

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэкенд разработка

Отчет

Лабораторная работа №3: Микросервисная архитектура

Выполнила:

Еремеева Арина

Группа К33412

Проверил:

Добряков Д. И.

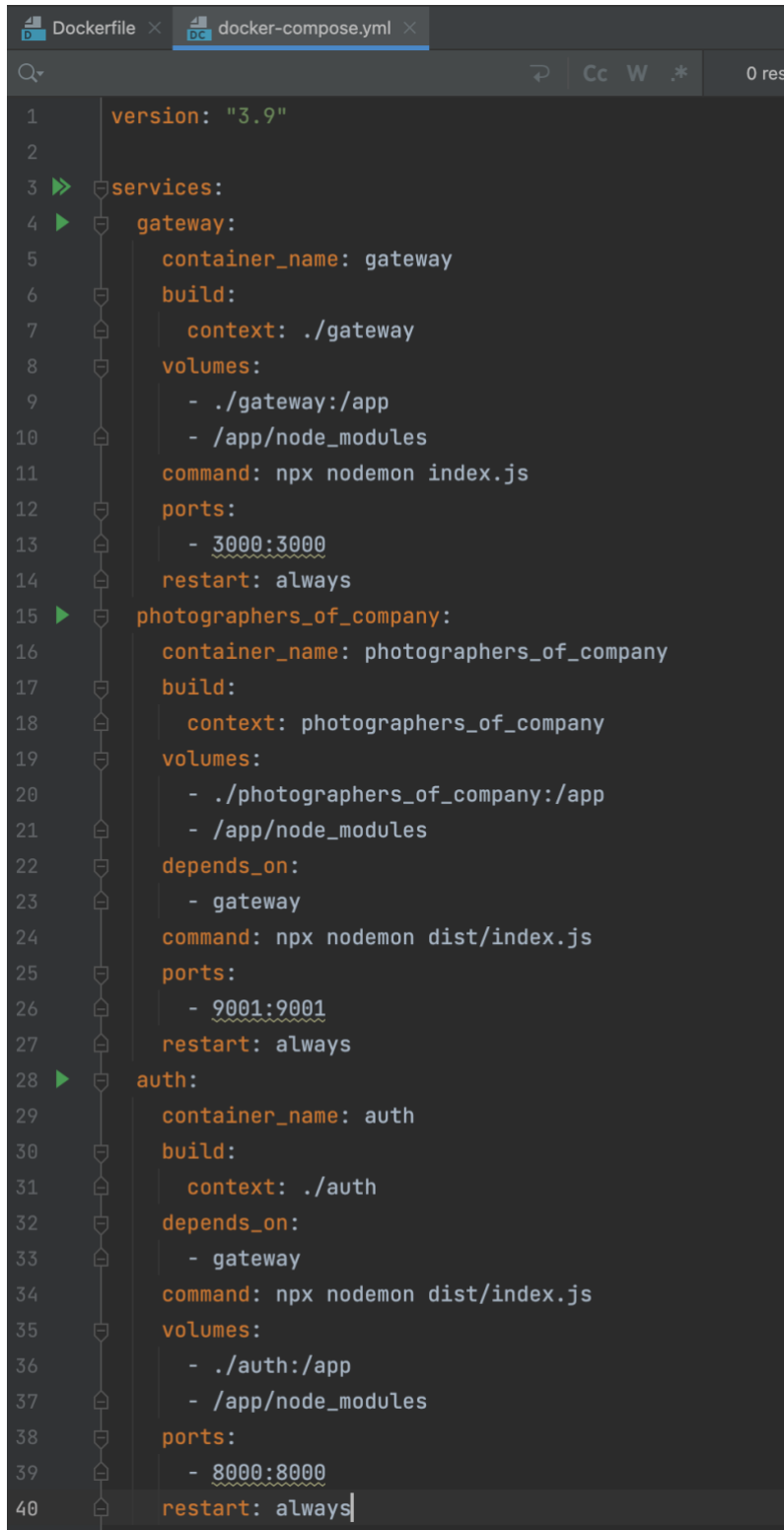
Санкт-Петербург
2023 г.

