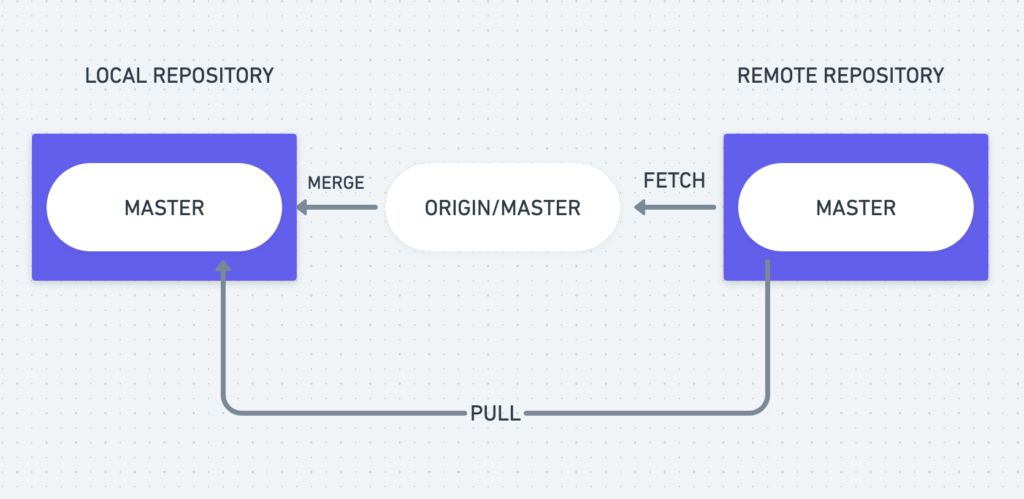
**Phần 3: Git Pull**

Git Pull là một lệnh dùng để tải xuống dữ liệu từ một Remote repository và cập nhật Local repository phù hợp với dữ liệu đó. Nói cách khác, Git Pull là lệnh hợp nhất các thay đổi từ Remote repository vào Local repository.

Về bản chất, Git Pull chính là sự kết hợp của 2 lệnh Git Fetch và Git Merge. Giai đoạn đầu, Git Pull sẽ thực thi lệnh Git Fetch ở phạm vi nhánh cục bộ mà HEAD được trỏ đến. Khi dữ liệu được tải xuống, Git Pull sẽ bắt đầu quy trình hợp nhất như Git Merge. Một merge commit mới sẽ được tạo và HEAD cũng được cập nhật để trỏ đến merge commit đó.

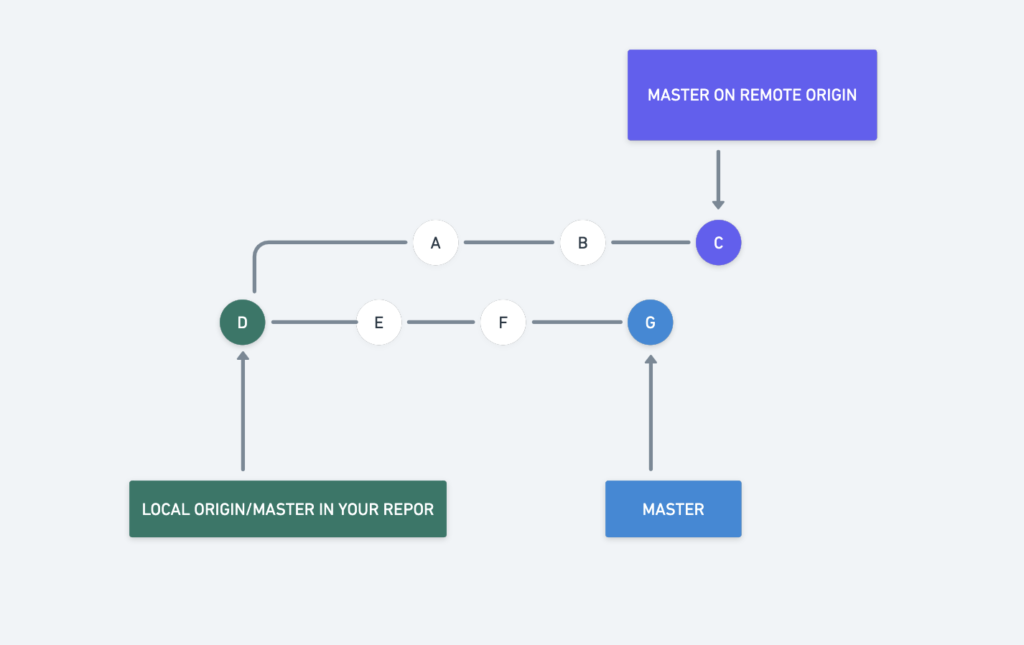
Git Pull là sự kết hợp của 2 lệnh Git fetch và Git Merge

