

Docker- контейнеры



План

- Создание docker host
- Создание своего образа
- Работа с Docker Hub

Требования

- Установленный docker

Docker machine

- Mac OS и Windows
идет в комплекте с docker (docker-machine -v)
- Linux
<https://docs.docker.com/machine/install-machine/>



- Создайте новый проект
<https://console.cloud.google.com/compute>
- Назовите его docker
- Запомните ID вашего проекта

gcloud

- Установите GCloud SDK
- <https://cloud.google.com/sdk/>

gcloud init

```
> gcloud init
```

```
You must log in to continue. Would you like to log in (Y/n)? Y
```

```
Your browser has been opened to visit:
```

```
https://accounts.google.com/o/oauth2/....
```

У вас должен запуститься браузер с выбором учетных записей Google. Выберите ту, к которой привязан GCE и разрешите доступ приложению

Если браузер не запустился, то запустите его самостоятельно и перейдите по ссылке выведенной в консоле

gcloud init

...

Pick cloud project to use:

[1] docker-181710

[2] Create a new project

Please enter numeric choice or text value (must exactly match list item): 1

...

Do you want to configure Google Compute Engine
(<https://cloud.google.com/compute>) settings (Y/n)? n

...

Мы сконфигурировали gcloud

gcloud auth

```
> gcloud auth application-default login  
Your browser has been opened to visit:
```

```
https://accounts.google.com/o/oauth2/....
```

У вас должен запуститься браузер с выбором учетных записей Google. Выберите ту, к которой привязан GCE и разрешите доступ приложению

Если браузер не запустился, то запустите его самостоятельно и перейдите по ссылке выведенной в консоле

gcloud auth

А теперь получили файл с аутентификационными данными.

Он будет использоваться docker-machine для работы с облаком.

Docker machine

```
> docker-machine create --driver google \  
--google-project docker-181710 \  
--google-zone europe-west1-b \  
--google-machine-type f1-micro \  
--google-machine-image $(gcloud compute images list --filter ubuntu-1604-lts --uri) \  
docker-host
```

Docker machine

Проверяем, что наш Docker-хост успешно создан...

```
> docker-machine ls
```

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

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

... и начинаем с ним работу

Namespaces



- Сравните вывод:
 - `docker run --rm -ti tehbilly/htop`
 - `docker run --rm --pid host -ti tehbilly/htop`

Новый репозиторий

- Создайте репозиторий "microservices" на GitHub'e
- Поставьте галочку "Initialize this repository with README"
- Добавьте следующих преподавателей в collaborators вашего github репозитория:
 - vitkhab

Новый репозиторий

- Загрузите репозиторий к себе на машину
- Создайте ветку "homework-01"

```
> git clone git@github.com:<your-login>/microservices.git  
> cd microservices  
> git checkout -b homework-01
```
- Дальше вся работа должна происходить в рамках НОВОЙ ветки
- Инструкций когда следует делать коммиты не будет. Частота и качество коммитов остается на ваше усмотрение

Структура репозитория

- Для дальнейшей работы нам потребуются четыре файла, их содержание вы найдете на следующих слайдах
 - Dockerfile - текстовое описание нашего образа
 - mongod.conf - подготовленный конфиг для mongod
 - db_config - содержит переменную со ссылкой на mongod
 - start.sh - скрипт запуска приложения

mongodb.conf

Where and how to store data.

storage:

dbPath: /var/lib/mongodb

journal:

enabled: true

where to write logging data.

systemLog:

destination: file

logAppend: true

path: /var/log/mongodb/mongod.log

network interfaces

net:

port: 27017

bindIp: 127.0.0.1

start.sh

```
#!/bin/bash
```

```
/usr/bin/mongod --fork --logpath /var/log/mongod.log --config /etc/mongodb.conf
```

```
source /reddit/db_config
```

```
cd /reddit && puma || exit
```

db_config

DATABASE_URL=127.0.0.1

Dockerfile

- Начнем создавать образ с приложением. За основу возьмем известный нам дистрибутив ubuntu версии 16.04
- Создадим файл "Dockerfile" и добавим в него строки:

```
> vim Dockerfile  
FROM ubuntu:16.04
```
- Вместо vim используйте любимый ваш редактор

