**CÔNG TY CÔNG NGHỆ THÔNG TIN VNPT**

****

**Hướng dẫn sử dụng hệ thống quản lý mã nguồn Git**

**Người thực hiện: Hà Ngọc Anh**

Hà Nội – 2019

Mục lục

[Danh mục hình ảnh 3](#_Toc25128987)

[1. Tổng quan về Git- Github 4](#_Toc25128988)

[1.1 Git là gì? 4](#_Toc25128989)

[1.2 Github là gì? 4](#_Toc25128990)

[1.3 Phạm vi và mục tiêu tài liệu 4](#_Toc25128991)

[2. Hướng dẫn sử dụng các lệnh cơ bản về Github 6](#_Toc25128992)

[2.1 Khởi tạo tài khoản user 6](#_Toc25128993)

[2.2 Khởi tạo repository 7](#_Toc25128994)

[2.3 Clone project 8](#_Toc25128995)

[2.4 Commit 9](#_Toc25128996)

[2.5 Push 10](#_Toc25128997)

[2.6 Pull 10](#_Toc25128998)

[2.7 Branch 11](#_Toc25128999)

[2.8 Pull request 13](#_Toc25129000)

[2.9 Merge 15](#_Toc25129001)

[3. Tổng kết 17](#_Toc25129002)

[4. Tài liệu tham khảo 18](#_Toc25129003)

[Phụ lục 19](#_Toc25129004)

# Danh mục hình ảnh

[Hình 1: Mô hình tổ chức dữ liệu phân tán 4](#_Toc24444847)

[Hình 3: Đăng kí tài khoản 6](#_Toc24444849)

[Hình 4: Đăng nhập vào hệ thống 7](#_Toc24444850)

[Hình 5: Trang chủ github 8](#_Toc24444851)

[Hình 6: Giao diện tạo mới repository 8](#_Toc24444852)

[Hình 7: Khởi tạo thành công repository 9](#_Toc24444853)

[Hình 8: Copy URL git để thực hiện clone 9](#_Toc24444854)

[Hình 9: Add repository local vào SourceTree để quản lý 10](#_Toc24444855)

[Hình 10: Add vào SourceTree thành công 10](#_Toc24444856)

[Hình 11: Commit thay đổi 11](#_Toc24444857)

[Hình 12: Push thay đổi lên repository 11](#_Toc24444858)

[Hình 13: Pull dữ liệu về 12](#_Toc24444859)

[Hình 14: Tạo Branch 13](#_Toc24444860)

[Hình 15: Push thay đổi lên serever 14](#_Toc24444861)

[Hình 16: Tạo branch và push thành công 14](#_Toc24444862)

[Hình 17: New pull request 15](#_Toc24444863)

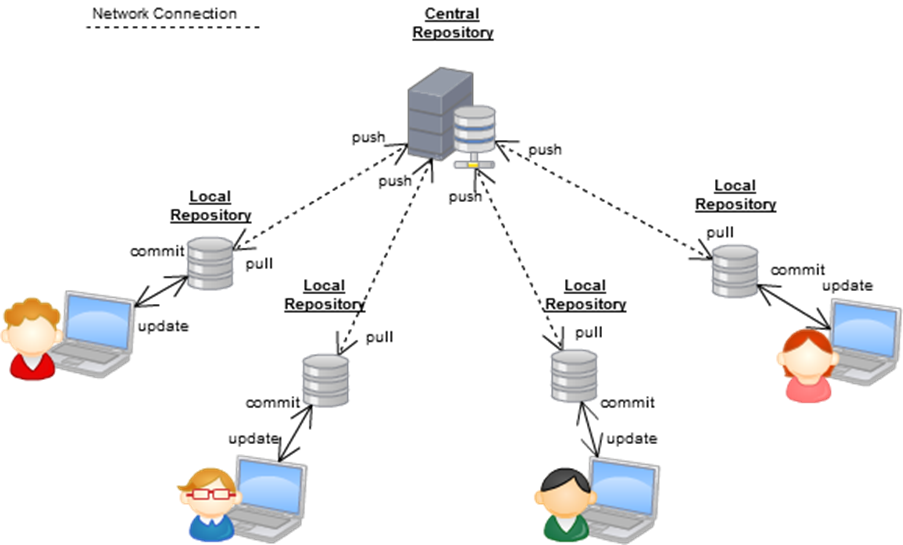
[Hình 18: Create pull request 15](#_Toc24444864)

[Hình 19: Merge request 16](#_Toc24444865)

# Tổng quan về Git- Github

## Git là gì?

Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) ra đời vào năm 2005 và hiện được dùng rất phổ biến. So với các hệ thống quản lý phiên bản tập trung khi tất cả mã nguồn và lịch sử thay đổi chỉ được lưu một nơi là máy chủ thì trong hệ thống phân tán, các máy khách không chỉ "check out" phiên bản mới nhất của các tập tin mà là sao chép (mirror) toàn bộ kho mã nguồn (repository). Như vậy, nếu như máy chủ ngừng hoạt động, thì bạn hoàn toàn có thể lấy kho chứa từ bất kỳ máy khách nào để sao chép ngược trở lại máy chủ để khôi phục lại toàn bộ hệ thống. Mỗi checkout thực sự là một bản sao đầy đủ của tất cả dữ liệu của kho chứa từ máy chủ.



Hình 1: Mô hình tổ chức dữ liệu phân tán

## Github là gì?

Github là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn Git cho các dự án phần mềm. Github có đầy đủ những tính năng của Git, ngoài ra nó còn bổ sung những tính năng về social để các developer tương tác với nhau.

## Phạm vi và mục tiêu tài liệu

Tài liệu hướng dẫn sử dụng công cụ quản lý phiên bản mã nguồn GitHub dành cho: Developer, Trưởng nhóm (lead), QA, Tester của trung tâm SI.

Mục tiêu: Tài liệu dùng để đào tạo cho người mới sử dụng Git hoặc bắt đầu áp dụng mô hình Gitflow, Forkflow.

Phạm vi tài liệu: Hướng dẫn các thao tác thường dùng với Git trên web, client khi thực hiện tạo project, push thay đổi mã nguồn, review mã nguồn, release các phiên bản phần mềm.

Các hướng dẫn được thực hiện trên môi trường client là máy tính sử dụng hệ điều hành windows.

Tài liệu không mô tả hướng dẫn vận hành hệ thống Git cho admin (tương đương tài khoản admin của git)

Danh sách người dùng tham gia sử dụng hệ thống quản lý mã nguồn

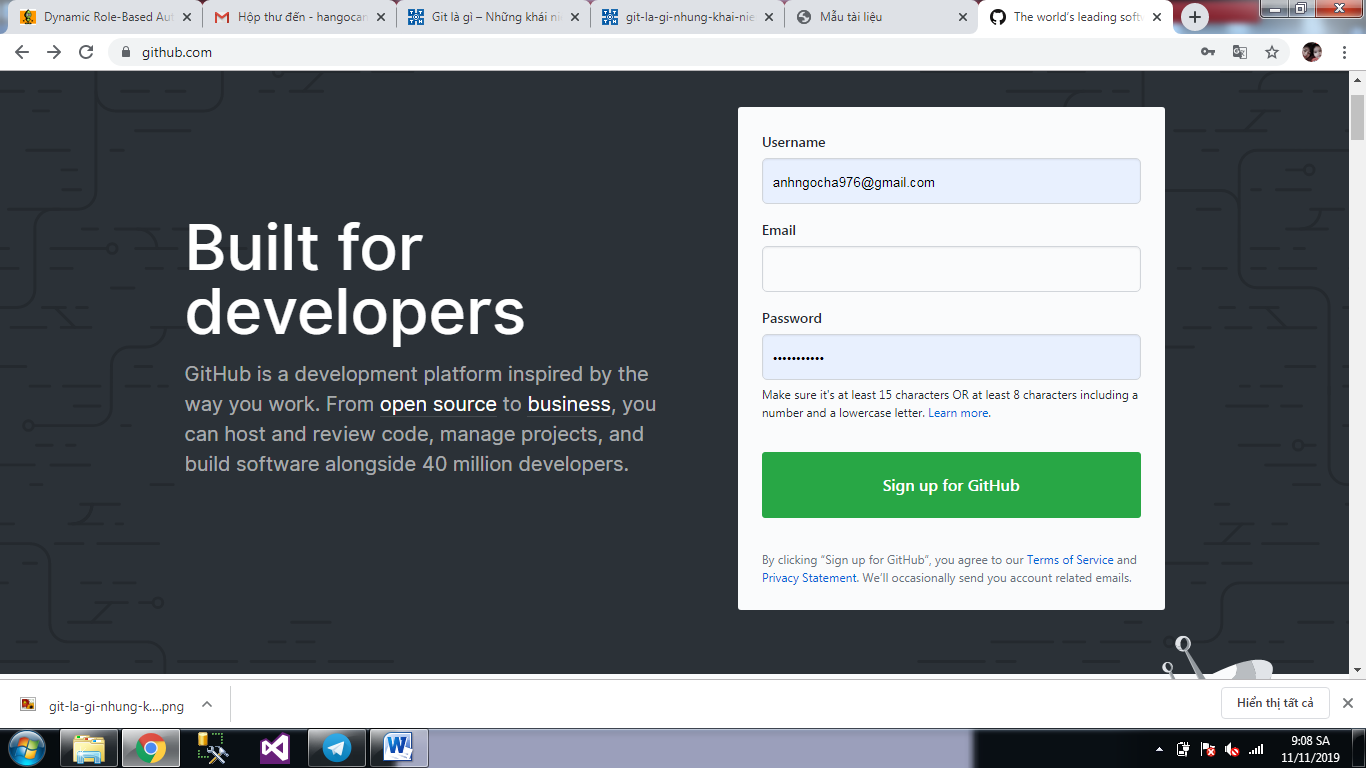
1. QA: Chuyên viên đảm bảo chất lượng của dự án
2. Tester: Kỹ sư kiểm thử phần mềm
3. Trưởng nhóm (Leader, LEAD): Trưởng nhóm lập trình
4. Developer (DEV): Kỹ sư lập trình
5. Admin : Người quản trị hệ thống Github

