

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Лабораторная работа №4

Выполнил:

Соловьева П.А.

Группа К3344

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. Создание docker-compose файла

```
version: '3.9'

services:
  postgres:
    image: postgres:latest
    container_name: postgres
    restart: always
    environment:
      POSTGRES_USER: user
      POSTGRES_PASSWORD: password
    ports:
      - "5432:5432"
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
    networks:
      - backend

  users-service:
    build:
      context: ./users_service
    container_name: users-service
    restart: always
    environment:
      DATABASE_URL: postgres://user:password@postgres:5432/users
      PORT: 5001
    ports:
      - "5001:5001"
    depends_on:
      - postgres
    networks:
      - backend

  property-service:
    build:
      context: ./property_service
    container_name: property-service
    restart: always
    environment:
      DATABASE_URL: postgres://user:password@postgres:5432/property
      PORT: 5002
    ports:
      - "5002:5002"
    depends_on:
```

```
- postgres
networks:
  - backend

rental-service:
  build:
    context: ./rental_service
  container_name: rental-service
  restart: always
  environment:
    DATABASE_URL: postgres://user:password@postgres:5432/rental
    PORT: 5003
  ports:
    - "5003:5003"
  depends_on:
    - postgres
    - property-service
  networks:
    - backend

networks:
  backend:
    driver: bridge

volumes:
  postgres_data:
```

## 2. Создание docker файлов

Для микросервиса пользователей:

```
FROM node:20-alpine

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm ci

COPY . .

RUN npm run build

EXPOSE 5001

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

Для микросервиса собственности:

```
FROM node:20-alpine

WORKDIR /usr/src/app

COPY package*.json ./
```

```
RUN npm ci

COPY . .

RUN npm run build

EXPOSE 5002

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

Для микросервиса аренды:

```
FROM node:20-alpine

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm ci

COPY . .

RUN npm run build

EXPOSE 5003

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

## Вывод

В процессе выполнения задания был реализован контейнеризированный подход к развёртыванию микросервисов. Для каждого сервиса (users-service, property-service и rental-service) были разработаны отдельные Dockerfile, позволяющие собрать образы и запускать сервисы в изолированных окружениях.

Также был создан общий файл docker-compose.yml, который объединяет все микросервисы и базу данных PostgreSQL в единую систему. С помощью сетевого драйвера bridge было настроено взаимодействие между сервисами, что обеспечивает их корректное взаимодействие и доступ к базе данных.

Использование Docker и docker-compose позволило:

- стандартизировать процесс запуска приложения;
- упростить настройку окружения для каждого микросервиса;
- обеспечить масштабируемость и переносимость приложения;
- автоматизировать управление зависимостями и сетевыми связями.

Таким образом, проект был успешно подготовлен к запуску в контейнеризированной среде, что делает его более гибким, надёжным и удобным для дальнейшего развития и эксплуатации.