

# Git教程

## windows版Git的安装和使用

### Git教程

windows版Git的安装和使用

一，下载git

二，使用前的配置

```
git --version :检查是否安装成功
git config --global user.name "xiaomi" : 配置用户名字
git config --global user.email "username@email.com" : 配置用户邮箱
git config --global --list检查是否配置成功：
```

三，git 命令

```
git init [path] 初始化
git add -A 表示当前工作区全部内容添加到暂存区
git add . 表示当前目录全部添加到暂存区
git add a.txt 添加指定文件到暂存区，可以有多个文件，以空格隔开
git add demo3.txt demo2.txt 添加多个文件
git commit -m "这是当前版本的说明信息" ,必须要有-m,添加待提交版本的信息文字
git status 查询当前仓库的状态
git log 查看历史记录
git relog 查看历史操作步骤，也就是查看命令操作指定
git diff demo.txt 当版本文件与工作区的文件内容进行比较
git log --graph 绘图
git log --graph --pretty=oneline 绘图更好
head@{1}大括号中的数字，代表之前第几个版本
git diff head@{0} head@{1} 指定版本间的比较 方式一
git diff head "head^" 指定版本间的比较 方式二，双引号中的head 代表head@{0}
git remote -v :查看已连接的远程仓库
ssh -T 用户名@主机地址：本地仓库连接远程仓库
```

### Linux系统配置git服务器，模拟远程仓库

```
yum -y install git --安装Linux_Git
git --version :检查是否安装成功
使用前的配置设置
groupadd git 创建用户组 useradd git -g git 创建用户 passwd git 创建密码
:git123456
为了防止分享的文件被删除，所以创建专门的linux用户
vi /etc/passwd:修改文件使远程连接不会显示命令行
mkdir /home/git/test.git ---创建远程仓库 git init --bare : 设置为裸仓库
chown -R git:git /home/git/test.git : 修改为git权限
ssh -T git@192.168.1.8 连接远程仓库
git clone git@192.168.1.8:/home/git/test.git 在本地的文件夹中克隆（本地没有git的目录下）
git remote add origin git@192.168.1.8:/home/git/test.git 本地仓库关联远程仓库
git push origin master 将本地仓库文件分享到远程仓库
git pull 远程仓库 远程分支：本地分支 用于拉取数据
ssh-keygen -t rsa -C "aa@qq.com" 制作公钥私钥，开通免密，一直回车，email是指本地仓库的email
在linux服务器（远程仓库）上配置公钥
在本地仓库使用clone或push都不用输密码了
```

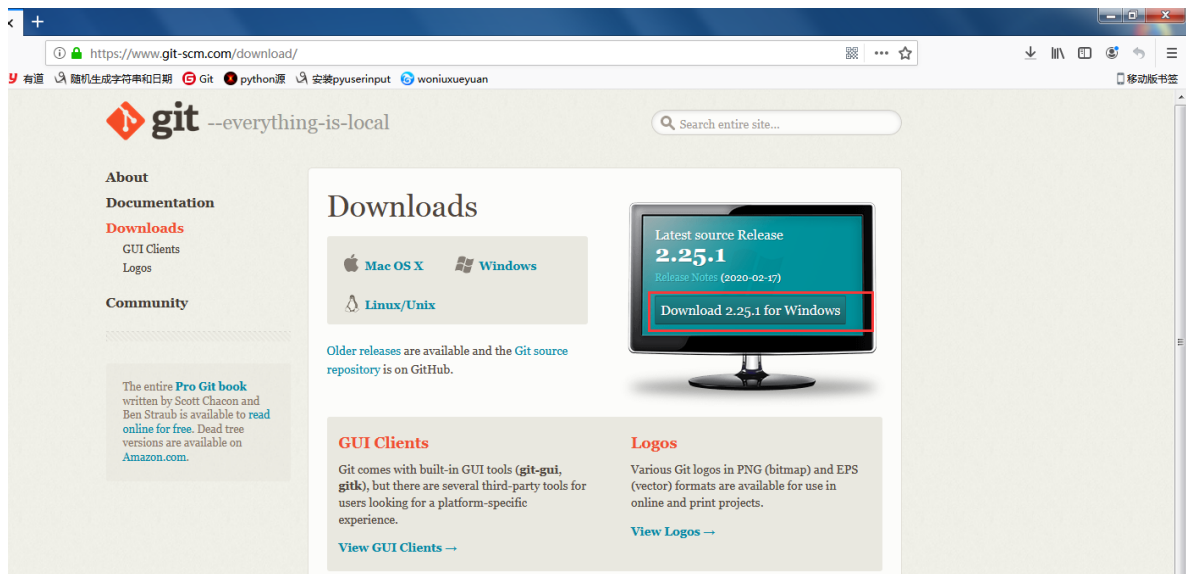
四，Git分支

```
git branch 查看本地分支
git branch -r 查看了远程分支
```

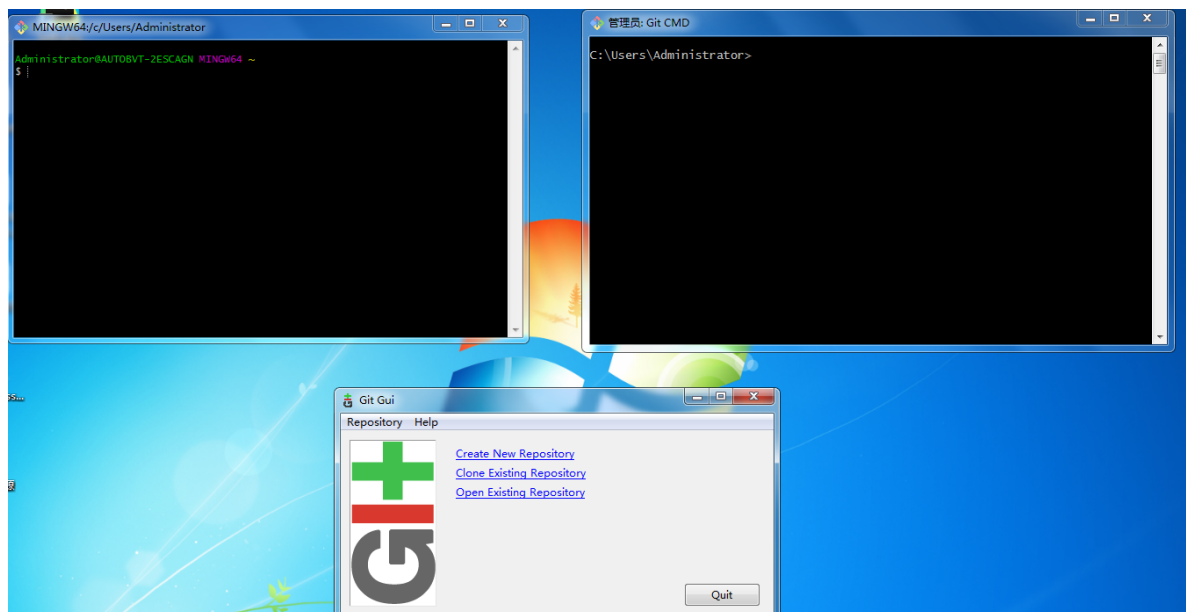
git branch -a 查看所有分支  
git branch xiaomi 创建分支，但head还是在当前分支  
git branch -d xiaomi 删除分支  
git branch -D 分支名 强制删除  
git branch -m old dev 修改分支名  
git checkout dev 切换分支  
git checkout -b xiaomi 创建分支并将head 切换到分支上

## 一，下载git

官网下载：<https://www.git-scm.com/download/>，并一直Next安装



打开软件：三种模式



## 二，使用前的配置

git --version :检查是否安装成功

git config --global user.name "xiaomi" : 配置用户名字

git config --global user.email "[username@email.com](mailto:username@email.com)" : 配置用户邮箱

git config --global --list检查是否配置成功 :

```
管理员: C:\Windows\System32\cmd.exe

D:\test>git --version
git version 2.25.1.windows.1

D:\test>git config --global user.name "dudu"

D:\test>git config --global user.email "dudu@qq.com"

D:\test>git config --global --list
user.name=dudu
user.email=dudu@qq.com
user.anme=dudu

D:\test>_
```

