**GIT**

**Git** is a widely adopted distributed version control system (VCS) essential for managing modifications in source code throughout the [software development process](https://www.simplilearn.com/tutorials/programming-tutorial/what-is-sdlc).

**A Git repository** is a location where your project resides, acting as a storage area. This repository can exist locally within a directory on your computer or on a cloud-based platform like GitHub.

**Git** is a tool for version control that enables you to monitor and record the evolution of your source code. **GitHub** is an online [hosting](https://www.simplilearn.com/free-web-hosting-sites-article) service that facilitates the management of Git repositories. GitHub offers a user-friendly web interface along with functionalities such as permission management, task organization, error tracking, and the capability to handle feature suggestions.

**In Git, branching** allows you to veer off from the primary development path and proceed with separate tasks without impacting the main workflow. This technique enables the isolated development of features, bug fixes, or experimentation within a specific section of the repository, ensuring that each process remains distinct from the others.

Branch in Git is used to keep your changes until they are ready. You can do your work on a branch while the main branch (master) remains stable. After you are done with your work, you can merge it with the main office.

**While**[**working on Git,**](https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner)**we actively use two repositories.**

* **Local repository**: The local repository is present on our computer and consists of all the files and folders. This Repository is used to make changes locally, review history, and commit when offline.
* **Remote repository**: The remote repository refers to the server repository that may be present anywhere. This repository is used by all the team members to exchange the changes made.

**Features of Git**

* Tracks history
* Free and open source
* Creates backups
* Scalable
* Supports collaboration
* Branching is easier
* Distributed development

**GIT COMMANDS**

To check git version**- git –version**

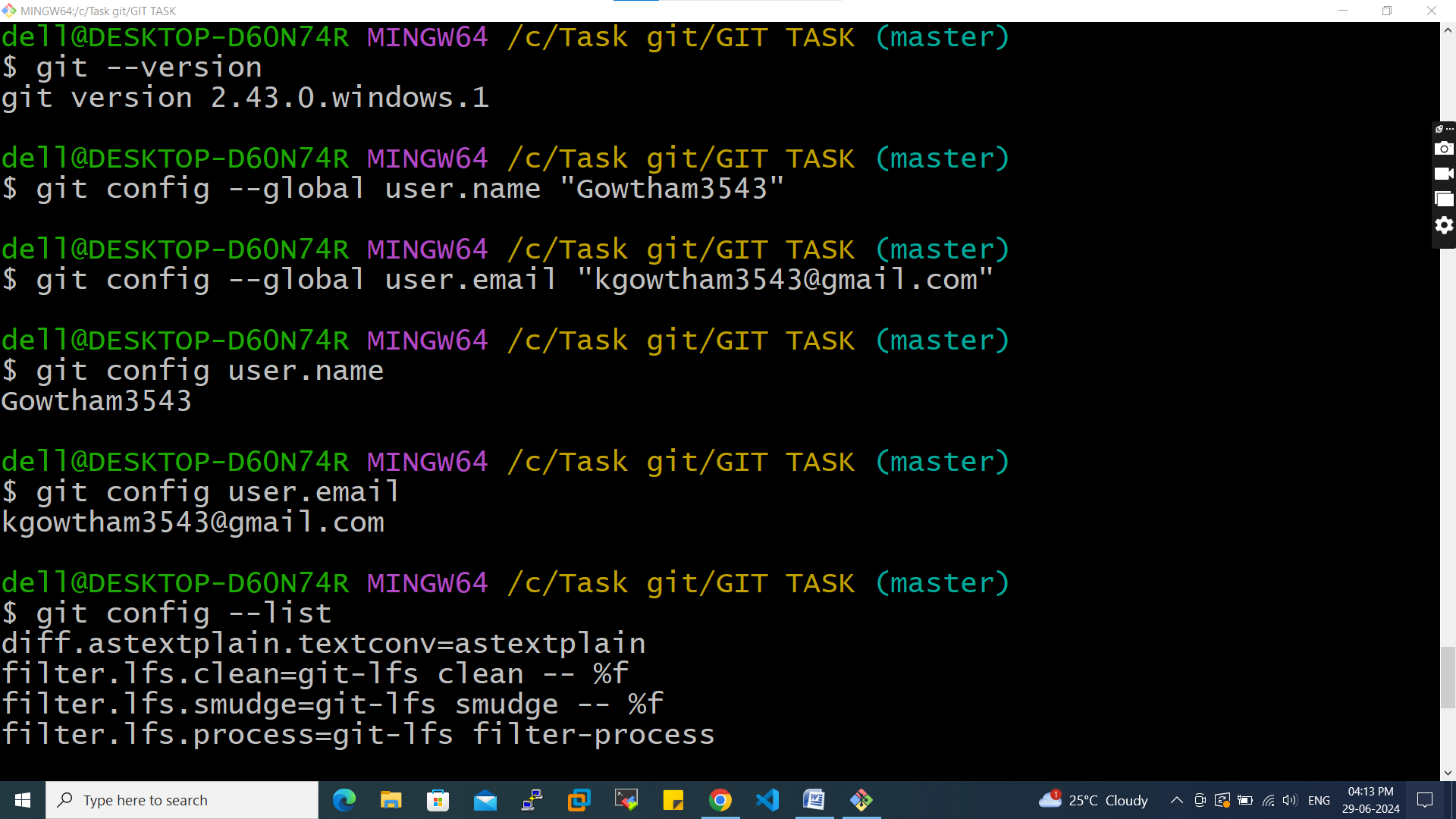
To configure username and email ID and to get list- **git config --global username<>**

