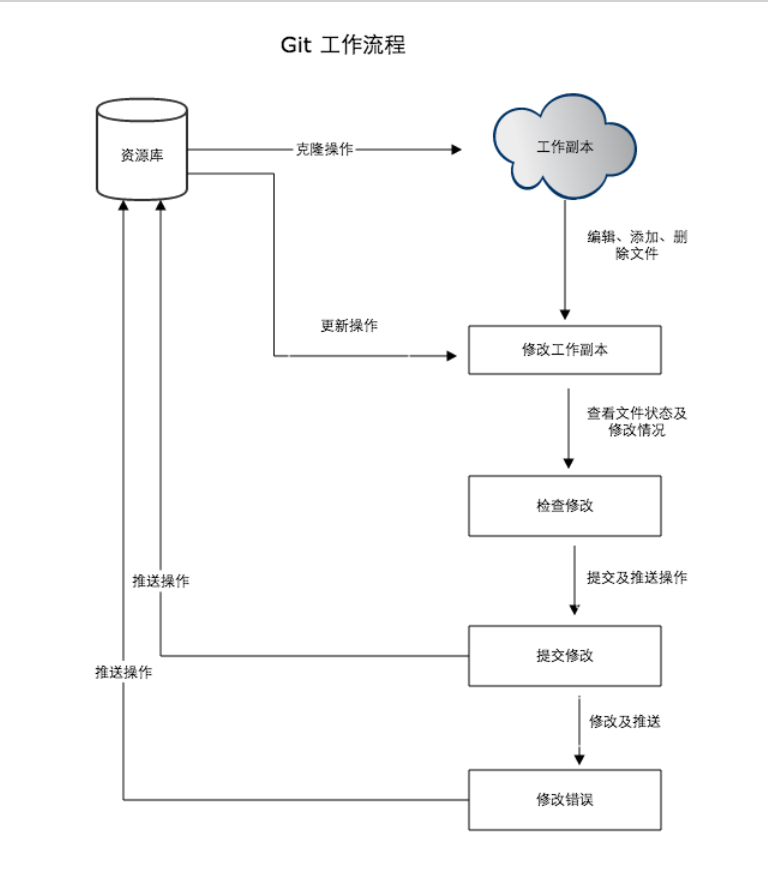
1. 在linux系统创建版本库：mkdir 库名；-> cd 库名-> 查看当前目录:pwd;
2. 通过git init命令把这个目录变成Git可以管理的库；
3. git工作流程：



### **创建仓库命令:**

1. 初始化命令:git init;
2. 从远程仓库克隆一个项目:git clone [url];

## 提交与修改命令:

1. Git add 命令：
   1. 添加一个或者多个文件到缓存区：git add [file1] [file2];
   2. 添加指定目录到缓存区，包括子目录:git add [dir];
   3. 添加当前目录所有文件到缓存区:git add;
2. git status 命令
   1. 用于查看项目的当前状态;
   2. AM 状态的意思是这个文件在我们将它添加到缓存之后又有改动。改动后我们再执**git add .** 命令将其添加到缓存中：
3. git diff 命令:
   1. git diff 有两个主要的应用场景。

* 尚未缓存的改动：**git diff**
* 查看已缓存的改动： **git diff --cached**
* 查看已缓存的与未缓存的所有改动：**git diff HEAD**
* 显示摘要而非整个 diff：**git diff --stat**

1. git commit 命令：
   1. 前面章节我们使用 git add 命令将内容写入暂存区。git commit 命令将暂存区内容添加到本地仓库中。
   2. 提交暂存区到本地仓库中:git commit -m [message] 注：message 可以是一些备注信息;
   3. 提交暂存区的指定文件到仓库区：$ git commit [file1] [file2] ... -m [message]
   4. 添加-a 参数设置修改文件后不需要执行 git add 命令，直接来提交$ git commit -a
2. git reset 命令
   1. git reset 命令用于回退版本，可以指定退回某一次提交的版本。注意：谨慎使用 –hard 参数，它会删除回退点之前的所有信息。

**HEAD 说明：**

HEAD 表示当前版本

HEAD^ 上一个版本

HEAD^^ 上上一个版本

HEAD^^^ 上上上一个版本

以此类推...

