**Tìm hiểu Git**

\***Version control system**: lưu trữ toàn bộ các file và ghi lại toàn bộ lịch sử thay đổi của file, ta có thể thể xem lại toàn bộ thay đổi của file như một timeline bao gồm: nội dung file bị thay đổi , ngày giờ sửa đổi , người thay đổi là ai , lí do thay đổi file ,…

\***Git**: hệ thống quản lí phiên bản phân tán.

\*Git là 1 hệ thống thuộc vcs, nên git cũng có thể lữu trữ lại toàn bộ file và ghi lại toàn bộ lịch sử thay đổi của file , khi mình sửa dòng code nào , xóa dòng code nào,bỏ dấu thừa ở đâu , … Tất tần tật được ghi lại.

\***Repository**:

- Local repository: là repo được lưu tại máy của mình , mình có thể thêm, sửa, xóa file , tạo “commit”.

- Server repository: là repository nhưng được lưu trữ tại server của các hosting-server của git như github,gitlab,… .Repo này có thể chia sẽ tới người khác ,mình có thể để private or public.

\***Commit**:

-Thay đổi tất cả những gì commit: Git commit -m “message thay đổi tất cả”.

-Lưu lại thay đổi và đè lên commit trước đó: Git commit --amend -m “message thay đổi của commit trước(last commit)”.

-Đè lên commit trước đó và đè lên người tạo commit : Git commit --amend --author “Nguyen Dai Nghia [<nghiabeo1605@gmail.com>”](mailto:<nghiabeo1605@gmail.com>\”).

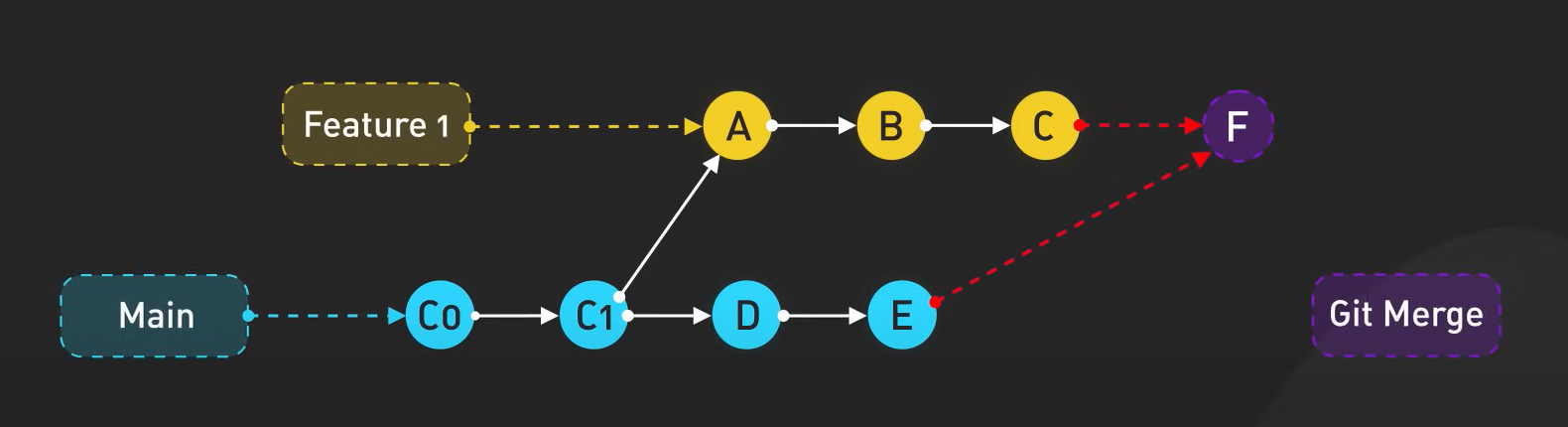
\*Log:

- Xem 1 hoặc danh sách các commit đã tạo gần đây: Git log -5

-Hiển thị các commit trên mỗi dòng: Git log --oneline 5

\*So sánh các thay đổi của code trước khi add thay đổi commit : Git diff

\*Git merge



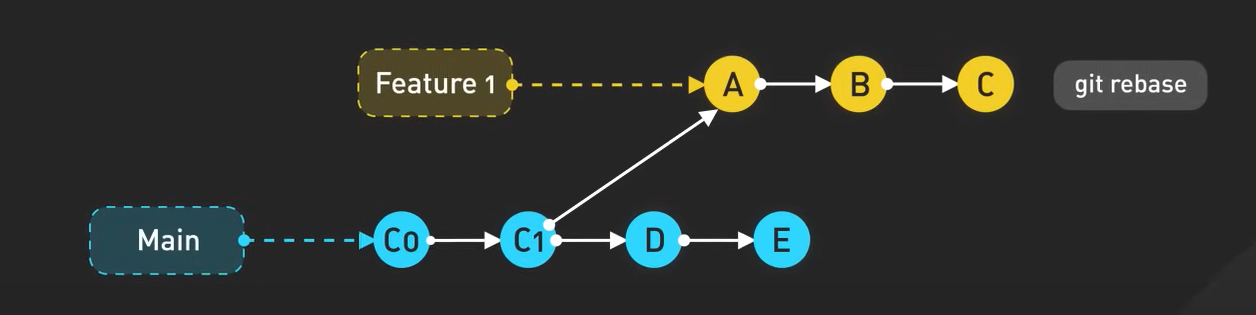
Khi sử dụng git merge nó sẽ lấy tất cả commit của 2 nhánh vào 1 nhánh và tạo ra 1 commit merger

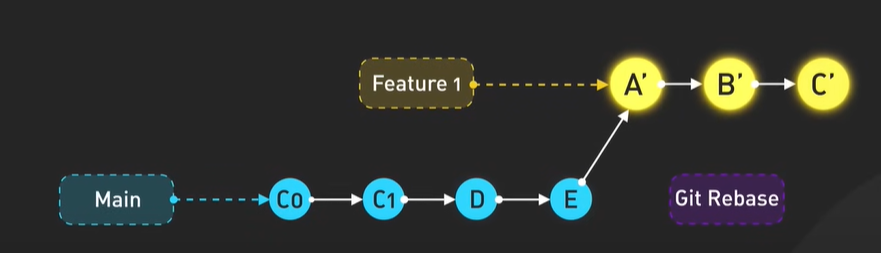
Ví dụ muốn merge tất cả commit từ branch feature sang branch main