**git config --global email id<email id>**

**git config –list**

In order to get only list of username and email ID- **git config user.name**

**git config user.email**

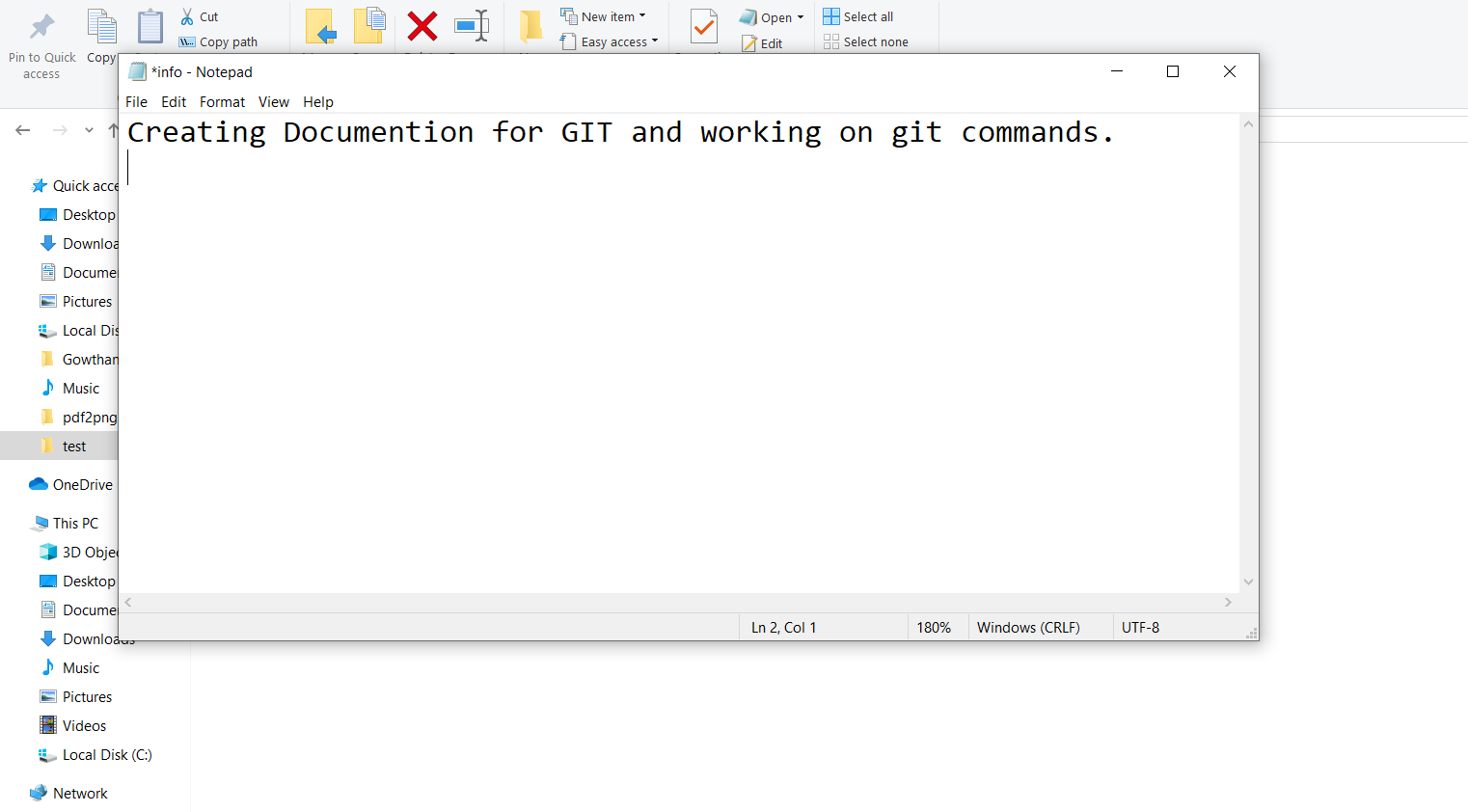
****

Now we will create file name TEST and work on git commands

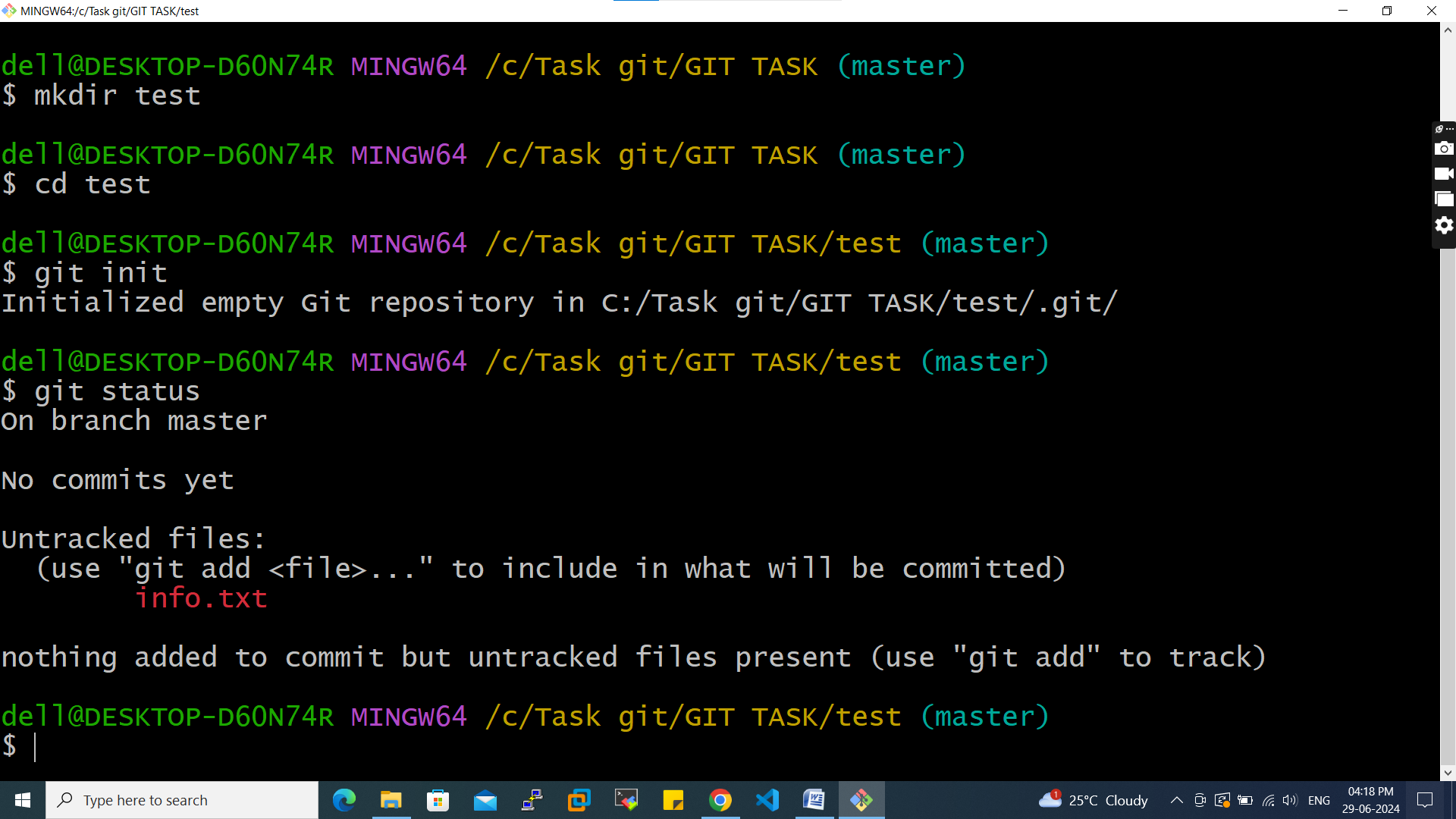
To create file **mkdir** test

Now go to test directory **cd** test

Create a new git instance for a project **git init .**

Now create a text file in test folder normally 

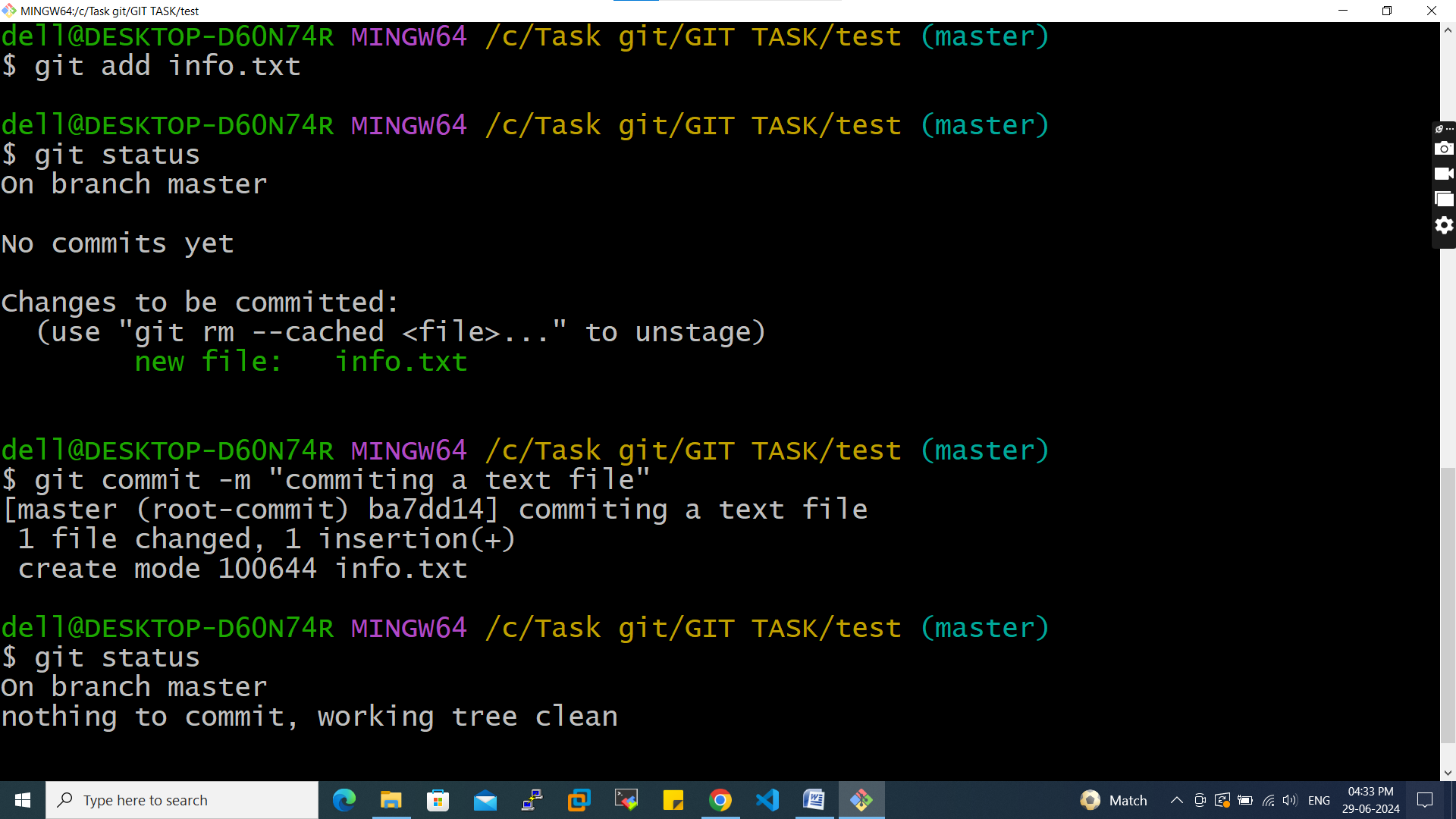
Then Check the status of the repository- **git status**

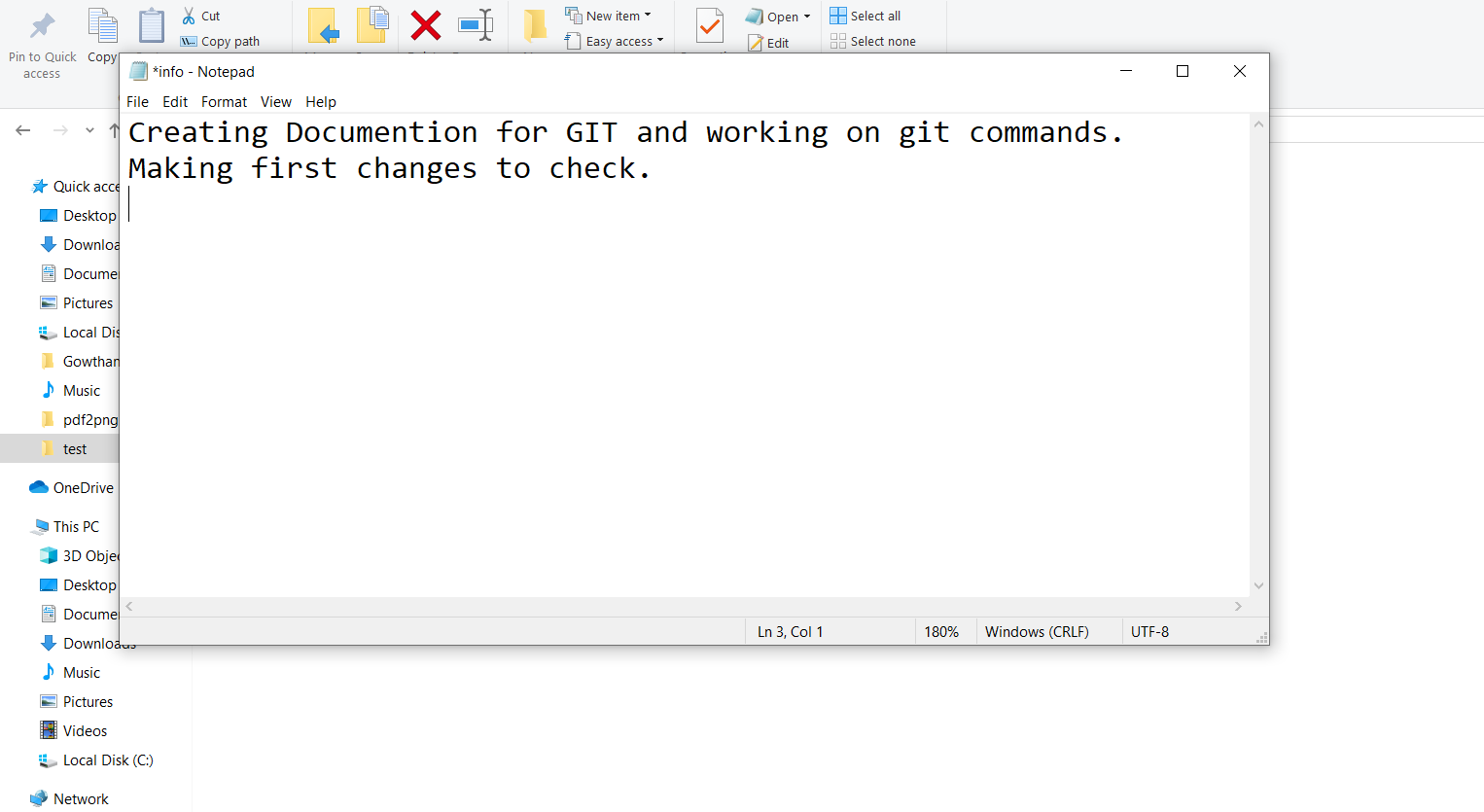


Now we need to add created text file so that we can commit- **git add <file name>**