### 三 , git 命令

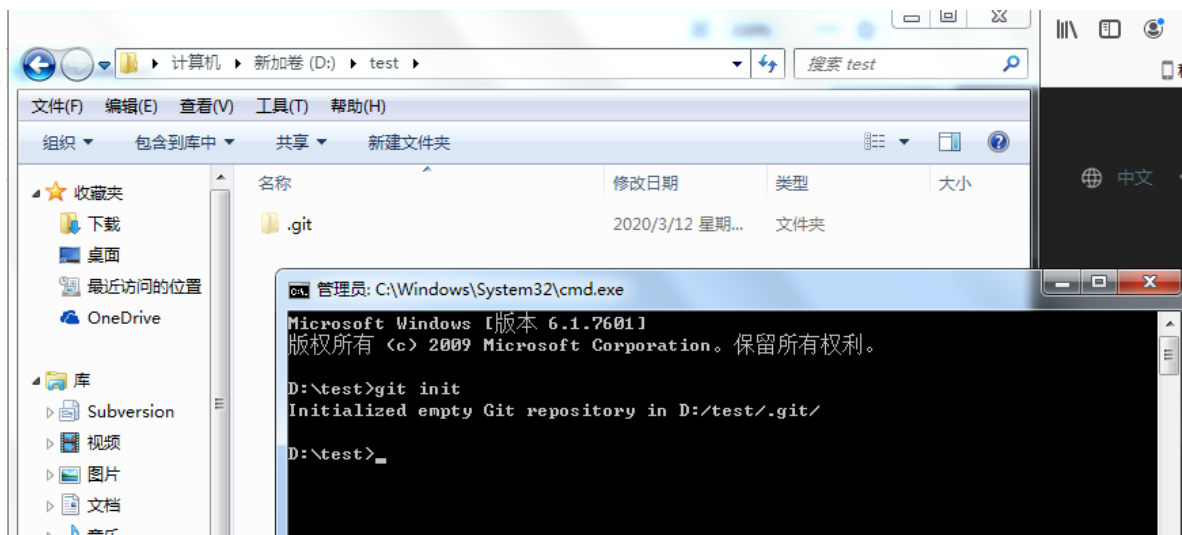
git init [path] 初始化

注:命令中不需要中括号

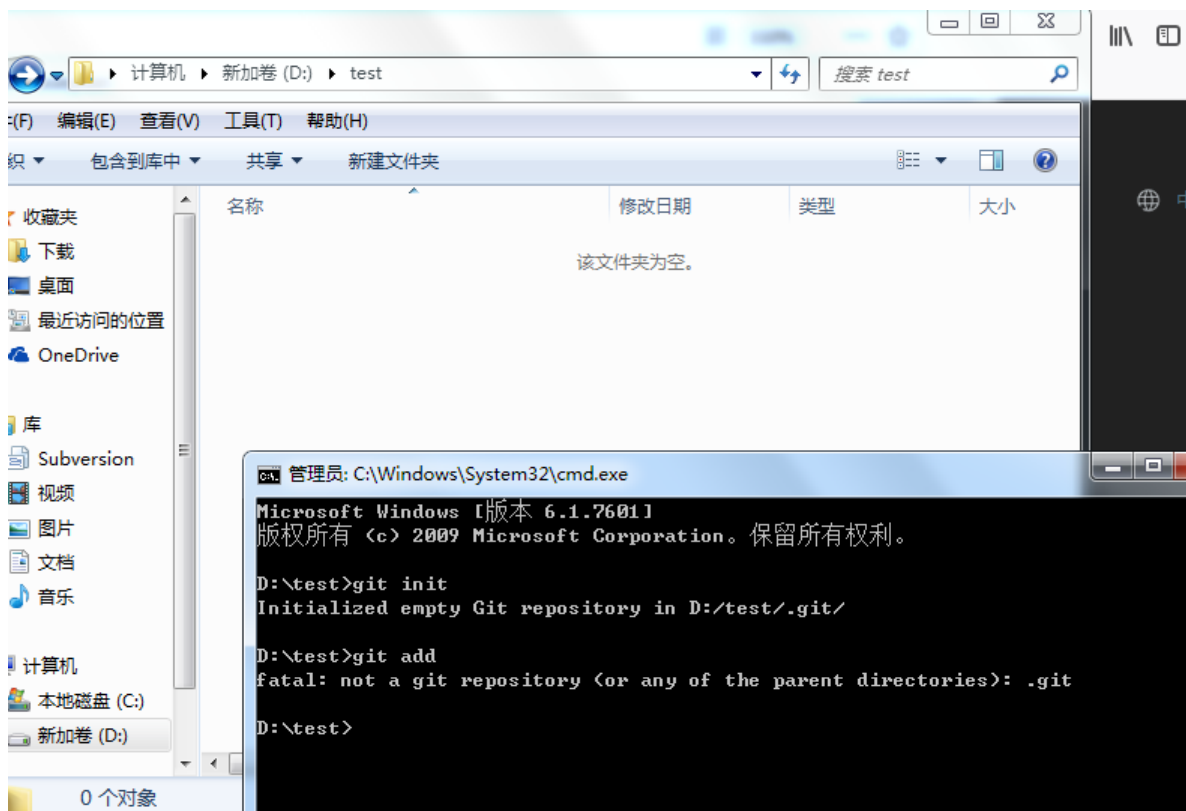
如果指定了path,就是对指定的path 进行初始化

如果没有指定,那就是对当前目录进行初始化

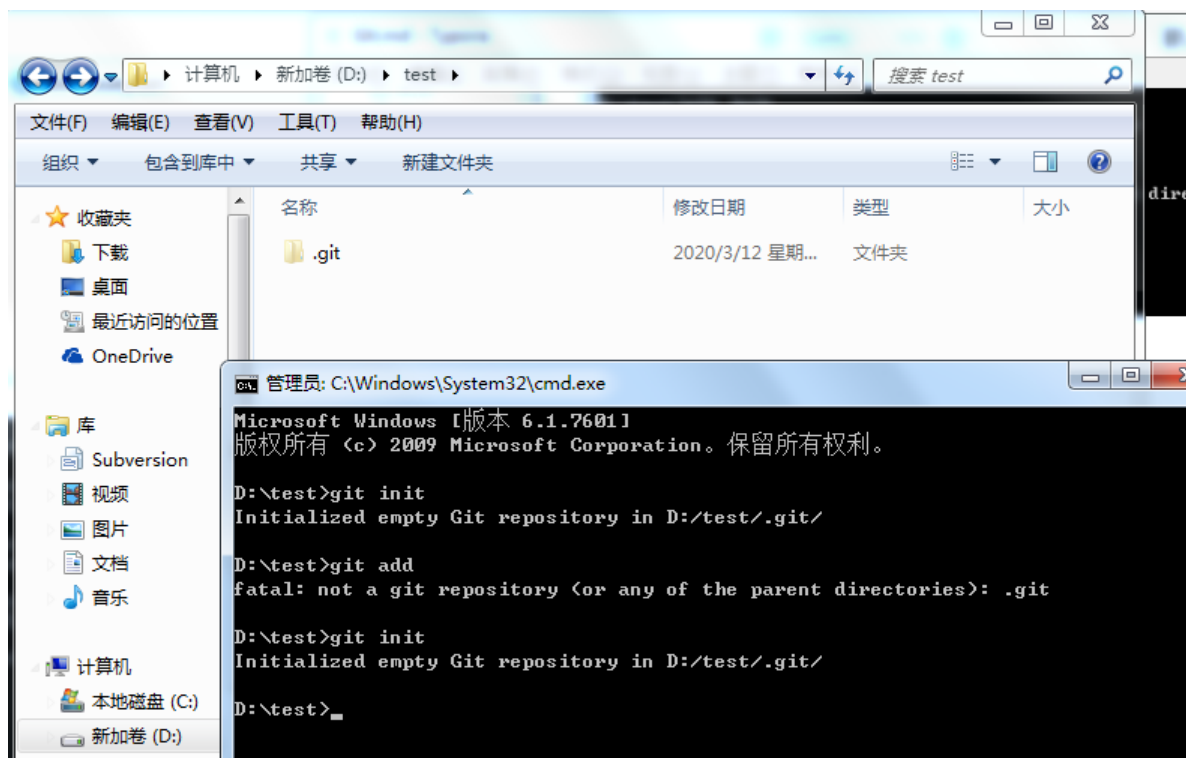
如果不是在当前路径下,需要在init后面加上路径,添加后可以看到隐藏的文件.git



如果.git的文件夹删除后,就可以回到windows的正常文件夹



再次设为git的工作区



git add -A 表示当前工作区全部内容添加到暂存区

```
C:\Windows\System32\cmd.exe

D:\test>git add .

D:\test>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   123/abd.txt
    new file:   a.txt
    new file:   b.txt

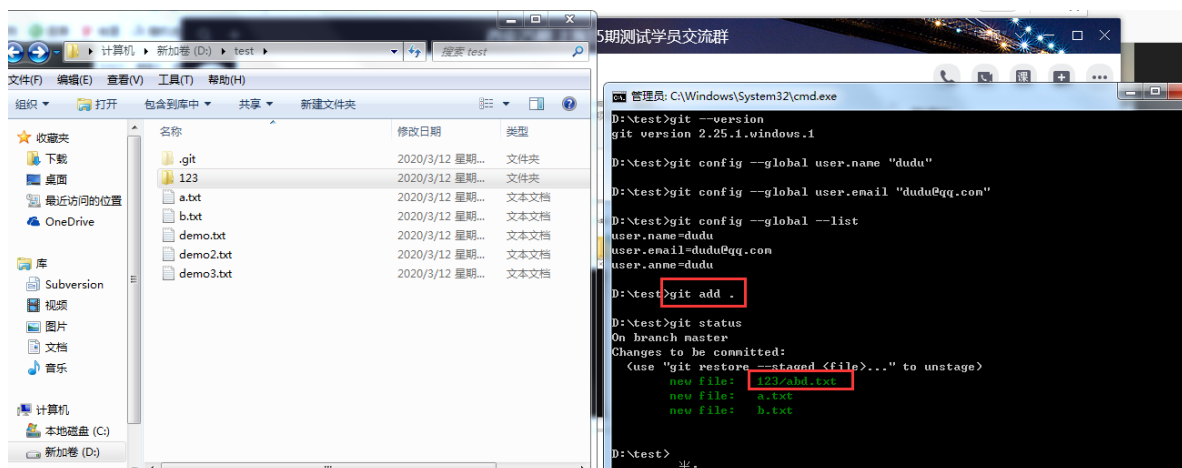
D:\test>git add -A

D:\test>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   123/abd.txt
    new file:   456/678/abcdefg/aaa.txt
    new file:   a.txt
    new file:   b.txt

D:\test>
```

