Git Setup Help using GitKraken (CSE 154)

Introduction:

Git enables keeping track of different versions of files as we keep editing them. To make sure we understand git properly, here are some terms you'll see being used quite often:

1. **Repositories:** Just a fancy way of saying a collection of files/folders
2. **Branch:** Since large projects can have multiple people working on the same set of files, a branch is basically a way in which users can keep uploading and saving their work individually. A branch is essentially a way to keep different portions of a project separate from each other, so they can be combined together later.
3. **Master Branch:** This is the main branch that git creates for you and saves all your code to (unless you specifically want it to go to another branch).

There are other terms that git uses that we'll talk about below:

NOTE: DO THIS ON A MACHINE YOU OWN OR HAVE ACCESS TO THROUGH THE REST OF THE QUARTER.

Step 1: Download GitKraken

Small caveat (**Mac users**):
If at some point, the OS asks you to download xcode to get git, we do not recommend doing that in lab since that can be very time consuming. Instead download and install git using https://git-scm.com/download/mac and skip xcode downloads until later. Following this you can continue using GitKraken as specified below.

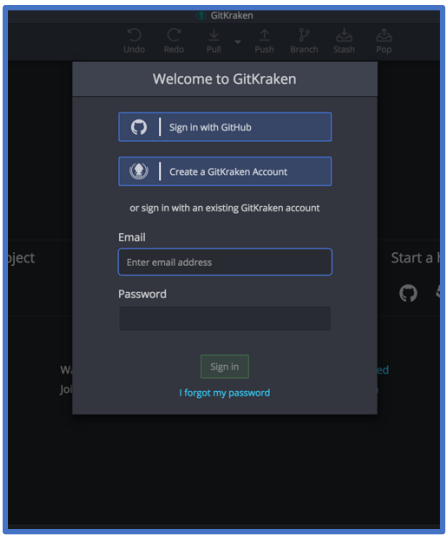
If when running git commands, you see the message: 'The "git" command requires the command line developer tools. Would you like to install the tools now?"

type(in terminal) : "`sudo mv /usr/bin/git /usr/bin/git-system`" and hit enter and things should work fine.

Step 1.a: Go to https://www.gitkraken.com/ and download the application onto your machine. (It's relatively large so this could take some time).

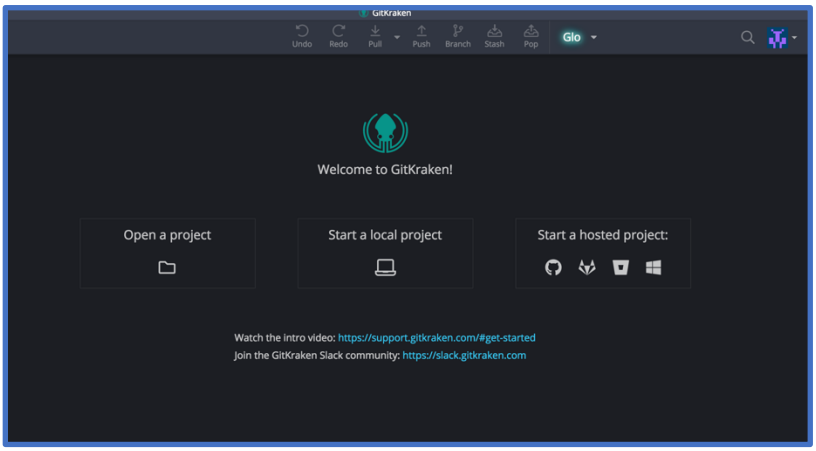
Step 1.b: Double click on Gitkraken to start it up. You can find the application once it has downloaded by searching for it, by either using the spotlight search (Mac) or the search bar located in the bottom task bar (Windows).

Step 1.c: Once it has loaded up, you should see this screen:



Step 1.d: If you already have a GitKraken account, you can log in using that or you can create a new GitKraken account using your email address and password. (This does not have to be the email/password that you'll later use for Gitlab).

Step 1.e: Once you've finished making an account and verifying it (via email), you should hopefully see this screen:



We can now move onto setting up our Gitlab account and getting ready to pull files onto our local file system.

