

# 一、Git是分布式版本控制系统

## 1. 版本控制 [什么是版本控制？ - Azure DevOps | Microsoft Learn](#)

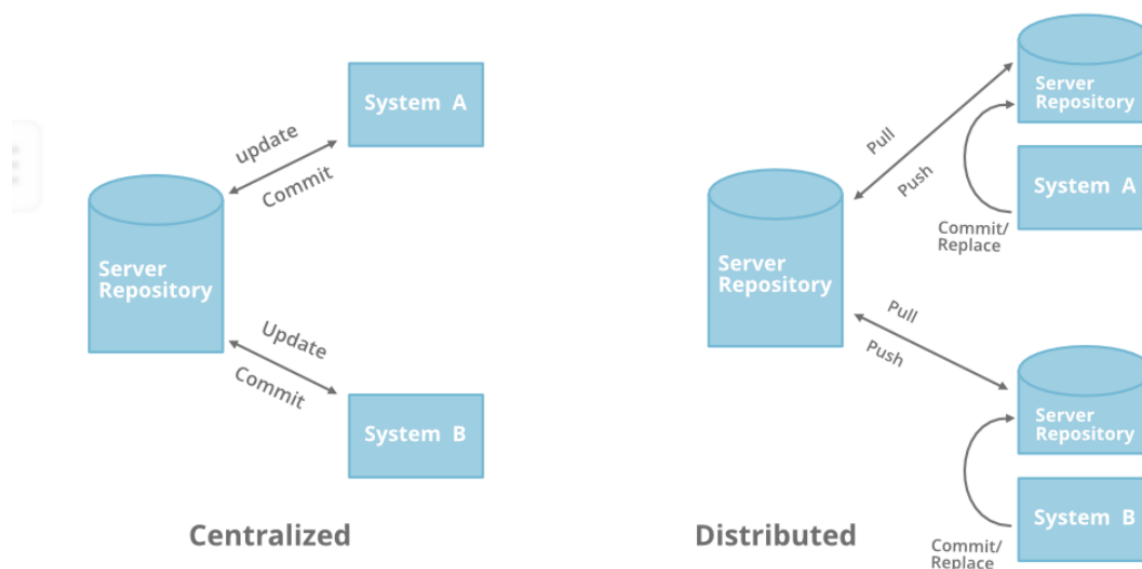
版本控制系统是一段时间内帮助跟踪代码中的更改的软件。当开发人员编辑代码时，版本控制系统会为文件拍摄快照。然后，它会永久保存该快照，以便以后可以根据需要重新调用它。

如果没有版本控制，开发人员会尝试在其计算机上保留多个代码副本。这很危险，因为很容易在错误的代码副本中更改或删除文件，可能会丢失工作。版本控制系统通过管理代码的所有版本来解决此问题，但一次向团队提供单个版本。

