# Git使用

## 一、初学笔录

1、

Administrator@USER-20150907XD MINGW64 ~

$ mkdir .ssh

$ ls –a

$ cd .ssh/

Administrator@USER-20150907XD MINGW64 ~/.ssh

$ ssh-keygen -t rsa -C "gy@zhubaoyi.cn"

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/Administrator.USER-20150907XD/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/Administrator.USER-20150907XD/.ssh/id\_rsa.

Your public key has been saved in /c/Users/Administrator.USER-20150907XD/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:lUtYPCYuHwJTZvG+B+WHEKti6E0wtsRjhOkraOrbjL4 gy@zhubaoyi.cn

The key's randomart image is:

+---[RSA 2048]----+

| o. .=.... |

|oo oo ..=+. |

|. O o .=o=. |

| = \* ooo\* o |

|. + + .+S.+ . |

|o+ + . .o . |

|+ . . . . |

|. + . |

|oEoo |

+----[SHA256]-----+

$ ls

id\_rsa id\_rsa.pub

$ vim id\_rsa.pub

$ cd e:

Administrator@USER-20150907XD MINGW64 /e

$ mkdir git\_pro

$ cd git\_pro/

Administrator@USER-20150907XD MINGW64 /e/git\_pro

$ git clone git@192.168.1.125:zhangxun/zhubaoyi.git

## 二、基本常用命令

1、 git init 把这个目录变成git可以管理的仓库

2、git add readme.txt添加到暂存区里面去

3、git commit告诉Git，把文件提交到仓库。

4、git status来查看是否还有文件未提交

5、git diff readme.txt文件到底改了什么内容

6、 git log查看下历史记录

git log --pretty=oneline

7、回退到上一个版本

git reset  –hard HEAD^

git reset  –hard HEAD^^ 回退到上2个版本

git reset  –hard HEAD~100回退到前100个版本

8、git reflog可获取到版本号

9、git reset  --hard 6fcfc89来恢复(增加内容XXXX的版本号是 6fcfc89.)

Git基本常用命令如下：

mkdir： XX (创建一个空目录 XX指目录名)

pwd： 显示当前目录的路径。

git init 把当前的目录变成可以管理的git仓库，生成隐藏.git文件。

git add XX 把xx文件添加到暂存区去。

git commit –m “XX” 提交文件 –m 后面的是注释。

git status 查看仓库状态

git diff XX 查看XX文件修改了那些内容

git log 查看历史记录

git reset –hard HEAD^ 或者 git reset –hard HEAD~ 回退到上一个版本

(如果想回退到100个版本，使用git reset –hard HEAD~100 )

