

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Лабораторная работа 4

Выполнил: Зайцев Кирилл  
Дмитриевич

Группа К33402

Проверил:  
Добряков Д. И.

Санкт-Петербург

**Задача:**

Необходимо упаковать ваше приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения, а также настроить общение микросервисов между собой посредством RabbitMQ. Делать это можно как с помощью docker-compose так и с помощью docker swarm. При разумном использовании swirl вы получите дополнительные баллы.

1. Сменим БД(теперь нам придется учитывать подключение БД в docker-compose)

```
const sequelize : Sequelize = new Sequelize( options: {  
  dialect: 'postgres',  
  host: 'postgres',  
  username: 'postgres',  
  password: 'Kirill307477',  
  database: 'dockerBD',  
  models: [Product]  
});
```

2. Уберем костыльный прямой прокси и добавим nginx и разделим функционал приложения полноценно:

```
import { authMiddleware } from './middleware/auth';

const app : Express = express();
const sequelize : Sequelize = new Sequelize( options: {
  dialect: 'postgres',
  host: 'postgres',
  username: 'postgres',
  password: 'Kirill307477',
  database: 'dockerBD',
  models: [Product]
});

app.use(bodyParser.json());

const productController : ProductController = new ProductController();

app.post( path: '/products', authMiddleware, productController.create);
app.put( path: '/products/:id', authMiddleware, productController.update);
app.get('/products/search', productController.search);
```

```
7 http {
8     include      mime.types;
9     default_type  application/octet-stream;
10    sendfile      on;
11
12    upstream app {
13        server app:3000;
14    }
15
16    upstream product_service {
17        server product-service:3001;
18    }
19
20    server {
21        listen 80;
22
23        location / {
24            proxy_pass http://app;
25            proxy_set_header Host $host;
26            proxy_set_header X-Real-IP $remote_addr;
27            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
28            proxy_set_header X-Forwarded-Proto $scheme;
29        }
30
31        location /products {
32            proxy_pass http://product_service;
33            proxy_set_header Host $host;
34            proxy_set_header X-Real-IP $remote_addr;
35            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
36            proxy_set_header X-Forwarded-Proto $scheme;
37        }
38    }
39 }
40
```

В каждом сервисе создадим докерфайл, и распишем docker-compose.yml для сборки сервисов, подключения postgresql и nginx:

```
version: '3.9'

services:
  app:
    build:
      context: ./app
      dockerfile: Dockerfile
    ports:
      - "3000"
    depends_on:
      - postgres
    environment:
      - DATABASE_URL=postgres://postgres:Kirill307477@postgres:5432/dockerBD
    volumes:
      - ./app:/usr/src/app:cached
      - /usr/src/app/node_modules
    networks:
      - my-network

  product-service:
    build:
      context: ./product-service
      dockerfile: Dockerfile
    ports:
      - "3001"
    depends_on:
      - postgres
    environment:
      - DATABASE_URL=postgres://postgres:Kirill307477@postgres:5432/dockerBD
    volumes:
      - ./product-service:/usr/src/app:cached
      - /usr/src/app/node_modules
    networks:
      - my-network

  nginx:
    image: nginx:alpine
    ports:
      - "80:80"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
    networks:
      - my-network

  postgres:
    image: postgres:alpine
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: Kirill307477
      POSTGRES_DB: dockerBD
    volumes:
      - pgdata:/var/lib/postgresql/data
    networks:
      - my-network

networks:
  my-network:
```

```
driver: bridge

volumes:
  pgdata:
```

4. Для каждой папки с докер файлом распишем докер-игнор:

```
1  node_modules
2  app/node_modules
3  product-service/node_modules
4  dist
5
```

## 5. Запуск и проверка работы

<input type="checkbox"/>	lab4		Running (4/4)	0.01%	52 seconds ago			
<input type="checkbox"/>	postgres-1	c2d37a850100	<a href="#">postgres:alpine</a>	Running	0.01%	53 seconds ago		
<input type="checkbox"/>	nginx-1	48076c7953d8	<a href="#">nginx:alpine</a>	Running	<a href="#">80.80</a>	0%	53 seconds ago	
<input type="checkbox"/>	product-service-1	308b9e018ace	<a href="#">lab4-product-service</a>	Running	<a href="#">25404:3001</a>	0%	52 seconds ago	
<input type="checkbox"/>	app-1	cc76f5137ce6	<a href="#">lab4-app</a>	Running	<a href="#">25403:3000</a>	0%	52 seconds ago	

## Регистрация

[HTTP](#) docker check / reg Save Share

POST

http://localhost/register

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

Beautify

```
1  {
2    "email": "test@example.com",
3    "password": "password123",
4    "isAdmin": true
5  }
```

