Git

# 介绍

1. Git是目前世界上最先进的分布式版本控制系统，c语言写的，
2. CVS及SVN都是集中式的版本控制系统，而Git是分布式版本控制系统。区别

<https://www.liaoxuefeng.com/wiki/896043488029600/896202780297248>

## 安装

### Linux上安装Git

1. 你可以试着输入git，看看系统有没有安装Git
2. Debian或Ubuntu Linux，通过一条sudo apt-get install git就可以直接完成Git的安装

### 在Windows上安装Git

在Windows上使用Git，可以从Git官网直接（https://git-scm.com/downloads）下载安装程序，然后按默认选项安装即可。

安装完成后，在开始菜单里找到“Git”->“Git Bash”，蹦出一个类似命令行窗口的东西，就说明Git安装成功！



安装完成后，还需要最后一步设置，在命令行输入：

$ git config --global user.name "Your Name"

$ git config --global user.email "email@example.com"

**git config --global user.name "zxh0120"**

**git config --global user.email 1445618731@qq.com**

因为Git是分布式版本控制系统，所以，每个机器都必须自报家门：你的名字和Email地址。你也许会担心，如果有人故意冒充别人怎么办？这个不必担心，首先我们相信大家都是善良无知的群众，其次，真的有冒充的也是有办法可查的。

### 创建版本库及命令

1. 创建一个版本库非常简单，首先，选择一个合适的地方，创建一个空目录：

$ mkdir learngit

$ cd learngit

$ pwd

/Users/michael/learngit

1. 第二步，通过git init命令把这个目录变成Git可以管理的仓库：

$ git init

Initialized empty Git repository **in** /Users/michael/learngit/.git/

瞬间Git就把仓库建好了，而且告诉你是一个空的仓库（empty Git repository），细心的读者可以发现当前目录下多了一个.git的目录，这个目录是Git来跟踪管理版本库的，没事千万不要手动修改这个目录里面的文件，不然改乱了，就把Git仓库给破坏了。

如果你没有看到.git目录，那是因为这个目录默认是隐藏的，用ls -ah命令就可以看见。

#### git add

第一步：用命令git add告诉Git，把文件添加到仓库

git add readme.txt

#### git commit

第二步：用命令git commit告诉Git，把文件提交到仓库：

$ git commit -m "wrote a readme file"

[master (root-commit) eaadf4e] wrote a readme file

1 file changed, 2 insertions(+)

create mode 100644 readme.txt

Git添加文件需要add，commit一共两步呢？因为commit可以一次提交很多文件，所以你可以多次add不同的文件，比如：

$ git add file1.txt

$ git add file2.txt file3.txt

$ git commit -m "add 3 files."

#### git status

运行git status命令看看结果：

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to **update** what will be committed)

(use "git checkout -- <file>..." **to** discard changes **in** working directory)

modified: readme.txt

**no** changes added **to** **commit** (use "git add" **and**/**or** "git commit -a")

git diff顾名思义就是查看difference，显示的格式正是Unix通用的diff格式

$ git add readme.txt

#### git log

git log命令显示从最近到最远的提交日志，我们可以看到3次提交，最近的一次是append GPL，上一次是add distributed，最早的一次是wrote a readme file。

如果嫌输出信息太多，看得眼花缭乱的，可以试试加上--pretty=oneline参数：

$ git log --pretty=oneline

1094adb7b9b3807259d8cb349e7df1d4d6477073 (HEAD -> master) append GPL

e475afc93c209a690c39c13a46716e8fa000c366 add distributed

eaadf4e385e865d25c48e7ca9c8395c3f7dfaef0 wrote a readme file

#### git reset --hard HEAD

你回退到了某个版本，关掉了电脑，第二天早上就后悔了，想恢复到新版本怎么办？找不到新版本的commit id怎么办？

在Git中，总是有后悔药可以吃的。当你用$ git reset --hard HEAD^回退到add distributed版本时，再想恢复到append GPL，就必须找到append GPL的commit id。Git提供了一个命令git reflog用来记录你的每一次命令：

