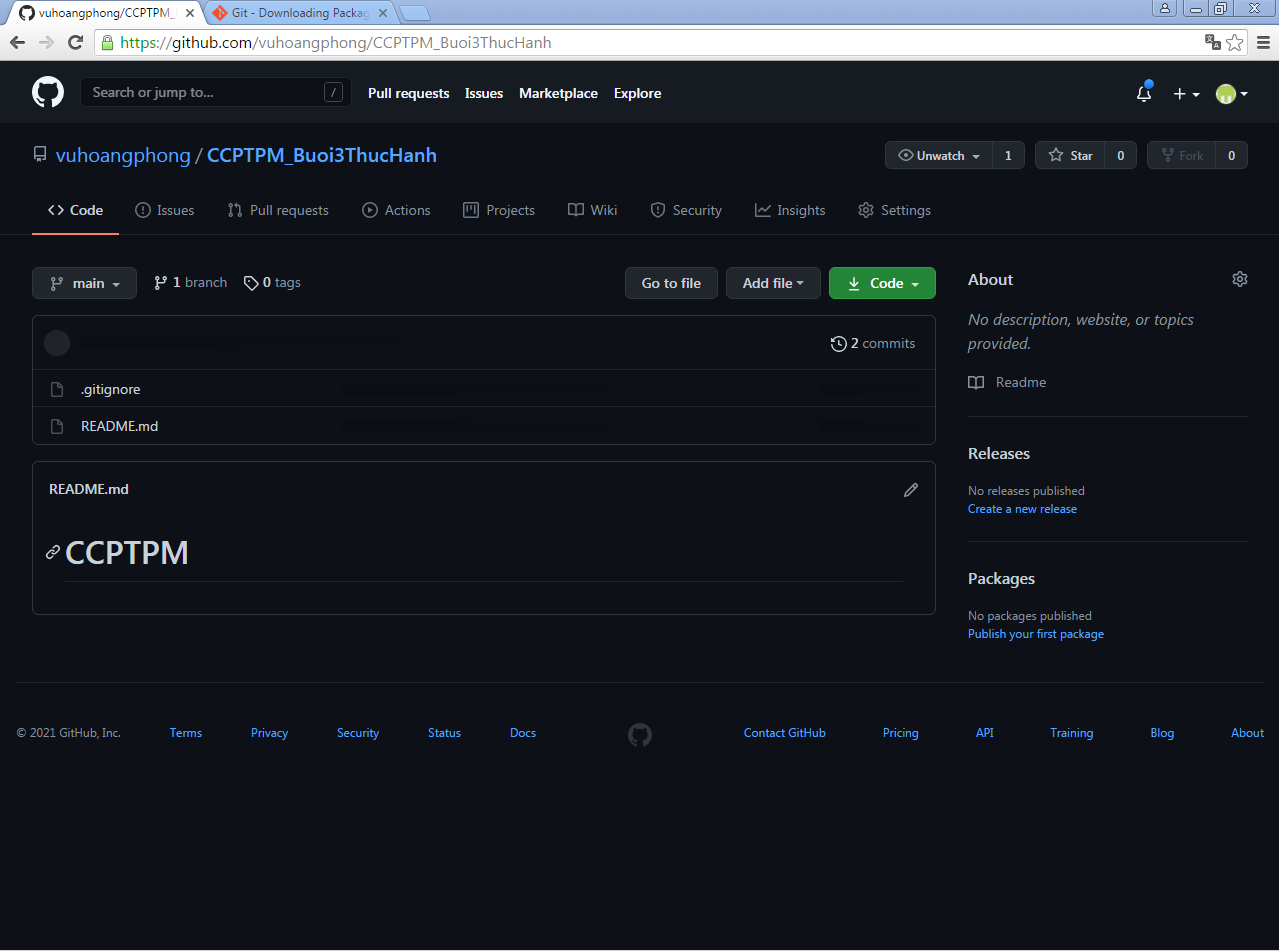
