

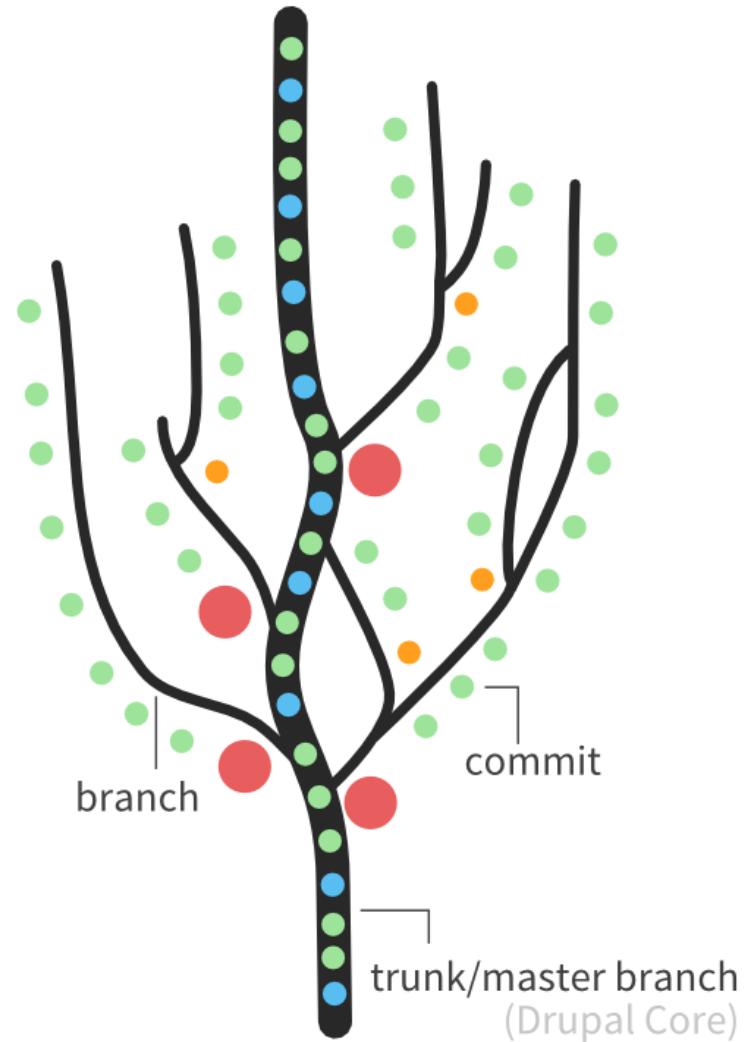


ADVANCED GIT FOR COLLABORATORS

Repository

Bao gồm:

- Một tập hợp các commit
- Commit có thể có 0, 1, hoặc 2 commit cha
- Chỉ duy nhất commit đầu tiên không có commit cha
- Từ một commit nhìn thấy nội dung của tất cả các commit cha, đệ quy
- Muốn nhìn thấy nội dung commit ở nhánh khác thì phải tạo ra commit merge (có 2 commit cha)

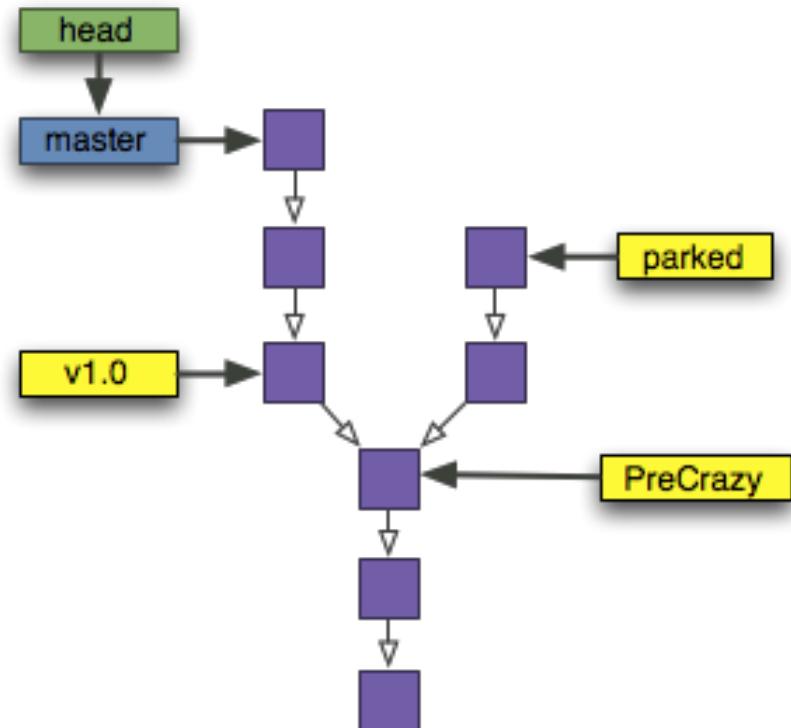


Repository (2)



(... tiếp theo)

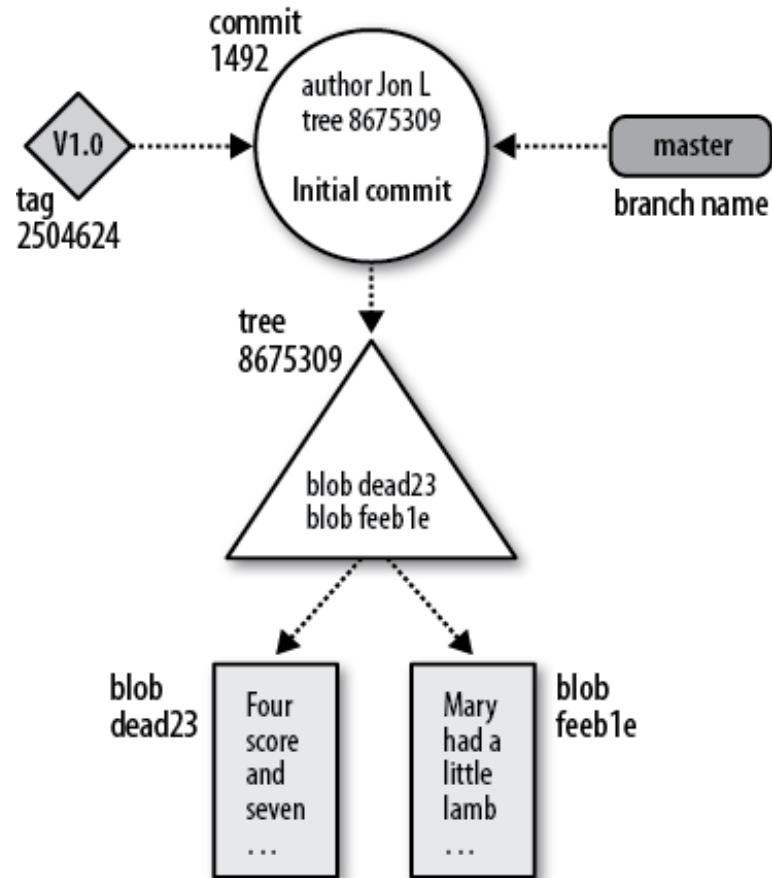
- Một tập hợp các tham chiếu được gọi chung là heads.
- Heads bao gồm các *nhánh*, các *tag*, và một tham chiếu đặc biệt gọi là HEAD
- Các lệnh của git có chức năng đọc trạng thái, thêm mới, thay đổi vị trí của các commit và heads.



Commit



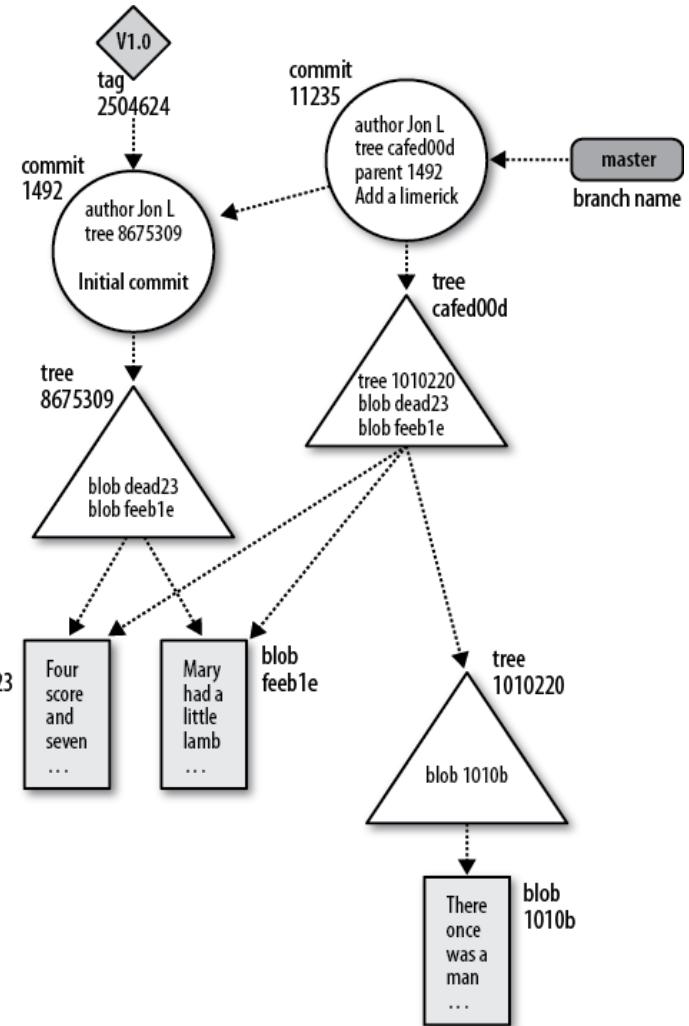
- Đại diện cho một ảnh chụp hoàn chỉnh của source code
- Đại diện bởi một *mã hash*
- Tham chiếu tới một đối tượng *tree*
- Đối tượng *tree* tham chiếu tới các *blob*
- Từ *blob* suy luận được nội dung file



