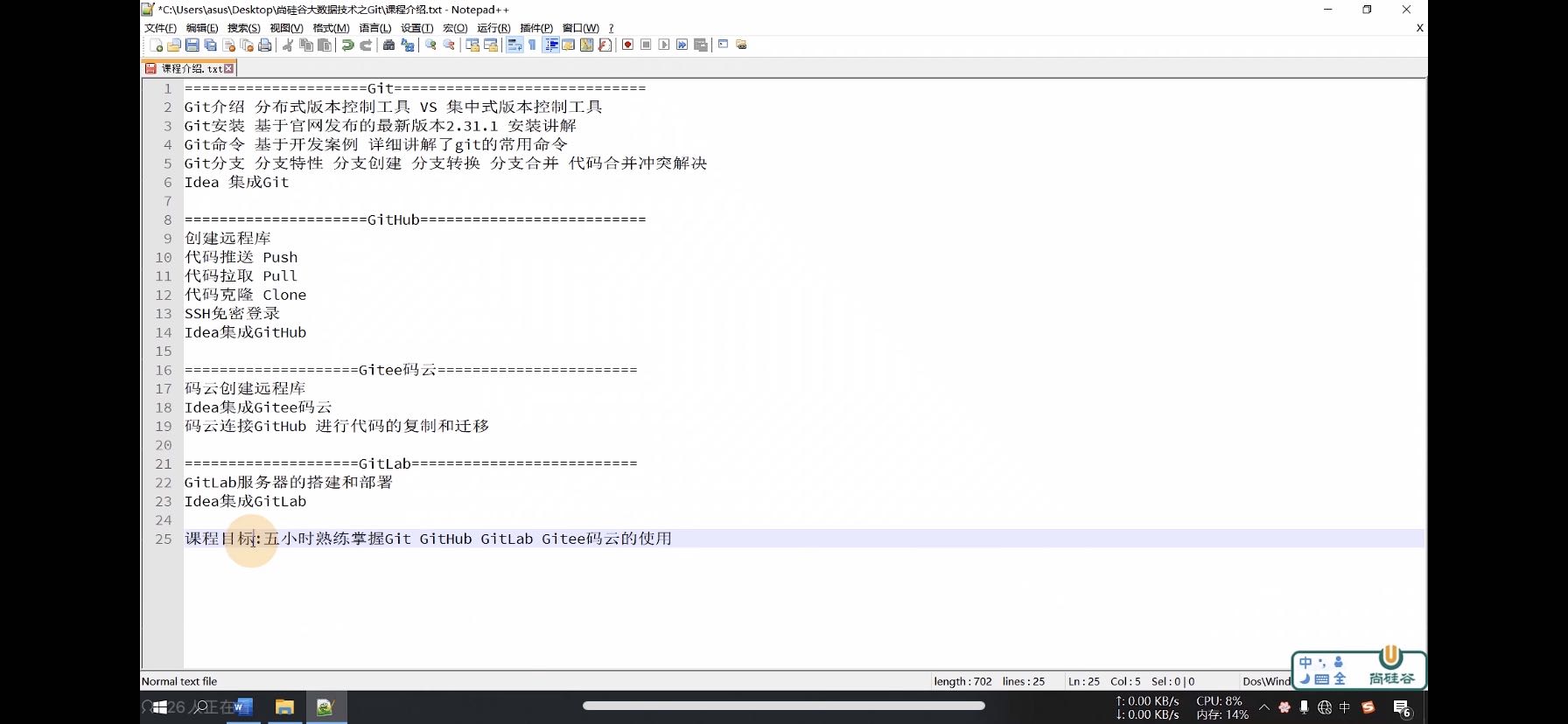
**Git Learning Notes**

1. Git 常用命令



1. Git 简介
2. 课程纲要

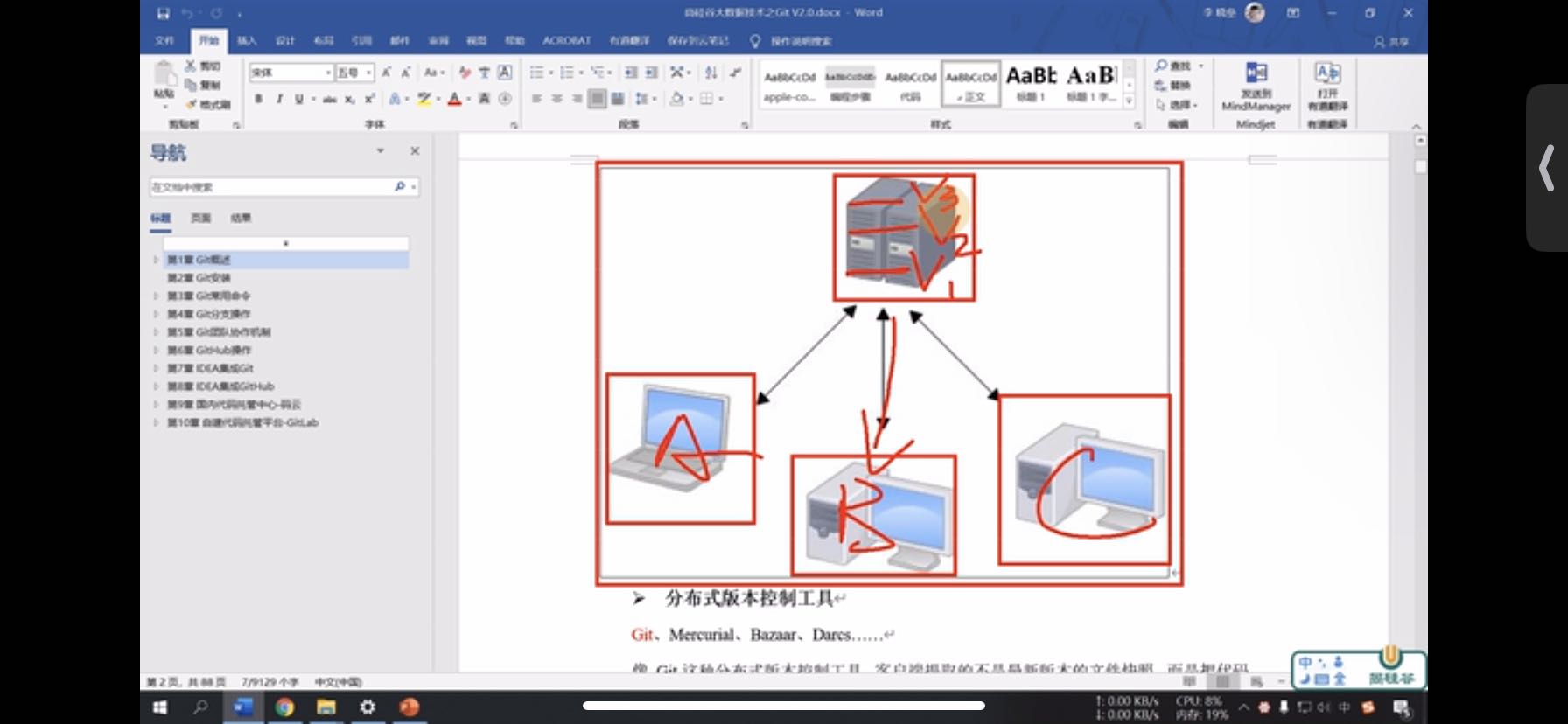


1. 版本控制工具

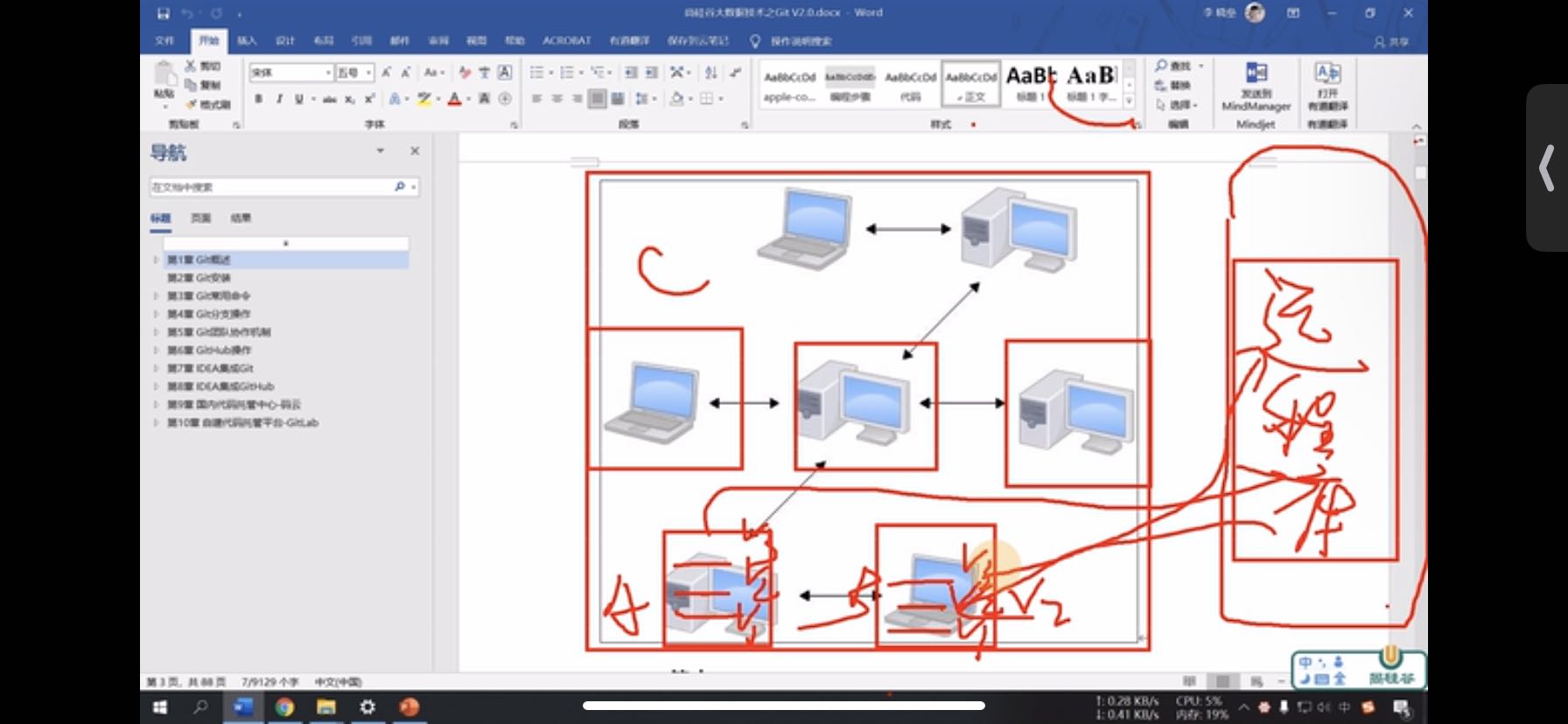
版本控制是一种记录文件内容变化，以便将来查阅特定版本修订情况的系统。

版本控制其实最重要的是可以**记录文件修改历史记录**，从而让用户能够查看历史版本，方便版本切换。

集中式版本控制工具,如SVN（Subversion）

