

Gitlab多人协同工作

Master & Developer

目录

- ❑ 开发前准备
- ❑ 开发中过程
- ❑ Developer操作说明
- ❑ Master操作说明
- ❑ 附录（Gitlab基本操作）

开发前准备

开发前准备

Gitlab多人协同工作，所有开发者会被添加进一个共同的工作组。协同工作开始前，群组管理员及开发者都有一些准备工作要做。

群组管理员（Master）：

- 1 - 在Gitlab上创建群组
- 2 - 为群组添加成员并赋予权限
- 3 - 为群组新建项目
- 4 - 为群组项目创建分支

开发者（Developer）：

1. 去Gitlab注册个人账号并添加个人SSH Key（参照本PPT附录）
2. 请群组管理员将你的账号添加进open-source群组

开发中过程

开发中过程

日常开发中开发者与群组管理员的分工。

开发者（Developer）：

1. 克隆远程仓库到本地（只在第一次做一次）
2. 获取远程最新版本并merge到本地
3. 切换到自己的分支
4. 在自己的分支上写代码并提交
5. 将自己分支的更新push到远程
6. 在Gitlab上发起请求，请求合并自己的分支到远程的master分支

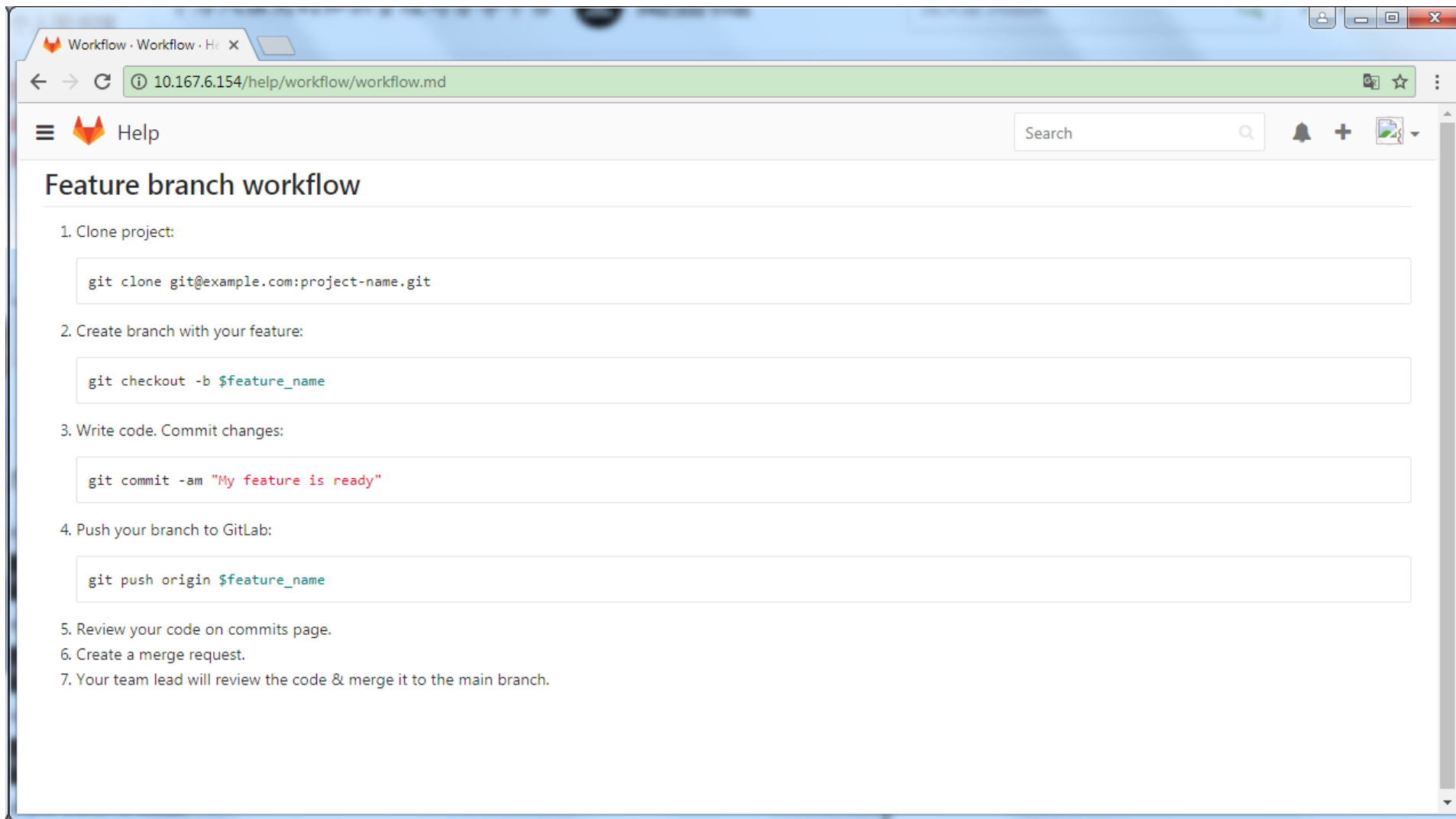
群组管理员（Master）：

1. 管理员收到合并请求，审查代码并决定是否通过合并请求

Developer

开发者Workflow

Gitlab建议的协同工作流



开发者Workflow

```
MINGW64:/f/Gitlab/F1322817/COMMON
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817
$ git clone git@10.167.6.154:open-source/COMMON.git
Cloning into 'COMMON'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (6/6), done.
Checking connectivity... done.

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817
$ cd COMMON/

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/COMMON (master)
$ git checkout f1322817Feature1
Branch f1322817Feature1 set up to track remote branch f1322817Feature1 from origin.
Switched to a new branch 'f1322817Feature1'

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/COMMON (f1322817Feature1)
$ git add README.md

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/COMMON (f1322817Feature1)
$ git commit -m 'update README.md'
[f1322817Feature1 eb15a7f] update README.md
1 file changed, 1 insertion(+), 1 deletion(-)

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/COMMON (f1322817Feature1)
$

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/COMMON (f1322817Feature1)
$ git push origin f1322817Feature1
Counting objects: 3, done.
Writing objects: 100% (3/3), 263 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: To create a merge request for f1322817Feature1, visit:
remote:   http://10.167.6.154/open-source/COMMON/merge_requests/new?merge_request%5Bsource_branch%5D=f1322817Feature1
remote:
remote: To git@10.167.6.154:open-source/COMMON.git
       8058d1f..eb15a7f  f1322817Feature1 -> f1322817Feature1

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/COMMON (f1322817Feature1)
$ |
```

① 克隆管理员创建好的项目到本地

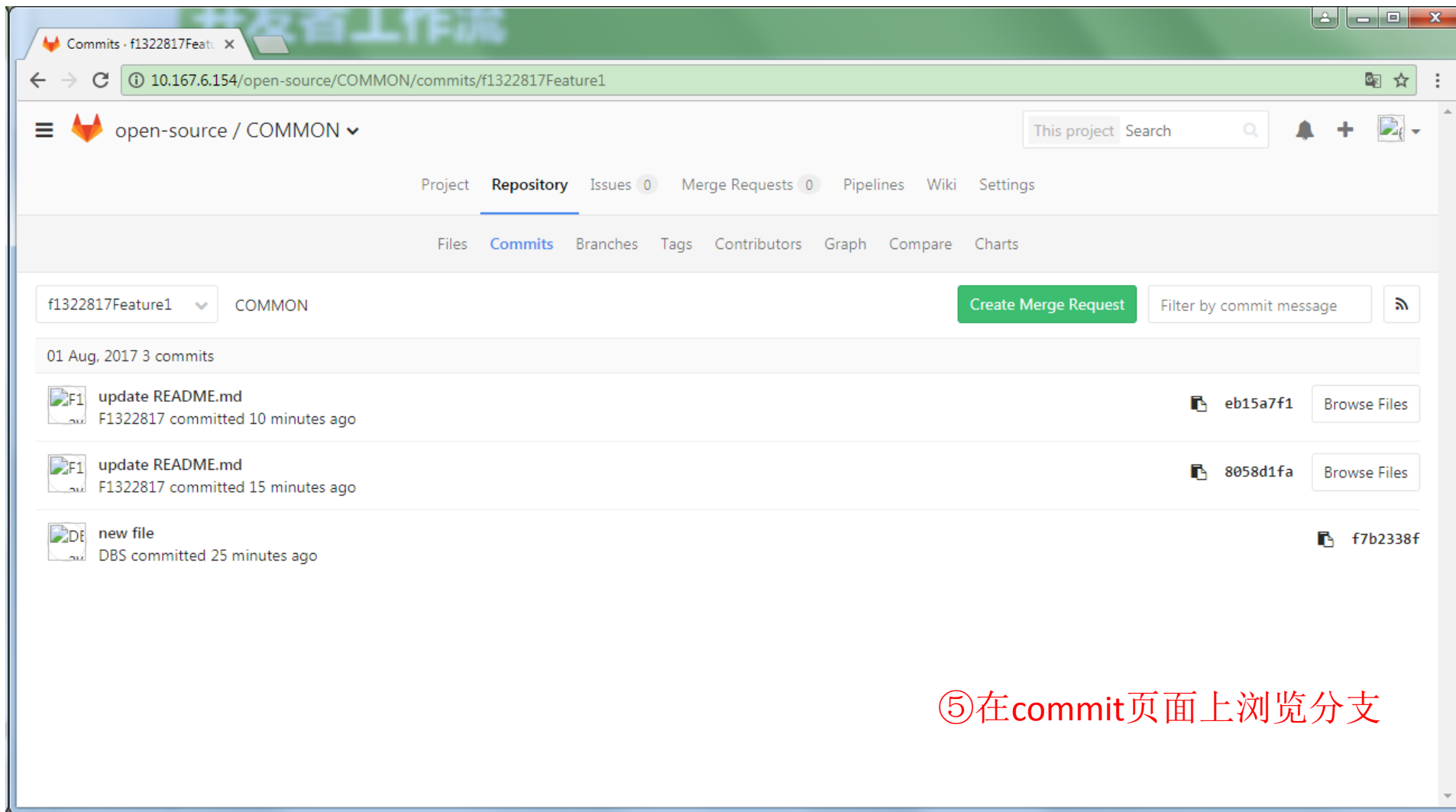
② 切换到自己的分支

③ 在自己的分支上写代码并提交

④ push到远程服务器(分支是自己的分支)

开发者Workflow

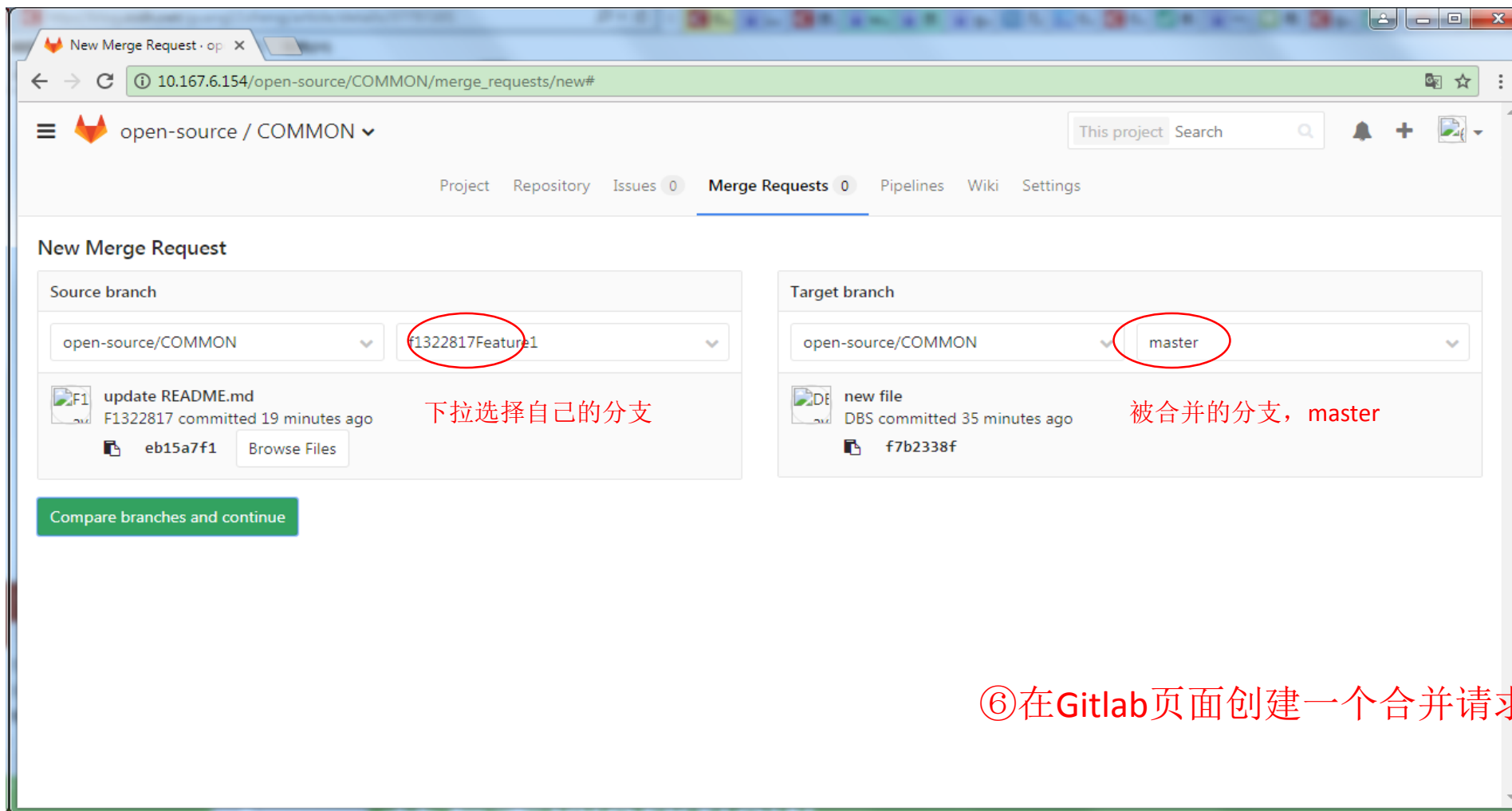
Projects -> 项目名称 -> Repository -> Commits -> 选择自己的分支，点击单个链接可以看到更新的具体内容。