## 2. 版本控制软件的基本功能

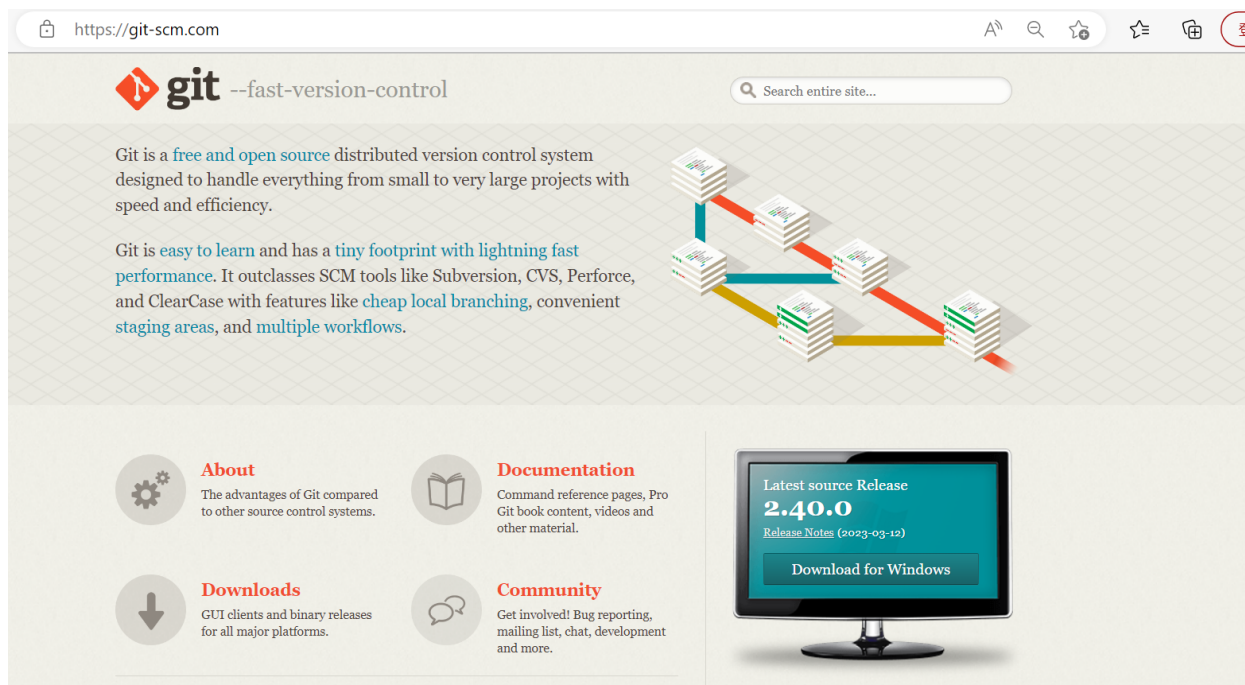
1. 保存和管理文件
2. 提供客户端工具进行访问
3. 提供不同版本文件的比对功能

