**项目协作流程**

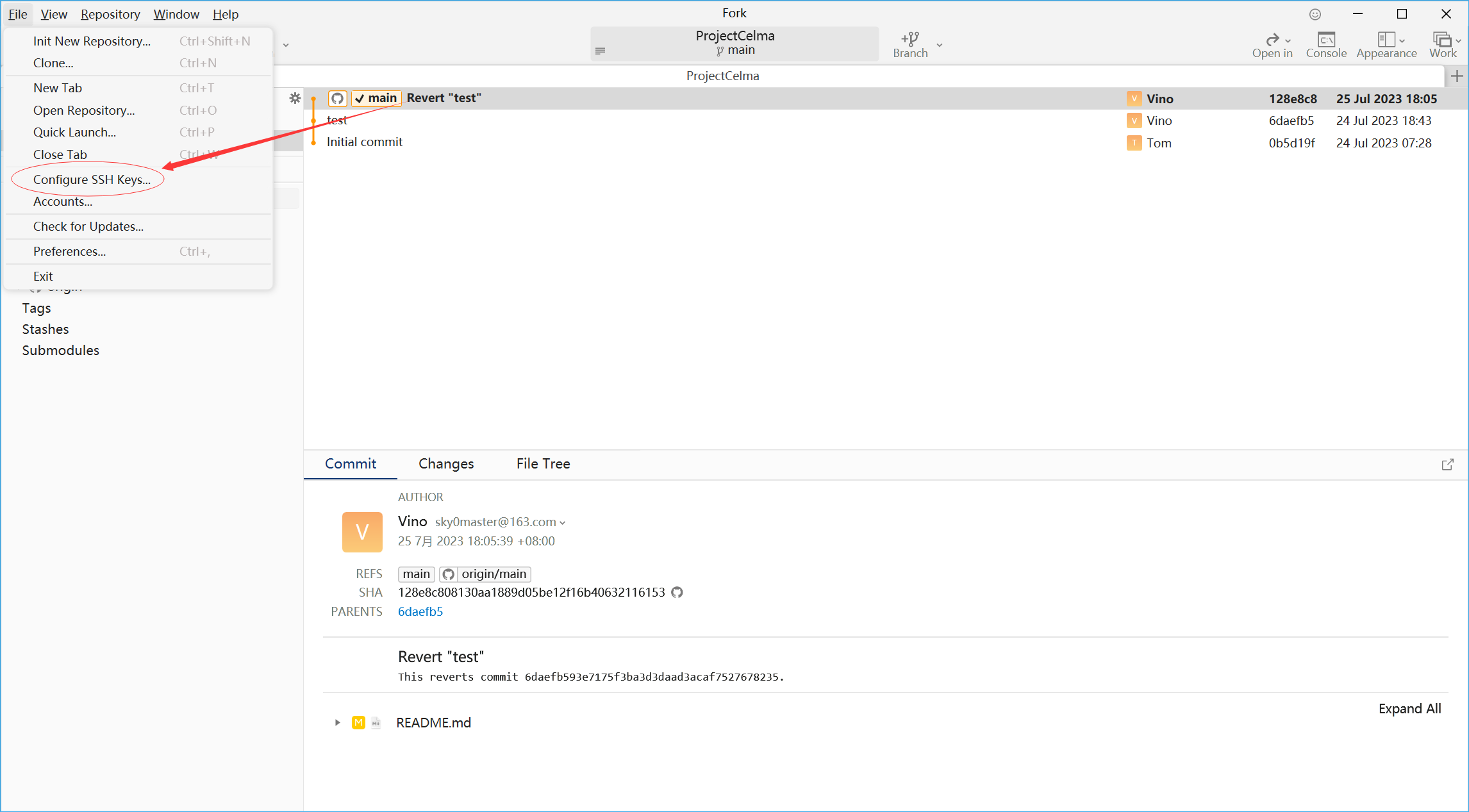
**作者：Vino**

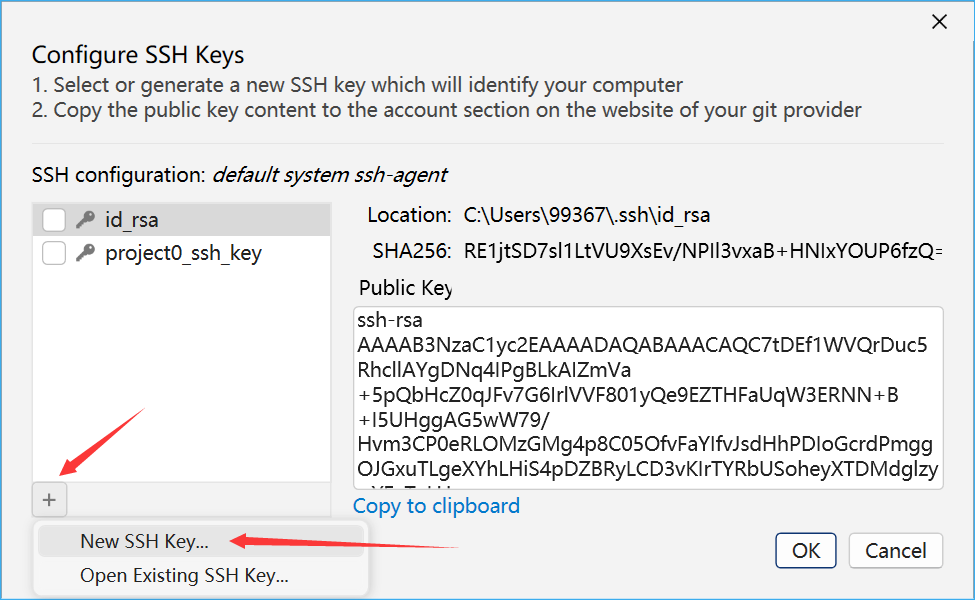
1. **环境配置**

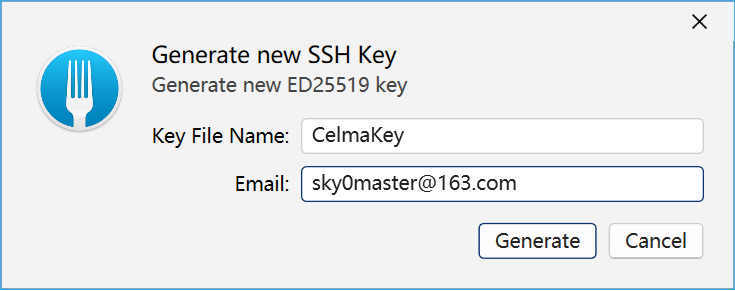
准备一个github账号，接受管理员的邀请加入组织，邀请以邮件形式发送。加入组织后即可访问项目仓库

