

Lesson 01 Demo 03

Pushing Files to GitHub Repository

Objective: To push a file to a GitHub repository to update the repository with new changes, making them accessible to collaborators and version control tracking

Prerequisite: GitHub Account

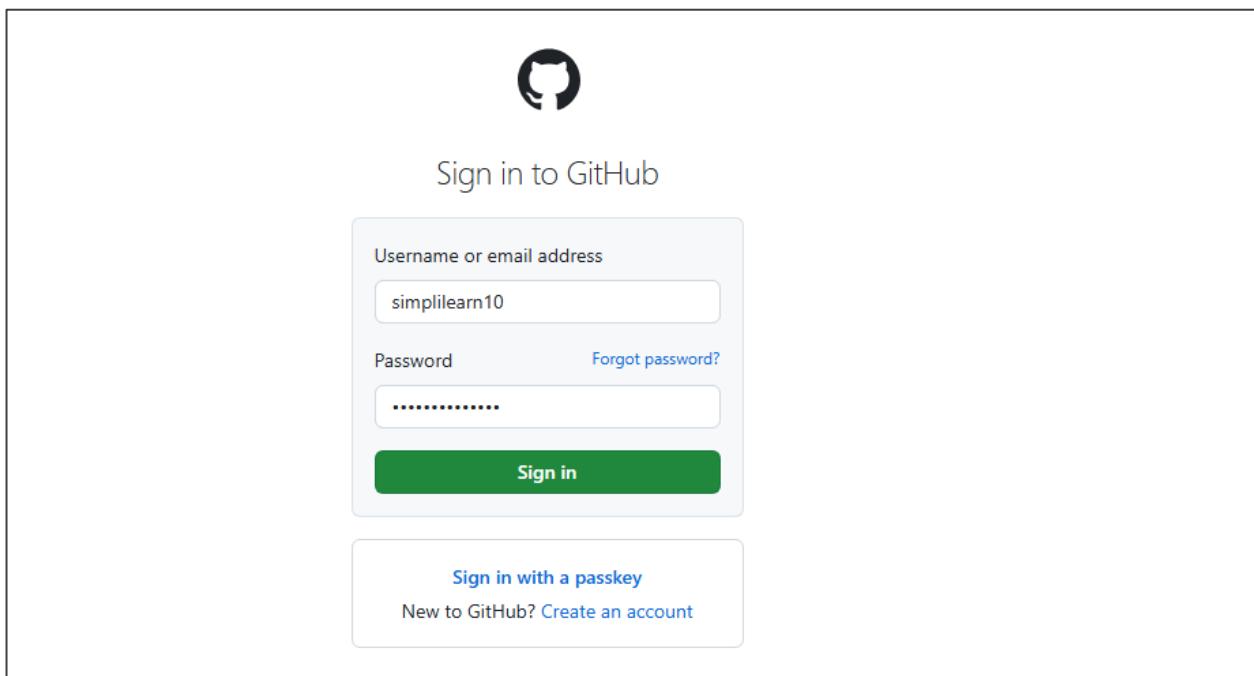
Tools required: Git

Steps to be followed:

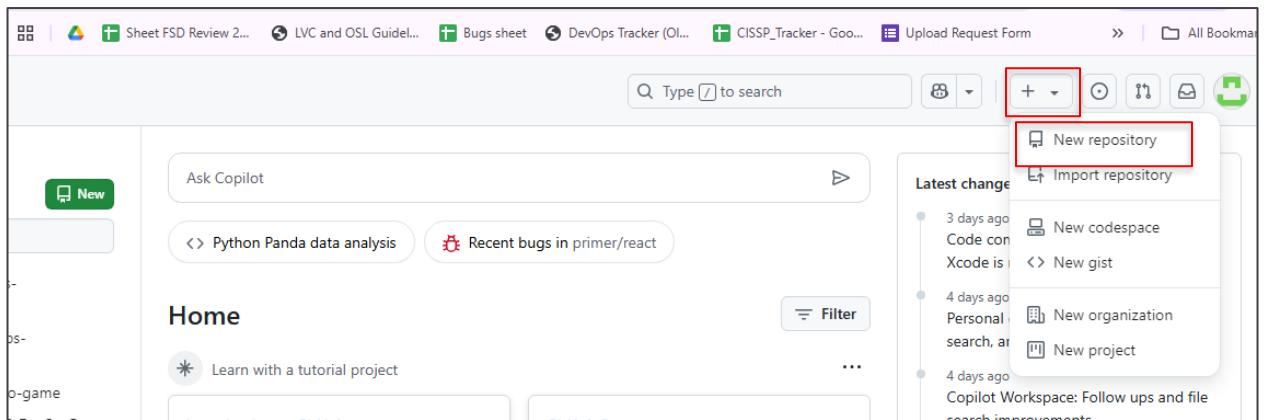
1. Create a GitHub repository
2. Create a repository on the local machine
3. Push the changes from local repository to GitHub
4. Check the status of the local and remote repository

Step 1: Create a GitHub repository

- 1.1 Go to GitHub.com and log in to your account



1.2 Click on the + icon from the upper-right corner of the page and select **New repository** from the drop-down menu



1.3 Enter the Repository name and click on **Create repository** button

The screenshot shows a 'Create a new repository' form. At the top, it says 'Create a new repository'. Below that, it asks if there is an existing repository elsewhere and provides a link to 'Import a repository'. It then states that required fields are marked with an asterisk (*). The 'Owner' field is set to 'simplilearn10'. The 'Repository name' field is highlighted with a red box and contains the text 'my new project'. Below the repository name, there is a note: 'your new repository will be created as my-new-project.' and a warning: 'The repository name can only contain ASCII letters, digits, and the characters ., -, and _.' There is a 'Description (optional)' field with a placeholder text area. At the bottom, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone on the internet can see this repository. You choose who can commit.' and the 'Private' option is described as 'You choose who can see and commit to this repository.'

