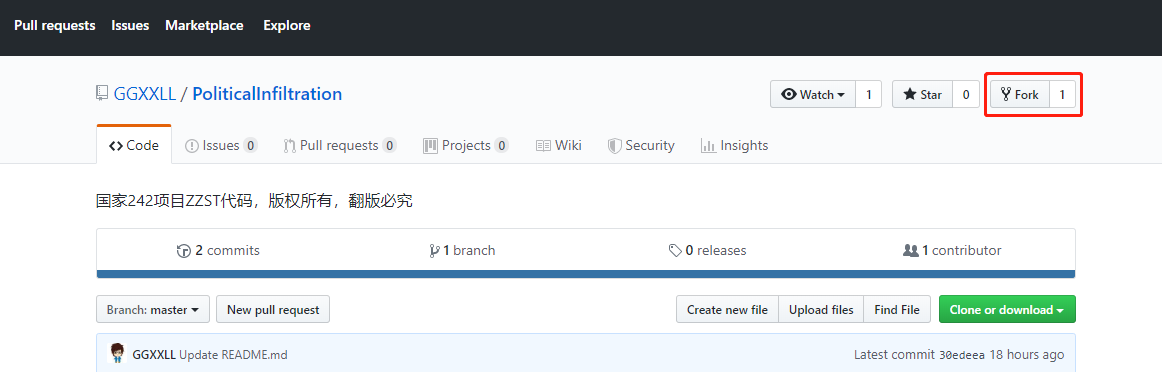
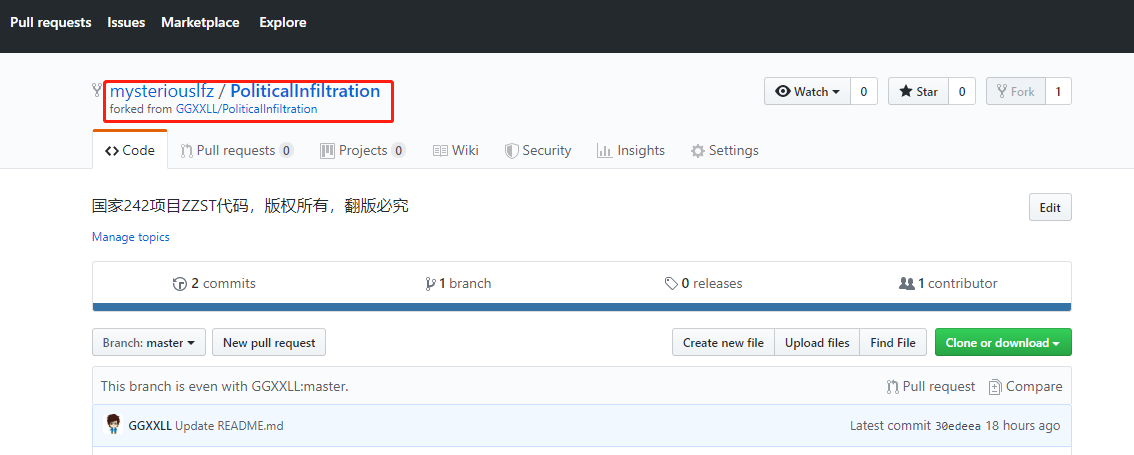
## GitHub及git命令基本使用说明

