**[Gerrit使用简介](https://www.cnblogs.com/lexuele/p/5133601.html)**

**1 Gerrit简介**

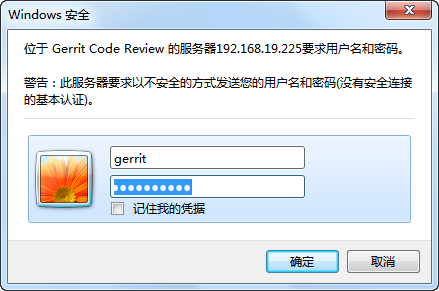
    Gerrit，一种免费、开放源代码的代码审查软件，使用网页界面。利用网页浏览器，同一个团队的软件程序员，可以相互审阅彼此修改后的程序代码，决定是否能够提交，退回或者继续修改。

**1 注册Gerrit**

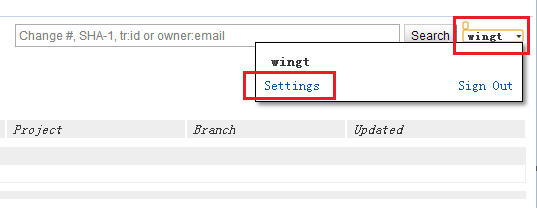
**1.1 账号密码**

    注册Gerrit需给Gerrit管理员（yangjunyu@wingtech.com）发送邮件获取账号密码。

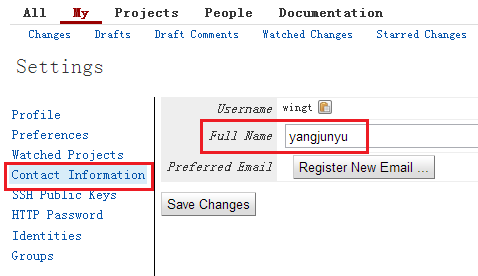
    从管理员处获取账号密码后，登录Gerrit：http://192.168.42.21:8083/



**1.1 修改名字和邮箱**



如邮箱为xxxx@google.com，则“Full Name”填写为xxxx：



注册邮箱：

注册成功后会给邮箱发送一封邮件，点击链接进行验证（如果某些浏览器无法打开验证链接，复制到IE上打开）：

**1.1 添加用户**

linux下的用户名须和管理员给的Gerrit账号名相同，如果没有，请添加新用户：

root@BuildHost:~# adduser gerrit

https://images2015.cnblogs.com/blog/727709/201601/727709-20160115155850710-1162132700.png

**1.1 添加ssh密钥**

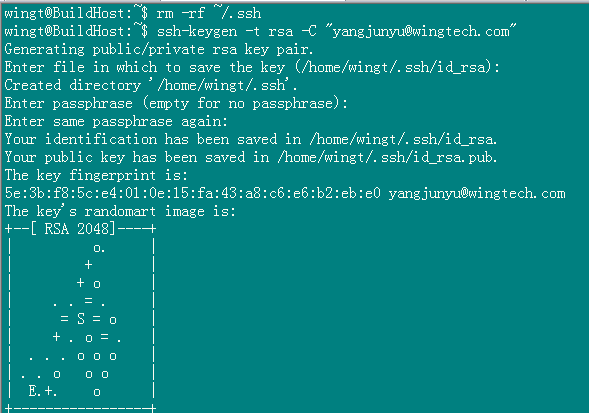
备份.ssh（如果存在的话）：

https://images2015.cnblogs.com/blog/727709/201601/727709-20160115155923944-945757811.png

生成密钥：

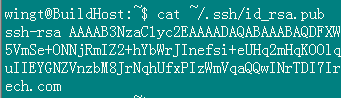
rm -rf ~/.ssh

ssh-keygen -t rsa -C "自己邮箱地址"   ###一直按回车键即可

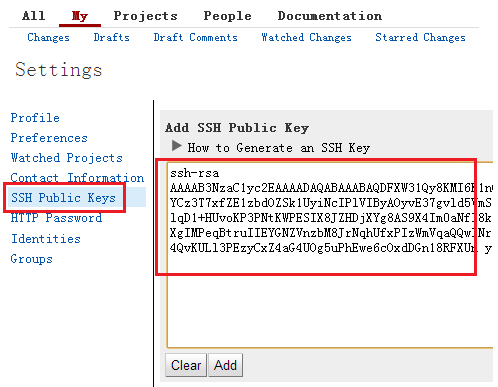


然后公钥添加到Gerrit：

把 ~/.ssh/id\_rsa.pub 内容复制好



粘贴到



**1 开发人员使用**

**1.1 下载代码**

正确配置好邮箱和ssh公钥后，根据Gerrit管理员给出的地址下载代码：

如：git clone ssh://用户名@192.168.42.21:29418/test

**1.1 下载commit-msg脚本**

【重要：使用Gerrit必须要有此脚本】

1. 拷贝脚本到本地

scp -P 29418 -p 192.168.42.21:/hooks/commit-msg .