Step 2: Setting up GitLab

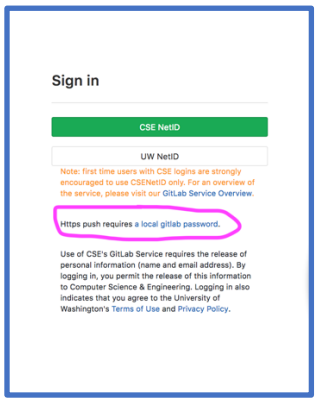
Step 2.a : Go onto Gitlab: https://gitlab.cs.washington.edu/users/sign_in

Step 2.b: Click on 'Sign in with UW Net ID' or 'CSE ID' (if you have one). Fill out your information and you should now have a Gitlab account!

Step 2.c: Once you have your account, sign out of Gitlab. You can do this by clicking on the drop down list on your profile (top right of the website).

Step 2.d: Once you're logged out, you should now be on the page you started out with before in Step 2.a

Step 2.e: Instead of signing on, click on 'Https push requires a local gitlab password.' shown here in the pink box



In the window that opens, enter the **email you used to create your Gitlab account.**
Click on 'Reset password' and it should tell you it will send you a link on your email to reset your password.

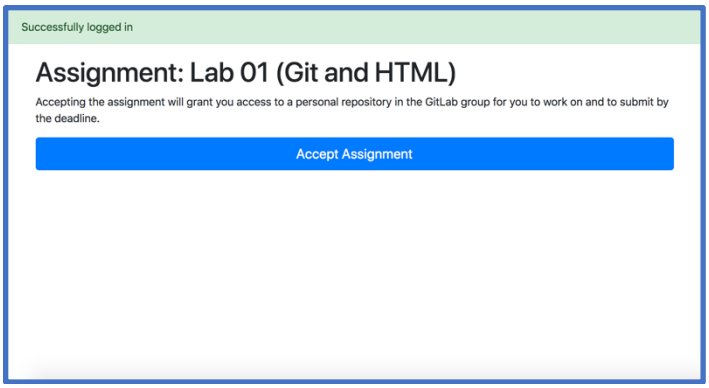
Follow the steps to set a **password THAT YOU WILL REMEMBER. OR NOTE IT DOWN**. You will use this email and password every time you want to get a new homework or submit your homework!

Once you've finished this and set a 'local password' you are now ready to start lab 1.

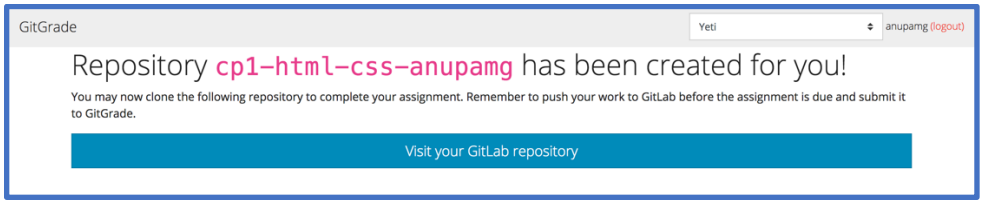
Step 3: Accepting the Assignment

Step 3.a: Go onto https://oxford.cs.washington.edu/gitgrade/student/assignment/3 and log into your Gitlab account.

Step 3.b: Once you're logged in, you should see something similar to this:



Step 3.c: Click on Accept Assignment and you should see something similar to this(with a different repository name):

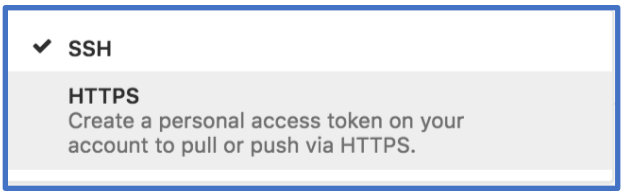


Step 3.d: Click on 'Visit your Gitlab repository' and that should lead you directly to the repository created for you!

