

Contents

- 1 Current System Installation Environment
- 2 Other Prerequisites
- 3 Individual Package Installation Notes
 - 3.1 PostgreSQL
 - 3.2 ViewCVS
 - 3.3 JPGraph
 - 3.4 PHP
- 4 Installation Technique #1: Install GForge First
 - 4.1 GForge
 - 4.2 Update Database
 - 4.3 Update Configuration Files
 - 4.4 Install NITRC Source
 - 4.5 More Documentation
- 5 Installation Technique #2: Start with Existing Database
- 6 Nutch Installation
 - 6.1 Prerequisites: Java
 - 6.1.1 Optional: set up dnsmasq
 - 6.2 Implementing a NITRC Nutch Installation
 - 6.2.1 Install NITRC-customized Nutch Java Application
 - 6.2.2 Update rc.local
 - 6.2.3 Open up iptables (firewall)
 - 6.2.4 Configure Install
 - 6.2.5 Test the Install
 - 6.2.6 Point NITRC back to your Nutch Install
 - 6.2.7 Finalize Install
- 7 Additional NITRC Configuration
 - 7.1 Limiting VSFTPD sessions
 - 7.2 Limiting HTTPD sessions
 - 7.3 Federated Data Site Maintenance and Review
 - 7.4 Disable Mailman Monthly Report

Current System Installation Environment

- CentOS 4.3
- PostgreSQL 8.1.11
- Apache/2.0.52
- PHP 5.1.6

Other Prerequisites

- Package prerequisites:
 - mailman
 - mod_dav_svn
 - subversion
 - fonts-xsorg-truetype
- External prerequisites:
 - JGraph: (<http://www.aditus.nu/jpgraph/>)
 - http://www.aditus.nu/jpgraph/old_versions/jpgraph-1.13.tar.gz
 - IPC Run for Perl
 - <http://gforge.org/frs/download.php/143/IPC-Run-0.78.tar.gz>
 - ViewCVS:
 - <http://gforge.org/frs/download.php/152/viewvc-1.0-dev-gforge.tar.bz2>
 - mod-auth-gforge:
 - <http://gforge.org/frs/download.php/204/libapache2-mod-auth-gforge-0.5.9.3-CENTOS41.1.i386.rpm>

Individual Package Installation Notes

PostgreSQL

```
In /var/lib/pgsql/data/pg_hba.conf, comment out:
```

```
#local all all ident sameuser
```

```
and add:
```

```
local all all trust
```

```
Restart postgres
```

```
[root]# /etc/init.d/pgsql restart
```

```
Created user gforge/**** for postgres (and added PL/PGSQL as a language):
```

```
[root]# adduser gforge
[root]# chown -R gforge:gforge /home/gforge
[root]# su - postgres
[postgres]# createuser -A -d -E -P gforge
(entered password)
[postgres]$ createlang plpgsql template1
[postgres]$ exit
```

```
Created database gforge
```

```
[root]# su - gforge
[gforge]# createdb gforge
[gforge]# exit
```

ViewCVS

```
wget http://gforge.org/frs/download.php/152/viewvc-1.0-dev-gforge.tar.bz2
tar -jxf viewvc-1.0-dev-gforge.tar.bz2
cd viewvc-1.0-dev-gforge.tar.bz2
* ran ./viewcvs-install
* told it to install into /opt/viewvc
* edited /opt/viewvc/viewcvs.conf:
root_parents = /svnroot : svn
cvs_roots = cvs: /cvsroot
docroot = /plugins/scmsvn/viewcvs/
```

JPGraph

```
wget http://www.aditus.nu/jpgraph/old_versions/jpgraph-1.13.tar.gz
tar -zxf jpgraph-1.13.tar.gz
* Installed JPGraph jpgraph-1.13.tar.gz
cd jpgraph-1.13
mv src /usr/lib/jpgraph
Edit /usr/lib/jpgraph/jpgraph.php:
Set USE_CACHE false
Set READ_CACHE false
```

