

# ubuntu 和MAC上Git安装与GitHub基本使用 (ssh 密钥版本)

有用的网站:

<https://zhuanlan.zhihu.com/p/423066660> ubuntu git 环境搭建以及通过 SSH 连接 Github

[📖 clash使用手册](#) 提前翻墙 (组内翻墙vpn)

<https://www.jianshu.com/p/7edb6b838a2e> MAC上Git安装与GitHub基本使用

<https://www.runoob.com/w3cnote/view-ssh-public-key.html>查看本机 ssh 公钥，生成公钥

[📖 04. Github基本操作](#) 飞书群里的

## github使用ssh密钥

git使用https协议，每次pull, push都要输入密码，相当的烦。

使用git协议，然后使用ssh密钥。这样可以省去每次都输密码

以下是实际操作：

Git 本地环境配置1. 安装 git

```
1 sudo apt-get install git
```

2. 配置用户信息

```
1 $ git config --global user.name "hzhang2001"
2 $ git config --global user.email canghang@buaa.edu.cn
```

3. 初始化本地仓库配置

```
1 git init
```

4. 通过 SSH 连接 Github1. 安装 SSH

```
1 sudo apt-get install ssh
```

首先 ssh-keygen 会确认密钥的存储位置和文件名（默认是 .ssh/id\_rsa），然后他会要求你输入两次密钥口令，留空即可。所以一般选用默认，全部回车即可。

## 2. 创建密钥文件

```
1 ssh-keygen -t rsa -C "canghang@buaa.edu.cn"
```

<https://www.runoob.com/w3cnote/view-ssh-public-key.html>查看本机 ssh 公钥，生成公钥

## 4. SSH 测试 (这里如果不翻墙会有加载不出来情况, [目clash使用手册](#) 翻墙 (组内翻墙vpn) 不卡顿)

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

```
dell@dell-Precision-7920-Tower:~/soft/learn_abacus$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeI0ttrVc98/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,20.205.243.166' (ECDSA) to the list of known hosts.
Hi hzhang2001! You've successfully authenticated, but GitHub does not provide shell access.
dell@dell-Precision-7920-Tower:~/soft/learn_abacus$
```

## Git 流程：

建立远程仓库：

- 登录GitHub账户，点击页面右上角的"+"按钮，选择"New repository"。
- 根据提示输入仓库名称、描述等信息，点击"Create repository"按钮。

### 1. 初始化 Git 本地仓库

在要管理代码的本地目录下，初始化一个 Git 仓库：使用 `git init` 命令在项目根目录下初始化一个 Git 仓库。这将在项目目录中创建一个隐藏的 `.git` 文件夹，用于存储版本控制信息和历史记录。

```
1 $ cd /path/to/local/folder
2 $ git init
```

## 2. Clone一个仓库：

- 在GitHub上找到您要下载的仓库，点击"Code"按钮，选择"SSH"选项卡，复制SSH URL。

```
1 git clone git@github.com:user/repo.git
```

- 其中，user是您在GitHub上的用户名，repo是要下载的仓库名称。

拉取最新代码：

- 在终端（Linux/MacOS）或Git Bash（Windows）中，进入到仓库目录，输入以下命令：

```
1 git pull origin master
2 // 替换操作(优先级别)
3 git fetch//用于从远程仓库中下载最新的代码和分支信息，但是并不会自动合并到当前分支中，不会
4 //远程分支的信息保存在本地的一个叫做“远程跟踪分支”（Remote Tracking Branch）的引用中。
5 git merge origin/master //上面的命令将最新的 origin/master 分支合并到当前分支中。
```

- 其中，origin是远程仓库的名称，master是分支名称。

## 3. 拉取指定分支代码：

- 在终端（Linux/MacOS）或Git Bash（Windows）中，进入到仓库目录，输入以下命令：

```
1 git pull origin branch_name
```

