**BÀI THỰC HÀNH GitHub, SourceTree 2**

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

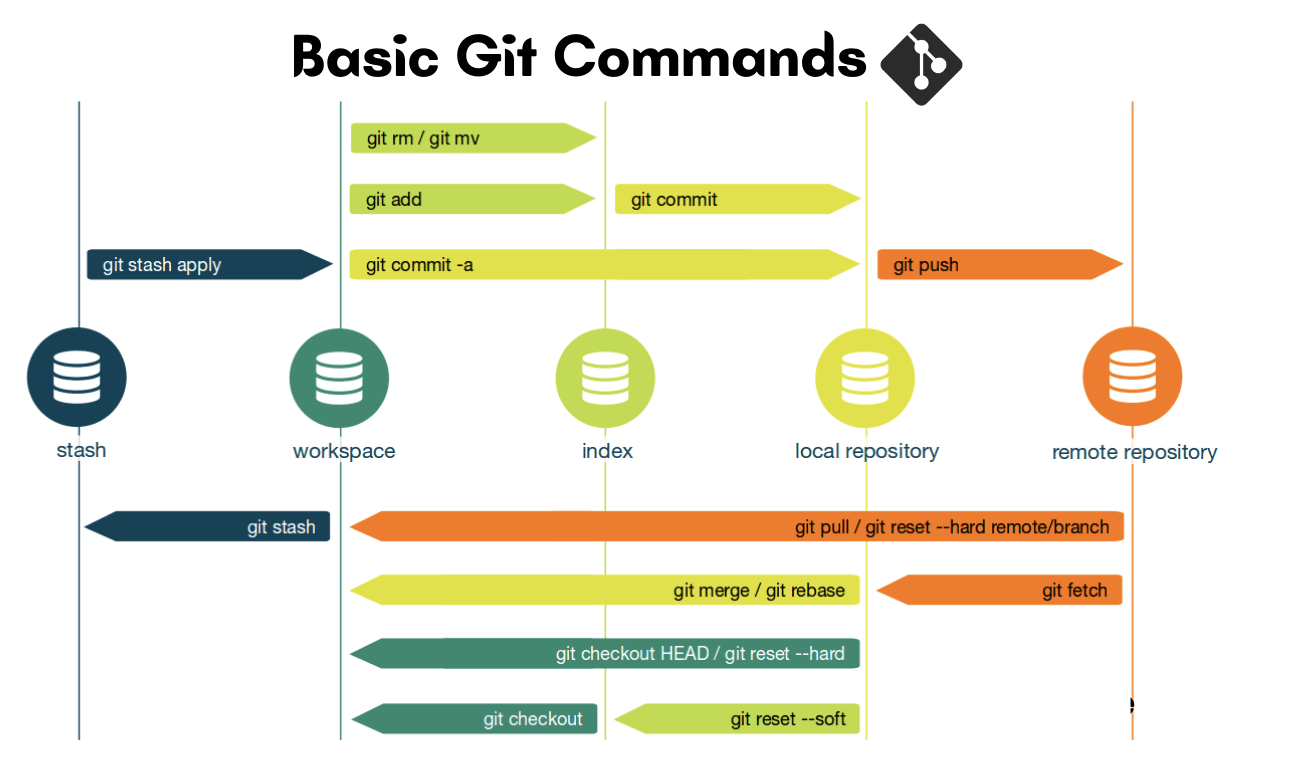
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**I. Mục tiêu**

* Hiểu về cách sử dụng hệ thống quản lý phiên bản (Version Control System - VCS) để quản lý mã nguồn.
* 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 lệnh cơ bản trên SourceTree.

