

unable to push to Heroku after importing thousands of records

Asked 8 months ago Active 8 months ago Viewed 336 times

2

I have a problem where I believe my sqlite3 database is too big. I imported around 100,000 records into a database and I was able to "git push" and "git push heroku." Now I probably made a mistake and imported too many records...500,000. I was able to push to git(and now it states around 336MB in bitbucket) and that seems to work but when i push to heroku this is what i get:

```
/workspace/new_feedback$ git push heroku
Counting objects: 26, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (25/25), done.
Writing objects: 100% (26/26), 159.26 MiB | 1.43 MiB/s, done.
Total 26 (delta 20), reused 1 (delta 0)
remote:
remote: !           Size of checkout and restored submodules exceeds 1 GB. Reduce
size and try pushing again.
remote:
To https://git.heroku.com/magnetic-beach-35611.git
! [remote rejected]  master -> master (pre-receive hook declined)
error: failed to push some refs to 'https://git.heroku.com/magnetic-beach-35611.git'
ubuntu@colin339-rails-tutorial-482323864:~/workspace/new_feedback$
```

I have suspicious that I have to split commits possibly. I've run the command `sqlite3 business.db;` and ran the command `VACUUM FULL;`. I've tried to push up various times, I've tried splitting the commits a couple of times in `REBASE`, and I'm not 100% sure if It's even the right way to go or I'm splitting it right(first time). This error always happens at 159.26 MiB | 1.43 MiB/s, but after some of the splits the (25/25) numbers have been increasing(previously (18/18)). Any ideas how I can resolve this push to Heroku?

[ruby](#) [git](#) [heroku](#) [sqlite](#) [ruby-on-rails-5](#)

asked Dec 6 '18 at 1:44



Dan

31 2

Size of checkout and restored submodules exceeds 1 GB. Reduce size and try pushing again seems pretty self explanatory. You're trying to put 2 gallons of water in a 1 gallon jug. – [Shawn](#) Dec 6 '18 at 2:18

Yes I'm aware, but if I knew how to fix it I wouldn't be asking this question would i? – [Dan](#) Dec 6 '18 at 22:49

2 Answers

Heroku limits git repos to 1GB in size for the entire commit history (not the current file size). Your repo likely exceeds 1GB.

2

<https://devcenter.heroku.com/articles/limits#git-repos>

You have at least two options:

- Do an interactive rebase to squash some of your recent commits.
- Remove the sqlite database from your history, then re-add it.
<https://help.github.com/articles/removing-sensitive-data-from-a-repository/>

Both will end up with rewritten history, but should allow you to shrink the size of your git repository.

Even if you vacuum, you still likely have a significant number of changes in your binary sqlite files, so I'm not sure any action you take on the database will make it any better (in fact, it will make it worse by adding additional commits and size)

Heroku doesn't appear to work with shallow clones and requires the full history so you may need to rewrite your history.

Your git repository is over 1GB in size. This is likely due the total size of all the previous commits in your history (heroku requires a full clone to deploy). Splitting and adding new commits is only going to continue to add to size. You need to determine the source of the bloat. It could be the continued addition of binary files or even the addition and then later deletion (via a git commit) of a large binary file.

You may be able to inspect the size of your repo by running `git count-objects -vH` locally and looking at the `size-pack size`.

You might also try using a script to compare the difference between commits and to get blob sizes:

- <https://stackoverflow.com/a/10847242/2892779>
- <https://stackoverflow.com/a/1318854/2892779>

Also some other options here on cleaning a repository that don't involve rewriting history:

- <https://stackoverflow.com/a/2116892/2892779>

If you've already fixed the repo locally you may need to force push to Heroku if it's rejecting your push. Other than that, I don't think there's anything you can do on the Heroku side to fix this: you need to either:

- Reduce the file size of your sqlite files in the current commit (if you just added them and they're just way too big)
- Rewrite your history to reduce the overall size of your repository
- Compress the files using one of the techniques in the post mentioned above

Since Heroku doesn't support lfs and github doesn't support file sizes larger than a certain size without gitlfs (100MB), and bitbucket doesn't seem to have a limit listed. This is most likely a scenario where you stacked a bunch of commits that had hundreds of MB worth of binary file changes on top of each other bringing you over the 1GB limit.

Bitbucket has more info on how you can figure out your actual repository size (not current default branch total file size) also: <https://confluence.atlassian.com/bitbucket/what-kind-of-limits-do-you-have-on-repository-file-size-273877699.html>

edited Dec 6 '18 at 12:33

answered Dec 6 '18 at 2:23



Jay Dorsey

2,595 2 12 19

I tried clearing the Heroku Cache using these commands: `$ heroku repo:purge_cache -a appname`. Do you think if i just delete some of the imported items....maybe by 200k-300k that it would allow me to push to Heroku after? Also I've been trying to split these commits....not squash them, first time really running into an issue like this. — Dan Dec 6 '18 at 3:30

The issue isn't on the heroku size. At this point adding new commits will just be stacking more data into your repository (because it's more to track). Splitting, again, is most likely going to make it worse. I posted a few links to some commands you can run to try and determine the file size on the diffis. I'd run it against any commit that changed that binary. You need to see if you've been storing a 100MB file. I'd go check this link too, because he provides lots of good ideas (probably better than mine to do a history rewrite): stackoverflow.com/a/2116892/2892779 — Jay Dorsey Dec 6 '18 at 12:20



I'm sure there is probably a better way to resolve this issue, but this is how I did it. I ended up pretty much deleting my database....and recreating it, but the issue remained. I found out the development.log file was around 1.8GB, so the issue actually remained even after I performed a reset on the database...once this file was deleted I was able to push to Heroku again. Also I was importing the records locally and planning on pushing it to git and Heroku, but I think that was actually the wrong way to do it. So going forward I will actually import the records directly on the website and make changes in the Heroku console(heroku run rails console), which should prevent me from running into this issue again. I suspect that I probably could of just deleted this development.log file from the get go to resolve the issue of not being able to push to Heroku.

answered Dec 10 '18 at 6:50



Dan

31 2