

若-飞

[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

gitlab+gerrit+jenkins持续集成框架

# 1.持续集成之 gitlab+gerrit+jenkins

## 1.1. GitLab

### 1.1.1. 简介

GitLab 是一个使用使用Ruby on Rails搭建的，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

### 1.1.2. 环境搭建

a) Install and configure the necessary dependencies

If you install Postfix to send email please select 'Internet Site' during setup. Instead of using Postfix you can also use Sendmail or [configure a custom SMTP server](#). If you wish to use Exim, please [configure it as an SMTP server](#).

On Centos 6 and 7, the commands below will also open HTTP and SSH access in the system firewall.

#### 公告

昵称：若-飞  
园龄：1年10个月  
粉丝：1  
关注：0  
[+加关注](#)

< 2017年9月 >						
日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

#### 搜索

 找找看 谷歌搜索

```
sudo apt-get install curl openssh-server ca-certificates postfix
```

b) Add the GitLab package server and install the package.

- Apt-get安装

```
Curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.deb.sh|sudo bash
```

```
sudo apt-get install gitlab-ee
```

If you are not comfortable installing the repository through a piped script, you can find the [entire script here](#).

Alternatively you can [select and download the package manually](#) and install using

- deb包安装

下载deb包

<https://packages.gitlab.com/gitlab/gitlab-ee>

安装deb包

```
sudo dpkg -i gitlab-ee_8.1.2-ee.0_amd64.deb
```

c) 1.1.2.3. Configure and start GitLab

```
sudo gitlab-ctl reconfigure
```

d) 1.1.2.4. Browse to the hostname and login

```
http://10.24.6.8:80
```

Username: root

Password: 5iveL!fe

e) 2.1.6. Upload your license(EE版本需要认证)

Go to Admin > License and upload your .gitlab-license file.

Don't have a license yet? [Get a subscription](#).

### 1.1.3. 创建TEST项目

#### 常用链接

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

#### 我的标签

CLUSTER(2)

MYSQL(2)

paas docker 架构(2)

REDIS(2)

SENTINEL(1)

持续集成(1)

敏捷开发(1)

DRBD + HEARTBEAT + MYSQL(1)

gitlab+gerrit+jenkins(1)

KEEPALIVE(1)

[更多](#)

#### 随笔分类

mysql(3)

nginx(1)

PAAS架构(3)

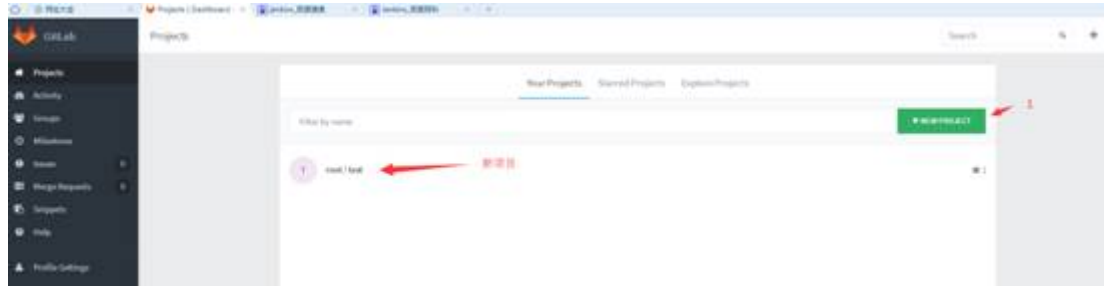
redis(2)

敏捷开发架构(1)

前端开发(1)

职业生涯(1)

## 创建test 项目



### 1.1.4. Gerrit Review支持

前面我们在Gitlab上搭建了一个 **test** 的工程，普通用户是没有办法去 push 的，只能使用 git review 命令提交. 而 git review 命令需要 .gitreview 文件存在于项目目录里。

#### 1.1.4.1安装支持库

需要安装git-review, readline

Apt-get install git-review

Linux : pip install readline

Windows : pip install pyreadline

```
logic_server2@drbd01:~/test$ sudo apt-get install git-review
[sudo] password for logic_server2:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  git-review
1 upgraded, 1 newly installed, 0 to remove and 162 not upgraded.
Need to get 18.1 kB of archives.
After this operation, 120 kB of additional disk space will be used.
Get:1 http://mirrors.163.com/ubuntu/trusty/universe git-review all 1.23-1 [18.1 kB]
debconf: delaying package configuration, since apt-utils is not installed
Fetched 18.1 kB in 0s (87.7 kB/s)
Selecting previously unselected package git-review.
Reading database ... 84046 files and directories currently installed.)
Preparing to unpack .../git-review_1.23-1_all.deb ...
Unpacking git-review (1.23-1) ...
Processing triggers for man-db (2.6.7.1-ubuntu1) ...
Setting up git-review (1.23-1) ...
logic_server2@drbd01:~/test$ ls
```

### 随笔档案

2016年4月 (5)

2016年3月 (3)

2016年1月 (2)

2015年12月 (2)

### 最新评论

1. Re:mysql集群之MYSQL CLUSTER

666的和我的不相上下

--男神之路

2. Re:gitlab+gerrit+jenkins持续集成框架

不错~

--狂奔的菜鸟

3. Re:gitlab+gerrit+jenkins持续集成框架

可真长.....

--wdwwtzy

4. Re:gitlab+gerrit+jenkins持续集成框架

之前搭过一次，没成功，收藏慢慢看

--Herb

### 阅读排行榜

1. gitlab+gerrit+jenkins持续集成框架(5327)

2. ubuntu14.04下nodejs + npm + bower的安装、调试和部署(3475)

3. mysql集群之MYSQL CLUSTER(1980)

4. Nginx文件上传下载实现与文件管理(1924)

5. mysql高可用之LVS + KEEPAFIVE + MYSQL(901)

### 1.1.4.2 用 admin用户给test项目添加.gitreview 文件

a) 拷贝gitlab的test项目

```
gerrit@gerrit:~/cfg$ git init cfg
```

```
gerrit@gerrit:~/cfg$ git init cfg
```

```
gerrit@gerrit:~/cfg/cfg$ git config --global user.name 'root'
```

```
gerrit@gerrit:~/cfg/cfg$ git config --global user.email xxxxxxxx@qq.com
```

```
gerrit@gerrit:~/cfg/cfg$ git clone http://10.24.6.8/root/test.git
```

```
gerrit@gerrit:~/cfg/cfg$ cd test/
```

```
gerrit@gerrit:~/cfg$ git init cfg
Initialized empty Git repository in /home/gerrit/cfg/cfg/.git/
gerrit@gerrit:~/cfg$ cd cfg/
gerrit@gerrit:~/cfg/cfg$ git config --global user.name 'root'
gerrit@gerrit:~/cfg/cfg$ git config --global user.email xxxxxxxx@qq.com
gerrit@gerrit:~/cfg/cfg$ git clone http://10.24.6.8/root/test.git
Cloning into 'test'...
Username for 'http://10.24.6.8': root
Password for 'http://root@10.24.6.8':
remote: Counting objects: 52, done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 52 (delta 16), reused 0 (delta 0)
Unpacking objects: 100% (52/52), done.
Checking connectivity... done.
gerrit@gerrit:~/cfg/cfg$ ls
test
gerrit@gerrit:~/cfg/cfg$ cd test/
gerrit@gerrit:~/cfg/cfg/test$ ls
t1.txt t2.txt t33.txt t3.txt t44.txt t55.txt t66.txt t77.txt t88.txt
gerrit@gerrit:~/cfg/cfg/test$ vim .gitreview
[gerrit]
    host=10.24.6.12
    port=29418
    project=test.git
```

b) 添加gitreview文件

```
gerrit@gerrit:~/cfg/cfg/test$ vim .gitreview
```

#### 评论排行榜

1. gitlab+gerrit+jenkins持续集成框架(3)
2. mysql集群之MYSQL CLUSTER(1)

#### 推荐排行榜

1. gitlab+gerrit+jenkins持续集成框架(2)

文件内容为：

```
[gerrit]
host=10.24.6.12
port=29418
project=test.git
```

gerrit服务的ip信息  
gerrit的端口信息  
gerrit项目名称

### c) 上传gitreview文件

```
gerrit@gerrit:~/cfg/cfg/test$ git add .gitreview
```

```
gerrit@gerrit:~/cfg/cfg/test$ git commit .gitreview -m 'add .gitreview file by admin.'
```

```
gerrit@gerrit:~/cfg/cfg/test$ git push origin master
```

```
gerrit@gerrit:~/cfg/cfg/test$ ls
t1.txt t2.txt t33.txt t3.txt t44.txt t55.txt t66.txt t77.txt t88.txt
gerrit@gerrit:~/cfg/cfg/test$ git add .gitreview
gerrit@gerrit:~/cfg/cfg/test$ git commit .gitreview -m 'add .gitreview file by admin.'
[master eeee4d6] add .gitreview file by admin.
1 file changed, 5 insertions(+)
create mode 100644 .gitreview
gerrit@gerrit:~/cfg/cfg/test$ git push origin master
Username for 'http://10.24.6.8': root
Password for 'http://root@10.24.6.8':
Counting objects: 6. done.
Compressing objects: 100% (3/3). done.
Writing objects: 100% (3/3), 333 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To http://10.24.6.8/root/test.git
1f2222d..eeee4d6 master -> master
gerrit@gerrit:~/cfg/cfg/test$ ^C
```

## 1.1.5. 用户创建

以admin账户登录创建账户



New user

Account

Name  1  
\* required

Username  2  
\* required

Email  3  
\* required

Password

Password Reset link will be generated and sent to the user.  
User will be forced to set the password on first sign in.

Access

Projects limit

Can create group ☐

Admin ☐

Profile

Avatar  未选择文件

Skype

Linkedin

Twitter

Website

4

### 1.1.6 SSH设置

- Su 进入linux对应账户
- 创建sshkey

```
$ sudo -u gerrit -H ls /home/gerrit/.ssh
```

```
$ sudo -u gerrit -H ssh-keygen -C xxxxxx@qq.com
```