To add all untracked file- **git add .**

Then we need to commit the changes- **git commit –m “commiting the text file”**

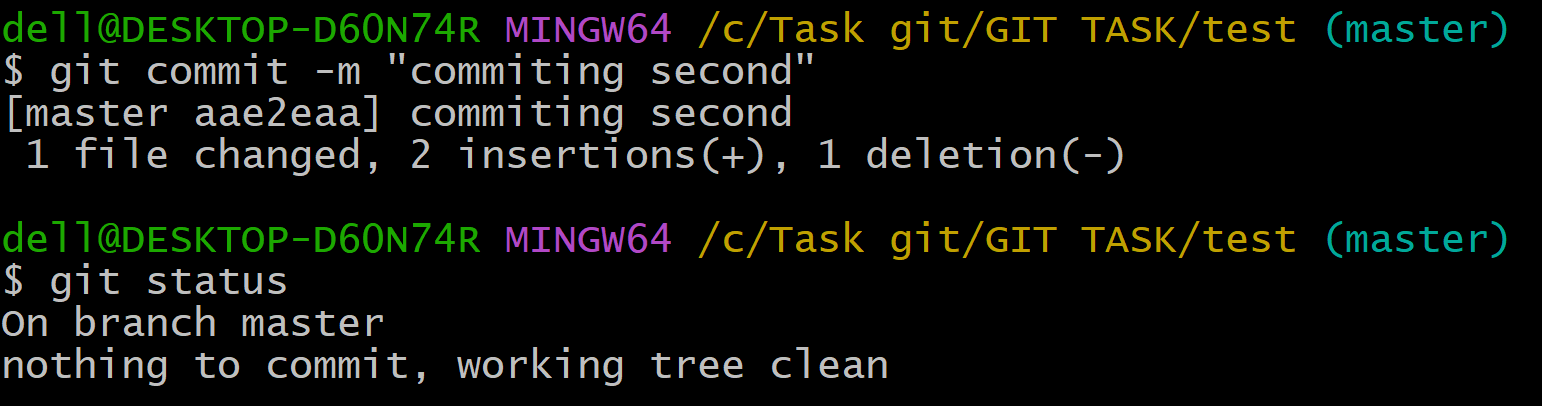
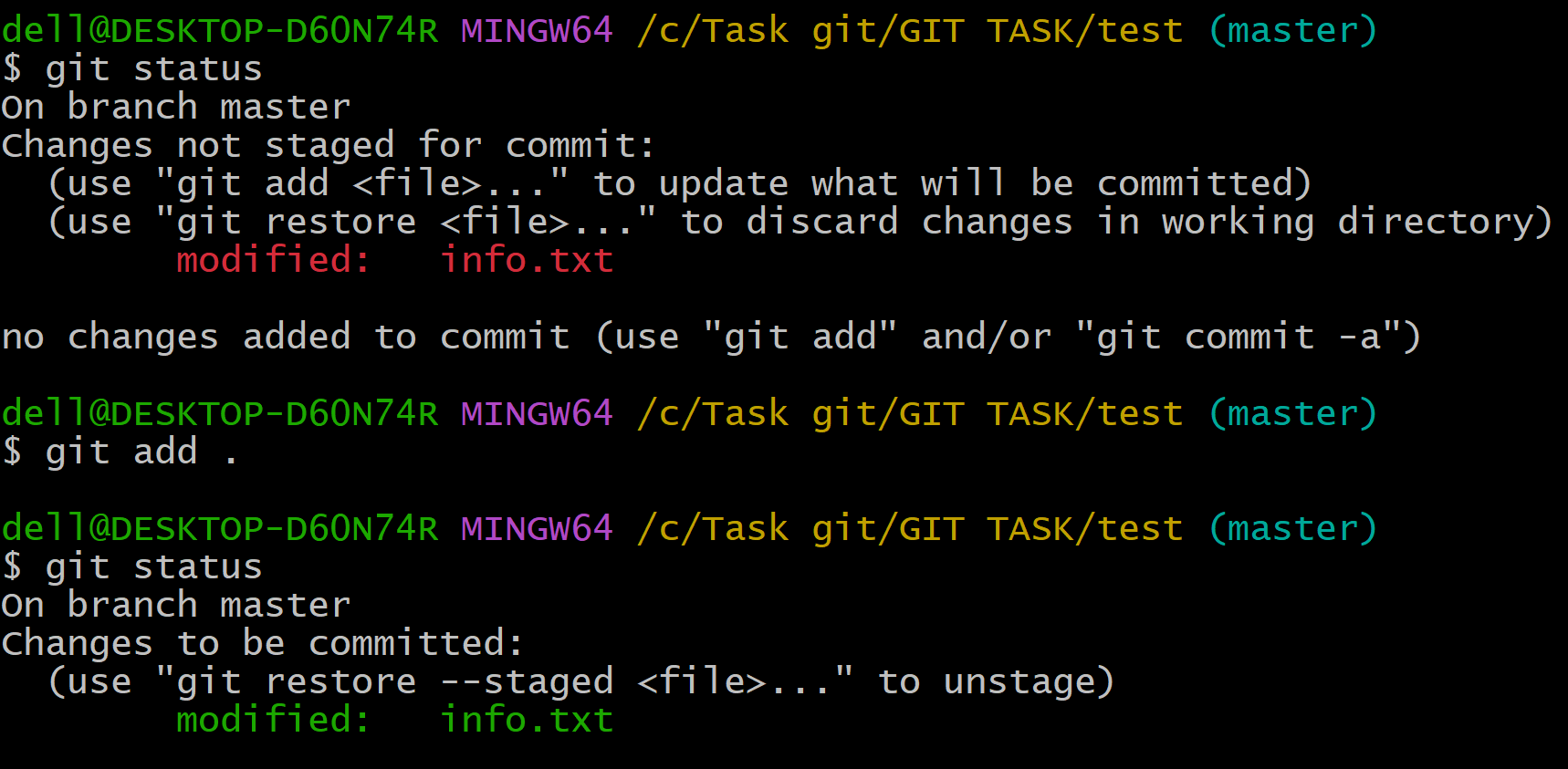


Now make changes in text file normally 

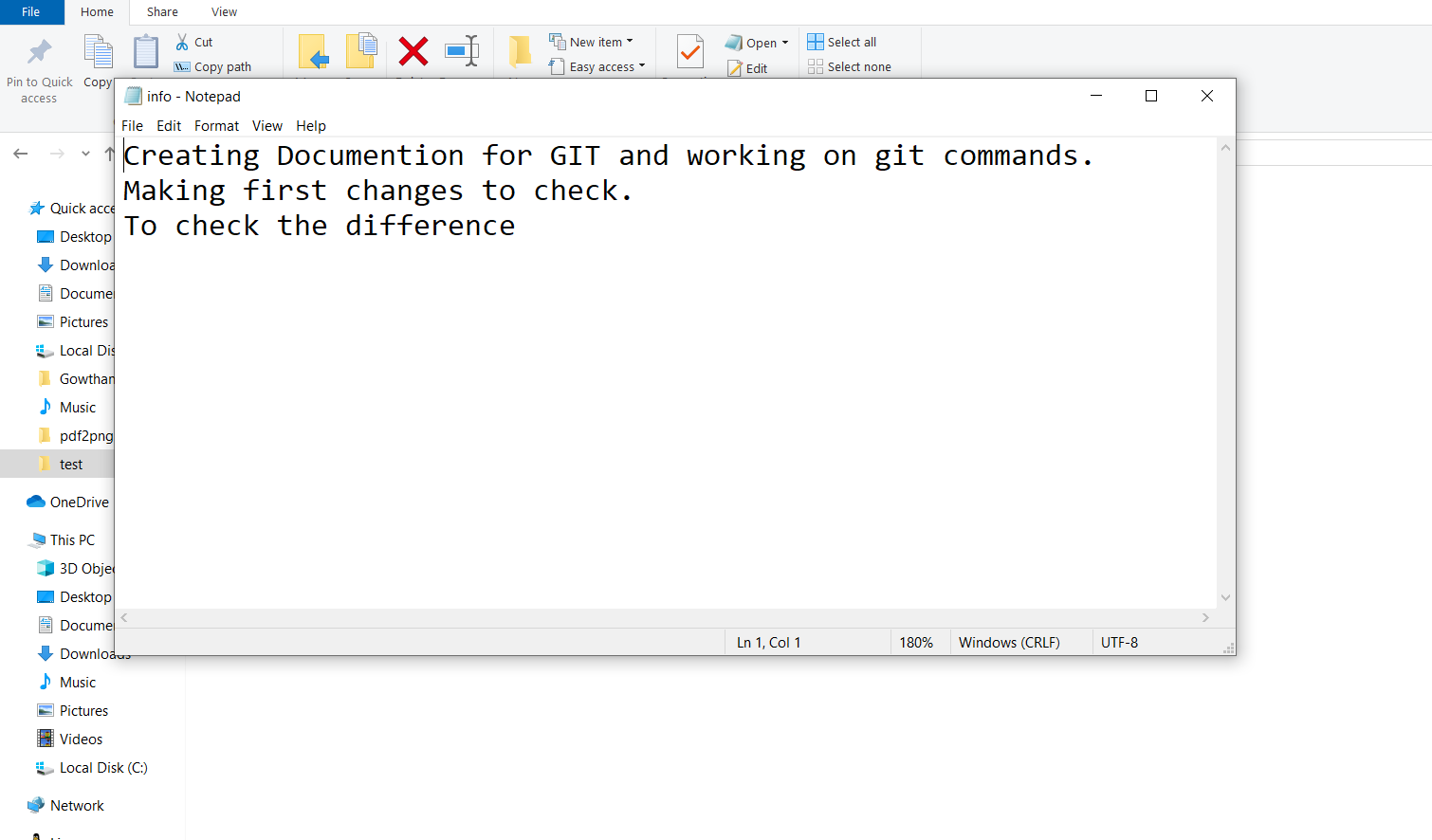
After making changes we need to add and commit

