



# 软件开发生产线 CodeArts

## 代码托管常见问题

天翼云科技有限公司

---

# 目 录

---

<b>1 仓库管理</b> .....	<b>4</b>
1.1 在一台电脑上，如何配置多个 SSH Key? .....	4
1.2 如何防止软件代码被他人 Copy? .....	5
1.3 同一用户创建的多个仓库都需设置 SSH 密钥吗? .....	6
1.4 所有用户是否可以使用同一个 SSH 密钥上传下载代码? .....	6
1.5 移除项目成员是否同时将其从代码仓库删除并删除其所创建的 SSH 密钥? .....	6
1.6 代码仓库可以相互转换“私有”或者“公开”吗? .....	6
<b>2 仓库使用</b> .....	<b>7</b>
2.1 是否要同时设置好 SSH/HTTPS 密码才能上传下载代码? .....	7
2.2 如何保证 SSH 密钥的安全? .....	7
2.3 是否支持对每个分支设置 SSH 密钥? .....	7
<b>3 仓库迁移</b> .....	<b>9</b>
3.1 是否支持批量下载多个仓库? .....	9
<b>4 通用问题</b> .....	<b>10</b>
4.1 项目成员为何看不到他人创建的代码仓库? .....	10
4.2 代码仓库对文件大小是否有限制? .....	10
4.3 成员提交的代码行数支持导出吗? .....	11
4.4 如何阻止涉密文件被推送到代码仓库? .....	11
4.5 为什么没有“同步仓库”功能页签? .....	11
4.6 删除 Git 分支的创建者，相关的 Git 代码分支也会自动删除吗? .....	12
<b>5 Git 相关问题</b> .....	<b>13</b>
5.1 Git 如何判断是代码仓库管理员? .....	13
5.2 如何获取代码下载的存储路径? .....	13
5.3 怎样获取代码仓库地址? .....	14
5.4 复制的仓库地址用在哪些场景? .....	15
5.5 代码托管服务是否支持 SVN? .....	15
5.6 向代码仓库中上传压缩包能否在线解压? .....	15
<b>6 常见报错解决方法</b> .....	<b>17</b>
6.1 私钥丢失.....	18

6.2 此密钥已存在，请重新生成密钥 .....	19
6.3 使用 SSH 协议克隆代码时一直提示输入密码 .....	20
6.4 使用 HTTPS 方式克隆代码时，输入登录密码报错 .....	20
6.5 提交代码之后，仓库找不到文件 .....	21
6.6 pathspec XXX did not match any files .....	21
6.7 Transport Error: cannot get remote repository refs. XXX.git: cannot open git-upload-pack .....	22
6.8 syntax error near unexpected token `newline' .....	23
6.9 unable to auto-detect email address .....	23
6.10 fatal:Authentication failed .....	24
6.11 'origin' does not appear to be a git repository .....	24
6.12 You are not allowed to push code to protected branches on this project .....	25
6.13 Not a git repository .....	25
6.14 src refspec master does not match any .....	26
6.15 destination path 'XXX' already exists and is not an empty directory .....	26
6.16 The requested URL returned error: 401 .....	27
6.17 向代码仓库推送代码失败 .....	28
6.18 使用 git pull 拉取代码失败 .....	28
6.19 fatal: refusing to merge unrelated histories .....	29
6.20 SSL certificate problem .....	30

---

# 1 仓库管理

---

在一台电脑上，如何配置多个 SSH Key?

如何防止软件代码被他人 Copy?

同一用户创建的多个仓库都需设置 SSH 密钥吗?

所有用户是否可以使用同一个 SSH 密钥上传下载代码?

移除项目成员是否同时将其从代码仓库删除并删除其所创建的 SSH 密钥?

代码仓库可以相互转换“私有”或者“公开”吗?

