Assignment Management with Git(Lab)

Git(Lab):

In this course, we will be using git and GitLab for assignment distribution and collection. If you're unfamiliar with version control systems like git, there are tons of online resources from major git organizations such as GitHub and Atlassian. Here are some great references to get you started with version control and git:

https://www.atlassian.com/git/tutorials
https://git-scm.com/book/en/v2
http://docs.gitlab.com/ee/gitlab-basics/start-using-git.html

It is **strongly** recommended that you go through the full git tutorial here before starting your first assignment: https://try.github.io

The course GitLab site can be reached at https://osgit.rnet.missouri.edu.

You should be able to login with your university credentials. Please be sure to check that your account is enabled at your earliest convenience. If you haven't signed up for the course yet, or have only done so recently, you may not be in the system. If you are not in the system, you will most likely receive the message

Your account is blocked. Retry when an admin has unblocked the account.

when attempting to login. Be sure to email our TA Jason with a request to be added.

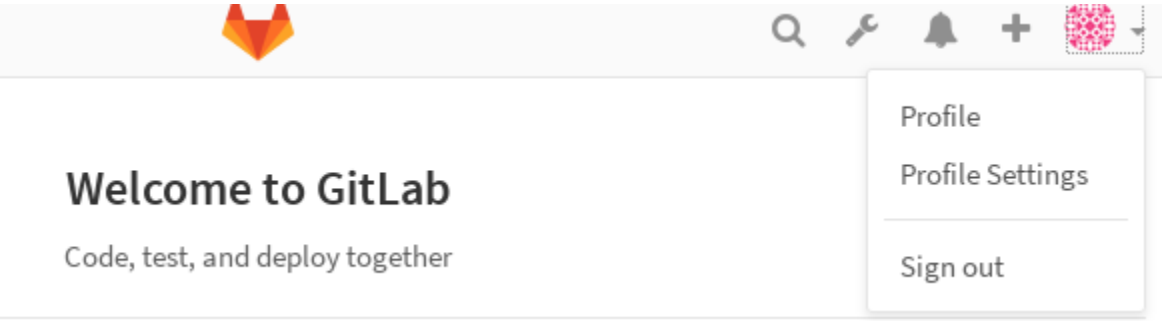
To: jason.james@mail.missouri.edu
Subject: Please add my pawprint to osgit
Message:

Please add <your_pawprint_here> to osgit.
Thank you,
<your name here>

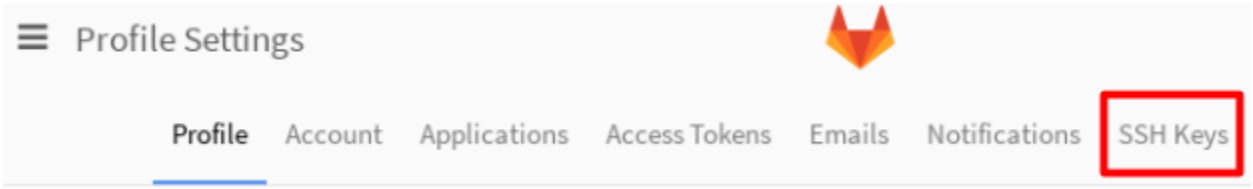
Account Setup:

You must add an SSH key to your account and use SSH URLs for interacting with remote resources. You should be able to reach the SSH key settings page here https://osgit.rnet.missouri.edu/profile/keys or follow the steps below:

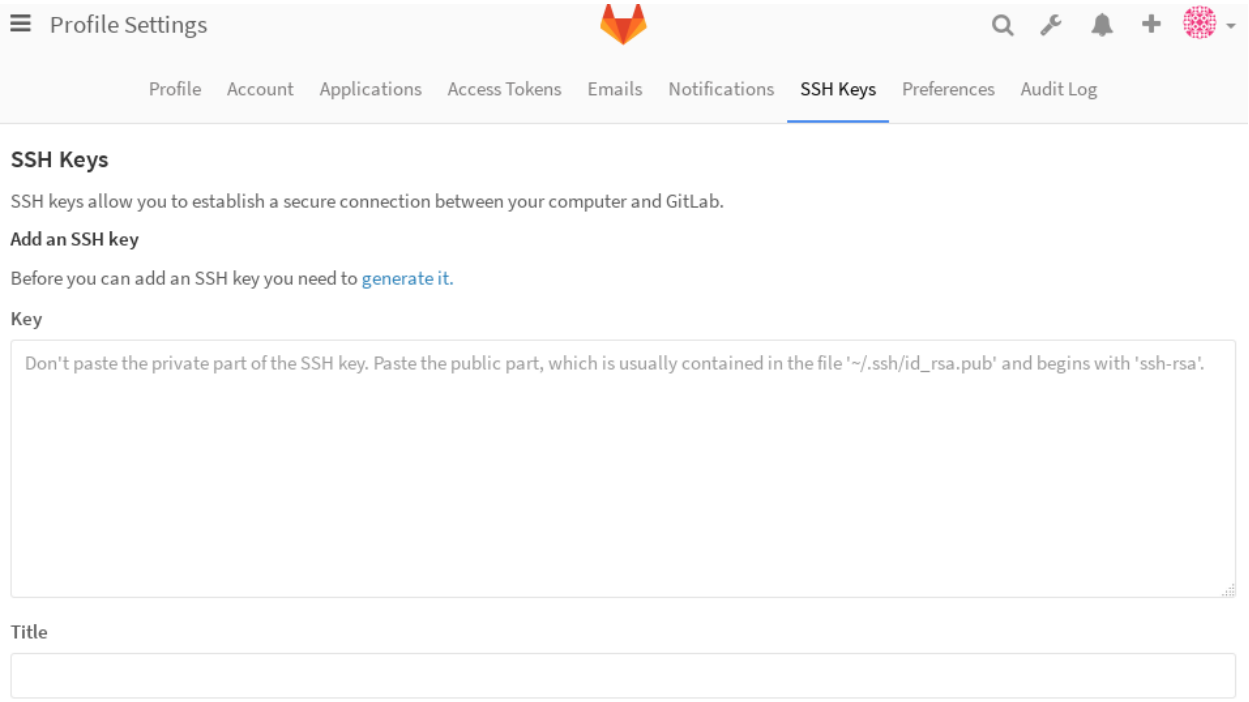
Go to Profile Settings



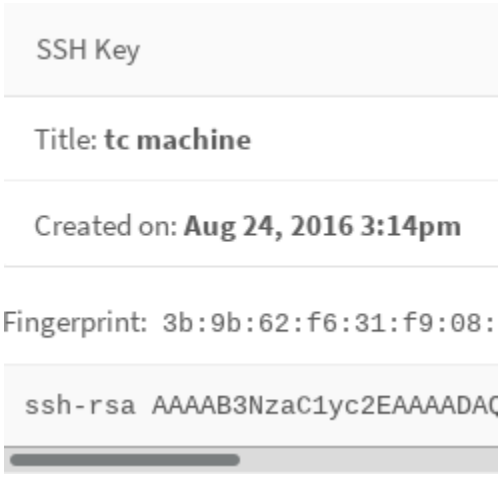
Select the *SSH Keys* Link

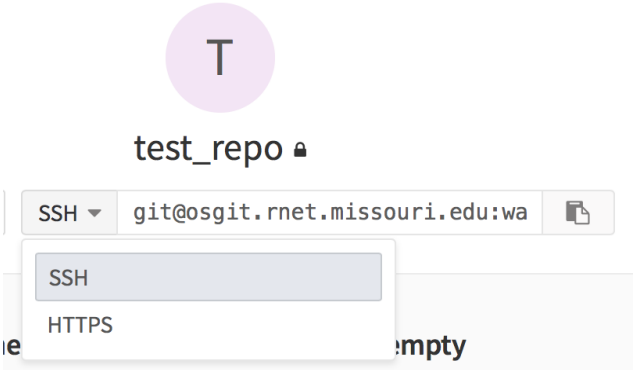


Add your SSH Key
See the Programming Environment Canvas Page for creating your SSH key.



Once you have pasted your key into the box, and added a title for the key, click the **Add key** button.





Once you have a key added to your account, you should use SSH URLs instead of HTTPS URLs when working with remotes. SSH URLs begin with ‘git@’.