# Hướng dẫn sử dụng các lệnh cơ bản về Github

## Khởi tạo tài khoản user

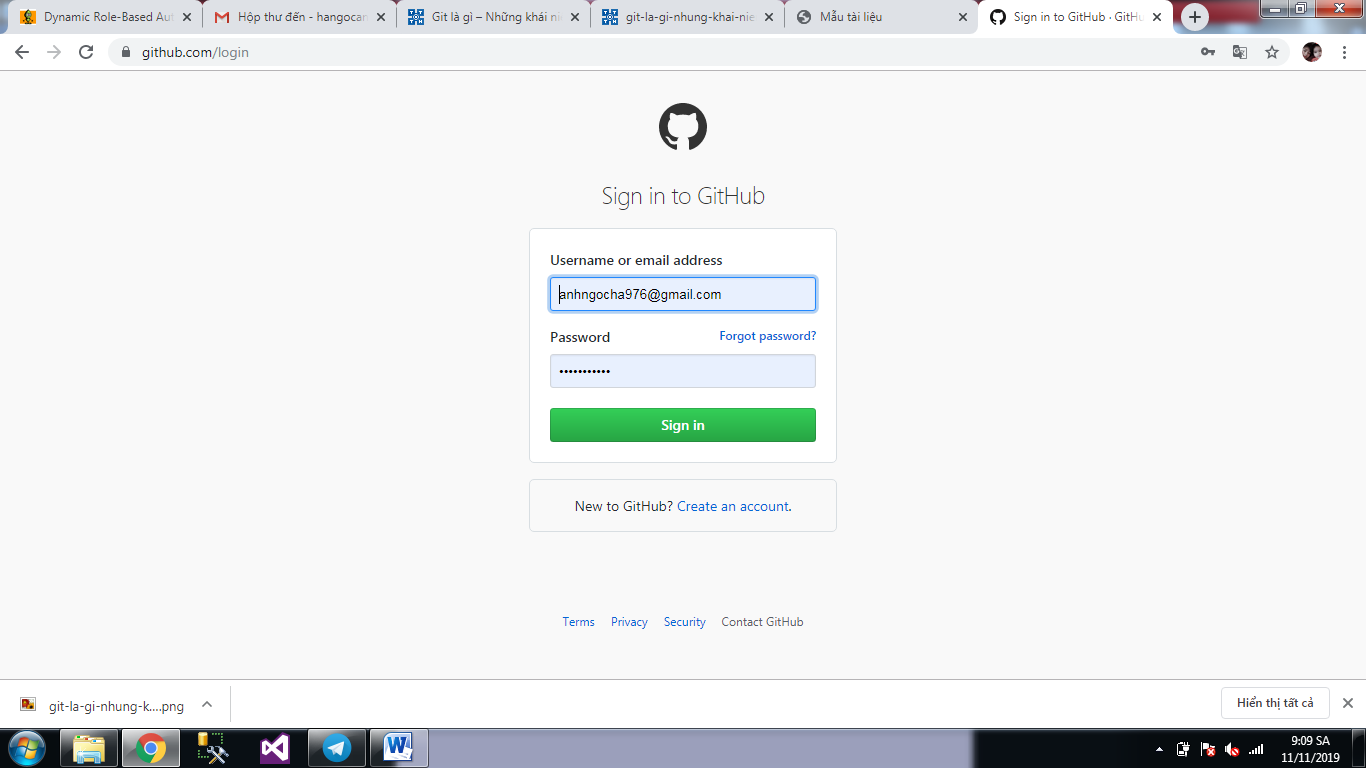
Để tạo tài khoản người dùng trên hệ thống ta truy cập vào website: <https://github.com>

* Giao diện đăng kí tài khoản



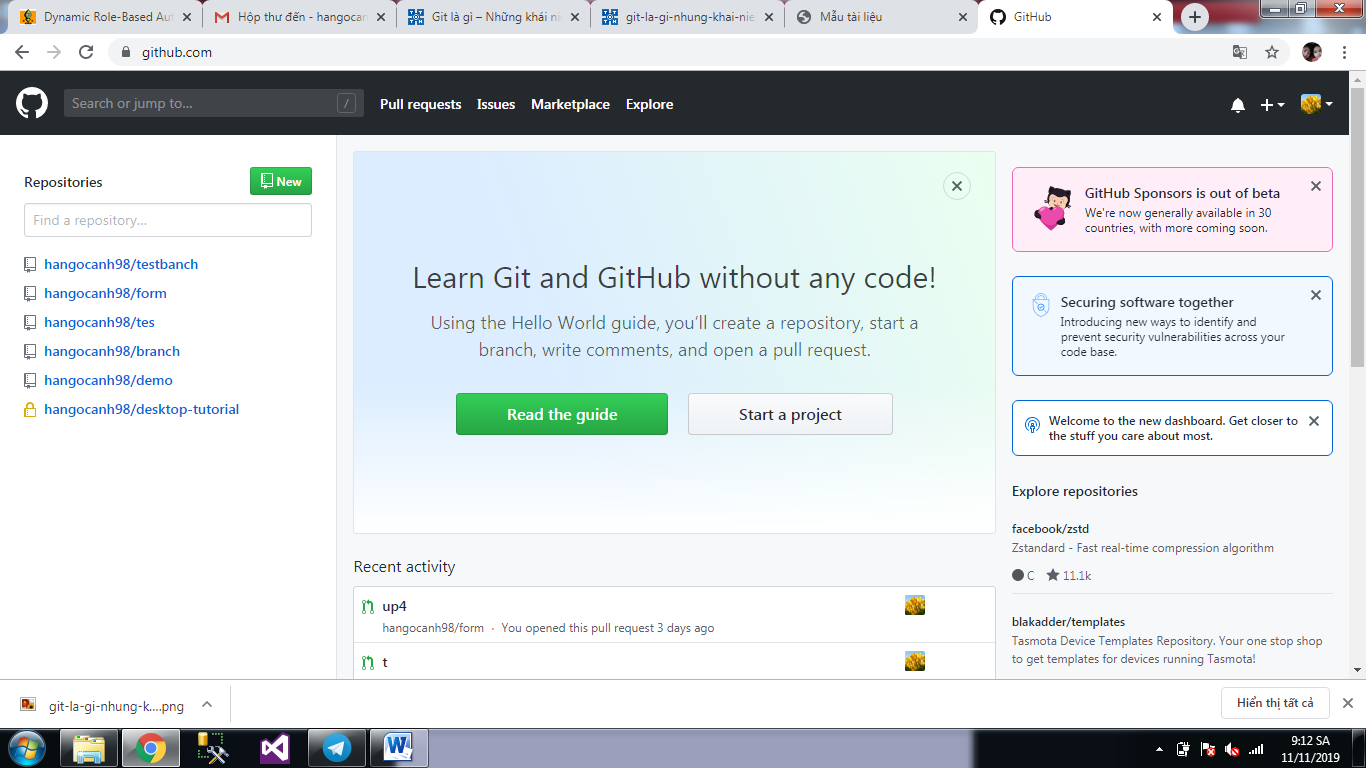
Hình 3: Đăng kí tài khoản

* Giao diện đăng nhập



Hình 4: Đăng nhập vào hệ thống

Sau khi đăng nhập thành công vào hệ thống giao diện sẽ hiển thị như sau:



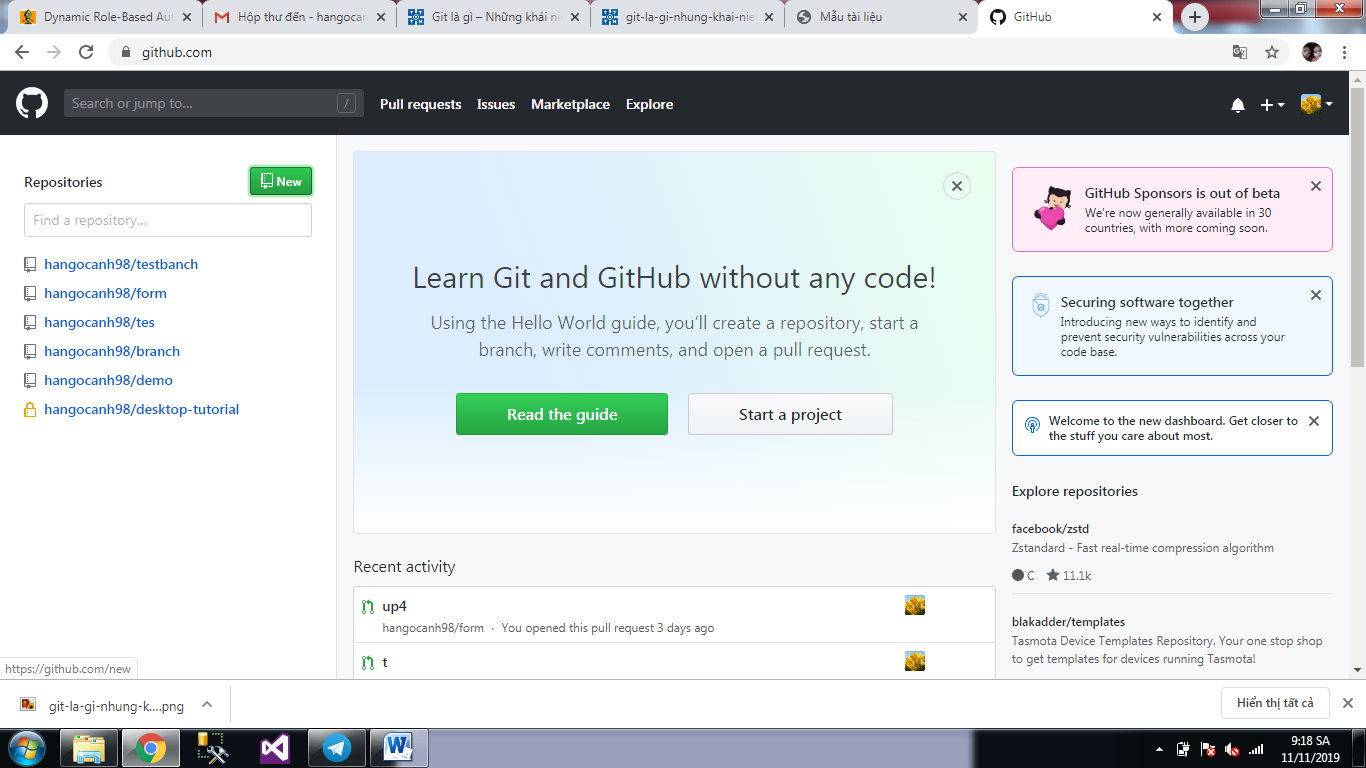
Hình 5: Trang chủ github

## Khởi tạo repository

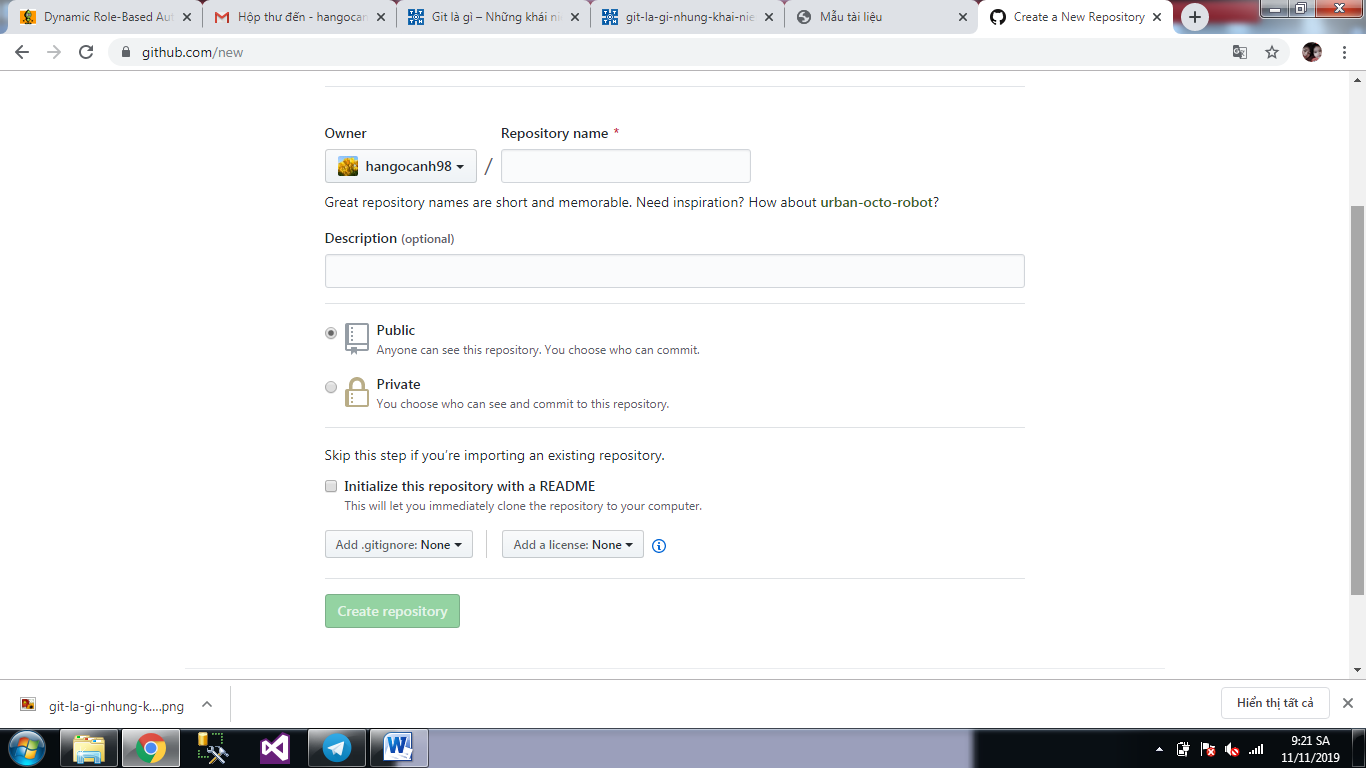
**Repository** là nơi lưu trữ, quản lý tất cả những thông tin cần thiết (thư mục, tệp tin, ảnh, video, bảng biểu, dữ liệu..) cũng như các sửa đổi, lịch sử của dự án

Để khởi tạo một repository như sau:

B1: Truy cập vào website và chọn New

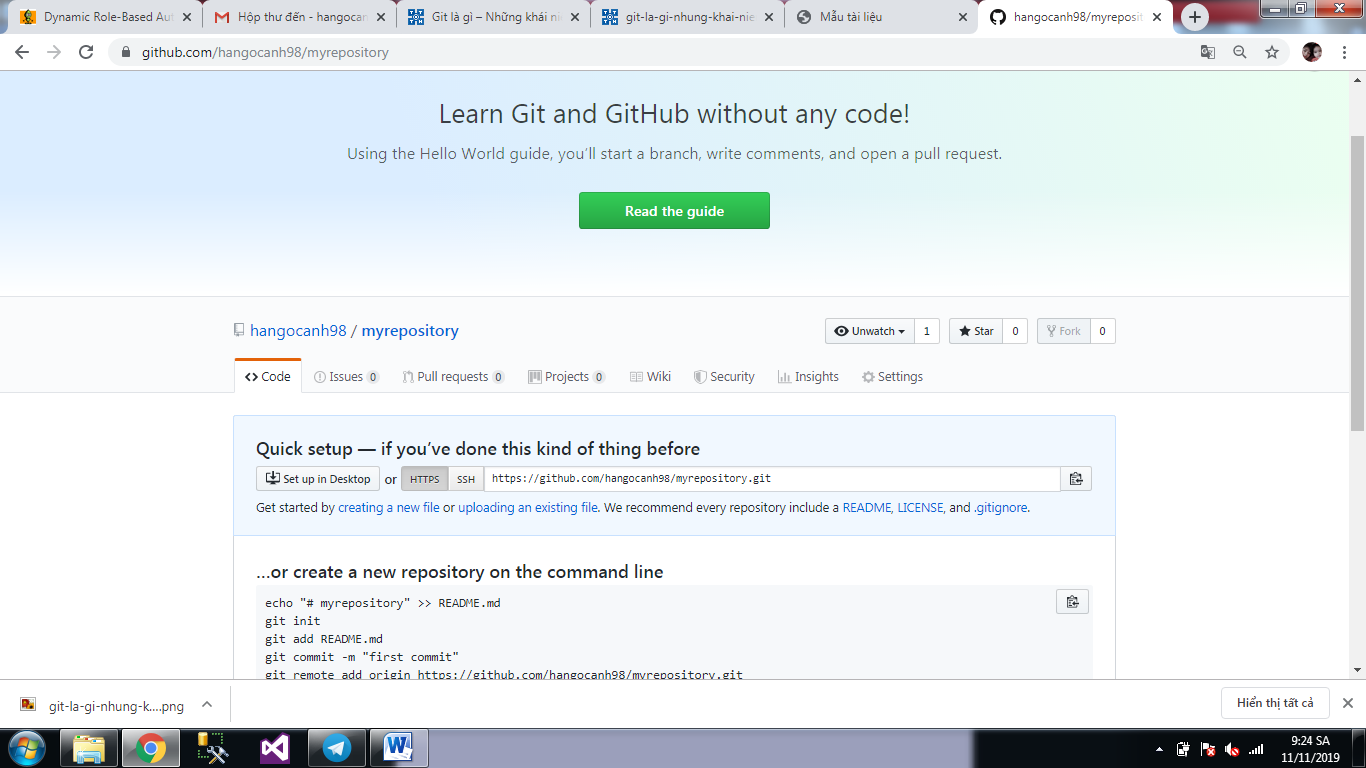


B2: Đặt tên project theo cú pháp được quy định và lựa chọn Visibility



Hình 6: Giao diện tạo mới repository

B3: Click chọn Create repository, github sẽ thực hiện khởi tạo repository và tự động chuyển hướng về repository mới được khởi tạo.

