**Tìm hiểu phần mềm quản lý nhóm (phiên bản) – GitHub**

GitHub là một dịch vụ cung cấp kho lưu trữ mã nguồn Git dựa trên nền web cho các dự án phát triển phần mềm. GitHub cung cấp cả phiên bản trả tiền lẫn miễn phí cho các tài khoản.

**Tại sao nên sử dụng Git?**

Có rất nhiều lợi thế để bạn nên sử dụng Git trong việc lập trình ngay từ hôm nay, bất kể là lập trình cái gì đi chăng nữa.

* Git dễ sử dụng, an toàn và nhanh chóng.
* Có thể giúp quy trình làm việc code theo nhóm đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch).
* Bạn có thể làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ kho chứa hoặc clone một phiên bản thay đổi nào đó từ kho chứa, hoặc một nhánh nào đó từ kho chứa.
* Dễ dàng trong việc deployment sản phẩm.
* Và nhiều hơn thế nữa.

**1. Nguyên tặc hoạt động**

1. Đăng ký một tài khoản GitHub và tạo một Repository (GitHub Repository).
2. Cài đặt GitHub Desktop, một công cụ trực quan quản lý Local Repository (Kho chứa dữ liệu địa phương).
3. Cấu hình để có thể đồng bộ hóa dữ liệu bằng GitHub Desktop lên Repository server.

**2. Khái niệm cơ bản trong Git**

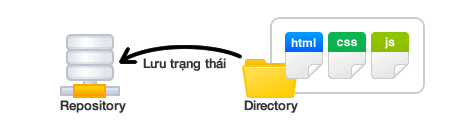
Git là một trong những Hệ thống Quản lý Phiên bản Phân tán, vốn được phát triển nhằm quản lý mã nguồn (source code) hữu hiệu của Linux.Git là một trong những Hệ thống Quản lý Phiên bản Phân tán, vốn được phát triển nhằm quản lý mã nguồn (source code) hữu hiệu của Linux.

Trên Git, có thể lưu trạng thái của file khi có nhu cầu dưới dạng lịch sử cập nhật. Vì thế, có thể đưa file đã chỉnh sửa một lần về trạng thái cũ hay có thể hiển thị sự khác biệt ở nơi chỉnh sửa.

Thêm nữa, khi định ghi đè (overwrite) lên file mới nhất đã chỉnh sửa của người khác bằng file đã chỉnh sửa dựa trên file cũ, thì khi đăng (upload) lên server sẽ hiện ra cảnh cáo. Vì thế, sẽ không xảy ra thất bại về việc đã ghi đè lên nội dung chỉnh sửa của người khác mà không hề hay biết.

**2.1 Repository quản lý lịch sử**

Repository là nơi sẽ ghi lại trạng thái của thư mục và file. Trạng thái được lưu lại đang được chứa như là lịch sử thay đổi của nội dung. Bằng việc đặt thư mục muốn quản lý lịch sử thay đổi dưới sự quản lý của repository, có thể ghi chép lại lịch sử thay đổi của thư mục và file trong thư mục đó.

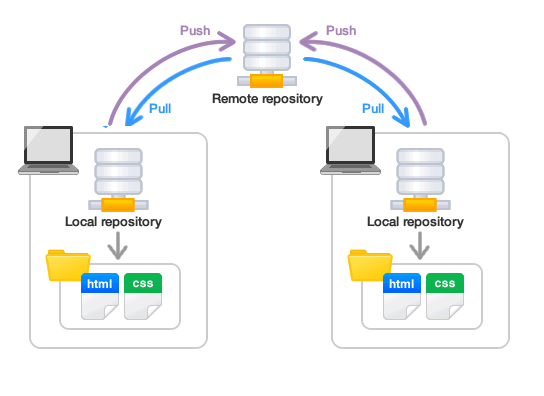


**2.2 Remote repository và local repository**

Đầu tiên, repository của Git được phân thành 2 loại là remote repository và local repository.

* Remote repository: Là repository để chia sẻ giữa nhiều người và bố trí trên server chuyên dụng.
* Local repository: Là repository bố trí trên máy của bản thân mình, dành cho một người dùng sử dụng.

Do repository phân thành 2 loại là local và remote nên với những công việc bình thường thì có thể sử dụng local repository và thực hiện trên toàn bộ máy sẵn có. Khi muốn công khai nội dung công việc mà bản thân đã làm trên local repository, thì sẽ upload lên remote repository rồi công khai. Thêm nữa, thông qua remote repository cũng có thể lấy về nội dung công việc của người khác.

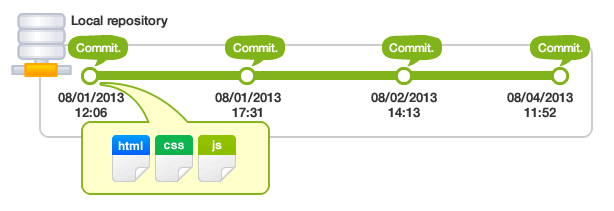


**2.3 Commit sẽ ghi lại thay đổi**

Để ghi lại việc thêm/ thay đổi file hay thư mục vào repository thì sẽ thực hiện thao tác gọi là Commit.

Khi thực hiện commit, trong repository sẽ tạo ra commit (hoặc revision) đã ghi lại sự khác biệt từ trạng thái đã commit lần trước đến trạng thái hiện tại.

Commit này đang được chứa tại repository trong trạng thái đã nối tiếp với nhau theo thứ tự thời gian như trong hình dưới đây. Bằng việc lần theo commit này từ trạng thái mới nhất thì có thể hiểu được lịch sử thay đổi trong quá khứ hoặc nội dung thay đổi đó.



**2.4 Working Tree và Index**

Trên Git, những thư mục được đặt trong sự quản lý của Git mà mọi người đang thực hiện công việc trong thực tế được gọi là working tree.

Và trên Git, giữa repository và working tree tồn tại một nơi gọi là index. Index là nơi để chuẩn bị cho việc commit lên repository.

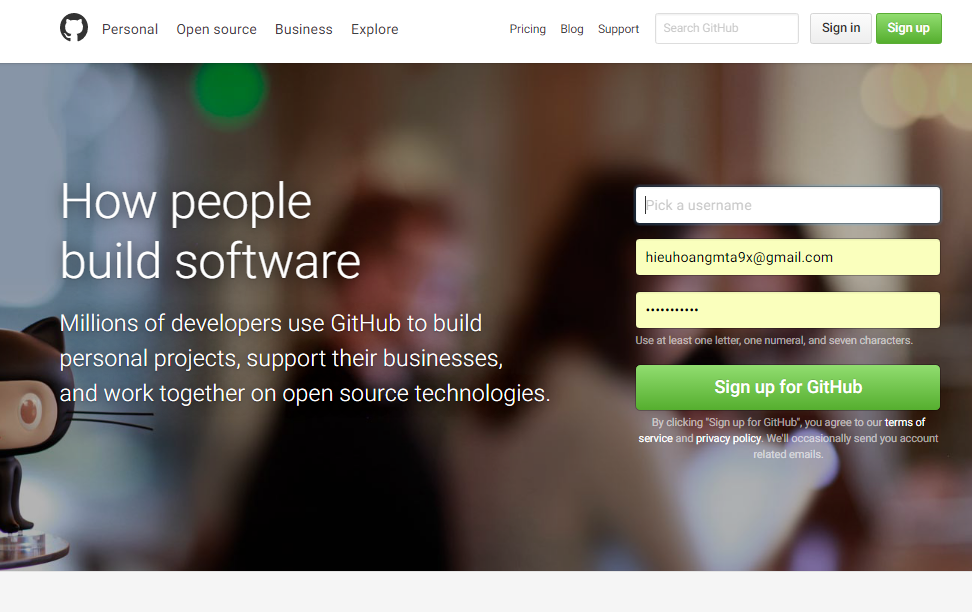
Trên Git, khi đã thực hiện commit thì trạng thái sẽ không được ghi trực tiếp trong repository từ working tree, mà sẽ ghi trạng thái đã được thiết lập của index được xây dựng ở giữa đó. Vì thế, để ghi lại trạng thái của file bằng commit thì trước hết cần đăng ký file trong index.

Với việc chèn index vào giữa như thế này, có thể thực hiện commit không bao gồm những file không cần thiết trong working tree, hay có thể đăng ký chỉ một phần thay đổi của file trong index rồi commit.

**3. Đăng ký tài khoản GitHub**

Trước hết bạn cần phải đăng ký miễn phí một tài khoản GitHub. Bạn có thể vào trang chủ của GitHut tại:

* <https://github.com>.



