

# Linux下使用git的方法

使用git主要是为了方便管理以及在虚拟机下上传文件的问题，由于linux系统本身以终端处理为主的特点，所以我觉得使用git来进行管理十分重要且合适，下面介绍的是linux下使用git并绑定GitHub仓库的办法，同时涉及文件的上传和下载删除操作。

第一次使用GitHub一定有一些烦恼，只要你成功创建了账号就算成功

当你找到本仓库(repository)的时候，请确保你同时打开虚拟机或者云主机并在Linux下打开终端(terminal)

## 一、Linux下Git的配置

首先在终端中运行下面的指令安装git

```
sudo apt-get install git
```

当安装完成后运行

```
git version
```

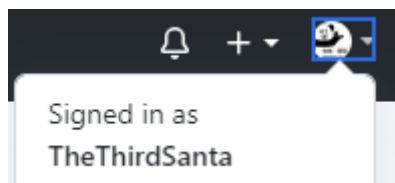
验证是否安装成功

```
jerry@ubuntu:~/Desktop$ git version  
git version 2.25.1
```

之后需要进行配置用户名和用户邮件

```
git config --global user.name "用户名"  
git config --global user.email "用户邮件"
```

用户名就是你自己在GitHub里面的名字，比如我是“TheThirdSanta”，用户邮件就是注册GitHub是用的邮件地址



输入后运行

```
git config --list
```

验证是否配置成功

```
jerry@ubuntu:~/Desktop$ git config --list  
user.name=TheThirdSanta  
user.email=787930211@qq.com
```

这里的`color.ui = auto`指的是终端默认的彩色模式打开，你要是觉得花里胡哨你不喜欢，可以自行关闭之后进行最重要的一步，**配置git的私钥和公钥**，如果没有这一步你的git没有办法和你的GitHub上的仓库联动，也无法从GitHub上下载或者上传任何文件到仓库中!!!

首先运行下面命令生成密钥

```
ssh-keygen -t rsa -C "用户邮箱"
```

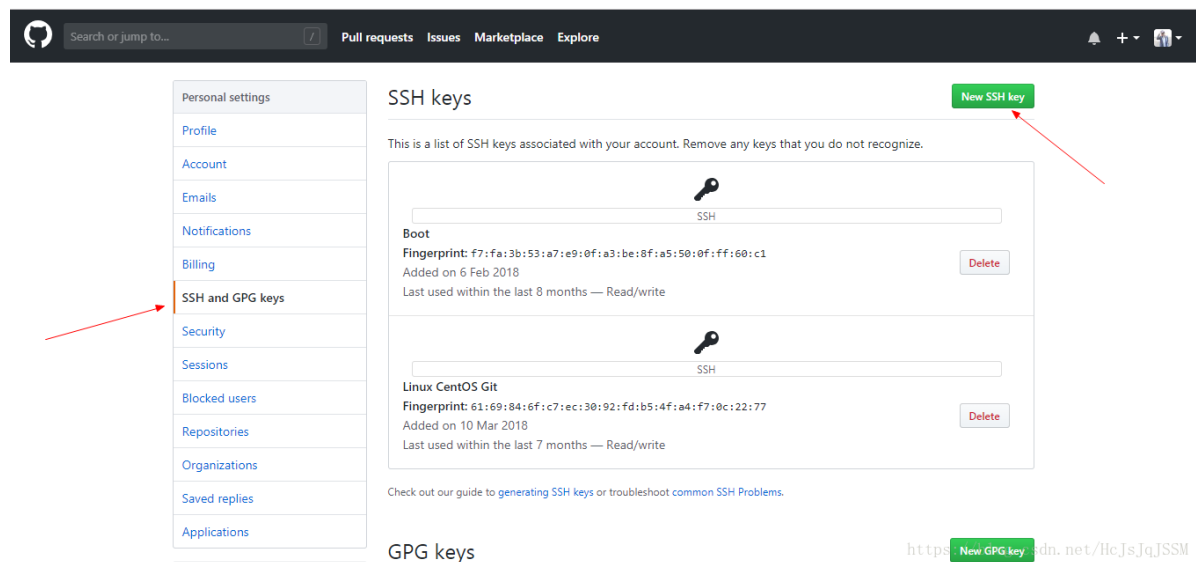
之后的界面会有多个Enter，你只需要无脑按Enter就可以了，之后便是将生成的密钥文件添加到GitHub中去

运行

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

查看是否包含一个名为`id_rsa.pub`的文件，如果存在那么进行下一步，如果不存在找我!

