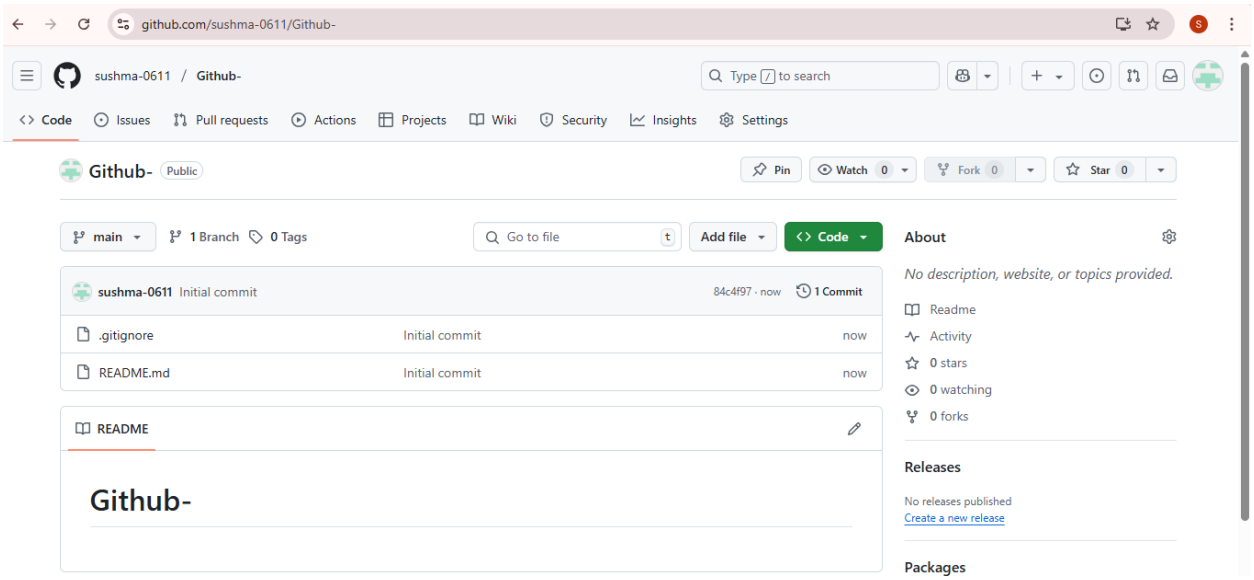
1)Install git.

```
Installed:
  git.x86_64 0:2.47.1-1.amzn2.0.2

Dependency Installed:
  git-core.x86_64 0:2.47.1-1.amzn2.0.2
  git-core-doc.noarch 0:2.47.1-1.amzn2.0.2
  perl-Error.noarch 1:0.17020-2.amzn2
  perl-Git.noarch 0:2.47.1-1.amzn2.0.2
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[root@ip-172-31-95-66 ec2-user]# git --version
git version 2.47.1
```

2)Create a repo in GitHub with README.md and ignore file.



