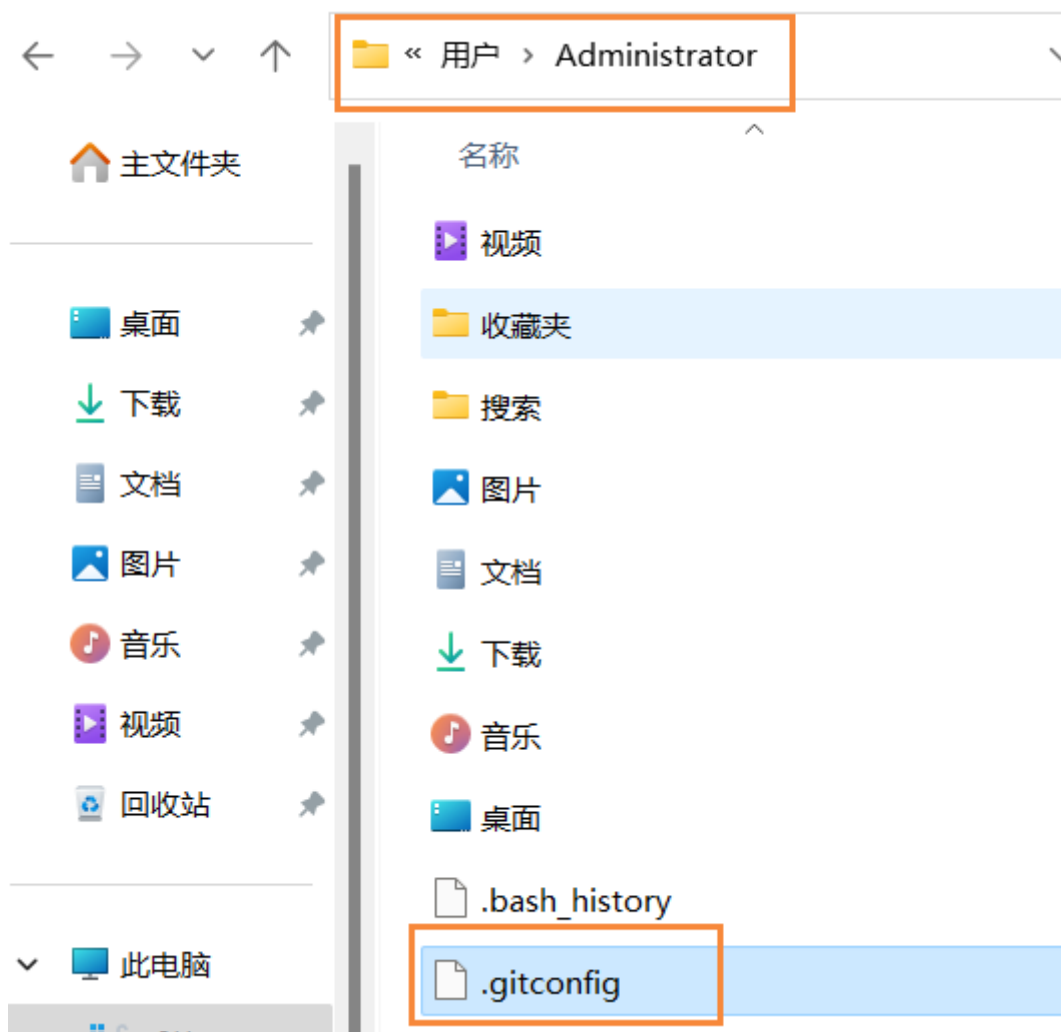


GitHub使用帮助

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。

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：

```
[credential "https://gitee.com"]
  provider = generic
[user]
  name = xxxxxx
  email = xxxxxx@xxx.com
[http "https://github.com"]
  proxy = http://127.0.0.1:7890
[safe]
  directory = *
[core]
  compression = 9
```

```
1 [http "https://github.com"]
2   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

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

```
1 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

```
MINGW64:/c/Users/Administrator

Administrator@Naaman MINGW64 ~
$ ssh-keygen -t rsa -C "2069682479@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Administrator/.ssh/id_rsa):
Created directory '/c/Users/Administrator/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Administrator/.ssh/id_rsa
Your public key has been saved in /c/Users/Administrator/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:a0+CJdDEWRIGIZuROCABIHvRha6ZHeffDshMwMxB72I 2069682479
The key's randomart image is:
+---[RSA 3072]-----+
|@o+ooBO+.
|=..*oBo+
|..+..* .
| . o..*
|  = +E S
| + ...= o
|      ..=.
|      ..=.
|      .o
+----[SHA256]-----+

Administrator@Naaman MINGW64 ~
$ |
```

