

THỰC HÀNH GITKRAKEN

PHẦN 2

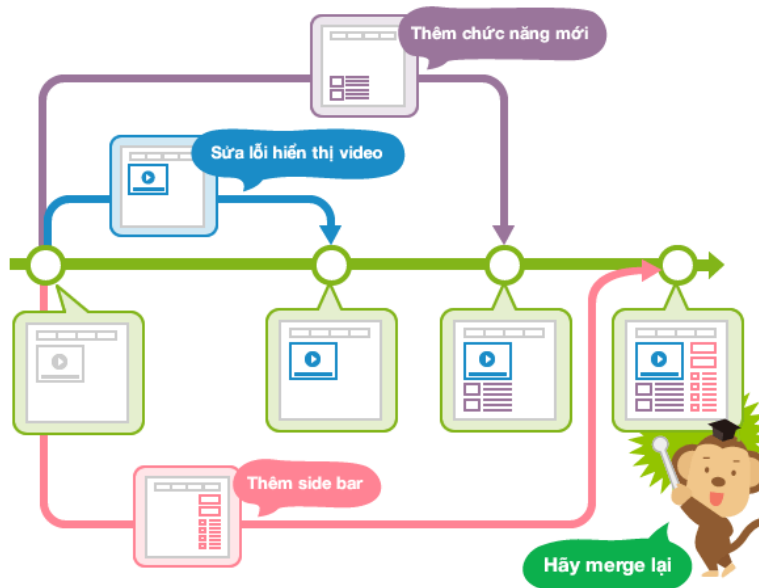
I. Khái niệm cơ bản về Branch, Merge

Tham khảo thêm tại link: https://backlog.com/git-tutorial/vn/stepup/stepup1_1.html

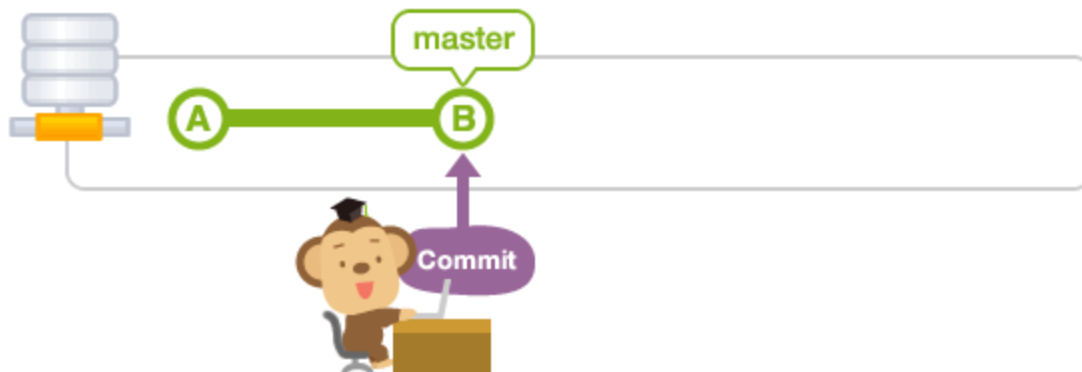
Trong việc phát triển phần mềm, thì ứng với một phần mềm có nhiều thành viên đồng thời tiến hành thêm chức năng hay là tiến hành chỉnh sửa lỗi cùng một lúc. Và ở tình trạng tồn tại của nhiều phiên bản đã phát hành thì cũng phải lưu giữ từng phiên bản.

Chính vì vậy để hỗ trợ quản lý phiên bản hay thêm nhiều chức năng được tiến hành song song, một chức năng được trang bị thêm được gọi là branch ở Git.

