**CentOS上搭建Git服务器步骤详解**

www.111cn.net 编辑:phper 来源:本站原创

CentOS上搭建Git是大家非常常用的一个分布式版本控制软件了，比起svn像是要强大了许多了，下面一聚教程小编就为各位介绍一下在CentOS上搭建Git服务器步骤吧。

Git是一个分布式版本控制软件，原来是[linux](http://www.111cn.net/list-156/" \t "_blank)内核开发者Linus Torvalds为了更好地管理linux内核开发而创立的。发展至今，Git已经成为了一个相当好用的版本管理工具。相比于SVN，如果想要保存一些微小的修改也必须得提交服务器保存才可以，这样使服务器的版本号过多，而Git解决了这个问题，一些小的修改只在本地提交即可，只需最后修改完成后再提交服务器。正是由于这样的便捷性，现在越来越多的社区项目都开始使用Git来取代SVN之类较为传统的版本管理工具进行开发。 。

使用[CentOS](http://www.111cn.net/list-232/" \t "_blank)搭建Git服务器是一件比较轻松的事儿，本次折腾主要涉及git, gitosis, gitweb的安装配置。其中，gitosis和gitweb是两种比较常用的方式，gitosis是以SSH方式访问和管理git， gitweb是通过http的方式访问和管理。利用这些工具即可满足Git服务器的基本功能。此外比较好的一点是，Git的管理工具几乎不会给服务器带来较大的性能压力。下面正式开始我们的Git安装配置记录。

一、安装Git

yum install git  
然后进行配置：

useradd --home /home/git git  
passwd git  
创建完用户后就可以切换到git用户下进行后面的设置，如用户名和邮箱：

su git  
git config --global user.name "somebody"  
git config --global user.email "somebody@example.com"  
设置默认将会保存在~/.gitconfig文件中。

此时，Git的功能就已经可以使用了。为了方便后面的操作，可以先来创建一个空版本库。

mkdir ~/repo  
然后建立项目目录

mkdir ~/repo/huhamhire-hosts  
切换到项目目录，并进行初始化

cd ~/repo/huhamhire-hosts  
git init -bare  
至此，一个初始的空项目版本库就配置完成了，后面安装了gitosis之后便可向库中推送我们的代码库内容。

二、安装gitosis

在安装之前，可以看一下gitosis的实现原理：

http://geeklu.com/2012/10/gitosis/

先切换回root权限。

su root  
并先安装[python](http://www.111cn.net/list-157/" \t "_blank)-setuptool

yum install python-setuptools  
然后开始安装gitosis，值得注意的是gitosis的安装程序本身就是由git管理的，需要使用git来获取。这里在/tmp目录下进行相关的安装操作：

cd /tmp  
git clone htt[ps](http://www.111cn.net/fw/photo.html)://github.com/res0nat0r/gitosis.git  
接下来进入下载的gitosis版本库进行安装:

cd gitosis  
python setup.py install  
安装完成后，便进入对gitosis的设置阶段。由于gitosis需要通过SSH进行管理，所以需要创建SSH密钥对，并将公钥放在服务器端，私钥放在客户端。一般的流程是客户端创建完密钥后，将公钥传到服务器上生效。不过，偷懒的话直接在服务器上操作问题也不大。

切换到git用户并建立文件夹.ssh：

su git  
mkdir /home/git/.ssh  
一定记得，在客户机上生成公钥，上传到服务器，或者在服务器上生成，下载到客户机。

进入~/.ssh目录并使用ssh-keygen生成公钥：

cd /home/git/.ssh  
ssh-keygen -t rsa  
注意不能忘记私钥的密码。默认会生成~/.ssh/id\_rsa.pub公钥文件。

有了密钥以后便可初始化gitosis，使gitosis获得对Git的管理权限：

gitosis-init < /home/git/.ssh/id\_rsa.pub  
初始化之后，会在/home/git/repositories创建gitosis-admin.git项目，可以通过维护这个项目来对gitosis进行配置。

除此以外，还需要对gitosis-admin.git/hooks/post-update目录赋上特殊权限：

chmod u+x /home/git/repositories/gitosis-admin.git/hooks/post-update  
至此，服务器端的gitosys配置就完成了。

