Một vài ý lý thuyết :

STAGING AREA : là nơi mà thư mục được chuẩn bị cho commit. Muốn COMMIT tập tin nào thì tập tin đấy phải ở trong STAGING AREA, và có trạng thái STAGED. Có thể bỏ qua STAGING AREA với tham số “ –a “ trong lệnh “ git commit “ . Ví dụ

COMMIT là một hành động để git lưu lại một bản chụp ( snapshot ) của các sự thay đổi trong thư mục làm việc, và các tập tin và thư mục được thay đổi nằm trong STAGING AREA. Để COMMINT thì phải có cả trạng thái TRACKED bên cạnh STAGED.

+Trạng thái TRACKED là tập tin được đánh dấu theo dõi để bạn làm việc với nó. Và trạng thái TRACKED sẽ đi kèm các trạng thái phụ là UNMODIFIED(Chưa chỉnh sửa ), MODIFIED ( đã chỉnh sửa ) và STAGED ( sẵn sang cho COMMIT ).

+Trạng thái UNTRACKED là tập tin còn lại mà bạn không muốn làm việc với nó trong Git.

-Khởi tạo một kho trong một danh mục đang tồn tại :

$ cd/Users/user/your\_respository

sẽ tạo ra một danh mục con đc đặt tên là “.git” . chứa toàn bộ những kho file cần thiết của bạn.

-Tạo local Repository

$ git int “tên\_thư\_mục\_kho\_chứa”

ví dụ $ git init git\_tur

thư mục bên trong kho chứa là “ .git/ “ sẽ bị ẩn, k cần phải đụng chạm gì.

-Nếu kho chứ có sẵn mã nguồn thì cần phải đưa các tập tin về trạng thái TRACKED nhằm có thể làm việc được với Git. Dùng lệnh “ git add tên\_file “ , có thể sử dụng “\*” để gom toàn bộ. Sau đó dùng “ git status” để xem các tập tin đã được TRACKED.

-Tạo Repository trên Github :

 Đường dẫn có dạng http://github.com/$user-name/$repository

ví dụ : http:/[github.com/hoanporsche/git\_example.git](http://github.com/hoanporsche/git_example.git)

-Đưa Repository từ Github về máy cá nhân :

$ git clone [url]

ví dụ : $ git clone <http://github.com/hoanporsche/git_example>.git

-Truy cập vào mục vừa clone về và tạo file

$ cd git\_example

$ echo “# Huong dan git co ban” > README.md

$ git add README.md //đưa tệp tin vào STAGING AREA

$ git commit -m “First commit on Github” //COMMIT với lời nhắn “…”

-Sau khi commit xong, tập tin đc commit vẫn chưa thể ở trên github, phải làm thêm lệnh nữa là “ git push “

$ git push origin master

sau đó phải đăng nhập tài khoản github

“ origin “ là tên remote và “ master “ là tên branch

-Xem “ git log “

Là để xem lịch sử các lần commit trước đó

$ git log

+Mỗi lần commit sẽ có một checksum riêng, và cũng ghi rõ ai à người commit vào và commit ngày tháng bao nhiêu

+Nếu log quá dài, dùng phím mũi tên lên xuống để đọc tiếp và ấn Ctrl + Z để thoát.

+ Muốn xem 1 lần commit gần nhất thì them -1 vào

$ git log -1

+ Một số tuỳ chọn :

--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.

Ví dụ : $ git log –author=contact@hoanporsche

-Lọc với –pretty

Tham số này rất có ích nếu bạn muốn loc xem đối tượng nào đó trong lịch sử commit, ví dụ như chỉ xem lời nhắn commit hoặc chỉ xem email của người commit .

Cách sử dụng là phải viết kèm các tag của nó :

$ git log –pretty=”%TAG”

+ Danh sách các tag :

%H : Commit hash

%h :

%T

%t

%P

%p

%an

%ae

%ad

%ar

%cn

%ce

%cd

%cr

%s : Subject

-Undo Commit

Nếu cần undo lần commit trước thì có thể sử dụng tham số “ -–amend “ trong lệnh “ git commit “

-Bỏ tập tin khỏi STAGING AREA

Dùng lệnh “ git reset HEAD “ tên\_file ”

Đánh dấu COMMIT với TAG

Trong khi COMMIT các lần chỉnh sửa thì mọi thứ sẽ được lưu vào log. Tuy nhiên COMMIT quá nhiều thì sẽ gây khó khăn cho việc xem lại nên cần phải gắn thẻ đánh dấu (tag) cho mỗi COMMIT và khi cần xem lại cần sử dụng lệnh” git show tên\_tag “ là có thông tin rõ ràng, dễ đối chiếu ( diff ) . K cần nhớ checksum mà chỉ cần nhớ tag là có thể taoj ra branch từ tag - > thuận tiện phân nhánh .

-Lightweight Tag và Annotated Tag

+ Lightweight Tag : các tag này đơn thuần là đánh dấu snapshot của commit

+ Annotated Tag : với tag này, bạn có thể đặt tiêu đề cho tag, và khi xem nó sẽ có thông tin về người tag , ngày tag ….

-Tạo Lightweight Tag :

b1: “ git tag “ để xem danh sách các tag có trong dự án

b2: “ git tag “tên\_tag”

ví dụ : $ git tag

$ git tag v1.0

2 bước này có thể thay đổi cho nhau.

$ git tag v1.0

$ git tag

Xem thông tin của lần COMMIT : “ git show “tên\_tag”

$ git show v1.0

-Tạo Annotated Tag

Cũng như Lightweight Tag nhưng them “ –a “ và “ –m “

$ git tag –a v1.0-an –m “Ra mat phien ban moi 1.0”

$ git show v1.0-an // để xem tag

-Thêm tag cho các COMMIT cũ

$ git log –pretty=online