

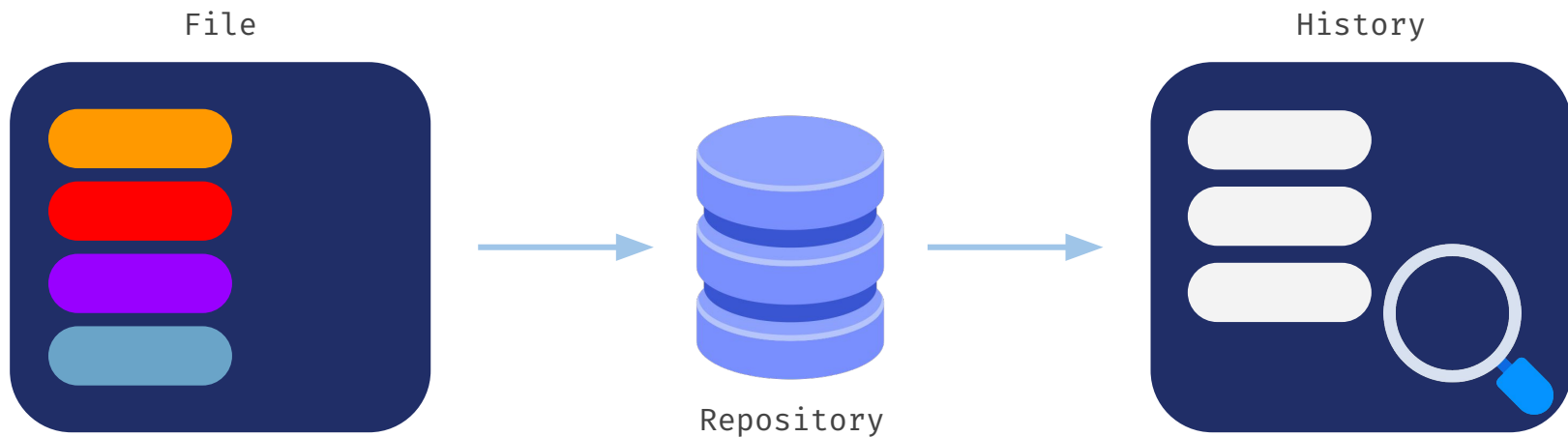


# Git Basic

Ba Nguyễn

# Quản lý phiên bản?

💡 Quản lý phiên bản (version control) là hệ thống ghi lại các thay đổi đối với tệp tin (files) theo thời gian, cho phép xem lại các thay đổi, đưa tệp tin về một phiên bản trước đó, ...



# Các hệ thống quản lý phiên bản



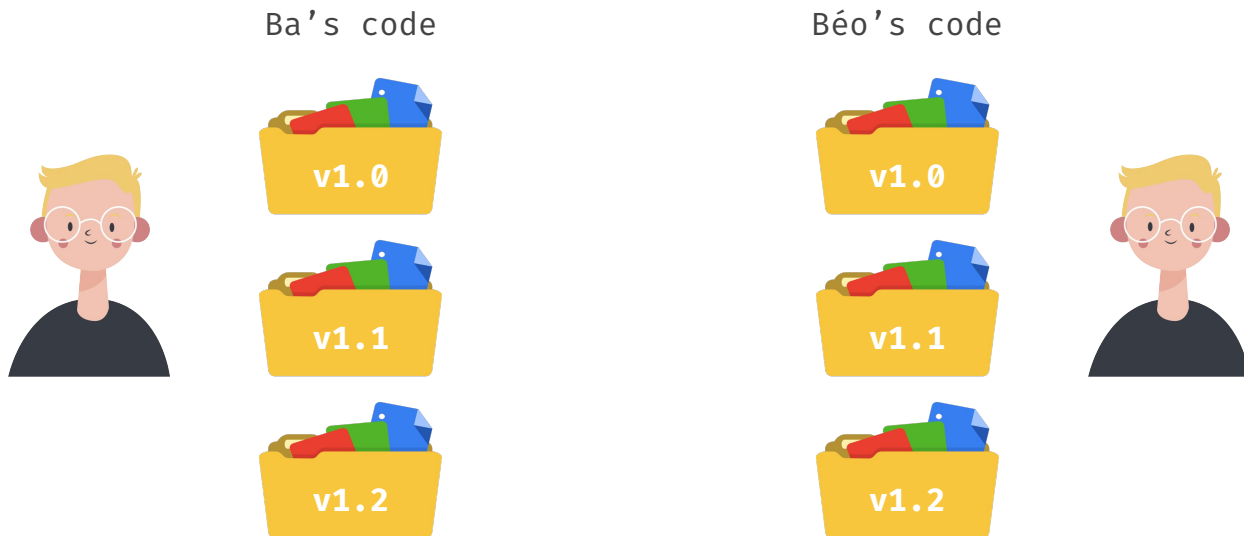
The diagram consists of three large circles arranged horizontally. The leftmost circle is light gray and contains the text 'Local (Cục bộ)'. The middle circle is purple and contains the text 'Centralized (Tập trung)'. The rightmost circle is light blue and contains the text 'Distributed (Phân tán)'.

