# Github là gì? Hướng dẫn sử dụng Github

Khi bạn tham gia một dự án nào đấy thì việc phải làm việc với nhiều người là điều thường thấy. Kể cả làm việc một mình thì việc quản lý source code là vô cùng quan trọng.

Theo như mình biết thì có 2 mô hình quản lý source code: Quản lý tập trung (SVN), Quản lý phân tán (git). Mỗi hình thức đều có ưu và nhược điểm riêng.

Bài viết này, mình sẽ hướng dẫn sử dụng Github để quản lý source code cho các bạn, một dịch vụ git rất nổi tiếng của Microsoft.

**Nội dung chính của bài viết**

* [#Github là gì?](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "github-la-gi)
* [#Tính năng API Github](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "tinh-nang-api-github)
* [#Tại sao nên lại là Github? Hướng dẫn sử dụng Github](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "tai-sao-nen-lai-la-github-huong-dan-su-dung-github)
  + [#Cách tạo một GitHub Repository](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "cach-tao-mot-github-repository)
  + [#Tạo branch trên Github](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "tao-branch-tren-github)
* [#Hướng dẫn sử dụng Github chi tiết](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "huong-dan-su-dung-github-chi-tiet)
  + [1. Commit Command](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "1-commit-command)
  + [2. Pull Command](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "2-pull-command)
  + [3. Merge command](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "3-merge-command)
* [#Cloning dự án từ Github](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "cloning-du-an-tu-github)
  + [#Tạm kết](https://vntalking.com/github-la-gi-cach-su-dung-github.html" \l "tam-ket)

## #Github là gì?

[GitHub](https://github.com/) là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn [Git](https://vi.wikipedia.org/wiki/Git_(phần_mềm)) 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.

Vài thông tin về GIT:

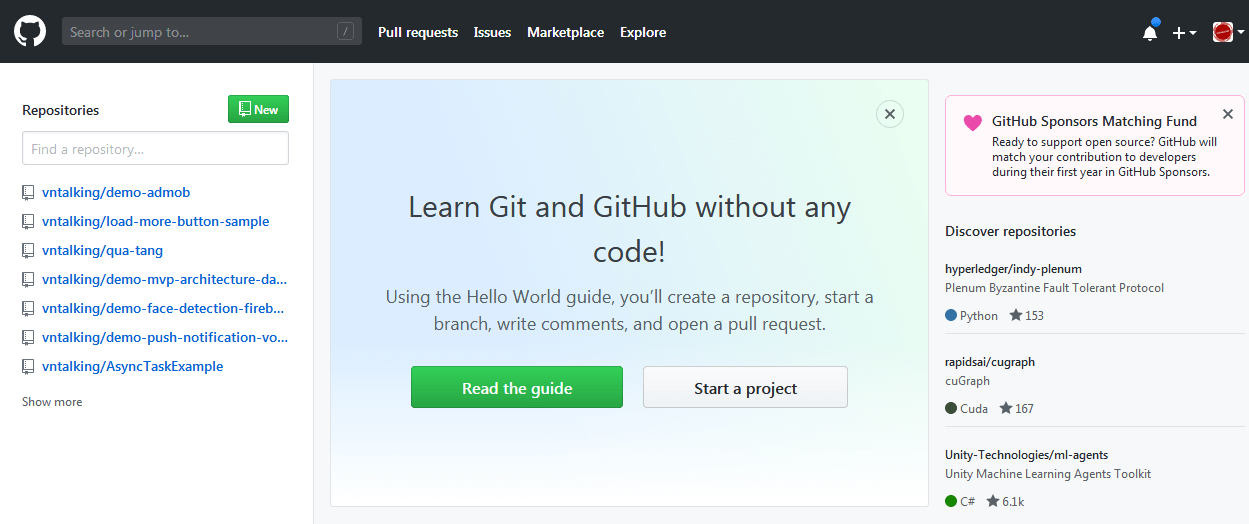
* Là công cụ giúp quản lý source code tổ chức theo dạng dữ liệu phân tán.
* Giúp đồng bộ source code của team lên 1 server.
* Hỗ trợ các thao tác kiểm tra source code trong quá trình làm việc (diff, check modifications, show history, merge source, …)

GitHub có 2 phiên bản: miễn phí và trả phí. Với phiên bản có phí thường được các doanh nghiệp sử dụng để tăng khả năng quản lý team cũng như phân quyền bảo mật dự án.