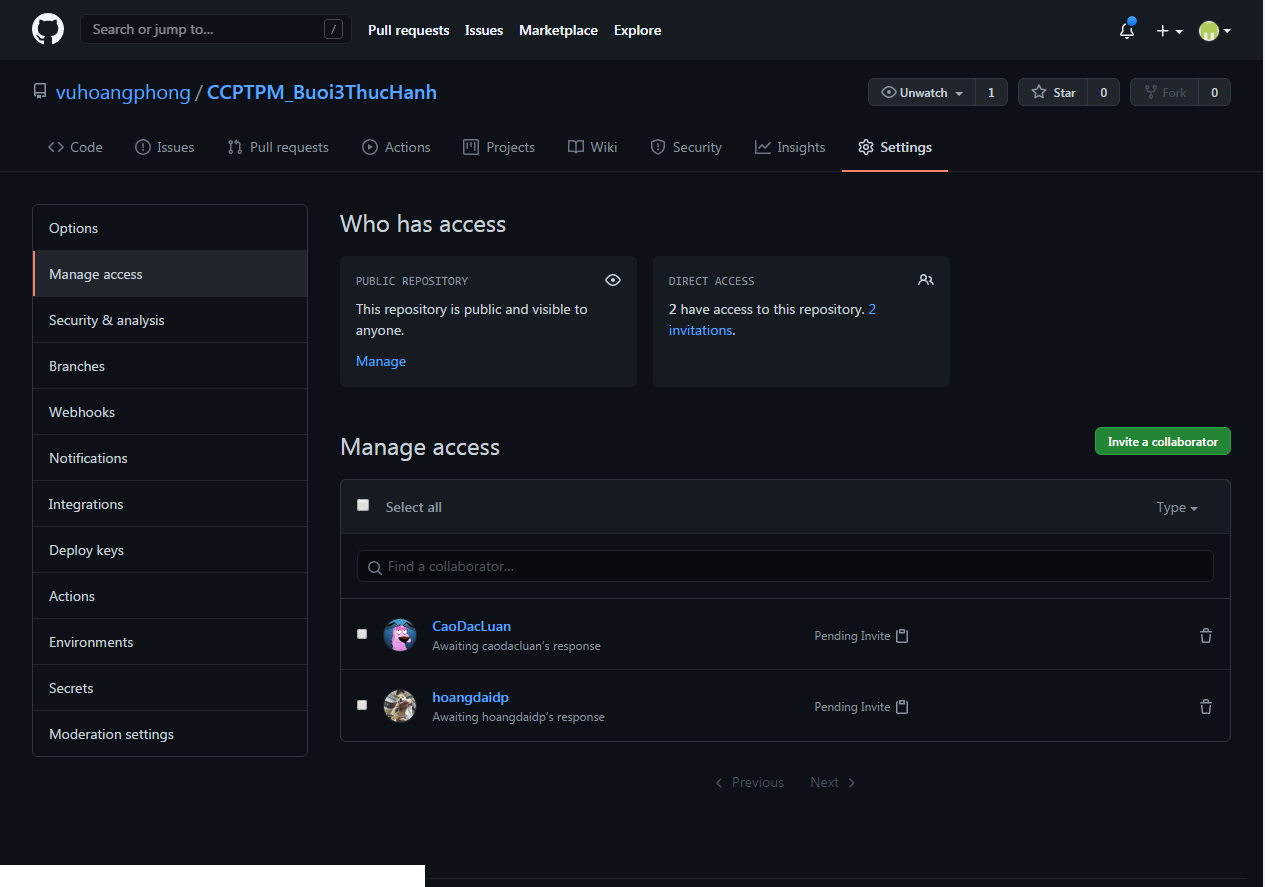
Họ Tên: Vũ Hoàng Phong – 1711060251

