

GIT - GITHUB

<https://cyberlearn.vn/>

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Mục Lục

- 01 Version Control System
- 02 Git và GitHub là gì
- 03 Cài đặt Git
- 04 Khởi tạo dự án mới
- 05 Tham gia vào dự án có sẵn
- 06 Phát triển dự án
- 07 Branch là gì

Version Control System

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Tại sao cần quản lý mã nguồn?

➤ Các vấn đề thường gặp khi quản lý mã nguồn:

- Khi làm việc nhóm, lúc cần merge code nằm ở các máy lại với nhau ta thường dùng cách gì?
- Nếu có 2 thành viên cùng sửa trên một file, và một người vô tình xóa mất code của người kia, ta phải làm sao phục hồi lại code cũ?



➡ Cần các hệ thống giúp quản lý phiên bản mã nguồn như Git

Version Control System

- Hệ thống giúp lưu trữ source code và có thể theo dõi sự thay đổi của code.
- Hỗ trợ nhiều người dùng làm việc trên cùng 1 source
- Kiểm tra những thay đổi và biết được ai là người thay đổi
- Revert các thay đổi, đưa code trở về các phiên bản cũ, không lo bị mất code

Git và GitHub là gì?

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Git là gì?



- Git là một trong những VCS , vốn được phát triển nhằm quản lý mã nguồn của Linux.
- Mỗi người sẽ tự làm việc trên máy tính của mình, sau khi thay đổi code, sẽ tiến hành cập nhật thay đổi lên source code chung thông qua một số quy trình

GitHub là gì?



- Trang web giúp lưu trữ các kho dữ liệu, có thể truy cập từ mọi nơi
- Cho phép nhiều người có thể làm việc trên cùng 1 source code

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Sơ đồ hoạt động giữa Git và GitHub

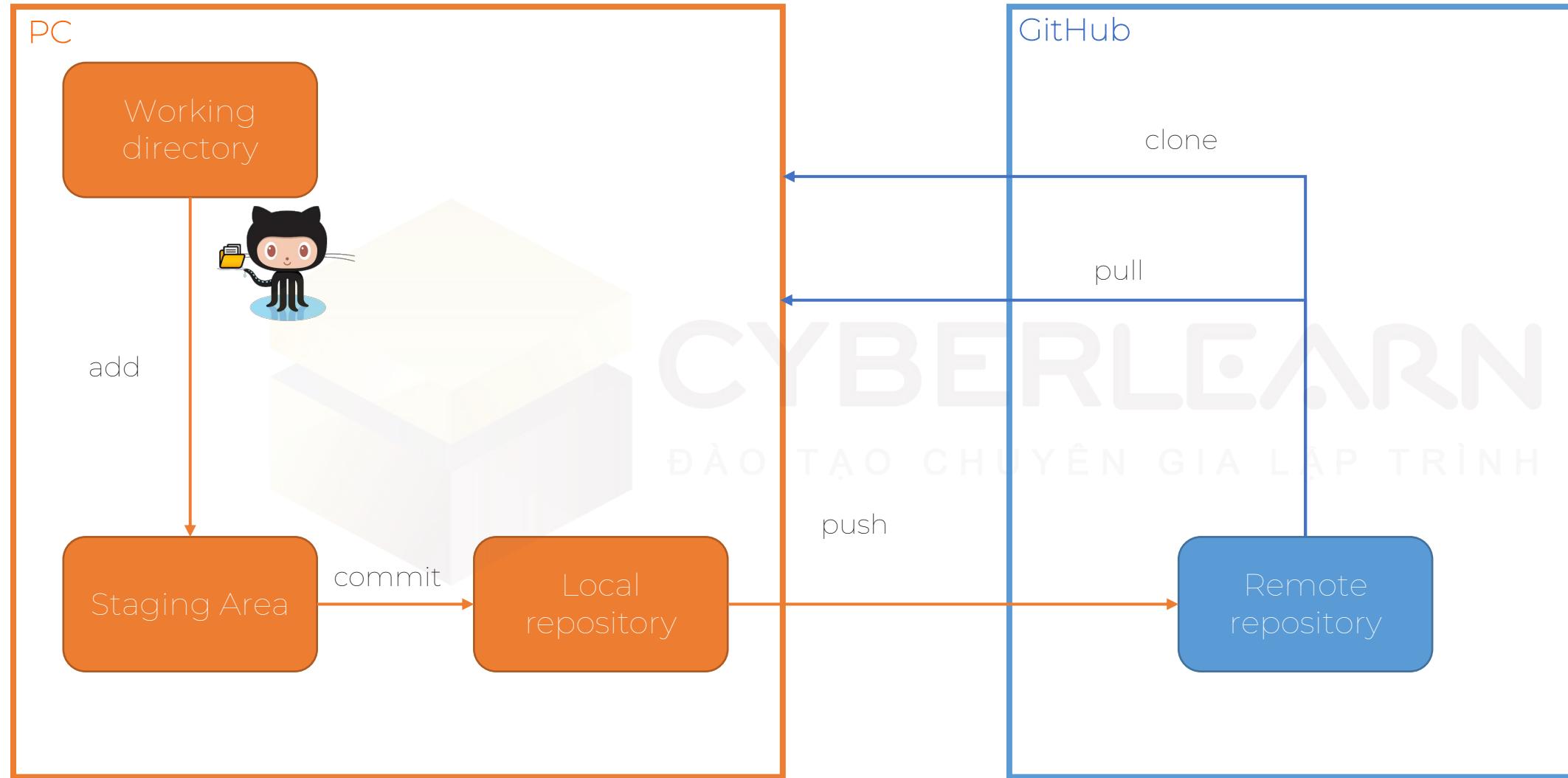
Các khái niệm cơ bản:

- Repository: kho lưu trữ source code và tất cả các thay đổi
 - Local repository : kho chứa trên máy tính chúng ta, mỗi máy sẽ là một repo
 - Remote repository : kho dữ liệu chứa trên server (kho lưu trữ online), có thể truy cập từ bất kì đâu

Sơ đồ hoạt động giữa Git và GitHub

- Staging Area: là một khu vực chứa thông tin các files sẽ được chuẩn bị cho quá trình commit
- Commit: mỗi lần thay đổi source code, tiến hành commit để tạo ra 1 phiên bản lưu trữ lại sự thay đổi, khi có lỗi thì có thể phục hồi code trở về version trước đó

Sơ đồ hoạt động giữa Git và GitHub



Thực hành Git



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

01

Cài đặt Git

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Cài đặt Git

- Mac:

Đối với macOS cần cài đặt theo 1 trong các cách ở link bên dưới.

<https://git-scm.com/download/mac>

- Window

<https://git-scm.com/>

Thiết lập cơ bản

Lần đầu tiên sử dụng git cho laptop, cần khai báo các thiết lập bên dưới.

- Cấu hình name và email

```
git config --global user.name "Your name"  
git config --global user.email "your@email.com"
```

Nếu muốn thiết lập lại name hoặc email mới thì chỉ cần gõ lại lệnh trên

- Kiểm tra sau khi cấu hình

```
git config --global user.name  
git config --global user.email
```



02

Khởi tạo dự án mới

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Khởi tạo dự án mới

- Đối với người khởi tạo dự án, thực hiện các quy trình sau:
 1. Tạo remote repository (GitHub)
 2. Tạo local repository (PC)
 3. Kết nối local repository với remote repository
 4. Khai báo các files chuẩn bị đưa lên remote repository
 5. Xác nhận các files cần đưa lên remote repository
 6. Đẩy lần đầu lên remote repository



Tạo dự án ở local và đẩy lên GitHub



Khởi tạo repository local
`git init`



Khai báo các file cần thêm vào local repository
`git add [tên file]` (thêm file cụ thể) hoặc `git add .` (thêm tất cả files)



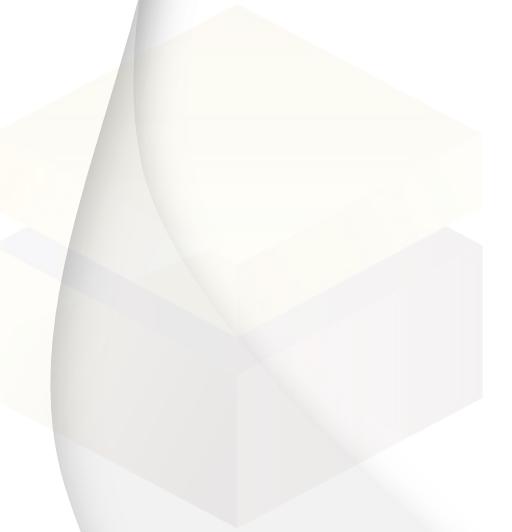
Xác nhận files cần thêm và lý do thêm files
`git commit -m "lý do"`



kết nối local repository với remote repository
`git remote add origin [đường dẫn trên github]`



