

Git的使用

1, 什么是Git

Git是目前世界上最先进的分布式版本控制系统。



2, 安装Git

为了验证是否安装了Git, 可以输入以下命令: 如果显示Git相关的命令行则说明你的电脑已经安装了Git

```
$git
```

3, 使用Git

3.1, 创建版本库(仓库/repository)

3.1.1, 选择一个合适的目录来创建仓库(这里选择的目录是桌面)

```
$cd Desktop/
```

3.1.2, 使用mkdir命令创建仓库

```
$mkdir gitname
```

3.1.3, 进入该仓库目录

```
$cd gitname
```

3.1.4, 将该git目录变成Git可以管理的仓库:

```
$git init
```

3.2, 提交文件到仓库

3.2.1, 将一个需要提交到版本库的文件readme放到gitname目录下;

3.2.2, 使用git add命令告诉Git, 把文件添加到仓库

```
$git add readme.txt
```

3.2.3, 使用git commit命令告诉Git, 把文件提交到仓库(后面是log信息)

```
$git commit -m "add new file readme.txt"
```

3.3, 修改文件内容

3.3.1, 修改`readme.txt`的内容:

3.3.2, 使用`git status`查看状态(文件是否有修改):

```
$git status
```

3.3.3, 使用`git diff`查看修改前后的区别:

```
$git diff
```

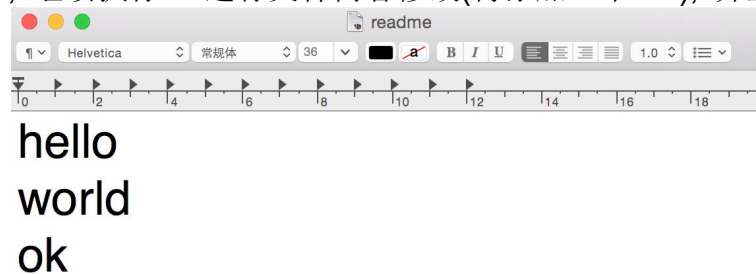
3.3.3, 使用`git add`和`git commit`再次提交:

```
$git add readme.txt
```

```
$git commit -m "commit changes readme.txt"
```

3.4, 版本回退

3.4.1, 继续执行3.3进行文件内容修改(再添加一个'ok'), 并且提交修改:



3.4.2, 使用`git log`查看历史日志(或者使用:`git log --pretty=oneline`):

```
$git log
```

3.4.3, 在Git中, `HEAD`表示当前版本, 也就是最新版本, 上一个版本使用`HEAD^`, 上上个版本使用`HEAD^^`, 往上100个版本使用`HEAD~100`, 然后使用`git reset`命令回退:

```
$git reset --hard HEAD^
```

3.4.4, 上面的方式是回退, 如果又想继续回到之前的高版本怎么办呢, `git`也是有办法的, 我们只要找到该高版本的`commit id`, 再执行以下命令即可:(`commit id`只要写前部分即可)

```
$git reset --hard 2044be7c
```

3.4.5, 使用`git reflog`可以查看你执行过的命令记录

```
$git reflog
```

(Git的版本回退速度非常快, 因为Git在内部有个指向当前版本的HEAD指针, 当你回退版本的时候, Git仅仅是把HEAD从指向了回退后的版本)

3.5, 工作区和暂存区

<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013745374151782eb658c5a5ca454eaa451661275886c6000>

4, GitHub创建远程仓库

4.1, 登录<https://github.com/> 注册GitHub账号,并登录;

4.2, 创建SSH Key;

