远程仓库

Git是分布式版本控制系统，同一个Git仓库，可以分布到不同的机器上。怎么分布呢？最早，肯定只有一台机器有一个原始版本库，此后，别的机器可以“克隆”这个原始版本库，而且每台机器的版本库其实都是一样的，并没有主次之分。

你肯定会想，至少需要两台机器才能玩远程库不是？但是我只有一台电脑，怎么玩？

其实一台电脑上也是可以克隆多个版本库的，只要不在同一个目录下。不过，现实生活中是不会有人这么傻的在一台电脑上搞几个远程库玩，因为一台电脑上搞几个远程库完全没有意义，而且硬盘挂了会导致所有库都挂掉，所以廖雪峰老师不教我也不知道怎么在一台电脑上克隆多个仓库。

实际情况往往是这样，找一台电脑充当服务器的角色，每天24小时开机，其他每个人都从这个“服务器”仓库克隆一份到自己的电脑上，并且各自把各自的提交推送到服务器仓库里，也从服务器仓库中拉取别人的提交。

现阶段不想自己搭建一台运行Git的服务器，所以用GitHub救急！

1.创建SSH Key

在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下有没有id\_rsa和id\_rsa.pub这两个文件，如果已经有了，可直接跳到下一步。如果没有，打开Shell（Windows下打开Git Bash），创建SSH Key：

$ ssh-keygen -t rsa -C "youremail@example.com"

然后一路回车使用默认值

如果一切顺利的话，可以在用户主目录里找到.ssh目录，里面有id\_rsa和id\_rsa.pub两个文件，这两个就是SSH Key的秘钥对，id\_rsa是私钥，不能泄露出去，id\_rsa.pub是公钥，可以放心地告诉任何人。

2.登陆 GitHub,setting→SSH and GPG keys→New SSH key

填上任意Title，在Key文本框里粘贴id\_rsa.pub文件的内容

当然，GitHub允许你添加多个Key。假定你有若干电脑，你一会儿在公司提交，一会儿在家里提交，只要把每台电脑的Key都添加到GitHub，就可以在每台电脑上往GitHub推送了。

啊，学了git以后终于能拯救我那台不能登陆GitHub客户端的老年笔记本了。。

添加远程库

现在的情景是，你已经在本地创建了一个Git仓库后，又想在GitHub创建一个Git仓库，并且让这两个仓库进行远程同步，这样，GitHub上的仓库既可以作为备份，又可以让其他人通过该仓库来协作，真是一举多得。

首先，登陆GitHub，然后，找到绿色按钮new respository，创建一个新的仓库：

在Repository name填入example，其他保持默认设置，点击“Create repository”按钮，就成功地创建了一个新的Git仓库：

目前，在GitHub上的这个example仓库还是空的，GitHub告诉我们，可以从这个仓库克隆出新的仓库，也可以把一个已有的本地仓库与之关联，然后，把本地仓库的内容推送到GitHub仓库。

现在，我们根据GitHub的提示，在本地的example仓库下运行命令：

$ git remote add origin git@github.com:yuanjiaCN/testGit.git

添加后，远程库的名字就是origin，这是Git默认的叫法，也可以改成别的，但是origin这个名字一看就知道是远程库。

然后就出现了很可怕的问题！报错了！

提示出错信息：fatal: remote origin already exists.

    解决办法如下：

    1、先输入$ [**git**](http://lib.csdn.net/base/git) remote rm origin

    2、再输入$ git remote add origin [git@github.com:djqiang/gitdemo.git](mailto:git@github.com:djqiang/gitdemo.git) 就不会报错了！

下一步，就可以把本地库的所有内容推送到远程库上：

$ git push -u origin master

把本地库的内容推送到远程，用git push命令，实际上是把当前分支master推送到远程。

由于远程库是空的，我们第一次推送master分支时，加上了-u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令。

推送成功后，可以立刻在GitHub页面中看到远程库的内容已经和本地一模一样：

但是果不其然又报错了！

1. Permission denied (publickey).
3. fatal: The remote end hung up unexpectedly
4. 应该是ssh key过期了吧
5. 试着重新创建ssh key，按以下步骤
6. ssh-keygen

然后回车回车，一切默认！

.ssh文档里又会有新的KEY了，复制一份，在GitHub上添加

重新重复$ git push -u origin master

搞定！

$ git push -u origin master

Warning: Permanently added the RSA host key for IP address '192.30.255.113' to the list of known hosts.

Counting objects: 3, done.

Writing objects: 100% (3/3), 219 bytes | 219.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To github.com:yuanjiaCN/example.git

\* [new branch] master -> master

Branch master set up to track remote branch master from origin.

从现在起，只要本地作了提交，就可以通过命令：

$ git push origin master

把本地master分支的最新修改推送至GitHub，现在，你就拥有了真正的分布式版本库！

SSH警告

当你第一次使用Git的clone或者push命令连接GitHub时，会得到一个警告：

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

RSA key fingerprint is xx.xx.xx.xx.xx.

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

这是因为Git使用SSH连接，而SSH连接在第一次验证GitHub服务器的Key时，需要你确认GitHub的Key的指纹信息是否真的来自GitHub的服务器，输入yes回车即可。

Git会输出一个警告，告诉你已经把GitHub的Key添加到本机的一个信任列表里了：

Warning: Permanently added 'github.com' (RSA) to the **list** of known hosts.

这个警告只会出现一次，后面的操作就不会有任何警告了。

#### 从远程库克隆

假设我们从零开发，那么最好的方式是先创建远程库，然后，从远程库克隆。

首先，登陆GitHub，创建一个新的仓库，名字叫test：

勾选Initialize this repository with a README，这样GitHub会自动为我们创建一个README.md文件。创建完毕后，可以看到README.md文件：

$ git clone git@github.com:yuanjiaCN/test.git

Cloning into 'test'...

remote: Counting objects: 3, done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Receiving objects: 100% (3/3), done.

已经看到test的文件夹和里面的.git以及README.md文件啦

git config --system http.sslcainfo " C:\Users\Administrator\Documents\GitHub\ yuanjiaCN.github.io

"