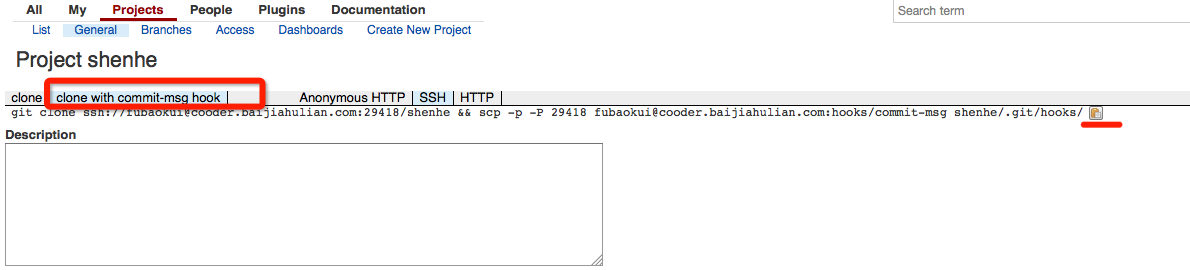
## 一、关联Git到Gerrit

直接clone ：

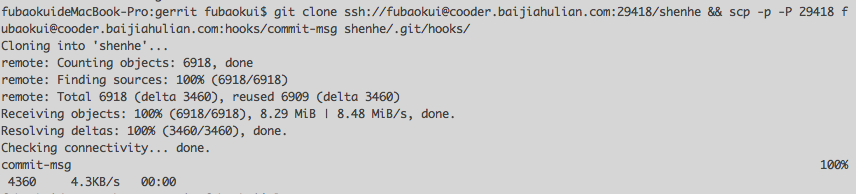
1.登录Gerrit

2.设置SSH  Public Keys。

3.进入要clone的项目，点击“clone with commit-msg hook”，copy下面出现的git命令到本地执行就可以了。（点击右侧“复制”按钮，就可以复制git命令）



复制完成如下：（以shenhe为例）



## 二、提交代码审核

1.git add xxx

2.git commit -m ‘comments'

3.提交代码到dev分支给xxx审核的申请：git push origin HEAD:refs/for/master%r=xxx,cc=ssss

     注：

     HEAD ：指向当前本地的git branch。如果当前的分支为“dev” ,提交审核时可以用”dev”替换”HEAD”.

     refs/for/<branch-name> ： “refs/for/“为Gerrit约定的提交代码审核申请的前缀。

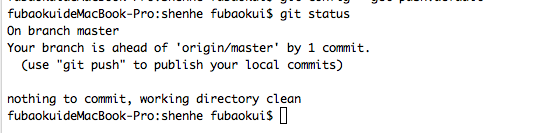
     r=xxx : 其中“r”为reviewer参数,后跟审核人得name，多个审核人员的话需要添加多个“r”参数。(不用添加公司的邮箱后缀)

     cc,抄送邮件对象。

      eg: git push origin HEAD:refs/for/dev%r=xxx,r=aaaaa

     成功提交审核申请后，会触发Gerrit的审核流程，在Gerrit中生成Change（审核记录），给审核人发邮件通知有审核请求。

      当前的状态:



4.

（1）审核通过：执行git pull orign <branch-name>，更新同步远程代码库。

（2）审核不通过

     a.重新修改代码

     b.git add xxx;

     c.git commit —amend (可以重新编辑注释，但最下面的Change-Id不能改动)

     d.再次进入审核流程。

3.refs/tags/\* 开放 “create reference”权限即可。不用审

## 三、提交代码审核的快捷命令配置

1.方式一：修改配置文件.git/config

[remote “dev\_fyx"]

