

Развёртывание проекта

1 Системные требования

Требование к аппаратуре сервера:

- минимальный размер оперативной памяти – 2 ГБ, желательно – 4 ГБ;
- минимально число процессорных ядер – 2, желательно – 4 ядра;
- минимальный размер диска – 12 ГБ, желательно – 32 ГБ;
- доступ в интернет на время установки и настройки системы.

Операционная система GNU/Linux дистрибутивы CentOS 7 или Debian 9.

2 Обновление операционных систем и установка дополнительных пакетов

2.1 Операционная система Debian

```
apt-get clean && apt-get update && apt-get upgrade -y
```

```
apt-get install -y --no-install-recommends vim procs git wget gnupg2 curl  
dirmngr ca-certificates unzip sudo apt-transport-https apache2 apache2-dev  
build-essential libaprutil1-dev libcurl4-openssl-dev libssl1.0-dev
```

```
reboot
```

2.2 Операционная система CentOS

```
yum clean all && yum makecache && yum update -y
```

```
yum groupinstall 'Development Tools'  
yum install git wget gnupg2 curl unzip sudo httpd httpd-devel
```

```
reboot
```

3 Установка СУБД PostgreSQL

Для работы системы необходима версия СУБД PostgreSQL не ниже 9.4, применение версии 10 не рекомендуется.

3.1 Установка в операционной системе Debian

```
echo "deb [arch=amd64] http://apt.postgresql.org/pub/repos/apt/ stretch-pgdg  
main" | tee -a /etc/apt/sources.list.d/pgdg.list
```

```
wget -t3 -cq4 -O - http://apt.postgresql.org/pub/repos/apt/ACCC4CF8.asc | apt-  
key add -
```

```
apt-get update
```

```
apt-get install -y postgresql-9.6 postgresql-server-dev-9.6
```

```
/etc/init.d/postgresql start
```

```
systemctl enable postgresql-9.6.service
```

3.2 Установка в операционной системе CentOS

```
rpm -ivh https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-  
x86_64/pgdg-redhat96-9.6-3.noarch.rpm
```

```
yum update -y
```

```
yum install postgresql96-server postgresql96-contrib postgresql96-devel
```

```
service postgresql initdb
```

```
systemctl enable postgresql-9.6
```

```
systemctl start postgresql-9.6
```

4 Настройка СУБД PostgreSQL

Сменить пользователя и запустить консоль базы данных:

```
su -l postgres
```

```
psql
```

Завести нового пользователя, базу, задать права доступа

```
CREATE USER <имя пользователя> WITH PASSWORD '<пароль>';
```

```
CREATE DATABASE <имя базы> OWNER <имя пользователя>;
```

```
GRANT ALL PRIVILEGES ON DATABASE <имя базы> TO <имя  
пользователя>;
```

Если есть желание и понимание происходящего, изменить дефолтные настройки базы данных для имеющихся аппаратных возможностей (ОЗУ – 4 ГБ, твердотельный накопитель – 16 ГБ):

```
ALTER SYSTEM SET max_connections = '24';  
ALTER SYSTEM SET shared_buffers = '512MB';  
ALTER SYSTEM SET effective_cache_size = '1536MB';  
ALTER SYSTEM SET work_mem = '10922kB';  
ALTER SYSTEM SET maintenance_work_mem = '128MB';  
ALTER SYSTEM SET min_wal_size = '1GB';  
ALTER SYSTEM SET max_wal_size = '2GB';  
ALTER SYSTEM SET checkpoint_completion_target = '0.7';  
ALTER SYSTEM SET wal_buffers = '16MB';  
ALTER SYSTEM SET default_statistics_target = '100';  
ALTER SYSTEM SET random_page_cost = '1.1';
```

Выйти из консоли СУБД

```
\q
```

Выйти из сеанса пользователя postgres

logout

5 Установка RoR и настройка веб-сервера

```
gpg2 --keyserver hkp://keys.gnupg.net --recv-keys  
409B6B1796C275462A1703113804BB82D39DC0E3  
7D2BAF1CF37B13E2069D6956105BD0E739499BDB
```

```
export rvm_make_flags="-j 3"
```

```
export rvm_gem_options="--no-rdoc --no-ri"
```

```
\curl -sSL https://get.rvm.io | bash -s stable --rails
```

```
source /usr/local/rvm/scripts/rvm
```

```
rvm install ruby-2.3.6
```

```
rvm --default use ruby-2.3.6
```

```
gem install pg -v '0.18.4'
```

```
gem install passenger --no-rdoc --no-ri
```

```
passenger-install-apache2-module --languages ruby -a
```

Дальше для операционной системы Debian в директории /etc/apache2/conf-available/ создать файл passenger.conf следующего содержания:

```
LoadModule passenger_module /usr/local/rvm/gems/ruby-  
2.3.6/gems/passenger-5.1.12/buildout/apache2/mod_passenger.so  
<IfModule mod_passenger.c>  
  PassengerRoot /usr/local/rvm/gems/ruby-2.3.6/gems/passenger-5.1.12  
  PassengerDefaultRuby /usr/local/rvm/gems/ruby-2.3.6/wrappers/ruby  
</IfModule>
```