Còn lại thì phần lớn chúng ta đều sử dụng Github với tài khoản miễn phí để lưu trữ source code.

Github cung cấp các tính năng social networking như feeds, followers, và network graph để các developer học hỏi kinh nghiệm của nhau thông qua lịch sử commit.

Nếu một comment để mô tả và giải thích một đoạn code. Thì với Github, commit message chính là phần mô tả hành động mà bạn thực hiện trên source code.



Github trở thành một yếu tố có sức ảnh hưởng lớn trong cộng động nguồn mở. Cùng với Linkedin, Github được coi là một sự thay thế cho CV của bạn. Các nhà tuyển dụng cũng rất hay tham khảo Github profile để hiểu về năng lực coding của ứng viên.

Giờ đây, kỹ năng sử dụng git và Github từ chỗ ưu thích sang bắt buộc phải có đối với các ứng viên đi xin việc.

**Thật thiếu sót nếu đọc về Github mà chưa biết gì về Git. Kính mời bạn đọc bài này ngay:**[**Git là gì? Cùng tìm hiểu cách quản lý source code với Git**](https://vntalking.com/cach-quan-ly-source-code-git.html)

## **#Tính năng API Github**

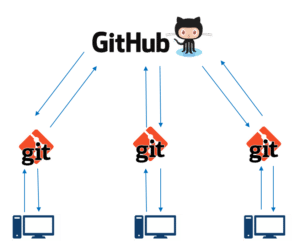
Github ngoài những tính năng tuyệt đỉnh, hỗ trợ bạn làm viết với Git cực kỳ mượt. Github còn hỗ trợ nhiều API với những tính năng quan trọng:

* **API to Update The Repository via HTTP**: Đây là tính đắt khá đắt giá, giúp bạn có thể chỉnh sửa mã nguồn từ trình duyệt thông qua HTTP POST
* **API to Access Compare Views**: API này cho phép bạn review và diff source của dự án thông quan việc xem commit, comments, các dòng khác nhau giữa 2 version của file code…
* **API to Manage Service Hooks**: API này hỗ trợ bạn đăng ký một URL cho các repository. Bất cứ khi nào có người push thay đổi lên repository, GitHub sẽ thông báo cho bạn thông qua URL mà bạn đã đăng ký trước đó.

Và còn rất nhiều API hay ho khác nữa, bạn có thể tự khám phá thêm [tại đây](https://developer.github.com/v3/) nhé.

## **#Tại sao nên lại là Github? Hướng dẫn sử dụng Github**

Mặc dù Github có nhiều lợi ích như vậy. Nhưng sẽ có nhiều bạn thắc mắc: Tại sao không sử dụng Dropbox hay các dịch vụ tương tự để quản lý source code? Các dịch vụ đó cũng cho phép đồng bộ source code của các bạn trong team về một server mà?



Để trả lời cho câu hỏi này, mình sẽ lấy một ví dụ: Giả sử dự án của bạn có 2 developer cùng sửa một file, sau đó commit đồng thời lên server. Với Dropbox thì người commit lên trước sẽ được ưu tiên sửa file.

Nhưng với Github, việc commit đồng thời không phải là vấn đề nghiêm trọng. Vì Git sẽ ghi lại lịch sử commit, đảm bảo các commit được tổ chức tốt, tránh sự hỗn loạn giữa các version của file sửa đổi được gửi lên server.

Do đó, việc sử dụng  Github sẽ tránh được tất cả sự nhầm lẫn khi commit. Đảm bảo cùng làm việc trên một source code trở lên dễ dàng hơn rất nhiều.

