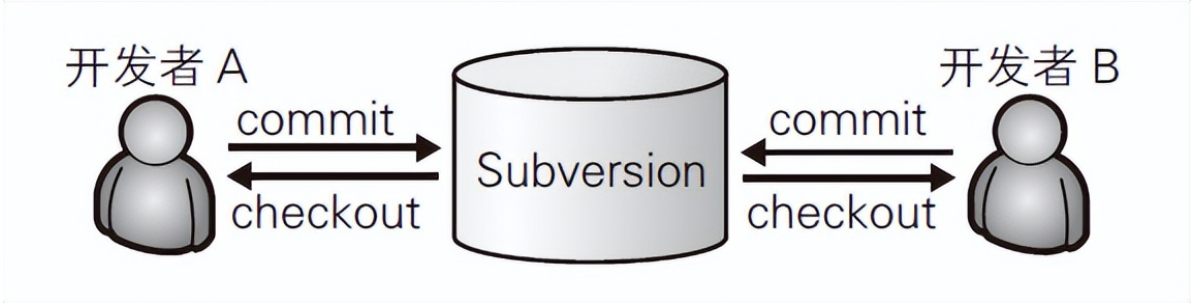
什么是git?

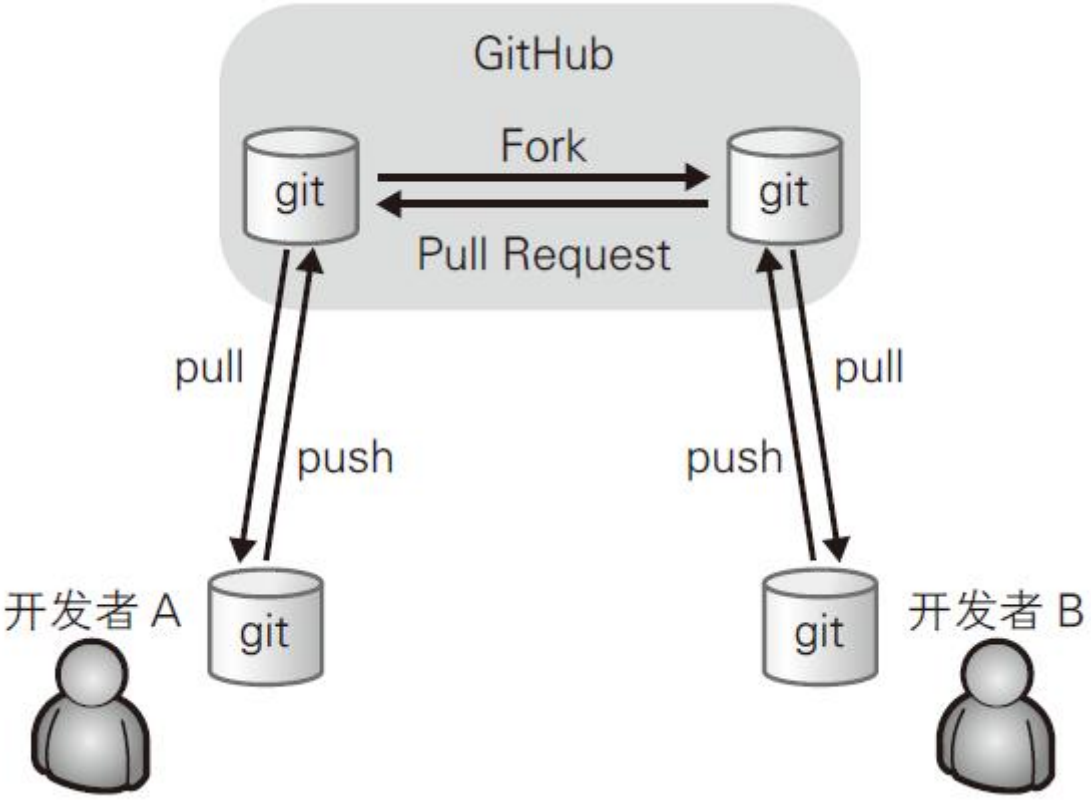
git是一种分布式版本控制系统，首先要理解什么是版本控制，我今天写了5000字论文，要经过导师多次建议修改，第二天我改了一些内容，比较稳妥的方案是，把论文复制一下做备份，之后改到第N版时觉得还是第1版方案好，那就复制粘贴回退到第1版，这就是版本控制。

那么对于程序员来说，项目开发的代码版本也是需要不断备份，且更多时候是多人协作，共同开发一个大型项目。这种对原始项目复制粘贴的备份方法显然过于笨重，因此就出现了很多软件版本控制，比如git、SVN等，可以自由选择回退到之前哪一个版本。

软件版本控制有分为集中式和分布式，比如SVN，需要一个中心服务器，所有软件历史版本都在中心服务器存储，每个开发者首先要从服务器获取最新的源代码，开发完成再向服务器推送最新代码。然而一旦服务器出现故障，开发者就无法获取最新的代码。

