

基本介绍

GitLab是一个自托管的Git项目仓库，可以自己搭建个人代码管理的仓库，功能与github类似。

安装

下载 **gitlab** 下载地址: <https://about.gitlab.com/downloads/>

安装依赖的包

1	<code>sudo yum install curl-devel</code>
2	<code>sudo yum install expat-devel</code>
3	<code>sudo yum install gettext-devel</code>
4	<code>sudo yum install openssl-devel</code>
5	<code>sudo yum install zlib-devel</code>
6	<code>sudo yum install perl-devel</code>
7	<code>sudo yum install curl</code>
8	<code>sudo yum install openssh-server</code>
9	<code>sudo yum install openssh-clients</code>
10	<code>sudo yum install postfix</code>
11	<code>sudo yum install cronie</code>

Ubuntu系统使用apt-get方式安装依赖包。

使用**gitlab**官网的脚本安装

1	<code>curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh sudo</code>
---	--

或者使用**gitlab**的**yum**安装**gitlab**

1	<code>sudo yum install gitlab-ce</code>
---	---

安装完毕后，使用**Web**登录



GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in

Register

Username or email

Password

☐ Remember me

[Forgot your password?](#)

Sign in

Didn't receive a confirmation email? [Request a new one.](#)

进入**gitlab**的管理页面，进行常用的分组，工程，用户等功能点的维护。

Admin Area

Search

Overview

Monitoring

Messages

System Hooks

Applications

Abuse Reports 0

Overview

Projects

Users

Groups

Builds

Runners

Statistics	Features	Components
Forks	0 Sign up	● GitLab
Issues	0 LDAP	⏻ GitLab Shell
Merge Requests	0 Gravatar	● GitLab Workhorse
Notes	0 OmniAuth	⏻ GitLab API
Snippets	0 Reply by email	⏻ Git
SSH Keys	8 Container Registry	⏻ Ruby
Milestones	0	Rails
Active Users	19	PostgreSQL

安装完**gitlab**后的运维操作

初次配置服务

1 | sudo gitlabctl reconfigure

启动服务

1 | sudo gitlabctl start

停止服务

```
1 | sudo gitlab-ctl stop
```

重启服务

```
1 | sudo gitlab-ctl restart
```

备份仓库

先修改 `/etc/gitlab/gitlab.rb`

配置要备份的路径，然后重新执行命令 `sudo gitlab-ctl reconfigure`，做重新配置更新。

```
## For setting up backups
## see https://gitlab.com/gitlab-org/omnibus-gitlab/blob/master
gitlab_rails['manage_backup_path'] = true
gitlab_rails['backup_path'] = "/data/gitlab/backups"
gitlab_rails['backup_archive_permissions'] = 0644 # See: http
gitlab_rails['backup_pg_schema'] = 'public'
```

执行如下命令，创建备份

```
1 | sudo gitlab-rake gitlab:backup:create
```

创建备份时会显示如下信息（会依次备份gitlab上的分组和所有的工程）。

```
Dumping database ...
Dumping PostgreSQL database gitlabhq_production ... [DONE]
done
Dumping repositories ...
* 依次备份git的所有分组和工程 [DONE]
```

备份文件格式：Unix时间戳_gitlab_backup.tar，例如下面做的一个备份文件举例。

```
sudo ls -lh backups/
-rw-r--r-- 1 root root 5M 12月 31 23:38 1483198680_gitlab_backup.tar
```

备份的仓库恢复

```
1 | gitlab-rake gitlab:backup:restore BACKUP=1483198680
```

恢复时，选择的版本就是备份是产生的Unix时间戳。

注：恢复时，先停掉服务，以免双写导致数据错误。

检查服务的日志信息

1	# 检查redis的日志
2	sudo gitlab-ctl tail redis
3	
4	# 检查postgresql的日志
5	sudo gitlab-ctl tail postgresql
6	
7	# 检查gitlab-workhorse的日志
8	sudo gitlab-ctl tail gitlab-workhorse
9	
10	# 检查logrotate的日志
11	sudo gitlab-ctl tail logrotate
12	
13	# 检查nginx的日志
14	sudo gitlab-ctl tail nginx
15	
16	# 检查sidekiq的日志
17	sudo gitlab-ctl tail sidekiq
18	
19	# 检查unicorn的日志
20	sudo gitlab-ctl tail unicorn

