## 利用GitExtensions进行本地和远程协同管理

## 工具：

1. GitHub环境搭建安装包（包括GitExtensions、TortoiseGit、TortoiseGit-LanguagePack）
2. GitHub网页版（已经fork了anack）

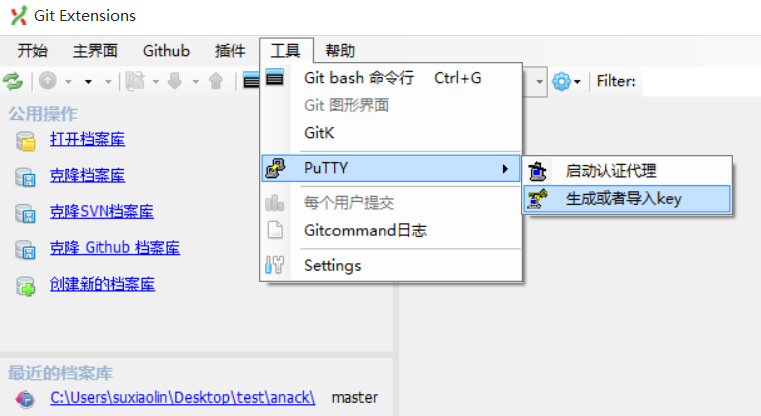
## 软件环境搭建

按顺序安装GitExtensions、TortoiseGit、TortoiseGit汉化补丁

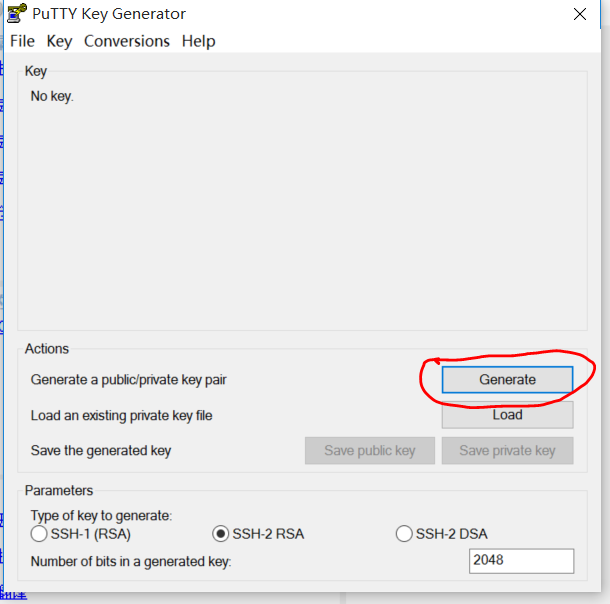
软件安装时一切默认，除了GitExtension安装的时候注意勾选一下安装kdiff3和git两个工具。（注意关闭杀毒软件）