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

```
1 | 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

```
1 | ps aux | grep ssh
```

- 然后杀死进程

```
1 | kill -9 进程号
```

- 进入用户名目录下的.ssh目录，执行如下命令

```
1 cd ~/.ssh
2 exec ssh-agent bash
3 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

```
1 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 (200900247-1)
```

2.3 配置GitHub

[查看公钥](#)

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

```
Administrator@Naaman MINGW64 ~/.ssh
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDd4Ux009iTqkRgOFM8z67KJgLf37Vvgf205QNYJWo
KTci9qRWUGzc4n81JYA8dmAUynwsE00p/v3P7LcrnEvOxADTW9DIO+66zGsNhX2UNEawS94P3GzV8CUL7
11B0u5d3r8ZTWKis1RpAvRU3CAkckZ3mNnnLsSeLw0ATm1jyir1ArhwXir20MaA0DtZjAQ0IYFBsne5A3
dNCHdb0cdme6euQSkcVUa8zjew21xpQFovni33LCUF3odm1xGhCtNnu+PChcICRAATDyWg5Jcm9zYt6hs
X/013wM1Uom15HxhXs6IkFTFK505tS23L21yzQ6UeYCefEYfR0FrJoUfBqGZyJysU/utIIkMJVvL/7vn1
4m91yRdCpyk0i3B6unqNpp3fCcczxt4k8omSqwzzGszVps9ggZC4kI6XVi2mrM2eXrBnGEBya2n+juUi
```

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

SSH keys / Add new

Title

aweij-windows

1

名称随意

Key type

Authentication Key

Key

ssh-rsa

4n81JYA8dmAUynwsEOOp/v3P7LcnnEvOxADTW9DIO+66zGsNhx2UNEawS94P3GzV8CUL7x/WYU11B0u5d3r8ZTWKISIRp
AvRU3CAkckz3mnnnnlsselwAtTz3jyid4rhwxR20Ma40D73AQOIVF3m5A7e45yPdNCHdb0cdme6euQSkcVUa8zjeW2lxp
QFovnl33LCUF3odm1xGhCtNnu+PChdCBAATD4AmQ5lcm9zYt6hcCUL7v013WMIUom15HxhXs6lkFTFK505T523L21yzQ
6UeYcefEYfR0fJoUFbqGZyJysU/uttlkMJVvL/7vnlPeVyz4m91yRdCpykOi3B0uqnp5fcccxt4k8omSqwwzGszVps9qgZC4
kl6XVi2mrM2eXrBnGEBya2n+juUimlElMeEFF9Lq1sBrrr24i356vtAVPDDuBJAcvTvAM=

2

粘贴复制过来的公钥

Add SSH key

2.4 连接测试

执行命令

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

执行结果如如下图所示

```
Administrator@Naaman MINGW64 ~/.ssh
$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCQQU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

输入yes, 执行结果如下图所示

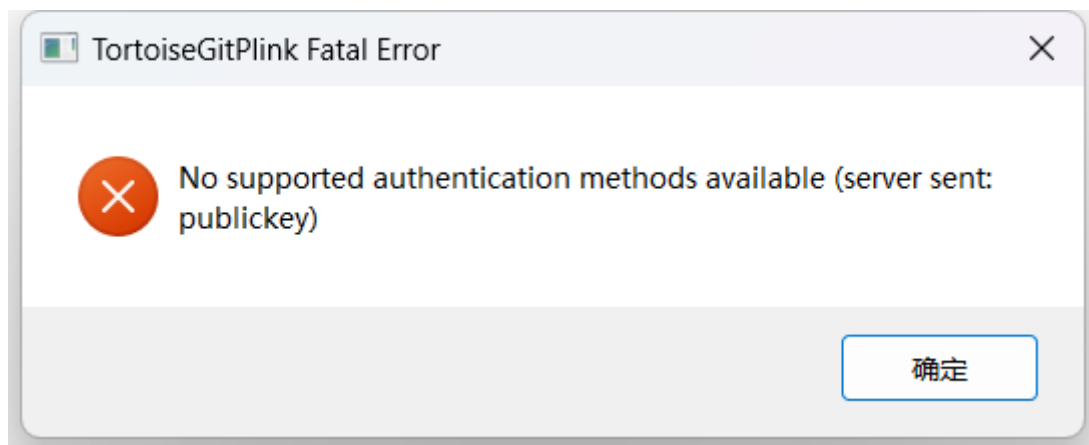
```
Administrator@Naaman MINGW64 ~/.ssh
$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhpbZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi zero-awei! You've successfully authenticated, but GitHub does not provide shell access.

