GitLab Walkthrough 0
Setting up SSH Authentication

Mark Eramian and Anthony Kusalik

November 23, 2020

1 GitLab

GitLab is a web-based hosting package for remote Git repositories. The Department of Computer Science maintains its own GitLab server, `git.cs.usask.ca` which you will be using. Before performing any other tasks with `git` we are going to walk you through how to set up an SSH key at git.cs.usask.ca. This will save having to continually provide passwords when accessing the remote repository. SSH keys are a public/private key authentication system that is an alternative to passwords. You provide the server with your *public* key. Then when you try to sign into the server, the server asks you to provide the *private* that matches the public key. If they match, you are authenticated. This method of authentication is less practical in many situations than passwords, but it is more secure. It is also the only way to avoid having enter passwords for git operations on a remote GitLab server.

2 Generating SSH Keys

Step 1: Generate SSH Keys

Setting up SSH keys on Mac OS X, LINUX, and UNIX are straightforward: run the terminal command `ssh-keygen`. The man page for `ssh-keygen` has a full description of the various options and arguments to the command. For our purposes, it is sufficient to run the following terminal command:

```
ssh-keygen -t rsa -C "<NSID>@<machine-name>"
```

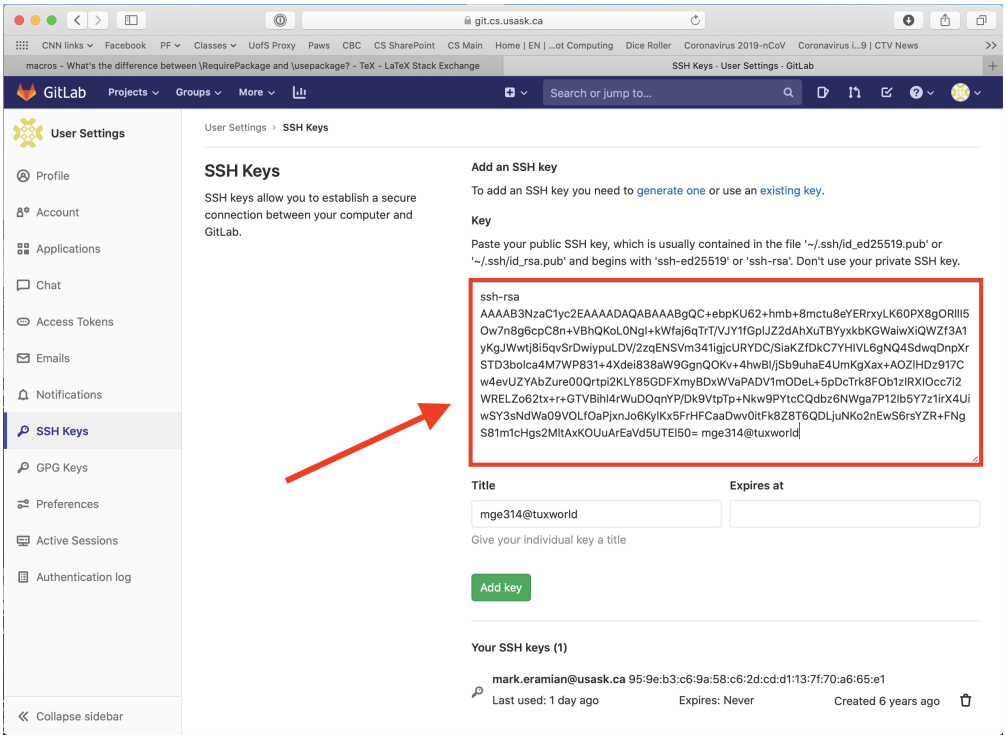
where `<NSID>` is your NSID, and `<machine-name>` is the name of your machine (it really donesn't matter what you put here). `ssh-keygen` will then prompt you a few times to enter some information. Just press "return" at each prompt, entering empty strings.

