## [Tìm Hiểu SVN](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

**MỤC LỤC**

[Tìm Hiểu SVN 1](#_Toc504428147)

[**I.** **Khái niệm** 2](#_Toc504428148)

[Tìm Hiểu Về SVNVersion Control System (VCS) là một phần mềm mà giúp các lập trình viên phần mềm làm việc với nhau và duy trì một lịch sử đầy đủ công việc của họ. 2](#_Toc504428149)

[**II. Mục tiêu** 2](#_Toc504428150)

[- Dưới đây là mục tiêu của một Version Control System: 2](#_Toc504428151)

[◦ Cho phép các lập trình viên phần mềm làm việc tương tác với nhau; 2](#_Toc504428152)

[◦ Không viết đè lên thay đổi của mỗi thành viên tạo ra; 2](#_Toc504428153)

[◦ Duy trì lịch sử của mỗi phiên bản; 2](#_Toc504428154)

[**III. Các loại VCS** 2](#_Toc504428155)

[◦ Hệ thống quản lý phiên bản tập trung (CVCS), và 2](#_Toc504428157)

[◦ Hệ thống quản lý phiên bản phân phối/phân cấp (DVCS). 2](#_Toc504428158)

[Thuật ngữ kiểm soát phiên bản (Version control) trong SVN 3](#_Toc504428160)

[**IV. Hướng dẫn sử dụng** 3](#_Toc504428161)

[1.Repository (Kho chứa): 3](#_Toc504428163)

[2.Trunk (Thân): 3](#_Toc504428164)

[3.Tags (Thẻ): 3](#_Toc504428165)

[5.Working copy (Bản sao công việc ): 4](#_Toc504428170)

[6.Các thay đổi commit:. 4](#_Toc504428171)

[**V.Các thao tác trong SVN** 4](#_Toc504428172)

[**VI. Sử dụng github** 7](#_Toc504428173)

[Chuẩn bị 7](#_Toc504428174)

[Xong bước chuẩn bị, bây giờ là thực hành 7](#_Toc504428175)

# **Khái niệm**

## [Tìm Hiểu Về SVNVersion Control System (VCS) là một phần mềm mà giúp các lập trình viên phần mềm làm việc với nhau và duy trì một lịch sử đầy đủ công việc của họ.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

# **II. Mục tiêu**

## - [Dưới đây là mục tiêu của một Version Control System:](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [◦ Cho phép các lập trình viên phần mềm làm việc tương tác với nhau;](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [◦ Không viết đè lên thay đổi của mỗi thành viên tạo ra;](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [◦ Duy trì lịch sử của mỗi phiên bản;](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

# **III. Các loại VCS**

## [Một VCS được chia thành hai kiểu:](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [◦ Hệ thống quản lý phiên bản tập trung (CVCS), và](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [◦ Hệ thống quản lý phiên bản phân phối/phân cấp (DVCS).](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [Trong phần hướng dẫn này, chúng ta sẽ chỉ tập trung vào hệ thống phiên bản tập trung mà đặc biệt là Subversion. Subversion sử dụng một máy chủ trung tâm để giữ tất cả các file và cung cấp quyền hợp tác của team.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [Thuật ngữ kiểm soát phiên bản (Version control) trong SVN](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

