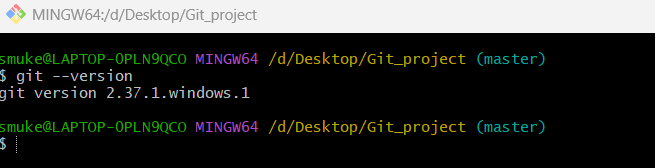
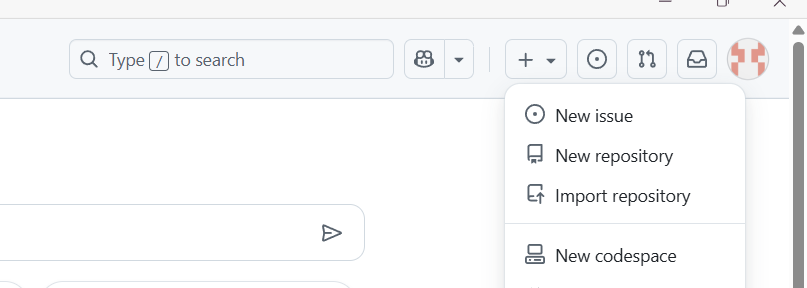
1. Install git – By default git will be available in our system

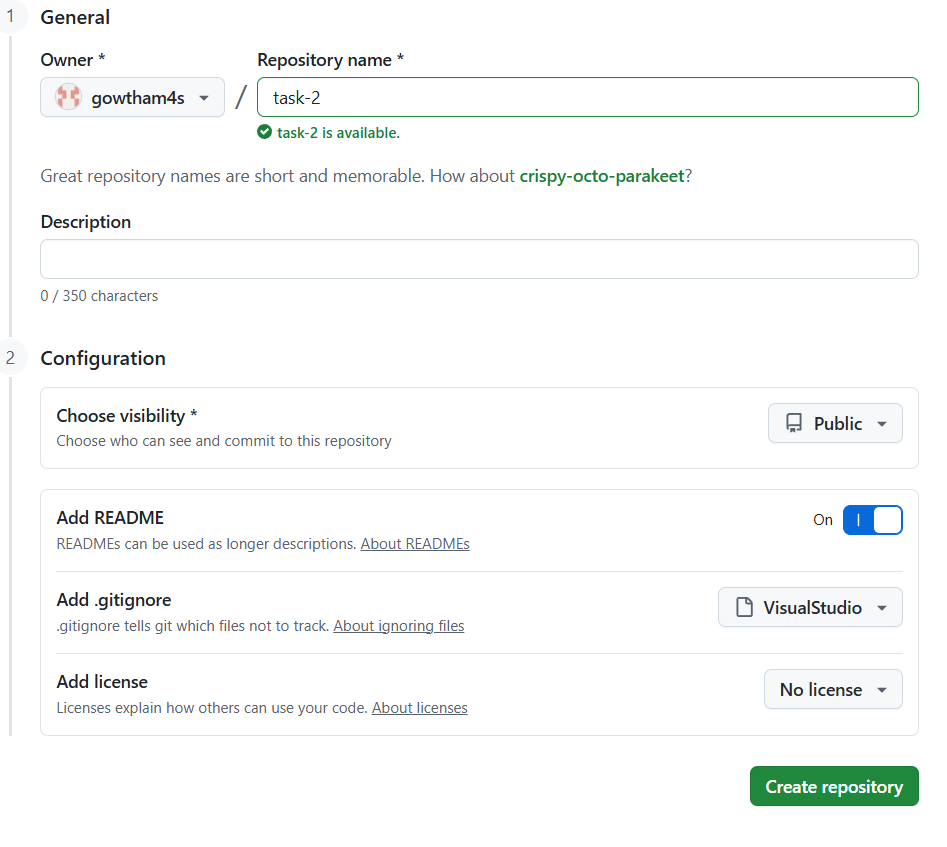


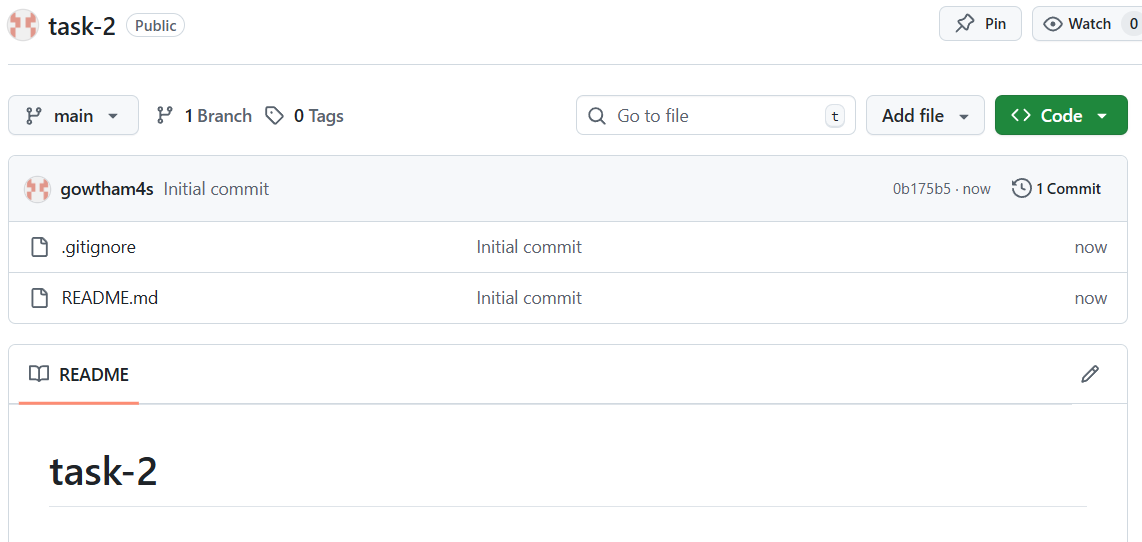
1. Create a repo in github with README.md and .ignore file.

Step 1: Select new repository



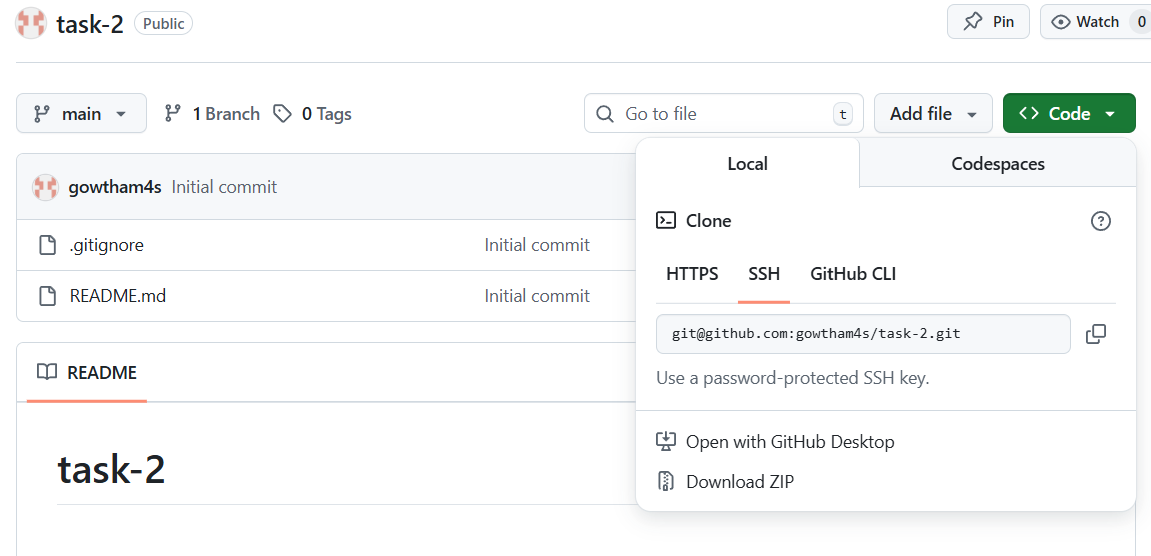
Step 2: Give name, turn Readme on, select .ignore and click create repository



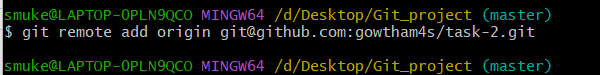


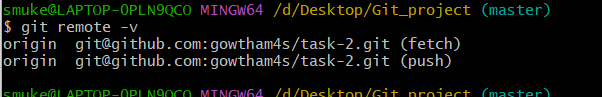
1. Clone the created repo to local.