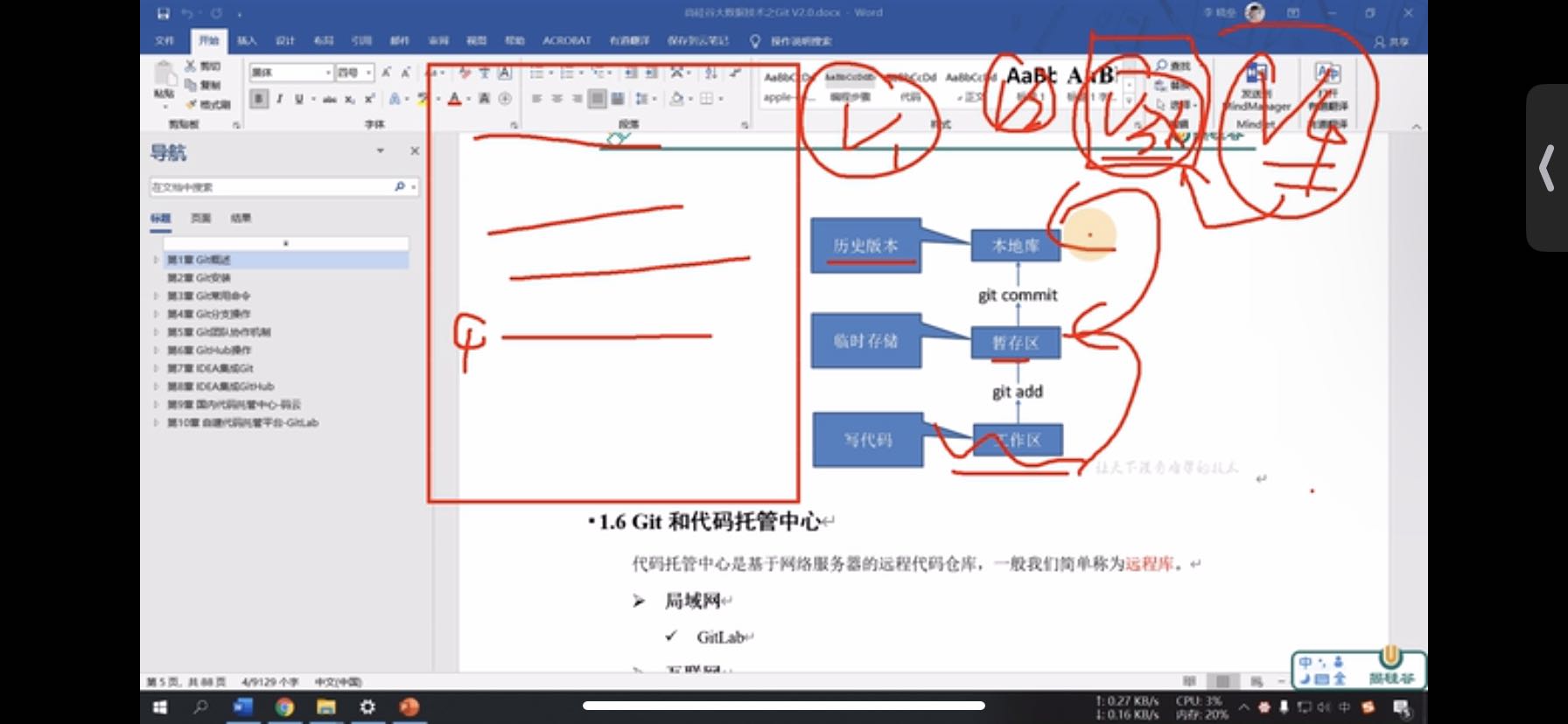


Git 是一种**分布式版本控制系统。**



1. Git的本地工作机制和代码托管中心

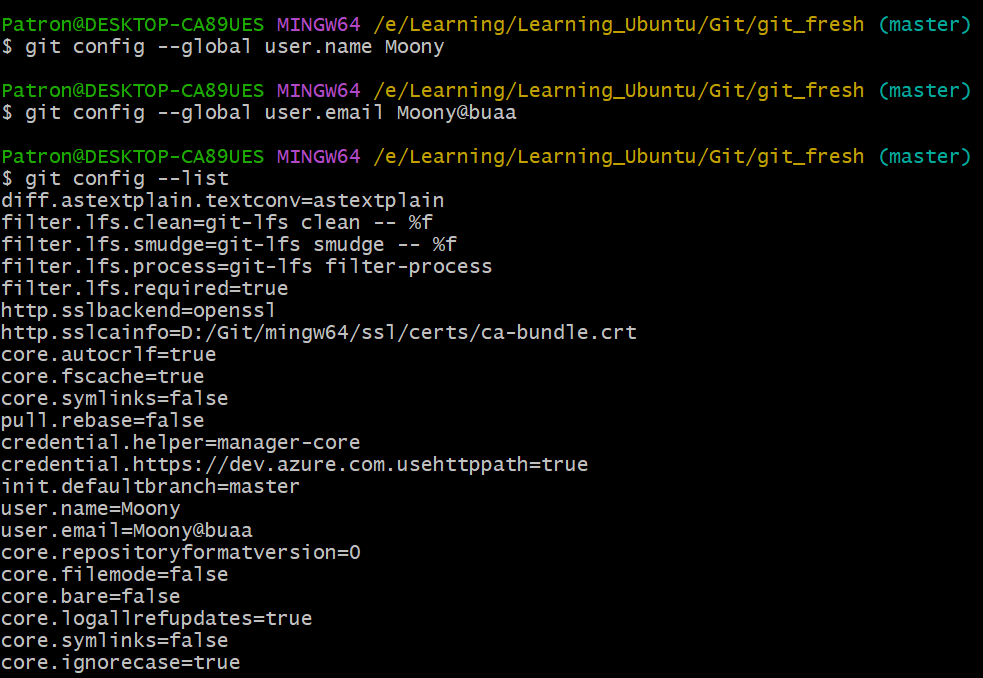
代码托管中心是基于网络服务器的远程代码仓库，一般我们简单称为远程库。



1. Git 常用命令详解



1. Git设置用户签名



**Git 首次安装必须设置一下用户签名**，否则无法提交代码。

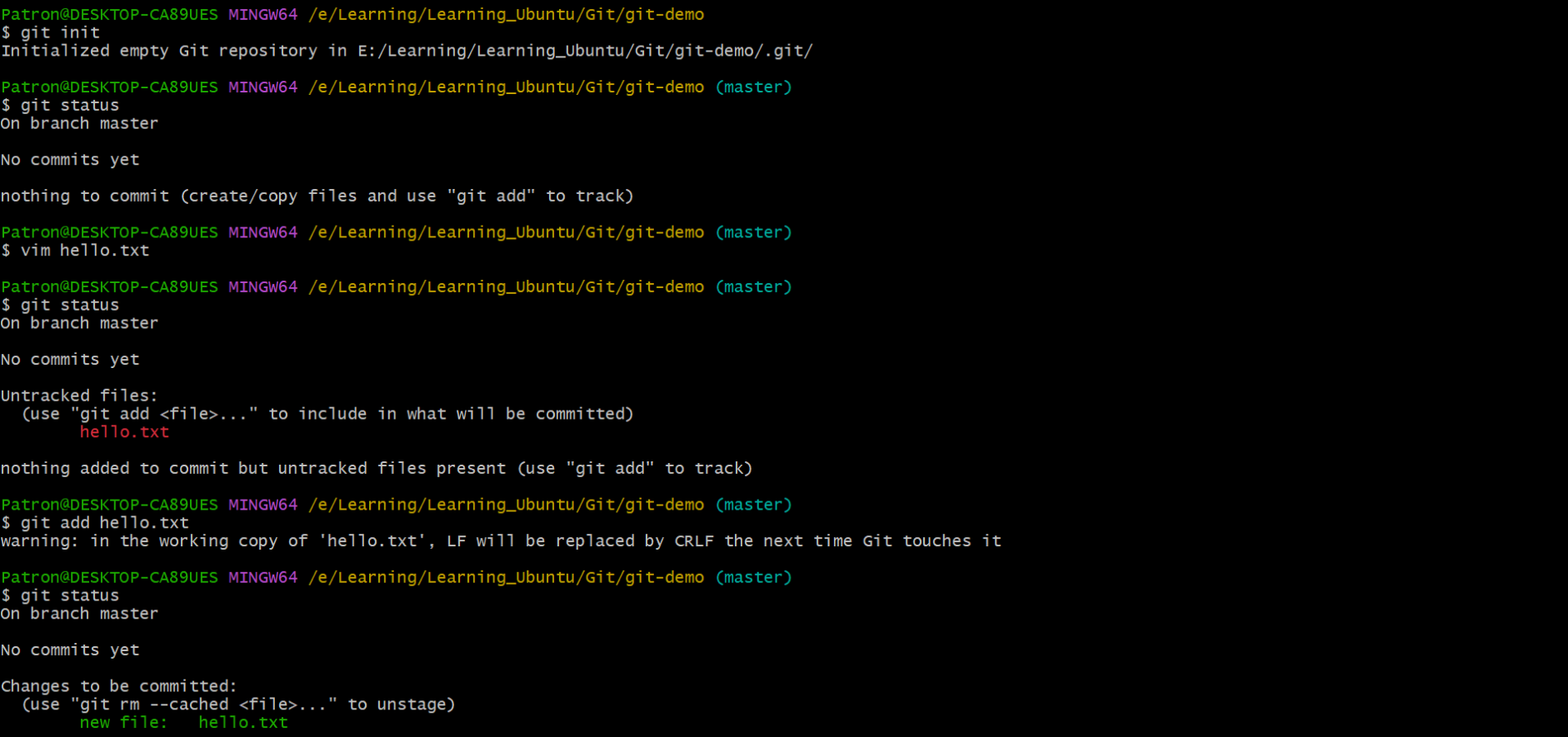
**※注意：**这里设置用户签名和将来登录 GitHub（或其他代码托管中心）的账号没有任

