**MACOS Git Setup & SSH connection to GitLab**

The aim of this document is to provide the user instructions on how to connect install Git and get connected to the GitLab repository storing all the course material. If you are not familiar with using Git please take time reading through [Git Reference Manual](https://git-scm.com/docs) as this will show you how to clone, pull, checkout, commit and push your files to and from GitLab. Everything you need to know about GitLab can also be found on [GitLab University](https://docs.gitlab.com/ee/university/).

We will not expect you to push anything to GitLab during this course. The goal is to get you set up and familiarized with Git and GitLab, by slowly introducing you to its features and uploading course material which you will be expected to pull at the start of every lesson.

Once you have pulled your information please store those files in a new, separate folder as any modifications to you pulled notebooks within that local repo, will be reflected as changes that you could potentially try to commit and push to the GitLab repo. This means that you should not open and modify the notebook directly in the local repo, just copy/paste them into your new working folder/directory that you will manually create.

Lastly; if you are having trouble setting up git, generating SSH Keys and connecting your local repo to the MLG GitLab repo (which means you did not follow the instructions correctly: yes; this is always the case) then you can always simply log into GitLab and download the material directly using the web UI.

NOTE: *GitHub and GitLab are both web-based Git repositories but, they are slightly different. You can think of a GitHub as an open shareable web-based Git repository, whereas GitLab has features that allow administrators to control access and keep the public out. Again, we will be using* ***GitLab****. For more information on what Git is and its relevance to GitLab and GitHub please read this* [*post*](https://usersnap.com/blog/gitlab-github/)

Definitions:

Local Repo: Simply a folder you create on your machine that will be the location to where pulled (downloaded) files and data will land when you type the ‘pull’ command.

GitLab Master Repo: The online storage bucket of the course material that you will be downloading.

**INSTRUCTIONS:**

**STEP 1: Install Git and Create GitLab Account**

1. Please ensure you have updated your MAC OS to Mojave Version 10.14.6
2. You should have received an invite to join the MLG GitLab. If you haven’t already created a [GitLab](https://about.gitlab.com/) account, please do so using you Deloitte email address. Create a secure passphrase and don’t forget it.
3. Please ensure you have [Git](https://git-scm.com/) installed on your system:
   1. Open Mac Terminal & type ‘git status’ and hit enter. If Git has not been installed MAC is clever enough to know you need the developer tools installed and will prompt you to install the developer tools. Click yes and wait for the developer tools to be installed. Here is a screen shot of what you should see if you have installed developer tools correctly.

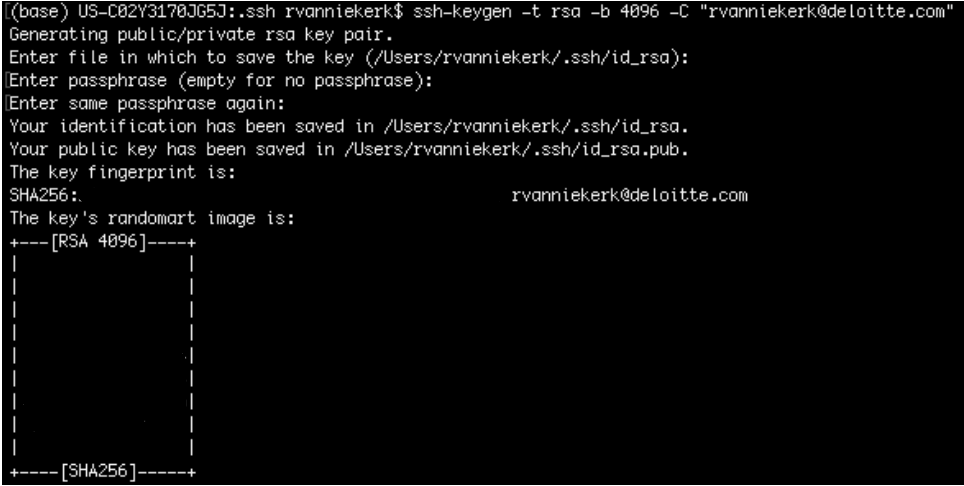


* 1. If you already have Git installed, you may get a print out of files waiting to be committed or nothing depending on whether the folder you are executing the cmd from has Git initiated in that specific folder.

