Simplified Git Workflow

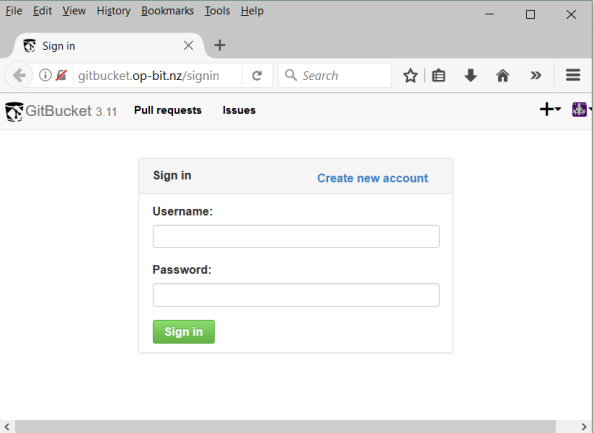
These are the steps in the simplified Git workflow you need for some of your Bachelor of Information Technology papers this semester. Each step is explained in detail below. This handout assumes that Git is installed on the machine you are using, and that the application path is correctly configured. Git is already installed and configured on all OP machines (be sure to use the Git cmd shell). For your personal machines, we recommend that you install the Git for Windows package, from here <https://git-scm.com/download/win>.

## Summary (detailed instructions for each step are given below)

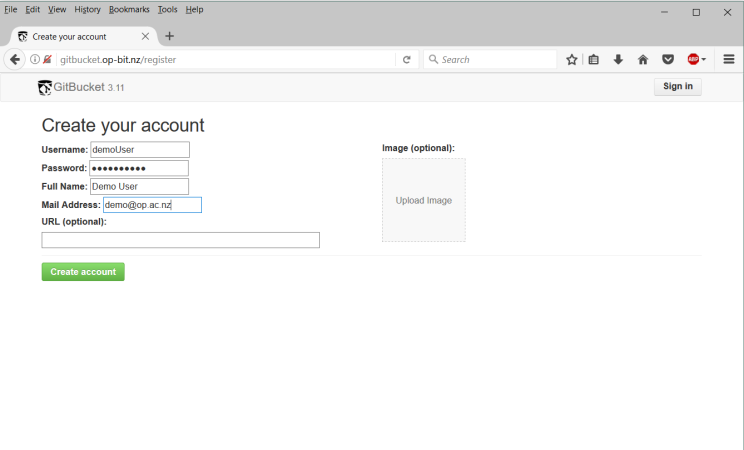
1. Create a Git repository (repo) called *PaperCodeYourOPUsername* on our GitBucket server. For example, if your OP username is smithz4 and you are creating your repo for IN710, it should be called IN710smithz4. Do not insert any additional characters, including spaces or underscores. **If you do not follow this naming protocol, your lecturers' automated marking systems will not be able to see your repo.**
2. Make your repo private and set your course lecturer as a collaborator (detailed instructions below).
3. Clone your repo to your H: drive (and any other machines you want to work on). This will be your local work folder; it is fine to have multiple copies on different machines.
4. Work on your code in (one of your) local folders.
5. Whenever you reach a sensible save point (at the end of each work session at a minimum), push (upload) your local copy to your remote GitBucket repo.

## Detailed Instructions

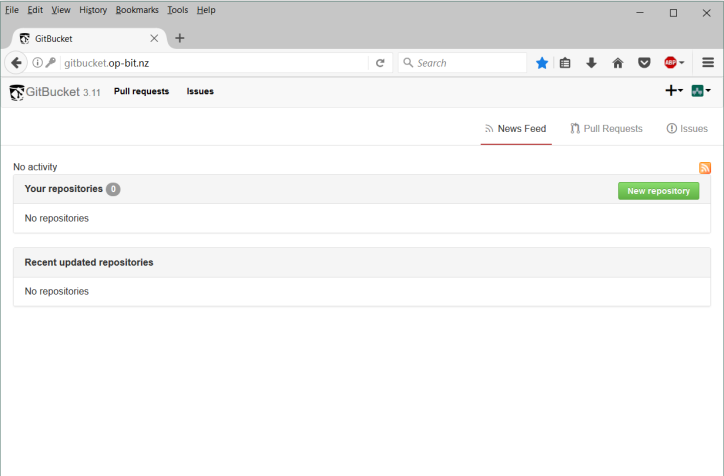
1. **Create a Git repository (repo) called** *PaperCodeYourOPUsername* **on our GitBucket server**.
   1. In any browser, go to <http://gitbucket.op-bit.nz/>. You will see the sign-in page of the IT department’s GitBucket installation.