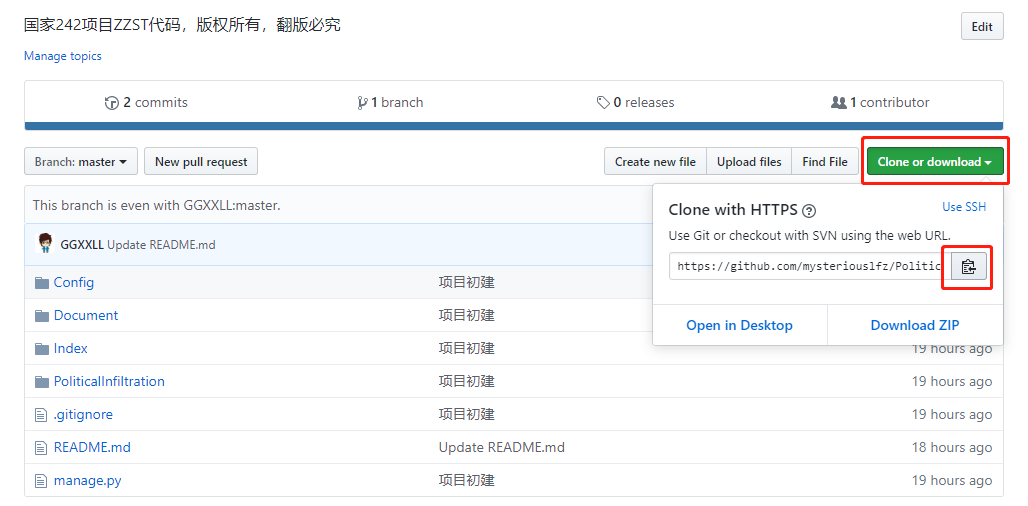
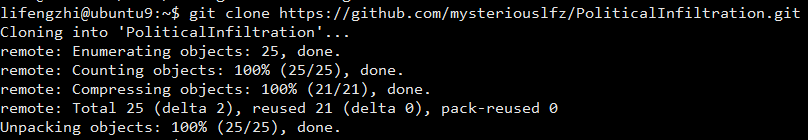
1. 注册后进入对应的项目首页（政治渗透：<https://github.com/GGXXLL/PoliticalInfiltration>，为项目主库，所有人的修改会提交到这里），首先fork到自己的目录下。

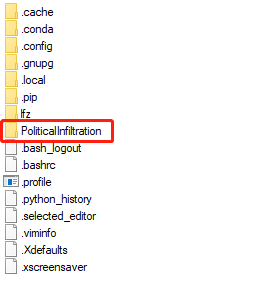


1. Fork后可以在自己的页面下看见这个项目



1. 点击按钮克隆项目，复制点击按钮辅助项目链接。在Linux终端输入git clone 进行克隆，克隆结束后会发现项目已经出现在本地（自己的分支库，先提交到这里，再合并的到主库）。

