САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа № 4

Выполнил:

Казанков Илья 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.

```yaml
services:
  app:
    container_name: app
    build:
      context: ./app
      dockerfile: Dockerfile
    environment:
      - PORT=8000
      - DB_NAME=postgres
      - DB_USERNAME=postgres
      - DB_PASSWORD=admin
      - DB_HOST=postgresdb
    depends_on:
      - postgresdb
    ports:
      - '8000:8000'
    networks:
      - mynetwork
```

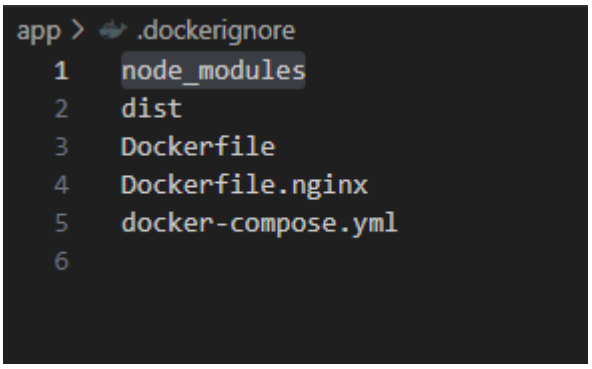
```yaml
product-service:
  container_name: product-service
  build:
    context: ./app
    dockerfile: Dockerfile
  environment:
    - PORT=5000
    - DB_NAME=postgres
    - DB_USERNAME=postgres
    - DB_PASSWORD=admin
    - DB_HOST=postgresdb
  depends_on:
    - postgresdb
  networks:
    - mynetwork
  ports:
    - '5000:5000'
```

```yaml
nginx:
  image: nginx:alpine
  ports:
    - "80:80"
  volumes:
    - ./nginx.conf:/etc/nginx/nginx.conf
  networks:
    - mynetwork

postgresdb:
```

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.

product-service > Dockerfile > ...

```dockerfile
FROM node:20

WORKDIR /product-service

COPY package.json ./

RUN npm i

COPY . .

RUN npm run build

EXPOSE 5000

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



По итогу полученное приложение может быть запущено с помощью docker compose одной командой.

Вывод

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