# FreeBSD Administration - Support #624

## Install GitLab 7.11 on FreeBSD

06/08/2015 01:05 PM - Daniel Curtis

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | 09/21/2014 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Daniel Curtis | **% Done:** | 100% |
| **Category:** | Source Code Management | **Estimated time:** | 2.00 hours |
| **Target version:** | FreeBSD 9 | **Spent time:** | 5.00 hours |

**Description**

# Setting up the Environment

- Start by making sure everything is up to date:

```
pkg update && pkg upgrade
portsnap fetch extract
pkg2ng
```

- Install portmaster:

```
cd /usr/ports/ports-mgmt/portmaster
make install clean
```

- Install a few dependencies

```
portmaster devel/git databases/redis lang/ruby21 security/krb5 devel/icu textproc/libxml2 text
proc/libxslt lang/python2 shells/bash security/sudo lang/gcc48 devel/gmake devel/autoconf deve
l/automake devel/libtool devel/bison devel/readline textproc/libyaml databases/sqlite3 databas
es/gdbm devel/cmake databases/postgresql94-client databases/postgresql-libpqxx sysutils/rubyge
m-bundler devel/libgit2 security/libssh2
```

- Add the GitLab user

```
pw add user -n git -m -s /usr/local/bin/bash -c "GitLab"
```

# Install Redis

- Start Redis and enable it to start at boot:

```
echo 'redis_enable="YES"' >> /etc/rc.conf
service redis start
```

- Add the following to /usr/local/etc/redis.conf:

```
unixsocket /var/run/redis/redis.sock
unixsocketperm 770
```

- Add git to the redis group:

```
pw usermod git -G redis
```

## Install PostgreSQL

- This environment will be setup with PostgreSQL 9.4:

```
portmaster databases/postgresql94-server
```

- Login to PostgreSQL:

```
sudo -u postgres psql -d template1
```

  - Create a user for GitLab:

```
CREATE USER git CREATEDB;
```

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

```
CREATE DATABASE gitlabhq_production OWNER git;
```

  - Quit the database session

```
\q
```

- Try connecting to the new database with the new user

```
sudo -u git -H psql -d gitlabhq_production
```

- Quit the database session

```
\q
```

# Install GitLab 7.11

- Change to the git user home directory and download GitLab 7.11

```
su - git
cd
git clone https://gitlab.com/gitlab-org/gitlab-ce.git -b 7-11-stable gitlab
```

- Create a config file

```
cd /home/git/gitlab
cp config/gitlab.yml.example config/gitlab.yml
```

- Edit the gitlat config file:

```
vi config/gitlab.yml
```

    - Change the gitlab: host: to git.example.com
    - If using https change the gitlab: port: from 80 to **443**
    - Change the git: bin_path: to **/usr/local/bin/git**
    - Change the satellites: path: to **/usr/home/git/gitlab-satellites/**
    - Change the gitlab_shell: path: to **/usr/home/git/gitlab-shell/**
    - Change the gitlab_shell: repos_path: to **/usr/home/git/repositories/**
    - Change the gitlab_shell: hooks_path: to **/usr/home/git/gitlab-shell/hooks/**

- Create a unicorn.rb file:

```
cp config/unicorn.rb.example config/unicorn.rb
```

- Create a rack_attack.rb file:

```
cp config/initializers/rack_attack.rb.example config/initializers/rack_attack.rb
```

- Create a database configuration file:

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

- Configure the GitLab system user:

```
git config --global user.name "GitLab"
git config --global user.email "example@example.com"
```

- Setup the directories and permissions:

```
chown -R git log/
chown -R git tmp/
chmod -R u+rwX log/
chmod -R u+rwX tmp/
chmod -R u+rwX tmp/pids/
chmod -R u+rwX tmp/sockets/
chmod -R u+rwX  public/uploads
```

- Create the gitlab-satellites:

```
mkdir /home/git/gitlab-satellites
chmod u+rwx,g=rx,o-rwx /home/git/gitlab-satellites
```

- Configure Redis connection settings

```
cp config/resque.yml.example config/resque.yml
```

- Configure bundler to build nokogiri using system libraries :

```
bundle config build.nokogiri "--use-system-libraries"
```

    - **NOTE**: This is required to prevent an error building nokogiri during install

- Configure bundler to build rugged using system libraries :

```
bundle config build.rugged "--use-system-libraries"
```

- **NOTE**: This is required to prevent an error building rugged during install

- Install GitLab gems

```
bundle install --deployment --without development test mysql aws
```

- **NOTE**: I needed to rerun the bundle install command as root to install the rugged gem without problems.

## Install GitLab Shell

- Run the installation task for gitlab-shell (replace `REDIS_URL` if needed)

```
bundle exec rake gitlab:shell:install[v2.6.3] REDIS_URL=redis://localhost:6379 RAILS_ENV=produ
ction
```

By default, the gitlab-shell config is generated from your main gitlab config.

- You can review (and modify) the gitlab-shell config as follows:

```
vi /home/git/gitlab-shell/config.yml
```

- **NOTE**: I needed to set the gitlab_url parameter to http://localhost:8080 for the gitlab-shell API backend to work properly.