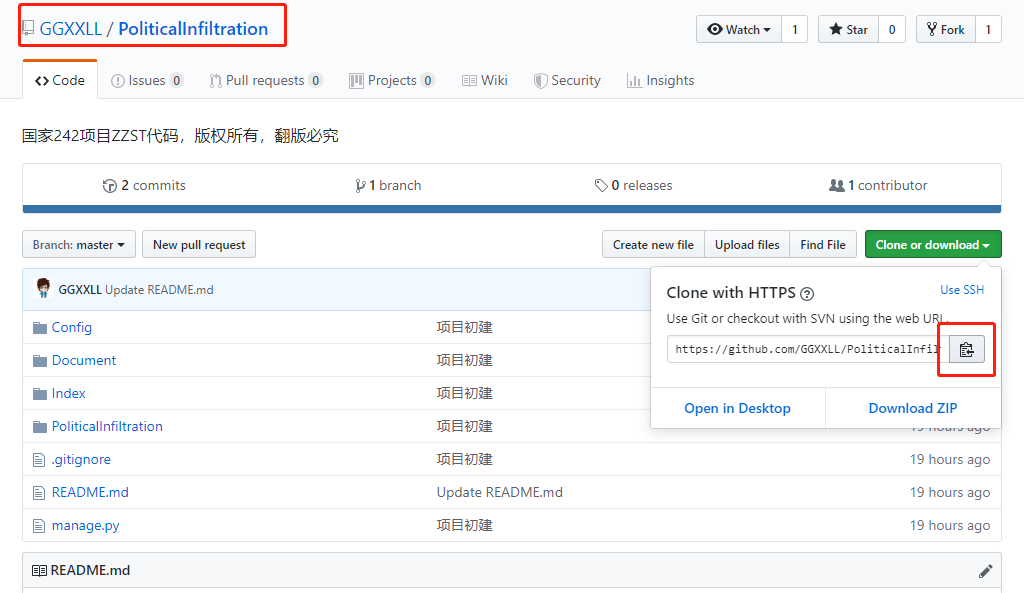
 

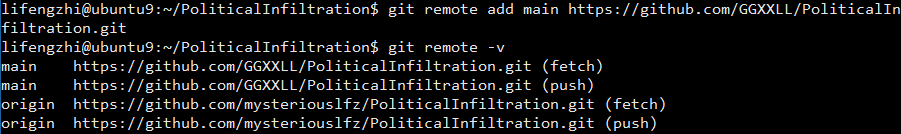


1. 输入git remote -v命令，可以查看到当前项目的分支，origin即为刚刚的自己的分支（自己的分支库）

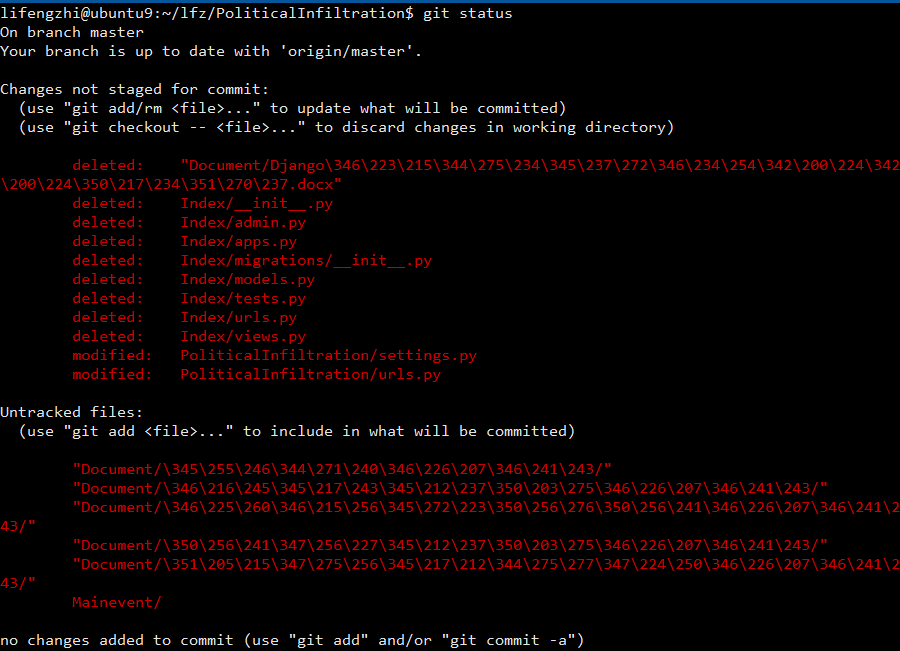


回到项目主库，复制其克隆链接，利用git remote add main <https://github.com/GGXXLL/PoliticalInfiltration.git> （这里“main”为主库的分支名字，可以随意取，后面的为主库的链接）将主库的分支加到当前目录下，这时候可以看到有main的分支了

