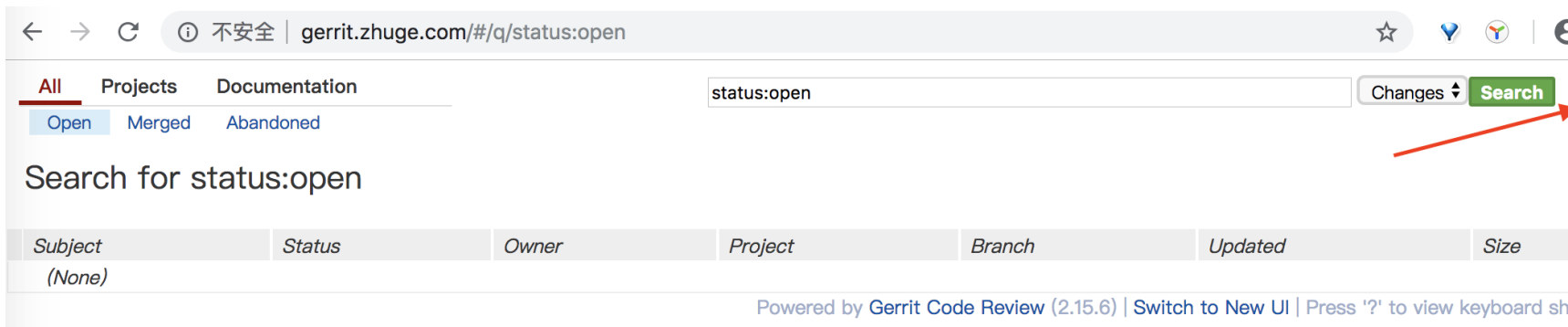
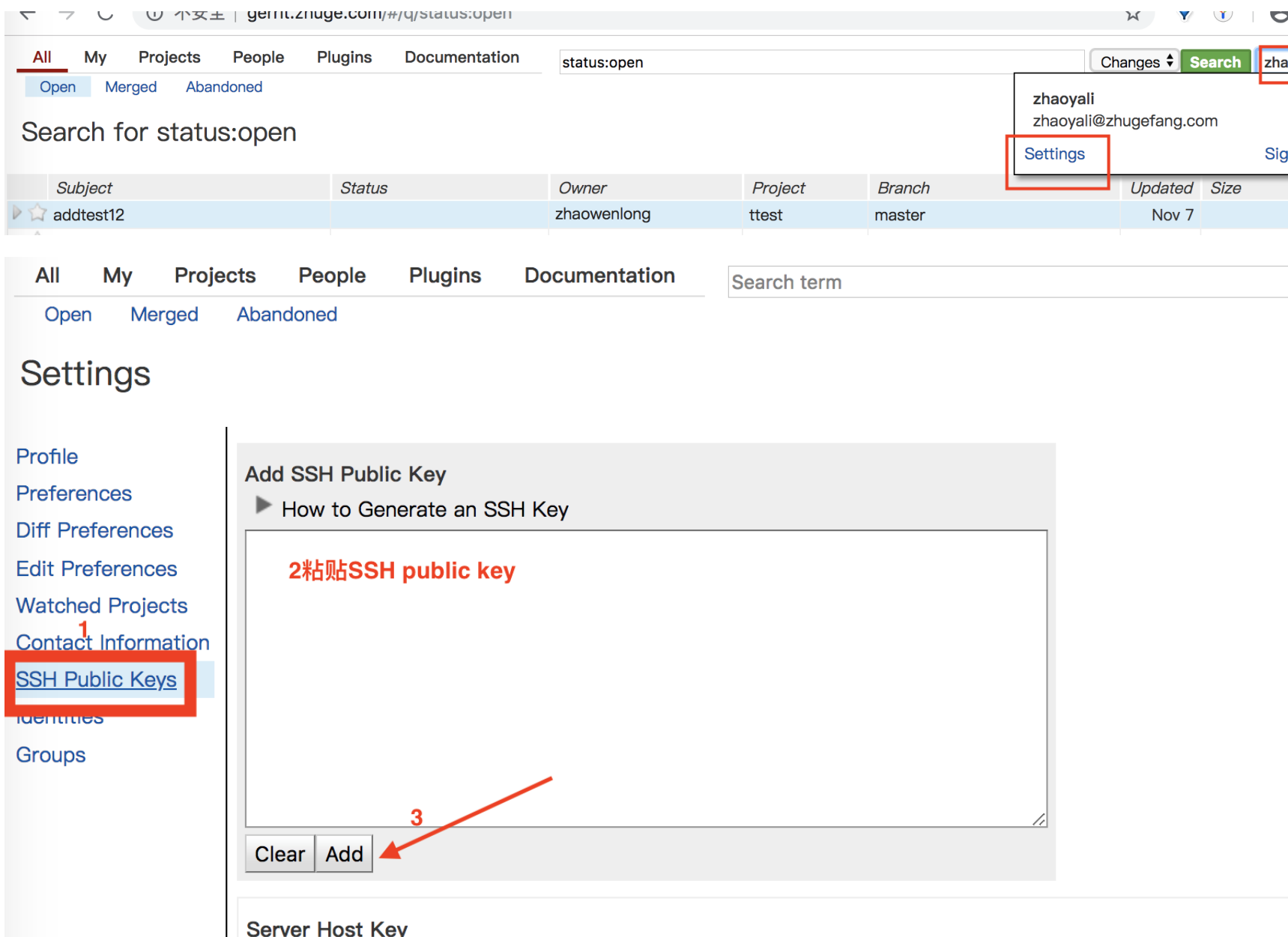