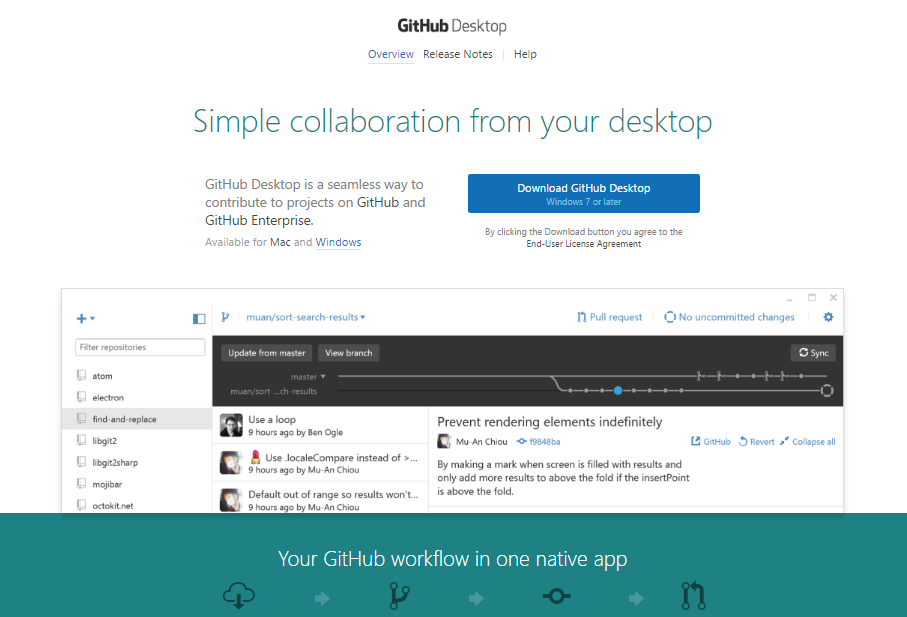
Việc đăng ký một tài khoản là đơn giản, bạn chỉ cần nhập username/password và địa chỉ email. Sau khi đăng ký xong bạn cần vào Email kích hoạt tài khoản. Mọi việc hoàn thành.

**4. Download & cài đặt GitHub Desktop**

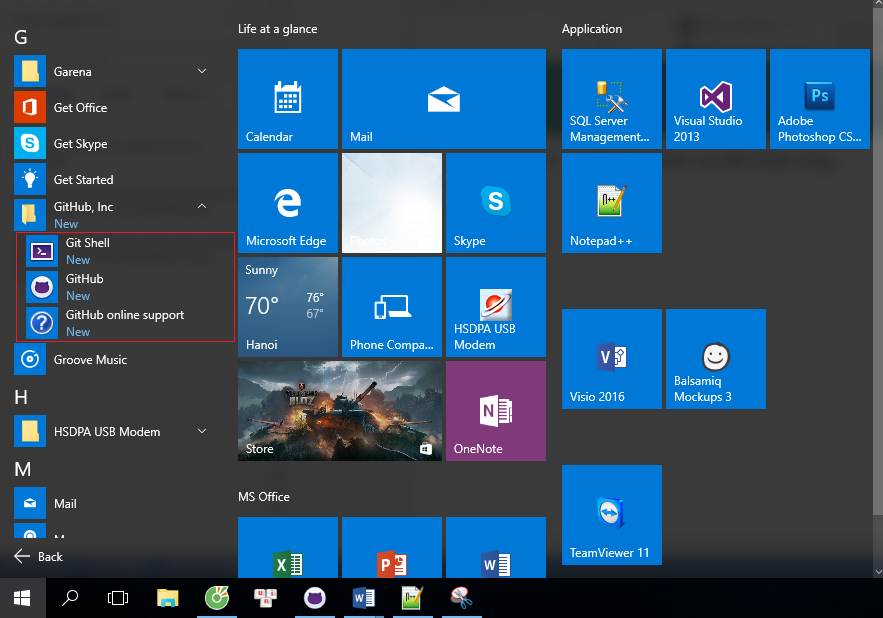
GitHub Desktop về bản chất là một công cụ trực quan cho phép bạn quản lý Local Repository (Kho chứa địa phương) trên máy tính của bạn.

Để download GitHub Desktop bạn vào địa chỉ:

* <https://desktop.github.com/>

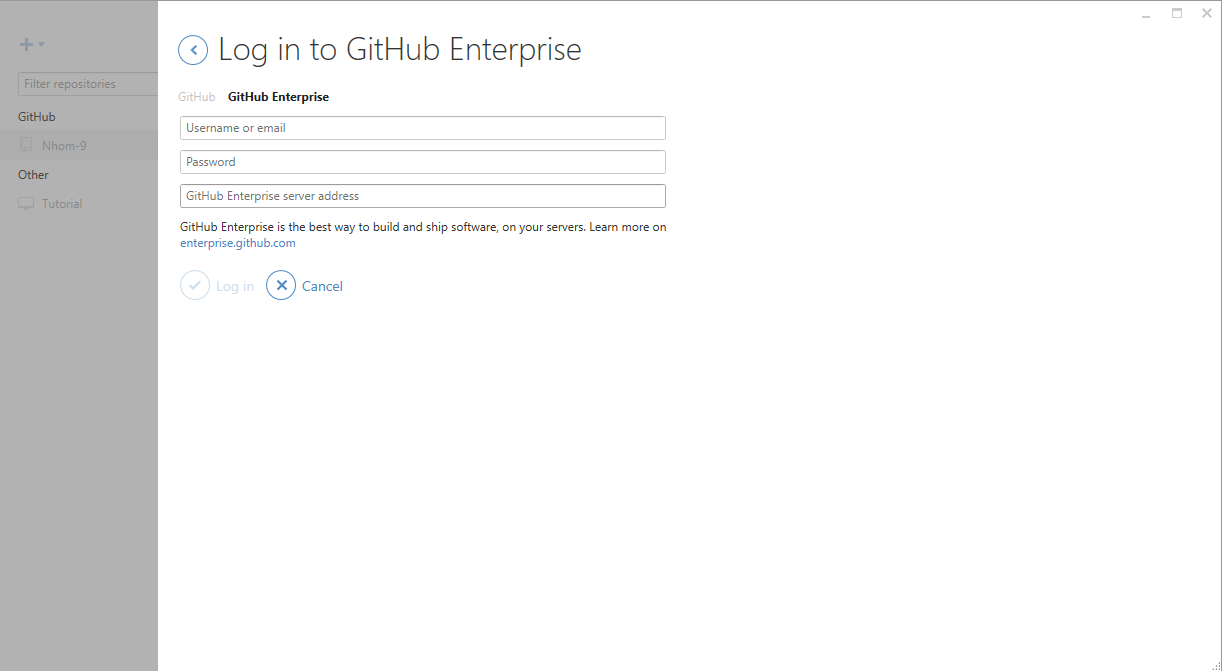


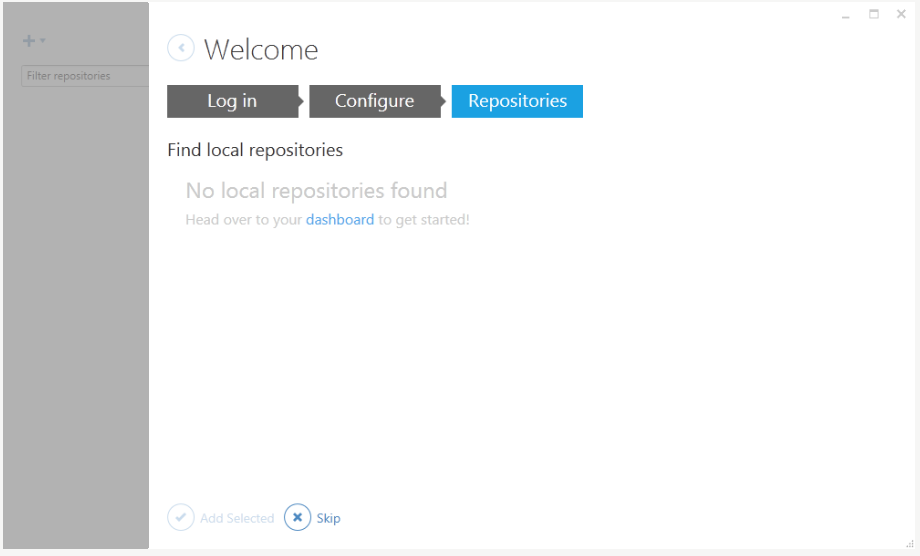
* GitHub đã được cài đặt thành công.



**5. Chạy GitHub Desktop**

Đăng nhập trên GitHub Desktop để kết nối vào tài khoản GitHub của bạn.