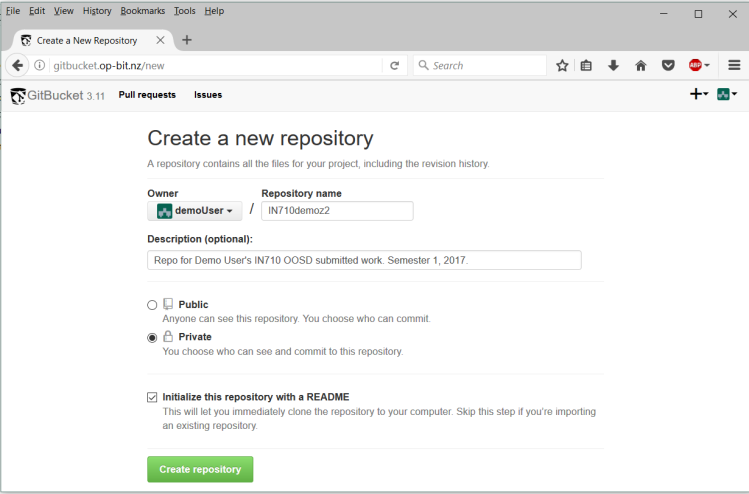
* 1. If you already have an account from last year, you may continue to use it. Otherwise, click “Create new account” and create an account using your OP username, and password of your choice. **NOTE:** This server does not have any facility for changing or retrieving your own password. Make sure you remember this password.



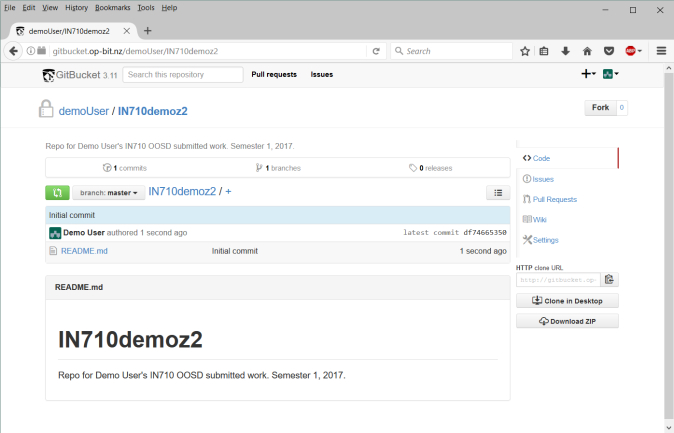
* 1. When you click “Create Account”, you will be returned to the sign in page. Sign in. You will then see your main account page (the physical layout will differ slightly depending on screen width).



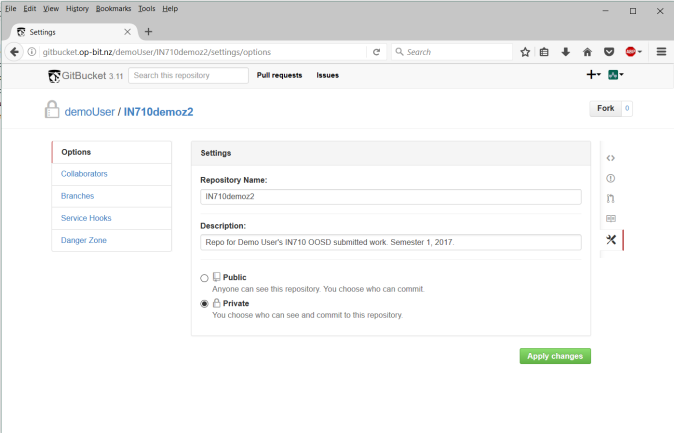
* 1. Click the "New Repository button". Name your repository *PaperCodeYourOPUsername.* As stated above, you **must** follow this naming protocol, or your submissions cannot be marked. Provide a sensible description, make sure the repo is private, and check the box to create a Readme.



