前言

GitLab：GitLab 是一个用于仓库管理系统的开源项目，使用Git作为代码管理工具，并在此基础上搭建起来的web服务。
功能：Gitlab 是一个提供代码托管、提交审核和问题跟踪的代码管理平台。对于软件工程质量管理非常重要。
版本：GitLab 分为社区版（CE） 和企业版（EE）。
配置：建议CPU2核，内存2G以上。

安装

一、gitlab安装

1、安装gitlab yum库

1.安装最新版gitlab-ee（企业版）

- curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.rpm.sh | sudo bash

2.安装最新版gilab-ce（社区版）

- curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | sudo bash

2、安装gitlab

其中ip为gitlab的访问地址。

1.企业最新版：

- EXTERNAL_URL="http://ip" yum install -y gitlab-ee

2.社区最新版：

- EXTERNAL_URL="http://ip" yum install -y gitlab-ce

3.安装指定版本：

- EXTERNAL_URL="http://ip" yum install -y gitlab-ee-12.1.9-ee.0.el7.x86_64
- EXTERNAL_URL="http://ip" yum install -y gitlab-ce-12.1.9-ce.0.el7.x86_64

4.安装完成基本指令：

查看运行状态

- gitlab-ctl status

启动

- gitlab-ctl start

停止

- gitlab-ctl stop

重启

- gitlab-ctl restart

查看版本

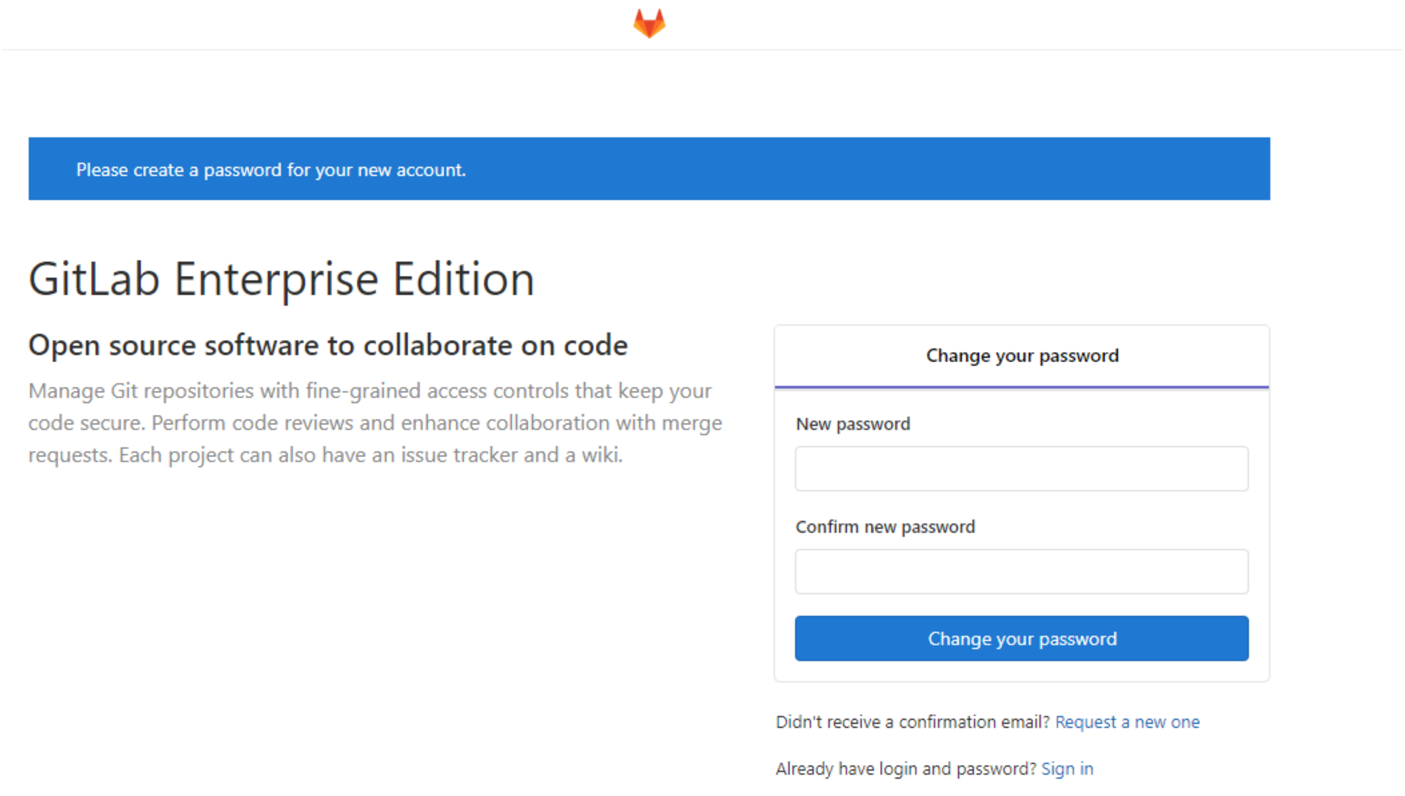
- cat /opt/gitlab/embedded/service/gitlab-rails/VERSION

