

FreeBSD Administration - Support #855

Install Refinery CMS on FreeBSD

09/30/2016 03:39 PM - Daniel Curtis

Status:	Closed	Start date:	09/30/2016
Priority:	Normal	Due date:	
Assignee:	Daniel Curtis	% Done:	100%
Category:	Web Server	Estimated time:	1.50 hour
Target version:	FreeBSD 10	Spent time:	2.00 hours

Description

This is guide for setting up Refinery CMS with Nginx and PostgreSQL on FreeBSD 10.

Prepare the Environment

- Make sure the system is up to date:

```
pkg update && pkg upgrade
```

Install PostgreSQL

- Install postgresql:

```
pkg install postgresql94-server
```

- Initialize, start, and enable postgresql at boot:

```
echo 'postgresql_enable="YES"' >> /etc/rc.conf  
service postgresql initdb  
service postgresql start
```

- Log in to postgresql user account

```
su - pgsq1
```

- Connect to postgresql database

```
psql -d template1
```

- Create a user for RefineryCMS:

```
CREATE USER refineryuser WITH PASSWORD 'SuperSecretPassword' CREATEDB SUPERUSER;
```

- Create the RefineryCMS production database & grant all privileges on database

```
CREATE DATABASE refinerydb OWNER refineryuser encoding='UTF8';
```

- Quit the database session

```
\q
```

```
exit
```

Install Refinery CMS

- Install a few dependencies:

```
pkg install ruby ruby22-gems rubygem-bundler rubygem-rails ImageMagick-nox11 postgresql-libpqx
```

- Install the refinercms gem:

```
gem install refinercms
```

- Create a new refinercms instance:

```
cd /usr/local/www  
refinercms refinery.example.com  
cd refinery.example.com
```

- Edit the Gemfile

```
vi Gemfile
```

- And add the pg gem to the dependencies:

```
gem 'pg'
```

- And run the install bundler again to install the pg gem:

```
bundle install
```

- Copy the sample postgresql database config over as the main database config:

```
cp config/database.yml.postgresql config/database.yml
```

- Then edit the database config:

```
vi config/database.yml
```

- And make the production section is correct:

```
production:  
  adapter: postgresql  
  encoding: unicode  
  host: db.example.com  
  database: refinerydb  
  pool: 5  
  username: refineryuser  
  password: SuperSecretPassword  
  min_messages: warning
```

- Run the database migrations:

```
env RAILS_ENV=production bundle exec rake db:setup
```

- Generate a secret key and copy the contents to the secrets.yml file:

```
env RAILS_ENV=production bundle exec rake secret
```

- Edit the secrets.yml file:

```
vi config/secrets.yml
```

- And replace `<%= ENV["SECRET_KEY_BASE"] %>` with the value generated above:

```
production:
  secret_key_base: 7604e60f8441442aec7ba23bd9139dbfe40403381382c4efeb1a697e476a81c2a9c0864
3d1a1b872dbb2cf1c06f03e9d86f52ad22ff8bab93b74cd5413dc5c15
```

- Precompile javascript and CSS files:

```
env RAILS_ENV=production bundle exec rake assets:precompile
```

- Test RefineryCMS by running the built-in web server:

```
env RAILS_ENV=production rails server -b 0.0.0.0:3000
```

- Open a web browser and go to <http://refinery.example.com:3000/refinery> to add the administrative user.
- Press **Ctrl + C** to kill the server.

Install Nginx

- Add the following to the make.conf file to use nginx while compiling passenger:

```
echo 'OPTIONS_UNSET+= APACHE22 DOCS NLS X11 EXAMPLES' >> /etc/make.conf
echo 'OPTIONS_SET+= PASSENGER SYMLINK NGINX' >> /etc/make.conf
```

- Install nginx with passenger support:

```
portmaster www/nginx
```

- Install Passenger

```
portmaster www/rubygem-passenger
```

- Start and enable nginx at boot:

```
echo 'nginx_enable="YES"' >> /etc/rc.conf
service nginx start
```

- Create a configuration directory to make managing individual server blocks easier

```
mkdir /usr/local/etc/nginx/conf.d
```

- Edit the main nginx config file:

```
vi /usr/local/etc/nginx/nginx.conf
```

- And strip down the config file and add the include statement at the end to make it easier to handle various server blocks:

```
load_module /usr/local/libexec/nginx/nginx_http_passenger_module.so;
load_module /usr/local/libexec/nginx/nginx_http_headers_more_filter_module.so;
load_module /usr/local/libexec/nginx/nginx_stream_module.so;

worker_processes 1;
error_log /var/log/nginx-error.log;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;

    # Load Phusion Passenger module globally
    passenger_root /usr/local/lib/ruby/gems/2.2/gems/passenger;
    passenger_ruby /usr/local/bin/ruby22;
    passenger_max_pool_size 15;
    passenger_pool_idle_time 300;

    # Load config files from the /etc/nginx/conf.d directory
    include /usr/local/etc/nginx/conf.d/*.conf;
}
```

- Add a **refinery.example.com** server block:

```
vi /usr/local/etc/nginx/conf.d/refinery.example.com.conf
```

- Add the following:

```
server {
    listen 80;
    server_name refinery.example.com;
    root /usr/local/www/refinery.example.com/public;
    access_log /var/log/refinery.example.com-access.log;
    error_log /var/log/refinery.example.com-error.log;

    passenger_enabled on;
    passenger_user www;
    passenger_group www;
}
```

- Restart nginx:

```
service nginx restart
```

- Open a web browser and go to <http://refinery.example.com>.

Resources

- <http://www.refinerycms.com/guides/installation-prerequisites>
- <http://www.refinerycms.com/guides/getting-started>
- <https://github.com/refinery/refinerycms>

History

#1 - 09/30/2016 03:53 PM - Daniel Curtis

- *Description updated*
- *Status changed from New to In Progress*
- *% Done changed from 0 to 50*

#2 - 10/01/2016 12:15 AM - Daniel Curtis

- *Status changed from In Progress to Resolved*
- *% Done changed from 50 to 100*

#3 - 12/03/2016 12:49 PM - Daniel Curtis

- *Status changed from Resolved to Closed*