The command will create two files: `~/.ssh/id_rsa.pub` (this is your *public key*), and `~/.ssh/id_rsa` (this is your *private key*). Below is a log of `ssh-keygen` being performed:

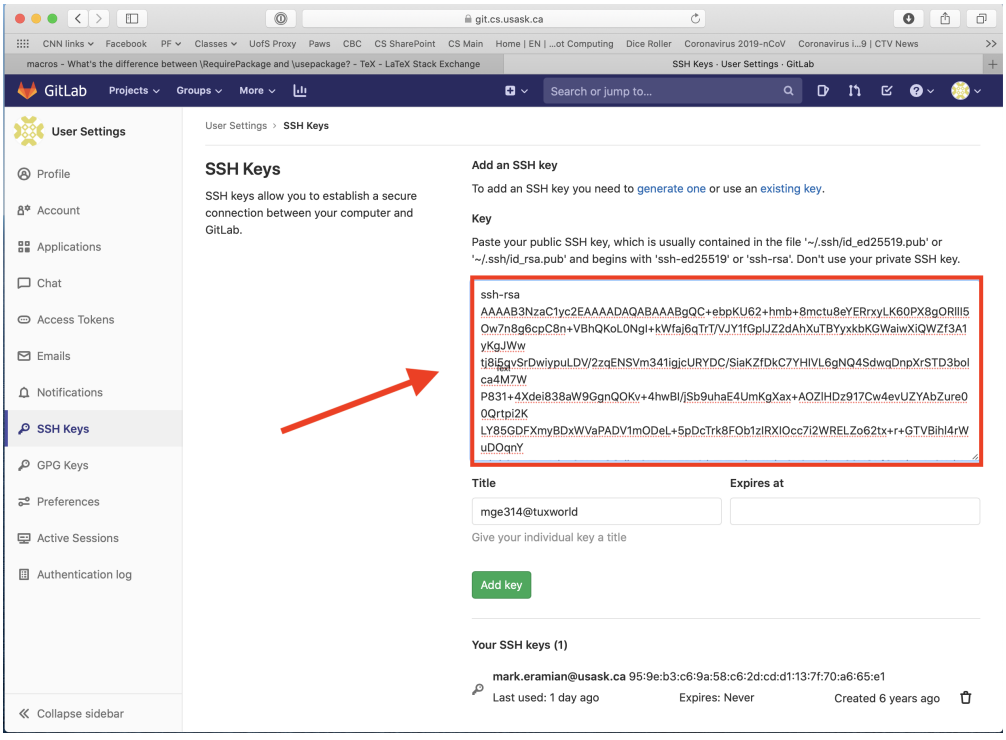
```
[eramian@tux6 ~]$  ssh-keygen -t rsa -C "mge314@tuxworld"
Generating public/private rsa key pair.
Enter file in which to save the key (/faculty/eramian/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /faculty/eramian/.ssh/id_rsa
Your public key has been saved in /faculty/eramian/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:PZr4fKhbU7LTrlP7jI3q0ecN0pIEtqhU/otTgmThVBo mge314@tuxworld
The key's randomart image is:
+---[RSA 3072]----+
|    E..          |
|    oo           |
|   o... o        |
|    +o o +       |
|   o..o S =      |
|   ....o.X.+     |
|    . .oXo*.+    |
|      .=o*oX o   |
|      +=B=+.= .  |
+----[SHA256]-----+
```

Step 2: Add Public SSH Key to GitLab

Once you have SSH keys created, use a web browser to log in to `git.cs.usask.ca` using your NSID and password. There is a pull-down menu under the avatar in the top right-hand corner of the window. Select "Settings". In the resultant window, notice the set of icons along the left side of the window. Click on "SSH Keys". Paste in your public ssh key (the contents of your new `~/.ssh/id_rsa.pub`) in the text box on the right. It should look something like this:



Be careful that the cut-and-paste operation did not wrap lines and, as a consequence, introduce newline or carriage return characters. If pasting resulted in any newlines you need to delete them. Here's a case where newlines were added that needs to be fixed:



Note that the first line consisting of `ssh-rsa` is on a line by itself only because there is a space between it and the key itself. There is no newline there, and it should be a space.

Once you have pasted the public SSH key, click the green **"Add Key"** button.

Remarks

Note that if you will be accessing the remote repository from multiple machines you can set multiple ssh keys (but you don't have to). Now that SSH keys are set up you will be able to use URLs of the form `git@git.cs.usask.ca:<NSID>/<repository-name>.git` where `<NSID>` and `<repository-name>` are replaced as necessary/appropriate.