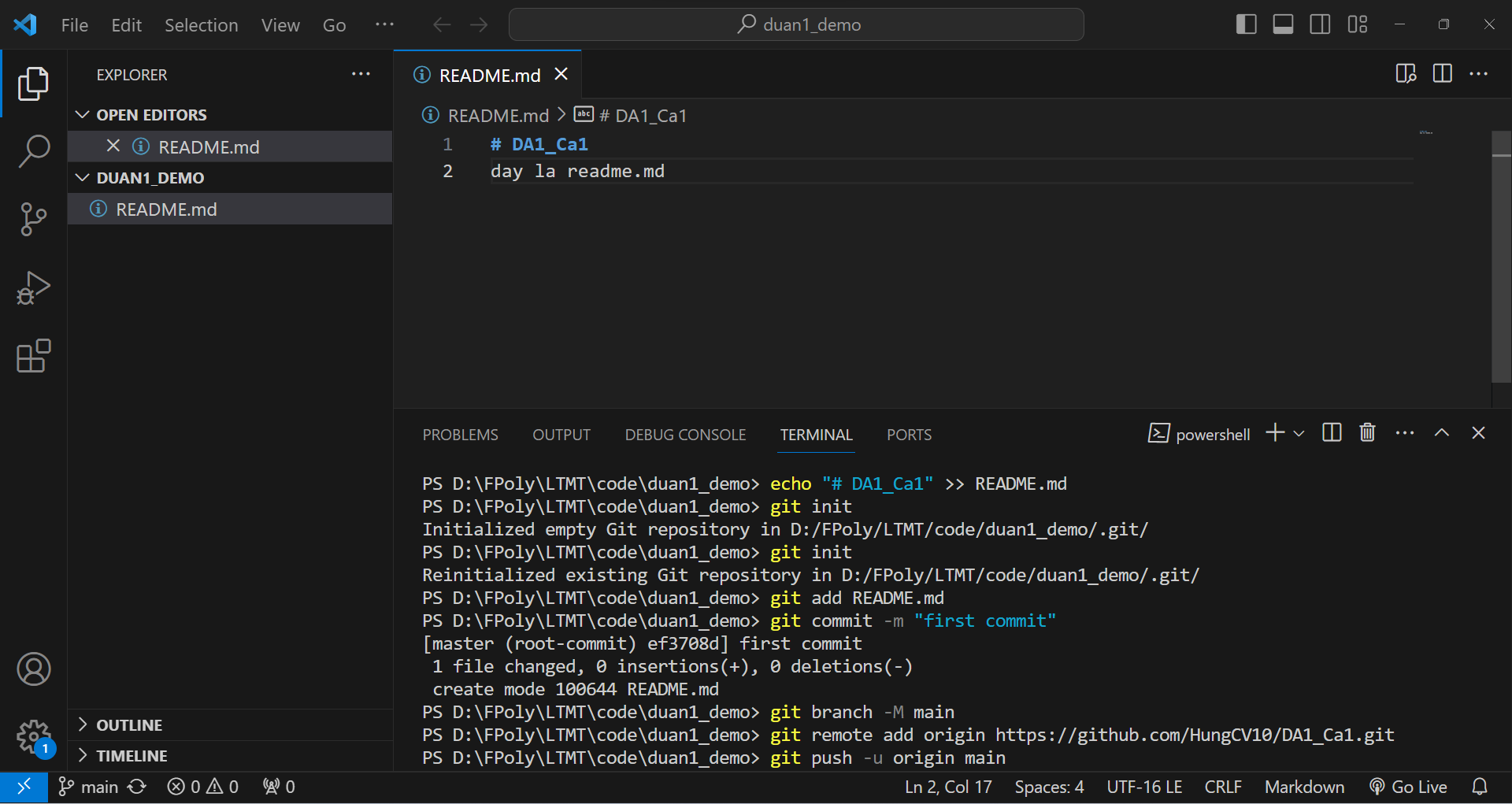
1. Tạo project và đẩy lên git

Tạo 1 project trên git,

Tiến hành thao tác các câu lệnh

Nhánh main không động tới.



