

证书与 key

<https://www.cnblogs.com/einstein-2014731/p/11492637.html>{
[Gitee 使用方法](#)

原文链接: <https://www.cnblogs.com/yiven/p/8465054.html>

1.创建仓库:

a.创建远程仓库

登入 Gitee 后, 点击头像旁边的"+"加号-->新建项目



填写项目基本信息后直接点击创建, 这样一个远程仓库就已经创建完成了

创建项目

归属

名称

提示：您还能免费创建 997 个项目

路径

介绍
非必填

用简短的语言来描述一下这个项目吧

选择语言



添加.gitignore



添加开源许可证



是否公开



公开




私有




使用Readme文件初始化这个项目



使用Issue模板文件初始化这个项目 



使用Pull Request模板文件初始化这个项目 

 导入已有项目

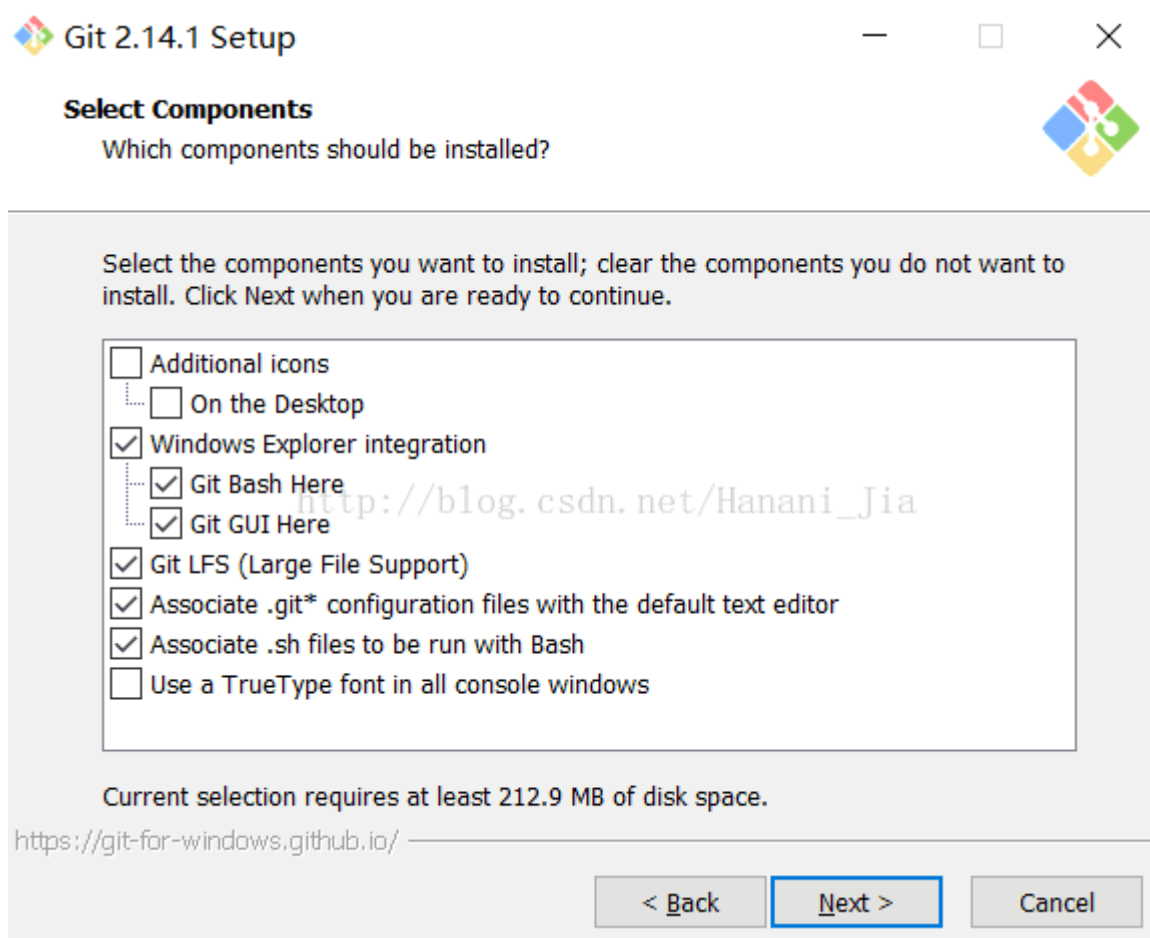
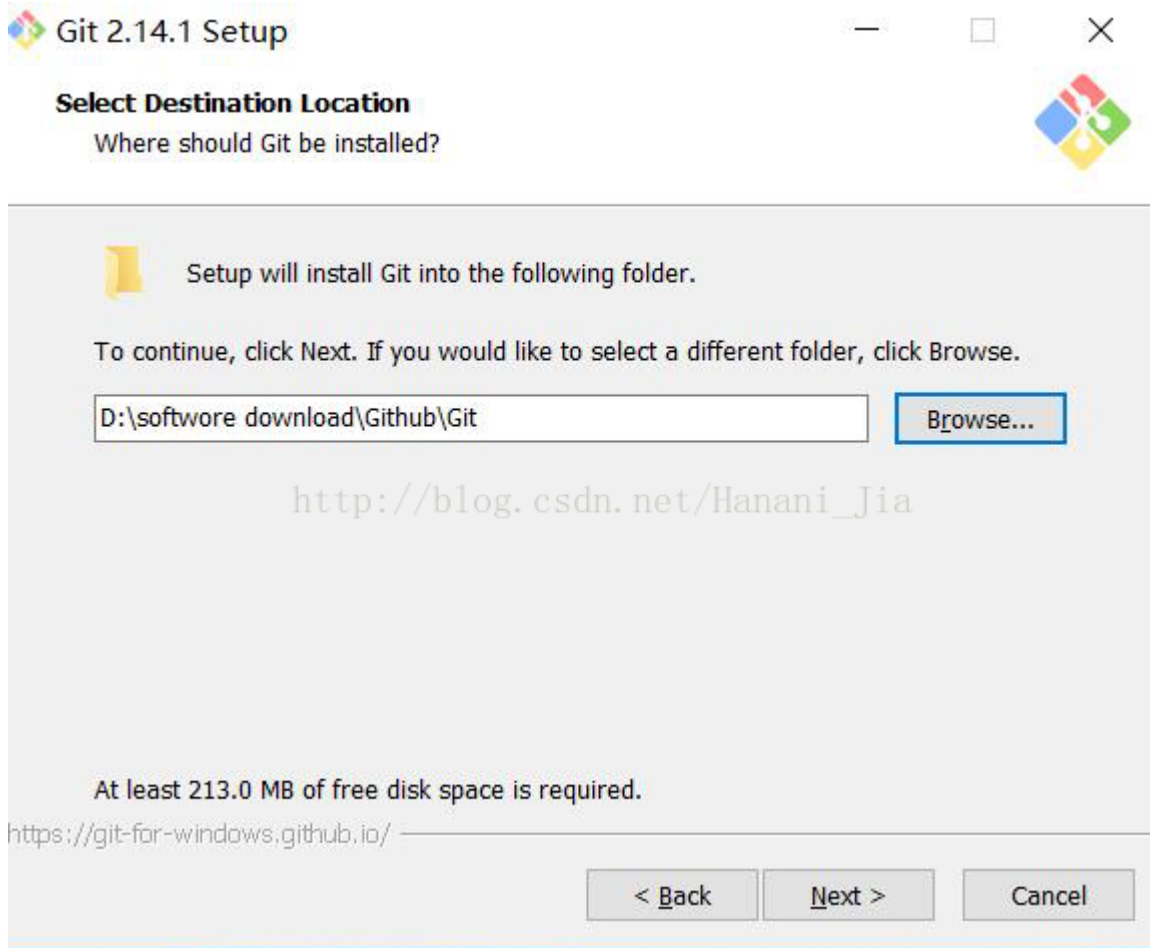
创建

b.初始化本地仓库

1)Git Bash 安装

本地仓库的初始化需要借助软件 Git Bash 来完成。[点击前往 Git 官网](#)

下载适用您的电脑的 Git 版本，下载完成后运行安装程序，过程中使用默认选项即可，若须修改安装路径，须保证安装路径为全英文。



Select Start Menu Folder

Where should Setup place the program's shortcuts?



Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Git

Browse...

http://blog.csdn.net/Hanani_Jia

Don't create a Start Menu folder

<https://git-for-windows.github.io/>

< Back

Next >

Cancel

完成安装后运行 Git Bash，出现下面这样的命令行窗口，

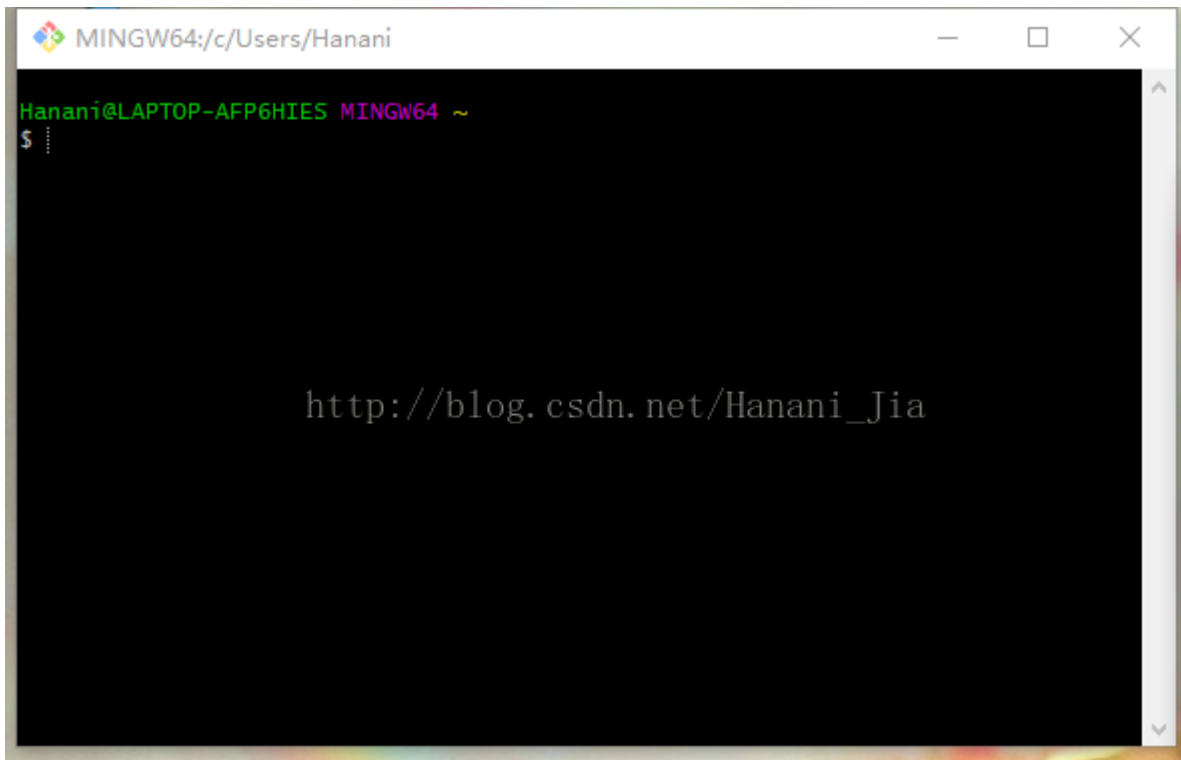
Hanani@LAPTOP-AFP6HIES MINGW64 ~ 是什么意思，

@前是计算机用户名，

@后是计算机名，

MINGW64 表示正在运行的 Git Bash 的信息

~表示根目录，默认情况下是 C:\Users\Administrator，这里是 C:\Users\Hanani



2)获取 SSHKey

首先要在本地创建一个 ssh key 这个的目的就是你现在需要在你电脑上获得一个密钥。

按如下命令来生成 sshkey:

```
$ ssh-keygen -t rsa -C "youremail@youremail.com"
# Generating public/private rsa key pair...
# 三次回车即可生成 ssh key
```

