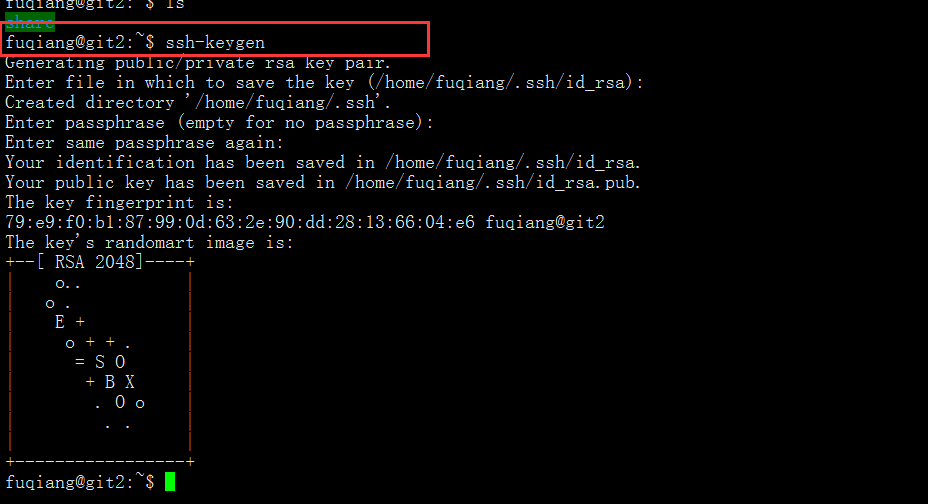
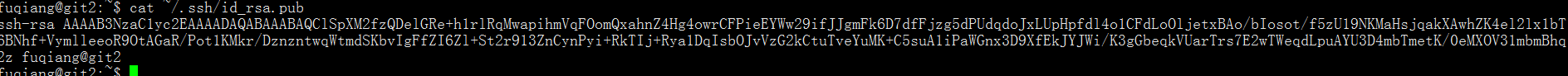
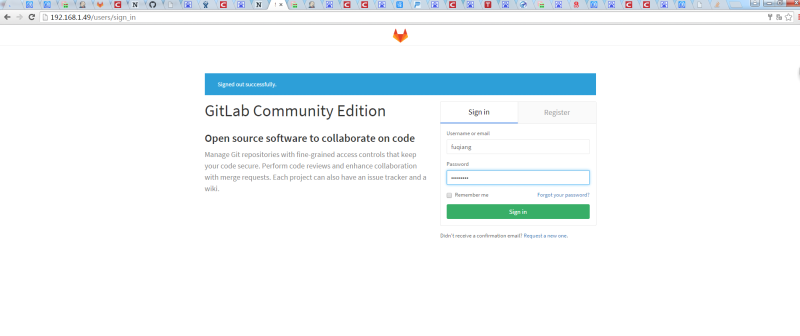
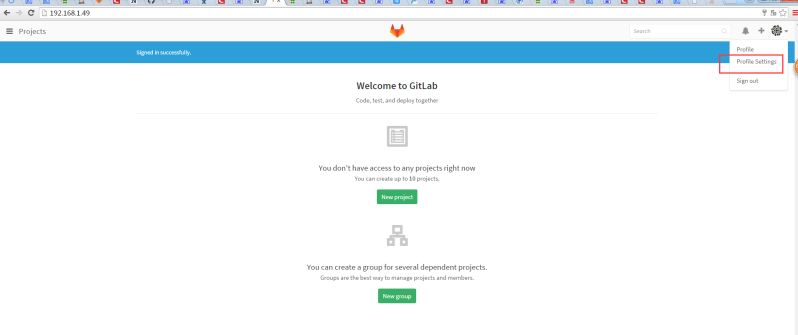