重载配置

- gitlab-ctl reconfigure

3、访问gitlab

http://ip 浏览器打开



Please create a password for your new account.

GitLab Enterprise 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.

Change your password

New password

Confirm new password

Didn't receive a confirmation email? Request a new one

Already have login and password? Sign in

设置root管理员初始密码。

设置邮箱及https等请参考原文。

二、gitlab备份、恢复

1.gitlab备份

1.设置gitlab备份目录、权限、有限期

```
[root@localhost /]# cat /etc/gitlab/gitlab.rb | grep -v "#"
external_url 'http://192.168.8.127'    //安装设定的访问域名
gitlab_rails['manage_backup_path'] = true
gitlab_rails['backup_path'] = "/data/gitlab/backups"     //备份路径
gitlab_rails['backup_archive_permissions'] = 0644      //备份文件权限
gitlab_rails['backup_keep_time'] = 7776000    //备份文件有效期 30天
```

2.创建备份路径

```
[root@localhost /]# mkdir -vp /data/gitlab/backups
mkdir: 已创建目录 "/data"
mkdir: 已创建目录 "/data/gitlab"
mkdir: 已创建目录 "/data/gitlab/backups"
```

3.重载配置

```
[root@localhost /]# gitlab-ctl reconfigure
Starting Chef Client, version 14.13.11
resolving cookbooks for run list: ["gitlab-ee"]
Synchronizing Cookbooks:
  - gitlab-ee (0.0.1)
  - package (0.1.0)
  - gitlab (0.0.1)
  - consul (0.1.0)
  - runit (4.3.0)
  - redis (0.1.0)
  - repmgr (0.1.0)
  - postgresql (0.1.0)
  - gitaly (0.1.0)
  - letsencrypt (0.1.0)
  - monitoring (0.1.0)
  - registry (0.1.0)
  - mattermost (0.1.0)
  - nginx (0.1.0)
  - acme (4.0.0)
  - crond (0.1.0)
Installing Cookbook Gems:
Compiling Cookbooks...
Recipe: gitlab::default
……
……
……
     (up to date)
Recipe: <Dynamically Defined Resource>
  * service[repmgrd] action nothing (skipped due to action :nothing)
Recipe: repmgr::repmgrd_disable
  * runit_service[repmgrd] action disable
```

```
    * ruby_block[disable repmgrd] action run (skipped due to only_if)
      (up to date)
Recipe: gitlab-ee::geo-secondary_disable
  * templatesymlink[Removes database_geo.yml symlink] action delete
    * file[/var/opt/gitlab/gitlab-rails/etc/database_geo.yml] action delete (up to date)
    * link[/opt/gitlab/embedded/service/gitlab-rails/config/database_geo.yml] action delete (up to date)
      (up to date)
Recipe: <Dynamically Defined Resource>
  * service[unicorn] action restart
    - restart service service[unicorn]
  * service[sidekiq] action restart
    - restart service service[sidekiq]
Recipe: gitlab::gitlab-rails
  * execute[clear the gitlab-rails cache] action run
    - execute /opt/gitlab/bin/gitlab-rake cache:clear
Running handlers:
Running handlers complete
Chef Client finished, 10/708 resources updated in 29 seconds
```