Đẩy files lên remote repository
`git push -u origin master`



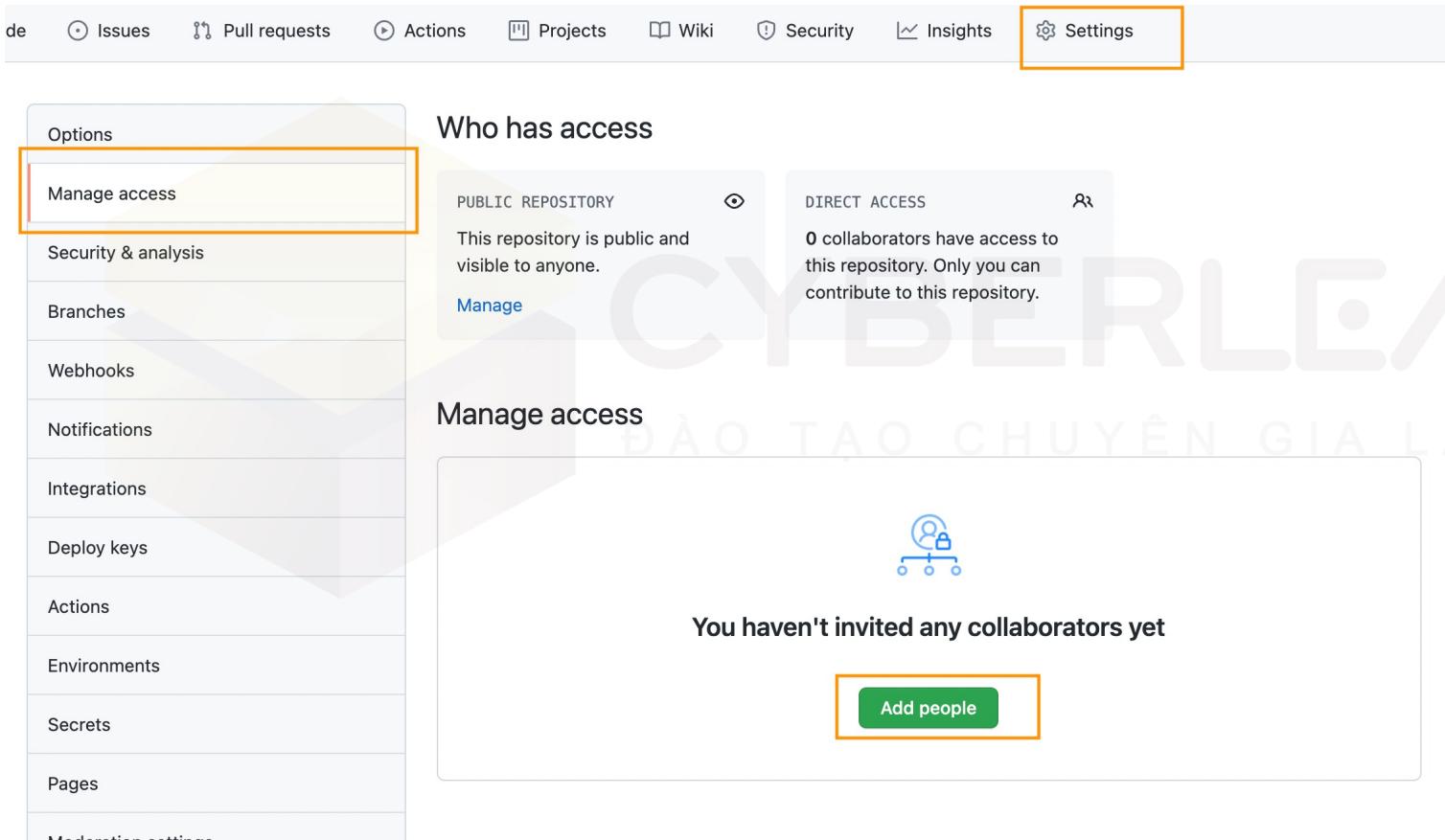
03

Tham gia vào dự án
có sẵn

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Tham gia vào dự án có sẵn

1. Người tạo dự án sẽ mời user vào dự án trên GitHub



Tham gia vào dự án có sẵn

2. Người tham gia sao chép dự án về local





04

Phát triển dự án

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Phát triển dự án

- Trước khi tiến hành chỉnh sửa phải cập nhật local repository để đảm bảo các files của local repo và remote repo giống nhau

```
git pull
```

- Sau khi chỉnh sửa, tiến hành đẩy thay đổi lên remote repository (GitHub)

```
git add [tenfile]
```

```
git commit -m "lý do thay đổi"
```

```
git push
```

