



# GitHub

**ỨNG DỤNG GITHUB TRONG  
QUẢN LÝ SOURCE CODE  
VÀ LÀM VIỆC NHÓM**

# Nội Dung Chính



1

Tổng Quan Về Git

2

Tổng Quan Về GitHub

3

Ứng dụng vào làm việc nhóm

4

GitHub Student Pack

# Tổng Quan Về Git



Giới thiệu về Git

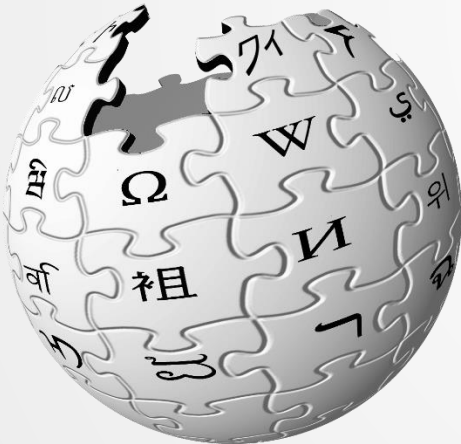


Làm rõ một số khái niệm trong Git



Hướng dẫn một số lệnh cơ bản trong Git

# Git là gì?



**WIKIPEDIA**  
*The Free Encyclopedia*

- Phần mềm quản lý mã nguồn phân tán
- Phần mềm mã nguồn mở theo giấy phép công cộng GPL2
- Là một dạng của hệ thống quản lý phiên bản – Version Control System (VCS)

**Nó ra đời khi nào?**



**Năm 2005**

# Ai là người tạo ra nó?

no wrong answers



# Ai là người tạo ra nó?



**Linus Torvalds**, *Linux Creator  
and Fellow at The Linux Foundation*

# Mục đích tạo ra nó là gì?

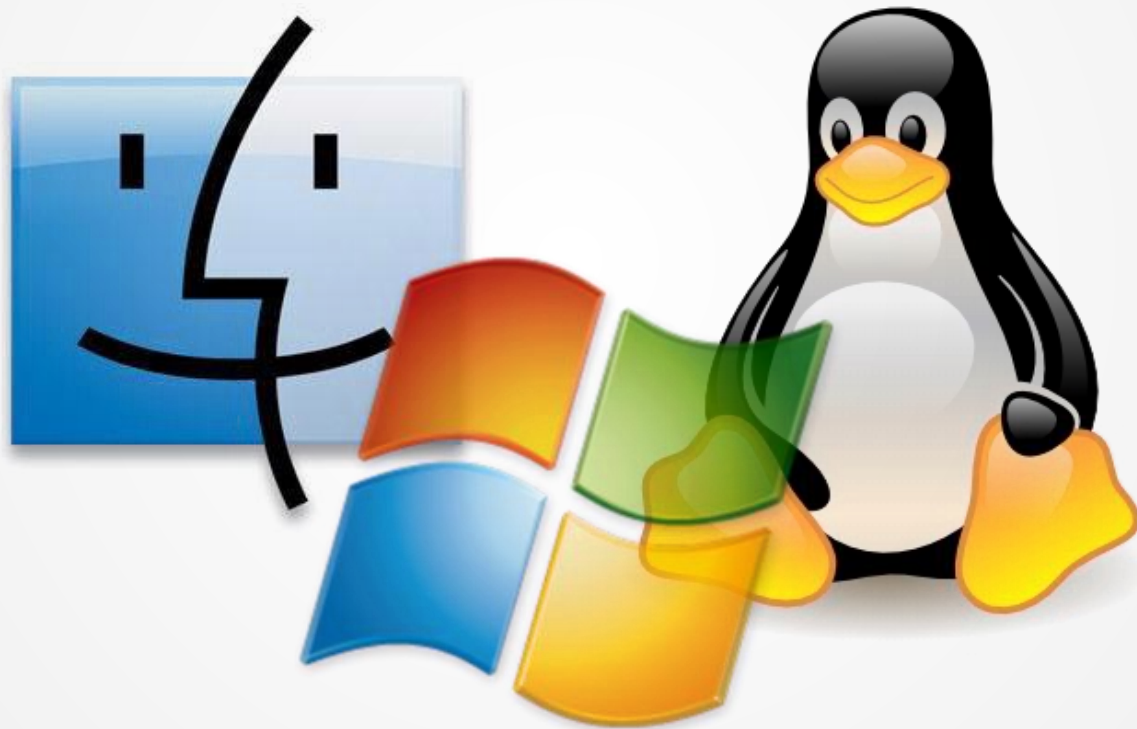
- Ban đầu là dành cho việc phát triển nhân Linux
- Hiện nay, Git trở thành một trong các phần mềm quản lý mã nguồn phổ biến nhất.



