It  **pull** and **fetch**are two commands that are regularly used by Git users. Let’s see the difference between both commands.

For the sake of context, it’s worth remembering that we’re probably working in a clone repo. What’s a clone? It's simply a duplicate of another repository. It is basically getting your own copy of someone else’s source code.

That said, to keep your clone updated with whatever changes may have been applied to the original, you’ll need to bring those to your clone.

That’s where fetch and pull come in.

git fetch is the command that tells your local git to retrieve the latest meta-data info from the original (yet doesn’t do any file transferring. It’s more like just checking to see if there are any changes available).

git pull on the other hand does that AND brings (copy) those changes from the remote repository.

For example:

**git pull origin ankur bugfix**

The takeaway is to keep in mind that there generally are at least three copies of a project on your workstation.

* One copy is your own repository with your own commit history (the already saved one, so to say).
* The second copy is your working copy where you are editing and building (not committed yet to your repo).
* The third copy is your local “cached” copy of a remote repository (probably the original from where you cloned yours).

You can use git fetch to know the changes done in the remote repo/branch since your last pull. This is useful to allow for checking before doing an actual pull, which could change files in your current branch and working copy (and potentially lose your changes, etc).

git fetch

git diff ...origin