**II. Nội dung thực hành**

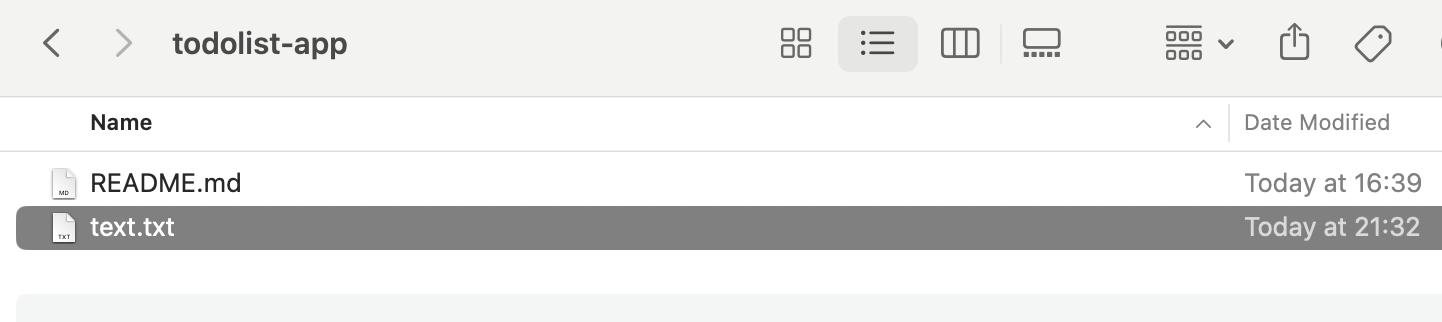
1. **Các tác vụ cơ bản với Git:**

****

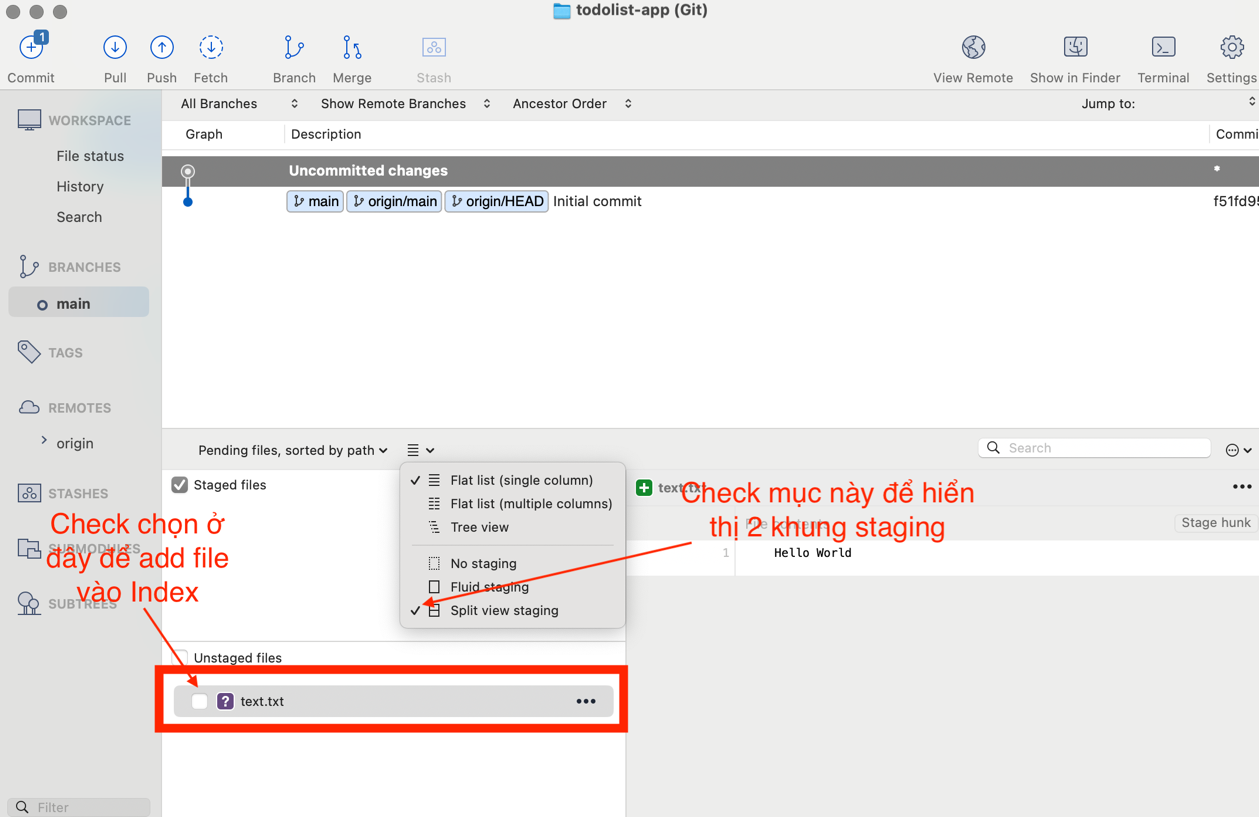
Giới thiệu về tất cả các chức năng cơ bản trong SourceTree để quản lý source code.

* 1. **Add changes to Index repository:**

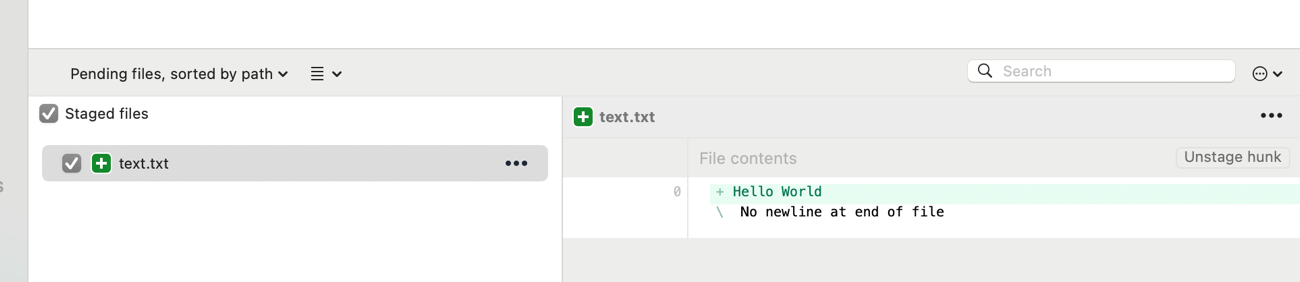
Đầu tiên mình tạo một file mới tên là text.txt với nội dung là “Hello World” trong thư mục dự án của mình.



