# Git 使用说明

## 安装 Git

### Linux/Unix：

RH/CentOS: yum install git

Ubuntu: apt-get install git

Windows

Git for windows: <http://msysgit.github.io/>

安装一路下一步即可

## Git for windows 介绍

该章节主要为使用 windows 的用户, 其他用户请忽略

git for windows 安装完成后右键出现如下菜单

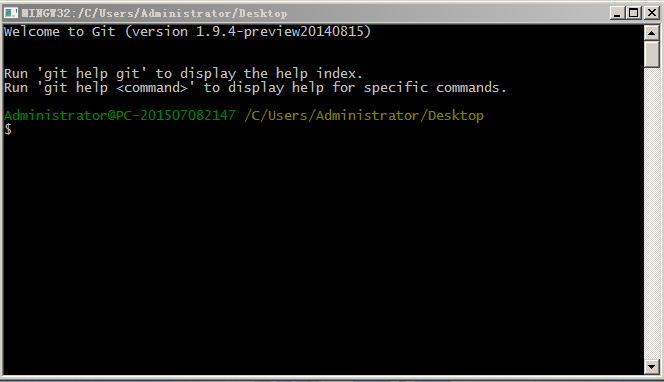


Git Init Here: 在此处初始化一个仓库

Git Gui: Git GUI 界面（不敢恭维）

Git Bash: Git Bash 界面 -- 主要操作界面

点击 Git Bash 开始操作



以下基本操作均在此窗口输入命令后回车执行

## Git 配置

### 用户信息配置

Git 使用必须配置以下两个 git 变量才能使用

user.name: 英文用户标示

user.email: 用户邮箱

打开 Git Bash

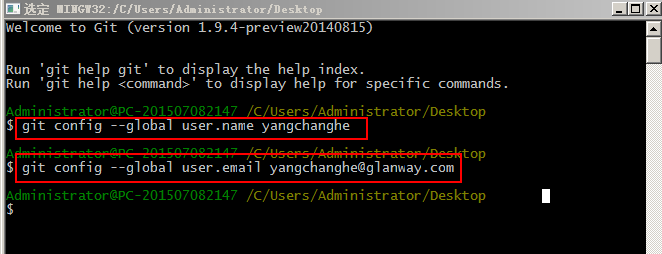
执行以下命令

git config --global user.name $username

git config --global user.email $email

其中 $username 和 $email 为 实际的用户标识和邮箱

eg:

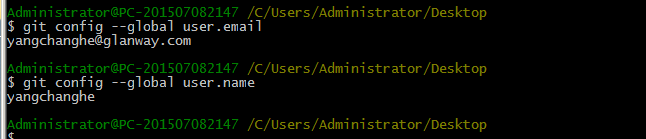


配置完成后可以通过

git config --global user.name

git config --global user.email

来进行查看确认



### 认证配置

1. 公钥认证配置

公钥认证允许通过生成的公钥私钥对，将公钥保存到服务器认证列表，通过持有私钥

来进行免密码登录

- 生成公钥

- 通过 ssh-keygen 生成

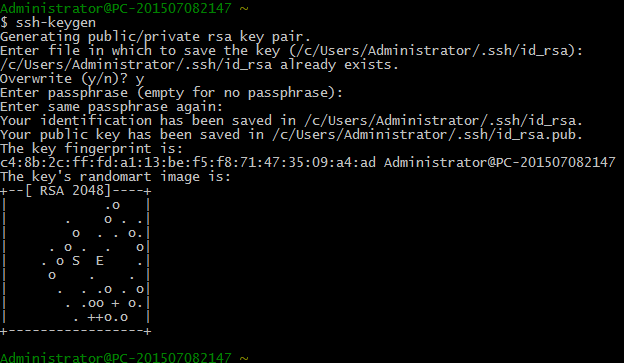
在 Git Bash 中输入 ssh-keygen 一路回车

（如果已经存在会询问是否覆盖，需要输入y) 生成后会有以下提示信息

此时生成的公钥和私钥在用户家目录下的.ssh 目录中

将公钥拷贝到服务器认证列表即可

对于 gitlab 请参考 Gitlab 章节



2. HTTP 认证配置

HTTP 认证需要借助 HTTP 服务器或应用， 有管理员开通即可

## Git 基本操作（可跳过）

### 创建本地仓库（可跳过）

git init 仓库名称来创建仓库

如果仅仅是作为仓库来使用， 可以加上 --bare 参数 即 git init --bare

### 从远程仓库克隆代码

git clone git地址

eg: git clone git@121.40.102.225:java/gone.git

注： 对于 gitlab 中git地址的获取请参考 Gitlab 相关章节

### 添加文件

在操作系统中创建文件后，该文件并没有被 git 进行管理

需要使用如下命令添加到git 管理中

git add file

eg: git add newfile

也可以使用通配符 git add \*.txt

### 删除文件

对于直接从操作系统删除的文件，也并没有在 git 中删除

因此需要使用

git rm file 来删除 eg: git rm README.md

### 提交修改

#### 提交修改

git commit -m '提交消息' 提交当前目录下所有文件

git commit -m '提交消息' file 提交 file 文件

git commit -m '提交消息' \*.txt 提交所有 .txt 后缀文件

#### 增补提交

对于提交以后，发现提交中部分文件遗漏，可以通过以下命令进行增补提交，该提交会将与给定版本提交进行合并，不会产生新的提交记录

git commit -C HEAD -a --amend

其中

-C 来指定要增补提交的版本, HEAD 表示最新版本

-a 表示提交所有文件

--amend 表示增补提交

### 查看状态

查看当前工作区状态 git status

查看提交日志 git log

查看引用日志 git reflog

### 撤销提交

#### 提交还原 - revert

git revert 版本: 用于将当前内容还原为给定版本的内容

eg: git revert 1jd82k1

git revert 后会自动提交，如果不希望自动提交可以使用 -n 参数

即: git revert 1jd82k1 -n

#### 复位/提交回退 reset

git reset [--hard] 版本： 用户回退提交

git reset 版本会将给定版本到当前的所有改变保存到暂存区后还原到给定版本

git reset --hard 称为“硬回退”， 直接将指针指向版本，不会保留当前版本

### 推送到远程分支

git push 用户将当前的提交推送到远程仓库

对于 git commit 只是提交代码到本地仓库，并没有推送到服务器

### 忽略文件

git 通过 .gitignore 来进行忽略控制

在仓库根目录创建 .gitignore 文件

其中要忽略的规则一行一条

eg:

.prolject

target/\*

## 从 Gitlab 上获取项目

### 账户开通

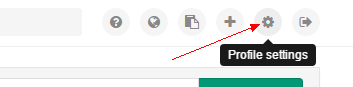
与管理员联系

### 公钥配置

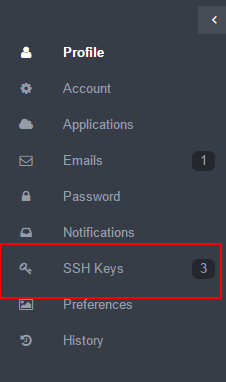
1. 生成 公钥私钥, 参考《git 配置 -- 认证配置》章节

2. 用记事本打开 id\_rsa.pub 拷贝

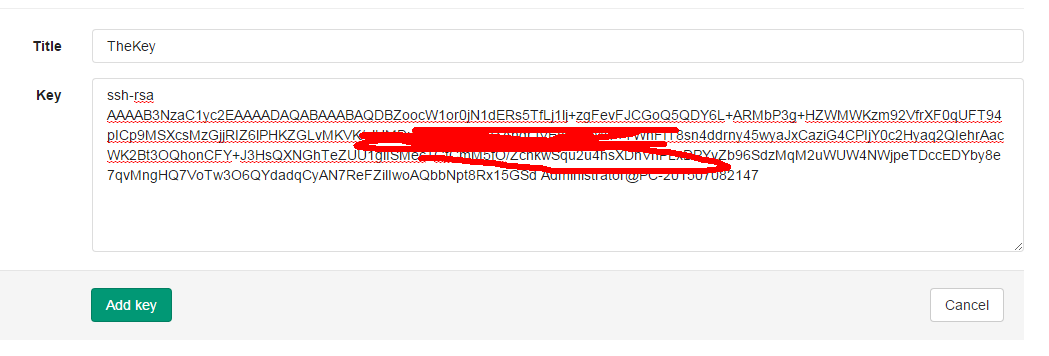
3. 登录 Gitlab 进入 Profile



4. 进入 SSH Keys 设置



5. 点击 “Add SSH Key" 填入 id\_rsa.pub 中内容， 标题随意

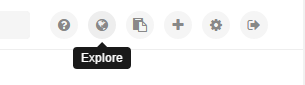


6. 点击 Add Key 即可

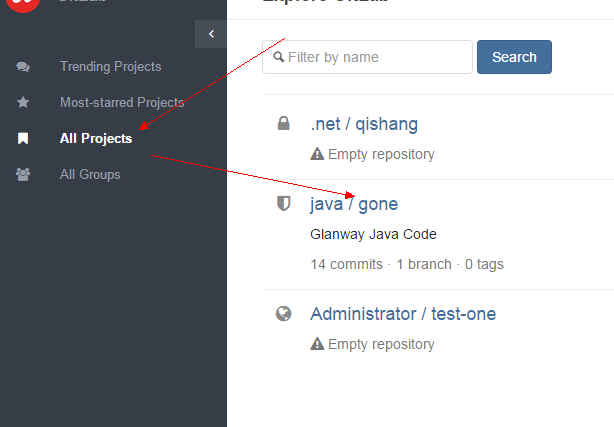
### 获取项目地址

1. 登录 gitlab

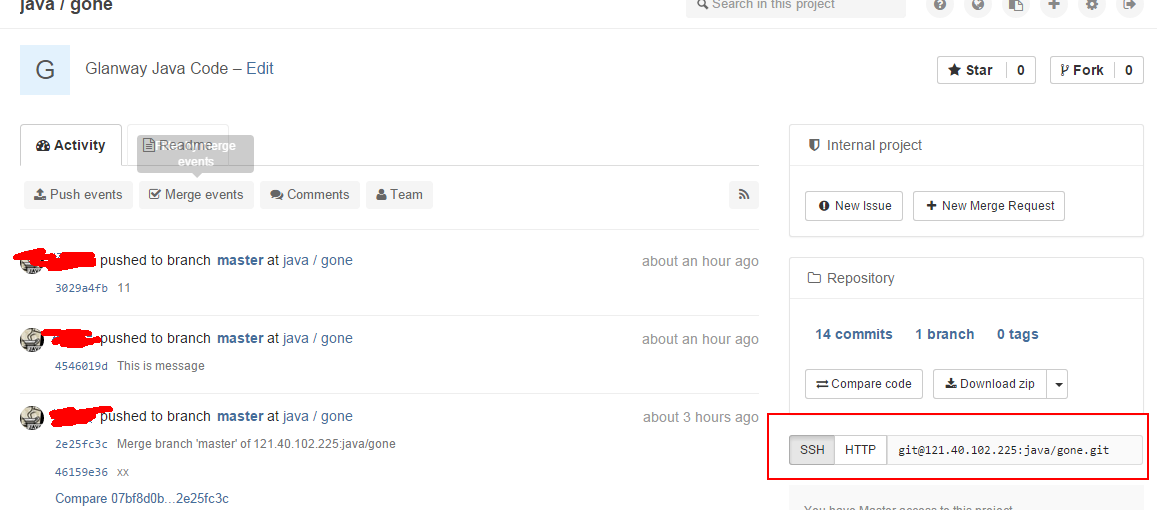
2. 进入 Explore



3. 进入 All Projects, 在右侧选择要查看的项目



4. 在最右侧查看 项目 git 地址



## 与 TortoiseGit 配合使用 (Windows)

因为 git 主要是命令操作，因此并不方便操作，且对状态没有图形化展示，因此可以与 TortoiseGit 配合使用

注： TortoiseGit 只是一个外壳，因此依然需要先安装Git for windows

### TortoiseGit 安装

1. 安装 Git for windows (参考安装Git 章节中 Git for windows 安装)

