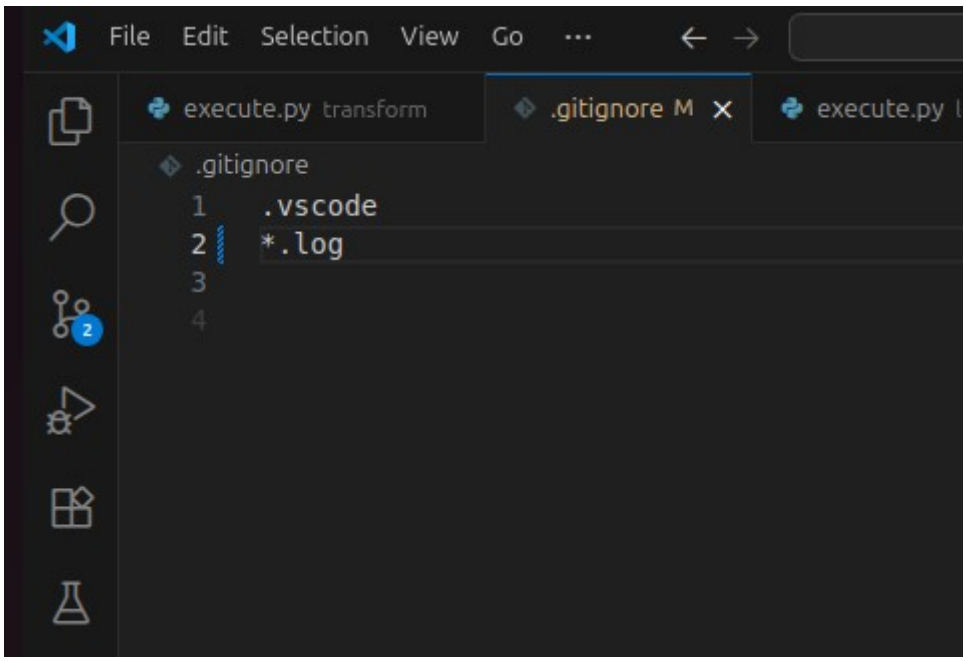
1. Initialize git repository in your working directory

```
(venv) ardent@ardent:~/Workspace/etl$ 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/ardent/Workspace/etl/.git/
```

