What is GitHub

GitHub, **GitHub is an online software development platform**. is an AI-powered developer platform that allows developers to create, store, and manage their code. It uses Git software, providing the **distributed version control** of Git plus access control, bug tracking, software feature requests, task management, continuous integration,

Installation Steps

Other Git for Windows downloads

Standalone Installer

https://git-scm.com/download/win

mkdir foldername

```
cd foldername
```

open right click with git bash

```
git –version
git config --global user.name "Sachin Sirohi"
git config user.name
git config --global user.email abc@gmail.com
git config user.email
git init
ls
ls -a

git config --global --edit
```

change something in the Addition.py file and show modified Changes

```
git add Addition.py
git status
git commit -m “Adding Message”
git log
```

commit all modified file

```
git add .
```

or

```
git add –all
```

or

```
git add -A

git commit -m “Commit All”
git status
```

create GitHub repository

https://github.com/

after creating GitHub profile and creating a new repository

```
git remote -v
git branch -m master
git remote add origin https://github.com/SACHINSIROHI47/Data-Science.git
git remote -v
git push -u origin master
```

create branch

```
git branch sachin
git branch
git checkout sachin
```

second branch

```
git checkout -b college/student
git status
git add studentinfo.py
git commit -m “Student File”
```

To add file another branch

```
git checkout sachin
git merge college/student
git log
```

to merge the sachin branch into master

```
git merge sachin
```

git Ignore we want to secure the file so that no one can read

```
touch .gitignore
```

```
git status
```

after add secret.txt file and .gitignore file into .gitignore into vs code

```
git status
```

after authorizing by id and password of git

```
git push -u origin college/student
```

add collaborator for give access for change

settings->collaborator->manage access->add people

fork to add other people’s repositories into their own repository

below repository of another person, we are locally clone into their own system first change branch

```
git clone https://github.com/SACHINSIROHI47/Data-Science.git
```

