Lab 7:Tìm hiểu các thông số liên quan đến một web hosting?

* Dung lượng lưu trữ(disk space):**là khoảng trống của ổ đĩa, dung lượng để chỉ sức chứa của gói host. Tương tự như dung lượng của USB, ổ cứng... tùy theo gói hosting mà bạn lựa chọn sẽ có dung lượng chứa khác nhau, bạn sẽ chỉ được phép tải file, upload mã nguồn trong sức chứa giới hạn của một gói hosting. Dung lượng lưu trữ trên Hosting, máy chủ có thể là vài trăm MB, vài GB thậm chí vài TB.**
* Băng thông(bandwidth):
  + **băng thông Website là một khái niệm mô tả lượng dữ liệu thông qua Website được phép truyền tải trong một thời gian nhất định (tải lên khi chỉnh sửa, thiết kế mới và đưa dữ liệu vào Web; tải xuống khi người dùng lướt Web và thực hiện các hành vi liên quan đến đọc dữ liệu). Nếu băng thông Web đã gần hết hoặc đã hết thì tốc độ truy cập Website từ người dùng sẽ rất chậm hoặc gần như không thể.**
* Cơ chế bảo mật, sao lưu và phục hồi dữ liệu:
  + **cơ chế bảo mật của server như thế nào (loại máy chủ sử dụng, công nghệ bảo mật đang dùng, cơ chế bảo mật là cứng (thiết bị) hay mềm (soft)…, bên cạnh đó cơ chế sao lưu và phục hồi dữ liệu khi xảy ra sự cố (đối với các máy chủ share Hosting vật lý truyền thống thì vấn đề này là tối quan trọng). Thông dụng và tốt nhất hiện nay là sao lưu 1 tuần/lần được áp dụng tại DIGISTAR, Mắt Bão, PA, Nhân Hòa… cần lưu ý nếu một ngày một lần sao lưu dữ liệu thì vừa tốn kém chi phí mà vừa không bảo mật (một nghịch lý trong ngành lưu trữ máy chủ Hosting)**
* Phần mềm hỗ trợ thao tác với web hosting(upload/download):
  + **thường là các phần mềm quản lý việc tải lên (upload) hoặc tải dữ liệu xuống máy tính người dùng (download), thông dụng là Direct Admin, cPanel, Kloxo nếu là máy chủ Hosting linux còn đối với máy chủ Hosting window là Plesk, Ecompak…**
* Khả năng tương thích với các: CMS, framework, ngôn ngữ lập trình(loại ngôn ngữ, phiên bản):
  + **gói dịch vụ Hosting của nhà cung cấp tương thích với phiên bản lập trình nào (ASP, ASP.NET, PHP 5.1, PHP 5.2, …), cũ hay mới (ví dụ PHP 4x hay 5x, ASP hay ASP.NET…). Và sẽ thật tuyệt vời nếu nhà cung cấp cho phép tùy biến phiên bản lập trình như DIGISTAR.**

Mã website:

lL%Af3Im^YsBJTkN1w^m

tìm hiểu về docker.