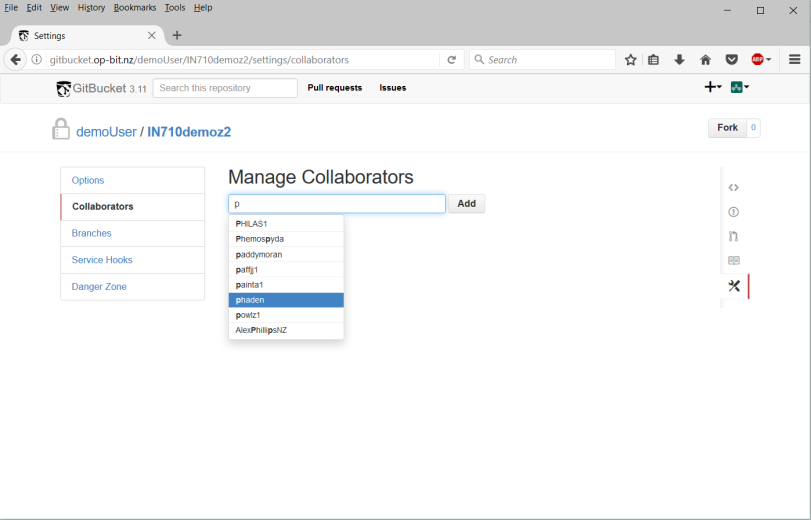
* 1. When you click the "Create Repository button", you will be taken to your main account screen, and you will see your new repo.



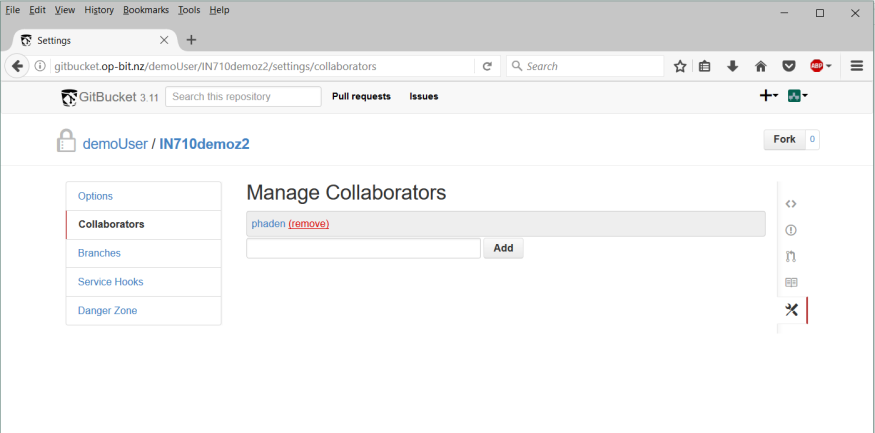
* 1. You need to add your lecturer as a collaborator. Lecturer accounts are first initial and last name (e.g. dparsons, kwood, phaden, etc.) Click on the Settings icon in the menu at the right edge of the screen (below “Issues, Pull Requests, Wiki”). You will see the settings screen.



* 1. In the Options menu at the left side of the screen, click Collaborators. On the resulting screen, type the first letter of your lecturer's username into the text box and look through the resulting menu for your lecturer. Select the username and click Add.



