Sajjad Goodarzi

Sreedevi Langoju

What is Git ?

To understand GitHub, you must first have an understanding of Git. Git is an open-source version control system that was started by Linus Torvalds—the same person who created Linux. Git is similar to other version control systems

So, Git is a version control system, but what does that mean? When developers create something (an app, for example), they make constant changes to the code, releasing new versions up to and after the first official (non-beta) release

What is Github?

GitHub is a web-based platform that allows developers to collaborate on projects, share code, and track changes to their code. It is primarily used for version control, which helps developers manage and track changes to their code over time. GitHub provides a centralized location for developers to store their code, collaborate with others, and manage their projects effectively. It also offers features like issue tracking, pull requests, and code reviews, making it a popular platform for open-source projects and software development teams.

How can Git & Github help us ?

GitHub allows multiple developers to work on a single project at the same time, reduces the risk of duplicative or conflicting work, and can help decrease production time. With GitHub, developers can build code, track changes, and innovate solutions to problems that might arise during the site development process simultaneously. Non-developers can also use it to create, edit, and update website content

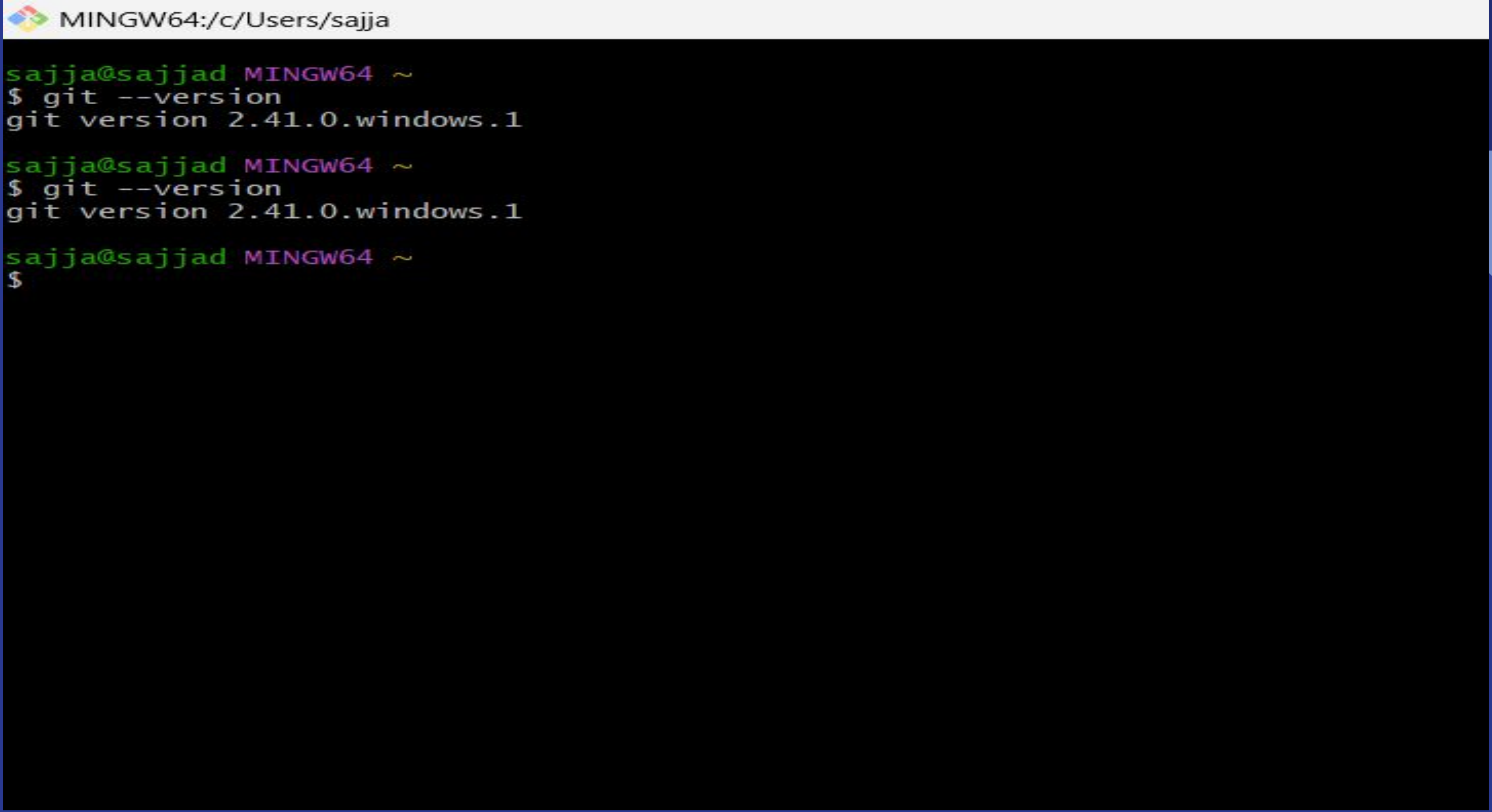
Install git on your computer

MAC OS:

git-osx-installer - Browse Files at SourceForge.net

WINDOWS OS:

Git for Windows



```
sajja@sajjad MINGW64 ~
$ git --version
git version 2.41.0.windows.1

sajja@sajjad MINGW64 ~
$ git --version
git version 2.41.0.windows.1

sajja@sajjad MINGW64 ~
$
```

Git init:

`git init` turns any directory into a Git repository.

Common usages and options for `git init`

- `git init`: Transform the current directory into a Git repository
- `git init <directory>`: Transform a directory in the current path into a Git repository
- `git init --bare`: Create a new bare repository (a repository to be used as a remote repository only, that won't contain active development)

Git Clone:

The `git clone` command is used to create a copy of a specific repository or branch within a repository.

Common usages and options for `git clone`

- `git clone [url]`: Clone (download) a repository that already exists on GitHub, including all of the files, branches, and commits.
- `git clone --mirror`: Clone a repository but without the ability to edit any of the files.
- `git clone --single-branch`: Clone only a single branch

Git Add:

The `git add` command adds new or changed files in your working directory to the Git staging area.

Common usages and options for `git add`

- `git add <path>`: Stage a specific directory or file
- `git add .`: Stage all files (that are not listed in the `.gitignore`) in the entire repository
- `git add -p`: Interactively stage hunks of changes

Git Commit:

`git commit` creates a commit, which is like a snapshot of your repository. These commits are snapshots of your entire repository at specific times. You should make new commits often, based around logical units of change

Common usages and options for Git Commit

- `git commit`: This starts the commit process, but since it doesn't include a `-m` flag for the message, your default text editor will be opened for you to create the commit message. If you haven't configured anything, there's a good chance this will be VI or Vim. (To get out, press esc, then `:w`, and then Enter.
- `git commit -m "descriptive commit message"`: This starts the commit process, and allows you to include the commit message at the same time.

Common usages and options for Git Commit

- `git commit -am "descriptive commit message"`: In addition to including the commit message, this option allows you to skip the staging phase. The addition of `-a` will automatically stage any files that are already being tracked by Git (changes to files that you've committed before).
- `git commit --amend`: Replaces the most recent commit with a new commit.

Git Status:

`git status` shows the current state of your Git working directory and staging area.

Common usages and options for `git status`

- `git status`: Most often used in its default form, this shows a good base of information
- `git status -s`: Give output in short format
- `git status -v`: Shows more "verbose" detail including the textual changes of any uncommitted files

Git Pull:

`git pull` updates your current local working branch, and all of the remote tracking branches

Common usages and options for `git pull`

- `git pull`: Update your local working branch with commits from the remote, *and* update all remote tracking branches.
- `git pull --rebase`: Update your local working branch with commits from the remote, but rewrite history so any local commits occur after all new commits coming from the remote, avoiding a merge commit.
-

Common usages and options for git pull

`git pull --force`: This option allows you to force a fetch of a specific remote tracking branch when using the `<refspec>` option that would otherwise not be fetched due to conflicts. To force Git to overwrite your current branch to match the remote tracking branch,

What is branch ?

- In Git, branches are used to create separate lines of development. They allow you to work on different features or bug fixes without affecting the main codebase
- You can create a new branch using the "git branch" command followed by the branch name.
- For example: `git branch new-feature`

Git Push:

`git push` uploads all local branch commits to the corresponding remote branch.