## 1、配置SSH key

i) 打开<http://gerrit.zhuge.com>网页,使用邮箱(不加后缀)和密码登录

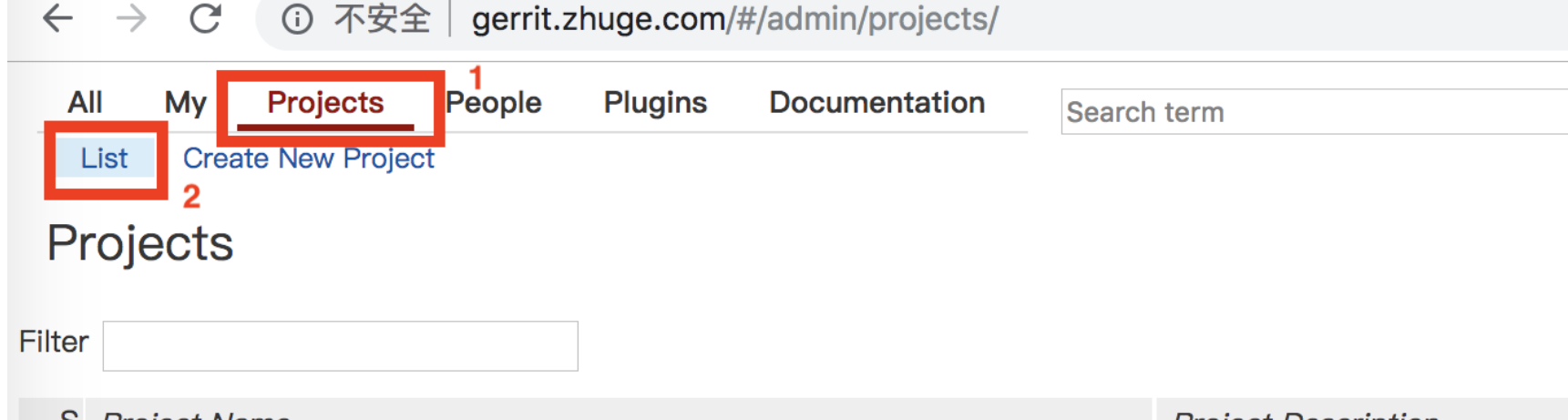


ii) 打开设置页面-SSH Public Keys,添加SSH Public Key



SSH Public Key的添加跟gitLab中key添加类似,可以参考!点击“add”按钮后会报504错误,点击continue刷新页面就添加成功了.

