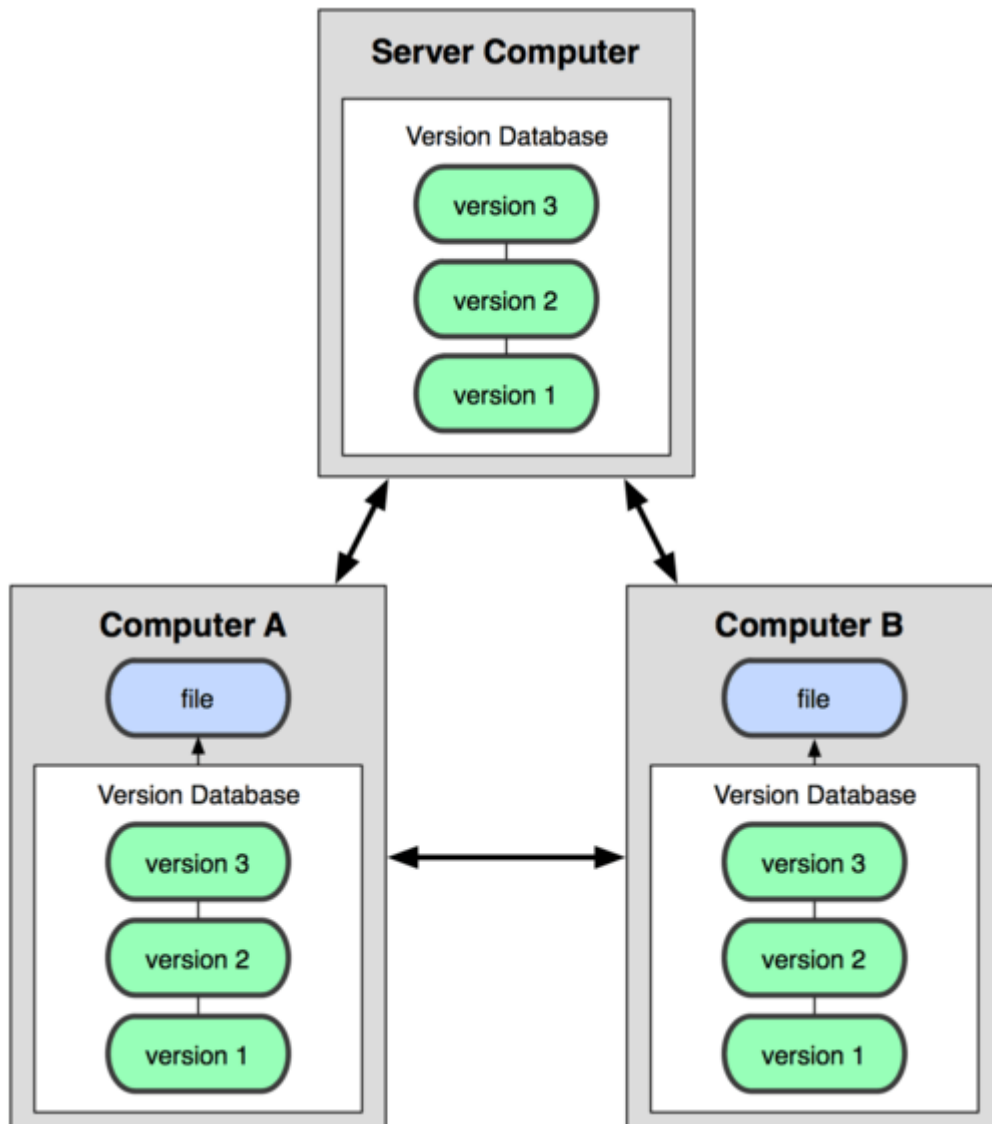


THỰC HÀNH: Git - Github

A. Lý thuyết

Git là tên gọi của một **Hệ thống quản lý phiên bản phân tán** (*Distributed Version Control System – DVCS*) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (**clone**) từ một kho chứa mã nguồn (**repository**), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (**commit**) rồi đưa lên máy chủ nơi đặt repository chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ repository hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là **Working Tree**.



Mô hình hoạt động của DVCS

Ngoài ra, có một cách hiểu khác về Git đơn giản hơn đó là nó sẽ giúp bạn lưu lại các phiên bản của những lần thay đổi vào mã nguồn và có thể dễ dàng khôi phục lại dễ dàng mà không cần copy lại mã nguồn rồi cất vào đâu đó. Và một người khác có thể xem các thay đổi của bạn ở từng phiên bản, họ cũng có thể đối chiếu các thay đổi của bạn rồi gộp phiên bản của bạn vào phiên bản của họ. Cuối cùng là tất cả có thể đưa các thay đổi vào mã nguồn của mình lên một repository mã nguồn.

