**Thiết lập chứng thực cá nhân**

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

git config --global user.email "username@gmail.com"

Lưu ý:

--global được sử dụng để áp dụng cho tất cả các projects. Nếu không sử dụng --global thì settings sẽ chỉ dùng cho riêng project đó.

**Tạo một kho chứa Git**

git init

Nếu như bạn muốn theo dõi một dự án trong Git, bạn cần ở trong thư mục của dự án đó. Lệnh này sẽ tạo một thư mục mới có tên **.git**, thư mục **.git** này chứa tất cả các thiết lập về Git cũng như toàn bộ thông tin về kho chứa.

**Sao chép một kho chứa đã tồn tại**

git clone https://github.com/user/repository.git

Câu lệnh trên sẽ tạo một thư mục mới có tên giống trên của repo.

**Nhánh trong git**

Ta có thể tạo ra nhiều nhánh (branch) khác nhau khi sử dụng Git:

Câu lệnh tạo mới một branch:

git branch <name\_branch>

Câu lệnh chuyển và tạo mới:

git branch -b <name\_branch>

Câu lệnh kiểm tra branch hiện tại:

git branch

**Chuyển nhánh**

Ta cần phải check out một nhánh trước khi thay đổi source code. Để checkout một nhánh, bạn dùng câu lệnh Git sau:

git checkout <name\_branch>

**Cập nhật thay đổi**

Sau khi bạn thay đổi source code: thêm mới, sửa, xoá files,… Bạn cần phải cập nhật lên Staging Area( khu vực trung gian giữa máy tính của bạn và Local Repository). Để cập nhật hết các files:

git add .

Bạn còn có thể bỏ tập tin ra khỏi Staging Area để không phải bị commit theo thì sử dụng câu lệnh:

git reset HEAD <file name>

Sau lệnh add, bạn cần sử dụng câu lệnh Commit để đẩy thông tin thay đổi lên Local Respository:

git commit -m "Message"

Note: "Message" là ghi chú những thay đổi của bạn khi bạn cập nhật file để người khác có thể biết bạn đã sửa đổi những gì.

**Cập nhật lên server**

Sau câu lệnh Commit, thông tin mới chỉ được cập nhật lên Local Repository. Nếu muốn cập nhật lên server thì bạn phải sử dụng câu lệnh push:

git push origin <name\_branch>

Ngoài ra, nếu chưa tồn tại remote trên server thì bạn cần phải add mới một remote trước rồi mới push:

git remote add origin <remote\_url>

git push origin <name\_branch>

**Gộp nhánh**

Sau một thời gian cập nhật các file và push lên git trên branch mới, bây giờ mình cần ghép (merge) code lại vào nhánh gốc (master). Trước tiên, cần phải checkout ra khỏi branch hiện tại cần gộp để vào branch master, sau đó thì dùng lệnh merge để ghép branch mới vào master:

git checkout master

git merge <new\_branch>

**Xem lại lịch sử commit**

git log

Lệnh git log sẽ cho bạn biết về người commit, ngày giờ, message của những lần commit đó.

Bạn có thể thêm -p vào để hiện chi tiết thông tin mỗi lần commit:

git log -p

Ngoài ra bạn có thể sử dụng thêm một số tuỳ chọn xem log khác để quá trình đọc log được tối ưu hơn:

--since, --after: Xem các lần commit kể từ ngày nhất định.

--until: Xem các lần commit trước từ ngày nhất định.

--author: Xem các lần commit của một người nào đó.

--grep: Lọc các chuỗi trong log và in ra.

**Xem thay đổi trước khi push**

git diff

Lệnh này giúp bạn biết những gì đã được thay đổi giữa nhánh hiện tại và nhánh trước nó.

**Gộp commit**

git rebase -i HEAD~

Sau dấu ~ là số commit bạn muốn gộp. Sau khi gõ lệnh này một cửa sổ trình soạn thảo hiện ra. Thay đổi ký tự pick của dòng các dòng sau dòng đầu thành s rồi lưu lại/kết thúc. Khi đó, trình soạn thảo để chỉnh sửa giải thích commit thiết lập cho commit sau khi đã tổng hợp sẽ được hiển thị, nên hãy chỉnh sửa lưu lại/kết thúc.

**Pull từ remote repository**

git pull origin master

Lệnh trên sẽ gộp những thay đổi mới kéo về từ máy chủ từ xa với nhánh hiện tại trên máy local.