

Useful Git Commands For Removing Accidentally Pushed or Committed Files



Felice Forby



Jul 22 Updated on Jul 23, 2019 · 3 min read

#git #beginners

I find myself needing to reset commits often because I committed a file I didn't want to or forgot to add something to a commit. Every once in a while, I also need to remove the file from the remote repository (but not locally) because I accidentally pushed it. I always forget the commands, so here they are for everyone's reference :)

Remove a file you already pushed to the git repository

You can use the following command to delete a file from your git repo, for example, if you accidentally pushed it up or if you just don't want it there any anymore.

```
git rm --cached name_of_file
```

This will not delete it locally, so it is safe on your computer if you want to keep it in there for reference without sharing on Git. To prevent it from being pushed to Git again, just add the file to your `.gitignore`.

Of course, if you just don't need the file any more at all, simply delete it from your system as well.

BUT! If the file had sensitive data like passwords or secret keys, then you need to do a little bit more because the commits can still be found in the repository's history.

If the pushed file has sensitive data like passwords

As noted by [Ölbaum's comment](#) below (thanks!), the above method will not *completely* remove the file from the commit history.

To remove the sensitive file from your history as well, you can use an open-source tool called the [BFG Repo-Cleaner](#) or use git's `git filter-branch`.

BFG Repo-Cleaner apparently makes it much easier and faster to get the job done compared to `filter-branch`. It allows you to delete files or, alternatively, replace the passwords or keys within the file with the text `***REMOVED***`. You can also remove huge files that you accidentally pushed.

I haven't used BFG myself, but check out [BFG Repo-Cleaner's documentation and download page](#) to see how to use it. The instructions are pretty strait forward.

To learn about the `git filter-branch`, check out git's [Using filter-branch](#) article on their help site and the [git filter-branch documentation](#).

Remove a file you accidentally committed in your last commit (but haven't pushed yet)

I have a habit of committing everything with `git add .` so sometimes I accidentally add files that I actually want to have in a separate commit.

To remove certain files from a commit that hasn't been pushed yet, first, undo the last commit with:

```
git reset --soft HEAD^1
```

Next, run:

```
git rm --cached <file-name>
```

to remove the file you don't want to commit. This removes it from the commit and sets it back to an untracked file. You should be able to confirm by doing a quick `git status`.

Now you can commit the other files as usual by running:

```
git commit -m "my commit message"
```

The removed file or files can then be added into a separate commit!