Cao Đắc Luận – 1611040196

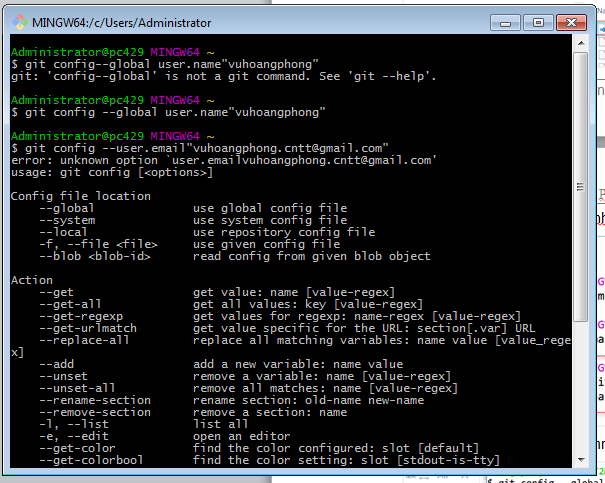
TẠO TÀI KHOẢN, TẠO REPO



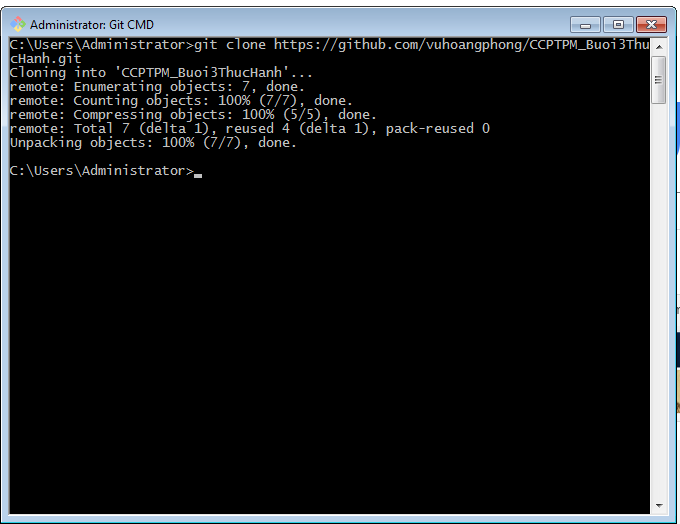
THÊM THÀNH VIÊN, THÊM QUYỀN



ĐỊNH DANH TÀI KHOẢN TẠI MÁY TÍNH ĐANG SỬ DUNG (GIT CONFIG)



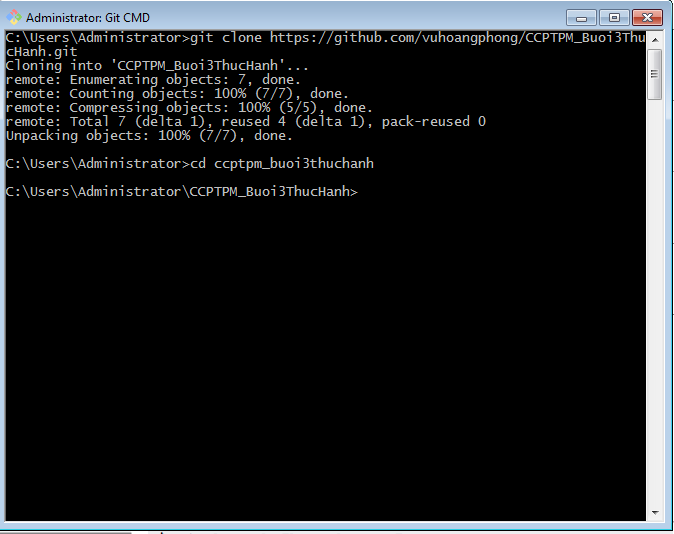
INIT DỰ ÁN, CLONE DỰ ÁN



Thực hiện các câu lệnh git

$ git init [project-name]

$ git clone [url]



