

Lab 0: Learning to use Git

Due Friday, 5 October 2018 11:59 PM

Minimum Submission Requirements

- Create a Lab0 folder (note the capitalization convention)
- README.txt in the Lab0 folder
- Commit and Push your repo to the GitLab server
- From the GitLab web interface, tag the commit that you would like to be graded
- All of the above must be completed by the lab due date

Lab Objective

This lab will introduce you to version control (Git) and how to use it within the context of this class. You will be copying and including both the personal responsibility and the class syllabus documents and checking them into your repo as verification that you have read them. Version control is an extremely important part of engineering; it is the basis for most engineering projects. Git is an industry standard tool for version control, and you should know how to use it as part of your engineering education (for example, Microsoft Windows development is version controlled using Git).

Note

If you are not able to complete this lab, you do not have the prerequisite knowledge to succeed in this class. If this is difficult for you, you will be better off taking a different class and coming back to CE12 when you are ready. Failure to meet the minimum submission requirements listed will result in failing the class regardless of any other scores. This will be true for all subsequent labs as well.

Git Setup

To complete this lab you need to set up a GitLab account by going to https://gitlab.soe.ucsc.edu/gitlab/users/sign_in and following the directions to create your account. DO THIS NOW as there is a delay between signing up for an account and getting a repository. The repository (or repo) is the set of files that Git is keeping track of.

Lab Procedure

Once you have your GitLab account and a repository (repo) for the class, you are going to verify that you have read both the class syllabus and personal responsibility documents. You will be “digitally signing” the documents by creating a README.txt to have your name in it.

Due to the size of the class, this and other labs will be machine graded. That is, a program will run that will check the files and generate your grade. It is extremely important to match the names and directories/folders exactly, as that is how the grading program needs to find your work. As in all engineering, attention to detail is paramount. Stories abound where small errors lead to catastrophic failures; in this class it will only lead to a failing grade, but you might find that catastrophic. Pay attention!

While these requirements may seem somewhat arbitrary, note that they flow from previous problems with the class (and other classes at UCSC) especially in the realm of academic misconduct. This lab serves to demonstrate that you are aware of what we expect of you such that there is no misunderstanding about what does and does not constitute improper use of material.

Steps

1. Register for an SOE GitLab account:
https://gitlab.soe.ucsc.edu/gitlab/users/sign_in
2. Read the git_cmpe012 document available on Canvas - this goes over the basics of Git and how to use it.
3. Clone your repository.
4. Create a Lab0 directory in your repository.
5. From canvas, download copies of the syllabus and personal responsibility document (personal_responsibility_cmpe012.pdf).
6. Read both documents carefully.
7. Create a file in the Lab0 directory called README.txt with your name - this is the equivalent of your digital signature and indicates that you have read the two files. Make sure your README has the .txt extension. Some operating systems omit the extension by default.
8. In README.txt, add the current date, and a sentence like "I, Homer Simpson, have read and understood the CE12 syllabus and Personal Responsibility Document." Of course, you should use your own name. This constitutes a digital signature of the document.
9. Add the README.txt to your Git repository (see the git_cmpe012.pdf document for more detailed instructions).
10. Commit your repository.
11. Push your commit to the server.
12. Go to the GitLab web interface and verify that the files have the right names and are in the right folders - i.e., verify that you have met the minimum submission requirements for this lab.
13. Tag the commit from the GitLab web interface (see below for tagging instructions).