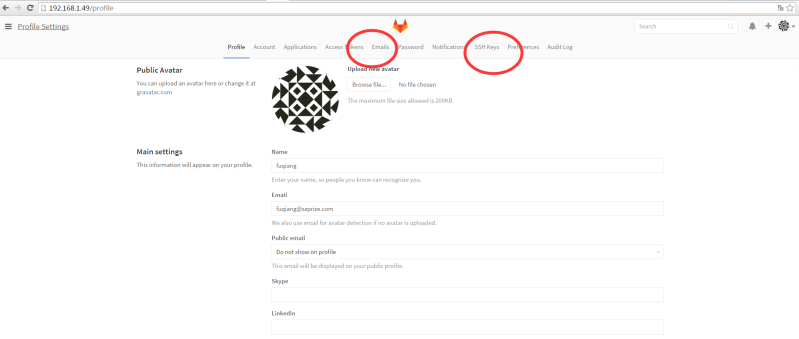
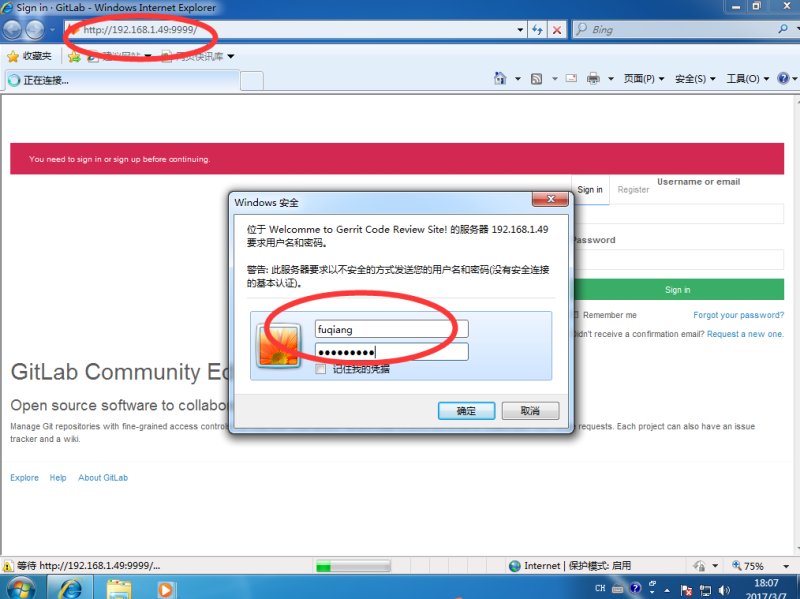
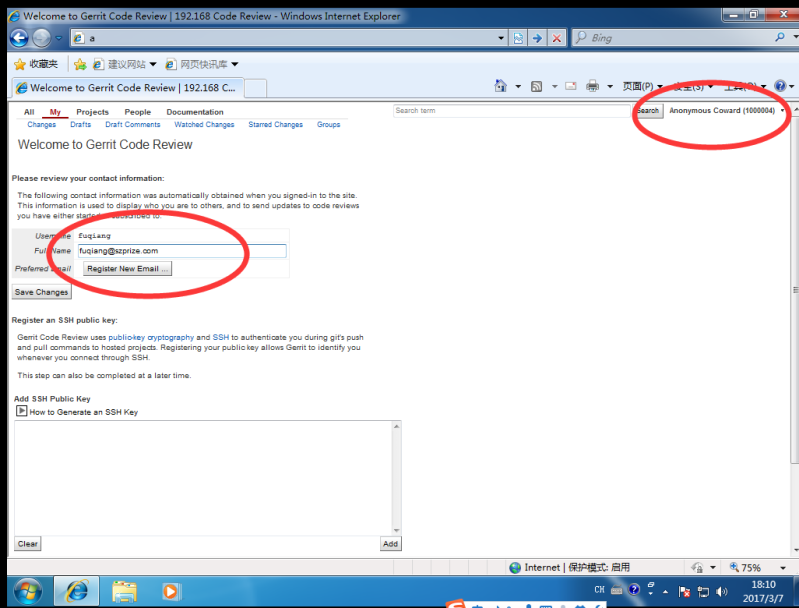
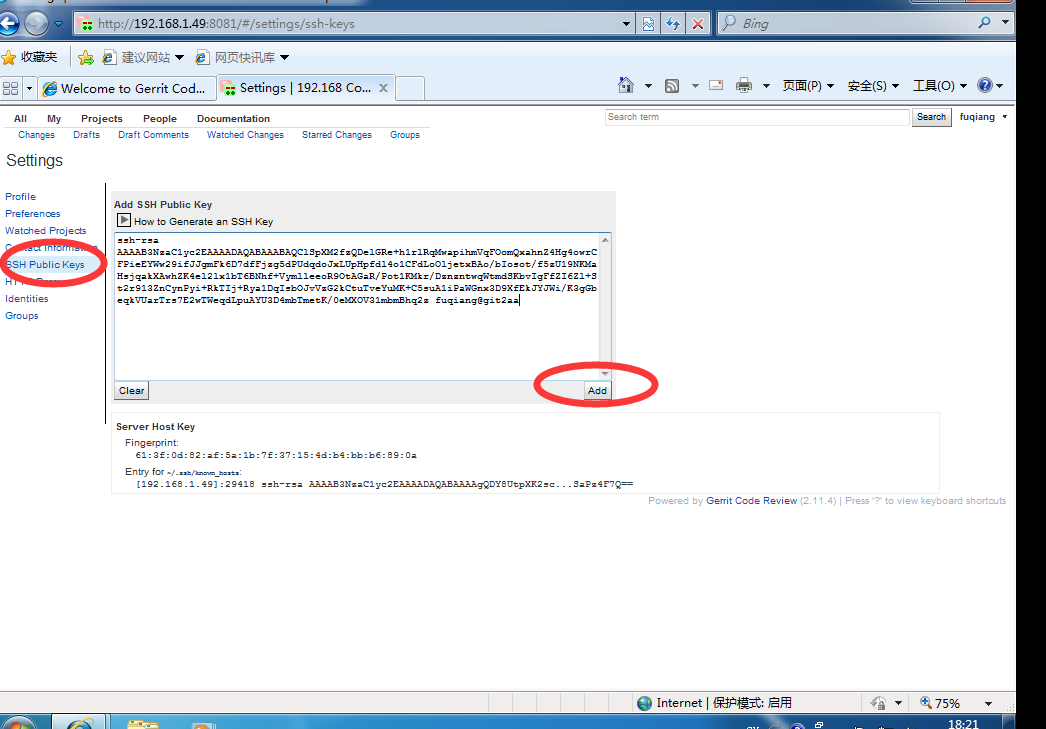