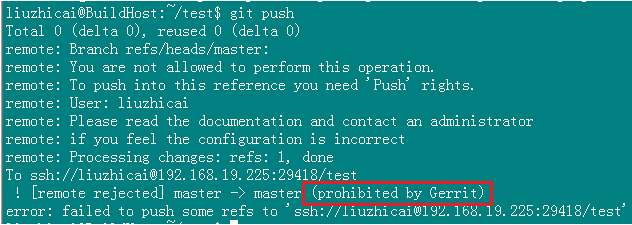
2. 把脚本拷贝到git钩子路径

cp commit-msg /usr/share/git-core/templates/hooks

chmod +x /usr/share/git-core/templates/hooks/commit-msg

**1.1 提交代码审核**

Gerrit不允许用户直接push代码到分支上（特别授权用户除外），否则报错：

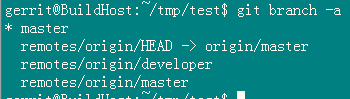


正确的做法是所有提交都push到 HEAD:refs/for/分支名 ：

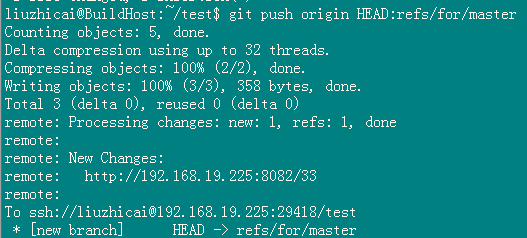
格式如：git push origin HEAD:refs/for/分支名

例1：

当前处于本地分支master

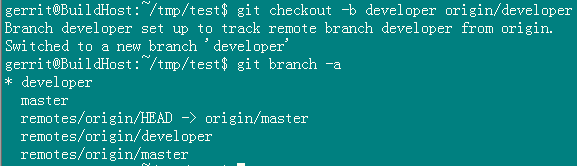


代码提交到远程分支master上，则：git push origin HEAD:refs/for/master



例2：

当前处于本地分支developer



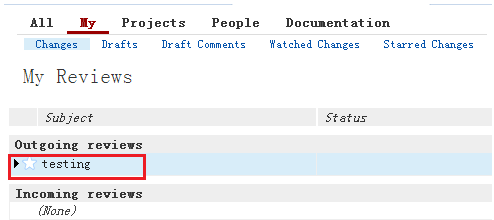
代码提交到远程分支deleloper上，则： git push origin HEAD:refs/for/developer

https://images2015.cnblogs.com/blog/727709/201601/727709-20160115160400569-389811362.png

