# git

### 1、安装git

文本

描述已自动生成

创建本地仓库

图片包含 图示

描述已自动生成

### 2、初始化仓库

文本

描述已自动生成

### 3、git的常用逻辑

1）将代码添加到暂存区 add

2）将暂存区的代码提交到本地仓库 commit

3）本地仓库的内容上传到远程仓库 push

### 4、编写代码



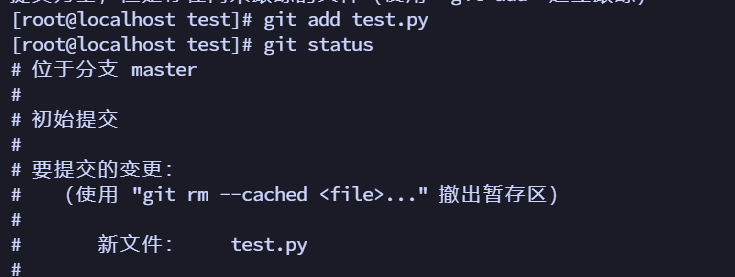
文本

描述已自动生成

图形用户界面, 文本

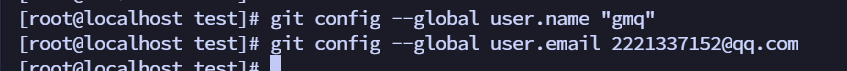
中度可信度描述已自动生成

### 5、查看新提交的文件



### 6、添加用户和邮箱

在第一次将代码提交到本地仓库之前，需要设置用户名和邮箱，（用于追踪提交的用户）之后会默认使用该用户，直到工作目录发生变化，会再次进行设置。



### 7、将暂存区的的代码提交到本地仓库

文本

描述已自动生成

### 8、代码回滚

#### 1、第一种情况：代码存储在暂存区中进行回滚（未提交到本地仓库）

更改代码 因为相同代码没办法反复提交

文本

描述已自动生成

文本

描述已自动生成

撤出暂存区

图形用户界面, 文本

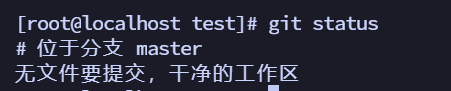
中度可信度描述已自动生成

代码回滚成功，丢弃掉工作区，将会吧代码回滚到上次修改之前。

文本

描述已自动生成

再次查看状态



#### 2、代码已经提交到本地仓库进行回滚

修改代码

文本

描述已自动生成

添加到暂存区 再添加到仓库 此时查看暂存区则是干净的 因为已经将文件存到了仓库’

文本

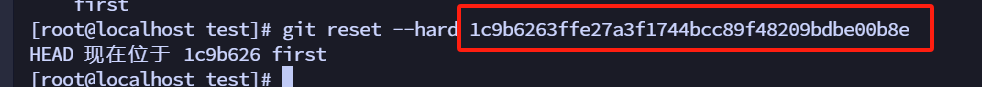
描述已自动生成

查看两次提交记录

文本

描述已自动生成

填写你回滚的ID 查看代码内容



回滚成功

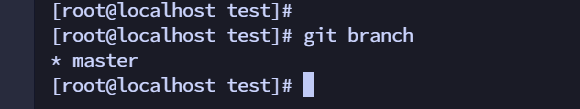
文本

描述已自动生成

### 9、分支

主干：git是以时间为主线对版本进行管理的，这条时间线就是git的主干，主干上每一个节点就是一次提交，即为一个版本，在主干上用户可以定义多个指针，指向不同的节点git默认会创建一个叫做master的指针，默认情况下用户操作都是在master上进行的，但是用户可以对操作的指针进行切换，用户每提交一次，就会形成一个新的节点，当前指针就会向前移动一次。