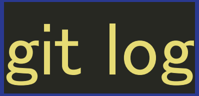
Common usages and options for `git push`

- `git push -f`: Force a push that would otherwise be blocked, usually because it will delete or overwrite existing commits *(Use with caution!)*
- `git push -u origin [branch]` : Useful when pushing a new branch, this creates an upstream tracking branch with a lasting relationship to your local branch
- `git push --all`: Push all branches
- `git push --tags`: Publish tags that aren't yet in the remote repository

Useful Git Commands

Git log command:

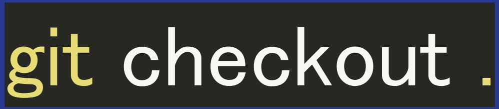
Display the history of commits (all modifications made on the project):



```
git log
```

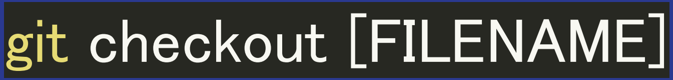
Git checkout command:

Revert all your changes since the last commit:



```
git checkout .
```

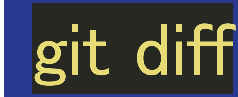
Revert all changes on a specific file since the last commit:



```
git checkout [FILENAME]
```

Git diff command:

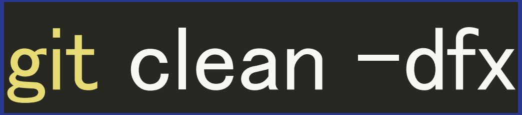
The "git diff" command is used to show the differences between the current state of the code and the previous commit



```
git diff
```

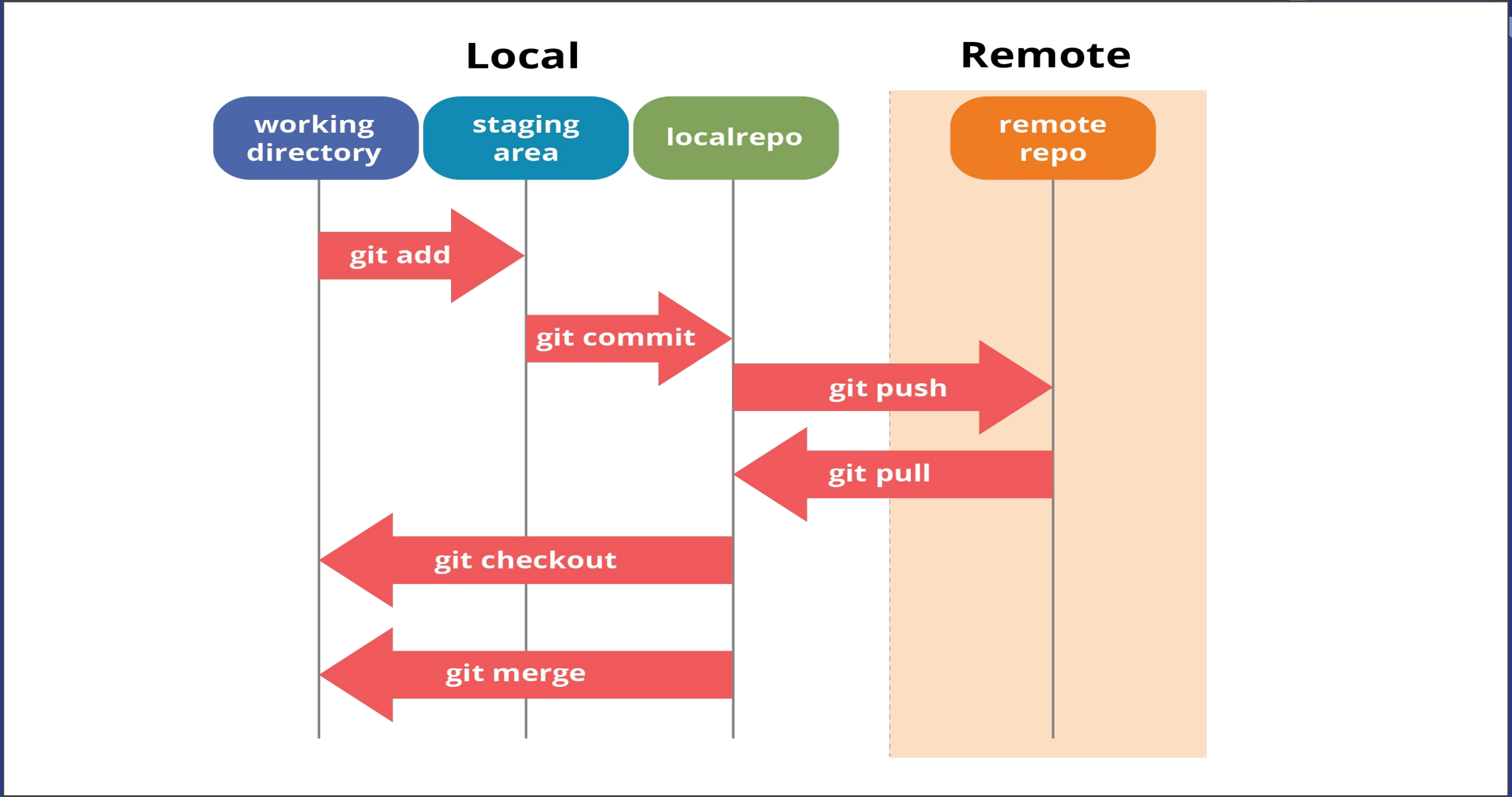
Git clean command:

Remove all unexpected files in your project (not committed):

