

## How do I delete a Git branch locally and remotely?

[Ask Question](#)

▲  
15066



4878

I want to delete a branch both locally and remotely.

### Failed Attempts to Delete Remote Branch:

```
$ git branch -d remotes/origin/bugfix
error: branch 'remotes/origin/bugfix' not found.

$ git branch -d origin/bugfix
error: branch 'origin/bugfix' not found.

$ git branch -rd origin/bugfix
Deleted remote branch origin/bugfix (was 2a14ef7).

$ git push
Everything up-to-date

$ git pull
From github.com:gituser/gitproject
* [new branch] bugfix -> origin/bugfix
Already up-to-date.
```

What should I do differently to successfully delete the remotes/origin/bugfix branch both locally and remotely?

[git](#)
[git-branch](#)
[git-remote](#)

edited May 2 at 18:51



[alec\\_a](#)

1

asked Jan 5 '10 at 1:12



[Matthew Rankin](#)

276k 34 104 147

882 **Moderator note: If you intend to answer this question, do note that there are already 40 answers posted. Will your new answer add any substantial value? –**

[Robert Harvey](#) ♦ Jun 11 '14 at 16:10



48 Note: for Git 2.5+ (Q2 2015), the exact message will be "deleted remote-tracking branch ": see [github.com/git/git/commit/...](https://github.com/git/git/commit/...) – [VonC](#)  
May 25 '15 at 14:57

39 Answers

1 2 next



## Executive Summary

18828



Note that in most cases the remote name is `origin`.

## Delete Local Branch

To delete the *local* branch use one of the following:

```
$ git branch -d branch_name  
$ git branch -D branch_name
```

**Note:** The `-d` option is an alias for `-delete`, which only deletes the branch if it has already been fully merged in its upstream branch. You could also use `-D`, which is an alias for `--delete --force`, which deletes the branch "irrespective of its merged status." [Source: `man git-branch`]

## Delete Remote Branch [Updated on 8-Sep-2017]

As of [Git v1.7.0](#), you can delete a **remote** branch using

```
$ git push <remote_name> --delete <br>
```

which might be easier to remember than

```
$ git push <remote_name> :<branch_name>
```

which was added in [Git v1.5.0](#) "to delete a remote branch or a tag."

Starting on [Git v2.8.0](#) you can also use `git push` with the `-d` option as an alias for `--delete`.

Therefore, the version of Git you have installed will dictate whether you need to use the easier or harder syntax.

## Delete Remote Branch [Original Answer from 5-Jan-2010]

From Chapter 3 of [Pro Git](#) by Scott Chacon:

## Deleting Remote Branches

Suppose you're done with a remote branch — say, you and your collaborators are finished with a feature and have merged it into your remote's master branch (or whatever branch your stable code-line is in). You can delete a remote branch using the rather obtuse syntax `git push [remotename] :[branch]`. If you want to delete your `server-fix` branch from the server, you run the following:

```
$ git push origin :serverfix
To git@github.com:schacon/simplegit
- [deleted]          serverfix
```

Boom. No more branch on your server. You may want to dog-ear this page, because you'll need that command, and you'll likely forget the syntax. A way to remember this command is by recalling the `git push [remotename] [localbranch]:[remotebranch]` syntax that we went over a bit earlier. If you leave off the `[localbranch]` portion, then you're basically saying, "Take nothing on my side and make it be `[remotebranch]`."

I issued `git push origin :bugfix` and it worked beautifully. Scott Chacon was right—I will want to [dog ear](#) that page (or virtually dog ear by answering this on Stack Overflow).