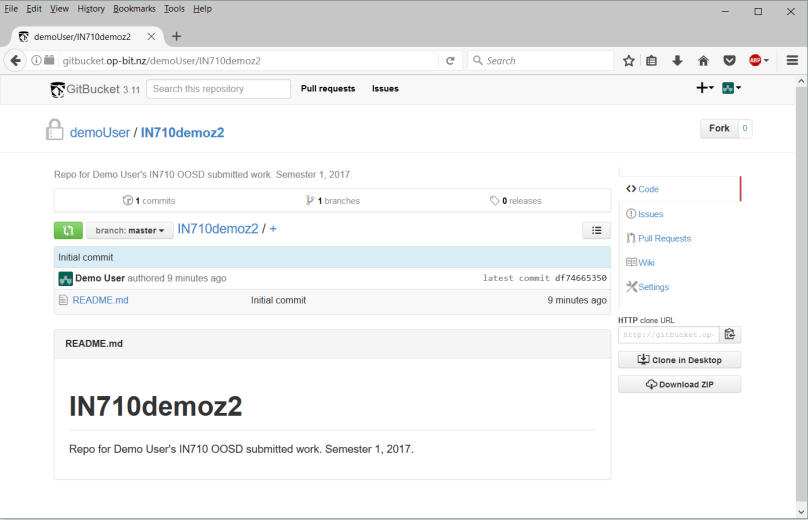
* 1. If everything works, your screen should look as follows.



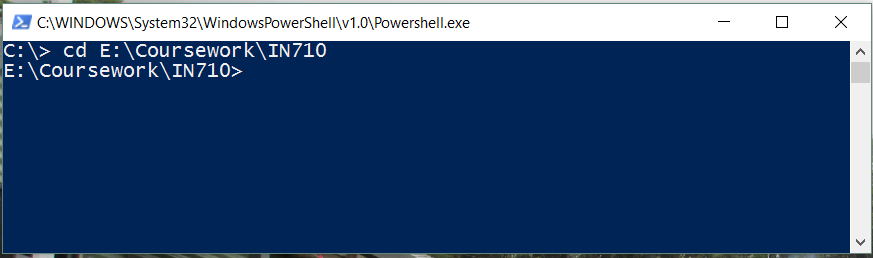
* 1. Click on the repo name link near the top of the screen (IN710demoz2 in the image above) to return to the main page for the repo.
  2. Continue on to step 2.

1. **Clone your repo to your H: drive (and any other machines you want to work on). This will be your local work folder; it is fine to have multiple copies on different machines.**
   1. While still on your GitBucket repo page, find the “Http clone URL” label at the right hand edge of the screen, and click the little copy icon beside it (see the red arrow in the figure below). This will place your repo’s URL in the clipboard. Open your favourite text editor (Notepad or Wordpad are fine), paste this URL into a new document and save it. You will need it for later. The URL should have this format, with your username and repo name:

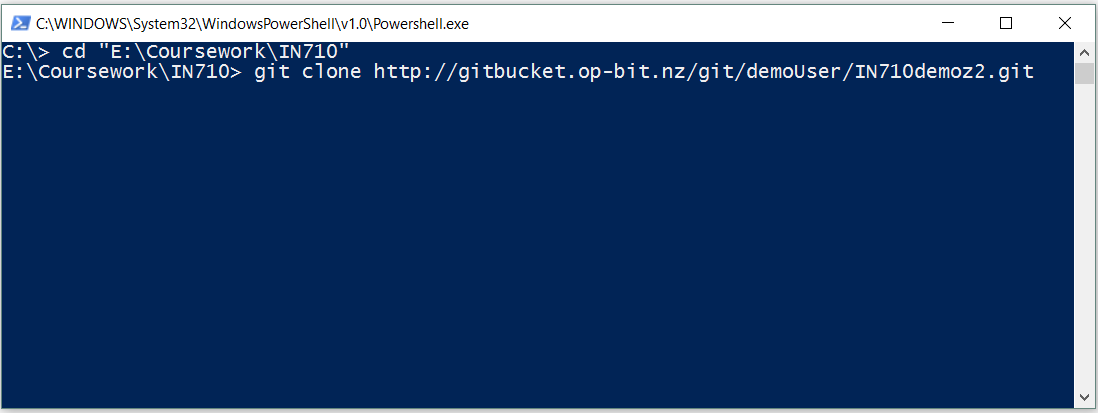
[http://gitbucket.op-bit.nz/git/demoUser/IN710demoz2.git](http://gitbucket.op-bit.nz/git/demoUser/IN710demoz2.git%20)