Step 1: Copy the SSH url of repo

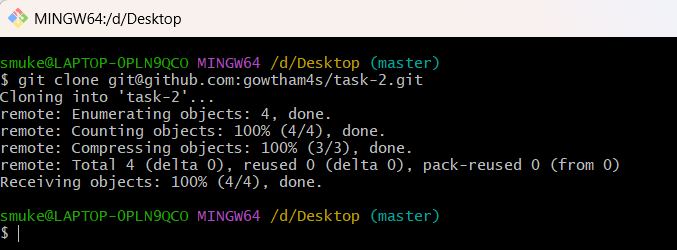


Step 2: Use **git remote add origin ssh-github-url** to add github to our local git and use **git remote -v** to check for added address



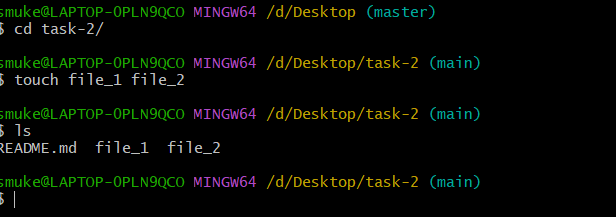


Step 3: Now use **git clone github-repo-link** to clone



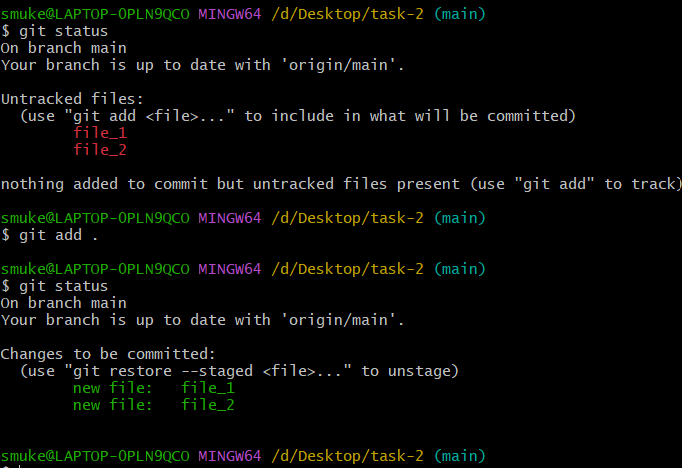
1. Create two files in local repo.

Step 1: goto repo and use **touch filename1 filename2** to create 2 files

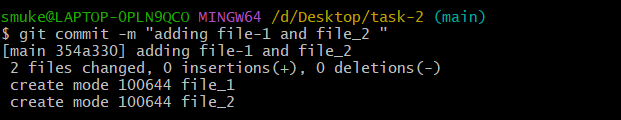


1. Commit two files and push to central Repository.

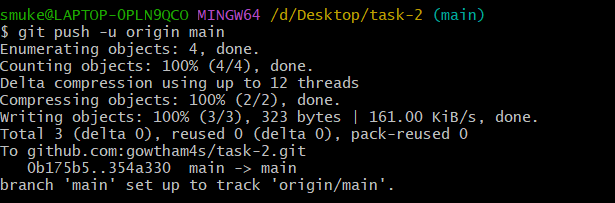
Step 1: use **git add .** to add all files from untracking area to tracking area.

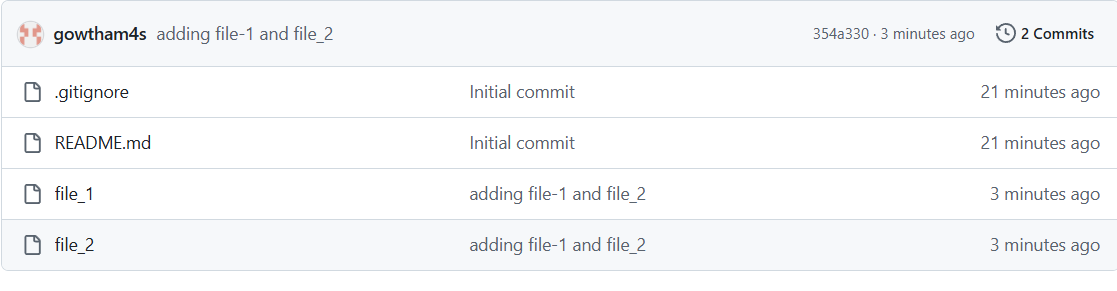


Step 2: use **git commit -m “message”** to commit all files.



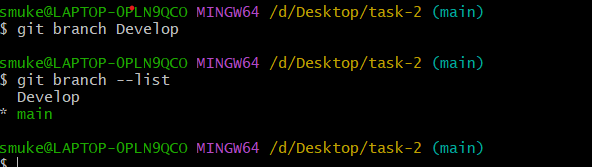
Step 3: use **git push -u origin main** to push file to GitHub.



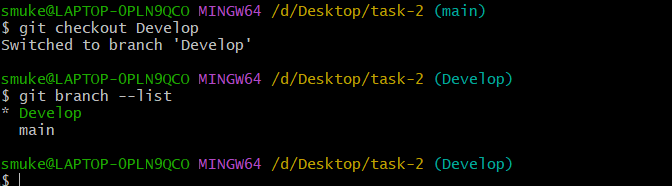


1. Create a branch in local and create a sample file and push to central.

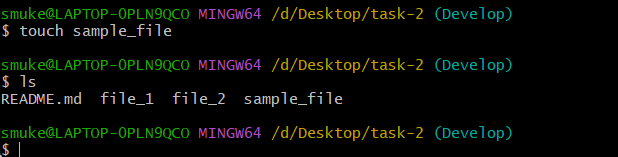
Step 1:Use **git branch branch\_name** to create a branch



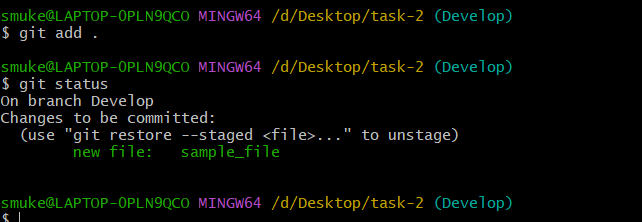
Step 2: Use **git checkout branch\_name** to switch to a branch



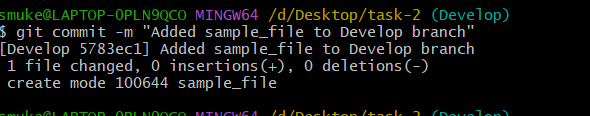
Step 3: Use **touch file\_name** to create a file



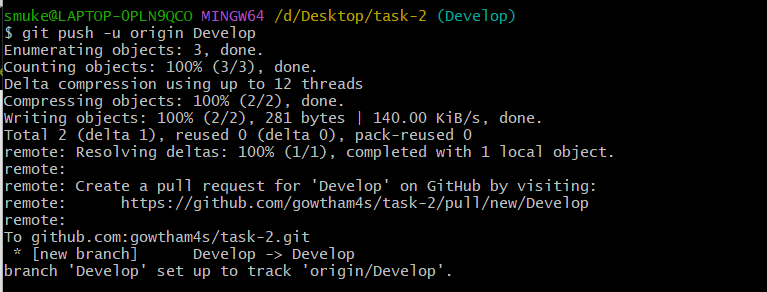
Step 4: Use **git add .** to move from untracking to tracking area



