

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-энд разработка

Отчет

Лабораторная работа 4: Docker, docker compose, docker
swarm

Выполнил:

Скороходова Елена

Группа
K33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Задачи

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.

Ход работы

Сначала создаем Dockerfile для микросервисов

```
TMO-ICT-Backend-2024 > labs > K33392 > Skorokhodova_Elena
1  FROM node:18
2
3  WORKDIR /app
4
5  COPY package.json package-lock.json ./
6
7
8  RUN npm install
9
10 COPY . .
11
12 CMD ["node", "dist/app.js"]
13
```

Также создаем docker compose

```

1  version: '3.8'
2  services:
3    rabbitmq:
4      image: rabbitmq:3-management
5      ports:
6        - "5672:5672"
7        - "15672:15672"
8      networks:
9        - mynetwork
10
11    authorize:
12      build:
13        context: ./authorize
14      volumes:
15        - ./authorize/dist:/app
16      depends_on:
17        - rabbitmq
18      networks:
19        - mynetwork
20      environment:
21        - RABBITMQ_HOST=rabbitmq
22      working_dir: /app
23      command: node app.js
24
25    otherfunctions:
26      build:
27        context: ./otherfunctions
28      volumes:
29        - ./otherfunctions/dist:/app
30      depends_on:
31        - rabbitmq
32      networks:
33        - mynetwork
34      environment:
35        - RABBITMQ_HOST=rabbitmq
36      working_dir: /app
37      command: node app.js

```

В config микросервисов создаем rabbitmq

```

1  import amqp from 'amqplib/callback_api';
2  import dotenv from 'dotenv';
3
4  dotenv.config();
5
6  const rabbitmqHost = process.env.RABBITMQ_HOST || 'localhost';
7
8  amqp.connect(`amqp://${rabbitmqHost}`, function(error0, connection) {
9    if (error0) {
10      throw error0;
11    }
12    connection.createChannel(function(error1, channel) {
13      if (error1) {
14        throw error1;
15      }
16
17      const queue = 'authorize';
18
19      channel.assertQueue(queue, {
20        durable: false
21      });
22
23      channel.sendToQueue(queue, Buffer.from('готово'));
24      console.log(" [x] Sent 'готово'");
25    });
26  });
27

```

```

import amqp from 'amqplib/callback_api';
import dotenv from 'dotenv';

dotenv.config();

const rabbitmqHost = process.env.RABBITMQ_HOST || 'localhost';

amqp.connect(`amqp://${rabbitmqHost}`, function(error0, connection) {
  if (error0) {
    throw error0;
  }
  connection.createChannel(function(error1, channel) {
    if (error1) {
      throw error1;
    }

    const queue = 'otherFunctions';

    channel.assertQueue(queue, {
      durable: false
    });

    channel.sendToQueue(queue, Buffer.from('готово'));
    console.log(" [x] Sent 'готово'");
  });
});

```

Далее собираем и запускаем контейнеры