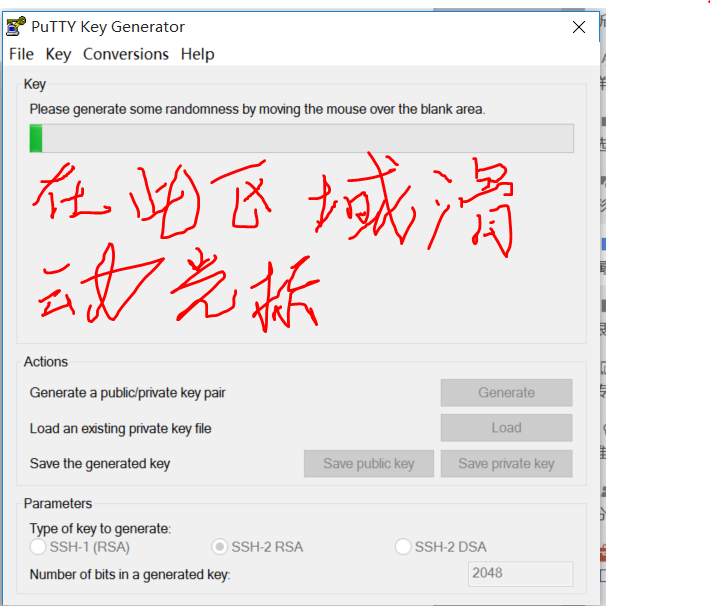
## SSH服务配置

1. 打开GitExtension软件，选择：工具->Putty->生成或者导入key

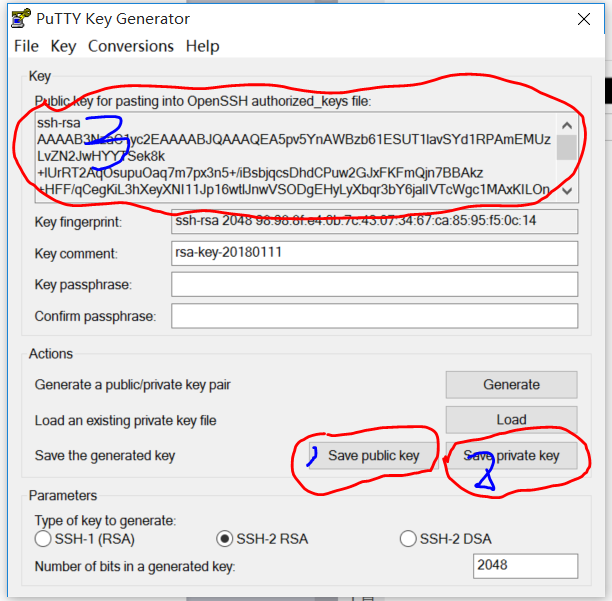


1. 点击Generate，鼠标在图示区域内**不断滑动**，进度条满了之后自动生成公钥和私钥

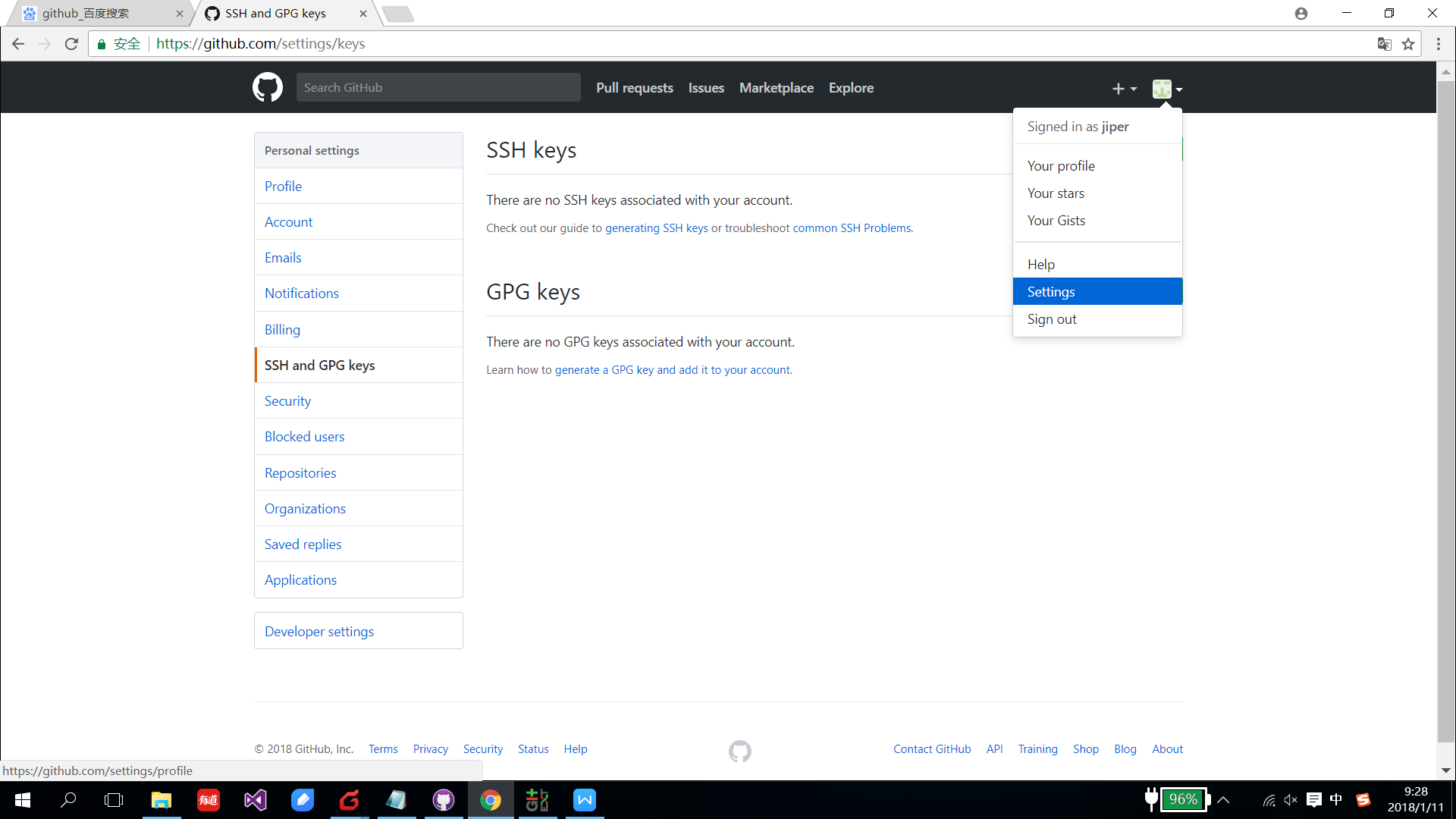




