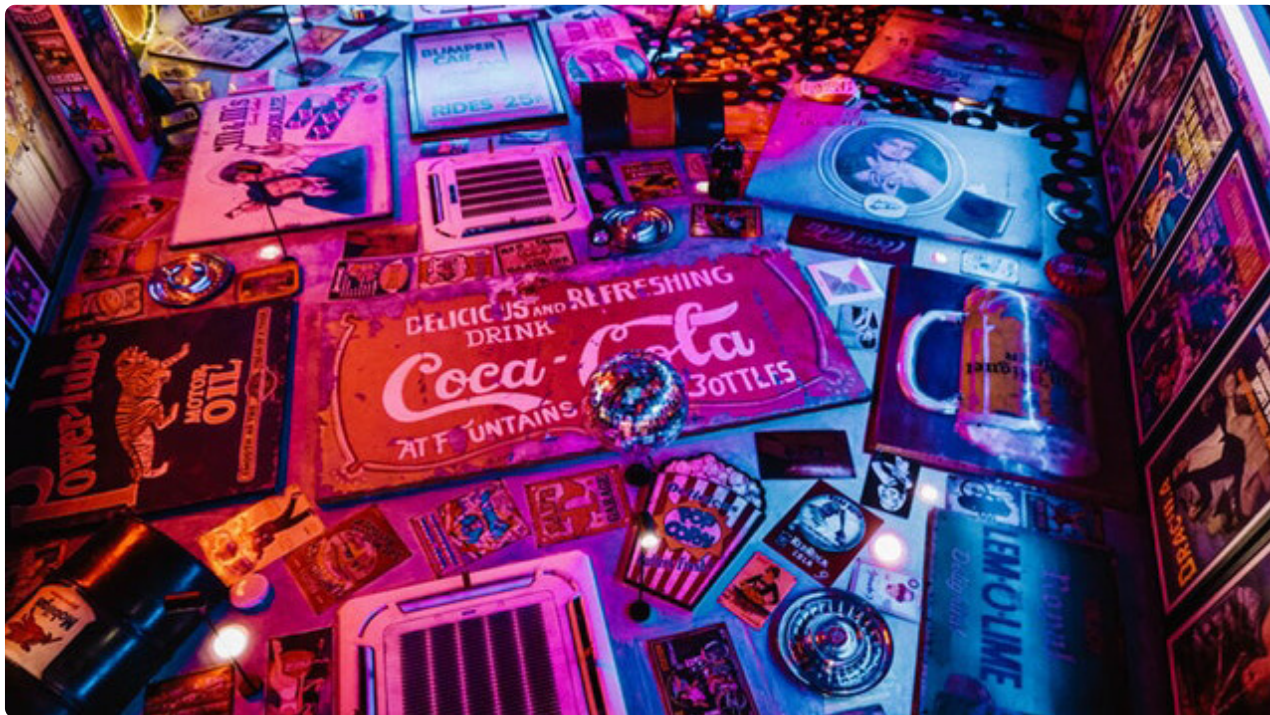


04 拉取远端代码：使用Git命令下载远程仓库到本地

更新时间：2019-09-23 19:08:40



“不要问你的国家能够为你做些什么，而要问你可以为国家做些什么。”

——林肯”

在上一节中，已经知道如何管理本地的代码了，在这一节中主要学习如何获取一个远程仓库的代码。为了方便大家的练习，我们会以国内的代码托管平台码云作为服务器为例；在拉取代码的过程中可能会碰到鉴权方面的问题，这一节内容中同样会提到如何处理。

4.1 创建远程代码仓库

我们首先在代码托管平台码云上新建一个自己的测试版本库：

4.1.1 注册账号

首先，去代码托管平台[码云](#)（用 [github](#) 也可以，考虑到大部分人的英文能力，建议大家先用码云入手）注册一个账号，然后在页面上登陆。

注册

已有帐号? [点此登录](#)

姓名

https://gitee.com/

个人空间地址

?

