

Git 介绍

Git 是非常牛的版本控制系统，已经超越 CVS、SVN、TFS 等，是 Linux 创始人 Linus 为了改进 Linux 代码管理而开发的工具，已经成为最流行的版本控制系统，连微软都全面用 Git 管理 Windows 源代码了，VS2017 都内置 Git 作为默认源码管理工具。

Git 有什么好？分为本地、服务器两个版本，本地版本也可以做版本、做分支等，完全可以断网使用，在需要联网的时候再推送到服务器上，也可以把服务器上的版本同步到本地，不像 SVN、TFS 等对于服务器的依赖性那么强。

除非仅进行本地版本管理，否则都要使用服务器的。服务器端既可以使用 github.com（全免费、全公开）、gitlab.com（有收费也有免费）等网站提供的，也可以自己搭建的服务器。先讲使用 github.com，如果怎么样也访问不了，就可以试试其他国内 Git 服务：[码云](#)、[coding.net](#)、[gitcafe](#)、[code.csdn.net](#) 等。

github.com 服务的注册

- 1、<https://github.com/> 首先【Sign Up】注册，



Step 1:
Create personal account



Step 2:
Choose your plan

Create your personal account

Username

rupenggit



This will be your username — you can enter your organization's username next.

Email Address

1393695730@qq.com



You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

.....



Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

Create an account



Completed

Set up a personal account



Step 2:

Choose your plan



Step 3:

Tailor your experience

Choose your personal plan

- ☒ Unlimited public repositories for free.
- ☐ **Unlimited private repositories** for \$7/month.

Don't worry, you can cancel or upgrade at any time.

☐ **Help me set up an organization next**

Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.

[Learn more about organizations](#)

☐ **Send me updates on GitHub news, offers, and events**




Unsubscribe anytime in your email preferences. [Learn more](#)

Continue

Both plans include:

- ✓ Collaborative code review
- ✓ Issue tracking
- ✓ Open source community
- ✓ Unlimited public repositories
- ✓ Join any organization

You'll find endless opportunities to learn, code, and create, @rupenggit.

 Completed Set up a personal account	 Step 2: Choose your plan	 Step 3: Tailor your experience
---	--	--

How would you describe your level of programming experience?

- ☒ Very experienced ☐ Somewhat experienced ☐ Totally new to programming

What do you plan to use GitHub for? (check all that apply)

- ☒ Research ☐ Design ☐ Development
☒ School projects ☐ Project Management ☐ Other (please specify)

Which is closest to how you would describe yourself?

- ☐ I'm a hobbyist ☐ I'm a student ☒ I'm a professional
☐ Other (please specify)

What are you interested in?

e.g. tutorials, android, ruby, web-development, machine-learning, open-source

[Submit](#) [skip this step](#)

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#)

[Start a project](#)

