**BÀI 2: TÌM HIỂU VỀ GITHUB**

## Mục tiêu

* Tìm hiểu về Github
* Cách sử dụng Github cơ bản

## Nội dung

### Tìm hiểu về Github

**Git là gì?**

Git là một trong những Hệ thống Quản lý Phiên bản Phân tán, được phát triển để quản lí mã nguồn. 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.

Github là một dịch vụ lưu trữ trên Web dành cho các dự án có sử dụng hệ thống kiểm soát Git revision.

**Những khái niệm cơ bản trong Git**

1. **Reponsitory**

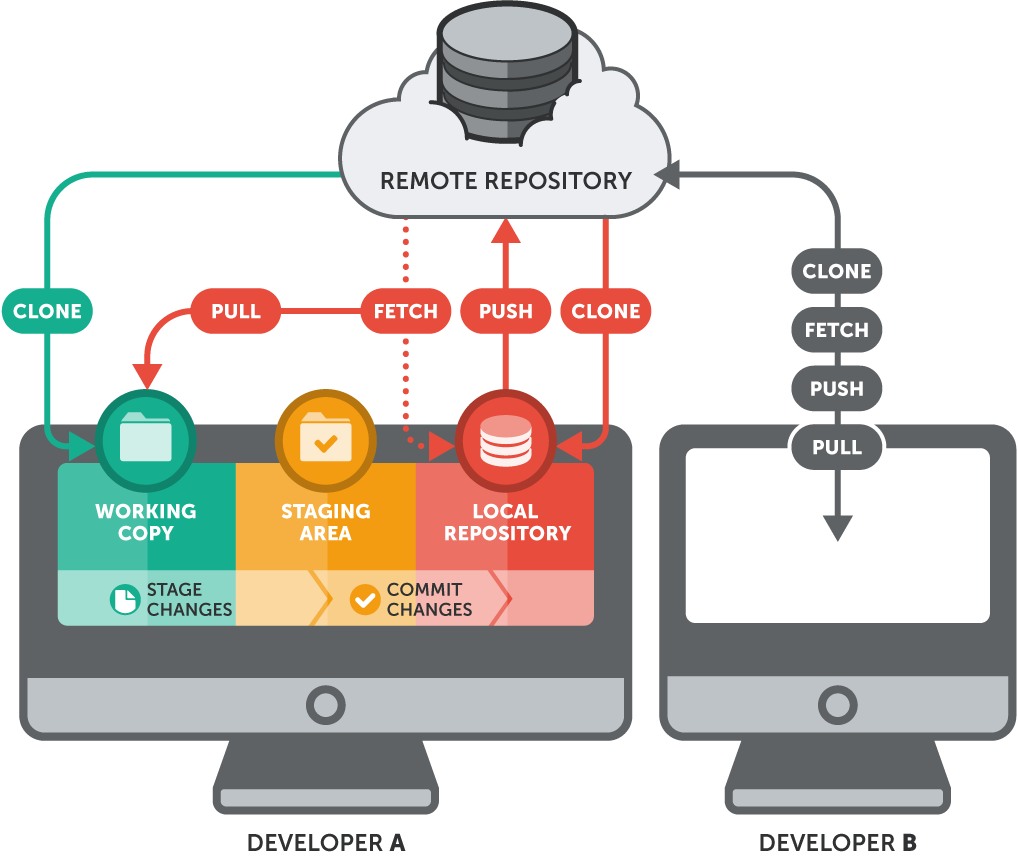
Repository hay được gọi tắt là Repo, đơn giản là nơi chứa tất cả những thông tin cần thiết để duy trì và quản lý các sửa đổi và lịch sử của toàn bộ project. Trong Repo có 2 cấu trúc dữ liệu chính là Object Store và Index. Tất cả dữ liệu của Repo đèu được chứa trong thư mục bạn đang làm việc dưới dạng folder ẩn có tên là .git

**Remote repository và local repository**

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. Khi muốn public nội dung công việc mà mình đã làm trên local repository, thì ta sẽ upload lên remote repository rồi public. Thêm nữa, thông qua remote repository bạn cũng có thể lấy về nội dung thay đổi của người khác.



1. **Nhánh (Branch)**

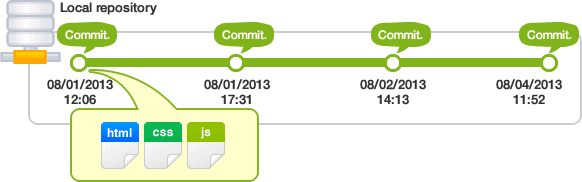
Một trong những thế mạnh của Git là nhánh. Với git, việc quản lý nhánh rất dễ dàng. Mỗi nhánh trong Git gần giống như một workspace. Việc nhảy vào một nhánh để làm việc trong đó tương tự việc chuyển qua ngữ cảnh làm việc mới, và sau đó có thể nhanh chóng quay lại ngữ cảnh cũ.

Nhánh (branch) được dùng để phát triển tính năng mới mà không làm ảnh hưởng đến code hiện tại.

Nhánh master là nhánh “mặc định” khi bạn tạo một repository. Nhánh master thông thường là nhánh chính của ứng dụng. Ví dụ bạn thử nghiệm một tính năng mới và muốn không ảnh hưởng đến code chính bạn có thể tạo một nhánh mới và sau khi xong sẽ hợp nhất lại với nhánh master. Việc hợp nhất 2 nhánh lại được gọi là merge.

1. **Commit**

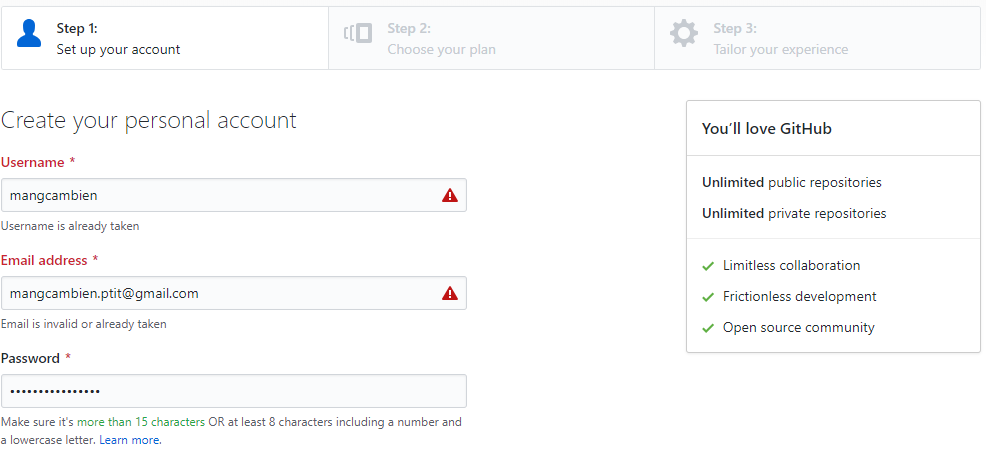
Để ghi lại việc thêm/thay đổi file hay thư mục vào repository thì sẽ thực hiện thao tác gọi là Commit. Khi thực hiện commit, trong repository sẽ tạo ra commit (hoặc revision) đã ghi lại sự khác biệt từ trạng thái đã commit lần trước với trạng thái hiện tại.



### Sử dụng Github cơ bản

**Tạo tài khoản**

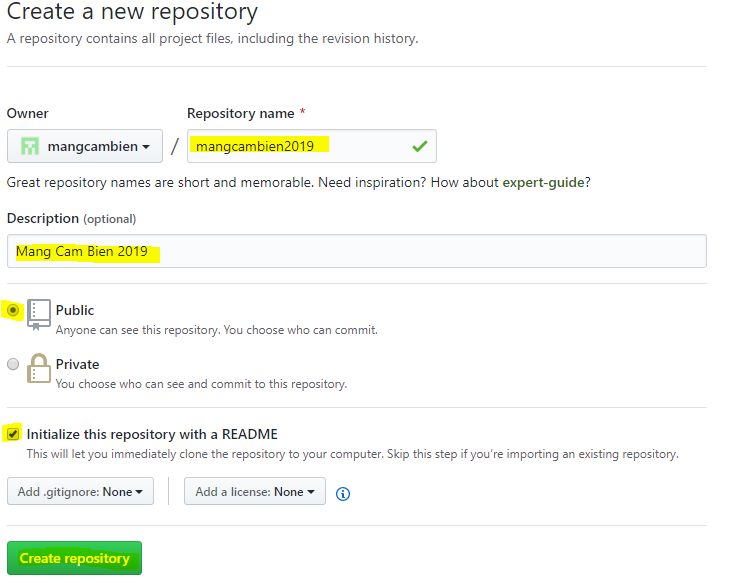
Truy cập vào đường dẫn <https://github.com/join?source=login> , tiến hành điền đầy đủ các thông tin để tạo một tài khoản Github

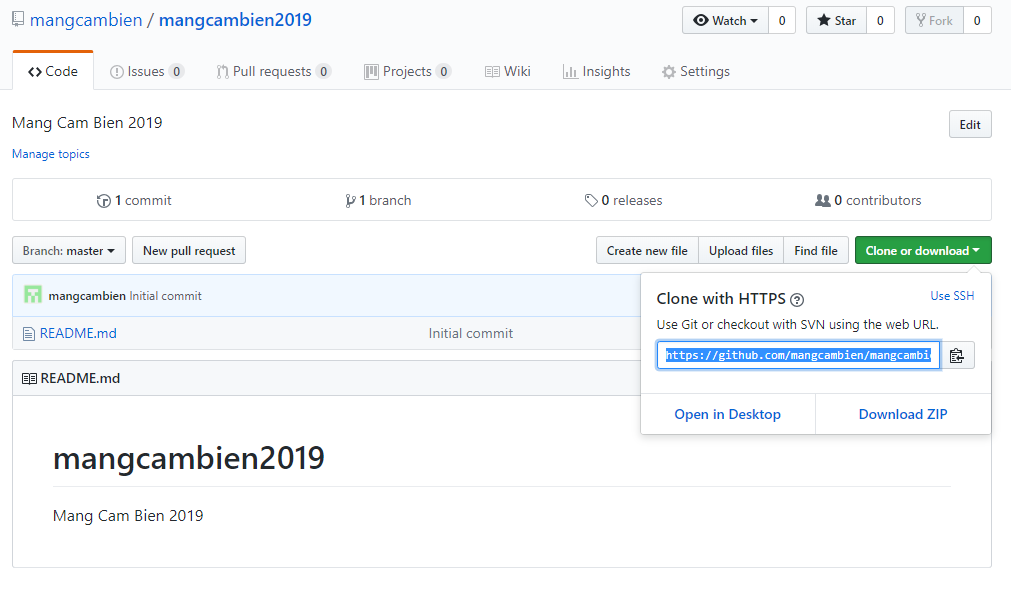


**Cài đặt Git trên Linux**

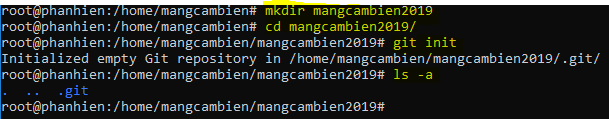
$ sudo apt-get install git

1. **Tạo một reponsitory mới**





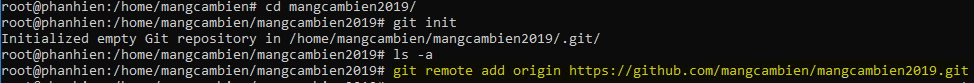
Trên Terminal sử dụng lệnh git init để khởi tạo



**(Kết nối với máy chủ Reponsitory) Add origin**

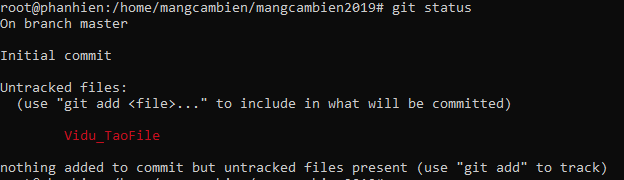
$ git remote add origin https://.....

(Đường dẫn lấy được ở bước tạo Reponsitory trên Github)



1. **Kiểm tra trạng thái (git status)**

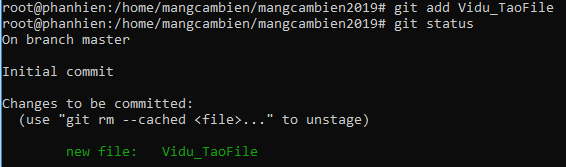
Cho biết trạng thái hiện tại. Ví dụ khi tạo một file mới có tên là Vidu\_TaoFile sau đó sử dụng git status sẽ cho thông báo như sau.



Ở đây có thông báo file Vidu\_TaoFile vừa được tạo ra nhưng chưa được Track

1. **Add và Commit**

$ git add Ten\_File



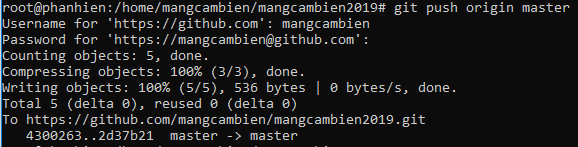
Sau khi add, file sẽ được Track

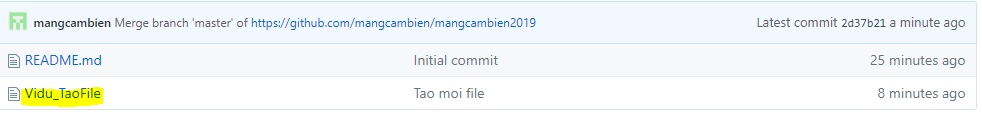
$ git commit –m “command”

Commit có tác dụng ghi chú lại những thay đổi

1. **Đưa file lên Github (git push)**

$ git push origin master

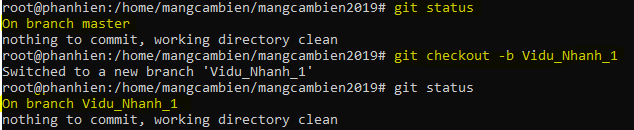




1. **Phân nhánh (Branch)**

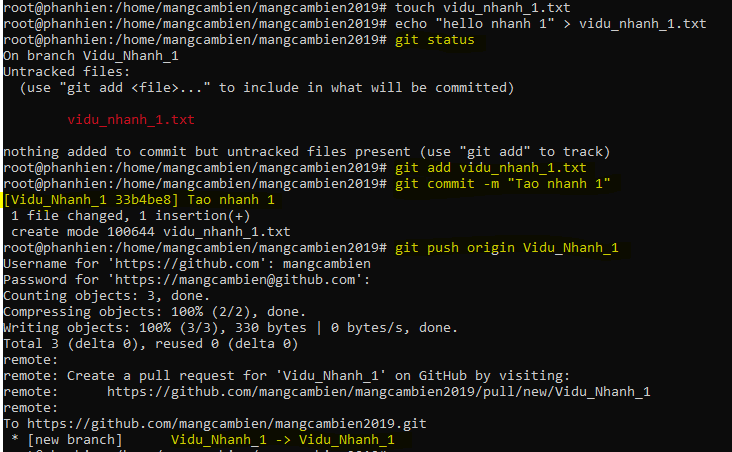
* Tạo một nhánh mới có tên là nhánh 1 rồi chuyển qua nhánh đó

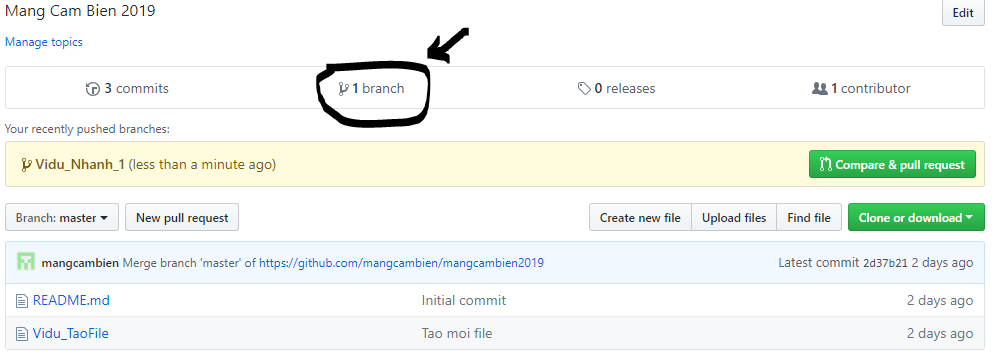
$ git checkout -b Tên\_Nhánh



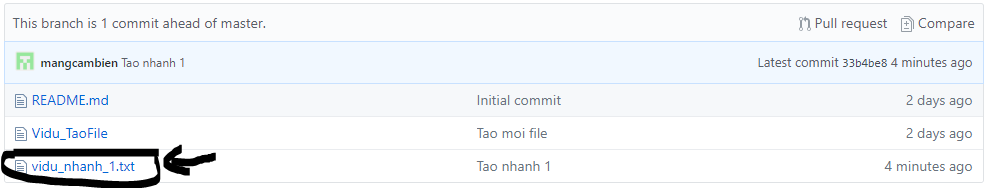
* Đưa lên Github

$ git push orgin tên\_nhánh









* Trở lại nhánh master

$ git checkout master

* Xóa một nhánh

$ git branch -D Tên\_Nhánh

1. **Git clone**

Giúp tải về mã nguồn từ Github

$ git clone htpps://github.com/mangcambien/mangcambien2019.git

## Câu hỏi và bài tập

* Tự tạo một tài khoản Github của mình
* Sử dụng Git để upload thư mục /var/www/html lên Github của mình