## 2、下载工程项目,projects→list,选择要下载的项目



### 3、下载提交代码的hook脚本

在仓库目录下执行命令: `scp -p -P 29418 用户名@gerrit.zhuge.com:hooks/commit-msg .git/hooks/` 注意替换红色部分的内容

执行成功的提示如下:

```
→ ttest git:(master) scp -p -P 29418 zhaoyali@gerrit.zhuge.com:hooks/commit-  
.git/hooks/  
commit-msg 100% 4780 799.0KB/s 00:00
```

### 4、提交review的流程

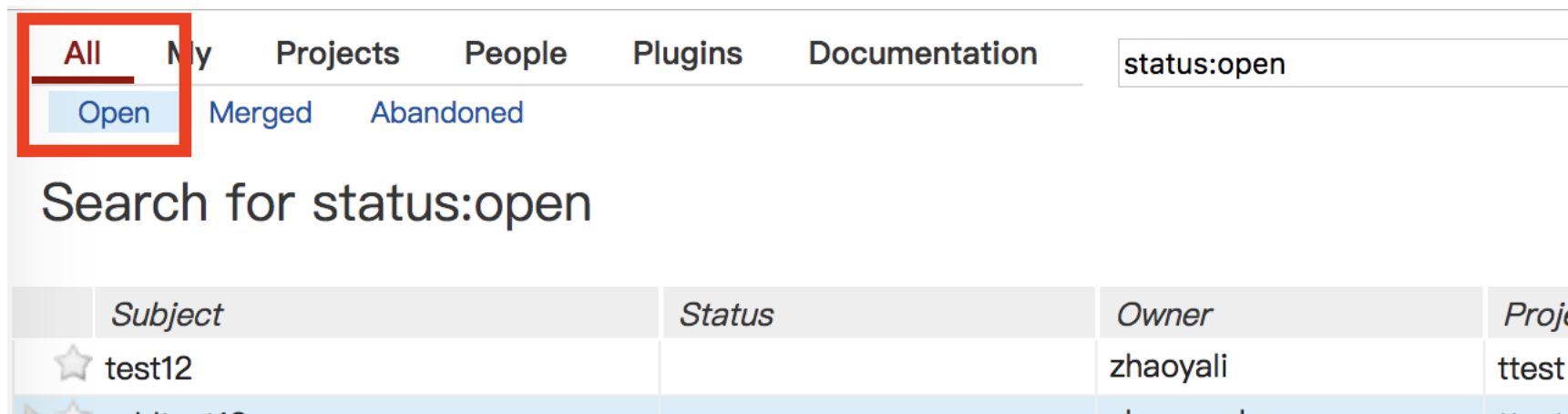
修改完本地代码后执行

1)git add.