手机或邮箱

密码不少于6位

☐ 已阅读并同意 [使用条款](#) 以及 [非活跃帐号处理规范](#)

立即注册

4.1.2 创建仓库

登陆之后，可以看到右上角有一个 **创建仓库** 选项，我们点击创建仓库，如下图所示：



在创建仓库的表单中，简单填写一下信息即可，如下图所示：

新建仓库

仓库名称 

test201907

归属



daxia

路径

test201907

仓库地址: <https://gitee.com/songboy/test201907>

仓库介绍 非必填

用简短的语言来描述一下吧

是否开源



私有



公开

选择语言

请选择语言

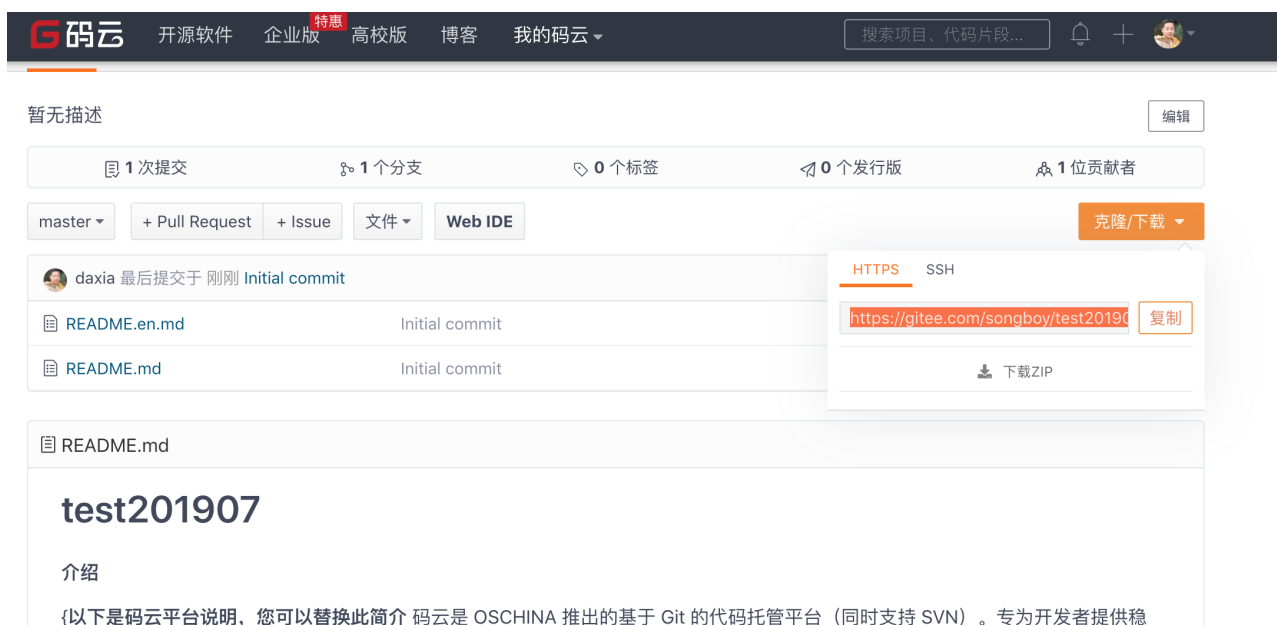
添加 .gitignore

请选择 .gitignore 模板

填写完表单之后，点击表单下方的提交按钮，表单提交上去之后，就会给你建立一个远程仓库。

4.1.3 进入仓库

创建完毕之后，浏览器会自动跳转到仓库的地址，如下图所示



码云 开源软件 企业版 高校版 博客 我的码云

搜索项目、代码片段...

暂无描述

1 次提交 1 个分支 0 个标签 0 个发行版 1 位贡献者

master + Pull Request + Issue 文件 Web IDE 克隆/下载

daxia 最后提交于 刚刚 Initial commit

| 文件 | 提交 |
|--------------|----------------|
| README.en.md | Initial commit |
| README.md | Initial commit |

README.md

test201907

介绍

(以下是码云平台说明,您可以替换此简介 码云是 OSCHINA 推出的基于 Git 的代码托管平台(同时支持 SVN)。专为开发者提供稳

建立远程仓库之后，默认里面有几个文件；接下来我们通过 `git` 的命令将远程仓库拉取到本地，一般会提供 `HTTP` 协议和 `SSH` 两种协议提供管理，两种协议所使用的鉴权方式不同，下面我们将两种方法都实践一次；