- Initialize Database and Activate Advanced Features

```
bundle exec rake gitlab:setup RAILS_ENV=production
```

- Verify that the install worked:

```
bundle exec rake gitlab:env:info RAILS_ENV=production
```

- Precompile the assets used by GitLab:

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

- Exit out of the GitLab user, back into the root account:

```
exit
```

- Create an init script

```
cp /usr/home/git/gitlab/lib/support/init.d/gitlab /usr/local/etc/rc.d/gitlab
```

- Start GitLab

```
service gitlab start
```

- Enable GitLab to start at boot

```
echo 'gitlab_enable="YES"' >> /etc/rc.conf
```

# GitLab on Nginx

## Install Nginx

- Install Nginx

```
portmaster www/nginx
```

  - **NOTE**: Make sure to enable **[X]PASSENGER** during the port configuration

- Start and enable nginx at boot:

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

## Install Phusion Passenger

- Reinstall Nginx with Passenger support

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

  - **NOTE**: Make sure to enable **[X]NGINX** during the port configuration
  - *Optional*: Enable **[X]SYMLINK** during the port configuration to make upgrading passenger easier later on.

- Edit the main nginx config file:

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

  - And add the Passenger config parameters:

```
#user  nobody;
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;
  #tcp_nopush     on;

  #keepalive_timeout  0;
  keepalive_timeout  65;

  #gzip  on;

  # Load Phusion Passenger module globally
  passenger_root /usr/local/lib/ruby/gems/2.1/gems/passenger;
  passenger_ruby /usr/local/bin/ruby21;
  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;

}
```

- **NOTE**: The above config assumes [X]SYMLINK was enabled during the port configuration. Match the config according to which ever version of passenger is installed at /usr/local/lib/ruby/gems/2.1/gems.

- Now create a git server block:

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

  - And add the following:

```
server {
  listen 80;
  server_name git.example.com;
  # enforce https
  return 301 https://$server_name$request_uri;
}

server {
  listen 443 ssl;
  server_name git.example.com;

  ssl_certificate /usr/local/etc/nginx/ssl/git.example.com.crt;
  ssl_certificate_key /usr/local/etc/nginx/ssl/git.example.com.key;

  root /home/git/gitlab/public;
  passenger_enabled on;
  passenger_user git;
  passenger_group git;

  # set max upload size
  client_max_body_size 1G;
  fastcgi_buffers 64 4K;
}
```

# GitLab on Apache

Running GitLab from an apache server using phusion passenger is simple.

- Edit the virtual host definition and add/modify it according to your needs:

```
vi /usr/local/etc/apache24/Vhosts/git.example.com.conf
```

  - And add/edit the following virtual host definition:

```
<VirtualHost *:80>
  DocumentRoot "/home/git/gitlab/public"
  ServerName git.example.com
  ErrorLog "/var/log/git.example.com-error_log"
  CustomLog "/var/log/git.example.com-access_log" common
  PassengerRuby /usr/local/bin/ruby

  <Directory /home/git/gitlab/public>
      Options -MultiViews
      AllowOverride All
      Order allow,deny
```

```
        Allow from all
        Require all granted
      </Directory>
    </VirtualHost>
```

- Restart apache:

```
service apache24 restart
```

- Start and enable apache24 at boot:

```
echo 'apache24_enable="YES"' >> /etc/rc.conf
service apache24 start
```

# Initial Login

The default username and password:

- user: **root**
- pass: **5iveL!fe**

# Resources

http://luizgustavo.pro.br/blog/2014/08/21/instalacao-gitlab-no-freebsd/
https://gitlab.com/gitlab-org/gitlab-ce/blob/7-8-stable/doc/install/installation.md
https://gitlab.com/gitlab-org/gitlab-ce/blob/7-8-stable/doc/install/database_mysql.md

**Related issues:**

| | | |
|---|---|---|
| Copied from FreeBSD Administration - Support #584: Install GitLab 7.8 on FreeBSD | **Closed** | **09/21/2014** |

### History

**#1 - 06/08/2015 01:05 PM - Daniel Curtis**

*- Copied from Support #584: Install GitLab 7.8 on FreeBSD added*

**#2 - 06/08/2015 03:23 PM - Daniel Curtis**

*- Description updated*

**#3 - 06/09/2015 10:59 AM - Daniel Curtis**

*- Description updated*

*- Status changed from New to In Progress*

*- % Done changed from 0 to 50*

**#4 - 06/09/2015 02:45 PM - Daniel Curtis**

*- Description updated*

*- % Done changed from 50 to 90*

**#5 - 06/09/2015 05:10 PM - Daniel Curtis**

*- Description updated*

*- Status changed from In Progress to Resolved*

*- % Done changed from 90 to 100*

**#6 - 06/12/2015 10:33 AM - Daniel Curtis**

*- Status changed from Resolved to Closed*

**#7 - 09/02/2015 08:51 PM - Daniel Curtis**

*- Description updated*

**#8 - 09/21/2015 10:40 AM - Daniel Curtis**

*- Description updated*