

Sử dụng Git & GitHub

Git là một trong các hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. Cài đặt Git:

- Tải Git tại <https://git-for-windows.github.io/>. Double-click vào file cài đặt và thực hiện theo hướng dẫn.
- Để kiểm tra cài đặt thành công, mở Command Prompt gõ lệnh “git --version” kết quả sẽ hiển thị giống như sau:

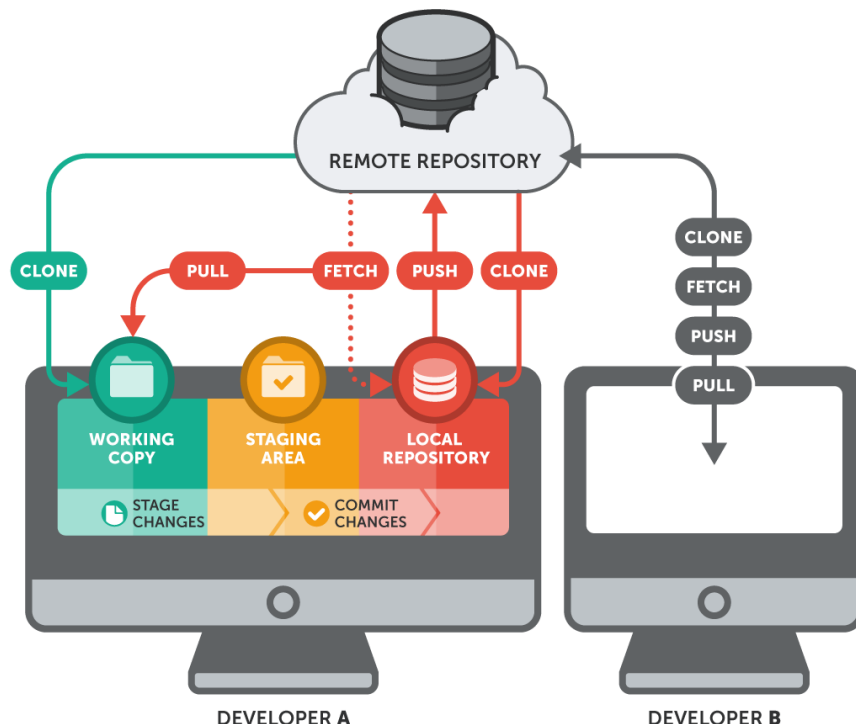
```
Command Prompt
Microsoft Windows [Version 10.0.19042.508]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\DELL>git --version
git version 2.42.0.windows.2
```

- Tại Command Prompt thiết lập username và email cho git.

```
C:\Users\DELL>git config --global user.name "thanh nguyen"
C:\Users\DELL>git config --global user.email "ntnthanhhou@gmail.com"
```

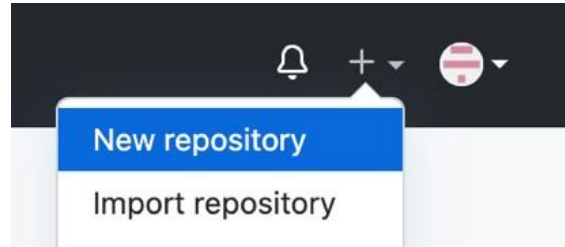
GitHub là một nền tảng cung cấp kho lưu trữ mã nguồn git. Nó cho phép nhiều người làm việc trên một project ở bất cứ nơi đâu. Địa chỉ GitHub: <http://github.com/>. Bên dưới các thao tác cơ bản làm việc git & github.



Các thao tác trên Github

Tạo thùng chứa

Sau khi đăng nhập, ở góc trên phải click biểu tượng dấu “+” chọn “New repository”





Một số thông tin cần chú ý khi tạo thùng chứa

- Tên thùng chứa thường là tên project và không trùng tên với các thùng chứa đã có.
- Chọn chính sách cho thùng chứa: nếu là **Public** thì **mọi người** đều có thể thấy thùng chứa, nếu là **Private** thì chỉ **mình và thành viên mình thêm vào** mới có thể thấy thùng chứa. Trong hai lựa chọn, ta đều có thể chỉ định người được commit thùng chứa.
- Ta có thể tích chọn “Add a README file” để tạo tập tin Readme chứa các mô tả chi tiết của thùng chứa. Mặc dù yêu cầu này là tùy chọn, nhưng ta nên luôn tạo ra tập tin này.
- Tạo tập tin .gitignore cấu hình những thành phần trong thùng chứa mà git không nên quan tâm, trong select box của nó ta có thể chọn ngôn ngữ thực hiện thùng chứa để Github tạo ra một số cấu hình có sẵn.

