

Git Gitlab Github(No)

目录

Git Gitlab Github(No)	1
前奏	1
Gitlab 远端仓库创建	1
本地文件管理	2
本地库与远程库连接	3
分支介绍	4
意见采纳	5
其他	6

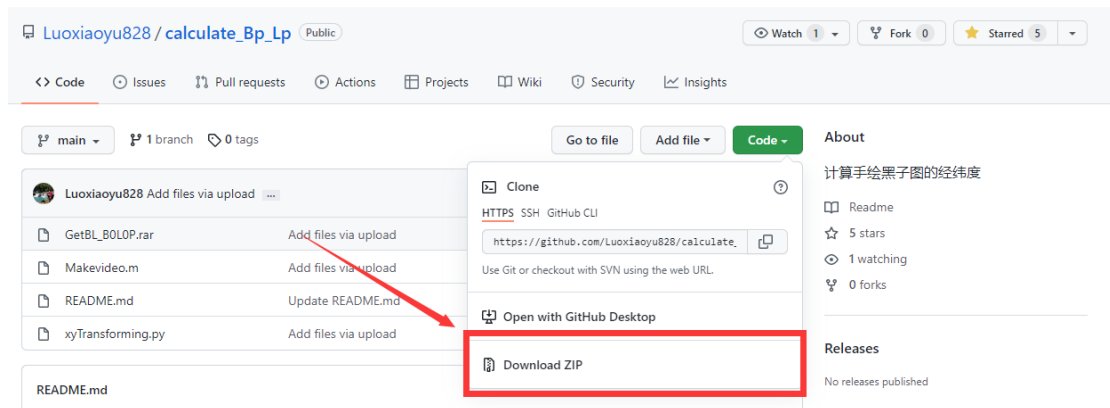
说明：就是纯粹的记录，非正式教学，平时用到的多的操作就记下了，有娱乐改~

前奏

选择性下载文件，如果有文件的话就不用下载，下载的目的是做测试文件用

文件地址：https://github.com/Luoxiaoyu828/calculate_Bp_Lp.git

（不是为 star 而来的，真的不是.....）



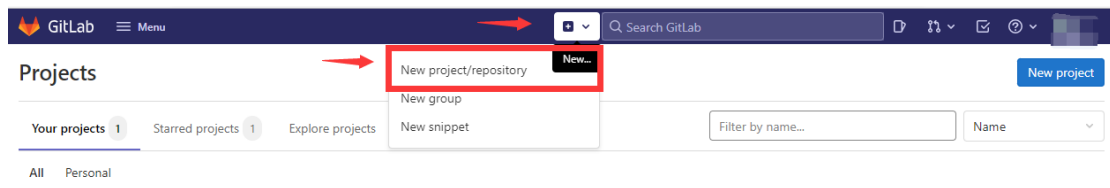
只是为了从 github 下一个文件，和 github 没啥关系，下好了后解压就 ok，不需要 github 了。如果没有 github 账号没关系，创一个 gitlab 的账号，此文档本就是讲 git 与 gitlab 的那些事儿。链接在这：[Projects · Dashboard · GitLab](#)（可以用 github 账号登录），远端的账户有了，然后本地需要安装 git，链接在这：[Git \(git-scm.com\)](#)，做连接 gitlab（当然 github 通用）的本地工具，安装过程一路默认肯定没错，如果认为不够细，参考此博客：[Git 详细安装教程](#)

*#####*卑微分界线*#####*

所以，现在有了本地文件，gitlab 账号，和安装好的 git。（时刻联网哈，没网锤子都传不了）

Gitlab 远端仓库创建

就是托管云端代码的仓库，简单建立如下（当然，可以加入其他人的仓库，在此之前先练习自己库.....）



GitLab

Create new project

Create blank project

Create a blank project to house your files, plan your work, and collaborate on code, among other things.

Create from template

Create a project pre-populated with the necessary files to get you started quickly.

Create blank project

Create a blank project to house your files, plan your work, and collaborate on code, among other things.

Project name
awesome project

Project URL
https://gitlab.com/awesome-project/

Project slug
awesome-project

Want to house several dependent projects under the same namespace? [Create a group](#).