⑤在commit页面上浏览分支

开发者Workflow

点击Merge Requests -> New Merge Request，在New Merge Request页面选择自己要合并的分支和要并入的分支（master），点击Compare branches and continue继续。



⑥在Gitlab页面创建一个合并请求

开发者Workflow

开发者操作总结：

1. 克隆管理员创建好的项目到自己本地
2. 切换到自己的分支（分支由管理员创建）
3. 在自己的分支上写代码，并提交
4. 推送到远程服务器，分支是自己的分支
5. 在commit页面上浏览分支
6. 创建一个合并请求

其中：远程仓库由管理员创建，开发者使用的分支也由管理员事先创建。开发者在Gitlab页面上创建了合并请求之后，由管理员审查代码并最终通过合并请求，完成分支到主分支的合并。

Master

Master – 创建群组

左侧菜单Project -> Groups -> New Group，填入群组名称和描述，选择权限选项点击Create group提交。



The screenshot shows the 'New Group' form in GitLab. The browser address bar indicates the URL is 10.167.6.154/groups/new. The form includes the following fields and options:

- Group path:** The text 'http://10.167.6.154/' is followed by a text input field containing 'open-source'. A red circle highlights the input field, and a red annotation '写入Group名称' (Write Group Name) points to it.
- Description:** A text area containing 'DBS Group'. A red circle highlights the text, and a red annotation '写入Group描述' (Write Group Description) points to it.
- Group avatar:** A button labeled 'Choose File ...' and a text input field for 'File name...'. Below this, it states 'The maximum file size allowed is 200KB.'
- Visibility Level:** Three radio button options are present:
 - Private:** Selected. Red annotation: '群组和群组下的Projects仅对群内部成员可见' (Group and projects under the group are only visible to group members).
 - Internal:** Red annotation: 'Gitlab登录用户均可见' (Visible to all GitLab logged-in users).
 - Public:** Red annotation: '所有人都可见，没有任何权限管控' (Visible to everyone, no permission control).

At the bottom of the form, there is a green 'Create group' button and a 'Cancel' button.

Master - 添加群组成员

左侧菜单Group -> 单击群组名称 -> Members进入该群组成员管理页面，按照指引添加成员

Members · open-source

10.167.6.154/groups/open-source/group_members

Group Issues 0 Merge Requests 0 Members

Members

Add new member to **open-source**

从下拉菜单选择要添加的用户

Search for members by name, username, or email, or invite new ones using their email address.

设定权限

过期时间，不写为永久有效

Read more about role permissions

On this date, the member(s) will automatically lose access to this group and all of its projects.

Existing members

Find existing members by name

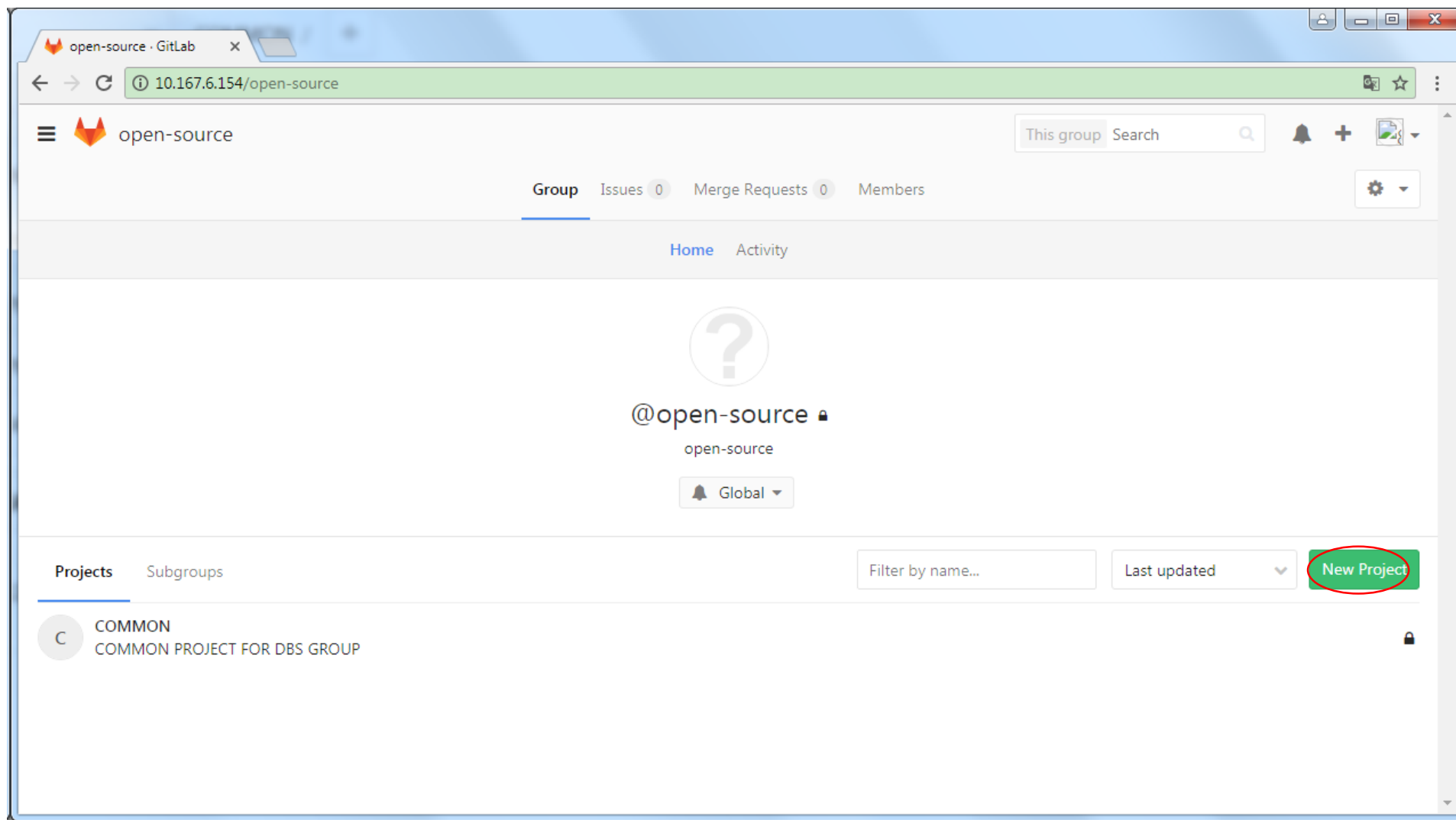
Name, ascending

Members with access to **open-source** 2

	DBS @DBS It's you Joined about 5 hours ago			Owner
	F1322817 @F1322817 Joined about 5 hours ago	<input type="text" value="Developer"/>	<input type="text" value="Expiration date"/>	<input type="button" value="Remove"/>

