

Деплой тестового приложения

Внимание!

Хотим вас предупредить, что копирование из pdf команд может вызвать некоторые трудности. Например, при копировании команд, не уместающихся в одну строку, нельзя скопировать команду целиком и придется копировать по частям :(Просим быть внимательными и сверять, команды которые вы используете с теми, что представлены на слайдах.

Установите Google Cloud SDK

Убедитесь, что у вас установлен и инициализирован gcloud . Инструкции по установке и инициализации для вашей ОС можно найти [здесь](#).

Проверить установку и настройку gcloud можно, используя команду `gcloud info`

Создаем SSH ключ для пользователя appuser

```
$ ssh-keygen -t rsa -f ~/.ssh/appuser -C appuser -P ""
```

Имя файла

Имя пользователя для SSH

Добавляем публичный ключ пользователя в GCP

The screenshot shows the Google Cloud Platform console interface. On the left, the 'Compute Engine' menu is open, with 'Metadata' selected. The main panel displays the 'Metadata' tab for a VM instance, with the 'SSH Keys' sub-tab active. A text area contains a public SSH key for a user named 'appuser'. Below the text area is an 'Add item' button. At the bottom, there are 'Save' and 'Cancel' buttons.

Compute Engine

- VM instances
- Instance groups
- Instance templates
- Disks
- Snapshots
- Images
- Committed use discounts
- Metadata**

Metadata

Metadata SSH Keys

appuser

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCbIG
LGxxaPLscKS95LNx6FAEwP6PZ7P9wLbfoGwUcub3xH
J+NIOpPq0iN25G3Jv0ak/uSn1zYpAdCGZ+aa6UNjTy
XwiteRMFg3SJkmP9v0q1sQCqkNDsDVucw6EDXjG0eR
Nxm00/ZyNt0LU2JhARPT8zA5hXrYpDqc6ZznJ8xIU4
|
```

+ Add item

Save Cancel

Создаем новый инстанс

Используем gcloud для создания инстанса, для проверки корректности CLI после настройки

```
$ gcloud compute instances create --boot-disk-size=10GB --  
image=ubuntu-1604-xenial-v20170815a --image-project=ubuntu-os-cloud --  
machine-type=g1-small --tags puma-server --restart-on-failure --  
zone=europe-west1-b reddit-app
```

[Ссылка на gist с командой](#)

Устанавливаем Ruby

Подключаемся к машине по SSH (не забудьте предварительно добавить приватный ключ в ssh агент):

```
$ ssh appuser@<instance_public_ip>
```

Устанавливаем RVM для пользователя appuser:

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys  
409B6B1796C275462A1703113804BB82D39DC0E3  
$ curl -sSL https://get.rvm.io | bash -s stable
```

Устанавливаем Ruby

Подгружаем RVM скрипт и устанавливаем зависимости для работы RVM и Ruby:

```
$ source ~/.rvm/scripts/rvm  
$ rvm requirements
```

Устанавливаем Ruby (версии 2.4.1) для пользователя appuser

```
$ rvm install 2.4.1
```


Устанавливаем Ruby

Укажите версию руби для пользователя по умолчанию

```
$ rvm use 2.4.1 --default
```

Установите bundler для управления зависимостями нашего приложения

```
$ gem install bundler -V --no-ri --no-rdoc
```

Проверьте установку Ruby и Bundler

```
$ ruby -v
```

```
$ bundle -v
```

Устанавливаем MongoDB

Добавляем ключи и ссылку на репозиторий:

```
$ sudo apt-key adv --keyserver hkp://  
keyserver.ubuntu.com:80 --recv EA312927
```

```
$ sudo bash -c 'echo "deb http://repo.mongodb.org/apt/  
ubuntu xenial/mongodb-org/3.2 multiverse" > /etc/apt/  
sources.list.d/mongodb-org-3.2.list'
```

Обновим индекс доступных пакетов и установим нужный пакет:

```
$ sudo apt-get update  
$ sudo apt-get install -y mongodb-org
```

