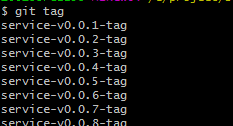
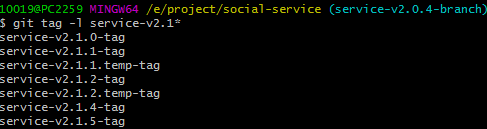
列出已有的标签

#git tag



通配符列出部分tag

git tag -l service-v2.1\*



Git 支持两种标签：轻量标签（lightweight）与附注标签（annotated）

轻量标签很像一个不会改变的分支——它只是某个特定提交的引用

而附注标签是存储在 Git 数据库中的一个完整对象， 它们是可以被校验的，其中包含打标签者的名字、电子邮件地址、日期时间， 此外还有一个标签信息，并且可以使用 GNU Privacy Guard （GPG）签名并验证。 通常会建议创建附注标签，这样你可以拥有以上所有信息。但是如果你只是想用一个临时的标签， 或者因为某些原因不想要保存这些信息，那么也可以用轻量标签。

**附注标签**

git tag -a v1.4 -m "my version 1.4"

git show 命令可以看到标签信息和与之对应的提交信息

### 轻量标签

轻量标签本质上是将提交校验和存储到一个文件中——没有保存任何其他信息

创建轻量标签，不需要使用 -a、-s 或 -m 选项，只需要提供标签名字

git tag v1.4-lw

在标签上运行 git show，你不会看到额外的标签信息。 命令只会显示出提交信息

轻量标签本质上是将提交校验和存储到一个文件中——没有保存任何其他信息。 创建轻量标签，不需要使用 -a、-s 或 -m 选项，只需要提供标签名字

### 后期打标签

git tag -a v1.2 9fceb02

要在那个提交上打标签，你需要在命令的末尾指定提交的校验和（commit id）

### 共享标签

默认情况下，git push 命令并不会传送标签到远程仓库服务器上。 在创建完标签后你必须显式地推送标签到共享服务器上。 这个过程就像共享远程分支一样——你可以运行 git push origin <tagname>

git push origin v1.5

如果有默认的远程仓库名，可以不用谢origin

git push v1.5

如果想要一次性推送很多标签，也可以使用带有 --tags 选项的 git push 命令。 这将会把所有不在远程仓库服务器上的标签全部传送到那里。

git push origin --tags

git push --tags

### 删除标签

要删除掉你本地仓库上的标签，可以使用命令 git tag -d <tagname>

git tag -d v1.4-lw

注意上述命令并不会从任何远程仓库中移除这个标签，你必须用 git push <remote> :refs/tags/<tagname> 来更新你的远程仓库

第一种变体是 git push <remote> :refs/tags/<tagname>

git push origin :refs/tags/v1.4-lw

含义是，将冒号前面的空值推送到远程标签名，从而高效地删除它

第二种更直观的删除远程标签的方式

git push origin --delete <tagname>

### 检出标签

检出某一个标签

如果你想查看某个标签所指向的文件版本，可以使用 git checkout 命令， 虽然这会使你的仓库处于“分离头指针（detacthed HEAD）”的状态——这个状态有些不好的副作用

在“分离头指针”状态下，如果你做了某些更改然后提交它们，标签不会发生变化， 但你的新提交将不属于任何分支，并且将无法访问，除非通过确切的提交哈希才能访问。 因此，如果你需要进行更改，比如你要修复旧版本中的错误，那么通常需要创建一个新分支：

$ git checkout -b version2 v2.0.0

Switched to a new branch 'version2'

如果在这之后又进行了一次提交，version2 分支就会因为这个改动向前移动， 此时它就会和 v2.0.0 标签稍微有些不同，这时就要当心了。

参考文档

<https://git-scm.com/book/zh/v2/Git-%E5%9F%BA%E7%A1%80-%E6%89%93%E6%A0%87%E7%AD%BE>