

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 6

Выполнил:

Якунин Андрей

**Группа
К3441**

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2026 г.

Задача

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

Ход работы

Написал докерфайл для каждого сервиса и докер.yml

authService

```
FROM node:20-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
EXPOSE 3000
CMD ["npm", "start"] |
```

chat-service

```
FROM node:20-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
EXPOSE 3003
CMD ["npm", "start"] |
```

Property-service

```
FROM node:20-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
EXPOSE 3002
CMD ["npm", "start"]
```

rental-service

```
FROM node:20-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
EXPOSE 3004
CMD ["npm", "start"]
```

Docker-compose.yml

```
version: '3.8'

services:

  postgres:
    image: postgres:15-alpine
    restart: always
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: microservices
    ports:
      - "5433:5432"
    volumes:
      - pgdata:/var/lib/postgresql/data
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql

  auth-service:
    build: ./authService
    ports:
      - "3000:3000"
    environment:
      - PORT=3000
```

```
 DATABASE_URL=postgres://postgres:postgres@postgres:5432/auth_db
```

```
 depends_on:
```

```
 - postgres
```

```
 user-service:
```

```
 build: ./user-service
```

```
 ports:
```

```
 - "3001:3001"
```

```
 environment:
```

```
 - PORT=3001
```

```
 DATABASE_URL=postgres://postgres:postgres@postgres:5432/user_db
```

```
 depends_on:
```

```
 - postgres
```

```
 property-service:
```

```
 build: ./property-service
```

```
 ports:
```

```
 - "3002:3002"
```

```
 environment:
```

```
 - PORT=3002
```

```
 DATABASE_URL=postgres://postgres:postgres@postgres:5432/property_db
```

```
 depends_on:
```

- postgres

```
chat-service:
```

```
 build: ./chat-service
```

```
 ports:
```

- "3003:3003"

```
 environment:
```

- PORT=3003

```
 DATABASE_URL=postgres://postgres:postgres@postgres:5432/message_db
```

```
 depends_on:
```

- postgres

```
rental-service:
```

```
 build: ./rental-service
```

```
 ports:
```

- "3004:3004"

```
 environment:
```

- PORT=3004

```
DATABASE_URL=postgres://postgres:postgres@postgres:5432/rental_db

depends_on:
  - postgres


api-gateway:
  build: ./api-gateway
  ports:
    - "4000:4000"
  environment:
    - PORT=4000
    - AUTH_SERVICE_URL=http://auth-service:3000


PROPERTY_SERVICE_URL=http://property-service:3002
  - CHAT_SERVICE_URL=http://chat-service:3003
  - RENTAL_SERVICE_URL=http://rental-service:3004

depends_on:
  - auth-service
  - user-service
  - property-service
  - chat-service
  - rental-service
```

```
volumes:
```

```
  pgdata:
```

Вывод

В ходе выполнения лабораторной работы были реализованы Dockerfile для каждого микросервиса и создан общий файл docker-compose.yml. Было настроено сетевое взаимодействие между сервисами и базой данных PostgreSQL, что позволило обеспечить корректный запуск и совместную работу всех компонентов системы. Использование Docker и Docker Compose упростило процесс развертывания и повысило воспроизводимость среды разработки