**Phần 14 Git – Merge**

Lệnh được sử dụng để kết hợp các thay đổi từ một hoặc nhiều nhánh vào nhánh hiện tại. Nó tích hợp lịch sử của các nhánh này, đảm bảo rằng tất cả các thay đổi đều được bao gồm và xung đột được giải quyết.

**git merge <tên nhánh>**

**Các bước để sáp nhập một chi nhánh**

Để đảm bảo quá trình hợp nhất diễn ra suôn sẻ, chúng ta cần thực hiện một loạt các bước hợp nhất bao gồm giải quyết mọi xung đột mà chúng ta có thể gặp phải. Dưới đây là các bước cơ bản liên quan.

**Bước 1: Tạo nhánh mới.**

Tạo một nhánh mới từ nhánh kho lưu trữ từ xa mà bạn muốn hợp nhất. Nếu gặp lỗi trong khi hợp nhất, chúng ta có thể quay lại phiên bản trước ngay lập tức.

**Bước 2: Đảm bảo luôn cập nhật những thay đổi mới nhất.**

Luôn đảm bảo rằng những thay đổi mới nhất được lấy từ cả hai nhánh như nhánh chính và nhánh bạn muốn hợp nhất trước khi hợp nhất.

**Bước 3: Giải quyết xung đột khi hợp nhất.**

Trong khi hợp nhất các nhánh, có thể một số xung đột hợp nhất sẽ được nêu ra, sau đó git sẽ nhắc bạn giải quyết xung đột hợp nhất. Nếu bất kỳ xung đột hợp nhất nào không được nêu ra, git sẽ tự động hợp nhất các nhánh.

**Bước 4: Mã đã hợp nhất cần được kiểm tra.**

Việc kiểm tra mã đã hợp nhất là điều cần thiết và chúng ta phải đảm bảo rằng mã không có lỗi nào và hoạt động bình thường. Để kiểm tra mã, chúng ta thực hiện tự động hoặc thủ công.

**Bước 5: Cam kết mã đã hợp nhất.**

Sau khi hoàn tất việc hợp nhất mã, nếu bạn hài lòng với công việc, hãy biết rằng đã đến lúc cam kết những thay đổi mới của mã vào nhánh mới.

**Bước 6: Đẩy nhánh đã hợp nhất.**

Cuối cùng, hãy làm cho các thành viên khác trong nhóm có thể truy cập nhánh mới bằng cách đẩy nó vào kho lưu trữ.

**Các biến thể**

git merge –commit: Tùy chọn này được sử dụng để thực hiện merge cùng với commit.

git merge –no-commit: Tùy chọn này thực hiện việc hợp nhất và dừng ngay trước khi tạo một cam kết hợp nhất, để cho người dùng có cơ hội kiểm tra kết quả hợp nhất trước khi cam kết.

git merge –edit: Tùy chọn này sẽ gọi trình soạn thảo trước khi xác nhận hợp nhất thành công và cho phép người dùng giải thích việc hợp nhất thay vì thông báo hợp nhất được tạo tự động.

git merge –no-edit: Câu lệnh này chỉ được dùng để chấp nhận các tin nhắn được tạo tự động.

git merge -m “msg”: Lệnh này thiết lập thông điệp cam kết sẽ được sử dụng cho cam kết hợp nhất.

git merge -ff: Khi quá trình hợp nhất được giải quyết theo dạng chuyển tiếp nhanh, nó chỉ cập nhật con trỏ nhánh mà không tạo ra cam kết hợp nhất.

git merge -no-ff: Trái ngược với hành vi mặc định, nó tạo ra một cam kết hợp nhất ngay cả khi quá trình hợp nhất được giải quyết dưới dạng chuyển tiếp nhanh.

**Các bước để hợp nhất nhánh Git vào nhánh Master**

Trước khi hợp nhất, hãy đảm bảo rằng thư mục làm việc của bạn sạch sẽ và nhánh của bạn được cập nhật.

Bước 1. Kiểm tra nhánh chính

**git checkout master**

Bước 2. Kéo những thay đổi mới nhất

Lấy và tích hợp những thay đổi mới nhất từ ​​kho lưu trữ từ xa.

**git pull origin master**

Bước 3. Chuyển sang nhánh bạn muốn hợp nhất vào nhánh chính.

**git checkout your-feature-branch**

Bước 4. Đảm bảo Branch được cập nhật

Cập nhật nhánh tính năng của bạn với những thay đổi mới nhất từ ​​nhánh chính để giảm thiểu xung đột.

**git pull origin master**

Bước 5. Chuyển lại Master

Quay lại nhánh chính để chuẩn bị hợp nhất.

**git checkout master**

Bước 6. Hợp nhất nhánh tính năng

Gộp các thay đổi từ nhánh tính năng vào nhánh chính.

**git merge your-feature-branch**

Bước 7. Giải quyết xung đột (nếu có)

Nếu Git báo cáo xung đột, bạn sẽ cần giải quyết chúng theo cách thủ công. Mở các tệp có xung đột, giải quyết các vấn đề, sau đó đánh dấu chúng là đã giải quyết.

**git add resolved-file**

Sau khi giải quyết hết mọi xung đột, hãy hoàn tất việc hợp nhất bằng:

**git commit**

Bước 8. Đẩy các thay đổi vào kho lưu trữ từ xa

Cuối cùng, đẩy những thay đổi đã hợp nhất vào kho lưu trữ từ xa.

**git push origin master**

**Các bước để thay thế nhánh chính**

**Bước 1: Đảm bảo thư mục làm việc của bạn sạch sẽ**

Trước khi thực hiện bất kỳ thay đổi nào, hãy đảm bảo thư mục làm việc của bạn sạch sẽ. Điều này có nghĩa là cam kết hoặc lưu trữ bất kỳ thay đổi nào chưa cam kết.

**git status**

Nếu có những thay đổi chưa được cam kết, hãy cam kết chúng:

**git add .**

**git commit -m "Lưu thay đổi trước khi thay thế nhánh master"**

Hoặc cất giấu chúng:

**git stash**

**Bước 2: Chuyển sang nhánh bạn muốn thay thế nhánh chính**

Kiểm tra nhánh mà bạn muốn sử dụng để thay thế nhánh chính.

**git checkout feature-branch**

**Bước 3: Cập nhật nhánh để được cập nhật**

Đảm bảo nhánh của bạn được cập nhật những thay đổi mới nhất từ ​​kho lưu trữ từ xa.

**git pull origin feature-branch**

**Bước 4: Xóa nhánh Master cục bộ**

Tiếp theo, chuyển sang nhánh bạn muốn giữ lại và xóa nhánh chính cục bộ.

**git branch -D master**

**Bước 5: Đổi tên nhánh hiện tại thành Master**

Đổi tên nhánh hiện tại (feature-branch) thành master.

**git branch -m master**

**Bước 6: Đẩy nhánh Master mới lên Remote**

Cuối cùng, đẩy nhánh master mới vào kho lưu trữ từ xa. Hãy rất cẩn thận với bước này vì nó sẽ ghi đè lên nhánh master từ xa.

**git push origin master --force**

**Bước 7: Xác minh các thay đổi**

Để đảm bảo mọi việc diễn ra suôn sẻ, bạn nên kiểm tra xem nhánh chính từ xa đã được cập nhật chưa.

**git fetch origin**

**git checkout master**

**git log**