Khương Quốc Việt :20001595

Giảng viên : Quản Thái Hà

**GIT VÀ GITHUB**

1. **Giới thiệu**

**1. Source control/version control**

Source control là một công cụ hỗ trợ quản lý mã nguồn của một dự án phát triển phần mềm. Nó cho phép nhiều người làm việc trên cùng một tập tin mã nguồn cùng một lúc mà không cần phải lo lắng về việc ghi đè lên nhau các thay đổi. Source control còn giúp bạn theo dõi lịch sử các thay đổi trong mã nguồn cũng như khôi phục các phiên bản cũ của mã nguồn nếu cần thiết. Có rất nhiều hệ thống quản lý mã nguồn khác nhau như Git, Mercurial, Subversion và CVS

**2. Git**

Git là một hệ thống quản lý mã nguồn phân tán (Distributed Version Control System - DVCS) được sử dụng rộng rãi trong lĩnh vực phát triển phần mềm. Nó cho phép bạn lưu trữ lịch sử các phiên bản của mã nguồn cũng như theo dõi các thay đổi trong mã nguồn của dự án của bạn. Git cũng cho phép nhiều người làm việc trên cùng một dự án cùng một lúc mà không cần phải lo lắng về việc ghi đè lên nhau các thay đổi. Nó được viết bởi Linus Torvalds, người sáng lập ra hệ điều hành Linux.

**3. Github**

GitHub là một dịch vụ lưu trữ mã nguồn công cộng dành cho các dự án sử dụng hệ thống quản lý mã nguồn Git. Nó cho phép người dùng lưu trữ và theo dõi các phiên bản của mã nguồn cũng như nhận được hỗ trợ từ cộng đồng các lập trình viên khác. GitHub còn cung cấp nhiều tính năng khác như bug tracking, feature requests và wikis cho các dự án được lưu trữ trên nền tảng của họ. Nó được sử dụng rộng rãi trong lĩnh vực phát triển phần mềm và được coi là một trong những nền tảng lưu trữ mã nguồn công cộng lớn nhất hiện nay.

**II. Các thuật ngữ quan trọng và một số lệnh cơ bản**

**1. Các thuật ngữ quan trọng**

- Repository: chứa tất cả các tệp dự án

- Branch: hay nhánh, là đại diện cho các phiên bản cụ thể của 1 kho lưu trữ tách ra từ project chính. Branch cho phép ta theo dõi các thay đổi thực hiện với kho lưu trữ và có thể hoàn trả về phiên bản cũ hơn.

- Commit: đại diện cho 1 thời điểm cụ thể trong lịch sử dự án.

- Checkout: để chuyển giữa các branch hoặc để trở về branch chính

- Fetch: là tìm nạp các bản sao và tải xuống tất cả các tệp branch vào máy tính cá nhân. Sử dụng để lưu các thay đổi mới nhất vào kho lưu trữ cá nhân.

- Fork: là 1 bản sao của 1 kho lưu trữ repository

- Head: là các commit ở đầu của 1 branch. Đại diện cho commit mới nhất mà ta từng làm việc

- Master: là nhánh chính của tất cả repository

- Merge: là thêm các thay đổi từ nhành này sang nhánh khác

- Origin: là phiên bản mặc định của repository

- Pull: thêm các thay đổi vào nhánh chính

- Push: cập nhật các nhánh từ xa với những thay đổi mới nhất mà ta đã commit

- Rebase: phân tích, di chuyển một commit nào đó

- Remote: là một bản sao của nhánh chính

- Upstream: là thuật ngữ để chỉ nơi có thể push các thay đổi trong nhóm chính.

**2. Một số lệnh cơ bản**

**Thiết lập chứng thực cá nhân**

$ git config --global user.name "User Name"

$ git config --global user.email "username@gmail.com"

Lưu ý: –global được sử dụng để áp dụng cho tất cả các projects. Nếu bạn ko sử dụng –global thì settings sẽ chỉ dùng cho riêng project đó.

**Tạo một kho chứa Git**

$ git init

Nếu như bạn muốn theo dõi một dự án cũ trong Git, bạn cần ở trong thư mục của dự án đó. Lệnh này sẽ tạo một thư mục mới có tên .git, thư mục này chứa tất cả các tập tin cần thiết cho kho chứa.

**Sao chép một kho chứa đã tồn tại**

$ git clone https://github.com/user/repository.git

Câu lệnh trên sẽ tạo một thư mục mới có tên giống trên của Repo.

**Nhánh trong git**

Khi sử dụng Git, bạn có thể tạo ra nhiều nhánh (branch) khác nhau. Câu lệnh Git này dùng để kiểm tra branch hiện tại:

$ git branch

Để tạo mới một branch:

$ git branch <name\_branch>

Để chuyển và tạo mới:

$ git branch -b <name\_branch>

**Chuyển nhánh**

Trước khi muốn thay đổi source code, điều đầu tiên mà bạn cần phải làm là checkout một nhánh. Để checkout một nhánh, bạn dùng câu lệnh Git sau:

$ git checkout <name\_branch>

**Cập nhật thay đổi**

Sau khi bạn thay đổi source code: thêm mới, sửa, xoá files,… Bạn cần phải cập nhật lên Staging Area. Để cập nhật hết các files:

$ git add .

Sau lệnh add, bạn cần sử dụng câu lệnh Commit để đây thông tin thay đổi lên Local Respository:

$ git commit -m "Message"

**Cập nhật lên server**

Sau câu lệnh Commit, thông tin mới chỉ được cập nhật lên Local Repository. Nếu muốn cập nhật lên server thì bạn phải sử dụng câu lệnh push:

$ git push origin <name\_branch>

Ngoài ra, nếu chưa tồn tại remote trên server thì bạn cần phải add mới một remote trước rồi mới push:

$ git remote add origin <remote\_url>

$ git push origin <name\_branch>

**Gộp nhánh**

Sau một thời gian cập nhật các file và push lên git trên branch mới, bây giờ mình cần ghép (merge) code lại vào nhánh gốc (master). Trước tiên, cần phải checkout ra khỏi branch hiện tại cần gộp để vào branch master, sau đó thì dùng lệnh merge để ghép branch mới vào master:

$ git checkout master

$ git merge <new\_branch>

**Xem lại lịch sử commit**

$ git log

Lệnh git log sẽ cho bạn biết về người commit, ngày giờ, message của những lần commit đó.

**Xem thay đổi trước khi push**

$ git diff

Lệnh này giúp bạn biết những gì đã được thay đổi giữa nhánh hiện tại và nhánh trước nó.

**Gộp commit**

$ git rebase -i HEAD~

Sau dấu ~ là số commit bạn muốn gộp. Sau khi gõ lệnh này một cửa sổ trình soạn thảo hiện ra. Thay đổi ký tự pick của dòng các dòng sau dòng đầu thành s rồi lưu lại/kết thúc. Khi đó, trình soạn thảo để chỉnh sửa giải thích commit thiết lập cho commit sau khi đã tổng hợp sẽ được hiển thị, nên hãy chỉnh sửa lưu lại/kết thúc.

**Pull từ remote repository**

$ git pull origin master

Lệnh trên sẽ gộp những thay đổi mới kéo về từ máy chủ từ xa với nhánh hiện tại trên máy local.