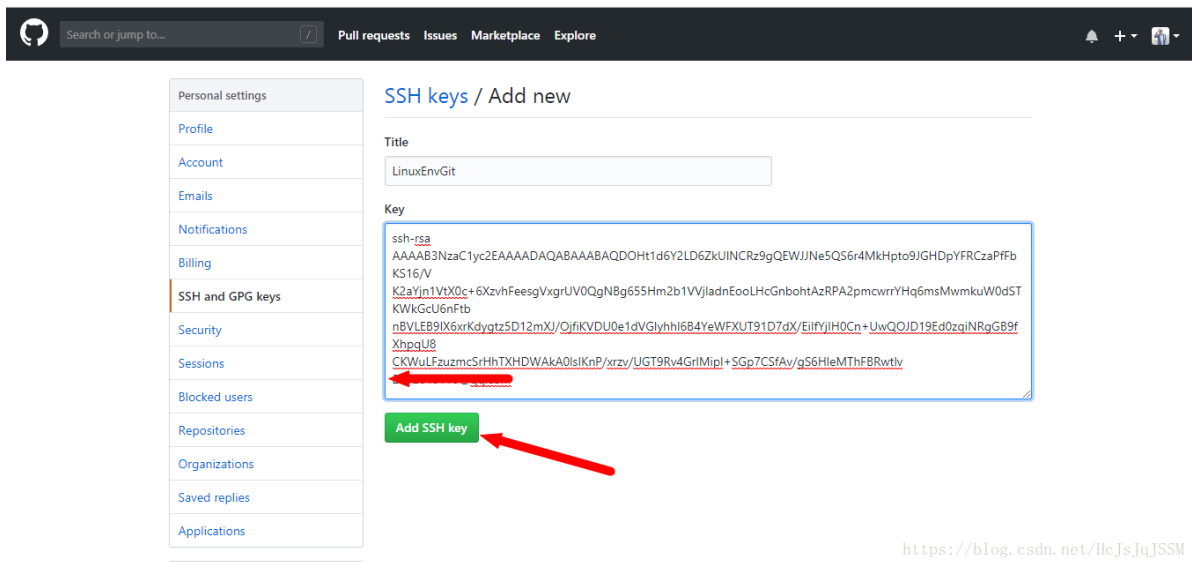
之后点击GitHub网站，点击个人的**setting**，选择选项**SSH and GPG keys**，新建一个SSH key



命名你的SSH Key，同时在linux中打开`id_rsa.pub`文件

```
cat id_rsa.pub
```

将打开后的所有内容复制到GitHub中Key的框框里面，之后点击添加即可



回到linux中，验证Git是否已经成功配置

```
ssh git@github.com
```

出现下面的情况就说明安装配置成功

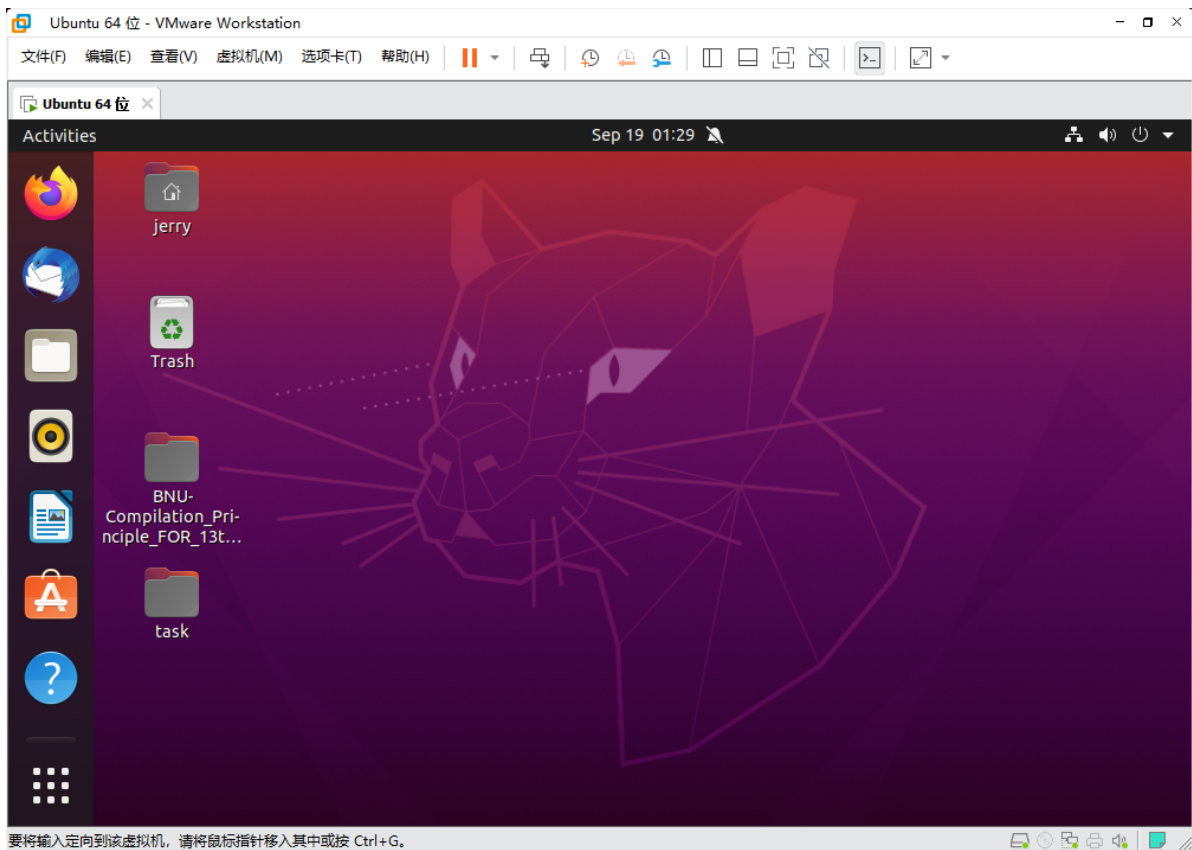
```
jerry@ubuntu:~/Desktop$ ssh git@github.com
PTY allocation request failed on channel 0
Hi TheThirdSanta! You've successfully authenticated, but GitHub does not provide
shell access.
Connection to github.com closed.
```