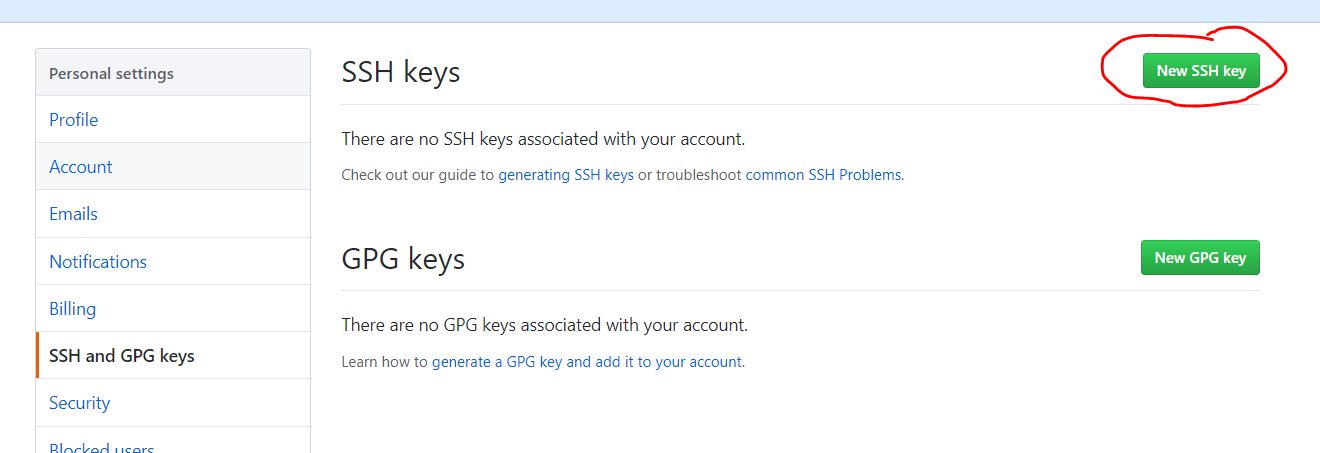
1. 生成公钥和私钥以后点击保存、复制公钥内容。

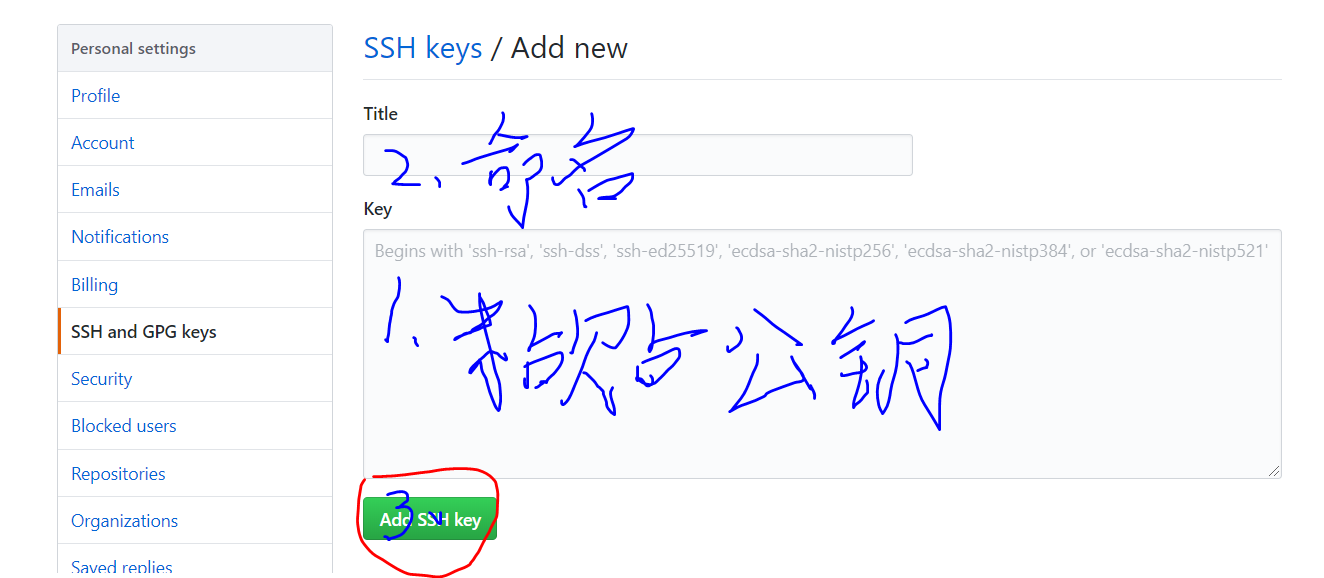


4.网页登录GitHub账号，点击Setting->SSH and GPG keys->new ssh key

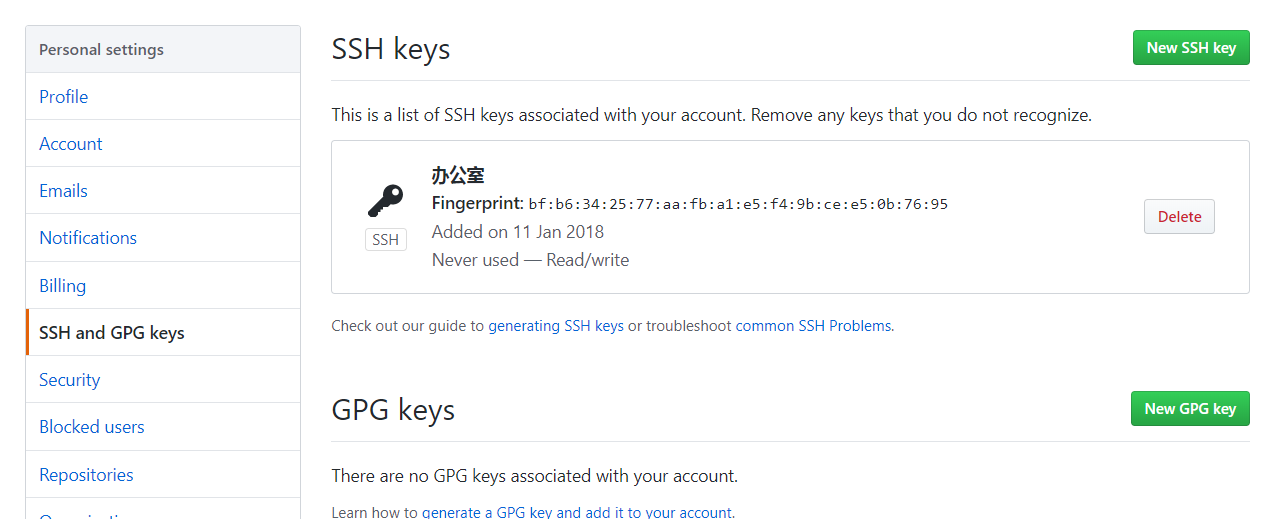


1. 填入上一步复制的公钥内容



