**Git环境搭建**

# 服务器端搭建

Git的公网远程托管服务器有<https://github.com/>（外国），<http://www.oschina.net/question/tag/github>（开源中国），但是这些公网的远程托管都是源码公开，如果不公开是要另外收费的，因此如果不希望源码被公开而且又不想另外缴费的情况，就要自己搭建git服务器。

## 1.1 Centos6或centos7下安装

在CentOS5的时代，由于yum源中没有git，所以需要预先安装一系列的依赖包。但在CentOS6的yum源中已经有git的版本了，可以直接使用yum源进行安装。

$ sudo yum install git

但是yum源中安装的git版本是1.7.1，太老了，Github等需要的Git版本最低都不能低于1.7.2 。所以我们一般不用上面的方法。而是下载git源码，编译安装。

（1）首先先更新系统

sudo yum update

（2）安装依赖的包

sudo yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel gcc perl-ExtUtils-MakeMaker

（3）下载git源码并解压缩（本人下载的是比较新的v2.10.2）

wget https://github.com/git/git/archive/v2.10.2.zip

unzip v2.10.2.zip

cd git-2.10.2

（4）编译安装

将其安装在“/usr/local/git”目录下。

make prefix=/usr/local/git all

sudo make prefix=/usr/local/git install

（5）此时你如果使用git --version 查看git版本的话，发现git仍然是1.7.1版本。这是因为它默认使用了"/usr/bin"下的git。

你可以用下面的命令查看git所在的路径：

$ whereis git

git: /usr/bin/git /usr/local/git /usr/share/man/man1/git.1.gz

（6）我们要把编译安装的git路径放到环境变量里，让它替换"/usr/bin"下的git。为此我们可以修改“/etc/profile”文件（或者/etc/bashrc文件）。

sudo vim /etc/profile

然后在文件的最后一行，添加下面的内容，然后保存退出。

export PATH=/usr/local/git/bin:$PATH

（7）使用source命令应用修改。

source /etc/profile

（8）然后再次使用git --version 查看git版本，发现输出2.3.0，表明安装成功。

（9） server服务器安装openssh

sudo yum install openssh-server

（10）最好将创建一个用户专门管理git

//新建用户

sudo adduser gituser

//修改密码

passwd gituser

//连续输入两次密码

（11）切换到gituser下创建Git仓库

su gituser

（12）先选定一个目录作为Git仓库，如我自己创建的git-workspace(sudo mkdir git-workspace 进入git-workspace)

sudo git init --bare sample.git

sudo git init --bare sample2.git

同时创建两个库，而且互不干涉

(--bare 参数就表示初始化 Git 仓库的时候不要创建本地工作目录，所以相当于 .git 文件夹下的所有内容直接创建到当前目录下，而不是被放到 .git 目录下)

Git就会创建一个裸仓库，裸仓库没有工作区，因为服务器上的Git仓库纯粹是为了共享，所以不让用户直接登录到服务器上去改工作区，并且服务器上的Git仓库通常都以.git结尾。然后，把owner改为git：

# 客户端配置

客户端git安装方式和服务器端的一致，请参考服务器端（1）到（8）的步骤。

# 客户端连接到公网服务器

## 3.1 设置Git

**如果自己搭建服务器端而非连到githut上，此步可流程图**

（1）设置用户名和email。

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

git config --global user.email "youremail@domain.com"

此时，Home目录下会新建一个.gitconfig文件

## 3.2 为GitHub账号添加SSH Keys

以公钥认证方式访问SSH协议的Git服务器时无需输入口令，而且更安全。（访问HTTP协议的Git服务器时，比如提交修改，每次都需要输入口令。）

（1）创建SSH key

$ ssh-keygen -t rsa -C "youremail@163.com"

系统会提示key的保存位置（一般是~/.ssh目录）和指定口令，保持默认，连续三次回车即可。

（2）Copy SSH Key

**如果是自己搭建服务器，非连到githug上，此步可不用**

然后用vim打开该文件，id\_rsa.pub文件内的内容，粘帖到github帐号管理的添加SSH key界面中。

vim ~/.ssh/id\_rsa.pub

（3）添加到GitHub

**如果是自己搭建服务器，非连到githug上，此步可不用**

登录github-> Accounting settings图标-> SSH key-> Add SSH key-> 填写SSH key的名称（可以起一个自己容易区分的），然后将拷贝的~/.ssh/id\_rsa.pub文件内容粘帖-> add key”按钮添加。

（4）测试

ssh -T [git@github.com](mailto:git@github.com)

## 3.3 为GitHub上的Repository提交修改

**如果是自己搭建服务器，非连到githug上，此步可不用**

（1）git clone已存在GitHub上的Repository。（在新建的~/MyTestFolder目录中）

git clone https://github.com/zhchnchn/ZhchnchnTest.git

（2）修改一个文件，然后提交

vim README.md

git status

git add README.md

git status

git commit -m "Edit by WorkUbuntu 1204"

git status

git remote add origin https://github.com/zhchnchn/ZhchnchnTest.git

这时会报错误：

fatal: remote origin already exists.

解决办法【3】：

$ git remote rm origin

然后再次执行 git remote add origin https://github.com/zhchnchn/ZhchnchnTest.git

（3）之后，需要将修改push到GitHub上

git push -u origin master

执行该条命令后，会要求输入GitHub账户的用户名和密码。

（4）提交完成后，查看GitHub上的Repository，会发现内容修改成功。

# SSH方式登录Git服务器

SSH方式，避免每次连接到服务器的时候都要输入密码

1. **客户端创建SSH key**

ssh-keygen -t rsa -C “[youremail@163.com](mailto:youremail@163.com)”

系统会提示key的保存位置（一般是~/.ssh目录）和指定口令，保持默认，连续三次回车即可。同时生成公匙和私匙

1. **在客户端.ssh目录中查到公匙id\_rsa.pub，将其上传到服务器端，其中gituser为自己在服务器端创建的用户名，192.168.0.103为服务器地址也可以改成服务器名称**

**scp id\_rsa.pub** [**gituser@192.168.0.103:/tmp**](mailto:gituser@192.168.0.103:/tmp)

1. 服务器端登陆gituser，有可能没有.ssh，创建此文件

gituser@Linux:~$ mkdir .ssh

gituser @Linux:~$ touch .ssh/authorized\_keys

gituser @Linux:~$ chmod 600 .ssh/authorized\_keys

gituser @Linux:~$

cat /tmp/id\_rsa.pub >> ~/.ssh/authorized\_keys

1. 打开文件sudo vim/etc/ssh/sshd\_config

RSAAuthentication yes            #开启RSA认证功能

PubkeyAuthentication yes      #开启公匙认证

StricModes no                          #据说不改会强制要求登录用户和文件拥有者用户相同

AuthorizedKeysFile      .ssh/authorized\_keys  
AuthorizedKeysCommand none  
AuthorizedKeysCommandRunAs nobody

1. 重启服务

service   sshd   restart