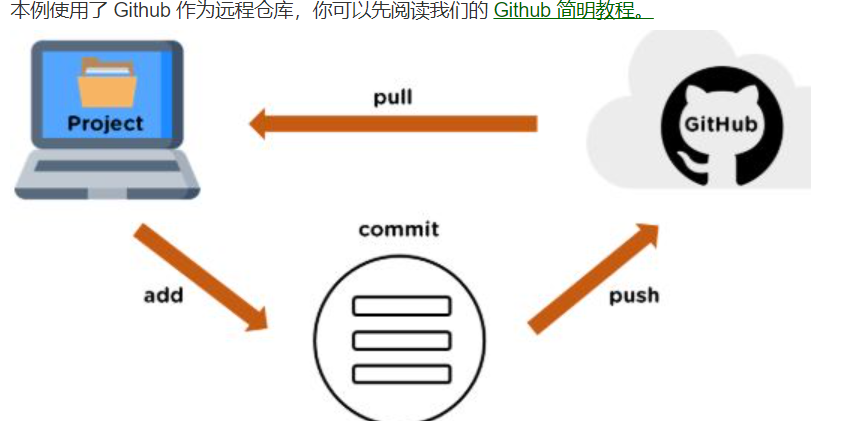
1. git rm 命令:git rm [file] ;强制删除命令:git rm -f [file];
2. git mv 命令用于移动或重命名一个文件、目录或软连接。

**提交日志**

1. git log 查看历史提交记录；

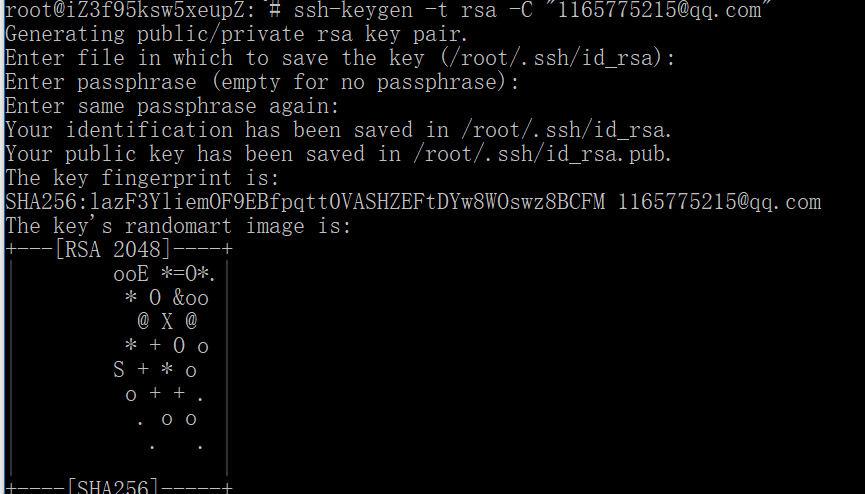
### **远程操作**

1. git remote 命用于在远程仓库的操作
2. git fetch 命令用于从远程获取代码库。

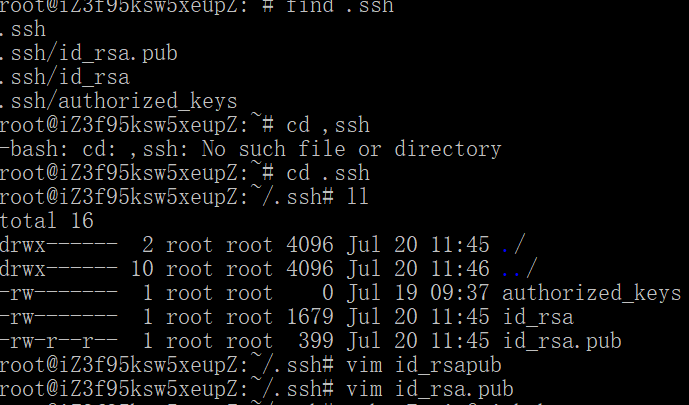
远程仓库：

连接远程仓库:

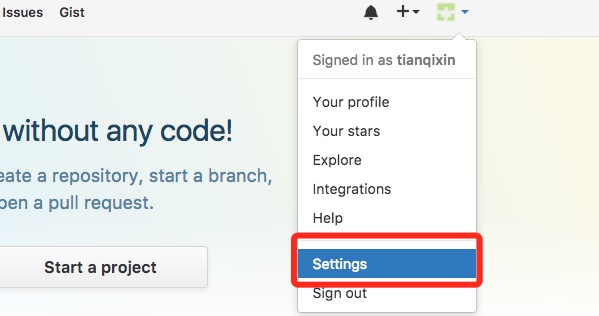
1. 由于你的本地 Git 仓库和 GitHub 仓库之间的传输是通过SSH加密的，所以我们需要配置验证信息：
2. 使用以下命令生成 SSH Key：**$ ssh-keygen -t rsa -C ["youremail@example.com";](mailto:\"youremail@example.com\";)**



1. 后面的 your\_email@youremail.com 改为你在 Github 上注册的邮箱，之后会要求确认路径和输入密码，我们这使用默认的一路回车就行。成功的话会在 ~/ 下生成 .ssh 文件夹，进去，打开 id\_rsa.pub，复制里面的 key。

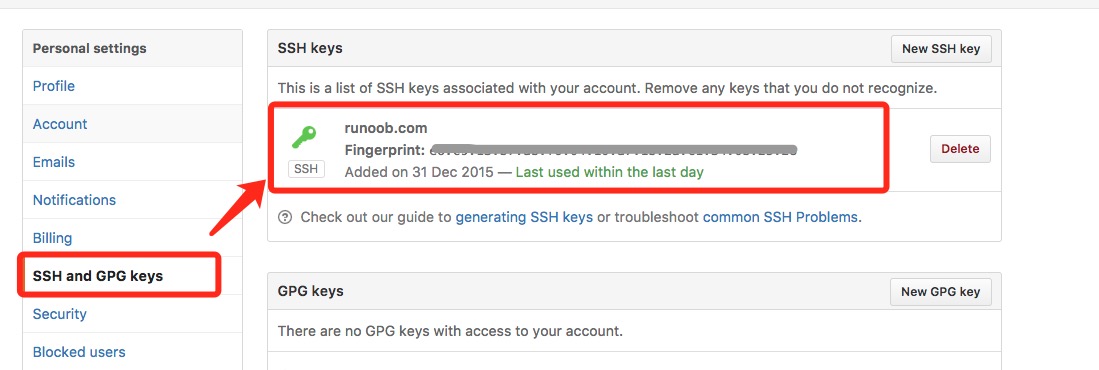


1. 在github上选择setting

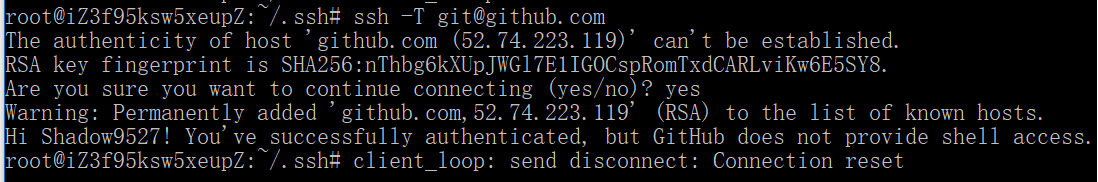


1. 左边选择 SSH and GPG keys，然后点击 New SSH key 按钮,title 设置标题，可以随便填，粘贴在你电脑上生成的 key。

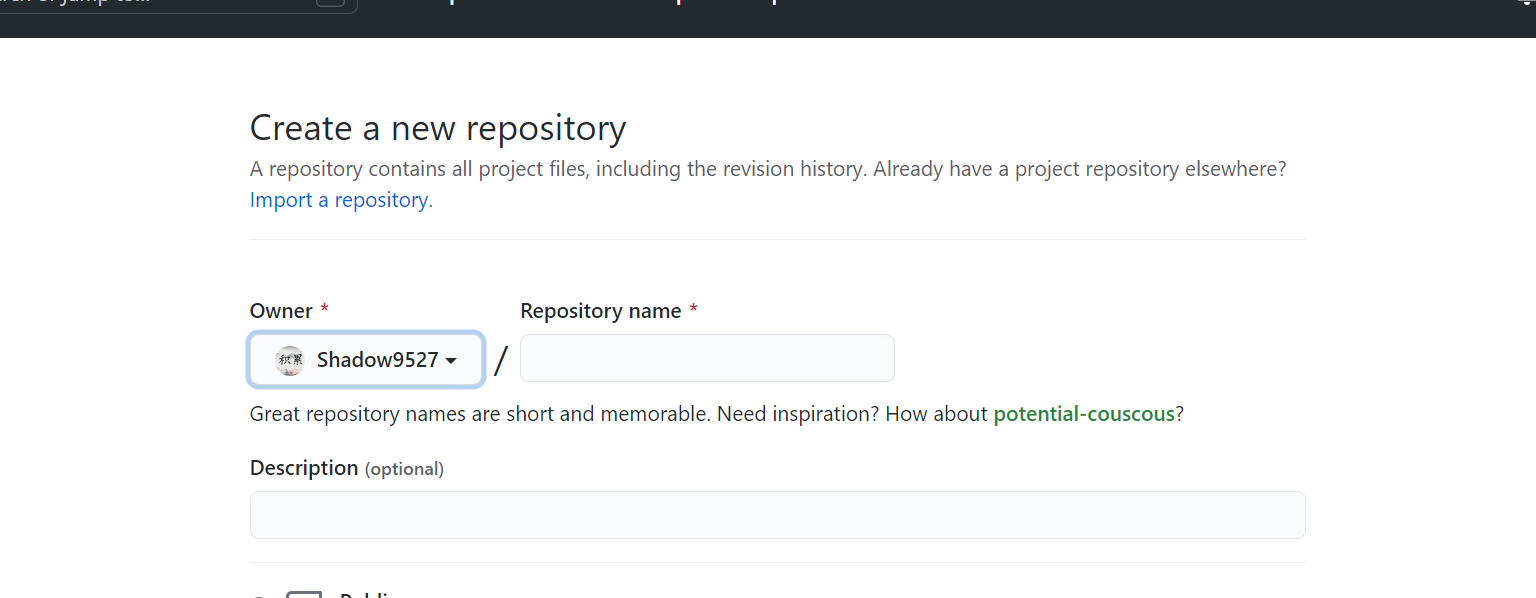
添加成功后：



1. 输入测试连接命令:**ssh -T [git@github.com](mailto:git@github.com)**
2. 出现下面界面则成功：



1. 在github上面新建仓库：

之后在在Repository name 填入 runoob-git-test(远程仓库名) ，其他保持默认设置，点击"Create repository"按钮，就成功地创建了一个新的Git仓库

1. 创建成功后：



1. 在本地运行命令：

$ mkdir runoob-git-test # 创建测试目录

$ cd runoob-git-test/ # 进入测试目录

$ echo "# 菜鸟教程 Git 测试" >> README.md # 创建 README.md 文件并写入内容

$ ls # 查看目录下的文件

README

$ git init # 初始化

$ git add README.md # 添加文件

$ git commit -m "添加 README.md 文件" # 提交并备注信息[master (root-commit) 0205aab] 添加 README.md 文件

1 file changed, 1 insertion(+)