## 1.1 在一台电脑上，如何配置多个 SSH Key?

### 场景描述

开发人员通常只会生成一个 SSH Key，名字叫 `id_rsa`，然后提交到多个不同的网站。

但是也存在另一种需要，在同一个网站上，注册了两个用户名，通常网站不会允许为这两个用户名，配置同一个 SSH Key，这时候就会有些麻烦。

### 操作步骤

步骤 1 在本地 Git 仓库生成两个不同的 SSH Key。

```
ssh-keygen -t rsa -C "email"
Generating public/private rsa key pair.
Enter file in which to save the key (~/.ssh/id_rsa):<不要直接回车，填写自己定义的名字>
Enter passphrase(empty for no passphrase):<不要直接回车，填写自己定义的密码>
```

#### 说明

这是第一个关键，如果要生成 2 个 Key，这里写成：`github_1` 和 `github_2` 这样，就生成了 2 个证书。

步骤 2 用不同的帐号，上传两个不同的证书。

读取 `*.pub` 的内容，粘贴到服务网站上。记住对应的用户名。

---

### 步骤 3 编辑~/.ssh/config 文件。

```
Host dc_1
HostName *****.com
IdentityFile ~/.ssh/dc_1
PreferredAuthentications publickey
User username1
Host dc_2
HostName *****.com
IdentityFile ~/.ssh/dc_2
PreferredAuthentications publickey
User username2
```

要点在于 Host 与 HostName 的区别：

- **HostName:** 是填写真实的服务地址。
- **Host:** 是填写别名，后面会用上。
- **IdentityFile:** 填写的是证书的所在位置，你也可以把证书保存在任何地方。

### 步骤 4 读写代码。

原本在 Web 页面上复制的 SSH URL，可以直接使用，例如：

```
git@*****.com:name/repo.git
```

但是，现在根据你的帐号不同，需要自行替换：

```
git@dc_1:name/repo.git 或 git@dc_2:name/repo.git
```

----结束

## 1.2 如何防止软件代码被他人 Copy?

### 问题现象

- 软件开发的各种行为，在云上进行。
- 随时随地，都能够方便的接入云。

### 原因分析

- 在 CodeArts 上写程序的常见流程：
  - a. 在 CodeArts 上，创建一个代码仓库，并确保是最新版本。
  - b. 所有程序员都可以将代码下载到本地，在本地开发、调试完成后，提交代码到服务器。
  - c. 通过 Code Review 过程，将程序员的代码，合入到主干。
- 令人担忧的情况：

程序员总要在本地保留一份代码，我们无法防止，这个程序员笔记本里的代码，被他 Copy 出去，或者打包以后上传到某个网盘上。

原因：那是他自己的电脑，在他自己的机器上，他的操作很难防范。

---

## 处理方法

- 封掉 USB 接口、封掉蓝牙接口。
- 在电脑里加装特殊的软件，监控并限制各种上传的行为。
  - 监控特定文件的上传：如何防止改名、压缩、混淆后的文件。
  - 监控上传到特定网站：如何防止上传到未知的网络服务、邮件地址、自建服务器。
  - 监控所有的电脑操作并记录：只能在代码泄露之后，作为起诉的证据之一，很难在事前防范。

### 1.3 同一用户创建的多个仓库都需设置 SSH 密钥吗？

不需要。

SSH 密钥在电脑和某个帐号下的代码托管服务之间建立安全连接，在一台电脑上配置了 SSH 密钥并添加公钥到代码托管服务中后，所有该帐号下的代码仓库与这台电脑之间都可以使用该密钥进行连接。

### 1.4 所有用户是否可以使用同一个 SSH 密钥上传下载代码？

不可以。

SSH 密钥在电脑和代码托管服务之间建立安全连接，不同的用户通常使用不同的电脑，在使用 SSH 方式连接代码仓库前需要在自己电脑配置各自的 SSH 密钥。

### 1.5 移除项目成员是否同时将其从代码仓库删除并删除其所创建的 SSH 密钥？

成员会从代码仓库中删除，但密钥不会。

当用户被管理员从项目中移除后，会自动将该用户从项目下的代码仓库中移除，但 SSH 密钥不会被删除。此时用户将没有访问该项目和代码的权限，不能再访问项目和代码仓库。

### 1.6 代码仓库可以相互转换“私有”或者“公开”吗？

可以。

进入代码仓库详情页面，在“设置”页签中对“基本设置>仓库信息>可见范围”进行设置。

---

# 2 仓库使用

---

是否要同时设置好 SSH/HTTPS 密码才能上传下载代码？

如何保证 SSH 密钥的安全？

是否支持对每个分支设置 SSH 密钥？

## 2.1 是否要同时设置好 SSH/HTTPS 密码才能上传下载代码？

不是。

SSH 和 HTTPS 是使用 Git 进行代码版本管理的两种连接方式，使用其中任何一种方式都可以进行代码的上传下载，密钥（密码）的设置根据您选择的连接方式设定即可：