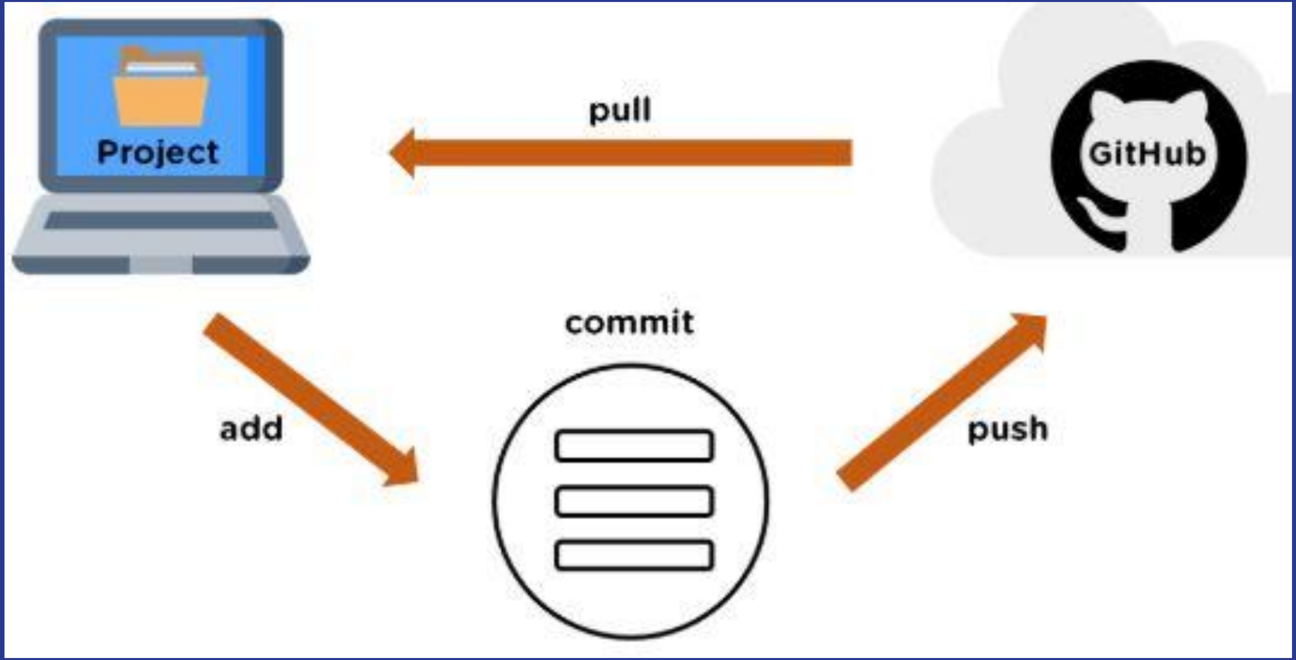


```
git clean -dfx
```

Git Architecture



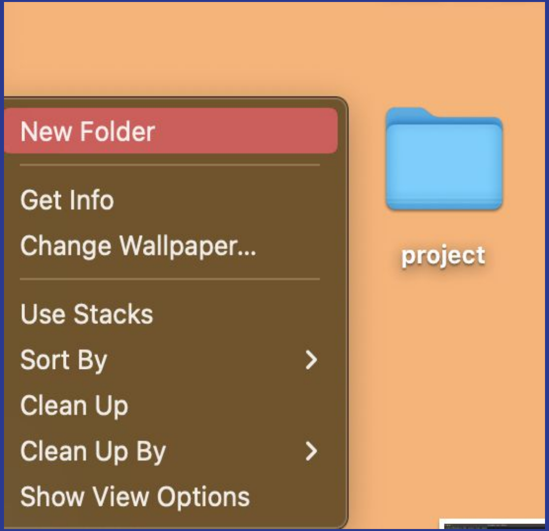
Git Push and Pull commands



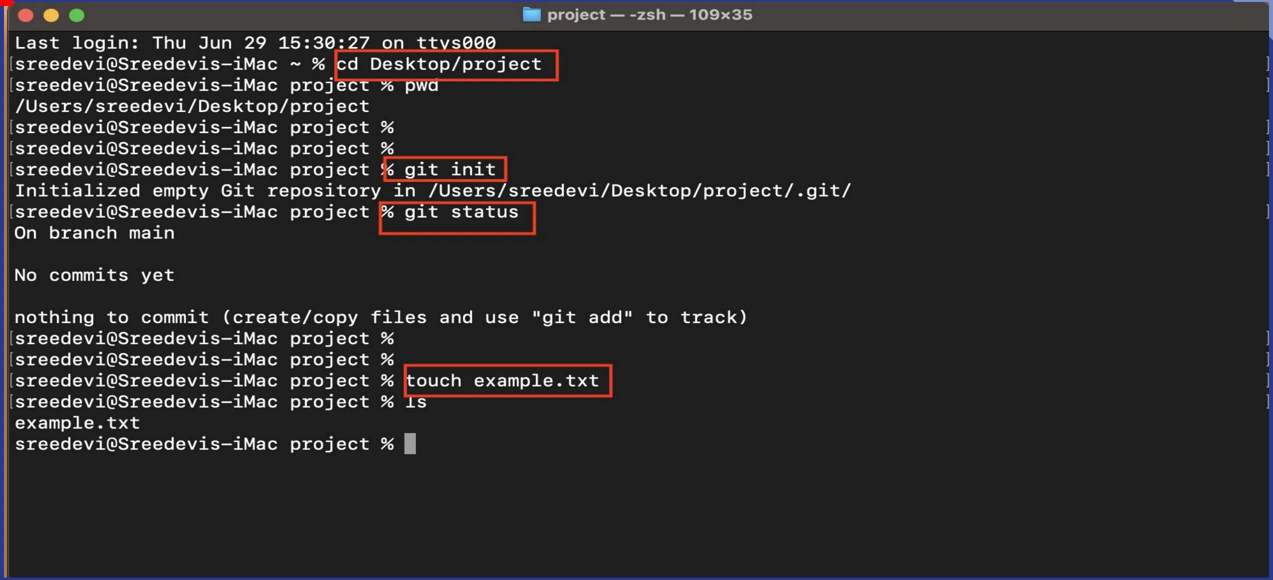
Push git repository to Github using git command line tool

Step1 : Install git & create local files in git repository

1



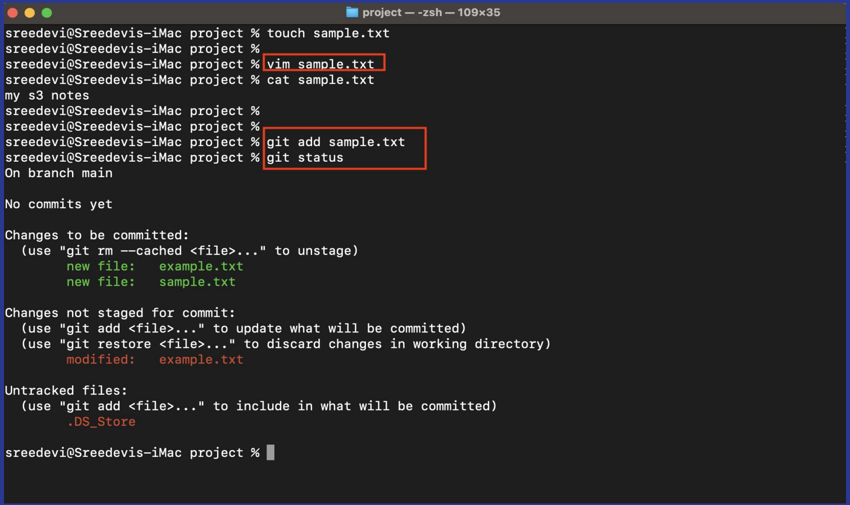
2



```
Last login: Thu Jun 29 15:30:27 on ttys000
[sreedevi@Sreedevis-iMac ~ % cd Desktop/project
[sreedevi@Sreedevis-iMac project % pwd
/Users/sreedevi/Desktop/project
[sreedevi@Sreedevis-iMac project %
[sreedevi@Sreedevis-iMac project %
[sreedevi@Sreedevis-iMac project % git init
Initialized empty Git repository in /Users/sreedevi/Desktop/project/.git/
[sreedevi@Sreedevis-iMac project % git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
[sreedevi@Sreedevis-iMac project %
[sreedevi@Sreedevis-iMac project %
[sreedevi@Sreedevis-iMac project % touch example.txt
[sreedevi@Sreedevis-iMac project % ls
example.txt
sreedevi@Sreedevis-iMac project %
```

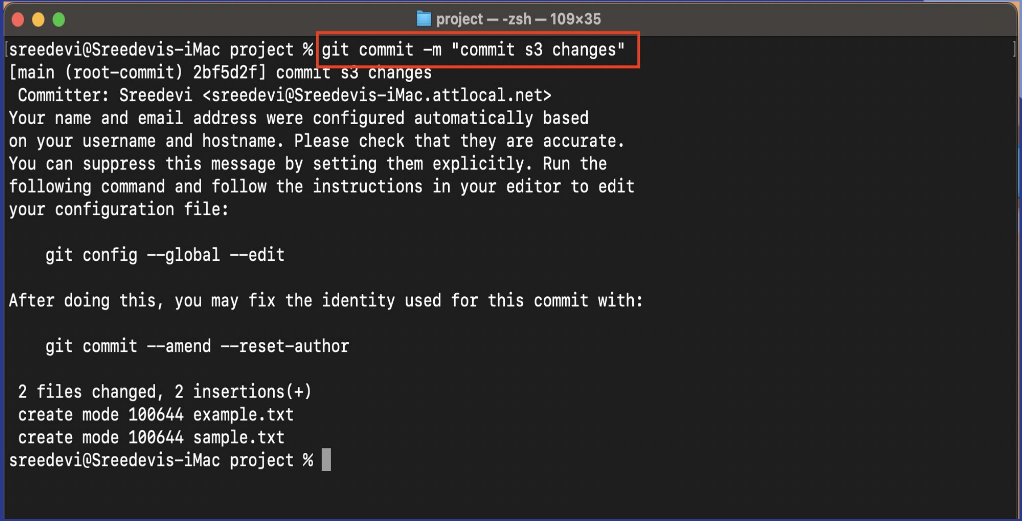
Step2 : Commit files changes to local git repository

3



```
sreedevi@Sreedevis-iMac project % touch sample.txt
sreedevi@Sreedevis-iMac project %
sreedevi@Sreedevis-iMac project % vim sample.txt
sreedevi@Sreedevis-iMac project % cat sample.txt
my s3 notes
sreedevi@Sreedevis-iMac project %
sreedevi@Sreedevis-iMac project %
sreedevi@Sreedevis-iMac project % git add sample.txt
sreedevi@Sreedevis-iMac project % git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   example.txt
        new file:   sample.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   example.txt

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

sreedevi@Sreedevis-iMac project %
```