Then you should execute this on other machines

```
git fetch --all --prune
```

to propagate changes.

edited Dec 13 '18 at 4:45



[Alive to Die](#)

56.6k 8 32 73

answered Jan 5 '10 at 1:13



[Matthew Rankin](#)

276k 34 104 147

574 If you know the syntax `git push origin local_branch:remote_branch`, then the syntax to delete a branch with `git push origin :remote_branch` is kind of cute. There's a void before the `:` – [Marc-André Lafortune](#) May 11 '12 at 4:05

261 Don't forget to do a `git fetch --all --prune` on other machines after deleting the remote branch on the server. ||| After deleting the local branch with `git branch -d` and deleting the remote branch with `git push origin --delete other` machines may still have "obsolete tracking branches" (to see them do `git branch -a`). To get rid of these do `git fetch --all --prune .` – [Trevor Boyd Smith](#) May 27 '15 at 16:51

7 in addition to [@TrevorBoydSmith's](#) `git branch -a` to view all branches, you can also use `git branch -r` to view remote branches only. see also `git remote show origin -source:` [gitready.com/intermediate/2009/02/13/list-remote-branches.html](http://gitready.com/intermediate/2009/02/13/list-remote-branches.html) – [Sandra](#) Sep 9 '15 at 9:53

12 I had to run `git branch -D Branch_Name` to get rid of the local branch – [Kolob Canyon](#) Mar 10 '16 at 1:32

5 [@KolobCanyon](#) You only have to use `-D` if the branch has not been merged into another branch. – [BTRUE](#) Apr 5 '16 at 16:27



3008

Matthew's answer is great for removing *remote* branches and I also appreciate the explanation, but to make a simple distinction between the two commands:

To remove a **local branch** from your machine:

```
git branch -d {the_local_branch} (use
-D instead to force deleting the branch
without checking merged status)
```

To remove a **remote branch** from the server:

```
git push origin --delete
{the_remote_branch}
```

Reference:

<https://makandracards.com/makandra/621-git-delete-a-branch-local-or-remote>

edited Dec 18 '17 at 12:33



Rin Minase

75 2 14

answered Jun 12 '12 at 14:51



Eric Brotto

36.5k 27 108 168

232 @megido well -D force deletes, -d gives you a warning if it's not already merged in. – [TankorSmash](#) Nov 6 '14 at 19:04

7 If your local branch is not merge with master and ran 'git branch -d your\_branch' then you will error like error: The branch 'your\_branch' is not fully merged. If you are sure you want to delete it, run 'git branch -D your\_branch'. – [geeks](#) Oct 31 '15 at 12:59

10 I would suggest using -d instead of -D because it is safer. If -d fails due to unmerged commits then you need to assess that and if it is definitely OK to remove then use -D. – [rooby](#) Feb 2 '16 at 3:47

5 Others with repository clones where remote branches have been removed should run `git remote prune <name>` (e.g. `git remote prune origin`) in order to locally remove stale branches that no longer exist in the remote. – [code\\_dredd](#) Apr 18 '16 at 23:07

1 I would like to add that -d gives a warning if it isn't merged in with the current HEAD. If you need clarity I recommend this command `git branch -a --merged origin/master` It will list any branches, both local and remote; that have been merged into master. [Additional information here](#) – [Eric](#) Feb 10 '17 at 13:10

## ▲ The Short Answers

1736



If you want more detailed explanations of the following commands, then see the long answers in the next section.

Deleting a remote branch:

```
git push origin --delete <branch> # Gi
git push origin :<branch>         # Gi
```

Deleting a local branch:

```
git branch --delete <branch>
git branch -d <branch> # Shorter versio
git branch -D <branch> # Force delete u
```

Deleting a local remote-tracking branch:

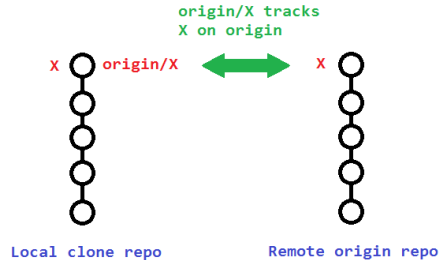
```
git branch --delete --remotes <remote>/
git branch -dr <remote>/<branch> # Shor

git fetch <remote> --prune # Delete mul
git fetch <remote> -p      # Shorter
```

## The Long Answer: there are 3 different branches to delete!

When you're dealing with deleting branches both locally and remotely, keep in mind that **there are 3 different branches involved**:

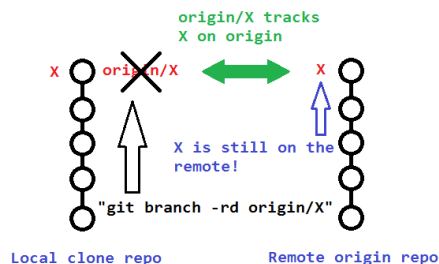
1. The local branch `x`.
2. The remote origin branch `x`.
3. The local remote-tracking branch `origin/X` that tracks the remote branch `x`.



The original poster used

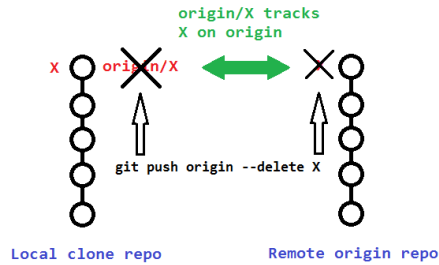
```
git branch -rd origin/bugfix
```

which only deleted his **local remote-tracking branch** `origin/bugfix`, and not the actual remote branch `bugfix` on `origin`.



To delete that actual remote branch, you need

```
git push origin --delete bugfix
```



## Additional Details

The following sections describe additional details to consider when deleting your remote and remote-tracking branches.

### Pushing to delete remote branches also deletes remote-tracking branches

Note that deleting the remote branch `x` from the command line using a `git push` **will also delete the local remote-tracking branch** `origin/X`, so it is not necessary to prune the obsolete remote-tracking branch with `git fetch --prune` Or `git fetch -p`, though it wouldn't hurt if you did it anyway.

You can verify that the remote-tracking branch `origin/X` was also deleted by running the following:

```
# View just remote-tracking branches
git branch --remotes
git branch -r

# View both strictly local as well as r
git branch --all
git branch -a
```

### Pruning the obsolete local remote-tracking branch `origin/X`

If you didn't delete your remote branch `x` from the command line (like above), then your local repo will still contain (a now obsolete) remote-tracking branch `origin/X`. This can happen if you deleted a remote branch directly

through GitHub's web interface, for example.

A typical way to remove these obsolete remote-tracking branches (since Git version 1.6.6) is to simply run `git fetch` with the `--prune` or shorter `-p`.

**Note that this removes all obsolete local remote-tracking branches for any remote branches that no longer exist on the remote:**

```
git fetch origin --prune
git fetch origin -p # Shorter
```

Here is the relevant quote from the [1.6.6 release notes](#) (emphasis mine):

**"git fetch" learned** `--all` and `--multiple` options, to run `fetch` from many repositories, and `--prune` **option to remove remote tracking branches that went stale**. These make "git remote update" and "git remote prune" less necessary (there is no plan to remove "remote update" nor "remote prune", though).

### Alternative to above automatic pruning for obsolete remote-tracking branches

Alternatively, instead of pruning your obsolete local remote-tracking branches through `git fetch -p`, *you can avoid making the extra network operation* by just manually removing the branch(es) with the `--remote` or `-r` flags:

```
git branch --delete --remotes origin/X
git branch -dr origin/X # Shorter
```

### See Also

- [git-branch\(1\) Manual Page](#).
- [git-fetch\(1\) Manual Page](#).
- [Pro Git § 3.5 Git Branching - Remote Branches](#).

edited Mar 10 '16 at 12:20



Valentin Lorentz

6,962 5 35 55

answered May 30 '14 at 18:32



- 
- 5 @huggie that's pretty much correct. Branches in Git are just bookmarks attached to commits. So in my graphs above, there are `X` and `origin/X` bookmarks in the local clone (2 branches), and then there is `X` on the remote (making 3 branches). – user456814 Feb 23 '16 at 7:33
- 
- 6 +1 for the remote tracking branch. This branch is what causes issues when you clone someone else's branch. It keeps on tracking your commits and asking you if you want to push to that person's branch. – [Kermit\\_ice\\_tea](#) Jun 21 '17 at 19:48
- 



### Steps for deleting a branch:

#### 1212 For deleting the **remote branch**:

`git push origin --delete <your_branch>`

For deleting the **local branch**, you have **three ways**:

- 1: `git branch -D <branch_name>`
- 2: `git branch --delete --force <branch_`
- 3: `git branch --delete <branch_name>`

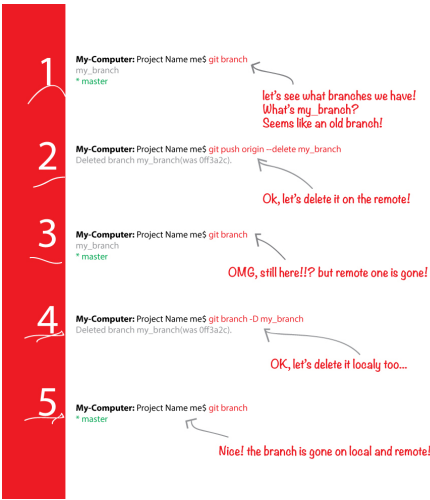
**Explain:** OK, just explain what's going on here!

Simply do `git push origin --delete` to **delete your remote branch ONLY**, add the name of the branch at the end and this will delete and push it to remote at the same time...

Also, `git branch -D`, which simply delete the local branch **ONLY!**...

`-D` stands for `--delete --force` which will delete the branch even it's not merged(force delete), but you can also use `-d` which stands for `--delete` which throw an error respective of the branch merge status...

I also create the image below to show the steps:



edited Mar 22 at 7:08

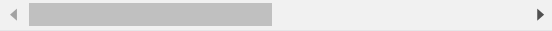
answered Jun 27 '17 at 13:13



Alireza

54.3k 14 183 126

34 git branch -a will display local and remote branches. It will be helpful for you to introduce. – LoranceChen Jul 27 '17 at 3:01



745 You can also use the following to delete the remote branch

git push --delete origin serverfix

+300 Which does the same thing as

git push origin :serverfix

but it may be easier to remember.

edited Dec 13 '18 at 4:44



Alive to Die

56.6k 8 32 73

answered Oct 27 '11 at 22:22



pagetrib

12.9k 3 20 18



356 If you want to delete a branch, first checkout to the branch other than the branch to be deleted.

`git checkout other_than_branch_to_be_del`

Deleting the local branch:

`git branch -D branch_to_be_deleted`

Deleting the remote branch:

`git push origin --delete branch_to_be_de`

edited Jun 9 '15 at 2:04



Willem Van Onsem

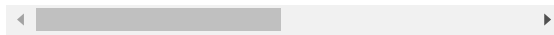
153k 17 158 245

answered Oct 7 '14 at 13:52



Praveen Hiremath

3,658 2 12 10



Tip: When you delete branches using

352

`git branch -d <branchname> # deLetes Loc`

or

`git push origin :<branchname> # deLetes`

only the references are deleted. Even though the branch is actually removed on the remote the references to it still exists in the local repositories of your team members. This means that for other team members the deleted branches are still visible when they do a `git branch -a`.

To solve this your team members can prune the deleted branches with

`git remote prune <repository>`

This is typically `git remote prune origin`.

edited Dec 13 '18 at 4:44



Alive to Die

56.6k 8 32 73

answered Nov 7 '12 at 13:02



pfrenssen

4,630 1 16 15

14 You should clarify that the above `git push` operation deletes the local branch *and* the remote branch. —


Annika Backstrom May 21 '13 at 13:51

[Jonathon Reinhart](#) May 21 '13 at 10:31

- 18 Note that `git remote prune` is a somewhat obsolete way to remove obsolete remote-tracking branches, the newer way to do it is to use `git fetch -prune` or `git fetch -p .` – [user456814](#) Jun 11 '14 at 16:30

- 1 @RRMadhav, indeed you won't see the deleted branch after deleting it since the reference to the remote branch will be removed for you locally. Anyone else on your team that has checked out that branch will still have that reference and will still see it unless they prune the branch. – [pfrenssen](#) Dec 5 '14 at 14:27



256  `git branch -D <name-of-branch>`  
`git branch -D -r origin/<name-of-branch>`  
`git push origin :<name-of-branch>`

[edited Aug 8 '14 at 20:04](#)

answered Oct 24 '13 at 17:36


[Felipe](#)

11k 7 52 80

- 6 Note that `-D` *forces* the deletion. It's always better to use `-d`, which will remind if you need to do something dangerous. – [Jonathon Reinhart](#) Jan 10 '15 at 1:08 ✎

- 11 ahahah :) it's up to you: use `-d` if you want to see git crying or `-D` if you want to cry. – [Felipe](#) Feb 13 '15 at 11:21



 This is simple: Just run the following command:

220  To delete a Git branch both locally and remotely, first delete the local branch using command:

`git branch -d example`(here `example` is the branch name)

And after that delete remote branch using command:

`git push origin :example`

edited Jun 9 '15 at 2:03



Willem Van Onsem

153k 17 158 245

answered Feb 15 '15 at 15:20



Syeful Islam

3,039 1 12 17



▲ Another approach is:-

191

```
git push --prune origin
```



**WARNING:** This will delete all remote branches that do not exist locally. Or more comprehensively,

```
git push --mirror
```

will effectively make the remote repository look like the local copy of the repository (local heads, remotes and tags are mirrored on remote).

edited Dec 13 '18 at 4:44



Alive to Die

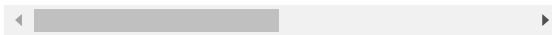
56.6k 8 32 73

answered Nov 18 '12 at 6:11



immanuelcostigan

2,523 3 11 17



▲ I use the following in my [Bash](#) settings:

160

```
alias git-shoot="git push origin --delete
```



Then you can call:

```
git-shoot branchname
```

edited Dec 13 '18 at 4:43



Alive to Die

56.6k 8 32 73

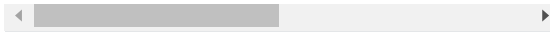
answered Apr 2 '13 at 22:11



crizCraig

5,353 3 36 44

10 I ended up just add the alias "shoot" into my .gitconfig shoot = push origin --delete  
– [hdost](#) Dec 4 '14 at 18:06



127 Since January 2013, GitHub included a  button next to each branch in your "Branches" page.

Relevant blog post: [Create and delete branches](#)

edited Dec 13 '18 at 4:42



[Alive to Die](#)

56.6k 8 32 73

answered Apr 29 '13 at 10:39



[Nacho Coloma](#)

4,851 1 31 40

- 1 I only started using Github this year, so I was wondering why this was such a highly rated question, and why none of the top answers were suggesting to just delete it from the Github web interface! Interesting that it's only a recent addition.  
– [Cam Jackson](#) Sep 11 '13 at 12:18
- 4 I was going to point this one out. Note that the button won't delete your local branch... see this answer for how to do that:  
[stackoverflow.com/a/10999165/901641](https://stackoverflow.com/a/10999165/901641) – [ArtOfWarfare](#) Oct 29 '13 at 14:02

119 If you want to complete both these steps with a single command, you can make an alias for it by adding the below to your `~/.gitconfig` :

```
[alias]
  rmbranch = "!f(){ git branch -d ${1}}
```

Alternatively, you can add this to your global config from the command line using

```
git config --global alias.rmbranch \
'!f(){ git branch -d ${1} && git push or
```

**NOTE:** If using `-d` (lowercase d), the branch will only be deleted if it has been merged. To force the delete to happen, you will need to use `-D` (uppercase D).

edited Jun 11 '14 at 16:26

[user456814](#)

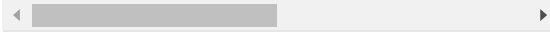
answered Jan 15 '13 at 19:05



[Ryan Kohn](#)

8,048 10 45 78

- 4 This is what I was looking for. My own shell function alias didn't work (Unexpected EOF) and I couldn't figure out why, but this works great! The only change I made was replacing `&&` with `;` so that even if the first command fails the second will still execute (sometimes only local or only remote exists). – [user1021726](#) Dec 16 '14 at 8:55



### ▲ Delete locally:

118 To delete a local branch, you can use:

▼ `git branch -d <branch_name>`

To delete a branch forcibly, use `-D` instead of `-d`.

`git branch -D <branch_name>`

### Delete remotely:

There are two options:

`git push origin :branchname`

`git push origin --delete branchname`

I would suggest you use the 2nd way as it is more intuitive.

[edited Dec 13 '18 at 4:45](#)



[Alive to Die](#)

56.6k 8 32 73

answered May 23 '15 at 8:56



[Rahul Gupta](#)

29.2k 9 56 73



### ▲ To delete your branch locally and remotely

112

- ▼
- Checkout to master branch - `git checkout master`
  - Delete your remote branch - `git push origin --delete <branch-name>`
  - Delete your local branch - `git branch --delete <branch-name>`

[edited Dec 13 '17 at 22:10](#)

[dippas](#)

39k 10 52 81

answered Jan 3 '16 at 21:08

[mfathy00](#)

1,395 1 9 23

106

You can also do this using `git remote prune origin`

```
$ git remote prune origin
Pruning origin
URL: git@example.com/yourrepo.git
* [pruned] origin/some-branchs
```

It prunes and deletes remote-tracking branches from a `git branch -r` listing.

edited Dec 13 '18 at 4:43

[Alive to Die](#)

56.6k 8 32 73

answered Mar 12 '13 at 14:57

[nickleefly](#)

3,065 1 21 29

102

In addition to the other answers, I often use the [git\\_remote\\_branch](#) tool. It's an extra install, but it gets you a convenient way to interact with remote branches. In this case, to delete:

```
grb delete branch
```

I find that I also use the `publish` and `track` commands quite often

edited Dec 13 '18 at 4:41

[Alive to Die](#)

56.6k 8 32 73

answered Mar 24 '12 at 2:21

[u2622](#)

2,561 3 19 26

92

**One liner** command delete *both local, and remote*:

```
D=branch-name; git branch -D $D; git
push origin :$D
```



or add the alias below to your

`~/.gitconfig`; usage: `git kill branch-name`

```
[alias]
  kill = "!f(){ git branch -D \"$1\";
```

edited Jan 15 '17 at 8:30

answered Nov 17 '16 at 1:03



[Vinnie James](#)

2,527 1 24 32

- 16 ⚠ Use `git branch -D` carefully in a script, since it force-deletes a branch without checking it has been merged. Use `-d` to be safe. – [caesarsol](#) Mar 13 '17 at 14:05



## [Deleting Branches](#)

90

Let's assume our work on branch "contact-form" is done and we've already integrated it into "master". Since we don't need it anymore, we can delete it (locally):

```
$ git branch -d contact-form
```

And for deleting the remote branch:

```
git push origin --delete contact-form
```

edited Nov 18 '15 at 19:51

answered Oct 30 '15 at 12:39



[Ulysses Alves](#)

1,352 1 11 25



## Delete remote branch

86

```
git push origin :<branchname>
```

## Delete local branch

```
git branch -D <branchname>
```

Delete local branch steps:

1. checkout to another branch
2. delete local branch

edited Feb 19 '16 at 18:05

answered Dec 31 '15 at 9:22

 [jayxhj](#)  
1,507 11 20

- 1 Does the remote branch deletion requires "git push" afterwards ? – [Samitha Chathuranga](#) Feb 17 '16 at 8:32

Simply say:

85

```
git branch -d <branch-name>
git push origin :<branch-name>
```

edited Jun 25 '16 at 12:52

 [Peter Mortensen](#)  
14k 19 87 114

answered Apr 23 '15 at 18:06

 [piyushmandovra](#)  
3,009 2 14 26

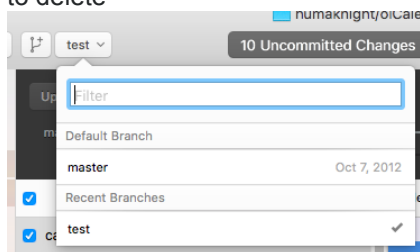


Now you can do it with the [GitHub Desktop](#) app.

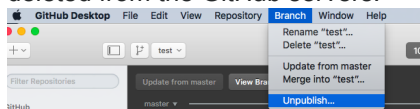
79

After launching the app

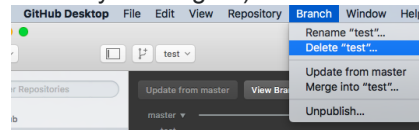
1. Click on the project containing the branch
2. Switch to the branch you would like to delete



3. From the "Branch" menu, select, "Unpublish...", to have the branch deleted from the GitHub servers.



4. From the "Branch" menu, select, 'Delete "*branch\_name*"...', to have the branch deleted off of your local machine (aka the machine you are currently working on)



edited Dec 16 '16 at 3:58

answered Oct 22 '15 at 7:38



Eric

4,827 5 31 55

- 14 I didn't downvote, but my thinking is that it isn't substantively helping. The question is obviously asking for a more commandline type answer without having to use an external program, if people were clicking into here, they likely won't be looking for a github for desktop ways. – [Daemedeor](#) Nov 6 '15 at 11:51

- 9 @Daemedeor , I disagree. In 2010 when the OP asked the question, the UI way of doing it didn't exist and the only option was command line. To indicate that you want a command line only option it should be stated in the question or with the tag, [command-line-interface](#), which in this case is no present. – [Eric](#) Nov 6 '15 at 16:00

- 1 The git command for deleting a remote branch sucks and I tend to forget it (both new and old). Luckily there are GUI tools that have the option. Git Gui, TortoiseGit and GitHub Desktop have it - I wish Git Extensions had this functionality too. Anyway, what I remember is to start Git Gui from within Git Extensions when I need to delete a remote branch. – [vezenkov](#) Mar 25 '16 at 21:59

▲ `git push origin --delete <branch Name>`

78

is easier to remember than



`git push origin :branchName`

edited Dec 13 '18 at 4:44



Alive to Die

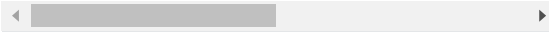
56.6k 8 32 73

answered May 2 '15 at 10:59



Smila

939 6 14

**To delete Locally - (Normal),**

74

`git branch -d my_branch`

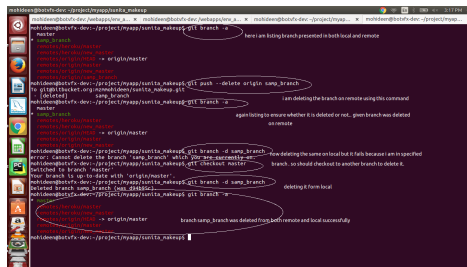
If your branch in rebasing/merging progress and that was not done properly means, you will get an error

Rebase/Merge in progress so in that case, you won't be able to delete your branch.

So either your need to solve rebasing/merging otherwise you can do force Delete by using,

`git branch -D my_branch`**To delete in Remote:**`git push --delete origin my_branch`

can do the same using ,

`git push origin :my_branch # easy to r`**Graphical Representation,**

edited Nov 19 '17 at 10:24



Alex Weitz

1,562 2 19 30

answered Jun 23 '17 at 6:42



Silent Spectator

7,949 4 52 64



64

***This won't work if you have a tag with the same name as the branch on the remote:***



```
$ git push origin :branch-or-tag-name
error: dst refspec branch-or-tag-name ma
error: failed to push some refs to 'git@
```

In that case you need to specify that you want to delete the branch, not the tag:

```
git push origin :refs/heads/branch-or-ta
```

Similarly, to delete the tag instead of the branch you would use:

```
git push origin :refs/tags/branch-or-tag
```

edited Dec 13 '18 at 4:43



[Alive to Die](#)

56.6k 8 32 73

answered Jul 29 '14 at 9:02



[Amiramix](#)

5,973 3 26 42

- 
- 2 Well, my scenario was that I was converting a branch to a tag and it made sense for the tag to have the same name as the branch. By converting I mean merging branch B to A and tagging the last commit in branch B with tag B so that after deleting branch B it can still be easily restored by simply checking out tag B. – [Amiramix](#) Jul 30 '14 at 11:59
- 



50 Many of the other answers will lead to errors/warnings. This approach is relatively fool proof although you may still need `git branch -D branch_to_delete` if it's not fully merged into `some_other_branch`, for example.

```
git checkout some_other_branch
git push origin :branch_to_delete
git branch -d branch_to_delete
```

Remote pruning isn't needed if you deleted the remote branch. It's only used to get the most up to date remotes available on a repository you're tracking. I've observed `git fetch` will add remotes, not remove them. Here's an example of when `git remote prune origin` will actually do something:

User A does the steps above. User B would run the following commands to see the most up to date remote branches

```
git fetch
git remote prune origin
git branch -r
```

edited Dec 13 '18 at 4:41



[Alive to Die](#)

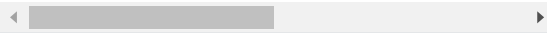
56.6k 8 32 73

answered Nov 27 '13 at 3:04



[Brandon Cook](#)

1,082 10 11



49 I got sick of googling for this answer, so I took a similar approach to [the answer that crizCraig posted](#) earlier.

Added the following to my Bash profile:

```
function gitdelete(){
    git push origin --delete $1
    git branch -D $1
}
```

Then every time I'm done with a branch (merged into `master`, for example) I run the following in my terminal:

```
gitdelete my-branch-name
```

...which then deletes `my-branch-name` from `origin` as well as locally.

edited May 23 '17 at 12:34



[Community](#) ♦

1 1

answered Feb 10 '16 at 19:47



[arthurakay](#)

4,408 7 28 53

7 expanding on this, `--delete "$@"` and `-D "$@"` instead of `$1` will handle it for multiple branches. – [kuni](#) Jun 27 '16 at 13:15

2 I suggest running `git branch -d` (with lowercase 'd') first to ensure changes have been merged, and then push if successful (put `&&` in between commands) – [bryn](#) Jul 19 '16 at 14:17



Before executing

```
git branch --delete <branch>
```

45

make sure you determine first what the EXACT name of the remote branch is by executing:

```
git ls-remote
```

This will tell you what to enter EXACTLY for <branch> value. ( branch is case sensitive!)

edited Jun 25 '16 at 12:56



Peter Mortensen

14k 19 87 114

answered May 20 '16 at 21:42



joey

2,248 23 29



44

```
git push origin :bugfix # Deletes remot
git branch -d bugfix    # Must delete l
```

If you are sure you want to delete it, run

```
git branch -D bugfix
```

Now to clean up deleted remote branches run

```
git remote prune origin
```

answered Apr 21 '16 at 16:49



Vivek

476 1 5 19



43

Mashup of all the other answers.  
Requires Ruby 1.9.3+ , tested **only** on OS X.

Call this file `git-remove` , make it executable, and put it in your path. Then use, for example, `git remove temp` .

```
#!/usr/bin/env ruby
require 'io/console'

if __FILE__ == $0
  branch_name = ARGV[0] if (ARGV[0])
  print "Press Y to force delete loc
  response = STDIN.getch
  if ['Y', 'y', 'yes'].include?(respon
    puts "\nContinuing."
    `git branch -D #{branch_name}`
  end
end
```

```
`git branch -D -r origin/#{branch_
`git push origin --delete #{branch_
else
  puts "\nQuitting."
end
end
```

edited Dec 13 '18 at 4:42



[Alive to Die](#)

56.6k 8 32 73

answered Nov 19 '13 at 21:02



[Dan Rosenstark](#)

42.6k 51 244 383

- 
- 2 one-letter get courtesy of this answer:  
[stackoverflow.com/a/8072675/8047](https://stackoverflow.com/a/8072675/8047) –  
[Dan Rosenstark](#) Nov 19 '13 at 21:04
- 
- 2 sorry, but install Ruby for that kind of  
work ? More logical is implementation on  
bash, which will work out of box. –  
[Reishin](#) May 21 '15 at 19:37
- 
- 2 @Yar this link is out of the context and  
have a more broader scope. I tell only  
about git and as topic is not originated  
only for OSX, that choose is strange for  
other systems (e.g. \*UNIX, Windows) –  
[Reishin](#) May 21 '15 at 20:33
- 



1

2

[next](#)

**protected** by [Community](#) ♦ Jan 20 '13  
at 21:48

Thank you for your interest in this question.  
Because it has attracted low-quality or  
spam answers that had to be removed,  
posting an answer now requires 10  
[reputation](#) on this site (the [association  
bonus does not count](#)).

Would you like to answer one of these  
[unanswered questions](#) instead?