Commit (2)

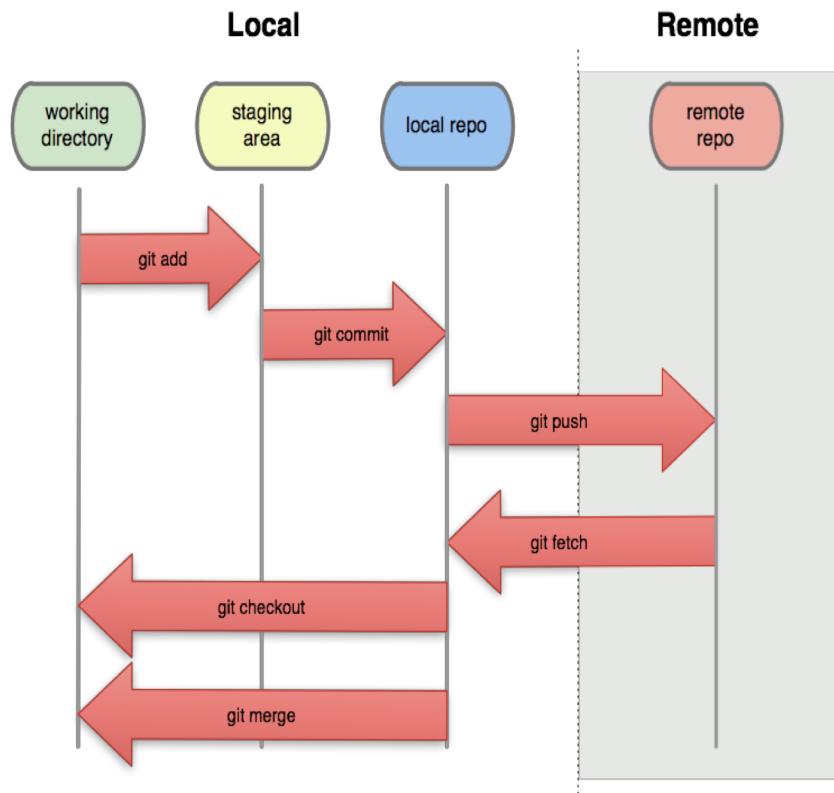


- Các tree sẽ tham chiếu tới cùng một blob nếu nội dung file không thay đổi.



Staging Area

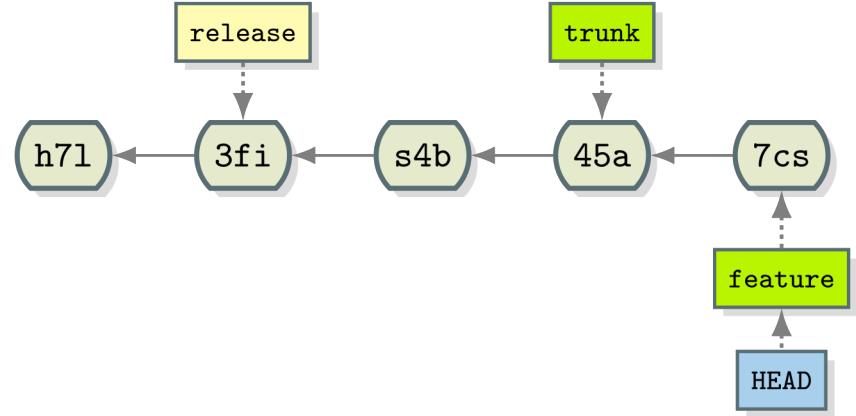
- Trước khi tạo commit thì phải dự thảo nội dung của nó
- Nội dung dự thảo được lưu trong một khu vực gọi là Staging, còn gọi là Index.
- Các lệnh git add, checkout – làm thay đổi nội dung dự thảo
- Lệnh git commit tạo commit mới và xóa nội dung tự thảo



HEAD



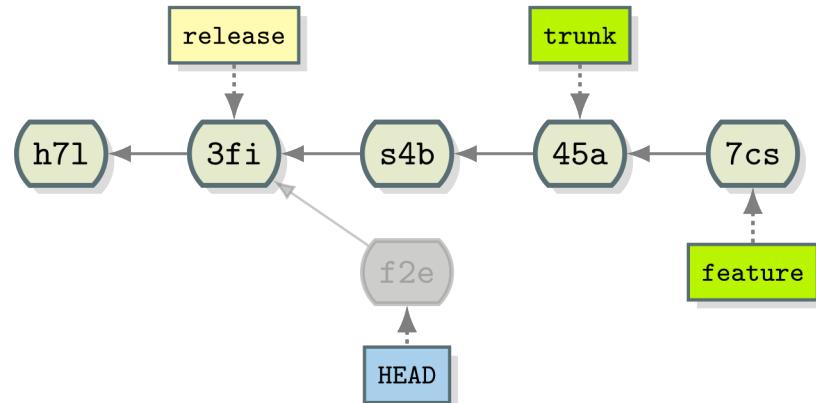
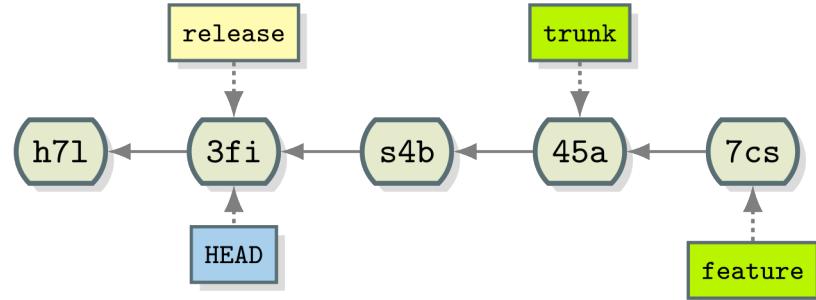
- Checkout tới đâu thì HEAD tham chiếu tới đó
- Các commit mới được tạo ra sẽ có cha là commit mà HEAD đang tham chiếu tới



Detached HEAD



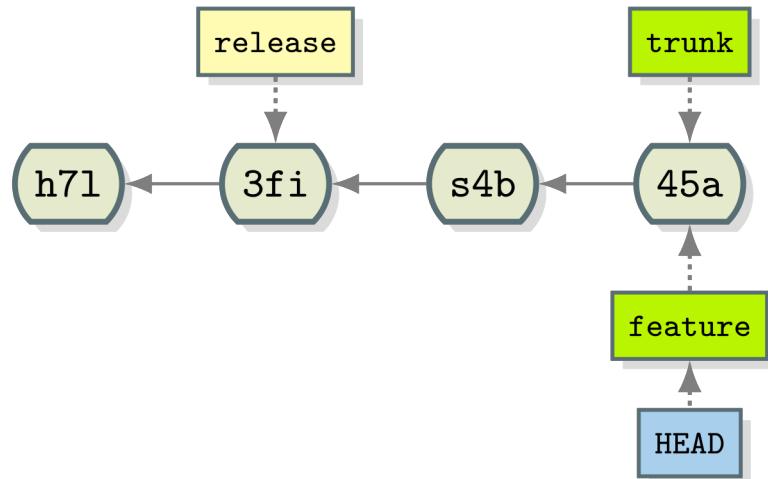
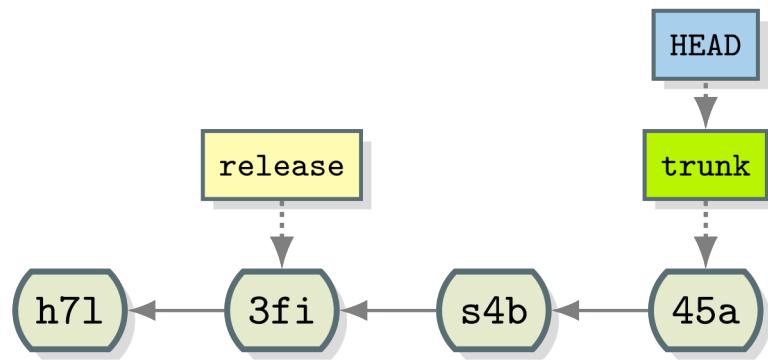
- Tình trạng HEAD không tham chiếu tới branch được gọi là **detached HEAD**.
- Phát triển tiếp trong tình trạng detached HEAD rất nguy hiểm, có thể gây ra mất mã nguồn. Trong những trường hợp đó cần cẩn trọng thao tác.



