

Authenticating with GitHub using SSH keys

Software Engineering for Scientist

SSH (Secure Shell) keys are a means to authenticate yourself with a remote server, similar to a username and password (u/p). The key practical difference is that once you set up SSH key authentication, you will no longer need to enter your u/p when pushing commits. GitHub will use your SSH keys to figure out who you are and if you have permission to push a commit, or pull a private repo, etc.

SSH keys come in pairs that include a private key and a public key. Just as your password in u/p authentication is a secret, the private key is also a secret and should not be shared. The public key can be shared freely, and we will end up storing our public key on GitHub. The relationship between a particular public/private key pair is that anything that the public key encrypts can only be decrypted by its associated private key. When you attempt to commit a new branch, GitHub uses your public key it has on file to encrypt some message. If you are able to decrypt that message, then you have proved that you have the right private and access is granted.

Generate a private/public key pair

Before generating a pair of keys, make sure that you do not already have one. In OS X and Linux systems, the keys are usually stored in the `$HOME/.ssh` directory. Check to see if there is anything in this directory

You do not have an SSH key pair:

```
$ ls $HOME/.ssh
ls: cannot access '/home/jovyan/.ssh': No such file or directory
```

You have an SSH key pair:

```
$ ls $HOME/.ssh
id_rsa  id_rsa.pub
```

If you do not have an SSH key pair use `ssh-keygen` to make one. If you do not want to enter your password every time you use your keys, just hit return for the passphrase question.