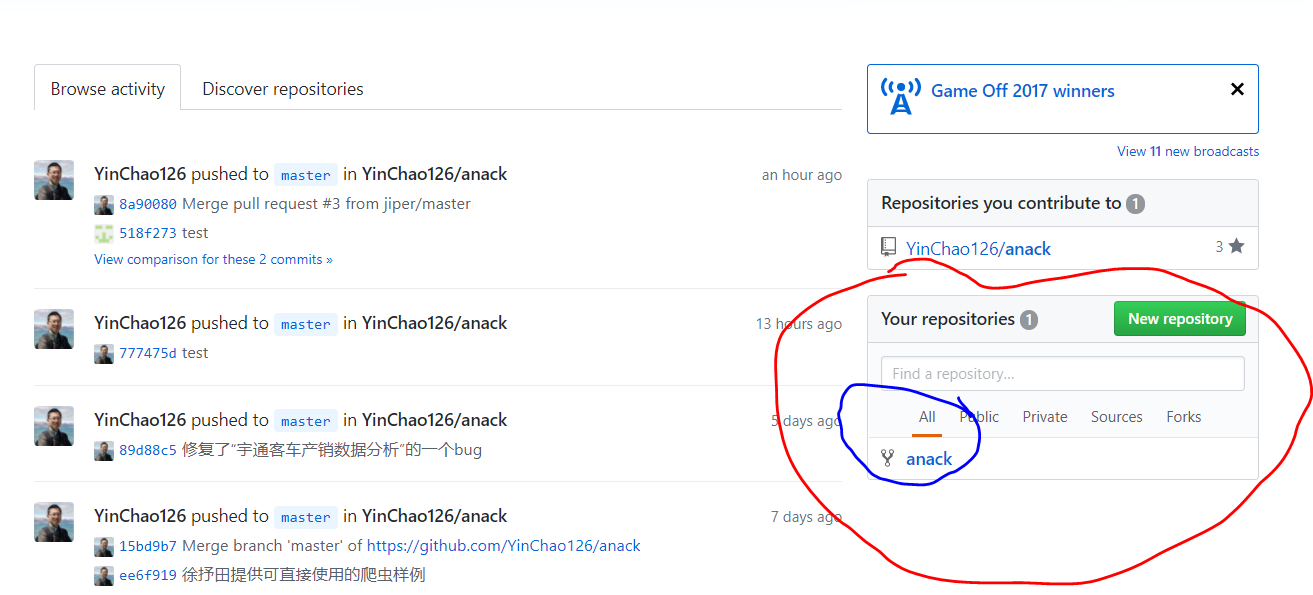


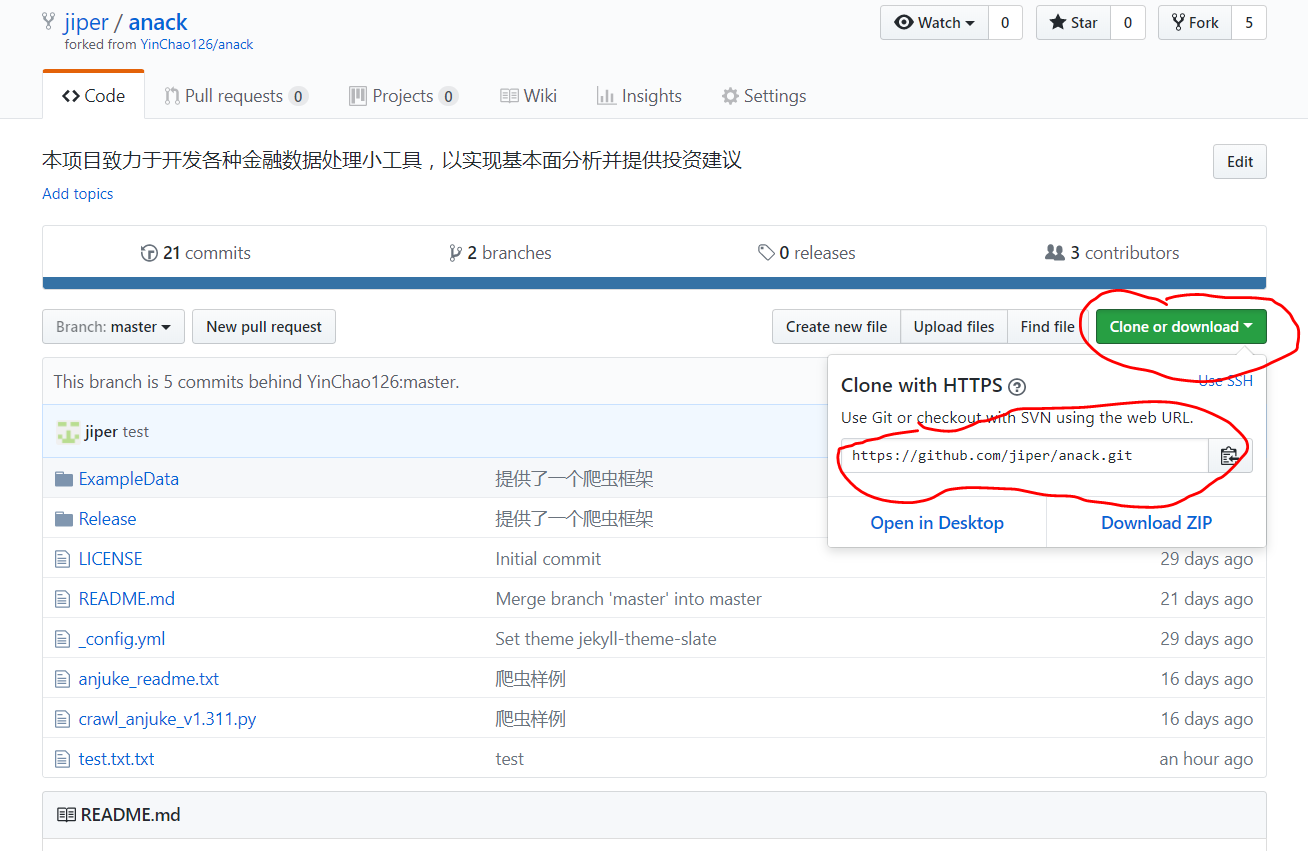
1. 完成了SSH服务的配置，可以实现本地和服务器端的共享访问了



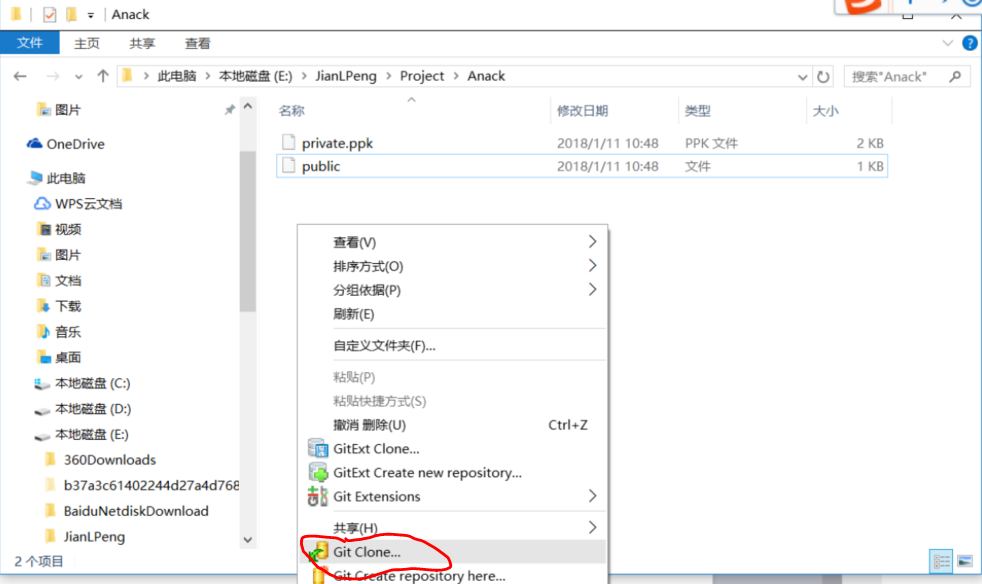
## 同步开发验证（看SSH服务认证是否成功）

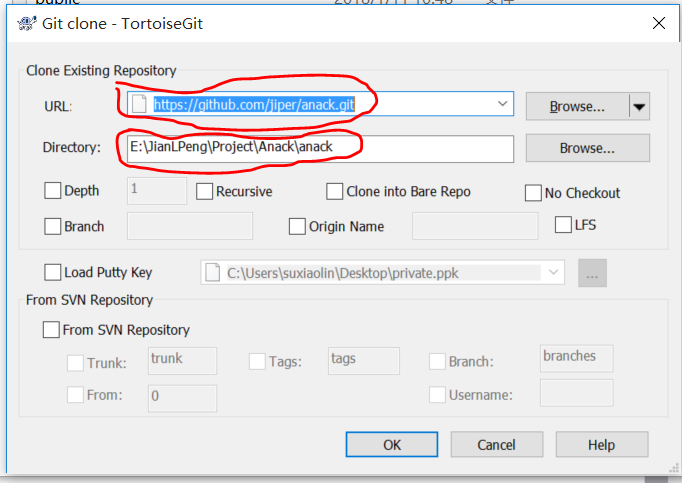
