Để có thể sử dụng được Git, bạn phải cài ứng dụng Git vào máy tính để có thể sử dụng các dòng lệnh của Git vì toàn bộ quy trình làm việc với Git đều diễn ra các dòng lệnh (mặc dù ở Windows có một số phần mềm sử dụng Git với GUI nhưng mình không đề cập ở serie này).

### **Cài Git vào Linux**

Nếu bạn đang sử dụng hệ điều hành Ubuntu/Debian thì có thể sử dụng lệnh sau để cài Git.

**$** sudo apt-get install git

Hoặc lệnh sau để cài trên CentOS/Fedora/RHEL.

**$** yum install git

### **Cài Git vào Mac OS**

Đối với Mac, bạn có thể sử dụng file installer tải tại địa chỉ [http://git-scm.com/download/mac](https://git-scm.com/download/mac) để cài đặt.

### **Cài Git vào Windows**

Nếu bạn dùng Windows thì có thể tải file .exe cài đặt Git tại địa chỉ [http://git-scm.com/download/win](https://git-scm.com/download/win). Khi cài bạn có thể để nguyên tùy chọn mặc định mà không cần tùy chỉnh gì thêm nếu bạn chưa hiểu về nó.

Sau khi cài đặt Git vào Windows, bạn sẽ cần mở ứng dụng **Git Bash** lên để bắt đầu sử dụng các dòng lệnh của Git.

### **Thiết lập chứng thực cá nhân**

Sau khi cài Git xong, việc đầu tiên bạn nên làm là khai báo tên và địa chỉ email vào trong file cấu hình của Git trên máy. Để làm điều này bạn sẽ cần sử dụng hai lệnh sau đây để thiết lập tên và email.

**$** git config --global user.name "Thach Pham"  
**$** git config --global user.email "contact@thachpham.com"

Sau khi thiết lập xong, bạn có thể kiểm tra thông tin chứng thực trên user của bạn bằng cách xem tập tin ~/.gitconfig (nhắc lại rằng dấu ~ nghĩa là thư mục gốc của user).

**$** cat ~/.gitconfig  
[user]  
 name = Thach Pham  
 email = contact@thachpham.com

Hoặc bạn cũng có thể dùng lệnh git config --list để ghi danh sách các thiết lập hiện tại mà bạn đã làm.

Như vậy là bạn đã xong bước ban đầu đó là cài đặt Git và thiết lập tên và email của mình vào Git để bắt đầu làm việc. Ở phần sau, mình sẽ hướng dẫn cách bạn tạo ra một local repository (kho chứa trên máy cá nhân) để xem Git hoạt động thế nào.

Phần 2: Cách sử dụng git cơ bản

**Repository** (kho chứa) nghĩa là nơi mà bạn sẽ lưu trữ mã nguồn và một người khác có thể sao chép (clone) lại mã nguồn đó nhằm làm việc. Repository có hai loại là *Local Repository* (Kho chứa trên máy cá nhân) và Remote *Repository* (Kho chứa trên một máy chủ từ xa).

Trong bài này, mình sẽ hướng dẫn bạn cách tạo local repository và remote repository (sử dụng Github) và làm việc với nó.

### **Tạo local repository**

Trước hết, để tạo một repository thì bạn cần truy cập vào thư mục của mã nguồn với lệnh cd, sau đó sử dụng lệnh git init để khởi tạo repository trong thư mục đó. Ở ví dụ này, mình sẽ tạo tạo ra một thư mục mới để chứa code sau này và khởi tạo repository cho nó, mình sẽ dùng lệnh git init tên\_folder để nó tự khởi tạo thư mục.

**$** git init git\_example  
Initialized empty Git repository in /home/thachpham/git\_example/.git/

Ở đoạn trên, nó hiển thị dòng thông báo mình đã khởi tạo một kho Git trống tại đường dẫn như trên. Lưu ý rằng thư mục ẩn .git/ là nơi nó sẽ chứa các thiết lập về Git cũng như lưu lại toàn bộ thông tin về kho chứa, bạn **không cần đụng chạm gì vào thư mục** .git/ này.

Nếu kho chứa của bạn đã có sẵn mã nguồn thì bạn cần **phải đưa các tập tin về trạng thái Tracked** nhằm có thể làm việc được với Git. Để làm việc này, bạn sẽ cần sử dụng lệnh git add tên\_file, có thể sử dụng dấu \* để gom toàn bộ. Sau đó có thể sử dụng lệnh git status để xem danh sách các tập tin đã được tracked.