### **#Cách tạo một GitHub Repository**

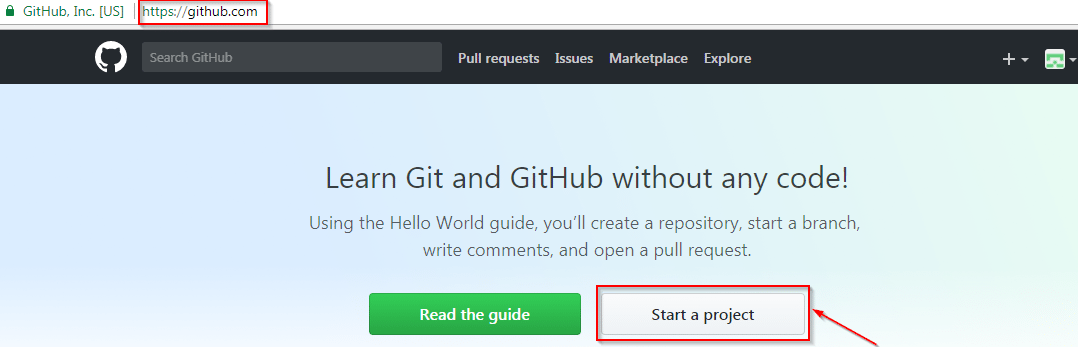
Repository là một không gian để lưu trữ dự án của bạn. Do tính chất phân tán của Git, nên có thể hiểu repository là nơi lưu trữ mã nguồn ở cả local và server.

Bạn có thể lưu trữ file code, text, hình ảnh hoặc bất kỳ loại tệp nào trong repository.

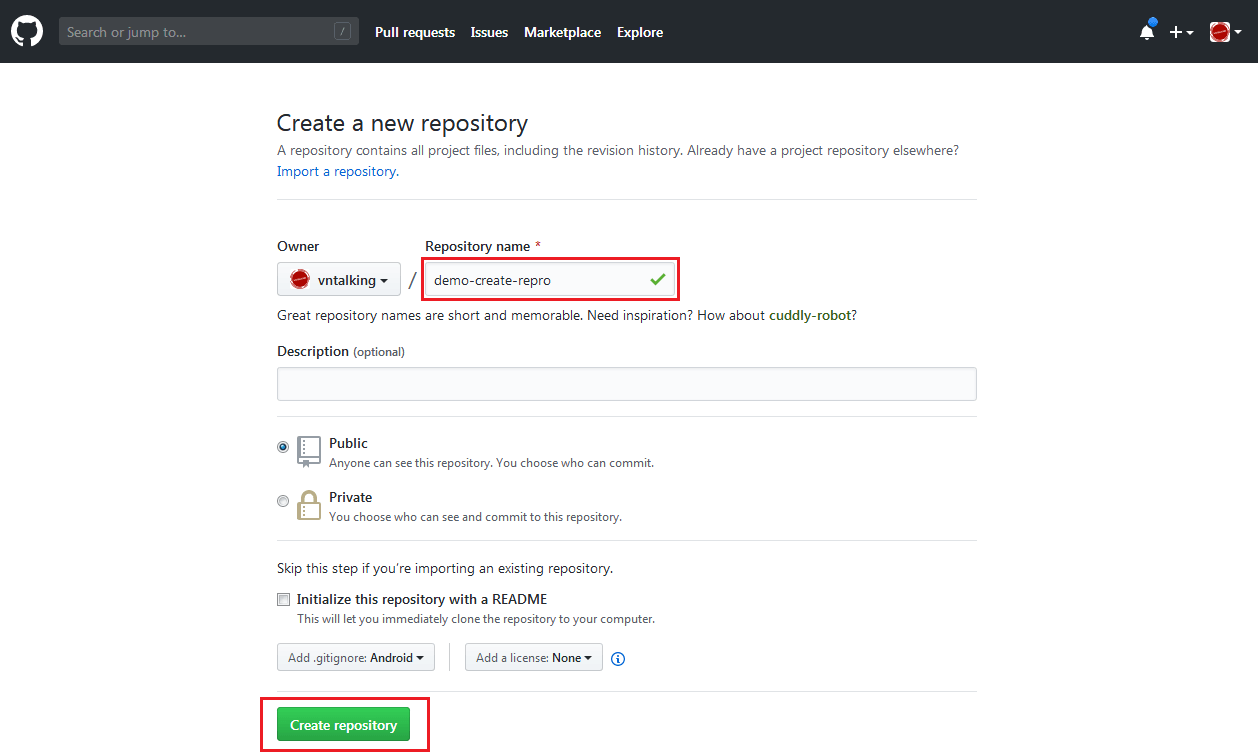
Để tạo một repository trên Github bạn làm như sau:

* Vào Github, đăng ký một tài khoản bằng cách click vào “Sign up for Github”.
* Sau khi đăng ký và kích hoạt thành công. Bạn bắt đầu tạo mới một project với “Start a new project”.