然后到邮箱中点击激活邮件中的激活链接，否则报错

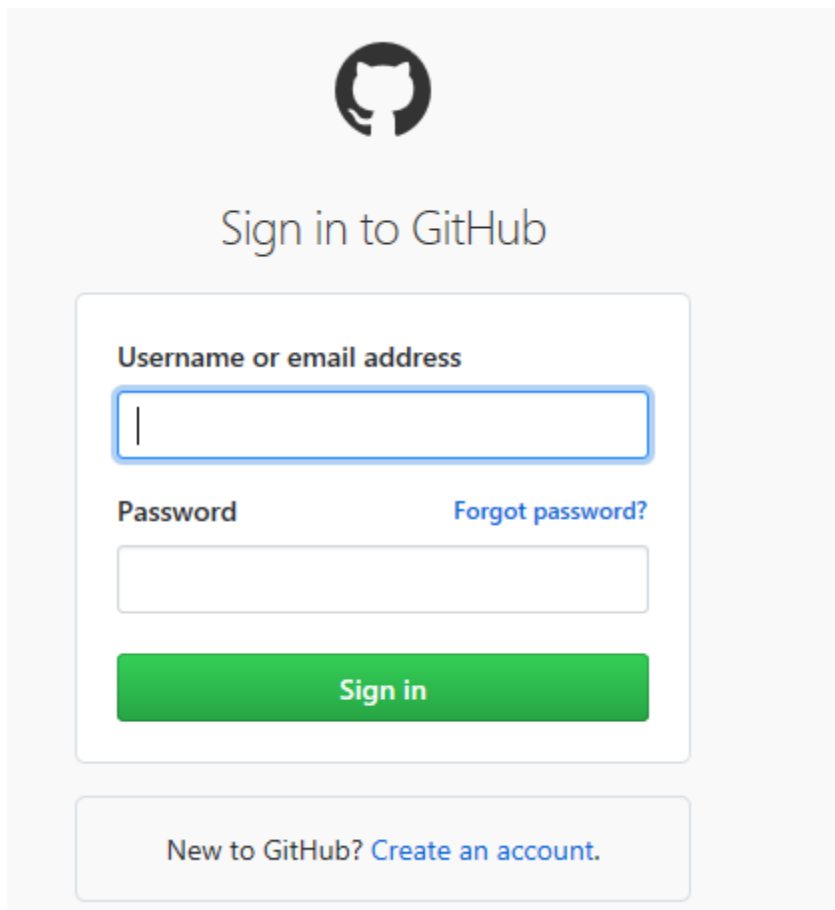


Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.
An email containing verification instructions was sent to **1393695730@qq.com**.

Didn't get the email? [Resend verification email](#) or [change your email settings](#).

点击激活链接之后需要登录才算完成激活



The image shows the GitHub sign-in interface. At the top is the GitHub logo (Octocat). Below it is the text "Sign in to GitHub". The main form has two input fields: "Username or email address" and "Password". To the right of the password field is a link "Forgot password?". Below the password field is a green "Sign in" button. At the bottom of the form is a link "New to GitHub? Create an account."

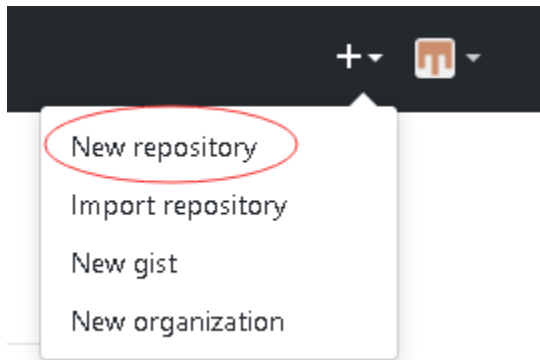
注册测试用：

用户名：rupenggit

邮箱：1393695730@qq.com


密码：yzk123456

2、新建存储库




可以建多个存储库，相关的项目放到一个存储库中。


Owner **Repository name**

 rupenggit / rupengdemo1 ✓

Great repository names are short and memorable. Need inspiration? How about **bookish-garbanzo**.

Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

ⓘ

Create repository

rupenggit / rupengdemo1

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. [Add topics](#) [Edit](#)

1 commit 1 branch 0 releases 1 contributor GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

rupenggit Initial commit Latest commit 30258df just now

.gitignore	Initial commit	just now
LICENSE	Initial commit	just now
README.md	Initial commit	just now

README.md

3、解释初始项目

- a) `.gitignore`: 不同语言的项目中都有一些不需要同步到代码库的文件, 比如 `bin/Debug`、`*.obj`、`*.suo`、`*.log`。`.gitignore` 文件制定哪些不需要同步。
- b) `ReadME.md`: 对项目进行描述的文件, 是 `MarkDown` 格式, 搞开源项目的写这个文件, 让别人知道你这个项目干啥的、怎么用, 自己的项目可以不管这个文件。