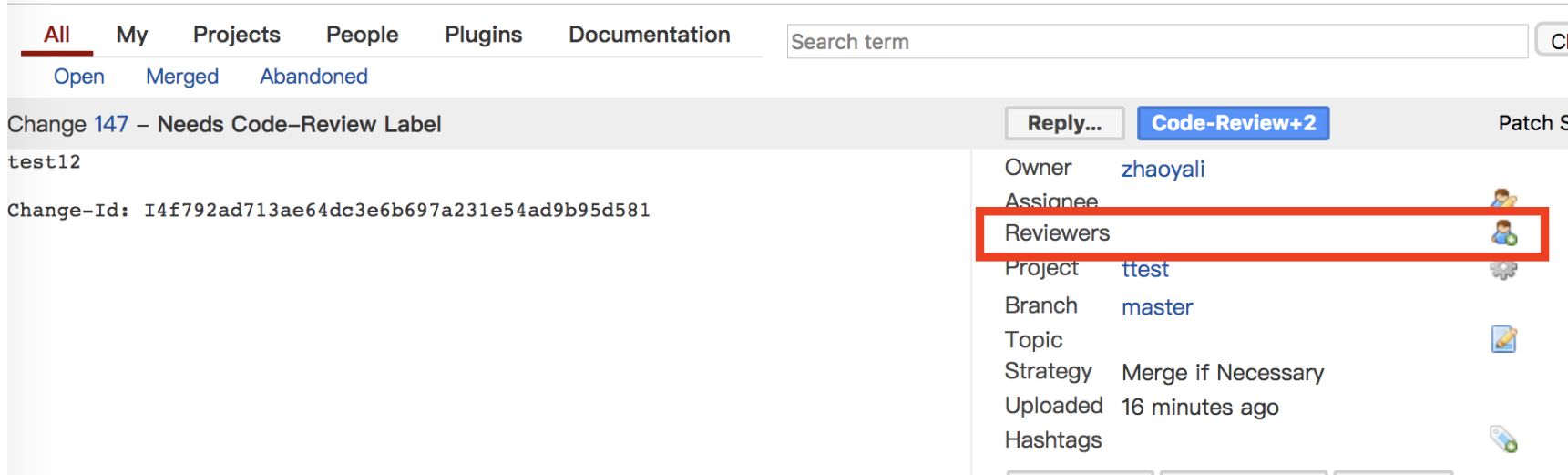
2)git commit

3)git push origin **branchName**:refs/for/**branchName** //注意替换红色部分的内容第一个位本地分支名,第二个位远端分支名;我们实行强制review机制,直接执行git push会报错

4)打开<http://gerrit.zhuge.com>,到ALL→Open列表中找到刚才提交的codeReview,打开后添加reviewer



Open:等待review的提交 Merged:审核通过已经入库的提交记录 Abandoned:取消的review



添加其他人帮你review代码

为了简化3、4步骤,同时避免分支名字写错的情况,可以使用以下脚本:



[review.sh](#)

- 1) 下载本脚本
- 2) 添加到git工程项目目录下
- 3) 修改脚本中62行,添加合适的reviewer
- 4) 想要提交review时执行命令 `sh review.sh` 即可.

## 5、review别人代码

- 1) 打开<http://gerrit.zhuge.com>,到ALL→Open列表中找到要review的提交  
红框中时本次提交修改的文件,点击可查看详细的修改行数

选中有问题的代码块,点击弹出的黄色小框“press C to comment”即可写出原因或改进意见,点击“Save”保存意见.

最后给本次提交打分,只有本次提交被+2之后才可以入库.

-1说明代码有问题,需要提交者修改后再次提交,再次审核,直到+2,点击submit入库

## 6、修改被打回的代码

修改好被打回的提交后,执行命令

```
git add .
```

```
git commit --amend //注意与第一次提交代码的区别,保证不会产生新的commitID和changeID  
然使用上次的
```

```
git push origin branchName:refs/for/branchName //如果下载了review脚本执行命令:sh review
```

说明:修改被带回提交时要保证修改的内容和开始创建review的提交在同一个提交里.

## 7、+2的代码不能submit

提交的review明明已经+2但submit按钮还是不能用

红色字“Cannot Merge”已经说明了问题,本次补丁的代码和一入库的代码合并会有冲突,需要先更新代码解决冲突

执行命令git pull --**rebase**,解决冲突之后执行git add. git rebase --continue,然后需要在重新提交一次review

这里要使用--rebase操作,不能直接merge;原因在下面的merge和rebase的区别部分有详细解释

# Gerrit原理分析和操作注意事项

## 1、Gerrit原理

由图可见 workflow:提交review请求创建补丁→code review并标记+2 → submit →同步gitLab

### 1、提review创建补丁

每次review请求会创建一个补丁,并用changeID作为补丁的唯一标识,会记录修改的代码、提交者、**本次修改基于的父节点commitID**等内容(具体样子[http://gerrit.zhuge.com/#/c/ios\\_zgzf/+/](http://gerrit.zhuge.com/#/c/ios_zgzf/+/))

changeID的产生是在每次做git commit操作是通过钩子脚本实现的,上面步骤中下载commit-m脚本到.git/hooks/的操作就是在下载钩子脚本

## 2、reviewer检查代码标记+2

reviewer也是通过changeID获取一次review的修改内容,检查无误给补丁加2标记说明本次补丁到可入库状态,有些代码规范不太重要的问题做-1标记并备注说明,有严重逻辑问题做-2并备注