$ git status

Lists all new or modiﬁed ﬁles to be commited

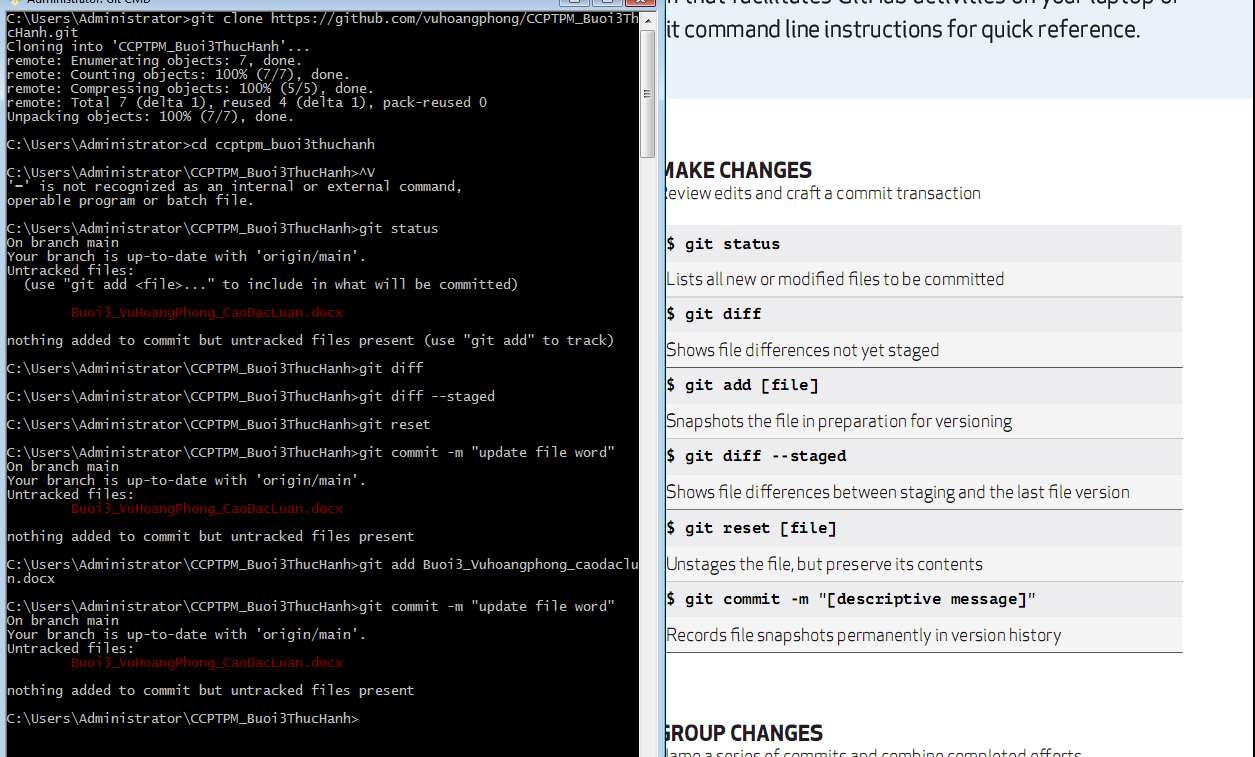
$ git add [file]

$ git reset [file]

$ git diff

$ git diff --staged

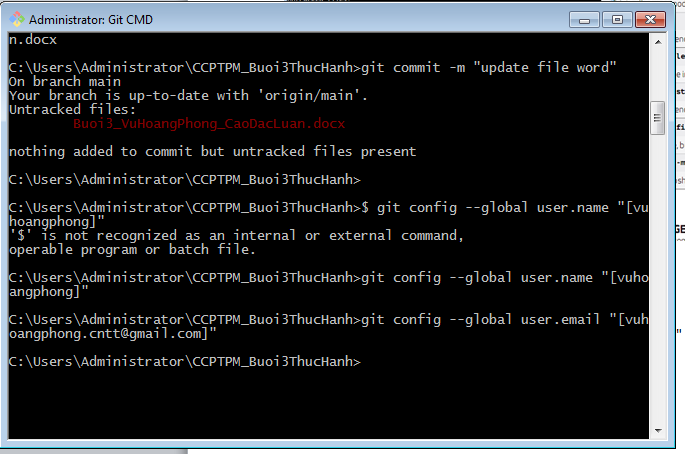
$ git commit -m "[descriptive message]"



$ git config --global user.name "[vuhoangphong]"

$ git config --global user.email "[vuhoangphong.cntt@gmail.com]"