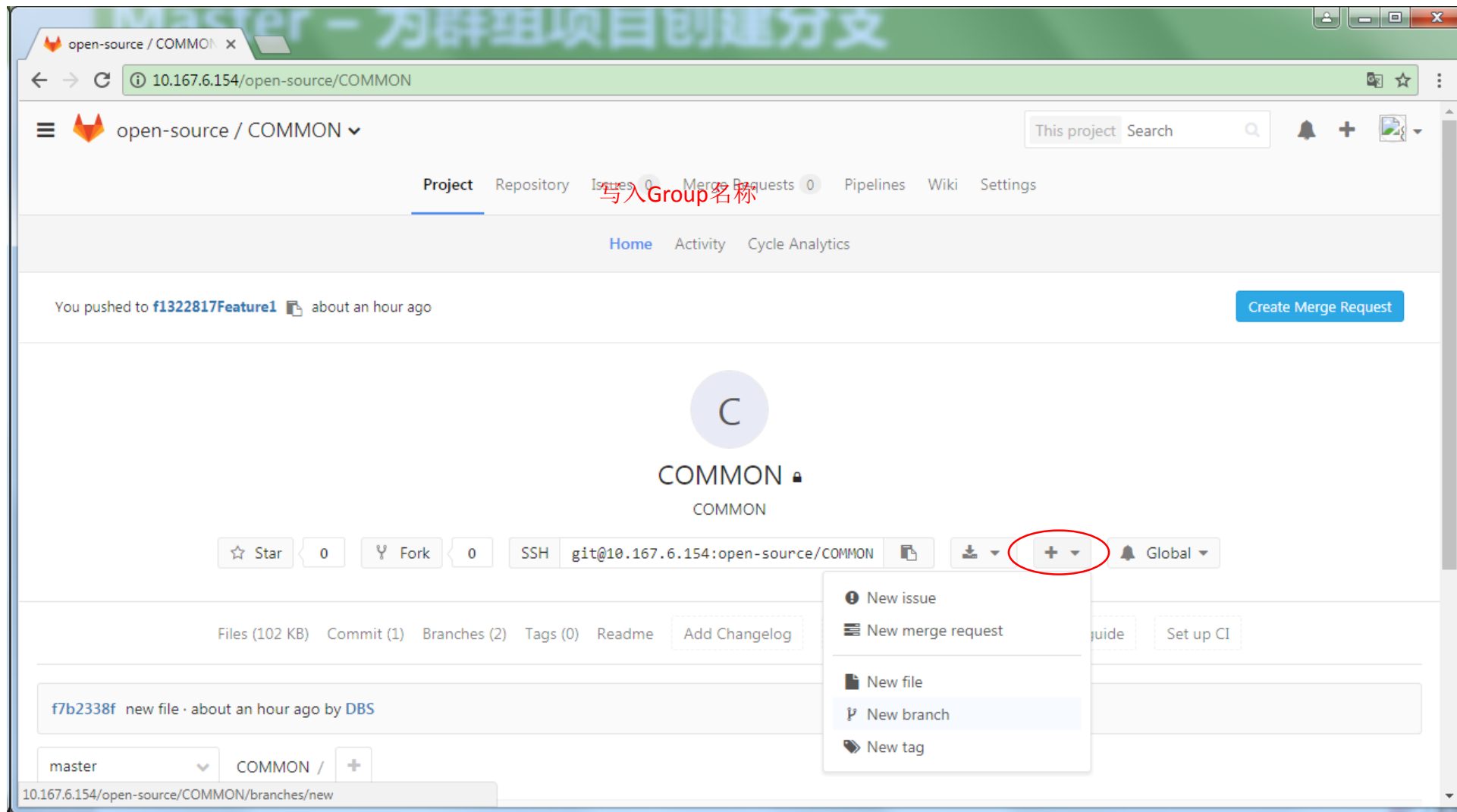
Master - 新建群组项目

单击左边菜单栏，选择Group -> 单击群组名称 -> New Project，按照指引新建项目。



Master – 为群组项目创建分支

进入Project页面，在SSH地址栏右边点+，下拉菜单中选择New branch，创建新的分支。



Master – 为群组项目创建分支

进入Project页面，在SSH地址栏右边点+，下拉菜单中选择New branch，创建新的分支。

New Branch · open-source X

10.167.6.154/open-source/COMMON/branches/new

open-source / COMMON

This project Search

Project Repository Issues 0 Merge Requests 0 Pipelines Wiki Settings

New Branch

Branch name f1322817Feature1

Create from master

Existing branch name, tag, or commit SHA

Create branch Cancel

分支的命名规则使用小驼峰式命名法，具体为开发人员工号+所开发的功能/要解决的BUG名，工号首字母小写。命名中不要使用特殊字符，不要使用点。

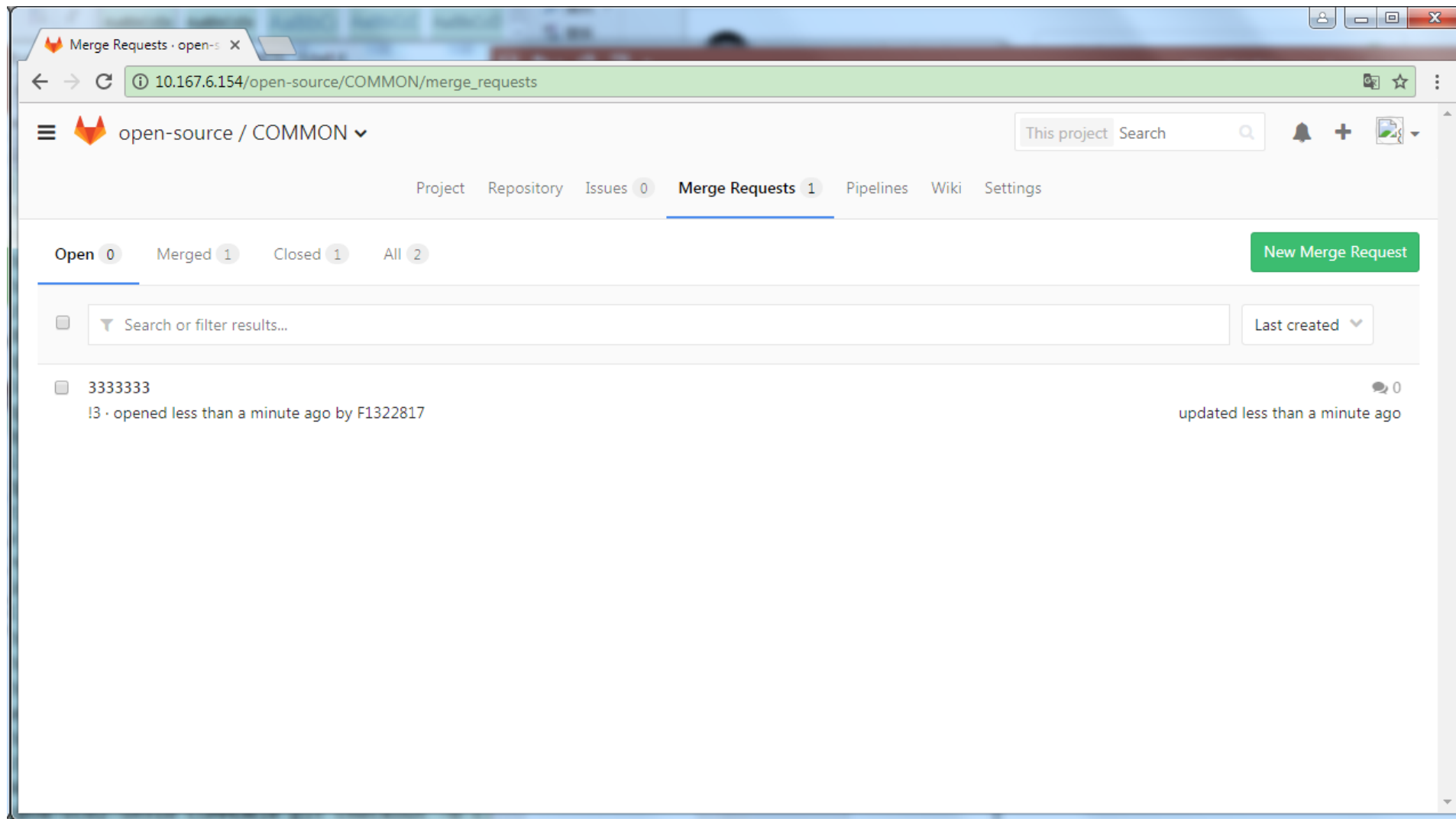
分支命名：

f1322817Feature1

f1322817Bug051

Master – 合并分支

管理员收到合并请求，审查代码并决定是否通过合并请求。点击Accept Merge Request通过，代码完成合并。

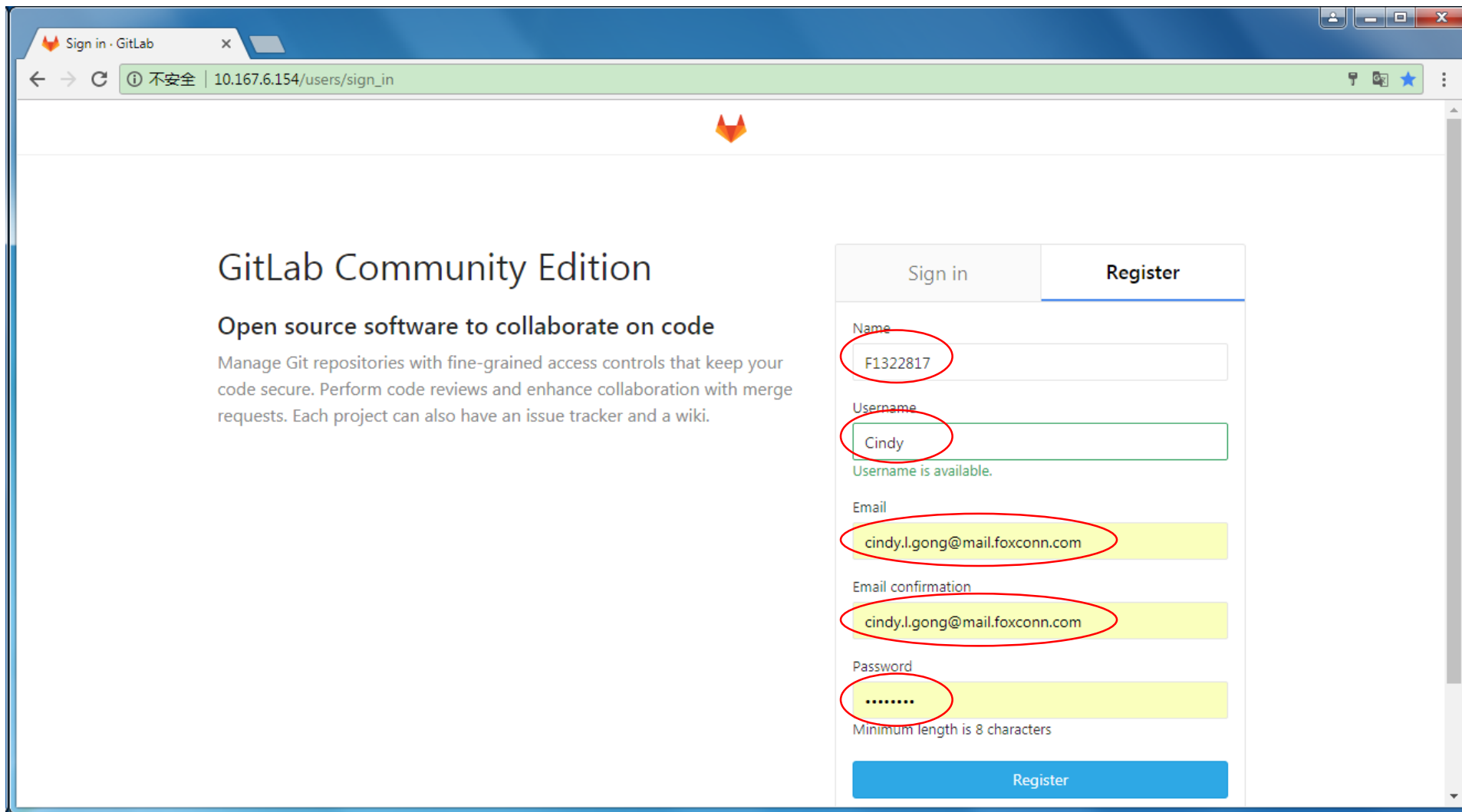


附录

Gitlab基本操作

1.用户注册

按要求填写完所有项目之后单击**Register**提交。



Sign in · GitLab

← → ↻ 不安全 | 10.167.6.154/users/sign_in

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.

Sign in

Register

Name

F1322817

Username

Cindy

Username is available.

Email

cindy.l.gong@mail.foxconn.com

Email confirmation

cindy.l.gong@mail.foxconn.com

Password

.....

Minimum length is 8 characters

Register

2.添加个人SSH KEY

在自己电脑上新开Git Bash窗口，用命令ssh-keygen.exe -t rsa生成SSH公钥。在隐形文件夹.ssh下生成的id_rsa.pub即为SSH公钥文件。

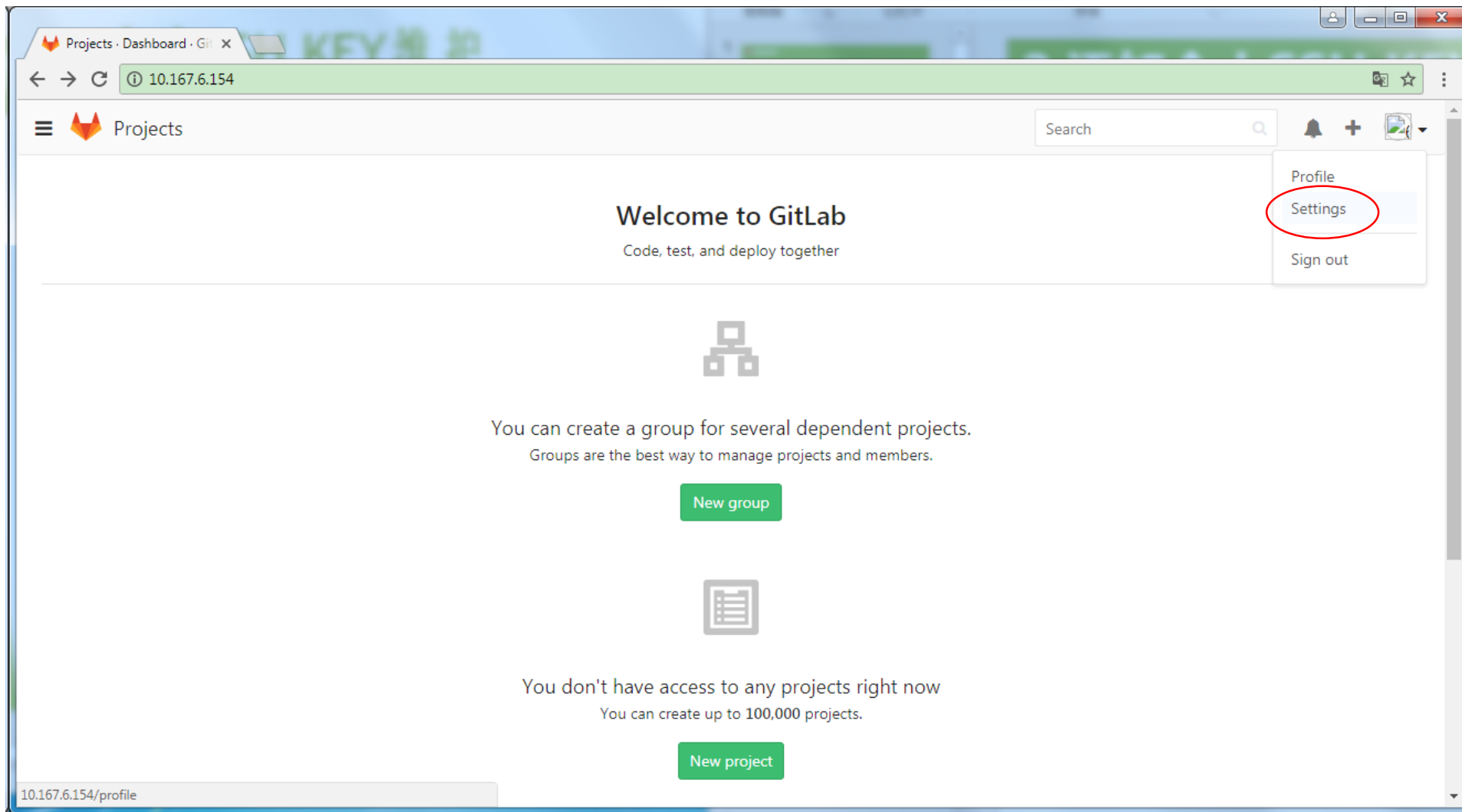
```
Cindy Gong@CINDY-PC MINGW64 ~
$ ssh-keygen.exe -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Cindy Gong/.ssh/id_rsa):
Created directory '/c/Users/Cindy Gong/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Cindy Gong/.ssh/id_rsa.
Your public key has been saved in /c/Users/Cindy Gong/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:4Td02JZxUqqPQzcz50Xm/qxr2azYcZKXgPVSLIu6o98 Cindy Gong@CINDY-PC
The key's randomart image is:
+---[RSA 2048]---+
|                 . .
|                o +
|               o O +
|              o % X
|             S ^ .
|            o X O .
|           . o B++
|          .o +oBo
|         .oo.E.=+o
+---[SHA256]-----+
```

```
Cindy Gong@CINDY-PC MINGW64 ~/.ssh
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDC3wD8nF3if1nT+XRfGNNn1H3dJcDAZw48uRbzTN07fZkJO1rZ/Y4XWn1O3
PBhXgbZ8V/WKFZGeET2xwqwhtrsoTn0fqZLuKiutvR2quKWT8t7Q0FipLNGZuxwK6XjhnyTyddzD8fUccDDRo/X/MJ4S/E8Z
jttqmm+5XzcB90vd/PvmeVviUQg2r3zRks0gwN2iiXNCw8nPIJx/9k13ASEPmSqLSZv0YvZf++I6kB054o2Rv1ccCKiGwLdomD
sbw22J57mZo5iM7fBu3AZ/NmyhJkr4L3AZUamrTYL1STTSjzD8dsHbwRXU6Hk1rRqKucKLG1vesX6PDP3Io3K0n1f Cindy G
ong@CINDY-PC
```

```
Cindy Gong@CINDY-PC MINGW64 ~/.ssh
```

