

▶ Using large file upload (LFS)

Available in Classic and VPC

The size of a file that can be uploaded to the Git repository is limited to a maximum of 500 MB, but if you utilize the Git LFS (Large file system), you can manage versions of large files (up to 5 GB per file) in the Git repository. Git LFS performs in a way that only metadata of the large file is committed to the repository, and the actual file content is saved in a separate remote repository.

Requirements for using Git LFS

Install the Git client and Git LFS on a local PC to use Git LFS.

- Download the Git client installation file from the [Official Git website](https://git-scm.com/downloads/guis) (<https://git-scm.com/downloads/guis>) and install to check the detailed instructions.
- Download the Git LFS installation file from the [Official Git LFS website](https://git-lfs.github.com/) (<https://git-lfs.github.com/>) and install to check the detailed instructions.
- Git LFS supports HTTPS only, not SSH connection.

Note

- If you share a repository with Git LFS applied for common projects, make sure that all project members install Git LFS on local PCs and recognize Git LFS's operation principle and how to use it.
- Managing large files and locking files using LFS features aren't available on local PCs without Git LFS installed.
- After downloading and installing Git LFS, reset the Git LFS from the path of the repository you're working on as follows.

Bash

Copy

```
$ git lfs install
```

```
Git LFS initialized.
```

Upload large files

The following describes how to upload a file by using Git LFS.

This guide describes how to upload (push) to the remote repository after resetting Git LFS in the local repository and setting tracking for a certain file. Refer to [Official Git LFS website](https://git-lfs.github.com/) (<https://git-lfs.github.com/>) for details about ^

how to use Git LFS.



1. Run the Git client on the local PC, and then go to the repository path to work on.
2. Reset Git LFS.

Bash	Copy
<pre>\$ git lfs install Git LFS initialized.</pre>	

3. Set tracking for the large file to be uploaded, and then add `.gitattributes` file to the commit list.
 - You can use wildcards (*) to specify multiple files.
 - The tracking history for the file is recorded in the `.gitattributes` file.

Bash	Copy
<pre>\$ git lfs track "*.psd" Tracking "*.psd" \$ git add .gitattributes</pre>	

4. Commit the large file to be uploaded, and then upload (push) it to the remote repository.

Bash	Copy
<pre>\$ git add sample.psd \$ git commit -m "Added design file" \$ git push origin master</pre>	

Note

- If errors like timeout (i/o timeout) occur while uploading a large file, you can extend the wait time as follows.

Bash	Copy
<pre>\$ git config lfs.activitytimeout 300</pre>	

- After the file is uploaded, you can check that the large file with tracking set is saved as LFS in the SourceCommit repository's **[Code]** tab.



Lock file



With Git LFS's file lock feature, you can set to modify or upload locked files only if you're using a certain account.

Note

- The file lock feature is only available in Git LFS 2.0 or later versions.
- Set the file lock feature in the user's local PC repository to make the Git client recognize the file lock status correctly.

The following describes how to set the file lock feature.

Bash	Copy
<pre>\$ git config lfs.{Git URL}/info/lfs.locksverify true</pre>	

Note

- You can unlock the file lock feature only in accounts with the file lock feature set.
- When you don't set the lock feature, the file lock feature may not operate properly like upload is performed with the following warning message, even when uploading a locked file to other accounts.

Bash	Copy
<pre>\$ git push origin master Locking support detected on remote "origin". Consider enabling it with: \$ git config lfs.{Git URL}/info/lfs.locksverify true Unable to push locked files: test.txt - psr (refs: master) WARNING: The above files would have halted this push.</pre>	

Example of using the file lock

After enabling the file lock, you can upload or unlock the file only in sub accounts with the file lock feature set.

- Lock settings in "sub account 1"





Bash

Copy

```
$ git lfs lock "test.txt"
Locked test.txt

$git lfs locks
test.txt {sub_account_name} ID:15
```

- Upload file from "sub account 2" (error occurred)

Bash

Copy

```
$ git add test.txt

$ git commit -m "Modify file"

$ git push origin master
Unable to push locked files:
* test.txt - {sub_account_name} (refs: master)
ERROR: Cannot update locked files.
```

Example of removing the file lock settings

- Remove lock settings in "sub account 1" (unlocked successfully)

Bash

Copy

```
$ git lfs unlock test.txt
Unlocked test.txt
```

- Remove lock settings in "sub account 2" (error occurred)

Bash

Copy

```
$ git lfs unlock test.txt
test.txt is locked {Sub Account1}
```



Previous

[Webhook event list](#)

Next

[File Safer malware scan](#)