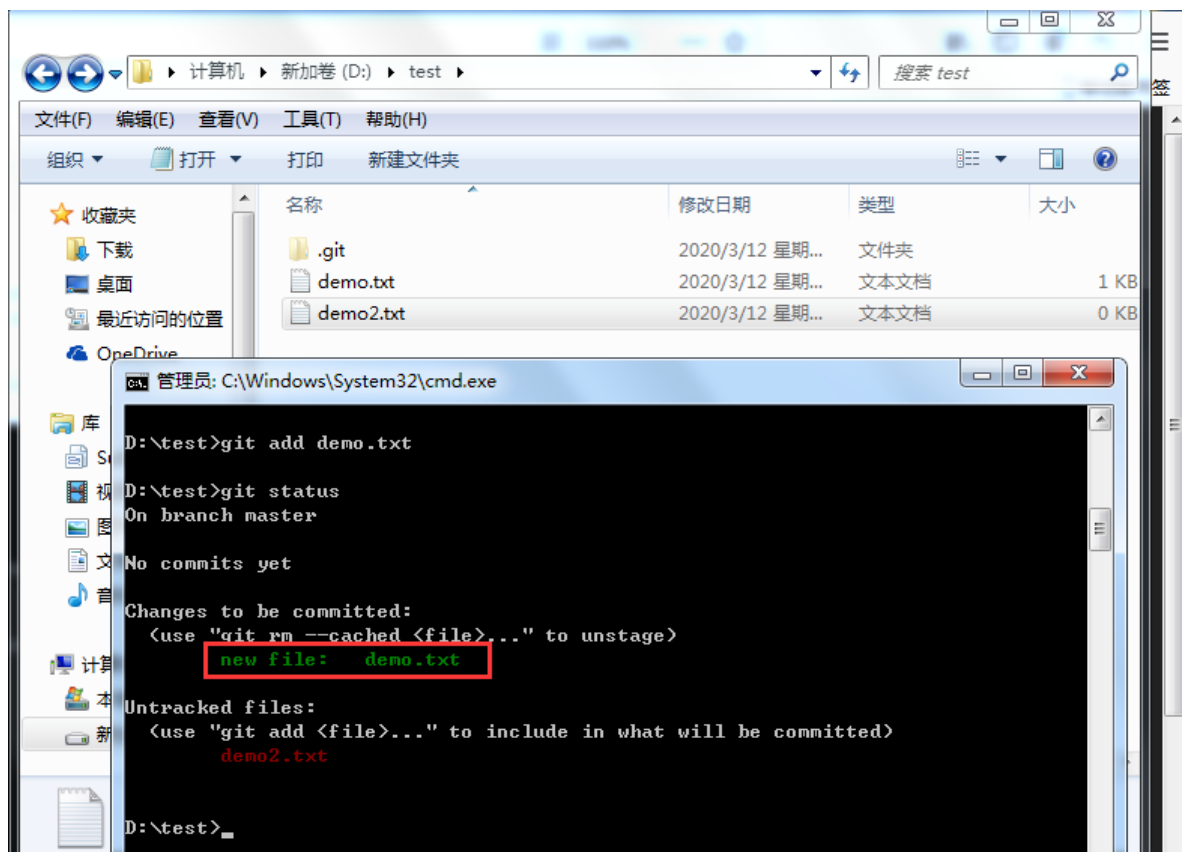
git add . 表示当前目录全部添加到暂存区

包含新增加的，修改过的，删除的不算

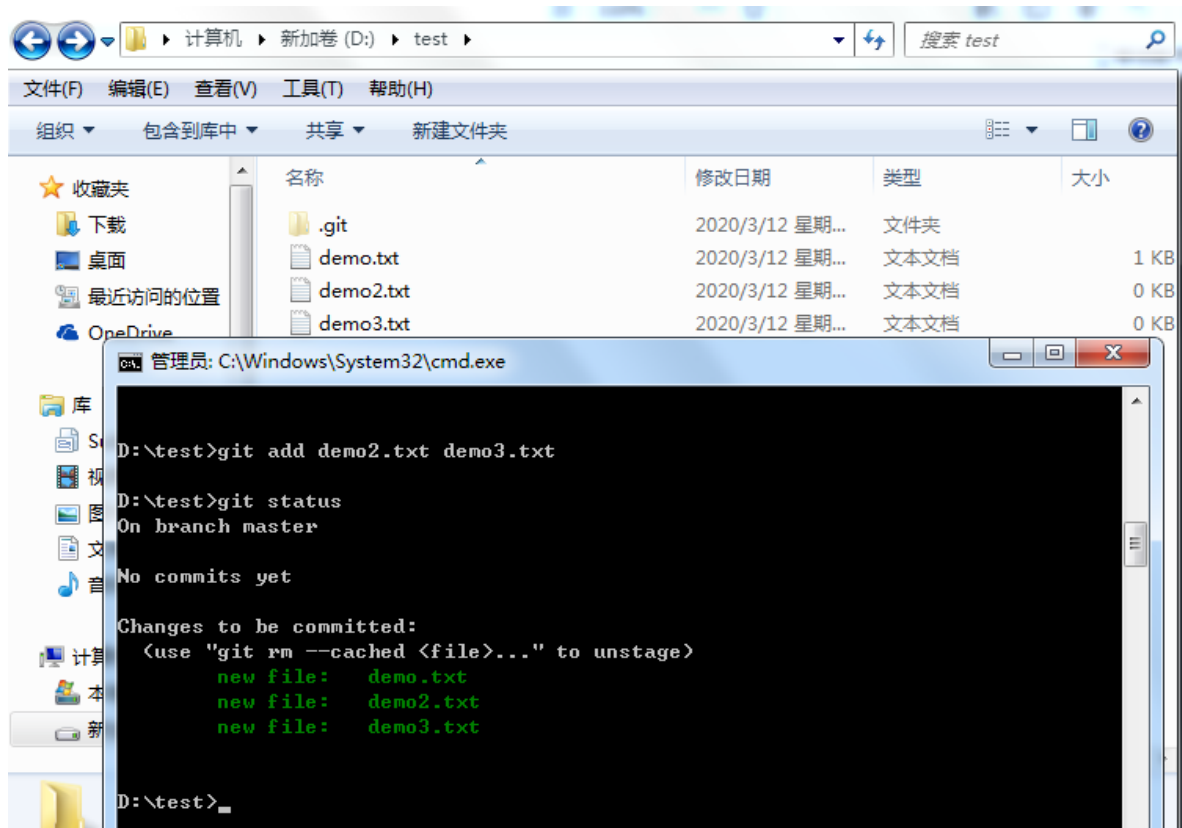


git add a.txt 添加指定文件到暂存区，可以有多个文件，以空格隔开

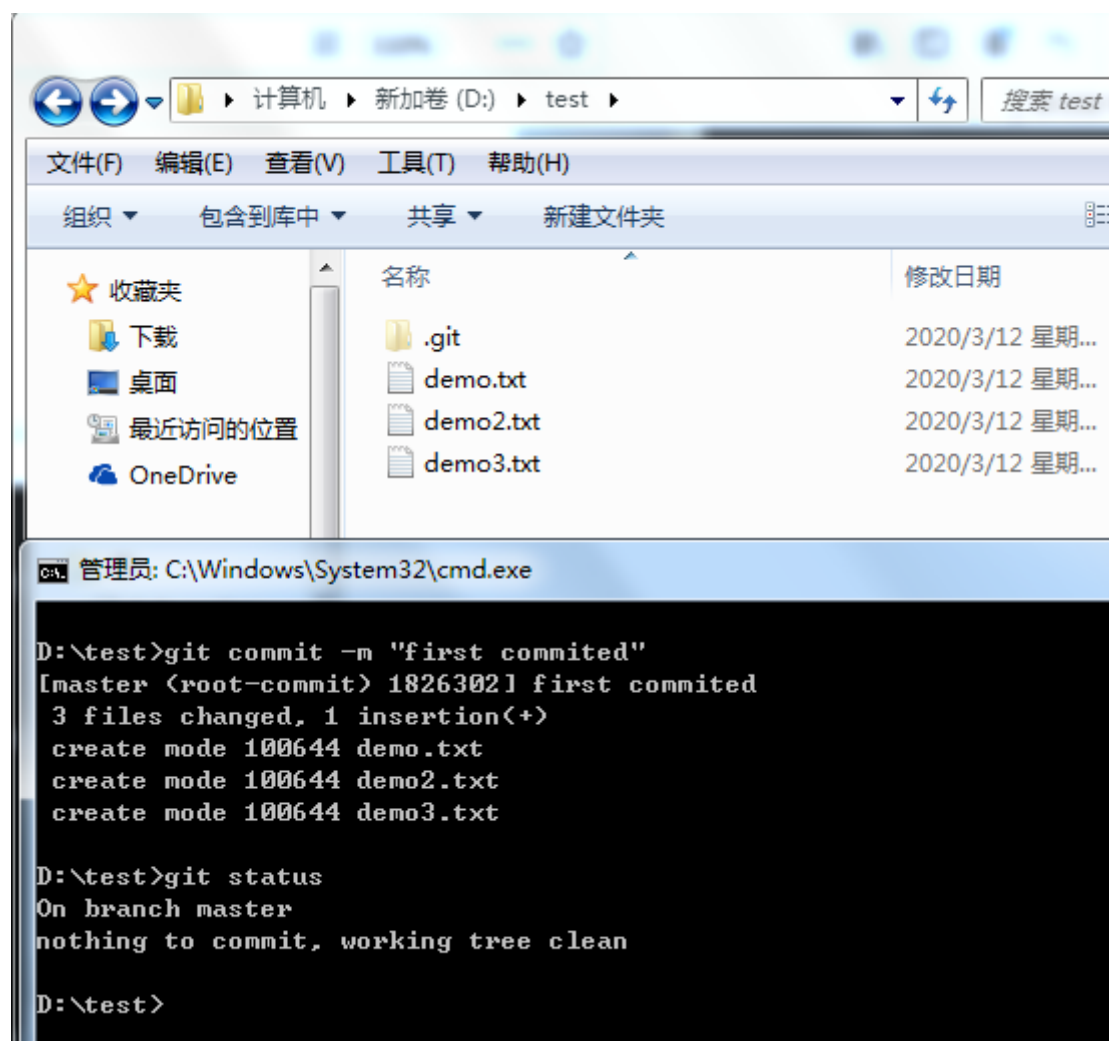
添加一个文件：git add demo.txt



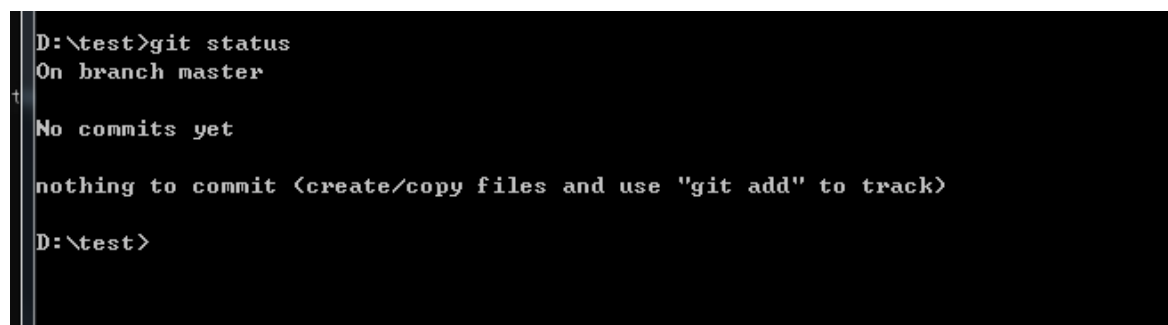
git add demo3.txt demo2.txt 添加多个文件



git commit -m "这是当前版本的说明信息",必须要有-m,添加待提交版本的信息文字



git status 查询当前仓库的状态



git log 查看历史记录

```
管理员: C:\Windows\System32\cmd.exe

D:\test>git log
commit 18263022b2c4fca661d7fa2492d96acb5e43edb5 <HEAD -> master
Author: xiaomi <2424076343@qq.com>
Date: Thu Mar 12 11:14:04 2020 +0800

    first committed

D:\test>
```

