


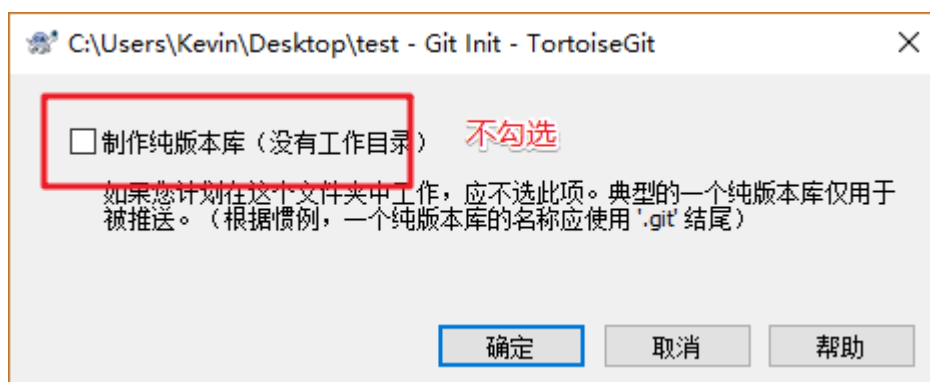
1. 创建本地版本库
  - 1.1 创建本地版本库
  - 1.2 工作区和暂存区
  - 1.3 添加新文件到暂存区并提交
2. 修改文件
  - 2.1 修改工作区文件
  - 2.2 还原修改
  - 2.3 查看修改历史
  - 2.4 差异比较
  - 2.5 删除文件
3. 案例: 添加一个本地项目到仓库
4. 添加远程仓库
  - 4.1 远程仓库的添加创建和删除操作
  - 4.2 本地仓库同步到远程仓库
  - 4.3 克隆远程仓库到本地
  - 4.4 ssh 配置
5. 管理分支
  - 5.1 创建分支
  - 5.2 切换分支
  - 5.3 合并分支
  - 5.4 解决冲突
6. 推送文件
7. 综合练习
  - 7.1 本地仓库相关操作:
  - 7.2 远程仓库相关操作

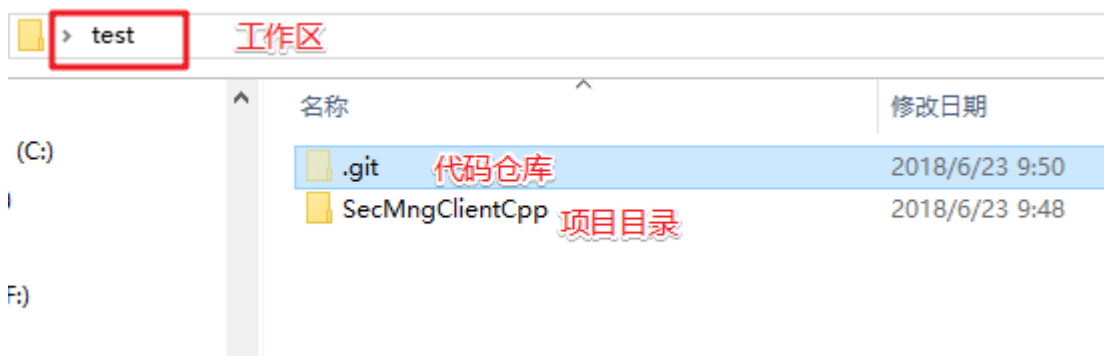
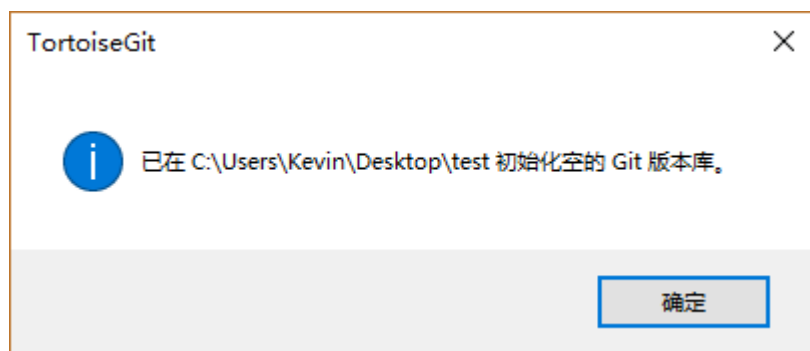
## 1. 创建本地版本库

### 1.1 创建本地版本库

创建一个目录如: test, 作为工作区, 在工作区中创建新版本库

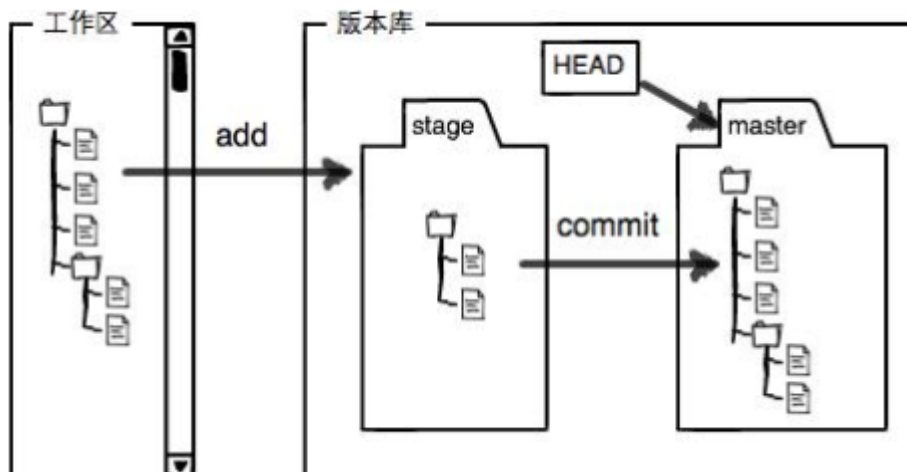
 Git 在这里创建版本库(Y)...





## 1.2 工作区和暂存区

在初始化git版本库之后会生成一个隐藏的文件 .git ，可以将该文件理解为git 的版本库 repository，而我们自己建立的项目文件夹即工作区 working directory，在 .git 文件夹里面还有很多文件，其中有一个 index 文件 就是暂存区也可以叫做 stage，git自动生成了一个分支 master 以及指向该分支的指针head，如下图



- 工作区: 存储项目文件的目录, 版本库需要创建到工作区中
- 版本库 - 创建出的隐藏目录 .git
  - stage - 暂存区
    - 当往工作区中添加了**新文件**之后, 需要将工作区文件添加到暂存区
      - 只需要做一次
  - master - 主分支
    - 默认只有这一个, 进行版本管理
    - HEAD - 操作master分支的指针
  - 暂存区和分支的关系

- 当暂存区的文件内容发送变化, 需要将其提交的master分支
- 只有提交之后才会形成一个节点(一个版本)

## 1.3 添加新文件到暂存区并提交

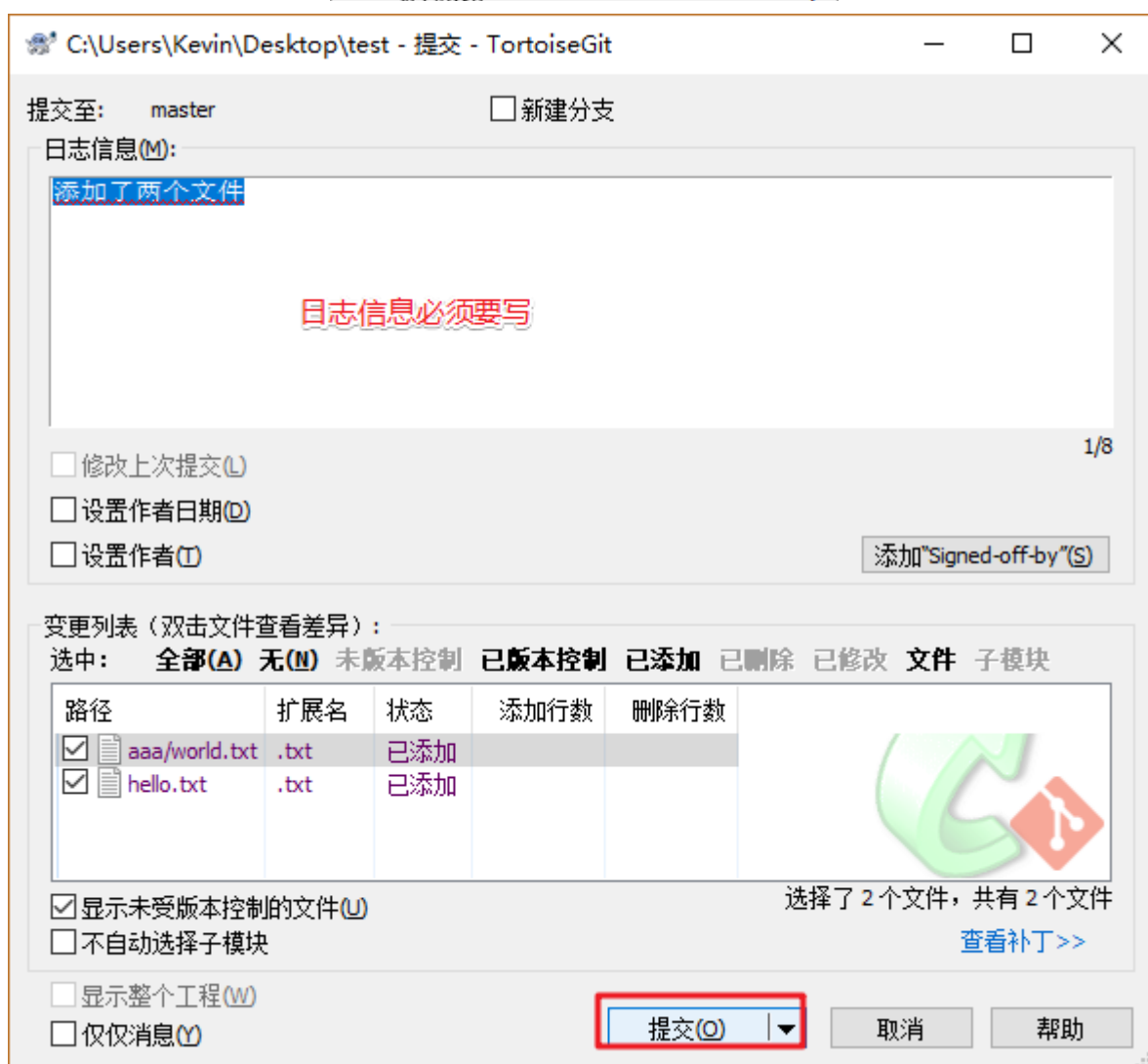
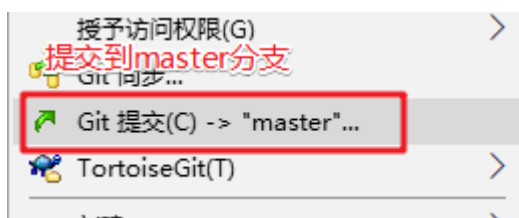
第一种方法:

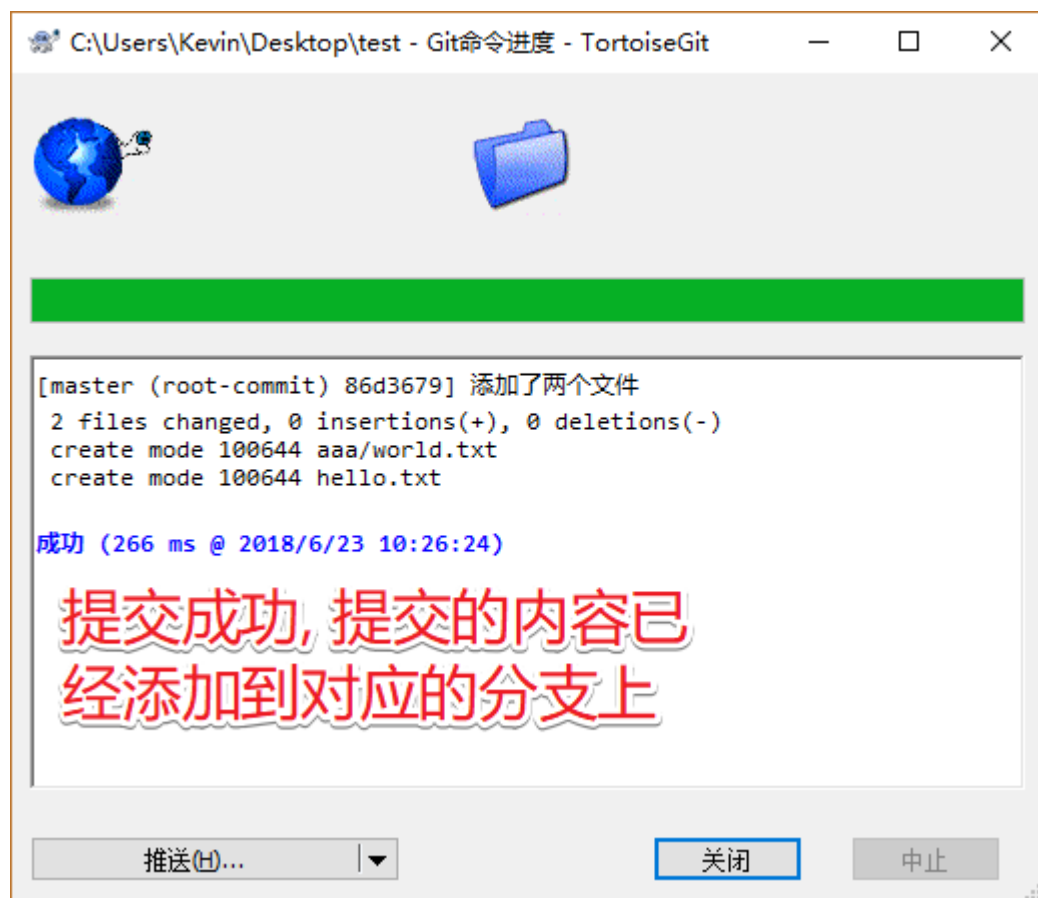
1. 在工作区创建一个新文件或者目录



后边直接next默认操作就可以了

2. 暂存区提交到分支





第二种方法:

- 新建好文件之后, 直接选择Git 提交也可以, 相当于将第一种方法的几步合成了一步.

## 2. 修改文件

### 2.1 修改工作区文件

- 直接修改文件, 然后提交即可

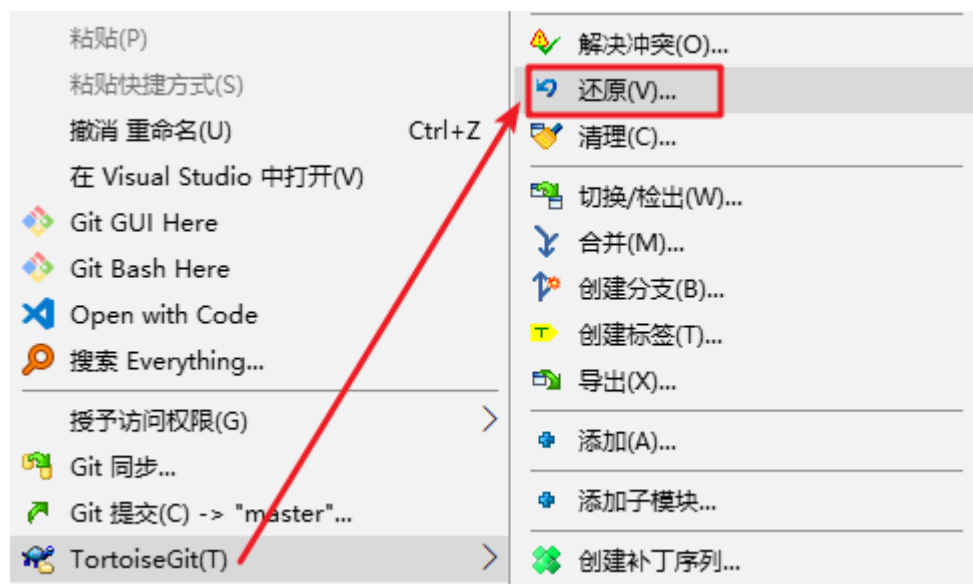
### 2.2 还原修改

1. 还原的本质:

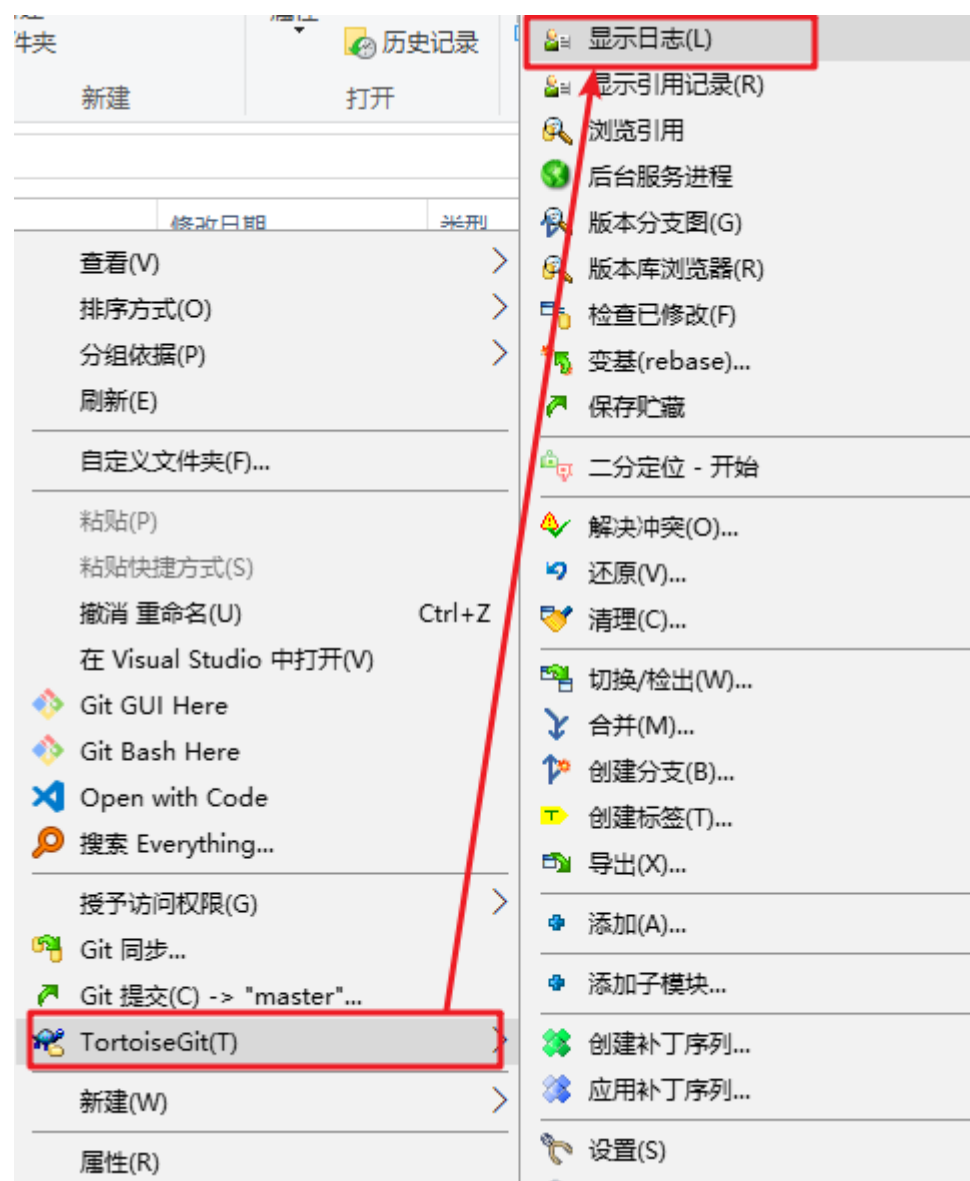
- 将工作区中修改的文件还原成**最后一个提交的版本**

2. 注意事项:

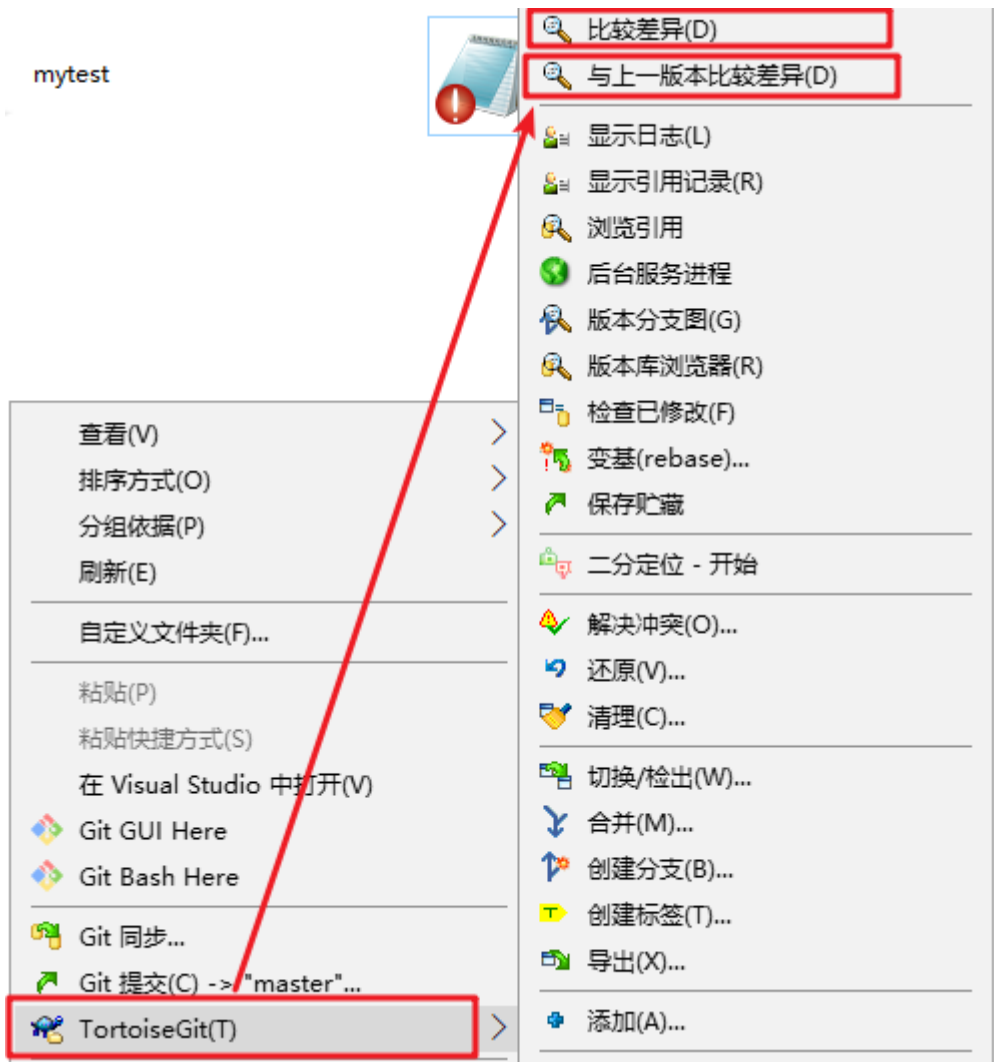
- 提交之前可以还原, 提交之后就无法通过该功能还原了
- 使用该还原功能之后, 文件内容就无法找回了
- 若工作区的文件不小心被删除, 可以通过该功能从版本库中重新还原文件, 但只能还原回最后一个版本.