查看你的 public key

```
$ cat ~/.ssh/id_rsa.pub
# ssh-rsa AAAAB3NzaC1yc2E... youremail@youremail.com
```

并把他添加到 Gitee (gitee.com [SSHKey 添加地址](#))



添加后，在终端中输入

```
#Gitee
$ ssh -T git@gitee.com
```

```
#GitHub
$ ssh -T git@github.com
```

第一次绑定的时候输入上边的代码之后会提示是否 continue, 输入 yes 后程序会自动连接，如果要求登录，直接输入登录信息即可。

再次执行上面的命令，检查是否成功连接，如果返回一下信息，则表示添加成功

```
#Gitee
Welcome to Gitee.com, YourName!
```

```
#GitHub
You've successfully authenticated, but GitHub does not provide shell access.
```

3) 设置基本信息

接下来还需要简单的设置一些东西。

```
$ git config --global user.name "yourname"
$ git config --global user.email "youremail@youremail.com"
```

name 尽量和码云或 GitHub 保持一致，但 email 必须是码云或 GitHub 注册时使用的邮箱。命令不分前后，没有顺序。

4) 初始化本地库

然后就是将你的远程仓库克隆到本地，或者你可以在本地初始化一个项目后再进行云端绑定。

克隆



```
#Gitee
$ git clone https://gitee.com/yourname/repository

#Github
$ git clone https://github.com/yourname/repository.git

#yourname 您在码云或 github 注册的用户名
#repository 您创建的远程仓库名称
```



本地初始化



```
#Gitee
$ cd d:/test //首先在文件系统中创建一个项目文件夹，然后在 Git 中 cd 到这个工程目录
$ git init //初始化本地项目
$ git remote add origin <远程仓库地址> //绑定远程仓库
#注:地址形式为 https://gitee.com/yourname/test.git 或 git@gitee.com:yourname/test.git

#Github
$ cd d:/test
$ git init
$ git remote add origin <远程仓库地址>
#注:地址形式为 https://github.com/yourname/test.git
```

复制代码



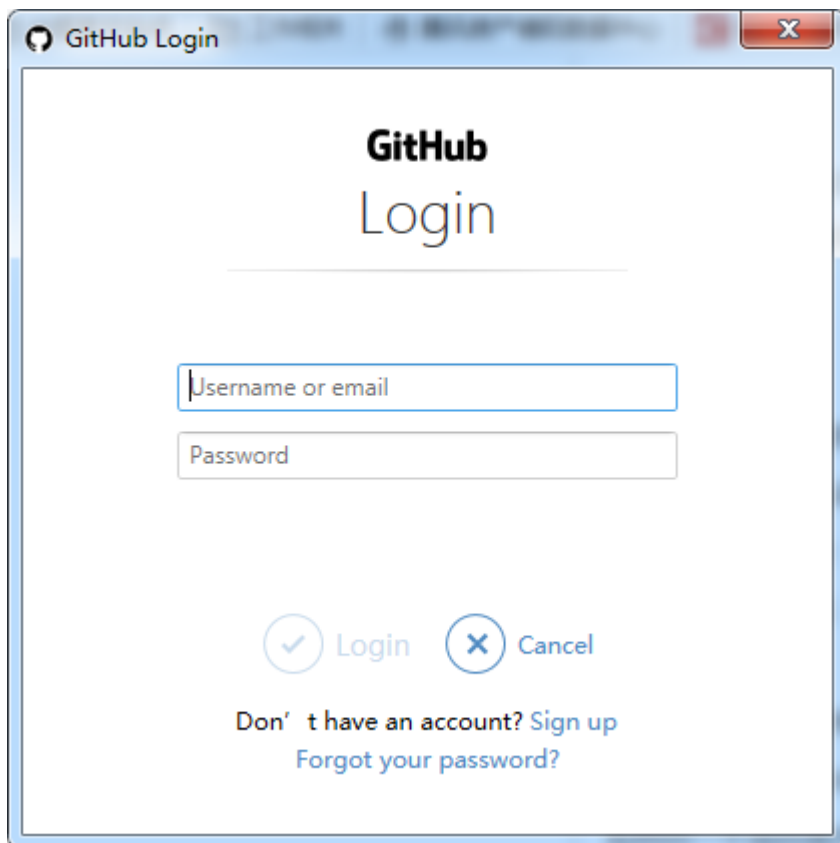
c.更新到远程仓库

在本地进行编辑后，更新到远程仓库

```
git add . //指定更新内容 . 表示全部更新，test.txt 表示更新指定文件
git commit -m "一些注释说明" //添加更新说明
git push origin master //执行更新操作
```