Nhánh



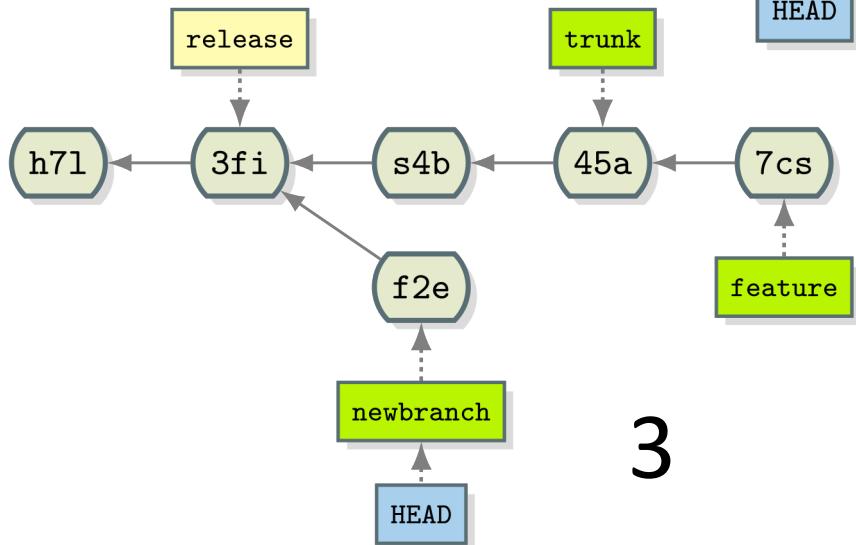
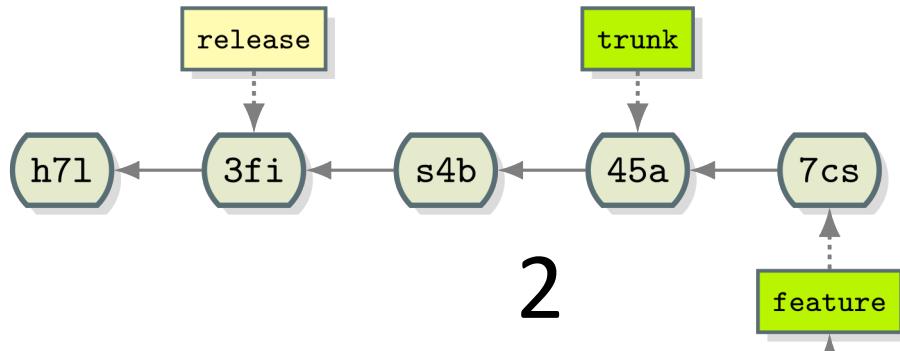
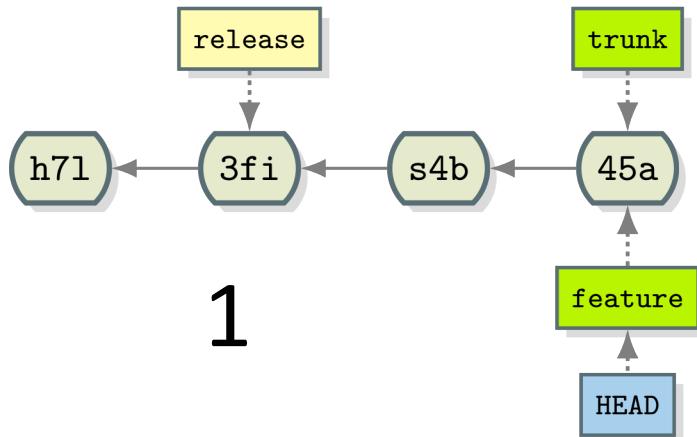
- Tạo ra một branch là tạo ra một head thuộc loại là *branch*
- HEAD có thể, và thường được để tham chiếu vào *branch* thay vì *commit*.



Nhánh (2)



- Tạo commit mới trong khi HEAD đang tham chiếu vào branch sẽ tạo ra tình trạng phân nhánh trên cây commit.





Nhánh master

- Là một nhánh được tạo ra ngay khi repository được khởi tạo, để repository có thể hoạt động bình thường
- Có ý nghĩa về mặt quy ước hơn là kỹ thuật

Merge nhánh

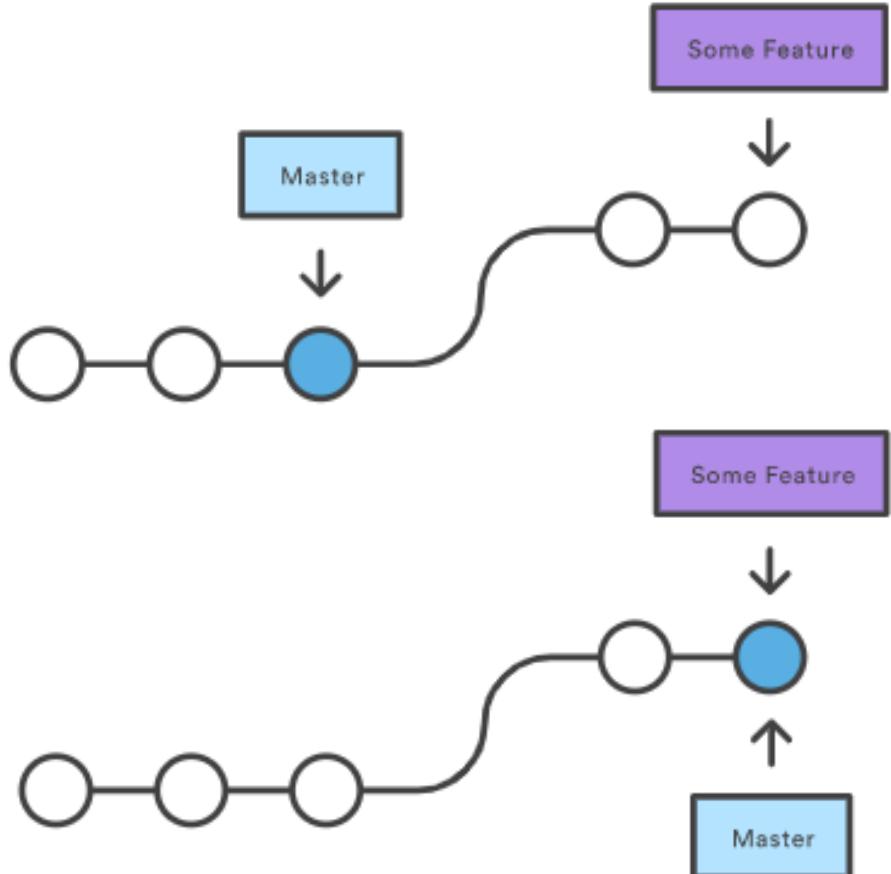


- Là thao tác khiến cho tại một commit của một nhánh có thể “nhìn thấy” commit tại một nhánh khác.

Merge Fast-forward



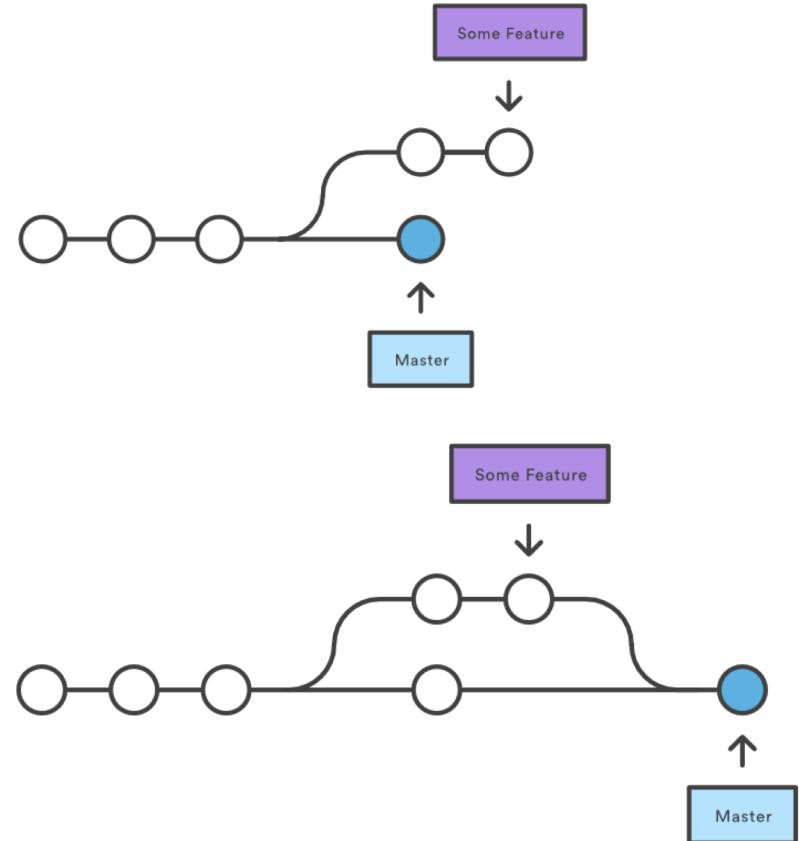
- Ưu điểm là ít gây xáo trộn cây lịch sử
- Khó nhận diện sự kiện merge khi xem lại cây lịch sử
- Thường chỉ dùng để merge nhánh local với nhánh remote