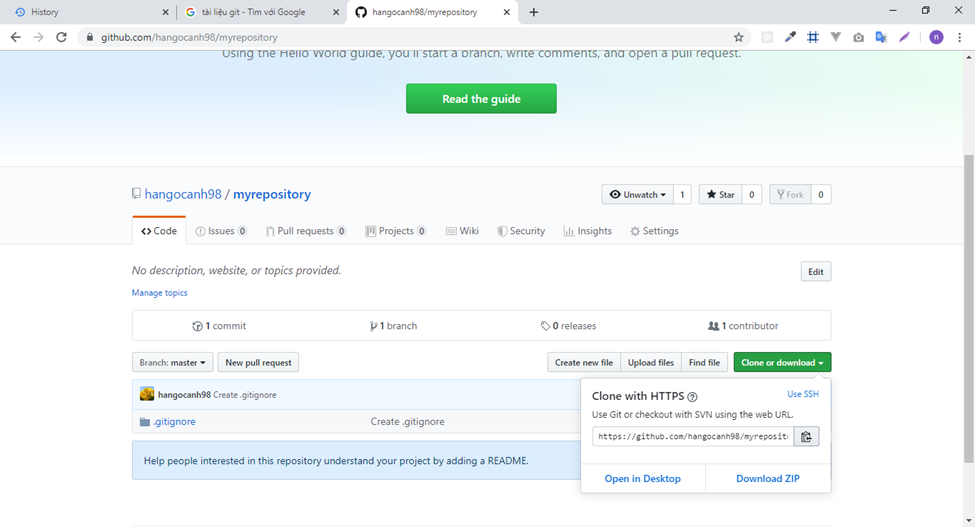


Hình 7: Khởi tạo thành công repository

## Clone project

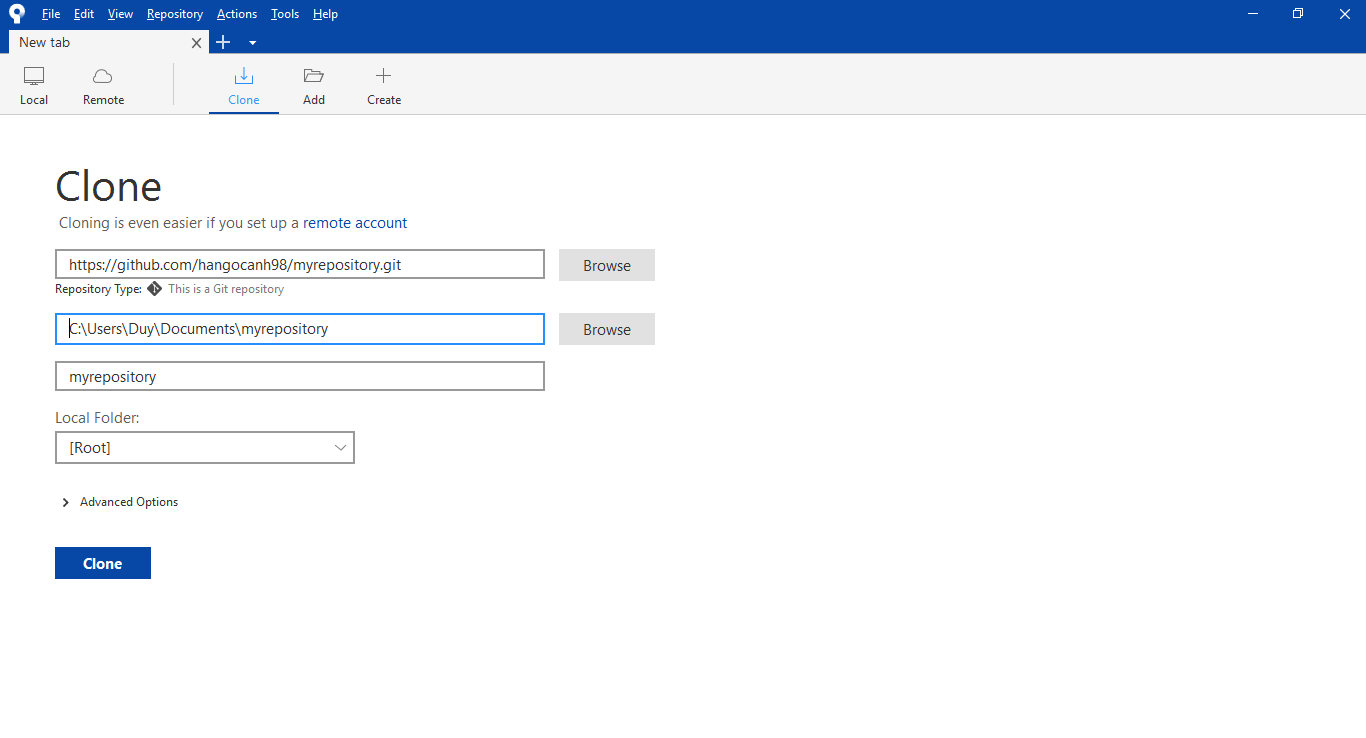
Clone là thao tác copy repo online của dev về máy tính

B1: Ckick vào clone or download để sao chép link



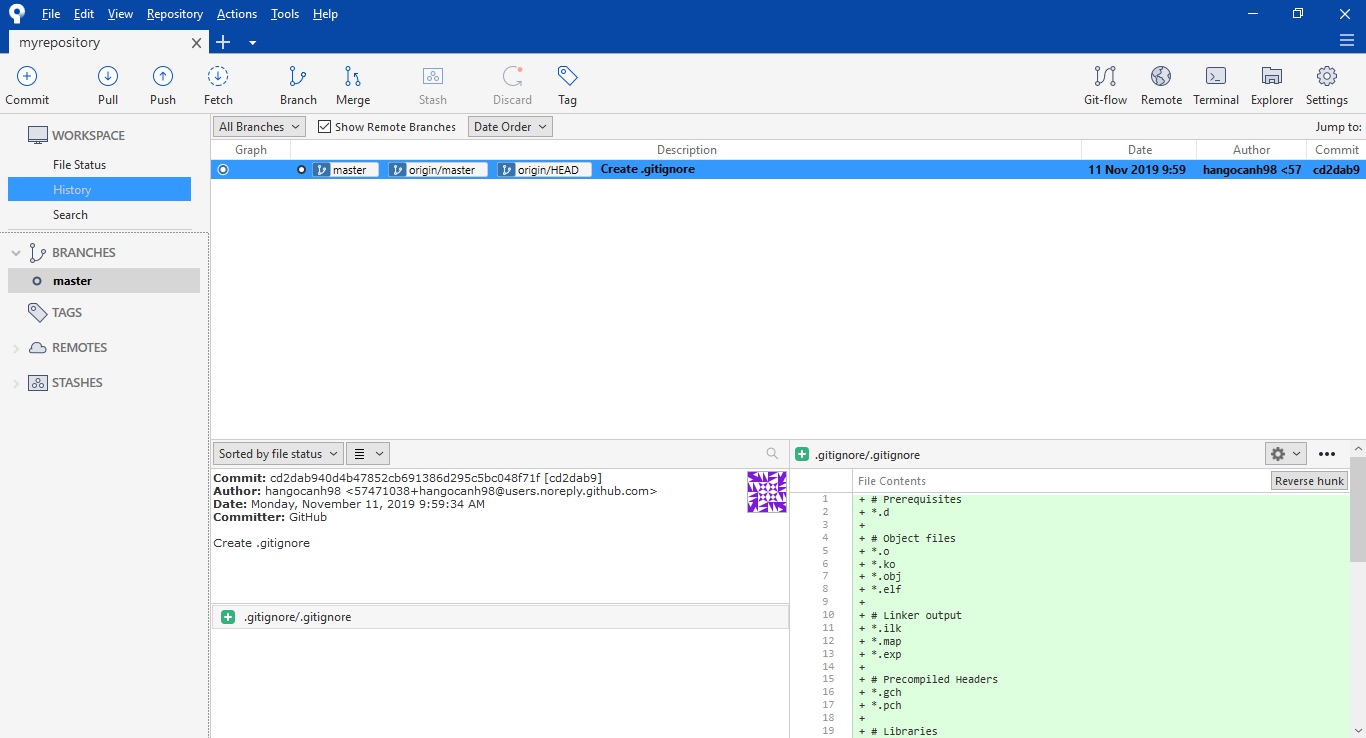
Hình 8: Copy URL git để thực hiện clone

B2: Cài đặt và mở SourceTree, chọn Clone và chèn link vừa copy vào và thực hiện clone



Hình 9: Add repository local vào SourceTree để quản lý

Sau đó SourceTree sẽ tự động lấy dữ liệu và hiển thị thành 1 tab quản lý như hình sau



Hình 10: Add vào SourceTree thành công

## Commit

**Commit** là thao tác báo cho hệ thống biết bạn muốn lưu lại trạng thái hiện hành, ghi nhận lại lịch sử các xử lý như thêm, xóa, cập nhật các file hay thư mục nào đó trên repository

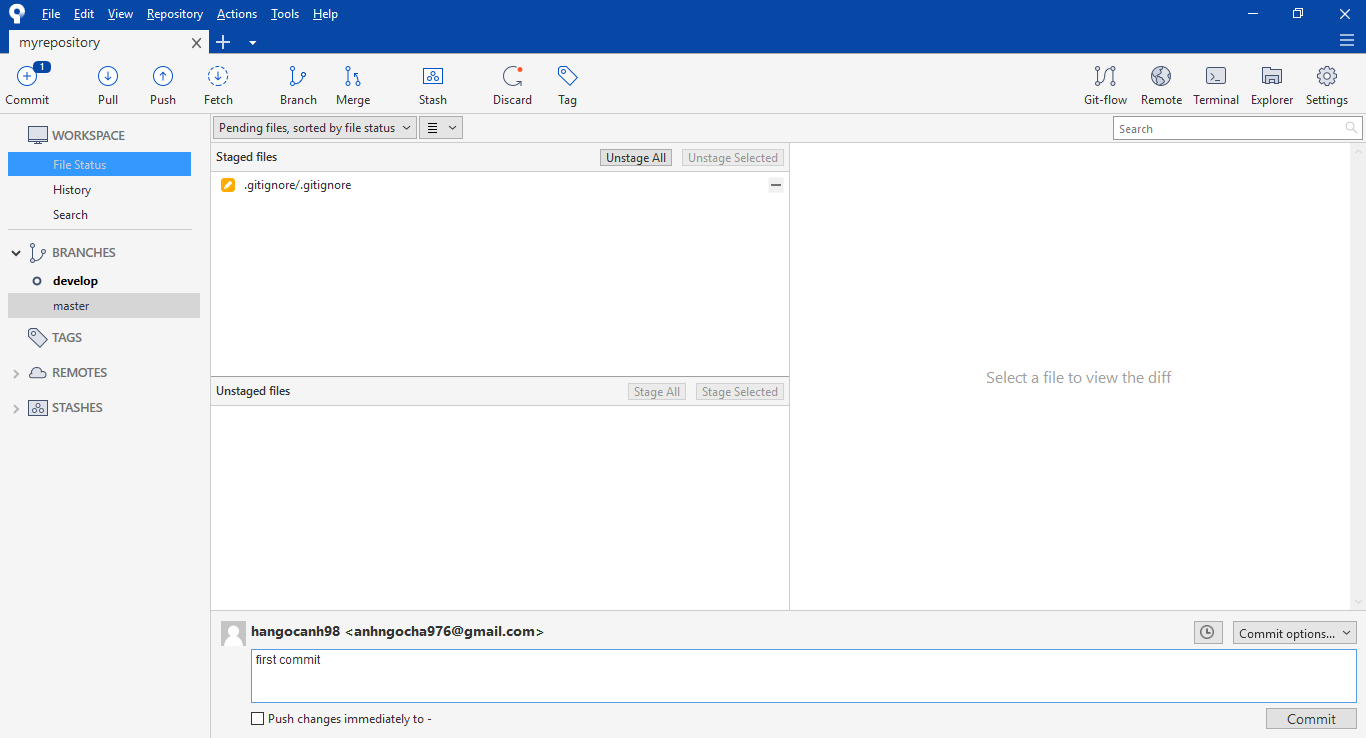
Khi thực hiện commit, hệ thống đều yêu cầu bạn phải nhập vào commit message để ghi chú những lần commit của bạn đã thực hiện những thay đổi nào

Thực hiện commit

Truy cập vào Working Copy và lựa chọn Stage những file cần upload lên.