在执行更新操作时，由于需要验证远程仓库的用户信息，程序反应较慢，不要以为 Git Bash 挂掉了哦

如果是第一次更新，那么可能会跳出下面这样的窗口，输入您的用户名和密码，点击登录即可。



如果后续进行更新操作时出现这种情况，那么表示登录失败了，需要重新记录您的用户信息，再次输入用户名、密码登录即可

```
Logon failed, use ctrl+c to cancel basic credential prompt.  
Username for 'https://github.com'
```

如何从远程仓库同步最新版本到本地

```
$ cd d:/test  
$ git pull origin master
```

如何清屏

```
$ clear
```

} <https://www.cnblogs.com/einstein-2014731/p/11492637.html> 完

自己实操:

```
$ pkg install git
```

```
$ ssh-keygen -t rsa -C "gitee@注册邮箱.com" #就敲3次 enter
```

Generating public/private rsa key pair.

Enter file in which to save the key (/data/data/com.termux/files/home/.ssh/id_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /data/data/com.termux/files/home/.ssh/id_rsa.

Your public key has been saved in /data/data/com.termux/files/home/.ssh/id_rsa.pub.

The key fingerprint is:

SHA256:1ShAH 脱敏省略 e9Nmc gitee@注册邮箱.com

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

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCvwLZ/脱敏省略+Ss7q9k= gitee@注册邮箱.com

#将上述字符串作为 ssh key 导入 gitee.com, 参见 <https://www.cnblogs.com/einstein-2014731/p/11492637.html>

```
$ ssh -T git@gitee.com
```

Hi gitee 用户名! You've successfully authenticated, but GITEE.COM does not provide shell access.

#but GITEE.COM does not provide shell access 属于正常现象

```
$ cd /data/data/com.termux/files/home/2011081040gitee
```

```
$ git config --global user.name "gitee用户名"
```

```
$ git config --global user.email "gitee@注册邮箱.com"
```

```
$ git init
```

2011081225 删除文件{

<https://zhuanlan.zhihu.com/p/95508176>

如何删除 github 上的文件



左手牽右手

程序员

1.首先先把文件拉去下来 `git clone <file>` 看到 done 基本就成功了。

```
Administrator@BF-201911181550 MINGW64 /e/zld/备份文华/boinc_api (master)
$ git clone git@github.com:1712 " " git
Cloning into 'boinc_api'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 2275 (delta 1), reused 4 (delta 1), pack-reused 2271
Receiving objects: 100% (2275/2275), 1.23 MiB | 20.00 KiB/s, done.
Resolving deltas: 100% (1408/1408), done.

Administrator@BF-201911181550 MINGW64 /e/zld/备份文华/boinc_api (master)
$
```

2.首先删除本地仓库文件 `git rm <file>`

```
$ git rm build.php
rm 'build.php'
```

此时本地文件没有了，github 仓库上还有此文件。

3.提交并添加备注 `git commit -m "备注"`

```
$ git commit -m "删除build.php文件"
[master c28c997] 删除build.php文件
1 file changed, 25 deletions(-)
delete mode 100755 build.php
```

4.把删除的文件恢复到最新版本 git checkout

```
$ git checkout
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
```

5. 将本地内容推送到远程仓库

```
$ git push
Warning: Permanently added the RSA host key for IP address '13.250.177.223' to t
he list of known hosts.
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 239 bytes | 239.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Lee1713/boinc_api.git
4195636..c28c997 master -> master
```