Start a new repository

A repository contains all of your project's files, revision history, and collaborator discussion.

thanh123455 /

- ☒  **Public**
Anyone on the internet can see this repository
- ☐  **Private**
You choose who can see and commit to this repository

Create a new repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * thanh123455 / Repository name * demo_1
demo_1 is available.

Great repository names are short and memorable. Need inspiration? How about [cautious-lamp](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

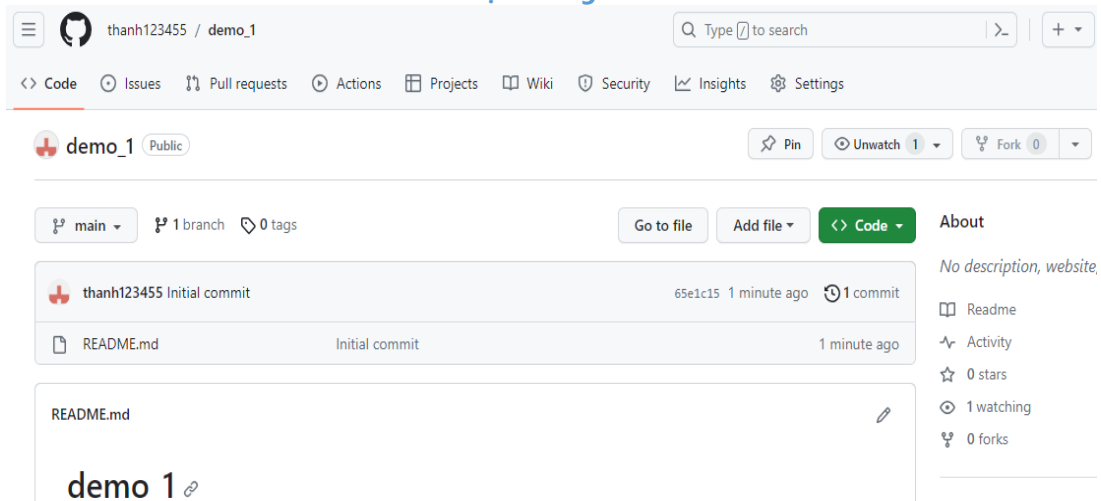
Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set main as the default branch. Change the default name in your [settings](#).

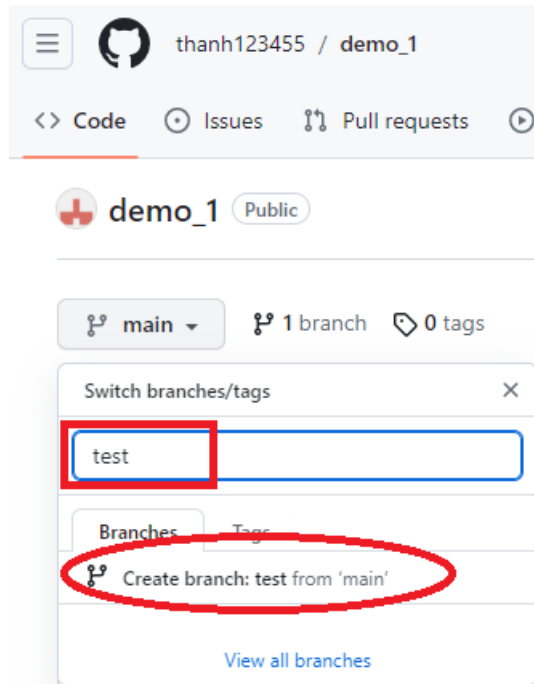
Hình 1. Tạo thùng chứa Github



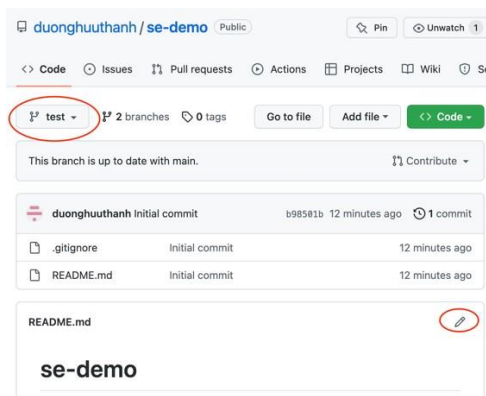
Tạo nhánh

Nhánh là cách thức làm việc trên một phiên bản khác của thùng chứa tại cùng thời điểm. Mặc định khi vừa tạo thùng chứa, GitHub tạo sẵn nhánh main (ta có thể đổi tên mặc định lúc tạo thùng chứa). Ví dụ tạo nhánh test từ nhánh main

