

Gerrit 简易手册

PDF编辑器-未注册

由 (注册后这些水印字将不会被保存) 创建

gerrit 使用网页界面管理了不同的开发仓库，对应不同的软件模块，每个仓库有自己的 权限设置，不同用户的权限设置不一样。目前公司 Gerrit 服务器有两个，198 服务器和 199 服务器，相应的服务器地址分别为：

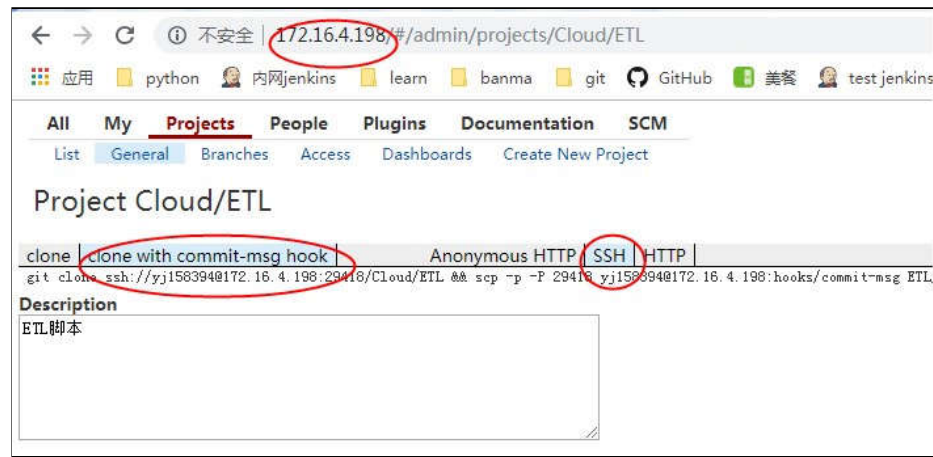
198 服务器，内网：<http://172.16.4.198> 外网：<https://gerrit102.zebred.com/#/>

199 服务器，内网：<http://172.16.4.199> 外网：<https://gerrit103.zebred.com/#/>

登录前请先确定LDAP账号已经申请并登录。原则上在阿里中心使用内网地址，而不在阿里中心的使用外网地址。

1.Gerrit代码下载git clone

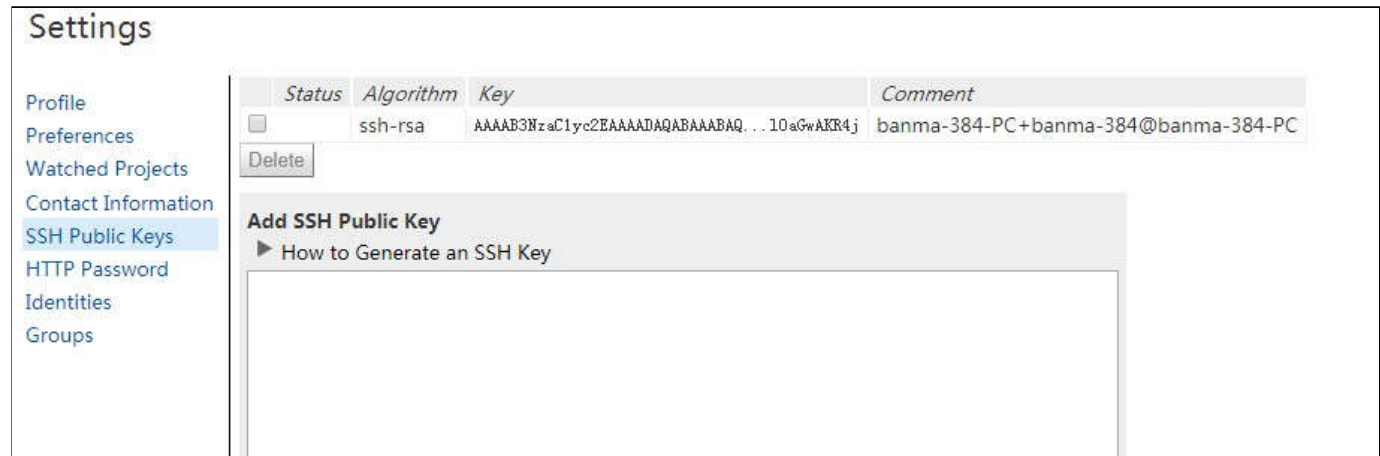
1) 内网使用ssh + clone with commit-msg hook ，如下图



