

# Gitlab 安装部署

GitLab，是一个使用 Ruby on Rails 开发的开源应用程序，与 Github 类似，能够浏览源代码，管理缺陷和注释，非常适合在团队内部使用。

## 安装步骤

- 升级系统并关闭 selinux 和 iptables
- 安装 Ruby
- 创建项目运行用户（创建 git 账号，方便权限管理）
- GitLab Shell
- 数据库（可以支持 mysql 和 PostgreSQL，这里使用 mysql）
- GitLab
- Web 服务器（可支持 nginx 和 apache，这里使用 nginx）

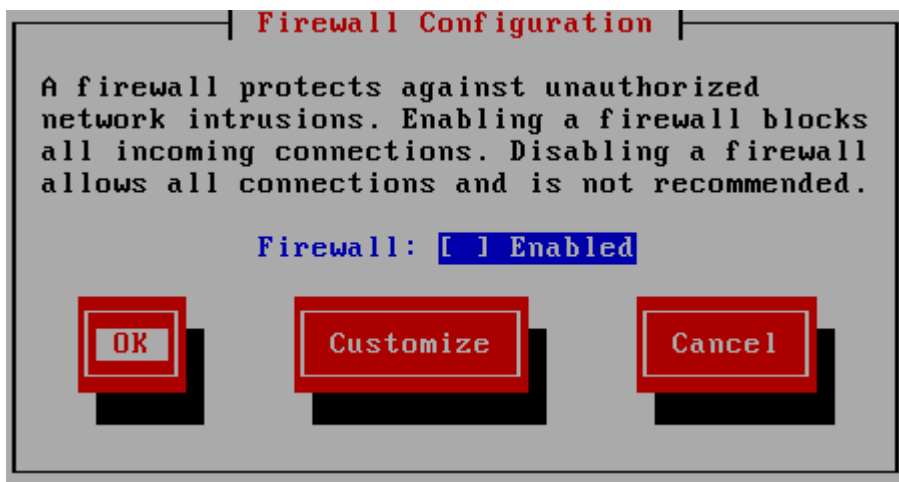
## 一、基础操作

### 1、升级系统

```
yum -y update
```

### 2、关闭 selinux 和 iptables

```
# vi /etc/selinux/config # 设置 SELINUX=disabled  
# setup #将 Firewall 选项取消
```



### 3、增加 EPEL 安装源

```
# wget -O /etc/pki/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
```

检验下是否安装成功

```
# rpm -qa gpg*
```

安装 epel-release-6-8.noarch 包

```
# rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

### 4、增加 PUIAS 安装源

创建/etc/yum.repos.d/PUIAS\_6\_computational.repo，并添加如下内容：

```
[PUIAS_6_computational]
name=PUIAS computational Base $releasever - $basearch
mirrorlist=http://puias.math.ias.edu/data/puias/computational/$releasever/$basearch/mirrorlist
#baseurl=http://puias.math.ias.edu/data/puias/computational/$releasever/$basearch
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-puias
```

下载并安装 GPG key

```
# 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-pui
```

检测源

```
# yum repolist
```

## 5、安装 GitLab 的所需依赖包和工具

```
# su -  
  
# yum -y groupinstall 'Development Tools'  
  
# yum -y install vim-enhanced readline readline-devel ncurses-devel gdbm  
-devel glibc-devel tcl-devel openssl-devel curl-devel expat-devel db4-de  
vel byacc sqlite-devel gcc-c++ libyaml libyaml-devel libffi libffi-devel  
libxml2 libxml2-devel libxslt libxslt-devel libicu libicu-devel system-  
config-firewall-tui python-devel redis sudo wget crontabs logwatch logro  
tate perl-Time-HiRes git
```

## 6、设置 redis 开机启动