**Local**  
(Cục bộ)

**Centralized**  
(Tập trung)

**Distributed**  
(Phân tán)

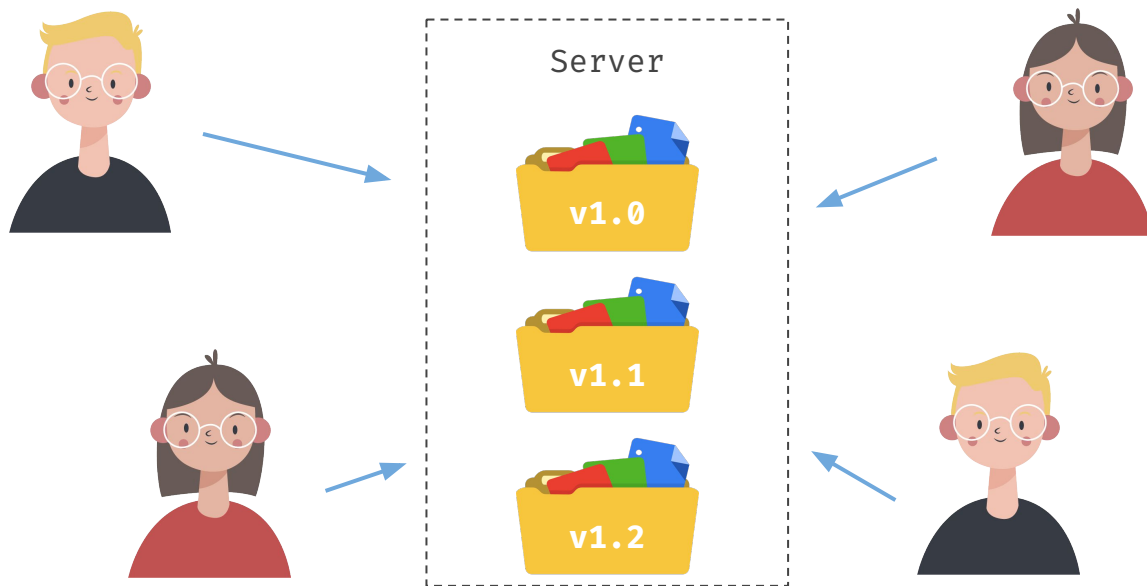
# Hệ thống quản lý phiên bản cục bộ



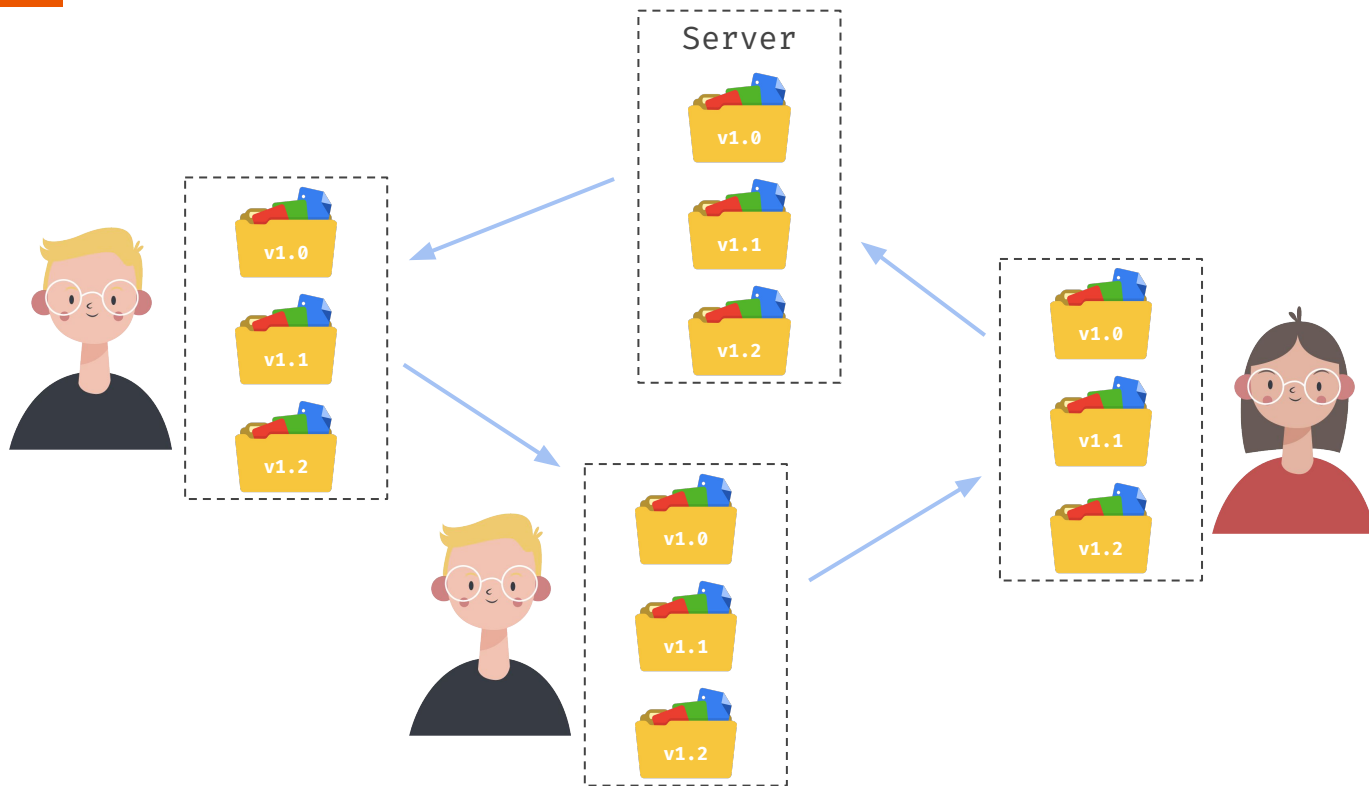
# Hệ thống quản lý phiên bản cục bộ



# Hệ thống quản lý phiên bản tập trung



# Hệ thống quản lý phiên bản phân tán



# Git là gì?