Merge No Fast-forward



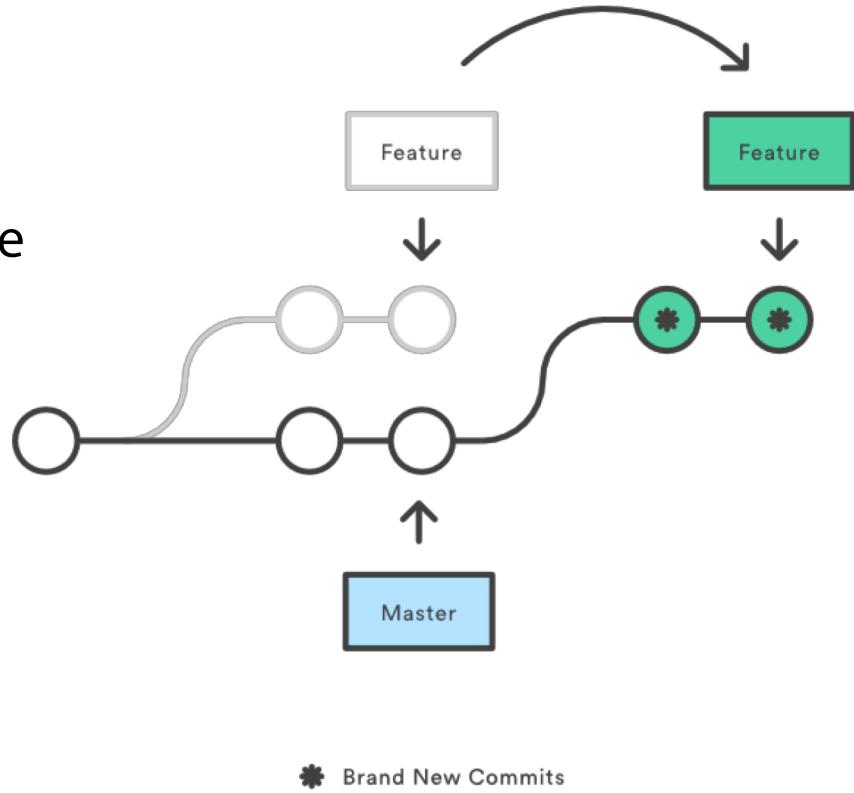
- Sử dụng một commit merge (có 2 commit cha) để đạt mục tiêu
- Luôn ưu tiên sử dụng để merge nhánh trong các git flows.



Rebase



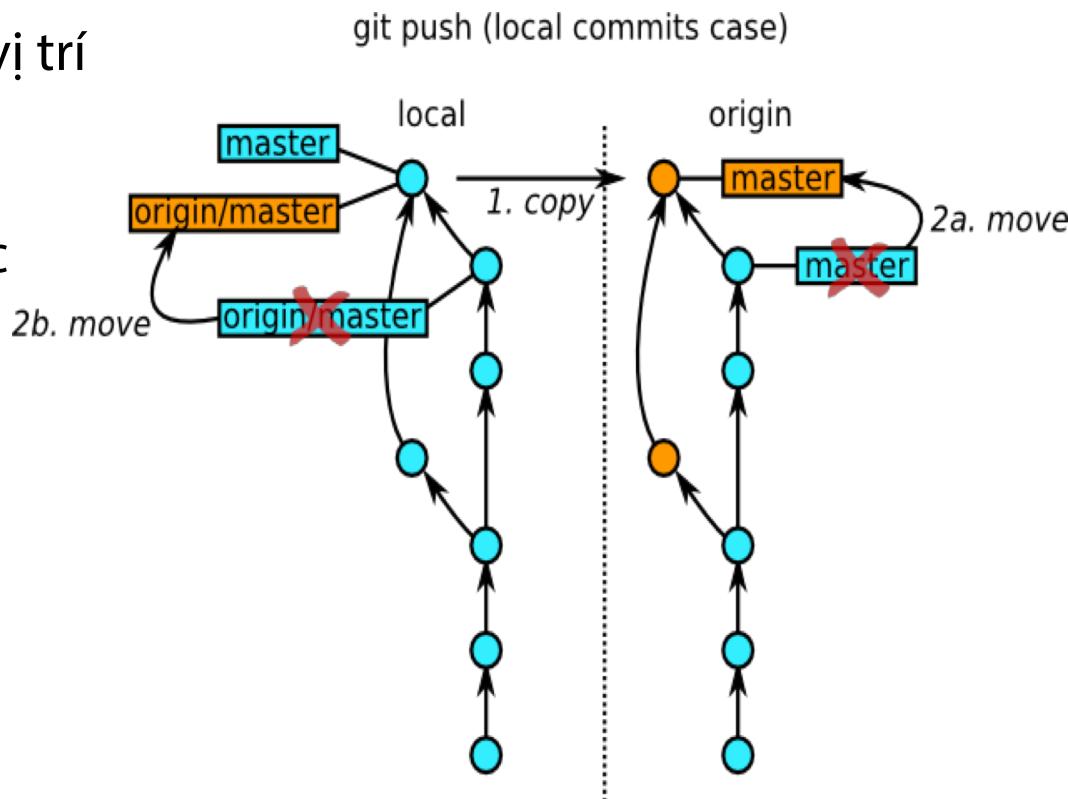
- Đạt được mục tiêu merge nhưng làm thay đổi lịch sử
- Nên sử dụng giới hạn trong nhánh chưa push hay để merge nhánh local với nhánh remote



Remote Repository



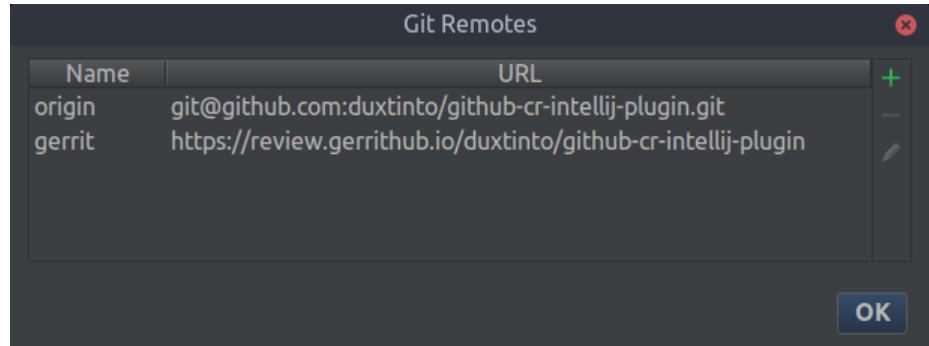
- Một nhánh của local repository được đặt tại một vị trí khác
- Giữa các repository xảy ra sự kiện đồng bộ (commit và các head)



Remote Tracking



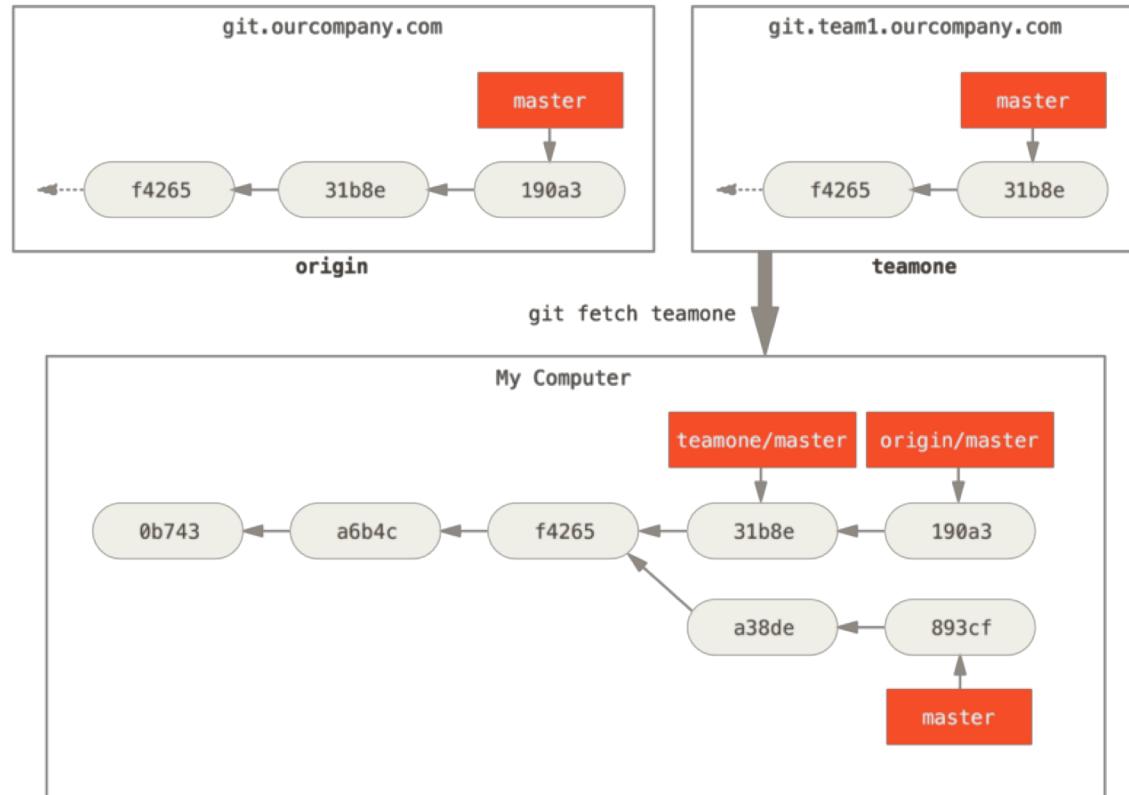
- Trạng thái của một remote được lưu dấu tại local, hành động này được gọi là remote tracking
- Kết quả tracking được lưu vào các “remote”
- Các remote được dùng để chỉ định mục tiêu đồng bộ