```
# chkconfig redis on  
  
# service redis start
```

## 二、安装 Ruby

```
# mkdir /tmp/ruby && cd /tmp/ruby  
  
# curl --progress ftp://ftp.ruby-lang.org/pub/ruby/2.0/ruby-2.0.0-p353.t  
ar.gz | tar xz  
  
# cd ruby-2.0.0-p353  
  
# ./configure --prefix=/usr/local/  
  
# make && make install
```

## 安装 bundle

```
# gem install bundler --no-ri --no-rdoc
```

### 三、创建 git 系统用户

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

### 四、配置 GitLab shell

克隆 gitlab shell

```
# su - git  
$ git clone https://github.com/gitlabhq/gitlab-shell.git  
$ cd gitlab-shell  
$ git checkout v1.8.0 #切换版本  
$ cp config.yml.example config.yml  
$ vi config.yml
```

gitlab\_url 修改成 gitlab 的访问域名。形如：<http://test.gitlab.com/>

如果 gitlab 是使用 https 访问，则需将 http 替换成 https，配置文件中的 self\_signed\_cert 要修改成 true，否则会出错。

安装

```
$ ./bin/install
```

### 五、安装 Mysql 数据库并设置开机启动

```
$ su -  
$ yum install -y mysql-server mysql-devel  
$ chkconfig mysqld on  
$ service mysqld start
```

设置 mysql root 账号的密码

```
/usr/bin/mysql_secure_installation
```

创建新用户和数据库给 gitlab 使用

```
mysql> CREATE USER 'gitlab'@'localhost' IDENTIFIED BY 'gitlab 账号的密码';  
  
# 创建 gitlab 使用的数据库  
  
mysql> CREATE DATABASE IF NOT EXISTS `gitlabhq_production` DEFAULT CHARACTER SET `utf8` COLLATE `utf8_unicode_ci`;  
  
# 给予 gitlab 用户权限  
  
mysql> GRANT SELECT, LOCK TABLES, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON `gitlabhq_production`.* TO 'gitlab'@'localhost';
```

## 六、安装 GitLab

```
$ su - git
```

克隆 GitLab 并切换分支到 6-3-stable

```
$ git clone https://github.com/gitlabhq/gitlabhq.git gitlab  
$ cd /home/git/gitlab  
$ git checkout 6-3-stable
```

配置

```
# 复制配置文件  
$ cp config/gitlab.yml.example config/gitlab.yml  
  
# 修改配置文件中的访问域名  
(your_domain_name 为项目的访问域名)  
$ sed -i 's|localhost|your_domain_name|g' config/gitlab.yml
```

# 设定 log 和 tmp 目录所有者和权限

```
$ chown -R git log/
```

```
$ chown -R git tmp/
```

```
$ chmod -R u+rwX log/
```

```
$ chmod -R u+rwX tmp/
```

# 创建 gitlab-satellites 目录

```
$ mkdir /home/git/gitlab-satellites
```

# 创建 tmp/pids/和 tmp/sockets/ 目录，确保 gitlab 有相应的权限

```
$ mkdir tmp/pids/
```

```
$ mkdir tmp/sockets/
```

```
$ chmod -R u+rwX tmp/pids/
```

```
$ chmod -R u+rwX tmp/sockets/
```

# 创建 public/uploads 目录

```
$ mkdir public/uploads
```

```
$ chmod -R u+rwX public/uploads
```

# 复制 unicorn 配置

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

# 编辑 unicorn 配置

（笔者这里采用默认配置）

```
$ vim config/unicorn.rb
```

# 配置 git 的用户和邮件

```
$ git config --global user.name "GitLab"
```

```
$ git config --global user.email "gitlab@your_domain_name"
$ git config --global core.autocrlf input
```

配置数据库访问文件

```
$ cp config/database.yml.mysql config/database.yml
```

编辑 config/database.yml , 设置其中连接数据库的账号密码

```
#
# PRODUCTION
#
production:
  adapter: mysql2
  encoding: utf8
  reconnect: false
  database: gitlabhq_production
  pool: 10
  username: gitlab
  password: "gitlab"
  # host: localhost
  # socket: /tmp/mysql.sock
```

修改其中 username 和 password 就可以了

确保该文件只有 git 账号有权限读取