$ git reflog

e475afc HEAD@{1}: reset: moving to HEAD^

1094adb (HEAD -> master) HEAD@{2}: commit: append GPL

e475afc HEAD@{3}: commit: add distributed

eaadf4e HEAD@{4}: commit (initial): wrote a readme file

#### git rm

一是确实要从版本库中删除该文件，那就用命令git rm删掉，并且git commit：

### 远程仓库

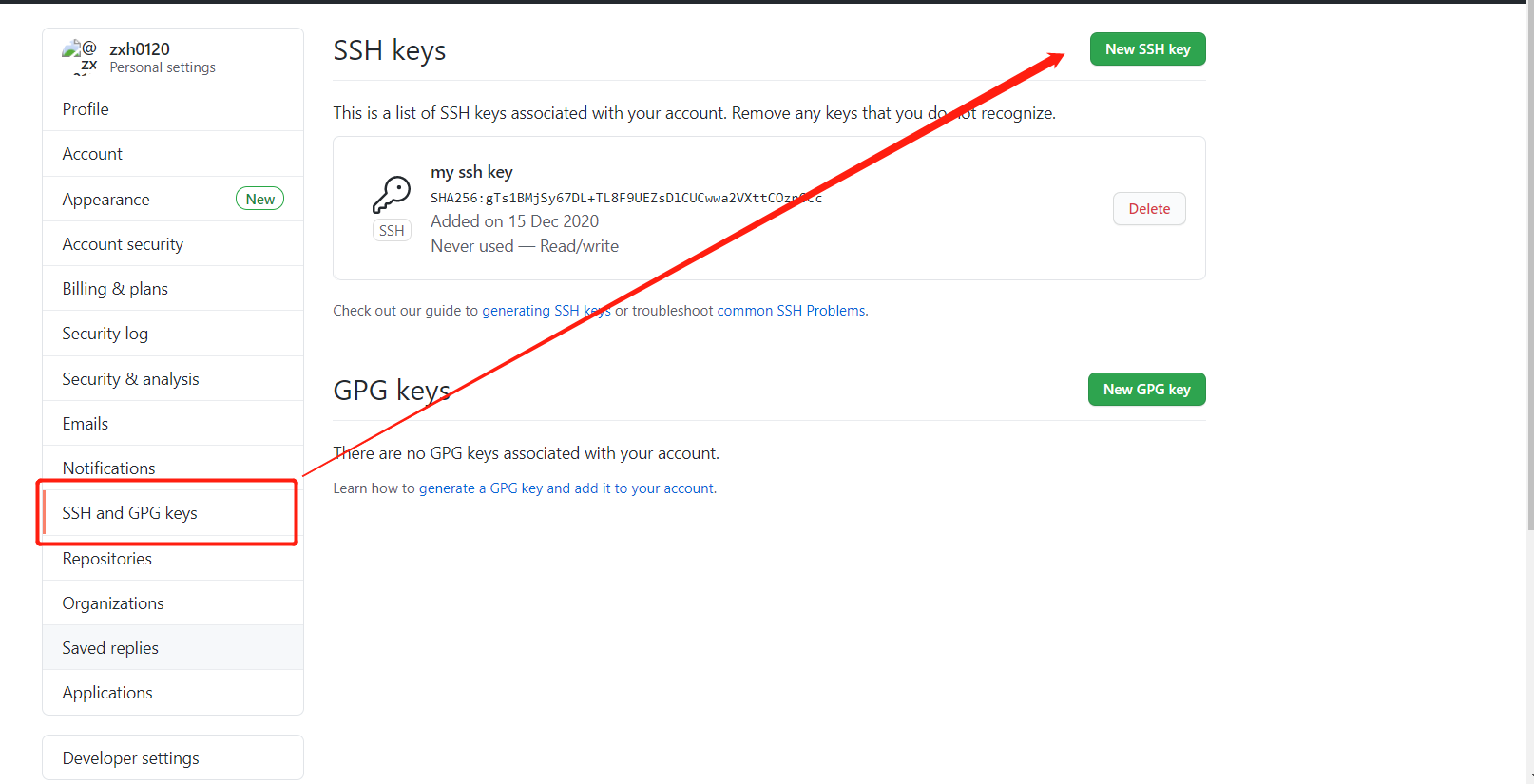
第1步：创建SSH Key。在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下有没有id\_rsa和id\_rsa.pub这两个文件，如果已经有了，可直接跳到下一步。如果没有，打开Shell（Windows下打开Git Bash），创建SSH Key：