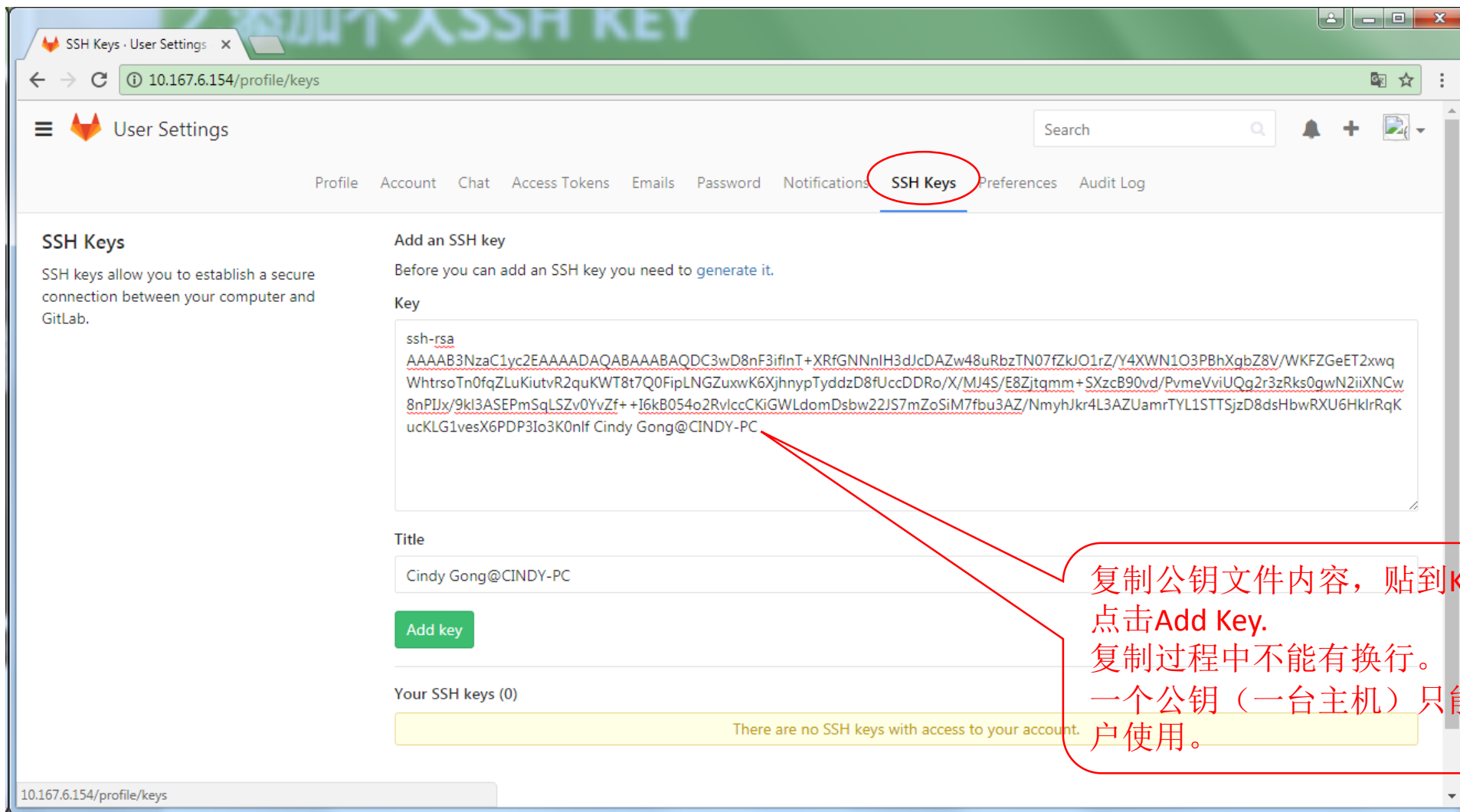
2.添加个人SSH KEY

单击用户头像，选择Settings。



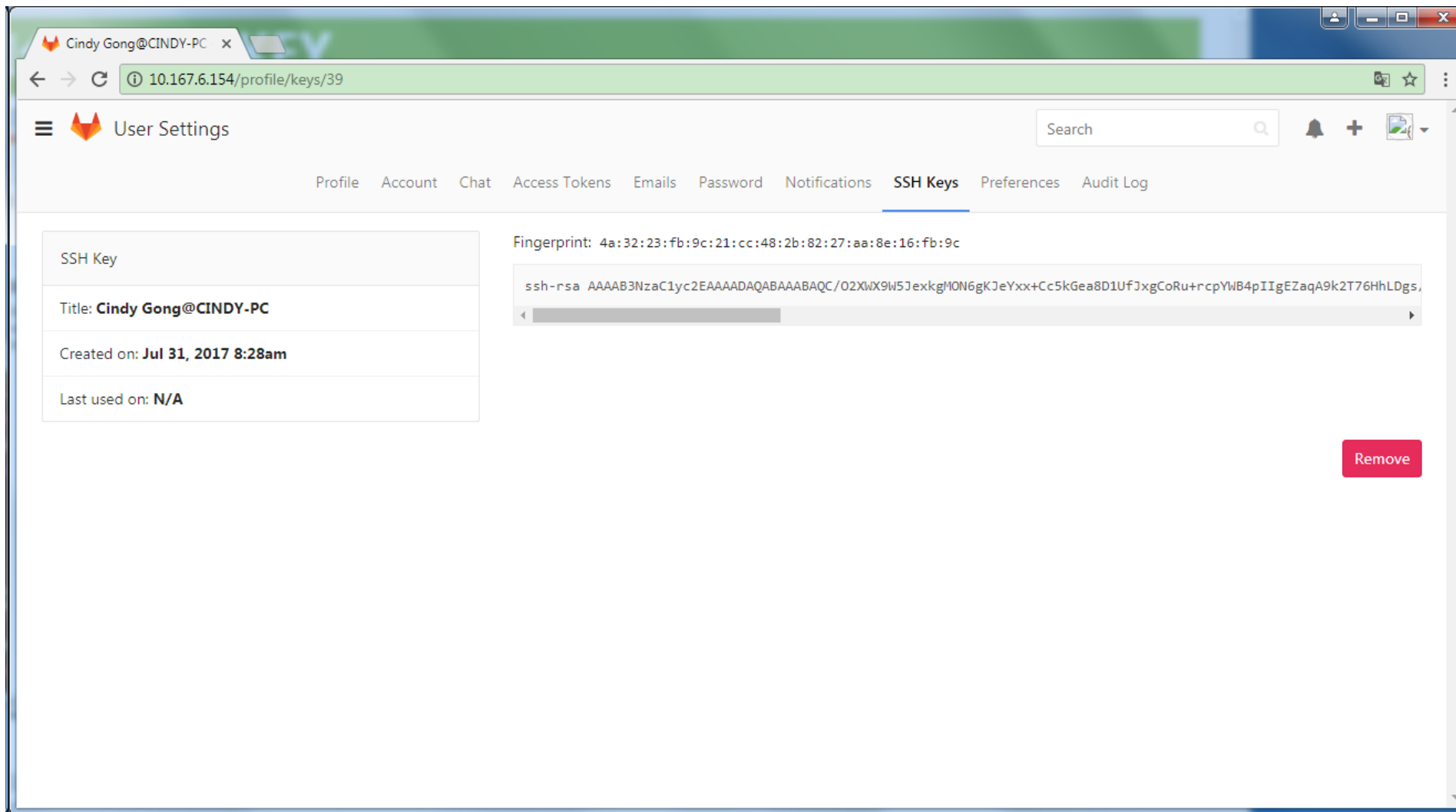
2.添加个人SSH KEY

单击SSH Keys切换到SSH Keys配置页面。复制刚才本地生成的公钥文件内容，贴到Key的框框中，点击Add Key添加。



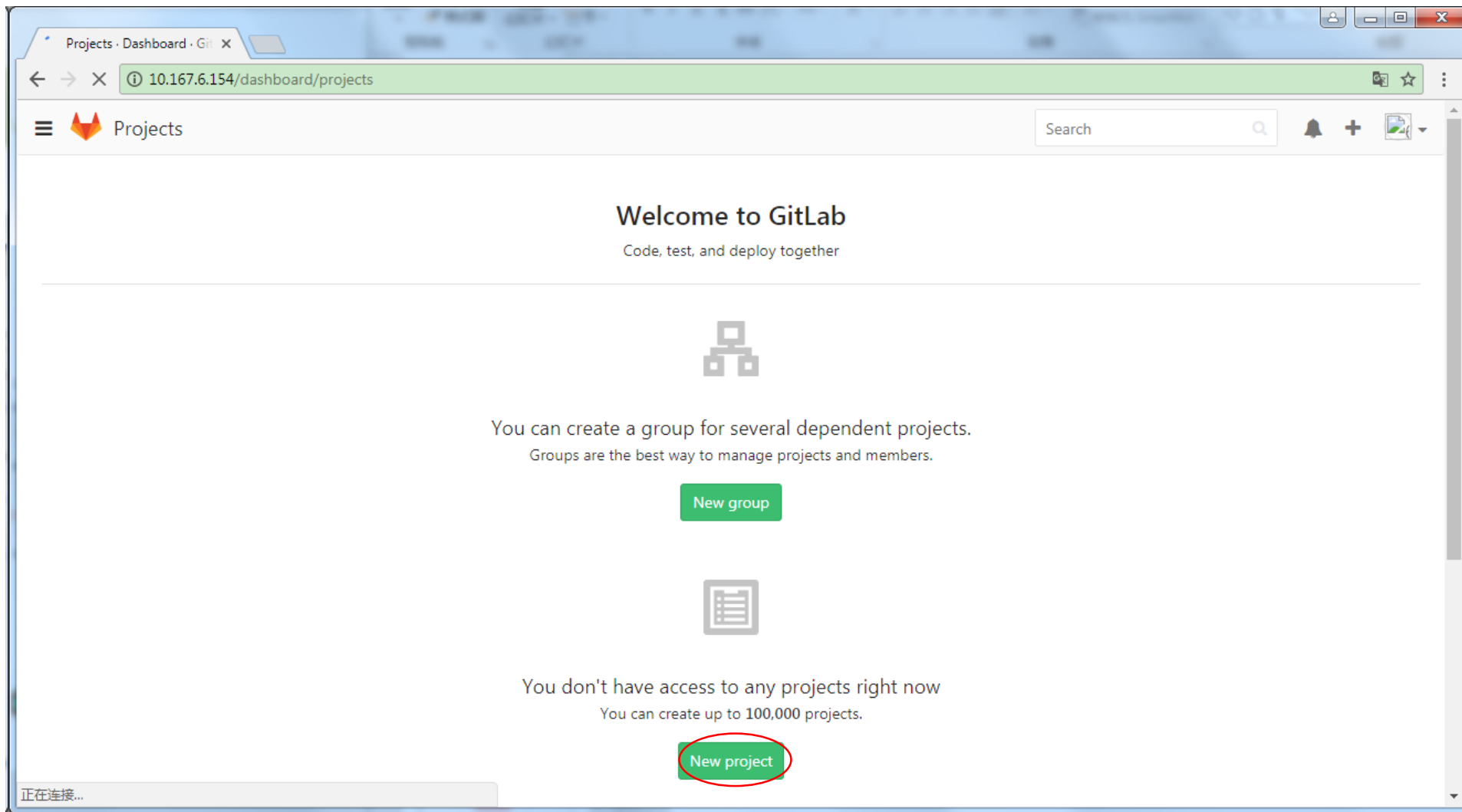
2.添加个人SSH KEY

至此，个人SSH KEY添加完成。



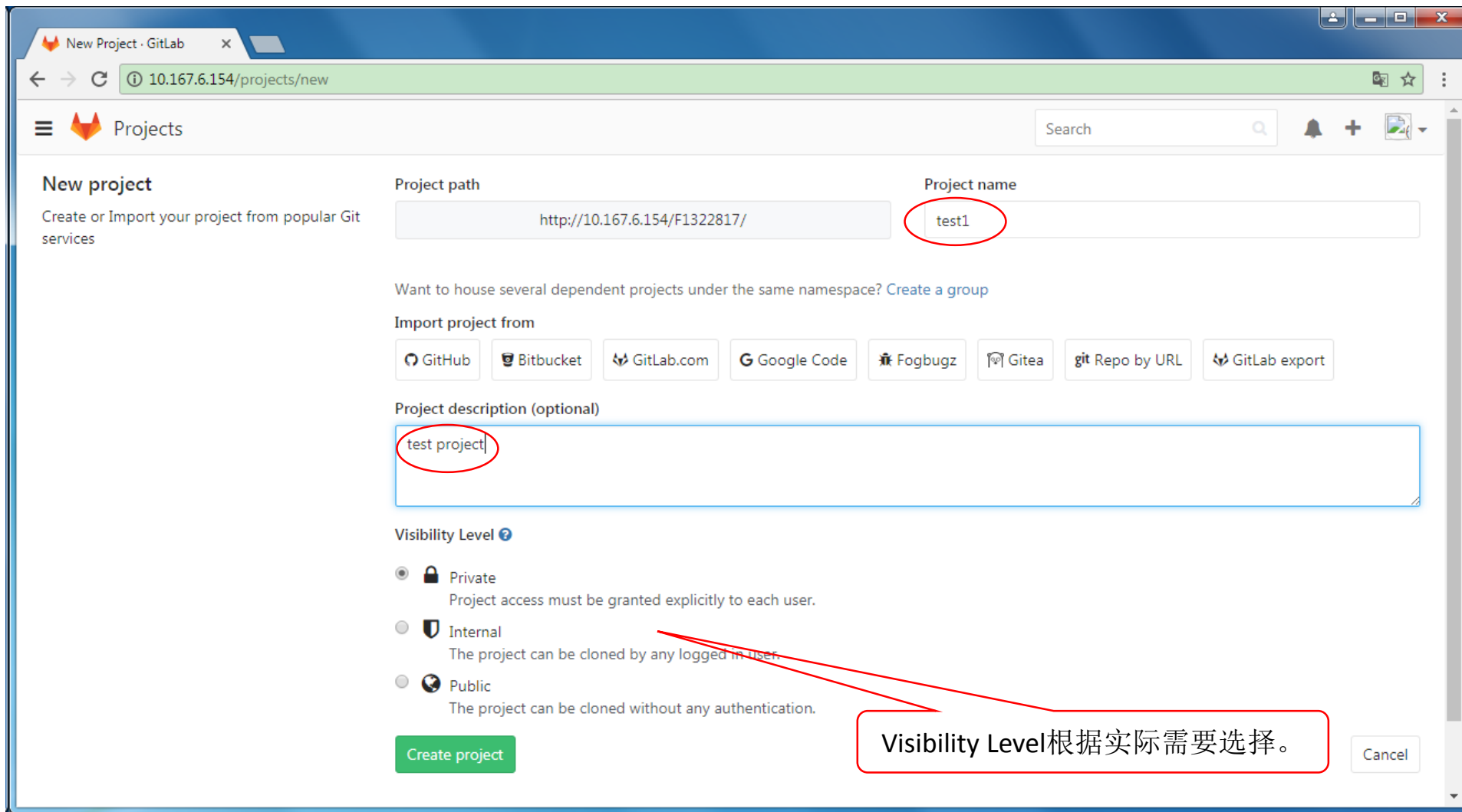
3.新建project

单击左边菜单栏，选择Projects -> New Project



3.新建project

写入项目名称，和项目描述，点击Create project。



The screenshot shows the 'New Project' form in GitLab. The browser address bar shows '10.167.6.154/projects/new'. The form includes fields for 'Project path' (containing 'http://10.167.6.154/F1322817/'), 'Project name' (containing 'test1'), and 'Project description (optional)' (containing 'test project'). Below these are buttons for 'Import project from' various services like GitHub, Bitbucket, GitLab.com, Google Code, Fogbugz, Gitea, git Repo by URL, and GitLab export. The 'Visibility Level' section has three radio buttons: 'Private' (selected), 'Internal', and 'Public'. A red box highlights the 'Project name' field, and another red box highlights the 'Project description' field. A red arrow points from the 'Visibility Level' section to a red box containing the text 'Visibility Level根据实际需要选择。'. The 'Create project' button is green, and the 'Cancel' button is grey.

New Project · GitLab

10.167.6.154/projects/new

Projects

New project
Create or Import your project from popular Git services

Project path: http://10.167.6.154/F1322817/

Project name: test1

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

Import project from

GitHub Bitbucket GitLab.com Google Code Fogbugz Gitea git Repo by URL GitLab export

Project description (optional): test project

Visibility Level

- ☒ Private
Project access must be granted explicitly to each user.
- ☐ Internal
The project can be cloned by any logged in user.
- ☐ Public
The project can be cloned without any authentication.