三、设置并使用gitosys

在服务器端完成了gitosys的配置之后，便可在客户端进行接下来的设置，以便使用Git服务器。

较为正规的做法是在客户端通过gitosis-admin版本库做管理设置，之后提交到服务器使项目权限生效，当然也可以使用操作系统的ssh登录方式进行验证，不过这里仅介绍前面一种方法。

在进行以下操作时，需要确认一下，你的公钥是不是已经放在客户机~/.ssh/目录下。如果你也在用github的话，那么你需要设置一下多公钥共存的东西。.ssh/config，在这个文件中写入：

Host github.com  
    HostName github.com  
    User git  
    IdentityFile C:/Users/abc/.ssh/id\_rsa  
Host git.oschina.net  
    HostName git.oschina.net  
    User git  
    IdentityFile C:/Users/abc/.ssh/id\_rsa\_a  
Host abc.ueder.info  
    HostName abc.ueder.info  
    User git  
    Port 1000  
    IdentityFile C:/Users/abc/.ssh/id\_rsa\_new  
如我就使用了好几个git服务，每个都有自己的公钥，需要配置文件来区分开来，并且我自己的服务器ssh端口已经不是默认端口，需要在配置文件中声明，否则在每次clone的时候要声明端口。

在客户机上下载gitosis-admin版本库，这里以linux客户机为例：

git clone git@VPS的IP/Domain:/home/git/repositories/gitosis-admin.git  
获取完成后对gitosis-admin/gitosis.conf文件进行设置，以上面新建的项目为例，新增：

[group huhamhire-hosts]  
writable = huhamhire-hosts  
members = hamhire@myhost  
随后要将客户端的公钥放到keydir目录下，并随后提交设置到服务器：

cp ~/.ssh/id\_rsa.pub ~/gitosis-admin/keydir/hamhire@myhost.pub  
   
cd ~/gitosis-admin  
git add ./  
git commit -a -m "add new repo"  
git push  
由于之前已经在/home/git/repo/目录下设置了huhamhire-hosts的版本库位置，所以可以直接进行推送操作。

在本例中可以通过hamhire@myhost:/home/git/repo/huhamhire-hosts的路径来提交项目。

至此，gitosys的配置全部完成。

四、安装gitweb

在配置完成了git服务器以后，如果需要方便在线查看，使用gitweb来提供一个简单网页版的版本显示界面是一个不错的选择。

在centos 下安装gitweb如下：

yum install fcgi-devel  
   
cd /usr/local/src/  
git clone git://github.com/gnosek/fcgiwrap.git  
cd fcgiwrap  
autoreconf -i  
./configure  
make  
make install  
至此，fcgiwrap已经安装到 /usr/local/sbin/fcgiwrap

然后再安装spawn-fcgi

yum install spawn-fcgi

安装好后:

vim /etc/sysconfig/spawn-fcgi

修改文件为：

# You must set some working options before the "spawn-fcgi" service will work.  
# If SOCKET points to a file, then this file is cleaned up by the init script.  
#  
# See spawn-fcgi(1) for all possible options.  
#  
# Example :  
#SOCKET=/var/run/php-fcgi.sock  
#OPTIONS="-u [apache](http://www.111cn.net/list-121/) -g apache -s $SOCKET -S -M 0600 -C 32 -F 1 -P /var/run/spawn-fcgi.pid -- /usr/bin/php-cgi"  
FCGI\_SOCKET=/var/run/fcgiwrap.socket  
FCGI\_PROGRAM=/usr/local/sbin/fcgiwrap  
FCGI\_USER=[nginx](http://www.111cn.net/list-195/" \t "_blank)  
FCGI\_GROUP=nginx  
FCGI\_EXTRA\_OPTIONS="-M 0700"  
OPTIONS="-u $FCGI\_USER -g $FCGI\_GROUP -s $FCGI\_SOCKET -S $FCGI\_EXTRA\_OPTIONS -F 1 -P /var/run/spawn-fcgi.pid -- $FCGI\_PROGRAM"  
然后设置开机运行：

chkconfig --levels 2345 spawn-fcgi on  
/etc/init.d/spawn-fcgi start  
这里已经完成了fcgi的安装运行。如果你用的nginx，还需要对nginx.conf进行配置，才能将.cgi的请求转发给fcgiwrap.socket

location /cgi-bin/ {  
 # Disable gzip (it makes scripts feel slower since they have to complete  
 # before getting gzipped)  
 gzip off;  
 # Set the root to /usr/lib (inside this location this means that we are  
 # giving access to the files under /usr/lib/cgi-bin)  
 root /var/www/www.example.com;  
 # Fastcgi socket  
 fastcgi\_pass unix:/var/run/fcgiwrap.socket;  
 # Fastcgi parameters, [include](http://www.111cn.net/tags.php/include/) the standard ones  
 include /etc/nginx/fastcgi\_params;  
 # Adjust non standard parameters (SCRIPT\_FILENAME)  
 fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  
 }  
最后重启nginx就可以了。