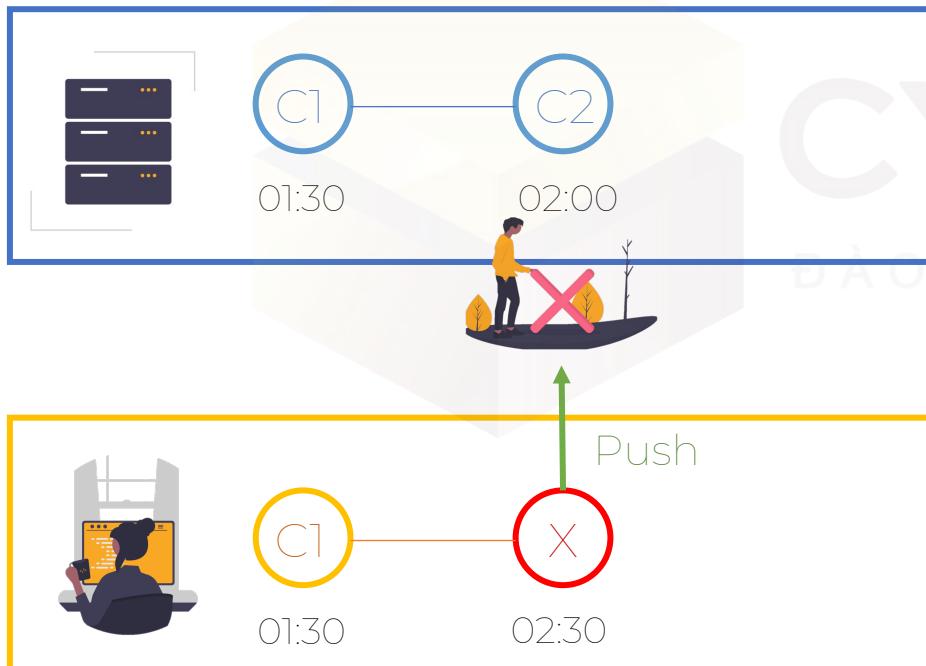
04

Branch là gì

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

➤ Lỗi xung đột (conflict) khi push

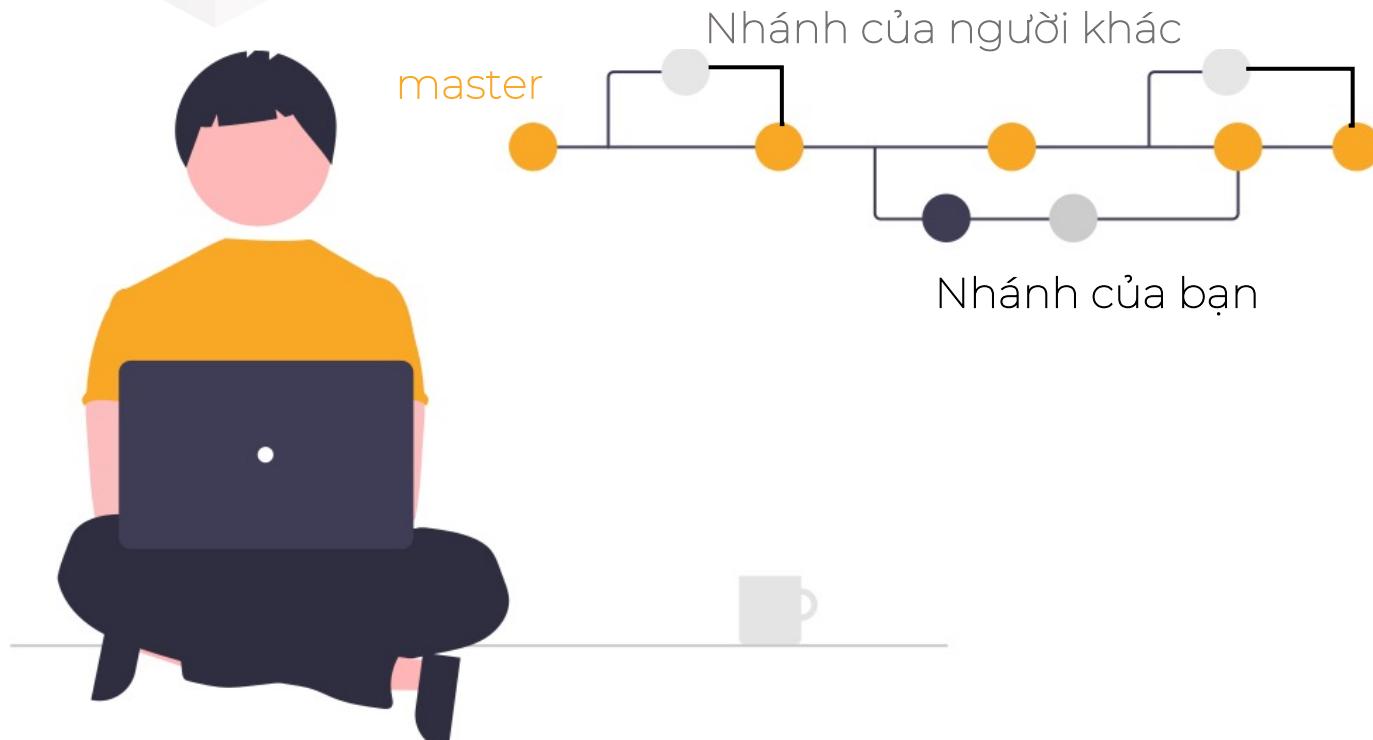
Trong khi bạn đang chỉnh sửa file ở local repository của bạn, nếu có một commit khác được đẩy lên remote repository, thì đến khi bạn đẩy code sẽ bị từ chối và xuất hiện lỗi như bên dưới:



```
MacBook-Pro-čua-THANH:GitProject thanhnguyen$ git push  
To https://github.com/khloe1425/GitProject.git  
! [rejected]          master -> master (non-fast-forward)  
error: failed to push some refs to 'https://github.com/khloe1425/GitProject.git'  
hint: Updates were rejected because the tip of your current branch is behind  
hint: its remote counterpart. Integrate the remote changes (e.g.  
hint: 'git pull ...') before pushing again.  
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

➤ Làm việc với branch để hạn chế xung đột

- Khi có quá nhiều người cùng làm việc trên một nhánh chính (master), dễ dẫn tới xung đột khi push hoặc đè mất code của nhánh chính
- Ta phải tiến hành chia nhánh, mỗi nhánh thực hiện một chức năng khác nhau, sau đó gộp lại vào nhánh chính, để giữ nhánh chính được toàn vẹn



➤ Các lệnh cơ bản với branch

- Tạo branch