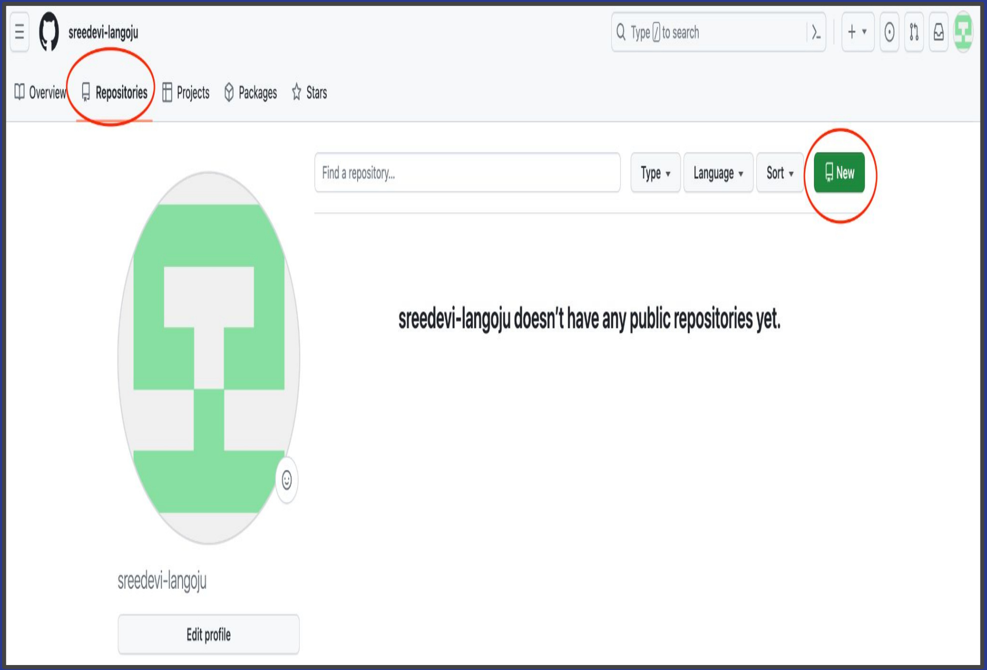
4



```
[sreedevi@Sreedevis-iMac project % git commit -m "commit s3 changes"
[main (root-commit) 2bf5d2f] commit s3 changes
 Committer: Sreedevi <sreedevi@Sreedevis-iMac.attlocal.net>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 2 files changed, 2 insertions(+)
 create mode 100644 example.txt
 create mode 100644 sample.txt
sreedevi@Sreedevis-iMac project %
```

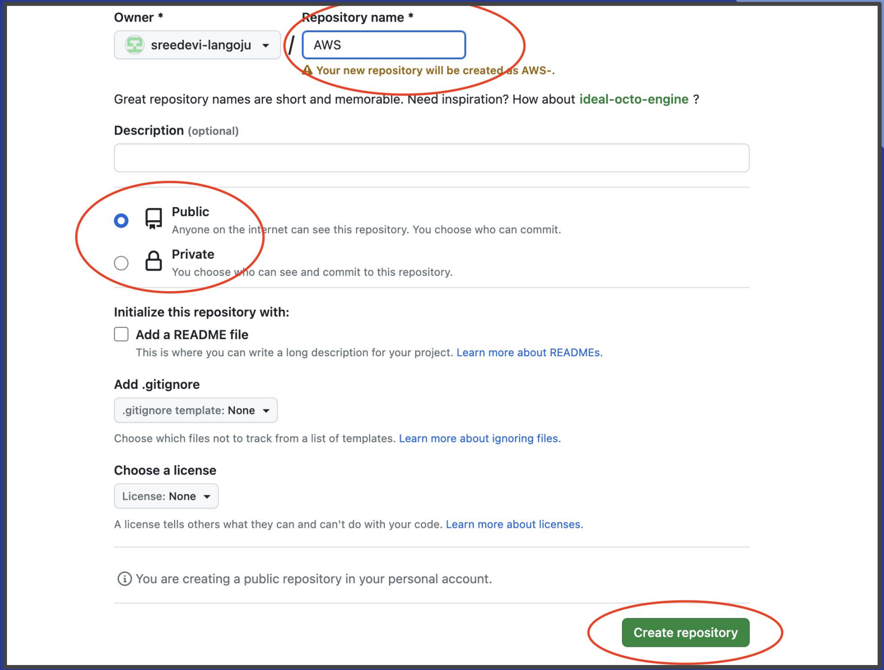
Step2 :Create a repository in Github

5



sreedevi-langoju doesn't have any public repositories yet.

6



Owner *

sreedevi-langoju /

Repository name *

AWS

Your new repository will be created as AWS-.

Great repository names are short and memorable. Need inspiration? How about ideal-octo-engine ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

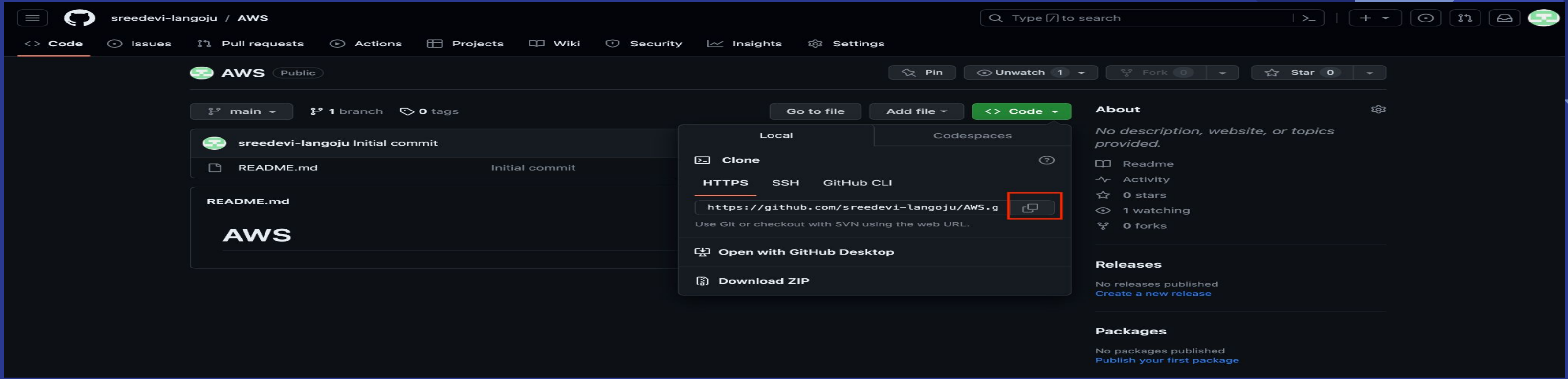
License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.

Step 3: Push local repository files to Github

7



https://github.com/sreedevi-langoju/AWS.g

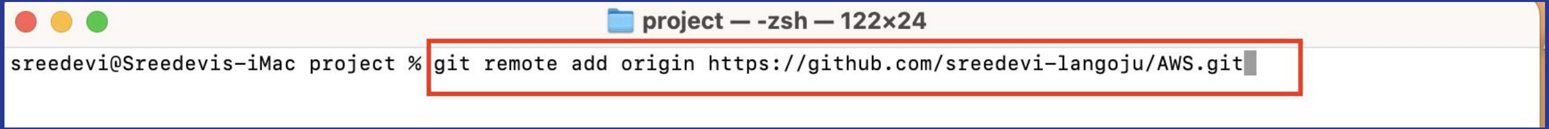
Use Git or checkout with SVN using the web URL.

No description, website, or topics provided.

No releases published

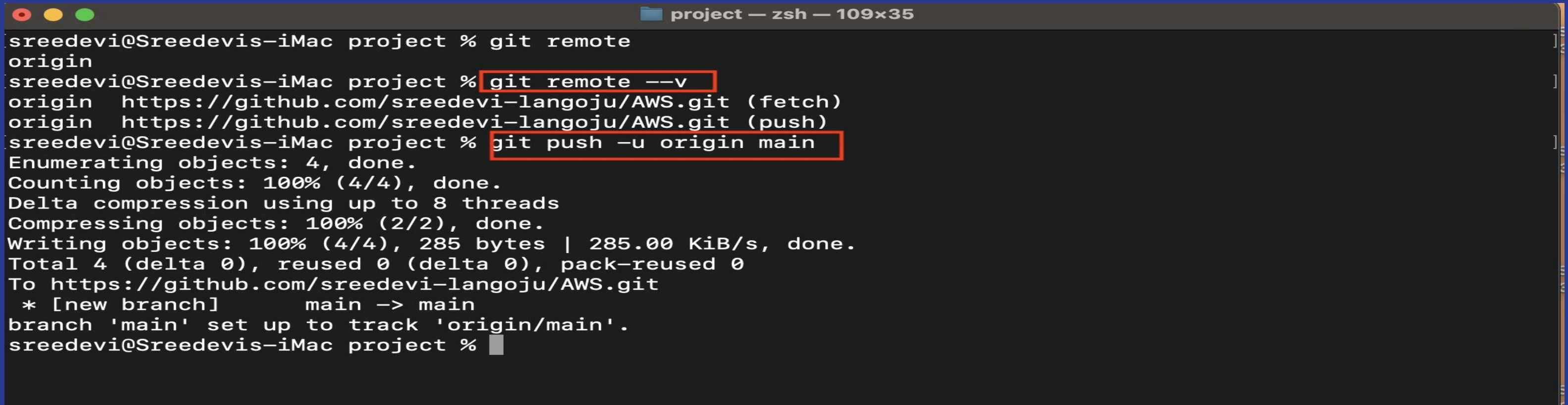
No packages published

8



```
sreedevi@Sreedevis-iMac project % git remote add origin https://github.com/sreedevi-langoju/AWS.git
```