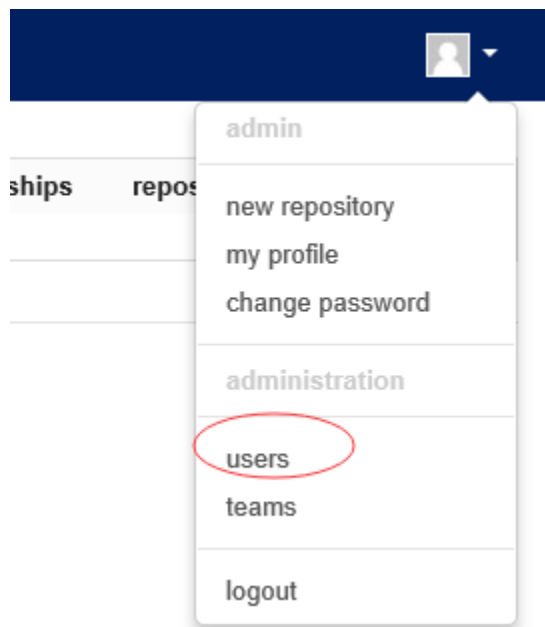
自己搭建 Git 服务器

如果公司要搭建自己的 Git 服务器, 可以使用 `gitblit` 这个开源的 Git 服务器

- 1、到 <http://gitblit.com/> 下载。
 - 2、确保电脑安装了 Java 运行时环境 JRE 或者 JDK (要求 ≥ 1.7) 安装文档 <http://www.cnblogs.com/android-joker/p/4513125.html> 主要注意不要端口冲突
- 不要把自己定位为 “.net 程序员”
不要把一切非微软技术视为敌人
我们是程序员
一切编程语言、技术都只是我们的工具!

微软都 Love Linux 了! 大清都亡了, 你还在反清复明!

- 3、我们是程序员!
- 4、默认管理员用户名、密码都是 `admin`, 登陆后改密码。一般不用 `admin` 来实际操作 `git`。
- 5、用户管理



新建一个普通用户 (yzk、123456)

+ new user

edit | delete

edit | delete

generalattributesaccess permissions

usernameyzk

password

confirm password

display name杨中科

email addressyzk@rupeng.com

can admin☐ can administer Gitblit server

can create☒ can create personal repositories 创建存储库的权限

can fork☐ can fork authorized repositories to personal repositories

exclude from federation☐ block federated Gitblit instances from pulling this account

disable user☐ prevent this account from authenticating

由管理员或者普通用户创建存储库

repositoriesfilestoreactivitysearchfilters

your own Git repositories.

+ new repository

descriptionownerlast change

project name
~yzk/ demo1 use '/' to group repositories. e.g.

description

Access Policy
Choose an access policy to control repository visibility and git permissions.

☐ Restrict Push (Authenticated)
Anyone can see and clone this repository. All authenticated users have RW+ push permission.

☐ Restrict Push (Named)
Anyone can see and clone this repository. You choose who can push.

☐ Restrict Clone & Push
Anyone can see this repository. You choose who can clone and push.

☒ Restrict View, Clone, & Push
You choose who can see, clone, and push to this repository.

☐ allow forks
allow authorized users to fork this repository

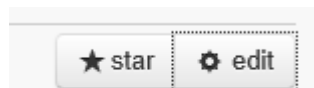
Initial Commit
This will allow you to git clone this repository immediately. Skip this step if you have already run git init locally.

☐ Include a README
This will generate a simple README document for your repository.

☒ Include a .gitignore file
This will insert a config file that instructs your Git clients to ignore files or directories that match defined patterns.

VisualStudio

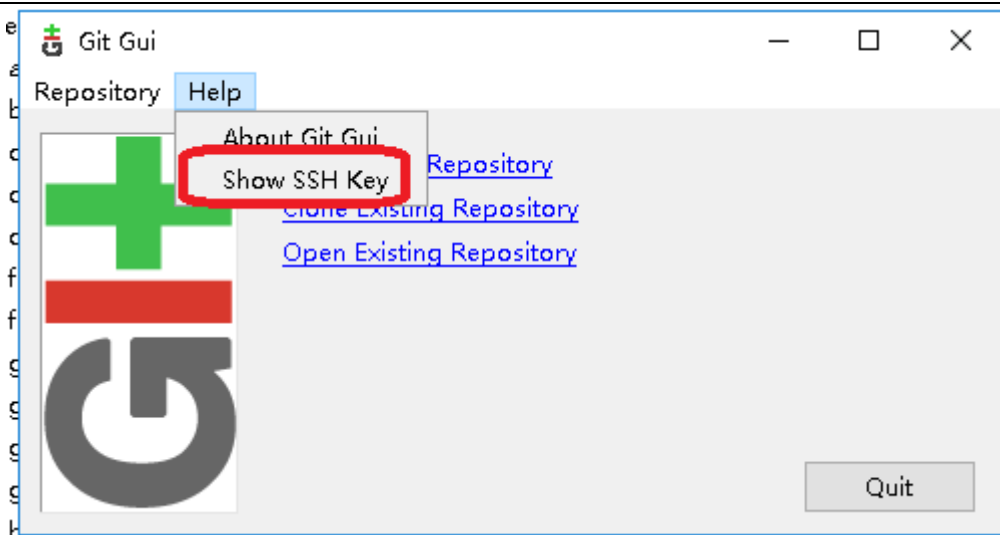
6、用新建的普通用户登录。在用户头像的 myprofile 的 repositories 中查看自己所有有权限



