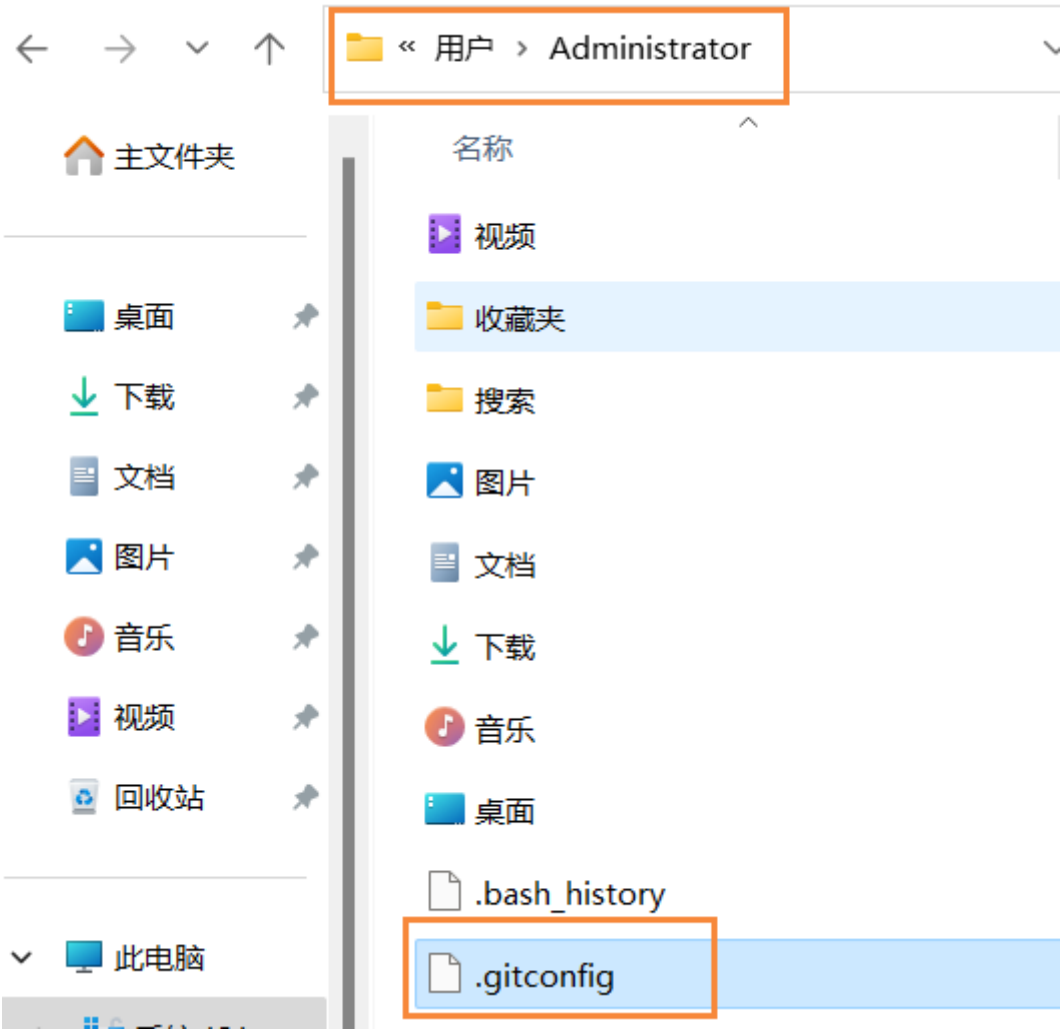
GitHub使用帮助

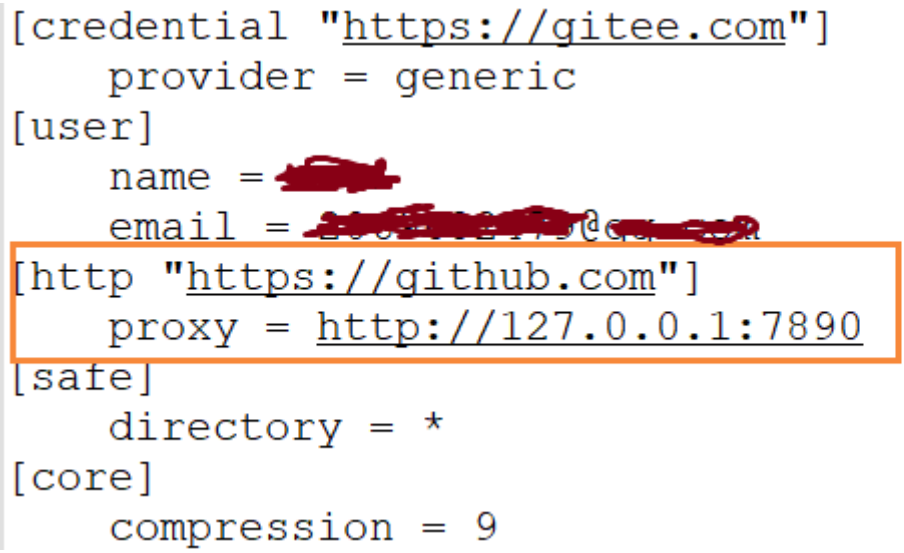
1 配置代理

由于网络原因，有的时候使用`https`的方式克隆、获取、拉取、推送代码会出现连接失败的问题，此时可以通过设置代理服务器的方式来解决问题。

使用代理的时候需要配置你的git，找到你的`.gitconfig`配置文件，一般在当前用户的根目录，如下图所示：



配置示例如下：




```
[http "https://github.com"]
    proxy = http://127.0.0.1:7890
```