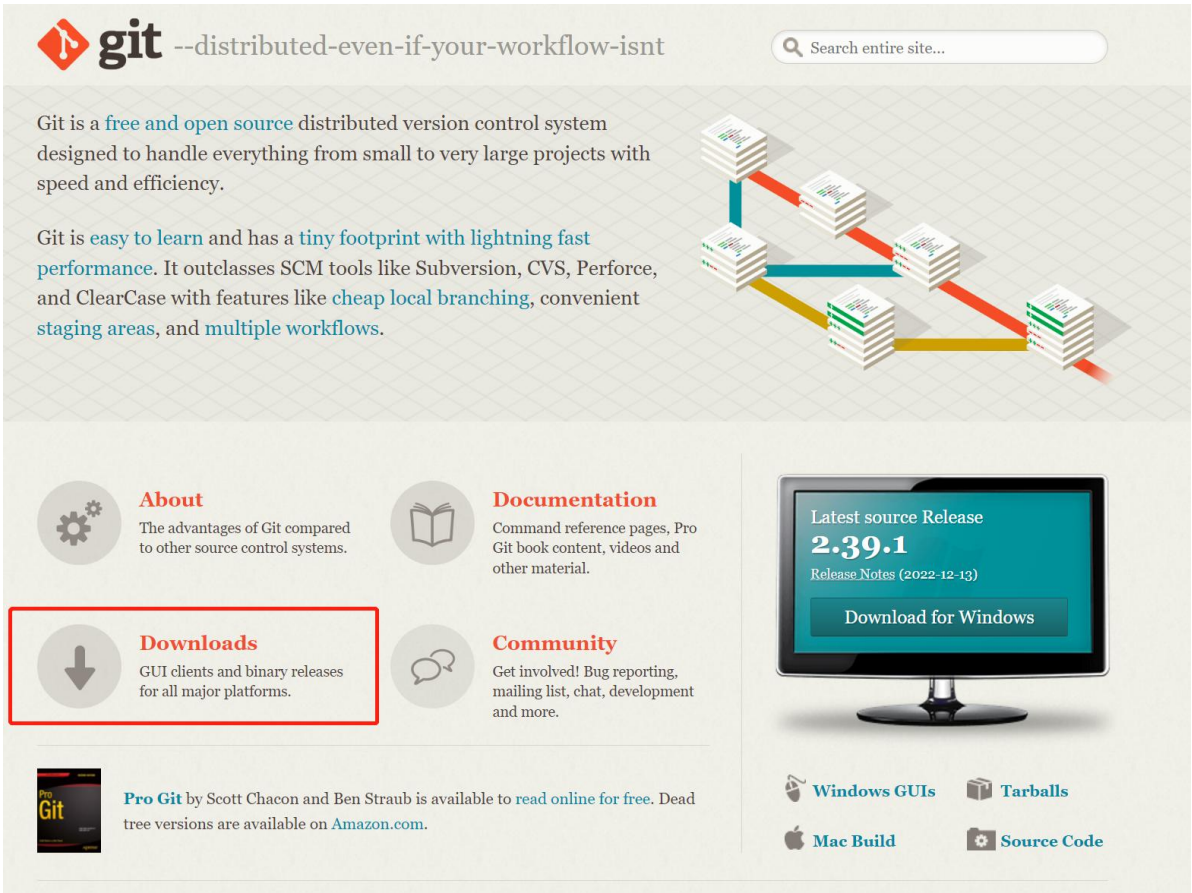


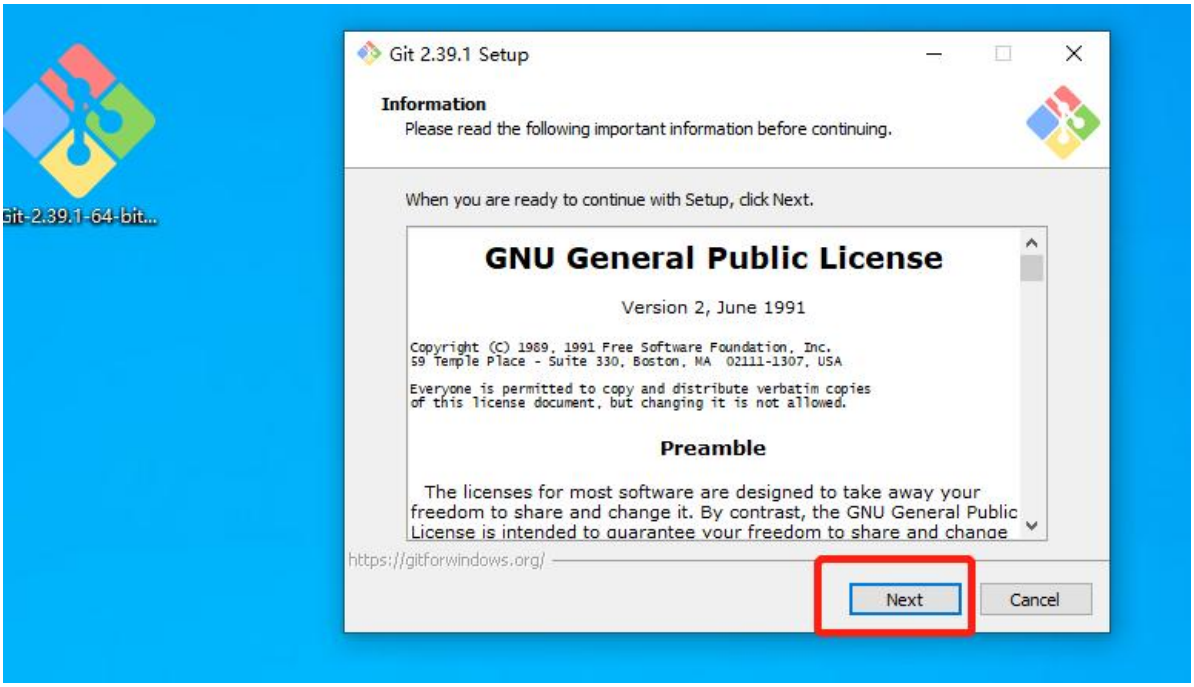
而git是分布式管理，每个开发者的电脑上都存储完整的历史版本，可以自己选择时间向服务器推送，然后服务器进行代码合并。开发团队可以自建中心服务器也可以使用GITHUB网站管理。即时中心服务器出现故障，每个客户端也存储着项目完整的历史版本。