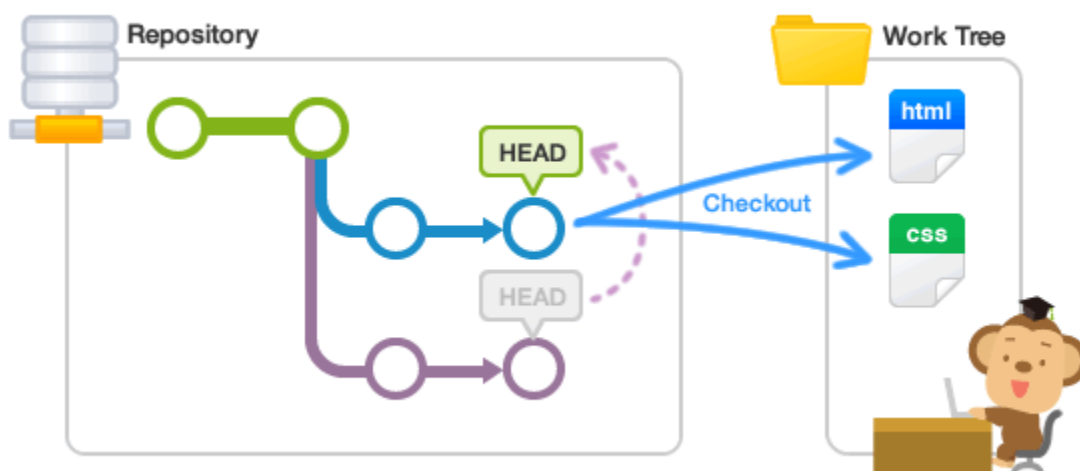
Branch là phương pháp dùng để phân nhánh và ghi lại luồng của lịch sử. Branch đã phân nhánh sẽ không ảnh hưởng đến branch khác nên có thể tiến hành nhiều thay đổi đồng thời trong cùng 1 repository.



Khi tiến hành commit lần đầu trong repository thì Git sẽ tạo ra một branch có tên là main (master). Vì thế những lần commit sau sẽ được thêm vào branch main cho đến khi chuyển đổi branch.



Để chuyển đổi branch làm việc thì sẽ thực hiện thao tác gọi là **checkout**. Khi thực hiện checkout, trước tiên nội dung của lần commit cuối cùng trong branch chuyển đến sẽ được mở ra trong work tree. Và commit đã tiến hành sau khi check out thì sẽ được thêm vào branch sau khi di chuyển đến.



Branch chủ đề đã hoàn thành công việc, cuối cùng sẽ được tích hợp vào branch tích hợp. Việc tích hợp branch thì có hai phương pháp, phương pháp sử dụng merge và phương pháp sử dụng rebase. Dựa vào việc sử dụng phương pháp nào mà lịch sử của branch sau khi tích hợp sẽ có sự khác biệt lớn.

Thao tác Merge:

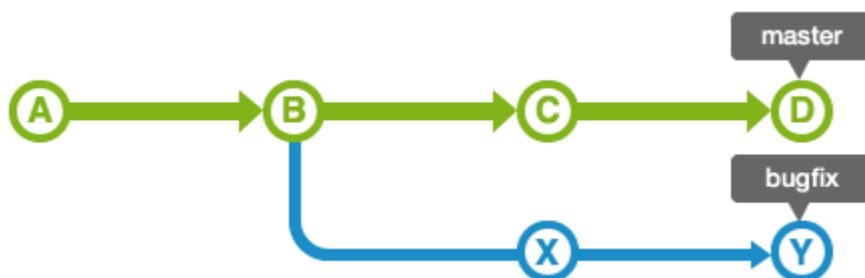
Khi sử dụng merge, có thể tổng hợp nhiều luồng lịch sử. Ví dụ, có branch bugfix phân nhánh ra từ branch master như sơ đồ bên dưới.



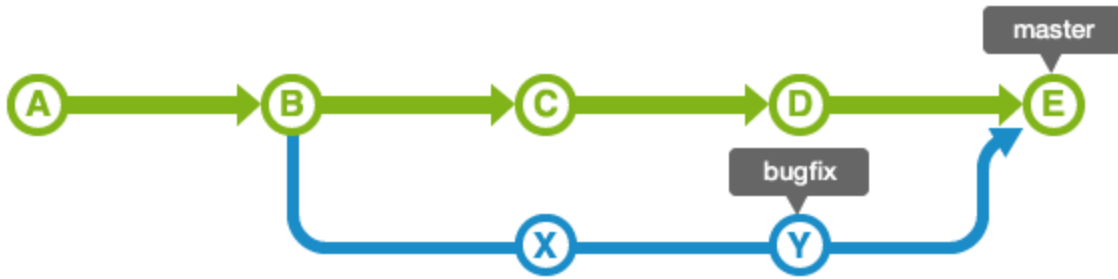
Khi merge branch bugfix này vào branch master, nếu trạng thái branch master không có gì thay đổi thì có thể thực hiện merge hết sức đơn giản. Vì lịch sử của branch bugfix sẽ bao gồm tất cả lịch sử của branch master, nên branch master chỉ có việc di chuyển đơn giản là có thể lấy được nội dung của branch bugfix. Và, người ta gọi merge như thế này là **merge fast-forward** (chuyển tiếp nhanh).