Body

Cookies

Headers (7)

Test Results

201 Created

82 ms

443 B

Save as example

More

Pretty

Raw

Preview

Visualize

JSON

More

```
1  {
2    "id": 14,
3    "email": "test@example.com",
4    "password": "$2b$10$ayz/2wVf11x0g2EjCN0jzOd5.NTNeDOUm1EtX3mNQDCxivKW7myZO",
5    "isAdmin": true,
6    "updatedAt": "2024-06-13T20:35:57.315Z",
7    "createdAt": "2024-06-13T20:35:57.315Z"
8  }
```

# Вход

HTTP

docker check / log

Save

Share

POST

http://localhost/login

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

1

2

3

4

{

"email": "test@example.com",

"password": "password123"

}

Body

Cookies

Headers (7)

Test Results

200 OK

63 ms

406 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

1

2

3

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTESImIzQWRtaW4iOnRydWUsIm1hdCI6MTcxODMxMDk3OSwiZXhwIjoxNzE4MzE0NTc5fQ.6TyYkssUEwFvLFb4UXQYkz1uIEpm5jC6qVR4RSq0hM8"

}

# Добавление товара:

HTTP

docker check / post prod

Save

Share

POST

http://localhost/products

Send

Params

Authorization

Headers (9)

Body

Scripts

Settings

Cookies

Auth Type

Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#).

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTESImIzQWRtaW4iOnRydWUsIm1hdCI6MTcxODMxMDk3OSwiZXhwIjoxNzE4MzE0NTc5fQ.6TyYkssUEwFvLFb4UXQYkz1uIEpm5jC6qVR4RSq0hM8

Body

Cookies

Headers (7)

Test Results

201 Created

22 ms

441 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

{

"discount": 0,

"id": 3,

"name": "Клавиатура 80 клавиш",

"category": "Периферия",

"price": 3500,

"stock": 15,

"updatedAt": "2024-06-13T20:36:56.991Z",

"createdAt": "2024-06-13T20:36:56.991Z"

}



# Поиск по id

HTTP

docker check / get

Save

Share

PUT

http://localhost/products/2

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Auth Type

Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#).

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body

Cookies

Headers (7)

Test Results

200 OK 18 ms 436 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

1 {

2     "id": 2,

3     "name": "Клавиатура 80 клавиш",

4     "category": "Периферия",

5     "price": 3500,

6     "stock": 15,

7     "discount": 0,

8     "createdAt": "2024-06-10T18:38:16.316Z",

9     "updatedAt": "2024-06-10T18:38:16.316Z"

10 }

## Поиск

HTTP docker check / search Save Share

GET ▼ http://localhost/products/search?query=Периферия Send ▼

Params ● Authorization Headers (6) Body Scripts Settings Cookies

Query Params

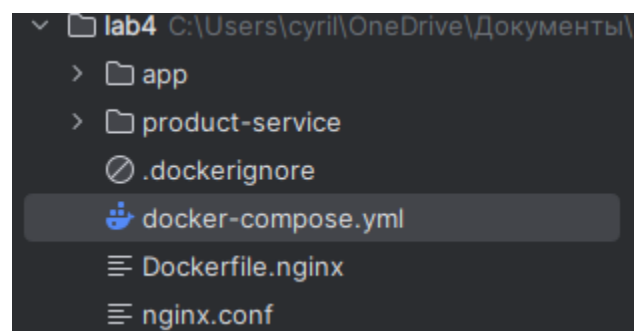
<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	query	Периферия			
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK 5 ms 843 B Save as example ...

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 [
2   {
3     "id": 1,
4     "name": "Клавиатура 80 клавиш",
5     "category": "Периферия",
6     "price": 3500,
7     "stock": 15,
8     "discount": 0,
9     "createdAt": "2024-06-10T17:58:35.088Z",
10    "updatedAt": "2024-06-10T17:58:35.088Z"
11  },
12  {
13    "id": 2,
14    "name": "Клавиатура 80 клавиш",
15    "category": "Периферия",
16    "price": 3500,
17    "stock": 15,
18    "discount": 0,
19    "createdAt": "2024-06-10T18:38:16.316Z",
20    "updatedAt": "2024-06-10T18:38:16.316Z"
21  }
22 ]
```

## Структура проекта:



**Вывод:**

В рамках лабораторной работы №4 было успешно упаковано приложение в Docker-контейнеры и настроено сетевое взаимодействие между его компонентами. Также было обеспечено общение микросервисов через Nginx, используя Docker Compose для оркестрации, а Docker Swarm использован для повышения масштабируемости и надежности системы.