

Git教程

git是用于Linux内核开发的版本控制工具。与CVS、Subversion一类的集中式版本控制工具不同，它采用了分布式版本库的作法，不需要服务器端软件，就可以运作版本控制，使得源代码的发布和交流极其方便。git的速度很快，这对于诸如Linux内核这样的大项目来说自然很重要。git最为出色的是它的合并追踪（merge tracing）能力。实际上内核开发团队决定开始开发和使用git来作为内核开发的版本控制系统的时候，世界上开源社区的反对声音不少，最大的理由是git太艰涩难懂，从git的内部工作机制来说，的确是这样。但是随着开发的深入，git的正常使用都由一些友善的命令稿来执行，使git变得非常好用。[\[引用Git-wikipedia\]](#)

Git安装和配置

在Linux系统安装Git，首先你可以试着输入git，看系统有没有安装Git：

```
$ git
The program 'git' is currently not installed. You can install it by typin
g:
sudo apt-get install git
```

如果系统没有安装Git会友好的提示如上所示，你只需要通过 `sudo apt-get install git` 命令直接安装Git。

第一次使用Git需要配置本地用户和邮箱，其作用是用来上传本地仓库到GitHub中，在GitHub中显示代码上传者，使用如下命令：

```
git config --global user.name "JoshYang" //设置用户名
git config --global user.email "13241153187@163.com" //设置邮箱
```

Git获取和创建项目

什么是版本库呢？版本库又名仓库，英文名repository，你可以简单理解成一个目录，这个目录里面的所有文件都可以被Git管理起来，每个文件的修改、删除，Git都能跟踪，以便任何时刻都可以追踪历史，或者在将来某个时刻可以“还原”。

git init

创建一个版本库非常的简单，第一步在系统中选择一个合适的地方创建一个空目录

```
$ mkdir learngit
$ cd learngit
$ pwd
/home/yangxiong/learngit
```

