

# How to Ed with Git

COMP2X23

2021

## *Pre-requirements*

First, make sure you have `git` installed. This can be done through a variety of ways, depending on your operating system.

This is a “version control” system which pushes updates of your code to a remote server. The cool thing about this is that you can go backwards in versions if you ever pushed up something that fails or is completely bad. It’s definitely useful to learn, so I would recommend taking some time to get familiar with it, because it’s useful for daily programming things, but very much helpful when you’re working with more than one person.

Please note, you can install software to do this (source tree, git ui, etc.), but there are known problems on some of the programs so they don’t work as well with Ed. I suggest getting familiar with the terminal since it will be useful later in life as well.

- Terminal with git installed.
  - Windows - Git **Bash** (<https://gitforwindows.org/>)
    - \* <https://gitforwindows.org/>
    - \* Install the above, and use the program **git bash** (I recommend git bash, because it works well with the ssh keys and miscellaneous things)
    - \* Side note: if you use “Git CMD” it won’t work with some of the commands below, **“Git Bash”** allows you to run commands similar to what mac/linux uses.
    - \* Here’s a few videos:
      - <https://www.youtube.com/watch?v=rWboGsc6CqI>
      - <https://www.youtube.com/watch?v=nbFwejIsHlY>
      - <https://www.youtube.com/watch?v=xueHs6fycTk>
  - Mac/OSX - git in your favourite terminal emulator (“iTerm2”, “terminal”, etc.)
    - \* In your terminal can you type `git` and it gives you information?
    - \* Tutorial to install: <https://www.atlassian.com/git/tutorials/install-git>
  - Linux (any terminal with git installed)
    - \* Ubuntu: `sudo apt install git`
    - \* Arch: `sudo pacman -Sy git`
    - \* Debian: `sudo apt install git`
    - \* Centos: `yum install git`

# Creating your Private/Public Key

To access Ed, we'll need to make a key pair, which allows us to connect to Ed without having to use our password/login details. Please follow the instructions on Ed, similar to below, or you can read through some examples:

- <https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys--2>
- <https://www.freecodecamp.org/news/the-ultimate-guide-to-ssh-setting-up-ssh-keys/>

Open your terminal (or git bash on windows). Then follow along:

## 1. Make sure the ssh directory exists in your home:

```
mkdir -p ~/.ssh
```

## 2. Generate the ssh key

```
ssh-keygen -t rsa -b 4096
```

Which will give you output like below, just follow the prompts<sup>1</sup>:

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/{your_user_here}/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/{your_user_here}/.ssh/id_rsa  
Your public key has been saved in /home/{your_user_here}/.ssh/id_rsa.pub  
The key fingerprint is:  
{SHA256:.....} {user}@{host}  
The key's randomart image is:  
..
```

---

<sup>1</sup>If you would like to change the name from *id\_rsa*, then make sure that you write the entire path `/home/yourusername/.ssh/`.

Now, if we look at the files created, there are two files:

- **id\_rsa** - Your private key, keep this **PRIVATE!**
- **id\_rsa.pub** - Your public key, anyone can have this.

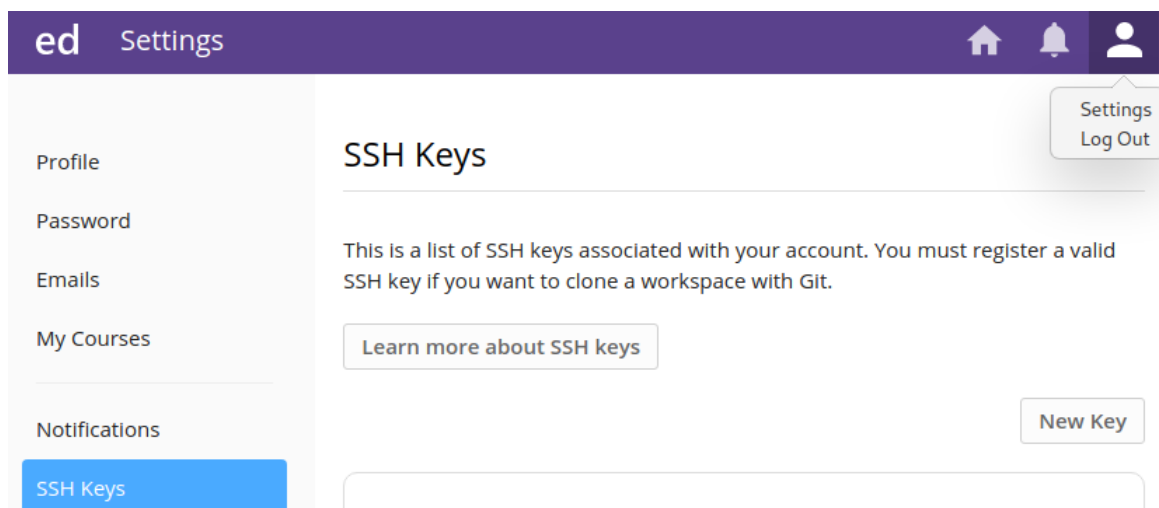
We want to open the **id\_rsa.pub**, and it should look something starting with **ssh-rsa** and ending with a **username@hostname**, such as:

```
ssh-rsa AAAAB3NzaC1yc2EAAA... Zfgi62itHm/xmPAqGy9Q== user@myhostname
```

### 3. Now, add it to ed, go to settings:



Then SSH > New SSH KEY



### SSH Keys

This is a list of SSH keys associated with your account. You must register a valid SSH key if you want to clone a workspace with Git.

[Learn more about SSH keys](#)

Title

Public key

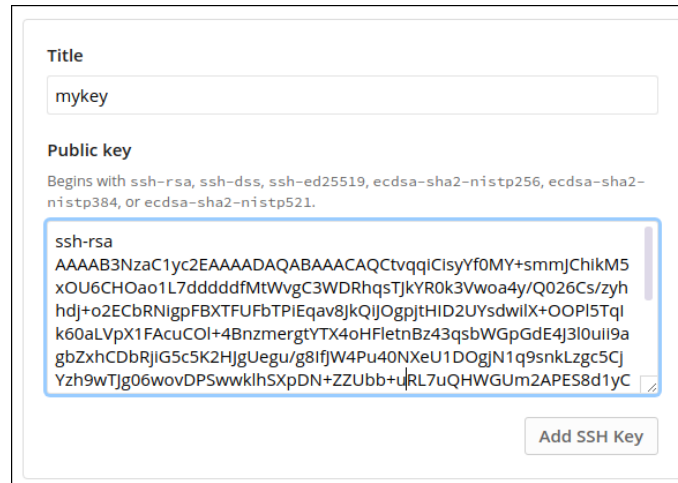
Begins with ssh-rsa, ssh-dss, ssh-ed25519, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, or ecdsa-sha2-nistp521.

[Add SSH Key](#)

In **title** - Add a title for your key, for example “Uni Laptop”. It helps you to know what the key belongs to.

Then in **public key** - add the contents of your **id\_rsa.pub**.

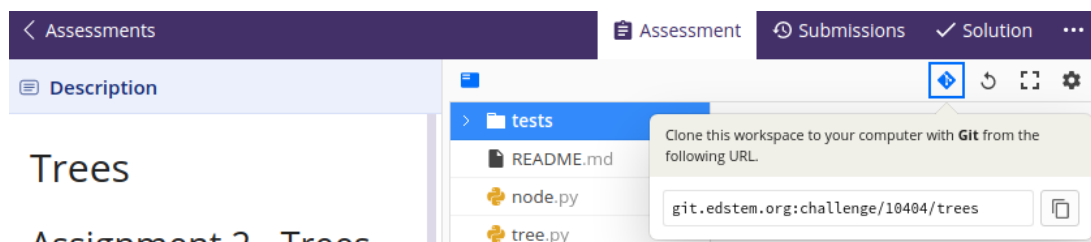
For example:



## Using Ed and Git

### 1. Cloning Your Assignment to your computer

First step, go to the assignment, and then up the top you want to click the “clone with ed” and copy the URL:



Next, go to your terminal (or git bash) and clone the assignment into your folder using the `git clone` command.

```
> git clone git.edstem.org:challenge/10404/trees
Cloning into 'trees'...
Warning: Permanently added the RSA host key for IP address '139.99.131.187' to the
list of known hosts.
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (65/65), done.
remote: Total 66 (delta 24), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (66/66), 17.84 KiB | 913.00 KiB/s, done.
Resolving deltas: 100% (24/24), done.
```

Note, if you get an error here, try run `ssh-add ~/.ssh/id_rsa` (substitute the id\_rsa for the name of your key) or check the faq below.

Confirm that the folder exists, use the `ls` command to see, then `cd` to get into the folder, in this example the assignment is called “trees”, so we will go into the trees folder and see the assignment files:

```
> ls  
  
trees  
  
> cd trees/  
  
node.py README.md tests tree.py
```

## 2. After you’ve done some work, submitting:

You will be editing the files *locally* using an editor of your choice. Once you have edited the files, you need to let Ed know of the changes.

### Submitting for the first time

```
git commit -a -m "add your own message here"
```

Then you want to do this the FIRST TIME you push::

```
git push
```

And it will submit your assignment, it will give you information about the submission.

### Submitting for the second time and onwards

Now, every time after you want to submit the same assignment, you can just do:

```
git commit -a -m "add your own message here"
```

```
git push
```

## FAQ

### Permission Denied (publickey)

This is an error saying that it can't find your public key. It means that (A) you didn't paste the correct key into Ed, or (B) your system isn't using the key to access Ed.

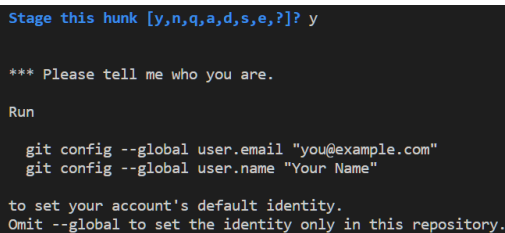
Run the command: (changing the `id_rsa` to whatever you named your key)

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

And then try again. Else, if you're on windows, there may be a few things causing issues:

- <https://serverfault.com/questions/194567/how-do-i-tell-git-for-windows-where-to-find-my-private-rsa-key>
- If you are using **GIT CMD**, it may differ again, please use **Git Bash** instead.

### \*\*\* Please tell me who you are



```
Stage this hunk [y,n,q,a,d,s,e,?]? y  
  
*** Please tell me who you are.  
  
Run  
  
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"  
  
to set your account's default identity.  
Omit --global to set the identity only in this repository.
```

This is telling you what to do in this error, it means `git` doesn't really know who you are, so you need to provide it with a *name* and *email*, so just follow the commands:

- `git config --global user.name "myname"`
- `git config --global user.email "abcd1234@uni.sydney.edu.au"`

*Note: If you don't want to set it globally, remove the `--global`*

### The authenticity of host git.edstem.org can't be established

This is just your computer saying "I have no idea who git.edstem.org is, do you trust them?".

The answer in this case, is most likely `yes`.

This happens when you're connecting to a new website your computer doesn't know.