**BÀI THỰC HÀNH GitHub, SourceTree 3**

***Chủ đề*: Quản lý source code với Git, GitHub, SourceTree**

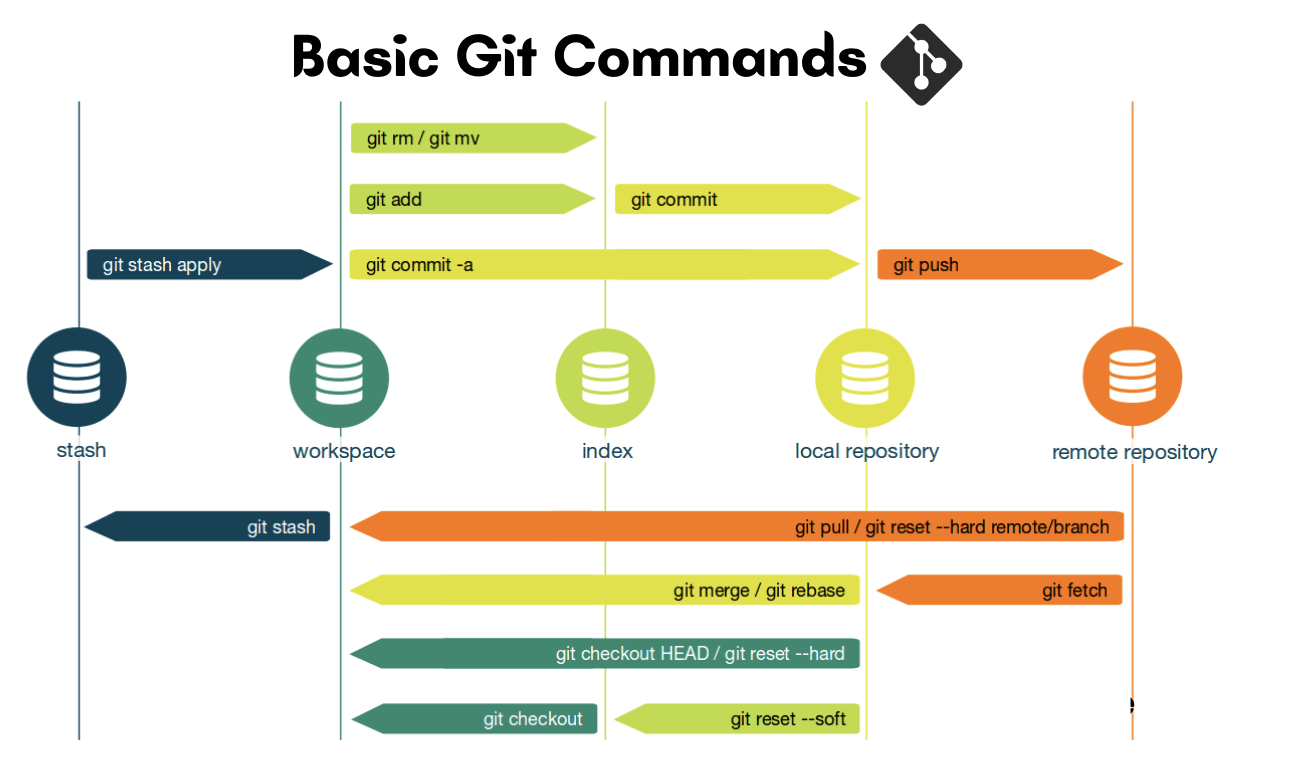
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**I. Mục tiêu**

* Nắm vững quy trình làm việc với Git và GitHub thông qua giao diện SourceTree.
* Có khả năng tạo, quản lý, và chia sẻ dự án trên GitHub.
* Hiểu và thực hành được các chức năng khi làm việc với Source Tree để quản lý source code:
  + Tạo Stash, apply Stash
  + Tạo pull request.
  + Reverse commit
  + Reset commit
  + Đánh tag version for release

**II. Nội dung thực hành**

1. **Các tác vụ cơ bản với Git:**

****

Bài thực hành hôm nay chúng ta sẽ đi sâu hơn những chức năng nâng cao trong việc quản lý source code với GitHub, SourceTree

* 1. **Save Stash và Apply Stash:**

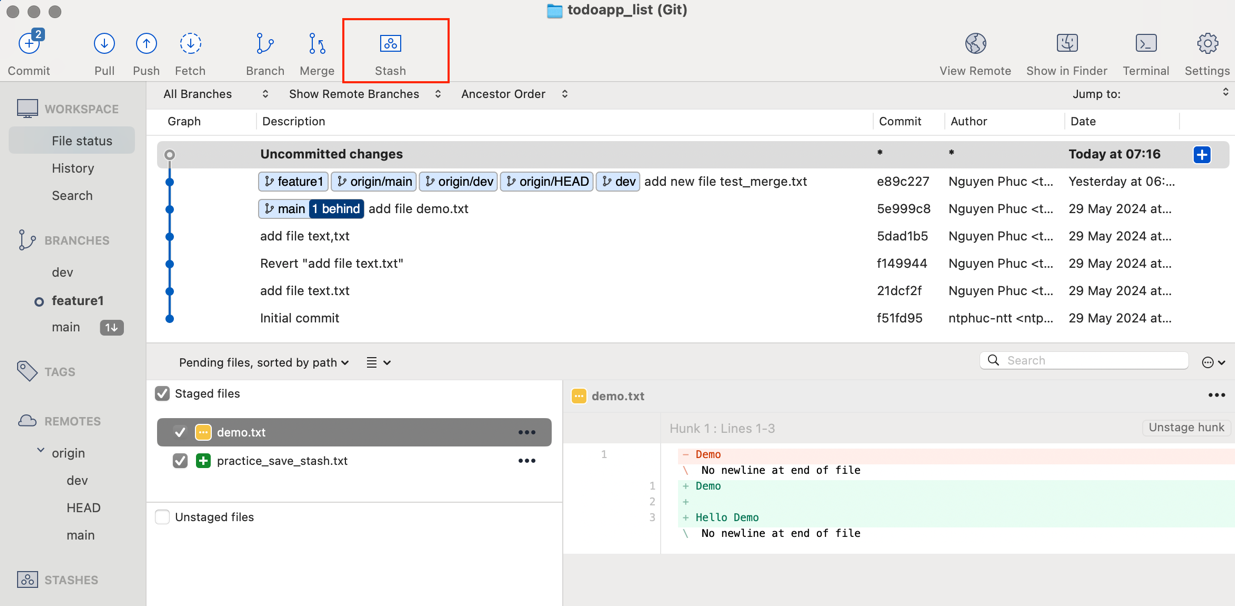
Ngữ cảnh:

Trong working space, có những sự thay đổi. Tuy nhiên vì người trong team của mình đã push lên những thay đổi mới và mình buộc phải cập nhật những thay đổi của họ và working space của mình để có những thay đổi mới nhất để tiếp tục phần chỉnh sửa của mình. Trong tình huống này, chúng ta lưu lại những thay đổi của mình ở dưới local trước. SourceTree sẽ hỗ trợ cho mình việc này bằng cách lưu vào Stashes.

Cách làm:

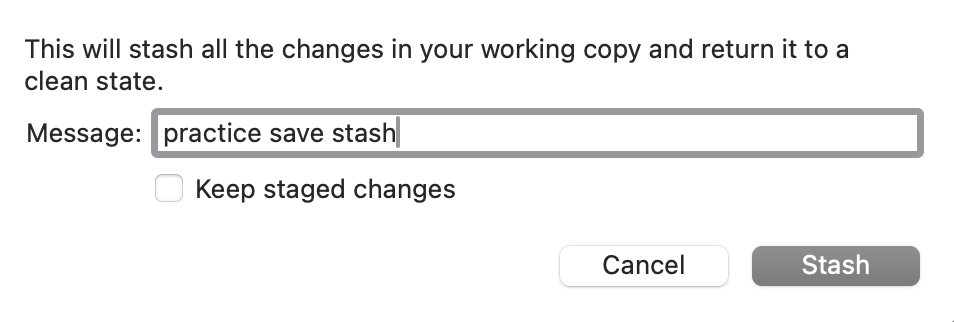
* **Bước 1: Thêm mới sự thay đổi**

Trong working space, tạo or checkout nhánh feature1 thêm một file mới tên là practice\_save\_stash.txt với nội dung là “practice save stash”, trong file demo thay đổi nội dung trong file thêm 1 dòng chữ “Hello demo”. Quan sát sự thay đổi trong repository trên SourceTree. Add những sự thay đổi này vào khung Staged như hình bên dưới.