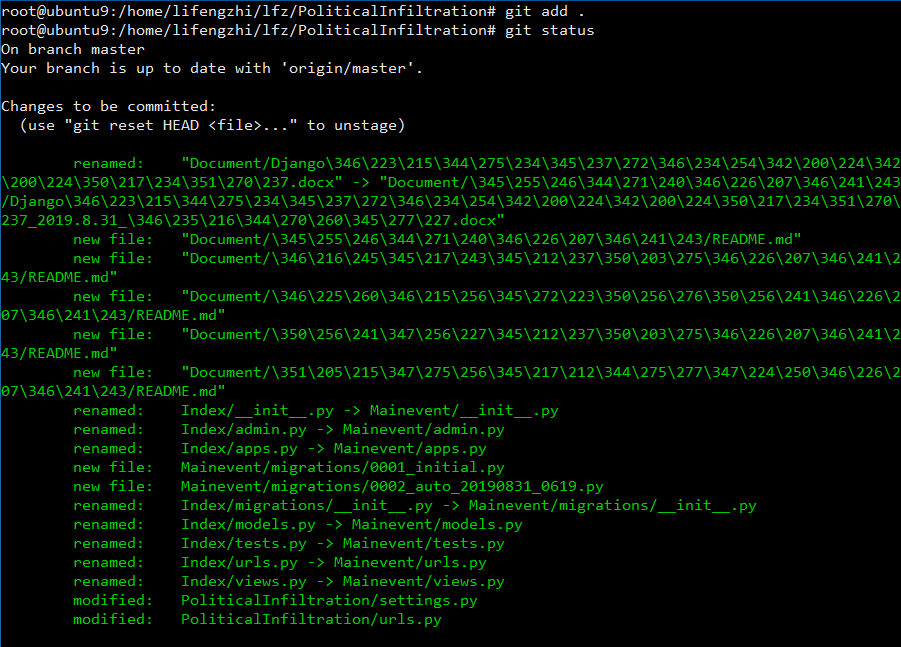




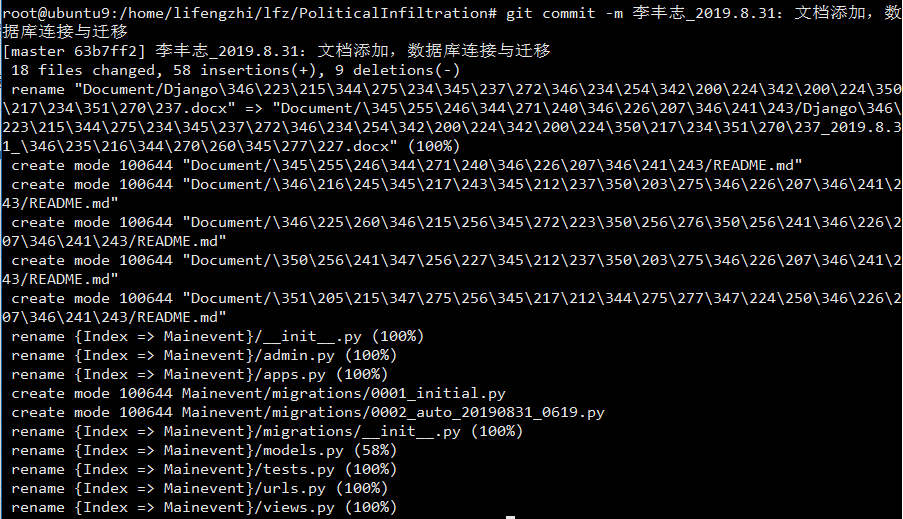
1. 在开发过程中，每次修改完要提交前，确认自己的修改文件是否正确，使用git status命令，显示红色的为这次修改或添加删除的文件



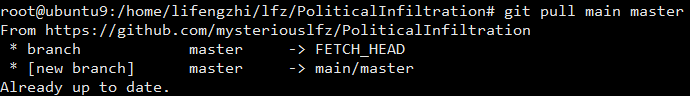
1. 确定无误后对修改进行确认，使用git add .可以确认所有文件，git add <filename>可以确认对应的文件，确认后确认的文件变为绿色