  url = ssh://fubaokui@cooder.baijiahulian.com:29418/shenhe

  push = HEAD:refs/for/dev%r=fanyixin,cc=fubaokui

$ git push dev\_fyx

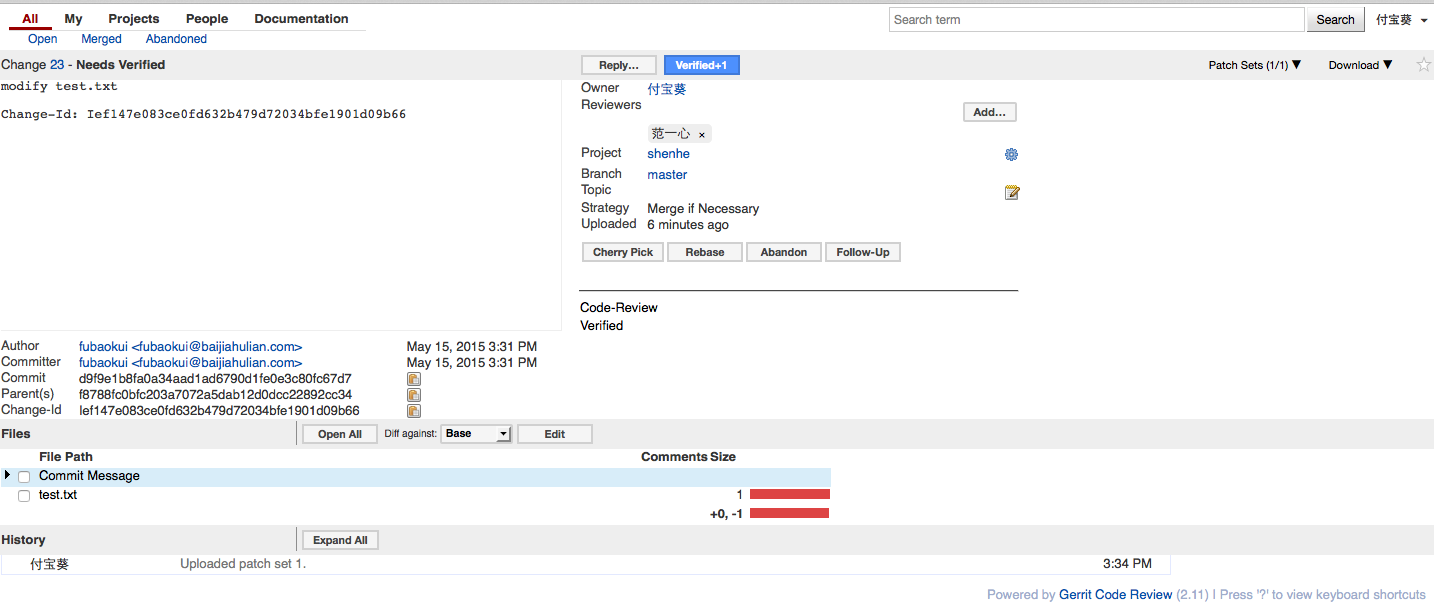
2.方式二：设置global变量

     $  git config --global alias.push-for-review "push origin HEAD:refs/for/master%r=fanyixn,cc=fubaokui"

