Git for windows入门使用指南

前言：

git gui：git gui是一个windows交互式管理工具，在本章中将会提到一些简单的使用案例

git bash：git bash是一个windows提供的dos界面，大部分git 命令都可以在git bash中执行，和在linux上执行几乎一样

一，下载与安装

https://git-scm.com/ 下载最新的git for windows安装即可

1.1 配置ssh-key

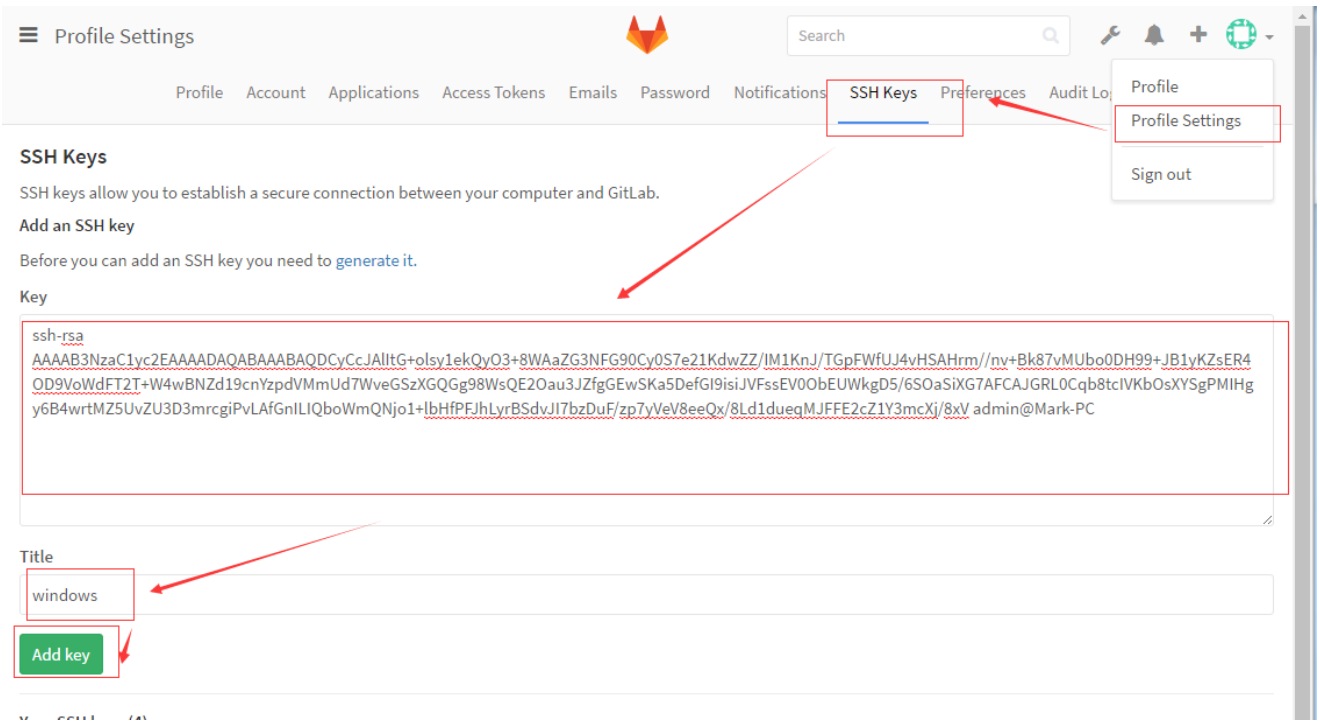
新建一个目录，鼠标右击选择git bash here，在弹出的界面中执行/usr/bin/ssh-keygen -t rsa生成秘钥

```
$ /usr/bin/ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Mark.Mark-PC/.ssh/id_rsa):
/c/Users/Mark.Mark-PC/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Mark.Mark-PC/.ssh/id_rsa.
Your public key has been saved in /c/Users/Mark.Mark-PC/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ajDxg67C5FcPsfCwsr3bxP67v6umPt3/XfZol5Xv0qk admin@Mark-PC
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|                 |
|    .            |
|   o =           |
|    O = S       .|
| o o.B o       ..|
|+ + oo* .      .B|
|.+ ++o + .    +=*|
| .oo+====++..Eo=o|
+----[SHA256]-----+
```

cat C:/users/USERNAME/.ssh/id_rsa.pub 这个是秘钥的位置，每台机器的用户各不相同，请各自修改

```
admin@Mark-PC MINGW64 /c/users/Mark.Mark-PC/.ssh
$ cat  C:/users/Mark.Mark-PC/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDOmeVXJI86KMRyxbKXFe37znpmmM7djp1kzHNCgG17P/uogG869QckXV6CUZ9tXATXLu
dWEzI//fhoUxwxpY4qP8FNByoNrP2hGsY/bhXhzR17I8JgFBYWj2DOt7TSrjYvjuHs6OjfxqOUGdBK5biKfW3JvGDZ6YyugsC3
LbM+IF6edjF+55Lrq6zl0MdM/REazjKlNXMQlI86He58VtUx4LwdJmk4SpWWKBN0cS33Ky+DNxeow5Ae1faKQtPMS3D9k8jniU
wqwaOPvI/CsNtcHZy/VkIcF2Q1hluKOQCwTpHzfNxRtetEVHtVJZVB6atIHLOUUQDoPl/rFdoeuutf admin@Mark-PC

admin@Mark-PC MINGW64 /c/users/Mark.Mark-PC/.ssh
```