1. 使用git commit -m <说明>进行提交，这里说明处填写的是本次提交的简单说明，可随意填写，但要有意义

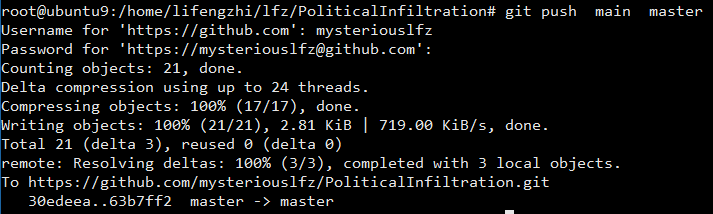


1. 在向自己的分支库提交之前，先从主库pull最新的代码，命令为git pull main master（这里的main为主库的分支名字）



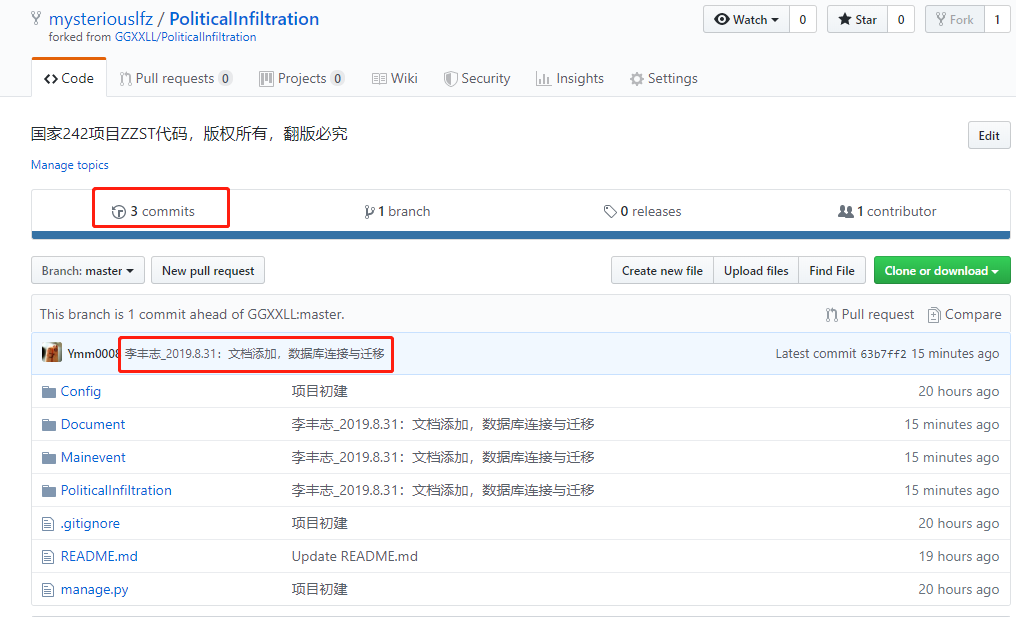
如果主库有更改，就会显示在下面，没有更改则会出现上述说明。有时因为修改同一文件可能会出现冲突而pull失败的情况，查看冲突文件，正确修改后提交修改可重新正常pull

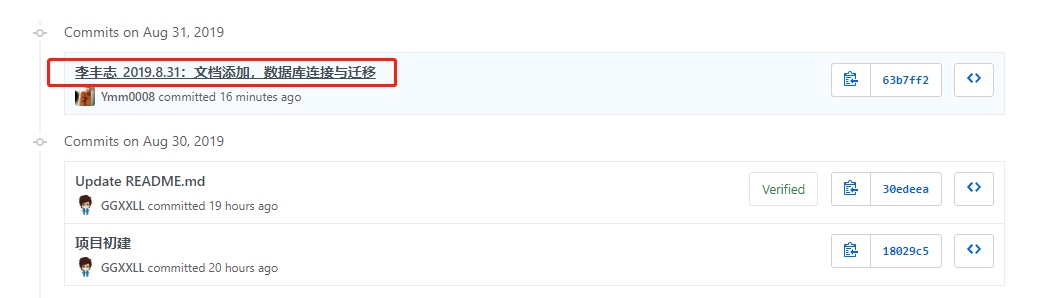
1. 在pull的操作后才可以进行push操作，使用git push origin master（origin为分支库）将修改推送到自己的分支库中，推送后则可以在自己的库里发现修改的文件及说明。

