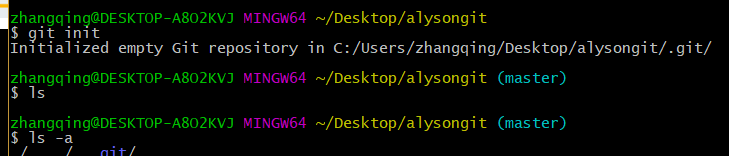


桌面创建一个目录并进入该目录：



# git init





在该目录下创建一个文件，随便写点内容



# git add

಩#把文件添加到仓库

# 

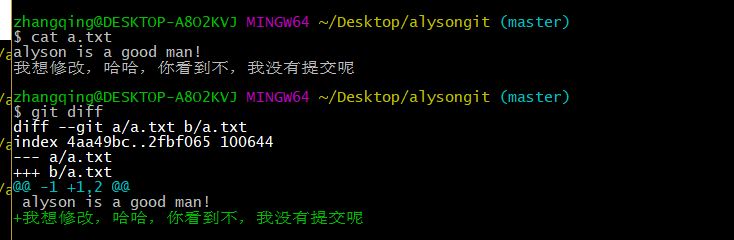
# git commit -m “描述信息”

#把文件提交到仓库



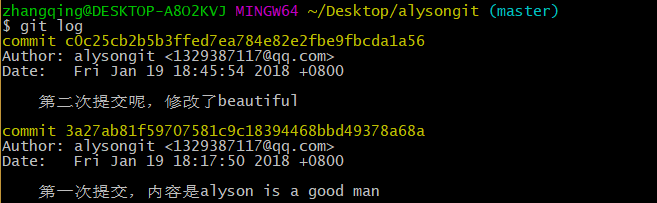
# git diff:

#修改文件，不git add和git commit -m “描述信息”,用git diff可以查看的修改的内容



# git log

#可以显示最近到最远的提交日志

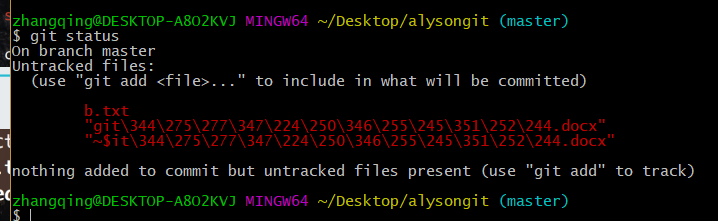


# git log - -pretty=oneline

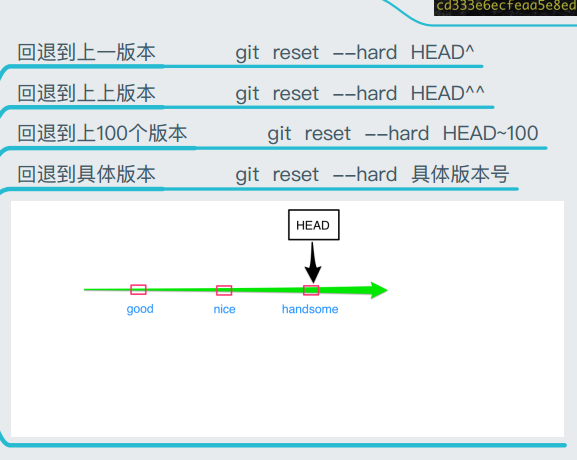


# git status

时刻掌握仓库当前状态：



# git reset - -hard HEAD~6



#回退到前面的版本





# git reflog:

#可以查看所有分支的所有操作记录

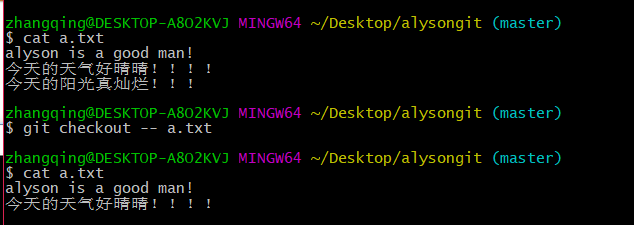


## Q: git reflog 和git log的区别

A: git reflog 可以查看所有分支的所有操作记录（包括（包括commit和reset的操作），包括已经被删除的commit记录，git log则不能察看已经删除了的commit记录