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

你需要把邮件地址换成你自己的邮件地址，然后一路回车，使用默认值即可，由于这个Key也不是用于军事目的，所以也无需设置密码。

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。

第2步：登陆GitHub，打开“Account settings”，“SSH Keys”页面：



然后，点“Add SSH Key”，填上任意Title，在Key文本框里粘贴id\_rsa.pub文件的内容：

在右上角找到“Create a new repo”按钮，创建一个新的仓库：

在Repository name填入learngit，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：





### 从远程库克隆

#### git clone

$ git clone git@github.com:michaelliao/gitskills.git

Cloning into 'gitskills'...

remote: Counting objects: 3, done.

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

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

## 分支管理

我们创建dev分支，然后切换到dev分支：

### git checkout -b

$ git checkout -b dev

git checkout命令加上-b参数表示创建并切换，相当于以下两条命令：

### git branch 创建分支

$ git branch dev

### git switch -c切换分支

$ git switch -c dev **~~git checkout（不建议）~~**

Switched to branch 'dev'

### git branch 查看当前分支

$ git branch

\* dev

master

### git merge 合并分支

现在，我们把dev分支的工作成果合并到master分支上：

$ git merge dev

Updating d46f35e..b17d20e

Fast-forward

readme.txt | 1 +

1 file changed, 1 insertion(+)

### git branch -d

删除dev分支了：

$ git branch -d dev

## Bug分支

## 标签管理

#### 创建标签

##### git tag <name>打一个新标签

$ git tag v1.0

#### git tag 查看所有标签

$ git tag

v1.0

## 本地项目上传github上面

1. 需要先创建一个本地的版本库（其实也就是一个文件夹）。你可以直接右击新建文件夹，也可以右击打开Git bash命令行窗口通过命令来创建。
2. 通过命令git init把这个文件夹变成Git可管理的仓库



