

Collaboration with Git Large File Storage

In this article

Viewing large files in pull requests

Pushing large files to forks

Further reading

With Git LFS enabled, you'll be able to fetch, modify, and push large files just as you would expect with any file that Git manages. However, a user that doesn't have Git LFS will experience a different workflow.

If collaborators on your repository don't have Git LFS installed, they won't have access to the original large file. If they attempt to clone your repository, they will only fetch the pointer files, and won't have access to any of the actual data.

Tip: To help users without Git LFS enabled, we recommend you set guidelines for repository contributors that describe how to work with large files. For example, you may ask contributors not to modify large files, or to upload changes to a file sharing service like [Dropbox](#) or [Google Drive](#). For more information, see "[Setting guidelines for repository contributors](#)."

Viewing large files in pull requests

GitHub Enterprise Cloud does not render some Git LFS objects in pull requests. Only the pointer file is shown, with contents similar to the following:

```
+version https://git-lfs.github.com/spec/v1
+id sha256:7194bdd797bde471a6e29b4fa9c8c2278b3c4dadfc5cb2c36d7f4531dc6cb8f
+size 17330
```

For more information about pointer files, see "[About Git Large File Storage](#)."

To view changes made to large files, check out the pull request locally to review the diff. For more information, see "[Checking out pull requests locally](#)."

Pushing large files to forks

Pushing large files to forks of a repository count against the parent repository's bandwidth and storage quotas, rather than the quotas of the fork owner.

You can push Git LFS objects to public forks if the repository network already has Git LFS objects or you have write access to the root of the repository network.

Further reading

- ["Duplicating a repository"](#)

Legal