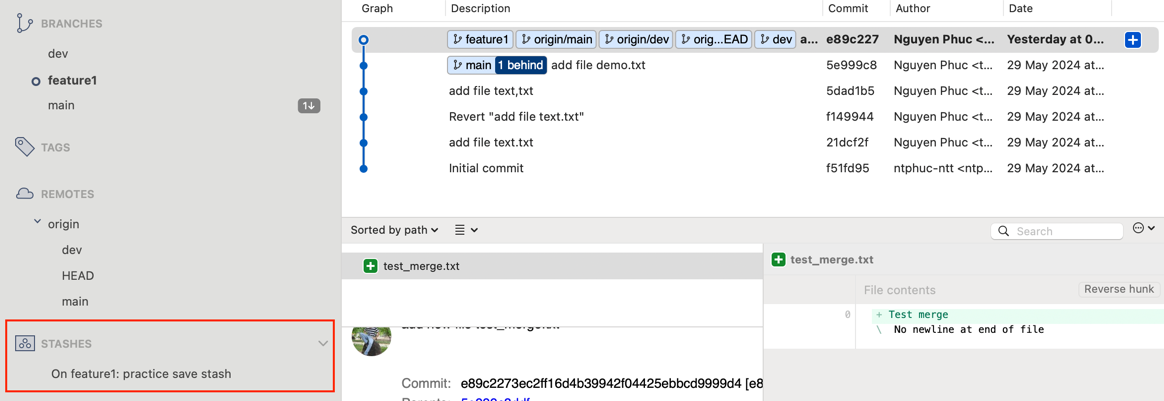


* **Bước 2: Save thay đổi và Stashes**

Sau đó click vào button Stash trên thanh tool bar và input nội dung để save thay đổi này vào Stashes

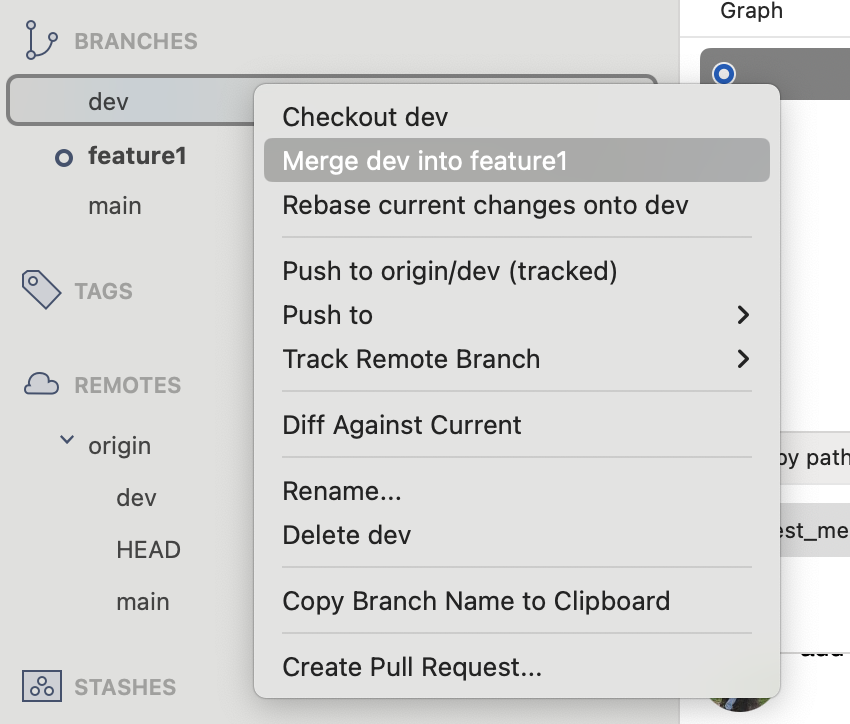


Sau đó mình sẽ thấy xuất hiện một file trong giao diện SourceTree.



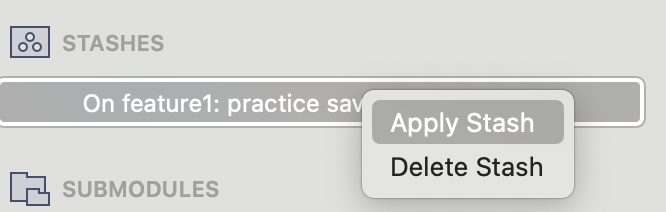
* **Bước 3: Cập nhật thay đổi mới từ remote repository và merge thay đổi vào nhánh hiện tại**

Check out nhánh dev -> Pull changes nhánh dev về -> Check out lại nhánh feature1 -> chọn nhánh dev click chuột phải và chọn Merge dev into feature1



* **Bước 4: Apply stash**

Apply stash đã save ở bước 2 vào nhánh hiện tại để tiếp tục công việc của mình.



* 1. **Tạo Pull request:**

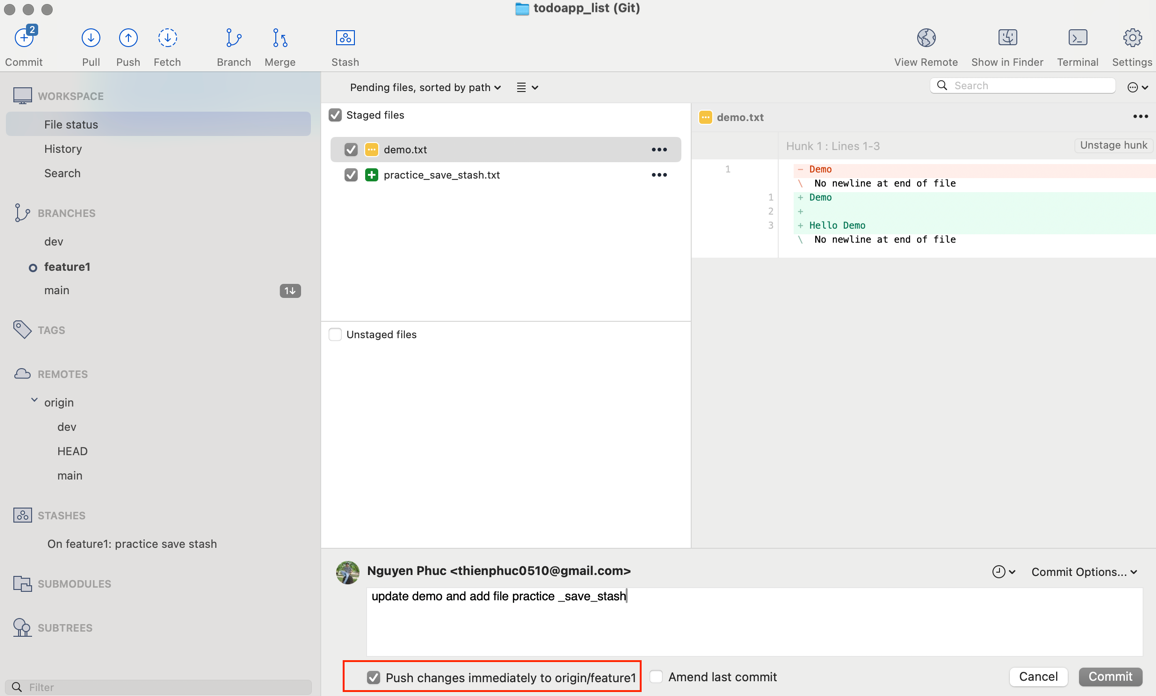
Ngữ cảnh: Việc này rất quan trọng trong việc làm nhóm, ý nghĩa là sau khi bạn push những sự thay đổi của mình lên remote repository. Để sự thay đổi của mình được merge vào nhánh chính thì cần phải có một bước là tạo Pull request để các team member khác review những sự thay đổi của mình, để làm được việc này thì cách làm như sau.