**Tìm hiểu về git:xài gitbash**

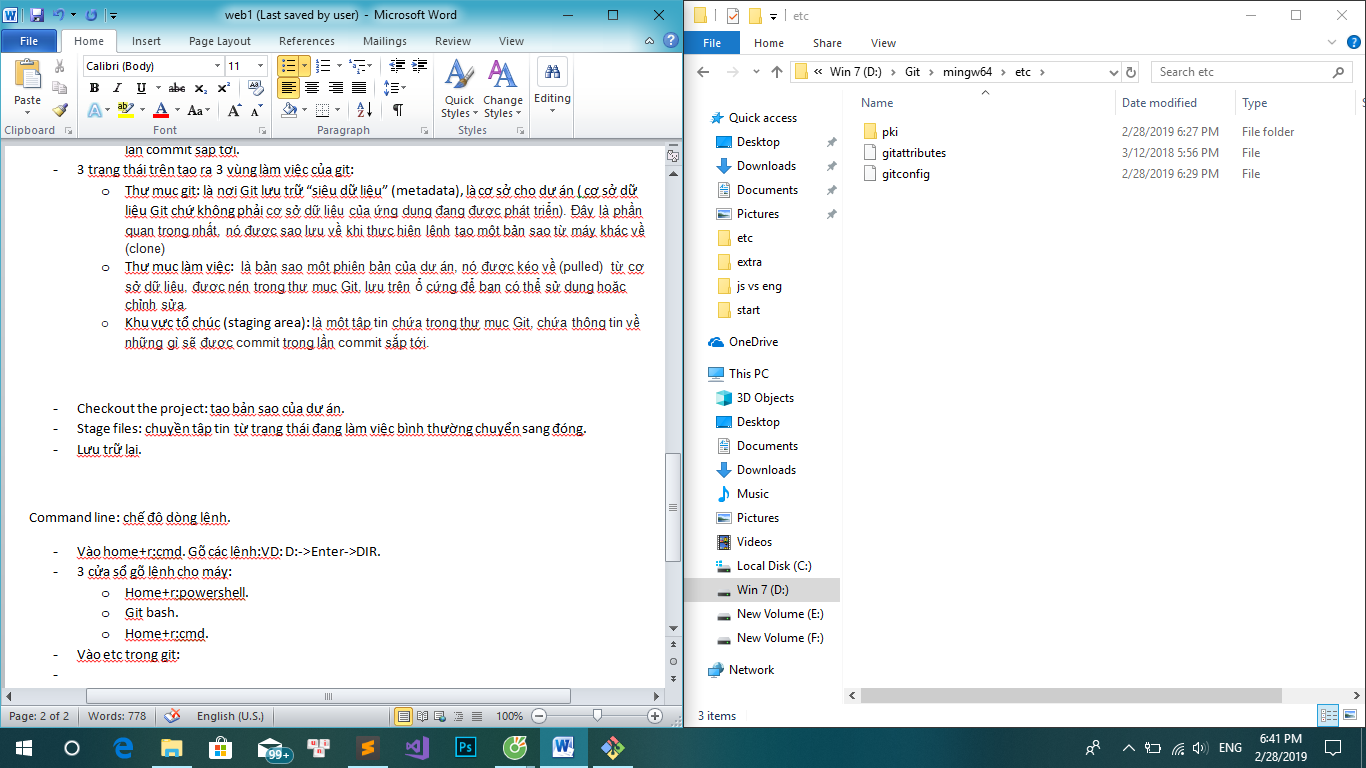
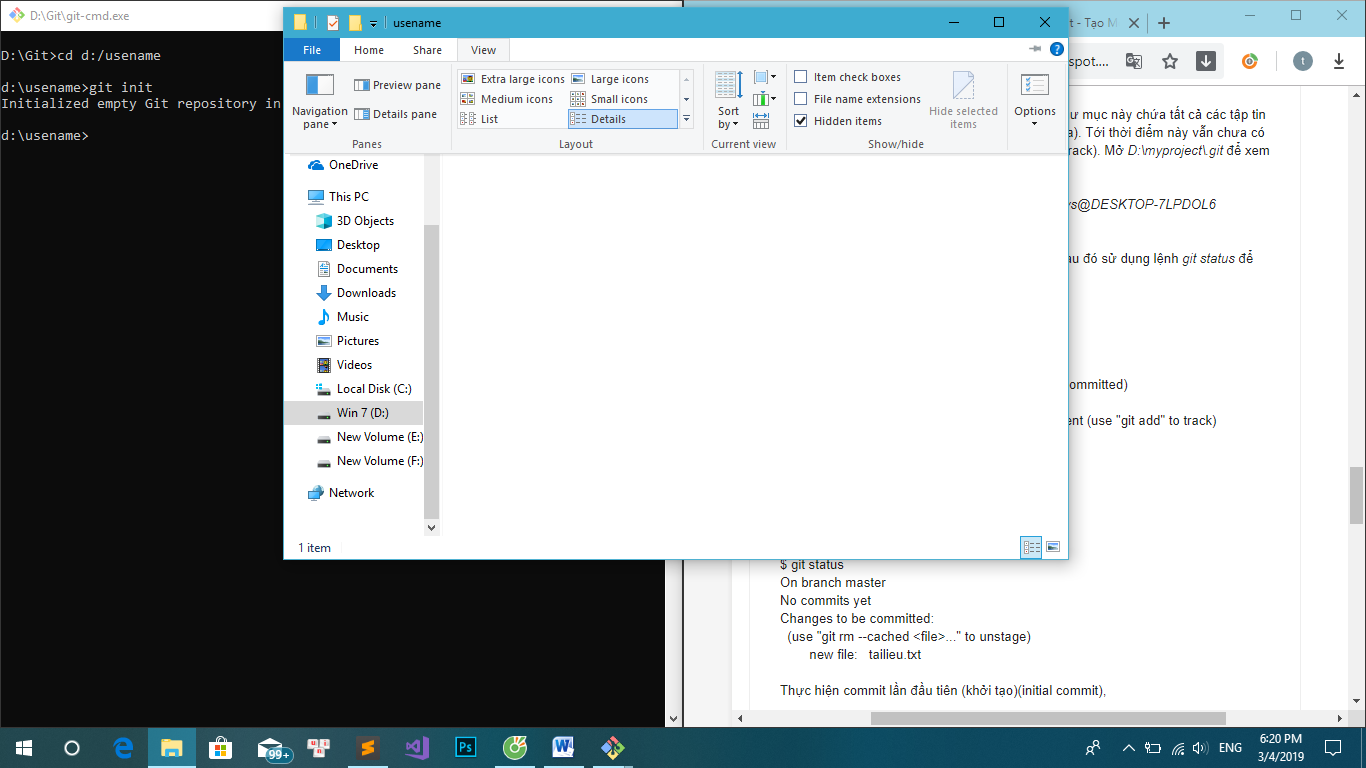
[**http://legiacong.blogspot.com/2019/02/lam-web-02-git-tong-quan.html**](http://legiacong.blogspot.com/2019/02/lam-web-02-git-tong-quan.html)

