

GIT

1. Download và cài đặt GitHub:

- Vào đường dẫn <https://git-scm.com/download> để download Git và tiến hành cài đặt
- Thiết lập tên và email

```
git config --global user.name "nvdinh185"  
git config --global user.email "nvdinh0905671805@gmail.com"
```

- kiểm tra thông tin chứng thực trên user
`cat ~/.gitconfig`

2. Tạo một repository ở máy tính cá nhân

- Tạo local repository
`git init`
- đưa các tập tin về trạng thái Tracked
`git add .`
- xem danh sách các tập tin đã được tracked.
`git status`

- Commit

```
git commit -m "commit to repository"
```

- Lệnh viết tắt add và commit

```
git commit -a -m "commit to repository"
```

3. Tạo repository trên Github

- đăng nhập vào [Github](https://github.com), sau đó tạo repository

4. Push project từ máy của mình lên GitHub

```
git remote add GitProject https://github.com/nvdinh185/appdemonvndinh.git  
git push --set-upstream GitProject master
```

5. clone repository từ GitHub về máy tính của mình

```
git clone https://github.com/nvdinh185/GitProject.git
```

6. remove remote git

```
git remote rm gitName
```

GITHUB CƠ BẢN

I. Tạo kho chứa ở máy tính cá nhân:

1. Tạo thư mục để sử dụng lưu trữ với git
2. Tạo thư mục .git để làm việc với git: `npm init`
3. Sau khi có thay đổi trong thư mục, dùng lệnh `git status` để kiểm tra trạng thái kho git
4. Đưa các file trong thư mục về trạng thái Tracked để git theo dõi: `git add .`
5. Lưu trữ các file ở trạng thái Tracked vào kho chứa local để git theo dõi: `git commit -m "comment gì do..."`

II. Tạo một kho git trên server

1. Đăng nhập vào trang chủ <https://github.com/> đăng ký một tài khoản
2. Tạo một kho chứa trên server

III. Push kho chứa từ máy cá nhân lên kho chứa trên server

1. Từ kho chứa cá nhân, tạo một kho chứa từ xa trên server: `git remote add <tên_remote> <đường dẫn kho git trên server>`
 2. Push dự án lên kho chứa trên server: `git push <tên_remote>`
- * Lưu ý:
- + Không được đặt tên_remote **origin** là
 - + Có thể tạo nhiều remote để lưu trữ 1 dự án trên git server

IV. Nhân bản dự án từ server về máy tính cá nhân

git clone <đường dẫn kho git trên server>

- Giả sử có 2 người clone một dự án ở git server về

I - HAI NGƯỜI LÀM VIỆC Ở 2 THƯ MỤC ĐỘC LẬP

- Người thứ nhất:

+ thêm thư mục của mình vào

+ thực hiện commit và push lên (git push)

- Người thứ hai:

+ thêm thư mục của mình vào

+ pull toàn bộ những gì có trên server về trước

+ thực hiện commit và push lên (git push)

* Lúc này, 2 người làm việc độc lập với nhau, mỗi người làm việc với thư mục của mình (thêm, sửa, xóa), trước khi push lên thì phải pull những gì có trên server về rồi mới push lên lại được

II - HAI NGƯỜI CÙNG LÀM VIỆC CHUNG VỚI MỘT FILE

- Người thứ nhất:

+ chỉnh sửa theo ý 1

+ thực hiện commit và push lên (git push)

- Người thứ 2:

+ chỉnh sửa theo ý 2

+ pull toàn bộ những gì có trên server về, có các lựa chọn sau:

* Accept Current Change: Chấp nhận những thay đổi 2

* Accept Incoming Change: Chấp nhận dữ liệu trên server về

* Accept Both Change: Trộn cả 2 lại với nhau (thay đổi 2 nằm trước)

+ Sau khi lựa chọn 1 trong 3 thì commit và push lên lại

- Lúc này dữ liệu trên server là những gì người thứ 2 push lên

- Người thứ nhất không thể push lên được nữa, nếu muốn push lên, phải pull về, và chấp nhận dữ liệu trên server (cẩn thận)

• thêm

- o khi clone dự án từ server về, nếu thêm vào file/thư mục của mình thì cứ thế mà làm việc
- o nếu có sửa chữa trong file chung thì khi pull về từ server sẽ bị lỗi, khi đó phải commit trước rồi mới pull về được, nếu xảy ra xung đột code thì phải chọn các tùy chọn thích hợp

GIT nâng cao

- nếu có file bị thay đổi thì trong dự án có dấu chấm phía sau
- `git add .` để add toàn bộ nhưng thay đổi
- `git commit -m "message"` để commit nhưng thay đổi
- `git remote -v` để xem những remote trên github
- `git push -u origin master` để push những thay đổi lên github

- `git checkout` để kiểm tra những file bị thay đổi
- `git checkout <ten_file>` để lấy lại file ban đầu (trước khi bị thay đổi)
- `git checkout -- .` để lấy lại tất cả những file bị thay đổi

Giả sử có một kho chứa trên server

- Nếu có hai người cùng clone về (`git clone...`), và thực hiện thay đổi, người nào thực hiện push lên trước thì được (`git push`), các người sau không push lên được
- Nếu người sau muốn push lên thì phải pull lại dữ liệu trên server về (`git pull`), để đồng bộ dữ liệu giữa máy local và server, sau đó mới push lên được (`git push`)
- Tiếp theo, phần nào của người nào sửa phần ấy, sau đó ai thực hiện push lên trước thì được, người sau không push lên được thì phải pull về sau đó mới push lên (`git pull` và `git push`)
- Khi đó thì phần của người nào sửa người ấy và không liên quan đến nhau

- Ban đầu mới tạo repository trên server, trên server sẽ có remote tên mặc định là origin, Khi clone về thì trên local cũng có remote này

- kiểm tra các remote có trên local hiện tại: `git remote -v`

- Xóa một remote trên local hiện tại: `git remote rm tên_remote`

=> chưa hiểu?

+ tạo một remote trên local (để làm gì?), `git remote add tên_remote`

`https://github.com/nvdinh185/tên_project.git`

+ `git push -u git1 master` (hoạt động như thế nào?)

- Để push lên kho lưu trữ cá nhân trên server:

+ Tạo một repository trên github

+ Tạo remote ở local: `git remote add tên_remote https://github.com/nvdinh185/tên_project_mới.git`

+ Thực hiện push lên kho cá nhân: `git push tên_remote`

- Nếu có remote cả kho chứa cá nhân và kho chứa chung,

thực hiện push lên kho nào sau cùng thì khi pull về cũng pull về từ kho đó

- Nếu muốn pull từ một kho xác định về, dùng lệnh: `git pull tên_remote master`

khi đó sẽ có những thành phần xung đột nhau

+ chọn Accept Current Change để giữ lại dữ liệu pull về đầu tiên

+ chọn Accept Incoming Change để chấp nhận dữ liệu pull về sau

+ chọn Accept Both Change để merge cả 2 lại

* Lưu ý: Nếu đã chọn rồi thì không thể pull về được nữa

=> chưa xong!!! Để push lên lại sau khi đã thay đổi, dùng lệnh `git push --set-upstream git2 master`