mv “…..zipfile” quantumsoft
- cd quantumsoft > cd config > vim Config.js –
    1.
    2. const config = {
    3. development: "http://13.232.38.254:8080", // Backend URL for local testing
    4. production: "/api"
    5.
    6. };
    7.
    8. // Automatically choose the correct environment
    9. const apiBaseURL = window.location.hostname === "http://13.232.38.254:8080/api/";
    10.
    11. // Expose the API URL globally
    12. window.apiBaseURL = apiBaseURL;
- MYSQL –

    docker run -d \ --name quantumsoft-mysql \ -e MYSQL_ROOT_PASSWORD=root \ -e MYSQL_DATABASE=quantumsoft \ -e MYSQL_USER=root \ -e MYSQL_PASSWORD=root \ -p 3306:3306 \ -v mysql-data:/var/lib/mysql \ mysql:8.4
- docker ps
- docker exec -it quantumsoft-mysql mysql -uroot -p
- SHOW DATABASES;
- USE quantumsoft;
- Exit,exit
- cd three-tier-app
- vim docker-compose.yml –
    1. services:
    2. backend:
    3. image: quantumsoft-backend:latest

```
    # Relative path to the backend project directory
    environment:
      SPRING_DATASOURCE_URL: jdbc:mysql://mysql-service:3306/quantumsoft
      SPRING_DATASOURCE_USERNAME: root
      SPRING_DATASOURCE_PASSWORD: root
    ports:
      - "8080:8080"
    depends_on:
      - mysql-service
    networks:
      - mynetwork  # Use the custom network
  frontend:
    image: quantumsoft-frontend # Relative path to the frontend project directory
    ports:
      - "5502:80"   # Change from 80 to 8081 to avoid conflict with port 80
    depends_on:
      - backend
    networks:
      - mynetwork  # Use the custom network
  mysql-service:
    image: mysql:8.4
    environment:
      MYSQL_ROOT_PASSWORD: "root"
      MYSQL_DATABASE: "quantumsoft"  # Automatically creates the database
      MYSQL_PASSWORD: "root"
    ports:
      - "3306:3306"
    volumes:
      - mysql-data:/var/lib/mysql  # Persist MySQL data across container restarts
    networks:
      - mynetwork  # Use the custom network
networks:
  mynetwork:
volumes:
  mysql-data:
```

- cd backend –

1. FROM openjdk:23
2. WORKDIR /app
3. COPY quantumsoft.jar app.jar
4. EXPOSE 8080
5. ENTRYPOINT ["java", "-jar", "app.jar"]

- Docker build -t quantumsoft-frontend
- Docker build -t quantumsoft-backend
- Docker-compose up -d
- Docker ps
- Docker ps -a
- Docker logs “container id”