1 关于 ssh

SSH 是较可靠，专为远程登录会话和其他网络服务提供安全性的协议。[[2]](https://copyfuture.com/blogs-details/202204171243051424#fn2)使用 SSH 协议可以连接远程服务器和服务并向它们验证。连接远程仓库时无需输入密码而且能实现对 github 的流畅访问。除此之外我还用 ssh 协议成功在 Windows Terminal 中连接上了自己的远程服务器。

## 2 检查现有的 ssh 密钥

Windows 打开 GitBash，Linux 下打开终端，输入：

$ ls -al ~/.ssh

如果你看到以下输出，那么说明你已经有 ssh 密钥（以 .pub 结尾的文件）：

$ ls -al ~/.ssh

total 26

drwxr-xr-x 1 197121 0 Mar 31 22:29 ./

drwxr-xr-x 1 197121 0 Apr 16 23:36 ../

-rw-r--r-- 1 197121 2610 Mar 31 22:27 id\_rsa

-rw-r--r-- 1 197121 573 Mar 31 22:27 id\_rsa.pub

-rw-r--r-- 1 197121 831 Apr 1 12:08 known\_hosts

-rw-r--r-- 1 197121 92 Mar 31 22:26 known\_hosts.old

如果你不想用原来的密钥或者没有密钥的话也不用着急，看下一步如何生成新的 ssh 密钥。

## 3 创建新的 ssh 密钥

### 3.1 生成新的 ssh 密钥

在 Windows 的 GitBash 或 Linux 终端输入：

$ ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

# 把引号里的替换为自己的邮箱

然后终端会让你输入一些东西，可以不用输入直接一路回车，看到：

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_rsa

Your public key has been saved in /root/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:t3925EJgD4kcUp69dBv/nqkM85n0I5mFRW4BlZwaVeo your\_email@example.com

The key's randomart image is:

+---[RSA 4096]----+

| .=o=|

| . . B |

| o + \* .|

| . \* =oo+ |

| So.B +A+ |

| .o.B.o.o|

| .o o+o.|

| .\*+B.\*|

| .OoBo|

+----[SHA256]-----+

代表密钥生成成功。

### 3.2 将SSH密钥添加到ssh-agent

#### 3.2.1 确保 ssh 正在运行

终端输入：

$ eval

$ ssh-agent -s

看到如下输出：

SSH\_AUTH\_SOCK=/tmp/ssh-xkZRdf3bu6fW/agent.1075848; export SSH\_AUTH\_SOCK;

SSH\_AGENT\_PID=1075849; export SSH\_AGENT\_PID;

echo Agent pid 1075849;

代表 ssh 正常运行。

#### 3.3.3 添加 ssh 到账户

执行以下两条命令：

$ ssh-agent bash

$ ssh-add ~/.ssh/id\_rsa # 这里如果文件名被改过要写你自己定义的文件名

看到输出：

Identity added: id\_rsa (your\_email@example.com)

添加成功！

## 4 将密钥添加到 github 账户

### 4.1 复制你的公钥

在终端使用：

$ cat ~/.ssh/id\_rsa.pub

看到：

ssh-rsa  your\_email@example.com

全给复制下来！

### 4.2 将公钥添加到 github 账户

然后去 gihub：

* 右上角下拉面板选择 **Settings**；
* 左侧选择 **SSH and GPG keys**；
* 点击 **New SSH key**；
* 随便起一个 title；
* 把公钥粘贴到下面。

可能会输入密码，添加完成！

## 5 测试 ssh 连接

在终端中输入：

$ ssh -T git@github.com

可能看到如下警告：

The authenticity of host 'github.com (20.205.243.166)' can't be established.

ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM.

Are you sure you want to continue connecting (yes/no/[fingerprint])?

输入 yes：

Warning: Permanently added 'github.com,20.205.243.166' (ECDSA) to the list of known hosts.

Hi xxxxxx You've successfully authenticated, but GitHub does not provide shell access.

如果 xxxxxx 是你的 github 用户名，说明成功。

备注：ssh -T [git@github.com](mailto:git@github.com) 后如果输出如下信息，表示域名对应关系没有添加

ssh: Could not resolve hostname github.com: Name or service not known

可如下操作：

$ ping github.com

Pinging github.com [20.205.243.166] with 32 bytes of data:

$ ssh -T git@20.205.243.166

The authenticity of host '20.205.243.166 (20.205.243.166)' can't be established.

ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/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 '20.205.243.166' (ED25519) to the list of known hosts.

Hi remoWorkSpace! You've successfully authenticated, but GitHub does not provide shell access.

## 6 配置 git 使用 ssh 密钥登录

首先将你的仓库 clone 下来到一个位置。然后进入你的仓库。

### 6.1 检查当前的 remote url

在仓库下输入：

$ git remote -v

看到输出：

origin https://xxxxx.git (fetch)

origin https://xxxxx.git (push)

说明当前使用的还是 https 协议，如果以 git 开头表示 git 协议。

### 6.2 修改 remote url 为 git 协议

上 github 仓库，点 **Code**，选择 **SSH**，复制链接：

image-20220417121125160

在终端输入：

$ git remote set-url origin git@github.com:<github账户名>/<github版本库名>.git

再检查 git 协议：

$ git remote -v

出现：

origin git@xxxxx.git (fetch)

origin git@xxxxx.git (push)

成功！

接下来可以快乐 push&pull 了。

## 7 向远程GitHub推送代码

第5步已经连接了GitHub，此处直接推送

$ git push origin master

输出

Enumerating objects: 608, done.

Counting objects: 100% (608/608), done.

Delta compression using up to 8 threads

Compressing objects: 100% (597/597), done.

Writing objects: 100% (608/608), 45.73 MiB | 1.79 MiB/s, done.

Total 608 (delta 81), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (81/81), done.

To 20.205.243.166:remoWorkSpace/QtProj.git

\* [new branch] master -> master

## 8 远程版本库提交代码

git remote -v可以看到远端连接的名称代号，如下为 origin：

$ git remote -v

origin https://github.com/remoWorkSpace/QtProj.git (fetch)

origin https://github.com/remoWorkSpace/QtProj.git (push)

向origin这个远端版本库地址推送 master分支的内容变化：

$ git push origin master

如果git remote -v下看不到远端版本库，则需要自己添加

如下添加代号为origin的远端版本库：

git remote add origin <server>

然后再执行git push origin master 提交代码