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

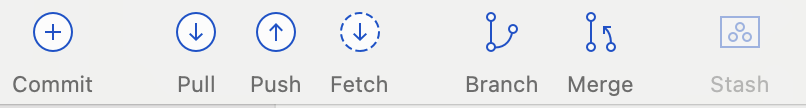


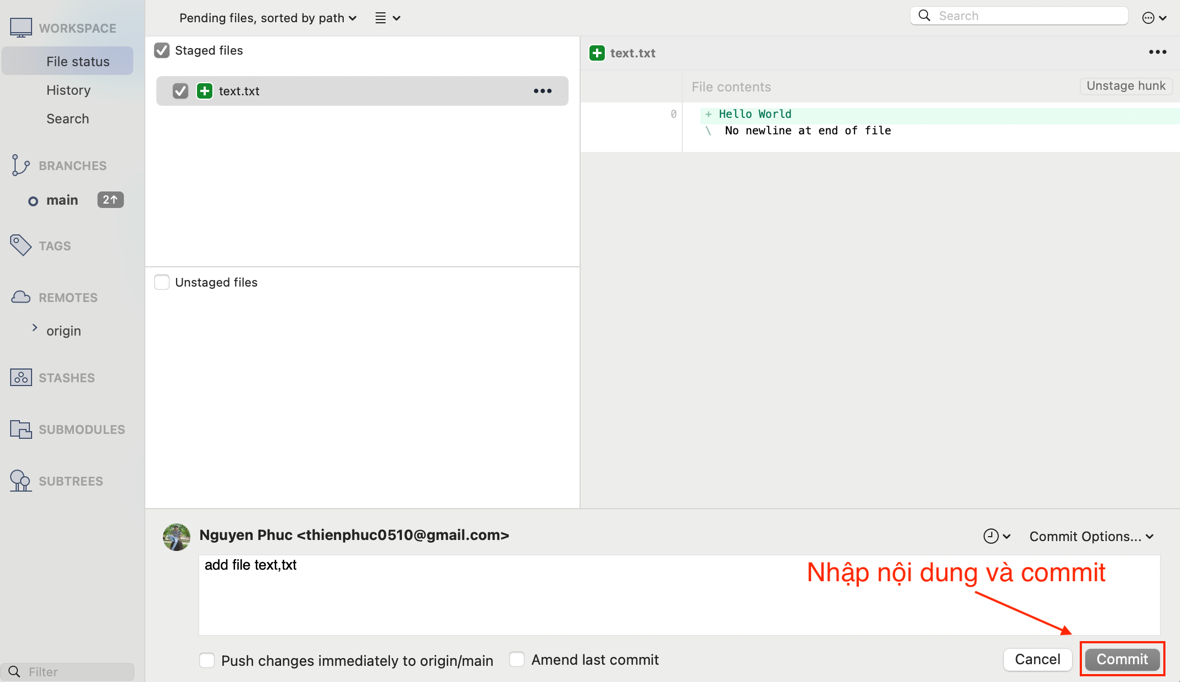
Sau khi check chọn add file text.txt, nó sẽ xuất hiện màn hình như sau:



* File text.txt đã được add vào Index (Staged)
  1. **Commit changes:**

Sau khi đã add file vào index thì chúng ta click vào button Commit trên thanh tool bar, rồi nhập nội dung và nhấn commit.



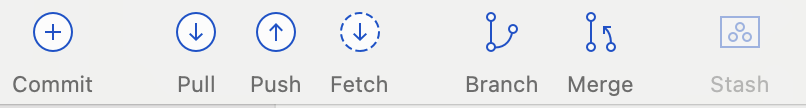
****

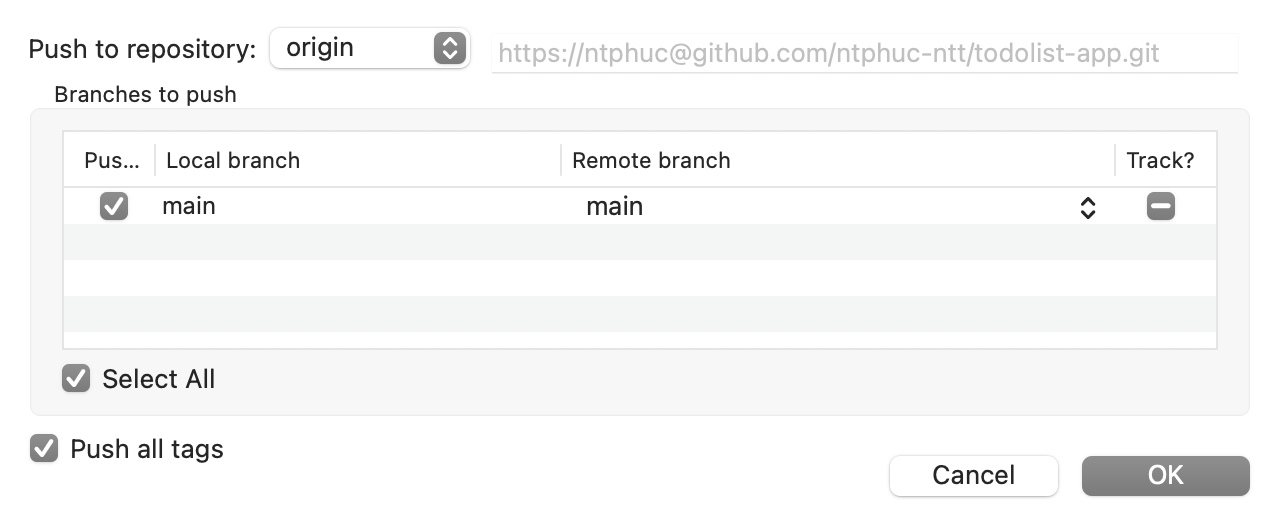
Lưu ý: Sau khi commit changes, thì changes vẫn ở dưới local repository chứ nó chưa được đưa lên mạng.