将key添加到gitlab中

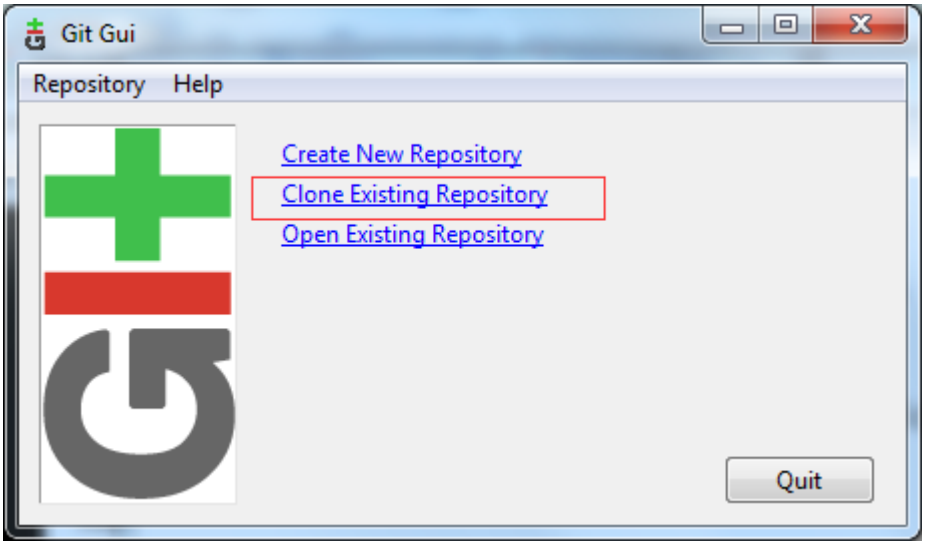


1.2 git bash署名信息

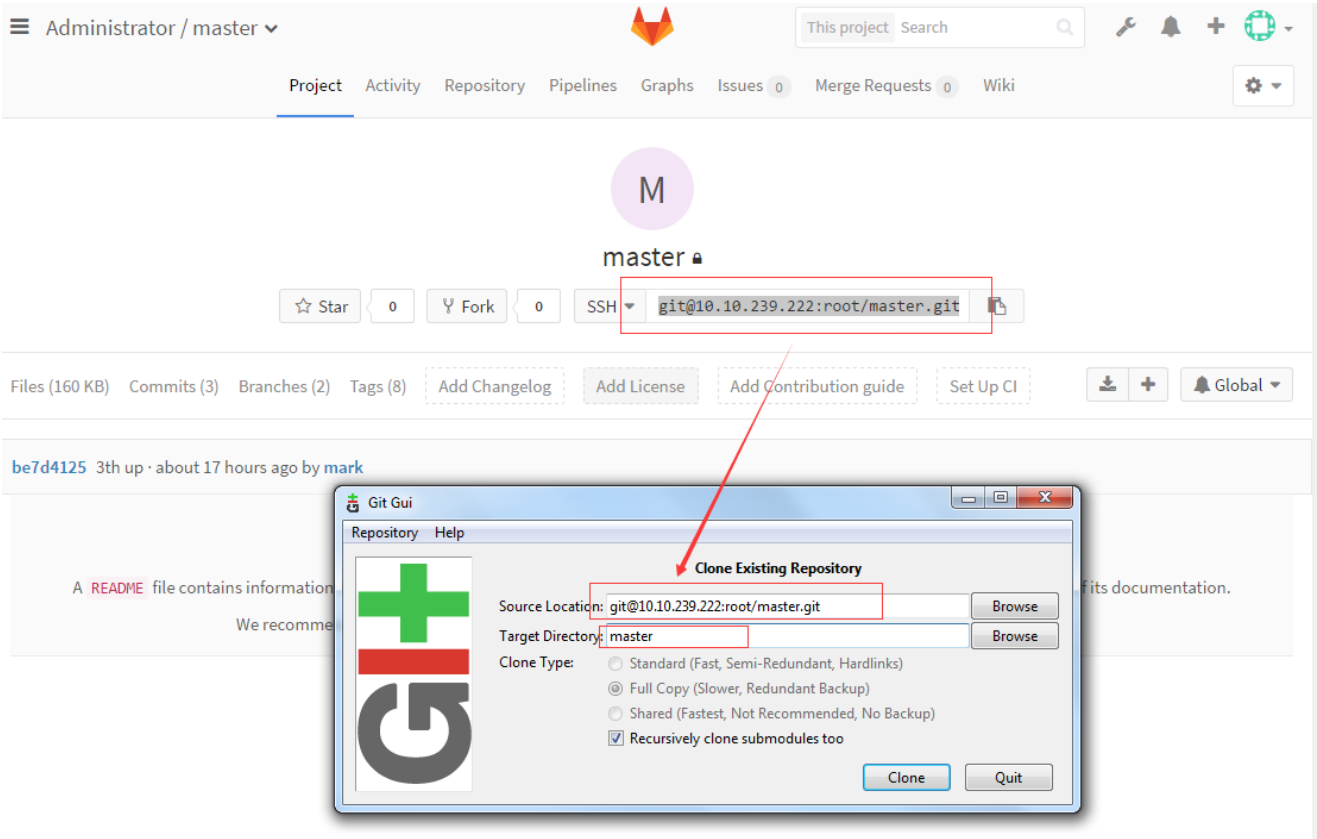
```
admin@Mark-PC MINGW64 ~
$ /mingw64/bin/git config --global user.name "mark1"

admin@Mark-PC MINGW64 ~
$ /mingw64/bin/git config --global user.email "mark@gmail.com"
```

1.3 克隆gitlab到本地

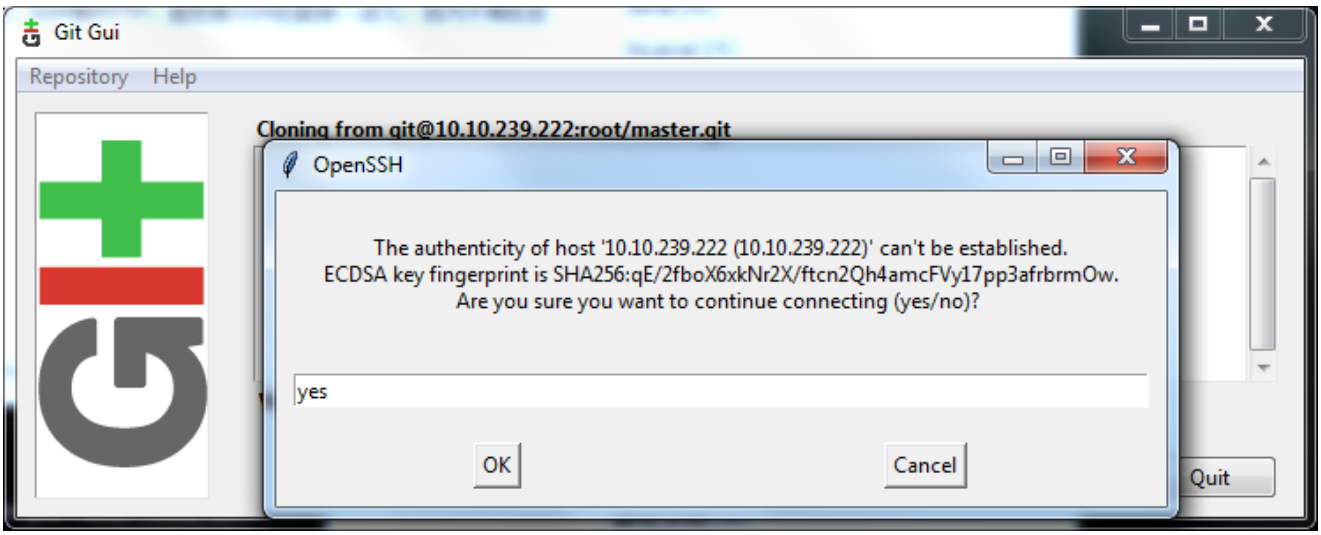
克隆gitlab到本地windows
当秘钥添加完成后，在当前目录中右击属性，打开git gui here.选择克隆，如下图