PHP

```
update /etc/php.ini:
register_globals = On
magic_quotes_gpc = On
file_uploads = On
upload_max_filesize = 1024M
post_max_size = 1024M
include_path = ".:usr/lib/gforge/:usr/lib/gforge/www/include:/etc/gforge/"
```

Installation Technique #1: Install GForge First

There are a number of nasty things GForge does to the Linux environment (such as copy users from /etc/passwd to /etc/passwd.org, run a number of cron jobs including one that automatically regenerates /etc/passwd, and set up symbolic links for plugins). A simple way to make sure this works is by first installing GForge 4.5.11 and then upgrading to our version of NITRC.

GForge

- Main package: <http://gforge.org/frs/download.php/219/gforge-4.5.11.tar.bz2>

```
Untar gforge into some directory
[root]# wget http://gforge.org/frs/download.php/219/gforge-4.5.11.tar.bz2
[root]# tar -jxf gforge-4.5.11.tar.bz2
[root]# cd gforge-4.5.11.tar.bz2
```

Populate the database

```
[root]# psql -U postgres gforge < db/gforge.sql > import.log
```

As root, run the gforge install scripts (replace www.nitrc.org and the IP address with your information)

```
[root]# ./gforge-install.sh www.nitrc.org apache apache 132.239.131.178
```

This moves gforge to /usr/lib/gforge

It also creates a /etc/gforge/httpd.conf

Make a symbolic link from /etc/httpd/conf.d/gforge.conf to
/etc/gforge/httpd.conf:

```
[root]# ln -s /etc/gforge/httpd.conf /etc/httpd/conf.d/gforge.conf
```

Edit /etc/gforge/httpd.conf

* Added (for mailman after Alias /pipermail):

DirectoryIndex index.html index.htm

Install mod_auth_gforge:

```
[root]# rpm -i libapache2-mod-auth-gforge-0.5.9.3-CENTOS41.1.i386.rpm
```

Restart HTTPD:

```
[root]# /etc/init.d/httpd restart
```

Log in to the web site and create the user "admin"

Made admin user an administrator

```
[gforge]# psql -U gforge gforge
```

```
psql> SELECT user_id FROM users WHERE user_name='admin';
```

```
psql> insert into user_group (user_id,group_id,admin_flags) values (102,1,'A');
```

```
psql> \q
```

Had to update:

```
* /usr/lib/gforge/plugins/scmcvs/bin/snapshots.sh:
```

```
CVSROOT=/var/lib/gforge/cvsroot
```

```
* /usr/lib/gforge/plugins/scmsvn/bin/snapshots.sh:
```

```
SVNROOT=/var/lib/gforge/svnroot
```

```
* /etc/gforge/plugins/scmcvs/config.php:
```

```
$sys_default_domain='www.nitrc.org';
```

```
$GLOBALS['sys_default_domain']=$sys_default_domain;
```

```
$default_cvs_server = $GLOBALS['sys_default_domain'] ;
```

```
* /etc/gforge/plugins/scmcvs/cvsweb.conf:
```

```
@CVSrepositories = (
```

```
'gforge' => ['GForge-CVS',          '/var/lib/gforge/cvsroot/'],
```

had to:

```
chmod 755 /etc/gforge/viewcvs.cgi
```

```
# Fix a series of symbolic links to gforge plugins
rm /etc/gforge/cvsweb
ln -s /usr/lib/gforge/plugins/scmcvs/cgi-bin/cvsweb /etc/gforge/cvsweb
chmod 755 /usr/lib/gforge/plugins/scmcvs/cgi-bin/cvsweb
rm -Rf /etc/gforge/plugins/cvstracker
ln -s /usr/lib/gforge/plugins/cvstracker/etc/plugins/cvstracker /etc/gforge/plugins/cvstracker
rm -Rf /etc/gforge/plugins/scmcvs
ln -s /usr/lib/gforge/plugins/scmcvs/etc/plugins/scmcvs /etc/gforge/plugins/scmcvs
rm -Rf /etc/gforge/plugins/scmsvn
ln -s /usr/lib/gforge/plugins/scmsvn/etc/plugins/scmsvn /etc/gforge/plugins/scmsvn
```