```
git branch [tennhanh]
```

- Chuyển branch

```
git checkout [tennhanh]
```

- Tạo và chuyển branch

```
git checkout -b [tennhanh]
```

- Đẩy nhánh mới lên remote repository

```
git push  
git push --set-upstream origin [tennhanh]
```

- Xoá branch

```
git branch -D [tennhanh]
```

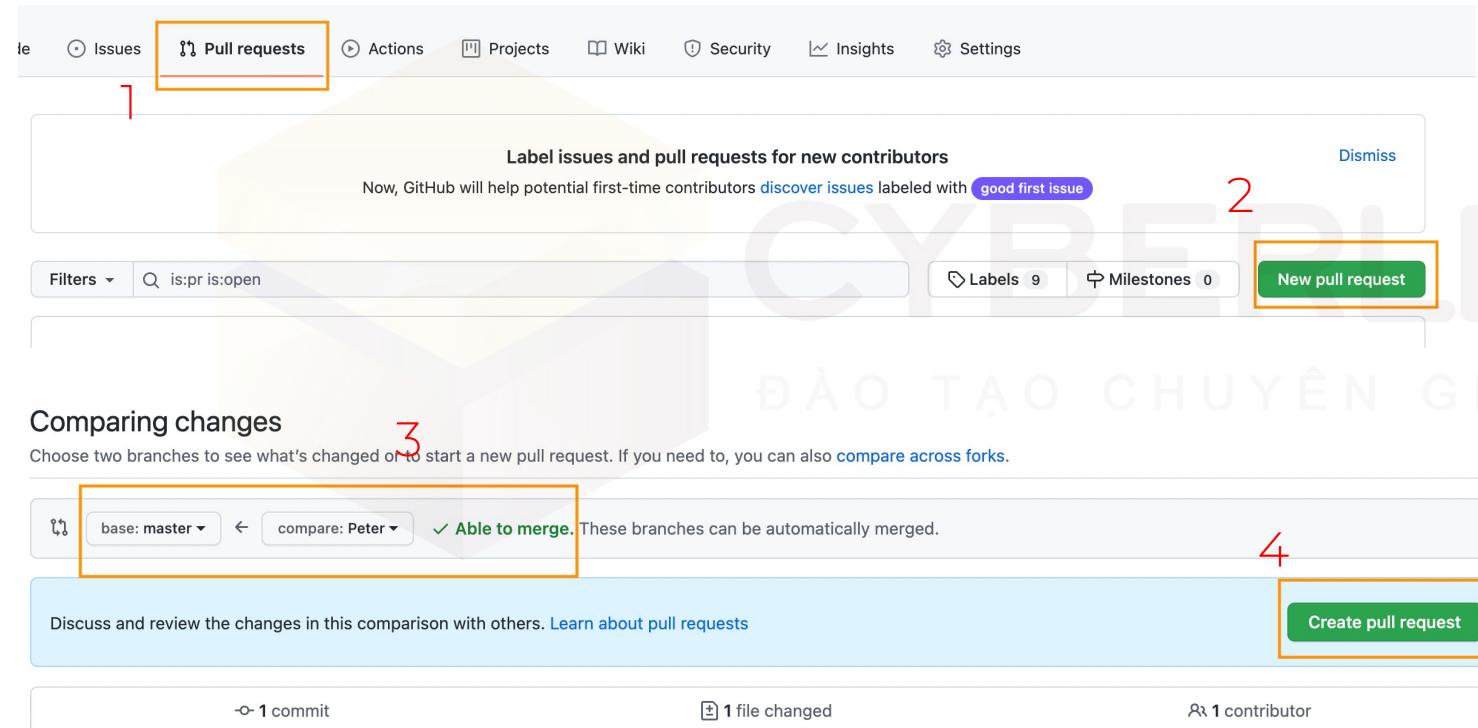
- Kiểm tra nhánh