Cách làm:

* **Bước 1: Push những sự thay đổi lên remote repository**

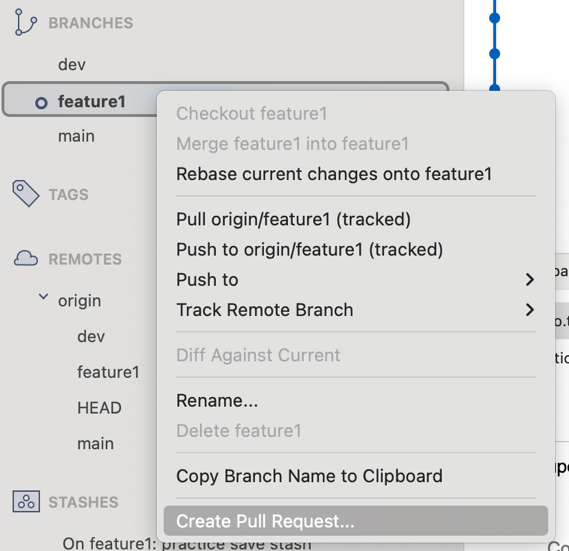
Ở phần 1 chúng ta đã tạo sự thay đổi trong file demo và add file practice\_save\_stash. Ở bước này chúng ta sẽ push sự thay đổi này lên remote repository.

Click button Commit trên thanh tool bar và check Push change…, nhập message để mô tả cái commit này.



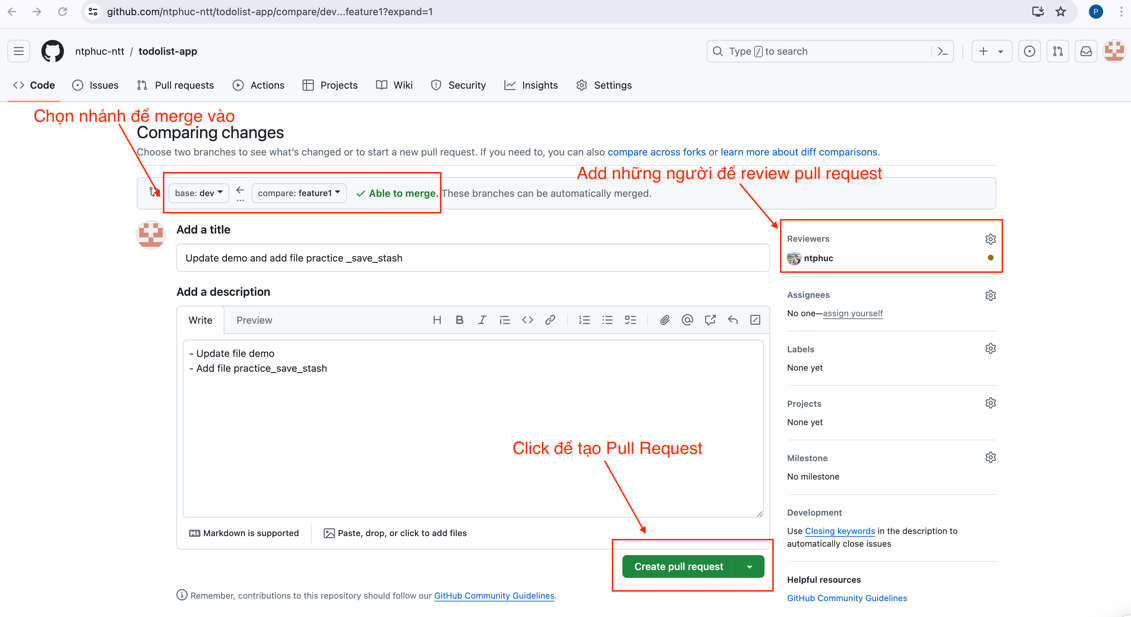
* **Bước 2: Tạo Pull Request**

Click chuột phải vào nhánh feature1 chọn Create Pull Request

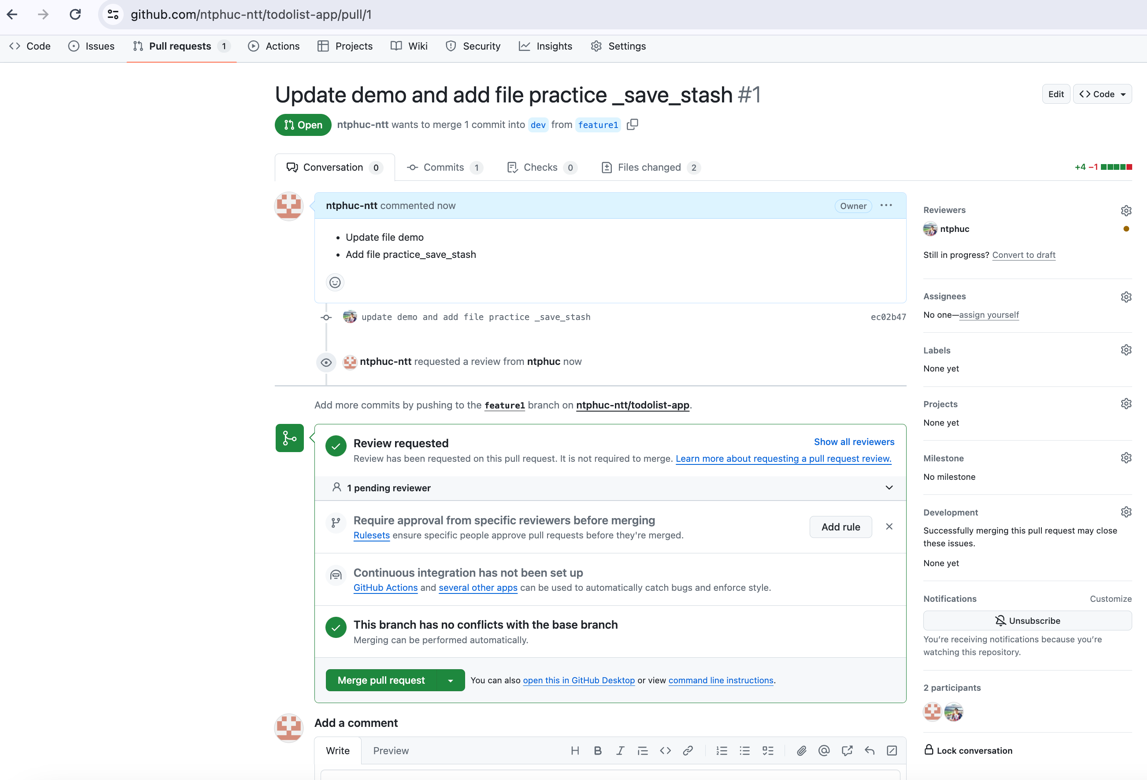
****

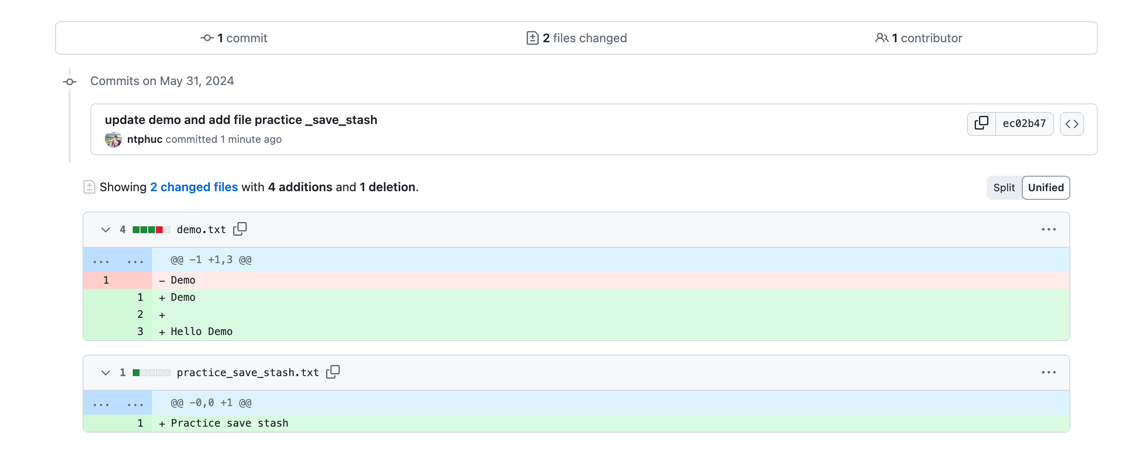
Sau khi click Create Pull Request, thì nó sẽ tự động mở trang web với nội dung như bên dưới.

