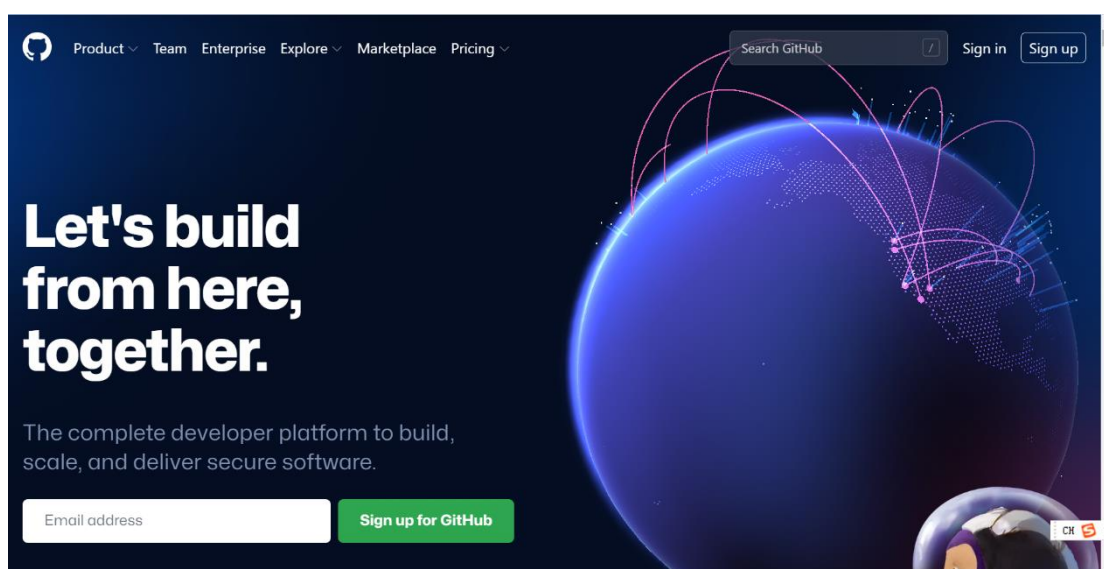


# 实战项目

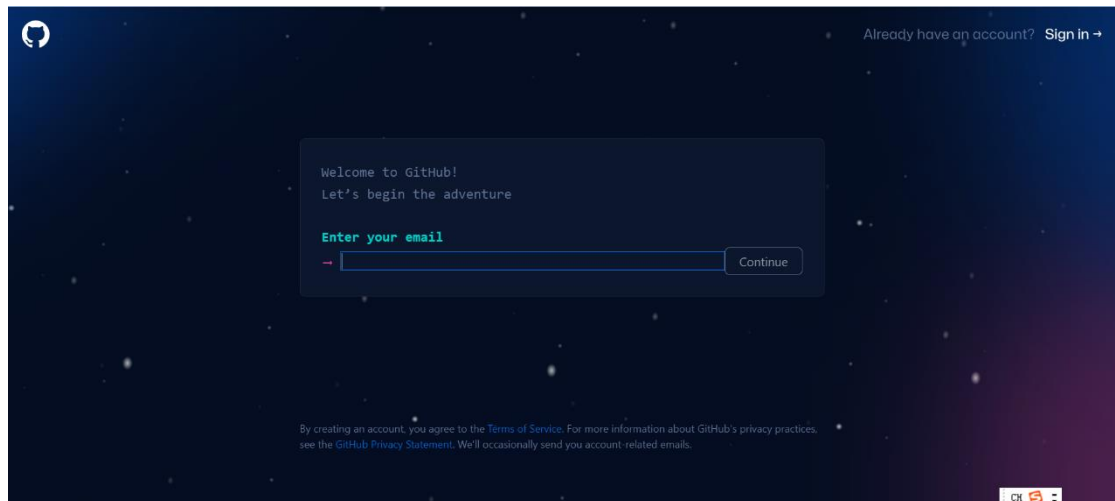
项目目标：在写代码的时候，通常会涉及到版本的更新迭代，保留旧版本的代码可以给新版本的代码提供基础，也是代码仓库。它可以直观的看到项目代码是如何一步步更新变化的。本项目将以 Github 为例，完成云端代码备份。

实战步骤：

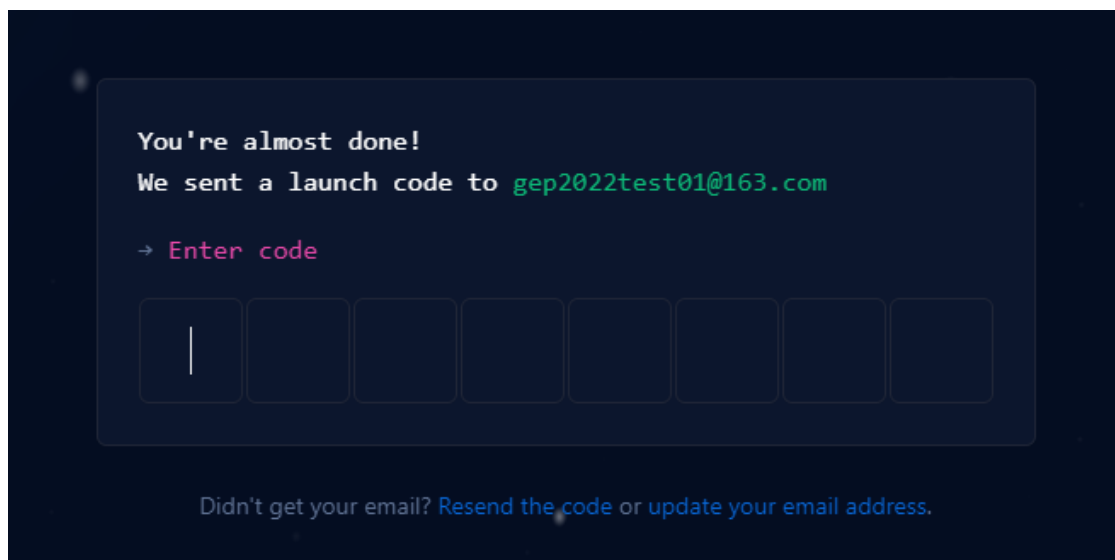
(1) Github 是一个代码仓库，可以用来管理代码版本，是程序员必备的工具。进入 Github 官网 <https://Github.com/>。