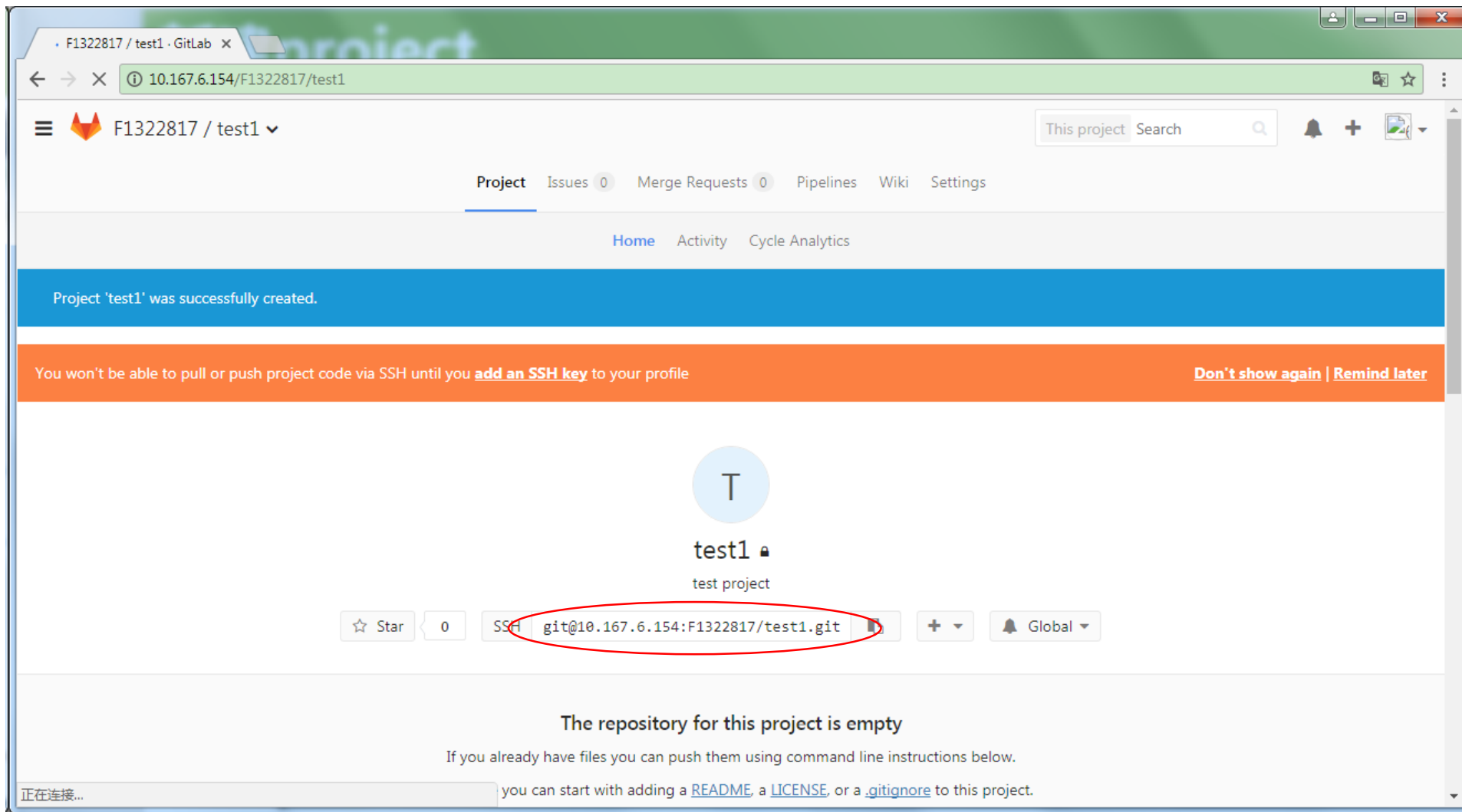
Create project

Cancel

Visibility Level根据实际需要选择。

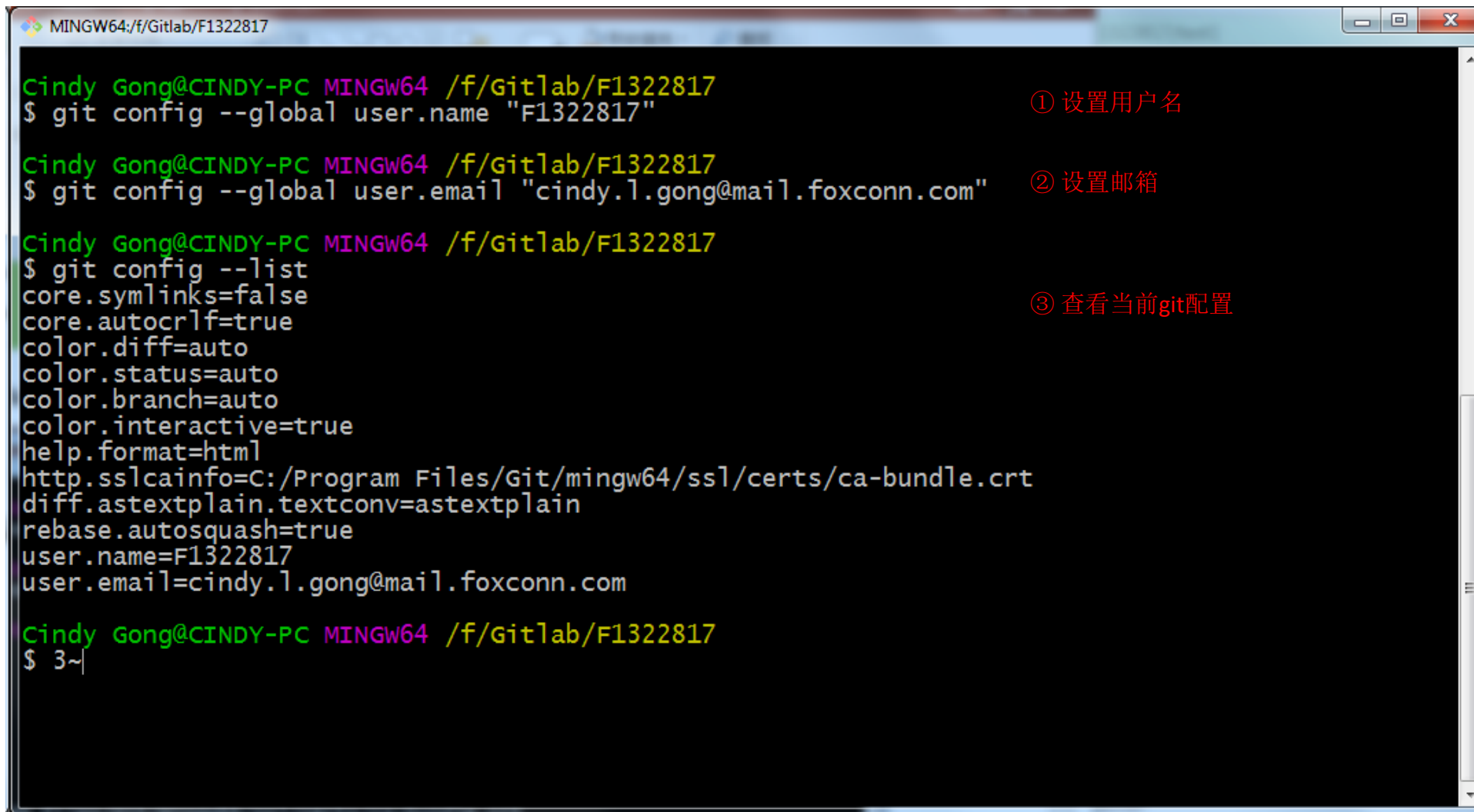
3.新建project

Project创建完成之后，会产生一个以project.git命名的仓库。剩下就是用git上传下载代码，管理自己的项目。



4.导入项目

在本地主机并设置下git的用户名和邮箱。

A screenshot of a Windows command prompt window titled "MINGW64:/f/Gitlab/F1322817". The window has a black background with green and white text. It shows three commands being executed to configure Git. To the right of the terminal window, there are three red annotations in Chinese: "① 设置用户名" (1. Set username), "② 设置邮箱" (2. Set email), and "③ 查看当前git配置" (3. View current git configuration).

```
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817
$ git config --global user.name "F1322817"

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817
$ git config --global user.email "cindy.l.gong@mail.foxconn.com"

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817
$ git config --list
core.symlinks=false
core.autocrlf=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
rebase.autosquash=true
user.name=F1322817
user.email=cindy.l.gong@mail.foxconn.com

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817
$ 3~|
```

4.导入项目

如果项目在本地已经存在，需要导入到gitlab，可以通过命令直接将项目导入上去。

```
MINGW64:/f/Gitlab/F1322817/mysite
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817
$ cd mysite/

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/mysite
$ git init
Initialized empty Git repository in F:/Gitlab/F1322817/mysite/.git/

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/mysite (master)
$ git remote add origin git@10.167.6.154:F1322817/test1.git

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/mysite (master)
$ git add .
warning: LF will be replaced by CRLF in README.md.

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/mysite (master)
$ git commit -m 'new project'
[master (root-commit) eb9e84b] new project

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/mysite (master)
$ git push -u origin master
Counting objects: 17, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (17/17), 3.62 KiB | 0 bytes/s, done.
Total 17 (delta 0), reused 0 (delta 0)
To git@10.167.6.154:F1322817/test1.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817/mysite (master)
$
```

① 进入项目文件夹

② 初始化git仓库

③ 添加远程仓库

④ 将项目文件存进暂存区

⑤ 提交

⑥ 将更新推送到远程git仓库

4.导入项目

如果本地没有项目仓库，可以从gitlab上把项目克隆下来。

```
MINGW64:/f/Gitlab/F1322817-demo/test1
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo
$ git clone git@10.167.6.154:F1322817/test1.git
Cloning into 'test1'...
remote: Counting objects: 17, done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 17 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (17/17), done.
Checking connectivity... done.

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo
$ cd test1

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ ls -la
total 13
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 ./
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 ../
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 .git/
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 apps/
-rwxr-xr-x 1 Cindy Gong 197121 259 Aug 4 11:46 manage.py*
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 mysite/
-rw-r--r-- 1 Cindy Gong 197121 0 Aug 4 11:46 PRD.docx
-rw-r--r-- 1 Cindy Gong 197121 923 Aug 4 11:46 README.md
-rw-r--r-- 1 Cindy Gong 197121 0 Aug 4 11:46 requirements.txt
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 utils/

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ |
```