Thực hiện nhập commit message và Click Commit

Mọi thông tin thay đổi đã được lưu lại tại kho lưu trữ cục bộ và chờ thao tác push để đưa kho lưu trữ cục bộ lên server

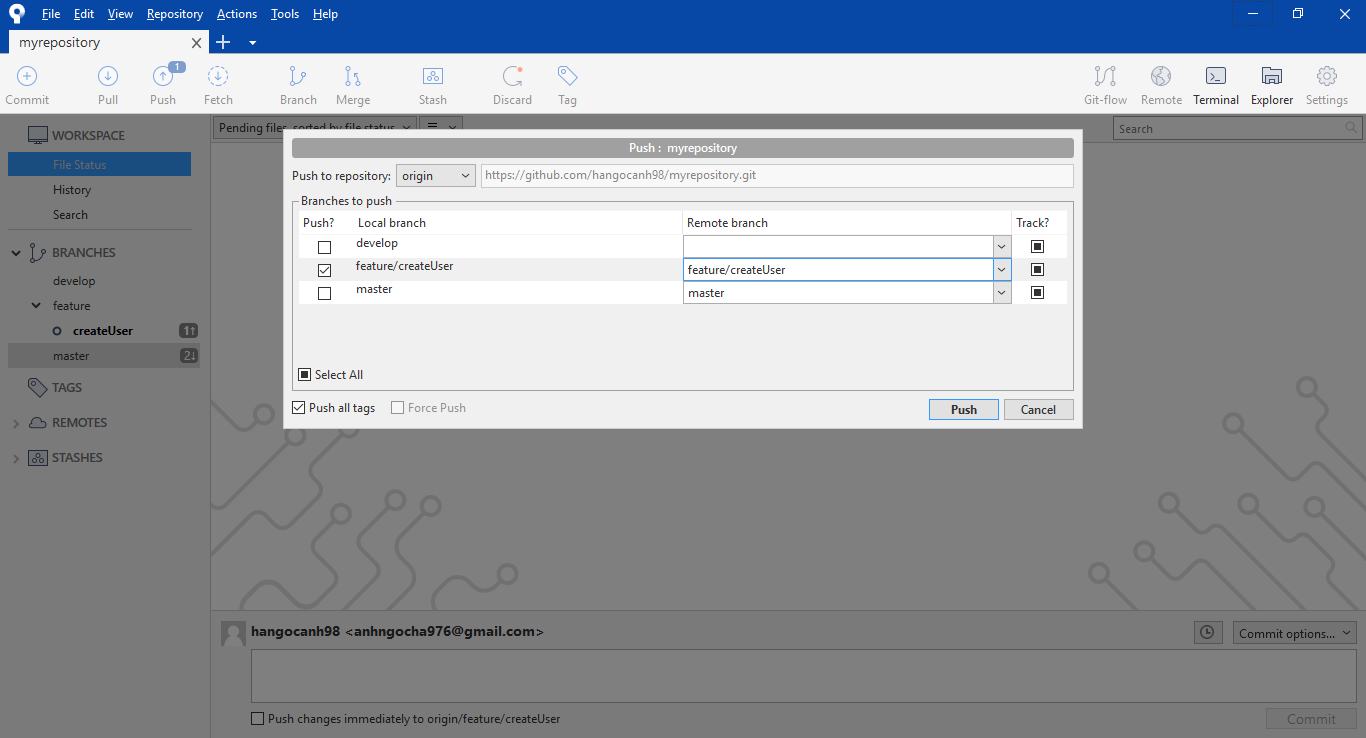


Hình 11: Commit thay đổi

## Push

Lệnh push được sử dụng để đưa nội dung lưu trữ cục bộ lên server. Push là cách bạn chuyển giao các commit từ kho lưu trữ cục bộ của bạn lên server

Trên giao diện SourceTree, click nút Push



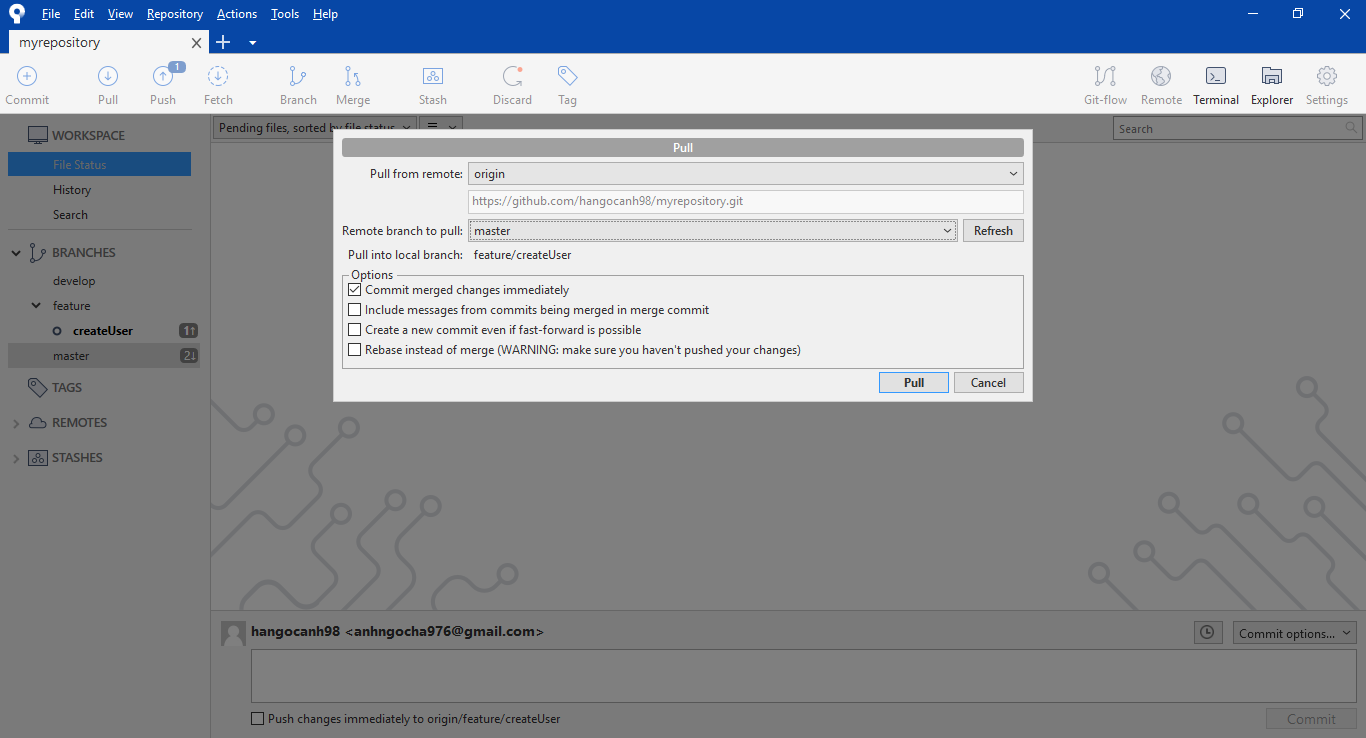
Hình 12: Push thay đổi lên repository

## Pull

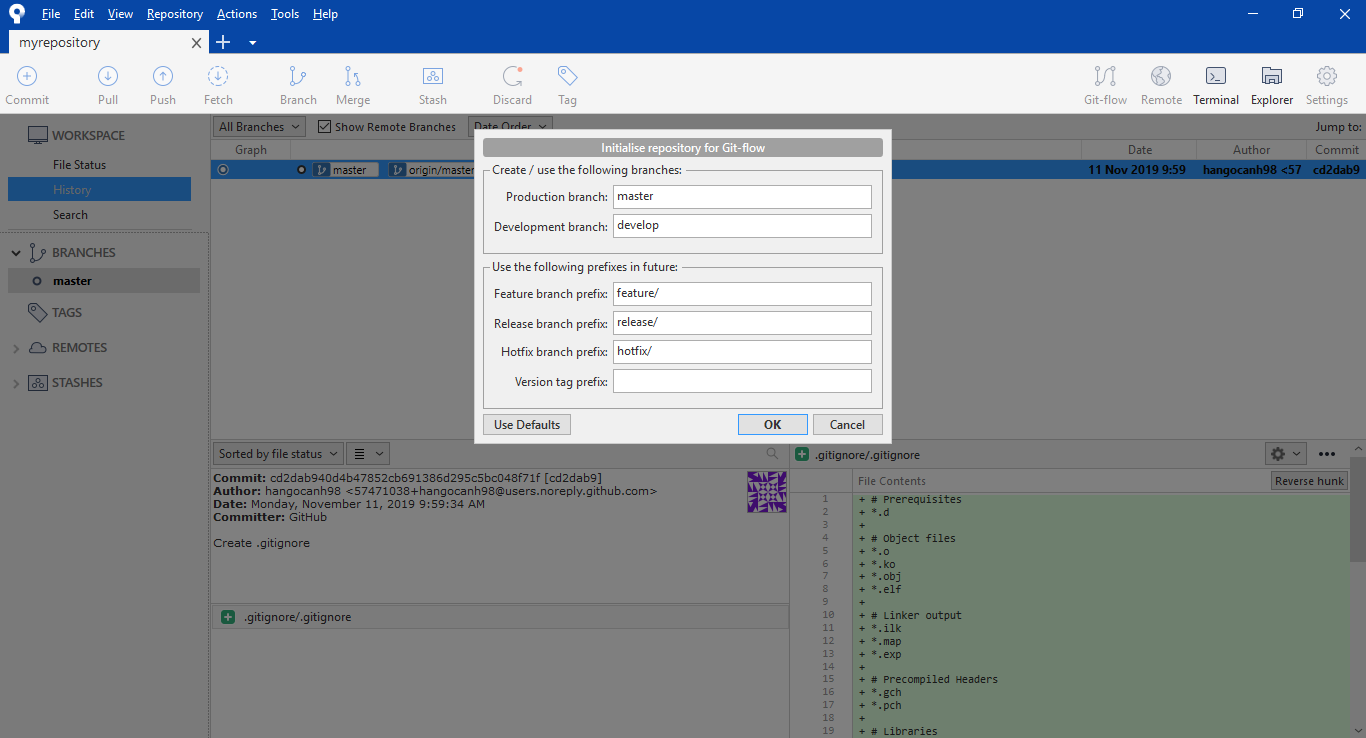
Để cập nhật local repository từ remote repository thì thực hiện thao tác gọi là Pull.

