Git — What, Why and How To Use It

Imagine you're cooking a delicious meal. Everything is almost ready, only the last bits are left on the stove when suddenly you get a call. It's your best friend whom you haven't spoken to in ages. Great, now you have good food and good company. Distracted by the phone and your bestie's general life drama, you realize that your precious meal is now burnt garbage. With your heart in shambles you wish that there was some to undo the done, to go back in time and save your food. Well your food might be beyond saving, but with Git at least your code can be reverted to the previous version.

What is Git ?

Git, in most basic terms, is a system that tracks changes across files and folders so that specific versions can be recalled later (like recalling the pre burnt version of food so you can cook it again properly). This system, also known as version control system, is used to view changes in the files over a period, compare versions, check who modified what among many other things.

Why Git ?

Git is a powerful tool that allows for open source and collaborative developments. Through branching, a process that allows developers to divert from the main project to add a bug or feature, multiple developers can work on different features at the same time without affecting the main project. Branches in git are easy and cheap to merge, unlike in centralized version control systems. This is what makes it an extremely popular choice for developers working on any kind of project.

Basic Terms—

Repository: defined as "a folder that can track changes", it is a directory that stores files, source codes and basically a collection of things whose history can be maintained and can be tracked.

Untracked: a place to modify, edit and create files

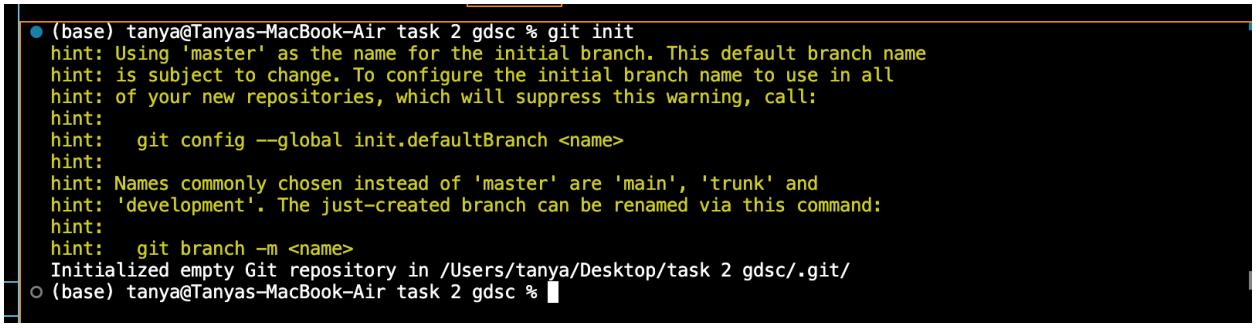
Commit: saves the changes of the file into the working repository

Tracked:

1. Staged: files that are ready to be committed to the repository.
2. Modified: when changes are made to a file, git sees them as modified and displays it.

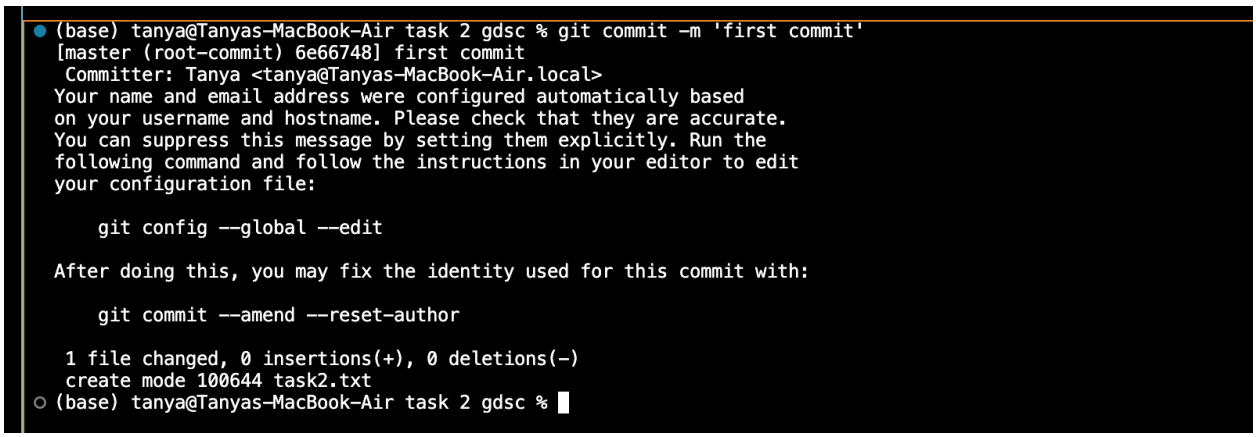
Git Commands—

Git init: to initialize a repository, git creates a hidden .git directory. This hidden repository is what separates regular directory from Git repository

