Push local Git repo to new remote including all branches and tags

Asked 8 years, 5 months ago Active 1 year, 4 months ago Viewed 177k times

512

I have a local Git repo that I would like to push to a new remote repo (brand new repo set up on Beanstalk, if that matters). My local repo has a few branches and tags and I would like to keep all of my history. It looks like I basically just need to do a git push, but that only uploads the master branch. How do I push everything so I get a full replica of my local repo on the remote?

git

195

asked Jul 28 '11 at 20:33

Cory Imdieke
11.3k 8 31 45

11 Answers

829

To push all your branches, use either (replace REMOTE with the name of the remote, for example "origin"):

```
git push REMOTE '*:*'
git push REMOTE --all
```

To push all your tags:

```
git push REMOTE --tags
```

Finally, I think you can do this all in one command with:

```
git push REMOTE --mirror
```

However, in addition `--mirror`, will also push your remotes, so this might not be exactly what you want.

edited Aug 3 '17 at 20:29 answered Jul 28 '11 at 20:38


Pramod K.
7 4


cmcginty
90.3k 32 135 146

47 `--all` instead of `*:*` seems more friendly – Idan K Jul 28 '11 at 20:42

53 my god..... i tore of the entire internet and i found out the ` --all ` switch is AAAAALLLLLLLLLLLLL i needed! – Rakib Oct 18 '11 at 22:47

19 Just noting that `git push REMOTE --all` returned `No refs in common and none specified;` doing nothing., while `git push REMOTE "*:*` actually pushed all branches to remote. – Im0rtality Aug 20 '13 at 10:03

10 Use --dry-run to inspect what will happen, in case you have "tmp" or "feature" branches locally that you don't really want to update in REMOTE – Jonno Apr 1 '14 at 23:11

48 If the original remote is still available, it is a nice idea to do `git clone --mirror old-remote-url; cd repo.git; git push --mirror new-remote-url .` – Suzanne Dupéron May 9 '14 at 14:41

142

In the case like me that you aquired a repo and are now switching the remote origin to a different repo, a new empty one...

So you have your repo and all the branches inside, but you still need to checkout those branches for the `git push --all` command to actually push those too.

You should do this before you push:

```
for remote in `git branch -r | grep -v master `; do git checkout --track $remote ; done
```

Followed by

```
git push --all
```

answered Oct 9 '12 at 17:51


Daniel
22k 12 104 148

4 This is also really helpful, as I had to checkout all the branches manually. This will be good for next time. – Cory Imdieke Oct 9 '12 at 20:23

7 Strangely, `git push '*:*'` pushed all branches. `git push -all` just pushed the master. I was transporting repo from github to bitbucket. – jerrymouse Sep 14 '13 at 14:46

3 Instead of checking out every single branch you should just do "git branch --track $remote". In huge repos checking out an old branch takes sometime –

Moataz Elmasry Oct 17 '14 at 11:05

2 I had to make a small change for this to work: `--track remotes/$remote` instead of `--track $remote` . Here's the complete command line: `for remote in `git branch -r | grep -v master `; do git checkout --track remotes/$remote ; done` – Adrian T Sep 18 '15 at 19:30

Thanks, it works for me, the above answer don't work as well. – Benyamin Jafari Aug 24 '18 at 8:35

86

Here is another take on the same thing which worked better for the situation I was in. It solves the problem where you have more than one remote, would like to clone all branches in remote `source` to remote `destination` but without having to check them all out beforehand.