Цель: необходимо упаковать ваше приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения.

Ход работы:

Для выполнения работы я создала следующие файлы:

Docker-compose.yml:



```
1 version: "3.9"
2
3 services:
4   gateway:
5     container_name: gateway
6     build:
7       context: ./gateway
8     volumes:
9       - ./gateway:/app
10      - /app/node_modules
11     command: npx nodemon index.js
12     ports:
13       - 3000:3000
14     restart: always
15   photographers_of_company:
16     container_name: photographers_of_company
17     build:
18       context: photographers_of_company
19     volumes:
20       - ./photographers_of_company:/app
21       - /app/node_modules
22     depends_on:
23       - gateway
24     command: npx nodemon dist/index.js
25     ports:
26       - 9001:9001
27     restart: always
28   auth:
29     container_name: auth
30     build:
31       context: ./auth
32     depends_on:
33       - gateway
34     command: npx nodemon dist/index.js
35     volumes:
36       - ./auth:/app
37       - /app/node_modules
38     ports:
39       - 8000:8000
40     restart: always
```

Dockerfile для auth:

```
index.js x docker-compose.yml x Dockerfile x
1  >> FROM node:16-alpine
2
3  WORKDIR /app
4
5  COPY package*.json ./
6
7  COPY . .
8
9  RUN npm install
10
11 EXPOSE 8000
12
13 CMD ["make", "start"]
```

Dockerfile для photographers_of_company:

```
Dockerfile x
1  >> FROM node:16-alpine
2
3  WORKDIR /app
4
5  COPY package*.json ./
6
7  COPY . .
8
9  RUN npm install
10
11 EXPOSE 9001
12
13 CMD ["make", "start"]
```

Dockerfile для gateway:

```
Dockerfile x
1  FROM node:16-alpine
2
3  WORKDIR /app
4
5  COPY package*.json ./
6
7  COPY . .
8
9  RUN npm install
10
11  EXPOSE 3000
12
13  CMD ["make", "start"]
```

Далее запустила docker-compose up

```
arinaeremeeva@Anta lr3 % docker-compose up
[+] Building 34.2s (24/24) FINISHED
=> [gateway internal] load build definition from Dockerfile
=> => transferring dockerfile: 157B
=> [gateway internal] load .dockerignore
=> => transferring context: 2B
=> [photographers_of_company internal] load metadata for docker.io/library/node:16-alpine
=> [photographers_of_company 1/5] FROM docker.io/library/node:16-alpine@sha256:296f32946d2fb5227e43fa51e608bbb11a208d648a833e9b264934fb0c7f404b
=> => resolve docker.io/library/node:16-alpine@sha256:296f32946d2fb5227e43fa51e608bbb11a208d648a833e9b264934fb0c7f404b
=> => sha256:fa670e2ca24696a81fa18278979804f501ad468af1dd93bdfd217493b5ef52d6 36.63MB / 36.63MB
=> => sha256:5307f402fec108cb9d6e29cc1cc2dd864e30cef3860dfdb6a6e2cae0d5e2a527 2.34MB / 2.34MB
=> => sha256:296f32946d2fb5227e43fa51e608bbb11a208d648a833e9b264934fb0c7f404b 1.43kB / 1.43kB
=> => sha256:f711d8a40d3515d7d44e344306382179fc8bfc4fe75f1a77b27a686a88649430 1.16kB / 1.16kB
=> => sha256:4874253f493bd62f59b8ca308ecc1c6a8c7af30a16e2c814d17998a5925dfe59 6.73kB / 6.73kB
=> => sha256:31e352740f534f9ad170f75378a84fe453d6156e40700b882d737a8f4a6988a3 3.40MB / 3.40MB
=> => extracting sha256:31e352740f534f9ad170f75378a84fe453d6156e40700b882d737a8f4a6988a3
=> => sha256:e3da03619ad76ffbd4bfd84c73dc89156840e122b1330ef999d2bcb99e45c5e2 450B / 450B
=> => extracting sha256:fa670e2ca24696a81fa18278979804f501ad468af1dd93bdfd217493b5ef52d6
=> => extracting sha256:5307f402fec108cb9d6e29cc1cc2dd864e30cef3860dfdb6a6e2cae0d5e2a527
=> => extracting sha256:e3da03619ad76ffbd4bfd84c73dc89156840e122b1330ef999d2bcb99e45c5e2
=> [gateway internal] load build context
=> => transferring context: 5.17MB
=> CACHED [photographers_of_company 2/5] WORKDIR /app
=> [gateway 3/5] COPY package*.json ./
=> [gateway 4/5] COPY . .
=> [gateway 5/5] RUN npm install
=> [gateway] exporting to image
=> => exporting layers
=> => writing image sha256:0c49c3a6776cb7f881d6f1c4c2519638f1ece70d2b76b24f9ba0b884bd23a0b6
=> => naming to docker.io/library/lr3-gateway
=> [auth internal] load .dockerignore
```

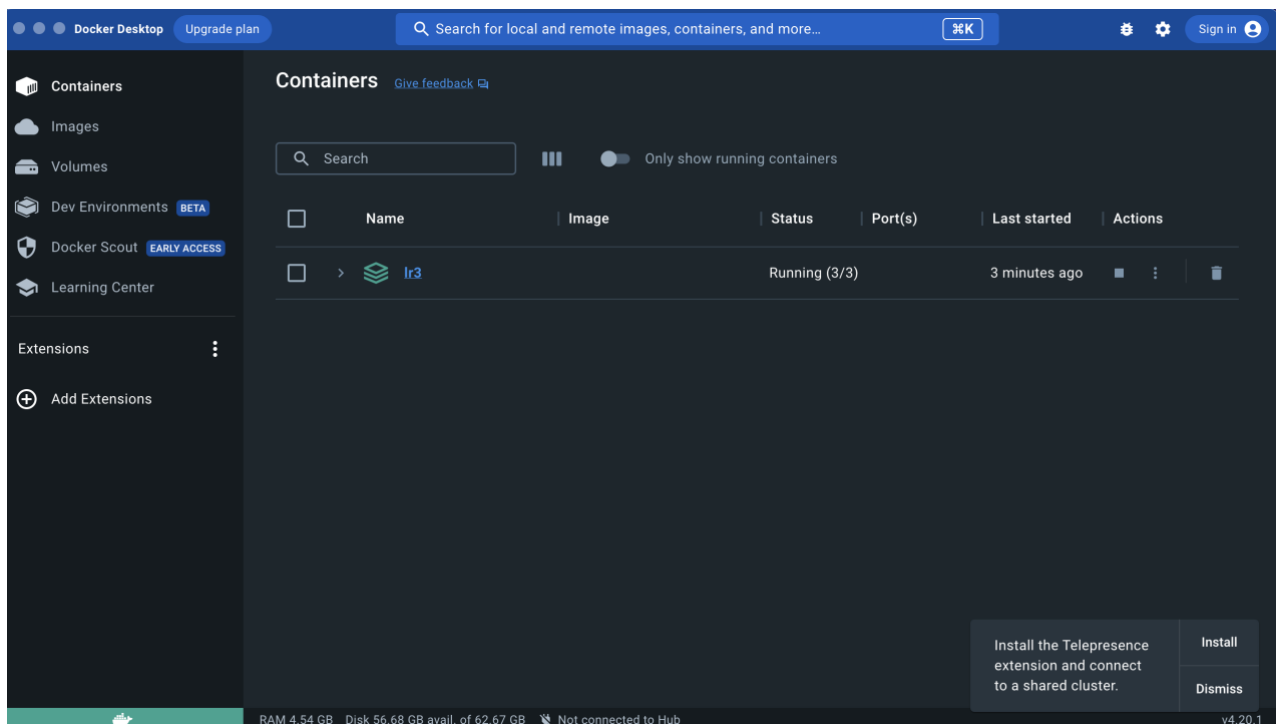
```

=> [auth] exporting to image
=> => exporting layers
=> => writing image sha256:617328e95d8e8c6d5a12850e04af9155ab07af71d1f96a7e5a5495ce2bbffffa
=> => naming to docker.io/library/lr3-auth
[+] Running 4/4
 ✓ Network lr3_default          Created
 ✓ Container gateway           Created
 ✓ Container auth               Created
 ✓ Container photographers_of_company Created
Attaching to auth, gateway, photographers_of_company
gateway          | [nodemon] 2.0.22
gateway          | [nodemon] to restart at any time, enter `rs`
gateway          | [nodemon] watching path(s): *.*
gateway          | [nodemon] watching extensions: js,mjs,json
gateway          | [nodemon] starting `node index.js`
gateway          | Gateway listening at http://localhost:3000
photographers_of_company | [nodemon] 2.0.22
photographers_of_company | [nodemon] to restart at any time, enter `rs`
photographers_of_company | [nodemon] watching path(s): src/**/*
photographers_of_company | [nodemon] watching extensions: ts
photographers_of_company | [nodemon] starting `ts-node ./src/index.ts dist/index.js`
auth              | [nodemon] 2.0.22
auth              | [nodemon] to restart at any time, enter `rs`
auth              | [nodemon] watching path(s): src/**/*
auth              | [nodemon] watching extensions: ts
auth              | [nodemon] starting `ts-node ./src/index.ts dist/index.js`
photographers_of_company | /app/node_modules/sequelize/src/dialects/abstract/connection-manager.js:81
photographers_of_company |     throw new Error(`Please install ${moduleName} package manually`);
photographers_of_company |           ^
photographers_of_company | Error: Please install sqlite3 package manually
photographers_of_company |     at ConnectionManager._loadDialectModule (/app/node_modules/sequelize/src/dialects/abstract/connection-manager.js:81:15

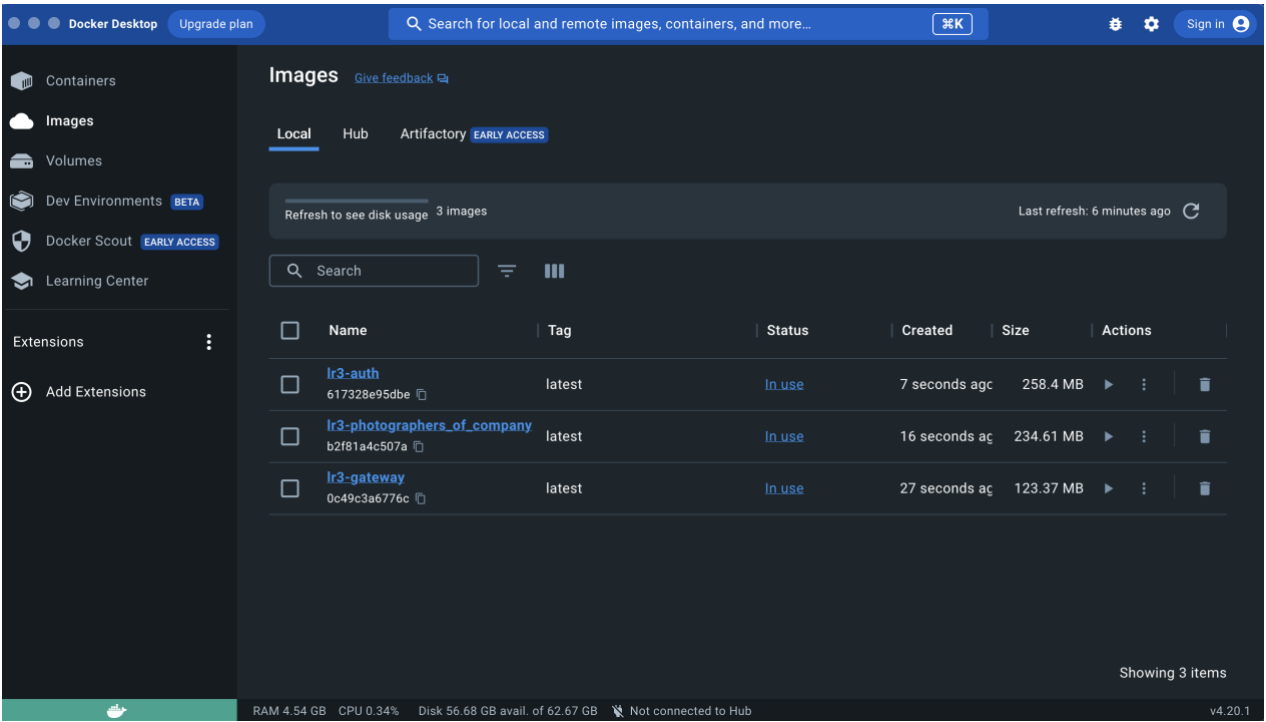
```

