

GIT & GITHUB

I. Git

1. Git là gì?

Git là một hệ thống kiểm soát phân tán mã nguồn phân tán (Open Source Distributed Version Control System/ hay DVCS). Các VCS là những hệ thống ghi lại những thay đổi của file xuyên suốt dự án và mỗi sự thay đổi thì sẽ được lưu trữ thành một phiên bản. VCS cũng là hệ thống cho phép các lập trình viên có thể lưu trữ nhiều phiên bản của các mã nguồn được nhân bản (clone) từ kho chứa các mã nguồn (repository).

2. Ưu điểm của Git

- **Quản lý code và lịch sử thay đổi:** Hãy tưởng tượng nếu bạn phát triển một tính năng mới và tạo ra một đồng bug. Lúc này bạn hối hận và muốn quay đầu. Nếu không có hệ thống này, bạn sẽ phải quay lại từ đầu để dò bug và chỉnh lại cho đúng. Git sẽ giúp bạn hoàn thành những công đoạn đó chỉ với một vài dòng lệnh.

- **Tiết kiệm thời gian:** Ngoài tính năng giúp lập trình viên chỉnh sửa thay đổi dễ dàng, hệ thống này còn đóng vai trò quan trọng trong phát triển tính năng mới. Khi phát triển tính năng mới, bạn có thể copy mã nguồn gốc để làm backup lỡ tính năng không được như mong muốn. Tuy nhiên, nếu bạn muốn phát triển thêm 2-3 tính năng nữa thì việc này chắc chắn sẽ rất mất thời gian. Do đó lúc này bạn sẽ cần đến Git để tối giản hoá quá trình này

- **Giúp làm việc nhóm hiệu quả hơn:** Một dự án sẽ bao gồm nhiều người đến từ nhiều bộ phận khác nhau. Do đó, việc dẫm lên chân nhau khi thực hiện dự án là hoàn toàn có thể xảy ra. Hệ thống này sẽ đóng vai trò quản lý hệ thống và giúp giảm thiểu mâu thuẫn giữa các bộ phận

3. Các thành phần của Git

Git gồm 2 thành phần chính: Repository và Branch

- **Repository:** là kho lưu trữ chứa code của các project được Git tạo ra
- **Branch:** là các repository tách ra từ repository của project chính.

4. Các khái niệm cần biết

- **Commit:** Một commit đại diện cho một thời điểm cụ thể trong lịch sử dự án của bạn. Sử dụng lệnh commit kết hợp với lệnh git add để cho git biết những thay đổi bạn muốn lưu vào local repository.

- **Checkout:** Sử dụng lệnh git checkout để chuyển giữa các branch. Chỉ cần nhập git checkout theo sau là tên của branch bạn muốn chuyển đến hoặc nhập git checkout master để trở về branch chính (master branch).

- **Fetch:** Lệnh git fetch tìm nạp các bản sao và tải xuống tất cả các tệp branch vào máy tính của bạn. Sử dụng nó để lưu các thay đổi mới nhất vào kho lưu trữ của bạn. Nó có thể tìm nạp nhiều branch cùng một lúc.

- **Fork:** Một fork là một bản sao của một kho lưu trữ (repository). Các lập trình viên thường tận dụng lợi ích của fork để thử nghiệm các thay đổi mà không ảnh hưởng đến dự án chính

- **Head:** Các commit ở đầu của một branch được gọi là head. Nó đại diện cho commit mới nhất của repository mà bạn hiện đang làm việc.

- **Index:** Bất cứ khi nào bạn thêm, xóa hoặc thay đổi một file, nó vẫn nằm trong chỉ mục cho đến khi bạn sẵn sàng commit các thay đổi. Nó như là khu vực tổ chức (staging area) cho Git. Sử dụng lệnh git status để xem nội dung của index của bạn.

- **Master:** Master là nhánh chính của tất cả các repository của bạn. Nó nên bao gồm những thay đổi và commit gần đây nhất.

- **Merge:** Lệnh git merge kết hợp với các yêu cầu kéo (pull requests) để thêm các thay đổi từ nhánh này sang nhánh khác.

- **Origin:** Origin là phiên bản mặc định của repository. Origin cũng đóng vai trò là cú pháp để truy cập nhánh chính.

- **Push:** Lệnh git push được sử dụng để cập nhật các nhánh từ xa với những thay đổi mới nhất mà bạn đã commit.

5. Một số lệnh Git cơ bản

- **git config:** Lệnh này được dùng để set tên author name và email tương ứng sẽ được sử dụng trong commit.

- **git init:** Lệnh này được dùng để khởi tạo một repository mới.

- **git clone:** Lệnh này được dùng để lấy một repository từ một URL hiện có

- **git commit:** Lệnh này được dùng để record hoặc snapshot 1 tệp trong version history.

- **git reset:** Bạn đã đưa một tệp nào đó vào staging area nhưng bây giờ bạn muốn loại bỏ nó ra khỏi đây nhưng nó vẫn giữ nguyên nội dung tệp. Khi đó bạn dùng lệnh git reset.

- **git branch:** Lệnh này liệt kê tất cả các branch cục bộ trong repository hiện tại.

- **git log:** Lệnh này dùng để liệt kê version history cho branch hiện tại

II. GitHub

1. GitHub là gì?

GitHub là một hệ thống quản lý dự án và phiên bản code, hoạt động giống như một mạng xã hội cho lập trình viên. Các lập trình viên có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để làm việc

2. Ưu điểm của GitHub

- Dễ dàng quản lý source code
- Thuận tiện cho việc làm việc nhóm

III. Làm việc với Git và GitHub

