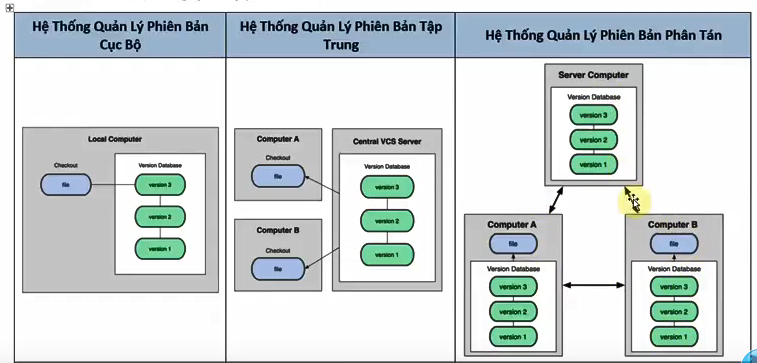
Link

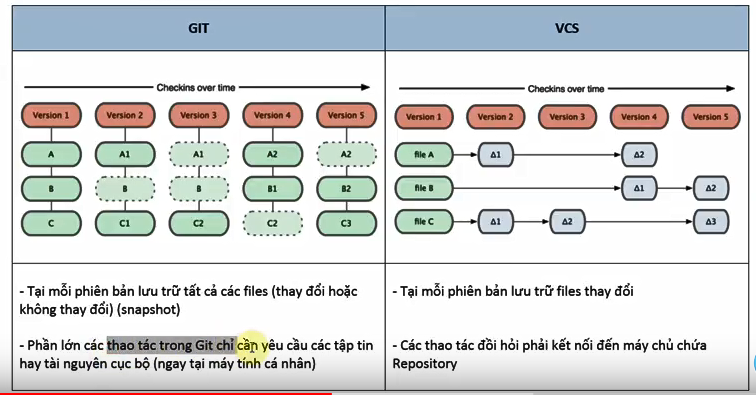
<https://www.youtube.com/watch?v=Q8hUbBtVwCo&list=PLv6GftO355Atx3v9_ZAu8hQEiAg-Nj3P3>

**Mô hình Git**



Git : Hệ thống quản lí phiên bản phân tán

Git: Tại mỗi version thì nó đều lưu tất cả các file , kể cả các file ko bị thay đổi



**Cài đặt git:**

<https://git-scm.com/>

Kiểm tra phiên bản git: git –version

**Cấu hình Git**

Cấu hình tên: git config –global user.name MonkeyDLuffy

Cấu hình email:git config –global user.email [votuanbk232@gmail.com](mailto:votuanbk232@gmail.com)

Kiểm tra tên: git config user.name

Xem thông tin cấu hình:

+git config –list

+git config X( X: thuộc tính trong list)

Xem thông số câu lệnh X nào đó( X: câu lệnh nào đó)

+git help X

+git X –help

+man git --X

**Repository**

1. Repository là gì

+Lưu trữ các trạng thái của tập tin,các hành vi của người dùng trên các tập tin này

+Thường bố trí repository=project

+Để ghi lại việc thay đổi các tập tin vào repository ta sử dụng commit

+Mỗi commit phân biệt bởi id, mỗi commit chính tạo ra 1 version

2.Phân biệt repository

+Local Repository: repository lưu trữ trên máy tính cá nhân, một người sử dụng

+Remote Repository: repository lưu trữ trên hệ thống máy chủ chuyên dụng,chia sẻ project cho nhiều người thực hiện

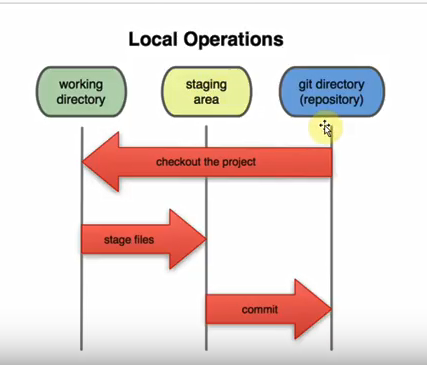
**3.Trạng thái tập tin và Local operations**

+Committed: dữ liệu đã được lưu trữ an toàn trong csdl

+Modifiled: dữ liệu đã được thay đổi nhưng chưa commit vào csdl

+Stayed:đánh dấu sẽ commit phiên bản hiện tại của một tập tin đã chỉnh sửa trong lần commit tiếp theo