下载并安装fork：[Fork - a fast and friendly git client for Mac and Windows (git-fork.com)](https://git-fork.com/),

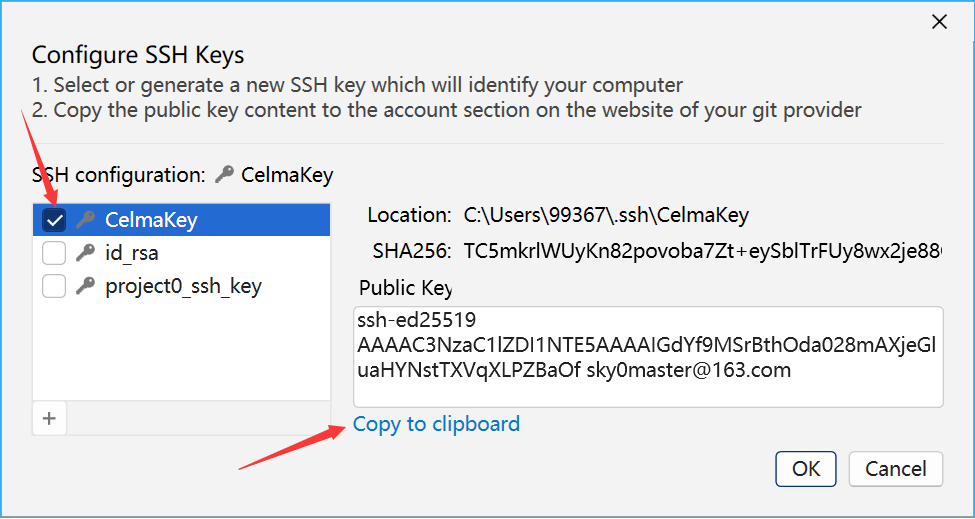
安装完成后运行，点击File->Configure SSH Keys..



点击此处加号，然后点New SSH Key...

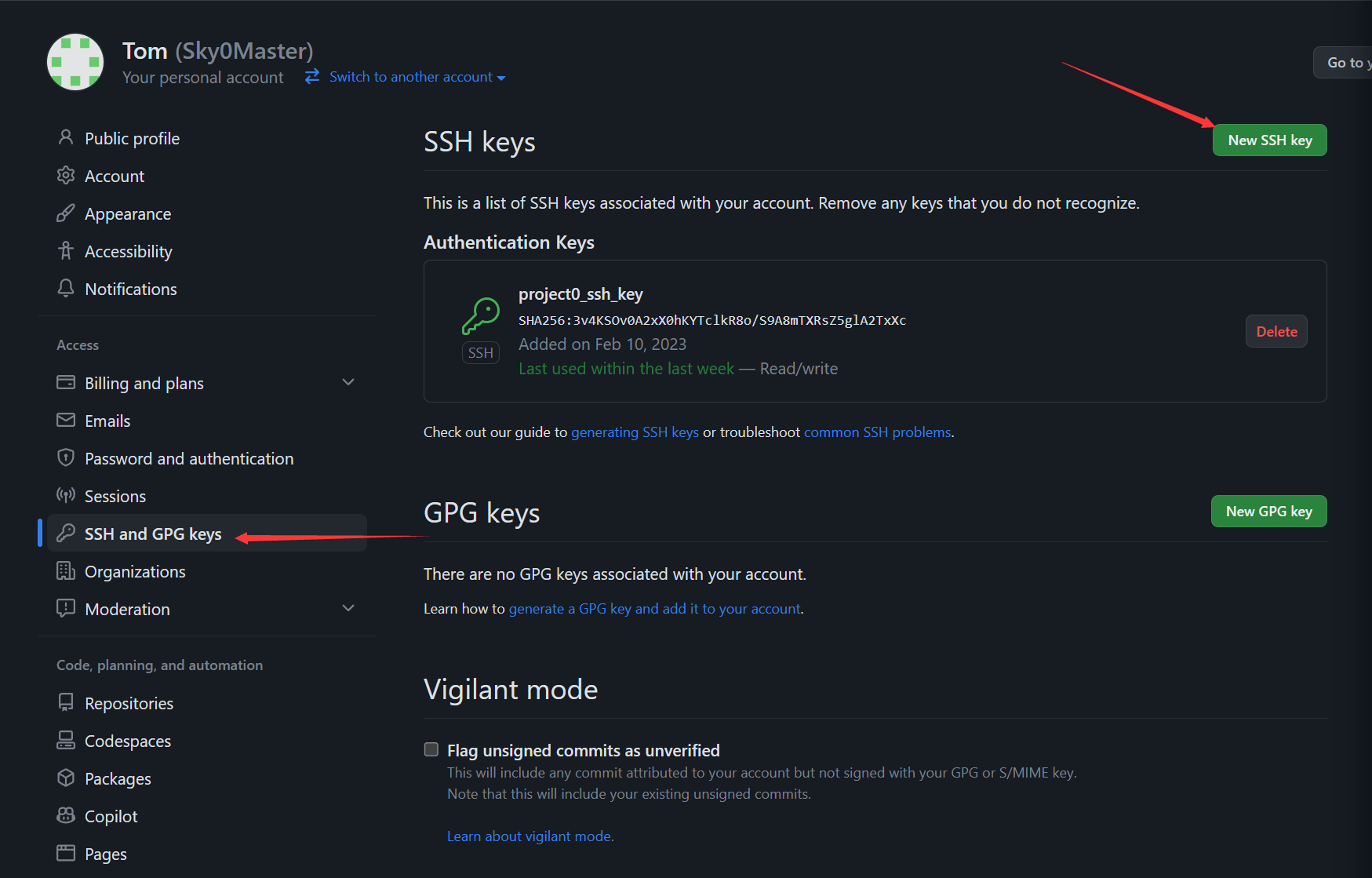
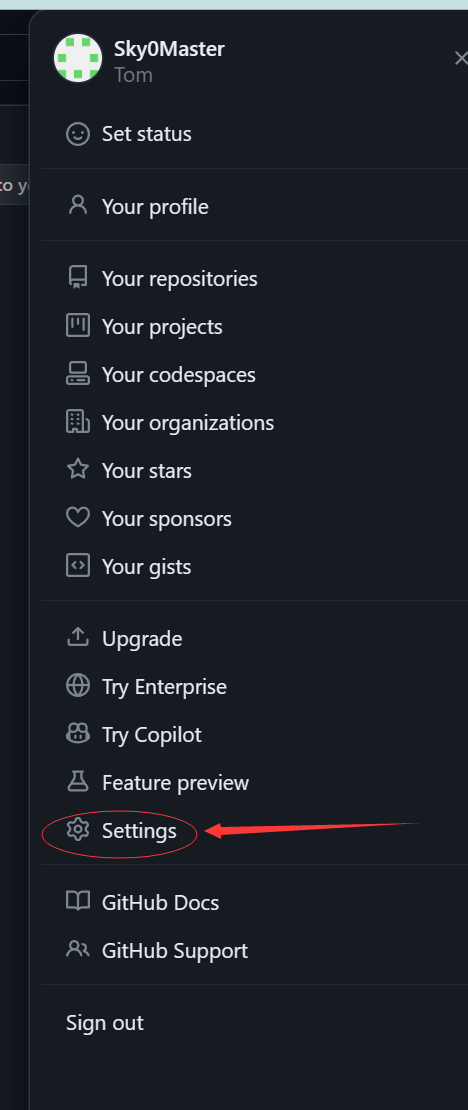