Khi thực hiện Pull, sẽ tải lịch sử thay đổi mới nhất từ remote repository về, rồi đưa nội dung đó vào local repository của bản thân.

Trên giao diện SourceTree, click nút Pull



Hình 13: Pull dữ liệu về

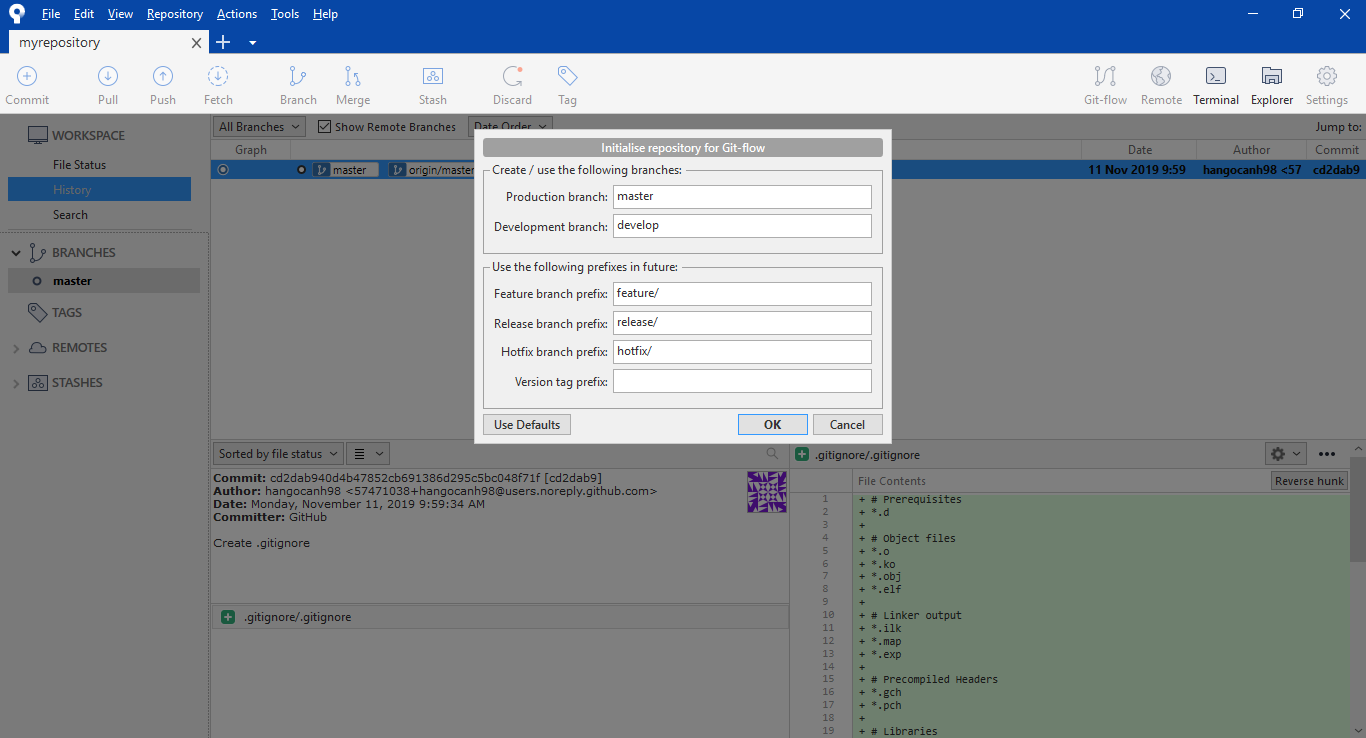


## Branch

Branch là một bản sao của repository. Bạn có thể sử dụng Branch để triển khai dự án theo hướng cô lập không ảnh hưởng đến dự án chính.

Làm việc với branch vì vậy sẽ không ảnh hưởng tới repostiroy chính hoặc những branches khác. Nếu bạn hoàn tất công việc, bạn có thể “Merge” (nhập) branch vào những branch khác khoặc repository chính bằng cách dùng lệnh Pull Request

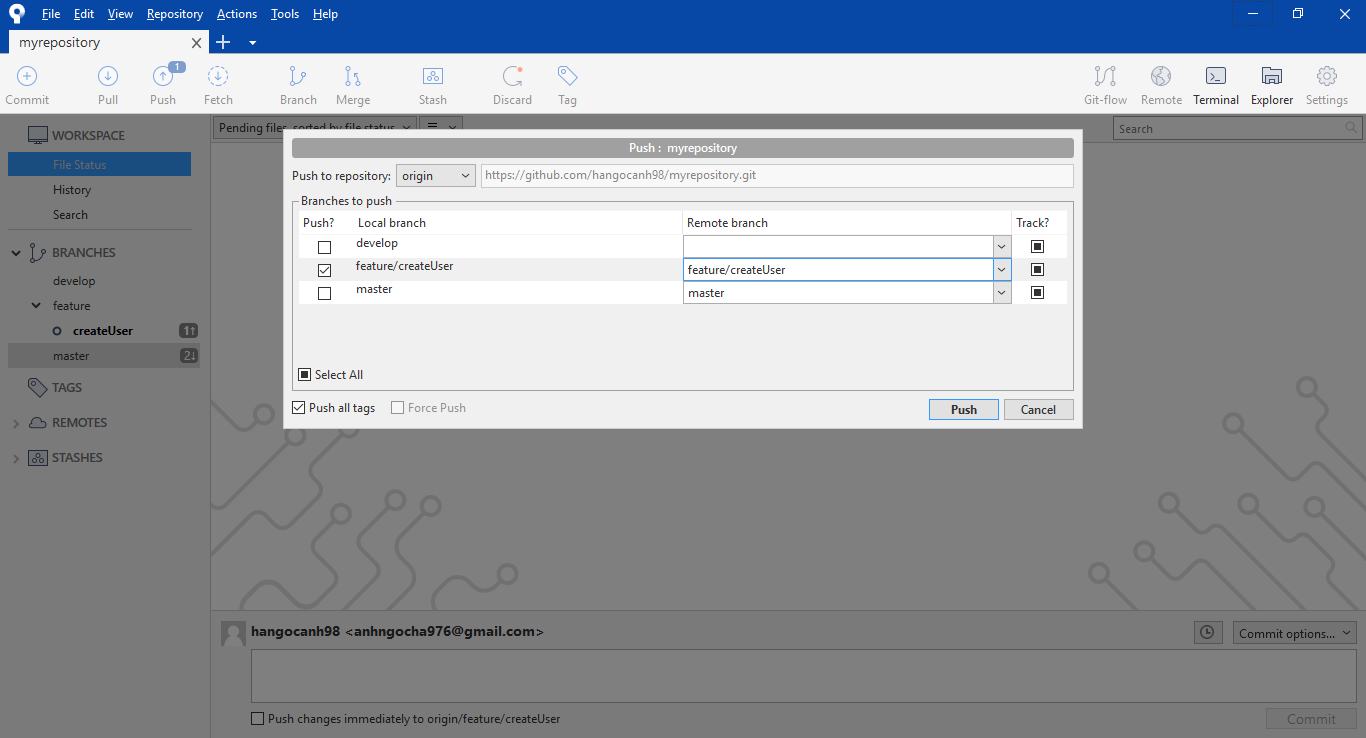
Thực hiện tạo branch:

B1: Trên giao diện SourceTree, click vào git- flow ta sẽ tạo được branch develop

Hình 14: Tạo Branch

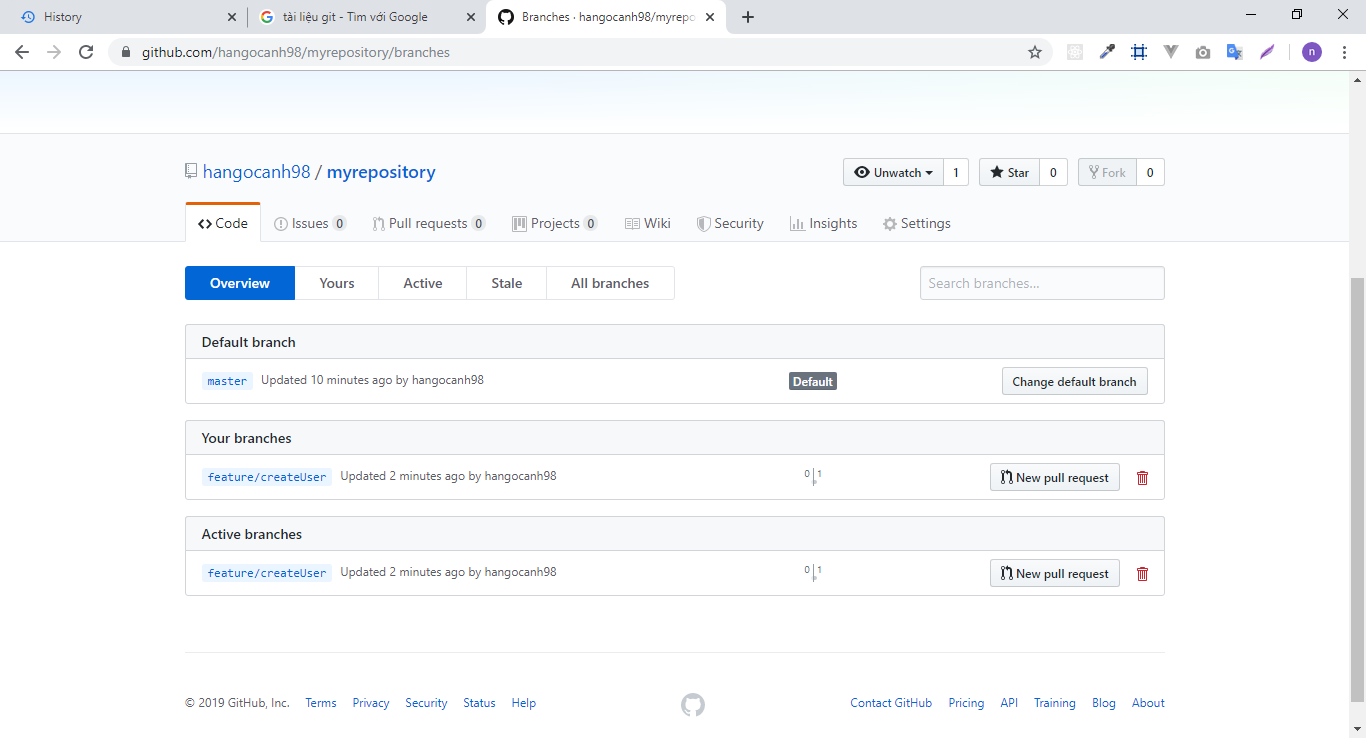
B2: Thực hiện commit

B3: Trên giao diện SourceTree, click nút Push, tích chọn vào branch feature/createUser mới tạo và bổ sung tính năng, bấm Push.