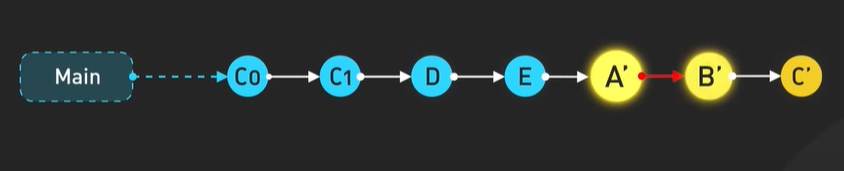
Git checkout main

Git merger feature

\*Git rebase

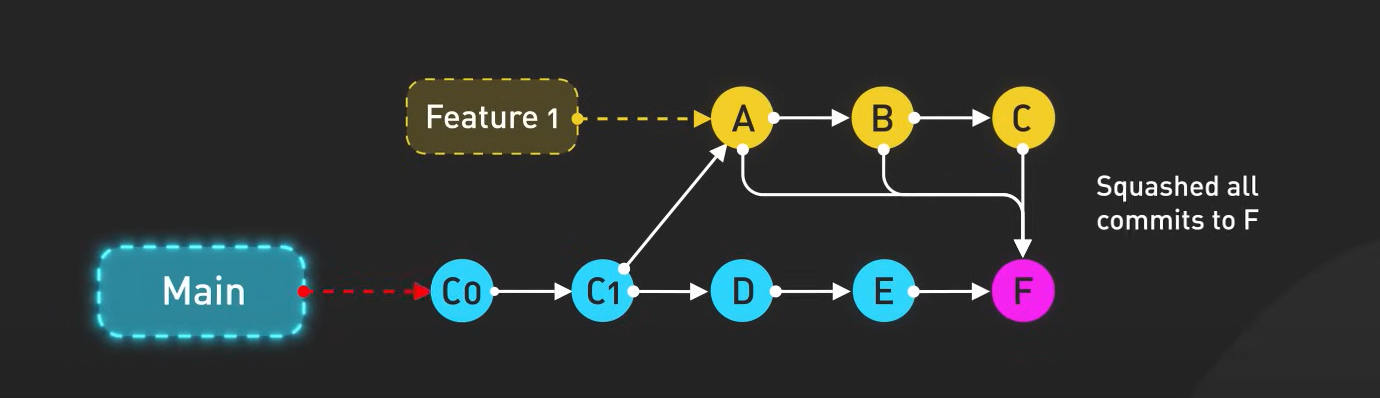


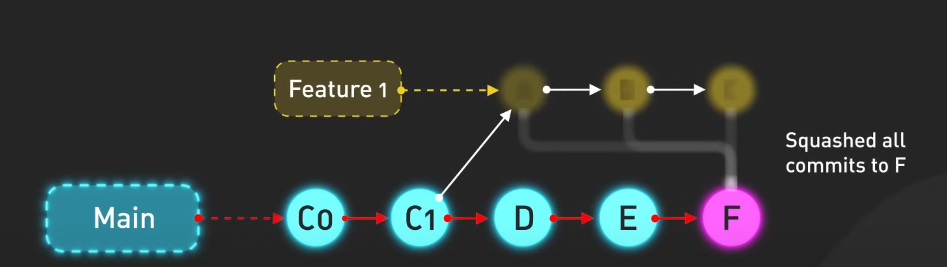




Chúng ta sẽ đưa các commit của nhánh feature lên 1 nhánh chính

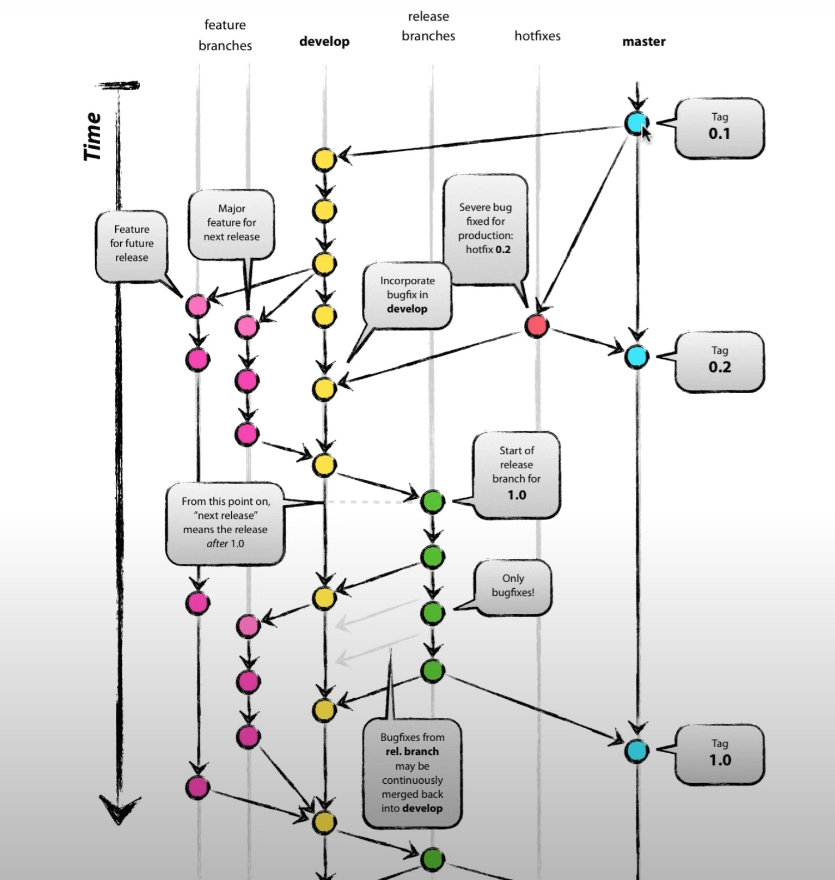
\*Git squash





Mình sẽ nén tất cả các commit lại thành 1 commit bao gôm tất cả các thay đổi và chỉ giữ lại commit cuối cùng.

**Git flow của tác giả Vincent Driessen**



\*Định nghĩa các nhánh và tác dụng của các nhánh đó:

**Master:**  push lên productions và không sửa, xóa code trực tiếp trên nhánh master(hoặc main) .

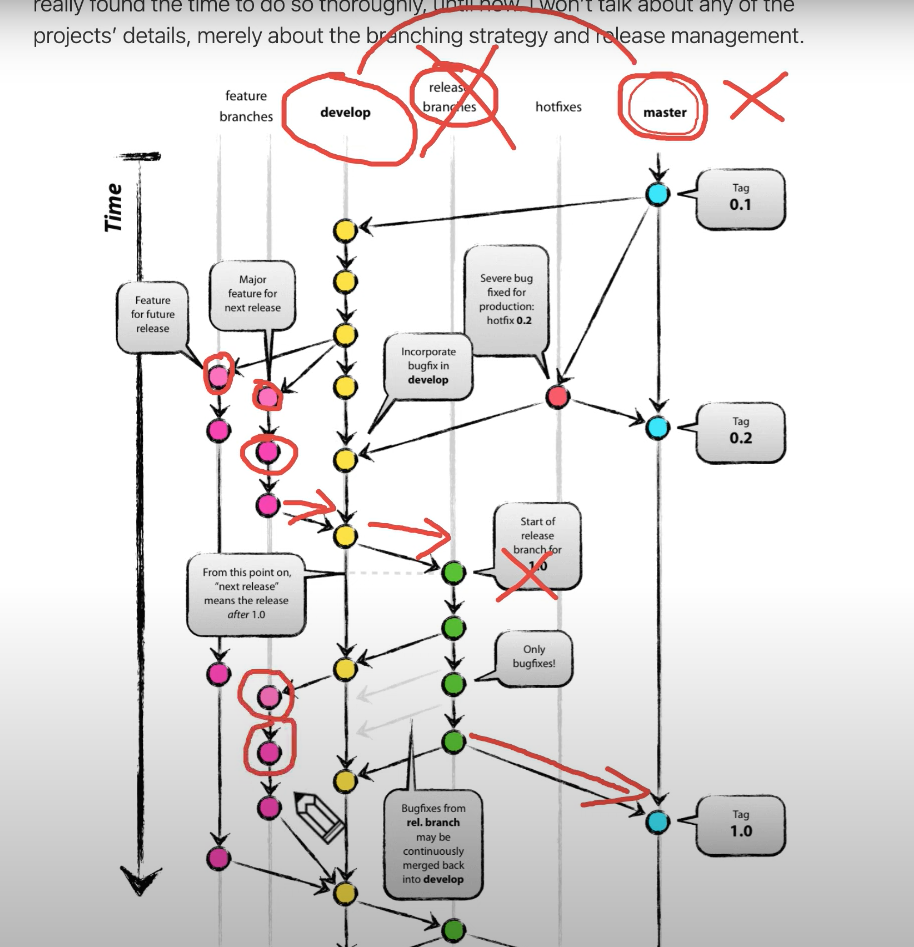
**Develop:** develop nhận chức năng được leader giao.

**Feature:** các develop sẽ chia nhỏ nhiệm vụ ra và sẽ push code lên ở đây.

**Release:** Hoàn thành các nhiệm vụ và đảm bảo rằng code chạy tốt rồi sẽ đẩy lên nhánh main và lên production.

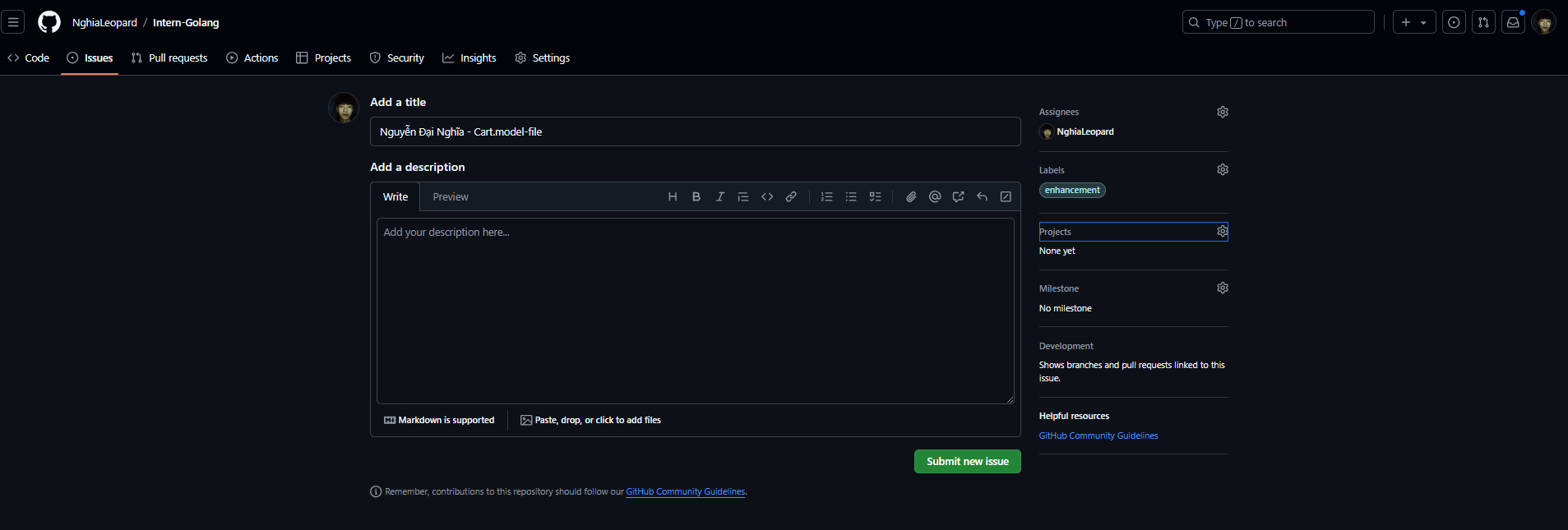
**Hotfixes:** Khi có lỗi sẽ đẩy từ branch main sang production.

