Git git git Advanced Software Engineering

Version Control using BitBucket, GitBash and SourceTree

Links

[GitHub Free Student Pack](https://education.github.com/pack)

[Learn Git Course](https://www.codecademy.com/learn/learn-git)

[GitBash for Windows](https://git-for-windows.github.io/)

[SourceTree](http://www.sourcetreeapp.com/download/)

***Disclaimer***

At the time of writing SourceTree is not available in the labs. GitBash is currently working if you configure it correctly.

## Introduction

BitBucket is an online repository. You can sign up for a free account.

[BitBucket](https://bitbucket.org/)

## GitBash/Git for Windows

GitBash is a command line utility that allows us to handle our version control. It’s actually quite easy to use and I do prefer this way to the graphical system (SourceTree). We can use cut and paste with GitBash or the standard command line. If you open a command line on Windows by typing “cmd” into the search box then once you have installed Git For Windows it will understand git commands. You should also be able to type “git-bash” and get a git only command window.

Open GitBash, in the labs it’s in the delivered applications window, but you may have installed it on your own machine so it will be available by the start menu or as an icon on your desktop.

To configure GitBash in our labs (not at home) do the following in the command prompt (type “cmd” into Windows’ search box).:

**git config --global http.proxy http://wwwcache.leedsmet.ac.uk:3128 ​**

**git config --global push.default simple**

You shouldn’t have to do this again, but it may be that different labs do not store your local setting, so if you have a problem repeat this step.

When we use version control we are working with our usual source files but also “committing” them to our version control repository. GitBash does this by firstly storing the repository of all the different versions on our local computer but we can also push that to the cloud, in our case we will use BitBucket, but we could use another system. First off though we will set it up on our local machine. This is easy, just start GitBash and then go to the directory where your source files are located. GitBash is Linux based and so uses Linux commands, such as “ls” for a directory listing, but it does have “cd” for change directory. Store your work on your student drive and “cd” to it. Your student drive is F:

Create a Visual Studio project on your F: drive, or use one of the ones you already have. I’ve called mine GitDemoProject and the place where my source files are located is

GitDemoProject\GitDemoProject

In GitBash--

cd f:\source\myProj

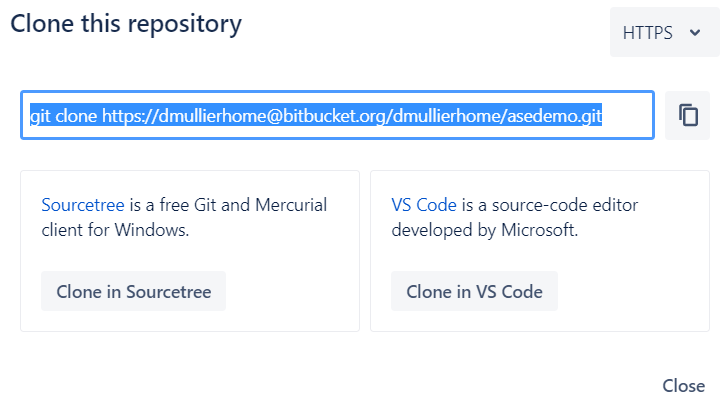
Assuming I have already created that directory, if not make it (you can do this using GitBash, Windows Explorer, or a normal command line prompt).

## Create A repository

Essentially we can do this one of two ways. We can create it on our local machine and send it to BitBucket, or we can create it on BitBucket and the copy it to our local machine. We can also “clone” a repository from BitBucket and put it on multiple local machines, so several people can work on them.

Go to the [BitBucket](https://bitbucket.org/) and click the “Repositories” menu and select “Create Repository”. Don’t create a readme file or you won’t get the page shown below. You need to have one for your assignment but you might want to just create a test one for now. Call it ASE and select C# as the language.

You should now be able to create a project in Visual Studio. Give it the same name as your repository. I will call mine ASEdemo and place it in my repos directory. I’ve kept it simple by making it a console project but it could be anything.