文件名随便填，邮箱填自己GitHub使用的邮箱

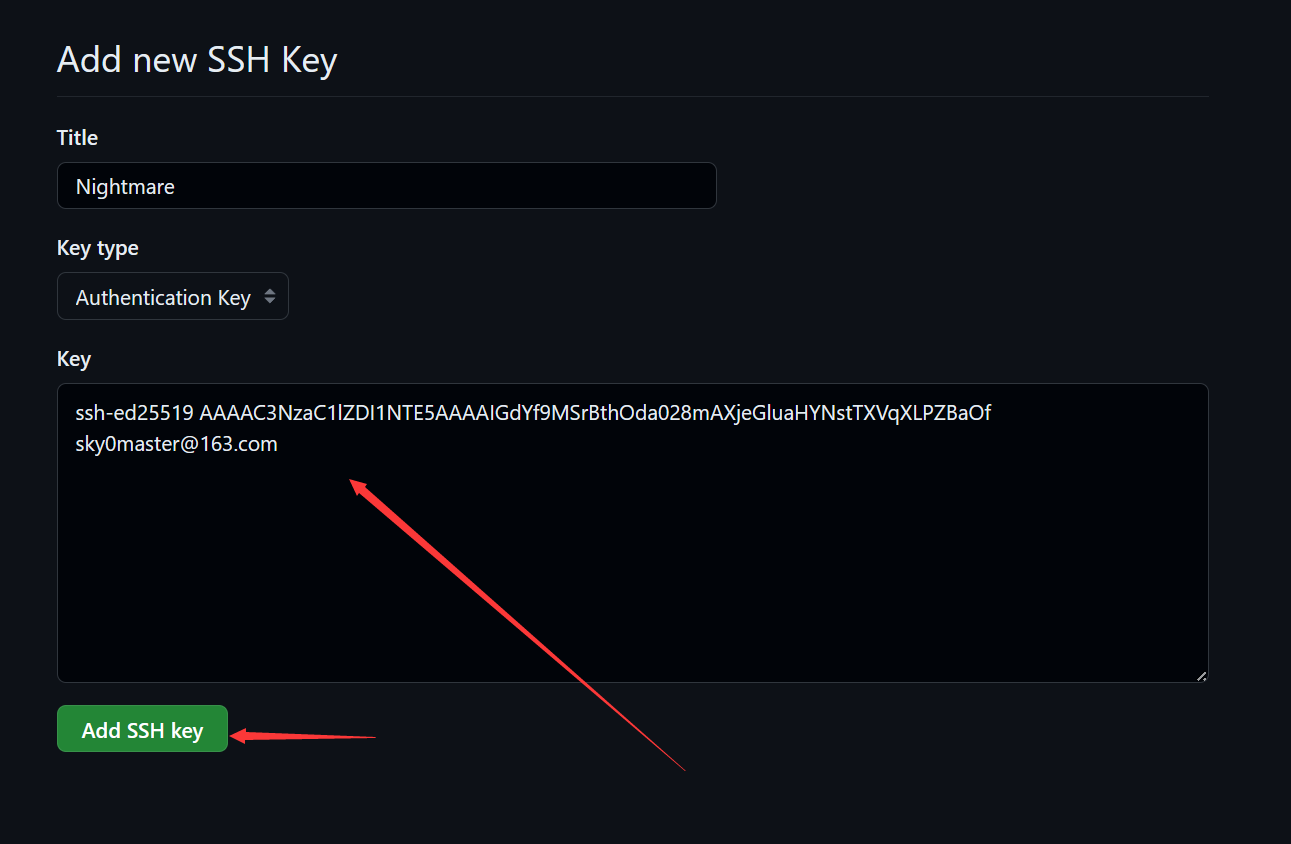


点此处勾选上，并且复制该Public Key

回到Github，打开个人settings