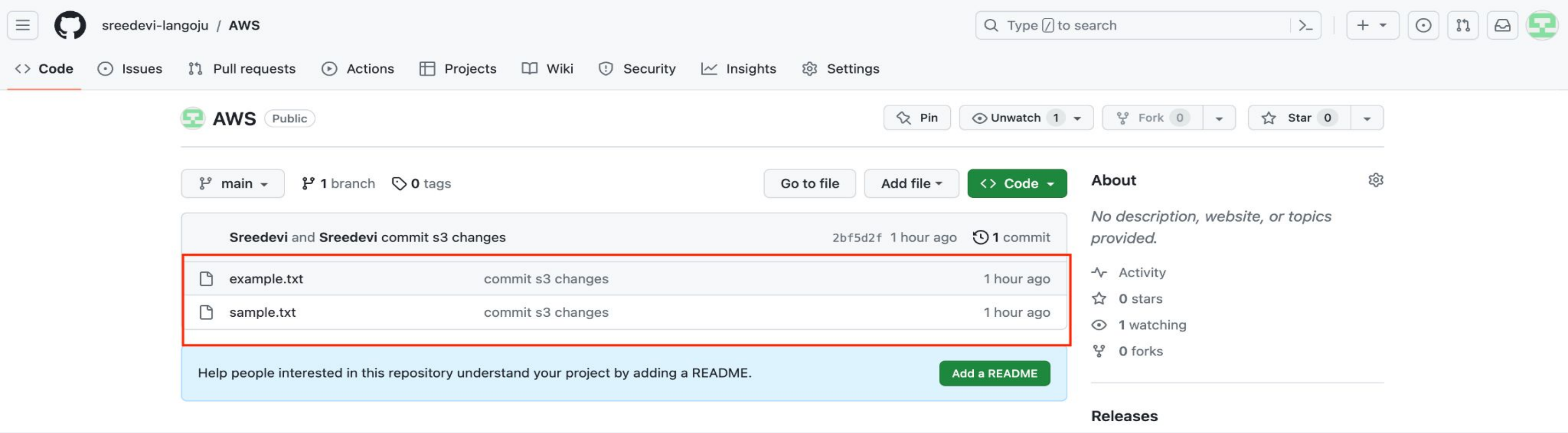
9



```
sreedevi@Sreedevis-iMac project % git remote
origin
sreedevi@Sreedevis-iMac project % git remote --v
origin  https://github.com/sreedevi-langoju/AWS.git (fetch)
origin  https://github.com/sreedevi-langoju/AWS.git (push)
sreedevi@Sreedevis-iMac project % git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 285 bytes | 285.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sreedevi-langoju/AWS.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
sreedevi@Sreedevis-iMac project %
```

Step 3: Push local repository files to Github

9



sreedevi-langoju / AWS

AWS Public

Sreedevi and Sreedevi commit s3 changes 2bf5d2f 1 hour ago 1 commit

example.txt	commit s3 changes	1 hour ago
sample.txt	commit s3 changes	1 hour ago

Help people interested in this repository understand your project by adding a README. Add a README

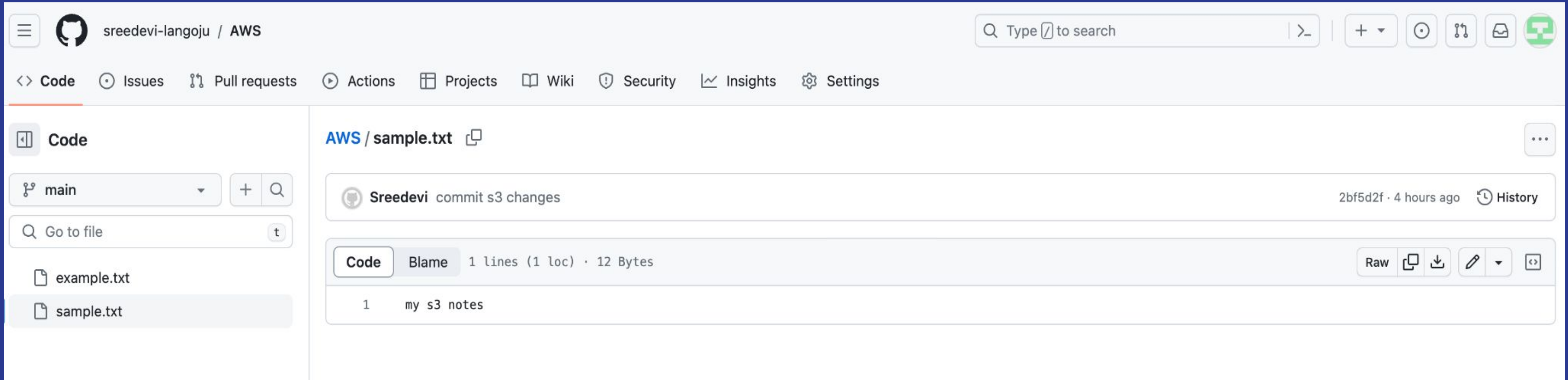
About

No description, website, or topics provided.

Activity
0 stars
1 watching
0 forks

Releases

10



sreedevi-langoju / AWS

AWS / sample.txt

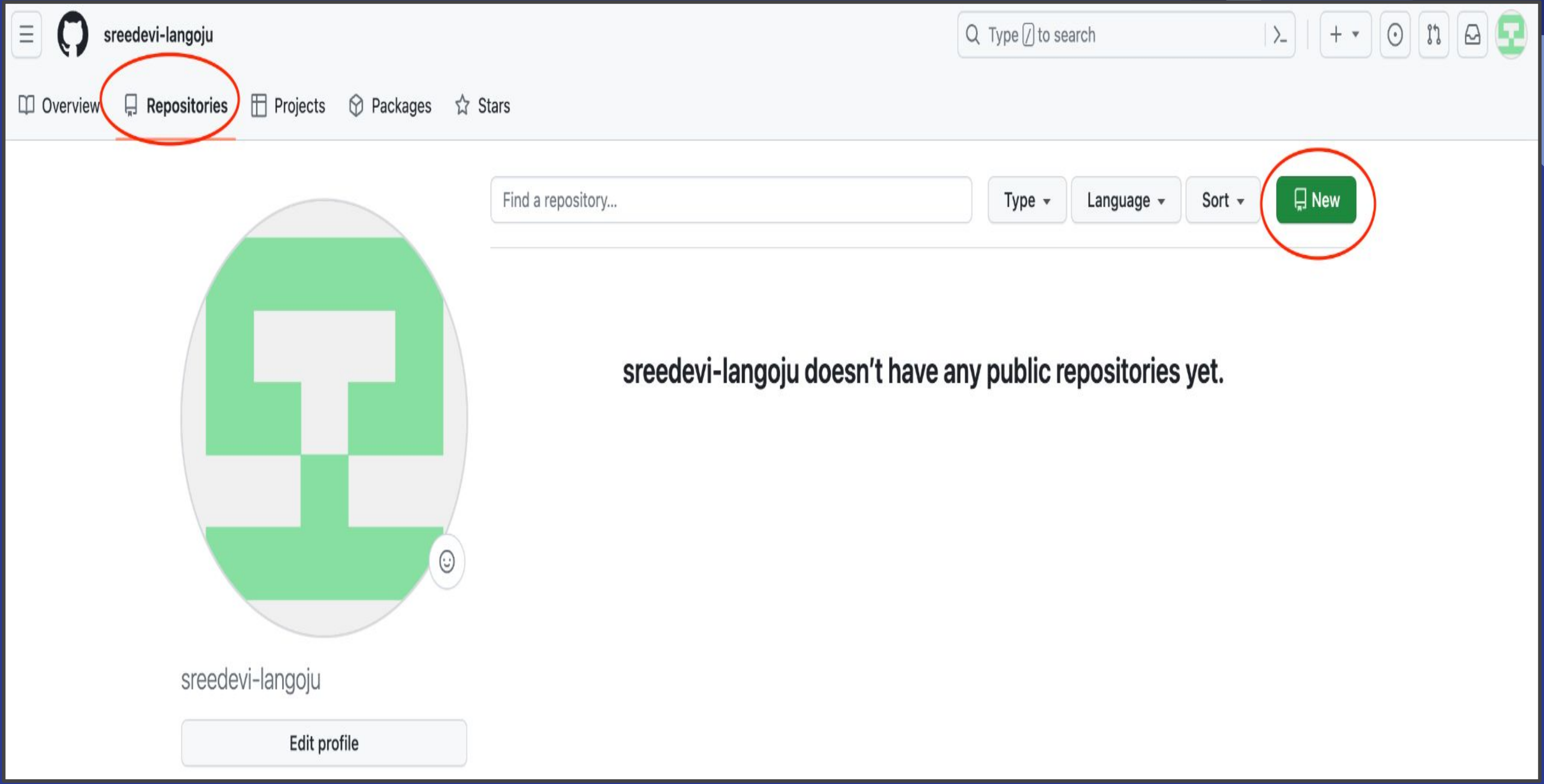
Sreedevi commit s3 changes 2bf5d2f · 4 hours ago History