- 若选择 SSH 方式，需要生成密钥并配置到代码托管中。
- 若选择 HTTPS 方式，可通过代码托管生成密码，获取并使用即可。

## 2.2 如何保证 SSH 密钥的安全？

1. 在生成 SSH 密钥时，可以为密钥设置密码以保证安全。
2. 在代码托管中，SSH 密钥只能对应一个用户名，且只对该用户可见。

## 2.3 是否支持对每个分支设置 SSH 密钥？

不支持。

代码仓库并不支持对每个分支添加 SSH 密钥。

SSH 密钥是一个全局的配置，是代码仓库识别用户的方式，所以不支持对于某一个项目或者某一个分支进行密钥设置。

---

例如，项目中有 5 个成员，每个人拉取一个分支进行开发，目前是不可以对每个分支设置 SSH 密钥来限制分支操作权限。

---

# 3 仓库迁移

---

是否支持批量下载多个仓库？

## 3.1 是否支持批量下载多个仓库？

不支持。

代码托管暂不支持批量下载或上传多个代码仓库，需要对每个代码仓库逐一操作。管理员要对本地仓库做备份，可以自行通过 Shell 或者批处理命令实现多个仓库下载。

---

# 4 通用问题

---

项目成员为何看不到他人创建的代码仓库？

代码仓库对文件大小是否有限制？

成员提交的代码行数支持导出吗？

如何阻止涉密文件被推送到代码仓库？

为什么没有“同步仓库”功能页签？

删除 Git 分支的创建者，相关的 Git 代码分支也会自动删除吗？

## 4.1 项目成员为何看不到他人创建的代码仓库？

为了保证代码的安全，代码托管服务中添加了一层仓库成员的管理，需要将项目成员设置为代码仓库成员，才能够看到仓库内容。

进入代码仓库“成员”页面，单击“添加成员”可完成代码仓库成员的添加。

## 4.2 代码仓库对文件大小是否有限制？

有，具体如下：

- 控制台上传代码时，单个文件不能超过 10M，否则会上传失败；（客户端支持上传 10M 以上的文件）
- 在线修改代码时，单次保存行数不能超过 5000 行，否则会修改失败；（但是客户端上传不限制行数）
- 每个代码仓库的容量上限是 2GB。
- 本地单文件推送大小限制是 200M，超出时建议使用 SSH 协议。

---

## 4.3 成员提交的代码行数支持导出吗？

不支持。

目前没有导出功能，可以在代码仓库的“仓库统计”页面中进行查询。

## 4.4 如何阻止涉密文件被推送到代码仓库？

### 背景信息

- 当用户需要避免上传涉密信息的时候，可以选择在仓库设置中设置提交规则，勾选拒绝包含秘密的提交选项。
- 通过选择复选框防止向代码仓库提交涉密的文件，当文件名与正则表达式匹配时，代码托管服务会阻止用户推送。

---

#### 须知

此推送规则不会限制已提交的文件。

---

以下列出代码托管服务目前会阻止的文件名称与正则表达式：

正则表达式	文件名例子
<code>(ssh config)\(personal server\)_\(rsa dsa ed\d+ ecdsa\)</code>	ssh_server_rsa
<code>_rsa\$</code>	id_rsa
<code>_dsa\$</code>	id_dsa
<code>_ed25519\$</code>	id_ed25519
<code>_ecdsa\$</code>	id_ecdsa
<code>\.(pem key)\$</code>	secret.key privatekey.pem
<code>"[. _]history\$"</code>	.bash_history

## 4.5 为什么没有“同步仓库”功能页签？

**Q:** 为什么我的仓库中“仓库管理”里，没有“同步仓库”这个页签？

---

### 同步仓库

源仓地址: <https://github.com:xxxxx.git>

源分支: 全部分支

同步分支: 默认分支

点击开始手动同步代码仓库。警告: 同步远端的仓库, 可能会覆盖本地已经提交的代码, 导致代码丢失!

