# Git服务器搭建

# [在 Linux 下搭建 Git 服务器](http://www.cnblogs.com/dee0912/p/5815267.html)

环境：

服务器 CentOS6.6 + git（version 1.7.1）  
客户端 Windows10 + git（version 2.8.4.windows.1）

## 安装 Git

Linux 做为服务器端系统，Windows 作为客户端系统，分别安装 Git

**服务器端：**

#yum install -y git

安装完后，查看 Git 版本

[root@localhost ~]# git --version

git version 1.7.1

 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。

编译安装的步骤是

（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

$ wget https://github.com/git/git/archive/v2.3.0.zip

$ unzip v2.3.0.zip

$ cd git-2.3.0

（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。

**客户端：**

下载 [Git for Windows](https://github.com/git-for-windows/git/releases/download/v2.9.3.windows.2/Git-2.9.3.2-64-bit.exe)，地址：[https://git-for-windows.github.io/](https://git-for-windows.github.io/" \t "_blank)

安装完之后，可以使用 Git Bash 作为命令行客户端。

安装完之后，查看 Git 版本

$ git --version

git version 2.8.4.windows.1

## 服务器端创建 git 用户，用来管理 Git 服务，并为 git 用户设置密码

[root@localhost home]# id git

id: git：无此用户

[root@localhost home]# useradd git

[root@localhost home]# passwd git

## 服务器端创建 Git 仓库

设置 /home/data/git/gittest.git 为 Git 仓库

然后把 Git 仓库的 owner 修改为 git

[root@localhost home]# mkdir -p data/git/gittest.git

[root@localhost home]# git init --bare data/git/gittest.git

Initialized empty Git repository in /home/data/git/gittest.git/

[root@localhost home]# cd data/git/

[root@localhost git]# chown -R git:git gittest.git/

## 客户端 clone 远程仓库

进入 Git Bash 命令行客户端，创建项目地址（设置在 d:/wamp64/www/gittest\_gitbash）并进入:

[复制代码](javascript:void(0);)

dee@Lenovo-PC MINGW64 /d

$ cd wamp64/www

dee@Lenovo-PC MINGW64 /d/wamp64/www

$ mkdir gittest\_gitbash

dee@Lenovo-PC MINGW64 /d/wamp64/www

$ cd gittest\_gitbash

dee@Lenovo-PC MINGW64 /d/wamp64/www/gittest\_gitbash

$

[复制代码](javascript:void(0);)

然后从 Linux Git 服务器上 clone 项目：

|  |  |
| --- | --- |
| 1 | $ git clone git@192.168.56.101:/home/data/gittest.git |



当第一次连接到目标 Git 服务器时会得到一个提示：

The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.

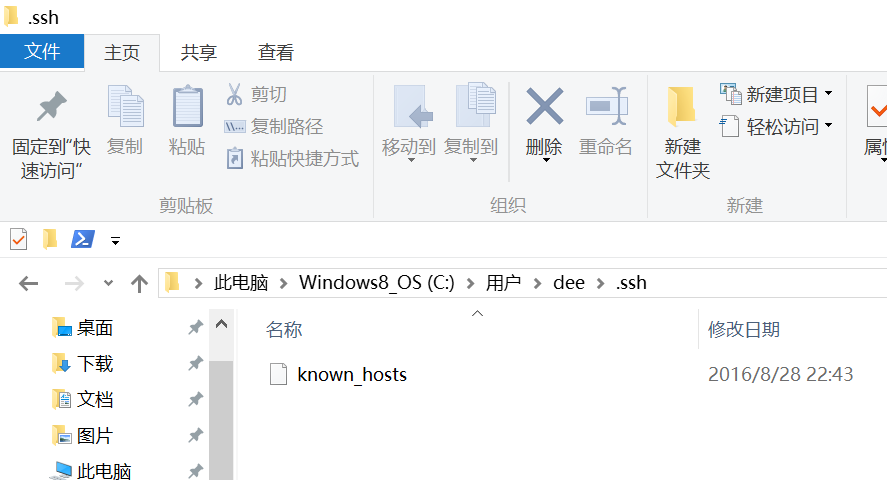
RSA key fingerprint is SHA256:Ve6WV/SCA059EqoUOzbFoZdfmMh3B259nigfmvdadqQ.

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

选择 yes：

Warning: Permanently added '192.168.56.101' (RSA) to the list of known hosts.