五、Gitweb的界面美化与中文化

个人觉得Gitweb默认的用户界面是在是有点简陋地惨不忍睹，实在觉得不舒服的话，完全可以使用样式表进行适当的装饰

浅谈Gitosis实现原理

git作为一个分布式的版本控制系统,其实client端和server的差别并不是很大，只是server端往往没有工作拷贝的需求，所以往往是一个裸库（bare repository），往往文件夹的名称为xxxx.git ，裸库的内容其实就相当于客户端某个工程下的.git目录了.   
  
使用git的时候，一般和服务器通讯使用的是ssh协议，用ssh的主要优点是速度快（传输前数据会先压缩，比HTTP快），安全，方便读写。 客户端通过ssh访问服务器端的验证方式一般有两种，一种是用户名密码的方式，一种是使用公私钥认证的方式. 使用公私钥的方式比较方便，无需每次登录输入密码。 某个受信任的客户端的公钥会被设置在服务器端的 ~/.ssh/authorized\_keys文件中，有关此文件的格式可以参见 sshd的用户手册 man sshd . authorized\_keys有个比较厉害的功能是 支持 command参数，使得每次用户使用此公钥进行验证的时候执行此后面的命令.这样就可以做一些逻辑处理了.   
  
一般git库的管理需要权限控制，如何方便简单的进行库的权限管理呢？ authorized\_keys是一个思路，指定特定command参数，每次验证好用户后首先执行相关逻辑，检测当前用户是否具有某个权限。 所以便有了gitosis，与其说gitosis是一个git权限管理系统，还不如说它是一个authorized\_keys文件管理器.   
gitosis的设计思路很巧妙，开天辟地之初，需要一个gitosis的管理员，所以你就在你的客户端机器上生成一对公私钥，将公钥拷贝到git服务器端的/tmp/下，然后通过gitosis在git用户home目录下运行如下初始化命令，后面的公钥便是gitosis首个管理员的公钥

sudo -H -u git gitosis-init < /tmp/id\_rsa.pub

执行之后，home目录下出现一个repositories的目录，目录下存在一个gitosis-admin.git的git库. 其实gitosis就是通过这个git库来管理所有git库的访问权限的。   
在刚才你的客户端机器上clone出这个库，因为你刚刚用你的公钥初始化了gitosis，所以你有权限访问服务器端的库（其实你可以在服务器端看看authorized\_keys文件便知道原因了）。 这个gitosis-admin的库中存在一个gitosis.conf和一个keydir的目录，gitosis.conf文件就是权限配置的地方，keydir目录下存放的是所有客户端的公钥，公钥名字必须和配置文件中的member名字对应.至于gitosis.conf的语法及用法可以去查看帮助。   
当修改完配置和添加好需要添加的公钥之后，提交并push到git服务器，这个库的存在hook，当提交后，会根据新的配置和公钥去更新服务器端git用户的authorized\_keys，以便加入新用户的公钥，用于权限控制命令，以及在裸库的目录下提取出新的gitosis.conf文件。  
  
所以 gitosis的核心在于authorized\_keys的command参数和git的hook功能.

## CentOS下Git服务器搭建与客户端安装

时间:2014-05-14 23:22来源:blog.51cto.com 作者:陈明乾 的BLOG [举报](http://www.centoscn.com/plus/erraddsave.php?aid=2972&title=CentOS%CF%C2Git%B7%FE%CE%F1%C6%F7%B4%EE%BD%A8%D3%EB%BF%CD%BB%A7%B6%CB%B0%B2%D7%B0) 点击:8355次

大纲：

一、前言

二、搭建Git服务器

* yum安装Git服务器
* 创建一个git用户，用来运行git服务
* 创建客户端登录证书
* 初始化Git仓库
* 禁用shell登录
* 克隆远程仓库

三、安装客户端

* Windows 客户端
* Linux 客户端

四、总结

注，测试机 CentOS 5.5 x86\_64，Git 服务器版本：git version 1.8.2.1，客户端版本：git version 1.9.2.msysgit.0。所有软件请到这里下载：http://msysgit.github.io/。