Создать символическую ссылку на этот файл:

```
ln -s /etc/apache2/conf-available/passenger.conf /etc/apache2/conf-enabled/passenger.conf
```

В директории /etc/apache2/sites-available создать файл ac.conf примерно следующего содержания (значение параметров DocumentRoot, CustomLog, ErrorLog, <Directory могут быть другими):

```
<VirtualHost *:80>
    #ServerName <имя проекта>.d.rnds.pro
    #ServerAlias www.<имя проекта>.d.rnds.pro <имя
проекта>.d.rnds.pro
    AddDefaultCharset UTF-8
    PassengerRuby /usr/local/rvm/gems/ruby-2.3.6/wrappers/ruby
    #RailsEnv production
    RailsEnv development
    DocumentRoot /var/www/esia/public
    ServerAdmin webmaster@<имя проекта>.s.rnd-soft.ru
    CustomLog /var/www/esia/log/access.log combined
    ErrorLog /var/www/esia/log/error.log
</VirtualHost>
<Directory /var/www/esia/public>
    Options +Includes -ExecCGI
    Require all granted
</Directory>
```

ВАЖНО! Directory и DocumentRoot должны указывать на папку public внутри проекта.

Если хотите чтобы сервер открывал Rails-приложение даже при обращении по ip, то необходимо сделать так, чтобы конфиг этого домена был первым в списке файлов папки sites-enabled. Например, можно удалить 000-default или просто прописать ваш конфиг в него.

Создать символическую ссылку на этот файл:

`ln -s /etc/apache2/sites-available/ac.conf /etc/apache2/sites-enabled/ac.conf`
Удалить идущий по умолчанию файл `000-default.conf`

`rm -f /etc/apache2/sites-enabled/000-default.conf`

В операционной системе CentOS необходимые правки выполняются в файлах `/etc/httpd/conf/httpd.conf` и `/etc/httpd/conf.d/userdir.conf`

После выполнения настроек перезапустить веб-сервер:

`/etc/init.d/apache2 restart` (debian)

`systemctl restart httpd` (centos)

6 Доставка кода проекта на сервер

Перейти в директорию проекта

`cd /var/www/<domain>`

Клонировать содержимое проекта

`git clone git@rnds.pro:<имя проекта>`

Проверить наличие файлов конфигурации
/var/www/<domain>config/database.yml и config/secrets.yml. **В случае
отсутствия создать из прилагаемых файлов *.example** или из
приведенных ниже шаблонов.

Примерное содержимое файла database.yml:

```
default: &default
  adapter: postgresql
  host: localhost
  database: <имя базы>
  username: <имя пользователя базы>
  password: '<пароль пользователя>'
  pool: 24
  timeout: 5000
```

development:

```
<<: *default
```

test:

```
<<: *default
  database: <имя базы>_test
```

production:

```
<<: *default
```

Прописать реальные значения имени пользователя, пароль и название БД.

Примерное содержимое файла secrets.yml

development:

```
secret_key_base: <key>
```

test:

```
secret_key_base: <key>
```

production:

```
secret_key_base: <key>
```

где <key> секретный ключ, который можно сгенерировать выполнив `rake secret` в папке проекта на локальной машине.

Добавить сертификат и ключ в директорию
`/var/www/<domain>/config/keys/`

Добавить пользователя веб-сервера в группу `rvm` и задать ему права доступа на проект

```
gpasswd -a www-data rvm
```

```
chown www-data:www-data /var/www/<domain>
```

Установить необходимые зависимости проекта, выполнив в папке проекта
`gem install bundle && bundle install`

Мигрировать и заполнить стартовым набором данных базу, выполнив в папке проекта команду

```
rake db:migrate db:seed
```

Ошибки на этом этапе могут быть:

- Проблемы запуска окружения самого проекта. Перепроверить можно выполнив `rails c` в папке проекта. Если консоль запустилась, значит с окружением все хорошо. Можно пробовать дописывать `bundle exec` перед всеми командами `rails\rake`.
- Проблемы подключения к БД. Может быть не запущен `postgres`, он может быть не настроен на прослушивание портов, неверно указали логин\пароль. И очень часто встречаются проблемы с заполнением прав доступа в `pg_hba.conf`.

Если запускается продакшен-версия то надо скомпилировать ассеты
`$ RAILS_ENV=production rake assets:precompile`

7 Очистка после установки, если требуется

7.1 В операционной системе Debian выполнить команды

```
apt-get purge -y --auto-remove wget gnupg2 curl unzip apt-transport-https git  
gawk zlib1g-dev libsqlite3-dev sqlite3 libgmp-dev libncurses5-dev libffi-dev  
libreadline-dev libssl-dev build-essential apache2-dev postgresql-server-dev-9.6
```

```
apt-get autoclean -y
```

```
apt-get clean
```

```
rm -rf /var/lib/apt/lists/*
```

```
rm -rf /var/cache/apt/archives/*
```

```
rm -rf /var/tmp/*
```

7.2 В операционной системе CentOS выполнить команды

```
yum groupremove 'Development Tools' -y
```

```
yum erase -y wget gnupg2 curl unzip sqlite3 httpd-devel postgresql96-devel
```

Примечание. Некоторые пункты для операционной системы CentOS не протестированы.