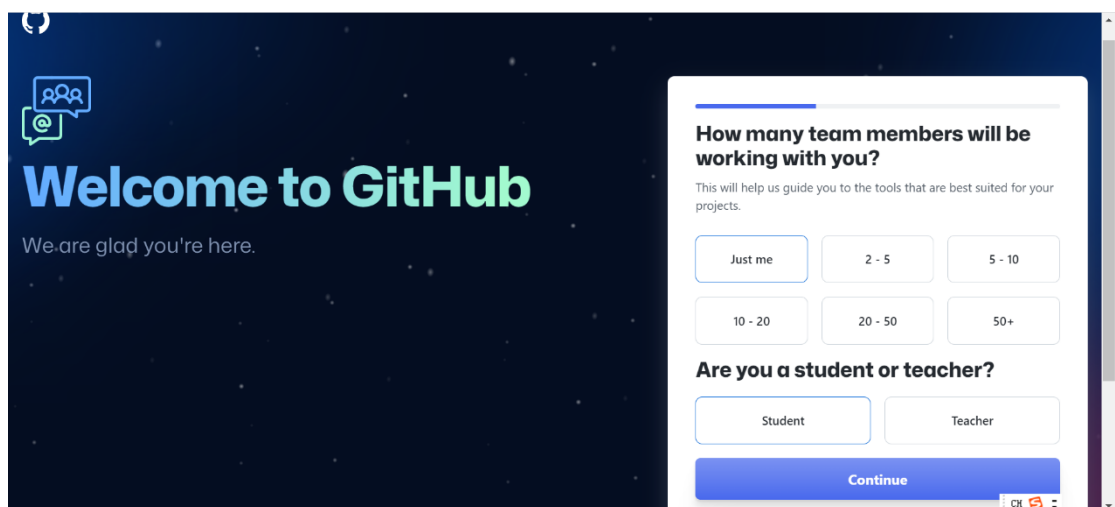
(2) 注册一个自己的账户。

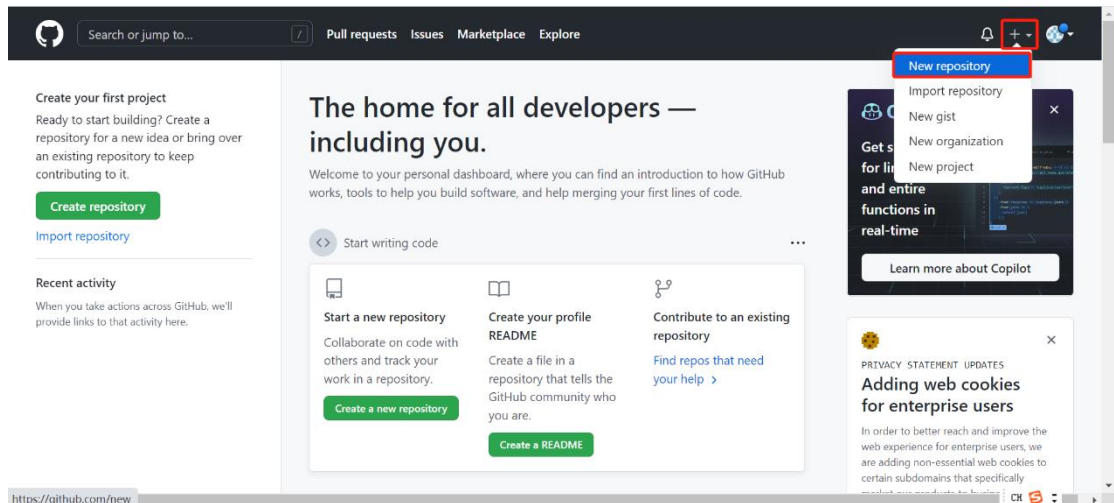


注册成功后，可以收到一个邮件。

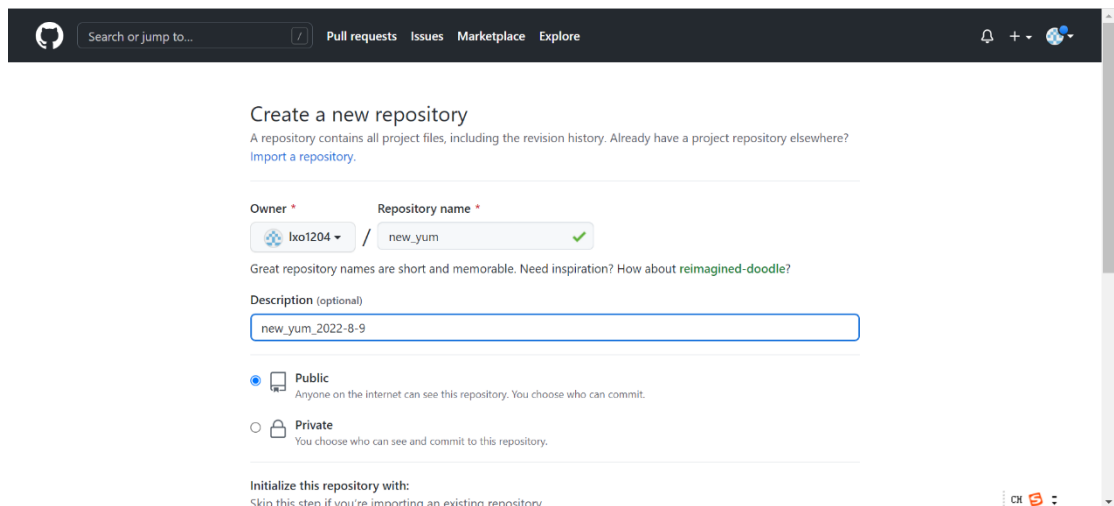


3、注册完成后，在右上角点击+号，创建一个 代码仓库。





#### 4、创建一个代码仓库。



Owner  /  仓库的名字

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

Description (optional) 对仓库的描述

☒ **Public**  
Anyone on the internet 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** 可选的生成README文件  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore:  | Add a license:  ⓘ

Owner \* Repository name \*

gepcloudtest / new\_yum ✓

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

Description (optional)

这是第一个云仓库

☒ Public  
Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:  
Skip this step if you're importing an existing repository.

☒ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

This will set `main` as the default branch. Change the default name in your [settings](#).

*i* You are creating a public repository in your personal account.

Create repository

5、创建完后进入管理界面，查看自己创建的代码仓库。

Search or jump to... Pull requests Issues Marketplace Explore

Overview Repositories 1 Projects Packages Stars

Popular repositories

new\_yum Public  
new\_yum\_2022-8-9

2 contributions in the last year

	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Mon										
Wed										
Fri										

Learn how we count contributions

NEW! View your contributions in 3D, VR and IRL!

Signed in as lxo1204

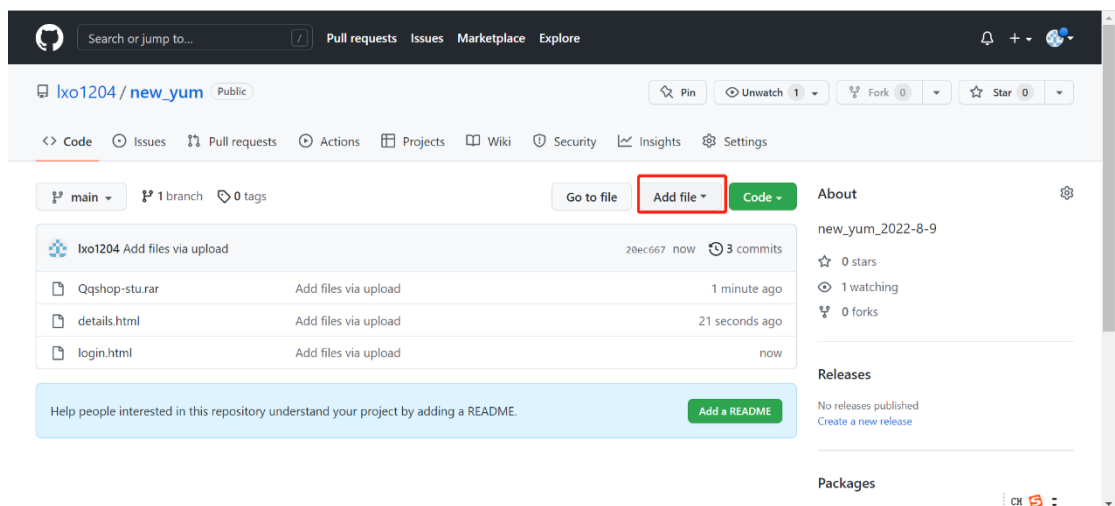
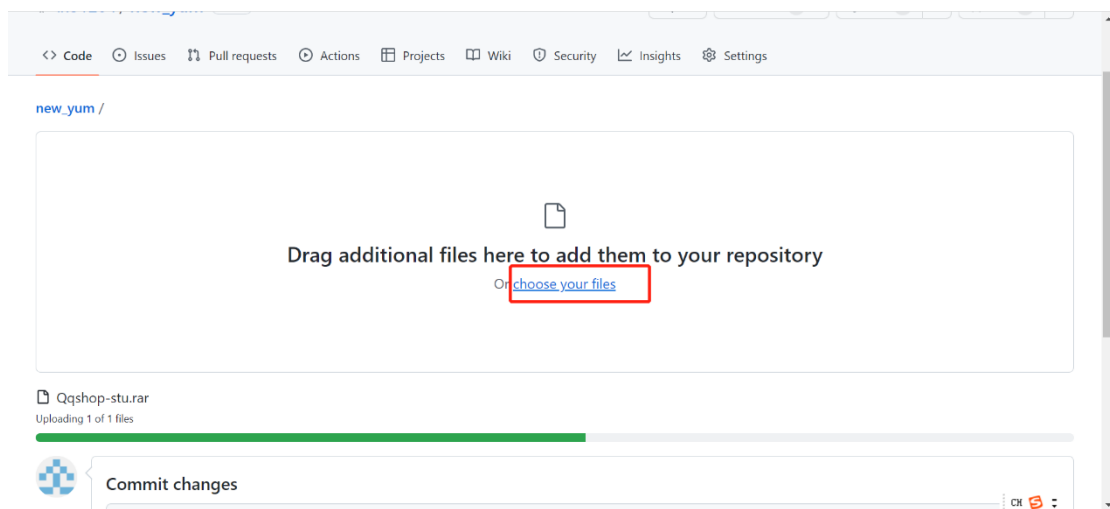
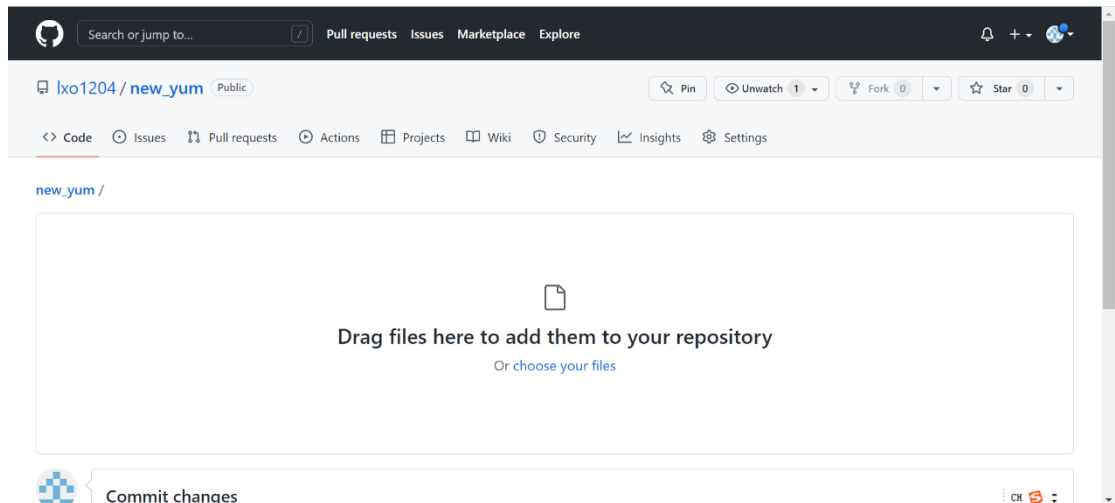
- Set status
- Your profile
- Your repositories
- Your codespaces
- Your projects
- Your stars
- Your gists
- Upgrade
- Feature preview
- Help
- Settings
- Sign out

