

Less than 10 days to RSVP! Virtually join us at our inaugural conference, everyone is welcome.



Delete last commit in bitbucket

Asked 9 years, 7 months ago Modified 1 year, 2 months ago Viewed 334k times



123



I made a mistake and I don't know how to delete my latest push in the repository. I pull the latest updates of the app but it has conflicts and I push it to repository.

How to delete my last commit? Or how to fix it?



repository

bitbucket

45



Share Edit Follow

asked Feb 12, 2013 at 16:07



catherine

22.1k ● 12 ● 59 ● 82

[Report this ad](#)

10 Answers

Sorted by:

Highest score (default)



114



In the first place, if you are working with other people on the same code repository, you **should not** delete a commit since when you force the update on the repository it will leave the local repositories of your coworkers in an illegal state (e.g. if they made commits after the one you deleted, those commits will be invalid since they were based on a now non-existent commit).



Said that, what you can do is **revert** the commit. This procedure is done differently (different commands) depending on the CVS you're using:



On [git](#):

```
git revert <commit>
```

On [mercurial](#):

```
hg backout <REV>
```

EDIT: The revert operation creates a new commit that does the opposite than the reverted commit (e.g. if the original commit added a line, the revert commit deletes that line), effectively removing the changes of the undesired commit without rewriting the repository history.

Share Edit Follow

edited Feb 22, 2018 at 10:03



Vadim Kotov

7,836 ● 8 ● 46 ● 61

answered Feb 13, 2013 at 16:39



asermax

3,003 ● 2 ● 22 ● 28

Thanks for the information. And I know the consequence if I delete the commit, I just don't know how to undo my action because it's my first time got a mistake. I need to undo my commit because the system is already live. We are only two who are working with that repository. – [catherine](#) Feb 13, 2013 at 16:47

`git backout <REV>` ? Or: `hg backout <REV>` ? – [Armando Pérez Marqués](#) Feb 24, 2013 at 16:02

But what if the commits added after the one to be reverted had made modifications on the line being deleted? – [rahs](#) Jun 1, 2018 at 9:21

@RahulSaha since what you're doing is basically trying to merge a commit into the branch, I'm pretty sure it'll give you a merge conflict you're going have to solve. – [asermax](#) Jun 1, 2018 at 12:52 ✎



103



If you are *not* working with others (or are happy to cause them significant annoyance), then it is possible to remove commits from bitbucket branches.

If you're trying to change a non-master branch:



```
git reset HEAD^          # remove the last commit from the branch history
git push origin :branch_name # delete the branch from bitbucket
git push origin branch_name # push the branch back up again, without the last commit
```

if you're trying to change the master branch

In git generally, the master branch is not special - it's just a convention. However, bitbucket and github and similar sites usually require there to be a main branch (presumably because it's easier than writing more code to handle the event that a repository has no branches - not sure). So you need to create a new branch, and make that the main branch:

```
# on master:
git checkout -b master temp
```

```
git checkout -b master_temp
git reset HEAD^      # undo the bad commit on master_temp
git push origin master_temp # push the new master to Bitbucket
```

On Bitbucket, go to the repository settings, and change the "Main branch" to `master_temp` (on Github, change the "Default branch").

```
git push origin :master      # delete the original master branch from Bitbucket
git checkout master
git reset master_temp        # reset master to master_temp (removing the bad commit)
git push origin master       # re-upload master to bitbucket
```

Now go to Bitbucket, and you should see the history that you want. You can now go to the settings page and change the Main branch back to `master`.

This process will also work with any other history changes (e.g. `git filter-branch`). You just have to make sure to reset to appropriate commits, before the new history split off from the old.

edit: apparently you don't need to go to all this hassle on github, as you can [force-push a reset branch](#).

Dealing with annoyed collaborators

Next time anyone tries to pull from your repository, (if they've already pulled the bad commit), the pull will fail. They will manually have to reset to a commit before the changed history, and then pull again.

```
git reset HEAD^
git pull
```

If they have pulled the bad commit, *and committed on top of it*, then they will have to reset, and then `git cherry-pick` the good commits that they want to create, effectively re-creating the whole branch without the bad commit.

If they never pulled the bad commit, then this whole process won't affect them, and they can pull as normal.

Share Edit Follow

edited May 23, 2017 at 12:03



Community Bot
1 ● 1

answered Sep 1, 2014 at 4:18



naught101
17.7k ● 19 ● 86 ● 134

- 1 For those more comfortable with using SourceTree: 1) To change main branch in the web UI for BitBucket, look at the Repo Settings. 2) To delete a branch in the web UI, look on the Branches page. 3) Reset local repo as per above. 4) Push your local changes up to recreate those branches. – [Richard Le Mesurier](#) May 10, 2015 at 8:10
- 2 As someone who accidentally pushed a secret, I very much appreciated the thoroughness of this answer. Thanks! – [Torque](#) Dec 17 2016 at 12:51

you can reset to `HEAD^` then force push it.