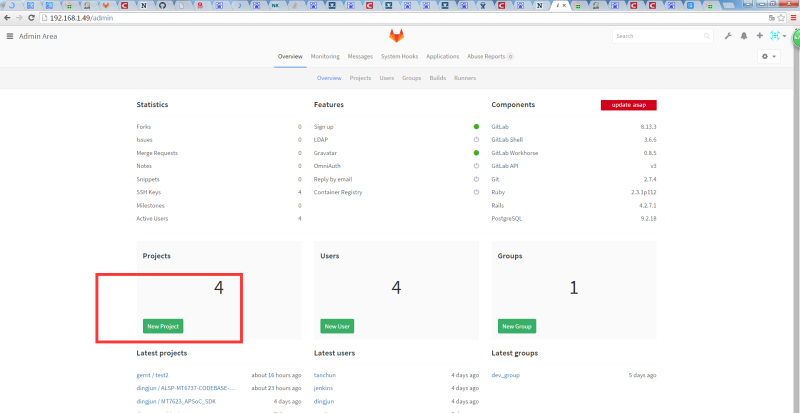
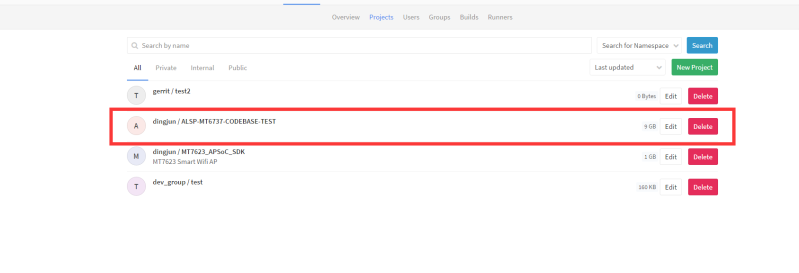
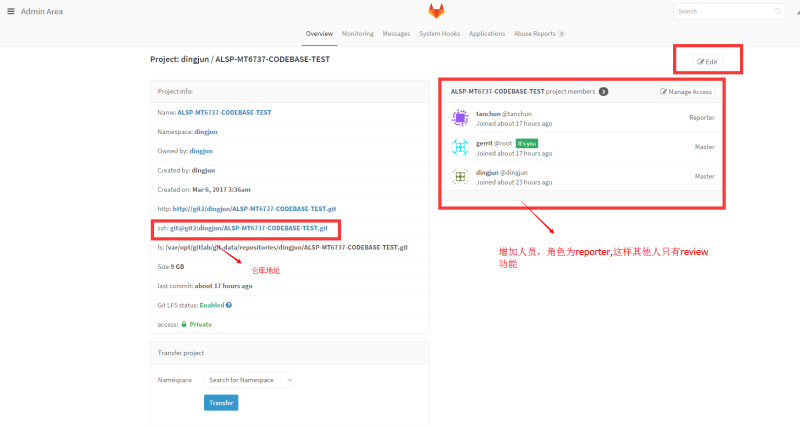
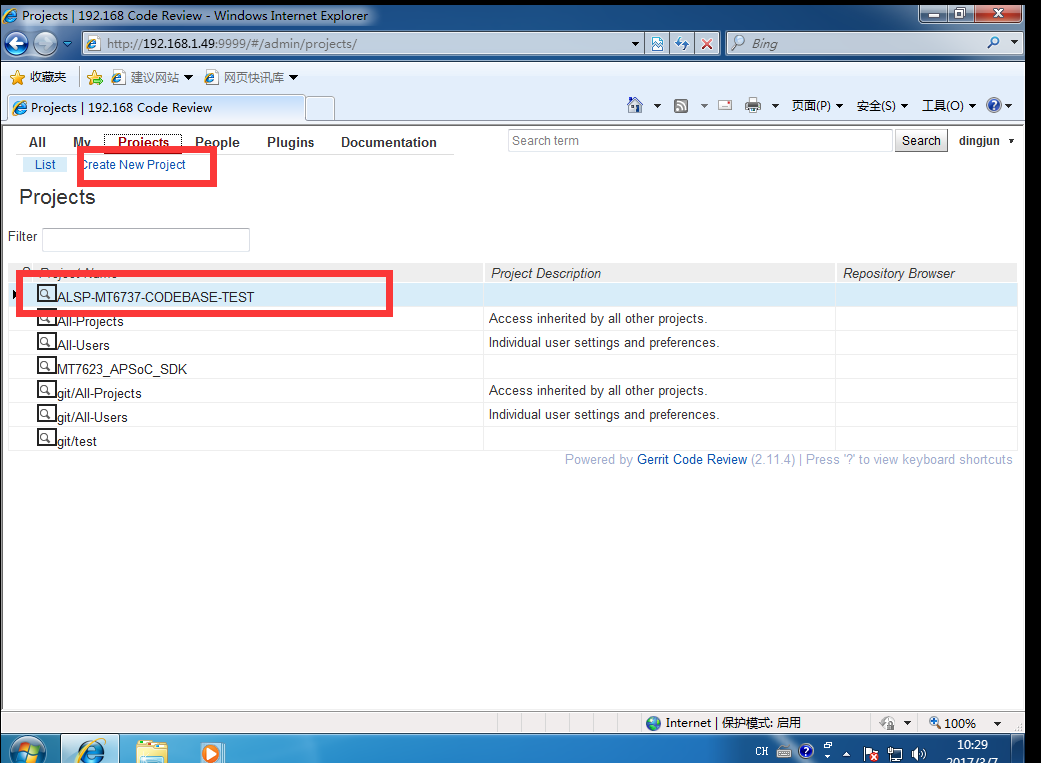