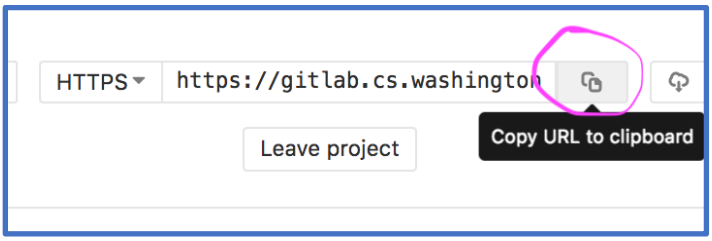
You should now see something similar to this:

Step 4: Now we're ready to get files onto our local computer.

Step 4.a: Click on the dropdown option shown in pink in the previous screenshot and click on the HTTPS option.



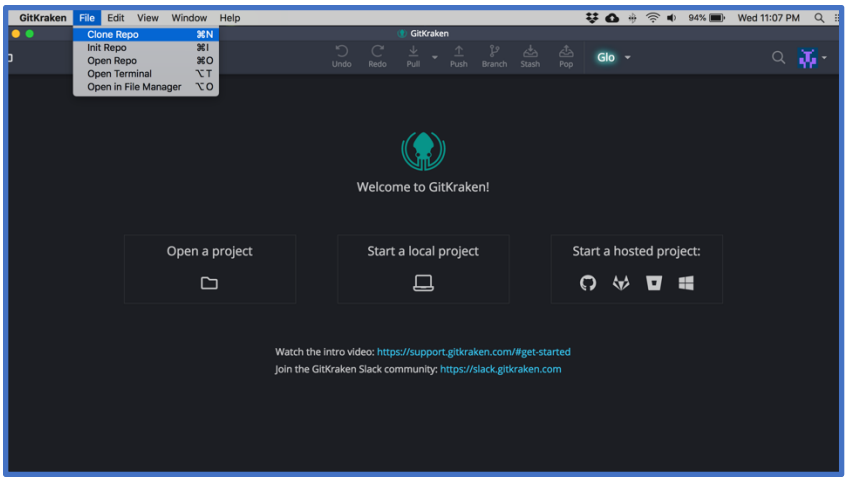
Step 4.b: Now copy the link shown onto your clipboard using the option shown in pink below:



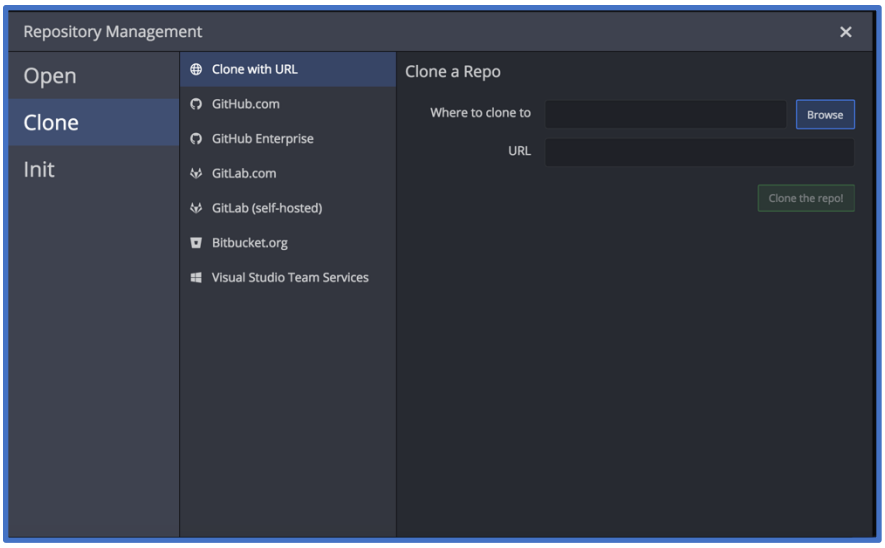
Step 5: Cloning the repository you just created

Step 5.a: Open up GitKraken again and look for the File option located on the top left.