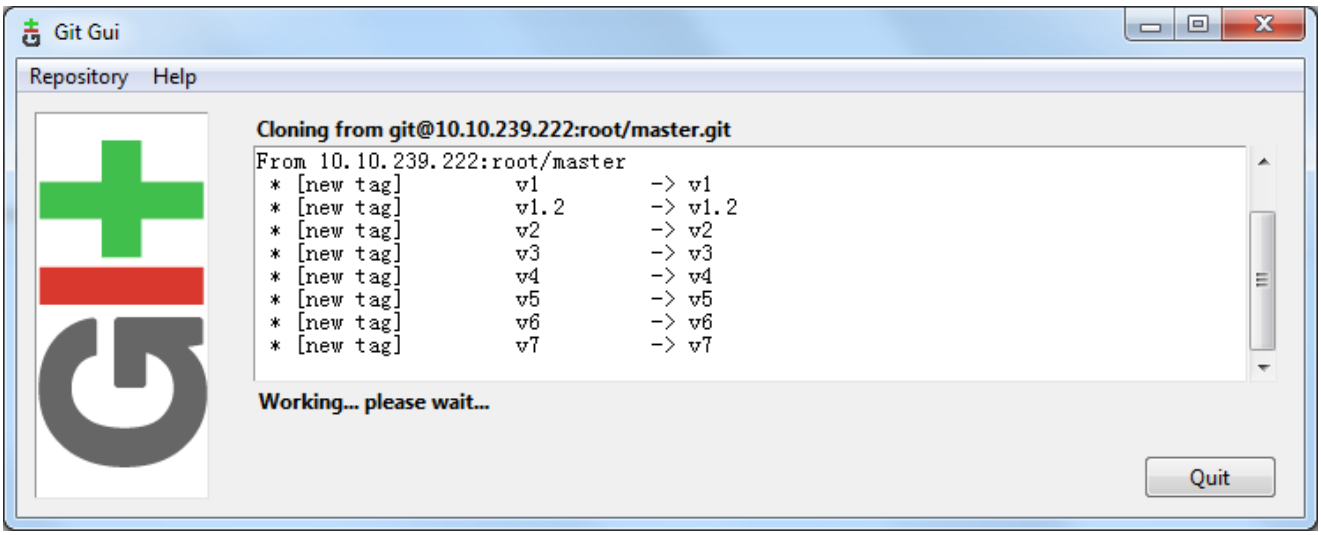


点击后跳转到克隆界面，source location输入的是gitlab中项目的位置、target directory输入的是当前目录下的名称

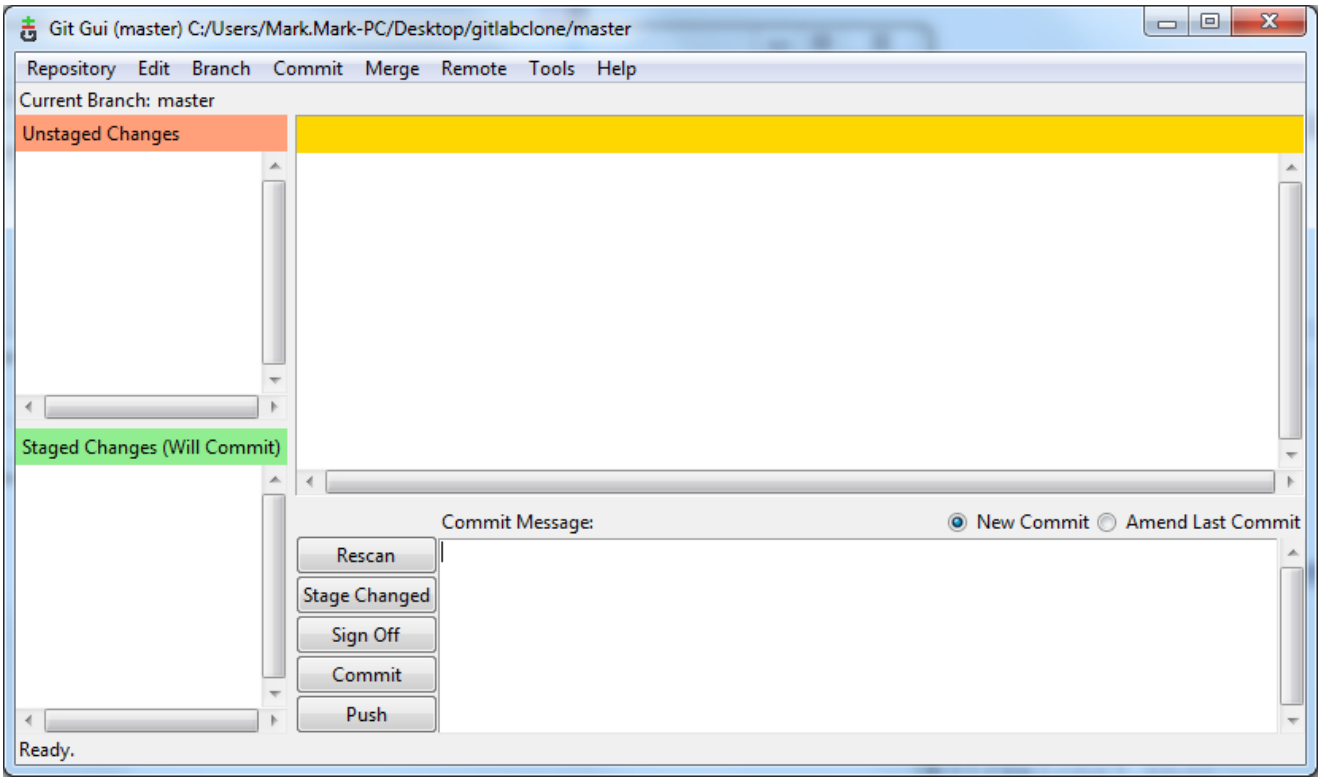


开始克隆，第一次会弹出验证，是否同意，yes即可


Git Gui
Repository
Help
Cloning from git@10.10.239.222:root/master.git
OpenSSH
The authenticity of host '10.10.239.222 (10.10.239.222)' can't be established.
ECDSA key fingerprint is SHA256:qE/2fboX6xkNr2X/ftcn2Qh4amcFVy17pp3afrbrmOw.
Are you sure you want to continue connecting (yes/no)?
yes
OK
Cancel
Quit