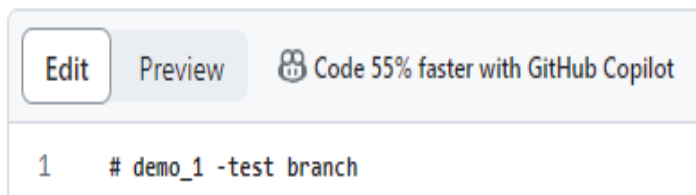
- Tại thùng chứa, click vào select box “main”
- Gõ tên nhánh (develop) muốn tạo ở và click “Create branch: test”



Ta thực hiện một số thay đổi trong nhánh test. Đầu tiên chọn nhánh test và chỉnh sửa tập tin README.



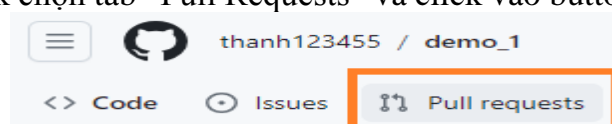
Chỉnh sửa xong tập tin README, ta thực hiện Commit changes.



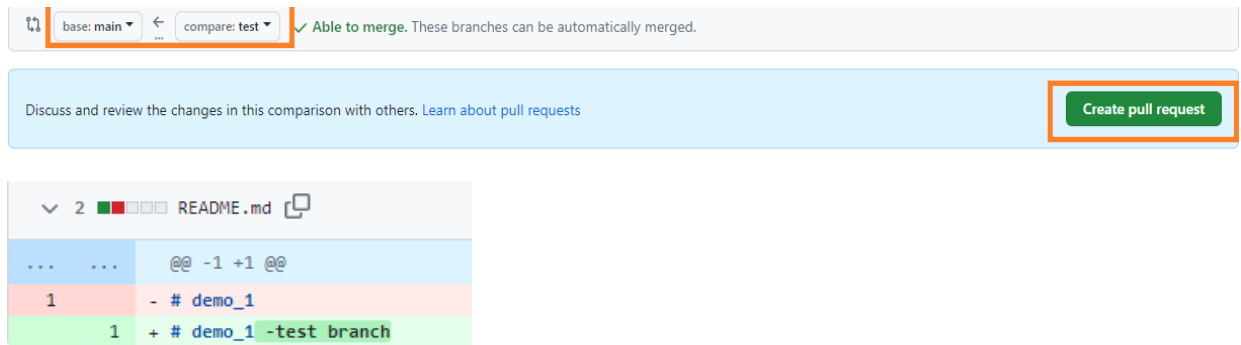
Tạo Pull Request

Pull Request (PR) dùng để đề nghị (proposing) những thay đổi và yêu cầu để ai đó xét duyệt (review) trộn (merge) từ một nhánh này vào nhánh nào khác. PR sẽ thể hiện sự khác nhau nội dung giữa hai nhánh

Tại thùng chứa, click chọn tab “Pull Requests” và click vào button “New pull request”