邮箱为gitlab的账户注册邮箱

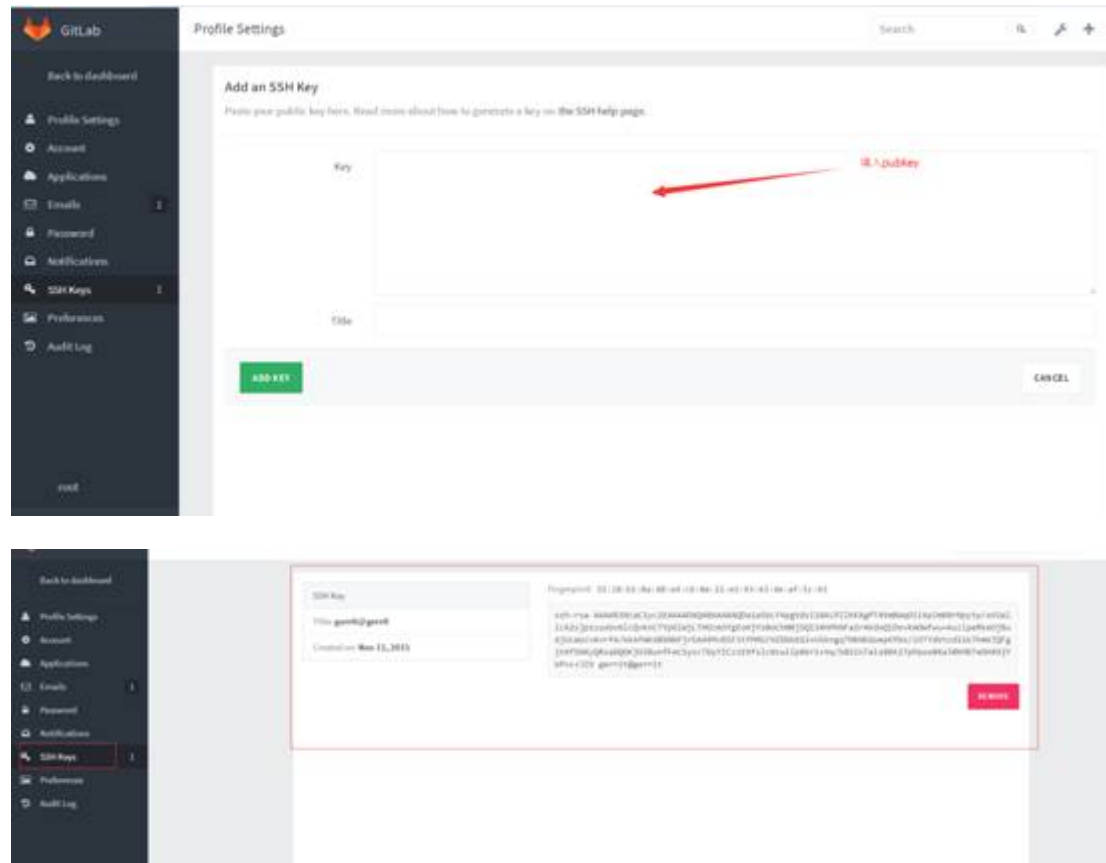
#### c) 获取ssh pub key

```
$ sudo -u gerrit -H cat /home/gerrit/.ssh/id_rsa.pub
```

```
zjc@zjc:~$ sudo -u gerrit -H ls /home/gerrit/.ssh
ls: cannot access /home/gerrit/.ssh: No such file or directory
zjc@zjc:~$ sudo -u gerrit -H ssh-keygen -C gerrit@vzjc.com
Generating public/private rsa key pair.
Enter file in which to save the key (/home/gerrit/.ssh/id_rsa):
Created directory '/home/gerrit/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gerrit/.ssh/id_rsa.
Your public key has been saved in /home/gerrit/.ssh/id_rsa.pub.
The key fingerprint is:
0b:15:90:5b:c0:2a:96:b1:4f:4d:ee:f3:07:b2:02:3d gerrit@vzjc.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .O+      |
|     + + ..    |
|    + = O.     |
|   = O +.      |
|  . = .. S     |
|   . E +...    |
|    . . =..    |
|     . . . .   |
|      . .      |
+-----+
zjc@zjc:~$ sudo -u gerrit -H cat /home/gerrit/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADepKNNcv6bjvs33amKnJsaRPwEeQ3w4JauTWaHXNH
q4JkEMz/vxc9ETZ85EaBSm2Dv1SuhKTbYhBs24bTLJH7Mw32TVcZnM4l4XffT6igAjuXPpnj6jwaxh3f
M+/NwQyDo2tWLxDRBGXltuBMP0p7OpfRM8ms+Eu143sFi4M7prufn62g+PUl0vWuHfKGwD6XesTs3Y4Z
8UuE1CFjJqJ5vjN6v660KT0WgEsiRo//7/8wFEu63QCzVUZT+yBQTMrsbRrnl95b3LqGV0lE16N50b1C
SiQFThuvbQbBHGU8hnCprPcCOGSMW0yCT/E1Hyk7hKKKRrAF0JaGCDPS+QKL gerrit@vzjc.com
zjc@zjc:~$
```

#### d) 登录新创建用户设置SSH Keys

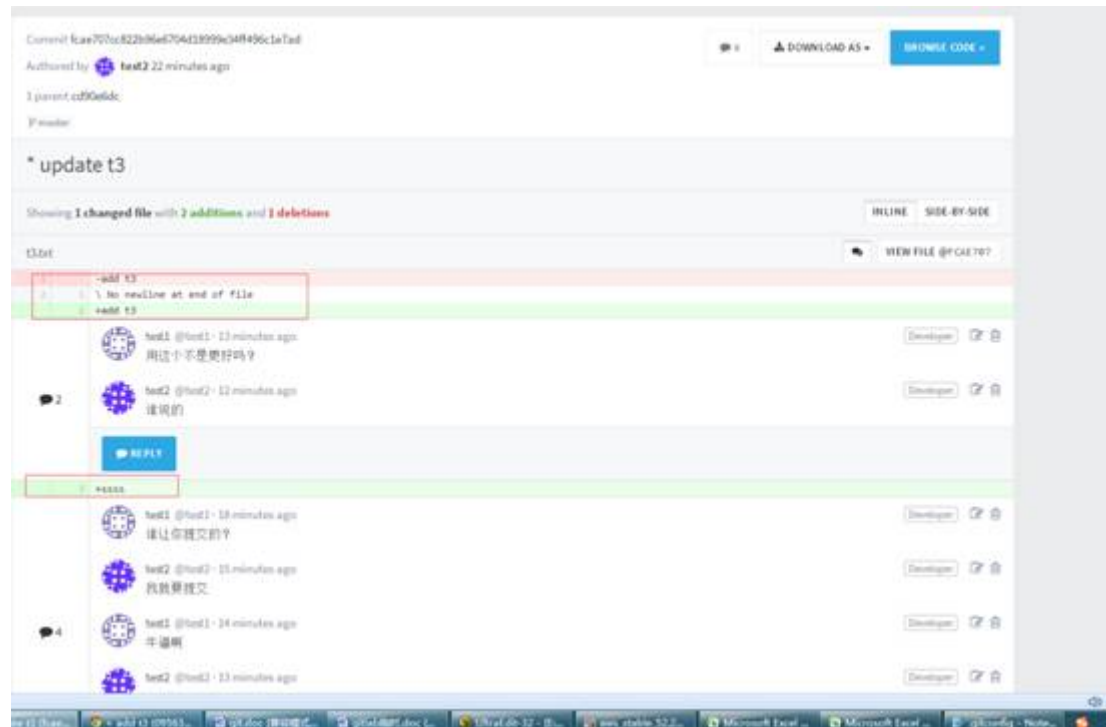




### 1.1.7. Git自带code review功能

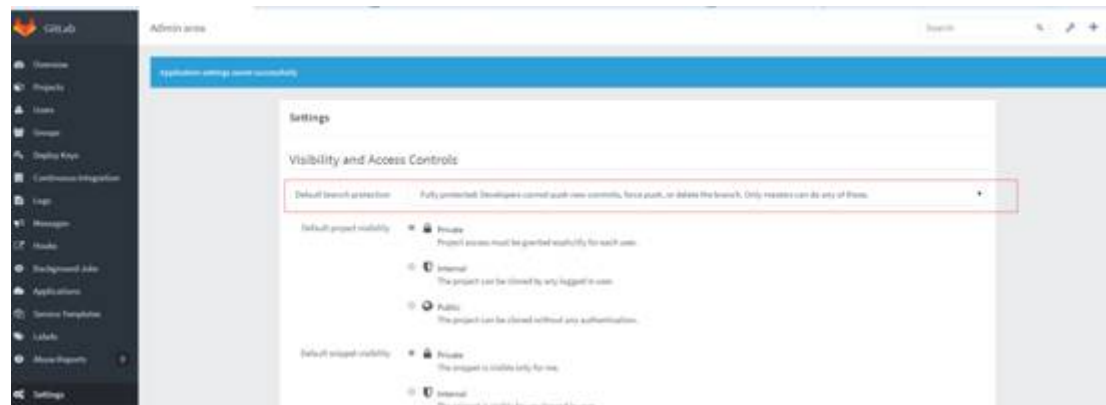
- 强制 Review : 在 Gitlab 上创建的项目, 指定相关用户只有 Reporter 权限, 这样用户没有权限使用 git push 功能, 只能 git review 到 Gerrit 系统上, Jenkins 在监听 Gerrit 上的项目事件会触发构建任务来测试代码, Jenkins 把测试结果通过 ssh gerrit 个这个项目打上 Verified 成功或失败标记, 成功通知其它人员 Review。
- Ø Gitlab 保护 Master 分支 : 在 Gitlab 上创建的项目可以把 Master 分支保护起来, 普通用户可以自己创建分支并提交代码到自己的分支上, 没有权限直接提交到 Master 分支, 用户最后提交申请把自己的分支 Merge 到 Master, 管理员收到 Merge 请求后, Review 后选择是否合并。