Nhánh remote



- Lưu dấu trạng thái các nhánh remote tại lần cuối tracking
- Được đặt tên mặc định theo quy ước “tên-remote/tên-nhánh”
- Được cập nhật trạng thái bằng thao tác fetch

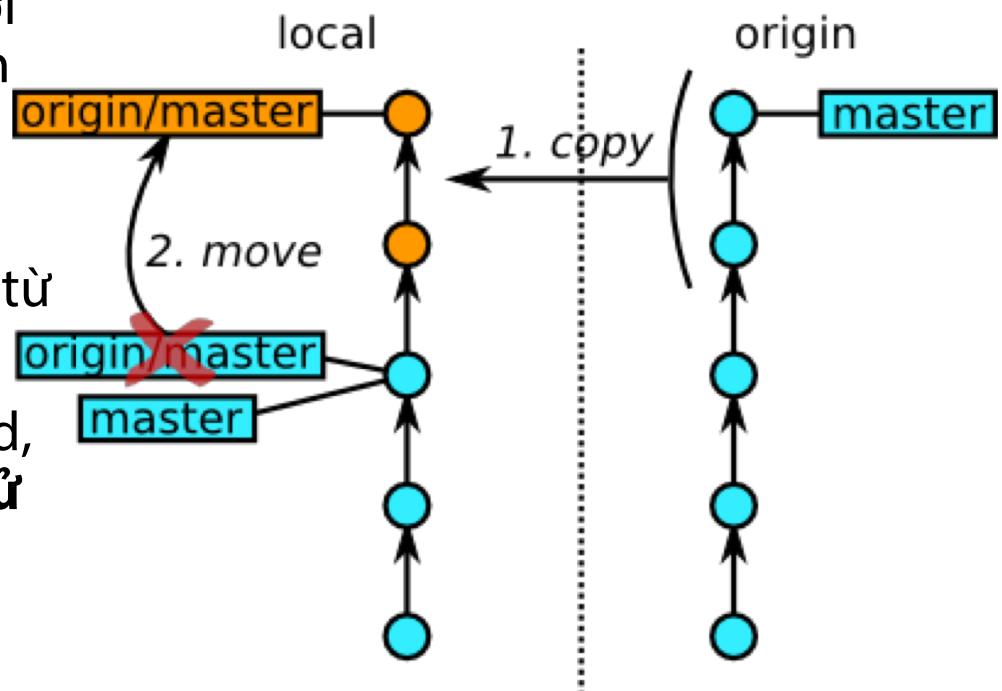


Update heads



- Mọi thao tác đồng bộ đều quy về cập nhật các head cùng với tải về toàn bộ các commit cần cho head đó
- Chủ động: *fetch*
- Bị động: khi nhận một “*push*” từ repo khác
- Trong mọi cách cập nhật head, ưu tiên **không làm cây lịch sử của remote bị xáo trộn**.

git fetch (fast forward case)

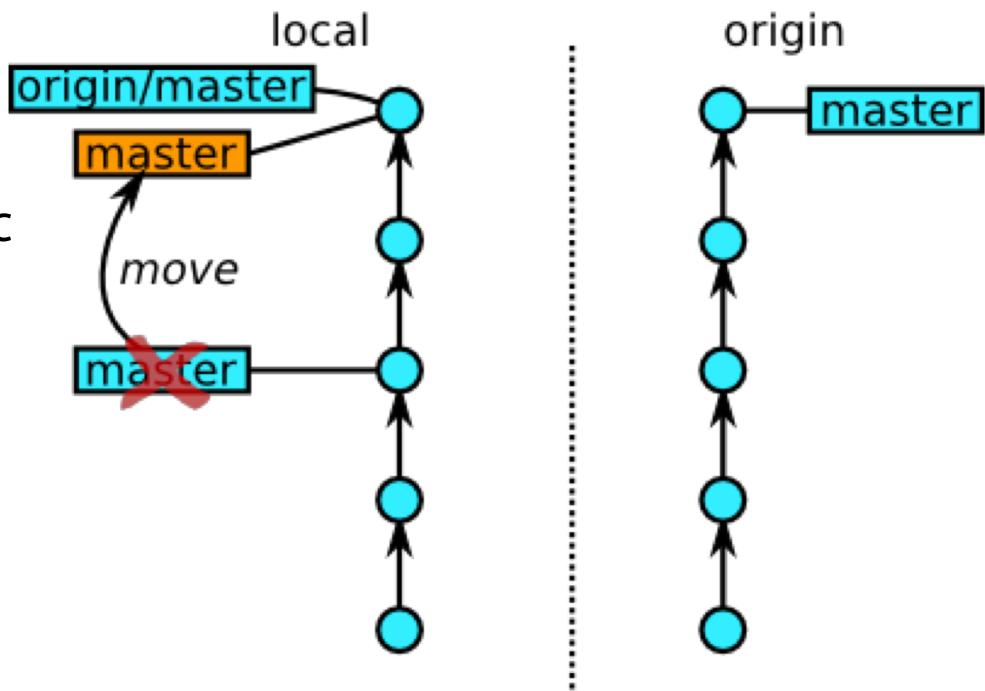


Update head: fast forward



- Thường được thực hiện đơn giản bằng thao tác pull
- Pull = Fetch + Merge
- Nếu có sự sê nhánh, thao tác pull sẽ không hoàn thành được

git merge (fast forward case)