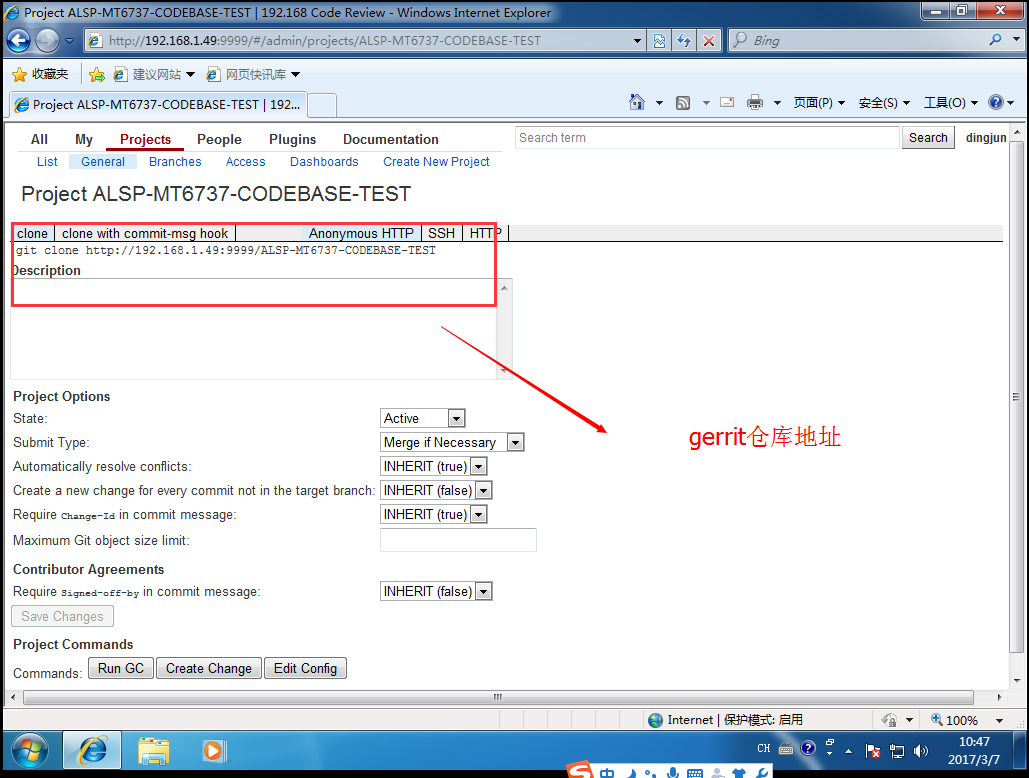
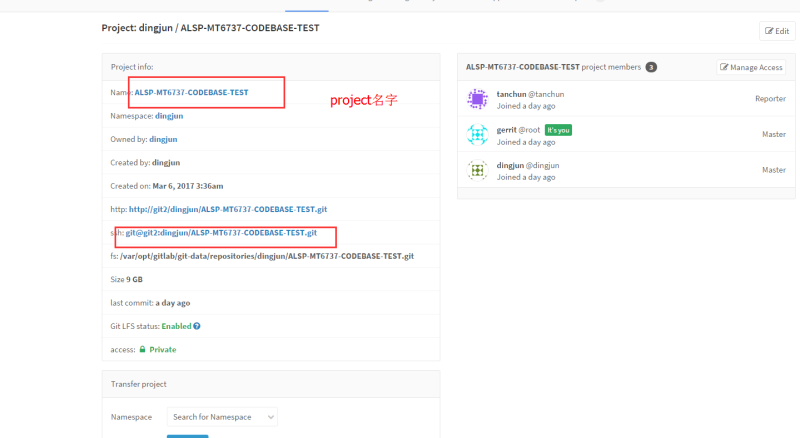
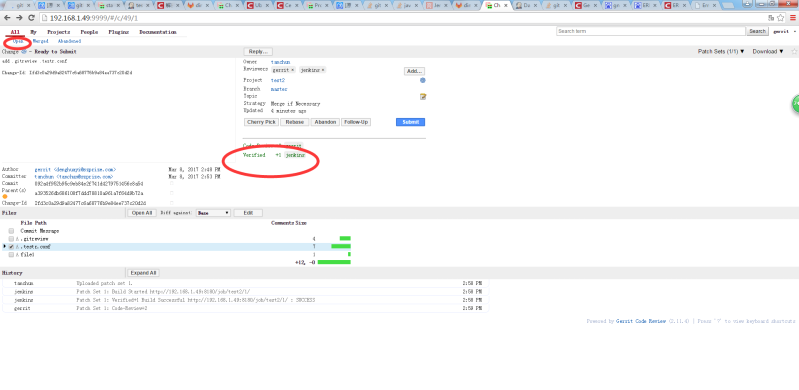
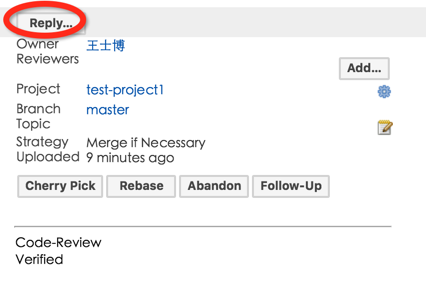