💡 **Git** là hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay, những lợi thế của **Git** so với các hệ thống quản lý phiên bản phân tán khác

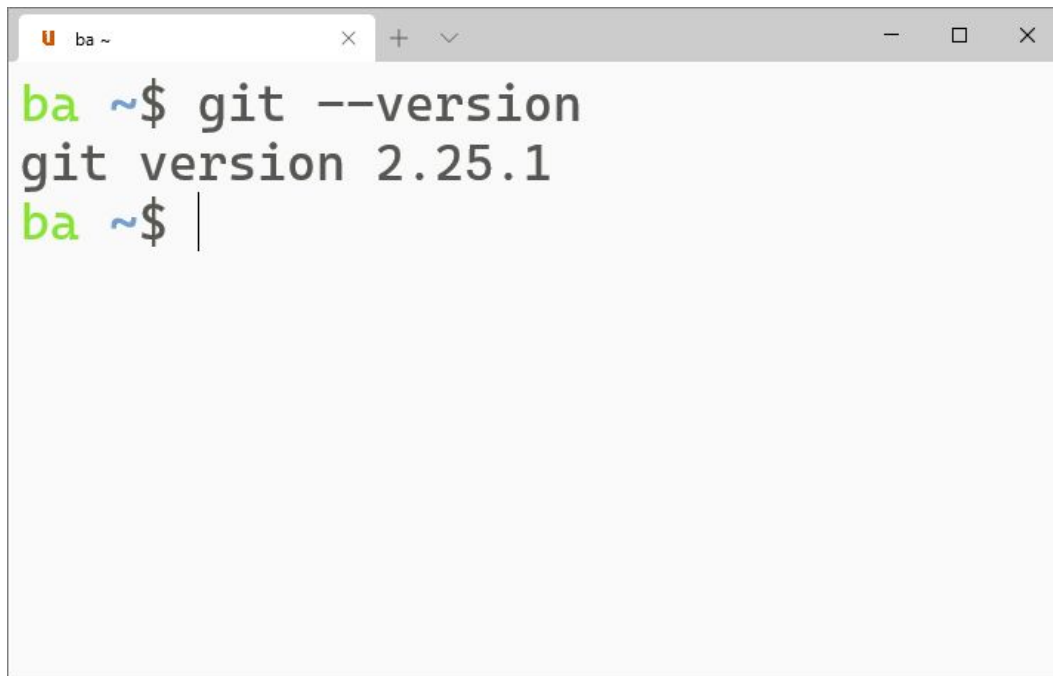
- Git là mã nguồn mở (open source) và hoàn toàn miễn phí
- Tốc độ nhanh
- Phân nhánh, gộp nhánh dễ dàng
- ...