1. 打开网页上的anack工程（fork后属于自己的那个），点击clone or download获取该项目的url地址，复制url地址



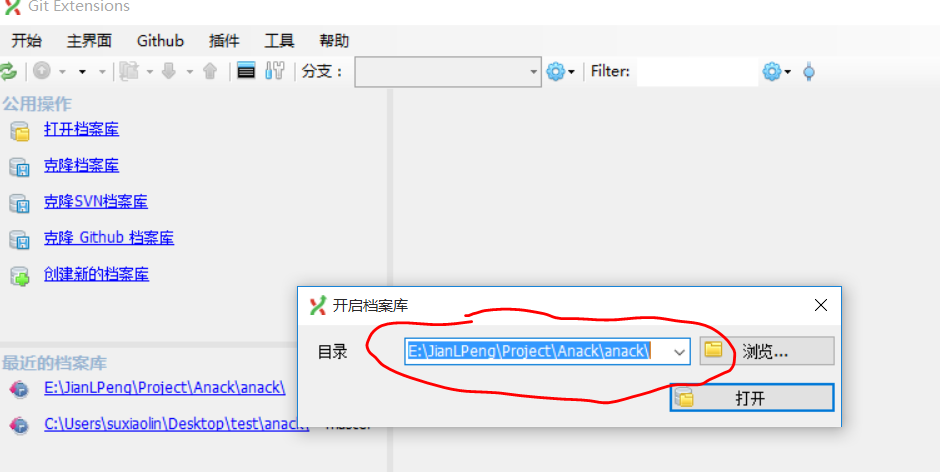


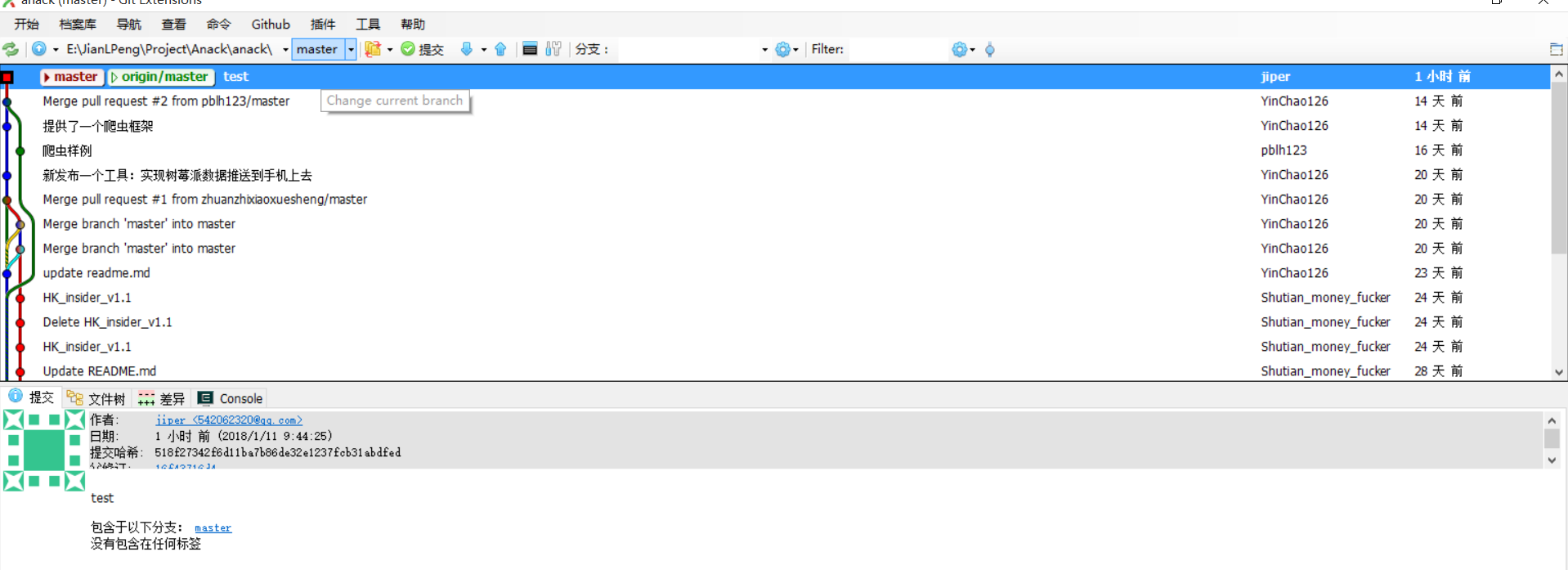
1. 直接在本地任意地点（最好选择在自己希望保存本工程的地方）右击，选择Git clone，填入url，勾选load putty key，浏览到之前保存好的私钥