# **IV. Hướng dẫn sử dụng**

## [Hãy cùng bàn về một số mục mà chúng ta sử dụng trong phần hướng dẫn này:](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## 1.[Repository (Kho chứa): Một repository là trái tim của bất cứ hệ thống quản lý phiên bản. Nó là khu vực trung tâm để các lập trình viên giữ tất cả công việc của họ. Kho lưu không chỉ lưu các file mà còn lưu giữ cả lịch sử làm việc. Kho lưu được truy cập thông qua mạng, thực hiện vai trò như một máy chủ và công cụ điều khiển phiên bản là một máy khách. Các máy khách có thể kết nối với repository và sau đó họ có thể giữ/lấy lại những thay đổi họ thực hiện tới/từ repository. Bằng cách giữ các thay đổi , một khách tạo các thay đổi này tới người khác và bằng cách lấy lại những thay đổi, một khách có thể nhận được các thay đổi được thực hiện bởi người khác như là một bản sao công việc.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## 2.[Trunk (Thân): Một thân là một thư mục tại đó tất cả các phát triển chính xảy ra và thường được kiểm tra bởi các lập trình viên làm việc trên dự án.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## 3.[Tags (Thẻ): Thư mục các thẻ được sử dụng để giữ các ảnh chụp nhanh (snapshot) được đặt tên của dự án. Hoạt động thẻ cho phép đặt các tên để mô tả và dễ ghi nhớ cho các phiên bản cụ thể trong repository.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [Ví dụ, LAST\_STABLE\_CODE\_BEFORE\_EMAIL\_SUPPORT thì dễ ghi nhớ hơn](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [7ceef8cb-3799-40dd-a067-c216ec2e5247 của repository UUID và](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [Revision: 13](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## 4.[Branches (Nhánh): Hoạt động nhánh được sử dụng để tạo một tiến trình khác của sự phát triển. Nó hữu ích khi bạn muốn tiến trình phát triển của bạn phân theo hai hướng khác nhau. Ví dụ, khi bạn công bố phiên bản 5.0, bạn có thể muốn tạo ra một nhánh để phát triển các tính năng phiên bản 6.0 mà được giữ riêng rẽ với phiên bản 5.0.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## 5.[Working copy (Bản sao công việc ): Bản sao công việc là hình chụp nhanh (snapshot) của repository. Kho lưu được chia sẻ bởi cả team, nhưng người ta không thể sửa nó một cách trực tiếp. Thay vào đó mỗi nhà lập trình kiểm tra bản sao công việc. Bản sao công việc là khu vực làm việc riêng, tại đó các nhà lập trình có thể thực hiện công việc của họ riêng rẽ một mình, cách biệt với phần còn lại của team.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## 6.[Các thay đổi commit: Commit là tiến trình của việc giữ các thay đổi từ khu vực làm việc riêng tới máy chủ trung tâm. Sau khi commit, các thay đổi sẵn sàng được sử dụng bởi các thành viên khác trong team. Các thành viên khác có thể lấy lại các thay đổi này bằng cách cập nhật bản sao công việc. Commit là một hoạt động nguyên tử (atomic operation) tức là không thể chia nhỏ hoạt động đó được. Hoặc là nó thành công hoặc là thất bại. Bạn không bao giờ nhìn thấy một commit hoàn thành giữa chừng.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

# **V.Các thao tác trong SVN**

Tạo repository trong SVN:

Kho lưu là địa điểm trung tâm mà các lập trình viên giữ tất cả những gì liên quan đến công việc của họ. Kho lưu khong chỉ giữ file mà còn lưu giữ cả lịch sử về các thay đổi. Điều này có nghĩa là nó duy trì một lịch sử của các sự thay đổi được thực hiện trong các file.

Tiến trình create được sử dụng để tạo một repository mới. Hầu hết thời gian tiến trình này được thực hiện chỉ một lần. Khi bạn tạo một repository mới, VCS của bạn sẽ hi vọng bạn nói gì đó để nhận diện nó, như là nơi bạn muốn được tạo hoặc tên mà bạn muốn đặt cho repository.

Tiến trình Checkout trong SVN

Tiến trình checkout được sử dụng để tạo ra một bản sao làm việc từ repository. Bản sao làm việc là khu vực làm việc riêng nơi mà các nhà lập trình thực hiện những thay đổi của họ, và sau đó, đệ trình những thay đổi này tới repository.

Tiến trình Update trong SVN

Như tên gọi đã đề cập, tiến trình update được sử dụng để cập nhật bản sao làm việc. Tiến trình này đồng bộ khu vực làm việc với repository. Khi repository được chia sẻ thì tất cả các thành viên trong team có thể commit những thay đổi của họ và bản sao làm việc trở lên cũ đi.

Bây giờ chúng ta giả sử rằng Tom và Jerry là hai nhà lập trình làm việc trên một dự án. Cả hai kiểm tra phiên bản mới nhất từ repository và bắt đầu làm việc. Tại thời điểm này, bản sao công việc là đồng bộ hoàn toàn với repository. Jerry hoàn thành công việc của anh ta rất hiệu quả và commit những thay đổi anh ta thực hiện tới repository.

Bây giờ khu vực làm việc của Tom chưa được cập nhật. Tiến trình update sẽ kéo những thay đổi mới nhất của Jerry từ repository và sẽ cập nhật khu vực làm việc của Tom.

Thực hiện các thay đổi trong SVN

Sau khi checkout, ai đó có thể thực hiện các tiến trình đa dạng để thực hiện các thay đổi. Chỉnh sửa là tiến trình thông thường nhất. Người nào đó có thể chỉnh sửa: thêm/gỡ bỏ nội dung file đang hiện hữu.