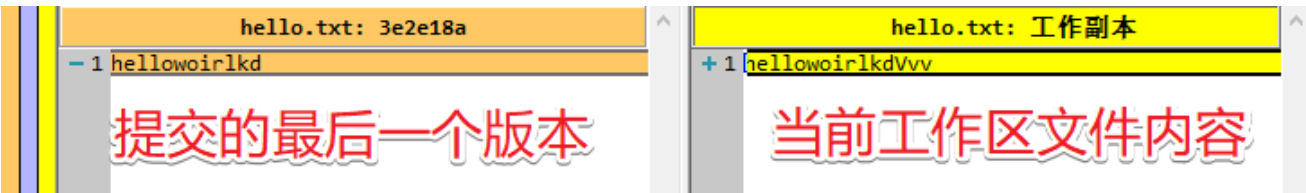
## 2.3 查看修改历史



## 2.4 差异比较



**比较差异:** 工作区的文件与最近提交的一个版本的差异



注意: 若修改后的文件已经提交, 则不能进行**差异比较**.

**与上一个版本比较差异:**

- 若修改文件后提交了, 则比较的是最后一个版本与上一次提交的版本之间差异.
- 若最后修改了文件没有提交, 则比较的是当前没有提交的文件和上一个版本的差异

## 2.5 删除文件

1. 第一种删除方式:
  - 右键单击鼠标然后直接使用del键删除
2. 使用删除功能

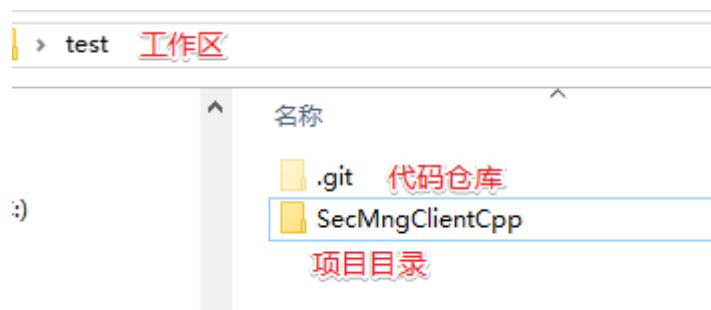


### 3. 注意事项:

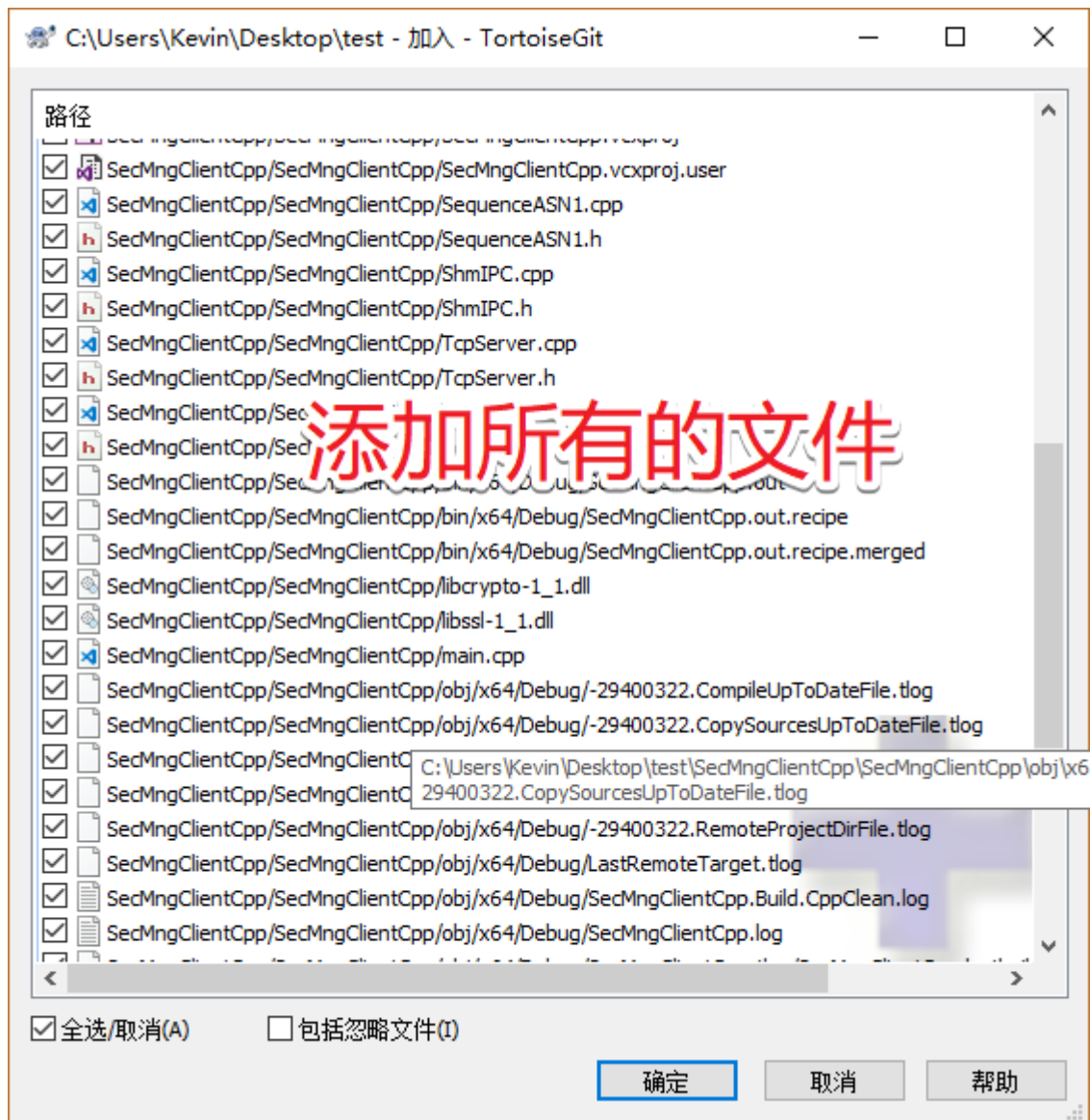
删除操作完成之后,必须要提交

## 3. 案例: 添加一个本地项目到仓库

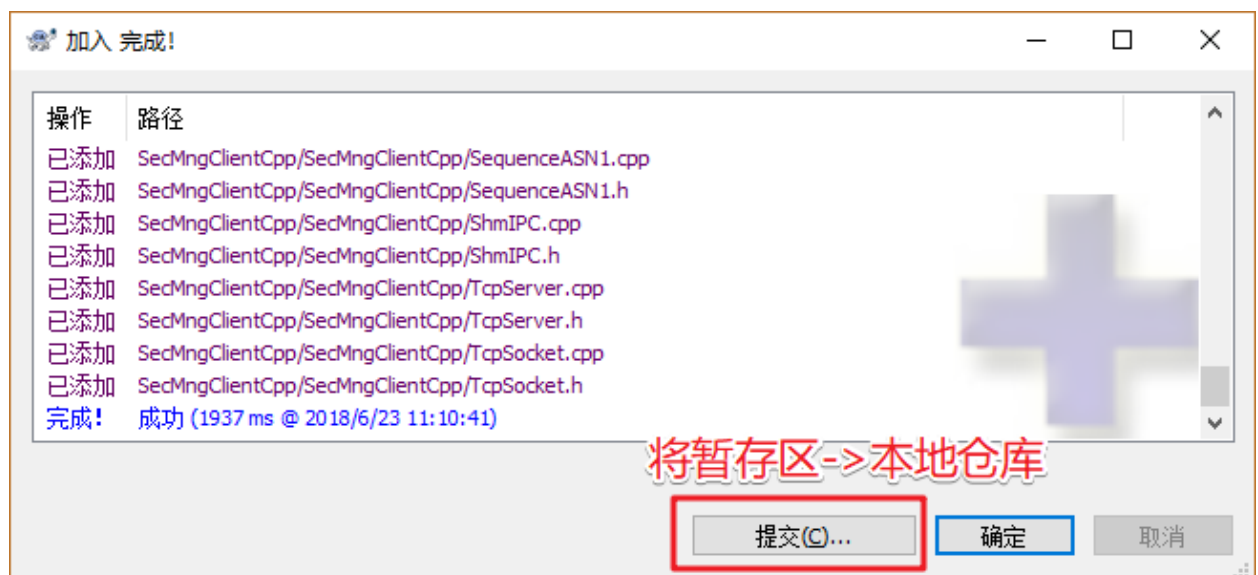
1. 创建一个工作区
2. 在工作区中添加一个代码仓库(版本库)
3. 将一个项目拷贝到工作区中



4. 将项目中所有的文件添加到暂存区



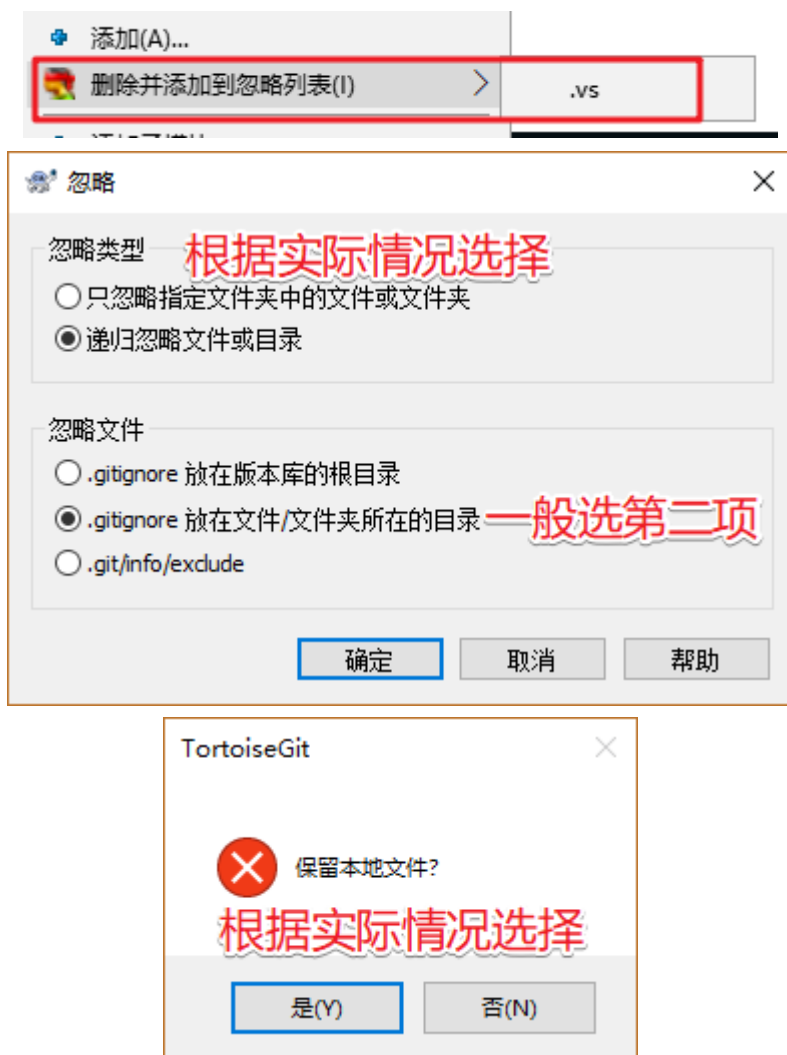
##### 5. 提交



##### 6. 设置文件忽略

在要忽略的文件上右键





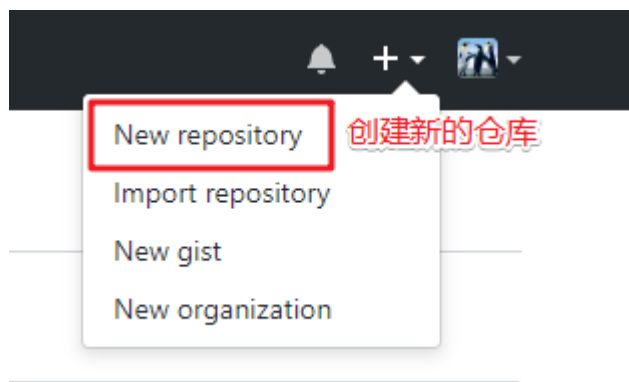
7. 提交修改

## 4. 添加远程仓库


在线代码托管平台

- Github: <https://github.com/>
- 码云: <https://gitee.com/>

### 4.1 远程仓库的添加创建和删除操作





Owner      Repository name

 subwen / MyTest **仓库名** ✓


Great repository names are short and memorable. Need inspiration? How about [verbose-journey](#).

Description (optional)

**可选项 - 描述**

- ☒  **Public**  
Anyone can see this repository. You choose who can commit.
- ☐  **Private**  
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. **不建议** you're importing an existing repository.

Add .gitignore: **None** ▼ | Add a license: **None** ▼ 

Create repository

快速设置 - 如果你以前做过这种事情 **访问该仓库的两种方式**

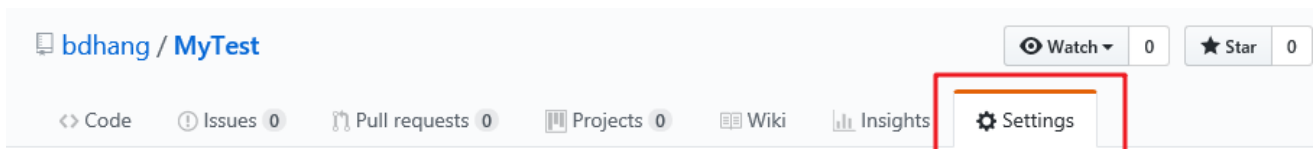
 在桌面上设置 要么 **HTTPS** **SSH** <https://github.com/subwen/MyTest.git>

我们建议每个存储库都包含一个 [README](#), [LICENSE](#)和[.gitignore](#)。

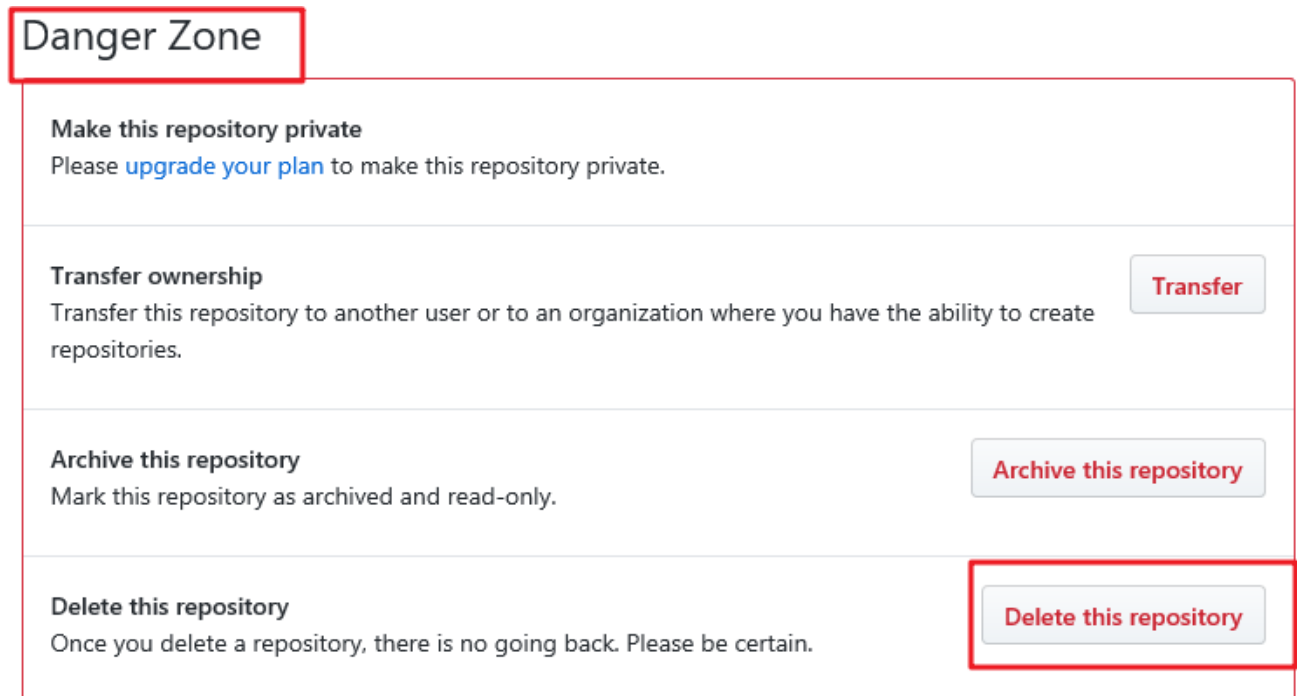
查看自己创建的远程仓库:

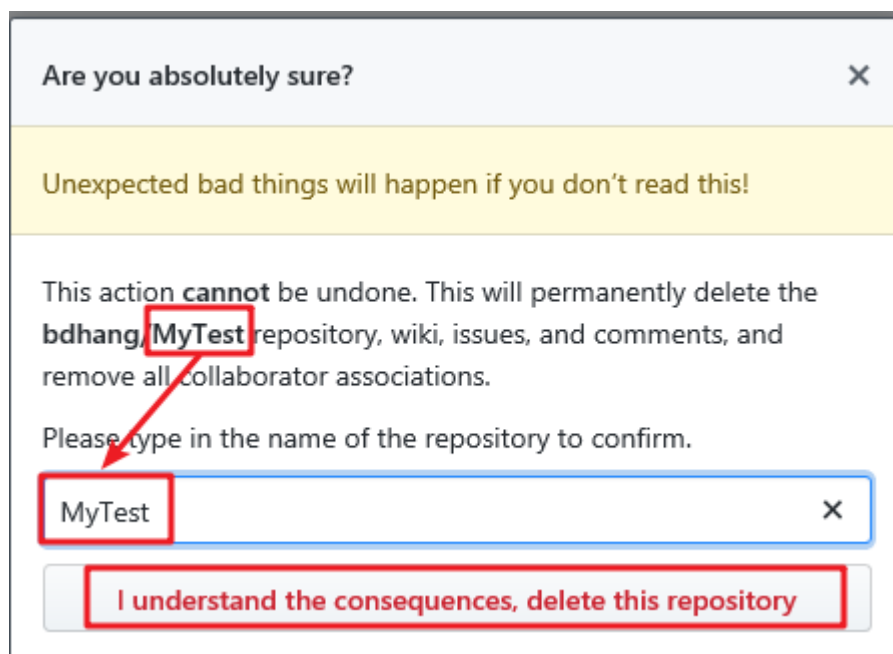


如何删除远程仓库?