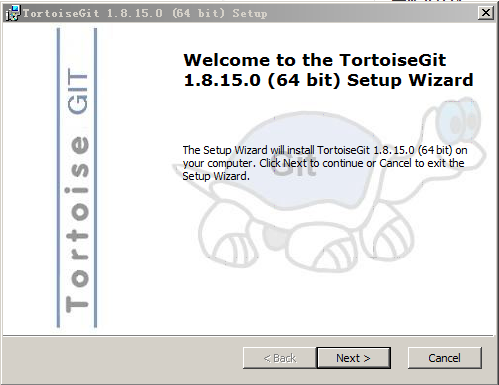
2. 下载 TortoiseGit

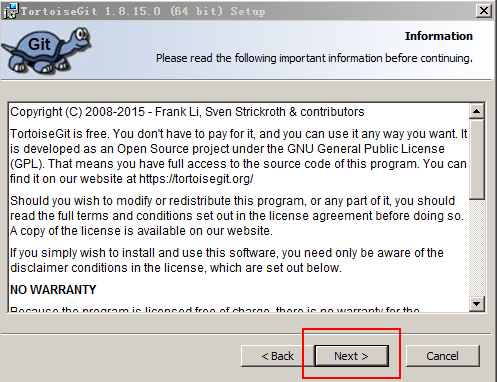
到以下地址下载与 git 版本对应的 TortoiseGit 版本

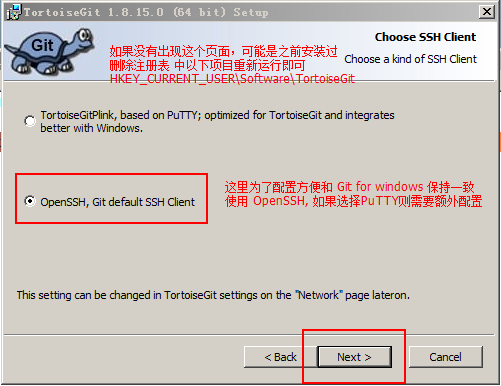
<http://download.tortoisegit.org/tgit/>

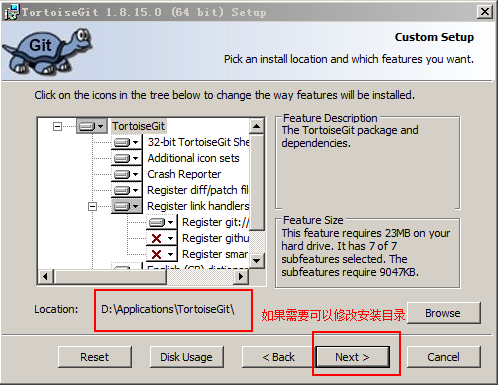
3. 安装 TortoiseGit

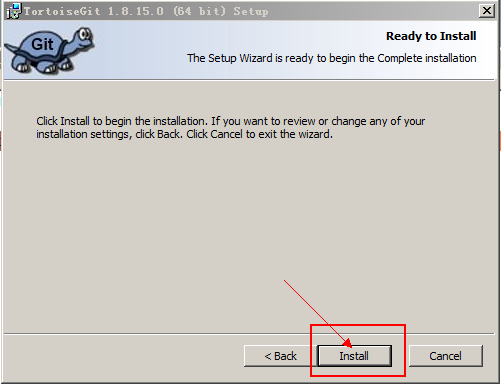
注意： 以下步骤如果有说明请按照说明操作

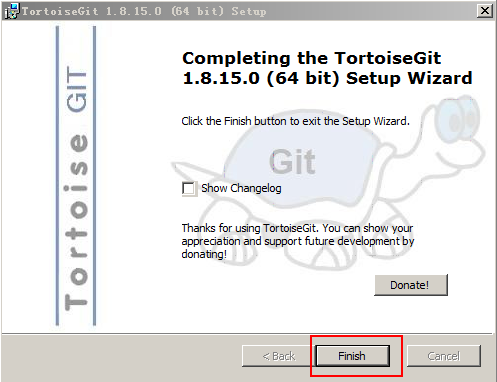












### TortoiseGit 使用

如果安装则右键菜单会出现以下菜单



#### 认证配置

一下两种认证方式可任选一种

1. 公钥认证(推荐, 后续如果未说明则均使用该方式)

公钥认证的好处是不用每次都输入账号密码

生成公钥 (如果使用 HTTP 协议可以忽略, HTTP 方式需要频繁输入账号密码）

2. HTTP 认证

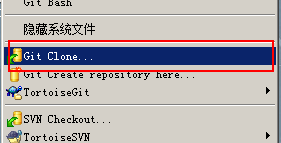
HTTP 认证的好处是，不用进行公钥认证的配置，但是如果工具没有记住密码功能

则需要每次都输入密码

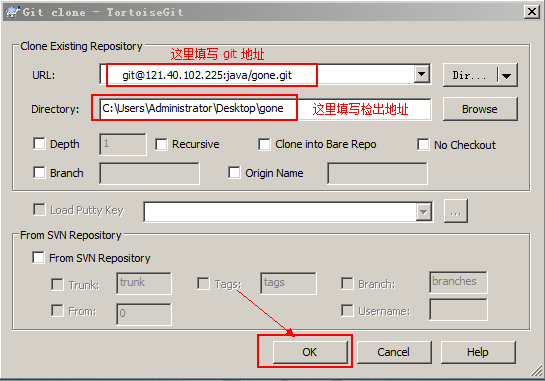
对于HTTP 账号申请即可

#### 从远程仓库克隆代码

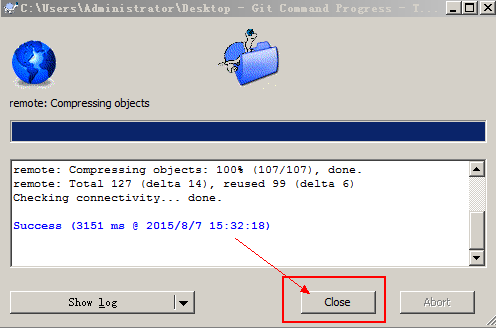
在右键，选择 Git Clone



填写 git 地址 （注意：公钥认证只在git协议有效 即git:// 开头地址)



