

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 4: Контейнеризация написанного
приложения средствами docker

Выполнил:

Лазебный Всеволод

Группа К3344

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Нужно обернуть работу в Докер и настроить взаимодействие образов между друг-другом.

Должно быть реализовано:

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

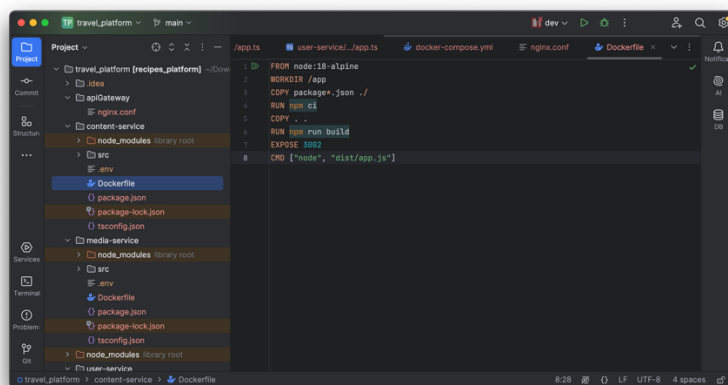
Ход работы

В моей работе смог обернуть три микросервиса в образы через docker-compose, также настроив взаимодействие с пакетами и доп параметрами.

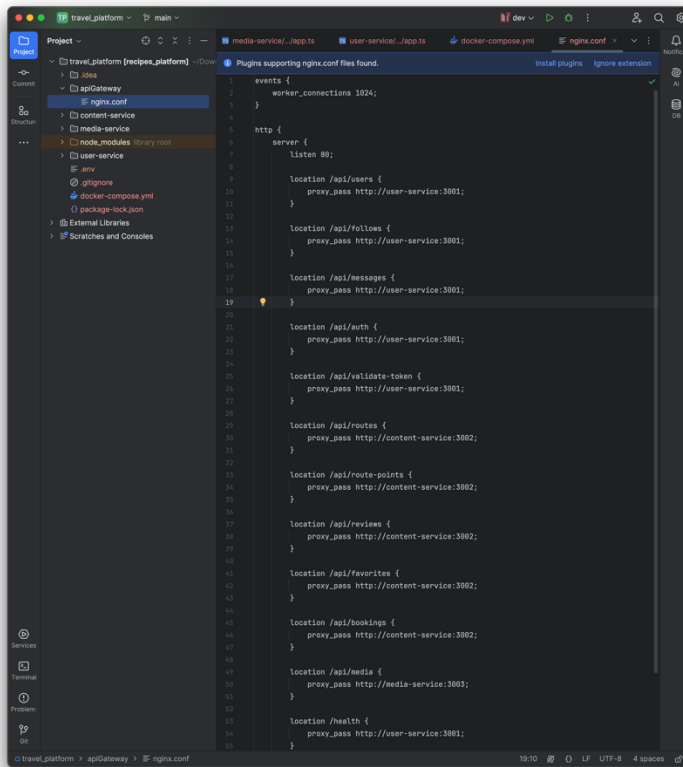
В ходе работы получилось 5 образов

- pgAdmin – образ с визуалом для бд;
- PostgreSQL – образ с хранилищем бд;
- media-service – образ для работы с медиафайлами по url-ссылке.
- user-service – образ в котором происходит регистрация, вход, проброс запросов по поиску пользователей и возможность подписки на путешествия;
- content-service – образ в котором происходит работа с основными сущностями и объектами;

Dockerfile:

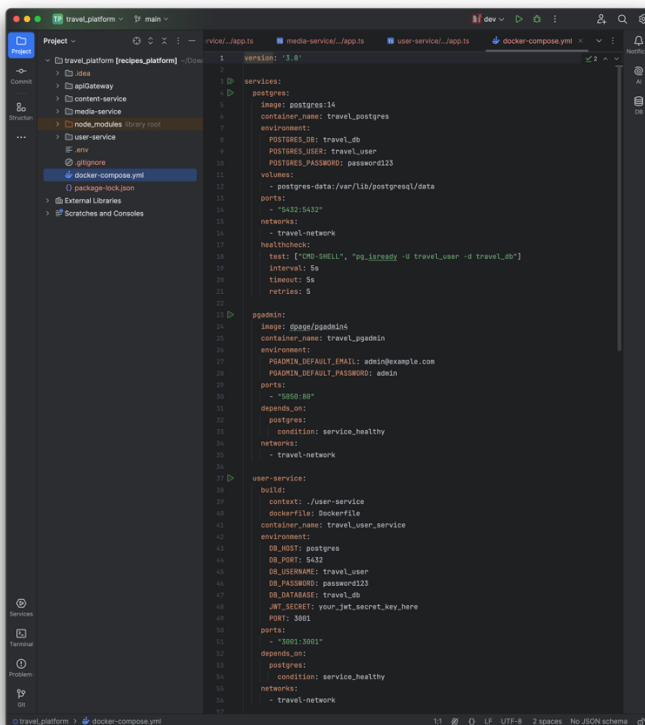


nginx:



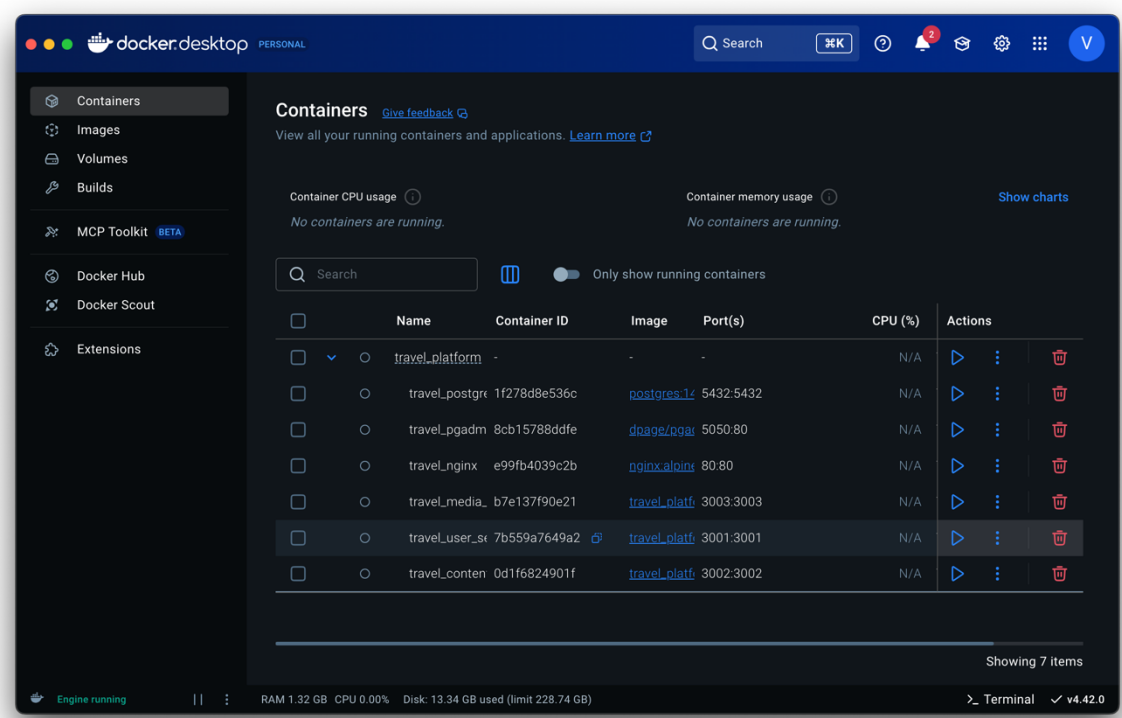
```
1 events {  
2     worker_connections 1024;  
3 }  
4  
5 http {  
6     server {  
7         listen 80;  
8  
9         location /api/users {  
10            proxy_pass http://user-service:3001;  
11        }  
12  
13        location /api/follows {  
14            proxy_pass http://user-service:3001;  
15        }  
16  
17        location /api/messages {  
18            proxy_pass http://user-service:3001;  
19        }  
20  
21        location /api/auth {  
22            proxy_pass http://user-service:3001;  
23        }  
24  
25        location /api/validate-token {  
26            proxy_pass http://user-service:3001;  
27        }  
28  
29        location /api/routes {  
30            proxy_pass http://content-service:3002;  
31        }  
32  
33        location /api/route-points {  
34            proxy_pass http://content-service:3002;  
35        }  
36  
37        location /api/reviews {  
38            proxy_pass http://content-service:3002;  
39        }  
40  
41        location /api/favorites {  
42            proxy_pass http://content-service:3002;  
43        }  
44  
45        location /api/bookings {  
46            proxy_pass http://content-service:3002;  
47        }  
48  
49        location /api/media {  
50            proxy_pass http://media-service:3003;  
51        }  
52  
53        location /health {  
54            proxy_pass http://user-service:3001;  
55        }  
56    }  
57 }
```

docker-compose:



```
1 version: '3.8'  
2  
3 services:  
4     postgres:  
5         image: postgres:14  
6         container_name: travel_postgres  
7         environment:  
8             POSTGRES_DB: travel_db  
9             POSTGRES_USER: travel_user  
10            POSTGRES_PASSWORD: password123  
11         volumes:  
12             - postgres-data:/var/lib/postgresql/data  
13         ports:  
14             - "5432:5432"  
15         networks:  
16             - travel-network  
17         healthcheck:  
18             test: ["CMD-SHELL", "ps -l && grep -q travel_user -s travel_db"]  
19             interval: 5s  
20             timeout: 5s  
21             retries: 5  
22  
23     pgadmin:  
24         image: dpage/pgadmin4  
25         container_name: travel_pgadmin  
26         environment:  
27             PGADMIN_DEFAULT_EMAIL: admin@example.com  
28             PGADMIN_DEFAULT_PASSWORD: admin  
29         ports:  
30             - "8080:80"  
31         depends_on:  
32             - postgres  
33         condition: service_healthy  
34         networks:  
35             - travel-network  
36  
37     user-service:  
38         build:  
39             context: ./user-service  
40             dockerfile: Dockerfile  
41         container_name: travel_user_service  
42         environment:  
43             DB_HOST: postgres  
44             DB_PORT: 5432  
45             DB_USERNAME: travel_user  
46             DB_PASSWORD: password123  
47             DB_DATABASE: travel_db  
48             JWT_SECRET: your_jwt_secret_key_here  
49         ports:  
50             - "3001:3001"  
51         depends_on:  
52             - postgres  
53         condition: service_healthy  
54         networks:  
55             - travel-network
```

Собранный контейнер:



Вывод

В ходе этой лабораторной были реализованы 5 образов, задана связь между микросервисами через nginx, а также собран контейнер со работающим бэкендом.