1. Tạo nhánh develop

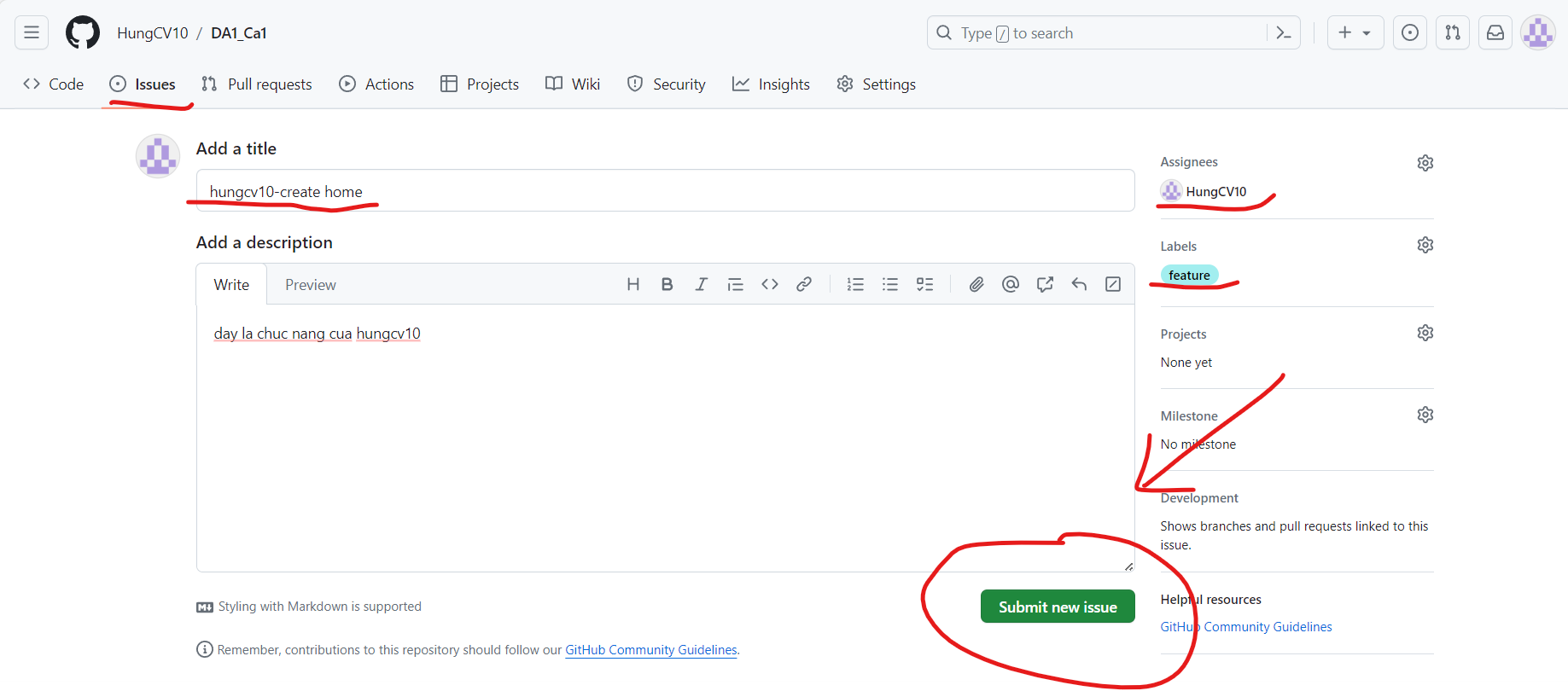
* git branch develop

sau đó checkout sang nhánh develop

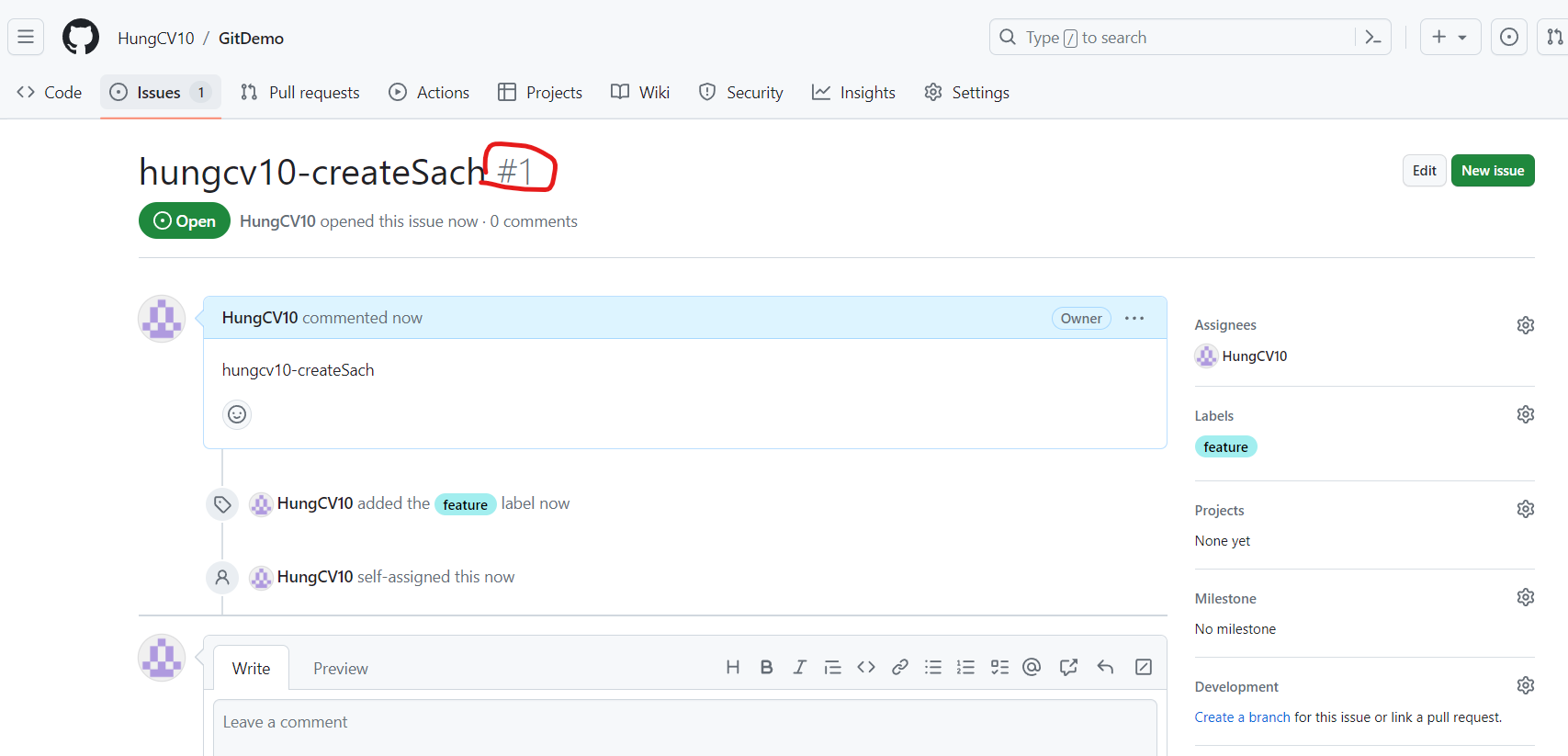
* git checkout develop

Lưu ý không làm việc trên nhánh main

1. Giao task cho thành viên



Click vào submit



Khi commit code lên phải nằm trong khu vực này

1. Push nhánh develop lên remote

git push -u origin develop