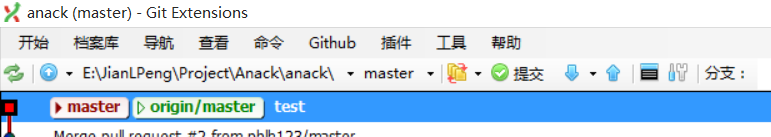


1. 点击OK，乌龟滚过去的时候就大功告成了。
2. 成功之后，打开GitExtension，快捷键Ctrl+O(或者进入 开始-打开 菜单)，之前保存项目的文件夹，打击打开，就可以看到这个工程的修改记录了





## 本地分支管理

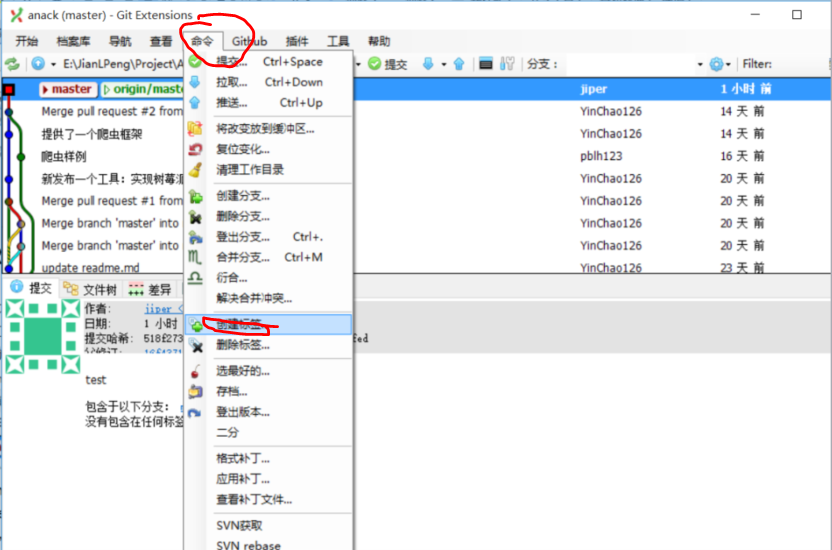
如图；

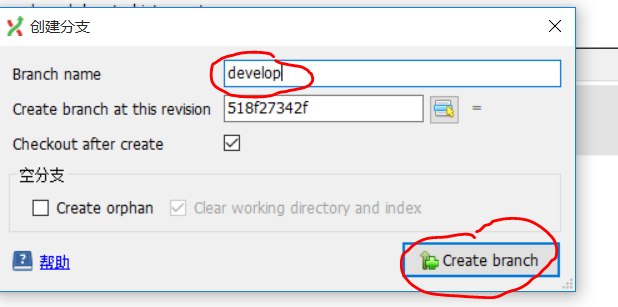
origin/master分支为远程服务器上的主分支（可以认为为网页版github上的分支）

master分支为同步远程分支时，自动创建的本地主分支

另外，一般我们需要在本地主分支之外，建立其他的分支，用于测试、调试，当测试通过后，可以将其他分支的内容合并在master分支，然后推送给远端的origin/master分支。