NOTE: GitBash is a Linux command line and not DOS, so you have to use forward slashes for your directories and not backslashes. Note the URLs below have come from me cutting and pasting from BitBucket’s useful set up when you create a new repository, obviously your URLs will be different to the ones referencing “dmullier” here.

mkdir /path/to/your/project

cd /path/to/your/project

I’m putting my project in



cd \users\duncan mullier\source\repos\asedemos

git init

This creates a subdirectory .git with all the necessary git files. If you look in explorer and check “hidden files” you’ll see it.

git remote add origin https://dmullier@bitbucket.org/dmullier/asedemo.git

The above is from the readme tutorial and the address is the same that comes from the lone button.

git push -u origin master.

This connects the remote BitBucket repository to your local files

We can also say that our project is simple and doesn’t involve complex branches by using (you should only need to do this once):

git push --set-upstream origin master

(If it doesn’t work, skip the above line, it is just trying to be thorough).

I can now add the appropriate files to my git project with:

git add \*.cs (or whatever you have called your file)

You can also use wildcards

git add \*.cs

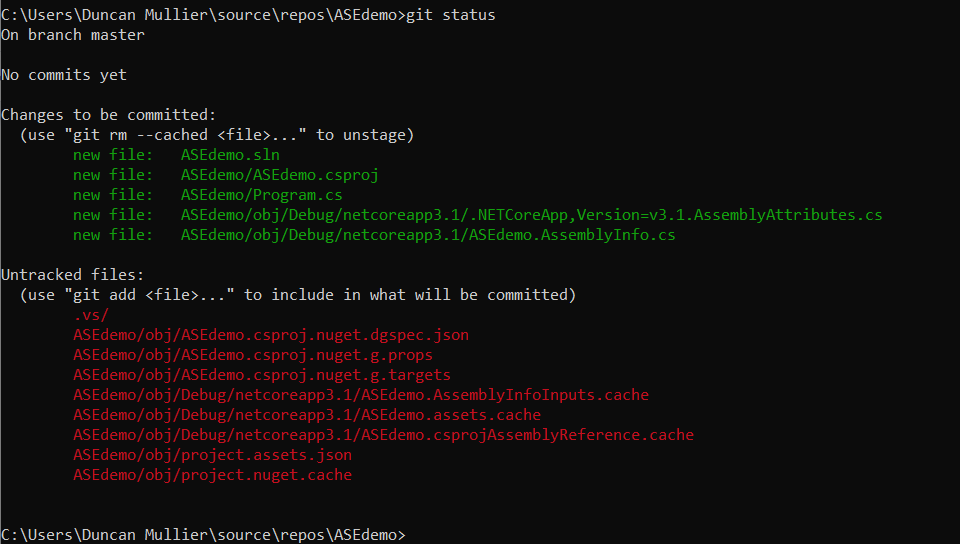
git add \*.csproj

git add \*.sln

It doesn't tell us anything at this point but if you issue the command:

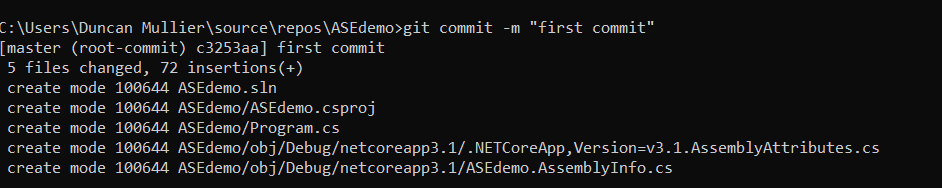
git status

This shows in green all files in all the sub-directories that have been “staged” and all those that haven’t.