**http://legiacong.blogspot.com/2019/02/lam-web-03-git-cai-at-cau-hinh-kho-chua.html**

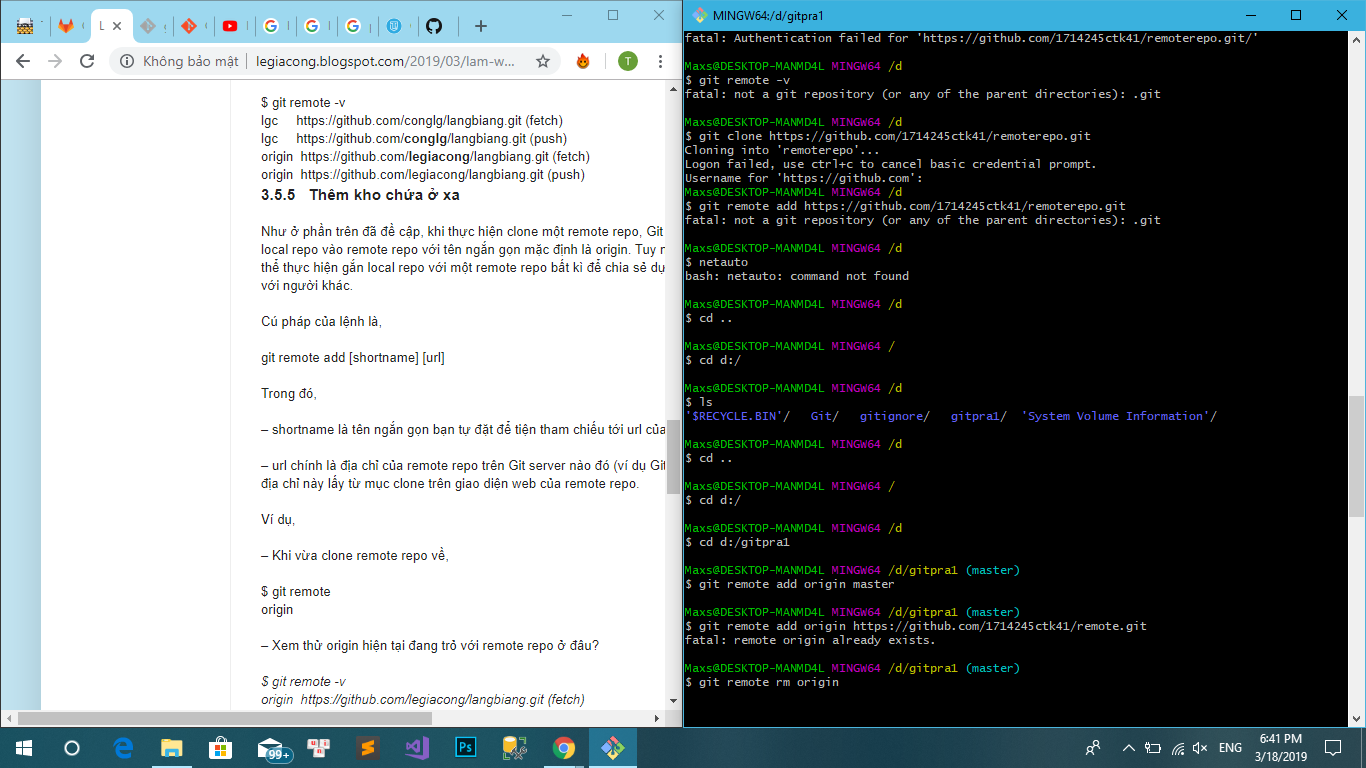
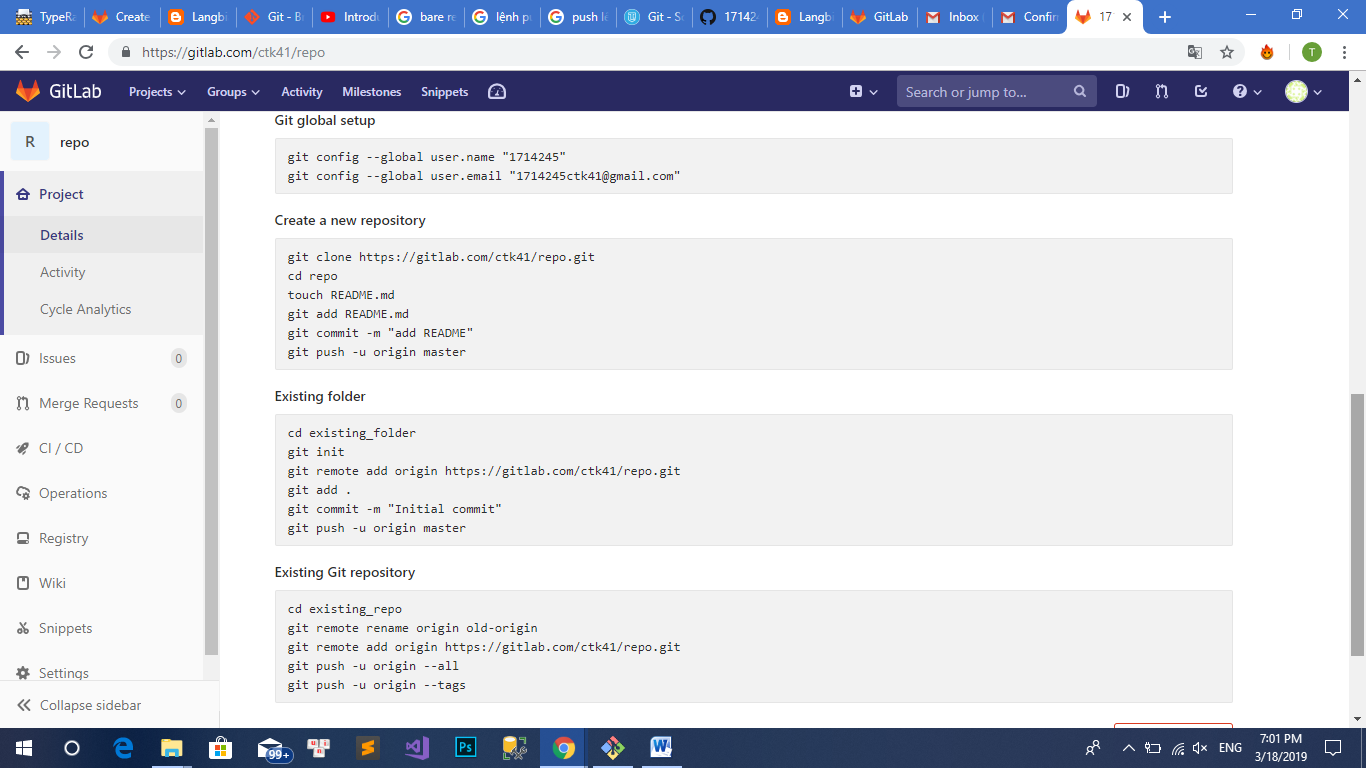
1. quay lui lại trạng thái ban đầu, cho phép bạn chuyển ngược lại tập tin bất kì về ban đầu
2. bạn có thể khôi phục lại trạng thái ban đầu.
3. chỉnh sửa , biết được ai là người chỉnh sửa.