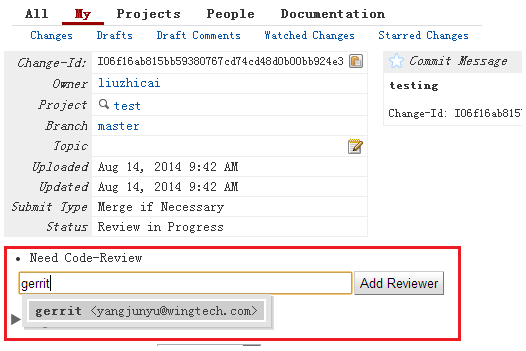
**1.1 添加审核者**

代码正确提交到分支 refs/for/\* 后，需要登录Gerrit界面为刚才的提交添加相应的审核者：

登录Gerrit：http://192.168.19.225:8083/， 在“changes”下可以看到刚刚的提交，名称显示为刚刚提交所写的注释



点击进入，在“Need Code-Review”下添加相应的代码审核者（请根据自己项目情况添加指定的审核者，此处仅作演示）



如果审核通过，会收到提示邮件（Code-Review为正数，说明审核通过，但代码未合入分支）：

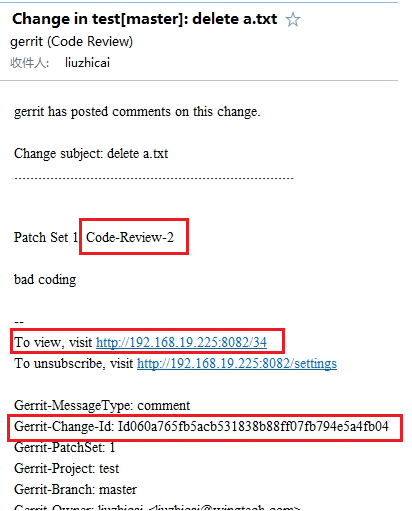


代码被合入后，会收到邮件提示（说明代码已经正确提合入到分支）：



**1.1 审核失败再提交**

当收到审核失败的邮件时（Code-Review为负数，需要记下Change-Id，下一步提交时会用上），需要修改代码重新提交：

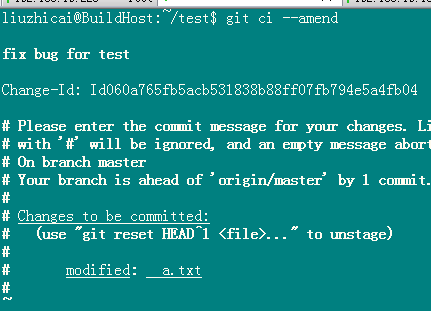


修改好代码后，重新提交代码：

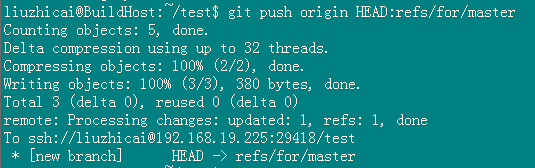
【重要：代码审核失败后，重新提交时必须使用 git commit --amend，否则会生成新的评审任务，无法与上一次审核任务关联】

https://images2015.cnblogs.com/blog/727709/201601/727709-20160115161122272-30072014.png

