

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №4
Docker, docker compose, docker swarm

Выполнила:
Зайцева А. А.
Группа К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2023 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

Прежде всего для выполнения данной лабораторной работы были скопированы файлы основного приложения и микросервиса из лабораторной работы №3.

Затем, для основного приложения и микросервиса были созданы соответствующие им файлы **Dockerfile**.

Содержимое **main/Dockerfile**:

```
# syntax=docker/dockerfile:1
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN npm install
RUN npm rebuild
RUN npm run build
RUN cp ./src/config/settings.ini ./dist/config
RUN npm run migrate
RUN npm run seed
CMD ["npm", "start"]
EXPOSE 8000
```

В **Dockerfile** приложения **main** не используется утилита **make**, т. е. она не включена в базовый образ **node:18-alpine**.

Содержимое **stat-service/Dockerfile**:

```
# syntax=docker/dockerfile:1
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN npm install
RUN npm run build
CMD ["npm", "start"]
EXPOSE 8008
```

Из исходного кода микросервиса **stat-service** было исключено использование пакета **dotenv** и **.env** файла, т. к. при использовании **Docker** переменные окружения можно удобным образом задать в конфигурационных файлах **Docker**.

Затем, в корне директории лабораторной работы был создан файл **docker-compose.yaml**.

Содержимое **docker-compose.yaml**:

```
services:
  main:
    build: ./main
    ports:
      - "8000:8000"
  stat:
    build: ./stat-service
    ports:
      - "8008:8008"
    environment:
      MAIN_HOST: main
      MAIN_PORT: 8000
```

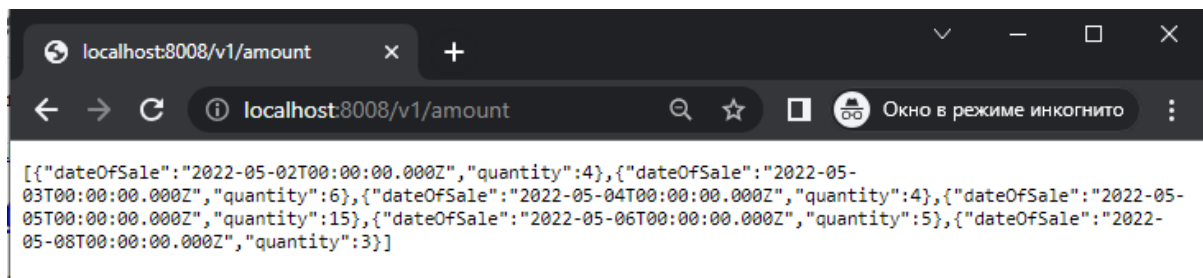
Сетевое взаимодействие обеспечивается автоматически посредством привязки имен хостов **main** и **stat** внутри контейнеров к соответствующим им контейнерам.

Запуск приложений осуществляется с помощью команды **docker compose up**.

```
~/p/study/6/ITMO-ICT-Backend-2023/labs/K33402/AZaytseva/4 lab4 !3 ?3 docker compose up
[+] Building 40.5s (23/23) FINISHED
=> [main internal] load build definition from Dockerfile
=> => transferring dockerfile: 276B
=> [main internal] load .dockerignore
=> => transferring context: 2B
```

```
4-main-1 | [nodemon] watching extensions: ts
4-main-1 | [nodemon] starting `ts-node ./src/index.ts dist/index.js`
4-stat-1 | Running server on port 8008
4-main-1 | Running server on port 8000
4-main-1 | Executing (default): SELECT name FROM sqlite_master WHERE type='table' AND name='Users';
4-main-1 | Executing (default): SELECT 1+1 AS result
4-main-1 | Executing (default): PRAGMA INDEX_LIST('Users')
```

Проверка работы микросервиса статистики:



Вывод

В ходе лабораторной работы была изучена платформа Docker, позволяющая создавать и запускать приложения в контейнерах без привязки к конкретному окружению запуска. Были изучены основы написания файлов Dockerfile и docker-compose.yaml, а также команды docker build, docker run, docker compose. Для сервисов из лабораторной работы №3 были сконфигурированы и успешно протестированы контейнеры.