

GITHUB EXERCISE

Submitted to,

Prof. Christelle Scharff



November 21, 2016

Anuja More

Pace University

Q.1. What is GitHub? When was it created? Why? By who? What similar platforms exist? Why would you use such a platform?

Answer:

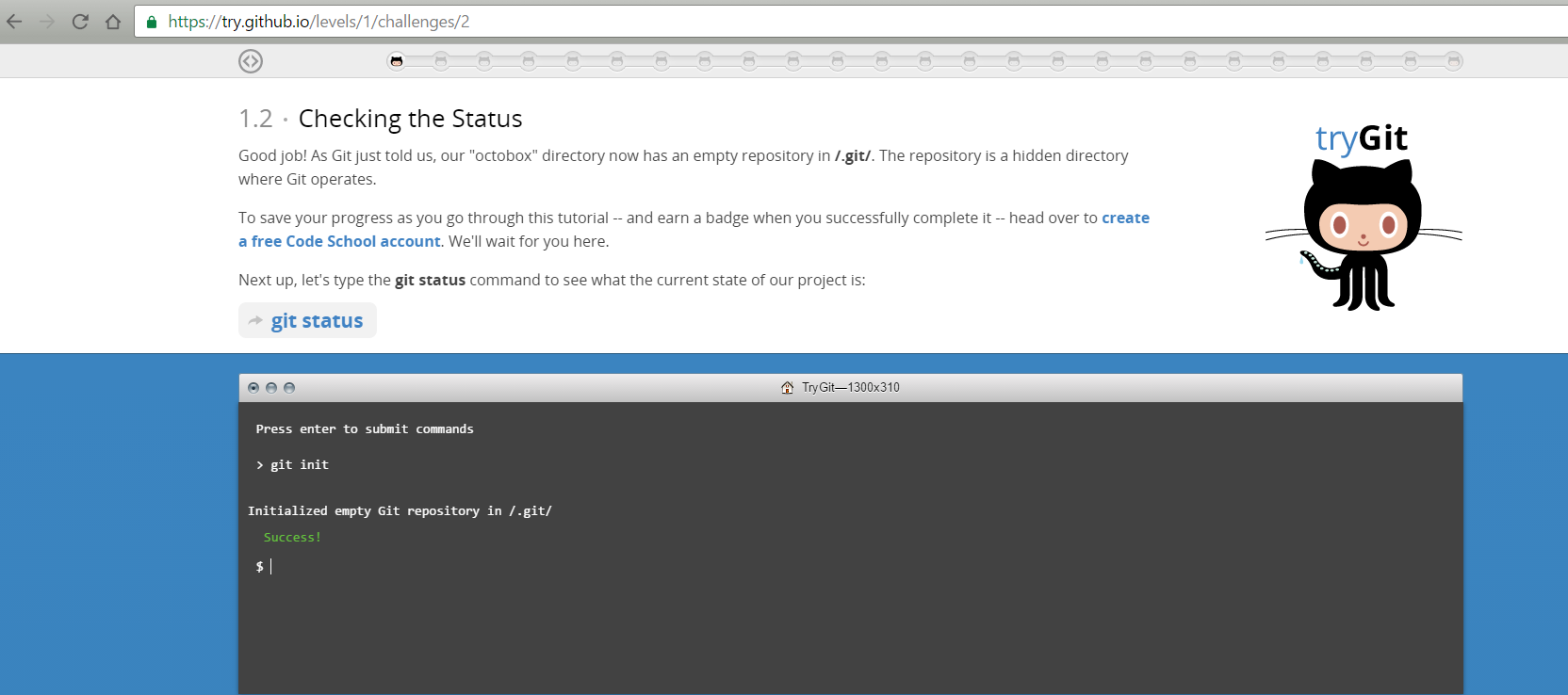
* Github is basically a Hub, i.e GitHub.com, where coders can save their projects and network with people with similar interests.
* Github is a web based graphical interface.
* It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project.
* GitHub was originally founded by Thomas Preston-Werner, Chris Wanstrath and PJ Hyett.
* There are few reasons why we should GitHub:

1. It is an open source.
2. Networking can help improving the skills.

**Tutorial steps:**

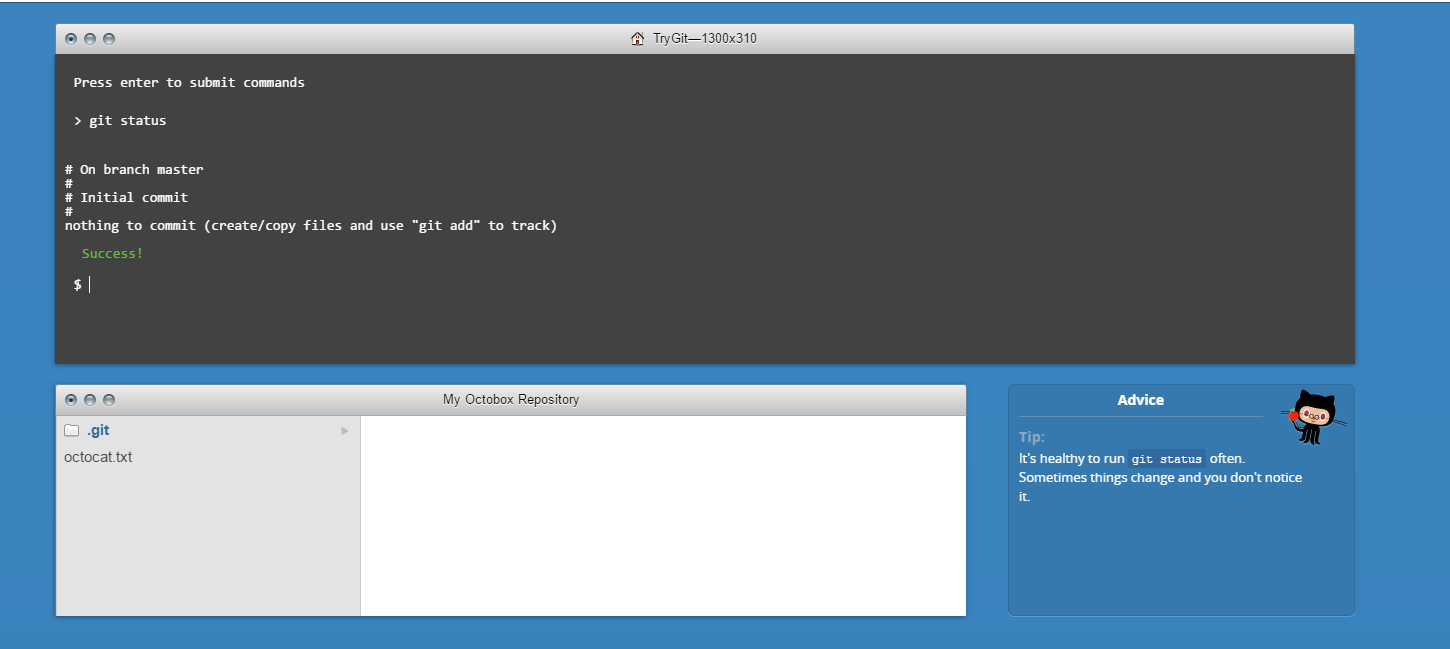
Step 1:

Initializing Git



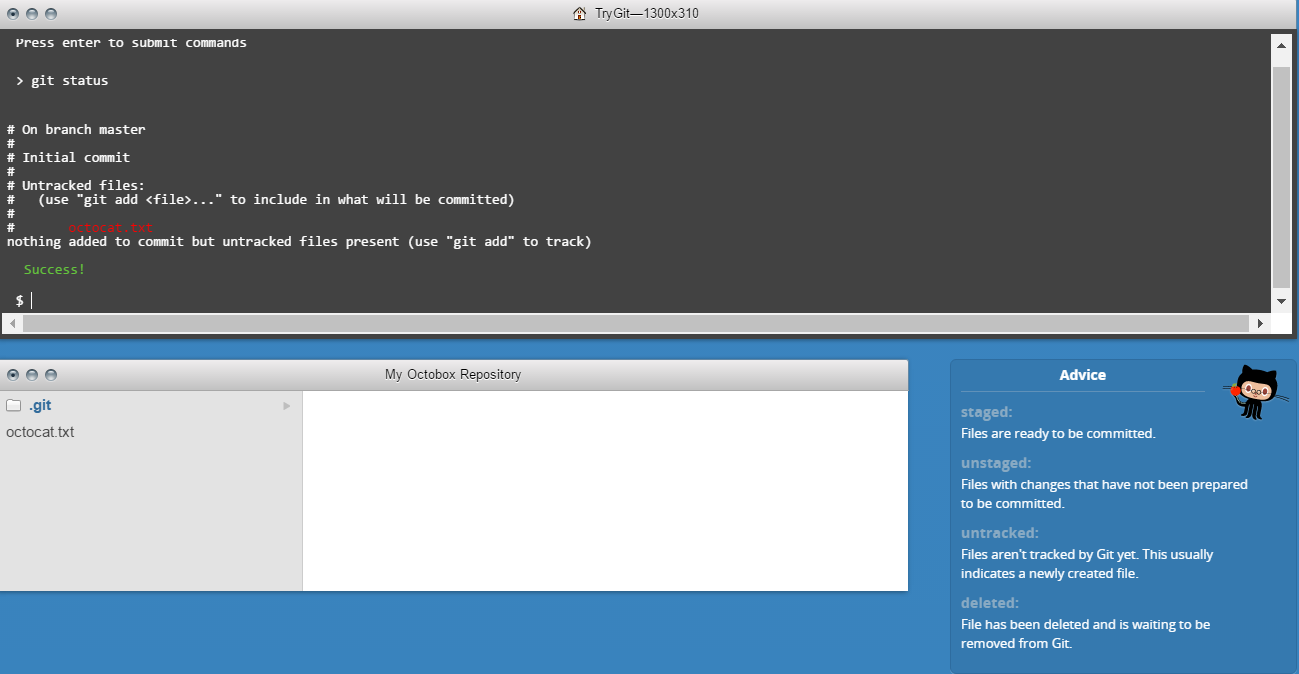
Step 2:

Checking the status



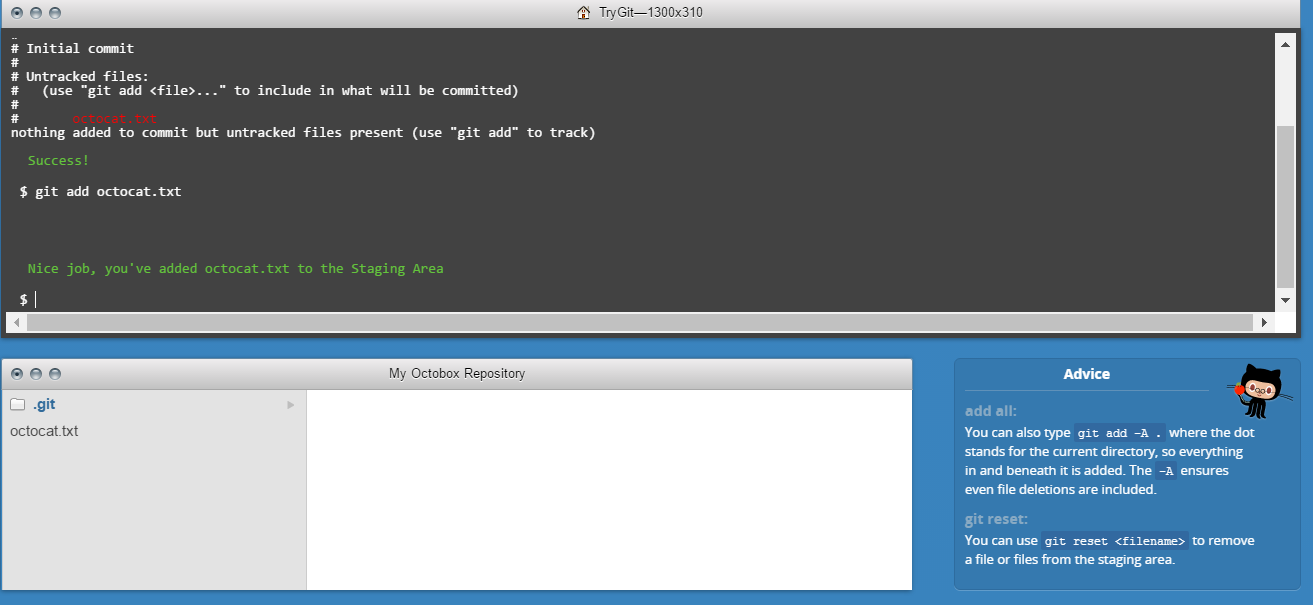
Step 3:

Adding and committing



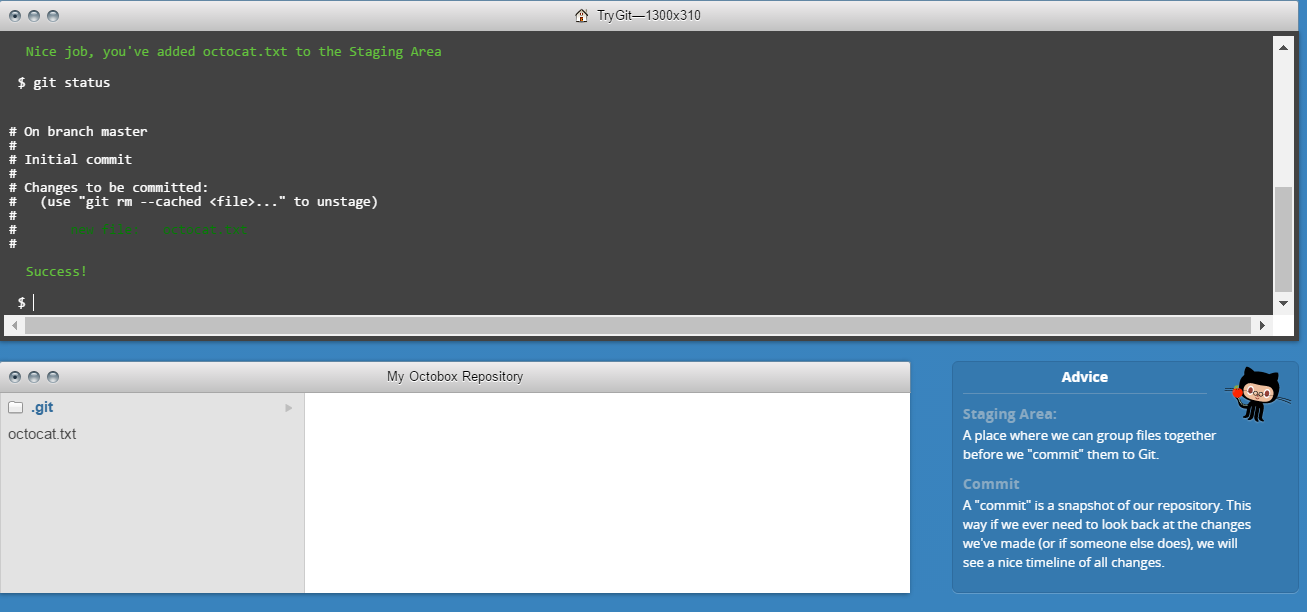
Step 4:

Adding Changes



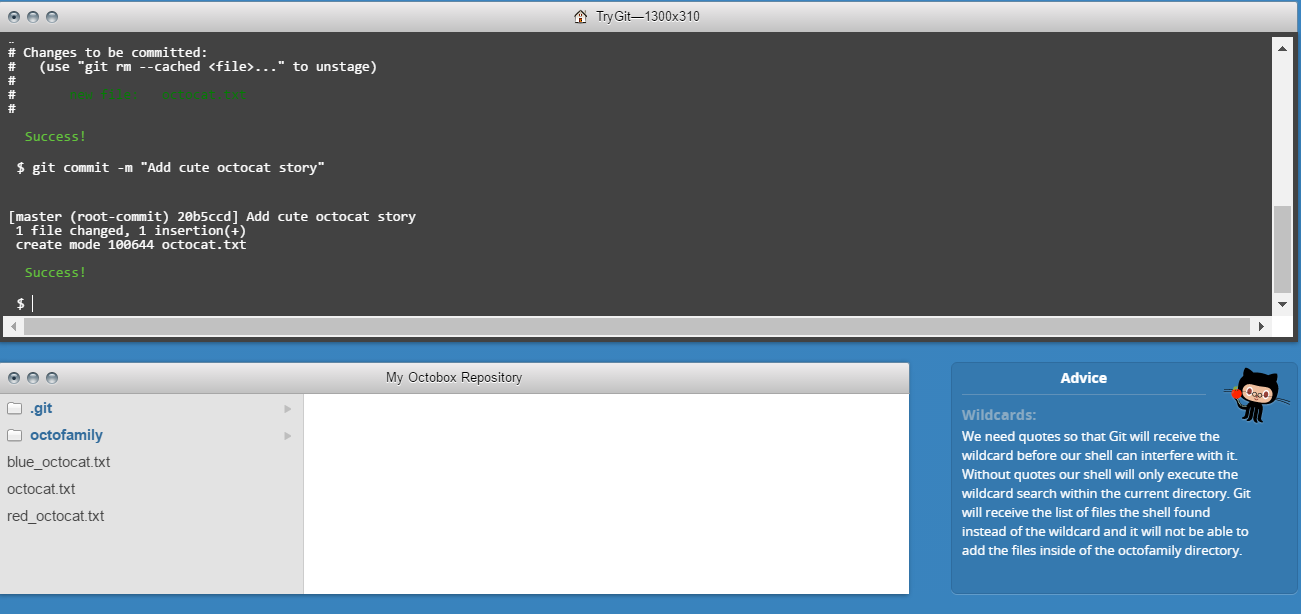
Step 5:

Checking for changes



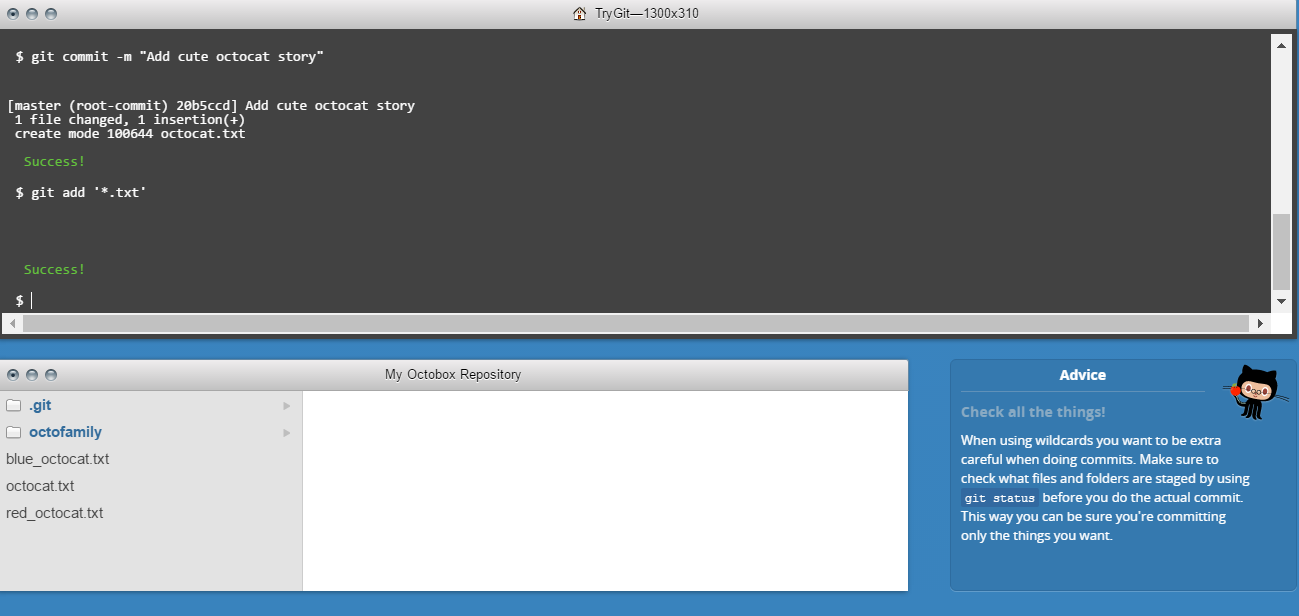
Step 6:

Committing



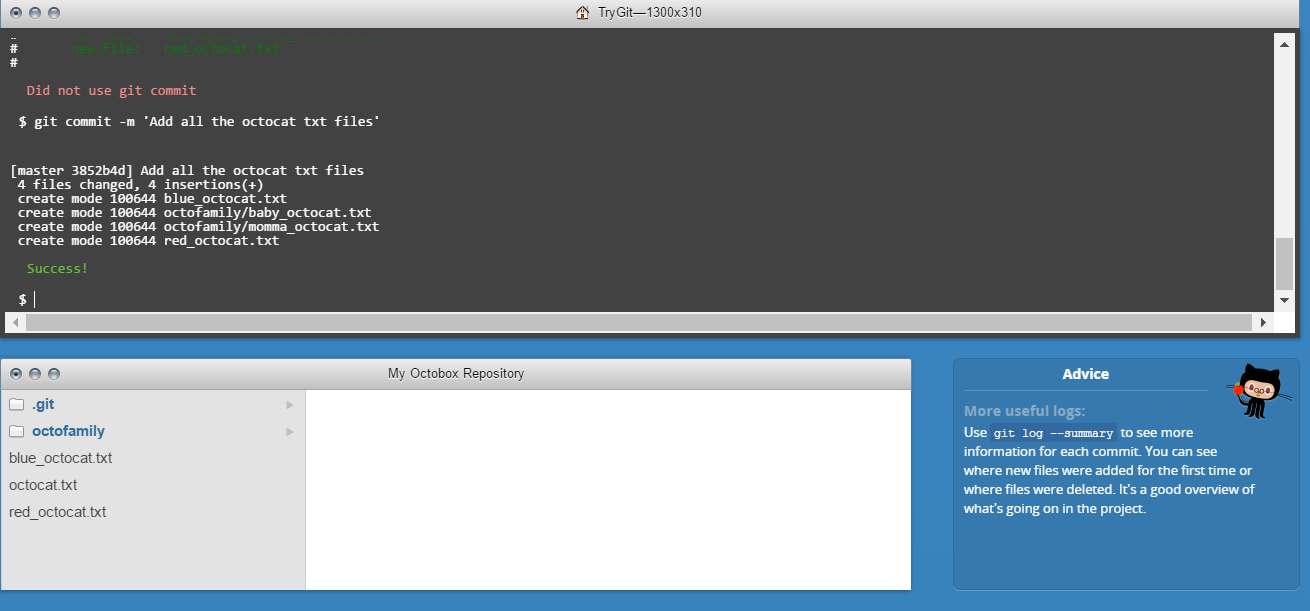
Step 7:

Adding all changes



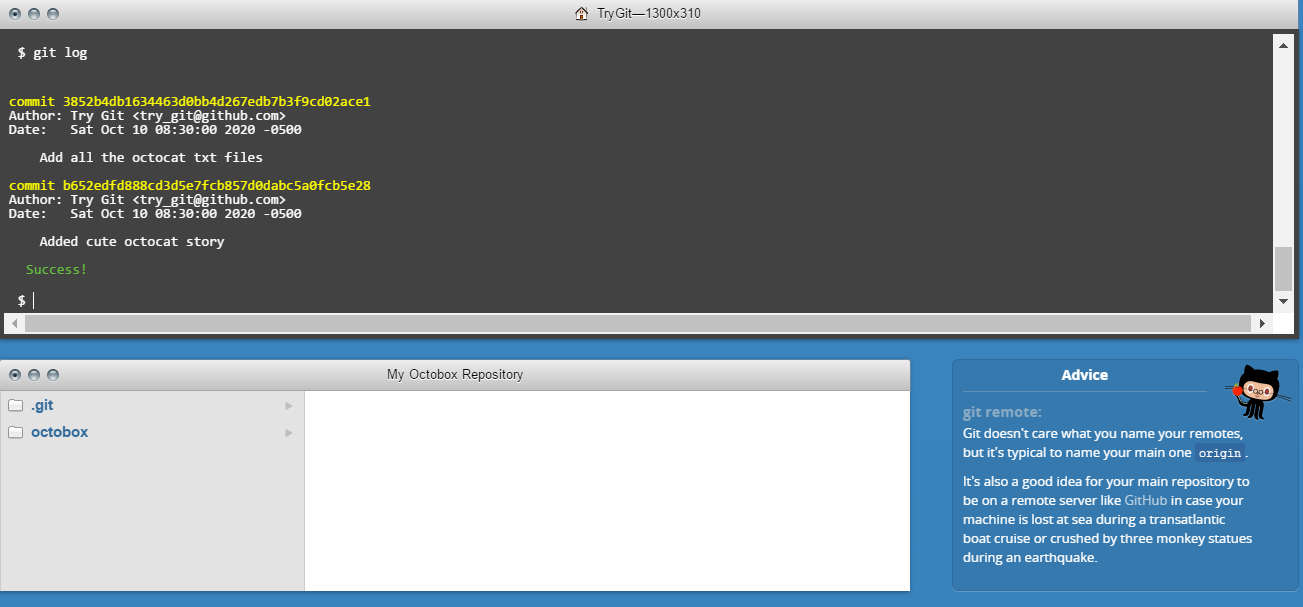
Step 8:

Committing all changes



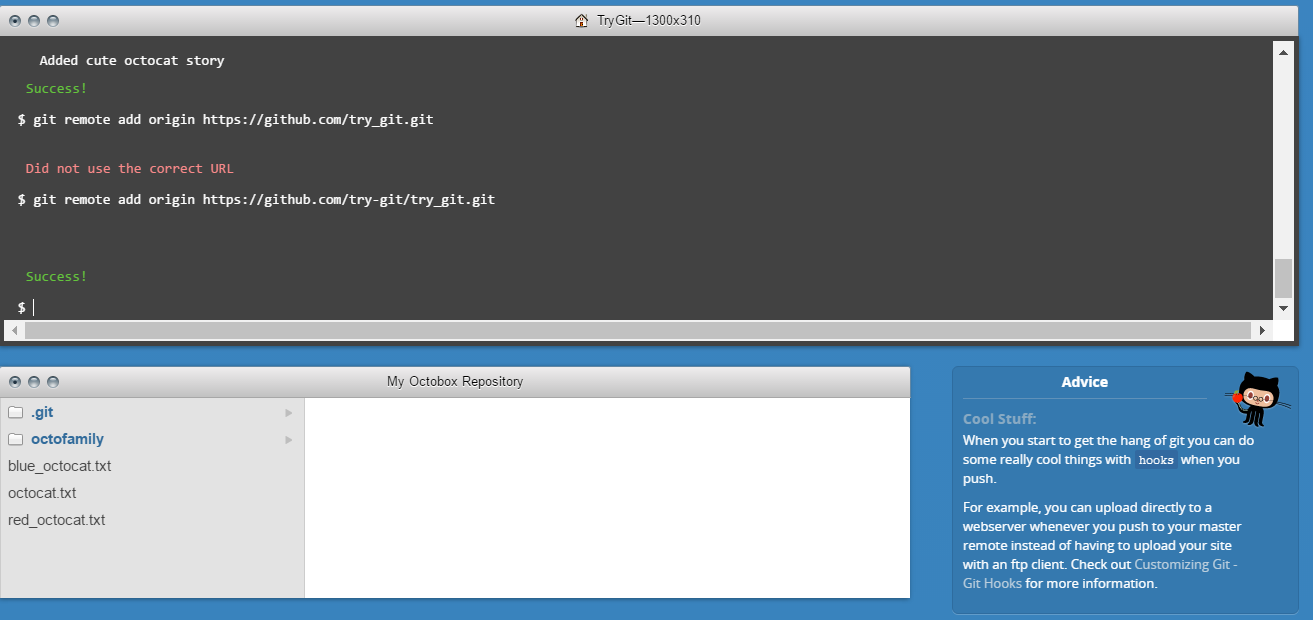
Step 9:

History



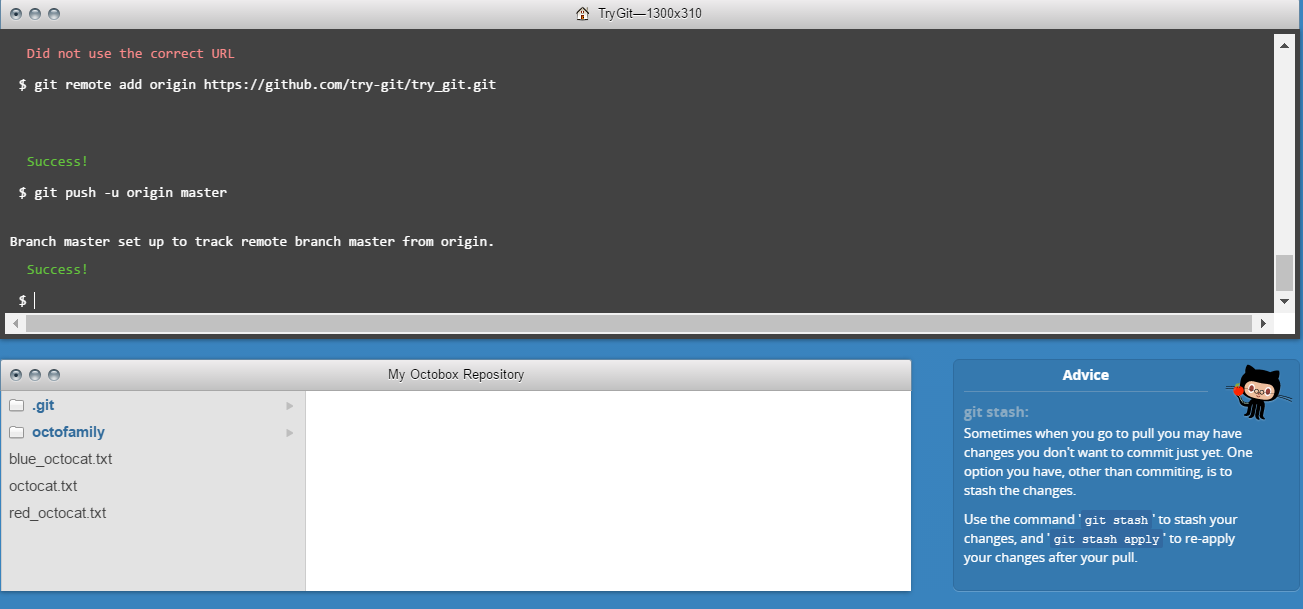
Step 10:

Remote Repositories



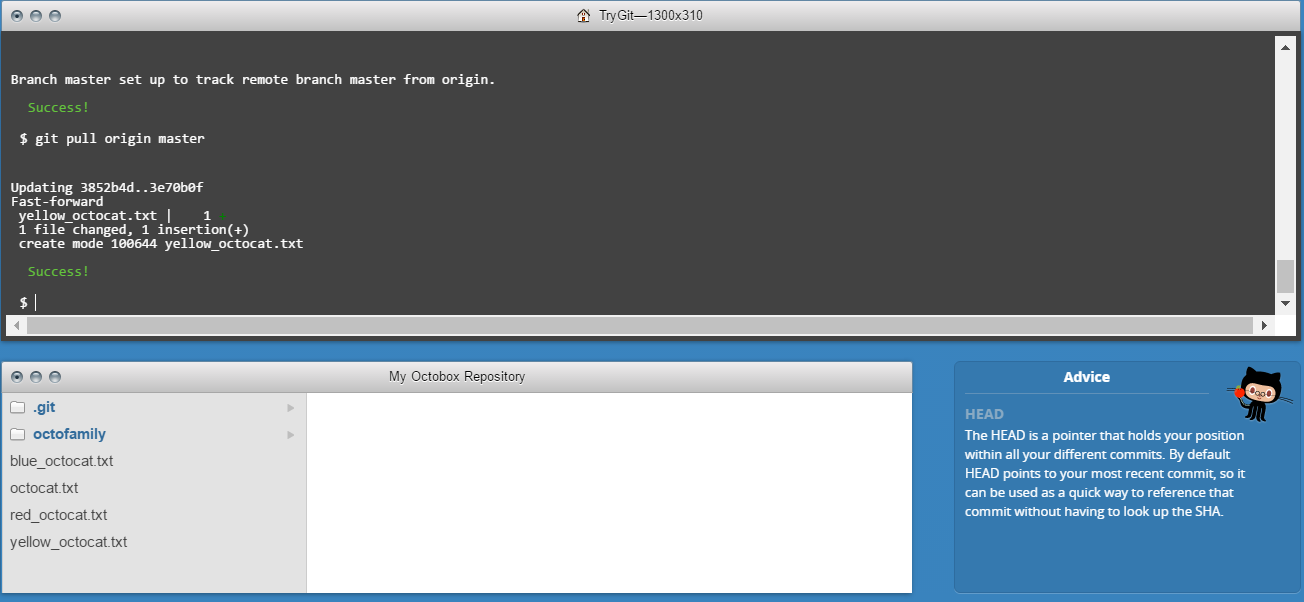
Step 11:

Pushing Remotely



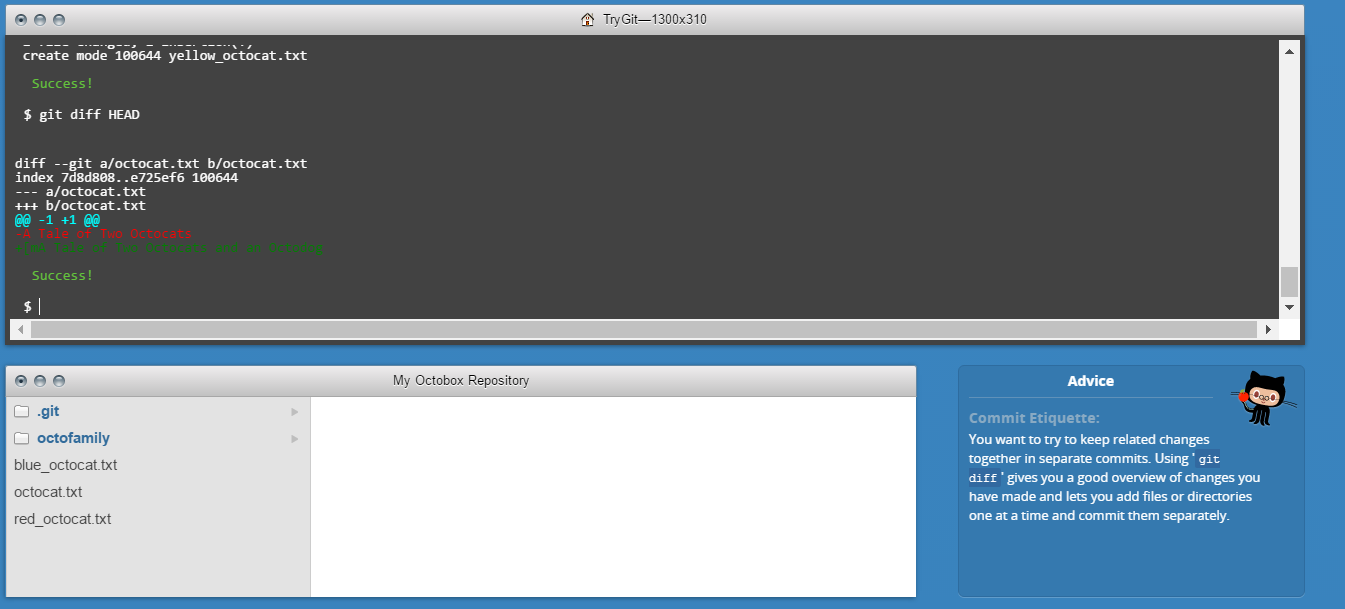
Step 12:

Pulling Remotely



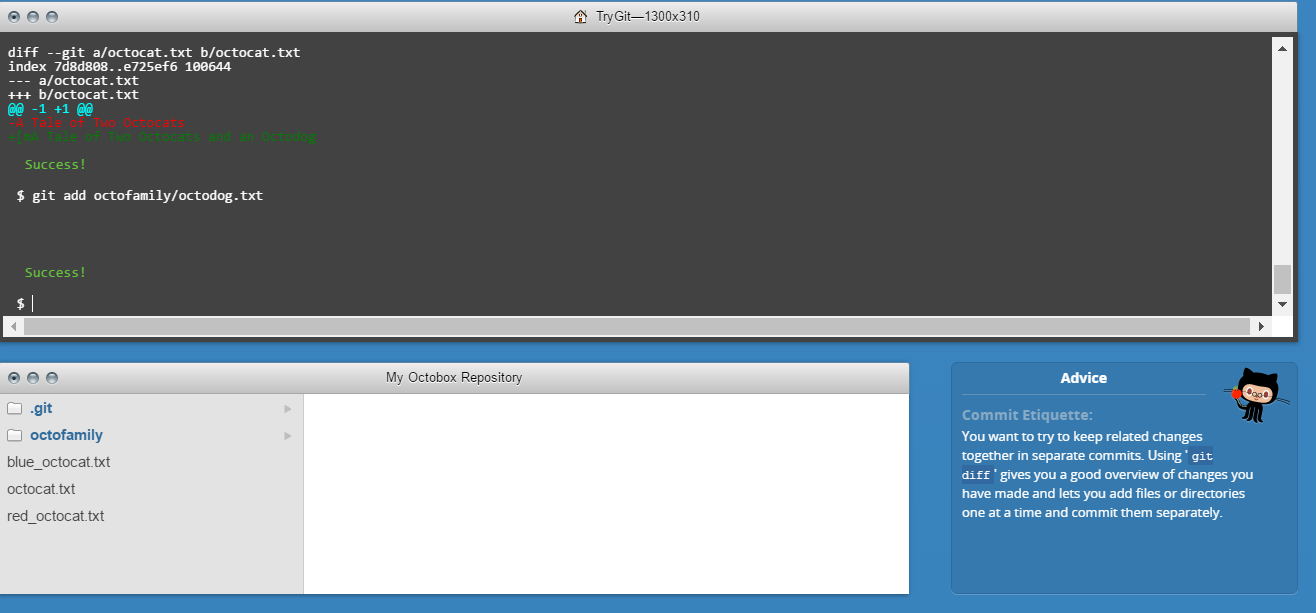
Step 13:

Differences



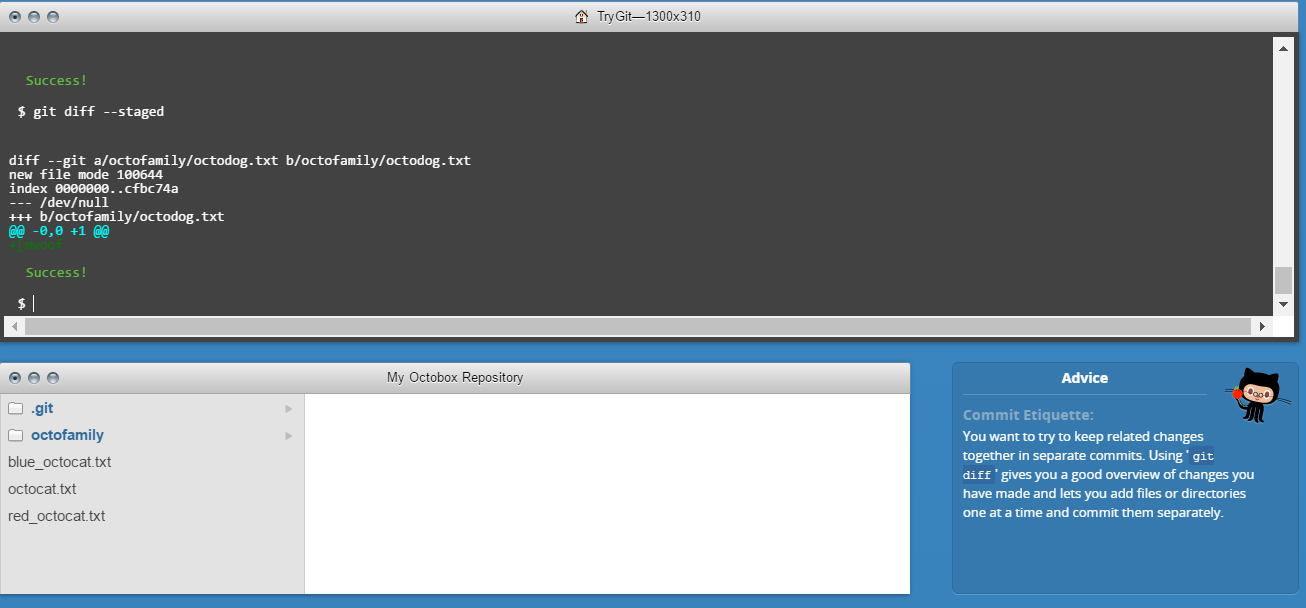
Step 14:

Staged Difference



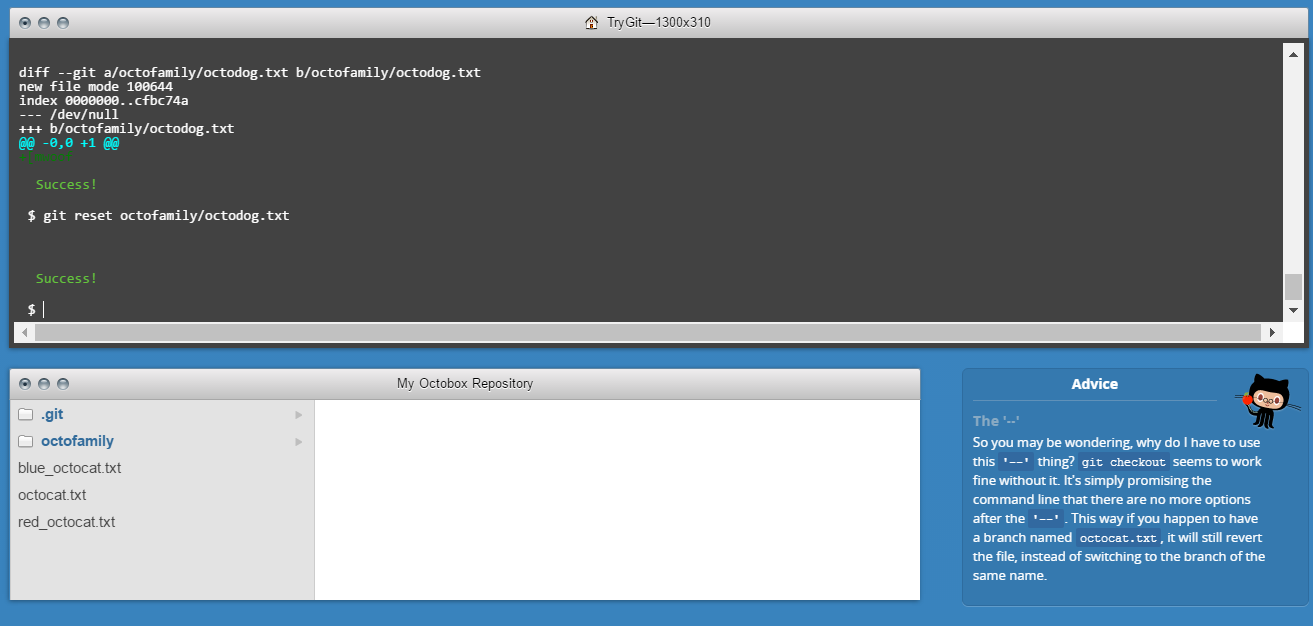
Step 15:

Staged Difference (Cont’d)



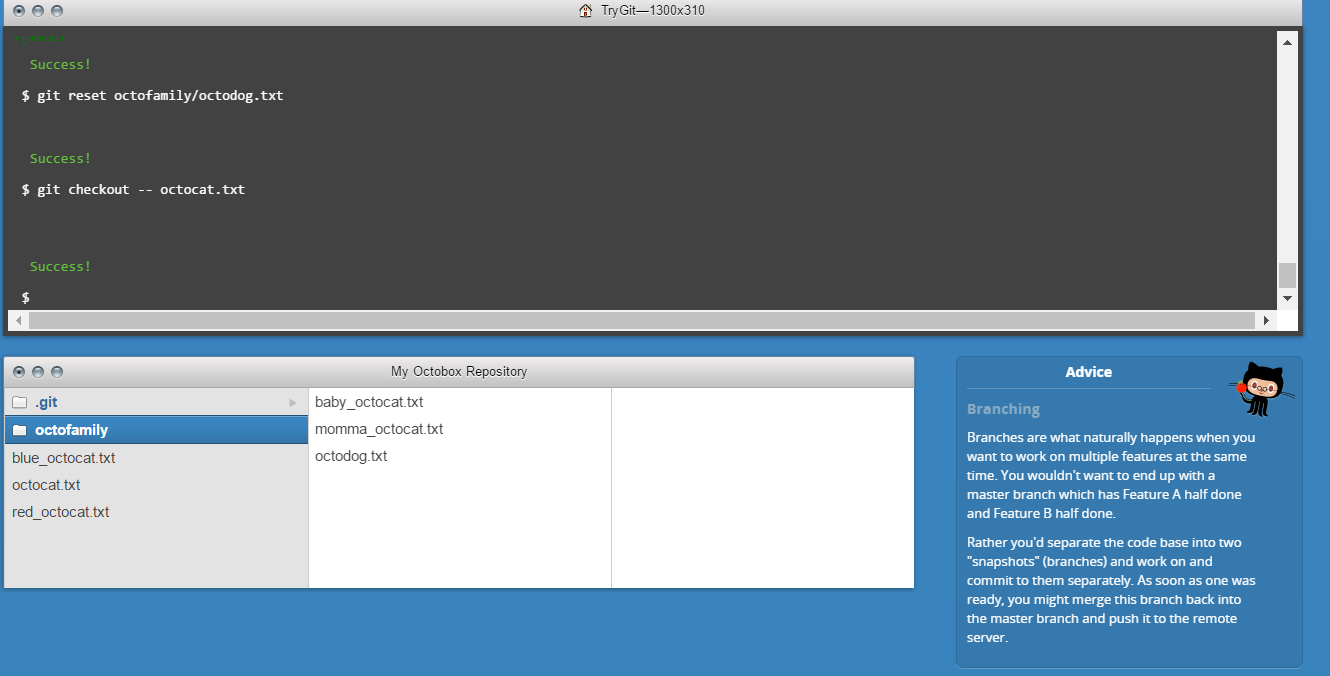
Step 16:

Resetting the stage



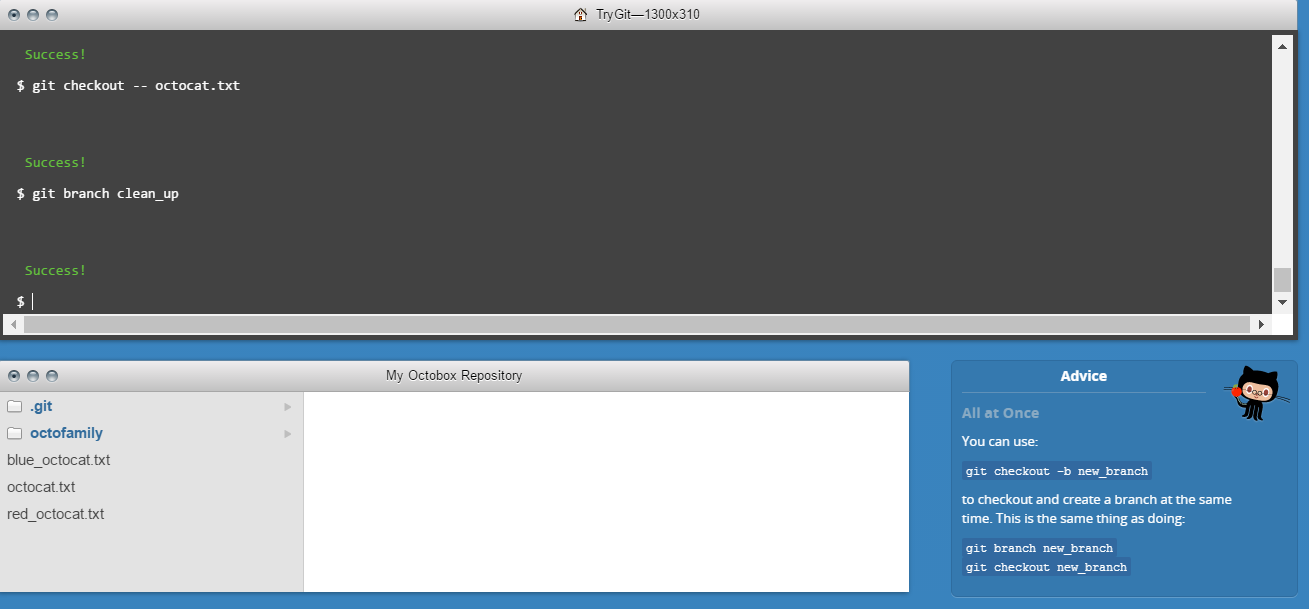
Step 17:

Undo



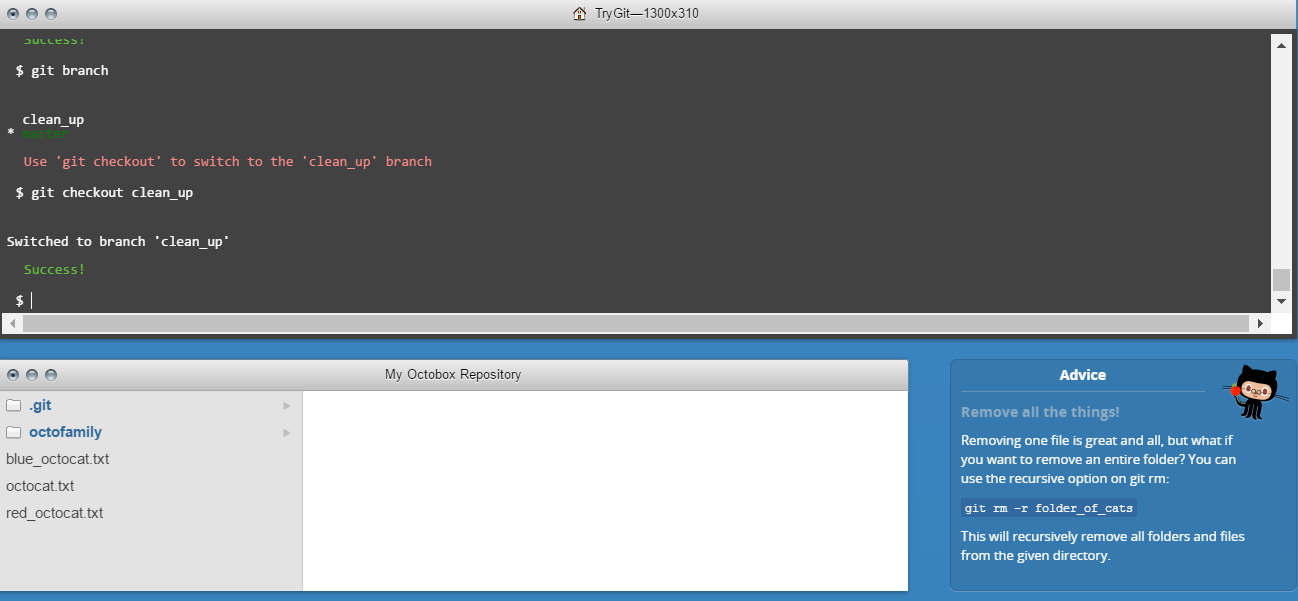
Step 18:

Branching out



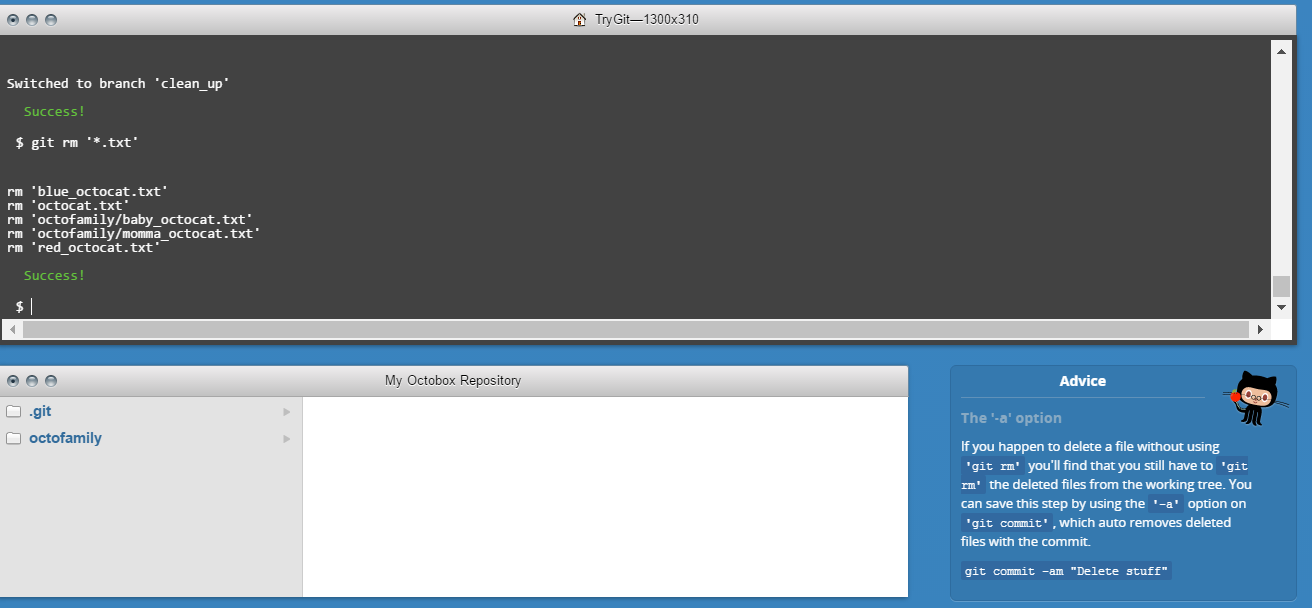
Step 19:

Switching Branches



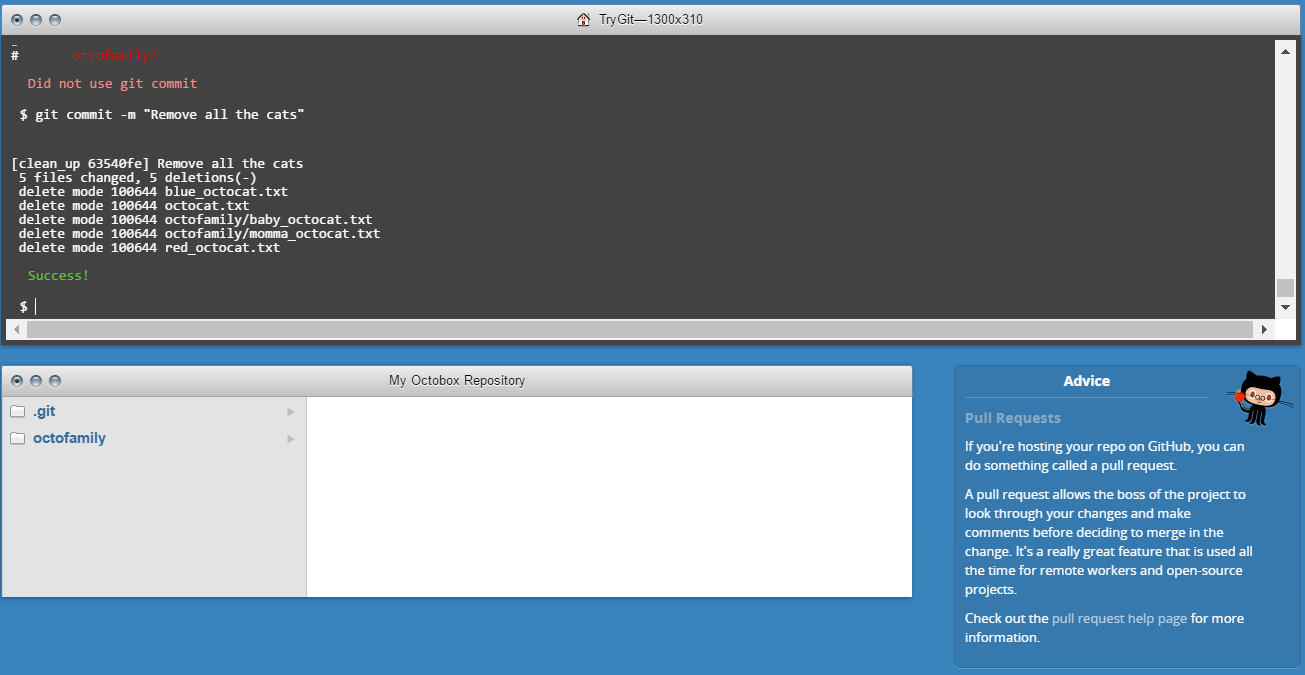
Step 20:

Removing all things



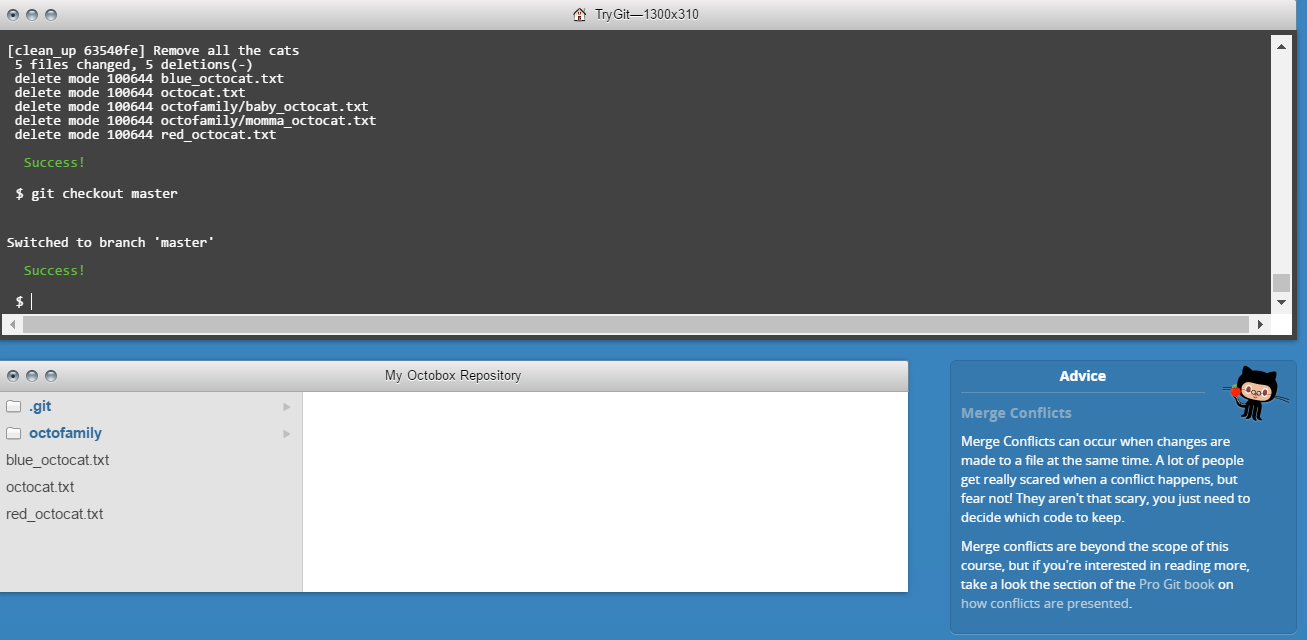
Step 21:

Committing Branch Changes



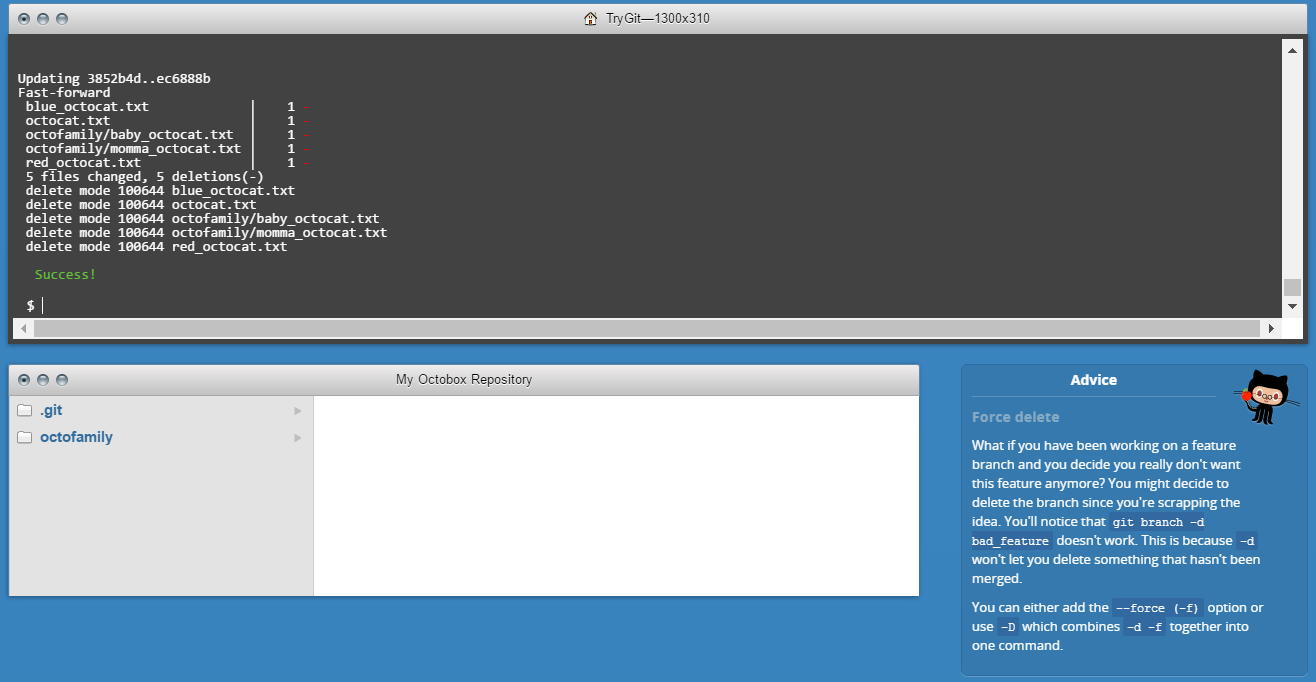
Step 22:

Switching back to master



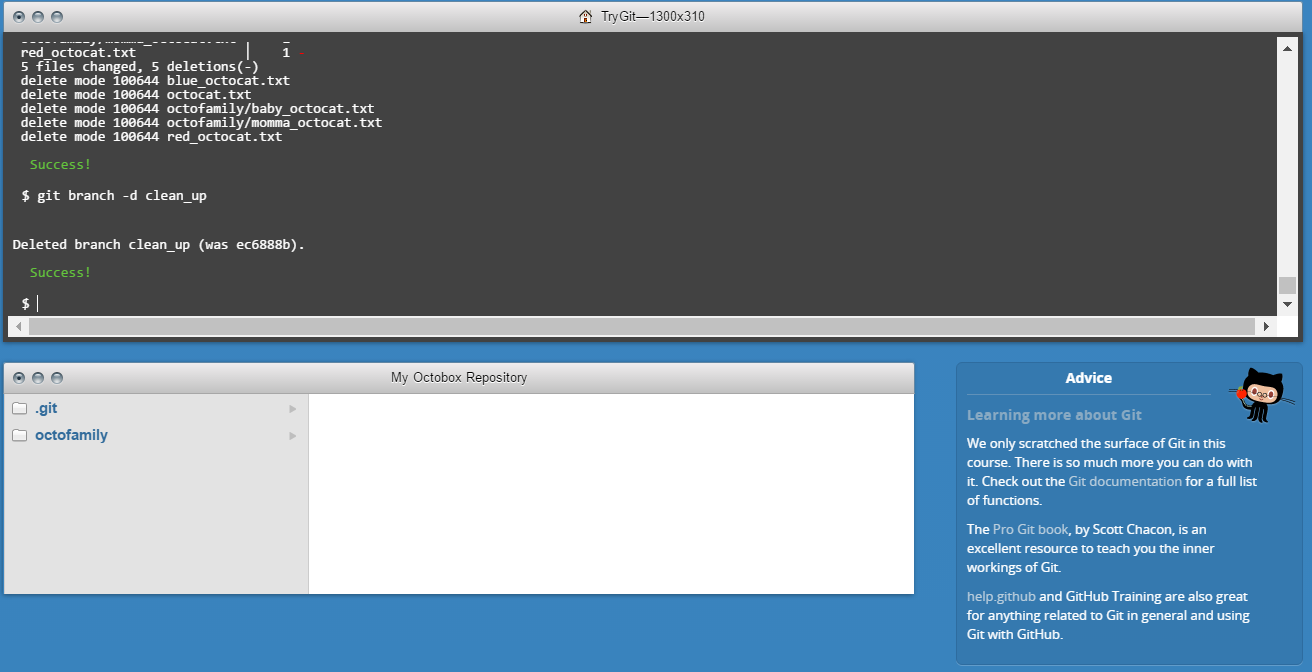
Step 23:

Preparing to merge



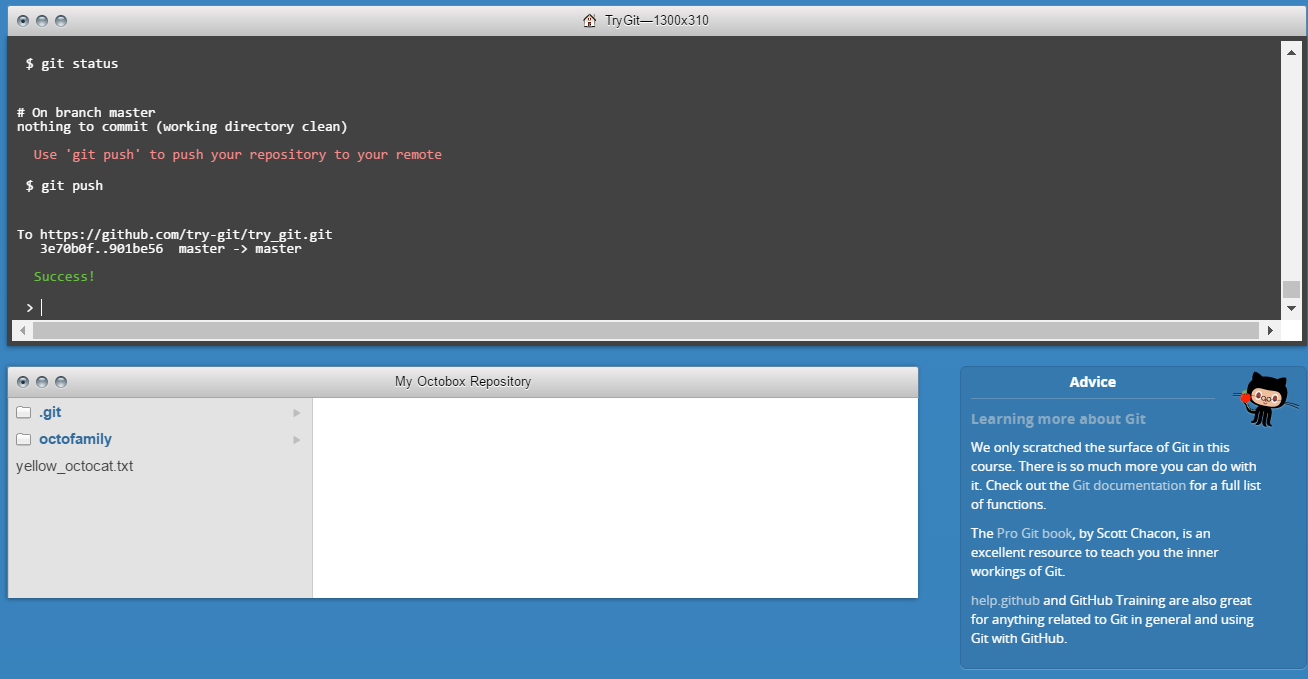
Step 24:

Keeping things clean



Step 25:

The final push



Q.2 Define the following terms in the context of Git (2 lines maximum):

Answer:

1. Repository:A repository contains all of the project files (including documentation), and stores each file's revision history.
2. Commit: A commit is an individual change to a file (or set of files). It creates an unique ID that allows you to keep record of what changes were made when and by who.
3. Branch: A branch is a similar version of a repository. It is contained within the repository, but does not affect the primary or master branch allowing you to work freely without disrupting the "live" version.
4. Fork: Forks allow you to freely make changes to a project without affecting the original. Forks remain attached to the original, allowing you to update the author with your changes.
5. Merge: Merging takes the changes from one branch (in the same repository or from a fork), and applies them into another.
6. Clone: Cloning a repo in simple words, “Creating a copy of a repo”. By this, developers/coders can make changes on the clone repo and check for different outputs.
7. Pull: Pull refers to fetching in changes and merging them. In its default mode, git pull is short hand for *git fetch*.
8. Pull request: Pull request contains the information about the changes made to a GitHub repo. A revision history can be checked by a Pull request.