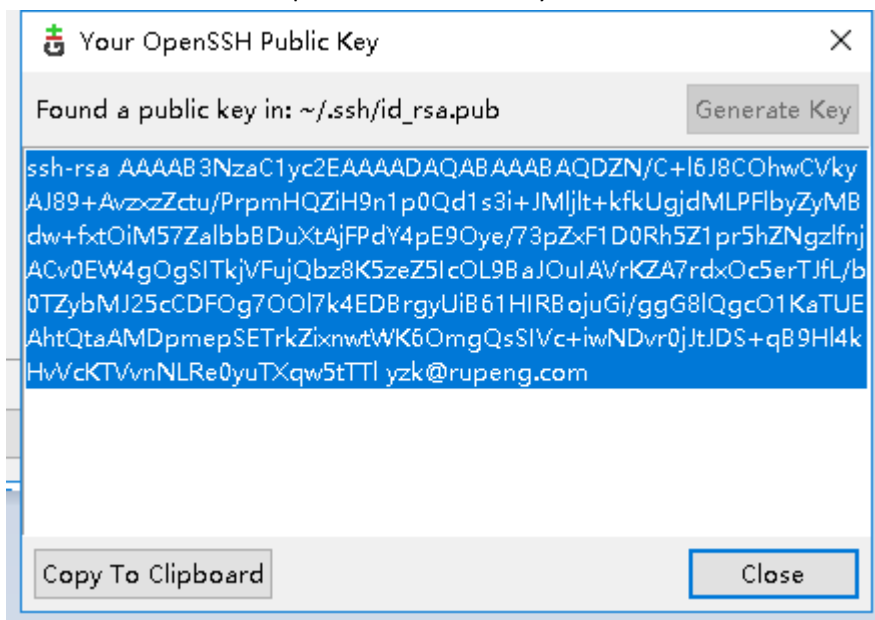
的存储库。在存储库的 edit 中可以编辑存储库的权限、哪些人能查看、提交、下载代码。

配置本地 Git 工具

- 1、首先安装 Git For Windows: <https://git-for-windows.github.io/> 一路 Next 安装
- 2、使用之前需要先将本地的 SSH 公钥配置到 Git 服务器上。查看 SSH 公钥的方法: 在任意文件夹下点击右键, 选择 GIT GUI here。在弹出的程序中:



选择主菜单的【Help】→【Show SSH Key】



把全部内容拷出来。

3、把 SSH Key 填到自己 Git 服务器用户的后台配置中。

Gitblit 在【头像】→【my profile】→【SSH Keys】，填上上一步的 SSHKey，然后【Add】

Github 在【头像】→【Settings】→【SSH and GPG Keys】→【New SSH Key】，填上上一步的 SSHKey，然后【Add SSH Key】