Step 5: Use **git commit -m “ commit\_message”** to commit all files

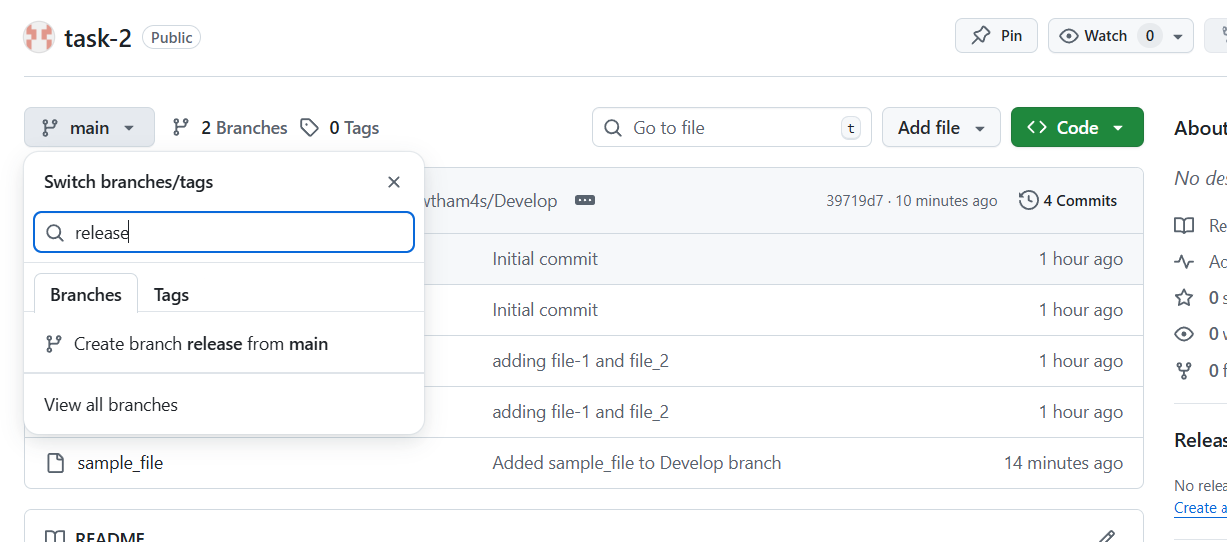


Step 6: Use **git push -u origin branch\_name** to push file to GitHub

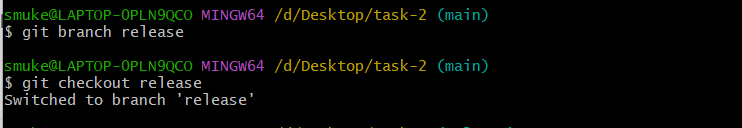


1. Create a branch in github and clone that to local.

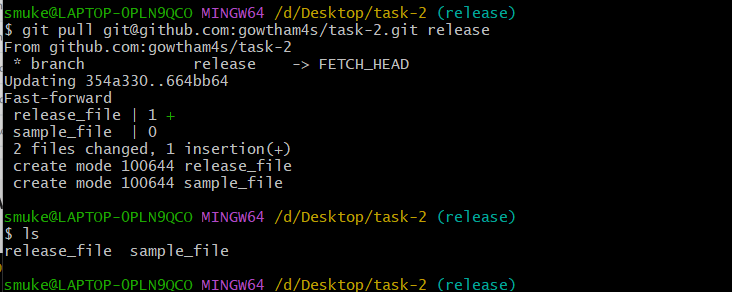
Step 1: In the code section select main branch then enter the name of the branch to be created and select create branch from main



Step 2: In git create a new branch of same name and switch to the new branch

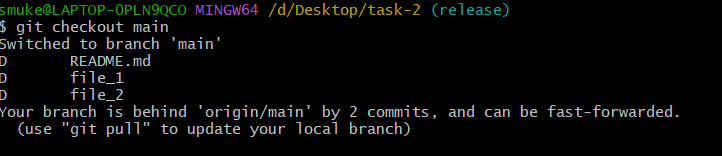


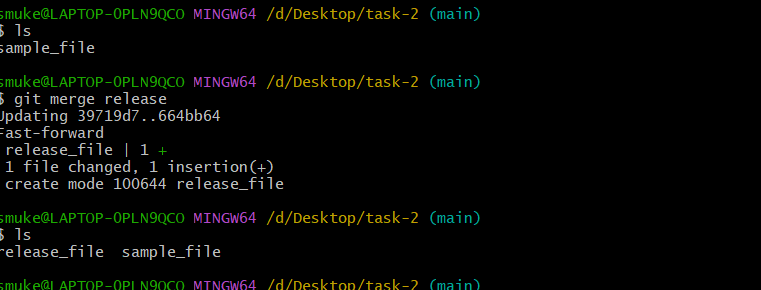
Step 3: Use **git pull github-url branch\_name** to pull files from GitHub



1. Merge the created branch with master in git local.

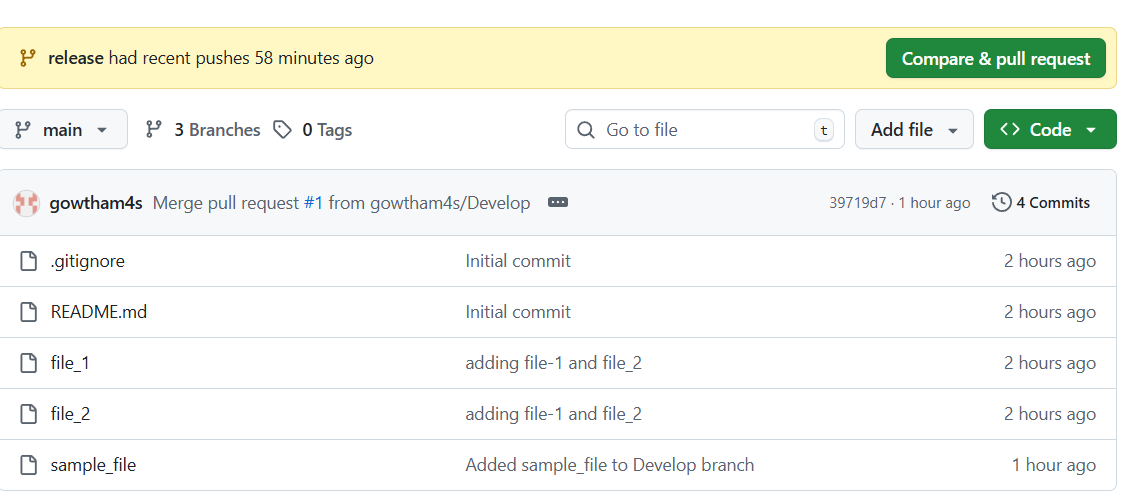
Step 1: Switch to main/master branch and give **git merge branch\_name**