* 3 trạng thái quan trọng của dữ liệu:
  + Committed: dữ liệu đã được lưu trữ an toàn trong cơ sở dữ liệu.
  + Modified:tập tin đã bị thay đổi nhưng chưa commit vào cơ sở dữ liệu.
  + Staged: đã đánh dấu sẽ commit phiên bản hiện tại của tập tin( đã được chỉnh sửa) trong lần commit sắp tới.
* 3 trạng thái trên tạo ra 3 vùng làm việc của git:
  + Thư mục git: là nơi Git lưu trữ “siêu dữ liệu” (metadata), là cơ sở cho dự án ( cơ sở dữ liệu Git chứ không phải cơ sở dữ liệu của ứng dụng đang được phát triển). Đây là phần quan trọng nhất, nó được sao lưu về khi thực hiện lệnh tạo một bản sao từ máy khác về (clone)
  + Thư mục làm việc:  là bản sao một phiên bản của dự án, nó được kéo về (pulled)  từ cơ sở dữ liệu, được nén trong thư mục Git, lưu trên ổ cứng để bạn có thể sử dụng hoặc chỉnh sửa.
  + Khu vực tổ chúc (staging area): là một tập tin chứa trong thư mục Git, chứa thông tin về những gì sẽ được commit trong lần commit sắp tới.
* Checkout the project: tạo bản sao của dự án.
* Stage files: chuyền tập tin từ trạng thái đang làm việc bình thường chuyển sang đóng.
* Lưu trữ lại.

Command line: chế độ dòng lệnh.