将 `http://127.0.0.1:7890` 替换成你自己的代理服务器和端口。

补充一下

我们在`Windows CMD`里面使用命令行下载资源或安装程序的时候，也可能需要使用代理才能完成，可以通过下面步骤给控制台设置代理。

在执行命令前，首先设置代理，根据你的需要选择性的执行下面的命令：

```
# 设置HTTP代理
set http_proxy=protocol://proxyserveraddress:port
# 设置HTTPS代理
set https_proxy=protocol://proxyserveraddress:port
# 如果代理服务器需要用户名和密码进行验证，则需要在命令中添加用户名和密码信息
set http_proxy=protocol://username:password@proxyserveraddress:port
set https_proxy=protocol://username:password@proxyserveraddress:port
```

参数说明：

- `protocol` 是代理服务器协议，根据你代理服务器实际情况来了，通常是`http`或`https`
- `proxyserveraddress` 是代理服务器的地址
- `port` 是代理服务器的端口号
- `username` 和 `password` 分别是代理服务器的用户名和密码

2 配置SSH

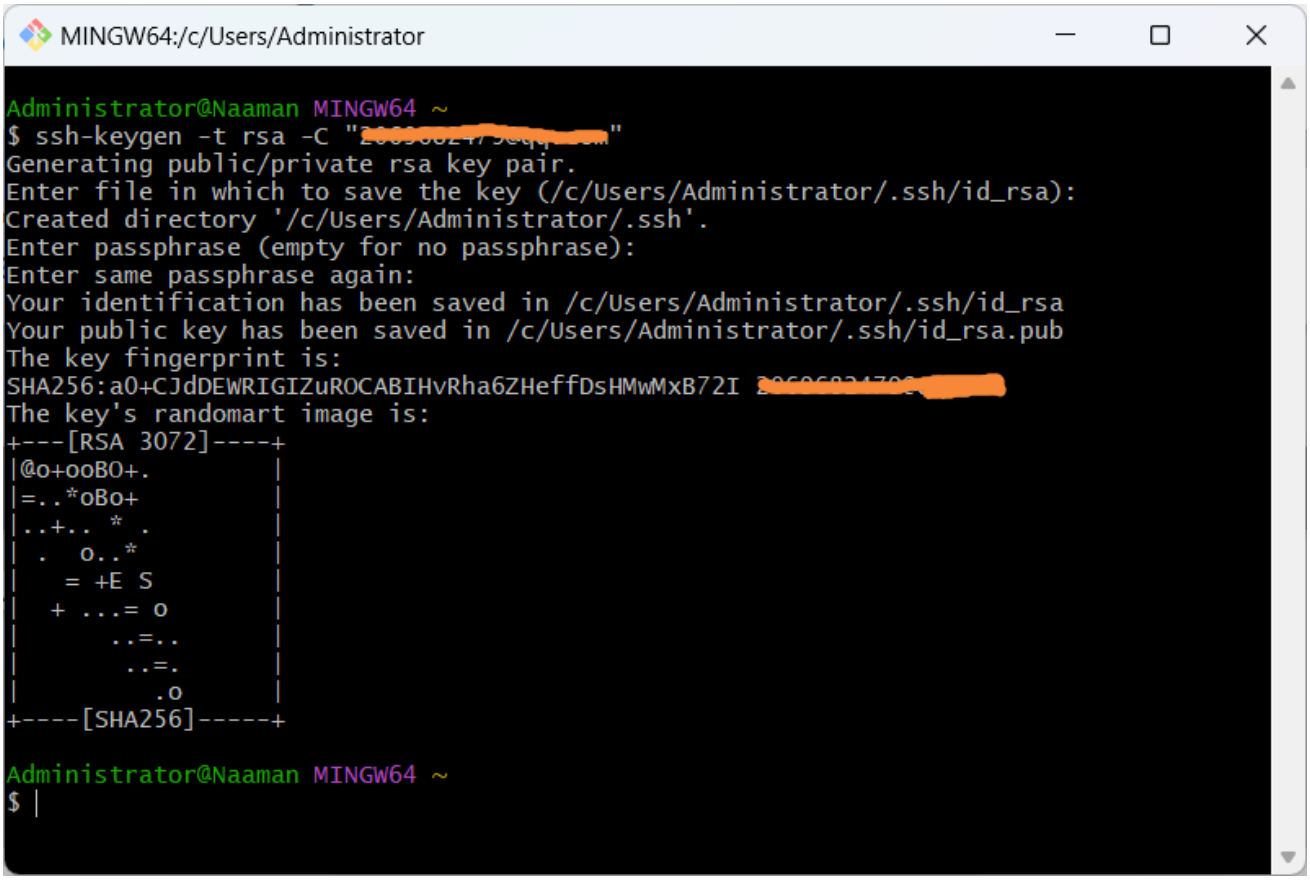
有时候通过SSH方式也能解决`GitHub`访问慢的问题。

2.1 生成秘钥对

打开Git bash工具，然后输入命令

```
ssh-keygen -t rsa -C "github邮箱"
```

遇到输入提示直接回车即可如下图所示



2.2 添加私钥到秘钥管理器

ssh-agent就是一个密钥管理器，运行ssh-agent以后，使用ssh-add将私钥交给ssh-agent保管，其他程序需要身份验证的时候可以将验证申请交给ssh-agent来完成整个认证过程。

确认一下秘钥管理器是否正常运行，执行`ssh-agent`命令，如下图所示。

```
Administrator@Naaman MINGW64 ~
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-37i9fhpQZAKY/agent.1972; export SSH_AUTH_SOCK;
SSH_AGENT_PID=1973; export SSH_AGENT_PID;
echo Agent pid 1973;
```