* Chọn nhánh muốn merge vào, thông thường là nhánh dev
* Nhập title, description
* Chọn những người review pull request
* Click Create pull request

****

Sau khi tạo pull request, chúng ta sẽ nhìn thấy một màn hình như sau:

****

****

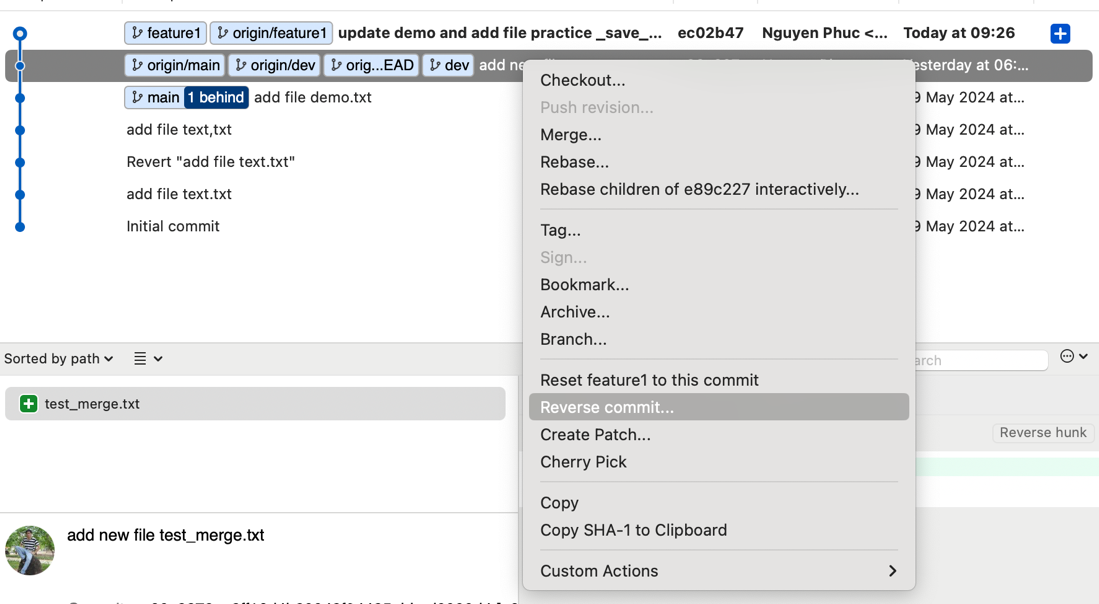
* 1. **Reverse commit:**

Ngữ cảnh: Trong quá trình push lên những thay đổi qua nhiều commit, nhưng giả sử chúng ta cần loại bỏ một commit trong số đó thì chúng ta sẽ thực hiện lênh revert code tại cái commit đó.

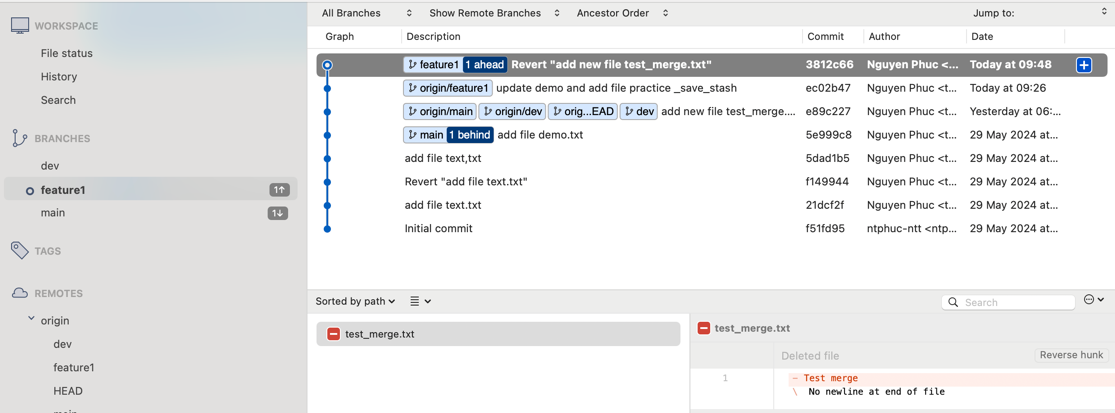
Cách làm: Trong các bước sau đây chúng ta sẽ thực hiện lệnh revert code tại commit “add new file test\_merge”. Cụ thể các bước sẽ làm như sau:

* **Bước 1: Chọn commit cần loại bỏ và reverse commit**

Chọn commit cần loại bỏ, click chuột phải và chọn Reverse commit…

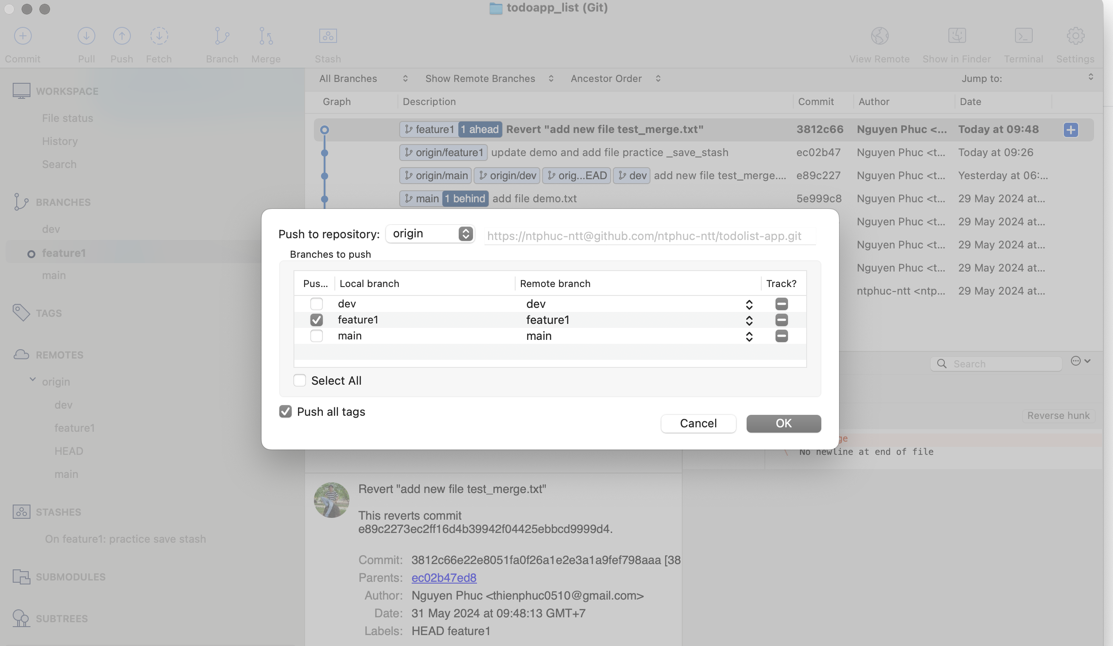


Sau khi xác nhận là reverse this commit, chúng ta sẽ thấy commit mới là Revert “add new file…” với file test\_merge đã bị remove. Bởi vì commit trước đó là add file test\_merge, nên sau khi revert commit này thì kể như những thay đổi của commit này sẽ không còn nữa.



