1. **Lịch sử của GitHub**

GitHub được tạo ra bởi Tom Preston – Werner, Chris Wanstrath và PJ Hyett, được viết bằng ngôn ngữ Ruby on Rails và Erlang. Và được phát hành chạy chính thức vào tháng 4 năm 2008. Tính đến tháng 3 năm 2018 GitHub được đánh giá là dịch vụ lưu trữ mã nguồn lập trình lớn nhất thế giới.

Hiện nay, GitHub có khoảng hơn 25 triệu người dùng và hơn 80 triệu mã nguồn dự án. Do đó GitHub trở thành 1 phần không thể thiếu đối với các cộng đồng phát triển mã nguồn mở và cộng đồng các Developer.

1. **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_(ph%E1%BA%A7n_m%E1%BB%81m)) 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.

Bạn có thể hình dung GitHub như một trang mạng xã hội dành riêng cho lập trình viên cung cấp các chức năng social networking như feeds, follower và network graph để các lập trình viên học hỏi kinh nghiệm từ lịch sử commit.

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.

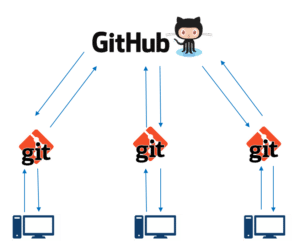
1. **Tính năng API Github**

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.

1. **Tại sao nên lại là 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.

1. **Cách thức hoạt động trên GitHub là gì?**

Biết về khái niệm GitHub là gì thôi là chưa đủ. Bạn muốn tìm hiểu rõ hơn về GitHub thì bạn phải nắm được cách thức hoạt động của nó. GitHub gồm hai nền tảng khi làm việc đó là local workflow và server workflow.

Với local workflow, bạn có thể thay đổi source code, sau đó mới xác nhận các thay đổi tại server workflow. Nhưng cũng cần lưu ý rằng, bản xác nhận trên server phải bản hoàn chỉnh, nếu chưa hoàn chỉnh có thể ảnh hưởng đến các thành viên khác trong team khi sử dụng kho lưu trữ.

Bạn sử dụng kho lưu trữ của server, người dùng phải cung cấp mã xác nhận, từ đó sẽ so sánh SSH key ở local của bạn và SSH key trên server tương ứng với tài khoản đã đăng ký trước đó.

1. Cách tạo một GitHub Repository

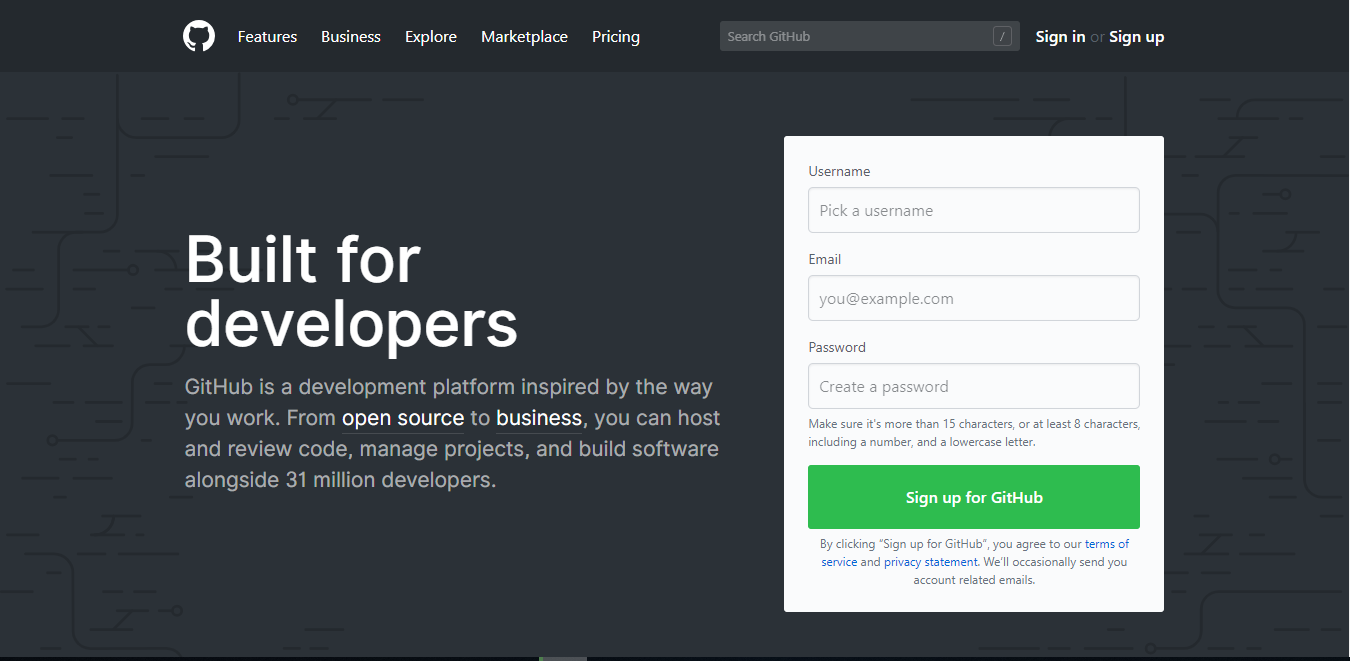
Thường sẽ sử dụng 2 lệnh là **git add** và **git commit**:

* **Git add**: Dùng để add file đã thay đổi vào stage.
* **Git commit**: Dùng để xác nhận các file đã add vảo satge lên repository trên local.

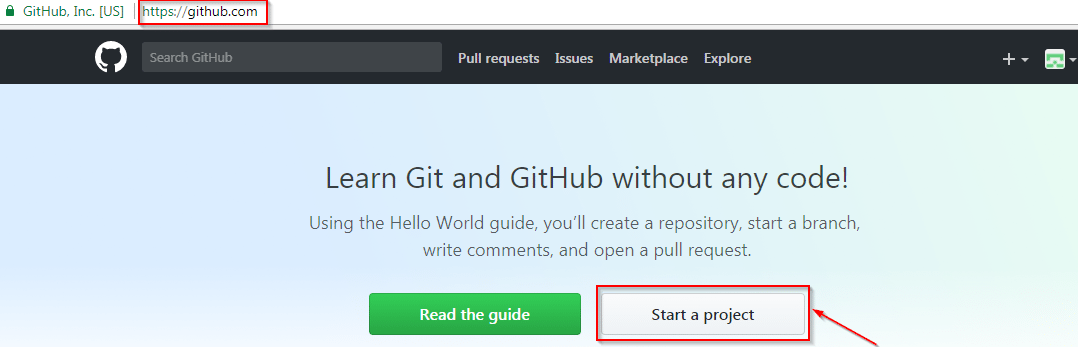
Repository là một không gian để lưu trữ dự án của bạn. Do tính chất phân tán của Git, nên có thể hiểu repository là nơi lưu trữ mã nguồn ở cả local và server.

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

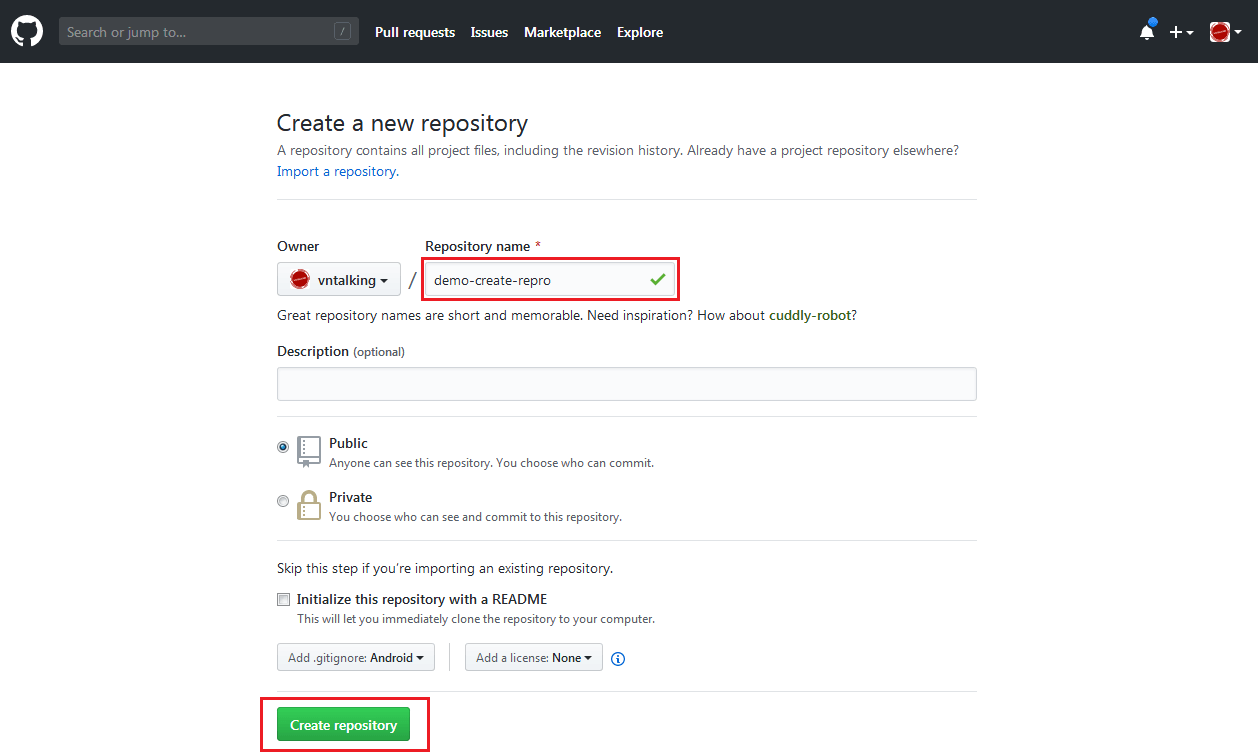
Để tạo một repository trên Github bạn truy cập: [**https://github.com/**](https://github.com/)



* 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”.



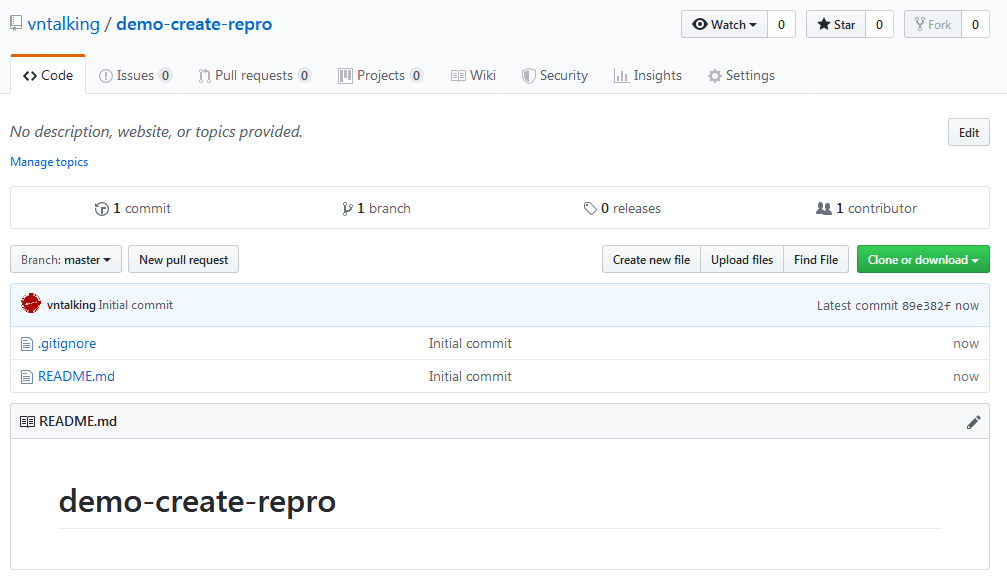
* Nhập tên Repositoty và nhấn nút “Create Repository”. Ngoài ra, bạn cũng có thể thêm mô tả cho repo ( Cái này chỉ là lựa chọn, không bắt buộc phải có).



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** của bạn sẽ như sau:



1. **Một số tính năng của GitHub**

GitHub là trang mạng xã hội lớn nhất dành cho lập trình viên bao gồm những tính năng cốt lõi như:

**Wiki**, **issue**, **thống** **kê**, **đổi tên dự án**.

**Watch project**: Có thể theo dõi hoạt động các dự án của người khác, có thể xem quá trình phát triển phần mềm và thực trạng dự án như thế nào?

**Follow user:** Theo dõi hoạt động của người khác.