## 二、将库添加到本地并上传或者删除文件

在仓库中点击绿色的Code按钮，将Clone下面的地址复制下来，之后在终端中运行（好吧我已经帮你复制好了）

```
git clone https://github.com/TheThirdSanta/BNU-
Compilation_Principle_FOR_13thGroup.git
```

这时会将库中文件下载到当前的目录，默认为桌面上，你可以在桌面上看到一个以库名命名的文件



可以在打开该文件夹查看里面的内容，和GitHub仓库中的内容以及分级一模一样，这样就成功将文件下载到本地

之后写完文件之后（可以是在当前的目录下，也可以是在其他目录下），记得一定要在终端中将文件移动到指定的位置，并运行上传一个指定的文件

```
git add 文件名
```

或者上传改目录下的所有文件

```
git add .
```

上传之后这时git会将这些文件暂时上传至缓冲区内，等待你下一步的确认上传操作

所有等待上传的文件均已添加到缓冲区后，运行

```
git commit -m "这里是注释"
```

注释内主要写你对仓库的修改

之后再运行

```
git push origin main
```

这里会将所有打好注释的缓冲区内容上传至GitHub的仓库中，这里会要求你填写GitHub的账户名称和密码，但是你填写密码后他会告诉你

```
jerry@ubuntu:~/Desktop/BNU-Compilation_Principle_FOR_13thGroup/Task1$ git push origin main
Username for 'https://github.com': TheThirdSanta
Password for 'https://TheThirdSanta@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/TheThirdSanta/BNU-Compilation_Principle_FOR_13thGroup/'
```

这是因为GitHub从2021.8.13之后因为安全原因不再支持本地密码输入管理远程仓库了，这时你需要填写个人的token来代替密码，配置个人token查看下面的部分，之后进行这一步操作都需要将密码替换为token

删除文件的操作也十分简单，运行下面命令即可


```
git rm FileName
git rm -r FolderName
```

上面一条命令是删除单个文件，下面一条命令是删除整个文件夹




运行完之后一样需要运行commit命令和push命令

### 三、配置个人token


在设置里面选择最下面的**Developer settings**,选择**Personal access tokens(PAT)**,选择新建一个token





PullsIssuesMarketplaceExplore



Settings / Developer settings

 GitHub Apps

 OAuth Apps

 Personal access tokens

## New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

What's this token for?

**Expiration \***

30 days

 The token will expire on Wed, Oct 19 2022

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects

**Note**中填写你关于这个token的用途，下面的**Expiration**选择Custom，设置一个尽可能久远的日期

下面的Select scopes中将前四个大类勾上：repo workflow write... delete...

点击下面的生成按钮，之后会生成一个token

**!!! 注意!!! 这里生成的token网页一旦关闭，则无法再看到，请一定要在这里选择复制token内容并妥善保存**

Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved. X

Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

ghp\_0BJHz8XNNVDF8sX1cr1JHA4jXs2FTd4E1rtp

Delete

GitForLinux — delete:packages, repo, write:packages

Last used within the last week

Delete

Expires on Sun, Jan 1 2023.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Terms

Privacy

Security

Status

Docs

Contact GitHub

Pricing

API

Training

Blog

About

© 2022 GitHub, Inc.

之后再进行push操作的时候输入改token作为密码即可

至此初步的环境配置全部完成，希望接下来的一个学期内我们合作愉快 😊