

CentOS6.5安装gitlab

GitLab, 是一个利用 Ruby on Rails 开发的开源应用程序, 实现一个自托管的Git项目仓库, 可通过Web界面进行访问公开的或者私人项目。

它拥有与Github类似的功能, 能够浏览源代码, 管理缺陷和注释。可以管理团队对仓库的访问, 它非常易于浏览提交过的版本并提供一个文件历史库。团队成员可以利用内置的简单聊天程序 (Wall) 进行交流。它还提供一个代码片段收集功能可以轻松实现代码复用, 便于日后有需要的时候进行查找。

开源项目地址: <https://github.com/gitlabhq/gitlabhq>

```
# Distribution      : CentOS release 6.7 (Final)
# GitLab version    : 7.4.5
# GitLab-shell      : 2.7.4
# Ruby version      : ruby 2.3.0
# Gem version       : 3.5.0
# Redis-server     : Redis server version 2.4.10 (00000000:0)
# Web Server        : Nginx/1.6.2
# Database          : MySQL/5.6.22
```

一, 安装源和依赖包

```
cd /usr/local/src
#增epel源,如果你是i686系统,请把x86_64修改下。
wget -O /etc/rpm/gpg/RPM-GPG-KEY-EPEL-6 https://www.fedoraproject.org/static/0608B895.txt
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm

#确认是否安装成功
rpm -qa gpg*
gpg-pubkey-0608b895-4bd22942

#增puias源
wget -O /etc/yum.repos.d/PUIAS_6_computational.repo https://gitlab.com/gitlab-org/gitlab-recipes/raw/master/install/centos/PUIAS_6_computational.repo
wget -O /etc/pki/rpm-gpg/RPM-GPG-KEY-puias http://springdale.math.ias.edu/data/puias/6/x86_64/os/RPM-GPG-KEY-puias
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-puias

#验证是否成功
rpm -qa gpg*
gpg-pubkey-41a40948-4ce19266
```

查看仓库源

```
yum repolist
已加载插件: fastestmirror
Loading mirror speeds from cached hostfile
* PUIAS_6_computational: www.puias.princeton.edu
* epel: ftp.sjtu.edu.cn
仓库标识                                仓库名称
状态
PUIAS_6_computational                    PUIAS computational Base 6 -
x86_64                                    2,638
base                                       CentOS-6 - Base - 163.com
6,518
*epel                                      Extra Packages for Enterpris
e Linux 6 - x86_64                        11,148
extras                                    CentOS-6 - Extras - 163.com
36
updates                                   CentOS-6 - Updates - 163.com
727
repolist: 21,067
```

如果此列表中没有epel和PUIAS_6_computational, 请执行以下命令

```
yum-config-manager --enable epel --enable PUIAS_6_computational
```

二, 安装依赖包

```
yum -y update
yum -y upgrade
yum -y groupinstall 'Development Tools'
yum -y install nodejs readline readline-devel ncurses-devel gdbm-devel glibc-devel tcl-devel openssl-devel curl-devel expat-devel db4-devel byacc sqlite-devel libyaml libyaml-devel libffi libffi-devel libxml2 libxml2-devel libxslt libxslt-devel libicu libicu-devel system-config-firewall-tui redis sudo wget crontabs logwatch logrotate perl-Time-HiRes git cmake libcom_err-devel.i686 libcom_err-devel.x86_64
```

安装邮件服务

postfix或者sendmail, 官网上安装是postfix, 本人默认就装来sendmail, 所以此步骤省略

三, 安装git

默认centos的git版本是2.7.4, 所以要先删除, 然后再下载源码安装

```
yum remove git -y

yum install zlib-devel perl-CPAN gettext curl-devel expat-devel gettext-devel openssl-devel

mkdir /tmp/git && cd /tmp/git

wget https://www.kernel.org/pub/software/scm/git/git-2.7.4.tar.gz && tar zxvf git-2.7.4.tar.gz

cd git-2.7.4/ && make prefix=/usr/local/git all && make prefix=/usr/local/git install && ln -s /usr/local/git/bin/* /usr/bin/
```

完成后验证

```
which git
/usr/bin/git

git --version
git version 2.7.4
```

四, 安装ruby

ruby版本需要2.0+, 所以先卸载系统已存在的

```
yum remove ruby
#如果是源码安装的
cd (your-ruby-source-path) && make uninstall

mkdir /home/source && cd /home/source

wget https://cache.ruby-lang.org/pub/ruby/2.3/ruby-2.3.0.tar.gz && tar zxvf ruby-2.3.0.tar.gz

cd ruby-2.3.0 && ./configure --disable-install-rdoc && make && make prefix=/usr/local install
```

完成后验证

```
which ruby
/usr/local/bin/ruby

ruby -v
ruby 2.3.0p0 (2015-12-25 revision 53290) [x86_64-linux]
```