4.开始备份

```
[root@localhost /]# gitlab-rake gitlab:backup:create
2019-09-16 11:00:23 +0800 -- Dumping database ...
Dumping PostgreSQL database gitlabhq_production ... [DONE]
2019-09-16 11:00:24 +0800 -- done
2019-09-16 11:00:24 +0800 -- Dumping repositories ...
2019-09-16 11:00:24 +0800 -- done
2019-09-16 11:00:24 +0800 -- Dumping uploads ...
2019-09-16 11:00:24 +0800 -- done
2019-09-16 11:00:24 +0800 -- Dumping builds ...
2019-09-16 11:00:24 +0800 -- done
2019-09-16 11:00:24 +0800 -- Dumping artifacts ...
2019-09-16 11:00:24 +0800 -- done
2019-09-16 11:00:24 +0800 -- Dumping pages ...
2019-09-16 11:00:24 +0800 -- done
2019-09-16 11:00:24 +0800 -- Dumping lfs objects ...
2019-09-16 11:00:24 +0800 -- done
2019-09-16 11:00:24 +0800 -- Dumping container registry images ...
2019-09-16 11:00:24 +0800 -- [DISABLED]
Creating backup archive: 1568602824_2019_09_16_12.2.5-ee_gitlab_backup.tar ... done
Uploading backup archive to remote storage  ... skipped
Deleting tmp directories ... done
done
done
done
done
done
done
done
Deleting old backups ... done. (0 removed)
Warning: Your gitlab.rb and gitlab-secrets.json files contain sensitive data
and are not included in this backup. You will need these files to restore a backup.
Please back them up manually.
Backup task is done.
```

红字部分表示 gitlab.rb 和 gitlab-secrets.json 两个文件包含敏感信息。未被备份到备份文件中。需要手动备份。

```
[root@localhost backups]# pwd
/data/gitlab/backups
[root@localhost backups]# ls -l
总用量 132
-rw-r--r-- 1 git git 133120 9月  16 11:00 1568602824_2019_09_16_12.2.5-ee_gitlab_backup.tar
[root@localhost backups]#
```

执行完成后在 /data/gitlab/backups 中生成了备份文件。

2.gitlab恢复备份

1.首先停用gitlab的数据连接部分服务

```
[root@localhost backups]# gitlab-ctl stop unicorn
ok: down: unicorn: 0s, normally up
[root@localhost backups]# gitlab-ctl stop sidekiq
ok: down: sidekiq: 0s, normally up
[root@localhost backups]# gitlab-ctl stop nginx
ok: down: nginx: 0s, normally up
[root@localhost backups]# gitlab-ctl status
run: alertmanager: (pid 30960) 3683s; run: log: (pid 30623) 3735s
run: gitaly: (pid 30846) 3685s; run: log: (pid 30081) 3828s
run: gitlab-monitor: (pid 30843) 3685s; run: log: (pid 30498) 3753s
run: gitlab-workhorse: (pid 30820) 3686s; run: log: (pid 30349) 3784s
run: grafana: (pid 30980) 3682s; run: log: (pid 30750) 3698s
run: logrotate: (pid 37253) 178s; run: log: (pid 30384) 3777s
down: nginx: 4s, normally up; run: log: (pid 30368) 3781s
run: node-exporter: (pid 30830) 3685s; run: log: (pid 30418) 3764s
run: postgres-exporter: (pid 30971) 3683s; run: log: (pid 30650) 3730s
run: postgresql: (pid 30127) 3825s; run: log: (pid 30139) 3821s
run: prometheus: (pid 30943) 3684s; run: log: (pid 30588) 3741s
run: redis: (pid 29960) 3838s; run: log: (pid 29972) 3835s
run: redis-exporter: (pid 30854) 3684s; run: log: (pid 30522) 3747s
down: sidekiq: 13s, normally up; run: log: (pid 30324) 3788s
down: unicorn: 17s, normally up; run: log: (pid 30307) 3792s
```