克隆完成后，到了界面中，在没有修改的情况下是不会有任何东西的



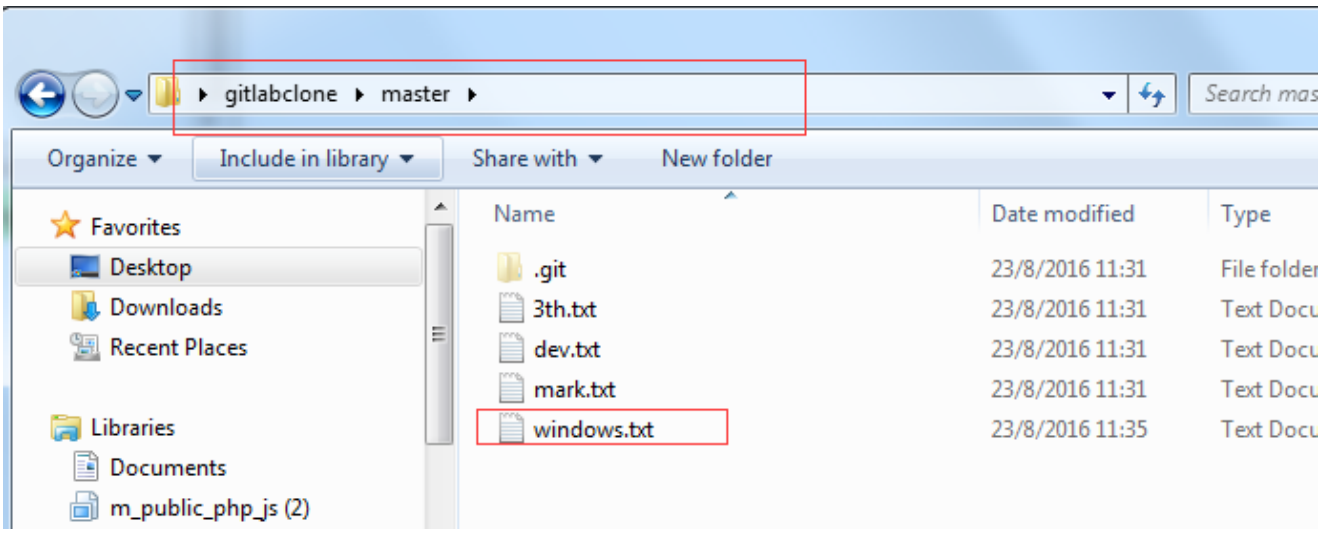
我们在目录中添加或者删除修改文件后，rescan即可

二， 上传

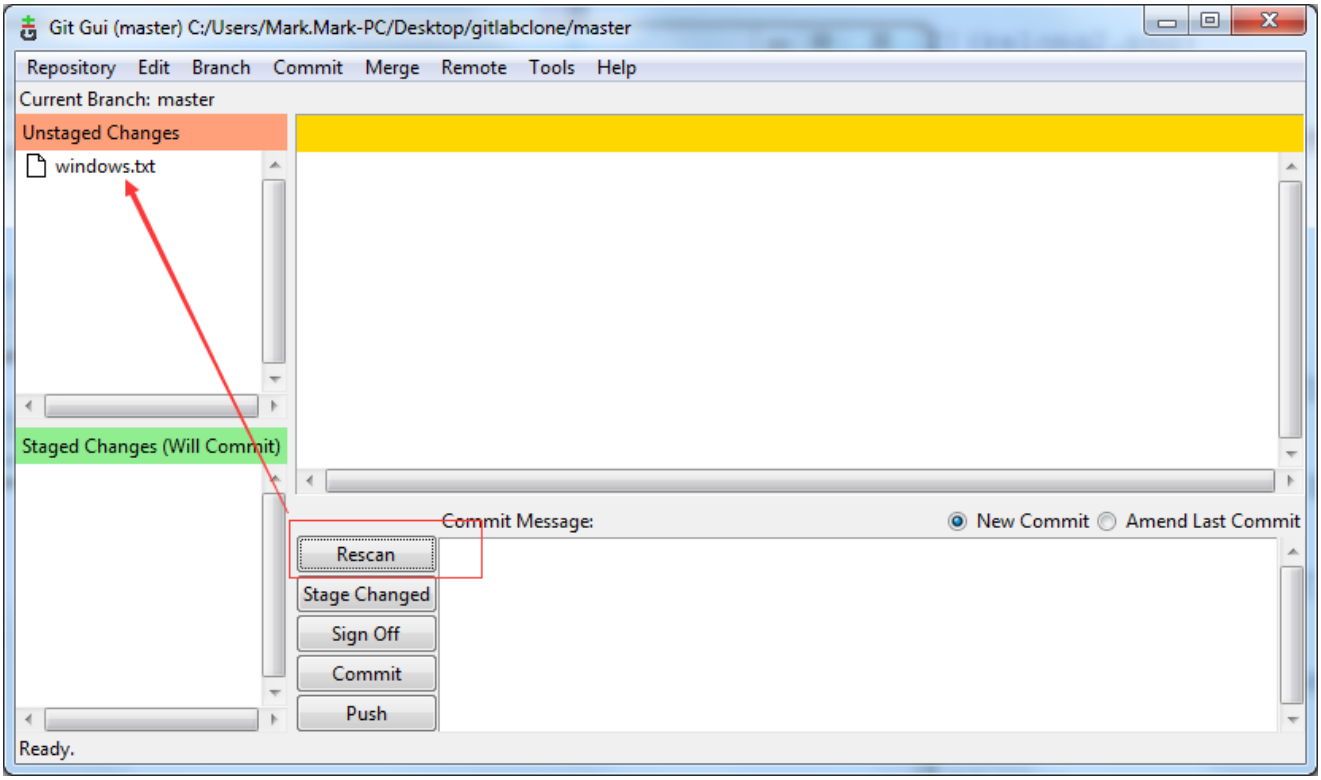
2.1 添加到暂存区

添加一个文件windows.txt，刷新即可见了

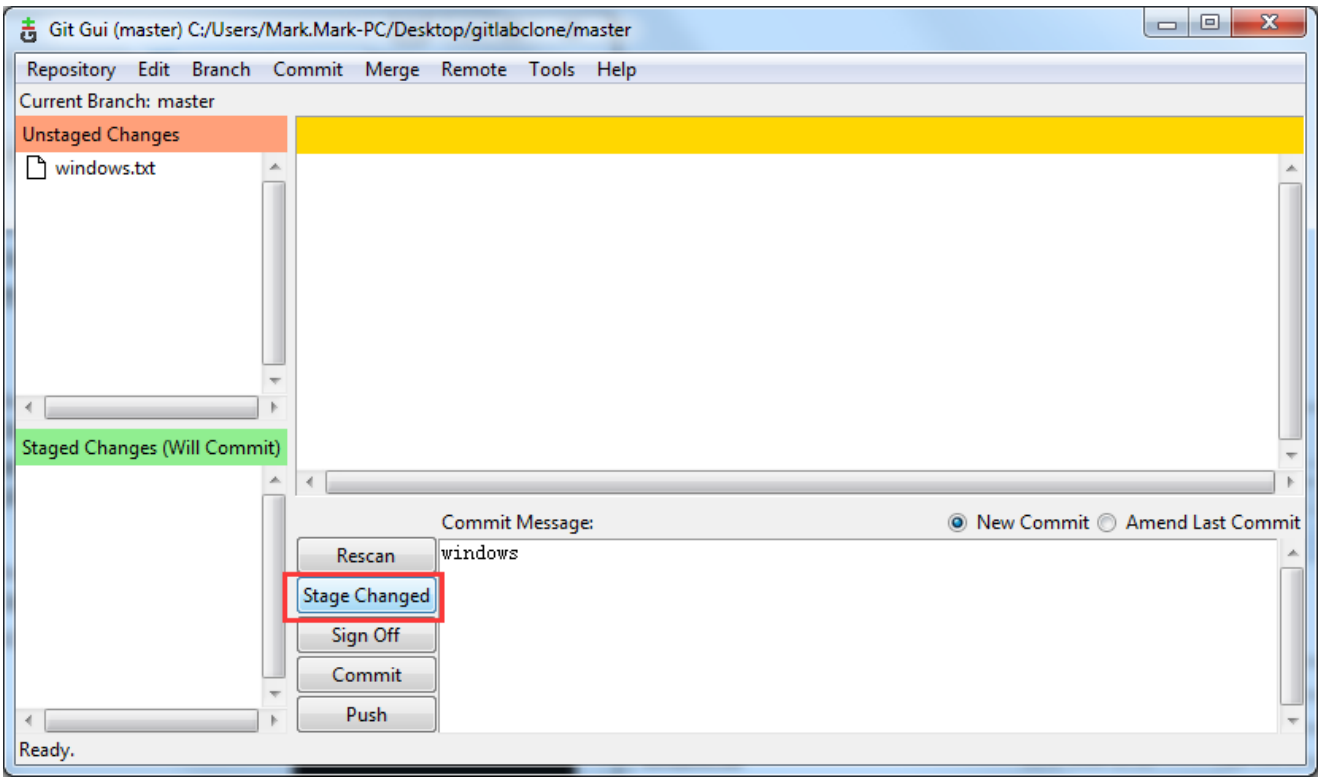
