CS 346 Cloud Computing

Lab: Intro to Git and GitHub Classroom

Overview

You're going to create a GitHub user ID (if you don't already have one), create your own fork of the repo for this assignment, clone it onto Lectura, and then create a dummy file in the repo to show that you know how to commit.

1 GitHub User ID

We'll be using `git` to store and backup your code for this class - and also for turning it in to me. While you can certainly use `git` without GitHub, many programmers use GitHub, because it's an easy way to manage your repository. Go to `https://github.com` and create a user ID (or, simply log in if you already have one).

Of course, it might take a couple of minutes to get your user ID set up. If you're waiting on getting the email, why don't you start getting connected to Lectura at the same time? Look below for information how...

NOTE:
In the end, if you are having trouble, it's OK for only one of you to do the assigned work today (since you'll be turning it in jointly). But make sure to get it fixed ASAP, since you will need to turn in your own separate files as part of the pre-Project!

2 CS User ID

Everybody needs to have a CS username, which you can use to log into various CS department services - the most important of which is our shared UNIX server, Lectura.

Your CS username is the same as your NetID: for example, my university email address is `russelll@email.arizona.edu`, and my CS username is `russelll`. However, it's important to note that your CS password is **different** than your NetID password.

If you already know your CS password, you can try to log on to Lectura immediately (see below). If you don't (if you've never used it, or you've forgotten your password) then:

- Go to `http://www2.cs.arizona.edu/computing/services/`. Log in using your NetID (**not** your CS password!)
- Follow the link to "Reset my forgotten Unix password"
- Watch your email for a reply (should come relateively soon) giving you a **temporary** password
- The first time that you log onto Lectura, it will demand that you change your password; pick a good one, and write it down so that you don't lose it!

3 Using ssh

ssh is a standard protocol that computers use to access each other remotely. It basically functions as a "remote keyboard," but encrypts all of the traffic, in both directions, so that no one can monitor or steal your information.

You will need a program on your computer to use ssh. If you're on Mac, Linux, or the Windows Subsystem for UNIX, then you can do this right on the command line:

```
    ssh your_CS_username@lectura.cs.arizona.edu
```

If you are on an older version of Windows, you have two options. One is to install Cygwin (http://cygwin.org/). This will give you a UNIX-like environment on Windows; it's basically the open-source equivalent to the Subsystem for UNIX (but it works on all version of Windows, not just Windows 10).

But if you really don't want to do that, you can instead install Putty (https://www.putty.org/). This will give you a little app (with a very trivial GUI) that will let you use ssh.

3.1 Your First Connection to Lectura

The first time that you connect to Lectura from a given machine, you'll get a warning from ssh (or Putty) that about some sort of Key that it needs to import. Go ahead and allow this[1].

Once you make the connection, Lectura may ask you to change your password; if it does, then change the password and write down the new password so that you don't lose it.

Once you logged in, you should see a "prompt" - which means a place where you can type commands. For me, I see the following

```
Last login: Wed Jul 17 06:21:32 2019 from xxx.xxx.xxx.xxx
russelll@lectura:~$
```

[1] ***Technically,*** this is a security risk, as there's no way to know for sure, the first time you connect, that somebody isn't intercepting your traffic on the way to Lectura. However, as a practical matter, the chances that you're being hacked, exactly at the moment you connect, are very slim. Note that, after you've accepted this host key from the server, you will be protected for all future connections: if somebody intercepts you, they won't be able to replicate the host key that you've downloaded, and ssh will know that you're being attacked.

4 Your First Repo

Once you have a GitHub user ID, go to the class website, and follow the link to create your own copy of the repository for this lab.

4.1 Teams

When you follow the link, you will be given the option to either create, or to join, a team. For the Labs, you will be working in pairs - so one of you should create the team, and the other one of you should join the team. (Talk with each other - I don't know how to change your team once you've set it.)

You will have to set up new teams each week. Sorry for the inconvenience - but I configured the system this way because I'm assuing that **some** students will want to switch partners (or, their partner might not come to class some week).

4.2 GitHub Creates the Repo

When the first person of your group creates the team, GitHub will automatically create a repo for your team. Both of the team members will have access to clone and update this repo.

Once you've joined (or created) your team, you will be able to browse the repo using the GitHub website.

4.3 Accessing the Repo

Now, it's time to clone the repo onto Lectura. Log on to Lectura, and navigate to the proper directory; I'd recommend that you create a directory for this class, and then `cd` into it. (New to the command line? Check the class website for a list of reminders about helpful command-line commands.)

Once you're in the proper location, you can clone the repo with the command

```
git clone https://github.com/get_this_URL_from_the_GitHub_website
```

This will create a new directory, which contains the current contents of the repo. You can `cd` into it.

Once you're in the directory, create a new file named `github_lab.txt` - please be **exact** about the filename. Write a short haiku (as silly as you like), and put it in the file. (Not sure how to use a text editor on UNIX? There's a help file for that on the class website, as well.)

When you're done, push your changes up to GitHub. This will require three commands

```
git status                              # this will show you the changes
git add name_of_file_you_created_or_changed
git commit -m "Description of the change goes here"
git push
```

Finally, log onto the GitHub website again, and confirm that the changes are up there.

5 Extra: Git from Other Places

Have a little extra time? You can also clone the repository on your own computer.

Again, the **best** way to do this, long-term, is to learn the command line. If you are on Mac, Linux, Windows Subsystem for UNIX, or Cygwin, you can clone your repository on your own computer exactly like you did on Lectura. See the instructions above; they should work just fine on any system that has a UNIX-style command line. (Most systems come with `git` pre-installed; if yours does not, then you'll have to install it. Google for instructions to see how to do this on your system.)

Another option - which is easier to do in the short term, but worse in the long term - is to use GitHub Desktop, a GUI provided by GitHub. Go to the GitHub website, open a repository you have access to (such as this lab) and hit the green button to clone it. One of the options is "Open in Desktop." If you click this, it will take you to a page where you can download the GitHub Desktop tool for your computer; once it's installed, you can use it to clone repositories without using the command line.

However you do it, explore a bit. Try making small modifications, on your computer, or on Lectura; push them up to the GitHub website, and then use `git pull` to bring them down, to the other machine. You'll learn how to share code, across multiple computers, quite quickly.