

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 4

Выполнил:

Тюмин Никита

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

Необходимо упаковать ваше приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения. Делать это можно как с помощью docker-compose так и с помощью docker swarm. При разумном использовании swirl вы получите дополнительные баллы.

Ход работы

Были написаны Dockerfile-ы для микросервисов:

```
FROM node:20-alpine
WORKDIR /app

COPY package*.json ./
RUN npm install
RUN npm update shared-core shared-database
COPY . .
RUN npm run build

EXPOSE 8000
CMD [ "node", "dist/app.js" ]
```

И docker-compose.yml файл:

```
version: '3.8'

services:
  auth:
    container_name: node_auth
    ports:
      - "4000:8000"
    restart: unless-stopped
    build:
      context: ./auth
    networks:
      - node
  recommendations:
    container_name: node_recommendations
    ports:
      - "6000:8000"
    restart: unless-stopped
    build:
      context: ./recommendations
    networks:
      - node
  tours:
    container_name: node_tours
    ports:
      - "5000:8000"
    restart: unless-stopped
    build:
      context: ./tours
    networks:
      - node
  core:
    depends_on:
      - postgres
    container_name: node_core
    restart: "no"
    build:
      context: ./core
    networks:
```

```

- node
command: ["sh", "-c", "npm run migrate && npm run seed"]

postgres:
  container_name: node_postgres
  restart: unless-stopped
  image: postgres:14
  ports:
    - "5432:5432"
  env_file:
    - ./pgsql/.env
  volumes:
    - node-pgsql-volume:/var/lib/postgresql/data
  networks:
    - node

networks:
  node:
    driver: bridge

volumes:
  node-pgsql-volume:
    driver: local

```

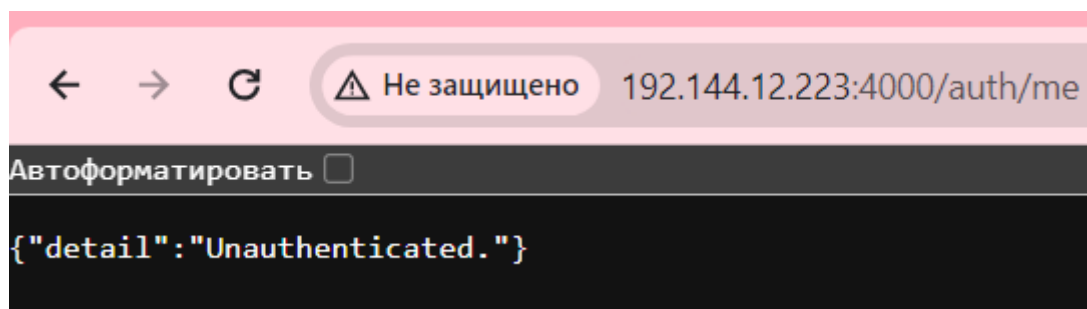
Приложение развернуто в облаке cloud.ru:

```

dev@myvm:~/ITMO-ICT-Backend-2024/labs/K33402/Тюмин Никита/lab4$ ls
auth  core  docker-compose.yml  pgsql  recommendations  report3  pdf  shared  tours
dev@myvm:~/ITMO-ICT-Backend-2024/labs/K33402/Тюмин Никита/lab4$ docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS        PORTS                               NAMES
1fa1403c6d40  lab4_recommendations  "docker-entrypoint.s..." 17 hours ago  Up 17 hours  0.0.0.0:6000->8000/tcp, :::6000->8000/tcp  node_recommendations
e1cf2861d454   lab4_tours            "docker-entrypoint.s..." 17 hours ago  Up 17 hours  0.0.0.0:5000->8000/tcp, :::5000->8000/tcp  node_tours
e7783c82732a   lab4_auth             "docker-entrypoint.s..." 17 hours ago  Up 17 hours  0.0.0.0:4000->8000/tcp, :::4000->8000/tcp  node_auth
26f2b13ad73c   postgres:14           "docker-entrypoint.s..." 17 hours ago  Up 17 hours  0.0.0.0:5432->5432/tcp, :::5432->5432/tcp  node_postgres
dev@myvm:~/ITMO-ICT-Backend-2024/labs/K33402/Тюмин Никита/lab4$

```

И доступно по публичному IP:



Сетевое взаимодействие контейнеров обеспечено через http. В docker-compose.yml файле создана сеть node, поэтому можно контейнеры могут обращаться друг к другу по имени:

```
dev@myvm:~/ITMO-ICT-Backend-2024/labs/K33402/Тюмин Никита/lab4$ cat tours/.env
PORT=8000
NODE_PATH=./src
APP_PATH=/var/www/app

AUTH_SERVICE_URL=http://node_auth:8000

DB_DATABASE=test
DB_DIALECT=postgres
DB_USERNAME=test
DB_PASSWORD=test
DB_STORAGE=
DB_HOST=node_postgres
DB_PORT=5432

JWT_ALG=HS256
JWT_SECRET=secret
JWT_ACCESS_TOKEN_TTL=1800
JWT_REFRESH_TOKEN_TTL=172800
dev@myvm:~/ITMO-ICT-Backend-2024/labs/K33402/Тюмин Никита/lab4$ |
```

Вывод

Приложение было упаковано в Docker и развернуто в vps с помощью docker-compose. Сетевое взаимодействие контейнеров обеспечено с помощью протокола http.