Một ai đó có thể thêm các file/thư mục. Nhưng ngay lập tức những file/thư mục này không không trở thành một phần của repository, trừ khi chúng đã được thêm trong quá trình diễn ra liệt kê thay đổi và trở thành một phần của repository sau tiến trình commit.

Tương tự, một ai đó có thể xóa file/thư mục. Tiến trình delete này ngay lập tức xóa file từ trong khu vực làm việc, nhưng sự xóa bỏ thực sự của file là được thêm vào trong quá trình diễn ra liệt kê thay đổi (pending change-list)và những thay đổi được hoàn thành tới repository sau tiến trình commit.

Tiến trình rename thay đổi tên của file/thư mục. Tiến trình move được sử dụng để di chuyển file/thư mục từ một địa điểm tới địa điểm khác trong cây repository.

Duyệt các thay đổi (review) trong SVN

Khi bạn kiểm tra hoặc cập nhật khu vực làm việc, thì khi đó khu vực làm việc của bạn được hoàn toàn đồng bộ với repository. Nhưng khi bạn thực hiện các thay đổi tới khu vực làm việc, nó trở nên mới hơn repository. Và nó là thói quen tốt để Review - duyệt lại những thay đổi của bạn trước khi thực hiện tiến trình commit.

Tiến trình status trong SVN liệt kê các chỉnh sửa mà đã thực hiện tới khu vực làm việc. Như chúng ta đã đề cập trước đó, bất cứ khi nào bạn gây ra thay đổi trong khu vực làm việc, tất cả những thay đổi đó trở thành một phần của quá trình diễn ra liệt kê thay đổi. Và tiến trình Status được sử dụng để quan sát quá trình diễn ra liệt kê thay đổi này.

Tiến trình status chỉ cung cấp một danh sách các thay đổi nhưng không đưa ra những chi tiết về chúng. Một ai đó có thể sử dụng tiến trình diff để quan sát các chi tiết này của sự chỉnh sửa mà đã được thực hiện tới khu vực làm việc.

Sửa các lỗi trong SVN

Chúng tôi giả sử rằng một ai đó đã gây ra các thay đổi tới khu vực làm việc của anh ta, nhưng bây giờ anh ta muốn dỡ bỏ các thay đổi này. Trong tình huống này, tiến trình revert sẽ giúp làm điều đó.

Tiến trình revert trả lại các chỉnh sửa mà đã thực hiện tới khu vực làm việc. Nó là có thể để trả lại một hoặc nhiều file/thư mục. Nó cũng có thể trả lại toàn bộ khu vực làm việc. Trong trường hợp này, tiến trình revert sẽ phá hủy quá trình diễn ra liệt kê thay đổi và sẽ mang khu vực làm việc trở lại trạng thái ban đầu.

Giải quyết các xung đột trong SVN:

Các xung đột có thể xảy ra tại thời gian của việc sáp nhập. Tiến trình merge tự động thực hiện mọi thứ mà có thể thực hiện theo một cách an toàn. Mọi thứ khác được cân nhắc là sự xung đột. Ví dụ, tệp hello.c được chỉnh sửa trong một nhánh và bị xóa trong nhánh khác. Trong tình huống như vậy cần một người đưa ra quyết định. Tiến trình resolve được sử dụng để giúp người sử dụng tính toán các thứ và để thông báo cho VCS về cách giải quyết các xung đột.

Ký thác các thay đổi (commit) trong SVN

Tiến trình commit được sử dụng để áp dụng các thay đổi từ khu vực làm việc tới repository. Tiến trình này chỉnh sửa repository và các nhà lập trình khác có thể quan sát những thay đổi này bằng cách cập nhật khu vực làm việc.

Trước khi commit, một ai đó phải thêm file/thư mục tới quá trình diễn ra liệt kê thay đổi. Đây là nơi mà những thay đổi chờ để được commit. Với commit, chúng ta thường cung cấp một thông báo log để giải thích tại sao ai đó gây ra các thay đổi. Thông báo log này trở thành một phần của lịch sử của repository. Ký thác là một tiến trình nguyên tử, mà có nghĩa là không thể chia tiến trình này được. Hoặc là nó được thực hiện hoặc không. Bạn không thể thấy một tiến trình commit bỏ dở giữa chừng.

