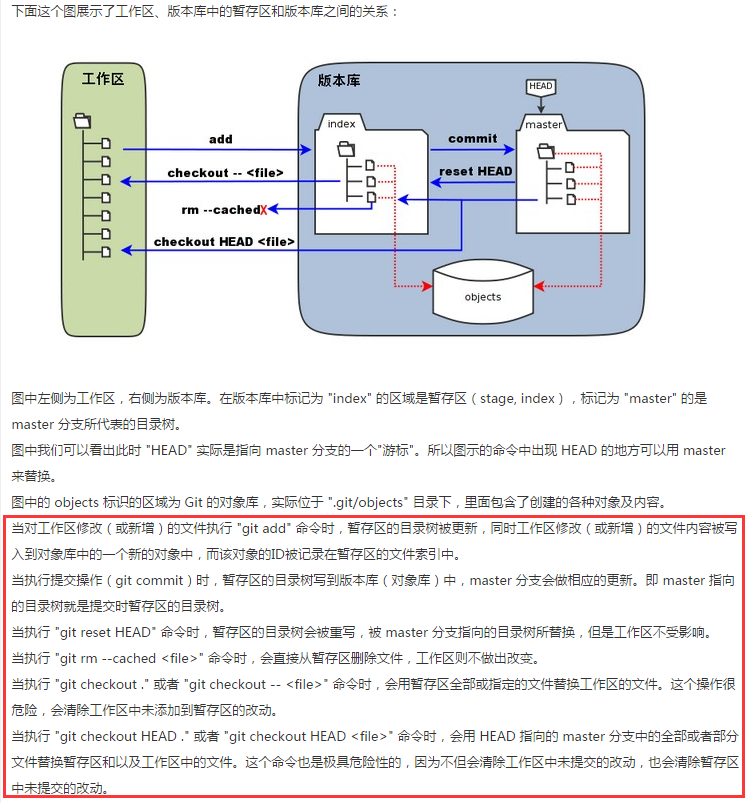
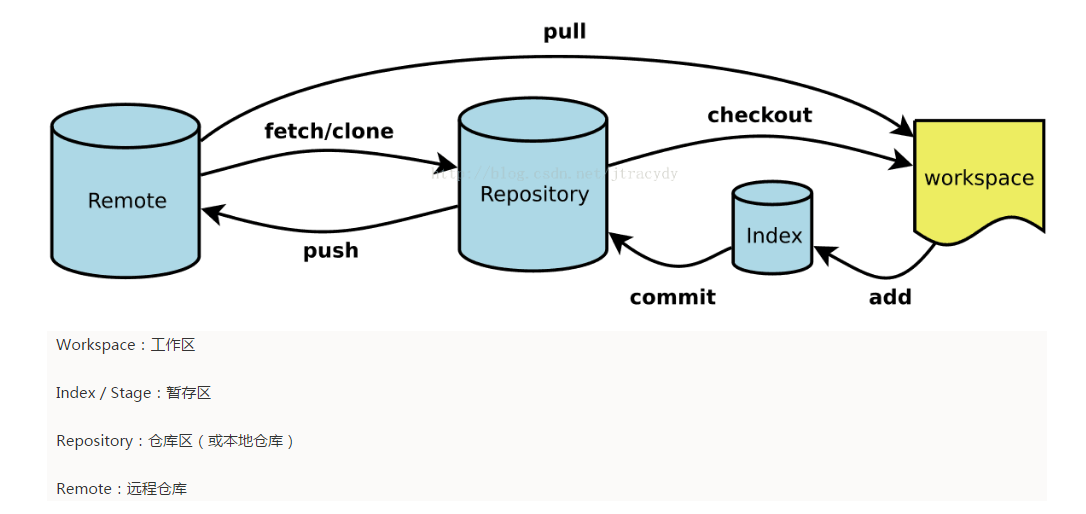
1. Git的工作原理：

