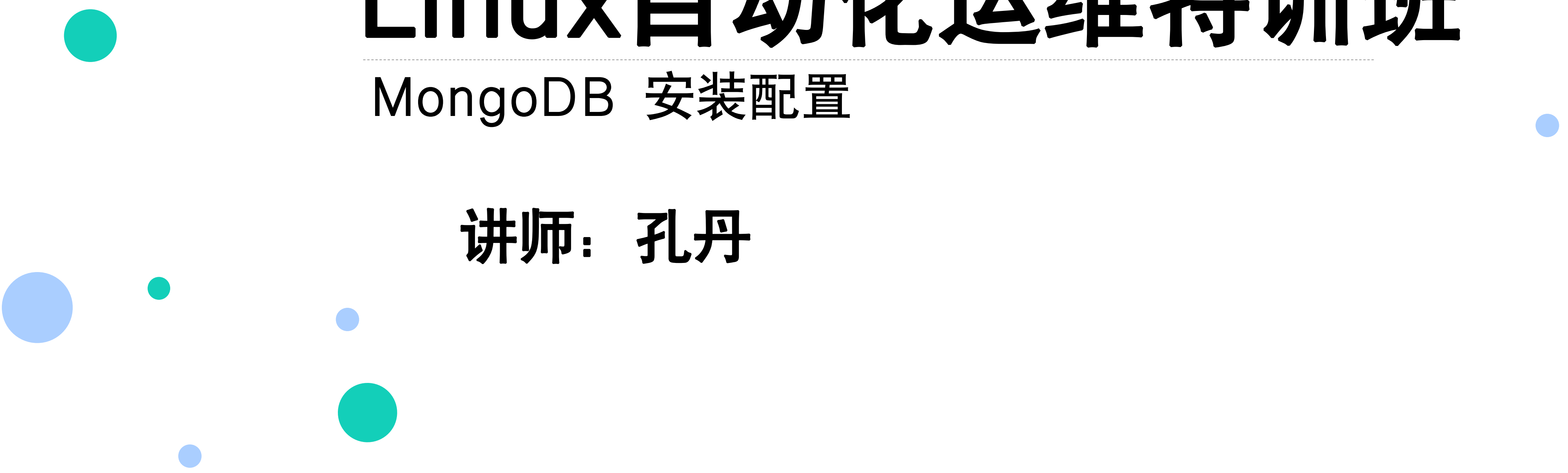




Linux自动化运维特训班

Gitlab使用

讲师：孔丹



大纲

- Gitlab简介
- Gitlab安装
- Gitlab基本配置
- Gitlab使用
- Gitlab备份恢复

Gitlab简介

- GitLab是利用 Ruby on Rails 一个开源的版本管理系统，实现一个自托管的Git项目仓库，可通过Web界面进行访问公开的或者私人项目。它拥有与Github类似的功能，能够浏览源代码，管理缺陷和注释。可以管理团队对仓库的访问，它非常易于浏览提交过的版本并提供一个文件历史库。团队成员可以利用内置的简单聊天程序(Wall)进行交流。它还提供一个代码片段收集功能可以轻松实现代码复用，便于日后有需要的时候进行查找。
- 一个基于GIT的源码托管解决方案
- 基于Ruby on rails开发
- 集成了nginx postgresQL redis sidekiq等组件

Gitlab简介

□ Git有三个类分别为

Git: 是本地版本控制系统工具。

Github: 是官方在线代码托管仓库，可自行注册使用，
<https://github.com>。

GitLab: 是自建的一个非官方的自有版本仓库。通俗一点就是公司内部。

官网: <https://about.gitlab.com/downloads>

清华镜像: <https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7/>

Gitlab安装

□ 安装

```
# yum install
```

```
https://mirrors.tuna.tsinghua.edu.cn/gitlab-  
ce/yum/el7/gitlab-ce-12.5.2-ce.0.el7.x86_64.rpm
```

□ 配置

```
vim /etc/gitlab/gitlab.rb
```

将external_url变量的地址修改为gitlab所在centos的ip地址。

因为修改了配置文件，故需要重新加载配置内容。

```
gitlab-ctl reconfigure
```

```
gitlab-ctl restart
```

查看日志: # gitlab-ctl tail

建议gitlab机器内存>=4G

Git安装

□ 访问 GitLab页面

第一次访问，系统会要求重置新密码，管理员的用户名为root

Please create a password for your new account.



GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Change your password

New password

.....

Confirm new password

.....

Change your password

Didn't receive a confirmation email? [Request a new one](#)

Already have login and password? [Sign in](#)

GitLab的命令

□ 语法：
gitlab-ctl command (subcommand)

Service Management Commands	
start	启动所有服务
stop	关闭所有服务
restart	重启所有服务
status	查看所有服务状态
tail	查看日志信息
service-list	列举所有启动服务
graceful-kill	平稳停止一个服务

GitLab的命令

□ 语法:
gitlab-ctl command (subcommand)

General Commands	
help	帮助
reconfigure	修改配置文件之后，需要重新加载下
show-config	查看所有服务配置文件信息
uninstall	卸载这个软件
cleanse	删除gitlab数据，重新白手起家

GitLab的组件

□ 组件

nginx: 静态Web服务器

gitlab-shell: 用于处理Git命令和修改authorized keys列表

gitlab-workhorse: 轻量级的反向代理服务器

logrotate: 日志文件管理工具

postgresql: 数据库

redis: 缓存数据库

sidekiq: 用于在后台执行队列任务（异步执行）

unicorn: GitLab Rails应用是托管在这个服务器上面的。

目录:

/var/opt/gitlab/git-data/repositories/root: 库默认存储目录

/opt/gitlab: 应用代码和相应的依赖程序

/var/opt/gitlab: gitlab-ctl reconfigure命令编译后的应用数据和配置文件, 不需要人为修改配置

/etc/gitlab: 配置文件目录

/var/log/gitlab: 此目录下存放了gitlab各个组件产生的日志

/var/opt/gitlab/backups/: 备份文件生成的目录

GitLab的组件

□ 变更主配置文件:

需要以下操作

1、gitlab-ctl reconfigure
件

2、gitlab-ctl show-config
件

3、gitlab-ctl restart
gitlab服务

重置配置文

验证配置文

重启

Gitlab邮箱配置

□ 修改配置文件

```
# vim /etc/gitlab/gitlab.rb
gitlab_rails['time_zone'] = 'Asia/Shanghai'
gitlab_rails['gitlab_email_display_name'] = 'gitlab'
gitlab_rails['gitlab_email_from'] = 'esen_monitor@163.com'
gitlab_rails['smtp_enable'] = true
gitlab_rails['smtp_address'] = "smtp.163.com"
gitlab_rails['smtp_port'] = 465
gitlab_rails['smtp_user_name'] = "esen_monitor@163.com"
gitlab_rails['smtp_password'] = "授权码"
gitlab_rails['smtp_domain'] = "smtp.163.com"
gitlab_rails['smtp_authentication'] = "login"
gitlab_rails['smtp_enable_starttls_auto'] = true
gitlab_rails['smtp_tls'] = true
user["git_user_email"] = "esen_monitor@163.com"
```

Gitlab邮箱配置

□ 修改配置文件

重新加载配置

```
gitlab-ctl reconfigure
```

```
#测试配置是否成功
```

```
# gitlab-rails console
```

```
-----  
GitLab:          12.5.2 (49482945d28)
```

```
GitLab Shell: 10.2.0
```

```
PostgreSQL:    10.9
```

```
-----Loading
```

```
production environment (Rails 5.2.3)
```

```
irb(main):001:0> Notify.test_email('esen_monitor@163.com',  
'test mail', 'test mail').deliver_now
```

Gitlab使用流程

- 1. 账号申请
- 2. 客户端安装
- 3. SSH key使用
- 4. 新建项目规则
- 5. 项目检出check
- 6. 创建分支branch
- 7. 代码提交Commit
- 8. 代码拉取Pull
- 9. 代码推送Push
- 10. 代码标签tag
- 11. 代码冲突解决
- 12. 创建忽略文件
- 13. Git Flow实战

Gitlab使用

□ 账号申请

1.开发人员提供以下资料给管理员:

姓名 (用于展示用户姓名)

邮箱 (用于接收密码接收推送通知等)