## 3. 分布式与集中式版本控制

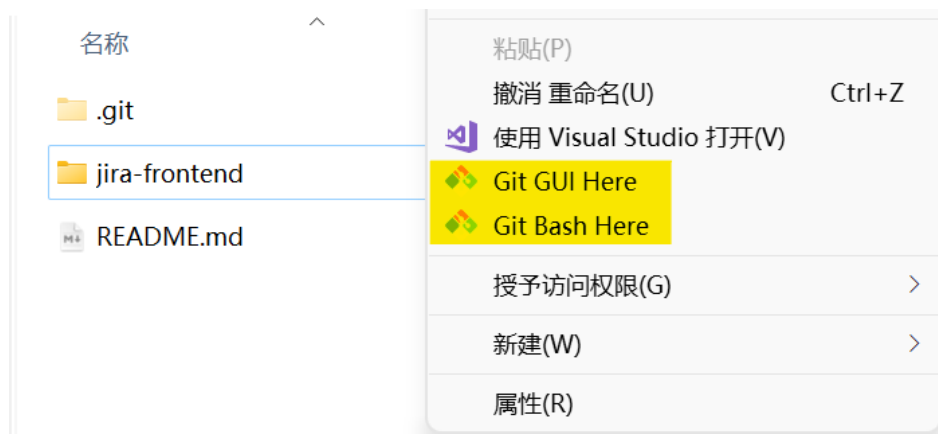


# 二、Git安装

## 1. [Git官网](#)



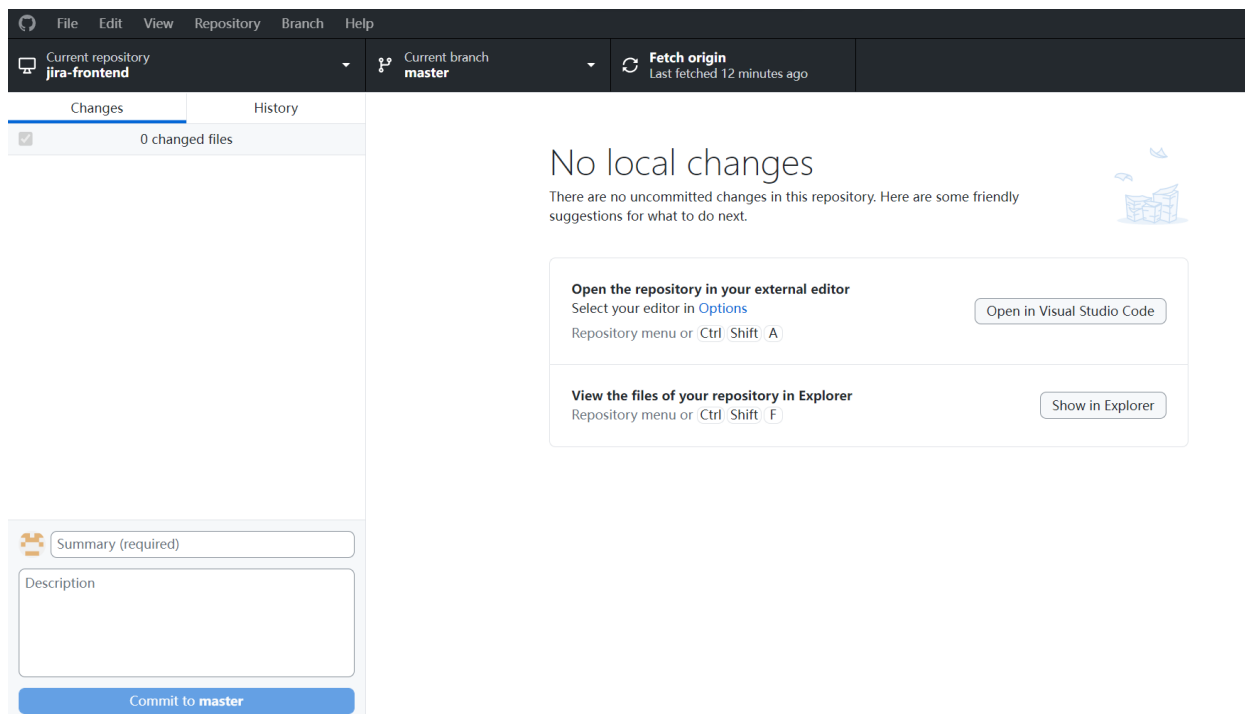
## 2. Git GUI Here和Git Bash Here



- **Git GUI Here** : Git的图形化操作入口
- **Git Bash Here** : Git的命令行形式入口

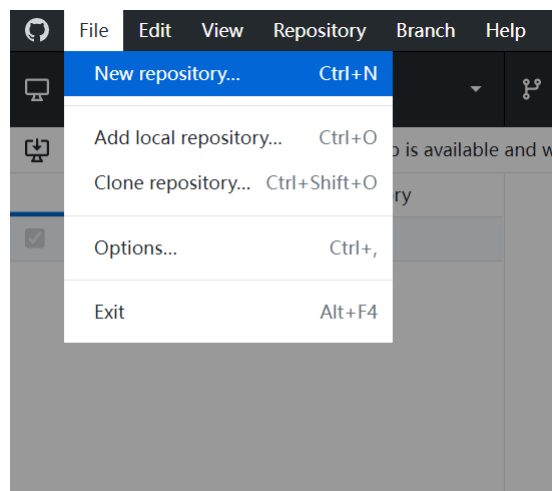
## 三、GitHub Desktop

### 1. 主页面



## 2. 创建仓库

### 2.1 File —> New repository



### 2.2 填写仓库名称，仓库描述，本地路径

Create a new repository

Name

testRepository

Description

test

Local path

C:\testSource

Choose...