**$** git add readme.txt  
**$** git status  
On branch master

Initial commit

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

new file: readme.txt

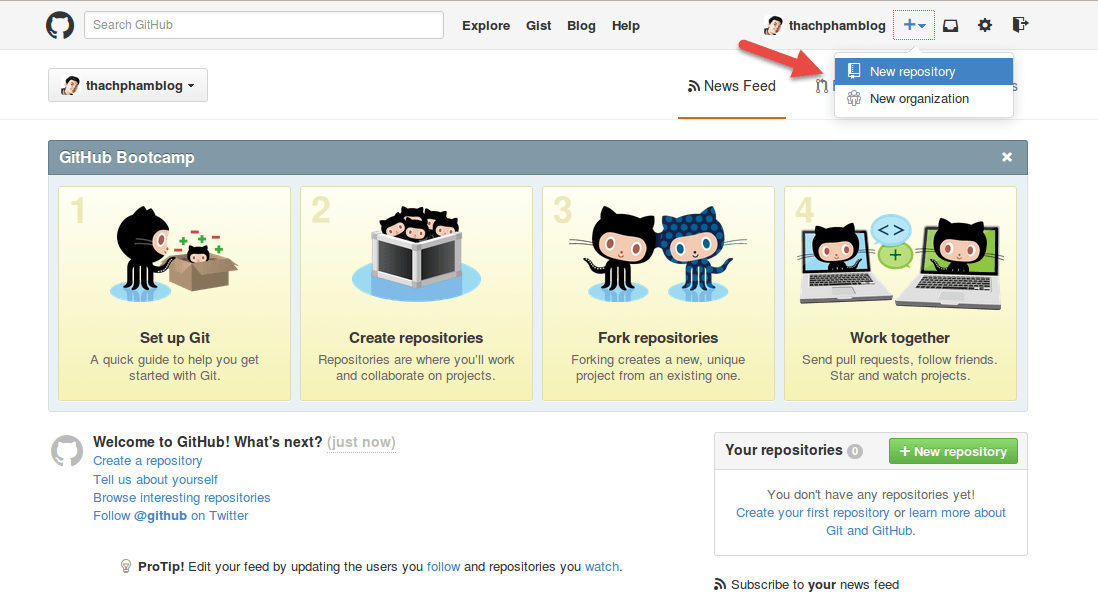
Và sau khi tập tin đã được đưa vào trạng thái tracked và nếu một tập tin đã tracked thì nó phải được đưa vào lại Staging Area (giải thích ở bài sau) cũng bằng lệnh git add thì bạn mới có thể tiến hành ủy thác (**commit**) nhằm lưu lại bản chụp các thay đổi. Lệnh commit sẽ có cấu trúc git commit -m "Lời nhắn", lúc này tất cả các tập đang trong trạng thái tracked (file mới) hoặc một tập tin đã được tracked nhưng có một sự thay đổi mới thì sẽ được commit.

$ git commit -m "First Commit"  
[master (root-commit) 799db56] First Commit  
 1 file changed, 0 insertions(+), 0 deletions(-)  
 create mode 100644 readme.txt

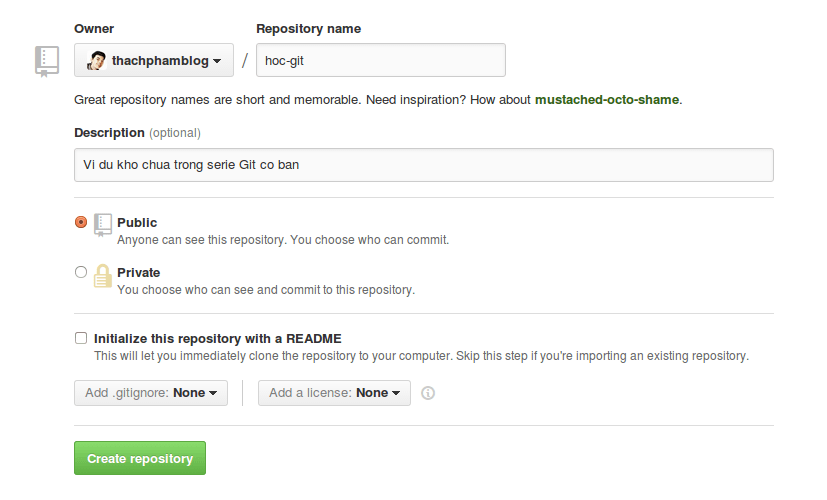
Bây giờ thì bạn đã hoàn thành việc commit lần đầu tiên các tập tin mà bạn đã đưa vào kho, mình sẽ nói kỹ hơn về việc commit ở các bài sau. Tóm lại là tới đây bạn đã có một kho chứa mã nguồn Git trên máy của bạn.

### **Tạo repository trên Github và làm việc**

Trước tiên bạn cần đăng nhập vào [Github](https://github.com/), sau đó ấn vào dấu + trên menu và chọn *New repository*.



Bạn sẽ cần đặt tên cho kho chứa của bạn. Bạn có thể chọn loại kho chứa là **Public** (ai cũng có thể clone) và **Private** (chỉ có những người được cấp quyền mới có thể clone).



