САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 4 Упаковка приложения в Docker

Выполнила:

Хорошкеева Ксения

Группа К33402

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. Делать это можно как с помощью docker-compose так и с помощью docker swarm. При разумном использовании swirl вы получите дополнительные баллы.

Ход работы

Dockerfile основного сервиса:

```
📭 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
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
COPY --chown=node:node src/configs/db.js ./dist/configs/db.js
COPY --chown=node:node .sequelizerc ./dist/
COPY --chown=node:node src/seeders/*.js ./dist/seeders/
EXPOSE 8000
```

Dockerfile сервиса бронирования:

```
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
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 8001
```

Файл docker-compose.yml:

```
environment:
         - MYSQL_USER=mysql
         - MYSQL_PASSWORD=mysql
       - MYSQL_DATABASE=mysql
          - MYSQL_ROOT_PASSWORD=mysql
        - 3306
         data_main:/var/lib/mysql
19 ▶ 🖯 db_booking:
          - MYSQL_USER=mysql
          - MYSQL_PASSWORD=mysql
          - MYSQL_DATABASE=mysql
         - MYSQL_ROOT_PASSWORD=mysql
         data_booking:/var/lib/mysql
```

```
main:

build:

context: main

ports:

- "8000:8000"

depends_on:

db_main:

condition: service_healthy

booking:

build:

context: booking

depends_on:

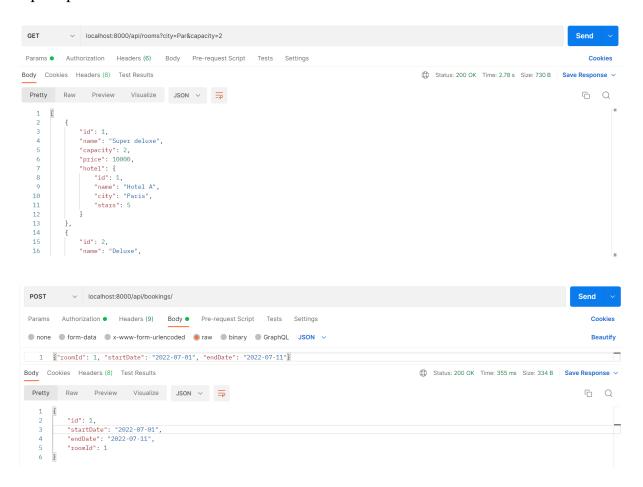
db_booking:

dd_db_booking:

dd_db_dooking:

dd_da_booking:
```

Проверка:



Вывод

Я упаковала приложение в контейнеры Docker с использованием утилиты Docker-compose.