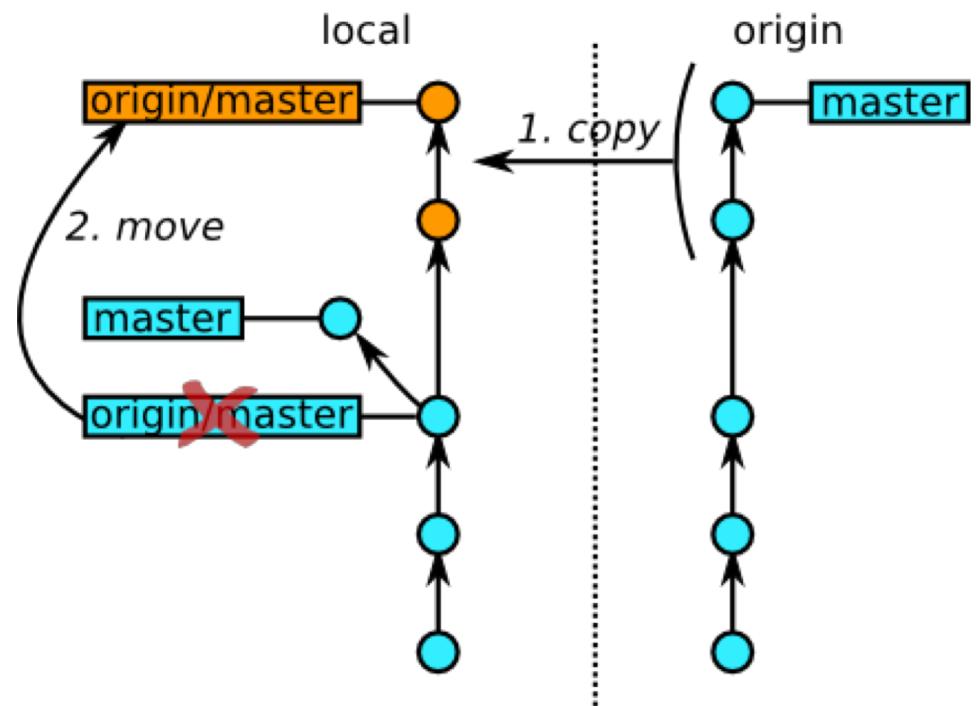


Update heads: Merge



- Xảy ra khi có sự rẽ nhánh, không fast forward được

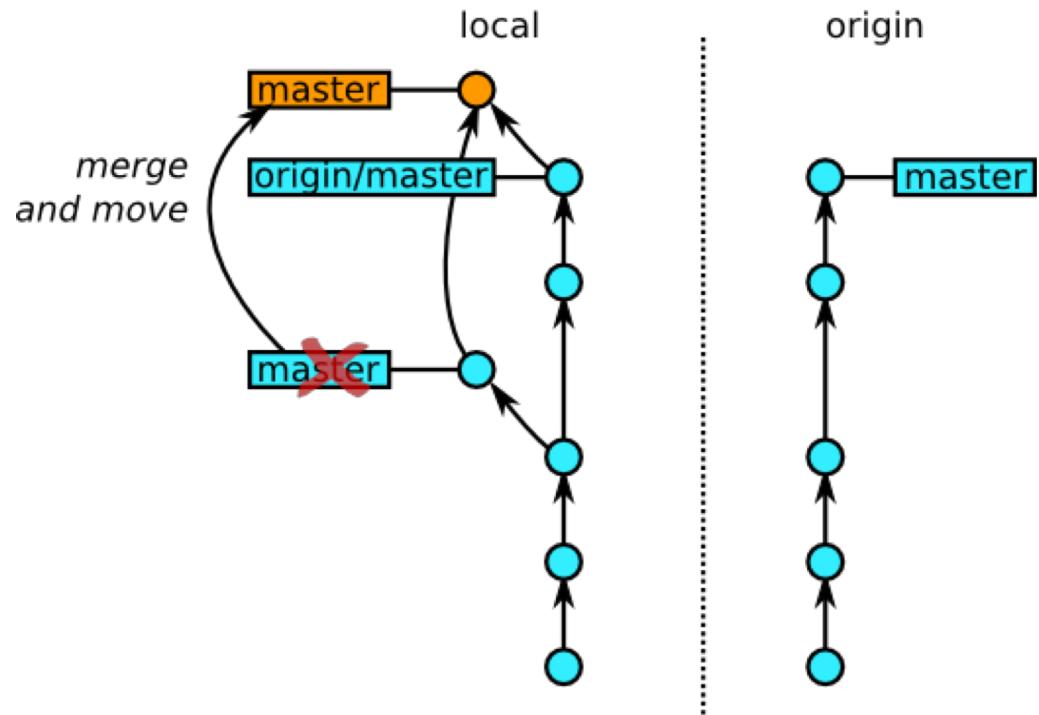
git fetch (local commits case)



Update heads: merge (2)

- ... lúc này cần thực hiện thao tác merge

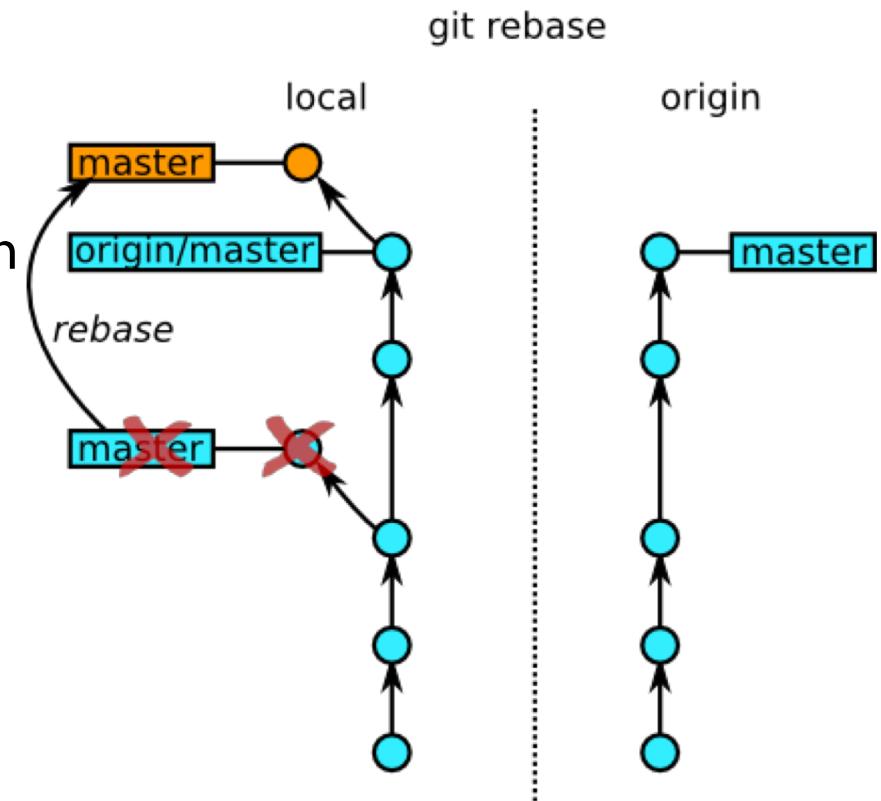
git merge (local commits case)



Update ref: rebase



- Ưu tiên sử dụng khi update cùng một nhánh
- Không khuyến khích sử dụng khi update chéo giữa các nhánh khác nhau (sẽ làm cho không nhận biết được hiện tượng merge)

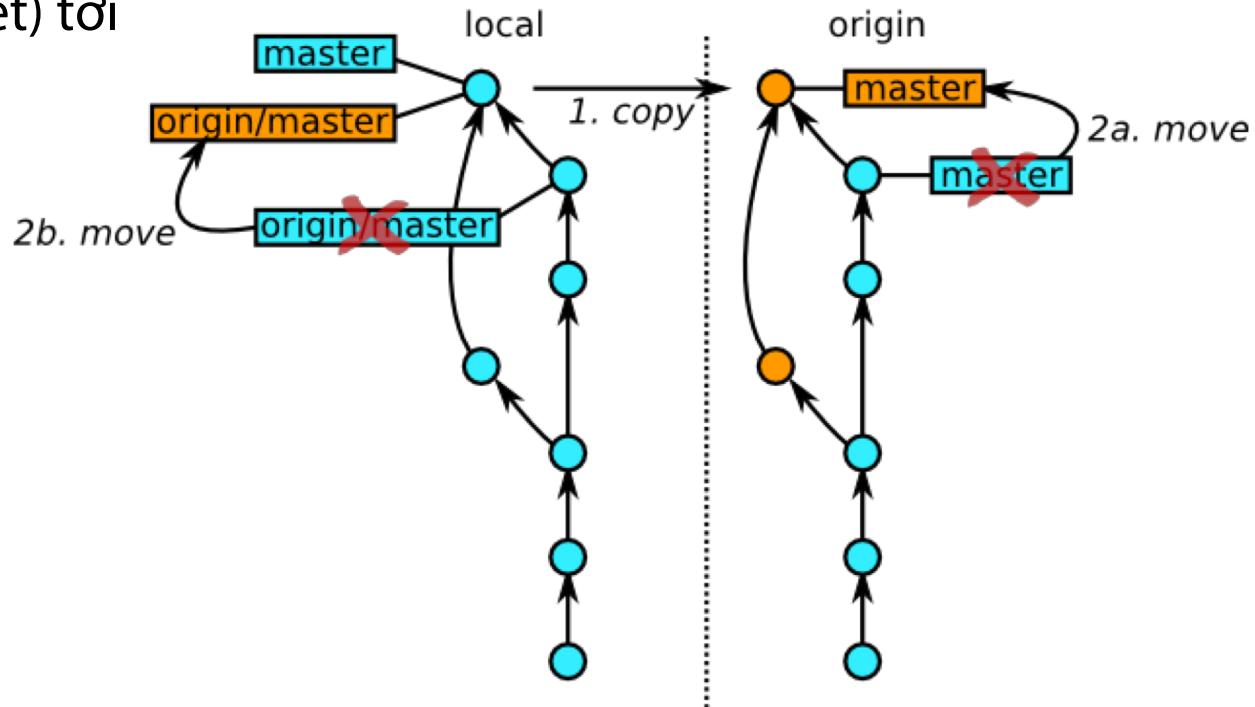


Push



- Là thao tác đồng bộ head (và các commit cần thiết) tới remote.

git push (local commits case)



Merge Request



- Là một ngỏ lời “ai đó” merge một nhánh vào một nhánh khác
- Tạo khoảng thời gian ngừng trước khi merge để có thể so sánh mã giữa hai nhánh
- Tạo cơ hội để team có thể review/thảo luận
- Giảm khả năng mã được merge khi chưa sẵn sàng
- Không phải là chức năng nguyên thủy của git, mà là chức năng của dịch vụ git (Github, Gitlab, Bitbucket...)
- Trong dịch vụ Github được gọi là Pull Request

Tạo Pull Request



A screenshot of a GitHub repository page. At the top, there's a header with a profile picture, the repository name '24,040 COMMITS', and a 'New pull request' button. Below the header, there's a dropdown menu for 'Branch: master' and some user information.

A screenshot of a context menu on a GitHub page. The menu has a green 'Create Pull Request' button at the top. Below it, there are two options: 'Create Pull Request' (marked with a checkmark) and 'Create Draft Pull Request'. The 'Create Draft Pull Request' option is described as 'Cannot be merged until marked ready for review'.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

A screenshot of a GitHub pull request creation interface. It shows two dropdown menus: 'base: master' and 'compare: update-readme-2'. To the right, a green checkmark indicates 'Able to merge. These branches can be automatically merged.'