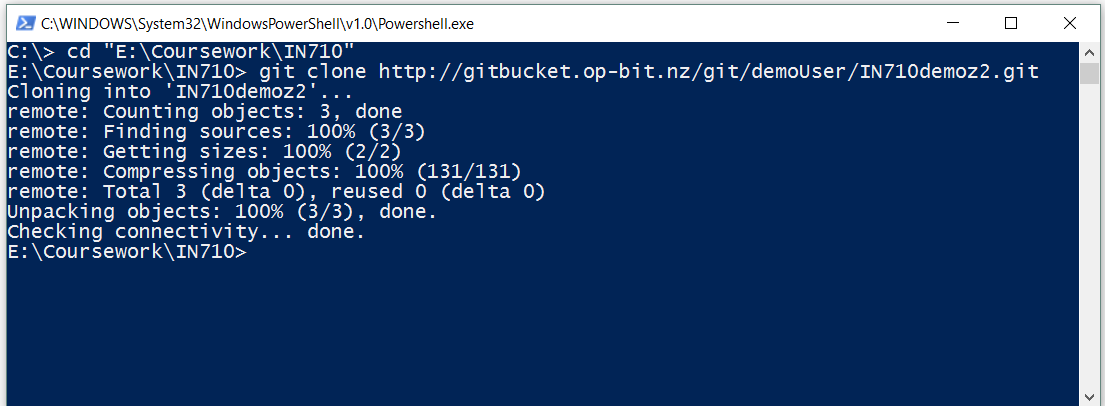
* 1. Open your favourite command shell, and cd to the folder ***where you want the local copy of your work folder to be stored****.* In the example, I will be storing my repo on the E: drive, in a folder Coursework/IN710, which I have previously created. Many students like to make one local copy of their repo on their H: drive for use when they are on campus, and another copy on their home machines. This is fine. As long as you are conscientious about keeping your repos synchronised, you can have as many copies as you wish.



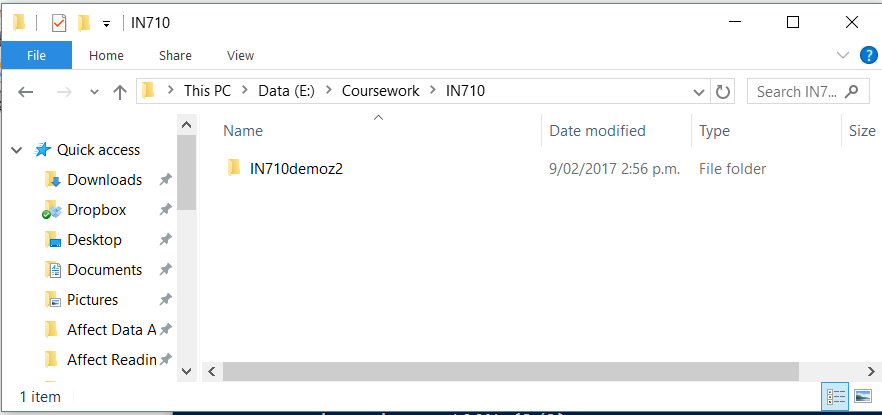
* 1. Clone your repo by typing **git clone** followed by the URL of your repo on GitBucket. This is the value you copied to the clipboard and pasted into a text file back in step 2a. (The figure below shows the URL of the demo repo; replace that with the URL you got from your GitBucket account page). Depending on how Git is configured on the machine you are using, you may be prompted for your username and password at this point. If so, give the account information you used to sign into GitBucket.