if you want to change name

```
git config --global –edit
```

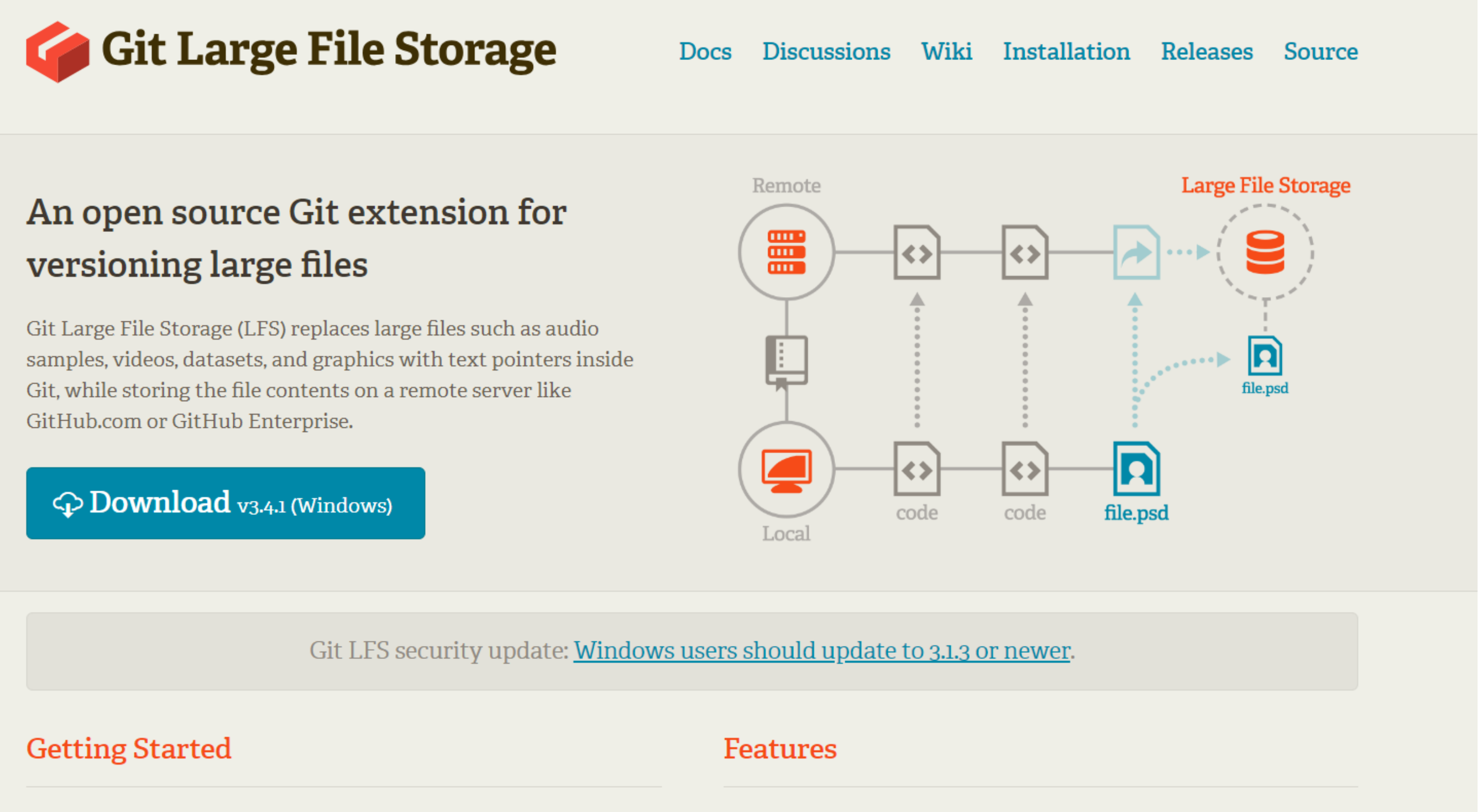
after changes

```
git add addition.py
```

```
git commit -m “file updated by another user”
```

```
git push
```