定时同步镜像仓:

打开定时同步功能时, 代码仓库为只读状态, 无法提交/上传代码。关闭自动同步后, 代码仓库可写, 但是无法再次恢复自动同步。

A: 只有导入的仓库, 才有此功能。

## 4.6 删除 Git 分支的创建者, 相关的 Git 代码分支也会自动删除吗?

不会, 删除 git 分支的创建者后, 这个分支就会移交给其他仓库成员, 比如仓库管理者。可以在设置-操作历史里看到删除过分支的记录, 这个层面是针对于整个仓库的动态, 要看具体分支的历史要到具体的分支-历史去查看历史操作。

---

# 5 Git 相关问题

---

- Git 如何判断是代码仓库管理员？
- 如何获取代码下载的存储路径？
- 怎样获取代码仓库地址？
- 复制的仓库地址用在哪些场景？
- 代码托管服务是否支持 SVN？
- 向代码仓库中上传压缩包能否在线解压？

## 5.1 Git 如何判断是代码仓库管理员？

- 若使用 HTTPS 方式，连接云端仓库时需要输入用户名与密码。  
用户名为 HTTPS 用户名，格式是：帐号/子帐号，如果当前登录的是主帐号用户，则格式为：帐号/帐号，通过用户名即可判断用户在仓库中的角色。
- 若使用 SSH 方式，连接云端仓库前需要先配置 SSH 密钥。  
在代码托管中，每个密钥只能对应一个用户名。在连接云端仓库时，将记录密钥与用户名的对应关系，来判断用户在仓库中的角色。

## 5.2 如何获取代码下载的存储路径？

- 若按照默认路径安装 Git，通过本地 PC 上的开始菜单栏打开 **Git Bash**，则通常默认的存储路径为“C:/User/XX 用户”。
- 若在某个文件夹内通过鼠标右键菜单打开 **Git bash Here**，则存储路径就是该文件夹。



## 5.3 怎样获取代码仓库地址？

步骤 1 进入代码托管首页，单击仓库列表中的仓库名进入仓库详情页。

步骤 2 单击右侧导航栏“克隆/下载”按钮，单击“用 SSH 克隆”获取到 SSH 协议地址。



步骤 3 单击“用 HTTPS 克隆”获取到 HTTPS 协议地址。



---结束

#### 须知

- 克隆地址仅用于克隆，不支持页面直接访问。
- 可用仓库的页面 url 进行页面访问。

## 5.4 复制的仓库地址用在哪些场景？

在连接云端仓库时，需要使用仓库地址，例如：

- 克隆云端代码仓库到本地时，使用命令：`git clone 云端仓库 url`
- 在本地 `init` 仓库后，需要与云端仓库连接时，使用命令：`git remote add origin 云端仓库 url`

## 5.5 代码托管服务是否支持 SVN？

不支持。

代码托管服务提供基于 Git 的分布式版本控制管理服务，能够更加便捷的进行异地协作。

虽然不支持 SVN，但可以将 SVN 代码库导入到代码仓库中。

## 5.6 向代码仓库中上传压缩包能否在线解压？

不能。

---

暂不支持在线解压缩，建议在本地解压之后使用 `Git` 命令上传。

---

# 6

## 常见报错解决方法

---

私钥丢失

此密钥已存在，请重新生成密钥

使用 SSH 协议克隆代码时一直提示输入密码

使用 HTTPS 方式克隆代码时，输入登录密码报错

提交代码之后，仓库找不到文件

pathspec XXX did not match any files

Transport Error: cannot get remote repository refs. XXX.git: cannot open git-upload-pack

syntax error near unexpected token `newline'