添加私钥到秘钥管理器，执行下列命令

```
ssh-add ~/.ssh/id_rsa
```

执行结果如下图所示，表示出现了错误

```
Administrator@Naaman MINGW64 ~
$ ssh-add ~/.ssh/id_rsa
Could not open a connection to your authentication agent.
```

通过下面的方式解决

- 首先查询进程

```
ps aux | grep ssh
```

- 然后杀死进程

```
kill -9 进程号
```

- 进入用户名目录下的.ssh目录，执行如下命令

```
cd ~/.ssh
exec ssh-agent bash
eval ssh-agent -s
```

```
Administrator@Naaman MINGW64 ~
$ ps aux | grep ssh
    1973      1    1973      13784  ?         197108 15:48:39 /usr/bin/ssh-agent

Administrator@Naaman MINGW64 ~
$ kill -9 1973

Administrator@Naaman MINGW64 ~
$ cd ~/.ssh

Administrator@Naaman MINGW64 ~/.ssh
$ exec ssh-agent bash
Administrator@Naaman MINGW64 ~/.ssh
$ eval ssh-agent -s
SSH_AUTH_SOCK=/tmp/ssh-K7mYXG14ZAJY/agent.2010; export SSH_AUTH_SOCK;
SSH_AGENT_PID=2011; export SSH_AGENT_PID;
echo Agent pid 2011;
```

- 再次执行添加命令

```
ssh-add ~/.ssh/id_rsa
```

执行结果如下图所示表示添加成功

```
Administrator@Naaman MINGW64 ~/.ssh
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/Administrator/.ssh/id_rsa (
```

2.3 配置GitHub

查看公钥

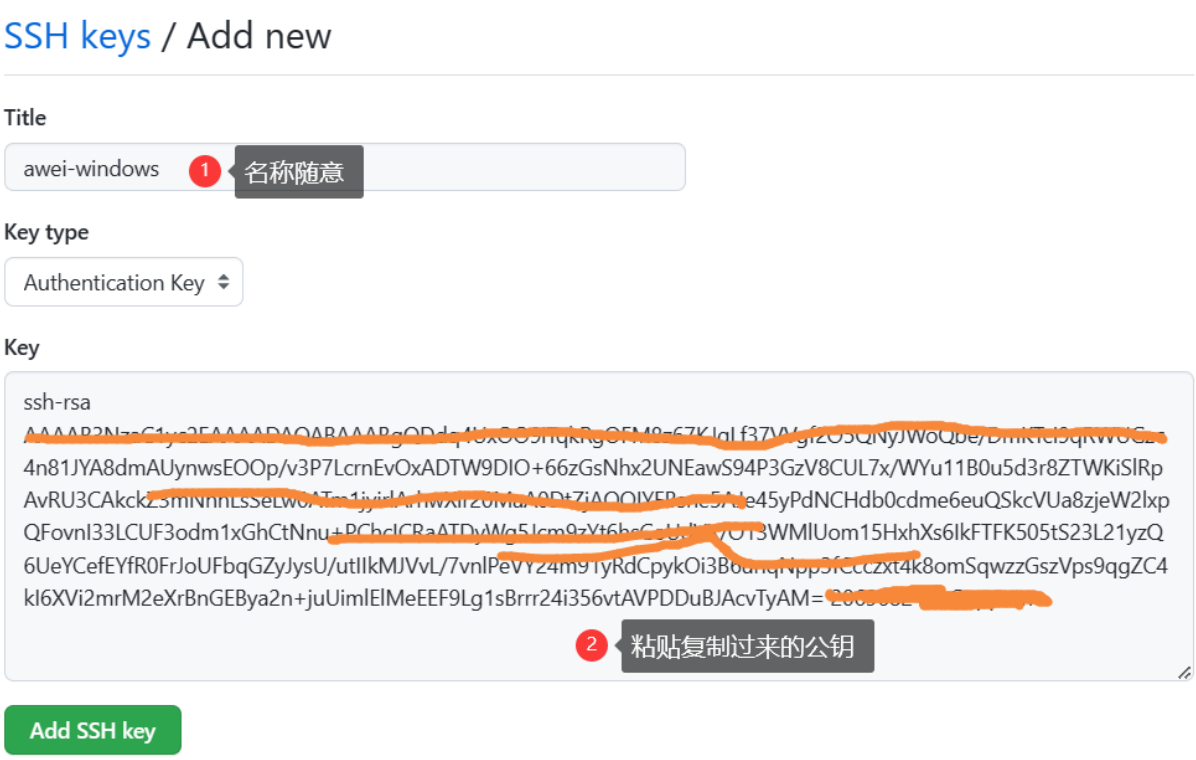
```
cat ~/.ssh/id_rsa.pub
```

```
Administrator@Naaman MINGW64 ~/.ssh
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDdq4UxOO9iTqkRgOFM8z67KJgLf37VVgf2O5QNyJWoC
KTci9qRWUGzc4n81JYA8dmAUynwsEOOp/v3P7LcrnEvOxADTW9DIO+66zGsNhx2UNEawS94P3GzV8CUL7
11BOu5d3r8ZTWKiSlRpAvRU3CAkckZ3mNnnLsSeLwOATm1jyirlArhwXir2OMaAODtZjAQOIYFBsne5Ai
dNCHdbOcdme6euQSkcVUa8zjeW2lxpQFovnI33LCUF3odm1xGhCtNnu+PChcICRaATDyWg5Jcm9zYt6hs
X/O13WMlUom15HxhXs6IkFTFK5O5tS23L21yzQ6UeYCefEYfROFrJoUFbqGZyJysU/utIIkMJVvL/7vnl
4m91yRdCpykOi3B6unqNpp3fCcczxt4k8omSqwzzGszVps9qgZC4kI6XVi2mrM2eXrBnGEBya2n+juUir
```