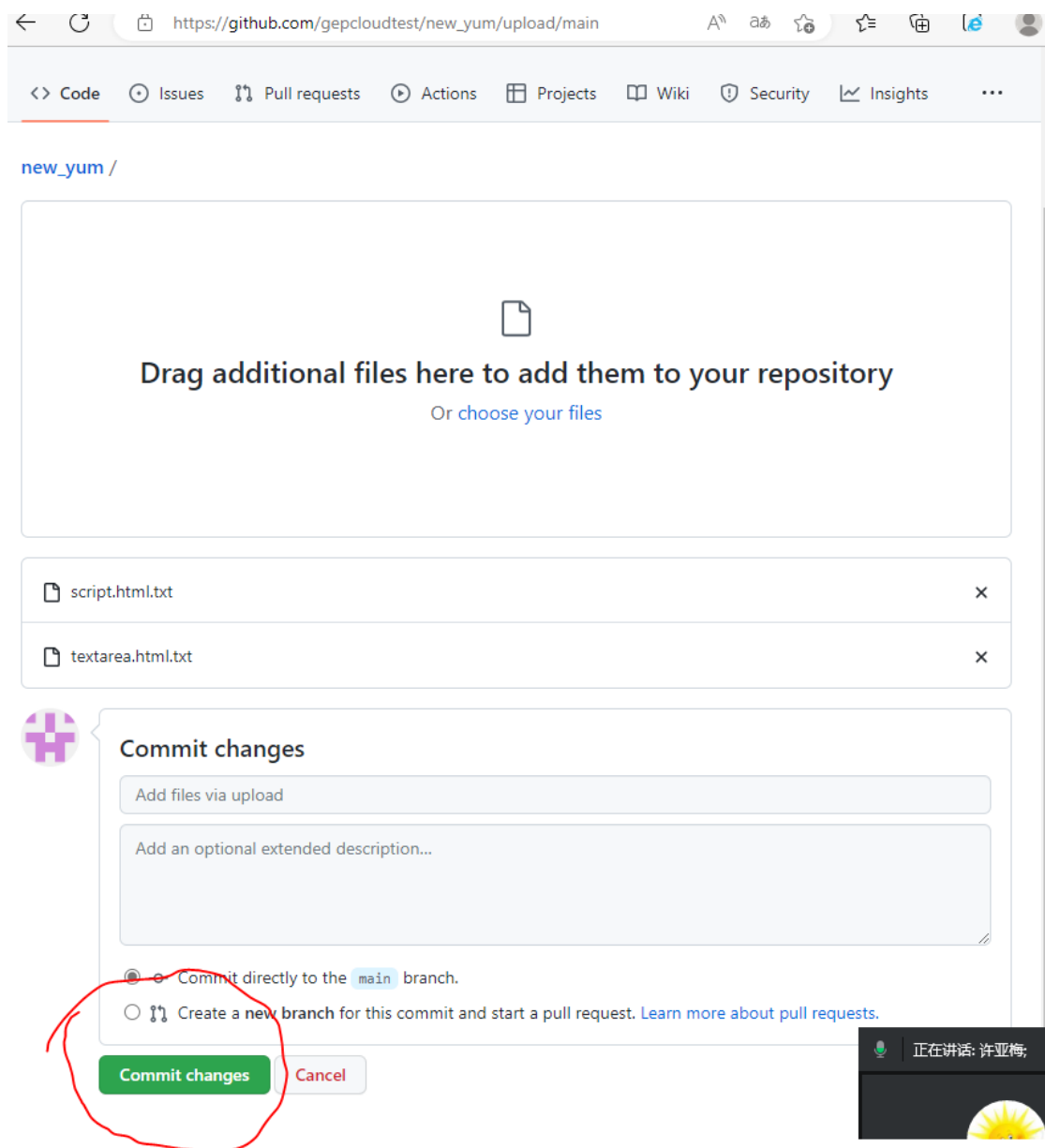
Less More

Joined 24 minutes ago

Edit profile

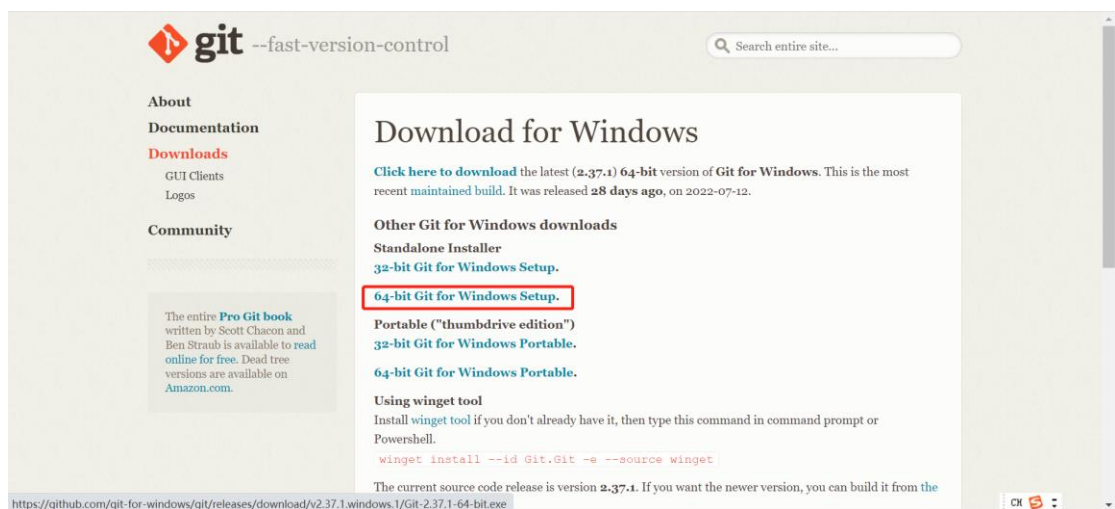
6、进入创建好的仓库。



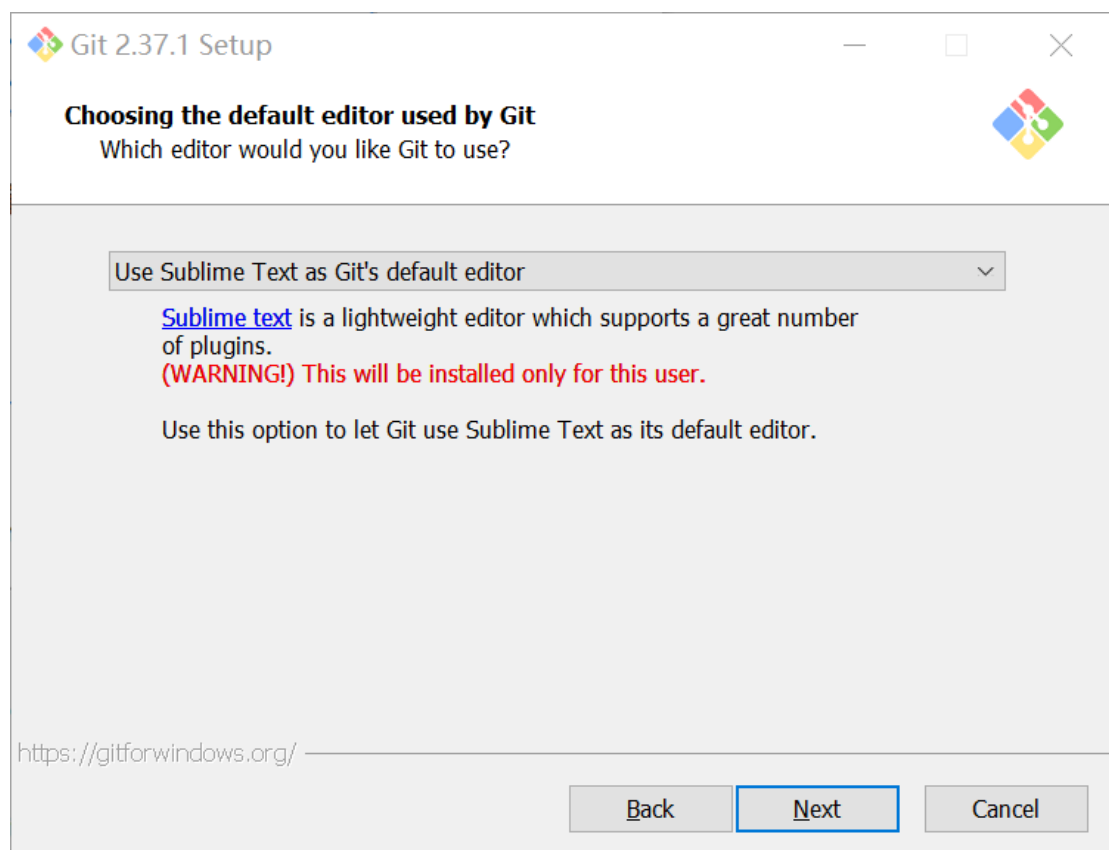
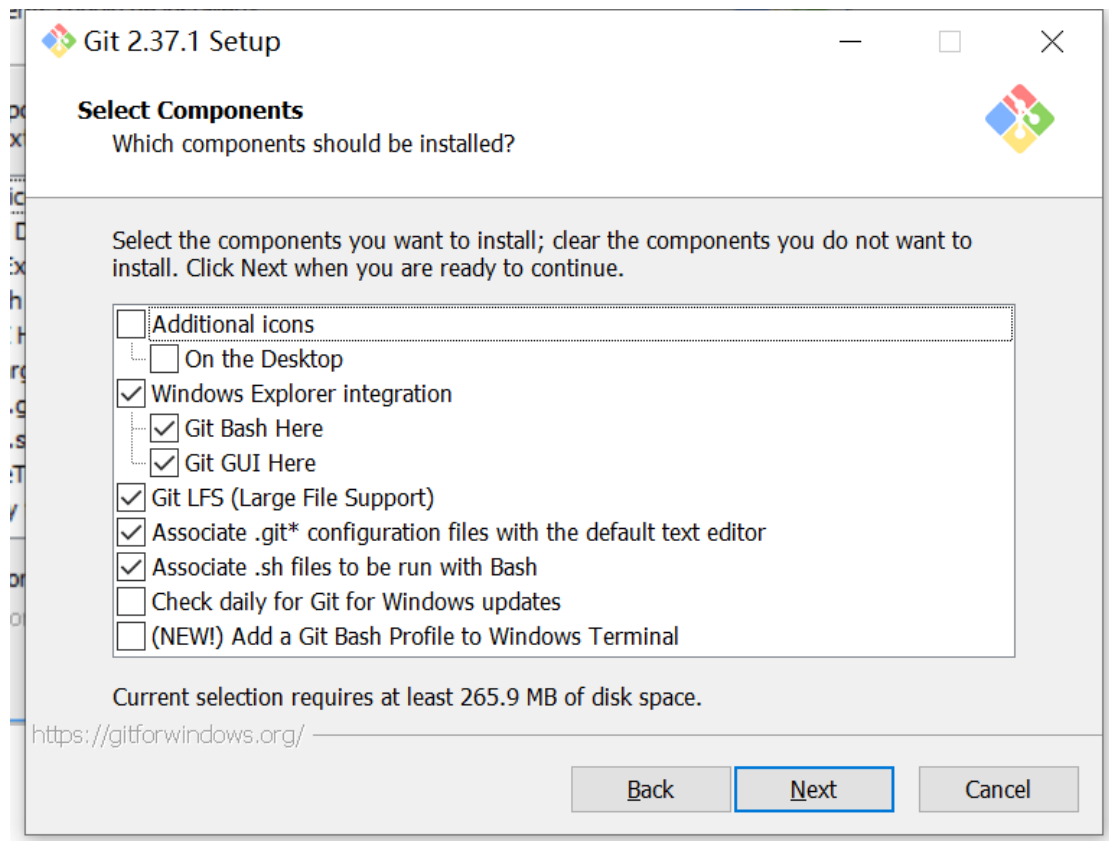


