**Hướng dẫn sử dụng git – Các lệnh cơ bản của GIT**

[**https://www.hostinger.vn/huong-dan/huong-dan-su-dung-git-lenh-git-co-ban**](https://www.hostinger.vn/huong-dan/huong-dan-su-dung-git-lenh-git-co-ban)

**1.Git là gì? Giới thiệu về Git.**

Git : là một hệ thống giúp cho chúng ta kiểm soát được các phiên bản (hệ thống VCS: version control system)

Quản lí các phiên bản là gì?

* Đó là lịch sử của một tập tin, một dự án, cái gì đã thực hiện, ai đã làm, làm khi nào, tại sao lại làm vậy…🡺Chúng ta có thể quản lí lịch sử của project đó, và có thể khôi phục lại

Nguyên tắc làm việc của Git:

* Có một kho lưu trữ trung tâm ( repository )
* Lấy dữ liệu về, thực hiện các công việc, cập nhật dữ liệu lên kho trung tâm
* Công cụ để giải quyết các xung đột

Lợi ích:

* Đảm bảo không làm hỏng bất cứ cái gì( chúng ta có thể quay lại các phiên bản trước đó của project)
* Giúp chúng ta có thể cộng tác với nhiều người
* Cho phép ta can thiệt và phát triển phiên bản mới mà không ảnh hưởng đến phiên bản cũ
* Có thể quay lại phiên bản trước bất kì lúc nào
* Chia sẻ với người khác

Git != GitHub: GitHub là kho lưu trữ trực tuyến và hỗ trợ tất cả các công nghệ của git.

Cách cài git: truy cập: git-scm.com và download + mở

**2.Các câu lệnh cơ bản trong git**

**Basic terminal commands: các câu lệnh cơ bản**

* Câu lệnh cd( change directory) : thay đổi thư mục làm việc
* Cd .. : quay về thư mục cha
* Cd tên\_thư\_mục\_con : di chuyển vào thư mục con
* Câu lệnh dir (ls với mac): hiện thị tất cả các danh sách tập tin bên trong thư mục hiện thành
* Câu lệnh mkdir “foldername” : tạo thư mục mới
* Câu lệnh touch “tenfile”: tạo một tập tin mới
* Câu lệnh echo “noidung” ( > “noicanxuatvao”) : in/xuất một nội dung
* Câu lệnh cat : hiện thị nội dung
* Câu lệnh diff filename1 filename2: so sánh xem file1 và file2 khác nhau chỗ nào
* Câu lệnh rm “name” : xóa file hoặc folder rỗng
* Câu lệnh rm -rf “name” : xóa tất cả mọi thứ của tập tin hoặc folder

**Cách tạo một repository**

Repository : kho lưu trữ

Commit: một đơn vị làm việc

Branch: nhánh

Main/master: tên của repo chính (main repo)

Merge/rebase: kết hợp 2 nhánh

Develop: tên của nhánh, lập trình viên

* Câu lệnh git - -help: hiện trợ giúp, hướng dẫn
* Câu lệnh git -v: hiện phiên bản của git
* Câu lệnh git status: hiện thị trạng thái kho lưu trữ
* Câu lệnh git log: hiện thị lịch sử các commit
* Câu lệnh git init [repo name]: tạo ra một kho lưu trữ
* Git clone [repo] [clone name]: tạo một bản sao được liên kết với repo

**Cấu hình một repo:**

Ai là người thực hiện, khi nào, cái gì 🡺 phải cấu hình

* Git config -l [--scope]: xem cấu hình hiện tại
* Git config [--scope] [option-name] [value] : Cấu hình git
* Scope:

+ --system =>ảnh hưởng đến tất cả người dùng

+ --global => liên quan đến tất cả repos

+ --local => liên quan đến 1 repo

**Trạng thái của git:**

-working directory (1) 🡺 sử dụng một câu lệnh git add 🡺 sẽ đưa wd và khu vực tổ chức(2) 🡺 sử dụng câu lệnh git commit 🡺 được đưa vào repo(3) 🡺 có 3 trạng thái

- Câu lệnh git add [filenames] : them tệp vào index

- câu lệnh git add . :them tệp (tất cả)