复制公钥备用。

打开GitHub SSH Key新建链接：https://github.com/settings/ssh/new

然后在执行下图所示的操作

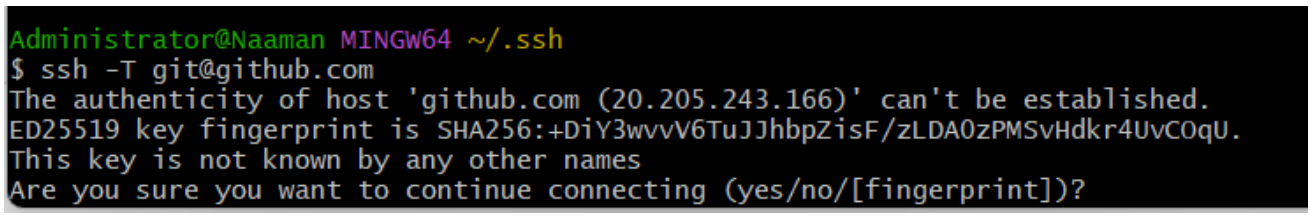


2.4 连接测试

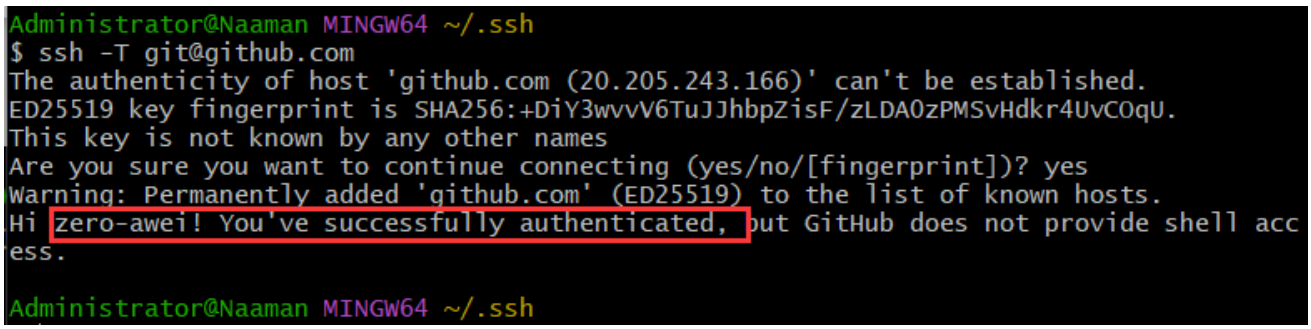
执行命令

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

执行结果如如下图所示



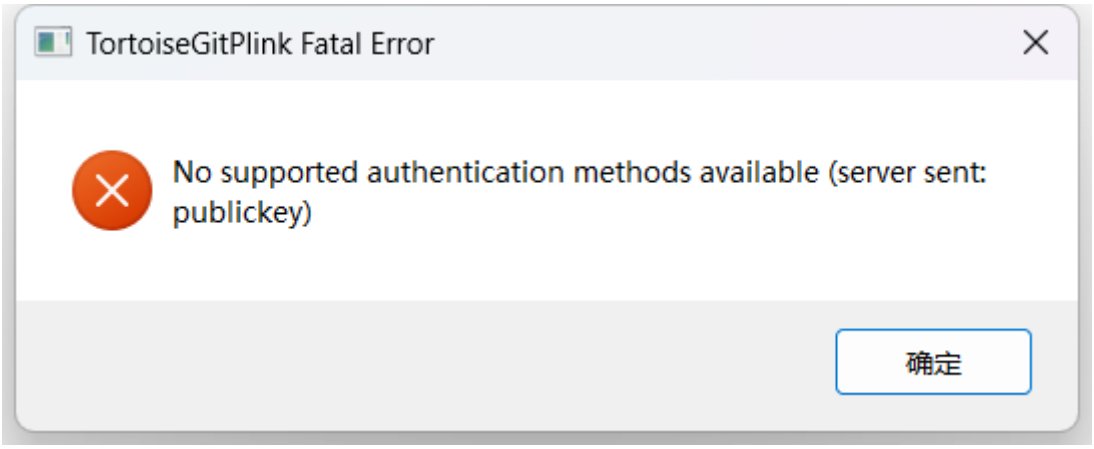
输入yes，执行结果如下图所示



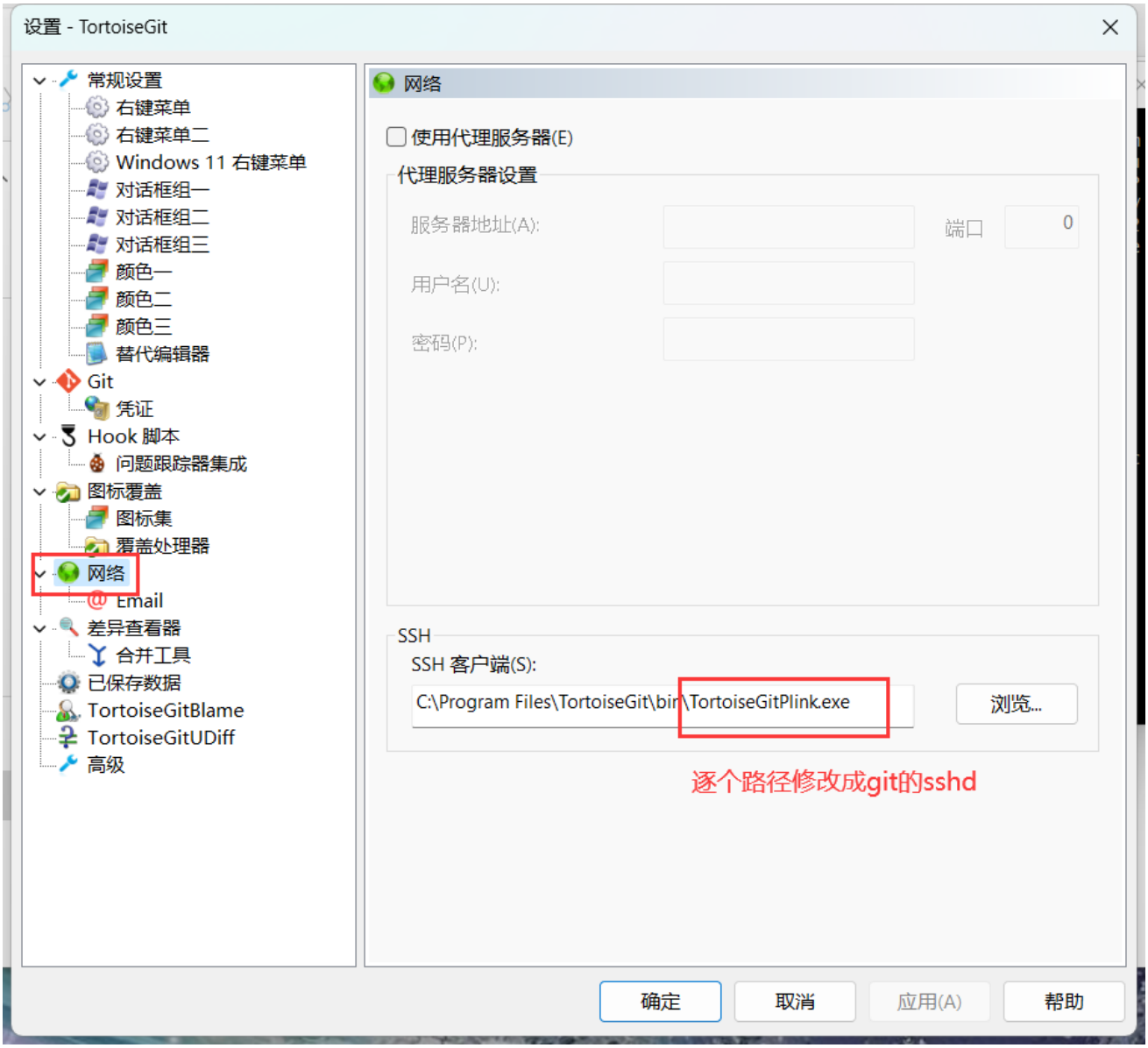
到此说明配置成功了，你就可以用ssh方式克隆和推送代码到`GitHub`了

提示：

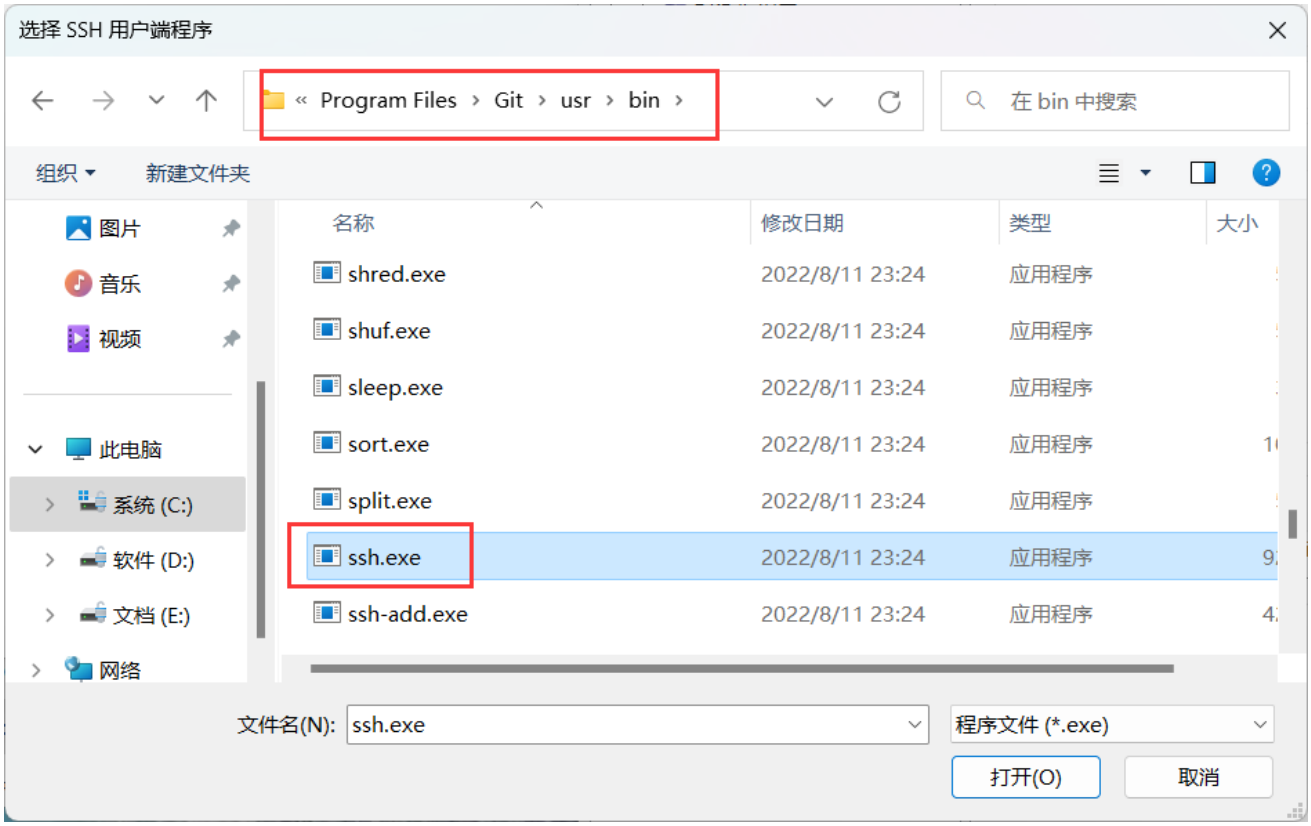
如果我们用乌龟clone项目的时候出现下图的错误提示，我们需要做以下处理。



我们需要修改`TortoiseGit`网络的SSH客户端为 git 服务器的`ssh.exe`



点击浏览，示例路径如下



再次尝试即可正常clone

2.5 扩展

如果需要配置多个凭证，可以在.ssh目录下面建立一个`config`文件书写配置来对应

下面是一个配置示例：

```
#github
Host github.com
HostName github.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/id_rsa
User zero-awei
```

配置文件说明：

每个账号单独配置一个`Host`，`HostName`和`IdentityFile`三个属性即可。

- `Host`：别名，可以取为自己喜欢的名字，不过这个会影响git相关命令，例如：
  - `Host mygithub`这样定义的话，命令如下，即`git@`后面紧跟的名字改为`mygithub`，相应的`clone`命令为：
  - ```
    git clone git@mygithub:xxxx/xxxxx.git
    ```
- `HostName`：映射`Host`对应的真正的域名，如：`github.com`
- `IdentityFile`：配置私钥文件所在位置，如：`~/.ssh/id_rsa`
- `PreferredAuthentications`：配置登录时用什么权限认证
  - 可设置为：`publickey,password publickey,keyboard-interactive`等
    ```

- `User`：配置使用用户名

3 个人访问令牌

有时候我们在使用 `https` + 用户名密码验证的方式访问私库时，你可能会出现下面的错误：

