

Docker-образы

Микросервисы

Проект *microservices* и проверка ДЗ

Создайте новую ветку в вашей репозитории *microservices* для выполнения данного ДЗ. Т.к. это третье задание, посвященное работе с Docker, то ветку назовите **docker-3**.

Проверка данного ДЗ будет производиться через Pull Request ветки с ДЗ к ветке **master**.

После того, как один из преподавателей сделает approve пул реквеста, ветку с ДЗ можно смерджить.

Цели задания

- Научиться описывать и собирать Docker-образы для сервисного приложения
- Научиться оптимизировать работу с Docker-образами
- Запуск и работа приложения на основе Docker-образов, оценка удобства запуска контейнеров при помощи `docker run`

План

- Разбить наше приложение на несколько компонентов
- Запустить наше микросервисное приложение

Рекомендации к выполнению задания

1. Для выполнения домашнего задания и дальнейшей работы с Docker-образами рекомендуется использовать linter - [ссылка](#)
2. Мы не задаем ограничение на использование линтеров. Если вы используете IDE, можно использовать встроенный плагин для проверки синтаксиса **Dockerfile**
3. Предоставленные примеры можно дополнять следуя обновлениям по документу рекомендуемых практик в написании **Dockerfile** - [ссылка](#)
4. Во всех образах, с которыми мы будем работать в этом задании, *используются неоптимальные инструкции*, обратите на это внимание и постарайтесь исправить их

Подготовка

Подключаемся к ранее созданному Docker host'y (см. предыдущее ДЗ):

```
> docker-machine ls
NAME            ACTIVE  DRIVER  STATE  URL                                     SWARM  DOCKER
docker-host     -       google  Running  tcp://<docker-host-ip>:2376          v17.09.0-
ce

> eval $(docker-machine env docker-host)
```

Скачайте [архив по ссылке](#) и распакуйте его содержимое внутри репозитория

! Не забудьте удалить ZIP-архив после распаковки

Новая структура приложения

Внутри репозитория у нас появится каталог `reddit-microservices`, переименуйте его в `src`

Каталог `src` теперь основной каталог этого домашнего задания

Теперь наше приложение состоит из трех компонентов:

- `post-py` - сервис отвечающий за написание постов
- `comment` - сервис отвечающий за написание комментариев
- `ui` - веб-интерфейс, работающий с другими сервисами

Для работы нашего приложения также требуется база данных MongoDB

Сервис post-py

Создайте файл `./post-py/Dockerfile` следующего содержания ([ссылка на код](#)):

```
FROM python:3.6.0-alpine

WORKDIR /app
ADD . /app

RUN pip install -r /app/requirements.txt

ENV POST_DATABASE_HOST post_db
ENV POST_DATABASE posts

CMD ["python3", "post_app.py"]
```


Сервис comment

Создайте файл `./comment/Dockerfile` следующего содержания ([ссылка на код](#)):

```
FROM ruby:2.2
RUN apt-get update -qq && apt-get install -y build-essential

ENV APP_HOME /app
RUN mkdir $APP_HOME
WORKDIR $APP_HOME
ADD Gemfile* $APP_HOME/
RUN bundle install
COPY . $APP_HOME

ENV COMMENT_DATABASE_HOST comment_db
ENV COMMENT_DATABASE comments

CMD ["puma"]
```

Сервис ui

Создайте файл `./ui/Dockerfile` следующего содержания ([ссылка на код](#)):

```
FROM ruby:2.2
RUN apt-get update -qq && apt-get install -y build-essential

ENV APP_HOME /app
RUN mkdir $APP_HOME
WORKDIR $APP_HOME
ADD Gemfile* $APP_HOME/
RUN bundle install
ADD . $APP_HOME
ENV POST_SERVICE_HOST post
ENV POST_SERVICE_PORT 5000
ENV COMMENT_SERVICE_HOST comment
ENV COMMENT_SERVICE_PORT 9292
CMD ["puma"]
```

Сборка приложения

Скачаем последний образ MongoDB:

```
docker pull mongo:latest
```

Соберем образы с нашими сервисами:

```
docker build -t <your-dockerhub-login>/post:1.0 ./post-py
```

```
docker build -t <your-dockerhub-login>/comment:1.0 ./comment
```

```
docker build -t <your-dockerhub-login>/ui:1.0 ./ui
```

Обратите внимание! Сборка **ui** началась не с первого шага. Подумайте - почему?

[Ссылка на файл с командами](#)

Запуск приложения

Создадим специальную сеть для приложения:

```
docker network create reddit
```

Запустим наши контейнеры

```
docker run -d --network=reddit \  
  --network-alias=post_db --network-alias=comment_db mongo:latest
```

```
docker run -d --network=reddit \  
  --network-alias=post <your-dockerhub-login>/post:1.0
```

```
docker run -d --network=reddit \  
  --network-alias=comment <your-dockerhub-login>/comment:1.0
```

```
docker run -d --network=reddit \  
  -p 9292:9292 <your-dockerhub-login>/ui:1.0
```

[Ссылка на файл с командами](#)

Запуск приложения

Что мы сделали?