$ git config --global color.ui auto



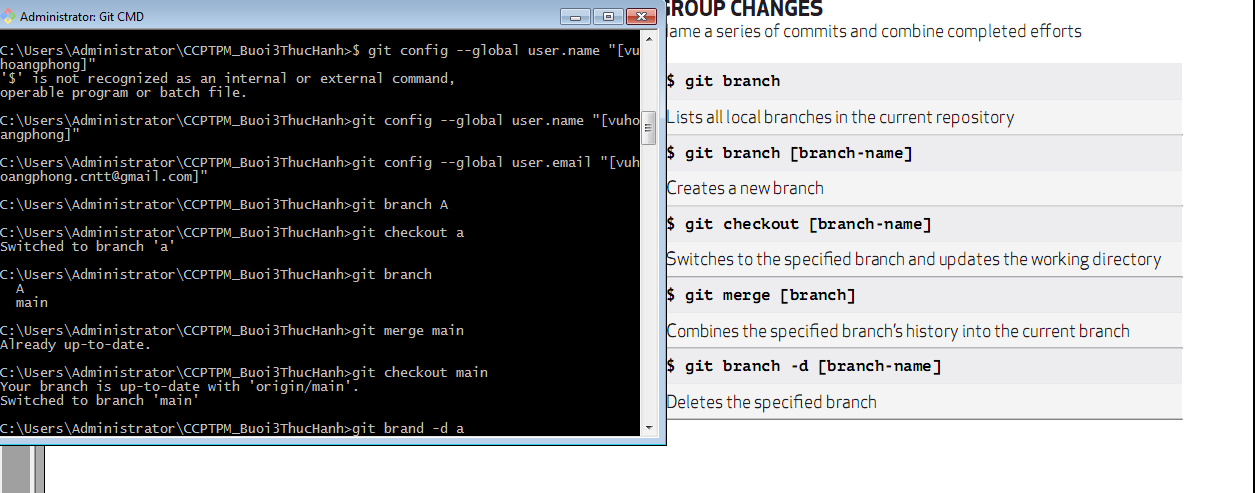
$ git branch

$ git branch [branch-name]

$ git checkout [branch-name]

$ git merge [branch]

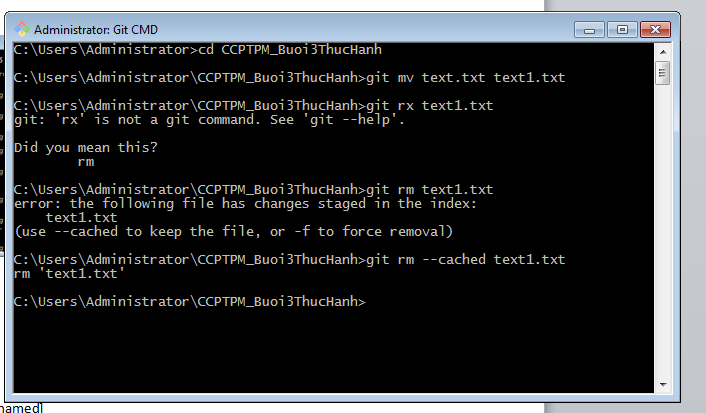
$ git branch -d [branch-name]



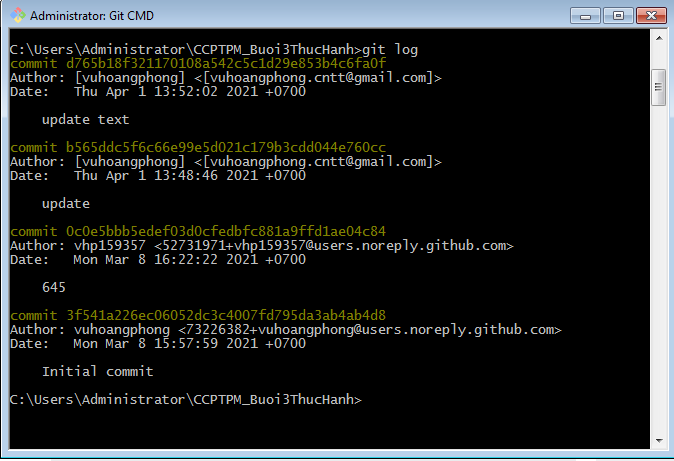
$ git rm [file]