- 其中，branch\_name是要拉取的分支名称。

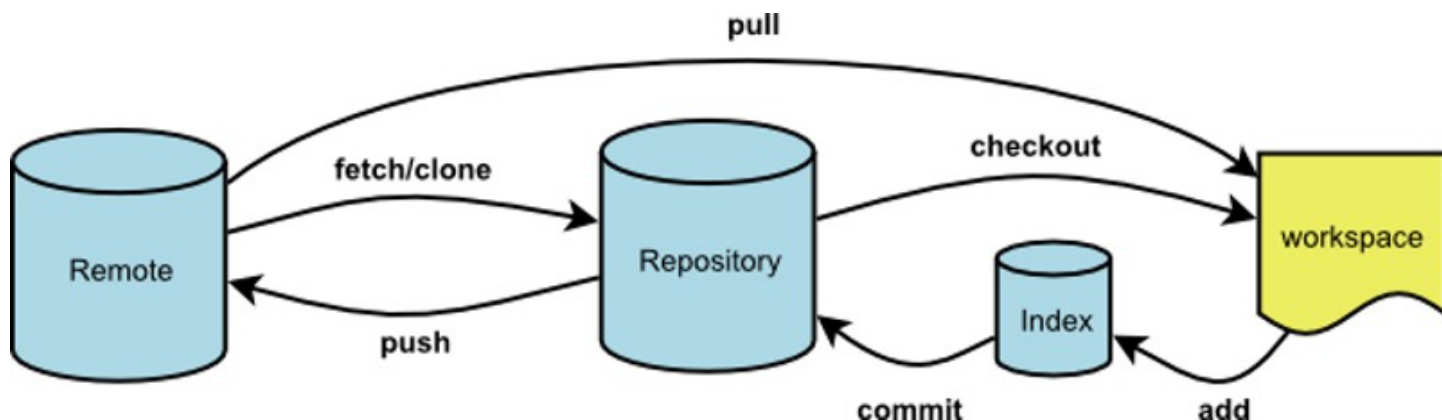
## 4. 推送更改到GitHub：

- 在本地修改文件后，使用以下命令将更改推送到GitHub：

```
1 git add . //添加要上传的文件到.git 本地仓库
2 git status // 查看文件状态
3 git commit -m "Commit message" // 暂存区的文件提交到 Git 仓库中，并编写描述信息
4 git push origin master //将本地Git仓库中的代码推送到名origin的远程仓库master分支（需要
```

- 将要提交的文件添加到 Git 的暂存区中，使用 `git add` 命令,可以使用 `git add .` 将当前目录下的所有文件添加到暂存区 `git add file1.txt file2.txt`
- 将添加到暂存区的文件提交到 Git 仓库中，使用 `git commit` 命令，`-m` 这个命令将打开一个文本编辑器，允许您编写一条提交消息，以便描述您所做的更改。

- origin 是远程仓库名字，master是分支，会出现远程仓库没有的报错，此时需要添加远程仓库，先查看远程仓库 `git remote -v`



```
dell@dell-Precision-7920-Tower:~/soft/learn_abacus$ git remote -v
dell@dell-Precision-7920-Tower:~/soft/learn_abacus$ git remote add origin git@github.com:hzhang2001/abacus-develop.git
dell@dell-Precision-7920-Tower:~/soft/learn_abacus$ git remote -v
origin  git@github.com:hzhang2001/abacus-develop.git (fetch)
origin  git@github.com:hzhang2001/abacus-develop.git (push)
dell@dell-Precision-7920-Tower:~/soft/learn_abacus$ git branch
* master
dell@dell-Precision-7920-Tower:~/soft/learn_abacus$ git push -u origin master
总共 0 (差异 0) , 复用 0 (差异 0) , 包复用 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/hzhang2001/abacus-develop/pull/new/master
remote:
To github.com:hzhang2001/abacus-develop.git
 * [new branch]      master -> master
分支 'master' 设置为跟踪 'origin/master'。
dell@dell-Precision-7920-Tower:~/soft/learn_abacus$
```

## 5. 配置远程仓库

如果你想将本地的仓库推送到 GitHub 上，可以先在 GitHub 上创建一个空仓库，然后使用以下命令将本地仓库与 GitHub 上的远程仓库关联起来：

```
1 git remote add origin git@github.com:<username>/<repository>.git
2 git remote -v // 查看远程仓库列表（下文的origin）
3 git branch // 查看本地仓库列表（下文的master）
```

其中，`origin` 是远程仓库的名称，`<username>` 是你的 GitHub 用户名，`<repository>` 是你想要推送的仓库名称。

关联完成之后，就可以使用 `git push` 命令将本地仓库推送到 GitHub 上了：

```
1 git push -u origin master
```

其中，`master` 是要推送的本地分支名称。第一次推送需要使用 `-u` 参数，表示将本地分支与远程分支关联起来，以后可以直接使用 `git push` 命令进行推送

## 其他有用的操作：

### 1. 分支管理：切换，创建，合并，删除分支

```
1 $ git branch new_branch //在 Git 仓库中创建一个新的分支，使用 git branch 命令。
2 $ git checkout new_branch //切换到新的分支中，使用 git checkout 命令。
3 $ git checkout -b new_branch 命令创建并切换到新的分支。
4 $ git branch // 查看本地分支，所在分支用*标明
5 $ git merge new_branch //将两个分支合并，使用 git merge 命令。
6 $ git branch -d feature-1 //没合分支，会显示（本地）
7 $ git branch -D feature-1 //强制删除分支（本地）
8 $ git push origin --delete feature-1 //远程仓库删除，origin 是远程仓库的名称，
9     <branch-name> 是要删除的远程分支名称。例如，要删除名为 feature-1 的远程分支
```

### 2. 查看历史记录：

```
1 $ git log //命令可以查看 Git 仓库中的提交历史记录
```

这将显示所有提交的、作者、时间戳和提交消息。`git log --oneline` 将历史提交显示为单行信息

3. 远程仓库操作：使用 `git remote` 命令可以查看远程仓库。使用 `git fetch` 命令可以从远程仓库获取最新的提交历史记录。

### 4. 撤销更改：

```
1 $ git checkout file.txt
```

撤销对文件的更改，使用 `git checkout` 命令。

```
1 $ git reset HEAD file.txt
```

撤销对文件的添加，使用 `git reset` 命令

```
1 $ git revert commit_id
```

撤销对文件的提交，使用 `git revert` 命令。