一、前言

在上一篇博客中我们主要讲解Git是什么，Git的基本原理，讲解一些有关Git的基础知识，让我们简单的了解一下Git，当然我们还有很多问题没有搞清楚，嘿嘿！不管理怎么样，在我们弄清楚这些问题之前，我们得有台Git服务器是吧，嘿嘿！好了，下面我们就来一起搭建一台Git服务器。

二、搭建Git服务器

1.yum安装Git服务器

|  |  |
| --- | --- |
|  | [root@git ~]# cd src/  [root@git src]# wget http://dl.fedoraproject.org/pub/epel/5/x86\_64/epel-release-5-4.noarch.rpm  [root@git src]# rpm -ivh epel-release-5-4.noarch.rpm  Preparing...                ########################################### [100%]      package epel-release-5-4.noarch is already installed  [root@git ~]# yum list  [root@git ~]# yum install -y git |

2.创建一个git用户，用来运行git服务

|  |  |
| --- | --- |
|  | [root@git ~]# adduser git |

3.创建客户端登录证书

注，收集所有需要登录的用户的公钥，就是他们自己生成的id\_rsa.pub文件，把所有公钥复制到/home/git/.ssh/authorized\_keys文件里，一行一个。嘿嘿！

1).客户端生成id\_rsa.pub文件的命令

|  |  |
| --- | --- |
|  | $ ssh-keygen -t rsa  $ cat  .ssh/id\_rsa.pub  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NwUHeNNi+PC6KlrcJrXXDmKxRMmgHIPp79sgX6zqfdSlmNj7rBPQeyEKS9Wg8yI6jd8aG2jsUx99Vjti2VK2vEXKkRHxwID7ri69gE71RfDtv6ekafnzLo14J8hAp0spMk+N3wEAQRYDmcYo1wmnm/jMBedGrHj4NJQ1vYy1hVtJasGMSzjcMrlz9qvaluWnQ5tQjKFQVVwKsRRRzs8qTvzVhLJt4NQ+CAN45tqfsRuf58Uba9QNK7/6xSUiIKXQiILz8PMGJ3MnlV+eN3wx2aeztdevxu9plggtG05SMmd8GNVzXrN1IaxXSvz0UwjQ2kygu7aCqO8AZWH49rouw== leo@LEO-PC |

注，一路回车即可，将生成的id\_rsa.pub，复制给管理员，帮你在服务器上增加一下，下次你用git时就不需要输入用户名和密码了。

2).查看服务器上authorized\_keys文件

|  |  |
| --- | --- |
|  | [root@git ~]# cat /home/git/.ssh/authorized\_keys  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAwMU4FKB6NRc3XSoIunWsdvVVuSoncbVb5Al6lB3ciswBVd++YmJFhqwkITNGccrO5sycROs9+Fbjgd6oBSzNuaBtCIbwNNsEyM/henTl2euI3XsnJQ/ITr6c/q0P3WoGl4E2QFQ2kZqs+1eDC0CgHcBrqvFv1Jr414sVYK9lfZwIF+jDdtaBOrSJuq1Agx9pGUFUEB4tQfkXxsWm/MvOmKAVvduKDE1eenUEL9zzyeELPcSXLe3NOoTjZhkX6EEXxQR1ZiZRFywLpfM4qopZ10to2KIUyVtzw6hx6V3cg7kn40lYVW0EAMATw9dVldwcRUI+kJzJSKUTKDVSwY3/+Q== root@CHENMINGQIAN  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAsmmJuR+KhRSpdSirCiL30RA8WbfgicY1z7itWVnKHJW6hTuJFhzruY59FilVjJR1hbQBluP9JnK3XPSK9PSg+bwiJ2iQRa39rXck35r+trVOLyNbPyfKVRfOemD8YuykMlyr5JeW8gZjsHEuLnJ8//RiCiYzd3RT/SSUQ4yawDoIIWkz3eUSL09xoCRZFBsAp+S/LD3vx2MN+FNOHwvqcE+yK3oRNIqjWwLoKE0e5TRnqNgrPQ95PJYB3XPUulzaeMwsWPZs7jIUMl/5yEqSgAkioa8SVMOsikYJG/erv99NNVgFmpCBIiWqKEGkNrIpYzLLHDSYQ4g5Gbci/RZ54Q== Administrator@WIN2003X323  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NwUHeNNi+PC6KlrcJrXXDmKxRMmgHIPp79sgX6zqfdSlmNj7rBPQeyEKS9Wg8yI6jd8aG2jsUx99Vjti2VK2vEXKkRHxwID7ri69gE71RfDtv6ekafnzLo14J8hAp0spMk+N3wEAQRYDmcYo1wmnm/jMBedGrHj4NJQ1vYy1hVtJasGMSzjcMrlz9qvaluWnQ5tQjKFQVVwKsRRRzs8qTvzVhLJt4NQ+CAN45tqfsRuf58Uba9QNK7/6xSUiIKXQiILz8PMGJ3MnlV+eN3wx2aeztdevxu9plggtG05SMmd8GNVzXrN1IaxXSvz0UwjQ2kygu7aCqO8AZWH49rouw== leo@LEO-PC |

