git-lfs tutorial

Alija Sabic, link:mailto:sabic.alija@gmail.com?subject=GitHub: · git-lfs-intro[sabic.alija@gmail.com]

Setup git repository with **git-lfs**

This repository was setup using following commands:

1. Create a directory

Create directory

```
~$ mkdir git-lfs-intro
~$ cd git-lfs-intro
```

2. Initialize git and git-lfs

> **`git lfs install`** `--local --skip-smudge` (explained below) adds configuration locally (folder). However, the local configuration file (`.git/config`) can't be tracked and hence, uploaded to a remote repository.
>
> Use **`git lfs install`** `--skip-smudge` to prevent LFS from downloading or cloning files (globally) unless explicitly specified.
>
> 
>
> > `--skip-smudge` adds following entries to the global git config at `~/.gitconfig` :
> >
> > 
> >
> > `.git/config` filter.lfs.required=true
> > `.git/config` filter.lfs.clean=git-lfs clean — %f
> > `.git/config` filter.lfs.smudge=git-lfs smudge --skip — %f
> > `.git/config` filter.lfs.process=git-lfs filter-process --skip

Initialize git and git-lfs

```
~/git-lfs-intro$ git init
Initialized empty Git repository in ~/git-lfs-intro/.git/

~/git-lfs-intro$ git lfs install --local --skip-smudge        1  2
Updated git hooks.
Git LFS initialized.

~/git-lfs-intro$ git lfs track '*.pdf'                        3
Tracking "*.pdf"
```

1 `--local` sets the "lfs" smudge and clean filters in the local repository's git config, instead of the global git config (`~/.gitconfig`).

2 `--skip-smudge` skips automatic downloading of objects on clone or pull. This requires a manual "git lfs pull" every time a new commit is checked out on your repository.

3 **`git lfs track`** starts tracking given pattern(s) through git LFS.

`--local` in combination with `--skip-smudge` adds following entries to the local git config at `.git/config` :

```
.git/config       filter.lfs.smudge=git-lfs smudge --skip -- %f
.git/config       filter.lfs.process=git-lfs filter-process --skip
.git/config       filter.lfs.required=true
.git/config       filter.lfs.clean=git-lfs clean -- %f
```

3. Create 'git-lfs' man-pages pdfs

Create data to track with LFS

```
~/git-lfs-intro$ apropos. | awk '{print $1}' | grep git-lfs | while read in; do man -t
"$in" | ps2pdf - "$in".pdf; done
~/git-lfs-intro$ ls
git-lfs-checkout.pdf       git-lfs-config.pdf      git-lfs-fetch.pdf           git-
lfs-install.pdf    git-lfs-logs.pdf
git-lfs.pdf                git-lfs-post-merge.pdf  git-lfs-pull.pdf            git-
lfs-status.pdf     git-lfs-unlock.pdf
git-lfs-clean.pdf          git-lfs-env.pdf         git-lfs-filter-process.pdf  git-
lfs-lock.pdf       git-lfs-ls-files.pdf
git-lfs-pointer.pdf        git-lfs-pre-push.pdf    git-lfs-push.pdf            git-
lfs-track.pdf      git-lfs-untrack.pdf
git-lfs-clone.pdf          git-lfs-ext.pdf         git-lfs-fsck.pdf            git-
lfs-locks.pdf      git-lfs-migrate.pdf
git-lfs-post-checkout.pdf  git-lfs-prune.pdf       git-lfs-smudge.pdf          git-
lfs-uninstall.pdf  git-lfs-update.pdf
```

4. Add files and `.gitattributes` and commit

Commit files

```
~/git-lfs-intro$ git add .gitattributes "*.pdf"
~/git-lfs-intro$ git commit -m "Initial commit"
```

Clone repository

To prevent lfs from downloading or cloning LFS tracked files *globally* enter:

Configure skipping LFS tracked files

```
$ git lfs install --skip-smudge
```

This will prevent downloading lfs files when cloning repositories. If you `clone` this repository now, by entering the command below, git will download only a *pointer* to the lfs file (named just as the original file).

To clone this LFS repository enter:

Clone the repository

```
~/temp$ git clone https://github.com/sabicalija/git-lfs-intro.git
(https://github.com/sabicalija/git-lfs-intro.git)
```

Load lfs file

You can download all or single files by entering following command:

- **Single file**

  Pull single LFS tracked file

  ```
  ~/temp/git-lfs-intro$ git lfs pull --include=filename
  ```

  To download the main git-lfs manpage pdf enter

  Pull file "git-lfs.pdf"

  ```
  ~/temp/git-lfs-intro$ git lfs pull --include="git-lfs.pdf"
  Git LFS: (1 of 1 files) 18.97 KB / 18.97 KB
  ```

  

  Stage the newly downloaded file. This gives git the chance, to see that the "new" file is actually already tracked by git-lfs.