**Mô hình hoạt động của Git Pull**

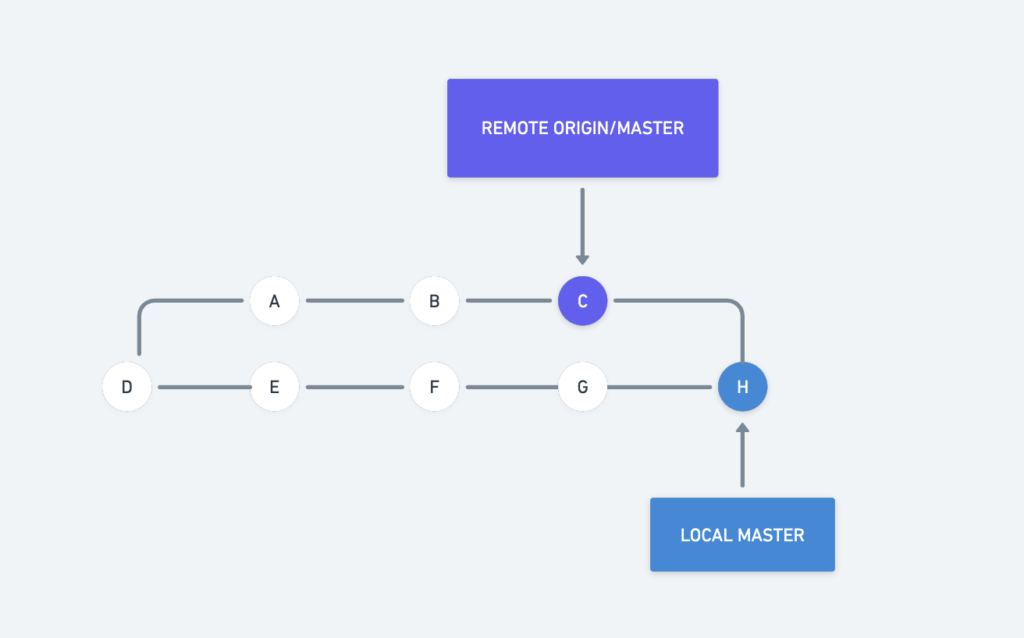
Sau khi tìm hiểu Git Pull là gì, ở phần này chúng ta sẽ khám phá mô hình hoạt động của Git Pull. Về cơ bản, Git Pull sẽ hoạt động theo một quy trình cơ bản dưới đây:

* Thực hiện lệnh Git Fetch để tải xuống dữ liệu từ Remote repository được chỉ định.
* Chạy lệnh Git Merge để hợp nhất các nội dung tham chiếu từ xa và hướng về một local merge commit mới.

Để hiểu rõ hơn mô hình hoạt động của Git Pull, chúng ta sẽ phân tích một ví dụ sau:

Mô hình hoạt động của Git Pull

Giả sử, chúng ta có một repository gồm một nhánh chính (master branch) và một nguồn từ xa (remote origin). Git Pull sẽ tải xuống tất cả các thay đổi từ vị trí mà local và master chuyển hướng. Trong mô hình trên, vị trí tải xuống là E. Git Pull sẽ thực thi lệnh Git Fetch để “nạp” các remote commit được chuyển hướng tại A-B-C. Sau đó, local merge commit mới (H) được tạo ra có nội dung đồng bộ với các remote commit A-B-C.

Theo sơ đồ, H chính là một merge commit mới

Câu lệnh Git Pull dùng để đồng bộ Remote repository vào Local repository thường có cấu trúc sau:

git pull origin branch

Trong đó, origin là tên Remote repository và branch là nhánh muốn đồng bộ.