Устанавливаем MongoDB

Запускаем MongoDB:

```
$ sudo systemctl start mongod
```

Добавляем в автозапуск:

```
$ sudo systemctl enable mongod
```

Проверяем статус:

```
$ sudo systemctl status mongod
```

Проверяем работу MongoDB

```
appuser@reddit-app:~$ sudo systemctl status mongod
• mongod.service - High-performance, schema-free document-oriented database
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2017-08-31 08:52:50 UTC; 53 ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 8805 (mongod)
    CGroup: /system.slice/mongod.service
            └─8805 /usr/bin/mongod --quiet --config /etc/mongod.conf
```

Деплой приложения

Копируем код приложения. Можно использовать HTTPS:

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

Переходим в директорию проекта и устанавливаем зависимости приложения

```
$ cd reddit && bundle install
```

Деплой приложения


Запускаем сервер приложения в папке проект:

```
$ puma -d
```


Проверьте что сервер запустился и на каком порту он слушает:


```
$ ps aux | grep puma
```


Открываем порт в файерволе


 Google Cloud Platform


My Project 4




 VPC network

 VPC networks

 External IP addresses

 Firewall rules


Firewall rules





Firewall rules control incoming or outgoing traffic to an instance. E
incoming traffic from outside your network is blocked. [Learn more](#)


Note: App Engine firewalls are managed [here](#).


Ingress Egress


 VPC network


 VPC networks


 External IP addresses

 Firewall rules


 Routes

 VPC network peering


 Shared VPC


 Create a firewall rule

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)


Name 

default-puma-server

Description (Optional) 

Network 

default

Priority 

Укажите имя
правила

Откройте доступ для всего мира, на серверы, в которых прописаны теги puma-server

Action on match ?

☒ Allow

☐ Deny

Targets ?

Specified target tags

Target tags

puma-server

Source filter ?

IP ranges

Source IP ranges ?

0.0.0.0/0

Second source filter ?

None

Protocols and ports ?

☐ Allow all

☒ Specified protocols and ports

tcp:80;

Create

Cancel

Откройте нужный tcp порт (помните мы проверяли порт сервера?)

на картинке намеренно допущена ошибка




Проверка работы приложения

Скопируйте внешний адрес вашего инстанса

VM instances [+ CREATE INSTANCE](#) [↓ IMPORT VM](#) [↻ REFRESH](#) [▶ START](#) [■ STOP](#)

☰ Filter VM instances

<input checked="" type="checkbox"/> Name ^	Zone	Machine type	Recommendation	Internal IP	External IP	Connect
<input checked="" type="checkbox"/>  reddit-app	europa-west1-b	1 vCPU, 1.7 GB		10.132.0.2	104.155.31.114	SSH ▾ ⋮

В адресной строке браузера укажите публичный адрес инстанса и порт сервера в формате `<external_ip>:<port>`

Monolith Reddit

Post successully published

^

0

▼

Trytravis - new cool tool !

30-08-2017
14:35

Go to the link

Menu

All posts

New post

Самостоятельная работа

Команды по настройке системы и деплоя приложения нужно завернуть в баш скрипты, чтобы не вбивать эти команды вручную:

- скрипт `install_ruby.sh` - должен содержать команды по установке руби.
- скрипт `install_mongodb.sh` - должен содержать команды по установке MongoDB
- скрипт `deploy.sh` должен содержать команды скачиванию кода, установлению зависимостей через `bundle` и запуску приложения.

Проверка ДЗ

- Создайте репозиторий с названием infra на GitHub. Добавьте в этот репозиторий README.md файл (с указанием, что это инфраструктурный репозиторий).
- Добавьте в collaborators пользователей Artemmkin и serjs.
- Создайте отдельную ветку для выполненного ДЗ, которую назовите config-scripts. Добавьте созданные скрипты в эту ветку.
- Создайте Pull Request для ветки мастер и добавьте в ревьюверы **Artemmkin** и **serjs**



Дополнительное задание

В качестве доп задания используйте созданные ранее скрипты для создания Startup script, который будет запускаться при создании инстанса. Передавать Startup скрипт необходимо как доп опцию уже использованной ранее команде gcloud. В результате применения данной команды gcloud мы должны получать инстанс с уже запущенным приложением. Startup скрипт необходимо закоммитить, а используемую команду gcloud вставить в описание репозитория (README.md)