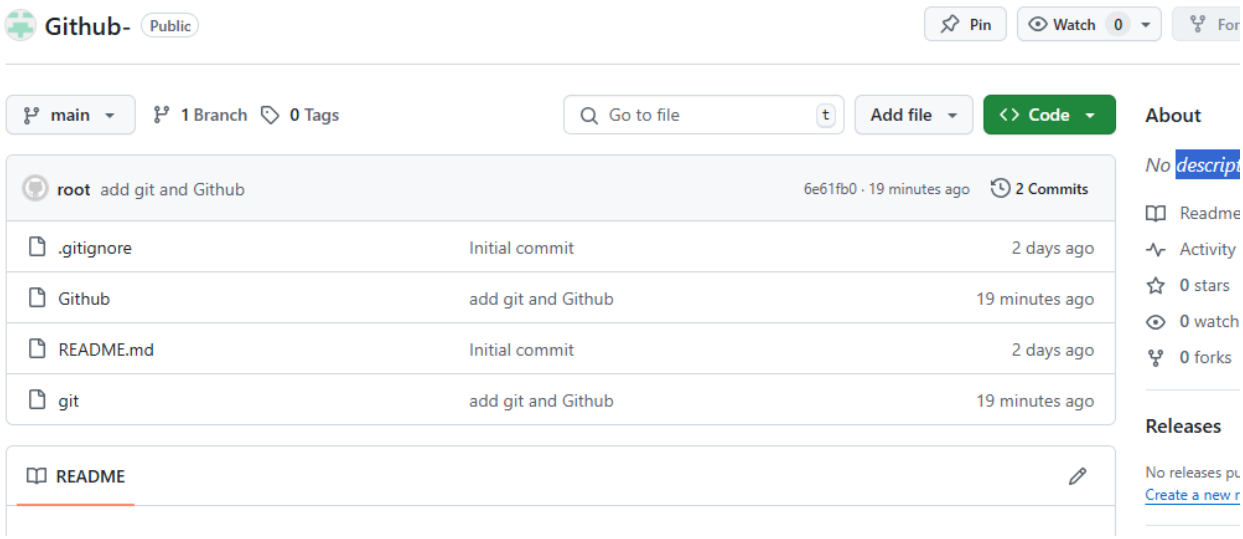
3)Clone the created repo to local.

```
[root@ip-172-31-95-66 ec2-user]# git clone https://github.com/sushma-0611/Github-.git
Cloning into 'Github-'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
```

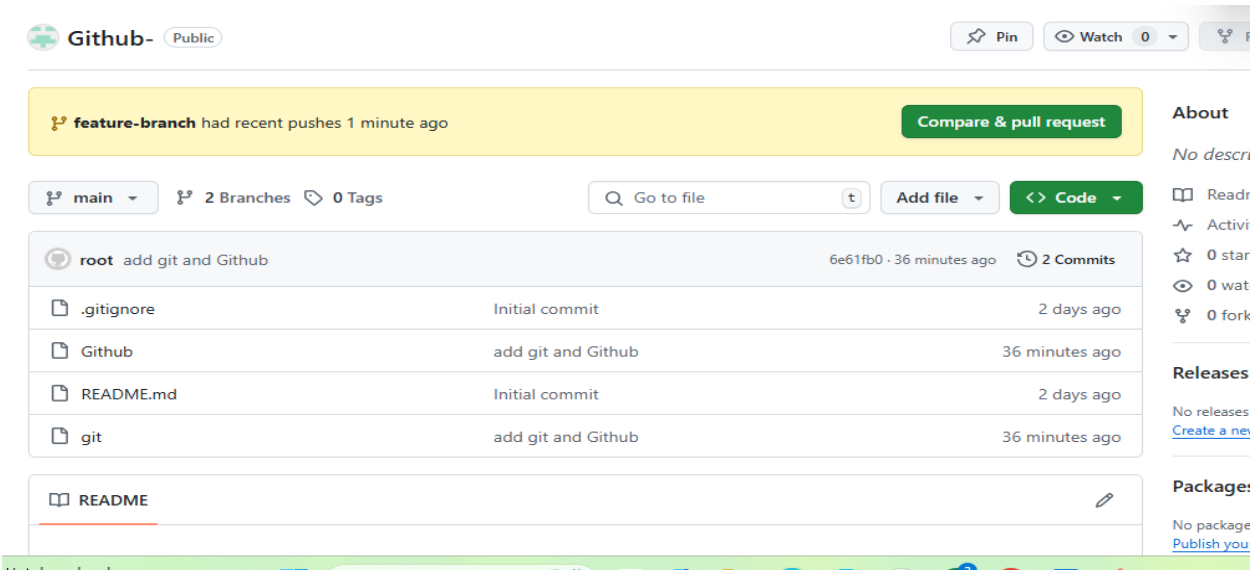
4)Create two files in local repo.

```
[root@ip-172-31-95-66 ~]# cd Github-/
[root@ip-172-31-95-66 Github-]# ls
README.md
[root@ip-172-31-95-66 Github-]# touch git
[root@ip-172-31-95-66 Github-]# touch Github
[root@ip-172-31-95-66 Github-]# ls
git  Github  README.md
```

5)Commit two files and push to central Repository.