* **Bước 2: Push commit**

Sau đó push commit này lên remote repository:

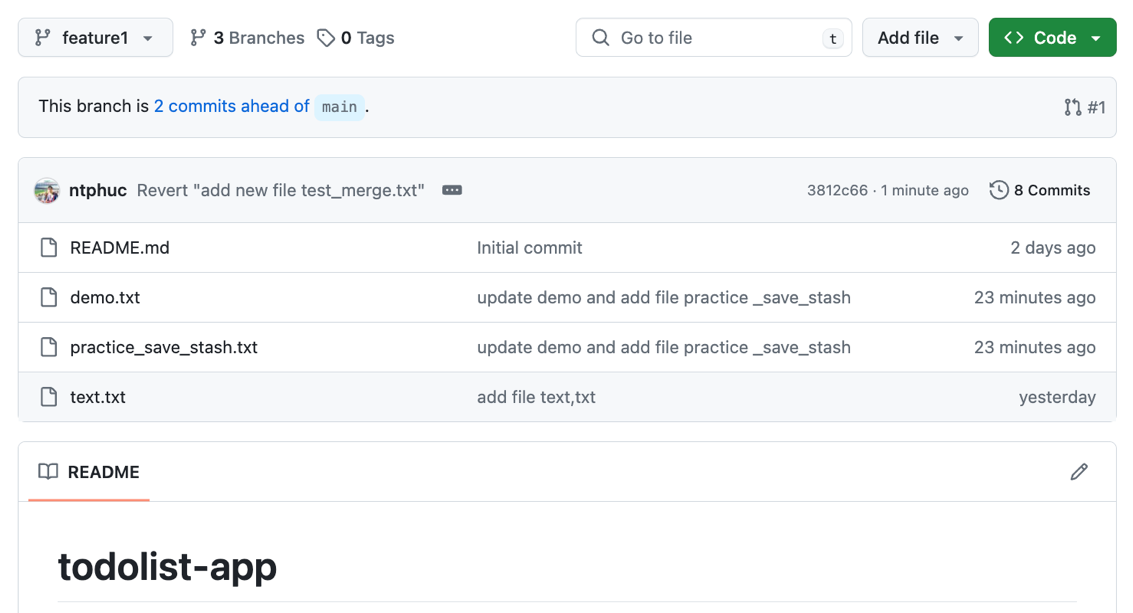
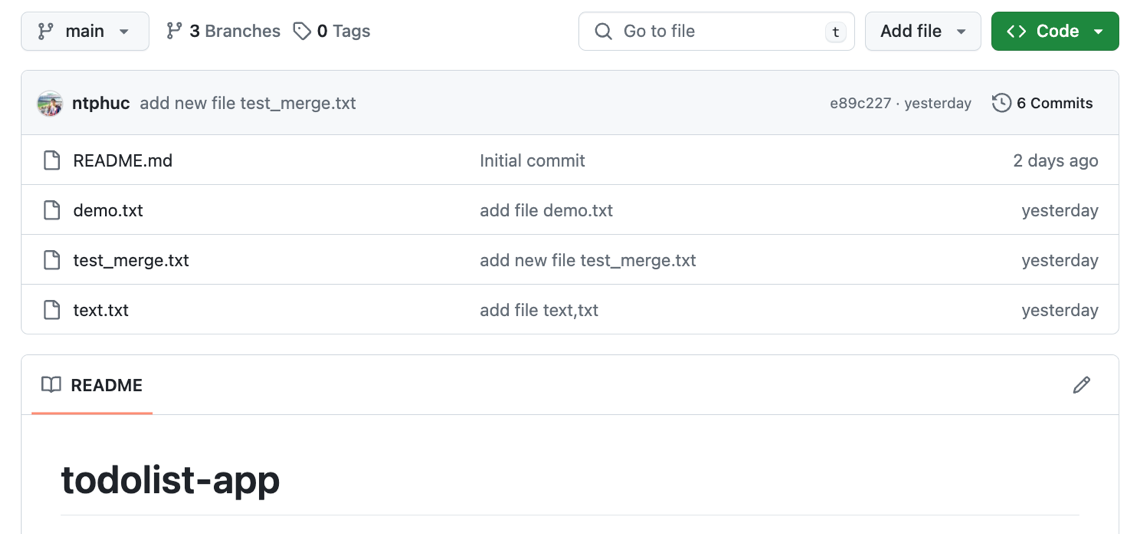


* **Bước 2: Kiểm tra kết quả**

Kiểm tra lại kết quả nhánh main và nhánh feature1:

Chúng ta thấy có sự khác nhau:

* Nhánh main: có file test\_merge
* Nhánh feature1: không có file test\_merge bởi vì chúng ta đã reverse commit này



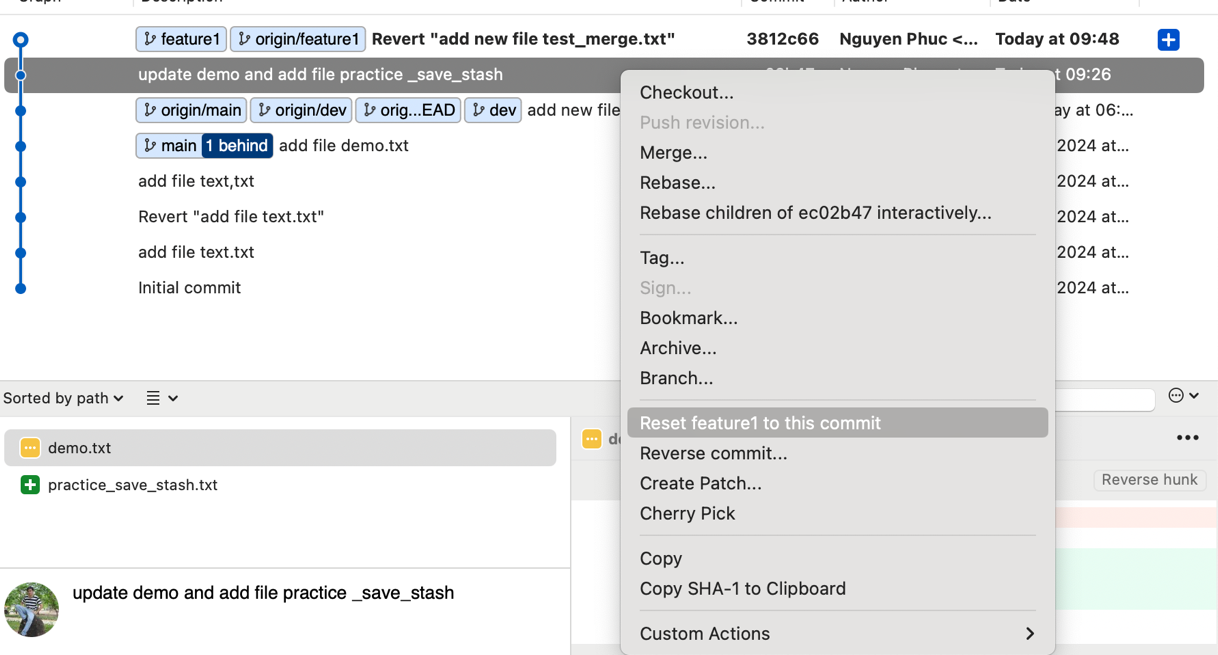
* 1. **Reset commit:**

Ngữ cảnh:

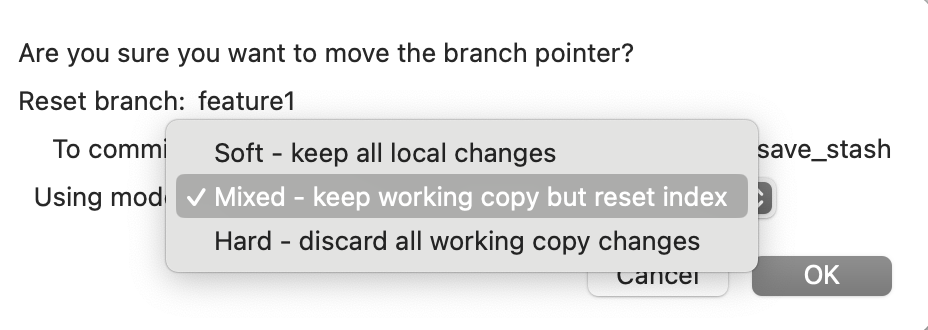
Trong một số trường hợp ví dụ như chúng ta có những commit sai, chúng ta cần khôi phục lại những commit này, thì chúng ta sẽ sử dụng reset commit.