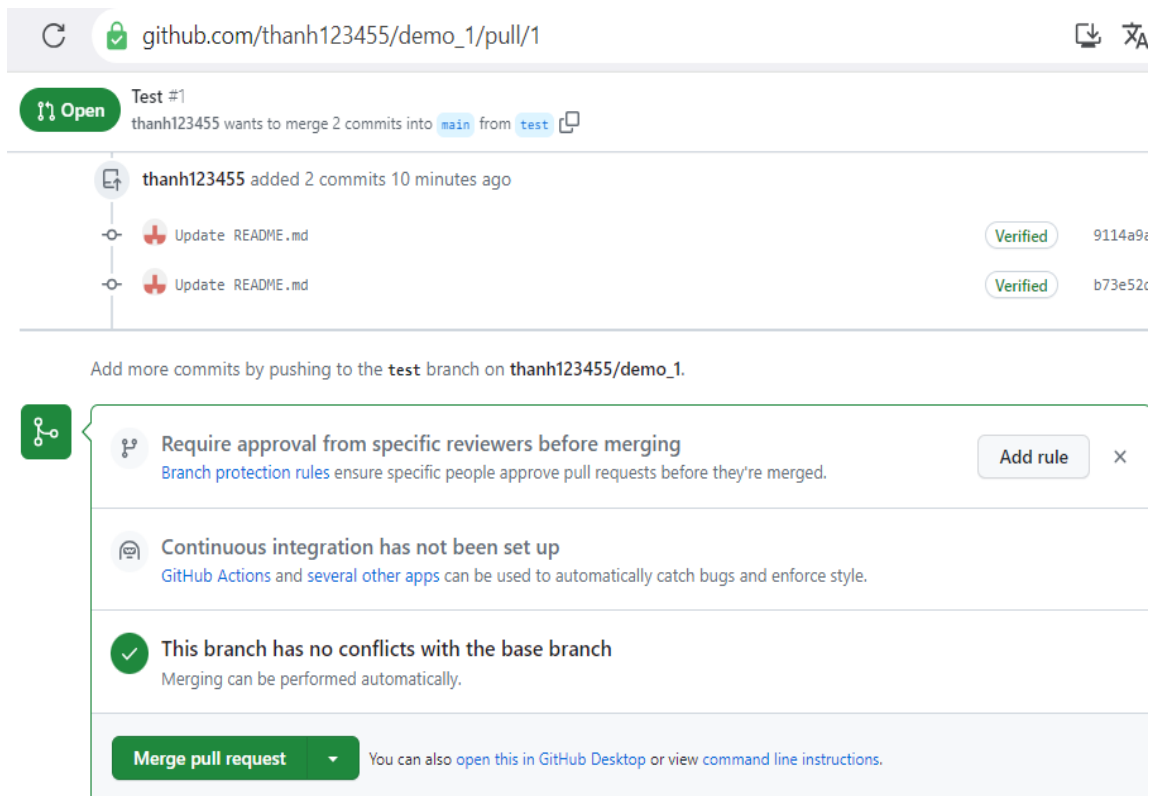


Để yêu cầu trộn nhánh test vào nhánh main, ta chọn như bên dưới và click “Create pull request”



Trộn Pull Request

Sau khi tạo PR, các thay đổi được tập trung trong PR để có thể xét duyệt và quyết định trộn hay không. Để merge PR, tại PR vừa tạo click “Merge pull request” và “Confirm merge”. Sau đó quay lại branch main, ta sẽ thấy những thay đổi trên test được áp dụng vào nhánh main.