Dockerfile

- Для работы приложения нам нужны mongo и ruby.
- Обновим кеш репозитория и установим нужные пакеты
- Добавьте в "Dockerfile" строки:

```
FROM ubuntu:16.04
```

```
RUN apt-get update
```

```
RUN apt-get install -y mongodb-server ruby-full ruby-dev build-essential git
```

```
RUN gem install bundler
```

Dockerfile

- Скачаем наше приложение в контейнер

```
FROM ubuntu:16.04
```

```
RUN apt-get update
```

```
RUN apt-get install -y mongodb-server ruby-full ruby-dev build-essential git
```

```
RUN gem install bundler
```

```
RUN git clone https://github.com/Artemmkin/reddit.git
```

Dockerfile

- Скопируем файлы конфигурации в контейнер:

```
FROM ubuntu:16.04
```

```
RUN apt-get update
```

```
RUN apt-get install -y mongodb-server ruby-full ruby-dev build-essential git
```

```
RUN gem install bundler
```

```
RUN git clone https://github.com/Artemmkin/reddit.git
```

```
COPY mongod.conf /etc/mongod.conf
```

```
COPY db_config /app/db_config
```

```
COPY start.sh /start.sh
```

Dockerfile

- Нам нужно установить зависимости приложения и произвести настройку:

```
FROM ubuntu:16.04
```

```
RUN apt-get update
```

```
RUN apt-get install -y mongodb-server ruby-full ruby-dev build-essential git
```

```
RUN gem install bundler
```

```
RUN git clone https://github.com/Artemmkin/reddit.git
```

```
COPY mongod.conf /etc/mongod.conf
```

```
COPY db_config /reddit/db_config
```

```
COPY start.sh /start.sh
```

```
RUN cd /reddit && bundle install
```

```
RUN mkdir -p /data/db
```

```
RUN chmod 0777 /start.sh
```


Dockerfile

- Выполняем старт сервиса при старте контейнера

```
FROM ubuntu:16.04
```

```
RUN apt-get update
```

```
RUN apt-get install -y mongodb-server ruby-full ruby-dev build-essential git
```

```
RUN gem install bundler
```

```
RUN git clone https://github.com/Artemmkin/reddit.git
```

```
COPY mongod.conf /etc/mongod.conf
```

```
COPY db_config /reddit/db_config
```

```
COPY start.sh /start.sh
```

```
RUN cd /reddit && bundle install
```

```
RUN mkdir -p /data/db
```

```
RUN chmod 0777 /start.sh
```

```
CMD ["/start.sh"]
```

Сборка образа

- Теперь мы готовы собрать свой образ
 - Выполните команду.
 - Точка в конце обязательна, она указывает на путь до Docker-контекста
 - Флаг -t задает тег для собранного образа
- > docker build -t reddit:latest .

...

Successfully built ad21718d3eb5
Successfully tagged reddit:latest

Сборка образа

- Проверим результат

```
> docker images -a
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	<none>	e0a3c979a0fb	2 minutes ago	687MB
<none>	<none>	0e07ff248072	2 minutes ago	687MB
reddit	latest	f7ff59db9594	2 minutes ago	687MB
<none>	<none>	96f7a69d48d2	2 minutes ago	687MB
<none>	<none>	a809efbf09a0	2 minutes ago	657MB
<none>	<none>	476d8d488857	2 minutes ago	657MB
<none>	<none>	b163309613e1	2 minutes ago	657MB
<none>	<none>	f7b02c42278f	2 minutes ago	657MB
<none>	<none>	746e7a1087d9	2 minutes ago	657MB
<none>	<none>	2d2469adc562	3 minutes ago	654MB
<none>	<none>	7ea520e2fdf1	6 minutes ago	161MB
ubuntu	16.04	2d696327ab2e	2 weeks ago	122MB

Запуск контейнера

- Отлично, теперь можно запустить наш контейнер командой

