**CODERDAIHIEP \_ GIT version2**

**1.Khái niệm cơ bản:**

**Repository ( hay Repo):** kho chức toàn bộ project + lịch sử commit + file/folder + người code,...

**Repository gồm:**

+ Remote repository: kho đặt trên server của github ( cái project mà mình clone về).

+ Local repository: kho đặt trên máy tính cá nhân (laptop) tức là cái project mình dùng để code trên máy.

**Branch**: là nhánh, 1 Repository(repo) có nhiều nhánh. ví dụ:

---------------- <--------- Nhánh 1 -----> Repository local máy 1

-Repository- <------------ Nhánh 2 -----> Repository local máy 2

- remote - <------------ Nhánh 3 -----> Repository local máy 3

---------------- <--------- Nhánh 4 -----> Repository local máy 4

+ Cả 4 máy đều được phép cập nhật <Repository remote> thông qua các nhánh 1,2,3,4( branch).

+ Chú ý: tránh nhầm lẫn nhánh local và nhánh remote.

**2. Cách mà các branch làm việc (phần này khó có thể tham khảo):**

(\*viết tắt br1, br2, br3 ứng với branch 1, branch 2, branch 3. br\_merge là branch chứa tất cả các công việc của 3 branch vừa rồi.A,B,C,... là công việc.Đường thẳng là thời gian thực hiện

công việc của từng branch)

**br1** O------------A------------------------B

**br2** O---------------C--------------D

**br3** O----------E----------F-----G

**Sau khi hợp nhất thì:**

br1 = br2 = br3 =br\_merge O----------E-A--C-----F-----G--D------B

**Cách mà git branch dùng merge lấy tất cả công việc:**

**B1:** Đầu tiên 3 branch br1,2,3 làm 3 công việc A,C,E.Chú ý so sánh độ dài OA, OC, OE Thì OC > OA > OE hay thời gian hoàn thành công việc của 3 branch tại repo local là T2 > T1 > T3 ( T là thời gian)

**=>** Thứ tự push code của 3 người sở hữu 3 br là 3,1,2. Vậy có nghĩa là người nào xong sớm thì phải push lên trước, br\_merge sẽ là:

**br\_merge** O----------E

**br1**  O------------A

**br2**  O---------------C

**br3**  O----------E

**B2:** Ta thấy: br\_merge = br3, vì đều có công việc E. Nhưng br1 và br2 thì lại chưa có công việc E. br1,br2 khác br\_merge

Vậy ta cần thêm công việc E vào 2 br1 và br2 ( có nghĩa là lúc này, để lấy được công việc E, thì br1, br2 phải pull code của br3 về)

**br1** O----------E--A

**br2** O----------E-----C

**B3:** Tiếp tục bổ sung A và C vào branch\_merge.So sánh br1 và br2 thì T2 > T1 ( do việc C làm lâu hơn việc A) -> phải hợp nhất A trước rồi mới đến C

**br**\_**merge** O----------E--A

**br1** O-------------A

**br2** O----------E--A--C

**br3** O----------E--A

**B4:** tương tự hợp nhất việc C vào 2 br1, br3 ta được

**br\_merge** O----------E--A--C

**br1** O----------E--A--C

**br2** O----------E--A--C

**br3** O----------E--A--C

**... Cho tới bước cuối**

**3.Tổng hợp câu lệnh git:**

4.1) git init

+Lệnh này được sử dụng để khởi tạo kho lưu trữ mới và trống (repo local)

4.2) git add .

+ Đưa tất cả file, folder vào kho git local chuẩn bị commit.

4.3) git add <tên\_file>:

+ chỉ đưa <tên\_file> đó vào kho git local

4.4) git commit -m "message":

+ "message" là chú thích, mô tả cho tất cả các file có trong git local.

4.5) git status

+ xem trạng thái các file, folder đã được đưa vào kho git local

4.6) git push

+ đưa dữ liệu ở repo local lên repo remote ở branch mình đã push trước đấy.

4.7) git pull origin <name\_branch>

+ lấy dữ liệu repo remote về repo local tại <name\_branch> bất kì.

4.8) git merge <name\_branch>

+ hợp nhất 2 branch thành 1 branch, với branch thứ nhất là branch mình đang đứng + thứ hai là <name\_branch> mình muốn hợp nhất.

4.9) git branch

+ trả về tên branch đang đứng.

4.10) git branch <branch-name>

+ tạo branch mới với tên là <branch-name>

4.11) git checkout <name\_branch>

+ di chuyển sang name\_branch

4.12) git checkout -b <name\_branch>

+ Làm 2 việc, việc 1 là tạo ra branch mới với tên là <name\_branch>, việc 2 sao chép dữ liệu branch đang đứng sang <name\_branch>

4.13) git push origin --delete <name\_branch>:

+ xóa nhánh ở repo REMOTE ( chú ý là remote chứ không phải local).

4.14) git branch --delete <old-branch>

+ xóa nhánh ở repo local. Trước khi xóa nhánh phải nhảy sang nhánh khác

4.15) git remote set-url origin <news\_https-url>

+ Thay đổi địa chỉ push lên với <news\_https-url> là địa chỉ mới

4.16) git remote add origin <https-url>

+ Thêm một đường dẫn ( địa chỉ) của repo remote để repo local dùng địa chỉ đó để push dữ liệu lên với <https-url> là địa chỉ

4.17) git config --global user.name "John Doe"

git config --global user.email "john@example.com"

+ --global được sử dụng để áp dụng cho tất cả các projects. Nếu bạn ko sử dụng --global thì settings sẽ chỉ dùng cho riêng project đó.

4.18) git clone https://github.com/user/repository.git

+ Clone một remote repo

4.19) git remote -v

+ Liệt kê các remote urls

4.20) git log và git log -2

+ Xem commit history , Xem commit history cho hai commits gần nhất.

4.21) git branch -a

+ Xem tất cả các nhánh: local + repo.

4.22) git fetch --all

+ Lấy tất cả branch

4.23) git push -f origin main

+có thể ghi đè bất kỳ kiểm tra nào mà Git thực hiện bằng cách sử dụng "ép buộc".