为什么要设置多个SSH key？

一个github账户的后台可以添加多个ssh key（可以多人修改该项目），但是同一个ssh key只能添加到一个github后台上（同一个ssh key可以添加到不同的服务器上）。这也可以理解，毕竟ssh key相当于密码，同一个密码当然只能对应一个账户了。那如果本地要修改多个github账户下的项目呢？只能添加多个git用户了。

生成新的ssh key

1.使git当前路径为**C:\Users\Administrator\.ssh**，可以使用$ cd ~/.ssh进入；

2. 在git中用$ ssh-keygen -t rsa -C "<name>@<email>"新建SSH，这里，<name>可以用户名，或者该ssh解决什么工作，只是描述；

3.当出现Enter file in which to save the key (/c/Users/Administrator/.ssh/id\_rsa): 时，在冒号后面填写ssh名称，一般样式为id\_ssh\_<…自己决定>；

我们打开**.ssh**文件夹可以发现生成了两个新的东西，**id\_rsa\_<…>**和**id\_rsa\_<…>.pub**，其中公钥就是用来添加到对应github账户的后台key setting中，但在添加之前仍需要一些进行下面的配置。

新增ssh的配置文件，并修改权限

$ touch ~/.ssh/config

$ chmod 600 ~/.ssh/config

此时**.ssh**文件夹下出现个**config**文件。

修改config文件，其中内容如下

# 该文件用于配置私钥对应的服务器

# Default github user(first@mail.com)

Host **github.com**

HostName github.com

User git

IdentityFile C:/Users/Administrator/.ssh/**id\_rsa**

# second user(second@mail.com)

# 建一个github别名，新建的帐号使用这个别名做克隆和更新

Host **github2**

HostName github.com

User git

IdentityFile C:/Users/Administrator/.ssh/**<新增的ssh文件名>**

需要注意两个**Host**后面的值不同，这个涉及到使用其进行**push**提交时**url**会有不同，所以接下来会去项目中设置**.git**文件夹下的配置文件**config**。

**修改具体项目的config文件**

当把别人github上的repo克隆clone下来时，相当于设置了这个本地仓库与别人远程仓库的链接，即配置了项目**.git**文件夹下的配置文件**config**文件，此时我们要使用**新配置的ssh key**与别人的远程仓库通讯，而**不是默认的**id\_rsa这个ssh key，就需要更改这个**config**文件中的内容，更改如下：

url = **git@github.com**:< 别人的account>/< 别人的repo >.git #修改前

url = **github2**: < 别人的account>/< 别人的repo >.git #修改后

将**git@github.com**改为**github2**，即改为我们之前在**C:\Users\Administrator\.ssh**文件夹下的**config**文件中的**第二个Host值**。

**测试ssh是否可以正常通讯**

$ ssh –T <Host>

这里的**Host**指**C:\Users\Administrator\.ssh**文件夹下的**config**文件的**Host**值，要测试哪个**ssh key**是否可以正常使用，就填哪个**ssh key**所对应的**Host**值。

例如，测试**前面的新ssh key**是否可用就$ ssh –T github2，如果返回welcome…或者hi…这类语句，则表示已经正常链接。