半:

git reflog 查看历史操作步骤，也就是查看命令操作指定

```
管理员: C:\Windows\System32\cmd.exe

D:\test>git log
commit 18263022b2c4fca661d7fa2492d96acb5e43edb5 <HEAD -> master
Author: xiaomi <2424076343@qq.com>
Date: Thu Mar 12 11:14:04 2020 +0800

    first committed

D:\test>git reflog
1826302 <HEAD -> master HEAD@0: commit (initial): first committed

D:\test>
```

git diff demo.txt 当版本文件与工作区的文件内容进行比较



```

D:\test>git diff demo.txt
diff --git a/demo.txt b/demo.txt
index b6cbc2d..fc76658 100644
--- a/demo.txt
+++ b/demo.txt
@@ -1,1 @@
-welcome
\ No newline at end of file
+welcome to china.
\ No newline at end of file

D:\test>

```

git log --graph 绘图

```

D:\test>git log --graph
* commit b56e3a8be1dfecd2ba01b005a19aae4b5d22ddc9 <HEAD -> master>
! Author: dudu <dudu@qq.com>
! Date: Thu Mar 12 11:48:22 2020 +0800
!
!     second committed.
!
* commit 18263022b2c4fca661d7fa2492d96acb5e43edb5
! Author: xiaomi <2424076343@qq.com>
! Date: Thu Mar 12 11:14:04 2020 +0800
!
!     first committed

D:\test>

```

git log --graph --pretty=oneline 绘图更好

```

D:\test>git log --graph --pretty=oneline
* b56e3a8be1dfecd2ba01b005a19aae4b5d22ddc9 <HEAD -> master> second committed.
* 18263022b2c4fca661d7fa2492d96acb5e43edb5 first committed

D:\test>

```

head@{1}大括号中的数字，代表之前第几个版本

git diff head@{0} head@{1} 指定版本间的比较 方式一

```
管理员: C:\Windows\System32\cmd.exe
* 18263022b2c4fca661d7fa2492d96acb5e43edb5 first committed

D:\test>git reflog
b56e3a8 <HEAD -> master> HEAD@{0}: commit: second committed.
1826302 HEAD@{1}: commit (initial): first committed

D:\test>git diff head@{0} head@{1} 对比两个仓库
diff --git a/123/abd.txt b/123/abd.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/456/678/abcdefgh/aaa.txt b/456/678/abcdefgh/aaa.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/a.txt b/a.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/b.txt b/b.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/demo.txt b/demo.txt
index fc76658..b6cbc2d 100644
--- a/demo.txt
+++ b/demo.txt
@@ -1,1 @@
-welcome to china.
\ No newline at end of file
+welcome
半:
```

git diff head "head^" 指定版本间的比较 方式二，双引号中的head 代表head@{0}

```
D:\test>git diff head "head^"
diff --git a/123/abd.txt b/123/abd.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/456/678/abcdefgh/aaa.txt b/456/678/abcdefgh/aaa.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/a.txt b/a.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/b.txt b/b.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/demo.txt b/demo.txt
index fc76658..b6cbc2d 100644
--- a/demo.txt
+++ b/demo.txt
@@ -1,1 @@
-welcome to china.
\ No newline at end of file
+welcome
\ No newline at end of file

D:\test>
半:
```

git reset 默认操作会修改版本仓库，修改暂存区（清空暂存区），保留工作区

git remote -v :查看已连接的远程仓库

```
管理员: C:\Windows\System32\cmd.exe
D:\test>git remote -v
origin  git@192.168.1.8:/home/git/test.git (fetch)
origin  git@192.168.1.8:/home/git/test.git (push)
D:\test>
```

ssh -T 用户名@主机地址 : 本地仓库连接远程仓库

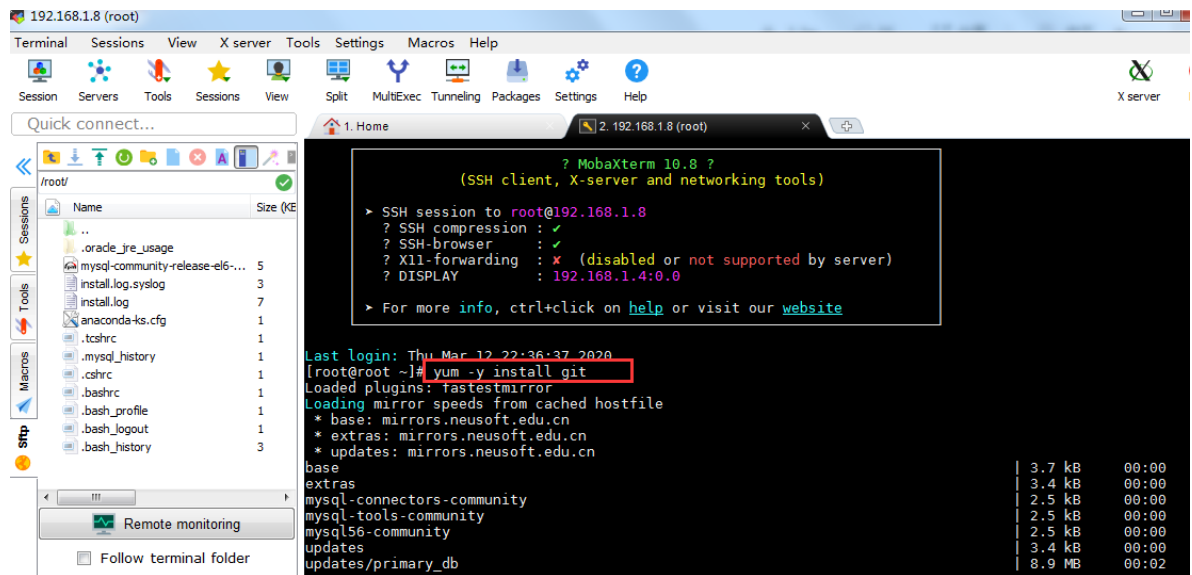
```
MINGW64:/
Administrator@AUTOBVT-2ESCAGN MINGW64 /
$ ssh -T git@192.168.1.8
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
RSA key fingerprint is SHA256:5idPDeSt8v6MLNKHnCn2X3G6uUGoY40TakCBCSpJ1Fg.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.

Administrator@AUTOBVT-2ESCAGN MINGW64 /
$ ssh -T git@192.168.1.8
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
RSA key fingerprint is SHA256:5idPDeSt8v6MLNKHnCn2X3G6uUGoY40TakCBCSpJ1Fg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.8' (RSA) to the list of known hosts.
fatal: What do you think I am? A shell?

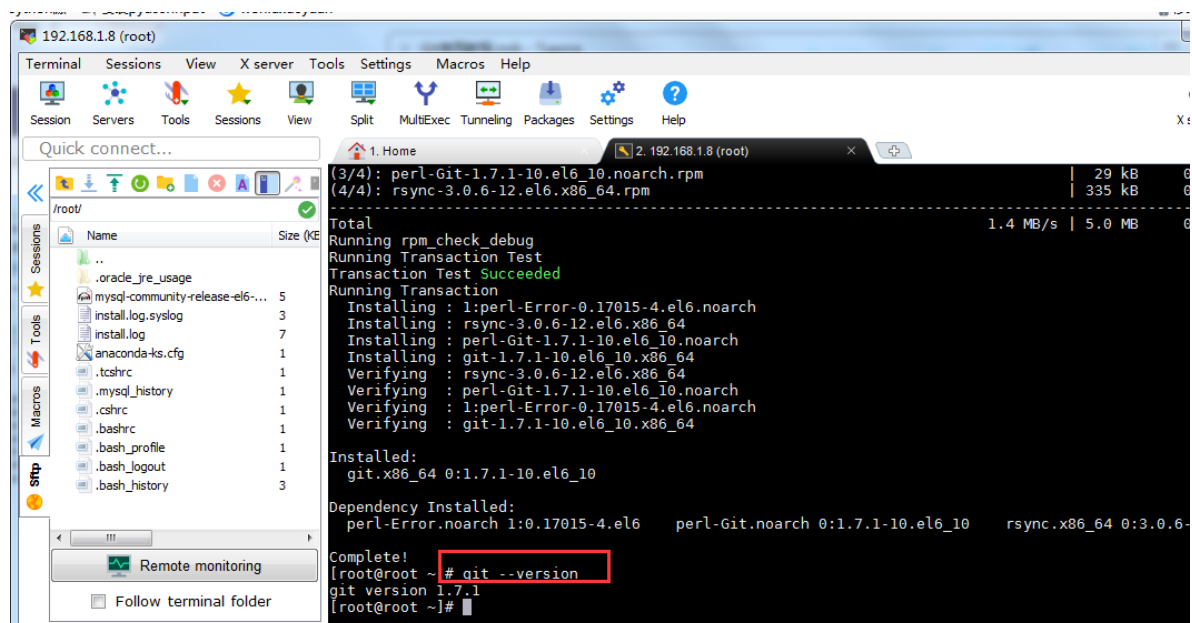
Administrator@AUTOBVT-2ESCAGN MINGW64 /
$
```