Bạn có thể xem hình bên dưới cho rõ thêm nhé:



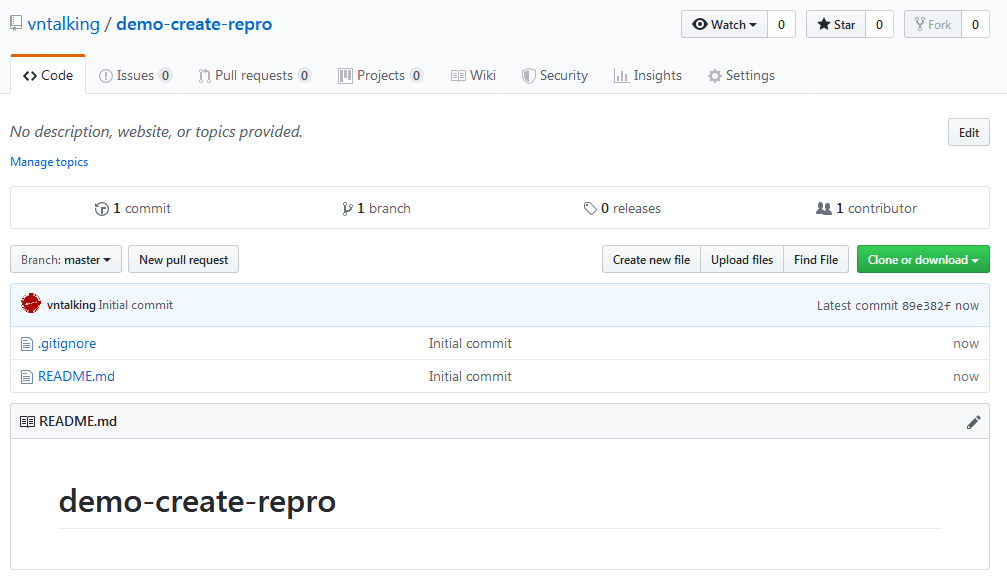
* Nhập tên Repositoty và nhấn nút “Create Repository”. Ngoài ra, bạn cũng có thể thêm mô tả cho repo ( Cái này chỉ là lựa chọn, không bắt buộc phải có).



Trong đó, bạn lưu ý 2 options sau:

* Theo mặc định thì repository để là public. Tức là ai cũng có thể xem được repo này của bạn. Nếu dự án của bạn chưa muốn công khai mà chỉ muốn quản lý nội bộ thì chọn Private.
* Bạn có thêm một README file để giới thiệu repo kèm với một file .gitignore. Github đã có sẵn template .gitignore cho bạn, cứ chọn một template phù hợp với mã nguồn dự án là được.

Khi tạo xong, repo sẽ như sau:



Khi đã có repository, bạn có thể clone, pull, push… source code của mình lên đó rồi.

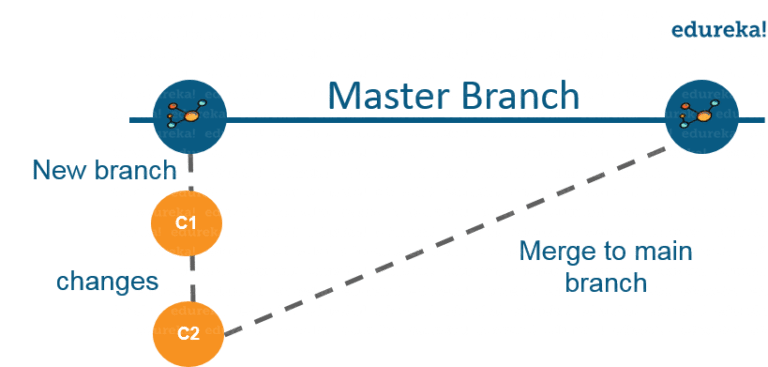
Phần tiếp theo của bài viết, chúng ta sẽ tìm hiểu về branch trên Github.

