

Lesson 04 Demo 03

Adding a File to GitHub Repository

Objective: To push a file to a GitHub repository for backup, collaboration, and development workflow integration

Tools required: Git and GitHub

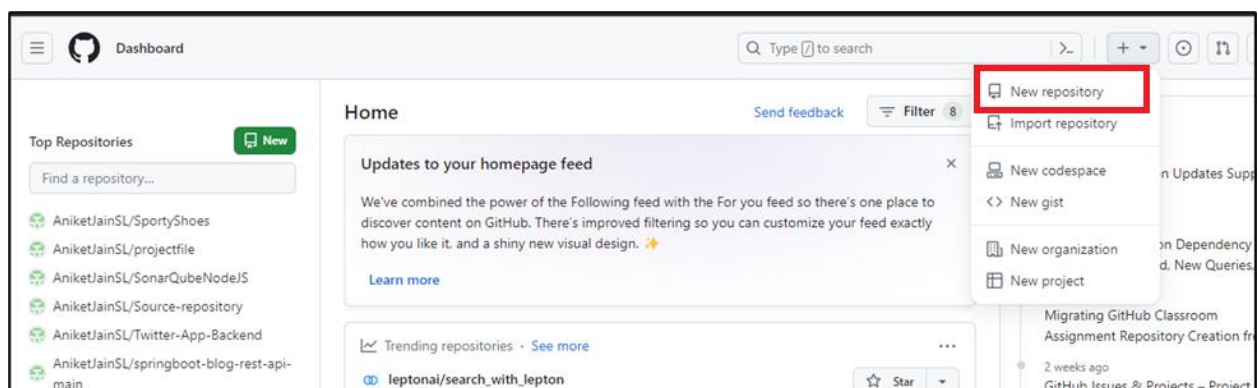
Prerequisites: None

Steps to be followed:

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

Step 1: Create a GitHub repository

- 1.1 Navigate to **github.com**, log in to your account, and click on the + button to create a new repository




1.2 Enter an arbitrary repository name and click on **Create Repository** button

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk (*).

Owner *


AniketJainSL


Repository name *


my new project

 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 [urban-potato](#) ?

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:

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

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$
```

```
manikumar@ip-172-31-71-23:~/createnewproject$ git add README.md
manikumar@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
manikumar@ip-172-31-71-23:~/createnewproject$ git branch -M main
manikumar@ip-172-31-71-23:~/createnewproject$
```

Step 3: Push the changes from the local repository to GitHub

- 3.1 Open the terminal on the local machine and use the following command to add a remote repository:

git remote add origin <Your HTTPS_URL>

```
root@ip-172-31-71-23:~/createnewproject$ git remote add origin https://github.com/GithubWorkstation/my-new-project.git
```

- 3.2 Use the following command to push the changes to the remote repository:

git push -u origin main

```
root@ip-172-31-71-23:~/createnewproject$ git remote add origin https://github.com/GithubWorkstation/my-new-project.git
root@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'.
root@ip-172-31-71-23:~/createnewproject$
```

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

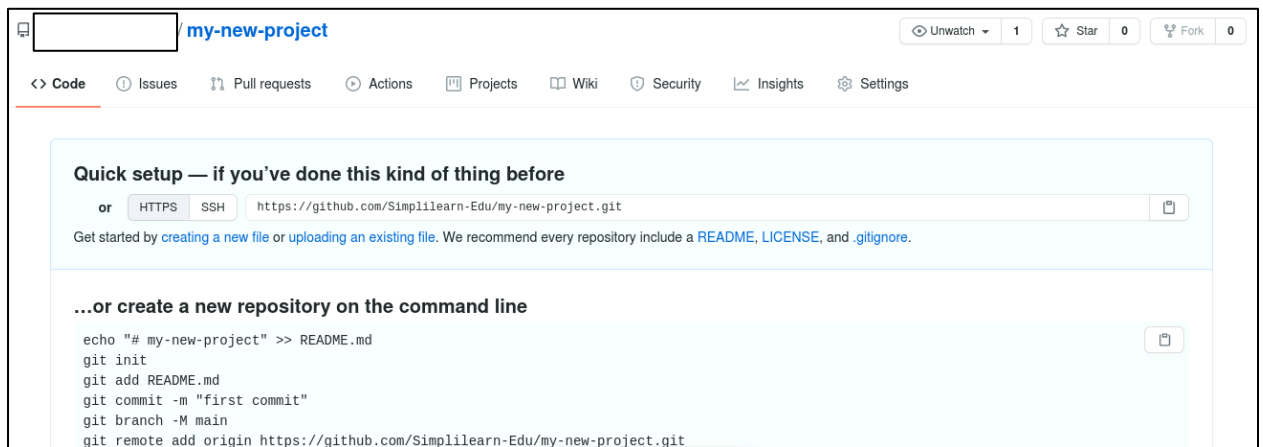
- 4.1 Execute the following command to check the status of the local repository

git status

```
root@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
root@ip-172-31-71-23:~/createnewproject$
```

4.2 Navigate to **github.com** and check the status of the remote repository



You can see that the changes have been pushed to the local repository **my new project**.

By following these steps, you have successfully pushed a file to GitHub for backup, collaboration, and to enable various development workflows.