💡 Tải và cài đặt **Git** tại đây: <https://git-scm.com/>



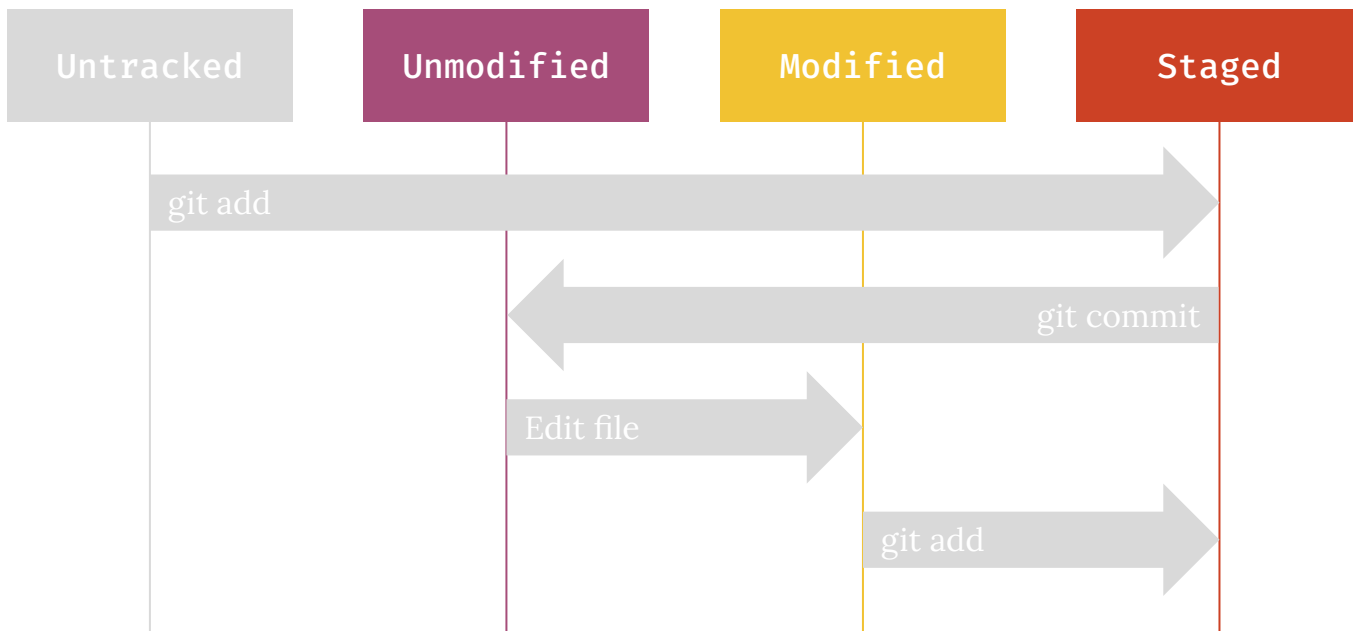


# Command-line

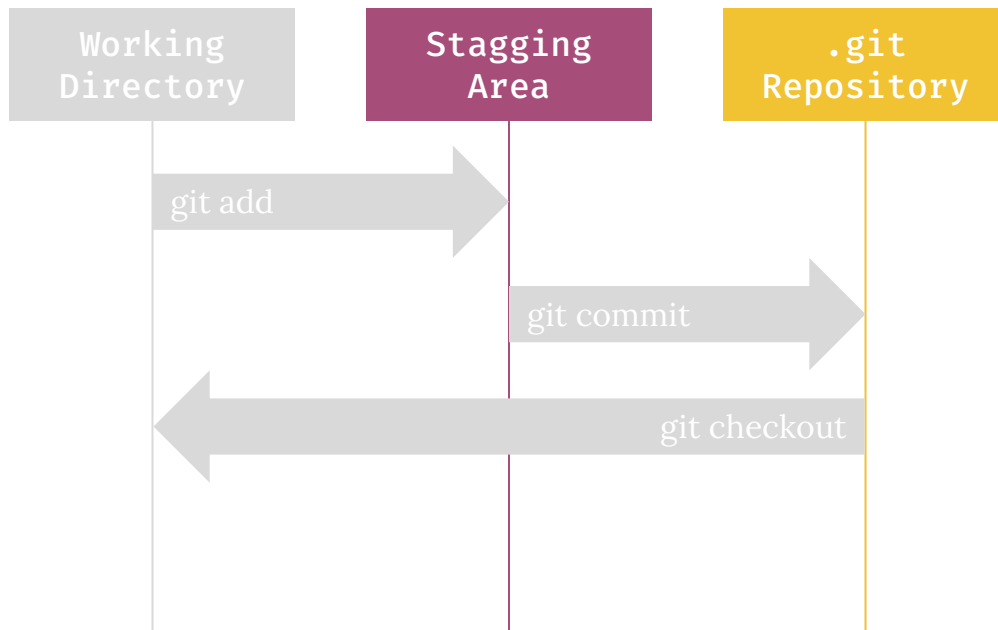
A terminal window with a title bar containing a red 'u' icon, the text 'ba ~', and window control buttons (close, maximize, and a dropdown arrow). The terminal content shows a user prompt 'ba ~\$' followed by the command 'git --version', the output 'git version 2.25.1', and a second prompt 'ba ~\$' with a vertical cursor.

```
ba ~$ git --version
git version 2.25.1
ba ~$ |
```

# Các trạng thái của file



# Work flow



# GIT Cheat Sheet

---

*# Git configuration*

```
$ git config --global user.name "Your Name"
$ git config --global user.email "Your Email"
$ git config --global color.ui auto
$ git config --global init.defaultBranch main
$ git config --global credential.helper store
```

*# Show git configuration*

```
$ git config --list --show-origin
```

# GIT Cheat Sheet



*# Tạo mới một local repository*

```
$ git init
```

*# hoặc*

```
$ git init project_name
```

*# Tải repository từ remote repository*

```
$ git clone project_url
```

# GIT Cheat Sheet

---

```
# Kiểm tra trạng thái repo  
$ git status  
# Thêm file vào staging  
$ git add [file]  
$ git add [folder]  
$ git add .  
# Hủy bỏ các thay đổi của tệp tin  
$ git checkout -- [file]  