```
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.
```

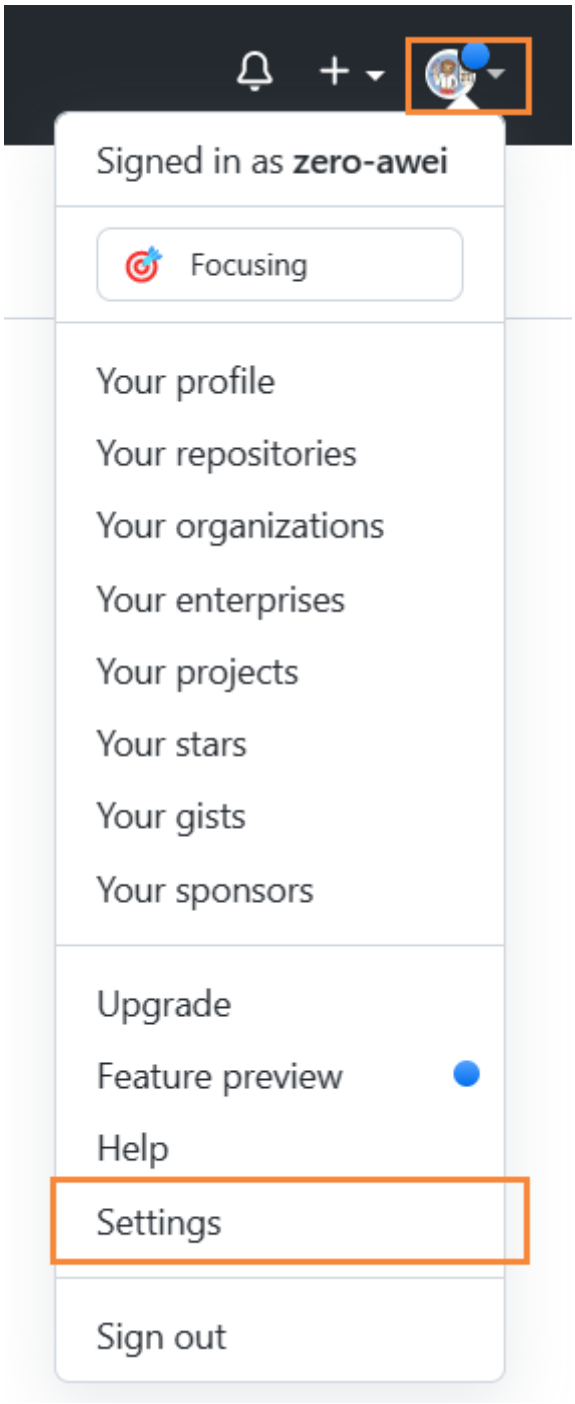
出现这个错误的原因是，2021年8月13日以后，不再支持用户名密码的方式验证了，需要使用个人访问令牌（personal access token），也就是把你的密码需要替换成token，当然你可以不使用token，使用ssh方式操作仓库也是可行的。

参考链接：https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/

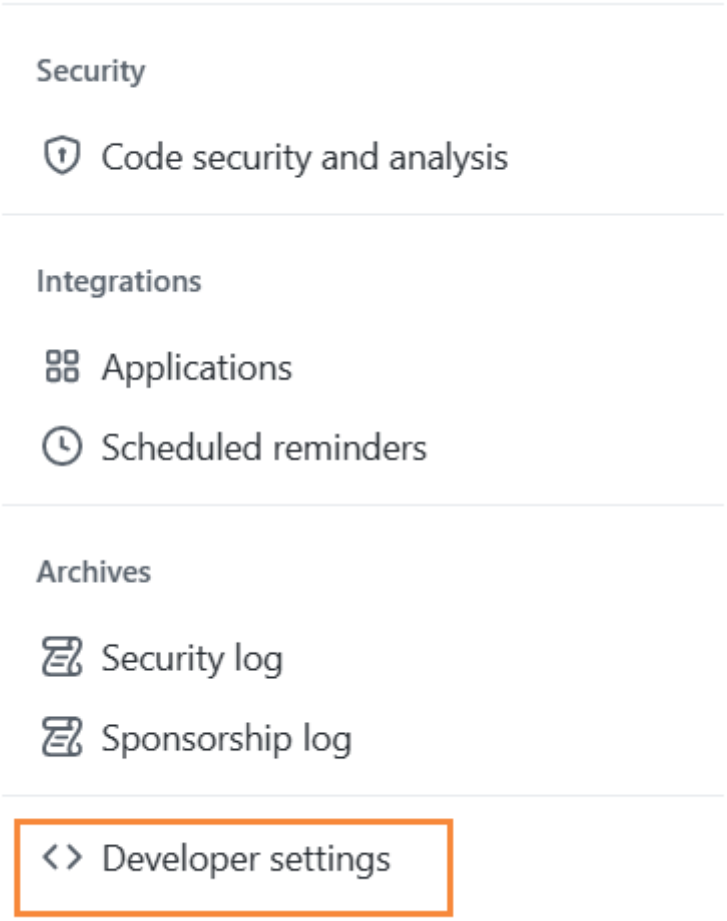
下面来看一看生成和使用令牌的步骤。

3.1 生成令牌

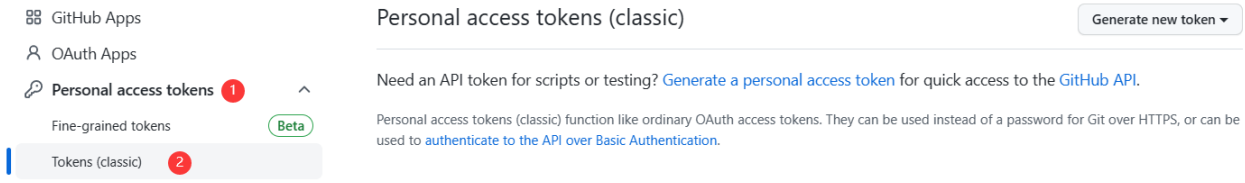
首先进入个人设置页面


Signed in as zero-awei
Focusing
Your profile