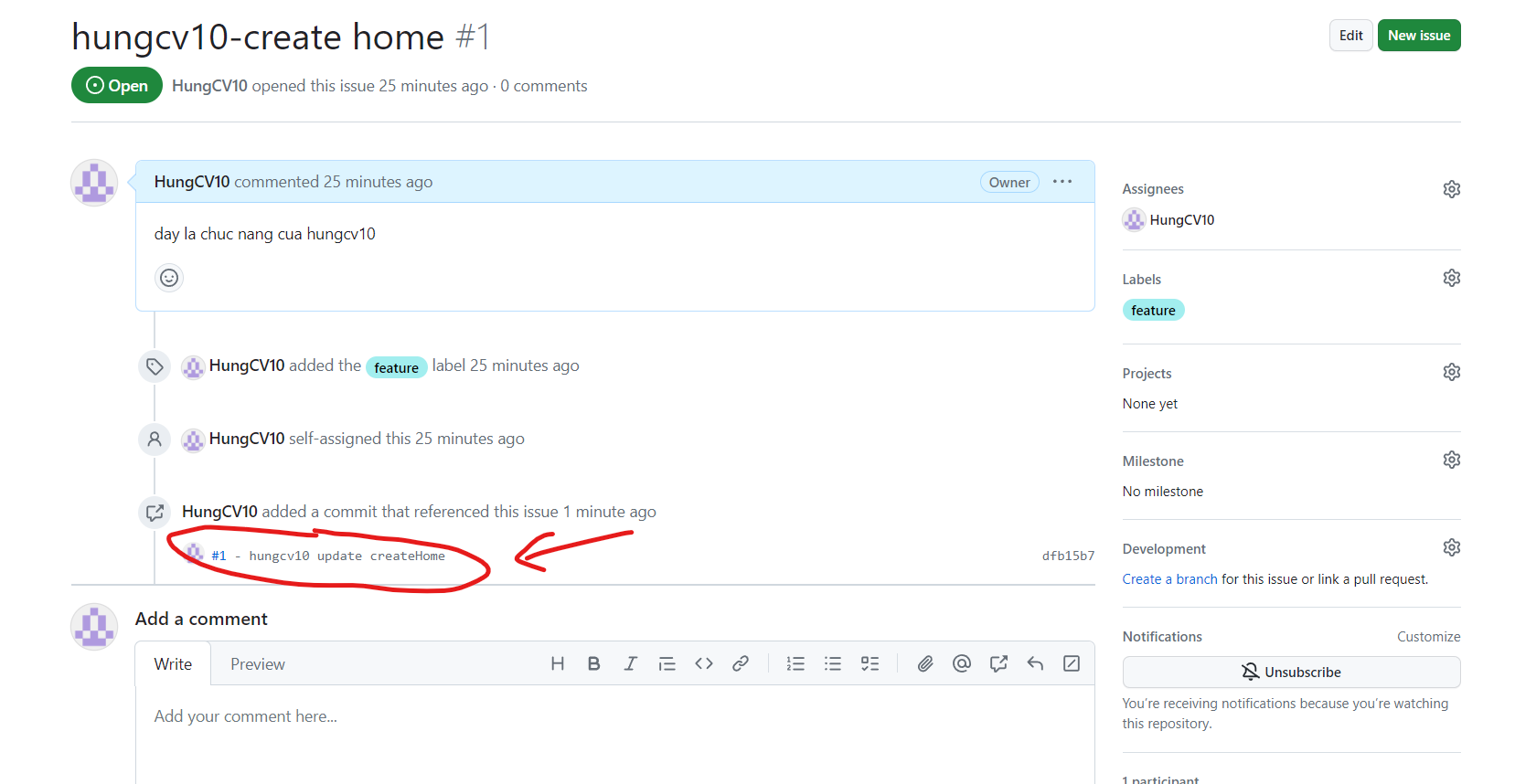
1. Checkout 1 tính năng mới

* git checkout -b feature/1-createHome develop

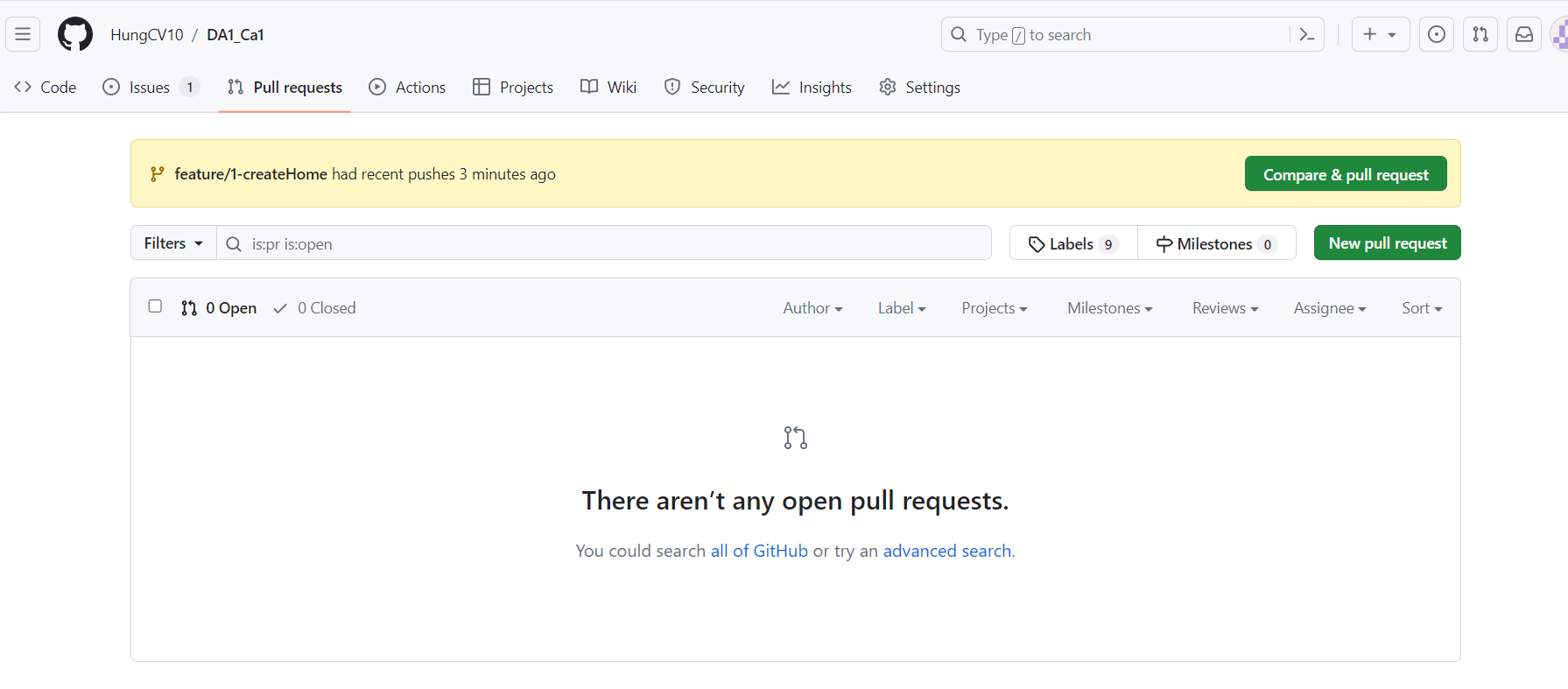
1. Giải thích: git checkout -b feature/1-createHome: Đây là lệnh kết hợp giữa git branch và git checkout. -b là viết tắt của branch. Lệnh này tạo ra một nhánh mới có tên là feature/1-createHome và chuyển đổi (check out) sang nhánh mới này ngay lập tức.
2. develop: Tham số này cho biết nhánh feature/1-createHome mới được tạo ra sẽ bắt đầu từ đâu. Trong trường hợp này, nó bắt đầu từ develop, nghĩa là nó sẽ chứa tất cả các commit mà nhánh develop đã có tại thời điểm tạo nhánh.
3. lệnh này tạo ra một nhánh mới tên là feature/1-createHome, bắt đầu từ nhánh develop, và chuyển công việc hiện tại sang nhánh mới này.