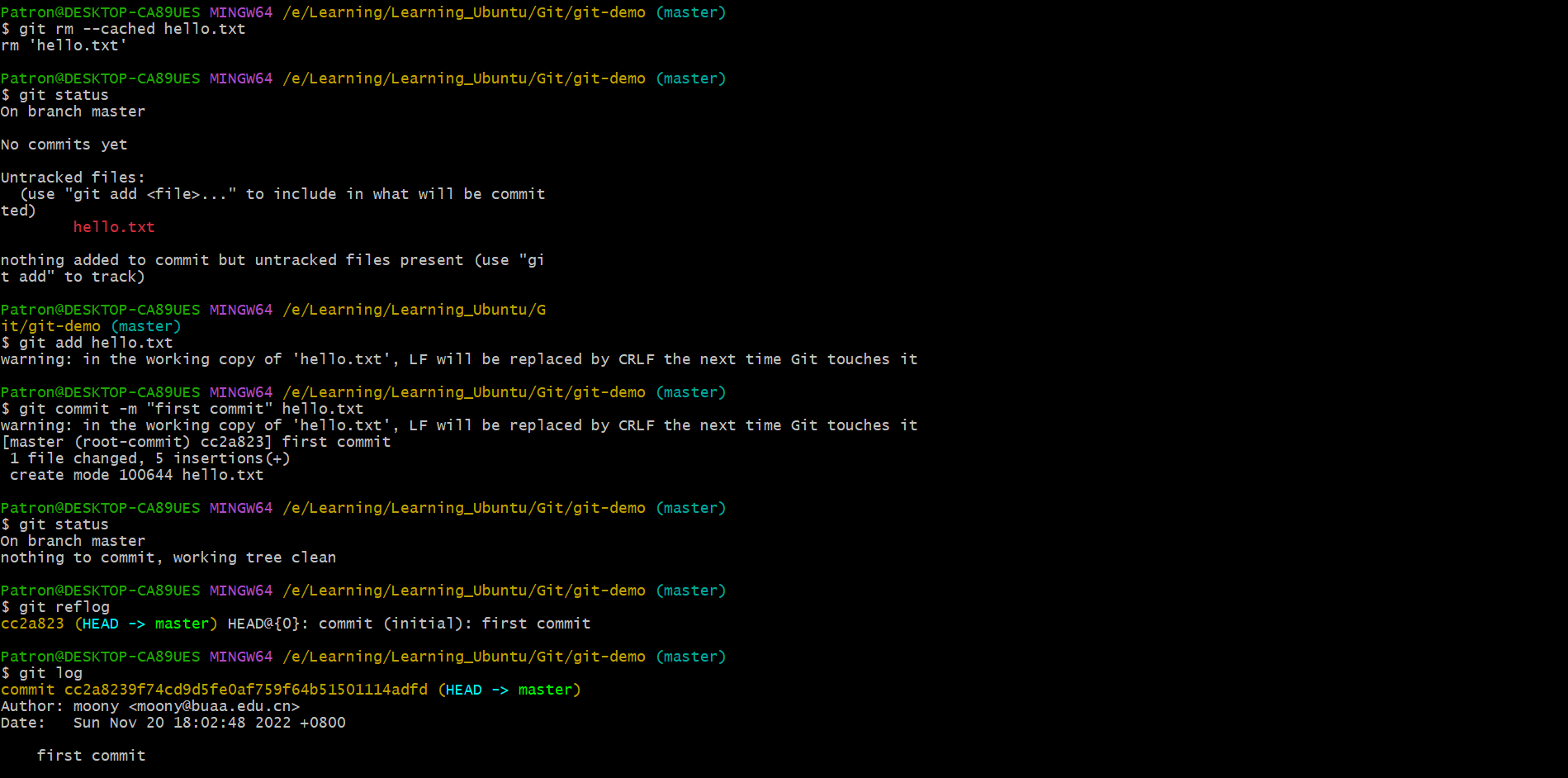
何关系。

这里设置的用户签名和邮箱并不用和GitHub的一致，甚至不用是有效的邮箱。这只是一个代称而已。

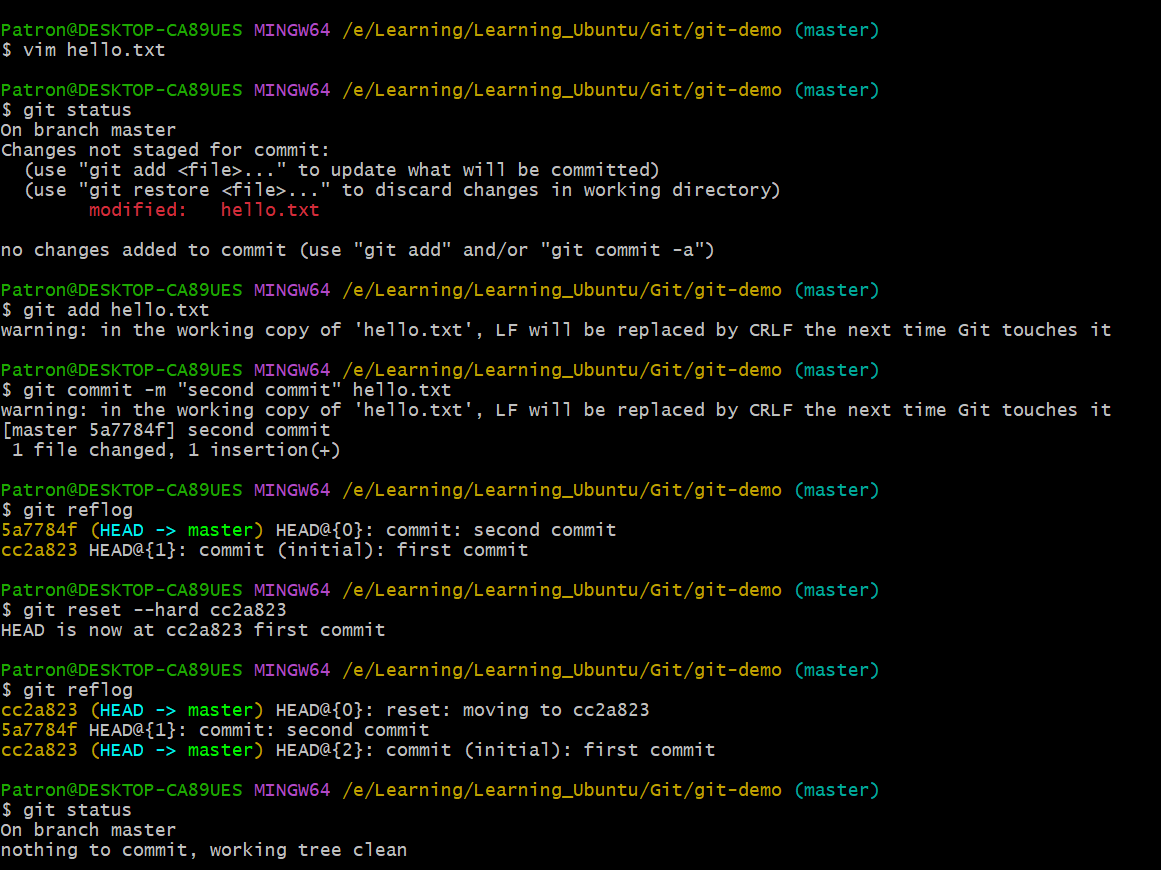
1. Git初始化本地库、查看本地库状态、添加、提交、修改、查看日志、版本穿梭…操作。