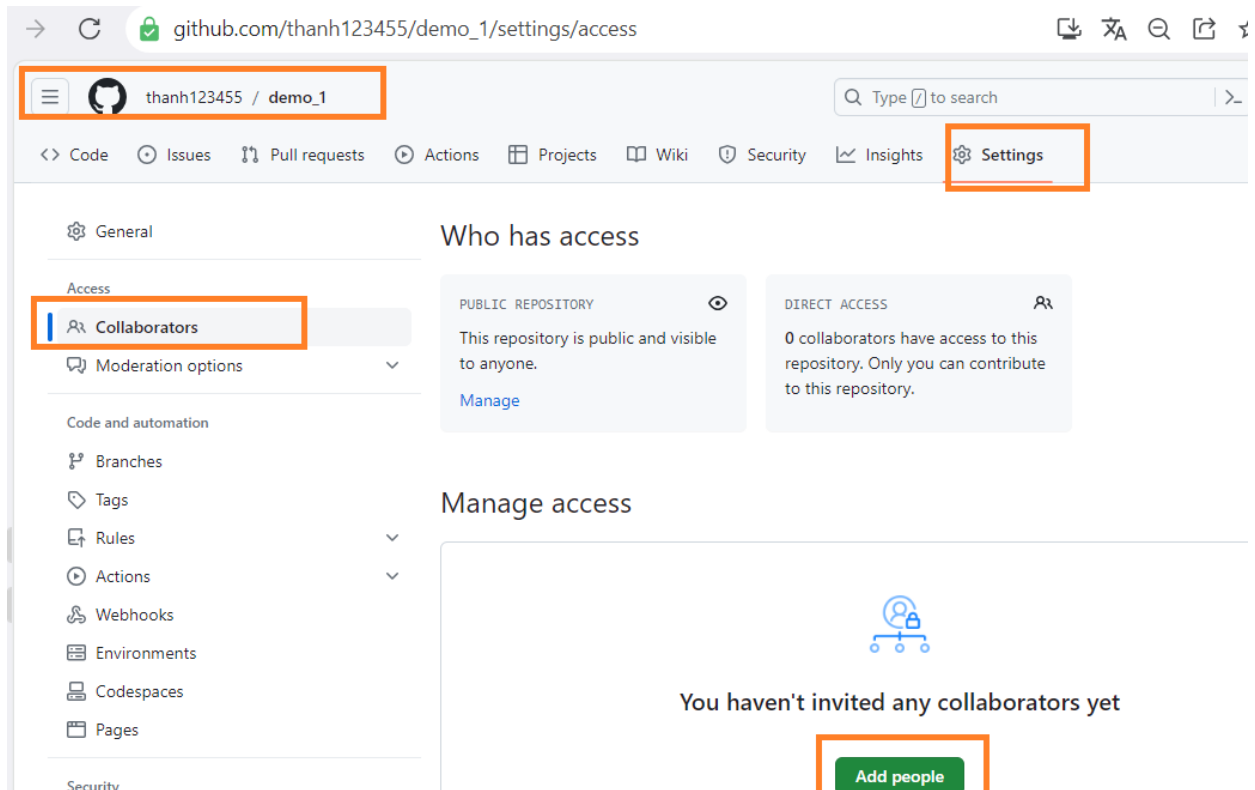
Thêm thành viên vào thùng chứa

Để các thành viên có thể push code vào thùng chứa cần thêm họ thành cộng tác (collaborator) vào thùng chứa. Các bước thực hiện

- Tại thùng chứa, click tab Settings, và click vào mục Collaborators ở mục menu bên

trái.

- Click “Add people” và nhập tên thành viên muốn mời, lời mời sẽ được gửi vào mail thành viên và chờ họ chấp nhận/từ chối.