Ví dụ, để đồng bộ master branch từ Remote repository vào Local repository, chúng ta sẽ dùng lệnh:

git pull origin master

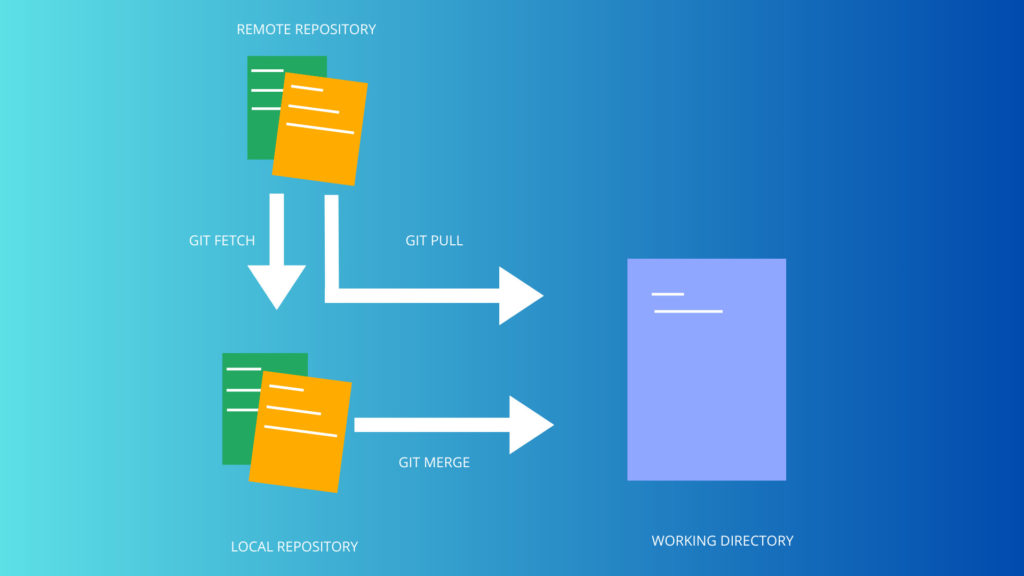
**Các tùy chọn phổ biến trong Git Pull**

Giống như bất kỳ lệnh nào khác trong [Git](https://vn.got-it.ai/blog/git-la-gi-kien-thuc-ve-git-cac-lap-trinh-vien-phai-biet), lệnh Git Pull cũng có một số tùy chọn giúp sử dụng lệnh một cách tự nhiên và hiệu quả.

* **git pull <remote>**: Lấy bản sao remote đã được chỉ định của nhánh hiện tại và hợp nhất nó vào bản sao local. Tùy chọn này giống như lệnh: git fetch origin/git merge origin .
* **git pull –no-commit**: Đây là một tùy chọn Git Pull không có cam kết. Thực hiện tìm nạp nội dung remote nhưng không tạo ra merge commit mới.
* **git pull –rebase**: Tùy chọn này là [Git Rebase](https://vn.got-it.ai/blog/git-rebase-la-gi-khi-nao-nen-su-dung-git-rebase) giúp hợp nhất remote branch với local branch nhanh chóng.
* **git pull –verbose**: Cung cấp verbose output để hiển thị nội dung tải xuống và các chi tiết hợp nhất.

**Sự khác biệt giữa Git Pull và Git Fetch**

Trong Git, Git Pull và Git Fetch là 2 lệnh có chức năng tương đồng với nhau. Cả hai đều được sử dụng để tải xuống dữ liệu mới từ một Remote repository. Tuy nhiên, Git Fetch thường được coi là một phiên bản an toàn hơn của Git Pull.

Git Pull và Git Fetch đều tải xuống dữ liệu từ Remote repository

Git Fetch chỉ tải xuống nội dung từ Remote repository mà không làm thay đổi trạng thái của Local repository. Trong khi đó, Git Pull sẽ tải xuống nội dung và cố gắng thay đổi trạng thái của Local repository cho phù hợp với nội dung đó. Điều này có thể dẫn đến hiện tượng xung đột hợp nhất (merge-conflict) trong Git.

Việc sử dụng Git Fetch thường được dùng trong trường hợp có nhiều người làm việc trên cùng một nhánh. Còn Git Pull chỉ nên sử dụng khi có một người làm việc trên nhánh để hạn chế xung đột. Bạn chỉ nên dùng lệnh Git Pull trên một thư mục làm việc sạch (không có thay đổi đã cam kết).