

	<b>VIETTEL AI RACE</b>	TD188
	<b>GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIỆN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ</b>	Lần ban hành: 1

Git hoạt động bằng cách lưu trữ, theo dõi và quản lý sự thay đổi của mã nguồn trong dự án phần mềm. Nền tảng cung cấp các công cụ để làm việc song song, quản lý nhánh và hợp nhất các thay đổi một cách linh hoạt và hiệu quả. Người học lập trình nên nắm bắt các thuật ngữ quan trọng trong Git.

## 1. Một số khái niệm cơ bản cần biết

### 1.1 Git là gì

Git là một hệ thống quản lý phiên bản phân tán được sử dụng rộng rãi trong phát triển phần mềm. Nền tảng cho phép các nhà phát triển lưu trữ mã nguồn và theo dõi lịch sử thay đổi của mã nguồn đó theo thời gian.

Ở mức cơ bản, Git hoạt động bằng cách lưu trữ các phiên bản khác nhau của dự án phần mềm trong một kho lưu trữ. Hệ thống cho phép người dùng theo dõi sự thay đổi giữa các phiên bản, khôi phục lại các phiên bản trước đó và làm việc song song trên cùng một tệp tin.



Git có tính phân tán mạnh mẽ hơn những công cụ khác. Điều này có nghĩa là mỗi người dùng đều nhận bản sao đầy đủ của toàn bộ kho lưu trữ, không phụ thuộc vào kết nối mạng hoặc máy chủ tập trung. Hệ thống cho phép các nhà phát triển làm việc nhanh chóng và phòng tránh mất dữ liệu do mất kết nối với máy chủ.

Hệ thống Git được sử dụng rộng rãi trong cộng đồng phát triển phần mềm vì tính linh hoạt, mạnh mẽ. Đặc biệt, công cụ có khả năng tương tác tốt với nhiều dịch vụ lưu trữ mã nguồn trực tuyến khác nhau như GitHub, Bitbucket và GitLab.

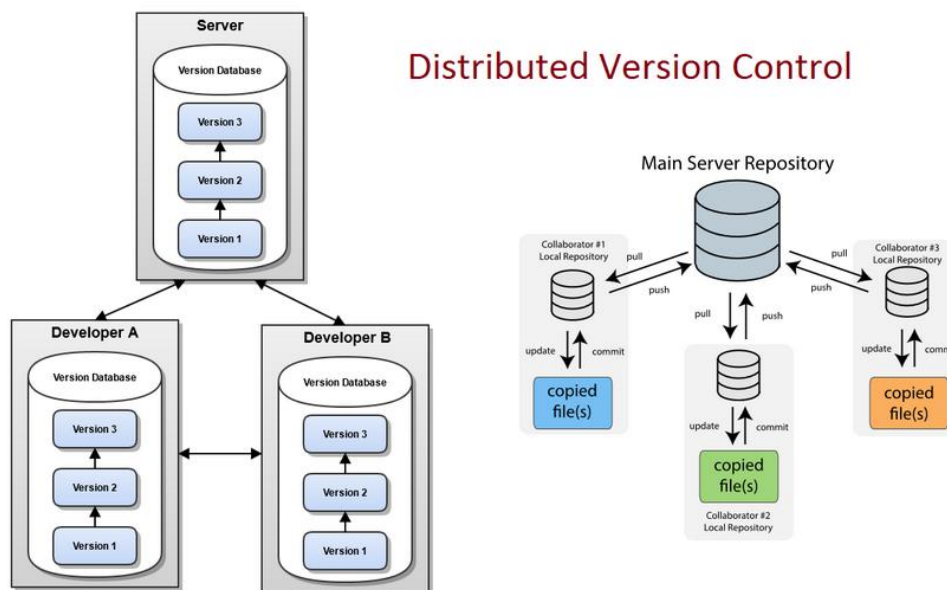
### 1.2 Version Control System – VCS là gì?

Hệ thống quản lý phiên bản (VCS) hay còn gọi là hệ thống quản lý mã nguồn. Đây là một hệ thống có tác quản lý, theo dõi và kiểm soát sự thay đổi của mã nguồn

	<b>VIETTEL AI RACE</b>	TD188
	<b>GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIỆN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ</b>	Lần ban hành: 1

phần mềm. Mục tiêu chính của VCS là theo dõi lịch sử thay đổi của tất cả các tập tin và cấu trúc thư mục trong dự án phần mềm.