*Điền thông tin khi tạo Repository trên Github*

Khi tạo xong nó sẽ dẫn bạn tới trang hướng dẫn làm việc với kho chứa vừa tạo. Và kho chứa của bạn bây giờ sẽ có địa chỉ là https://github.com/$user-name/$repository, ví dụ https://github.com/thachphamblog/hoc-git.

Việc của bạn bây giờ là hãy clone cái kho chứa này về máy của mình bằng lệnh git clone địa\_chỉ.

**$** git clone https://github.com/thachphamblog/hoc-git  
Cloning into 'hoc-git'...  
warning: You appear to have cloned an empty repository.  
Checking connectivity... done

Bây giờ hãy truy cập vào thư mục working tree (thư mục vừa clone repository về) và thử tạo ra một file tên là *README.md*, sau đó dùng lệnh git add để đưa file này vào Staging Area.

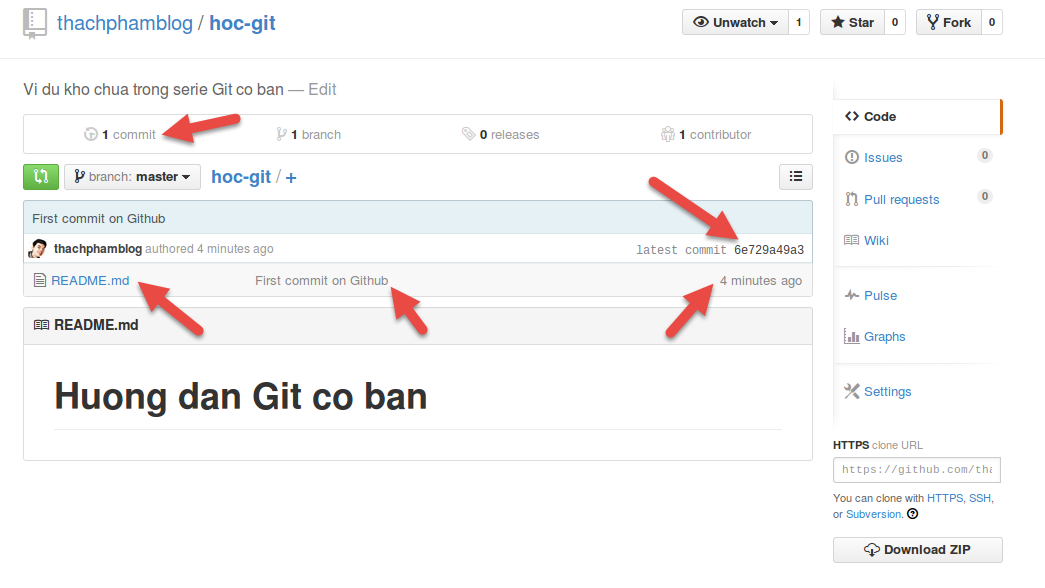
**$** cd hoc-git  
**$** echo "# Huong dan Git co ban" > README.md  
**$** git add README.md  
**$** git commit -m "First commit on Github"  
[master (root-commit) 6e729a4] First commit on Github  
 1 file changed, 1 insertion(+)  
 create mode 100644 README.md

Tuy nhiên sau khi commit xong, tập tin đã được commit sẽ vẫn không thể xuất hiện trong kho chứa trên Github mà bạn phải làm thêm một việc nữa đó là dùng lệnh git push để đẩy các tập tin đã được commit lên Github. Lưu ý rằng **bạn sẽ cần nhập tài khoản và mật khẩu Github**.

**$** git push origin master

Counting objects: 3, done.  
Writing objects: 100% (3/3), 244 bytes | 0 bytes/s, done.  
Total 3 (delta 0), reused 0 (delta 0)  
To https://github.com/thachphamblog/hoc-git  
 \* [new branch] master -> master

origin nghĩa là tên remote (xem ở bài sau) và master là tên branch, hai cái này mình sẽ giải thích kỹ hơn ở bài riêng của nó. Bây giờ bạn có thể kiểm tra kho chứa của bạn trên Github rồi đó.



*Kết quả sau khi push mã nguồn đã được commit lên Github*

Có thể bạn sẽ thấy mỗi khi push lên Github thì nó hiển thị ra thông báo quá dài dòng, bạn có thể cho ẩn các thông báo đi mà chỉ hiển thị dòng khai báo username và password trên Github thì có thể thiết lập với lệnh này.

git config --global push.default simple

Tuy nhiên nếu bạn không thích bị hỏi mật khẩu nữa thì có thể sử dụng SSH với Github, mình sẽ nói cái này sau.

### **Lời kết**

Bây giờ bạn đã biết cách tạo một repository cho Git rồi chứ? Thật đơn giản phải không nào, bây giờ để tiếp tục các bài sau dễ dàng hơn, hãy thực hành tạo một reposity trên máy tính và tập commit, hoặc bạn có thể sử dụng Github để tập đưa mã nguồn của mình vào kho chứa cho thuần thục nhé.