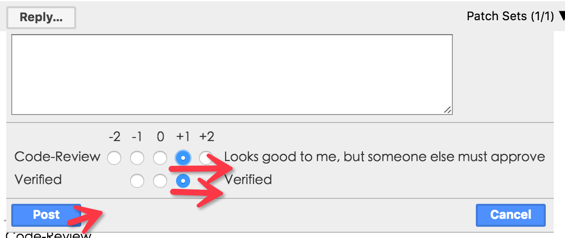
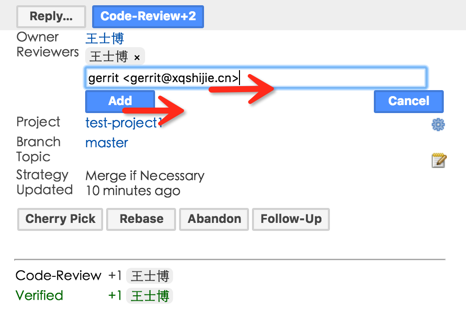
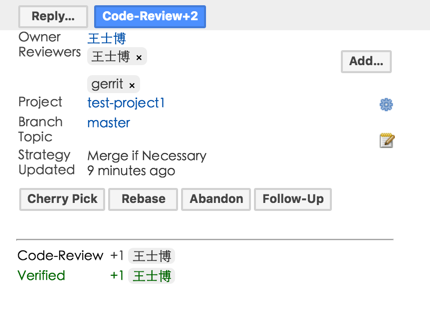
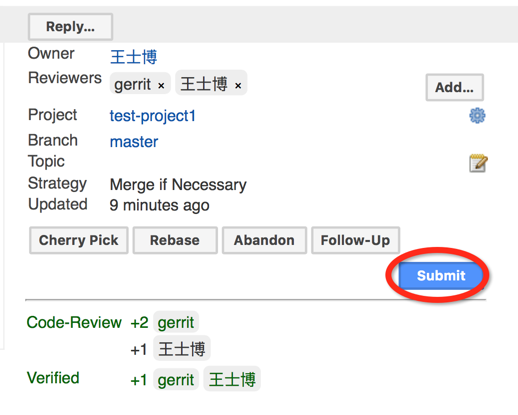
举例:  
  