VCS cho phép các nhà phát triển làm việc song song trên cùng một dự án mà không gặp xung đột. Công nghệ hỗ trợ lưu trữ phiên bản của mã nguồn tại mọi thời điểm và khôi phục lại các phiên bản trước đó nếu cần.



Ngoài ra, VCS cũng giúp quản lý việc tích hợp mã nguồn từ nhiều nguồn khác nhau và tạo ra các nhánh (branch) của mã nguồn. Mục đích nhằm phát triển song song các tính năng hay chế độ sửa lỗi mà không ảnh hưởng đến nhau.

Có hai loại chính của VCS là hệ thống quản lý phiên bản cục bộ (Local Version Control System) và hệ thống quản lý phiên bản phân tán (Distributed Version Control System). Git là một ví dụ điển hình của hệ thống quản lý phiên bản phân tán. Trong khi đó, Subversion (SVN) chính là một hệ thống quản lý phiên bản cục bộ phổ biến khác.

## 2. Những tác dụng phổ biến của VCS

Hệ thống quản lý phiên bản (VCS) có tác dụng quản lý và theo dõi sự thay đổi của mã nguồn trong dự án phần mềm. Cụ thể, VCS thực hiện các chức năng sau:


	<b>VIETTEL AI RACE</b>	<b>TD188</b>
	<b>GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIỆN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ</b>	Lần ban hành: 1

