**GitHub - Git配置多个用户的SSH-Key**

**1、为什么要配置多个SSHKey**

因为有的时候会出现这样的Git使用场景，就比如我们在公司里面，会有一个公司的GitHub地址。然后我们自己平时练习写一些代码，会存到自己的Github上。而且现在很多公司用的是Gitlab，（实际上Github和Gitlab命令、操作等都是一模一样的，然后配置的方法他们也是一样的），Gitlab也会有一个地址。这样简短的一段描述，就会配置三个SSHKey账户。

所以在我们开发过程中，可能会遇到使用同一台机器，既要向公司Git服务器提交代码，也要向Github或者Gitlab等提交代码。向不同的仓库中提交代码，所需要的用户名信息也是不一样的，此时就需要配置多个SSH Key管理多个用户。

**2、配置多个sshkey**

**步骤1：生成SSHkey**

生成一个公司用的SSH-Key密钥

|  |
| --- |
| Ruby $ ssh-keygen -t rsa -C 'xxxxx@company.com' -f ~/.ssh/company\_id\_rsa |

生成一个Github用的SSH-Key密钥

|  |
| --- |
| Ruby $ ssh-keygen -t rsa -C 'xxxxx@qq.com' -f ~/.ssh/github\_id\_rsa |

此时，~/.ssh目录下应该有4个文件：company\_id\_rsa文件和company\_id\_rsa文件，还有github\_id\_rsa文件和github\_id\_rsa.pub文件。

|  |
| --- |
| **注意：**：密钥文件的名称不要和之前定义的密钥文件名称冲突，不过最好还是起一个比较有意义的名字。 |

**步骤2：将的公钥配置到GitHub官网上**

详细步骤请看前面文章。

**步骤3：在~/.ssh目录下创建一个config文件**

在.ssh目录创建一个config文本文件，并完成相关配置。

每个账号单独配置一个**Host**，每个**Host**要取一个别名，每个Host主要配置**HostName**和**IdentityFile**两个属性即可。

常用编辑配置说明：

* Host：定义**Host**的名字，可以任取自己喜欢的名字，不过这个会影响Git相关命令。 例如：Host mygithub 这样定义的话，即git@后面紧跟的名字改为**mygithub**。 命令如下：git clone git@mygithub:PopFisher/AndroidRotateAnim.git。 一般都会和HostName属性其一样的名字。
* HostName：这个是真实的域名地址，要登录主机的主机名。（建议与Host一致）
* IdentityFile：指定私钥文件的路径，也就是id\_rsa文件的绝对路径。
* User：配置登录名，例如：GitHub的username。
* Port：端口号（如果不是默认22端口，则需要指定端口号）
* PreferredAuthentications：配置登录时用什么权限认证，可设为publickey,password publickey,keyboard-interactive等。

示例：

|  |
| --- |
| Ruby # 1.个人gitHost github.com（github基本固定写法）  HostName github.com（github基本固定写法）  User \*\*\*\*IdentityFile /Users/LLL/.ssh/github\_id\_rsa  PreferredAuthentications publickey  # 2.公司gitHost github.com（写入公司的git地址）  HostName github.com（写入公司的git地址）  User \*\*\*\*（公司的用户名）  IdentityFile /Users/LLL/.ssh/company\_id\_rsa（公司密钥）  PreferredAuthentications publickey   # 3.gitee and github # 注意：多个Host公用一个公钥时， # 对应的HostName和Port必须跟在Host后面Host gitee.com  HostName gitee.com  Host github.com  HostName github.com  PreferredAuthentications publickey  IdentityFile ~/.ssh/id\_rsa |

**步骤4：测试SSHKey否配置成功**

使用ssh -T命令校验，以GitHub为例：

|  |
| --- |
| CSS ssh -T git@github.com |

如果成功会显示如下: Hi cary! You've successfully authenticated, but GitHub does not provide shell access.：您已经成功通过身份验证。

|  |
| --- |
| 提示：如果是云码，则执行$ ssh -T git@gitee.com命令。 |

多个账户分别验证通过即可。

说明多账户SSHKey配置成功。

如果测试没有成功的话，教你解决问题的终极办法。

比如测试GitHub远程仓库：

|  |
| --- |
| CSS ssh -vT git@github.com |

-v参数是输出编译信息，然后根据编译信息自己去解决问题吧。（一般来说，都是config文件中的host那块写错了）

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/c82a24c9d752 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。