**WIKIPEDIA**  
*The Free Encyclopedia*



**Nó là phần mềm  
vậy nó chạy được ở đâu?**



# Tại sao phải sử dụng git



- Git dễ sử dụng, an toàn và nhanh chóng.
- Có thể giúp quy trình làm việc code theo nhóm đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch).
- Bạn có thể làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ kho chứa hoặc clone một phiên bản thay đổi nào đó từ kho chứa, hoặc một nhánh nào đó từ kho chứa.
- Dễ dàng trong việc deployment sản phẩm.
- Và nhiều hơn thế nữa.

# Tổng Quan Về Git



Giới thiệu về Git



Làm rõ một số khái niệm trong Git



Hướng dẫn một số lệnh cơ bản trong Git

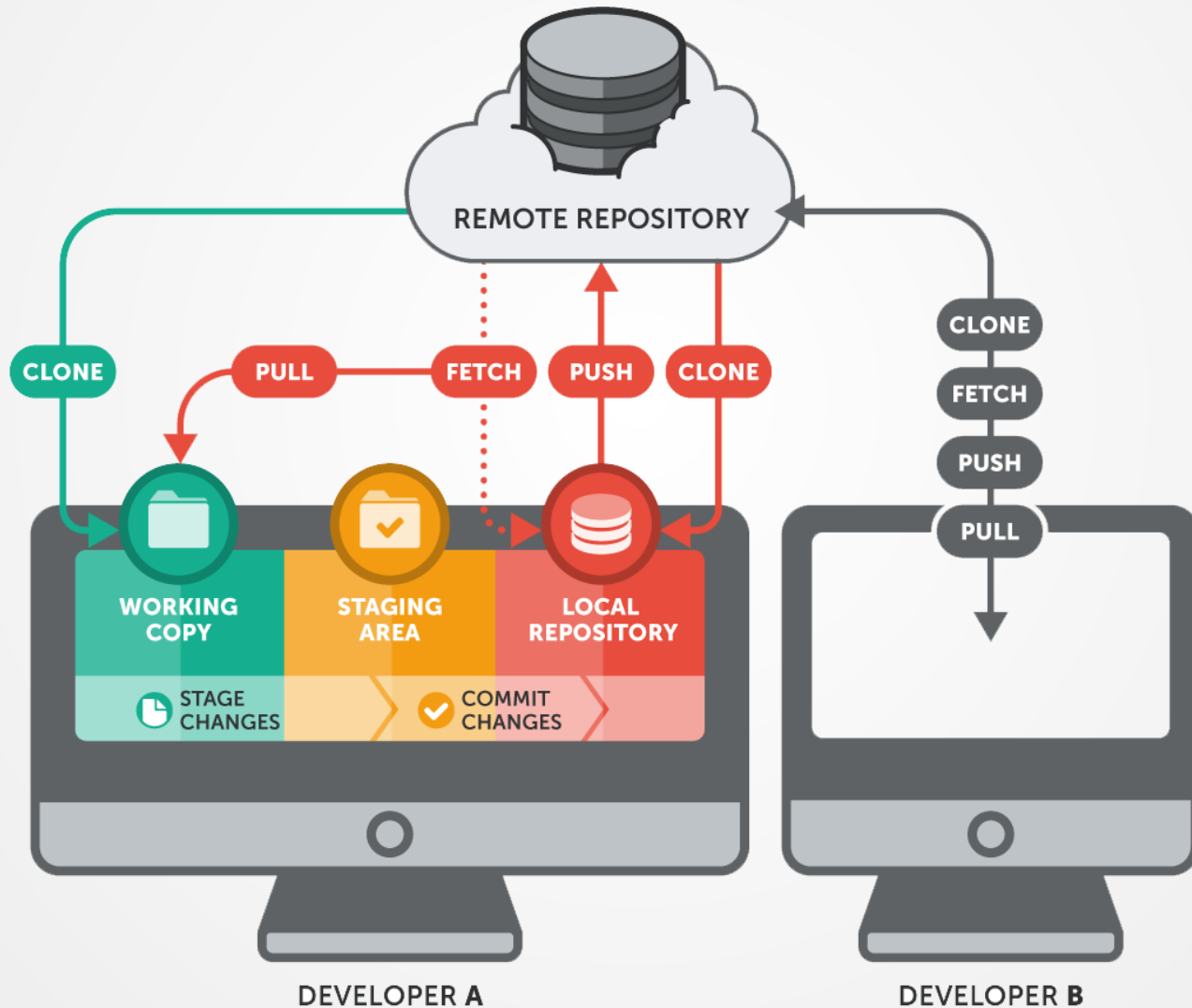
# Kho ( Repository )