2. Create a .gitignore file and add the files and folders you do not want to track/push

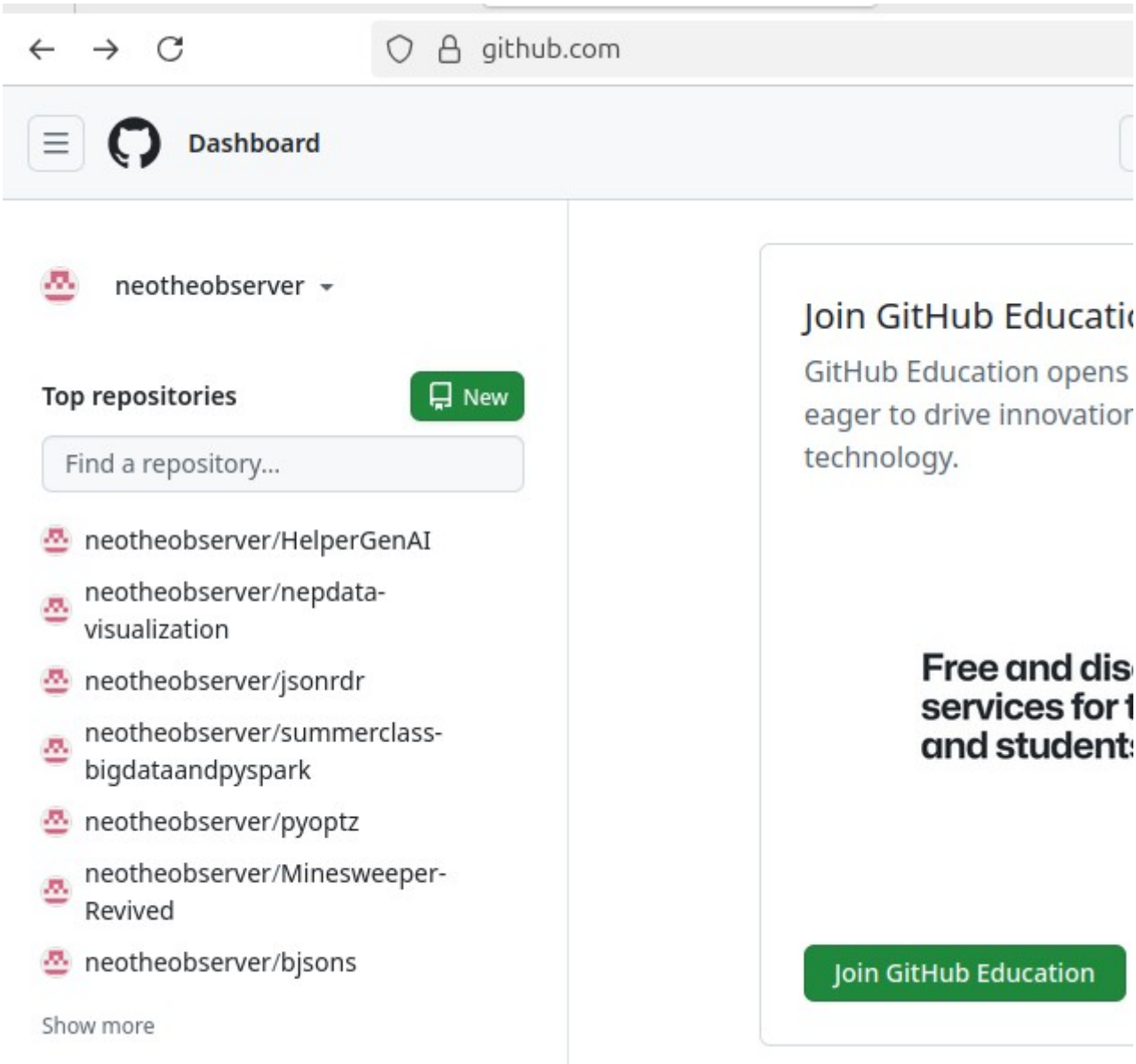


I have only added .vscode and all the files with .log extension for now as I want to push/track all others except for the launch.json file which contains my passwords and configs.

3. Then start tracking the file using git add and commit the changes using git commit. Use . With add command to include all files and use -m flag with commit command to write as message for that commit. Follow the standard for writing messages. Do not copy mine. Do your own

```
(venv) ardent@ardent:~/Workspace/etl$ git add .
(venv) ardent@ardent:~/Workspace/etl$ git commit -m "ref: for aws"
[master (root-commit) 887c237] ref: for aws
 4 files changed, 349 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 load/execute.py
 create mode 100644 transform/execute.py
 create mode 100644 utility/utility.py
(venv) ardent@ardent:~/Workspace/etl$
                                        ardent (5 minutes ago)   Ln 2, C
```