$ git rm --cached [file]

$ git mv [file-original] [file-renamed]

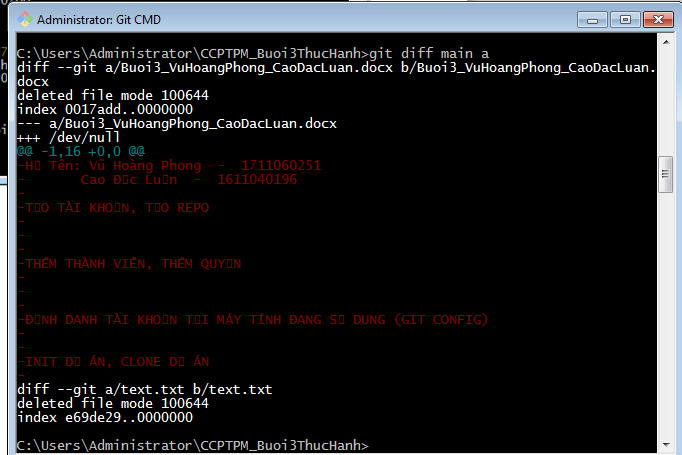


$ git log

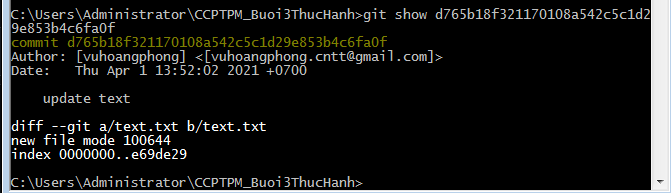


$ git log --follow [file]

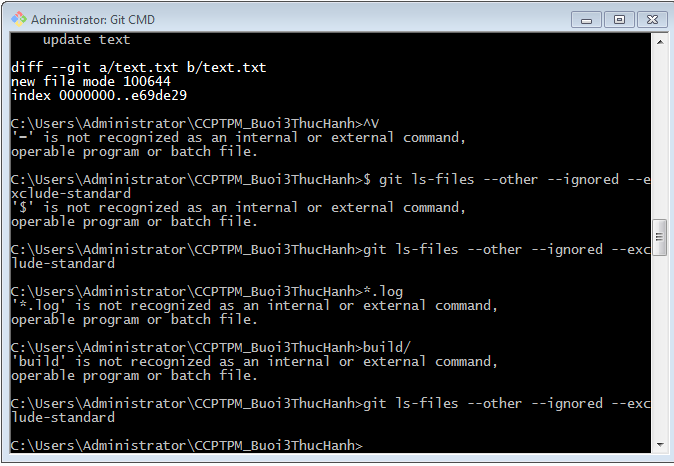
$ git diff [first-branch]...[second-branch]



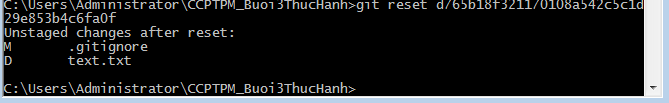
$ git show [commit]



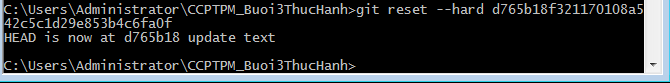
$ git ls-files --other --ignored --exclude-standard



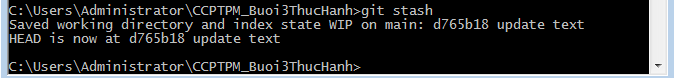
$ git reset [commit]



$ git reset --hard [commit]



