

Развертывание ruby-приложений с использованием vagrant и capistrano.

1. Установка необходимых приложений.

- Устанавливаем vagrant - <https://www.vagrantup.com>
- Устанавливаем virtual box - <https://www.virtualbox.org>
- Создаём директорию для виртуальной машины
- Из директории выполняем команду
vagrant init ubuntu/xenial64; vagrant up --provider virtualbox
- В Vagrantfile раскомментируем строку
config.vm.network "private_network", ip: "192.168.33.10"
- Если вываливается ошибка про "vagrant a virtualbox machine with the name already exists" (или типа того), то идем в Vagrantfile и добавляем имя для машины:

```
config.vm.provider "virtualbox" do |vbl|  
  vb.name = "domochat-vagrant"  
end
```

С помощью этой команды мы создаём конфигурационный файл Vagrantfile, с помощью которого будет инициализироваться наша виртуальная машина и указываем, что в качестве провайдера в ней будет использоваться VirtualBox

2. Настраиваем виртуальную машину

- заходим на виртуальный сервер: ***vagrant ssh***
- залогиниться под superuser'ом: ***sudo -s***
- ставим ruby - используя rbenv, rvm или ***apt-get install ruby ruby-dev***
- ***apt-get install postgresql postgresql-server-dev-all redis-server nginx git nodejs imagemagick***
- установить компилятор C++: ***apt-get install gcc g++***
- создаём пользователя deploy, с помощью которого мы будем разворачивать приложение на сервере: ***adduser deploy***
- залогиниться под юзером deploy: ***su -l deploy***
- добавляем наш публичный ssh-ключ из ***.ssh/id_rsa.pub*** в ***.ssh/authorized_keys*** пользователя deploy на сервере
- добавляем github в известные хосты на сервере ***ssh -T git@github.com***
- устанавливаем на сервере bundler - ***gem install bundler***
- добавить пользователя deploy в группу www-data: ***vigr*** (потом найти строчку www-data и в ее конец добавить имя пользователя deploy)
- залогиниться под дефолтным юзером postgres, чтобы создать пользователя базы данных: ***su -l postgres***
- Добавляем пользователя (например ***db_user***) в БД на сервере
createuser --interactive db_user -P
- Создаём БД на сервере и даём права на неё созданному пользователю db_user
createdb domochat -O db_user
- Выйти из юзера postgres (ctrl + D) и создать папку domochat (или другое имя проекта):
mkdir /var/www/domochat

- Изменить владельца созданной директории (из /var/www): **chown -R www-data:www-data /var/www/domochat**
- Устанавливаем нужные права для пользователя deploy (доступ к созданной папке): **chmod 775 -R /var/www/domochat**

3. Настраиваем приложение

- Добавляем gem-ы в Gemfile

```
group :development do
  gem 'capistrano',      require: false
  gem 'capistrano-sidekiq', require: false
  gem 'capistrano-rails', require: false
  gem 'capistrano-bundler', require: false
  gem 'capistrano3-puma', require: false
end
```

- Выполняем команду **bundle install**
- Выполняем команду **cap install**
- Добавляем в Capfile строки

```
require 'capistrano/rails/assets'
require 'capistrano/rails/migrations'
require 'capistrano/rails'
require 'capistrano/bundler'
require 'capistrano/sidekiq'
require 'capistrano/puma'
```

- Указываем в production.rb или staging.rb (папка config/deploy) реквизиты нашего сервера

```
server '192.168.33.10', user: 'deploy', roles: %w{app db web}
```

- Конфигурируем наш deploy.rb - <https://gist.github.com/alec-c4/8761bd9cf0d8f0983b2ebeb7516d3d>
- Выполняем команды из директории проекта (проверка наличия папок, а потом копирование конфигурационных файлов):

```
cap staging deploy:check
cap staging deploy:setup
```

или

```
cap production deploy:check
cap production deploy:setup
```

- Редактируем конфигурационные файлы на сервере в директории **/var/www/domochat/shared/config/** и указываем корректные значения (меняем название environment на актуальное, например production)
- Конфигурируем nginx, чтобы он обрабатывал запросы и отправлял их в сервер наш приложений. Для этих целей логинимся под пользователем root (ctrl + D или sudo -s), открываем файл **/etc/nginx/sites-enabled/default** и прописываем следующие параметры

```
upstream puma {
  server unix:///var/www/domochat/shared/tmp/sockets/puma.sock;
}
```

```

server {
    listen 80 default_server deferred;
    listen [::]:80 default_server;

    root /var/www/domochat/public;

    index index.html index.htm;

    server_name _;

    location ^~/assets/ {
        gzip_static on;
        expires max;
        add_header Cache-Control public;
    }

    try_files $uri/index.html $uri @puma;

    location @puma {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_redirect off;

        proxy_pass http://puma;
    }

    error_page 500 502 503 504 /500.html;
    client_max_body_size 10M;
    keepalive_timeout 10;
}

```

- делаем команду на сервере ***sudo service nginx restart*** и проверяем работоспособность сервера
- с локального компьютера делаем деплой: ***cap production deploy (или cap staging deploy)***

При необходимости правим конфигурационные файлы nginx и deploy.rb