**git add .**

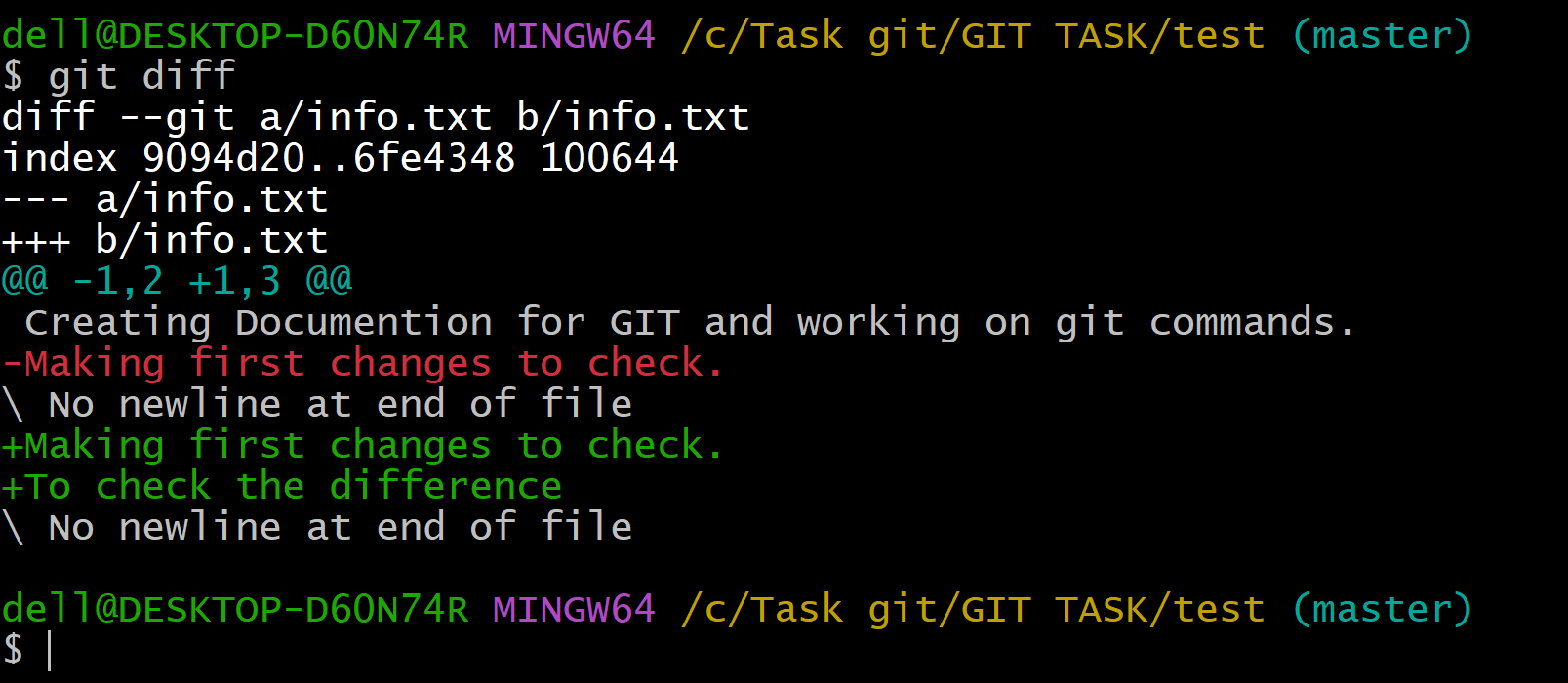
**git commit –m “commiting second”**

****

Now again we make changes in text file normally to compare the difference since last commit

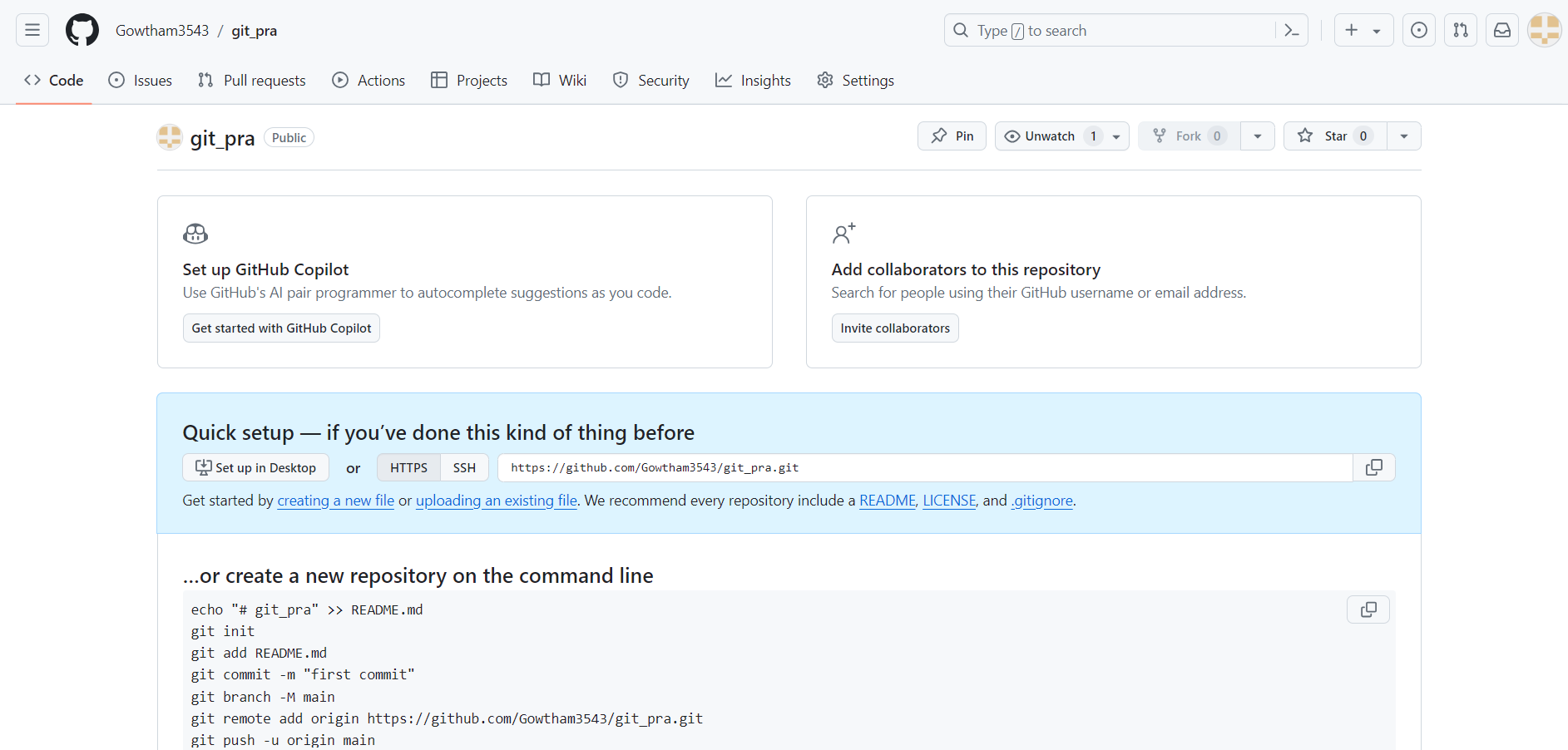


To compare changes**- git diff**



As we already add GITHUB username in GIT configuration

(git config - -global user.name <username>)

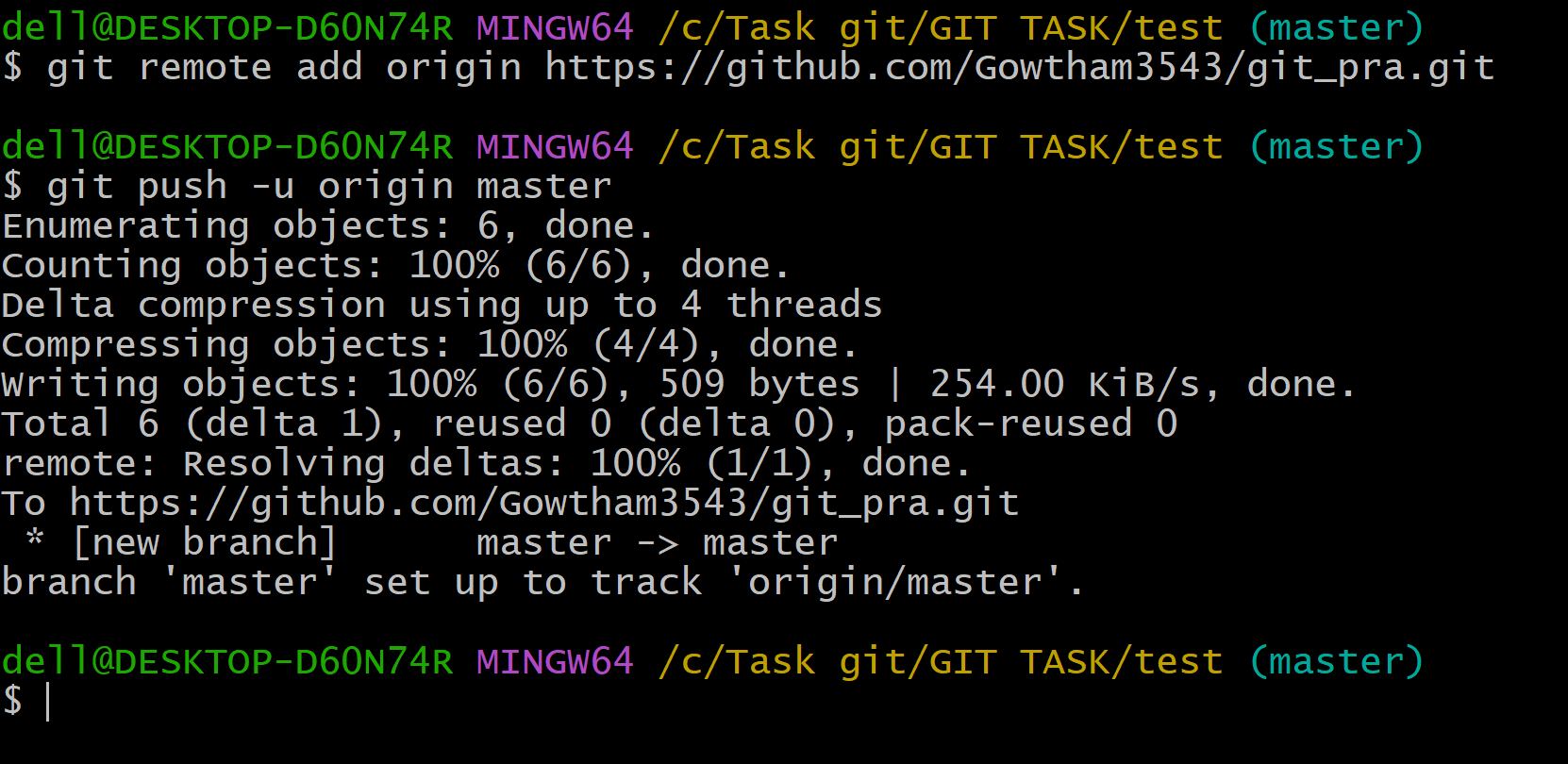
Now we need to create remote repository – git\_pra