☐ Initialize this repository with a README

Git ignore

None

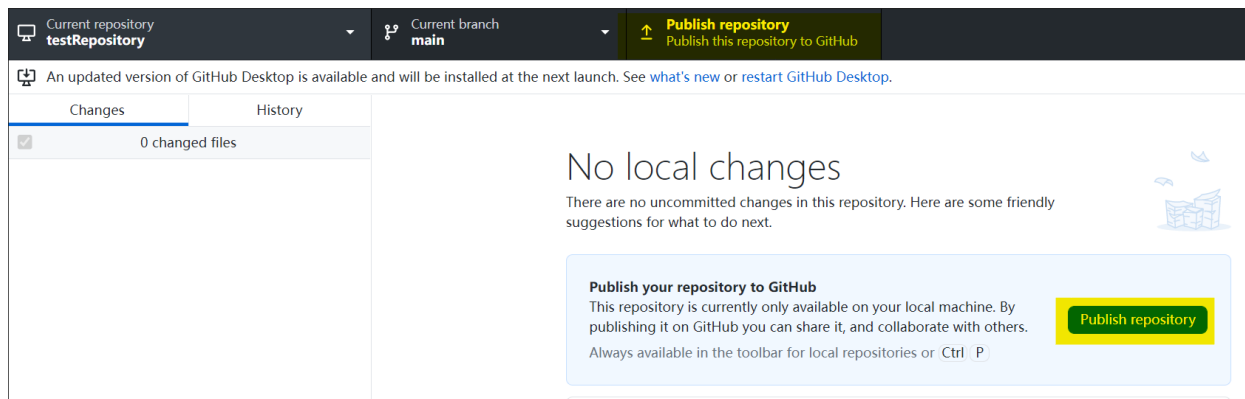
License

None

Create repository

Cancel

## 2.3 发布到远程仓库



## 2.4 输入远程仓库名称和描述，选择仓库是否为私有

Publish repository

GitHub.com

GitHub Enterprise

Name

testRepository

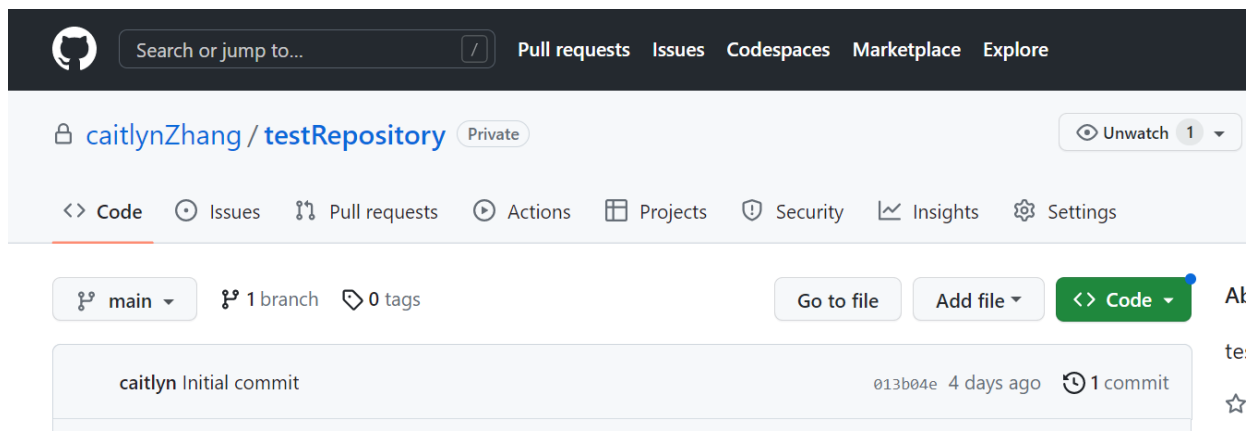
Description

☒ Keep this code private

Publish repository

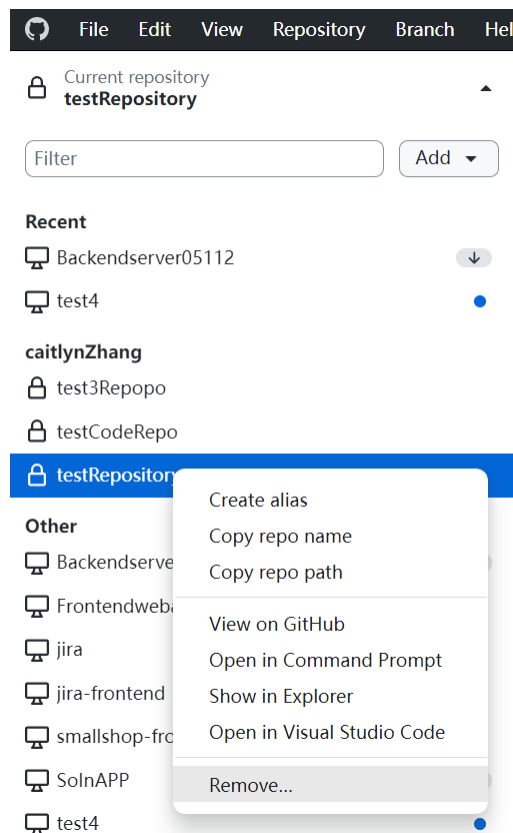
Cancel

## 2.5 成功发布到远程仓库

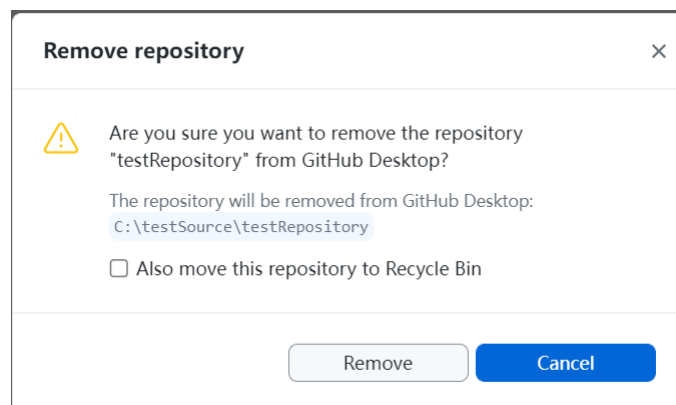


## 3. 移除仓库

### 3.1 选择仓库 —>右键remove

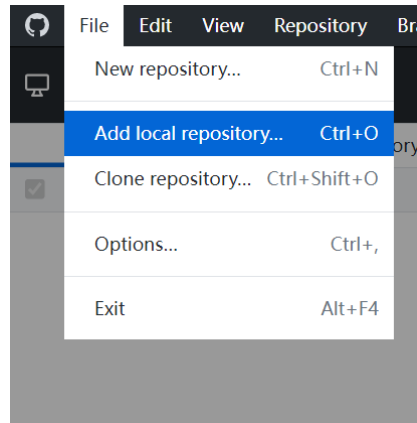