如图依次点击SSH and GPG keys, New SSH Key

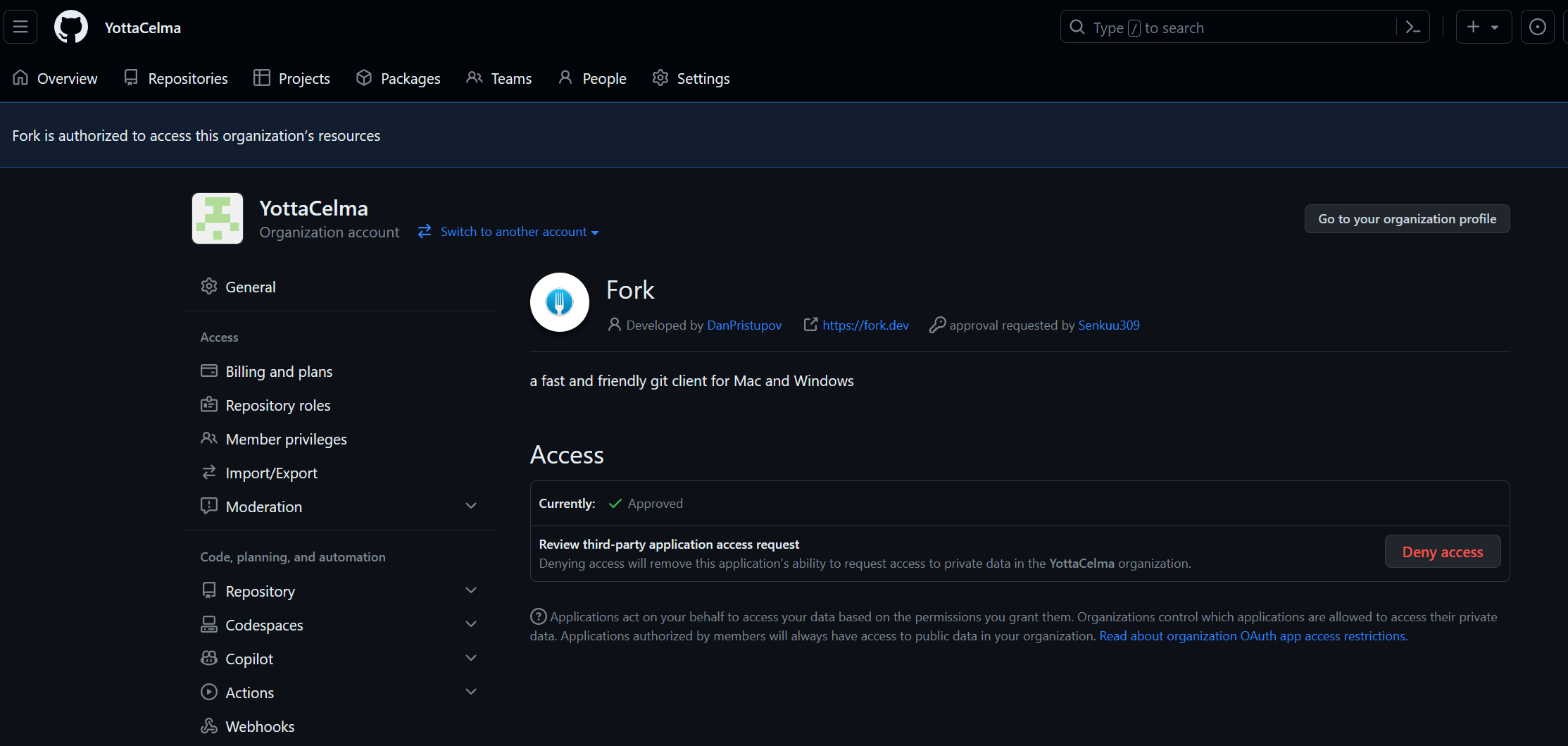
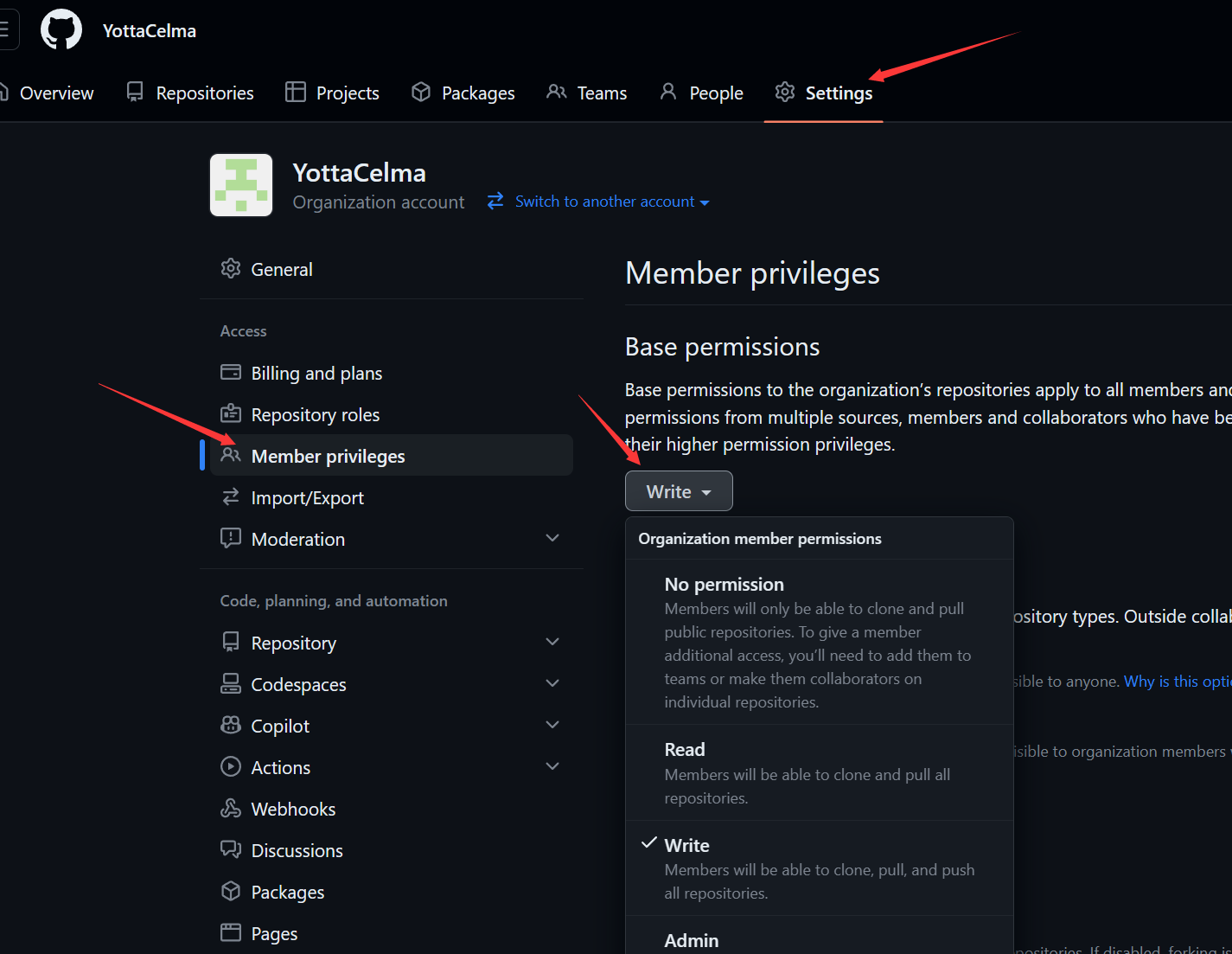


