BÀI TẬP THỰC HÀNH
MÔN HỌC: LINUX VÀ PMMNM
CHƯƠNG 1: MỞ ĐẦU

HỌ TÊN SV: MSSV:
MÃ LỚP: MÃ HỌC PHẦN:

Bài TH 1: CÀI ĐẶT UBUNTU

Ở bài thực hành này các bạn sẽ tiền hành cài đặt bản phân phối Ubuntu.

Bước 1: Tải về file iso của bản phân phối mới nhất tại link:
https://www.ubuntu.com/download/desktop

Bước 2:
Bây giờ bạn có một trong 3 sự lựa chọn:

1. Cài trực tiếp máy thật
2. Cài máy ảo
3. Cài và sử dụng Ubuntu thông qua Docker

Nếu bạn chọn cài trực tiếp máy thật:
Ghi một bản cài đặt lên một USB có dung lượng ít nhất 2GB.
Bạn cài đặt ứng dụng Rufus để ghi tại link sau: https://rufus.akeo.ie/
Sau đó mở ứng dụng Rufus và chọn ghi lên thiết bị USB từ file iso bạn vừa tải được ở Bước 1.
Tiếp theo, mở chương trình BIOS, chọn khởi động từ USB, cắm USB vào và ấn khởi động lại máy.
Tiếp tục làm theo các bước hướng dẫn để cài đặt.

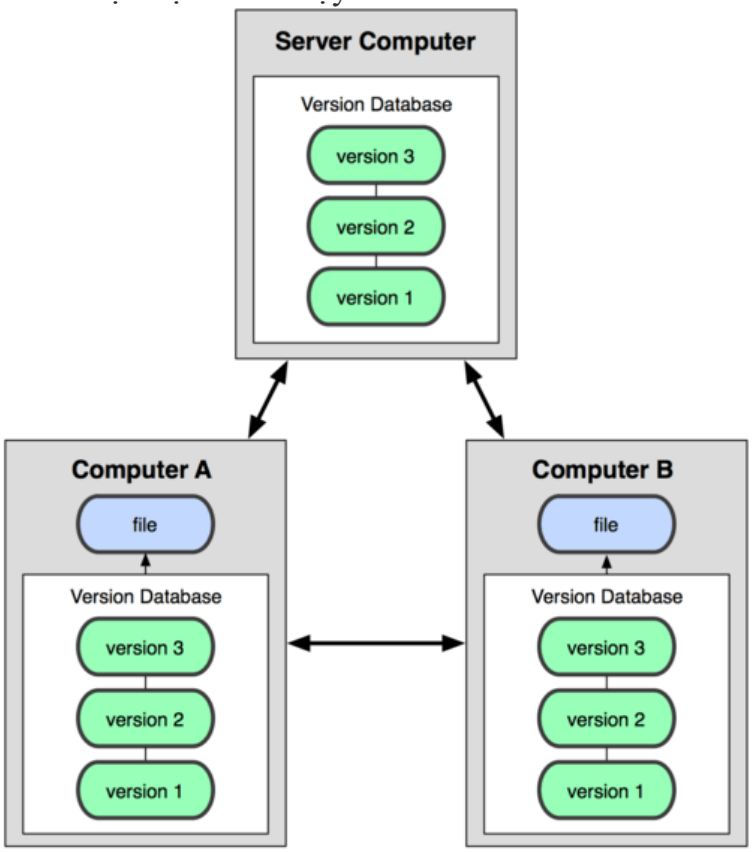
Nếu chọn cài đặt bằng máy ảo:
Đầu tiên bạn phải tải chương trình VirtualBox tại link:
https://www.virtualbox.org/wiki/Downloads

Sau đó chạy chương trình VirtualBox, ấn chọn New để tạo 1 mảy ảo mới. Sau đó chọn cài đặt từ file iso vừa tải được ở Bước 1.

Nếu chọn sử dụng Ubuntu thông qua Docker:
Cách này yêu cầu bạn phải có chút khái niệm về sử dụng Docker.
Link tải Docker: https://store.docker.com/editions/community/docker-ce-desktop-windows
Sau khi đã cài đặt Docker, bạn chạy ubuntu bằng lệnh: run –it ubuntu
Nếu chưa tải về thì máy sẽ tải image ubuntu về sau đó chạy như bình thường.

Bài TH 2: Git và GitHub

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 kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa 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**. Đại loại là như vậy.