说明：我这里有三个用户登录服务器，所以我这里就有三个ssh-rsa，大家可以看一下。

4.初始化Git仓库

注，先选定一个目录作为Git仓库，这里是/data/git/project.git。

|  |  |
| --- | --- |
|  | [root@git ~]# cd /data/git/  [root@git git]# git init --bare project.git  [root@git project.git]# ls  branches  config  description  HEAD  hooks  index  info  objects  refs |

执行以上命令 Git命令，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：

|  |  |
| --- | --- |
|  | [root@git git]# chown -R git.git project.git  [root@git git]# ls -l  总计 4  drwxr-xr-x 7 git git 4096 05-09 13:50 project.git |

5.禁用shell登录

注，出于安全考虑，第二步创建的git用户不允许登录shell，这可以通过编辑/etc/passwd文件完成。找到类似下面的一行：

|  |  |
| --- | --- |
|  | [root@git ~]# cat /etc/passwd | grep git  git:x:1001:1001:git version control:/home/git:/bin/bash |

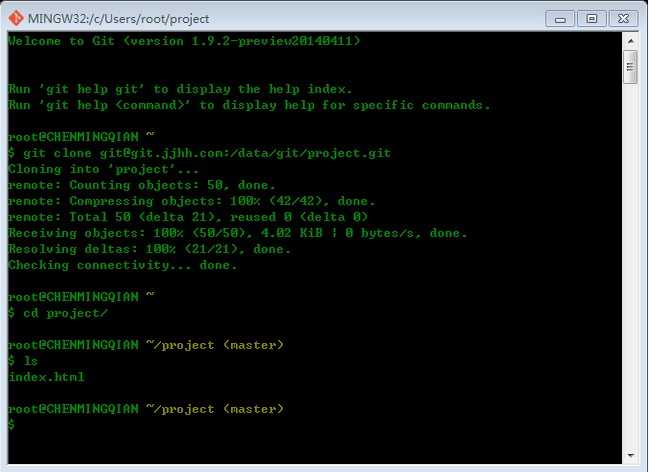
改为：

|  |  |
| --- | --- |
|  | [root@git ~]# vim /etc/passwd  git:x:1001:1001:git version control:/home/git:/usr/bin/git-shell |

这样，git用户可以正常通过ssh使用git，但无法登录shell，因为我们为git用户指定的git-shell每次一登录就自动退出。

6.克隆远程仓库

注，现在可以通过git clone命令克隆远程仓库了，在各自的电脑上运行：



注，$ git clone git@git.jjhh.com:/data/git/project.git，其中git用户名，git.jjhh.com服务器，/data/git/prgject.git是仓库路径。好了，到这里服务器的搭建到这里就完成了，下面我们来安装一下客户端。