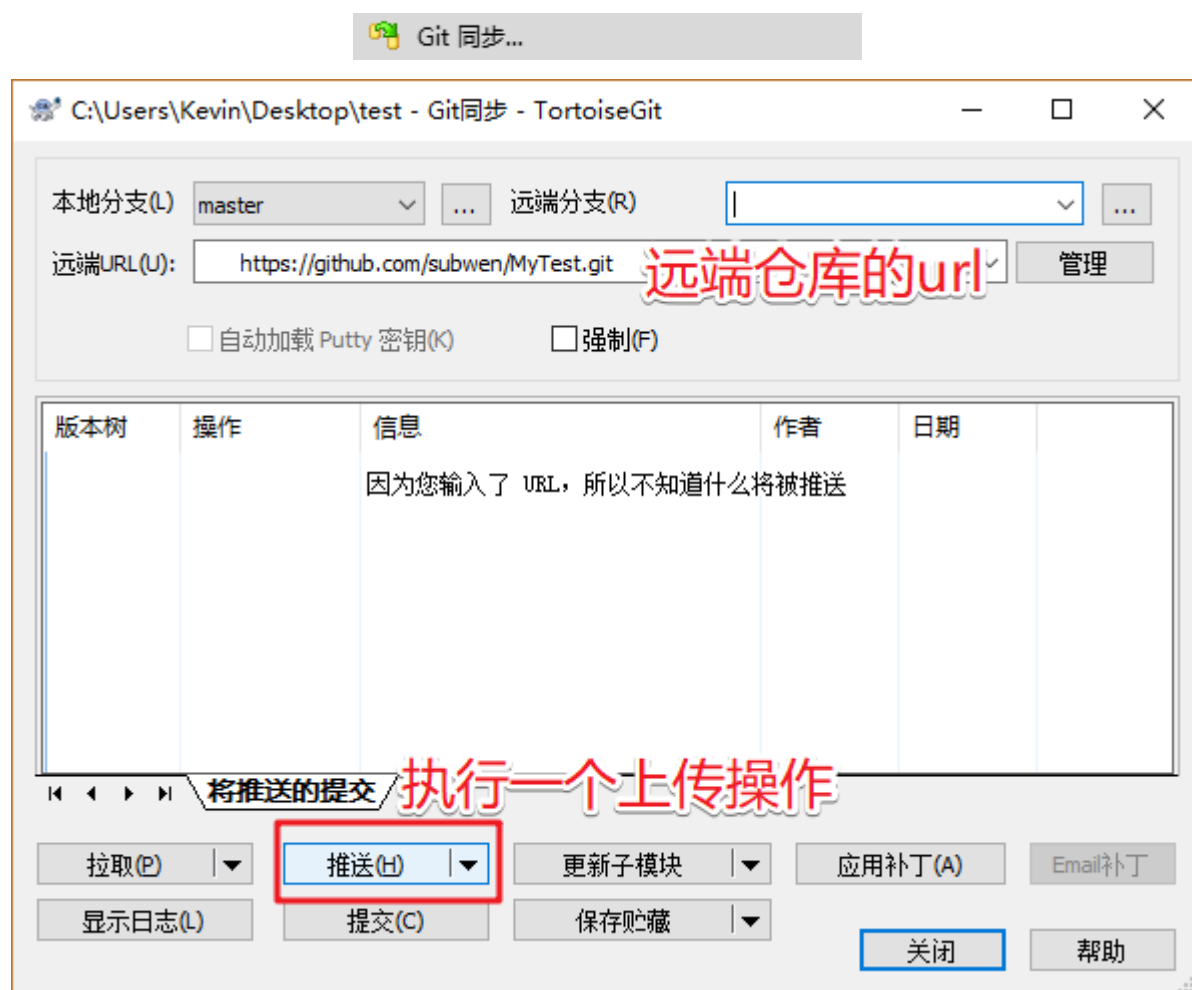
然后在最后面的如图所示:

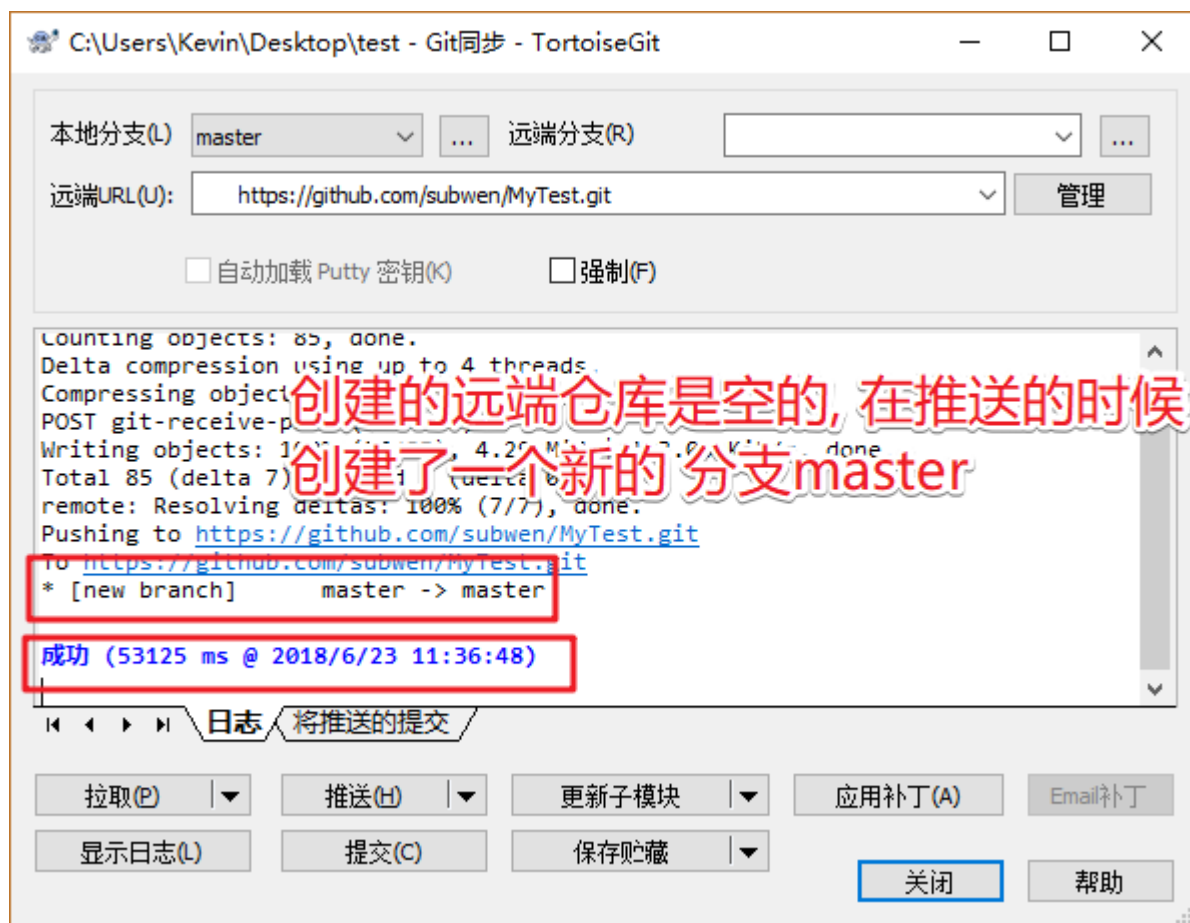




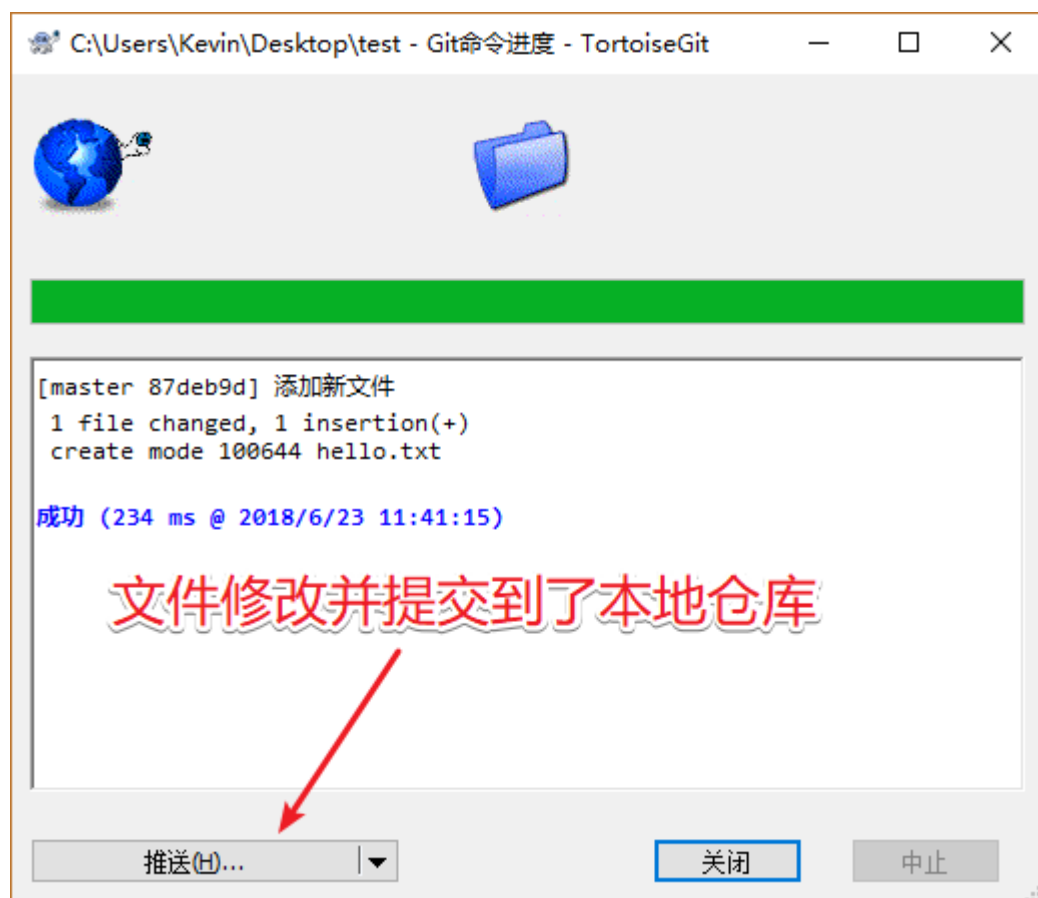
## 4.2 本地仓库同步到远程仓库

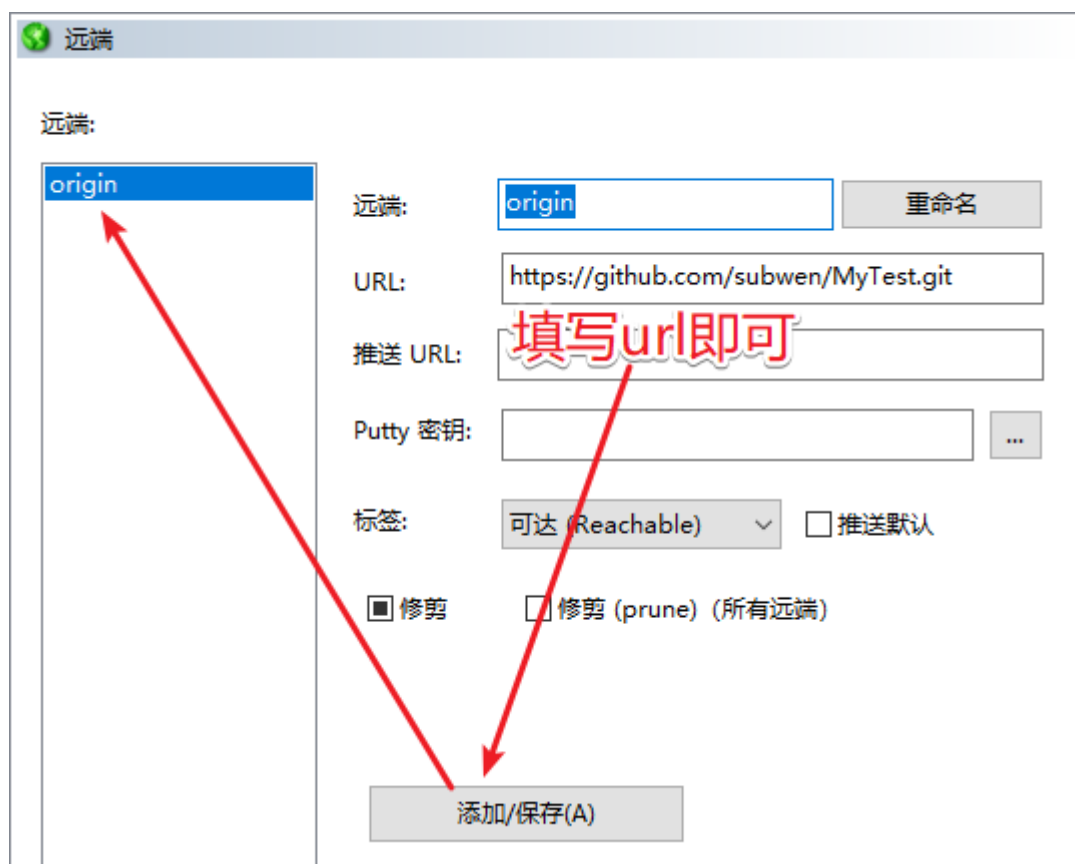
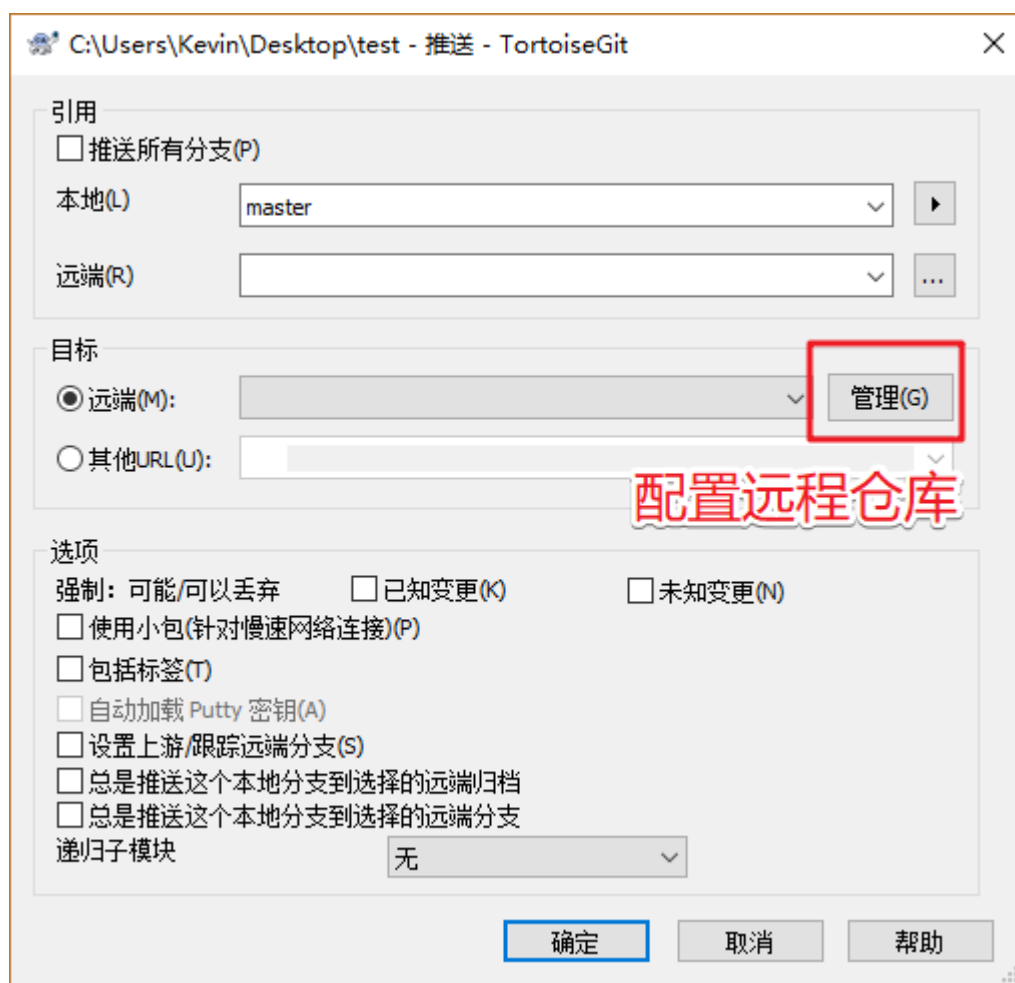
- 前提条件:
  - 创建的远程仓库是空的, 意味着还没有master分支





2. 修改了本地文件 -> 同步到远端仓库





C:\Users\Kevin\Desktop\test - 推送 - TortoiseGit

引用

☐ 推送所有分支(P)

本地(L)  ▶

远端(R)  ...

目标

☒ 远端(M):  管理(G)

☐ 其他URL(U):

选项

强制: ☐ 可能/可以丢弃 ☐ 已知变更(K) ☐ 未知变更(N)

☐ 使用小包(针对慢速网络连接)(P)

☐ 包括标签(T)

☐ 自动加载 Putty 密钥(A)

☐ 设置上游/跟踪远端分支(S)




☐ 总是推送这个本地分支到选择的远端归档

☐ 总是推送这个本地分支到选择的远端分支


递归子模块

确定 取消 帮助

C:\Users\Kevin\Desktop\test - Git命令进度 - TortoiseGit

Writing objects

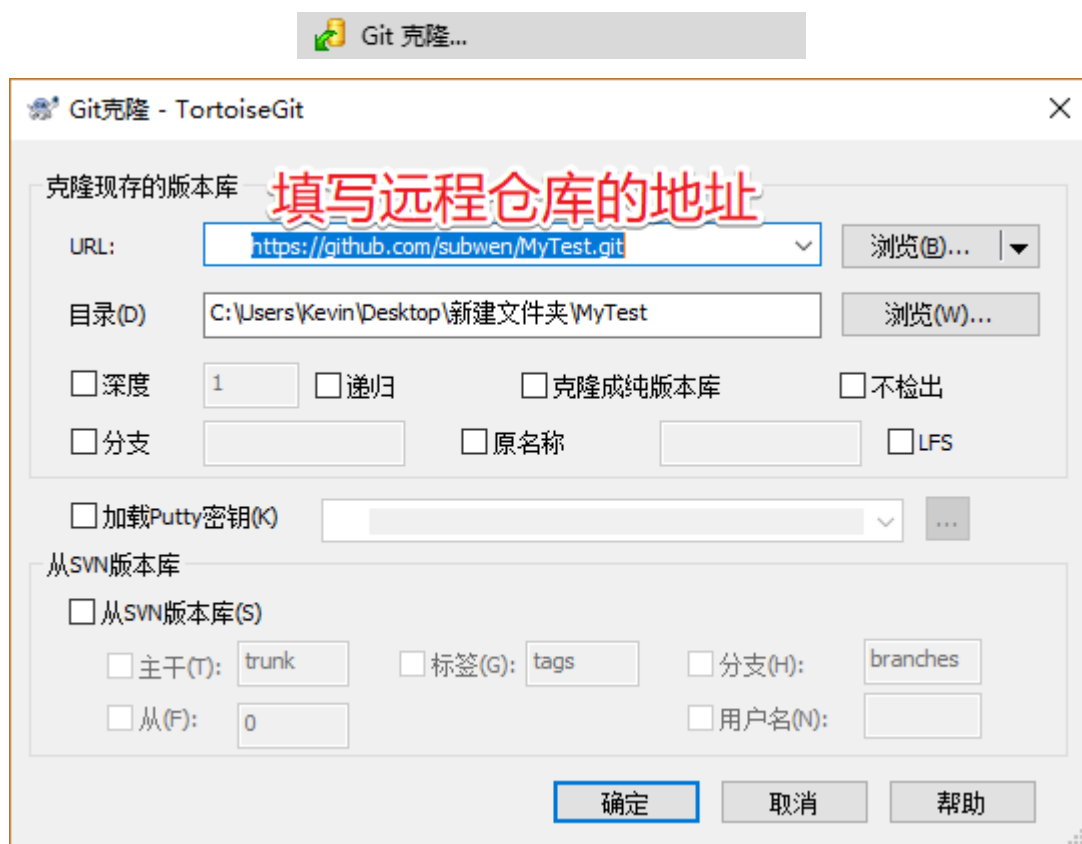


```
git.exe push --progress "origin" master:master  
  
Counting objects: 3, done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 306 bytes | 102.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0)  
To https://github.com/subwen/MyTest.git  
22f9a51..87deb9d master -> master
```

成功 (10922 ms @ 2018/6/23 11:45:18)

创建拉取请求(R) | 关闭 中止

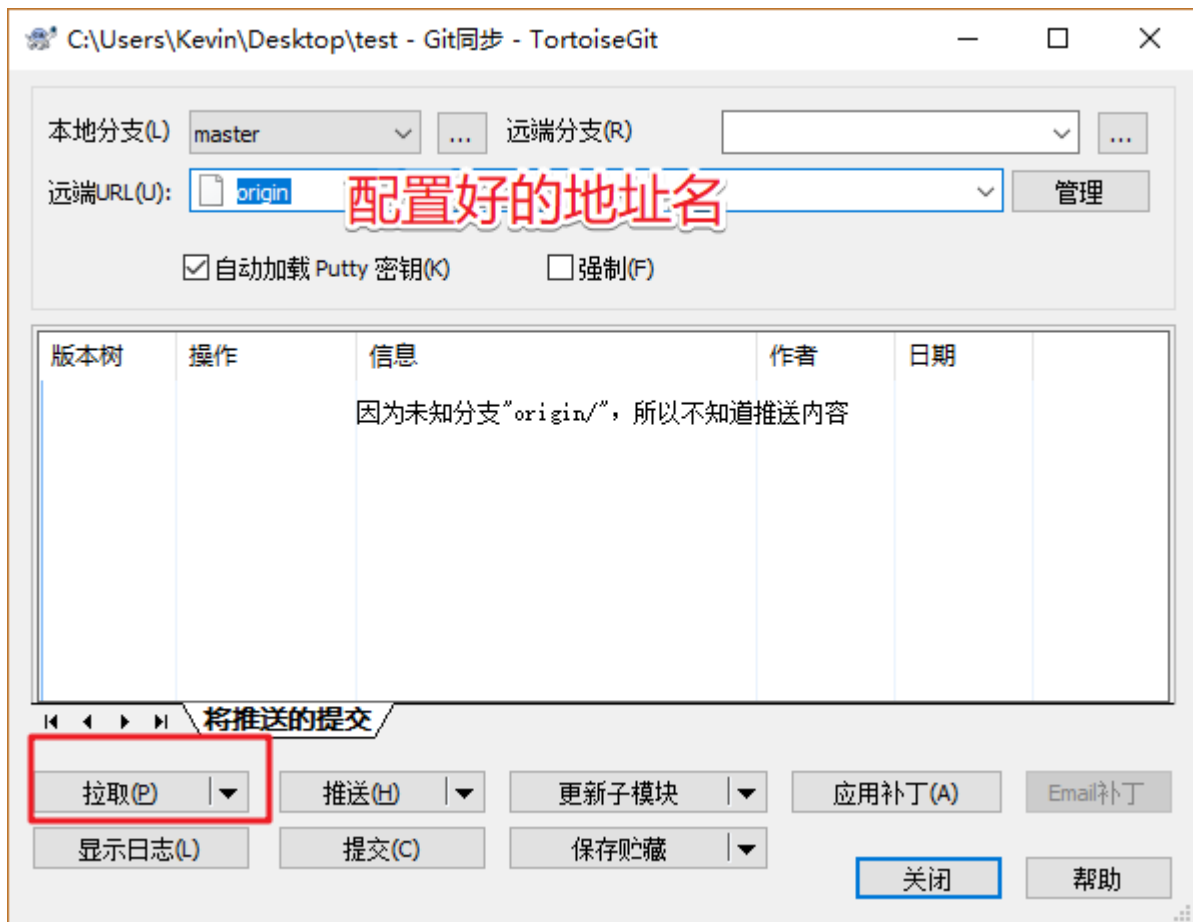
## 4.3 克隆远程仓库到本地



1. 从远程仓库将某个文件更新到本地







## 4.4 ssh 配置

1. 加密的方式是非对称加密, 需要生成一密钥对

- 客户端拿私钥
- github服务器拿公钥

2. 执行生成公钥和私钥的命令:

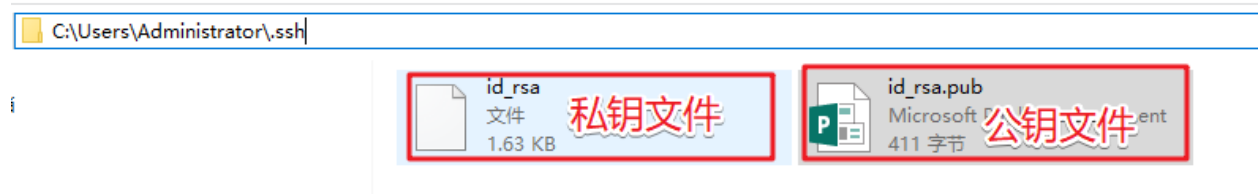
单击右键菜单, 选择[Git Bash Here], 然后执行ssh-keygen -t rsa

命令, 其中rsa为非对称加密算法

```
Kevin@X1-Carbon-2017 MINGW64 ~/Desktop
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Kevin/.ssh/id_rsa):
Created directory '/c/Users/Kevin/.ssh'.
Enter passphrase (empty for no passphrase): 直接回车 * 3
Enter same passphrase again:
Your identification has been saved in /c/Users/Kevin/.ssh/id_rsa.
Your public key has been saved in /c/Users/Kevin/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Kea+0y15Ci5B8hytu1j0y9DMoBk6QkCRPh5IHYqxdqQ Kevin@X1-Carbon-2017
The key's randomart image is:
+---[RSA 2048]-----+
|.O+O.|
|O=+|.
|BE . .
|++O O . .
|O.+ = 00 S
|O= ==O .
|* . =00. O
|.. Boo..+ O
| . *++O.+
+----[SHA256]-----+
Kevin@X1-Carbon-2017 MINGW64 ~/Desktop
$
```

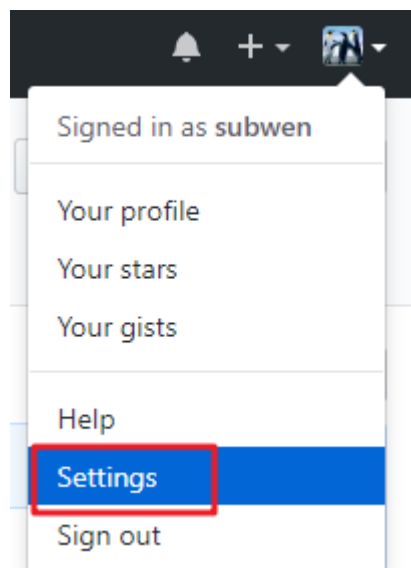
密钥对生成成功了

密钥对的位置:



使用ssh协议上传文件, 首先要在git服务端和本地分别配置公钥和私钥

1 将公钥添加到github上



Personal settings
Profile
Account
Emails
Notifications
Billing
SSH and GPG keys

## SSH keys

[New SSH key](#)

There are no SSH keys associated with your account.