- Gọi tắt là repo.
- Là nơi chứa tất cả những thông tin cần thiết để duy trì và quản lý các sửa đổi và lịch sử của toàn bộ project.
- Trong Repo có 2 cấu trúc dữ liệu chính là Object Store và Index.
- Tất cả dữ liệu của Repo đều được chứa trong thư mục bạn đang làm việc dưới dạng folder ẩn có tên là .git

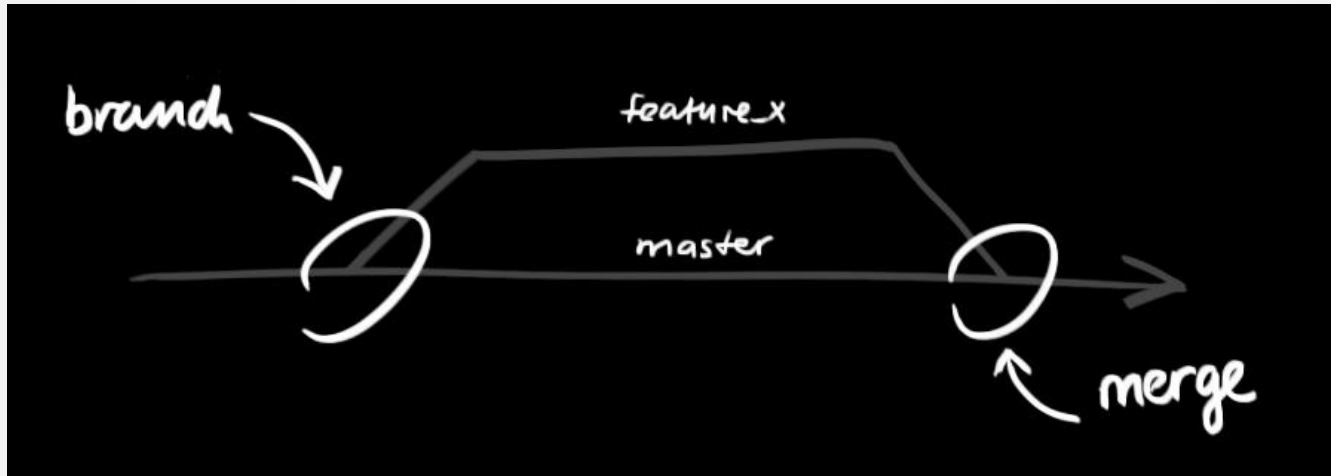
# Remote Repository và Local Repository

- Repository của Git được phân thành 2 loại là remote repository và local repository.
- 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.
- Công việc bình thường thì có thể sử dụng local repository
- Khi muốn public nội dung công việc mà mình đã làm trên local repository, thì ta sẽ upload lên remote repository rồi public
- Thông qua remote repository bạn cũng có thể lấy về nội dung thay đổi của người khác.