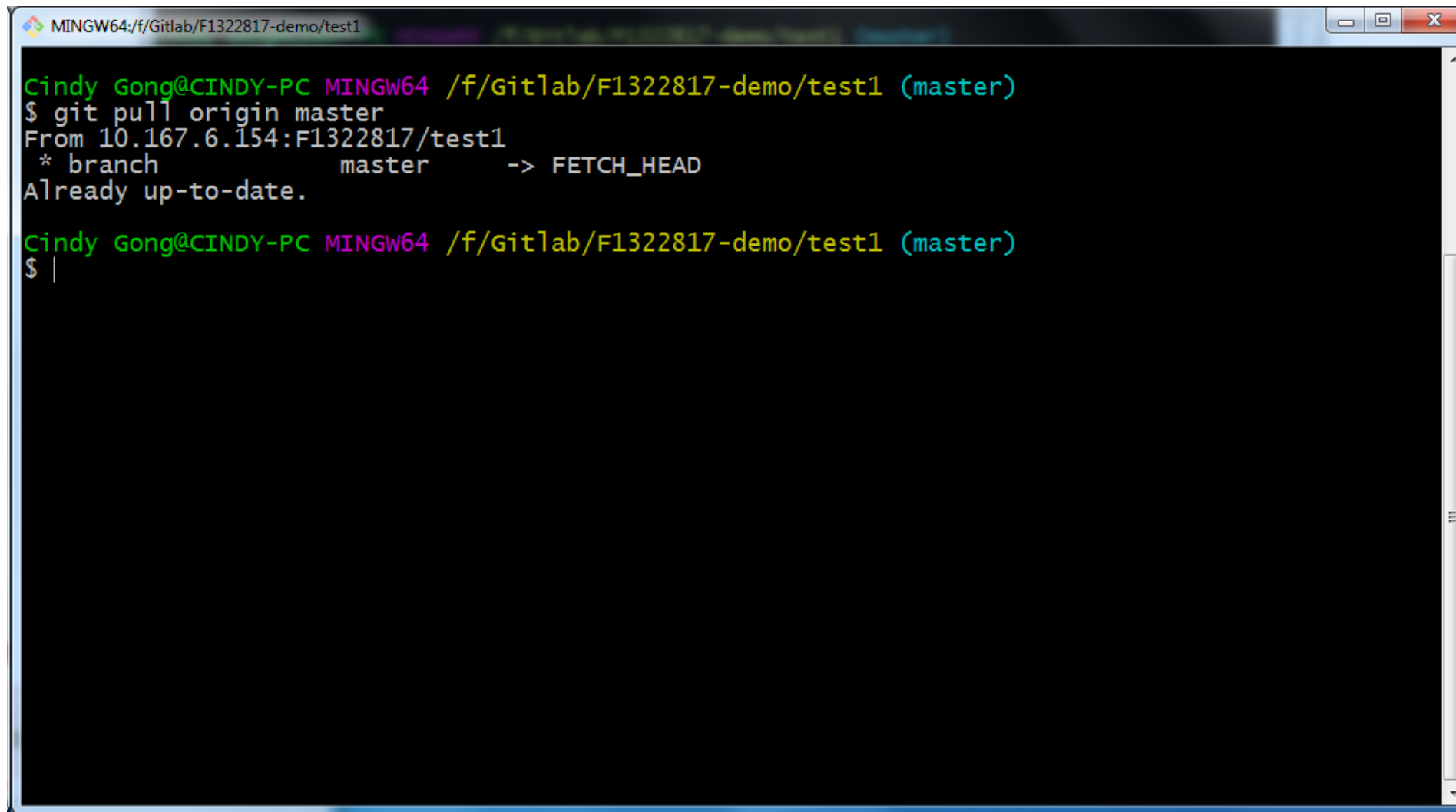
① 克隆远程仓库到本地电脑

② 进入项目文件夹

③ 查看项目文件目录

4.导入项目

如果已经拉取过远程仓库，每次在本地编辑代码之前，应该从远程获取最新版本并merge到本地。

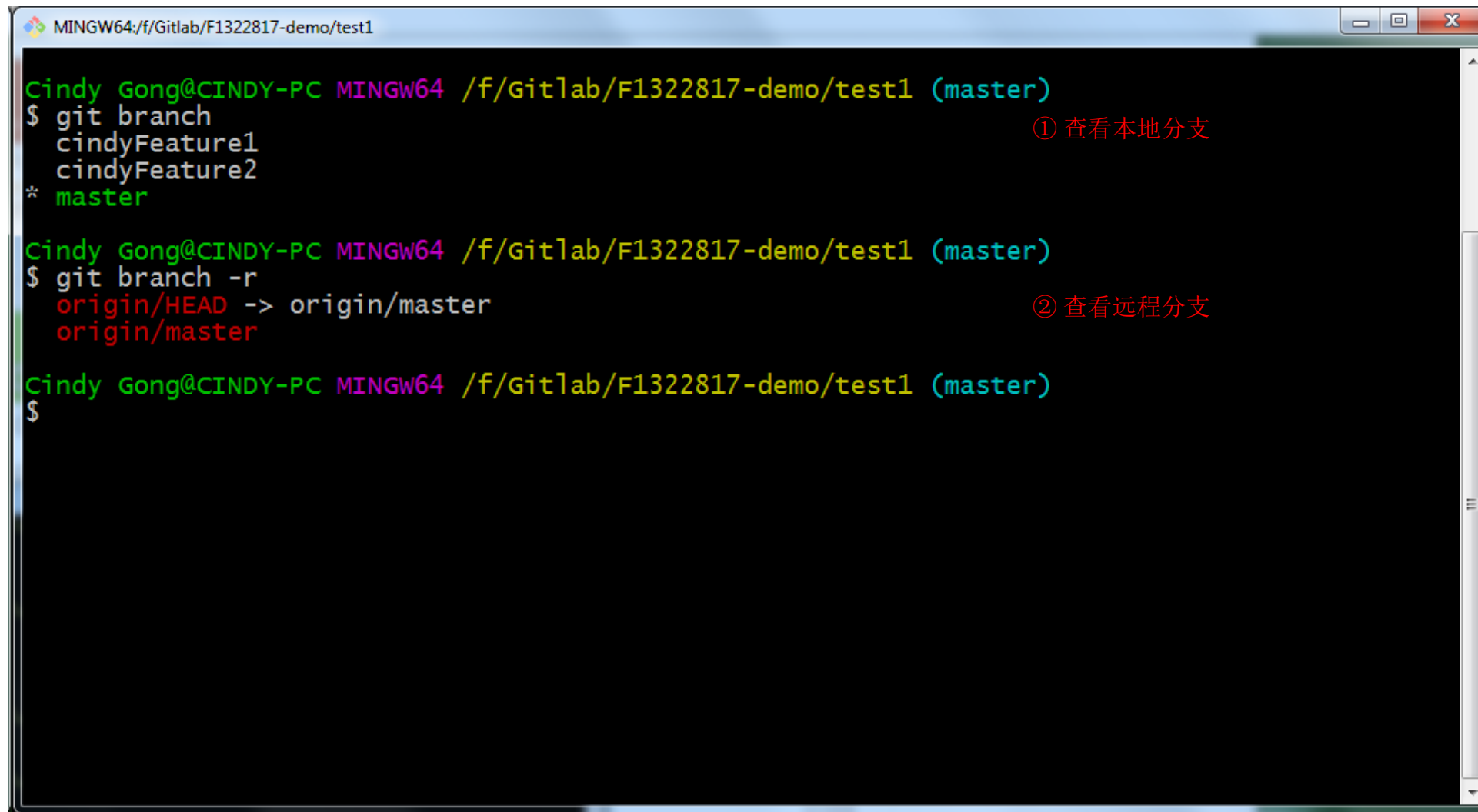
A screenshot of a Windows command prompt window titled "MINGW64:/f/Gitlab/F1322817-demo/test1". The prompt shows the user "Cindy Gong" at "CINDY-PC" in the "MINGW64" environment. The user enters the command "\$ git pull origin master". The output shows the source as "From 10.167.6.154:F1322817/test1", the branch mapping as "* branch master -> FETCH_HEAD", and the status "Already up-to-date." followed by a new prompt line "\$ |".

```
MINGW64:/f/Gitlab/F1322817-demo/test1
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git pull origin master
From 10.167.6.154:F1322817/test1
 * branch          master      -> FETCH_HEAD
Already up-to-date.

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ |
```


5.分支(branch)操作

查看分支: git branch



```
MINGW64:/f/Gitlab/F1322817-demo/test1

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch
cindyFeature1
cindyFeature2
* master

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch -r
origin/HEAD -> origin/master
origin/master

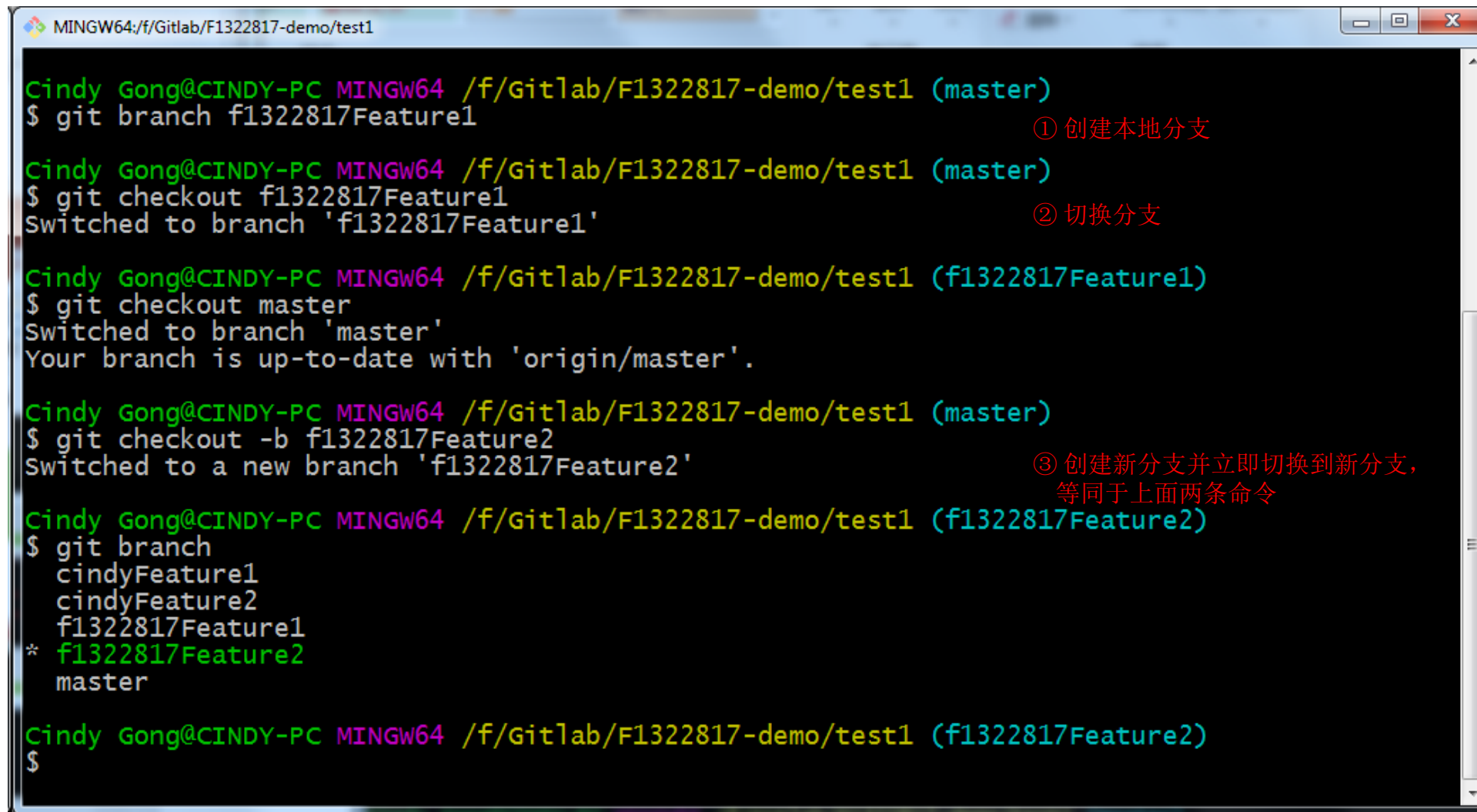
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$
```

① 查看本地分支

② 查看远程分支

5.分支(branch)操作

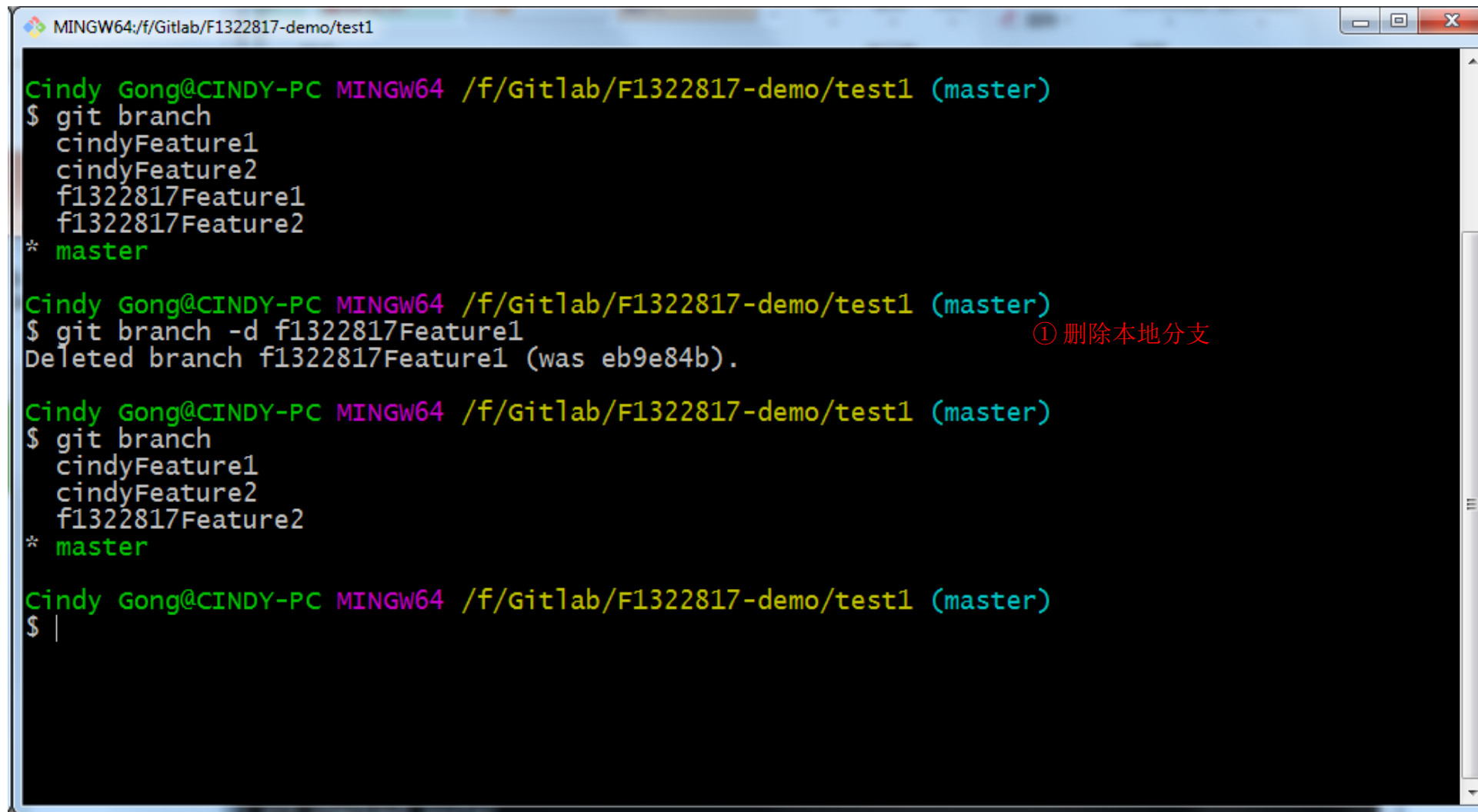
创建本地分支并切换到分支: `git branch [name]` , `git checkout [name]`

A terminal window titled 'MINGW64:/f/Gitlab/F1322817-demo/test1' showing a series of Git commands and their outputs. The commands are: 1. 'git branch f1322817Feature1' (labeled ① 创建本地分支), 2. 'git checkout f1322817Feature1' (labeled ② 切换分支), 3. 'git checkout master' (labeled ③ 创建新分支并立即切换到新分支, 等同于上面两条命令), 4. 'git checkout -b f1322817Feature2' (labeled ③ 创建新分支并立即切换到新分支, 等同于上面两条命令), and 5. 'git branch' (showing a list of branches with 'f1322817Feature2' as the current branch).

```
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch f1322817Feature1
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git checkout f1322817Feature1
Switched to branch 'f1322817Feature1'
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (f1322817Feature1)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git checkout -b f1322817Feature2
Switched to a new branch 'f1322817Feature2'
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (f1322817Feature2)
$ git branch
cindyFeature1
cindyFeature2
f1322817Feature1
* f1322817Feature2
master
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (f1322817Feature2)
$
```

5.分支(branch)操作

删除本地分支：git branch -d [name]。注意删除分支时该分支不能存在还未合并的代码。



```
MINGW64:/f/Gitlab/F1322817-demo/test1

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch
cindyFeature1
cindyFeature2
f1322817Feature1
f1322817Feature2
* master

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch -d f1322817Feature1 ① 删除本地分支
Deleted branch f1322817Feature1 (was eb9e84b).

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch
cindyFeature1
cindyFeature2
f1322817Feature2
* master

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ |
```