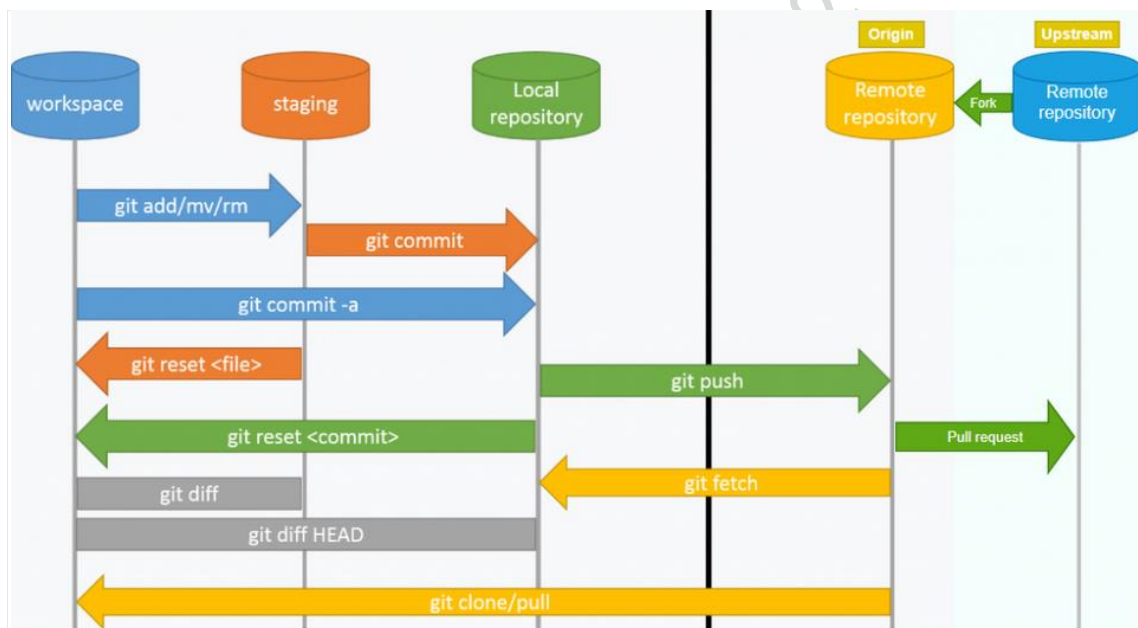


- Lưu trữ phiên bản của mã nguồn: VCS lưu trữ tất cả các phiên bản khác nhau của mã nguồn, cho phép nhà phát triển quay lại các phiên bản trước đó nếu cần.
- Theo dõi lịch sử thay đổi: Nền tảng tự động ghi lại tất cả các thay đổi được thực hiện trên mã nguồn, cung cấp lịch sử rõ ràng về sự phát triển của dự án.
- Quản lý tích hợp từ nhiều nguồn: VCS cho phép tích hợp mã nguồn từ nhiều nguồn khác nhau, giúp tăng tính linh hoạt và hiệu quả trong quá trình phát triển.
- Phát triển song song: Hệ thống hỗ trợ nhà phát triển làm việc song song trên cùng một dự án mà không gặp xung đột, thông qua việc tạo và quản lý các nhánh (branch) riêng biệt.
- Hỗ trợ hợp nhất mã nguồn: VCS áp dụng chế độ hợp nhất (merge) mã nguồn từ các nhánh khác nhau, giúp nhà phát triển kết hợp các tính năng và sửa lỗi một cách hiệu quả.

### 3. Nguyên lý hoạt động chung của Git

Git hoạt động bằng cách lưu trữ dự án phần mềm trong một kho lưu trữ (repository) và theo dõi lịch sử các thay đổi của mã nguồn trong dự án. Quá trình hoạt động chính của Git được giới thiệu như sau:

	VIETTEL AI RACE	TD188
	GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIỆN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ	Lần ban hành: 1



### 3.1 Khởi tạo kho lưu trữ (Repository)

Người dùng sẽ khởi tạo một kho lưu trữ (Repository) để lưu trữ mã nguồn của dự án. Kho lưu trữ có thể ở địa phương trên máy tính hoặc trên các dịch vụ lưu trữ mã nguồn trực tuyến như GitHub, Bitbucket, GitLab.

### 3.2 Làm việc trên mã nguồn

Người dùng thực hiện các thay đổi trên mã nguồn một cách trực tiếp. Trong đó bao gồm việc thêm, sửa đổi và xóa tập tin trong dự án phần mềm.

### 3.3 Theo dõi thay đổi

Sau khi thực hiện các thay đổi, người dùng sẽ sử dụng lệnh "Git Add" để chấp nhận những thay đổi cụ thể và chuẩn bị chúng để được Commit vào lịch sử.

### 3.4 Commit thay đổi

Người dùng sẽ sử dụng lệnh "Git Commit" để lưu trữ các thay đổi đã được chấp nhận vào lịch sử của dự án. Mỗi lần Commit sẽ tạo ra một Checkpoint để lưu trữ các tập tin và thay đổi tương ứng.

	<b>VIETTEL AI RACE</b>	TD188
	<b>GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIỆN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ</b>	Lần ban hành: 1



### 3.5 Công việc với nhánh (Branch)

Git cho phép người dùng tạo và quản lý các nhánh (Branch) để phát triển tính năng, sửa lỗi mà không ảnh hưởng đến nhau. Việc làm việc trên các nhánh riêng biệt giúp tăng tính linh hoạt và an toàn trong quá trình phát triển.

### 3.6 Hợp nhất (Merge) và xung đột

Git cho phép người dùng hợp nhất các thay đổi từ các nhánh khác nhau lại với nhau. Trong trường hợp có xung đột (Conflict), Git sẽ cung cấp cách giải quyết xung đột thông qua việc so sánh và hợp nhất sự khác biệt giữa các phiên bản.

## 4. Lợi ích đặc biệt khi sử dụng Git

Khi sử dụng phần mềm Git, người dùng có thể hưởng lợi ích từ các tính năng cơ bản, bao gồm:



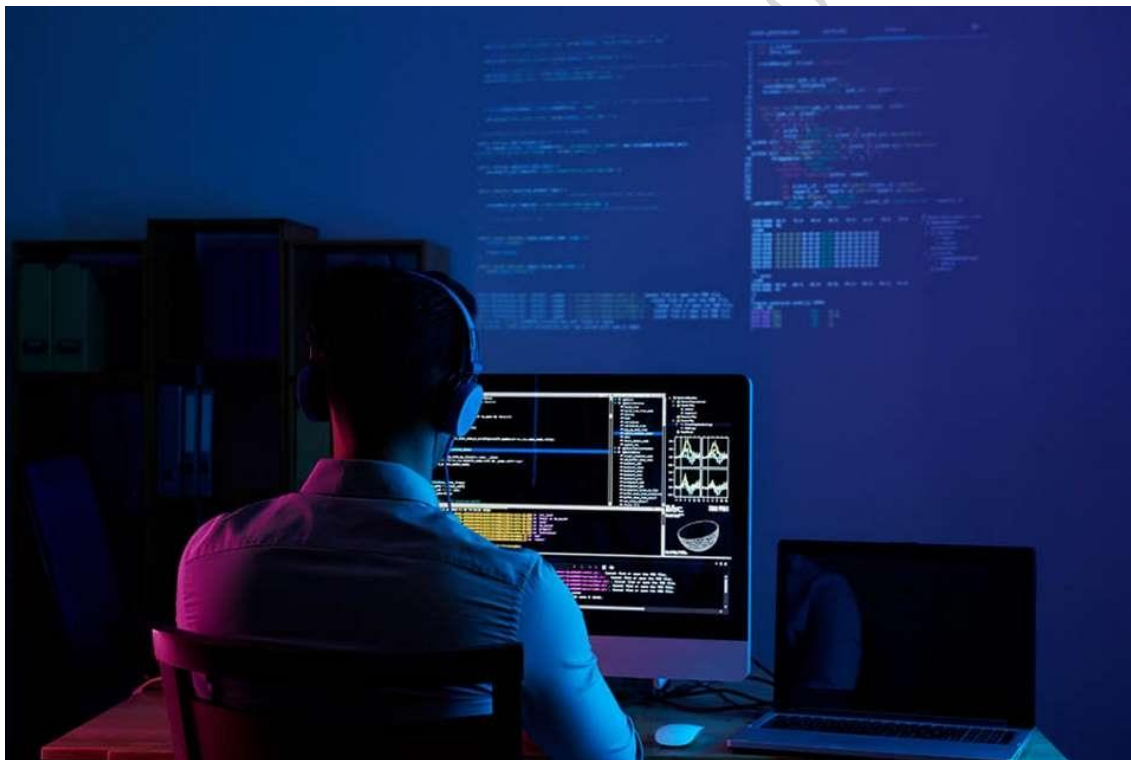
	VIETTEL AI RACE	TD188
	GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIỆN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ	Lần ban hành: 1



- Hệ thống cho phép người dùng theo dõi lịch sử thay đổi của mã nguồn, bao gồm việc xem thông tin về ai đã thực hiện thay đổi, khi nào và thay đổi gì. Điều này giúp trong việc xác định nguyên nhân sự cố và theo dõi tiến độ công việc.
- Công cụ hỗ trợ làm việc song song và độc lập trên cùng một dự án thông qua việc tạo và quản lý các nhánh (branch). Điều này tạo ra tính linh hoạt và an toàn khi thực hiện phát triển các tính năng và sửa lỗi.
- Git cung cấp cách thức hợp nhất các thay đổi từ các nhánh khác nhau về một nhánh chính một cách hiệu quả, giúp người dùng kết hợp các tính năng và sửa lỗi một cách dễ dàng.
- Người dùng có thể sao lưu mã nguồn và lịch sử thay đổi trên các máy chủ từ xa hoặc dịch vụ lưu trữ mã nguồn trực tuyến. Điều này giúp bảo vệ mã nguồn khỏi mất mát dữ liệu.
- Git cung cấp cơ chế để phối hợp làm việc trong nhóm, cho phép các nhà phát triển cùng làm việc trên các phiên bản khác nhau của dự án một cách hiệu quả.

## 5. Tổng hợp các thuật ngữ quan trọng

	VIETTEL AI RACE	TD188
	GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIỆN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ	Lần ban hành: 1



### • Branch

Branch trong Git là một phiên bản song song của dự án phần mềm. Khi người dùng tạo một nhánh, họ đang tạo ra một bản sao của toàn bộ lịch sử và tập tin trong dự án.


Việc sử dụng nhánh cho phép người dùng phát triển tính năng mới, sửa lỗi hoặc thực hiện các thay đổi mà không ảnh hưởng đến phiên bản chính của dự án. Khi một tính năng hoặc sửa lỗi được hoàn thành trong nhánh, người dùng có thể hợp nhất (Merge) nhánh đó với nhánh chính của dự án để tích hợp các thay đổi mới vào phiên bản chính.

### • Commit

Trong Git, thuật ngữ "Commit" được sử dụng để mô tả hành động lưu trữ một bản sao mới của lịch sử thay đổi và tập tin của dự án. Khi người dùng thực hiện Commit, họ chấp nhận các thay đổi đã được "Staged" (chuẩn bị) và lưu trữ chúng vào lịch sử của dự án.

