

grrrit提交

git push origin HEAD:refs/for/your_version

tag

老版本使用tag方案，因此可以直接使用 `git checkout {tag}` 切换到对应分支

cherry-pick

```
1 | git cherry-pick commit_id //将此commit id的提交合并到当前分支
```

gerrit提交代码失败

设置一下hook

git打补丁

```
1 | 比如你现在提交的是d576b826，需要在这个上面打patch
2 | 1.git checkout d576b826
3 | 2.正常修改
4 | 3.git add .
5 | 4.git commit --amend
6 | 5.再次git push到gerrit上即可
```

git push 错了怎么回退

1、git log 查看版本号。[commit](#)后面那一串字符就是版本号，第一条记录就是你最后提交的一次的版本号。我们需要回退到它前面那个版本。复制第二条记录的版本号。然后按英文 q退出。

2、git [reset](#) --hard 版本号（前面要求复制的）这会删除本地修改文件，也就是本地和远程版本内容一致。

git reset --soft 版本号 只会将远程的版本回退。本地文件不动。（建议使用这条）如果只是回退上一个版本，指令可以这样写。git reset --soft HEAD^。

3、git push -f origin master 这步错了大概率就是master要改成你那边分支的名字。

git rebase

git rebase 命令修改第一笔提交，而不影响第二笔提交。

假设你的本地分支的提交历史记录如下：

```
1 | c2f02d5 (HEAD -> master) second commit
2 | 614a3a7 first commit
```

其中，第一笔提交标记为 614a3a7。

要修改第一笔提交，需要执行以下步骤：

1. 运行 `git rebase -i HEAD~2` 命令，将最后两个提交（即 614a3a7 和 c2f02d5）传递给 rebase 命令。此命令将打开与当前编辑器关联的交互式文件。

2. 在文本编辑器中，找到要修改的提交，并在行首将 `pick` 更改为 `edit`。保存并关闭文件。
3. 运行 `git commit --amend` 命令来修改提交。这将打开默认编辑器，其中包含您上次提交的消息。
4. 请修改提交消息。然后保存并关闭编辑器。
5. 运行 `git rebase --continue` 命令以完成重新定义基础行操作。如果有冲突需要解决，请解决冲突并使用 `git add` 命令添加更改后执行 `git rebase --continue`。

完成这些步骤后，你会得到新的提交历史：

```
1 | 复制代码 * f455b5d (HEAD -> master) first commit
2 | * c2f02d5 second commit
```

现在，第一笔提交已被修改为 `f455b5d`，而第二笔提交仍然保持不变为 `c2f02d5`。

需要注意的是，这种方法只应在本地分支上使用。如果您的更改已经被推送到共享分支中，那么对历史记录进行重新整理可能会引起合并冲突和其他问题。