## a) 针对某次提交进行评论

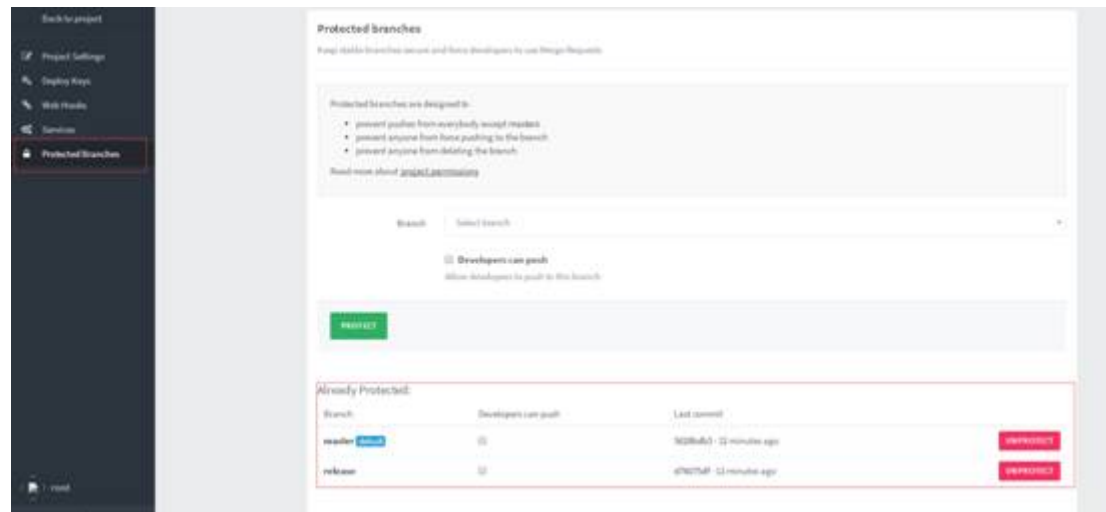


## b) 负责人设置master/release分支受保护，只有负责人可以提交，其他人不能push

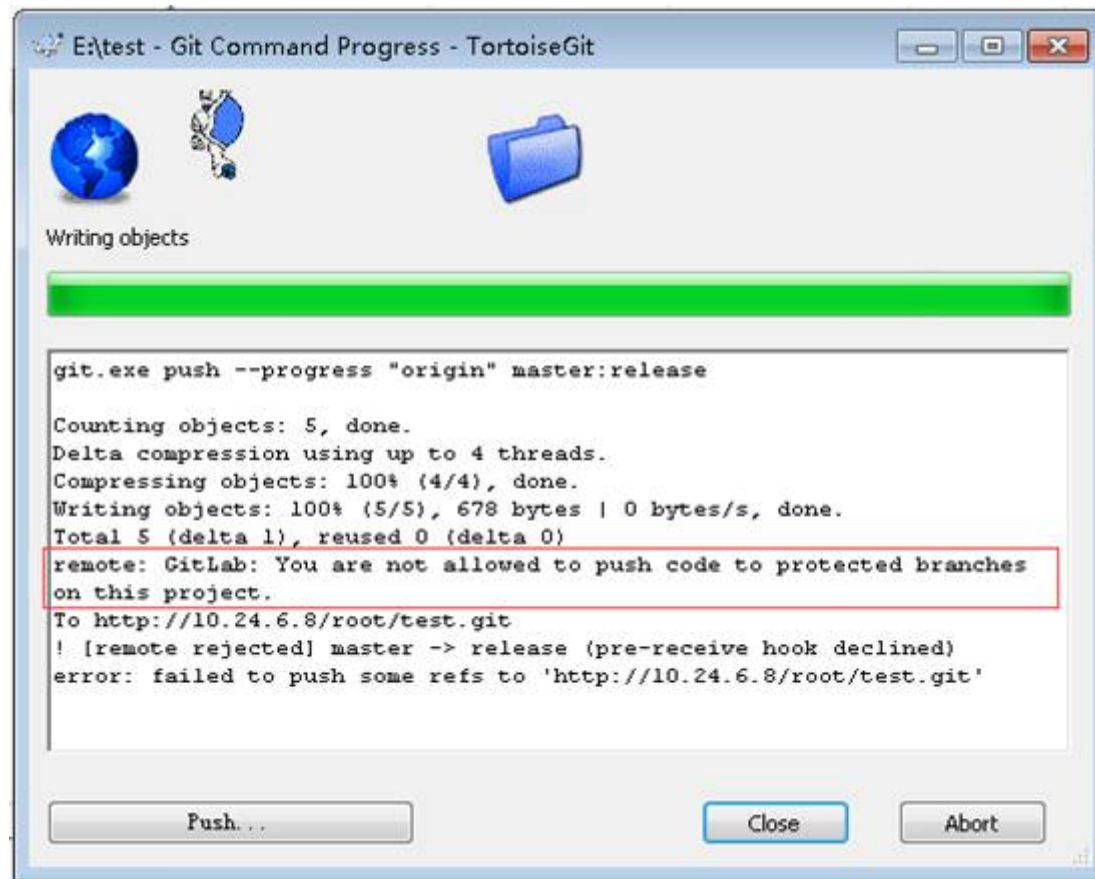
## Branch protected总开关



## 项目分支开关



c) 没被允许提交的用户无法提交



d) GitLab提交合并请求给负责人

同时发送邮件给该负责人，让他知道有人要合并分支

[- Merge Requests Search in this project 🔍 +

Title \* Test2  
Start the title with `[WIP]` or `WIP:` to prevent a Work in Progress merge request from being merged before it's ready.

Description Write Preview [Edit in full screen](#)  
 测试2工作完成。请求合并

Attach with GitLab Planned Markdown Attach files by dragging & dropping or selecting them

Assign to root [ASSIGN TO ME](#)

Milestone No open milestones available

Labels No labels yet. [Create new label](#)

Source branch test2

Target branch master

[Change branches](#)

[SUBMIT NEW MERGE REQUEST](#) [CANCEL](#)

[Open](#) Merge Request #1 · created by test2 · 5 minutes ago [CLOSE](#) [EDIT](#)

**Test2**  
测试2工作完成。请求合并

Request to merge test2 into master [CHECK OUT BRANCH](#) [DOWNLOAD AS](#)

**Ready to be merged automatically**  
Ask someone with write access to this repository to merge this request.

You can also accept this merge request manually using the command line.

[Discussion](#) [Commits](#) [Changes](#)

2 participants root/test2

root @root · about a minute ago [Dismiss](#)  
还有让谁审核？

test2 @test2 · less than a minute ago [Dismiss](#) [QF](#)  
测试已经完成了啊

Write Preview [Edit in full screen](#)

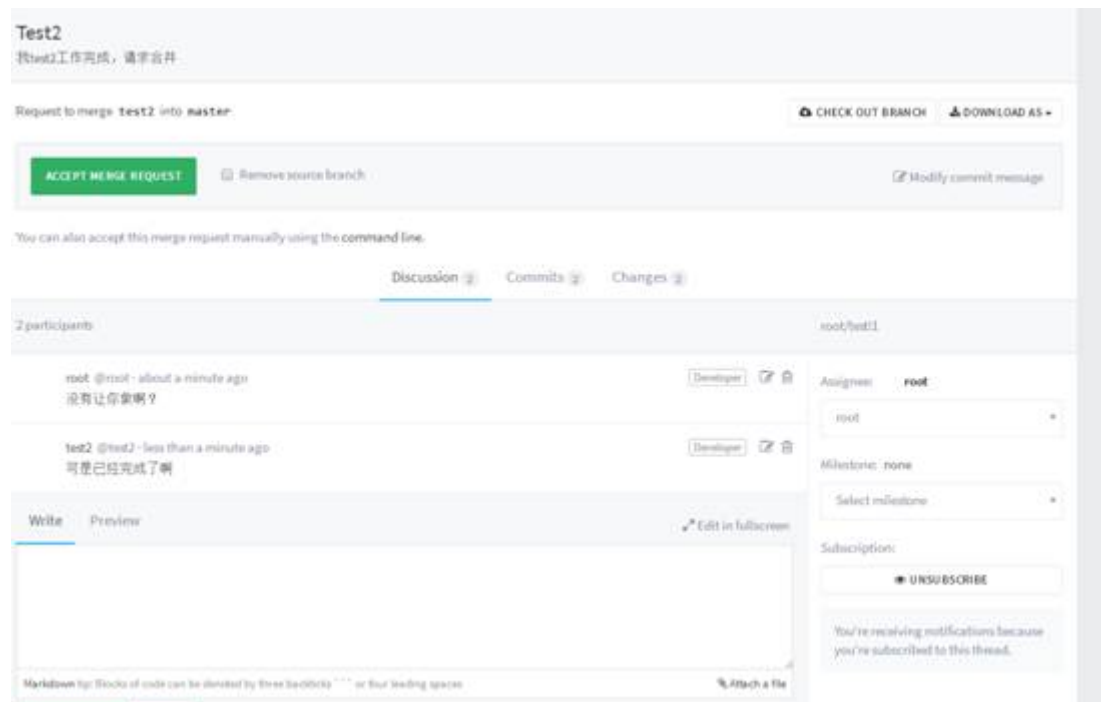
Assigned: root

Milestone: none

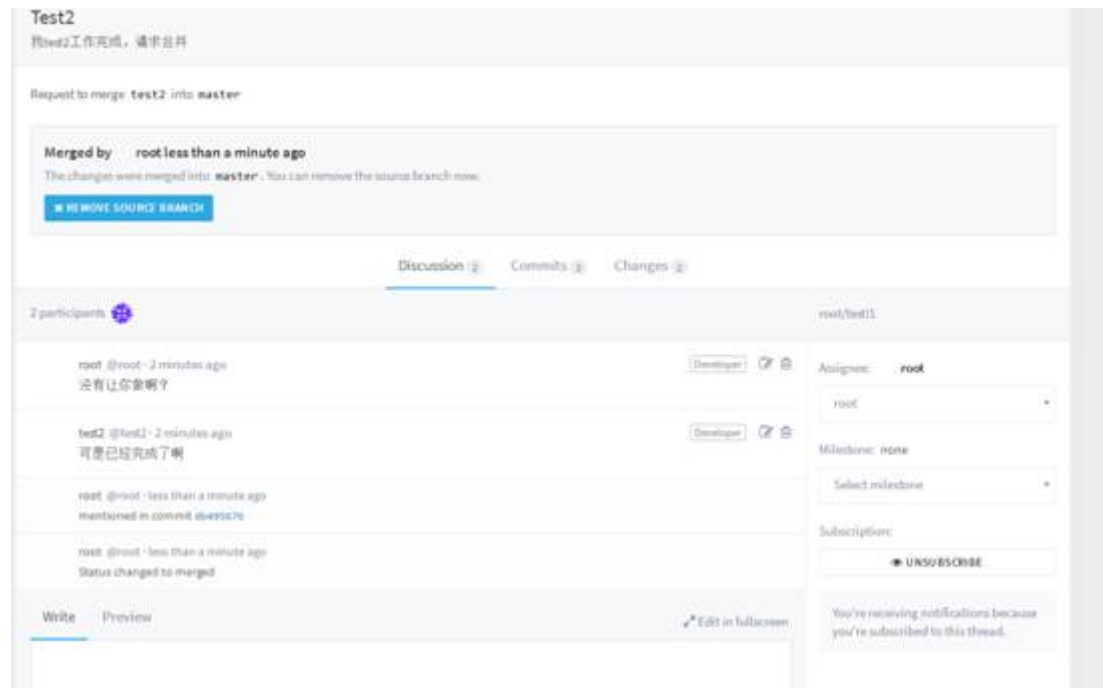
Subscription [UNSUBSCRIBE](#)

You're receiving notifications because you're subscribed to this thread.

e) 负责人接收到合并请求



f) 接受合并



## 1.1.8 备份

- Gitlab 创建备份

使用Gitlab一键安装包安装Gitlab非常简单, 同样的备份恢复与迁移也非常简单. 使用一条命令即可创建完整的Gitlab备份:

```
gitlab-rake gitlab:backup:create
```

使用以上命令会在/var/opt/gitlab/backups目录下创建一个名称类似为1393513186\_gitlab\_backup.tar的压缩包, 这个压缩包就是Gitlab整个的完整部分, 其中开头的1393513186是备份创建的日期.

- Gitlab 修改备份文件默认目录

你也可以通过修改/etc/gitlab/gitlab.rb来修改默认存放备份文件的目录:

```
gitlab_rails['backup_path'] = '/mnt/backups'
```

/mnt/backups修改为你想存放备份的目录即可, 修改完成之后使用gitlab-ctl reconfigure命令重载配置文件即可.

- Gitlab 自动备份

也可以通过crontab使用备份命令实现自动备份:

```
sudo su -
```

```
crontab -e
```

加入以下, 实现每天凌晨2点进行一次自动备份:

```
0 2 * * * /opt/gitlab/bin/gitlab-rake gitlab:backup:create
```

- Gitlab 恢复

同样, Gitlab的从备份恢复也非常简单:

```
# 停止相关数据连接服务
```

```
gitlab-ctl stop unicorn
```

```
gitlab-ctl stop sidekiq
```

```
# 从1393513186编号备份中恢复
```

```
gitlab-rake gitlab:backup:restore BACKUP=1393513186
```

```
# 启动Gitlab
```

```
sudo gitlab-ctl start
```

- Gitlab迁移

迁移如同备份与恢复的步骤一样, 只需要将老服务器/var/opt/gitlab/backups目录下的备份文件拷贝到新服务器上的/var/opt/gitlab/backups即可(如果你没修改过默认备份目录的话). 但是需要注意的是新服务器上的Gitlab的版本必须与创建备份时的Gitlab版本号相同. 比如新服务器安装的是最新的7.60版本的Gitlab, 那么迁移之前, 最好将老服务器的Gitlab 升级为7.60在进行备份.

- 其他

最新版本的Gitlab已经修复了HTTPS设备的BUG, 现在使用[官方HTTPS配置](#)即可轻松启用HTTPS.