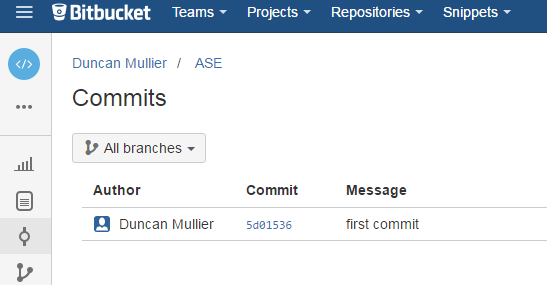
This does not cause anything to be sent to bit bucket. I do that by “committing” it. When I do this it will send everything that has been added and has changed (green). In this case all the file types I have committed are classed as changed because they haven’t been committed at all before. I should also specify a commit message. This example is trivial, but you may be doing a proper project and each commit is a milestone or bug fix in your project and your commit message should succinctly describe that milestone.

git commit -m "first commit"



git push -u origin master

The push command finally send all you changes to BitBucket and if you click “commits” on BitBucket in your browser you will see:



You can now change your files and set up another commit by adding them to your next commit.

Change a file

Use git add <filename> to add it to the commit.

Set up the commit with

git commit -m "second commit"

Push your second commit to BitBucket with:

git push -u origin master

At any time you can see the status of your files with:

git status

There is a lot more to version control and Git than this (such as branches etc) but this will give you enough for your assignment.

## NOTE

## If you are having difficulty adding to an already existing repository, i.e. when you

git remote add origin https://dmullier@bitbucket.org/dmullier/ase.git

You get an error: [fatal: remote origin already exists](http://stackoverflow.com/questions/25420452/git-fatal-remote-origin-already-exists)

Then use the command

git remote set-url origin https://dmullier@bitbucket.org/dmullier/ase.git

# SourceTree

# (additional if you want to at home)

SourceTree is a visual front end. It isn’t available in the labs but you can use it at home. It will take the repository online and store a local copy on your machine. You then edit the source files on your machine and when you are ready you can “commit” your changes, using SourceTree, to the BitBucket server. Your code is stored in all its committed versions safely on the BitBucket server.

This is done by “cloning” your repository from BitBucket to your local machine.

## TASK 1. Clone the repository to your local machine

Cloning makes a local copy of the repository for you.

1. On Bitbucket, click the **Clone** button.
2. The system selects the URL for you.
3. Choose **Clone in SourceTree**.
4. SourceTree starts up and displays the **Clone New** dialog.
5. Click the **Clone** button.
6. SourceTree clones the repo from Bitbucket and opens it for you.