这里是文件列表，左侧是文件名称，右侧是文件最后修改时间，中间对文件的描述。

7、创建完成代码仓库后需要安装一个软件，Git Bash，用于管理 Github 。首先进入 Git 官网 <https://git-scm.com/>。下载适合自己电脑版本的 Git。

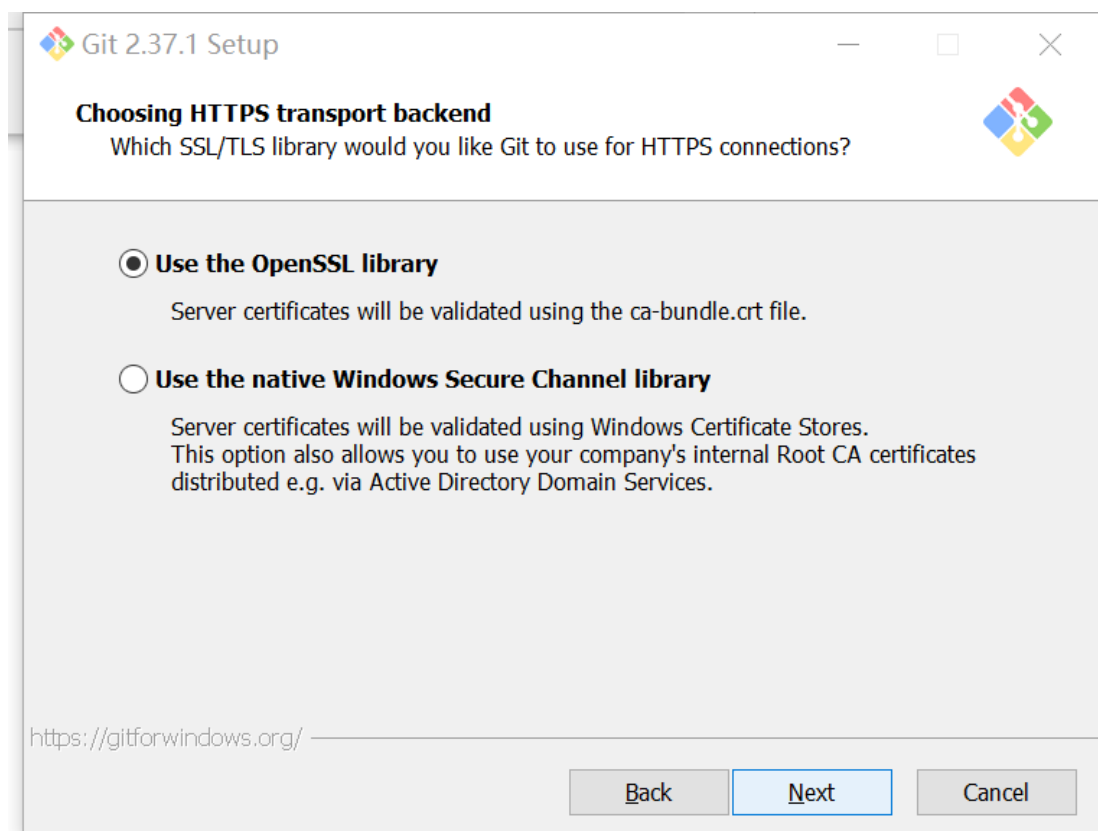
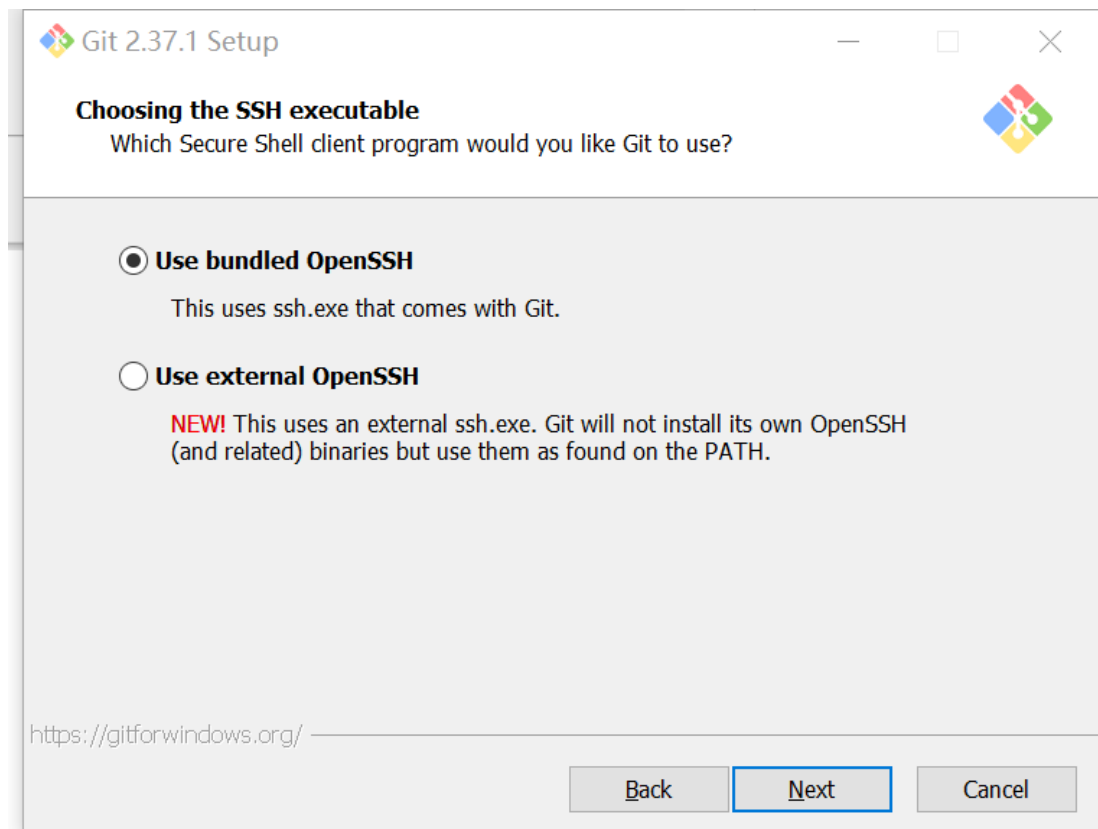