### • Checkout

Thuật ngữ được sử dụng để mô tả hành động chuyển đổi giữa các nhánh hoặc thay đổi "HEAD" (trạng thái hiện tại của repository). Khi thực hiện Checkout nghĩa là bạn đang chuyển đổi trạng thái làm việc của bạn từ một nhánh hoặc Commit sang một nhánh hoặc Commit khác.

	VIETTEL AI RACE	TD188
	GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIÊN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ	Lần ban hành: 1



- **Fetch**

"Fetch" trong Git được sử dụng để tải về mọi thay đổi mới từ Repository từ xa mà không thực hiện hợp nhất (Merge) với bất kỳ nhánh nào.

Quá trình Fetch giúp cập nhật thông tin về các nhánh và Commit từ Repository từ xa, nhưng không thay đổi trạng thái làm việc hiện tại của bạn. Điều này cho phép bạn xem xét các thay đổi từ Repository từ xa trước khi quyết định hợp nhất với nhánh hiện tại của bạn.

- **Fork**

Fork không phải là một thuật ngữ trực tiếp trong Git mà thường được sử dụng trong ngữ cảnh của dịch vụ lưu trữ mã nguồn trực tuyến như GitHub.

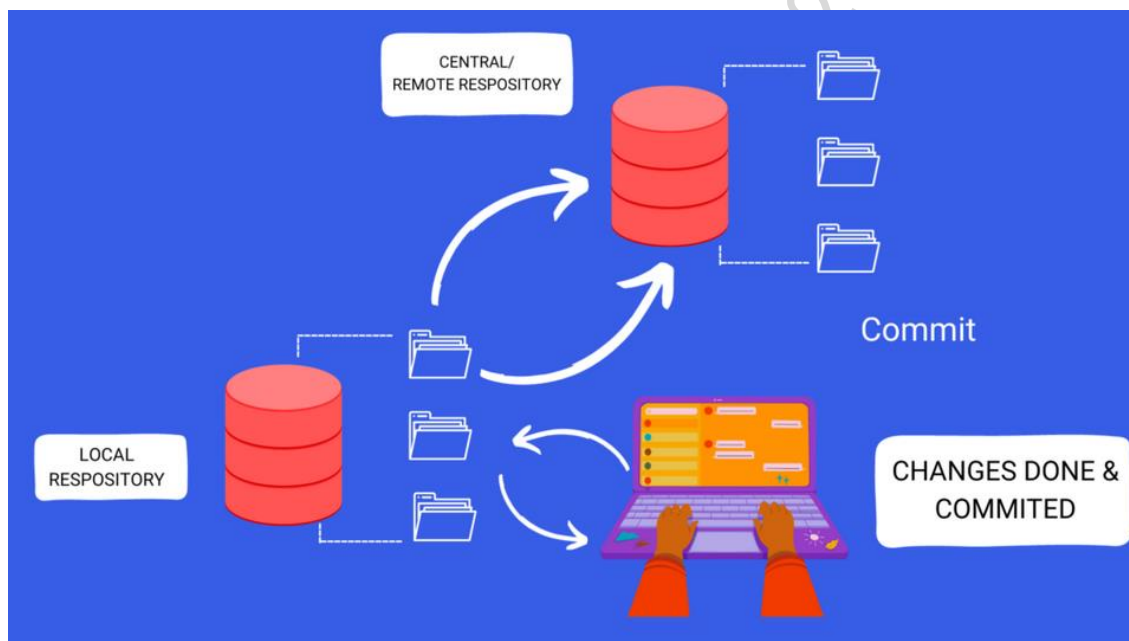
Khi Fork một Repository trên GitHub nghĩa là bạn đang tạo ra một bản sao độc lập của Repository đó trong tài khoản GitHub. Bản Fork có thể được sử dụng để đề xuất các thay đổi cho Repository gốc thông qua các Pull Request mà không ảnh hưởng trực tiếp đến Repository gốc.

- **Head**

Head là một tham chiếu tới phiên bản hiện tại của mã nguồn trong Repository. Công cụ thường trở tới Commit cuối cùng của nhánh hiện tại mà bạn đang làm việc. Khi bạn chuyển đổi nhánh hoặc thực hiện Checkout tới một Commit cụ thể, Head sẽ thay đổi để phản ánh trạng thái hiện tại của bạn trong Repository.