初始化、查看本地库状态、添加、



去除添加、提交到本地库、查看日志

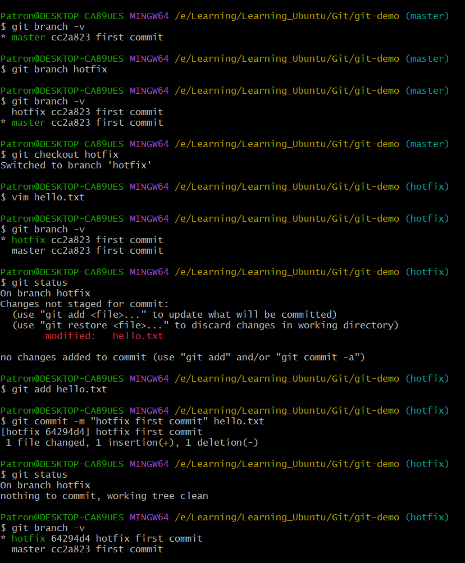
修改、版本穿梭



Git 切换版本，底层其实是移动的 HEAD 指针。

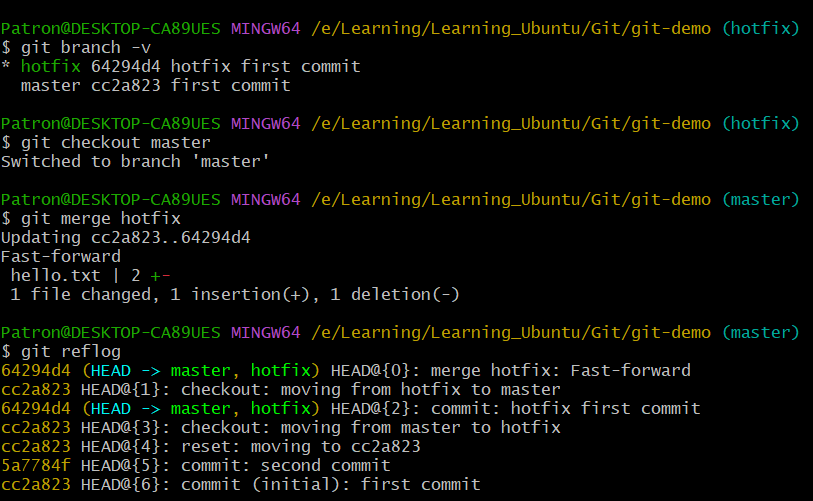
1. **Git分支及其操作**



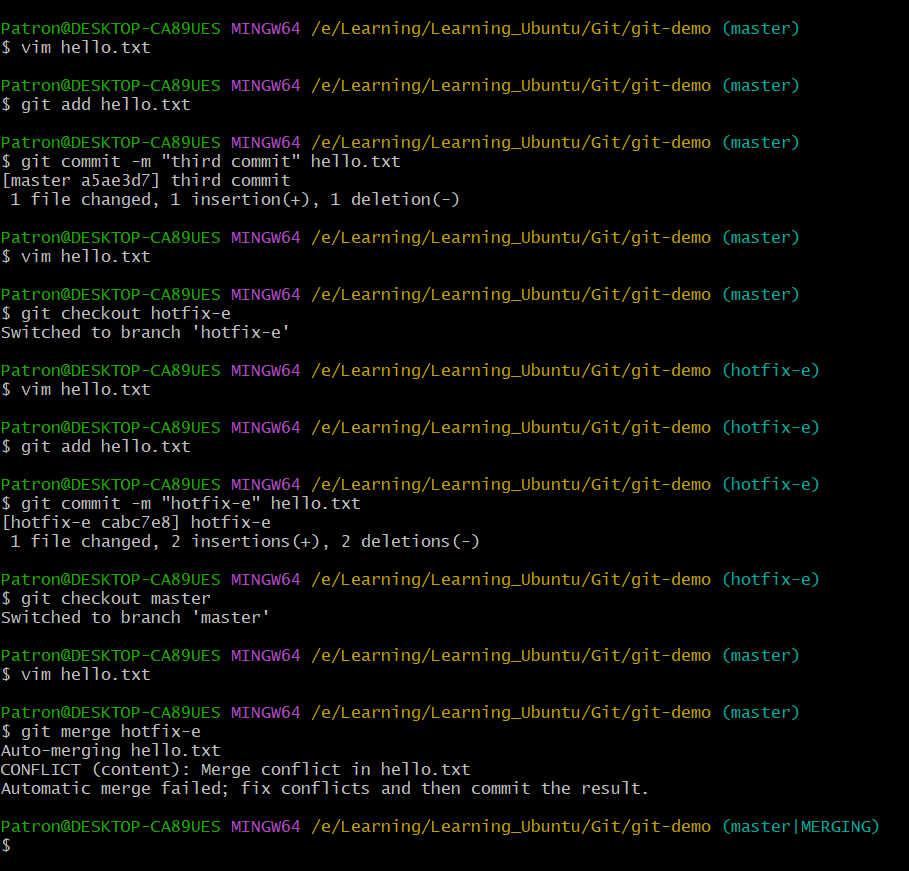
分支的创建、查看、切换。

分支的合并（正常）

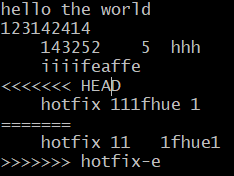
在master分支上合并hotfix分支

