

Лабораторная работа №2

Docker

Перед установкой нужно зарегистрироваться на <https://hub.docker.com>

Мой Docker-Hub: <https://hub.docker.com/u/yuraprogrammer>

Задание 0

Установить docker и docker-compose

```
pc-mount@pcmount-Aspire-V5-531G:~/Ruby/MicroProfile$ docker -v; docker-compose -v
Docker version 19.03.6, build 369ce74a3c
docker-compose version 1.25.4, build 8d51620a
```

Задание 1

Описать Dockerfile, позволяющий запускать Rails-приложение в контейнере.

После описания Dockerfile, отдельно запустить в контейнере БД, которая используется в Rails-проекте.

Если не работает, проверить проброс портов в контейнеры.

Запустить контейнер с Rails.

Проверить, что установлен connection между двумя контейнерами.

Описанный DockerFile:

```
1 FROM ruby:2.6.3
2 RUN apt-get update -qq && apt-get install -y nodejs yarn postgresql-client
3 RUN mkdir /myapp
4 WORKDIR /myapp
5 COPY Gemfile /myapp/Gemfile
6 COPY Gemfile.lock /myapp/Gemfile.lock
7 COPY package.json .
8 COPY yarn.lock .
9 RUN bundle check || bundle install
10 COPY . /myapp
11 # Add a script to be executed every time the container starts.
12 COPY entrypoint.sh /usr/bin/
13 RUN chmod +x /usr/bin/entrypoint.sh
14 ENTRYPOINT ["entrypoint.sh"]
15 # Проброс порта
16 EXPOSE 3000
17 # Start the main process.
18 CMD ["rails", "server", "-b", "0.0.0.0"]
```

Строим образ:

```
docker build -t micorprofile .
```

Запуск контейнера БД:

```
docker run -d -it --env POSTGRES_PASSWORD=postgres --env
DB_NAME=MicroProfile_development --name mydbcontainer postgres:12.2
```

До этого был запущен пробный контейнер который был запущен и удалён.

```
docker run --rm --env RAILS_ENV=development --env POSTGRES_USER=postgres --env
POSTGRES_PASSWORD=postgres --link mydbcontainer:db --volume ${PWD}:/app micorprofile
bin/rails db:create db:migrate
```

Запуск контейнера с rails приложением

```
docker run --rm -it --env RAILS_ENV=development --env POSTGRES_USER=postgres --env POSTGRES_PASSWORD=postgres --publish 3000:3000 --volume ${PWD}:/app --link mydbcontainer:db microprofile
```

Проверка соединения :

Так как я не указывал какое будет имя у контейнера с rails-app оно было выбрано рандомно (в данном случае jolly_hugle).

```
docker inspect -f "{{ .HostConfig.Links }}" jolly_hugle.
```

Вывод в консоли:

```
[/mydbcontainer:/jolly_hugle/db]
```

Проверяем при помощи:

docker ps — показывает список запущенных контейнеров.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
10e1b54f9de3	microprofile	"entrypoint.sh rails..."	29 minutes ago	Up 29 minutes	0.0.0.0:3000->3000/tcp	jolly_hugle
89e40b17c82a	postgres:12.2	"docker-entrypoint.s..."	17 hours ago	Up 17 hours	5432/tcp	mydbcontainer

Задание 2

Научиться собирать image на основе описанного Dockerfile

Строим образ:

```
docker build -t micorprofile .
```

microprofile	latest	2f879f88f694	19 hours ago	1.27GB
--------------	--------	--------------	--------------	--------

Задание 3

Запустить собранный образ на hub.docker.com и дать ему открытый доступ.

(Следует обратить внимание – по выполнению каждой из лабораторных работ, надо будет образ каждый раз пересобирать и ставить на него тэг по номеру лабораторной, к примеру, 0.0.2-0.0.8, без буквенных и других символов, только цифры с точками)

В ходе выполнения работы я переименовал образ на yuraprogrammer/microinstagram:0.0.2

```
docker tag old name new name
```

```
docker rmi old name
```

```
docker login -u=myname -p=mypass
```

```
docker push yuraprogrammer/microinstagram
```

Задание 4

Описать docker-compose-1.yml, позволяющий запускать сразу контейнер с БД и приложением. Образ для контейнера с приложением должен быть вашим запущенным в хаб образом.

```
version: '3'
services:
  db:
    image: postgres:12.2
    #volumes:
    # - ./tmp/db:/var/lib/postgresql/data
    ports:
    - "5435:5432"
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
  web:
    image:
      yuraprogrammer/microinstagram:0.0.2
    #build: .
    command: bash -c "rm -f tmp/pids/server.pid && bundle exec rails s -p 3000 -b '0.0.0.0'"
    volumes:
    - ./myapp
    ports:
    - "3000:3000"
    depends_on:
    - db
```

Описать docker-compose-2.yml, позволяющий запускать сразу контейнер с БД и приложением. Образ для контейнера с приложением должен билдиться на основе Dockerfile, который был описан в Задании 1.

```
version: '3'
services:
  db:
    image: postgres:12.2
    volumes:
    - ./tmp/db:/var/lib/postgresql/data
    ports:
    - 5435:5432
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
  web:
    build: .
    command: bash -c "rm -f tmp/pids/server.pid && bundle exec rails s -p 3000 -b '0.0.0.0'"
    volumes:
    - ./myapp
    ports:
    - "3000:3000"
    depends_on:
    - db
```

Вывод:

В ходе выполнения лабораторной работы были изучены основные возможности которые предлагает Docker для контейнеризации приложения со всем его окружением и зависимостями.