

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №4

Выполнил:

Шалунов Андрей

Группа К3340

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. Разработка Dockerfile

Выбран multi-stage билд на базе node:20-alpine.

В первом этапе (builder) устанавливаются все зависимости (npm ci), компилируется TypeScript в dist/.

Во втором этапе (runner) копируются в образ только продакшен-зависимости и готовый JS-код, задаётся переменная PORT и команда запуска node dist/app.js.

```
FROM node:20-alpine AS builder

WORKDIR /app

COPY package*.json ./

RUN npm ci

COPY tsconfig*.json ./

COPY src ./src

RUN npm run build

FROM node:20-alpine AS runner

WORKDIR /app

COPY --from=builder /app/dist ./dist

COPY package*.json ./

RUN npm ci --production
```

```
COPY --from=builder /app/node_modules ./node_modules

COPY .env ./

ARG PORT=3000

ENV PORT=${PORT}

EXPOSE ${PORT}

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

2. Сборка docker-compose.yml

Для каждой службы БД создан контейнер postgres:13-alpine с уникальным томом и переменными POSTGRES_DB, POSTGRES_USER=test, POSTGRES_PASSWORD=test.

Для каждого микросервиса указаны блоки build.context, build.args (порт), env_file (локальный .env), ports, depends_on и restart: always.

API Gateway зависит от всех бэкенд-сервисов и проксирует внешние запросы по правилам /api/auth → auth-service, /api/users → user-service и т.п.

3. Настройка среды

В каждом .env прописаны DB_HOST (имя сервиса-БД), DB_NAME, DB_USER, DB_PASSWORD, порт, а при необходимости URL соседних сервисов (USER_SERVICE_URL, BOOKING_SERVICE_URL и др.).

Пример .env сервиса аутентификации:

```
PORT=3000

DB_HOST=auth-db

DB_PORT=5432

DB_NAME=auth_db

DB_USER=test

DB_PASSWORD=test
```

```
JWT_SECRET_KEY=jwt_secret_auth  
  
JWT_ACCESS_TOKEN_LIFETIME=3600  
  
USER_SERVICE_URL=http://user-service:3001
```

Все контейнеры запущены командой `docker-compose up --build` или через Docker Desktop Stacks.

4. Тестирование

Проверено, что все контейнеры имеют статус Up.

Через Gateway на `http://localhost:8000` выполнены запросы на регистрацию и логин, создание и чтение сущностей во всех сервисах.

Межсервисные запросы (Message → User, Message → Booking) отработали корректно.

Вывод

Контейнеризация позволила упаковать каждый микросервис в свой образ с минимальным размером и чётко задать зависимости, а общий `docker-compose.yml` организовал автоматический запуск всей системы в единой сети.