GitHub

Tebogo Dube

Version Control

The meaning of version control is to continuously save your work and be able to refer back to old saves. As I write this file, I am constantly overwriting my changes and I am unable to go back to an older version I **commited** , so this is not version control, but if I was using GitHub, I would be able to check back to old drafts. There are two types of version control systems, centralized and distributed - GitHub being the latter.

Centralized version control system

This works by having one **central** copy of the project on a server, and having each individual commit their section of code to this one central copy, and other people working on the same project can see those changes in the central copy, and they can copy those changes to their workstation if they wish without needing to copy the whole project.

- The CVCS will automatic changes
- Access control - for security
- Copy the files you need only
- Takes a lot more coordination with your teams to avoid conflicts when merging

Distributed version control system

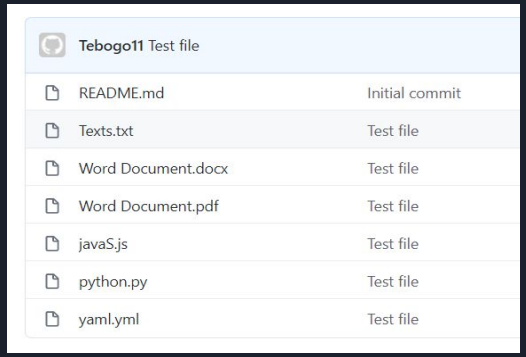
DVCS involves distributing clone of the metadata of the project(all files related to the project and its history) to the local repository of anyone who requests it,

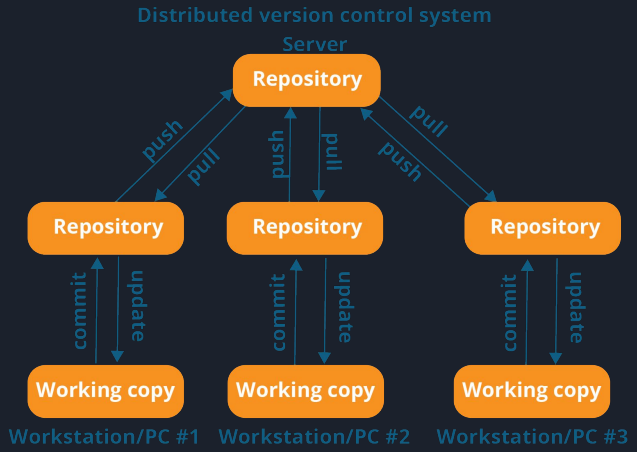
- Merging can be automated and is easier to handle, also branch is much easier
- You don't need to be connected to the network to have access to the main project
- Allows for personalized work without affecting the main project

Introduction - What is GitHub

GitHub is not just for code sharing and publishing service, it can be used for some much more, this has being said by McCullough a GitHub Trainer. It can be also used for managing and storing projects draft of various document types - in summary it's like a normal file system with features that make sharing work with other colleagues easier and more efficient.

GitHub is accurately referred to as a distributed version control system, meaning that the full codebase and version history is all mirror on the computer of everyone involved in the project. Allowing each developer to work on the project on their own machines anywhere they wish, as well as offline. This copies could be also called backups, as they are similar to the original copy on the central repository, with the users changes





GitHub and Git

GitHub and Git are two different components that complement each other,

Git was created by the Linux creator, and it runs in your command line, allowing you to keep track of your code, as well as create saves which you can go back to. It offers various functions in the command line to accomplish its task. Function such as :

- git init - Use this to create a place for git to store files.
- git add . - this command will add all the files to git, so git can start tracking them
- git commit - tells git to save a version of our code
- git diff - you can see all the changes between files with this command => (-) = Deletions / (+) =Insertions
- git checkout $ID - you can use git checkout to see old versions of code, the id can be found by calling ‘git log’

GitHub

GitHub is UI platform that servers as the origin for your codebase, its free and comes with various tools to help manage your development process such as a wiki to document your code as well as a user friendly web UI that allows you to traverse through the files in your codebase . Git actionsmis also a powerful tool in github which allows you automate process, like merging and deployment.