(The problem I had with Daniel's solution was that it would refuse to checkout a tracking branch from the `source` remote if I had previously checked it out already, ie, it would not update my local branch before the push)

```
git push destination +refs/remotes/source/*:refs/heads/*
```

> Note: If you are not using direct CLI, you must escape the asterisks:
>
> ```
> git push destination +refs/remotes/source/*:refs/heads/*
> ```
>
> - @mattalxndr

this will push all branches in remote `source` to a head branch in `destination` , possibly doing a non-fast-forward push. You still have to push tags separately.

edited May 23 '17 at 11:55
Community ♦
1 1

answered Apr 16 '13 at 18:54

Pieter Breed
5,059 4 39 57

8 +1 This worked for me, cloning from one `remote` to another. Thanks! – Laurence Apr 24 '13 at 12:37

3 I had to escape the asterisks: `git push destination +refs/remotes/source/\*:refs/heads/\*` – mattalxndr Oct 17 '14 at 15:40

1 For me, this wound up pushing a branch called HEAD, which I don't think is intentional in this scenario. – maxwellb Apr 29 '16 at 17:57

1 This answer was incredibly useful to me. The only mention in the git-push(1) man page of this use of asterisks is in a tiny example for the `--prune` option. – laindir May 17 '16 at 14:17

3 Excellent answer, far different than the usual `--mirror` parameter everybody recommends. Works perfectly for scenarios where you just want to keep

sync'd two remotes for automation or auditing purposes. – Vinicius Xavier May 24 '16 at 3:01

13

The manpage for `git-push` is worth a read. Combined with this website I wrote the following in my `.git/config`:

```
[remote "origin"]
    url = …
    fetch = …
    push = :
    push = refs/tags/*
```

The `push = :` means "push any 'matching' branches (i.e. branches that already exist in the remote repository and have a local counterpart)", while `push = refs/tags/*` means "push all tags".

So now I only have to run `git push` to push all matching branches and all tags.

Yes, this is not quite what the OP wanted (all of the branches to push must already exist on the remote side), but might be helpful for those who find this question while googling for "how do I push branches and tags at the same time".

answered Dec 19 '13 at 19:22

scy
6,245 2 21 30

12

This is the most concise way I have found, provided the destination is empty. Switch to an empty folder and then:

```
# Note the period for cwd >>>>>>>>>>>>>>>>>>>>>>> v
git clone --bare https://your-source-repo/repo.git .
git push --mirror https://your-destination-repo/repo.git
```

Substitute `https://...` for `file:///your/repo` etc. as appropriate.

answered Dec 16 '16 at 2:08

ta.speot.is
24.8k 8 56 90

In my case what worked was.

9

```
git push origin --all
```

answered Oct 12 '16 at 6:54

nomulex
141 2 2

2 Simple and easy. It works! `origin` is alias for remote URL Git repository. – nhuvy Dec 2 '16 at 6:03

7

Mirroring a repository

Create a bare clone of the repository.

```
git clone --bare https://github.com/exampleuser/old-repository.git
```

Mirror-push to the new repository.

```
cd old-repository.git
git push --mirror https://github.com/exampleuser/new-repository.git
```

Remove the temporary local repository you created in step 1.

```
cd ..
rm -rf old-repository.git
```

Mirroring a repository that contains Git Large File Storage objects

Create a bare clone of the repository. Replace the example username with the name of the person or organization who owns the repository, and replace the example repository name with the name of the repository you'd like to duplicate.

```
git clone --bare https://github.com/exampleuser/old-repository.git
```

Navigate to the repository you just cloned.

```
cd old-repository.git
```

Pull in the repository's Git Large File Storage objects.

```
git lfs fetch --all
```

Mirror-push to the new repository.

```
git push --mirror https://github.com/exampleuser/new-repository.git
```

Push the repository's Git Large File Storage objects to your mirror.

```
git lfs push --all https://github.com/exampleuser/new-repository.git
```

Remove the temporary local repository you created in step 1.

```
cd ..
rm -rf old-repository.git
```

Above instruction comes from Github Help: https://help.github.com/articles/duplicating-a-repository/

edited Aug 21 '18 at 15:10

answered Aug 21 '18 at 12:52

Michał Zalewski
1,942 1 18 29

1 Whilst this may theoretically answer the question, it would be preferable to include the essential parts of the answer here, and provide the link for reference. See here for instructions how to write *better* "link-based" answers. Thanks! – GhostCat says Reinstate Monica Aug 21 '18 at 13:02

4

I found above answers still have some unclear things, which will mislead users. First, It's sure that `git push new_origin --all` and `git push new_origin --mirror` can't duplicate all branches of origin, it just duplicate your local existed branches to your new_origin.