```
$ chmod o-rwx config/database.yml
```

安装 Gems

**由于国外 gem 源访问非常慢，建议更换国内 gem 源。**

```
# vi Gemfile
```

编辑 Gemfile 文件，将 gem 源更换为 [ruby.taobao.org](http://ruby.taobao.org)

```
#source "https://rubygems.org"
```

```
source http://ruby.taobao.org
```

修改 Gemfile 文件，将 gem "modernizr", "2.6.2"修改为 gem "modernizr-rails", "2.7.1"。

修改 Gemfile.lock 文件，将 modernizr (2.6.2)修改为 modernizr-rails (2.7.1)。

否则无法正确安装。

```
# gem install charlock_holmes --version '0.6.9.4'
```

安装 mysql 包

```
$ cd /home/git/gitlab/
```

```
$ bundle install --deployment --without development test postgres puma a  
ws
```

初始化数据

```
$ cd /home/git/gitlab
```

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

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



这步完成后，会生一个默认的管理员账号：

```
admin@local.host  
5iveL!fe
```

安装启动脚本

```
$ su -  
  
$ wget -O /etc/init.d/gitlab https://raw.githubusercontent.com/gitlabhq/gitlab-recipes/master/init/sysvinit/centos/gitlab-unicorn  
  
$ chmod +x /etc/init.d/gitlab  
  
$ chkconfig --add gitlab
```

开机时启动

```
$ chkconfig gitlab on
```

检测应用程序状态

```
$ su - git  
  
$ cd gitlab/  
  
$ bundle exec rake gitlab:env:info RAILS_ENV=production  
  
$ exit
```

可以查看到系统、Ruby、GitLab 和 GitLab Shell 的版本和其他信息。

启动 GitLab 实例

```
$ service gitlab start
```

查看应用更加详细的信息

```
$ su - git
$ cd gitlab/
$ bundle exec rake gitlab:check RAILS_ENV=production
```

## 七、安装 web 服务器 nginx

```
$ su -
$ yum -y install nginx
$ chkconfig nginx on
$ mkdir /etc/nginx/sites-available
$ mkdir /etc/nginx/sites-enabled
$ wget -O /etc/nginx/sites-available/gitlab https://raw.githubusercontent.com/gitlabhq/gitlab-recipes/master/web-server/nginx/gitlab-ssl
$ ln -sf /etc/nginx/sites-available/gitlab /etc/nginx/sites-enabled/gitlab
```

编辑/etc/nginx/nginx.conf , 将 include /etc/nginx/conf.d/\*.conf; 替换成 include /etc/nginx/sites-enabled/\*; , 就是修改额外加载的配置文件目录。

编辑/etc/nginx/sites-available/gitlab , 将配置中 server\_name 替换成实际访问的域名。

将 nginx 加入 git 用户组

```
$ usermod -a -G git nginx
$ chmod g+rx /home/git/
```

添加 ssl 证书或者自己生成一个

```
$ cd /etc/nginx
$ openssl req -new -x509 -nodes -days 3560 -out gitlab.crt -keyout gitlab.key
```

启动 nginx

```
$ service nginx start
```

登录 Gitlab 如下图所示:

# GitLab

GitLab is open source software to collaborate on code.  
[Sign in](#) or [browse for public projects](#).

Sign in

☐ Remember me

Sign in

[Forgot your password?](#)

Did not receive confirmation email? [Send again](#)

## **rpm 安装 Gitlab**

详情请查看官网: <https://www.gitlab.com/downloads/>

启动 gitlab

```
# /opt/gitlab/bin/gitlab-ctl start  
ok: run: nginx: (pid 2505) 0s  
ok: run: postgresql: (pid 2509) 1s  
ok: run: redis: (pid 2518) 0s  
ok: run: sidekiq: (pid 2523) 1s  
ok: run: unicorn: (pid 2528) 0s
```