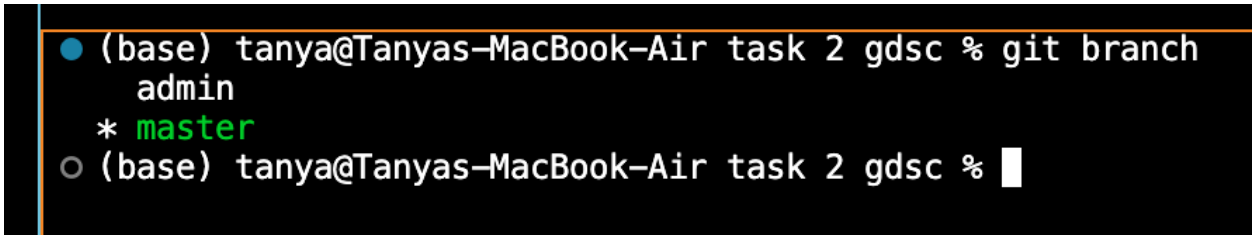


Git add: used to save changes to the file and push the file to the staging area

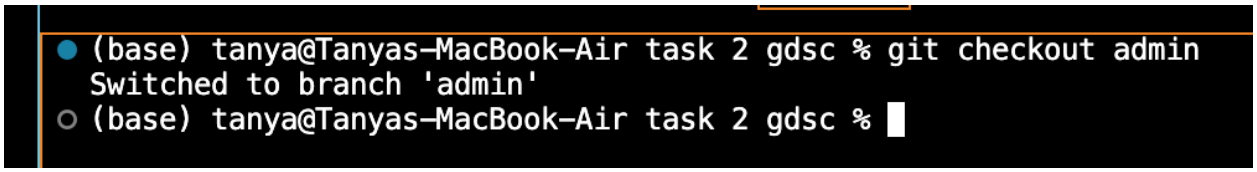
Git commit: creates a commit, basically a snapshot of the repository in its current state



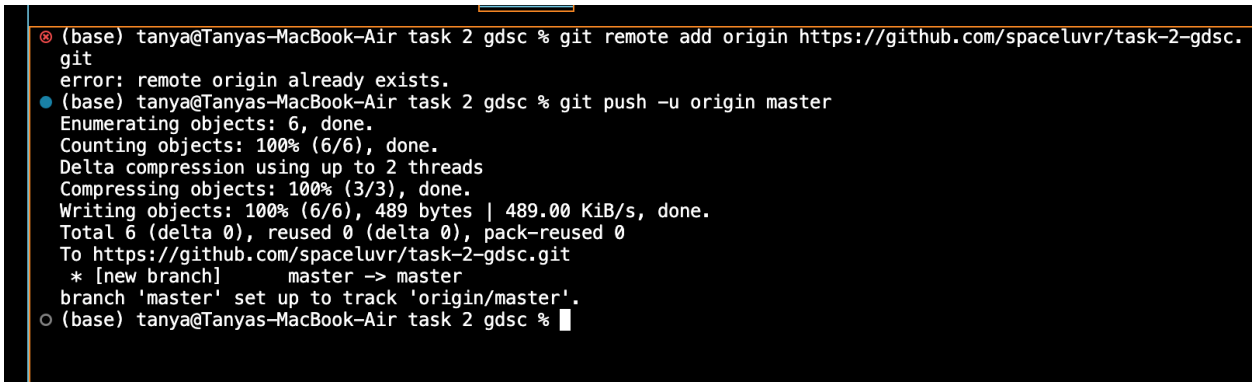
Git branch: used to create a new branch/check the names of existing branches



Git checkout: used to switch branches



Git push: used to upload all local files and commits to corresponding local branch.



Conclusion—

While Git might be intimidating to use at first, it is incredibly useful and makes development easier, flexible and faster. Once learnt, you will never want to use anything that's not Git.