

# Docker-образы. Микросервисы.



# План

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

# Преподготовка

Подключаемся к docker host'y

```
> docker-machine ls
```

NAME	ACTIVE	DRIVER	STATE	URL	SWARM	DOCKER
<b>docker-host</b>	-	google	Running	tcp:// <b>35.195.54.250</b> :2376		v17.09.0-ce

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

# Преподготовка

- Работа будет осуществляться в репозитории `microservices`
- Создайте директорию "monolith" внутри репозитория и переместите в нее предыдущие наработки
- Создайте новую ветку "homework-02"
- Скачайте архив <https://goo.gl/ScXXG5> и распакуйте его содержимое внутрь репозитория

# Новая структура приложения

- Теперь наше приложение состоит из трех компонент:
  - `post-ru` - сервис отвечающий за написание постов
  - `comment` - сервис отвечающий за написание комментариев
  - `ui` - веб-интерфейс для других сервисов
- Для работы ему требуется `mongodb`

# post-py

- Создайте файл ./post-py/Dockerfile следующего содержания

```
FROM python:3.6.0-alpine
```

```
RUN pip install flask pymongo
```

```
WORKDIR /app
```

```
ADD . /app
```

```
ENV POST_DATABASE_HOST post_db
```

```
ENV POST_DATABASE posts
```

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

# comment

- Создайте файл `./comment/Dockerfile` следующего содержания

```
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 COMMENT_DATABASE_HOST comment_db
ENV COMMENT_DATABASE comments

CMD ["puma"]
```



- Создайте файл `./ui/Dockerfile` следующего содержания

```
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 образа с нашими сервисами

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

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

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

- Обратите внимание, что сборка ui началась не с первого шага. Почему?

# 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-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:1.0

# Запуск приложения

- Что мы сделали?
  - Создали bridge-сеть для контейнеров, так как сетевые алиасы не работают в сети по умолчанию
  - 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://35.195.54.250:9292/>
- Напишите пост
- Работает!



# Запуск приложения

- Остановите контейнеры

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

- Задайте контейнерам другие сетевые алиасы
- Переопределите соответствующие переменные окружения при запуске новых контейнеров через `docker run`
- Проверьте работоспособность сервиса

# Образа 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

- О нет, наши образа занимают столько места!



- **Поменяем содержание ./ui/Dockerfile**

```
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"]
```

# Образа приложения

- Пересоберем ui (с какого шага началась сборка?)

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

- И посмотрим на результат

```
> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ui	2.0	fb6b3a0dade1	9 seconds ago	456MB
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

- Уже лучше





# А предел ли это?

- Попробуйте собрать образ на основе alpine linux

# А предел ли это?

- Придумайте еще способы как можно уменьшить размер образа
- Если их долго или сложно реализовать напишите о них vitkhab в slack'e

# 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-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
```

# Проверка

- Зайдите на <http://docker-host:9292>
- Рабо... эй! а где наш пост?
- Наши данные пропали вместе с остановкой контейнера mongo
- Создадим docker volume

> `docker volume create reddit_db`

- И подключим его к mongodb

# Перезапуск приложения

- Выключим старые копии контейнеров

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

- Запустим новые копии контейнеров

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

# Проверка

- Зайдите на <http://35.195.54.250:9292/>
- Напишите пост
- Перезапустите контейнеры
- Проверьте, что пост остался на месте

# Проверочное задание

- Создайте PR с вашими наработками
- Пригласите одного из следующих преподавателей на review вашего PR:
  - vitkhab