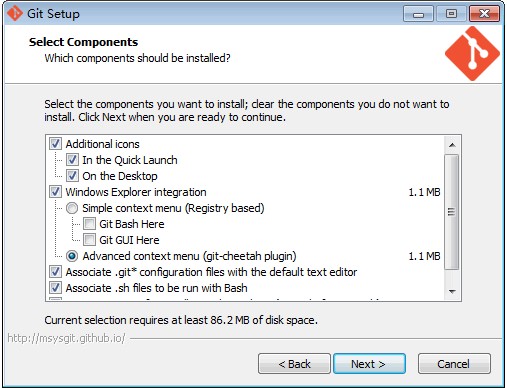
三、安装客户端

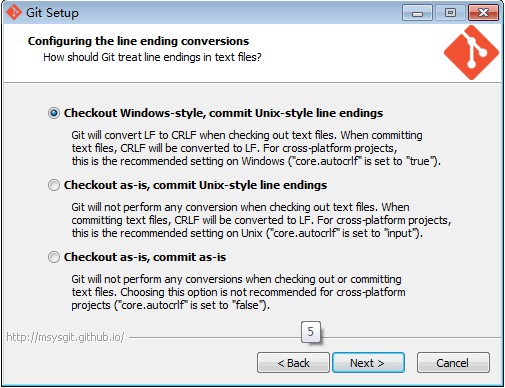
1.Windows 客户端

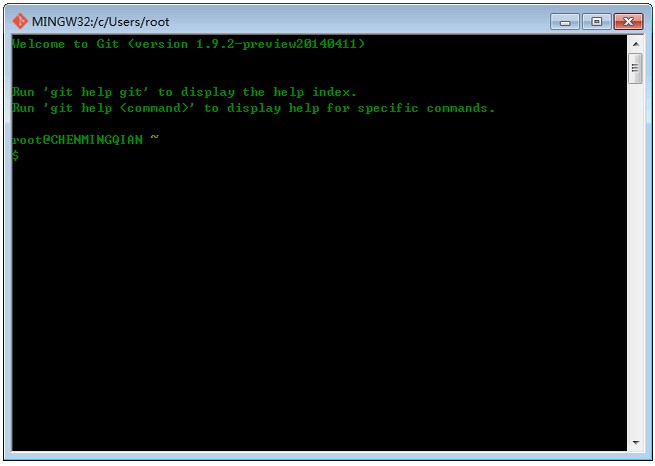
1).下载客户端

注，大家到这里下载http://msysgit.github.io/。下面简单演示一下安装过程，比较简单：

  
好了，到这里就安装完成了，安装好以后会在桌面上有个图标，你双击打开即可。如下图：

2.Linux 客户端

注，Linux客户端安装就比较简单了，直接用yum安装一下就好！

|  |  |
| --- | --- |
| 1 | [root@test ~]# yum install -y git |

到这里git就安装完成了，下面我们查看一下版本：

|  |  |
| --- | --- |
| 1  2 | [root@test ~]# git --version  git version 1.8.2. |

下面我们生成公钥并复制到服务器上：

|  |  |
| --- | --- |
|  | [root@test ~]# ssh-keygen -t rsa  Generating public/private rsa key pair.  Enter file in which to save the key (/root/.ssh/id\_rsa):  Created directory '/root/.ssh'.  Enter passphrase (empty for no passphrase):  Enter same passphrase again:  Your identification has been saved in /root/.ssh/id\_rsa.  Your public key has been saved in /root/.ssh/id\_rsa.pub.  The key fingerprint is:  48:3c:22:76:02:f1:a2:e5:27:22:cb:4f:a7:a0:98:9d root@test.com  [root@test ~]# cat .ssh/id\_rsa.pub  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAwrmgNAca77BKzXl6BP11fdaclAoWfW+x4kPDZMPj1zFoML9JfOAS7DGY9quLMitwwfg5+U7zP/egPmAhSKCECubmodzMucnRSkJGgKCt1SlctRNDRN3OIvrecAtdZfDcBaWKLjaVzGPmhweLhr6HzfqeZU09Ccis4yK3RMwip2f+K1ZZVIOKUmL4AVFl9dqtxnQ4HlW61PUxvM/ug2M68Z0jQk5DVG8w5+dvGqX1qr5YABD1NX5Jz6aUbVVx7yyoORENxu6SK83GI/V7eOkrvBMDh9nDdwvDhPhuhBDSfE+Xi92hPJPKdNroF1vx8HMvpHME20MmjSwavPrz+B18MQ== root@test.com |