请注意，这里的所有操作都是在此目录中。master是当前目录下的master

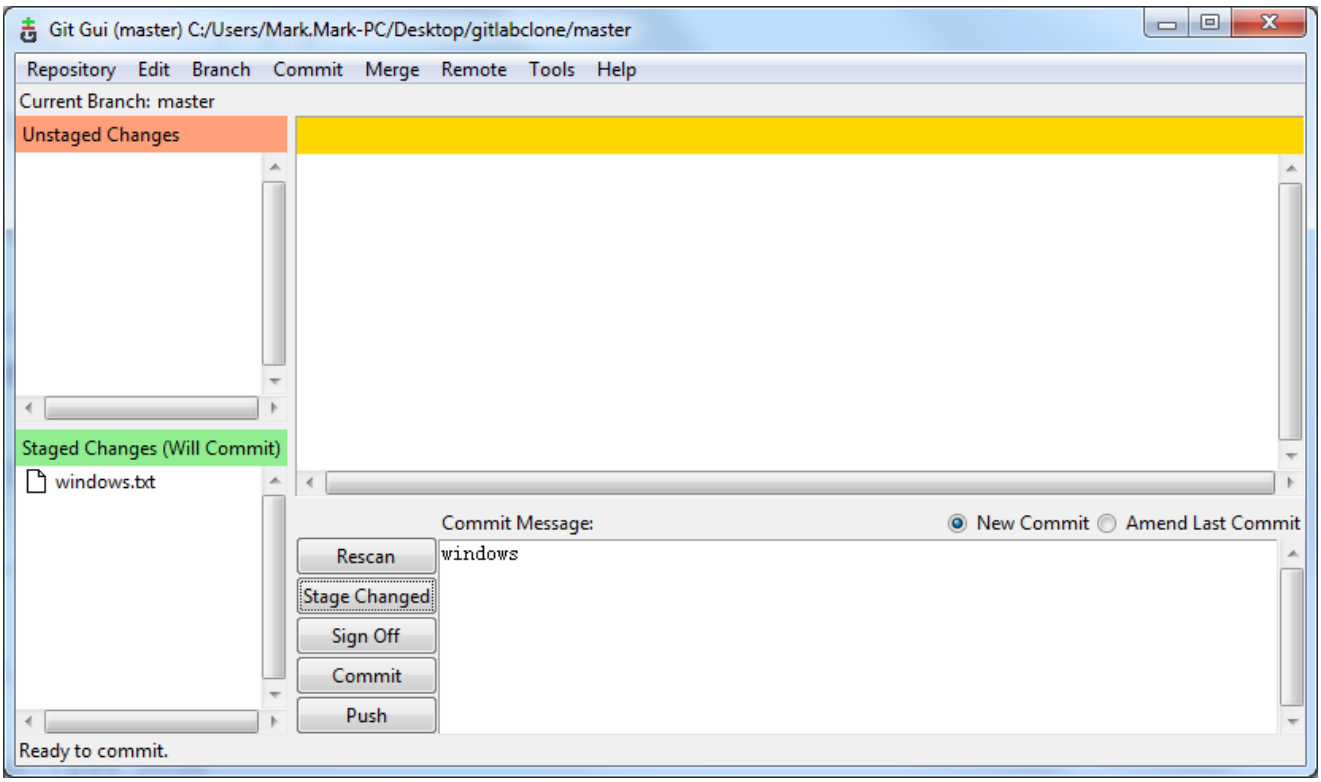


刷新即可见



stage changed相当于git add,我们将他添加到暂存区

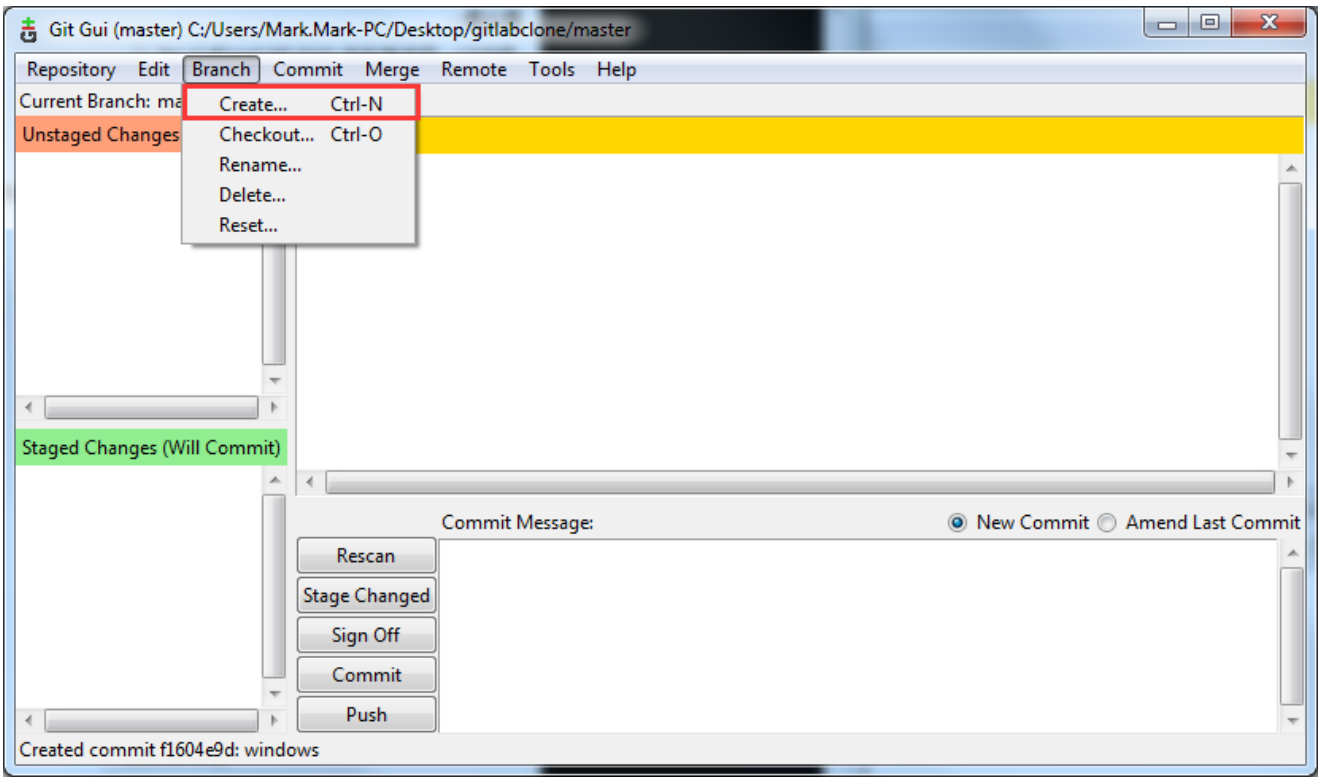




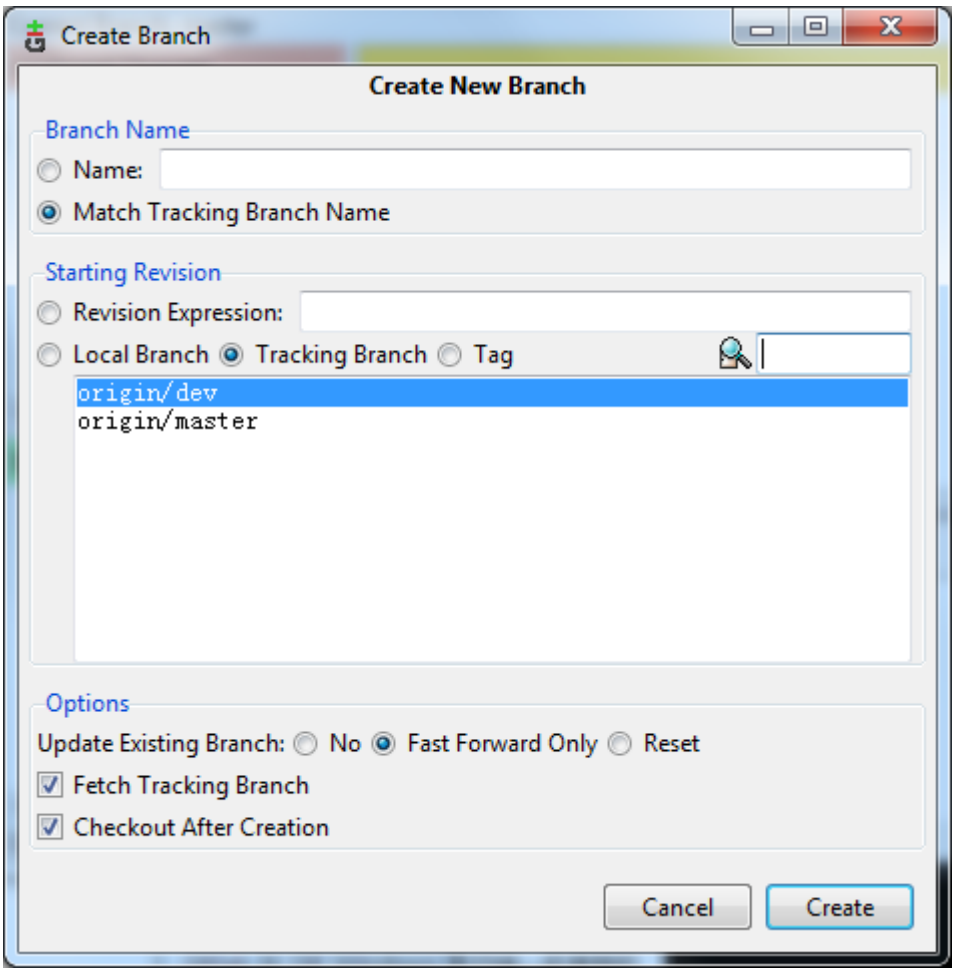
那么现在，stage changed完成后，就进行commit ，commit需要备注，而后commit即可

