1. Tạo 1 file 0.txt điền nội dung và gõ git add . để add file, git commit C0, lúc này nếu ta push lên giả sử là master thì nhánh master sẽ có commit là C0
2. Tạo 1 file 1.txt điền nội dung và gõ git add . để add file, git commit C1, lúc này nếu ta push lên thì master sẽ có 2 commit là C0, C1
3. Lúc này sơ đồ như sau

C0--🡪C1-🡪Master--🡪HEAD

1. Ta tạo 1 nhánh mới với câu lệnh git branch nhanhmoi1 thì nhanhmoi1 sẽ kế thừa các commit từ master trước đó là C0, C1 . Điểm commit C1 là điểm chung của 2 nhánh master và nhanhmoi1 và sơ đồ như sau:
2. C0--🡪C1-🡪Master--🡪HEAD

Nhanhmoi1

1. Lúc này con trỏ Head vẫn đang trỏ về nhánh master, để nhảy sang nhánh nhanhmoi1 ta sử dụng câu lệnh git checkout nhanhmoi1
2. Lúc này khi ta push lên github git push --set-upstream origin nhanhmoi1 thì ta có nhánh mới nhanhmoi1 được tạo ra trên repository ở github và có các commit và nội dung giống hệt với master
3. Lúc này ở local trên nhanhmoi1 ta tạo file 3.txt và ta git add . sau đó git commit -m “C3” thì ta sẽ có commit C3 và file 3.txt ở trên nhanhmoi1 mà không có ở trên nhánh master. Tương tự nếu ở bước này mà ta lại checkout sang nhánh master sau đó git add . và git commit -m”C3” thì ta sẽ có commit C3 và file 3.txt ở trên nhánh master mà không có trên nhánh nhanhmoi1
4. Sau khi ta commit C3 và push lên nhanhmoi1 thì trên repository github sẽ thông báo có sự khác nhau giữa các nhánh. Compare and Pulll (so sánh và kéo về bản mới nhất nếu muốn)
5. Sơ đồ lúc này như sau

HEAD

C0--🡪C1-🡪Master

CommitC3

Nhanhmoi1

Trên nhanhmoi1 ta sửa dổi nội dung file 1.txt sau đó git add . và git commit-m “C4” thì sơ đồ sẽ như sau C0--🡪C1-🡪Master

HEAD

CommitC4

Nhanhmoi1

CommitC3

1. Đang từ nhanhmoi1 ta checkout sang nhánh master ở nhánh master ta chỉnh sửa nội dung file 0.txt và them mới file 3.txt và git add . và git commit “Commit C5”
2. Khi đó ta có sơ đồ nhu sau
3. C0--🡪C1-🡪Master

HEAD

CommitC5

CommitC4

Nhanhmoi1

CommitC3

1. Tạo một nhánh mới sualoigap đứng từ nhánh nhanhmoi1, lúc này nhánh sualoigap sẽ kế thừa các commit của nhanhmoi1 là C0, C1, C3, C4. Nếu đứng từ nhánh master tạo nhánh sualoigap thì nhánh sualoigap sẽ kế các commit C0, C1, C5 từ master
2. C0--🡪C1-🡪Master

HEAD

CommitC5

CommitC4

Nhanhmoi1

CommitC3

Sua loi gap

16 .Bây giờ trên nhánh sualoigap ta chỉnh sửa và git add . , git commit C6 thì không ảnh hưởng gì đến nhánh master hay nhánh nhanhmoi1

Sơ đồ như sau

1. C0--🡪C1-🡪Master

HEAD

CommitC5

CommitC4

Nhanhmoi1

CommitC3

Sua loi gap

CommitC6

1. Sau khi đã chỉnh sửa xong trong nhánh sửa lỗi gập, ta cần merge nhánh sualoigap về nhánh chính ta sử dụng git checkout master để về nhánh master sau đó đứng ở master ta nhập tên nhánh cần merge: git merge sualoigap (khi đó nó sẽ so sánh sự khác nhau giữa 2 commit cuối của 2 nhánh mà có cho phép merge ngay hay khong)
2. Sau khi đã gộp nhánh sửa lỗi gấp vào nhánh master thì có thể ta không cần đến nhánh sualoigap nua thì ta có thể xóa nhánh đó đi bằng cách : git branch -D sualoigap. Việc xóa này chỉ xóa nhánh ở máy tính (cục bộ) trong khi nhánh đó ở git remote vẫn chưa được xóa. Để tiếp tục xóa nhánh đó ở remote ta dung câu lệnh: ***git push origin --delete tennhanh***
3. Bây giờ ta muốn tiếp tục merge nhánh alpha và nhánh master. Do có việc sửa đổi đồng thời ở alpha và master nên sẽ dẫn đến xung đột(conflict) nên git không biết sử dụng cái nào chọn cái nào. Lúc này ta git status để xem file nào xung đột và tiến hành lựa chọn chỉnh sửa nào