6)Create a branch in local and create a sample file and push to central.



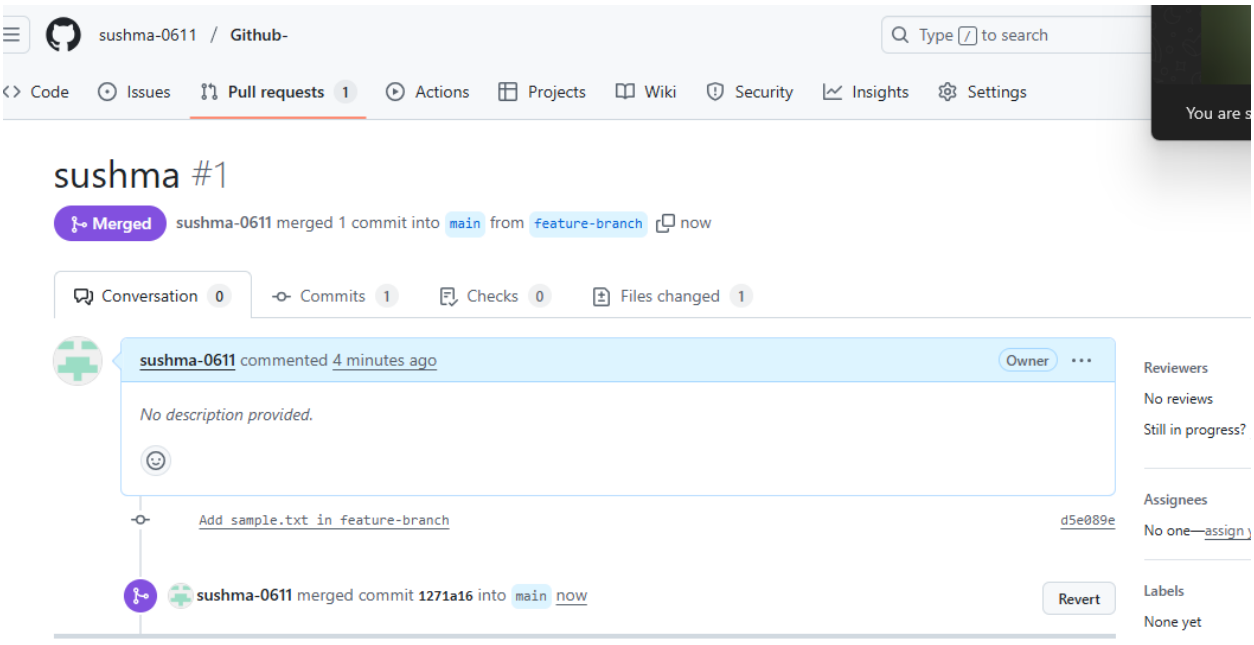
7)Create a branch in GitHub and clone that to local.

```
[root@ip-172-31-95-66 Github-]# git fetch origin
From https://github.com/sushma-0611/Github-
 * [new branch]      sandeep-branch -> origin/sandeep-branch
[root@ip-172-31-95-66 Github-]# git checkout sandeep-branch
branch 'sandeep-branch' set up to track 'origin/sandeep-branch'.
Switched to a new branch 'sandeep-branch'
[root@ip-172-31-95-66 Github-]# git branch
  feature-branch
  main
* sandeep-branch
```

8)Merge the created branch with master in git local.

```
[root@ip-172-31-95-66 Github-]# git push origin main
Username for 'https://github.com': sushma-0611
Password for 'https://sushma-0611@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 295 bytes | 295.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sushma-0611/Github-.git
   6e61fb0..c6aaf9e  main -> main
```

9)Merge the created branch with master in GitHub by sending a pull request.