4.2 HTTP (S) 获取远程仓库

HTTP 协议方式拉取代码相对来说比较简单，直接执行 `git` 的 `clone` 命令即可，不需要额外的配置，但相对 `ssh` 协议来说安全性较低。

4.2.1 首次拉取

HTTP 协议首次拉取代码的命令格式如下所示：

```
git clone 版本库地址 [本地文件夹名称]
```

假设我需要把刚才新建的仓库代码拉取到本地，并且本地的文件夹名称叫 `httpstest` (也可以不指定本地文件夹名称，默认名字为远程仓库名字)，参考命令如下所示

```
git clone https://gitee.com/songboy/test201907.git httpstest
```

命令执行完成后，会要求你输入用户名和密码，只有当你输入正确的用户名和密码之后代码才能正常拉取。

```
❏ ~ git clone https://gitee.com/songboy/test201907.git httpstest
Cloning into 'httpstest'...
Username for 'https://gitee.com': 78778443@qq.com
Password for 'https://78778443@qq.com@gitee.com':
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (4/4), done.
```

4.2.2 更新代码

假设远程代码有变更，你想把本地代码更新时，可以在本地的版本库目录下通过 `git pull` 命令更新，不需要再指定远程地址，参考命令如下

```
git pull
```

默认情况下会再次提示你输入密码，因为 `git` 默认没有缓存 HTTP 认证权限，如下结果所示

```
❏ httpstest git:(master) git pull
Username for 'https://gitee.com': 78778443@qq.com
Password for 'https://78778443@qq.com@gitee.com':
Already up to date.
```

4.2.3 临时记住密码

如果你不想每次都输入 `git` 的认证信息，可以设置缓存认证数据，默认记住 15 分钟，如下命令所示：

```
git config --global credential.helper cache
```

如果你想缓存更长时间，也可以指定缓存时长，比如下面是自定义配置记住 1 小时的命令：

```
git config credential.helper 'cache --timeout=3600'
```

4.2.4 永久记住密码

如果你不想每次提交代码都要输入用户名密码，也可以让 `Git` 永久记住密码，参考命令如下所示

```
git config --global credential.helper store
```

命令执行完毕之后，会在当前用户主目录的 `.gitconfig` 文件中新增一项配置，配置如下所示

```
[credential]
  helper = store
```

在上面的命令中，如果没有 `--global`，则会在当前项目下的 `.git/config` 文件增加配置

从这里可以看出，`git` 永久记住密码其实是根据配置文件所决定，所以你也可以直接复制上面生成的配置到配置文件中。

4.3 SSH 拉取

现在我们再来看看 `SSH` 方式，相比 `HTTP(S)` 来说更加安全，因为使用的是非对称加密，采用公钥与私钥的方式，不过相对来说配置起来会麻烦一些；好处是一次配置之后，后续不需要每次都进行认证，也更加安全。

4.3.1 尝试拉取代码

`ssh` 方式首次拉取代码的命令没有什么变化，相对来说只是远程地址有变化，如下命令所示

```
git clone git@gitee.com:songboy/test201907.git sstest
```

回车执行后，会提示需要权限验证，返回信息如下所示：

```
❏ ~ git clone git@gitee.com:songboy/test201907.git sstest
Cloning into 'sstest'...
The authenticity of host 'gitee.com (218.11.0.86)' can't be established.
ECDSA key fingerprint is SHA256:FQGC9Kn/eye1W8icdBgrQp+KkGYoFgbVr17bmjey0Wc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'gitee.com,218.11.0.86' (ECDSA) to the list of known hosts.
git@gitee.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

因为并没有配置公钥与私钥，所以拉取代码并没有成功。

4.3.2 创建一个 `ssh key`

通过 `ssh` 协议拉取代码首先要保证当前用户的主目录存在一个 `.ssh` 的文件夹，并且里面已经存在私钥文件，如果没有的话我们可以通过 `ssh-keygen`，生成一份公钥与私钥，如下命令所示：

```
❏ ~ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:288AB3QWkp0hB5TOwzTCzkZTnLIO7IZ6YEgXpmXSLF0 root@93268ac888a1
The key's randomart image is:
+---[RSA 2048]-----+
|  + oE+BB*+  |
| o O =.O*o   |
| B = O.o     |
| o . + O.    |
| . o . =S..  |
| . o . +=    |
| . o ..o     |
| . . +       |
| . o         |
+---[SHA256]-----+
```

在执行命令的交互中，可以直接回车使用默认选项，最终会在当前用户目录下生成公钥于私钥，查看生成的公钥的命令为 `cat ~/.ssh/id_rsa.pub`，在返回的信息中可以看到类似如下信息：