2.收到重置密码邮件以后进行密码重置

密码需要设置8位以及以上, 建议使用自己的姓名+数字组成比如

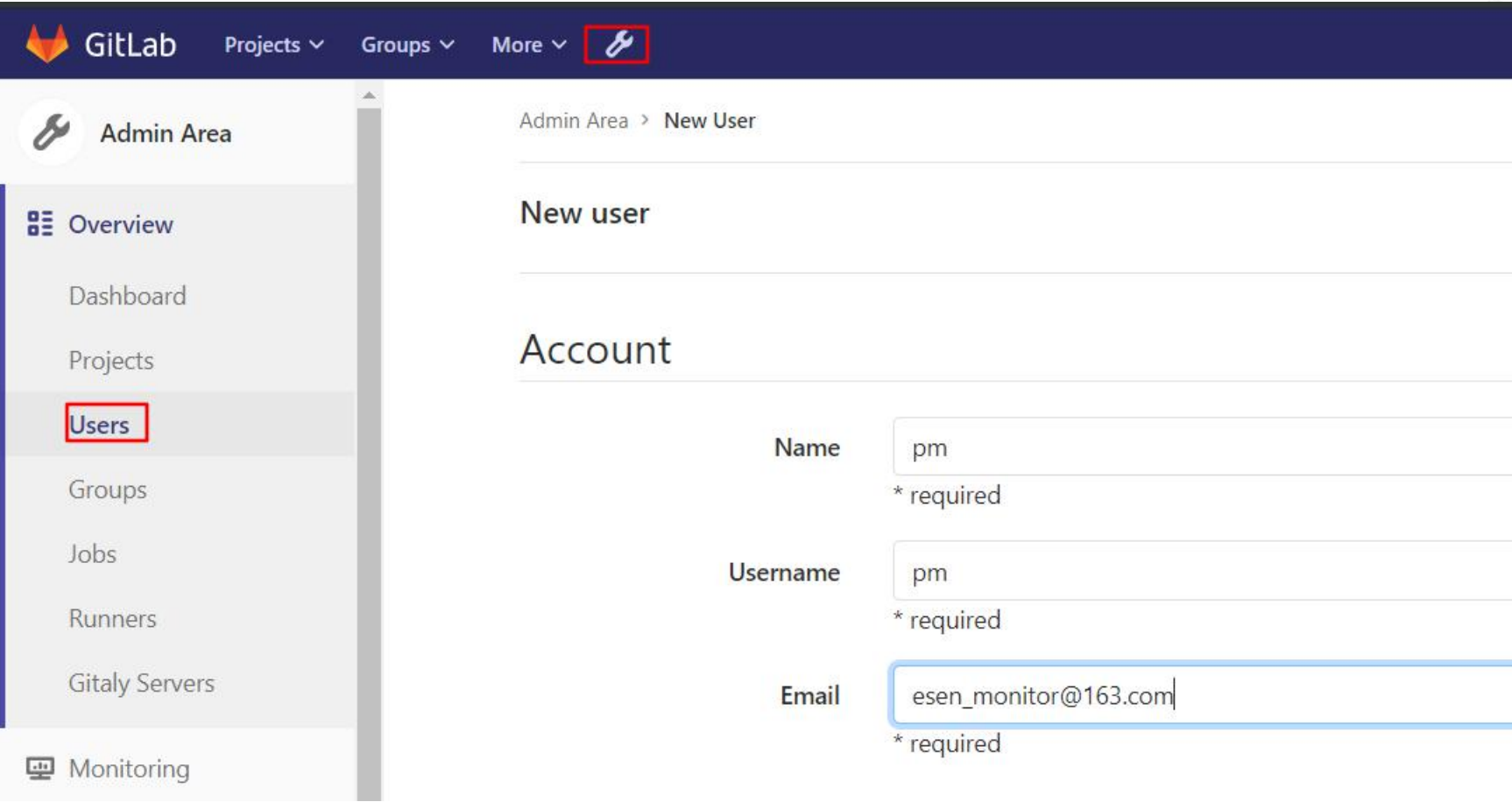
3.登录

Gitlab使用

□ 账号申请
创建组，组对应公司里开发团队




创建账号
分别创建用户pm（项目经理）、tom（开发）、bob（开发）
账户创建后默认密码会发送邮件，第一次登录要求修改密码



Gitlab使用

□ 账号申请


测试时，没有那么多邮箱，也可以管理员为用户设置密码


 GitLab


Projects ▾

Groups ▾

More ▾



 Admin Area

 Overview

Dashboard

Projects


Users


Groups


Jobs


Runners


Gitaly Servers

 Monitoring

 Messages

 System Hooks

 Applications

 Abuse Reports 0

Admin Area > Users

Edit user: tom

Account

Name

tom

* required

Username

tom

* required

Email

sxkongdan@tom.com

* required

Password

Password

.....

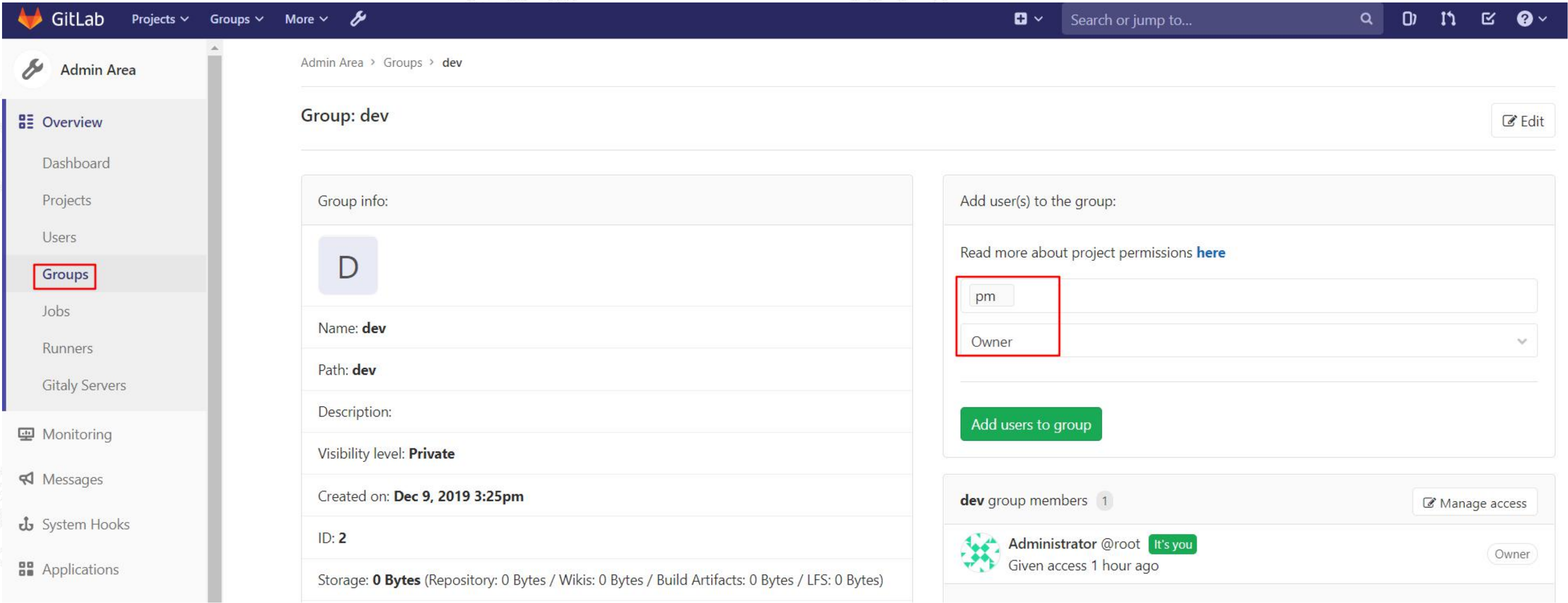
Password confirmation

.....

Gitlab使用

□ 账号申请

用户加入组： pm用户作为组的所有者



Gitlab使用

□ 客户端安装

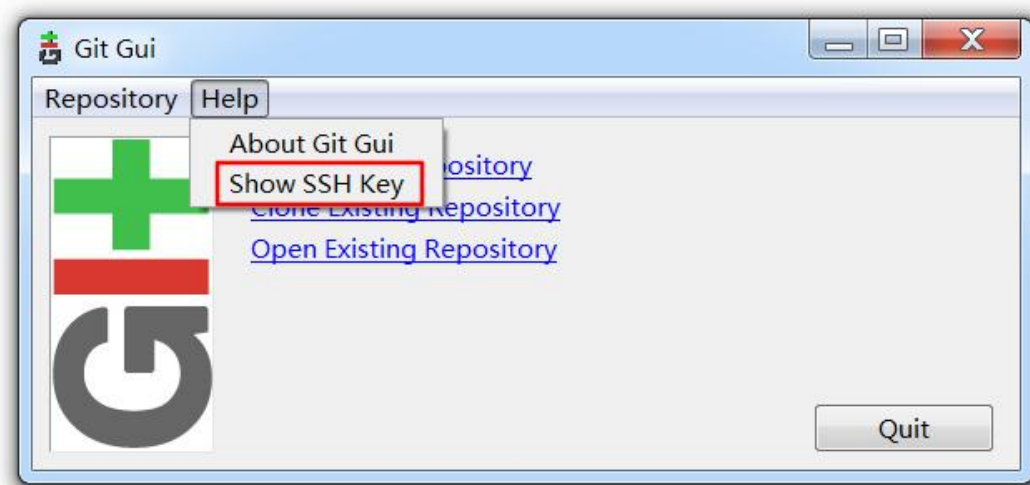
首先安装git，安装过程略。
然后安装TortoiseGit。

Gitlab使用

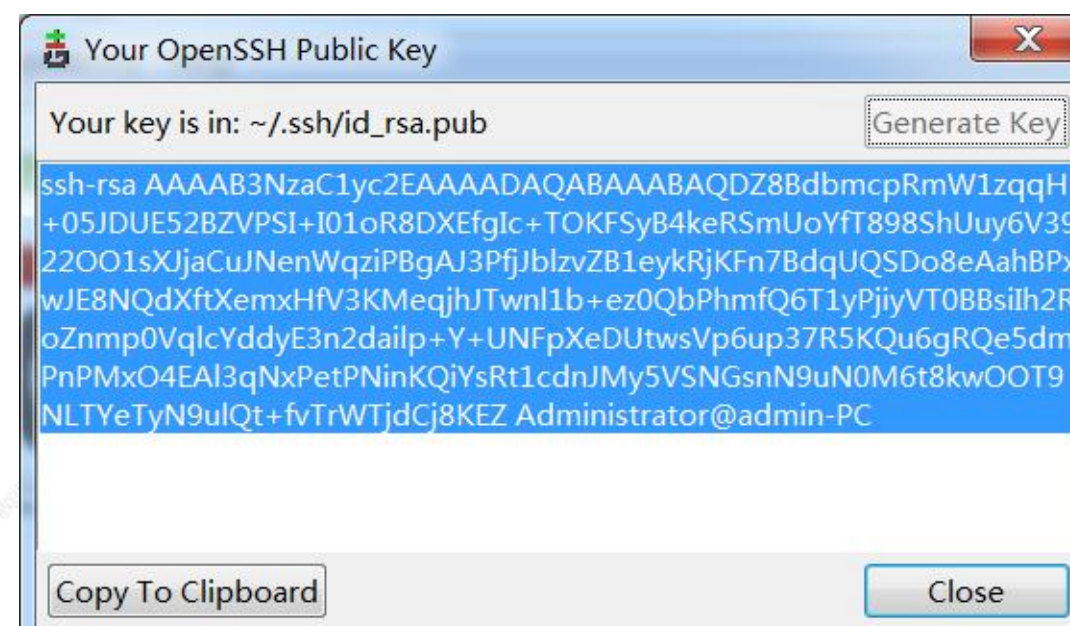
□ ssh key

生成SSHkey。

创建一个SSH Key，在任意文件夹下点击右键，选择 Git GUI Here。
在弹出的程序中选择主菜单的【Help】→【Show SSH Key】：



如果没有Key，则点击“Generate Key”生成一个SSH Key：



命令行方式创建：

```
$ ssh-keygen.exe -f ~/.ssh/id_rsa -P " " -C sxkongdan@tom.com -q
```