Hình 15: Push thay đổi lên serever

Truy cập website để kiểm tra branch feature/createUser đã được push thành công



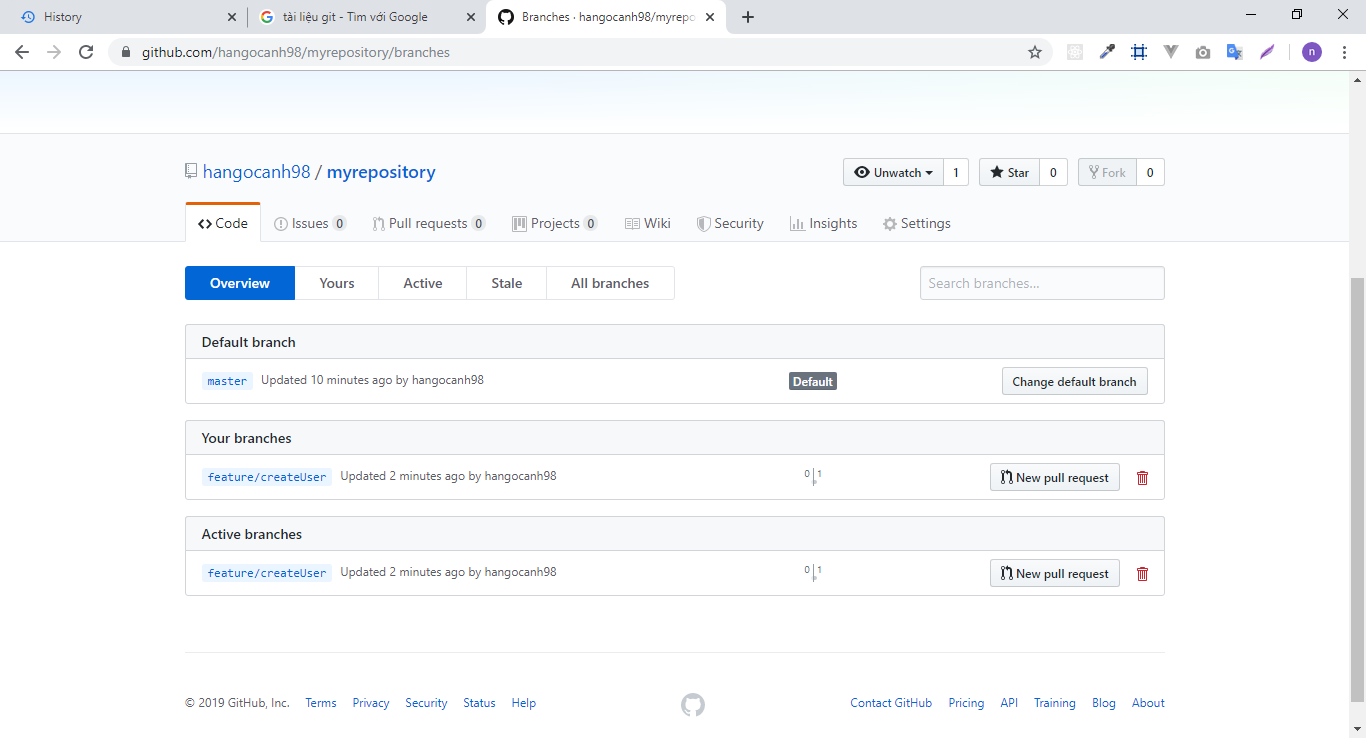
Hình 16: Tạo branch và push thành công

## Pull request

**Pull request** có nghĩa là bạn thông báo với những người khác rằng bạn đã đẩy những thay đổi của Branch lên Repository tổng (master respository). Các cộng tác viên của repository này sẽ có cập nhật hoặc từ chối pull request này. Khi nó được mở ra, bạn có thể thảo luận và xem lại công việc với những người cùng làm khác.

Các bước để tạo một pull request trong GitHub là:

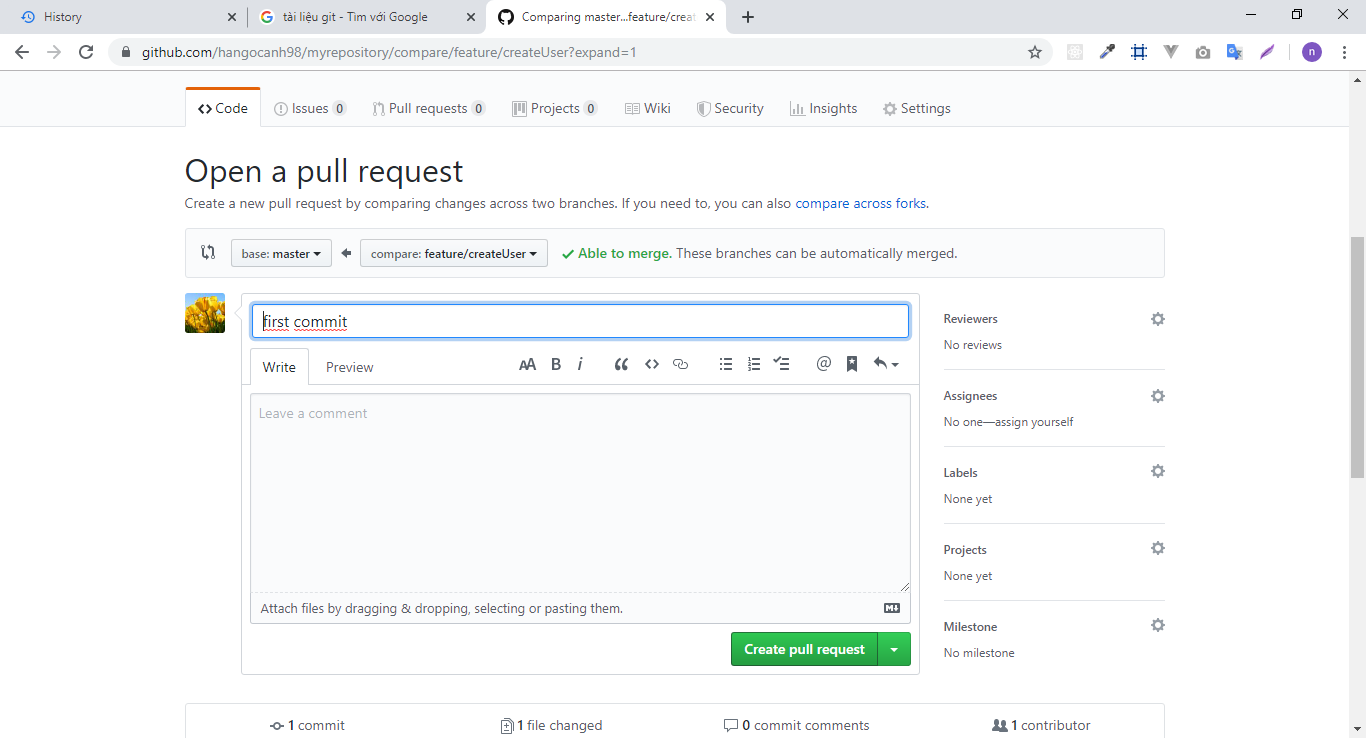
B1: Chuyển tới repository và tìm menu branch. Trong branch menu, chọn branch chứa thay đổi của bạn