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

可以通过这个链接查看生成密钥对的命令

## GPG keys

[New GPG key](#)

There are no GPG keys associated with your account.


Learn how to [generate a GPG key and add it to your account](#).

Personal settings
Profile
Account
Emails
Notifications
Billing
SSH and GPG keys

## SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



SSH

mykey

Fingerprint: 6a:88:6e:93:47:4a:13:d5:0a:e8:40:e9:f2:40:10:64

Added on 23 Jun 2018

Never used — Read/write

Delete

Personal settings
Profile
Account
Emails
Notifications
Billing
SSH and GPG keys

## SSH keys

[New SSH key](#)

There are no SSH keys associated with your account.

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

## GPG keys

[New GPG key](#)

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

## SSH keys / Add new

Title

newkey

Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

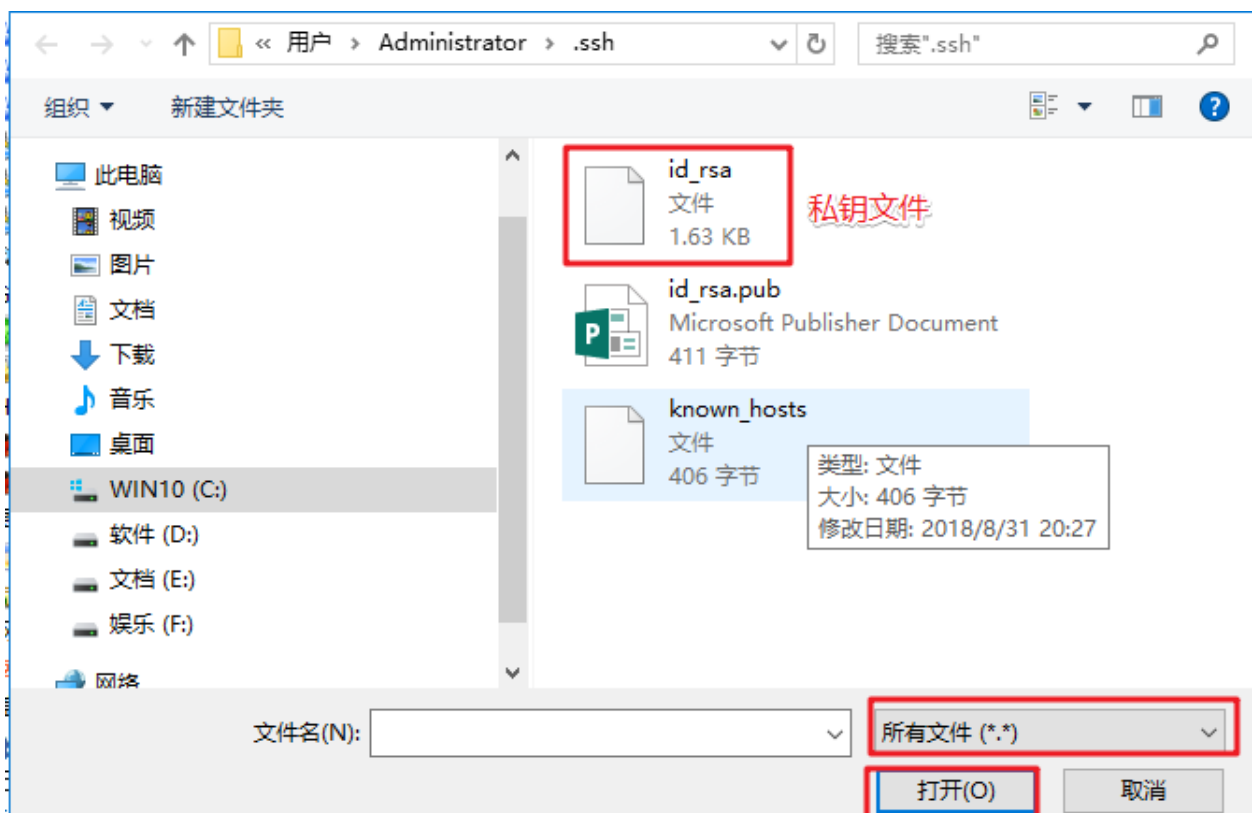
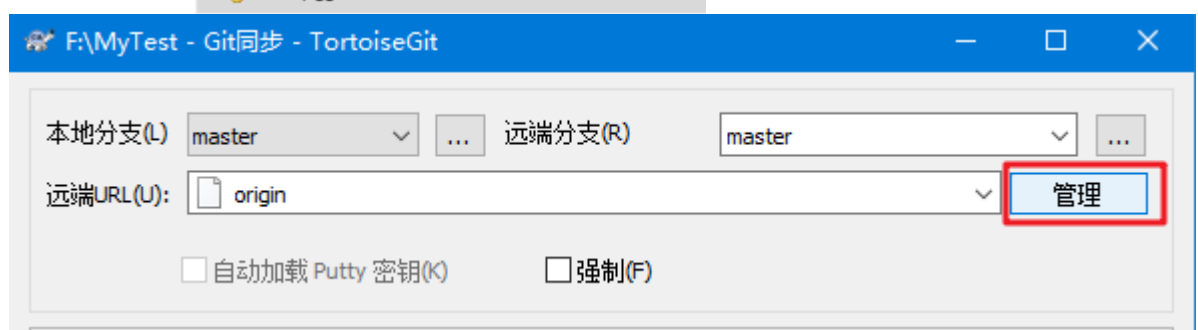
将公钥复制到这里

[Add SSH key](#)

点击该按钮

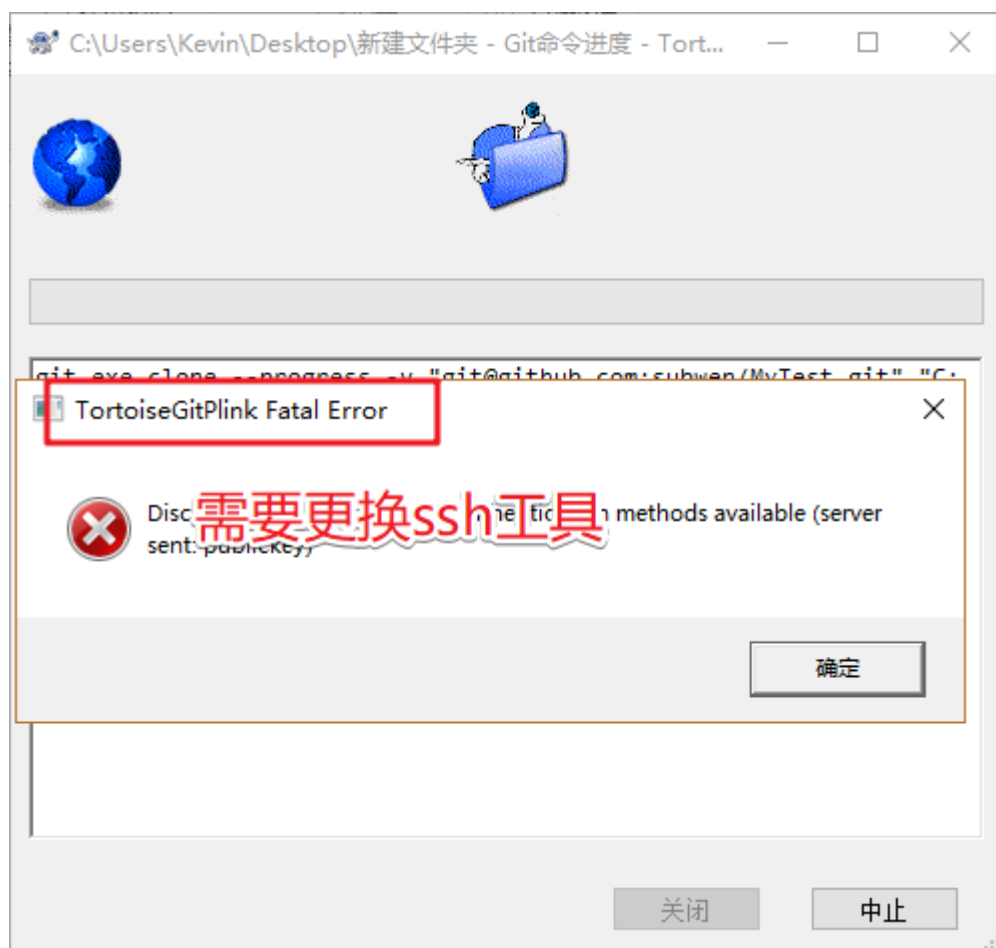
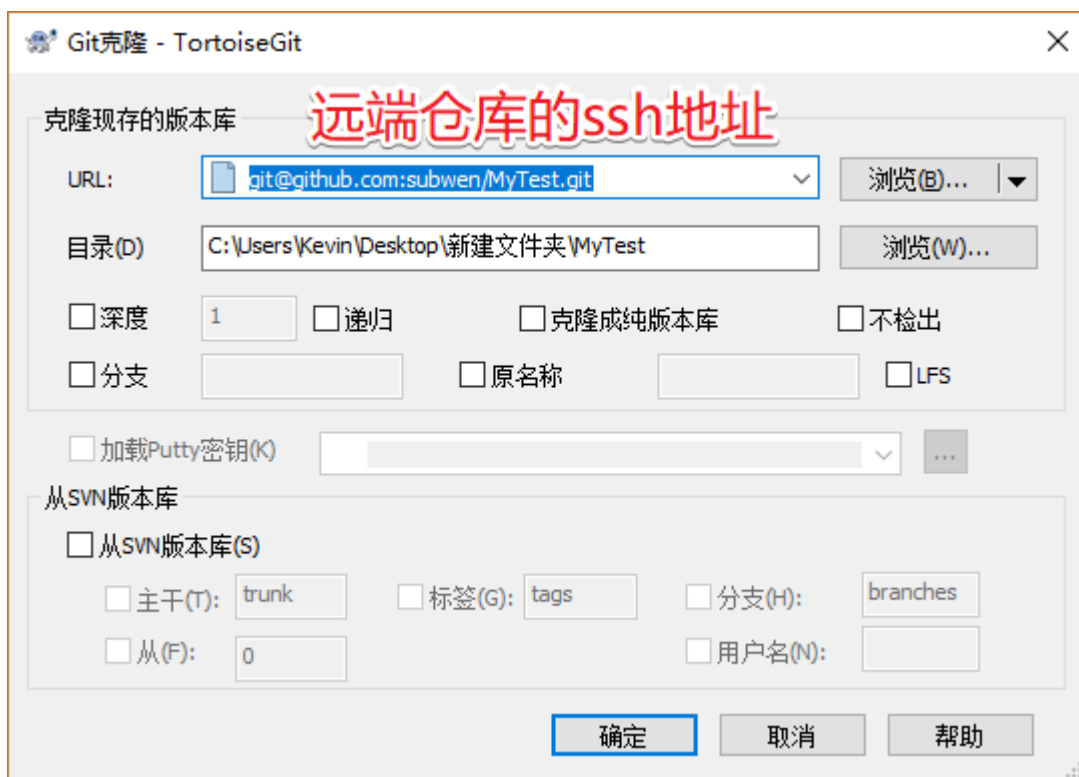
## 2 将私钥添加到本地

- 右键选择[同步]

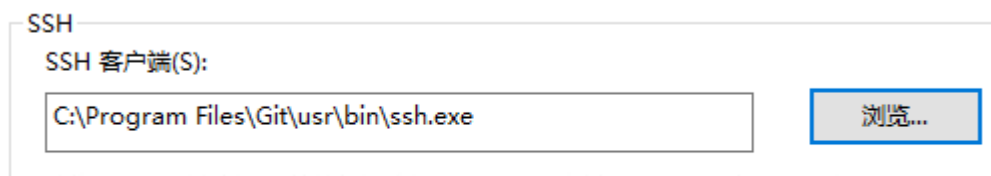


经过上面的两个步骤之后, 就可以使用ssh方式进行同步文件了.

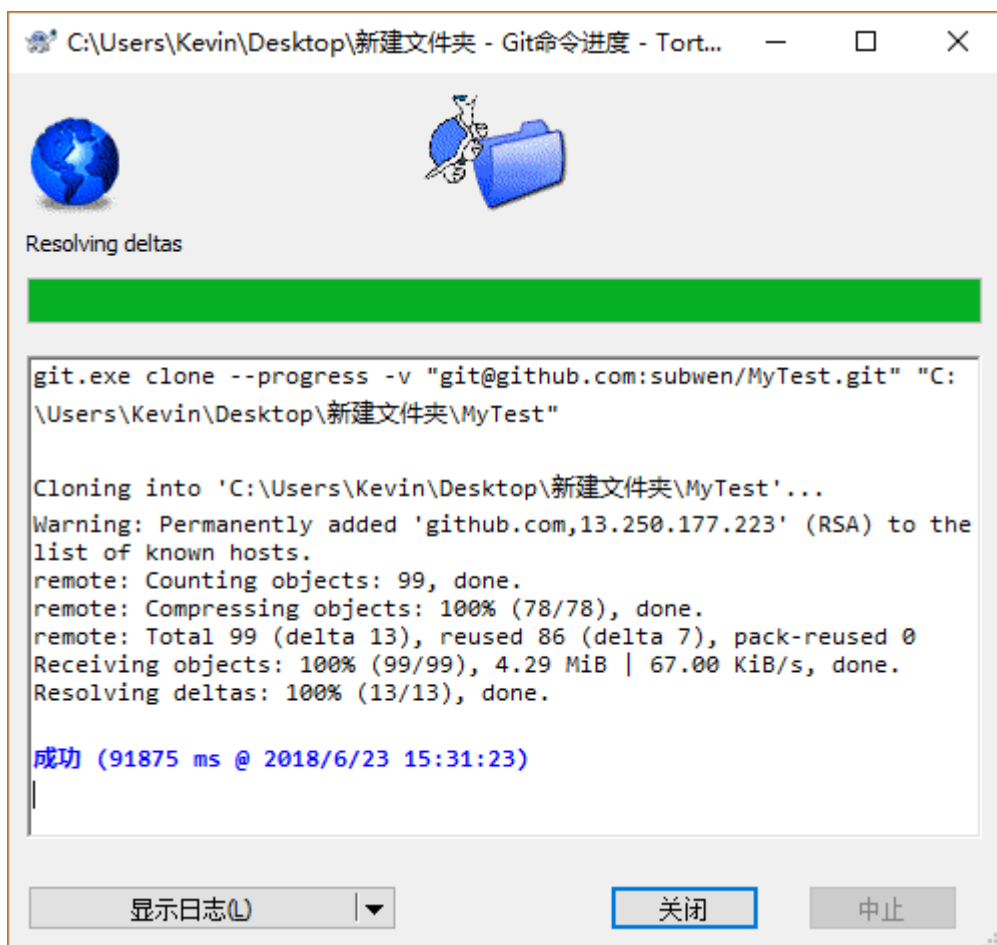
测试: 使用ssh的方式将远程仓库克隆到本地



更换ssh工具 - 打开小乌龟的设置窗口 -> 网络



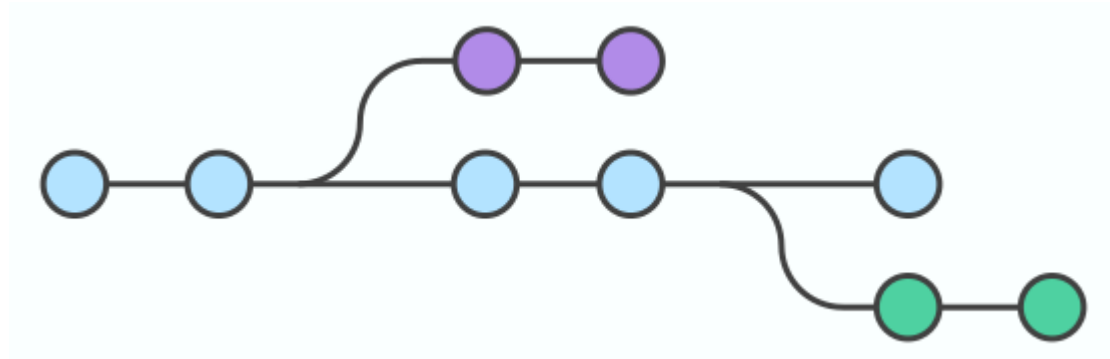
将默认的替换为git安装目录中ssh工具



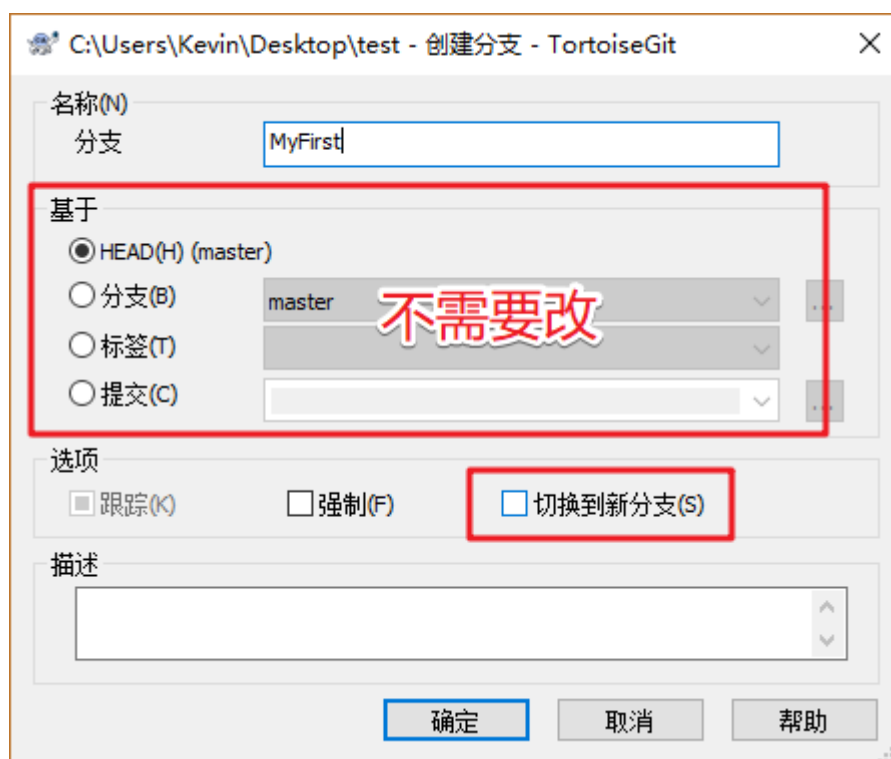
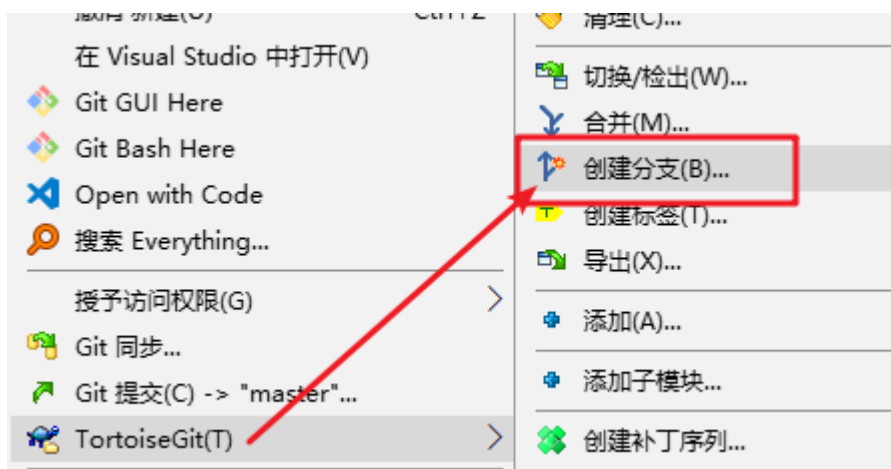
## 5. 管理分支