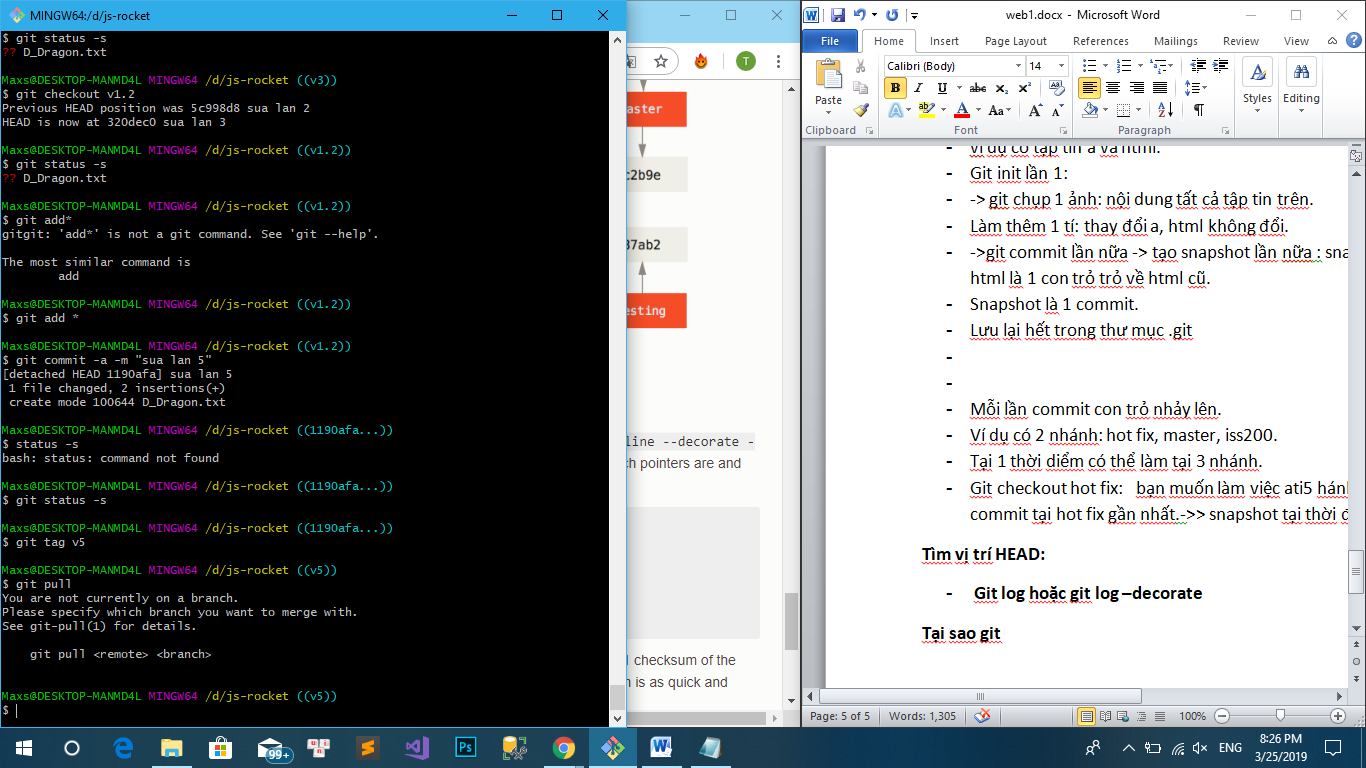
* Vào home+r:cmd. Gõ các lệnh:VD: D:->Enter->DIR.
* 3 cửa sổ gõ lệnh cho máy:
  + Home+r:powershell.
  + Git bash.
  + Home+r:cmd.
* Vào etc trong git:
* 
* Vào git config --edit --system: cách thoát: :qa->enter.
* Tìm đúng đường dẫn trong :git config –global core.editor notepad.
* Bật phần hướng dẫn trong git: git config –help.
* Tạo kho chứa cho git:
  + Kho chứa là gì: vd tạo thư mục không sẽ có thư mục bình thường, thêm git vào sẽ thành kho chứa.
    - 2 kho chứa:
      * Kho chứa cục bộ: là kho chứa nằm trên chính máy của lập trình viên,
      * Kho chứa ở xa: là kho chứa nằm trên máy server chuyên dụng. Ví dụ, các kho chứa trên các server của Github, Gitlab, Bitbucket.
  + Có 2 cách để tạo kho chứa:
    - một là tạo kho chứa từ thư mục có sẵn mà chưa được tích hợp hệ thống Git bên trong ( 1 kho có sẵn).
    - hai là sao chép (clone) một kho chứa đã tồn tại
  + tạo kho chứa từ thư mục có sẵn:
    - cd d:/myproject
    - git init
  + đổi tên mới:phải đổi bằng chương trình git:
* git init ->tạo 1 tập .git(ẩn): từ khi có nó sẽ trở thành kho chứa.
* chứa tập tin .gitconfig: users->max->.gitconfig
* thư mục: tracked:mọi thứ đang được theo dõi.
* Clean: sạch sẽ.
* Tạo 1 tập tin(code.c++): untracked.
* Add git vào ->tracked.
* Vs: thêm vào dòng code vào tập tin-> modified.
* Git add code.c++->staged-> commit->commited.
* Thấy được .git:
* 
* Có MM: mở tập tin ra thêm vào code->modified-> gõ git add chính tập tin đó.
* Mở ra thêm code1 ->git status->MM.
* Muốn commit thì phải thêm git add.