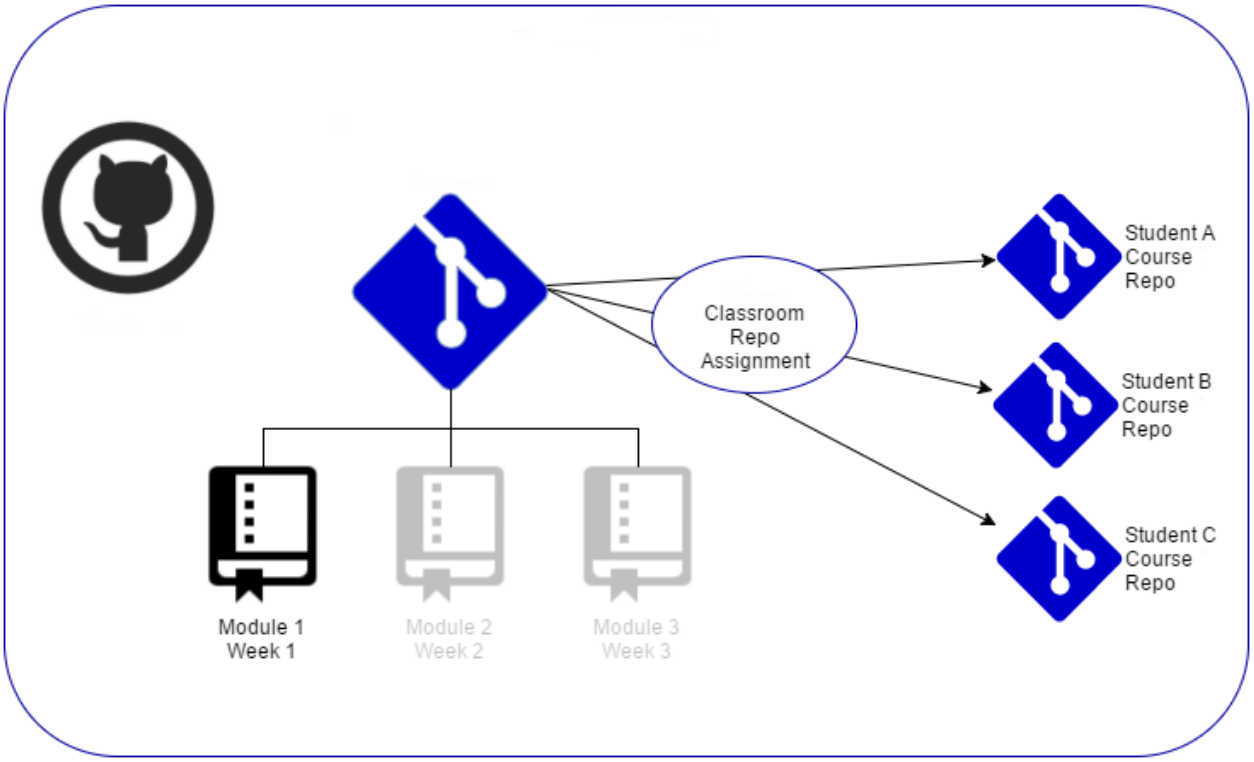
Note that you may want to add keys for all machines you will use to connect to OSGit. For example, I have keys for tc.rnet.missouri.edu as well as my Windows desktop and my Windows laptop machines.

Graphical GIT Client

A graphical git client is also recommended if you are not yet comfortable with the occasionally nasty git commands, or want additional features. Atlassian provides SourceTree for OS X and Windows, Auxosoft provides GitKraken, which works on OS X, Linux, and Windows, and GitHub provides GitHub Desktop for OS X and Windows.

Assignment Distribution and Collection:

When an assignment is created, you should receive an email letting you know that you have been granted access to a repository. Usually you will receive two, one granting read access to the master repository, and one granting developer access to your personal repository (it will have your pawprint in the name). The master repository, often referred to as an upstream, will be where you will receive updates from as needed. Your personal copy is where you will do your work and push your completed code.



When the assignment is due, an automated system will collect your work for grading. Please note that only the **master** branch will be recorded, so be sure that you have merged any work branches into master before the due date.

Assignment Workflow Example:

Let's pretend that the first assignment has been released. You should receive two emails from GitLab, the first granting read access to the master repository, os/f20/assignment1, and one granting developer access to your personal repository, os/f20/assignment1_**xxxxxx** (where **xxxxxx** is your pawprint).