Hình 17: New pull request

B2: Nhấn vào nút **New pull request** bên cạnh menu branch

B3: Thêm tiêu đề và mô tả vào pull request của bạn



Hình 18: Create pull request

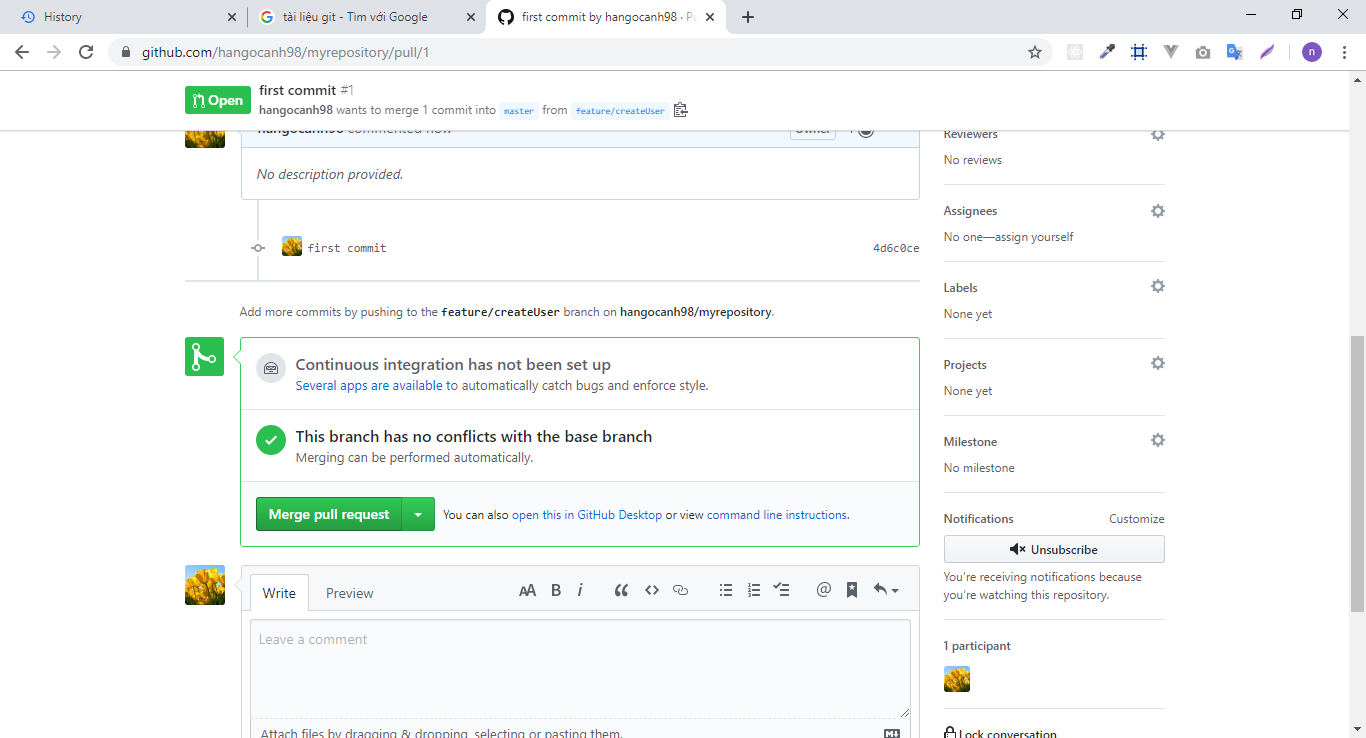
B4: Nhấn nút **Create pull request**

## Merge

Merge request là một yêu cầu cập nhật thay đổi vào kho dữ liệu chung, LEAD sẽ thực hiện review và quyết định có cho phép đưa các thay đổi này vào dứ án hay không.

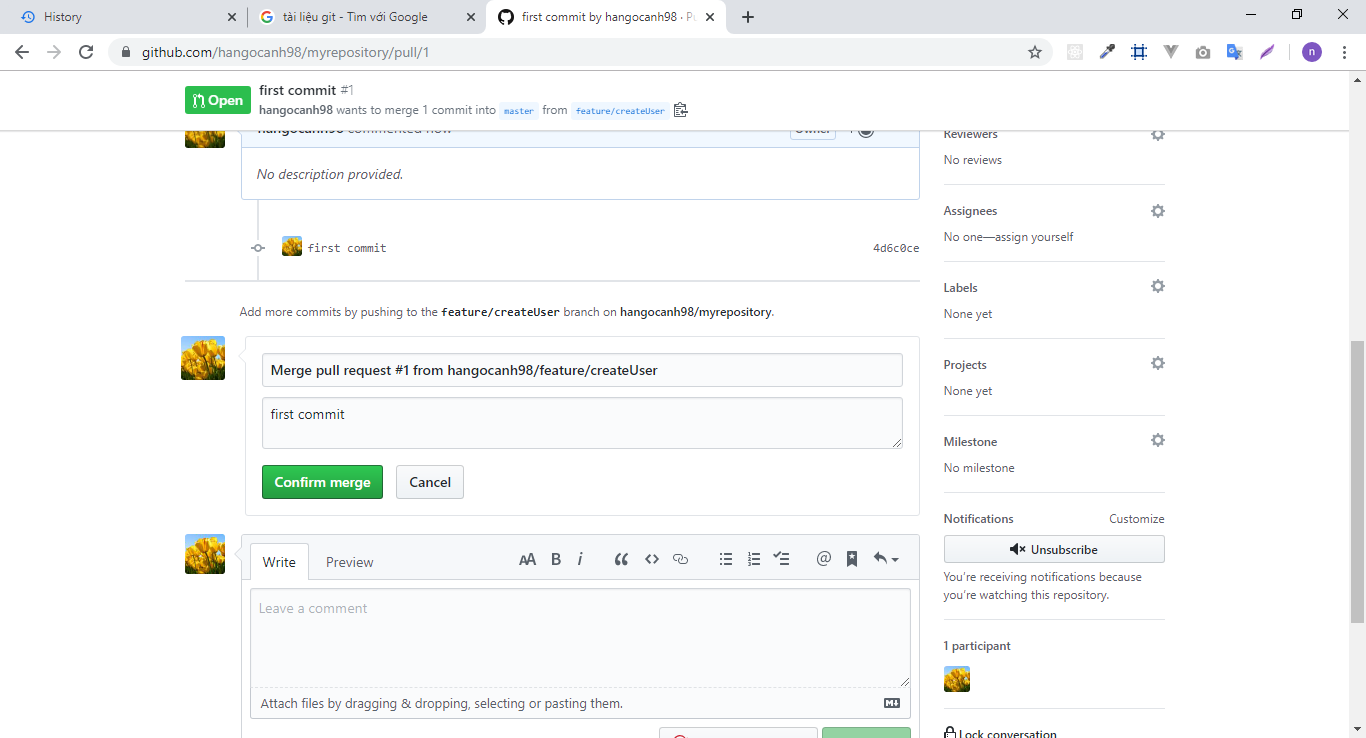
B1: Truy cập vào web

Khi nhận thấy thay đổi trên repo của dev so với dữ liệu fork về ban đầu, hệ thống sẽ tự động hiển thị thông báo tạo Merge Request trên màn hình.

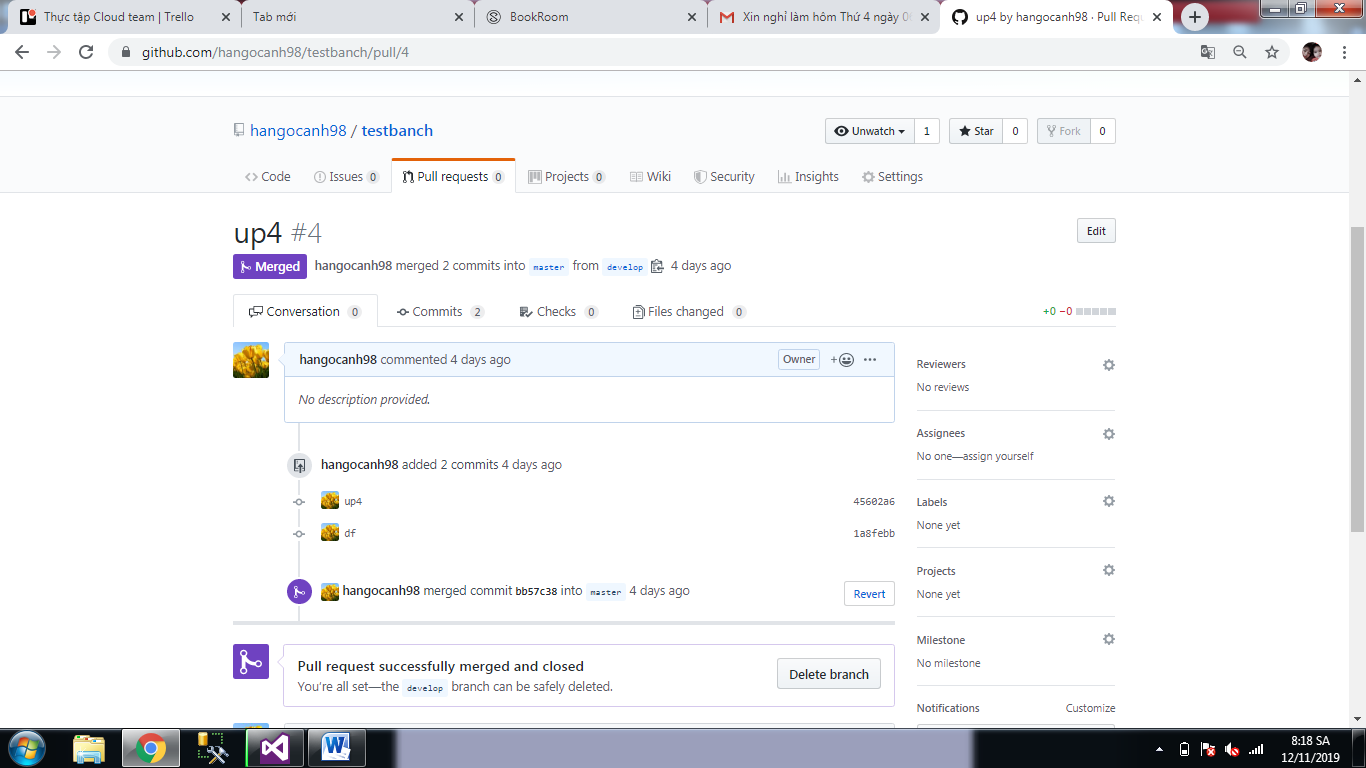


Hình 19: Merge request

B2: Click Merge pull request



B3: Cuối cùng click Confirm merge. Hệ thống sẽ hiển thị kết quả như sau chứng tỏ đã tạo merge request thành công.



# Tổng kết

Các kiến thức cần nhớ:

* Hiểu và nắm được cách hoạt động của github
* Tạo được repository
* Clone project
* Commit và push
* Pull và Pull request
* Branch
* Merge

# Tài liệu tham khảo

[1] Hướng dẫn Git: <https://backlog.com/git-tutorial/vn/intro/intro1_1.html>

[2] Thực hành Git online: <https://learngitbranching.js.org/>

[3] <https://jordankasper.com/lessons-learned-teaching-git/>

## Phụ lục

***SourceTree*** *là công cụ Git Client kèm theo GUI, hoàn toàn miễn phí, giúp đơn giản hóa việc quản lý mã nguồn Git, đặc biệt SourceTree tích hợp sẵn việc quản lý bằng Gitflow và cho phép tích hợp công cụ xử lý xung đột mã nguồn KDiff3.*

*Tài liệu hướng dẫn này sẽ sử dụng SourceTree làm công cụ Git client chính (thay vì sử dụng câu lệnh như nhiều hướng dẫn online khác). Ngoài ra, có một số ít thao tác khi xử lý conflict vẫn phải sử dụng bằng command line.*

1. Link download source tree : <https://www.sourcetreeapp.com/>
2. Link download Kdiff3: <https://sourceforge.net/projects/kdiff3/>