将之前复制的SSH Key复制在此处，Title随意，点击Add

该ssh key的作用是将你本地的fork使用ssh登录你的GitHub账户，如此本地的fork便有权限访问你的GitHub账户有权限访问的一切仓库。

---------------------------------------------------

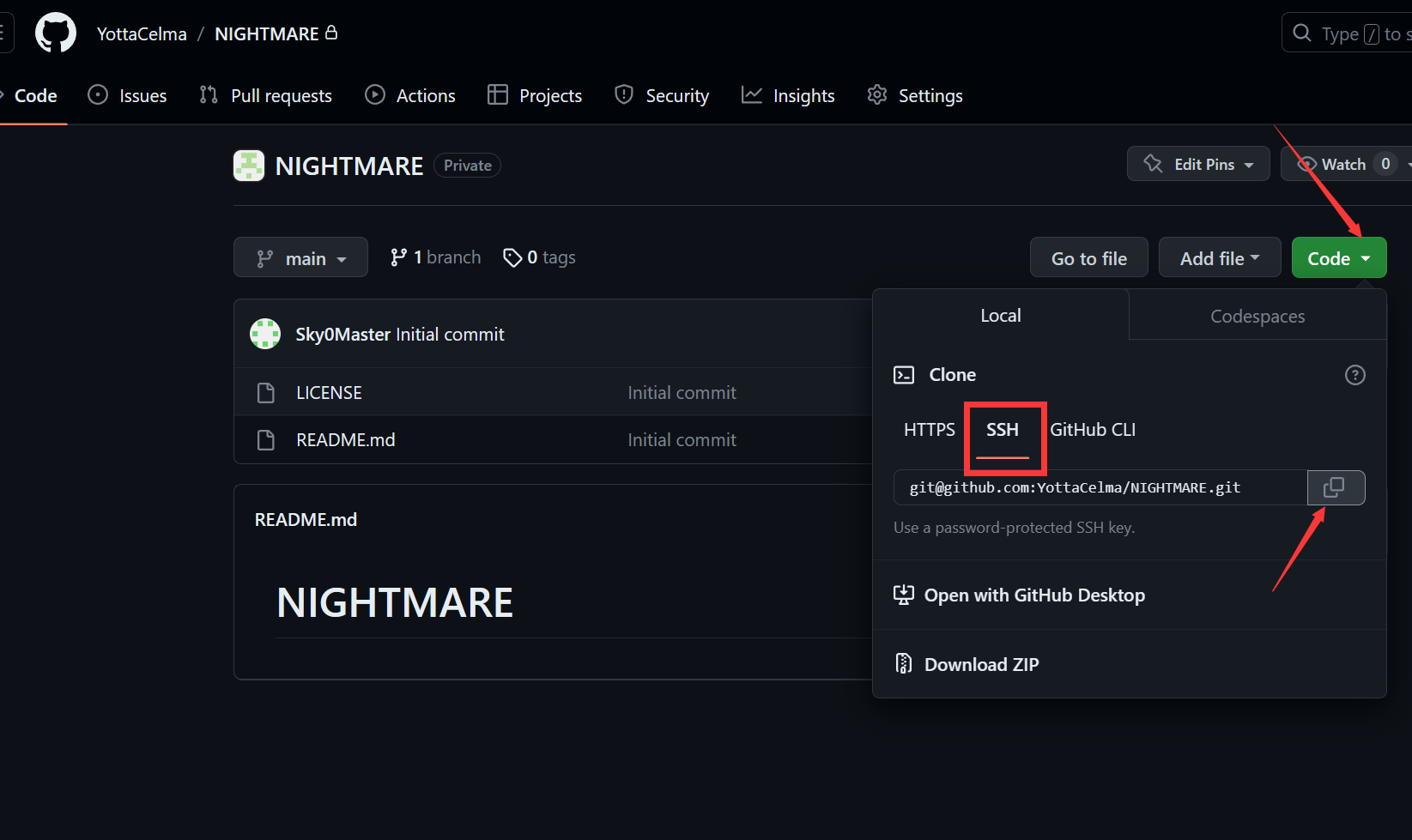
(#仅组织管理员：需要在组织的settings中配置组织成员的Write权限否则成员只能pull无法push，并且还需要授予第三方软件的权限）