10)create a file in local and send that to branch in GitHub.

```
[root@ip-172-31-95-66 Github-]# git push origin sandeep-branch
Username for 'https://github.com': sushma-0611
Password for 'https://sushma-0611@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sushma-0611/Github-.git
   6e61fb0..e589347  sandeep-branch -> sandeep-branch
```

11)clone only a branch from GitHub to local.

```
[root@ip-172-31-95-66 sushma]# git clone --branch sandeep-branch --single-branch
Cloning into 'Github-'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 10 (delta 1), reused 6 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (1/1), done.
```

12)create a file with all passwords and make that untrackable with git.

```
[root@ip-172-31-95-66 Github-]# git push origin sandeep-branch
Username for 'https://github.com': sushma-0611
Password for 'https://sushma-0611@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/sushma-0611/Github-.git
   e589347..9c30445  sandeep-branch -> sandeep-branch
```

13)make a commit and make that commit reset without savings changes.

```
[root@ip-172-31-95-66 Github-]# git push origin sandeep-branch
Username for 'https://github.com': sushma-0611
Password for 'https://sushma-0611@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/sushma-0611/Github-.git
   e589347..9c30445  sandeep-branch -> sandeep-branch
[root@ip-172-31-95-66 Github-]# git add .gitignore
[root@ip-172-31-95-66 Github-]# git add .gitignore
[root@ip-172-31-95-66 Github-]# git commit -m "Ignore secrets.txt file"
On branch sandeep-branch
Your branch is up to date with 'origin/sandeep-branch'.

nothing to commit, working tree clean
[root@ip-172-31-95-66 Github-]# git reset --hard HEAD~1
HEAD is now at e589347 Add newfile.txt to sandeep-branch
```

14)Revert a committed commit to the older version.

```
[root@ip-172-31-95-66 Github-]# git log --oneline
e589347 (HEAD -> sandeep-branch) Add newfile.txt to sandeep-branch
5e61fb0 add git and Github
84c4f97 Initial commit
[root@ip-172-31-95-66 Github-]# git revert <commit hash>
```

15)push a file to stash without savings the changes and work on another file.

```
PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
$ git stash push file1.txt
Saved working directory and index state WIP on main: 84c4f97 Initial commit

PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
```

16)undo the stash file and start working on that again.

```
PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
$ git stash pop
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file1.txt

Dropped refs/stash@{0} (50b6dda86b4ef611d47f089587d41c6d0ecbc158)

PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
$ ls
README.md  file1.txt

PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
$ cat file1.txt
test content
```