安装bundler

```
gem install bundler --no-ri --no-doc
```

五, 创建系统用户

```
adduser --system --shell /bin/bash --comment 'GitLab' --create-home --home-dir /home/git/ git
```

增加/usr/local/bin

```
visudo
#修改以下内容
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin
```

六, 安装mysql

官网给出两个选择, mysql、postgreSQL, 我使用的mysql

centos默认会安装5.1版本的mysql, 所以要源码安装, 此步骤掠过。。。

```
#mysql的版本至少5.5.14或更新
mysql --version
mysql Ver 14.14 Distrib 5.6.22, for Linux (x86_64) using EditLine wrapper
```

创建数据库用户并授权

```
mysql -u root -p

mysql> CREATE USER 'git'@'localhost' IDENTIFIED BY 'gitpwd';

mysql> show variables like "%engine";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| default_storage_engine | InnoDB |
| storage_engine | InnoDB |
+-----+-----+
2 rows in set (0.01 sec)

#如果不是InnoDB引擎, 需执行下面命令
mysql> SET storage_engine=INNODB;
```

创建数据库

```
mysql> CREATE DATABASE IF NOT EXISTS `gitlabhq_production` DEFAULT CHARACTER SET `utf8` COLLATE `utf8_unicode_ci`;
```

给用户授权

```
mysql> GRANT SELECT, LOCK TABLES, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON `gitlabhq_production`.* TO 'git'@'localhost';
mysql> flush privileges;
```

测试是否能够成功登录

```
mysql -ugit -pgitpwd -D gitlabhq_production
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 5.5.40-log Source distribution

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> \q
```

七, 配置redis

```
chkconfig redis on

cp /etc/redis.conf /etc/redis.conf.orig

#修改监听端口
sed 's/^port .*/port 0/' /etc/redis.conf.orig |tee /etc/redis.conf

#增加内容
echo 'unixsocket /var/run/redis/redis.sock' |tee -a /etc/redis.conf

echo -e 'unixsocketperm 0770' |tee -a /etc/redis.conf
```

创建目录改权限

```
mkdir /var/run/redis

chown redis:redis /var/run/redis

chmod 755 /var/run/redis
```

启动服务

```
service redis restart
```

附加git到redis组

```
usermod -aG redis git
```

八, 安装gitlab

```
cd /home/git

#下载源码
sudo -u git -H git clone https://gitlab.com/larryli/gitlab.git -b 8-0-zh gitlab
```

修改配置

```
cd gitLab/

sudo -u git -H cp config/gitlab.yml.example config/gitlab.yml

sudo -u git -H vim config/gitlab.yml
gitlab:
  ## Web server settings (note: host is the FQDN, do not include http://)
  host: localhost #修改为本机IP地址或者域名如为域名确保域名映射本机地址需修改/etc/hosts文件。
  port: 88
  https: false
```

修改目录权限

```
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
```

新建卫星目录拷贝示例文件

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

编辑配置文件unicorn.rb

```
sudo -u git -H cp config/unicorn.rb.example config/unicorn.rb

#查看系统核心数
nproc
4

#编辑配置
sudo -u git -H vim config/unicorn.rb
worker_processes 4
```

拷贝配置文件rack_attack.rb

```
sudo -u git -H cp config/initializers/rack_attack.rb.example config/initializers/rack_a
ttack.rb
```

定义全局的用户和邮箱

```
sudo -u git -H git config --global user.name "GitLab"
sudo -u git -H git config --global user.email "example@example.com"
sudo -u git -H git config --global core.autocrlf input
```

编辑连接redis配置

```
#拷贝配置
sudo -u git -H cp config/resque.yml.example config/resque.yml

#连接redis配置, 默认配置, 未修改
sudo -u git -H vim config/resque.yml
```

九, 配置gitlab数据库文件

```
sudo -u git cp config/database.yml.mysql config/database.yml

#编辑配置文件
sudo -u git -H vim config/database.yml
production:
  adapter: mysql2
  encoding: utf8
  collation: utf8_general_ci
  reconnect: false
  database: gitlabhq_production
  pool: 10
  username: git
  password: "gitpwd" #修改此处
  host: localhost
  socket: /tmp/mysql.sock

#修改文件权限, 只有git用户可读
sudo -u git -H chmod o-rwx config/database.yml
```

十, 安装gem

```
cd /home/git/gitlab

sudo gem install charlock_holmes --version '0.6.9'

修改 Gemfile 首行内容 source "https://rubygems.org" 修改为 source "https://ruby.taobao.
org"

注意:
gem sources -l
*** CURRENT SOURCES ***
https://ruby.taobao.org
# 请确保只有 ruby.taobao.org

sudo -u git -H bundle install --deployment --without development test postgres sqlite
```

