

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-энд разработка

Отчет

Лабораторная работа №4: Docker, docker compose

Выполнил:

Жигалова Анастасия
K33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

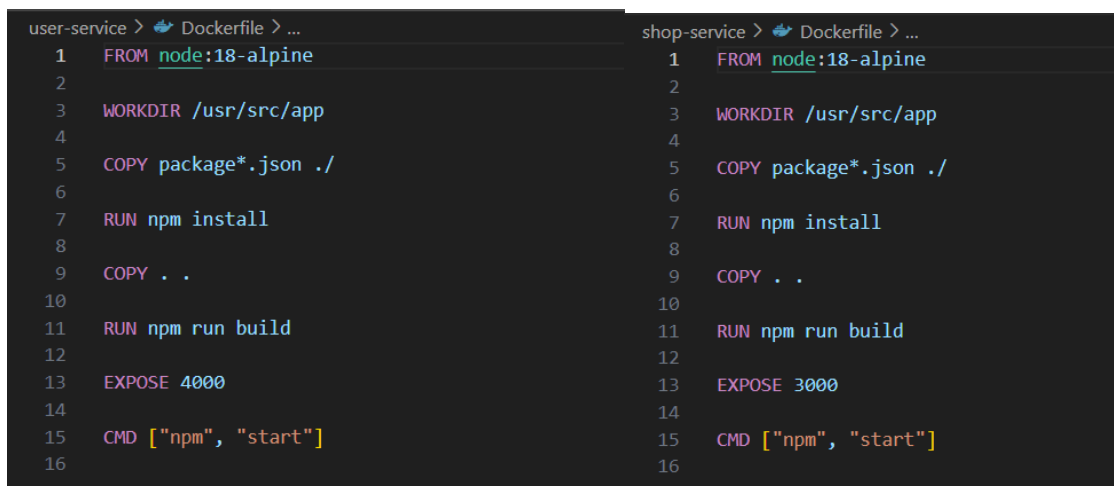
Задача

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.

Вариант : Сервис для работы с магазином одежды. Требуемый функционал: регистрация, авторизация, создание профиля, работа с товарами, просмотр количества единиц товара, управление скидками и акциями, работа с базой клиентов.

Ход работы

Создадим Dockerfile в микросервисах



The image shows a side-by-side comparison of two Dockerfiles. The left Dockerfile is for 'user-service' and the right is for 'shop-service'. Both are based on 'node:18-alpine' and follow a similar structure: setting the working directory to '/usr/src/app', copying package files, installing npm dependencies, running a build, and exposing a port. The only difference is the port exposed: 4000 for user-service and 3000 for shop-service.

user-service > Dockerfile > ...	shop-service > Dockerfile > ...
1 FROM node:18-alpine	1 FROM node:18-alpine
2	2
3 WORKDIR /usr/src/app	3 WORKDIR /usr/src/app
4	4
5 COPY package*.json ./	5 COPY package*.json ./
6	6
7 RUN npm install	7 RUN npm install
8	8
9 COPY . .	9 COPY . .
10	10
11 RUN npm run build	11 RUN npm run build
12	12
13 EXPOSE 4000	13 EXPOSE 3000
14	14
15 CMD ["npm", "start"]	15 CMD ["npm", "start"]
16	16

Создадим gateway с помощью nginx в файле nginx.conf и Dockerfile для него

```

deployment > gateway > src > conf > nginx.conf
1
2 server {
3     listen 80;
4
5     location /shop {
6         proxy_pass http://shop-service:3000;
7         proxy_set_header Host $host;
8         proxy_set_header X-Real-IP $remote_addr;
9         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
10        proxy_set_header X-Forwarded-Proto $scheme;
11    }
12
13    location /user {
14        proxy_pass http://user-service:4000;
15        proxy_set_header Host $host;
16        proxy_set_header X-Real-IP $remote_addr;
17        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
18        proxy_set_header X-Forwarded-Proto $scheme;
19    }
20 }
21

```

```

deployment > gateway > src > Dockerfile > ...
1 FROM nginx:alpine
2
3 COPY ./conf/nginx.conf /etc/nginx/nginx.conf
4

```

Создадим сервисы для баз данных, микросервисов в docker-compose

```

version: '1.0'

services:
  shop-service:
    build:
      context: ./shop-service
      dockerfile: Dockerfile
    ports:
      - "3000:3000"
    depends_on:
      - db_shop
      - rabbitmq
    environment:
      DB_HOST: db_shop
      DB_PORT: 5432
      DB_USER: shop_user
      DB_PASSWORD: shop_password
      DB_NAME: shop_db

  user-service:
    build:
      context: ./user-service
      dockerfile: Dockerfile
    ports:
      - "4000:4000"
    depends_on:
      - db_user
      - rabbitmq
    environment:
      DB_HOST: db_user
      DB_PORT: 5432
      DB_USER: user_user
      DB_PASSWORD: user_password
      DB_NAME: user_db