Project description (optional)
my first project

Project deployment target (optional)
Select the deployment target

Visibility Level
☒ Private
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.
☐ Public
The project can be accessed without any authentication.

Project Configuration
☒ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.
☐ Enable Static Application Security Testing (SAST)
Analyze your source code for known security vulnerabilities. [Learn more](#).

Create project Cancel

main awesome-project / +

History Find file Web IDE Clone

Initial commit
authored 10 seconds ago

Upload File README Add LICENSE Add CHANGELOG Add CONTRIBUTING

Set up CI/CD Configure Integrations

Name	Last commit
README.md	Initial commit

Clone with SSH
git@gitlab.com:awesome-project/awesome-project.git

Clone with HTTPS
https://gitlab.com/awesome-project/awesome-project.git

Open in your IDE
Visual Studio Code (SSH)
Visual Studio Code (HTTPS)

本地文件管理

啥都不处理，它就是实打实本地文件，和 gitlab（或者 github）无任何关系，就静静地在那。

此电脑 > 新加坡 (E:) > 我的课题文件 > calculate_Bp_Lp-main

名称	修改日期	类型	大小
GetBL_BOLOP.rar	2022/4/6 11:04	WinRAR 压缩文件	15 KB
Makevideo.m	2022/4/6 11:04	M 文件	2 KB
README.md	2022/4/6 11:04	MD 文件	1 KB
xyTransforming.py	2022/4/6 11:04	JetBrains PyCharm	8 KB
calculate_Bp_Lp-main.zip	2022/4/14 13:39	WinRAR ZIP 压缩文件	21 KB

Git 登场



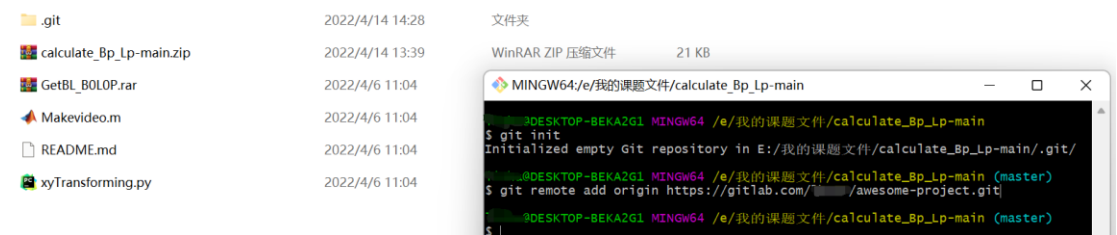
初始化命令: `git init`



执行没动静不着急, 红框为隐藏文件夹, 让它显示。初始化后, 你的本地仓库就建好了。

本地库与远程库连接

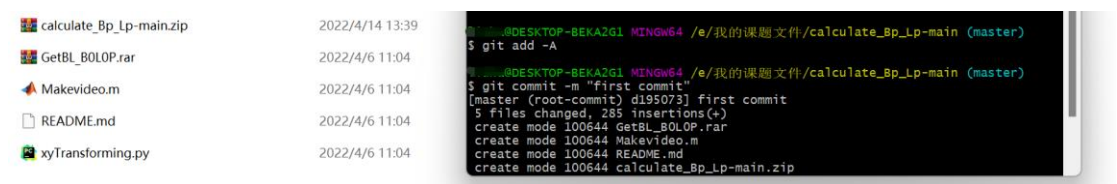
命令: `git remote add origin [仓库名]`, 就是上面远端库说要用到的链接



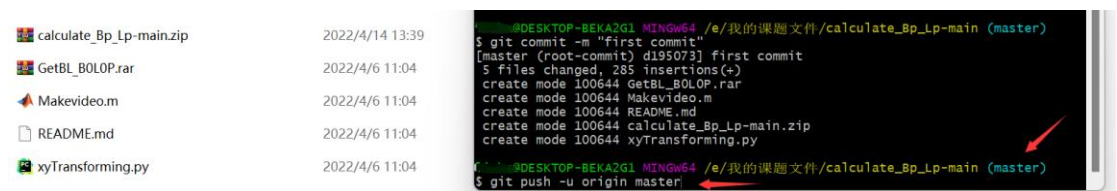
这时, 远端的仓库是空的, 提交操作如下, 本地仓库>>暂存区>>远端仓库

命令: `git add -A` 将所有文件添加到暂存区, `git commit -m "[提交修改的信息]"`

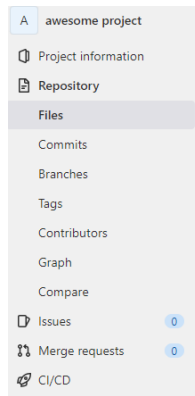
另外 `git add <filename>` 指定文件到暂存区, 初次提交 all