Nhưng, cũng có trường hợp lịch sử của branch master vẫn đang tiến triển sau khi phân branch bugfix. Trường hợp này thì cần thiết phải tổng hợp nội dung thay đổi của branch master và nội dung thay đổi của branch bugfix thành một.



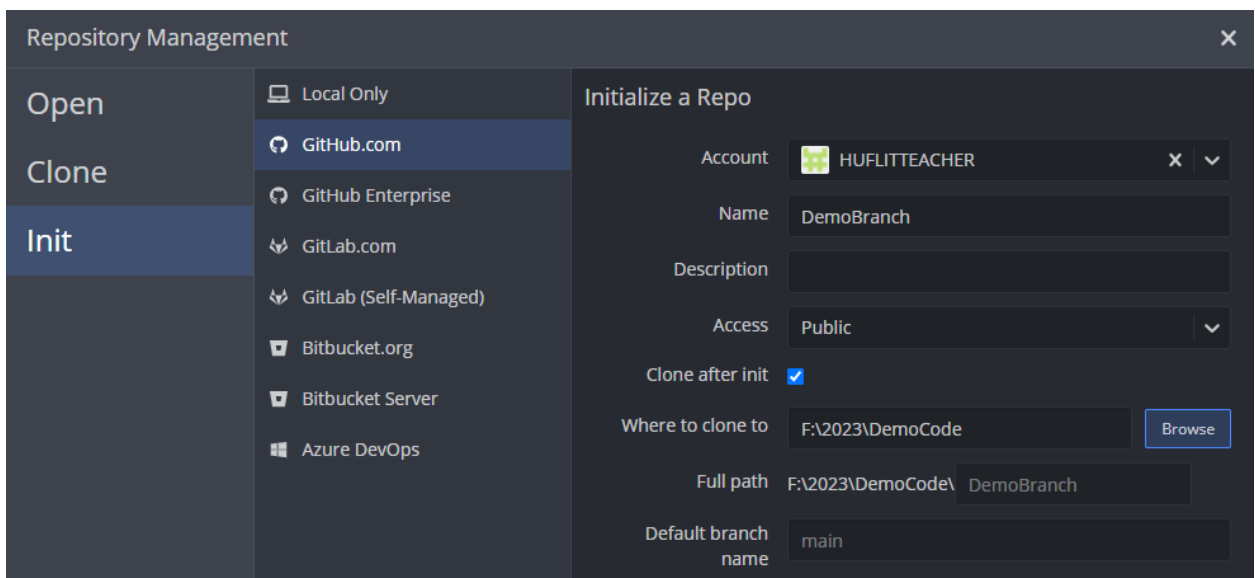
Vì thế, merge commit đã lấy thay đổi của cả hai branch sẽ được tạo ra. Đầu branch master sẽ di chuyển đến commit đó.



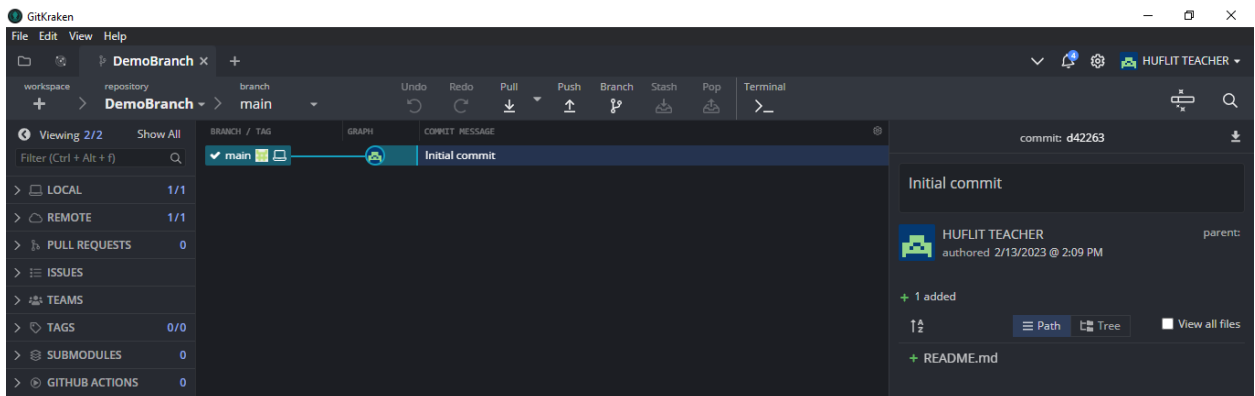
II. Thực hành các thao tác trên Branch với GitKraken