2.执行恢复

远程复制的备份需要 chmod 777 1568602824_2019_09_16_12.2.5-ee.....   赋予权限

```
[root@localhost backups]# gitlab-rake gitlab:backup:restore BACKUP=1568602824_2019_09_16_12.2.5-ee
Unpacking backup ... done
Before restoring the database, we will remove all existing
```

```
tables to avoid future upgrade problems. Be aware that if you have
custom tables in the GitLab database these tables and all data will be
removed.
..........................................
Do you want to continue (yes/no)? yes
This task will now rebuild the authorized_keys file.
You will lose any data stored in the authorized_keys file.
Do you want to continue (yes/no)? yes

Deleting tmp directories ... done
done
done
done
done
done
done
done
Warning: Your gitlab.rb and gitlab-secrets.json files contain sensitive data
and are not included in this backup. You will need to restore these files manually.
Restore task is done.
```

备份tar包一定要放到备份路径下。恢复是删除原有数据，恢复备份tar包中的数据。

如果是在其他服务器恢复备份，一定要记得将 gitlab.rb 和 gitlab-secrets.json 手动复制到相应路径下。

gitlab.rb路径：/etc/gitlab/gitlab.rb

gitlab-secrets.json路径：/etc/gitlab/gitlab-secrets.json

3.重载配置

- gitlab-ctl reconfigure

4.重启gitlab，check检查

```
[root@localhost backups]# gitlab-ctl restart
ok: run: alertmanager: (pid 30960) 3785s
ok: run: gitaly: (pid 30846) 3787s
ok: run: gitlab-monitor: (pid 30843) 3787s
ok: run: gitlab-workhorse: (pid 30820) 3788s
ok: run: grafana: (pid 30980) 3784s
ok: run: logrotate: (pid 37253) 280s
ok: run: nginx: (pid 37535) 0s
ok: run: node-exporter: (pid 30830) 3787s
ok: run: postgres-exporter: (pid 30971) 3785s
ok: run: postgresql: (pid 30127) 3927s
ok: run: prometheus: (pid 30943) 3786s
ok: run: redis: (pid 29960) 3940s
ok: run: redis-exporter: (pid 30854) 3786s
ok: run: sidekiq: (pid 37547) 1s
ok: run: unicorn: (pid 37553) 0s
[root@localhost backups]# gitlab-rake gitlab:check SANITIZE=true
Checking GitLab subtasks ...

Checking GitLab Shell ...

GitLab Shell: ... GitLab Shell version >= 9.3.0 ? ... OK (9.3.0)
Running /opt/gitlab/embedded/service/gitlab-shell/bin/check
Check GitLab API access: OK
Redis available via internal API: OK

Access to /var/opt/gitlab/.ssh/authorized_keys: OK
gitlab-shell self-check successful

Checking GitLab Shell ... Finished

Checking Gitaly ...

Gitaly: ... default ... OK

Checking Gitaly ... Finished

Checking Sidekiq ...

Sidekiq: ... Running? ... yes
Number of Sidekiq processes ... 1

Checking Sidekiq ... Finished

Checking Incoming Email ...

Incoming Email: ... Reply by email is disabled in config/gitlab.yml

Checking Incoming Email ... Finished

Checking LDAP ...

LDAP: ... LDAP is disabled in config/gitlab.yml

Checking LDAP ... Finished

Checking GitLab App ...

Git configured correctly? ... yes
Database config exists? ... yes
All migrations up? ... yes
Database contains orphaned GroupMembers? ... no
GitLab config exists? ... yes
GitLab config up to date? ... yes
Log directory writable? ... yes
Tmp directory writable? ... yes
```

```
Uploads directory exists? ... yes
Uploads directory has correct permissions? ... yes
Uploads directory tmp has correct permissions? ... skipped (no tmp uploads folder yet)
Init script exists? ... skipped (omnibus-gitlab has no init script)
Init script up-to-date? ... skipped (omnibus-gitlab has no init script)
Projects have namespace: ... can't check, you have no projects
Redis version >= 2.8.0? ... yes
Ruby version >= 2.5.3 ? ... yes (2.6.3)
Git version >= 2.22.0 ? ... yes (2.22.0)
Git user has default SSH configuration? ... yes
Active users: ... 1
Elasticsearch version 5.6 - 6.x? ... skipped (elasticsearch is disabled)

Checking GitLab App ... Finished


Checking GitLab subtasks ... Finished
```