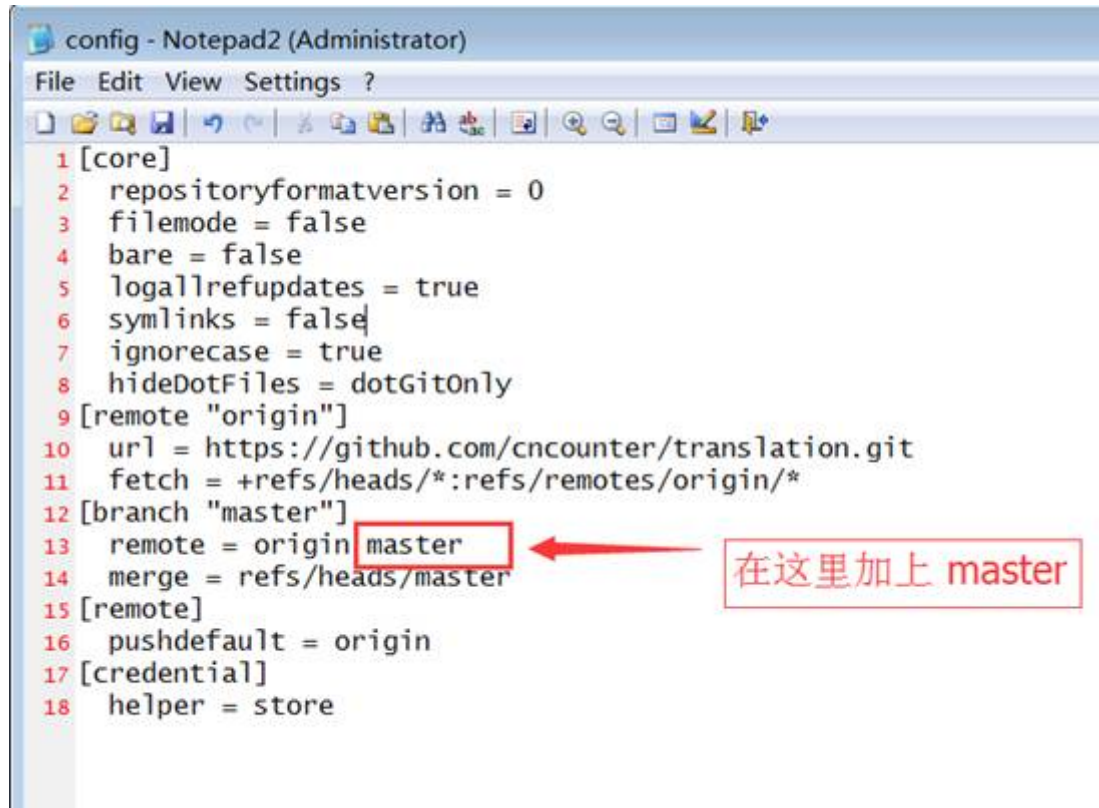
### 1.1.9 遇到问题

a) Your configuration specifies to merge with the ref 'master'

原因: 这可能是 TortoiseGit 当前版本的一个 BUG

解决:





```
1 [core]
2 repositoryformatversion = 0
3 filemode = false
4 bare = false
5 logallrefupdates = true
6 symlinks = false
7 ignorecase = true
8 hideDotFiles = dotGitOnly
9 [remote "origin"]
10 url = https://github.com/cncounter/translation.git
11 fetch = +refs/heads/*:refs/remotes/origin/*
12 [branch "master"]
13 remote = origin master
14 merge = refs/heads/master
15 [remote]
16 pushdefault = origin
17 [credential]
18 helper = store
```

在这里加上 master

## 1.2. Gerrit

### 1.2.1 简介

gerrit的工作模式是所有真正的merge由gerrit负责，我们的提交只是提交给了gerrit，gerrit会通知项目的负责人来审核代码，同时也会调用一些自动化测试构建工具来检查代码是否符合要求，只有都通过后才会的合并进主干代码。所以可以把gerrit看成是代码和提交者之间的一道门。

### 1.2.2. 环境搭建

#### a) Jdk安装

安装最新版本Oracle JRE/JDK，使用PPA(personal package archive)WEBUPD8的方式安装。

```
sudo apt-get install python-software-properties
```

```
sudo add-apt-repository ppa:webupd8team/java
```

```
sudo apt-get update
```

如果你正在使用OPenJDK，那么先将它移除以免产生任何的冲突

```
sudo apt-get remove --purge openjdk*
```

然后使用下面的命令安装Oracle Java 7

```
sudo apt-get install oracle-java7-installer
```

b) 添加gerrit用户

```
adduser gerrit
```

c) 建立存放代码的目录

```
mkdir /home/gerrit/gerrit_dir/
```

d) 安装git

```
sudo apt-get install git
```

e) 安装

```
sudo java -jar gerrit-2.11.4.war init -d /home/gerrit/gerrit_dir/
```

一路按回车，先以默认方式安装，之后再手动修改配置文件。

f) 配置gerrit

```
vim review_site/etc/gerrit.config
```

```
[gerrit]
  basePath = /home/gerrit/gerrit_dir
  canonicalWebUrl = http://10.24.6.12:8081
[database]
  type = h2
  database = db/ReviewDB
[index]
  type = LUCENE
[auth]
  type = HTTP
[sendemail]
  enable = true
  smtpServer = 
  smtpUser = 
  smtpPass = 
  from = 
[container]
  user = gerrit
  javaHome = /opt/java/jdk1.8/jre
[sshd]
  listenAddress = *:29418
[httpd]
  listenUrl = proxy-https://*:8081/
[cache]
  directory = cache
~
```

代码存放目录  
访问的url  
默认db  
SMTP邮件设置  
JDK路径

## g) NGINX 安装

```
sudo apt-get install nginx
```

## h) NGINX 配置

查看/etc/nginx/nginx.conf配置文件读取那些sites配置文件

```

##
# Basic Settings
##

sendfile on;
tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 65;
types_hash_max_size 2048;
# server_tokens off;

# server_names_hash_bucket_size 64;
# server_name_in_redirect off;

include /etc/nginx/mime.types;
default_type application/octet-stream;

##
# Logging Settings
##

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

##
# Gzip Settings
##

gzip on;
gzip_disable "msie6";

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

##
# nginx+maxi config
##
# Uncomment it if you installed nginx+maxi

#include /etc/nginx/maxi_core.rules;

##
# nginx+passenger config
##
# Uncomment it if you installed nginx+passenger

#passenger_root /usr;
#passenger_ruby /usr/bin/ruby;

##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;

```

nginx默认加载的site地址

发现这里只include /etc/nginx/sites-enabled里面的文件

在/etc/nginx/sites-enabled新增一个端口映射虚拟服务器文件gitlab

配置nginx端口映射，8082映射到gerrit的8081端口

```

##
# Basic Settings
##

sendfile on;
tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 65;
types_hash_max_size 2048;
# server_tokens off;

# server_names_hash_bucket_size 64;
# server_name_in_redirect off;

include /etc/nginx/mime.types;
default_type application/octet-stream;

##
# Logging Settings
##

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

##
# Gzip Settings
##

gzip on;
gzip_disable "msie6";

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

##
# nginx+maxi config
##
# Uncomment it if you installed nginx+maxi
##

#include /etc/nginx/maxi.conf;

##
# nginx+passenger config
##
# Uncomment it if you installed nginx+passenger
##

#passenger_root /usr;
#passenger_ruby /usr/bin/ruby;

##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;

```

nginx默认加载的site地址

i) 服务重启

### 1.2.3 账户创建

以下模拟admin账户创建

a) NGINX 账户访问：创建 htpasswd.conf 文件，并添加 admin 用户、密码到文件中

```
# touch /etc/gerrit_account/htpasswd.conf
```

```
# htpasswd /etc/gerrit_account/htpasswd.conf admin
```

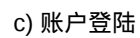
默认第一个登录 Gerrit 的用户是 Admin。

```
gerrit@gerrit:~/gerrit_dir/bin$ sudo htpasswd /etc/gerrit_account/htpasswd.conf admin
New password:
Re-type new password:
Adding password for user admin
gerrit@gerrit:~/gerrit_dir/bin$
```

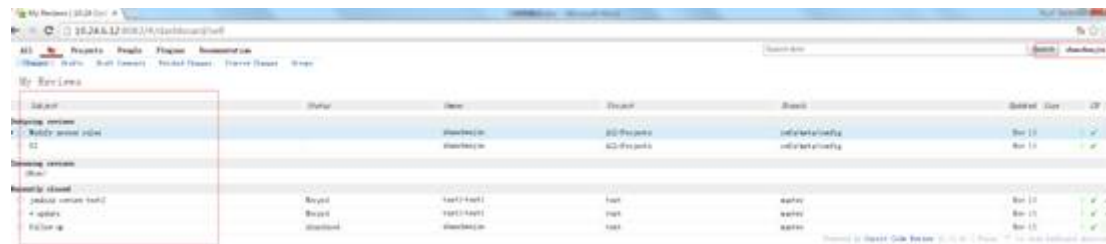
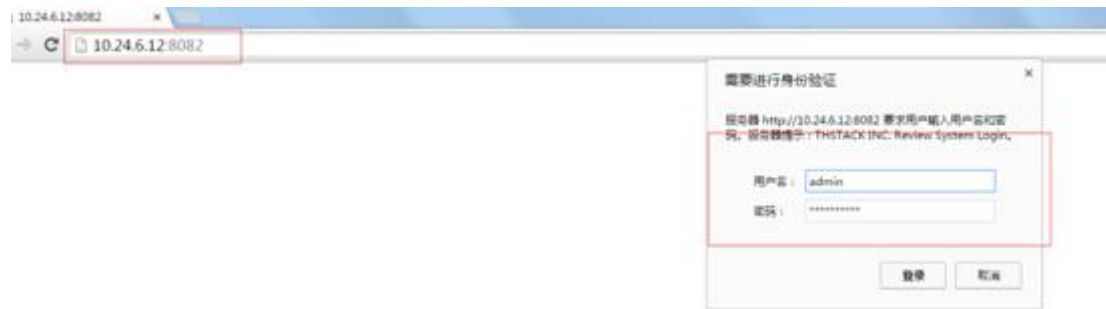
---

## b) 邮箱验证

注册邮箱，并添加对应密钥



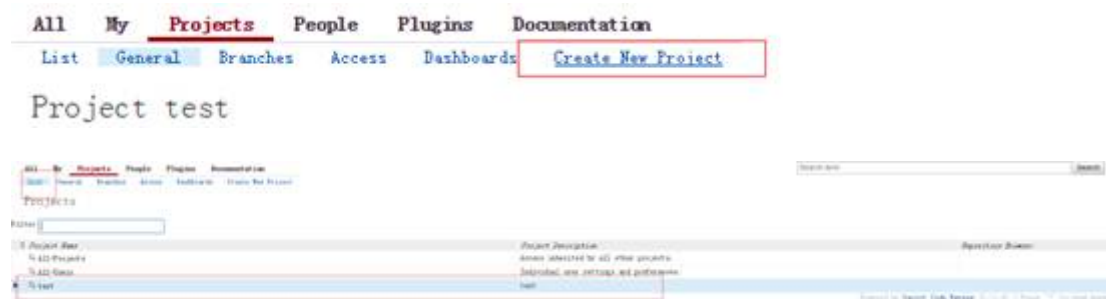
<http://www.cnblogs.com/zhanchenjinn/p/5032218.html>



## 1.2.4 Gitlab 支持

### 1.2.4.1 克隆gitlab test项目

#### a) 创建test项目



#### b) 使用gitlab的 gerrit身份从gitlab克隆 test项目

注意：gerrit的ssh key需要同时能登陆gitlab和gerrit)



```
$ cd /home/gerrit/gerrit_dir
```

```
$ rm -rf test.git
```

```
$ git clone --bare git@10.24.6.8:root/test.git
```

```
all-projects.git All-Users.git bin cache data db etc index lib logs plugins static test.git tap
gerrit@gerrit:~/gerrit_dir$ rm -rf test.git/
gerrit@gerrit:~/gerrit_dir$ git clone --bare git@10.24.6.8:root/test.git
Cloning into bare repository 'test.git'...
remote: Counting objects: 52, done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 52 (delta 16), reused 0 (delta 0)
Receiving objects: 100% (52/52), done.
Resolving deltas: 100% (16/16), done.
Checking connectivity... done.
gerrit@gerrit:~/gerrit_dir$ ls
all-projects.git All-Users.git bin cache data db etc index lib logs plugins static test.git tap
gerrit@gerrit:~/gerrit_dir$
```