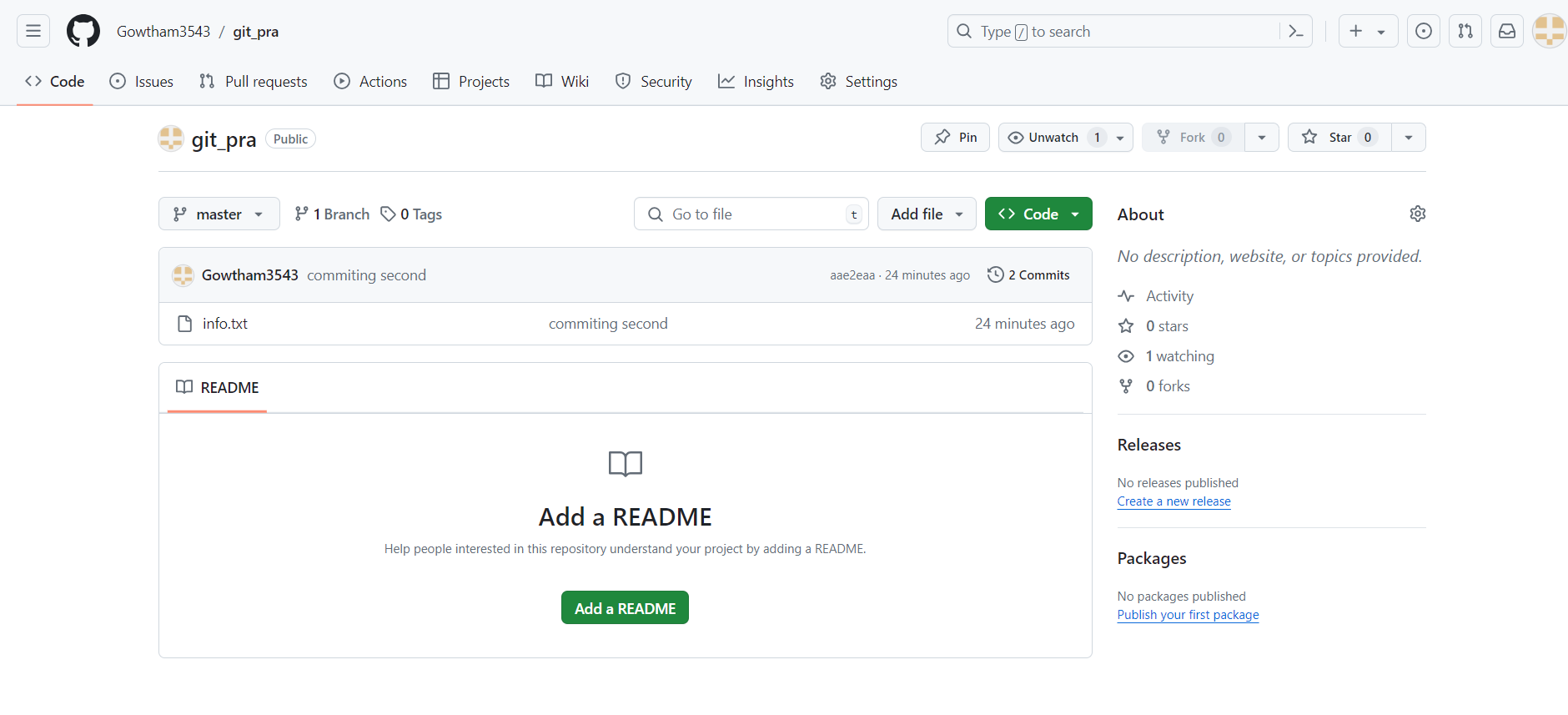
Then connect the local repo to remote repo