Below is two useful methods I have tested:

1,duplicate by clone bare repo. `git clone --bare origin_url` , then enter the folder, and `git push new_origin_url --mirror` .By this way, you can also use `git clone --mirror origin_url` , both `--bare` and `--mirror` will download a bare repo,not including workspace. please refer this

2,If you have a git repo by using `git clone` , which means you have bare repo and git workspace, you can use `git remote add new_origin new_origin_url` , and then `git push new_origin +refs/remotes/origin/\*:refs/heads/\*` ,and then `git push new_origin --tags`

By this way, you will get a extra head branch, which make no sense.

edited Sep 1 '16 at 11:11

answered Sep 1 '16 at 6:32

yanzi1225627
633 7 13

2

Based in @Daniel answer I did:

```
for remote in \`git branch | grep -v master\`
do
    git push -u origin $remote
done
```

edited May 23 '17 at 12:18

Community ♦
1 1

answered Jan 8 '15 at 13:56

Arthur Julião
591 9 22

2 Even better, `| grep -v master` can be replaced with `| sed 's/\*//'` (I'm assuming you excluded `master` to avoid the nasty little `*` that's prepended to the currently selected branch) which allows you to include `master` and avoid any problems when `master` is not your currently selected branch. Also sorry for necroposting, it's just that this answer helped me today and I wanted to share my modification if it can help others in my position... – ToVine Feb 7 '18 at 10:03

2

To push branches and tags (but not remotes):

```
git push origin 'refs/tags/*' 'refs/heads/*'
```

This would be equivalent to combining the `--tags` and `--all` options for `git push` , which git does not seem to allow.

answered Jul 10 '16 at 3:52

CEL
728 5 10

Is there an additional option to push from another remote ? For instance with `+refs/remotes/source/*` – Yves Martin Mar 27 '17 at 14:51

1

I found that none of these seemed to work properly for me. Feel free to flame this to death but for some reason couldn't get the other options to work properly.

Expected result was a repo "cloned" to another remote (ie from Github to another provider):

- All branches are created on new remote
- All branch history are created on new remote
  - (this was missed on every solution I tried)
- All tags are created on new remote
- Source moves over (a given)
- Non-destructive (giving pause to the --mirror option)

The major issue I was seeing was either all remote branches didn't get recreated in the new remote. If a command did, the new remote did not have the branch history (ie doing a `git checkout branch; git log` wouldn't show the expected branch commits).

I noticed `git checkout -b branchname` is NOT the same as `git checkout branchname` (the latter being what I needed). I notice `git checkout --track branchname` didn't appear to pull the branch history.

My Solution (powershell based):

```
Function Git-FetchRemoteBranches {
$originalbranch = (git symbolic-ref HEAD).split("/")[-1]

Foreach ($entry in (git branch -r)) {

If ($entry -like "*->*") {
  $branch = $entry.split("->")[2].split("/")[1]
}
  else {$branch = $entry.split("/")[1]}

Write-Host "--Trying git checkout " -NoNewline
```

```
Write-Host "$branch" -Foreground Yellow

git checkout $branch

Remove-Variable branch -Force

""}

#Switch back to original branch, if needed
If ( ((git symbolic-ref HEAD).split("/")[-1]) -ne $originalbranch) {
"Switching back to original branch"
git checkout $originalbranch
Remove-Variable originalbranch -Force
}
}

git clone http://remoterepo
cd remoterepo
Git-FetchRemoteBranches
git remote add newremote
git push newremote --all
git push newremote --tags #Not sure if neeeded, but added for good measure
```

answered Mar 30 '17 at 18:46

PotatoFarmer

1,336 1 7 16