**Cách vận hành git:**



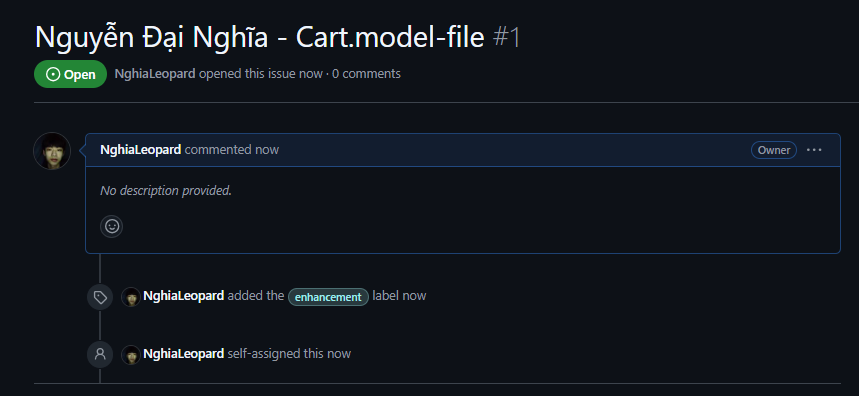
Bước 1: Develop nhận nhiệm vụ của sếp

Ví dụ: Leader sẽ tạo ra 1 issue và ghi rõ ai và nhận nhiệm vụ gì ở title + các tag ở bên phải



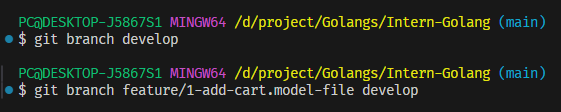
Đây sẽ là nhánh develop khi mình nhận nhiệm vụ của leader.

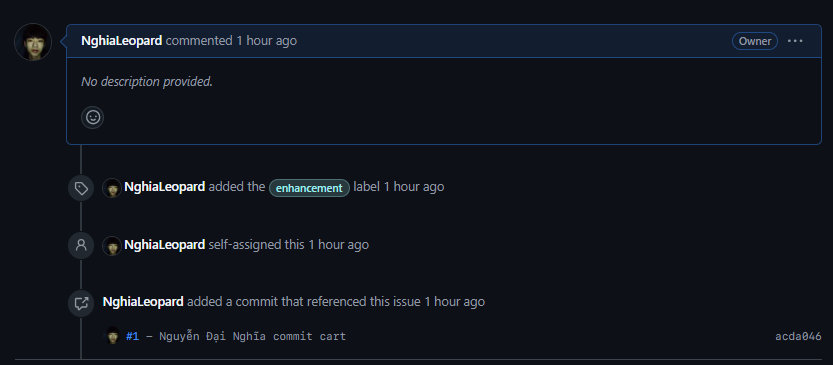
Leader sẽ tạo các issue để dev nhận và thực hiện nó.