此时 C:\Users\用户名\.ssh 下会多出一个文件 known\_hosts，以后在这台电脑上再次连接目标 Git 服务器时不会再提示上面的语句。



后面提示要输入密码，可以采用 SSH 公钥来进行验证。

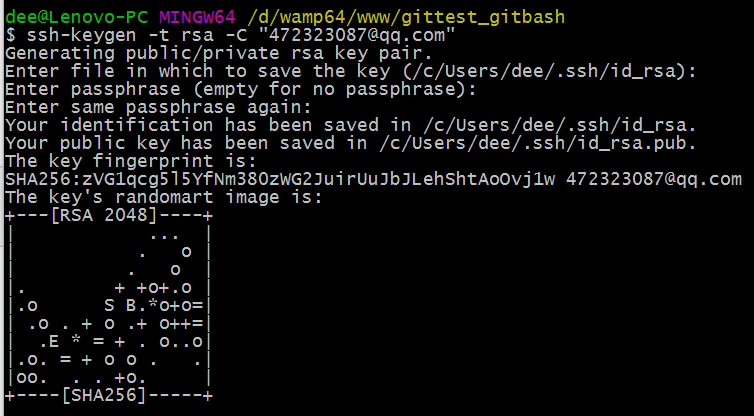
## 客户端创建 SSH 公钥和私钥

$ ssh-keygen -t rsa -C "472323087@qq.com"

此时 C:\Users\用户名\.ssh 下会多出两个文件 id\_rsa 和 id\_rsa.pub

id\_rsa 是私钥

id\_rsa.pub 是公钥



## 服务器端 Git 打开 RSA 认证

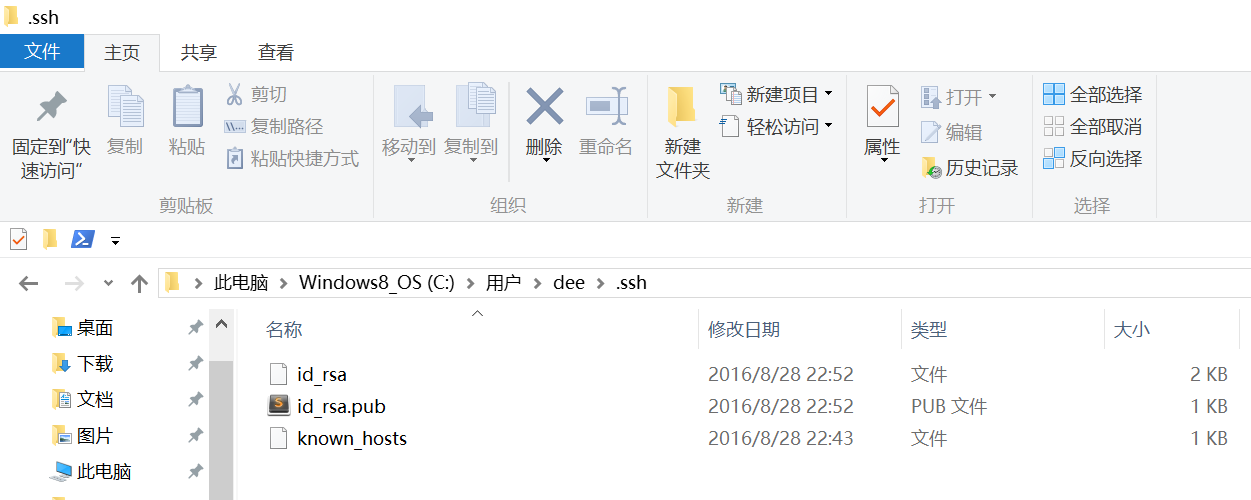
进入 /etc/ssh 目录，编辑 sshd\_config，打开以下三个配置的注释：

RSAAuthentication yes

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized\_keys

保存并重启 sshd 服务：

[root@localhost ssh]# /etc/rc.d/init.d/sshd restart

由 AuthorizedKeysFile 得知公钥的存放路径是 .ssh/authorized\_keys，实际上是 $Home/.ssh/authorized\_keys，由于管理 Git 服务的用户是 git，所以实际存放公钥的路径是 /home/git/.ssh/authorized\_keys

在 /home/git/ 下创建目录 .ssh

[root@localhost git]# pwd

/home/git  
[root@localhost git]# mkdir .ssh  
[root@localhost git]# ls -a   
. .. .bash\_logout .bash\_profile .bashrc .gnome2 .mozilla .ssh

然后把 .ssh 文件夹的 owner 修改为 git

[复制代码](javascript:void(0);)

[root@localhost git]# chown -R git:git .ssh

[root@localhost git]# ll -a

总用量 32

drwx------. 5 git git 4096 8月 28 20:04 .

drwxr-xr-x. 8 root root 4096 8月 28 19:32 ..