使用**本地仓库：**

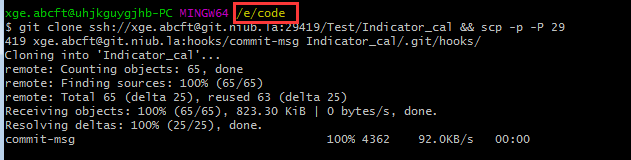


本例中使用到了**远程仓库**，架构如图：



1. 下载git上的项目到本地：

**注意：最好进入到存储项目的文件夹后，再执行下载命令，下载命令可从网页复制**

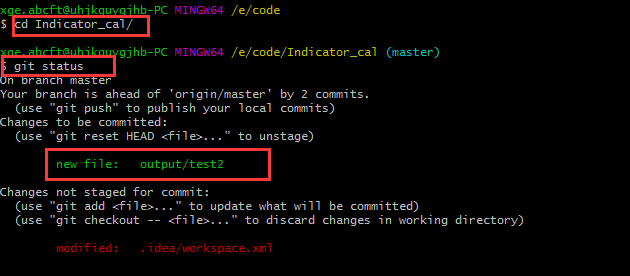


1. 对下载下来的项目进行更改，可以用**git status 命令查看项目的当前状态**

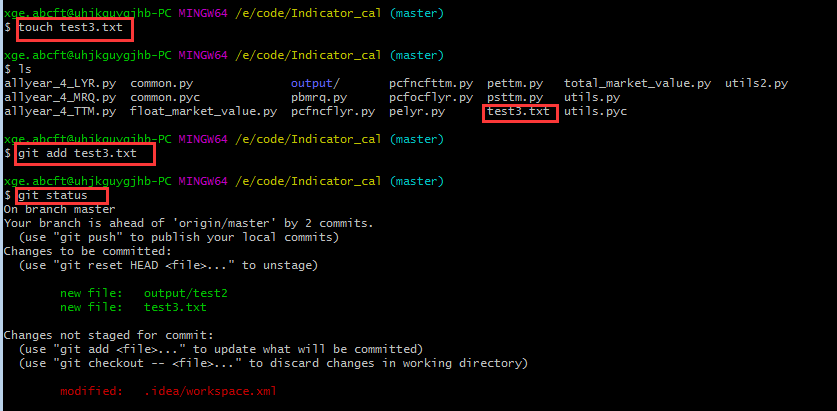
如在当前项目新增一个文件，有两种方式：

3.1直接在IDE中用交互式的方式新建，比如：test2.txt

新建之后，查看项目当前状态：



3.2 用git命令新建文件并添加到缓存，比如：用touch命令新建 test3.txt，然后用**git add 命令将该文件添加到缓存**



1. Git分支管理（单独再列出来）：

**git branch 列出所在的分支**

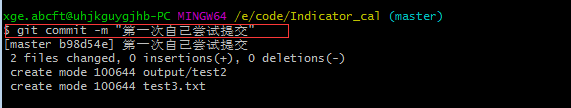


**git checkout 切换到要修改的分支**(本例子中用的是当前所在的master分支，并未切换到其他)

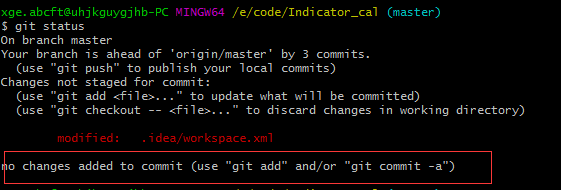


（5）代码提交，使用**git commit命令将缓存区内容提交到本地仓库**

**可用-m参数提供提交注释**

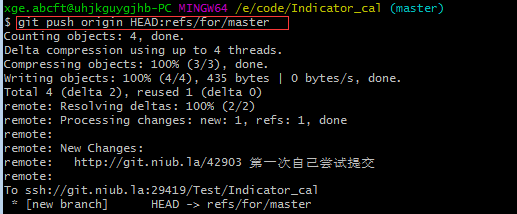


提交之后，再git status 命令查看项目的当前状态,就会提示:

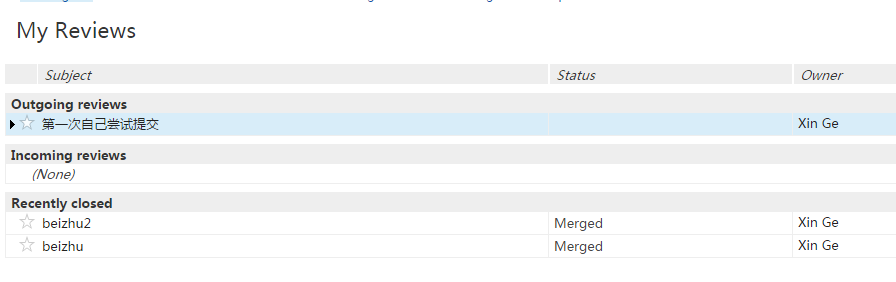


1. 把本地仓库的更新推送到远程仓库：使用**git push命令**

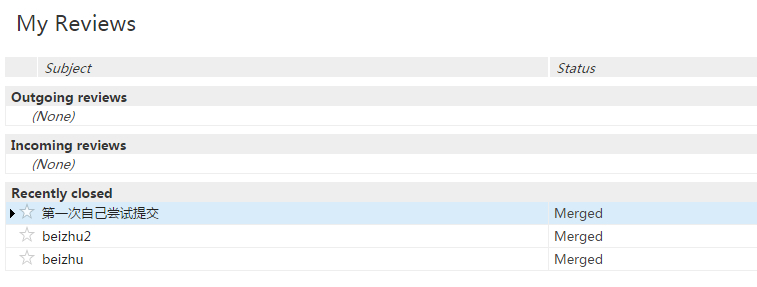
注释： refs/for/master表示需要经过Code review 才可以提交到远程仓库



推送后，可以在远程仓库看到自己提交的内容：



Code review之后：



1. 在项目里删除更新：

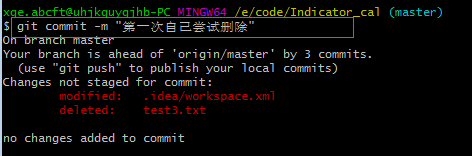
注意：首先应在项目里删除掉，然后提交及推送才会有效

6.1 手动在项目里删除：

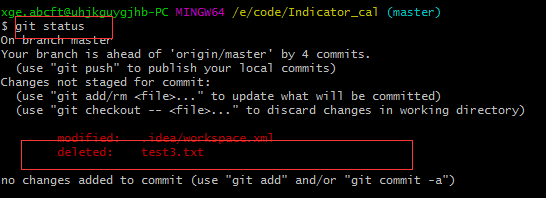
6.2 用linux **rm**命令删除：



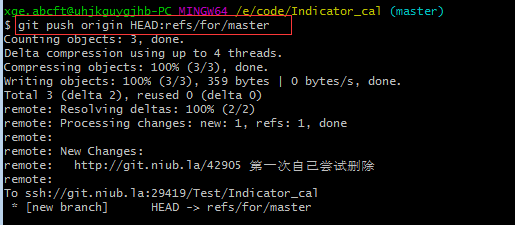
1. 同上，首先提交到本地代码库，然后推送到远程代码仓库：



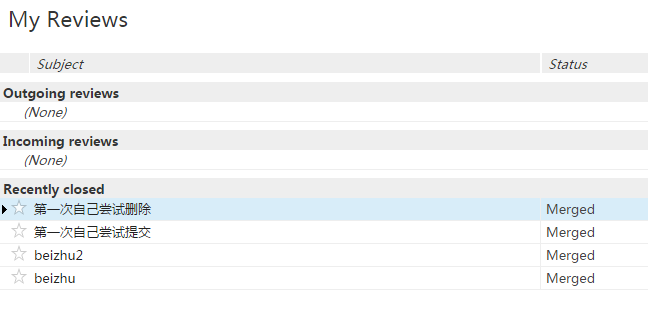
提交后 可以再次查看项目最新状态：



推送到远程服务器仓库：

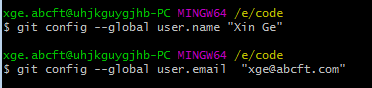


Code review之后



以上，就完成了git上传和删除更新的基本操作。

PS：git配置用户名和邮箱：



git查看用户名和邮箱：