上传

在分支中-->新建

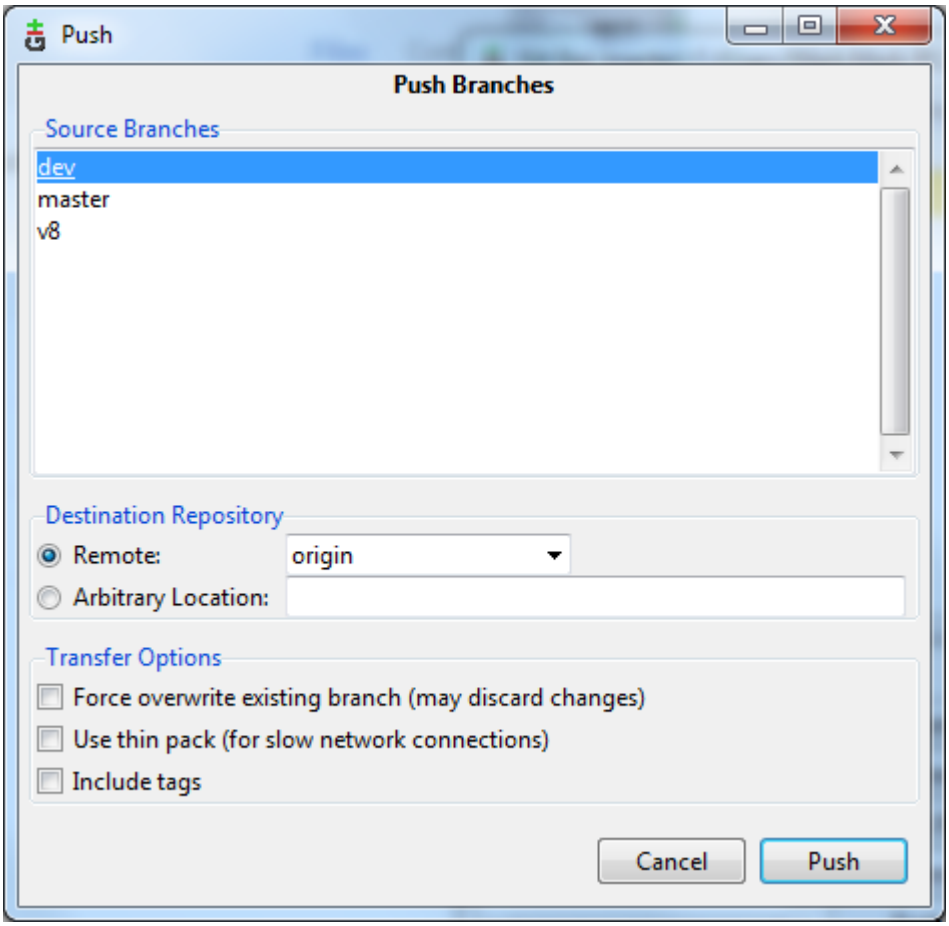

Git Gui (master) C:/Users/Mark.Mark-PC/Desktop/gitlabclone/master
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: ma
Unstaged Changes
Create... Ctrl-N
Checkout... Ctrl-O
Rename...
Delete...
Reset...
Staged Changes (Will Commit)
Commit Message:
New Commit
Amend Last Commit
Rescan
Stage Changed
Sign Off
Commit
Push
Created commit f1604e9d: windows

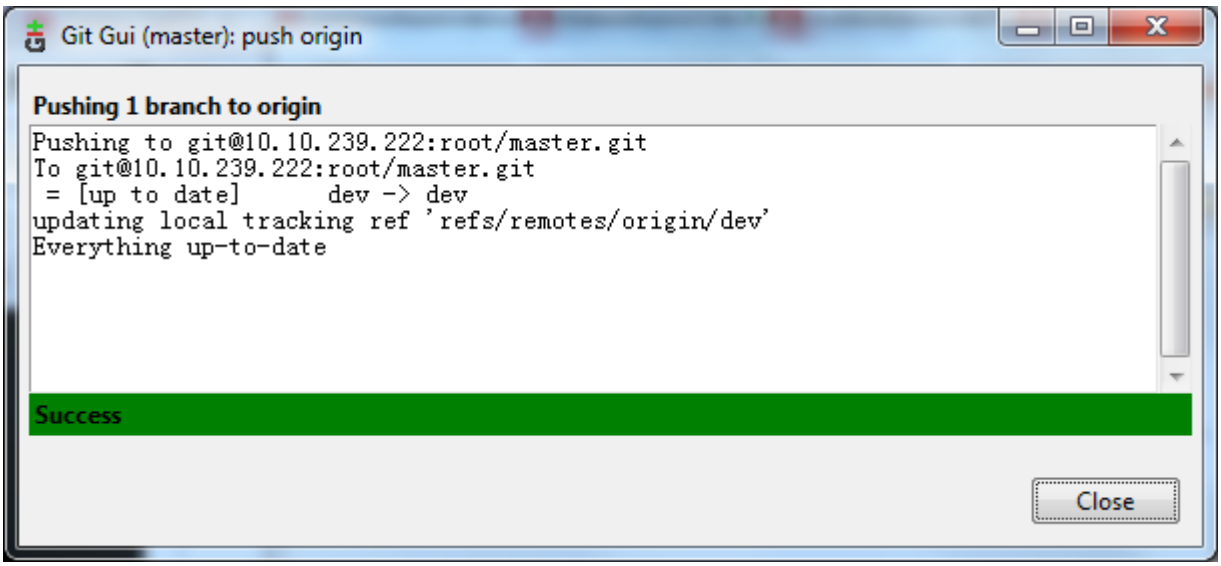
而后create即可


Create Branch
Create New Branch
Branch Name
Name:
Match Tracking Branch Name
Starting Revision
Revision Expression:
Local Branch
Tracking Branch
Tag
origin/dev
origin/master
Options
Update Existing Branch:
No
Fast Forward Only
Reset
Fetch Tracking Branch
Checkout After Creation
Cancel
Create