```
> docker run --name reddit -d --network=host reddit:latest
```

- Проверим результат

```
> docker-machine ls
```

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

- Откройте в браузере ссылку <http://35.195.54.250:9292>

Ошибка



This site can't be reached

35.195.54.250 took too long to respond.

Search Google for [195 250 9292](#)

ERR_CONNECTION_TIMED_OUT

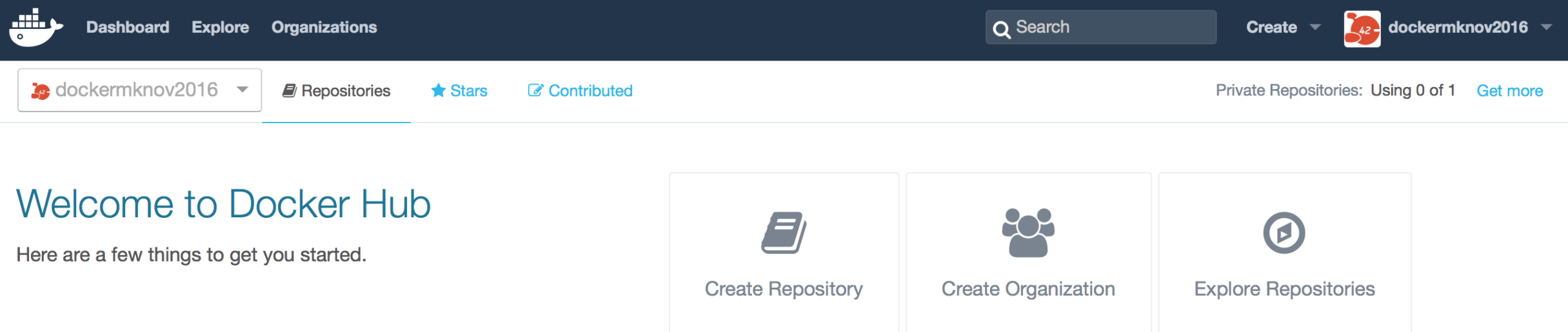
Настройка firewall'a

- Перейдите на страницу настройки firewall'a в GCE (убедитесь, что у вас выбран проект docker)
<https://console.cloud.google.com/networking/firewalls/list>
- Создайте новое правило
 - Название => reddit-app
 - Теги целевых экземпляров => docker-machine
 - Диапазоны IP-адресов источников => 0.0.0.0/0
 - Протоколы и порты => Указанные протоколы и порты => tcp:9292
- Попробуйте снова открыть ссылку (hint: должен открыться наш сервис)
<http://35.195.54.250:9292>

Docker hub: регистрация

- Перейдите по ссылке <https://hub.docker.com/>
- Зарегистрируйте учетную запись, если у вас ее еще нет

Docker hub: регистрация



Docker hub: аутентификация

- Аутентифицируемся на docker hub для продолжения работы

> docker login

Login with your Docker ID to push and pull images from Docker Hub.

If you don't have a Docker ID, head over to <https://hub.docker.com> to create one.

Username: your-login

Password:

Login Succeeded

Docker hub: push

- Загрузим наш образ на docker hub для использования в будущем

```
>docker tag reddit:latest <your-login>/otus-reddit:1.0
```

```
>docker push <your-login>/otus-reddit:1.0
```

The push refers to a repository [docker.io/<your-login>/otus-reddit]

c6e5100de1e0: Pushed

...

a2022691bf95: Pushed

1.0: digest: sha256:77c6070400a5b04f8db3f7c129a2c16084c2fcf186aa6b436c8d6f57e0014378
size: 3448

Проверочное задание

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