# [**VI. Sử dụng github**](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [Chuẩn bị](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Đầu tiên đăng kí tài khoản Github https://github.com/347](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[đăng kí 2 cái, để mò mẫm các chức năng của Github cho dễ](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Sau đó là cài đặt Github for Windows(GFW) https://windows.github.com/607](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

## [Xong bước chuẩn bị, bây giờ là thực hành](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

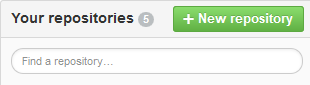
[Github có 2 thể loại thế này:](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

1. [Tạo một kho lưu code (sau này gọi là repo) cho một project của mình,.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)
2. [Tham gia vào một project nào đó.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

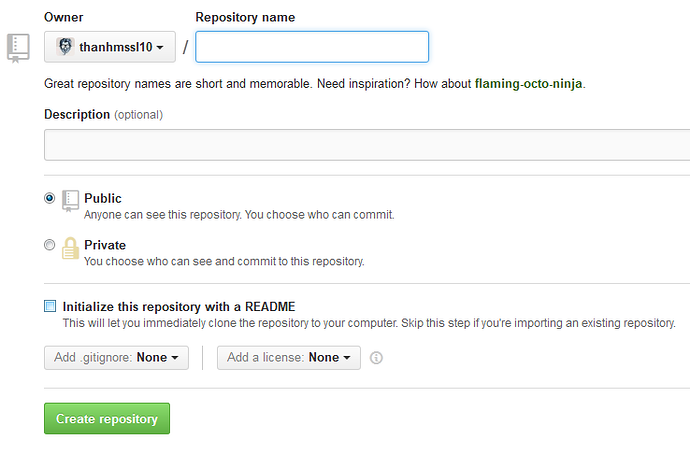
[Bây giờ ta sẽ cho tài khoản A tạo 1 cái repo chứa 1 file test.txt chẳng hạn, tài khoản B sẽ tham gia vào project của tài khoản A.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Tạo repo cho A](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Trên giao diện web của A ấn nút New Repository](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

 [New Repository](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Điền tên repo với description là được, mấy cái còn lại thì cứ kệ nó nhá (kệ vì nó không ảnh hưởng lắm đến bài viết này, chứ không phải không quan trọng đâu nhá](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)[)](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

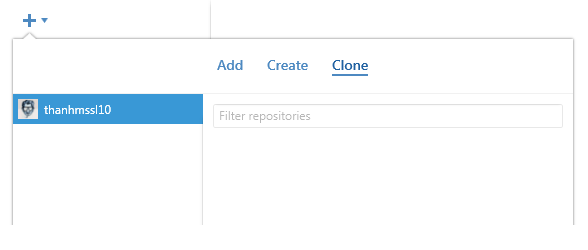


[Creat Repository là xong](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

**newrepo2.png738x486 25 KB**

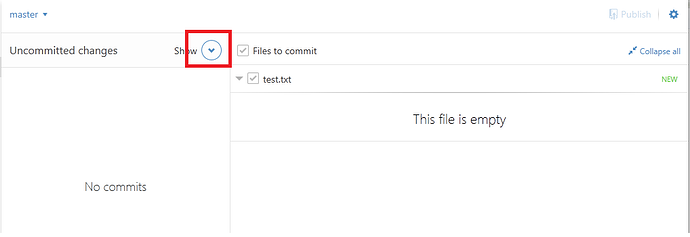
[Clone về máy](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Mở Github for Windows (GFW) để clone repo về máy](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

 [Ấn vào dấu cộng góc trên cùng bên trái, chọn Clone, rồi nó sẽ hiện ra repo mình vừa tạo ở trên, ấn Clone Repository](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Thực hiện commit đầu tiên nào!!!](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Clone xong thì trên máy có một folder mang tên repo của bạn, bạn cho test.txt vào đó, khi đó quay lại GFW ta sẽ thấy "cảnh tượng" như này](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)



[Ấn Show rồi commit](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

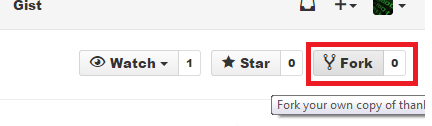
**commitfirst.png1018x350 14.3 KB**

[Vậy là thay đổi của bạn giờ được lưu vào lịch sử thay đổi trên máy của bạn, sau đó ta sẽ cho lên repo trên github](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