## Linux系统配置git服务器，模拟远程仓库

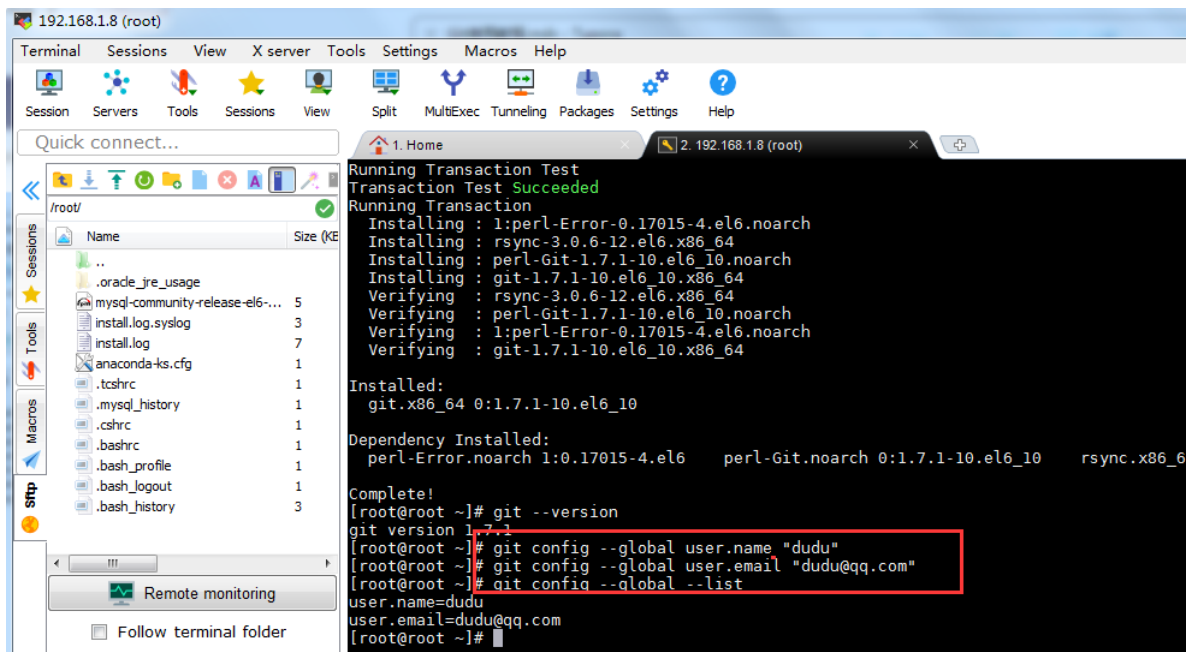
yum -y install git --安装Linux\_Git



git --version :检查是否安装成功

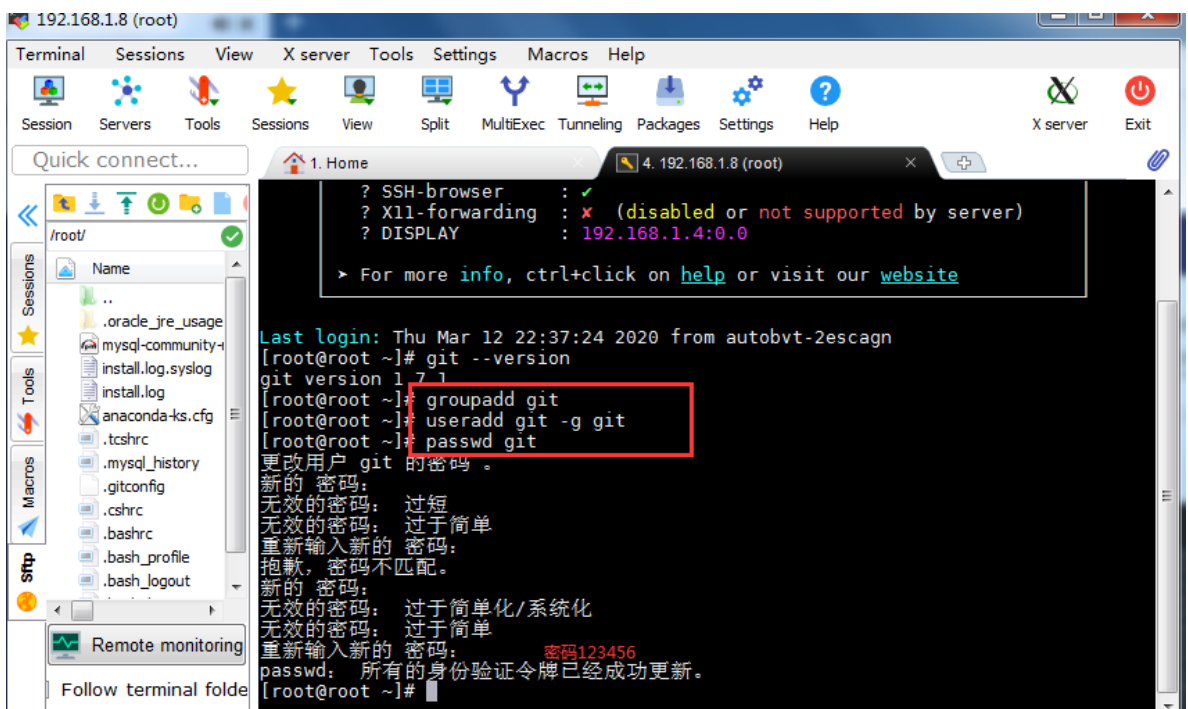


使用前的配置设置



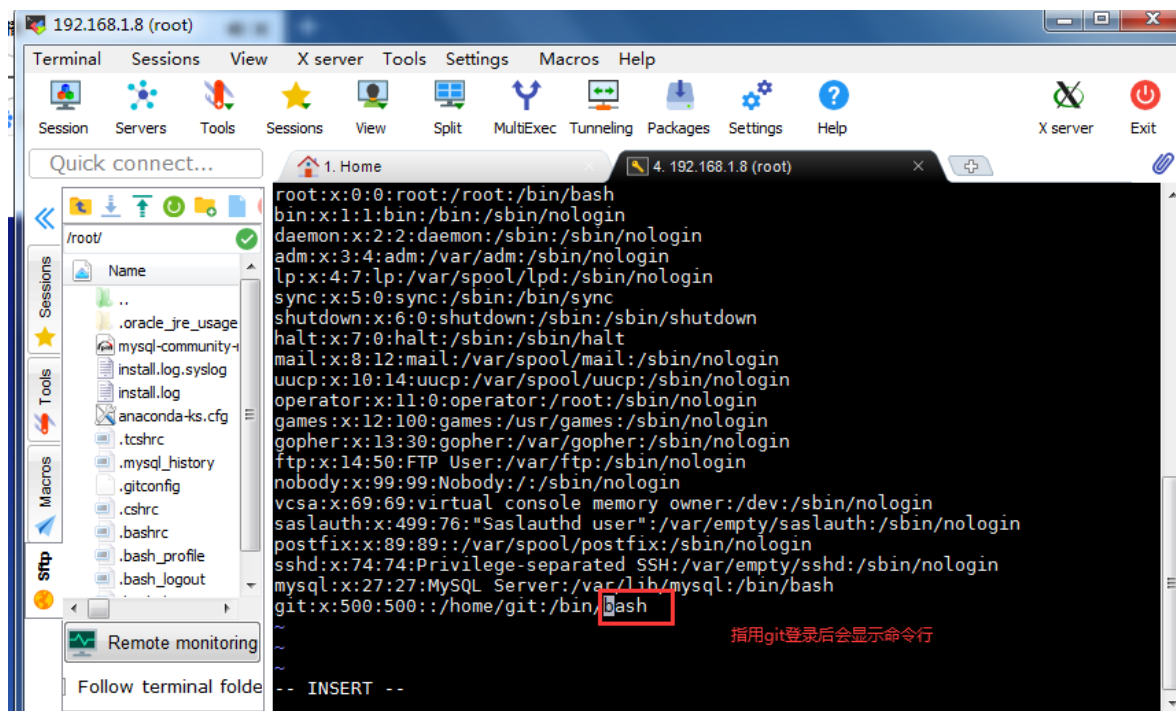
groupadd git 创建用户组 useradd git -g git 创建用户 passwd git 创建密码 :git123456

