

Deploy production server on CentOS 7.4

Overview

1. Setup deploy user
2. Install [Ruby](#)
3. Install [MySQL](#)
4. Install [Redis](#)
5. Install [RabbitMQ](#)
6. Install [Bitcoind](#)
7. Install [Nginx with Passenger](#)
8. Install JavaScript Runtime
9. Install ImageMagick
10. Configure DCL

1. Setup deploy user

Create (if it doesn't exist) deploy user, and assign it to the sudo group:

```
sudo adduser deploy
sudo usermod -a -G sudo deploy
```

Hints: To support sudo, you should edit /etc/sudoers to enable corresponding group.

Re-login as deploy user

2. Install Ruby

Make sure your system is up-to-date.

```
sudo yum install ruby
```

Installing [rbenv](#) using a Installer

```
sudo yum install curl git-core zlib zlib-devel gcc gcc-c++ make openssl-devel \
    patch readline readline-devel libxml2-devel libxml2 libyaml-devel
libxslt-devel \
    libffi-devel bzip2 autoconf \
    automake libtool bison sqlite-devel

cd

git clone git://github.com/sstephenson/rbenv.git .rbenv
echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
echo 'eval "$(rbenv init -)"' >> ~/.bashrc
exec $SHELL

git clone git://github.com/sstephenson/ruby-build.git ~/.rbenv/plugins/ruby-build
echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.bashrc
exec $SHELL
```

Install Ruby through rbenv:

```
rbenv install 2.2.1
rbenv global 2.2.1
```

Hints: maybe install failed, you should `sudo yum install openssl readline zlib openssl-devel readline-devel`

Install bundler

```
echo "gem: --no-ri --no-rdoc" > ~/.gemrc
gem install bundler
rbenv rehash
```

3. Install MySQL

```
sudo yum install mariadb-server mariadb mariadb-devel
```

4. Install Redis

Be sure to install the latest stable Redis, as the package in the distro may be a bit old:

```
sudo yum install epel-release
sudo yum install jemalloc
sudo yum install redis-server
```

5. Install RabbitMQ

Please follow instructions here: <https://www.rabbitmq.com/install-debian.html>

```
# Download the latest RabbitMQ package using wget:
wget http://www.rabbitmq.com/releases/rabbitmq-server/v3.2.2/rabbitmq-server-3.2.2-1.noarch.rpm

# Add the necessary keys for verification:
rpm --import http://www.rabbitmq.com/rabbitmq-signing-key-public.asc

# Install the .RPM package using YUM:
yum install rabbitmq-server-3.2.2-1.noarch.rpm

#if erlang >= R13B-03 for package (download aliyun repo)

sudo rabbitmq-plugins enable rabbitmq_management
sudo service rabbitmq-server restart
wget http://localhost:15672/cli/rabbitmqadmin
chmod +x rabbitmqadmin
sudo mv rabbitmqadmin /usr/local/sbin
```

6. Install Bitcoind

```
dcl support cold wallet, not needed.
```

7. Installing Nginx & Passenger

Please refer to <https://www.phusionpassenger.com/library/install/nginx/install/oss/el6/>

```
sudo yum install epel-release
sudo yum install -y pygpgme curl
sudo curl --fail -sSLo /etc/yum.repos.d/passenger.repo https://oss-binaries.phusionpassenger.com/yum/definitions/el-passenger.repo
sudo yum install -y nginx passenger || sudo yum-config-manager --enable cr && sudo yum install -y nginx passenger
```

Next, we need to update the Nginx configuration to point Passenger to the version of Ruby that we're using. You'll want to open up `/etc/nginx/nginx.conf` in your favorite editor,

```
sudo vim /etc/nginx/nginx.conf
```

find the following lines, and uncomment them:

```
sudo vim /etc/nginx/conf.d/passenger.conf
passenger_root /usr/lib/ruby/vendor_ruby/phusion_passenger/locations.ini;
passenger_ruby /usr/bin/ruby;
```

update the second line to read:

```
passenger_ruby /home/deploy/.rbenv/shims/ruby;  
               /home/bitex
```

8. Install JavaScript Runtime

A JavaScript Runtime is needed for Asset Pipeline to work. Any runtime will do but Node.js is recommended.

```
sudo yum install nodejs
```

9. Install ImageMagick

```
sudo yum -y install imagemagick  
gsfonts?
```

10. Setup production environment variable

```
echo "export RAILS_ENV=production" >> ~/.bashrc  
source ~/.bashrc
```

Clone the Source

```
mkdir -p ~/dcl  
git clone https://github.com/Changdao/dcl.git ~/dcl/current  
cd dcl/current  
  
# Install dependency gems  
bundle install --without development test --path vendor/bundle
```

Configure DCL

Prepare configure files

```
bin/init_config
```

Setup Pusher

- DCL depends on [Pusher](#). A development key/secret pair for development/test is provided in `config/application.yml` (uncomment to use). PLEASE USE IT IN DEVELOPMENT/TEST ENVIRONMENT ONLY!

More details to visit [pusher official website](#)

```
# uncomment Pusher related settings (Delete the extra three lines below )
vim config/application.yml
```

Setup bitcoind rpc endpoint

```
# replace username:password and port with the one you set in
# username and password should only contain letters and numbers, do not use email
# as username
# bitcoin.conf in previous step
vim config/currencies.yml
```

Config database settings

```
vim config/database.yml

# Initialize the database and load the seed data
bundle exec rake db:setup
```

Precompile assets

```
bundle exec rake assets:precompile
```

Run Daemons

```
# start all daemons
bundle exec rake daemons:start

# or start daemon one by one
bundle exec rake daemon:matching:start
...

# Daemon trade_executor can be run concurrently, e.g. below
# line will start four trade executors, each with its own logfile.
# Default to 1.
TRADE_EXECUTOR=4 rake daemon:trade_executor:start

# You can do the same when you start all daemons:
TRADE_EXECUTOR=4 rake daemons:start
```

When daemons don't work, check `log/#{daemon name}.rb.output` or `log/peatio:amqp:#{daemon name}.output` for more information (suffix is '.output', not '.log').

SSL Certificate setting

For security reason, you must setup SSL Certificate for production environment, if your SSL Certificated is been configured, please change the following line at `config/environments/production.rb`

```
config.force_ssl = true
```

Passenger:

```
sudo ln -s /home/bitex/dcl/current/config/nginx.conf /etc/nginx/conf.d/dcl.conf
sudo service nginx restart
```

Liability Proof

```
# Add this rake task to your crontab so it runs regularly
RAILS_ENV=production rake solvency:liability_proof
```

如果出现connection refused,可能是https没有配的原因

解决方法:

1.将config.force_ssl = true 改为false

2.配置https

```
vim /etc/nginx/conf.d/dcl.conf
```

配置https

```
openssl genrsa -des3 -out server.key 1024
```

```
openssl req -new -key server.key -out server.csr
```

```
cp server.key server.key.org
```

```
openssl x509 -req -days 365 -sha256 -in server.csr -signkey server.key -out server
new.crt # # 命令生成v1版证书
```

```
cp servernew.crt /etc/nginx/server.crt
```

```
cp server.key /etc/nginx/server.key
```

开启ssl区域