- git commit -m “nội dung commit” :tạo commit 🡺repo

-git status :sự khác biệt giữa 3 cây

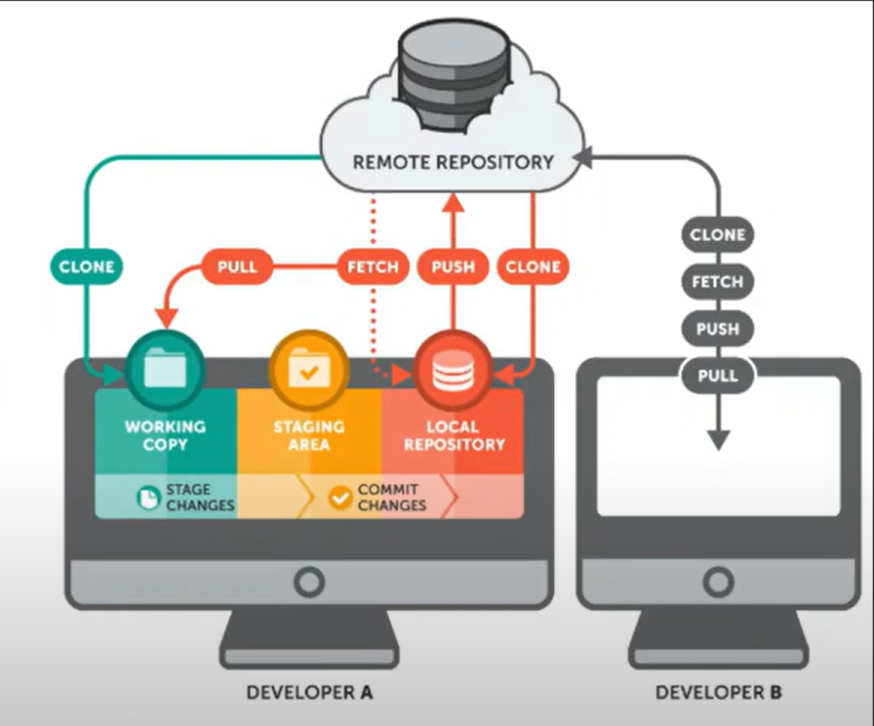
- git diff : so sánh với commit cuối cùng

- git log: để xem lại lịch sử các commit

- git log - -oneline: xem các lịch sử các commit nma ko chi tiết = log

Cấu hình .gitignore : chỉ định tập tin/ thư mục không nên theo dõi bởi git (bỏ qua thư mục đó)

**Làm việc với kho lưu trữ trung tâm (central repository / remote repository)**



* Các câu lệnh  
  + git init - -bare : tạo một central repo

+ git clone [repo name] [clone name] : sao chép và liên kết repo name

+ git fetch : lấy các thông tin về commit mới từ central

+ git pull : lấy dữ liệu từ central về local repo

+ git push: đẩy các commit từ local về central

**Xử lí lỗi conflict:**

Lỗi conflict là gì?

Lỗi **conflict** (xung đột) trong Git xảy ra khi hai hoặc nhiều thay đổi mâu thuẫn với nhau, và Git không thể tự động hợp nhất (merge) chúng. Điều này thường xảy ra khi bạn cố gắng **merge** hoặc **pull** các thay đổi từ một nhánh hoặc remote repository vào nhánh hiện tại, và có sự khác biệt giữa các tệp mà cả hai bên đã sửa đổi.

🡺**Cách khắc phục**: Tự phải sửa thôi 😊, vào file bị lỗi conflict và tự phải sửa xem nên giữ và xóa những gì.

**Chuyển đổi giữa các commit :** git checkout <commit\_hash> : dùng để chuyển đổi những nhánh hoặc giữa các commit.

**Tạo nhánh mới:**

* Git branch <tennhanh> : tạo nhánh mới
* Git checkout <tennhanh> : chuyển qua nhánh khác
* Git branch -l : xem danh sách các nhanh
* Git push -u origin <tennhanh> : tạo nhánh mới trên centralrepo.

**Kết hợp các nhánh**

* Git merge <tennhanh1>: kết hợp nhánh 1 và nhánh hiện tại.
* Git rebase <tennhanh> : tái cơ sở cho nhánh, thay đổi vị trí bắt đầu của nhánh B đi sang vị trí cuối cùng của nhánh A.
* Git rebase - -continue : tiếp tục áp dụng những gì mà ta thay đổi
* Git rebase - -skip : bỏ qua những thay đổi

**Cách xóa nhánh**

Tại sao phải xóa nhánh? Do khi ta merge 2 nhánh A và B thành 1 nhánh thì sẽ còn 1 nhánh vẫn còn tồn tại 🡺 để làm cho hệ thống repository gọn gàng thì ta xóa nhánh thừa đi.

* Git branch -d <tên nhanh>: xóa nhánh local
* Git branch -a : xem các nhanh trong centralrepo
* Git push origin -d <tên nhánh> : xóa nhánh trong centralrepo

**Cách hủy bỏ commit**

* Git reset - -soft <commit id> : di chuyển head về vị trí commit, trạng thái của stage và tất cả sự thay đổi của file được giữ nguyên
* Git reset - -hard<commit id> : di chuyển head về vị trí commit, trạng thái của stage và tất cả sự thay đổi của file được sẽ được quay lại vị trí mong muốn và các commit sau đó bị hủy bỏ.

**Cách quay lại commit trước đây nhưng không hủy bỏ commit mới**

* Git revert <commit id>: di chuyển head về vị trí commit, trạng thái của stage và tất cả sự thay đổi của file được sẽ được quay lại vị trí mong muốn và các commit ( Có thể xảy ra lỗi conflict).

**Làm việc với GitHub**

**Cách kết nối repository với github**

B1: tạo một repository mới trên github của mình

B2: vào repository của mình rồi viết câu lệnh: git remote add origin <tên link repository trên github>.

B3: git config

B4: chuyển sang nhánh chính : git branch -M main

B5: git push -u origin main

**Cách clone một repository trên github về máy tính**

* Dùng git clone <tên link repository trên github>