difference between git merge origin/master and git pull

Asked 10 years, 10 months ago Modified 10 years, 6 months ago

Viewed 213k times

I'm working on a local branch "BDD-local" and would like to get the changes from other developers. The other developers are using their own branch and once they are happy with the unit tests, they push the changes to remote repo (origin/master).

I was going through several posts here and getting conflicting information. Few people talk about using:

```
git fetch origin
git merge origin/master
```

Some others think, 'git pull' should get the changes.

One of our developers asked to use 'git merge origin/master' without the 'git fetch'

Does anyone know which of these options is better. I tried 'git pull' on my local branch but it didn't seem to work. But if I do 'git pull' on local master it works fine (however I want it to work for the local branch)

git version-control merge

Share

Improve this question

Follow

edited May 30, 2014 at 13:24

user456814

asked Feb 13, 2014 at 14:12

vijay pujar

1,903 ● 4 ● 21 ● 32

1 What were the exact arguments you supplied for `git pull`? – dcastro Feb 13, 2014 at 14:13

3 Answers

Sorted by: Highest score (default)

201

`git pull` is the same as `git fetch` + `git merge`

The command

```
git pull <remote> <branch>
```

is really just the same as

```
git fetch <remote>
git merge <remote>/<branch>
```

So there is no practical difference between

```
git pull origin master
```

and

```
git fetch origin
git merge origin/master
```

Documentation

As stated in the official Linux Kernel `git pull` documentation:

> In its default mode, `git pull` is shorthand for `git fetch` followed by `git merge` `FETCH_HEAD`.
>
> More precisely, *git pull* runs *git fetch* with the given parameters and calls *git merge* to merge the retrieved branch heads into the current branch.

Recommended Reading

- Pro Git § 2.5 Git Basics - Working with Remotes - Fetching and Pulling from Your Remotes.

Share Improve this answer

Follow

answered May 30, 2014 at 13:34

user456814

3 IMHO This is the most specific and succinct answer to the OP's question. I also love that you point to the git documentation which explains it explicitly. Thanks for that. I do appreciate the extended explanation in Mike's answer, but this answer was exactly what I was looking for.
– DryLabRebel Jan 26, 2023 at 22:35

183

fetch, merge, and pull

`git fetch` and `git merge origin/master` will fetch & integrate remote changes. Let me explain a common scenario. origin/master is at C. Someone pushed D. You worked on E & F. Note that you will not see D in your local repository until you run `git fetch`.

```
   origin/master
    v
A-B-C-E-F < master
     \
     (D) < master on remote
```

Now you run `git fetch`. Now you can see D, and origin/master is updated to match the remote repository that it's tracking.

```
A-B-C-E-F < master
     \
      D < origin/master, master on remote
```

Now you run `git merge`, giving you this:

```
A-B-C-E-F
     \   \
      D---G < master
      ^
     origin/master, master on remote
```

So now you've integrated your changes on master (E, F) with the new commits on origin/master (D).

`git pull` is simply a shortcut for the above steps.

git merge without fetching

Running `git merge origin/master` without the `git fetch` is pointless. Without a `git fetch`, your local repository is unaware of any potential changes on the remote repository and origin/master will not have moved. So you're at this state, where D is only on the remote and not present locally:

```
   origin/master
    v
A-B-C-E-F < master
     \
     (D) < master on remote
```

Since your local repository does not have D, a `git merge origin/master` will simply yield:

```
Already up-to-date.
```

Because hey, as far as your local repository is concerned, master already has everything in origin/master.

What's best?

None of the above. :)

```
git fetch
git rebase origin/master master
```

or a shortcut, `git pull -r`, but personally I prefer to see the changes before I rebase.
This will replay your changes on master (E, F) on top of origin/master (D) without a yucky merge commit. It yields:

```
A-B-C-D-E'-F' < master
      ^
   origin/master, master on remote
```

Note how everything is in a single line, you're ready to push, and the history doesn't look like a friendship bracelet.

One warning - never rebase any commits that have already been pushed. Note that E & F became E' & F' after rebasing. The commits are entirely rewritten, with a new SHA and everything. If you rebase commits that are already public, developers will have their history re-written for them when they pull. And that's awful, and everyone will give you evil eyes and shun you.

edited Feb 13, 2014 at 18:28

answered Feb 13, 2014 at 15:21

Mike Monkiewicz

4,689 ● 1 ● 24 ● 19

6 I've been warned many times that git rebase is dangerous. And the git log problem has its workarounds when using git pull and git merge. – Basil Musa Mar 9, 2016 at 11:22

1 I would argue that the rebase method is *not* dangerous when your master branch is something that should never be edited directly and you just need to pull down changes to make a branch of it. – JustGage Sep 27, 2016 at 19:47

what if I have local branch that has been pushed 2-3 times as local branch not merge in master. Now I have few more changes to go but before I wanted to do **rebase** like you suggest. Will it impact anyhting/any loss of data ? Please suggest. – Vishal Patoliya ツ Jul 21, 2017 at 7:20

3 @BasilMusa Rebasing is not dangerous if ONLY DONE LOCALLY. I like the final warning in Mike's answer, though I wish it stood out more. – Josiah Yoder May 6, 2019 at 13:27

1 @Mike Monkiewicz `history doesn't look like a friendship bracelet.` hahaha, lmao XD – CodeTalker Jun 15, 2020 at 15:59

29

A `git pull` is going to run a `git fetch` and then a `git merge`. If you want to bring your local repository up to speed with a remote repository that is what you would run.

A `git fetch` is going to import commits from a remote repo without merging them, which gives you the opportunity to review them before integrating.

Share Improve this answer

Follow

answered Feb 13, 2014 at 14:26

Peter Foti

5,654 ● 6 ● 35 ● 48