cat XX 查看XX文件内容

git reflog 查看历史记录的版本号id

git checkout — XX 把XX文件在工作区的修改全部撤销。

git rm XX 删除XX文件

git remote add origin https://github.com/tugenhua0707/testgit 关联一个远程库

git push –u(第一次要用-u 以后不需要) origin master 把当前master分支推送到远程库

git clone https://github.com/tugenhua0707/testgit 从远程库中克隆

git checkout –b dev 创建dev分支 并切换到dev分支上

git branch 查看当前所有的分支

git checkout master 切换回master分支

git merge dev 在当前的分支上合并dev分支

git branch –d dev 删除dev分支

git branch name 创建分支

git stash 把当前的工作隐藏起来 等以后恢复现场后继续工作

git stash list 查看所有被隐藏的文件列表

git stash apply 恢复被隐藏的文件，但是内容不删除

git stash drop 删除文件

git stash pop 恢复文件的同时 也删除文件

git remote 查看远程库的信息

git remote –v 查看远程库的详细信息

git push origin master Git会把master分支推送到远程库对应的远程分支上

## 三、Git撤销修改和删除文件操作

### ****A：撤销修改****

在readme.txt文件里面增加一行 内容为555555, 在我未提交之前，发现添加有误，得马上恢复以前的版本，有如下几种方法可以做修改：

第一：如果我知道要删掉那些内容的话，直接手动更改去掉那些需要的文件，然后add添加到暂存区，最后commit掉。

第二：我可以按以前的方法直接恢复到上一个版本。使用 git reset  –hard HEAD^

--------------不使用上面2种方法，直接使用撤销命令呢？------------------------

git checkout  --  readme.txt 意思是: 把readme.txt文件在工作区做的修改全部撤销，这里有2种情况，如下：

1. readme.txt自动修改后，还没有放到暂存区，使用 撤销修改就回到和版本库一模一样的状态。
2. 另外一种是readme.txt已经放入暂存区了，接着又作了修改，撤销修改就回到添加暂存区后的状态。

对于第二种情况，假如对readme.txt添加一行 内容为6666666，git add 增加到暂存区后，接着添加内容77777通过撤销命令让其回到暂存区后的状态。如下所示：

git checkout -- readme.txt

### ****B：删除文件****

1、假如向版本库a目录添加一个文件1.txt,然后提交了。如何删除？

方法：先 rm 1.txt ，再执行git commit命令 提交掉。

2、没有commit之前，如果要在版本库中恢复1.txt？

方法： git checkout  -- 1.txt

## 四、创建SSH Key

打开命令行，输入如下命令：

ssh-keygen  -t rsa –C “youremail@example.com”, 由于我本地此前运行过一次，所以本地有，如下所示：



id\_rsa是私钥，不能泄露出去，id\_rsa.pub是公钥，可以放心地告诉任何人。

## 五、远程库

### 1、把一个已有的本地仓库与远程库关联

在本地的testgit仓库下运行命令：

git remote add origin https://github.com/tugenhua0707/testgit.git

### 2、把本地库分支master内容推送到远程仓库去

在本地的testgit仓库下运行命令：

git push –u origin master

加上了 –u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令。

从现在起，只要本地作了提交，就可以通过如下命令：

git push origin master

把本地master分支的最新修改推送到github上了，现在你就拥有了真正的分布式版本库了。

### 3、从远程库克隆

远程库已经准备好了，下一步是使用命令git clone克隆一个本地库了。如下所示：



接着在我本地目录下 生成testgit2目录了，如下所示：



### 4、查看关联

当你从远程库克隆时候，实际上Git自动把本地的master分支和远程的master分支对应起来了，并且远程库的默认名称是origin。

1. 要查看远程库的信息 使用 git remote
2. 要查看远程库的详细信息 使用 git remote –v

如下演示：



## 六、分支

在  版本回填退里，你已经知道，每次提交，Git都把它们串成一条时间线，这条时间线就是一个分支。截止到目前，只有一条时间线，在Git里，这个分支叫主分支，即master分支。HEAD严格来说不是指向提交，而是指向master，master才是指向提交的，所以，HEAD指向的就是当前分支。

### 查看分支

git branch查看分支，会列出所有的分支，当前分支前面会添加一个星号。

### 2、创建、切换分支



git checkout 命令加上 –b参数表示创建并切换，相当于如下2条命令

git branch dev

git checkout dev

### 3、合并分支

把dev分支上的内容合并到分支master上了，可以在master分支上，使用如下命令 git merge dev 如下所示：



注意到上面的Fast-forward信息，Git告诉我们，这次合并是“快进模式”，也就是直接把master指向dev的当前提交，所以合并速度非常快。

### 4、删除分支

删除dev分支了，操作如下：



### 5、命令总结

总结创建与合并分支命令如下：

   查看分支：git branch

   创建分支：git branch name

   切换分支：git checkout name

创建+切换分支：git checkout –b name

合并某分支到当前分支：git merge name

删除分支：git branch –d name

### 6、分支管理策略

git一般使用”Fast forward”模式，在这种模式下，删除分支后，会丢掉分支信息，现在我们来使用带参数 –no-ff来禁用”Fast forward”模式。