Cách làm:

Chọn commit cần để reset.



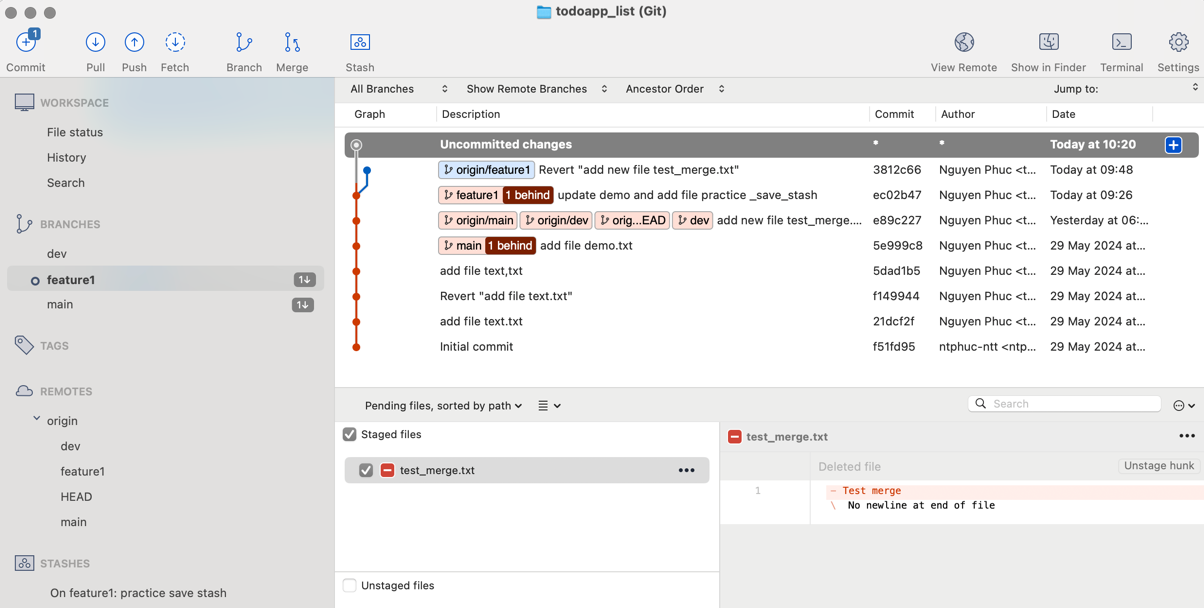
Chúng ta sẽ thấy những options để chọn cho việc reset commit:



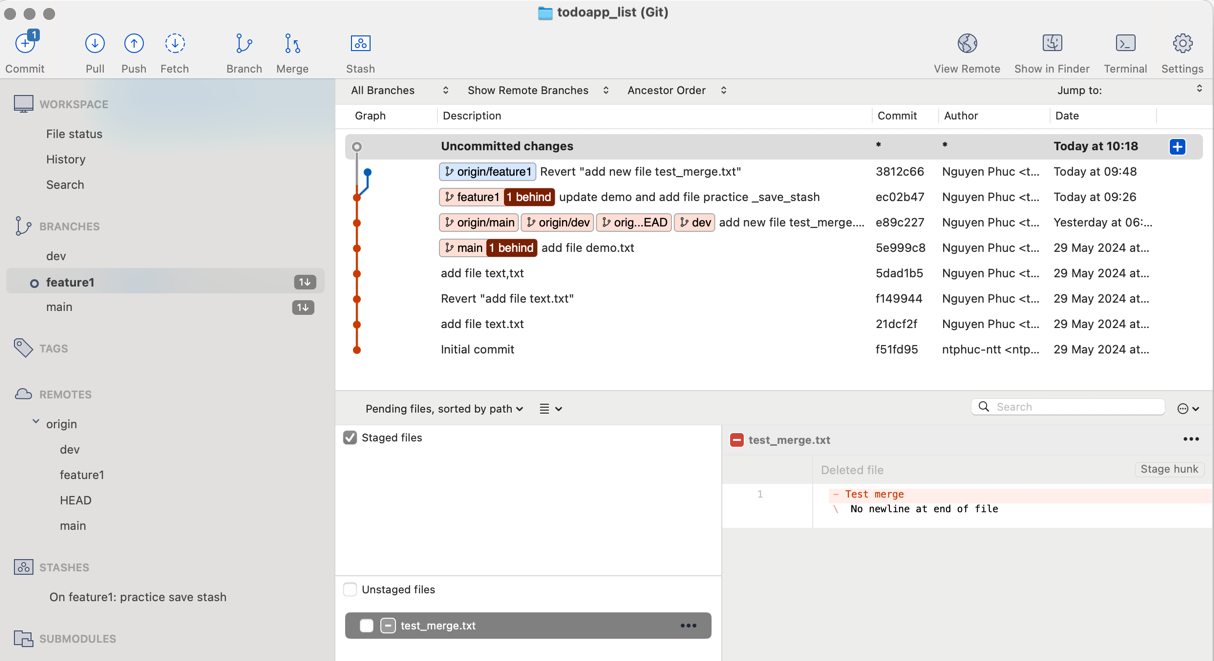
1. **Hard reset:**
   * Khi bạn thực hiện hard reset, Git sẽ đặt lại trạng thái của HEAD (commit hiện tại) và thay đổi working directory và staging area để phản ánh trạng thái của commit đã chọn.
   * Tất cả các thay đổi trong working directory và staging area sẽ bị mất và được thay thế bởi các thay đổi từ commit đã chọn.
   * Hard reset sử dụng khi bạn muốn loại bỏ tất cả các thay đổi và đưa repository về trạng thái của một commit cụ thể.
2. **Soft reset:**
   * Soft reset giữ lại các thay đổi trong working directory và staging area sau khi reset commit.
   * Các thay đổi từ commit đã chọn sẽ được thêm vào staging area để chuẩn bị cho commit mới nhất.
   * Soft reset sử dụng khi bạn muốn "undo" một commit mà không mất đi các thay đổi trong working directory và staging area, giúp bạn chỉnh sửa lại commit hoặc thực hiện lại commit với các thay đổi mới.
3. **Mixed reset:**
   * Mixed reset giữ lại các thay đổi trong working directory sau khi reset commit, nhưng loại bỏ các thay đổi trong staging area.
   * Các thay đổi từ commit đã chọn sẽ không được thêm vào staging area, nhưng vẫn được giữ lại trong working directory để bạn có thể commit lại sau khi đã xem xét.
   * Mixed reset thường được sử dụng khi bạn muốn loại bỏ các thay đổi đã được thêm vào staging area mà không ảnh hưởng đến các thay đổi trong working directory.

* Khuyến nghị xài Soft Reset để chúng ta có thể kiểm soát những file nào cần giữ và cần loại bỏ.