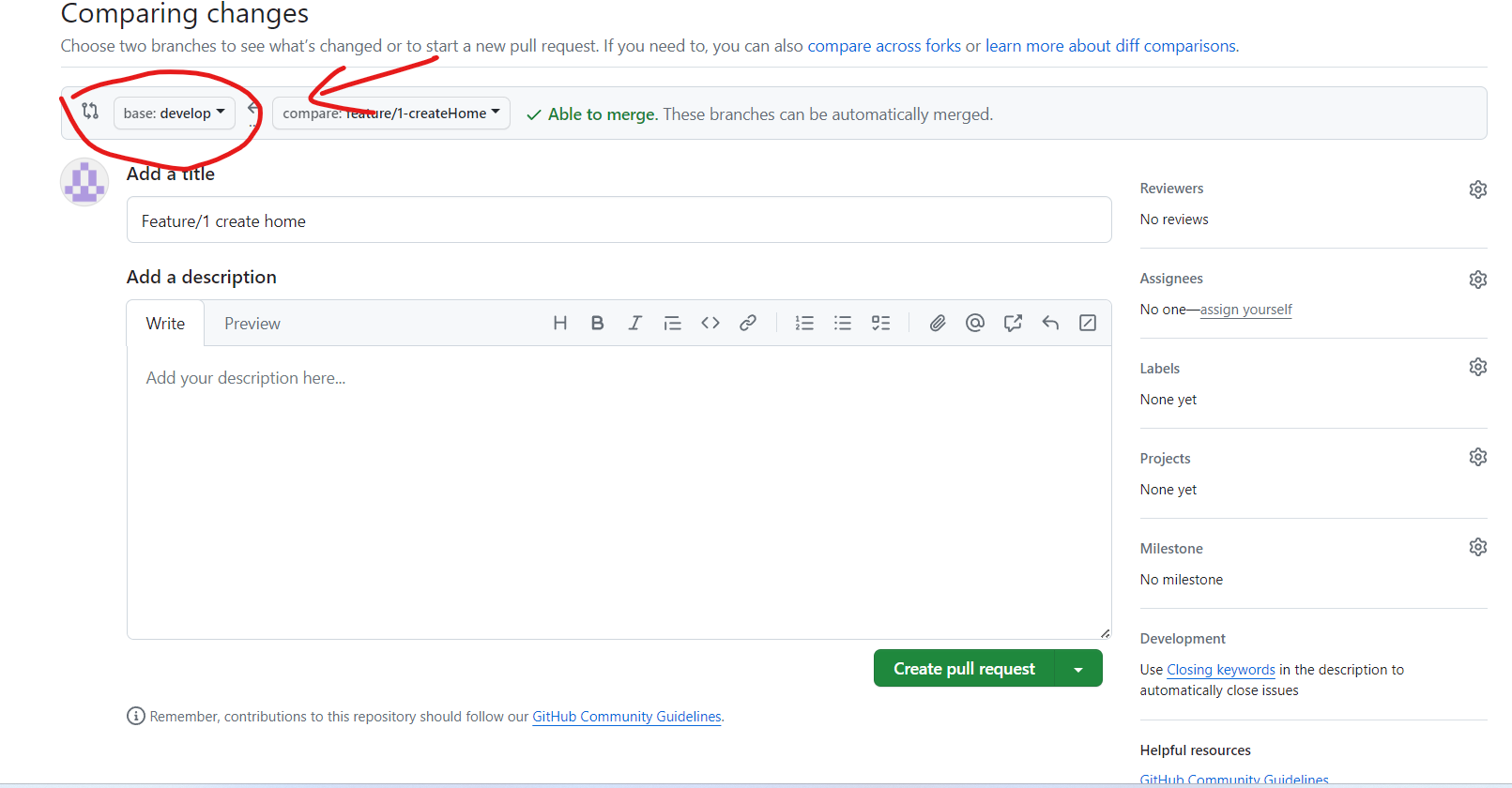
* Tạo các file cần thiết cho dự án

Tiến hành tạo file sau đó add lên git

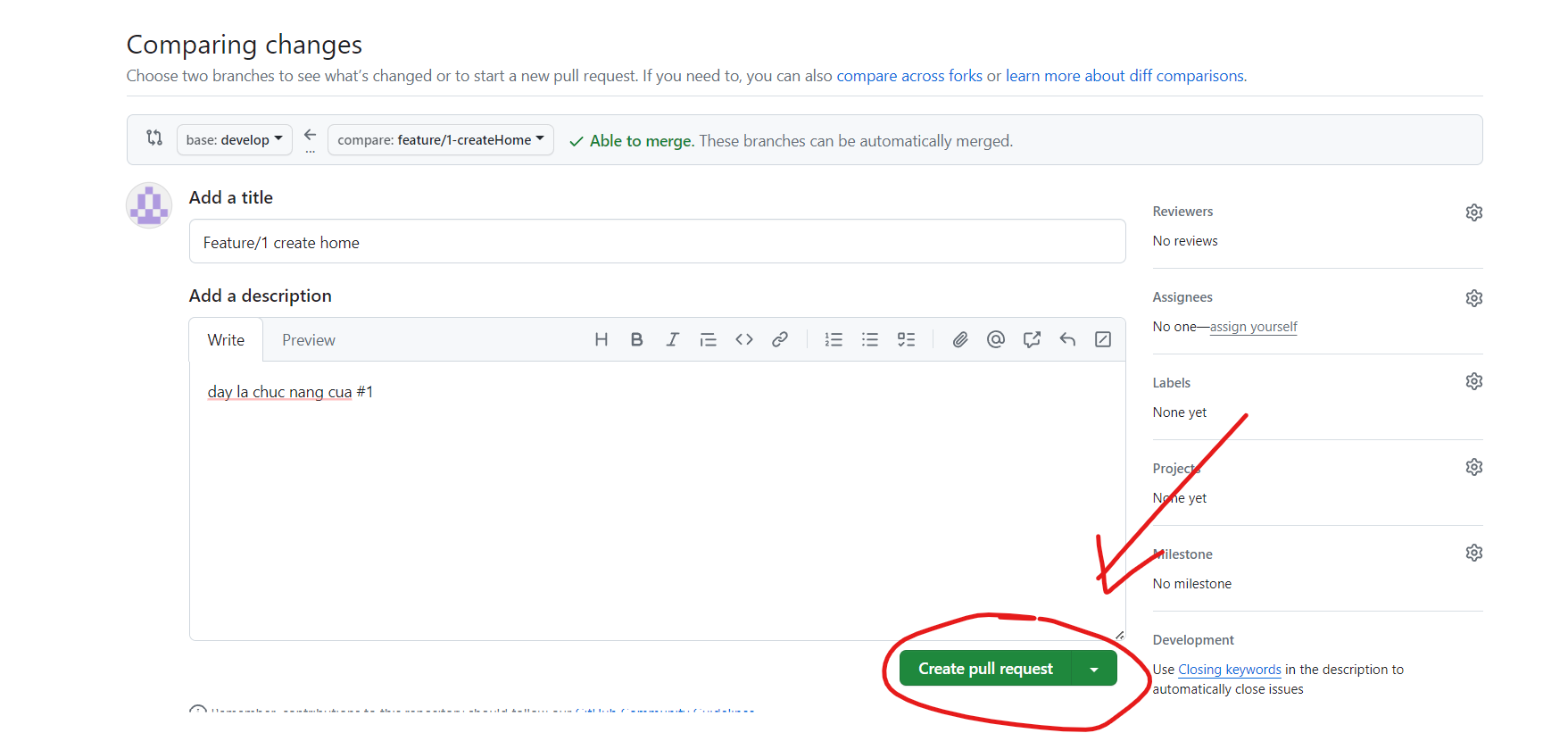
* git add .
* git commit -m '#1 - hungcv10 update createHome'
* Giải thích: '#1 - hungcv10 update createHome': Đây là thông điệp commit mà bạn đã chọn. Thông điệp này nên mô tả ngắn gọn những thay đổi mà bạn đã thực hiện. Trong trường hợp này, thông điệp cho biết người dùng 'hungcv10' đã cập nhật 'createHome'. Số '1' có thể là tham chiếu đến một vấn đề hoặc yêu cầu kéo trong hệ thống quản lý công việc.
* git push
* git push --set-upstream origin feature/1-createHome
* 