Gitlab使用

□ ssh key

Gitlab 配置SSH keys。

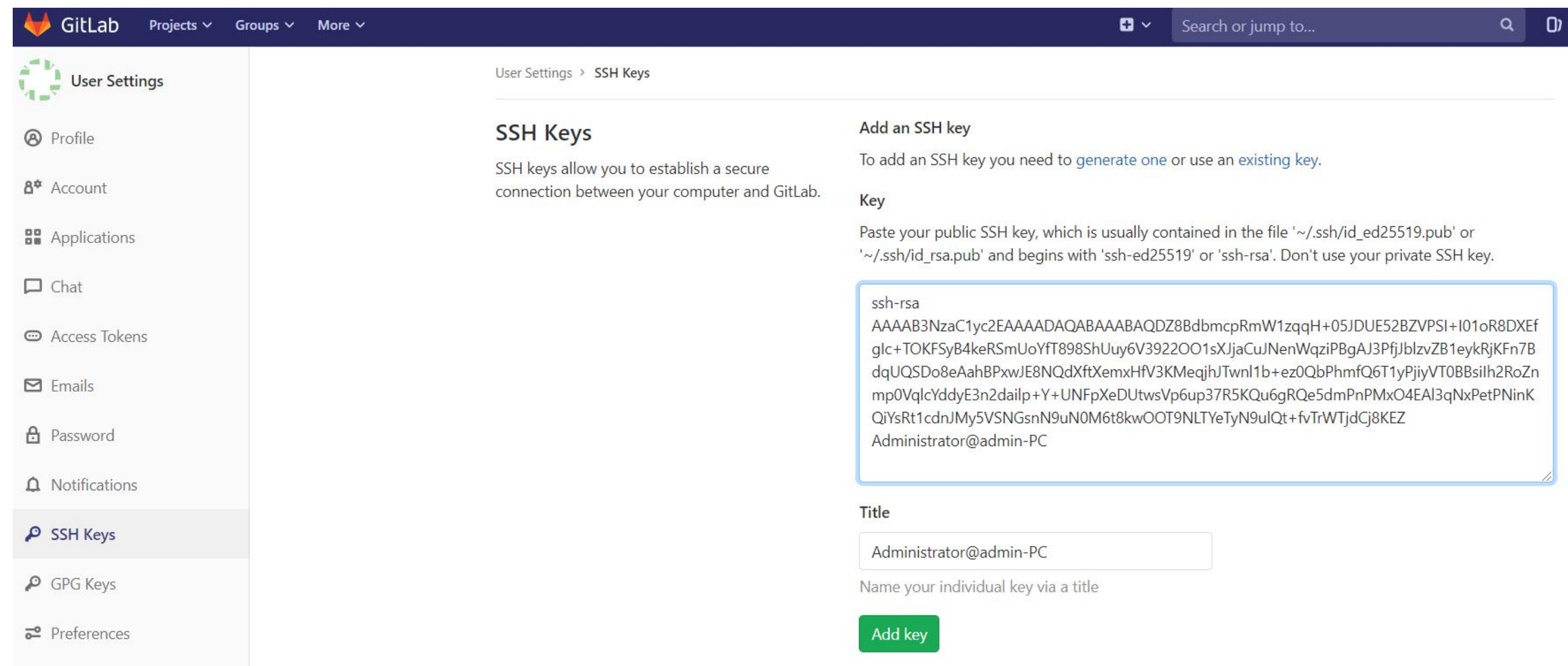
打开Gitlab登录自己的账户，

1.进入用户设置， 找到SSHkeys

2.点击右侧 Add SSH Key

3.输入上一步生成的key

4.点击Add key 即添加成功一个key







Gitlab使用

□ 新建项目规则

创建项目组


使用之前创建的项目组dev


继续添加用户bob和tom，开发人员


dev group members 4		Manage access
	pm @pm Given access 5 minutes ago	Owner
	Administrator @root It's you Given access 1 hour ago	Owner
	tom @tom Given access right now	Developer
	bob @bob Given access just now	Developer

Gitlab使用

- 新建项目规则
- 创建新项目：创建一个属于dev组的项目app1

 GitLab

Projects ▾ Groups ▾ More ▾ 

 ▾ Search or jump to...

New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), [among other things](#).

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

Information about additional Pages templates and how to install them can be found in our [Pages getting started guide](#).

Tip: You can also create a project from the command line. [Show command](#)

Blank project

Create from template

Import existing project

Project name

app1

Project URL


Project slug


http://192.168.150.16/ dev app1

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


Project description (optional)


Description format

Visibility Level 

☒  Private

Project access must be granted explicitly to each user.

☐  Internal

☐  Public

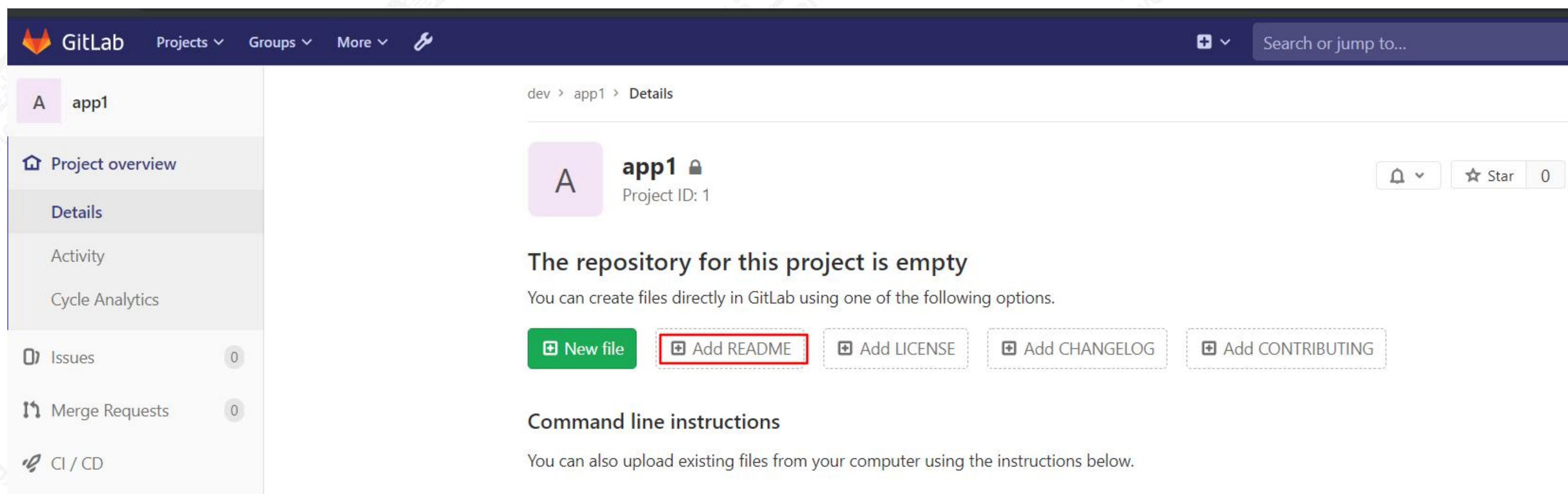
☐ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project

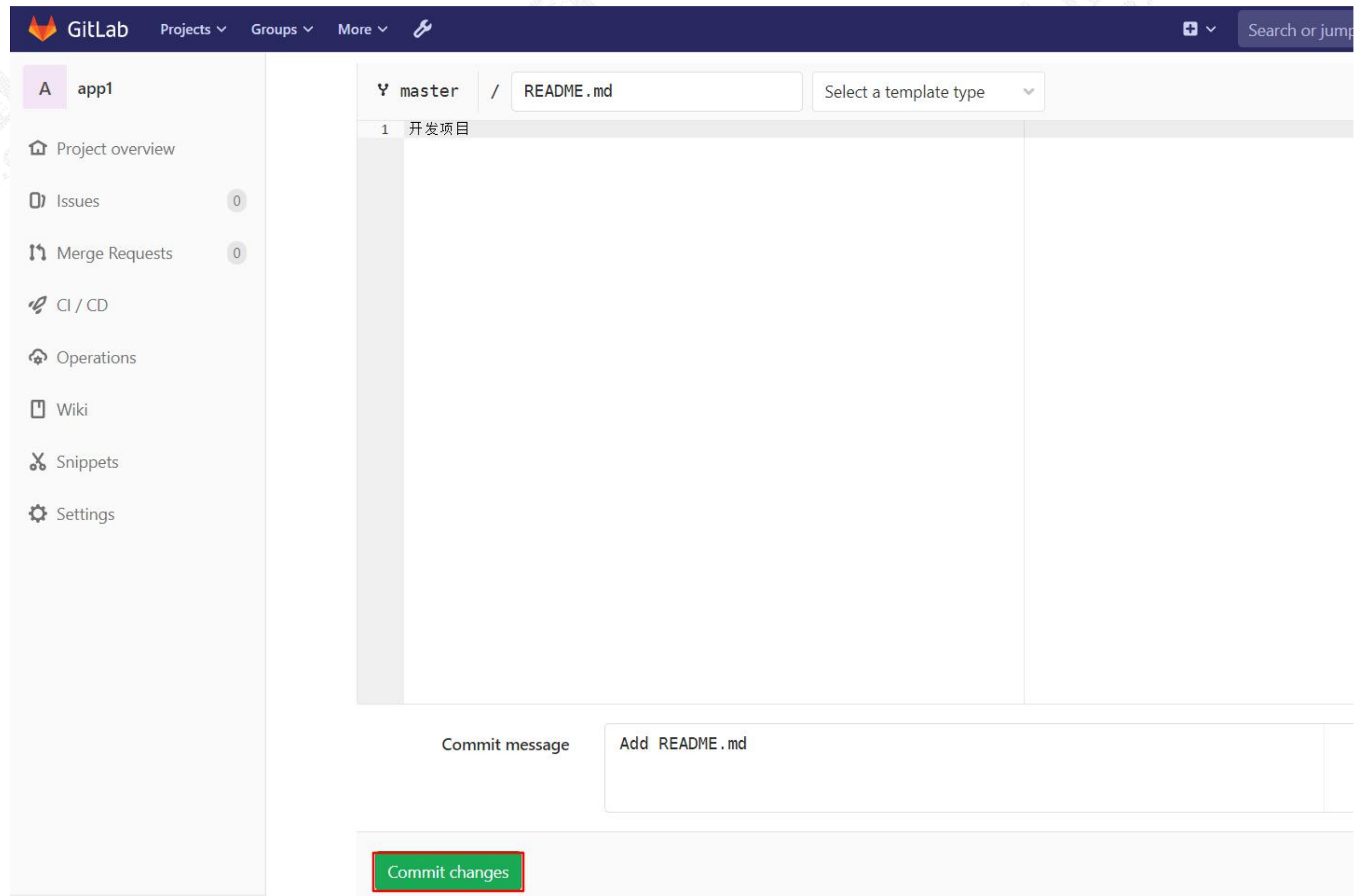
Gitlab使用

- 新建项目规则
为项目创建master



Gitlab使用

□ 新建项目规则 为项目创建master



Gitlab使用

- 新建项目规则
项目经理制定开发计划

app1	需求	开发者	完成日期
v1.0	首页	tom	2.5
	新闻	tom	2.7
	支付	bob	2.7
	博客	bob	2.8

Gitlab使用

- 新建项目规则
 - 项目经理制定开发计划--创建里程碑

GitLab Projects Groups More

Search or jump to...

dev > app1 > Milestones > New

New Milestone

Title: 官网v1.0

Start Date: 2019-12-09

Due Date: 2020-03-13

Description: 官网v1.0

Markdown is supported

Attach a file

Create milestone

Gitlab使用

- 新建项目规则
- 项目经理制定开发计划--分配任务

GitLab

Projects

Groups

More

+

▼

Search or jump to...