1 lines (1 loc) · 12 Bytes

```
my s3 notes
```

Creating a new repository in GitHub

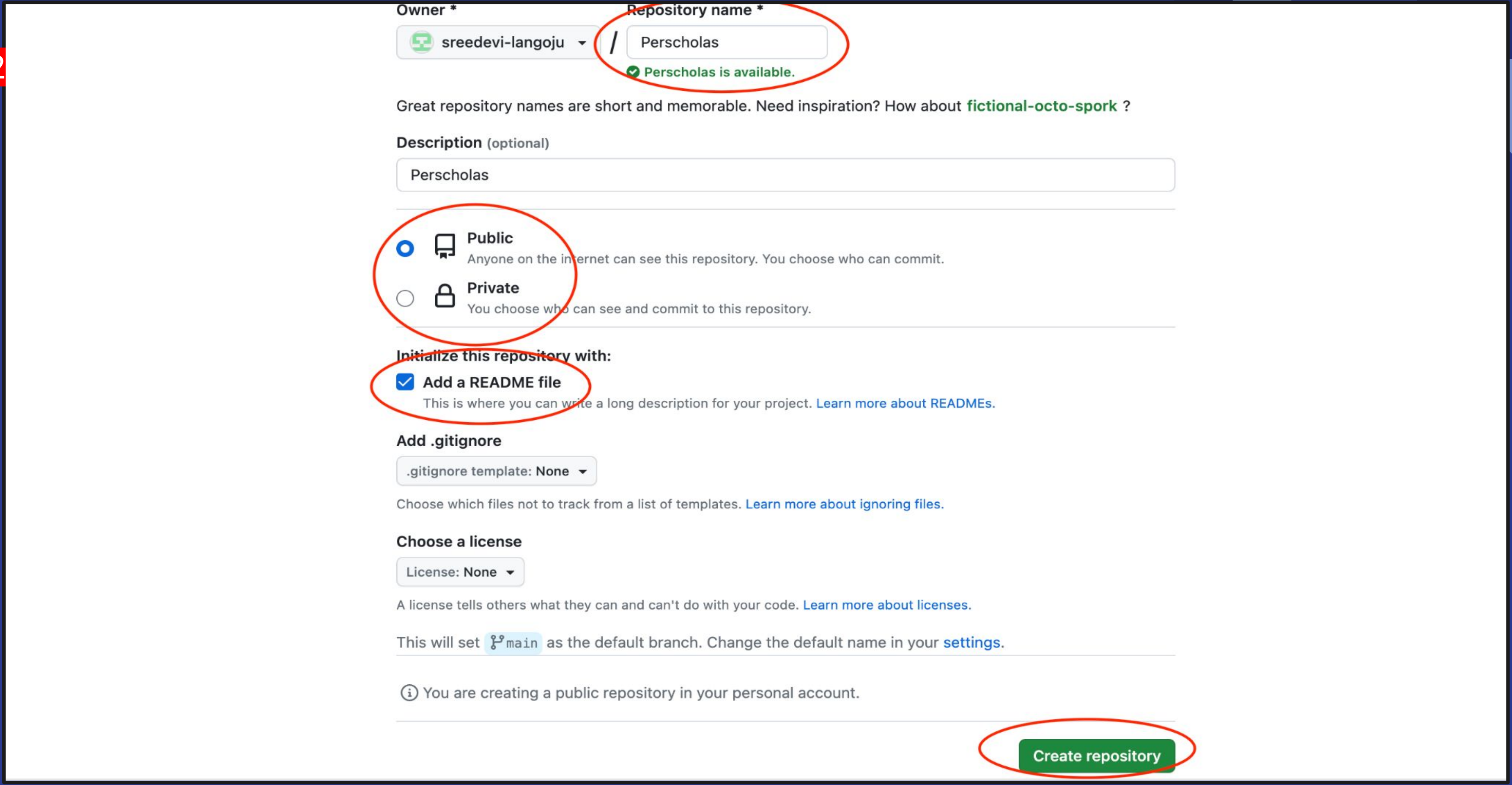
1



sreedevi-langoju doesn't have any public repositories yet.

Creating new repository in GitHub

2



Owner * sreedevi-langoju / Repository name * Perscholas

Perscholas is available.

Great repository names are short and memorable. Need inspiration? How about fictional-octo-spork ?

Description (optional)

Perscholas

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☑ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

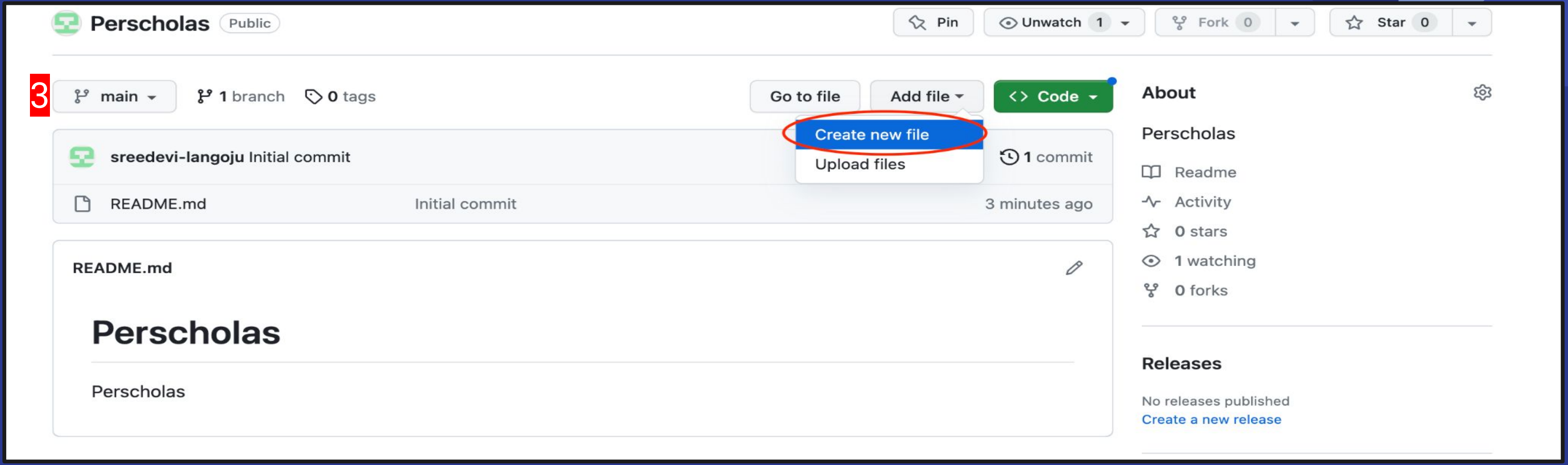
A license tells others what they can and can't do with your code. Learn more about licenses.

This will set `main` as the default branch. Change the default name in your settings.

You are creating a public repository in your personal account.