在git中默认只有一个分支 master

如果创建了分支, 各个分支都是独立的, 互不影响的

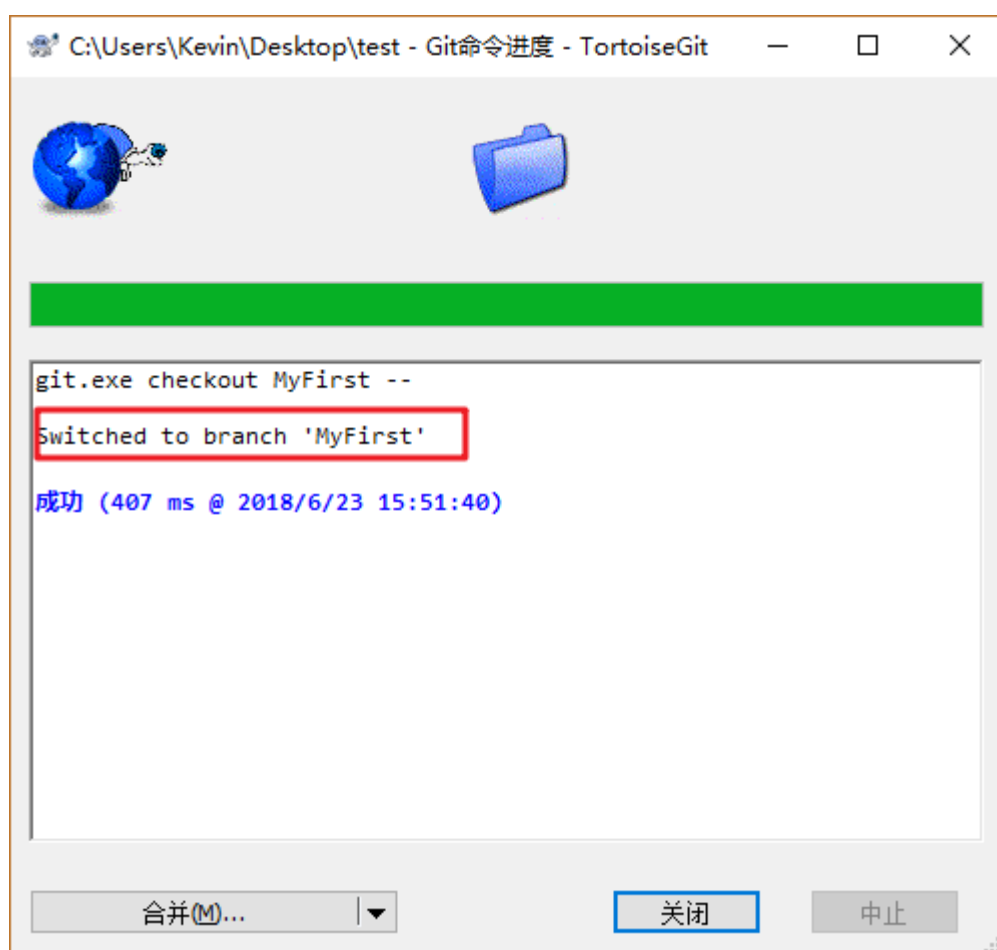
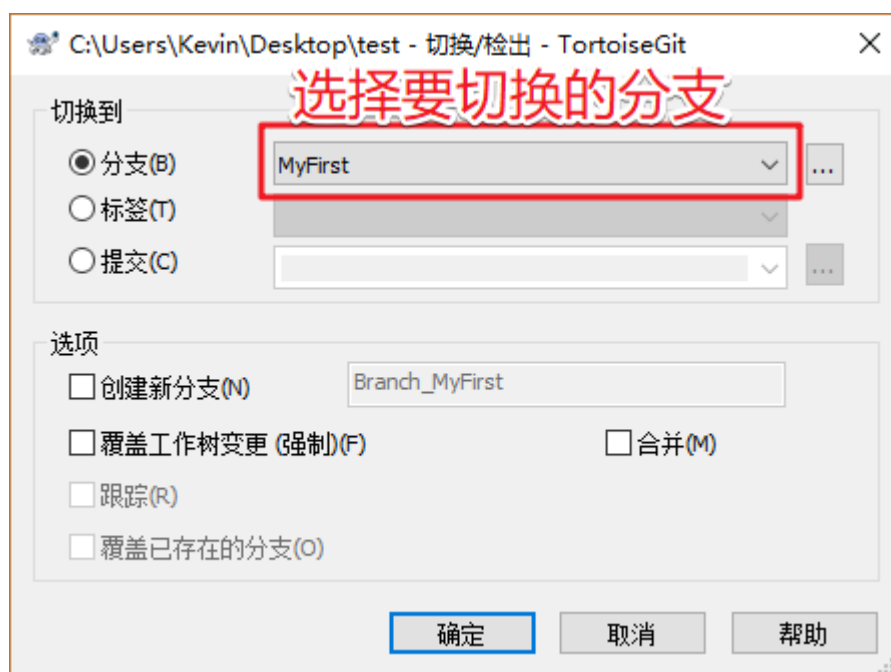


### 5.1 创建分支



## 5.2 切换分支



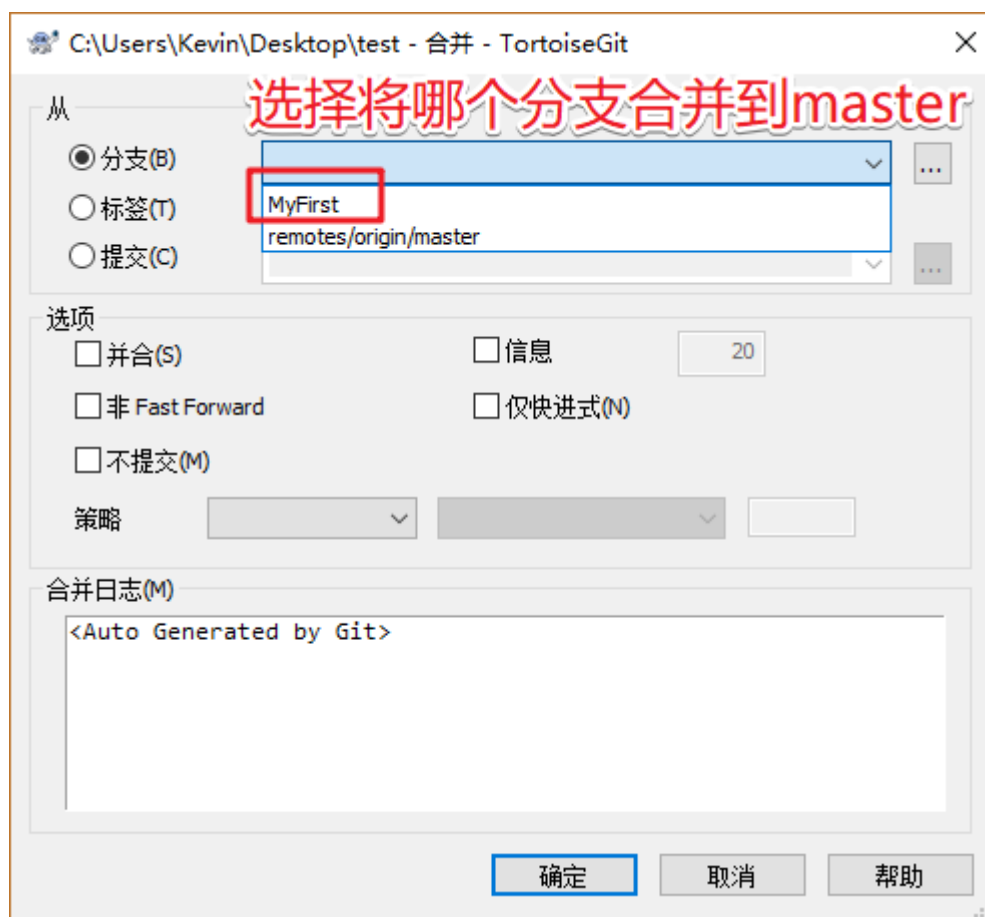
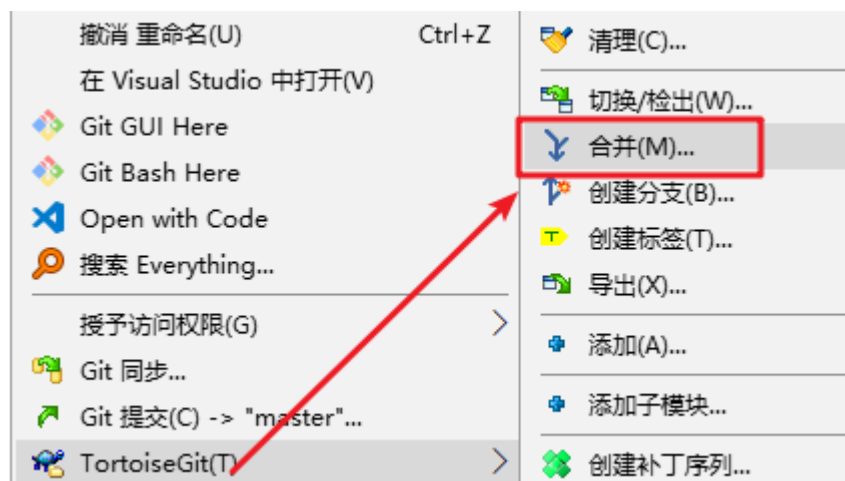


## 5.3 合并分支

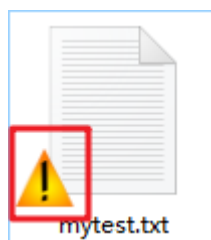
将添加的myFirst分支合并到master分支

- 先切换到master分支
- 合并分支





## 5.4 解决冲突



<<<<<< HEAD 冲突的开始标记  
我是master分支的文件  
===== 两部分内容的分隔线  
我是myfirst分支的一个文件  
>>>>>> MyFirst 冲突的结束

解决冲突(O)...

C:\Users\Kevin\Desktop\test - 提交 - TortoiseGit

提交至: master ☐ 新建分支

日志信息(M):

```
Merge branch 'MyFirst'
# Conflicts:
# mytest.txt
```

1/1

☐ 修改上次提交(L) ☐ 设置作者日期(D) ☐ 设置作者(T) 添加"Signed-off-by"(S)

变更列表 (双击文件查看差异):

选中: 全部(A) 无(N) 未版本控制 已版本控制 已添加 已删除 已修改 文件 子模块

路径	扩展名	状态	添加行数	删除行数
<input checked="" type="checkbox"/> aaa.txt	.txt	已添加	0	0
<input checked="" type="checkbox"/> mytest.txt	.txt	已修改	3	1

☒ 显示未受版本控制的文件(U) ☐ 不自动选择子模块

选择了 2 个文件, 共有 2 个文件 [查看补丁>>](#)

☐ 显示整个工程(W) ☐ 仅仅消息(Y)

提交(O) 取消 帮助

## TortoiseGit



看上去您的提交信息中有一个冲突提示（一行如 "# Conflicts:"）。此提示由 Git 为 cli 用户自动添加，不需要保留它。

您是否要忽略本警告并保留这些行，或者中止提交以编辑提交信息？

您可以在 TortoiseGit 设置中启用“剥离提交信息中以 # 开头的行”来自动移除这些行。

→ 忽略(I)

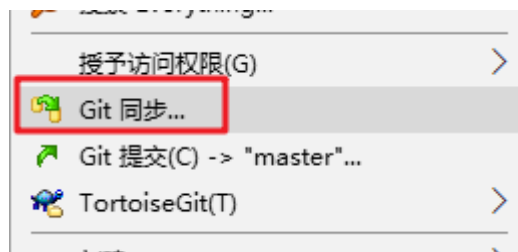
→ 放弃(A)

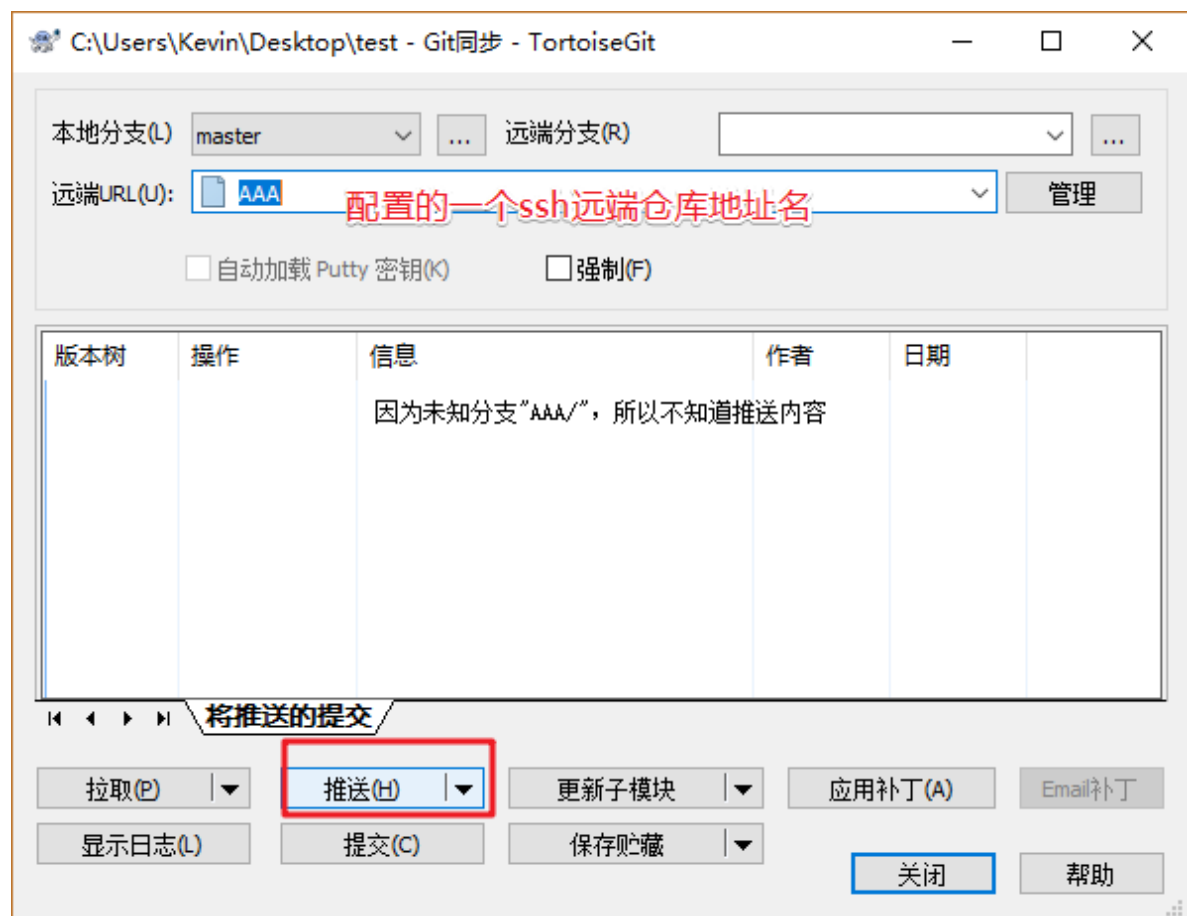
☐ 不再显示此消息(D)