将代码提交到服务器

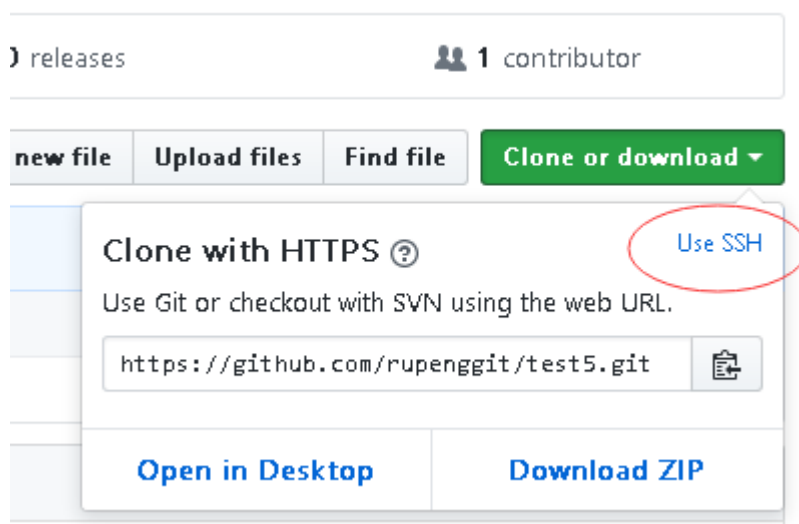
无论是新建一个存储库，将新建项目提交到服务器；还是下载已有的存储库，后续的操作都是一样的。新建存储库也需要先建服务器端存储库，再下载到本地（初始文件），再上传。

1、从 web 服务器页面上得到对应存储库的 giturl 路径

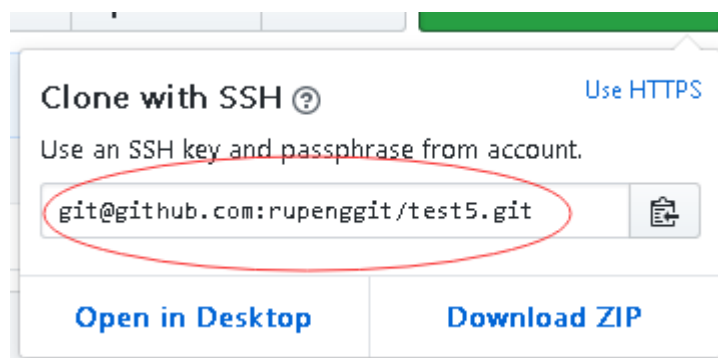
Gitblit 在存储库页面这个位置，一定要使用 ssh 版本



GitHub 在存储库页面这个位置



注意要 “UseSSH”



- 2、命令行进入项目根目录文件夹，执行“git init”创建 git 文件结构（隐藏的.git 文件夹。 .git 文件夹保存着历史版本信息以及 git 服务器的配置信息，删除之后就是一个干净的普通文件夹了）。
- 3、执行“git pull git 路径”把初始结构下载到本地（服务器端至少要有个文件，否则 pull 的时候会报错 fatal: Couldn't find remote ref HEAD）
- 4、执行“git add .”（注意最后的.）把修改提交到本地缓存
- 5、执行“git commit -m '提交的注释信息'”提交到本地库
- 6、执行“git remote add origin git 路径”把本地库设置同步到远程服务器
- 7、执行“git push --set-upstream git 路径 master”设定默认上传地址，并且进行第一次同步到服务器

8、下载一个旧项目也如此。

日常开发的操作

- 1、本地代码修改后，执行如下的指令：
 - a) `git add -A` 把本地所有的修改提交到缓存
 - b) `git commit -m'修改描述'` 把修改提交到本地
 - c) `git push` 把修改推送到服务器
- 2、获取服务器端的修改：`git pull`
- 3、`add` 的不同参数：
 - a) `git add -A` 提交所有变化
 - b) `git add -u` 提交被修改(modified)和被删除(deleted)文件，不包括新文件(new)
 - c) `git add .` 提交新文件(new)和被修改(modified)文件，不包括被删除(deleted)文件
- 4、在受 Git 管理的文件夹下点右键【Git Gui Here】使用 GUI 界面进行操作。

Visual Studio Git 插件使用

可以使用命令行进行全部 Git 操作，如果想使用图形界面，也可以使用 Visual Studio Git 插件。

- 1、VS2017 内置 Git 插件，其他版本自己装“Visual Studio Git 插件”
- 2、还是使用之前的命令行的方式来下载或者提交代码。初始化完成后，再使用 VS 图形界面。图形界面操作也分成“提交(commit)”和“推送(push)”两个操作，也可以使用“提交并且推送”这个二合一的操作。