How to create and commit a file in Github

3



README.md

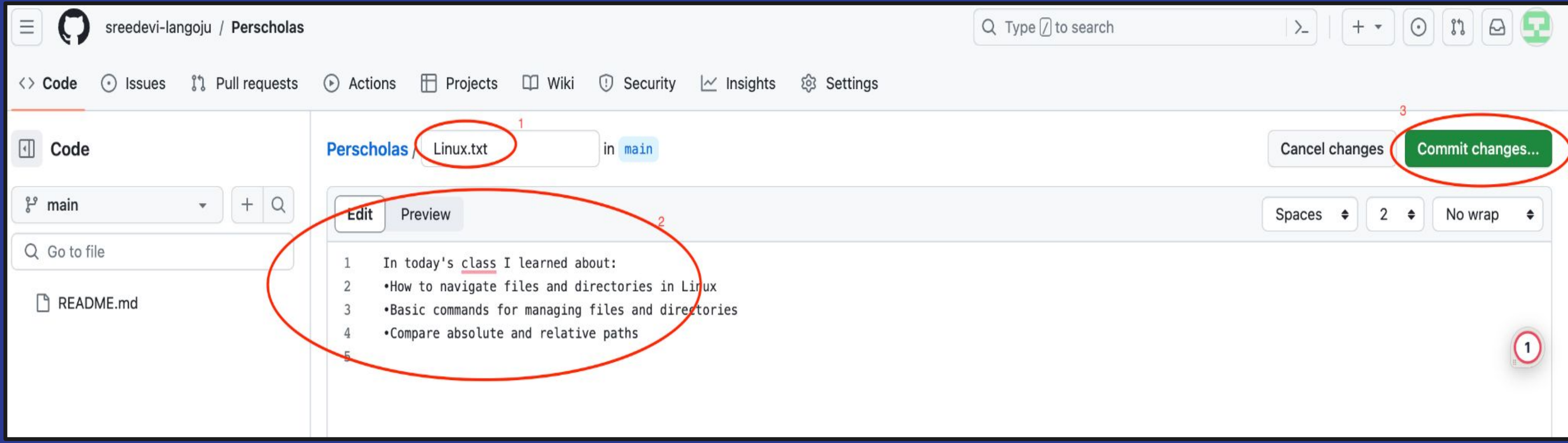
Perscholas

Perscholas

About

Perscholas

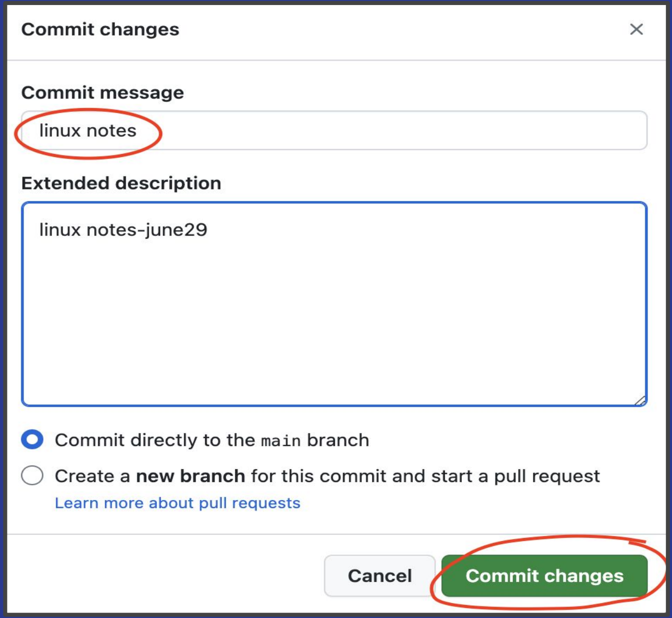
4



```
In today's class I learned about:
•How to navigate files and directories in Linux
•Basic commands for managing files and directories
•Compare absolute and relative paths
```

How to create and commit a file in github

5



Commit changes

Commit message

linux notes

Extended description

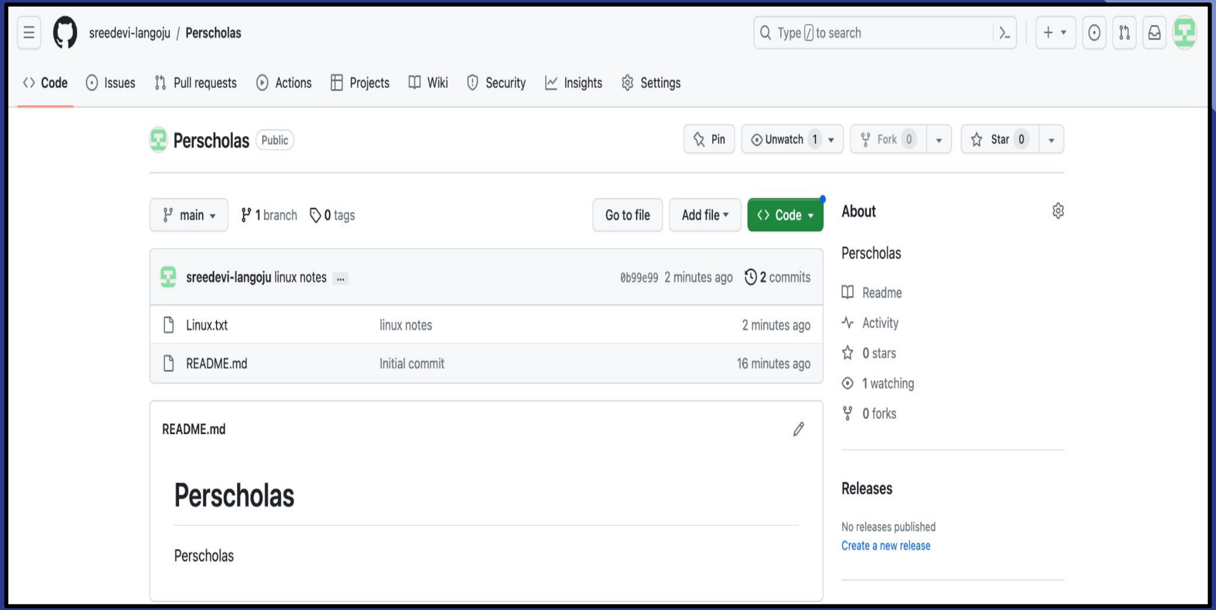
linux notes-june29

● Commit directly to the `main` branch

○ Create a **new branch** for this commit and start a pull request

Learn more about pull requests

6



sreedevi-langoju / Perscholas

Perscholas Public

sreedevi-langoju linux notes ... 0b99e99 2 minutes ago 2 commits

Linux.txt	linux notes	2 minutes ago
README.md	Initial commit	16 minutes ago

README.md

Perscholas

Perscholas

About

Perscholas

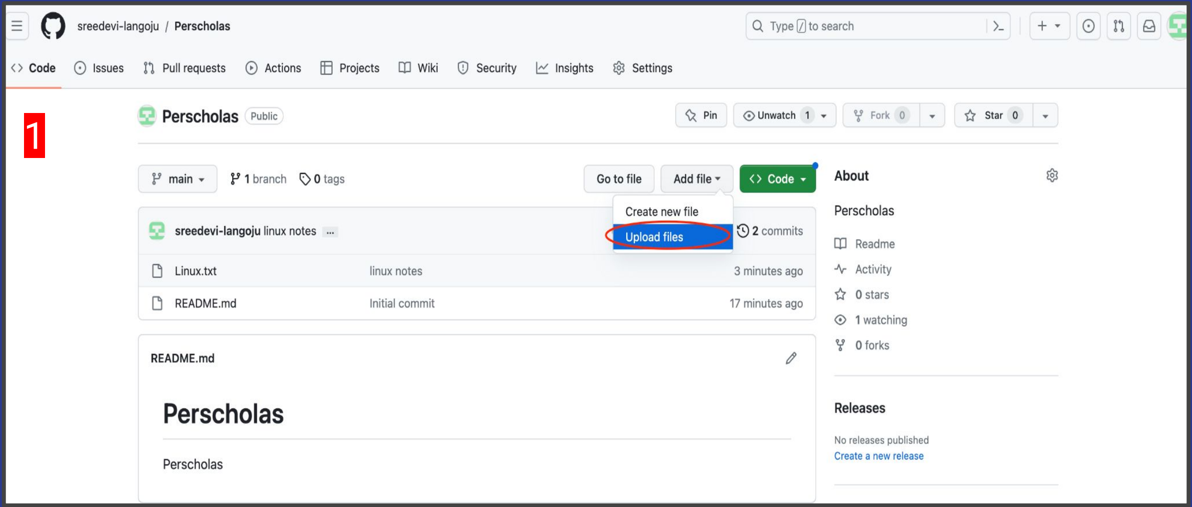
- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

Releases

No releases published

Create a new release

How to upload and commit a file in Github



sreedevi-langoju linux notes ... 2 commits

Linux.txt	linux notes	3 minutes ago
README.md	Initial commit	17 minutes ago

README.md

Perscholas

Perscholas

About

Perscholas

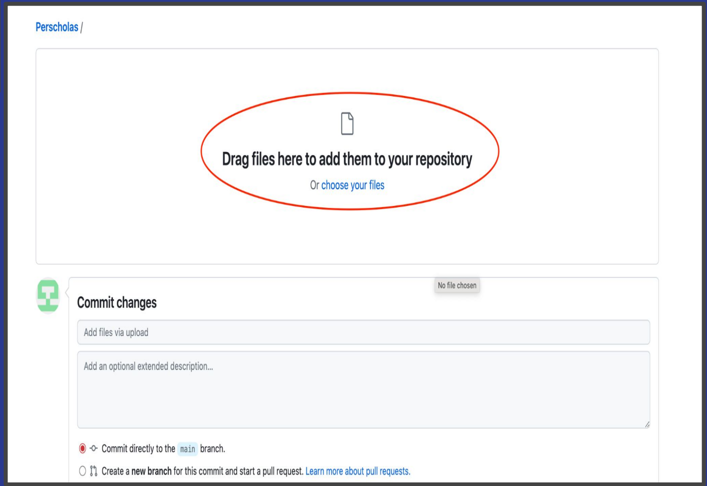
- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