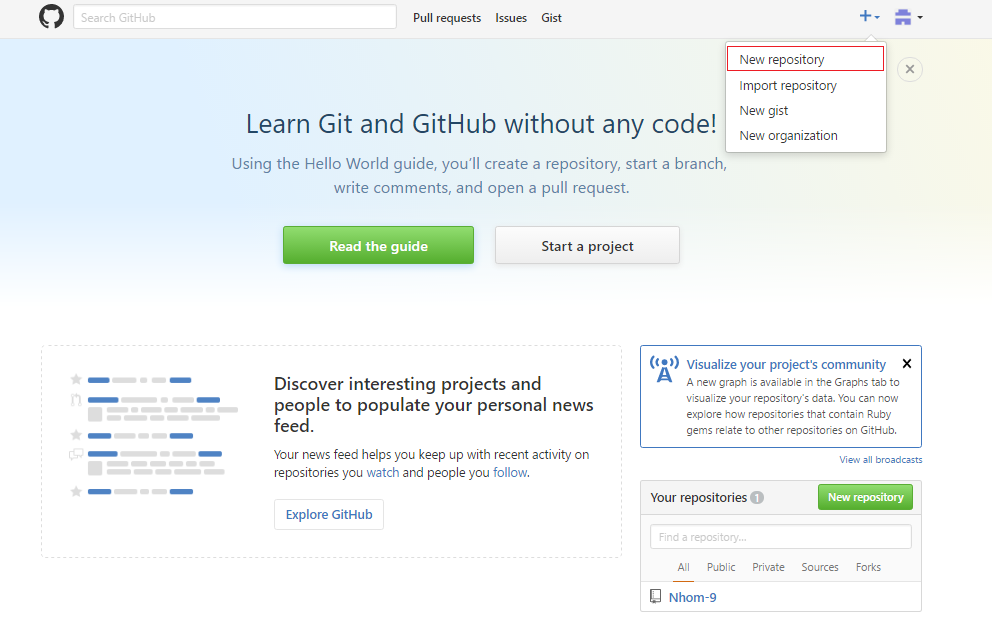


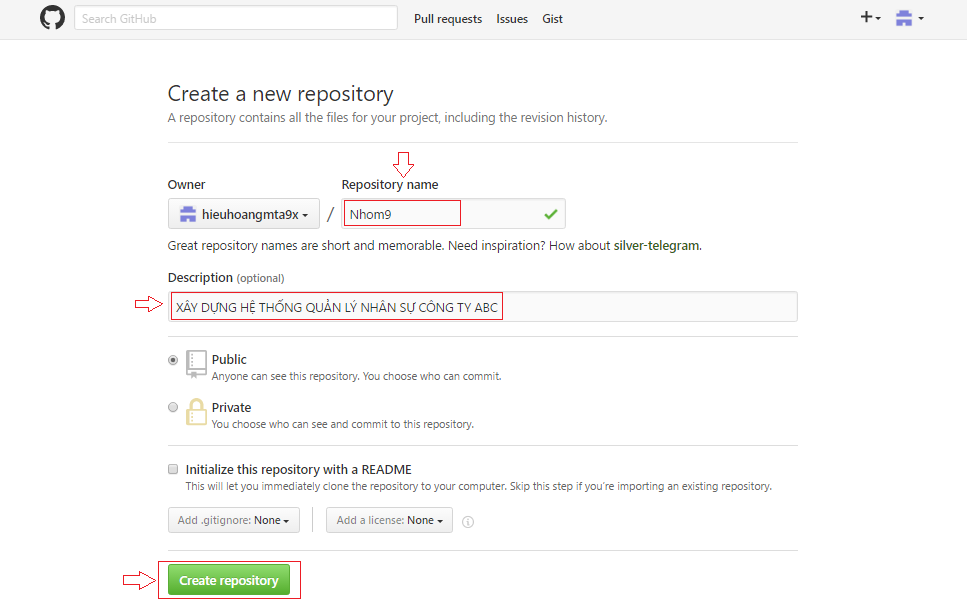
Cho tới lúc này trên máy tính của bạn chưa có một Local Repository nào.

**6. Tạo GitHub Repository**

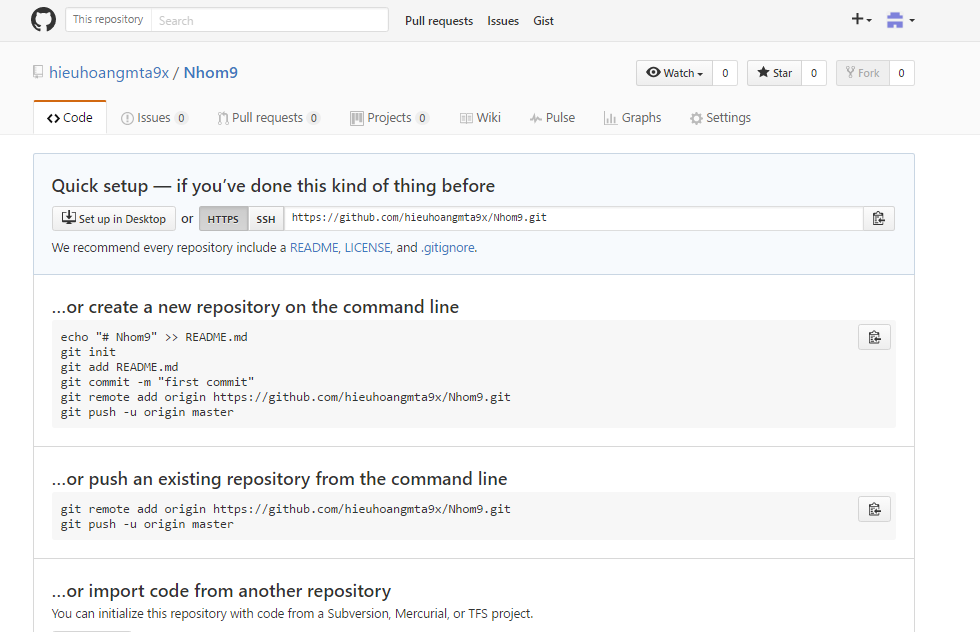
Sau khi đăng ký xong tài khoản GitHub, và đăng nhập vào. Bạn có thể tạo một GitHub Repository.

**Tạo remote repository trên Backlog**





Repository đã được tạo ra.



Ta có được địa chỉ Server(Tên host) chứa mã nguồn dự án:

* <https://github.com/hieuhoangmta9x/Nhom9.git>

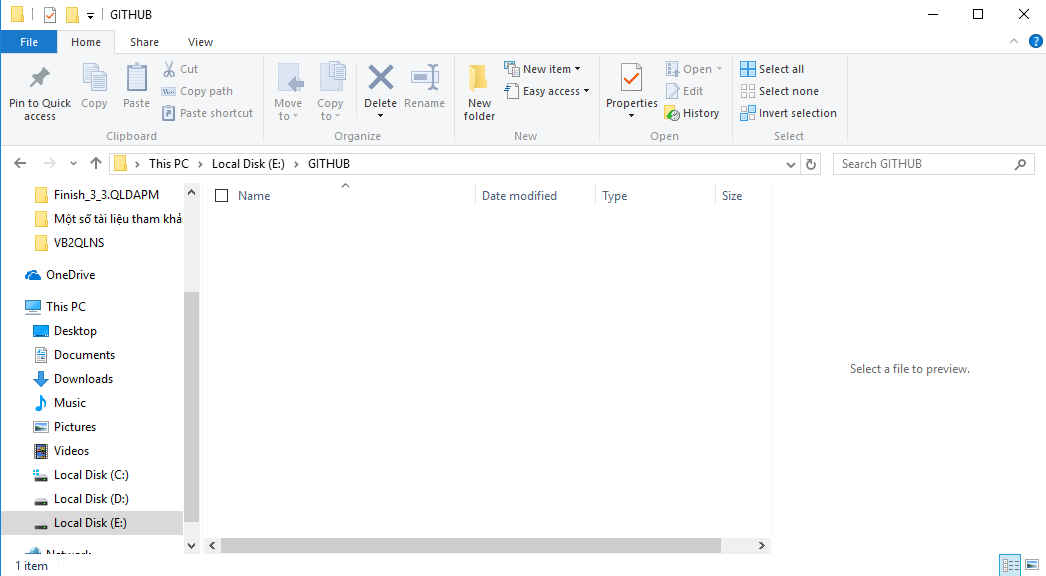
**7. Kết nối GitHub và GitHub Desktop**

**Push lên remote repository**

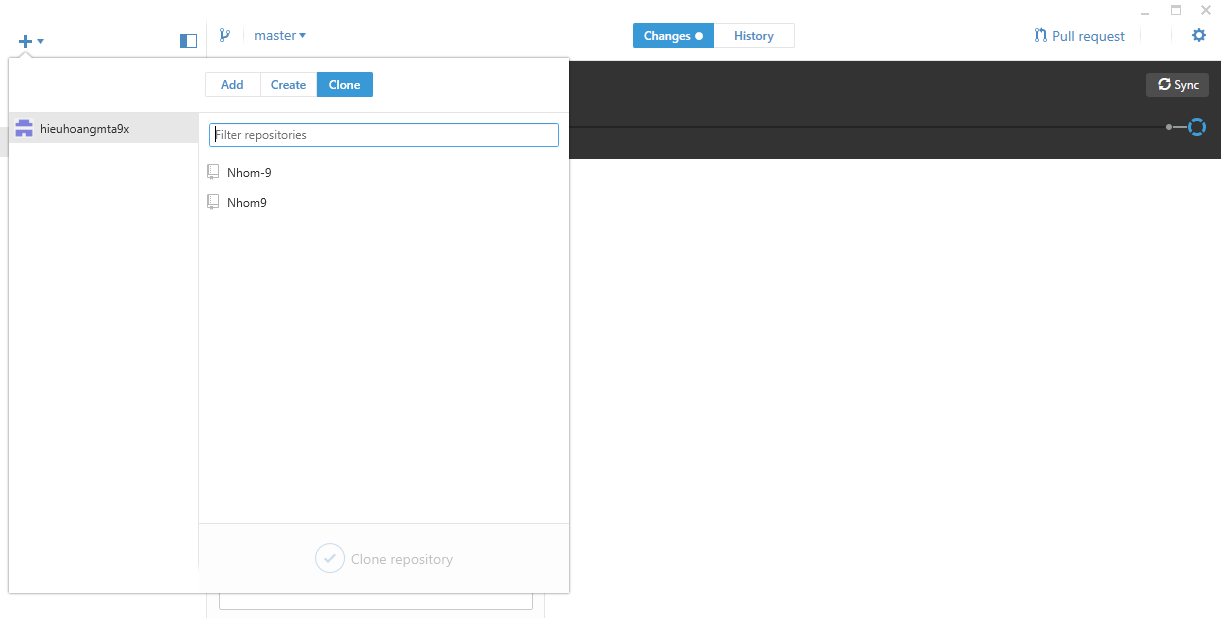
Trước hết cần chọn một thư mục rỗng để làm vị trí chứa dữ liệu địa phương.

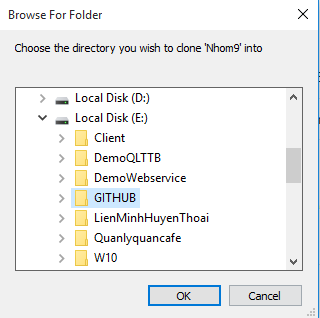
Chẳng hạn:

* E:/GITGUB

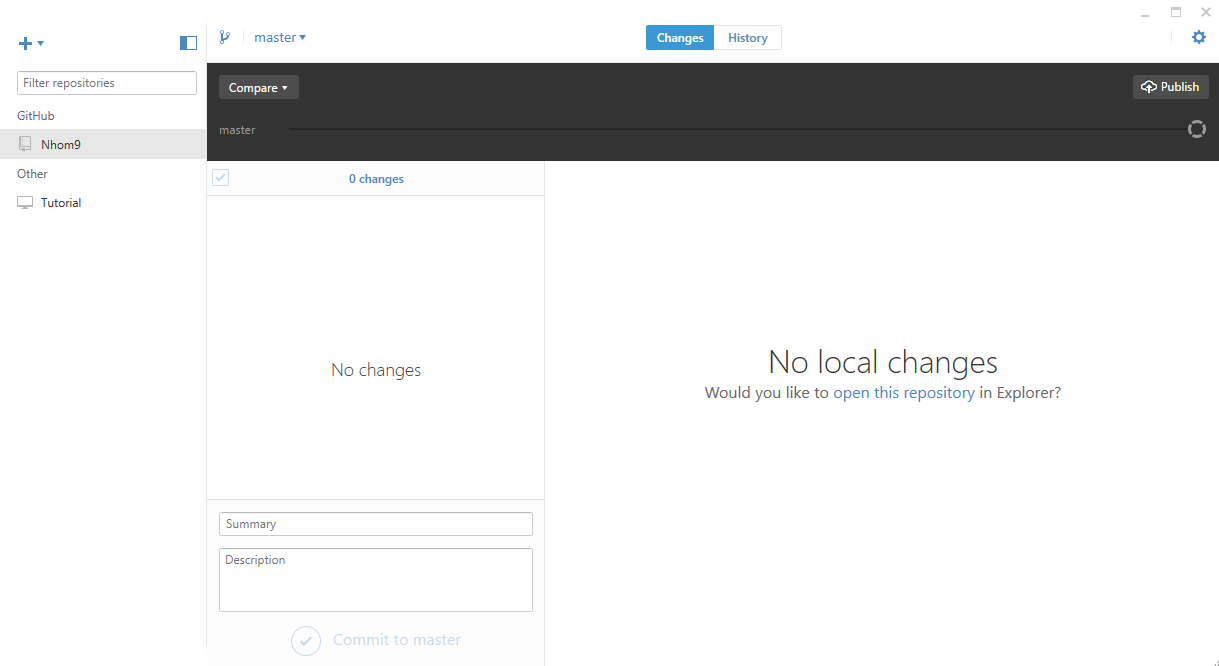


Trên GitHub Desktop, lựa chọn một Repository bạn đã tạo trên GitHub để Clone thành một bản ở máy tính địa phương của bạn.

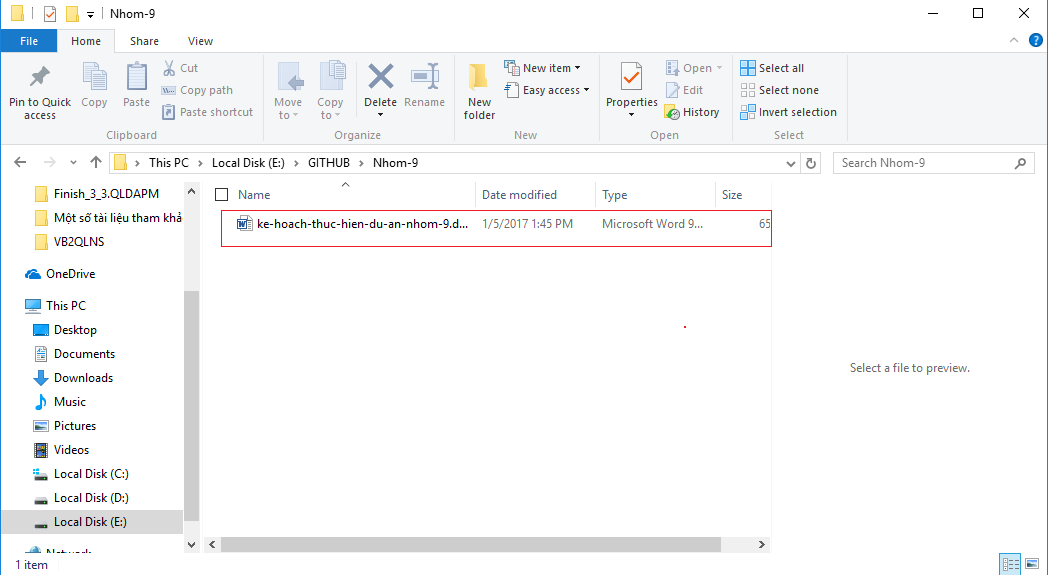




Lúc này trên GitHub Desktop bạn sẽ thấy một Local Repository đã được tạo ra.