发现里面多了个.git文件夹，它是Git用来跟踪和管理版本库的。如果你看不到，是因为它默认是隐藏文件，那你就需要设置一下让隐藏文件可见。

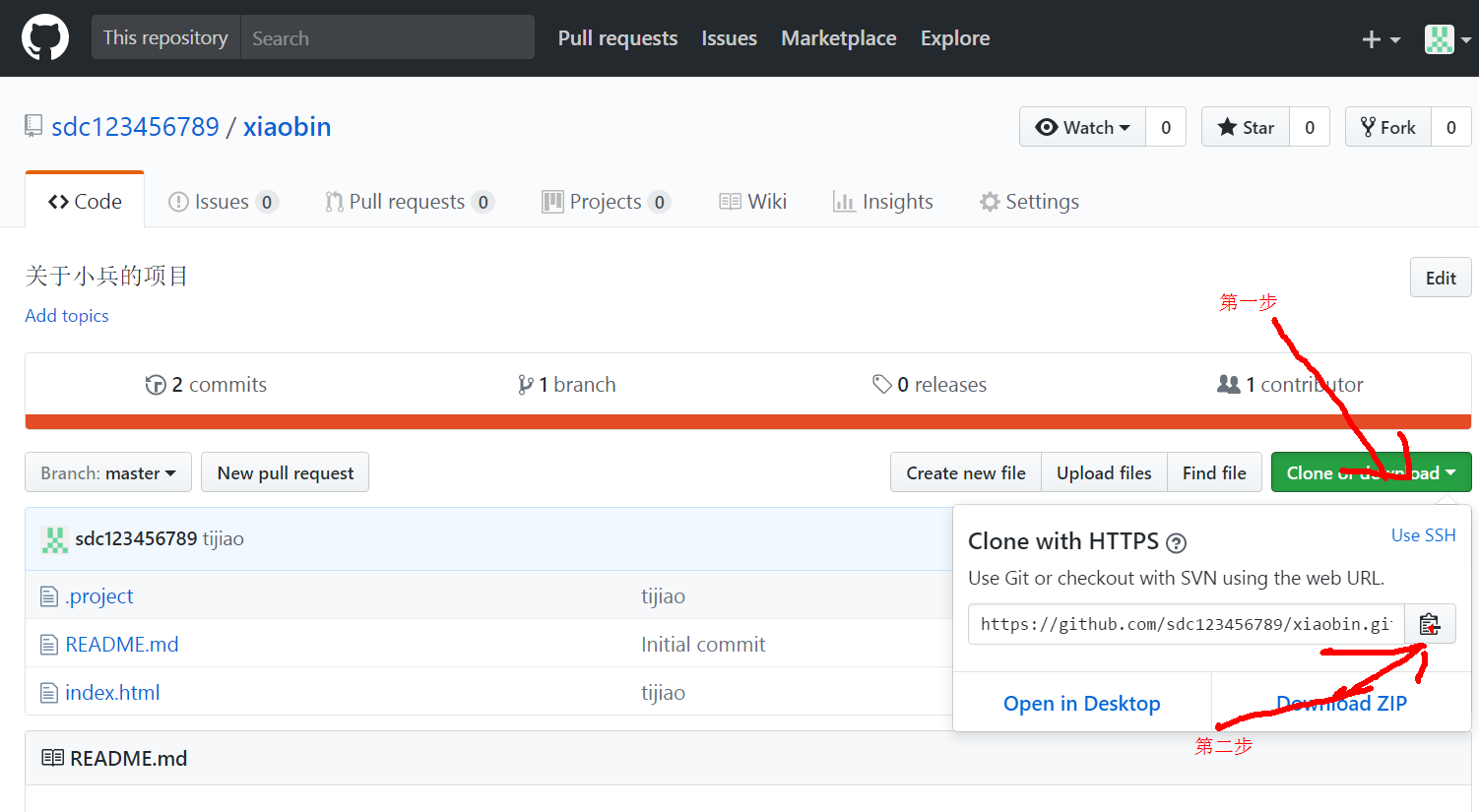
git add把项目添加到仓库（或git add .把该目录下的所有文件添加到仓库，注意点是用空格隔开的）

