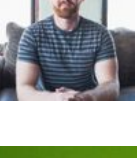


APRIL 26, 2021 / #GIT

Git Push to Remote Branch – How to Push a Local Branch to Origin



John Mosesman



The basic command for pushing a local branch to a remote repository is `git push`.

This command has a variety of options and parameters you can pass to it, and in this article you'll learn the ones that you will use the most often.

How to push a local Git branch to Origin

If you run the simple command `git push`, Git will by default choose two more parameters for you: the **remote repository** to push to and the **branch** to push.

The general form of the command is this:

```
$ git push <remote> <branch>
```

By default, Git chooses `origin` for the remote and your *current branch* as the branch to push.

If your current branch is `main`, the command `git push` will supply the two default parameters—effectively running `git push origin main`.

In the example below, the `origin` remote is a GitHub repository, and the current branch is `main`:

```
(main)$ git remote -v
origin  git@github.com:johnmosesman/burner-repo.git (fetch)
origin  git@github.com:johnmosesman/burner-repo.git (push)

(main)$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 274 bytes | 274.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:johnmosesman/burner-repo.git
b7f661f..ab77dd6  main -> main
```

From the output you can see that the local `main` branch was pushed to the remote `main` branch:

```
To github.com:johnmosesman/burner-repo.git
b7f661f..ab77dd6  main -> main
```

How to force push a branch in Git

Normally, you will push to a branch and add to its commit history.

But, there are times when you need to forcefully **overwrite** the history of a branch.

There are a couple reasons you may want to do this.

The first reason is to fix a mistake—although it is probably better to just make a new commit [reverting the changes](#).

The second and more common scenario is after an action like a **rebase**—which changes the commit history:

Internally, Git accomplishes [a rebase] by creating new commits and applying them to the specified base. It's very important to understand that even though the branch looks the same, it's composed of entirely new commits.

A rebase creates *entirely new commits*.

This means that if you try to push a branch that has been rebased locally—but not on the remote—the remote repository will recognize that the commit history has changed, and it will prevent you from pushing until you settle up the differences:

```
(my-feature)$ git push
To github.com:johnmosesman/burner-repo.git
! [rejected]        my-feature -> my-feature (non-fast-forward)
error: failed to push some refs to 'git@github.com:johnmosesman/burner-repo.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

You could do a `git pull` here to merge the differences, but if you *really* want to overwrite the remote repository you can add the `--force` flag to your push:

```
(my-feature)$ git push --force origin my-feature
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 184 bytes | 184.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To github.com:johnmosesman/burner-repo.git
+ edb64e2...52f54da my-feature -> my-feature (forced update)
```

(Note: you can use `-f` as a shorthand instead of `--force`.)

A force push is a destructive action—only use it when you're certain it's what you want to do.

Force push with lease

Sometimes you may want to force push—but only if no one else has contributed to the branch.

If someone else contributes to your branch and pushes up their changes to the remote—and you force push over it—you will overwrite their changes.

To prevent this scenario, you can use the `--force-with-lease` option.

Again [from the documentation](#):

`--force-with-lease` alone, without specifying the details, will protect all remote refs that are going to be updated by requiring their current value to be the same as the remote-tracking branch we have for them.

Basically, you're telling Git to force update this branch *only if* it looks the same as when you last saw it.

If you're collaborating with others on your branch, it would be good to either avoid using `--force` or at least use `--force-with-lease` to prevent losing changes other collaborators have made.

How to push to a branch of a different name on Git

You will usually push your local branch to a remote branch of the same name—but not always.

To push to a branch of a different name, you just need to specify *the branch you want to push* and the name of the branch you want to *push* to separated by a colon (`:`).

For example, if you want to push a branch called `some-branch` to `my-feature`:

```
(some-branch)$ git push origin some-branch:my-feature
Total 0 (delta 0), reused 0 (delta 0)
To github.com:johnmosesman/burner-repo.git
+ 728f8df...8bf84ea some-branch -> my-feature
```

How to push all local branches to the remote

You won't need to push all branches from your local very often, but if you do you can add the `--all` flag:

```
(main)$ git branch
* main
  my-feature

(main)$ git push --all
--> github.com:johnmosesman/burner-repo.git
b7f661f..6e36148  main -> main
* [new branch]    my-feature -> my-feature
```

Conclusion

The `git push` command is one you'll be using often, and there are tons of options that can be used with it. I encourage you to [read the documentation](#) for helpful options and shortcuts.

If you liked this tutorial, I also talk about topics like this [on Twitter](#), and write about them [on my site](#).

John Mosesman

A simple web developer who likes helping others learn how to program.

If you read this far, tweet to the author to show them you care.

[Tweet a thanks](#)

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers. [Get started](#)