Cơ chế lưu trữ phiên bản của Git là nó sẽ tạo ra một “ảnh chụp” (*snapshot*) trên mỗi tập tin và thư mục sau khi commit, từ đó nó có thể cho phép bạn tái sử dụng lại một ảnh chụp nào đó mà bạn có thể hiểu đó là một phiên bản. Đây cũng chính là lợi thế của Git so với các DVCS khác khi nó không “lưu cứng” dữ liệu mà sẽ lưu với dạng snapshot.

Git là tên gọi của một mô hình hệ thống. Các máy tính có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các repository của riêng mình để có thể làm việc.

Cài Đặt Git Trên Windows

Cài đặt Git trên Windows rất đơn giản. Dự án msysGit cung cấp một cách cài đặt Git dễ dàng hơn. Đơn giản chỉ tải về tập tin cài đặt định dạng exe từ Github, và chạy:

```
http://git-for-windows.github.io
```

Cấu Hình Git Lần Đầu

Git cung cấp sẵn git config cho phép xem hoặc chỉnh sửa các biến cấu hình để quản lý toàn bộ các khía cạnh của Git như giao diện hay hoạt động. Các biến này có thể được lưu ở ba vị trí khác nhau:

- `/etc/gitconfig` : Chứa giá trị cho tất cả người dùng và repository trên hệ thống.

Sử dụng `--system` khi chạy `git config`, thao tác đọc và ghi sẽ được thực hiện trên tập tin này.

- `~/.gitconfig` : Riêng biệt cho tài khoản của bạn. Có thể chỉ định Git đọc và ghi trên tập tin này bằng cách sử dụng `--global`.

Trên Windows, Git sử dụng tập tin `.gitconfig` trong thư mục `$HOME` (`%USERPROFILE%` trên môi trường Windows), cụ thể

là `C:\Documents and Settings\%USER%` hoặc `C:\Users\%USER%`, tùy thuộc vào phiên bản Windows đang sử dụng (`%USER%` là `%USERNAME%` trên môi trường Windows).

Thiết lập chứng thực cá nhân

Việc đầu tiên nên làm khi cấu hình Git là chỉ định tên tài khoản và địa chỉ e-mail. Điều này rất quan trọng vì mỗi Git sẽ sử dụng chúng cho mỗi lần commit, những thông tin này được gắn vào các commit:

```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email johndoe@example.com
```

Nếu muốn sử dụng tên và địa chỉ e-mail khác cho một dự án riêng biệt nào đó, có thể chạy lại lệnh trên không sử dụng `--global` trên dự án đó.

Repository (kho chứa) là nơi 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).

Tạo local repository

Để tạo một repository 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.

Ví dụ tạo một thư mục mới để chứa code sau này và khởi tạo repository, dùng lệnh `git init` tên_folder để 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 đã khởi tạo một kho Git trống tại đường dẫn như trên.

