САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 4

Выполнил:

Литвак Игорь

Группа K33401

Проверил:
Добряков Д. И.

Санкт-Петербург

2022 г.

Задача

Необходимо упаковать ваше приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения.

Ход работы

Dockerfile основного сервиса:

```
# Build
FROM node:18-alpine as builder
ENV NPM_CONFIG_PREFIX=/home/node/.npm-global
ENV PATH=$PATH:/home/node/.npm-global/bin
RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
WORKDIR /home/node/app
COPY --chown=node:node package*.json ./
USER node
RUN npm config set unsafe-perm true
RUN npm install -g typescript
RUN npm install -g ts-node
RUN npm install
COPY --chown=node:node . .
RUN npm run build
```

```
# Install
FROM node:18-alpine
ENV NODE_ENV=production
RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
WORKDIR /home/node/app
COPY --chown=node:node package*.json ./
USER node
RUN npm install --production
COPY --from=builder /home/node/app/dist ./dist
COPY --chown=node:node .sequelizerc ./dist/
COPY --chown=node:node src/configs/settings.ini ./dist/configs/settings.ini
COPY --chown=node:node src/sequelize/index.js ./dist/sequelize/index.js
COPY --chown=node:node src/seeders/*.js ./dist/seeders/
COPY --chown=node:node docker-entrypoint.sh .
RUN chmod +x ./docker-entrypoint.sh

EXPOSE 5000
ENTRYPOINT ./docker-entrypoint.sh
```

Dockerfile сервиса авторизации:

```dockerfile
# Build
FROM node:18-alpine as builder
ENV NPM_CONFIG_PREFIX=/home/node/.npm-global
ENV PATH=$PATH:/home/node/.npm-global/bin
RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
WORKDIR /home/node/app
COPY --chown=node:node package*.json ./
USER node
RUN npm config set unsafe-perm true
RUN npm install -g typescript
RUN npm install -g ts-node
RUN npm install
COPY --chown=node:node . .
RUN npm run build

```

```dockerfile
# Install
FROM node:18-alpine
ENV NODE_ENV=production
RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
WORKDIR /home/node/app
COPY --chown=node:node package*.json ./
USER node
RUN npm install --production
COPY --from=builder /home/node/app/dist ./dist
COPY --chown=node:node  src/configs/settings.ini ./dist/configs/settings.ini

EXPOSE 5001
CMD [ "node", "dist/index.js" ]
```

docker-compose.yml:

```yaml
version: '3.9'
services:
  auth_db:
    image: mysql
    environment:
      - MYSQL_USER=mysql
      - MYSQL_PASSWORD=mysql
      - MYSQL_DATABASE=mysql
      - MYSQL_ROOT_PASSWORD=mysql
    expose:
      - 3306
    volumes:
      - volume_auth:/var/lib/mysql
    healthcheck:
      test: [ "CMD", "mysqladmin" ,"ping", "-h", "localhost" ]
      timeout: 20s
      retries: 10
  main_db:
    image: mysql
    environment:
      - MYSQL_USER=mysql
      - MYSQL_PASSWORD=mysql
      - MYSQL_DATABASE=mysql
      - MYSQL_ROOT_PASSWORD=mysql
    expose:
      - 3306
    volumes:
      - volume_main:/var/lib/mysql
    healthcheck:
      test: [ "CMD", "mysqladmin" ,"ping", "-h", "localhost" ]
      timeout: 20s
      retries: 10
  auth_service:
    build:
      context: auth_service
    depends_on:
      auth_db:
        condition: service_healthy

```

```
  main_service:
    build:
      context: main_service
    ports:
      - "5000:5000"
    depends_on:
      main_db:
        condition: service_healthy
volumes:
  volume_auth:
  volume_main:
```

Вывод

Я упаковал микросервисное приложение в контейнеры Docker и настроил их взаимодействие при помощи docker-compose