Step 5.b: Select the “Clone Repo” option in the drop down menu:



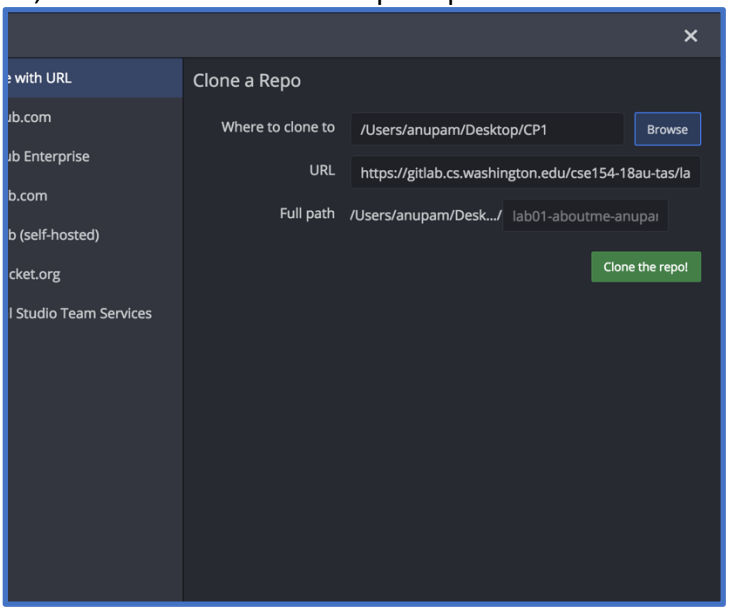
Step 5.c: Click on the “Clone with URL” option in the window that opens up:



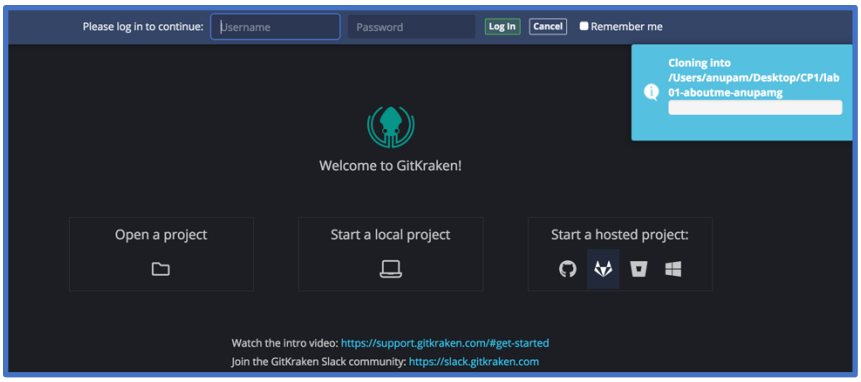
Step 5.d: In the "Where to clone to" option on GitKraken, Browse to select where you would want your files you see online be copied.

Step 5.e: In the URL input box, input the link you just copied in Step 4.b from Gitlab

Step 5.f: Following this, click on the "Clone the repo" option that will become active:

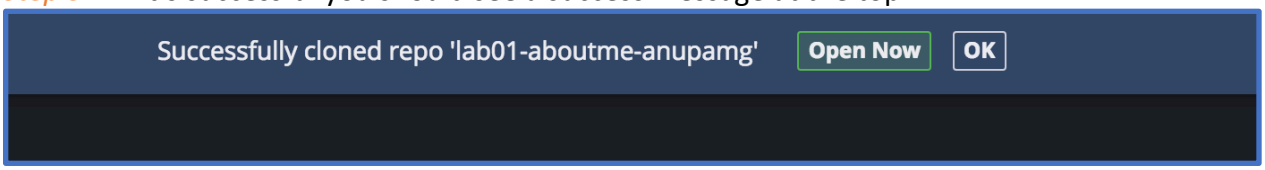


Step 5.g: This should now bring you to a screen that prompts you for your log-in username and password:



Step 5.h: For the username and password, enter the credentials you created in Step 2.e (scroll up) and then click on Log in.

Step 5.i: If it's successful you should see a success message at the top:



Step 6: Ready to work locally

Step 6.a: Click on "Open Now" and you should be able to see a very detailed commit log and several other things going on (which might be extremely confusing but no need to worry). We'll come back to GitKraken in a bit.

Step 6.b: We can now open up the folder which we had specified in Step 5.c and we should see that we've "downloaded" the files from Gitlab into our local machine now.