Required fields are marked with an asterisk (*).

Owner *

Repository name *

✓ your new repository will be created as my-new-project.
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about [probable-carnival](#) ?

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.

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 2: Create a repository on the local machine

2.1 In the terminal, execute the following commands to create and initialize a Git repository:

```
mkdir createnewproject
cd createnewproject
echo "# create new file for my project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
```

```
manikumar@ip-172-31-71-23:~$ mkdir createnewproject
manikumar@ip-172-31-71-23:~$ cd createnewproject
manikumar@ip-172-31-71-23:~/createnewproject$ echo "# create new file for my project" >> README.md
manikumar@ip-172-31-71-23:~/createnewproject$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/manikumarsimpli/createnewproject/.git/
manikumar@ip-172-31-71-23:~/createnewproject$ █
```

```
manikumarsimpli@ip-172-31-71-23:~/createnewproject$ git add README.md
manikumarsimpli@ip-172-31-71-23:~/createnewproject$ git commit -m "first commit"
[master (root-commit) 016a159] first commit
Committer: manikumarsimpli <manikumarsimpli@ip-172-31-71-23.ec2.internal>
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
```

1 file changed, 1 insertion(+)
create mode 100644 README.md

```
manikumarsimpli@ip-172-31-71-23:~/createnewproject$ git branch -M main
manikumarsimpli@ip-172-31-71-23:~/createnewproject$
```

Step 3: Push the changes from local repository to GitHub

3.1 Open the terminal on the local machine, then use the following command to add a remote repository and push the changes to the remote repository:

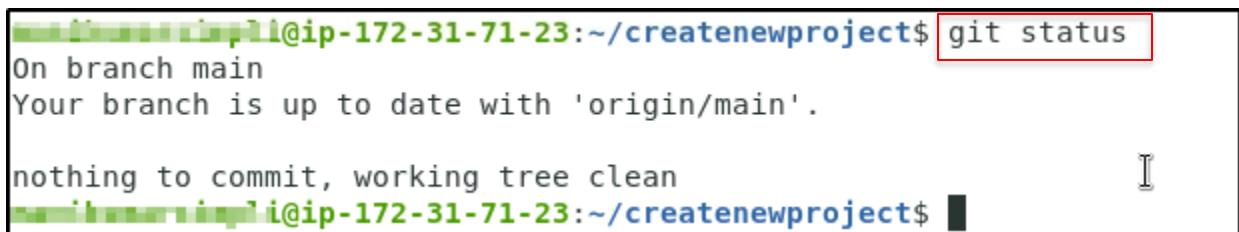
```
git remote add origin <Your HTTPS_URL>
git push -u origin main
```

```
manikumarsimpli@ip-172-31-71-23:~/createnewproject$ git remote add origin https://github.com/GithubWorkstation/my-new-project.git
manikumarsimpli@ip-172-31-71-23:~/createnewproject$ git push -u origin main
Username for 'https://github.com': GithubWorkstation
Password for 'https://GithubWorkstation@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 258 bytes | 258.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/GithubWorkstation/my-new-project.git
 * [new branch]    main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
manikumarsimpli@ip-172-31-71-23:~/createnewproject$
```

Step 4: Check the status of the local and remote repository

4.1 Run the following command to check the status of the local repository:

```
git status
```

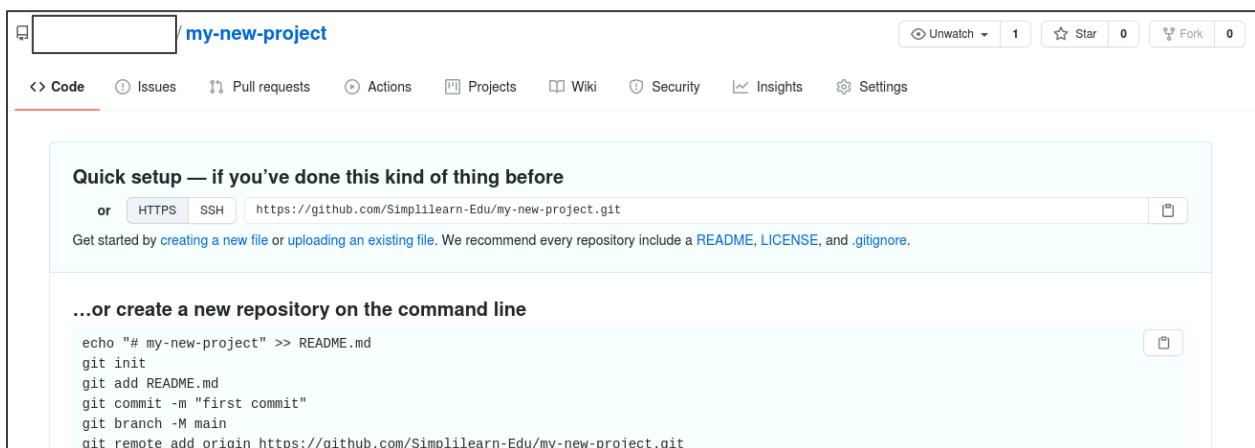


```
shalin@ip-172-31-71-23:~/createnewproject$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
shalin@ip-172-31-71-23:~/createnewproject$
```

A terminal window showing the output of the 'git status' command. The command is highlighted with a red box. The output shows the branch is 'main' and it is up to date with 'origin/main'. There is nothing to commit and the working tree is clean.

4.2 Go to GitHub.com and check the remote repository



By following these steps, you have successfully pushed a file to a GitHub repository to update the repository with new changes, making them accessible to collaborators and version control tracking.