# Remote Repository và Local Repository

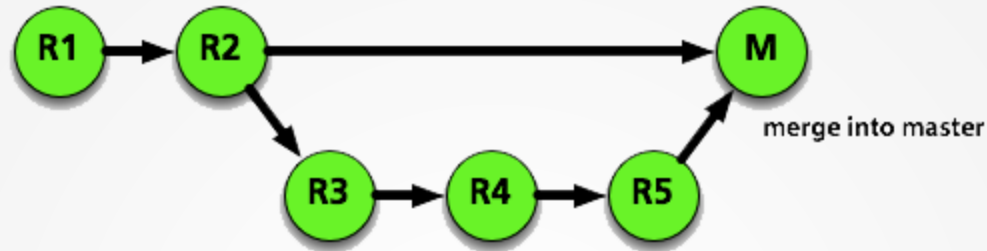


# Nhánh ( Branch )



- Mỗi nhánh trong Git gần giống như một workspace
- Việc nhảy vào một nhánh để làm việc trong đó tương tự việc chuyển qua ngữ cảnh làm việc mới, và sau đó có thể nhanh chóng quay lại ngữ cảnh cũ.
- Nhánh (branch) được dùng để phát triển tính năng mới mà không làm ảnh hưởng đến code hiện tại.
- Nhánh master là nhánh “mặc định” khi bạn tạo một repository.

# Trộn ( Merge )



- Việc hợp nhất 2 nhánh lại được gọi là merge.
- Chú ý:
  - Kiểm tra branch hiện đang làm việc trước khi merge
  - Phải đẩy tất cả những thay đổi dưới máy local lên Git trước khi merge
  - Trước khi merge phải lấy hết những thay đổi mới nhất của các branch khác, hay ít nhất là branch cần merge về máy
  - Merge thành công thì nên đẩy source lên lại lên server
  - Nên merge bằng GUI tool.



# Xung đột ( Conflict )

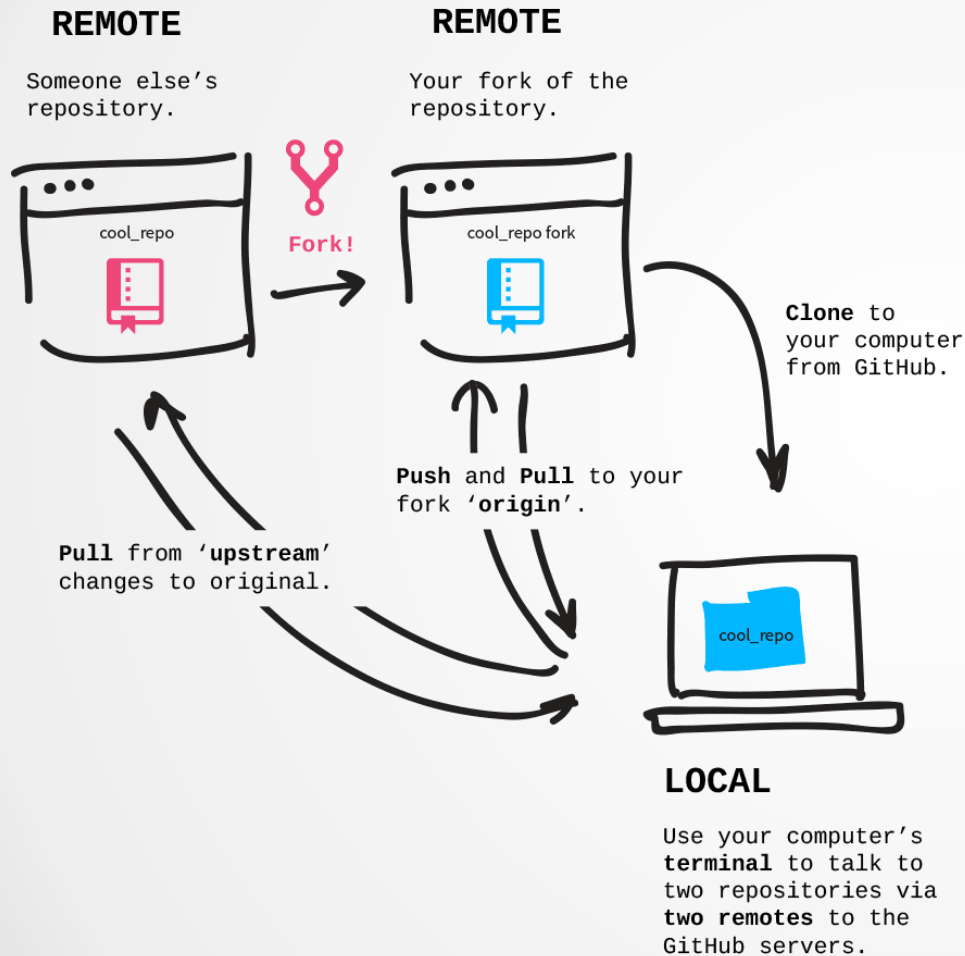
```
1 <<<<<< HEAD
2
3 Here is the original change.
4 =====
5 Here is the modified change.
6 >>>>>> 58326c301d09b58f3ac23d616e73f7b478424cc5
7
```

