

What does "Auto packing the repository for optimum performance" mean?

[Ask Question](#)

196



I'm having a problem with my git repo. For the last couple of days whenever I do a push to the server I get this message: "Auto packing the repository for optimum performance", and it does not seem to go away and return the shell.



38

I also tried checking out to a new branch and then doing a rebase on my previous branch and then did `git gc` to remove the unused history objects and then did a push but still this message appears. Please let me know what's going on with my repo.

[git](#)[git-rebase](#)[git-push](#)

edited Oct 3 '13 at 11:45

[Flimm](#)**56.4k**

23

144

164

asked Dec 26 '11 at 8:03

[Furqan Asghar](#)**1,410**

4

14

25

5 Answers

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

PUBLIC

 Stack Overflow

Tags

Users

Jobs

Teams
Q&A for work[Learn More](#)

271



During most operations which can potentially increase the number of loose (unpacked) objects in the repository (including pushes), Git invokes `git gc --auto`. If there are enough loose objects (by default, at least 6700), it will then invoke `git repack -d -l` to pack them. If there are too many separate packs, it will also repack them into one.

A pack is a delta-compressed single file, containing a large number of objects. It's more efficient to store objects in packs, but it takes time to pack (compress) objects, so Git initially creates loose objects, then packs them in batches now and then, via automatic invocation of `git gc --auto`.

If you let Git finish repacking, this won't happen again for a while. It can indeed take a while, especially if you have a lot of large binary objects, but if it's triggering, then it's a sign that it will probably drastically reduce the amount of disk space taken by the repo. If you really don't want it to happen, you can change the config parameter `gc.auto`. If you increase it to something much larger than 6700, it will happen less frequently, but take longer when it does. If you decrease it, it'll still have to do your current repack, but subsequently it will happen more often and finish more quickly. If you set it to 0, it will disable automatic repacking.

See `man git-gc` (under `--auto`) and `man git-config` (under `gc.auto`) for more information.

answered Dec 26 '11 at 18:31

[Cascabel](#)**343k** 56 333 296

13 Indeed, this took about 5 minutes for me, but it did finish up.
Great answer — [Joshua Pinter](#) Mar 2 '12 at 18:24

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

-
- 2 @dpk: That should not happen in normal circumstances - the number of objects in a single push should not be large enough to trigger it (unless your repository is *enormous* and/or you're pushing a ton of commits), so once it successfully completes (you are letting it complete, right?) it should not happen again until you build up to it. If you can't figure it out, ask a separate question. – [Cascabel](#) Jul 18 '12 at 2:46
-
- 5 "If you let Git finish", *and it can...* fatal: Out of memory, malloc failed (tried to allocate 79610689 bytes) error: failed to run repack -- this is what I get for sticking our entire codebase into one git repo. Guess I'm going to kill apps and force repack "manually" – [ruffin](#) Feb 7 '13 at 15:07
-
- 10 I'm getting it every single time I do a git pull. I've done a manual git gc, but it still happens every time I pull. Weird. – [Barry Kelly](#) Dec 15 '14 at 13:50
-



37



While Jefroni is correct that sometimes the auto-packing just needs time to complete, if the auto-packing message persists over multiple days as OP describes, there's a good chance that git's cleanup is missing dangling objects, as described in [this question](#).

To see whether dangling objects are triggering ongoing messages about auto-packing, try running `git fsck`. If you get a long list of dangling commits, you can clean them with

```
git gc --prune=now
```

I usually have to run this on my repo every 2-3 months when the auto-packing message doesn't go away after a single pull.

[edited May 22 '17 at 12:10](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

answered Aug 29 '16 at 19:40

[wbharding](#)

1,430 2 16 17

-
- 4 While not the accepted answer, this was exactly what I needed. I got the message every time I did a `git pull`, over several days, and `fsck` indeed showed a ton of dangling commits. – [Jörn Zaefferer](#) Mar 28 '17 at 8:37
-



To disable for one project:

30

```
cd your_project_dir
git config gc.auto 0
```



To disable globally:

```
git config --global gc.auto 0
```

edited Feb 11 '15 at 8:01

answered Apr 26 '13 at 9:28

[Anders Lindén](#)

3,659 3 34 75

-
- 3 +1. How would you re-enable? – [Adrian Keister](#) Jul 25 '13 at 3:01
-
- 2 I think I found out how: go to the `.git` folder, open up the config file, and delete the text `'auto = 0'`, and save. That seems to re-enable autopacking. – [Adrian Keister](#) Jul 25 '13 at 3:17
-
- 13 `git config --unset gc.auto` – [jtatum](#) Nov 1 '13 at 18:58

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

10

Git is running `git-repack`, which packs many objects(=files, commits and trees) into one pack file. Git does this sometimes, when a heuristic says that there can be space saved (a pack file contains compressed object deltas, while each file in the objects/ directory contains the compressed full file content)

answered Dec 26 '11 at 13:02



Rudi

15.5k 3 46 71

1

Hopefully, that `git gc --auto` step is now (git 2.0.1, June 25th, 2014) more efficient.

See [commit 62aad18](#) by [Nguyễn Thái Ngọc Duy](#) ([pclouds](#))

`gc --auto` : **do not lock refs in the background**

[9f673f9](#) (`gc` : config option for running `--auto` in background - 2014-02-08, Git 2.0.0) puts "`gc --auto`" in background to reduce user's wait time.

Part of the garbage collecting is pack-refs and pruning reflogs. These require locking some refs and may abort other processes trying to lock the same ref.

If `gc --auto` is fired in the middle of a script, `gc`'s holding locks in the background could fail the script, which could never happen before [9f673f9](#).

Keep running `pack-refs` and "`reflog --prune`" in foreground to stop parallel ref updates. The remaining background operations (repack, prune and rerere) should not impact running git processes.

[edited Apr 27 at 20:43](#)

answered Jul 27 '14 at 19:09



VonC

861k 309 2755

3322

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).