after accepting the pull request into the GitHub repository



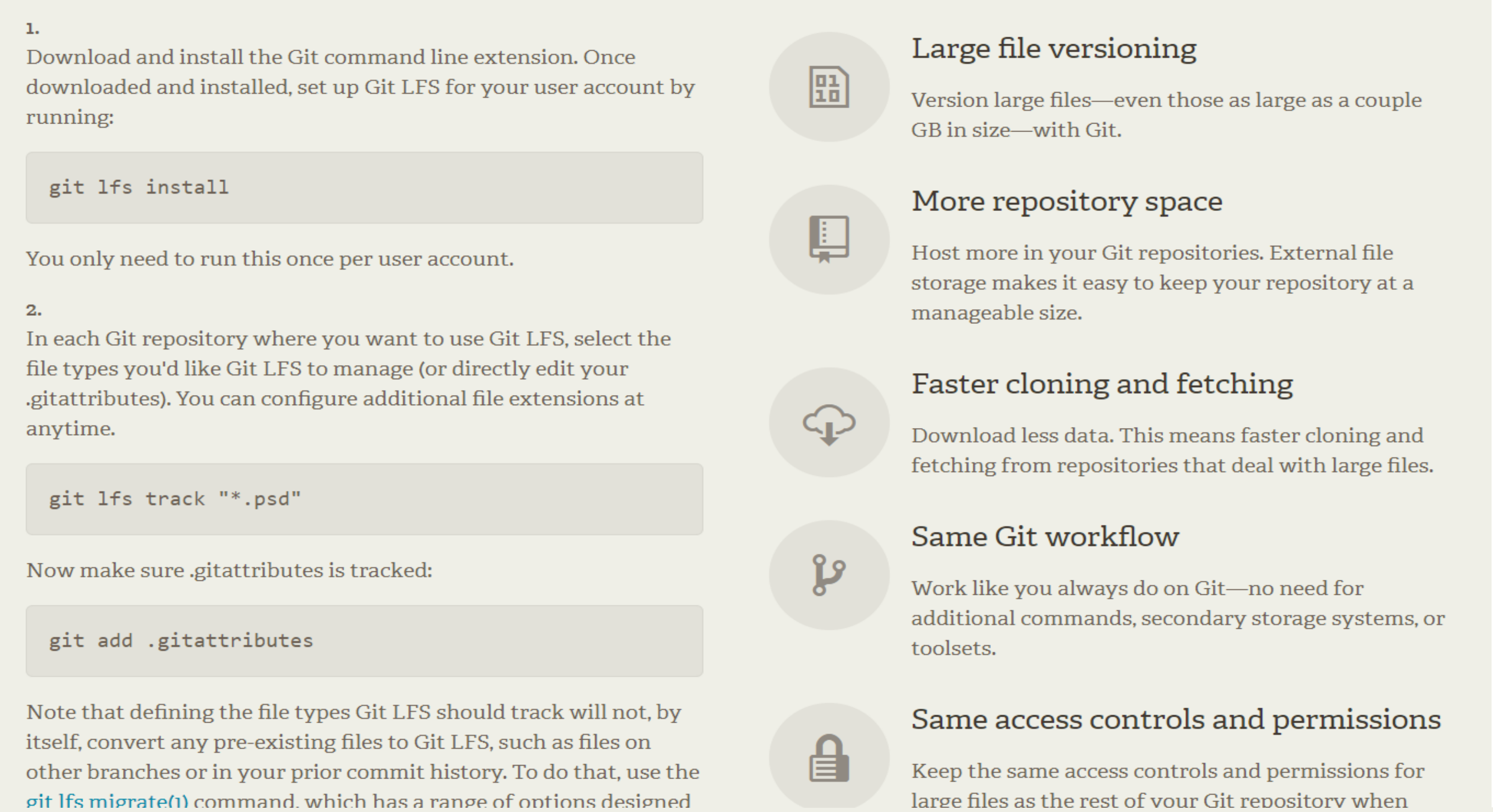
An open source Git extension for versioning large files

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.

Git LFS security update: Windows users should update to 3.1.3 or newer.

Getting Started

Features



1.
Download and install the Git command line extension. Once downloaded and installed, set up Git LFS for your user account by running:

```
git lfs install
```

You only need to run this once per user account.

2.
In each Git repository where you want to use Git LFS, select the file types you'd like Git LFS to manage (or directly edit your .gitattributes). You can configure additional file extensions at anytime.

```
git lfs track "*.psd"
```

Now make sure .gitattributes is tracked:

```
git add .gitattributes
```

Note that defining the file types Git LFS should track will not, by itself, convert any pre-existing files to Git LFS, such as files on other branches or in your prior commit history. To do that, use the git lfs migrate(1) command, which has a range of options designed

Large file versioning

Version large files—even those as large as a couple GB in size—with Git.

More repository space

Host more in your Git repositories. External file storage makes it easy to keep your repository at a manageable size.