暂存区文件提交至远端仓库, 注意分支, 后面介绍



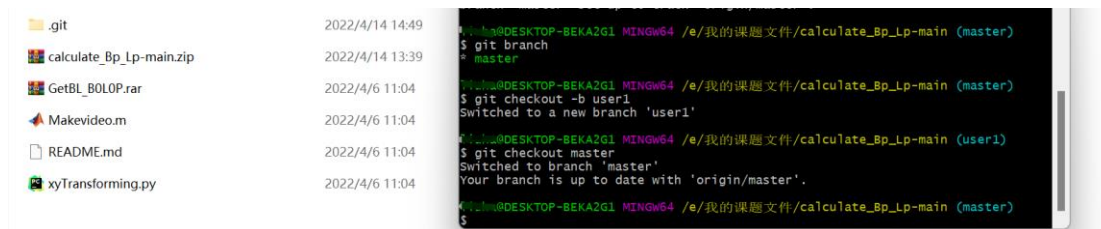
远端仓库 master 分支下就同步本地的文件了。



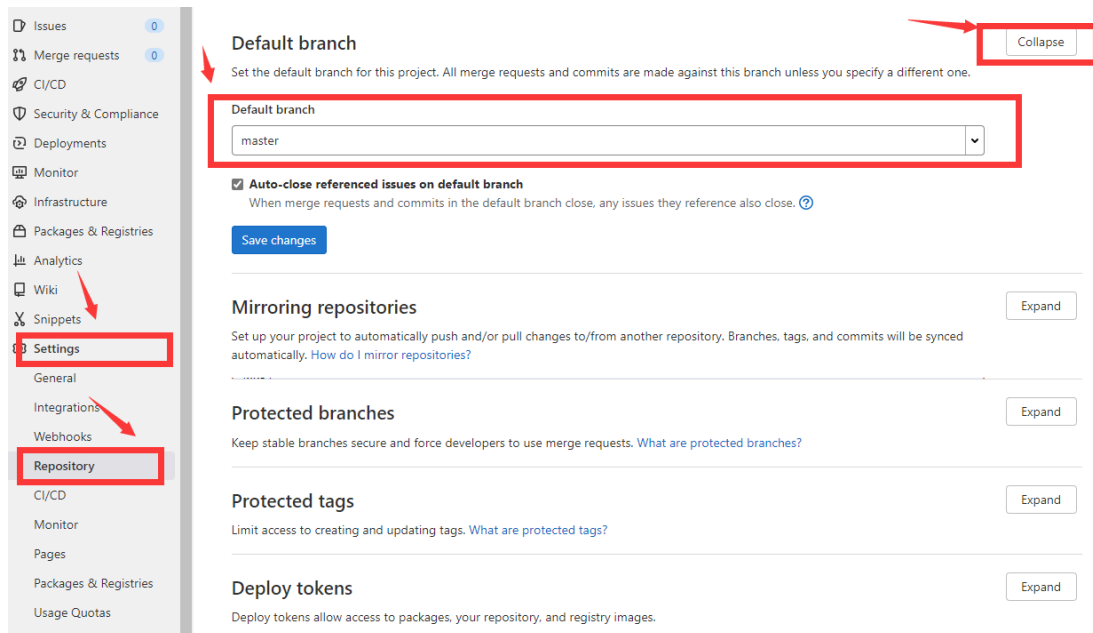
master	awesome-project / +	History	Find file	Web IDE	Download	Clone
first commit authored 7 minutes ago d195073a						
Name	Last commit	Last update				
GetBL_B0LOP.rar	first commit	7 minutes ago				
Makevideo.m	first commit	7 minutes ago				
README.md	first commit	7 minutes ago				
calculate_Bp_Lp-main.zip	first commit	7 minutes ago				
xyTransforming.py	first commit	7 minutes ago				