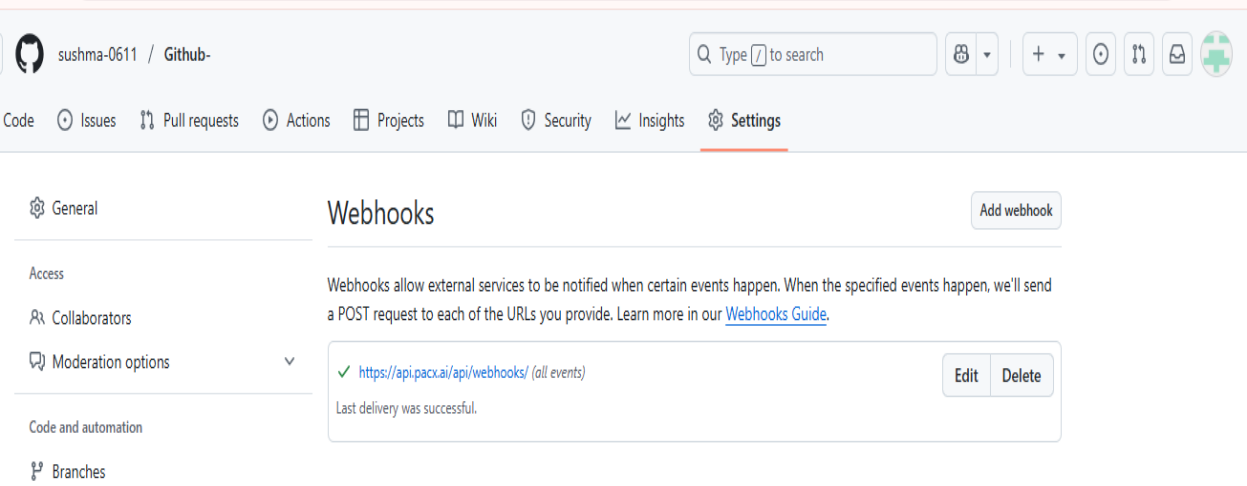
17)generate a ssh-keygen and configure into GitHub.

```
PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
$ ssh-keygen -t ed25519 -C "deegojusushma23@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/PC/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/PC/.ssh/id_ed25519
Your public key has been saved in /c/Users/PC/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:LPNkCk/eaCPFbiucA5i3W2Wo+O4H3yo6TL7fE9NTi7A deegojusushma23@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|                 |
|                 |
|     + . .       |
| o  o % S .      |
|oo+. E & .       |
|=..*oo% +        |
|.+o.O=.o         |
|.*B=.=o          |
+----[SHA256]-----+
```

```
PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
$ ssh -T git@github.com
Enter passphrase for key '/c/Users/PC/.ssh/id_ed25519':
git@github.com: Permission denied (publickey).

PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
$ ssh -T git@github.com
Enter passphrase for key '/c/Users/PC/.ssh/id_ed25519':
Hi sushma-0611! You've successfully authenticated, but GitHub does not provide s
hell access.
```

18)configure webhooks to GitHub.



19)basic understanding of .git file.

The .git folder is a hidden directory located at the root of every Git repository. It stores all the metadata and objects that Git needs to track your project history.
When you run:
bash
CopyEdit
git init
Git creates the .git folder to start managing your project.
Think of .git as Git's database – it contains everything needed to:
Track changes
Manage branches
Handle commits
Perform merges, rebases, etc.

Folder/File	What it does
HEAD	Points to the current branch you have checked out
config	Project-level Git configuration settings
descripti	Description of the repo (used by some UIs, not Git.

Folder/File	What it does
on itself)	
refs/	Stores references like branches and tags
objects/	Contains all data (commits, trees, blobs –

everything is hashed)
logs/ Records updates to refs – helpful for recovery
index Tracks staged files (your staging area)
hooks/ Scripts that run at certain points in the Git workflow
info/ Misc info like excluded files (exclude is like .gitignore)
rebase-* Temporary folders when rebasing (you're seeing this now)
Example Workflow Summary:
1. You make changes → Git sees them.
2. You stage them → Stored in index.
3. You commit → Snapshot saved to objects/ as a commit.
4. You switch branches → HEAD points to another ref in refs/heads/.
5. You push/pull → Git uses .git/config and remote refs.

20)Check all the logs of git.

```
PC@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
$ git log
commit 84c4f97a522cb9cb03b2c327efa614ce902cedc0 (HEAD -> main, origin/main, orig
in/HEAD)
Author: sushma-0611 <deegojusushma23@gmail.com>
Date:   Fri Jun 6 17:50:18 2025 +0530

    Initial commit
```

21)Rename the commit message.

```
C@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
 git commit --amend -m "New commit message"
main 444a61b] New commit message
Author: sushma-0611 <deegojusushma23@gmail.com>
Date: Fri Jun 6 17:50:18 2025 +0530
3 files changed, 24 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 file1.txt

C@DESKTOP-8IMOQCN MINGW64 ~/Github- (main)
```

22)Merge multiple commits into single commit.