### **#Tạo branch trên Github**

Branch giúp bạn làm việc trên các phiên bản khác nhau của cùng một repository. Mình lấy ví dụ: Bạn muốn phát triển thêm một tính năng mới cho dự án. Nhưng bạn lại phần sửa đổi đó lại ảnh hưởng tới dự án chính. Lúc này branch ra đời.

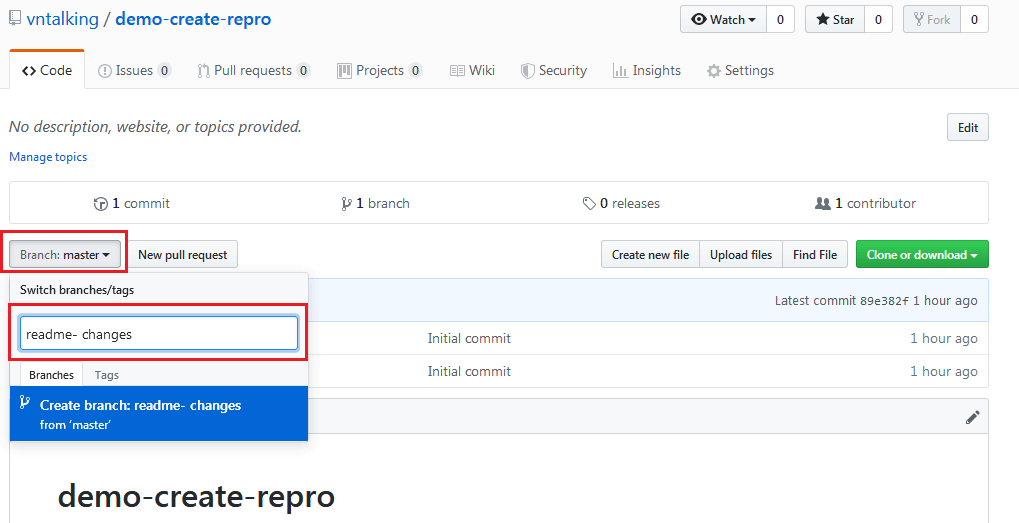
Branch cho phép bạn chuyển đổi qua lại giữa các trạng thái và phiên bản khác nhau của dự án.

Với kịch bản ví dụ ở trên, bạn có thể tạo mới một branch và phát triển tính năng mới trên đó mà không ảnh hưởng tới master branch. Sau khi đã implement xong, test kỹ càng thì có thể merge vào master branch.



Để tạo một branch trên Github, bạn làm như sau:

* Click vào dropdown branch: Branch: master
* Khi danh sách branch xổ ra, bạn có thể tạo thêm một branch mới tại đây