#### 1.2.4.2 自动同步项目到gitlab

同步 Gerrit 的test项目到 Gitlab 上的test项目

当用户 git review 后，代码通过 jenkins 测试、人工 review 后，代码只是 merge 到了 Gerrit 的 test-project1 项目中，并没有 merge 到 Gitlab 的 test 项目中，所以需要当 Gerrit test项目仓库有变化时自动同步到 Gitlab 的 test项目仓库中。Gerrit 自带一个 Replication 功能，同时我们在安装 Gerrit 时候默认安装了这个 Plugin。现在只需要添加一个 replication.config 给 Gerrit

a) 进入gerrit代码存放目录gerrit\_dir,增加replication.config

```
gerrit@gerrit:~/gerrit_dir$ vim etc/replication.config
```

文件内容为：

```
[remote "test"]

projects = test

url = git@10.24.6.8:root/test.git

push = +refs/heads/*:refs/heads/*

push = +refs/tags/*:refs/tags/*

push = +refs/changes/*:refs/changes/*

threads = 3
```

```
etc/replication.config NEW! /L, 194C with diff
gerrit@gerrit:~/gerrit_dir$ vim etc/replication.config
[remote "test"]
    projects = test
    url = git@10.24.6.8:root/test.git
    push = +refs/heads/*:refs/heads/*
    push = +refs/tags/*:refs/tags/*
    push = +refs/changes/*:refs/changes/*
    threads = 3
..
```

#### b) 设置gerrit用户的 ~/.ssh/config

文件内容为：

Host 10.24.6.8:

IdentityFile ~/.ssh/id\_rsa

PreferredAuthentications publickey

```
gerrit@gerrit:~/gerrit_dir$ vim /home/gerrit/.ssh/config
Host 10.24.6.8:
    IdentityFile ~/.ssh/id_rsa
    PreferredAuthentications publickey
..
```

#### c) 在gerrit用户的 ~/.ssh/known\_hosts 中，给 10.24.6.8(gitlab) 添加 rsa 密钥

gerrit@gerrit:~/gerrit\_dir\$ sh -c "ssh-keyscan -t rsa 10.24.6.8 >> /home/gerrit/.ssh/known\_hosts"

```
gerrit@gerrit:~/gerrit_dir$ sh -c "ssh-keyscan -t rsa 10.24.6.8 >> /home/gerrit/.ssh/known_hosts"
# 10.24.6.8 SSH-2,0-OpenSSH_6.6p1 Ubuntu-2ubuntu2.3
111jBSczy4nGvskKJaoIaAB19vrr1Uc1sHrP5WZ1pmKdG1KCEVkbS8=
cdsa-sha2-nistp256 AAAAEZVjZRRhLn0Y1itbLzdhRgNTYAAAAB1bLzdhRgNTYAAAAB
B3K5oG6xUShPOOSLB1Y1tPULHdFv464dMFPz/9Rv8N6z9vHISCot/PwF0/PZaQ1Q0d8vTHHFIQZvYMcU/r0=
11f0vxxxIXsDy3+F3v6cP+8u3w=13LerFPYrYBNUdIBHocGa2skyoI= ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgOCeLxKOKVUDhbnHw70B+P5yB4B61ow9L
zeb1BUZGVE0jdt=30AE+804W8JrjVUMzP9cB12NcEEogckW79hLB+v8R4KJEPjAkaufAApG7Y7BIsLGKJvSkbV5gkH0tSxHaoJ2/OP10JD/Rvc8VPWfU7/QJ5IA6grs
KJIE9MQ==
10.24.6.8 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgOCeLxKOKVUDhbnHw70B+P5yB4B61ow9L
Fv9UcLjY1pJLCJB4uIRPhcOF301vDkn0A9pBVfJpNGnd00bWJ1n9LUDFNCEzenf3j5FrVMS3eMTE41pkv/fYb1F/DP+uuAe8L6OustRagKaSPsV55YC7ZghNFbLFUJ
RAnsy76d300RUH66/CHjgLuQVQJRYKyzf5GvdeKzJQ1/nvsy8vqQIH5Ez1PC03FK1dr8erF+2k3P1NCX6tyvHEfakvM8FOft+H21saQrVEKp9jtu3eInVf4UE5FG3
```

gerrit@gerrit:~/gerrit\_dir\$ sh -c "ssh-keygen -H -f /home/gerrit/.ssh/known\_hosts"

```
gerrit@gerrit:~/gerrit_dir$ sh -c "ssh-keygen -H -f /home/gerrit/.ssh/known_hosts"
/home/gerrit/.ssh/known_hosts updated.
Original contents retained as /home/gerrit/.ssh/known_hosts.old
WARNING: /home/gerrit/.ssh/known_hosts.old contains unhashed entries
Delete this file to ensure privacy of hostnames
```

#### d) 重新启动 Gerrit 服务

```
gerrit@gerrit:~/gerrit_dir$ ./bin/gerrit.sh restart
```

```
gerrit 10000 10431 0 11:17 pty/z 00:00:00 grep --color=auto gerrit
gerrit@gerrit:~/gerrit_dir$ ./bin/gerrit.sh restart
Stopping Gerrit Code Review: OK
Starting Gerrit Code Review: OK
gerrit@gerrit:~/gerrit_dir$
```

#### e) Gerrit 的自动功能配置完毕

在 gerrit 文档中有一个 \${name} 变量用来复制 Gerrit 的所有项目，这里并不需要。如果有多个项目需要复制，则在 replication.config 中添加多个 [remote ....] 字段即可。务必按照上面步骤配置复制功能。

## 1.2.5. Jenkins 支持

### 1.2.5.1. Jenkins 账户

#### a) 创建用户jenkins

```
$ sudo -u gerrit -H htpasswd /home/gerrit_account/htpasswd.conf Jenkins
```

```
jenkins@gerrit:/home/gerrit/gerrit_dir$ sudo htpasswd /etc/gerrit_account/htpasswd.conf jenkins
[sudo] password for jenkins:
Sorry, try again.
[sudo] password for jenkins:
New password:
Re-type new password:
Adding password for user jenkins
jenkins@gerrit:/home/gerrit/gerrit_dir$
```

#### b) 用jenkins用户登录Gerrit更新邮箱xxxxxxx@qq.com并验证

#### c) 创建jenkins密钥

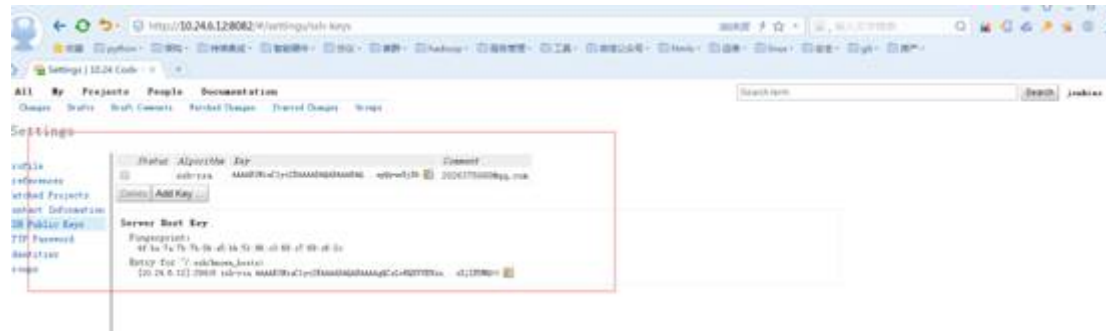
```
$ sudo ssh-keygen -C xxxxxx@qq.com
```

```
jenkins@gerrit:/home/gerrit/gerrit_dir$ sudo ssh-keygen -C 2026375060@qq.com
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /home/jenkins/.ssh/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jenkins/.ssh/id_rsa.
Your public key has been saved in /home/jenkins/.ssh/id_rsa.pub.
The key fingerprint is:
9e:0f:1a:b9:c0:a5:1b:d3:19:22:0f:92:27:f5:88:21 2026375060@qq.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
|E .                    |
|o .                    |
|o o o S                |
|* = = =                |
|B = +                  |
| = + o                  |
|. o .                  |
+-----+

```

## 获取公钥

```
jerkins@errit:/home/errit/errit$ cat /home/jerkins/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAI2H89C/TJ6u5466FKK4YMTBLo6eJpDHT+g58Kw37r1vv601ebnFCE+Uz32v4yFP0+5kYUJbGw3cnoc5s
n1S2K1v167s0HF3dc0r4uJqk16u1h+Kzd,4B0Hexp:0u0xrcwvZC+140h0q11u04Jk1v5r0sh3ZP06Jhvs1/4D05+4dF0B51n0R0K1N5h0U0C6R0r1J07s
P0k.B2S0V0P7gealuhZ0Cd0d0q0h0q0F+Y0z0t0Bd01Df0q31Zv62r0H7/T2+99R0r1B0Z1P53zeJF4rCv90gR1n1d0f0e3J0c2dep0u0JbJ0d 2026375060@qq.c
om
```

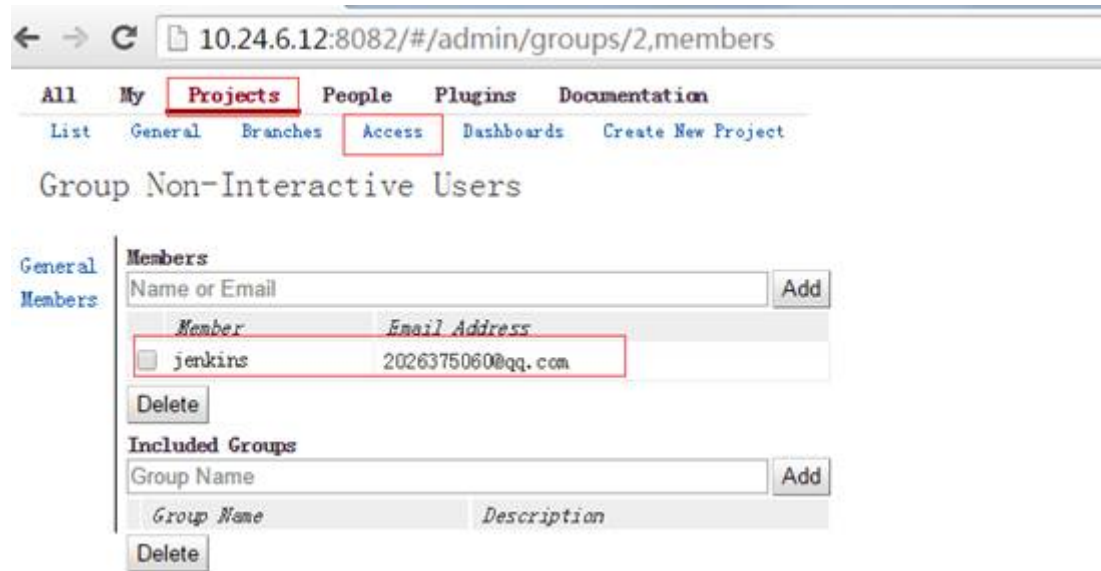


## 管理员登录gerrit

Projects->Access

Global Capabilities->Stream Events 点击 Non-Interactive Users

添加 Jenkins@zjc.com 用户到 'Non-Interactive Users' 组



### 1.2.5.2 Verified功能支持

Verified功能，就是利用jenkins确定gerrit提交的代码是否有问题，需要jenkins给个确认

a) 如果安装Gerrit时没有或者没有选择添加Verified标签功能['lable Verified']，需要自己添加

```
gerrit@gerrit:~$ cd cfg/

gerrit@gerrit:~/cfg$ git config --global user.name "admin"

gerrit@gerrit:~/cfg$ git config --global user.email "423884230@qq.com"

gerrit@gerrit:~/cfg$ git remote add origin ssh://admin@10.24.6.12:29418/All-Projects

gerrit@gerrit:~/cfg$ git pull origin refs/meta/config

gerrit@gerrit:~/cfg$ vim project.config
```