```
git branch
```

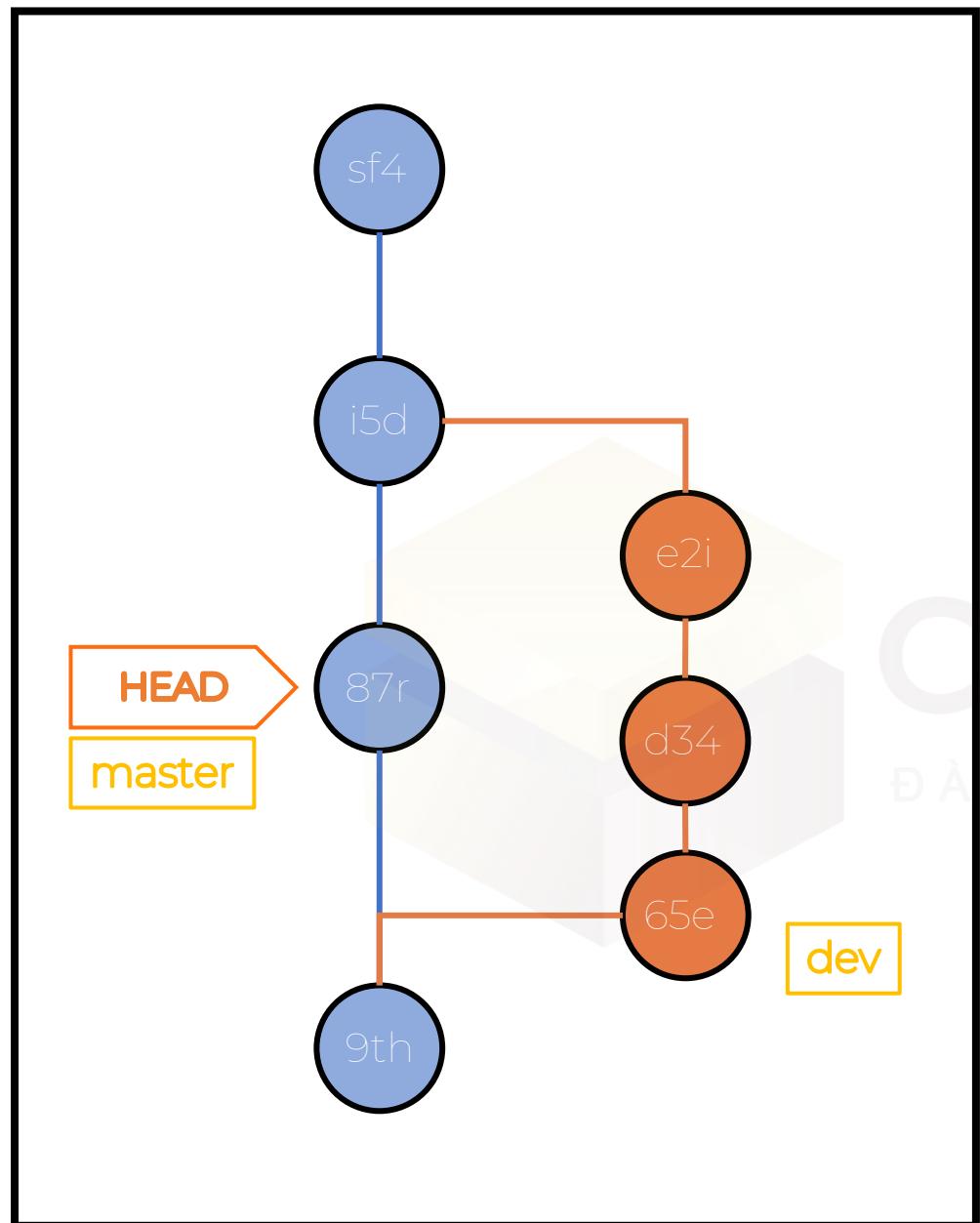
Merge nhánh

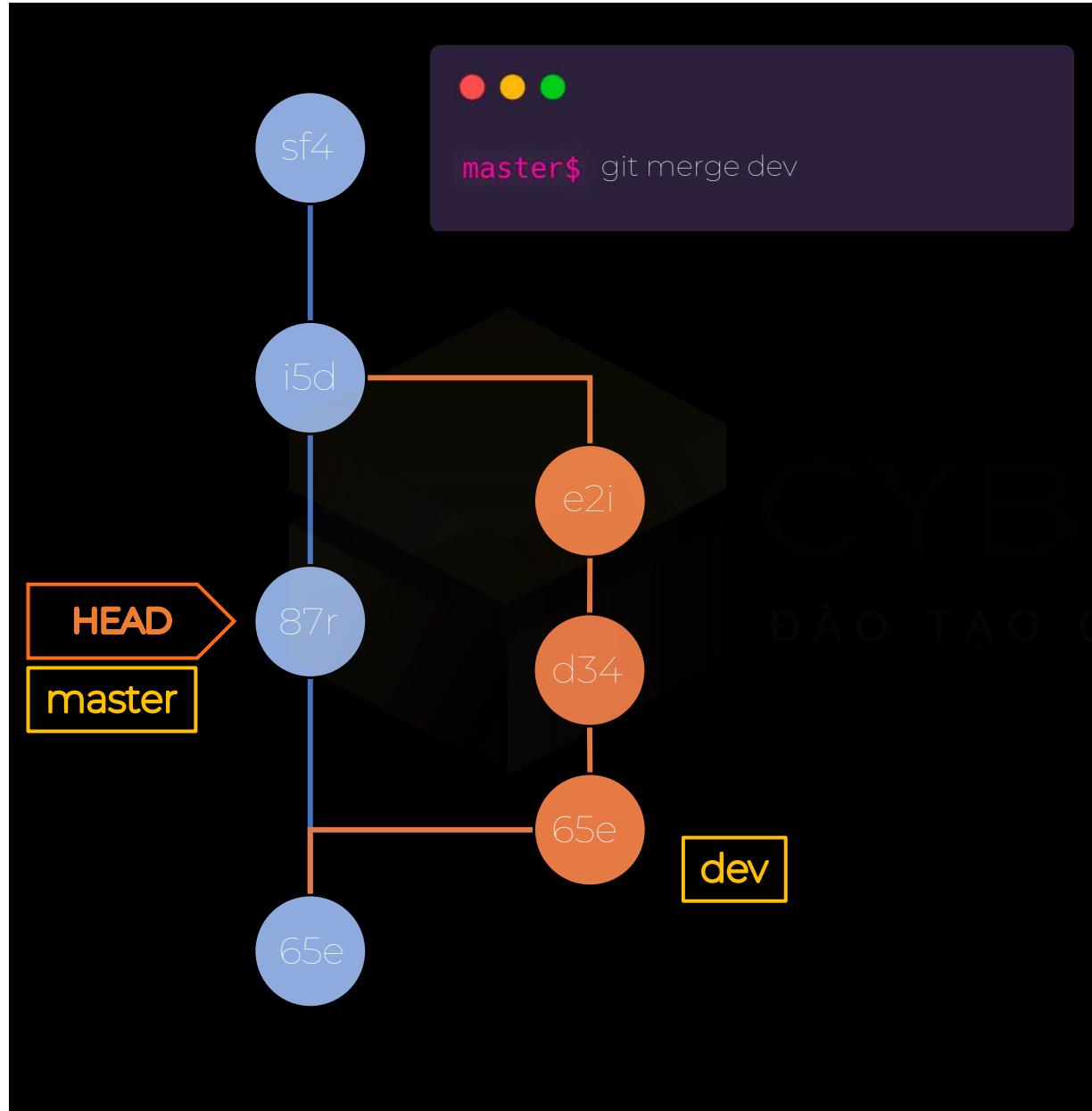
Để merge nhánh riêng vào nhánh master, ta có các cách sau

