Assignment-1Aug21F2020

I have found that most engineers do not have a deep appreciation and competence in git — they know enough to get by — which is also just enough to be dangerous. by<Burson Kendrick>

- 4 parts of storage in git (stash, stage, refspec, workspace)

- clone vs origin

- branch vs fork

- commit vs add

- fetch vs merge

git stash [list, drop, clear, diff…]

Stash-

git stash-

when you want to record the current state of the working directory and the index, but want to go back to a clean working directory. The command saves your local modifications away and reverts the working directory to match the HEAD commit.

git stash list-

The modifications stashed away by this command can be listed with git stash list,

git stash show-

Show the changes recorded in the stash entry as a diff between the stashed contents and the commit back when the stash entry was first created.

inspected with git stash show, and

git stash apply-

restored (potentially on top of a different commit) with git stash apply.

git stash push-

Calling git stash without any arguments is equivalent to git stash push.

refs/stash-

The latest stash you created is stored in refs/stash

git stash drop-

clear stash without applying it into working directory

git stash clear-

Remove all the stash entries. Note that those entries will then be subject to pruning, and may be impossible to recover

git diff-

Show changes between working directory and staging area.

the differences are what you could tell Git to further add to the index but you still haven’t. You can stage these changes by using git-add.

A close up of text on a black background

Description automatically generated

Staging-

The working area is where files that are not handled by git. These files are also referred to as "untracked files." Staging area is files that are going to be a part of the next commit, which lets git know what changes in the file are going to occur for the next commit. ... They can also be called untracked files.

Staging is a step before the commit process in git. That is, a commit in git is performed in two steps: staging and actual commit.

The staging area (aka index) is a container where git collects all changes which will be part of the next commit. If you are editing a versioned file on your local machine, git recognizes that your file is modified - but it will not be automatically part of your next commit and is therfore unstaged.

Refspec-

A refspec maps a branch in the local repository to a branch in a remote repository. This makes it possible to manage remote branches using local Git commands and to configure some advanced git push and git fetch behavior. ... Refspecs can be used with the git push command to give a different name to the remote branch.

git push some\_repo some\_branch

Ex: git push angular-ui sprint-19

angular-ui=remote repo

sprint 19 = remote branch in remote repo, if it's not available, it will create one.

git push some\_repo local\_branch:refs/heads/new\_branch

Ex: git push angular-ui sprint\_local\_19:refs/heads/sprint\_20

angular-ui=remote repo

sprint\_local\_19= local branch

refs/heads/sprint\_20= remote branch in remote repo, if it's not available, it will create one.

git clone <origin>

git clone ssh:angular/example.git

Workspace-

A workspace is simply a set of git repositories. They are listed in the "Repos" tab in a main window's left pane. You can have several different sets of repositories, each one stored in a different workspace file.

Having an option to group sets of git repositories is useful when you have unrelated sets which you may want to keep separate

clone vs origin

clone-

Downloads a project with entire history from the remote repository

cloning is a process. Cloning is done through the command 'git clone' and it is a process of receiving all the code files to the local machine.

Origin-

"origin" is a shorthand name for the remote repository that a project was originally cloned from. More precisely, it is used instead of that original repository's URL - and thereby makes referencing much easier. Note that origin is by no means a "magical" name, but just a standard convention.

“origin” is the default name for a remote when you run git clone.

branch vs fork

branch-

create new branches , list the new branches

fork-

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea.

Forking is a concept while cloning is a process. Forking is just containing a separate copy of the repository and there is no command involved.

commit vs add

commit-

a commit adds the latest changes to [part of] the source code to the repository,

create new commit , changes added from staging area .Commit must have a message.

Add-

The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit. However, git add doesn't really affect the repository in any significant way—changes are not actually recorded until you run git commit

fetch vs merge

fetch-

Fetch changes from the remote but not update the tracking branches

merge-

join specified branch in to your current branch,

The "merge" command is used to integrate changes from another branch.

References-

<https://git-scm.com/docs>

<https://sites.google.com/site/gitforcetool/help/workspaces#:~:text=Concept%20of%20a%20Workspace&text=A%20workspace%20is%20simply%20a%20set%20of%20git%20repositories.&text=You%20can%20have%20several%20different,may%20want%20to%20keep%20separate.>

source:

endpoint.com/blog/2008/07/30/git-push-know-your-refspecs