**STEP 2: The next set of instructions actually exist online** [**Generating a new SSH key pair**](https://gitlab.com/help/ssh/README#generating-a-new-ssh-key-pair)**. I will step you through this below as well, but I’d say open this above link ^ and find the instructions from here as well. They are very detailed and if you follow the instructions exactly as described, you should not have any issues. You will be generating an ‘RSA SSH’ key and using that to link your local repo to the MLG GitLab. The details below are to help guide you through these steps:**

Open terminal on your Mac and enter the following chunk of cmd line code (please be sure to use you own Deloitte email address that you used to create your GitLab account). This will generate an SSH key:

* 1. ssh-keygen -o -t rsa -b 4096 -C "<your\_name>@deloitte.com"



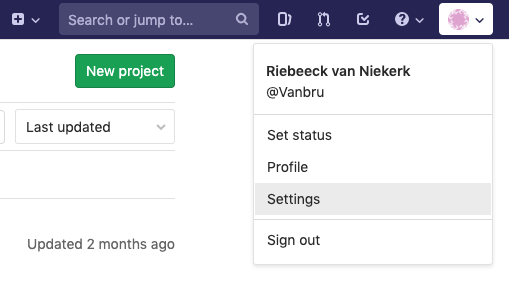
* 1. You will get a prompt asking you to save the rsa ssh key somewhere. Just use the default location by hitting ‘enter’! Please note that the default location is a hidden folder in your home directory.
  2. You will get a second prompt asking you to ‘Enter passphrase’ twice. This is important to remember as you will use this each time you want to push or pull data from the GitLab repo. I suggest using the same passphrase as the one you use to log into GitLab with, this way you don’t have to remember a bunch of different passwords depending on whether you’re logging into GitLab or pushing/pulling data from GitLab using your Git by running commands in the command line interface.
  3. Run the following cmd line code in your terminal. This will list all the files contained within your .ssh directory: $ ls -al ~/.ssh
  4. You should see and id\_rsa and id\_rsa.pub file listed along with any other SSH keys you have previously generated and saved in this default location. Please note your screen will obviously not look exactly the same since you are not ‘rvanniekerk’ (that’s me). You will see your computer’s <user\_name> printed there…



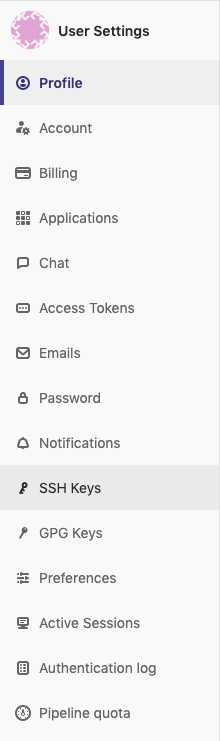
* 1. Run the next cmd line code in terminal to copy your id\_rsa.pub contents to your clipboard. You will not see a printout but if you now hit cmd+v (paste) you should be able to paste the contents to a file or entry field. Keep this last step in mind as we move on to step 3 below.

**STEP 3: The next step is simple and requires you to log into your GitLab account and paste your copied id\_rsa.pub file contents to the SSH Key empty text field. Please see the screen grabs below for navigating to the location on GitLab to where you will paste and create a new ssh key ‘linking’ it to your local system.**