Bước 1: Chuẩn bị

Vào GitKraken, chọn File → Init Repo, điền thông tin tương ứng và bấm “Create Repository and Clone”



Trạng thái ban đầu:

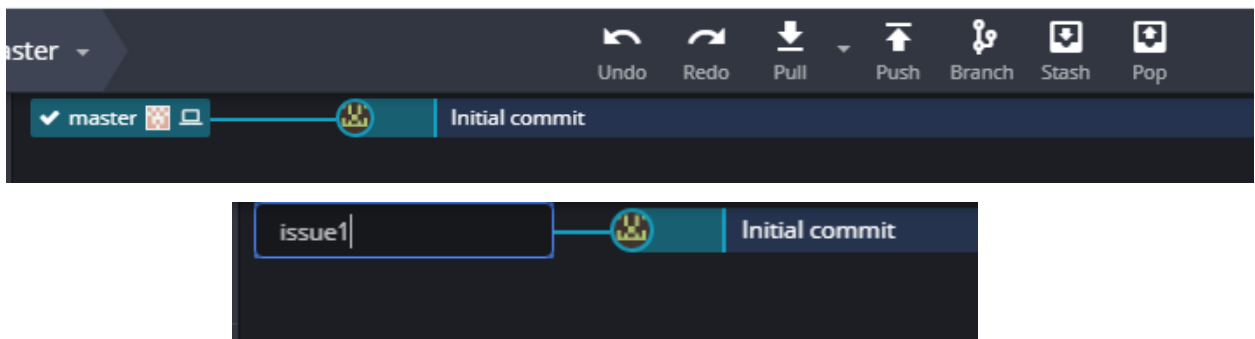


Lịch sử tại thời điểm này thì giống thể này.

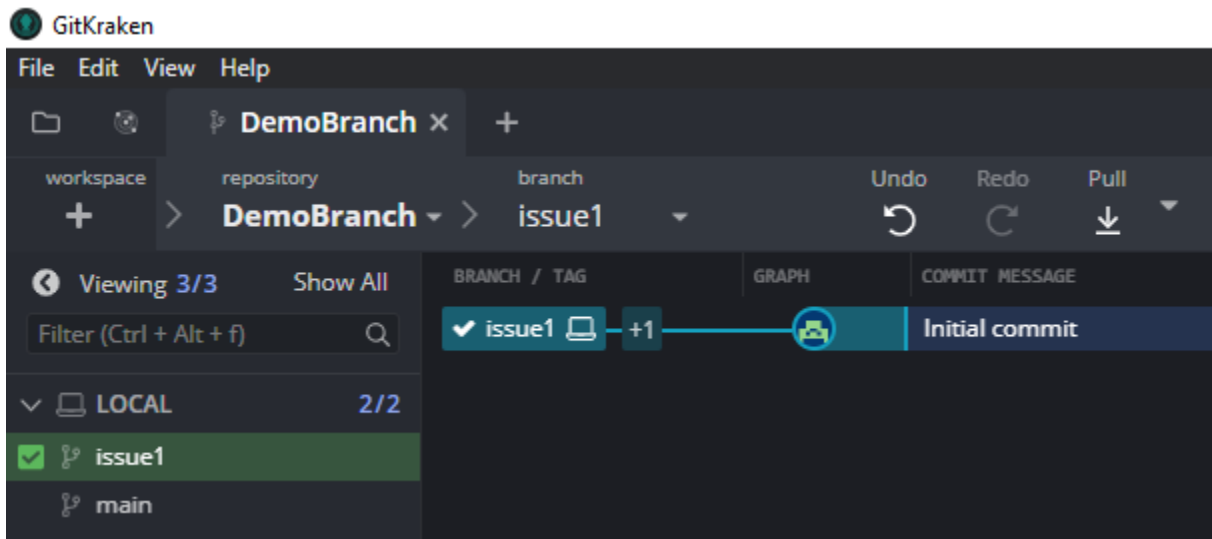


Bước 2: Tạo branch

Giả sử ta cần tạo một branch tên là “issue1”, ta bấm nút “Branch” như hình dưới



Kết quả: (lưu ý quan sát thấy ta đang checkout ở branch main/master)