- Là trường hợp có 2 sự thay đổi trong một dòng code và máy tính không thể tự quyết định dòng code nào là đúng với ý đồ của lập trình viên.
- Để giải quyết mâu thuẫn, bạn chỉ việc nhìn vào file bị conflict và tự quyết định dòng code nào giữ lại, dòng nào xóa bỏ.

# Commit

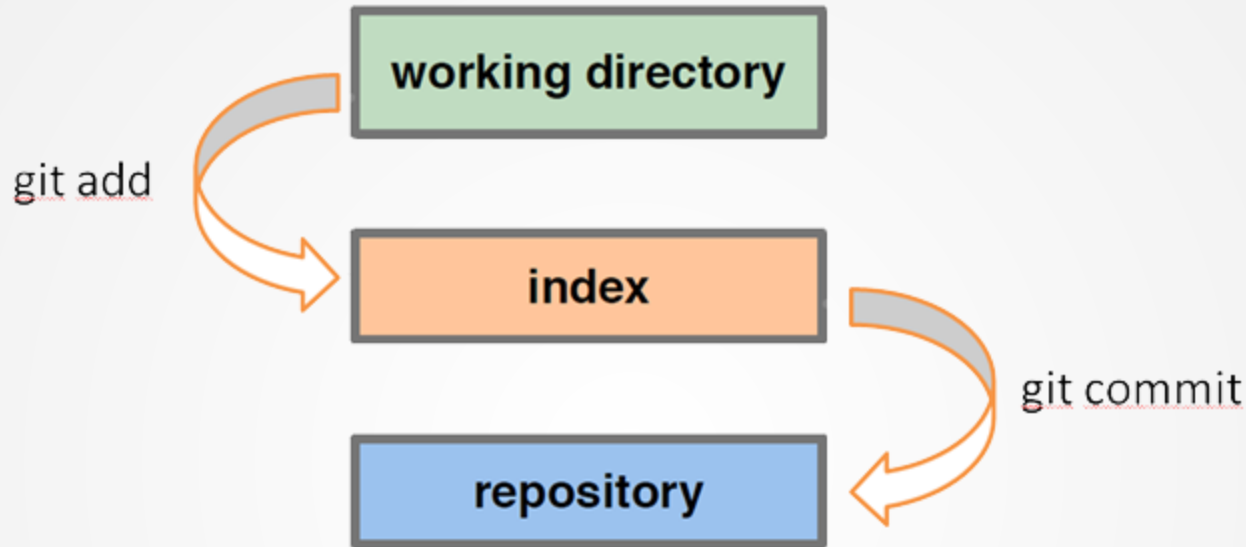
- Để ghi lại việc thêm/thay đổi file hay thư mục vào repository thì sẽ thực hiện thao tác gọi là Commit.
- Khi thực hiện commit, trong repository sẽ tạo ra commit (hoặc revision) đã ghi lại sự khác biệt từ trạng thái đã commit lần trước với trạng thái hiện tại.
- Commit này đang được chứa tại repository, các commit nối tiếp với nhau theo thứ tự thời gian. Bằng việc lần theo commit này từ trạng thái mới nhất thì có thể biết được lịch sử thay đổi trong quá khứ hoặc nội dung thay đổi đó.
- Các commit này, được đặt tên bởi 40 ký tự alphabet (mã md5 thì phải) không trùng nhau được băm từ thông tin commit. Bằng việc đặt tên cho commit, có thể chỉ định commit nào từ trong repository.
- Mỗi commit đều có yêu cầu phải có commit message, để giải thích commit này là bạn đã làm gì trong này.

