Cài đặt git:

+ github cung cấp cho khách hang sử dụng máy tính bao gồm đồ họa giao diện người dùng cho hầu hết các hành động về lưu trữ và các câu lệnh tự động cập nhật của git của các phiên bản nâng cao.

GitHub for Windows

htps://windows.github.com

GitHub for Mac

htps://mac.github.com

Các phiên bản cho hệ thống linux và posix đã chính thức ra mắt trên git scm website.

oﬃcial Git SCM web site.

Git for All Platforms

htp://git-scm.com

Công cụ thiết lập cấu hình:

Cấu hình thong tin người dùng cho tất cả các kho lưu trữ

$ git config --global user.name "[name]"

Đặt tên bạn muốn dùng cho việc cam kết

$ git config --global user.email "[email address]"

Đặt tên email muốn dùng cho việc cam kết

$ git config --global color.ui auto

Mở chức năng thay đổi màu chữ cho các dòng lệnh khác nhau

Tạo kho lưu trữ:

Bắt đầu tạo 1 kho lưu trữ mới hoặt sử dụng từ cái đã có sẵn thong qua URL

**$ git init [project-name]**

**Tạo mới 1 kho lưu trữ với 1 tên cụ thể**

**$ git clone [url]**

**Tải xuống 1 dự án và toàn bộ các phiên bản từ trước**

**Thay đổi:**

**Xem lại và chỉnh sửa**

**$ git status**

**Liệt kê trạng thái các tập tin mới**

**$ git add [file]**

**Theo dõi các tập tin mới**

**$ git diff**

**Dùng để biết chính xác tập tin nào thay đổi so với phiên bản trước(chưa staged)**

**$ git diff --staged**

**Dùng để biết chính xác tập tin nào thay đổi so với phiên bản trước (đã staged)**

**$ git commit -m "[descriptive message]"**

**Lưu lại sự thay đổi hiện tại so với phiên bản trước**

**$ git branch**

**Liệt kê các nhánh trong kho lưu trữ hiện tại**

**$ git branch [branch-name]**

**Tạo nhánh mới**

**$ git checkout [branch-name]**

**Di chuyển sang nhánh đó**

**$ git merge [branch]**

**Đồng bộ 1 nhánh cụ thể với nhánh hiện tại**

**$ git branch -d [branch-name]**

**Xóa 1 nhánh**

**Thay đổi tên tập tin**

**Di chuyển và loại bỏ tập tin**

**$ git rm [file]**

**Xóa 1 tập tin khỏi thư mục đang làm việc**

**$ git rm --cached [file]**

**Loại bỏ tập tin có trong server nhưng ko loại bỏ trong tập tin hiện tại**

**$ git mv [file-original] [file-renamed]**

**Thay đổi tên file và chuẩn bị để commit**

**SUPPRESS TRACKING**

**Loại trừ các tập tin và đường dẫn tạm thời**

**$ git ls-files --other --ignored --exclude-standard**

**Liệt kê tất cả các tập tin bị loại bỏ trong project**

**\*.log**

**build/**

**temp-\***

**1 tập tin tên .gitignore ngăn chặn các phiên bản ngẫu nhiên của tập tin và các đường dẫn xác định phù hợp với tập tin.**

**SAVE FRAGMENTS**

**Đánh dấu và khôi phục những thay đổi chưa hoàn chỉnh**

**$ git stash**

**Tạm thời lưu trữ những tệp được theo dõi đã thay đổi**

**$ git stash list**

**Liệt kê tất cả các thay đổi**

**$ git stash pop**

**Khôi phục các tập tin được thay đổi gần đây**

**$ git stash drop**

**Hủy bỏ các thay đổi gần đây nhất được lưu lại**

**Xem lại lịch sử**

**Tìm và kiểm tra sự tiến triển của các tập tin trong dự án**

**$ git log**

**Liệt kê các phiên bản trước cho nhánh hiện tại**

**$ git log --follow [file]**

**Liệt kê các phiên bản trước cho 1 tập tin , bao gồm những tập đã thay đổi tên**

**$ git diff [first-branch]...[second-branch]**

**Hiển thị sự khác nhau giữa 2 nhánh**

**$ git show [commit]**

**Xuất ra dữ liệu và nội dung đã thay đổi của một commit cụ thể**

**Đồng bộ các thay đổi**

**Đăng ký 1 kho lưu trữ và trao đổi các phiên bản trước**

**$ git fetch [bookmark]**

**Tải xuống tất cả các thay đổi từ kho lưu trữ được đánh dấu**

**$ git merge [bookmark]/[branch]**

**Đồng bộ nhánh được đánh đấu với nhánh hiện tại**

**$ git push [alias] [branch]**

**Commit lên tất cả các nhánh hiện tại trên github**

**$ git pull**

**Tải xuống các thay đổi được đánh dấu kết hợp các thay đổi**

**Tái lập commit**

**Xóa bỏ các lỗi và những lịch sử đã thay đổi**

**$ git reset [commit]**

**Quay lại các lần commit trước lần commit vừa rồi. tránh sự thay đổi cục bộ**

**$ git reset --hard [commit]**

**Loại bỏ tất cả các thay đổi trước và quay lại lần commit cụ thể**