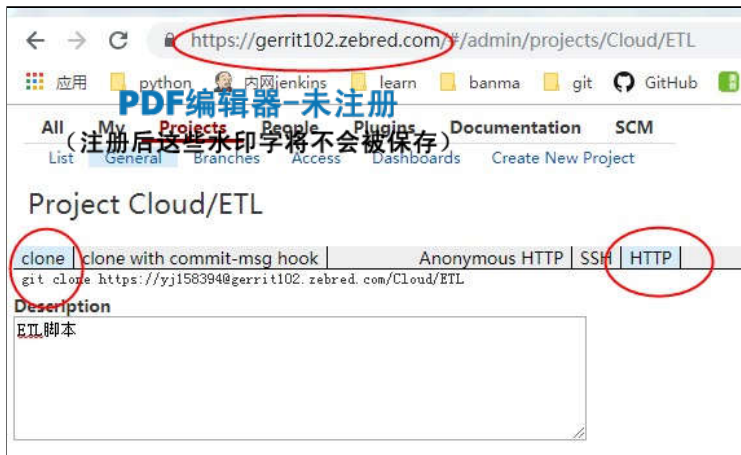
配置sshkey ,

执行如下命令：

- 1) ssh-keygen
- 2) cd ~/.ssh
- 3) cat id_rsa.pub 将文件内容复制到Gerrit用户设置界面如下图所示
- 4) git config --global user.name <gerrit_name>
- 5) git config --global user.email <gerrit_email>



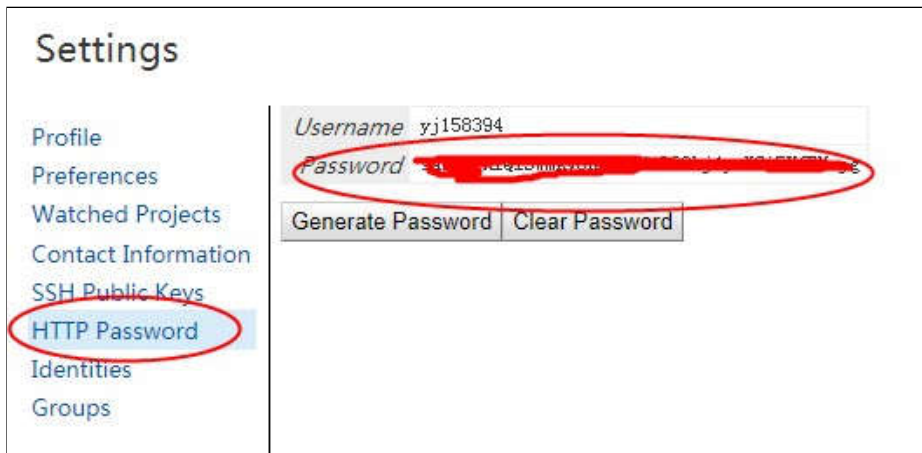
2) 外网使用http + clone ，如下图



本地配置，需要执行如下命令：

- 1) git config --global user.name <gerrit_name>
- 2) git config --global user.email <gerrit_email>
- 3) git config --global http.sslVerify false
- 4) git config --global credential.helper 'cache --timeout=360000'

同时当需要输入密码时，输入的不是LDAP登录密码，而是gerrit个人HTTP PASSWORD，如下图处获取



注意：外网HTTP clone时候没有下载commit-msg，可以直接从本页面的附件中复制到本地仓库.git\hooks\路径下面。

2. 向Gerrit推送代码

推送代码到gerrit的命令如下

1. git diff #查看修改的代码部分
2. git add . #将所有本地修改都加入到暂存区
3. git commit -m "JiralD:ZXQ-201*****" #在本地仓库提交，并写入提交信息
4. git push origin HEAD:refs/for/branch_name #推送到远程的branch_name分支

```
barma-384-PC-barma-384@barma-384-PC:~/workspace/repo_manifest/repo_manifest (yunos60_6x_16P_dev)
$ git push origin HEAD:refs/for/yunos60_ford_dev
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2)
remote:
remote: Processing changes: new: 1, refs: 1, done
remote:
remote: 本次提交将会影响到以下repo分支:
remote:
remote:   Branch Name           项目PM空
remote:
remote: 请确认以下Repo分支是否需要该提交，若需要，请及时cherry-pick:
remote:
remote:   Branch Name           项目PM空
remote: 相关帮助文档链接请查看:
remote:
remote:   ftp://30.16.70.18/SCM_DOC
remote:
remote: Gerrit doesn't validate this project: git/yunos/repo_manifest
remote: Gerrit skips the validation on commit: 6189208dd40c72a3968b4baaf37bcb3b86f4c35e
remote:
remote: New Changes:
remote:   http://172.16.4.199:80/98273
remote:
remote: To ssh://172.16.4.199:29418/git/yunos/repo_manifest
 * [new branch] HEAD -> refs/for/yunos60_ford_dev
```