- Cách 1: Tạo pull request trên github và tiến hành merger nhánh



Merge nhánh





Merge nhánh

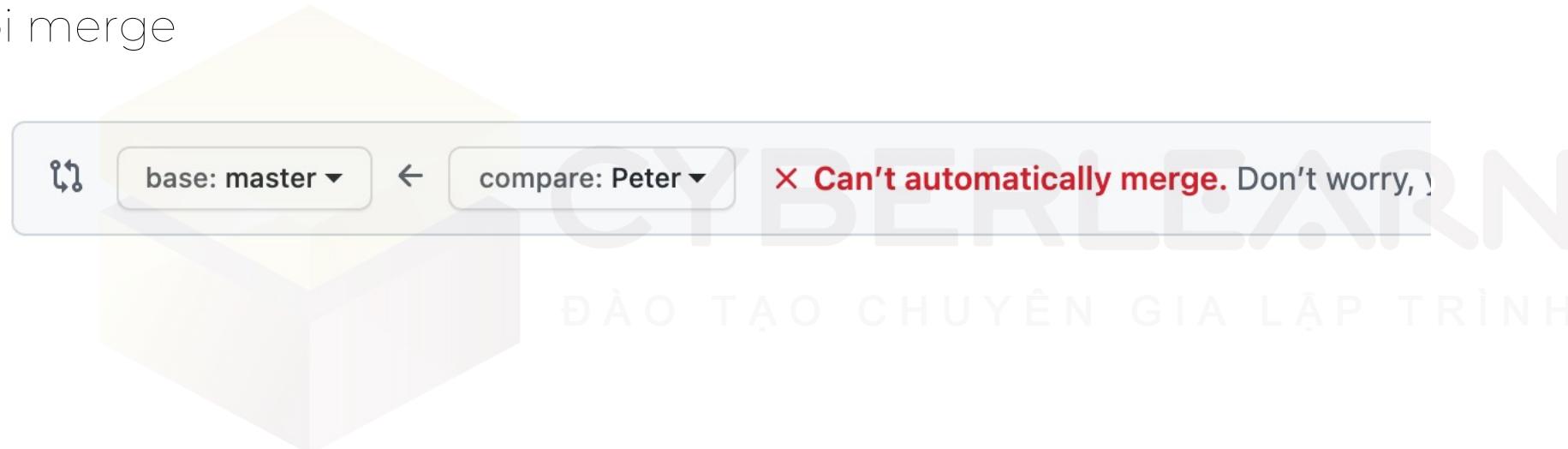
- Cách 2: Merge nhánh bằng lệnh
 1. Khi merge vào master phải checkout về master


