

Git push existing repo to a new and different remote repo server?

 stackoverflow.com/questions/5181845/git-push-existing-repo-to-a-new-and-different-remote-repo-server

Say I have a repository on git.fedorahosted.org and I want to clone this into my account at github to have my own playground aside from the more "official" repo on fedorahosted. What would be the steps to initially copy that over? Within github there is this nice "fork" button, but I can't use this for obvious reasons.

And how would I track changes in the fedorahosted repo into the github one?

[git github](#)

6 Answers

[up vote](#) 455
[down vote](#)
accepted

1. Create a new repo at github.
2. Clone the repo from fedorahosted to your local machine.

3. `git remote rename origin upstream`

4. `git remote add origin URL_TO_GITHUB_REPO`

5. `git push origin master`

Now you can work with it just like any other github repo. To pull in patches from upstream, simply run `git pull upstream master && git push origin master`.

up vote 65 down
vote

There is a deleted answer on this question that had a useful link:
<https://help.github.com/articles/duplicating-a-repository>

The gist is

0. create the new empty repository (say, on github)
1. make a bare clone of the repository in some temporary location
2. change to the temporary location
3. perform a mirror-push to the new repository
4. change to another location and delete the temporary location

OP's example:

On your local machine

```
$ cd $HOME
$ git clone --bare
https://git.fedorahosted.org/the/path/to/my_repo.git
$ cd my_repo.git
$ git push --mirror https://github.com/my_username/my_repo.git
$ cd ..
$ rm -rf my_repo.git
```

To push your existing repo into different, you need to:

1. Clone the original repo first.

```
git clone  
https://git.fedorahosted.org/cgit/rhq/rhq.git
```

2. Push the cloned sources to your new repository:

```
cd rhq  
git push https://github.com/user/example  
master:master
```

You may change `master:master` into `source:destination` branch.

If you want to push specific commit (branch), then do:

1. On the original repo, create and checkout a new branch:

```
git checkout -b  
new_branch
```

2. Choose and reset to the point which you want to start with:

```
git log # Find the interesting hash  
git reset 4b62bdc9087bf33cc01d0462bf16bbf396369c81 --  
hard
```

```
git cherry-  
pick
```

Alternatively select the commit by `pick` to append into existing HEAD.

3. Then push to your new repo:

```
git push https://github.com/user/example  
new_branch:master
```

If you're rebasing, use `-f` for force push (not recommended). Run `git reflog` to see history of changes.

up
vote 8
down
vote

Do you really want to simply push your local repository (with its local branches, etc.) to the new remote or do you really want to mirror the old remote (with all its branches, tags, etc) on the new remote? If the latter here's a great blog on [How to properly mirror a git repository](#).

I strongly encourage you to read the blog for some very important details, but the short version is this:

In a new directory run these commands:

```
git clone --mirror git@example.com/upstream-  
repository.git  
cd upstream-repository.git  
git push --mirror git@example.com/new-location.git
```

up vote 3 down
vote

Try this <https://developer.atlassian.com/blog/2016/01/totw-copying-a-full-git-repo/>

1. Create a local repository in the temp-dir directory using:

```
git clone temp-dir
```

2. Go into the temp-dir directory.

3. To see a list of the different branches in ORI do:

```
git branch -  
a
```

4. Checkout all the branches that you want to copy from ORI to NEW using:

```
git checkout branch-  
name
```

5. Now fetch all the tags from ORI using:

```
git fetch --  
tags
```

6. Before doing the next step make sure to check your local tags and branches using the following commands:

```
git tag
```

```
git branch -  
a
```

7. Now clear the link to the ORI repository with the following command:

```
git remote rm  
origin
```

8. Now link your local repository to your newly created NEW repository using the following command:

```
git remote add origin <url to NEW  
repo>
```

9. Now push all your branches and tags with these commands:

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

```
git push --tags
```

10. You now have a full copy from your ORI repo.

[up vote](#) 1
[down vote](#)

I have had the same problem.

In my case, since I have the original repository in my local machine, I have made a copy in a new folder without any hidden file (.git, .gitignore).

Finally I have added the .gitignore file to the new created folder.

Then I have created and added the new repository from the local path (in my case using GitHub Desktop).

answered Dec 27 '15 at 13:33



[itrascastro](#)

537413

Your Answer

Sign up or [log in](#)

Sign up using Google

Sign up using Facebook

Sign up using Email and Password

Post as a guest

By posting your answer, you agree to the [privacy policy](#) and [terms of service](#).

Not the answer you're looking for? Browse other questions tagged [git github](#) or [ask your own question](#).