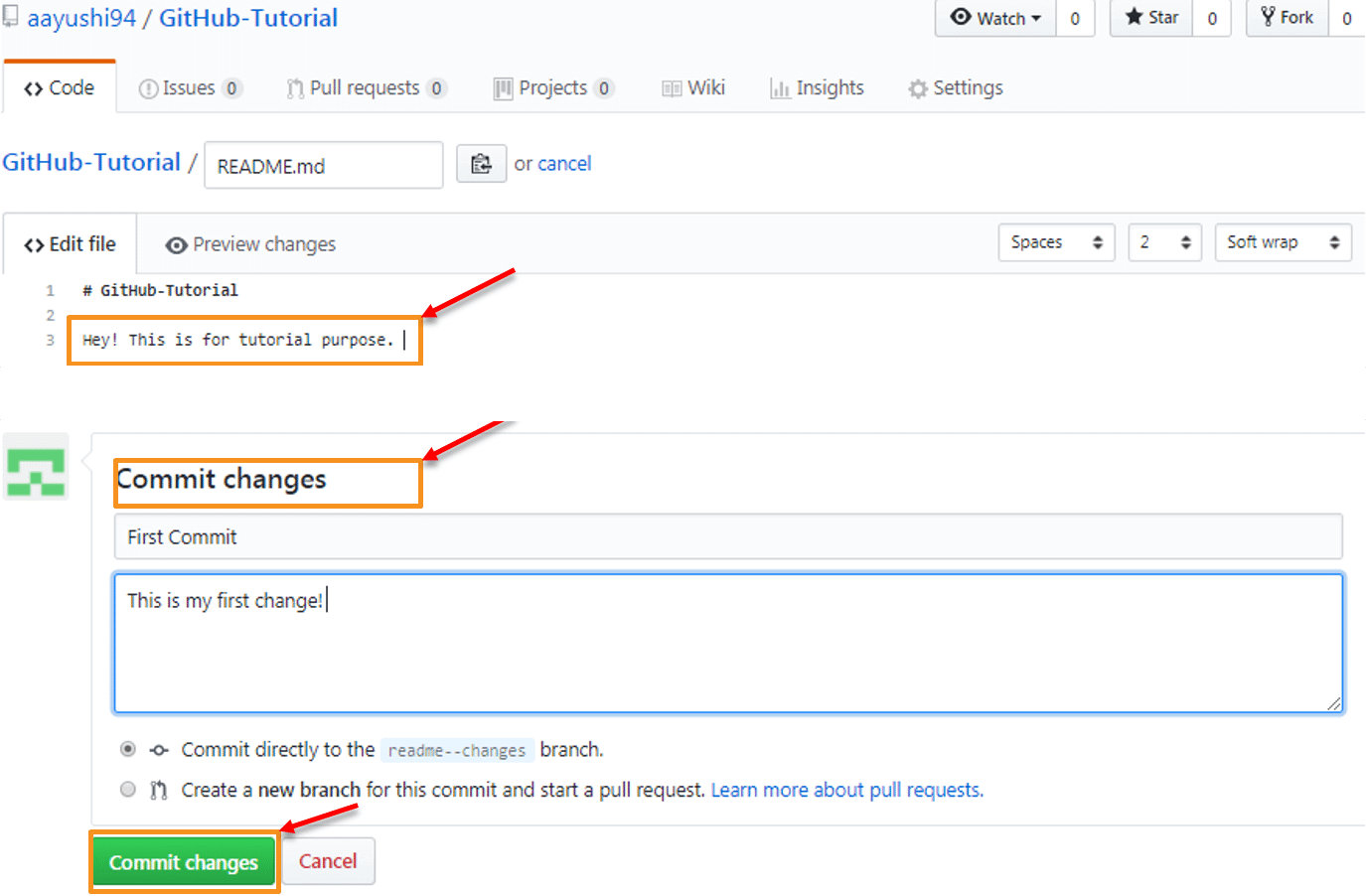
## **#Hướng dẫn sử dụng Github chi tiết**

Github có đầy đủ những command của git. Nhưng do bài viết này quá dài nên mình sẽ không liệt kê hết tất cả chúng ở đây. Mình sẽ tập trung hướng dẫn những command cơ bản nhất, hay dùng trên Github.

### **1. Commit Command**

Commit command cho phép bạn lưu lại những thay đổi của file. Khi bạn commit, nên viết mô tả rõ ràng trong commit message. Điều này sẽ giúp cho quản lý dự án tốt hơn, có thể theo dõi, review những thay đổi source code sau này.  
Để tạo commit, bạn làm như sau:

* Chọn file muốn sửa
* Chọn “Edit” để sửa file.
* Sau khi sửa xong thì điền thông tin message và nhấn Commit.



### **2. Pull Command**

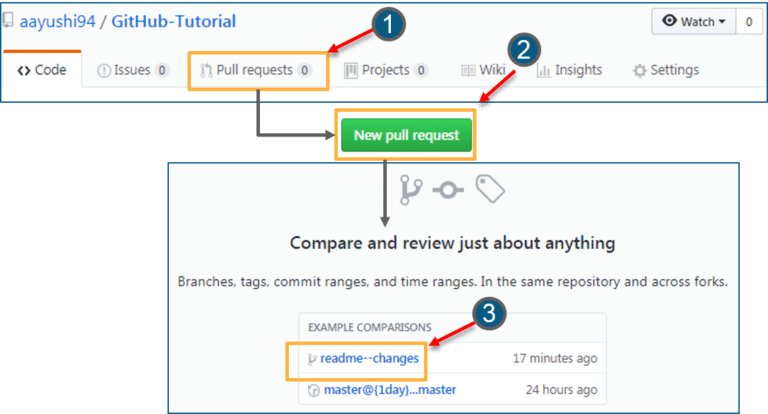
Lệnh PULL request là lệnh quan trọng  nhất trên Github. Nó cho biết những thay đổi trong source code, và yêu cầu owner của source code xem xét nó và merge nó vào master branch.