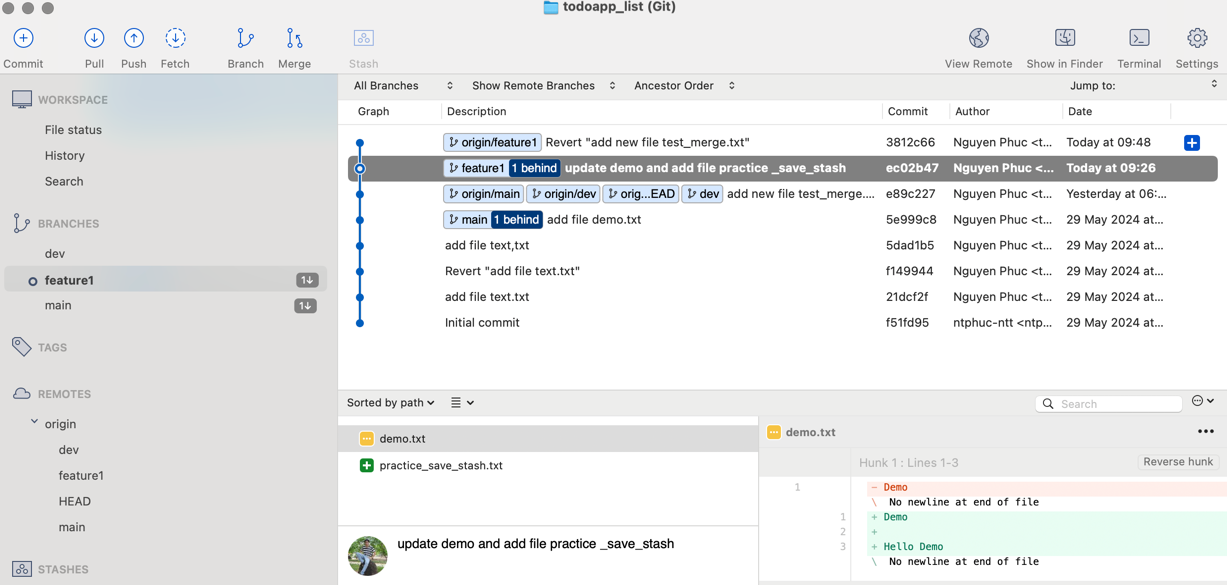
Sau khi thực hiện Soft Reset:



Sau khi thực hiện Mixed Reset:



Sau khi thực hiện Hard Reset: chúng ta sẽ không thấy những thay đổi nào trong Staged files và Unstaged files.



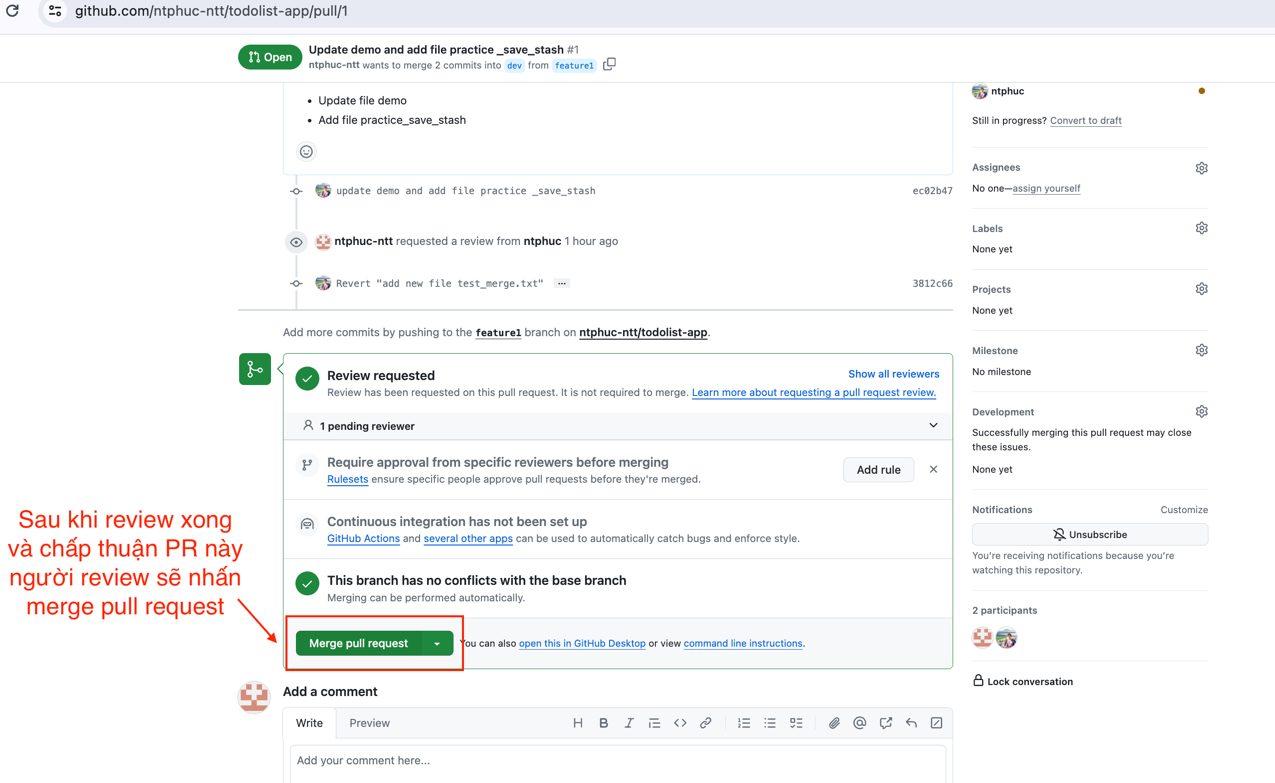
* 1. **Đánh tag version cho release:**

Ngữ cảnh: Sau khi các team member review những Pull Request và merge vào nhánh dev thì chuẩn bị cho release sản phẩm, thì chúng ta cần đánh tag version cho bản build này tại một commit này.

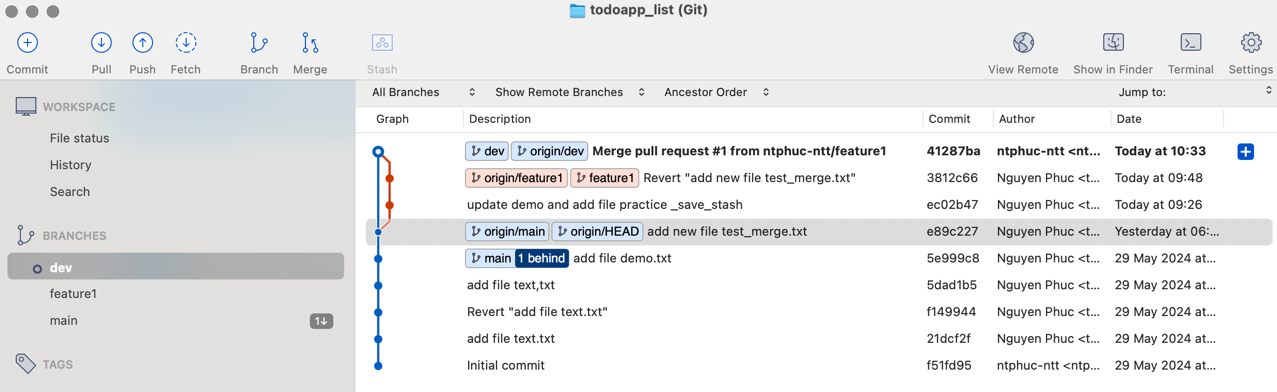
Cách làm:

Sau khi chúng ta review những Pull Request xong hết cho một release, thì chúng ta cần merge nó vào nhánh dev để chuẩn bị đánh tag version release cho commit build.

Lưu ý: việc đánh tag này cũng ko cần nhất thiết phải review PR và merge vào dev. Nhưng đúng qui chuẩn khi làm việc nhóm thì chúng ta cần thiết để review code trong các PR của các team members mình làm để đảm bảo chất lượng code phát triển sau này.