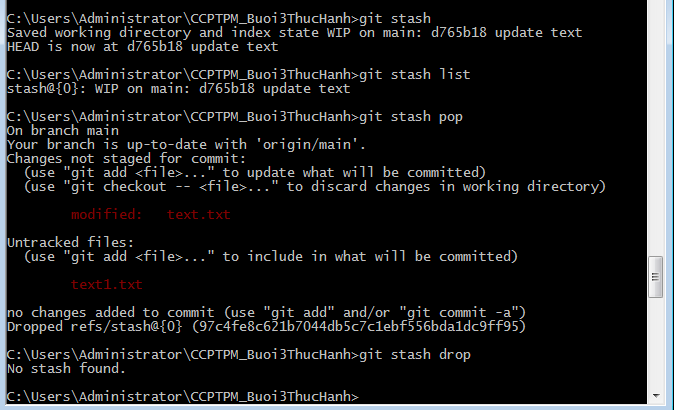
$ git stash



$ git stash list

$ git stash pop

$ git stash drop

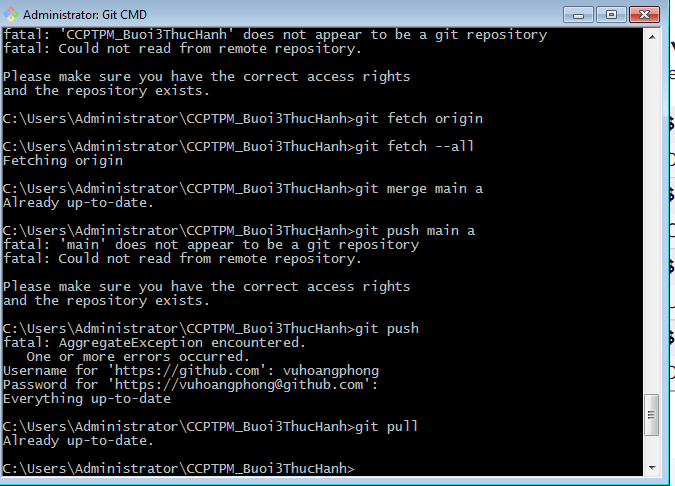


$ git fetch [bookmark]

$ git merge [bookmark]/[branch]

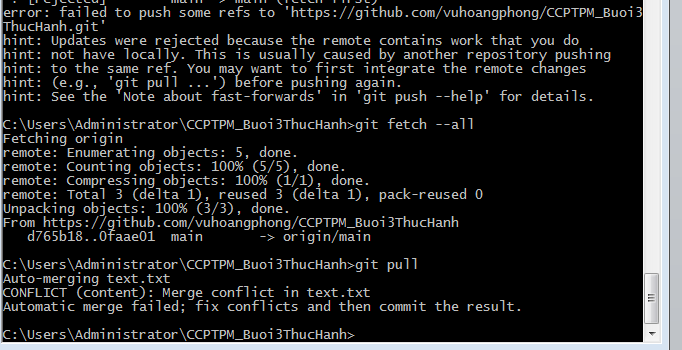
$ git push [alias] [branch]

