

Git Basic.

Working git flow.

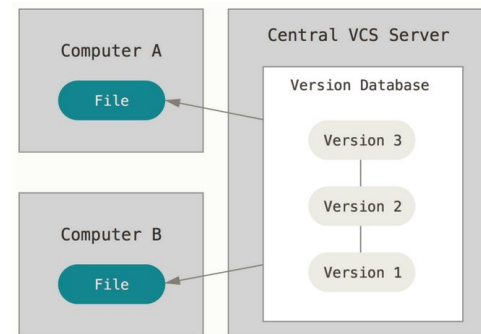
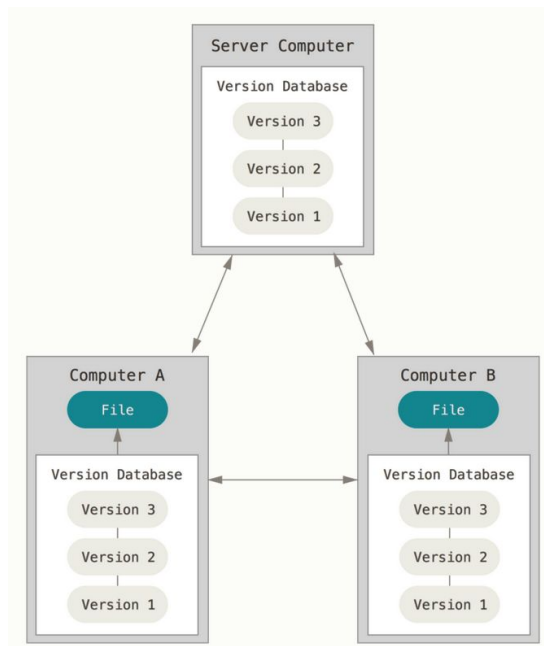
---

# Mục lục

- Version Control. Git History
- Git Basic
- Braching
- Github. Working git flow.
- Practice

# Version Control

- Hệ thống giúp ghi lại những thay đổi của files qua thời gian, có thể “revert” lại những version nhất định.
- Tại sao cần “version control”
- Local
- Centralized
- Distributed



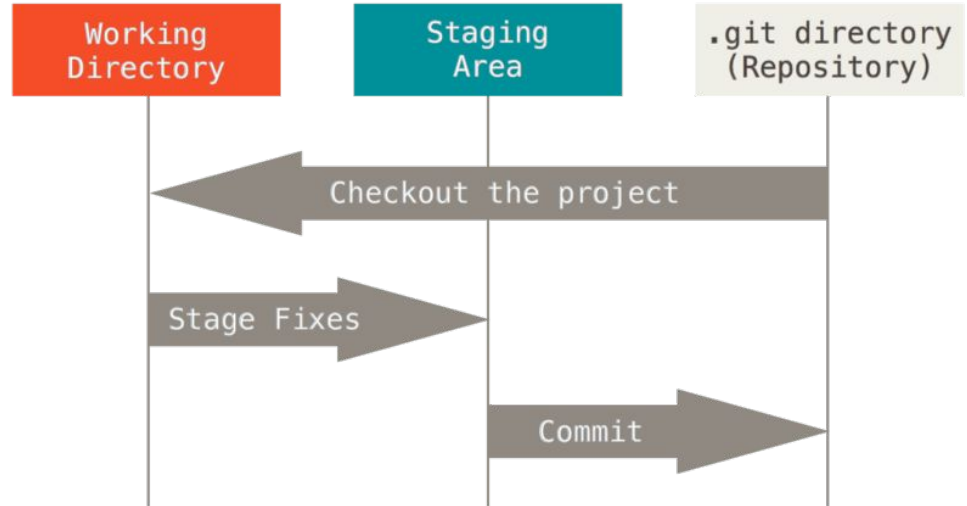
# Git

- Được tạo ra bởi cộng đồng nhà phát triển Linux (Dẫn đầu bởi Linus Torvalds) vào năm 2005.
- Nhanh, đơn giản, hỗ trợ non linear developments, full distributed, đủ sức mạnh để quản lý các project lớn (như Linux kernel)