### 3.2 选择是否将仓库从本地删除

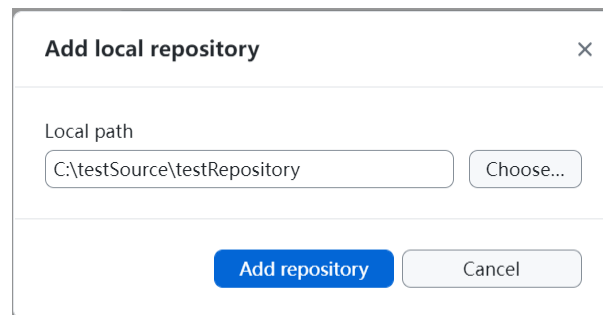


## 4. 添加仓库

### 4.1 File—> Add local repository

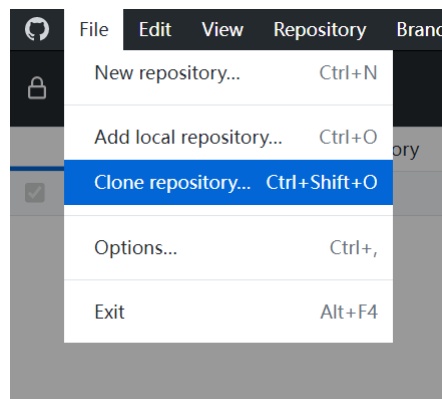


### 4.2 选择仓库路径



## \*\*5. 从远程仓库克隆代码

### 5.1 File—>Clone repository



### 5.2 输入远程仓库地址和本地存放路径

Clone a repository

GitHub.com

GitHub Enterprise

URL

Repository URL or GitHub username and repository  
( hubot/cool-repo )

http://zhangqi@49.232.68.165:8081/r/jira-frontend.git

Local path

C:\SourceCode\jira-frontend.git

Choose...

Clone

Cancel

## **\*\*6. Commit和Push**