在文件末添加

```
[label "Verified"]
```

```
function = MaxWithBlock
```

```
value = -1 Fails
```

```
value = 0 No score
```

```
value = +1 Verified
```

```
gerrit@gerrit:~/cfg$ git commit -a -m "Update permissions"
```

```
gerrit@gerrit:~/cfg$ git push origin HEAD:refs/meta/config
```

```
gerrit@gerrit:~$ rm -rf cfg/
```

### b) Verified 功能设置

现在提交的 Review 请求只有 Code Rivew 审核，我们要求的是需要 Jenkins 的 Verified 和 Code Review 双重保障，在 Projects 的 Access 栏里，针对 Reference: refs/heads/ 项添加 Verified 功能：

I Projects -> List -> All-Projects

I Projects -> Access -> Edit -> 找到 Reference: refs/heads/\* 项

I -> Add Permission -> Label Verified -> Group Name 里输入 Non-Interactive Users -> 回车 或者 点击Add 按钮 -> 在最下面点击 Save Changes 保存更改

## 1.2.6. 注意

如果想 Gitlab 上创建的项目使用 Gerrit 的 Code Review 功能，两个系统的用户必须统一，也就是说不管哪个用户使用 Gerrit，前提是这个用户在 Gitlab 和 Gerrit 上都已注册，邮箱一致、sshkey 一致。

## 1.2.7 遇到问题

- Error Missing project All-Projects

重装java -jar gerrit.war init -d /home/gerrit/gerrit\_dir

**注意：**如果重新初始化了gerrit，并且版本库路径与以前的一样，记得删除版本看路径下的All-projects.git 这个目录，不然第一个用户登录失败，无法获取超级管理员权限

- **Unable to authenticate user by Authorization request header. Check container or server configuration**

问题原因：由于版本库下有子目录，通过nginx代理是指定的目录不正确导致 把 proxy\_pass

http://192.168.0.122:8090/; 改为 proxy\_pass http://192.168.0.122:8090; ##不要最后的斜杠

- **com.google.gerrit.httpd.auth.container.HttpLoginServlet : Unable to authenticate user by Authorization request header. Check container or server configuration.**

**注意：**邮箱验证的时候，使用nginx提供的端口进入，不要直接使用gerrit的端口进入验证

测试这里是使用8082端口，不要访问8081端口

- 使用nginx用户验证的方式，无法使用sign out， gerrit无法调用nginx

## 1.3. Jenkins

### 1.3.1 简介

Jenkins是一个开源软件项目，旨在提供一个开放易用的软件平台，使软件的持续集成变成可能。

### 1.3.2. 环境搭建

#### a) Install Jenkins

```
wget -q -O - https://jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add -
```

```
sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ > /etc/apt/sources.list.d/jenkins.list'
```

```
sudo apt-get update
```

```
sudo apt-get install jenkins
```

#### b) Starting Jenkins

```
sudo nohup java -jar /usr/share/jenkins/jenkins.war > /var/log/jenkins_log 2>&1 &
```

#### c) 开启

```
http://xxxxx:8080/
```

#### d) 插件安装

Git Plugin

SSH Plugin

URL Trigger Plugin

Gerrit Trigger Plugin

#### e) 注册用户

先开启用户注册按钮，用户注册完毕以后，关闭用户注册



Gerrit Trigger

### 1.3.2. Gerrit Verified支持

#### 1.3.2.1. Gerrit Triggler插件配置

##### a) 安装gerrit trigger插件

Gerrit Trigger 配置文档<https://wiki.jenkins-ci.org/display/JENKINS/Gerrit+Trigger>

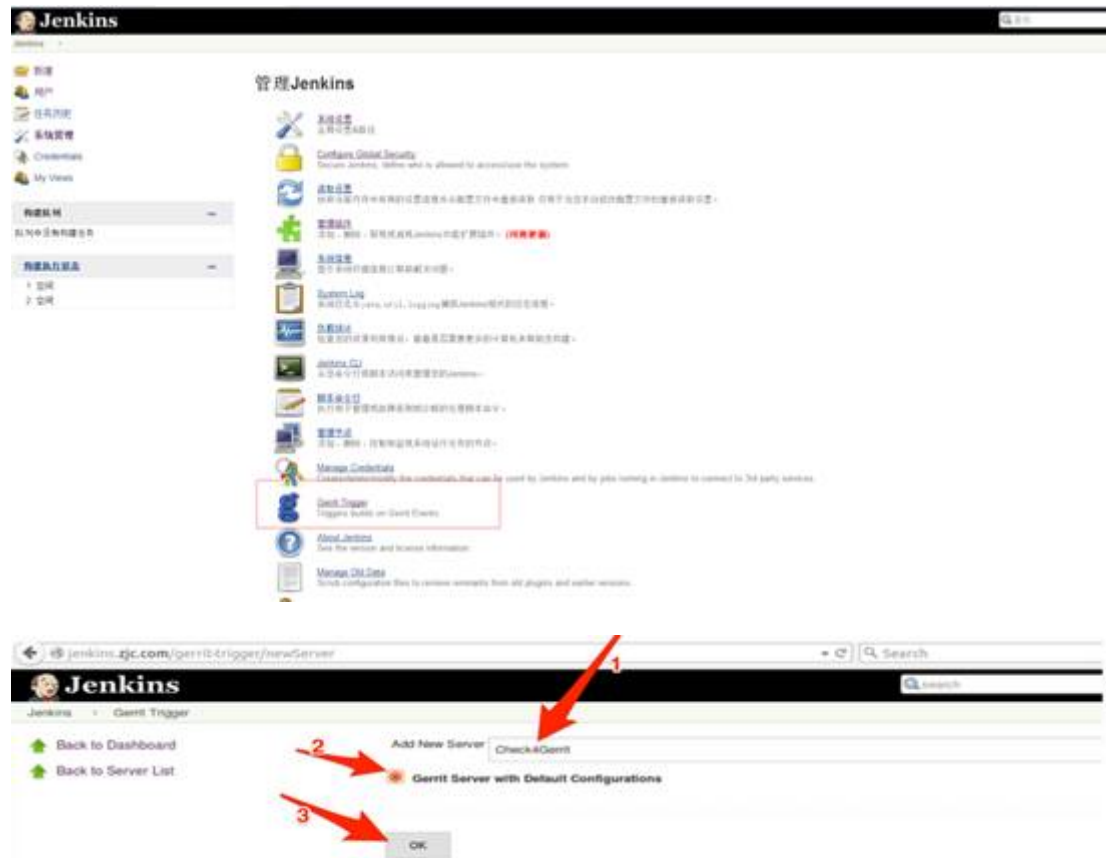
##### b) 设置Gerrit Trigger

主页面->Manage Jenkins->Gerrit Trigger->Add New Server

Add New Server : Check4Gerrit

勾选 Gerrit Server With Default Configurations





c) 填写gerrit的jenkins账户信息，并测试

如果点击Test Connection后出现如图所示的错误User jenkins has no capability to connect to Gerrit event stream.,这是因为因Jenkins 没有权限监听，参考Gerrit 的 'Stream Events'配置

Name: CheckGerrit

No Connection On Startup: ☐

Hostname: 10.24.6.12

Frontend URL: http://10.24.6.12:8082

SSH Port: 29418

Proxy:

Username: jenkins

Email: zhanchenjin.2008@163.com

SSH Keyfile: /home/jenkins/.ssh/id\_rsa

SSH Keyfile Password:

☐ Build Current Patches Only

User jenkins has no capability to connect to Gerrit event stream.

Test Connection

## d) Gerrit Triggler设置成功

Gerrit Mixed Events Playback is not supported. Verify if the connection has the REST API enabled and that the Gerrit Eventing plugin is installed and configured on the Gerrit Server.

Gerrit Connection Setting

Name: CheckGerrit

No Connection On Startup: ☐

Hostname: 10.24.6.12

Frontend URL: http://10.24.6.12:8082

SSH Port: 29418

Proxy:

Username: jenkins

Email: zhanchenjin.2008@163.com

SSH Keyfile: /home/jenkins/.ssh/id\_rsa

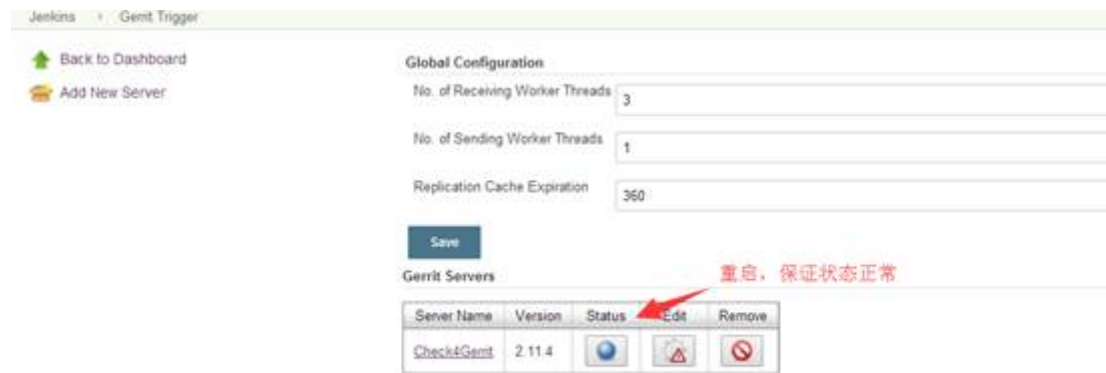
SSH Keyfile Password:

☐ Build Current Patches Only

Success

Test Connection

## e) Gerrit Triggler重启保证状态正常



### 1.3.2.1. Verified项目配置

只要gerrit有提交代码，jenkins都能检测到，并verified代码正确性，然后把结果告诉gerrit

#### a) Jenkins上创建test项目

用 jenkins 登录<http://10.24.6.12:8080/>



#### b) Git页面设置

参考文档<https://wiki.jenkins-ci.org/display/JENKINS/Gerrit+Trigger>

**项目设置**

☐ None  
☐ CVS  
☐ CVS Projectset  
☒ Git  
 Repositories

Repository URL:  1  
 Credentials:   2  
 Name:  3  
 Refspec:  3

Branches to build:   4

源代码位置:  6

Additional Behaviours:

☐ must pass security  
☒ Gerrit event

**Gerrit Trigger**  
 Choose a Server:   6

Silent Mode: ☐

Trigger on:
 

- ☐ Patchset Created 2
- ☐ Exclude Drafts
- ☐ Exclude Trivial Refbase
- ☐ Exclude No Code Change
- ☐ Draft Published 3
- ☐ Change Merged 4
- ☐ Comment Added
  - Verdict Category:  5
  - Value:  7