## 6. 推送文件

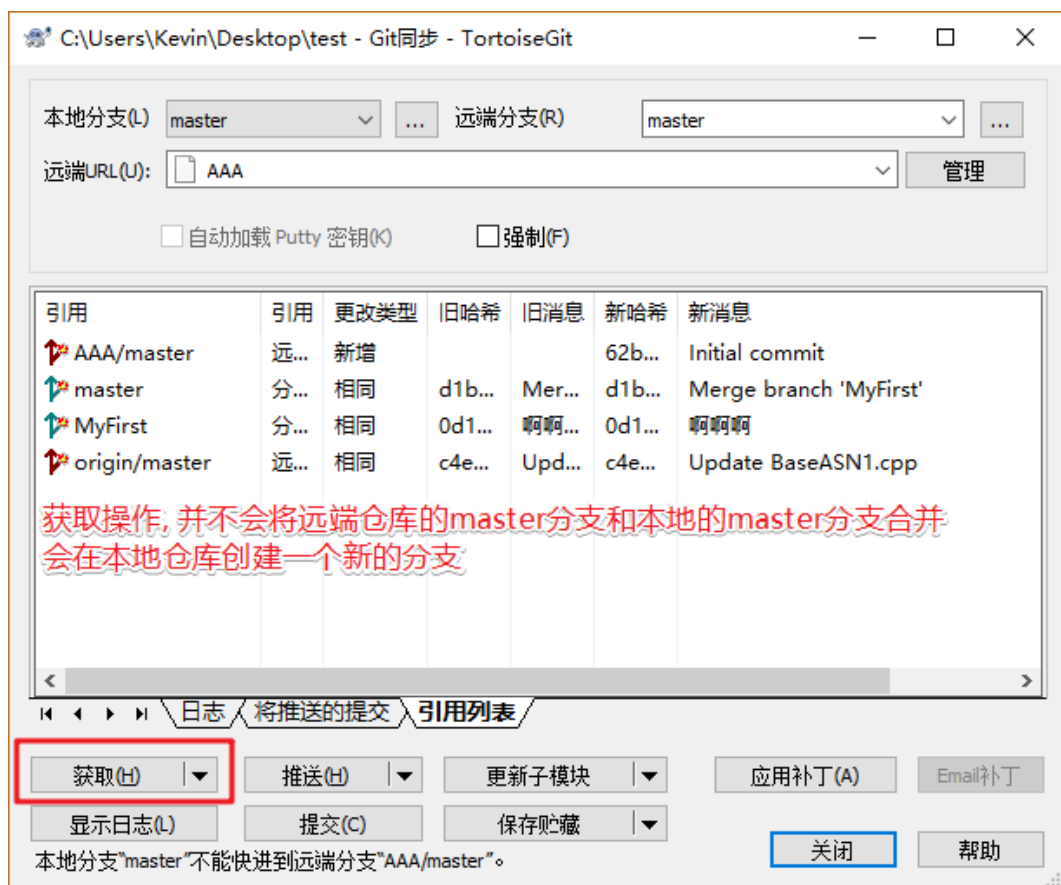
在本地有一个仓库, 管理了一个项目, 需要将本地仓库备份到远端仓库

- 在github/码云创建一个仓库, 创建的仓库并不是空仓库(在创建的时候指定readme就成了非空仓库)
- 将本地仓库文件推送到远程仓库
- 在本地工作区

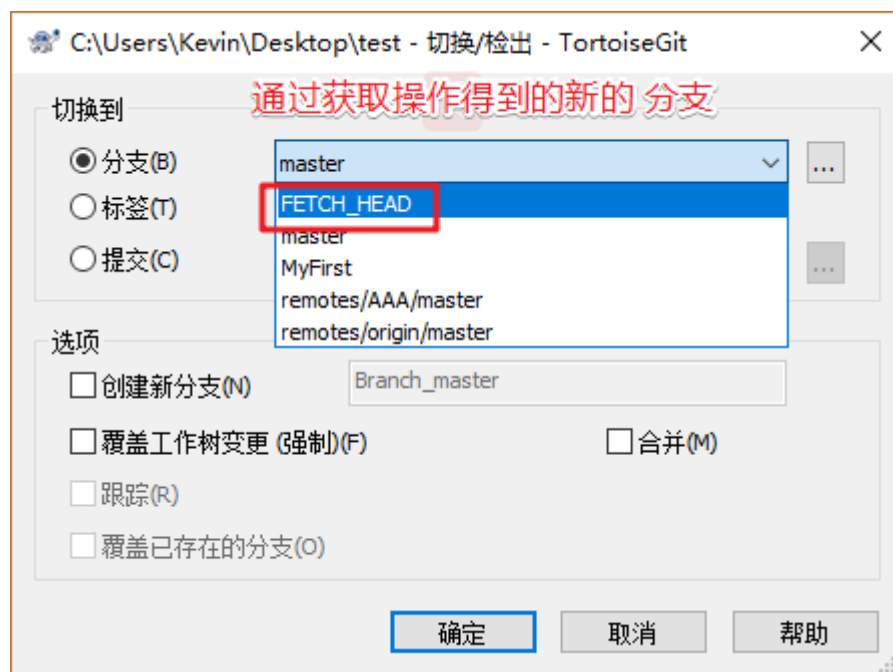




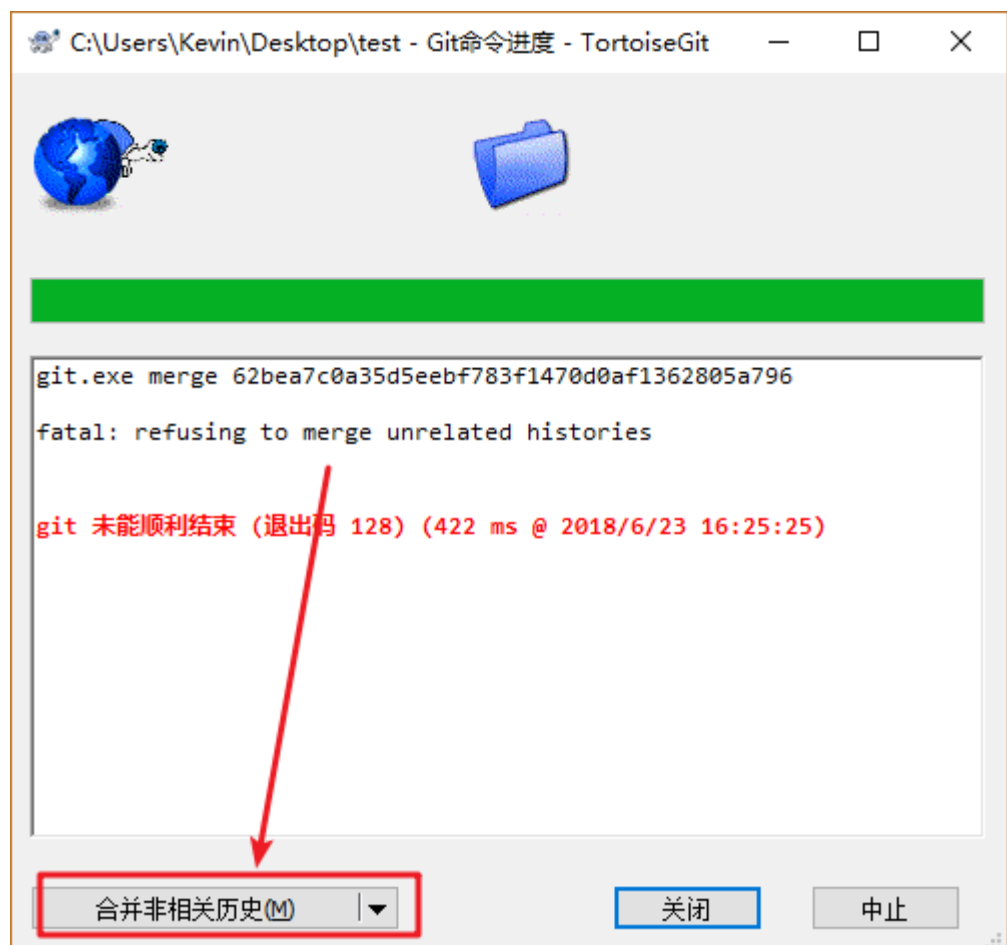
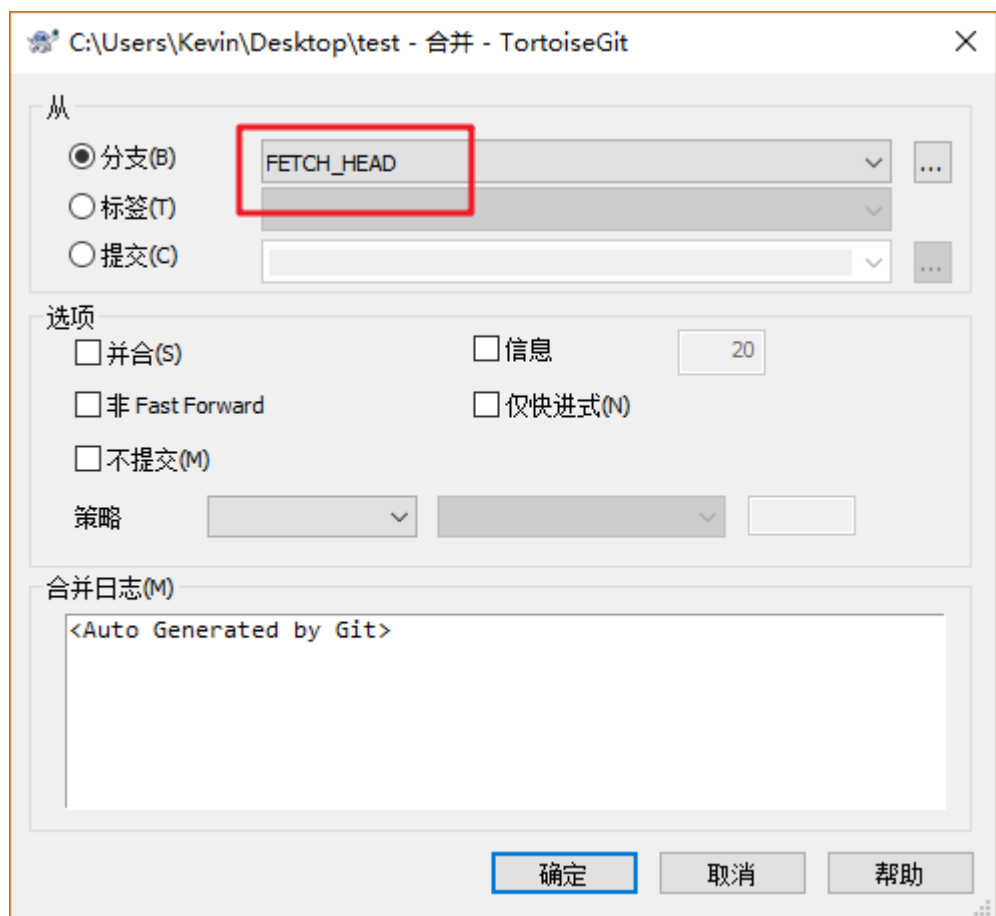
- 结果 - 推送失败
  - 首先需要先进行获取操作

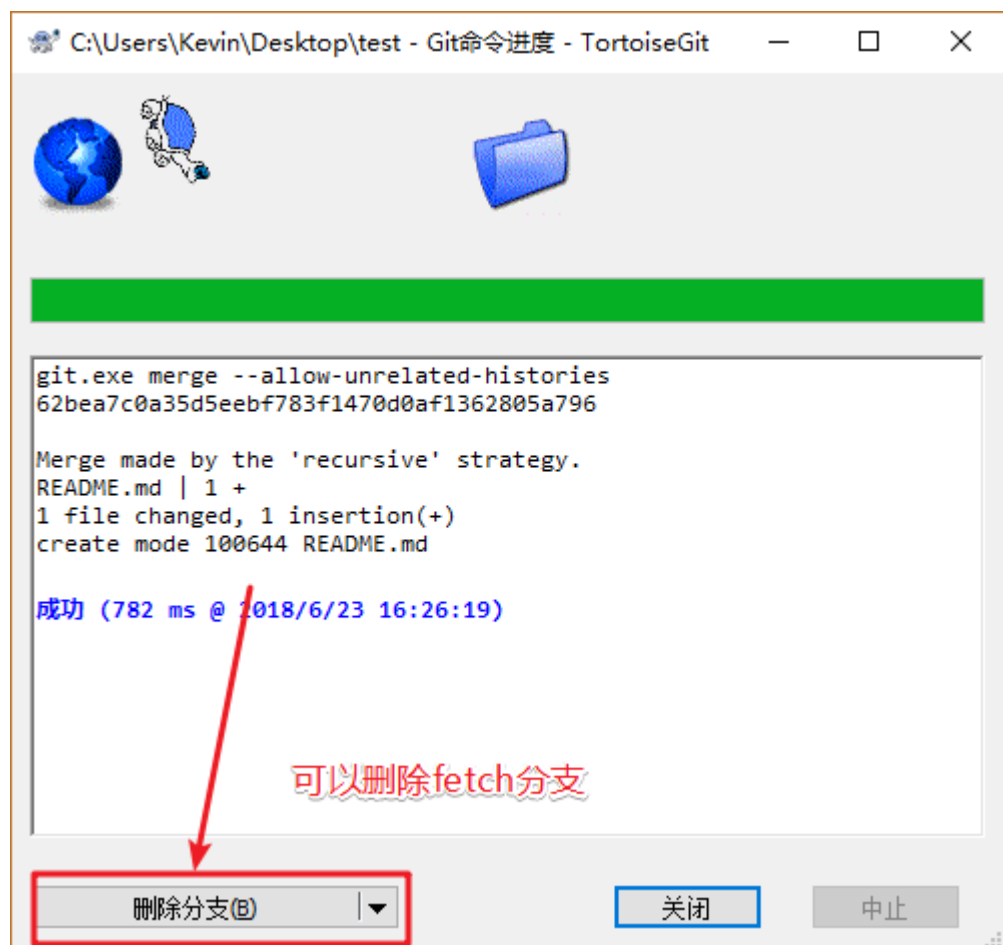


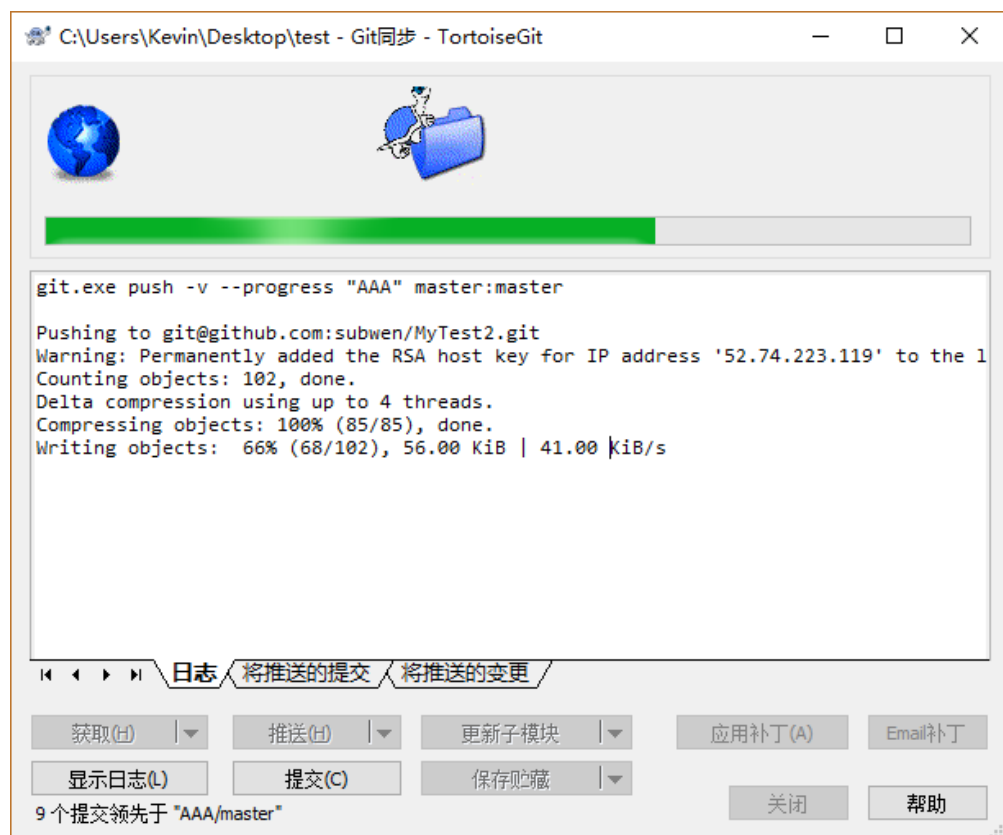
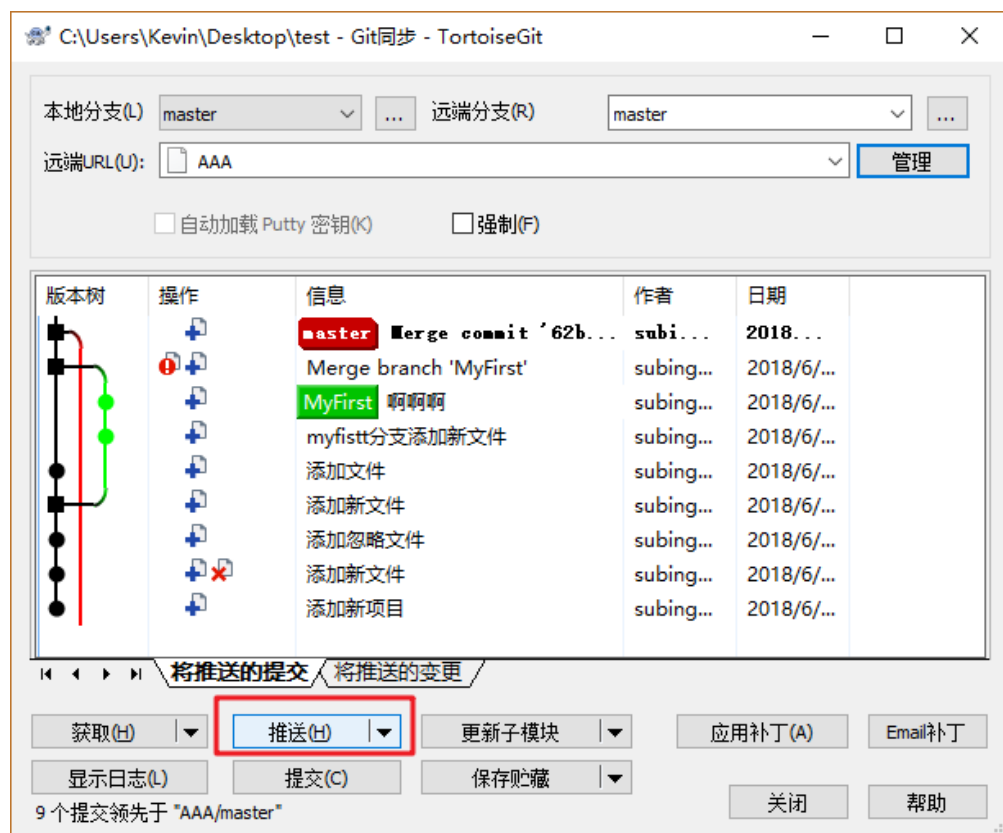
获取之后会将远端仓库的文件下载到本地的一个分支中, FETCH\_HEAD中  
可以切换到FETCH\_HEAD分支中;



- 将得到的fetch分支合并到master分支



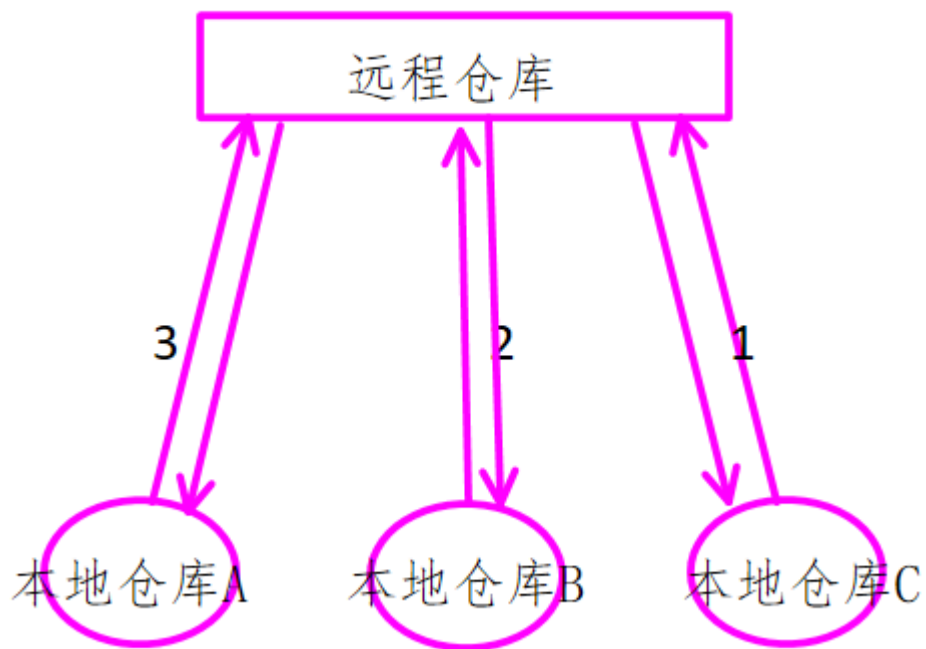




总结:

- 若远程仓库没有master分支, 则可以直接同步;
  - 若远程仓库有master分支, 则需要先将远程仓库的获取到本地仓库, 作为本地仓库的一个分支 FETCH\_HEAD, 然后进行合并, 最后合并完成之后就可以同步到远程仓库了.
- 本地仓库和远程仓库的同步





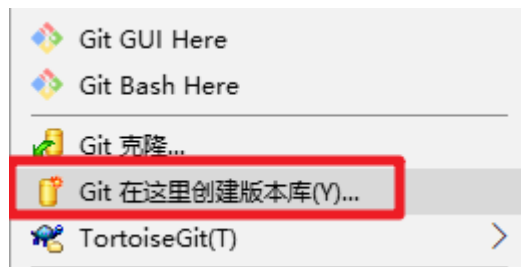
1. 本地修改文件
2. 将本地仓库文件推送到远程仓库
  - 很大的概率执行失败
3. 所以不会直接推动, 先做**获取**操作
  - 将远程仓库的文件直接和本地文件合并
    - 有可能发生冲突
    - 解决冲突
4. 最后在推送

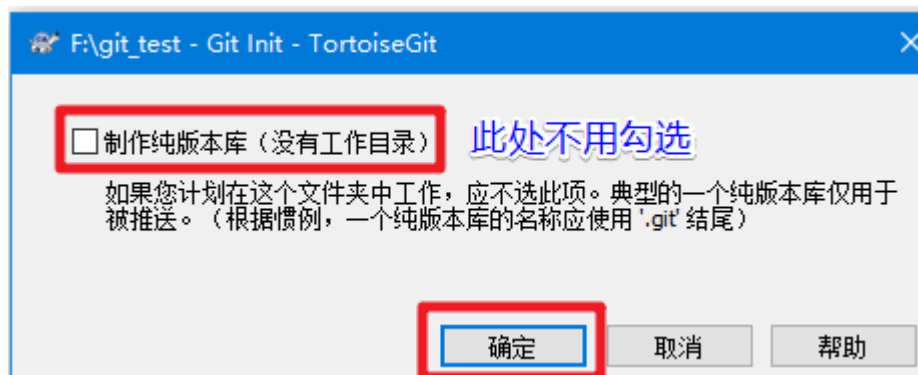
## 7. 综合练习

### 7.1 本地仓库相关操作:

创建步骤:

- 在本地创建一个工作目录
- 单击鼠标右键, 如下图所示:

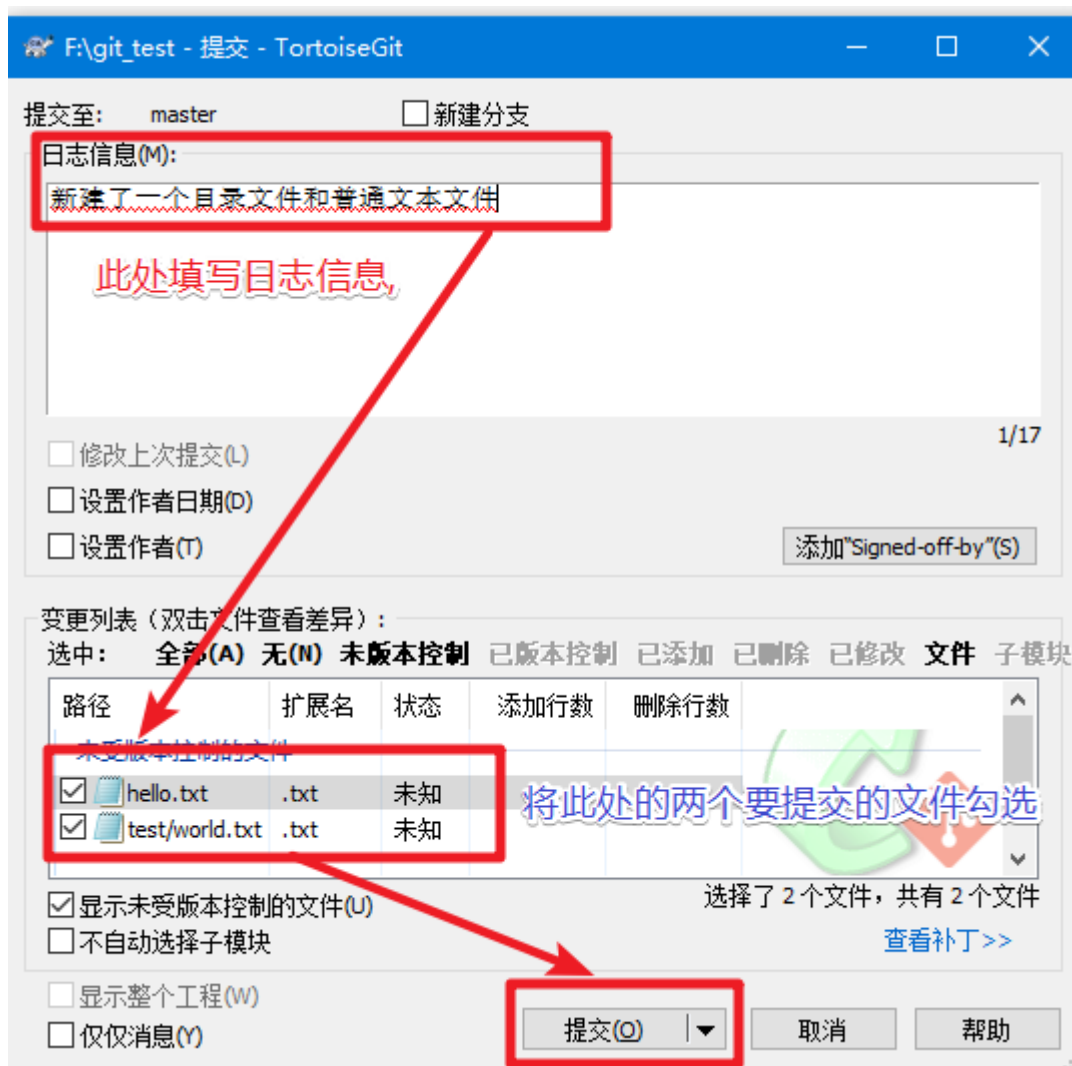




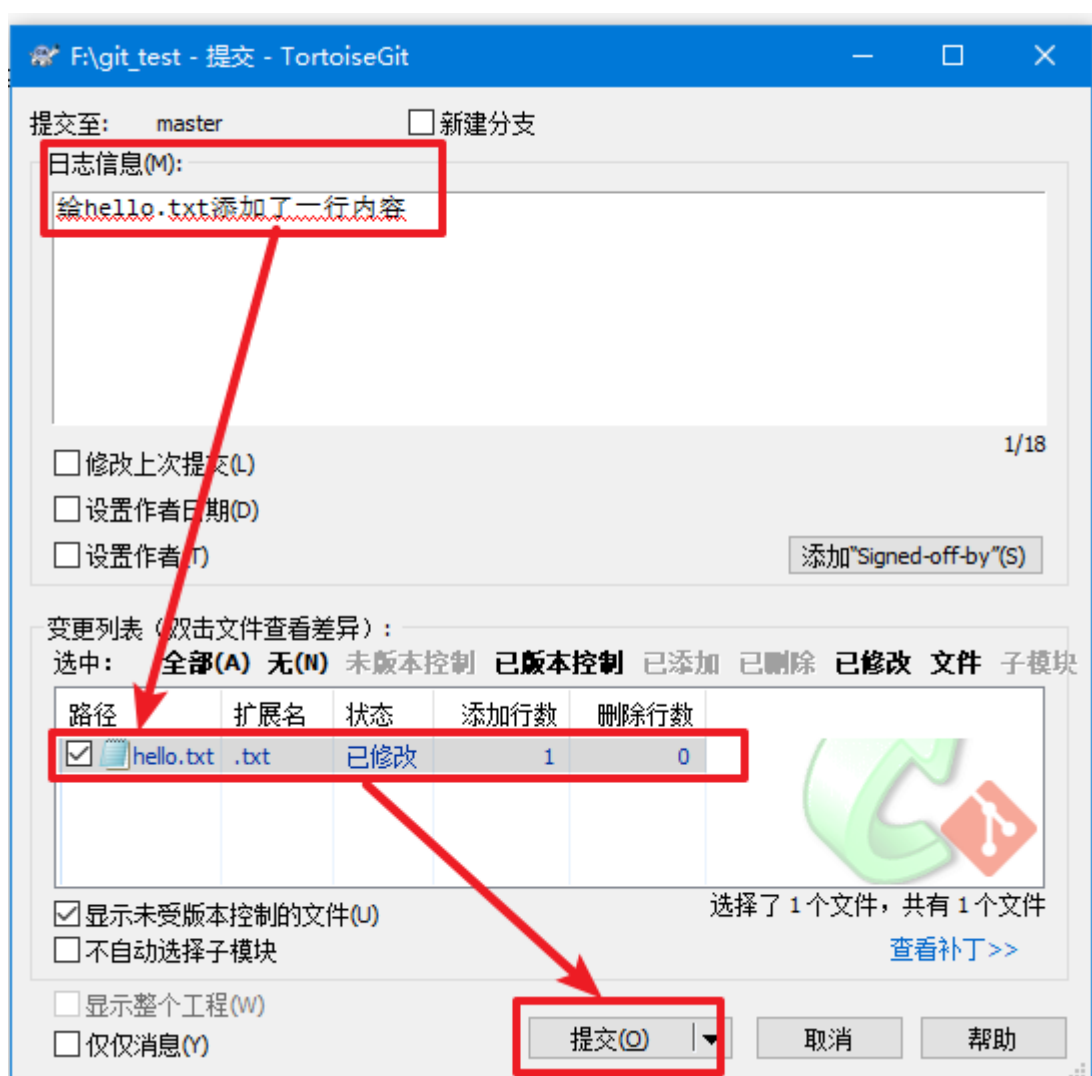
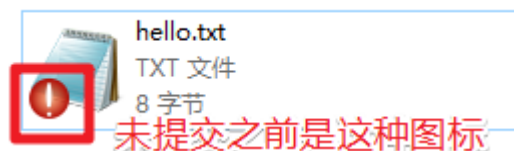
- 在本地创建一个目录和普通文件
- 单击右键, 然后如图所示:



- 按照图示操作:



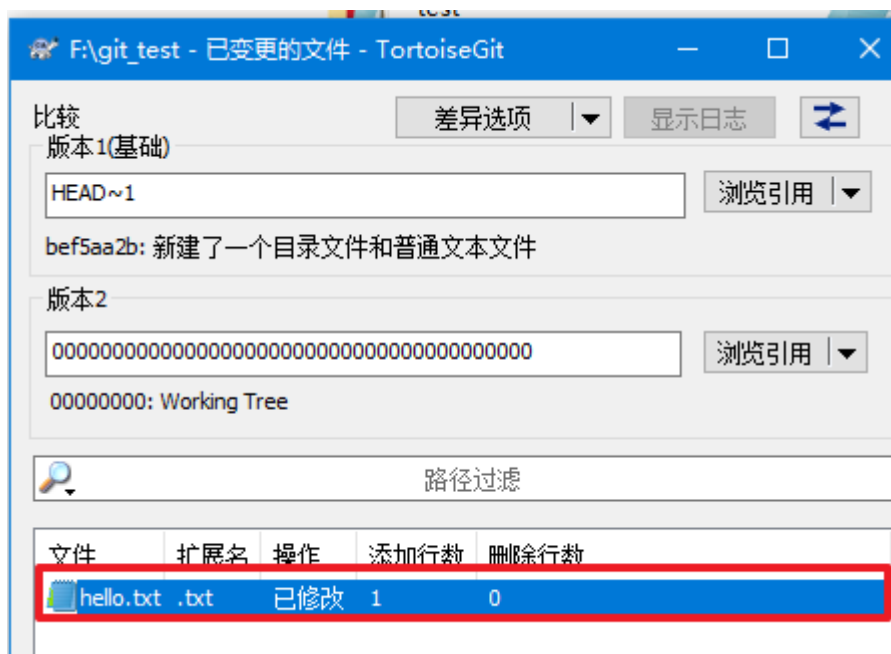
- 对文件进行修改, 然后进行提交



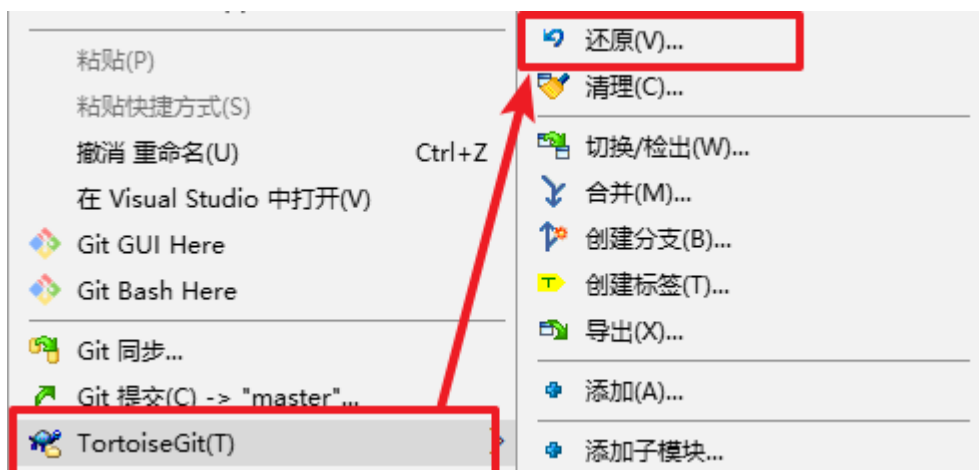
- 将刚刚提交的版本和上一个版本进行比较, 查看文件差异



选择hello.txt文件, 然后双击就可以查看文件差异了.



- 若不小心将本地的hello.txt文件删除了, 可以使用本地仓库进行还原



- 若对文件进行了修改, 但是还未提交, 想于版本库的最后一个版本进行比较





## 7.2 远程仓库相关操作

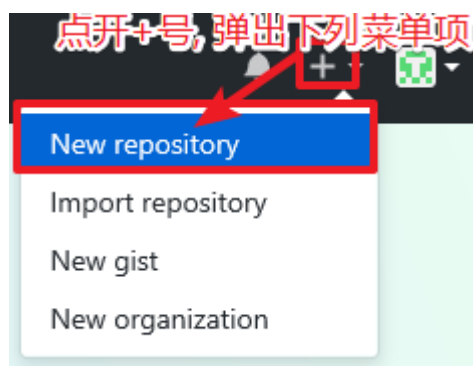
登录:

- Github: <https://github.com/>
- 码云: <https://gitee.com/>

这里以Github来介绍:

首先登录Github网站, 并完成用户注册.

- 1 如何创建远程仓库



## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: bdhang / **远程仓库的名字**

Great repository names are short and memorable. Need inspiration? How about **ideal-invention**.

Description (optional): **描述信息, 可写可不写**

☒ **Public**  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

**Create repository**

建好远程仓库之后, 可以查看到, 描述信息也可以在此处显示

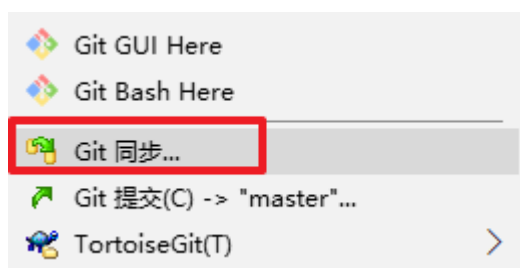
Overview Repositories **1** Stars **0** Followers **0** Following **0**

---

Popular repositories

**MyTest**  
我的第一个远程仓库

- 2 将本地仓库同步到远程仓库



[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)

### Quick setup — if you've done this kind of thing before

Set up in Desktop

 or 

HTTPS

SSH

https://github.com/bdhang/MyTest.git

将远程仓库的网络地址复制

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

配置远程仓库信息, 最后按[添加/保存]

远端:

origin

远端:

origin

重命名

URL:

https://github.com/bdhang/MyTest.git

推送 URL:

将刚刚复制的远程仓库的网络地址复制到此处

Putty 密钥:

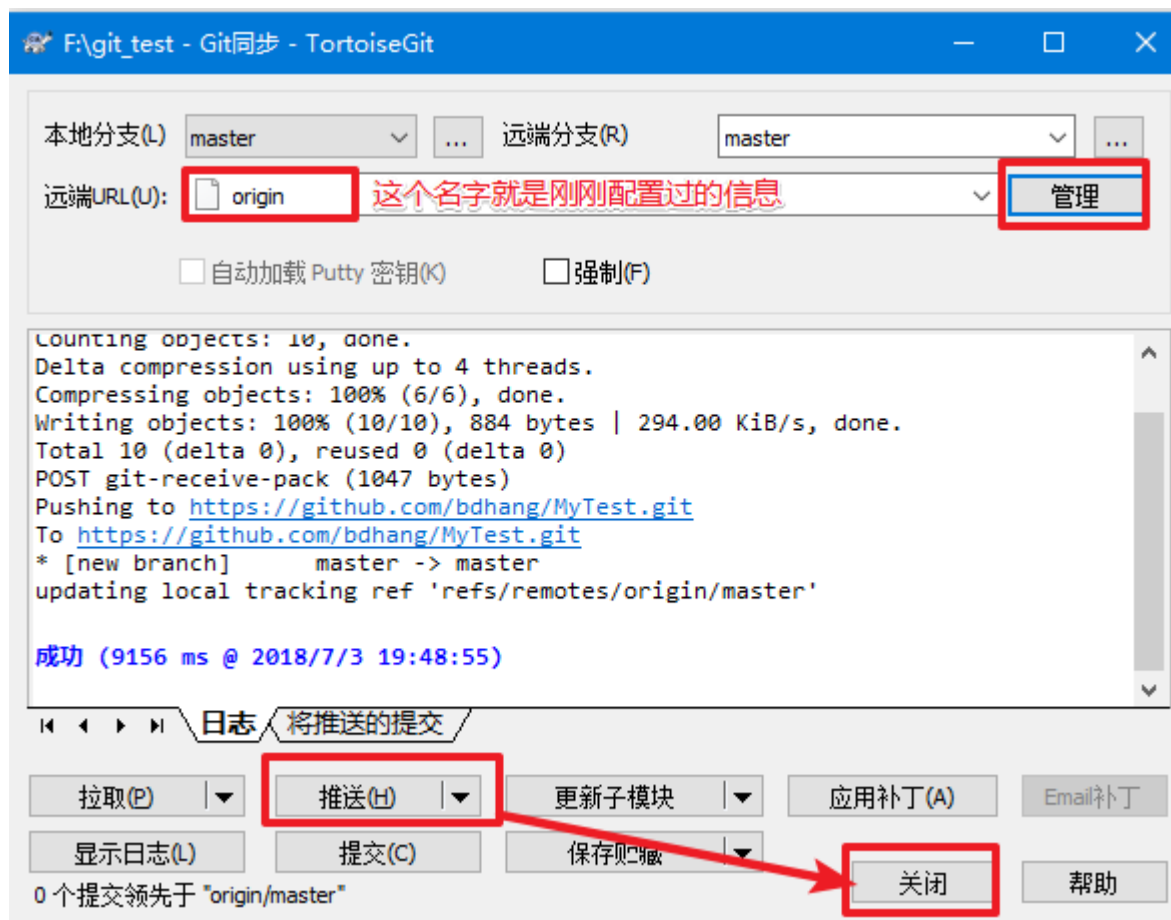
...

标签:

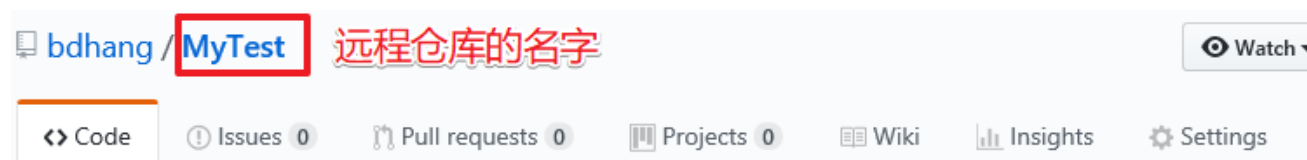
可达 (Reachable) ▾

☐ 推送默认





查看远程仓库是否推送成功



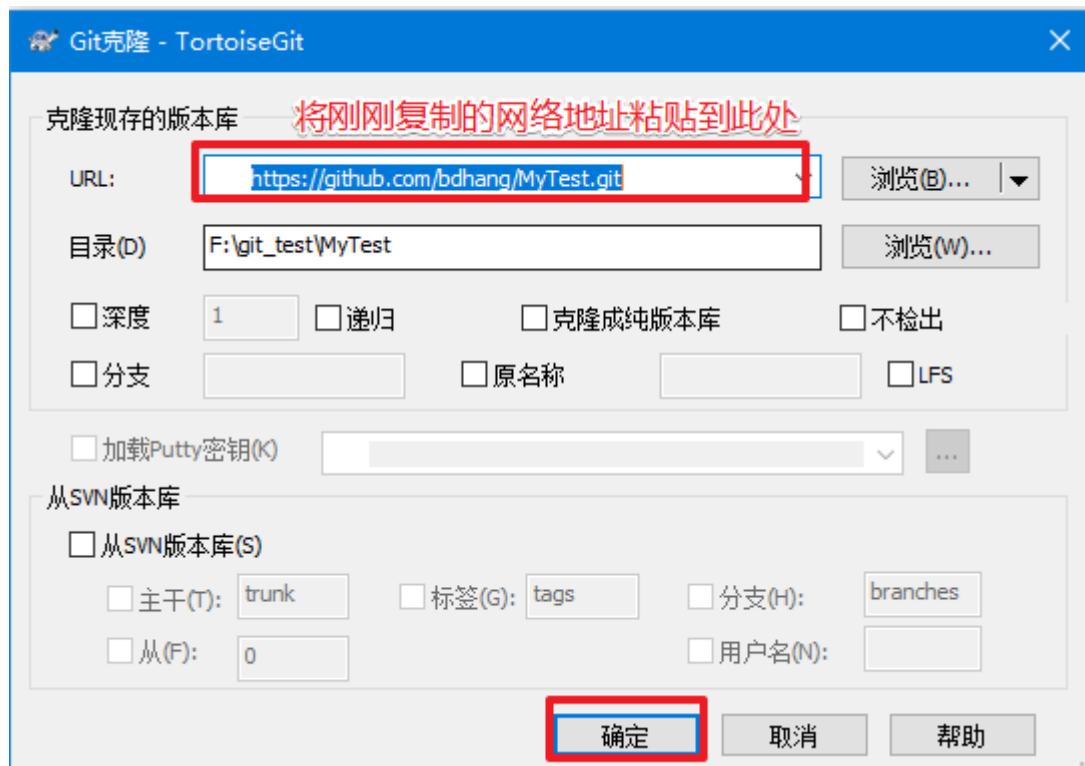
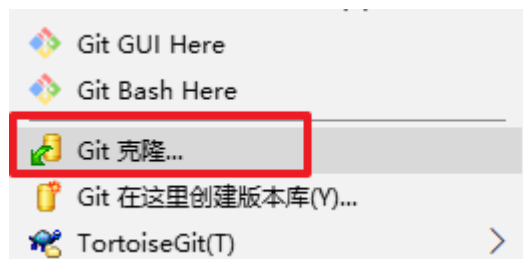
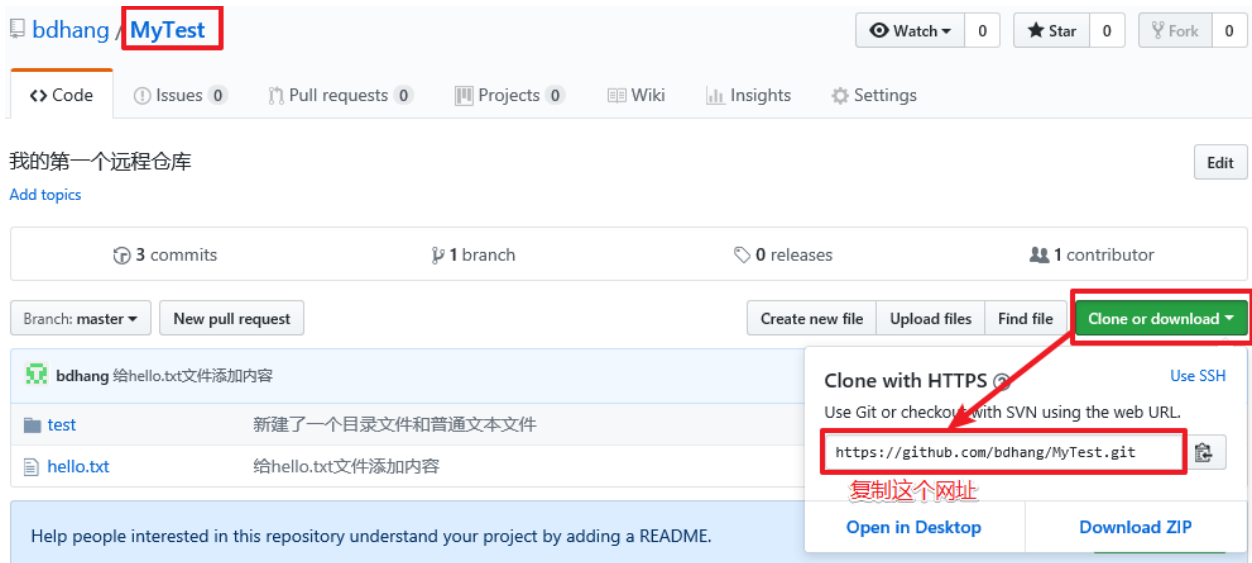
我的第一个远程仓库

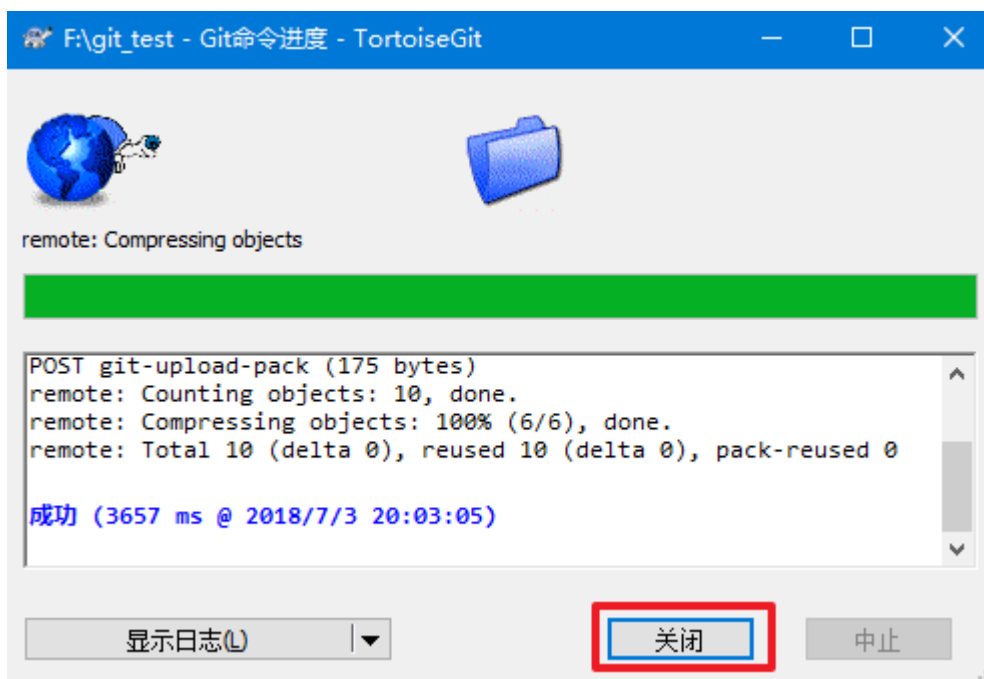
[Add topics](#)



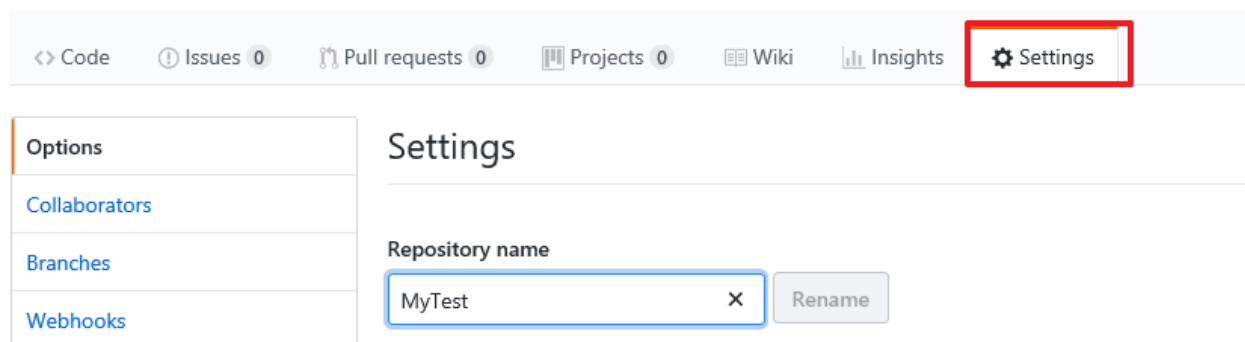
3 如何将远程仓库的文件同步到本地?

- 先获得远程仓库的网络地址:





- 如何删除远程仓库



## Danger Zone

### Make this repository private

Please [upgrade your plan](#) to make this repository private.