**Xóa exe tập tin muốn xoa trong git:**

* + **\*.txt**
  + **\*.exe**
  + **!test2.exe**
  + **Vendor/**
* **Lệnh commit trong git(commit:lưu phiên bản)**
  + **1 dư án(project): có hàng loạt phiên bản.**
  + **Vd:**
    - **Từ commit 4 có lỗi: tách từ commit 3 ra:**
      * **1: sửa lỗi.**
      * **2: làm tiếp.**
  + **Tất cả nằm trong .git.**
  + **Undo:khôi phục(hủy thao tác).**
  + 
  + 
  + **Git pull -> git push**
  + **Dir-> xem file trên origin master.**
* **Git init->commit lên->log in rồi commit (3)->home(commit)(4): xong step 1.**
  + **Đánh dấu tag 1: bản chất là con trỏ.**
* Snapshot
* ví dụ có tập tin a và html.
* Git init lần 1:
* -> git chụp 1 ảnh: nội dung tất cả tập tin trên.
* Làm thêm 1 tí: thay đổi a, html không đổi.
* ->git commit lần nữa -> tạo snapshot lần nữa : snap shot 2: a đã thay đổi, html là 1 con trỏ trỏ về html cũ.
* Snapshot là 1 commit.
* Lưu lại hết trong thư mục .git
* Mỗi lần commit con trỏ nhảy lên.
* Ví dụ có 2 nhánh: hot fix, master, iss200.
* Tại 1 thời diểm có thể làm tại 3 nhánh.
* Git checkout hot fix: bạn muốn làm việc ati5 hánh hot fix. Làm việc trên commit tại hot fix gần nhất.->> snapshot tại thời điểm commit gần nhất.