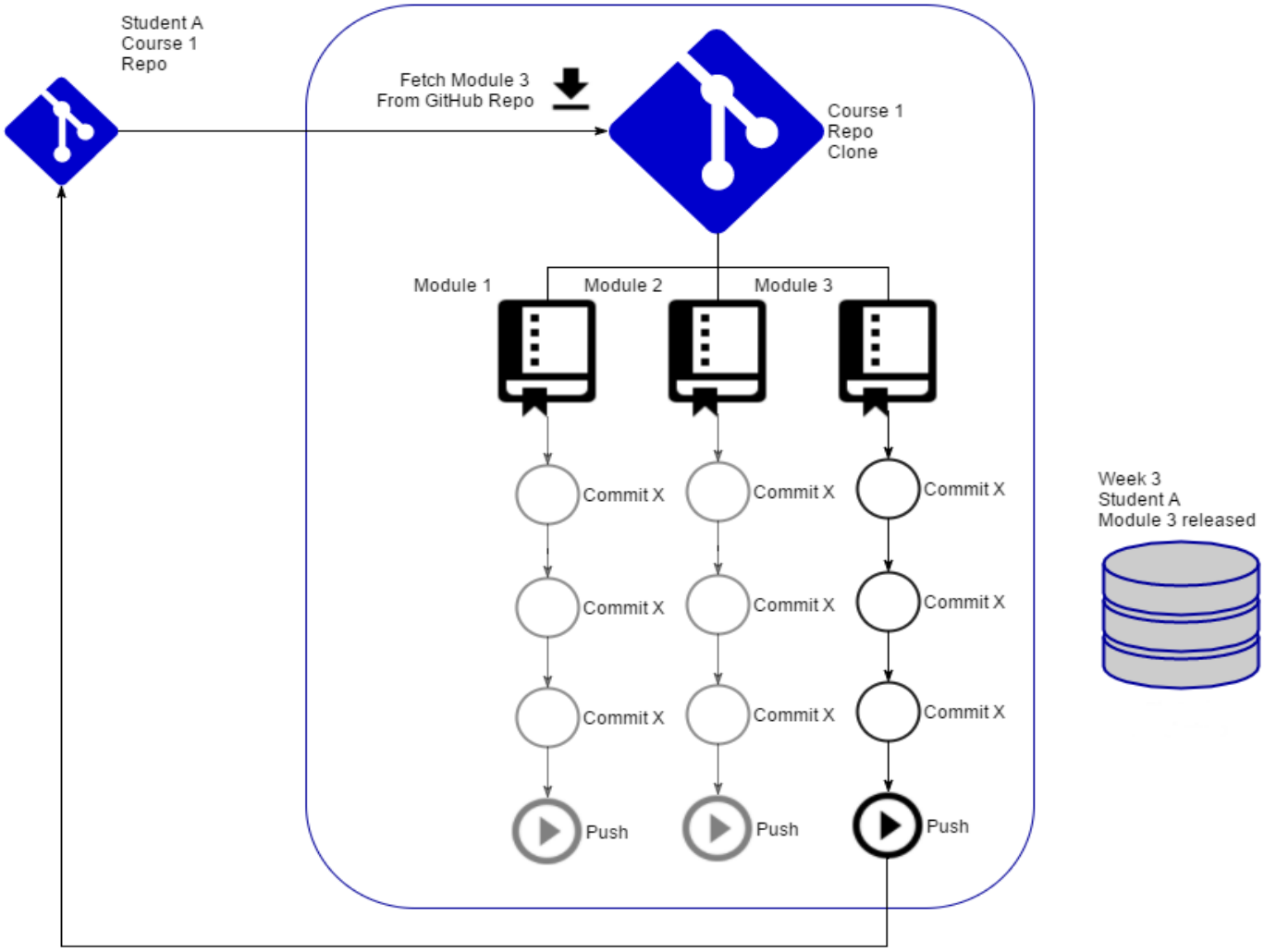
When you are ready to begin, you should clone your personal repository and add the master repository as an upstream remote. You can do this in cone step with

`git clone --config remote.upstream.url=git@osgit.rnet.missouri.edu:os/f20/assignment1.git git@osgit.rnet.missouri.edu:os/f20/assignment1_xxxxxx.git`

if you're feeling adventurous.

Many assignments will be broken up into multiple milestones, and it's suggested that you do all your development inside a branch until you are ready to submit. Here we will use the branch milestone1. You can create and switch to a branch with git checkout -b milestone1. New branches won't be sent to the remote repository until you push them. To push a new branch up to your personal repository, use git push -u origin milestone1. This will create the new branch at the remote and configure your repository to use it. This only needs to be done once per branch, after which you can push commits with git push.

After many commits, pushes, pulls, and debugging, you are ready to submit your work. All you need to do is merge your branch with upstream and push it back up to GitLab. First, checkout the master branch with git checkout master, merge with git merge milestone1, fill out the commit generated, and push with git push.



Getting Help:

Git can be hard. If you ever have trouble, GitLab provides powerful tools for managing git. You can create branches, edit and add files, merge branches, and even fix conflicts from your web browser. Graphical git clients often provide similar features and will often check for upstream updates for you. Don’t forget that Google exists! If you’re having a problem with git, there’s a 99.999% chance that at least one million other people have experienced the same problem. If you are still having issues, try posting on the git discussion board on canvas, other students or TAs may be able to help with your issue.