59

```
git reset HEAD^
git push -u origin master --force
```



It will delete your last commit and will reflect on bitbucket as commit deleted but will still remain on their server.

Share Edit Follow

answered Aug 9, 2015 at 18:10



Shub

2,562 • 17 • 25

Github will allow this, but as of this writing, Bitbucket does not allow force pushes. – [JamesQMurphy](#) Sep 2, 2015 at 18:12

- 4 actually bitbucket does support force pushes just the diff is instead of completely removing the un-tracked commits it marks them as deleted . – [Shub](#) Sep 8, 2015 at 15:36 ✎

The recommended way from bitbucket is using revert. But i think in my case this is more appropriate. Since i'm the only one using it, and it feel more clean than using revert. Thanks. – [nafsaka](#) Oct 12, 2015 at 5:19

- 1 Yes, this works on Bitbucket still. You just saved me big time! I did a commit/push from my Eclipse and got mixed up to other project I had. In part due to a bad practice from my part by using both command line and built-in eclipse for working on git. – [DarkCygnus](#) Sep 21, 2017 at 23:46 ✎

brilliant, or easier : reset current branch to this commit , then using that force push to server fix my issue – [Anthony Kal](#) May 3, 2018 at 5:30

Here is a simple approach in up to 4 steps:

16

0 - Advise the team you are going to fix the repository

Connect with the team and let them know of the upcoming changes.



1 - Remove the last commit

Assuming your target branch is `master` :

```
$ git checkout master          # move to the target branch
$ git reset --hard HEAD^       # remove the last commit
$ git push -f                  # push to fix the remote
```

At this point you are done if you are working alone.

2 - Fix your teammate's local repositories

On your teammate's:

```
$ git checkout master      # move to the target branch
$ git fetch                # update the local references but do not merge
$ git reset --hard origin/master # match the newly fetched remote state
```

If your teammate had no new commits, you are done at this point and you should be in sync.

3 - Bringing back lost commits

Let's say a teammate had a new and unpublished commit that were lost in this process.

```
$ git reflog                # find the new commit hash
$ git cherry-pick <commit_hash>
```

Do this for as many commits as necessary.

I have successfully used this approach many times. It requires a team effort to make sure everything is synchronized.

Share Edit Follow

answered Sep 25, 2019 at 5:24



[rbento](#)

7,704 ● 3 ● 54 ● 57

1 Working alone and first solution work as expected ! Thks!! – [Fryla- Cristian Marucci](#) Feb 8 at 14:29



This commands worked Well for me!!

4

In my case, my teammate has done commit with wrong code. So I have to revert that commit in any case, So I tried to



run this command to REVERT the Commit

```
git revert [LastCommitID]
```

After this, changes were not reflected in repository, So for that

I executed this command

```
git push --force origin master
```

and this command reflected changes by removing that Last commit

and this command reflected changes by removing that last commit

Share Edit Follow

answered Jul 8, 2021 at 6:45

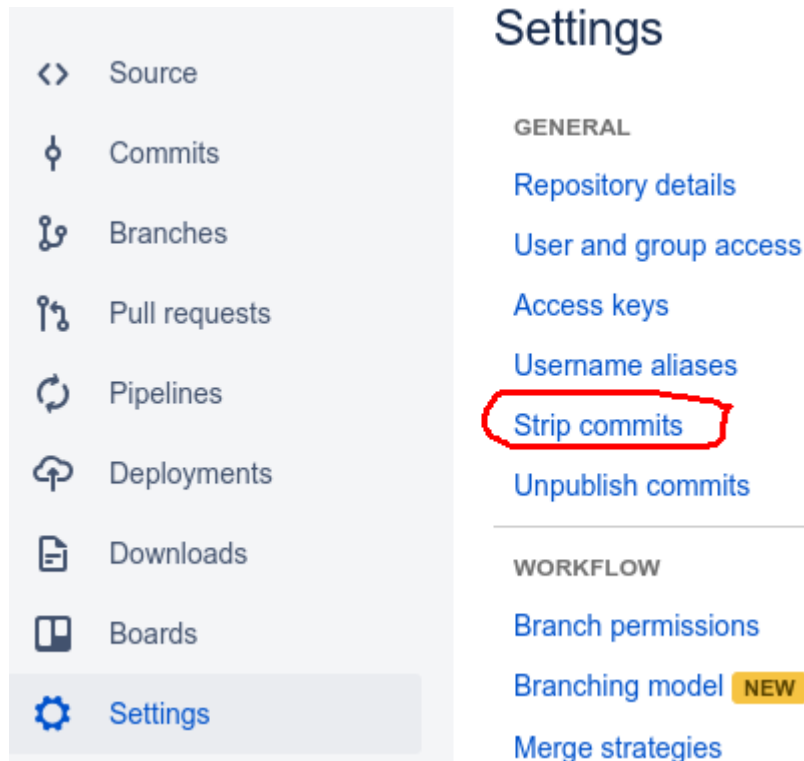


Nachiket Gohil

732 ● 8 ● 14

As others have said, usually you want to use `hg backout` or `git revert`. However, sometimes you really want to get rid of a commit.

First, you'll want to go to your repository's settings. Click on the `Strip commits` link.



Enter the changeset ID for the changeset you want to destroy, and click `Preview strip`. That will let you see what kind of damage you're about to do before you do it. Then just click `Confirm` and your commit is no longer history. Make sure you tell all your collaborators what you've done, so they don't accidentally push the offending commit back.

Share Edit Follow

answered Apr 15, 2019 at 1:21



nmichaels

48.1k ● 12 ● 102 ● 132

Which version is that, and which permissions are required? It looks different on the v6.4.3 I'm looking at, and this item isn't there. I am repository admin. – Raphael Nov 20, 2019 at 8:34

I've had trouble with `git revert` in the past (mainly because I'm not quite certain how it works.) I've had trouble reverting because of merge problems..

My simple solution is this.

Step 1.

```
git clone <your repos URL> .
```

your project in another folder, then:

Step 2.

```
git reset --hard <the commit you wanna go to>
```

then Step 3.

in your latest (and main) project dir (the one that has the problematic last commit) paste the files of step 2

Step 4.

```
git commit -m "Fixing the previous messy commit"
```

Step 5.

Enjoy

Share Edit Follow

answered May 24, 2013 at 23:15



[SudoPlz](#)

18.5k ● 9 ● 80 ● 117

1 Cutting and pasting is really messy. Plus, this wouldn't solve the problem from the question. – [Daniel Lee](#) May 27, 2013 at 9:06

1 `git revert` is simple: it creates a new commit that does the reverse of the changes in a previous commit (or multiple commits). It doesn't delete commits, so it's not relevant for this question. Also, the cloning step here isn't really necessary. – [naught101](#) Sep 1, 2014 at 7:32

1 Git revert is the equal mechanism as above. – [nafsaka](#) Oct 12, 2015 at 5:23

You can write the command also for Bitbucket as mentioned by [Dustin](#):

```
git push -f origin HEAD^:master
```

Note: instead of master you can use any branch. And it deletes just push on Bitbucket.

To remove last commit [locally](#) in git use:

```
git reset --hard HEAD~1
```

Share Edit Follow

edited Jun 24, 2019 at 15:59

answered Jun 24, 2019 at 13:05



invzbl3

4,898 ● 7 ● 30 ● 60

Once changes has been committed it will not be able to delete. because commit's basic nature is not to delete.

Thing you can do (easy and safe method),

Interactive Rebase:

1) `git rebase -i HEAD~2` #will show your recent 2 commits

2) Your commit will list like , Recent will appear at the bottom of the page LILO(last in Last Out)

Terminal

```
mohideen@botvfx-dev: ~/project/easysms
centos@ip-172-16-1-11:/webapps2/env_shotb... x mohideen@botvfx-dev: ~/project/easysms
GNU nano 2.2.6 File: ...t/easysms/.git/rebase-merge/git-rebase-todo
pick 36f58b4 login fixed
pick e5818ba sms rechanged
# Rebase 6bc72c5..e5818ba onto 6bc72c5
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Once changes has been committed it will not be able to delete. because commit's basic nature is not to delete.

Thing you can do (easy and safe method),

Interactive Rebase:

`git rebase -i HEAD~2` #will show your recent 2 commits

Post Your Answer discard

Delete the last commit row entirely

3) save it by `ctrl+X` or `ESC:wq`

now your branch will updated without your last commit..

Share Edit Follow

answered Oct 24, 2017 at 7:50



Mohideen bin
Mohammed



0



By now, cloud bitbucket (I'm not sure which version) allows to revert a commit from the file system as follows (I do not see how to revert from the Bitbucket interface in the Chrome browser).

-backup your entire directory to secure the changes you inadvertently committed

-select checked out directory

-right mouse button: tortoise git menu

-repo-browser (the menu option 'revert' only undoes the uncommitted changes)

-press the HEAD button

-select the uppermost line (the last commit)

-right mouse button: revert change by this commit

-after it undid the changes on the file system, press commit

-this updates GIT with a message 'Revert (your previous message). This reverts commit so-and-so'

-select 'commit and push'.

Share Edit Follow

answered May 7, 2018 at 11:36



smriti

11