```
❏ ~ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC65v2fz8/7N8/dw3sfjlkxav75MdLKLDtVtIs1XMj3PitPUXdUgsr7RR2WfpJUbtU6xGAxF9SpNFw275it
hvk85qx6PebQxfTTzqypawNwAOMy4CAOsRNybQWp//whtWfCUR2TvVtOQErq9ISEYhi+YQgoRg2ykYz9VZj8cFz99/Gtb3ApN3oHtgD9qcGUDPvL7MKjt
a3qrAX4KZHM++8FXz0qYrDgz9J/8+oLSebC6MOJiPuc7ut0rfICKaAU7XS4xvU39sNtES/j531AB/Xixb/ufaMPUKhldASmUFP1WFoVU4268mtW1dZ99t6As
dQ9X2wjNI1QAVX/IJQe2Ox root@93268ac888a1
```

4.3.3 添加公钥到服务器

当确认公钥于私钥生成完毕之后，我们还需要将公钥放到远程的 `git` 仓库中去，在码云的版本库中，右上角有一个管理，在管理页面的左侧菜单中有一个添加公钥的选项，我们将上面的公钥内容复制进去，如下图所示

🔒 daxia / test201907

👁 Unwatch 1 🍴 Fork 0

</> 代码

📄 Issues 0

🔗 Pull Requests 0

📎 附件 0

📖 Wiki 0

📊 统计

∞ DevOps ▾

🛠 服务 ▾

👤 管理

▶ 仓库设置

▶ 仓库成员管理

▼ 部署公钥管理

⚙ 公钥管理

➕ 添加公钥

集成与服务

WebHooks

只读文件管理 New

添加部署公钥

如果您想要对仓库进行写操作，请 [添加个人公钥](#)

标题

root@93268ac888a1

在这里粘贴公钥，查看 [怎样生成公钥](#)

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAQC65v2fz8/7N8/dw3sfjlkxav75MdLKLDtVtIs1XMj3PitPUXdUgsr7RR2WfpJUbtU6xGAxF9SpNFw275ithvk85qx6PebQxfTTzqypawNwAOMy4CAOsRNybQWp//whtWfCUR2TvVtOQErq9ISEYhi+YQgoRg2ykYz9VZj8cFz99/Gtb3ApN3oHtgD9qcGUDPvL7MKjta3qrAX4KZHM++8FXz0qYrDgz9J/8+oLSebC6MOJiPuc7ut0rfICKaAU7XS4xvU39sNtES/j531AB/Xixb/ufaMPUKhldASmUFP1WFoVU4268mtW1dZ99t6AsdQ9X2wjNI1QAVX/IJQe2Ox root@93268ac888a1

添加

4.3.4 拉取代码

当公钥添加进去之后，就已经完成了权限配置，此时我们再次使用 `ssh` 方式拉取代码，就不会提示没有权限，执行结果如下所示

```
❏ ~ git clone git@gitee.com:songboy/test201907.git sshtest
Cloning into 'sshtest'...
The authenticity of host 'gitee.com (218.11.0.86)' can't be established.
ECDSA key fingerprint is SHA256:FQGC9Kn/eye1W8icdBgrQp+KkGYoFgbVr17bmjey0Wc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'gitee.com,218.11.0.86' (ECDSA) to the list of known hosts.
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (4/4), done.
```

可以看到在代码执行之后，代码已经拉取完成。

4.3.5 更新代码

ssh 方式更新代码命令和上面的 http 方式拉取代码命令一致，同样需要在 sshtest 目录下执行命令：`git pull`，然后可以看到 git 成功的拉取到了代码

```
❏ sshtest git:(master) git pull
Already up to date.
```

4.4 小结

在这一节中主要讲解了如何使用码云创建一个远程仓库，以及两种协议拉取代码中的鉴权解决方法：

1. Git 远程交互通常同时支持 `HTTP(S)` 和 `SSH` 协议访问
2. `HTTP(S)` 协议交互默认每次需要输入账号密码，但可以通过缓存认证方式处理
3. `SSH` 协议需要将生成的公钥放到 Git 服务器当中去，配置之后 Git 会自动通过 ssh 协议进行鉴权，不需要通过账号加密码

}