十一, 安装gitlab-shell

官网上给的gitlab-shell版本是2.1.0, 后面会有问题(本地到远程不能连接), 网上查找问题是版本问题, 需要gitlab-shell的版本是2.0.1

```

sudo -u git -H git clone https://github.com/gitlabhq/gitlab-shell.git
sudo -u git -H git checkout v2.0.1
sudo -u git -H cp config.yml.example config.yml

#编辑配置
sudo -u git -H vim /home/git/gitlab-gitlab/shell/config.yml
---
user: git
gitlab_url: https://localhost/
http_settings:
  self_signed_cert: true
repos_path: "/home/git/repositories/"
auth_file: "/home/git/.ssh/authorized_keys"
redis:
  bin: "/usr/bin/redis-cli"
  namespace: resque:gitlab
  socket: "/var/run/redis/redis.sock"
log_level: INFO
audit_usernames: false
#安装
./bin/install

```

十二，初始化数据库

```

sudo -u git -H bundle exec rake gitlab:setup RAILS_ENV=production

```

出现错误: Bundler::GemRequireError: There was an error while trying to load the gem 'coffee-rails'.

解决方式: yum install nodejs

#可以设置管理员密码(此步骤可省略。。。)

```

sudo -u git -H bundle exec rake gitlab:setup RAILS_ENV=production GITLAB_ROOT_PASSWORD=newpassword

```

十三，下载服务脚本

```

wget -O /etc/init.d/gitlab https://gitlab.com/gitlab-org/gitlab-recipes/raw/master/init/sysvinit/centos/gitlab-unicorn

chmod +x /etc/init.d/gitlab

chkconfig --add gitlab

chkconfig gitlab on

#设置logrotate
cp lib/support/logrotate/gitlab /etc/logrotate.d/gitlab

#检测应用状态
sudo -u git -H bundle exec rake gitlab:env:info RAILS_ENV=production

sudo -u git -H bundle exec rake assets:precompile RAILS_ENV=production

#启动服务
service gitlab start
sudo -u git -H bundle exec rails s -e production

```

Ok,你现在可以访问GitLab了，默认端口是 @3000@， 访问 http://你的域名或IP:3000
 第一访问会比较慢，因为要编译很多js和css。
 !!! 注意:
 3000端口可能受防火墙影响，关闭防火墙 `chkconfig iptables off` (重启后永久性生效) `service iptables stop` (及时生效，重启后失效)，或修改/etc/sysconfig/iptables文件
 添加 `-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 3000 -j ACCEPT`

十四，配置web服务

```
#本人使用的nginx
yum -y install nginx

chkconfig nginx on

wget -O /usr/local/nginx/conf/vhost/gitlab.conf https://gitlab.com/gitlab-org/gitlab-ce/raw/master/lib/support/nginx/gitlab-ssl
```

添加nginx用户到git组

```
usermod -a -G git www

chmod g+rx /home/git/
```

创建SSL证书:

由于此证书没有通过验证,所以在git客户端需要关闭证书验证功能

```
mkdir -p /etc/nginx/ssl/
cd /etc/nginx/ssl/
sudo openssl req -newkey rsa:2048 -x509 -nodes -days 3560 -out gitlab.crt -keyout gitlab.key
sudo chmod o-r gitlab.key
```

修改配置,修改server_name

```
vim /usr/local/nginx/conf/vhost/gitlab.conf

upstream gitlab-workhorse {
    server unix:/home/git/gitlab/tmp/sockets/gitlab-workhorse.socket fail_timeout=0;
}

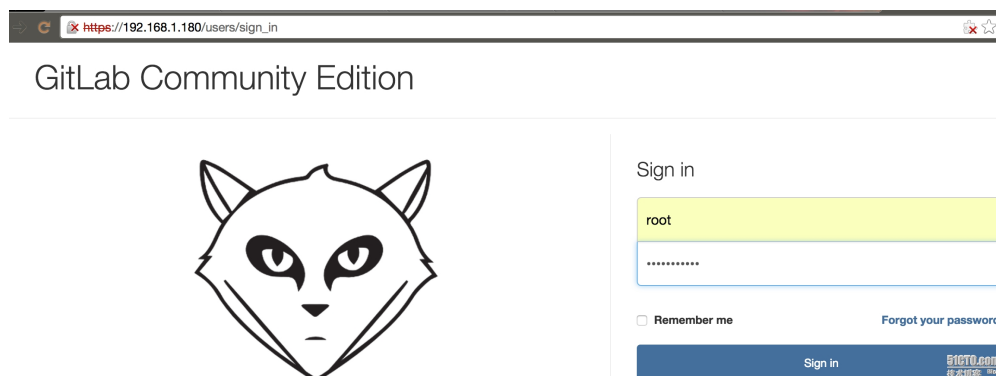
server_name YOUR_SERVER_FQDN; #修改YOUR_SERVER_FQDN为实际域名

location ~ ^/(404|422|500|502)\.html$ {
    root /home/git/gitlab/public;
    internal;
}
```

启动服务

```
service nginx start

#访问登录
#用户名: root
#密码: 5iveL!fe
```



本文出自 “charlie_cen” 博客, 请务必保留此出处<http://charlie928.blog.51cto.com/3741218/1613257>