  Restore respository state

  ```
  ~/temp/git-lfs-intro$ git add git-lfs.pdf
  ```

  or reset the repository by entering:

  Restore (reset) repostiory state

  ```
  ~/temp/git-lfs-intro$ git reset HEAD .
  ```

- **All files**

  Pull all LFS tracked files

  ```
  ~/temp/git-lfs-intro$ git lfs pull
  Git LFS: (29 of 29 files) 475.39 KB / 475.39 KB
  ```

Convert lfs file to pointer

In case you don’t need the original binary anymore, you can convert it back to a pointer file. To achieve this enter following commands:

Convert LFS tracked binary to pointer

```
~/temp/git-lfs-intro$ git lfs pointer --file=filename
Git LFS pointer for filename

version https://git-lfs.github.com/spec/v1 (https://git-lfs.github.com/spec/v1)
oid sha256:dc26a22ac0dceb6ce27dd870a1a9f6791f45d4fbc75499093779de3d30b7b5ba
size 19425
```

This will output the generated pointer to the `stdout`. *Copy* the content to a new file, *delete* the downloaded LFS file, and rename the newly created pointer file using it's original name. Now, stage the file again, to restore the repository's state.

To convert the main `git-lfs.pdf` manpage pdf to a pointer file enter:

Convert "git-lfs.pdf" to pointer

```
~/temp/git-lfs-intro$ git lfs pointer --file="git-lfs.pdf" > git-lfs.pdf.pt
Git LFS pointer for git-lfs.pdf

~/temp/git-lfs-intro$ rm git-lfs.pdf
~/temp/git-lfs-intro$ mv git-lfs.pdf.pt git-lfs.pdf
~/temp/git-lfs-intro$ git add git-lfs.pdf
```

Delete a lfs file

The files exchanged by a pointer file are removed from the repository, but not from disk.

To remove the original binary (tracked by git LFS) locate the file you wish to remove at `.git/lfs/objects` and delete it there.

Now, if you want to get the binary again, simply enter `git lfs pull --include=filename`.

You can use the `oid` from the pointer file, to print the location of the binary file to `stdout` or delete the file "automatically".

Read the oid from pointer file

```
~/temp/git-lfs-intro$ cat git-lfs.pdf
version https://git-lfs.github.com/spec/v1 (https://git-lfs.github.com/spec/v1)
oid sha256:dc26a22ac0dceb6ce27dd870a1a9f6791f45d4fbc75499093779de3d30b7b5ba
size 19425

~/temp/git-lfs-intro$ cat git-lfs.pdf | grep oid
oid sha256:dc26a22ac0dceb6ce27dd870a1a9f6791f45d4fbc75499093779de3d30b7b5ba

~/temp/git-lfs-intro$ cat git-lfs.pdf | grep oid | cut -d ":" -f 2
dc26a22ac0dceb6ce27dd870a1a9f6791f45d4fbc75499093779de3d30b7b5ba
```

Print binary file location to terminal

```
~/temp/git-lfs-intro$ find . | grep $(cat git-lfs.pdf | grep oid | cut -d ":" -f 2)
./.git/lfs/objects/dc/26/dc26a22ac0dceb6ce27dd870a1a9f6791f45d4fbc75499093779de3d30b7b5ba
```

You could also use following short-cut, but be carefull. The commands need to be entered subsequently.

Print binary file location to terminal (short cut)

```
~/temp/git-lfs-intro$ cat git-lfs.pdf | grep oid | cut -d ":" -f 2
dc26a22ac0dceb6ce27dd870a1a9f6791f45d4fbc75499093779de3d30b7b5ba

~/temp/git-lfs-intro$ find . | grep $(!!)
find . | grep $(cat git-lfs.pdf | grep oid | cut -d ":" -f 2)
./.git/lfs/objects/dc/26/dc26a22ac0dceb6ce27dd870a1a9f6791f45d4fbc75499093779de3d30b7b5ba
```

The location of the binary, or the names of the folders where it is stored matches a pattern of the oid hash value.

Delete binary file automatically

```
~/temp/git-lfs-intro$ rm $(find . | grep $(cat git-lfs.pdf | grep oid | cut -d ":" -f 2))
```

or with a short cut, just like above:

Delete binary file automatically (short cut)

```
~/temp/git-lfs-intro$ cat git-lfs.pdf | grep oid | cut -d ":" -f 2
dc26a22ac0dceb6ce27dd870a1a9f6791f45d4fbc75499093779de3d30b7b5ba

~/temp/git-lfs-intro$ find . | grep $(!!)
find . | grep $(cat git-lfs.pdf | grep oid | cut -d ":" -f 2)
./.git/lfs/objects/dc/26/dc26a22ac0dceb6ce27dd870a1a9f6791f45d4fbc75499093779de3d30b7b5ba

~/temp/git-lfs-intro$ rm $(!!)
rm $(find . | grep $(cat git-lfs.pdf | grep oid | cut -d ":" -f 2))

~/temp/git-lfs-intro$
```

Last updated 2017-09-19 15:03:58 CEST