	VIETTEL AI RACE	TD188
	GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIỆN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ	Lần ban hành: 1



- **Index**

Index cũng được gọi là "Staging Area" và đây là nơi mà các thay đổi được chuẩn bị sẵn sàng để được Commit. Khi thay đổi các tập tin trong Repository thì bạn cần thêm các thay đổi đó vào "Index" trước khi Commit để chúng được lưu trữ vào lịch sử của Repository.

- **Merge**

Merge là quá trình kết hợp các thay đổi từ một nhánh Git vào nhánh hiện tại của bạn. Khi muốn tích hợp công việc đã thực hiện từ một nhánh phụ (ví dụ nhánh đang phát triển một tính năng mới) vào nhánh chính (thường là "Master" hoặc "Main"), bạn có thể sử dụng thao tác "Merge" để thực hiện việc này.

- **Origin**

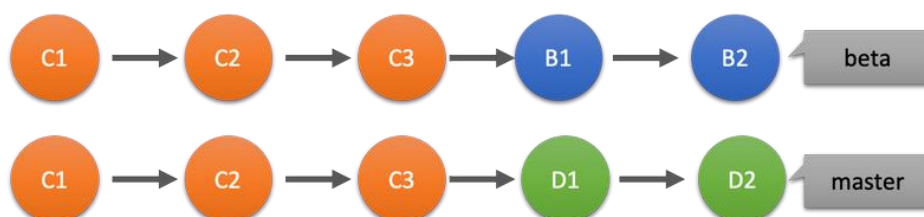
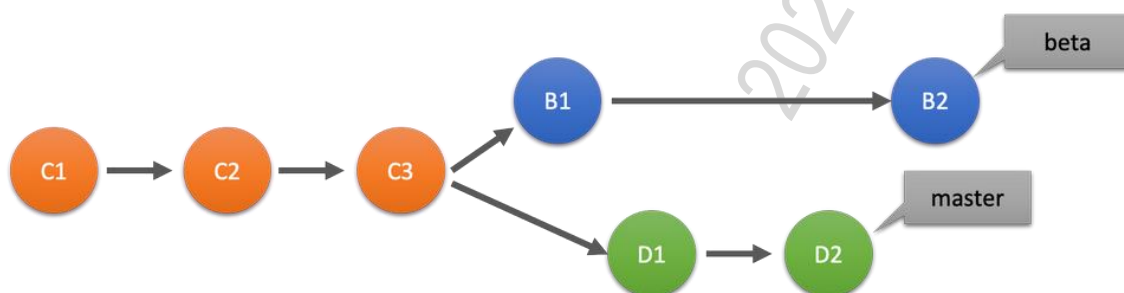
Trong Git, Origin là tên được sử dụng mặc định cho Repository từ xa, thường là Repository gốc mà bạn đã sao chép (clone) dự án từ đó.

Khi bạn sao chép một repository từ xa, Git tự động thiết lập một Remote (Repository từ xa) mặc định và đặt tên là "Origin". Khi bạn Push (đẩy) hoặc pull (kéo) các thay đổi đến Repository từ xa, bạn sẽ sử dụng tên "Origin" để định danh đến Repository đó.

- **Master**

Master là tên chuẩn của nhánh chính trong Git. Thường thì nhánh chính này được gọi là "Master" nhằm đại diện cho phiên bản ổn định nhất của dự án. Tuy nhiên, sau các thay đổi gần đây, nhiều người đang chuyển sang sử dụng thuật ngữ khác như "Main" để thay thế cho "Master" để tránh các từ ngữ mang theo ý nghĩa không tốt về mặt xã hội.

	VIETTEL AI RACE	TD188
	GIỚI THIỆU NHỮNG THUẬT NGỮ TRONG GIT CƠ BẢN ĐỂ LẬP TRUNG VIỆN ỨNG DỤNG CHUẨN XÁC VÀ HIỆU QUẢ	Lần ban hành: 1



## 6. Tạm kết

Những chia sẻ trong bài viết trên giới thiệu hàng loạt kiến thức cơ bản về Git. Đặc biệt hơn, người học lập trình có thể cập nhật rất nhiều thuật ngữ liên quan đến Git.