**CONFIGURE TOOLING**

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

Nhập username của tài khoản git

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

Nhập email của tài khoản git

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

Đổi mầu của các dòng lệnh xuất ra

**CREATE REPOSITORIES**

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

Tạo mới 1 Repo

**$ git clone [url]**

Tải dự án trên tài khoản Git về máy

**MAKE CHANGES**

**$ git status**

Xem tất cả các file mới hoặc đã chỉnh sửa sau lần commit cuối

**$ git diff**

Hiện tất cả những file mới hoặc đã sửa khác ở các nhánh khác nhau

**$ git add [file]**

Đưa tất cả các file đã sửa và mới vào trạng thái tạm thời chuẩn bị update cho phiên bản mới đang hiện hành

**$ git diff –staged**

Hiện ra tất cả các file có các sự khác nhau giữa nhữnng phiên bản hiện tại và các file của phiên bản trước đó

**$ git reset [file]**

Lùi phiên bản của file đó nhưng vẫn giữ nguyên nội dung của file

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

Lưu file trong trạng thái tạm thời lên Local Git và tạo ghi chú cho các file thay đổi

**GROUP CHANGES**

**$ git branch**

Hiển thị ra tất cả các nhánh hiện tại có trong dự án hiện tại trong Local

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

Tạo mới 1 nhánh và đặt tên tùy chọn

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

Chuyển đổi vào nhánh khác và cập nhập công việc theo nhánh đó

**$ git merge [branch]**

Kết hợp các nhánh riêng biệt vào nhánh hiện tại

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

Xóa nhánh

**REFACTOR FILENAMES**

**$ git rm [file]**

Xóa File khỏi thư mục hiện tại

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

Gỡ file khỏi thư mục hiện tại nưng vẫn giữ file

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

Đổi tên file và commit lên local

**SUPPRESS TRACKING**

**\*.log**

**build/**

**temp-\***

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

Tất các các file đã được bbor qua trong project

**SAVE FRAGMENTS**

**$ git stash**

Lưu trữ tất cả các Modified

**$ git stash pop**

Khôi phục dữ liệu gần nhất

**$ git stash list**

Hiển thị tất cả các file đã thay đổi trong thư mục

**$ git stash drop**

Hủy tập hợp các thay đổi được lưu trữ gần đây nhất

**REVIEW HISTORY**

**$ git log**

Liệt kê lịch sử phiên bản cho nhánhhiện tại

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

Liệt kê lịch sử phiên bản cho một ﬁle, bao gồm cả tên

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

Hiển thị nội dung khác nhau giữa hai nhánh

**$ git show [commit]**

Dữ liệu đầu ra và các thay đổi nội dung của cam kết cụ thể

**REDO COMMITS**

**$ git reset [commit]**

Hoàn tác tất cả các cam kết afer [cam kết], duy trì các thay đổi cục bộ

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

Hủy bỏ tất cả lịch sử và các thay đổi trở lại cam kết cụ thể **SYNCHRONIZE CHANGES**

**$ git fetch [bookmark]**

Tải xuống tất cả lịch sử từ dấu trang của kho lưu trữ

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

Kết hợp chi nhánh của dấu trang vào chi nhánh cục bộ hiện tại

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

Tải tất cả cam kết của chi nhánh địa phương lên GitHub

**$ git pull**

Tải xuống lịch sử đánh dấu và kết hợp các thay đổi