**git remote add origin** [**https://github.com/Gowtham3543/git\_pra.git**](https://github.com/Gowtham3543/git_pra.git)

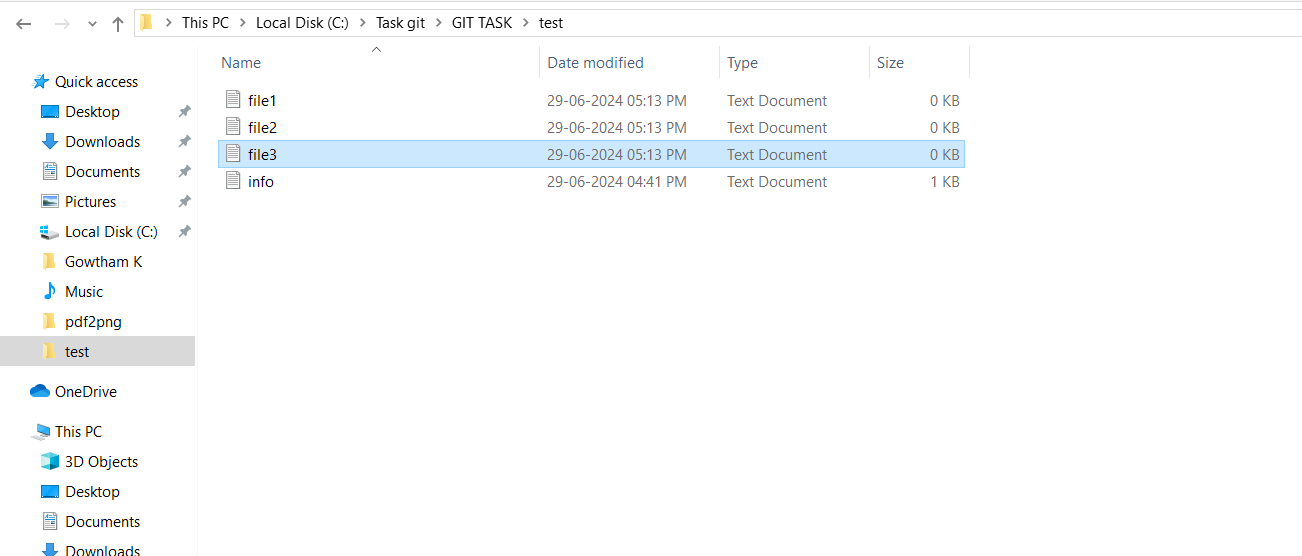
Then push the file to remote repo

**git push -u origin master**

****

File add to remote repo****

Now we need to create three more files normally



Now we need to create new branch first\_branch

**git branch first\_branch**

To check list of branch

**git branch**

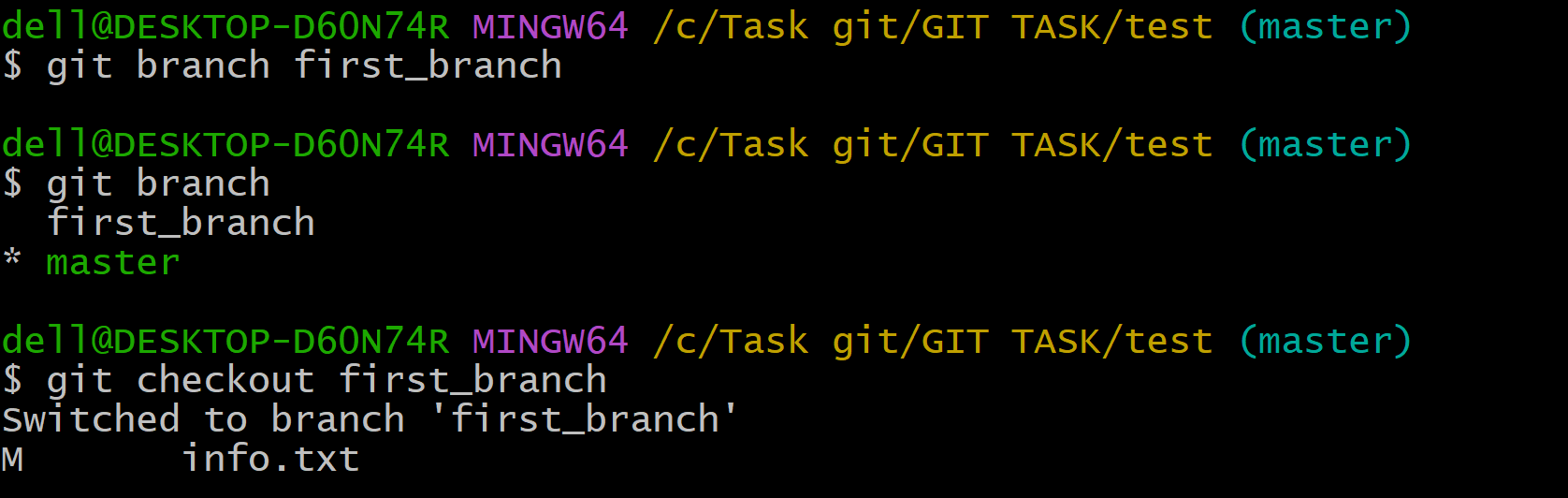
Now get in to new branch

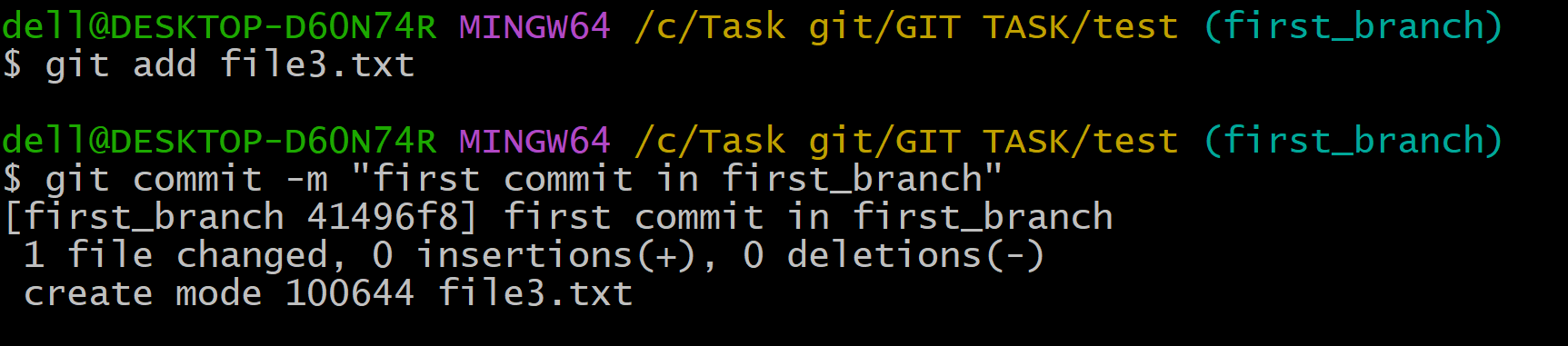
**git checkout first\_branch**

Now we need to add file3 and commit it

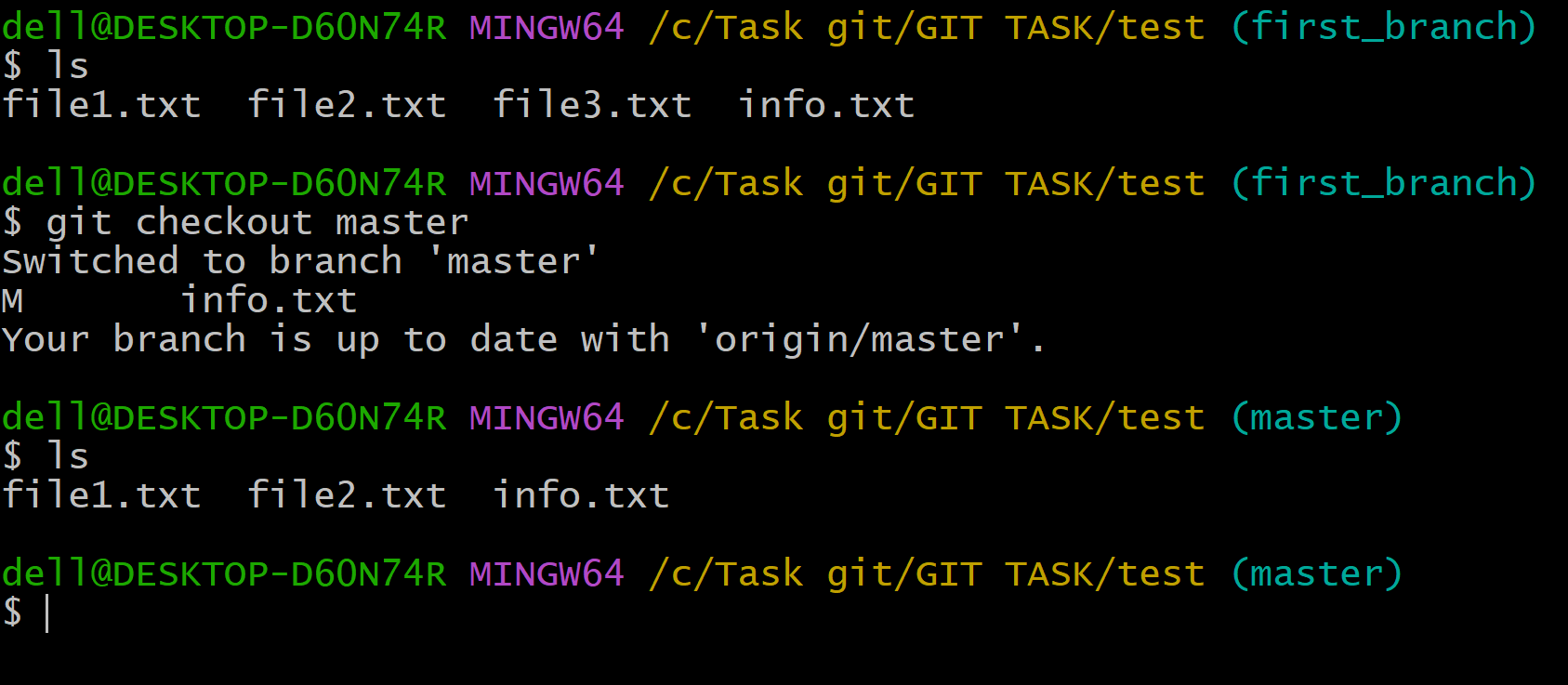
**git add file3**

**git commit –m “first commit in first\_branch”**

****

****

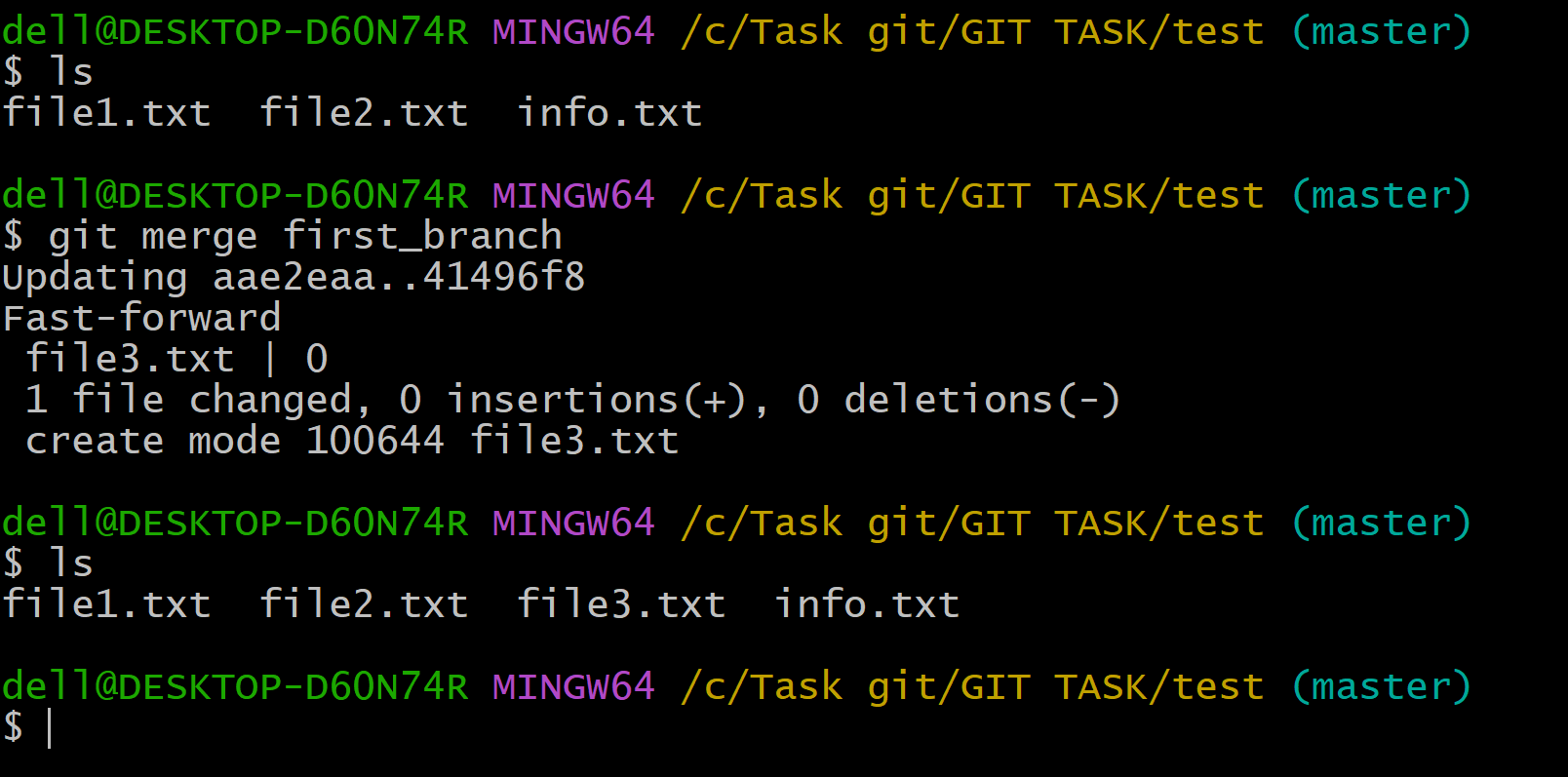
Now if we see first\_branch as access to all files but master branch doesn’t have file3 as we committed in first\_branch only