这个时候查看相关patch url，当需要再修改时候可以执行如下命令行：

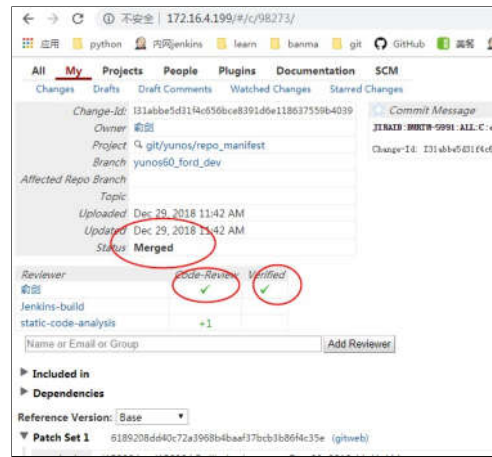
- 1. git add . #将新增的修改文件加入到暂存区
- 2. git commit -s -m '修改' #修改并一次提交，这时可以重新修改提交message
- 3. git push origin HEAD:refs/for/branch_name #再一次推送到远程的branch_name分支

PDF编辑器未注册
(注册后这些水印字将不会被保存)

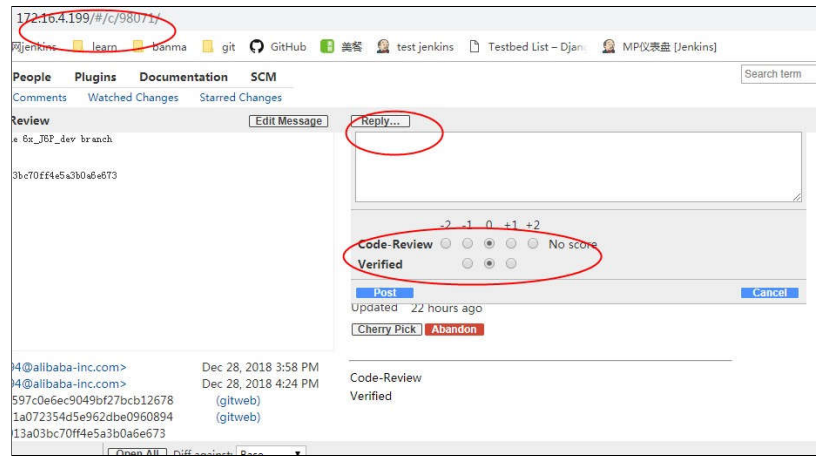
3. 代码入库

通过上面的推送代码后，代码还并没有入库。需要打开相应的patch url。

入库条件需要满足，至少一个code-review+2，以及一个verified+1，才能保证修改入库。



(此为旧UI)



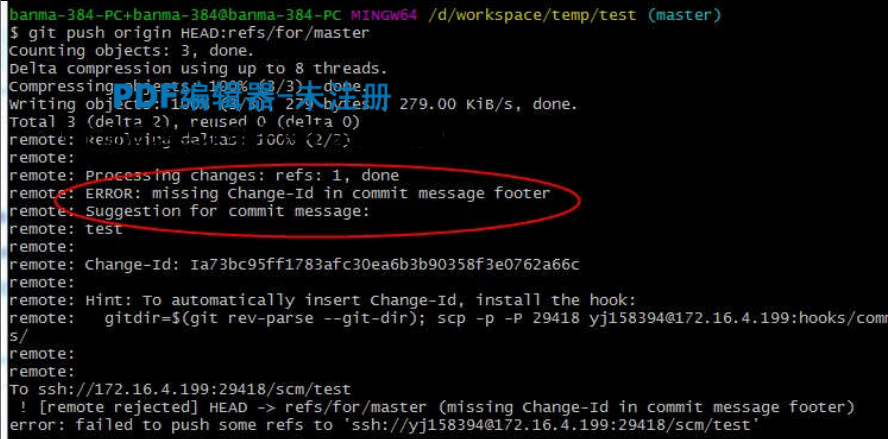
(此为新UI，点击Reply显示code-review以及verified界面)

4 常见错误处理

1) git clone下来的仓库没有代码怎么办？

A：(1) 请先确定该仓库是否为空仓库。(2) 如果不是空仓库，请确定是否在正确的分支位置，切换分支请执行git checkout -b branch origin/branch