检查服务状态

1	sudo gitlab-ctl status
---	------------------------

一般服务状态显示信息

显示格式：

状态： 进程名称：（进程ID）运行时间（秒）；进程的日志服务进程和运行时间

1	run: gitlab-workhorse: (pid 11892) 281s; run: log: (pid 8630) 4742472s
2	run: logrotate: (pid 11904) 280s; run: log: (pid 8631) 4742472s
3	run: nginx: (pid 11911) 280s; run: log: (pid 8796) 4742455s
4	run: postgresql: (pid 12866) 18s; run: log: (pid 8627) 4742472s
5	run: redis: (pid 11989) 249s; run: log: (pid 8638) 4742472s
6	run: sidekiq: (pid 12850) 20s; run: log: (pid 8634) 4742472s
7	run: unicorn: (pid 12022) 247s; run: log: (pid 8629) 4742472s

状态	说明
run	运行状态
down	服务停止

常见的问题

1. 页面显示**500, Whoops, something went wrong on our end.**

1	500
2	
3	Whoops, something went wrong on our end.
4	
5	Try refreshing the page, or going back and attempting the action again.
6	
7	Please contact your GitLab administrator if this <u>problem</u> persists.

如何检查和定位问题？

使用命令检查所有服务的状态

1	<code>sudo gitlab-ctl status</code>
---	-------------------------------------

检查服务状态如下

1	run: gitlab-workhorse: (pid 11892) 91s; run: log: (pid 8630) 4742282s
2	run: logrotate: (pid 11904) 90s; run: log: (pid 8631) 4742282s
3	run: nginx: (pid 11911) 90s; run: log: (pid 8796) 4742265s
4	down: postgresql: 1s, normally up, want up; run: log: (pid 8627) 4742282s
5	run: redis: (pid 11989) 59s; run: log: (pid 8638) 4742282s
6	run: sidekiq: (pid 12201) 2s; run: log: (pid 8634) 4742282s
7	run: unicorn: (pid 12022) 57s; run: log: (pid 8629) 4742282s

定位问题

从服务状态信息中显示数据库postgresql的状态是down，即服务停止。

检查数据库postgresql的运行日志，检查出现什么错误？

1	<code>\$ sudo gitlab-ctl tail postgresql</code>
2	<code>==> /var/log/gitlab/postgresql/state <==</code>
3	
4	<code>==> /var/log/gitlab/postgresql/current <==</code>
5	<code>2016-12-24_01:39:39.00188 FATAL: data directory "/var/opt/gitlab/postgresql/data" has group or</code>
6	<code>2016-12-24_01:39:39.00190 DETAIL: Permissions should be u=rwx (0700).</code>

```
7 | 2016-12-24_01:39:40.00698 FATAL: data directory "/var/opt/gitlab/postgresql/data" has group or
8 | 2016-12-24_01:39:40.00700 DETAIL: Permissions should be u=rwx (0700).
```

日志显示，数据库的访问权限应该是只有用户本身有读写执行的权限，用户组和其他用户不能有权限。

修改数据库数据的权限后，检查服务运行正常。

了解了问题的定位和解决方式，其他问题也很容易在日志中发现和解决，问题可能是磁盘空间少，用户权限错误或者其他原因。

2. gitlab管理员密码忘记，怎么重置密码

Gitlab 修改root用户密码

使用**rails**工具打开终端

```
1 | sudo gitlab-rails console production
```

查询用户的**email**，用户名，密码等信息，**id:1** 表示**root**账号

```
1 | user = User.where(id: 1).first
```

重新设置密码

```
1 | user.password = '新密码'
2 | user.password_confirmation = '新密码'
```

保存密码

```
1 | user.save!
```

完整的操作**ruby**脚本

```
1 | user = User.where(id: 1).first
2 | user.password = '新密码'
3 | user.password_confirmation = '新密码'
4 | user.save!
```

然后使用重置过的密码重新登录。

来源: <https://www.cnblogs.com/voipman/p/gitlab.html>