Administrator@Naaman MINGW64 ~/.ssh
```

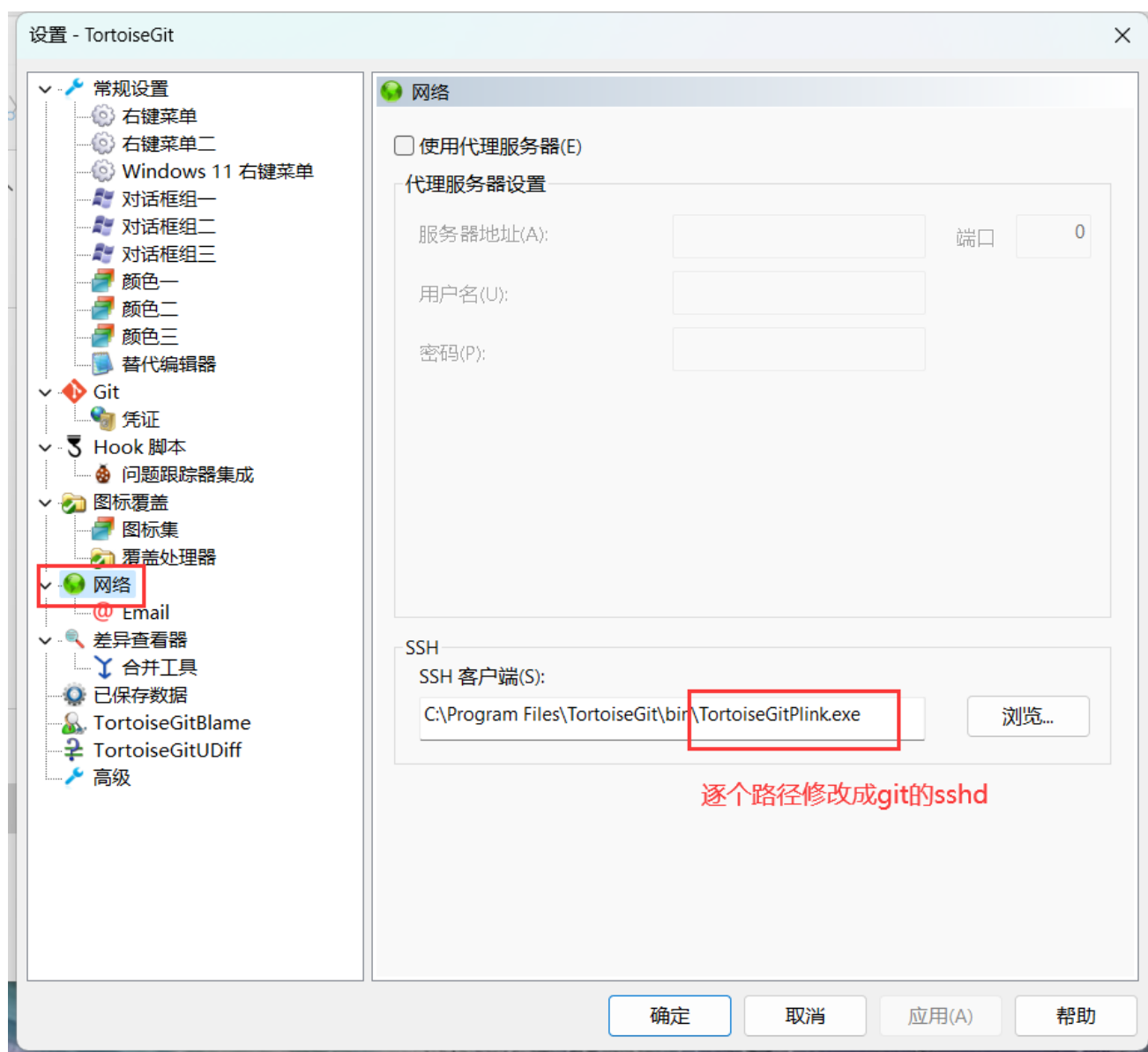
到此说明配置成功了，你就可以用ssh方式克隆和推送代码到GitHub了

提示：

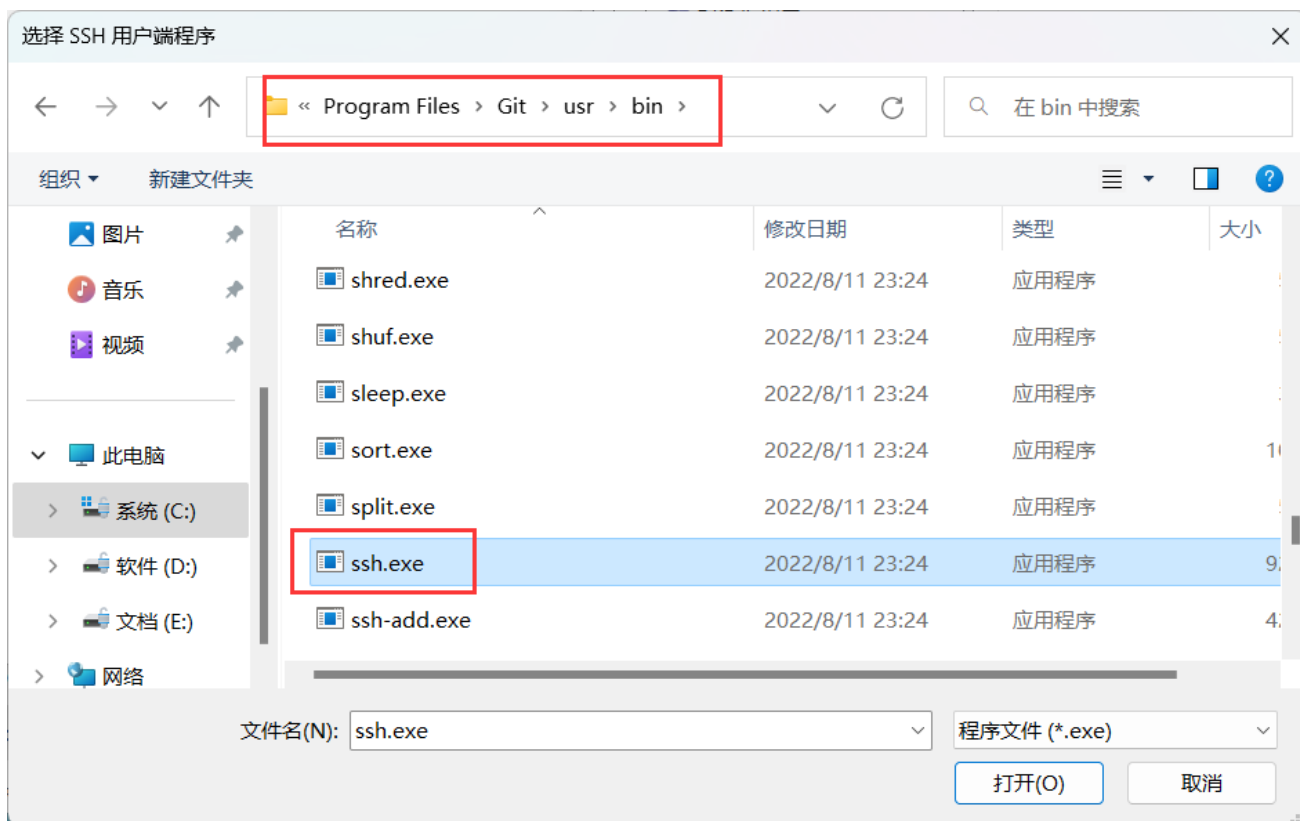
如果我们用乌龟clone项目的时候出现下图的错误提示，我们需要做以下处理。



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：

```
1 #github
2 Host github.com
3 HostName github.com
4 PreferredAuthentications publickey
5 IdentityFile ~/.ssh/id_rsa
6 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：

```
1 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