

Git Large File Storage

1. [LFS란?](#)

- LFS의 정의
- 왜 사용하나요?

2. [LFS 사용하기](#)

- 사용방법
- 주의사항

3. [참고하기 좋은 자료](#)

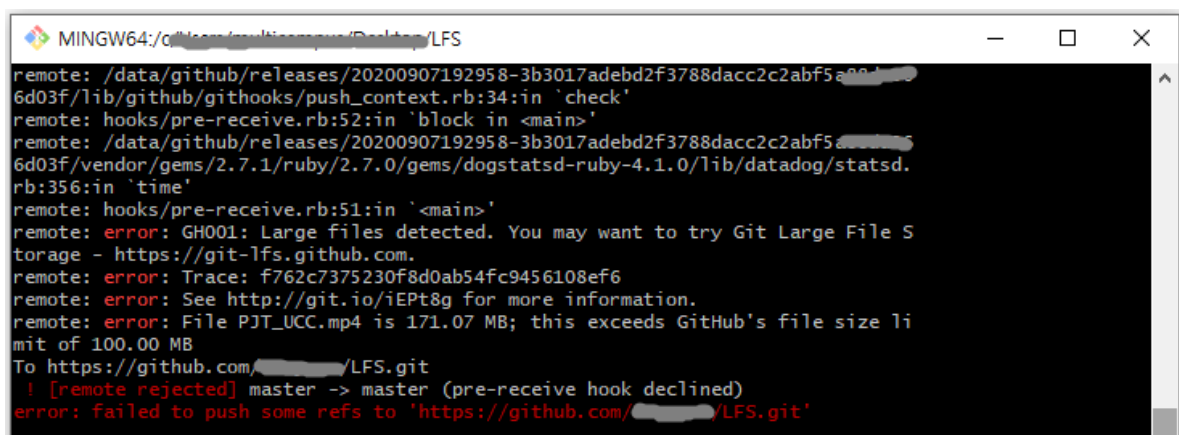
1. LFS란?

LFS의 정의

- Git에서 대용량 파일도 버전관리 할 수 있도록 도와주는 오픈 소스 프로그램
- 텍스트 파일이 아닌 바이너리 파일(모델링, 오디오, 이미지, 데이터 셋 등)을 github과 같은 remote server에 따로 업로드하여 관리합니다.

왜 사용하나요?

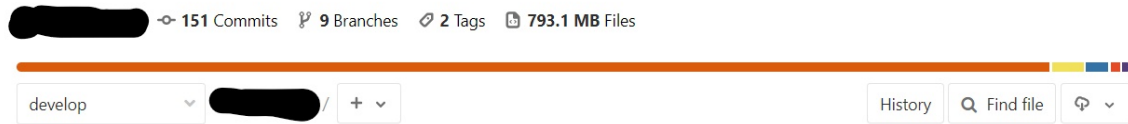
Git은 텍스트 파일과 같은 형태로 등록한 파일이 변화한 내역을 누적하여 형상관리를 합니다. 작업 영역의 버전 관리 내역인 .git 자체는 용량이 매우 작습니다! 그리고 Github에 50MB가 넘는 파일을 올리려 하면 경고가 뜨며, 100MB 이상의 파일은 올릴 수 없다는 에러가 발생하기도 합니다.

A terminal window titled 'MINGW64: C:\Users\multiuser\Desktop\LFS' showing a Git push command. The output shows the file being pushed is 171.07 MB, which exceeds GitHub's 100 MB limit. The terminal displays an error message: 'remote: error: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.' and 'remote: error: File PJT_UCC.mp4 is 171.07 MB; this exceeds GitHub's file size limit of 100.00 MB'. The push is rejected, and the message 'error: failed to push some refs to 'https://github.com/[redacted]/LFS.git'' is shown at the bottom.

```
remote: /data/github/releases/20200907192958-3b3017adebd2f3788dacc2c2abf5a001b5
6d03f/lib/github/githooks/push_context.rb:34:in `check'
remote: hooks/pre-receive.rb:52:in `block in <main>'
remote: /data/github/releases/20200907192958-3b3017adebd2f3788dacc2c2abf5a001b5
6d03f/vendor/gems/2.7.1/ruby/2.7.0/gems/dogstatsd-ruby-4.1.0/lib/datadog/statsd.
rb:356:in `time'
remote: hooks/pre-receive.rb:51:in `<main>'
remote: error: GH001: Large files detected. You may want to try Git Large File S
torage - https://git-lfs.github.com.
remote: error: Trace: f762c7375230f8d0ab54fc9456108ef6
remote: error: See http://git.io/iEPt8g for more information.
remote: error: File PJT_UCC.mp4 is 171.07 MB; this exceeds GitHub's file size li
mit of 100.00 MB
To https://github.com/[redacted]/LFS.git
! [remote rejected] master -> master (pre-receive hook declined)
error: failed to push some refs to 'https://github.com/[redacted]/LFS.git'
```

error 메시지를 보면 친절하게 LFS를 사용하라고도 알려주네요

또한 프로젝트를 할 때 node modules 같은 대용량 파일들은 .gitignore 등록을 통해 git에 올라가지 않도록 관리합니다. 다시 설치하면 되니까요. 그렇다면 다른 팀원들과 함께 공유해야 하는 파일들은 어떻게 관리할 수 있을까요? 그러한 파일의 용량이 큰 경우나, 수정 내역이 많은 경우에는 버전관리를 어떻게 해야 할까요?



793.1MB... 사실 저는 1GB 넘는 Git 도 본 적이 있습니다. 제 프로젝트였죠.