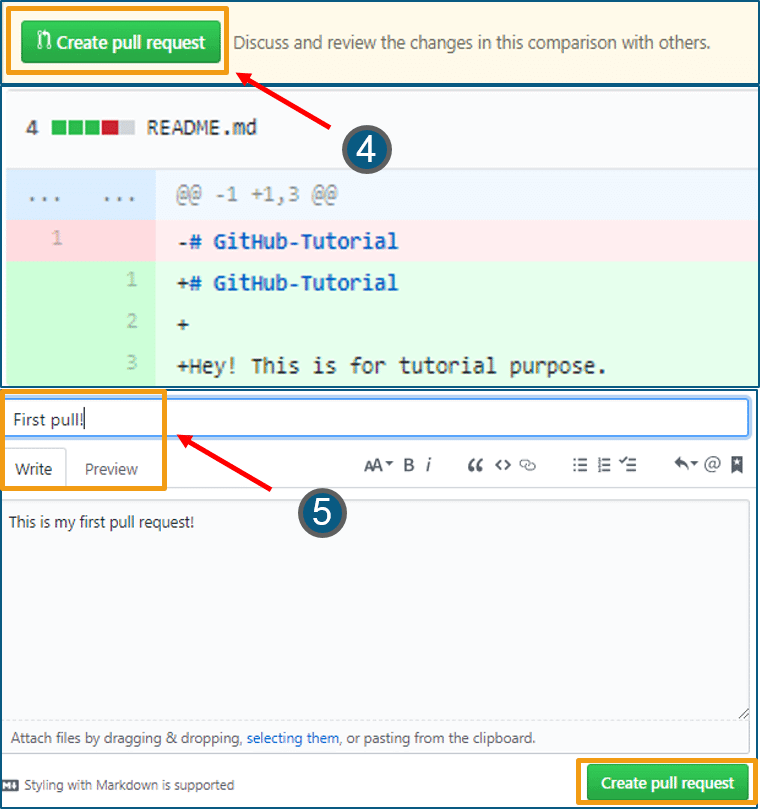
Một khi commit xong, bất kể ai cũng có thể update sự thay đổi và thảo luận về sự thay đổi đó.

Tính năng này rất hay cho các dự án mã nguồn mở. Khi mà bất kì cũng có thể đóng góp công sức cho dự án. Tất nhiên, mọi sự thay đổi đều phải được sự đồng ý của owner dự án.

Ở đây, mình cần làm rõ hơn với các bạn đỡ nhầm lẫn về lệnh Pull:

* Lệnh pull request : Là lệnh yêu cầu chủ owner dự án xem xét một thay đổi nào đó trước khi merge vào master branch.
* Lệnh Pull: đây là lệnh của git, đơn thuần có thể hiểu là lệnh update source code từ server về local. Nếu có bất kì sự xung đột code nào (conflict) thì bạn cần phải resolve nó.



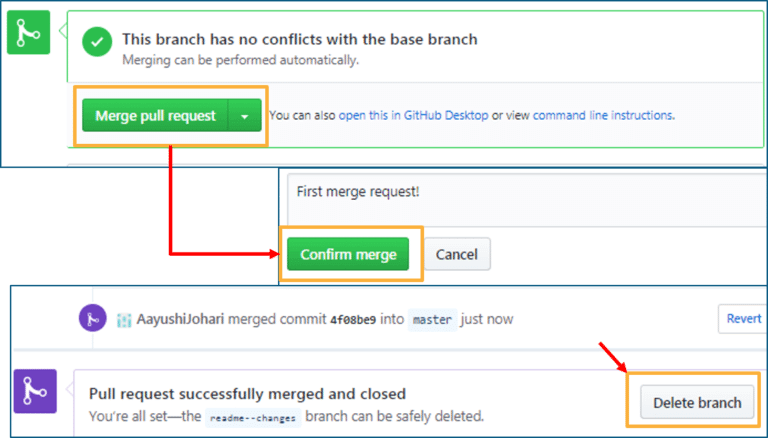


### **3. Merge command**

Lệnh cơ bản cuối cùng mà mình muốn nhắc đến là merge. Lệnh merge này cho phép bạn hợp nhất những thay đổi vào một branch.