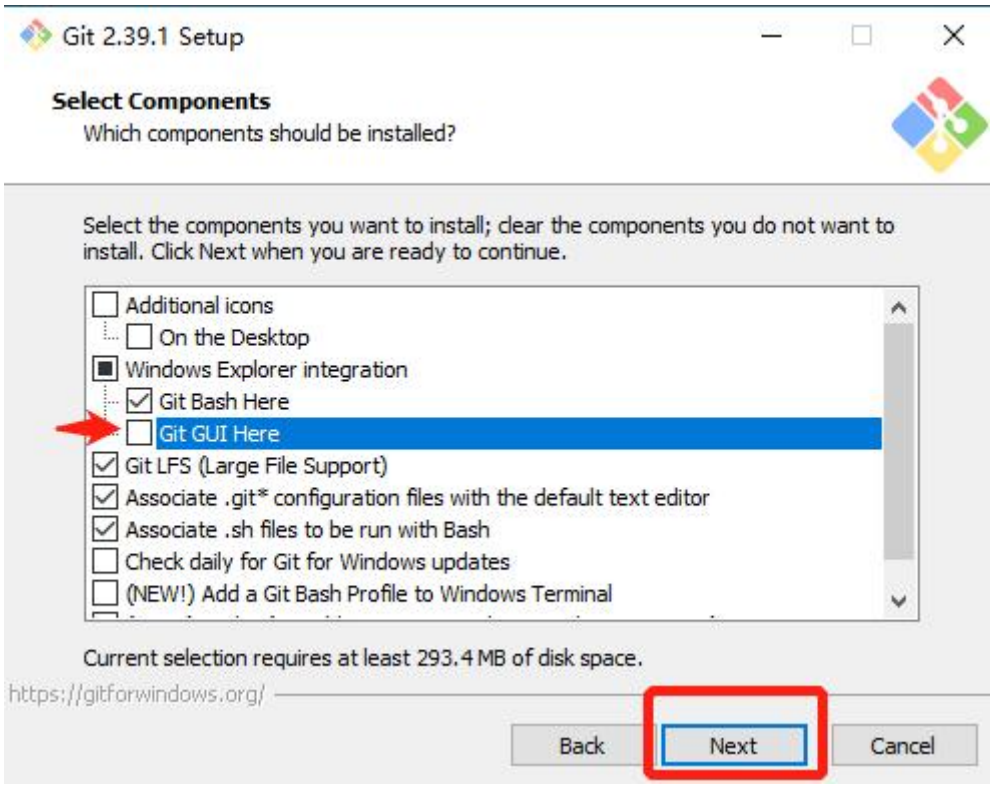
git配置

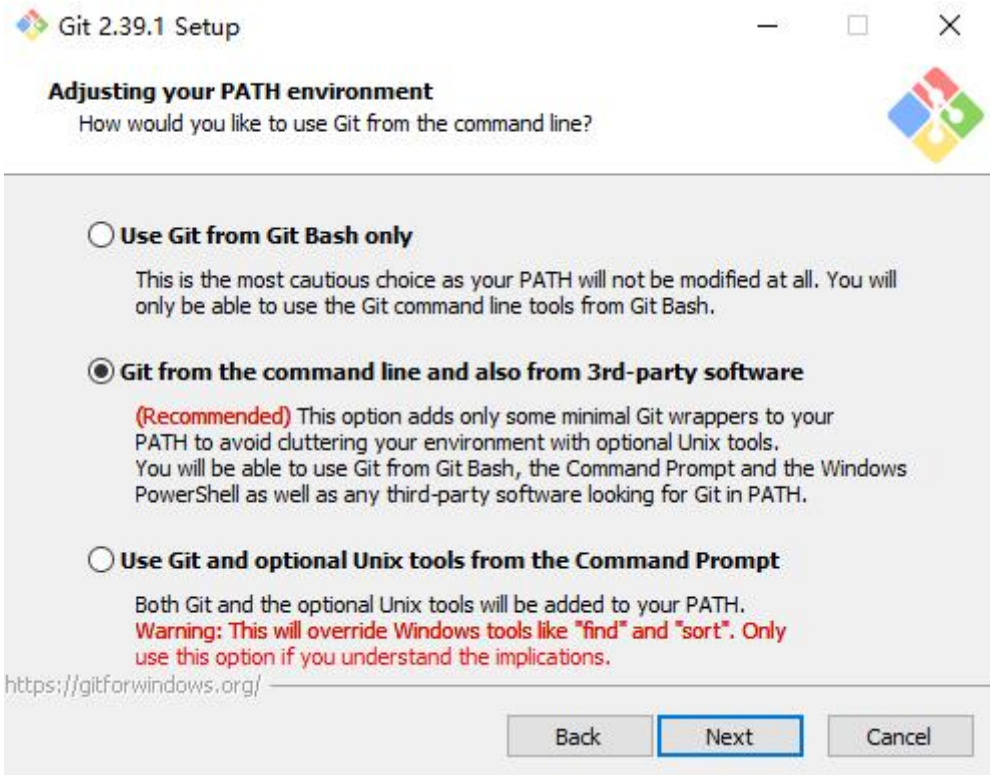
git支持多种操作系统，Windows系统直接官网下载git安装包，双击安装就可以：



