

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №4

Выполнил:

Цой Степан

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. Реализация Dockerfile для каждого сервиса

Мы создали оптимизированные Dockerfile для всех четырех сервисов приложения, используя базовый образ node:18-alpine для минимального размера контейнеров. Каждый Dockerfile включает многоэтапную сборку: копирование package.json, установку зависимостей (включая devDependencies для TypeScript компиляции), построение проекта и финальный запуск скомпилированного приложения.

2. Создание общего docker-compose.yml

Мы разработали единый docker-compose.yml файл, который оркестрирует все сервисы как связанную систему, определяя порядок запуска через зависимости (depends_on) и условия готовности (condition: service_healthy). В конфигурации мы настроили проброс портов (3000-3003), переменные окружения для подключения к локальной PostgreSQL базе данных через host.docker.internal, и сеть recipe-network для изолированного взаимодействия.

3. Настройка сетевого взаимодействия между сервисами

Мы организовали сетевое взаимодействие через bridge-сеть recipe-network, которая обеспечивает изоляцию контейнеров от внешней среды при сохранении связи между ними. Для доступа к локальной базе данных использовали специальный DNS-адрес host.docker.internal, работающий на Windows/Mac, с fallback-решением для Linux систем. Настроили корректные URL для межсервисного взаимодействия (например, http://auth-service:3001), что позволяет микросервисам общаться по внутренним DNS-именам, а также реализовали правильную последовательность запуска, гарантирующую, что Gateway начинает работать только после полной готовности всех зависимых сервисов.

Вывод

Мы реализовали Dockerfile для каждого сервиса, написали общий docker-compose.yml и настроили сетевое взаимодействие между сервисами.