  
第一步：xshell连接192.168.1.49生成相应的ssh-key,账号为各人的名字，密码为123456，同时可以映射对应的文件夹（方便上传代码）。  
  
  
输入指令：ssh-keygen 直接enter确认两次，执行完毕   
  
  
输入命令：cat ~/.ssh/id\_rsa.pub，得出公钥：并复制   
  
分别打开gitlab页面：http://192.168.1.49 和gerrit页面：http://192.168.1.49:9999  
  
1、gitlab页面：输入用户：fuqiang 密码：123456789，（第一次登录会让你重新设置新的密码再次登录的）  
  
点击右上角的profile setting,开始设置email和ssh-key  
  
  
  
把刚刚得出的公钥填入其中。  
  
  
2、gerrit页面：http://192.168.1.49:9999 输入账号：fuqiang ,密码：123456789 （请注意，你使用的浏览器类型，以后如果收到类似邮件的话，里面的url链接，必须默认以这个浏览器打开）  
  
  
  
初次登录需要注册邮箱，输入你的个人邮箱 ，如fuqiang@szprize.com,点击Regitster New email  
检查个人邮箱，点击里面的链接确认，然后在弹出页面：输入名字，点击save change,会发现右上角的anonmous变成fuqiang了，然后注册完毕  
  
点击左侧的sshkey，把前面的公钥，增加进去  
  
  
  
  
  
第二步：创建项目，以dingjun用户为例子  
  
一、dingjun用户先后分别在gitlab页面：http://192.168.1.49和gerrit页面：http:192.168.1.49:9999登录进去，创建new project(注意，在gerrit界面上，首先需要超管理员把dingjun加入到adminstrator组才有组建project的权限 )  
  