2) 代码push失败，并且出现如下错误怎么办？

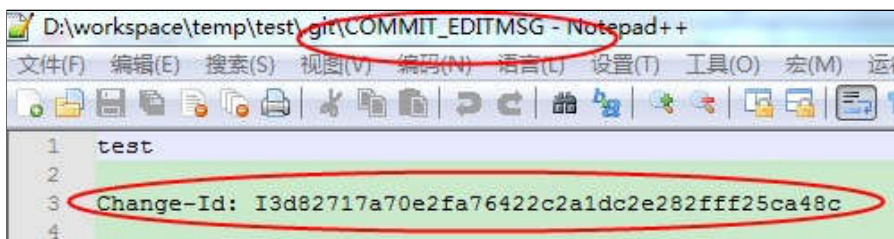


```
banma-384-PC+banma-384@banma-384-PC MINGW64 /d/workspace/temp/test (master)
$ git push origin HEAD:refs/for/master
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing object: 100% (2/2), 279.00 Kib/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: resolving deltas: 100% (2/2)
remote:
remote: Processing changes: refs: 1, done
remote: ERROR: missing Change-Id in commit message footer
remote: Suggestion for commit message:
remote: test
remote:
remote: Change-Id: Ia73bc95ff1783afc30ea6b3b90358f3e0762a66c
remote:
remote: Hint: To automatically insert change-Id, install the hook:
remote: gitdir=$(git rev-parse --git-dir); scp -p -P 29418 yj158394@172.16.4.199:hooks/commit-msg
remote:
remote: To ssh://172.16.4.199:29418/scm/test
! [remote rejected] HEAD -> refs/for/master (missing change-Id in commit message footer)
error: failed to push some refs to 'ssh://yj158394@172.16.4.199:29418/scm/test'
```

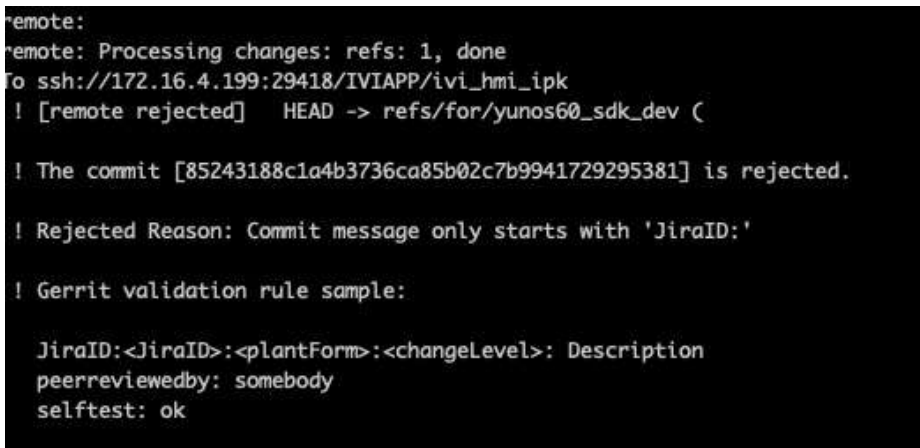
A：这个是缺少了commit-msg文件在.git/hooks/文件夹下，请先下载本页面的commit-msg附件到本地仓库的.git/hooks/路径下，然后执行如下命令

git commit --amend (可能会需要执行多次)

直到出现change-id这一行，如图。

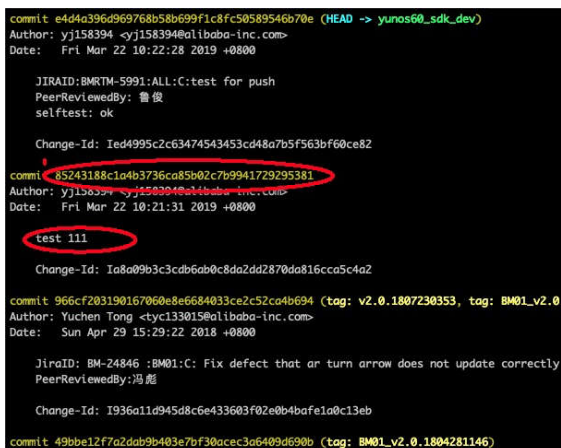


3) 提交被rejected了，是因为什么？如下图的错误



```
remote:
remote: Processing changes: refs: 1, done
To ssh://172.16.4.199:29418/IVIAPP/ivi_hmi_ipk
! [remote rejected] HEAD -> refs/for/yunos60_sdk_dev (
! The commit [85243188c1a4b3736ca85b02c7b9941729295381] is rejected.
! Rejected Reason: Commit message only starts with 'JiraID:'
! Gerrit validation rule sample:
JiraID:<JiraID>:<plantForm>:<changeLevel>: Description
peerreviewedby: somebody
selftest: ok
```