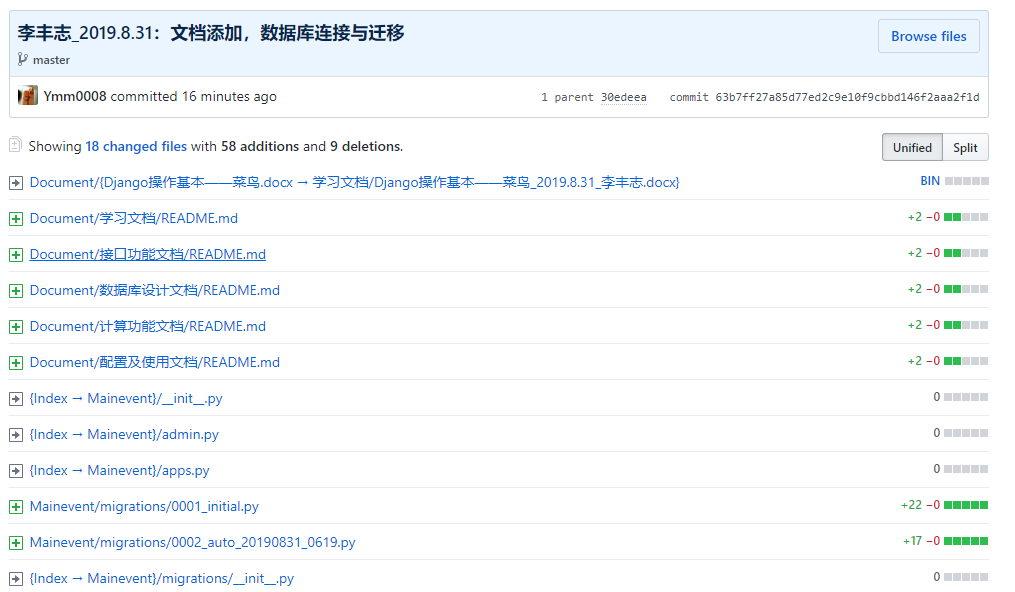


这里需要输入GitHub的用户名和密码，网上可查询到很多不需要输入密码的操作，最简单的方式就是存储在本地服务器，使用git config credential.helper store命令，第一次输入后后续无需再输入。

