c. Tạo 1 Repository

- Kho chứa Repository là mục ẩn .git

- Để tạo repository trong một project của bạn cần mở terminal trên Linux, trỏ đến thư mục chứa project đó và gõ lệnh:

*$git init*

*-* Để thêm tập tin của Project vào git thì ta dùng lệnh:

*$ git add .*

*hoặc*

*$git add \**

*-* Thêm 1 tập tin nào đó vào git ta dùng lệnh

*$ git add [Tên tệp tin]*

*-* Với thao tác git add với các tệp tin tức là bạn đã đưa chúng vào trạng thái tracked. Với những thay đổi kèm theo nội dung thay đổi thì chúng ta phải commit chúng với lệnh:

*$git commit -m “[nội dung message]”*

-m để thực hiện ghi chú, chú thích lại lần thay đổi đó.

d. Sao chép một kho chứa

Để sao chép một kho chứa đã tồn tại trên server thì ta dùng lệnh:

*$ git clone url*

Muốn sao chép kho chứa này vào một thư mục có tên khác dùng lệnh:

*$ git clone url [tên thư mục]*

*e.* Vòng đời các trạng thái của tập tin



Ban đầu khi các tập tin chưa thêm vào kho chứa git thì nó ở trạng thái *untracked*, khi thêm vào kho chứa git bằng lệnh **git add** thì nó chuyển sang trạng thái *tracked.* Bạn stage các tập tin bị thay đổi và *commit* tất cả các thay đổi đã được *staged* đó. Quá trình đó cứ lặp đi lặp lại.

- Kiểm tra trạng thái của tập tin dùng lệnh:

*$ git status*

Cũng có thể xem bản ngắn gọn của trạng thái bằng lệnh:

*$ git status -s*

f. Quản lý các tập tin đã thay đổi

Khi thay đổi một tập tin nào đó cần chạy lại lệnh

*$ git add [Tên tập tin cần thay đổi]*

Khi đã sẵn sàng commit nhưng bạn lại muốn sửa gì đó trong tập tin đó, sau khi sửa xong thì vẫn cần chạy lệnh

*$ git add [Tên tập tin cần thay đổi]*

Để đưa tập tin vào Staged và sẵn sàng commit.

g. Bỏ qua các tập tin

Thường thì có những tập tin được git tạo ra tự động mà mình không cần theo dõi chúng. Trong trường hợp này ta tạo 1 tập tin .gitignore

h. Xem các thay đổi Staged và Unstaged

Để xem chi tiết các thay đổi chưa được staged ta dùng lệnh

$ git diff

Lưu ý tập tin đã được staged thì khi git diff sẽ không hiển thị gì cả. Ví dụ bạn có một thay đổi nào đó và chưa git add lại thì có thể dùng git diff để xem chi tiết thay đổi, nhưng một khi bạn git add lại tức là đưa nó vào staged thì git diff sẽ không thể hiển thị những thay đổi cho bạn được.

i. Thay đổi nội dung commit

Lệnh:

*$ git commit -m “[message]”*

k. Xóa tập tin

Khi gõ lệnh

*$ rm [tên tập tin]*

Thì nó chỉ bị xóa khỏi thư mục làm việc và vẫn trong thư mục của git.

Nhưng khi chạy lệnh:

*$ git rm [tên tập tin]*

*rm “[tên tập tin]”*

Thì tập tin sẽ bị xóa ra khỏi git

Nhưng nếu muốn giữ tập tin trên ổ cứng máy mà không muốn git theo dõi nó nữa thì ta dùng lệnh

*$ git rm --cached [tên tập tin]*

l. Di chuyển tập tin

*$ git mv [tên tập tin muốn chuyển] [Tập tin đích]*

m. Xem lịch sử commit

$ git log :liệt kê tất cả các commit với commit mới nhất ở đầu và theo thứ tự vậy, gồm cả tên người commit và địa chỉ mail của họ, ngày tháng commit

$ git log -p :Hiển thị các diff của commit

$ git log -p -2: Giới hạn xem 2 commit gần nhất

$ git log --stat: xem một số thống kê tóm tắt cho mỗi commit

$ git log --pretty=online : Hiển thị mỗi commit trên một dòng

$ git log –since=2.weeks : hiển thị các commit trong 2 tuần gần nhất

n. Xem các kho chứa từ xa

Lệnh:

*$ git remote -v*

3. Git Branching

- Lệnh tạo nhánh mới:

$ git branch [tên nhánh]

Để chuyển sang nhánh khác ta dùng lệnh

$ git checkout [Tên nhánh muốn chuyển sang]

Tạo 1 branch mới và chuyển đến branch đó

$ git checkout -b [tên nhánh]

Và từ đây ta có thể thực hiện commit với nhánh mới

-Trộn nhánh:

Ban đầu chuyển về nhánh master

$ git branch master

sau đó tiến hành trộn nhánh. Ví dụ trộn nhánh Fixed với nhánh master

$ git merge Fixed

- Xóa nhánh

Xóa nhánh Fixed

$ git branch -d Fixed

- Xem các nhánh hiện tại:

$ git branch

-Xem commit mới nhất trên từng nhánh:

$ git branch -v

-Xem nhánh nào đã được tích hợp vào nhánh hiện tại:

$ git branch --merged

-Xem các nhánh chưa được tích hợp

$ git branch --no-merged