

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №4

Выполнил:

Бархатова Наталья

Группа К3339

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

- выделить самостоятельные модули в вашем приложении;
- провести разделение своего API на микросервисы (минимум, их должно быть 3);
- настроить сетевое взаимодействие между микросервисами.

Ход работы

Сначала был написан Dockerfile для сборки всех микросервисов, Используется лёгкий образ node:22-alpine. Рабочая директория устанавливается в /app. На первом этапе копируются файлы зависимостей (package*.json) и выполняется установка модулей с помощью npm install, потом копируется весь исходный код, выполняется сборка TypeScript-приложения (npm run build), после чего запускается готовое приложение через node dist/app.js.

```
FROM node:22-alpine

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

RUN npm run build

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

Затем был написан docker-compose.yml файл, который описывает микросервисную архитектуру фитнес-приложения, включающую четыре сервиса (auth_service, user_service, workout_service, blog_service) и соответствующие базы данных PostgreSQL (кроме сервиса аутентификации). Каждый сервис разворачивается в отдельном контейнере, использует собственную БД и подключён к общей виртуальной сети fitness-network. Для хранения данных определены внешние тома, а зависимости между сервисами указаны с помощью depends_on.

```
version: '3.8'

services:
  auth_service:
```

```
    build: ./auth_service
    ports:
        "3000:3000"
    environment:
        PORT=3000
        AUTH_SECRET=supersecretkey
    networks:
        fitness-network

user_service:
    build: ./user_service
    ports:
        "3001:3001"
    depends_on:
        user_db
        auth_service
    environment:
        PORT=3001
        DB_HOST=user_db
        DB_PORT=5432
        DB_USER=nata
        DB_PASSWORD=1
    - DB_NAME=fitness_app_user
    - AUTH_SERVICE_URL=http://auth_service:3000
    networks:
        - fitness-network

user_db:
    image: postgres:15
    environment:
        - POSTGRES_USER=nata
        - POSTGRES_PASSWORD=1
        - POSTGRES_DB=fitness_app_user
    volumes:
        user_db_data:/var/lib/postgresql/data
    networks:
        - fitness-network

workout_service:
    build: ./workout_service
    ports:
        - "3002:3002"
    depends_on:
        - workout_db
        - auth_service
    environment:
        - PORT=3002
        - DB_HOST=workout_db
```

- DB_PORT=5432
- DB_USER=nata
- DB_PASSWORD=1
- DB_NAME=fitness_app_workout
- AUTH_SERVICE_URL=http://auth_service:3000

networks:

- fitness-network

workout_db:

- image: postgres:15
- environment:
 - POSTGRES_USER=nata
 - POSTGRES_PASSWORD=1
 - POSTGRES_DB=fitness_app_workout
- volumes:
 - workout_db_data:/var/lib/postgresql/data
- networks:
 - fitness-network

blog_service:

- build: ./blog_service
- ports:
 - "3003:3003"
- depends_on:
 - blog_db
 - auth_service
- environment:
 - PORT=3003
 - DB_HOST=blog_db
 - DB_PORT=5432
 - DB_USER=nata
 - DB_PASSWORD=1
 - DB_NAME=fitness_app_blog
 - AUTH_SERVICE_URL=http://auth_service:3000
- networks:
 - fitness-network

blog_db:

- image: postgres:15
- environment:
 - POSTGRES_USER=nata
 - POSTGRES_PASSWORD=1
 - POSTGRES_DB=fitness_app_blog
- volumes:
 - blog_db_data:/var/lib/postgresql/data
- networks:
 - fitness-network

```
volumes:
user_db_data:
workout_db_data:
blog_db_data:

networks:
fitness-network:
  driver: bridge
```

```
PS D:\ITMO-ACS-Backend-2025\БР1.1\Бархатова Наталья\labs\lab4> docker compose up -d
[+] Running 7/7
✔ Container lab4-blog_db-1           Started           0.0s
✔ Container lab4-auth_service-1      Started           0.0s
✔ Container lab4-user_db-1           Started           0.0s
✔ Container lab4-workout_db-1        Started           0.0s
✔ Container lab4-workout_service-1   Started           0.0s
✔ Container lab4-blog_service-1      Started           0.0s
✔ Container lab4-user_service-1      Started           0.0s
PS D:\ITMO-ACS-Backend-2025\БР1.1\Бархатова Наталья\labs\lab4> |
```

Вывод

В ходе выполнения работы была реализована микросервисная архитектура фитнес-приложения с использованием Docker и Docker Compose. Каждый компонент системы — аутентификация, управление пользователями, тренировками и блог — был вынесен в отдельный сервис с собственной базой данных PostgreSQL. Для каждого сервиса был создан единый Dockerfile, обеспечивающий сборку, установку зависимостей и запуск приложения в изолированной среде. Использование общей пользовательской сети и внешних томов обеспечило взаимодействие между сервисами и сохранность данных.