1 git服务器搭建文档

① 安装 Git

② 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

③ 服务器端创建 Git 仓库

④ 客户端 clone 远程仓库

⑤ 客户端创建 SSH 公钥和私钥

⑥ 服务器端 Git 打开 RSA 认证

⑦ 将客户端公钥导入服务器端 /home/git/.ssh/authorized\_keys 文件

⑧ 客户端再次 clone 远程仓库

⑨ 禁止 git 用户 ssh 登录服务器

1 上传文件

git-2.8.3.tar.gz

2 解压文件

tar -zxvf git-2.8.3.tar.gz

3 进入git目录下进行编译

cd git-2.8.3

4 进行编译安装

[root@localhost git-2.8.3]# make prefix=/usr/common/git

[root@localhost git-2.8.3]# make prefix=/usr/common/git install

5 配置环境变量

[root@localhost git]# vim /etc/pofile

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

6 重新激活一下邮件

[root@localhost git]# source /etc/profile

7 此时已经可以在服务器端敲命令了 测试

[root@localhost git]# git --version

git version 2.8.3

8 服务端创建git用户 用来管理git服务 并为git用户设置密码

[root@bogon project1]# id git

id: git: No such user

[root@bogon project1]# useradd git

[root@bogon project1]# passwd git

9 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

10 创建git仓库

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

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

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

[root@bogon git]# cd data/git/

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

此时服务器端已经构建好了

11 客户端从Linux Git服务器端clone项目到本地

$ git clone git@192.168.232.131:/home/data/gittest.git

如果SSH用的不是默认的22端口，则需要使用以下的命令（假设SSH端口号是7700）：

$ git clone ssh://git@192.168.232.131:22/home/data/gittest.git

12 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.

如果不是提示这个，可以到administrator/目录下删除.ssh目录

13 此时C:\Users\Administrator\.ssh目录下会出现known\_hosts文件

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。

14 客户端创建ssh公钥

ssh-keygen -t rsa -C "894154899@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

15 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

[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

16 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：此时ssh-key已经导入服务器端了

Administrator@PC-201709252329 MINGW64 ~

$ ssh git@192.168.232.131 'cat >> .ssh/authorized\_keys' < ~/.ssh/id\_rsa.pub

git@192.168.232.131's password

17 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。

18 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

19 客户端再次 clone 远程仓库

git clone git@192.168.232.131:/home/data/git/gittest.git

git clone ssh://git@192.168.232.131:22/home/data/git/gittest.git

20 可能报错

$ git clone root@192.168.232.131:/home/data/git/gittest.git

Cloning into 'gittest'...

root@192.168.232.131's password:

Permission denied, please try again.

root@192.168.232.131's password:

bash: git-upload-pack: command not found

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.

原因git安装不在服务器的默认路径下/usr/bin/git-upload-pack 所以要建立软连接

服务器端找到git的安装路径 ln -s /usr/common/git/bin/git-upload-pack /usr/bin/git-upload-pack

同样的也会碰到 ln -s /usr/common/git/bin/git-receive-pack /usr/bin/git-receive-pack

此时就可以git clone 成功了

21 可能报错

$ git pull

There is no tracking information for the current branch.

Please specify which branch you want to merge with.

See git-pull(1) for details.

git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

git branch --set-upstream-to=<remote>/<branch> master

git本地新建一个分支后，必须要做远程分支关联。如果没有关联，

git会在下面的操作中提示你显示的添加关联。

关联目的是如果在本地分支下操作： git pull, git push ，

不需要指定在命令行指定远程的分支．推送到远程分支后，

你只要没有显示指定，git pull的时候，就会提示你,立连接。

22 可能报错

error: src refspec master does not match any.

error: failed to push some refs to 'git@github.com:hahaha/ftpmanage.git'

出现这个的原因是因为仓库是空的，所示要在仓库里添加文件

23 此时客户端已经可以用了

Administrator@PC-201709252329 MINGW64 /e/newre/gittest (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

1.txt

nothing added to commit but untracked files present (use "git add" to track)

Administrator@PC-201709252329 MINGW64 /e/newre/gittest (master)

$ git add \*

Administrator@PC-201709252329 MINGW64 /e/newre/gittest (master)

$ git commit -m "A"

[master (root-commit) ddea193] A

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 1.txt

Administrator@PC-201709252329 MINGW64 /e/newre/gittest (master)

$ git pull origin master

fatal: Couldn't find remote ref master

fatal: The remote end hung up unexpectedly

Administrator@PC-201709252329 MINGW64 /e/newre/gittest (master)

$ git push origin master

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Writing objects: 100% (3/3), 199 bytes | 49.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To 192.168.232.131:/home/data/git/gittest.git

\* [new branch] master -> master

Administrator@PC-201709252329 MINGW64 /e/newre/gittest (master)

$ git pull

Already up to date.