🔍

0

A

app1

Project overview

Repository

Issues0

List

Boards

Labels

Milestones

Merge Requests0

CI / CD

Operations

Wiki

Snippets

Settings

dev > app1 > Issues > New

New Issue

Title

首页

Add description templates to help your contributors communicate effectively!

Description

WritePreview

B

I

”

</>

🔗

⋮

网站首页开发

Markdown and quick actions are supported

☐

This issue is confidential and should only be visible to team members with at least Reporter access.

Assignee

tom

▼

Assign to me

Due date

2020-02-05

Milestone

官网v1.0

▼

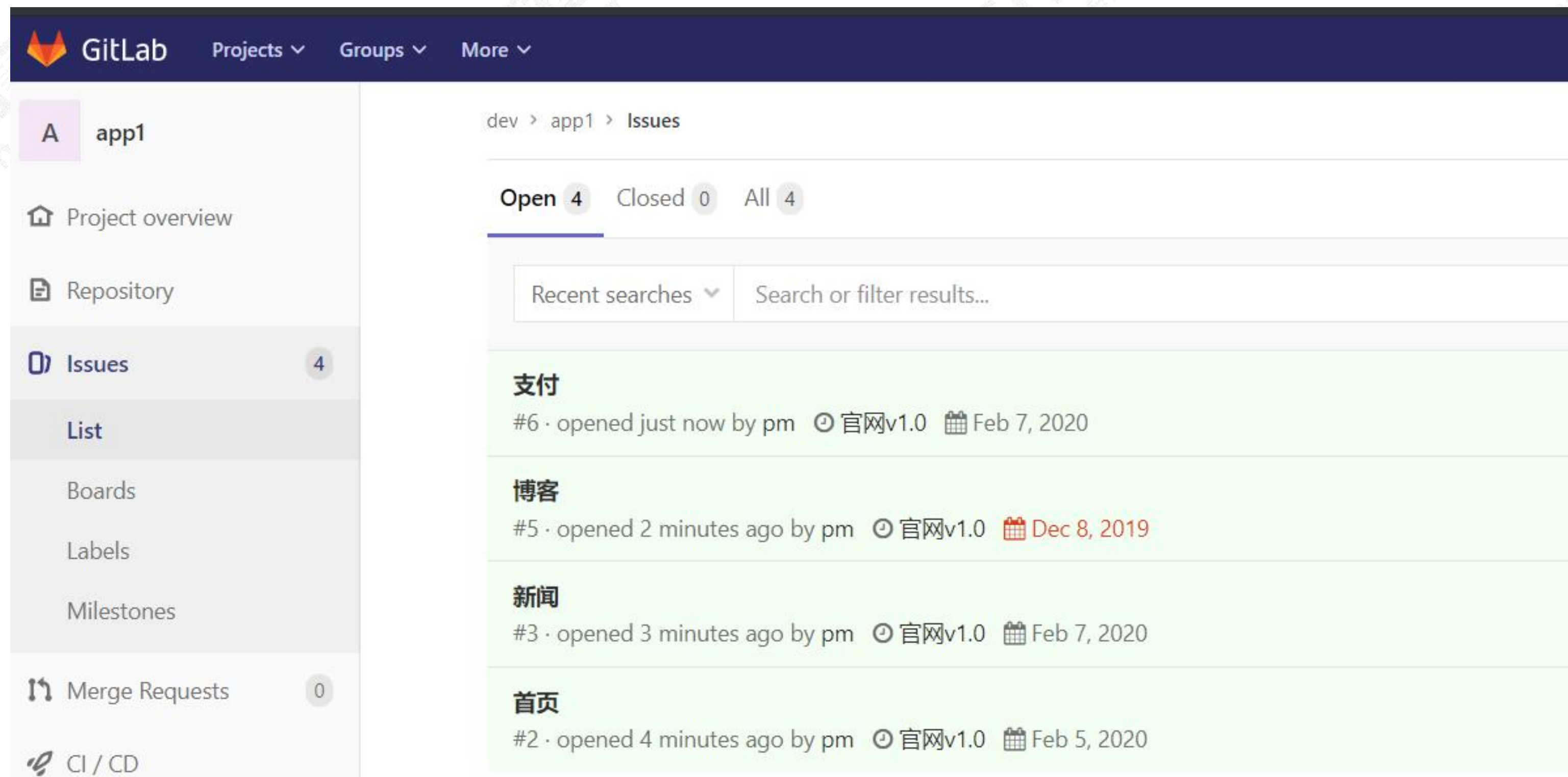
Labels

Labels

▼

Gitlab使用

- 新建项目规则
 - 项目经理制定开发计划--分配任务

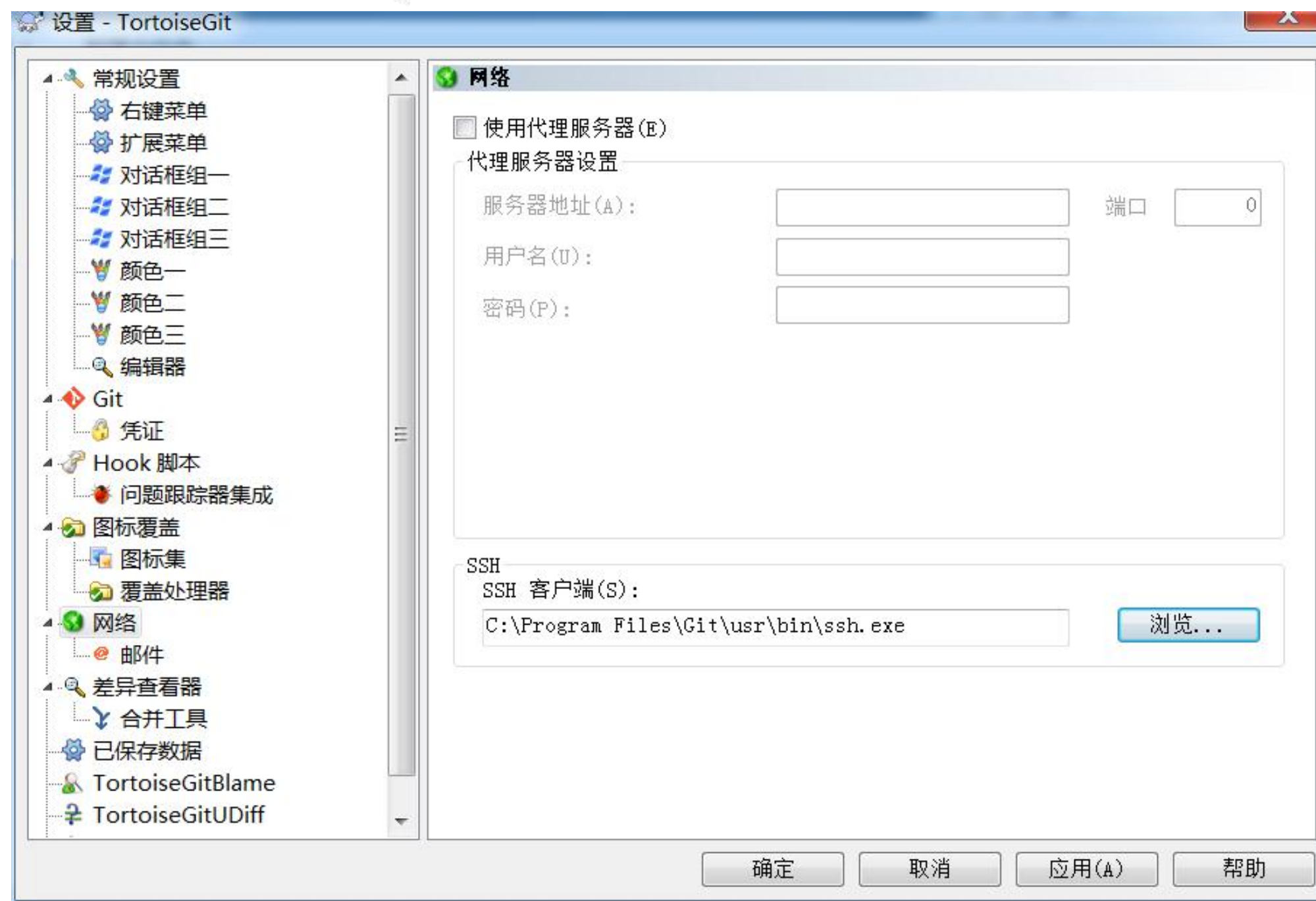


Gitlab使用

□ 项目检出

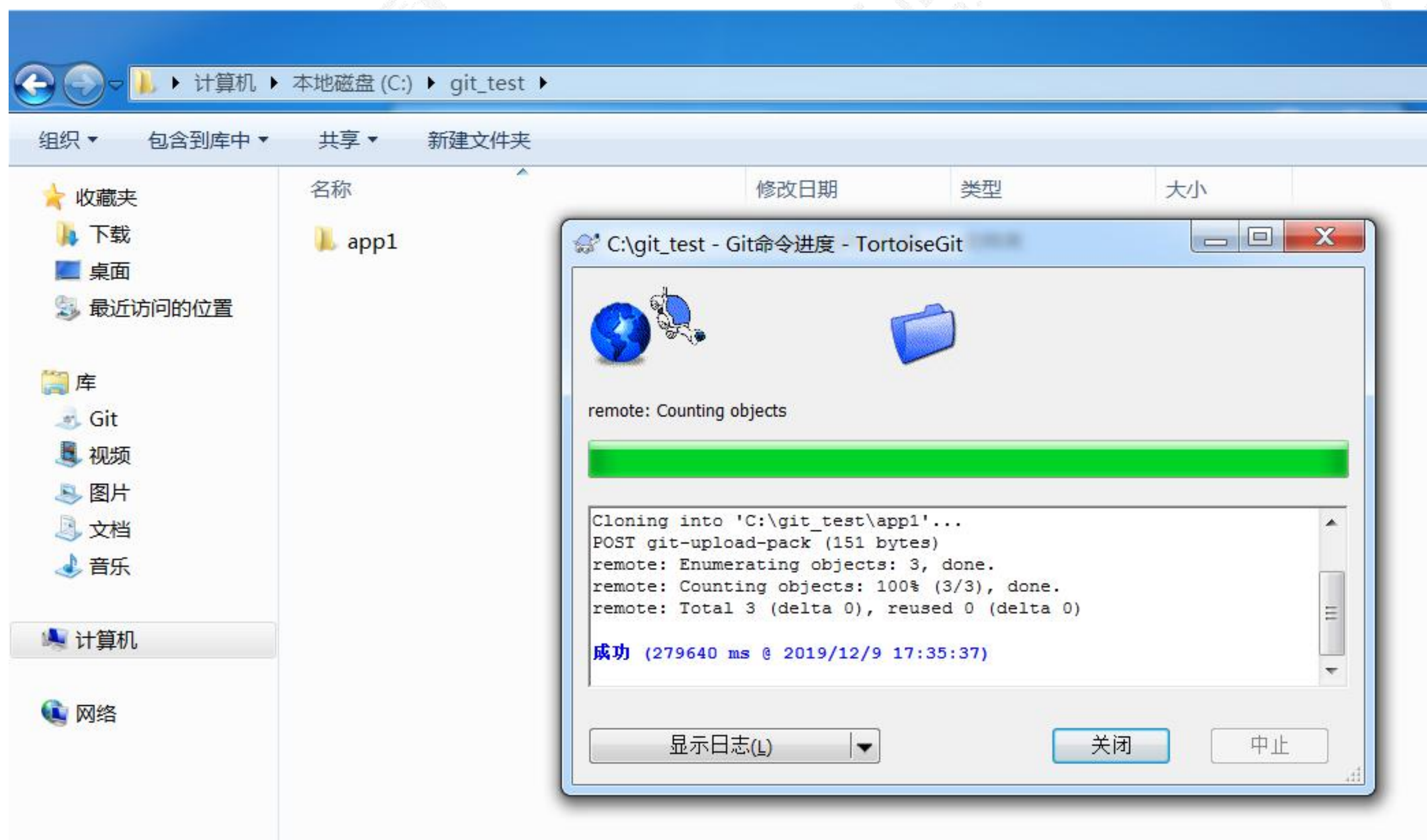
以tom为例：

这里只是需要把TortoiseGit设置网络里面的ssh客户端位置C:\Program Files\TortoiseGit\bin\TortoiseGitPlink.exe，修改成git安装目录下的ssh.exe即可。如果你安装过程已经选择了使用OpenSSH方式我想这里也应该不用修改了。不然拉取等操作的时候都会提示需要输入密码。



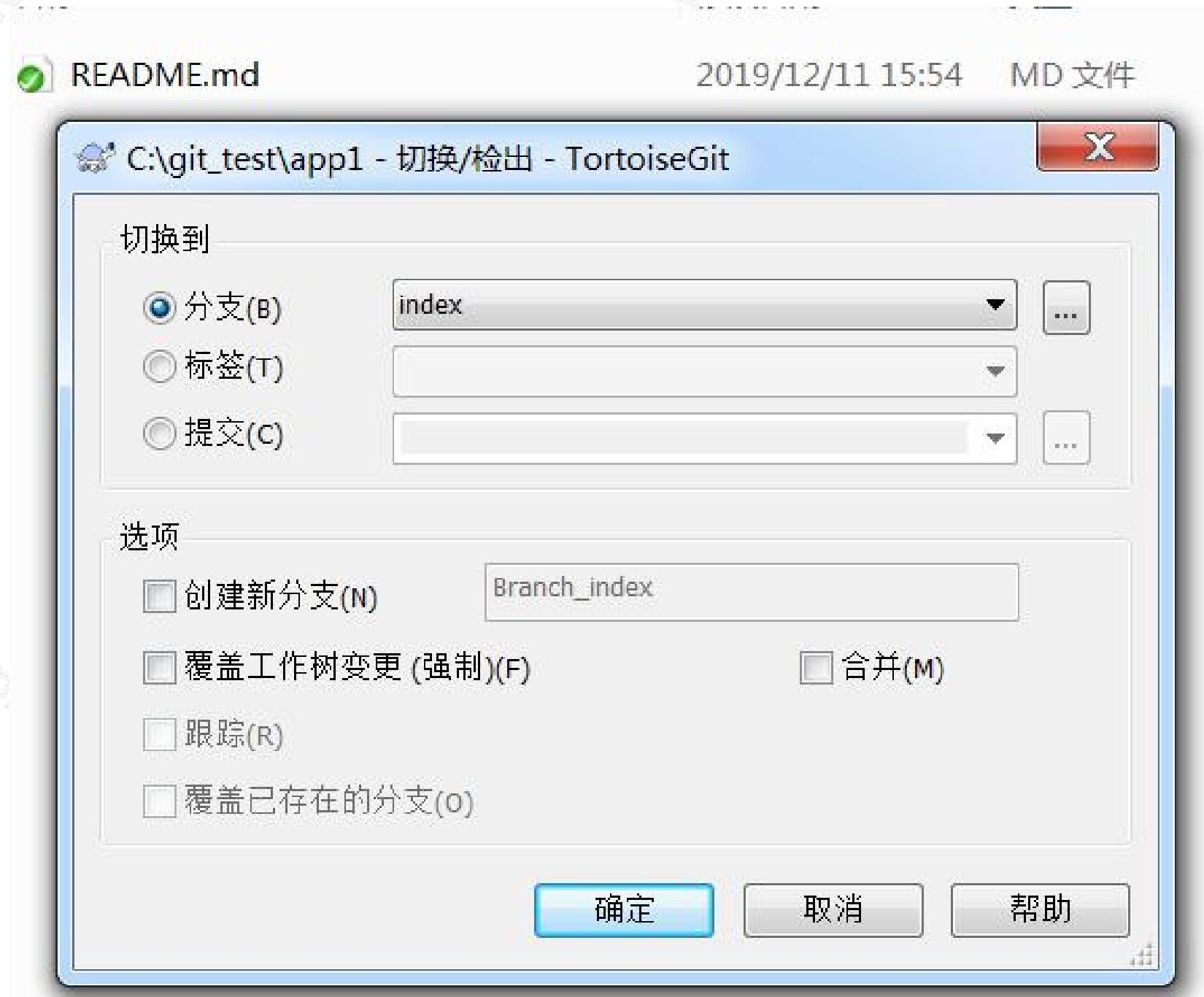
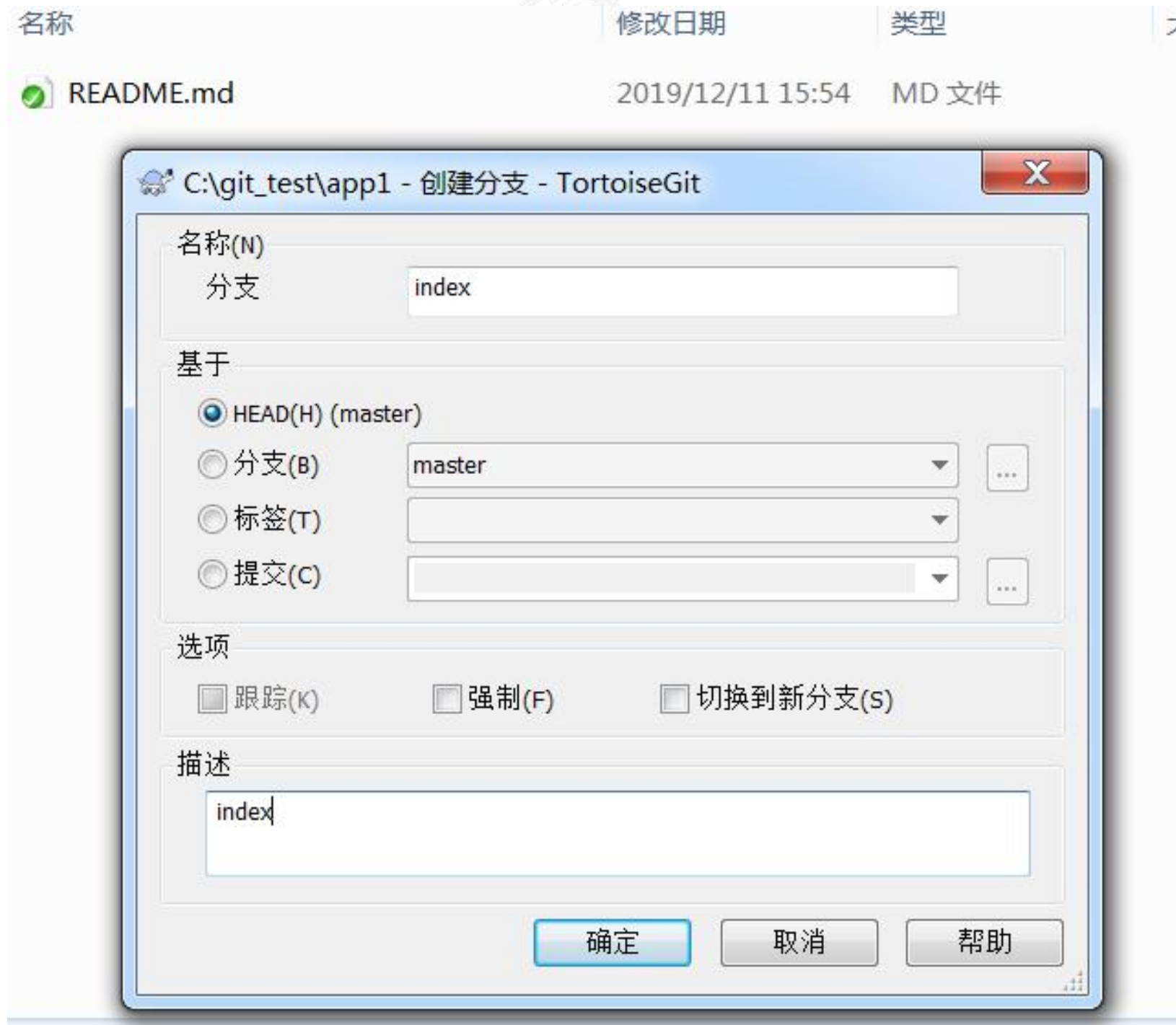
Gitlab使用

□ 项目检出
以tom为例:



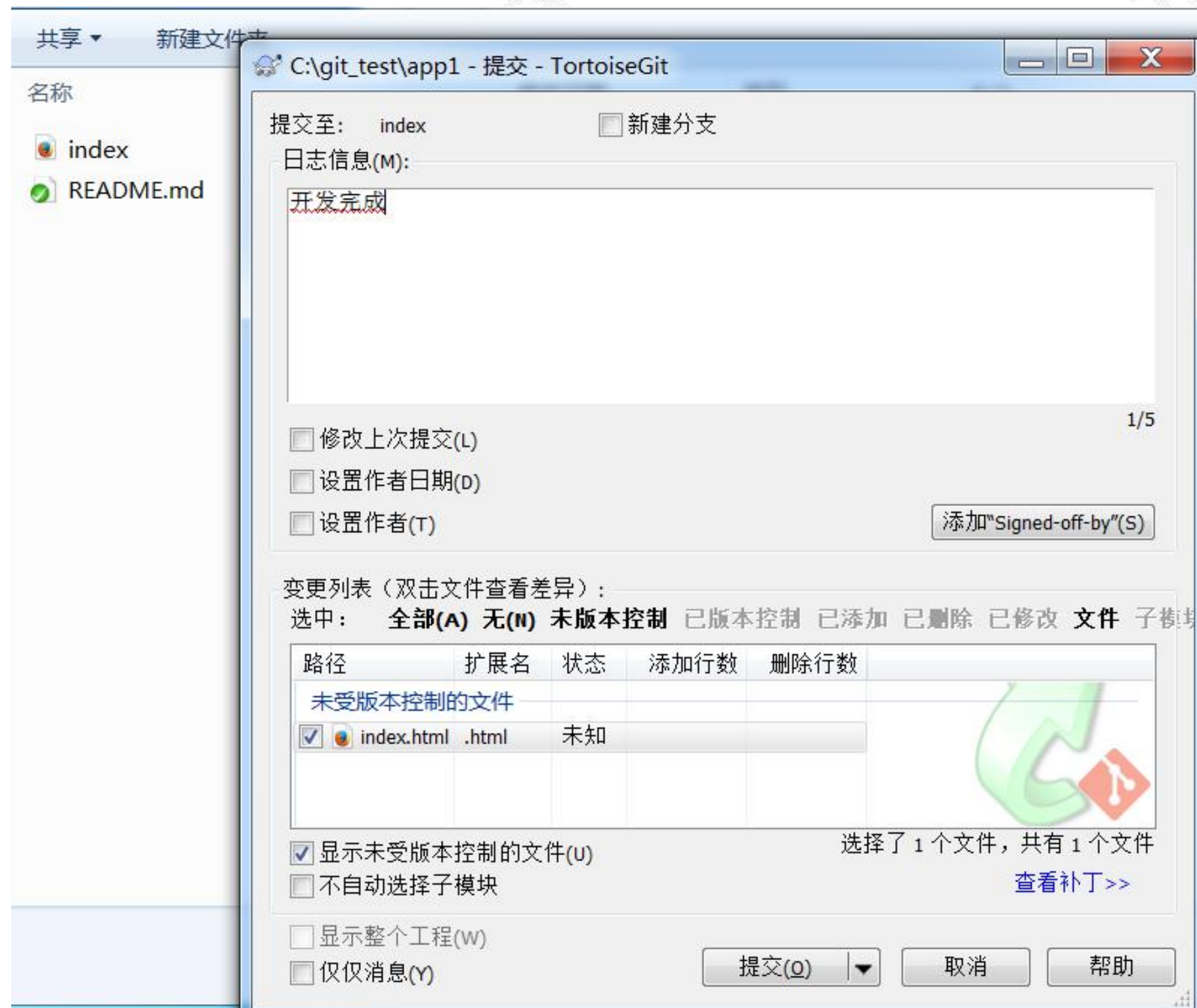
Gitlab使用

□ 创建分支和切换分支
以tom为例：



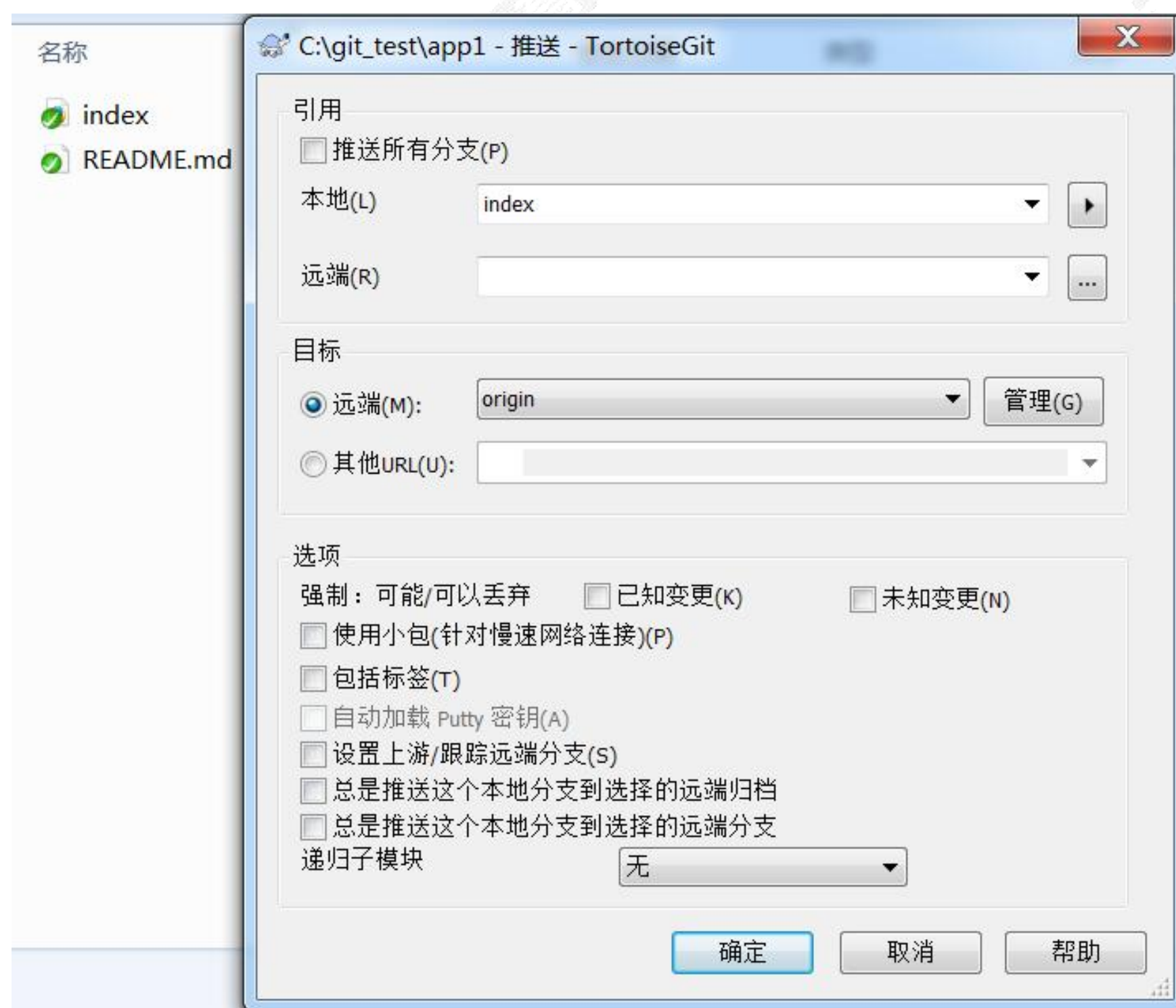
Gitlab使用

□ 提交 以tom为例：



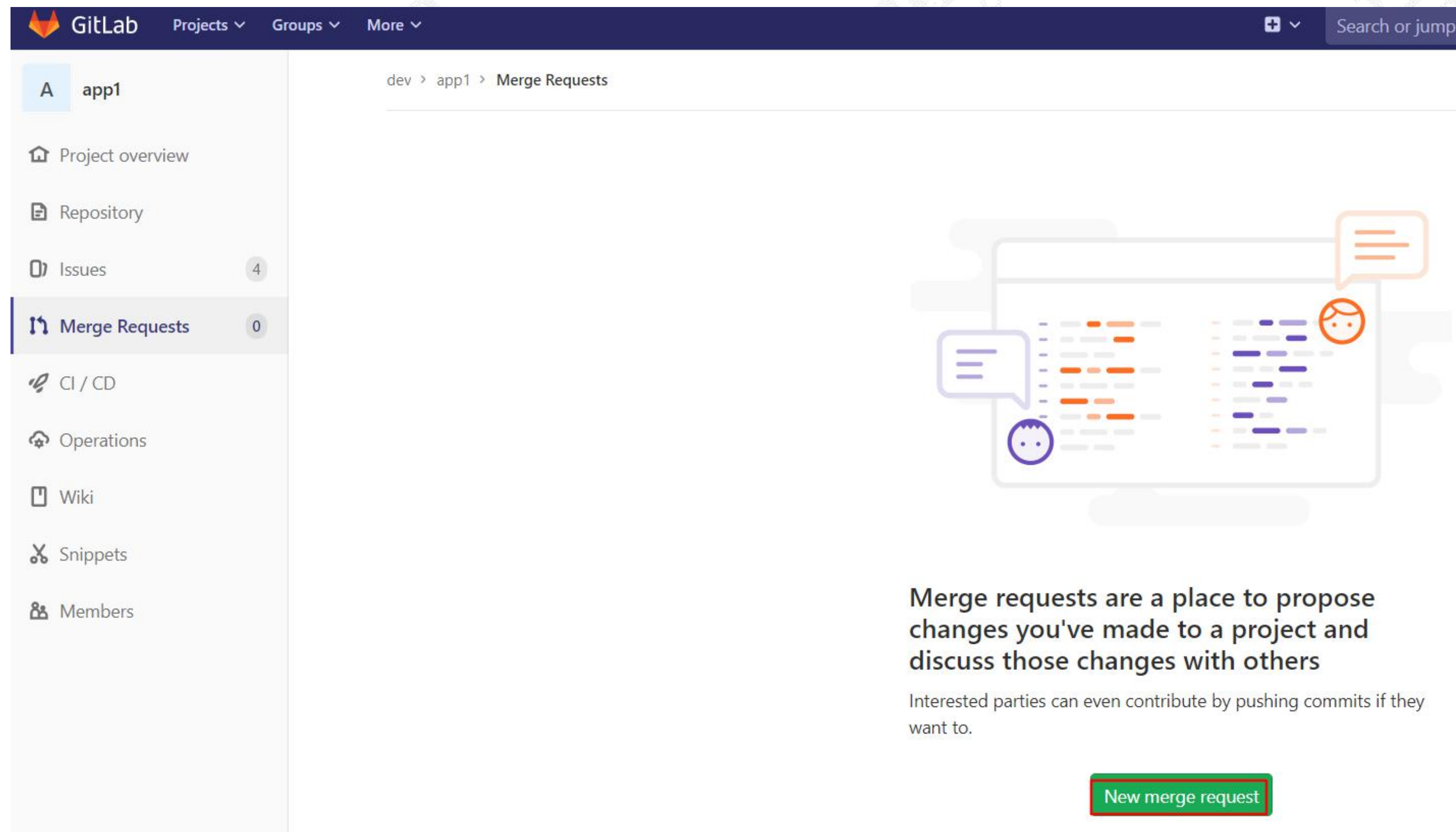
Gitlab使用

□ 推送
以tom为例：



Gitlab使用

□ 发出请求合并 以tom为例：登录gitlab



Gitlab使用

□ 发出请求合并

以tom为例： 登录gitlab

GitLab

Projects ▾ Groups ▾ More ▾

+

Search or jump to...