```

```

gateway:
  build:
    context: ../deployment/gateway/src
    dockerfile: Dockerfile
  ports:
    - "80:80"
  depends_on:
    - shop-service
    - user-service

db_shop:
  image: postgres:13
  environment:
    POSTGRES_USER: shop_user
    POSTGRES_PASSWORD: shop_password
    POSTGRES_DB: shop_db
  ports:
    - "5432:5432"

db_user:
  image: postgres:13
  environment:
    POSTGRES_USER: user_user
    POSTGRES_PASSWORD: user_password
    POSTGRES_DB: user_db
  ports:
    - "5433:5432"

rabbitmq:
  image: rabbitmq:3-management
  ports:
    - "5672:5672"
    - "15672:15672"

networks:
  default:

```

Containers

Give feedback

Container CPU usage

0.58% / 800% (8 CPUs available)

Container memory usage

238.11MB / 3.66GB

Show charts

Search

Only show running containers

	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<div>lr4</div>		Running (4/6)		0.58%	49 minutes ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<div>db_user-1</div> <div>a2e45082d404</div>	postgres:13	Exited (1)	5433:5432	0%	49 minutes ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<div>db_shop-1</div> <div>03d6cfef90c6</div>	postgres:13	Exited (1)	5432:5432	0%	49 minutes ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<div>rabbitmq-1</div> <div>6b6643616fb3</div>	rabbitmq:3-management	Running	15672:15672 Show all ports (2)	0.58%	49 minutes ago	<div></div> <div></div> <div></div>

Showing 14 items

```

[+] Running 6/6
  Container lr4-db_user-1      Created
  Container lr4-rabbitmq-1     Created
  Container lr4-db_shop-1      Created
  Container lr4-shop-service-1 Created
  Container lr4-user-service-1 Created
  Container lr4-gateway-1      Created
Attaching to db_shop-1, db_user-1, gateway-1, rabbitmq-1, shop-service-1, user-service-1
db_user-1 | exec /usr/local/bin/docker-entrypoint.sh: exec format error
db_shop-1 | exec /usr/local/bin/docker-entrypoint.sh: exec format error
db_shop-1 exited with code 1
db_user-1 exited with code 1
rabbitmq-1 | =INFO REPORT==== 29-May-2024::21:02:20.755395 ===
rabbitmq-1 |     alarm_handler: {set,{system_memory_high_watermark,[]}}
shop-service-1 |
shop-service-1 | > lr2@1.0.0 start
shop-service-1 | > tsc && node dist/index.js
shop-service-1 |
user-service-1 | > lr3@1.0.0 start
user-service-1 | > tsc && node dist/index.js
user-service-1 |
gateway-1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
gateway-1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
gateway-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
gateway-1 | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
gateway-1 | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
gateway-1 | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
gateway-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
gateway-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
gateway-1 | /docker-entrypoint.sh: Configuration complete; ready for start up
user-service-1 | Server is running on port 8081
user-service-1 | Executing (default): SELECT name FROM sqlite_master WHERE type='table' AND name='Users';
user-service-1 | Executing (default): SELECT 1+1 AS result
user-service-1 | Executing (default): PRAGMA INDEX_LIST('Users')
user-service-1 | Connection has been established successfully.
user-service-1 | Executing (default): PRAGMA INDEX_INFO('sqlite_autoindex_Users_1')
user-service-1 | synced models
rabbitmq-1 | 2024-05-29 21:02:26.245272+00:00 [notice] <0.44.0> Application syslog exited with reason: stopped
rabbitmq-1 | 2024-05-29 21:02:26.252385+00:00 [notice] <0.254.0> Logging: switching to configured handler(s); following messages may not be visible in this log output
rabbitmq-1 | 2024-05-29 21:02:26.253165+00:00 [notice] <0.254.0> Logging: configured log handlers are now ACTIVE
rabbitmq-1 | 2024-05-29 21:02:26.266095+00:00 [info] <0.254.0> ra: starting system quorum_queues
rabbitmq-1 | 2024-05-29 21:02:26.266291+00:00 [info] <0.254.0> starting Ra system: quorum_queues in directory: /var/lib/rabbitmq/mnesia/rabbit@6b6643616fb3/quorum/rabbit@6b6643616fb3
shop-service-1 | Server is running on port 8080
shop-service-1 | Executing (default): SELECT name FROM sqlite_master WHERE type='table' AND name='Discounts';
shop-service-1 | Executing (default): SELECT 1+1 AS result
shop-service-1 | Executing (default): PRAGMA INDEX_LIST('Discounts')
shop-service-1 | Connection has been established successfully.
shop-service-1 | Executing (default): SELECT name FROM sqlite_master WHERE type='table' AND name='Promotions';
shop-service-1 | Executing (default): PRAGMA INDEX_LIST('Promotions')

```

Вывод

В ходе данной работы было упаковано приложение в dockerконтейнеры и обеспечено сетевое взаимодействие между различными частями приложения с помощью docker-compose и nginx.