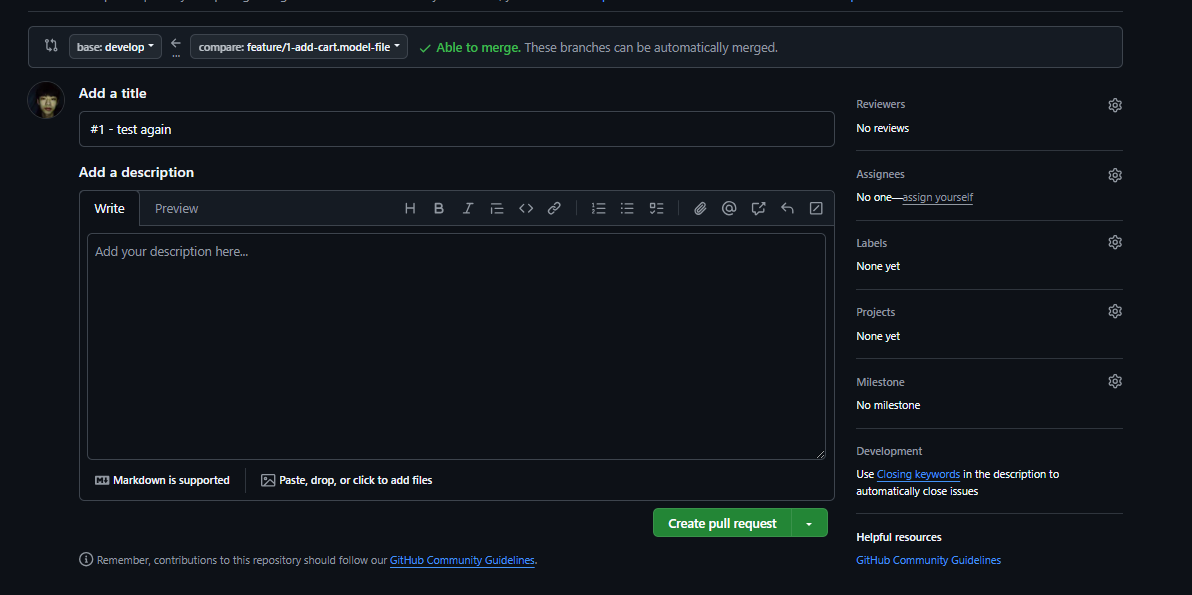
Bước 2: Tạo một nhánh mới để vô chuyển ngay vào làm việc:

Syntax: git checkout -b <tên\_nhánh\_mới> <nhánh\_nguồn>

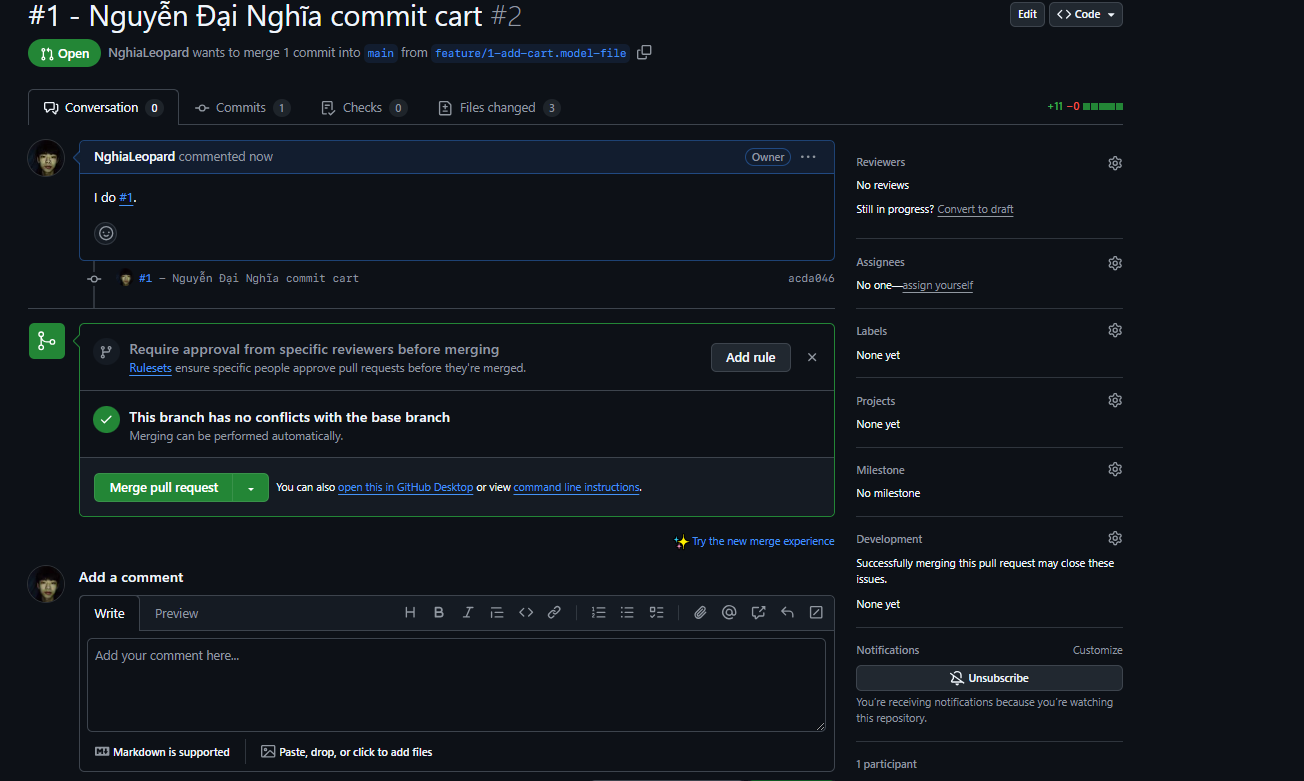


Khi push sẽ push vào issue mà leader tạo trong branch develop 

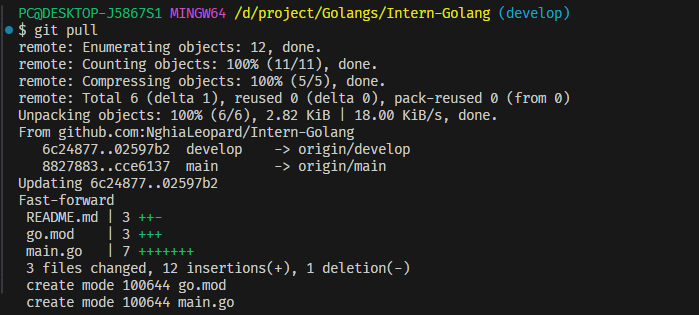
Mình sẽ comment , người review code và người làm



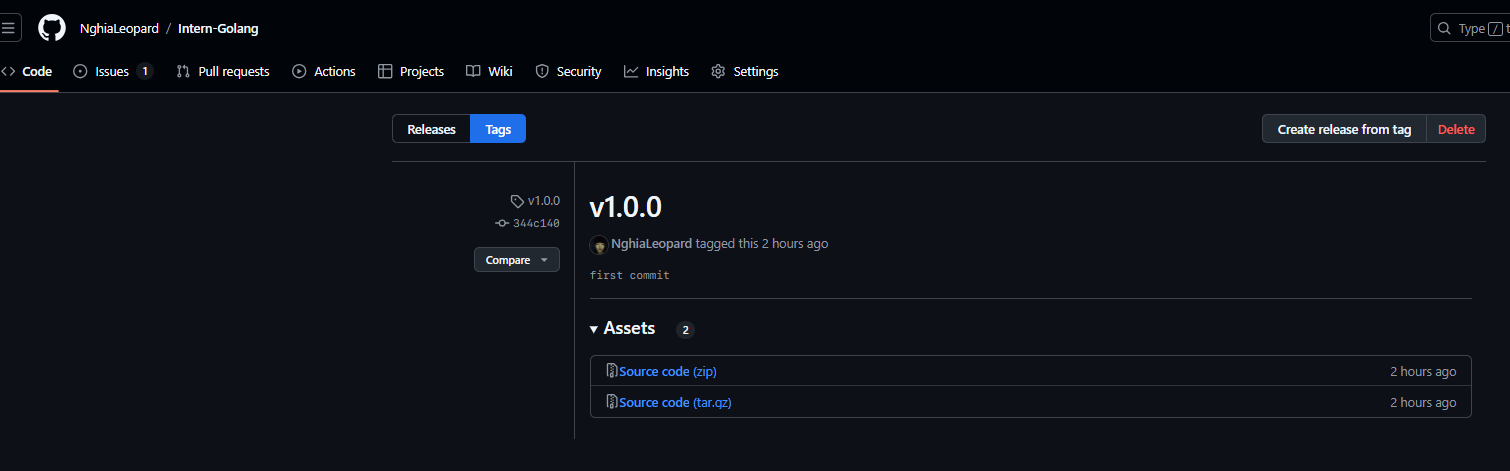
Step 3: Leader sẽ check ,comment và merger



Step 4: Pull code đã được merger ở branch develop rồi push lên branch release.



Tạo tag để đánh dấu cụ thể phiên bản của commit đã được release



Step 5: xóa các branch đã được đưa lên production