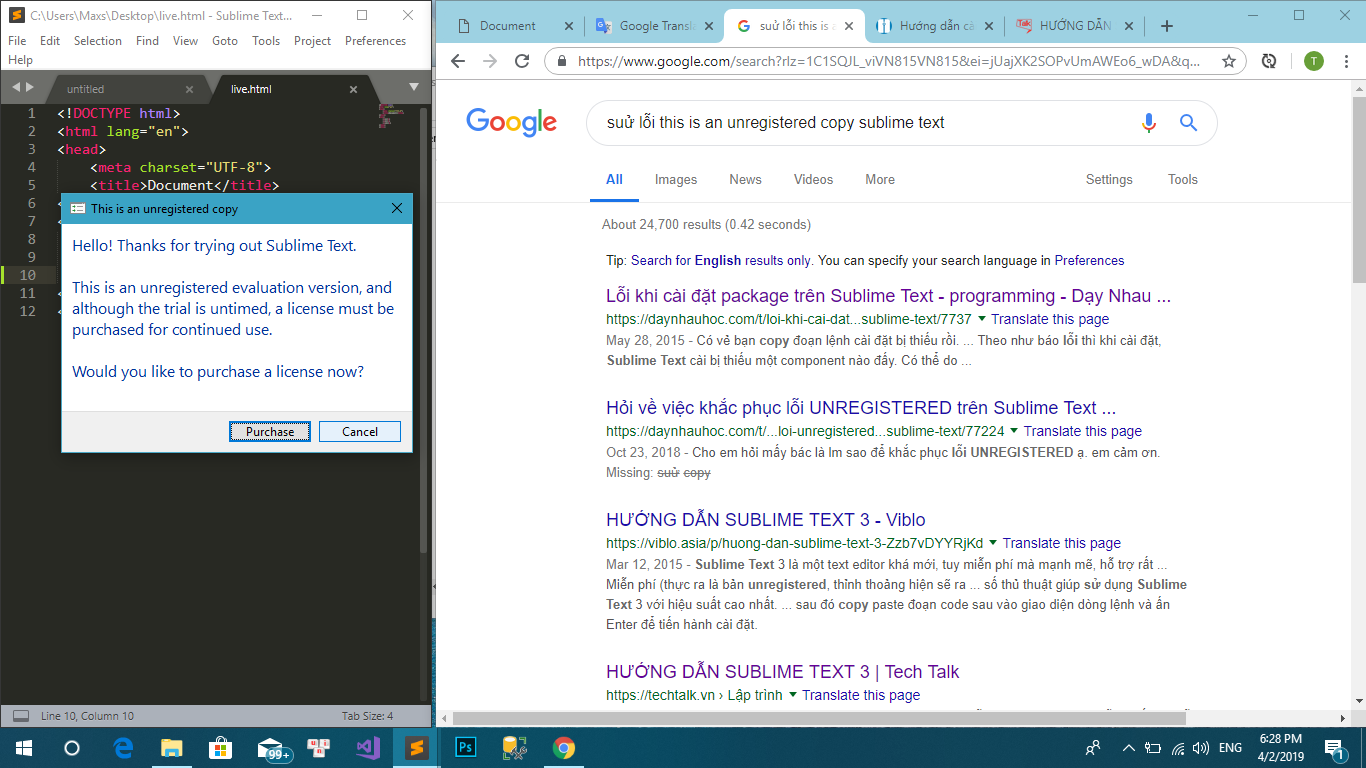
**Tìm vị trí HEAD:**

* **Git log hoặc git log –decorate**



**Tại sao git pull không thực hiện được khi ở v5 branches?**

**Framework: tối ưu hóa trang web.**

* **HTML**
* **CSS**
* **jSS  
  Jquery**
* **bookstrap:framework: kết hợp 3 cái trên.**
* **vueJS/ Angular/ React: 1 trong 3 cái.**
* **UI/UX: trải nghiệm ứng dụng.**
* **Backend:**
  + **PHP,C#, Ruby, Python, Java/ NodeJS: 1 trong những cái này.**
* **Học thêm framework cho từng các backend trên:**
  + **Laravel: catePhp**
  + **C#: ASP.net**
  + **Rubyenreals.**
* **CMS( wordpress/prupal) MySQL, SQL.**
* 

**Website**

**Pass: 2rGHz1!CqttDuT%Ojr)p**

**dhkufdkgjr.000webhostapp.com**

**dhkufdkgjr1**

**AVBFMvm\*C2KxH4bm&vPm : pass.**

**Tubi1234**

**Pass:** **3ADy!)UM#yHD(SJWwUOC**

**Vào tập tin httpd-vhost.conf**