8、安装运行，安装过程中，选项默认选择的即可。选择适合的默认编辑器。



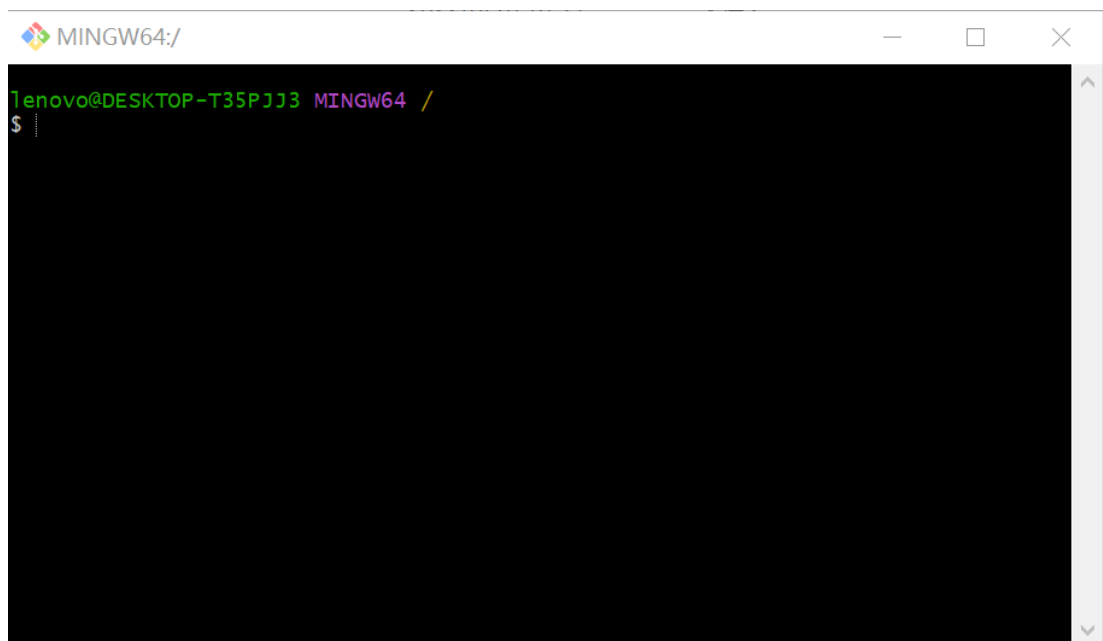
9、选择加密的通信形式。





10、除了以上步骤需要注意，其他步骤直接默认选择“next”即可。安装后，在安装目录中找到“git-bash.exe”，双击运行。

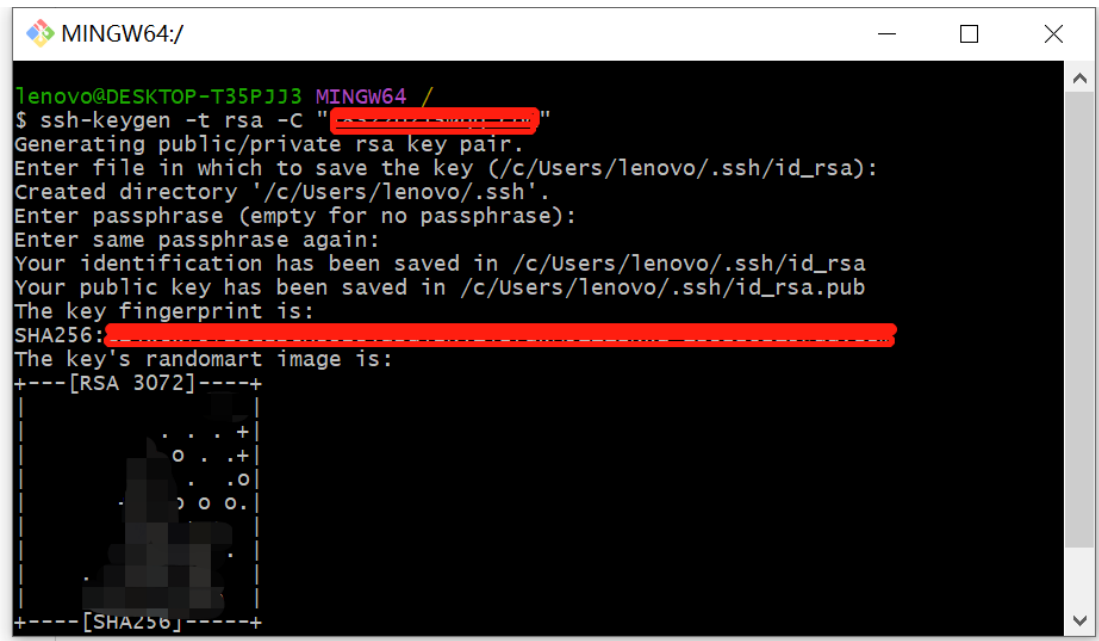
名称	修改日期	类型	大小
bin	2022/8/10 10:33	文件夹	
cmd	2022/8/10 10:33	文件夹	
dev	2022/8/10 10:33	文件夹	
etc	2022/8/10 10:33	文件夹	
mingw64	2022/8/10 10:33	文件夹	
tmp	2022/8/10 10:33	文件夹	
usr	2022/8/10 10:33	文件夹	
git-bash.exe	2022/7/11 18:32	应用程序	135 KB
git-cmd.exe	2022/7/11 18:32	应用程序	135 KB
LICENSE.txt	2022/7/11 18:51	文本文档	19 KB
ReleaseNotes.html	2022/7/11 18:52	Microsoft Edge HT...	219 KB
unins000.dat	2022/8/10 10:33	dat file	1,227 KB
unins000.exe	2022/8/10 10:27	应用程序	3,137 KB
unins000.msg	2022/8/10 10:33	Outlook 项目	24 KB



11、接下来需要获取密钥，管理代码必须的认证。密钥存放位置为“C/users/lenovo/.ssh/id\_rsa.pub”（可以在配置中修改，回车则是默认给出的位置）。输入以下代码。

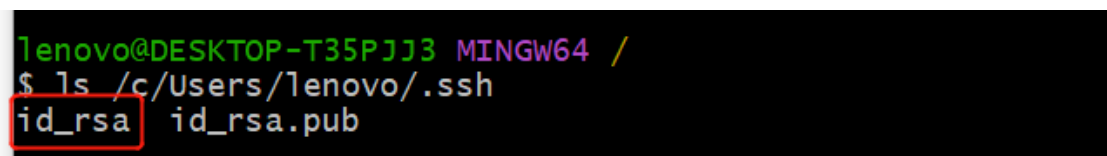
```
$ ssh-keygen -t RSA -C "your_email@youremail.com"
```

引号内改成自己注册 Github 时绑定的邮箱。

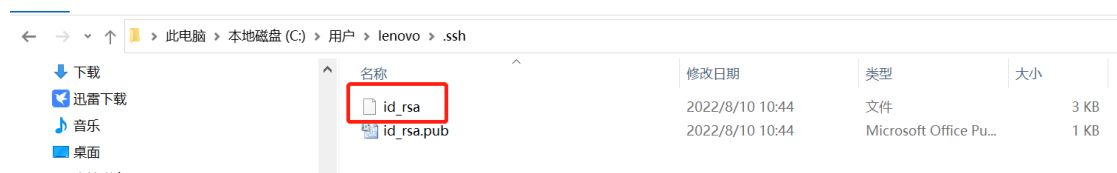


```
lenovo@DESKTOP-T35PJ33 MINGW64 /
$ ssh-keygen -t rsa -C "[redacted email]"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/lenovo/.ssh/id_rsa):
Created directory '/c/Users/lenovo/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/lenovo/.ssh/id_rsa
Your public key has been saved in /c/Users/lenovo/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:[redacted]
The key's randomart image is:
+----[RSA 3072]-----+
|
| . . . +
| o . . +
| . o
| . o
| o o
+-----[SHA256]-----+
```

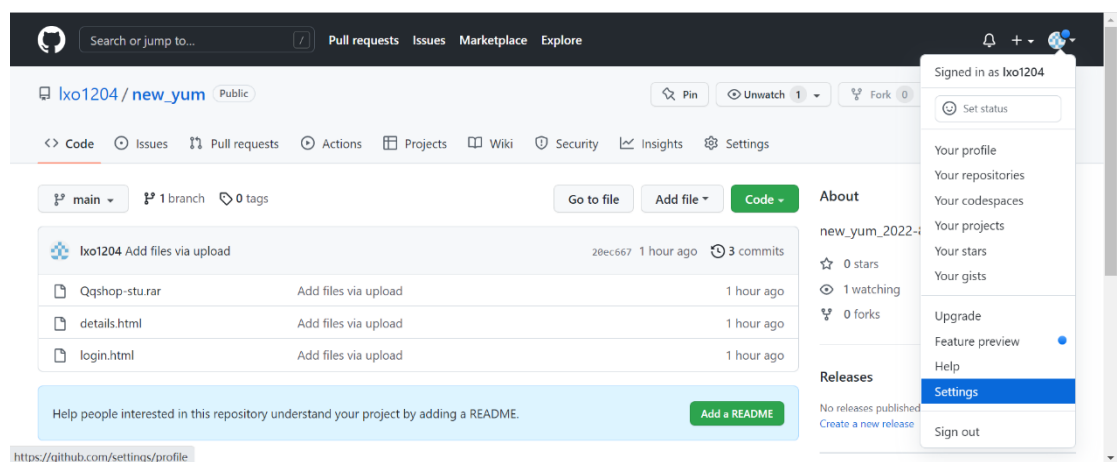
12、成功获取到密钥。到之前提示存放的位置找到密钥文件。

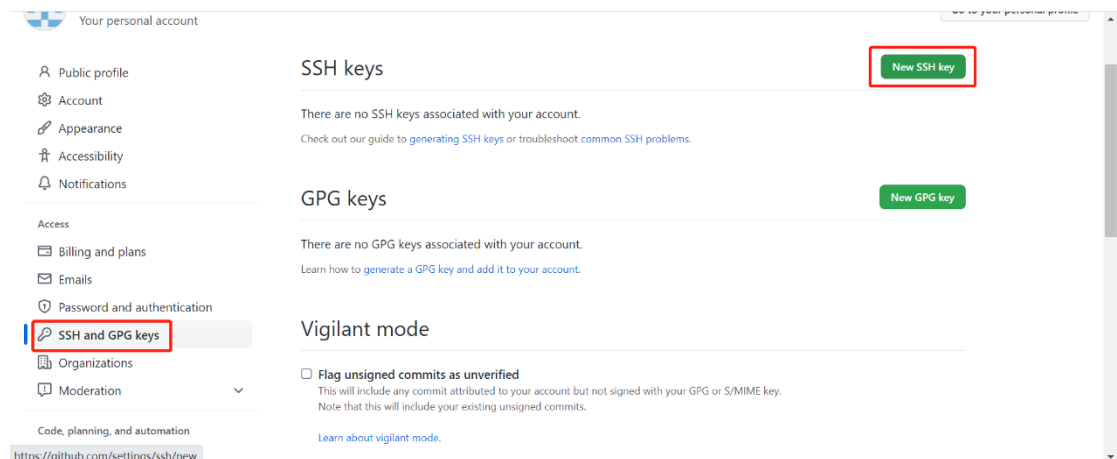


```
lenovo@DESKTOP-T35PJ33 MINGW64 /
$ ls /c/Users/lenovo/.ssh
id_rsa id_rsa.pub
```

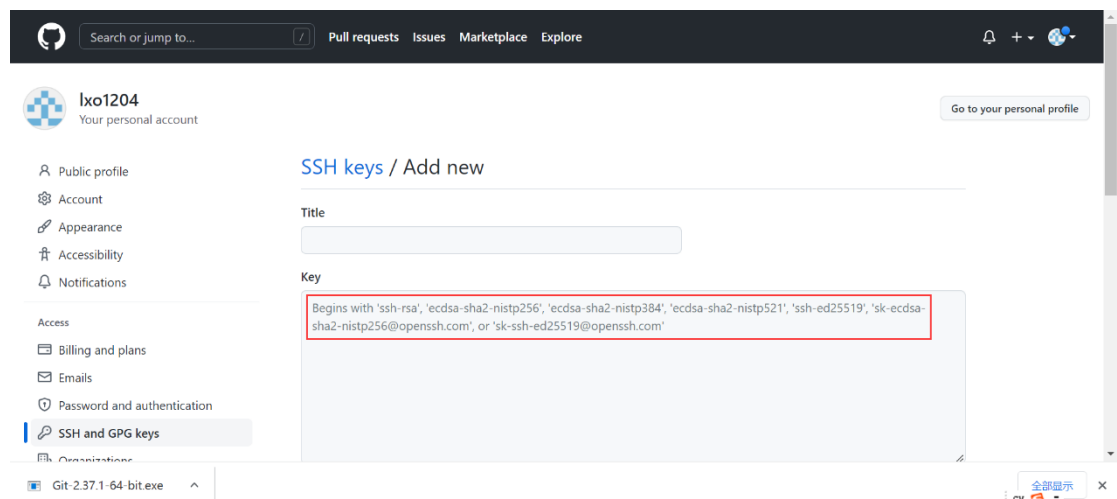


13、找到密钥后，回到 Github，进入设置界面。





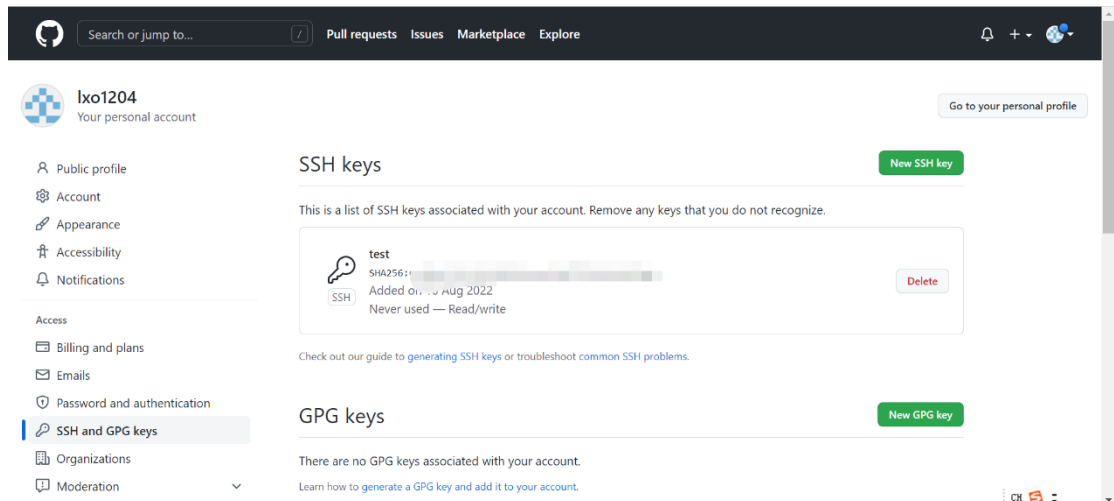
14、点击 New SSH key 新建一个 ssh 的密钥，把之前生成的公钥文件 (id\_rsa.pub) 里的内容复制到 Key 里面。 点击 Add SSH key 添加即可。回到 git bash，输入 ssh -T git@Github .com。



复制密钥时不能直接打开文件复制，会改变其格式，因此使用以下方法将密钥复制出来，然后在到 key 文本框中进行粘贴。

```
lenovo@DESKTOP-T35PJ33 MINGW64 /
$ cd /c/users/lenovo/.ssh

lenovo@DESKTOP-T35PJ33 MINGW64 /c/users/lenovo/.ssh
$ clip < id_rsa.pub
```



15、成功添加后，会发送验证信息到注册邮箱，进入邮箱后点击链接进入。



16、回到 git bash，输入 `ssh -T git@Github.com`，输入你的密钥，如果出现图下内容，则说明绑定成功。接下来需要做一些简单的设置。

```
git config --global user.name "your name"
```

```
git config --global user.email "your email@yourmail.com"
```

```

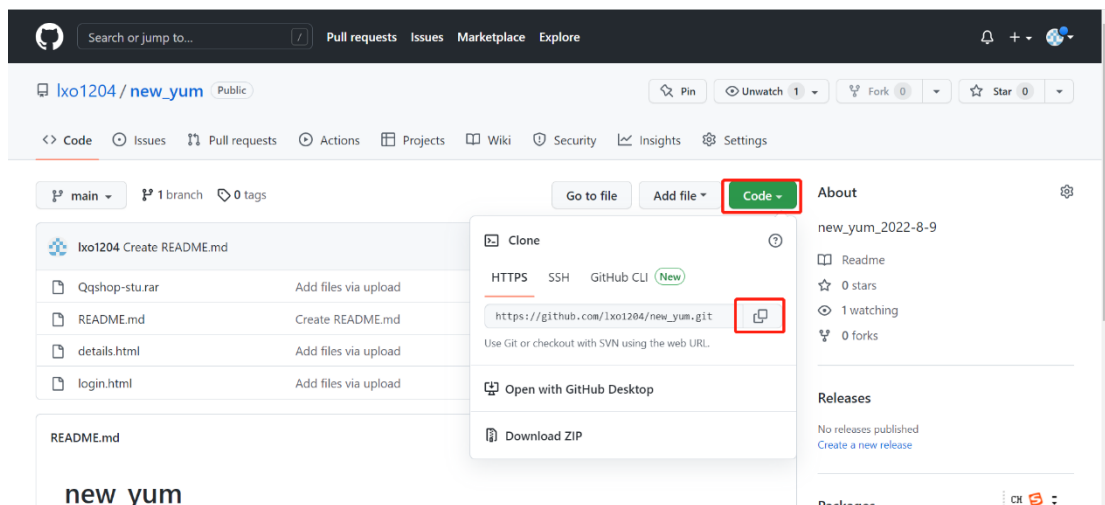
lenovo@DESKTOP-T35PJ33 MINGW64 /f/test/new_yum (main)
$ ssh -T git@github.com
Enter passphrase for key '/c/Users/lenovo/.ssh/id_rsa':
Hi [redacted]! You've successfully authenticated, but GitHub does not provide shell
access.

lenovo@DESKTOP-T35PJ33 MINGW64 /f/test/new_yum (main)
$ git config --global user.name "[redacted]"

lenovo@DESKTOP-T35PJ33 MINGW64 /f/test/new_yum (main)
$ git config --global user.email "[redacted]@qq.com"

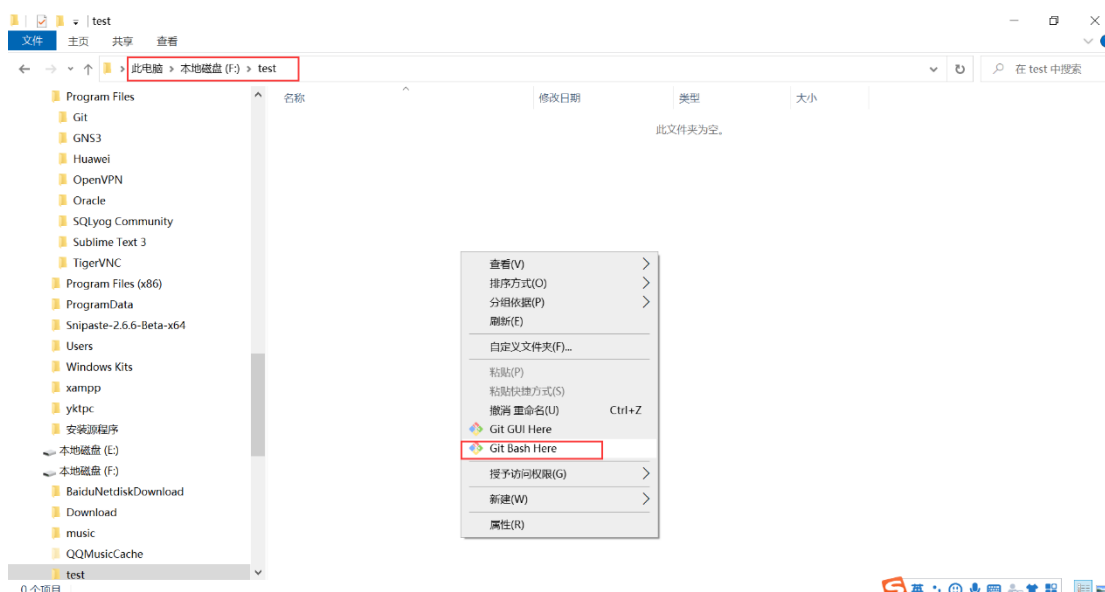
```

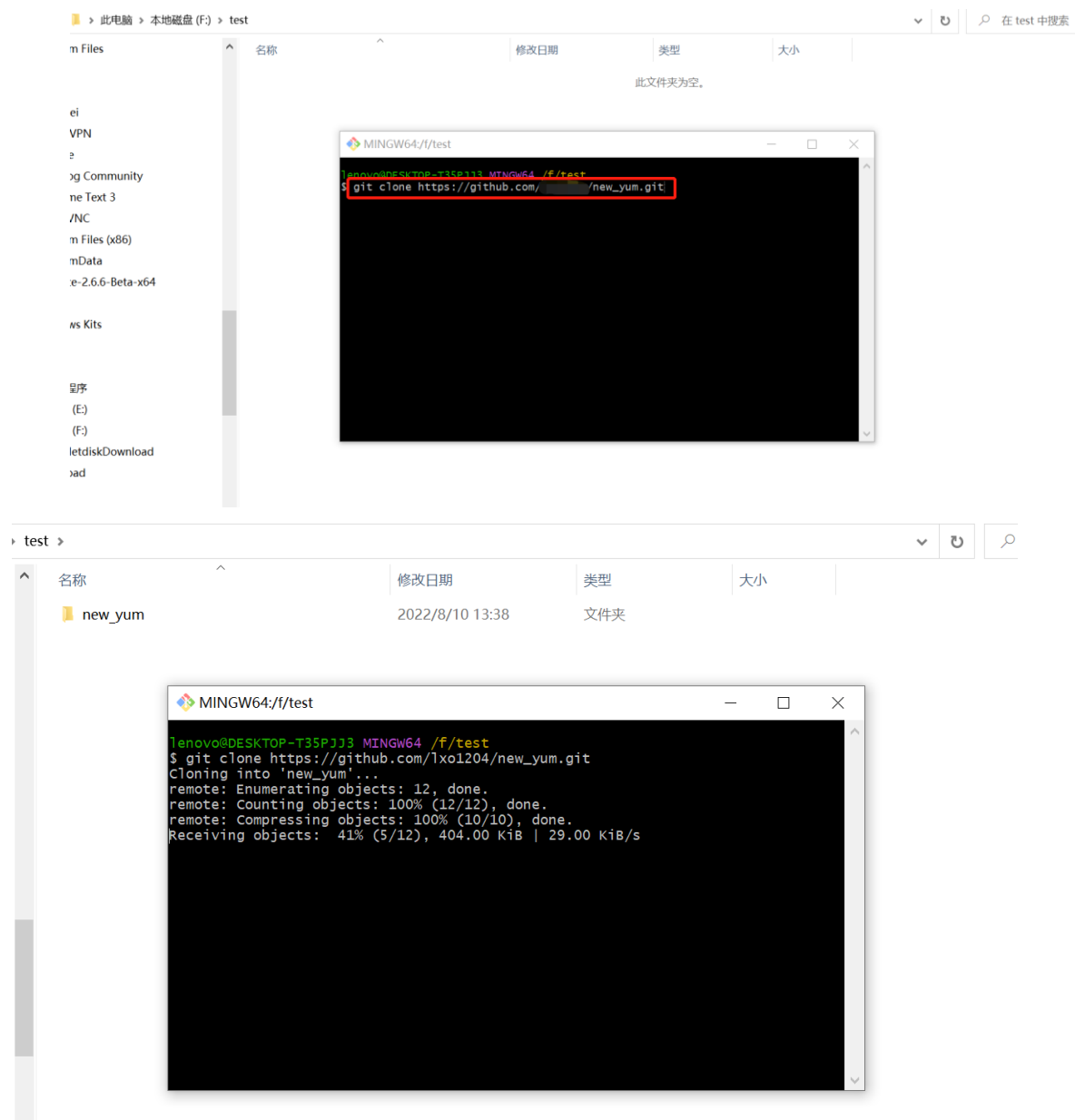
17、回到 Github 仓库主页面，现在需要将 Github 克隆到本地，在 Github 上找到克隆的地址。



