BÁO CÁO GIT

# I. Git là gì

Git là 1 công cụ giúp quản lý mã nguồn, quản lý phiên bản mã nguồn

# II. Các câu lệnh Git cơ bản

## Config

$ git config --global user.name "Your Name"

$ git config --global user.email youremail

## Clone code

$ git clone linkcode

## Git status

$ git status

Cho biết trạng thái các file có thay đổi / thêm mới / xóa:

* Untracked files: Các file mới trong thư mục làm việc chưa được Git theo dõi.
* Changes not staged for commit: Các thay đổi đã được thực hiện trong các file đang được theo dõi, nhưng chưa được thêm vào staging area.
* Changes to be committed: Các thay đổi đã được thêm vào staging area và sẵn sàng để commit.
* Branch status: Tên của nhánh hiện tại, số commit đã hoặc chưa được đẩy lên kho lưu trữ từ xa, và liệu có bất kỳ xung đột nào giữa nhánh cục bộ và nhánh từ xa không.

Kiểm tra các tệp bị bỏ qua

git status --ignored

## Git add

Thêm các file cụ thể vào staging area.

$ git add <file1> <file2> <file3>

Thêm tất cả các thay đổi trong thư mục hiện tại (bao gồm cả các thư mục con) vào staging area.

$ git add .

Thêm tất cả các thay đổi, bao gồm cả việc theo dõi các file mới, xóa các file đã xóa, và cập nhật các file đã thay đổi vào staging area.

$ git add -A

Thêm tất cả các thay đổi trong một thư mục cụ thể vào staging area.

$ git add <directory>

## Git commit

Ghi lại các thay đổi từ staging area vào lịch sử phiên bản của kho lưu trữ Git (.git)

$ git commit -m “Messenge”

Commit tất cả các file đã sửa đổi

git commit -a -m "Messenge"

Chỉnh sửa commit cuối cùng

git commit --amend -m "Messenge 2"

## Git push

Đẩy các commit từ local repository lên remote repository

git push <remote> <branch>

* <remote>: Tên của remote repository (thường là origin).
* <branch>: Tên của nhánh muốn đẩy lên

Đẩy tất cả các nhánh trong local repository lên remote repository.

git push --all <remote>

Đẩy tất cả các tag đã được tạo trong local repository lên remote repository.

git push <remote> --tags

Đẩy nhánh lên remote và thiết lập nó làm nhánh mặc định để theo dõi các thay đổi (tracking branch). Sau khi sử dụng -u, bạn có thể chỉ cần sử dụng git push mà không cần chỉ định nhánh trong các lần sau.

git push -u <remote> <branch>

## Git log

Xem lịch sử commit: người commit, ngày giờ, message của những lần commit đó

## Git init

Khởi tạo một kho lưu trữ Git mới trong một thư mục (.git) trong thư mục hiện tại

Git init

## Git remote add origin

Thêm một remote repository và đặt tên cho nó là origin.

git remote add origin <url>

## Git merge

Hợp nhất thay đổi giữa nhánh phụ vào nhánh chính

$ git merge branch\_name

## Git pull

Cập nhật local branch với các thay đổi mới nhất từ remote, đồng bộ hóa các thay đổi giữa remote và local.

git pull <remote> <branch>

Cách hoạt động của git pull:

* **Fetch**: thực hiện git fetch tải về các thay đổi từ remote về local. Lệnh này cập nhật kho lưu trữ cục bộ với các thay đổi mới nhất từ remote.
* **Merge**: Sau khi tải về các thay đổi, thực hiện git merge để hợp nhất các thay đổi từ remote với nhánh hiện tại. Điều này có thể dẫn đến các xung đột nếu có các thay đổi khác nhau trong cùng một phần của mã nguồn.

Áp dụng các thay đổi từ remote lên trên các thay đổi cục bộ

git pull --rebase origin master

## Git checkout

Tạo mới 1 branch

git checkout -b branch\_name

Di chuyển đến 1 branch đã có

git checkout branch\_name

## Git gitignore

Tệp .gitignore là tệp cấu hình trong Git chỉ định các tệp và thư mục không muốn Git theo dõi hoặc đưa vào kho lưu trữ, giúp loại bỏ các tệp không cần thiết hoặc các tệp tự động sinh ra từ môi trường phát triển (như các tệp log, tệp biên dịch, hoặc tệp tạm thời).

Tạo tệp .gitignore

touch .gitignore

vi .gitignore

Cấu trúc:

config.json # Bỏ qua tệp cấu hình cụ thể

logs/ # Bỏ qua thư mục

\*.log # Bỏ qua tất cả các tệp .log

build/ # Bỏ qua tất cả các tệp trong thư mục build/

\*\*/\*.tmp # Bỏ qua tất cả các tệp .tmp trong bất kỳ thư mục nào

\*.o # Bỏ qua tệp biên dịch

\*.pyc # Bỏ qua tệp biên dịch

.env # Bỏ qua các tệp .env (tệp môi trường)

## Git remove

Xóa 1 tệp khỏi remove

git rm --cached <file>

## Git stash

Lưu trữ tạm thời các thay đổi chưa được commit trong working directory và khôi phục trạng thái của working directory về trạng thái sạch

git stash

git stash save “messenge”

Xem danh sách stash, mỗi stash sẽ có một tên định danh như stash@{0}, stash@{1}

git stash list

Khôi phục thay đổi từ stash và không xóa stash đó

git stash apply

git stash apply stash@{0}

Khôi phục thay đổi và xóa stash

git stash pop

Xóa stash

git stash drop stash@{0}

git stash clear

Lưu vào stash các untracked files

git stash -u