为了防止分享的文件被删除，所以创建专门的linux用户

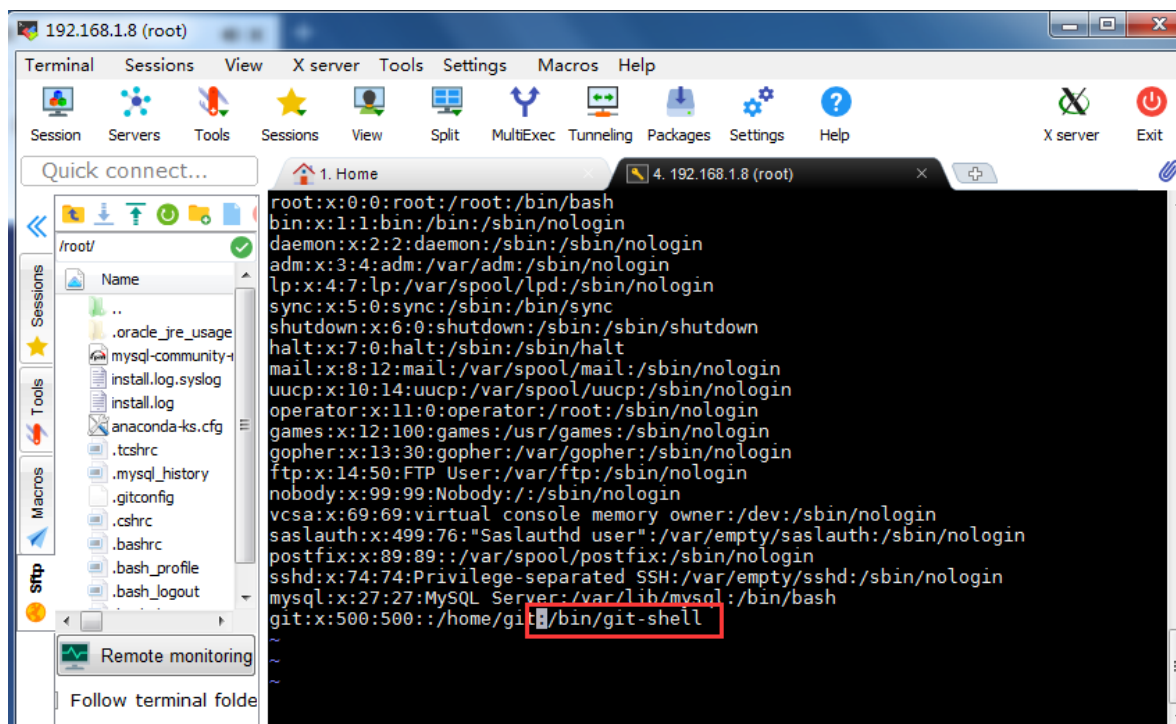


vi /etc/passwd:修改文件使远程连接不会显示命令行

注：由于版本差异，有可能你需要改为:/usr/bin/git-shell



使远程不会出现命令行



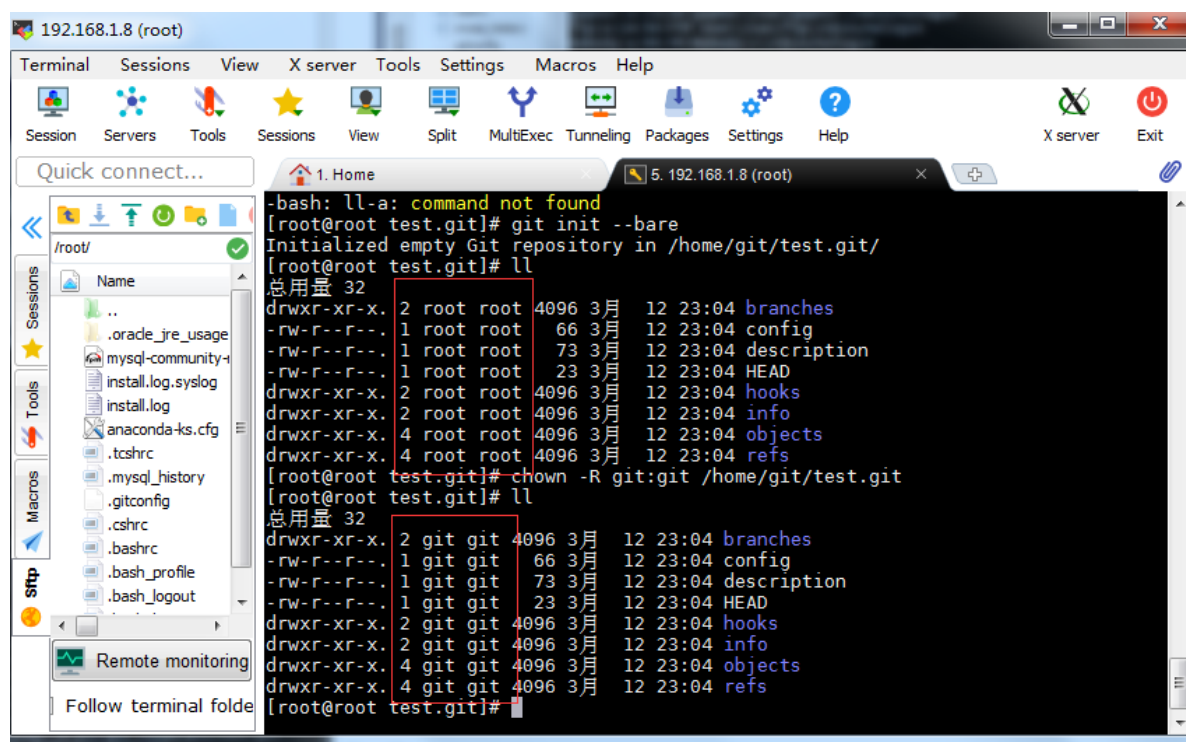
mkdir /home/git/test.git ---创建远程仓库 git init --bare : 设置为裸仓库

git init --bare建立裸仓库，没有工作区和暂停区

只能分享，不能做add和commit的动作，可以做cd check out 和 push

```
[root@root test.git]# git init --bare
Initialized empty Git repository in /home/git/test.git/
[root@root test.git]# ll
总用量 32
drwxr-xr-x. 2 root root 4096 3月 12 23:04 branches
-rw-r--r--. 1 root root 66 3月 12 23:04 config
-rw-r--r--. 1 root root 73 3月 12 23:04 description
-rw-r--r--. 1 root root 23 3月 12 23:04 HEAD
drwxr-xr-x. 2 root root 4096 3月 12 23:04 hooks
drwxr-xr-x. 2 root root 4096 3月 12 23:04 info
drwxr-xr-x. 4 root root 4096 3月 12 23:04 objects
drwxr-xr-x. 4 root root 4096 3月 12 23:04 refs
[root@root test.git]#
```

chown -R git:git /home/git/test.git : 修改为git权限



ssh -T git@192.168.1.8 连接远程仓库

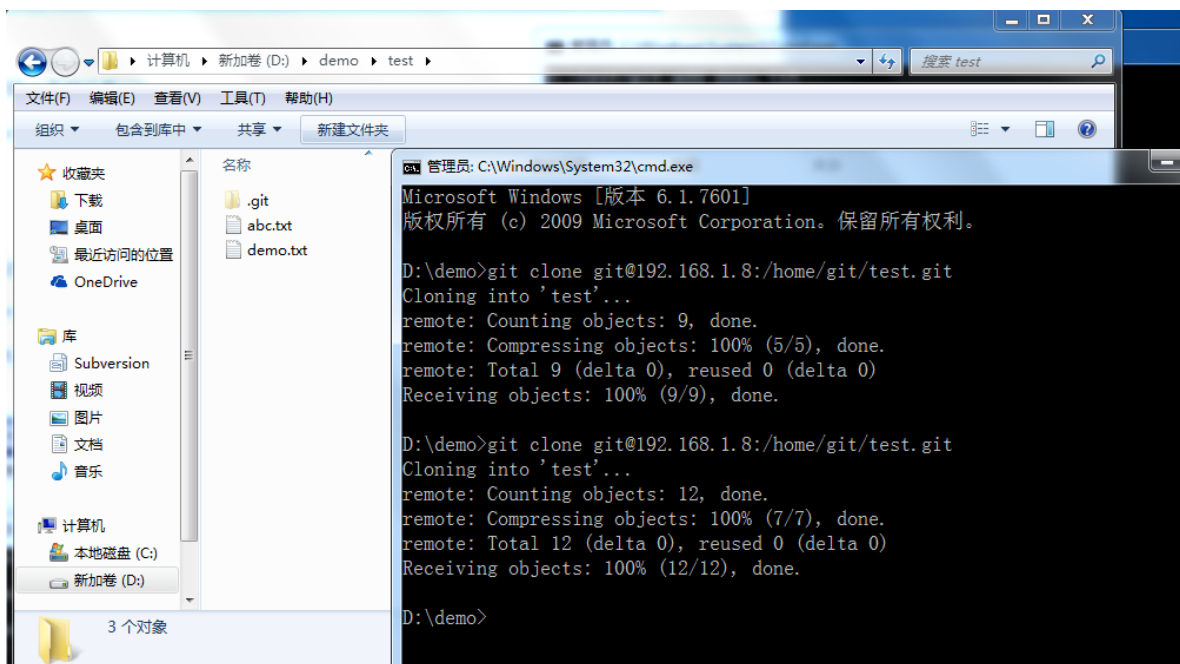


```
Administrator@AUTOBVT-2ESCAGN MINGW64 /
$ ssh -T git@192.168.1.8
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
RSA key fingerprint is SHA256:5idPDeSt8v6MLNKHnCN2X3G6uUGoY40TakCBCSpJ1Fg.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.

Administrator@AUTOBVT-2ESCAGN MINGW64 /
$ ssh -T git@192.168.1.8
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
RSA key fingerprint is SHA256:5idPDeSt8v6MLNKHnCN2X3G6uUGoY40TakCBCSpJ1Fg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.8' (RSA) to the list of known hosts.
fatal: What do you think I am? A shell?

Administrator@AUTOBVT-2ESCAGN MINGW64 /
$
```

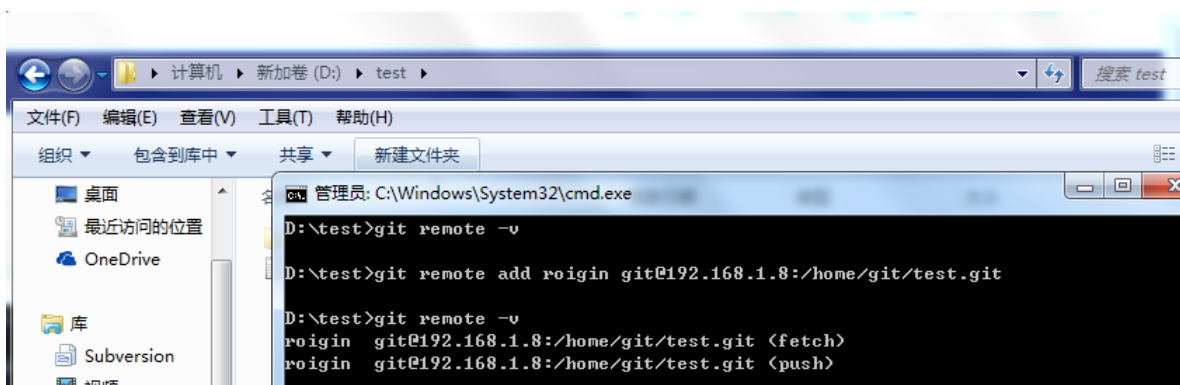
git clone git@192.168.1.8:/home/git/test.git 在本地的文件夹中克隆（本地没有git的目录下）