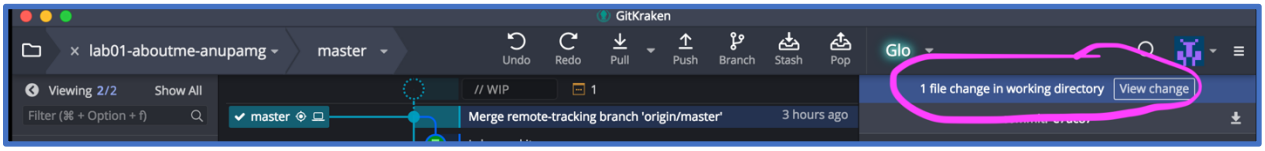
Step 6.c: Once you can see the files that were in your repository locally, you should be able to edit them. **For this lab, we just need to edit the aboutme.html page to fill in information about ourselves** **(Please read the Readme.md)**. To facilitate this, you could use any text editor you like. We recommend using Atom which can be downloaded and installed as specified here: https://atom.io/.

Remember to save your files when you're editing them locally using your text editor.

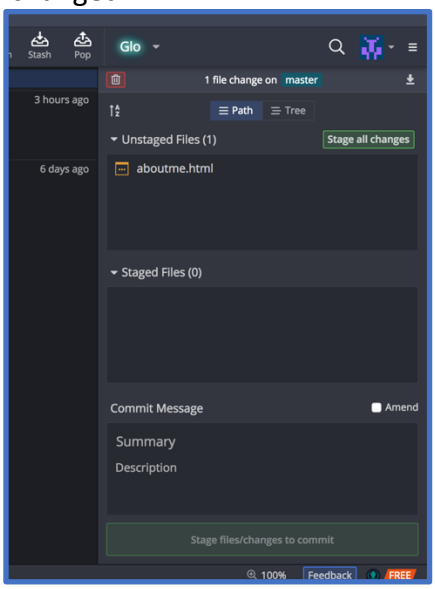
Step 6.d: Once you are done editing files or reached a spot during development where you think you should save your changes on your repository online, go back to GitKraken.

Step 7: Ready to Push to Repository

Step 7.a: When you go to GitKraken you should see something similar to this. For this example, I've made changes to aboutme.html so I get the message:



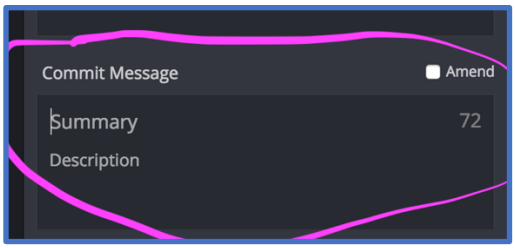
Step 7.b: Click on the View Change (shown in pink) option and the window on the right should change to show the files you've changed:

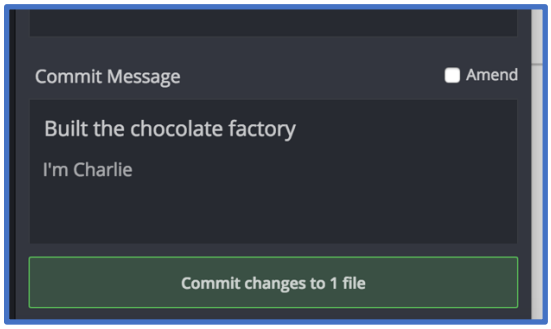


Step 7.c: You can either hover over the file you changed and select the Stage File option or select the Stage all changes option if you want to push all the files you edited up to your online repository. NOTE THAT IF AN ASSIGNMENT SAYS THAT SOME FILES SHOULD NOT BE CHANGED BY YOU EXPLICITLY, YOU DON'T WANT TO STAGE THEM IN THIS STEP. You only want to stage files that you should be changing. You can stage and push many times so you're more than welcome to push incomplete code on files that you should be changing. However, you shouldn't stage files that you changed locally but you were asked not to change. Although this is reversible, this can often be a lengthy, pain staking process.

Step 7.d: Once you stage the file you want to stage, it should appear in the staged files section of the column.