* 1. Assuming everything is working, you will see output like this:

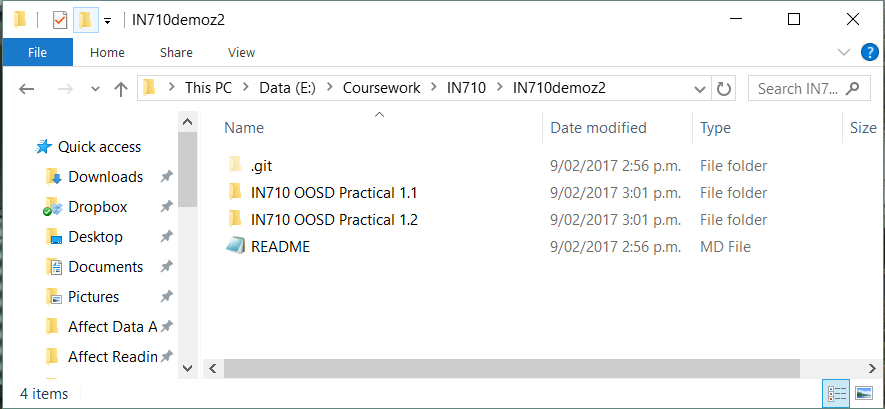


* 1. In Windows Explorer, go look at the folder *you cloned into*. There will be a new folder in there, with the same name as your repo.



* 1. Open this folder. It will contain a readme and a folder called .git. The readme contains the description you provided for your repo; the .git folder contains the files and folders Git uses to maintain the history of your repo. You generally will never need to touch any of these files (if it should be required, we will explain what you need to do).
  2. You should now save all your Visual Studio projects for the current paper to this newly created directory. You can place any files, and any directory structure you want in this folder. When you push the repo (instructions below), everything in this folder will automatically be transferred to your GitBucket repo, and be visible to you and your collaborator(s).

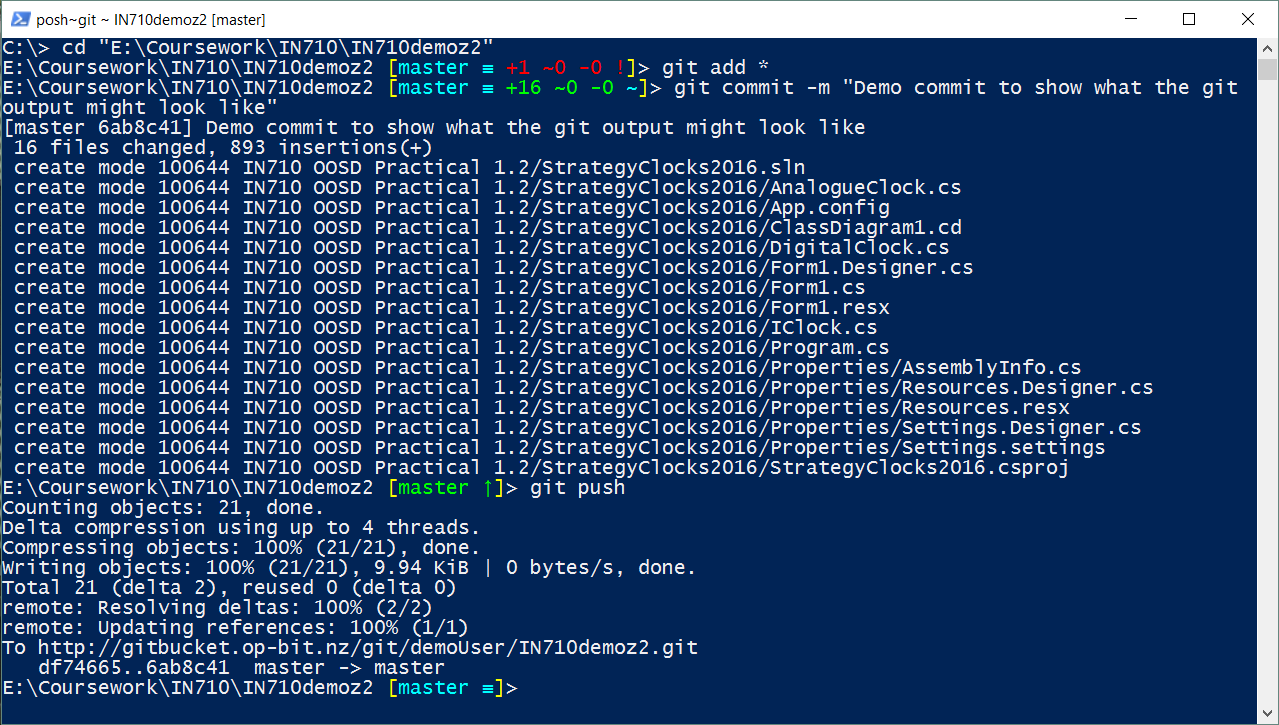
1. **Work on your code in (one of your) local folders.**
   1. When you create a VS project for a practical or assignment, save it in your local repo folder. You can also create a more complex directory structure, if you like, for example, to make separate folders for each week of the semester. This local repo folder can be treated exactly like any other Windows directory.