****

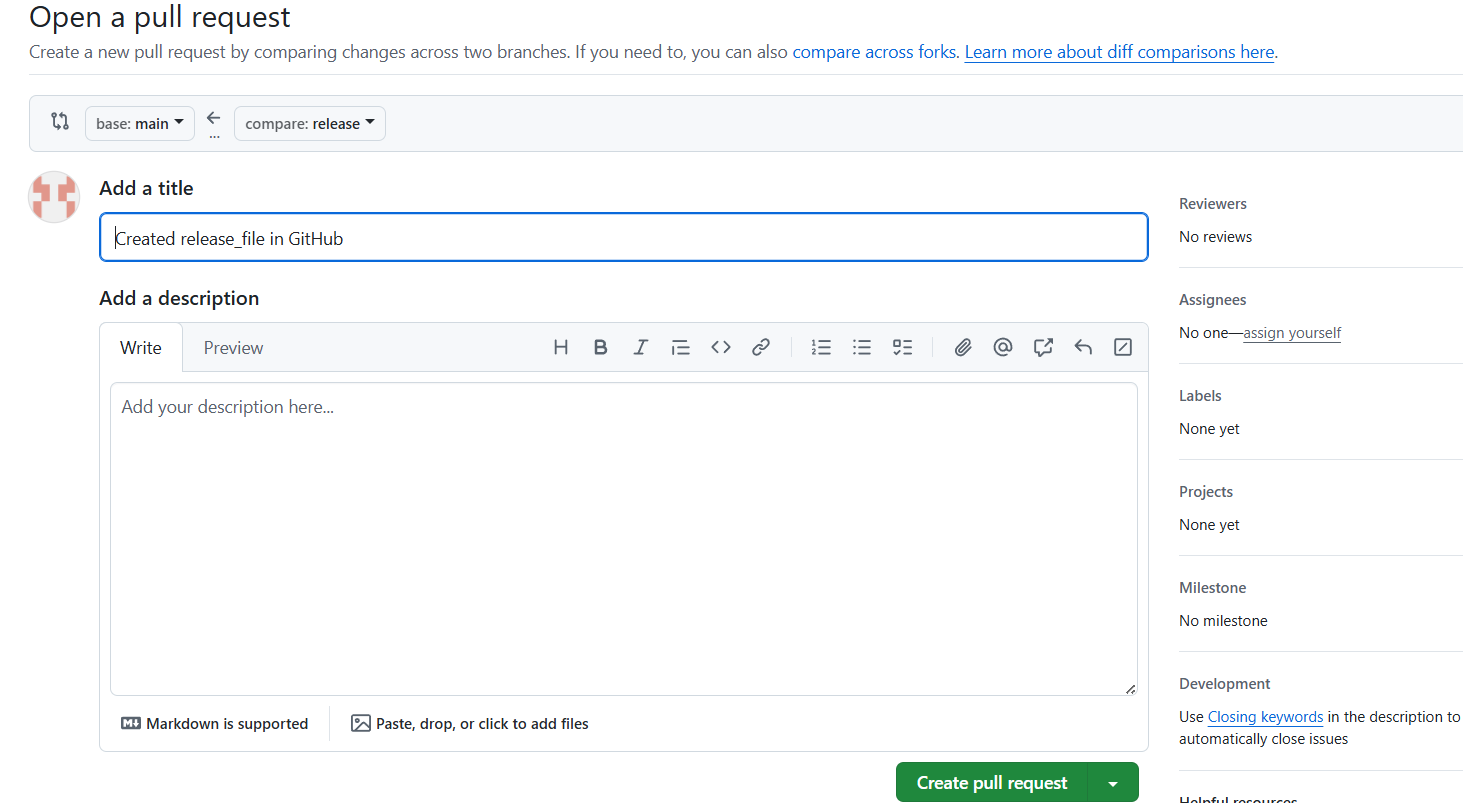
****

1. Merge the created branch with master in github by sending a pull request.

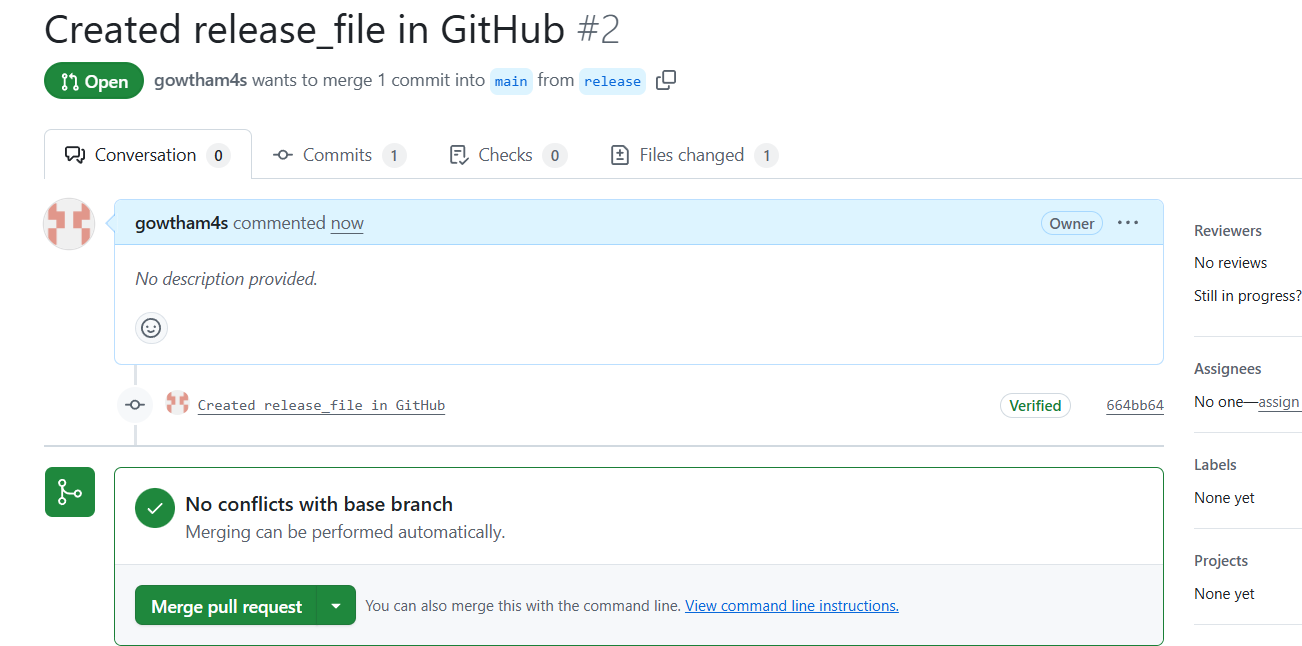
Step 1: Select **compare & pull request.**



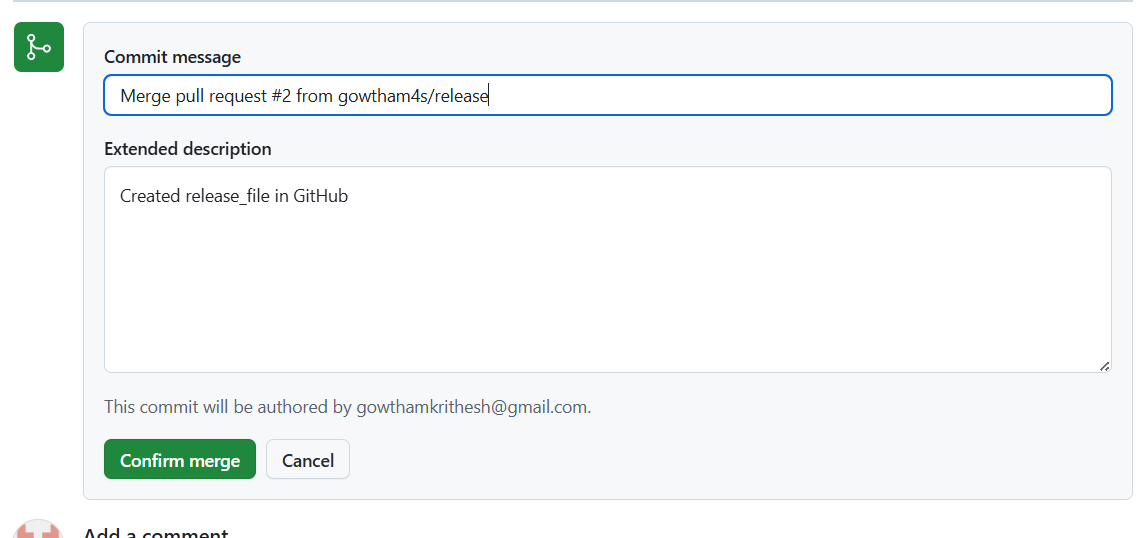
Step 2: Add title and reviewer if needed &select create pull request



