

Linux 下如何安装和使用 Git

本文介绍的是在 linux 系统下的 github 的安装以及使用，windows 下的相关步骤可以与此类似，可以参考本文配置。

一、Git 安装：

1、二进制方式安装（在线安装）：

```
$ sudo apt-get install git 或者 sudo yum install git
```

安装完成后，在终端中输入 git 就可以看到相关的命令了。如果只是需要使用 git 来管理本地的代码，或者你只想将 github 上感兴趣的代码拷贝到本地，那么现在就可以使用了。如果需要和 github 上的项目结合，还需要做其他的一些操作。比如同步本地代码，代码协助等，那么你需要继续下面的步骤。

2、github 帐号的申请

github 就是为了开源的目的，首先去 github.com 上注册一个帐号。记住你申请时使用的邮箱，供后面配置 ssh 连接时使用。如本文使用的邮箱是 850692428@qq.com，默认的用户名是邮箱，而根目录是以邮箱前面数字作为默认根目录，若嫌不好记，可以更改，本文更改成的用户名 "cunwyl"，具体的方法：点击 setting -> Account -> Change username。取一个自己喜欢的名字，这个名字不能是中文，因为这个名字将会作为你 github 的根目录。

3、在本地建立一个文件夹，这个文件夹将会作为你上传或者下载工程文件的根目录。然后做一些全局变量的初始化

```
$ git config --global user.name = "cunwyl"
$ git config --global user.email = "850692428@qq.com"
```

这两个选项会在以后的使用过程中自动添加到代码中。

4、创建验证用的公钥

这个是比较复杂和困扰大多数人的地方，因为 git 是通过 ssh 的方式访问资源库的，所以需要在本地创建验证用的文件。使用命令：

```
$ ssh-keygen -C '850692428@qq.com' -t rsa -f .ssh/id_rsa.github
```

-C 指定产生哪个用户的密钥对，-f 指定生成的密钥文件名。注意：若你原来在当前用户目录下已经生成过其他的密钥，则会覆盖其密钥，覆盖则原来密钥将会失效，这是我们不想要的。所以我们需要另外指定生成的密钥使用者和密钥文件。按一系列回车键后，密钥生成。密钥生成后，关键点来了，我们现在有这个需求，我们想在同一个用户下 ssh 访问不同的不同用户的仓库，按 ssh 的规则会产生不同对的密钥对，而默认的访问 ssh 会调取本地的私钥 id_rsa 与远程主机上的公钥进行匹配，但是现在本地产生了多个私钥怎么办？而且私钥暂时它是不支持将所有的私钥都放到 id_rsa 文件里的，他现在只是支持但私钥的，你不能在 id_rsa 里面放置多个私钥，没有办法匹配。那问题怎么解决呢？有一种方法可以解决，在.ssh下面新建一个配置文件 config，这个文件的用途相当于做了一个指定私钥的作用。就是你访问哪些主机我就让你去读取与那个主机在本机上对应的私钥，这里远程主机上保存的是本机的公钥。我们要求访问 github 服务器时就让它去匹配本地的私钥 id_rsa.github，config

文件内如下：

```
host github.com
  identityfile ~/.ssh/id_rsa.github
  user wang
```

host: 指定访问主机（域名或者 ip 都可以）

identityfile: 指定我要匹配的本地私钥文件

user : 指定本机的用户名

这里不需要在这里配置 id_rsa 私钥，不添加表示当本机登录其他主机时都会以这个私密钥为默认密钥进行匹配，你也可以添加都没问题，这里主要是考虑到我访问的主机 ip 在变化，所以没添加。

查看一下本机的.ssh，该目录下现在有这些文件，其中 github 的密钥文件很明显我已经特别标出，还需注意必须有 config 文件

wang@kubuntu:~/.ssh\$ tree

```
-- authorized_keys
-- config
-- id_rsa
-- id_rsa.github
-- id_rsa.github.pub
-- id_rsa.pub
-- known_hosts
```

我们折腾这么就，就是为了解决单个用户下多个管理多个私钥共存的情况。这种情况在服务器上比较常见。如果你之前没有配置过 ssh，那就不用折腾这些了，直接一阵回车后就 OK 了。下一步是上传公钥。更详细的帮助可参考 <https://www.zybuluo.com/yangfch3/note/172120>

5、上传公钥

在 github.com 的界面中 选择右上角的 Account -> Settings，然后选择 SSH Public Keys，选择新加。Title 可以随便命名，Key 的内容拷贝自 ~/.ssh/id_rsa.github.pub 中的内容，完成后，可以再使用 ssh -v git@github.com 进行测试。看到下面的信息表示验证成功。

```
wang@kubuntu:~/Repository/2016$ ssh git@github.com
Warning: Permanently added the RSA host key for IP address '192.30.252.122' to the list of known hosts.
PTY allocation request failed on channel 0
Hi cnuwyl! You've successfully authenticated, but GitHub does not provide shell access.
Connection to github.com closed.
```

二、Git 配置与使用

利用 github 来管理自己的项目，可以按照下面的步骤进行

1.建立远端仓库。

在 Github 创建账号后，点击 New Repository，填写一些本仓库相关的信息，如仓库名称、描述、是否公开。设置完毕后，点击 Create repository 即可创建新仓库。如建立 2016 仓库。

2.建立本地仓库

在需要建立项目的文件夹中，初始化仓库的建立。完成后，可以看到文件家中多了一个 .git 隐藏目录。

```
$ git init
```

3.添加文件到缓存

使用 git add . 来进行初始文件的添加。这里 . 表示将文件夹下所有的文件都添加进去，我们也可以指定文件进行添加。

```
$git add .
```

4.提交文件到本地仓库, 可以将编辑的内容进行提交。提交到本地仓库。

```
$ git commit -m '第一次提交' ,
```

5 提交至 github 仓库

```
$ git push origin master
```

若提示 origin 已经存在，则可以使用如下命令解决

```
$ git remote rm origin  
$ git remote add origin git@github.com:cnuwyl/2016.git  
$ git push -u origin master
```

6 若要更新仓库

```
$ git push -u origin master
```

7 删除远程仓库文件。使用 git rm 命令，有两种选择.

一种是 git rm --cached "文件路径"，不删除物理文件，仅将该文件从缓存中删除；

一种是 git rm --f "文件路径"，不仅将该文件从缓存中删除，还会将物理文件删除

假如你有文件不小心 commit 到了服务器那么你想要删除它,可以使用:

```
$ git rm --cached "路径+文件名"
```

```
$ git commit -m "delete file"
```

```
$ git push
```