 1

**Dynamic Trigger Configuration**

Type	Pattern	Type	Pattern
Path	test	Path	-

 2
   

  
☐ Double Strict Forbidden File Verification

☐ Poll SCM  
☐ [URLTrigger] - Poll with a URL

## c) Build页面设置

Add build step -> Execute shell

```
cd $WORKSPACE
```

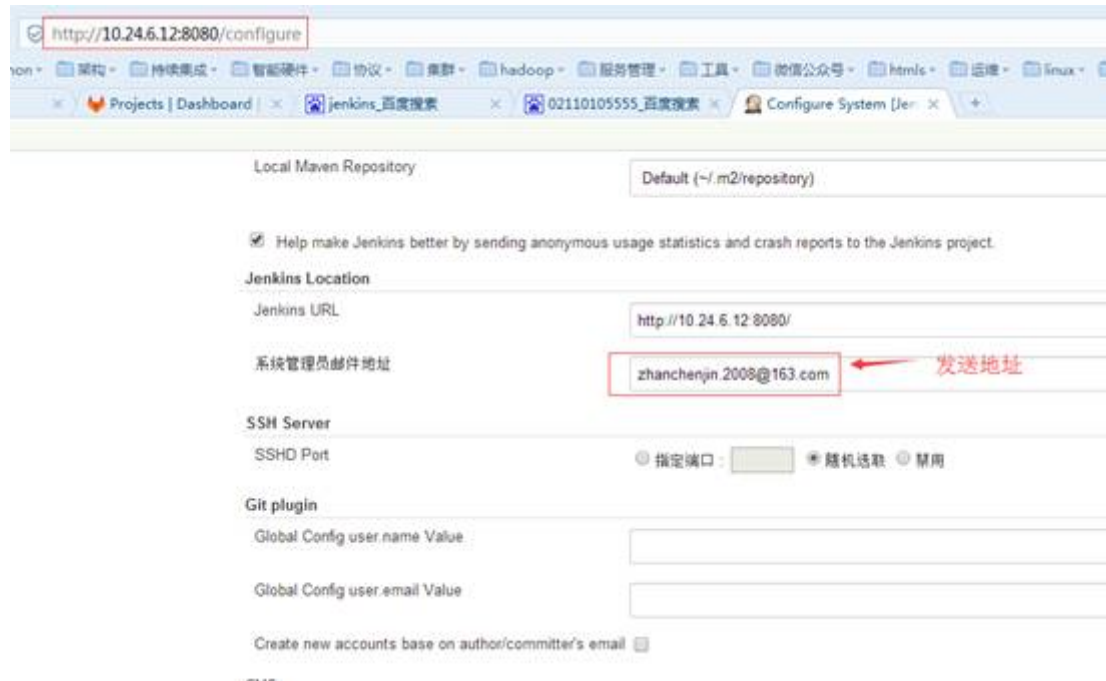
```
[ ! -e .testrepository ] && testr init
```

```
testr run
```

此处测试代码可依据项目情况而定，可为空



## 1.3.2. 邮件支持



The screenshot shows the 'Configure System [jenkins\_...]' page in a web browser. The 'SMTP' section is highlighted with a red box. The 'SMTP server' field is set to 'smtp.163.com'. The 'SMTP authentication' section is also highlighted with a red box, showing 'Username' as 'zhanchenjin.2008' and 'Password' as a masked field. The 'SMTP port' is set to '25'. The 'Reply-To Address' and 'From' fields are empty.

邮件发送测试

The screenshot shows the 'Test configuration' dialog box. It contains a 'Test email recipient' field with the value '423884230@qq.com'. A red arrow points to the 'Test configuration' button.

## 1.4. Git 客户端 review支持

需要安装git-review, readline

Apt-get install git-review

Linux : pip install readline

Windows : pip install pyreadline

```

logic_server2@drbd01:~/test$ sudo apt-get install git-review
sudo: password for logic_server2:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  git-review
1 upgraded, 1 newly installed, 0 to remove and 162 not upgraded.
Need to get 18.1 kB of archives.
After this operation, 120 kB of additional disk space will be used.
Get:1 http://mirrors.163.com/ubuntu/trusty/universe git-review all 1.23-1 [18.1 kB]
Fetched 18.1 kB in 0s (87.7 kB/s)
Selecting previously unselected package git-review.
Reading database ... 84046 files and directories currently installed.)
Preparing to unpack .../git-review_1.23-1_all.deb ...
Unpacking git-review (1.23-1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up git-review (1.23-1) ...
logic_server2@drbd01:~/test$ ls

```

## 1.5. Review+Test+Push实例

### 1.5.1. 流程

#### a) 下载项目

git clone <http://10.24.6.8/root/test.git>

cd test

#### b) 设置添加change\_id钩子

增加一个hook，自动增加chang\_id到提交日志里面，必须

gitdir=\$(git rev-parse --git-dir); scp -p -P 29418 [test1@10.24.6.12:hooks/commit-msg](mailto:test1@10.24.6.12:hooks/commit-msg) \${gitdir}/hooks/

#### c) 改变代码：

vim t99.txt

#### d) 上传代码：

git add .