1、gitlab:页面操作：  
  
创建成功，点击edit,增加其他组员，选择他们的角色为reporter  
  
记住：gitlab仓库地址：git@git2:dingjun/ALSP-MT6737-CODEBASE-TEST.git  
project名： ALSP-MT6737-CODEBASE-TEST  
  
  
2、gerrit页面操作：  
  
点击create new projec, 填入project名： ALSP-MT6737-CODEBASE-TEST ，创建。  
  
  
记下仓库地址：git clone http://192.168.1.49:9999/ALSP-MT6737-CODEBASE-TEST  
  
  
  
二、同步数据。  
创建者在现有的项目的根目录里面多增加两个文件.gitreview和.**.testr.conf** ,内容如下：  
  
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192.168.1.49上）  
gerrit@gerrit:~/cfg/cfg/test$ vim .gitreview  
文件内容为：  
[gerrit]  
host=192.168.1.49   
port=29418  
project=你在gitlab上面新建的project名  
  
vim **.testr.conf** 内容如下：  
[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 $IDFILE  
test\_list\_option=-list  
  
  
  
并push到gitlab服务器上面： gitlab 地址[t2:dingjun/ALSP-MT6737-CODEBASE-TEST.git](http://192.168.1.49/dingjun/ALSP-MT6737-CODEBASE-TEST)  
  
  
push完成后，通知服务器管理人员进行gerrit服务器的数据同步。  
  
  
三、其他开发人员从gitlab地址，克隆代码下来进行操作  
  
命令行输入：  
git clone git@git2:dev\_group/test.git  
  
cd /test  
git config --global user.name tanchun  
git config --global user.email tanchun@szprize.com  
gitdir=$(git rev-parse --git-dir) ------这两步会加入ID  
scp -p -P 29418 tanchun@192.168.1.49:hooks/commit-msg ${gitdir}/hooks/ ------这两步会加入id  
然后操作其他动作，如 touch file 1等  
git add .  
git commit --amend -----注意一定要使用commit --amend  
  
  
  
git review  
  
  
  
四、开发人员git review功能提交后，登录gerrit页面:http://192.168.1.49:9999,输入用户跟密码，点击open会发现有提交的消息，当jenkins测试后，会在下面出现verified +1 jenkins  
  
  
  
开发本人进行两步操作：  
第一步：点击reply,进行代码评分，+1  
第二步：点击 reviewer那一栏，选择你要增加的审核的人员，然后点击add增加人员  
  
  
  
通过这步操作，要审核的人员会收到相关的邮件。  
  
五、  
  
  
三、review功能  
  
1）上传代码者（自己先verified核实，然后通知审核者审核）  
修改代码，验证后提交到 Gerrit 上。  
代码提交后登陆 Gerrit，自己检查代码（重点看缩进格式跟原文件是否一致；去掉红色空格部分；修改内容是否正确；命名是否有意义；注释内容是否符合要求等）。  
自己检查没问题后，点 “Reply”按钮，在“Verified”中 ＋1，在“Code Review”中 +1，并点“Post“  
在”Reviewer”栏中，点击”Add"添加审核者 [如果不添加审核者，上传者自己也可以审核并完成提交。注意：只有Review是＋2的时候，才能出现submit的提交按钮]  
如果代码审核没有通过，请重复步骤1，2，3。  
流程截图：  
代码提交后，上传者自行登陆gerrit，找到提交的subject，点击"Reply"  
  
   
  
  
  
2）审核者  
收到邮件通知后登陆 Gerrit，审核代码。  
如果审核通过，点 “Reply”按钮，在“Verified”中 ＋1，在“Code Review”中 +2，并点“Post”，最后点击“Submit“提交！  
如果代码审核没有通过，点 “Review”按钮，在“Code Review”中 -2，写好评论后，点“Post”。  
流程截图：  
如上，subject的owner添加审核者后，审核者登陆gerrit进行review  
点击“Reply"  
  
   
  
  
这样，就完成了一个代码的审核全部过程！  
登陆gitlab，就会发现gerrit上审核通过并提交后的代码已经同步过来了！