# Git Remote



- Để kết nối được với một repo khác người ta sử dụng một khái niệm gọi là remote.

# Working Tree và Index



- Những thư mục được đặt trong sự quản lý của Git mà mọi người đang thực hiện công việc trong thực tế được gọi là working tree.
- Giữa repository và working tree tồn tại một nơi gọi là index. Index là nơi để chuẩn bị cho việc commit lên repository.

# Tổng Quan Về Git



Giới thiệu về Git



Làm rõ một số khái niệm trong Git



Hướng dẫn một số lệnh cơ bản trong Git

# Một Số Lệnh Cơ Bản

| Lệnh Git                                                                                                        | Công dụng                                            |
|-----------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| git init                                                                                                        | Tạo một repository mới                               |
| git clone /đường-dẫn-đến/repository/<br><br>git clone<br>tênusername@địachủmáychủ:/đường-<br>dẫn-đến/repository | Sao chép (clone) một repository                      |
| git branch                                                                                                      | Kiểm tra nhánh hiện tại                              |
| git branch <Tên nhánh>                                                                                          | Tạo nhánh mới                                        |
| git branch -d <Tên nhánh>                                                                                       | Xóa một nhánh                                        |
| git checkout <Tên nhánh>                                                                                        | Chuyển nhánh                                         |
| git checkout -b <Tên nhánh>                                                                                     | Tạo một nhánh mới và chuyển qua nhánh đó (từ master) |
| git checkout master                                                                                             | Trở lại nhánh master                                 |

# Một Số Lệnh Cơ Bản

| Lệnh Git                                                                                                                                                                                                                | Công dụng                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| <p>#Cập nhật tất cả<br/>git add .</p> <p>#Cập nhật tất cả các file có dạng<br/>git add *.&lt;tên đuôi&gt;</p> <p>#Cập nhật file theo tên<br/>git add &lt;tên file 1&gt; [&lt;tên file 2&gt; ... &lt;Tên file n&gt;]</p> | Cập nhật tình trạng các file( thêm, xóa, sửa) trong project được quản lý bởi GIT |
| git commit -am "<Ghi chú>"                                                                                                                                                                                              | Xác nhận và lưu những thay đổi của project                                       |
| git push origin <Tên nhánh>                                                                                                                                                                                             | Cập nhật những thay đổi của project và gửi lên server                            |
| git remote add origin <máy-chủ>                                                                                                                                                                                         | Kết nối repository của bạn đến máy chủ remote                                    |
| git fetch <Tên nhánh>                                                                                                                                                                                                   | Lấy source mới nhất trên server về để lên source hiện tại trên máy local         |
| git pull <Tên nhánh>                                                                                                                                                                                                    | Lấy source mới nhất trên server về và tiến hành trộn                             |

# Một Số Lệnh Cơ Bản

| Lệnh Git                 | Công dụng          |
|--------------------------|--------------------|
| git tag 1.0.0 1b2e1d63ff | Gắn nhãn (tagging) |
| git log                  | Lấy mã commit      |



# Tổng Quan Về GitHub



Giới thiệu về GitHub



Hướng dẫn cách tạo tài khoản GitHub

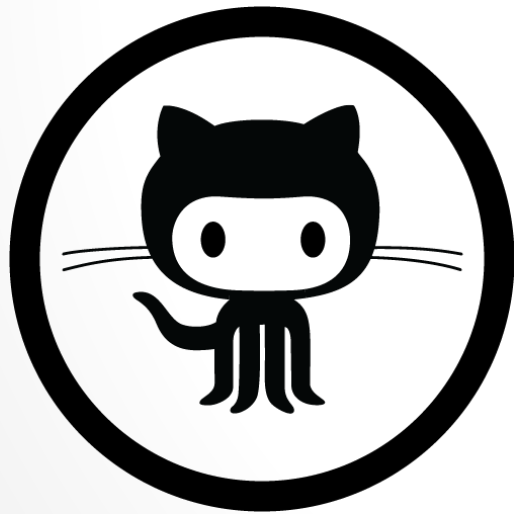
# GitHub là gì?

