

## TÌM HIỂU VỀ GIT

### Git

Git là hệ thống quản lý phiên bản phân tán (DVCS), giúp bạn quản lý phiên bản mã nguồn.

Trên Git, có thể lưu trạng thái của file dưới dạng lịch sử cập nhật. Vì thế, có thể đưa file đã chỉnh sửa một lần về trạng thái cũ hay có thể hiển thị sự khác biệt ở nơi chỉnh sửa.

Thêm nữa, khi định ghi đè (overwrite) lên file mới nhất đã chỉnh sửa của người khác bằng file đã chỉnh sửa dựa trên file cũ, thì khi đăng (upload) lên server sẽ hiện ra cảnh cáo. Vì thế, sẽ không xảy ra thất bại về việc đã ghi đè lên nội dung chỉnh sửa của người khác mà không hề hay biết.

Khi quản lý file bằng Git, lịch sử cập nhật sẽ được lưu trong Git. Vì không cần copy trước file dùng để sao lưu (Backup) sẵn có nên rất thuận tiện.

### Repository

Repository là nơi sẽ ghi lại trạng thái của thư mục và file. Trạng thái được lưu lại đang được chứa như là lịch sử thay đổi của nội dung. Bằng việc đặt thư mục muốn quản lý lịch sử thay đổi dưới sự quản lý của repository, có thể ghi chép lại lịch sử thay đổi của thư mục và file trong thư mục đó.

- Remote repository: Là repository để chia sẻ giữa nhiều người và bố trí trên server chuyên dụng.
- Local repository: Là repository bố trí trên máy của bản thân mình, dành cho một người dùng sử dụng.

Do repository phân thành 2 loại là local và remote nên với những công việc bình thường thì có thể sử dụng local repository và thực hiện trên toàn bộ máy sẵn có. Khi muốn công khai nội dung công việc mà bản thân đã làm trên local repository, thì sẽ upload lên remote repository rồi công khai. Thêm nữa, thông qua remote repository cũng có thể lấy về nội dung công việc của người khác.

#### *-Tạo repository*

Cách thứ nhất là tạo repository hoàn toàn mới, cách thứ hai là sao chép remote repository rồi tạo.

### Snapshot

Sau mỗi lần lưu trạng thái sẽ tạo ra 1 “ảnh chụp” lưu lại nội dung file, thư mục tại thời điểm đó rồi tạo tham chiếu tới snapshot đó. Sau đó khi cần bạn hoàn toàn có thể khôi phục và sử dụng lại một snapshot, hay còn gọi là phiên bản nào đó. Đó cũng là lợi thế của

git khi nó không lưu dữ liệu mà lưu dưới dạng snapshot, giúp tiết kiệm không gian lưu trữ.

## **Branch**

Với nhánh bạn có thể tách riêng các tính năng của dự án, thử nghiệm các tính năng mới,... Khi tạo một repository hoặc clone repository, bạn sẽ có 1 nhánh chính tên là master, là branch chứa toàn bộ mã nguồn chính. Từ nhánh này bạn có thể rẽ thêm nhiều nhánh khác tùy theo nhu cầu. Tất cả các nhánh đều được lưu lại lịch sử commit.

## **Commit**

Để ghi lại việc thêm/ thay đổi file hay thư mục vào repository thì sẽ thực hiện thao tác gọi là Commit.

Khi thực hiện commit, trong repository sẽ tạo ra commit (hoặc revision) đã ghi lại sự khác biệt từ trạng thái đã commit lần trước đến trạng thái hiện tại.

Khi thực hiện commit có yêu cầu nhập giải thích commit (commit message). Vì commit message là bắt buộc nên nếu để trống mà thực hiện thì commit sẽ thất bại.

## **Push**

Để chia sẻ lịch sử thay đổi của local repository mà bản thân đang có bằng remote repository, cần phải upload lịch sử thay đổi trong local repository. Vì thế, sẽ thực hiện thao tác gọi là Push trên Git.

## **Clone**

Để sao chép remote repository, sẽ thực hiện thao tác gọi là "clone".

## **Pull**

Để cập nhật local repository từ remote repository thì thực hiện thao tác gọi là Pull.

## **Fetch**

Truy cập vào repository trên server và kéo toàn bộ dữ liệu chưa có từ repository trên server về.

## **Các lệnh git cơ bản:**

-Thiết lập chứng thực cá nhân

```
$ git config --global user.name "User Name"  
$ git config --global user.email "username@gmail.com"
```

-Tạo một kho chứa Git

```
$ git init
```

-Sao chép một kho chứa đã tồn tại

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

-Nhánh trong git

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

```
$ git checkout <name_branch>
```

-Cập nhật thay đổi

```
$ git add .
```

```
$ git commit -m "Message"
```

-Cập nhật lên server

```
$ 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

```
$ git checkout master
```

```
$ git merge <new_branch>
```

-Xem lại lịch sử commit

```
$ git log
```

-Xem thay đổi trước khi push

```
$ git diff
```

-Gộp commit

```
$ git rebase -i HEAD~
```

-Pull từ remote repository

```
$ git pull origin master
```

Link repo: [https://github.com/rongqui/Bonus\\_lab\\_2](https://github.com/rongqui/Bonus_lab_2)