**CẤU HÌNH NGƯỜI DÙNG**

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

Chỉ định tên tài khoản của bạn đã đăng kí trên GitHub.

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

Chỉ định tài khoản email của bạn đã đăng kí trên GitHub.

**$ git config --global color.ui auto**

Cho phép các màu sắc trên các dòng lệnh in ra màn hình.

**Tạo chỗ chứa**

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

Tạo mới chỗ chứa với tên người dùng chỉ định.

**$ git clone [url]**

Tải xuống dự án và toàn bộ các lịch sử phiên bản của dự án đó.

**Tạo thay đổi**

**$ git status**

Liệt kê tất các các tập tin mới hay đã được sửa để commit.

**$ git diff**

Liệt kê các xung đột.

**$ git add [file]**

Thêm file vào vào index.

**$ git diff –staged**

Xem những gì bạn đã staged mà chuẩn bị được commit.

**$ git reset [file]**

Không hiển thị tập tin, nhưng vẫn giữ nội dung của tập tin.

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

Ghi lại ảnh chụp nhanh vĩnh viễn trong lịch sử phiên bản.

**Thay đổi nhóm**

**$ git branch**

Liệt kê các nhánh có trong kho chứa.

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

Tạo tên nhánh.

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

Chuyển nhánh và cập nhật lại đường dẫn hiện tại.

**$ git merge [branch]**

Kết hợp lịch sử của nhánh cụ thể vào nhánh hiện tại.

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

Xóa nhánh được chỉ định tên.

**Đổi tên file**

**$ git rm [file]**

Xóa file trong thư mục gốc và tại nơi đang lưu trữ.

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

Loại bỏ các tập tin từ kiểm soát phiên bản, nhưng bảo vệ tập tin cục bộ.

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

Thay đổi tên file và sửa chúng để commit.

**TRỢ GIÚP**

**\*.log**

**build/**

**temp-\***

Một tập tin văn bản có tên là .gitignore ngăn chặn phiên bản ngẫu nhiên của

tệp và đường dẫn phù hợp với các tệp tin được xác định.

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

Liệt kê các file bị chặn trong dự án.

**LƯU TRỮ CÁC THÀNH PHẦN**

**$ git stash**

Tạm thời lưu trữ tất cả các tệp được theo dõi đã sửa đổi.

**$ git stash pop**

Phục hồi các tệp tin đã được lưu gần đây nhất.

**$ git stash list**

Liệt kê tất cả các thay đổi được đặt.

**$ git stash drop**

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

**XEM LẠI LỊCH SỬ**

**$ git log**

Liệt kê lịch sử phiên bản cho chi nhánh hiện tại.

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

Liệt kê lịch sử phiên bản cho một tập tin, bao gồm cả đổi tên.

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

Hiển thị nội dung giữa hai nhánh.

**$ git show [commit]**

Xuất ra siêu dữ liệu và thay đổi nội dung của cam kết cụ thể.

**LÀM LẠI COMMIT**

**$ git reset [commit]**

Hủy tất cả các cam kết afer [cam kết], giữ thay đổi cục bộ.

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

Loại bỏ tất cả lịch sử và thay đổi trở lại với cam kết cụ thể.

**ĐỒNG BỘ HÓA SỰ THAY ĐỔI**

**$ git fetch [bookmark]**

Tải xuống tất cả lịch sử từ dấu trang của kho.

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

Kết hợp chi nhánh của bookmark vào chi nhánh hiện tại.

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

Tải lên tất cả các chi nhánh địa phương cam kết với GitHub.

**$ git pull**

Tải xuống lịch sử dấu trang và kết hợp các thay đổi.