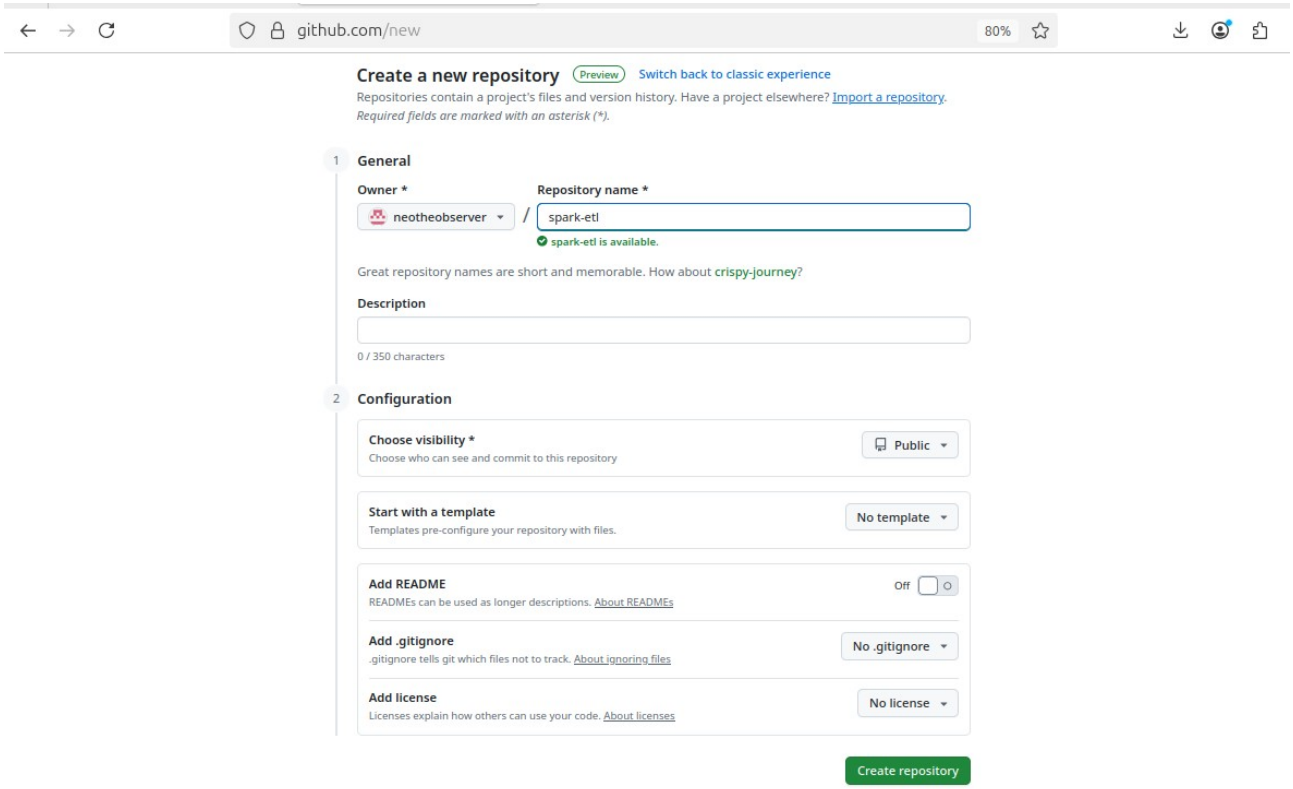
4. Login to Github.com and Go to your dashboard



5. Click the New Button to create new repository

Give a name to the repository. Leave everything else to default unless you know what you are doing. Make sure the visibility is set to public if you want others to see this repo.