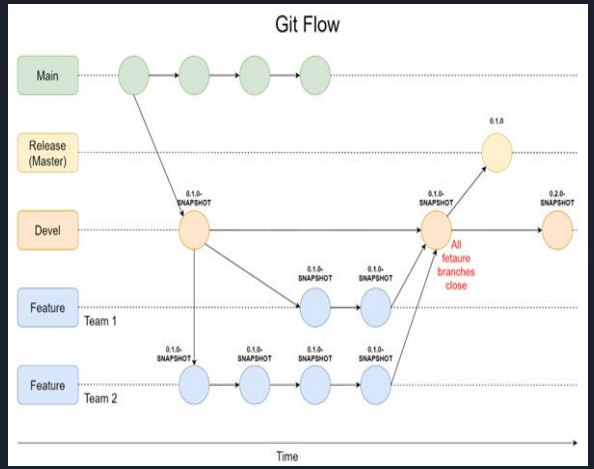
Branches

Branches

When working in a group Branches become a useful option to consider, it allows developers to split their working section away from the main code (where the most up-to-date working code can be kept). By doing this it allows developer to have version control on their own section of code, where they can build and test at will without affecting the origin/master - Then when they are ready they can easily Merge their code to the main branch (origin/master).

- Git checkout -b new-branch = Use this command to create or access a branch
- Git merge $branch name - to add the branch back to the origin use git branch to find branch name

Branch work better with pull request however, most definitely when in a group project



Pull Request

Pull request are a way of, managing request to add new code to your codebase when working on a group project. They work with Branches, by allowing an administrator to see what changes you are trying to add,- then they can review it and merge it if they consider it acceptable. Comments and Title can also be added to further explain what the code does and what it is trying to do.

All you will need to do on git is use the command - git push origin head (head meaning current branch your on). After that step you will see a notification on Github that stats ‘Compare and pull request’ , in this section you can add comments and a title, then simply click create pull request. Keeping in mind you can make pull request to more than just the main branch. Then it is as simple as leaving a review on the code, or just simply merging if their no problems.

Then you can use git pull origin $branchName to retrieve the changes added by the merge from the pull request

Reference

Finley, K., 2012. *Techcrunch Is Now A Part Of Verizon Media*. [online] Techcrunch.com. Available at: <https://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAB9jPNsnWqt-lLsca-46Ol0sW_u3NlzYwtGixbW94338FICGFCuTU9NPbJzyzrdg4MlrO_E1m40dd8FsmsgH2LgfoNcaRoiBiFJsQ9kXgVFDd_vRHoSkMt9DCmFz2yuTLs4VdiQHqINB6NZFAOybhbiKt3SgaoPiOWpnVyhqCRO6> [Accessed 14 July 2014].

Gehman, C., 2020. *What Is DVCS Anyway? | Perforce Software*. [online] Perforce Software. Available at: <https://www.perforce.com/blog/vcs/what-dvcs-anyway> [Accessed 25 July 2018].

GitHub. 2020. *Github: Where The World Builds Software*. [online] Available at: <https://github.com/> [Accessed 16 December 2020].

Lubański, M., 2019. *Centralized Vs Distributed Version Control Systems*. [online] Medium. Available at: <https://medium.com/faun/centralized-vs-distributed-version-control-systems-a135091299f0> [Accessed 30 April 2019].

Wilkie, A., 2020. *What Is Github? What Is Git? And How To Use These Developer Tools.*. [online] freeCodeCamp.org. Available at:

Reference

Wilkie, A., 2020. *What Is Github? What Is Git? And How To Use These Developer Tools.*. [online] freeCodeCamp.org. Available at: <https://www.freecodecamp.org/news/what-is-github-what-is-git-and-how-to-use-these-developer-tools/> [Accessed 5 May 2020].

Dear, M., 2020. *Gallenc/Solent6public*. [online] GitHub. Available at: <https://github.com/gallenc/solent6Public/wiki/Git-Flow-Diagram> [Accessed 28 November 2020].'

Dear, M., 2020. *Git Flow Diagram*. [image] Available at: <https://github.com/gallenc/solent6Public/wiki/Git-Flow-Diagram> [Accessed 28 November 2020].