---------------------------------------------------

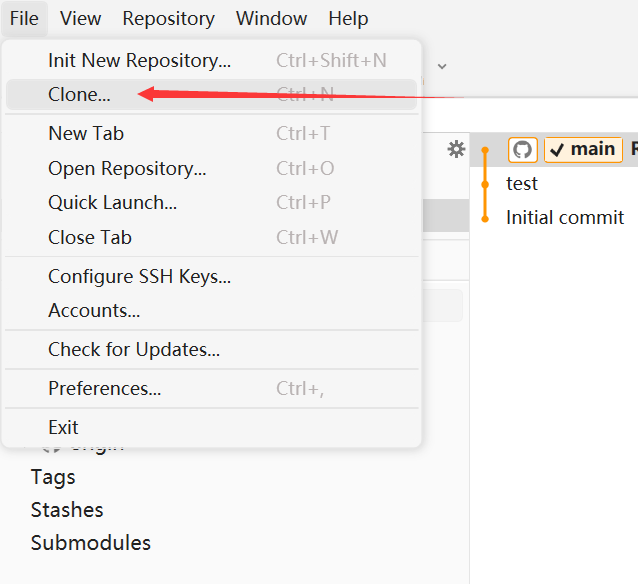
接着

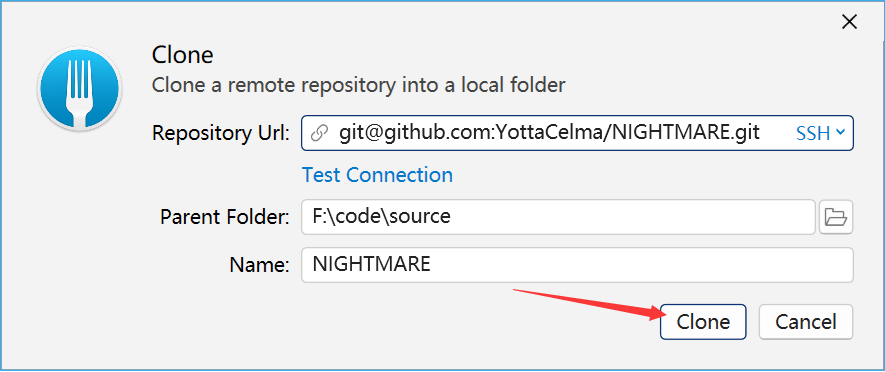
访问项目仓库，



如图依次点击，复制该链接

回到fork，点击File>Clone





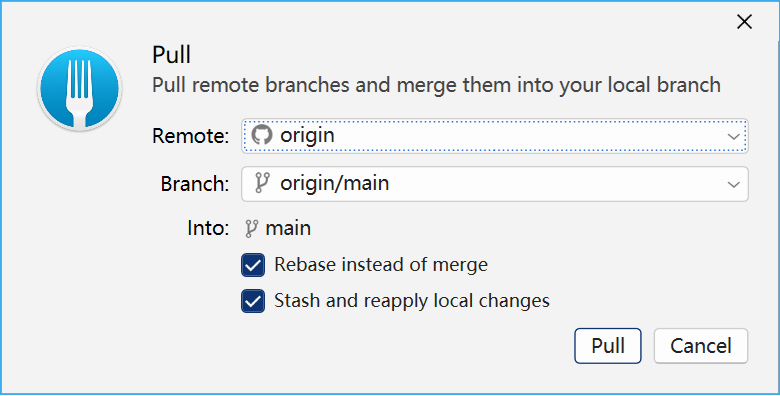
粘贴URL，一般会自动粘贴，点clone，配置就全部完成啦！

1. **操作教程**

Git实质上是维护了两个仓库：远端仓库和本地仓库。

考虑几种情况：