<https://help.github.com/en/articles/creating-a-pull-request>



Tạo Pull Request (2)

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

The screenshot shows the GitHub interface for creating a pull request. At the top, there are dropdown menus for 'base: master' and 'compare: update-readme-2'. A green checkmark indicates 'Able to merge'. Below this, a yellow box says 'Before you submit a pull request'. On the left, there's an octocat icon and buttons for 'Add octocat', 'Write', and 'Preview'. A modal window titled 'Choose a head branch' is open, showing a dropdown with 'update' selected. Other options in the list include 'revert-4-updates-to-release-v1.2' (which is highlighted in blue), 'update-readme-1', 'update-readme-2', 'update-readme', 'updates-to-release-v1.2', and 'update'. To the right of the modal, there are sections for 'Reviewers' (empty), 'Assignees' (empty), and 'Labels' (empty).

The screenshot shows the GitHub repository dashboard for 'octo-org / octo-repo'. The repository is private. At the top, there are buttons for 'Code', 'Issues 5', 'Pull requests 24' (which is highlighted in blue), 'Security', 'Actions', 'Projects 8', 'Wiki', 'Insights', and 'Settings'. In the center, there's a list of pull requests. At the bottom, there's a search bar and a link to 'https://help.github.com/en/articles/creating-a-pull-request'.

<https://help.github.com/en/articles/creating-a-pull-request>

Review



```
115 115     lineElements.css 'background-color', ''
116 116
117 117     if lineRange.length is 1
118 -   $("\\#{anchorPrefix}LC#{lineRange[0]}").css 'background-color', "#ffc"
118 +   $("\\#{anchorPrefix}LC#{lineRange[0]}").css 'background-color', "#f8eec7"
119 119 
120 120     else if lineRange.length > 1
121 121         i = lineRange[0]
122 122         while i <= lineRange[1]
123 -   $("\\#{anchorPrefix}LC#{i}").css 'background-color', "#ffc"
123 +   $("\\#{anchorPrefix}LC#{i}").css 'background-color', "#f8eec7"
124 124         i++
125 125
126 126 # Highlight and scroll to the lines in the current location hash.
```



24  app/assets/javascripts/github/pages/diffs/linkable-line-number.coffee



@@ -9,3 +9,27 @@

<https://help.github.com/en/articles/commenting-on-a-pull-request>

Thảo luận



Add author section #8

Merged emilyistoofunky merged 1 commit into master from add-authors a minute ago

Conversation 0 · Commits 1 · Files changed 1

emilyistoofunky commented 5 minutes ago

I added a section where we can list project authors to the README file. @octo-org/octo-team, please review and let me know what you think!

Add author section 8ffc474

emilyistoofunky added the enhancement label 2 minutes ago

octocat commented 2 minutes ago

Looks great to me! 🎉

emilyistoofunky merged commit 7453107 into master a minute ago

<https://help.github.com/en/articles/commenting-on-a-pull-request>

[Revert](#)

Merge Pull Request



This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or [view command line instructions](#).

This branch has no conflicts with the base branch
Merging can be performed automatically.

Squash and merge

You can also [open this in GitHub Desktop](#) or [view command line instructions](#).

Create a merge commit
All commits from this branch will be added to the base branch via a merge commit.

Squash and merge
The 4 commits from this branch will be combined into one commit in the base branch.

Rebase and merge
The 4 commits from this branch will be rebased and added to the base branch.

<https://help.github.com/en/articles/merging-a-pull-request>

Đóng Pull Request



Leave a comment

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.

 **ProTip** Add comments to specific lines under Files changed.

[Close pull request](#) [Comment](#)

<https://help.github.com/en/articles/closing-a-pull-request>



Xóa nhánh đã merge

  bernars merged commit 5d60df4 into master from cleanup-ui 2 minutes ago [Revert](#)

  bernars closed the pull request 2 minutes ago

Pull request successfully merged and closed

You're all set—the cleanup-ui branch can be safely deleted.

 [Delete branch](#)



 octocat deleted the main-41812-rad-dir-change branch on Jan 20

[Restore branch](#)

<https://help.github.com/en/articles/deleting-and-restoring-branches-in-a-pull-request>

Git flows (danh từ chung)

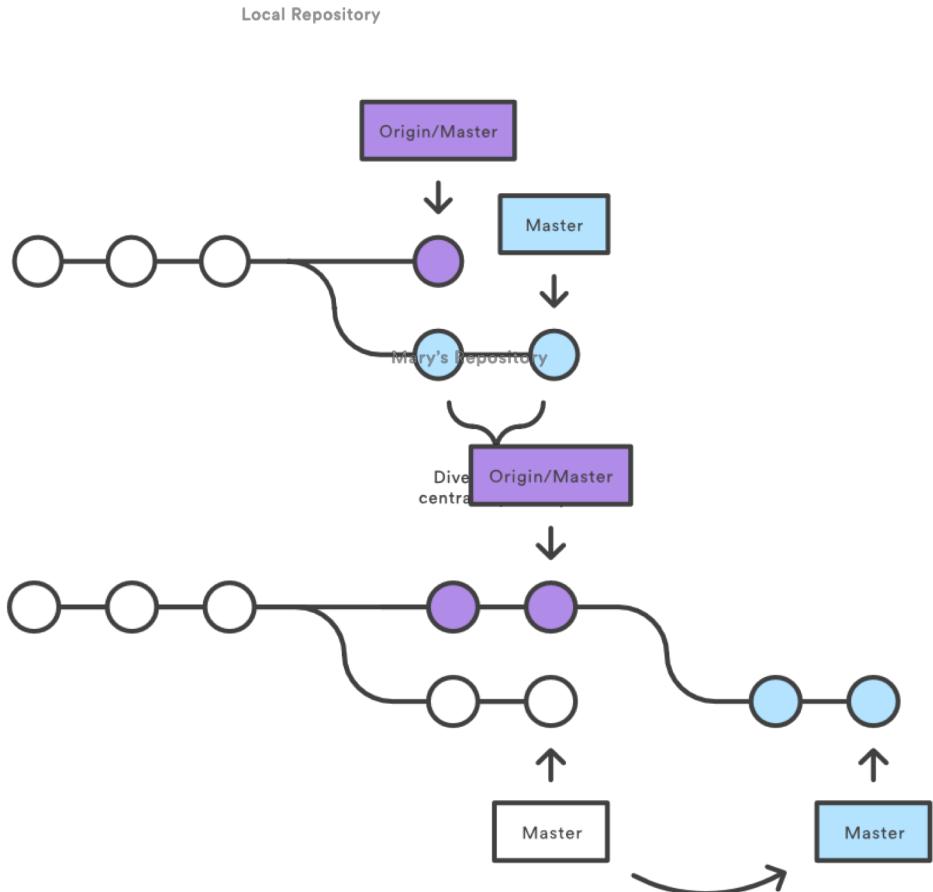
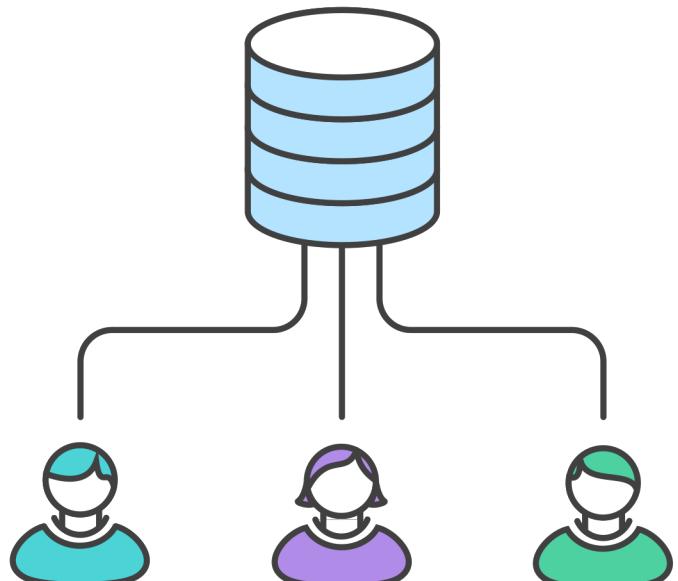


- Là những chiến thuật tạo nhánh và merge nhánh được thiết kế để làm việc cộng tác hiệu quả