git remote add origin git@192.168.1.8:/home/git/test.git 本地仓库关联远程仓库

url结构：git@192.168.1.8:/home/git/test.git

在本地已有的项目跟远程仓库关联(本地已经是.git),关联前可以git remote -v查看



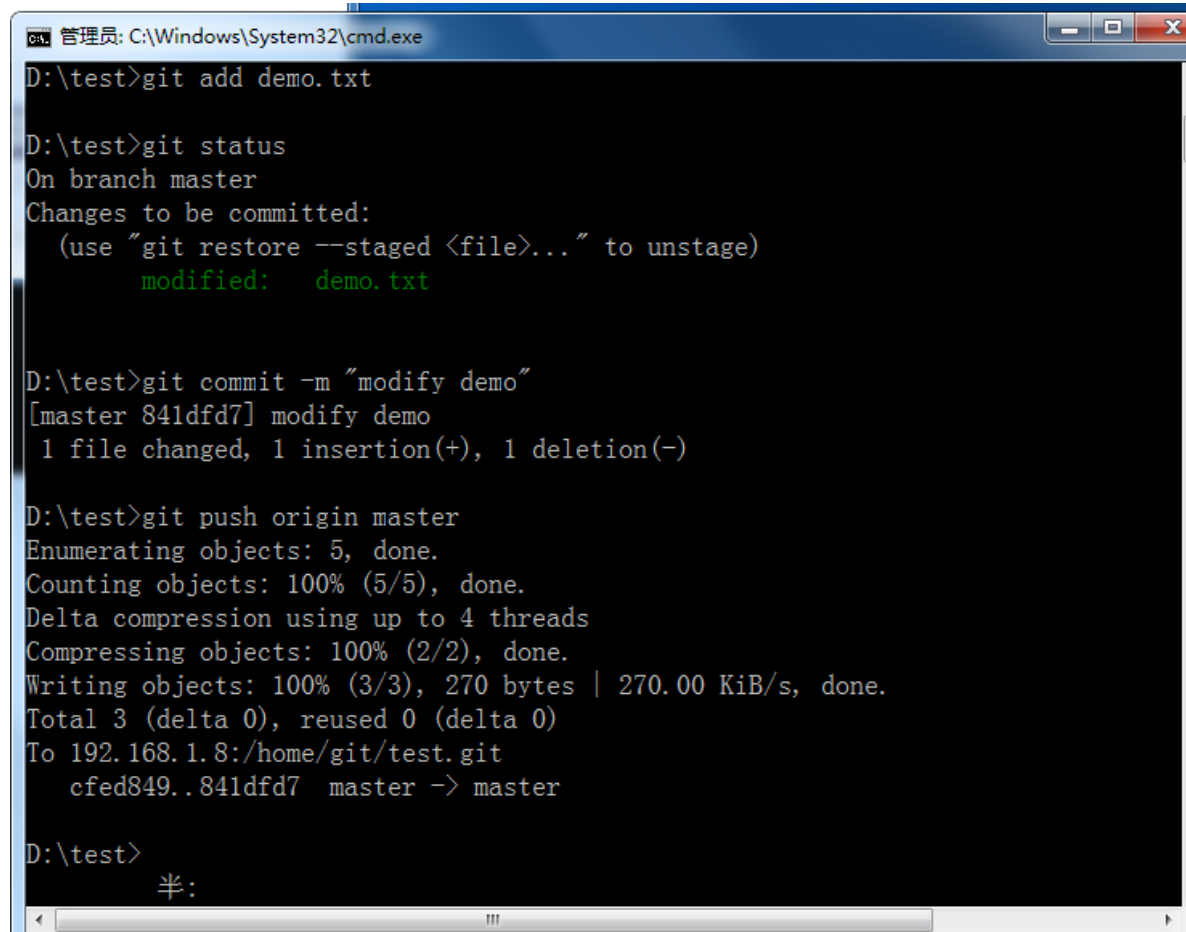


**git push origin master** 将本地仓库文件分享到远程仓库

git push [-u] 远程仓库 本地分支：远程分支

-u参数可选可不选,如果使用后,在其之后的push就只需要输入git push

才下是免密情况下,没有输密码



```
管理员: C:\Windows\System32\cmd.exe
D:\test>git add demo.txt

D:\test>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   demo.txt

D:\test>git commit -m "modify demo"
[master 841dfd7] modify demo
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\test>git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 270 bytes | 270.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To 192.168.1.8:/home/git/test.git
   cfed849..841dfd7  master -> master

D:\test>
```

**git pull** 远程仓库 远程分支：本地分支 用于拉取数据

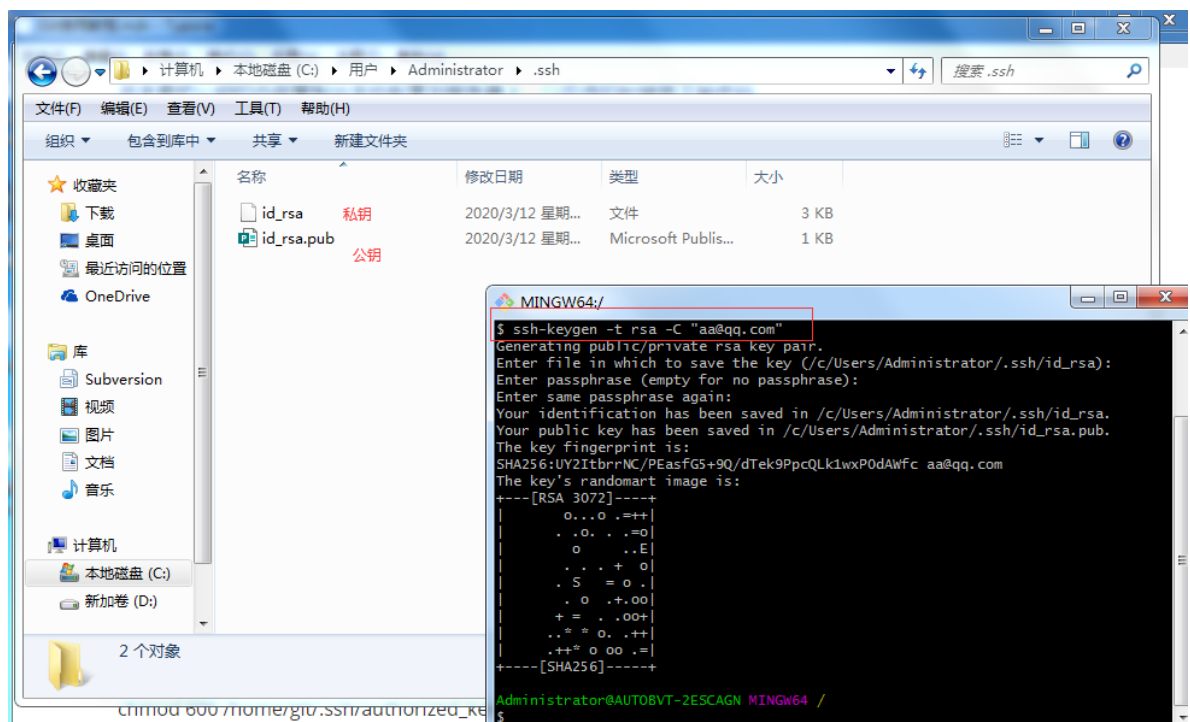
git pull = git fetch + git merge

**ssh-keygen -t rsa -C "aa@qq.com"** 制作公钥私钥, 开通免密,一直回车, email是指本地仓库的 email

在windows下生成一个证书 (也就是客户端)