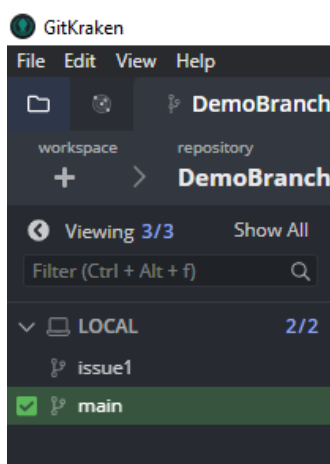


Lịch sử tại thời điểm này như sau.



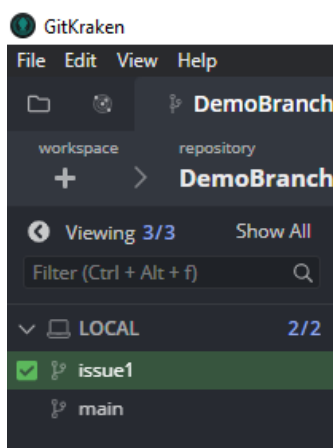
Bước 3: Chuyển đổi qua lại giữa các Branch – Thao tác “Checkout”

Để chuyển đổi qua lại giữa các nhánh, ta double click vào từng nhánh



Chọn nhánh ta muốn chuyển đến

Kết quả:





Lịch sử tại thời điểm này như sau.



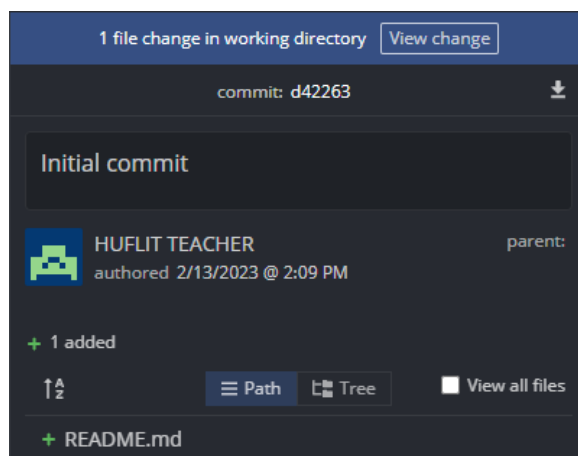
Bước 4: Thử cập nhật bên branch issue1

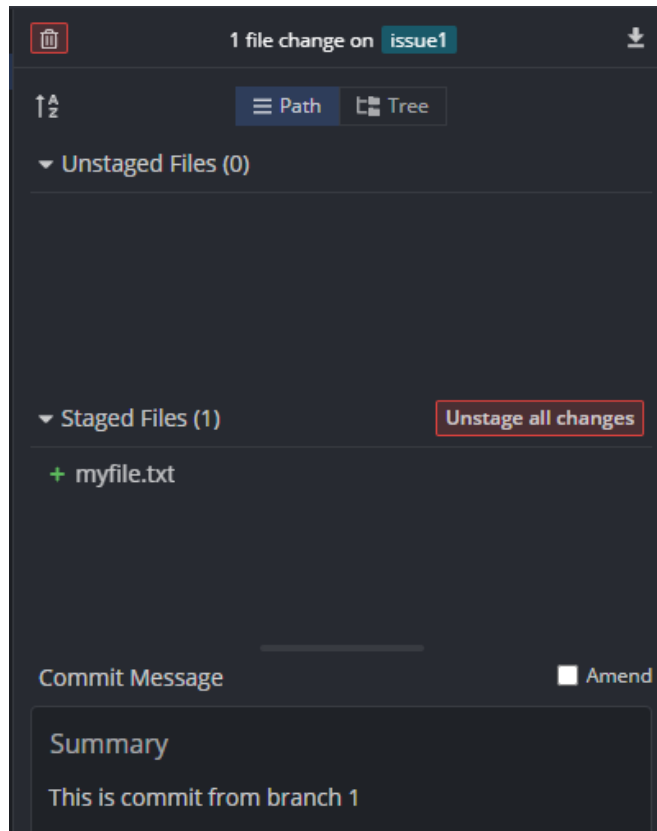
Đầu tiên vào thư mục của branch issue1 trên local, tạo file myfile.txt