-rw-r--r--. 1 git git 18 10月 16 2014 .bash\_logout

-rw-r--r--. 1 git git 176 10月 16 2014 .bash\_profile

-rw-r--r--. 1 git git 124 10月 16 2014 .bashrc

drwxr-xr-x. 2 git git 4096 11月 12 2010 .gnome2

drwxr-xr-x. 4 git git 4096 5月 8 12:22 .mozilla

drwxr-xr-x. 2 git git 4096 8月 28 20:08 .ssh

[复制代码](javascript:void(0);)

## 将客户端公钥导入服务器端 /home/git/.ssh/authorized\_keys 文件

回到 Git Bash 下，导入文件：

$ ssh git@192.168.56.101 'cat >> .ssh/authorized\_keys' < ~/.ssh/id\_rsa.pub

需要输入服务器端 git 用户的密码



回到服务器端，查看 .ssh 下是否存在 authorized\_keys 文件：

[root@localhost git]# cd .ssh

[root@localhost .ssh]# ll

总用量 4

-rw-rw-r--. 1 git git 398 8月 28 20:08 authorized\_keys

可以查看一下是否是客户端生成的公钥。

**重要：**

**修改 .ssh 目录的权限为 700**

**修改 .ssh/authorized\_keys 文件的权限为 600**

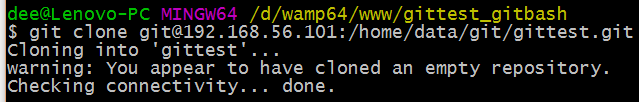
[root@localhost git]# chmod 700 .ssh

[root@localhost git]# cd .ssh

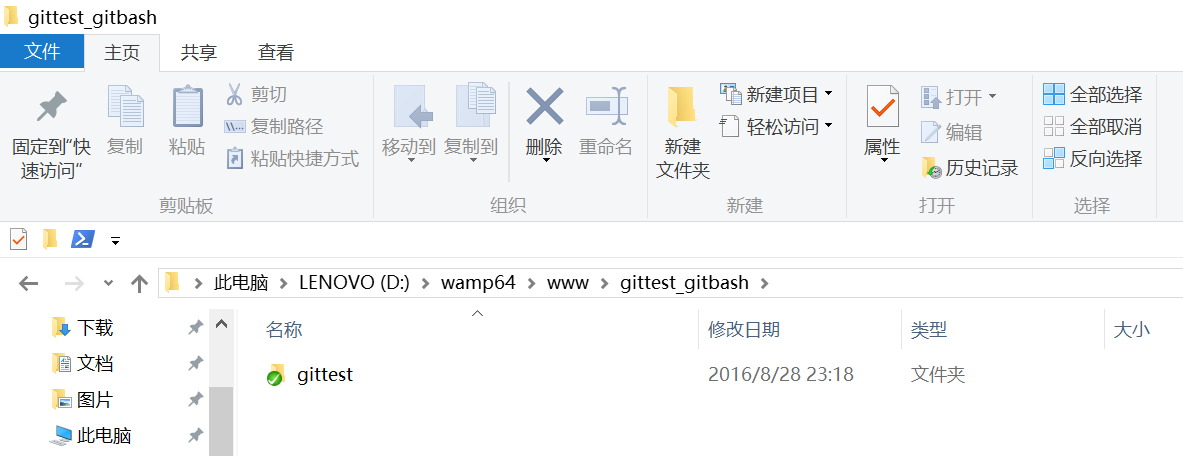
[root@localhost .ssh]# chmod 600 authorized\_keys

## 客户端再次 clone 远程仓库

$ git clone 账号@192.168.56.101:/home/data/git/gittest.git

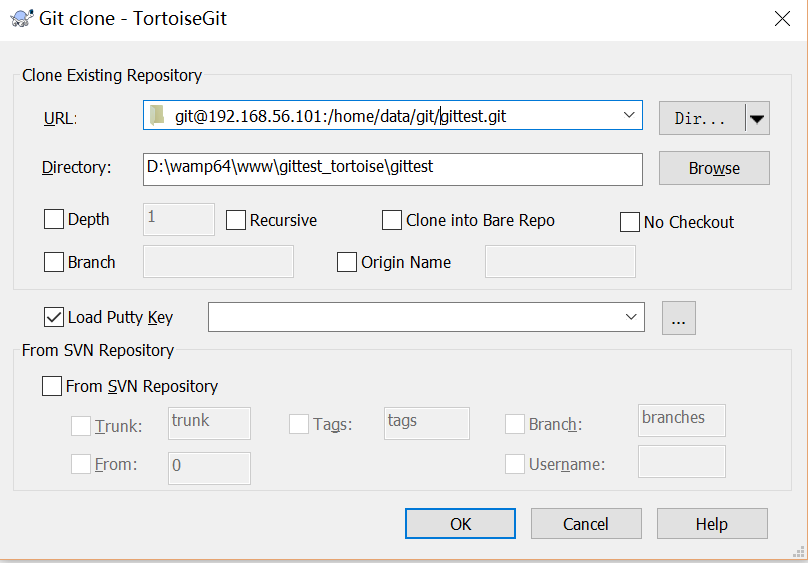


查看客户端项目目录：



项目已经 clone 了。

也可以使用 tortoiseGit 客户端来管理项目：



## 禁止 git 用户 ssh 登录服务器

之前在服务器端创建的 git 用户不允许 ssh 登录服务器

编辑 /etc/passwd

找到：

git:x:502:504::/home/git:/bin/bash

修改为

git:x:502:504::/home/git:/bin/git-shell

此时 git 用户可以正常通过 ssh 使用 git，但无法通过 ssh 登录系统。