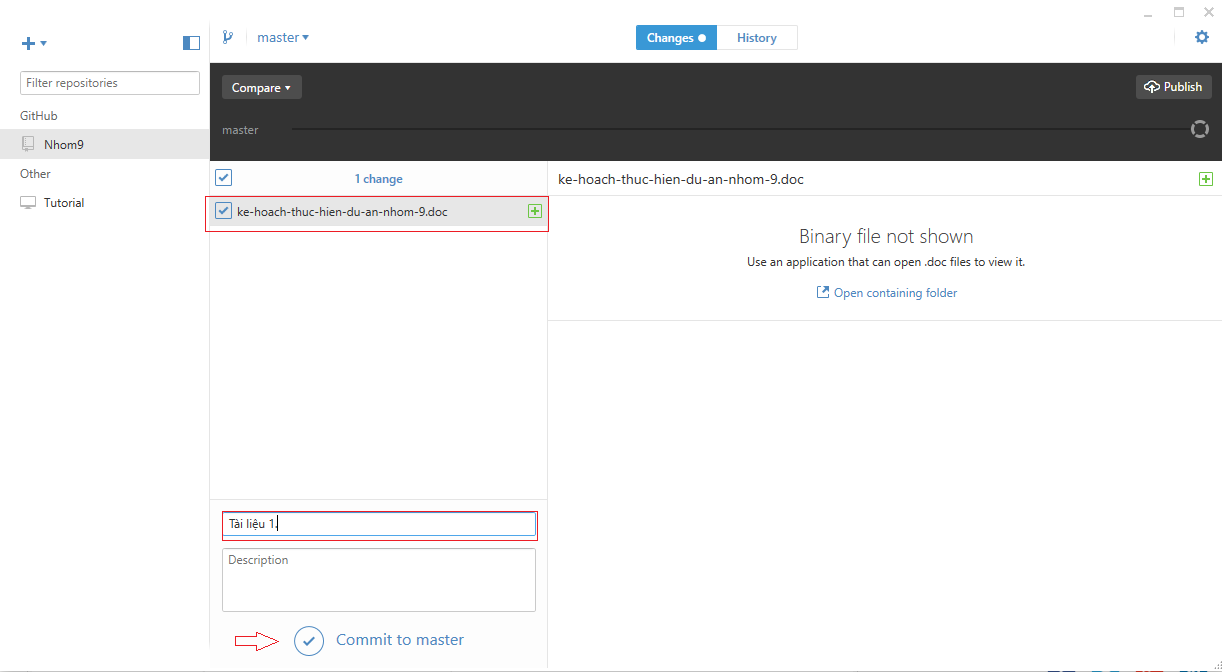


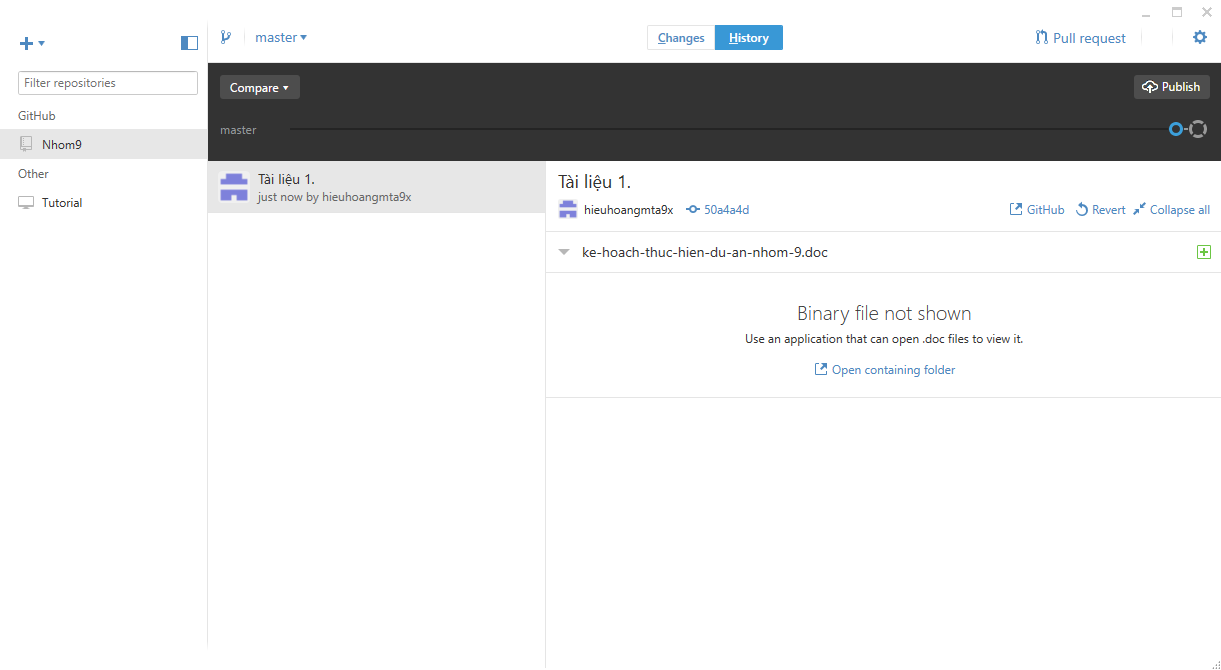
Copy một vài file dữ liệu của bạn vào Local Repository:



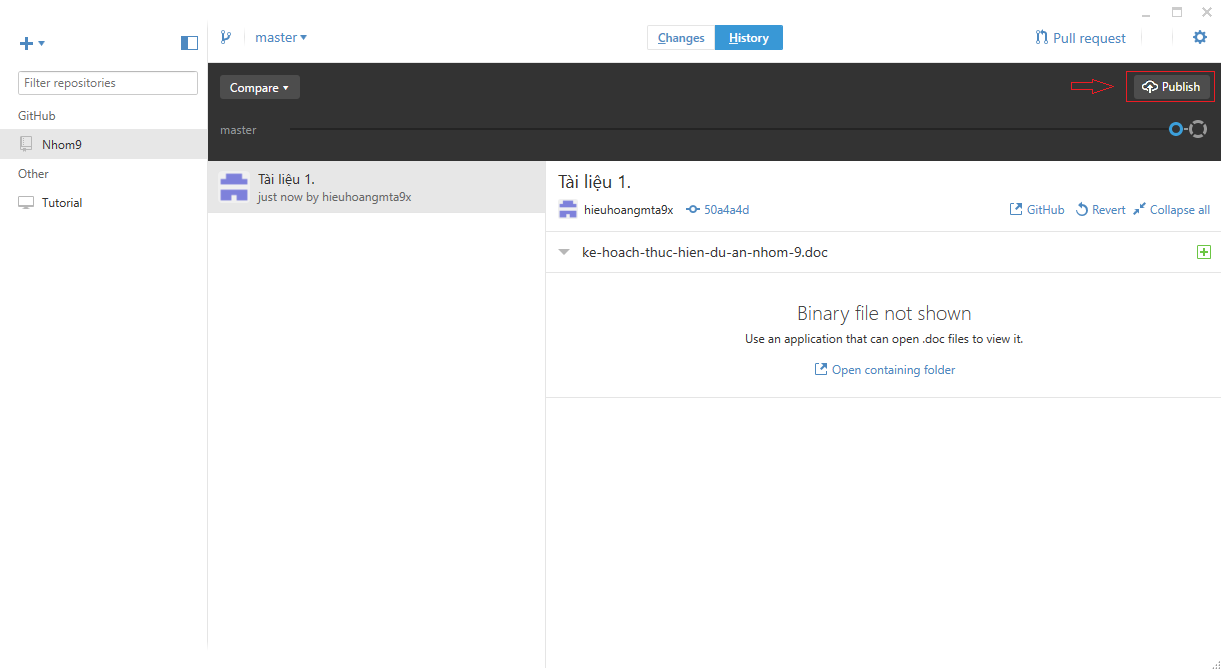
GitHub Desktop ngay lập tức nhận biết được các thay đổi tại Local Repository.

Nhập thông tin ghi chú (Comment) và nhấn Commit dữ liệu

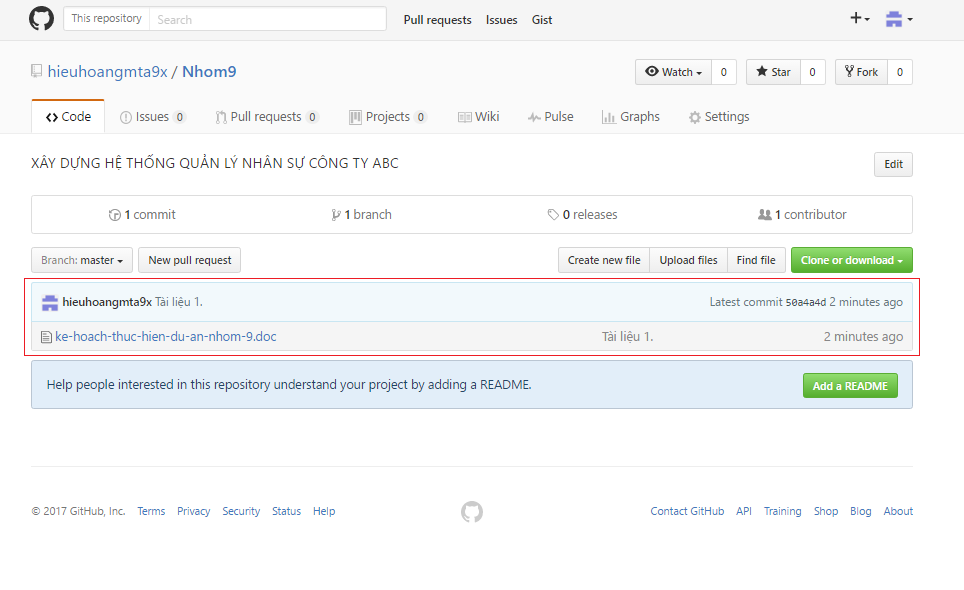




Cuối cùng là công khai cho mọi người có thể xem:



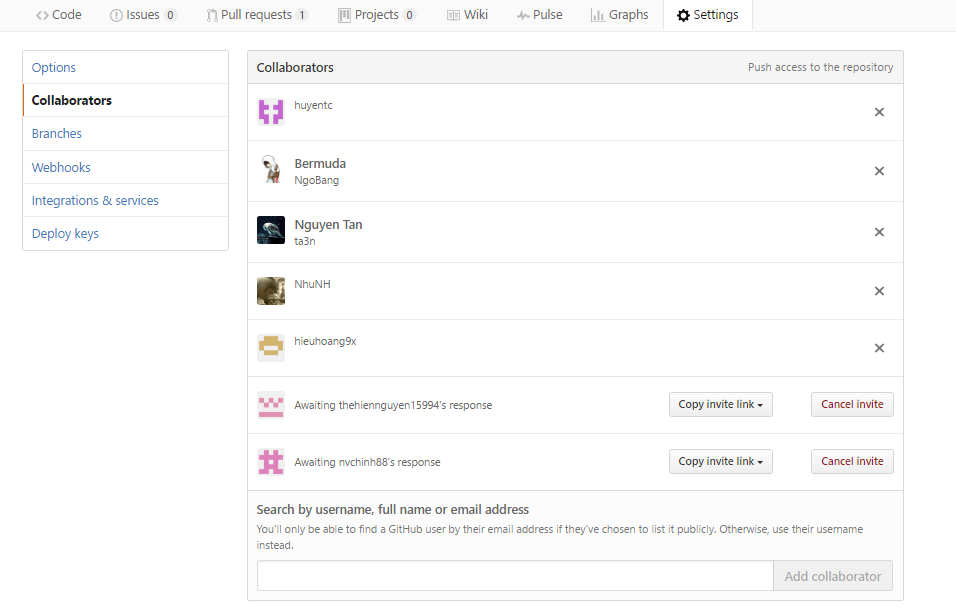
Các file dữ liệu bạn có thể nhìn thấy trên Server.



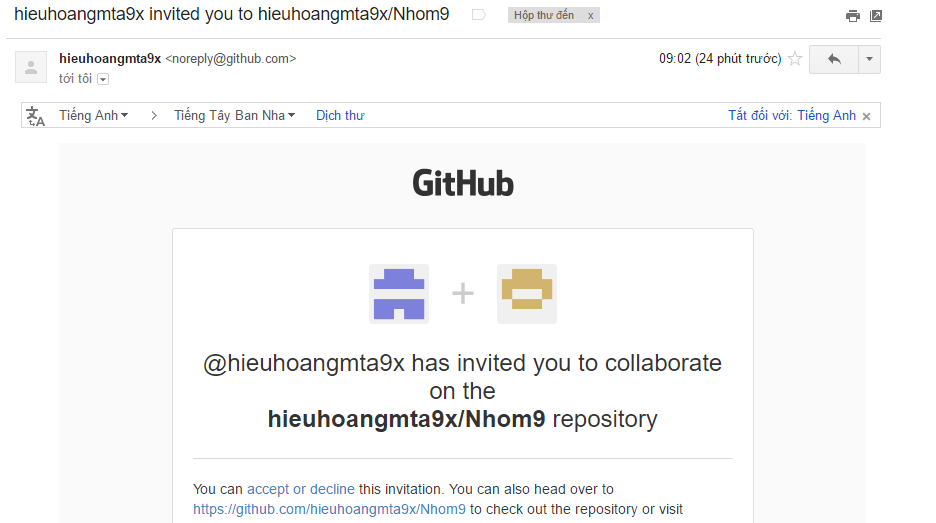
**8. Mời thành viên**

Settings 🡪 Collaborators.

Trong tab Collaborators điền tên một thành viên và add Collaborato thành viên mới tham gia dự án



Thành viên xác nhận mail tham gia dự án



**9. Branch là gì**

Branch là cái dùng để phân nhánh và ghi lại luồng của lịch sử. Branch đã phân nhánh sẽ không ảnh hưởng đến branch khác nên có thể tiến hành nhiều thay đổi đồng thời trong cùng 1 repository.

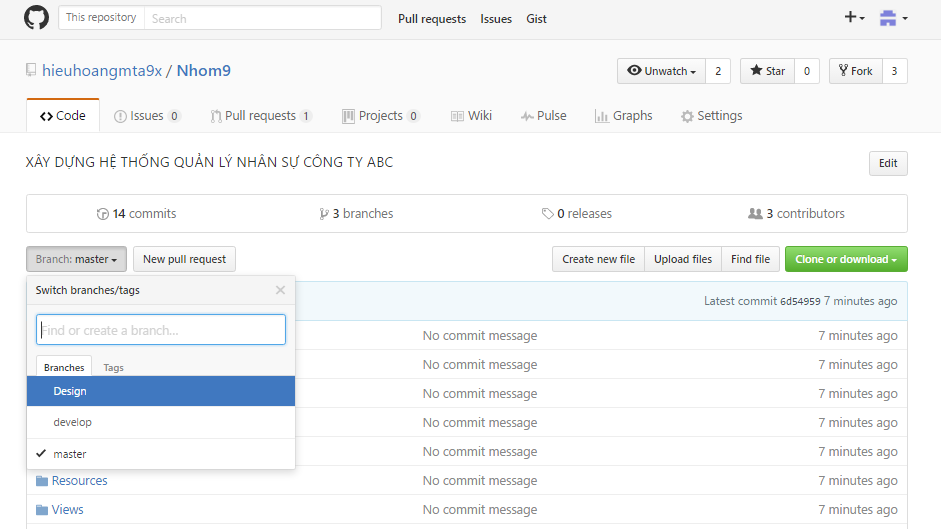
Hơn nữa, branch đã phân nhánh có thể chỉnh sửa tổng hợp lại thành 1 branch bằng việc hợp lại (merge) với branch khác.

Các thành viên của nhóm sẽ tạo branch dùng riêng cho công việc của mình từ branch chính để không ảnh hưởng đến công việc của các thành viên khác. Sau đó, những thành viên đã hoàn thành công việc của mình sẽ thực hiện đưa thay đổi của mình vào branch chính. Theo cách như vậy, sẽ không bị ảnh hưởng từ công việc của các thành viên khác, và bản thân mình có thể thực hiện công việc của mình.

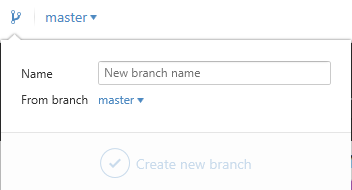
Hơn nữa, bằng việc để lại lịch sử theo đơn vị công việc, trong trường hợp có phát sinh vấn đề thì việc điều tra nguyên nhân ở những vị trí thay đổi cũng như việc tiến hành đối sách khắc phục sẽ trở nên dễ dàng hơn.