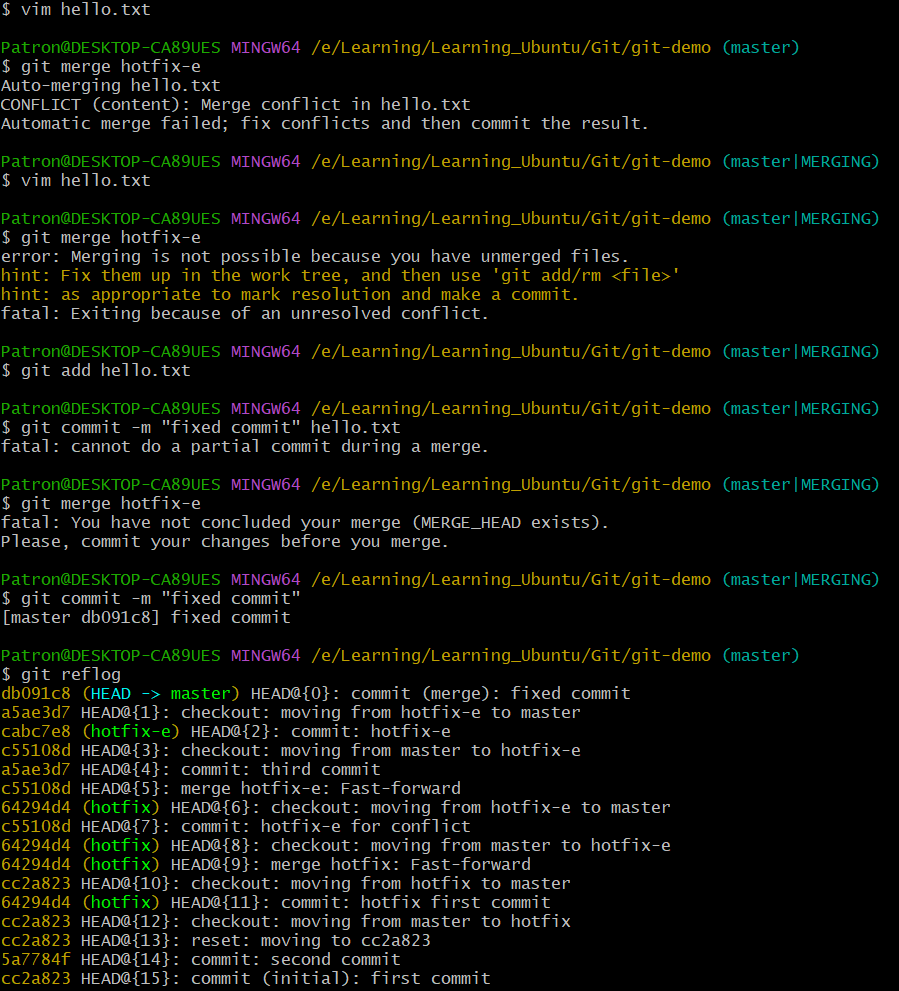


分支的合并（异常/产生异常）

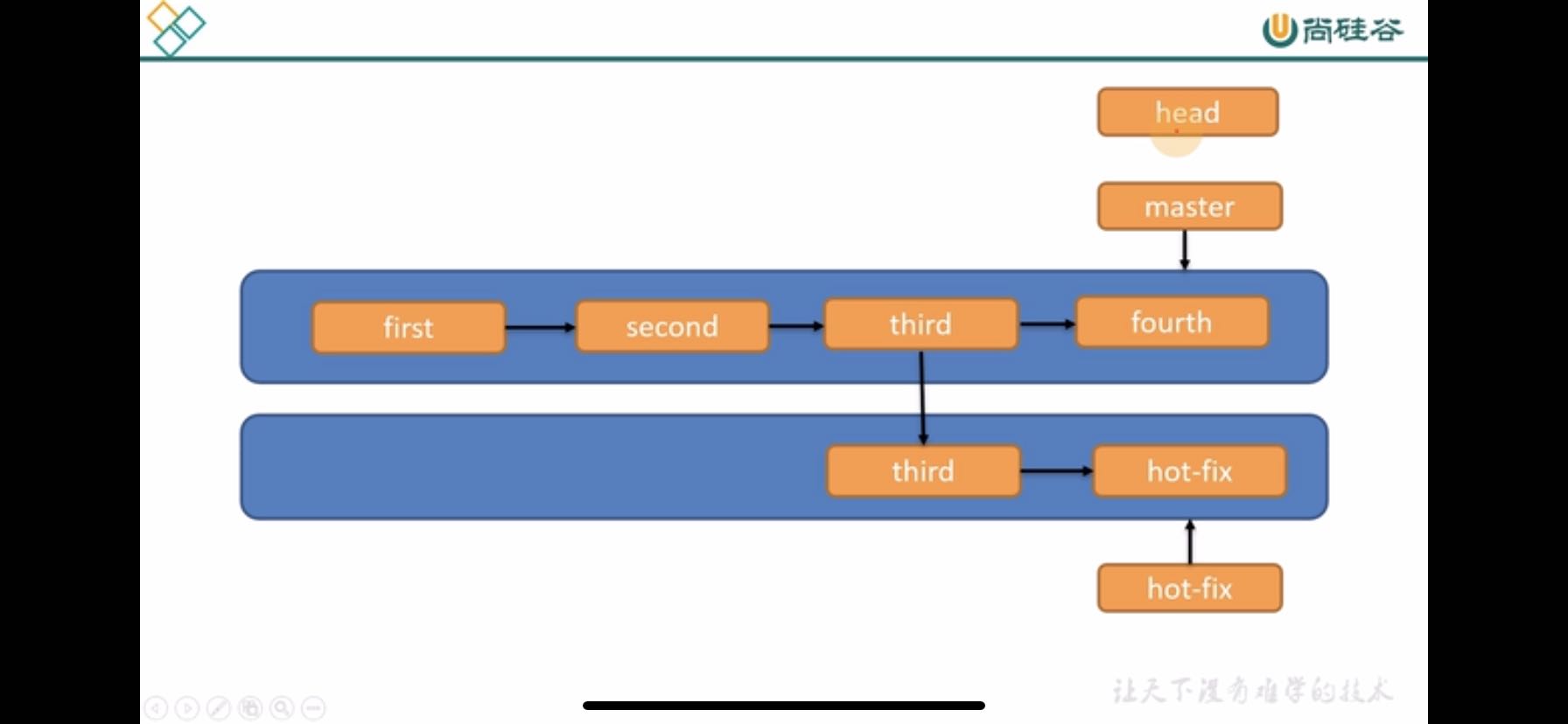


解决冲突，打开工作区中的相应文件，可看到有冲突标记。



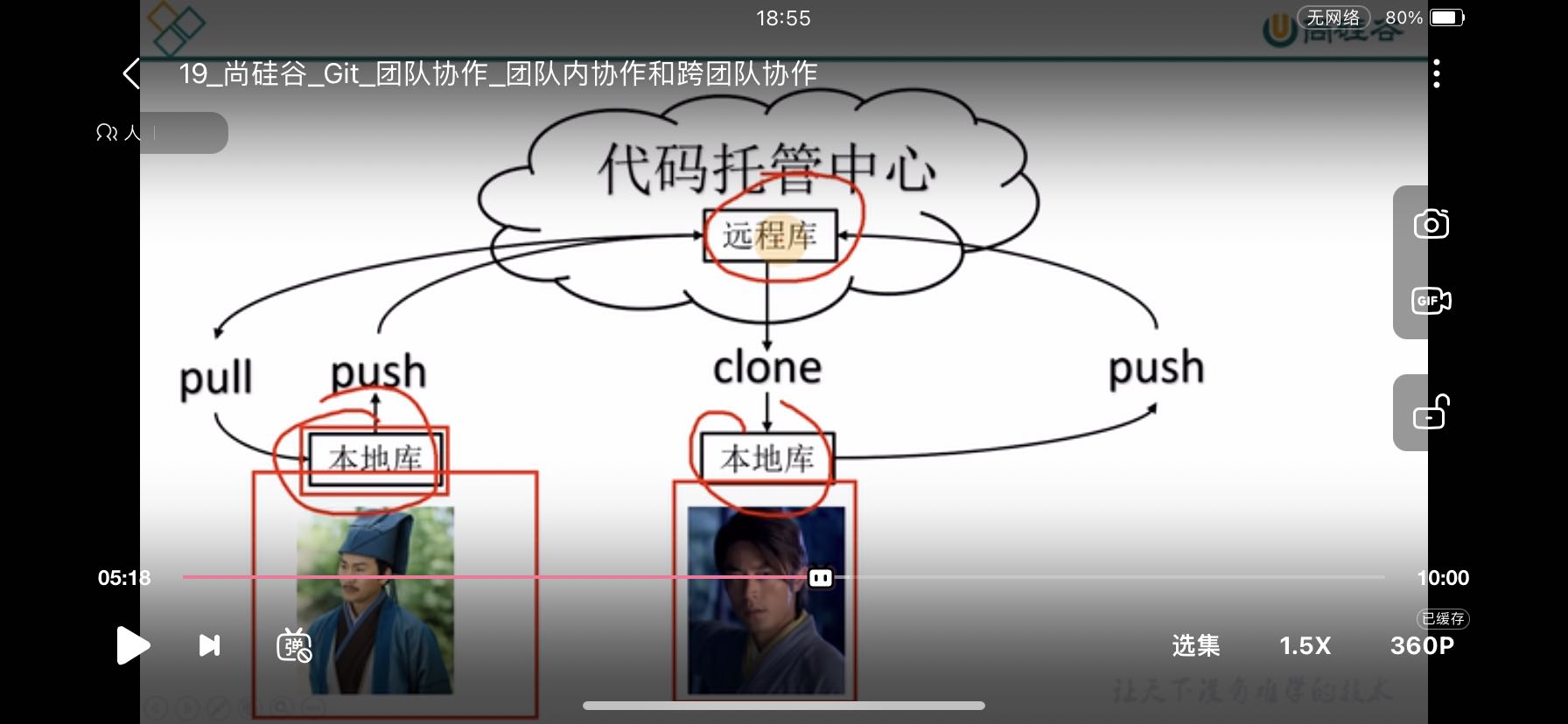


Git 切换不同分支的实质还是指针

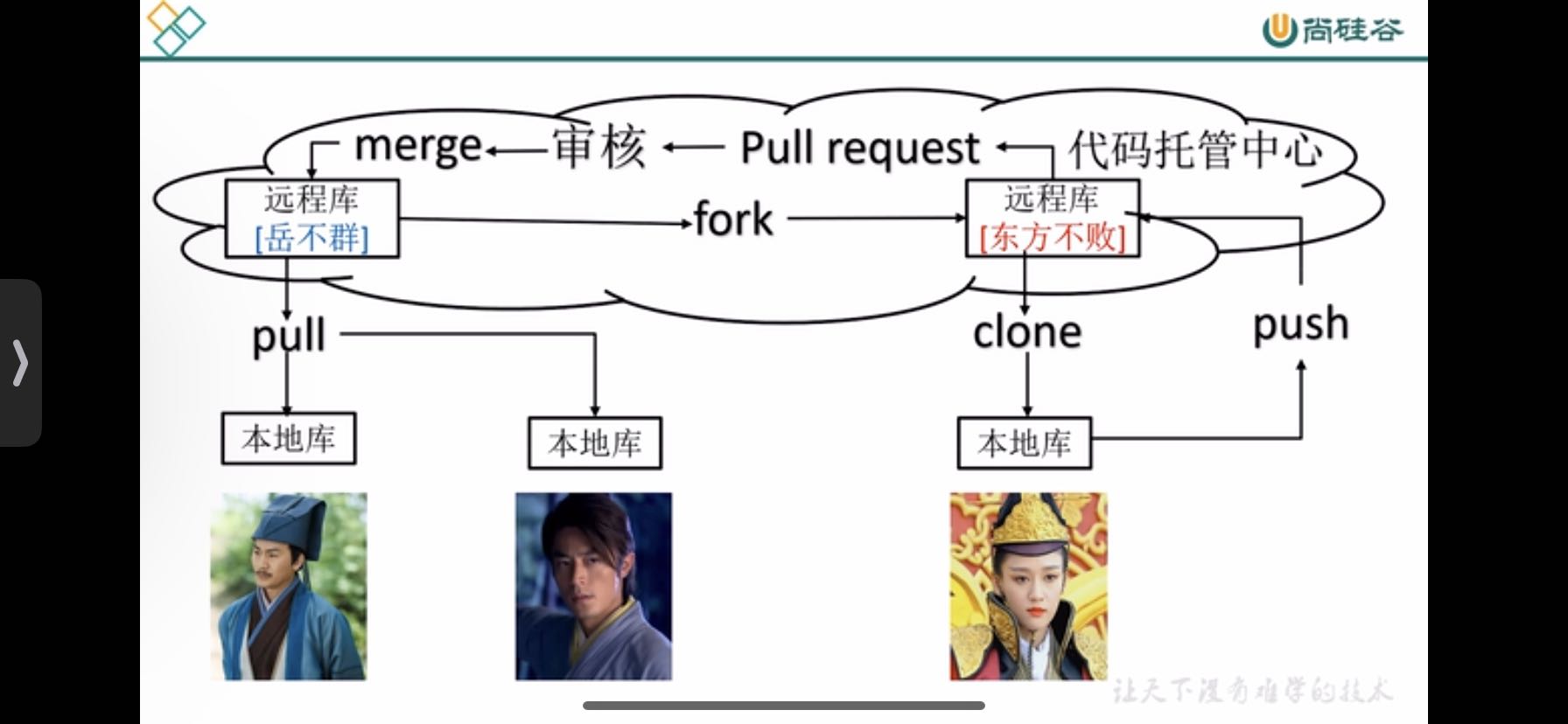


1. Git 团队协作

团队内协作



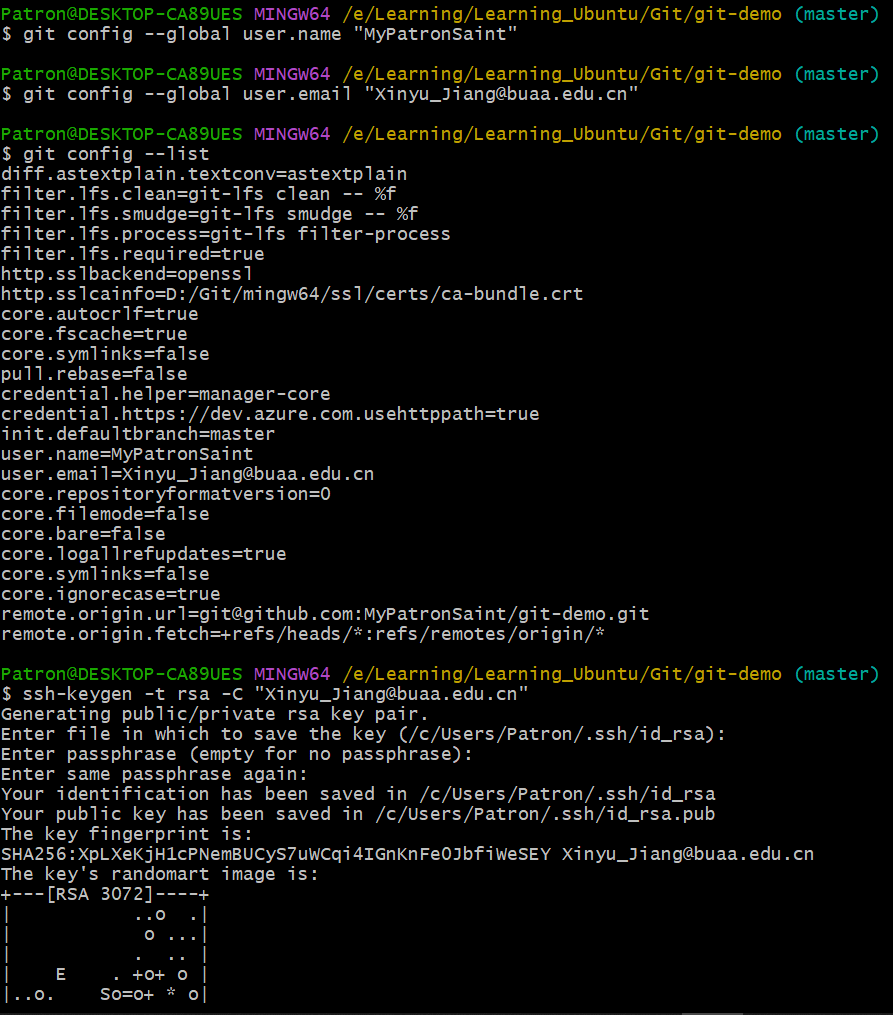
跨团队协作