分支介绍

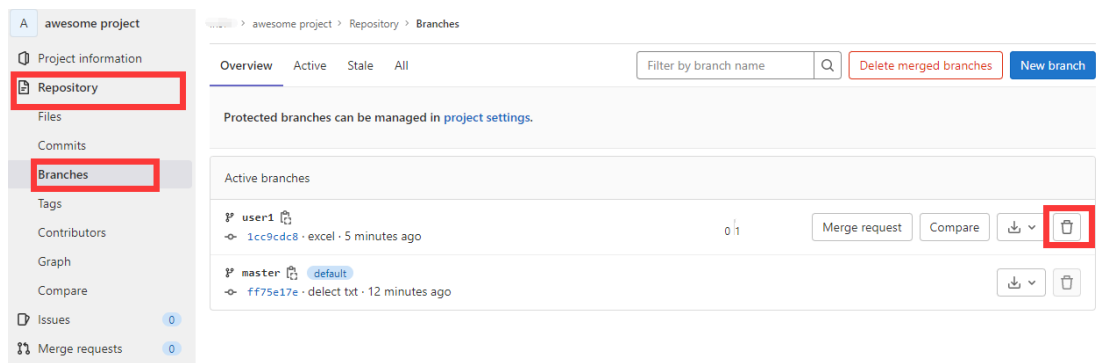
命令: `git branch` 分支列举, `git checkout -b xxx` 创建 xxx 的分支, `git checkout xxx` 选择 xxx 分支, `git branch -D xxx` 删除 xxx 分支, `git merge xxx` 合并分支



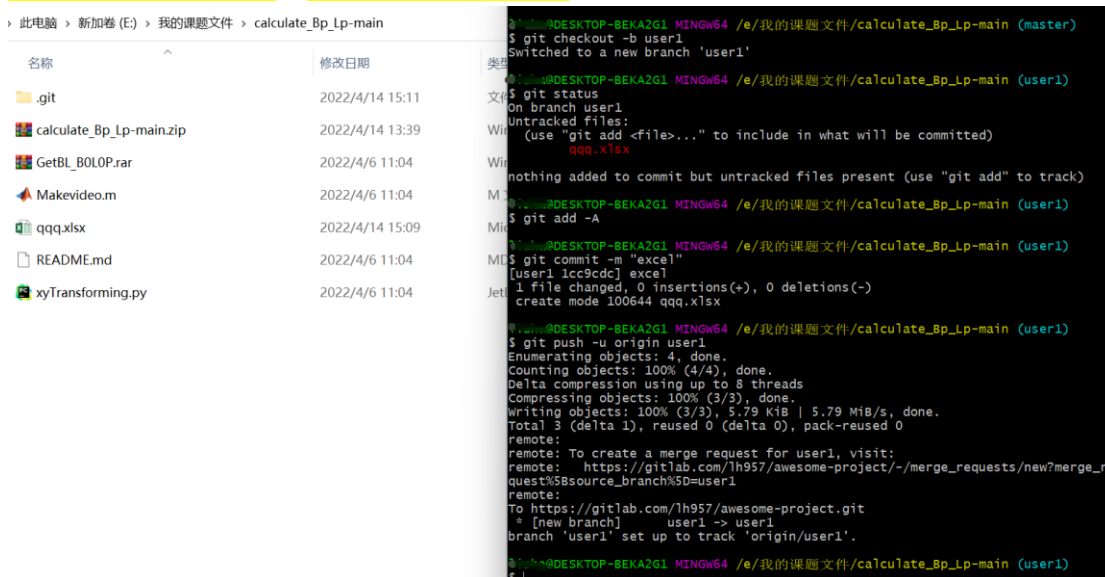
关于 master 分支与远端仓库的 main 分支存在冲突的问题, 修改远端仓库默认分支即可:



选择默认分支为 master, 删除 main 分支即可:

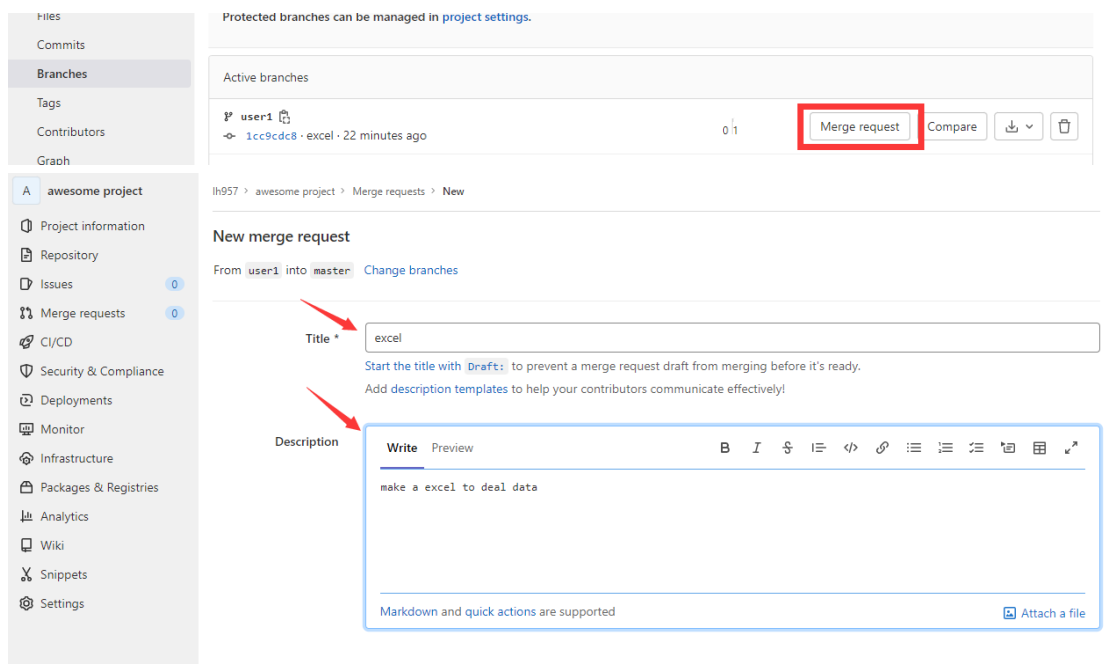


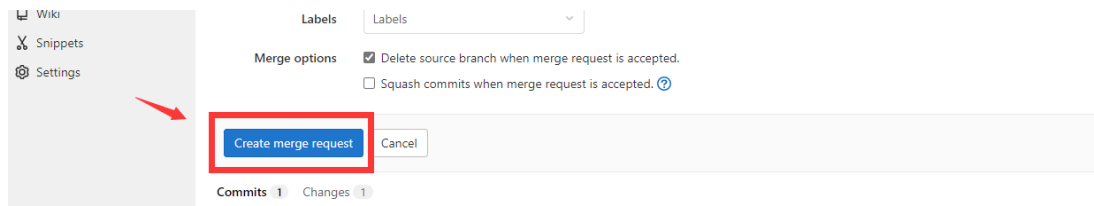
OK, 现在建立自己的分支, 然后提交到远端仓库, 命令如下: `git checkout -b user1`>>`git checkout user1`>>修改自己本地文件 (调试)>>`git add -A` (或者 `git add <指定文件>`)>>`git commit -m "提交说明"`>>`git push -u origin user1`



