4b. Software Installation Guide - SSH Keys

Z620: Quantitative Biodiversity, Indiana University

OVERVIEW

While QB adheres to a philosophy of openness and collaboration, it is also important for institutions and individuals to protect their data and computing environments. One of many ways to achieve this type of security is to use **SSH keys**. SSH stands for *secure shell*. To use SSH, you generate a pair of keys: one public and one private. To date, in QB, you have been using **Hyper Text Transfer Protocol Secure (HTTPS)** for cloning and pushing files from and to GitHub. While HTTPS offers security, you are frequently asked to supply your username and passphrase, which can get annoying after a while. In this short document, we show you how to add SSH keys to your local computers so that you can use GitHub and do reproducible science with a little less hassle. Note: if you want to connect to GitHub with SSH keys, you will need to complete the following steps for *each* computer that you use.

1) GENERATE AN SSH KEY

Using Rstudio

One way to generate an SSH key is through RStudio. Go to Preferences and choose the Git/SVN tab (with box-looking icon). Depending on the version of RStudio on your local computer, this same tab is found in a dialog box that is accessed by opening the Tools tab and clicking on Global Options. Make sure the box is checked for `Enable version control interface for RStudio projects`. In the Git executable box, it may say `/usr/bin/git` and SVN executable box may say `usr/bin/svn`. For Windows users the Git executable box may contain `C:/Program Files/Git/bin/git.exe`. In the `SSH RSA Key` box, you should type `~/.ssh/id_rsa/`. You can then hit the button that says `Create RSA Key...` You can now view the public key. Copy this, so you can paste it into GitHub.

Using Terminal

Alternatively, you can generate an SSH key by typing the following at the command prompt:

```
ssh-keygen -t rsa -C "your_email@example.com"
```

You will be asked to enter and re-enter a passphrase. After that, you need to add the new key to the SSH-agent using the following commands, which will generate an agent pid (process identifier).

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/id_rsa
```

To obtain the SSH key you just generated, type the following command at the Terminal. (Note: your key may be named one of the following instead of id_rsa.pub: id_dsa.pub, id_ecdsa.pub or id_ed25519.pub)

```
pbcopy < ~/.ssh/id_rsa.pub
```

2) PUT SSH KEY INTO GITHUB

Log in to your GitHub site at www.github.com. In the upper right hand corner, click on your profile and choose `Settings`. Now, on the left hand side under `Personal Settings`, click on `SSH and GPG keys`. (Alternatively, type https://github.com/settings/keys) In the upper right, click on the green button that says `New SSH Key`. On the new page that opens, add a descriptive title (e.g., Jay's MacBook). Now paste the SSH key into the key window and hit the green `Add SSH Key` button. You may be asked to supply your GitHub password.

3) ADD KEY TO LOCAL COMPUTER KEY CHAIN

On Mac: Open the terminal and type the following. This will add your ssh key to Apple's keychain (K)

```
ssh-add -K ~/.ssh/id_rsa
ssh-add
~/.ssh/id_rsa
```

On PC:

For windows users who are using git bash, the above "ssh" commands will all end in .exe (e.g. `ssh-keygen.exe`).

Additionally, `pbcopy` is a Mac specific program, but you can use `clip.exe` command instead:

```
clip.exe < ~/.ssh/id_rsa.pub
```