* 1. **Push changes:**

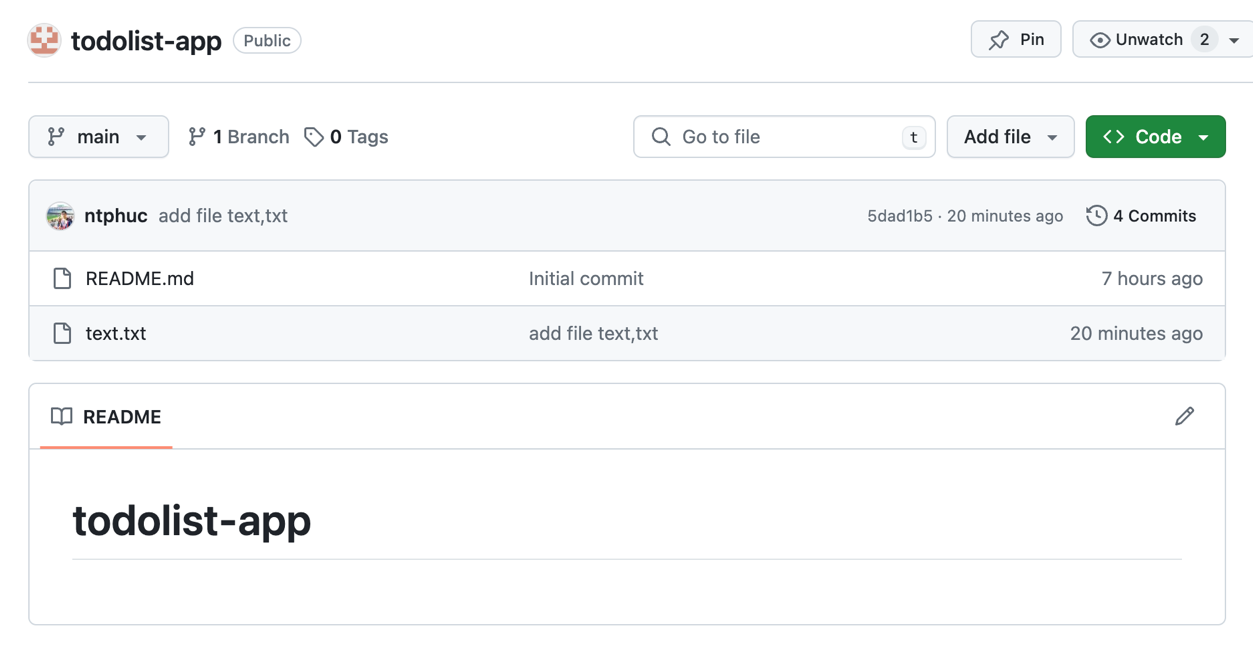
Quá trình push changes này sẽ giúp đẩy file lên mạng (remote remote repository)

Chọn button Push trên thanh tool bar và click OK





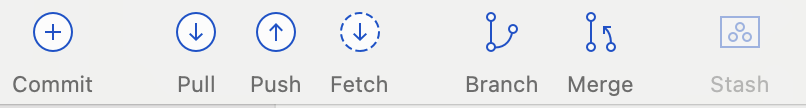
Sau khi push files lên, chúng ta sẽ thấy file xuất hiện trên GitHub repository.



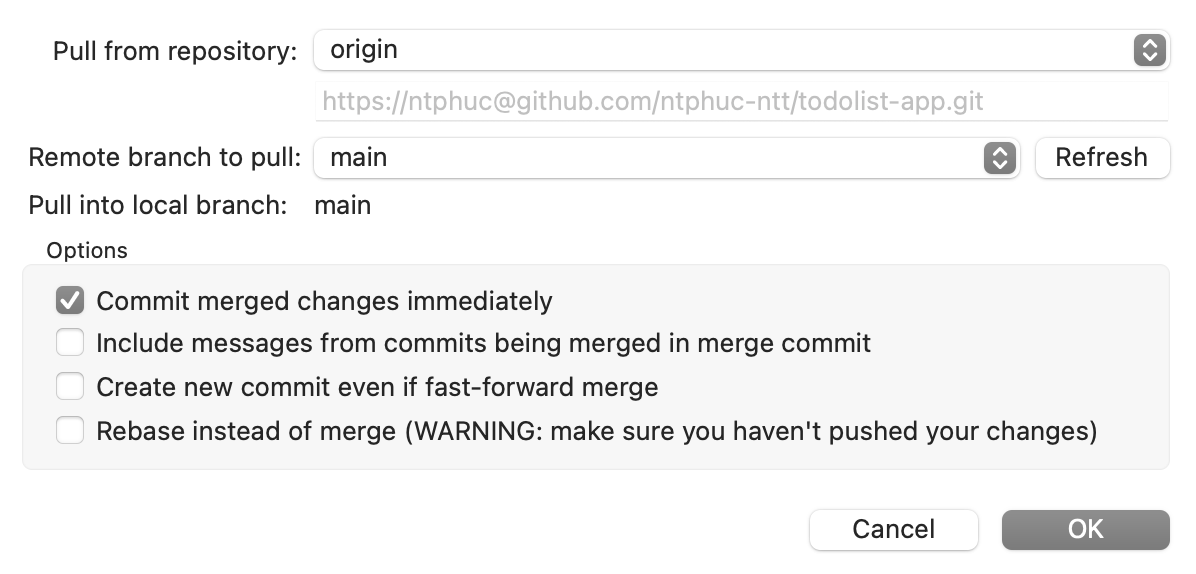
* Như vậy là ta đã Push những files lên cho các team member khác cùng sử dụng
  1. **Pull changes:**

Ngữ cảnh: khi một team member khác push những thay đổi lên remote repository, trước khi mình làm việc với working space của mình, thì mình cần cập nhật những thay đổi đó. Để làm được việc này chúng ta phải thực hiện action pull files về.