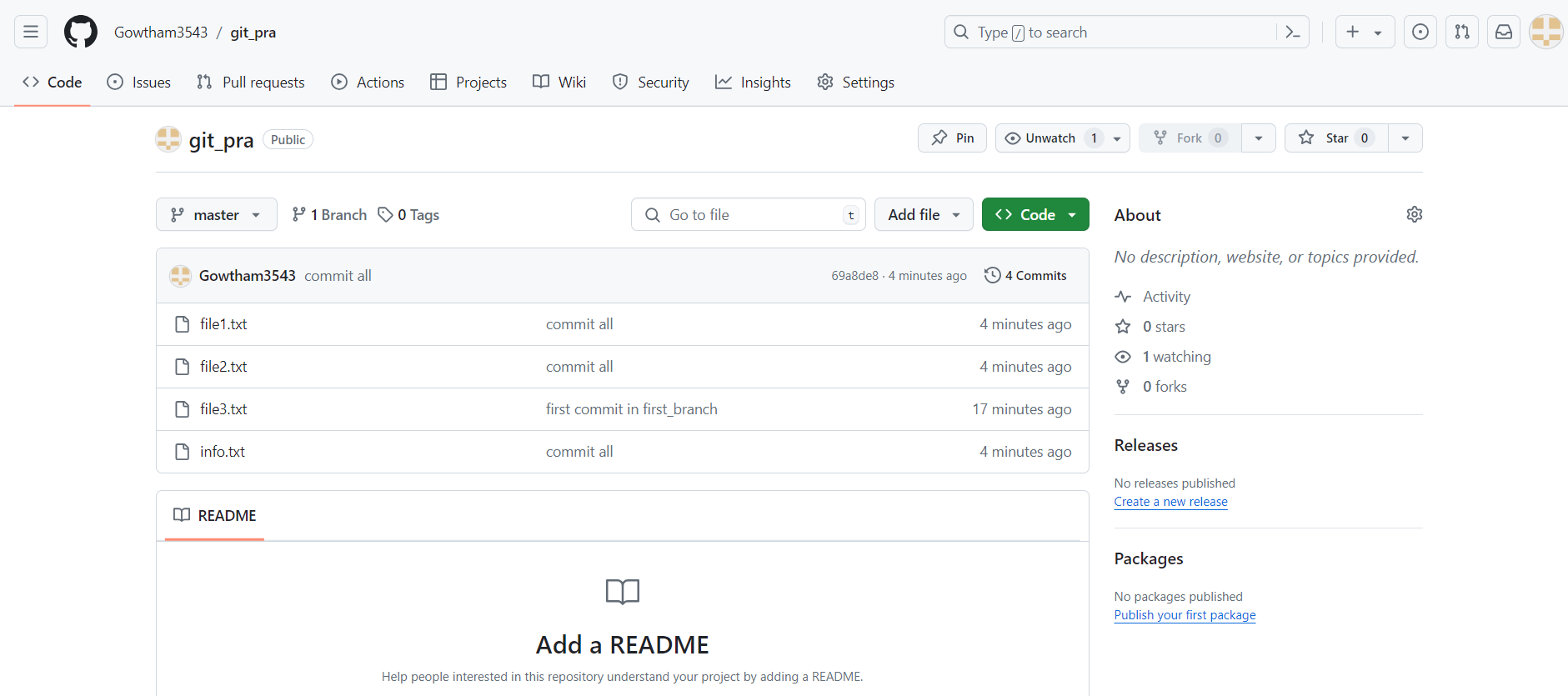
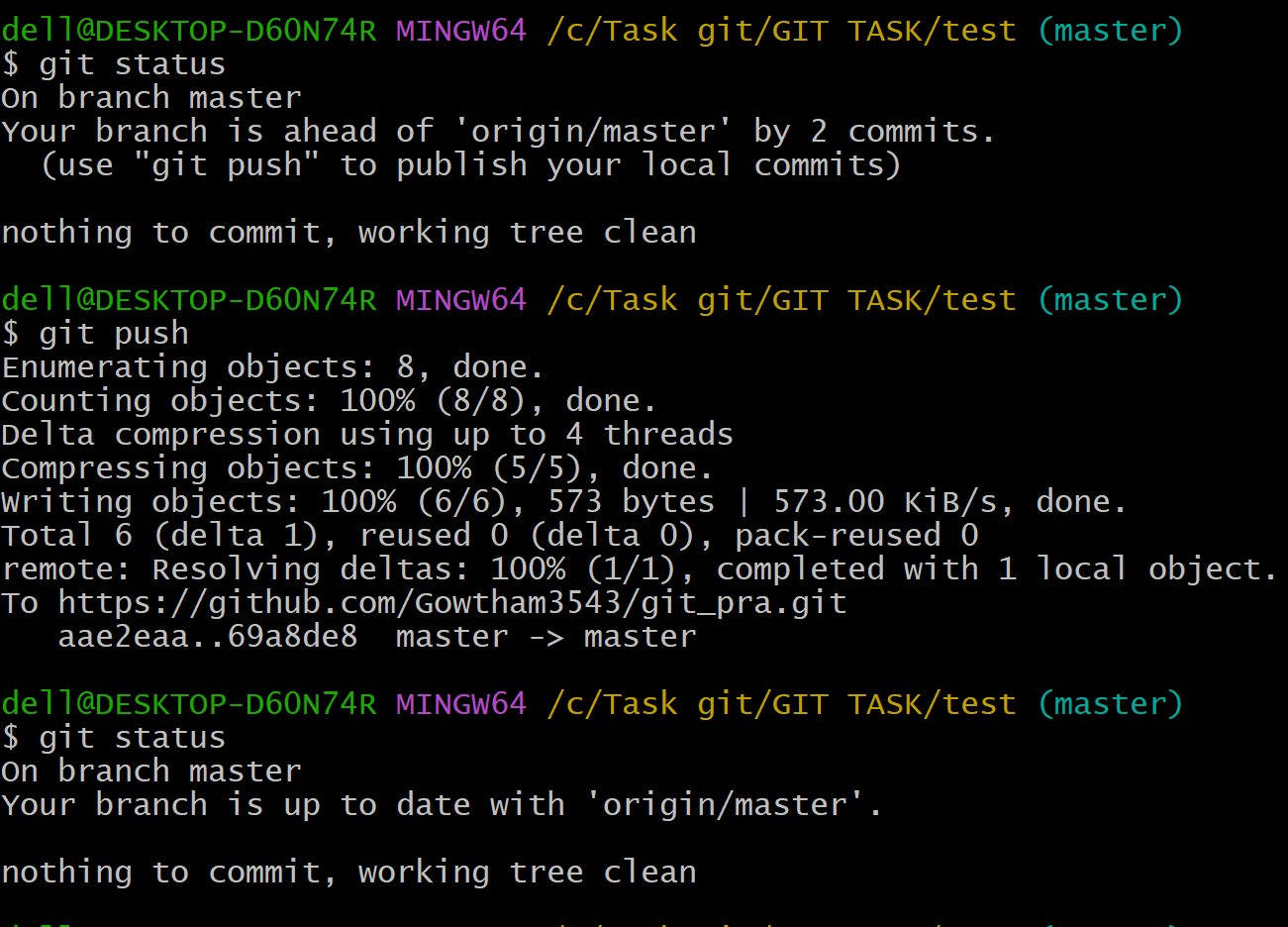
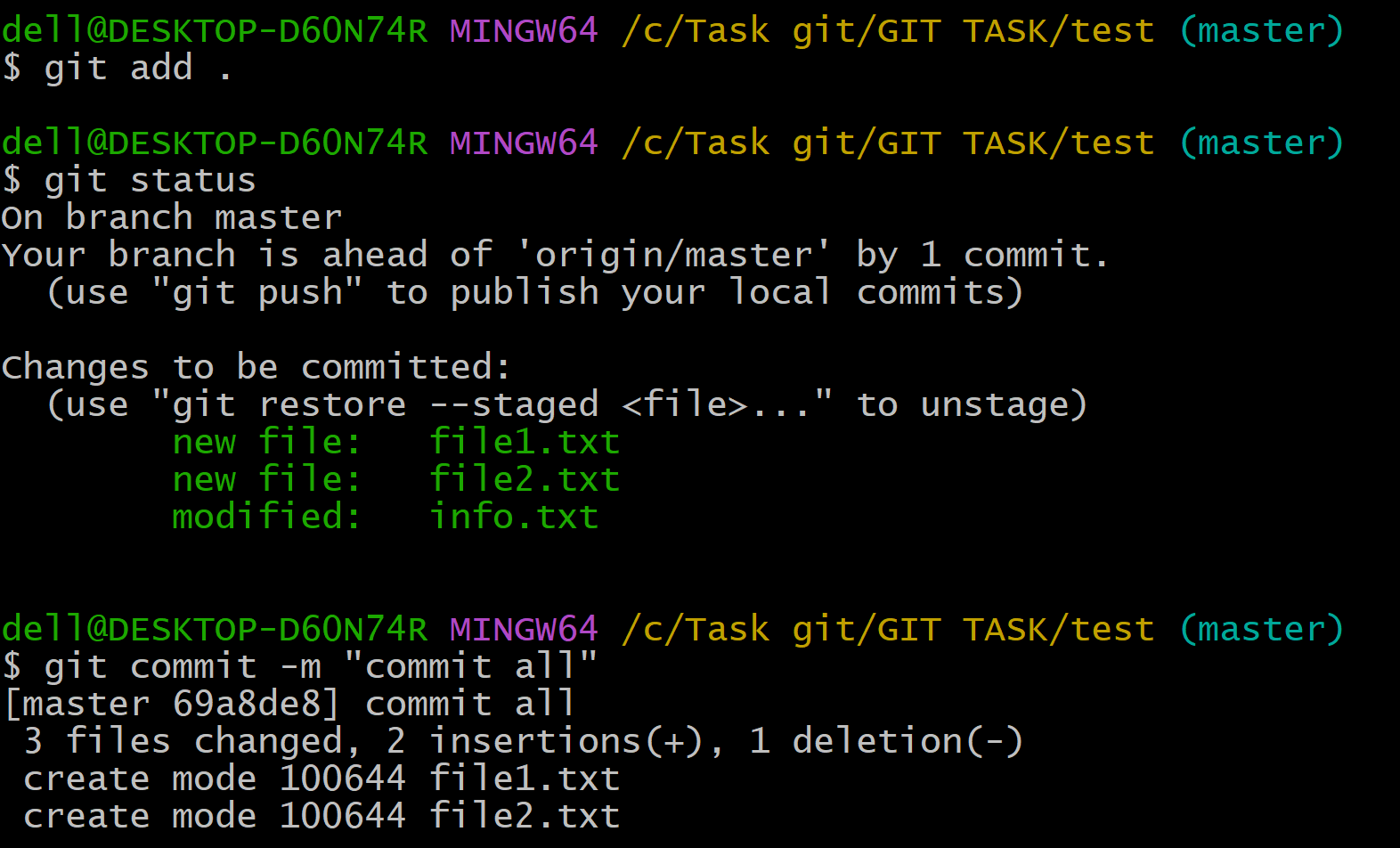
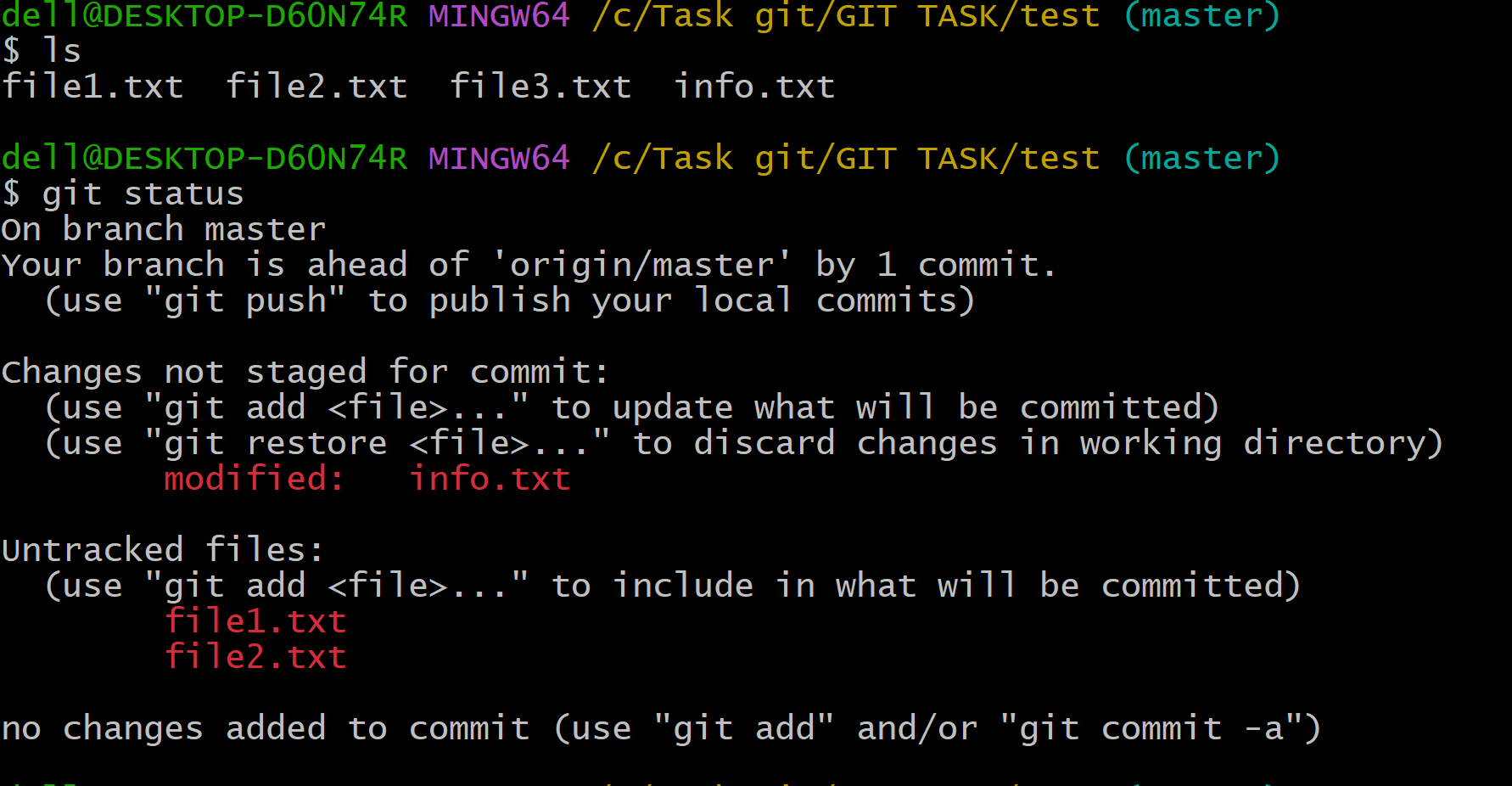
In order to merge the file from first\_branch to master branch