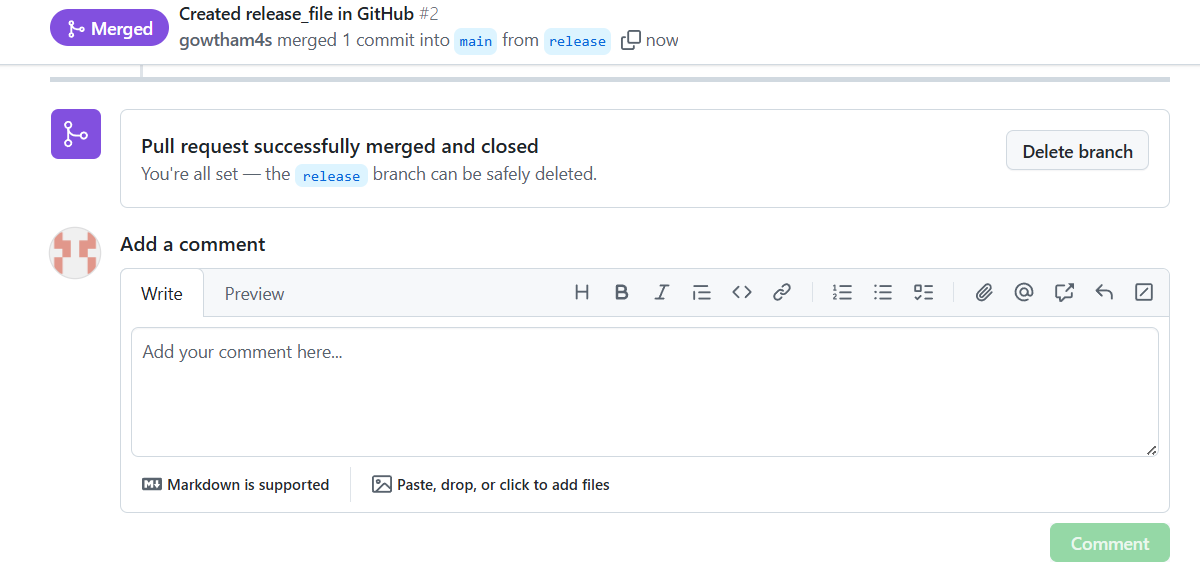
Step 3: Select **merge pull request.**



Step 4: Give commit message and select **confirm merge.**

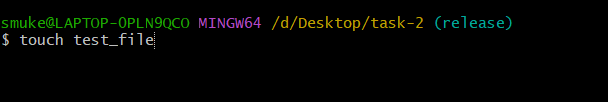


Step 5: pull request will be successfully executed and wii be merged

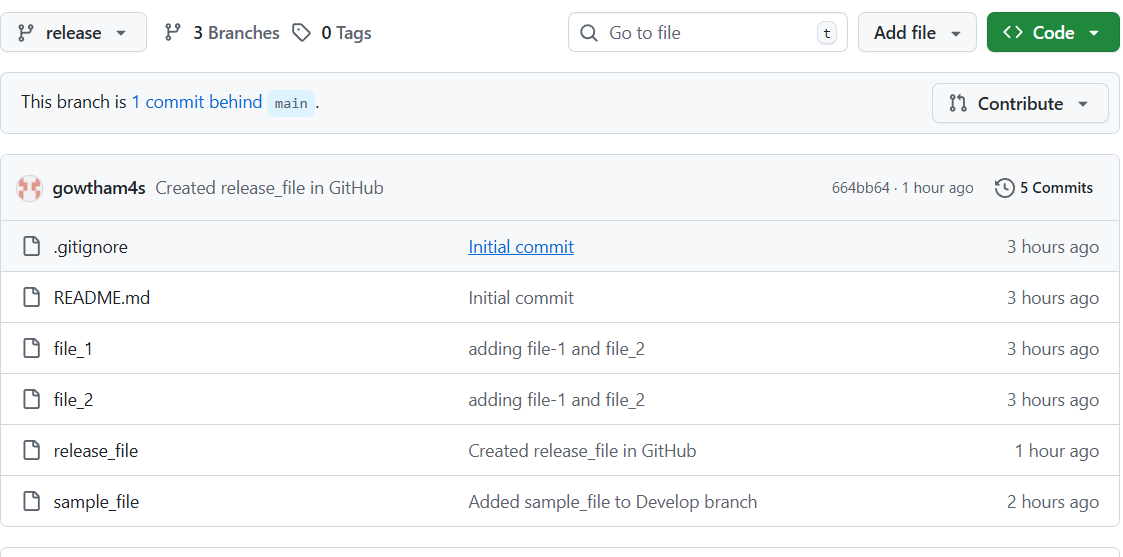


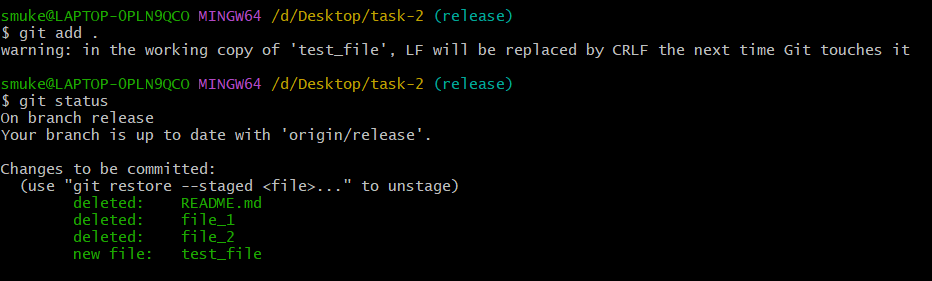
1. Create a file in local and send that to branch in github.

Step 1: Use **touch file\_name** to create a file

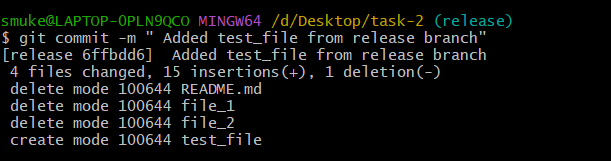


Step 2: Use **git add .** to add changes from untracking to tracking area

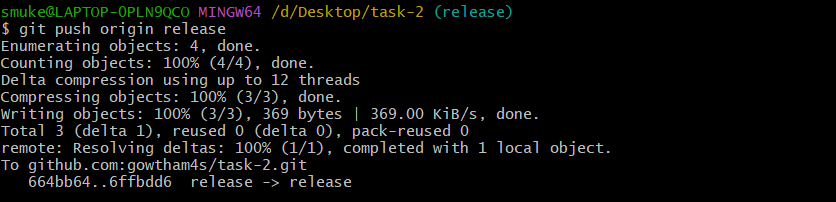


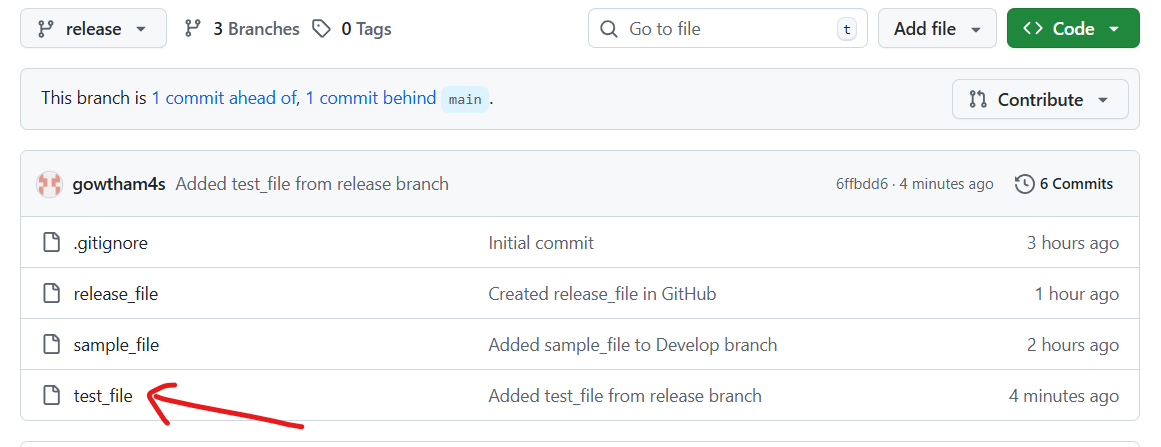


Step 3: Use **git commit -m “commit\_message”** to commit all changes



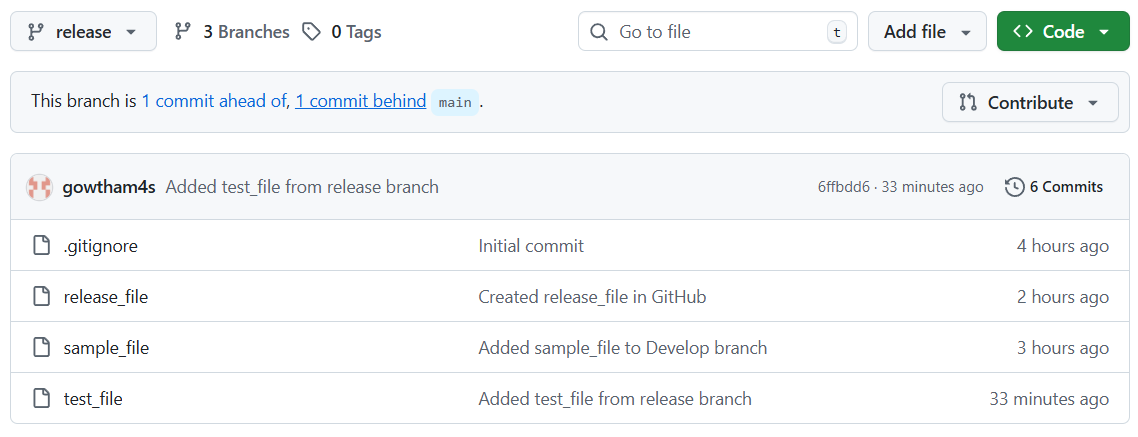
Step 4: Use **git push origin branch\_name** to push changes to the branch in GitHub.