Your repositories
Your organizations
Your enterprises
Your projects
Your stars
Your gists
Your sponsors
Upgrade
Feature preview
Help
Settings
Sign out

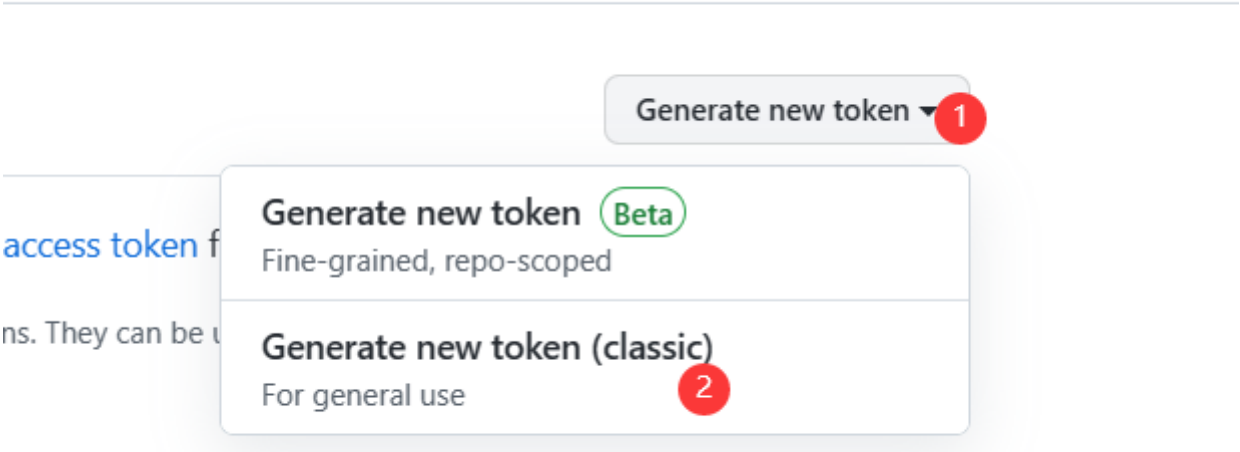
找到并打开开发设置



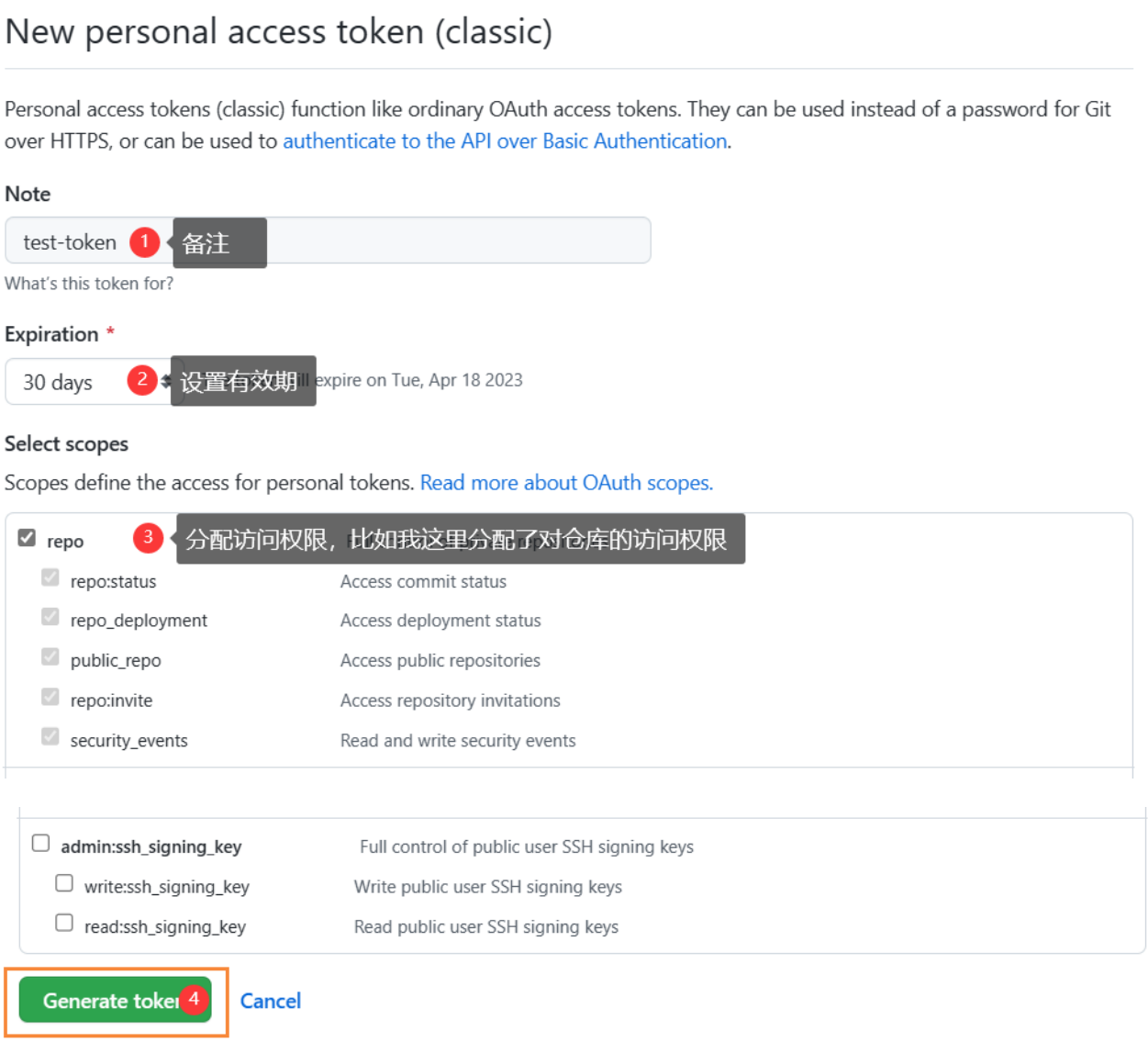
找到个人访问令牌



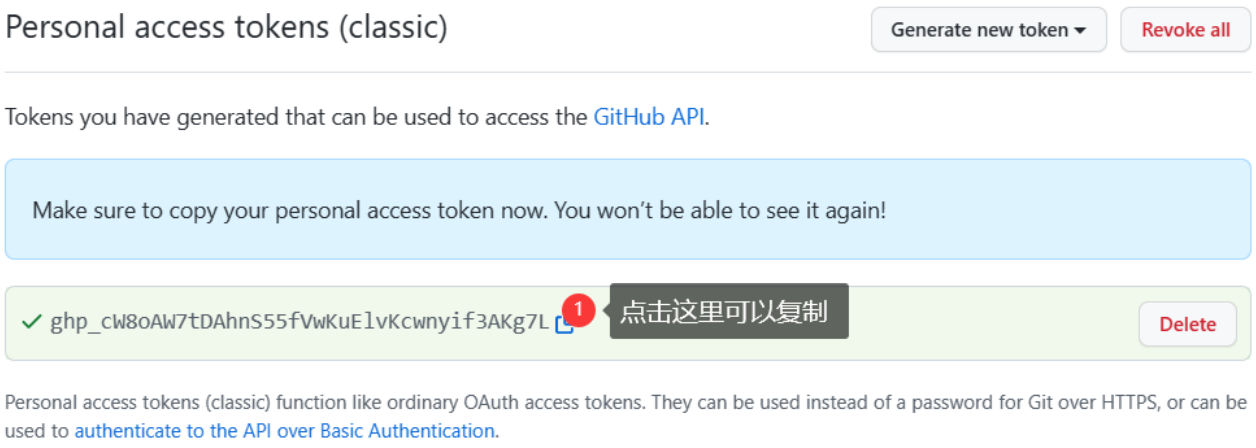
点击生成令牌



输入相关参数并生成令牌



如下是生成的令牌，记得保存，下次就看不到了。



3.2 使用令牌

用法一：直接将token作为密码使用

在需要输入密码的地方，换成token即可。

用法二：修改现有项目的`url`

```
git remote set-url origin https://<TOKEN>@github.com/<USERNAME>/<REPO>.git
```

- `<TOKEN>`：换成你刚生成的令牌
- `<USERNAME>`：换成你 `GitHub` 的用户名
- `<REPO>`：换成你对应的仓库名称

用法三：在克隆项目的时候在`github.com`前面加个令牌

```
git clone https://<TOKEN>@github.com/<USERNAME>/<REPO>.git
```

后面两种方式都可以通过修改`.git`目录下面的`config`文件实现，只需要要找到`remote`配置，然后在`url`链接值里面加上令牌即可

```
[remote "origin"]
    url = https://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@github.com/zero-awei/test.git
    fetch = +refs/heads/*:refs/remotes/origin/*
```