САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №4

Выполнил:

Кадникова Екатерина

Группа К3341

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

- реализовать Dockerfile для каждого сервиса;
- написать общий docker-compose.yml;
- настроить сетевое взаимодействие между сервисами.

Ход работы

1. Подготовка Dockerfile

Для каждого сервиса был создан Dockerfile (пример Dockerfile для Auth сервиса - на Листинге 1).

Общая структура одинакова для всех сервисов. Был выбран базовый образ - node:18-alpine. Сначала копируются и устанавливаются все зависимости, а потом уже копируется исходный код, собирается сам проект и открывается нужный порт.

Листинг 1 - Dockerfile для Auth сервиса

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
EXPOSE 3000
CMD ["npm", "start"]
```

2. Подготовка docker-compose

Был создан общий docker-compose файл с описанием каждого сервиса, определением томов для данных и сетевых настроек.

Для сервиса PostgreSQL (см. Листинг 2) были заданы основные параметры окружения, постоянное хранение данных через volume, явное удаление креденшелов.

Листинг 2 - Настройки для PostgreSQL

```
postgres:
 image: postgres:13
 environment:
   POSTGRES_USER: postgres
   POSTGRES_PASSWORD: richard2023
   POSTGRES_DB: lab3_db
 volumes:
```

```
    - postgres-data:/var/lib/postgresql/data
  ports:
    - "5433:5432"
  networks:
    - microservices-network
  healthcheck:
    test: [ "CMD-SHELL", "pg_isready -U postgres" ]
    interval: 5s
    timeout: 5s
    retries: 10
```

Для каждого сервиса (пример секции - на Листинге 3) были заданы переменные окружения, проброс портов, зависимости запуска (например, от других сервисов), политика перезапуска, подключение к сети microservices-network для изолированного взаимодействия с другими сервисами по имени и логирование.

Листинг 3 - Настройки для User-service

```
  user-service:
   build:
     context: ./user-service
     dockerfile: Dockerfile
   environment:
     PORT: 3001
     DB_HOST: postgres
     DB_PORT: 5432
     DB_USERNAME: postgres
     DB_PASSWORD: richard2023
     DB_NAME: lab3_db
     JWT_SECRET: richardrichardrichard
     JWT_EXPIRES_IN: 360000
     AUTH_SERVICE_URL: http://auth-service:3000
   ports:
     - "3001:3001"
   depends_on:
     - postgres
     - auth-service
   restart: on-failure
   networks:
     - microservices-network
   logging:
     driver: json-file
     options:
       max-size: "10m"
       max-file: "3"
```

3. Результат сборки

В результате запуска docker-compose было проверено, что каждый сервис успешно поднят (см. Рисунок 1). Также была проверена работоспособность каждого сервиса и их взаимодействия.

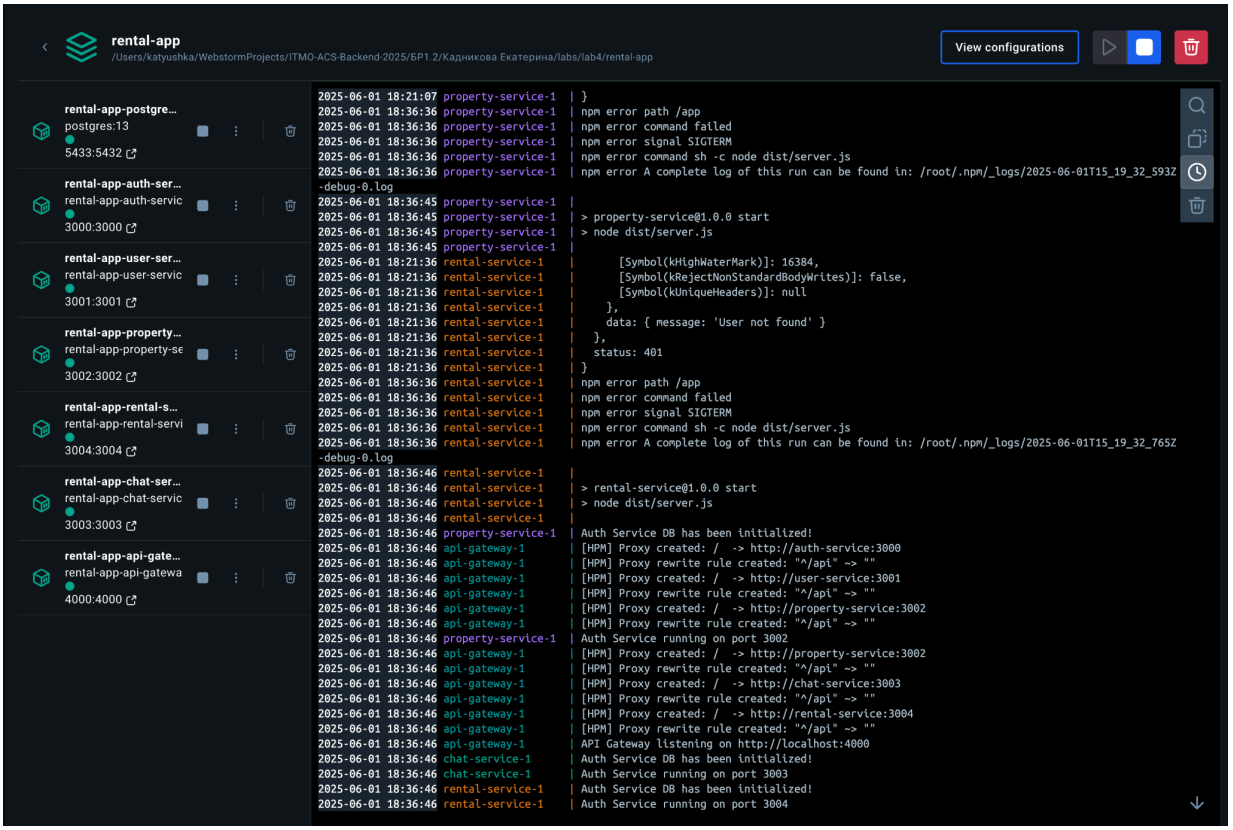

Рисунок 1 - Состояние запущенного контейнера

Вывод

В рамках работы было контейнеризировано приложение, настроено сетевое взаимодействие между контейнеризированными сервисами и проверено состояние каждого сервиса и их работоспособность.