首先我们来做demo演示下：

1. 创建一个dev分支。
2. 修改readme.txt内容。
3. 添加到暂存区。
4. 切换回主分支(master)。
5. 合并dev分支，使用命令 git merge –no-ff  -m “注释” dev
6. 查看历史记录

截图如下：



**分支策略：**首先master主分支应该是非常稳定的，也就是用来发布新版本，一般情况下不允许在上面干活，干活一般情况下在新建的dev分支上干活，干完后，比如上要发布，或者说dev分支代码稳定后可以合并到主分支master上来。

### 7、隐藏工作

stash功能，可以把当前工作现场 ”隐藏起来”，等以后恢复现场后继续工作。如下：



工作现场去哪里呢？我们可以使用命令 git stash list来查看下。如下：



工作现场还在，Git把stash内容存在某个地方了，但是需要恢复一下，可以使用如下2个方法：

1. git stash apply恢复，恢复后，stash内容并不删除，你需要使用命令git stash drop来删除。
2. 另一种方式是使用git stash pop,恢复的同时把stash内容也删除了。

         演示如下



## 七、多人协作

### 1、推送分支

      推送分支就是把该分支上所有本地提交到远程库中，推送时，要指定本地分支，这样，Git就会把该分支推送到远程库对应的远程分支上：

      使用命令 git push origin master

比如我现在的github上的readme.txt代码如下：



本地的readme.txt代码如下：



现在我想把本地更新的readme.txt代码推送到远程库中，使用命令如下：



我们可以看到如上，推送成功，我们可以继续来截图github上的readme.txt内容 如下：



可以看到 推送成功了，如果我们现在要推送到其他分支，比如dev分支上，我们还是那个命令 git push origin dev

那么一般情况下，那些分支要推送呢？

1. master分支是主分支，因此要时刻与远程同步。
2. 一些修复bug分支不需要推送到远程去，可以先合并到主分支上，然后把主分支master推送到远程去。

### 2、抓取分支

多人协作时，大家都会往master分支上推送各自的修改。现在我们可以模拟另外一个同事，可以在另一台电脑上（注意要把SSH key添加到github上）或者同一台电脑上另外一个目录克隆，新建一个目录名字叫testgit2

但是我首先要把dev分支也要推送到远程去，如下



接着进入testgit2目录，进行克隆远程的库到本地来，如下：



现在目录下生成有如下所示：



现在我们的小伙伴要在dev分支上做开发，就必须把远程的origin的dev分支到本地来，于是可以使用命令创建本地dev分支：git checkout  –b dev origin/dev

现在小伙伴们就可以在dev分支上做开发了，开发完成后把dev分支推送到远程库时。

如下：



小伙伴们已经向origin/dev分支上推送了提交，而我在我的目录文件下也对同样的文件同个地方作了修改，也试图推送到远程库时，如下：



由上面可知：推送失败，因为我的小伙伴最新提交的和我试图推送的有冲突，解决的办法也很简单，上面已经提示我们，先用git pull把最新的提交从origin/dev抓下来，然后在本地合并，解决冲突，再推送。



*git pull*也失败了，原因是没有指定本地dev分支与远程origin/dev分支的链接，根据提示，设置dev和origin/dev的链接：如下：



这回*git pull*成功，但是合并有冲突，需要手动解决，解决的方法和分支管理中的 解决冲突完全一样。解决后，提交，再push：

我们可以先来看看readme.txt内容了。



现在手动已经解决完了，我接在需要再提交，再push到远程库里面去。如下所示：  


因此：多人协作工作模式一般是这样的：

1. 首先，可以试图用git push origin branch-name推送自己的修改.
2. 如果推送失败，则因为远程分支比你的本地更新早，需要先用git pull试图合并。
3. 如果合并有冲突，则需要解决冲突，并在本地提交。再用git push origin branch-name推送。

## 八、随记

20151203更新分支上最新记录：

38 git status

39 git checkout develop

40 git pull

41 ls

42 git checkout master

43 git pull

44 git branch -a