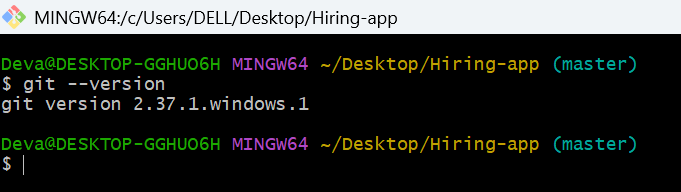
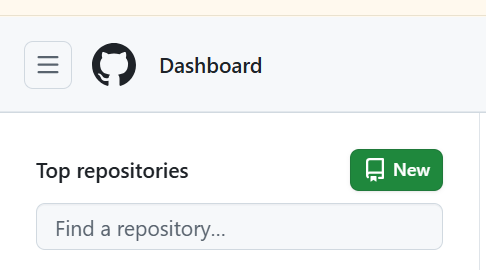
**Git&GitHub Task - 2**

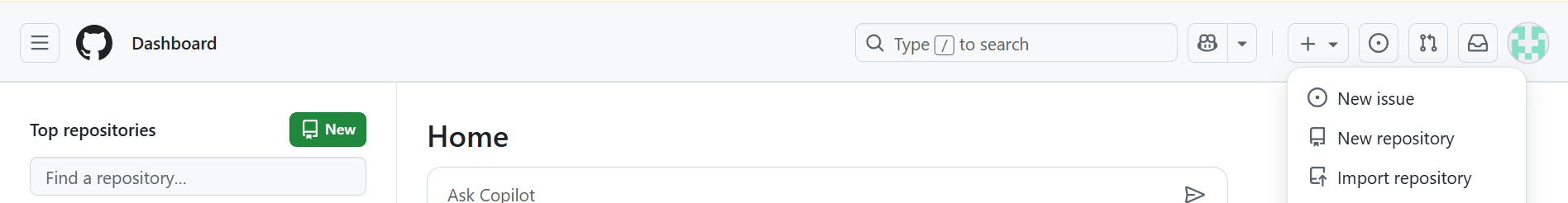
1) Install git



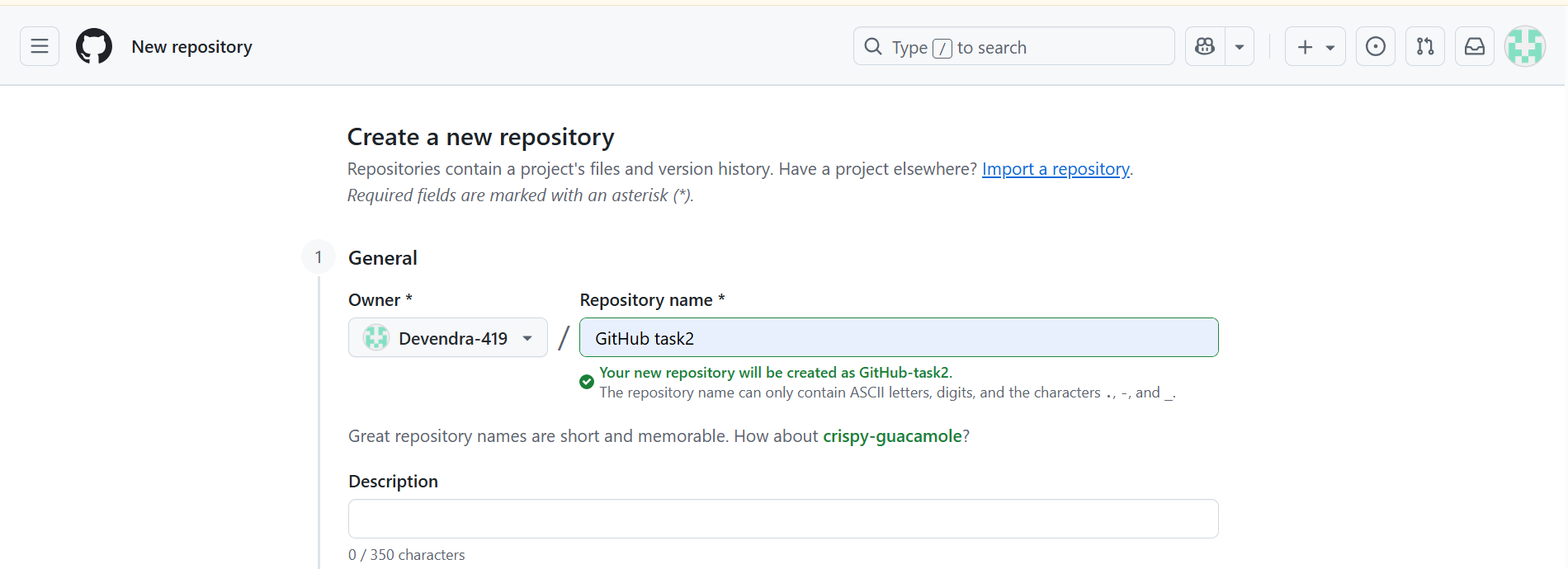
2) Create a repo in github with README.md and .ignore file.

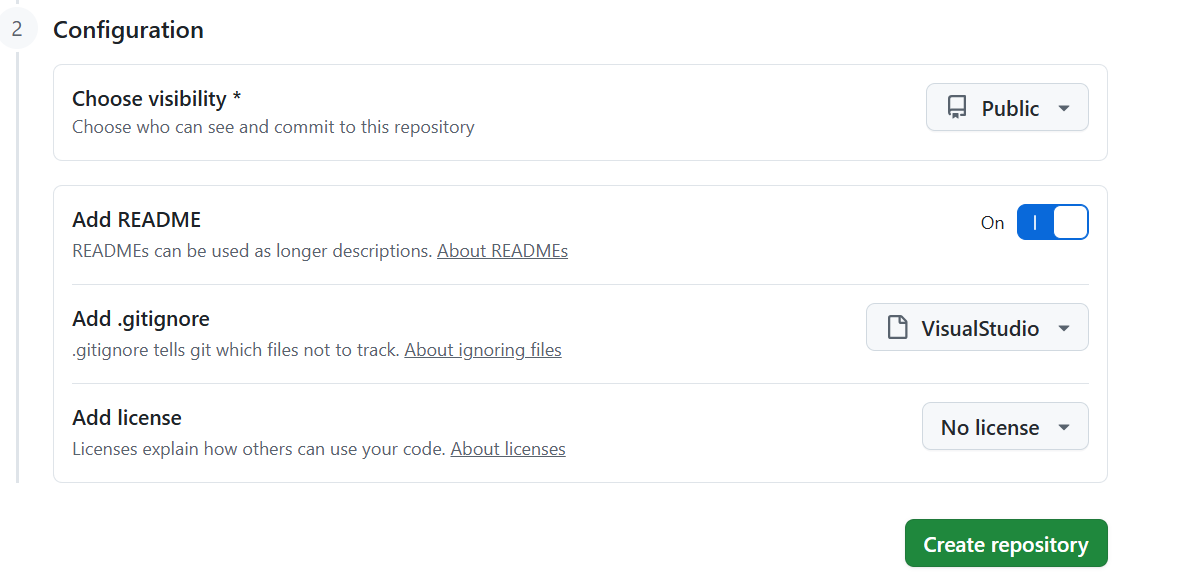
* Select new repository

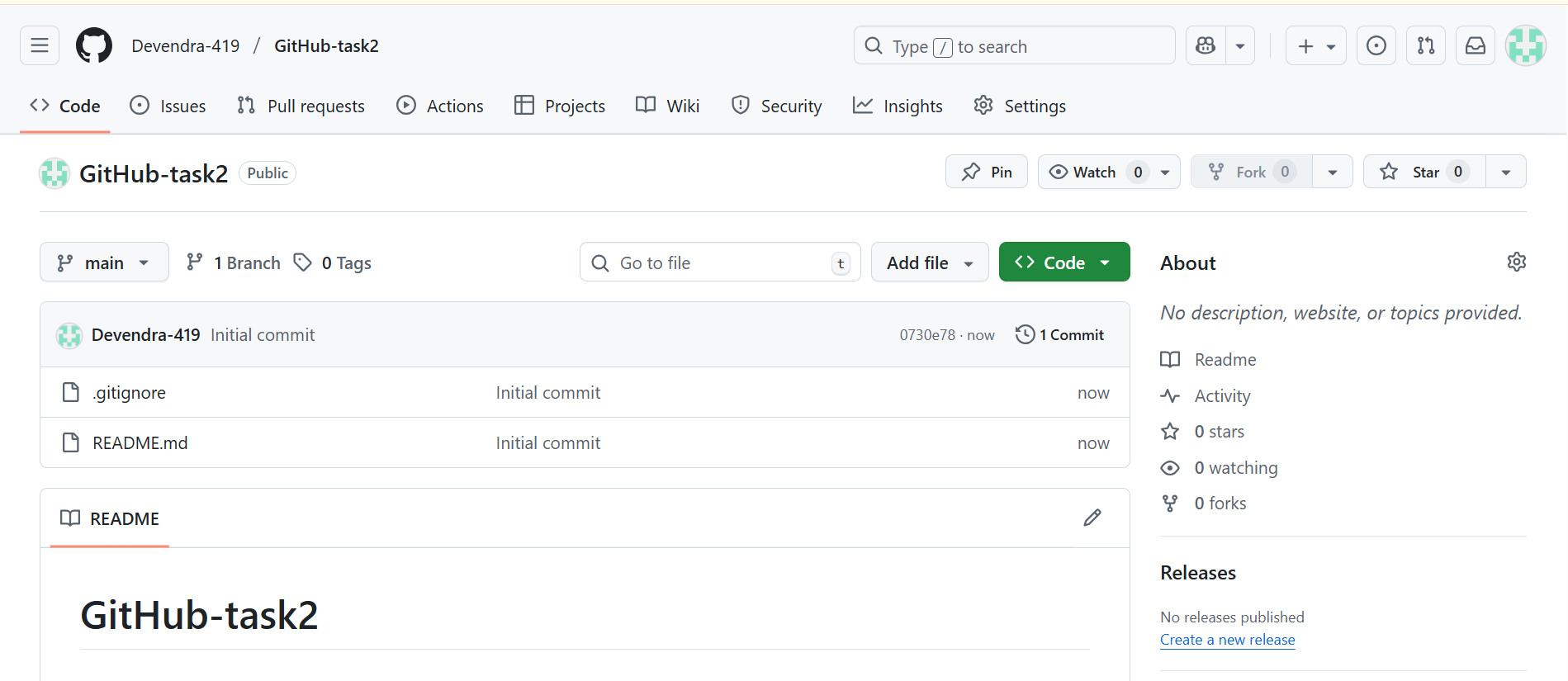




* Give name, turn Readme on select .ignore and click create repository

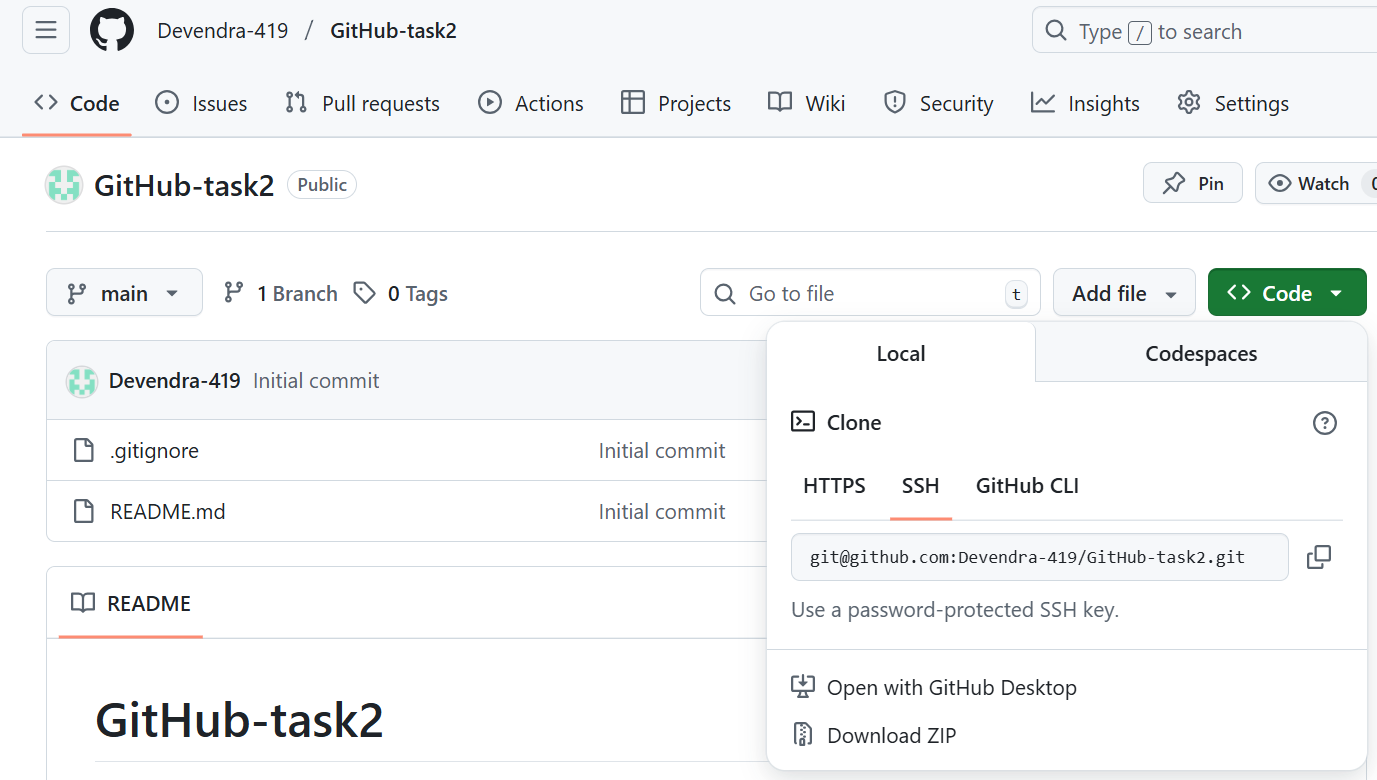




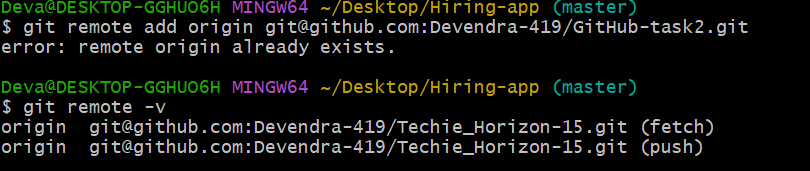


3) Clone the created repo to local

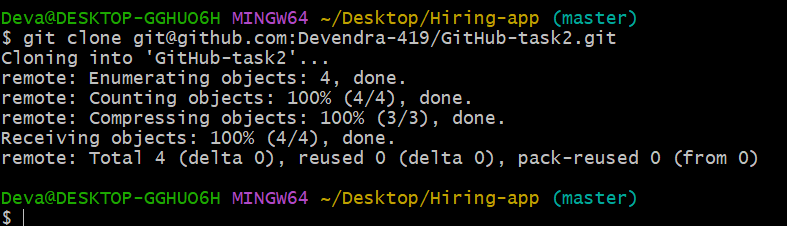
* Copy the SSH url of repo



* Use **git remote add origin ssh-github-url**  to add github to our local git
* Use **git remote –v** to check for added address

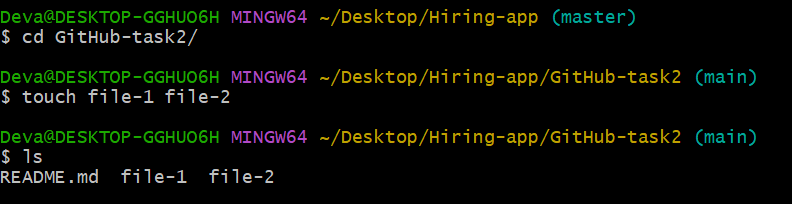


* Now use **git clone github-repo-link** to clone



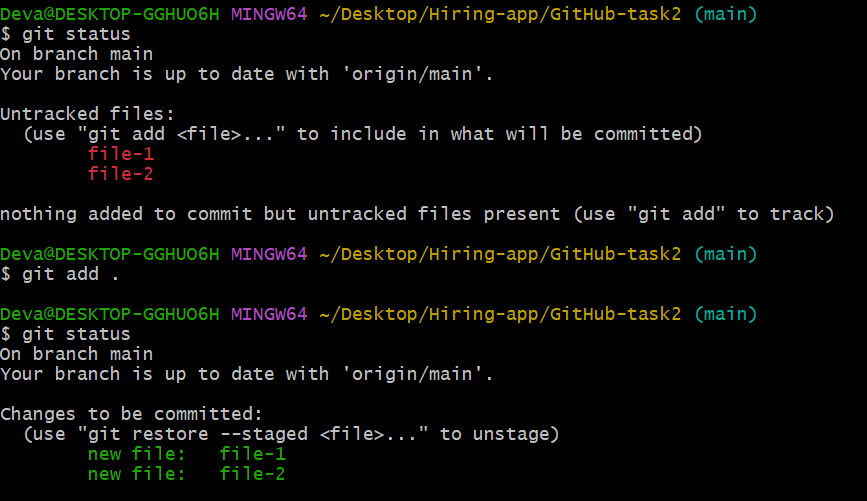
4) Create two files in local repo

* Go to repo and use touch **filename1 filename2** to create files

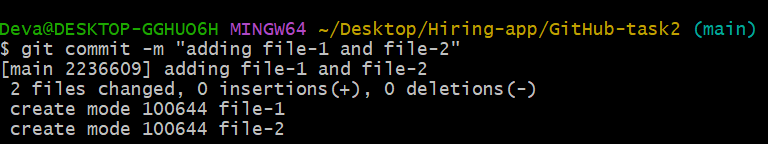


5) Commit two files and push to central Repository

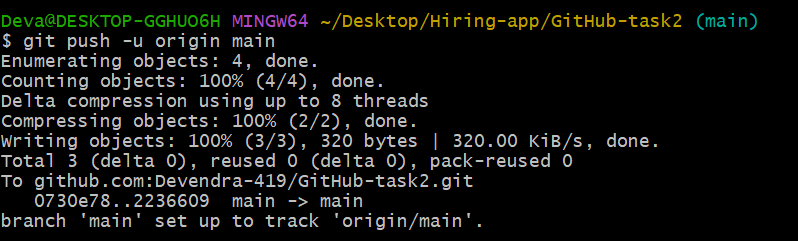
* Use **git add .** to add all files from untracking area to tracking area



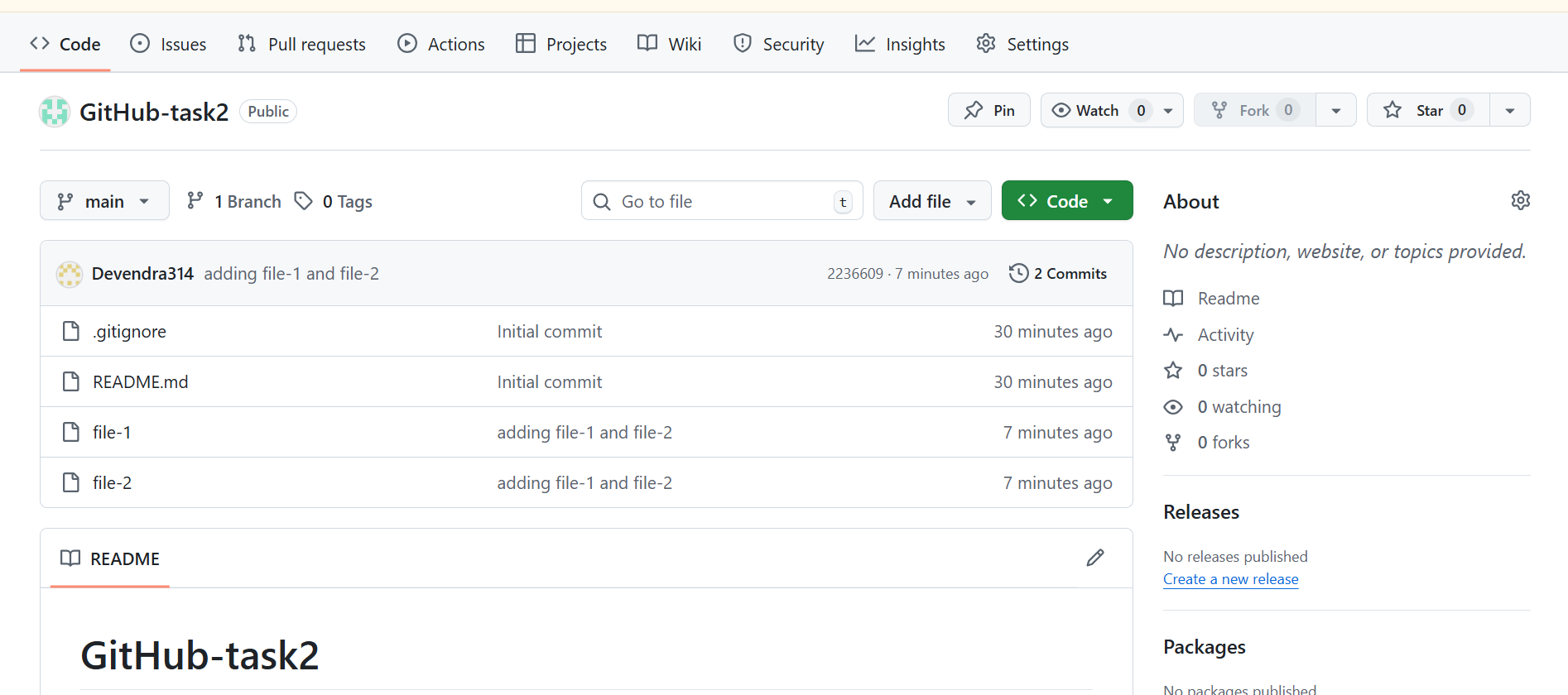
* Use **git commit –m “message”** to commit all files



* Use **git push –u origin main** to push file to GitHub

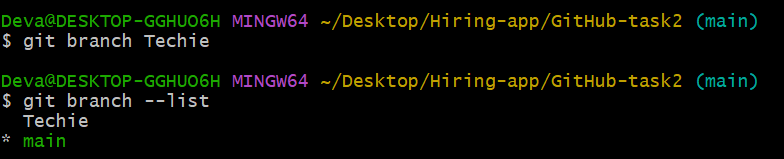


* And check in repo the files are added or not

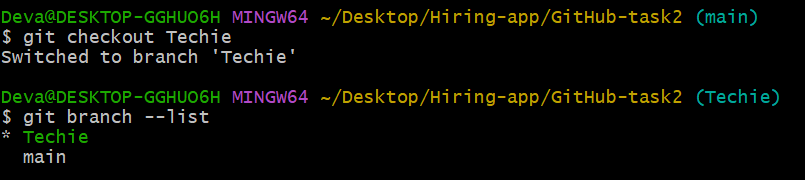


6) Create a branch in local and create a sample file and push to central

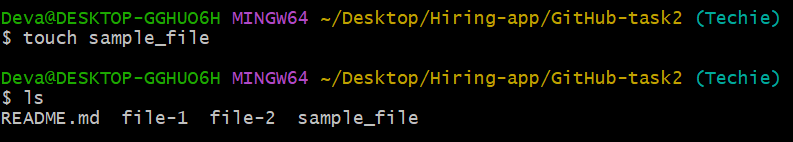
* Use **git branch branch\_name** to create a branch



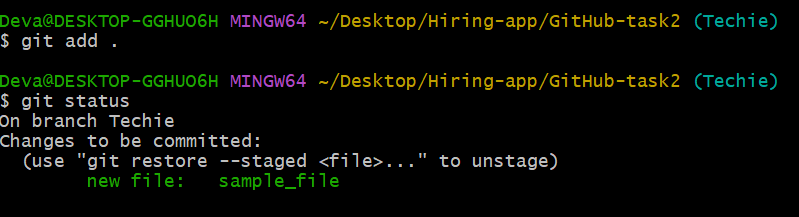
* Use **git checkout branch\_name**  to switch to a branch



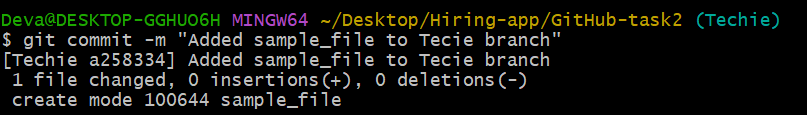
* Use **touch file\_name** to create a file



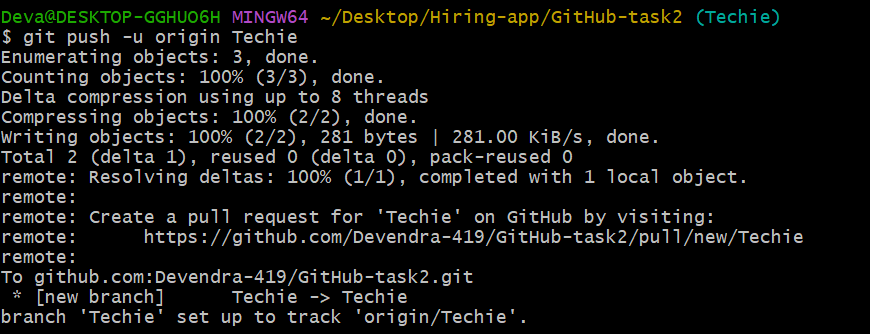
* Use **git add .** to move from untracking to tracking area



* Use **git coomit –m “commit\_message”** to commit all files

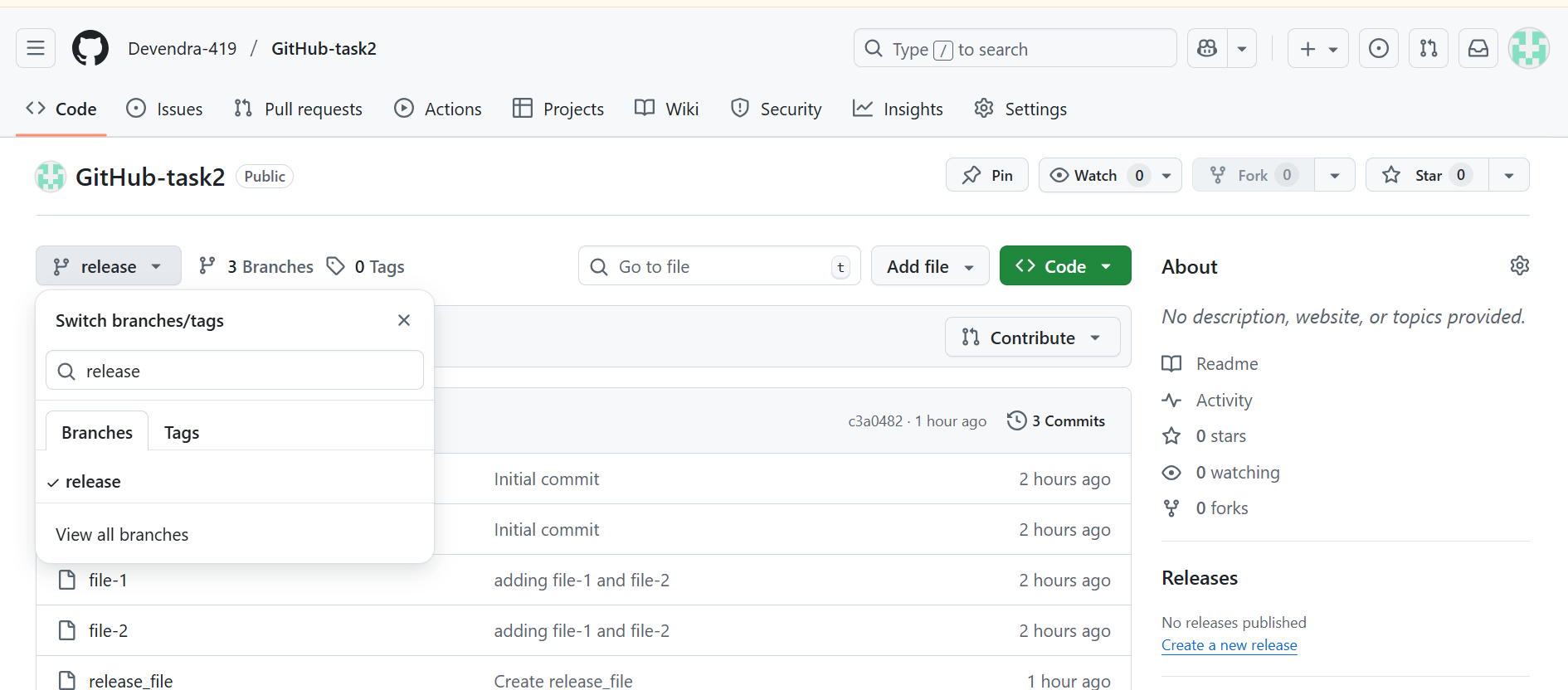


* Use **git push –u origin branch\_name** to push file to GitHub

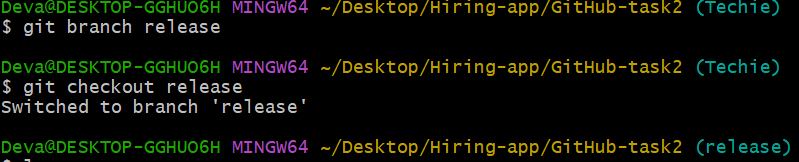


7) Create a branch in github and clone that to local

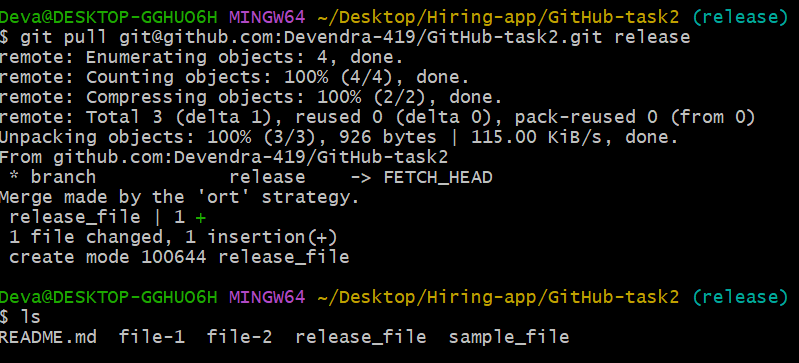
* In the code section select mail branch then enter the name of the branch to be created and select create branch from main



* In git create a new branch of same name and switch to the new branch

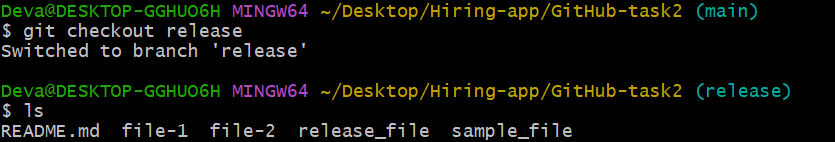


* Use **git pull=-url branch\_name** to pull files from GitHub

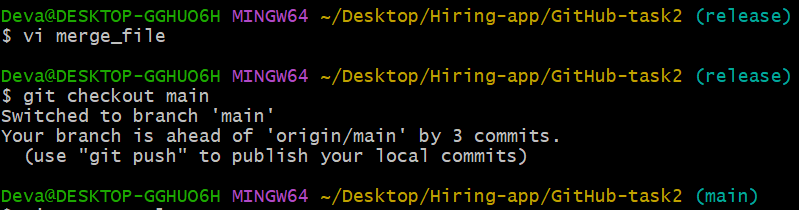


8) Merge the created branch with master in git local

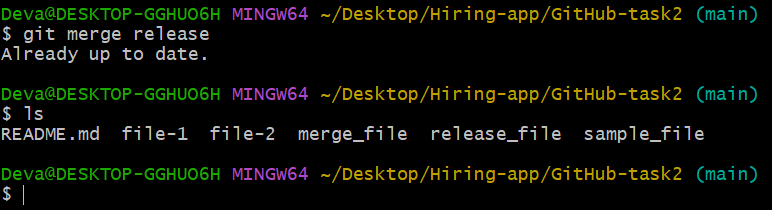
* Switch main to release branch



* Created one file in release branch
* And checkout to main branch

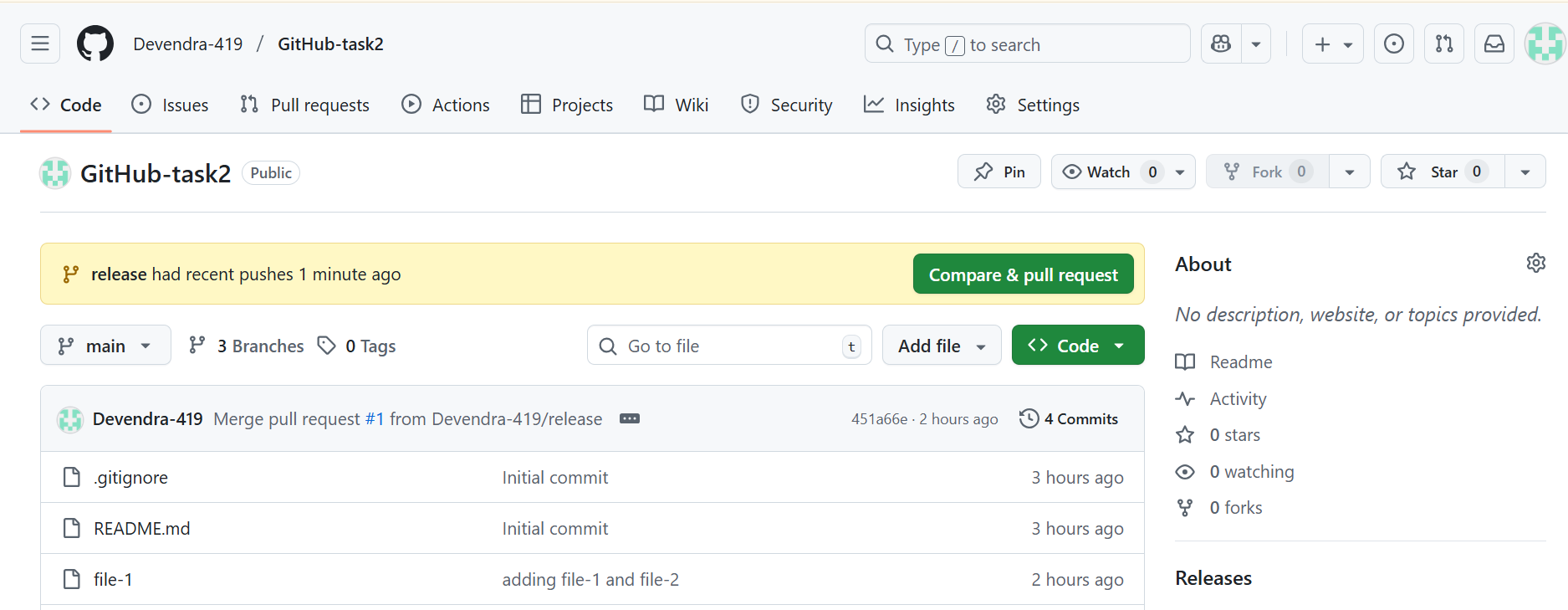


* Switch to main/master branch and give **git merge** and check the files

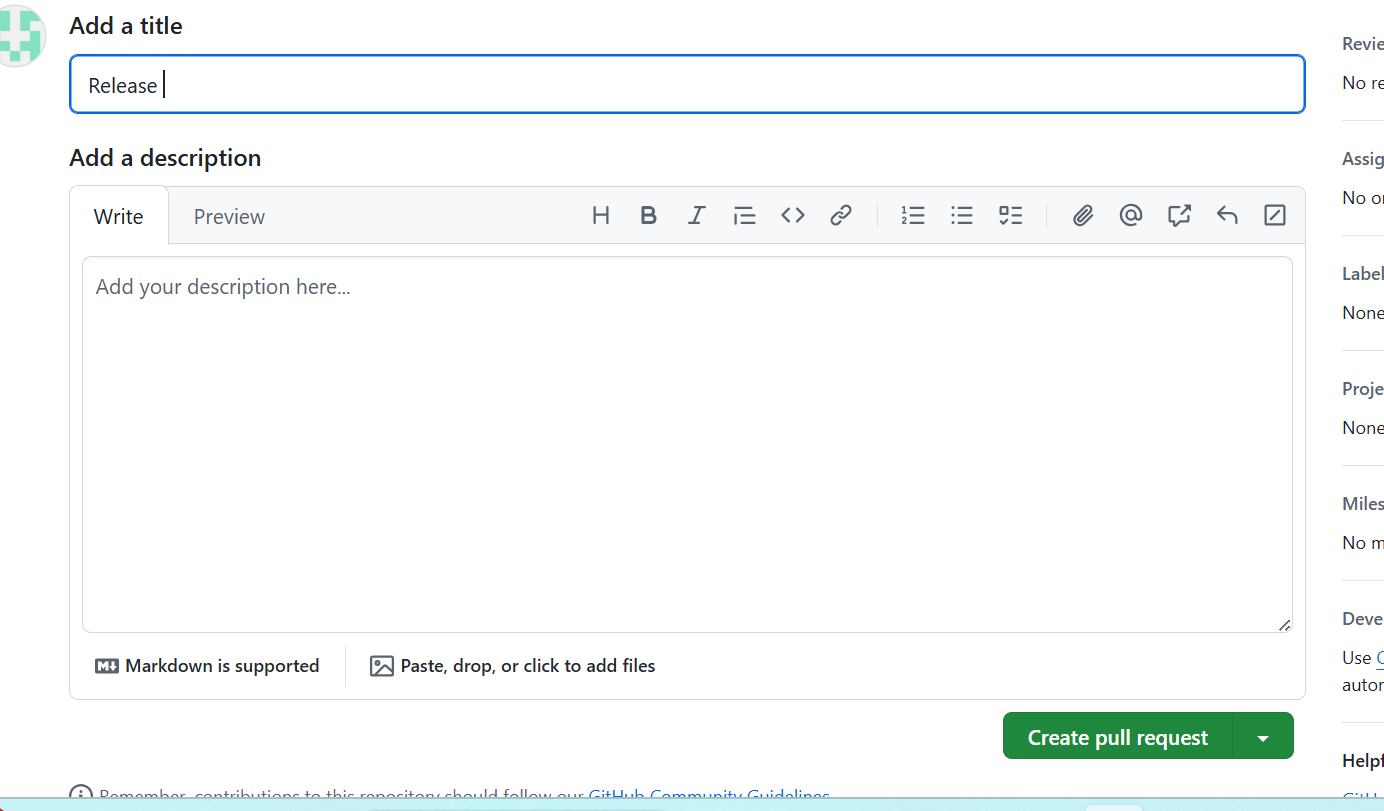


9) Merge the created branch with master in github by sending a pull request

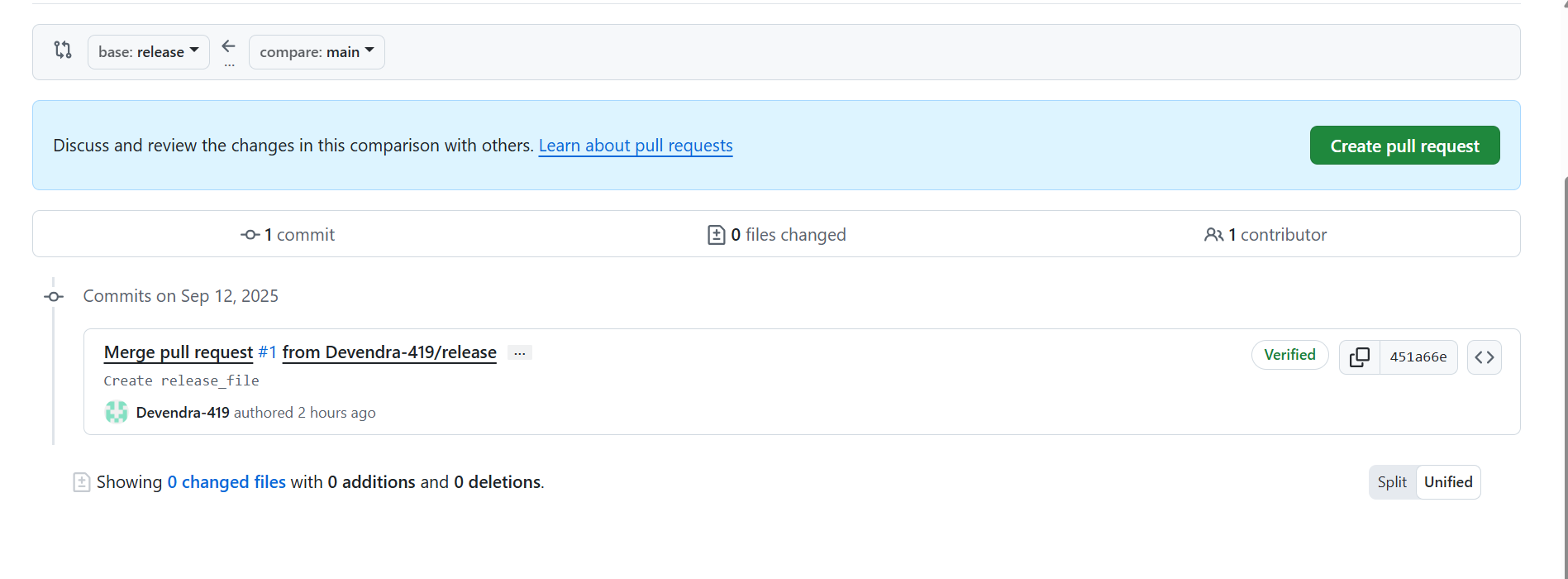
* Select **Compare & pull request**



* Add title and reviewer if needed &select create pull request

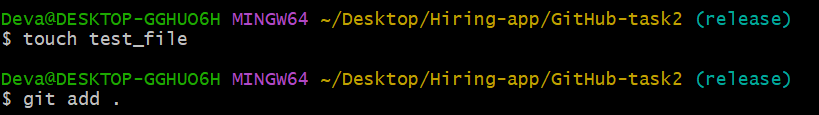


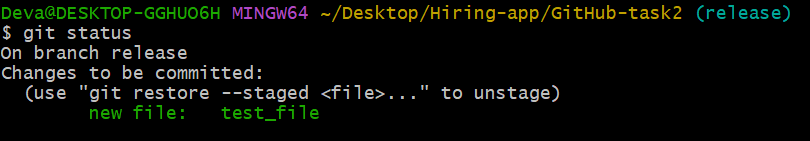
* Pull request will be successfully executed and will be merged



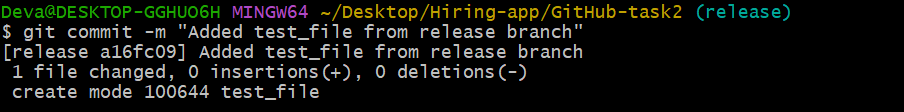
10) Create a file in local and send that to branch in github

* Use **touch file\_name** to create a file
* Use **git add .** to add changes from untracking to tracking area

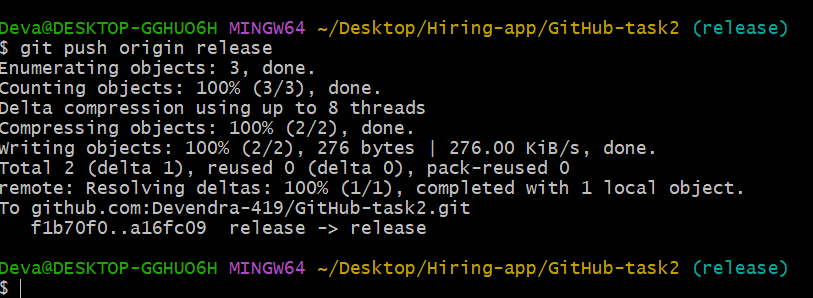
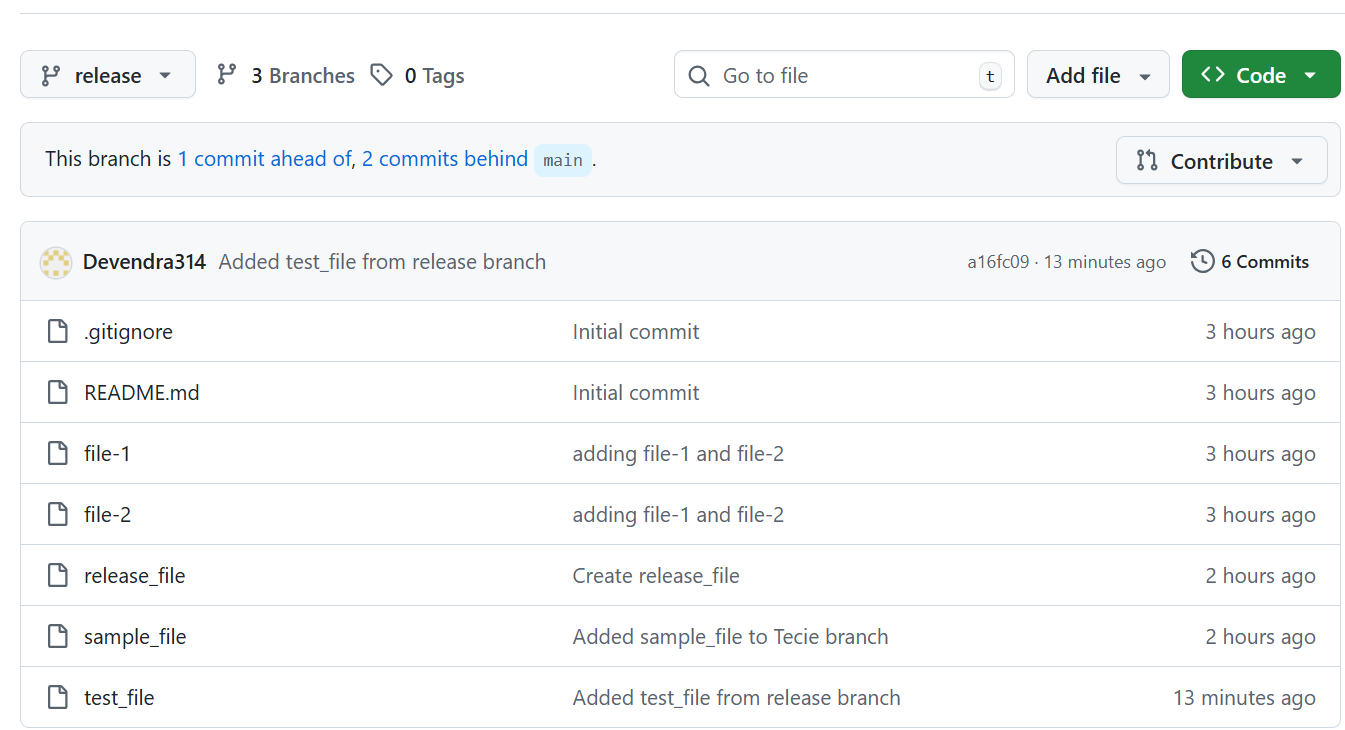




* Use **git commit –m “commit\_message”** to commit all changes

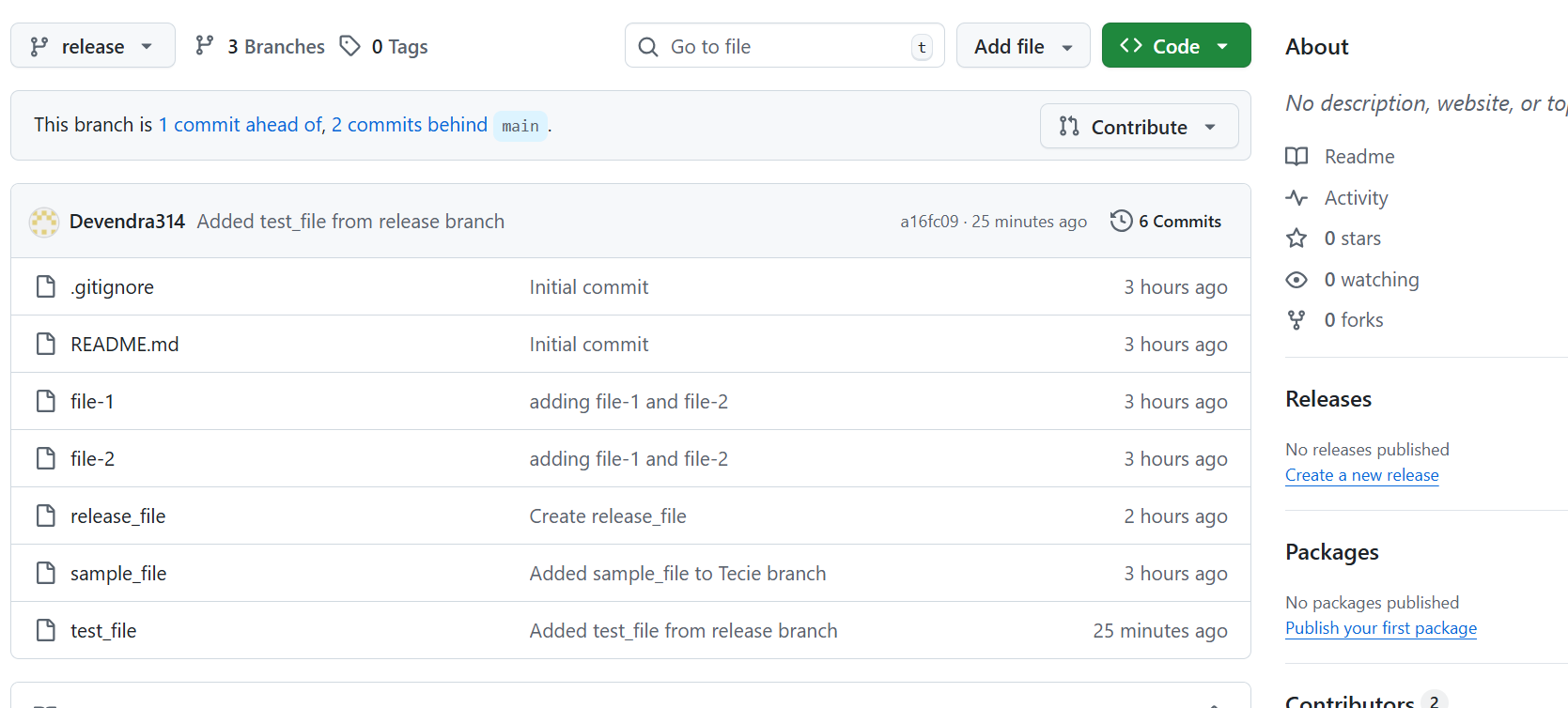


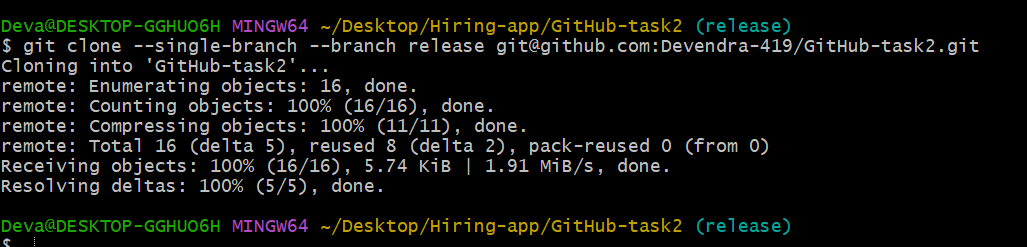
* Use **git push origin branch\_name** to push changes to the branch in GitHub

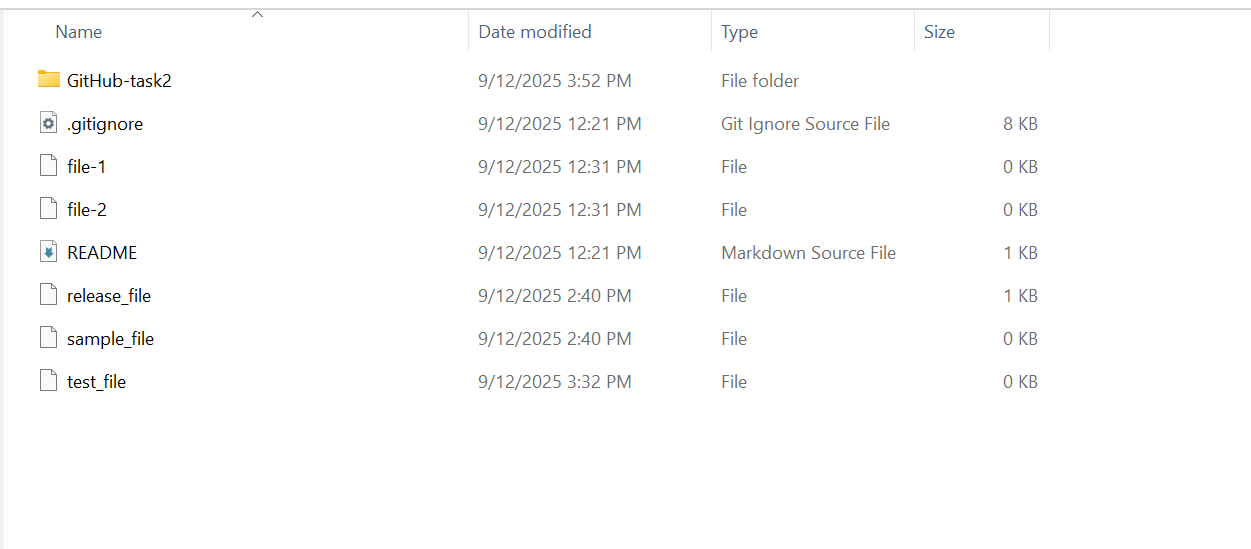
 

11) Clone only a branch from github to local

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





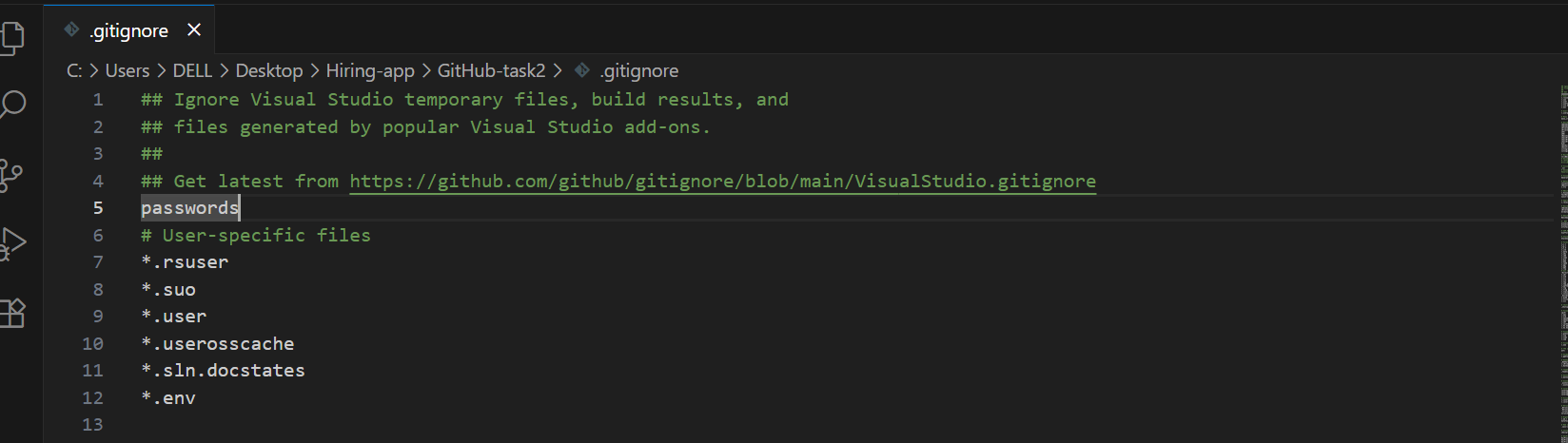


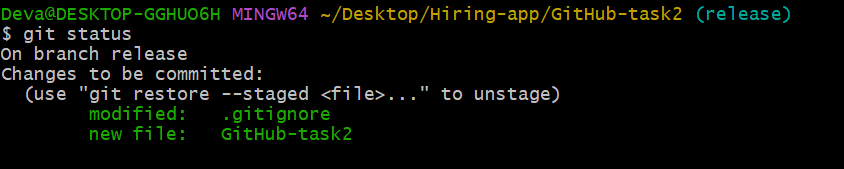
12) Create a file with all passwords and make that untrackable with git

* Create a file containing password and check status



* Using visual studio create. Gitignore file and add file name to be untracked and again check status





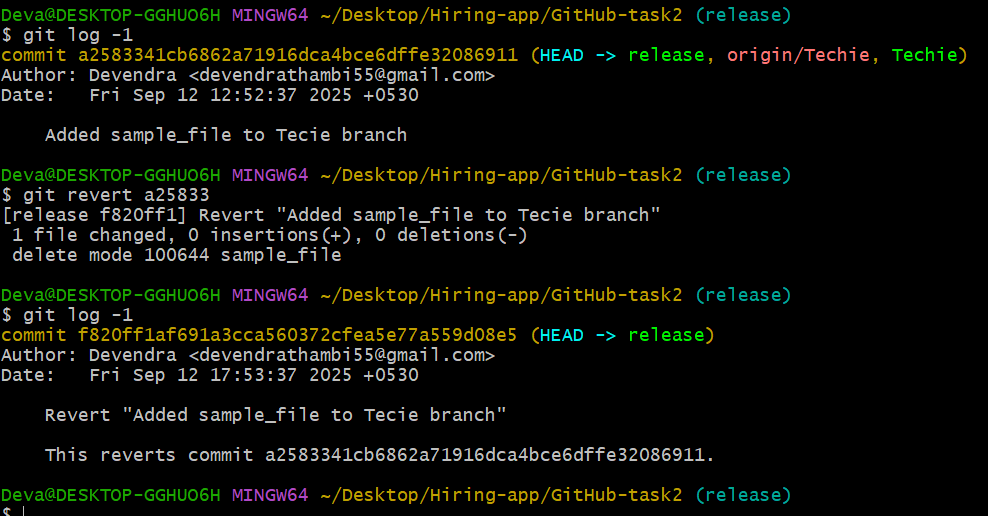
13) Make a commit and make that commit reset without savings changes

* Use git reset –hard HEAD~1 - to reset without changes



14) Revert a commited commit to the older version

* Use **git revert <commit\_id>** to revert a commited commit to old version.



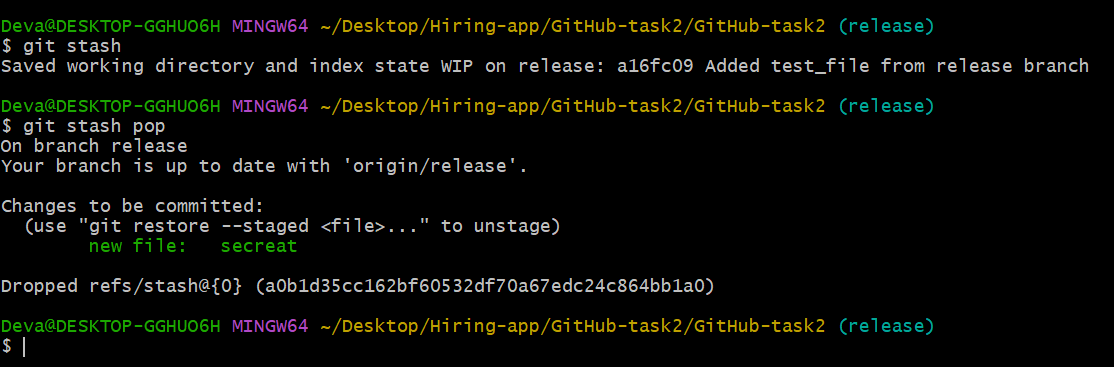
15) Push a file to stash without savings the changes and work on another file

* Create a new file **secreat**
* **Git add .**
* **Git stash –**Move the file to the stash memory temperory

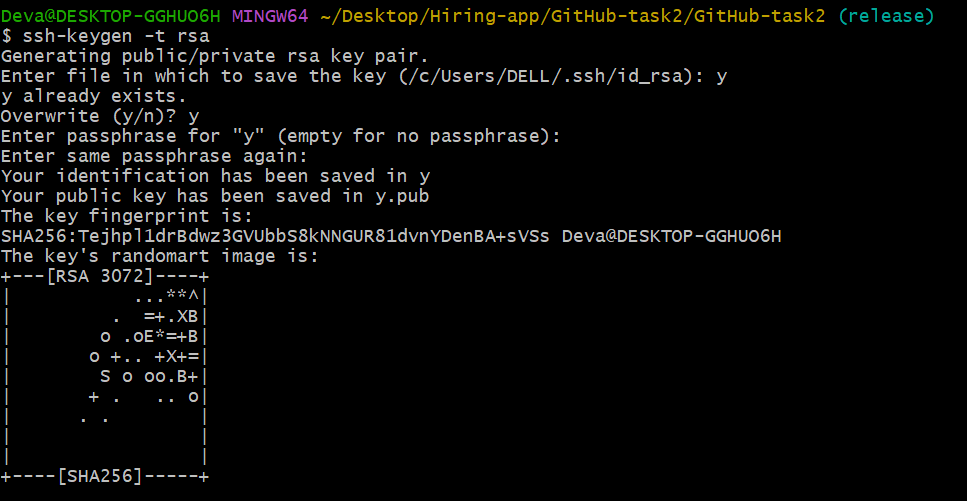


16) Undo the stash file and start working on that again

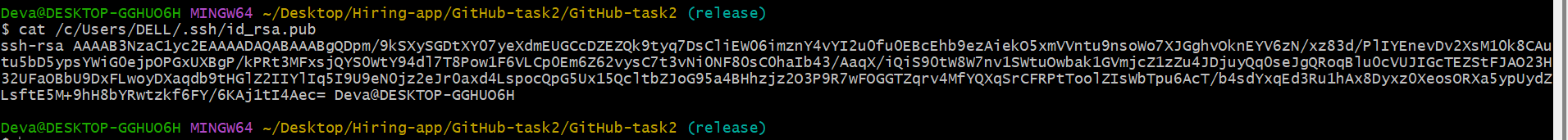
* **git stash pop** : To undo the stash file and work on that



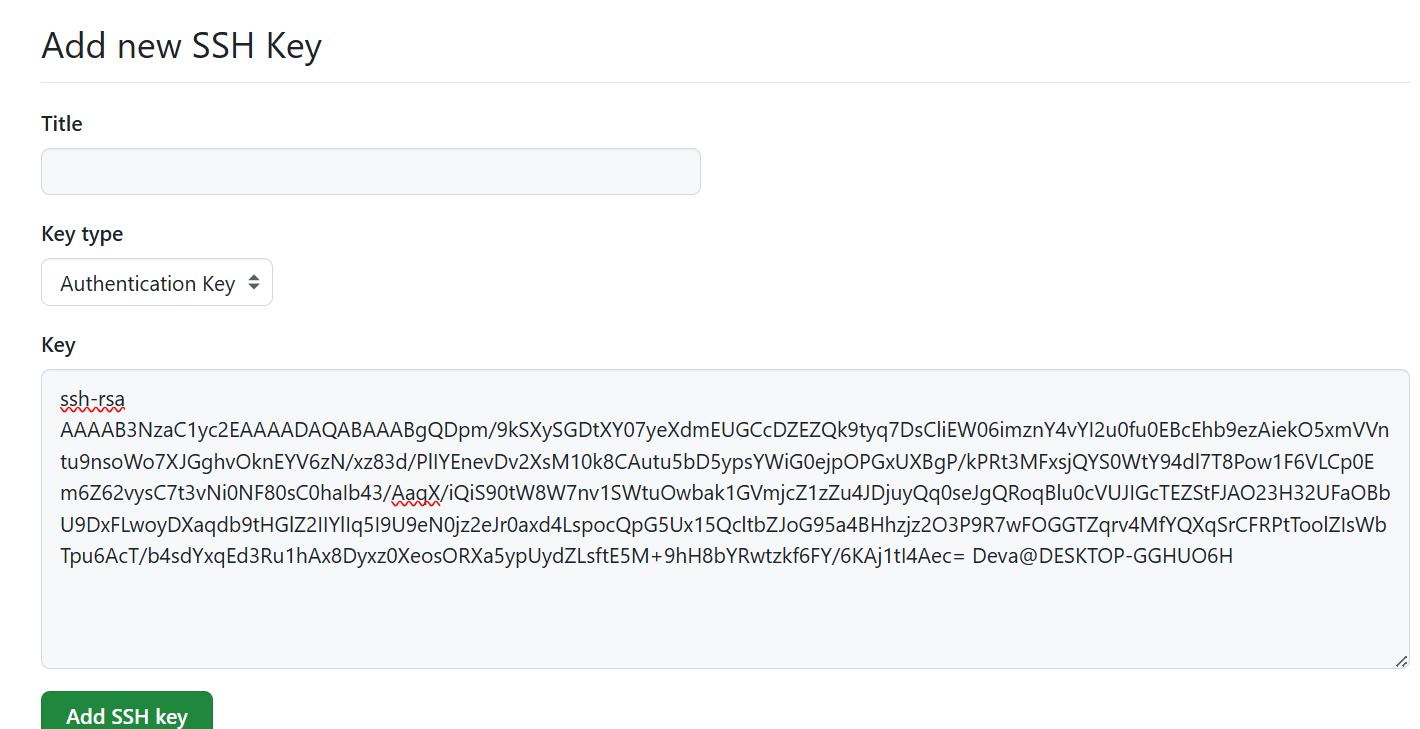
17) Generate a ssh-keygen and configure into github



* **ssh-keygen –t –rsa** : To generate sshkey
* The key will be created local machine



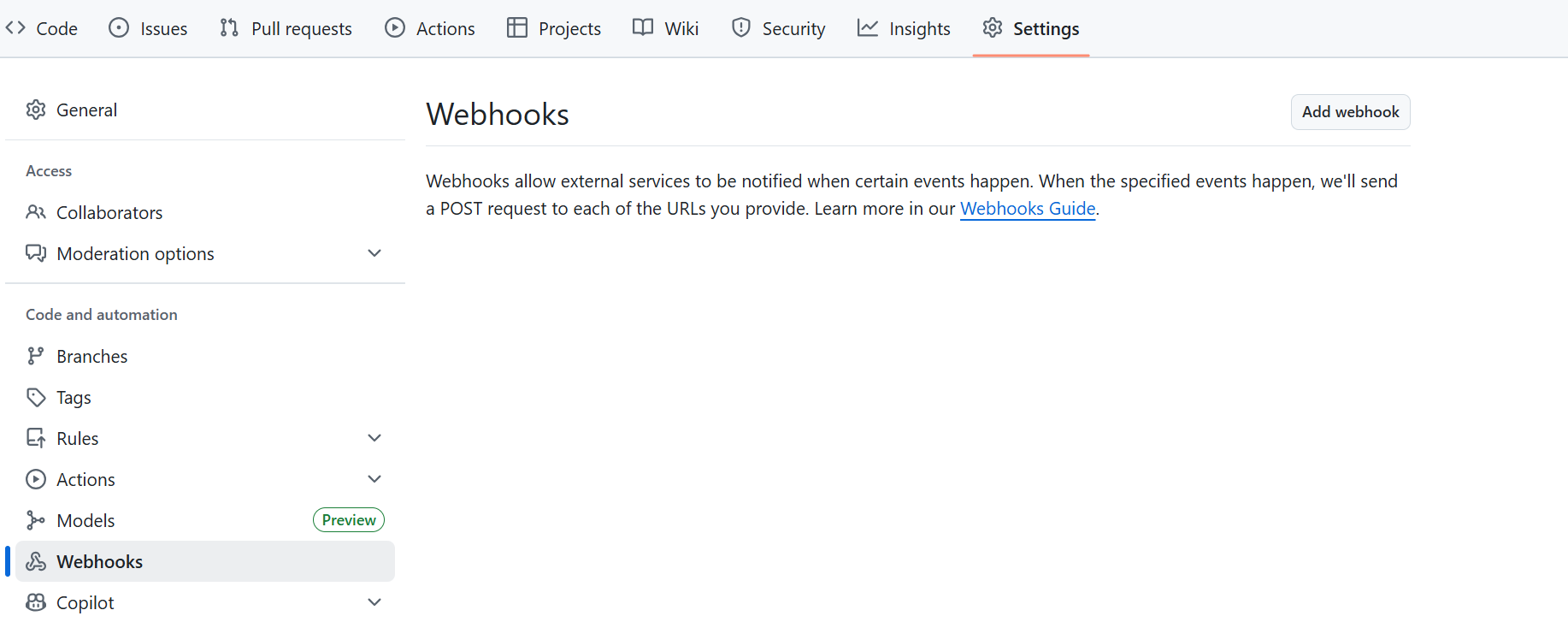
* **Cat “path/to/key”** : To view the sshkey
* Copy the key



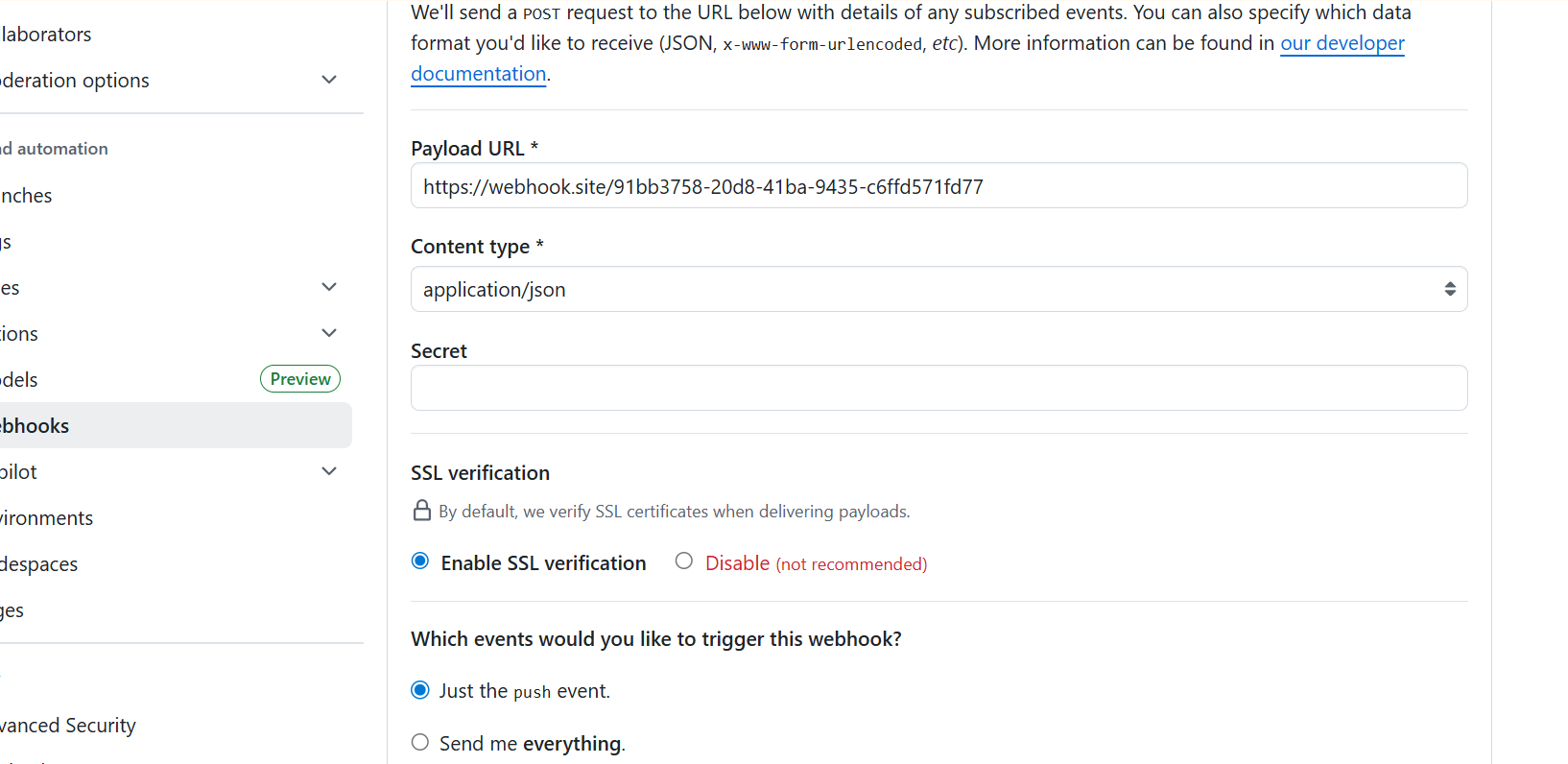
* Add ssh key confirm through the code and configuration is done.

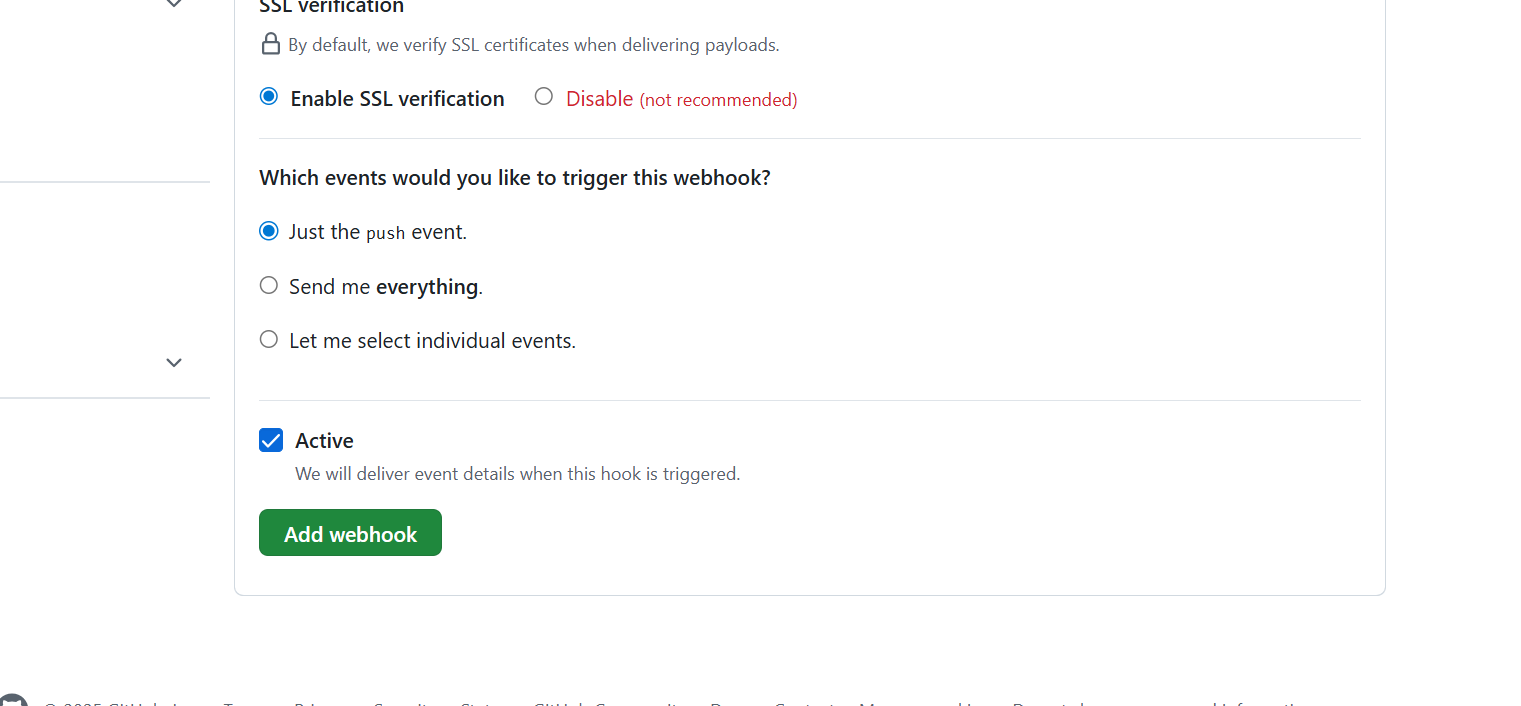
18) Configure webhooks to github

* Open your repository and open settings
* In the left menu select **webhooks** and click add webhook

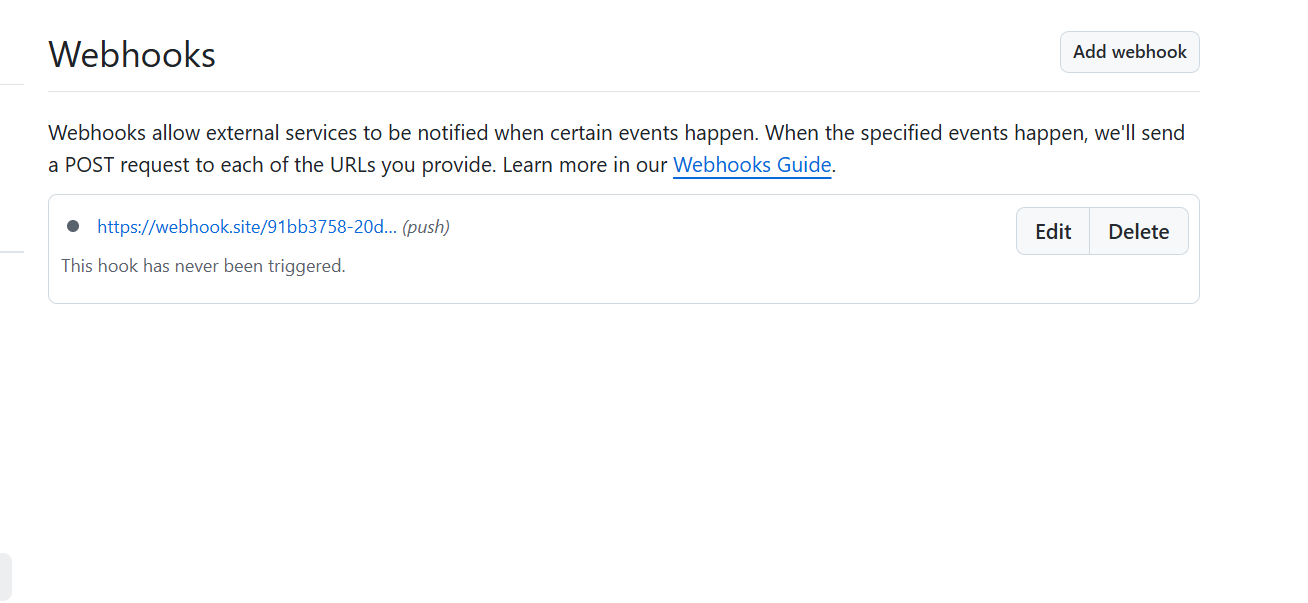


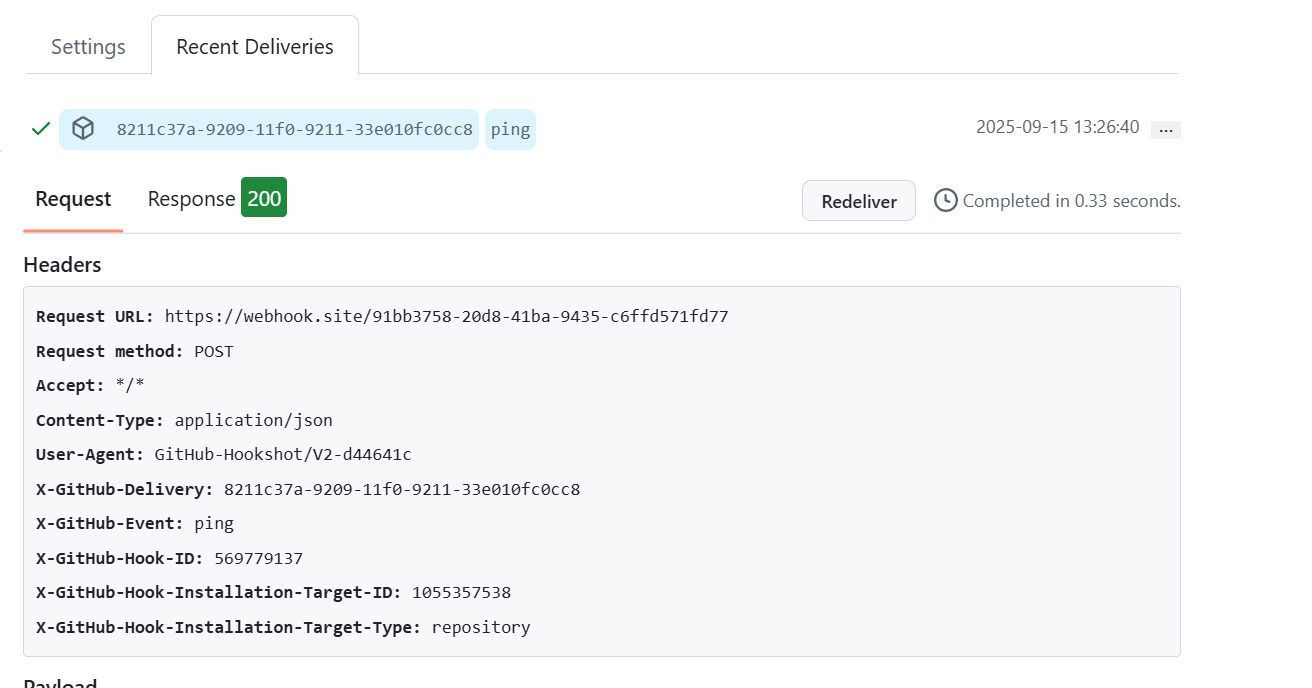
* Give one payload URL
* Give content type 🡪 usually **application/json**
* Give content type choose events trigger the webhook





* And add **webhook**



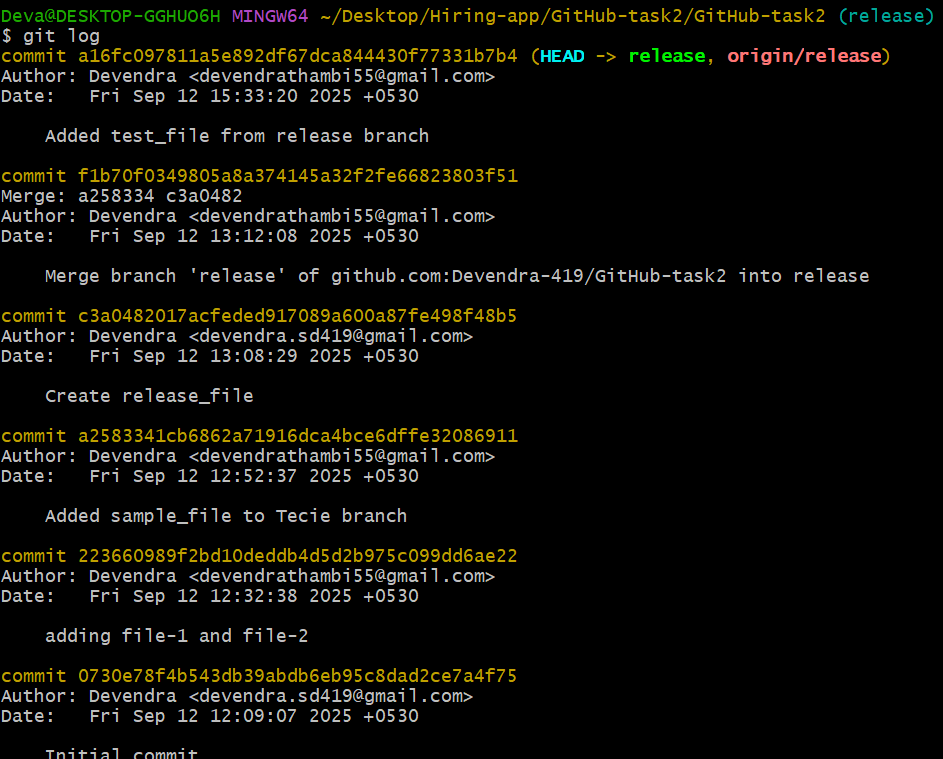


19) Basic understanding of .git file

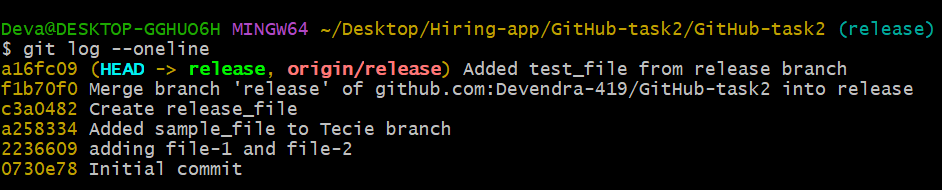
* When you run git init in a project, Git creates a hidden folder called **.git** at the root.
* This folder is the **local database of git.**
* It stores **everything Git needs** : commit history, branches, tags, configuration, and objects.
* If you delete the **.git** folder, you lose version control history for that project.
* **“.git is a hidden directory created by Git. It stores all repository metadata like commits, branches, and configurations. This is what makes a folder a Git repository.”**

20) Check all the logs of git

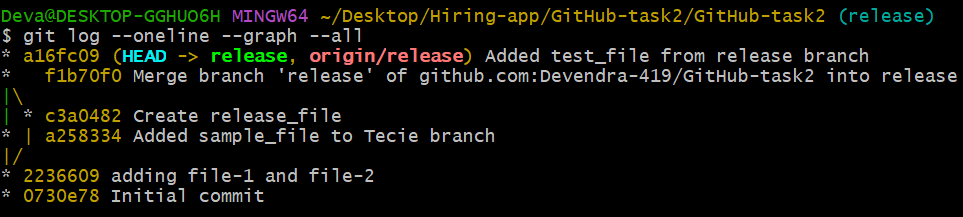
* **git log** - Shows the commit history (commit id, author, date, and message)



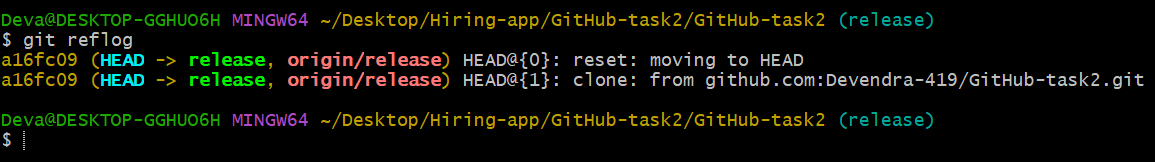
* **git log –oneline** – if we want short form ,we use it



* **git log --oneline --graph --all** -Shows commits as a tree structure, with all branches.

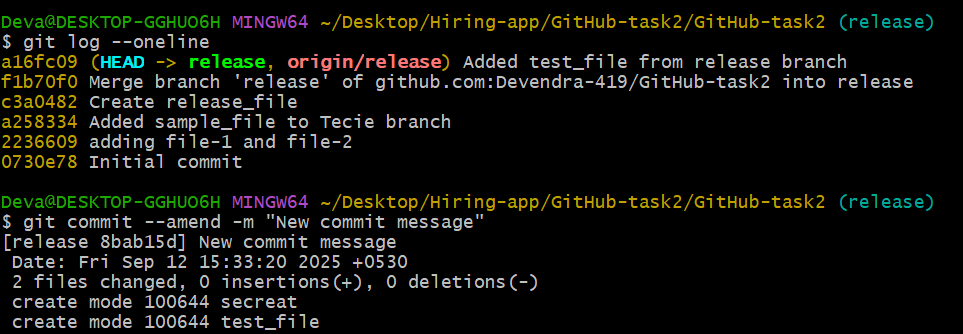


* **git reflog -** This shows all actions, like commit, checkout, reset, rebase, etc..
* Even if you lose a branch, git reflog helps recover commitd



21) Rename the commit message

* It changes only the most recent commit message



22) Merge multiple commits into single commit

* Git log and I want to merge the 2 commits into a single commit
* **Git rebase –I HEAD~n**
* (where n = number of commits back you want to edit)