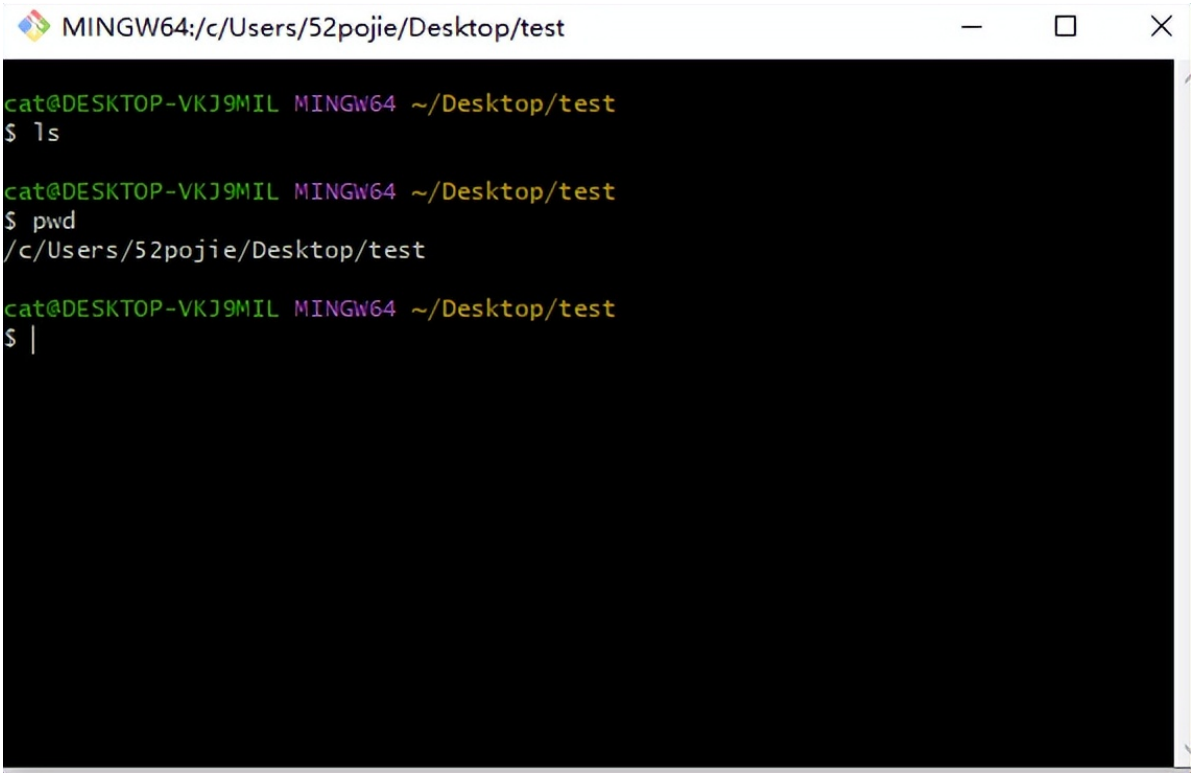


GUI图形化一般不需要，可以根据需求自主选择：





新建项目文件夹test，进入打开git，鼠标右键-git bash打开git终端：

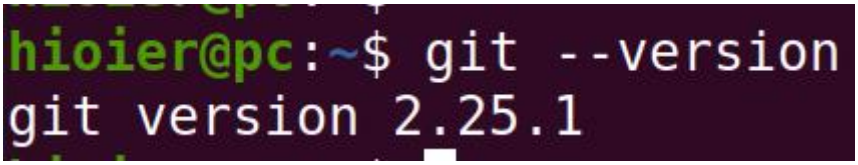


终端中基础命令和Linux基础命令大部分相同，当然，也可以使用cmd或者powershell等其他终端，这一步在安装git过程中进行了选择。

在Ubuntu20.04中，通过apt包安装git：

```
sudo apt install git
```