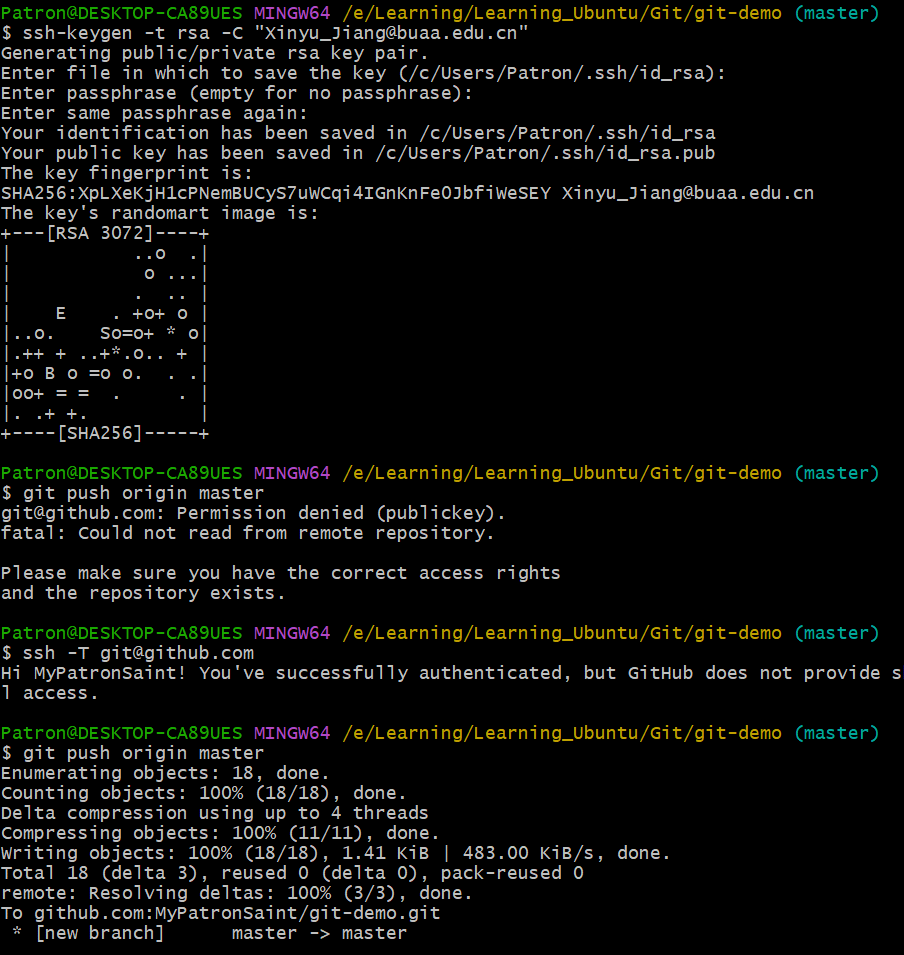


1. GitHub 操作

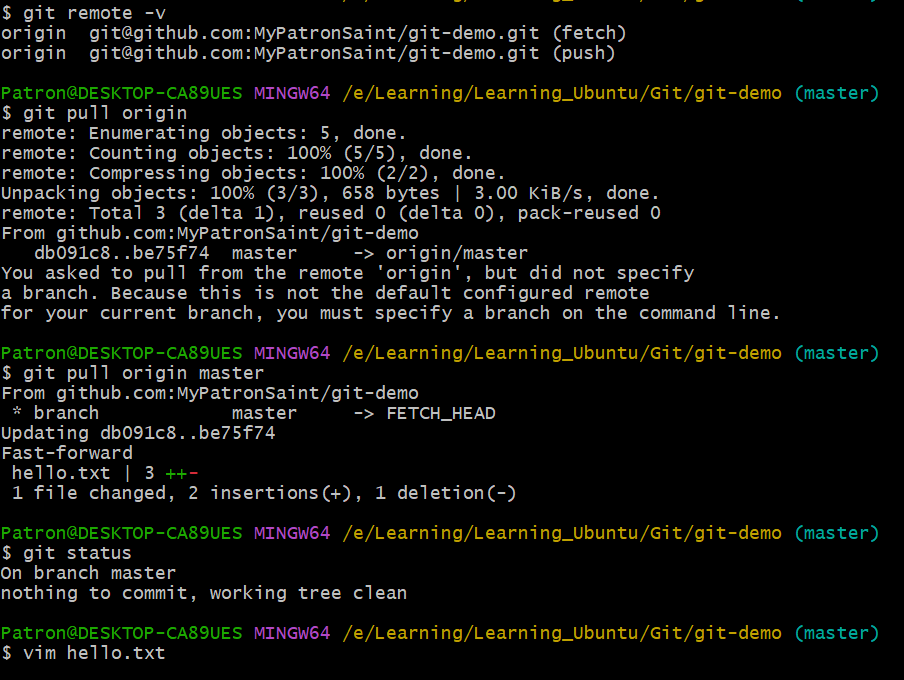


1. 目前不能使用http了，需要使用ssh。 以下是push操作

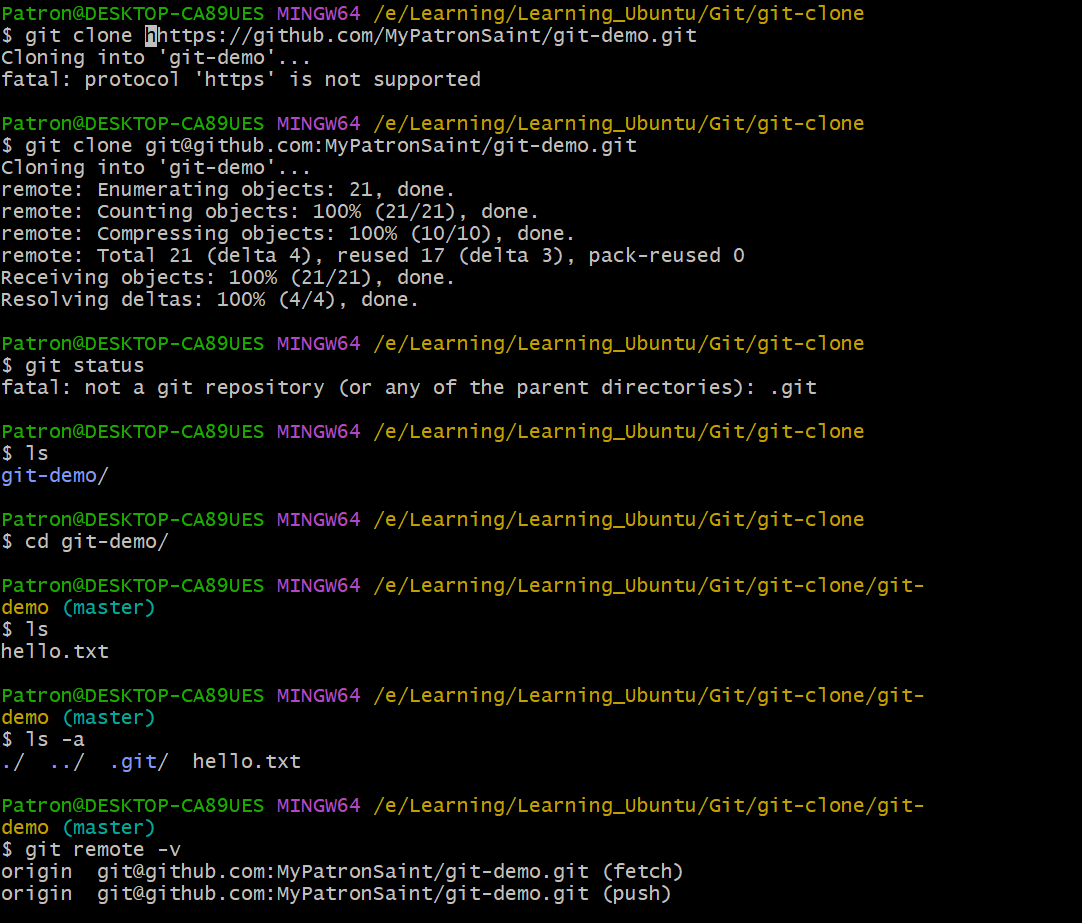




1. Pull操作（原来已经有本地库，pull只是为了修改和远程库中不一样的部分）



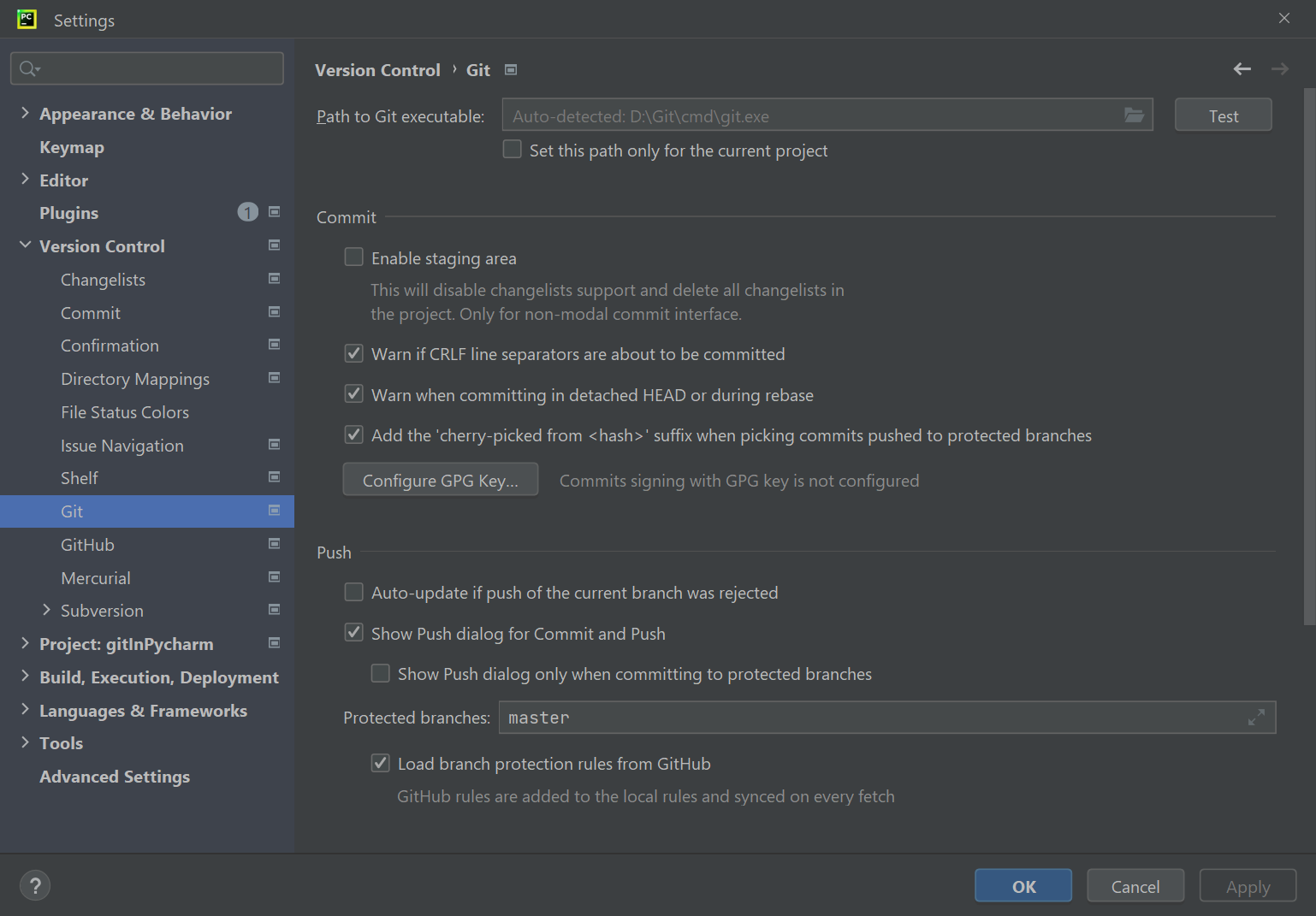
1. Clone操作（从无到有，完整的复制远程库，并且自动初始化和命名别名为origin）