**9.1 Tạo một nhánh mới.**

Tạo trên **Remote repository**



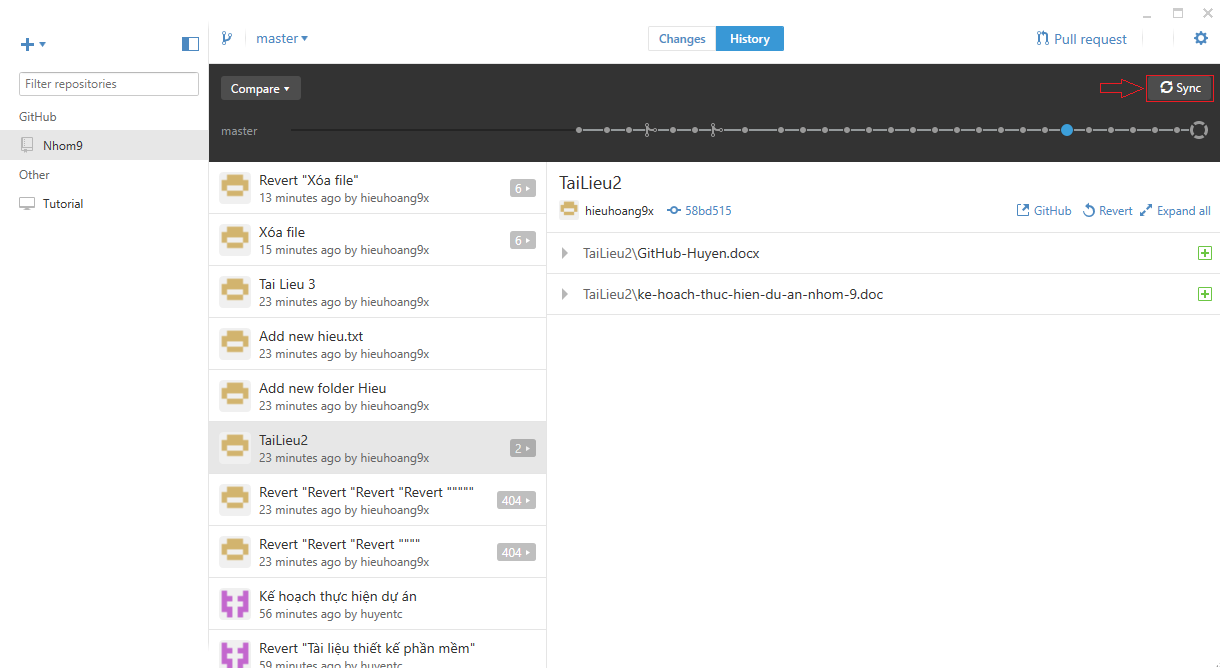
Tạo trên **local repository**



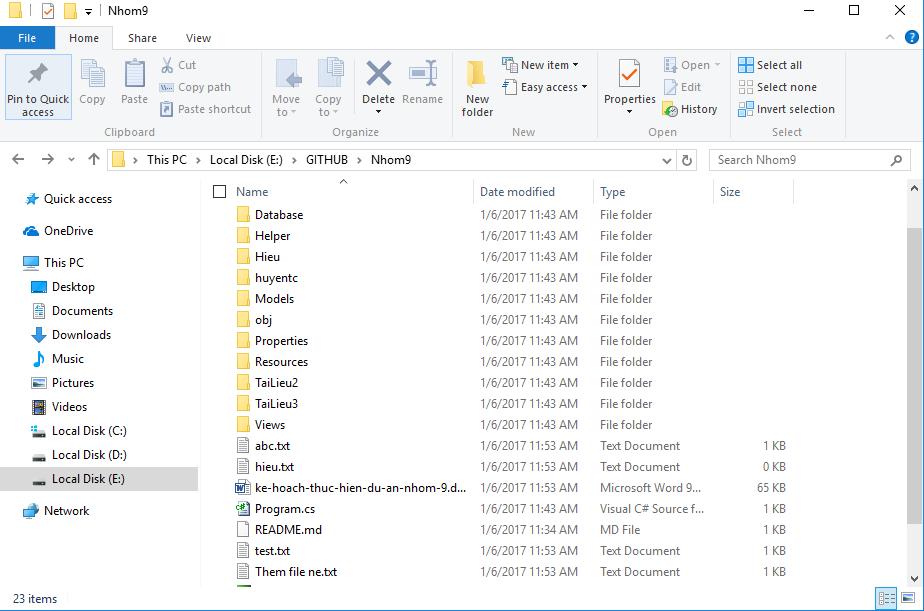
**10. Cách lấy tài liệu từ thư mục dự án trên máy chủ về máy các thành viên nhóm: Sử dụng phần mềm client hay plugin nào để lấy tài liệu trên máy chủ về**

Sử dụng GitHub desktop để lấy tài liệu trên máy chủ về

Khi một thành viên trong dự án đẩy một thư mục từ Client lên Server, các thành viên khác có thể update Client của họ để xem tài liệu của các thành viên khác bằng cách click ***Sync.***



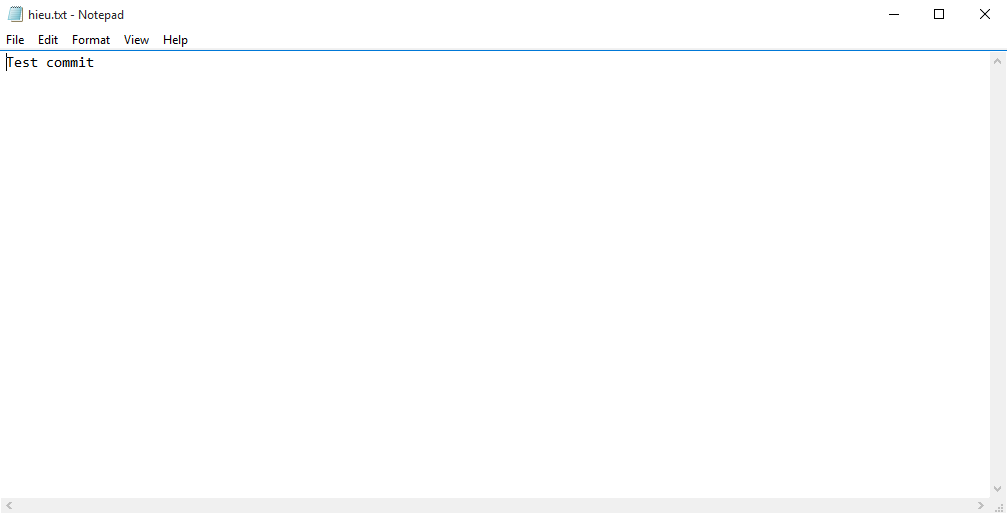
Local Repository sau khi được update.



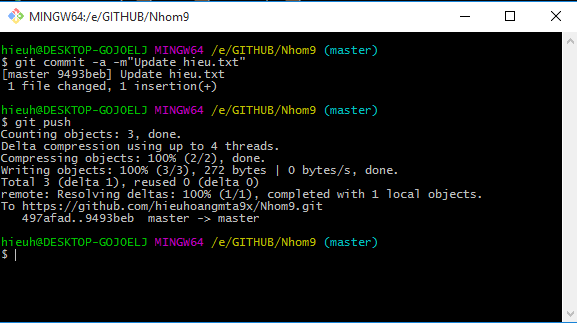
**11. Xem xét, thử nghiệm tình huống nhiều thành viên cùng thay đổi một chỗ trong một file của dự án và upload đồng thời lên server.**

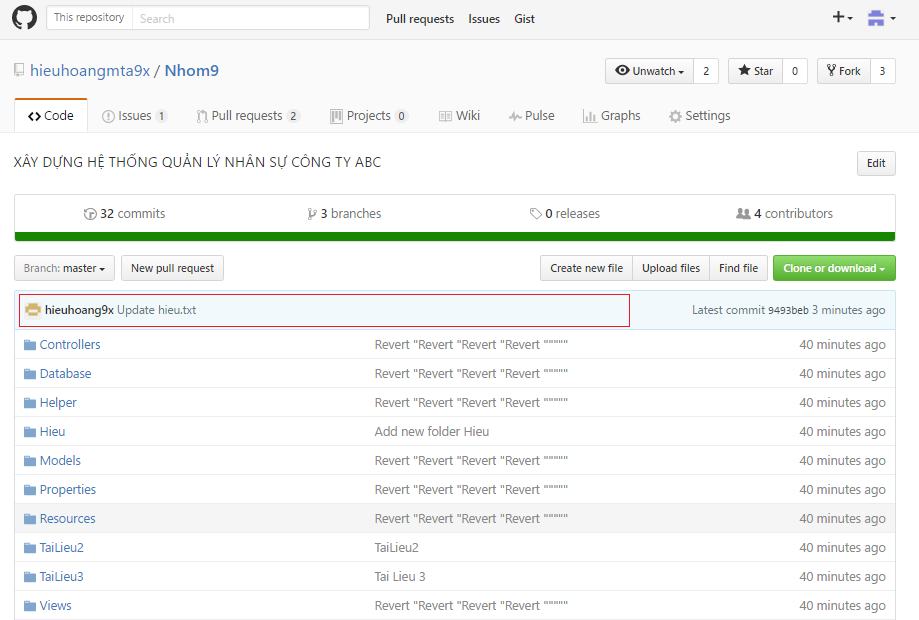
**11.1 Update Local Repository file hieu.txt**

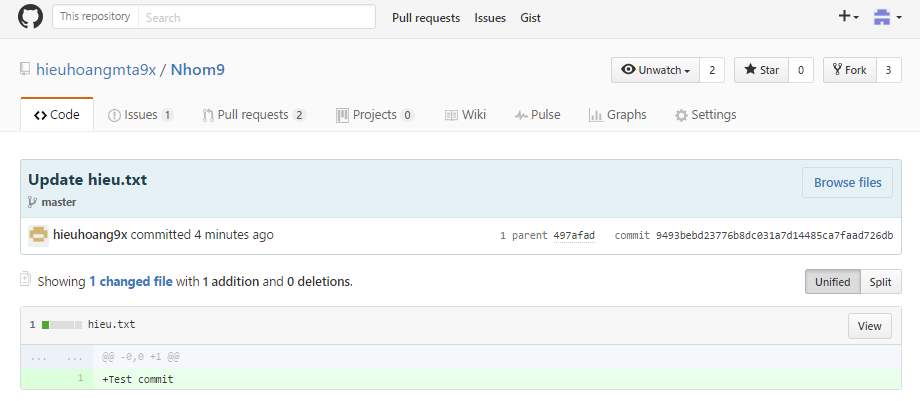
Add “Test commit” trong file hieu.txt



Comit file hieu.txt và đồng bộ lên Server.