### Transfer ownership

Transfer this repository to another user or to an organization where you have the ability to create repositories.

Transfer

### Archive this repository

Mark this repository as archived and read-only.

Archive this repository

### Delete this repository

Once you delete a repository, there is no going back. Please be certain.

Delete this repository

### Are you absolutely sure?



Unexpected bad things will happen if you don't read this!

This action **cannot** be undone. This will permanently delete the **bdhang, MyTest** repository, wiki, issues, and comments, and remove all collaborator associations.

Please type in the name of the repository to confirm.

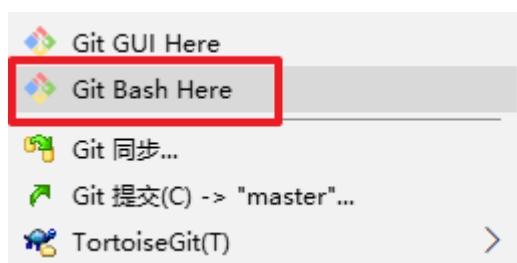
MyTest

填写要删除的本地仓库的名字



I understand the consequences, delete this repository

- 使用ssh协议同步本地仓库到远程仓库
  - 需要进行配置
    - 生成秘钥



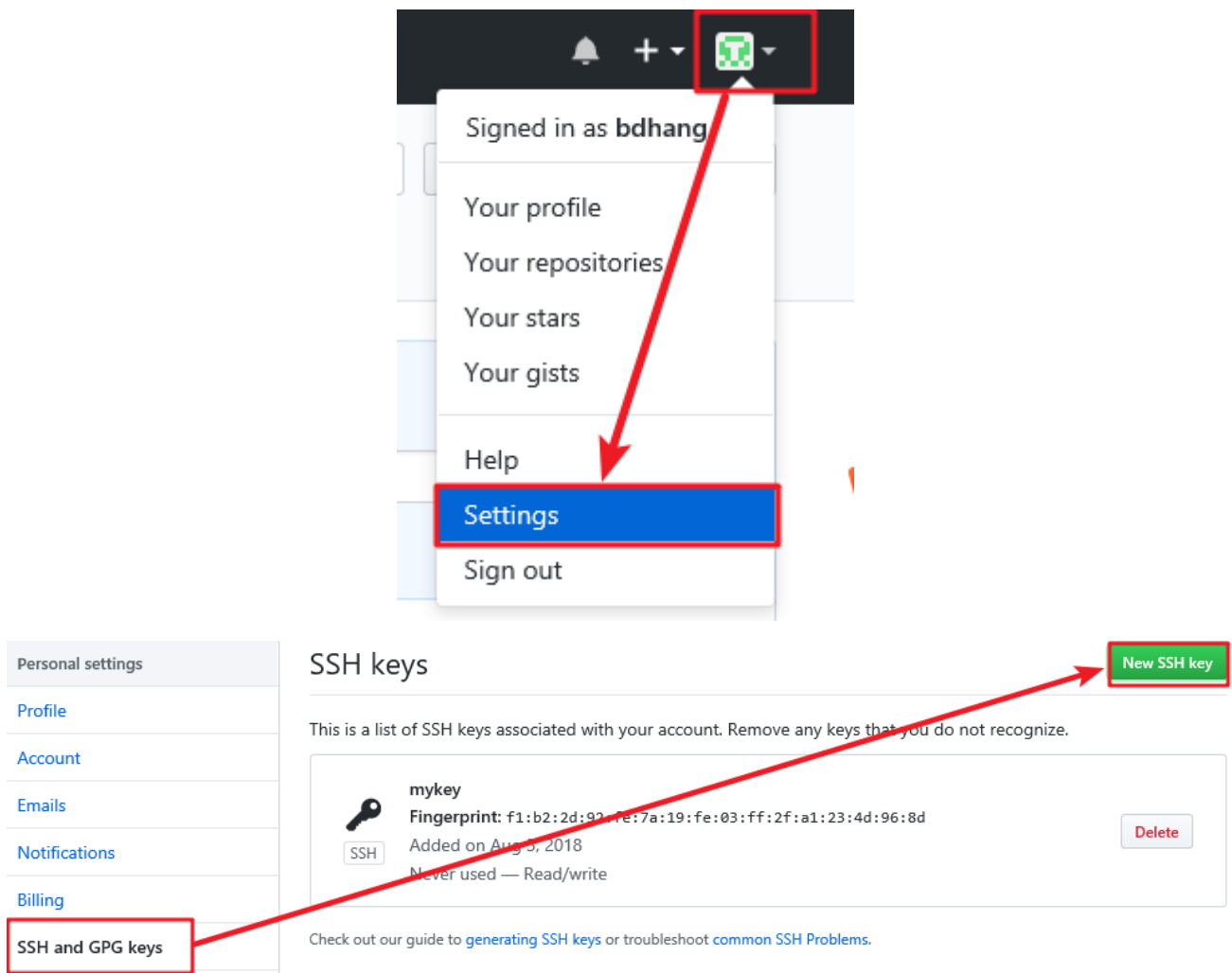
```
MINGW64:/f/git test/MyTest
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Administrator/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Administrator/.ssh/id_rsa.
Your public key has been saved in /c/Users/Administrator/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:oICGR2qi1rYcIdV3cBL4ypd18DbdsZqkd1zXeszCIlc Administrator@PCOS-1803141627
The key's randomart image is:
+---[RSA 2048]---+
|    . . . .+o.    |
| .+.  o  .oo     |
| =+o.  + . o . . +|
| =.o... o . = oE+o|
| .. +o . S o =o++|
| . o oo o .oo=+ +|
|   o . .o.. o   |
|-----[SHA256]-----+
Administrator@PCOS-1803141627 MINGW64 /f/git_test/MyTest (master)
$
```

执行命令, 然后连续回车

生成的公钥和私钥保存的文件路径

执行到此处表示成功

- 在远程仓库配置公钥



Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

## SSH keys / Add new

Title

mykey

起一个名字

Key

ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQDFJye22bfuFF4oMsqvp+Ak+2zulHcfFRwpbyg3ZX7BWvFKm/+ov6n5S/UUT  
l+eKGiBQ5gGvH+uiVwnWeagRsod4l1EhSFyhwJis4ZF9CMCGCvG5W+/H+AiUcSGBcgY0KMLv2aScSuQjtu/q1W70Nk0  
2+Gr0ZB1VMR1CbZk1/zbLS13kCzV59nj6z114KW7PBCUzsLI9YrUdgDcxK6eB9+lwjILDcXvY8BxMPsTw73AVfehVWwfJ2  
Zv5/1KW02LAOKep1oh/7ci5VOVui0O7cuXMEUXBYTjMnHI39HMQM5a/JsRDNa1TFtOxYHvTMBSAdBYZamljuR2ymb  
0CrxNLov Administrator@PCOS-1803141627

将公钥文件中的内容全选, 然后复制到此处

Add SSH key

Personal settings

Profile

Account

Emails

Notifications


Billing

SSH and GPG keys

## SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

 mykey

Fingerprint: f1:b2:2d:92:fe:7a:19:fe:03:ff:2f:a1:23:4d:96:8d

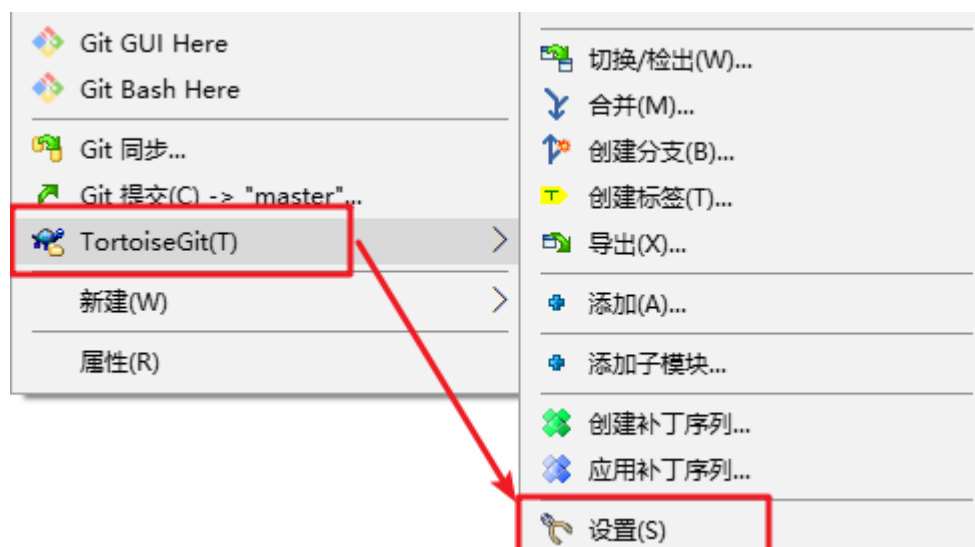
Added on Aug 3, 2018

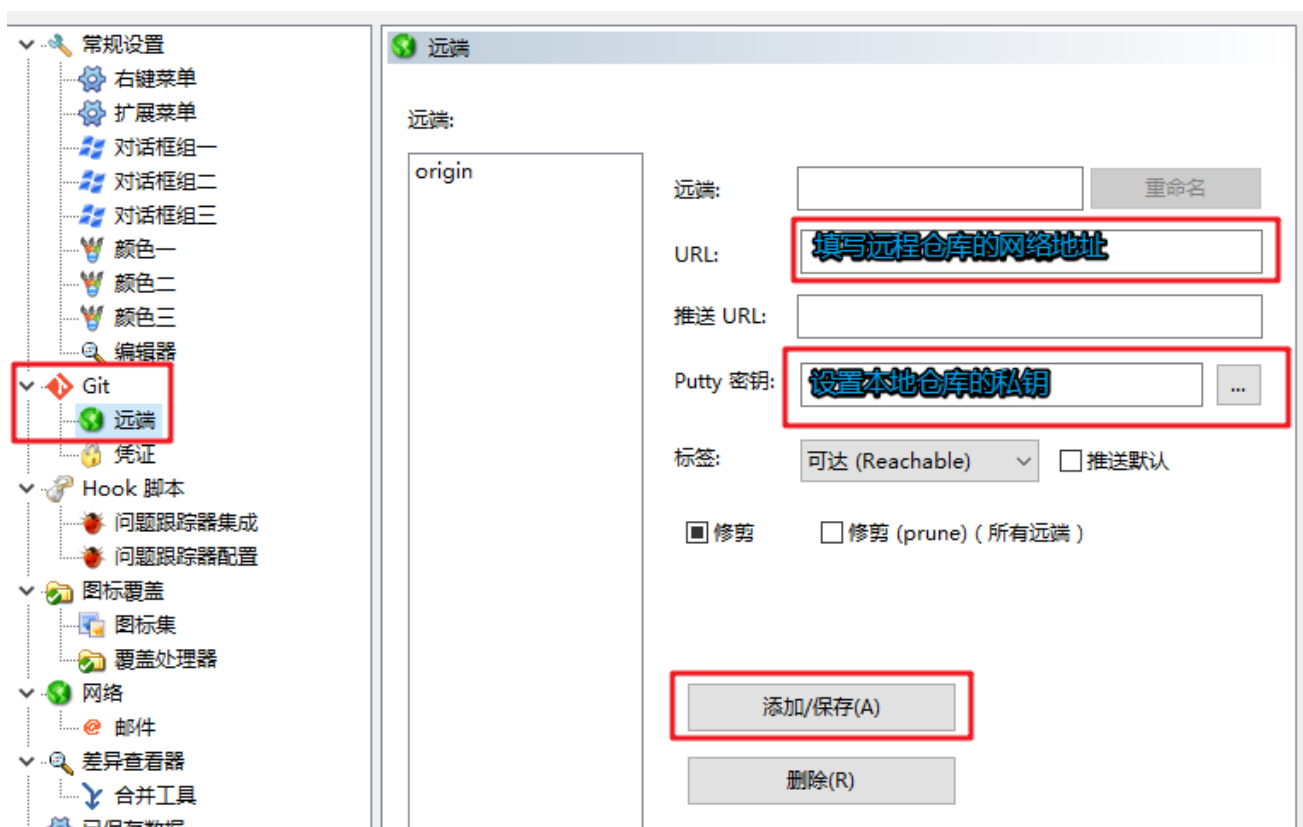
Last used within the last week — Read/write

Delete

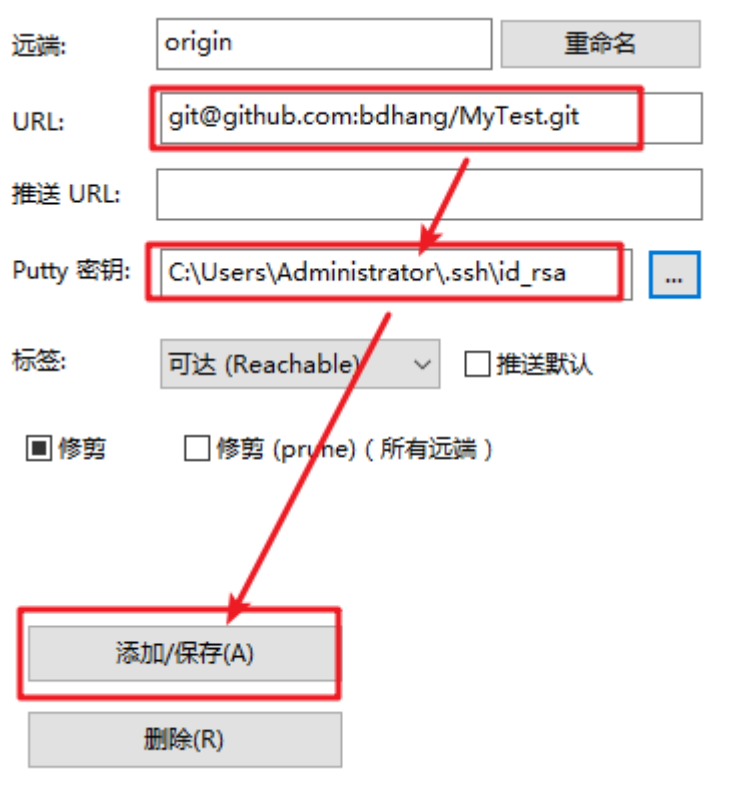
Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

在本地仓库设置私钥:

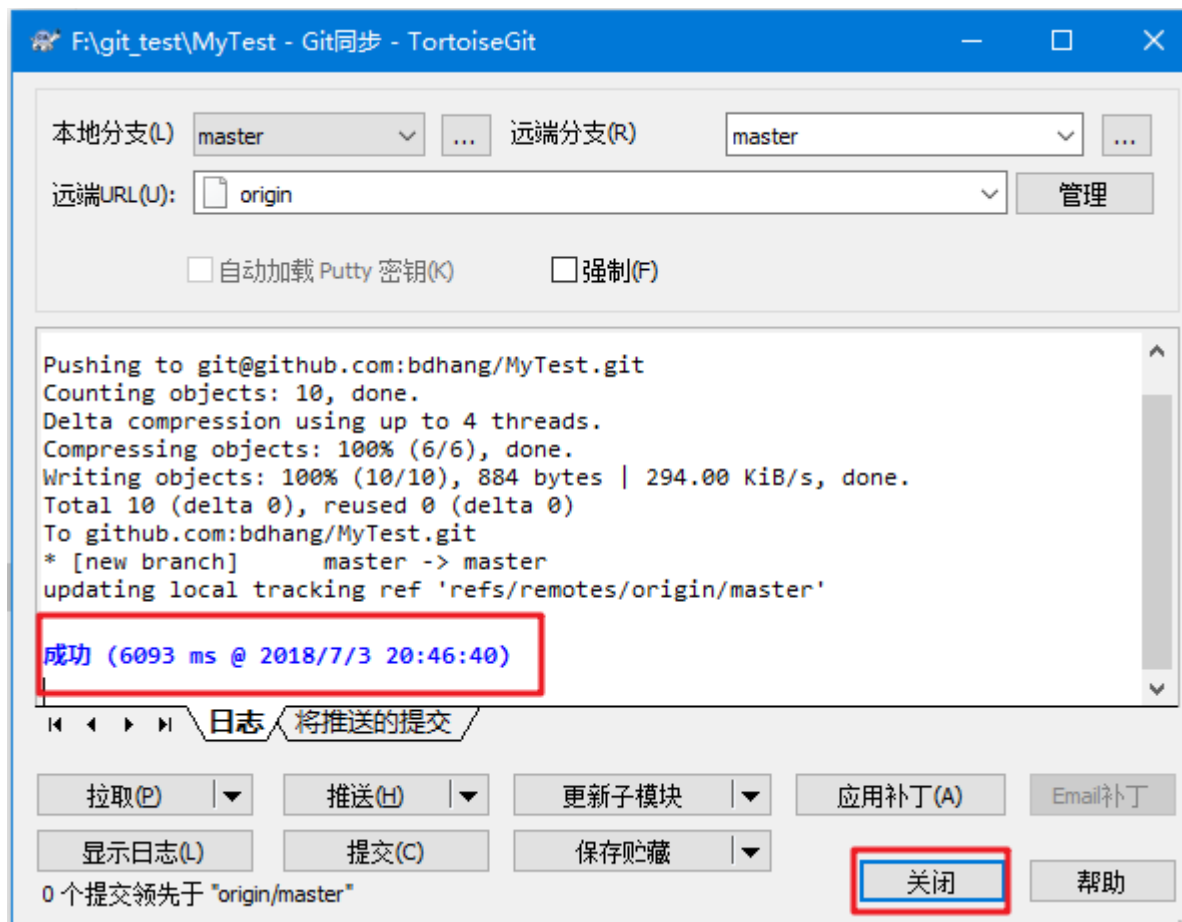
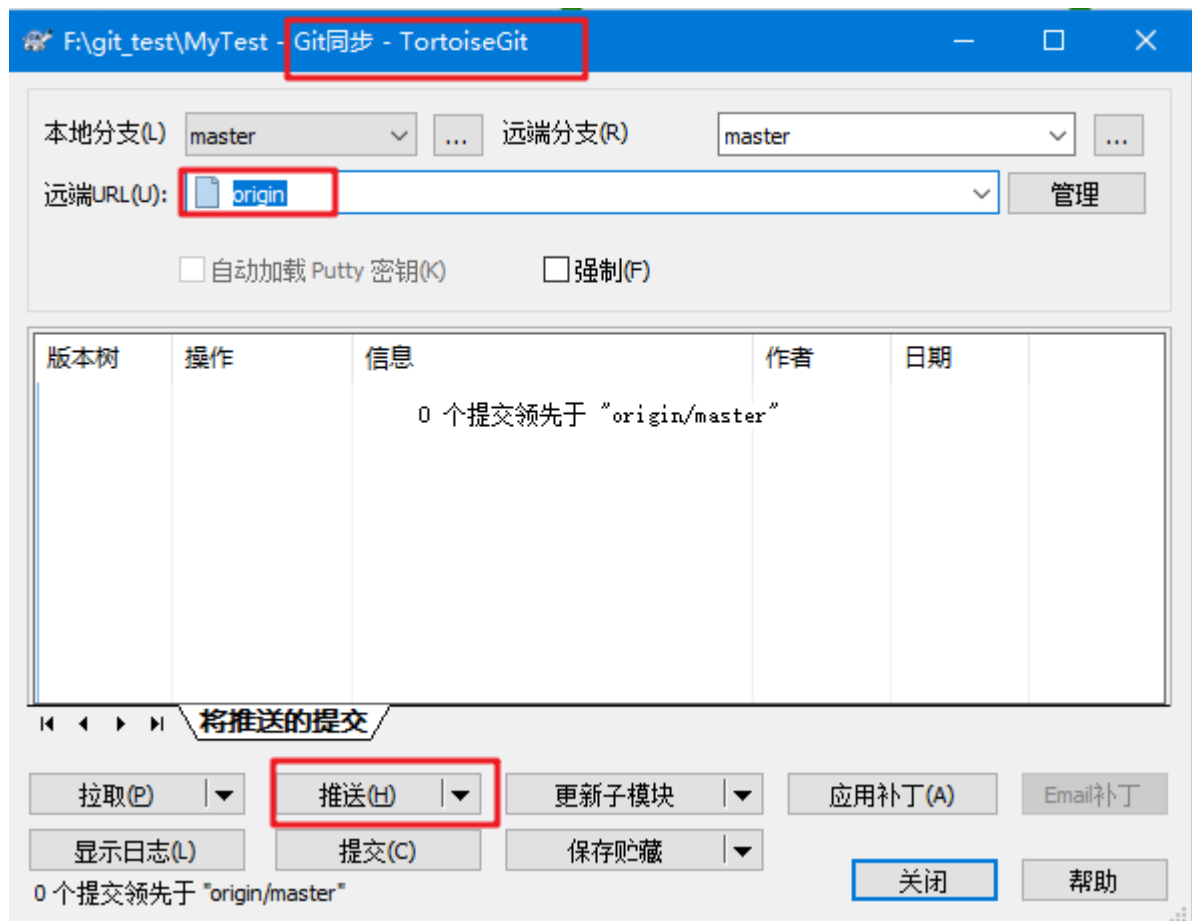




复制远程仓库的网络地址:



然后在本地工作目录空白处, 右键选择同步:

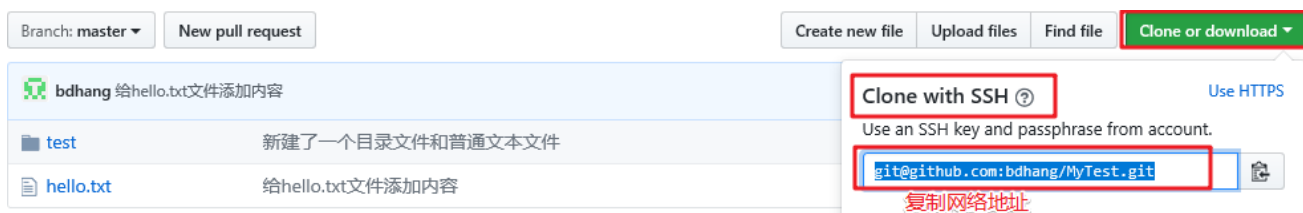




使用ssh协议将项目文件从远程仓库同步到本地仓库:

首先在本地新建一个工作目录:

然后从远程仓库复制网络地址:



1530622548658