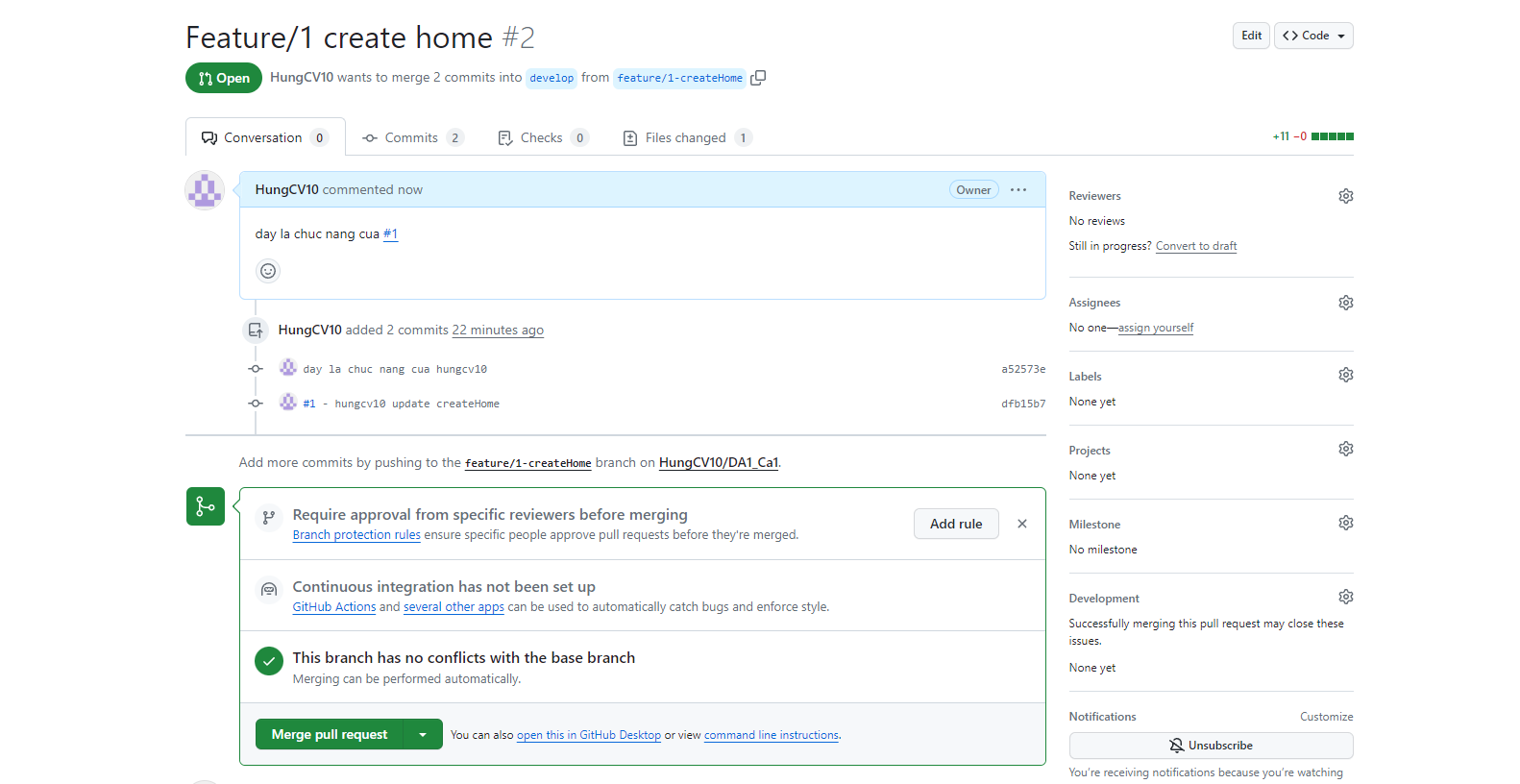
Sau đó người quản lý tiến hành tạo pull request