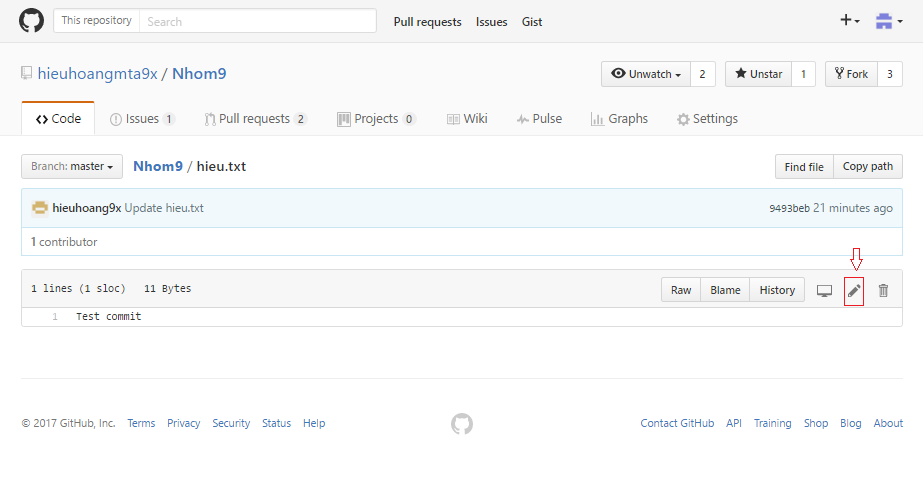




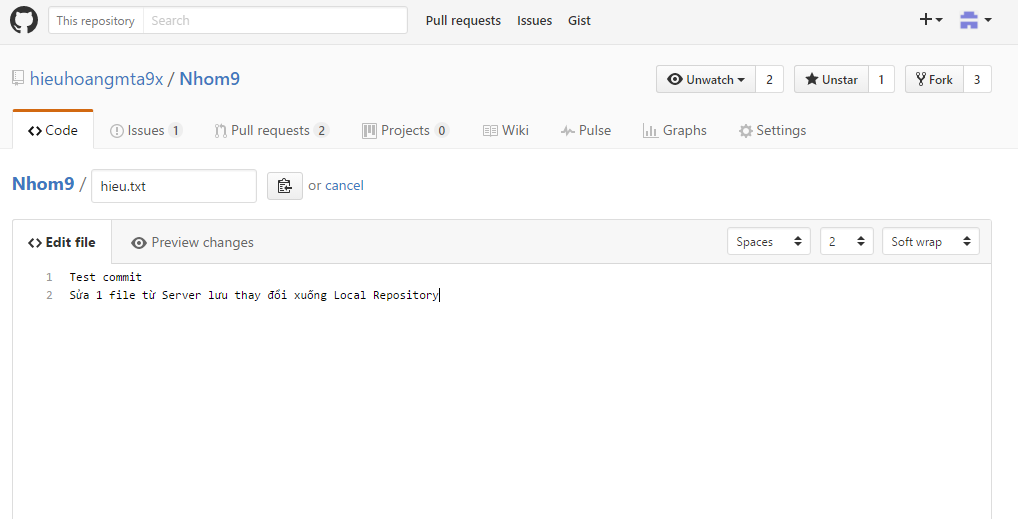


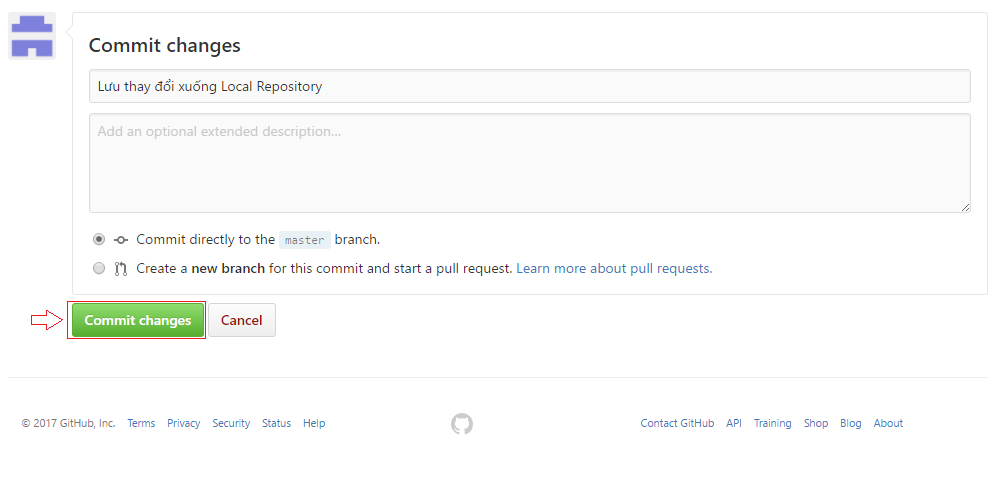
**11.2 Sửa 1 file từ Server lưu thay đổi xuống Local Repository**

Edit file hieu.txt

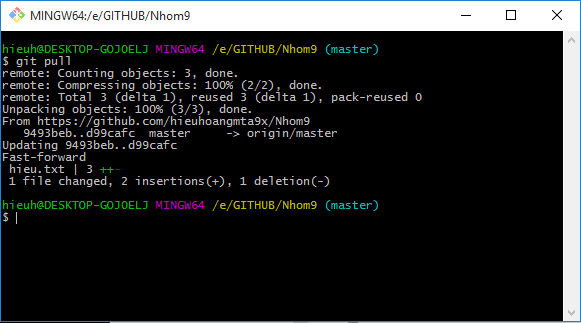


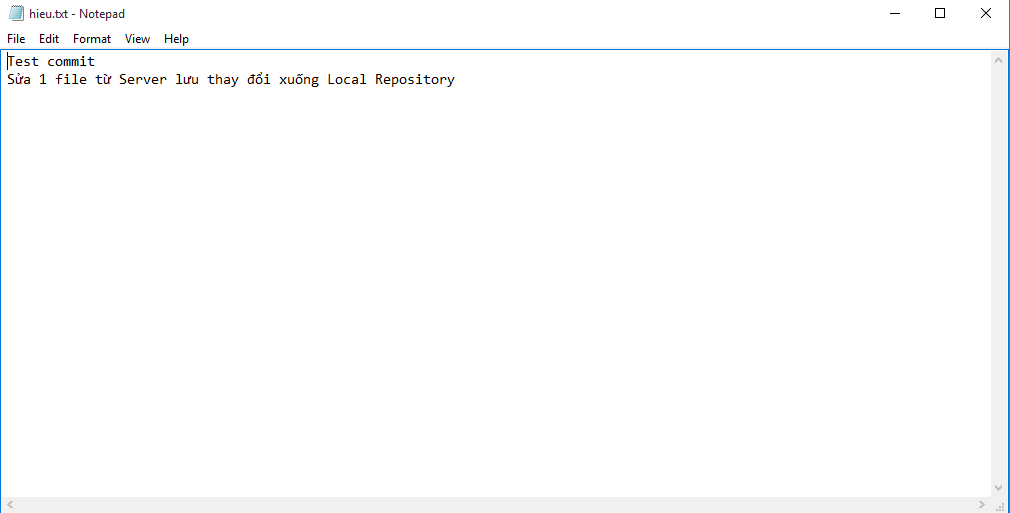
Thêm 1 line và lưu sự thay đổi



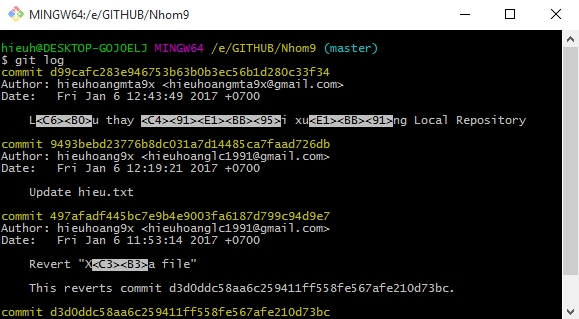


Đồng bộ từ Server xuống Local Repository

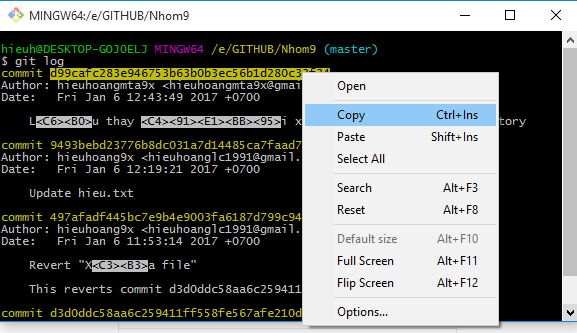




**11.3 Revert dữ liệu từ các phiên bản trước đó**



Chọn 1 phiên phản để Revert



Lệnh git Revert <Phiên bản>

Local Repository sẽ quay về phiên bản bạn muốn thay đổi

**12. Thống kê lưu lượng truy cập của từng thành viên (GitHub).**