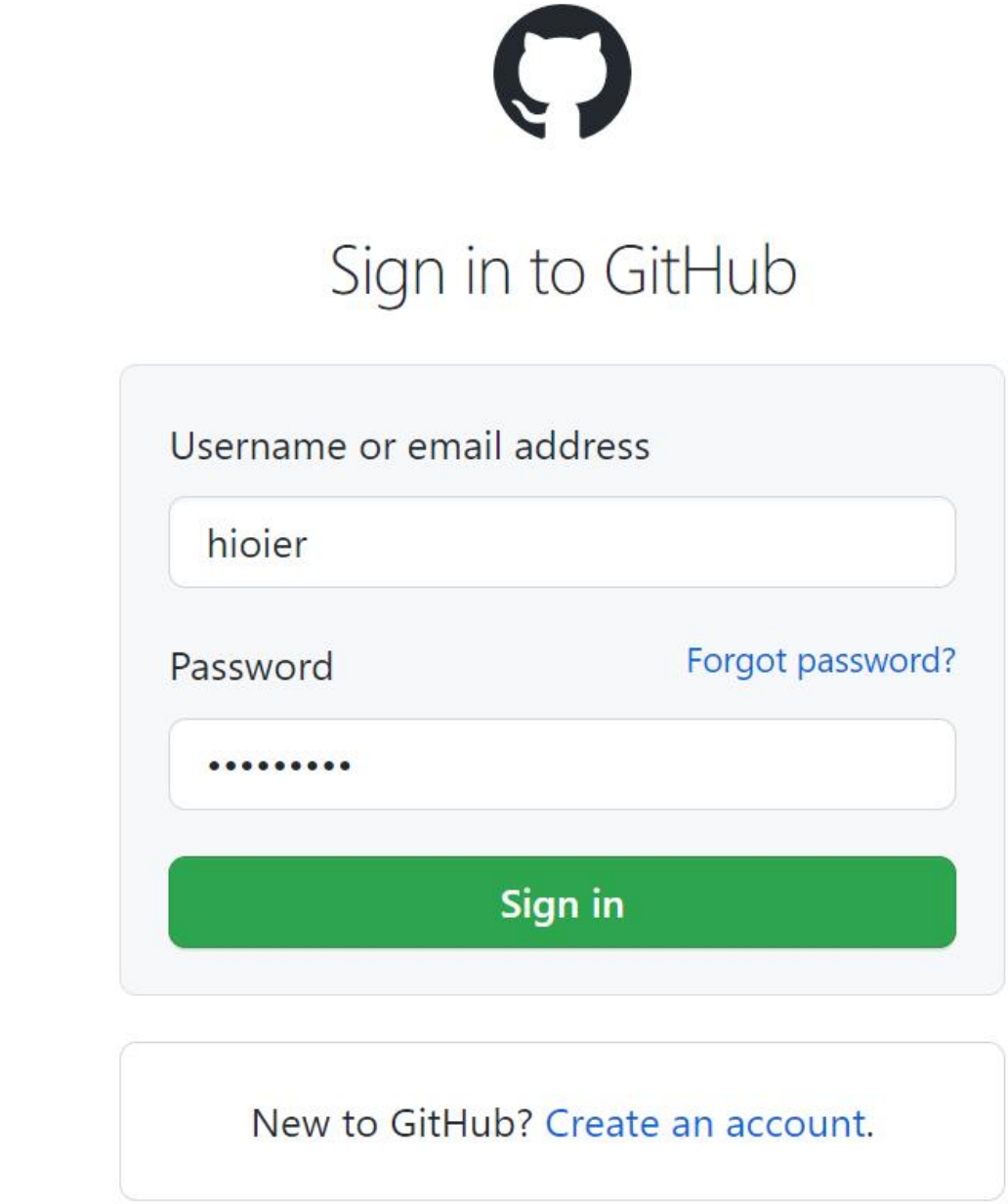
安装完毕，输出git版本，检测是否成功：



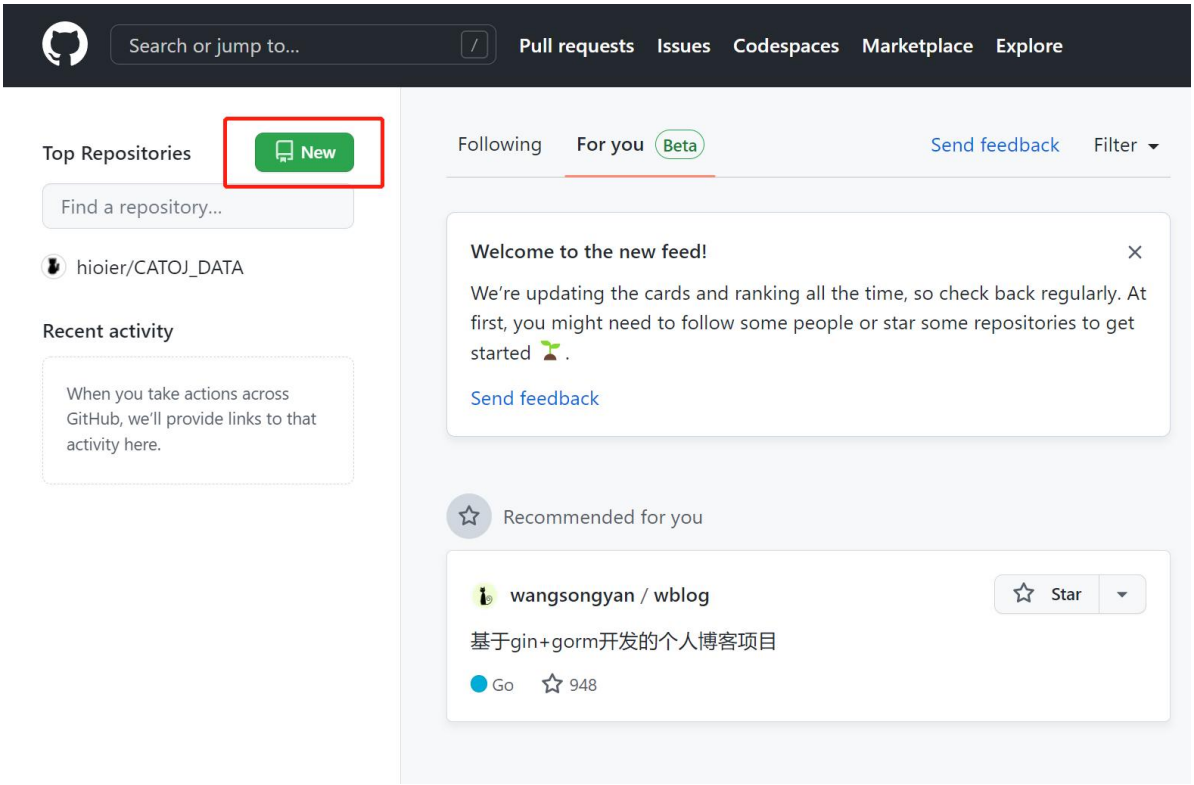
连接Github

github网站被誉为是全球最大的同性交友网站（男性程序员较多），每个程序员的项目都可以通过github服务器做管理，而不需要自己搭建服务器。而且互联网的开源精神影响，程序猿也热衷于将自己的项目在github开源，其他人认为你的项目有价值会打星、fork，因此，很多明星项目都是在github上开源，逐渐形成社区，比如Vue、jQuery、Ruby.....github项目的火热程度也作为程序员技术水平衡量的重要标准。

