

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-энд разработка

Отчет

Лаборатория работа 4

Выполнил:

Ле Тхи Лан Ань

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

Customer/Dockerfile

```
FROM node:16

WORKDIR /app/customer

COPY package.json .

RUN npm install

COPY . .

EXPOSE 8001

CMD ["npm", "start"]
```

Booking/Dockerfile

```
FROM node

WORKDIR /app/Booking

COPY package.json .

RUN npm install
```

```
COPY . .
```

```
EXPOSE 8003
```

```
CMD ["npm", "start"]
```

docker-compose.yml

```
version: '3'
```

```
services:
```

```
  nosql-db1:
```

```
    image: mongo
```

```
    ports:
```

```
      - "27017:27017"
```

```
    container_name: nosql-db
```

```
    volumes:
```

```
      - ./db:/data/db
```

```
  booking:
```

```
    build:
```

```
      dockerfile: Dockerfile
```

```
      context: ./Booking
```

```
    container_name: Booking
```

```
    ports:
```

```
      - "8003:8003"
```

```
restart: always

depends_on:

  - "nosql-db1"

volumes:

  - ../app

  - /app/Booking/node_modules

customer:

  build:

    dockerfile: Dockerfile

    context: ./customer

  container_name: customer

  ports:

    - "8001:8001"

  restart: always

  depends_on:

    - "nosql-db1"

  volumes:

    - ../app

    - /app/customer/node_modules

nginx-proxy:

  build:

    dockerfile: Dockerfile

    context: ./proxy
```

```
depends_on:

  - booking

  - customer

ports:

  - "80:80"
```

proxy/Dockerfile

```
FROM nginx

RUN rm /etc/nginx/nginx.conf

COPY nginx.conf /etc/nginx/nginx.conf
```

proxy/nginx.conf

```
worker_processes 3;

events { worker_connections 1024; }

http {

    server {

        listen 80;

        charset utf-8;
```

```

location ~ ^/booking {

    rewrite ^/boooing/(.*) /$1 break;

    proxy_pass http://booking:8003;

    proxy_http_version 1.1;

    proxy_set_header Upgrade $http_upgrade;

    proxy_set_header Connection 'Upgrade';

    proxy_set_header Host $host;

    proxy_cache_bypass $http_upgrade;

}

location ~ ^/customer {

    rewrite ^/customer/(.*) /$1 break;

    proxy_pass http://customer:8001;

    proxy_http_version 1.1;

    proxy_set_header Upgrade $http_upgrade;

    proxy_set_header Connection 'Upgrade';

    proxy_set_header Host $host;

    proxy_cache_bypass $http_upgrade;

}

}

```

Вывод

Научитесь использовать программное обеспечение docker,
познакомьтесь с docker-compose