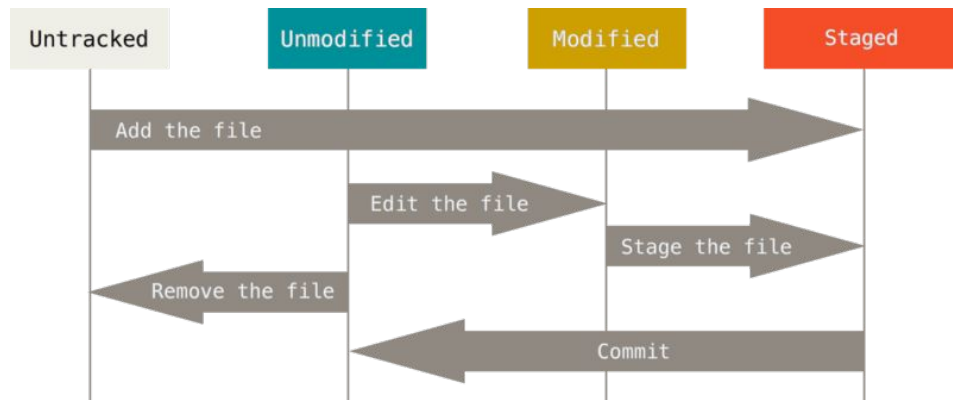
# Cơ chế hoạt động

- Snapshot
- Nearly Every Operation Is Local
- Git Has Integrity
- Git Generally Only Adds Data
- The Three States: modified, staged, committed



# Git Basic

- Cài đặt: <https://git-scm.com/>
- Setup repository: git init vs git clone
- git status
- git add
- git ignore
- git diff
- git commit
- git rm
- git mv
- git log



# Undoing things

- Lưu ý: Có một số trường hợp không thể undo các undos, nên cần cẩn thận trong từng commit
- `git commit --amend`
- `git reset HEAD <file>`
- `git checkout -- <file>`
- `git restore`

# Working with remote and tags

- Remote là gì?
  - git clone
  - git remote
  - git fetch
  - git push
  - git pull
  - git remote show <remote>
  - git remote rename <remote>
- Tag là gì?
  - git tag
  - Lightweight vs Annotated Tags



# Branch - Nhánh - Killer Feature của git

- **Quay lại:** Cách git xử lý mỗi commit
- **Branch** - đơn giản là một **con trỏ** trỏ vào commit
- **HEAD** - con trỏ đặc biệt, luôn trỏ vào branch hiện tại. Lệnh git checkout <branch> đơn giản là đặt lại HEAD
- Thực hành tạo branch và tìm hiểu kỹ ví dụ:

<https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>

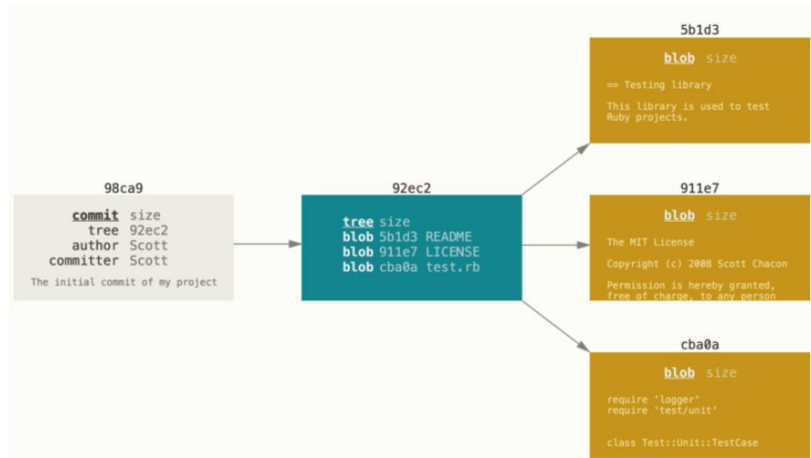
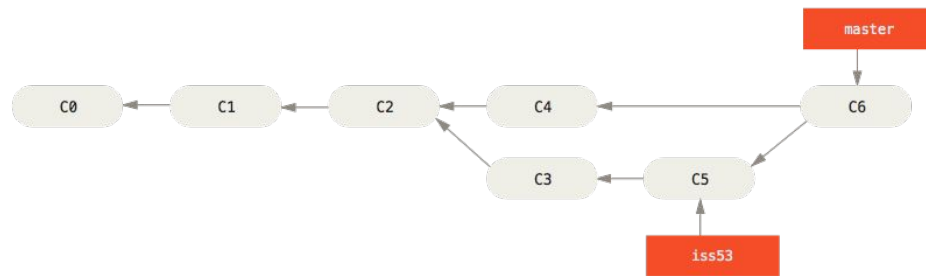
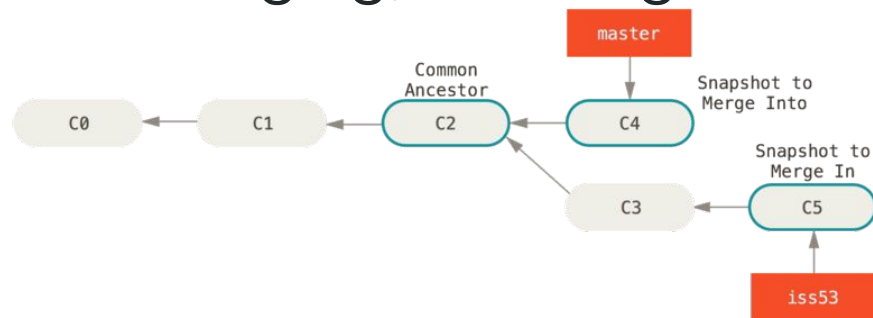


Figure 2. A commit and its tree

```
$ git add README test.rb LICENSE
$ git commit -m 'Initial commit'
```

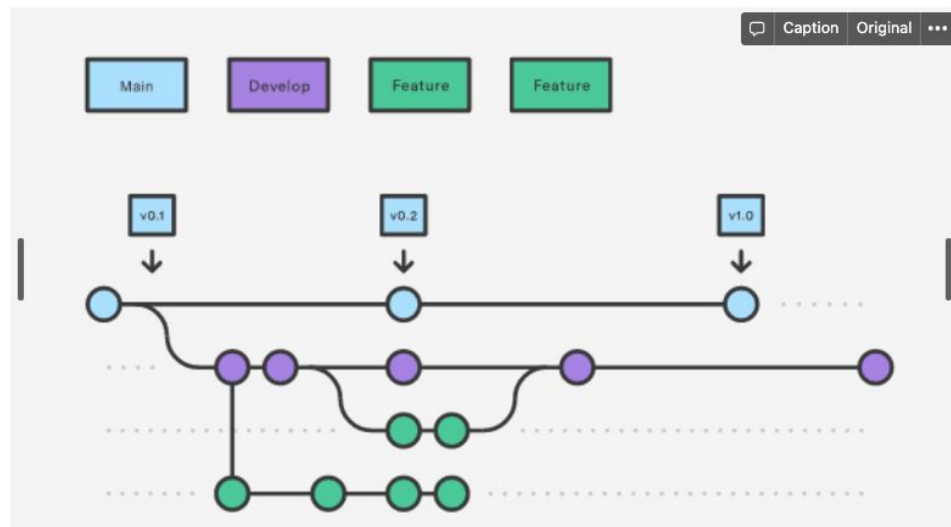
# Các lệnh cơ bản với branch, branch merging, rebasing

- git checkout -b
- git checkout
- git merge
- Conflict là gì? Cách resolve conflict?
- Rebase là gì?



# Git WorkFlow

- Tư tưởng chính
- Master branch và dev branch: Thay vì có 1 nhánh chính (master) mình sẽ có 2 nhánh chính: master và dev. Nhánh master chỉ chứa các official release history (chỉ chứa các commit: v1, v1.1, v2...). Nhánh dev chứa hoạt động như một công cụ để tích hợp các feature trên đó, chứa toàn bộ lịch sử feature của mọi người.
- Luồng làm việc trên git



# Tài liệu tham khảo

- Cài đặt git
- Sách Pro Git.pdf
- Tool: Git Kraken, Source Tree
-

Thank for listening