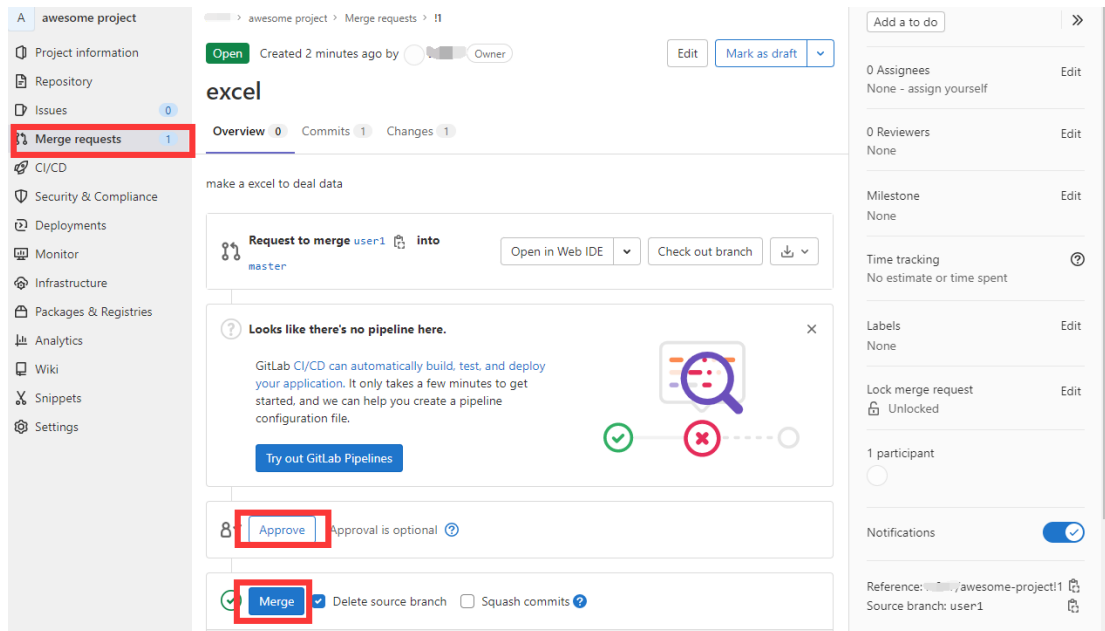
意见采纳

用户 user1 提交了本地的修改文件, 并在远端仓库提交了修改意见

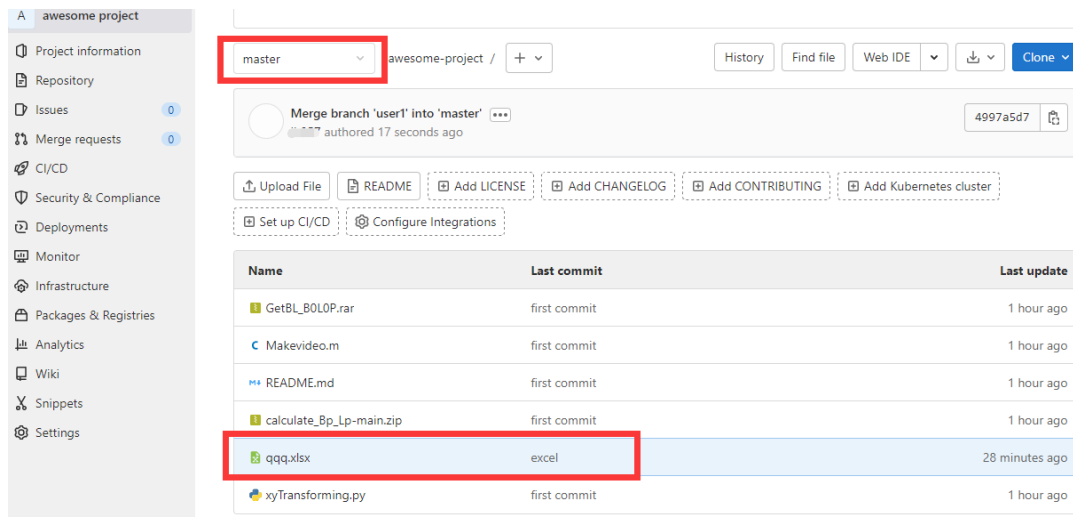




主分支 master 确定是否需要此修改



处理完成后，master 就有更新的文件了



其他

Git 的常用操作已经介绍完了，现给出几个另外的常用的命令：
`git pull` 将远端仓库下拉到本地中，`git log -stat` (或者 `git status`)，`git checkout <filename>` 将工作区文件打回去，一般用不到，`git reset HEAD^1` 提交撤回，`git remote -v` 查询所连接的远程仓库，`git remote remove origin` 断开连接本地仓库与远程仓库

耶~

更多学习，进这里：[百度一下，你就知道 \(baidu.com\)](http://baidu.com)

*#####*有误提交至 2874713957@qq.com*#####*