**查看当前分支（\*代表当前所在的分支）**



创建分支

文本

描述已自动生成

切换分支

文本

描述已自动生成

修改代码文件 提交文件到仓库

文本

描述已自动生成

切换到master主干上查看文件内容 **主干和分干的内容是互不影响的**

文本

描述已自动生成

将分支代码合并到主干

文本

描述已自动生成

### 10、GitHub

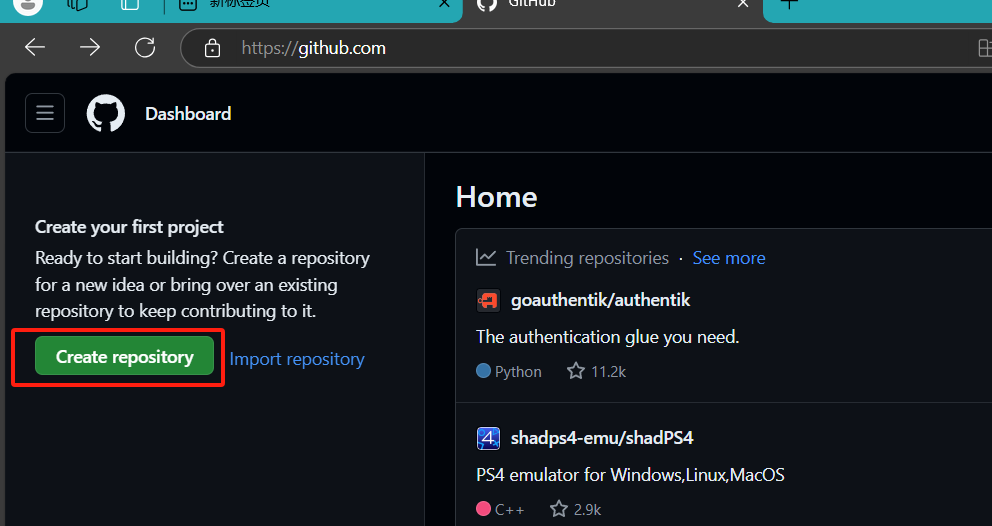
GitHub远程仓库（[www.github.com](http://www.github.com)）

登录github

电脑萤幕的截图

描述已自动生成

创建仓库



创建一个aaa的仓库

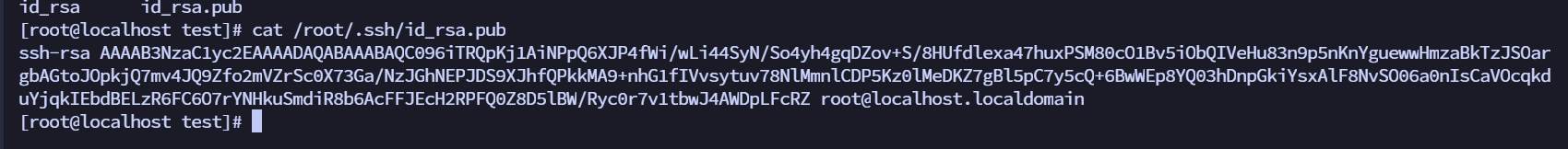


本地仓库与远程仓库进行链接（需要用到密钥对）

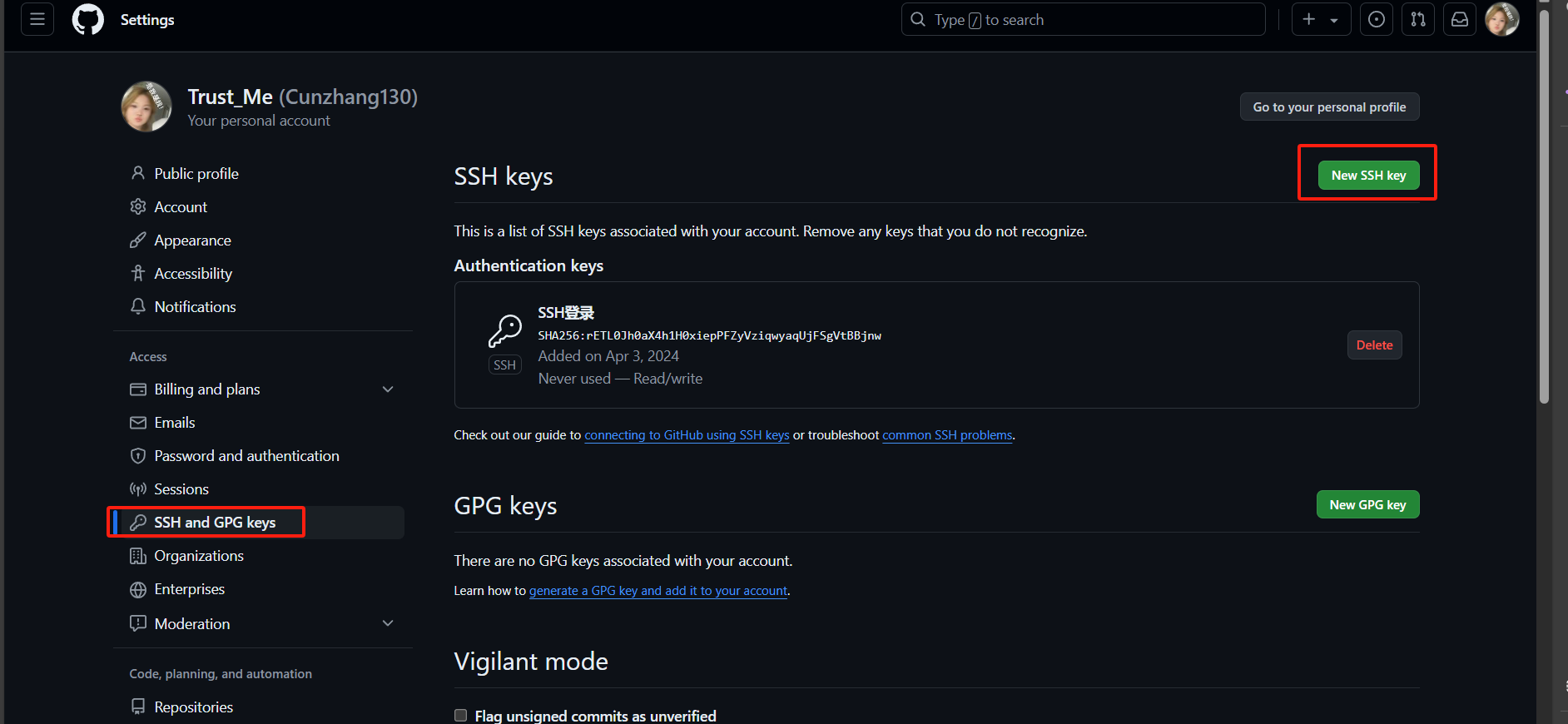
在本地生成密钥对



查看公钥文件的内容



回到github页面添加ssh



添加公钥

图形用户界面, 文本, 应用程序

描述已自动生成

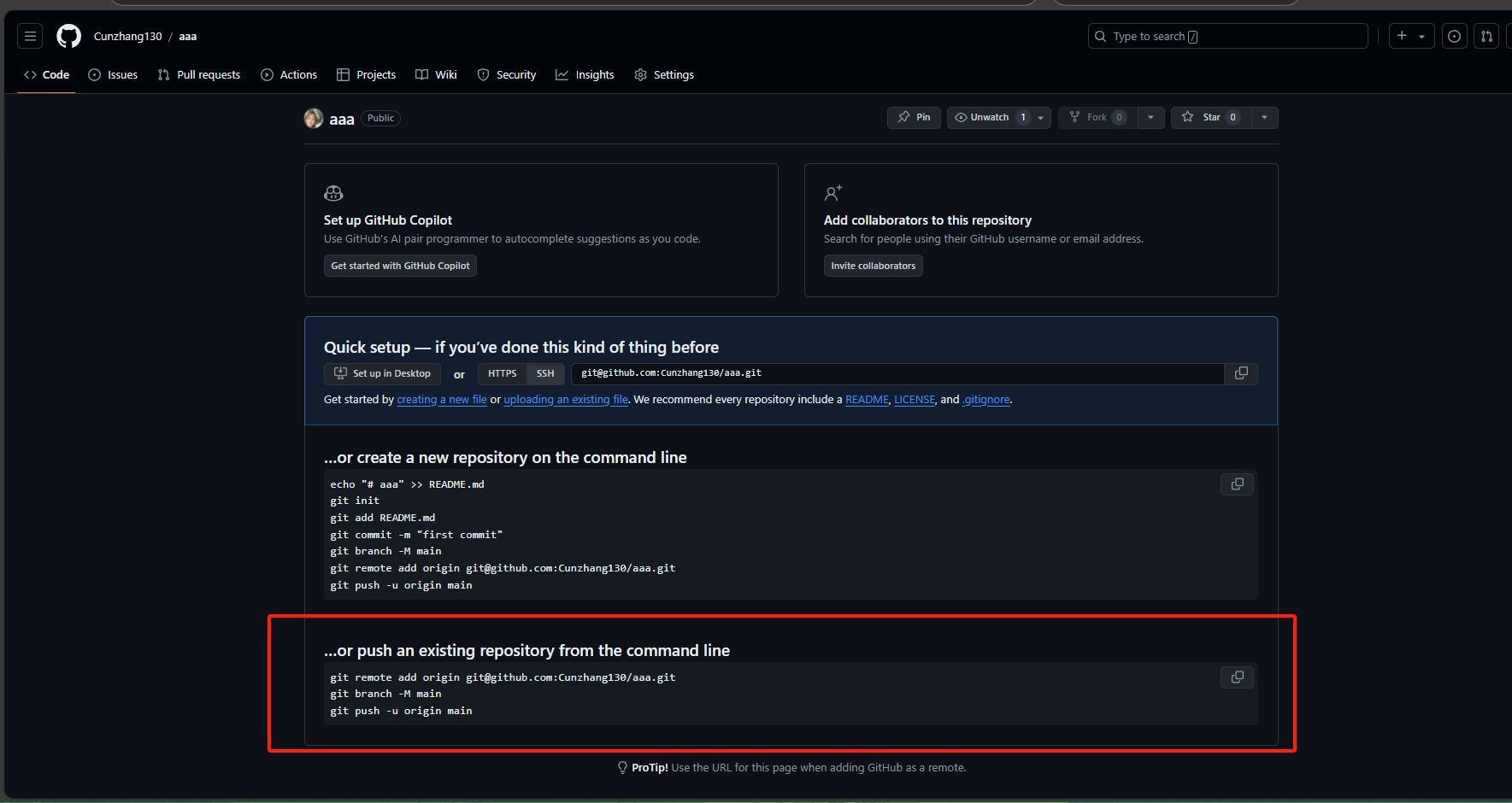
添加完成之后本地链接一下远程仓库测试是否链接

图片包含 文本

描述已自动生成

向远程仓库推送数据

返回仓库下翻查看 添加仓库的方式



设置选项内——开发者设置

电脑萤幕的截图

描述已自动生成

将虚拟机的数据推送到远程仓库

文本