Centralized Workflow



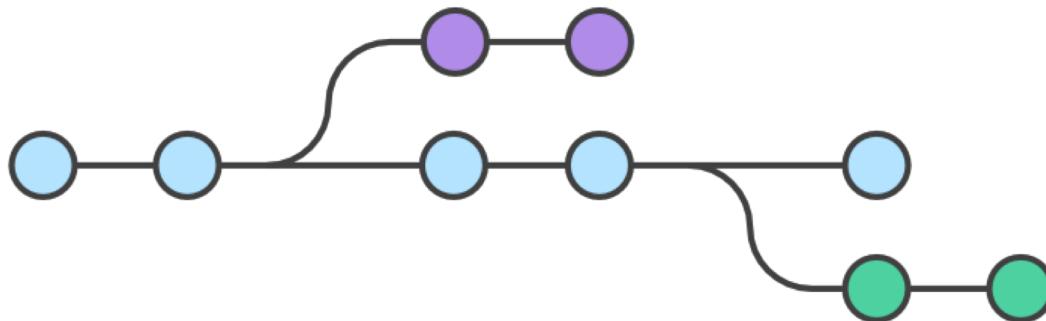
- Một repository trung tâm/một nhánh trung tâm



Feature Branch Workflow



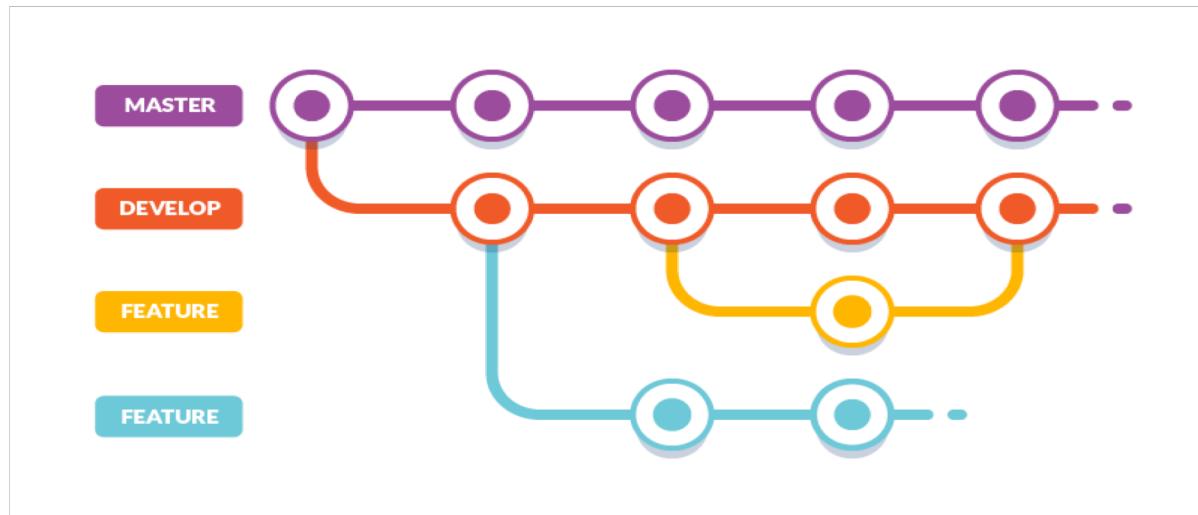
- Duy trì một nhánh trung tâm, các chức năng được phát triển trên các nhánh feature trước khi được merge.



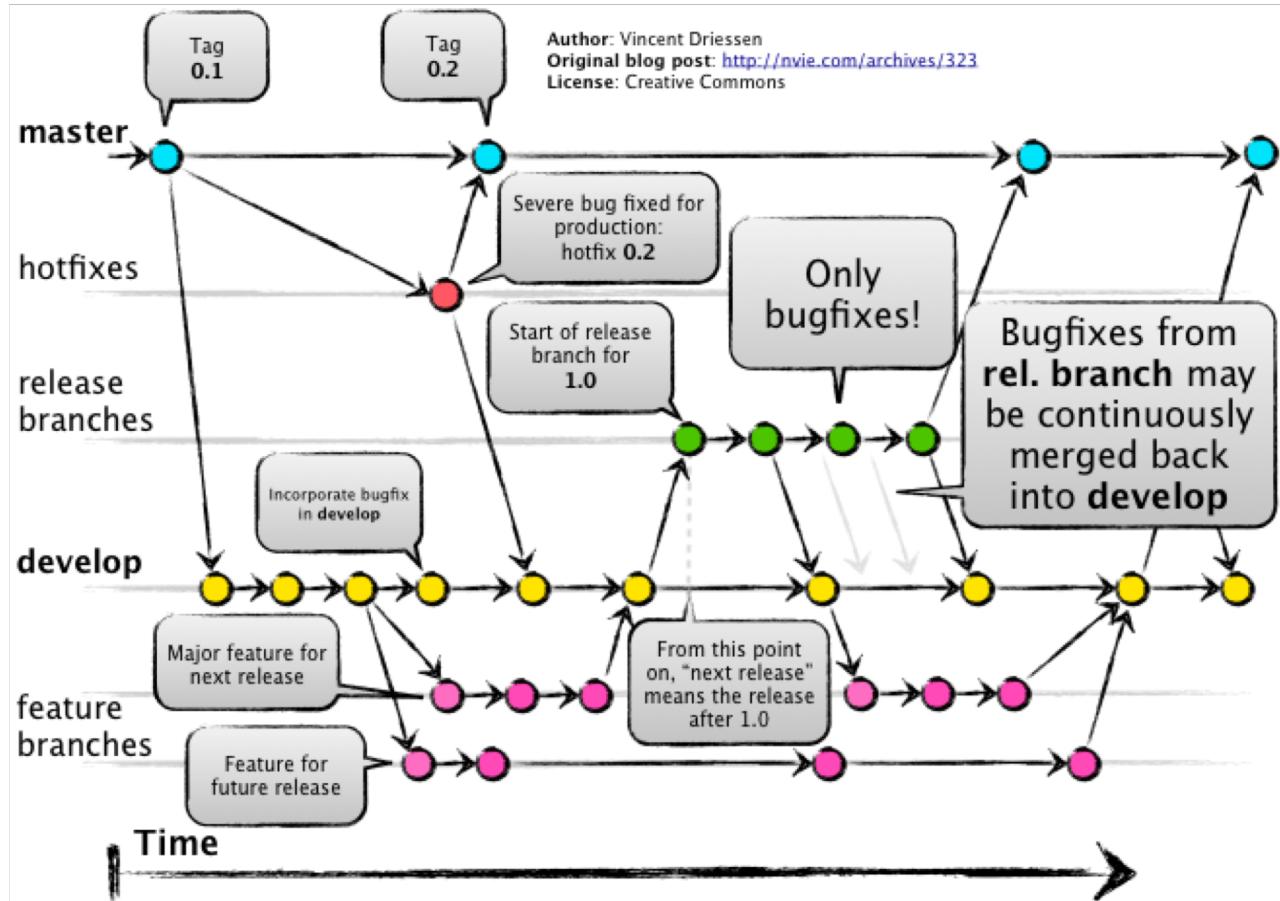
Gitflow fit flow



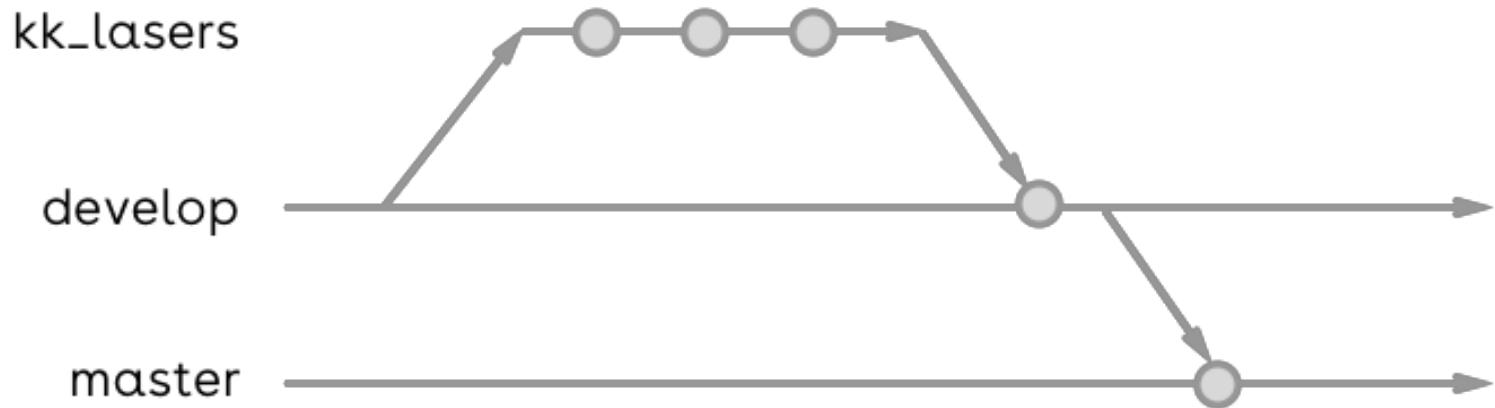
- Duy trì một nhánh **dev** để đệm giữa các nhánh feature và nhánh master
- Mã trên nhánh dev thể hiện sản phẩm đang kiểm thử
- Mã trên nhánh master thể hiện sản phẩm đang chạy



Gitflow git flow (2)



Simpliced Gitflow





Thực hành

- <https://learngitbranching.js.org/>
- <https://help.github.com/en/articles/about-pull-requests#draft-pull-requests>