## 3、submit本次补丁到gerrit服务器仓库

补丁入库的操作是[查找到parentID在库中的位置](#),在此节点基础上把修改的内容写入到仓库中  
所以只有达到+2状态的补丁才可以提交入库,补丁的父节点在仓库只存在才可以提交

## 4、gerrit仓库由更新后服务器会自动同步到gitLab仓库

正常流程之外的情况:

### 1、更新已提交的补丁

提交被reviewer打回,自己发现问题后漏提交文件等都需修改,可以通过两种方式

a)gerrit上[Abandon](#)本次补丁,本地代码回滚reset --soft或--hard,重新修改提交

b)直接修改代码,git add. git commit--[amend](#),重新提交执行review.sh脚本;使用--amend是修改内容,changeID不变,但commitID会改变

### 2、累计多次提交

本地积攒多次commit,最后把整个功能开发完之后在提交review,这样的操作会让你抓狂,最好不要积攒,有一次就提交一次

a.多次review中但凡当中一个被打回,就要修改本次补丁,git commit --amend会使补丁的commitID发生改变,changeID不变,

b.在修改的补丁后面的提交记录的parentID不会随之更新

b.补丁入库的操作使先找到补丁中标记的parentID

这样就会导致修改补丁后面的补丁即使+2也无法入库,所以最好使一次提交就提一次review,入之后在提第二次的review

## 常见问题

### 1、clone第二个工程,提交review失败,提示如下

需要执行上面的“3、下载提交代码的hook脚本”，

## 2、如果习惯使用source tree的可以在source中添加自定义的action

source tree自定义动作的入口:菜单栏 动作→自定义操作

提交review时候,在source tree commit时不要勾选push选项,然后执行自定义动作review, review动作的配置如下:

修改完被打回的review,重新提交时,不要使用source tree提交,直接执行自定义动作updateReview, updateReview动作的配置如下:

## 3、merge分支代码

场景:在分支A提交了很多次代码,并全部review通过入库;现在要把分支A代码merge到分支B,使用merge还是rebase

merge和rebase的区别

merge的特点:

- a.原来节点中的commitID都没有发生变化,在最后增加了一个merge节点(有两个parentID),冲突在一次全部提出
- b. 保留完整的提交历史,在sourceTree看会有很多分叉合并的点

rebase的特点:

- a.变基操作,部分节点会在新的父节点基础上重做提交操作,原来节点的parentID和commitID都发生了变化,commiter会变,author不会变(sourceTree上显示的是author),有冲突会分多次解决
- b.没有增加新的节点,显示的是线性树,没有分叉和合并点

基于以上对merge和rebase的分析 和上面对gerrit的工作原理(commitID和changeID),推荐大家根据不同的情况使用rebase和merge

1. 分支合并的时候使用merge,保留完整的提交历史,merge的时候要保证merge分支的提交已

部review通过并入库

2、同一个分支拉取远端代码使用rebase,git pull --rebase,在每次提交review前先拉取远端代码  
少统一分支中分叉和合并的点,尽量做到线性化

## 4.可能需要的错误排查指令

1)确定是否执行了 scp 命令

2)若没有change-id可以尝试重新设置提交信息 git commit --amend

3)尝试进行软重置 不会丢代码 git reset --soft b64856500a5f0f05a5aced05b607493eaca290

4)如果commit时产生了change-id,但是无法使用source tree提交,报错为'fatal : could not read Username for '<https://gerrit.zhuge.com>': Device not configured',可以先用终端执行脚本 sh review.sh 尝试一下 终端内会提示输入username 而source tree不会有这个提示直接报错