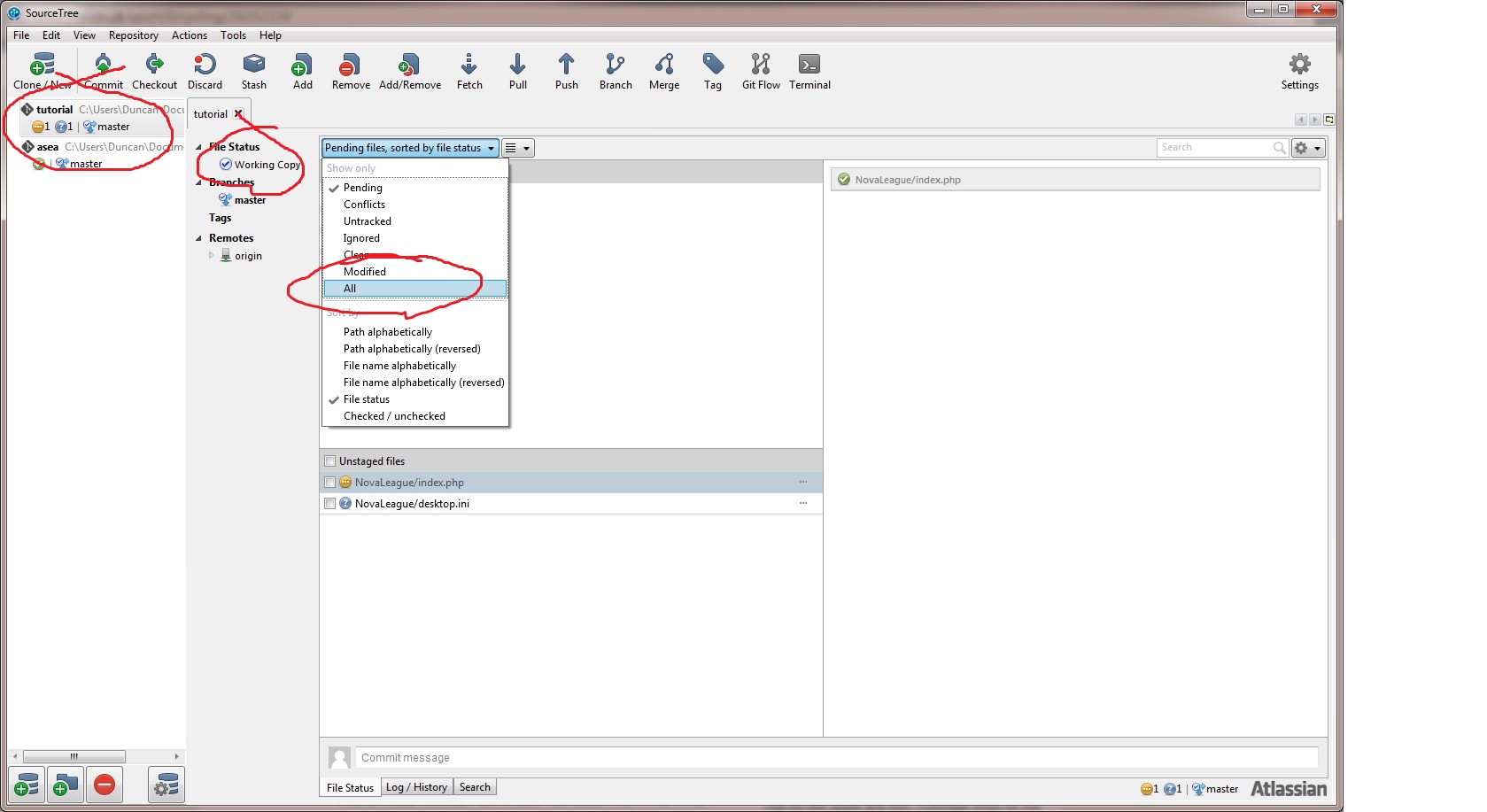
That's it, you've cloned your first repository! Keep the repo open and try the next task.

Now you’ll have a directory (SourceTree automatically created a directory but you could have changed it) which is empty. Put some code in it.

