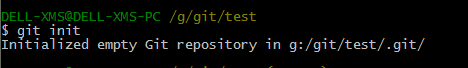
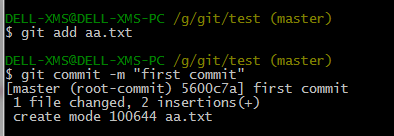
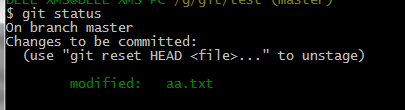
1.创建版本库



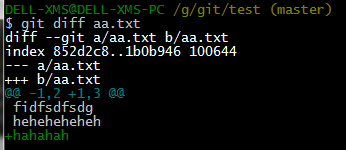
2.提交文件到仓库



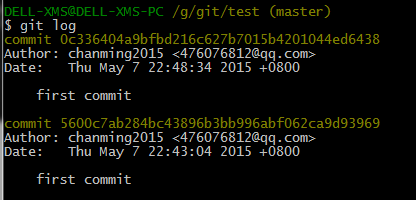
3.查看仓库当前状态



4.查看文件修改详细

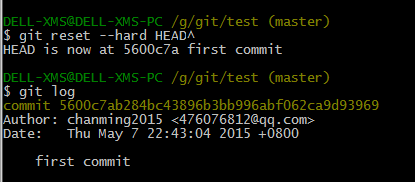


5.查看版本库日志记录

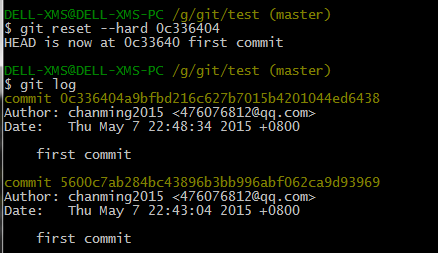


6.版本回退（HEAD表示当前版本，HEAD^表示前一个版本，如果太多用HEAD~100）

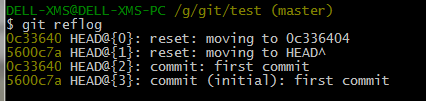
git reset –hard commit\_id



版本前进

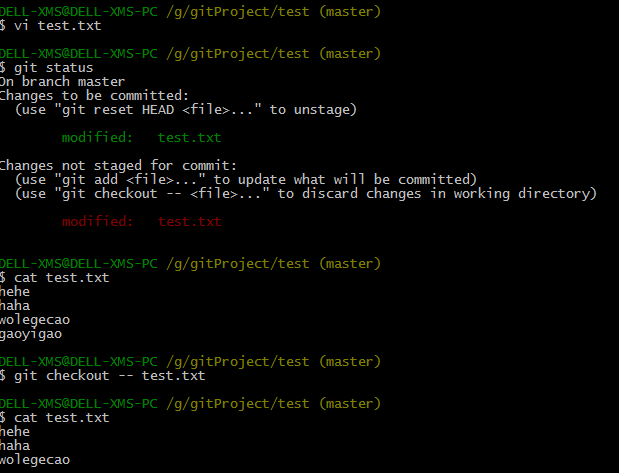


7.查看命令记录



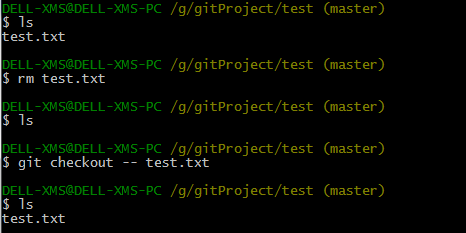
Git和其他版本控制系统如SVN的一个不同之处就是有暂存区的概念。

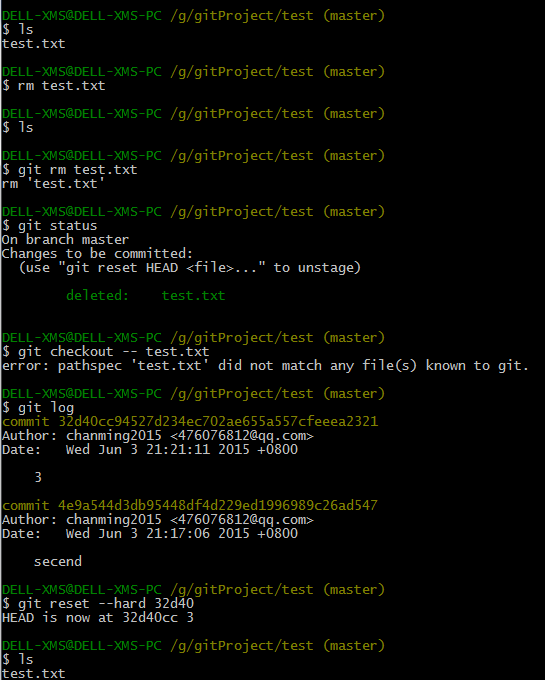
8.撤销修改：（仅限工作区）



撤销暂存区到工作区：git reset HEAD test.txt

9.git checkout其实是用版本库里的版本替换工作区的版本





Rm text.txt 只是删除工作区的文件，还需要git rm test.txt并且commot

10.生成密钥和公钥：ssh-keygen -t rsa -C [476076812@qq.com](mailto:476076812@qq.com)

11. 要关联一个远程库，使用命令git remote add origin git@server-name:path/repo-name.git；

关联后，使用命令git push -u origin master第一次推送master分支的所有内容；

此后，每次本地提交后，只要有必要，就可以使用命令git push origin master推送最新修改；

克隆：git clone git@github.com:chanming2015/xxx.git

12.Git分支：

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>

快速合并某分支到当前分支：git merge <name>

普通合并和提交gt merge --no-ff -m "info" <name>

删除分支：git branch -d <name>

如果要丢弃一个没有被合并过的分支，可以通过git branch -D <name>强行删除。

用git log --graph命令可以看到分支合并图。

修复bug时，我们会通过创建新的bug分支进行修复，然后合并，最后删除；

当手头工作没有完成时，先把工作现场git stash一下，然后去修复bug，修复后，再git stash pop，回到工作现场。

git stash list命令查看保存的工作现场

查看远程库信息，使用git remote -v；

本地新建的分支如果不推送到远程，对其他人就是不可见的；

从本地推送分支，使用git push origin branch-name，如果推送失败，先用git pull抓取远程的新提交；

在本地创建和远程分支对应的分支，使用git checkout -b branch-name origin/branch-name，本地和远程分支的名称最好一致；

建立本地分支和远程分支的关联，使用git branch --set-upstream branch-name origin/branch-name；

从远程抓取分支，使用git pull，如果有冲突，要先处理冲突。

命令git tag < tagname >用于新建一个标签，默认为HEAD，也可以指定一个commit id；

命令 git log --pretty=oneline --abbrev-commit可以查看历史提交的commit id

可以用git show <tagname>查看标签信息

git tag -a <tagname> -m "blablabla..."可以指定标签信息；

命令git tag可以查看所有标签

命令git push origin <tagname>可以推送一个本地标签；

命令git push origin --tags可以推送全部未推送过的本地标签；

命令git tag -d <tagname>可以删除一个本地标签；

命令git push origin :refs/tags/<tagname>可以删除一个远程标签。

让git显示颜色git config --global color.ui true

设置别名git config --global alias.st status（用st代替Status）