Git LFS Large files in Git

CONCEPTS

Track a pattern before you add the files: git lfs track only affects files added afterwards, and the rule lives in .gitattributes, which you must commit.

Large file in working tree
↓ git add (clean)
Pointer file tiny text in Git
↓ git push (pre-push)
LFS server real contents
↓ checkout (smudge)
Working tree real file again

Git itself stores only a small pointer; the contents are uploaded on push and downloaded on checkout.

A file that shows a few lines starting with 'version https://git-lfs' is an unfetched pointer: run git lfs pull.

SETUP

git lfs install
Set up LFS filters for your user

git lfs install --local
Set up LFS in this repo only

git lfs install --skip-smudge
Clone with pointers, no download

git lfs version
Show the installed version

git lfs env
Show endpoint, paths and config

TRACKING

git lfs track "*<pattern>*"
Track files matching a pattern

git lfs track
List tracked patterns

git lfs untrack "*<pattern>*"
Stop tracking a pattern

git lfs track --filename *<file>*
Track one exact filename

.gitattributes entry

```
*.psd filter=lfs diff=lfs merge=lfs -text
```

INSPECT

git lfs ls-files
List LFS files in this checkout

git lfs ls-files -s
Include object sizes

git lfs ls-files -a
Files from all history

git lfs ls-files *<ref>*
List LFS files at a ref

git lfs status
LFS files changed or not yet pushed

git lfs status --porcelain
Script-friendly status

git lfs migrate info
Size summary by file type

TRANSFER

git lfs pull
Fetch objects and update files

git lfs fetch
Download objects, keep files as is

git lfs fetch --all
Download every referenced object

git lfs fetch --recent
Also fetch recent branches

git lfs pull -I "*<paths>*"
Only fetch matching paths

git lfs checkout
Write fetched content into files

git lfs push origin *<branch>*
Upload objects for a branch

git lfs push --all origin
Upload all local history

git lfs push --dry-run origin
Show what would be uploaded

MIGRATE

git lfs migrate import --include="*<patterns>*"
Convert history to LFS

git lfs migrate import --include="*<patterns>*" --everything
Convert all refs

git lfs migrate import --no-rewrite *<files>*
Convert without rewriting history

MIGRATE (cont.)

git lfs migrate export --include="*<patterns>*"
Convert LFS files back to Git

Migrate import/export rewrite history and change commit hashes. Commit or stash first, check the result, then force-push and tell collaborators to re-clone.

CLEAN UP

git lfs prune
Delete old local LFS objects

git lfs prune --dry-run
Preview what would be deleted

git lfs fsck
Check LFS objects for corruption

LOCKING

git lfs lock *<path>*
Lock a file against other users

git lfs locks
List current locks

git lfs unlock *<path>*
Release your lock

git lfs unlock --force *<path>*
Break someone else's lock

RECIPES

Start tracking a file type

```
git lfs install
git lfs track "*.psd"
git add .gitattributes design.psd
git commit -m "Track PSD files in LFS"
git push
```

Move existing history to LFS

```
git lfs migrate info --everything
git lfs migrate import --include="*.zip" --everything
git lfs ls-files
git push --force-with-lease --all
```

Clone without big files, fetch later

```
GIT_LFS_SKIP_SMUDGE=1 git clone <url>
cd <repo>
git lfs pull -I "assets/**"
```

Diagnose pointer files in the tree

```
git lfs ls-files   # '-' marks pointers
git lfs env
git lfs pull
```