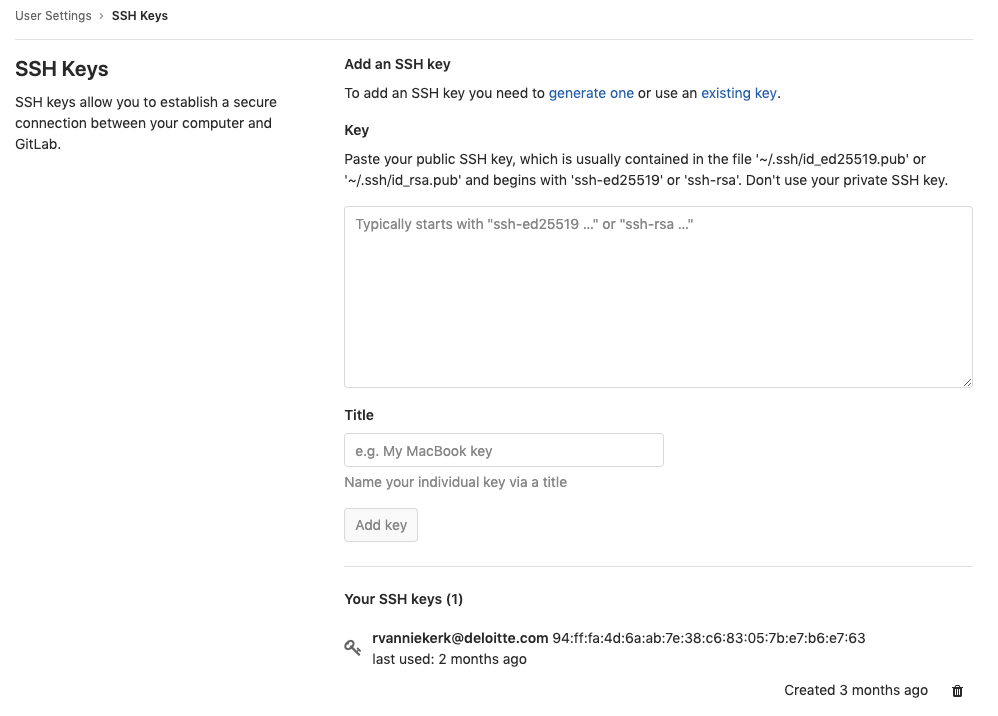
1. Log into GitLab and navigate to the top Right-hand corner, click on your avatar drop down and select settings:



1. Once in ‘Settings’ navigate to ‘SSH Keys’ on the Left-hand selection pane.

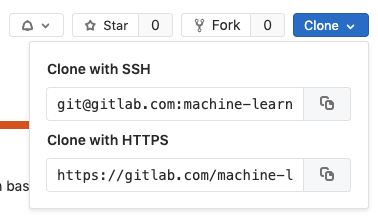


1. You will now see an empty text box. This is where you must paste your id\_rsa.pub file contents (remember you’ve copied it to your clipboard. If not, go back to step 2.6). Give it a title and click on the ‘Add Key’ button. Once added you will see your new key saved to your profile.



**STEP 3: Clone the GitLab repository to a local folder on your Machine. Again, these instructions can be found** here GitLab **and it would be advised to follow along with those if you are running into issues:**

1. Clone the Gitlab Repository:
   1. Open GitLab & open the mlg-05-us folder (this is the folder you want to clone).
   2. You will see a ‘Clone’ button on the top right-hand corner of the screen. Click on that and copy the contents of the clip-box under **‘Clone with SSH’.**



Please note; it possible to also clone with HTTPS. It’s just less secure and would require you to enter your password each time you want to pull or push files to the GitLab repo. We advise against this.

1. Navigate to your terminal and run the following command line clone IN THE FOLDER THAT YOU WISH TO USE AS YOUR LOCAL REPO. This will be the location to where you will clone all the course data and pull into on a daily basis during the bootcamp.
   1. git clone git@gitlab.com:<folder\_name\_in\_gitlab\_stuff>
2. Once cloned you should be able to run git status & git pull to check if everything is working as expected. Please note that you may still be expected to enter your password when you want to pull file down form the repository.

If at this point something is still wrong. Please remove the ssh key files (id\_rsa & id\_rsa.pub) you created from the ~/.ssh directory and delete the SSH key that you’ve saved on GitLab and go through the entire process again. It’s most likely that you skipped a step or did not follow prompts or instructions correctly.

There are a ton of great YouTube videos showing you how to do this as well but be sure to follow these instructions along with the information provided on [GitLab University](https://docs.gitlab.com/ee/university/) and specifically here [Generating a new SSH key pair](https://gitlab.com/help/ssh/README#generating-a-new-ssh-key-pair) before you try this on your own.

Lastly; if you are still having issues, we will set up skype session on the afternoon of Friday, September 13th where instructors will sit down with students who are having issues to assist in getting them set up git Git and GitLab. Please look out for this invite if you are having any trouble.