回车, 再回车, 再回车, 会生成一个证书

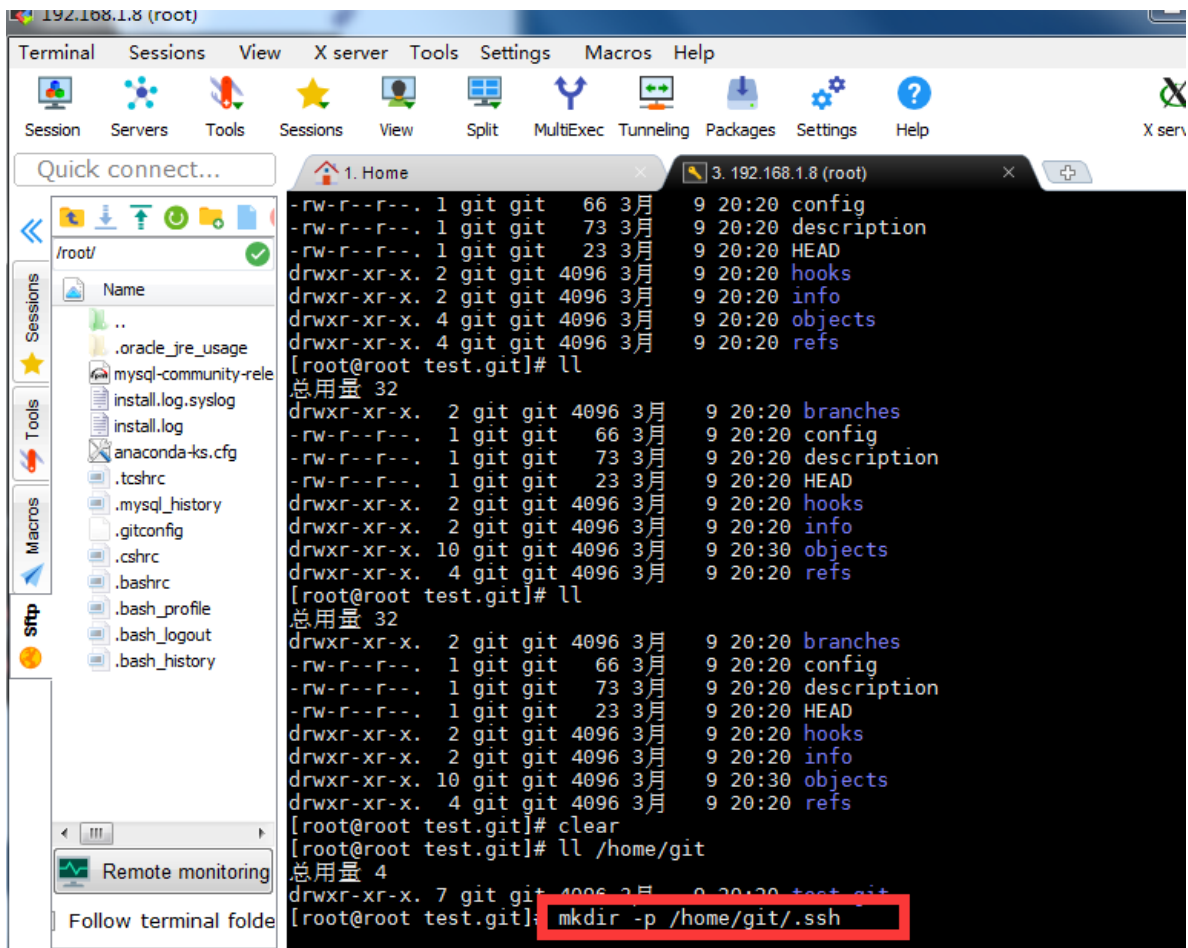
将来要把公钥的内容复制出来给配置到服务器上, 以后传的时候就不输密码,



在linux服务器（远程仓库）上配置公钥

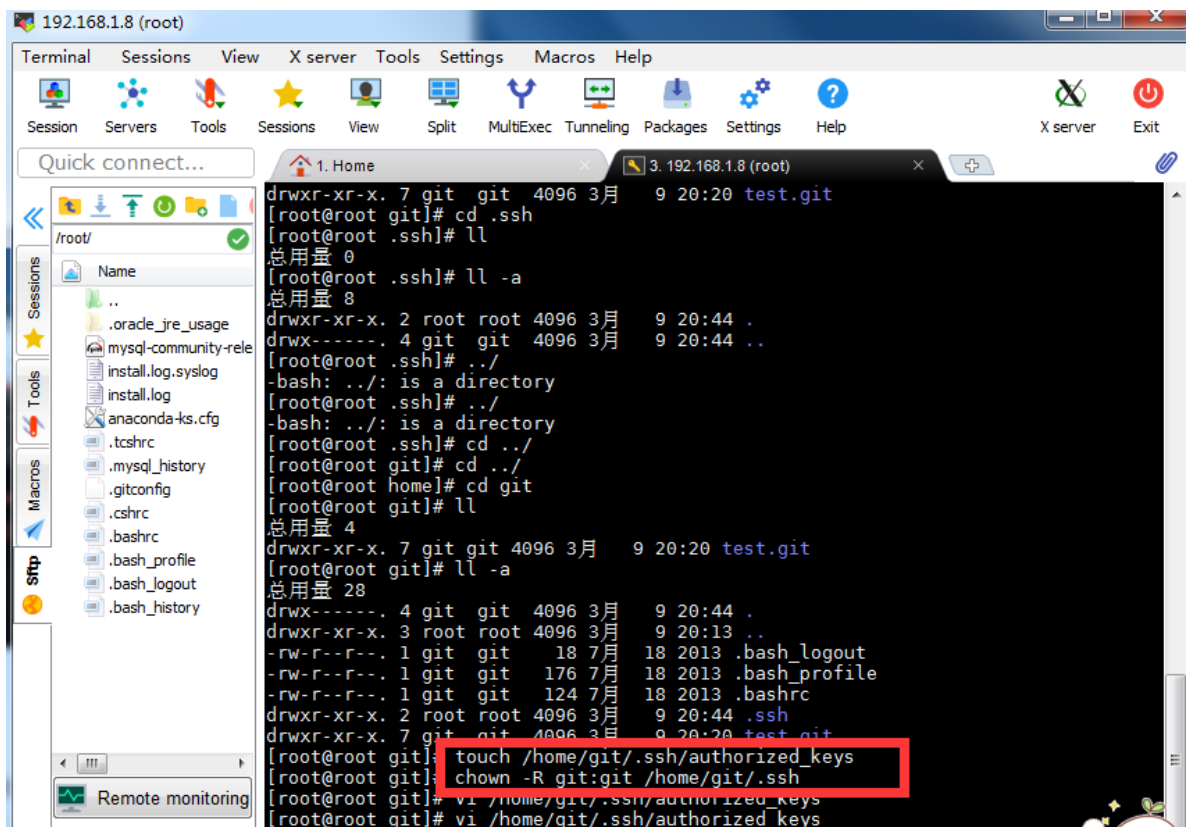
`ll /home/git`

`mkdir -p /home/git/.ssh` 回车，必须是.ssh

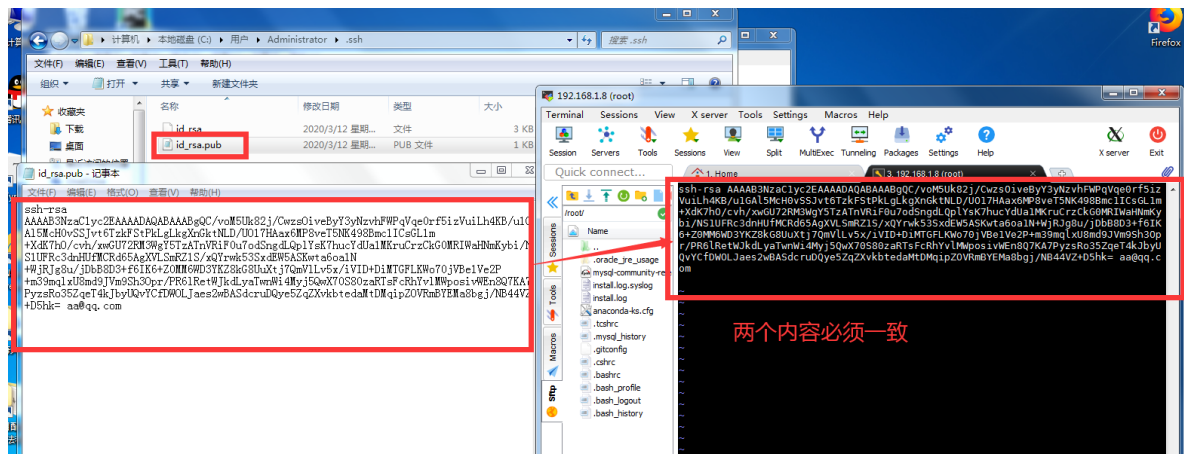


`touch /home/git/.ssh/authorized_keys` 回车 必须是authorized\_keys

`chown -R git:git /home/git/.ssh` 修改为git用户的权限

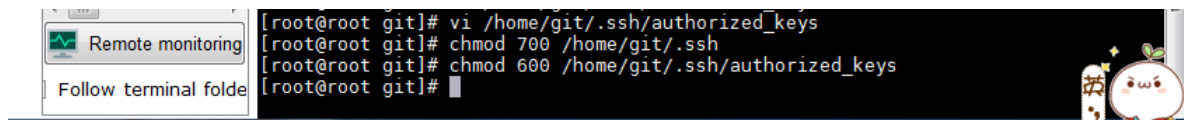


`vi /home/git/.ssh/authorized_keys` 将公钥的内容全选复制进来，存盘退出

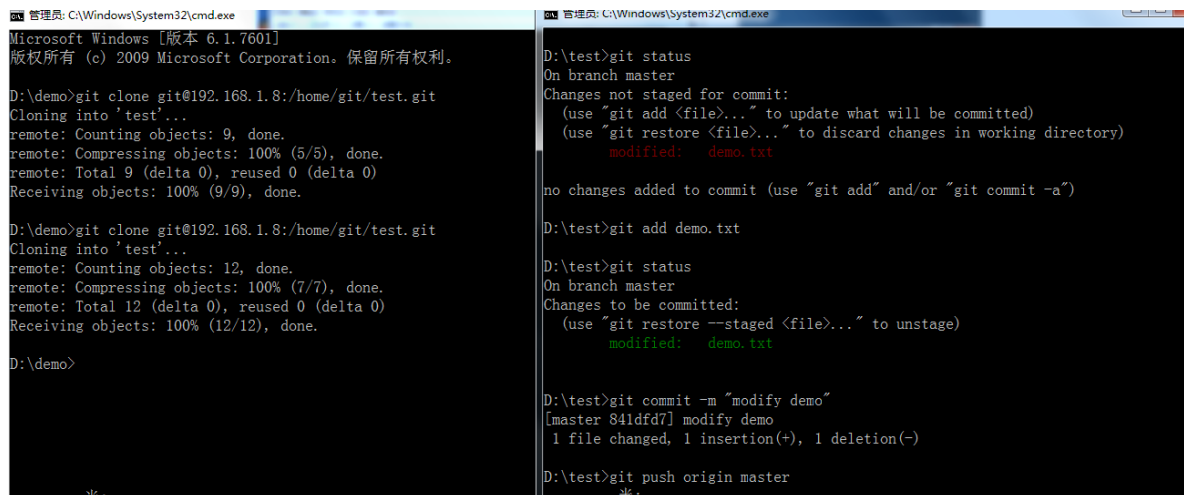


chmod 700 /home/git/.ssh 将这个文件夹做700的授权 必须这样

chmod 600 /home/git/.ssh/authorized\_keys 将这个文件做600的授权 必须这样



在本地仓库使用clone或push都不用输密码了



## 四, Git分支

git branch 查看本地分支

git branch -r 查看了远程分支

git branch -a 查看所有分支

```
C:\Windows\System32\cmd.exe
D:\test>git branch
* master

D:\test>git branch -r
origin/master

D:\test>git branch -a
* master
remotes/origin/master

D:\test>
```

**git branch xiaomi** 创建分支，但head还是在当前分支

**git branch -d xiaomi** 删除分支

如果分支已经有版本（提交），并未执行过合并（merge）的动作，删除失败

**git branch -D 分支名** 强制删除

**git branch -m old dev** 修改分支名

```
D:\test>git branch
* master
  xiaomi

D:\test>git branch -d xiaomi
Deleted branch xiaomi (was 841dfd7).

D:\test>git branch
* master

D:\test>git branch old

D:\test>git branch
* master
  old

D:\test>git branch -m old dev

D:\test>git branch
  dev
* master

D:\test>
```

半:

#### git checkout dev 切换分支

如果git checkout后面跟的分支的名字就变成切换分支的功能

```
D:\test>git branch
  dev
* master

D:\test>git checkout dev
Switched to branch 'dev'

D:\test>git branch
* dev
  master

D:\test>
```

切回去

```
D:\test>git checkout master
Switched to branch 'master'

D:\test>git branch
  checkout
  dev
* master

D:\test>
```

git checkout -b xiaomi 创建分支并将head 切换到分支上

-b 表示指定分支名如果在，那么就切换，如果不存在就先创建分支，然后再切换

```
管理员: C:\Windows\System32\cmd.exe

D:\test>git checkout -b xiaomi
Switched to a new branch 'xiaomi'

D:\test>git branch
  checkout
  dev
  master
* xiaomi

D:\test>
```