특히 특화 프로젝트의 경우 학습시킨 모델 파일이나, 데이터셋을 팀원들과 공유하고, 관리해야하는 경우가 생깁니다. 한 번 Git 에 등록 되면 변화 내역이 지속적으로 커밋에 쌓이게 되는데, 이 경우 잠깐 방심하면 1GB 가 넘어가게 됩니다.

- 질문. Git 에서 commit 내역이 크고, 용량이 많이 누적되는 경우, 어떤 일이 발생할까요?

```
error: RPC failed; curl transfer closed with outstanding read data remaining
```

clone 이나 pull 을 하고자 할 때에도 에러가 발생합니다. *배포 해야하는데, 클론이 안 돼....* 🙄

이러한 문제를 방지하기 위해 LFS 를 활용합니다!

LFS 등록하고 커밋하면 다른 팀원들도 함께 해당 파일의 버전관리를 할 수 있으며, 위와 같은 대참사를 예방할 수 있습니다.

2. LFS 사용하기

사용방법

1. 공식 사이트에서 다운로드 합니다. <https://git-lfs.github.com/>



Git Large File Storage

[Docs](#) [Downloads](#) [Source](#)

An open source Git extension for versioning large files

Git Large File Storage (LFS) replaces large files such as audio samples, videos, datasets, and graphics with text pointers inside Git, while storing the file contents on a remote server like GitHub.com or GitHub Enterprise.

[Download v2.11.0 \(Windows\)](#)



GitHub.com support now available. [Install the client to get started.](#)

2. 설치해주세요!

```
$ git lfs install
```

3. LFS 를 사용 할 Git repository 에서 어떤 파일 타입을 LFS 를 통해 관리할 것인지 등록해 줍니다. 최초 등록 시 `.gitattributes` 라는 파일이 생성됩니다. 또는 해당 파일을 직접 생성, 작성 또는 추가 하셔도 됩니다.

```
$ git lfs track ".mp4"
```

`.gitattributes` 파일을 열면 아래와 같이 작성되어 있습니다.

```
*.mp4 filter=lfs diff=lfs merge=lfs -text
```

4. `.gitattributes` 를 파일을 git 에 등록해주세요. 스테이징한 이후로는 `.gitattributes` 의 변동 사항을 git 이 알아서 트래킹 하겠죠?

```
$ git add .gitattributes
```

5. 평소에 하듯 add, commit, push 를 진행하면 됩니다.

- 폴더의 상태 입니다. 영상의 크기가 170MB가 되네요.

```
$ ls -la
total 200809
drwxr-xr-x 1 [redacted] 197121 0 9월 8 11:11 ./
drwxr-xr-x 1 [redacted] 197121 0 9월 8 11:37 ../
drwxr-xr-x 1 [redacted] 197121 0 9월 8 11:11 .git/
-rw-r--r-- 1 [redacted] 197121 86 9월 8 10:50 .gitattributes
-rw-r--r-- 1 [redacted] 197121 26215404 8월 13 12:30 LFS_TEST.mp4
-rw-r--r-- 1 [redacted] 197121 179383550 9월 8 09:48 PJT_UCC.mp4
```

- add, commit, push를 하자 LFS 에 파일을 올리고 있다고 알려줍니다.

```
multicampus@DESKTOP-[redacted] MINGW64 ~/Desktop/LFS (master)
$ git add PJT_UCC.mp4

multicampus@DESKTOP-[redacted] MINGW64 ~/Desktop/LFS (master)
$ git commit -m 'Add PJT UCC file'
[master 327dbb6] Add PJT UCC file
2 files changed, 4 insertions(+)
create mode 100644 .gitattributes
create mode 100644 PJT_UCC.mp4

multicampus@DESKTOP-[redacted] MINGW64 ~/Desktop/LFS (master)
$ git push
Uploading LFS objects: 0% (0/1), 75 MB | 4.2 MB/s
```

- 잘 올라가네요!!b

```

multicampus@DESKTOP-... MINGW64 ~/Desktop/LFS (master)
$ git push
Uploading LFS objects: 100% (1/1), 179 MB | 5.8 MB/s, done.
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 493 bytes | 493.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/.../LFS.git
 d39a12c..327d... master -> master

```

Code Issues Pull requests Actions Projects Security Insights

master 1 branch 0 tags Go to file Add file Code

Add LFS_TEST.mp4		97d0883 43 minutes ago 4 commits
.gitattributes	Mod .gitattributes	1 hour ago
LFS_TEST.mp4	Add LFS_TEST.mp4	43 minutes ago
PJT_UCC.mp4	Add PJT_UCC file	1 hour ago

주의사항

저는 이 문서를 작성하기 위해 PJT_UCC.mp4 라는 파일을 LFS 에 등록하기 이전에 git 에 add 한 상태였습니다.

그런데, LFS 를 통해 파일을 관리하려면 **Git 이 해당 파일을 tracking 하기 이전에 즉, staging 하기 이전에, 쉬운 말로 git add 하기 이전에!!!** LFS 에 등록해 주어야 합니다. 이는 Git 의 작동 원리와의도 이어 지는데요.. Git 설명 문서가 아니니 자세한건 생략할게요. 이미 이전에 add 한 내역을 취소하거나, commit 도 되돌릴 수 있으니 관련해서는 자료를 찾아봅시다. 저는 commit 을 취소하고 위의 작업을 진행했었습니다.

```

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  PJT_UCC.mp4

no changes added to commit (use "git add" and/or "git commit -a")

```

3. 참고하기 좋은 자료

- [Git LFS 공식 문서](#)
- [Git LFS \(Large File Storage\) 사용해보기](#)
- [Git Lfs를 사용하여 대용량 파일 관리하기](#)
- Git 에 대한 추가자료: [Git 의 기초 - 수정하고 저장소에 저장하기](#)