CONFIGURE TOOLING  
  
$ git config –global user.name “[name]”  
Đặt tên bạn muốn đính kèm vào các giao dịch cam kết  
  
$ git config –global user.email”[email]”  
Đặt email bạn muốn đính kèm vào các giao dịch cam kết  
  
$git config –global color.ui auto   
Cho phép tạo màu phù hợp cho đầu ra dòng lệnh  
  
CREATE REPOSITORIES

Tạo thư mục chứa tài nguyên  
$ git init [project-name]  
Tạo mới một thư mục với tên cụ thể

$ git clone [url]  
Tải về 1 dự án và tất cả các phiên bản của nó  
  
  
MAKE CHANGES

$ git status  
Liệt kê tất cả các tập tin mới hoặc sửa đổi để được cam kết  
  
$ git add [file]  
Thêm tệp để chỉnh sửa  
  
$ git reset [file]  
Bỏ trạng thái tập tin, nhưng bảo vệ nội dung của nó  
  
$ git diff  
Cho thấy các kết quả chưa được hiển thị  
  
$ git diff –staged  
Hiển thị các thông tin sai lệch giữa dàn dựng và phiên bản cuối cùng  
  
$ git commit -m "[descriptive message]"  
Ghi lại tập tịnh ơ trạnh thái vĩnh viễn trong lịch sử phiên bản

GROUP CHANGES

$ git branch  
Liệt kê tất cả các chi nhánh trong kho hiện tại  
  
$ git branch [branch-name]  
Tạo 1 nhánh mới

$ git checkout [branch-name]  
Chuyển sang 1 nhánh cụ thể và cập nhật thư mục làm việc

$ git merge [branch]  
Kết hợp lịch sử của nhánh cụ thể vào nhánh hiện tại

$ git branch -d [branch-name]  
Xóa 1 nhánh cụ thể  
  
  
REFACTOR FILENAMES  
$ git rm [file]  
Xóa tập tin khỏi thư mục làm việc và xóa bỏ trạng thái  
  
$ 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 tập tin và chuẩn bị cam kết  
  
  
  
SUPPRESS TRACKING

$ git ls-files --other --ignored --exclude-standard  
Liệt kê tất cả các tệp bị bỏ qua trong dự án này

\*.log  
build/  
temp-\*

Một tập tin văn bản có tên là .gitignore ngăn chặn phiên bản ngẫu nhiên của  
tệp và xác định đường dẫn phù hợp với các tệp tin

SAVE FRAGMENTS

$ git stash  
Tạm thời lưu trữ tất cả các tệp được theo dõi đã sửa đổi

$ git stash list  
Liệt kê tất cả các thay đổi được đặt

$ git stash pop  
Phục hồi các tệp tin đã được lưu gần đây nhất

$ git stash drop  
Hủy bỏ các thay đổi gần đây nhất được lưu lại  
  
  
  
REVIEW HISTORY

$ 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  
  
  
  
REDO COMMITS

$ git reset [commit]  
Hoàn tất tất cả các cam kết sau khi [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  
  
  
  
  
  
SYNCHRONIZE CHANGES

$ git fetch [bookmark]  
Tải xuống tất cả lịch sử từ thư mục được chọn

$ git merge [bookmark]/[branch]  
Kết hợp nhánh được đánh dấu vào nhánh hiện tại

$ git push [alias] [branch]  
Tải lên tất cả các nhánh local cam kết với GitHub

$ git pull  
Tải xuống lịch sử dấu trang và kết hợp các thay đổi