Nếu repository đã có sẵn mã nguồn thì cần **phải đưa các tập tin về trạng thái Tracked** nhằm có thể làm việc được với Git. Để làm việc này, 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 cũng bằng lệnh `git add` thì 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"
```

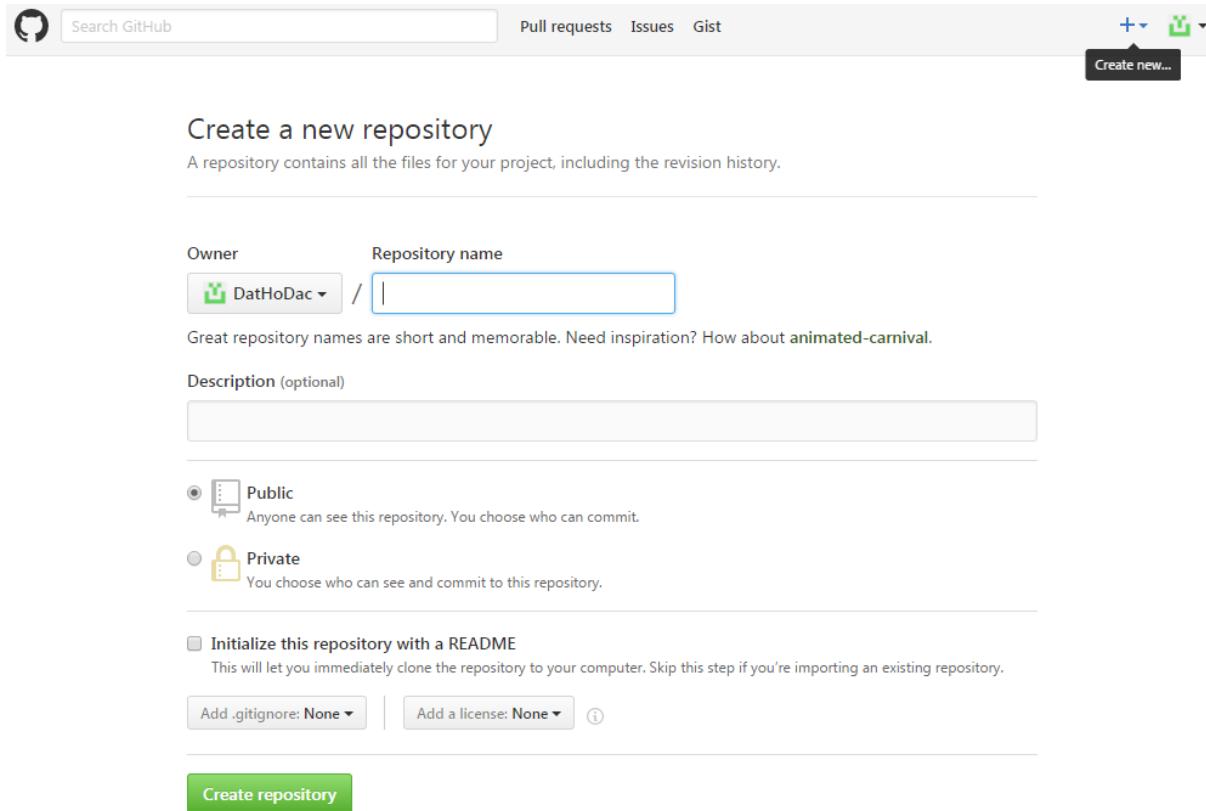
```
[master (root-commit) 799db56] First Commit
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
create mode 100644 readme.txt
```

Tạo repository trên Github

Trước tiên cần đăng nhập vào [Github](#), sau đó ấn vào dấu + trên menu và chọn Create new repository. Đặt tên cho repository. Có thể chọn loại repository là **Public** (ai cũng có thể clone) và **Private** (chỉ có những người được cấp quyền mới có thể clone).



Search GitHub Pull requests Issues Gist Create new...

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: DatHoDac / Repository name:

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

Description (optional):

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Khi tạo xong nó sẽ dẫn tới trang hướng dẫn làm việc với repository vừa tạo. Và repository bây giờ sẽ có địa chỉ là [https://github.com/\\$user-name/\\$repository](https://github.com/$user-name/$repository), ví dụ <https://github.com/DatHodac/hoc-git>.

Tiếp theo, clone repository này về máy của mình bằng lệnh `git clone địa_chỉ`.

```
$ git clone https://github.com/dathodac/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 repository trên Github do đó cần dùng lệnh `git push` để đẩy các tập tin đã được commit lên 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/dthodac/hoc-git

* [new branch] master -> master
```

B. Thực hành

Thực hiện các thao tác sau:

- 1 Cài đặt git trên windows
- 2 Tạo tài khoản trên Github.
- 3 Tạo thư mục repository local trên máy
- 4 Tạo thư mục repository trên server.
- 5 Dùng lệnh git clone <link> để sao chép repository về máy.
- 6 Dùng lệnh cd để di chuyển vào thư mục ThucHanhGit.
- 7 Dùng lệnh git add . để thêm tất cả các file mới vào repository.
- 8 Dùng lệnh git commit -m “...” để commit thay đổi vào repository.
- 9 Dùng lệnh git tag Tag1 để đánh dấu tag theo yêu cầu.
- 10 Dùng lệnh git push origin master để đẩy cập nhật branch master.
- 11 Dùng lệnh git push origin Tag1 để đẩy tag gửi nội dung lên github.
- 12 Dùng lệnh git rm và git rm -r “...” để xóa một vài file trong thư mục.
- 13 Xem danh sách các trạng thái thay đổi của thư mục (log).
- 14 Tạo project trên Github.
- 15 Thử các câu lệnh: Git fetch, Git stash, VIM, vi vimrc, git revert, git cherry-pick