- Là một dịch vụ lưu trữ dựa trên web cho các dự án phát triển phần mềm
- Sử dụng các hệ thống kiểm soát phiên bản Git.



**WIKIPEDIA**  
*The Free Encyclopedia*

# GitHub có những loại tài khoản nào?



- GitHub hiện có 2 loại tài khoản:
  - Trả tiền dành cho kho tư nhân.
  - Tài khoản miễn phí dành cho các dự án mã nguồn mở.

**Số người sử dụng GitHub  
hiện tại là bao nhiêu?**

**3,4 triệu người dùng.**

**Đang trở thành code host lớn nhất trên thế giới**

# Tổng Quan Về GitHub



Giới thiệu về GitHub



Hướng dẫn cách tạo tài khoản GitHub

# Đăng ký tài khoản GitHub như thế nào?

Join GitHub  
The best way to design, build, and ship software.

Step 1: Set up a personal account | Step 2: Choose your plan | Step 3: Tailor your experience

Create your personal account

Username  
  
This will be your username — you can enter your organization's username next.

Email Address  
  
You will occasionally receive account related emails. We promise not to share your email with anyone.

Password  
  
Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

Create an account

You'll love GitHub

- Unlimited collaborators
- Unlimited public repositories
- ✓ Great communication
- ✓ Frictionless development
- ✓ Open source community

Bạn có thể tạo tài khoản GitHub rất dễ dàng, miễn phí và nhanh chóng tại đây:

CLICK HERE  
↓ ↓

<https://github.com/join>

# Ứng dụng GitHub trong quản lý source code và làm việc nhóm

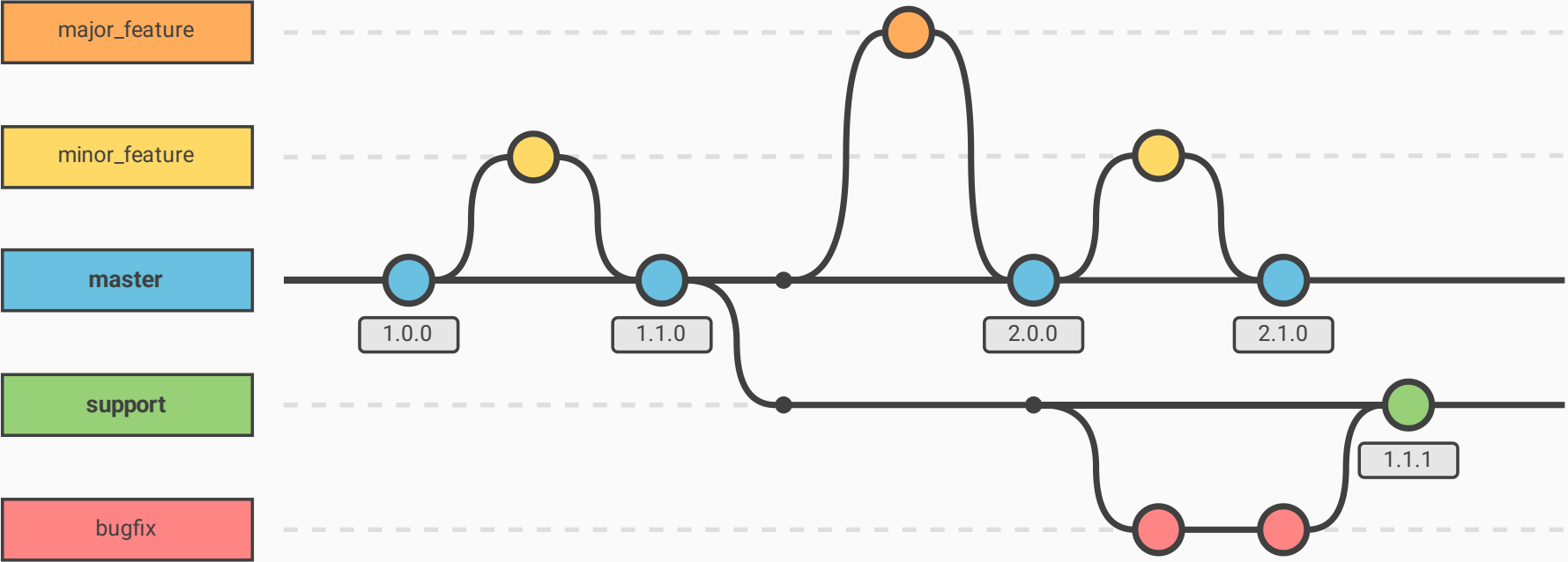


Hướng dẫn về cách sử dụng GitHub trong quản lý source code khi làm việc nhóm.

# Hướng Dẫn Làm Việc



# Git Diagram (1)

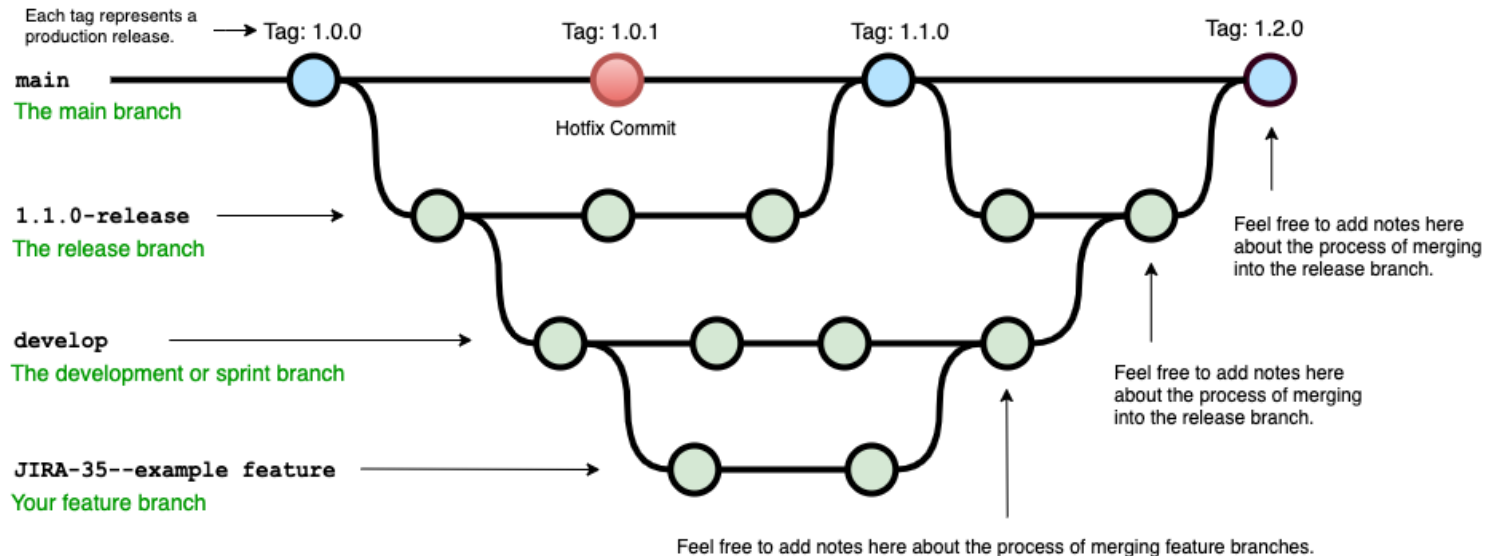


# Git Diagram (2)

## Example Git Branching Diagrams

### Example diagram for a workflow similar to "Git-flow" :

See: <https://nvie.com/posts/a-successful-git-branching-model/>



### Example diagram for a workflow with a simpler branching model:

See: <https://gist.github.com/jbenet/ee6c9ac48068889b0912> or <https://www.endoflineblog.com/oneflow-a-git-branching-model-and-workflow>

