PBCore Repository Installation

From WNET Archive Wiki

Here is how to install the PBCore Record Repository on a CentOS server.

Note: this is technical documentation intended for systems administrators already familiar with Linux and CentOS. As with any recipe, read the whole thing before proceeding.

Contents

Install prerequisite software

Our front-end webserver will be nginx (http://wiki.nginx.org/Main) . So, if the machine in question is already running Apache httpd, get rid of it:

```
service httpd stop
yum remove httpd
```

(You could also use apache httpd instead, but doing so is a bit tricky and prone to performance woes.)

Now, install the packages we'll need:

```
rpm -i http://download.fedora.redhat.com/pub/epel/5/x86_64/epel-release-5-3.noarch.rpm
yum install nginx git libxml2-devel.x86_64 mysql.x86_64 mysql-server mysql-devel.x86_64 gcc gcc-c++
```

Next, install ruby (http://www.ruby-lang.org/) **1.8.6** or later. Version 1.8.5, as provided with CentOS 5, is **not** sufficient. If you manage to find some alternate source for suitable RPM packages, install `ruby ruby-devel ruby-irb ruby-libs ruby-rdoc` and, optionally, `ruby-docs`

Only after installing this suitable version of Ruby, install rubygems (http://www.rubygems.org/) either from its website or by doing `yum install rubygems`

Next, download Sphinx (http://www.sphinxsearch.com/) . Get the latest version in the

0.9.8 series. There may be compatibility issues with the 0.9.9 series. Install by doing the standard

```
./configure && make && make install
```

And, finally, install some needed ruby packages by doing

```
gem install chronic libxml-ruby rake rack thin packet
gem install mysql -- --with-mysql-config=/usr/bin/mysql_config
```

Configuring the prerequisites

Configure MySQL

If you're concerned about security, edit `/etc/my.cnf`, adding the line

```
bind-address = 127.0.0.1
```

to the `[mysqld]` section. Then, start mysqld and make sure it stays started:

```
/etc/init.d/mysqld start
chkconfig mysqld on
```

Now would be a good time to set a mysql password for the root user if you've not done so already.

Create the database we'll be using, and a user for the web application to use:

```
mysqladmin create pbcore
echo "GRANT ALL PRIVILEGES ON pbcore.* TO pbcore@localhost IDENTIFIED BY 'my_secure_password'" | mysq
```

If you are migrating a database from an existing install, now's the time to copy the data over. If this is to be a fresh install, we'll set up the schema later.

Configure nginx

Here's a sample nginx configuration file. Download it, modify it (or don't), and stick it in `/etc/ngnix.conf`.

Then start `nginx` and make sure it stays start:

```
/etc/init.d/nginx start
chkconfig nginx on
```

Create a user

We don't want our software to run as either root or nginx, so create a user for it to run as:

```
useradd www
```

Then (this is important) copy **your** SSH public key into /home/www/.ssh/authorized_keys. Make sure that this file has permissions mode 0700 and that both the file and its containing directory are owned by the www user.

Install the repository software

On the server make a directory /var/www/pbcore where the software will live.

We use capistrano (http://www.capify.org/) for deployment, so now you will have to do some things on your **local machine**. The following instructions assume your local machine is POSIX (GNU/Linux or Mac OS X). It may somehow be possible to do all this on Windows, perhaps with cygwin (http://www.cygwin.com/) , but this is untested.

If you don't already have them, install ruby, rubygems, and git (http://git-scm.com/) . Get capistrano by doing

```
sudo gem install capistrano capistrano-ext
```

Now, grab a copy of the software (again, on your **local machine**):

```
git clone git://git.mlcastle.net/pbcore.git pbcore; cd pbcore
```

Inside the pbcore directory, you'll see a directory config/deploy with a file in it called site.rb.example (http://git.mlcastle.net/?p=pbcore.git;a=blob;f=config/deploy/site.rb.example;hb=HEAD) . Copy this to some file reminding you of your server (I used wnet.rb) and then edit the copy: put the name of your server in all three role lines and configure the :app_db_pass to match the MySQL password you assigned above. If you're a capistrano wizard, you can also of course put any other settings changes you like in here.

Now, ask your local machine to create the proper directory structure on the server by doing the following command from the pbcore diretory:

```
cap my_site deploy:setup
```

In that command and all cap commands which follow, replace my_site with the name of the site configuration file you set up, without the .rb extension. So, I would do cap wnet deploy:setup.

If you're migrating an existing installation of the repository, copy /var/www/pbcore/shared/configs/site_key.txt (including the permissions thereupon) from the old server to the new one; otherwise, you will not be able to log in to the repository. If this is a new install, a new site_key.txt will be automatically created and you don't have to worry about it.

Now it's time to ask your local machine to install the software to the server. This will also set up or upgrade the database schema if necessary:

```
cap my_site deploy:migrations
```

Now, ask the search server to index the data (even if there is no data yet, it will create an empty index). Then, start everything up:

```
cap my_site sphinx:index
cap my_site deploy:start
```

Things can be a bit hairy on the first deployment, so you might have to manually start one or more services:

```
cap my_site thin:start sphinx:start backgroundrb:start
```

(If you'd like to have a list of all defined `cap` tasks, just do `cap -T`).

You should now be able to browse and use the repository!

Recurrent and exceptional tasks

Sphinx keeps two search indices: one with the bulk of the data, and one with all the data which has changed since the last full reindex. If you don't do a full reindex every so often, this latter bit will get quite large and take a long time to regenerate every time anything in the database changes. So, set up a cronjob to reindex the database every so often by doing (as root on the server):

```
echo '56 10,22 * * * www (cd /var/www/pbcore/current ; rake RAILS_ENV=production ts:in) > /dev/null'
```

Now, we want to make sure things come back up properly should the server ever be rebooted. CentOS provides init scripts for `nginx` and MySQL, but we need one for thin. Download this one, install it, and then

```
chkconfig thin on
```

Upgrading

If changes are pushed to the master PBCore repository and you want them for your installation, just go to the directory on your local machine into which you checked out the repository and do

```
git pull
cap my_site deploy
```

If there were any changes to the database schema (known as *migrations* in Rails), replace the second line with

```
cap my_site deploy:migrations
```

If you're unsure, it's safe to do this on every deploy.

Retrieved from "http://wnet-wiki.thirteen.org/wiki/PBCore_Repository_Installation"
Categories: PBCore | Applications

- This page was last modified on 12 May 2009, at 16:50.