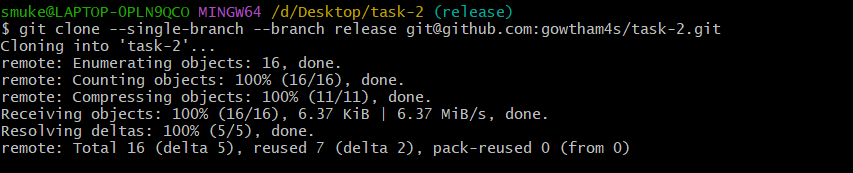


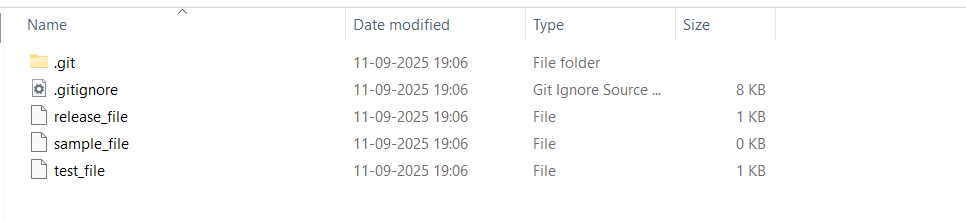


1. Clone only a branch from github to local.

Step 1: Use **git clone --single-branch --branch <branch\_name> <repository\_url>** to clone only one branch from GitHub



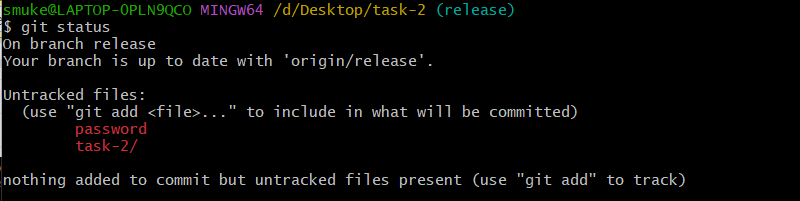




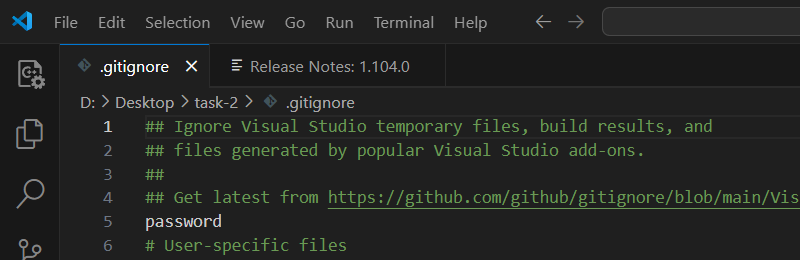
1. Create a file with all passwords and make that untrackable with git.

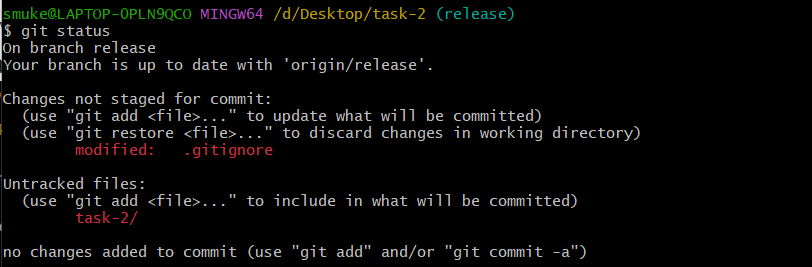
Step 1: Create a file containing password and check status





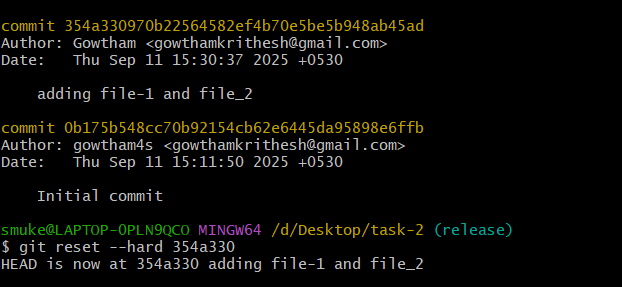
Step 2: Using visual studio create .gitignore file and add file name to be untracked and again check status





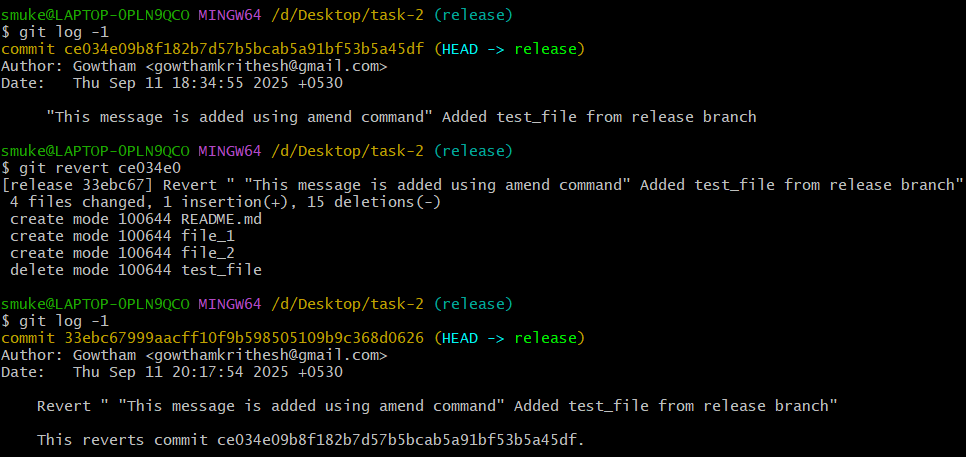
1. Make a commit and make that commit reset without savings changes.

Step 1: Use **git reset --hard commit\_id** to reset without changes

****

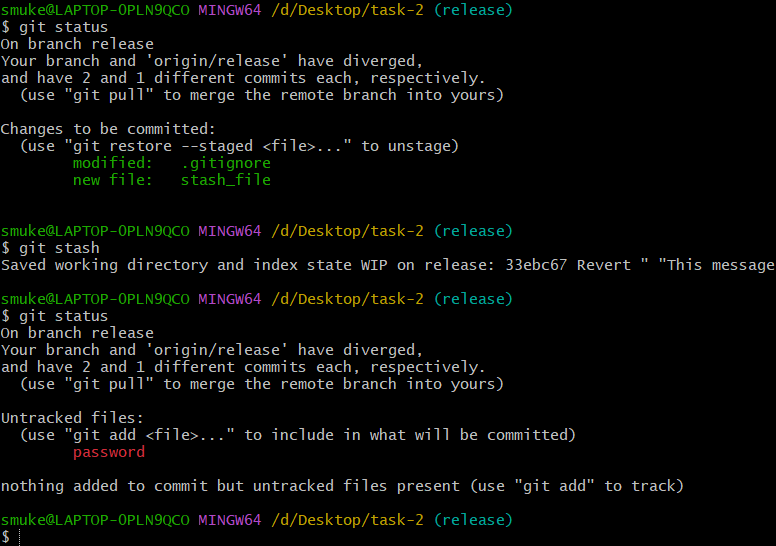
14. Revert a commited commit to the older version.