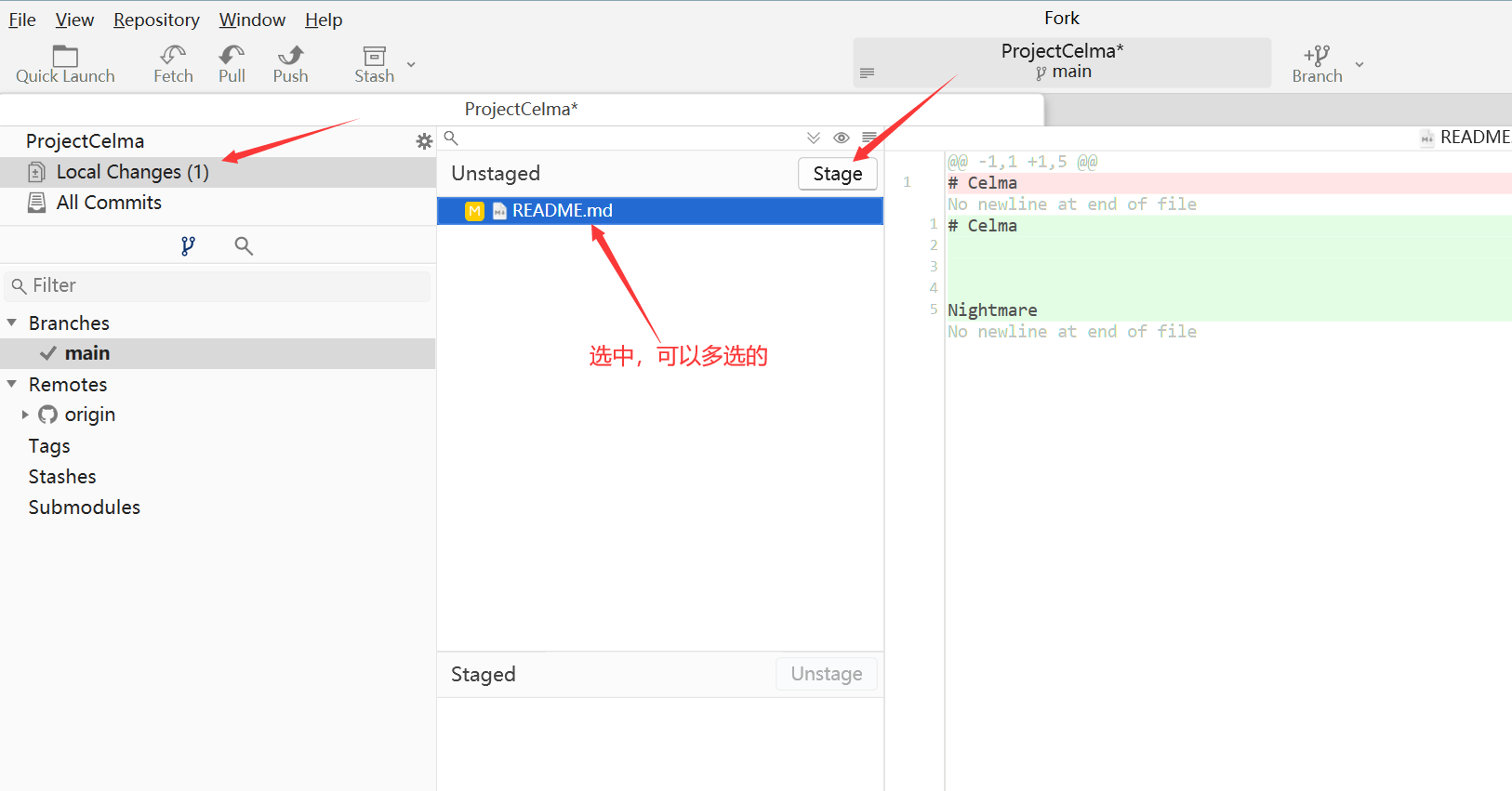
1. 别人有新的改动提交，你需要同步到本地仓库



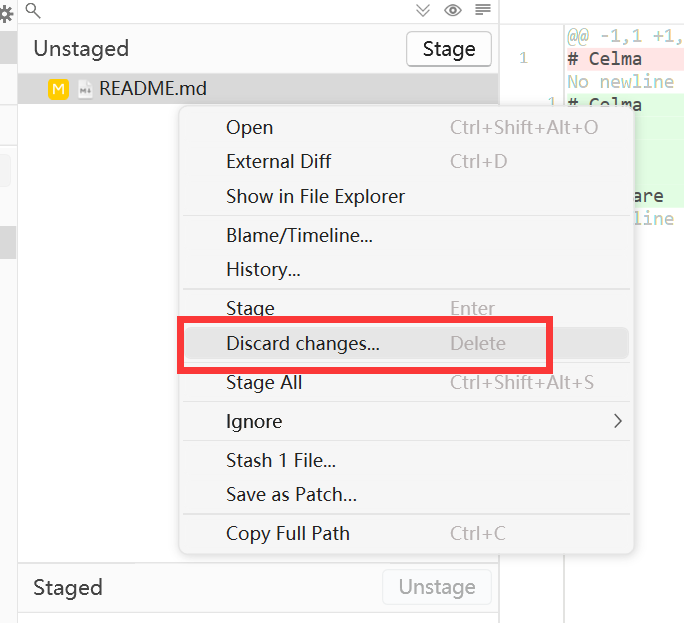
点击pull，勾选两项（暂时不用管他们什么意思），将会保存你本地未上传的改动（local changes）然后将远端仓库上的新的提交更新为本地仓库的提交。当存在文件冲突时将会提醒你进行处理，如果是文本类型的文件能直接合并别人的改动和你本地的改动，如果是二进制文件如蓝图，则需要选择保留本地的改动还是远端的改动，大多数情况都是保留远端的改动，然后再在本地重新加一遍之前被抛弃的本地改动。因此开发者们应该避免同时修改同一个二进制文件。