**git merge first\_branch**

When you’re merging you need to be in the branch where you want file to come.

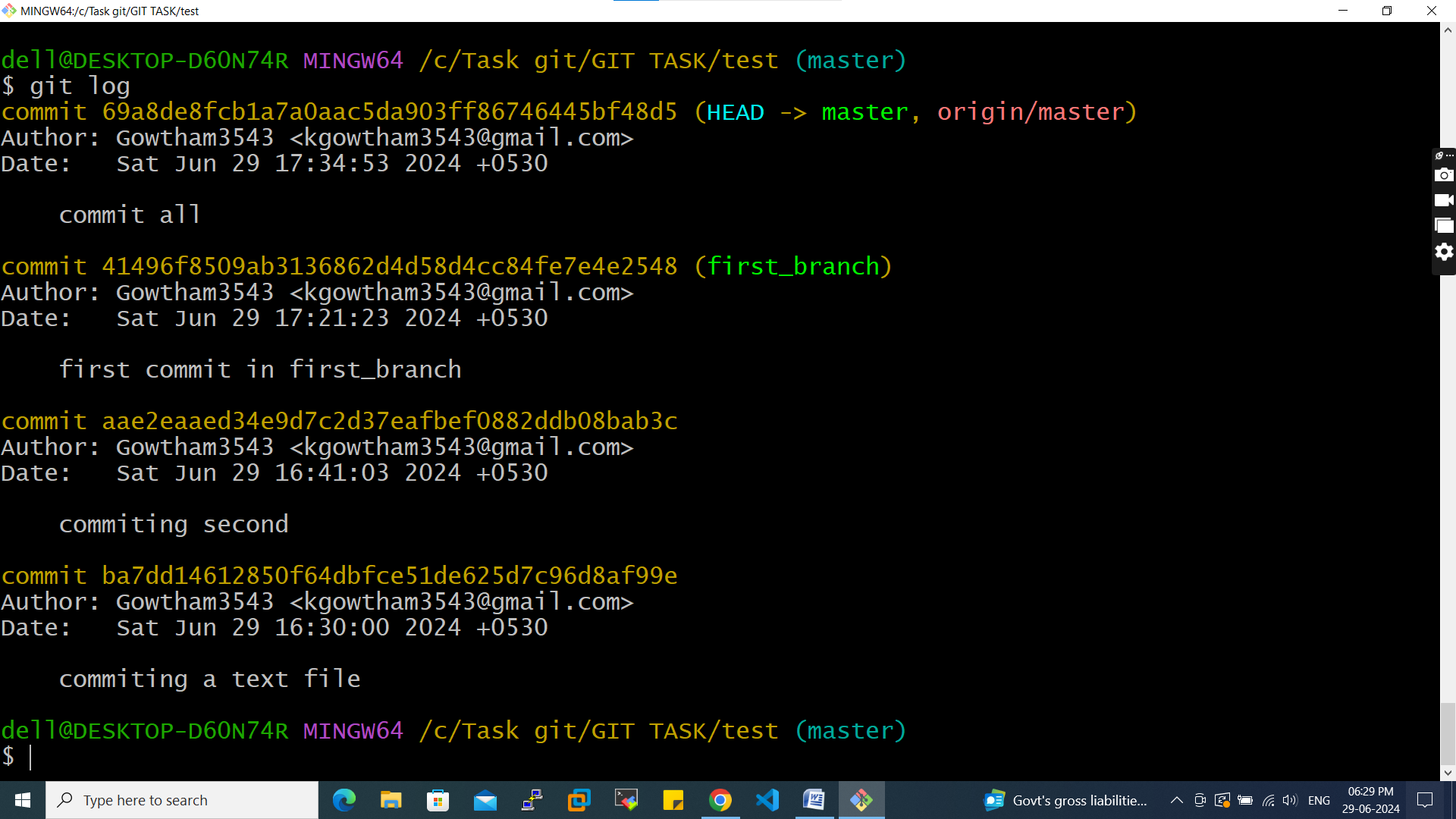


Now I have add all files, committed and pushed them to remote repo



Check all the information regarding the commits that were made.

**git log**

****

I have added file5.text in remote repo tried fetching and pulling

**git fetch**: Downloads changes from a remote repository but does not merge them into the local branch.

**git pull**: Downloads changes from a remote repository and immediately attempts to merge them into the current branch.

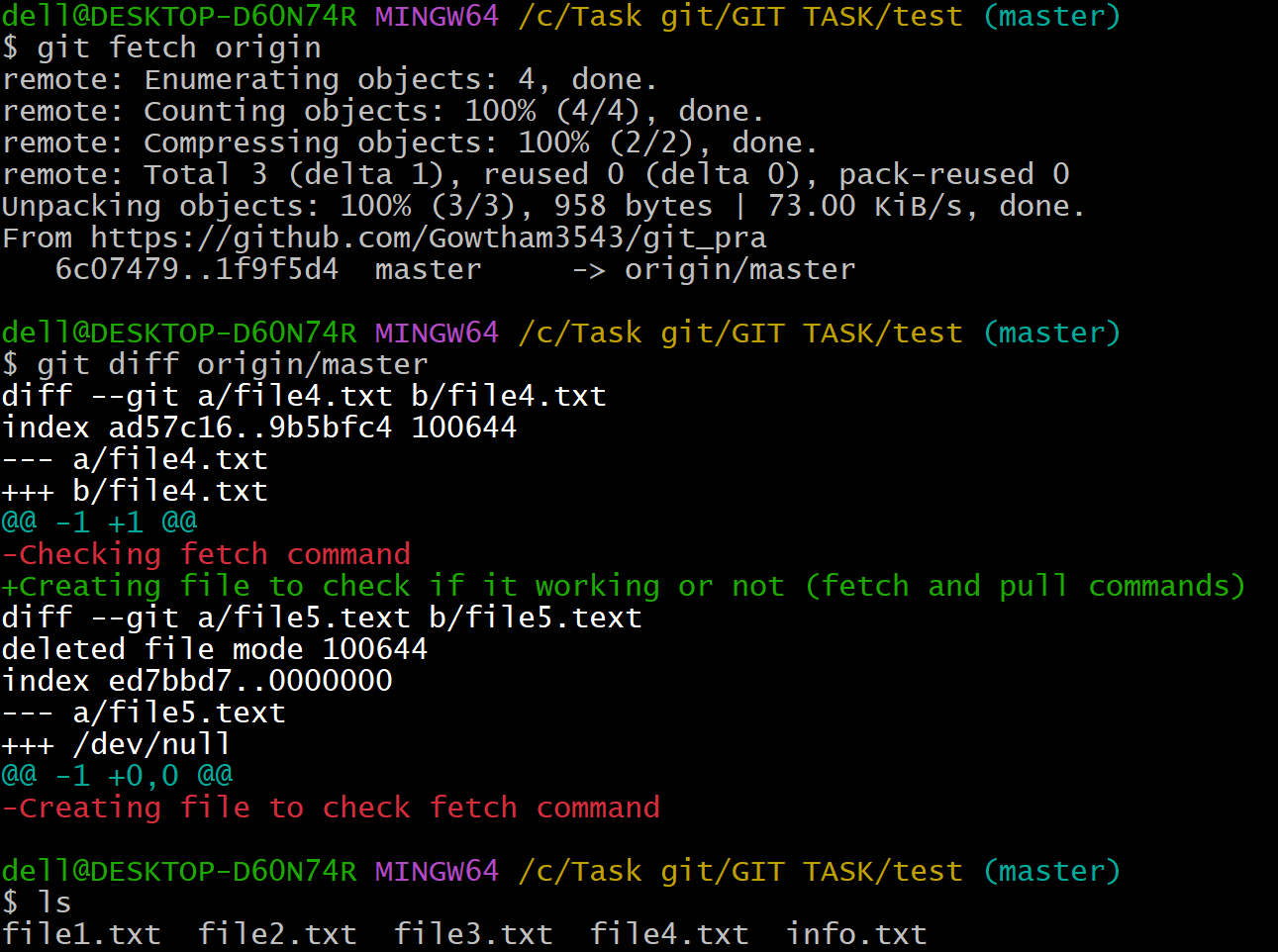
I have fetched file and saw what changes as made before I merge to local repo so we use fetch (you see that file5.text is not in local but we can see what changes are there)

Whereas pull will not allow you to see what changes made it will immediately merge

**git fetch origin**

We use below command to see changes

**git diff origin/master**



Now I deleted file4.text in remote repo and I used fetch command it didn’t delete in local whereas when I tried pull command it deleted as it makes changes immediately