下面我们复制id\_rsa.pub里的公钥到服务器的authorized\_keys文件中。

|  |  |
| --- | --- |
|  | [root@git ~]# su git  bash-3.2$ cd  bash-3.2$ vim .ssh/authorized\_keys  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAwMU4FKB6NRc3XSoIunWsdvVVuSoncbVb5Al6lB3ciswBVd++YmJFhqwkITNGccrO5sycROs9+Fbjgd6oBSzNuaBtCIbwNNsEyM/henTl2euI3XsnJQ/ITr6c/q0P3WoGl4E2QFQ2kZqs+1eDC0CgHcBrqvFv1Jr414sVYK9lfZwIF+jDdtaBOrSJuq1Agx9pGUFUEB4tQfkXxsWm/MvOmKAVvduKDE1eenUEL9zzyeELPcSXLe3NOoTjZhkX6EEXxQR1ZiZRFywLpfM4qopZ10to2KIUyVtzw6hx6V3cg7kn40lYVW0EAMATw9dVldwcRUI+kJzJSKUTKDVSwY3/+Q== root@CHENMINGQIAN  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAsmmJuR+KhRSpdSirCiL30RA8WbfgicY1z7itWVnKHJW6hTuJFhzruY59FilVjJR1hbQBluP9JnK3XPSK9PSg+bwiJ2iQRa39rXck35r+trVOLyNbPyfKVRfOemD8YuykMlyr5JeW8gZjsHEuLnJ8//RiCiYzd3RT/SSUQ4yawDoIIWkz3eUSL09xoCRZFBsAp+S/LD3vx2MN+FNOHwvqcE+yK3oRNIqjWwLoKE0e5TRnqNgrPQ95PJYB3XPUulzaeMwsWPZs7jIUMl/5yEqSgAkioa8SVMOsikYJG/erv99NNVgFmpCBIiWqKEGkNrIpYzLLHDSYQ4g5Gbci/RZ54Q== Administrator@WIN2003X323  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NwUHeNNi+PC6KlrcJrXXDmKxRMmgHIPp79sgX6zqfdSlmNj7rBPQeyEKS9Wg8yI6jd8aG2jsUx99Vjti2VK2vEXKkRHxwID7ri69gE71RfDtv6ekafnzLo14J8hAp0spMk+N3wEAQRYDmcYo1wmnm/jMBedGrHj4NJQ1vYy1hVtJasGMSzjcMrlz9qvaluWnQ5tQjKFQVVwKsRRRzs8qTvzVhLJt4NQ+CAN45tqfsRuf58Uba9QNK7/6xSUiIKXQiILz8PMGJ3MnlV+eN3wx2aeztdevxu9plggtG05SMmd8GNVzXrN1IaxXSvz0UwjQ2kygu7aCqO8AZWH49rouw== leo@LEO-PC  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAwrmgNAca77BKzXl6BP11fdaclAoWfW+x4kPDZMPj1zFoML9JfOAS7DGY9quLMitwwfg5+U7zP/egPmAhSKCECubmodzMucnRSkJGgKCt1SlctRNDRN3OIvrecAtdZfDcBaWKLjaVzGPmhweLhr6HzfqeZU09Ccis4yK3RMwip2f+K1ZZVIOKUmL4AVFl9dqtxnQ4HlW61PUxvM/ug2M68Z0jQk5DVG8w5+dvGqX1qr5YABD1NX5Jz6aUbVVx7yyoORENxu6SK83GI/V7eOkrvBMDh9nDdwvDhPhuhBDSfE+Xi92hPJPKdNroF1vx8HMvpHME20MmjSwavPrz+B18MQ== root@test.com |

下面我们clone一个仓库到本地的目录中。

|  |  |
| --- | --- |
|  | [root@test ~]# cd /data/dev  [root@test dev]# git clone git@git.jjhh.com:/data/git/project.git  Cloning into 'project'...  The authenticity of host 'git.jjhh.com (114.112.173.150)' can't be established.  RSA key fingerprint is ca:ec:a2:7e:e6:89:ca:19:d3:93:7f:4b:c3:c0:c7:fd.  Are you sure you want to continue connecting (yes/no)? yes  Warning: Permanently added 'git.jjhh.com,114.112.173.150' (RSA) to the list of known hosts.  remote: Counting objects: 50, done.  remote: Compressing objects: 100% (42/42), done.  remote: Total 50 (delta 21), reused 0 (delta 0)  Receiving objects: 100% (50/50), 4.02 KiB, done.  Resolving deltas: 100% (21/21), done.  [root@test dev]# ls  project  [root@test dev]# cd project/  [root@test project]# ls  index.html |

好了，到这里我们的git客户端到这里就全部安装完成了。

四、总结

Q&A，有什么问题欢迎大家交流讨论。最后，希望大家有所收获^\_^……