

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-энд разработка

Отчет
Домашнее задание №4

Выполнил:
Горбатов Дмитрий Алексеевич
гр. К33402

Преподаватель:
Добряков Давид Ильич

Задание

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.

Ход работы

Docker-compose.yml:

```
🐳 docker-compose.yml
1  version: "3.9"
2  services:
3    rabbitmq:
4      image: rabbitmq:3-management
5      ports:
6        - "5672:5672"
7        - "15672:15672"
8      networks:
9        - app-network
10
11    auth:
12      build:
13        context: ./auth
14        dockerfile: Dockerfile
15      ports:
16        - "8000:8000"
17      depends_on:
18        - rabbitmq
19      environment:
20        PORT: 8000
21        RABBITMQ_HOST: rabbitmq
22        RABBITMQ_PORT: 5672
23      command: ["wait-for-it", "rabbitmq:5672", "--", "npm", "start"]
24      networks:
25        - app-network
26
27    main:
```

```
26
27   main:
28     build:
29       context: ./main
30       dockerfile: Dockerfile
31     ports:
32       - "8001:8001"
33     depends_on:
34       - auth
35     environment:
36       PORT: 8001
37       AUTH_SERVICE: http://auth:8000
38     networks:
39       - app-network
40
41   gateway:
42     build:
43       context: ./gateway
44       dockerfile: Dockerfile
45     ports:
46       - "2000:2000"
47     depends_on:
48       - auth
49       - main
50     environment:
51       PORT: 2000
52       AUTH_SERVICE: http://auth:8000
53     networks:
54       - app-network
55
56   networks:
57     app-network:
58       driver: bridge
59
```

Dockerfile в микросервисе криптобиржи:

```
main > Dockerfile > ...
1 FROM node:22
2 WORKDIR /usr/src/app
3
4 COPY package*.json ./
5
6 RUN npm install
7
8 COPY . .
9
10 RUN npm run build
11
12 EXPOSE 8001
13
14 CMD ["npm", "start"]
```

Dockerfile в микросервисе аутентификации:

```
auth > Dockerfile > FROM
1 FROM node:22
2 WORKDIR /usr/src/app
3
4 COPY package*.json ./
5
6 RUN npm install
7
8 COPY . .
9
10 RUN npm run build
11
12 ADD https://raw.githubusercontent.com/vishnubob/wait-for-it/master/wait-for-it.sh /usr/local/bin/wait-for-it
13 RUN chmod +x /usr/local/bin/wait-for-it
14
15 EXPOSE 8000
16
17 CMD ["npm", "start"]
18
```

Wait-for-it скрипт, который позволяет дождаться запуска rabbitmq.

Dockerfile в gateway, который связывает эти два микросервиса на одном порте:

```
gateway > dockerfile > ...
1 FROM nginx:latest
2
3 COPY config.conf /etc/nginx/nginx.conf
4
5 CMD ["nginx", "-g", "daemon off;"]
6
```

Сам конфиг для gateway:

```
gateway > config.conf
1  worker_processes auto;
2
3  events {
4      worker_connections 1024;
5  }
6
7  http {
8      server {
9          listen 2000;
10
11         location /getId {
12             internal;
13             proxy_pass      http://auth:8000/users/current;
14             proxy_redirect  off;
15             proxy_set_header Host $host;
16             proxy_set_header X-Real-IP $remote_addr;
17             proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
18             proxy_set_header X-Forwarded-Host $server_name;
19             proxy_set_header Content-Length "";
20             proxy_pass_request_body off;
21         }
22
23         location /auth {
24             rewrite ^/auth/(.*) /$1 break;
25
26             proxy_pass      http://auth:8000;
27             proxy_redirect  off;
28             proxy_set_header Host $host;
29             proxy_set_header X-Real-IP $remote_addr;
30             proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
31             proxy_set_header X-Forwarded-Host $server_name;
32         }
33
34         location /main {
35             auth_request /getId;
36             auth_request_set $user_id $sent_http_user_id;
37
38             rewrite ^/main/(.*) /$1 break;
39
40             proxy_pass      http://main:8001;
41             proxy_redirect  off;
42             proxy_set_header Host $host;
43             proxy_set_header X-Real-IP $remote_addr;
44             proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
45             proxy_set_header X-Forwarded-Host $server_name;
46             proxy_set_header User-Id $user_id;
47         }
48     }
49 }
50
```

Он позволяет взаимодействовать между собой микросервисам.

<input type="checkbox"/>	NAME	STARTED	STATUS	
<input type="checkbox"/>	4 4 containers		running (4/4)	Open
	rabbitmq:3-management 992699038fcc (4_rabbitmq_1)	about 1 hour	running	
	4_auth 32a3268ea97f (4_auth_1)	about 1 hour	running	
	4_main 0ae8ca3ffb13 (4_main_1)	about 1 hour	running	
	4_gateway 7f4598b5f516 (4_gateway_1)	about 1 hour	running	

Вывод:

Упаковал приложение в docker-контейнеры и обеспечил сетевое взаимодействие между различными частями приложения.