unable to auto-detect email address

fatal:Authentication failed

'origin' does not appear to be a git repository

You are not allowed to push code to protected branches on this project

Not a git repository

src refspec master does not match any

destination path 'XXX' already exists and is not an empty directory

The requested URL returned error: 401

向代码仓库推送代码失败

使用 git pull 拉取代码失败

fatal: refusing to merge unrelated histories

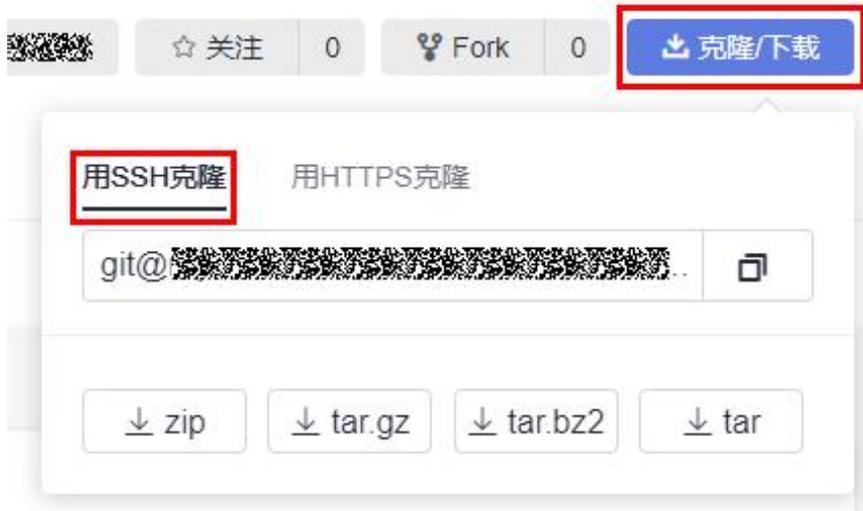
SSL certificate problem



图 6-2 复制密钥

```
[root@... ~]# cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA...  
[root@... ~]#
```

5. 在“SSH 密钥管理”页面，单击“添加 SSH 密钥”添加新的 SSH 密钥。
6. 在代码托管首页单击代码仓库所在行“SSH”复制 SSH 协议地址，如图所示。



7. 进行代码 Clone。  
执行 `git clone` 代码库 SSH 协议地址，如图 6-3 所示。

图 6-3 clone 代码

```
[root@... ~]# git clone git@codehub-.../master.git  
Initialized empty Git repository in /root/master/.git/  
\\Hremote: Counting objects: 16, done.  
remote: Compressing objects: 100% (11/11), done.  
remote: Total 16 (delta 2), reused 0 (delta 0)  
Receiving objects: 100% (16/16), done.  
Resolving deltas: 100% (2/2), done.
```

## 6.2 此密钥已存在，请重新生成密钥

### 问题现象

添加 SSH 密钥时，提示“此密钥已存在，请重新生成密钥”。

### 原因分析

每个密钥只能对应一个用户名，通常是由于密钥在其它帐户下被添加过。

---

## 处理方法

- 找到添加过该密钥的用户，删除密钥。
- 重新生成一次 SSH 密钥。

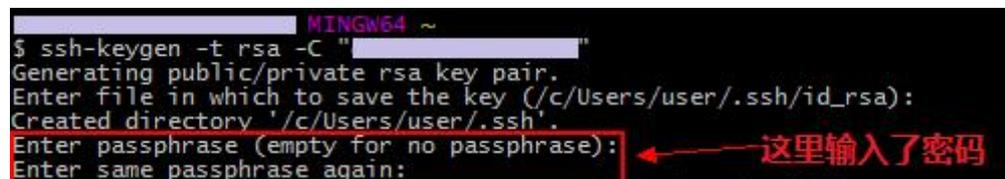
## 6.3 使用 SSH 协议克隆代码时一直提示输入密码

### 问题现象

使用 SSH 协议克隆代码时，一直提示输入密码。

### 原因分析

由于在生成 SSH 密钥时，添加了密码，需要输入的是 SSH 密钥的密码。



```
MINGW64 ~
$ ssh-keygen -t rsa -C " "
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/user/.ssh/id_rsa):
Created directory '/c/Users/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

### 处理方法

- 输入在生成 SSH 密钥时设置的密码。
- 删掉 SSH 密钥，重新生成一次，在上图标记出的位置按回车键，不输入任何内容。

## 6.4 使用 HTTPS 方式克隆代码时，输入登录密码报错

### 问题现象

使用 HTTPS 方式克隆代码时，输入登录密码后一直报错。



```
ty@DESKTOP ~ MINGW64 ~
$ git clone https://[redacted]/test.git
Cloning into 'test'...
fatal: Authentication failed for 'https://[redacted]/test.git/'
```

### 原因分析

使用 HTTPS 方式连接代码仓库时，输入的密码不是 CodeArts 登录密码，而应该是在代码托管服务中设置的密码。

### 处理方法

登录您的代码托管服务仓库列表页，单击右上角昵称，单击“个人设置 > HTTPS 密码管理”，进入页面，重置 HTTPS 密码即可。



#### 📖 说明

1. 使用 HTTPS 方式连接代码仓库时，用户名要输入完整用户名（xxx/xxx）。
2. 邮箱验证码通过单击“发送邮箱验证码”获取。
3. 忘记密码，设置新的 HTTPS 密码即可。

## 6.5 提交代码之后，仓库找不到文件

### 问题现象

使用 `git commit` 提交代码之后，仓库中找不到文件。

### 处理方法

`git commit` 命令是将本地修改过的文件提交到本地库中，若想推送到云端仓库中，需要使用 `git push` 命令。

## 6.6 pathspec XXX did not match any files

### 问题现象

使用 `git add` 命令时，提示“pathspec XXX did not match any files”。

