Nguyễn Thị Yến

**Báo cáo Git**

**1. Repository**

* Là nơi ghi lại trạng thái của thư mục và file.
* Có 2 loại:
* Remote repository: là repository để chia sẻ giữa nhiều người, bố trí trên serve.
* Local repository: là repository bố trí trên máy tính cá nhân, cho một người sử dụng.
* Tạo local repo: 2 cách
* Tạo mới
* Sao chép remove repo rồi tạo

**2. Working Tree, Index**

* những thư mục đc đặt tr sự quản lý của git mà mội người đang thực hiện công việc trong thực tế gọi là working tree.
* giữa repo và working tree tồn tại 1 nơi là Index, nơi để chuẩn bị cho việc commit lên repo.

**3. Commit**

* Ghi lại việc thêm, thay đổi file hay thư mục vào repo.
* Khi thực hiện commit, trong repo sẽ tạo ra commit đã ghi lại sự khác biệt từ trạng thái đã commit lần trước đến trạng thái hiện tại.
* commit làm việc: git commit –m “nội dung commit”
* Trước khi tạo commit -> kiểm tra trạng thái: git status -> thực hiện thay đổi muốn đưa vào commit: git add [file] (nếu k thêm bất cứ file nào thì lệnh git commit sẽ k hoạt động).

git add –p: xem các thay đổi

thực hiện lệnh commit-> cập nhật lên remove repo: git push -> duyệt và kiểm tra sự phát triển của các tệp: git log

* Cách sử dụng:

+ git commit -m “nd commit”

+ git commit -am “nd commit”: tự động các thay đổi vs file đã commit trước đó

+ git commit --amend: thay thế commit gần đây bằng commit mới.

* Git revert: thay đổi lịch sử với Git
* Git resert: xóa commit (--soft, --mixed, --hard)
* Mixed: đưa trạng thái của index đã thay đổi về ban đầu (mặc định)
* Hard: xem như hoàn toàn k có commit gần đây
* Soft: xem như chỉ commit là k có
* Git reflog: nhật ký của mọi commit mà HEAD trỏ đến.

**4. Push**

* Chia sẻ lịch sử thay đổi của local repo bằng romote repo. Update lịch sử trong local repo lên remote repo.
* cách sử dụng:

+ git push –f: buộc đẩy nếu k sẽ bị chặn

+ git push –u origin [tên nhánh]: hữu ích khi đẩy 1 nhánh mới

+ git push --all: push tất cả các nhánh

+ git push --tags: đẩy các thẻ chưa có trong remove repo.

**5. Pull**

* Cập nhật local repo từ remove repo (cập nhật 1 số phần của local repo với các thay đổi từ remove repo)
* là sự kết hợp git fetch + git merge, khi clone 1 repo-> clone nhánh đang hoạt động (master)

+ git fetch: cập nhật các nhánh remote branch

+ git merge: hợp nhất nhánh hiện tại của bạn với bất cứ commit mới nào trên remote branch.

* Cách sử dụng:

+ git pull: cập nhật nhánh làm việc tại local với các commit và update từ remote repo.

+ git pull --rebase: cập nhật nhưng viết lại lịch sử để mọi commit local xảy ra sau commit remote, tránh merge commit.

+ git pull --force

+ git pull --all: fetch tất cả các remote

**6. Merge**

* Hợp nhất các nhánh trong Git, hợp nhất chuỗi commit thành 1 lịch sử thống nhất.
* Hợp nhất bằng git merge

+ Thực thi git status -> git checkout: chuyển sang nhánh nhận hợp nhất

+ git fetch: tìm nạp các remote commit

+ git pull: cập nhật nhánh chính

+ git merge [tên nhánh gửi]: hợp nhất các thay đổi vào nhánh nhận

-> merge fast- forward

**7. Conflict**

* Xung đột xảy ra khi 2 hoặc nhiều người cùng thay đổi 1 tệp, các xung đột xuất hiện tại local hoặc remote.
* Giải quyết conflict bằng merge: chỉnh sửa nội dung trên file mà nó dẫn tới xung đột, sau đó thực hiện add, commit cho file đó.

-> non fast-forward merge

**8. Rebase**

- Sử dụng để nhập 1 nhánh đã gần hoàn thiện vào nhánh gốc.

- Rebasing là quá trình chuyển đổi gốc của 1 nhánh từ commit này tới commit khác.

- Tác dụng: lưu giữ lịch sử làm việc của dự án 1 cách tuyến tính.

+ git rebase [tên nhánh]

**9. Pull request**

- Là chức năng cho phép lập trình viên thông báo với những người trong nhóm rằng họ đã hoàn thành công việc của mình.

- Các thành viên còn lại của nhóm sẽ bắt đầu xem xét, thảo luận, chú thích hay thay đổi mã sao cho phù hợp.

- Sau khi hoàn tất, người quản lý sẽ hợp nhất các tính năng vào repo chính và đóng pull request của lập trình viên lại.