A app1

Project overview

Repository

Issues 4

Merge Requests 0

CI / CD

Operations

Wiki

Snippets

Members

dev > app1 > Merge Requests > New

New Merge Request

From index into master [Change branches](#)

Title

开发完成

Start the title with WIP: to prevent a **Work In Progress** merge request from being merged before it's ready.

Add [description templates](#) to help your contributors communicate effectively!

Description

Write Preview

Describe the goal of the changes and what reviewers should be aware of.

Markdown and [quick actions](#) are supported

Assigneepm ▾

Milestone官网v1.0 ▾

LabelsLabels ▾

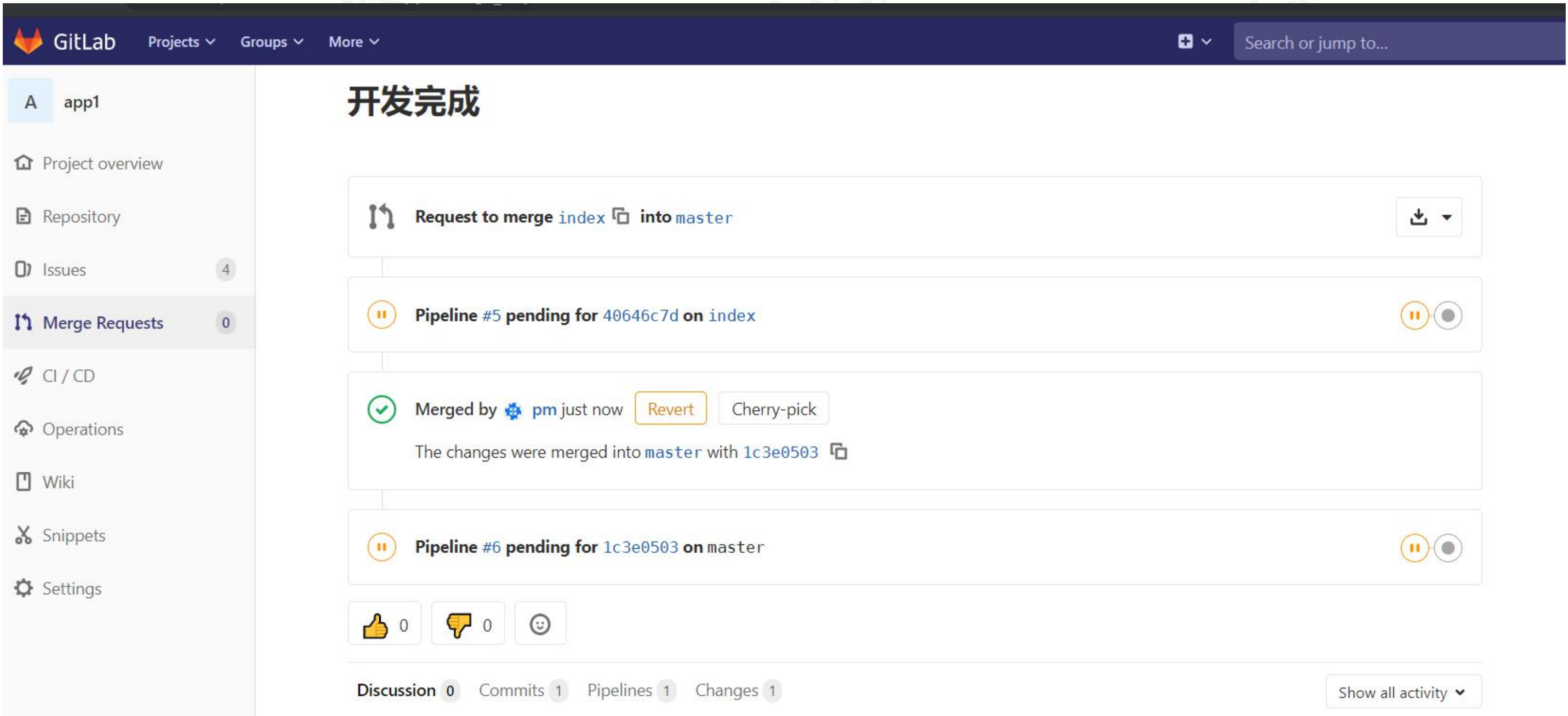
[Assign to me](#)

Merge options

☒ Delete source branch when merge request is accepted.

Gitlab使用

- 确认请求合并
- pm登录后确认合并请求



Gitlab使用

□ 关闭任务
pm关闭任务

The screenshot displays the GitLab web interface. At the top, the navigation bar includes the GitLab logo, 'Projects', 'Groups', and 'More' dropdowns, along with a search bar. The left sidebar shows the project 'app1' and a list of navigation items: 'Project overview', 'Repository', 'Issues' (with 4 items), 'List', 'Boards', 'Labels', 'Milestones', and 'Merge Requests' (with 0 items). The main content area shows the issue '#1' in the 'dev' branch for 'app1'. The issue is 'Open' and was 'Opened 33 minutes ago by pm'. A red box highlights the 'Close issue' button. Below the issue title, there is a '首页' (Home) section with a 'Write' button and a 'Preview' button. The right side of the 'Write' button has a rich text editor with various formatting options like bold, italic, quote, code, link, list, and table. The bottom of the page shows a list of recent activity for the issue, including a change in the due date to February 05, 2020, and a change in the milestone to '%官网v1.0'.

Gitlab使用

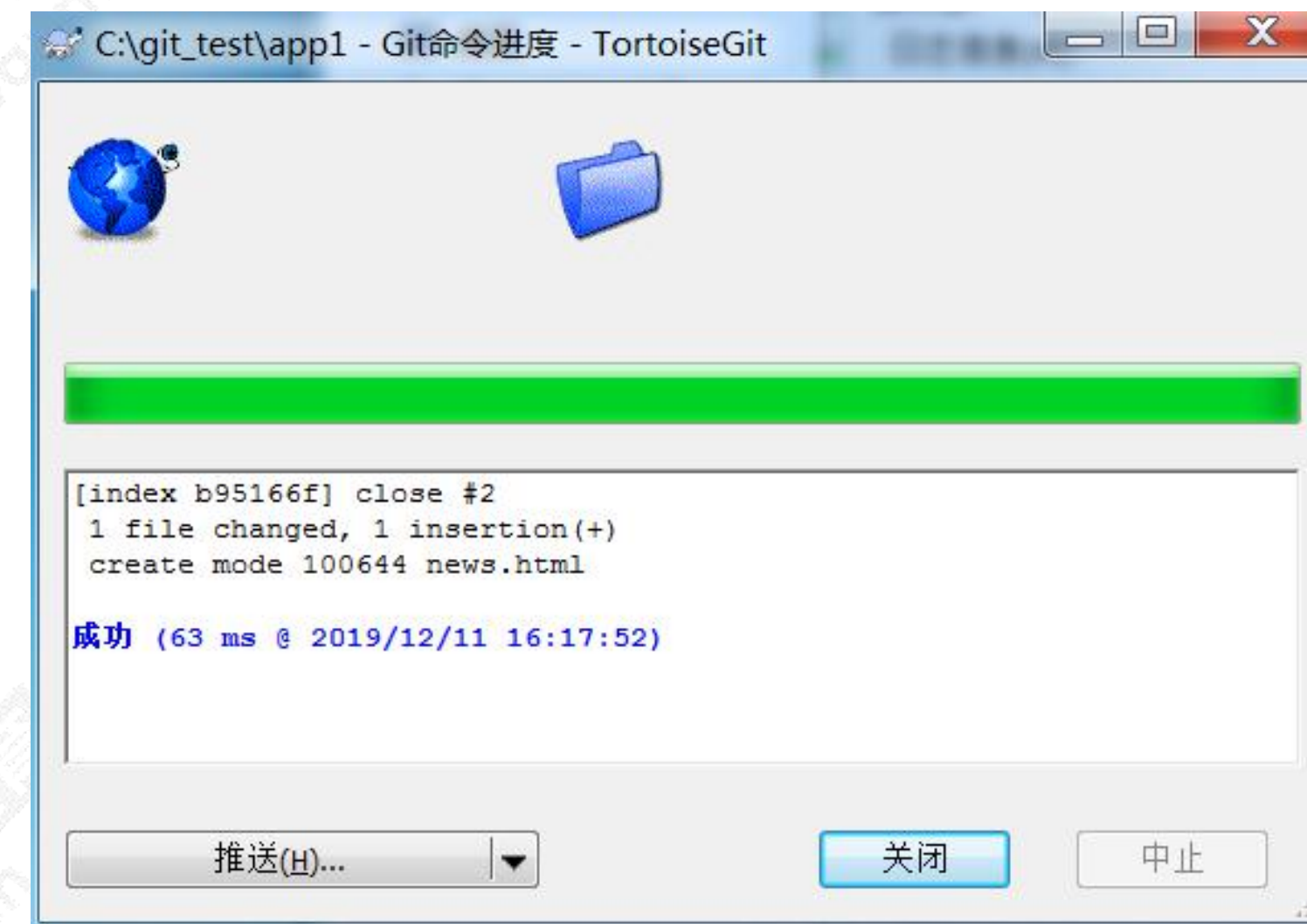
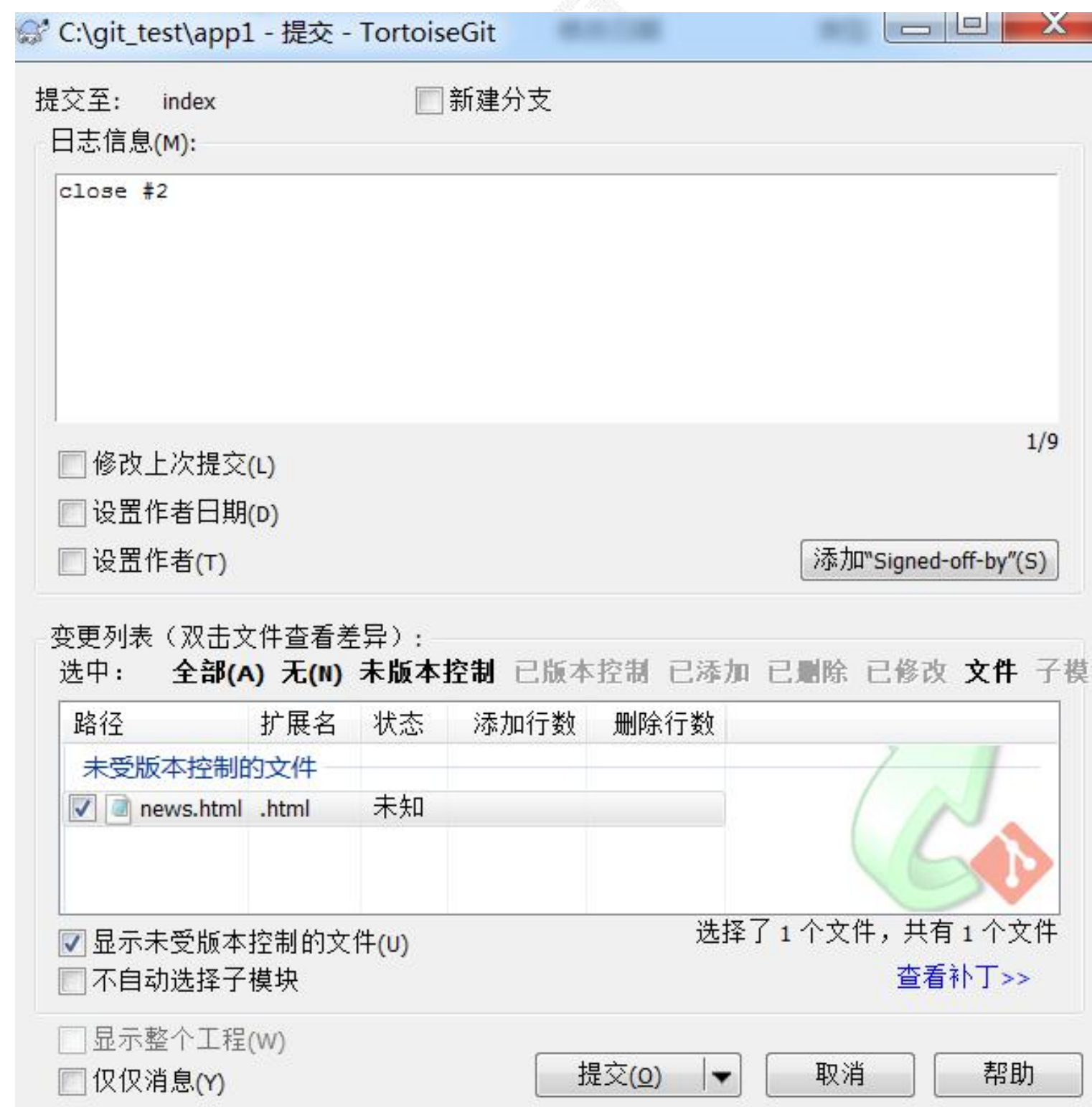
□ 拉取

