

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Лабораторная работа 4  
Упаковка приложения в Docker

Выполнила:

Хорошкеева Ксения

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2022 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

Dockerfile основного сервиса:

```
1 FROM node:18-alpine as builder
2   ENV NPM_CONFIG_PREFIX=/home/node/.npm-global
3   ENV PATH=$PATH:/home/node/.npm-global/bin
4   RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
5   WORKDIR /home/node/app
6   COPY --chown=node:node package*.json ./
7   USER node
8   RUN npm config set unsafe-perm true
9   RUN npm install -g typescript
10  RUN npm install -g ts-node
11  RUN npm install
12  COPY --chown=node:node . .
13  RUN npm run build
14
15 FROM node:18-alpine
16  ENV NODE_ENV=production
17  RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
18  WORKDIR /home/node/app
19  COPY --chown=node:node package*.json ./
20  USER node
21  RUN npm install --production
22  COPY --from=builder /home/node/app/dist ./dist
23  COPY --chown=node:node src/configs/settings.ini ./dist/configs/settings.ini
24  COPY --chown=node:node src/configs/db.js ./dist/configs/db.js
25  COPY --chown=node:node .sequelizerc ./dist/
26  COPY --chown=node:node src/seeder/*.* ./dist/seeder/
27  EXPOSE 8000
28  CMD [ "node", "dist/index.js" ]
```

Dockerfile сервиса бронирования:

```

1  ➤ FROM node:18-alpine as builder
2  ENV NPM_CONFIG_PREFIX=/home/node/.npm-global
3  ENV PATH=$PATH:/home/node/.npm-global/bin
4  RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
5  WORKDIR /home/node/app
6  COPY --chown=node:node package*.json ./
7  USER node
8  RUN npm config set unsafe-perm true
9  RUN npm install -g typescript
10 RUN npm install -g ts-node
11 RUN npm install
12 COPY --chown=node:node . .
13 RUN npm run build
14
15 FROM node:18-alpine
16 ENV NODE_ENV=production
17 RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
18 WORKDIR /home/node/app
19 COPY --chown=node:node package*.json ./
20 USER node
21 RUN npm install --production
22 COPY --from=builder /home/node/app/dist ./dist
23 COPY --chown=node:node src/configs/settings.ini ./dist/configs/settings.ini
24 EXPOSE 8001
25 CMD [ "node", "dist/index.js" ]

```

Файл docker-compose.yml:

```
1   version: '3.9'
2   >> services:
3   > db_main:
4       image: mysql
5       environment:
6         - MYSQL_USER=mysql
7         - MYSQL_PASSWORD=mysql
8         - MYSQL_DATABASE=mysql
9         - MYSQL_ROOT_PASSWORD=mysql
10      expose:
11        - 3306
12      volumes:
13        - data_main:/var/lib/mysql
14      healthcheck:
15        test: [ "CMD", "mysqladmin" ,"ping", "-h", "localhost" ]
16        timeout: 10s
17        retries: 15
18
19 > db_booking:
20     image: mysql
21     environment:
22       - MYSQL_USER=mysql
23       - MYSQL_PASSWORD=mysql
24       - MYSQL_DATABASE=mysql
25       - MYSQL_ROOT_PASSWORD=mysql
26     expose:
27       - 3306
28     volumes:
29       - data_booking:/var/lib/mysql
30     healthcheck:
31       test: [ "CMD", "mysqladmin" ,"ping", "-h", "localhost" ]
32       timeout: 10s
33       retries: 15
```

```

34
35 ▶ main:
36   build:
37     context: main
38   ports:
39     - "8000:8000"
40   depends_on:
41     db_main:
42       condition: service_healthy
43
44 ▶ booking:
45   build:
46     context: booking
47   depends_on:
48     db_booking:
49       condition: service_healthy
50
51 volumes:
52   data_main:
53   data_booking:

```

## Проверка:

GET localhost:8000/api/rooms?city=Par&capacity=2 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results Status: 200 OK Time: 2.78 s Size: 730 B Save Response

Pretty Raw Preview Visualize JSON Copy

```

1 {
2   {
3     "id": 1,
4     "name": "Super deluxe",
5     "capacity": 2,
6     "price": 10000,
7     "hotel": {
8       "id": 1,
9       "name": "Hotel A",
10      "city": "Paris",
11      "stars": 5
12    }
13  },
14  {
15    "id": 2,
16    "name": "Deluxe",

```

POST localhost:8000/api/bookings/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

1 {"roomId": 1, "startDate": "2022-07-01", "endDate": "2022-07-11"} Copy

Body Cookies Headers (8) Test Results Status: 200 OK Time: 355 ms Size: 334 B Save Response

Pretty Raw Preview Visualize JSON Copy

```

1 {
2   "id": 1,
3   "startDate": "2022-07-01",
4   "endDate": "2022-07-11",
5   "roomId": 1
6 }

```

## **Вывод**

Я упаковала приложение в контейнеры Docker с использованием утилиты Docker-compose.