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jovyan/.ssh/id_rsa):
Created directory '/home/jovyan/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jovyan/.ssh/id_rsa
Your public key has been saved in /home/jovyan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:49x0WuGLHecsLZzpDaM8sdDEBps7oJ8zs6nLiMSxuv4 jovyan@jupyter-ryla3858
The key's randomart image is:
+---[RSA 3072]-----+
|
|      .
|      =
|*o=Eo=  o.o .
+-----[SHA256]-----+
```

Your public key is at `/home/jovyan/.ssh/id_rsa.pub`

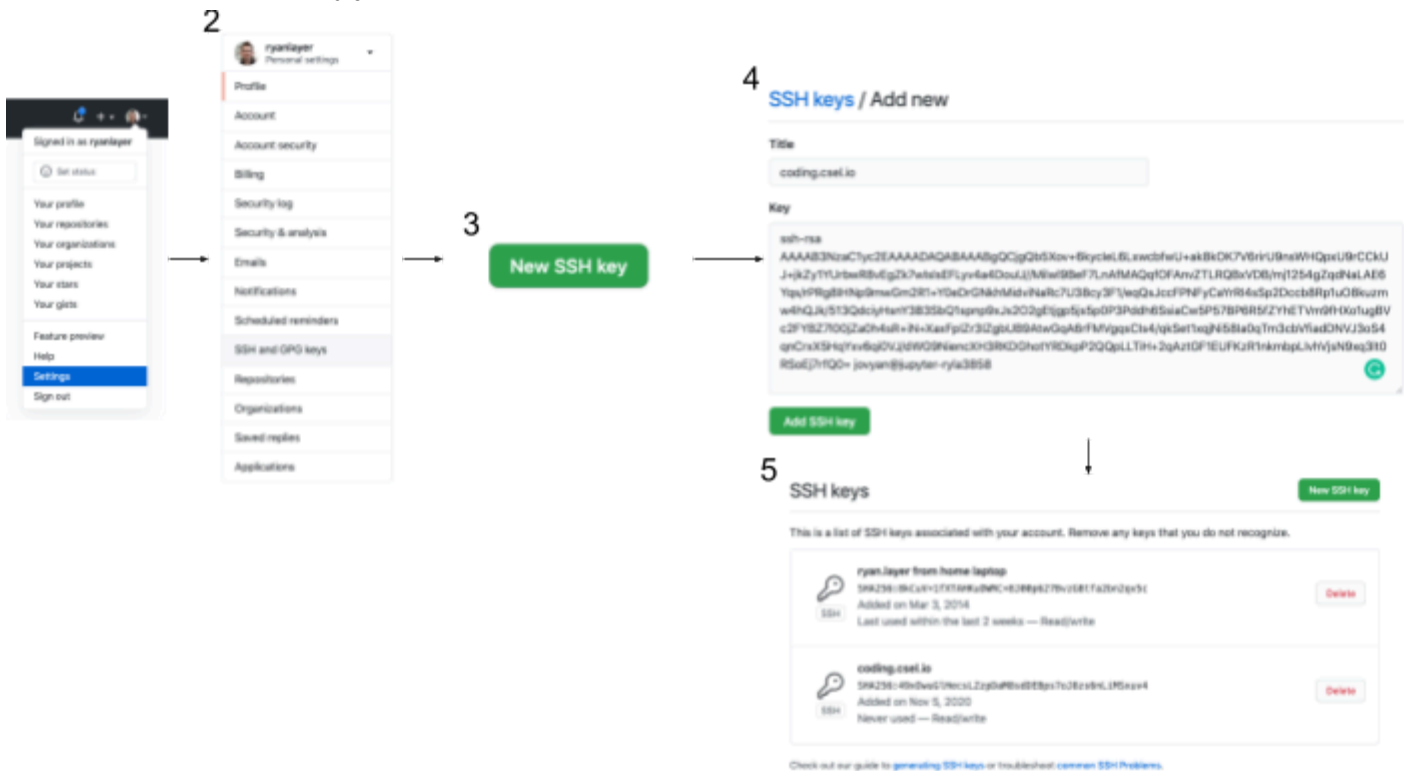
```
$ cat /home/jovyan/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGCjgQb5Xov+6kycIeL6LxwcbfwU+ak8kOK7V6rirU9nsWH
QpxU9rCCkUJ+jkZy1YUrbwR8vEgZk7wIsIsEFLyv4a4DouU//Milwl9BeF7LnAfMAQqf0FAnvZT
LRQBxVDB/mj1254gZqdNaLAE6Yqs/rPRg8IHNp9mwGm2R1+Y0eDrGNkhMidviNaRc7U38cy3F1/
eqQsJccFPNFyCaYrRI4sSp2Dccb8Rp1u08kuzmw4hQJk/513QdcyHsnY3B3SbQ1spnp9xJs202
gEtjgp5js5p0P3Pddh6SsiaCw5P57BP6R5fZYhETVm9fHXo1ugBVc2FYBZ7l00jZa0h4sR+iN+X
axFpIZr3IZgbUB9AtwGqA6rFMVgqsCIs4/qkSet1xqjNi58Ia0qTm3cbVfiadDNVJ3oS4qnCrXx
5HqYxv6qi0VJ/dWG9NiencXH3RKDGhotYRDkpP2QQpLLTiH+2qAztGF1EUFKzR1nkmbpLIvhVjs
N9xq3lt0RSoEj7rfQ0= jovyan@jupyter-ryla3858
```

Your private key at `home/jovyan/.ssh/id_rsa`. **Do not share your private key. Leave it right there and do nothing with it.**

Store your public key on GitHub

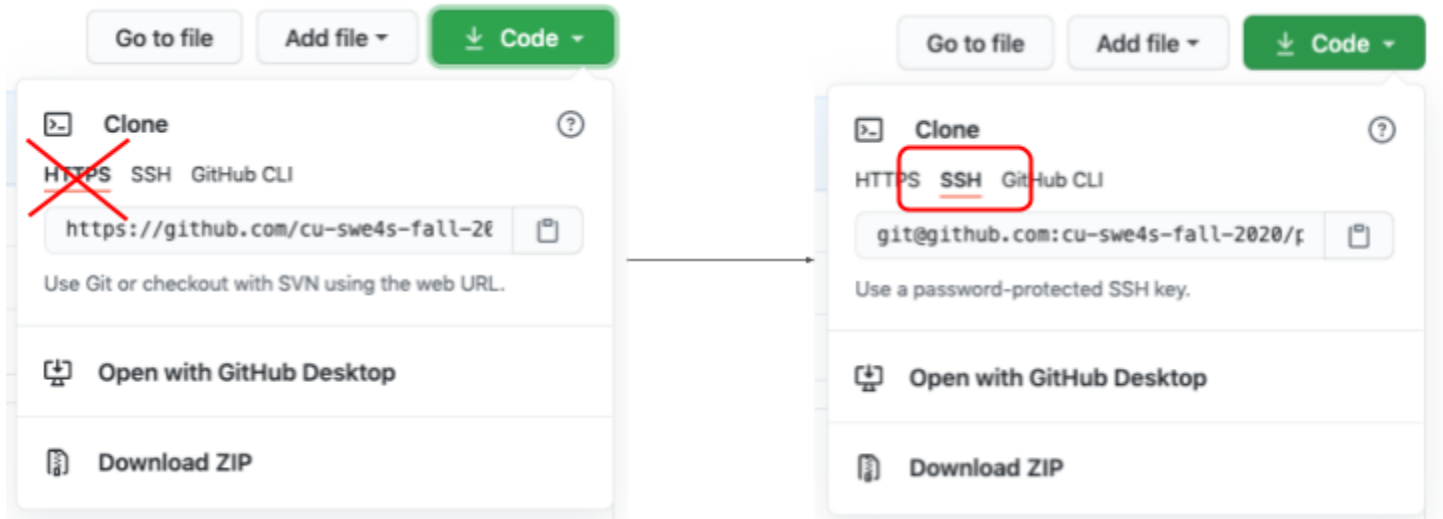
Log into your GitHub account

1. Goto the settings page
2. Select SSH and GPG keys
3. Click the “New SSH key” Button
4. Give you key a title. I usually select what computer the private key is stored on (i.e, my laptop, the research cluster, etc). Copy and paste the key into the next box. Remember to include the first line that says “ssh-rsa”. Then click the Add SSH key button.
5. The new public key joins the list



Start using SSH to interact with your repositories

You can either start fresh and clone a repo using SSH



Or edit the configurations of the repos you have already cloned. The config is stored in `.git/config` of the root directory of your repository. You can use a text editor to directly modify this file.

HTTP-based config

```
$ cat ss_plots/.git/config
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = https://github.com/swe4s/ss_plots.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
  remote = origin
  merge = refs/heads/main
```

SSH-based config

```
$ cat ss_plots/.git/config
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = git@github.com:swe4s/ss_plots.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
  remote = origin
  merge = refs/heads/main
```

The first time you connect to github.com using SSH, your computer will want to store some information about github.

```
$ git clone git@github.com:swe4s/ss_plots.git
Cloning into 'ss_plots'...
The authenticity of host 'github.com (140.82.113.3)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Answer yes (type “yes”) if you want this to work.

```
$ git clone git@github.com:swe4s/ss_plots.git
Cloning into 'ss_plots'...
The authenticity of host 'github.com (140.82.113.3)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,140.82.113.3' (RSA) to the list of
known hosts.
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 34 (delta 13), reused 25 (delta 8), pack-reused 0
Receiving objects: 100% (34/34), 152.75 KiB | 2.39 MiB/s, done.
Resolving deltas: 100% (13/13), done.
```