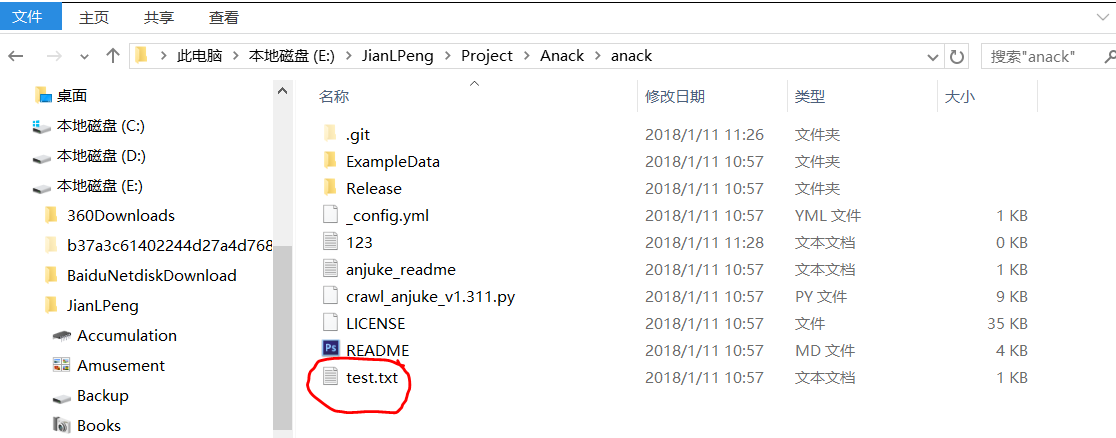
1. 创建分支（路径为 命令-创建分支）



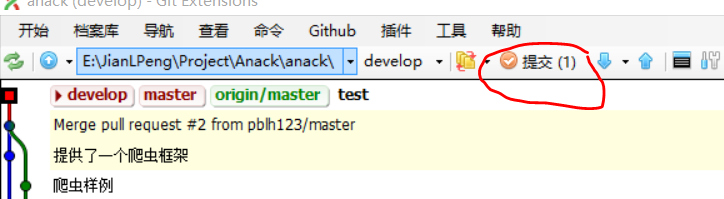




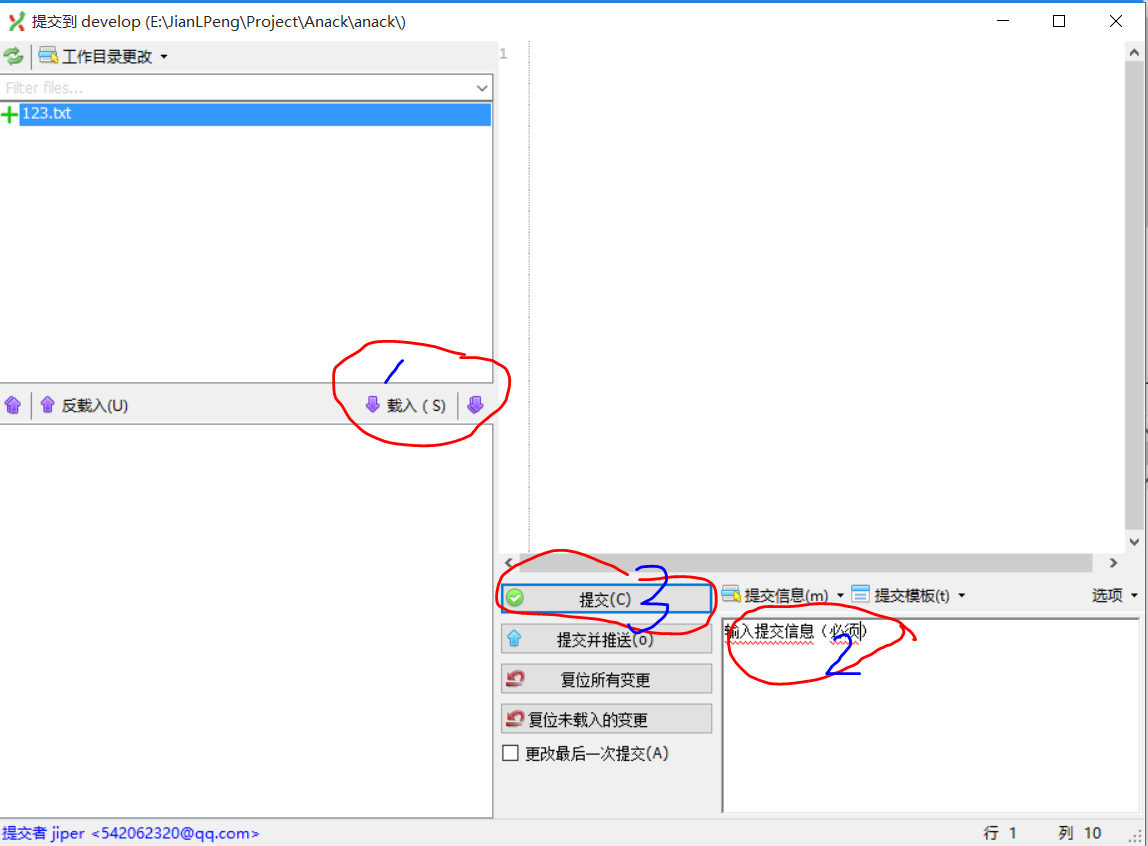
1. 修改工程，假如就是在anack文件夹下新增123.txt的文档



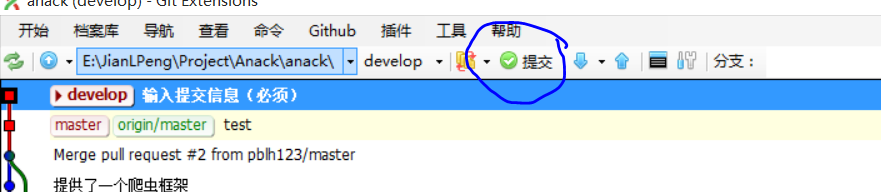
4、打开GitExtension，发现提交按钮变成了橙色（表示有未提交的内容），（1）代表新增、修改或删除的文件为1个，点击提交



