


# 如何在 ubuntu 下使用 Github

github (2) 

版权声明：本文为博主原创文章，如需转载，请注明出处：[http://blog.csdn.net/tina\\_ttl](http://blog.csdn.net/tina_ttl)

目录 [\(?\)](#) [\[+\]](#)

## 安装 Git

一个全新的 ubuntu 系统，需要安装 [Git](#)（系统是不具有该工具的），方法如下：  
在 terminal 中输入如下命令：

```
sudo apt-get install git git-core git-gui git-doc git-svn git-cvs gitweb gitk git-email  
git-daemon-run git-el git-arch
```

### 接下来需要检查 SSH

因为 GitHub 会用到 SSH，因此需要在 shell 里检查是否可以连接到 GitHub：

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

如果看到：

```
Warning: Permanently added 'github.com, 204.232.175.90' (RSA) to the list of known  
hosts.  
Permission denied (publickey).
```

则说明可以连接。

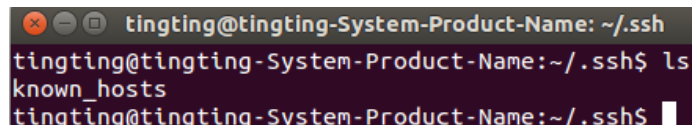
这里假设你已经就有了 GitHub 用户（如果没有，需要去注册 GitHub）

## 安装 SSH keys

在安装 GitHub 之前，需要先安装 SSH keys

**第一步：**检查是否已经具有 ssh keys，如果已经具有，则进行第二步，否则，进行第三步

```
cd ~/.ssh  
ls
```



```
tingting@tingting-System-Product-Name: ~/.ssh  
tingting@tingting-System-Product-Name: ~/.ssh$ ls  
known_hosts  
tingting@tingting-System-Product-Name: ~/.ssh$
```

查看该目录下是否已经具有 ssh keys，发现并没有 id\_rsa（私钥）和 id\_rsa.pub（公钥）这两个文件

**第二步：**备份并移除已经存在的 ssh keys

```
mkdir key_backup  
cp id_rsa* key_backup
```

```
rm id_rsa*
```

即将已经存在的 id\_rsa, id\_rsa.pub 文件备份到 key\_backup 文件夹

**第三步：**执行如下命令（不具有 ssh keys 时）：

```
ssh-keygen -t rsa -C "你自己的 github 对应的邮箱地址"
```

注 1：“”是需要的

得到结果如下：

```
tingting@tingting-System-Product-Name:~/.ssh$ ssh-keygen -t rsa -C "tingtingspringweb@163.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tingting/.ssh/id_rsa): id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa.
Your public key has been saved in id_rsa.pub.
The key fingerprint is:
80:50:fe:0d:27:ea:e1:1d:bc:55:01:11:44:6f:c0:8d tingtingspringweb@163.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      ... +B*      |
|      o .  Eo..    |
|      o + . +      |
|      + * o        |
|      o + S        |
|      o o +        |
|      o o          |
|                   |
+-----+

```

发现，id\_rsa（私钥）和 id\_rsa.pub（公钥）这两个文件被创建了  
（通过 ls 查看 ~/.ssh 下面的所有内容查看）

**第四步：**将刚刚创建的 ssh keys 添加到 github 中

- （1）利用 gedit/cat 命令，查看 id\_rsa.pub 的内容
- （2）在 GitHub 中，依次点击 Settings -> SSH Keys -> Add SSH Key，将 id\_rsa.pub 文件中的字符串复制进去，注意字符串中没有换行和空格。

**第五步：**再次检查 SSH 连接情况（在 ~/.ssh 目录下）：

输入如下命令：

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

- 1
- 1

如果看到如下所示，则表示添加成功：

```
Hi alioth310! You've successfully authenticated, but GitHub does not provide shell access.
```

此时，发现 github 上已有了 SSH keys

注 1：之前在设置公钥时如果设置了密码，在该步骤会要求输入密码，那么，输入当时设置的密码即可。

注 2：通过以上的设置之后，就能够通过 SSH 的方式，直接使用 [git](#) 命令访问 GitHub 托管服务器了

---

# 开始使用 github

参考[廖雪峰 github 教程](#)；[Github 简明教程](#)；[Linux 操作 Git 远程仓库与本地仓库同步的教程](#)；

## 配置 git

即利用自己的用户名和 email 地址配置 git

```
git config --global user.name "你的 github 用户名" git config --global user.email "你的 github 邮箱地址"
```

### • 1 如何推送本地内容到 github 上新建立的仓库

#### github 上新建立仓库

具体内容不做介绍，假设，新建的仓库为 dockerfiles

#### 在本地建立一个目录

该目录名称与 github 新建立的目录相同，假设本地目录为~/Document/dockerfiles

#### 本地仓库初始化

```
cd ~/Document/dockerfiles
git init
```

### • 2 对本地仓库进行更改

例如，添加一个 Readme 文件

```
touch Readme
```

### • 1 对刚刚的更改进行提交

该步不可省略！

```
git add Readme
git commit -m 'add readme file'
```

## push

首先，需要将本地仓库与 github 仓库关联

注：<https://github.com/>你的 github 用户名/你的 github 仓库.git 是 github 上仓库的网址

```
git remote add origin https://github.com/你的 github 用户名/你的 github 仓库.git
```

然后，push，此时，可能需要输入 github 账号和密码，按要求输入即可

```
git push origin master
```

## 如何推送本地内容到 github 上已有的仓库

**注：经验证，git clone; git commit; git push 即可**

从 github 上将该仓库 clone 下来

```
git clone https://github.com/你的github用户名/github仓库名.git
```

对 clone 下来的仓库进行更改

例如，添加一个新的文件

```
touch Readme_new
```

对刚刚的更改进行提交

该步不可省略！（其实是提交到 git 缓存空间）

```
git add Readme_new  
git commit -m 'add new readme file'
```

### push

首先，需要将本地仓库与 github 仓库关联

注：<https://github.com/>你的 github 用户名/你的 github 仓库.git 是 github 上仓库的网址

```
git remote add origin https://github.com/你的github用户名/你的github仓库.git
```

有时，会出现 fatal: remote origin already exists.，那么，需要输入 git remote rm origin 解决该问题

然后，push，此时，可能需要输入 github 账号和密码，按要求输入即可

```
git push origin master
```

注：有时，在执行 git push origin master 时，报错：error:failed to push som refs to.....，那么，可以执行

```
git pull origin master
```

至此，github 上已有的仓库的便有了更新

---

如果需要添加文件夹，有一点需要注意：该文件夹不能为空！否则不能成功添加

---

# 操作命令小结

- 克隆 github 上已有的仓库

```
git clone https://github.com/你的 github 用户名/github 仓库名.git
```

- 或者是在 github 上新建仓库并且在本地新建同名的仓库

```
cd ~/Document/dockerfiles  
git init
```

- 对本地仓库内容进行更改（如果是多次对本地的某个仓库进行这样的操作，直接从此步开始即可，不要前面的操作了，因为本地仓库已有具有了 github 仓库的.git 文件了）
- 对更改内容进行提交

```
git add 更改文件名或者是文件夹名或者是点".  
git commit -m "commit 内容标注"
```

- 本地仓库与 github 仓库关联

```
git remote add origin https://github.com/你的 github 用户名/你的 github 仓库.git
```

- push

```
git push origin master
```

注：另外可能用到的命令

```
git remote rm origin  
git pull origin master
```