5.分支(branch)操作

创建远程分支：（创建本地分支，并push到远程）`git push origin [name]`



```
MINGW64:/f/Gitlab/F1322817-demo/test1

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch -r
origin/HEAD -> origin/master
origin/master

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git push origin f1322817Feature2
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: To create a merge request for f1322817Feature2, visit:
remote:   http://10.167.6.154/F1322817/test1/merge_requests/new?merge_request%5Bsource_branch%5D=f1322817Feature2
remote:
To git@10.167.6.154:F1322817/test1.git
* [new branch]      f1322817Feature2 -> f1322817Feature2

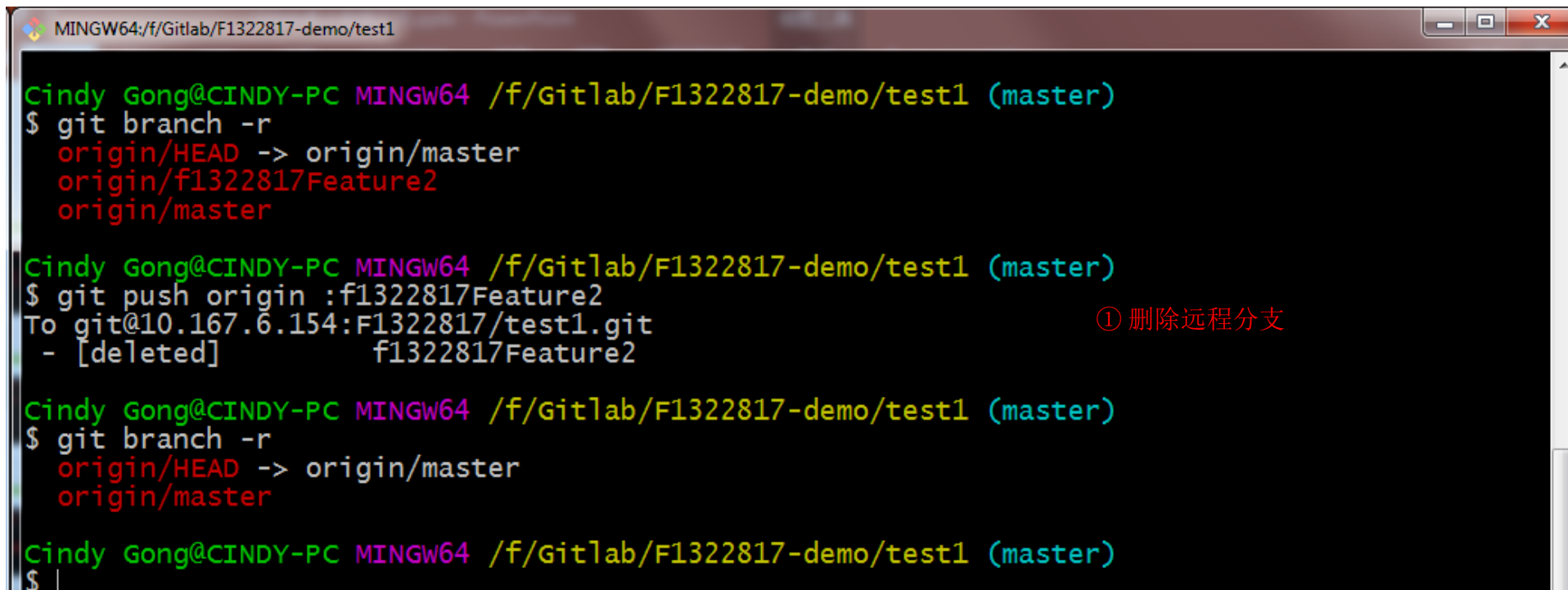
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch -r
origin/HEAD -> origin/master
origin/f1322817Feature2
origin/master

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ |
```

①将本地分支push到远程

5.分支(branch)操作

删除远程分支：git push origin :[name]



```
MINGW64:/f/Gitlab/F1322817-demo/test1

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch -r
origin/HEAD -> origin/master
origin/f1322817Feature2
origin/master

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git push origin :f1322817Feature2
To git@10.167.6.154:F1322817/test1.git
- [deleted]          f1322817Feature2

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git branch -r
origin/HEAD -> origin/master
origin/master

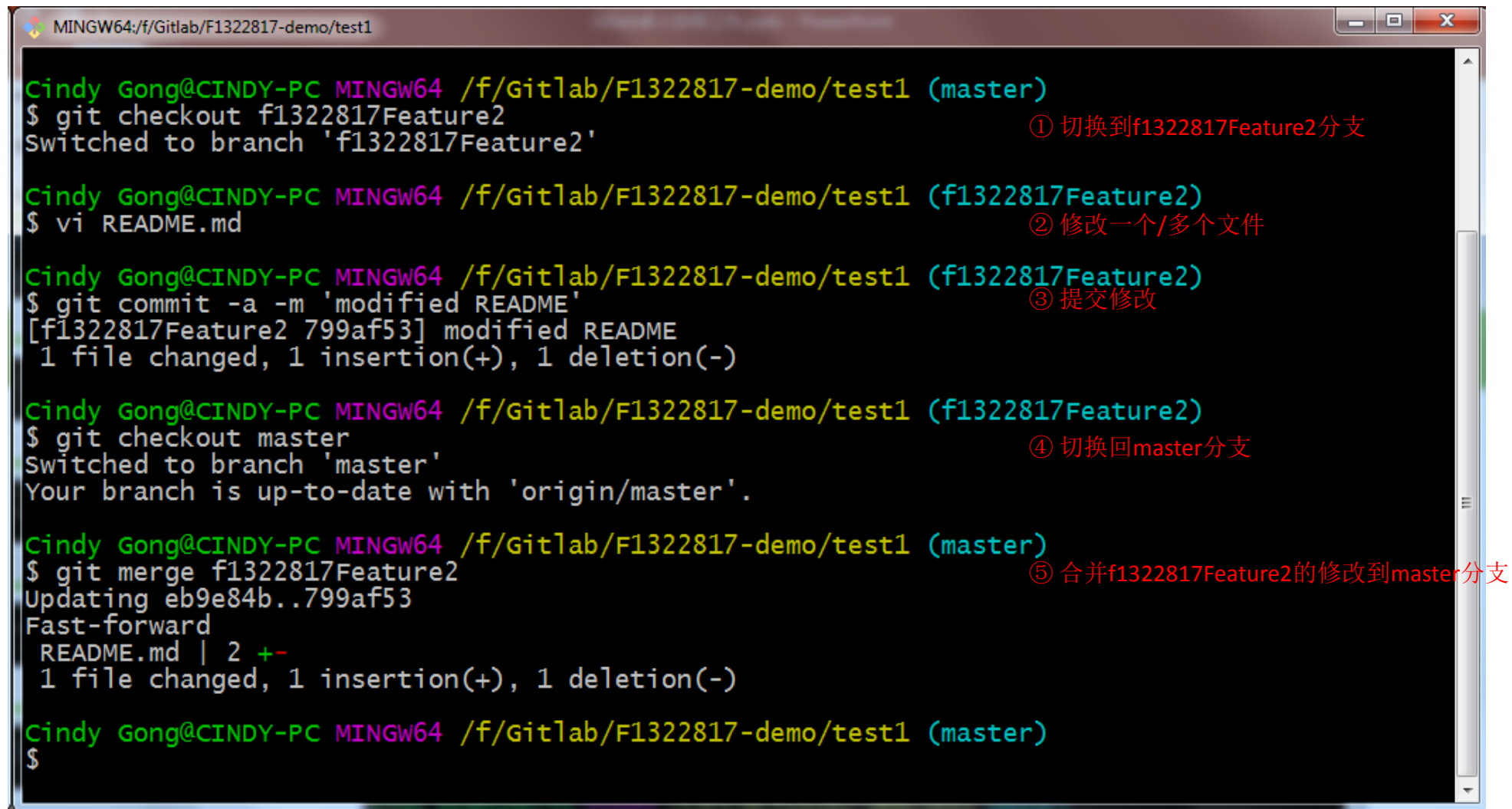
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$
```

① 删除远程分支

把本地某个分支提交到远程仓库，并作为远程仓库的分支，完整命令是：git push origin [name]:[name]。
删除远程分支的命令其实就是让左边的分支为空，其作用是删除远程的分支。但是该分支在本地还是会保存的。

5.分支(branch)操作

合并分支: `git merge [name]`



```
MINGW64:/f/Gitlab/F1322817-demo/test1

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git checkout f1322817Feature2
Switched to branch 'f1322817Feature2'

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (f1322817Feature2)
$ vi README.md

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (f1322817Feature2)
$ git commit -a -m 'modified README'
[f1322817Feature2 799af53] modified README
1 file changed, 1 insertion(+), 1 deletion(-)

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (f1322817Feature2)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git merge f1322817Feature2
Updating eb9e84b..799af53
Fast-forward
 README.md | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$
```

① 切换到f1322817Feature2分支

② 修改一个/多个文件

③ 提交修改

④ 切换回master分支

⑤ 合并f1322817Feature2的修改到master分支

6.子模块(submodule)操作

添加子模块: `git submodule add [url][path]`

```
MINGW64:/f/Gitlab/F1322817-demo/test1
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git submodule add git@10.167.6.154:open-source/COMMON.git
Cloning into 'COMMON'...
remote: Counting objects: 24, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 24 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (24/24), done.
Checking connectivity... done.
warning: LF will be replaced by CRLF in .gitmodules.
The file will have its original line endings in your working directory.

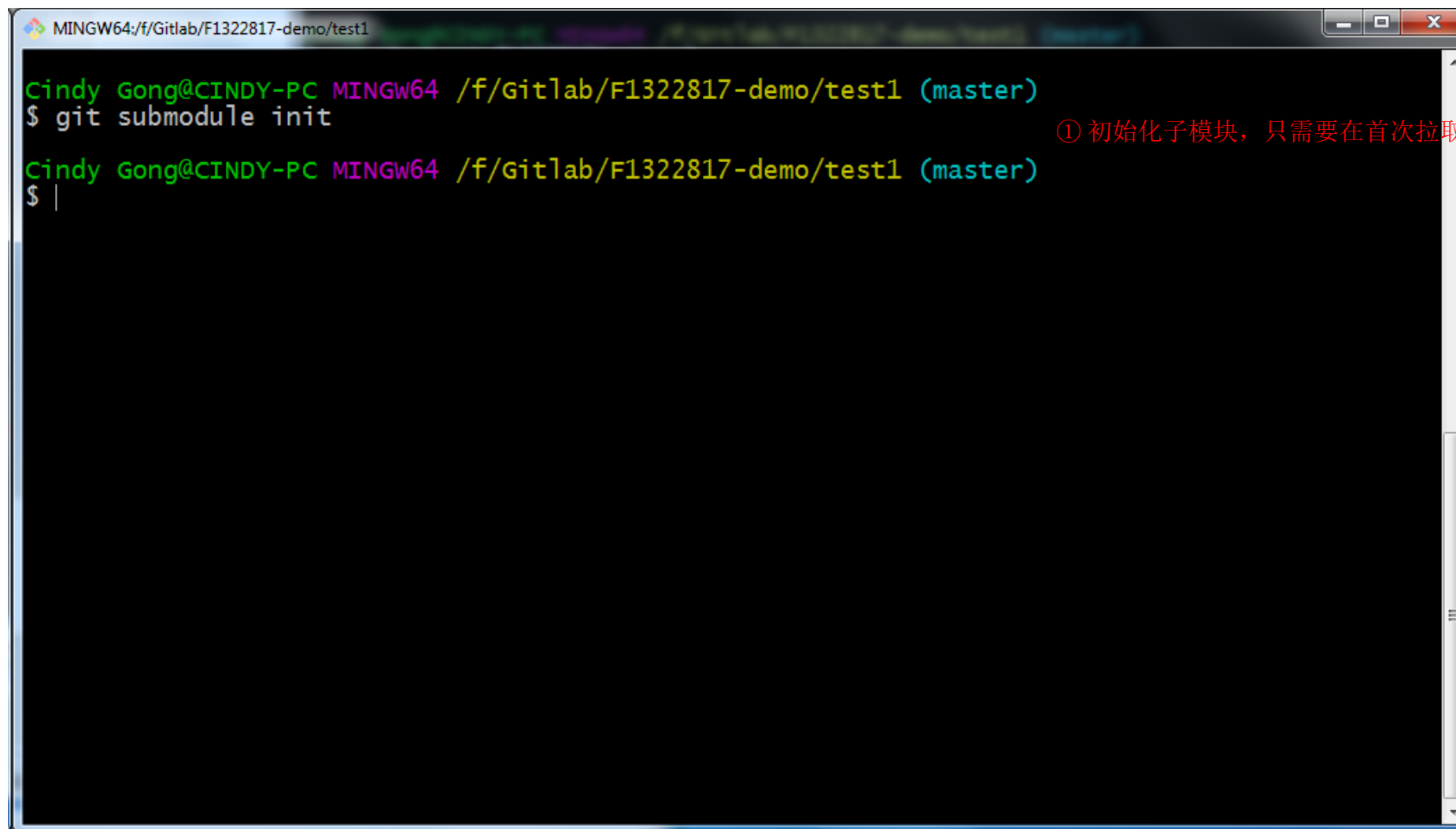
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ ls -la
total 14
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 14:57 ./
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 ../
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 14:57 .git/
-rw-r--r-- 1 Cindy Gong 197121 83 Aug 4 14:57 .gitmodules
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 apps/
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 14:57 COMMON/
-rwxr-xr-x 1 Cindy Gong 197121 259 Aug 4 11:46 manage.py*
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 mysite/
-rw-r--r-- 1 Cindy Gong 197121 0 Aug 4 11:46 PRD.docx
-rw-r--r-- 1 Cindy Gong 197121 926 Aug 4 14:19 README.md
-rw-r--r-- 1 Cindy Gong 197121 0 Aug 4 11:46 requirements.txt
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 utils/

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$
```