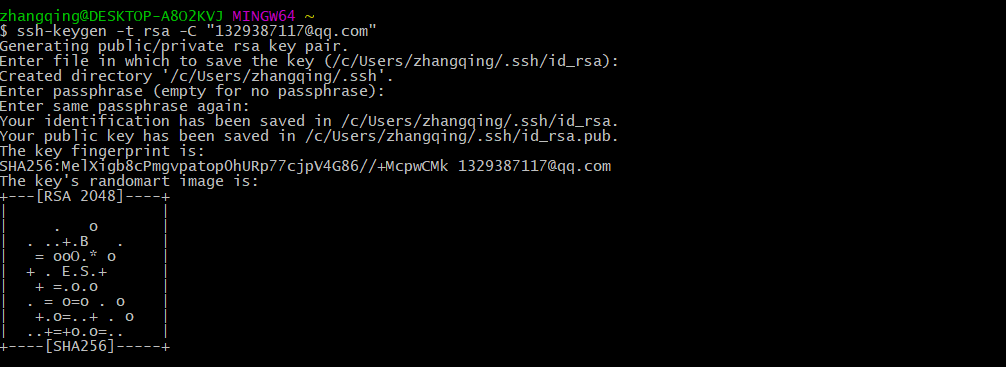
# git checkout -- a.txt

让文件回到最后一次git commit 或者git add 的状态

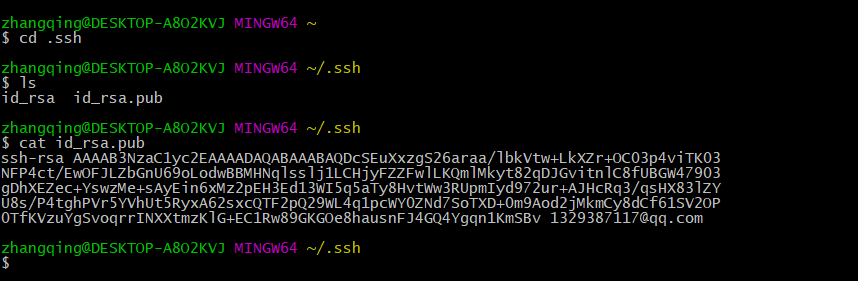


ssh-keygen -t rsa -C [1329387117@qq.com](mailto:1329387117@qq.com)

