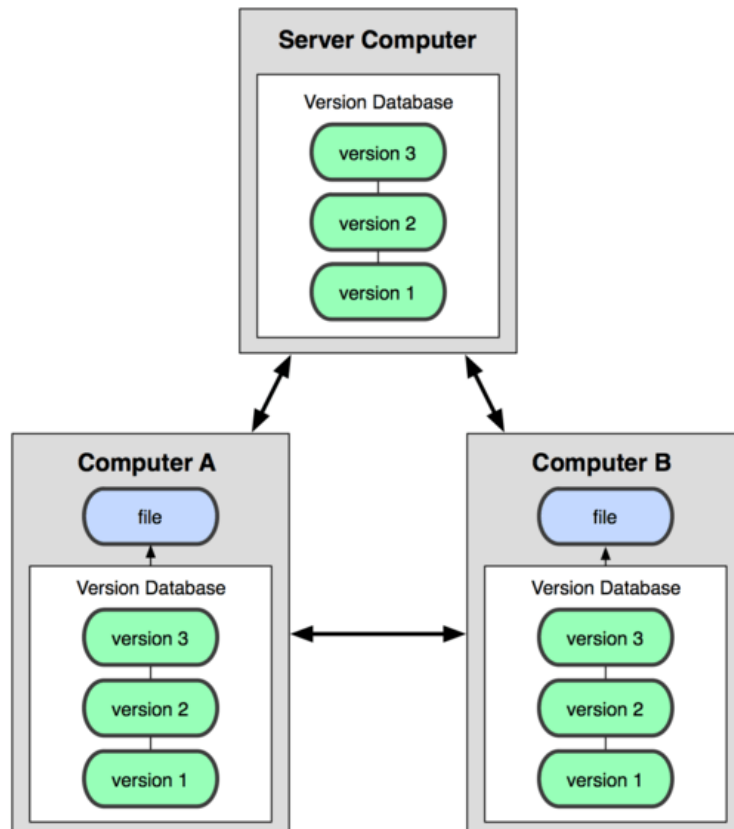


TÌM HIỂU VỀ GIT

1. Git là gì?

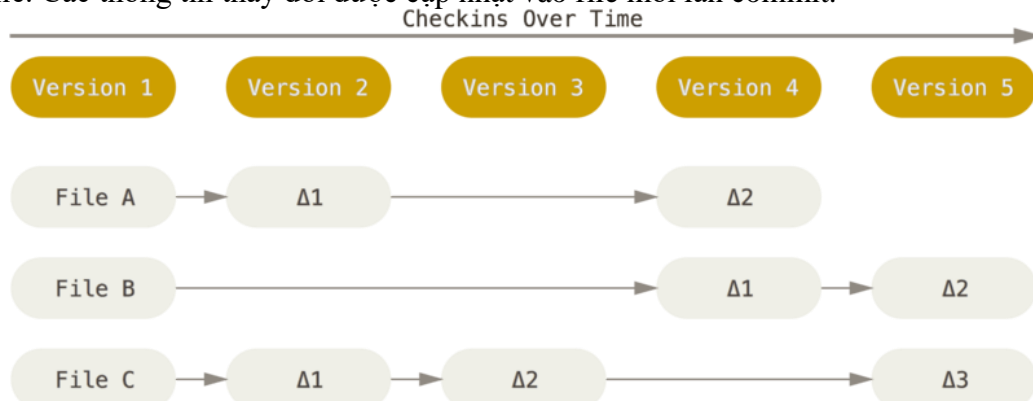
Git là một hệ thống quản lý phân tán (Distributed Version Control System – DVCS), là một trong những hệ thống quản lý phân tán phổ biến nhất hiện nay.

DVCS là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được clone từ repository, mỗi thay đổi vào mã nguồn sẽ có thể commit rồi đưa lên máy chủ đặt vào repository. Một máy tính khác cũng có thể clone lại mã nguồn từ repository hoặc clone lại một tập các thay đổi mới nhất trên máy tính kia. Đối với Git, thư mục làm việc trên máy tính gọi là Working Tree.