 [Ấn Publish để đưa lên repo trên github,chữ publish sau lần đầu tiên sẽ đổi thành sync](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

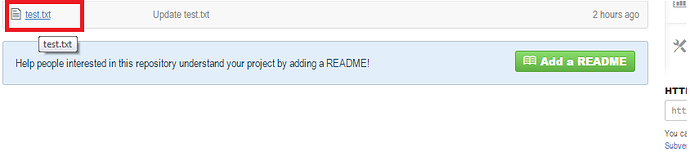
[Fork về tài khoản B](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Bây giờ đến lượt ông B, ông B sẽ "chôm" repo của ông A về tài khoản của mình (hành động này gọi là Fork)  
Từ giao diện web của B tìm đến repo của A rồi ấn nút Fork](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

 [Bây giờ B đã có 1 repo chứa dữ liệu giống hệt repo của A](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

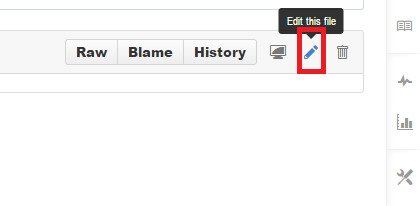
[Chỉnh sửa test.txt và Pull request (sau này gọi là PR) từ B tới A](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

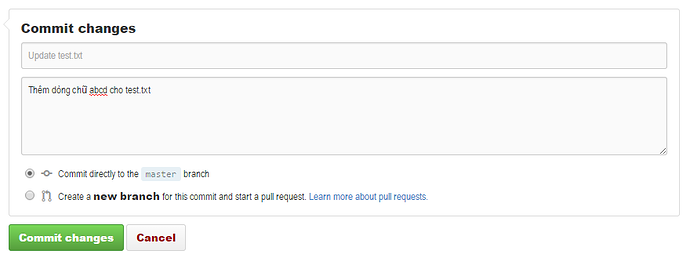
[Bây giờ B sẽ chỉnh sửa file test.txt, (chỉnh sửa trực tiếp trên giao diện web của B), khi đó file test.txt giữa A và B đã có sự khác nhau và ta có thể gửi PR từ B tới A , PR nghĩa là B nói với A:"B đã sửa file test.txt, yêu cầu A hợp nhất vào repo của A", nếu như A đồng ý với yêu cầu đó,thì file test.txt trên repo của A sẽ thay đổi theo file trên repo của B](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)



[Bắt đầu chỉnh sửa file test.txt trên repo B](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

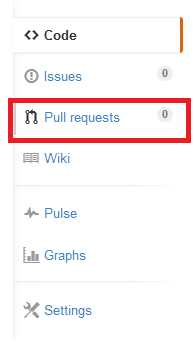
**edit1.png844x192 12.3 KB**

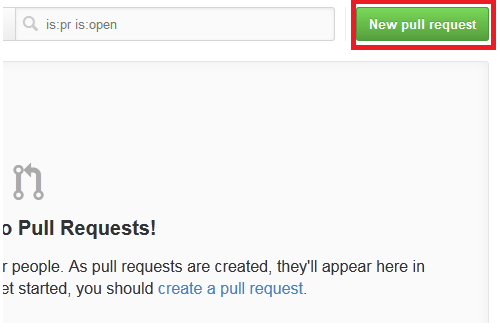


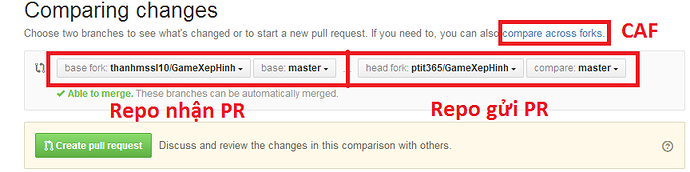


**edittest.png880x338 17.3 KB**

[Và đây là quá trình gửi PR](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)





