

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Лабораторная работа

Выполнила:

Олейникова Полина

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

Создадим Dockerfile в микросервисах:

```
Dockerfile X
1 FROM node:18-alpine
2
3 WORKDIR /backend/service-auth
4
5 COPY package*.json ./
6 RUN npm install
7
8 COPY . .
9
10 EXPOSE ${PORT}
11 CMD [ "npm", "run", "dev" ]

Dockerfile X
1 FROM node:18-alpine
2
3 WORKDIR /backend/service-articles
4
5 COPY package*.json ./
6 RUN npm install
7
8 COPY . .
9
10 EXPOSE ${PORT}
11 CMD [ "npm", "run", "dev" ]
```

Создадим gateway с помощью nginx

```
prodconf X
1 server {
2     listen 8080;
3
4     location /check-token {
5         internal;
6         proxy_pass http://service-auth:8080/v1/check-token;
7         proxy_redirect off;
8         proxy_set_header Host $host;
9         proxy_set_header X-Real-IP $remote_addr;
10        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
11        proxy_set_header X-Forwarded-Host $server_name;
12        proxy_set_header Content-Length "";
13        proxy_pass_request_body off;
14    }
15
16    location /auth {
17        rewrite ^/auth/(.*) /$1 break;
18
19        proxy_pass http://service-auth:8080;
20        proxy_redirect off;
21        proxy_set_header Host $host;
22        proxy_set_header X-Real-IP $remote_addr;
23        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
24        proxy_set_header X-Forwarded-Host $server_name;
25    }
26
27    location /articles {
28        auth_request /check-token;
29        auth_request_set $user_id $sent_http_user_id;
30        auth_request_set $user_role $sent_http_user_role;
31
32        rewrite ^/articles/(.*) /$1 break;
33
34        proxy_pass http://service-articles:8080;
35        proxy_redirect off;
36        proxy_set_header Host $host;
37        proxy_set_header X-Real-IP $remote_addr;
38        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
39        proxy_set_header X-Forwarded-Host $server_name;
40        proxy_set_header User-Id $user_id;
41        proxy_set_header User-Role $user_role;
42    }
43 }
```

И Dockerfile для него:

```
Dockerfile X
1 FROM nginx:1.18
2 COPY conf/prod.conf /etc/nginx/conf.d/prod.conf
3
```

Создадим docker-compose с сервисом gateway

```
docker-compose.yml X
1 version: '1.0'
2 services:
3   gateway:
4     container_name: gateway
5     build:
6       context: ./gateway/src
7     restart: always
8     ports:
9       - 8080:8080
10
```

Создадим сервисы для баз данных в docker-compose:

```
auth-db:
  image: postgres
  container_name: auth-db
  restart: always
  expose:
    - 5432
  environment:
    POSTGRES_USER: ${DB_USERNAME}
    POSTGRES_PASSWORD: ${DB_PASSWORD}
    POSTGRES_DB: ${DB_NAME_AUTH}
  volumes:
    - ./backend/auth_data:/var/lib/postgresql/data

articles-db:
  image: postgres
  container_name: articles-db
  restart: always
  expose:
    - 5432
  environment:
    POSTGRES_USER: ${DB_USERNAME}
    POSTGRES_PASSWORD: ${DB_PASSWORD}
    POSTGRES_DB: ${DB_NAME_ARTICLES}
  volumes:
    - ./backend/articles_data:/var/lib/postgresql/data
```

Создадим сервисы для микросервисов в docker-compose:

```

service-auth:
  build:
    context: ../auth
  container_name: service-auth
  restart: always
  expose:
    - 8008
  env_file:
    - service/auth/.env
  depends_on:
    - auth-db
  volumes:
    - /backend/service-auth/node_modules

service-articles:
  build:
    context: ../articles
  container_name: service-articles
  restart: always
  expose:
    - 8005
  env_file:
    - service/articles/.env
  depends_on:
    - articles-db
  volumes:
    - /backend/service-articles/node_modules

```

Запустим:

```

PS C:\Users\olein\copy\Учеба\бек\lr3\deployment> docker-compose -p articles-management-5 -f docker-compose.yaml up -d
[+] Running 5/6
 - Network articles-management-5_default Created
 ✓ Container articles-db Started
 ✓ Container gateway Started
 ✓ Container auth-db Started
 ✓ Container service-articles Started
 ✓ Container service-auth Started

```

Проверим:

POST

http://localhost:8080/auth/v1/register

Send

Params

Authorization

Headers (12)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```

1 {
2   "name": "Polina",
3   "email": "email",
4   "password": "password"
5 }

```

Body

Cookies

Headers (8)

Test Results

200 OK 123 ms 272 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

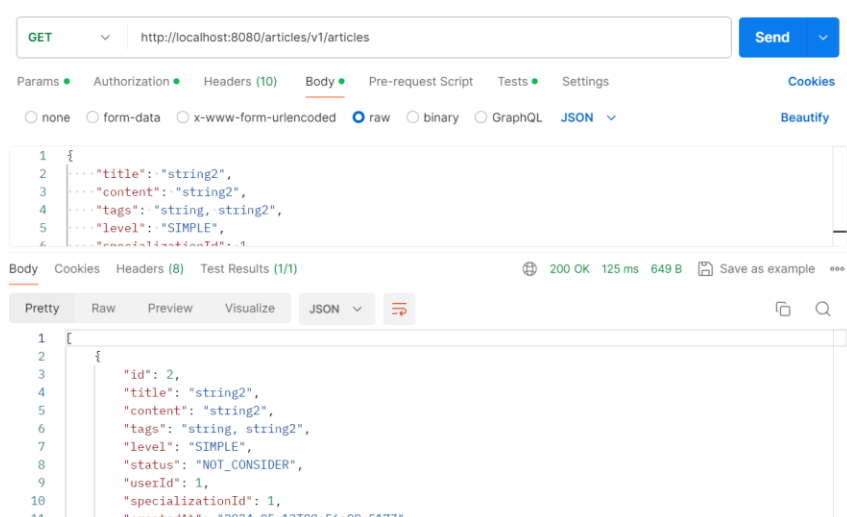
Copy

Search

```

1 {
2   "id": 2
3 }

```



## Вывод

В ходе данной работы было упаковано приложение в docker-контейнеры и обеспечено сетевое взаимодействие между различными частями вашего приложения с помощью docker-compose так и nginx.