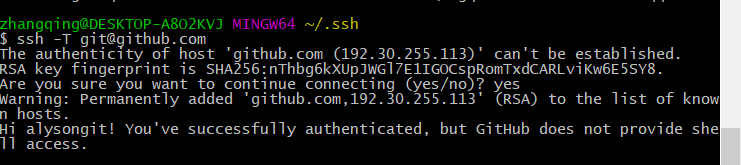
#创建SSH Key

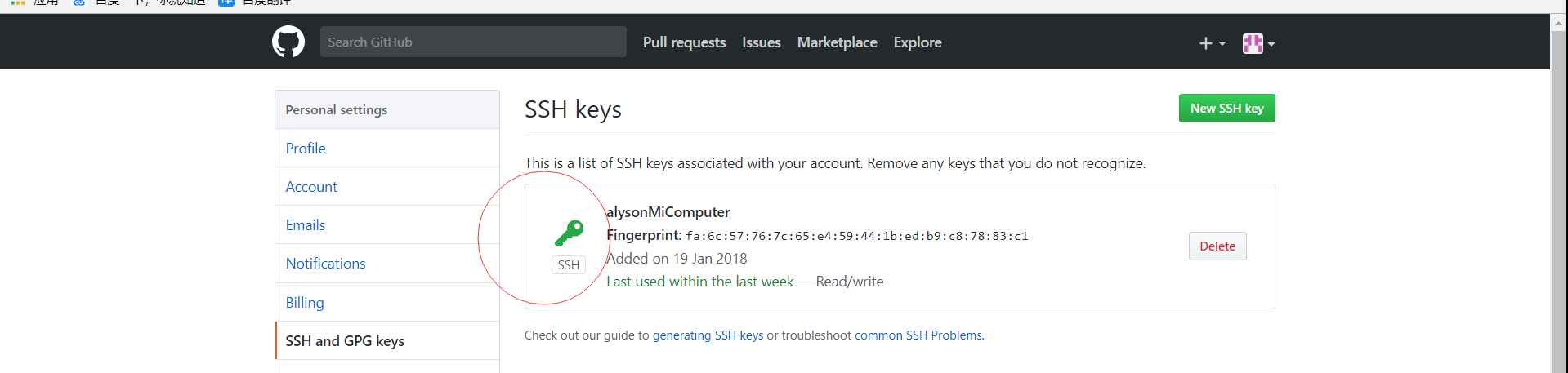


# #进入生成的.ssh的目录下，cat出公钥

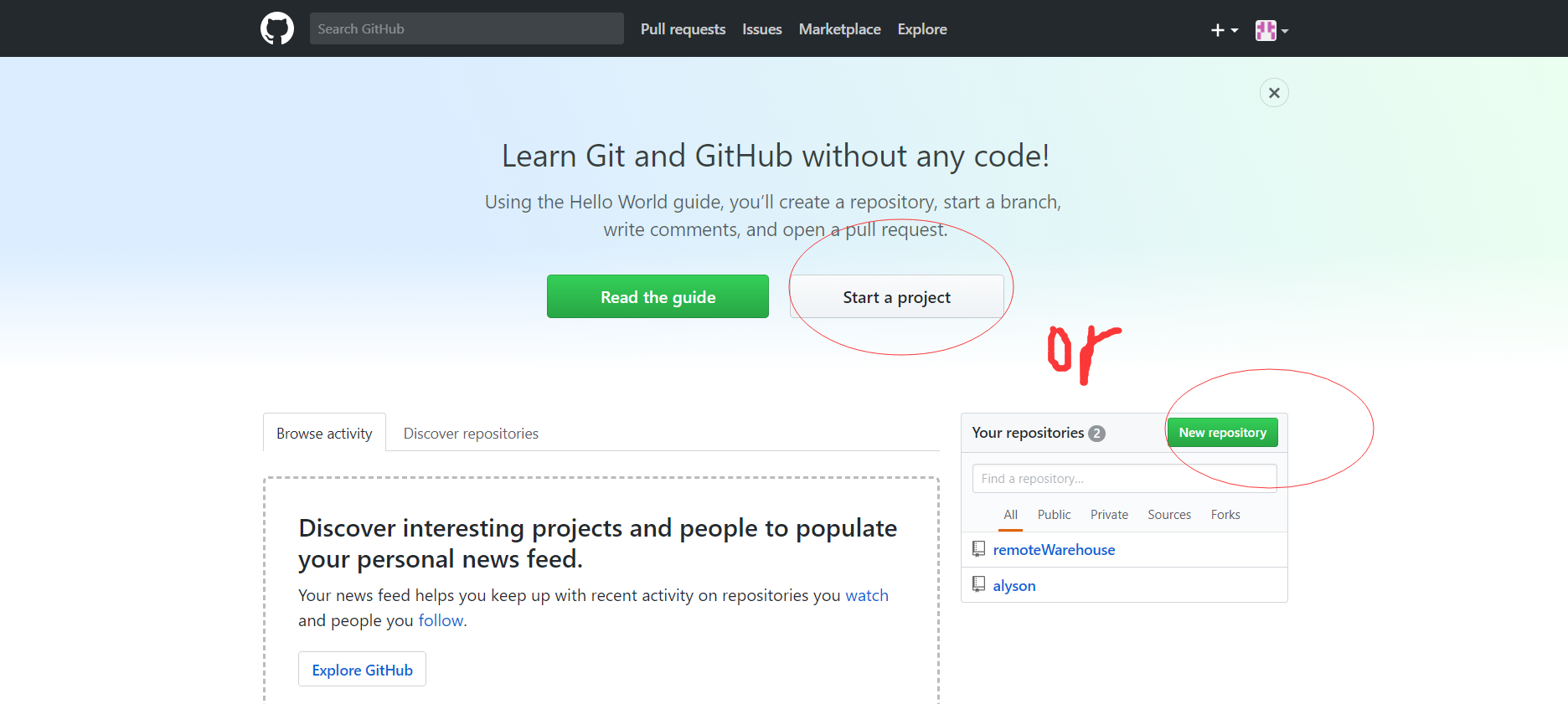


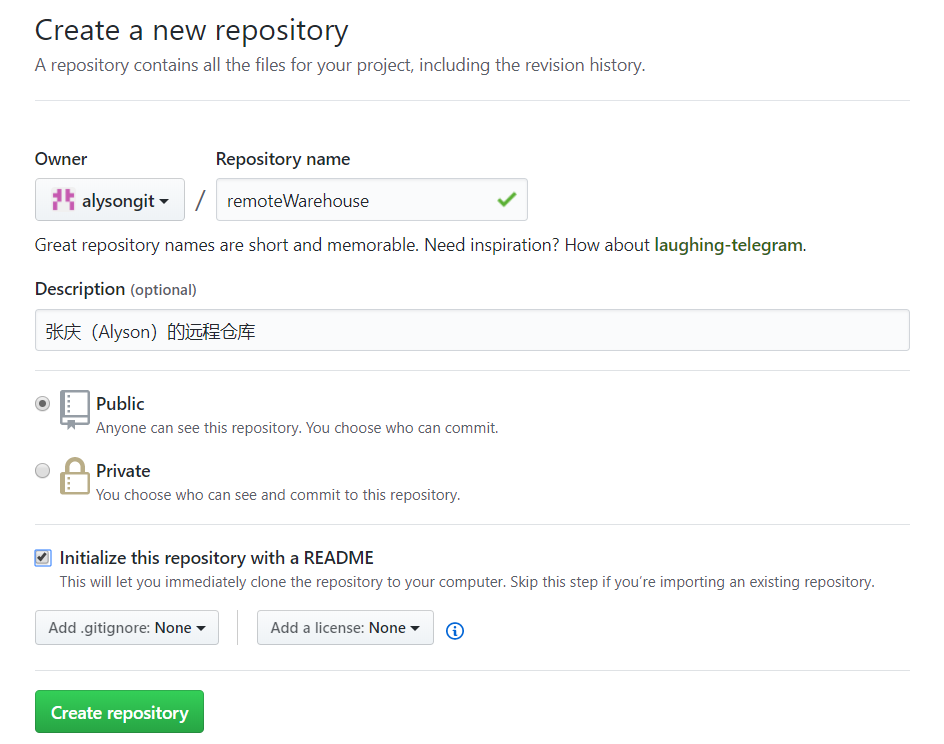
# #判断公钥好不好用，注意第一次一定要输入yes





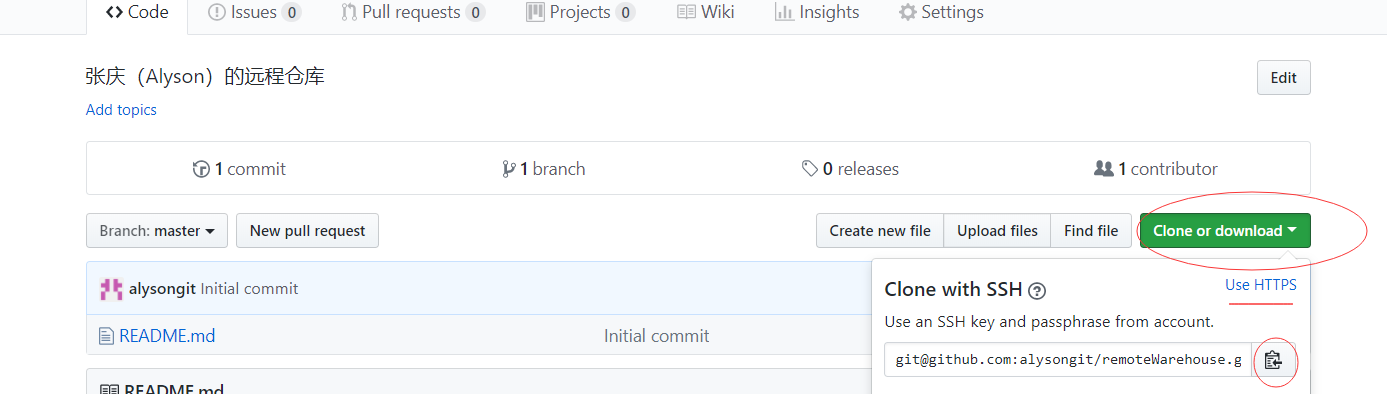
# 创建远程仓库：

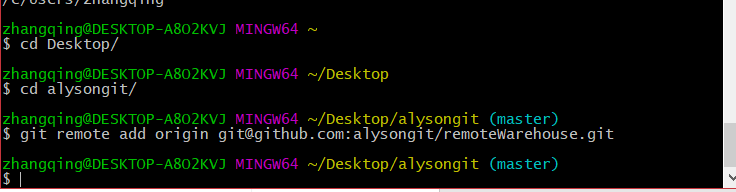




# 关联远程仓库：

## git remote add origin 远程仓库地址





git push origin master#推送本地内容到远程库：

第一次使用需要先将远程库拉到本地

可能本地还有刚修改没上传的需要git add . 然后git commit -m “随便写点注释信息”

