**GITHUB CƠ BẢN**

**Git** - một phần mềm quản lý source code

**Setting git**

1. Download git trên trang: https://git-scm.com/downloads/win

2. Cài đặt phần mềm git.exe vừa cài

3. Tạo tài khoản github

4. config username & email:

+ gg > git config email username

+ vào folder bất kỳ > phải chuột > git bash here

+ git config --list > enter

+ git config --global user.name "FIRST\_NAME LAST\_NAME"

+ git config --global user.email "emaildadkgithub@gmail.com"

+ git config --list

**Clone/Kéo dự án đã có từ github về pc**

1. Vào reponsitory có code cần clone về

2. Chọn "Code" > Chọn HTTPS/SSH > biểu tượng copy

(SSH chỉ xuất hiện khi bản thân người clone là owner)

3. Chọn nơi cần clone code về > chuột phải > git bash here

4. git clone <shift + ins> > enter

[done = thành công]

**Tạo 1 project trên github**

1. Đăng nhập vào github

2. Tạo reponsitory

**Push code lên github -- TH đã có project sẵn, tải về, code và update lên lại**

1. Tạo 1 project trên github

2. Copy link reponsitory > clone code về

2. Tạo 1 file mới bất kỳ trong reponsitory vừa clone về

4. Chuột phải > git bash here > git status

5. git add . (thêm all các file hiện có, nếu muốn add file cụ thể thì thay "." = tên file)

6. git commit -m "noi\_dung"

7. git push origin main

**Push code đã tồn tại trên PC lên github -- đã code xong hoàn toàn**

1. Tạo 1 project trên github

2. Vào folder cần đẩy lên github (trên pc)

3. Chuột phải > git bash here

4. git init (khởi tạo .. )

5. git status

6. git commit -m "noi\_dung"

7. git remote add origin https://.... (copy trên reponsitory)

8. git push origin master

**Sự khá biệt giữa HTTPS và SSH:**

HTTPS: cần xác thực thông qua username và password tài khoản github

SSH: cấu hình 1 lần và những lần sau không cần password để push hay clone code

=> rcm sử dụng SSH

Cách setting ssh: https://youtu.be/pC1s6JOwglE?si=HWfEomEcnUKoUrA4

**Cách thức hoạt động:**

Search "git stage cycle"

Local: máy tính cá nhân

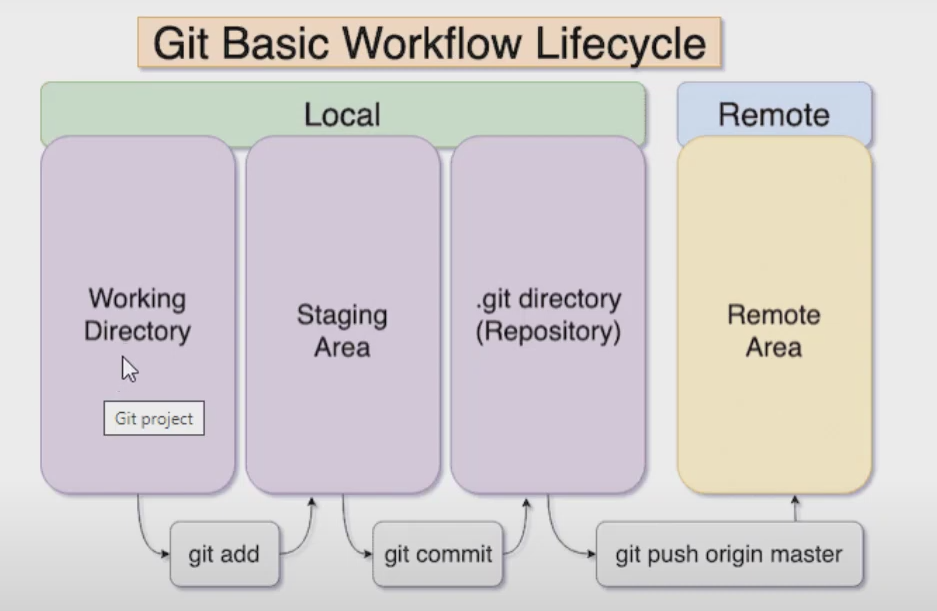
+ source code dự án = working directory

+ staging area = nơi lưu trữ trạng thái git

+ .git directory (quan trọng nhất) - lưu trữ lịch sử

Remote: nơi lưu trữ code online

Xem full tại: <https://youtu.be/YnLoU_d53js?si=KwWxlw5aLNgj1RsK>



**Mục đích của việc chia brand trên github:**

Để hoàn thiện 1 chức năng > test > sau khi ổn thì gộp hết code lại vào main

Không ảnh hưởng ở code chính

**Một số câu lệnh thường dùng:**

1. Kéo code mới nhất về: git pull original main

(git pull = git fetch + git merge <=> ghép các file trên git và merge vào code local, không merge trực tiếp do có trường hợp các thành viên cùng làm và cùng merge cùng 1 file => xảy ra conflict)

1. git checkout -b ten\_nhanh (tạo nhánh mới)
2. git branch (check xem có những branch nào)
3. git checkout main (chuyển nhánh)
4. Muốn đưa code từ nhánh A sang nhánh B: => merge code (ghép code từ nhánh này sang nhánh kia & cộng gộp thành 1)

> chọn compare & pull request

1. 2 hoặc nhiều người cùng sửa 1 file > conflict

fix:

+ git pull (check lỗi) > ngay dòng conflict

+ mở lại file bị lỗi và fix lại (xóa đi các ký tự đặc biệt và giữ lại nội dung muốn giữ) > save

+ git add .

+ git commit -m "noi\_dung"

+ git push original <ten\_nhanh>

1. Kiểm tra user đang được cấu hình trong Git

git config --global user.name

git config --global user.email

1. chuyển đổi giữa các tài khoản github

<https://vinasupport.com/tim-kiem-va-thay-doi-git-username-va-email-khi-commit/>

**Tài liệu tham khảo:**

<https://youtube.com/playlist?list=PLncHg6Kn2JT6nWS9MRjSnt6Z-9Rj0pAlo&si=C8mTLh3f6QP0nSSX>

**Tài khoản github:** <https://github.com/cotrandev>