① 添加子模块

6.子模块(submodule)操作

初始化子模块: `git submodule init`

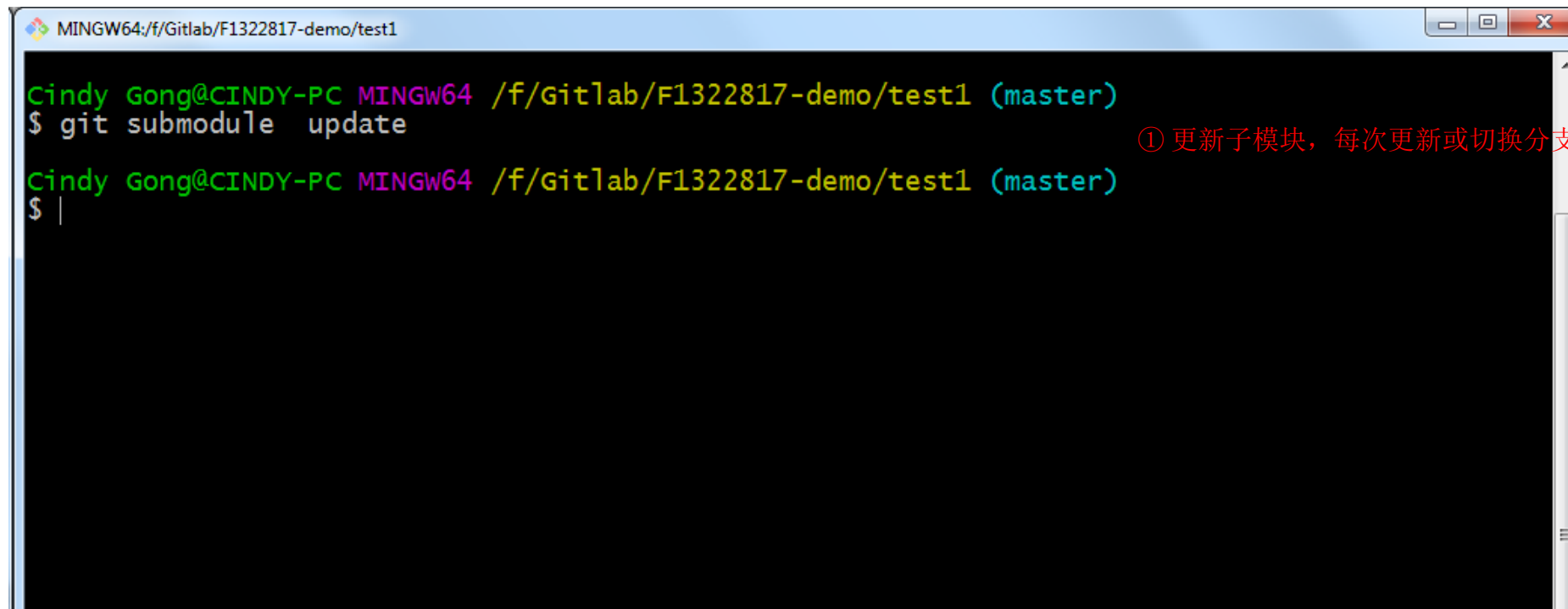
A screenshot of a Windows command prompt window titled 'MINGW64:/f/Gitlab/F1322817-demo/test1'. The prompt shows the user 'Cindy Gong' at 'CINDY-PC' in a 'MINGW64' environment. The command 'git submodule init' has been entered and executed. The prompt is now at the start of a new line, ready for the next command.

```
MINGW64:/f/Gitlab/F1322817-demo/test1
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git submodule init
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ |
```

① 初始化子模块，只需要在首次拉取的时候运行一次。

6.子模块(submodule)操作

更新子模块: `git submodule update`

A screenshot of a Windows command prompt window titled 'MINGW64:/f/Gitlab/F1322817-demo/test1'. The prompt shows the user 'Cindy Gong' at 'CINDY-PC' in the 'MINGW64' environment, located at '/f/Gitlab/F1322817-demo/test1' on the 'master' branch. The command '\$ git submodule update' has been entered. The prompt is now '\$ |' on the next line.

```
MINGW64:/f/Gitlab/F1322817-demo/test1
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git submodule update
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ |
```

① 更新子模块，每次更新或切换分支后都需要运行一下。

submodule项目和它的父项目本质上是2个独立的git仓库。只是父项目储存了它依赖的submodule项目的版本号而已。如果同组的同事更新了submodule，然后更新了父项目中依赖的版本号，你需要在git pull之后，调用git submodule update来更新submodule的信息。

6.子模块(submodule)操作

删除子模块，有4个步骤

```
MINGW64:/f/Gitlab/F1322817-demo/test1
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ git rm --cached COMMON/
rm 'COMMON'

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ vi .gitmodules

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ vi .git/config

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ rm -rf COMMON/

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ ls -la
total 13
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 16:06 ./
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 ../
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 16:06 .git/
-rw-r--r-- 1 Cindy Gong 197121 0 Aug 4 16:06 .gitmodules
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 apps/
-rwxr-xr-x 1 Cindy Gong 197121 259 Aug 4 11:46 manage.py*
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 mysite/
-rw-r--r-- 1 Cindy Gong 197121 0 Aug 4 11:46 PRD.docx
-rw-r--r-- 1 Cindy Gong 197121 926 Aug 4 14:19 README.md
-rw-r--r-- 1 Cindy Gong 197121 0 Aug 4 11:46 requirements.txt
drwxr-xr-x 1 Cindy Gong 197121 0 Aug 4 11:46 utils/

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$
```

① git rm -cached 子模块

② 编辑.gitmodules，将子模块相关配置节点删除

③ 编辑.git/config，将子模块相关配置节点删除

④ 手动删除子模块残留的目录

6.子模块(submodule)操作 - 子模块实例

场景：本地已有ST项目的仓库，从远程拉取COMMON作为ST项目的子模块，更新子模块并push到远程。

```
MINGW64:/f/Gitlab/ST/COMMON
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST (master)
$ pwd
/f/Gitlab/ST

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST (master)
$ git checkout f1322817Feature1
Branch f1322817Feature1 set up to track remote branch f1322817Feature1 from origin.
Switched to a new branch 'f1322817Feature1'

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST (f1322817Feature1)
$ git submodule add git@10.167.6.154:open-source/COMMON.git
Cloning into 'COMMON'...
remote: Counting objects: 24, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 24 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (24/24), done.
Checking connectivity... done.
warning: LF will be replaced by CRLF in .gitmodules.
The file will have its original line endings in your working directory.

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST (f1322817Feature1)
$ git status
On branch f1322817Feature1
Your branch is up-to-date with 'origin/f1322817Feature1'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   .gitmodules
    new file:   COMMON
```

① 在ST项目文件夹下切换到自己的分支

② 添加子模块

③ 查看git状态，此时ST项目文件夹下多了两个文件/文件夹

6.子模块(submodule)操作 - 子模块实例

场景：本地已有ST项目的仓库，从远程拉取COMMON作为ST项目的子模块，更新子模块并push到远程仓库。

```
MINGW64:/f/Gitlab/ST/COMMON
Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST (f1322817Feature1)
$ cd COMMON/

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST/COMMON (master)
$ git checkout f1322817Feature1
Branch f1322817Feature1 set up to track remote branch f1322817Feature1 from origin.
Switched to a new branch 'f1322817Feature1'

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST/COMMON (f1322817Feature1)
$ vi README.md

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST/COMMON (f1322817Feature1)
$ git status
On branch f1322817Feature1
Your branch is up-to-date with 'origin/f1322817Feature1'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST/COMMON (f1322817Feature1)
$ git commit -a -m 'test submodule'
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
[f1322817Feature1 warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
```

④ 进入子模块文件夹

⑤ 切换到子模块下自己的分支

⑥ 修改子模块的文件

⑦ 查看修改后的状态

⑧ 提交修改

6.子模块(submodule)操作 - 子模块实例

场景：本地已有ST项目的仓库，从远程拉取COMMON作为ST项目的子模块，更新子模块并push到远程仓库。

```
MINGW64:/f/Gitlab/ST/COMMON
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
1 file changed, 1 insertion(+), 1 deletion(-)

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST/COMMON (f1322817Feature1)
$ git push origin f1322817Feature1
Counting objects: 3, done.
Writing objects: 100% (3/3), 259 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: To create a merge request for f1322817Feature1, visit:
remote:   http://10.167.6.154/open-source/COMMON/merge_requests/new?merge_request%5Bsource_branch%5D=f1322817Feature1
remote:
To git@10.167.6.154:open-source/COMMON.git
    eb15a7f..1b1f02d  f1322817Feature1 -> f1322817Feature1

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/ST/COMMON (f1322817Feature1)
$ |
```

⑨ push修改到远程自己的分支

之后的操作就是去Gitlab网页上发起合并分支申请，等待管理员合并子模块的代码去master分支了。

特别注意：

在主项目(ST)和子模块（COMMON）之间切换目录的时候，切换目录的同时特别要注意切换自己的分支，否则所有的操作都跟自己的用户无关，最后的修改无效无法提交。

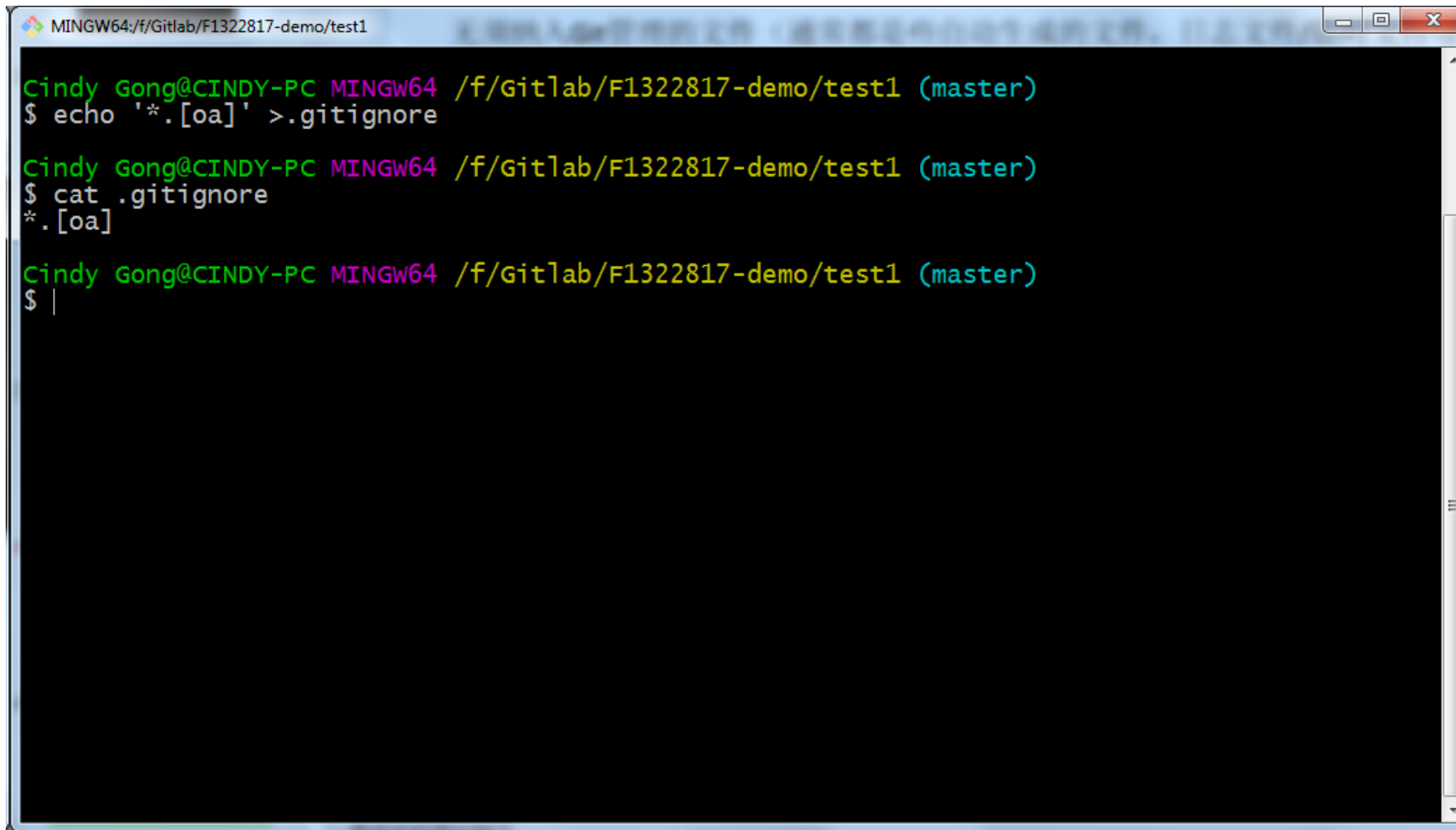
6.子模块(submodule)操作

使用submodule注意事项:

1. 对submodule做修改操作前，确保切换到正确的submodule项目分支。因为进入submodule文件夹，分支状态还保留在父项目的分支下。对submodule做修改前，要立即切换到submodule下你的本地分支。
2. `git pull` 之后，立即执行 `git status`，如果发现submodule有修改，立即执行 `git submodule update`。
3. 尽量不要使用 `git commit -a`，`git add`存在的意义就是让你对加入缓存区的文件做第二次确认，而`git commit -a`相当于跳过了这个确认过程。

7.忽略的文件/文件夹

无须纳入Git管理的文件（通常都是些自动生成的文件，日志文件/临时文件等），可以通过创建一个名为.gitignore的文件来管理。



```
MINGW64:/f/Gitlab/F1322817-demo/test1

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ echo '*. [oa]' >.gitignore

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ cat .gitignore
*. [oa]

Cindy Gong@CINDY-PC MINGW64 /f/Gitlab/F1322817-demo/test1 (master)
$ |
```


Git官方文档（中文）

<https://git-scm.com/book/zh/v2>

The End