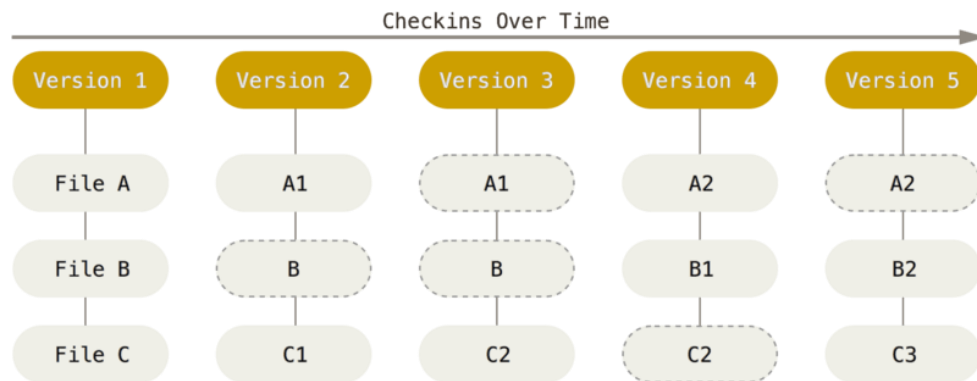


Ví dụ về working tree của git

Điểm khác biệt của Git và các DVCS khác là cách Git nghĩ về dữ liệu. Về mặt ý tưởng, hầu hết các hệ thống khác đều lưu trữ các thông tin dưới dạng một danh sách các thay đổi dựa trên file. Các thông tin thay đổi được cập nhật vào file mỗi lần commit.



Git lưu trữ thông tin dưới dạng snapshot sau mỗi lần commit hoặc lưu lại trạng thái của project. Mỗi lần lưu lại snapshot, Git sẽ lưu lại một con trỏ đến snapshot đó. Nếu file nào không có sự thay đổi, Git không lưu lại file mà chỉ liên kết đến file cũ.



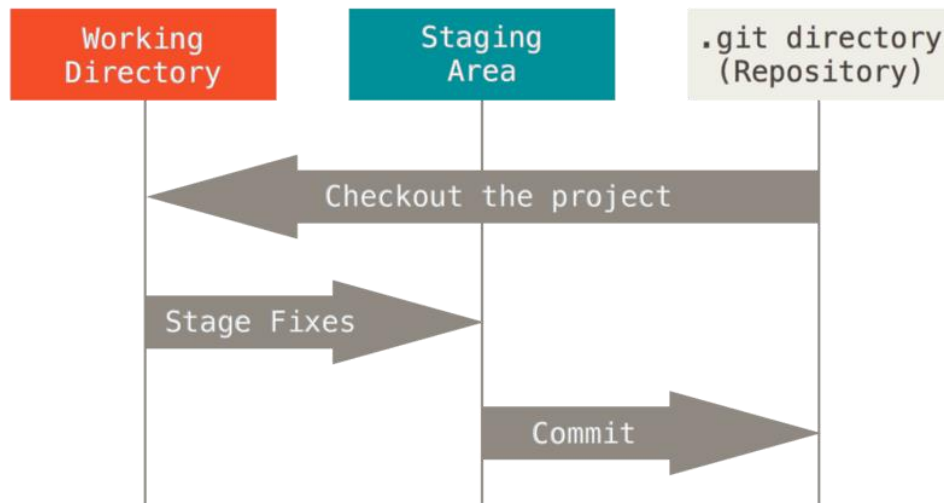
Git lưu trữ các thông tin một cách cục bộ, điều đó khiến cho Git có tốc độ rất cao trong nhiều tác vụ. Ví dụ như khi lục lại lịch sử project, Git chỉ đọc thẳng dữ liệu từ database cục bộ, nghĩa là lịch sử được đưa ra ngay lập tức, không hề thời gian. Ngoài ra, nếu không có mạng, Git commit lên database cục bộ và tự động commit lên server khi có lại internet.