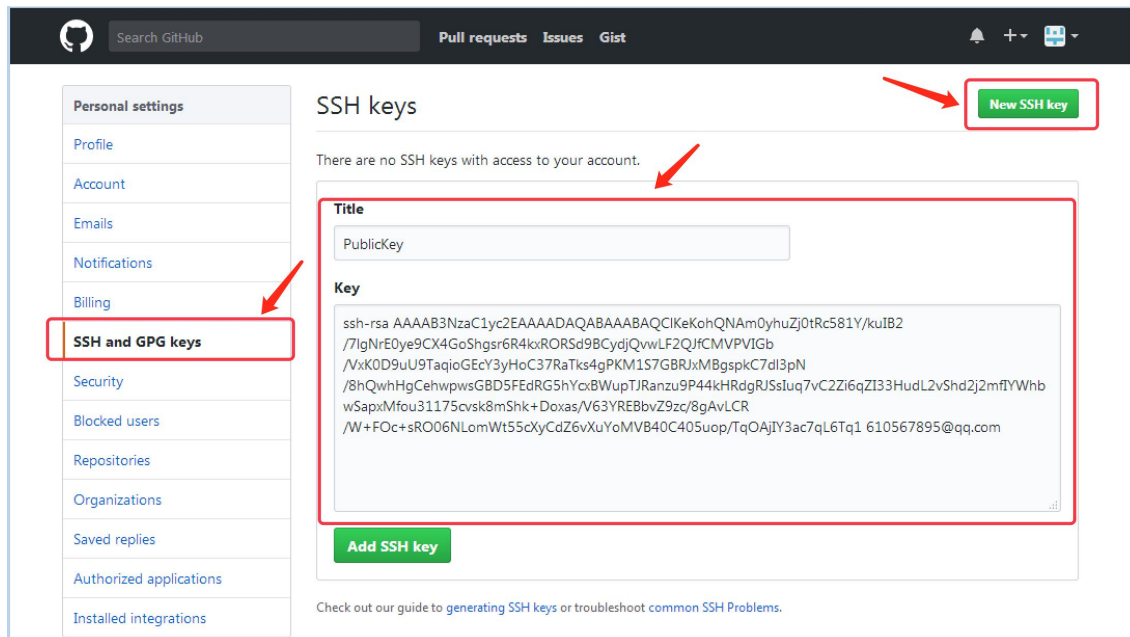
4.2.1, 生成rsa私钥和公钥(邮箱地址写自己的邮箱, 然后一路按回车)

```
$ssh-keygen -t rsa -C "610567895@qq.com"
```

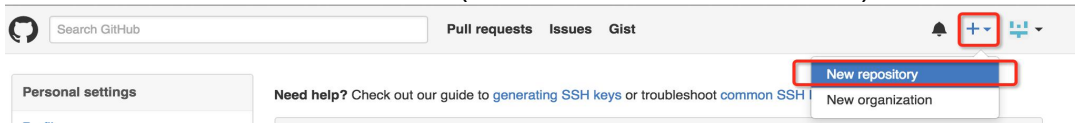
4.2.2, 进入电脑用户的目录查看私钥和公钥(把ijeff换成你电脑的用户名)

```
$cd /c/Users/ijeff/.ssh/
```

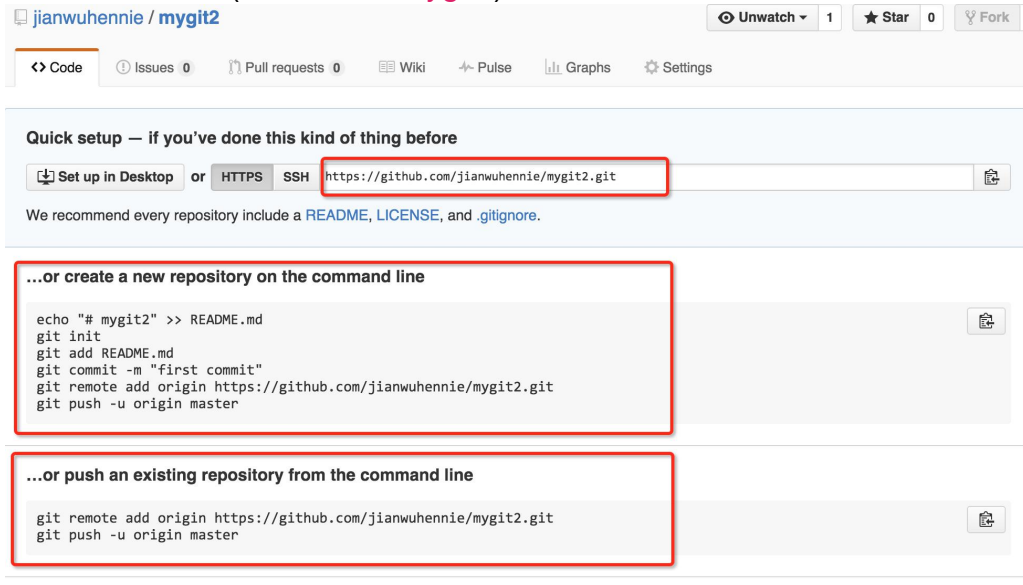
4.3, 在GitHub上添加SSH Key:<https://github.com/settings/ssh>, 将公钥id_rsa.pub里面的值添加:



4.4, 在GitHub上添加远程仓库(填入仓库名称,其他默认即可)



添加之后的仓库路径(仓库名称为mygit2):



4.5, 将GitHub远程仓库和本地仓库关联, 并将本地仓库推送(push)到远程仓库;

4.5.1, 在终端中使用cd命令进入本地的仓库目录下

4.5.2, 远程仓库和本地仓库建立关联; 根据上面4.4的提示, 在本地的仓库目录下执行以下命令:(其中origin是GitHub远程仓库的名称, jianwuhennie是GitHub的用户名)

```
$git remote add origin https://github.com/jianwuhennie/mygit2.git
```

```
或者 $git remote add origin git@github.com:jianwuhennie/mygit2.git
```

(以上两种写法都可以, git协议和https协议, git协议较快)

4.5.3, 将本地仓库内容推送到远程仓库;(其中master是仓库下的分支, 第一次push 需要加上-u参数) 执行以下命令来进行推送(push), 过程中可能需要输入GitHub的用户名和密码

```
$git push -u origin master
```

以后push就不需要再加-u参数

```
$git push origin master
```

添加后可以在GitHub查看远程仓库的内容

The screenshot shows the GitHub interface for a repository named 'mygit2' owned by 'jianwuhennie'. A red arrow points to the repository name. The repository has 2 commits, 1 branch, and 0 releases. The 'Code' tab is selected. Below the repository header, there is a commit history table. The first commit is highlighted with a red box, showing the file 'readme.rtf' was added. Below the commit history, the content of 'readme.rtf' is displayed, which is a raw text file containing escape sequences for a terminal display.

```
{\rtf1\ansi\ansicpg936\cocoartf1348\cocoasubrtf170
{\fonttbl{\f0\fswiss\fcharset0 Helvetica;}}
{\colortbl;\red255\green255\blue255;}
\paperw11900\paperh16840\marg11440\margin1440\vieww10800\viewh8400\viewkind0
\pard\tx566\tx1133\tx1700\tx2267\tx2834\tx3401\tx3968\tx4535\tx5102\tx5669\tx6236\tx6803\pardirnatural

\fs72 \cf0 hello\
world\
ok}
```

4.6, 先在GitHub先创建远程仓库, 再将远程仓库克隆(clone)到本地仓库;

4.6.1, 先在GitHub先创建远程仓库mygit3,同时勾选创建时带一个README.md文件

Create a new repository

A repository contains all the files for your project, including the revision history.

The screenshot shows the 'Create a new repository' form. The 'Owner' is 'jianwuhennie'. The 'Repository name' is 'mygit3', which is highlighted with a red box and a green checkmark. Below the repository name, there is a note: 'Great repository names are short and memorable. Need inspiration? How about propitious-capsicum.' The 'Description (optional)' field is empty. The 'Public' option is selected, with a note: 'Anyone can see this repository. You choose who can commit.' The 'Private' option is also visible. The 'Initialize this repository with a README' checkbox is checked and highlighted with a red box. Below this, there is a note: 'This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.' At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None'. A green 'Create repository' button is at the bottom.

4.6.2, 再将远程仓库克隆(clone)到本地仓库

4.6.2.1, 在终端中使用cd命令进入你需要创建本地仓库的指定目录

4.6.2.2, 执行clone命令:

```
$git clone https://github.com/jianwuhennie/mygit3.git
```

或者

```
$git clone git@github.com:jianwuhennie/mygit3.git
```

如有以下警告,则输入yes

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

5, Git分支管理及其他(了解)