2.2 push文件到gitlab

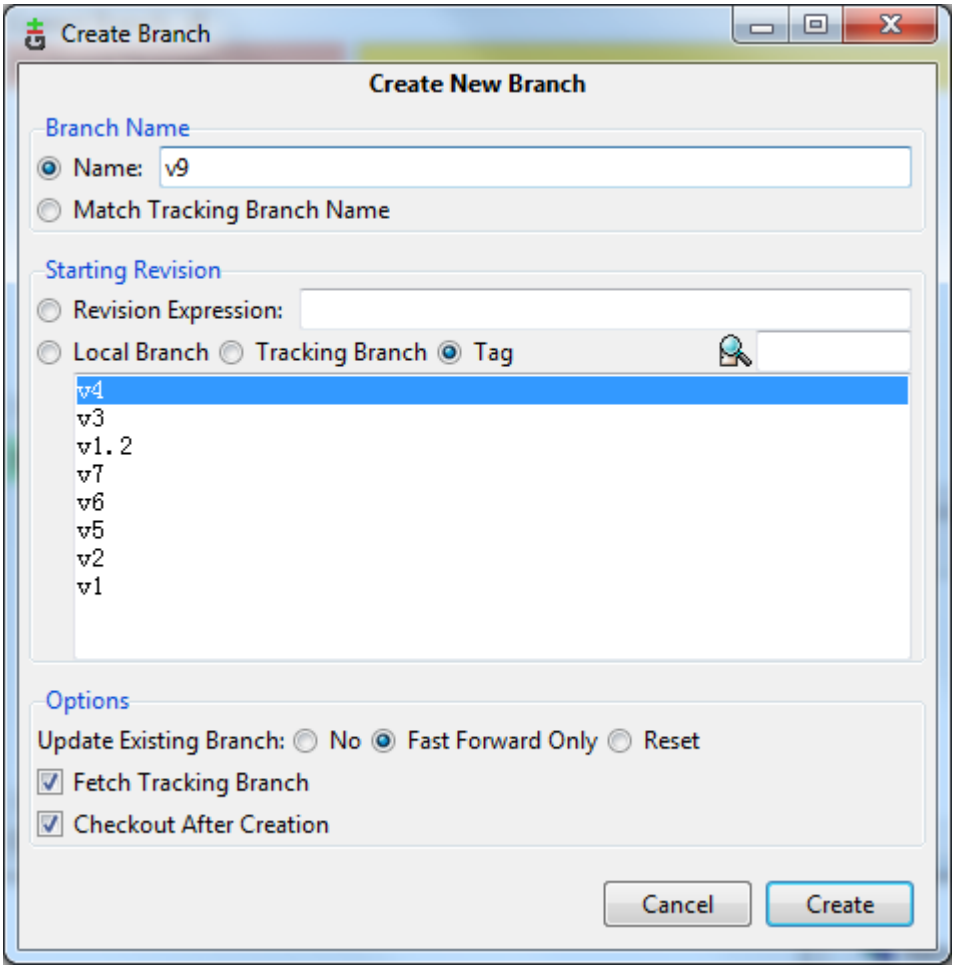
我们将他上传至dev分之下push



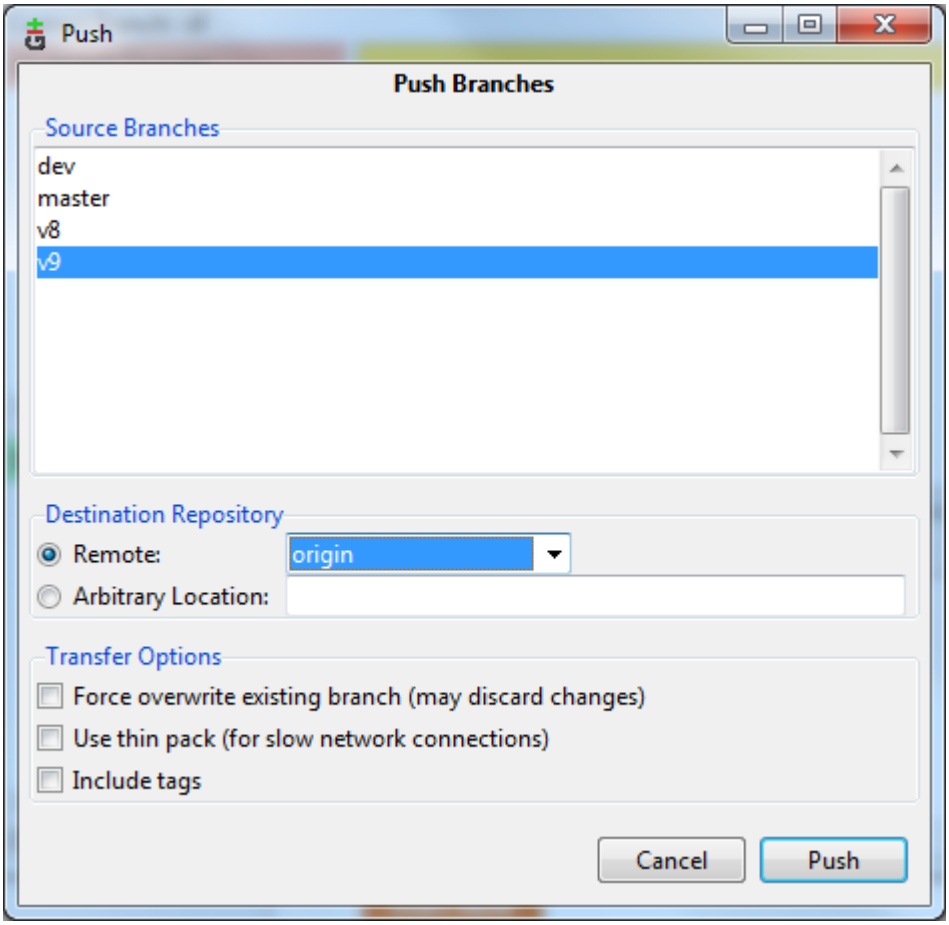


三，新建分之

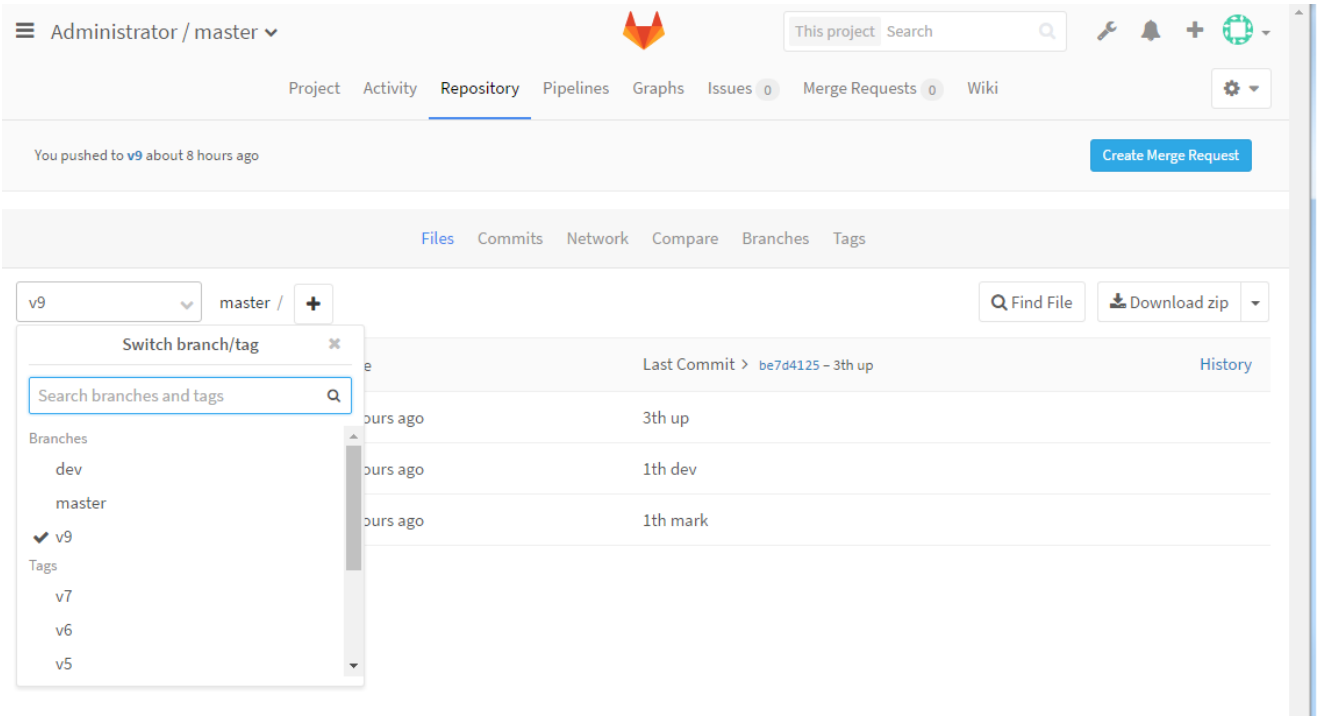
新建分之9


Create Branch
Create New Branch
Branch Name
Name: v9
Match Tracking Branch Name
Starting Revision
Revision Expression:
Local Branch
Tracking Branch
Tag
v4
v3
v1.2
v7
v6
v5
v2
v1
Options
Update Existing Branch:
No
Fast Forward Only
Reset
Fetch Tracking Branch
Checkout After Creation
Cancel
Create