解决本地库与远程库不一样的方法：

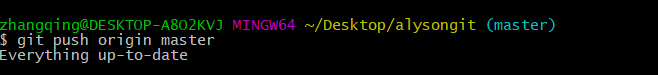
git pull origin master - - allow-related-histories

将远程库拉到本地

git pull origin master

# 推送本地内容到远程库：

git push origin master

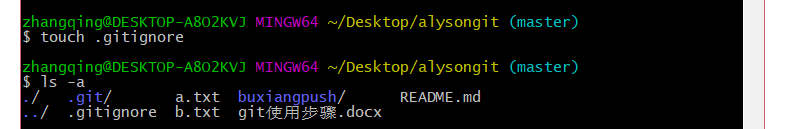


删除关联：

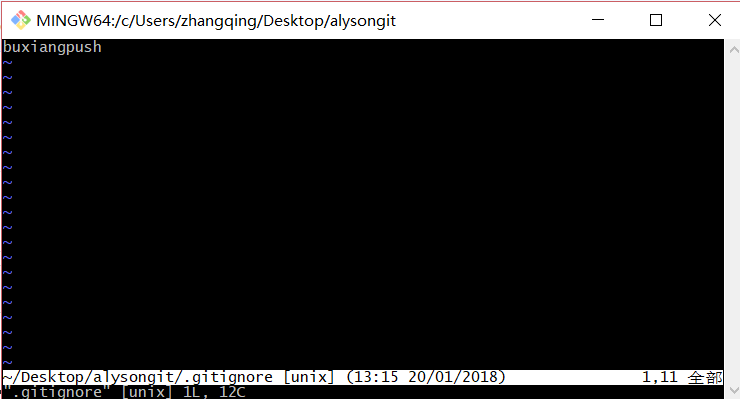
git remote rm origin

不想上传的文件：

touch .gitignore



将不想上传的文件名放在这里



创建分支：

git branch dev

切换分支：

git checkout dev

查看分支：

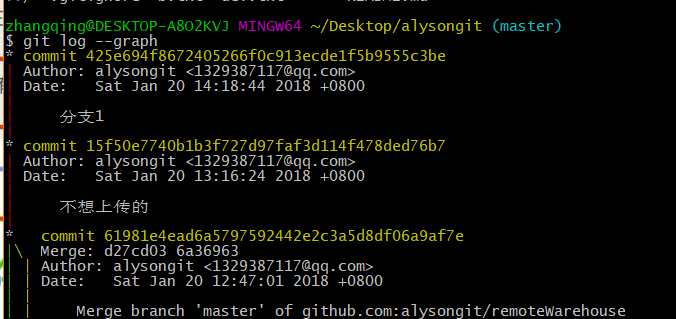
git branch



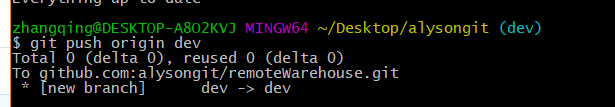
# 合并分支：



git log --graph

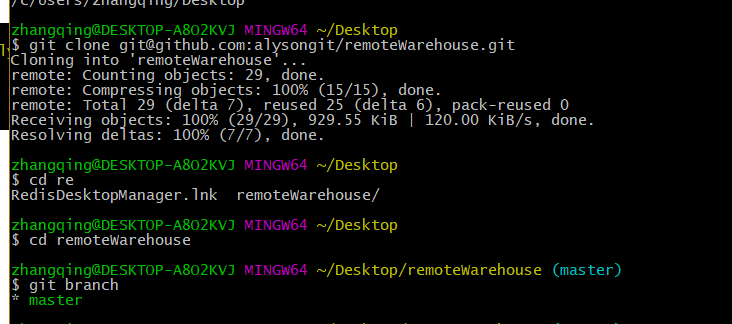


推送分支：



删除分支：

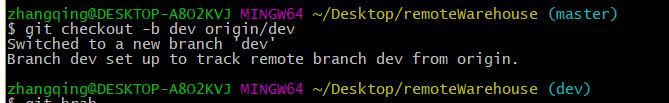
从远程库clone 到本地的时候只有master分支：





# git checkout -b dev origin/dev

# #从远程库下载分支：



打标签：