     $ git push-for-review

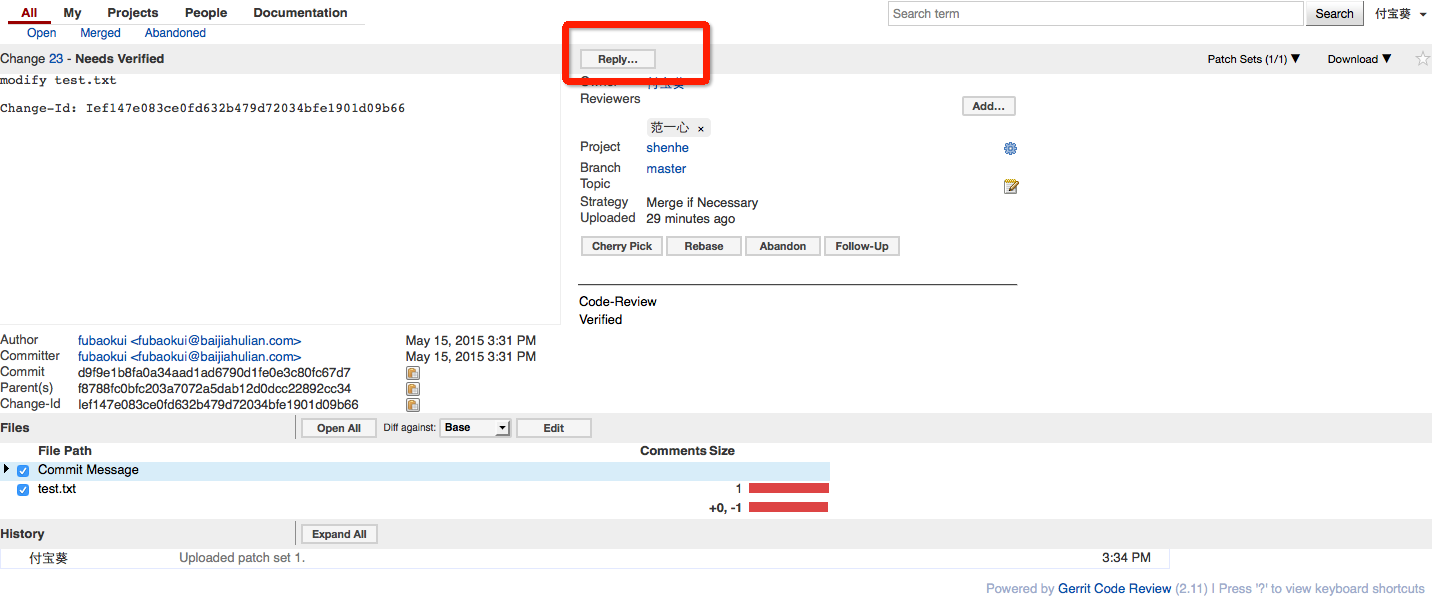
## 四、审核代码流程

1.收到审核请求的通知邮件后，通过邮件的里面的链接，跳转到Gerrit的审核页面；



2.在页面右下角审核修改的文件 ，审核完成后打分。

3.点击“Reply”，进行打分。需要打分项有：Code-Review和Verified



（1）审核通过：在“Verified”中 ＋1，在“Code Review”中 +2，并点“Publish”。之后会出现“submit”按钮，点击就可以把当前patch同步到gerrit管理的代码库中。最后Gerrit自己会把代码再push到Gitlab上。

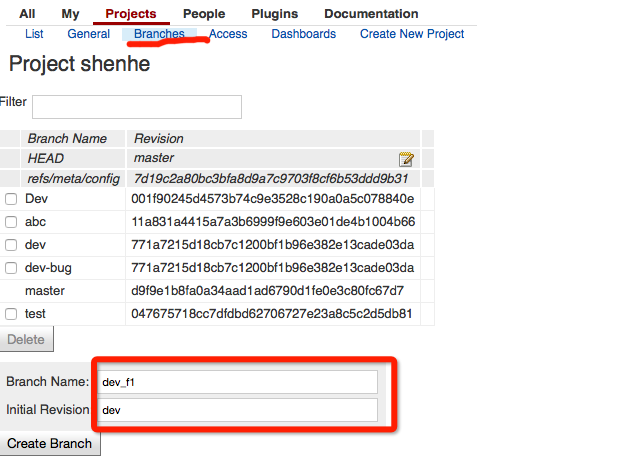
（2）审核不通过：在“Code Review”中 -2，写好评论后，点“Publish”即可。

## 五、New branch的方式

new branch的方式一、web界面

1.登录Gerrit，新建分支。

如图，基于dev新建分支dev\_f1.



2.在本地执行：

git checkout -b dev\_f1

git pull origin dev\_f1

二，命令行

 语法：ssh -p <port> <host> gerrit create-branch <project-name> <new-branch-name> <base-branch>

     eg:

          git 命令创建分支：（在shenhe下，新建基于dev的新分支mydev）

          $ ssh -p 29418 cooder.baijiahulian.com gerrit create-branch shenhe mydev dev

git打tag

git checkout -b branchname origin/xxx

git tag -a tagname -m '说明'

git push origin tagname HEAD:refs/heads/branchname

merge的时候，如果没有生成change-id,需要使用git reset –soft commitId

审核不通过的话，

1：git log 里面最新的commit

再次修改完代码后，在提交的时候执行“git commit --amend”就行了。这个是把当前的修改合并到之前的提交。

2：git log 里面不是最新的commit