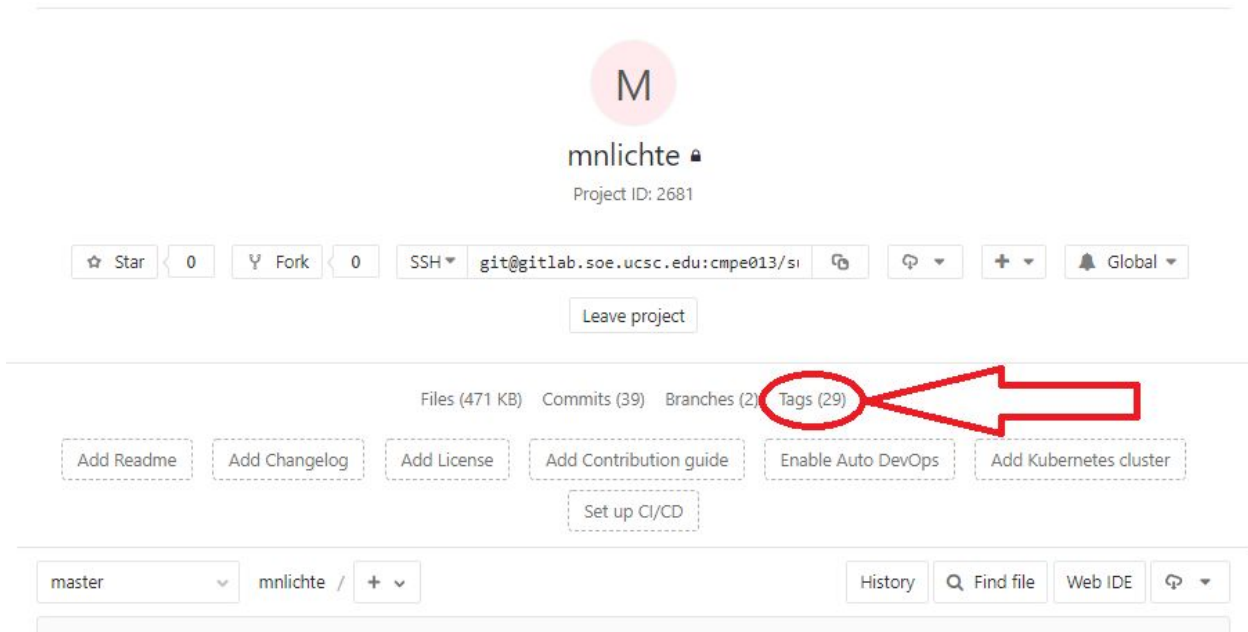
Tag your commit

In CMPE12, each assignment is submitted in the form of a git commit. A repo should have many commits, so you will have to tag a commit to indicate that it is the commit you would like us to grade.

Tag labels must be in the following format: *Lab0_submission_#* where # is any positive integer value. Every time you update the tag for the commit you would like us to grade, increment this number. For instance, I might tag a commit as *Lab0_submission_0*. Then the next day I modified a file, so I add a new tag to the new commit labeled *Lab0_submission_1*. We will grade the submission with the highest #.

Be careful with capitalization, spelling, and underscores! Our grading script will ignore tags like *lab0_submission_100*, *Lab0submission12*, and *Lab0_Submission_5*.

To tag a commit, log onto the GitLab web interface and click the "Tags" link:



From there, click the “New Tag” button. Under “Name,” write Lab0_submission_0 (or Lab0_submission_99 or any other valid submission tag). Usually, you will leave the “Create From” field set to “Master,” but you can also paste the commit ID of an earlier commit, if you like.

When you’re done, make sure you can see your new tag in the tag viewer, and check to make sure it has your work.

Notes

This lab is quite short, but if you are unfamiliar with Git it can take some time. Start early and there should be no issues with finishing it well before the due date. This lab is really a check on the prerequisite knowledge required to succeed in this class; **if you cannot complete this easily, we strongly recommend you take a different class and come back another term when you are ready.**

This is how you will submit **ALL** labs in this class, and they will all have a similar set of minimum submission requirements. It is important that you learn to do this right at this point in time. You can tag as many commits as you’d like, but we will always grade the one with the greatest number in the suffix.

The tag determines when your assignment is submitted. Even if you committed your files locally or pushed them to the server, they are not considered “submitted” until you tag the commit.

While this process may seem arbitrary, complying with the requirements is a very big part of engineering (regardless of which flavor you are learning). There is a lot of help available to you in the form of lab sections, TA and instructor office hours and we will go over this in the lecture as well.

No further work in the class will be graded unless this lab is completed; this is our verification that you have been informed of our expectations of you. Following

instructions carefully may seem arbitrary (to you) but is absolutely necessary both from an engineering and pedagogical viewpoint.

One final note

You should get in the habit of committing early and often. This is very easy to do, and guarantees that you have something turned in. Every time you make a change, add a file, modify a file, go ahead and commit and push. You will always be able to return to that place and never lose your work.