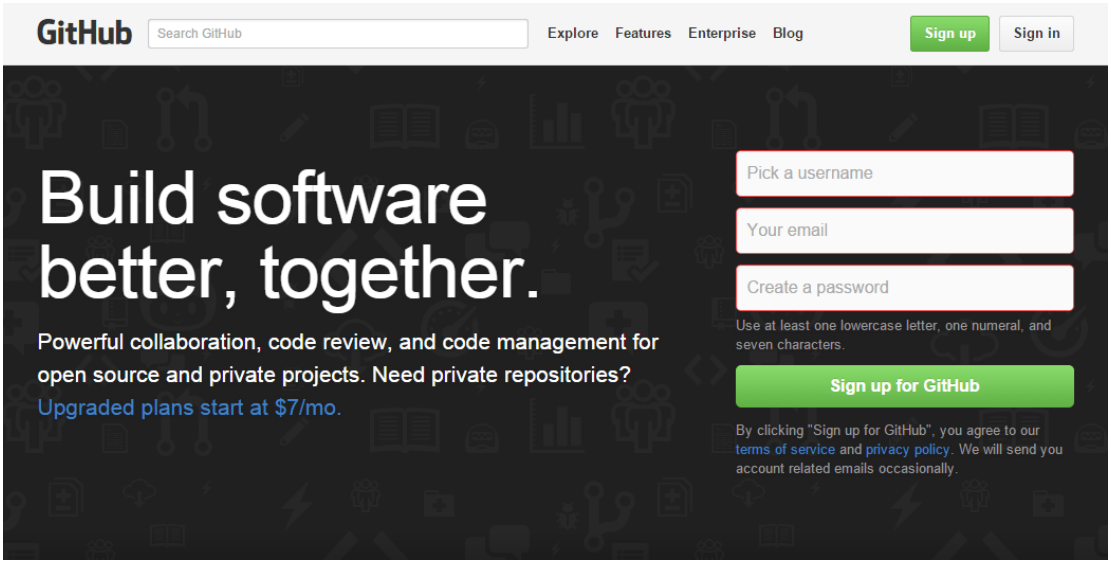


Ngoài ra, có một cách hiểu khác về Git đơn giản hơn đó là nó sẽ giúp bạn lưu lại các phiên bản của những lần thay đổi vào mã nguồn và có thể dễ dàng khôi phục lại dễ dàng mà không cần copy lại mã nguồn rồi cất vào đâu đó. Và một người khác có thể xem các thay đổi của bạn ở từng phiên bản, họ cũng có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

Github là gì?



Có rất nhiều bạn khi nghe nói đến Git sẽ nghĩ ngay đến Github và có thể sẽ có một số hiểu lầm với họ. Cũng xin nhắc lại rằng, Git là tên gọi của một mô hình hệ thống. Như mình đã giải thích ở trên, **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 kho chứa của riêng mình để có thể làm việc.
Mặc dù Git có thể làm việc với bất kỳ trên máy chủ Linux nào nhưng để dễ hiểu và thực tế hơn, bài thực hành này sẽ hướng dẫn các bạn các thao tác cơ bản nhất của Git và Github.

Tại sao nên sử dụng Git?

Có rất nhiều lợi thế để bạn nên sử dụng Git trong việc lập trình ngay từ hôm nay, bất kể là lập trình cái gì đi chăng nữa.

- Git dễ sử dụng, an toàn và nhanh chóng.
- Có thể giúp quy trình làm việc code theo nhóm đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch).
- Bạn có thể làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ kho chứa hoặc clone một phiên bản thay đổi nào đó từ kho chứa, hoặc một nhánh nào đó từ kho chứa.
- Dễ dàng trong việc deployment sản phẩm.
- Và nhiều hơn thế nữa.

Nếu bạn là một lập trình viên thì Git là một hệ thống bạn cần phải biết cách sử dụng, ít nhất là ngay từ bây giờ.

Cài đặt GIT trên máy bằng lệnh sau

```
$ sudo apt-get install 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 "Tran Anh"
$ git config --global user.email "contact@gmail.com"
```

Sau khi thiết lập xong, hãy xem nội dung file ~/.gitconfig bằng lệnh `cat ~/.gitconfig` ; Em thấy nội dung gì?

Trả lời:

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

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).

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, chúng ta sẽ tạo tạo ra một thư mục mới để chứa code sau này và khởi tạo repository cho nó, chúng ta sẽ dùng lệnh git init tên_folder để nó tự khởi tạo thư mục.

```
$ git init git_example
Initialized empty Git repository in /home/trananh/git_example/.gi
t/
```

Ở đoạn trên, nó hiển thị dòng thông báo 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
```

Em thấy hiện nội dung gì ra màn hình?

Trả lời:

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 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, chúng ta 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, 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).

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/trananh/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/trananh/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
```

Sau khi gõ những lệnh trên, hãy thử kiểm tra xem tập tin đã có được trên Github chưa? Vì sao?

Trả lời:

Bây giờ chúng ta 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/trananh/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.