【重要：注释可以重新写，但是其中“Change-Id”必须为上一次审核任务失败的Change-Id，可在Gerrit或者上一步的邮件中查看得到】



再push代码到Gerrit：

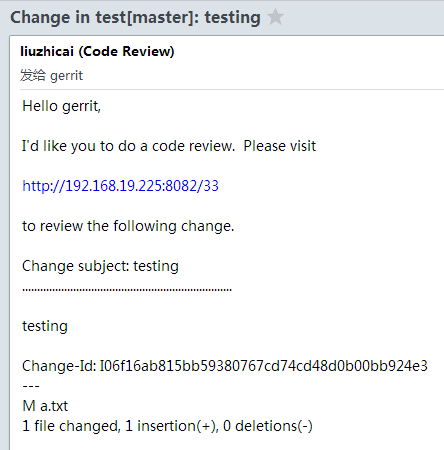


最后无需重复步骤“3.3 提交代码审核”，Gerrit会自动给原来的审核者发送相应邮件

**4 审核人员使用**

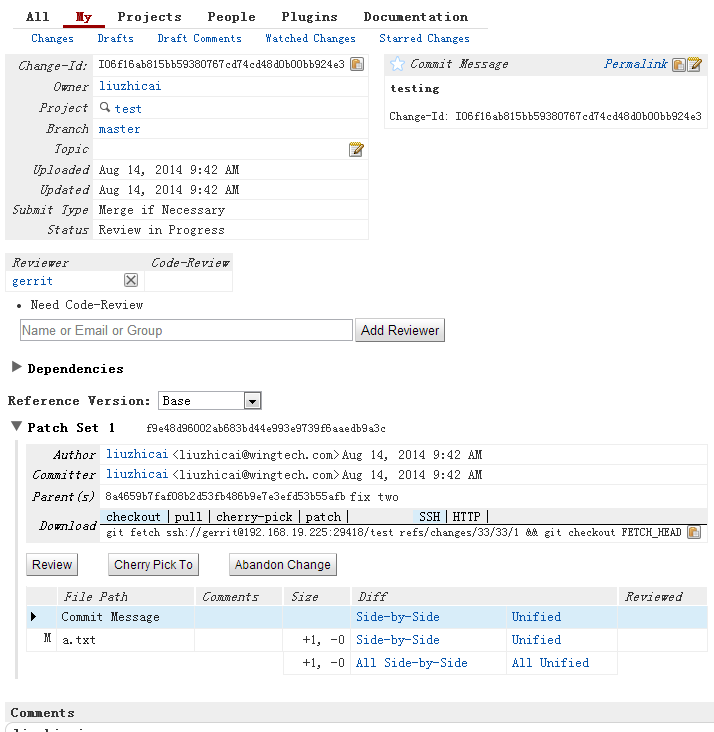
**4.1 收到审核邮件**

当有审核任务邮件时，点击链接打开Gerrit



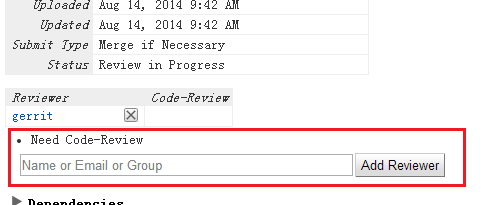
**4.2 查阅审核任务**

打开审核任务如下：



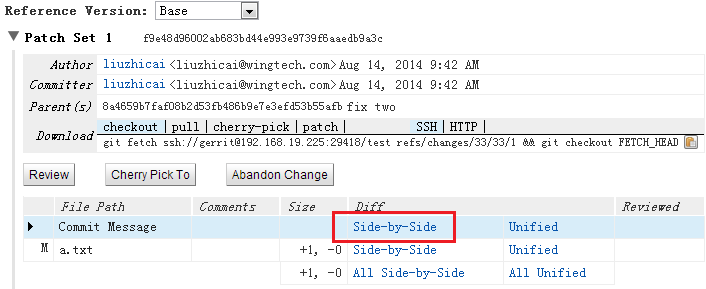
**4.3 添加其他审核者**

如果需要，这里可以添加其他人员审核（需要有审核权限）

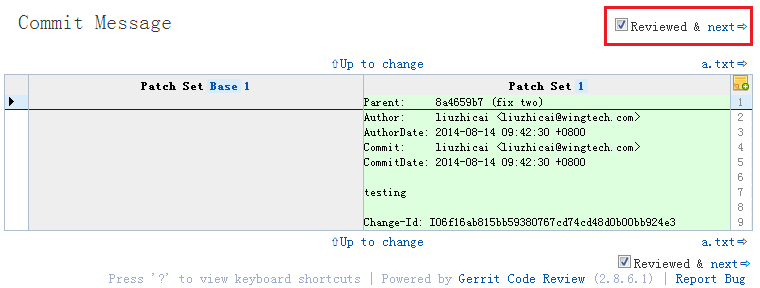


**4.4 查看修改内容**

点击“Side-by-Side”按钮，查看修改内容：

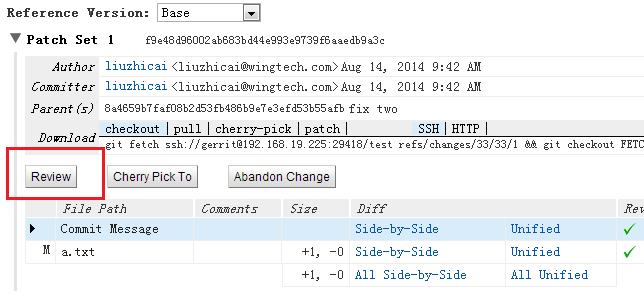


点击“next”查阅下一项修改：



**4.5 审核通过/失败**

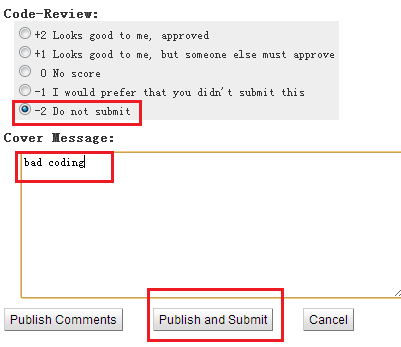
点击“Review”



审核通过：



审核失败：



代码提示被blocked：



审核失败的任务会给提交者发送邮件，通知其修改

**5 其他**

**5.1 常用别名设置**

git config --global alias.st status

git config --global alias.ci commit

git config --global alias.co checkout

git config --global alias.br  branch

git config --global color.ui true

git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset %C(bold blue)<%an>%Creset ---%C(yellow)%d%Creset %s %Cgreen(%cr)' --abbrev-commit"