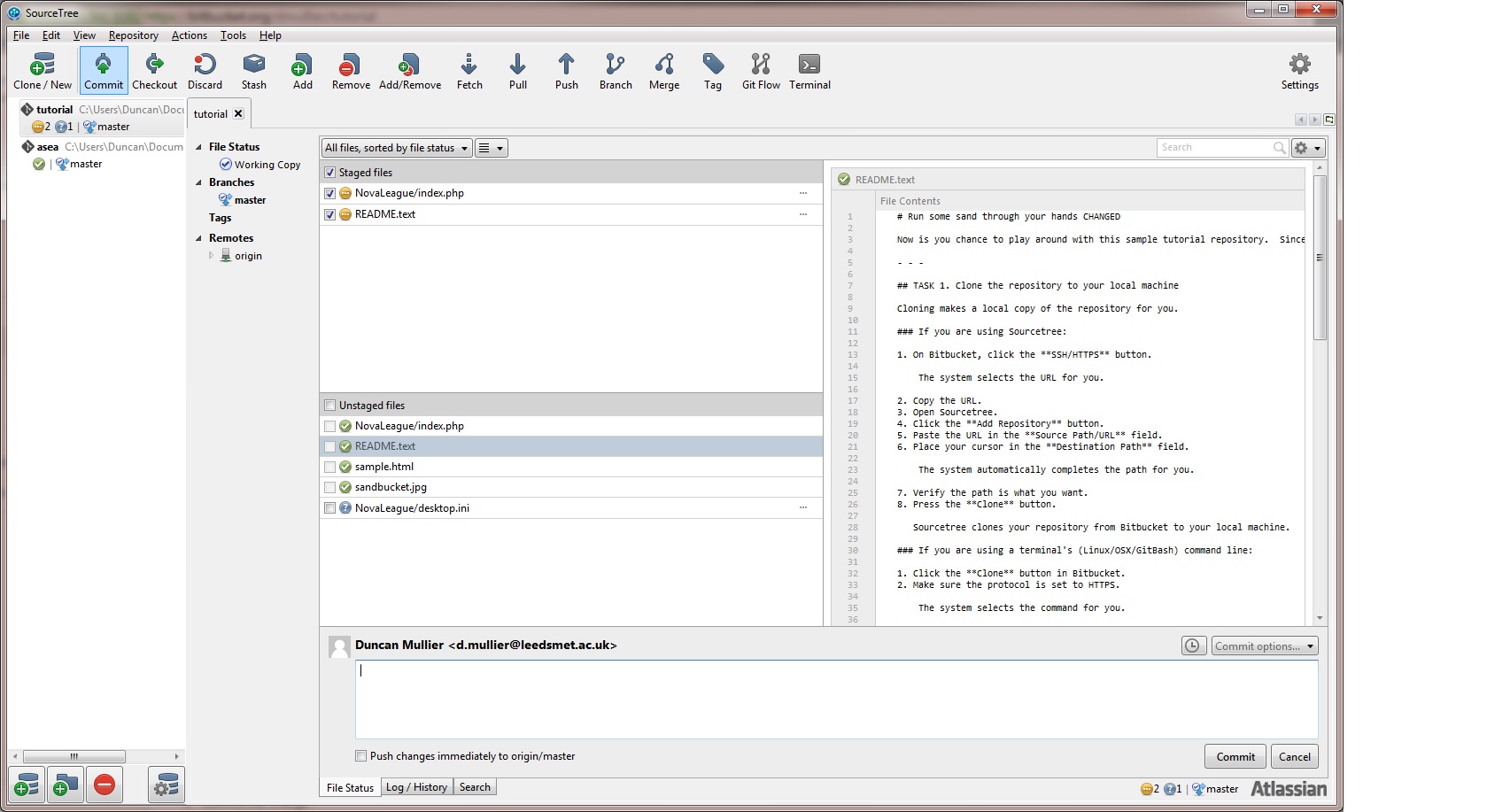
## TASK 2. Make a change

Make a change in the sample.html source file and push the change back to Bitbucket.

1. Double-click on your tutorial repo.
2. Select the **Working Copy** from the left hand navigation.
3. Select **Show All** from the dropdown.
4. SourceTree shows the files in the working tree.