```
MINGW64 /d/GitStudy (master)
$ git add README.md
fatal: pathspec 'README.md' did not match any files
```

---

## 原因分析

本地并不存在“README.md”文件。`git add`命令是将已经存在的文件添加到暂存区，并没有创建文件的功能。

## 处理方法

方法一：在仓库中手动创建“README.md”文件，再进行 `add` 操作。

方法二：使用命令 `touch README.md` 创建文件，再进行 `add` 操作。

```
MINGW64 /d/GitStudy (master)
$ touch README.md

MINGW64 /d/GitStudy (master)
$ git add README.md

MINGW64 /d/GitStudy (master)
$ |
```

## 6.7 Transport Error: cannot get remote repository refs. XXX.git: cannot open git-upload-pack

### 问题现象

在 Eclipse 使用 HTTPS 方式连接云端仓库时，提示“Transport Error: cannot get remote repository refs. XXX.git: cannot open git-upload-pack”。

### 原因分析

由于 Eclipse 中 Egit 插件的配置问题。

### 处理方法

1. 在本地开发工具 Eclipse 中，选择“Windows > Preferences > Team > Git > Configuration > User Settings”。
2. 单击“Add Entry”，添加一个键值对：**http.sslVerify=false**



## 6.8 syntax error near unexpected token `newline'

### 问题现象

使用 **git add** 命令时，提示 “syntax error near unexpected token `newline'”。

```
MINGW64 /d/GitStudy (master)
$ git add <README.md>
bash: syntax error near unexpected token `newline'
```

### 原因分析

命令中有无效的占位符 “<”。

### 处理方法

将 “<” 符号去掉即可，即输入命令：**git add README.md**。

## 6.9 unable to auto-detect email address

### 问题现象

使用 **git commit** 命令后，提示 “unable to auto-detect email address”。

```
MINGW64 ~/GitStudy (master)
$ git commit -m "first commit"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got '          .(none)')
```

## 原因分析

由于没有进行用户名、邮箱设置所致。

## 处理方法

1. 执行下面两行命令完整设置。

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

2. 运行 **git commit** 即可。

## 6.10 fatal:Authentication failed

### 问题现象

使用 HTTPS 方式克隆代码，即 **git clone** 时，提示 “fatal:Authentication failed”。

### 原因分析

提示 “Authentication failed” 认证失败，通常是由于 HTTPS 用户名密码填写错误导致。

### 处理方法

请检查填写的用户名密码是否正确。

## 6.11 'origin' does not appear to be a git repository

### 问题现象

向云端仓库推送代码时，提示 “'origin' does not appear to be a git repository”。

### 原因分析

云端的代码仓库没有初始化成功。

---

## 处理方法

使用 `git push -u origin master` 命令进行强制关联并推送。

## 6.12 You are not allowed to push code to protected branches on this project

### 问题现象

向代码仓库推送代码时，提示 “You are not allowed to push code to protected branches on this project”。

### 原因分析

该分支为受保护分支，用户没有权限推送代码到这个分支。

### 处理方法 1: 修改分支保护设置

进入代码仓库详情页，选择 “设置 > 策略设置 > 保护分支”，解除对该分支的保护。

### 处理方法 2: 修改仓库成员角色

进入代码仓库详情页，单击 “成员” 页签，搜索目标用户，单击设置管理员图标 ，修改仓库成员角色，将该成员设置为仓库管理员。

## 6.13 Not a git repository

### 问题现象

使用 `git add` 命令时，提示 “Not a git repository”。

```
MINGW64 ~  
$ git add README.md  
fatal: Not a git repository (or any of the parent directories): .git
```

### 原因分析

当前所处的目录并非代码仓库目录。

### 处理方法 1

需要找到本地代码仓库目录地址，通过命令：`cd 仓库名称`，切换到代码仓库中再进行 `add` 操作。

---