A：这是因为本地有提交有问题，可以使用git log查看rejected commit id的 commit 信息是否正确。如下图



```
commit e4d4a396d969768b58b699f1c8fc50589546b70e (HEAD -> yunos60_sdk_dev)
Author: yj158394 <yj158394@alibaba-inc.com>
Date: Fri Mar 22 10:22:28 2019 +0800

JiraID:BMRTM-5991:ALL:C:test for push
PeerReviewedBy: 曹俊
selftest: ok

Change-Id: Ied4995c2c63474543453cd48a7b5f563bf60ce82

commit 85243188c1a4b3736ca85b02c7b9941729295381
Author: yj158394 <yj158394@alibaba-inc.com>
Date: Fri Mar 22 10:21:31 2019 +0800

test 111

Change-Id: Ia8a09b3c3cd6ab0c8da2dd2870da816cca5c4a2

commit 966cf283190167060e8e6684033c2c52ca4b694 (tag: v2.0.1807230353, tag: BM01_v2.0)
Author: Yuchen Tong <tyc133015@alibaba-inc.com>
Date: Sun Apr 29 15:29:22 2018 +0800

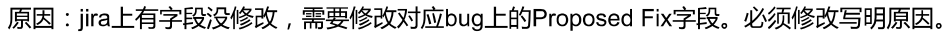
JiraID: BM-24846 :BM01:C: Fix defect that ar turn arrow does not update correctly
PeerReviewedBy: 冯彪

Change-Id: I936a11d945d8c6e433603f02e0b4bafela0c13eb

commit 49bbe12f7a2dab9b403e7bf30ace3a6409d690b (tag: BM01_v2.0.1804281146)
Author: Yuchen Tong <tyc133015@alibaba-inc.com>
Date: Sun Apr 29 15:29:22 2018 +0800
```

解决方案，git checkout {commit_id}，git commit --amend修改提交信息，然后git push再次提交

错误如图：



1.1 Gerrit代码提交规范-代码提交时（同时适用AVN软件及TBox及平台B）

- ## 1.2 Gerrit代码提交(merae)规范-代码review时 (同时适用AVN软件及TBox及平台B)

- 5) 代码提交到了SOP分支，但是没有merge到对应的dev分支（判断标准是相同的changeid要存在于SOP分支的dev分支），错误原因和说明如下



5:48 PM

Patch Set 1:

PDF编辑器-未注册
merge warning ::: Code-Review+2 should not only be yourself
(注册后这些水印字将不会被保存)

→ 不能自己review+2

Gerrit Code Review 5:48 PM

Change contains an invalid project configuration: One of the plugin configuration parameters has a value that is not permitted.

Jenkins-build 5:48 PM

Patch Set 1: Code-Review+2

5:48 PM

Patch Set 1:

merge warning ::: jirald [BMRTM-5991] is not in the white list

→ 不在白名单列表

Gerrit Code Review 5:48 PM

Change contains an invalid project configuration: One of the plugin configuration parameters has a value that is not permitted.

Jenkins-build 5:48 PM

Patch Set 1:

merge warning ::: This patch has not been merged into the main branch : yunos60_6x_dev

→ 提交没有入库到主线分支

Gerrit Code Review 5:48 PM

Change contains an invalid project configuration: One of the plugin configuration parameters has a value that is not permitted.

Jenkins-build 5:49 PM

Patch Set 1: Cherry Picked

→ cherry-pick到dev分支后
此sop分支提交才能入库

This patchset was cherry picked to branch yunos60_6x_dev as commit 246243e29b5aa86d5e023ef3cf77721fad624d72

5:50 PM

Patch Set 1: Code-Review+1

Jenkins-build 5:50 PM

Change has been successfully merged into the git repository.

Jenkins-build 5:50 PM

Patch Set 1:

Bug ID List : <http://172.16.4.200:8090/browse/BMRTM-5991>

分支合并操作。

需求：将B分支的代码和更新为和A分支一样，同时保留A分支和B分支的提交记录

- 1 . git checkout B
- 2 . git merge -s ours A
- 3 . git read-tree -um A
- 4 . git commit --amend(为了生成changeid，同时将暂存区的内容提交，此时A分支和B分支内容会一致)
- 5 . git push origin HEAD:refs/for/B



无标签