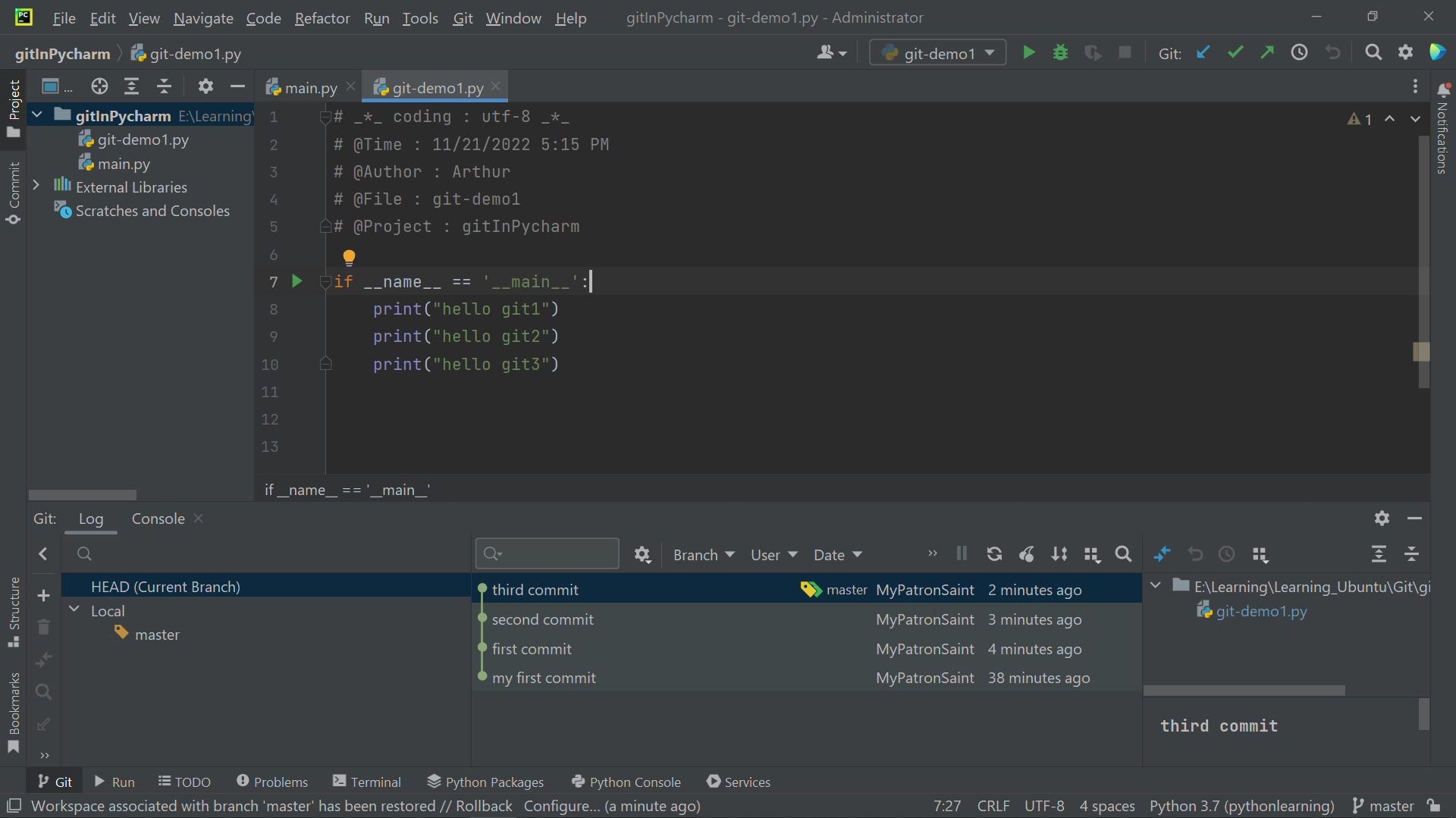
值得注意，目前clone已经不能用http了，可以用ssh或者token。

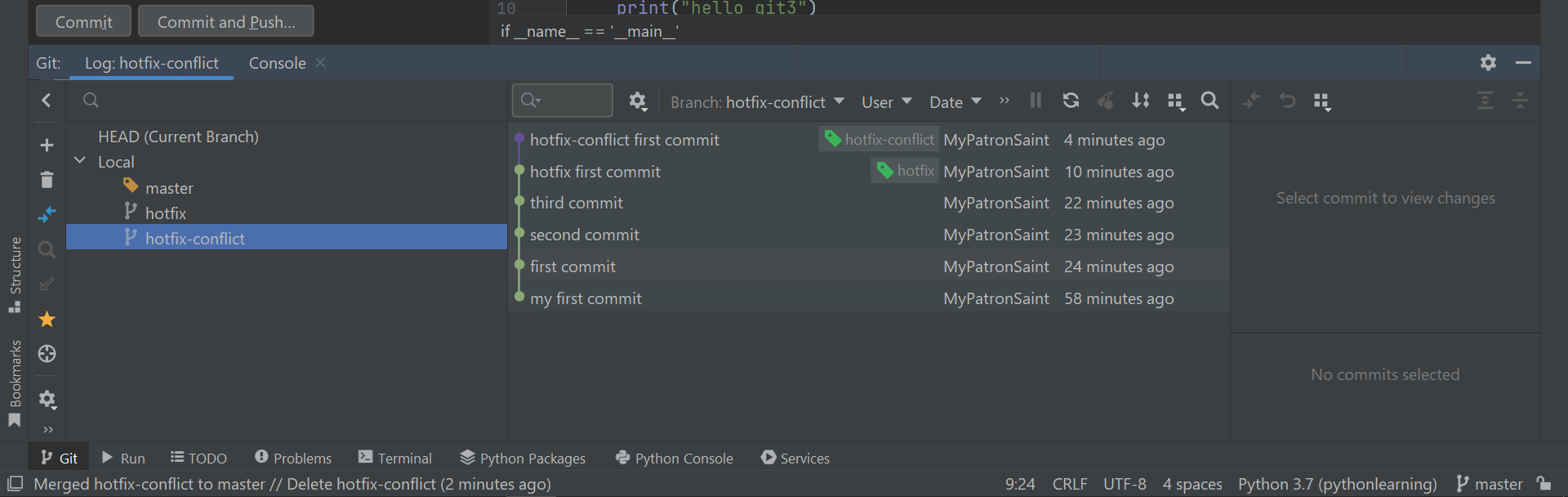
Clone完成了拉取代码、初始化本地库和创建别名（origin）这三件事情。

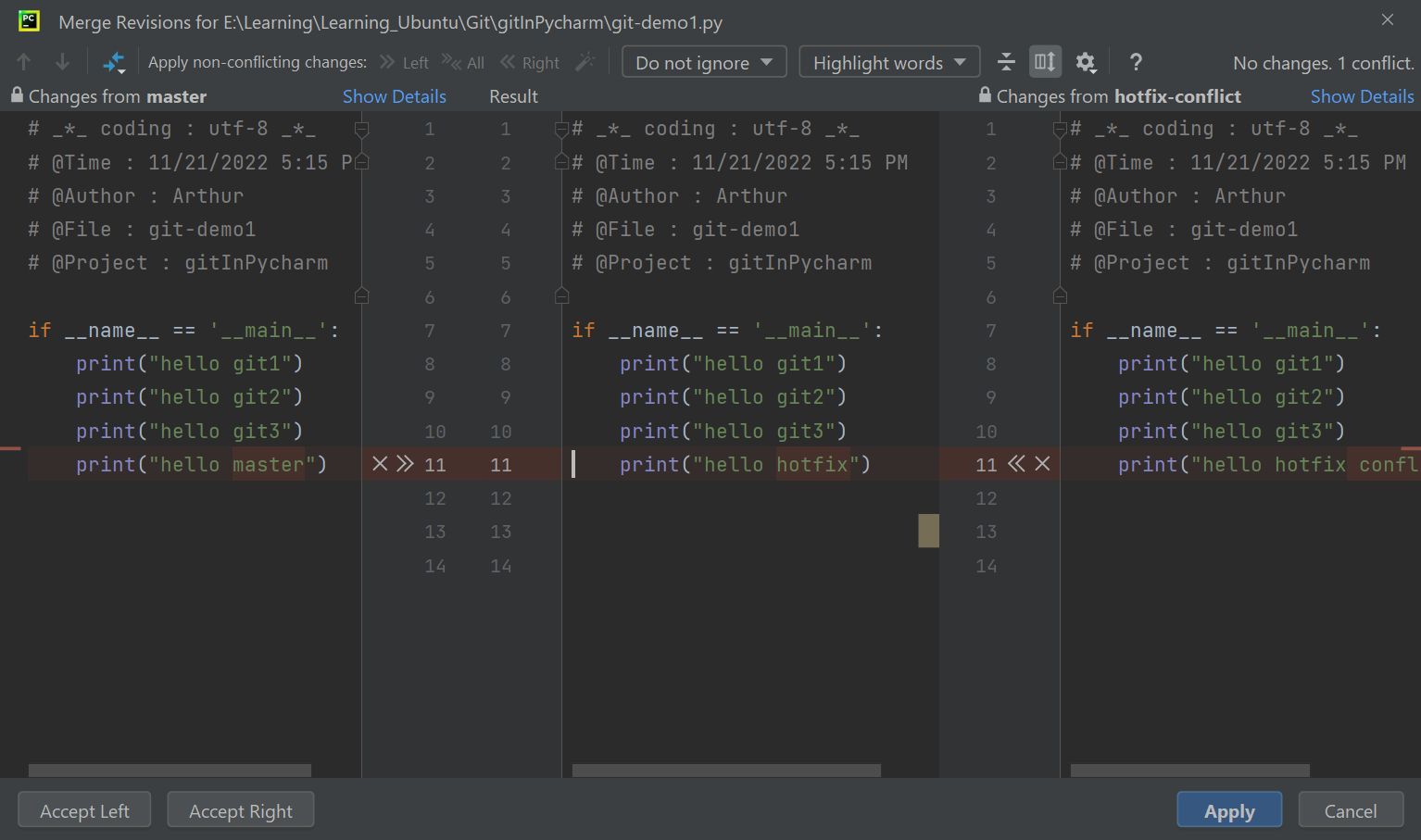
1. **在PyCharm中使用Git和GitHub**
2. 基本配置

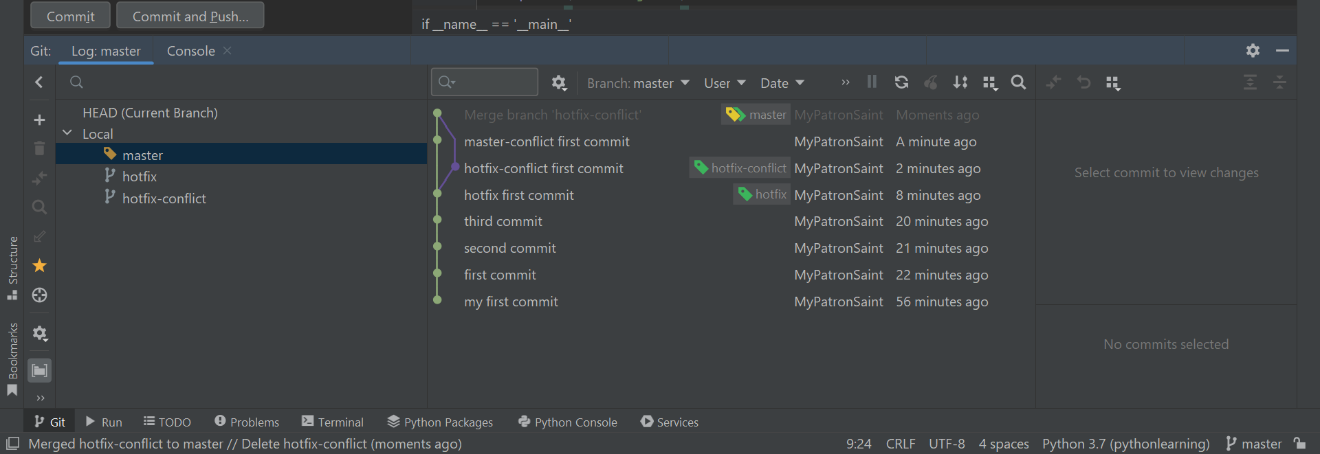


