Deploy production server on CentOS 6.8

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 git-core curl zlib1g-dev build-essential \
                     libssl-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3
 \
                     libxml2-dev libxslt1-dev libcurl4-openssl-dev \
                     python-software-properties libffi-dev

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 sudo yum install readline sudo yum install zlib sudo yum install openssl-devel sudo yum install readline-devel sudo yum install zlib-devel

Install bundler

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

3. Install MySQL

```
sudo yum install mysql-server  mysql-client  libmysqlclient-dev
```

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 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

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.phusio
npassenger.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:

```
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;
```

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 git://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
vim config/application.yml（注释相同三行）
```

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 rm /etc/nginx/sites-enabled/default
sudo ln -s /home/deploy/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
```