当在git clone gitbub上的代码的时候发现没有权限

Administrator@BN-PC204 MINGW64 /d/Linux\_pi\_git

$ git clone git@github.com:yuyong2320/first-rep.git

Cloning into 'first-rep'...

git@github.com: Permission denied (publickey).

fatal: Could not read from remote repository.

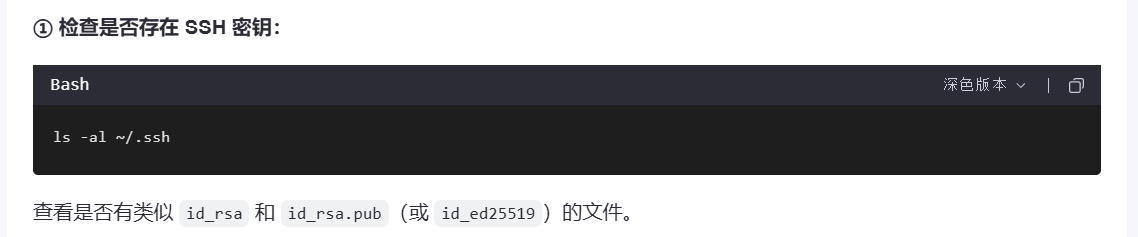
Please make sure you have the correct access rights

and the repository exists.

上面输出显示是没有权限

1 检查是否有ssh密钥

Git配置相关问题



## **没有的话可以生成**

## **生成新的 SSH 密钥**

运行以下命令来生成新的密钥（推荐使用更安全的 ed25519 算法）：

ssh-keygen -t ed25519 -C "你的邮箱@example.com"

具体步骤：

Administrator@BN-PC204 MINGW64 /d/Linux\_pi\_git

$ ssh-keygen -t ed25519 -C "1467998597@qq.com"

Generating public/private ed25519 key pair.

Enter file in which to save the key (/c/Users/Administrator/.ssh/id\_ed25519):

Enter passphrase for "/c/Users/Administrator/.ssh/id\_ed25519" (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/Administrator/.ssh/id\_ed25519

Your public key has been saved in /c/Users/Administrator/.ssh/id\_ed25519.pub

The key fingerprint is:

SHA256:IPw7xOGuVMPrd2nkDL/V2QbGzYUE85X+Ppd6BtmrDXU 1467998597@qq.com

The key's randomart image is:

+--[ED25519 256]--+

| o.. o|

| . + + |

| o o + .|

| \* o . +.|

| O S \*.E|

| + +. . =.=o|

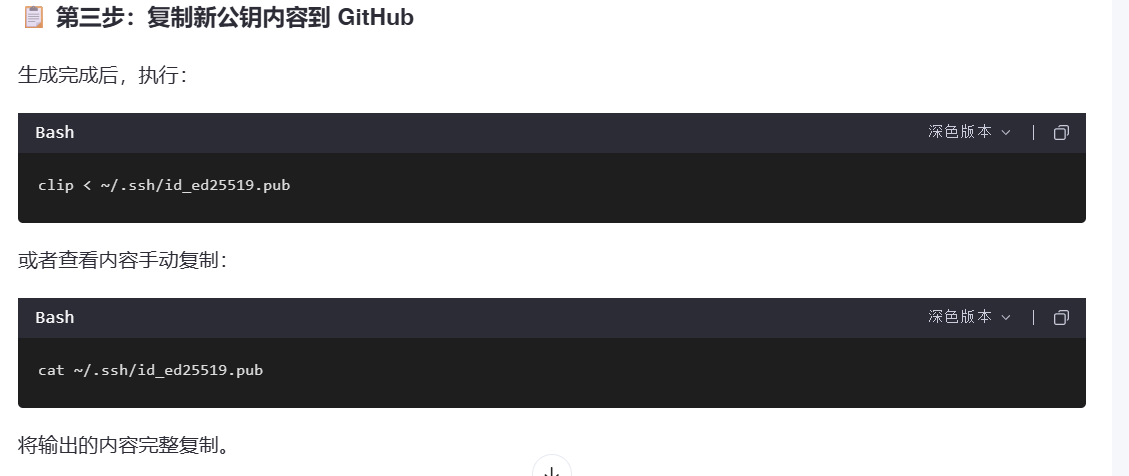
| . = \* ...+.=|

| . o .. B. oBo|

| . .. o.. o=.o|

+----[SHA256]-----+

3、复制公钥到github



4、打开github粘贴

打开github ssh配置界面

[SSH and GPG keys](https://github.com/settings/keys)



5、启动ssh angent 并加载新密钥



eval $(ssh-agent) # 启动代理

ssh-add ~/.ssh/id\_ed25519 # 添加私钥

Administrator@BN-PC204 MINGW64 /d/Linux\_pi\_git

$ eval $(ssh-agent)

Agent pid 1147

Administrator@BN-PC204 MINGW64 /d/Linux\_pi\_git

$ ssh-add ~/.ssh/id\_ed25519

Identity added: /c/Users/Administrator/.ssh/id\_ed25519 (1467998597@qq.com)



## **查看远程所有分支（包括本地没有的）**

Git branch 查看本地分支（\*为当前分支）

Git branch -r 查看远程分支

Administrator@BN-PC204 MINGW64 /d/Linux\_pi\_git/first-rep (main)

$ git branch -r

origin/HEAD -> origin/main

origin/dev

origin/main

## **查看本地和远程分支的对应关系**

这个命令会显示本地分支跟踪的是哪个远程分支

Administrator@BN-PC204 MINGW64 /d/Linux\_pi\_git/first-rep (main)

$ git branch -vv

\* main cce3028 [origin/main] feat1

！！！！！！切换分支 git checkout ...



Git 会自动创建本地 dev 分支并跟踪远程的 origin/dev。

、、、

## **分两步操作（先创建后切换）**

git branch dev # 创建分支

git checkout dev # 切换到该分支

如果你后来发现这个分支不需要了，可以删除它

git branch -d dev0620

如果该分支有未合并的更改，Git 会提示你是否要强制删除

git branch -D dev0620 # 强制删除

Git 合并代码