```
git checkout master
```
 2. Tiến hành merge


```
git merge [tên nhánh]
```
 3. Commit master lên remote repository

Xung đột khi merge nhánh

Xung đột xảy ra khi nhánh master bị nhánh khác merge vào trước bạn và làm thay đổi. Sau khi bạn hoàn thành chức năng và cần merge vào master thì sẽ bị từ chối merge

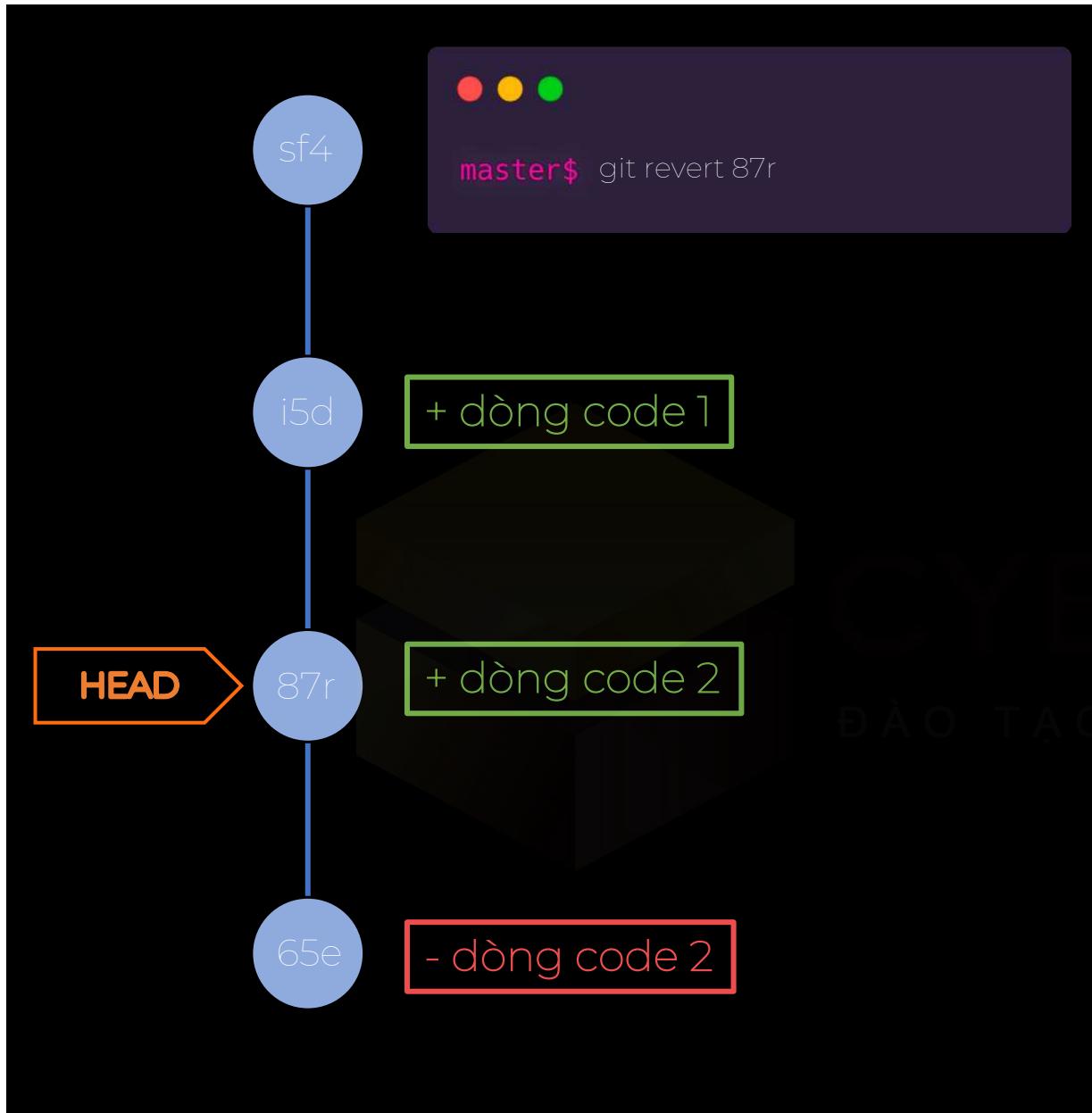


Merge nhánh

- Giải quyết xung đột khi merge bằng lệnh
 - 1. Cập nhật code mới từ remote master về local branch của bạn
`git pull origin master`
 - 2. So sánh và xử lý xung đột code ở VS code
 - 3. Commit code sau khi sửa lỗi trên remote branch riêng của bạn
 - 4. Tiến hành tạo pull request trên GitHub và merge vào master

Revert

- Revert sẽ tạo một commit mới, trong đó sẽ loại bỏ thay đổi của commit cũ.



BERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

THANK YOU
CYBERLEARN
https://cyberlearn.vn/

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH