

BLOG

Git và Github là gì? Cách sử dụng Git – Cập nhật 2020

13/04/2020

Git là gì?

Git là một **phần mềm kiểm soát phiên bản** nguồn mở, miễn phí. Nó được tạo bởi Linus Torvalds vào năm 2005. Công cụ trong [cộng đồng IT](#) này là một hệ thống kiểm soát phiên bản ban đầu được phát triển để làm việc với một số nhà phát triển trên nhân Linux.

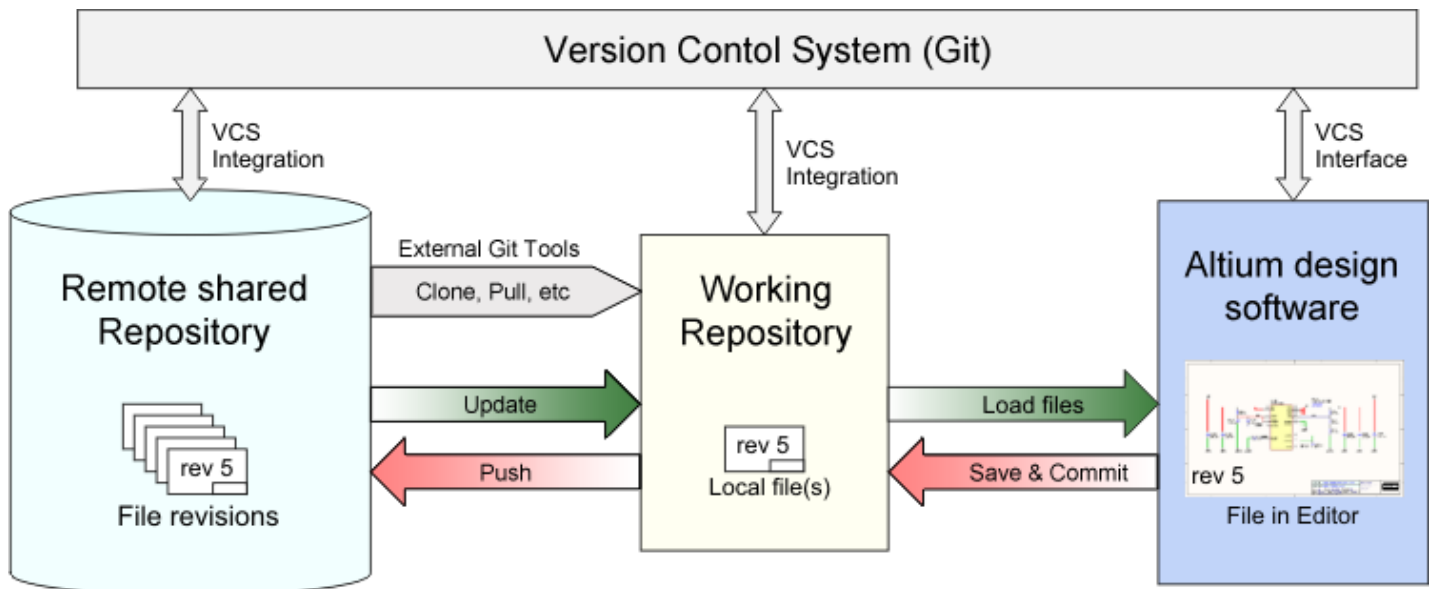


Điều này về cơ bản có nghĩa là Git là một trình theo dõi nội dung. Vì vậy, Git có thể được sử dụng

Các dự án thực tế thường có nhiều nhà phát triển làm việc song song. Vì vậy, họ cần một hệ thống kiểm soát phiên bản như Git để đảm bảo rằng không có xung đột code giữa chúng.

Ngoài ra, các yêu cầu trong dự án như vậy thay đổi thường xuyên. Vì vậy, một hệ thống kiểm soát phiên bản cho phép các nhà phát triển quay lại phiên bản cũ hơn của code của họ.

Hệ thống chi nhánh trong Git cho phép các nhà phát triển làm việc riêng lẻ trên một tác vụ (Ví dụ: Một nhánh -> Một tác vụ HOẶC Một nhánh -> Một nhà phát triển). Về cơ bản, hãy nghĩ về Git như một ứng dụng phần mềm nhỏ kiểm soát cơ sở code của bạn, nếu bạn là nhà phát triển.



Cách Git hoạt động

Nếu chúng ta muốn bắt đầu sử dụng Git, chúng ta cần biết nơi lưu trữ các repository của mình.

Một kho lưu trữ (hay gọi tắt là Repo) là một dự án chứa nhiều tệp. Trong trường hợp của chúng ta, một kho lưu trữ sẽ chứa các tệp dựa trên code.

Có hai cách bạn có thể lưu trữ kho của bạn. Một là trực tuyến (trên đám mây) và thứ hai là ngoại tuyến (tự cài đặt trên máy chủ của bạn).



Trước khi sử dụng Git, chúng ta nên biết lý do tại sao chúng ta cần nó

Git giúp dễ dàng đóng góp cho các dự án nguồn mở

Gần như mọi dự án nguồn mở đều sử dụng GitHub để quản lý các dự án của họ. Sử dụng GitHub là miễn phí nếu dự án của bạn là nguồn mở và nó bao gồm wiki và trình theo dõi vấn đề giúp dễ dàng đưa vào tài liệu chuyên sâu hơn và nhận phản hồi về dự án của bạn.

Nếu bạn muốn đóng góp, bạn chỉ cần rẽ nhánh (lấy một bản sao) của một dự án, thực hiện các thay đổi của bạn và sau đó gửi pull request dự án bằng giao diện web của GitHub. Pull request này là cách bạn nói với dự án là bạn đã sẵn sàng để họ xem xét các thay đổi của bạn.

Tài liệu

Bằng cách sử dụng GitHub, bạn sẽ dễ dàng có được tài liệu tuyệt vời. Phần trợ giúp và hướng dẫn của họ có bài viết cho gần như bất kỳ chủ đề nào liên quan đến Git mà bạn có thể nghĩ đến.

Tùy chọn tích hợp

GitHub có thể tích hợp với các nền tảng phổ biến như Amazon và Google Cloud, với các dịch vụ như Code Climate để theo dõi phản hồi của bạn và có thể đánh dấu cú pháp trong hơn 200 ngôn ngữ lập trình khác nhau.

Theo dõi các thay đổi trong code của bạn qua các phiên bản



GitHub xử lý vấn đề này bằng cách theo dõi tất cả các thay đổi đã được đẩy vào kho lưu trữ.

Giống như sử dụng Microsoft Word hoặc Google Drive, bạn có thể có lịch sử phiên bản code của mình để các phiên bản trước không bị mất. Thật dễ dàng để trở lại phiên bản trước và đóng góp công việc của bạn.

Trưng bày công việc của bạn

Bạn có phải là nhà phát triển muốn thu hút nhà tuyển dụng? GitHub là công cụ tốt nhất bạn có thể dựa vào cho việc này.

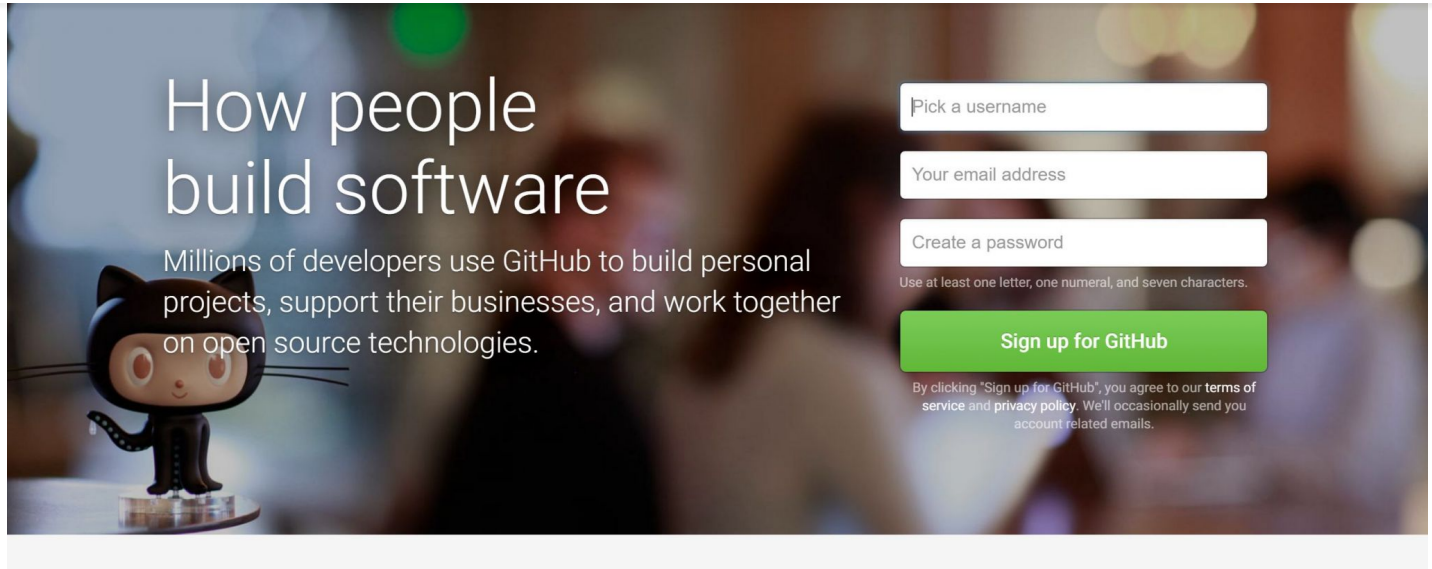
Ngày nay, khi tìm kiếm tân binh cho các dự án của họ, hầu hết các công ty đều nhìn vào hồ sơ GitHub. Nếu hồ sơ của bạn có sẵn, bạn sẽ có cơ hội được tuyển dụng cao hơn ngay cả khi bạn không đến từ một trường đại học hoặc cao đẳng lớn.

Bây giờ chúng ta sẽ tìm hiểu cách sử dụng Git & GitHub

Tạo tài khoản GitHub

Để tạo tài khoản của bạn, bạn cần truy cập trang web của [GitHub](#) và điền vào mẫu đăng ký.





Trang web chính thức của GitHub

Cài đặt Git

Bây giờ chúng ta cần cài đặt các công cụ của Git trên máy tính. Chúng ta sẽ sử dụng CLI để liên lạc với GitHub.

Đối với Ubuntu:

1. Đầu tiên, cập nhật các gói của bạn.

sudo apt update

2. Tiếp theo, cài đặt Git và GitHub với apt-get

sudo apt-get install git

3. Cuối cùng, xác minh rằng Git đã được cài đặt đúng

git -version



```
git config --global user.name "MV Thanoshan"
```

```
git config --global user.email "example@mail.com"
```

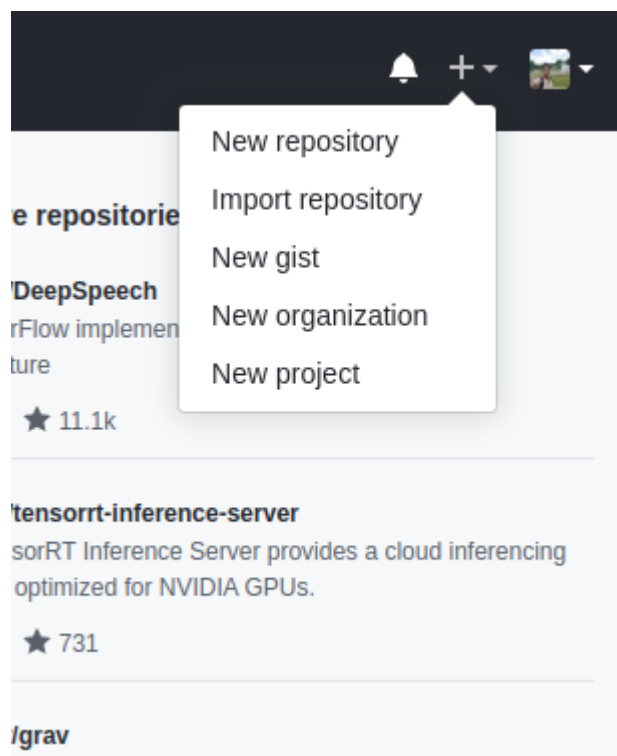
Làm việc với các dự án GitHub

Chúng ta sẽ làm việc với các dự án GitHub theo hai cách.

LOẠI 1: TẠO KHO LƯU TRỮ, SAO CHÉP NÓ VÀO PC CỦA BẠN VÀ LÀM VIỆC VỚI NÓ. (KHUYẾN KHÍCH)

Loại 1 liên quan đến việc tạo một kho lưu trữ hoàn toàn mới trên GitHub, sao chép nó vào máy tính, làm việc với dự án và đẩy nó trở lại.

Tạo một kho lưu trữ mới bằng cách nhấp vào nút Kho lưu trữ mới của Wikipedia trên trang web GitHub.




Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner

Repository name *

 ThanoshanMV ▾

/ My-GitHub-Project ✓

Great repository names are short and memorable. Need inspiration? How about [stunning-octo-funicular?](#)

Description (optional)

This is my first project



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Add a license: **None** ▾ ⓘ

Create repository

Kho GitHub đầu tiên của bạn được tạo.

Nhiệm vụ đầu tiên của bạn là lấy một bản sao của kho lưu trữ vào máy tính của mình. Để làm được điều đó, bạn cần phải sao chép lại kho lưu trữ trên máy tính của bạn.

Để sao chép một kho lưu trữ có nghĩa là bạn đang lấy một kho lưu trữ trên máy chủ và sao chép nó vào máy tính của bạn – giống như tải xuống. Trên trang kho lưu trữ, bạn cần lấy địa chỉ HTTPS.



This is my first project

Edit

[Manage topics](#)

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

ThanoshanMV Initial commit

README.md

Initial commit

README.md

Clone with HTTPS

[Use SSH](#)

Use Git or checkout with SVN using the web URL.

<https://github.com/ThanoshanMV/My-GitHub-Project>

Download ZIP

My-GitHub-Project

This is my first project

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

Khi bạn có địa chỉ của kho lưu trữ, bạn cần sử dụng terminal của mình. Sử dụng lệnh sau trên terminal của bạn. Khi bạn đã sẵn sàng, bạn có thể nhập lệnh này:

git clone [HTTPS ADDRESS]

Lệnh này sẽ tạo một bản sao cục bộ của kho lưu trữ tại địa chỉ đã cho.

```
Cloning into 'My-GitHub-Project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

Thông báo đầu ra của lệnh git clone bản sao

Bây giờ, kho lưu trữ của bạn là trên máy tính của bạn. Bạn cần di chuyển trong nó bằng lệnh sau.

cd [NAME OF REPOSITORY]



LƯU Ý: Khi bạn sao chép, Git sẽ tạo một kho lưu trữ trên máy tính của bạn. Nếu bạn muốn, bạn có thể truy cập dự án của mình bằng giao diện người dùng máy tính thay vì sử dụng lệnh 'cd' trên terminal.

Bây giờ, trong thư mục đó, chúng ta có thể tạo các tệp, làm việc với chúng và lưu chúng cục bộ. Để lưu chúng ở một nơi xa xôi – như GitHub – chúng ta đã thực hiện một quy trình gọi là commit. Để làm điều này, quay trở lại terminal của bạn. Nếu bạn đã đóng nó, như tôi đã nói trước đây, hãy sử dụng lệnh 'cd'.

cd [NAME OF REPOSITORY]

Bây giờ, trong terminal, bạn đang ở trong thư mục kho lưu trữ của bạn. Có 4 bước trong một commit: 'status', 'add', 'commit' và 'push'. Tất cả các bước sau đây phải được thực hiện trong dự án của bạn. Chúng ta hãy đi qua từng cái một.

1. 'status': Điều đầu tiên bạn cần làm là kiểm tra các tệp tin bạn đã sửa đổi. Để làm điều này, bạn có thể gõ lệnh sau để làm cho một danh sách các thay đổi xuất hiện.

git status

```
thanos18@life companion:~/My-GitHub-Project$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        sample.html

nothing added to commit but untracked files present (use "git add" to track)
```

2. 'add': Với sự trợ giúp của danh sách thay đổi, bạn có thể thêm tất cả các tệp bạn muốn tải lên bằng lệnh sau,

git add [FILENAME] [FILENAME] [...]

Trong trường hợp của chúng ta, chúng ta sẽ thêm một tệp HTML đơn giản.



3. commit: Bây giờ chúng ta đã thêm các tệp mà chúng ta chọn, chúng ta cần viết một thông điệp để giải thích những gì chúng ta đã làm. Thông báo này có thể hữu ích sau này nếu chúng ta muốn kiểm tra lịch sử thay đổi. Dưới đây là một ví dụ về những gì chúng ta có thể đặt trong trường hợp của chúng ta.

git commit -m "Added sample HTML file that contain basic syntax"

```
thanos18@lifecompanion:~/My-GitHub-Project$ git commit -m "Added sample HTML file that contain basic syntax"
[master b234227] Added sample HTML file that contain basic syntax
1 file changed, 12 insertions(+)
create mode 100644 sample.html
```

4. 'push': Để làm điều đó, chúng ta phải 'đẩy' các tệp của mình lên Remote. Remote là một bản sao trùng lặp của kho lưu trữ ở một nơi khác trên máy chủ từ xa. Để làm điều này, chúng ta phải biết tên của Remote (Chủ yếu là từ xa được đặt tên gốc). Để tìm ra tên đó, gõ lệnh sau.

git remote

Như bạn có thể thấy trong hình trên, nó nói rằng tên từ là origin. Bây giờ chúng ta có thể 'đẩy' công việc của mình một cách an toàn bằng lệnh sau.

git push origin master

Bây giờ, nếu chúng ta truy cập kho lưu trữ của mình trên trang web GitHub, chúng ta có thể thấy tệp sample.html mà chúng ta đã đẩy đến từ xa – GitHub!



This is my first project

[Edit](#)[Manage topics](#)

2 commits

1 branch

0 releases

1 contributor

Branch: master

[New pull request](#)[Create new file](#)[Upload files](#)[Find File](#)[Clone or download](#) **ThanoshanMV** Added sample HTML file that contain basic syntax

Latest commit b234227 14 minutes ago

[README.md](#)

Initial commit

1 hour ago

[sample.html](#)

Added sample HTML file that contain basic syntax

14 minutes ago

README.md

My-GitHub-Project

This is my first project

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

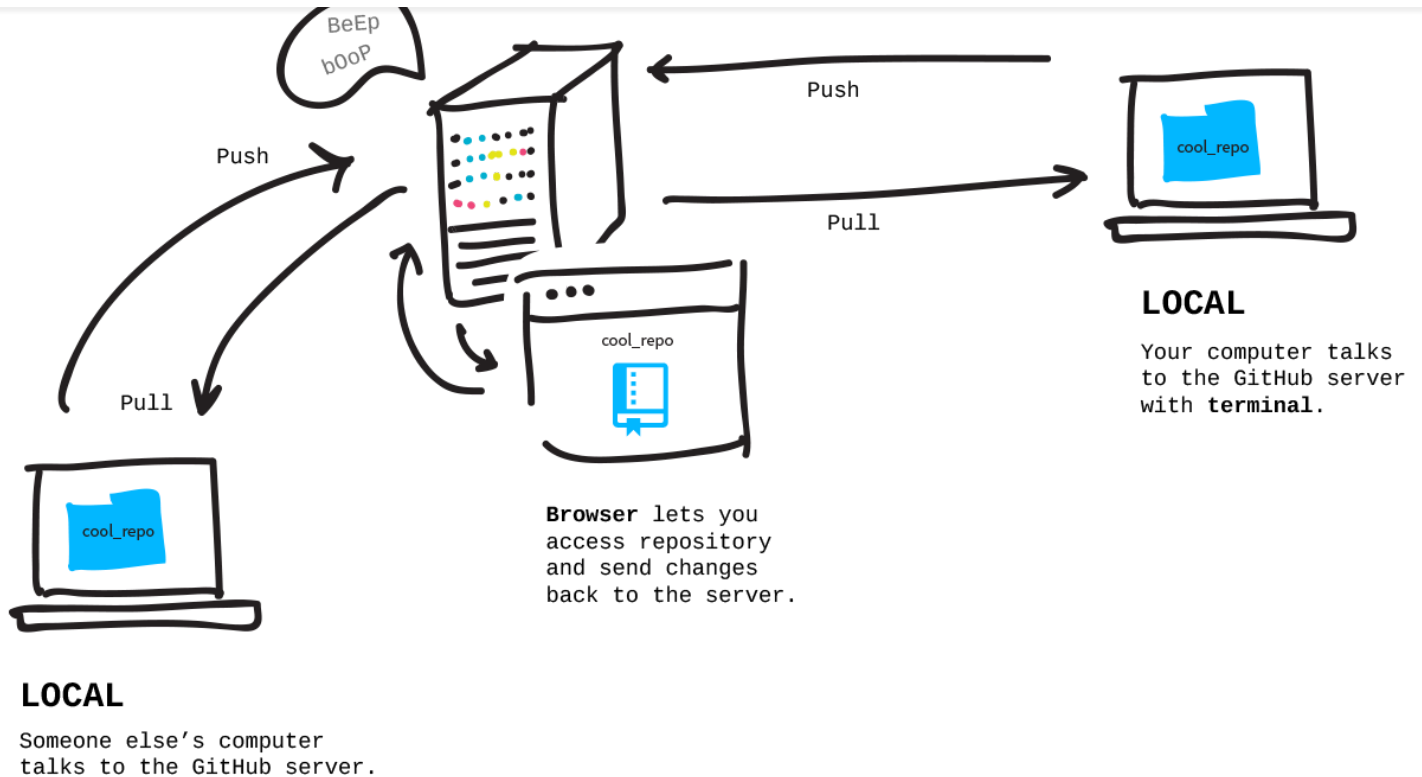
LƯU Ý : Đôi khi, khi bạn đang sử dụng các lệnh Git trong terminal, nó có thể dẫn bạn đến trình soạn thảo văn bản VIM (trình soạn thảo văn bản dựa trên CLI). Vì vậy, để thoát khỏi nó, bạn phải gõ

:q

và ENTER.

Mô tả cách pull và push làm việc:





- Pull là hành động nhận từ GitHub.
- Push là hành động gửi đến GitHub.

LOẠI 2: LÀM VIỆC TRÊN DỰ ÁN CỦA BẠN CỤC BỘ SAU ĐÓ TẠO KHO LƯU TRỮ TRÊN GITHUB VÀ ĐẨY NÓ VÀO REMOTE

Loại 2 cho phép bạn tạo một kho lưu trữ mới từ một thư mục hiện có trên máy tính và gửi nó đến GitHub. Trong rất nhiều trường hợp, bạn có thể đã thực sự tạo ra một cái gì đó trên máy tính mà bạn muốn đột nhiên biến thành một kho lưu trữ trên GitHub.

Tôi sẽ giải thích điều này với bạn bằng một dự án web dạng Khảo sát mà tôi đã thực hiện trước đó chưa được thêm vào GitHub.

Như tôi đã đề cập, khi thực hiện bất kỳ lệnh Git nào, chúng ta phải đảm bảo rằng chúng ta đang ở đúng thư mục trong terminal.

Theo mặc định, bất kỳ thư mục nào trên máy tính không phải là kho lưu trữ Git – nhưng ta có thể biến nó thành kho lưu trữ Git bằng cách thực hiện lệnh sau trong terminal.



```
thanos18@life companion: ~/Code-School/WebDev/Survey Form Project$ git init
Initialized empty Git repository in /home/thanos18/Code-School/WebDev/Survey Form Project/.git/
```

Sau khi chuyển đổi thư mục của chúng ta sang kho lưu trữ Git, điều đầu tiên chúng ta cần làm là kiểm tra các tệp chúng ta có bằng cách sử dụng lệnh sau.

git status

```
thanos18@life companion: ~/Code-School/WebDev/Survey Form Project$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        index.html
        style.css

nothing added to commit but untracked files present (use "git add" to track)
```

Vì vậy, có hai tệp tin trong thư mục đó mà chúng ta cần để thêm vào Repo của chúng ta.

git add [FILENAME] [FILENAME] [...]

LƯU Ý: Để thêm vào tất cả các tệp trong Kho lưu trữ của chúng ta, chúng ta có thể sử dụng lệnh sau:

git add .

Sau khi khu vực tổ chức (quá trình thêm) hoàn tất, chúng ta có thể kiểm tra xem các tệp có được thêm thành công hay không bằng cách thực hiện git status

Nếu những tệp cụ thể đó có màu xanh lục như hình dưới đây, bạn đã hoàn thành công việc của mình!



```
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   index.html
    new file:   style.css
```

Sau đó, chúng ta phải commit với một mô tả trong đó.

git commit -m "Adding web Survey form"

Nếu kho lưu trữ của tôi bắt đầu trên GitHub và tôi đã đưa nó xuống máy tính của mình, một Remote đã được gắn vào nó (Loại 1). Nhưng nếu tôi đang khởi động kho lưu trữ của mình trên máy tính thì nó không có Remote, vì vậy tôi cần thêm Remote đó (Loại 2).

Vì vậy, để thêm Remote, chúng ta phải vào GitHub trước. Tạo một kho lưu trữ mới và đặt tên cho nó bất cứ điều gì bạn muốn lưu trữ trong GitHub. Sau đó nhấp vào nút Tạo kho lưu trữ.



LƯU Ý: Trong Loại 2, Vui lòng không khởi tạo kho lưu trữ với tệp README khi tạo kho lưu trữ mới trên trang web GitHub.



A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner

Repository name *

 **ThanoshanMV** / **SOCS-Survey-Form** 

Great repository names are short and memorable. Need inspiration? How about [miniature-umbrella?](#)

Description (optional)

A survey form that is created to my University's Society of Computer Science(SOCS).|



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: **None**

Add a license: **None**



Create repository



Sau khi nhấp vào nút Tạo kho lưu trữ, bạn sẽ tìm thấy hình ảnh bên dưới dưới dạng trang web.



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# SOCS-Survey-Form" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/ThanoshanMV/SOCS-Survey-Form.git
git push -u origin master
```



...or push an existing repository from the command line


```
git remote add origin https://github.com/ThanoshanMV/SOCS-Survey-Form.git
git push -u origin master
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

 **ProTip!** Use the URL for this page when adding GitHub as a remote.

Sao chép địa chỉ HTTPS. Bây giờ chúng ta sẽ tạo Remote cho kho lưu trữ của mình.

git remote add origin [HTTPS ADDRESS]

Sau khi thực hiện lệnh này, chúng ta có thể kiểm tra xem chúng ta đã thêm thành công Remote hay chưa bằng lệnh sau

git remote

Và nếu nó xuất ra Origin thì bạn đã thêm Remote vào dự án của mình.

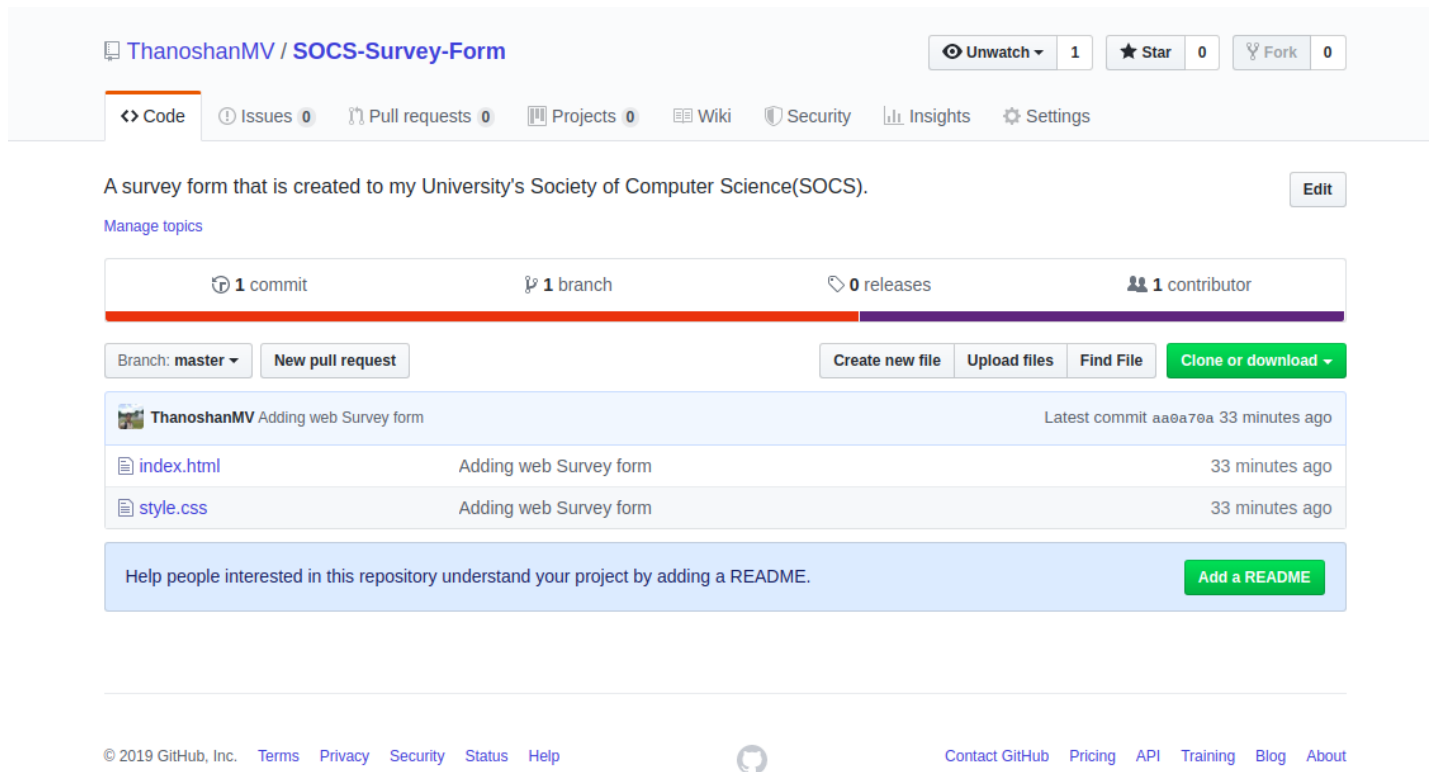
LƯU Ý: Chỉ cần nhớ rằng chúng ta có thể nêu bất kỳ tên nào cho Remote bằng cách thay đổi tên xuất xứ. Ví dụ:

git remote add [REMOTE NAME] [HTTPS ADDRESS]

Bây giờ, chúng ta có thể đẩy dự án của mình lên GitHub mà không gặp vấn đề gì!



Sau khi hoàn thành từng bước một, nếu bạn truy cập GitHub, bạn có thể tìm thấy kho lưu trữ của mình với các tệp!



Kết luận

Cảm ơn mọi người đã đọc. Tôi chỉ giải thích những điều cơ bản của Git và GitHub. Tôi đặc biệt khuyến khích tất cả các bạn đọc thêm các bài viết liên quan về Git và GitHub. Tôi hy vọng bài viết này đã giúp ích cho bạn.

Theo [Freecodecamp](#) – Được dịch và biên soạn bởi đội ngũ nội dung của [Nordic Coder](#)

BÀI VIẾT NỔI BẬT



Name Cheap là gì? Có nên mua tên miền của Name Cheap?



Top những khoá học về business analysis

28/01/2021



Chứng chỉ Certified Business Analysis Professional là gì

27/01/2021



Lương của business analyst hiện nay

26/01/2021



Sài Gòn

Hết giãn cách, cấp sách tới trường

Ưu đãi lên đến **5.000.000** đồng

Áp dụng cho tất cả các khóa **Offline** và **Online**

Thời gian đăng ký: **15/10/2021** đến hết **31/10/2021**

ĐĂNG KÝ NGAY





Hết giãn cách, cấp sách tới trường

Ưu đãi lên đến **5.000.000** đồng

Thời gian đăng ký: 15/10/2021 đến hết 31/10/2021

Áp dụng cho tất cả các khóa **Offline** và **Online**

XEM NGAY



Đăng ký để được nhận ngay những tin tức độc quyền về sự kiện hấp dẫn và chương trình khuyến mãi sắp diễn ra!

Email

Gửi

0 |

TRUNG TÂM ĐÀO TẠO
LẬP TRÌNH VIÊN QUỐC TẾ

 Lầu 22, tòa nhà TNR, 180-192 Nguyễn Công Trứ, Quận 1, TP. HCM, Việt Nam.

 028 668 668 60 - 0916 051 771

 admission@nordiccoder.com

 business@nordiccoder.com

Nordic Coder

Giới thiệu



[Liên hệ](#)

Đào tạo

[Blog](#)

[Đội ngũ giảng viên](#)

[Quy trình tuyển sinh](#)

Chính sách

[Chính sách bảo mật](#)

[Điều khoản sử dụng](#)

[Chính sách bảo lưu và bồi hoàn](#)



Nordic Coder là một công ty thành viên của tập đoàn **fram^**

© Nordic Coder 2021. Đã đăng ký Bản quyền.