Releases

No releases published

Create a new release

2



Drag files here to add them to your repository

Or choose your files

Commit changes

Add files via upload

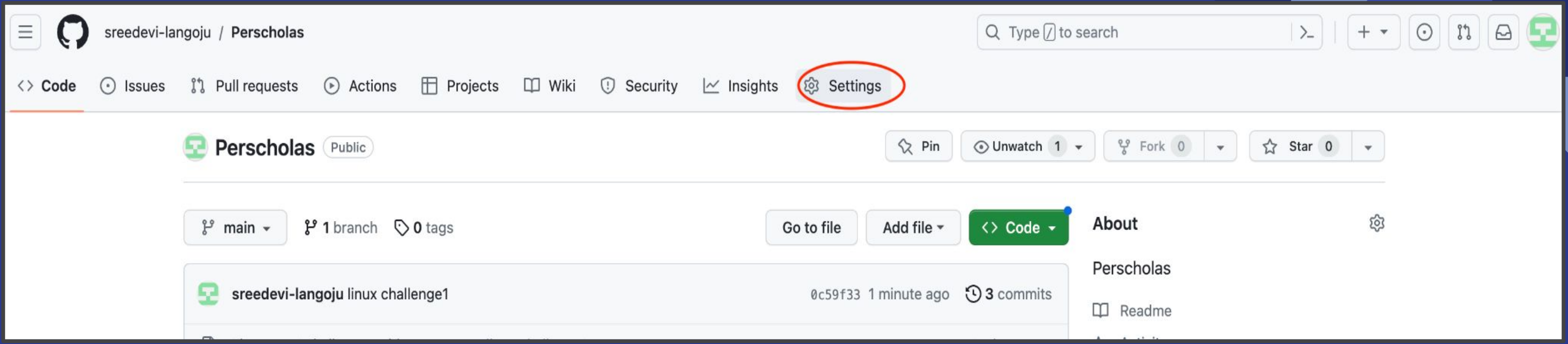
Add an optional extended description...

● Commit directly to the main branch.

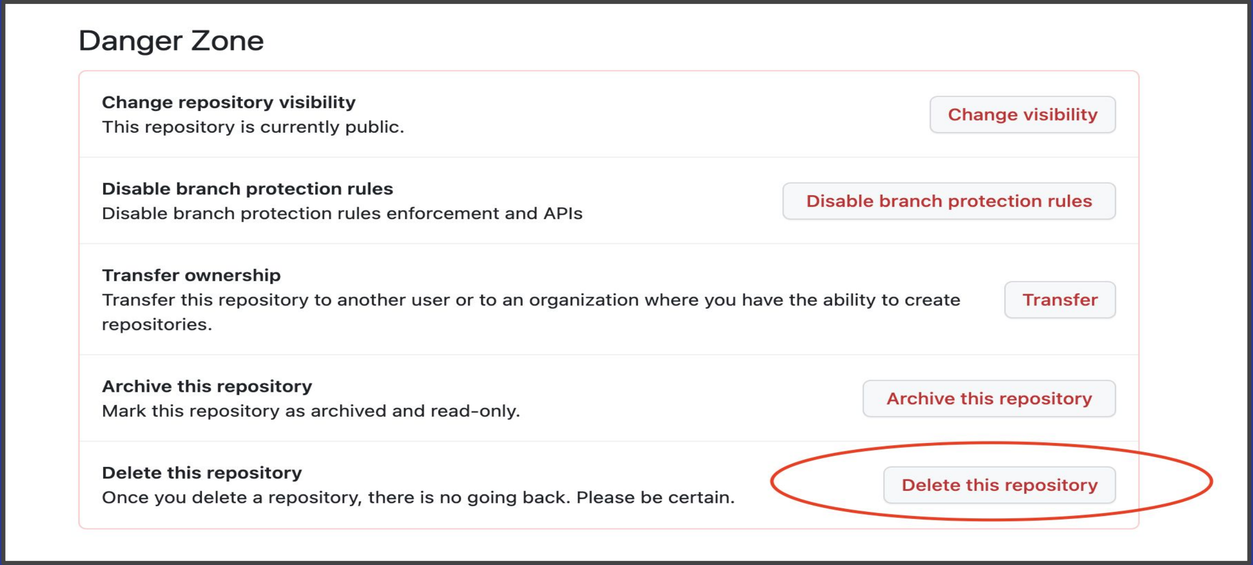
○ Create a new branch for this commit and start a pull request. Learn more about pull requests.

How to delete a repository in Github

1



2



Danger Zone

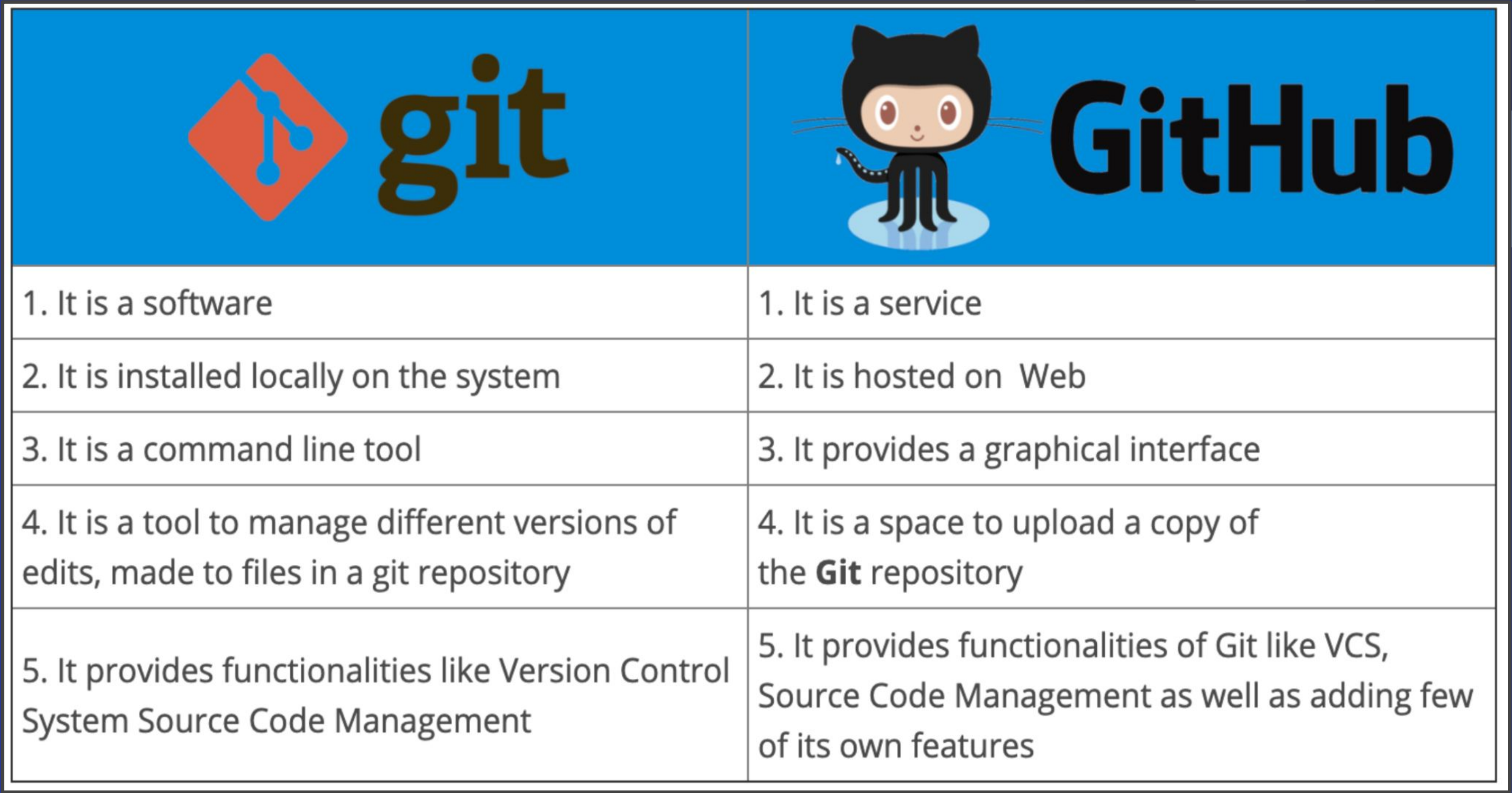
Change repository visibility
This repository is currently public.

Disable branch protection rules
Disable branch protection rules enforcement and APIs

Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories.

Archive this repository
Mark this repository as archived and read-only.

Delete this repository
Once you delete a repository, there is no going back. Please be certain.



git	GitHub
1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the **Git** repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

Thank You