🡺 3 vùng làm việc với git:



+git directory: nơi git lưu trữ siêu dữ liệu và csdl

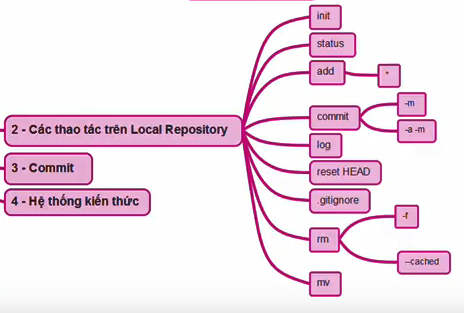
+working directory: các tập tin nằm trong dự án

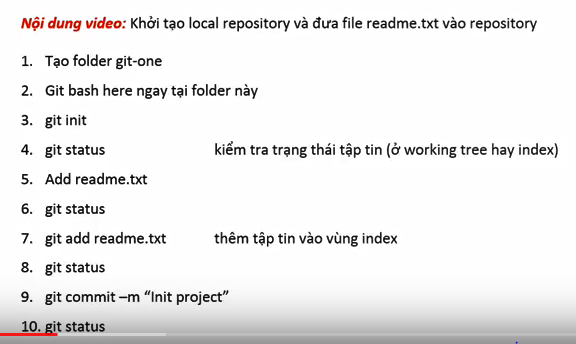
+Staging area: các tập tin sẵn sàng đưa lên git directory

4. Quy trình commit

Từ thư mục working directory sẽ đánh dấu tới trạng thái staged area và sau đó được đẩy tới git repository

**Các thao tác trên Local Repository**





**Để bỏ qua bước add vào staging khi commit**

+Đối với file đã được thêm vào vùng staging trước đó rồi( tức đã được theo dõi

+Git commit –a –m “Commit here”

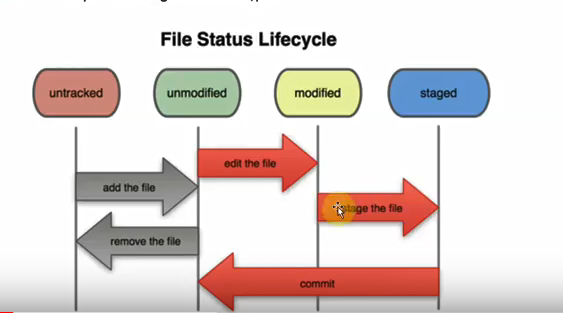
**Xem lịch sử commit**

**Git log**

**Staging Multi File**

**Git add \***

**LifeCycle:Vòng đời tập tin**



+Mỗi tập tin trong thư mục làm việc có 2 trạng thái tracked và untracked

+Tập tin tracked là các tập tin đã có mặt trong ảnh (snapshot) trước

A, Unmodifield

B,Modifield

C,Staged

+Tập tin untracked là các tập tin còn lại ko cùng 3 trạng thái kia( ko ở trong ảnh( commit trước) hay ko ở trong khu vực staging area)

**Staged và Unstaged**

Xóa file ra khỏi thư mục staging

Sau khi add file report.txt tới trạng thái staged=> xóa khỏi trạng thái staged

+git reset HEAD report.txt

+hoặc git rm –cached report.txt

**Sử dụng gitignore**

Để không theo dõi các tập tin được chỉ định( Từ chối một số file được thêm vào vùng staged)

++Ví dụ : từ chối thư mục libs(chứa file string.txt ..) và file me.txt

+Thêm file gitignore bằng câu lệnh: touch .gitignore

+Cập nhật nội dung cho file gitignore:

// File me.txt

Me.txt

# Folder libs

/libs

Ví dụ ko commit các file đuối xml: \*.xml

\*\*\*Quy tắc

+abc/ thực mục abc

+# ghi chú được bỏ qua

+!abc.txt không phải tập tin abc.txt

+không theo dõi tập tin đuôi a \*.a

+bỏ qua \*.a nhưng vẫn theo dõi tập tin lib.a !lib.a

+chỉ bỏ qua thư mục todo ở thư mục gốc, chứ ko ở thư mục consubdir/todo

/todo

+bỏ qua tất cả tập trong thư mục build build/

Thư viện ignore online: <https://www.gitignore.io/>

**Remove**

Xóa tập tin ra khỏi thư mục git

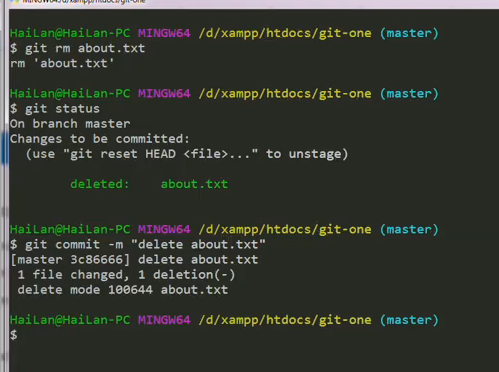
TH1: Sau khi tạo một file mới ( đang ở vùng working tree)=> ta xóa bình thường do tập tin đó chưa đc theo dõi

Xóa bằng tay hay bằng câu lệnh: rm abc.txt

TH2: Xóa tập tin trong Git

+Xóa khỏi danh sách tracked(xóa khỏi vùng staged) git rm <filename>

+Sau đó commit : git commit –m”delete filename”



TH3: Tập tin readme.txt đã được staged,edit nội dung của nó về trạng thái modifield , chưa thêm vào staged, xóa thế nào.

Git rm –f readme.txt

Để phục hồi tập tin: git reset HEAD readme.txt

Git checkout – readme.txt

TH4: tập tin account.txt đã đc staged, ko edit gì cả

Xóa: git rm account.txt

Sau đó: git commit

TH5: readme.txt đã đc staged, edit nội dung và đưa về trạng thái staged

B1: git rm –f readme.txt

B2: git commit –m”delete”

**RENAME**

Đưa file task-01.txt về dạng staged

Git mv task-01.txt task-02.txt

Sau đó git commit

**Checkout: Phục hồi**

Quay về trạng thái trước của một tập tin( đầu tiên đưa tập tin về dạng unstaged nếu cần)

+Khi thay đổi file readme.txt nhưng lại ko muốn commit nó và muốn quay về trạng thái ban đầu

+git checkout – readme.txt khi chưa thêm vào vùng staged

Khi đã thêm vào vùng staged ta thêm : git reste HEAD readme.txt

**Amend**: Commit và lịch sử commit

Vấn đề: Bạn nhận 1 task tạo 2 file A, B nhưng bạn chỉ tạo 1 file A và commit, sau đó bạn tạo file B và commit tiếp=> có 2 commit, trong khi chỉ có 1 yêu cầu commit

>>>>

+A.txt-> git add \*->git commit –m “add A”

+B.txt-> git add \*->git commit --amend ( lưu tại commit add A luôn)

( Ấn :x để thoát)

Log: xem lịch sử commit

Git log: lịch sử commit

Gitk : lịch sử commit hiển thị bằng GUI

Git log -p -3 hiển thị 3 commit gần nhất

Git log –stat hiển thị lịch sử commit với thông tin vắn tắt

Git log –pretty=oneline lịch sử commit với mỗi commit là 1 dòng

Git log –pretty=format:”%h-%an,%ar:%s”

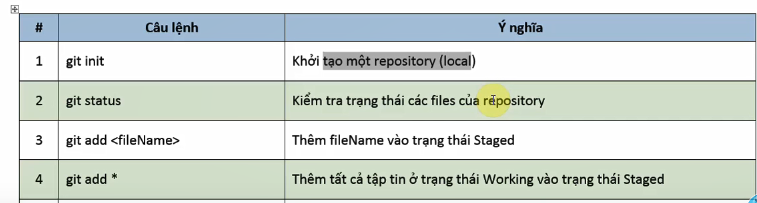
**Quay về trạng thái commit bất kì**

Git checkout id\_commit

**Quay về trạng thái đang làm việc hiện tại**

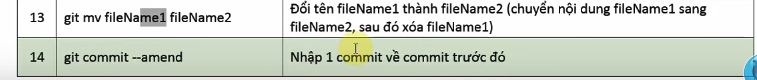
Git checkout master

**Hệ thống kiến thức**









**Đẩy 1 project tại máy tính lên 1 project trên github**