Update Database

In NITRC/promotion/dbscripts you will find a series of upgrade scripts (starting with Upgrade_gforge-4-5-11_to_v1.0.0-1.sql). These need to be run in sequence until you reach the goal version. Please note that some baselines did not have database changes, and therefore do not have scripts associated with them. In addition, there were "hotfixes" that have a slightly different upgrade filename. Please run all the scripts in numerical order of the specified version numbers.

A script is run as:

```
psql -U gforge < Upgrade_gforge-4-5-11_to_v1.0.0-1.sql
```

Update Configuration Files

In NITRC/promotion/docs, there are a series of Upgrade documents that list configuration modifications that need to be done manually. Some items can be ignored (such as sftp access) depending upon the criticality of the duplication requirements).

Install NITRC Source

In NITRCSRC/promotion/package, you will find a Makefile. Setting the TAG to the latest tagged version of NITRC (or HEAD) and running Make will create a tar ball of the latest source and the needed symbolic links designed to be installed in /usr/lib/gforge. You can then cd to "/usr/lib/gforge" and then un-tar (tar -zxf) the contents right into /usr/lib/gforge.

More Documentation

It should be noted that a step-by-step of what we did to install NITRC 1.0.3-2 onto a Rocks (UCSD cluster) machine is located at NITRCSRC/installation/FullInstallNotes.txt

Installation Technique #2: Start with Existing Database

Using an existing functional NITRC system as a source, it should be possible to replicate it without installing gForge from scratch. This would involve:

- Copying /etc/passwd, /etc/shadow, and /etc/groups to *.org files so that auto-generation of users functions

- Copying the database image and other files (see nitrcint:Migrate data from production to stage)
- Creating and untarring the contents of the NITRC source into /usr/lib/gforge (see instructions above)
- Copying configuration files from NITRCSRC/installation/config to their correct locations (and editing system name and IP address as needed)
- Installing NITRCSRC/installation/root.crontab.txt as root's crontab (as needed)
- Creating /var/lib/gforge as on the original system
- Adding symbolic links for /cvsroot /svnroot /etc/gforge as on the original system

Nutch Installation

Prerequisites: Java

Nutch has one main prerequisite: Java 1.6 or greater

Perform the following steps to install Java on Linux.

1. Get the desired JDK RPM files from Sun at <http://developers.sun.com/downloads/> . For example, for JDK 1.6 download the file: `jdk-6u12-linux-i586-rpm.bin` .
2. Make the file executable via:

```
chmod 755 jdk-6u12-linux-i586-rpm.bin
```

3. Execute the install script via:

```
slide ./jdk-6u12-linux-i586-rpm.bin
```

Note: By default the java files will be installed in /usr/java . Hence, the JAVA_HOME for this installation will be in /usr/java/jdk1.6.0_12 .

4. Run `java -version` to verify the installation.
5. Set the Java_Home (e.g. `export JAVA_HOME=/usr/local/jdk-1.5.0.05`).

Optional: set up dnsmasq

dnsmasq allows caching of DNS requests so that nutch doesn't overload the local DNS servers.

For Linux based installs run:

```
yum install dnsmasq
```

edit /etc/dnsmasq.conf (so that 127.0.0.1 is used)