* Click vào “Merge pull request” để hợp nhất những thay đổi vào master branch.
* Click vào “Confirm merge”.

Bạn có thể tham khảo hình bên dưới:

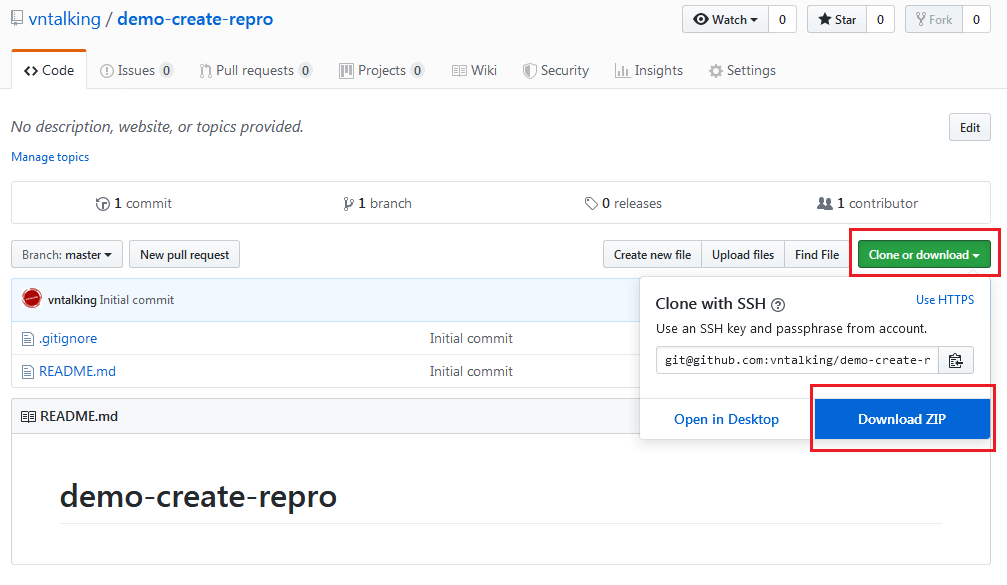


## **#Cloning dự án từ Github**

Tiếp tục hướng dẫn sử dụng Github. Đây có lẽ là thao tác bạn hay dùng nhất khi tìm kiếm mã nguồn mở trên mạng. Khi bạn thấy một dự án nào đó hay ho và có thể ứng dụng được cho dự án của mình, bạn muốn download dự án này về máy tính để tham khảo.

Có 2 cách để tải dự án từ Github:

* Một là bạn chọn Zip toàn bộ dự án và tải về



* Hai là bạn có thể clone dự án về bằng lệnh git. Bạn cũng click vào “Clone or Download”. Sau đó copy đường dẫn và gõ trong cửa sổ terminal trên máy tính như sau
* git clone git@github.com:vntalking/demo-create-repro.git

### **#Tạm kết**

Trên đây, mình đã hoàn thành hướng dẫn cách sử dụng Github cơ bản. Tất cả các thao tác đều thực hiện trên web.

Nếu bạn muốn tương tác với Github qua command hoặc một phần mềm quản lý Git thì có thể tham khảo thêm. Mình có gợi ý một số bài viết như:

* [Những lệnh Git cơ bản](https://github.com/joshnh/Git-Commands)
* Phần mềm quản lý Git tốt nhất: [Source Tree](https://www.sourcetreeapp.com/), [Gitkraken](https://www.gitkraken.com/)

Mình hi vọng, qua bài viết này, bạn sẽ có những kỹ năng cần thiết để làm việc với Git.