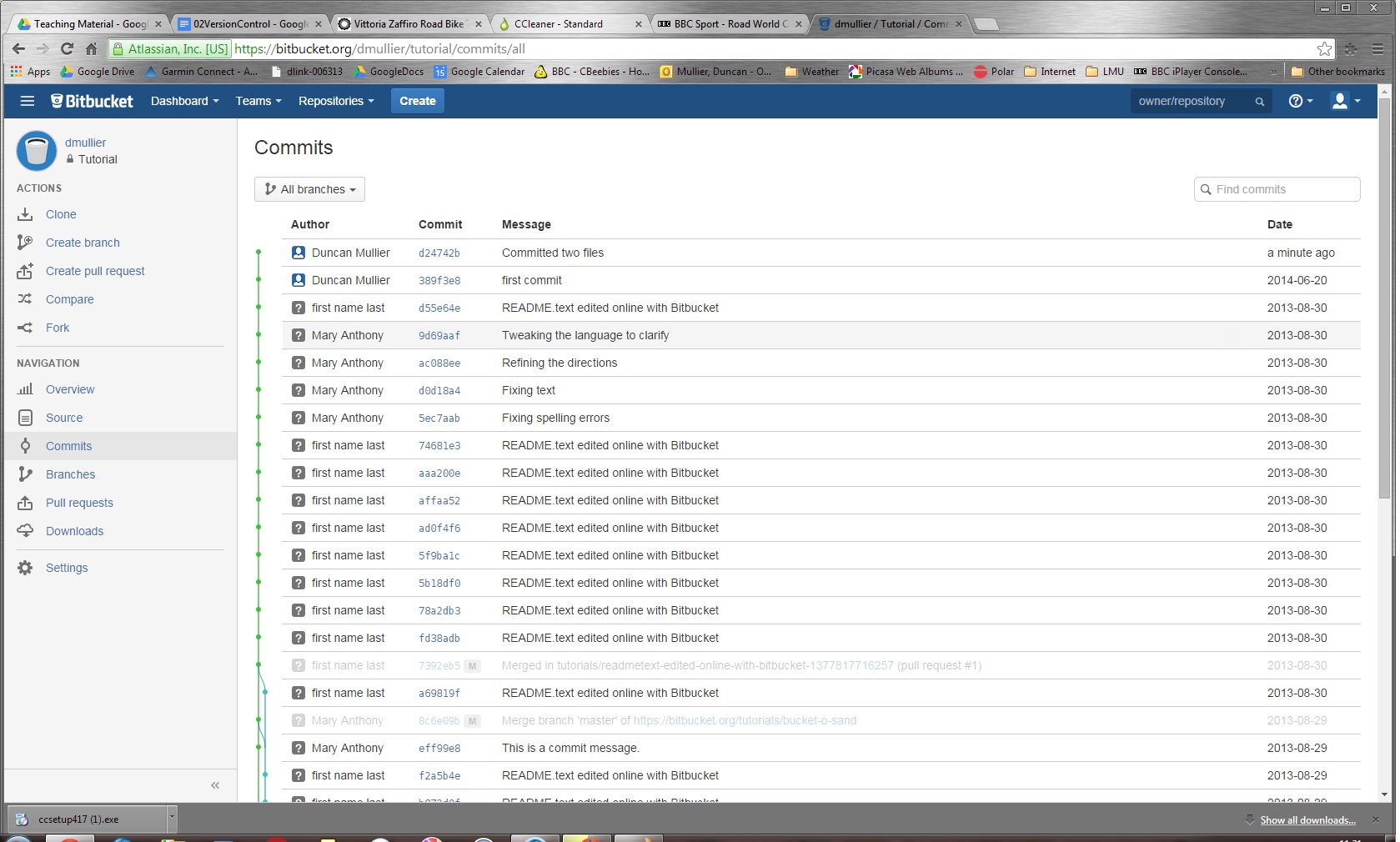
1. Locate the **sample.html** file in the repository.
2. You can view the file in a browser by double-clicking the filename.
3. Right click the **sample.html** file and choose **Open**.
4. SourceTree opens a terminal window to the directory containing the file.
5. Using your favorite editor, edit the sample.html file.
6. Change the heading from My First File to Kick over the bucket.
7. Save and close the file.
8. Return to SourceTree.
9. SourceTree changes the file's icon to modified.
10. Press **Stage File** or drag the file into the staging area.
11. SourceTree moves the file into the sample.html staged area.
12. Choose **Commit**.
13. Enter a commit message.
14. Press **Commit**.
15. Now, the **Push** icon shows you have a single commit ready to push to your repository.
16. Press **Push** to send your changes to Bitbucket.
17. Press **OK** when prompted.



1. After the push finishes, use the **Commits** tab on Bitbucket to view your change.

That's it for task 2. You've completed your first commit and push on Bitbucket!

Now try it with your own repository.



## Adding Users to Your Repository

Using BitBucket you can add users to your repository very easily. Just click on the repository that you want to share and then click “Send Invitation” which is located at the top right. I guess for now that you could add one of your friends to see that it works.

Now that you have your version control system set up you should use it for all your coding.