/ Volume (F:) > 2023 > DemoCode > DemoBranch

Name	Date modified	Type	Size
 myfile.txt	13/02/2023 2:12 PM	Text Document	0 KB
 README.md	13/02/2023 2:09 PM	MD File	1 KB

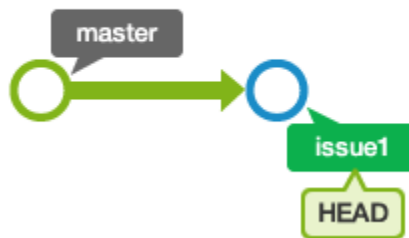
Thực hiện thao tác commit file này trên branch issue1



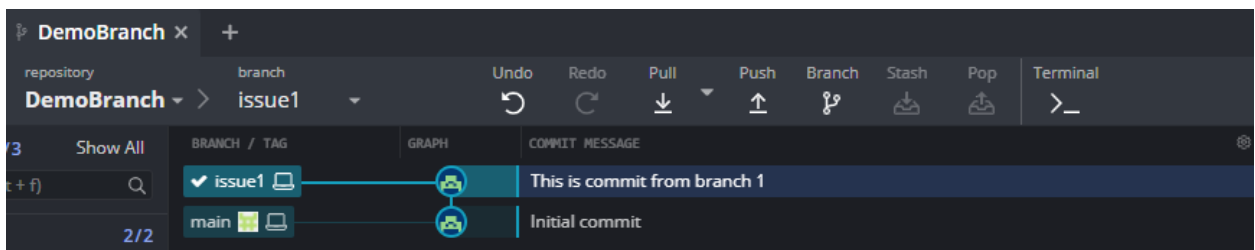


Ta để Summary là “This is commit from branch 1”

Lịch sử tại thời điểm này như sau.

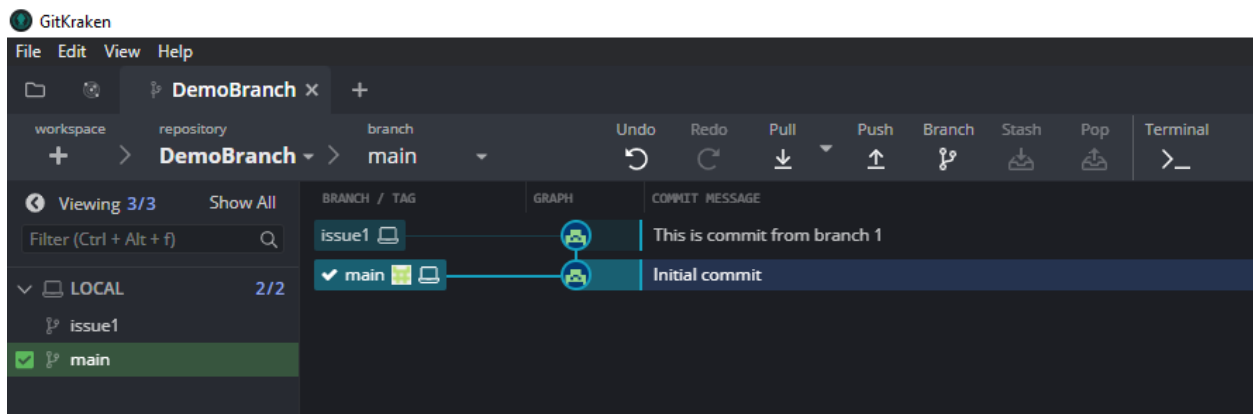
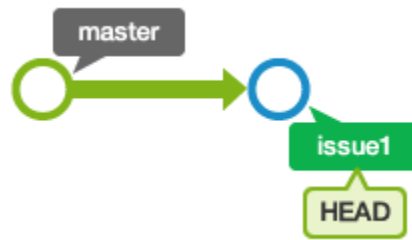