push v9分之即可



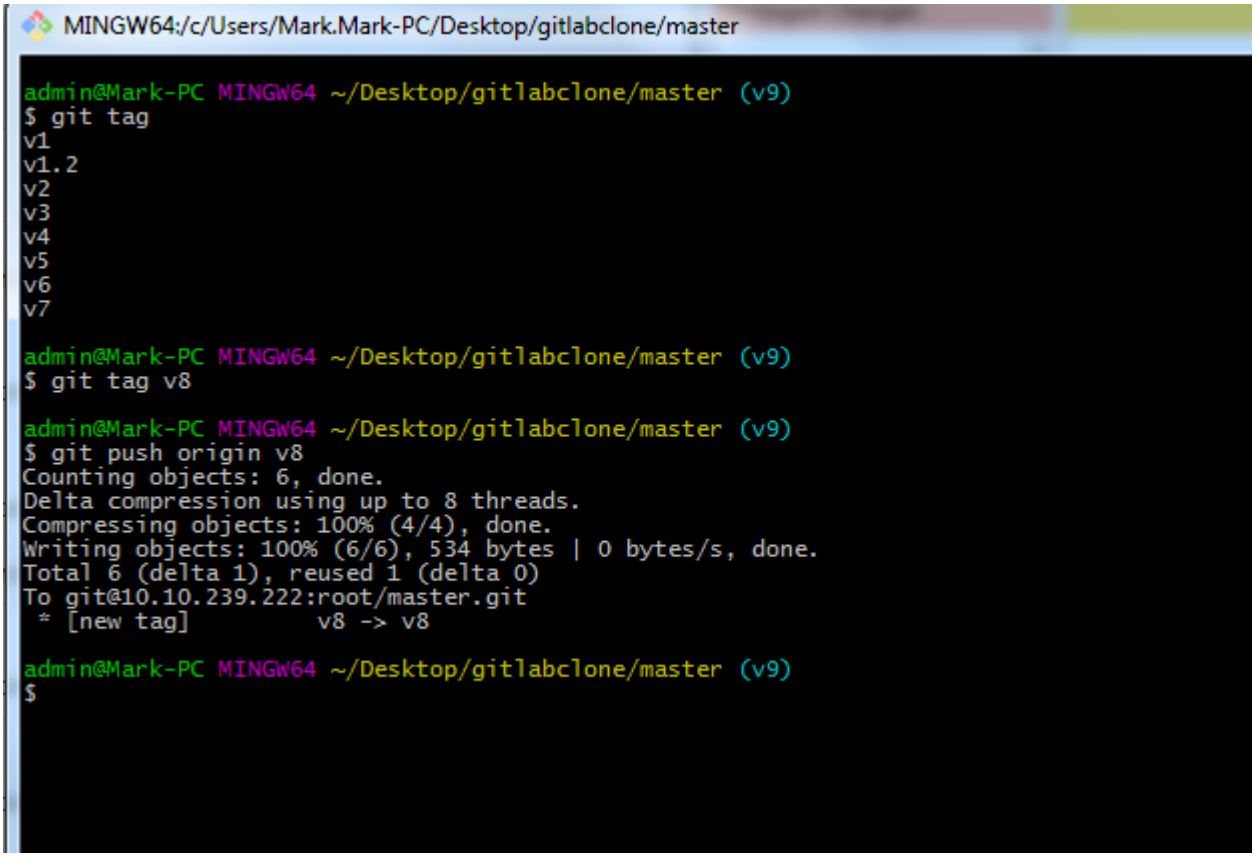
在web中即可看到v9已经上传

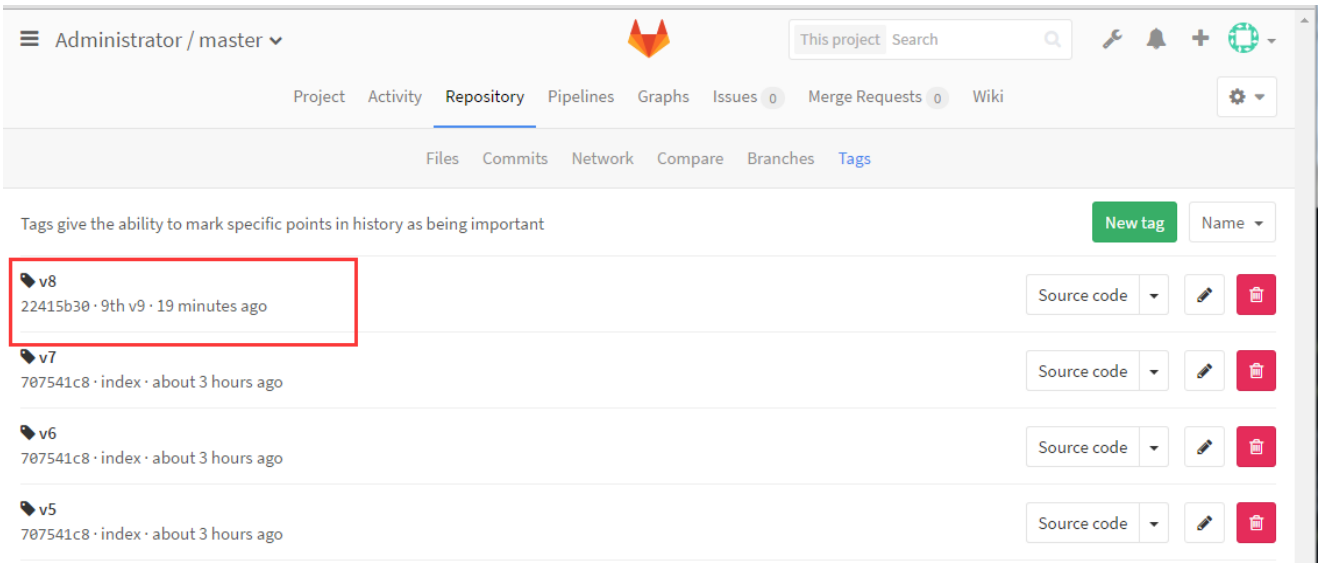


四，标签

4.1 简单的标记

我们切换到git目录右击git bash here,如下所示即可





4.2 拉取标签

```
$ git clone git@10.10.239.222:root/master.git
Cloning into 'master'...
remote: Counting objects: 22, done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 22 (delta 3), reused 0 (delta 0)
Receiving objects: 100% (22/22), done.
Resolving deltas: 100% (3/3), done.
Checking connectivity... done.

$ git checkout v8
Note: checking out 'v8'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at 22415b3... 9th v9

admin@Mark-PC MINGW64 ~/Desktop/gitlabclone/master1/master ((v8))
$ ls
3th.txt  dev.txt  mark.txt  tag10.txt  tag9.txt
```

123