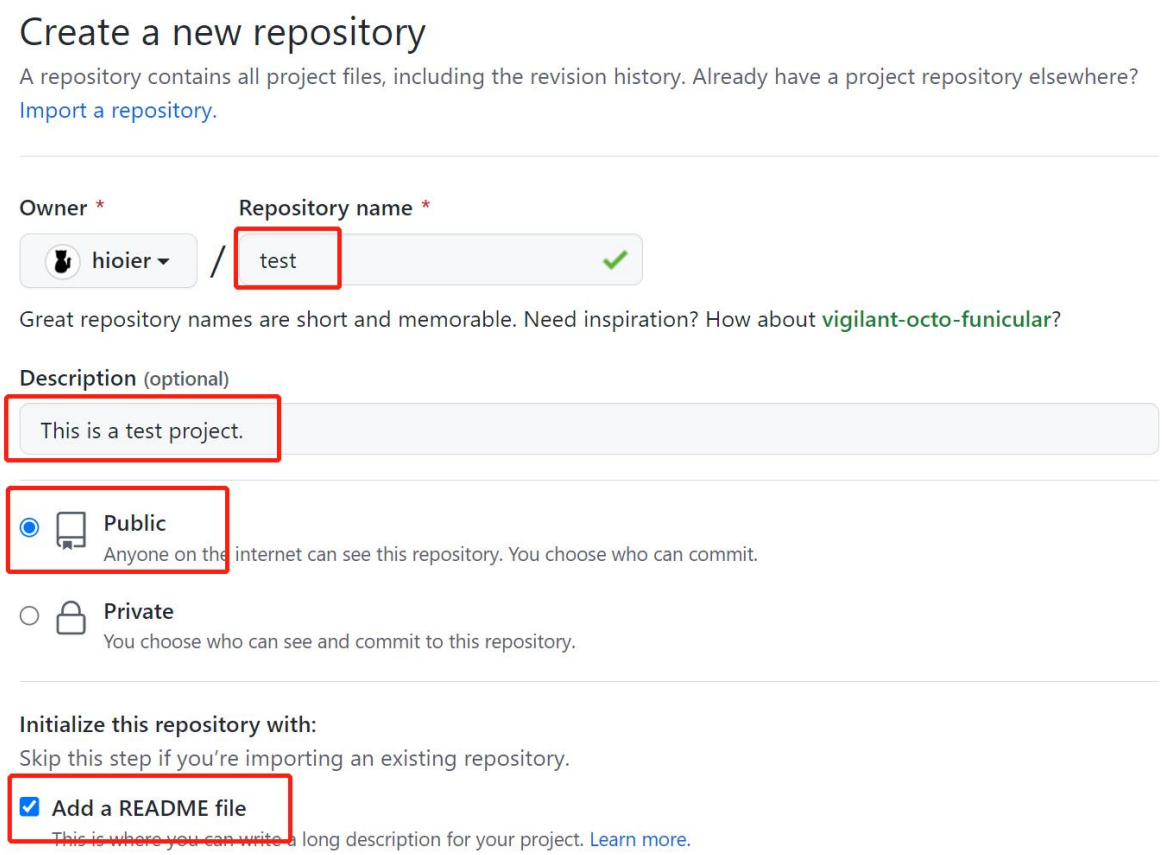
首先，要登陆github官网，注册账号并登录：



然后点击新建：



输入项目名称、描述等信息，Public是公开项目，创建私有项目要进行收费，READEME文件是项目简介：



第一次连接github需要本地服务器和github建立ssh连接，首先配置全局用户名和密码：

```
git config --global user.name "hioier"
git config --global user.email xypip@qq.com
git config --global --list # 查看所有配置
```

```
hioier@pc:~$ git config --global --list
user.name=hioier
user.email=xypip@qq.com
```

创建ssh：

```
ssh-keygen -t rsa -C "xypip@qq.com"
```

然后一直按回车就可以。

```
hioier@pc:~$ ssh-keygen -t rsa -C "xypip@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hioier/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hioier/.ssh/id_rsa
Your public key has been saved in /home/hioier/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:TOT4eSNq0OKqqaui00qTlfJsRrljxhOufUR1N/DEm3g xypip@qq.com
The key's randomart image is:
+---[RSA 3072]----+
|         ..o.    |
|       = .o+     |
|      o + o.+    |
|     oo + o E    |
| . *+ . S +      |
|  X.o+ . o .     |
| + &o o          |
|+.0.oo           |
|/*...            |
+----[SHA256]-----+
```

复制pub公钥：

```
hioier@pc:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDUBAiQEgn9vEez4HhoMTcjEjlX9HUxjH0Cu3/d/mU
BalDPbP0MrgGxhjD+9Vkayv+mUr6KyOr+5iSScdbC30daNrkQTqVMLtsAYiknux3ECpLY0l72NJ4lS
+St+YBVr2g73WeV7SEOSvQsrCcSJ/q27CinrjGWgdbc9bkRZGeejWrQc1sW5i4yL5XTp4CrJj14fslk
xFPXil9ja0xCgXnaG28bmz/W5yMVfQhaBymxeXWPb07k6PWhbEjX7UwqlC3xy0OwpoZl6QXqmnEAdd9
OLznsAqPIfGnRSvNS6TfWFKVxWU/6ZvBlvfCWig+gGCjs0sabWASco2M3mtimwpLtzmbDhCn2jgpSXJ
HQH7dNCJA+IkBSsma4AJtkfnM1mBX2Dwcw+h5t63pg7JqvcWrJn3pLTVr0ZJs/SmVls/rTA9Cw5Lmks
OiY8Mb8sKEPOEkSVRCr+BweWt6Dz4Pz1HLXLuY3c6LXlT6513xoRmOKGpSsCUv+n8ta7+66ty3Nns=
xypip@qq.com
```