Kết quả (tại Local)



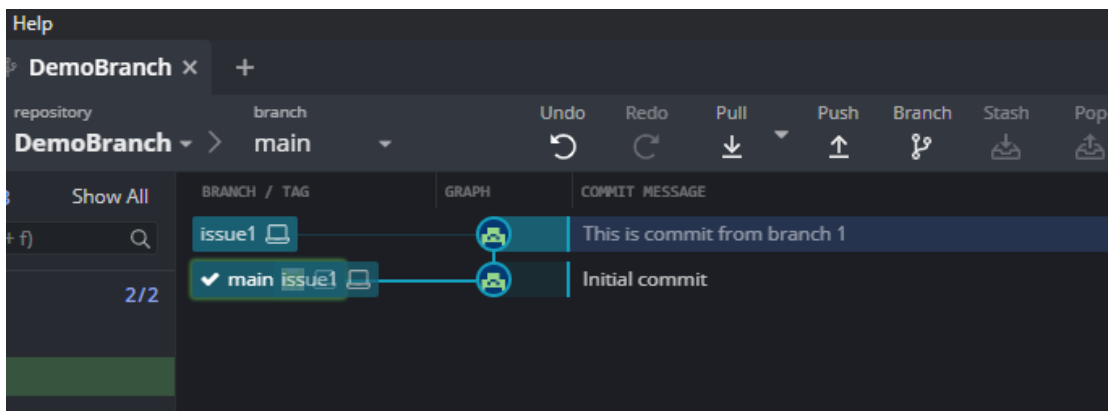
Bước 5: Tiến hành thao tác Merge

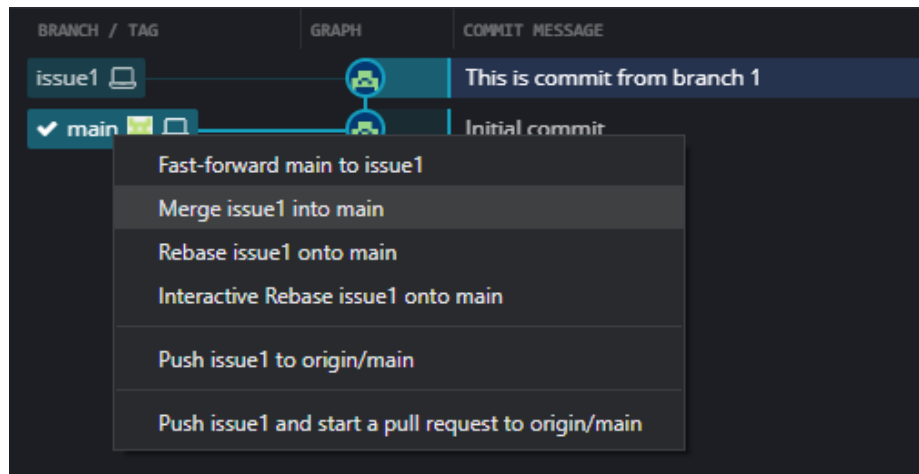
Hiện tại:



Để merge issue1 vào master, đầu tiên cần checkout tại main/master (bằng cách double click vào main)

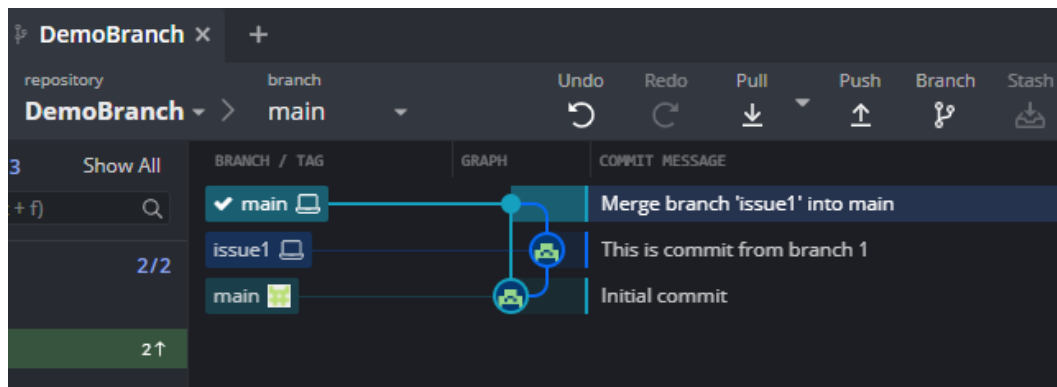
Thực hiện kéo issue1 vào main:





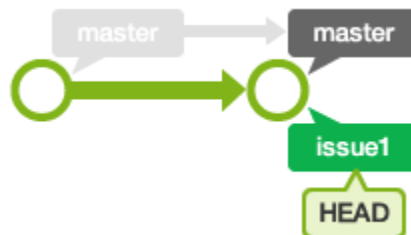
Chọn “Merge issue1 into main”

Kết quả:



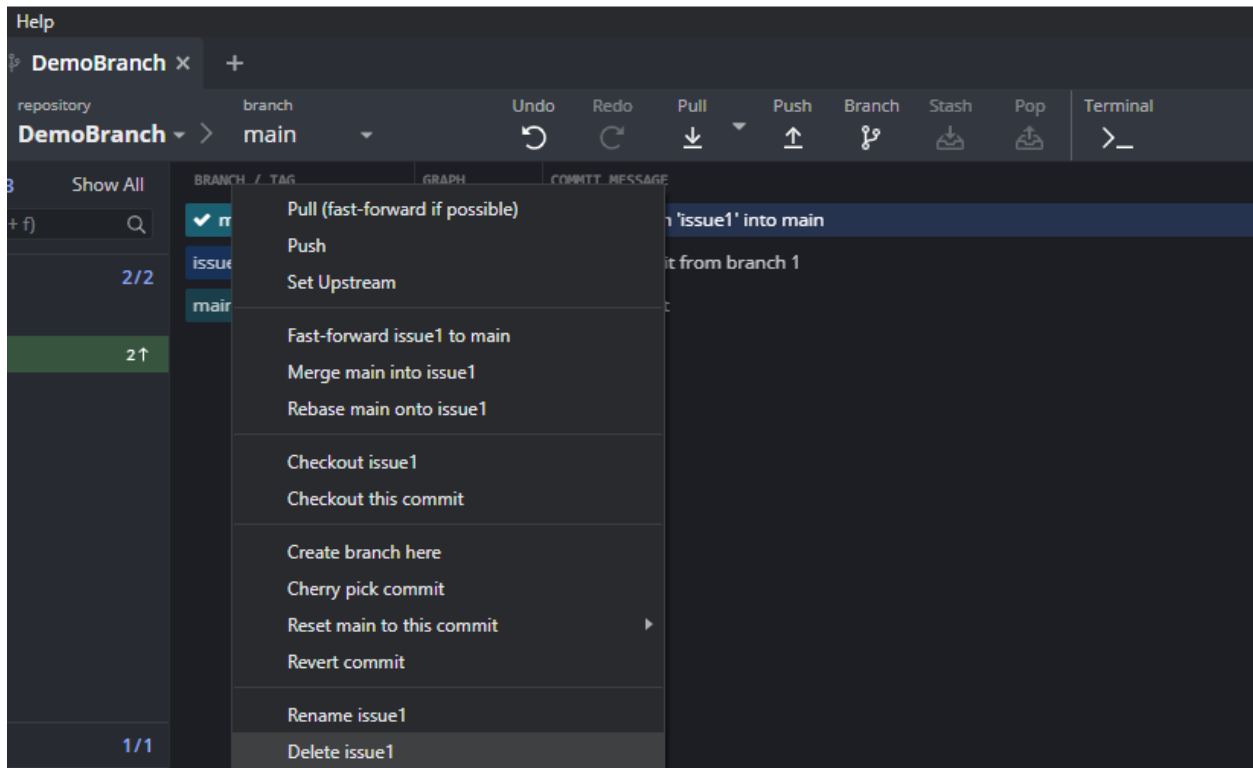
Do ở đây ta không có đổi gì ở main nên dạng merge này gọi là non fast-forward

Nếu fast-forward:



Bước 6: Xóa Branch

Để xóa branch cần checkout main (double click vào main), chọn “Delete issue1”



Kết quả:



Cuối cùng đơn giản PUSH lên GitHub