В приложении Docker создались и запустились:

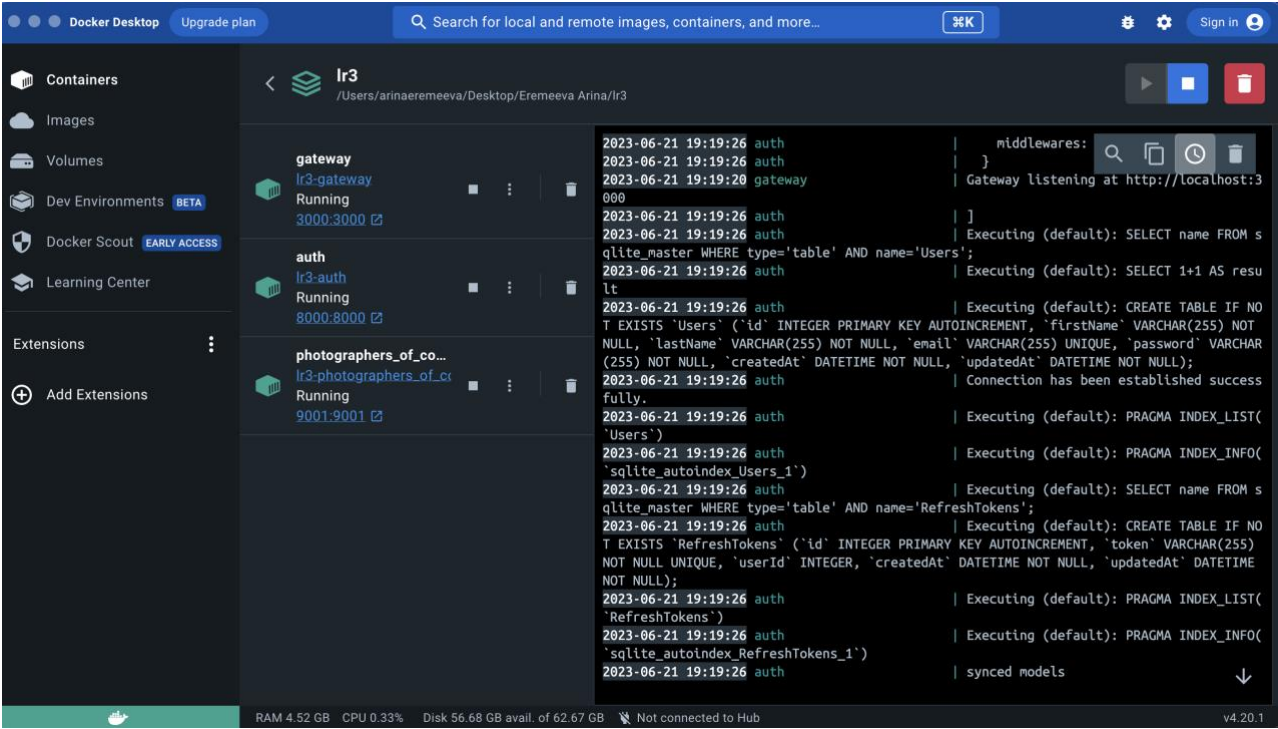
1) Контейнер:



2) Представления:



3) Порты для микросервисов и gateway:



Вывод

В данной лабораторной работе я развернула приложение с помощью docker-контейнеров и обеспечила сетевое взаимодействие между ними.