```
MINGW64 ~
$ cd Devcloud
MINGW64 ~/Devcloud (master)
$ git add README.md
MINGW64 ~/Devcloud (master)
$ |
```

## 处理方法 2

使用 `git init` 命令将当前目录初始化为代码仓库，再进行 `add` 操作。

```
MINGW64 ~
$ git init
Initialized empty Git repository in C:/Users/[redacted]/.git/
MINGW64 ~ (master)
$ git add README.md
MINGW64 ~ (master)
$
```

## 6.14 src refs spec master does not match any

### 问题现象

向代码仓库推送代码时，提示“src refs spec master does not match any”。

### 原因分析

本地的版本库中没有任何可供上传的文件。

### 处理方法

使用 `add` 与 `commit` 命令，将修改后的文件提交至暂存区中，再使用 `push` 命令推送至云端代码仓库中。

## 6.15 destination path 'XXX' already exists and is not an empty directory

### 问题现象

从云端克隆代码时，提示“destination path 'XXX' already exists and is not an empty directory”。

### 原因分析

所克隆的代码仓库已存在并且不为空。

---

## 处理方法

克隆项目至新目录。

1. 进入目录:

```
cd 本地代码仓库目录
```

2. 克隆云端代码仓库到临时目录 tmp 中:

```
git clone --no-checkout 云端代码仓库地址 tmp
```

3. 将 tmp 目录下的 .git 目录移到当前目录:

```
mv tmp/.git 本地代码仓库目录名
```

4. 删除 tmp 目录:

```
rmdir tmp
```

5. 回退至上个版本:

```
git reset --hard HEAD
```

## 6.16 The requested URL returned error: 401

### 问题现象

在 Centos 系统下使用 https 方式克隆代码时，系统报 “The requested URL returned error: 401”。

### 原因分析

通常是 Git 版本的问题。

### 处理方法

通过 **git --version** 命令查看系统自带的版本，**Centos6.5** 自带的 Git 版本通常是 1.7.1。

1. 卸载 Centos 自带的 Git1.7.1。

```
# yum remove git
```

2. 到 Git 官网下载最新版本，并将 Git 添加到环境变量中。

```
# wget https://github.com/git/git/archive/版本号.tar.gz
# tar zxvf 版本号.tar.gz
# cd git-版本号
# make configure
# ./configure --prefix=/usr/local/git --with-iconv=/usr/local/libiconv
# make all doc
# make install install-doc install-html
# echo "export PATH=$PATH:/usr/local/git/bin" >> /etc/bashrc
# source /etc/bashrc
```

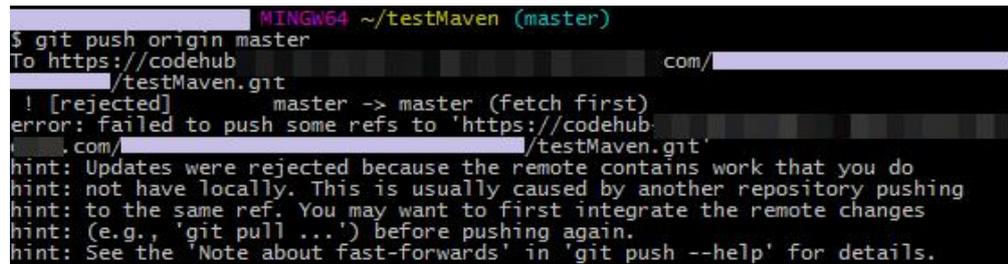
---

## 6.17 向代码仓库推送代码失败

### 问题现象

向代码仓库推送代码时，系统提示提交失败，提交失败提示信息如下图所示。

图 6-4 提示信息



```
MINGW64 ~/testMaven (master)
$ git push origin master
To https://codehub.com/~/testMaven.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://codehub.com/~/testMaven.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

### 原因分析

从提示信息（hint）看，由于云端代码仓库有变化，与本地仓库代码不一致，所以从本地提交代码的操作被拒绝。

### 处理方法

先使用 **git pull** 命令拉取云端仓库，与本地仓库合并，再使用 **push** 命令推送代码。

## 6.18 使用 git pull 拉取代码失败

### 问题现象

使用 **git pull** 命令拉取代码失败，无法继续操作，拉取代码失败信息如下图所示。