Faster cloning and fetching

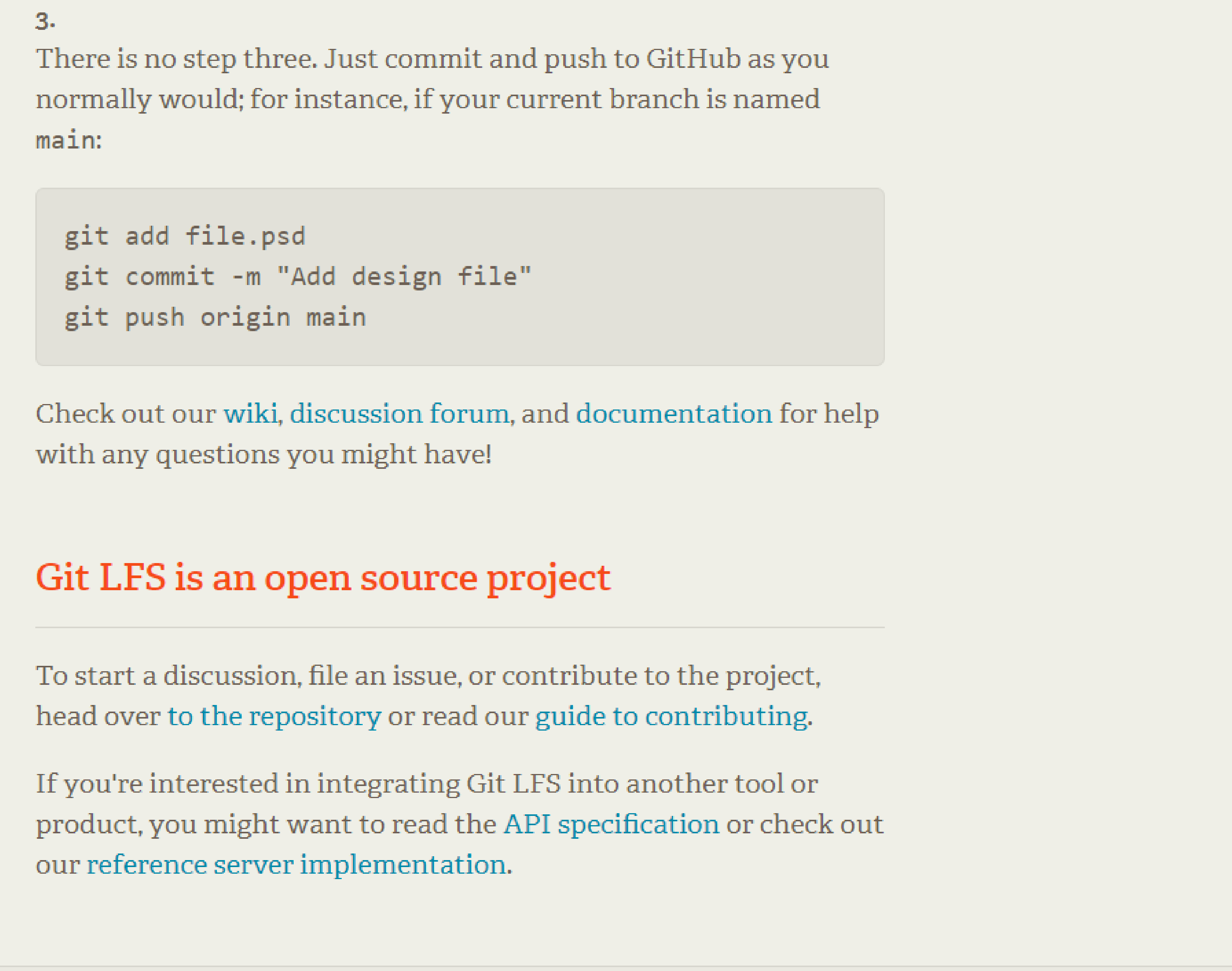
Download less data. This means faster cloning and fetching from repositories that deal with large files.

Same Git workflow

Work like you always do on Git—no need for additional commands, secondary storage systems, or toolsets.

Same access controls and permissions

Keep the same access controls and permissions for large files as the rest of your Git repository when



3.
There is no step three. Just commit and push to GitHub as you normally would; for instance, if your current branch is named `main`:

```
git add file.psd
git commit -m "Add design file"
git push origin main
```

Check out our wiki, discussion forum, and documentation for help with any questions you might have!

Git LFS is an open source project

To start a discussion, file an issue, or contribute to the project, head over to the repository or read our guide to contributing.

If you're interested in integrating Git LFS into another tool or product, you might want to read the API specification or check out our reference server implementation.