还有下边的GitHub账号用token关联一下就行了。

1. 版本穿梭
2. 创建分支，分支合并（正常），分支合并（冲突）

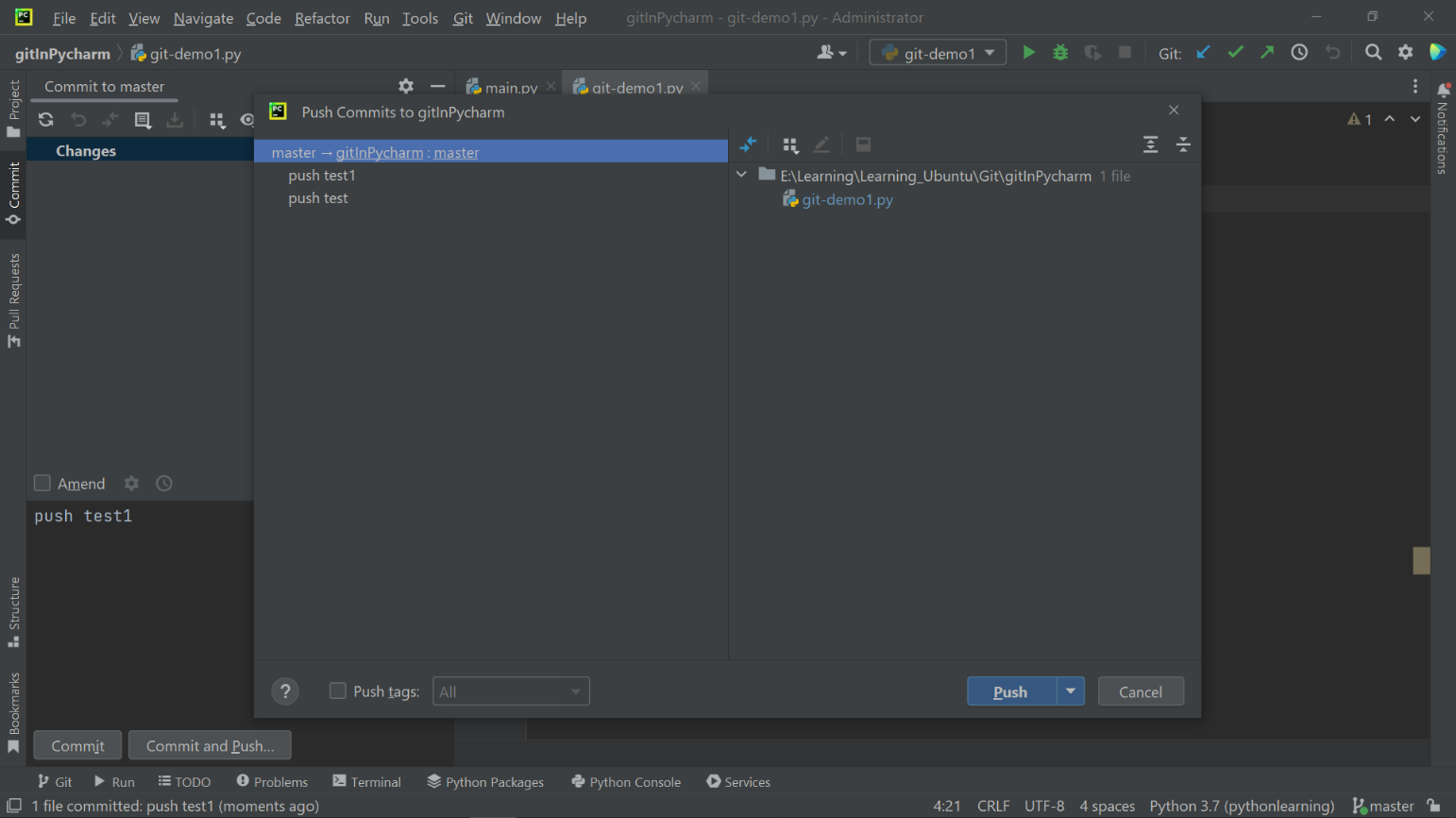
分支创建结果

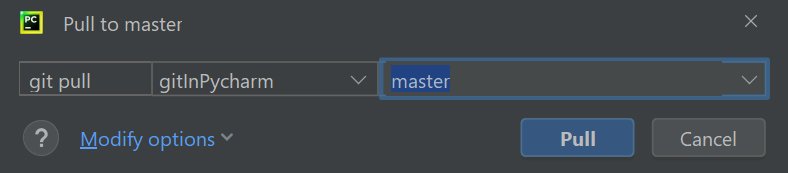
冲突的分支合并



1. GitHub 中的push, pull, clone 等操作

**使用push之前，先从远程库pull到最新的代码。**

使用ssh 进行push

从远程库中pull最新的代码

从远程库中clone代码