Tiến hành merge pull request





1. Sau khi hoàn thành chức năng thì tiến hành pull về develop

* git checkout develop
* git pull

1. tạo nhánh release

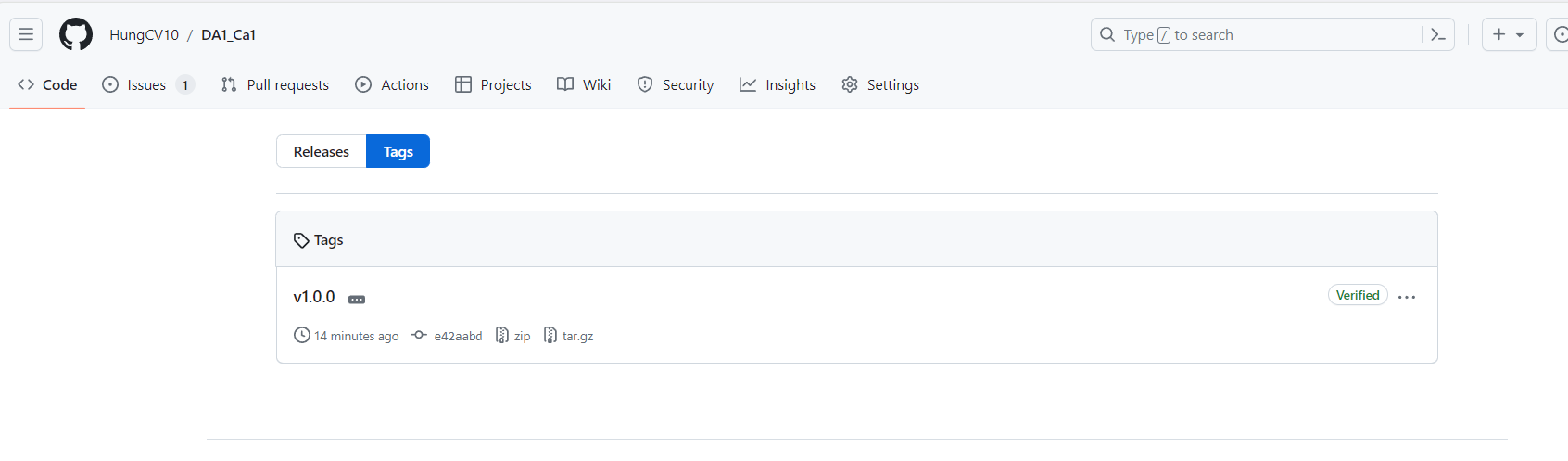
* git checkout -b release-1.0.0 develop

tạo tag:

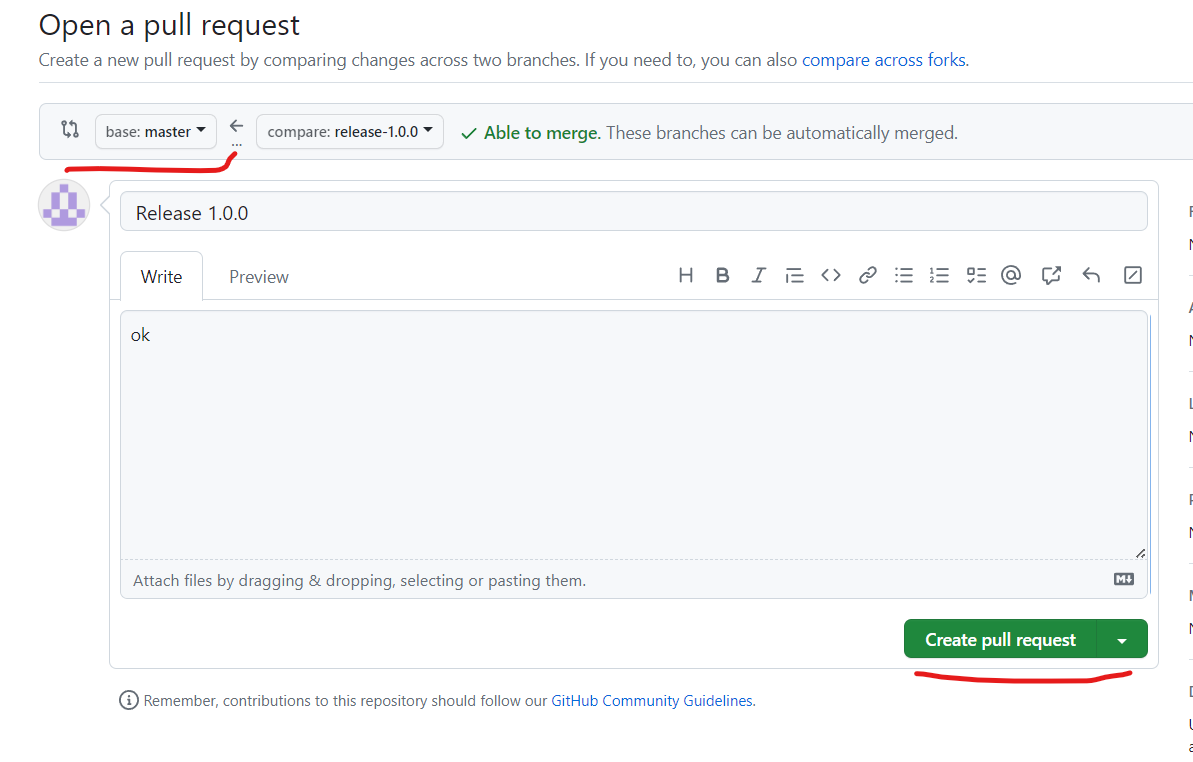
* git tag 'v1.0.0'
* git push --tag

Sau đó merge từ develop sang release

* git merge develop
* git push
* git push --set-upstream origin release-1.0.0



1. Thành công thì nhiệm vụ sẽ là merge từ release sang main



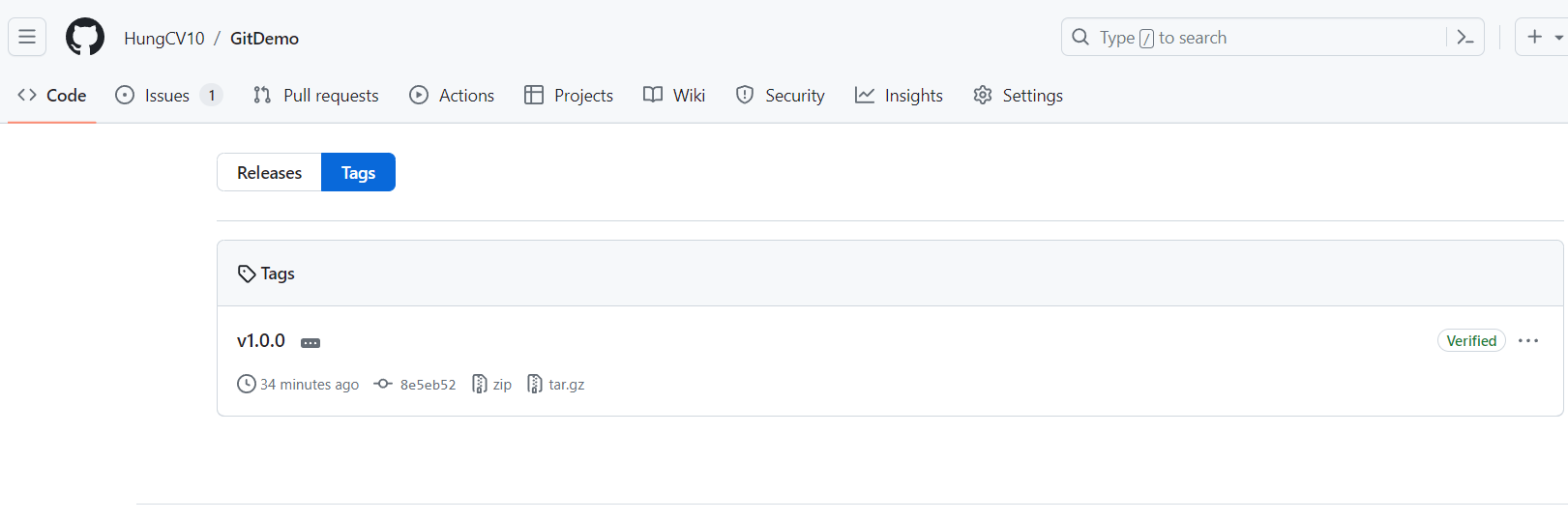
Sau đó tiến hành merge

Rồi checkout sang main để kéo về

* git checkout master
* git pull

Tạo phiên bản cho product:

* git tag 'v1.0.0'
* git push –tag



Sau khi xong sẽ xóa các chức năng

* git branch -d release-1.0.0 (xóa ở local)
* git push origin -d release-1.0.0
* git branch -d feature/1-createHome
* git push origin -d feature/1-createHome

1. **Settup GIT**

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| git config --list | Xem thông tin |
| git config –global user.name “…” | Điền tên đăng nhập … |
| git config –global user.email “…” | Điền email vào … |

1. **Các bước đẩy code lên GIT**

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| git init | Tạo kho chứa Repository phía local |
| git add … | Điền tên file cần đẩy vào … |
| git add . | Đẩy toàn bộ file |
| git status | Kiểm tra trạng thái |
| git commit -m”…” | Điền thông điệp vào … |
| git remote add origin … | Điền URL Sever Git vào … |
| git push -u origin … | Điền tên nhánh vào … ( Cho lần đầu tiên khi đẩy code lên ) |
| git push | Git push cho các lần sau |

1. **Cần cập nhập code thì quay lại bước 2 ( git add )**
2. **Khi cần thêm 1 thứ gì đó trong đoạn code đã commit**

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| git reset | Giúp xoá đi cái commit gần nhất |
| Quay lại bước 2 Git add |  |

1. **Cần khôi phục khi đã xoá hết file**

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| git restore . | Khôi phục toàn bộ các file |
| git restore <tên file muốn khôi phục > |  |

1. **Câu lệnh Git branch cơ bản**

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| git branch | Kiểm tra những nhánh có trên local (nhánh ta đang dùng sẽ có màu xanh đánh dấu \* ) |
| git branch -r | Kiểm tra những nhánh có trên remote |
| Git branch -a | Kiểm tra những nhánh có cả trên remote và local |

1. **Tạo một nhánh mới**
2. Tạo một nhánh mới nhưng không chuyển qua nhánh đó

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| Git branch … | Điền tên nhánh cần tạo vào chỗ … |

1. Tạo một nhánh mới và chuyển qua luôn nhánh đó

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| Git checkout -b … | Điền nhánh mới cần tạo vào chỗ … |
| Git switch -c … | Điền nhánh mới cần tạo vào chỗ … |

1. **Chuyển qua lại giữa các nhánh**

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| git checkout “ten\_nhanh” |  |
| git switch “ten\_nhanh” |  |

1. **Sửa tên nhánh đã có**

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| git branch -m”ten\_nhanh\_moi” | Đứng tại nhánh cần sửa tên mới |
| git branch -m”ten\_nhanh\_cu” “ten\_nhanh\_moi” |  |

1. **Xoá nhánh**

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| git branch -D “ten\_nhanh” | Xoá một nhánh trên local |
| git push origin –delete “ten\_nhanh” | Xoá một nhánh trên remote |
| git fetch | Cập nhập các nhánh từ remote xuống local |

1. **Đưa một nhánh từ local lên remote**
2. Cách 1: Chỉnh sửa code tại nhánh mới và push code lên remote

Git add .

Git commit -m “Message thông điệp”

Git push -u origin “ten\_nhanh\_moi”

1. Cách 2: Không muốn thay đổi code chỉ muốn cập nhập nhánh ( một trong 2 cách sau )

Git push -u origin “ten\_nhanh\_moi”

Hoặc git push -u origin HEAD

1. **Đẩy code lên nhưng không muốn pull về mà muốn thay đổi trên git**

**--force hoặc -f**: Ép đẩy các thay đổi của bạn lên kho lưu trữ từ xa, bất kể các thay đổi trên nhánh từ xa. Sử dụng lệnh này khi bạn chắc chắn rằng bạn muốn ghi đè các thay đổi trên kho lưu trữ từ xa.

**--force-with-lease**: An toàn hơn so với --force, chỉ đẩy các thay đổi nếu nhánh từ xa không thay đổi kể từ lần cuối bạn kiểm tra.