Mọi tác vụ thực hiện trong Git đều được kiểm tra và mọi corruption của file đều được Git phát hiện dựa vào cơ chế SHA-1 hash. Git cũng chỉ thêm vào dữ liệu, không xoá đi, cho nên không thể xảy ra sự biến mất dữ liệu như khi lưu trên các dịch vụ lưu trữ khác.

2. Các trạng thái của file

Có 3 trạng thái cơ bản của file trong Git: modified, staged, committed:

- modified: file được thay đổi nhưng chưa commit
- staged: file được modified được đánh dấu là bản hiện tại và sẽ đi vào lần snapshot commit kế tiếp
- committed: dữ liệu đã được lưu vào database cục bộ



Có 3 vùng của một project git như trên.

Working tree là một phiên bản riêng biệt của project và được đưa từ repository về máy để chỉnh sửa.

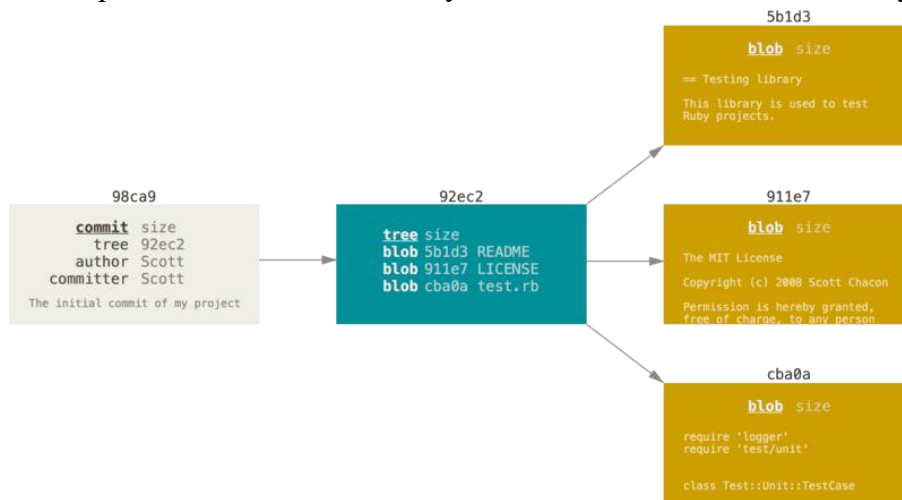
Staging area là một file lưu thông tin về các gì sẽ được đưa vào lần commit kế tiếp.

Git repository là nơi Git lưu lại toàn bộ dữ liệu về project, là nơi quan trọng nhất và là cái được cop về khi clone một repository từ một máy khác.

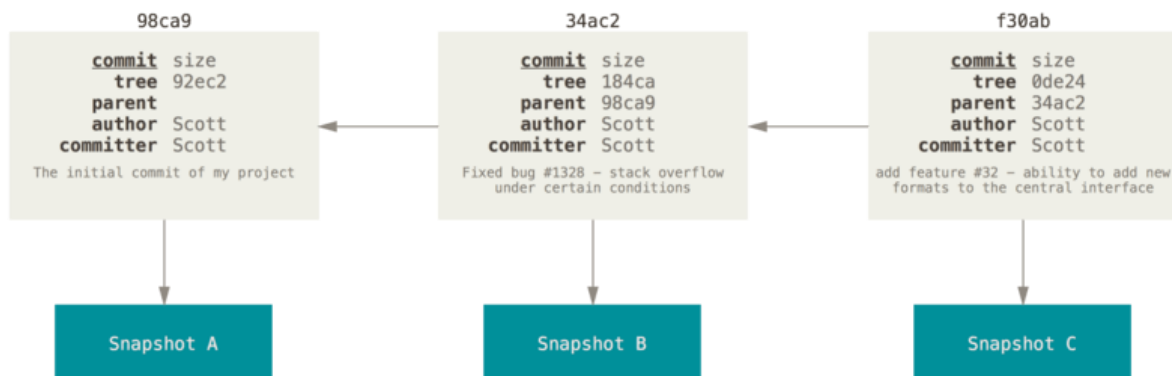
3. Rẽ nhánh

Một ưu điểm khác của Git cũng như các DVCS khác là hỗ trợ khả năng rẽ nhánh.