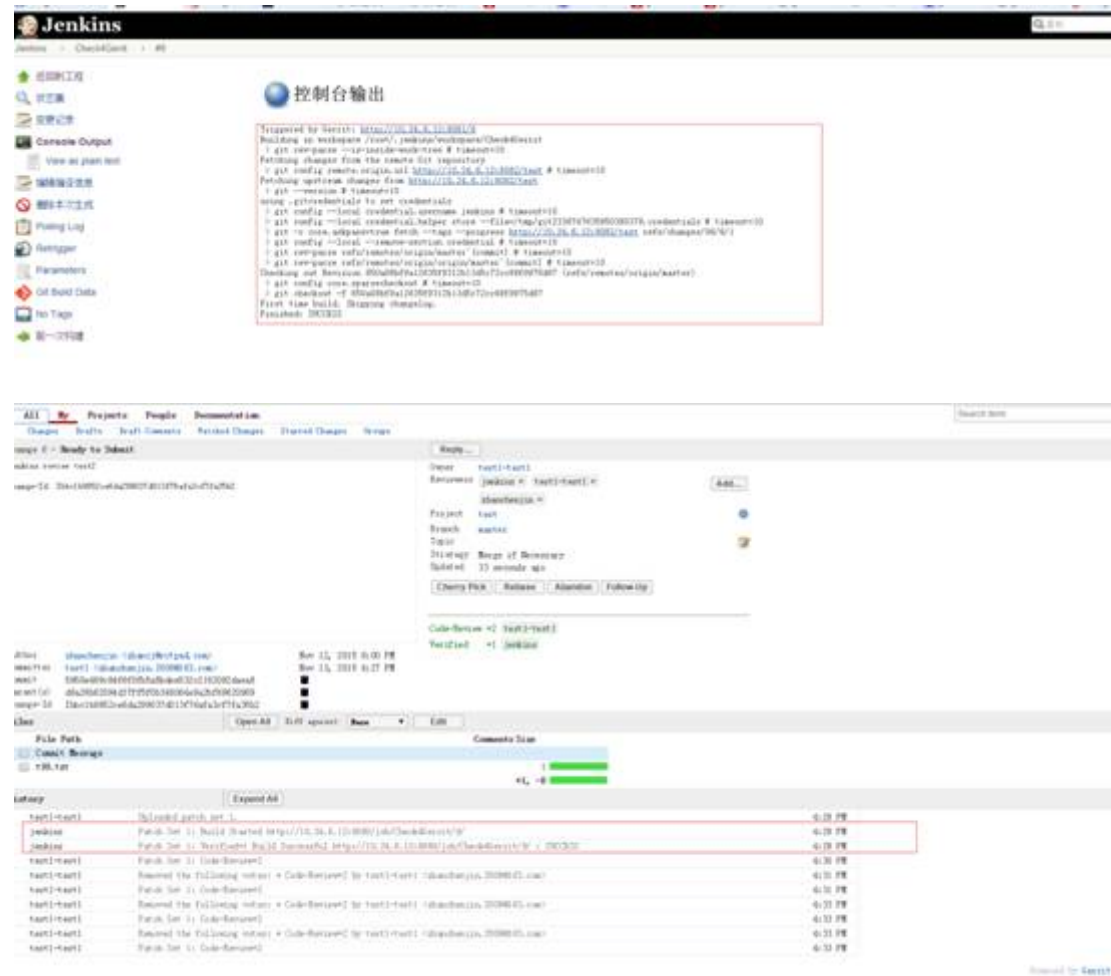
注意使用amend进行提交changid

git commit --amend

#### e) 在gerrit上看到提交



f) Jenkins 自动获取提交信息并判断是否verified

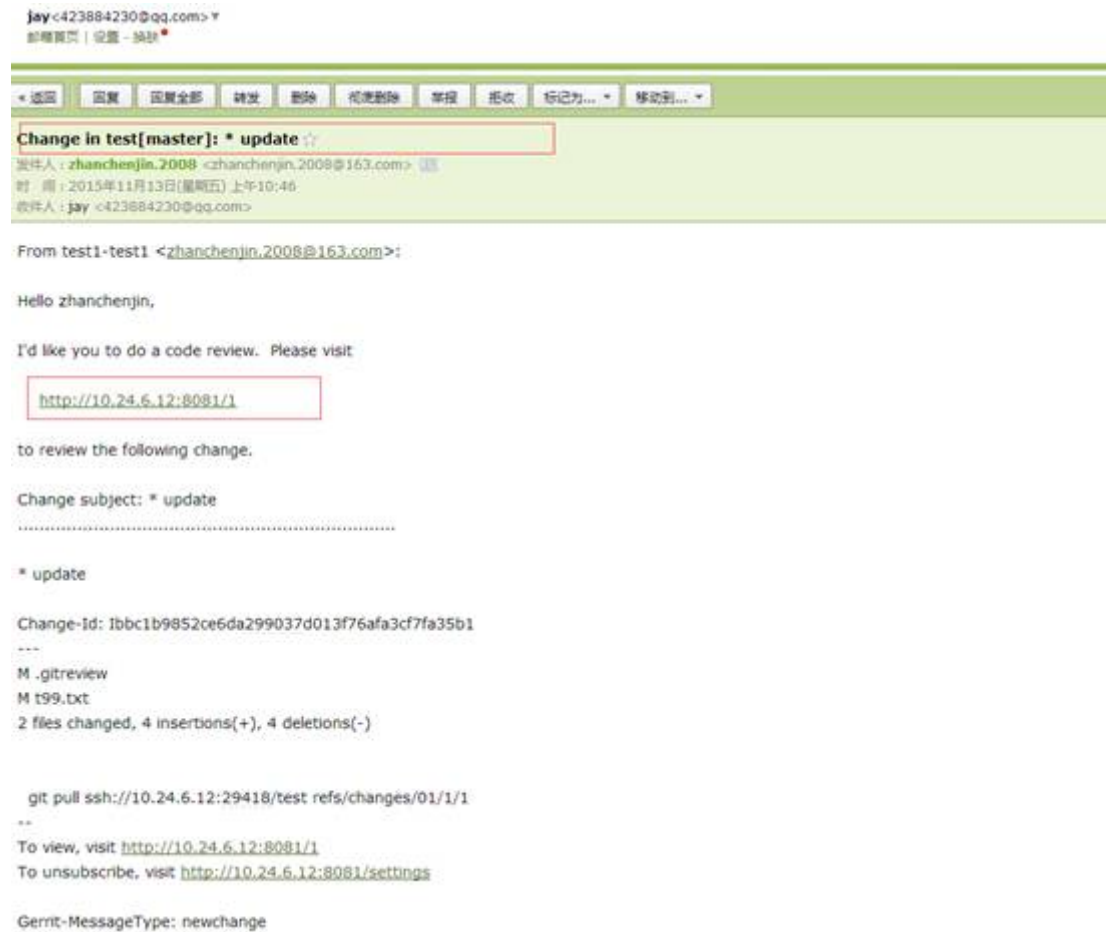


g) 在gerrit上添加review成员

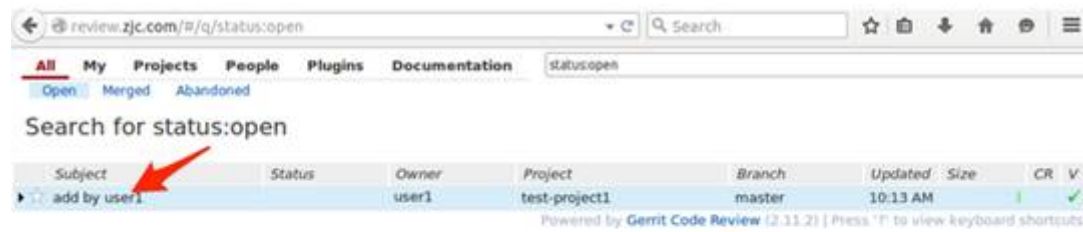




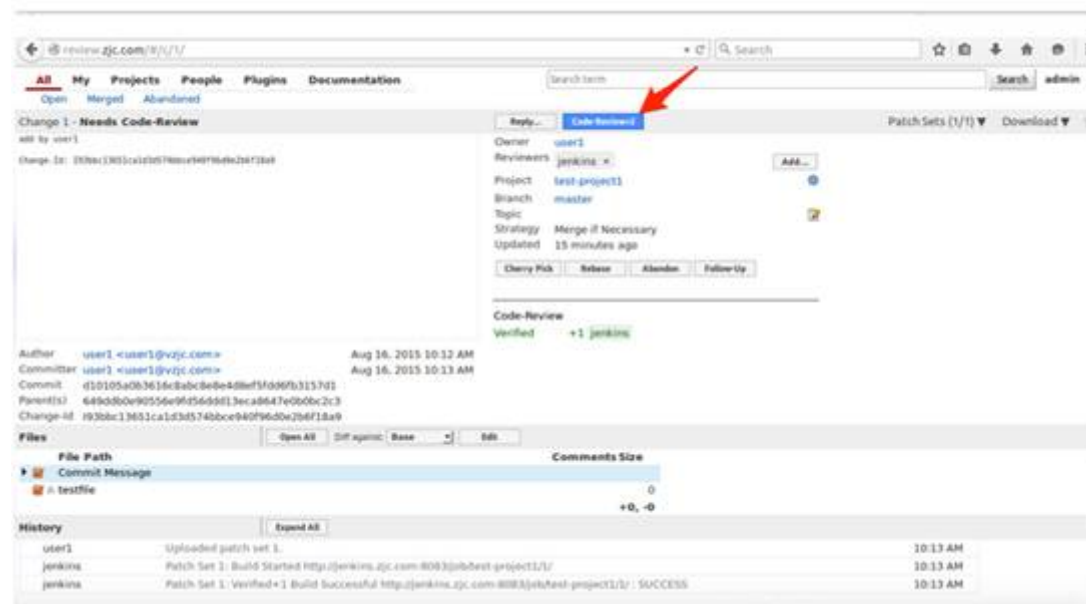
h) REVIEW成员收到邮件



#### i) REVIEW 人员登陆处理



j) admin点击CodeReview2



k) Jenkins自动点击，然后submit

Reply...

Owner

test1-test1

Reviewers

jenkins x test1-test1 x zhanchenjin x

Add...

Project

test

Branch

master

Topic

Strategy

Merge if Necessary

Updated

64 seconds ago

Cherry Pick

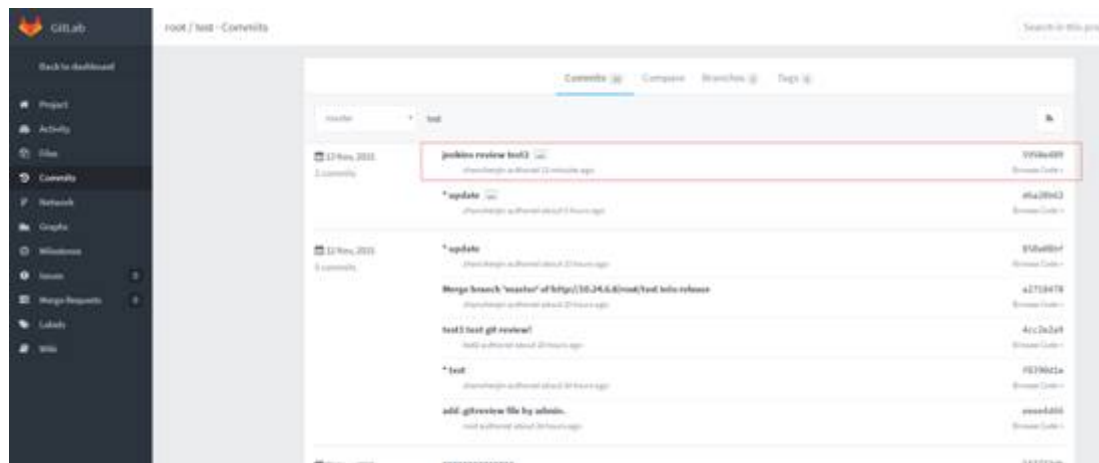
Rebase

Abandon

Follow-Up

Submit

## l) GitLab 显示commit记录



## m) 邮件通知REVIEW人员说明

Reply... 3

还行, 提交吧 4

-2 -1 0 +1 +2

Code-Review ☐ ☐ ☐ ☐ ☒ Looks good to me, approved

.gitreview

Line 2: sxxxxx 2

t99.txt

Line 1: 怎么可以写成这样? 1

Post Cancel

## n) REVIEW 提交者收到消息

<< 返回 回复 回复全部 转发 删除 举报 标记为 移动 更多

Change in test[master]: \* update

发件人: 我 <shanchenjin.2008@163.com>

收件人: test1-test1 <shanchenjin.2008@163.com>

时 间: 2015年11月13日 10:58 (星期四)

您11天来花了多少时间? 用网页看邮件, 轻松记录邮件开发! 马上使用发邮件>>

From shanchenjin <423884230@qq.com>:

shanchenjin has posted comments on this change.

Change subject: \* update

.....

Patch Set 1:

(2 comments)

还行, 提交吧

http://10.24.6.32:8081/#/c/1/1/.gitreview

File .gitreview:

Line 2

xxxxxx

http://10.24.6.32:8081/#/c/1/1/t99.txt

File t99.txt:

Line 1:

## 1.5.2. 遇到问题

### a) 设置git提交的默认编辑器为vim

If you want to set the editor only for Git, do either (you don't need both):

- Set core.editor in your Git config: `git config --global core.editor "vim"`
- Set the GIT\_EDITOR environment variable: `export GIT_EDITOR=vim`

### b) missing Change-Id in commit message footer

需要安装插件，并使用 `git commit --amend` 提交代码，看截图部分

```
ogic_server2@drbd01:~/test$ git review
You are about to submit multiple commits. This is expected if you are
submitting a commit that is dependent on one or more in-review
commits. Otherwise you should consider squashing your changes into one
commit before submitting.

The outstanding commits are:

7de2c1 (HEAD, master) *test111 Change-Id: I33b7ab7a64d0604356b17fa0d9dd28540cb6d62a
#22e78 Merge branch 'master' of http://10.24.6.8/root/test
a32228 test

Do you really want to submit the above commits?
Type 'yes' to confirm, other to cancel: yes
error: Processing changes: refs: 1, done
error: ERROR: missing Change-Id in commit message footer
error:
error: Hint: To automatically insert Change-Id, install the hook:
error:  gitdir=$(git rev-parse --git-dir); scp -p -P 29418 test1@10.24.6.12:hooks/commit-msg $(gitdir)/hooks/
error: And then amend the commit:
error:  git commit --amend
error:
o ssh://test1@10.24.6.12:29418/test.git
! [remote rejected] HEAD -> refs/publish/master (missing Change-Id in commit message footer)
error: failed to push some refs to 'ssh://test1@10.24.6.12:29418/test.git'
ogic_server2@drbd01:~/test$
```

### c) GIT REVIEW remote rejected for change closed

remote: Processing changes: refs: 1, done

To ssh://xxxxx@gerrit.dev.xxxxx.net:29418/xxxxxx

! [remote rejected] HEAD -> refs/for/master (change 14823 closed)

error: failed to push some refs to 'ssh://xxxxx@gerrit.dev.xxxxx.net:29418/xxxxxx'

原因是注释和之前提交的一样，并且已经被closed

使用 `git commit --amend` 修改注释

### d) git rebase --continue Bug for need merge

```
t99.txt" 2L, 38L written
ogic_server2@drbd01:~/test$ git rebase --continue
t99.txt: needs merge
You must edit all merge conflicts and then
mark them as resolved using git add
ogic_server2@drbd01:~/test$
```

例如冲突文件为foo.c

使用git-svn rebase获取svn服务器上的最新文件，导致冲突，

不过此时svn版本信息已经添加到本地git库中(通过git log可以查看)

打开foo.c，修改代码，解决冲突；

执行git rebase --continue，git提示我：

You must edit all merge conflicts and then

mark them as resolved using git add

执行git add foo.c，告知git已完成冲突解决；

再次执行git rebase --continue，提示"Applying: git v1"，此时"git v1"版本又一次成功加入本地版本库，你可通过git log

查看；

执行git-svn dcommit将foo.c的改动同步到svn中心库，到此算是完成一次冲突解决。

git比较智能，会提示你解决问题的方法，所以遇到问题时注意看提示文档。

2015-12-09 11:13:03

分类: [敏捷开发架构](#)

标签: [敏捷开发](#), [持续集成](#), [gitlab+gerrit+jenkins](#)

好文要顶

关注我

收藏该文



若-飞

关注 - 0

粉丝 - 1

+加关注

2

0

» 下一篇：[Nginx文件上传下载实现与文件管理](#)

posted @ 2015-12-09 11:14 若-飞 阅读(5327) 评论(3) 编辑 收藏

## 评论列表

#1楼 2015-12-09 12:21 Herb

之前搭过一次，没成功，收藏慢慢看

支持(0) 反对(0)

#2楼 2015-12-09 14:48 wdwwtzy

可真长.....

支持(0) 反对(0)

#3楼 2016-02-26 17:15 =金刚=

不错~

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】腾讯云上实验室 1小时搭建人工智能应用

【推荐】可嵌入您系统的“在线Excel”！SpreadJS 纯前端表格控件

【推荐】阿里云“全民云计算”优惠升级





#### 最新IT新闻:

- 世界首条量子通信干线开通 首次洲际量子通信实现
  - 天猫承认说明和文案抄袭 永久下线"智能测肤"功能
  - 巨人网络豪掷8.2亿控股投哪网 签严苛对赌协议
  - 饿了么科技值要爆炸：继无人机后，语音点餐系统横空出世
  - 继中国后，韩国成为第二个明确禁止ICO的国家
- » 更多新闻...



#### 最新知识库文章:

- 如何阅读计算机科学类的书
  - Google 及其云智慧
  - 做到这一点，你也可以成为优秀的程序员
  - 写给立志做码农的大学生
  - 架构腐化之谜
- » 更多知识库文章...

Copyright ©2017 若-飞