Chọn button Pull trên tool bar để kéo files từ remote repository về working space.

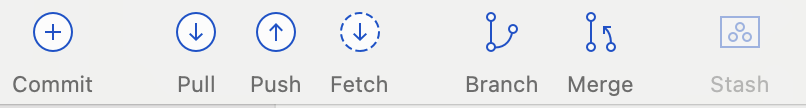


Sau đó click OK.

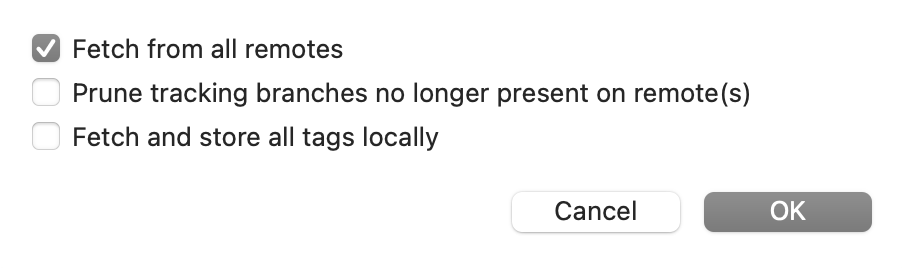


* 1. **Fetch changes:**

Thao tác "Fetch" được sử dụng để lấy dữ liệu từ remote repository về máy tính của bạn mà không hợp nhất (merge) với bất kỳ thay đổi nào trong nhánh hiện tại của bạn. Thực hiện "Fetch" sẽ cập nhật thông tin về các nhánh và commit mới nhất từ remote repository mà **không làm thay đổi** các thay đổi trong working directory của bạn.

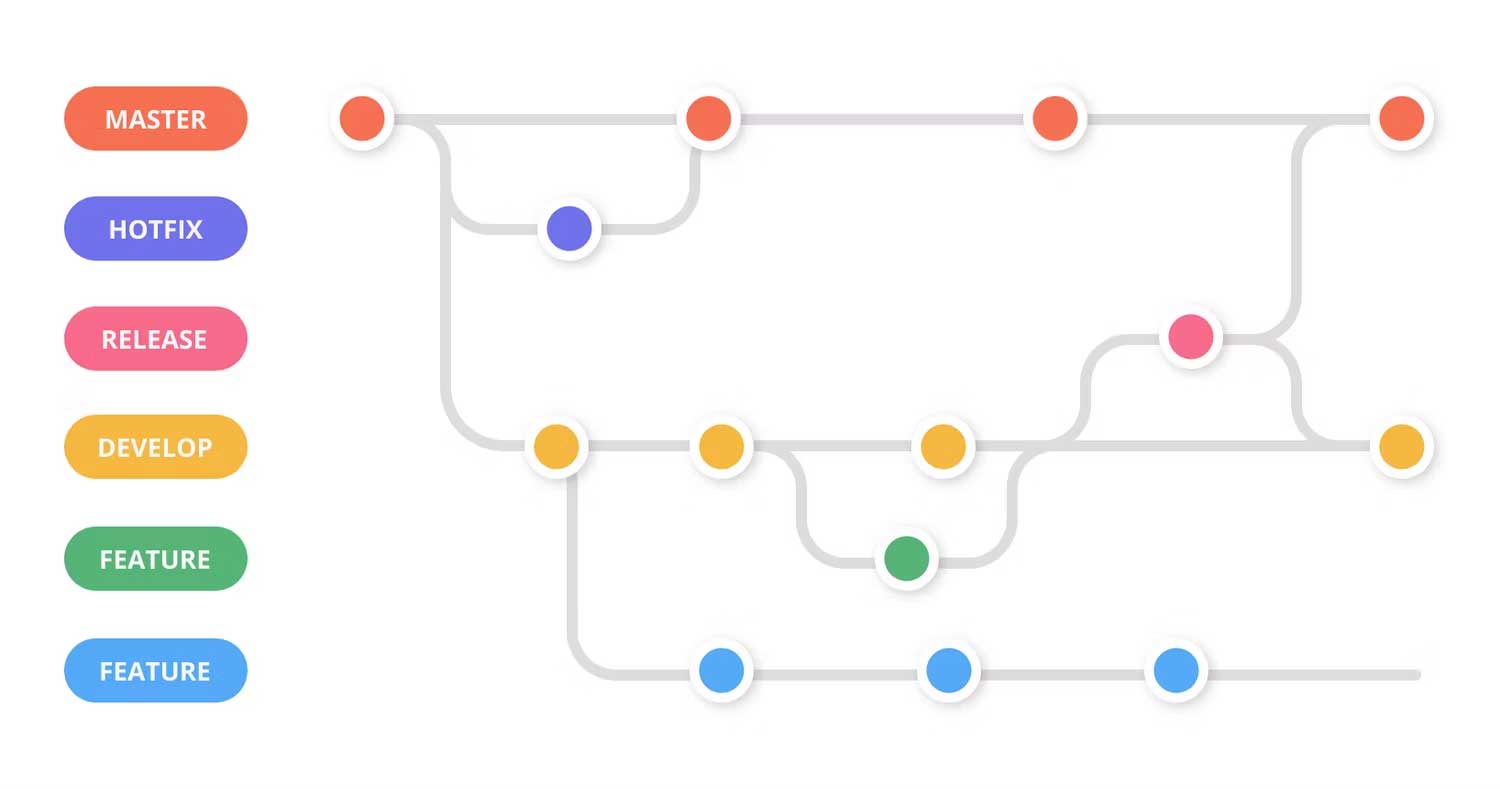


Chọn button Fetch và click OK.



