САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа № 4

Выполнил:

Пырков Владислав

Группа К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

Необходимо упаковать ваше приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения, а также настроить общение микросервисов между собой посредством RabbitMQ. Делать это можно как с помощью docker-compose так и с помощью docker swarm.

Ход работы

Для данной работы необходимо было создать docker - контейнеры для каждой части нашего приложения. В файле docker-compose была описана работа со всеми частями приложения: контейнеры для основного сервера и микросервиса авторизации, образ PostgreSQL для работы с базой данных, образ для работы с RabbitMQ.

```yaml
services:
  auth:
    build:
      context: ./auth
      dockerfile: Dockerfile
    volumes:
      - ./auth/db.sqlite:/app/db.sqlite
    depends_on:
      - rabbitmq
    networks:
      - mynetwork
    environment:
      - RABBITMQ_HOST=rabbitmq
    working_dir: /app
    ports:
      - '8001:8001'
```

```yaml
app:
  build:
    context: ./app
    dockerfile: Dockerfile
  volumes:
    - ./app/db.sqlite:/app/db.sqlite
  depends_on:
    - rabbitmq
  networks:
    - mynetwork
  environment:
    - RABBITMQ_HOST=rabbitmq
    - AUTH_SERVICE=http://auth:8001
  working_dir: /app
  ports:
    - '8000:8000'
```

```yaml
rabbitmq:
  image: rabbitmq:3-management
  ports:
    - '5672:5672'
    - '15672:15672'
  networks:
    - mynetwork

postgres:
  image: postgres:16
  environment:
    POSTGRES_DB: 'postgres'
    POSTGRES_USER: 'postgres'
    POSTGRES_PASSWORD: 'toraha01'
  ports:
    - '5432:5432'
```

Имеющиеся контейнеры были описаны с помощью Dockerfile. При этом для предотвращения ошибок и улучшения производительности контейнеров были созданы .dockerignore файлы, описывающие директории, которые не нужно использовать при сборке контейнеров.

labs > K33402 > Пырков_Владислав > lab1 > auth > Dockerfile > ...

```dockerfile
FROM node:20

WORKDIR /app

COPY package.json ./

RUN npm i

COPY . .

EXPOSE 3001

CMD ["npm", "start"]
```

labs > K33402 > Пырков_Владислав > lab1 > auth > .dockerignore

```
node_modules

.gitignore
.git/
Dockerfile
```

Также в отдельной директории была настроена работа с RabbitMQ для более удобного взаимодействия с процессом авторизации и регистрации.

```
dotenv.config()

const rabbitmqHost = process.env.RABBITMQ_HOST || 'localhost'

amqp.connect(`amqp://${rabbitmqHost}`, function (error0, connection) {
    if (error0) {
        throw error0
    }
    connection.createChannel(function (error1, channel) {
        if (error1) {
            throw error1
        }

        const queue = 'authorize'

        channel.assertQueue(queue, {
            durable: false,
        })

        channel.sendToQueue(queue, Buffer.from('готово'))
        console.log(" [x] Sent 'готово'")
    })
})
```

По итогу полученное приложение может быть запущено с помощью docker compose одной командой, а взаимодействие его частей регулируется с помощью RabbitMQ.

Вывод

В процессе работы упростили запуск и сборку приложения с помощью docker и docker-compose, а также настроили взаимодействие отдельных частей приложения, с помощью RabbitMQ. Получили навыки работы с данными инструментами.