回到 git-bash 窗口，输入命令 git clone，将仓库克隆到本地，本地的位置需要在输入命令前选定。

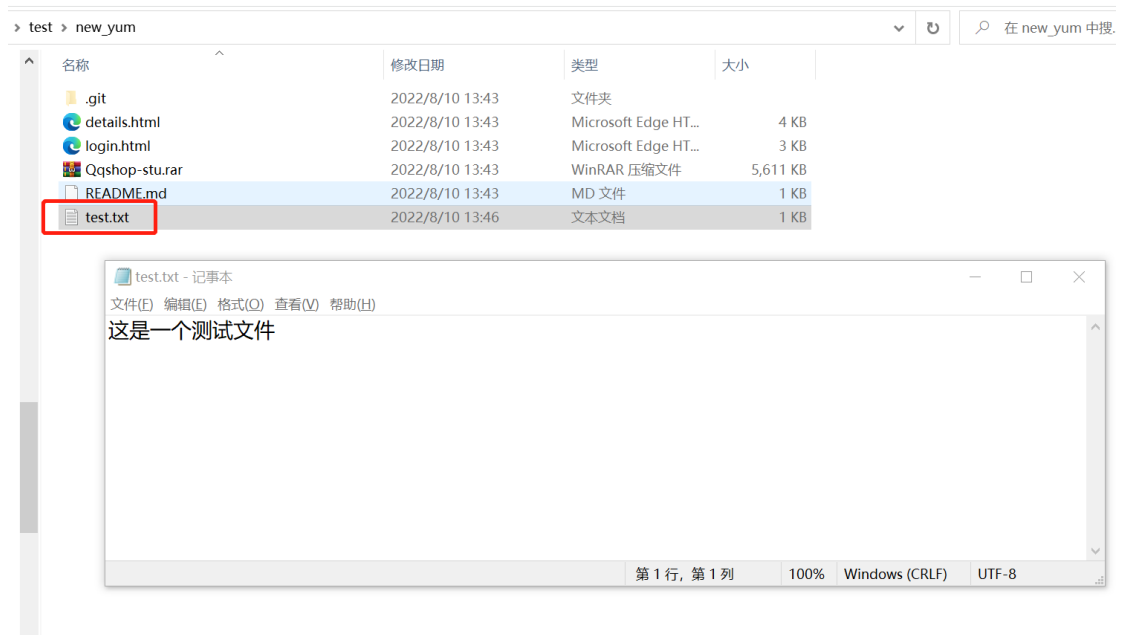
选择指定的文件夹，进入后单击右键，打开 git-bash 窗口。





18、输入 `ls` 可以查看当前目录下的文件信息，用 `git add` 命令添加需要上传的文件。

进入 new\_yum 文件夹，创建一个测试文件 test.txt。



```
lenovo@DESKTOP-T35PJ33 MINGW64 /f/test/new_yum (main)
$ ls
Qqshop-stu.rar  README.md  details.html  login.html  test.txt

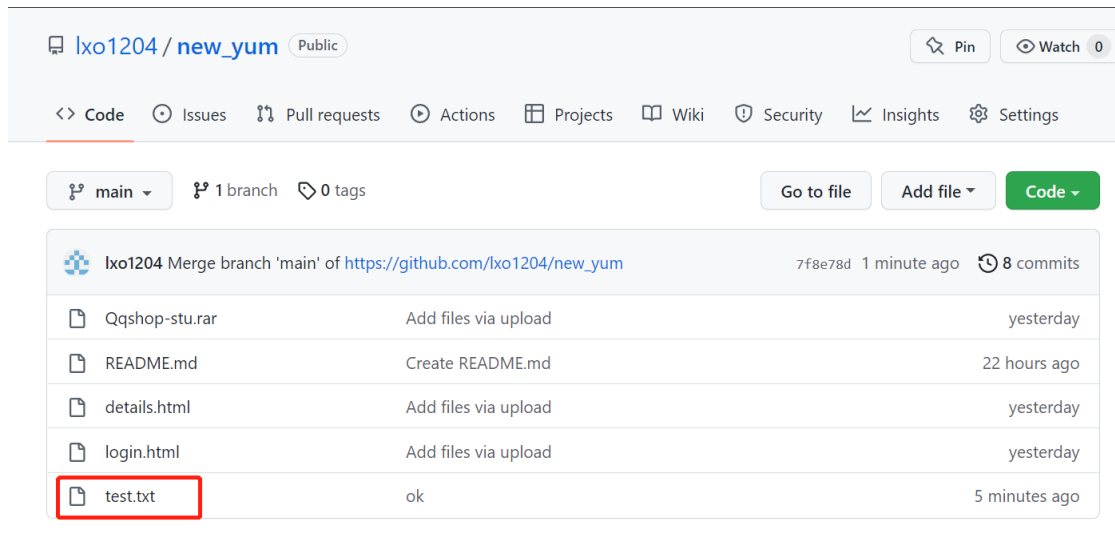
lenovo@DESKTOP-T35PJ33 MINGW64 /f/test/new_yum (main)
$ git add test.txt

lenovo@DESKTOP-T35PJ33 MINGW64 /f/test/new_yum (main)
$ git commit -m "ok"
[main a611ad2] ok
1 file changed, 1 insertion(+)
create mode 100644 test.txt
```

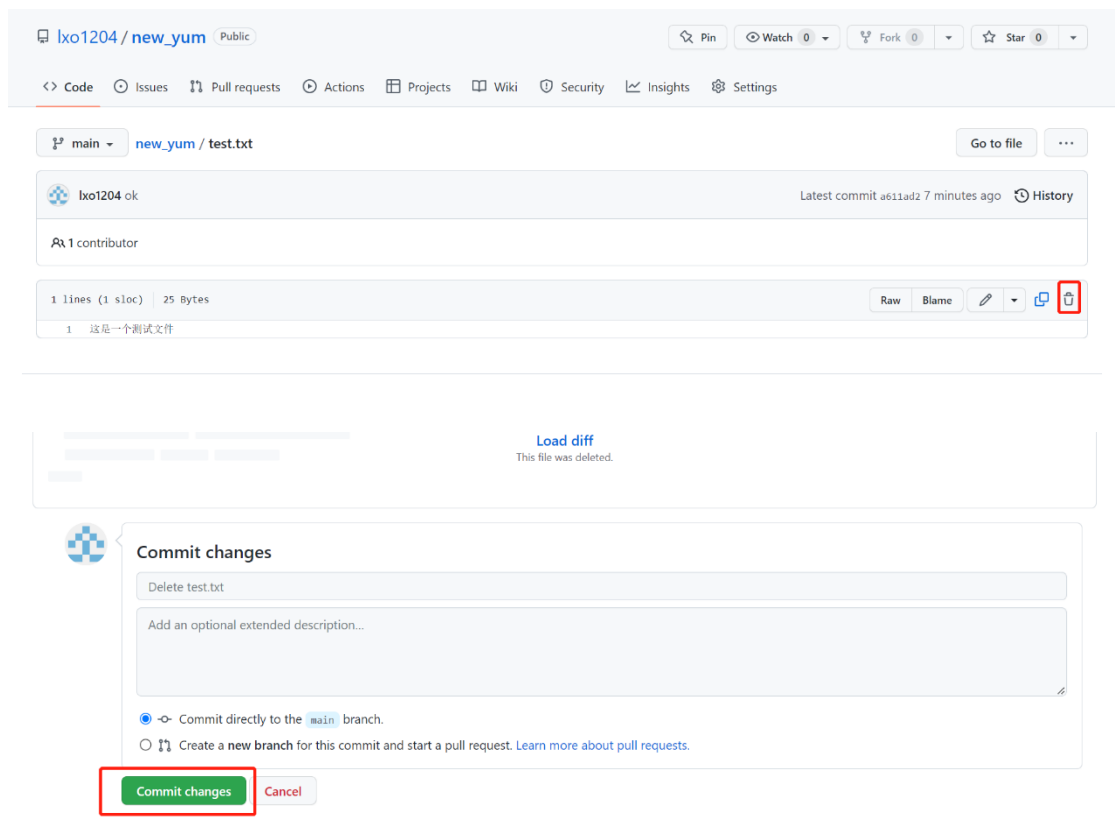
19、添加了之后用 `git push origin main` 命令将添加的文件上传到 Github 。文件就上传成功。

```
lenovo@DESKTOP-T35PJ33 MINGW64 /f/test/new_yum (main)
$ git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 477 bytes | 477.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/lxol204/new_yum.git
34b2448..7f8e78d main -> main
```





20、如果对文件有修改，比如删除，可以在页面上操作。



回到本地，使用 `git pull` 命令，则会把页面中删除的内容在本地也进行更新。 `git pull` 命令用于从远程获取代码并合并本地的版本。



Github 参考资料:

<https://www.runoob.com/git/git-basic-operations.html>