ok，至此恢复完成

三、gitlab版本升级

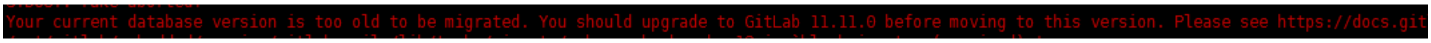
1.停止数据传输服务

```
[root@localhost backups]# gitlab-ctl stop unicorn
ok: down: unicorn: 0s, normally up
[root@localhost backups]# gitlab-ctl stop sidekiq
ok: down: sidekiq: 0s, normally up
[root@localhost backups]# gitlab-ctl stop nginx
ok: down: nginx: 1s, normally up
[root@localhost backups]# gitlab-ctl status
run: gitaly: (pid 73164) 939s; run: log: (pid 72843) 968s
run: gitlab-monitor: (pid 73232) 938s; run: log: (pid 73021) 957s
run: gitlab-workhorse: (pid 75742) 136s; run: log: (pid 72919) 962s
run: logrotate: (pid 72985) 960s; run: log: (pid 72984) 960s
down: nginx: 5s, normally up; run: log: (pid 72960) 961s
run: node-exporter: (pid 73003) 959s; run: log: (pid 73002) 959s
run: postgres-exporter: (pid 73253) 937s; run: log: (pid 73086) 948s
run: postgresql: (pid 72644) 997s; run: log: (pid 72643) 997s
run: prometheus: (pid 73241) 938s; run: log: (pid 73064) 950s
run: redis: (pid 72598) 1003s; run: log: (pid 72597) 1003s
run: redis-exporter: (pid 73048) 951s; run: log: (pid 73047) 951s
down: sidekiq: 7s, normally up; run: log: (pid 72818) 974s
down: unicorn: 10s, normally up; run: log: (pid 72778) 976s
```

2.下载最新的ce/ee rpm包（清华源https://mirrors.tuna.tsinghua.edu.cn/gitlab-ee/yum/el7/ 或者https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7/）

- wget https://mirrors.tuna.tsinghua.edu.cn/gitlab-ee/yum/el7/gitlab-ee-12.2.5-ee.0.el7.x86_64.rpm

gitlab10.0.0版本升级最新版本需要先升级到11.11.0，再升级12.2.5！



```
Your current database version is too old to be migrated. You should upgrade to GitLab 11.11.0 before moving to this version. Please see https://docs.git
```

3.升级ce/ee

- rpm -Uvh gitlab-ee-12.2.5-ee.0.el7.x86_64.rpm

4.重载配置

- gitlab-ctl reconfigure

5.重启gitlab

- gitlab-ctl restart

四、gitlab解决内存消耗问题

```
[root@storage100 backups]# cat /etc/gitlab/gitlab.rb | grep -v "#"
#进程超时时间
unicorn['worker_timeout'] = 60
#进程数
unicorn['worker_processes'] = 10
#进程最小内存
unicorn['worker_memory_limit_min'] = "200 * 1 << 20"
#进程最大内存
unicorn['worker_memory_limit_max'] = "300 * 1 << 20"
#并发数
sidekiq['concurrency'] = 16
#数据库缓存
postgresql['shared_buffers'] = "256MB"
#数据库并发数
postgresql['max_worker_processes'] = 8
```