或者通过pbcopy命令复制到剪切板：

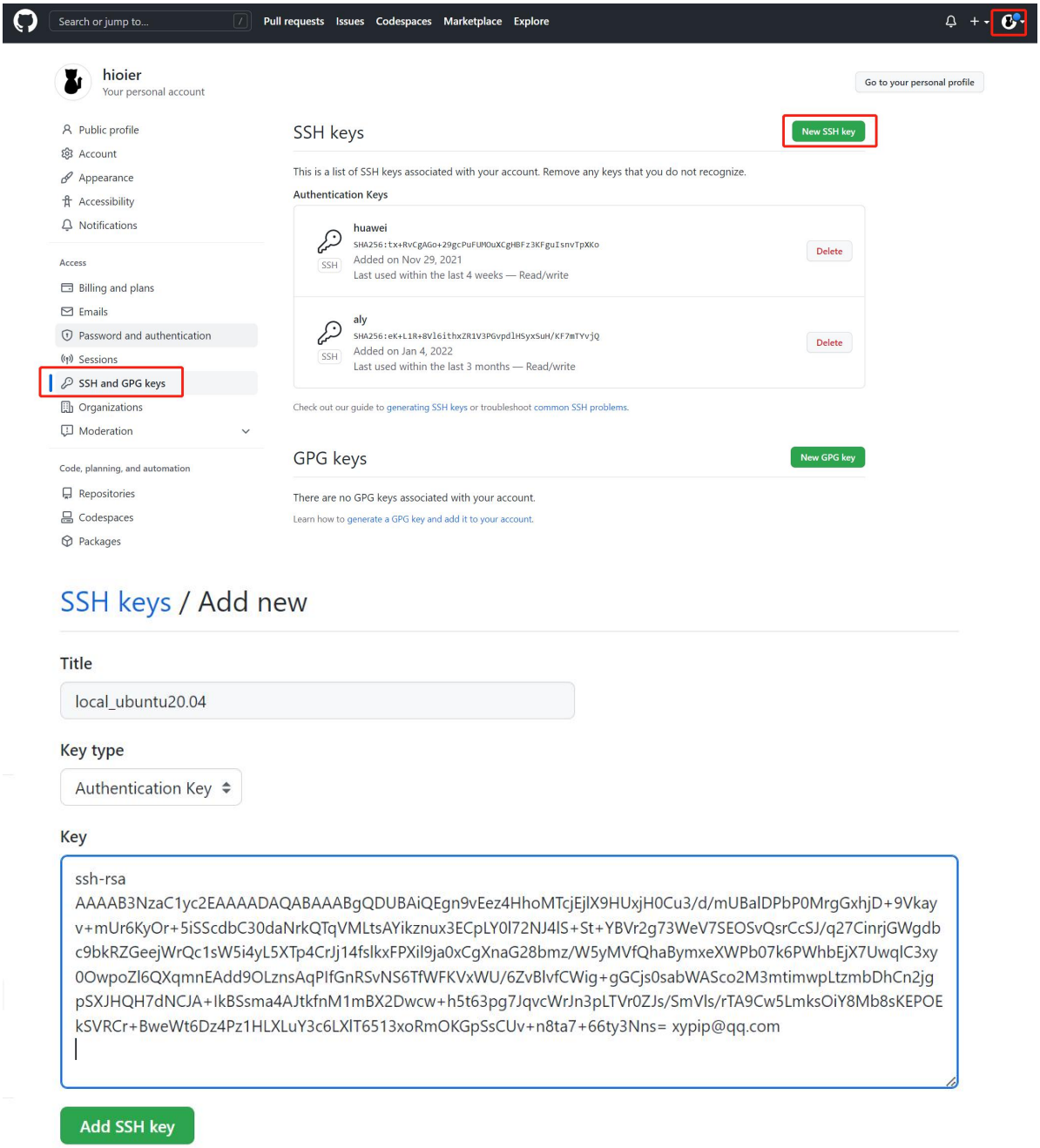
```
sudo apt install xclip

vim ~/.bashrc
# 添加内容
alias pbcopy='xclip -selection clipboard'
alias pbpaste='xclip -selection clipboard -o'

source ~/.bashrc

pbcopy < ~/.ssh/id_rsa.pub
```