知乎 @Zld

6.刷新 github 仓库 看到选定删除的文件没有了。

有问题可以私下沟通。

发布于 2019-12-05

自己实操：

```
$ pwd
/data/data/com.termux/files/home/2011081040gitee
$ git clone git@gitee.com:gitee用户名/it #这是网上你账户里已有的仓库
Cloning into 'it'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 47 (delta 15), reused 47 (delta 15), pack-reused 0
Receiving objects: 100% (47/47), 22.86 MiB | 3.18 MiB/s, done.
Resolving deltas: 100% (15/15), done.
$ cd it
$ ls
各种文件.....
1.docx
1.pdf

$ git rm 1.docx
rm '1.docx'
$ git commit -m "rm_w"
[master b9ca203] rm_w
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 1.docx
$ git checkout
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 214 bytes | 107.00 KiB/s, done.
```

Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Powered by GITEE.COM [GNK-5.0]
To gitee.com:gitee用户名/it
116f621..b9ca203 master -> master
上 [www.gitee.com/gitee 用户名/it/](http://www.gitee.com/gitee/用户名/it/)看, 网上的1.docx 也没了

```
$ pwd
/data/data/com.termux/files/home/2011081040gitee/it
$ ls
各种文件.....
1.pdf
$ cp 1.pdf ../..
$ git rm 1.pdf
rm '1.pdf'
$ git commit -m "1"
[master e81b05e] 1
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 1.pdf
$ git checkout
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 223 bytes | 223.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Powered by GITEE.COM [GNK-5.0]
To gitee.com:gitee用户名/it
b9ca203..e81b05e master -> master
上 www.gitee.com/gitee 用户名/it/看, 1.pdf 也没了
}2011081225 删除文件 完
```

上传文件

```
$ pwd ; ls
/data/data/com.termux/files/home/2011081040gitee/it
各种文件.....
$ touch test.txt #可以是创建, 也可以是从别处传过来
$ git add test.txt
$ git commit -m "add_test"
[master d10c482] add_test
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.txt
$ git checkout
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 263 bytes | 263.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Powered by GITEE.COM [GNK-5.0]
To gitee.com:gitee用户名/it
e81b05e..d10c482 master -> master
上 www.gitee.com/gitee 用户名/it/看, 也出现了 test.txt
对于 gitee、介于 10 与 100MB 之间的文件, 可以用此方法
```

新建文件夹

似乎不能上传空文件夹

```
$ pwd ; ls
各种文件.....
$ mkdir testDir
$ cd testDir/ ; touch desktop.ini
$ cd ..
$ git add testDir #不可以带/
$ git commit -m "1" ; git checkout ; git push
上 www.gitee.com/gitee 用户名/it/看, 也出现了 testDir
```

删除刚刚创建的文件夹

```
$ git rm -r testDir/ ; git commit -m "1" ; git checkout ; git push
rm 'testDir/desktop.ini'
[master 06c5b2e] 1
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 testDir/desktop.ini
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 207 bytes | 207.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Powered by GITEE.COM [GNK-5.0]
To gitee.com:gitee用户名/it
ba7ca11..06c5b2e master -> master
上 www.gitee.com/gitee 用户名/it/看, testDir 没了
```

termux/linux/unix 同一用户切换 gitee、github{

参考 https://blog.csdn.net/weixin_30399155/article/details/96124793{

git 中的 ssh 和 https 方式的使用 (gitee 为例)

转载



1101



收藏

[weixin_30399155](#) 2018-12-16 18:25:00

文章标签: [运维](#) [git](#) [开发工具](#)

版权

在使用 git 管理代码, 或者使用 github, 国内的码云(gitee)的时候, 有两种方式可以使用, 分别是 https 和 ssh, 以下均使用 gitee 为例。

1.