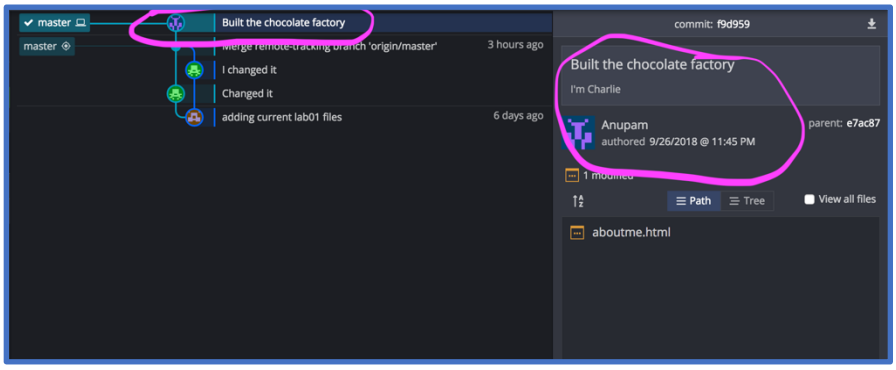
Step 7.e: Scroll down to the "Commit Message" portion of the page and enter a message that will remind you about what you've finished implementing/why you're pushing to your repository.





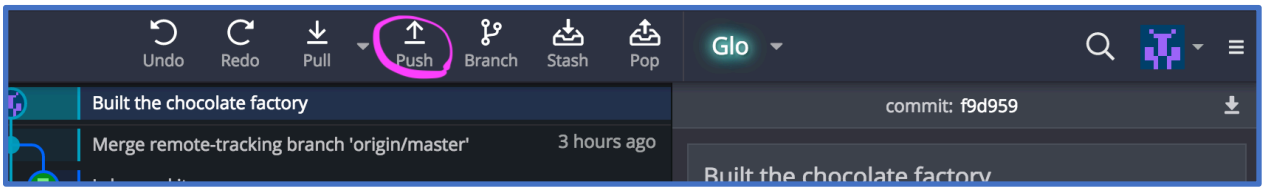
Step 7.f: Once you're ready, Click on the "Commit Changes to {n} file(s)" option.

Step 7.g: Once you're done, you should see the message you typed as a commit message on the commit log as well as on the rightmost terminal.



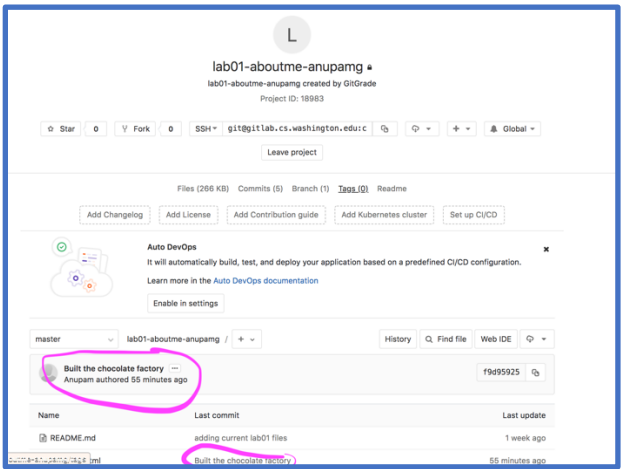
Step 8: Ready to push

Step 8.a: Now we're ready to push the commit we just generated into our online repository. To do this, click on the "Push" icon located in the top bar(shown in pink):



Step 8.b: Once you click on push, you should be able to see a success message that says “Pushed successfully master to origin”. This means you’ve pushed your changes to your online repository.

Step 8.c: You can check that all your changes have been pushed by going to your repository on Gitlab (same as the screenshot for step 3.d) and you should now see your commit message and changes (shown in pink)!



Step 9: Now that we’re all done, we just need to turn in the assignment to indicate we’ve finished what we wanted to do.

NOTE: ONLY DO THIS WHEN YOU’RE READY TO TURN IN (You can also turn in multiple times so no need to worry). YOU DON’T NEED TO DO THIS EVERY TIME YOU PUSH.

To do this, we go to the turn in link:
https://oxford.cs.washington.edu/gitgrade/student/assignment/3/turnin
And accept that we’ve turned in the assignment and we should be good to go!