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 /home/yangxiong/learnngit/.git/
```

创建完成后，在当前目录下输入ls -ah命令可以看到一个.git的目录，这个目录是Git来跟踪管理版本库的。

远程仓库

我们只需要注册一个GitHub账号，就可以免费获得Git远程仓库。由于你的本地Git仓库和GitHub仓库之间的传输是通过SSH加密的，所以需要一点设置：

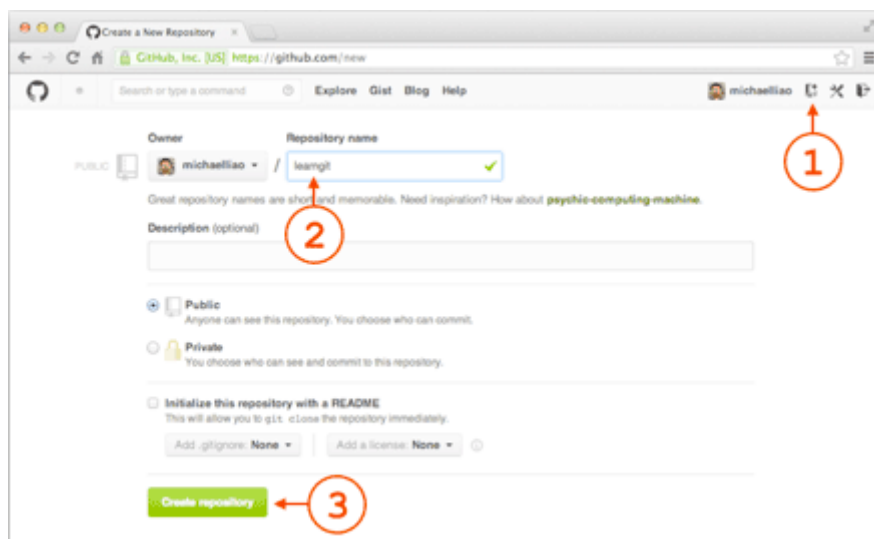
第一步，创建SSH Key。在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下有没有id_rsa和id_rsa.pub这两个文件，如果已经有了，可直接跳到下一步。如果没有，创建SSH Key命令如下（邮件地址换成你自己的邮件地址，然后一路回车，使用默认值即可）：

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

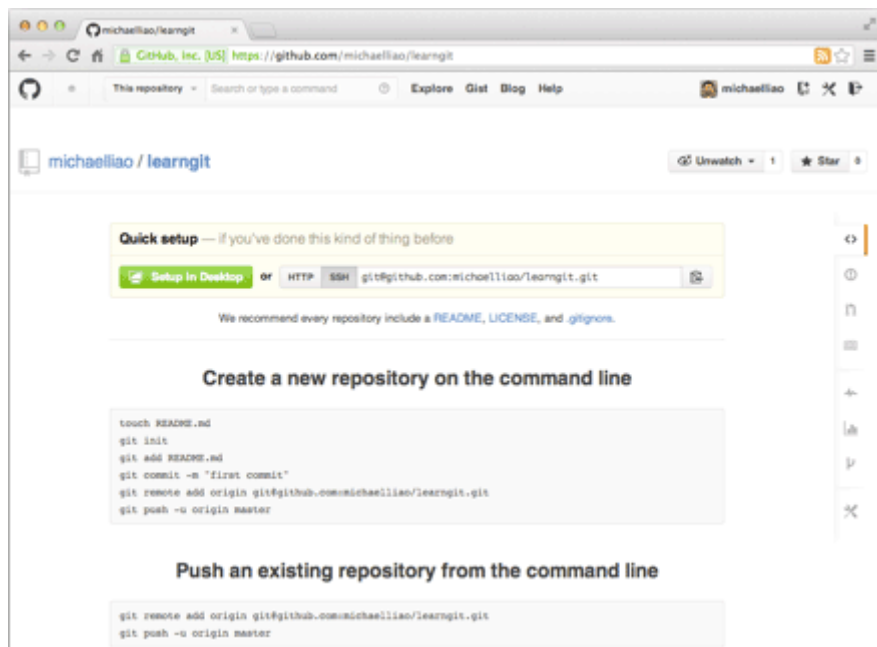
第二步，登录GitHub，找到个人settings项点击后，再找到SSH and GPG keys项，最后点击右上角的New SSH key按钮，按照要求填写Title和Key，其中Key文本框里粘贴id_rsa.pub文件的内容。如果多人同时维护远程仓库也可以添加多个SSH key。

在GitHub创建一个Git仓库，实现与本地仓库进行远程同步。

第一步，登录GitHub，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learnngit，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learngit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learngit仓库下运行命令：

```
$ git remote add origin git@github.com:yangxiong868/learngit.git
```

第二步，就可以把本地库的所有内容推送到远程库上：

```
$ git push -u origin master
Counting objects: 19, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (19/19), done.
Writing objects: 100% (19/19), 13.73 KiB, done.
Total 23 (delta 6), reused 0 (delta 0)
To git@github.com:michaelliao/learngit.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

使用命令 `git push -u origin master` 第一次推送master分支的所有内容；此后，每次本地提交后，只要有必要，就可以使用命令 `git push origin master` 推送最新修改；

git clone

GitHub远程仓库克隆到本地命令如下：

```
git clone git@github.com:yangxiong868/gitskills.git
Cloning into 'gitskills'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.

$ cd gitskills
$ ls
README.md
```

注意：GitHub给出的地址不止一个，还可以用<https://github.com/yangxiong868/gitskills.git>这样的地址。实际上，Git支持多种协议，默认的git://使用ssh，但也可以使用https等其他协议。

Git基本快照

- add
- status
- diff
- commit
- reset
- rm
- mv

Git分支与合并

- branch
- checkout
- merge
- mergetool
- log
- stash
- tag

Git共享和更新项目

- fetch
- pull
- push
- remote

Git检查与比较

- show
- log
- diff
- shortlog
- describe

参考链接：

1. [Git官网](#)
2. [Git用户手册](#)
3. [Git维基百科](#)
4. [廖雪峰Git教程](#)
5. [Git从入门到精通](#)