1. ssh 方式

配置 ssh，如果不配置 ssh 的话，clone 项目的时候会报错，原因是没有配置公钥。

1 Permission denied (publickey)

2 Could not read from remote repository.

首先，进入用户目录，查看本地是否已经存在公钥文件。

1 cd ~/.ssh

2 ls

如果显示文件夹不存在，则本地还没有生成密钥文件，则使用 git 提供的密钥生成工具进行生成。

1 ssh-keygen -t rsa -C "xxxxxx@xxxxxx.com"

注意，回车后有几次需要回车，第二部需要你设置密码，如果不设置直接回车，则以后在使用 ssh 方式拉取提交代码的时候不需要再输入密码。

完成后，用户文件夹下多出了一个 .ssh 文件夹，里面有两个文件，一个 id_rsa 和 id_rsa.pub 文件，使用命令打开 id_rsa.pub 文件，复制内容

1 cat ~/.ssh/id_rsa.pub

然后粘贴到 github 或 gitee 上的公钥中，可以给项目添加公钥，也可以给自己的账号添加公钥，二者的区别官方的解释是：

项目的 sshkey 只针对项目，且我们仅对项目提供了部署公钥，即项目下的公钥仅能拉取项目，这通常用于生产服务器拉取仓库的代码。而用户的 key 则是针对用户的，用户添加了 key 就对用户名下的项目和用户参加了的项目具有权限，一般而言，用户的 key 具有推送和拉取的权限，而项目的 key 则只具有拉取权限。

使用SSH公钥可以在任何计算机和服务器之间使用安全连接（Git的Remote需要使用SSH地址）

您当前的SSH公钥数: 0

你还没有添加任何SSH公钥

添加公钥

标题

公钥标题(key)

公钥

把你的公钥粘贴到这里，查看 [怎样生成公钥](#)

支持以 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384' or 'ecdsa-sha2-nistp521' 开头

确定

输入命令进行测试：

1 ssh -T git@gitee.com

如果返回 "Hi gitee 用户名 xxxxxx....." 说明配置成功了。

设置好了之后，然后拉取代码，第一次拉取的时候，会询问是否信任连接，yes 就行了，就可以正常拉取，提交代码了，也不需要设置每次都输入账号和密码了。

注意，信任完 gitee 的连接后，.ssh 文件夹里多出了一个 known_hosts 文件，这里就记录了 gitee 的相关连接信息。这里会有一个小问题，那假如我有多个 git 账号，比如：1，gitee 账号，用于自己个人 2，github 账号，用于公司开发，那么怎么配置多个密钥呢？

解决办法：1，生成多个密钥文件

1 ssh-keygen -t rsa -C 'xxxxxx@company.com' -f ~/.ssh/gitee_id_rsa

```
2 ssh-keygen -t rsa -C 'xxxxx@qq.com' -f ~/.ssh/github_id_rsa
```

2, 在 ~/.ssh 目录下新建一个 config 文件, 添加如下内容 (其中 Host 和 HostName 填写 git 服务器的域名, IdentityFile 指定私钥的路径)

```
1 # gitee
2 Host gitee.com
3 HostName gitee.com
4 PreferredAuthentications publickey
5 IdentityFile ~/.ssh/gitee_id_rsa
6 # github
7 Host github.com
8 HostName github.com
9 PreferredAuthentications publickey
10 IdentityFile ~/.ssh/github_id_rsa
```

3, 用 ssh 命令分别测试

```
1 $ ssh -T git@gitee.com
2 $ ssh -T git@github.com
```

2. https 方式

配置 credential.helper, 首先简单介绍一下 credential.helper 这个配置项的含义: 这个配置项允许用户自行指定 git 所使用的凭据管理工具。SSH 协议并不采用这里讨论的凭证存储。

首先, 检查 credential.helper 配置, 输入命令

```
1 git config -l|grep credential.helper
```

没有配置的话, 显示为空, windows 安装如果安装的是 GitGUI 的时候, 默认是勾选 manager 的,

credential.helper=manager, 如果没有配置的话, 每次拉取, 提交都需要密码。使用以下命令配置:

```
1 git config credential.helper manager
```

再次尝试 pull 代码的时候会弹出窗口要求输入用户名密码 (只需要输入这一次就 ok 了), 以后就不需要再输入密码了, windows 而言, 这个凭据放在 windows 的凭据管理器中

管理你的凭据

查看并删除你为网站、已连接的应用程序和网络保存的登录信息。



Web 凭据



Windows 凭据

[备份凭据\(B\)](#) [还原凭据\(R\)](#)

Windows 凭据

[添加 Windows 凭据](#)

