1.2.4 Gitlab 支持 1.2.4.1 克隆gitlab test项目a) 创建test\_project项目（gitlab的测试project同名）,请注意，只有验证正确邮箱后，gerrit页面才会出现create new project按键   
  
file://C:\Users\pc304\AppData\Local\Temp\ct_tmp/1.png  
  
  
  
  
  
b) 使用gitlab的gerrit身份从gitlab克隆test项目   
（项目克隆下来，需要做几步操作：第一步：增加.review文件，这步因为测试tanchun用户无法push的实验，提前做了。第二步：增加replication文件，同步gerrit下面的test\_project到gitlab里面）  
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@192.168.0.177:root/test.git 为了保持gitlab与gerrit的数据一致  
  
file://C:\Users\pc304\AppData\Local\Temp\ct_tmp/3.png  
  
  
  
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（增加replication文件，注意路径是/home/gerrit/gerrit\_dir/etc/）  
  
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 Gerrittest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  
  
  
  
第一步：  
1.1.4.2 用 gerrit(gitlab中的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 423884230@qq.com  
gerrit@gerrit:~/cfg/cfg$ git clone http://192.168.0.177/root/test.git  
gerrit@gerrit:~/cfg/cfg$ cd test/  
  
  
b) 添加gitreview文件 （注意，里面增加的是gerrit服务器的ip,因为reivew的工作是在192.168.0.179上）  
gerrit@gerrit:~/cfg/cfg/test$ vim .gitreview  
文件内容为：  
[gerrit]  
host=192.168.0.179   
port=29418  
project=你在gitlab上面新建的project名  
  
  
  
  
第二步：  
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"] -----------远程  
projects = test  
url = git@192.168.0.177:root/test.git 就是克隆下来时的路径  
push = +refs/heads/\*:refs/heads/\*  
push = +refs/tags/\*:refs/tags/\*  
push = +refs/changes/\*:refs/changes/\*  
threads = 3  
  
  
第三步：（增加jenkins的触发文件.testr.conf）  
用gerrit用户添加.testr.conf 文件  
Python 代码我使用了 testr，需要先安装 testr 命令  
首先切换到root用户，apt-get install python-setuptools，安装 easy\_install命令  
  
[root@115]# easy\_install pip   
[root@115]# pip install testrepository  
在 test-project1 这个项目中添加 .testr.conf 文件  
[root@115]#su - gerrit  
[gerrit@115]$cd test-project1  
  
[gerrit@115]$vim .testr.conf  
[DEFAULT]  
test\_command=OS\_STDOUT\_CAPTURE=1  
OS\_STDERR\_CAPTURE=1  
OS\_TEST\_TIMEOUT=60  
${PYTHON:-python} -m subunit.run discover -t ./ ./ $LISTOPT $IDOPTION  
test\_id\_option=--load-list   
$IDFILEtest\_list\_option=-list  
  
提交到版本库中  
[gerrit@115]$git add .testr.conf  
[gerrit@115]$git commit .testr.conf -m 'add .testr.conf file by gerrit'  
[gerrit@115]$git push origin master  
  
  
  
  
  
  
其他的) 设置 gerrit 用户的 ~/.ssh/config   
文件内容为：  
Host 10.24.6.8:  
IdentityFile ~/.ssh/id\_rsa  
PreferredAuthentications publickey  
  
file://C:\Users\pc304\AppData\Local\Temp\ct_tmp/6.png  
c) 在 gerrit 用户的 ~/.ssh/known\_hosts 中，给 192.168.0.177 (gitlab) 添加 rsa 密钥 ------------------------加密gitlab的ssh时ip信息  
gerrit@gerrit:~/gerrit\_dir$ sh -c "ssh-keyscan -t rsa 192.168.0.177 >> /home/gerrit/.ssh/known\_hosts"  
  
  
gerrit@gerrit:~/gerrit\_dir$ sh -c "ssh-keygen -H -f /home/gerrit/.ssh/known\_hosts"  
  
file://C:\Users\pc304\AppData\Local\Temp\ct_tmp/8.png  
d) 重新启动 Gerrit 服务  
gerrit@gerrit:~/gerrit\_dir$ ./bin/gerrit.sh restart  
  
file://C:\Users\pc304\AppData\Local\Temp\ct_tmp/9.png  
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