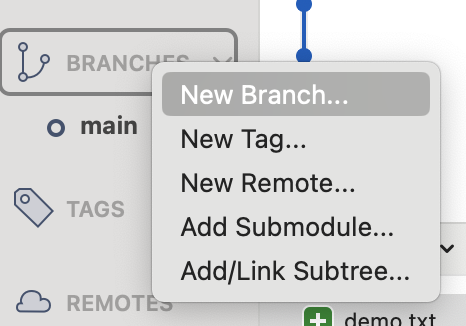
* 1. **Tạo nhánh (Branch) mới:**

Ngữ cảnh: Ngay từ lúc đầu repository mặc định nó sẽ có 1 nhánh là main. Tuy nhiên để quản lý source code để tốt hơn chúng ta cần tạo thêm những nhánh khác để phát triển source code ví dụ như develop, feature, release…

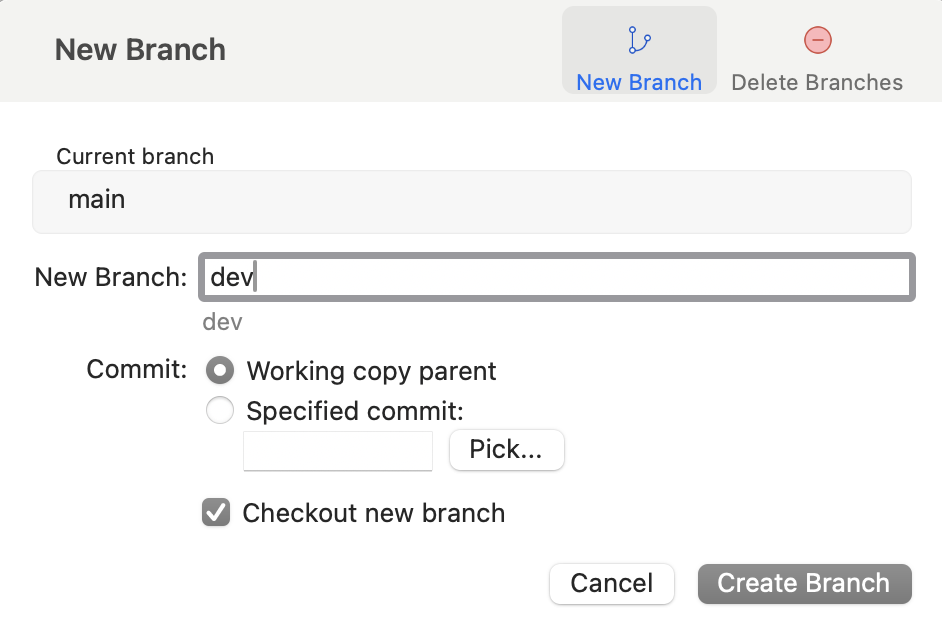


Hình trên là best practice để làm việc với Git.

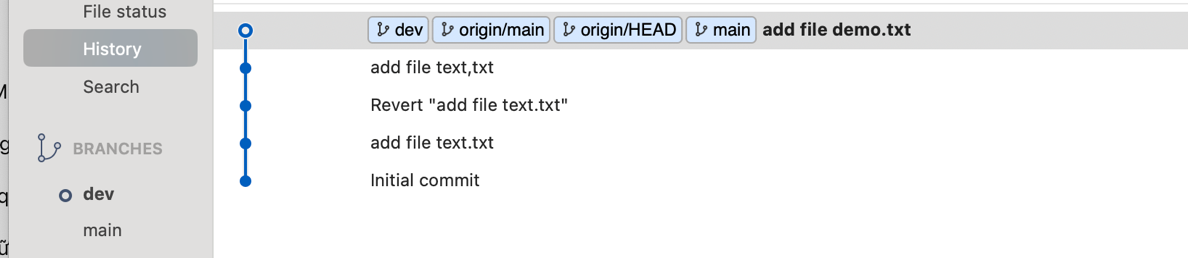
Trong SourceTree, để làm được việc này thì các bạn click chuột phải chỗ phần Branches -> New Branch.



Nhập tên để tạo một branch mới:



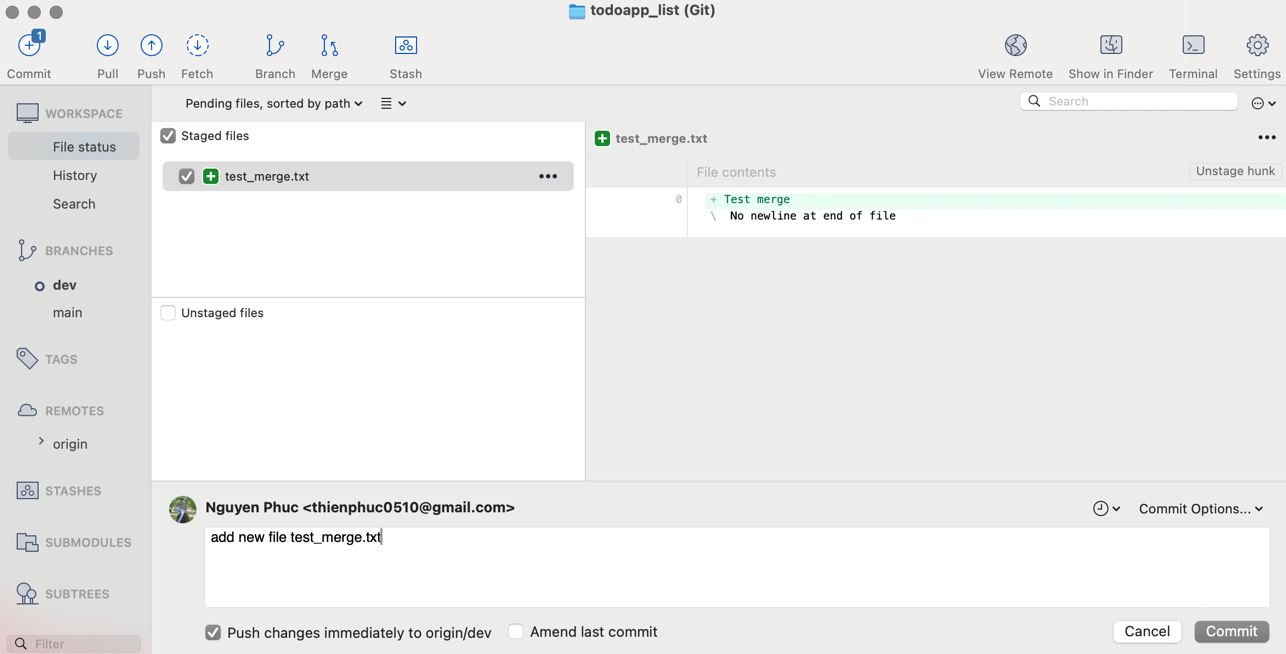
Sau đó chúng ta sẽ thấy trên màn hình nhánh mới đã được tạo:



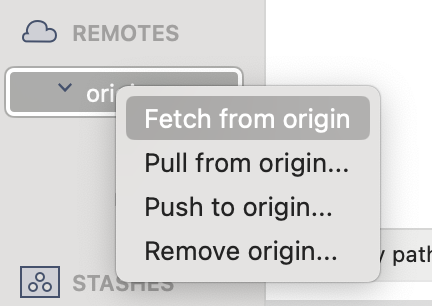
* 1. **Merge branch khác vào branch làm việc hiện tại:**

Ngữ cảnh: Một team member khác tạo một nhánh tên là dev, sau khi bạn này có những thay đổi và đã đẩy lên remote repository. Và bạn này thông báo cho mình cần phải cập nhật thay đổi đó vào nhánh làm việc của mình thì các bạn cần checkout nhánh mới đó về và merge và nhánh hiện tại của mình. Cụ thể sẽ làm từng bước sau:

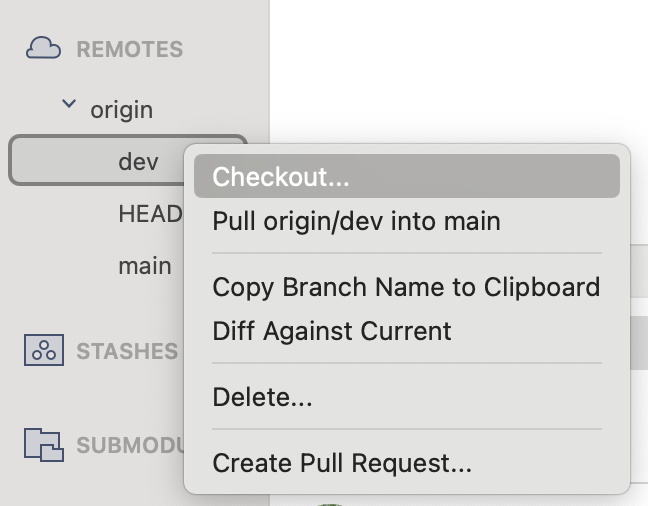
Đầu tiên bạn đó sẽ tạo một file mới, và nhánh mới tên dev, rồi commit & push file này lên remote repository.



Ở working space SourceTree của mình, để thấy được những branch mới được tạo bởi những member khác thì các bạn sẽ vào phần Remotes->origin->Fetch from origin.



Sau khi Fetch thì nhánh dev sẽ xuất hiện, để làm việc với nhánh dev thì các bạn chọn nhánh dev click chuột phải và chọn Checkout…