1. 你有新的改动需要提交到远端仓库

Local Change会显示本地改动，将确认要提交的改动stage，

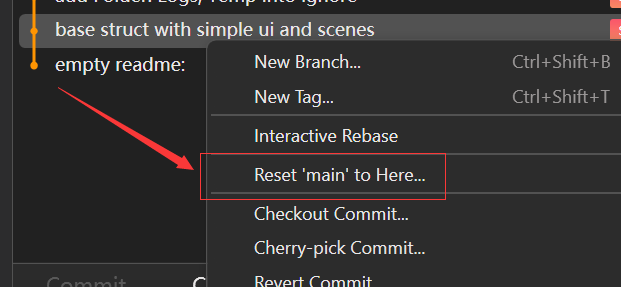


如果有错误的改动需要还原的上一次的状态，可以按delete或者右键discard

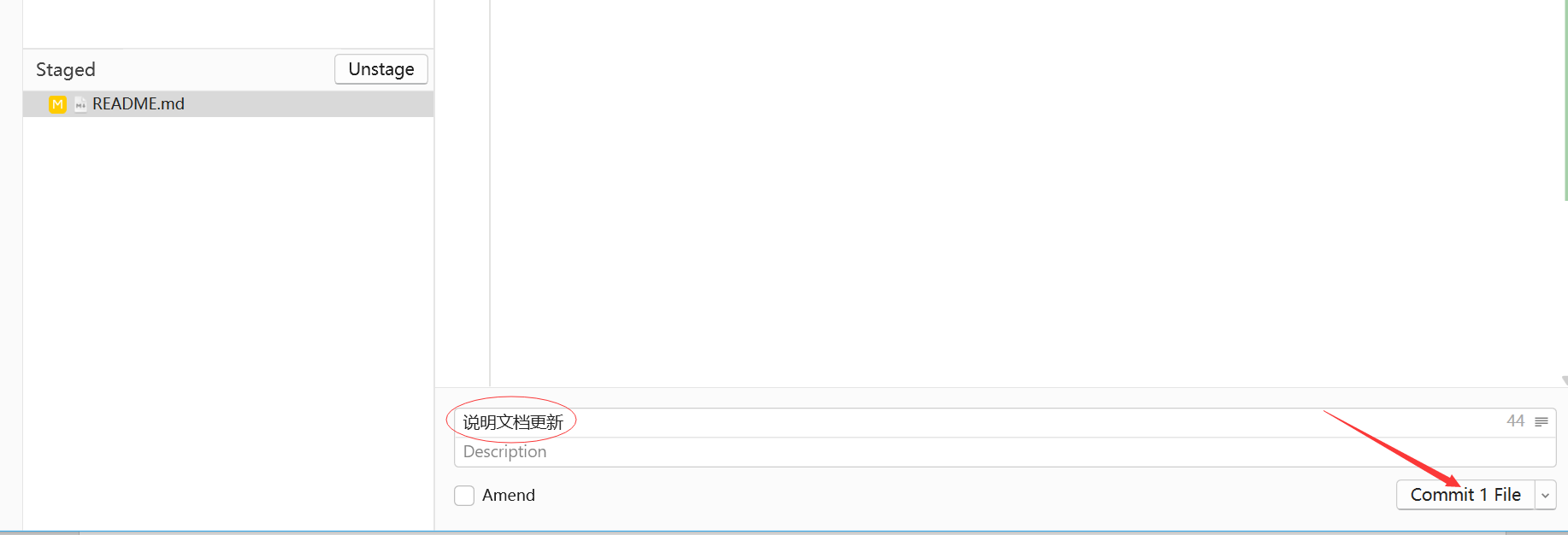


，该文件将会退回上一次提交的状态

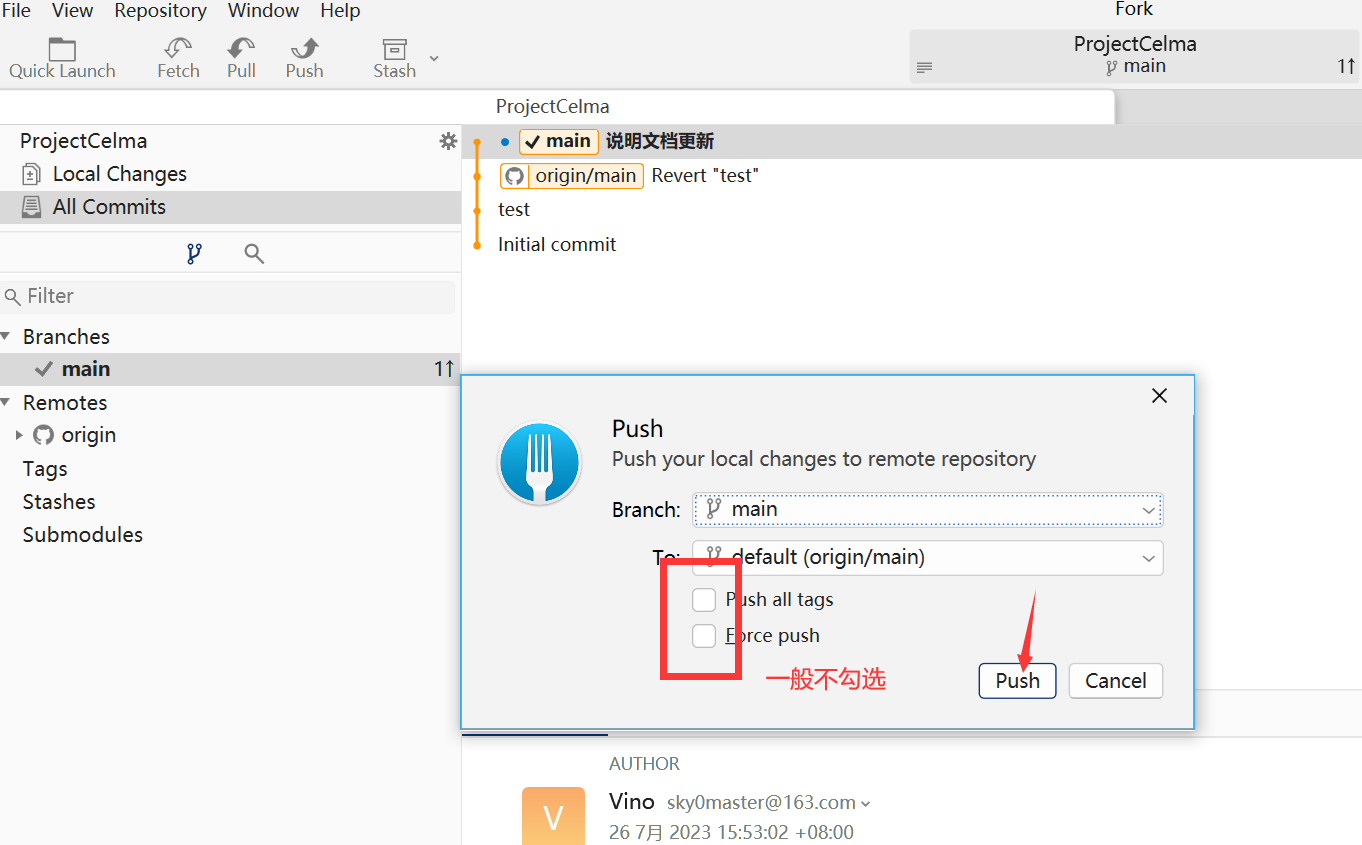
如果需要退回到某个已提交的节点（版本）之前，可以右键点击需要退回到的节点，点击Reset ‘main’ to Here



Stage了的文件在右侧填写提交内容的信息，最好简要描述一下本次提交包含什么内容。



Commit之后就会提交到本地的分支，如果此时远端有新的提交，需要先pull将本地的分支与远端分支同步再点push，如果没有要下载的改动则可以直接push



Push成功后，远端仓库将会和本地仓库同步。远端仓库上会出现和刚刚在本地仓库上一样的提交记录，此时其他人就可以通过pull来更新你刚刚的提交记录到他们的本地仓库。