tom再次开发，需要pull获取最新版本库



Gitlab使用

- 开发news
- tom再次开发，需要pull获取最新版本库



注意：提交时 -m “close #2”，确认请求后会自动关闭问题。

Gitlab备份与恢复

□ 备份管理

配置文件中加入：# vim /etc/gitlab/gitlab.rb

gitlab_rails['backup_path'] = '/data/backup/gitlab'

gitlab_rails['backup_keep_time'] = 604800

如果自定义备份目录需要赋予git权限

mkdir /data/backup/gitlab -p

chown -R git.git /data/backup/gitlab

重置配置：gitlab-ctl reconfigure

设定计划任务：

定时任务Crontab中加入

0 2 * * * /usr/bin/gitlab-rake gitlab:backup:create

策略建议：本地保留三到七天，在异地备份永久保存

手动备份测试：gitlab-rake gitlab:backup:create

ls /data/backup/gitlab/

1576053273_2019_12_11_12.5.2_gitlab_backup.tar

注意：gitlab.rb 和 gitlab-secrets.json 两个文件包含敏感信息。

未被备份到备份文件中。需要手动备份

Gitlab备份与恢复

□ 恢复管理

停止数据写入服务

```
gitlab-ctl stop unicorn
```

```
gitlab-ctl stop sidekiq
```

```
gitlab-rake gitlab:backup:restore
```

```
BACKUP=1576053273_2019_12_11_12.5.2
```

```
# date -d @1576053273
```

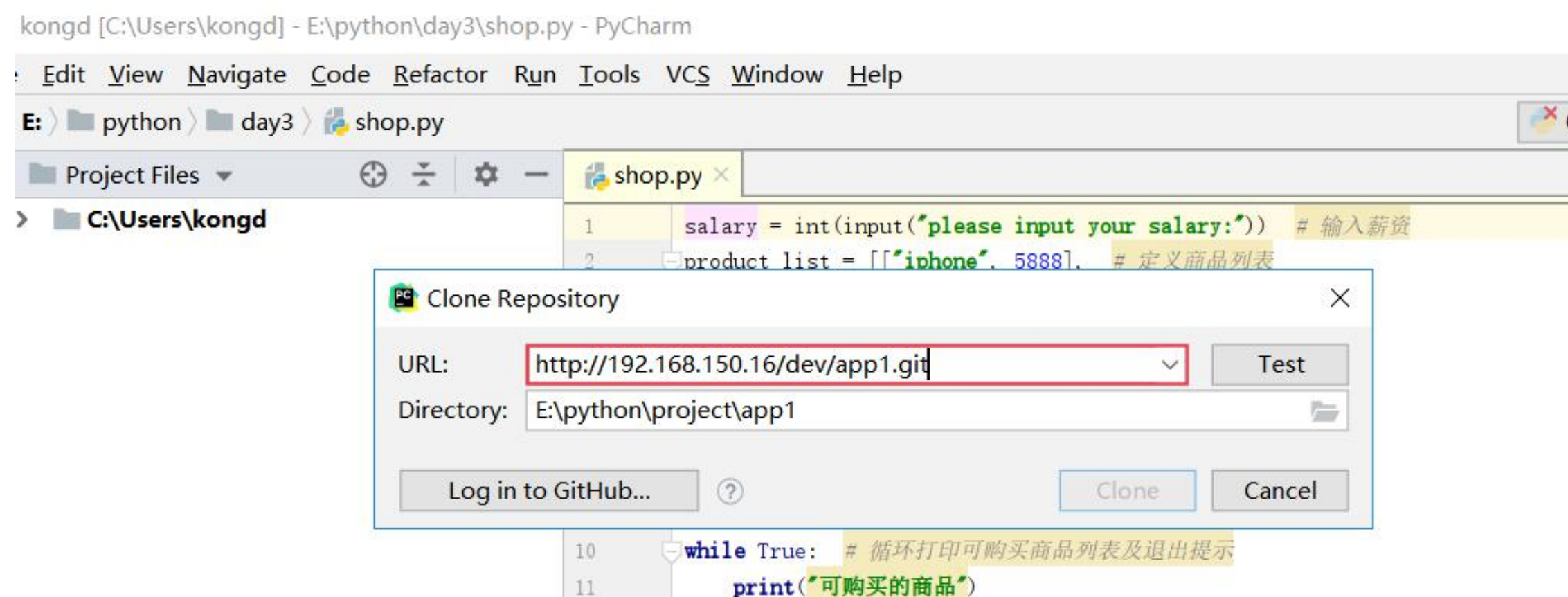
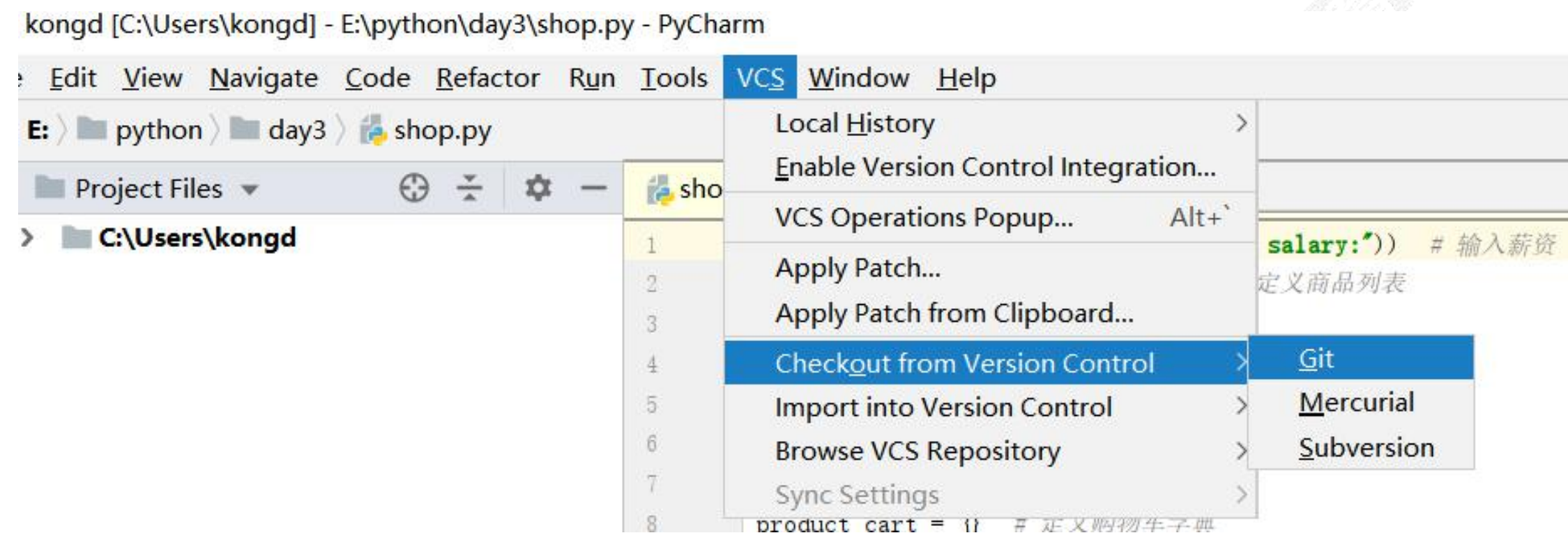
```
2019年 12月 11日 星期三 16:34:33 CST
```

备份tar包一定要放到备份路径下。恢复是删除原有数据，恢复备份tar包中的数据。

如果是在其他服务器恢复备份，一定要记得将 `gitlab.rb` 和 `gitlab-secrets.json` 手动复制到相应路径下。

pycharm使用gitlab

□ 前提：首先需要安装git



总结

- Gitlab简介
- Gitlab安装
- Gitlab基本配置
- Gitlab使用
- Gitlab备份恢复



谢谢观看

更多好课，请关注[万门大学APP](#)