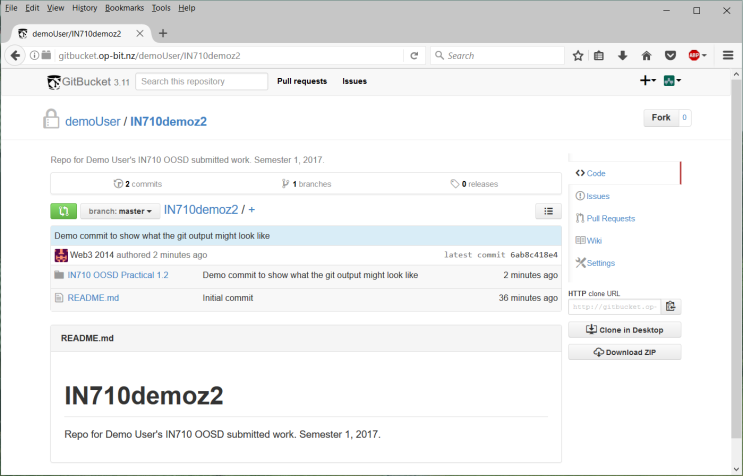
1. **Whenever you reach a sensible save point (at the end of each work session at a minimum), push (upload) your work folder to your GitBucket repo.**
   1. Open a command shell and navigate to your local repo folder. Make sure you get all the way into the folder that contains the .git directory and the Readme.
   2. You will then need to give three Git commands: **add**, **commit**, and **push**, with various command arguments (see below). The **add** command will tell Git to save all files that have been changed since your last commit. The **commit** command will tack on a message in which you explain what changes you have made, and will update ***the local copy*** of your repo metadata. The **push** command will upload everything to the remote GitBucket server. Git actually provides much more functionality than this, but these are all the commands we need for this semester (barring unusual events). The following table shows the commands as you need to type them. The output Git will generate depends on exactly what files you are pushing. (See the Git documentation for more details).

|  |
| --- |
| git add \* |
| git commit –m “Enter a meaningful commit message. There are marks for this.” |
| git push |

The following figure shows an example of what the Git output looks like, assuming you installed Git for Windows and are using PowerShell. If you have a different configuration, it will look slightly different, but the basic content is representative. Again, you may be prompted for your username and password when you push; provide them as needed.



To verify that this all worked, go back to your GitBucket account page in the browser (refresh if necessary). All the new files in your working directory will be there, safely uploaded to the remote server.



If your local work folder ever becomes damaged, or you want to access your files from a different machine, you can simply clone your GitBucket repo again, and all the contents at the last push will be downloaded.

If you want to learn more about Git, try this [simple interactive tutorial.](https://try.github.io/levels/1/challenges/1) If you'd like to read more about how Git works under the hood, this [core documentation](https://git-scm.com/book/en/v2/Getting-Started-Git-Basics) is the best place to start. If you want to see the full glory of Git for large software teams, try this [very complete set of tutorials from Atlassian](https://www.atlassian.com/git/tutorials/). And finally, if you want a more scholarly discussion, this is the [classic Git textbook](https://git-scm.com/book/en/v2/Getting-Started-Git-Basics), available as a free eBook.