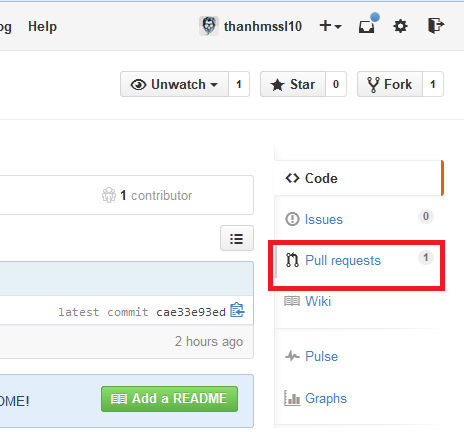


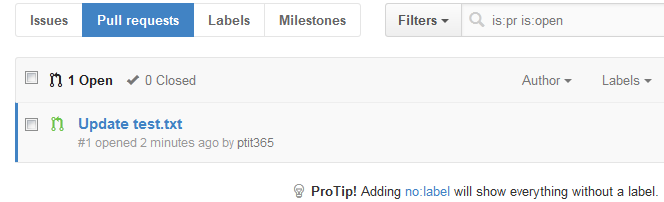
[Thông thường thì github tự động hiện bên nhận PR là repo của A và bên gửi PR là repo của B, nhưng nếu không hiện đúng như vậy thì bạn phải tự chỉnh bằng tay, trong lúc đổi bên nhận gửi PR, thì sẽ có thời điểm là bên nhận và bên gửi là cùng 1 repo, khi đó phải dùng nút CAF để giải quyết](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

**pr3.png864x216 21.3 KB**

[Sang bên A xem có gì xảy ra nào](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

[Sau khi B gửi PR đến A, thì bên A sẽ hiện thông báo có PR, bây giờ ta sẽ chấp nhận PR đó, nội dung của file test.txt trên repo A sẽ bị thay đổi theo file test.txt của bên B](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)



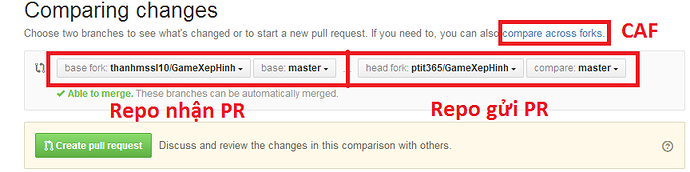




[Và bây giờ những thay đổi của B đối với file test.txt đã được chấp thuận và được hợp nhất vào file test.txt của A](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

**merge3.png774x186 17.4 KB**

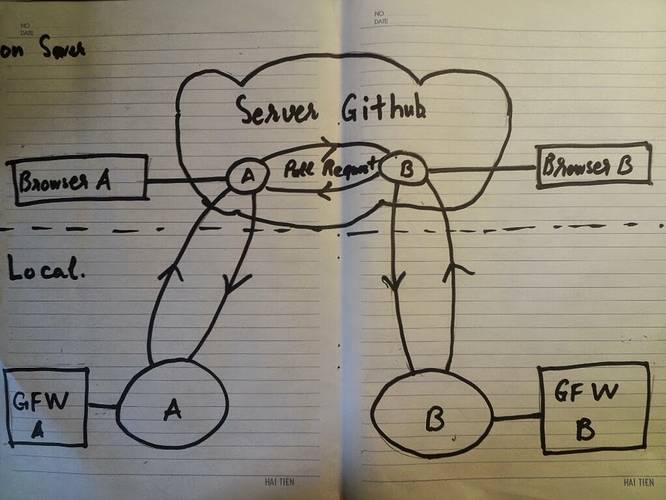
[Update cho B  
Một ngày đẹp trời A sửa file test.txt trên repo của A, nếu B muốn file test.txt của mình cũng được thay đổi giống như A thì sao nhỉ? Xoá repo đi fork lại? gọi điện cho A bảo: "Alo ông gửi pull request cho tôi đi để tôi hợp nhất vào repo của tôi"? Không cần rắc rối như vậy, github cho phép B "tự sướng" tạo một PR từ A đến B rồi tự đồng ý để merge (hợp nhất) vào repo của mình.](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)



[Nghĩa trên giao diện web của B, chỉ cần tạo PR : bên nhận là repo B còn bên gửi là repo A](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)

**pr3.png864x216 21.3 KB**

[Đây là sơ đồ khái quát quá trình trên, cái GFW của B ta không dùng đến vì ta sửa file trực tiếp trên Browser B ,trong thực tế rất ít khi làm như vậy, hầu hết là chỉnh sửa trên local trước rồi mới cho lên (vẽ tay cho nhanh, khi nào phải vẽ lại trên máy tính cho nó chuyên nghiệp mới được](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)[)](https://daynhauhoc.s3-ap-southeast-1.amazonaws.com/original/2X/c/c64a6a85b35aae3b196eaff79eecf0982716a2a7.jpg)



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_