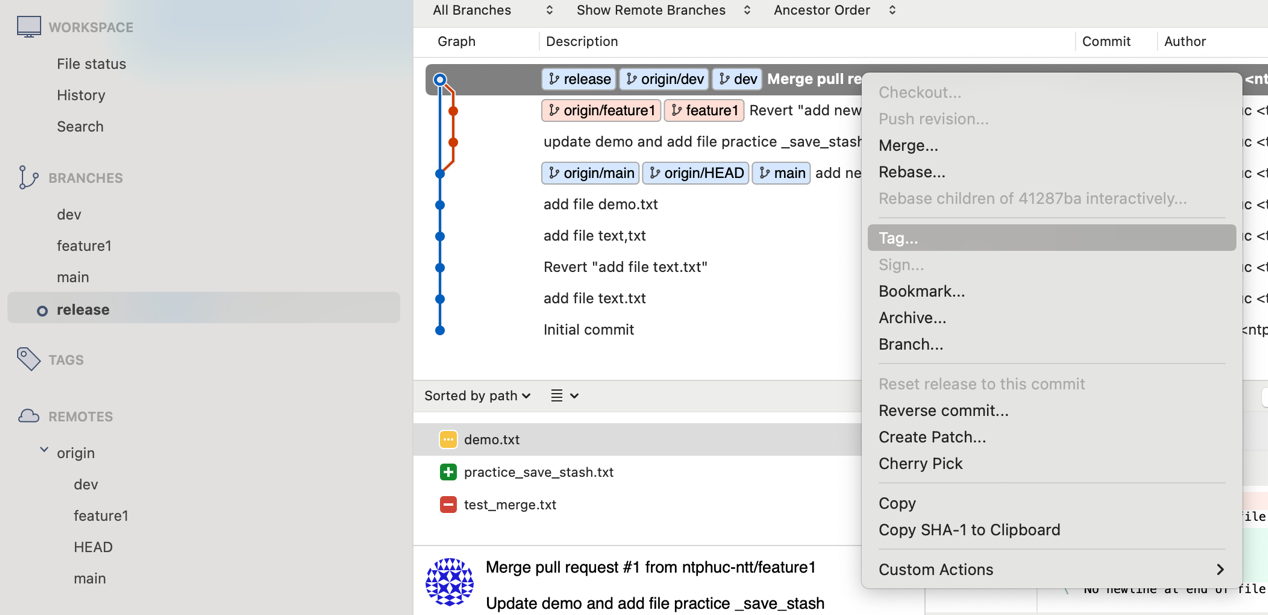


Bước kế tiếp ở SourceTree chúng ta sẽ Check out nhánh dev -> Pull những thay đổi mới nhất về

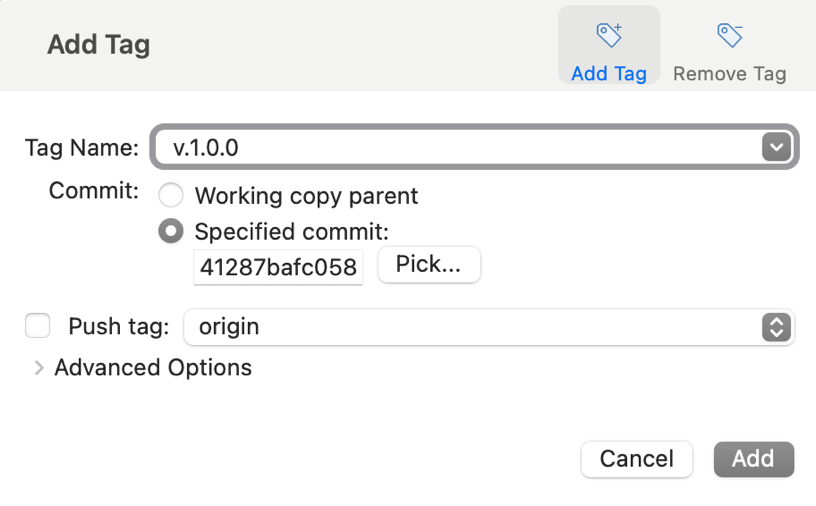


Tạo một nhánh release, chọn commit muốn đánh tag version, click chuột

phải chọn Tag.



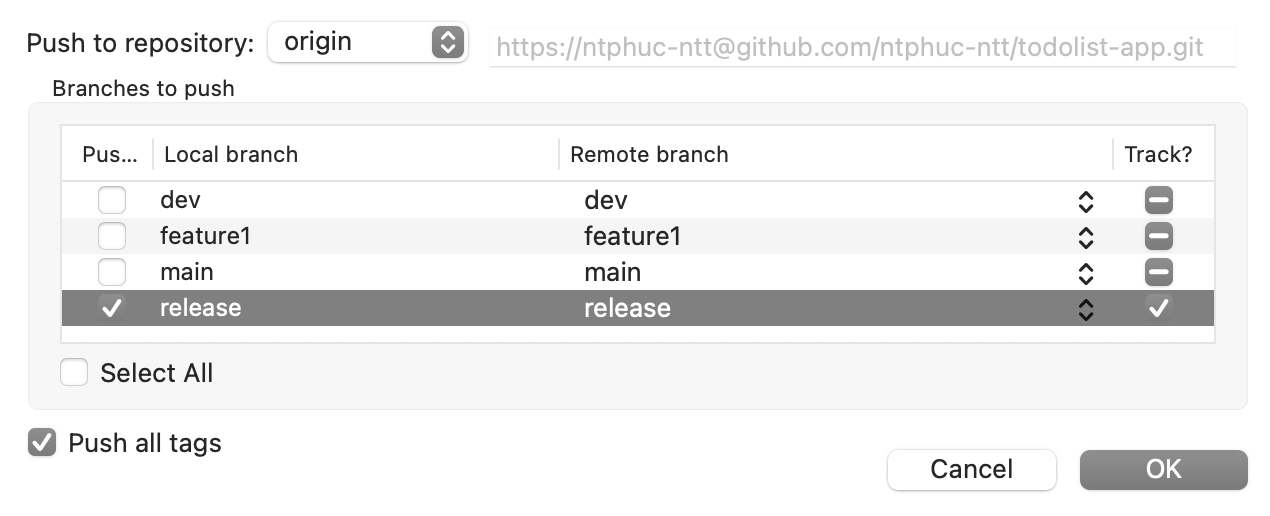
Nhập tên version và click button Add:



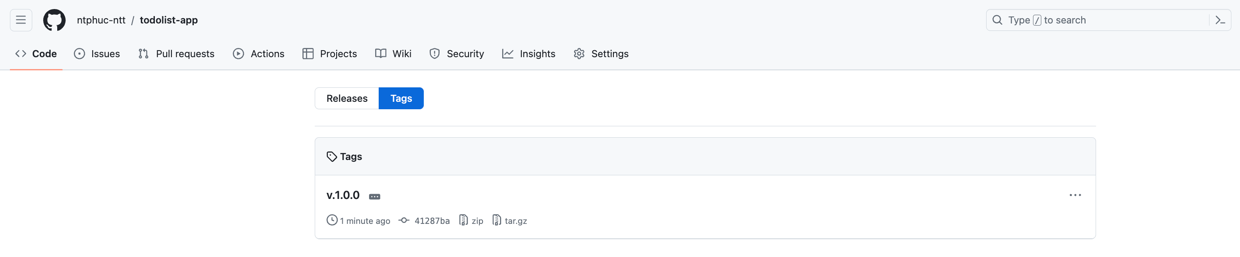
Sau đó chúng ta sẽ thấy version mới tạo xuất hiện dưới khung Tag:



Push nhánh Release và check Push all tags lên remote repository



Kiểm tra kết quả trên web



1. **Bài tập thực hành:**
   1. Tạo một nhánh mới feature2, thực hiện thay đổi nội dung working space, sau đó save thay đổi này vào stashed. Check out nhánh dev, pull nhánh dev về. Merge nhánh dev vào nhánh feature2. Rồi apply thay đổi đã lưu ở stash.
   2. Tạo thay đổi trên nhánh feature2, sau đó thực hiện lệnh commit, push thay đổi này lên remote repository. Rồi sau đó tạo một Pull Request để assign team member khác vào review những thay đổi mình mới tạo.
   3. Chọn 1 commit cách commit hiện tại khoảng 2 commit, thực hiện lệnh Reverse commit tại commit này -> push lên remote repository và quan sát sự thay đổi của commit trước và sau khi thực hiện lệnh Reverse commit.
   4. Chọn 1 commit cách commit hiện tại khoảng 2 commit, thực hiện lệnh Reset commit tại commit này -> chọn các mode Soft Reset, Mixed Reset,

Hard Reset để quan sát sự khác nhau của options này.

* 1. Approve Pull Request đã tạo ở bài tập 2. Ở SourceTree, check out nhánh dev -> pull nhánh dev -> tạo một nhánh mới release. Sau đó đánh tag version 1.1.0 cho bản build release này.