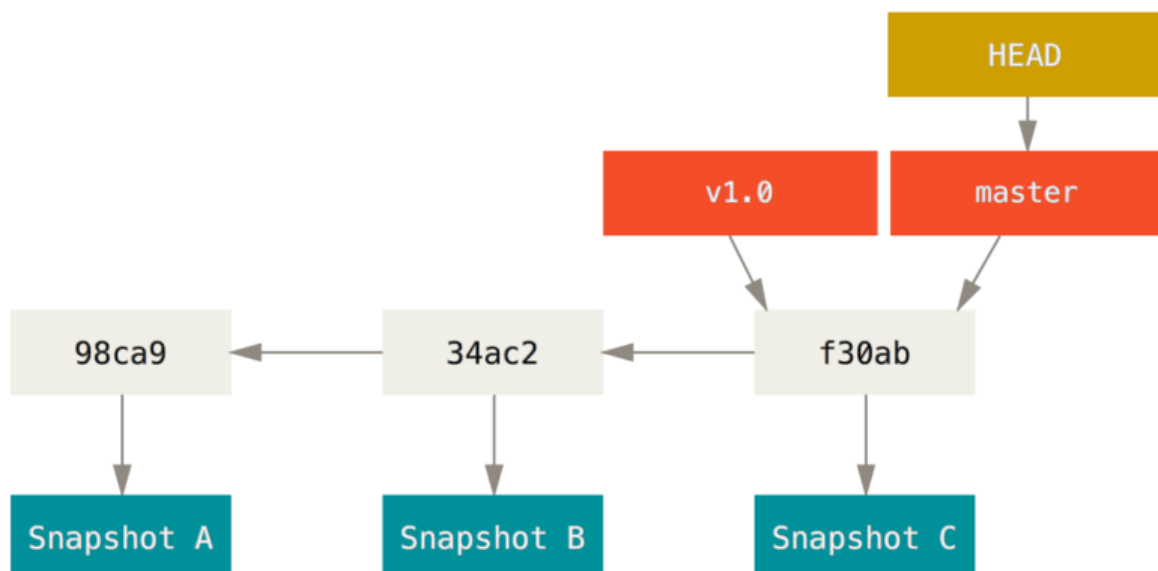
Khi commit, Git sẽ tạo ra 1 commit object lưu lại metadata và con trỏ đến project tree để có thể snapshot khi cần. Nếu directory có 3 file thì commit sẽ tạo ra 5 objects.



Nếu có thay đổi và commit lần nữa, commit đó sẽ lưu một con trỏ đến commit ngay trước đó.



Nhánh là một con trỏ dung lượng thấp trỏ đến một trong số các commit, nhánh mặc định gọi là master. Mỗi lần commit, nhánh master sẽ tự động trỏ đến commit mới nhất một cách tự động.



Khi tạo ra một nhánh mới, nó sẽ trở vào commit hiện tại và không chuyển con trỏ HEAD sang branch mới. Khi tiếp tục tạo thêm commit, chỉ con trỏ được trỏ tới bởi HEAD mới trở đến commit mới nhất. Có thể chuyển đổi giữa các nhánh một cách đơn giản.

4. GitHub

Git cần một dịch vụ để có repository và GitHub chính là một dịch vụ cung cấp repository, mọi người có thể tạo tài khoản và sử dụng dịch vụ để tạo ra các repository cho riêng mình. Đường dẫn của người viết:

https://github.com/sonLe-Thanh/Operating_system_192

5. Tại sao cần dùng Git

Git cho phép nhiều người dùng có thể commit lên cùng một repository, điều này khiến việc cùng thực hiện một project tiện hơn so với việc cùng gửi lên một dịch vụ lưu trữ đám mây. Ngoài ra, nếu ai đó bỗng dưng thêm một tính năng và xuất hiện lỗi, có thể quay lại các lần trước đó, tránh các sự việc đáng tiếc như lỗi không sửa được hay mất code.

THAM KHẢO

[1] <https://techmaster.vn/posts/35408/huong-dan-day-code-len-github>

[2] <https://git-scm.com/book/en/v2>