描述已自动生成

将权限全部添加

屏幕上有字

描述已自动生成

复制生成的默认值

图形用户界面, 文本, 应用程序

描述已自动生成

复制第一行指令 键入到本地 对远程仓库进行链接

屏幕上有字

描述已自动生成

建立链接

在工作目录下键入指令才可以生效

将在github仓库复制的链接指令键入到本地终端中建立链接



修改选项——添加默认值 建立链接



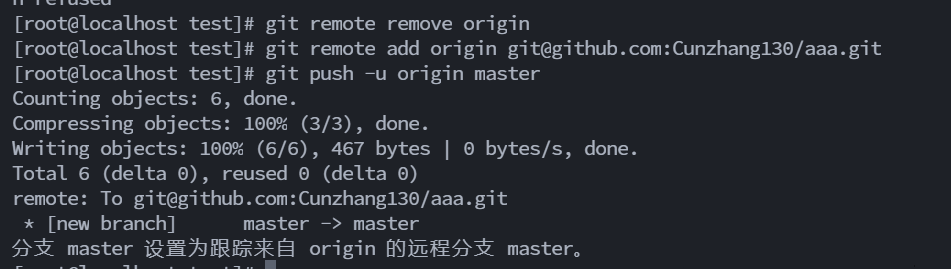
将master分支更改名称

git branch -M <主干名称>

将本地仓库代码上传到远程仓库

git push -u origin <主干名称>

这边数据上传为notfound 尝试删除建立的远程链接使用ssh重新链接 && 再次上传



### 11、克隆（下载）

克隆github上的代码到本地

git clone