TESTBITS

VERSIONING CONTROL WITH GITHUB

GitHub Desktop

Presented By : Shahir

TOPICS

- What is Versioning Control?
- Why use Versioning Control?
- List of versioning control tool
- GitHub
- GitHub Flow
- Branches
- Commit
- Pull Request
- Git Bash
- Git Configuration
- Demo
- Practice

OBJECTIVES

- By the end of this session, you should be able to
  - Understand the concepts of versioning or source control
  - Understand the basics of GitHub
  - Explain about GitHub flow
  - Practice GitHub usage in a project with GitHub Desktop

PRE-REQUISITES

- Participants should have
  - Registered for a GitHub account (it's free)
  - Downloaded and installed
    - GitHub Desktop - https://desktop.github.com/
    - Git - https://git-scm.com/downloads

VERSIONING CONTROL

Brief overview of Versioning Control

WHAT IS VERSIONING CONTROL?

A component of Software Configuration Management

Also known as Revision Control, Source Control

Manage changes of

Documents

Computer programs/software

etc

WHY USE VERSION CONTROL?

Easier to track changes

Error can be fixed more effectively as there are historical data

Team members can collaborate effectively

EXAMPLE OF VERSIONING CONTROL TOOL

- **Local data model**
  - Revision Control System (RCS)
  - Source Code Control System (SCCS)
- **Client-server model**
  - Concurrent Version System (CVS)
  - Suberversion (SVN)
  - AccuRev
- **Distributed model**
  - Git
  - Microsoft Visual Studio Team Services (VSTS)
  - And many more..

GitHub

Visual Studio Team Foundation Server

CVS

git

HelixCore

GITHUB

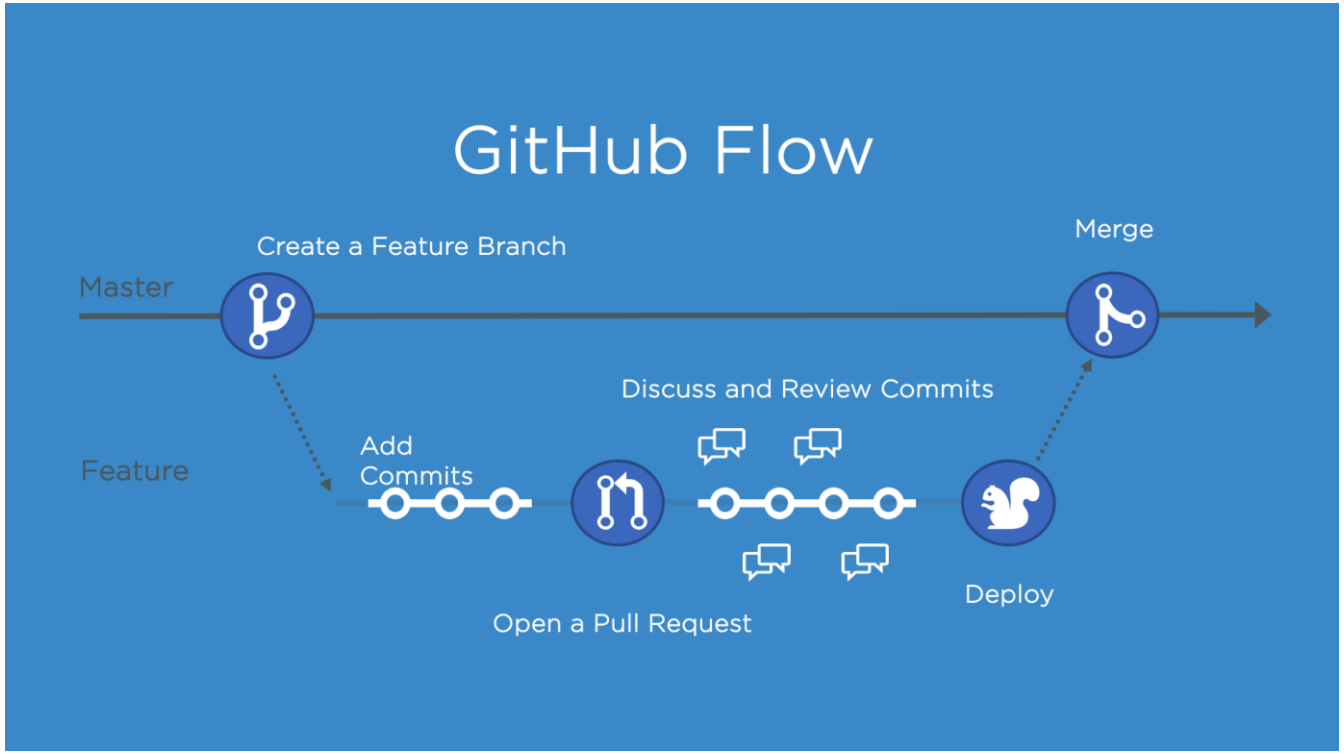
Introduction to GitHub

GITHUB

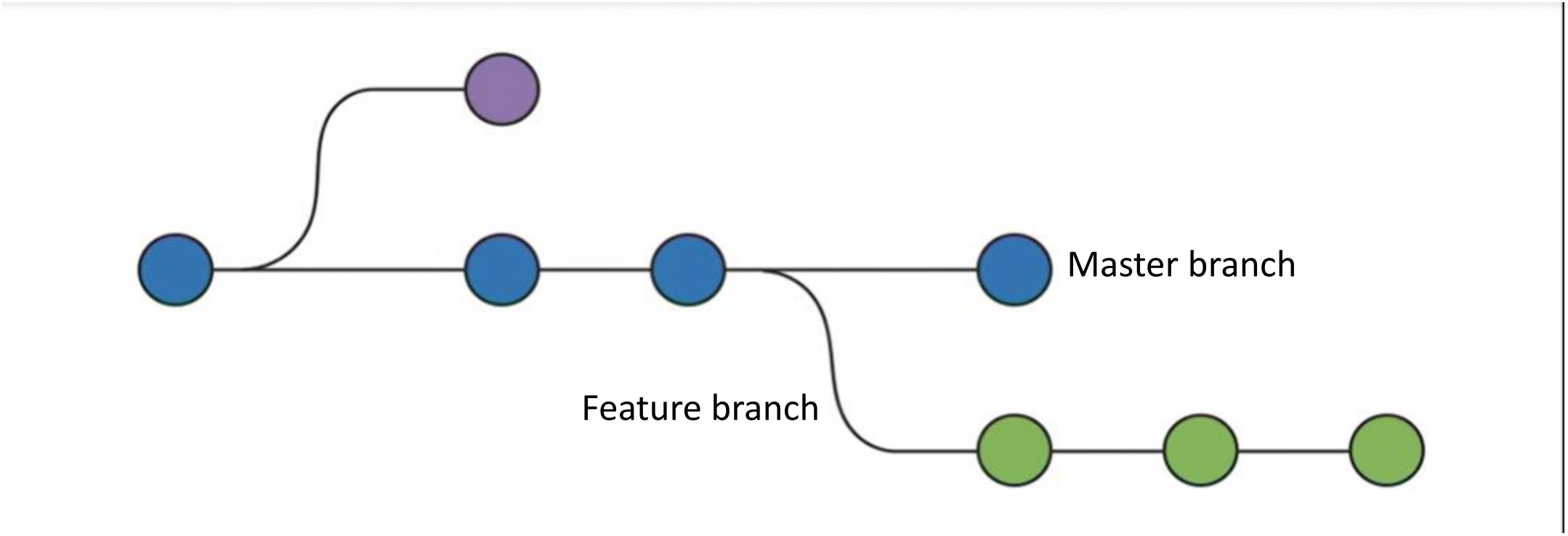
- GitHub is a Git repository hosting service
- It adds a lot of its own features
- Provides a GUI over traditional CLI
- Provides access control
- Other collaboration features:
  - Wikis
  - Basic task management tools

GITHUB FLOW

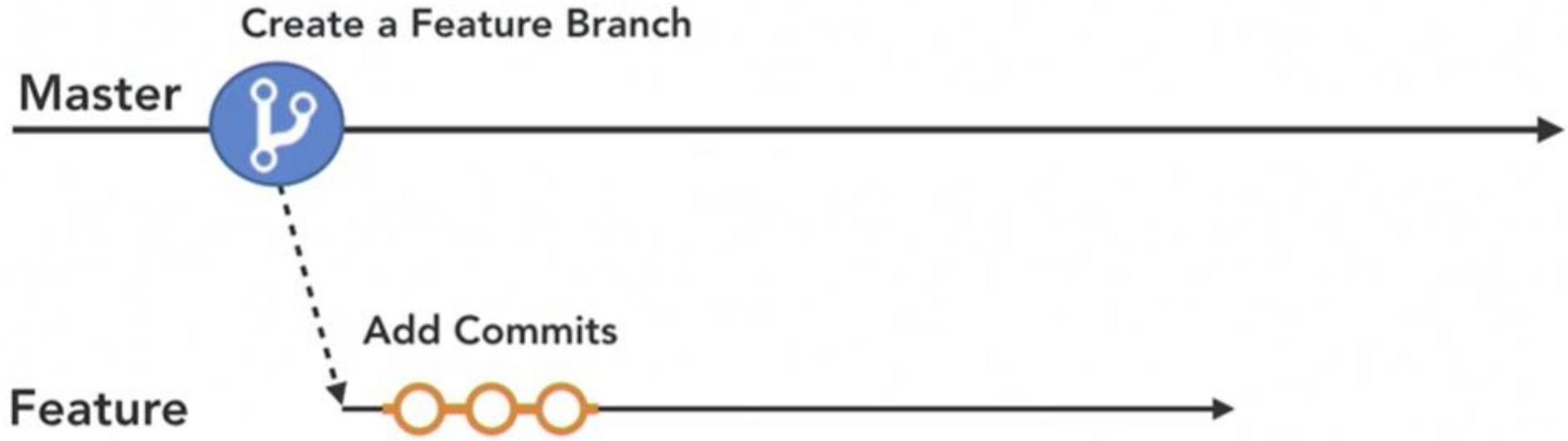
- A lightweight, branch-based workflow.



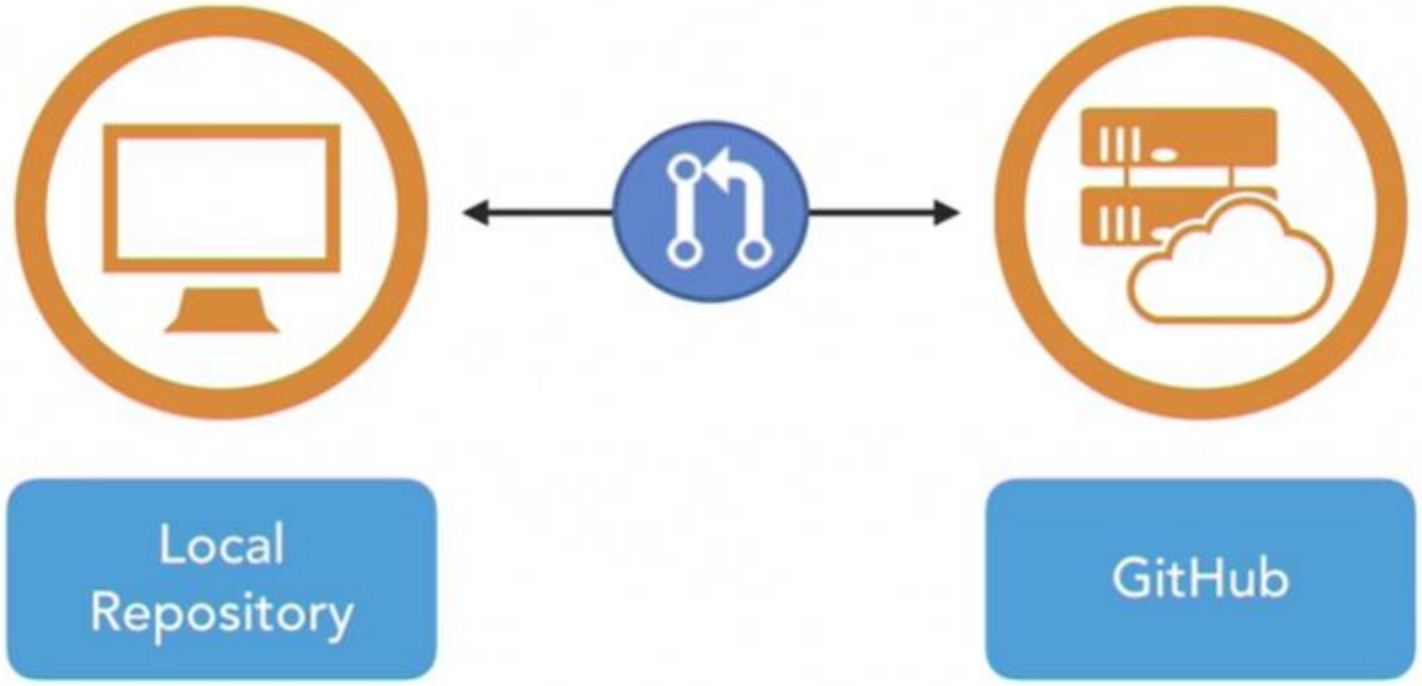
BRANCHES



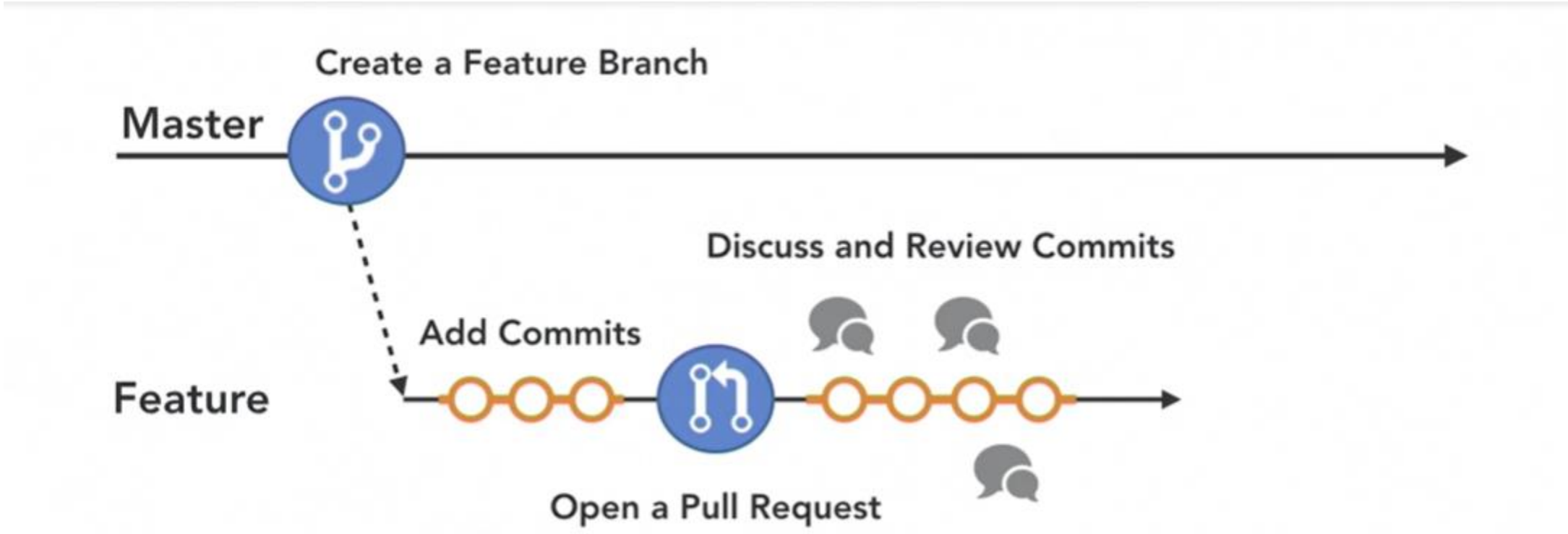
COMMITS



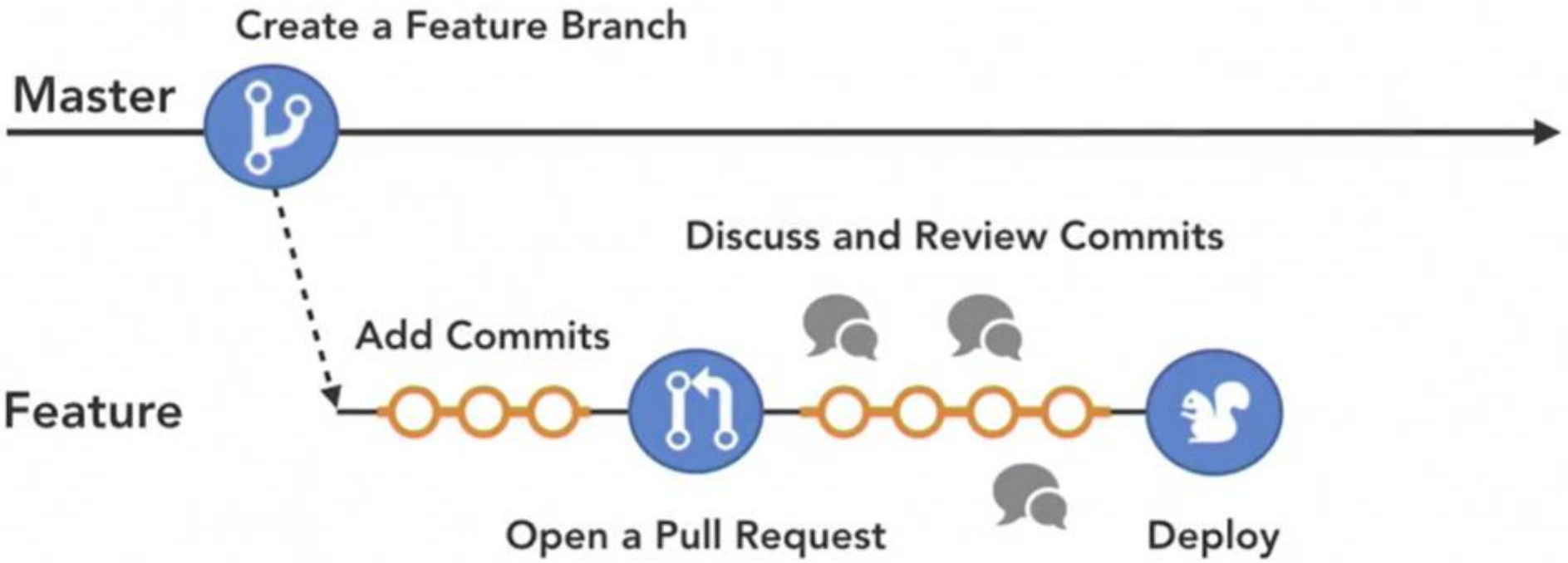
PULL REQUEST



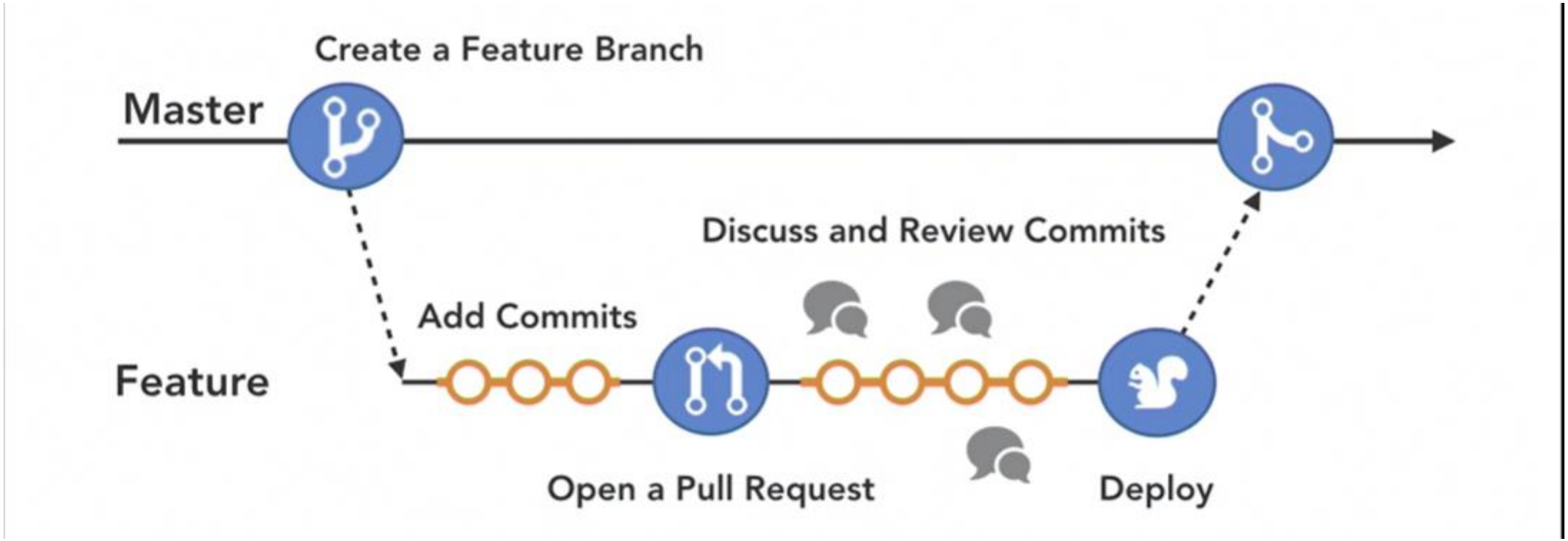
REVIEW PULL REQUEST



DEPLOY PULL REQUEST



MERGE PULL REQUEST



DIFFERENT WAYS OF USING GIT/GITHUB

Command line interface

Git Bash

GUI Clients

GitHub Desktop
SourceTree
TortoiseGit
Etc..

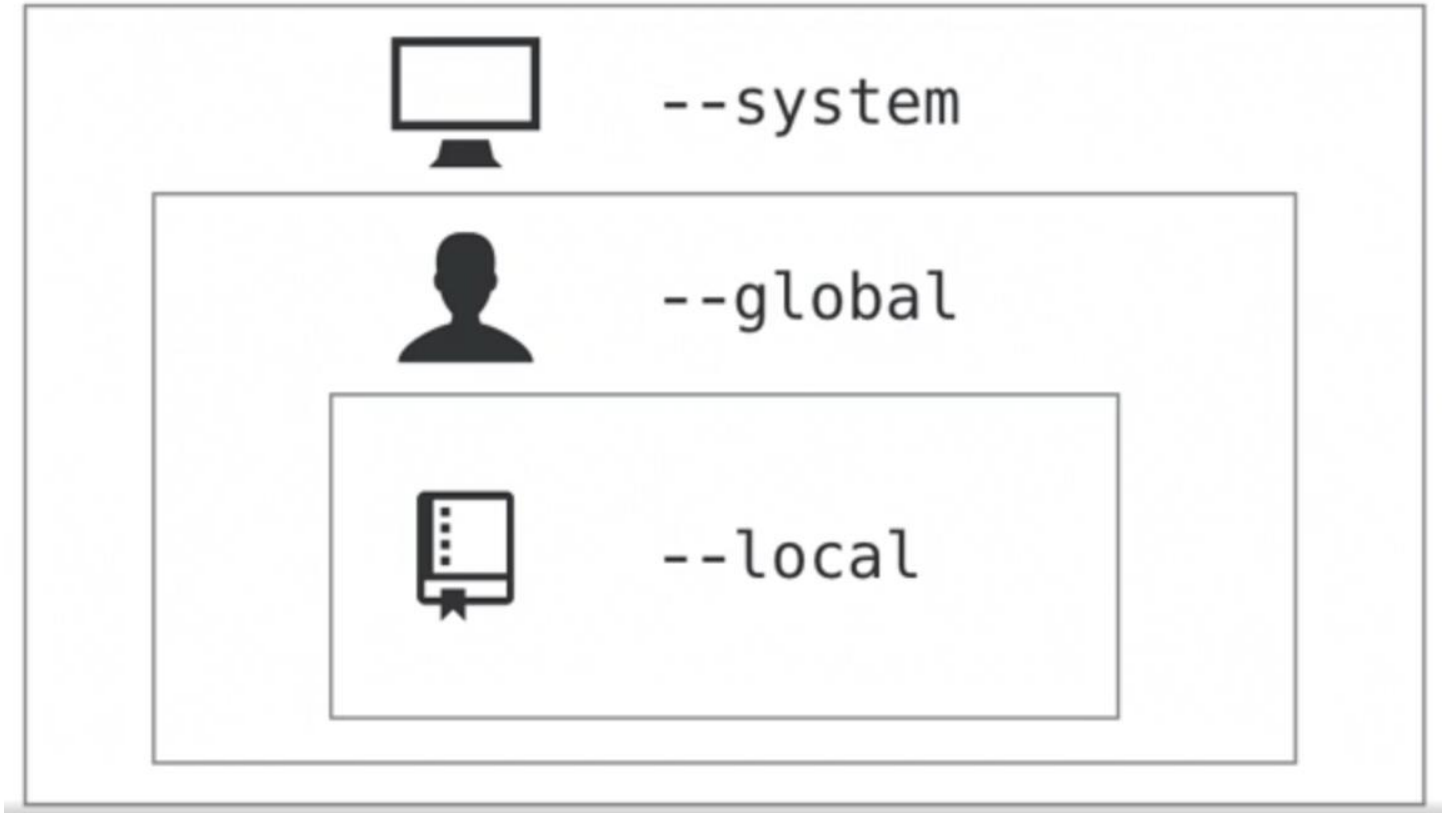
IDEs

Visual Studio
Eclipse
Etc..

BASIC LIST OF COMMANDS

pwd	• print working directory
cd	• change directory
dir	• list down content in drectory
copy con	• create an empty file
mkdir	• create an empty folder
clear	• to clear the console screen
git – version	• check if Git installed
git pull	• update the local repository to follow latest version in remote repository (GitHub)
git add	• to stage changes before commit
git commit	• commit changes to local branch
git push	• push changes done on a local branch to remote branch
git checkout -- <file>	• to undo changes on a file

GIT CONFIGURATION LEVEL



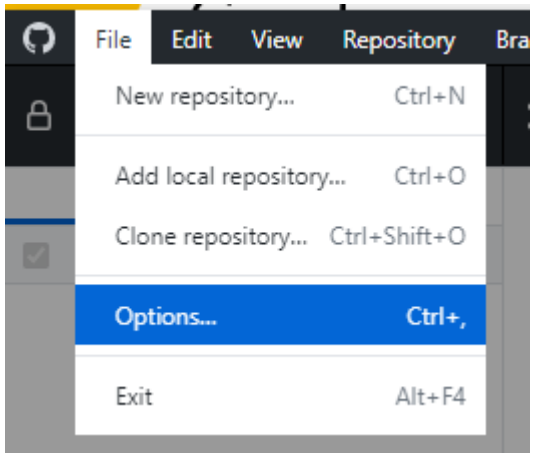
CONFIGURE USERNAME & EMAIL

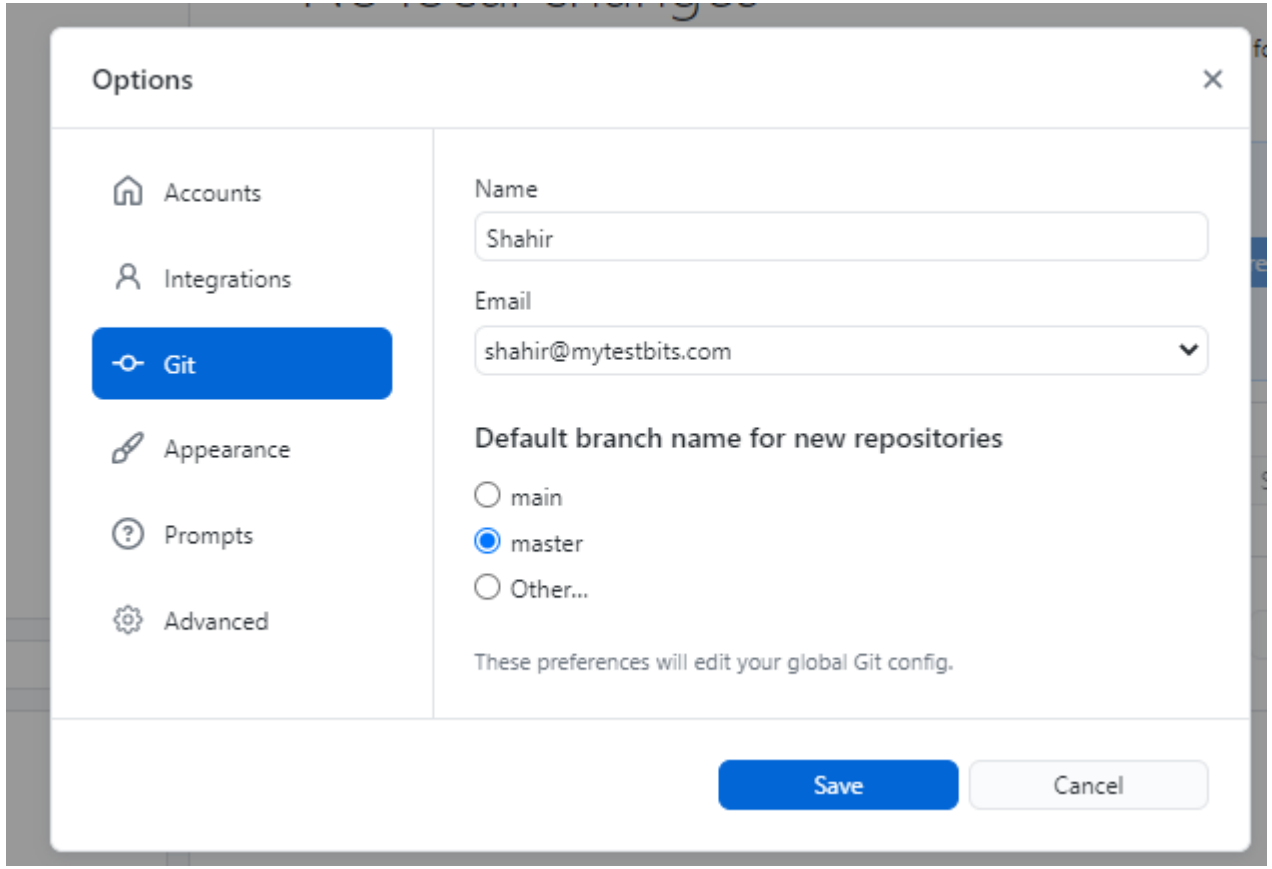
```
git config --global user.name ”username"

git config --global user.email "email@example.com"
```

CONFIGURE USERNAME & EMAIL – GITHUB DESKTOP

- File > Options
- On ‘Options’ popup, select ‘Git’





CONFIGURE DEFAULT TEXT EDITOR

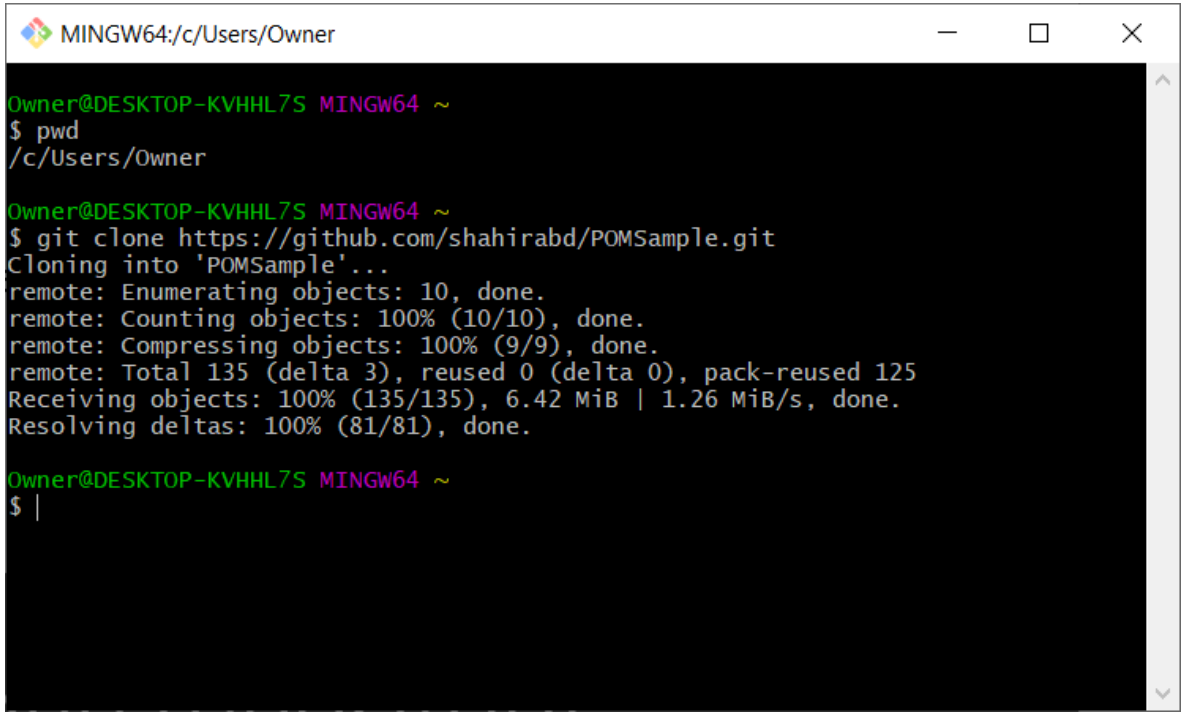
```
git config --global core.editor "'C:\Program
Files\Notepad++\notepad++.exe'"
```

DEMO

GitHub Desktop

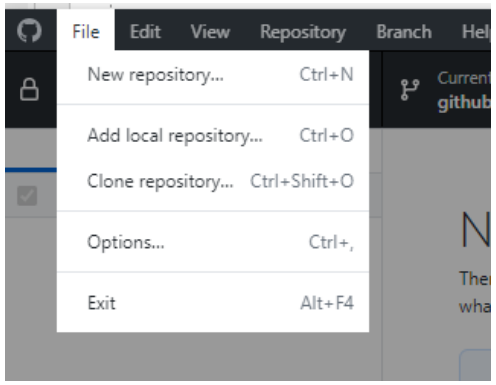
CLONE REPOSITORY - CLI

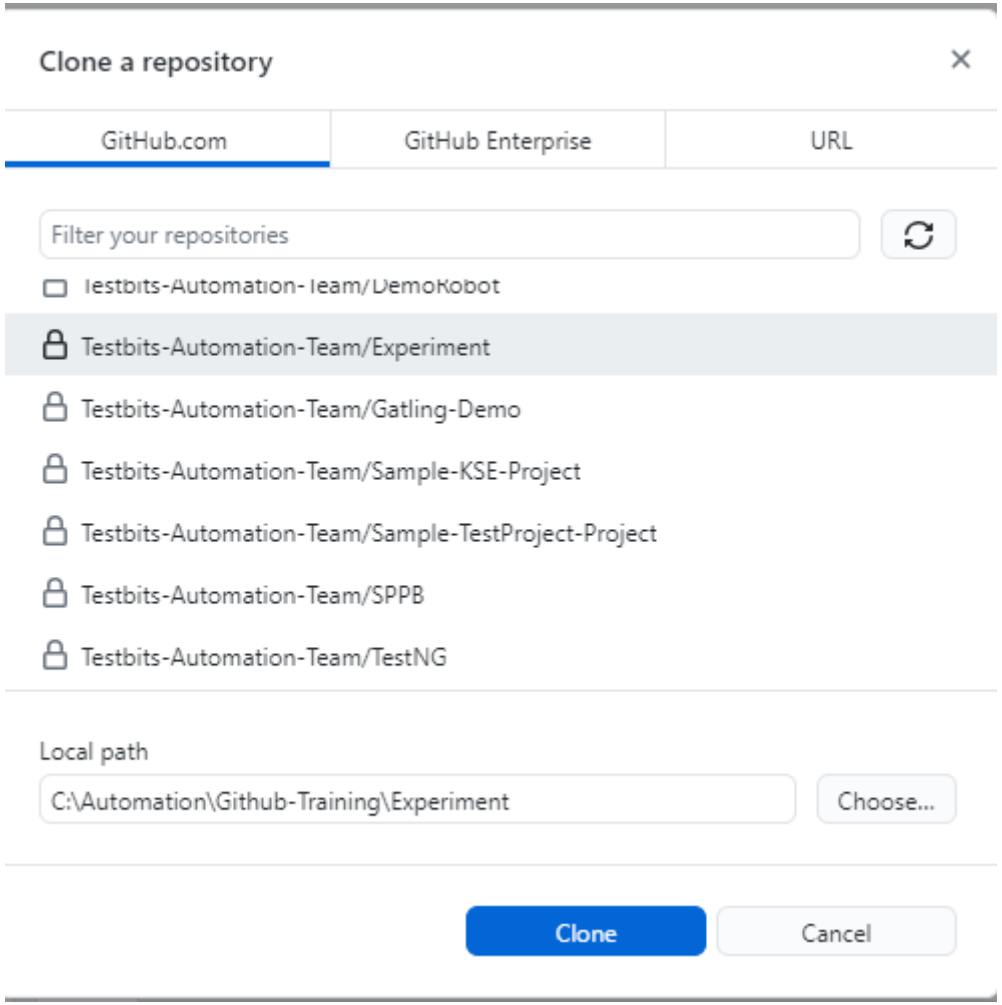
```
git clone <repository URL>
```



CLONE REPOSITORY – GITHUB DESKTOP

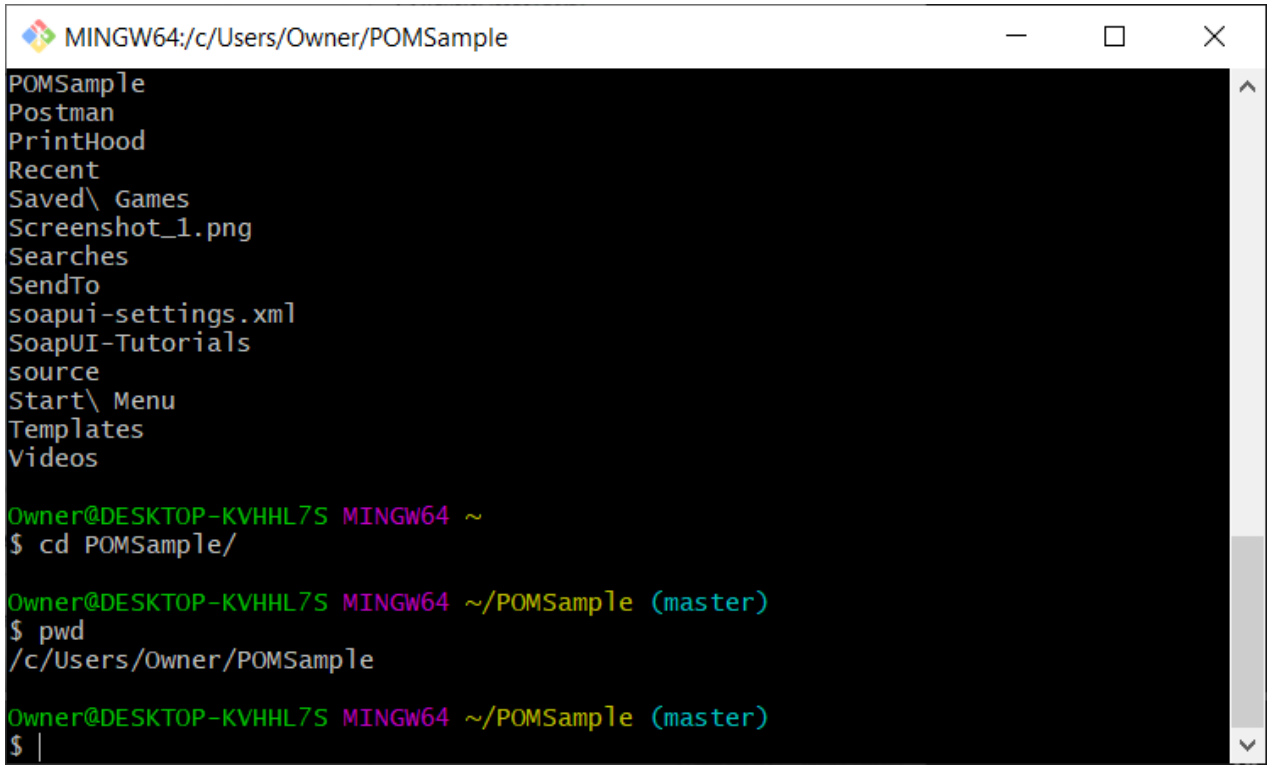
- File > Clone repository
- Select which repository you wish to clone to your local machine
- Select the local path of your repository
- Click the ‘Clone’ button





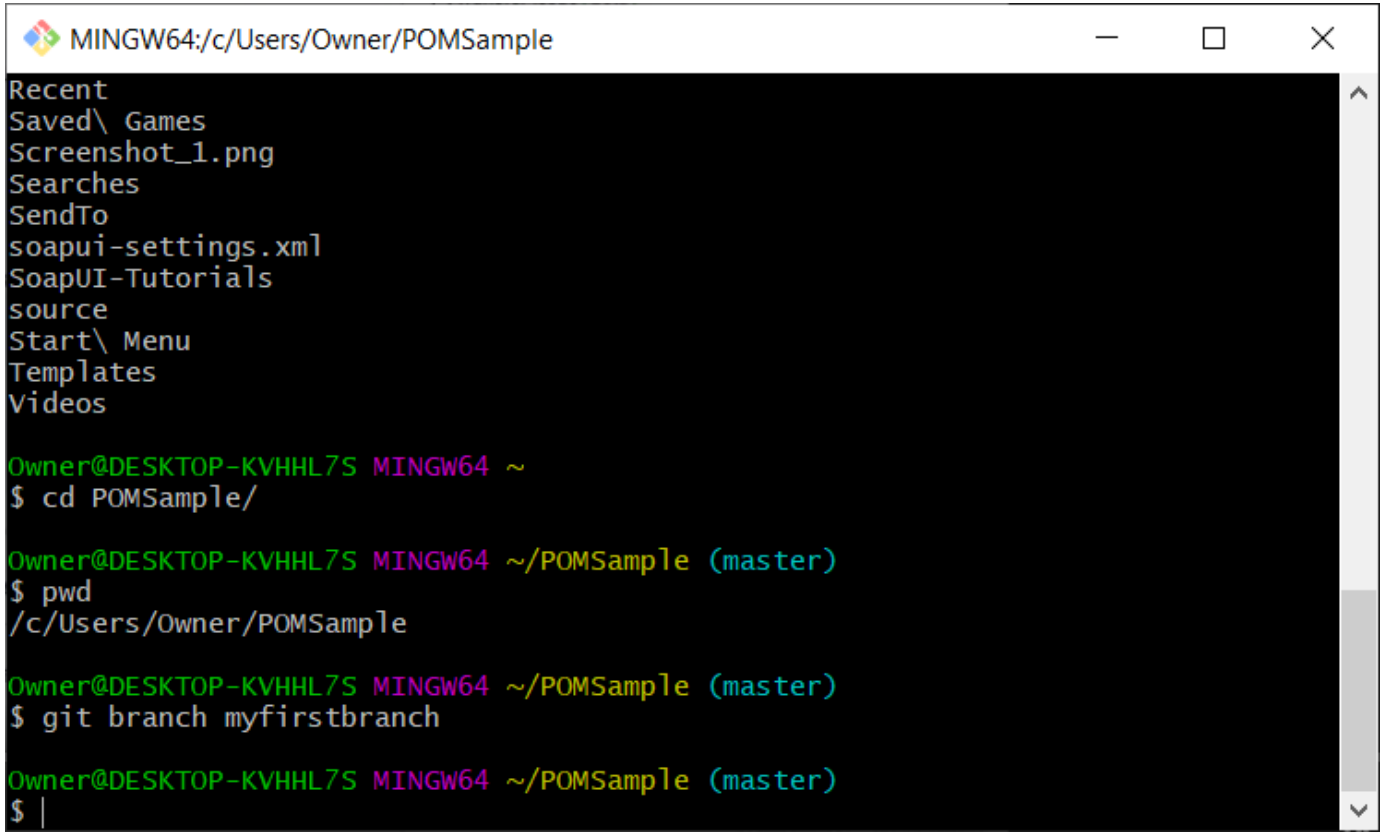
CHANGE DIRECTORY - CLI

- `cd <folder_name>`



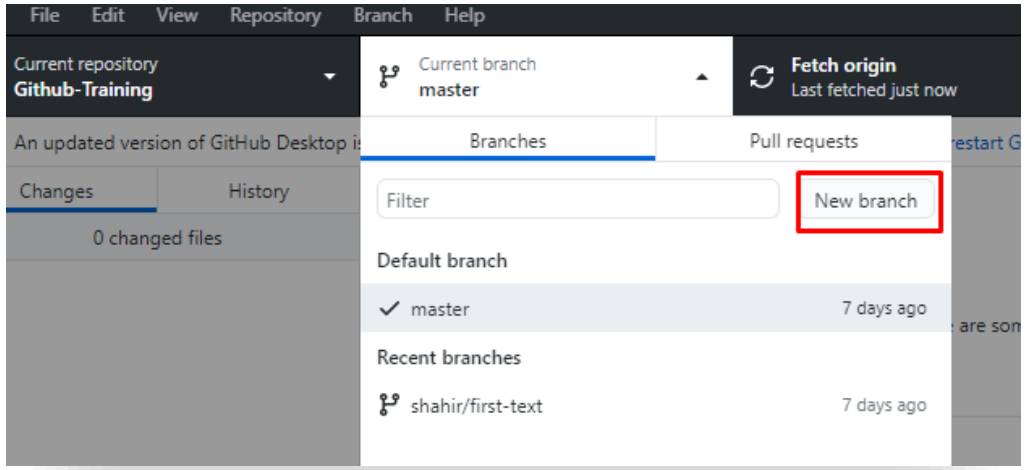
CREATE BRANCH - CLI

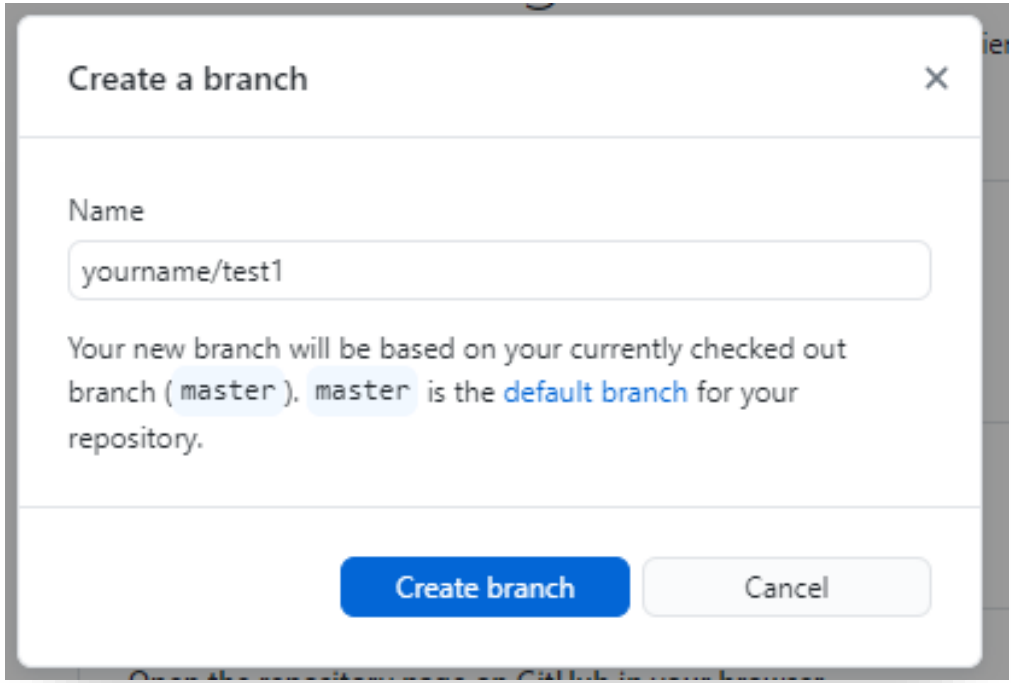
- `git branch <branch_name>`
- `git branch --all`



CREATE BRANCH – GITHUB DESKTOP

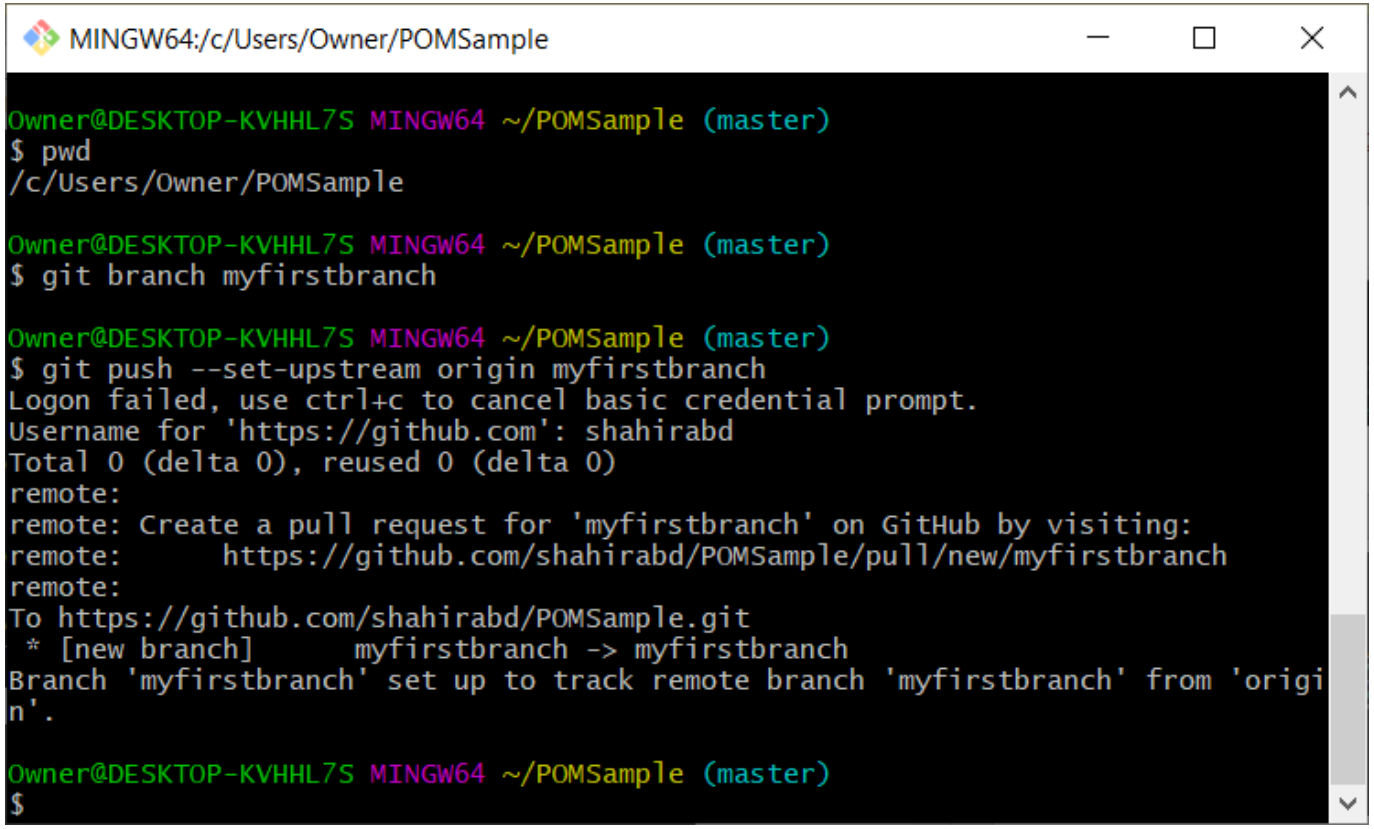
- Click on ‘Current branch’ > ‘New branch’





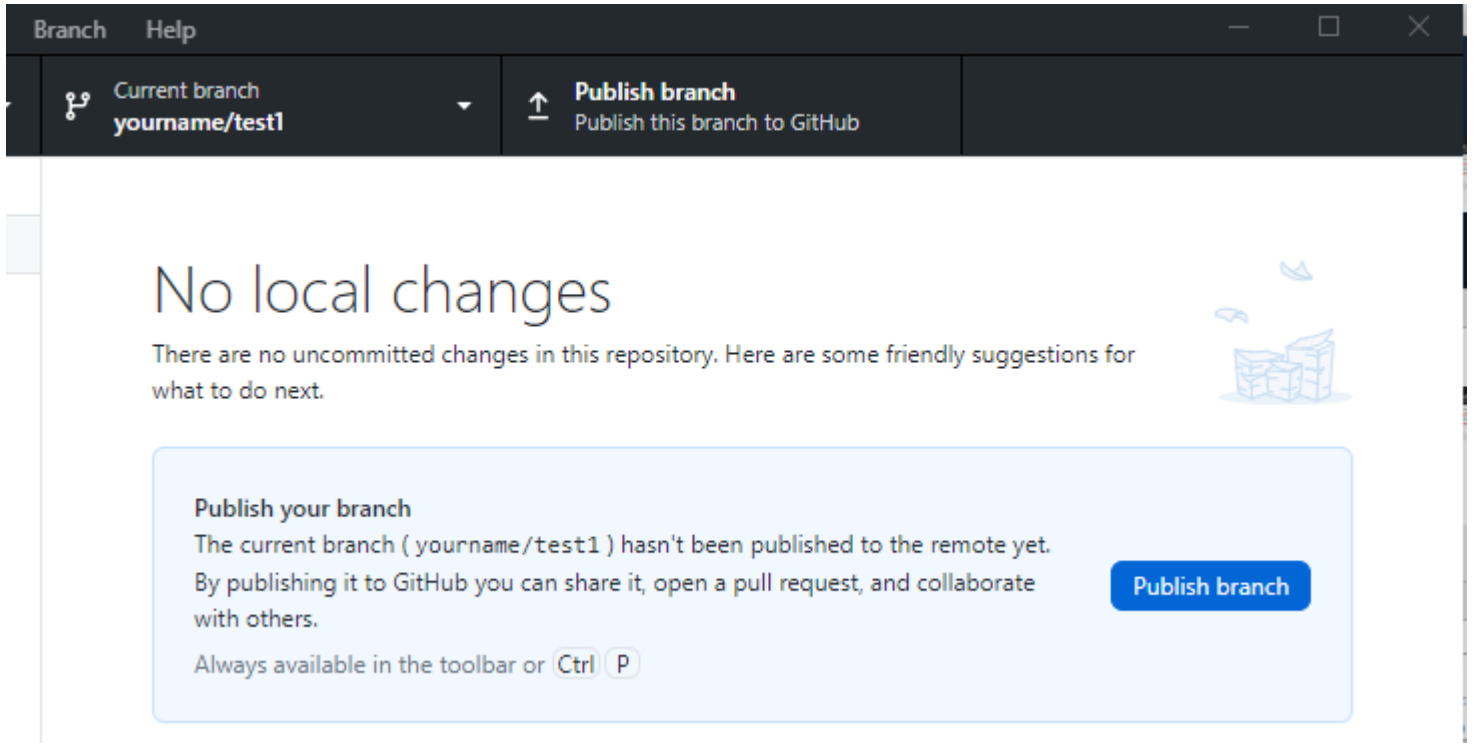
PUBLISH THE BRANCH – CLI

- git push --set-upstream origin <branch_name>
- When prompted, enter your GitHub username and password



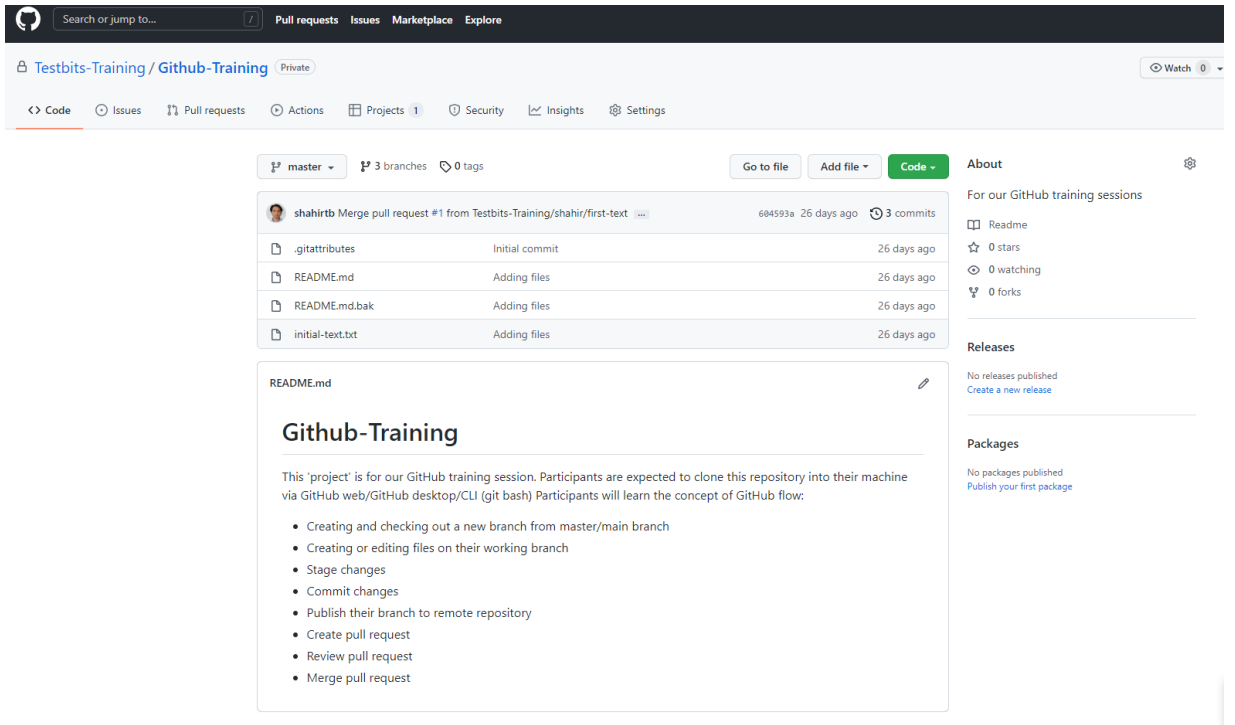
PUBLISH THE BRANCH – GITHUB DESKTOP

- Click on ‘Publish branch’ button



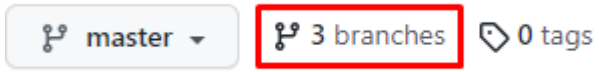
BROWSE TO GITHUB PORTAL

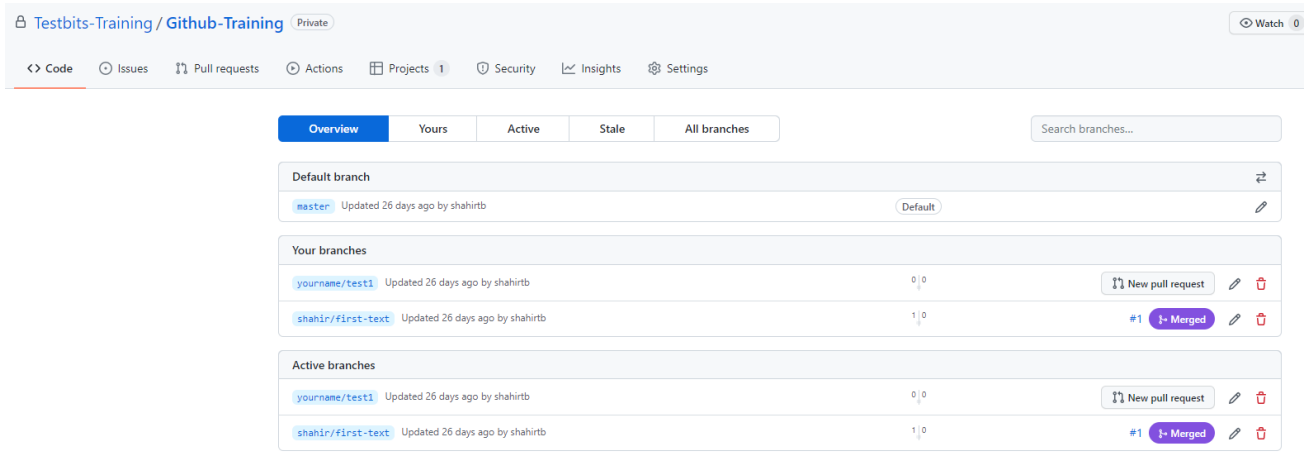
- Since we are not making any changes to the project, there are no changes displayed



BROWSE TO GITHUB PORTAL

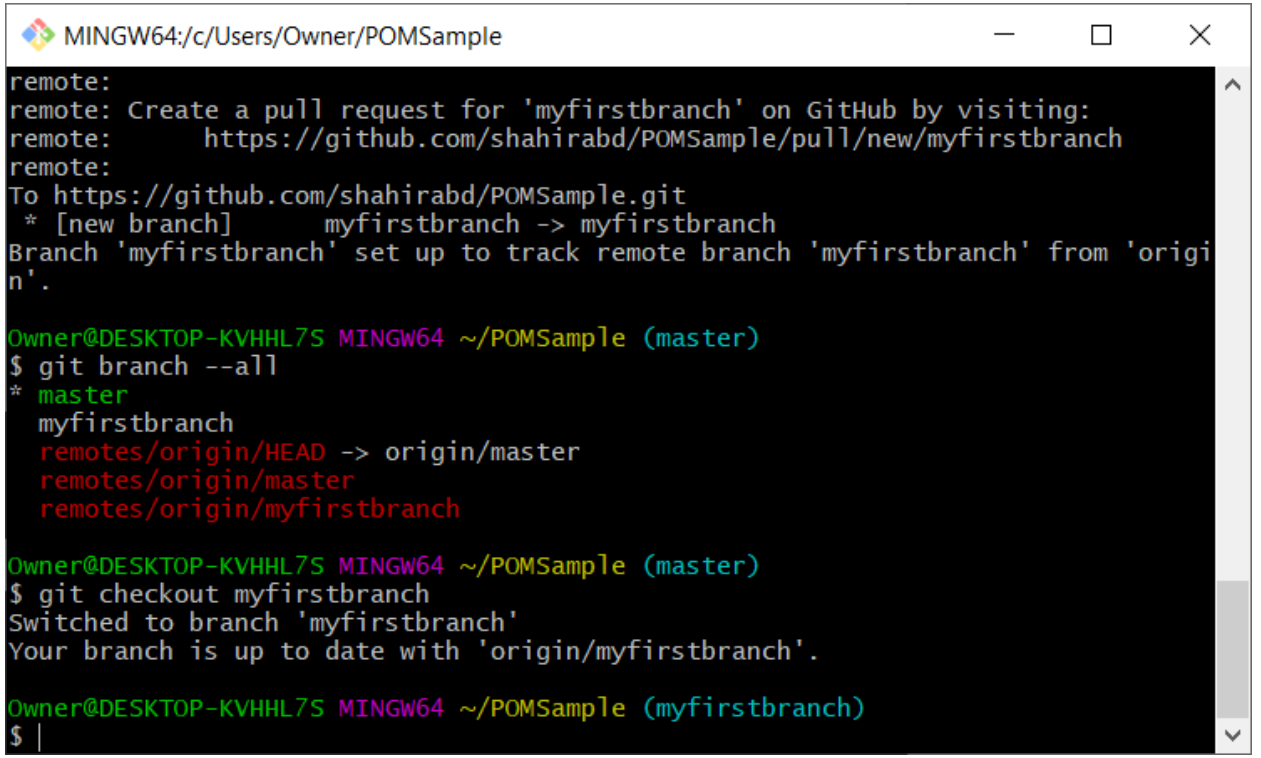
- View your branch and other branches in the repository
- Click on the ‘branches’ link





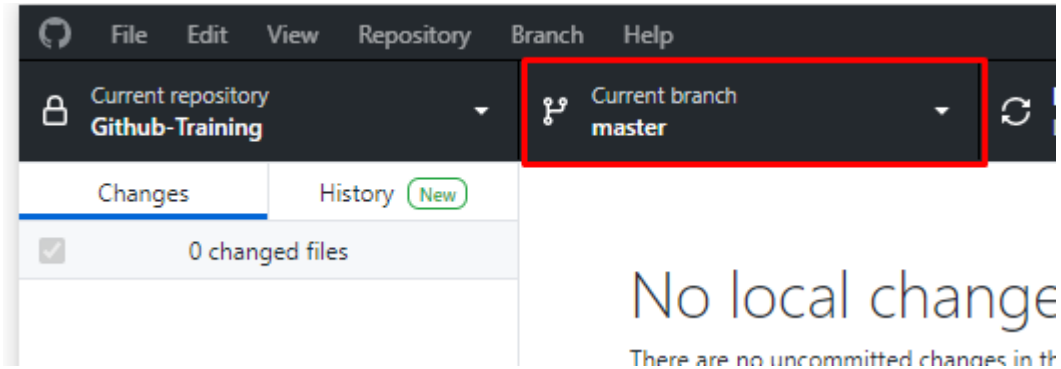
CHECKOUT BRANCH - CLI

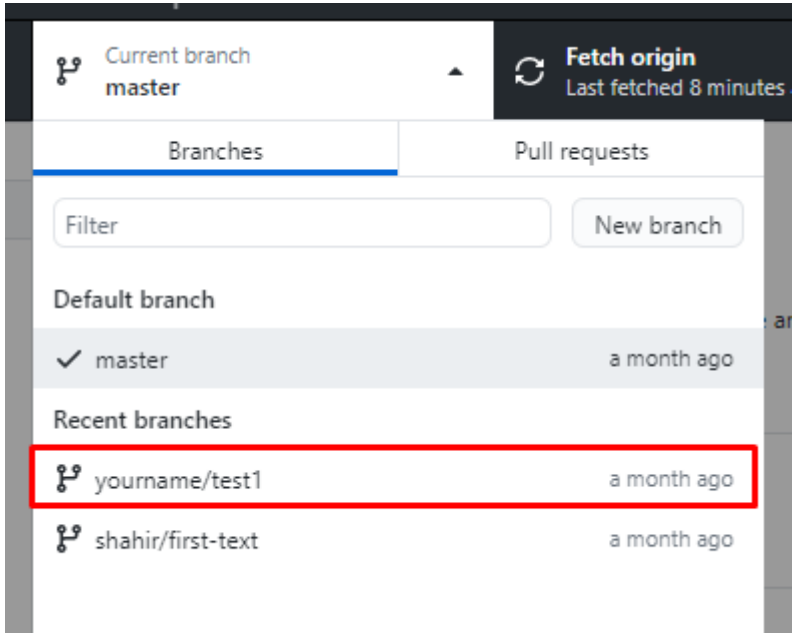
- git checkout <your_branch_name>



CHECKOUT BRANCH – GITHUB DESKTOP

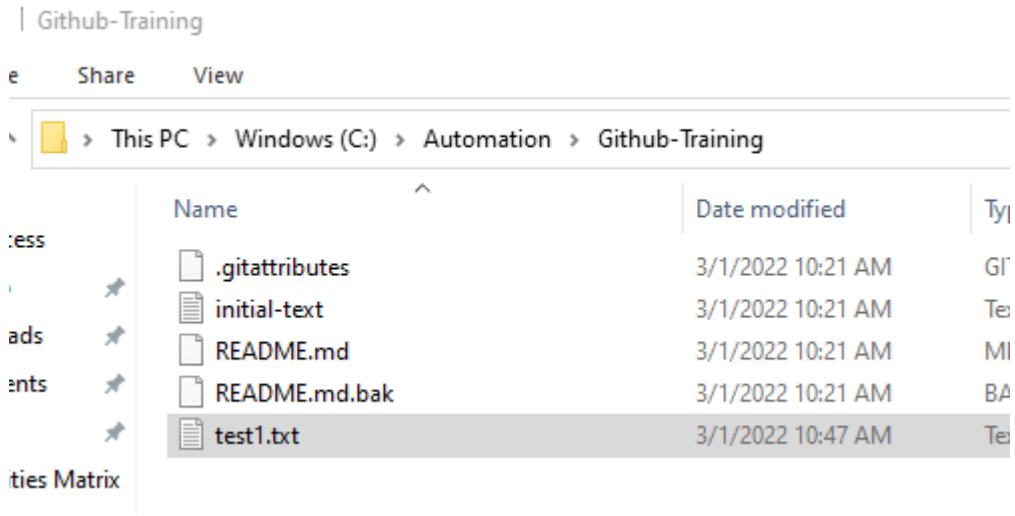
- Click on ‘Current branch’ > ‘yourname/test1’ branch or your branch name





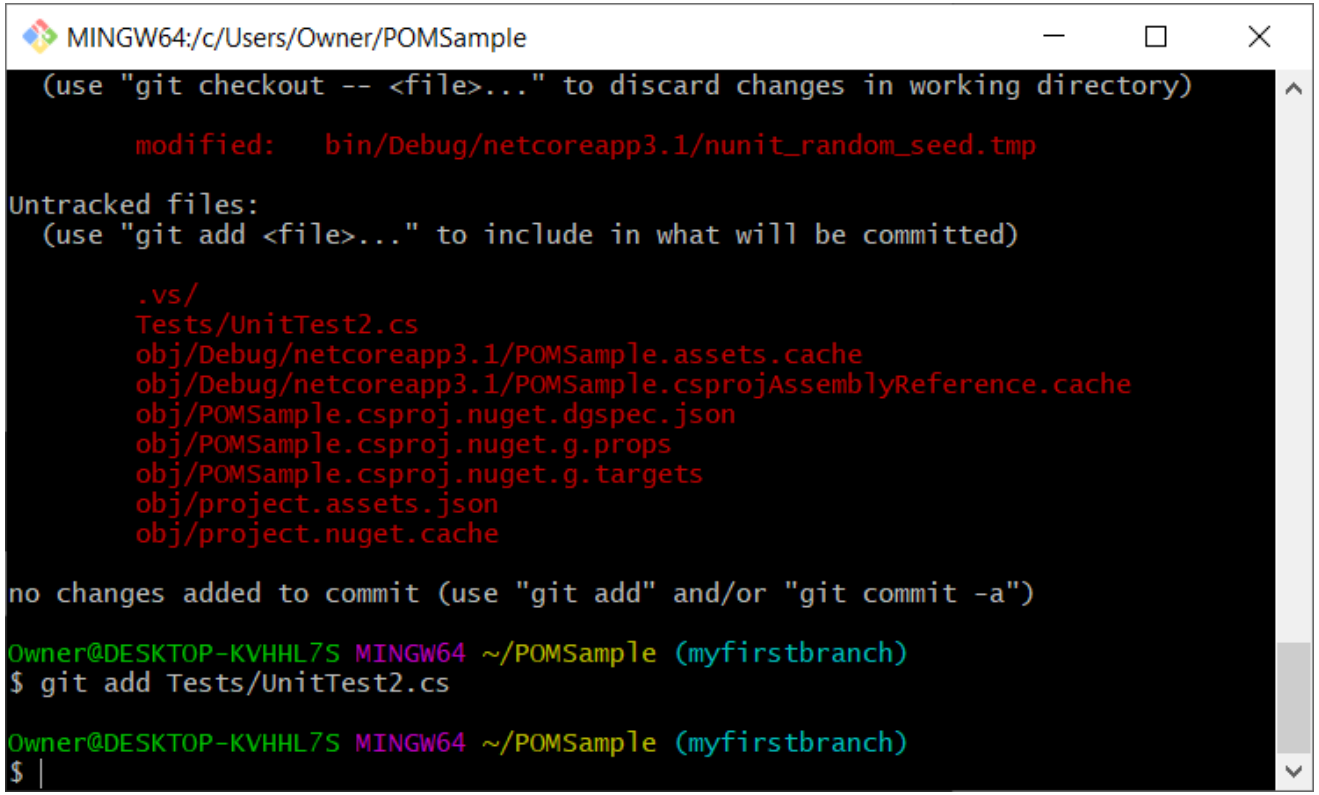
ADD A NEW FILE TO THE PROJECT FOLDER

- Open the project folder in in Windows Explorer
- Add a text file under ‘Github-Training’ folder
- Open the new text file and add some text in there



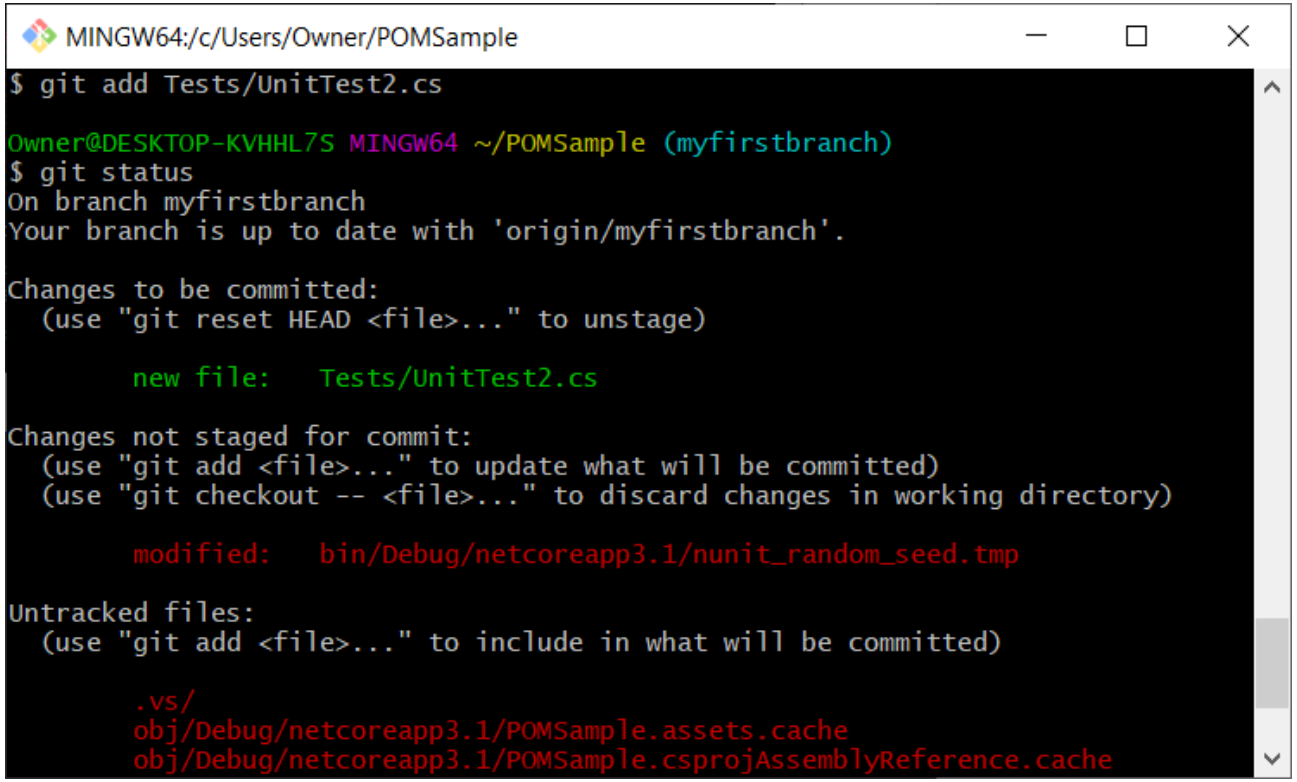
STAGE THE FILE - CLI

- Go back to Git Bash console
- To view the status
  - git status
- To stage the file
  - git add <filename>



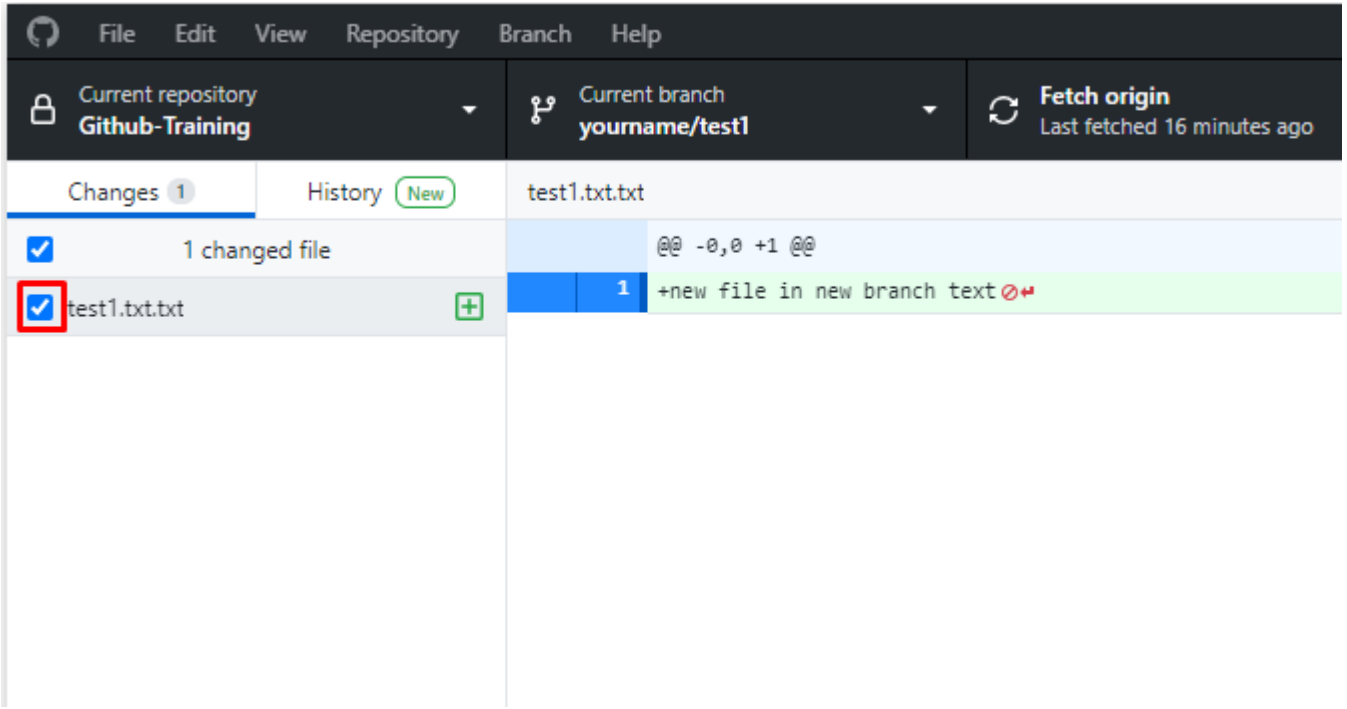
CHECK STATUS - CLI

- To view the status
  - git status
- Notice that the file we staged is in green font



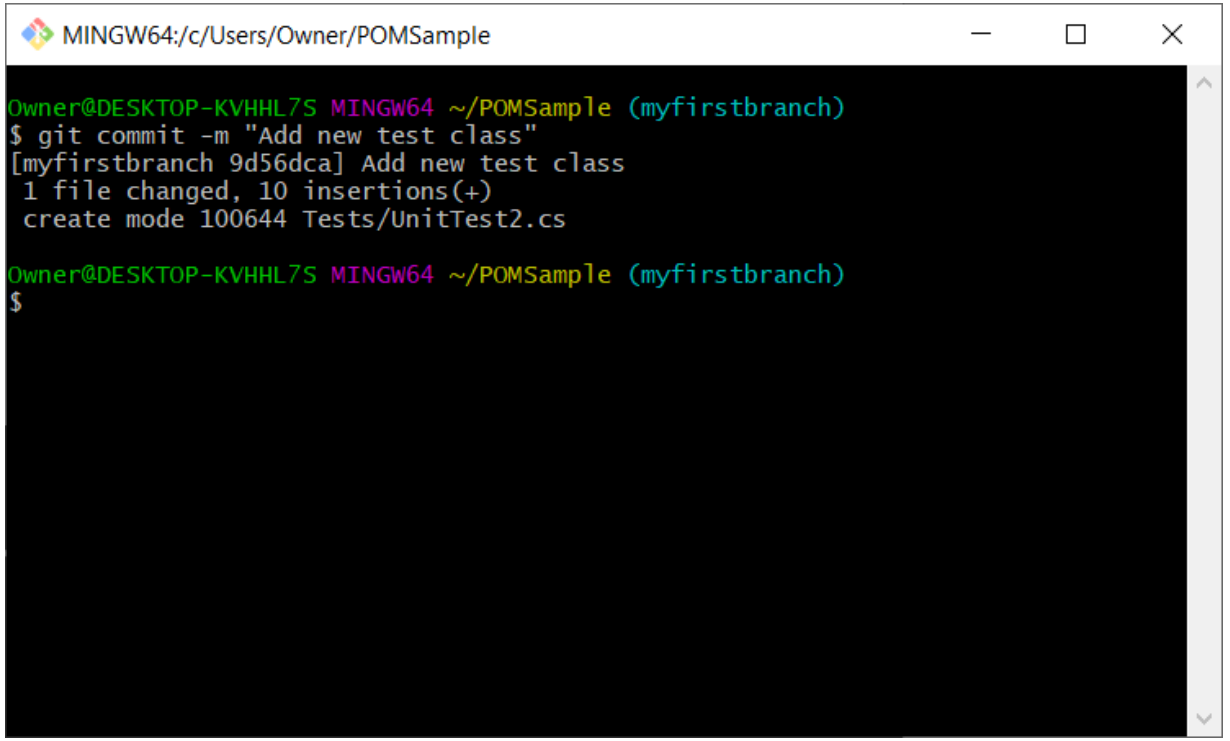
STAGE THE FILE – GITHUB DESKTOP

- Tick the checkbox on the left to stage the file



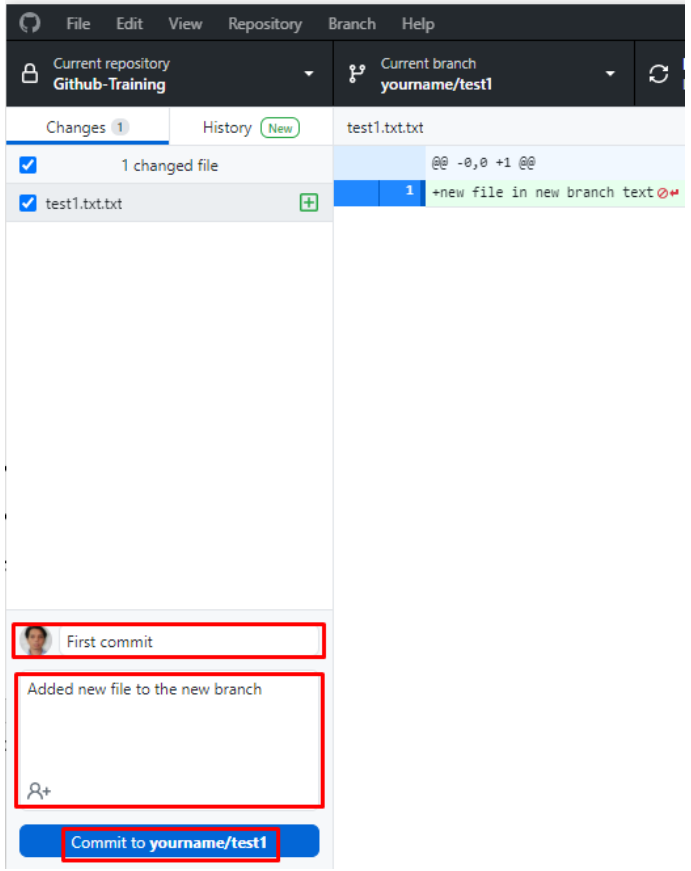
COMMIT – CLI

- git commit -m “file_name"



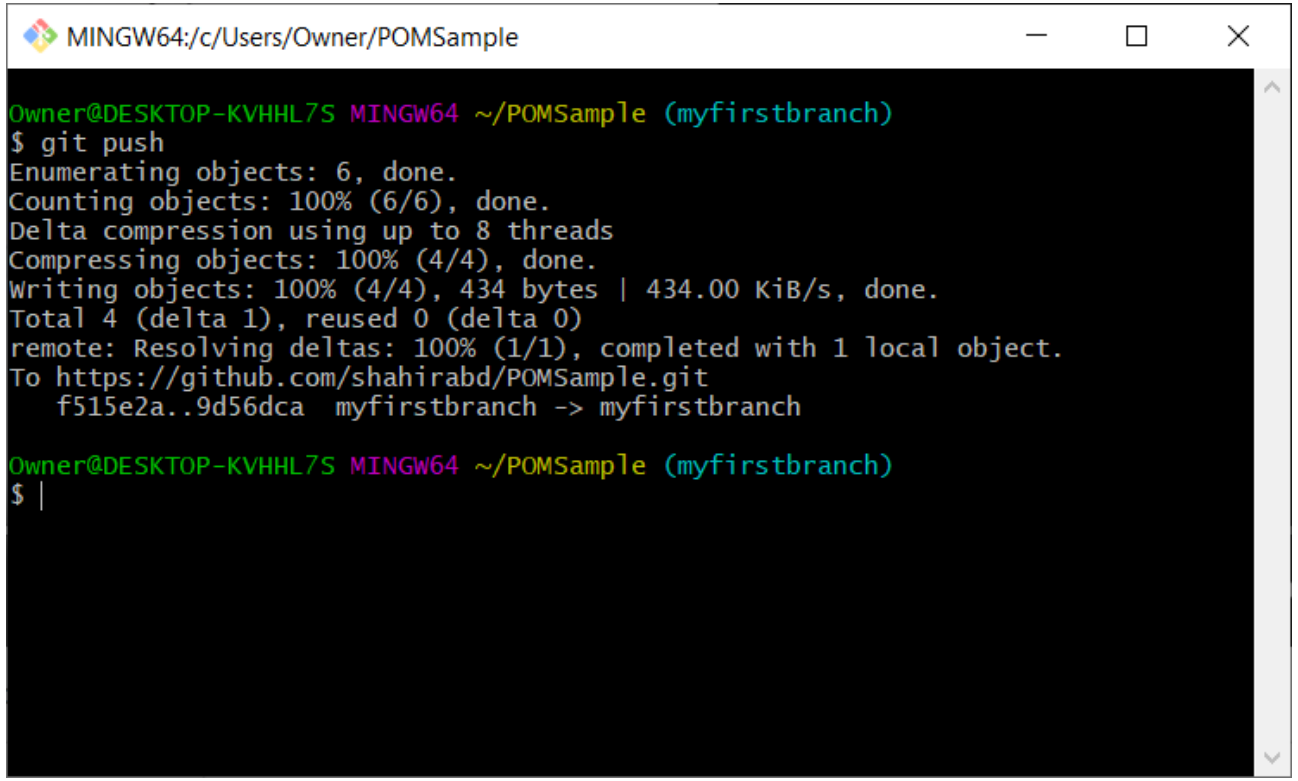
COMMIT – GITHUB DESKTOP

- Enter the commit title and summary, then click on the ‘Commit to..’ button



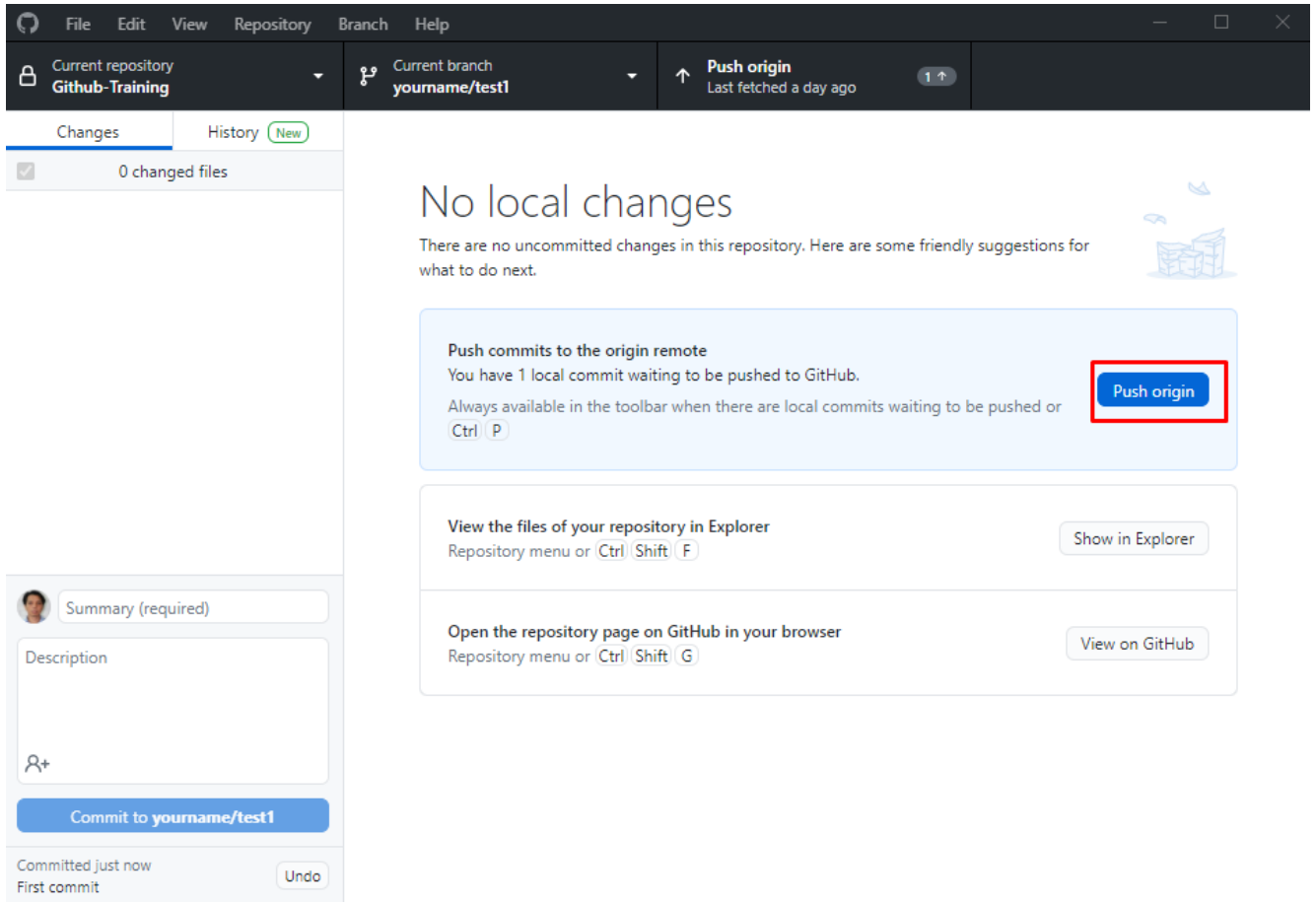
PUSH – CLI

- git push



PUSH – GITHUB DESKTOP

- Click on the ‘Push origin’ button



DEMO
RECAP

GitHub Desktop

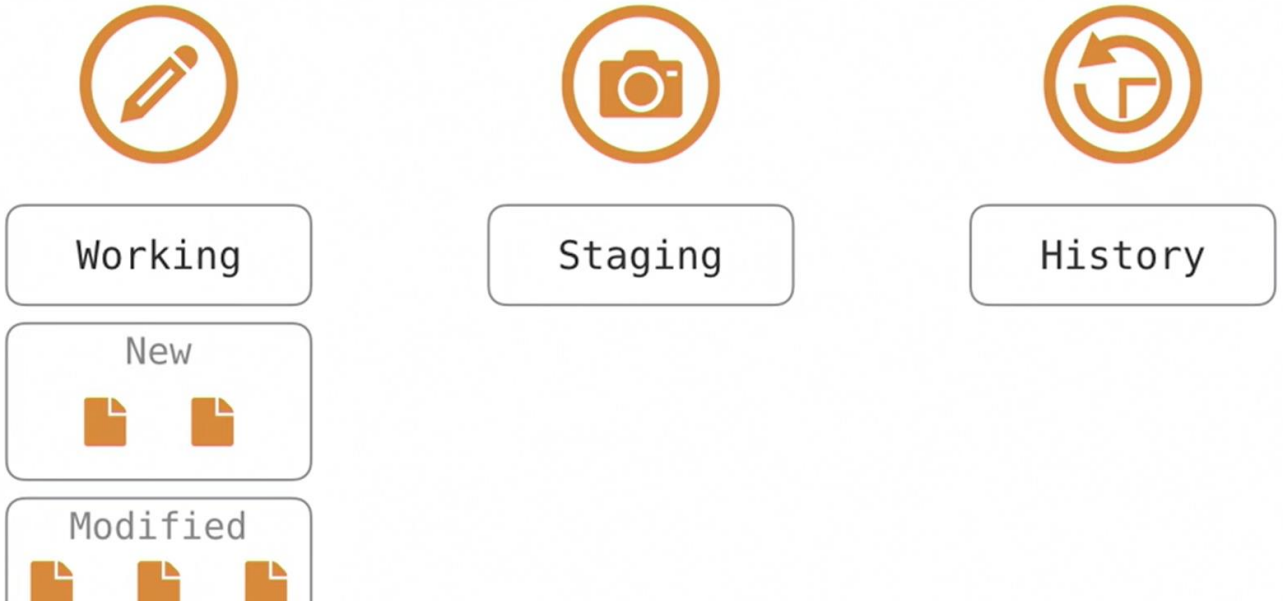
STATES OF A FILE

Working

Staging

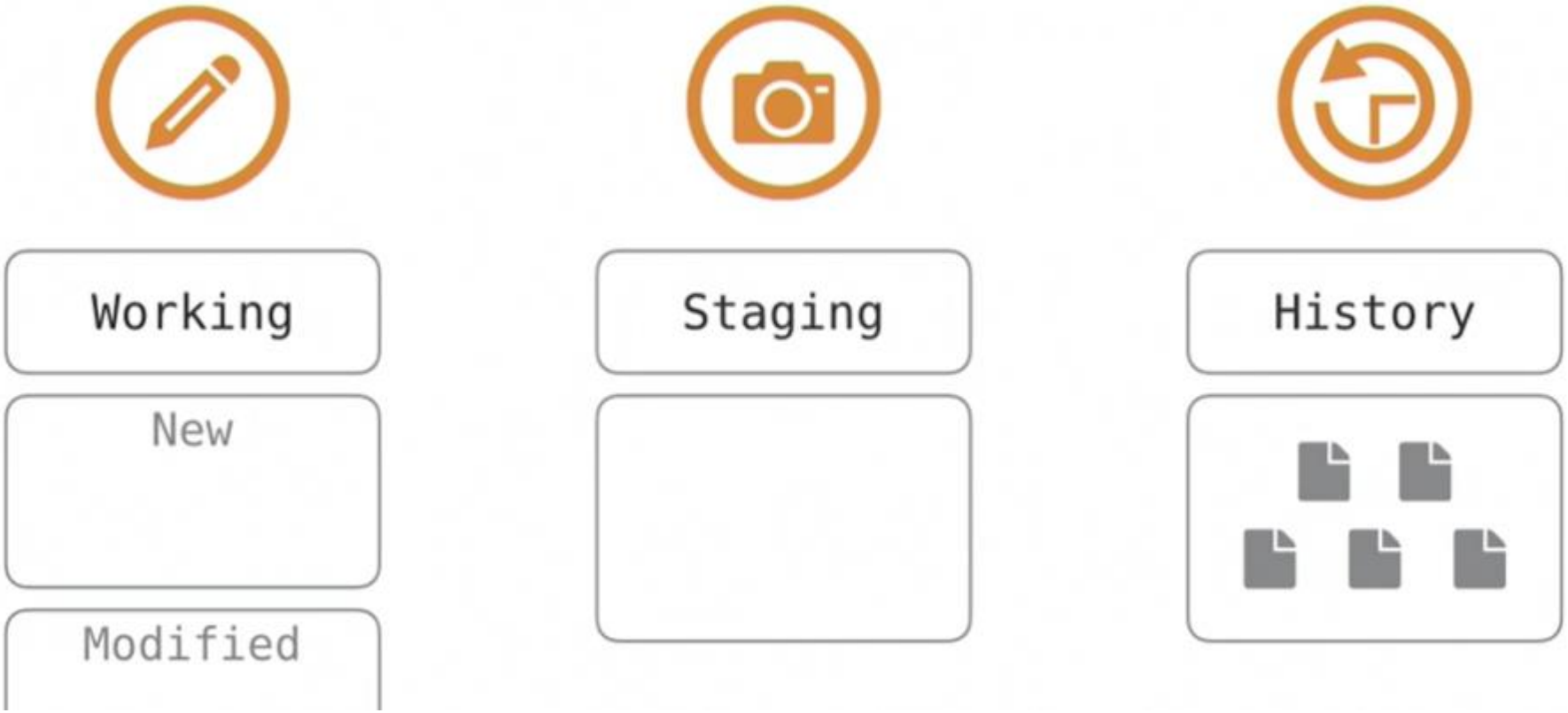
History

WORKING DIRECTORY



STAGING AREA

COMMITTED HISTORY

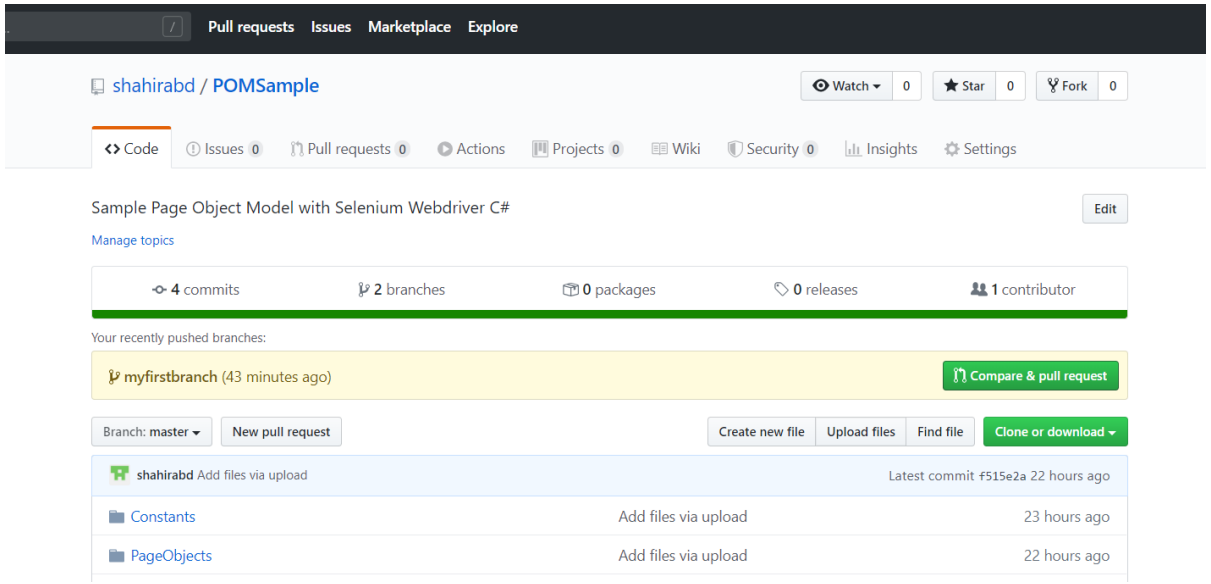


DEMO

Pull Requests

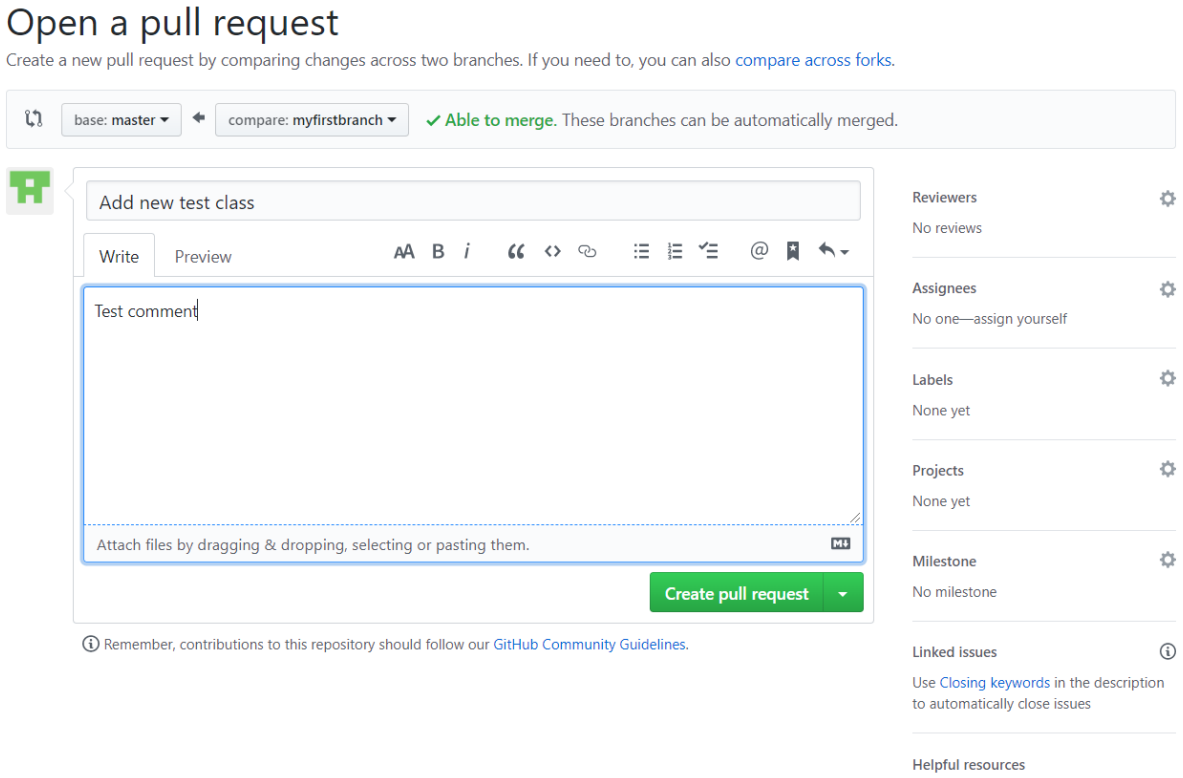
CREATE PULL REQUEST

- Go to your repository in GitHub
- You can see the new branch and the green button ‘Compare and pull request’, click on it



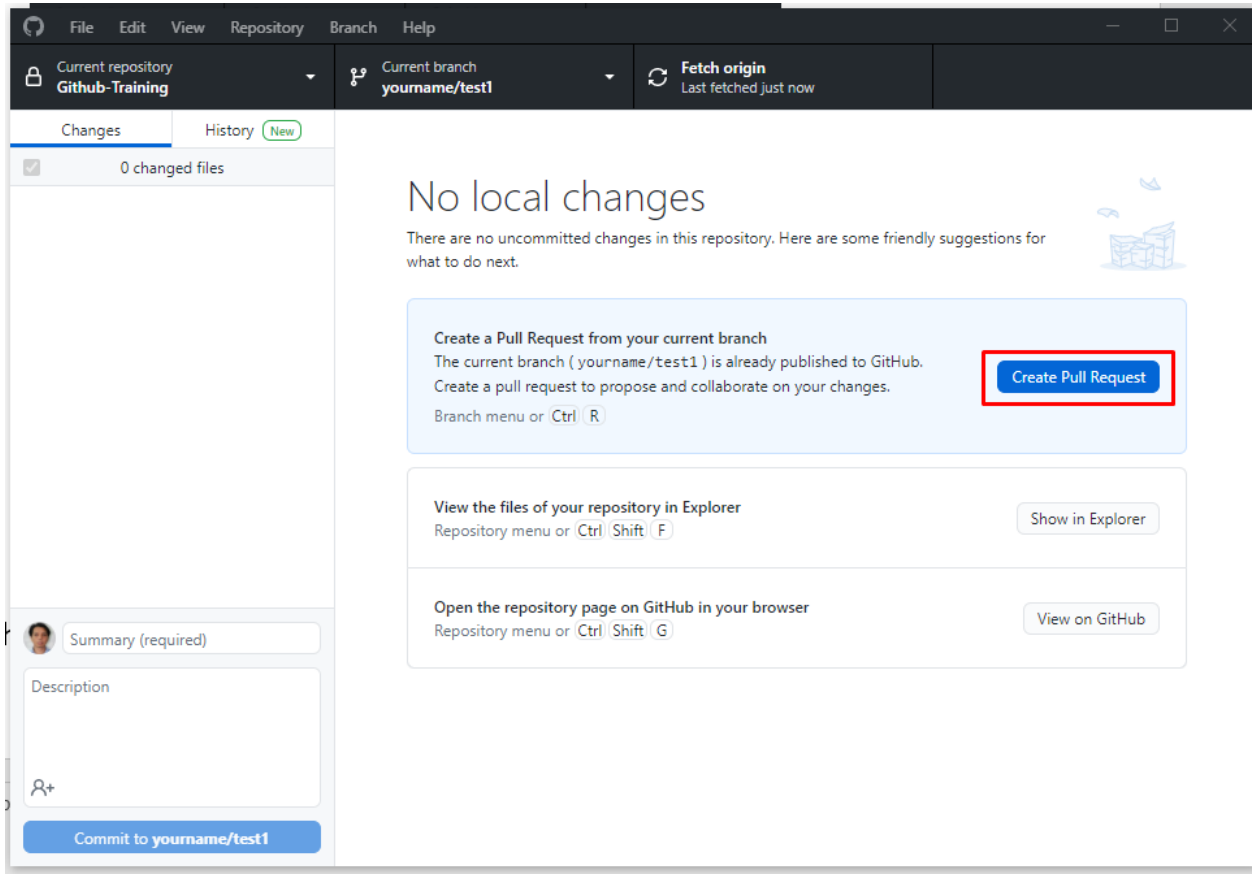
CREATE PULL REQUEST

- Once clicked, this page will be opened
- We can add comment and perform other actions like adding reviewers etc
- Make sure the base branch is ‘master’



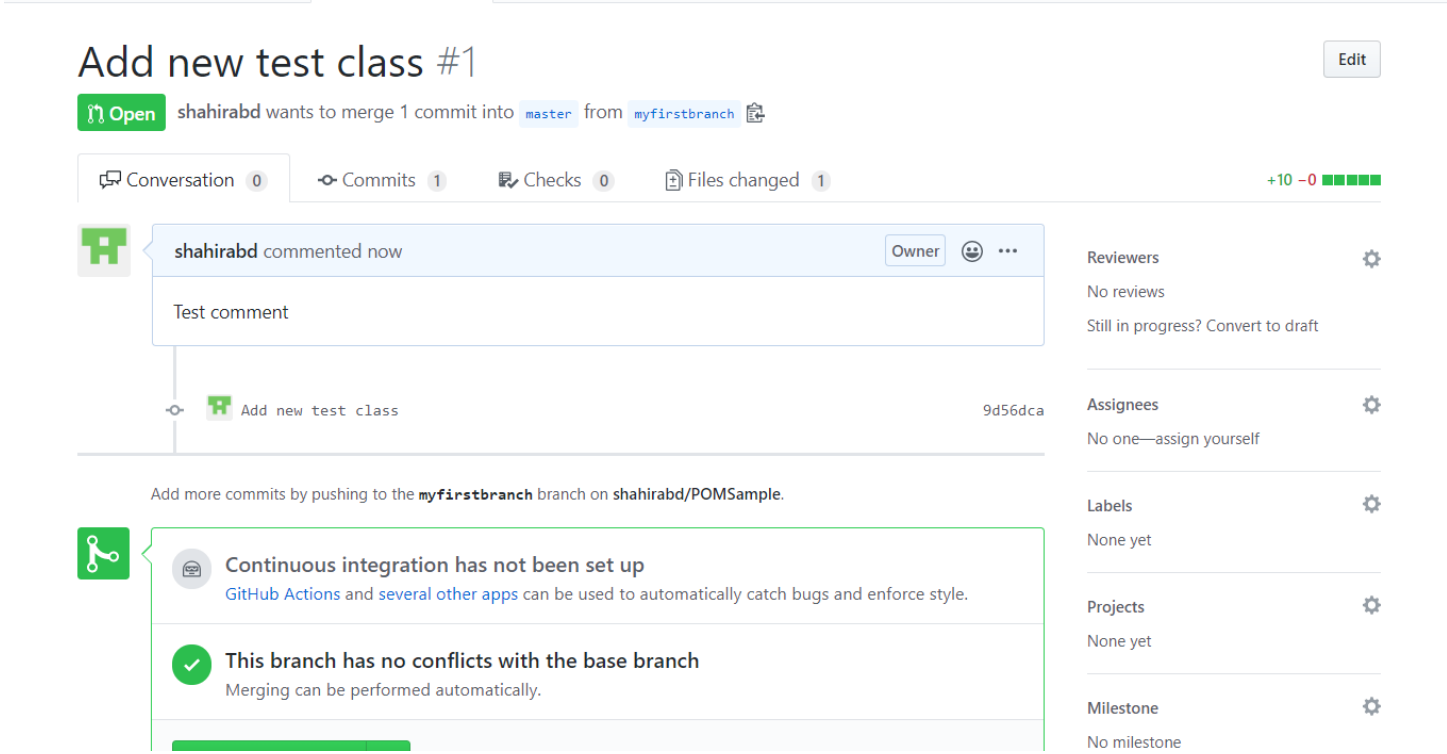
CREATE PULL REQUEST – GITHUB DESKTOP

- Click on the ‘Push origin’ button’



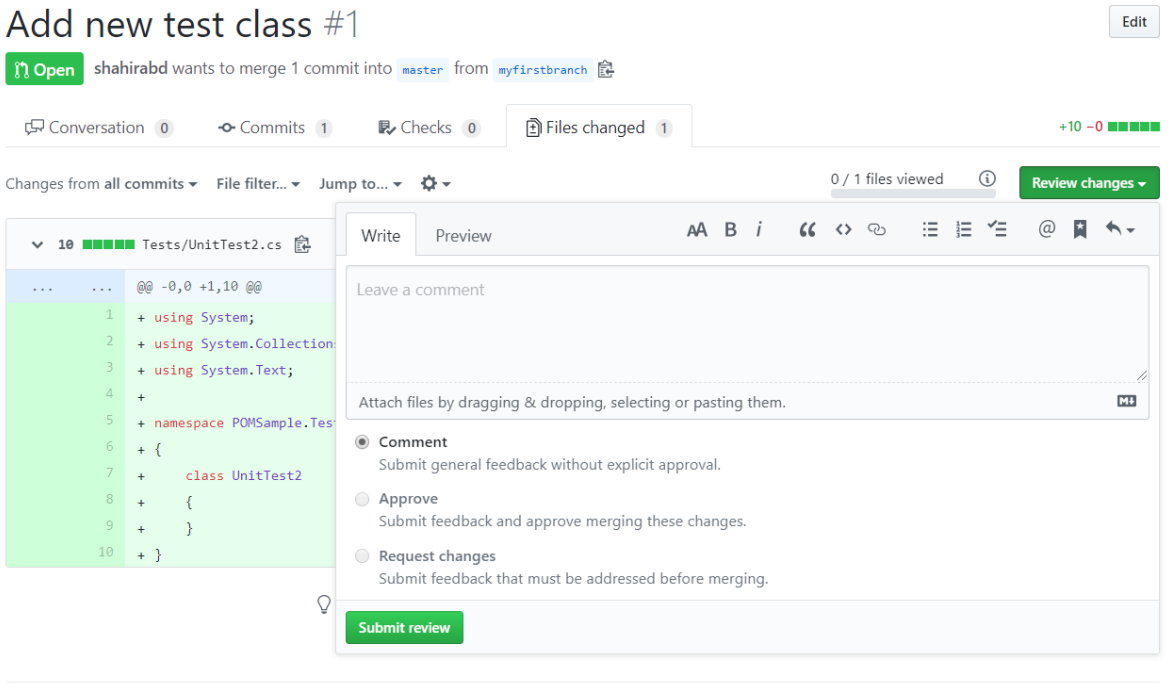
CREATE PULL REQUEST

- Pull request created



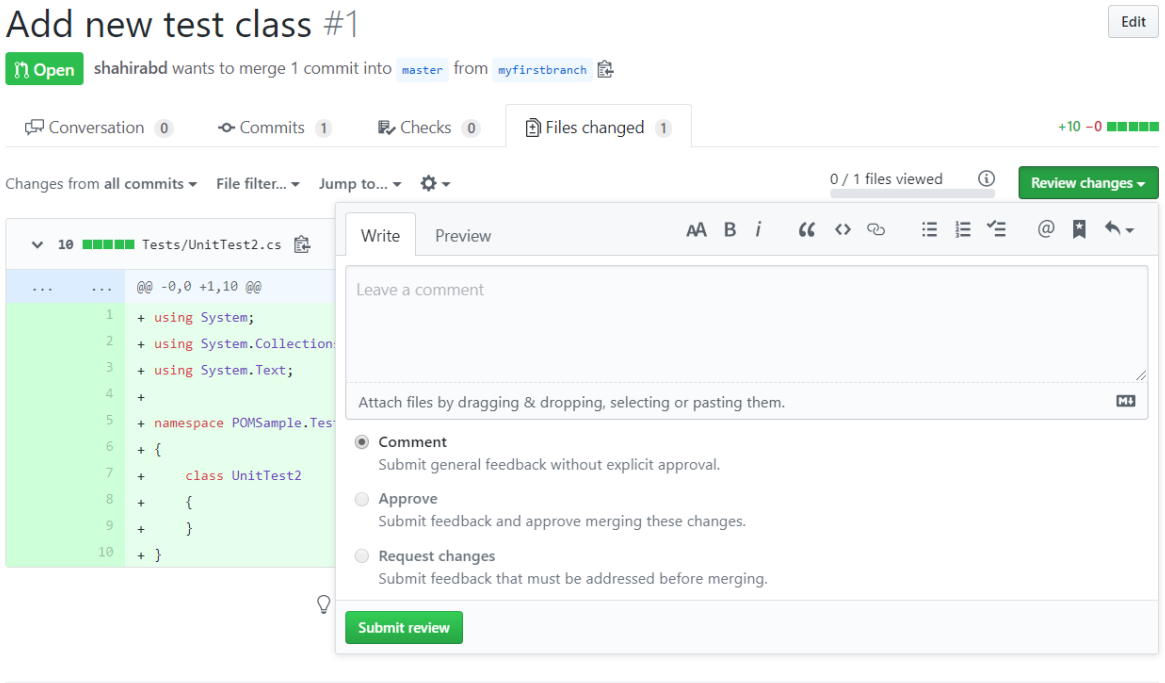
REVIEW PULL REQUEST

- Reviewer can approve or request change on the pull request
- Who should review?
  - Other team members
  - Leads
  - Managers
  - …who are working on the project



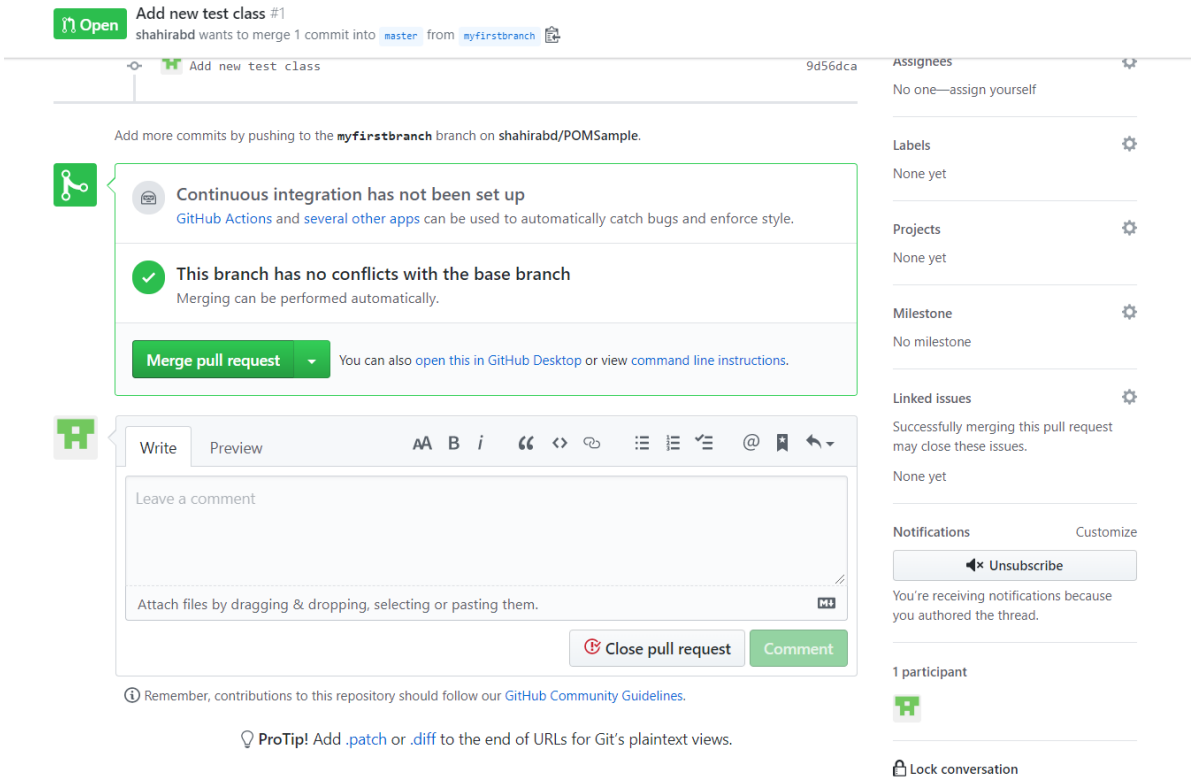
RESPOND TO A REVIEW

- Once review is done, we need to make changes accordingly (if any).
  - Git pull
  - Git checkout <branch-name>
  - Repeat the whole process (change the file, stage, commit, push)
  - Once pushed, the pull request updated automatically, we don't have to create a new pull request from the GitHub website.
- Finally, we can merge the working branch to master branch



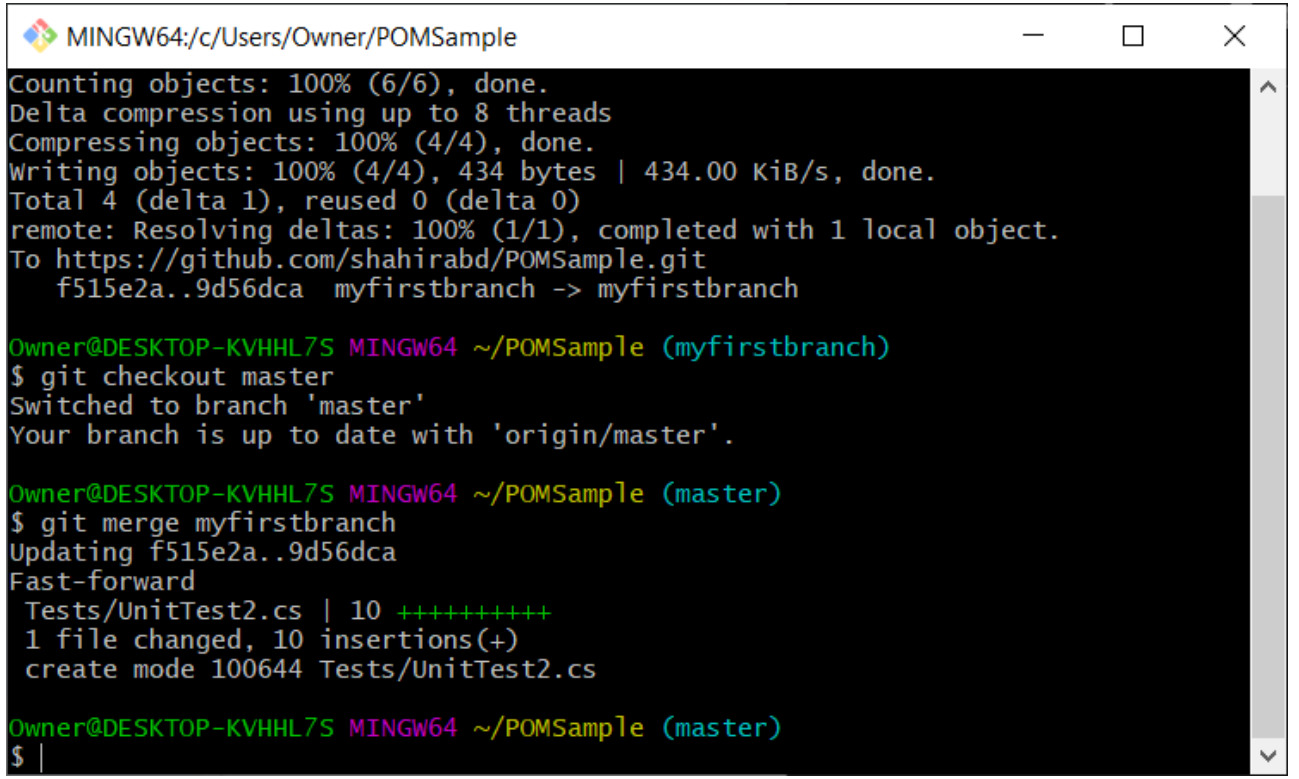
MERGE PULL REQUEST

- Click on the ‘merge pull request’ button



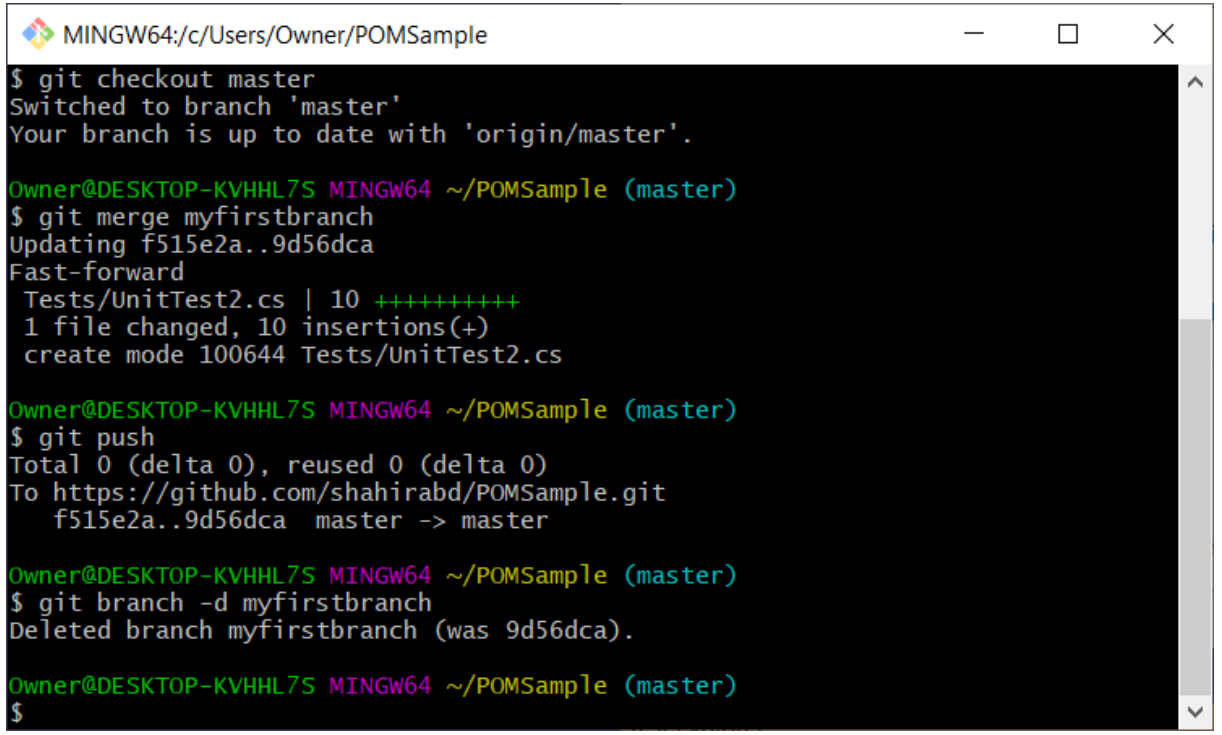
MERGE PULL REQUEST - CLI

- Or use the command line
  - git checkout master
  - git merge <branch-name>



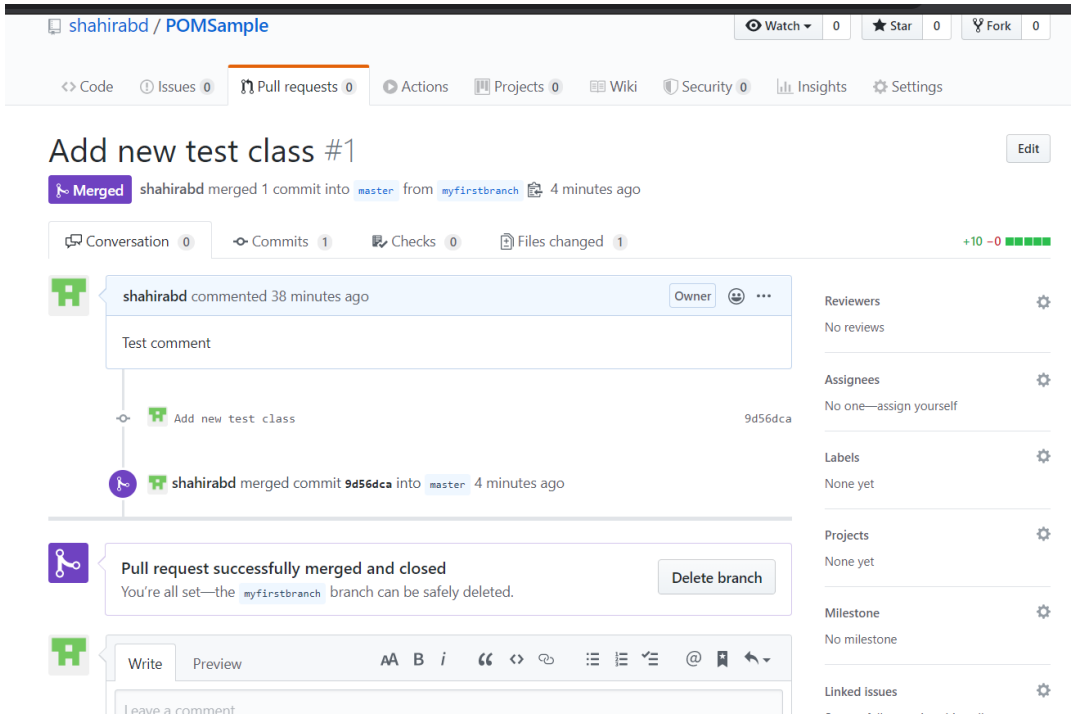
MERGE PULL REQUEST - CLI

- git push
- git branch –d <branch-name>



MERGE PULL REQUEST

- Check the PR on GitHub website

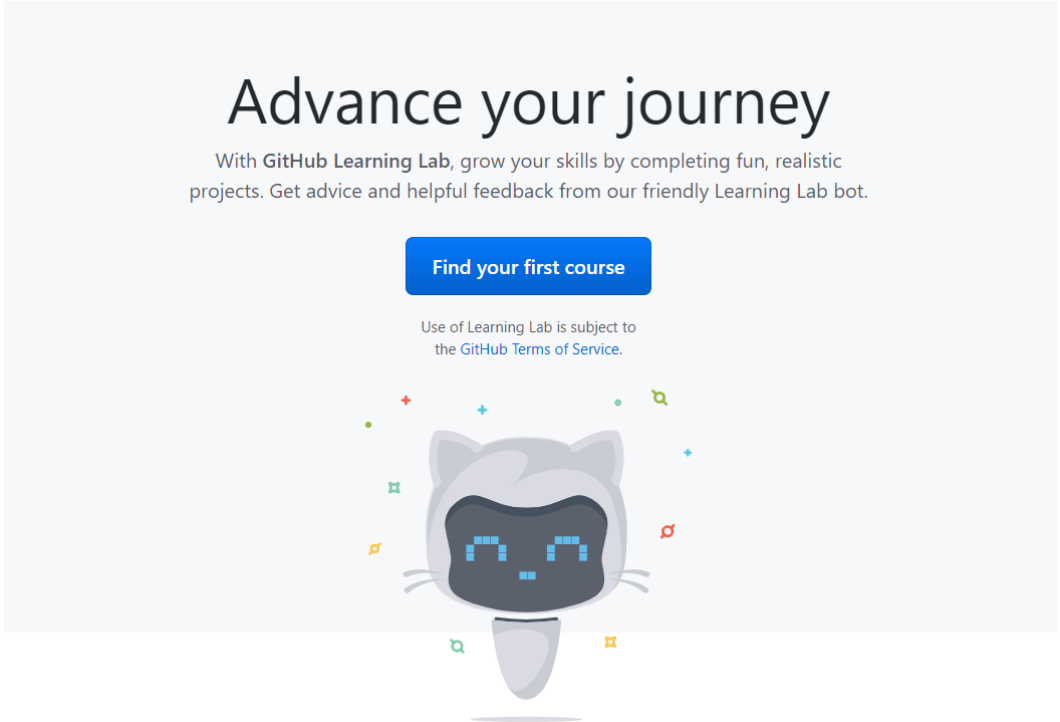


ADDITIONAL COURSE(S)

For you to explore on your own

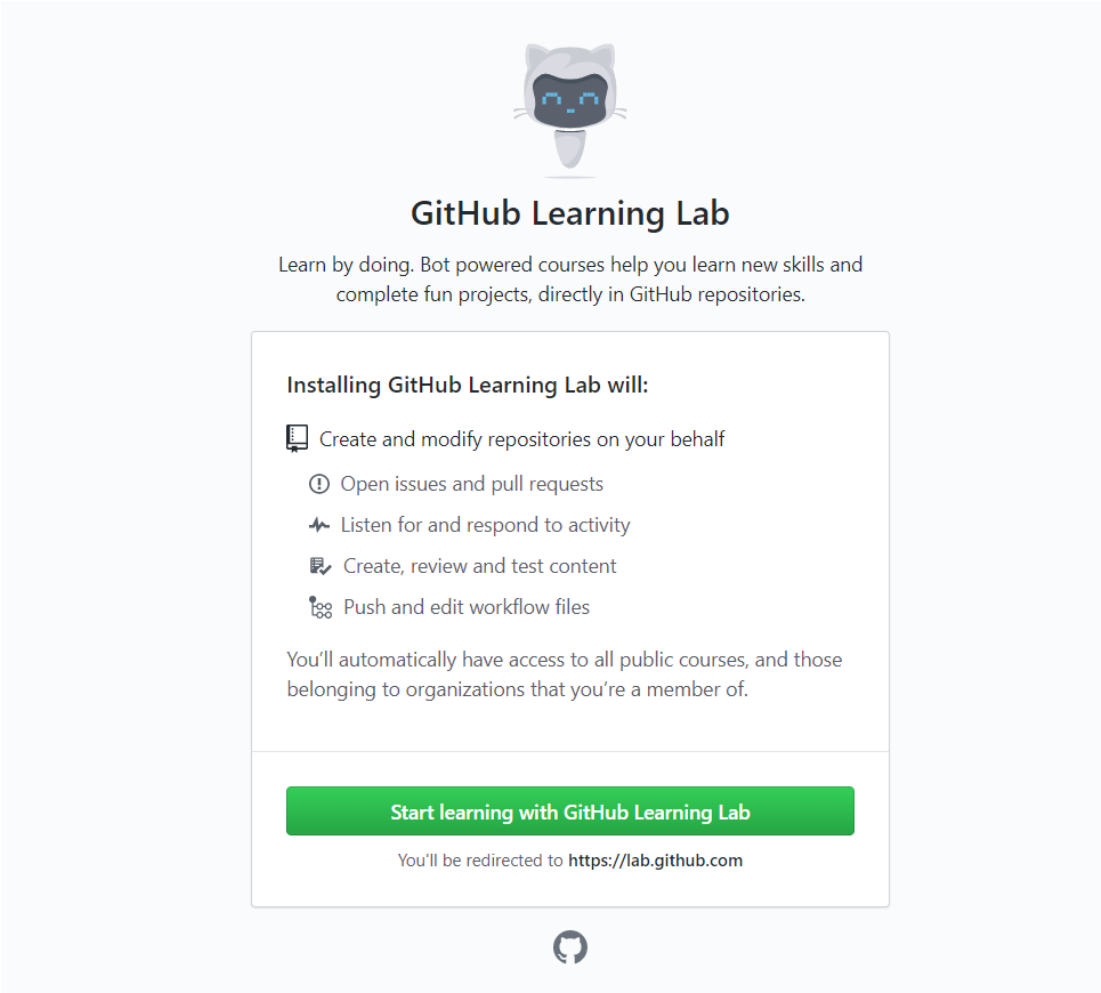
SELF-LEARNING

- Go to lab.github.com
- Login with your GitHub account



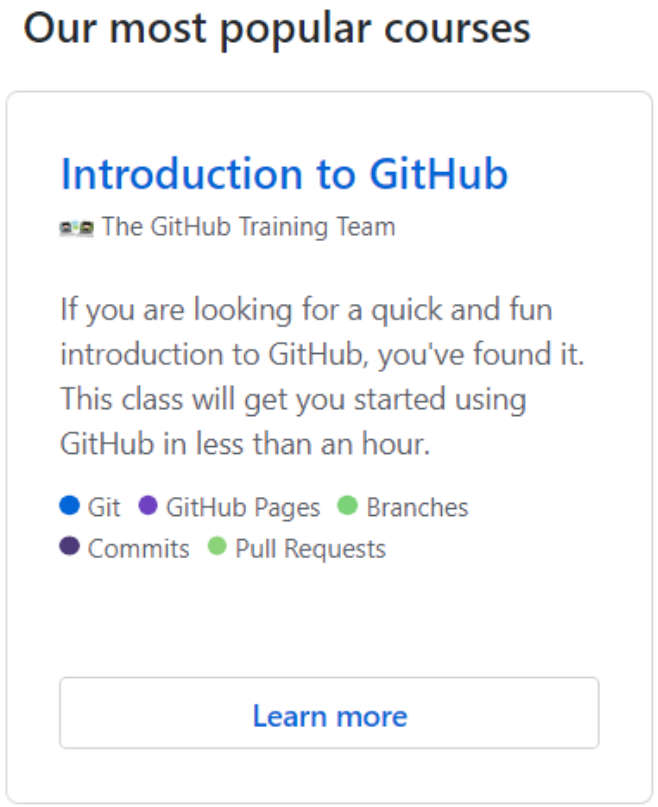
SELF-LEARNING

- Click on start learning button



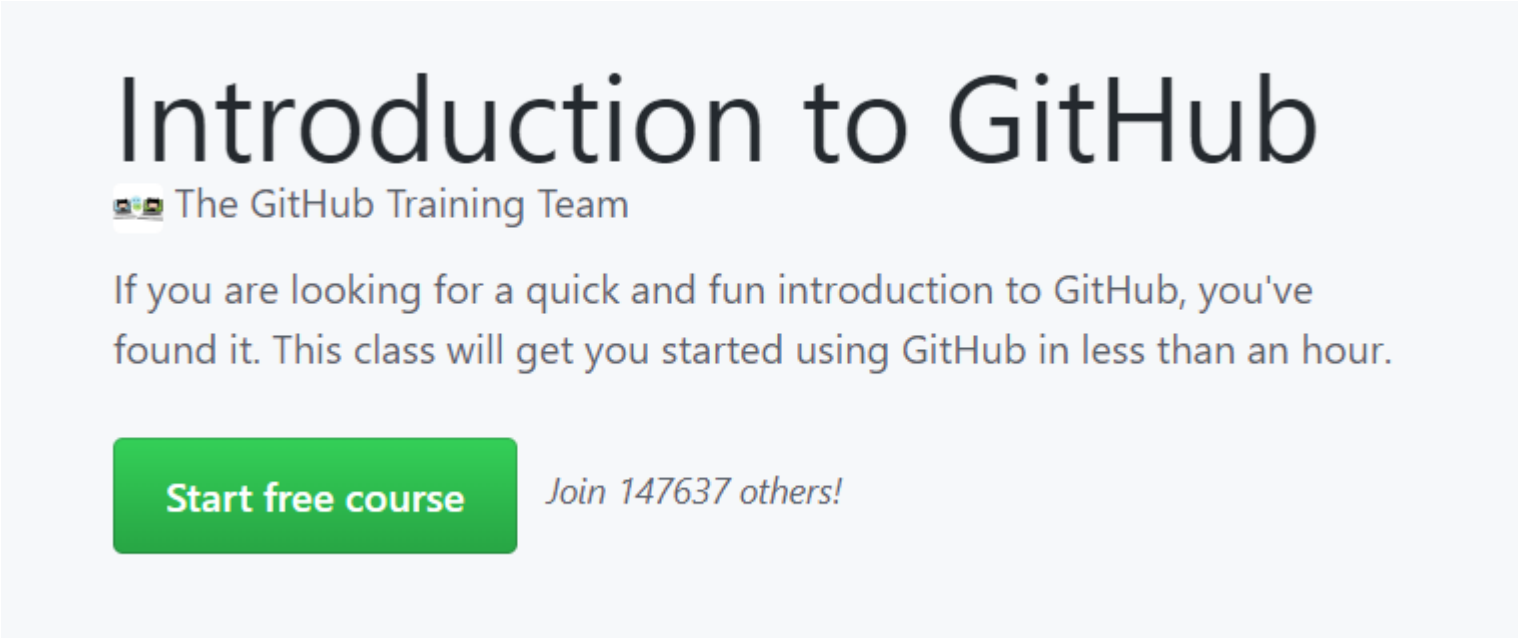
SELF-LEARNING

- Look for this course



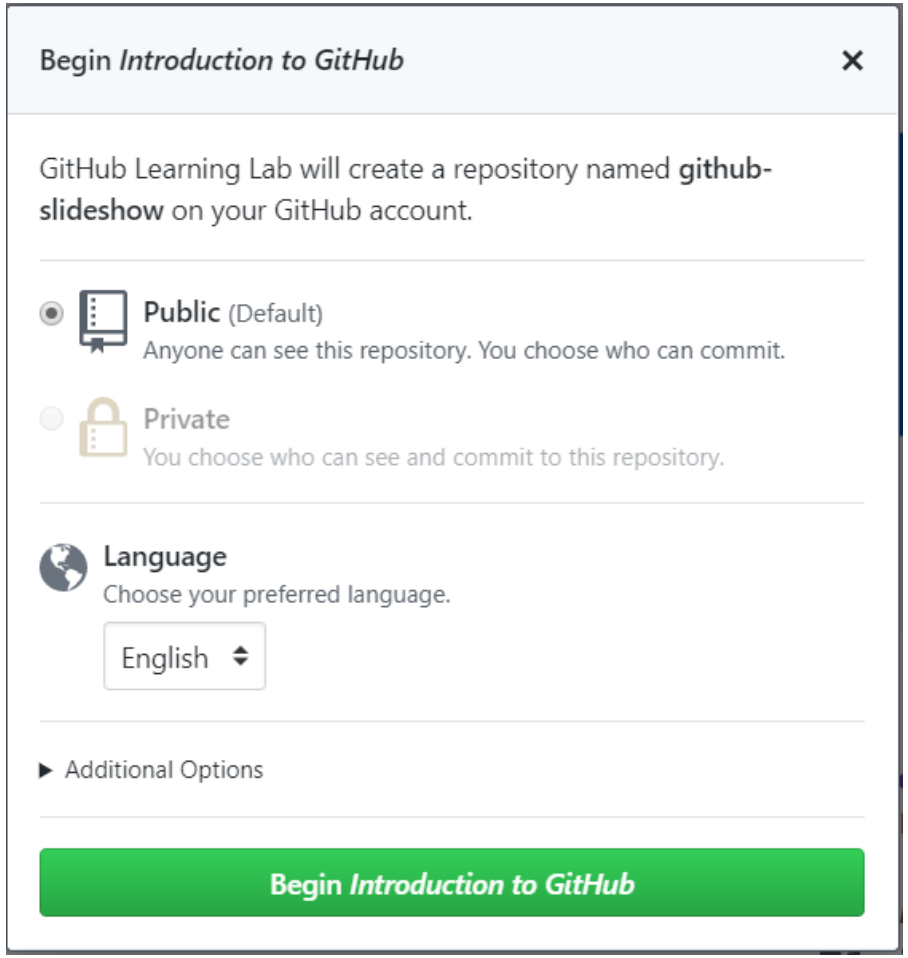
SELF-LEARNING

- Click on the start free course button



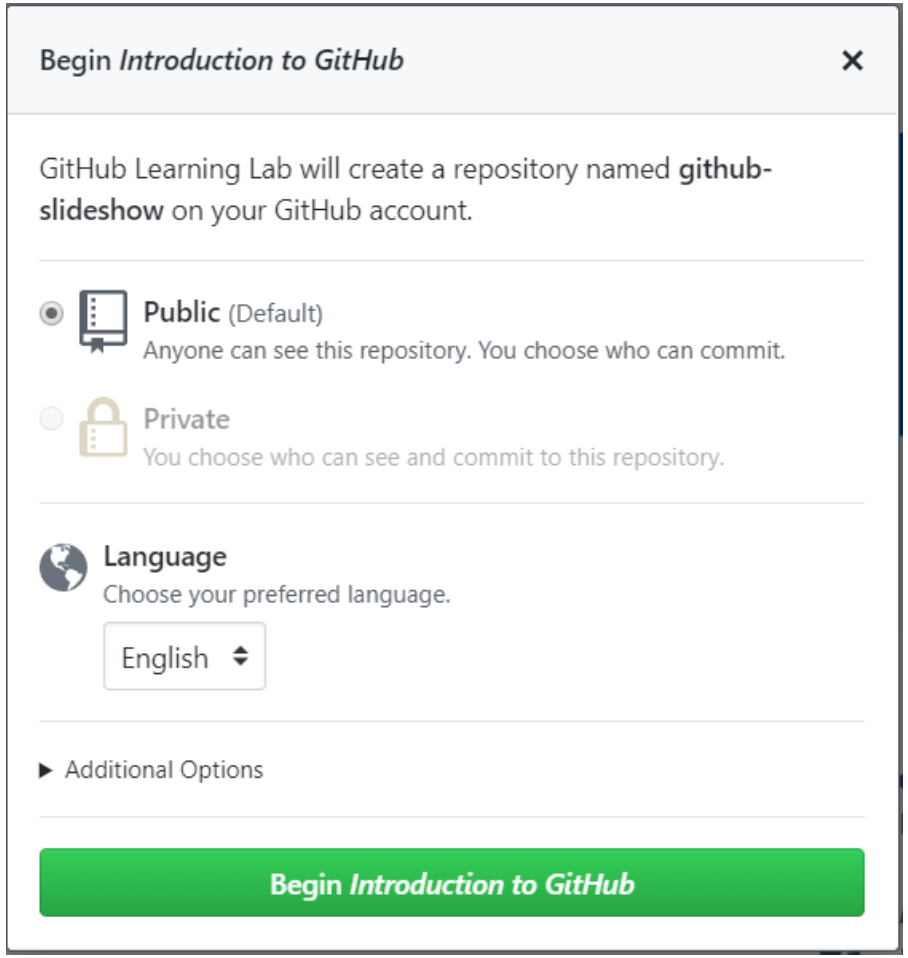
SELF-LEARNING

- Click on the start free course button



SELF-LEARNING

- Click on the start free course button



ONLINE
ASSESSMENT

ONLINE ASSESSMENT

- https://extendedforms.io/form/956e038d-3877-4159-81dc-ae3ac7af1418/login
- Please complete this assessment, once you have studied and practiced
- You can refer to this slide and the official GitHub documentation - docs.github.com.