

Git/GitHub set-up: Git config and GitHub SSH authentication

Start this process in CSEL (coding.csel.io) in the home directory (/home/jovyan/)

Git config

You must add your GitHub user information to the git config with the following:

```
git config --global user.email "<account email address>"
git config --global user.name "<GitHub username>"
```

GitHub SSH authentication

An SSH key is a security measure that allows you to authenticate your identity with some remote service. It serves the same function as a username and password (e.g. your CU Identikey credentials). In order to communicate between your coding environment (e.g. your home computer or a remote coding environment like CSEL) and GitHub you must create a key on the machine you will be working on and then provide the correct portion of that key to GitHub. That way when you go to pull down information from GitHub or push new code up to it, the site can correctly verify your identity without you having to manually type in a password. The following is a step by step process for this set up, starting at your computer's command-line.

- 1) Generate the key (use the email address associated with your GitHub account). Here we are using the ed25519 hash algorithm. There are other equivalent options like RSA.

```
jovyan@jupyter-jast1849:~$ ssh-keygen -t ed25519 -C "jast1849@colorado.edu"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/jovyan/.ssh/id_ed25519):
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_ed25519
Your public key has been saved in /home/jovyan/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:DH/JdIIoX1CMcJaAF6Et2xUf2/toSr3RqTEw8iUY6/g jast1849@colorado.edu
The key's randomart image is:
+--[ED25519 256]--+
|  .====o          |
|  .o.o.=.=        |
|  o.o = = + .     |
|  + + O o =       |
|  . . = S *       |
|    o o B + .     |
|  . . o B +       |
|  . . o *         |
|    E . o         |
+-----[SHA256]-----+
jovyan@jupyter-jast1849:~$
```

2) Confirm your key (you MUST keep your private key private)

```
jovyan@jupyter-jast1849:~$ ll ~/.ssh/  
total 8.0K  
-rw----- 1 jovyan users 411 Aug  4 19:26 id_ed25519  
-rw-r--r-- 1 jovyan users  98 Aug  4 19:26 id_ed25519.pub
```

3) Start `ssh-agent` (the manager app for your keys) in background

```
jovyan@jupyter-jast1849:~$ eval $(ssh-agent -s)  
Agent pid 157
```

4) Add your new key to the key manager

```
jovyan@jupyter-jast1849:~$ ssh-add ~/.ssh/id_ed25519  
Identity added: /home/jovyan/.ssh/id_ed25519 (jast1849@colorado.edu)
```

5) Now that your key is generated and being managed by your computer environment, you need to provide the ***PUBLIC*** portion of the key (i.e. `~/.ssh/id_ed25519.pub`) to GitHub. The contents of the public half of your key should look something like this:

```
jovyan@jupyter-jast1849:~/.ssh$ cat ~/.ssh/id_ed25519.pub  
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDj6d75z/g+108zAS7A19Xf1y7HnsSdXzhfEf  
YK6a8Qr jast1849@colorado.edu
```

6) DO NOT show the contents of the private portion of your key (i.e. `~/.ssh/id_ed25519`) to anyone. This would be akin to telling someone your password.

7) Copy the entirety of the contents of `~/.ssh/id_ed25519.pub` to your clipboard.


- 8) Go to <https://github.com/> and navigate to the “settings” page of your profile and select “SSH and GPG keys”

The screenshot shows the GitHub 'Public profile' settings page for a user named Jacob Stanley. On the left is a sidebar with 'Account settings' and a list of sub-items: Profile, Account, Appearance, Account security, Billing & plans, Security log, Security & analysis, Emails, Notifications, Scheduled reminders, SSH and GPG keys, Repositories, and Packages. The main content area is titled 'Public profile' and contains several sections: 'Name' with a text input field containing 'Jacob Stanley'; 'Public email' with a dropdown menu set to 'Select a verified email to display'; 'Bio' with a text area containing 'Postdoctoral researcher at CU Boulder'; 'URL' with an empty text input field; and 'Twitter username' with an empty text input field. Each section has a brief explanatory text below the input field.

- 9) Click the button “New SSH key” then give your key an informative title and past in the contents of the *PUBLIC* key file that you copied. It should look something like this:

The screenshot shows the 'SSH keys / Add new' page in the GitHub settings. The left sidebar is identical to the previous screenshot, but 'SSH and GPG keys' is now selected. The main content area is titled 'SSH keys / Add new' and has a 'Title' input field with the text 'CSEL environment for swe4s'. Below it is a 'Key' text area containing a long SSH public key string: 'ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDj6d75z/g+l08zAS7A19Xf1y7HnsSdXzhfEYK6a8Qrjast1849@colorado.edu'. At the bottom of the key area is a green button labeled 'Add SSH key'.

- 10) Click the button “Add SSH key” and you should be prompted to put in your password. After doing so, check that the key is listed. You may have multiple keys that are used to authenticate other machines.



CSEL environment for swe4s
SHA256:DH/JdIIoX1CMcJaAF6Et2xUf2/toSr3RqTEw8iUY6/g
Added on Aug 4, 2021
Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).