检出完毕后，点击 “Close”即可



#### 添加文件到 Git

在文件夹新建文件， 右键"TortoiseGit" --> "Add" 即可

添加成功会看到“+” 图标, 标示已经添加，但未提交



添加成功后如下图



#### 编辑文件

直接编辑可以，编辑后 显示如下图标表示编辑后未提交



#### 提交修改及推送到服务器

1. 右键 --> "Git Commit --> master "

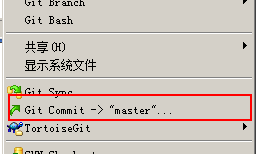
2. 在弹出窗口中选择要提交的修改以及填写提交消息即可

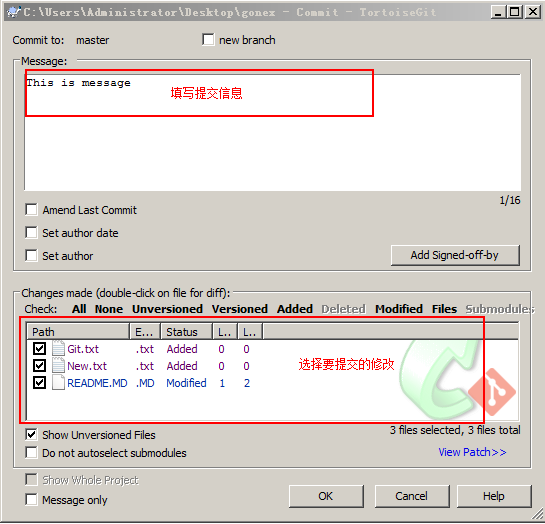
3. 在完成对话框中，

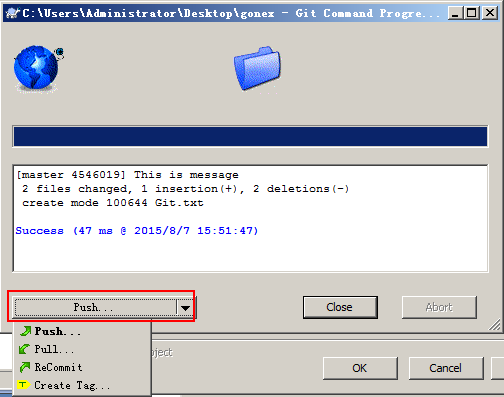
如果只是本地提交，点击关闭即可

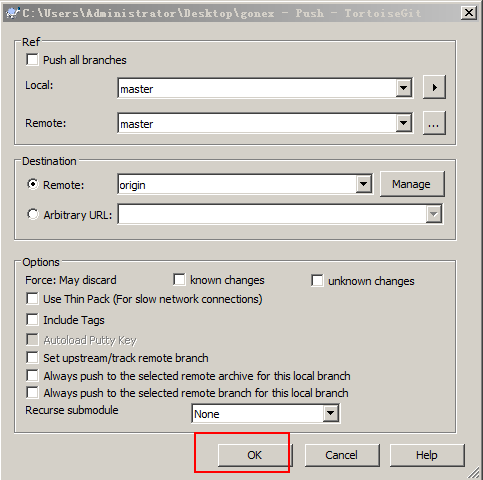
如果需要提交到服务器，则点击需要选择 Push 来推送到服务器

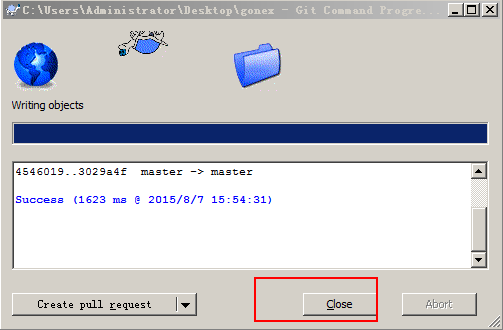
点击 Push 后在弹出窗口直接选择OK， 完成后关闭窗口即可











#### 获取更新

1. 在 git 管理的目录中，点击右键选择“Git Sync”

2. 在弹出窗口选择相应的拉取策略即可 Pull/Fetch 等

3. 拉取最新代码后会显示相关信息