Step 1: Use **git revert <commit\_id>** to revert a commited commit to old version



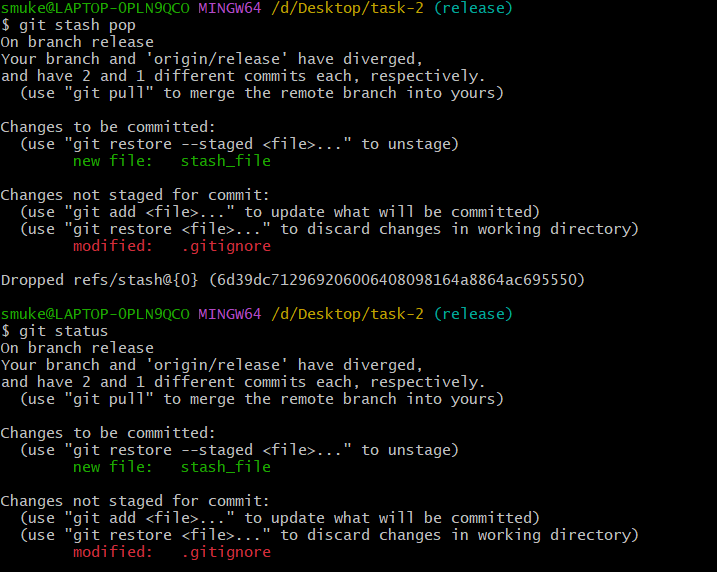
15. Push a file to stash without savings the changes and work on another file.

Step 1: Use **git stash** to push file to stash



16. Undo the stash file and start working on that again.

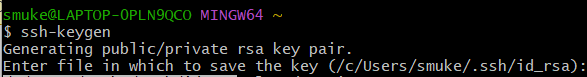
Step 1: Use **git stash pop** to undo the stash

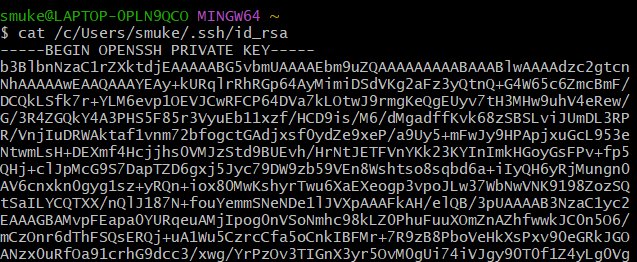


17. Generate a ssh-keygen and configure into github.

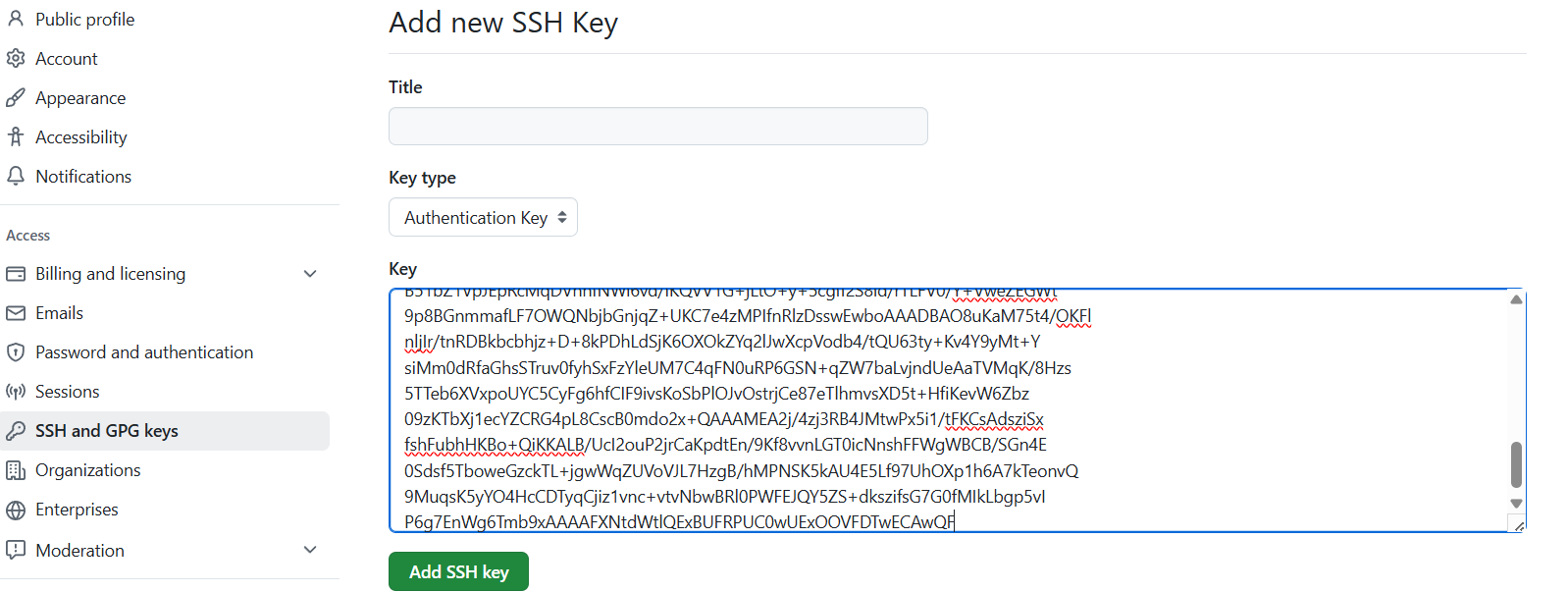
Step 1: Use **ssh-keygen**  to generate keygen and use **cat /path/to/key**

to display key



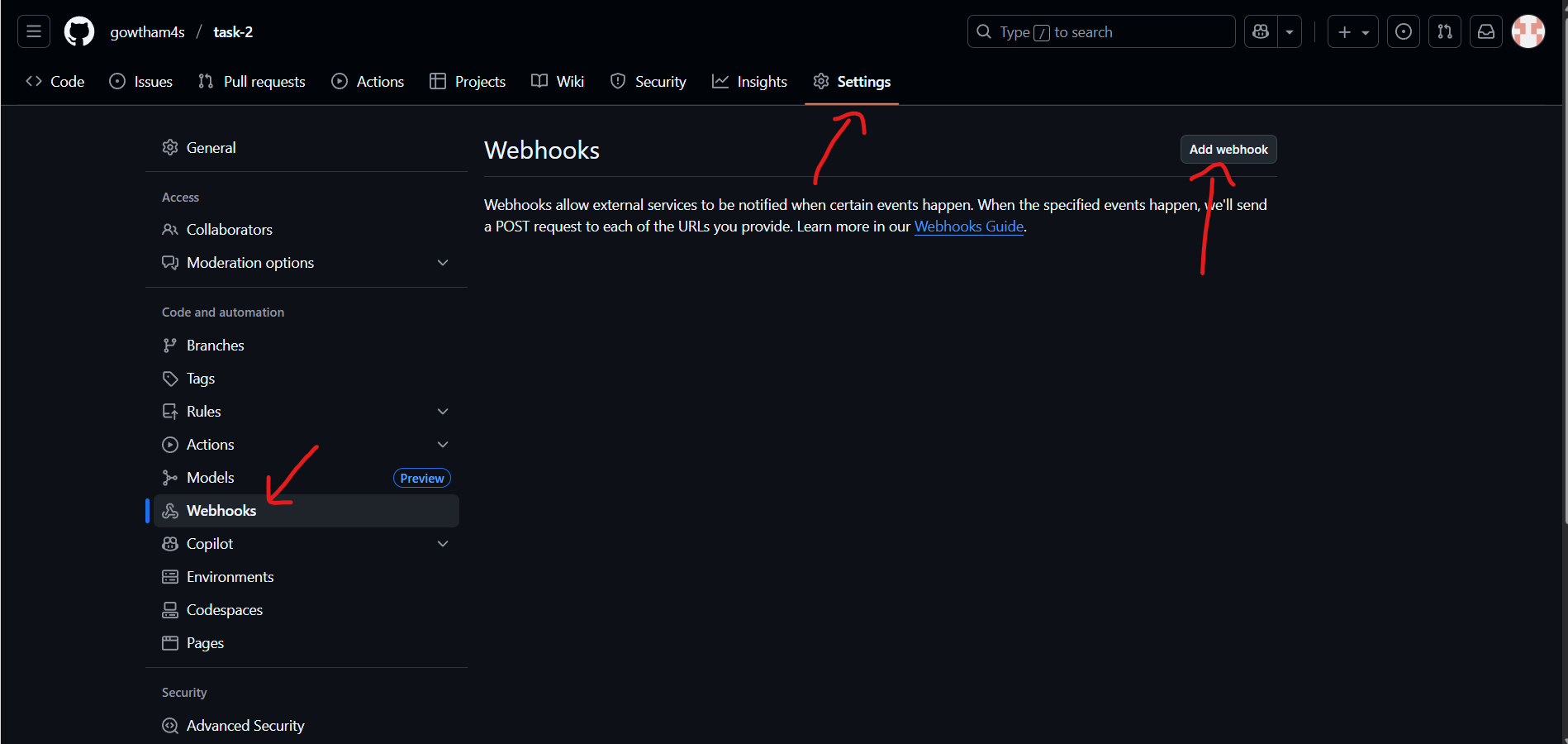


Step 2: In github goto settings 🡪SSH and GPG keys and select new SSH KEY and paste the key 🡪 Add ssh key