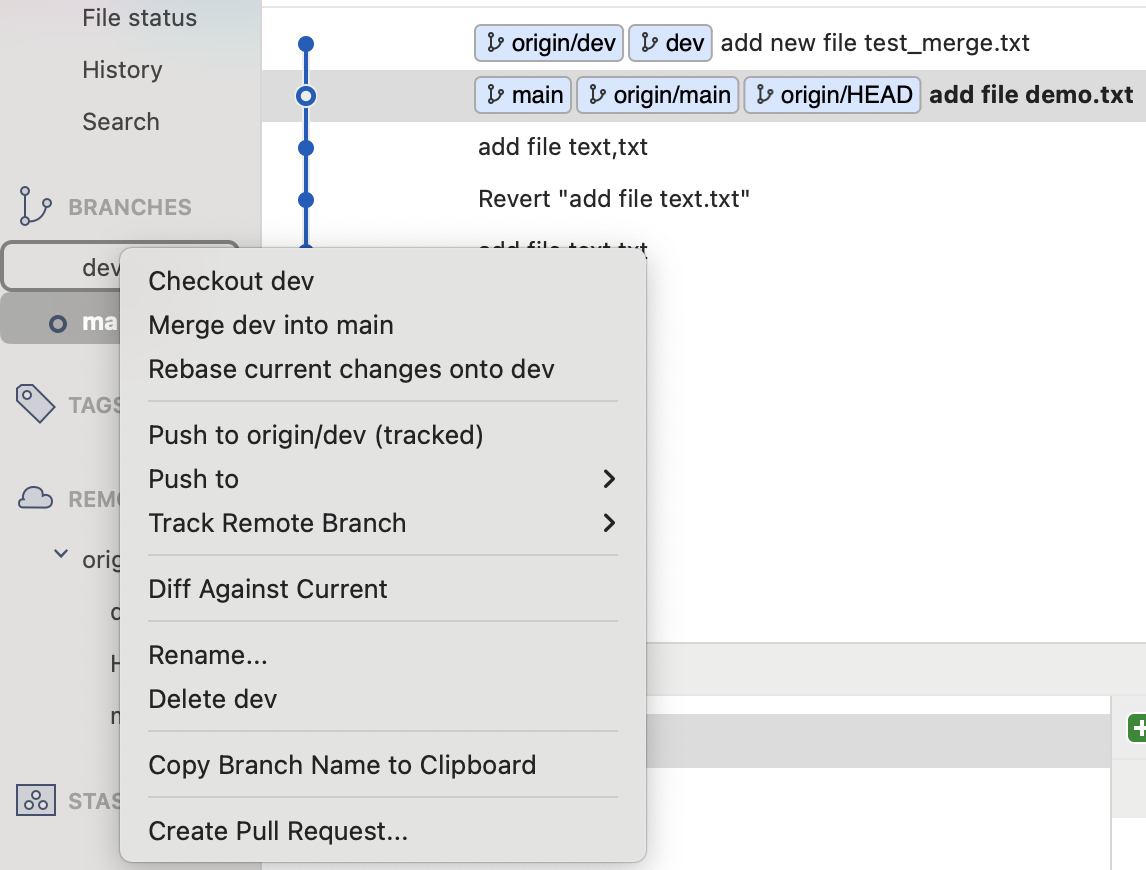
Khi checkout nhánh dev xong, thì chúng ta sẽ thấy 2 nhánh là nhánh dev và main. Giữa hai nhánh này hiện tại có sự khác nhau, là nhánh dev có được hơn 1 commit.



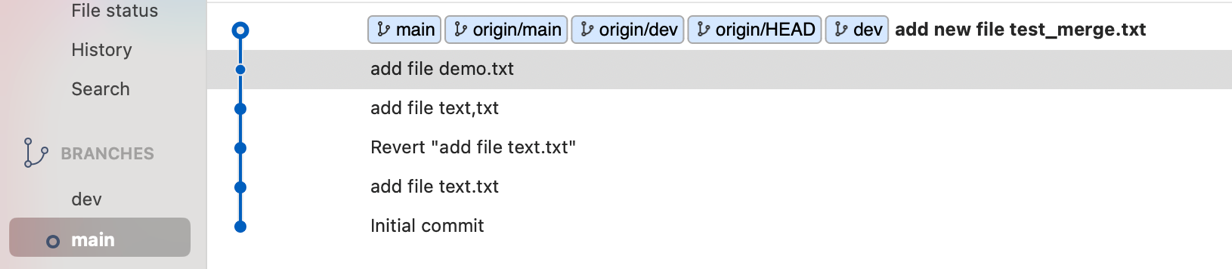
Và chúng ta muốn 2 nhánh này đồng nhất với nhau thì cần thực hiện một lệnh merge từ nhánh dev và nhánh main.

Để làm được việc này, chúng ta cần chọn Check out nhánh main, chuột phải nhánh main sau đó click Checkout main.

Sau đó tiếp tục nhấn chuột phải ở nhánh main và chọn **Merge dev into main**.



Sau khi thực hiện lệnh này xong thì nhánh dev và main đã được đồng nhất commits ở working space. Để nó đồng nhất trên remote repository thì mình cần push những thay đổi ở nhánh main lên remote repository nếu chúng ta cần cập nhật những thay đổi cho team members khác ở nhánh main này. Sau khi thực hiện xong chúng ta sẽ thấy màn hình như thế này:



1. **Bài tập thực hành:**
   1. Thêm 1 file mới tên là test2.txt nội dung là “Hello Test 2”, sau đó thực hiện lệnh commit file và push file lên remote repository.
   2. Trong repository trên GitHub của mình, invite ít nhất 2 người bạn cộng tác chung trong repository. Nhờ 1 người bạn commit và push file thay đổi phía họ lên remote repository. Sau đó ở phía mình thực hiện lệnh fetch, pull những thay đổi về cập nhật vào chính working space của mình.
   3. Tạo một nhánh mới tên là feature1, rồi thay đổi nội dung trong test2.txt thêm 1 dòng nữa là “Test feature tạo branch mới và merge”. Sau đó commit và push sự thay đổi này lên trong nhánh feature1.

Nhờ một bạn khác trong team của mình pull về nhánh mới này và merge vào nhánh dev của họ và kiểm tra xem họ đã cập nhật sự thay đổi này chưa.

* 1. Cũng tương tự như bài tập 3, nhưng nhờ người bạn trong team mình tạo một nhánh mới feature2, rồi thay đổi là thêm một file mới tên là test3.txt với nội dung là “Hello Test3”, sau đó thực hiện lệnh commit và push sự thay đổi này lên trong nhánh feature2.

Sau đó ở phía working space của mình thì pull sự thay đổi này về và merge vào nhánh dev của mình và kiểm tra xem phía mình đã thấy được sự thay đổi là thêm file test3.txt hay chưa.