这个过程中你可以一直使用git status来查看你当前的状态。如果文件内有东西会出现红色的字，不是绿色，这不是错误

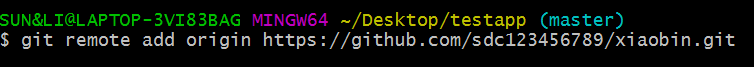
git commit -m "日志" 把项目提交到仓库。

Github上创建一个Git仓库。

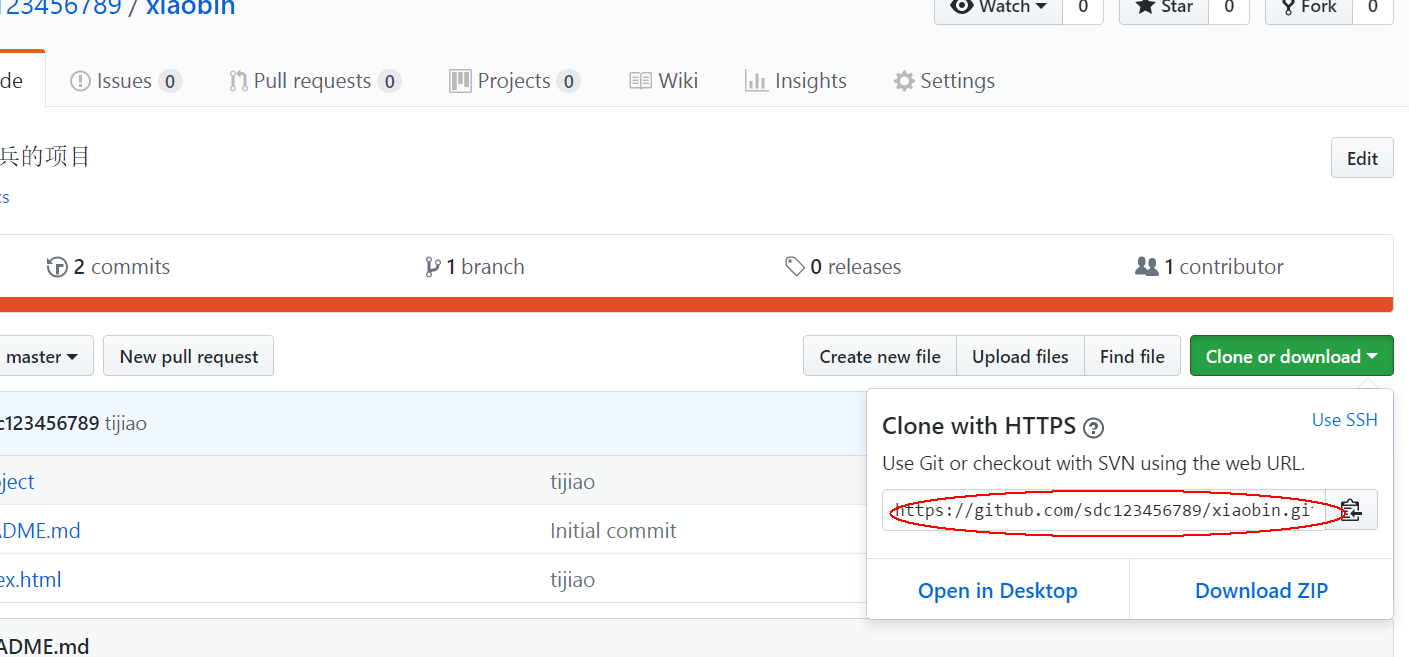
  你可以直接点New repository来创建，比如我创建了一个xiaobing的仓库.进入仓库



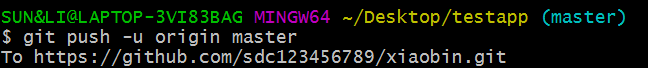
在Github上创建好Git仓库之后我们就可以和本地仓库进行关联了，根据创建好的Git仓库页面的提示，可以在本地testapp仓库的命令行输入：



注意origin后面加的是你Github上创建好的仓库的地址,就是上图第二步点击复制的地址。



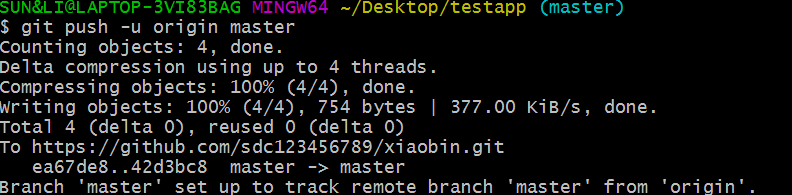
关联好之后我们就可以把本地库的所有内容推送到远程仓库（也就是Github）上了，通过：$ git push -u origin master



 由于新建的远程仓库是空的，所以要加上-u这个参数，

等远程仓库里面有了内容之后，下次再从本地库上传内容的时候只需下面这样就可以了：

**$ git push origin master**  
  
上传项目的过程可能需要等一段时间，完成之后是这样的：



创建新仓库的指令：

　　git init //把这个目录变成Git可以管理的仓库

　　git add README.md //文件添加到仓库

　　git add . //不但可以跟单一文件，还可以跟通配符，更可以跟目录。一个点就把当前目录下所有未追踪的文件全部add了

　　git commit -m "first commit" //把文件提交到仓库

　　git remote add origin git@github.com:wangjiax9/practice.git //关联远程仓库

　　git push -u origin master //把本地库的所有内容推送到远程库上

安装git ,上传项目参考链接：https://www.cnblogs.com/specter45/p/github.html

## 参考网站：

<https://www.liaoxuefeng.com/wiki/896043488029600/896954117292416>

命令参考：

https://www.ruanyifeng.com/blog/2015/12/git-cheat-sheet.html