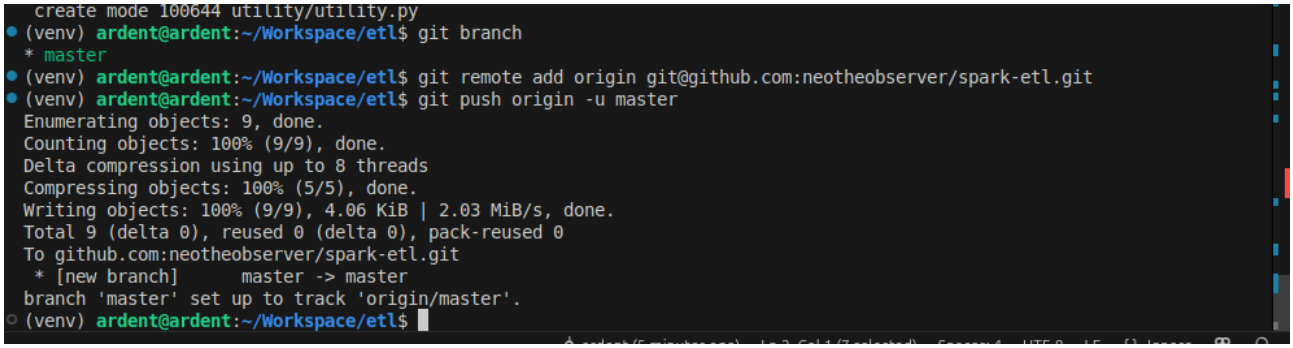
Then click on Create repository



6. In the next page follow the instruction at the below. To push an existing repository from command line as this is our case. We have already created added and committed an local repo.

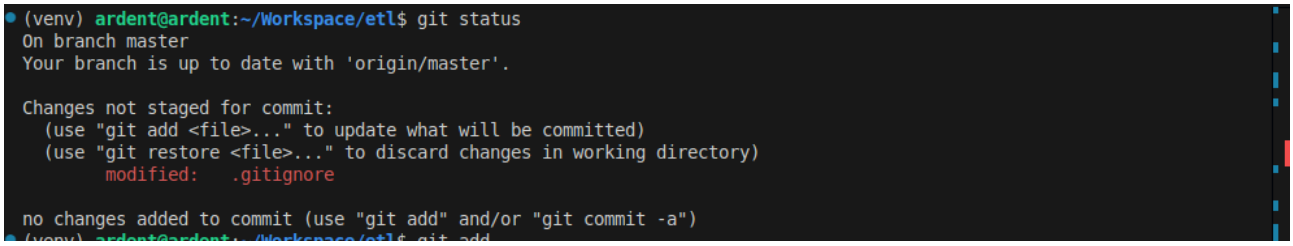
Make sure you use your own url. Do not copy mine. Also in our case we will be using branch master instead of main so no need to execute the second command in the third command write master instead of main. As that is what was created by default

Use below image as reference but use your own unique url

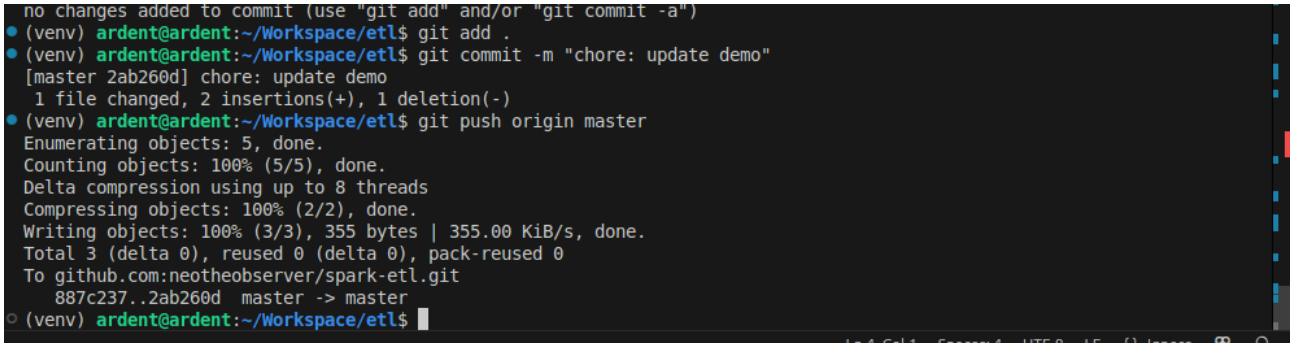


Congratulation now you have synced your local repository with Github/remote repository. Any time you make change make sure you add commit and then push

First view the changes using git status



Update them if required or add to .gitignore if you don’t want something that has been added or changed to be reflected in remote. Finally use add, commit and push to update the repo



This is the most basics of git commands there are many more things to be done but this should suffice to get started and working with git.