参考CSDN博主"欧阳鹏",原文链接：https://blog.csdn.net/ouyang_peng/article/details/84066417

附gitlab备份脚本：

```
#!/bin/bash

#Gitlab 备份地址
```

```
LocalBackDir=/home/gitlab/data/backups

#服务机gitlab配置文件地址
#正式环境地址:
ConfigDir=/etc/gitlab/gitlab.rb

#nginx配置文件地址
#正式环境地址: NginxDir=/home/dgd/gitlab/data/nginx/conf

#邮件配置地址
#MailDir=/etc/postfix/main.cf

#Backup server 存储路径
RemoteBackDir=/home/gitlab/data/backups

#远程备份使用用户及端口
RemoteUser=root
RemotePort=22

#备份服务器IP
RemoteIP=192.168.8.200

#以当前时间戳创建备份目录
bakname=$(date -d "today" +"%Y%m%d_%H%M%S")
BakDir=$LocalBackDir/$bakname
mkdir $BakDir

#备份日志文件
LogFile=$LocalBackDir/remote_backup.log

#新建备份日志文件
touch $LogFile

#记录配置文件备份日志
echo "Gitlab configure file auto backup at local server, start at  $(date +"%Y-%m-%d %H:%M:%S")" >>  $LogFile
echo "--------------------------------------------------------------------------" >> $LogFile

#拷贝配置文件
cp $ConfigDir $BakDir >> $LogFile 2>&1
#cp -r $NginxDir $BakDir >> $LogFile 2>&1
#cp $MailDir $BakDir >> $LogFile 2>&1

#记录本地生成gitlab备份日志
echo "Gitlab auto backup at local server, start at  $(date +"%Y-%m-%d %H:%M:%S")" >>  $LogFile
echo "--------------------------------------------------------------------------" >> $LogFile

#执行gitlab本地备份
gitlab-rake gitlab:backup:create >> $LogFile 2>&1

# $?符号显示上一条命令的返回值，如果为0则代表执行成功，其他表示失败
if [ $? -eq 0 ];then
   #追加日志到日志文件
   echo "--------------------------------Success!-------------------------------" >> $LogFile
   echo "Gitlab auto backup at local server, end at $(date +"%Y-%m-%d %H:%M:%S")" >> $LogFile
else
   #追加日志到日志文件
   echo "--------------------------------Failed!----------------------------------" >> $LogFile
   echo "Gitlab auto backup at local server failed at $(date +"%Y-%m-%d %H:%M:%S")" >> $LogFile
fi


#查找本地备份目录修改时间为10分钟以内且后缀为.tar的Gitlab备份文件
Backfile_Send_To_Remote=`find $LocalBackDir -type f  -mmin -10 -name '*.tar'` >> $LogFile 2>&1

#移动生成的备份文件到配置文件备份地址
mv -bfu $Backfile_Send_To_Remote $BakDir


#记录备份日志
echo "$(date +"%Y-%m-%d %H:%M:%S") Gitlab auto backup to remote server." >> $LogFile
echo "--------------------------------------------" >> $LogFile

#打印每次备份的档案名
echo "The files need send to remote server is: $Backfile_Send_To_Remote" >> $LogFile

# 本地传输Gitlab备份档案到远程
scp -r  $BakDir $RemoteUser@$RemoteIP:$RemoteBackDir

# 备份结果追加到备份日志
if [ $? -eq 0 ];then
  echo ""
  echo "$(date +"%Y-%m-%d %H:%M:%S") Gitlab Remote Backup Succeed!" >> $LogFile
else
  echo "$(date +"%Y-%m-%d %H:%M:%S") Gitlab Remote Backup Failed!" >> $LogFile
fi
```