然后进入GITHUB，点击右上角头像处选择设置：



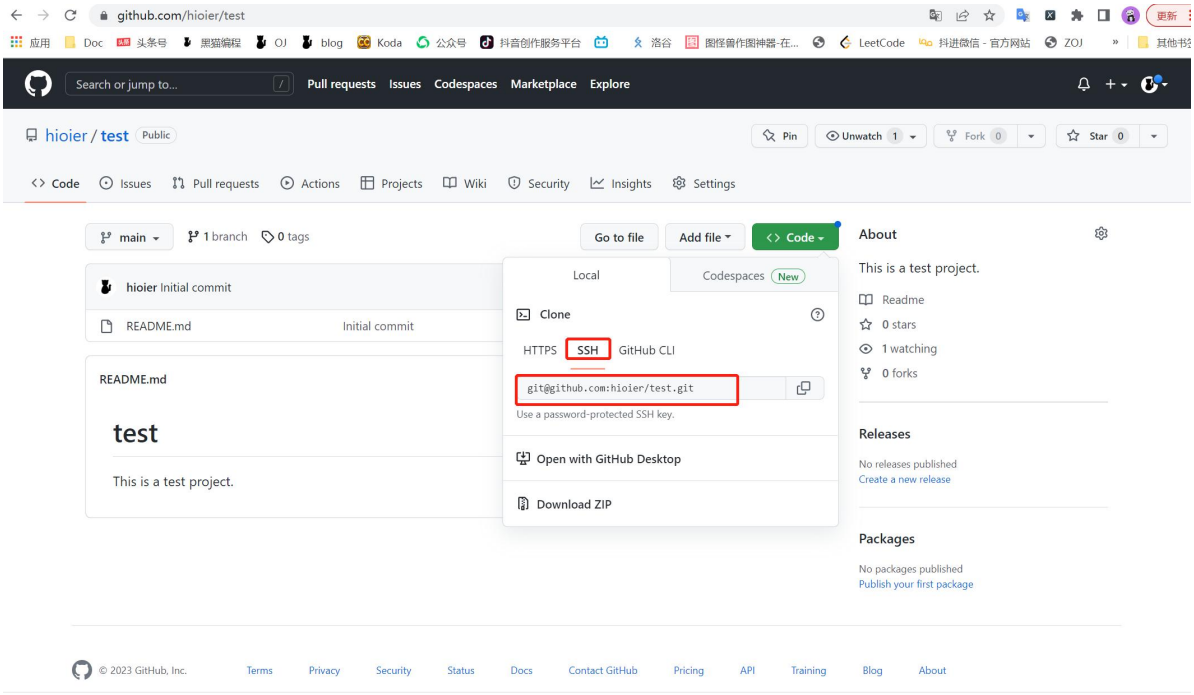
测试是否成功：

```
ssh -T git@github.com
```

```
hioier@pc:~$ ssh -T git@github.com
Hi hioier! You've successfully authenticated, but GitHub does not provide shell
 access.
```

克隆项目

克隆就是将github项目完整的下载到本地：

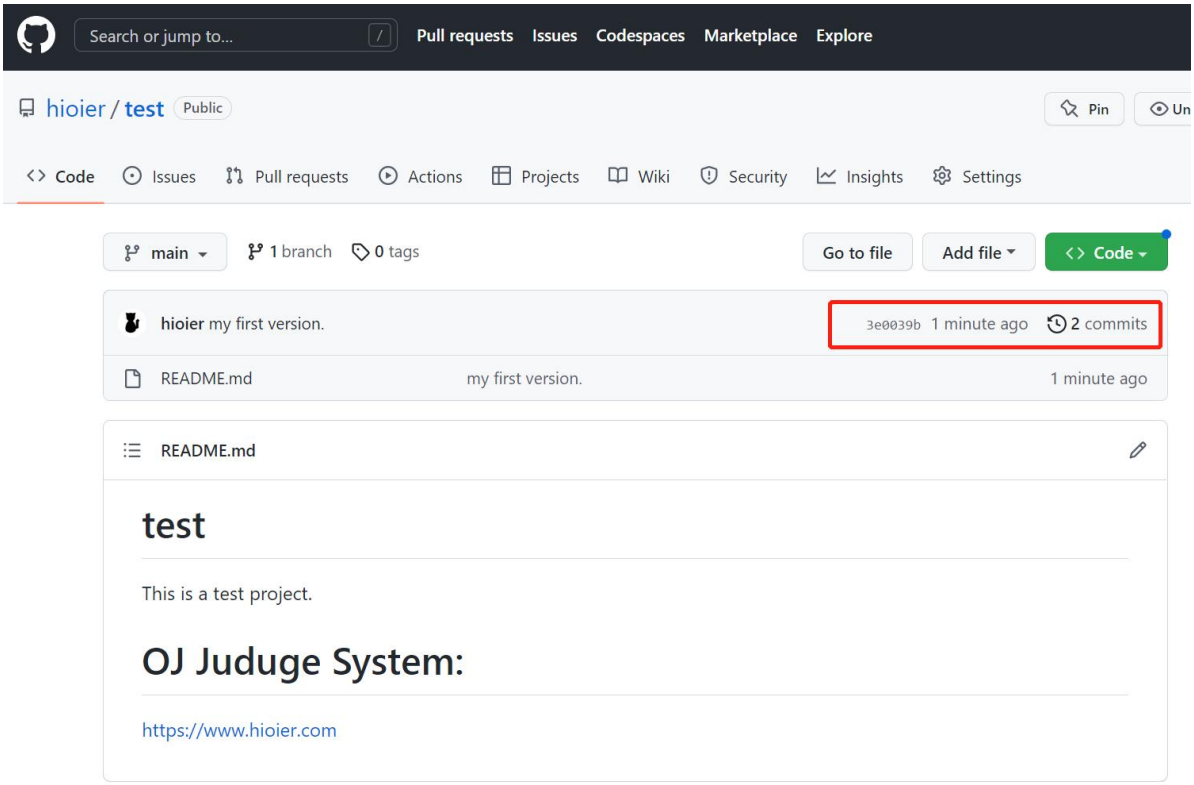



```
1 git clone git@github.com:hioier/test.git
```

```
hioier@pc:~$ git clone git@github.com:hioier/test.git
Cloning into 'test'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
hioier@pc:~$ ls
Desktop  Documents  Downloads  index.py  test
hioier@pc:~$ cd test/
hioier@pc:~/test$ ls
README.md
hioier@pc:~/test$ cat README.md
# test
This is a test project.
```

这样，一次本地服务器和github之间就已经连通啦，接下来简单实现git三部曲，修改一些READEME文件，然后上传至github服务器：

```
hioier@pc:~/test$ git add .
hioier@pc:~/test$ git commit -m "my first version."
[main 3e0039b] my first version.
 1 file changed, 3 insertions(+)
hioier@pc:~/test$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 103.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:hioier/test.git
   90de2dd..3e0039b  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```



进入github刷新页面，新添加的信息已经更新成功，这就是github最基础的三部曲：git add添加，git commit 备注，git push 提交程序。