# Xóa file khỏi staging  
$ git reset [file]  
# Tạo một commit mới với các thay đổi trong staging  
$ git commit -m "Message"
```

# GIT Cheat Sheet

---

*# Xem lịch sử commit*

`$ git log --oneline`

*# Bỏ theo dõi một tệp/thư mục*

*# Tạo file .gitignore ở thư mục gốc*

*# Không theo dõi thư mục*

`folder`

*# Không theo dõi file*

`file.extension`

`*.extionsion`

# Github

---

💡 **Github** là máy chủ lưu trữ mã nguồn Git (repository), nó cung cấp dịch vụ lưu trữ miễn phí cho các dự án mã nguồn mở, đồng thời cung cấp một máy chủ web miễn phí cho các trang web tĩnh



💡 Đăng ký tài khoản github tại: <https://github.com/>



# Github

Tạo repository mới trên Github:

Lưu ý tên repo chính có dạng:


`yourusername.github.io`

Trang web của bạn:

`yourusername.github.io`

## Create a new repository

Owner \*

 banx9x ▾

Repository name \*

banx9x.github.io ✓

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

**Initialize this repository with:**

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

# Github Cheat Sheet

---

*# Thêm một remote repo vào local repo hiện tại*  
\$ git remote add remote\_name remote\_url

*# Tải về các thay đổi mới nhất từ remote repo*  
*# và hợp nhất các thay đổi với local repo*  
\$ git pull [remote]

*# Đẩy các thay đổi mới nhất từ local repo*  
*# lên remote repo*  
\$ git push -u [remote]