Working with large files and repositories

lecoursen Staff Jul '18

A Git repository contains every version of every file, which works really well for small text files like code. However, adding multiple revisions of large files like audio samples, videos, and datasets can increase the clone and fetch times for every user of a repository. To keep GitHub fast for everyone, we do not allow files larger than 100 MB to be tracked in repositories.

We also recommend that you keep your entire repository under 1GB to prevent excessive load on our servers. This limit is usually easy to stay within if large files are kept out of the repository. For more information about both limits, see "**What is my disk quota?** "

Does your project involve files or repositories above these limits? Let's explore your options for working on GitHub!

Large files

If you try to push a file larger than 100 MB to GitHub, you'll see an error message like one of these:

```
remote: error: File peanut/butter/jelly.so is 118.99 MB; this exceeds GitHub's file size limit o
```

```
The push operation includes a file which exceeds GitHub's file size restriction of 100MB. Please
```

First, ask yourself if you really need the large file on GitHub. If not, you can remove the file from your repository's history. For instructions, see "**Removing sensitive data from a repository** ." You can also use a `.gitignore` file to prevent similar files from being added in the future. For more information, see "**Ignoring files** ."

If you do need the large file on GitHub, ask yourself why. If you're trying to distribute a large binary file, such as an installer, you can use Releases instead. Releases are GitHub's way of packaging and providing software to your users. For more information, see "**About Releases** ."

If you need to version the large file, you can use Git Large File Storage (Git LFS). Git LFS is an open source extension to Git that allows you to work with large files, up to 2GB each, the same way as other text files. These files do not count towards the 1GB repository limit. For more information, see "**Versioning large files** ."

If the large file is already commited to your repository, you can use **git-lfs-migrate** . This tool will add the large file to LFS *and* remove the file from the regular Git repository's history. You'll only be able to push to GitHub after both tasks are complete.

Large repositories

After any files over 100MB are removed or moved to Releases or Git LFS, your repository will probably be under our recommended limit of 1GB. However, some repositories, especially those with extensive histories, may be larger.

It's okay if your repository is slightly larger than 1GB, but we don't allow pushes over 2GB. If you try to push a repository larger than 2GB to GitHub, you'll see a message like this:

```
remote: fatal: pack exceeds maximum allowed size
fatal: The remote end hung up unexpectedly
```

If you've already removed any large files, you can use this command to push your repository in chunks smaller than 2GB:

```
git push <remotename> <commit SHA>:<branch>
```

However, some projects, especially those over 5GB, are just too large for GitHub. If your repository causes excessive load on our servers, you'll receive a polite email from GitHub Support requesting that you reduce the size of the repository.

You can use `git sizer` to compute various size metrics for a Git repository, flagging those that might cause problems.

Need help?

Please, **contact the Support Team** if you have any questions about large files and repositories. We'll be happy to help!

🔗 **Unable to push sizeable data to non-LFS repo**

tonycrook Aug '18

Thanks for clarifying the large file limitation. It make since to keep the data storage limit smaller since most developers value GitHub primarily for tracking version control of executable API's. However, I thought the explanation fell short of offering altenative methods to handle large files that GitHub administration has seen more savvy clients using. Do other clients execute their API's on their large data files kept in other cloud services like Google Cloud or AWS? If so, please explan those methods.

lecoursen **Staff** **Aug '18**

@tonycrook , thanks for asking! Most users who contact the Support Team use one of the methods I mentioned: Releases or Git LFS. I haven't spoken to anyone specifically about keeping large files in Google Cloud or AWS, but feel free to **contact Support** if you'd like to ask specific questions about that option!

Terms **Privacy** **Security** **Status** **Docs** **Contact GitHub** **API** **Training**