1. 到自己的分支库查看提交情况

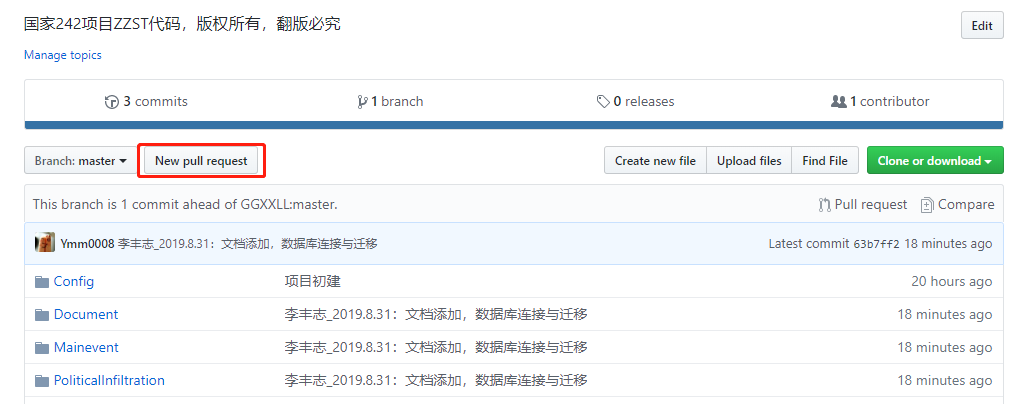


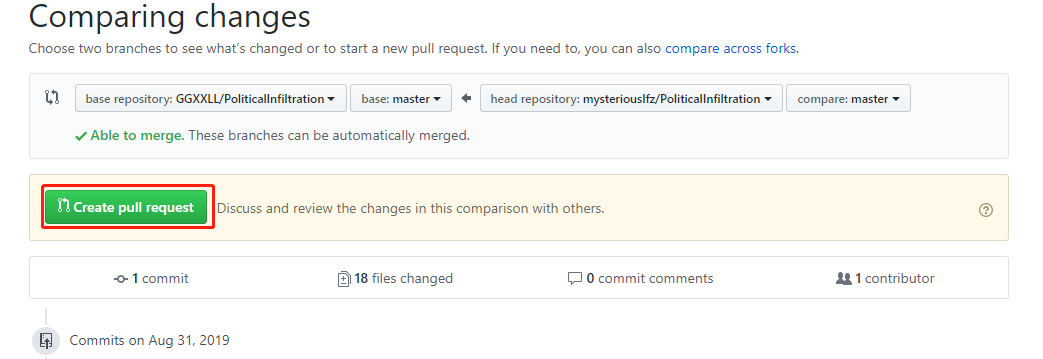


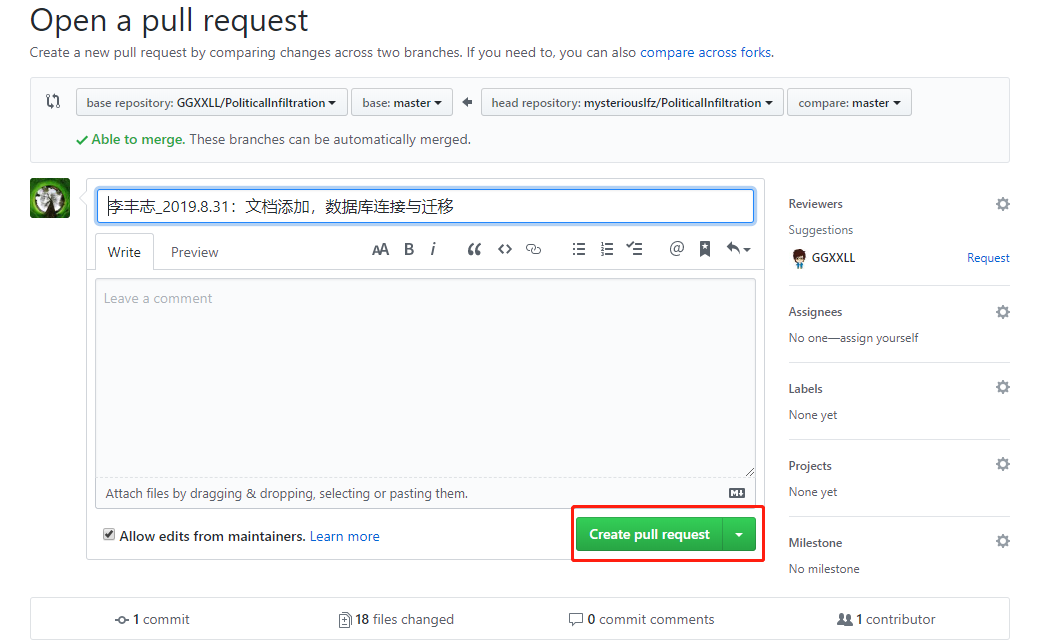


确认自己的提交是否已经正确到达自己的服务器分支

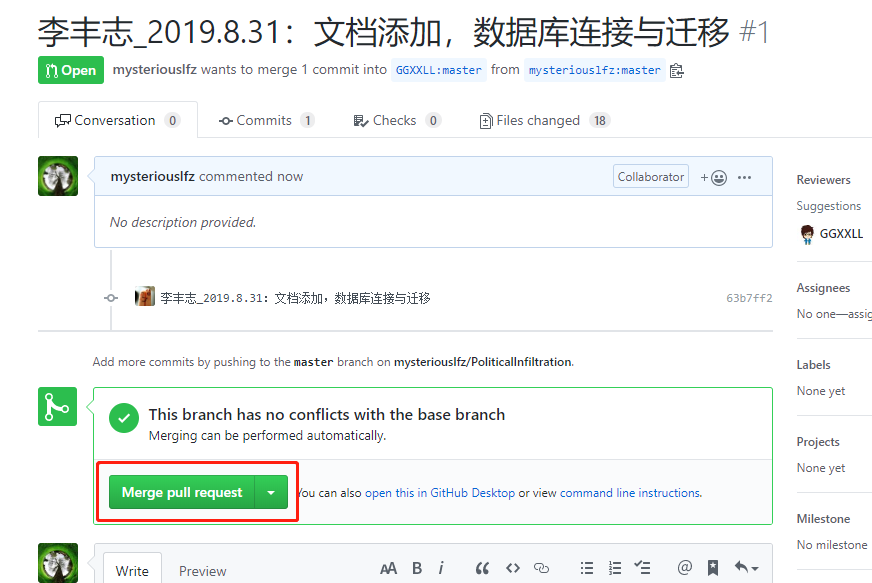
1. 确认无误后，点击New pull request按钮发起合并请求，将分主库的合并请求发至主库

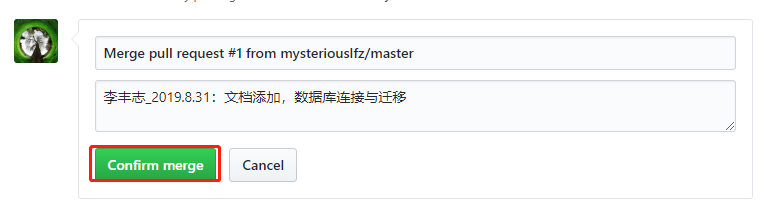






提交请求后该请求会转到主库，由于大家都是主库的协作者，所以给大家开通自己合并的权限，所以要求大家在合并前一定确定好自己的提交是否正确，提交后会覆盖主库之前版本的代码，避免因为疏忽导致其他人的代码被覆盖掉。





合并后在主库也可以看到自己的修改及合并请求了