Add 127.0.0.1 to /etc/resolv.conf

Implementing a NITRC Nutch Installation

Install NITRC-customized Nutch Java Application

- Download <http://int.nitrc.org/packages/nitrc-nutch.tar.gz>
 - This is a Nutch 0.9 installation with customizations necessary to allow tools to be associated with search results (the customizations are in CVS under NITRCSRC/nutch)
 - Create /usr/local/nutch with 755 permissions
 - `untar (tar -zxf) nitrc-nutch.tar.gz` into /usr/local/nutch
- Download <http://int.nitrc.org/packages/nitrc-tomcat.tar.gz>
 - This is a standard tomcat installation with the customized nutch files deployed in it
 - Create /usr/local/tomcat with 755 permissions
 - `untar (tar -zxf) nitrc-tomcat.tar.gz` into /usr/local/tomcat
- Create /var/lib/nutch with 755 permissions
- Create /var/lib/nutch/seed with 755 permissions
- Set up /opt/nutch as a symbolic link to /usr/local/nutch
 - `ln -s /usr/local/nutch /opt/nutch`
- Set up /opt/tomcat as a symbolic link to /usr/local/tomcat
 - `ln -s /usr/local/tomcat /opt/tomcat`

Update rc.local

```
Add the following to /etc/rc.local

# Start nutch search engine
export JAVA_HOME=/usr/lib/java/
/usr/local/tomcat/bin/startup.sh
```

Open up iptables (firewall)

If you have iptables installed, enter the following line to the iptables config file:

```
# public NITRC Nutch access
-A INPUT -m state --state NEW -p tcp --dport 9080 -j ACCEPT
```

Configure Install

- Update /opt/nutch/do_crawl_once to grab the seeds from your NITRC installation

- specifically change the line: `wget http://www.nitrc.org/search/nutch_seeds.php ...`

- Start tomcat:

```
export JAVA_HOME=/usr/lib/java/  
/usr/local/tomcat/bin/startup.sh
```

- **ALSO: Change the "rm -Rf /var/lib/nutch/crawl2; ./bin/nutch crawl /var/lib/nutch/seed/ ..." line to only use a "-depth" of 1, not 5. Otherwise your test run will take a long time and you may annoy your clients with deep crawls**
- Run the crawl once:

```
/opt/nutch/do_crawl_once
```

Test the Install

The default setup creates a search service at port 9080, for example:

`http://nutch-stage.nitrc.org:9080/nutch-0.9/search.jsp?query=tools&hitsPerPage=10&lang=en`

You can hit the service at port 9080 to test to make sure Tomcat is working and that it correctly connects to your crawled database.

Point NITRC back to your Nutch Install

On the NITRC box, update `/etc/gforge/local.inc` and set `$sys_nutch_installation` to point to your new Nutch install.

Finalize Install

```
Correct /opt/nutch/do_crawl_once to use a depth of 5
```

To have nutch continuously crawl (only suggested for production) add the following line to root's crontab:

```
*/5 * * * * /opt/nutch/do_crawl 1>/dev/null
```

Additional NITRC Configuration

Limiting VSFTPD sessions

We configured vsftpd over xinet.d for NITRC. In order to avoid bots trashing the system, we set up user ftp with no directory listing ability: `"dirlist_enable=NO"` and `"per_source = 2"` in `/etc/xinet.d/vsftpd`.

Limiting HTTPD sessions

To limit bots on web transactions, we configured <http://dominia.org/djao/limitipconn2.html> as directed, with

```
LoadModule limitipconn_module /usr/lib/httpd/modules/mod_limitipconn.so

ExtendedStatus On

<IfModule mod_limitipconn.c>
# Set a server-wide limit of 10 simultaneous downloads per IP,
# no matter what.
MaxConnPerIP 10
<Location /frs>
# This section affects all files under http://your.server/somewhere
MaxConnPerIP 3
# exempting images from the connection limit is often a good
# idea if your web page has lots of inline images, since these
# pages often generate a flurry of concurrent image requests
NoIPLimit themes/*
</Location>
</IfModule>
```

Federated Data Site Maintenance and Review

- Technical Information on NITRC Federated Data

Disable Mailman Monthly Report

To reduce mail sent to all tool administrators, we disabled the monthly mailman spam report

- Edit `/etc/cron.d/mailman`
 - Comment out `0 5 1 * * mailman /usr/lib/mailman/cron/mailpasswds`