无 Windows 凭据。

基于证书的凭据

[添加基于证书的凭据](#)

无证书。

普通凭据

[添加普通凭据](#)

[git:https://gitee.com](#)

修改时间: 2018/12/1



[virtualapp/didlogical](#)

修改时间: 2018/12/5



其他项目

[SSO_POP_Device](#)

修改时间: 2018/12/15



整个过程可以这样描述：当配置 `credential.helper` 后，`git` 在需要用户名密码时会首先向指定的凭据管理器查找凭据，如果凭据不存在（对应第一次 `pull`），则弹窗提示用户输入用户名密码，然后凭据管理器会记录这个凭据；如果凭据存在（对应第二次 `pull`），则直接使用该凭据进行对应的 `git` 操作。

注意：当你需要在同一台机器上使用多个 `git` 账号这么搞就不行了，因为 2 个账号必定使用的不同的凭据。

转载于：<https://www.cnblogs.com/rucker/p/10127058.html>

}参考 https://blog.csdn.net/weixin_30399155/article/details/96124793 完

自己实操：

```
$ mkdir 2011081400github
$ cd 2011081400github/
$ ssh-keygen -t rsa -C "github@注册邮箱.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/data/data/com.termux/files/home/.ssh/id_rsa): githubkey
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in githubkey.
Your public key has been saved in githubkey.pub.
The key fingerprint is:
SHA256:+rC9IIz脱敏省略3o7/Yw github@注册邮箱.com
The key's randomart image is:
+---[RSA 3072]---+
|      o*  . .o oo |
|    .+ . oo o.o+ |
|     .. .o o o * |
|      . o  + o .o |
```

```

|      =.S.  . . . |
|    + + *O  . . |
|    O B =O.. . O |
|    . = *O    =. |
|    . . +.    E = |
+----[SHA256]-----+
$ ls -lh
total 16K
-rw----- 1 u0_a282 u0_a282 2.5K Nov 8 14:10 githubkey
-rw----- 1 u0_a282 u0_a282 566 Nov 8 14:10 githubkey.pub
$ cd ../.ssh;ls
authorized_keys id_rsa id_rsa.pub known_hosts
$ pwd
/data/data/com.termux/files/home/.ssh
$ cd -
/data/data/com.termux/files/home/2011081400github
$ cat githubkey.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDHhQ1aBVRcYmPyCPo8chbm0Cx3ZEwLAM6mtFU9s0Y+脱敏省略
sXx7cg30bhBu94vFvQfXDuw6ifBKwli9U4uExBK9A13n8pHw/OoSA7Iw2uD8YbU= github@注册邮箱.com
#将其存入github.com的ssh key
$ mv githubkey* ../.ssh/
$ cd -
/data/data/com.termux/files/home/.ssh
$ ls -a
id_rsa.pub      known_hosts      authorized_keys  githubkey        githubkey.pub    id_rsa
$ touch config #新建一个文件，以使ssh区分不同网站用的不同key
$ nano config
$ cat config
# gitee
Host gitee.com
HostName gitee.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/id_rsa
# github
Host github.com
HostName github.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/githubkey
$ cd -
/data/data/com.termux/files/home/2011081400github
$ ssh -T git@github.com
The authenticity of host 'github.com (192.30.255.112)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCsPRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,192.30.255.112' (RSA) to the list of known hosts.
Hi Github用户名! You've successfully authenticated, but Github does not provide shell access.
$ git config --global user.name "Github用户名"
$ git config --global user.email github@注册邮箱.com
$ rm -rf .git/
$ git init
Initialized empty Git repository in /data/data/com.termux/files/home/2011081400github/.git/
$ git clone git@github.com:Github 用户名/it
.....这就可以了
} termux/linux/unix 同一用户切换 gitee、github 完

```

Windows

windows 类似。可在清华源下载。安装完 git for windows 后，将上述的.ssh 文件夹放在 windows 上 git 里的~/中即可
如果要生成 key，也跟上上述步骤差不多

其他

<https://www.cnblogs.com/jsdy/p/12322172.html> git add . 是添加所有当前目录的所有文件