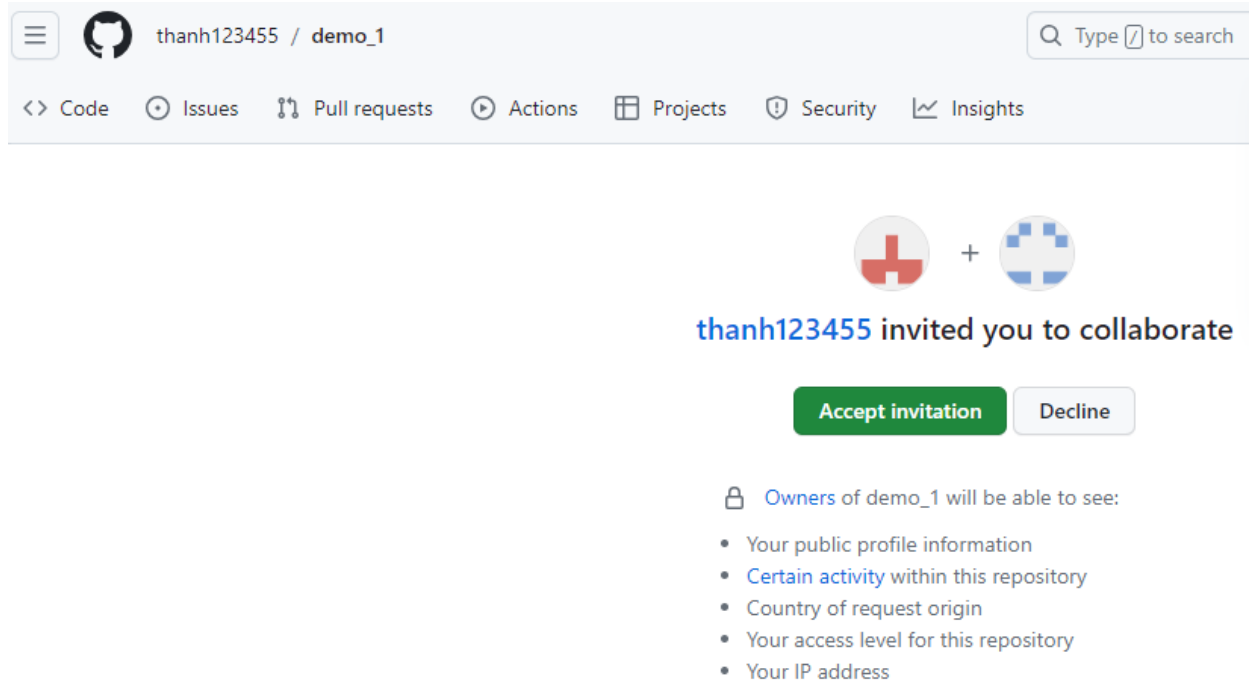


Add a collaborator to demo_1

@gmail.com

Invite to demo_1

Select a collaborator above



Các thao tác cơ bản trên git

- git init: khởi tạo thùng chứa cục bộ

- git clone: lấy thùng chứa từ remote về local.
- git add: thêm những thay đổi từ working copy sang staging area
- git commit: thêm thay đổi từ staging area đến local repository
- git push: đưa thay đổi từ local repository lên remote repository (github)
- git stash: lưu tạm các thay đổi nhánh hiện tại vào stash
- git diff: so sánh sự khác nhau trạng thái hiện tại thùng chứa với trạng thái ban đầu, hoặc giữa hai commit.
- git status: xem trạng thái hiện tại các tập tin trong thùng chứa.
- git merge: trộn thay đổi hai nhánh.
- git log: xem lịch sử commit
- git branch: tạo nhánh hoặc xem danh sách nhánh
- git checkout branch-name: chuyển sang làm việc trên 1 nhánh
- git checkout file-name: phục hồi thay đổi trên tập tin về trạng thái ban đầu.
- git reset file-name: đưa 1 tập tin từ staging area trở về working copy.
- git pull: cập nhật những thay đổi thùng chứa remote về thùng chứa local.
- git fetch: lấy tất cả các nhánh của thùng chứa remote về thùng chứa local.
- git rm file-name: xóa tập tin khỏi project và đưa vào staging area để chuẩn bị commit.


Thực hành 1

- Tạo thùng chứa trên github
- Lấy thùng chứa về máy cục bộ (git clone)
- Tạo một số thay đổi tùy ý trong thùng chứa và thực hiện quá trình đưa lên thay đổi lên github. (git add → git commit → git push)
- Hai thành viên ngồi cạnh, thêm thành viên kế bên vào thùng chứa, yêu cầu bạn thực hiện push thay đổi lên thùng chứa của mình.

Hướng dẫn:

1. Hướng dẫn tạo thùng chứa trên github

Owner * Repository name *


 thanh123455 / test_ktpm


✓ test_ktpm is available.

Great repository names are short and memorable. Need inspiration? How

Description (optional)

code chung cplusplus

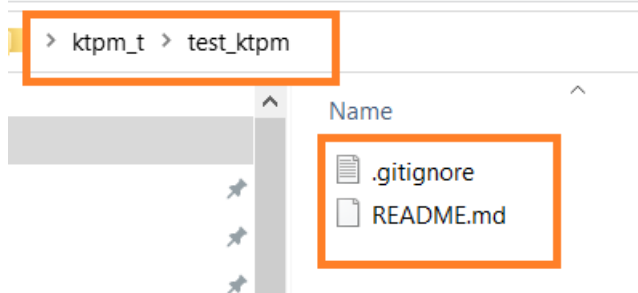
☒  **Public**
Anyone on the internet can see this repository. You choose who can commit to this repository.

☐  **Private**
You choose who can see and commit to this repository.

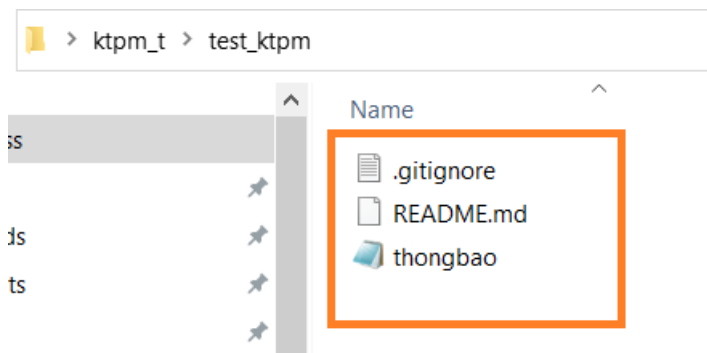
2. Lấy thùng chứa về máy cục bộ (git clone)

Lấy thùng chứa về máy, tạo thư mục ktpm_t trên máy local, trên github copy đường dẫn vào

```
C:\Users\DELL\Desktop\ktpm_t>git clone https://github.com/thanh123455/test_ktpm.git
Cloning into 'test_ktpm'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```



Tạo 1 pro là java hay c#, ở đây tạo file .cpp



File mới thêm vào sẽ có màu xanh trong project

Xem trạng thái thùng chứa, git status

```
C:\Users\DELL\Desktop\ktpm_t>
C:\Users\DELL\Desktop\ktpm_t>cd test_ktpm
C:\Users\DELL\Desktop\ktpm_t\test_ktpm>git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    thongbao.cpp

nothing added to commit but untracked files present (use "git add" to track)
```

Thongbao.cpp màu đỏ -- ie là đang làm trên thư mục working copy

Lệnh git add chuyển từ working copy qua staging area

Nếu là project

Git add .gitignore TenProject/ -> sẽ có màu xanh

Chuyển từ staging area qua local repo

```
C:\Users\DELL\Desktop\ktpm_t\test_ktpm>git add thongbao.cpp

C:\Users\DELL\Desktop\ktpm_t\test_ktpm>git commit -m "ly do them file thong bao cpp"
[main d960980] ly do them file thong bao cpp
1 file changed, 12 insertions(+)
create mode 100644 thongbao.cpp

C:\Users\DELL\Desktop\ktpm_t\test_ktpm>
```

Xem lại status

```
C:\Users\DELL\Desktop\ktpm_t\test_ktpm>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

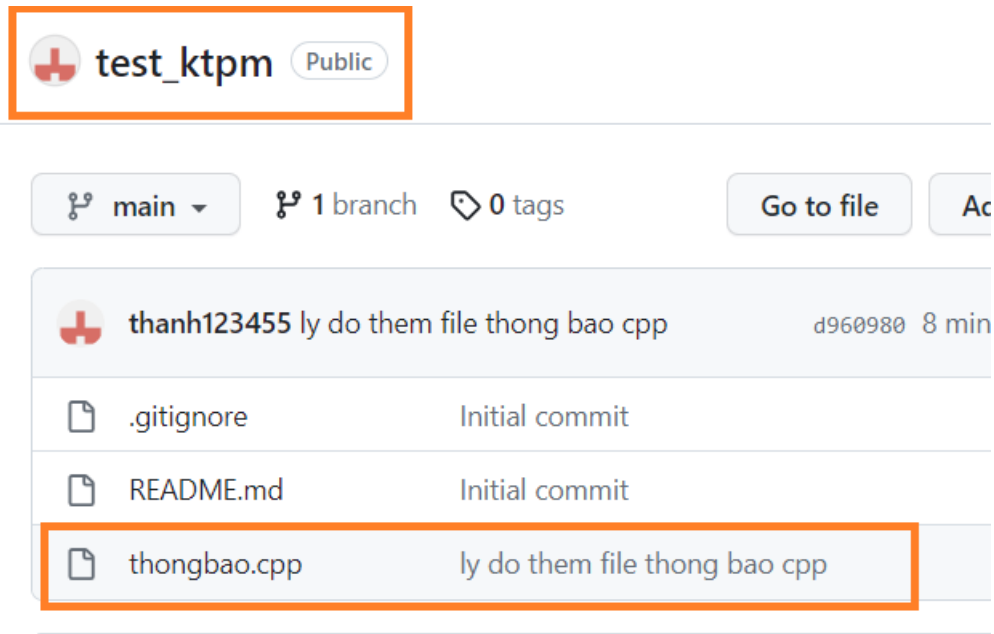
Ko có màu đỏ, đã sẵn sàng push lên server

Đưa từ local lên server

Git push origin (tên kết nối) main(tên nhánh push lên) mặc định là main

```
C:\Users\DELL\Desktop\ktpm_t\test_ktpm>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 421 bytes | 421.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/thanh123455/test_ktpm.git
38771a7..d960980  main -> main
```

Lên github kiểm tra lại thùng chứa.



Vào dev c++, thêm dòng in hello them ra màn hình

```
C:\Users\DELL\Desktop\ktpm_t\test_ktpm>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   thongbao.cpp

no changes added to commit (use "git add" and/or "git commit -a")
```

....

Các bước tiếp theo để làm được bài thực hành 1, sv tự bổ sung để hoàn thiện.