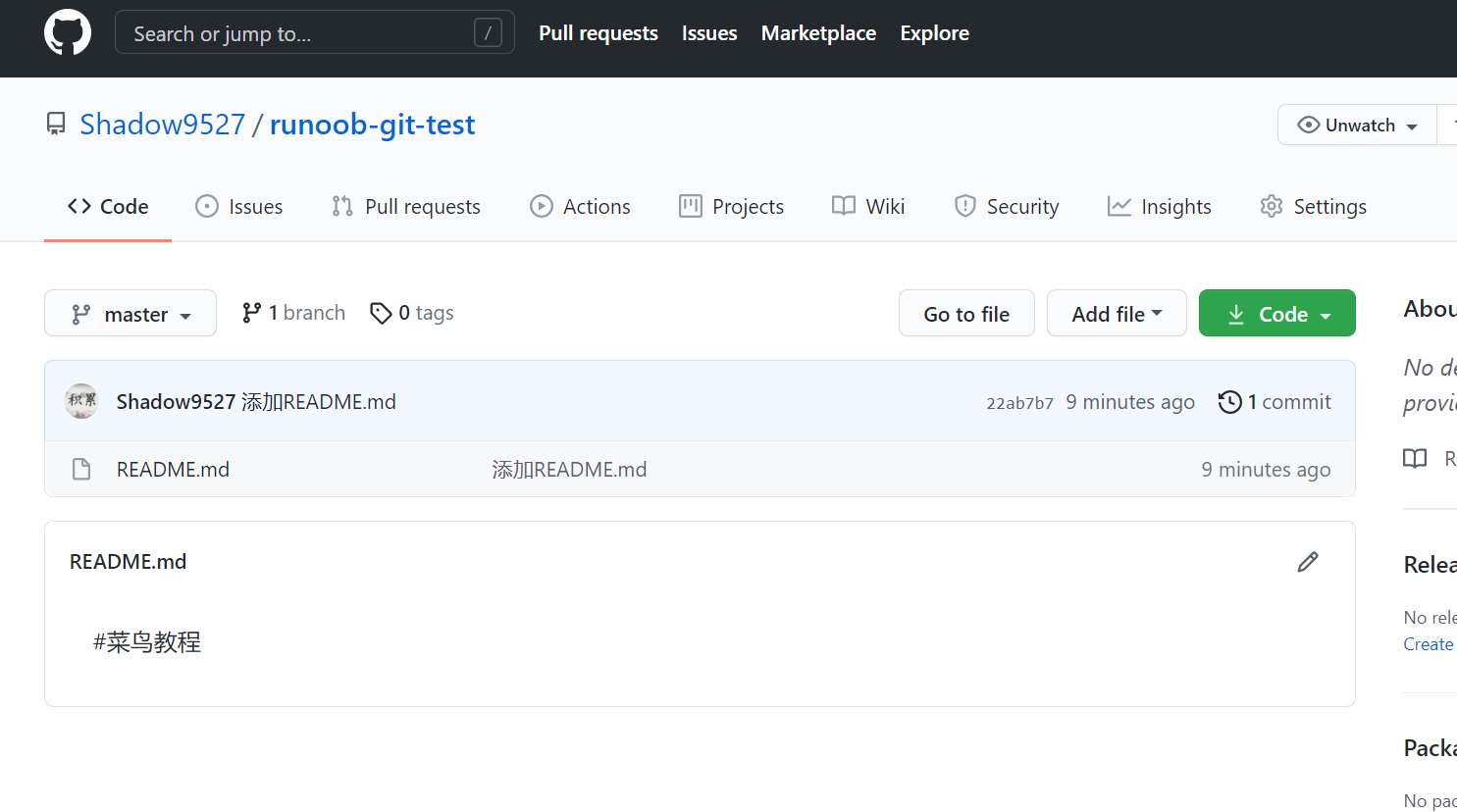
create mode 100644 README.md

# 提交到 Github

$ git remote add origin git@github.com:tianqixin/runoob-git-test.git

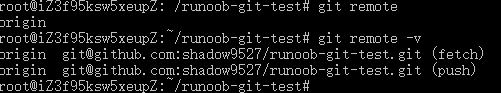
$ git push -u origin master

1. 推送上去后：



## 查看当前的远程库

1. Git remote 查看当前配置有哪些远程仓库：



## 提取远程仓库

1. 从远程仓库下载新分支与数据：git fetch
2. 从远端仓库提取数据并尝试合并到当前分支：git merge
3. 再执行：git fetch origin
4. 查看文件内容:cat README.md

## 删除远程仓库

1. git remote rm [别名]

$ git remote -v

origin    git@github.com:tianqixin/runoob-git-test.git (fetch)

origin    git@github.com:tianqixin/runoob-git-test.git (push)

# 添加仓库 origin2

$ git remote add origin2 git@github.com:tianqixin/runoob-git-test.git

$ git remote -v

origin    git@github.com:tianqixin/runoob-git-test.git (fetch)

origin    git@github.com:tianqixin/runoob-git-test.git (push)

origin2    git@github.com:tianqixin/runoob-git-test.git (fetch)

origin2    git@github.com:tianqixin/runoob-git-test.git (push)

# 删除仓库 origin2

$ git remote rm origin2

$ git remote -v

origin    git@github.com:tianqixin/runoob-git-test.git (fetch)

origin    git@github.com:tianqixin/runoob-git-test.git (push)