- Создали bridge-сеть для контейнеров, так как сетевые алиасы не работают в сети по умолчанию (о сетях в Docker мы поговорим на следующем занятии)
- Запустили наши контейнеры в этой сети
- Добавили сетевые алиасы контейнерам

Сетевые алиасы могут быть использованы для сетевых соединений, как доменные имена

Проверка

- Зайдите на <http://<docker-host-ip>:9292/>
- Напишите пост
- Работает!

Задание со ★

- Остановите контейнеры:

```
docker kill $(docker ps -q)
```

- Запустите контейнеры с другими сетевыми алиасами
- Адреса для взаимодействия контейнеров задаются через **ENV**-переменные внутри **Dockerfile**'ов
- При запуске контейнеров (**docker run**) задайте им переменные окружения соответствующие новым сетевым алиасам, не пересоздавая образ
- Проверьте работоспособность сервиса

Образы приложения

А что там с образами?

```
> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ui	1.0	17474322520e	About an hour ago	778MB
comment	1.0	0f5d7c15efe5	About an hour ago	774MB
post-py	1.0	a94c2e9bfc6b	About an hour ago	102MB
ruby	2.2	a7d32d62bed5	2 weeks ago	728MB
mongo	latest	39f5c173b5d4	3 weeks ago	361MB
python	3.6.0-alpine	cb178ebbf0f2	7 months ago	88.6MB

Похоже образы небольшого приложения занимают немало места!

Сервис ui - улучшаем образ

Поменяем содержимое `./ui/Dockerfile` ([ссылка на код](#)):

```
FROM ubuntu:16.04
RUN apt-get update \
    && apt-get install -y ruby-full ruby-dev build-essential \
    && gem install bundler --no-ri --no-rdoc

ENV APP_HOME /app
RUN mkdir $APP_HOME
WORKDIR $APP_HOME
ADD Gemfile* $APP_HOME/
RUN bundle install
ADD . $APP_HOME
ENV POST_SERVICE_HOST post
ENV POST_SERVICE_PORT 5000
ENV COMMENT_SERVICE_HOST comment
ENV COMMENT_SERVICE_PORT 9292
CMD ["puma"]
```

Образы приложений

Пересоберем **ui** (с какого шага началась сборка?):

```
> docker build -t <your-login>/ui:2.0 ./ui
```

```
> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ui	2.0	fb6b3a0dade1	9 seconds ago	453MB
ui	1.0	17474322520e	About an hour ago	778MB
comment	1.0	0f5d7c15efe5	About an hour ago	774MB
post-py	1.0	a94c2e9bfc6b	About an hour ago	102MB
ubuntu	16.04	2d696327ab2e	2 weeks ago	122MB
ruby	2.2	a7d32d62bed5	2 weeks ago	728MB
mongo	latest	39f5c173b5d4	3 weeks ago	361MB
python	3.6.0-alpine	cb178ebbf0f2	7 months ago	88.6MB

И посмотрим на результат - уже лучше! А предел ли это?

Задания со ★

- Попробуйте собрать образ на основе Alpine Linux
- Придумайте еще способы уменьшить размер образа
- Можете реализовать как только для UI сервиса, так и для остальных (**post**, **comment**)
- Все оптимизации проводите в **Dockerfile** сервиса.
Дополнительные варианты решения уменьшения размера образов можете оформить в виде файла **Dockerfile. <цифра>** в папке сервиса

Перезапуск приложения

Выключим старые копии контейнеров:

```
> docker kill $(docker ps -q)
```

Запустим новые копии контейнеров:

```
> docker run -d --network=reddit \
  --network-alias=post_db --network-alias=comment_db mongo:latest

> docker run -d --network=reddit \
  --network-alias=post <your-dockerhub-login>/post:1.0

> docker run -d --network=reddit \
  --network-alias=comment <your-dockerhub-login>/comment:1.0

> docker run -d --network=reddit \
  -p 9292:9292 <your-dockerhub-login>/ui:2.0
```

[Ссылка на файл с командами](#)

Проверка

- Зайдите на `http://<docker-host-ip>:9292`
- Рабо... 🤖 Эй! А где ранее созданный пост?

Наши данные пропали вместе с остановкой контейнера `mongo`

Создадим Docker volume:

```
> docker volume create reddit_db
```

И подключим его к контейнеру с MongoDB...

Перезапуск приложения с volume

Выключим старые копии контейнеров:

```
> docker kill $(docker ps -q)
```

Запустим новые копии контейнеров:

```
> docker run -d --network=reddit --network-alias=post_db \
  --network-alias=comment_db -v reddit_db:/data/db mongo:latest

> docker run -d --network=reddit \
  --network-alias=post <your-login>/post:1.0

> docker run -d --network=reddit \
  --network-alias=comment <your-login>/comment:1.0

> docker run -d --network=reddit \
  -p 9292:9292 <your-login>/ui:2.0
```

[Ссылка на файл с командами](#)

Проверка

- Зайдите на <http://<docker-host-ip>:9292/>
- Напишите пост
- Перезапустите контейнеры
- Проверьте, что пост остался на месте

Проверка ДЗ

- Результаты вашей работы находятся в ветке **docker-3** вашего репозитория **microservices**
- В **README** внесите описание того, что сделано
- Создайте *Pull Request* к ветке **master** (описание PR нужно заполнять!)
- Добавьте "Labels" **docker-3** к вашему *Pull Request*
- После того, как один из преподавателей сделает approve пулл реквеста, ветку с ДЗ можно смерджить и закрыть PR