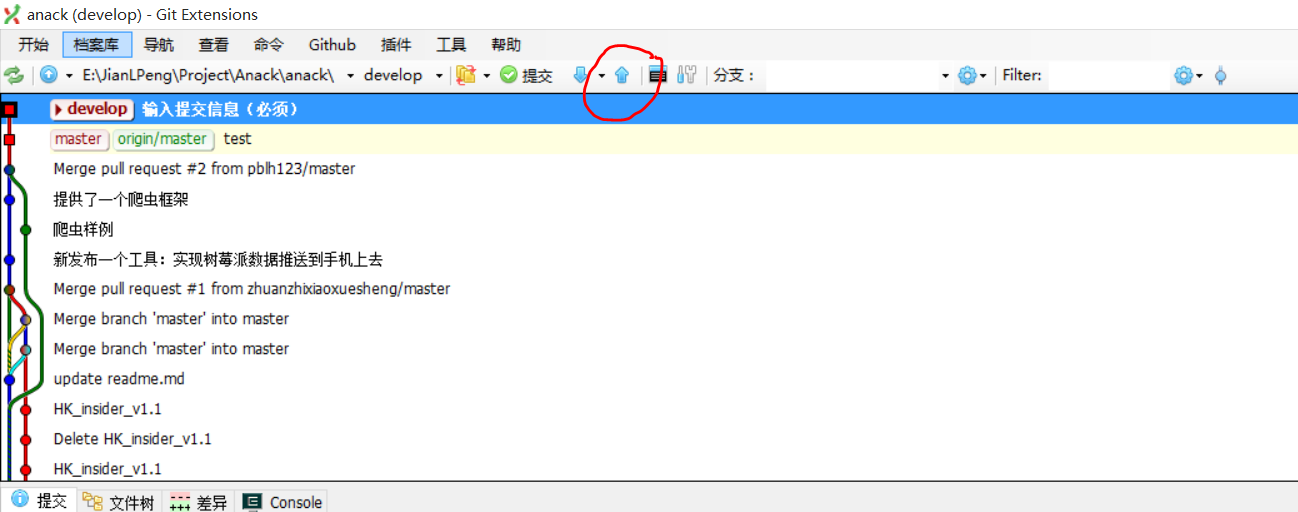
5、提交修改

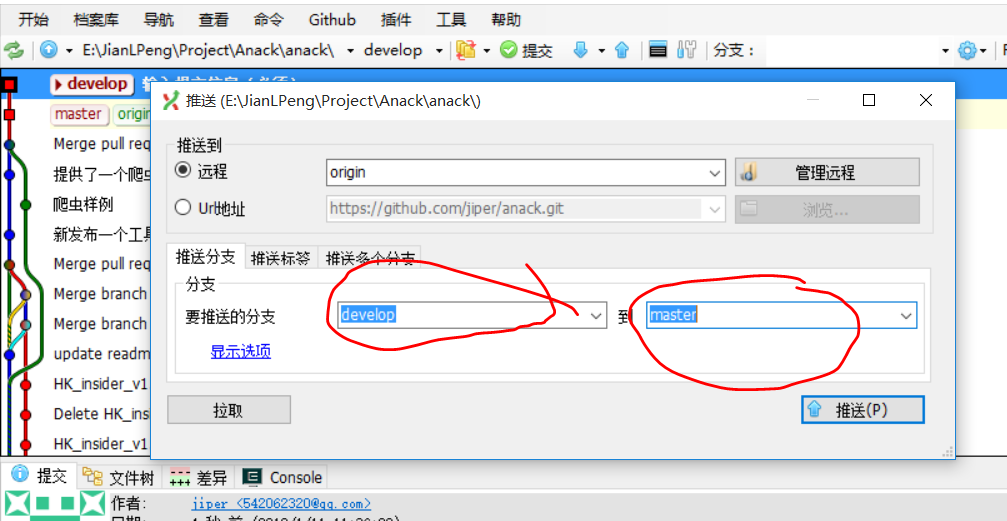


1. 提交成功后，提交按钮变回了绿色，表示所有信息都已经是最新的



1. 如果确保提交的内容无误，可以将版本推送给远端（也可以和本地master分支合并后推送）-----详见下一章节

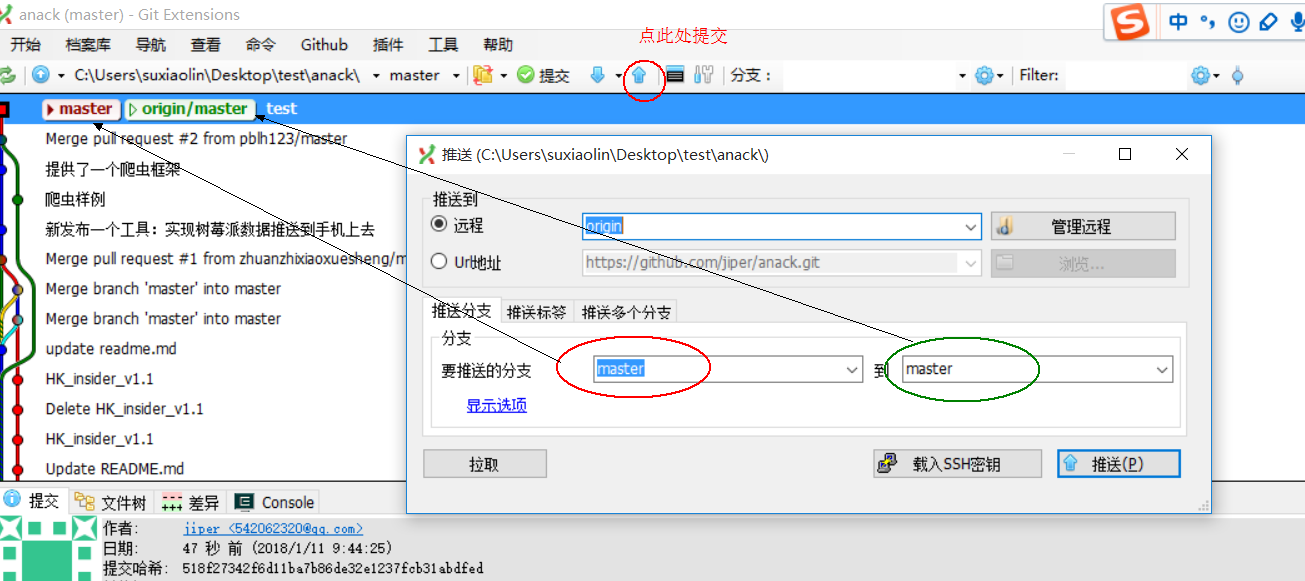




## Pull Request实现网络协同开发

同步之后，假如你在本地修改了这个工程，那么怎么将修改的内容推送给别人呢？假如我再本地master分支下新增了test.txt文档，我要将master分支推送给远程的original master分支。

1. 本地修改后利用GitExtension推送到自己fork的anack服务器端。保证本地和user/anack服务器端的工程同步



1. User/anack工程中提交pull request，实现最终的协同开发