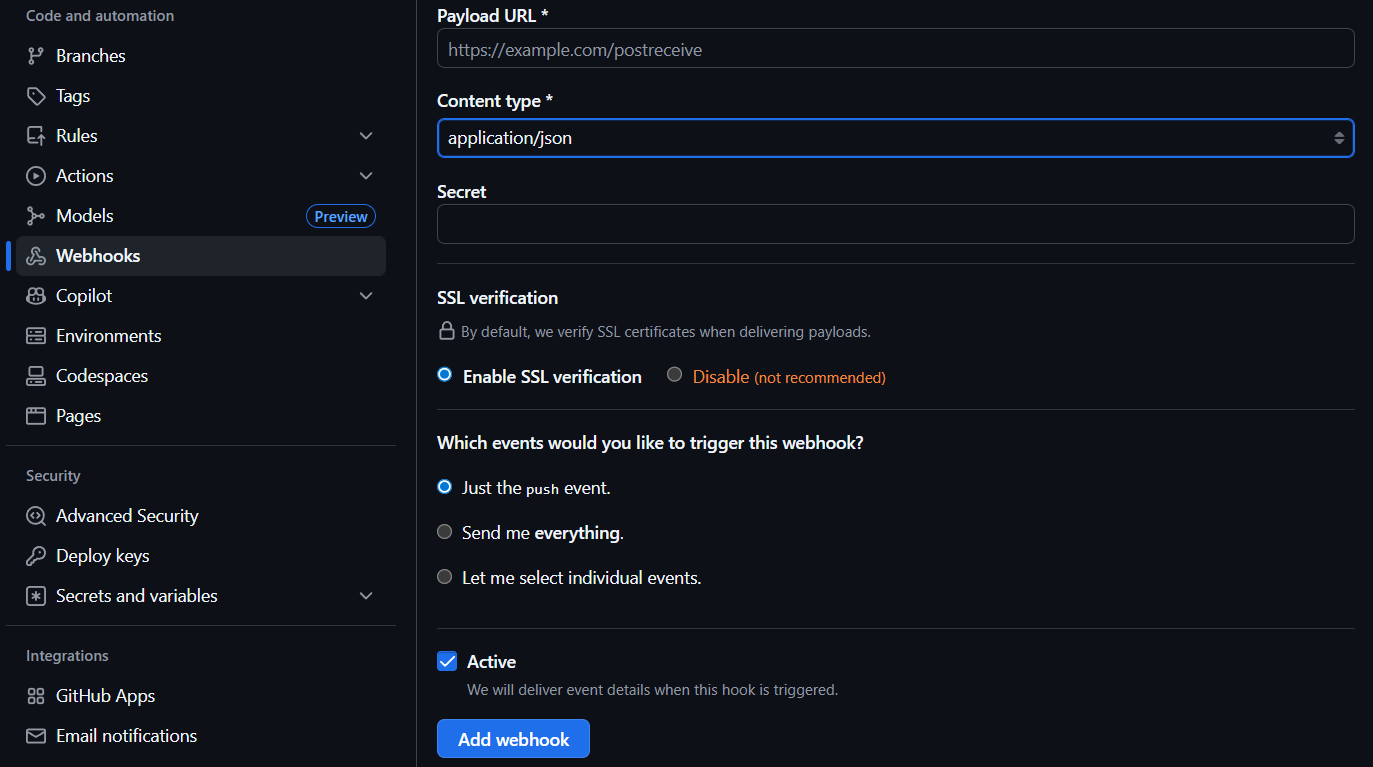


18. Configure WEBHOOKS to GitHub

Step – 1: Goto settings 🡪 webhooks and select Add webhook

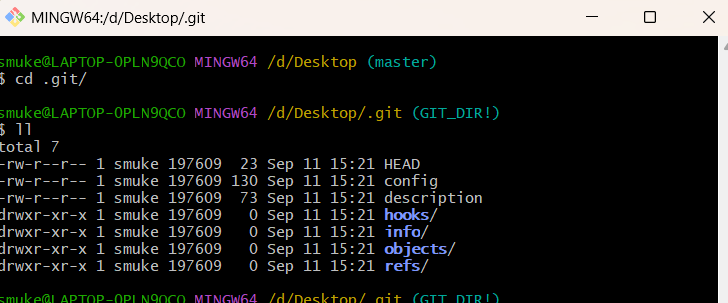


Step 2: Add the URL you need to get notified **www.url.com/git-webhook/** and configure settings and select **Add webhook.**

****

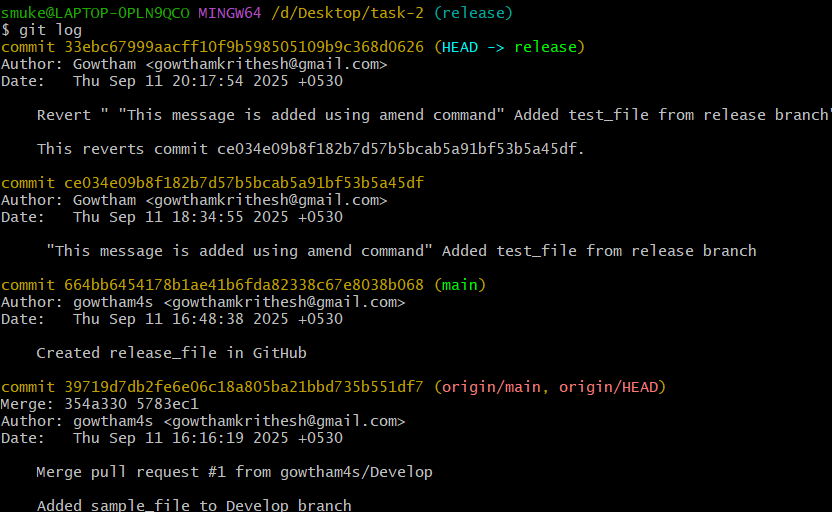
19. Basic understanding of **.git file**

When we initialize git by using **git init**  command hidden **.git/** directory will be automatically generated at the root of the project. It contains all the necessary information for Git to track changes, manage versions, and perform various operations. Moreover, it is where Git stores the entire history and metadata of our project.



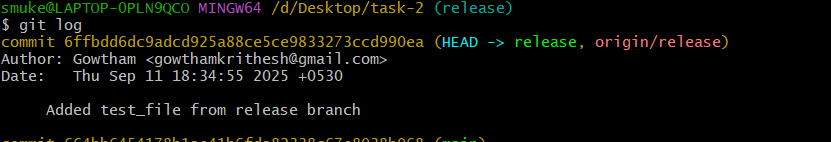
20. Check all the logs of git

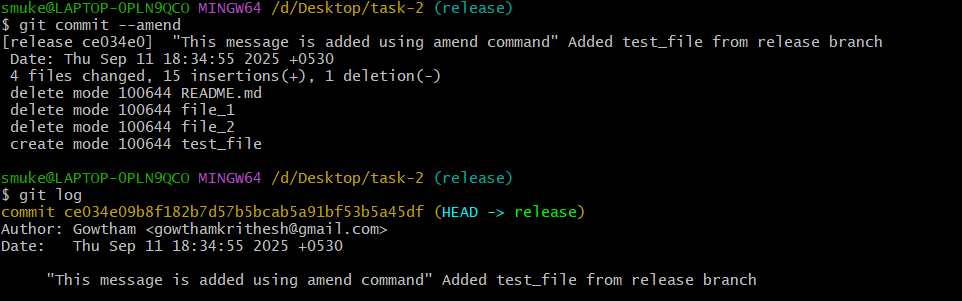
Step :Use **git log** to check all log



21. Rename the commit message

Step 1: Use **git commit –amend** to rename the commit message





22. Merge multiple commit into single commit

Step 1: Use **git rebase -I HEAD~4** and replace pick with squash