$ git pull

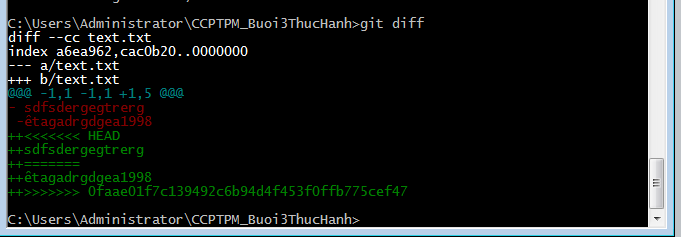


TẠO XUNG ĐỘT MÃ NGUỒN VÀ CÀI ĐẶT CÔNG CỤ P4MERGE ĐỂ HỖ TRỢ GIẢI QUYẾT XUNG ĐỘT

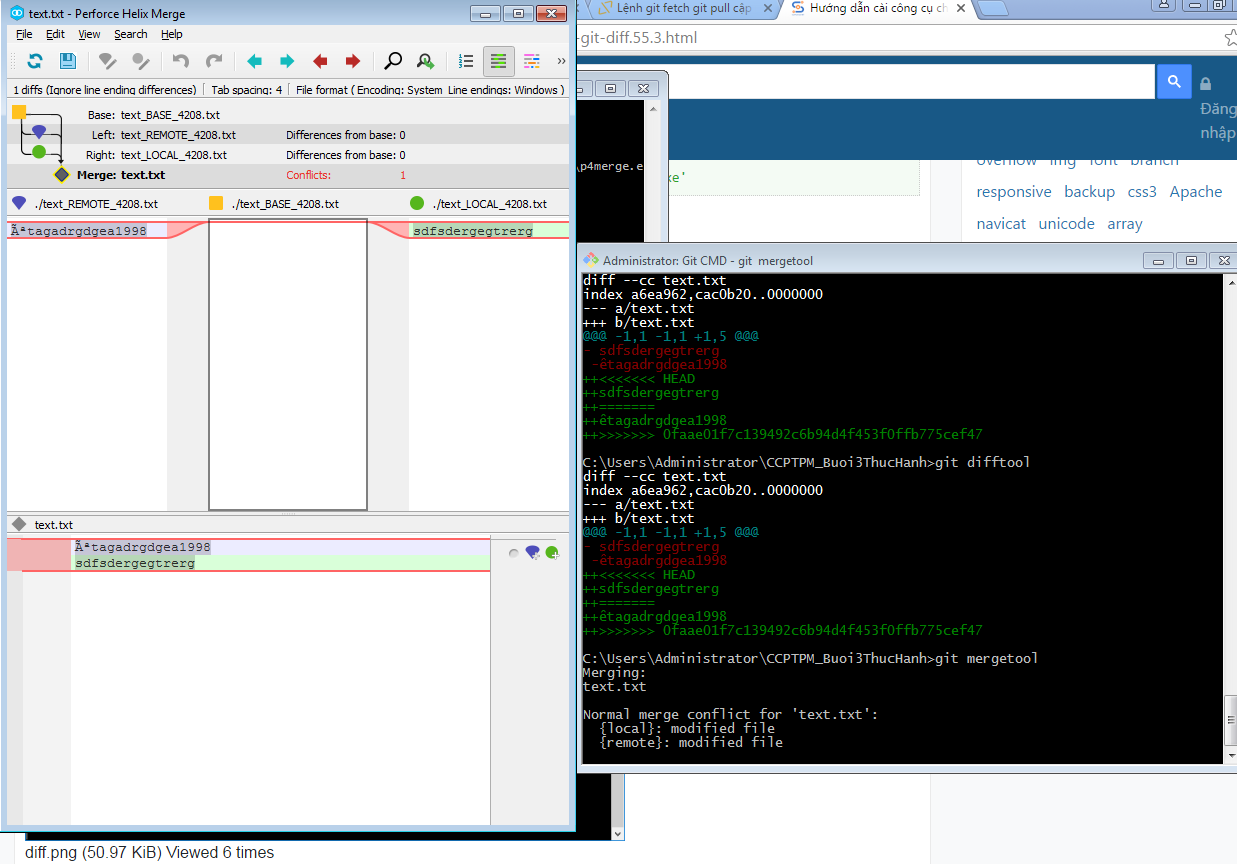
Tạo Xung Đột



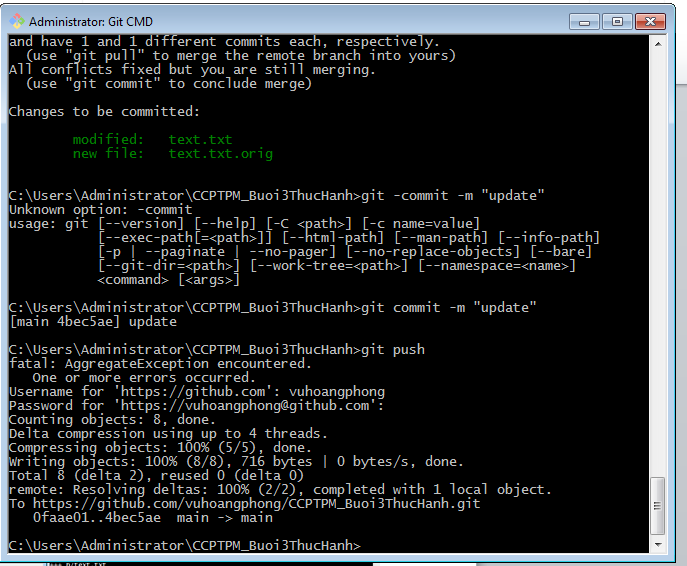
Xem sự khác biệt



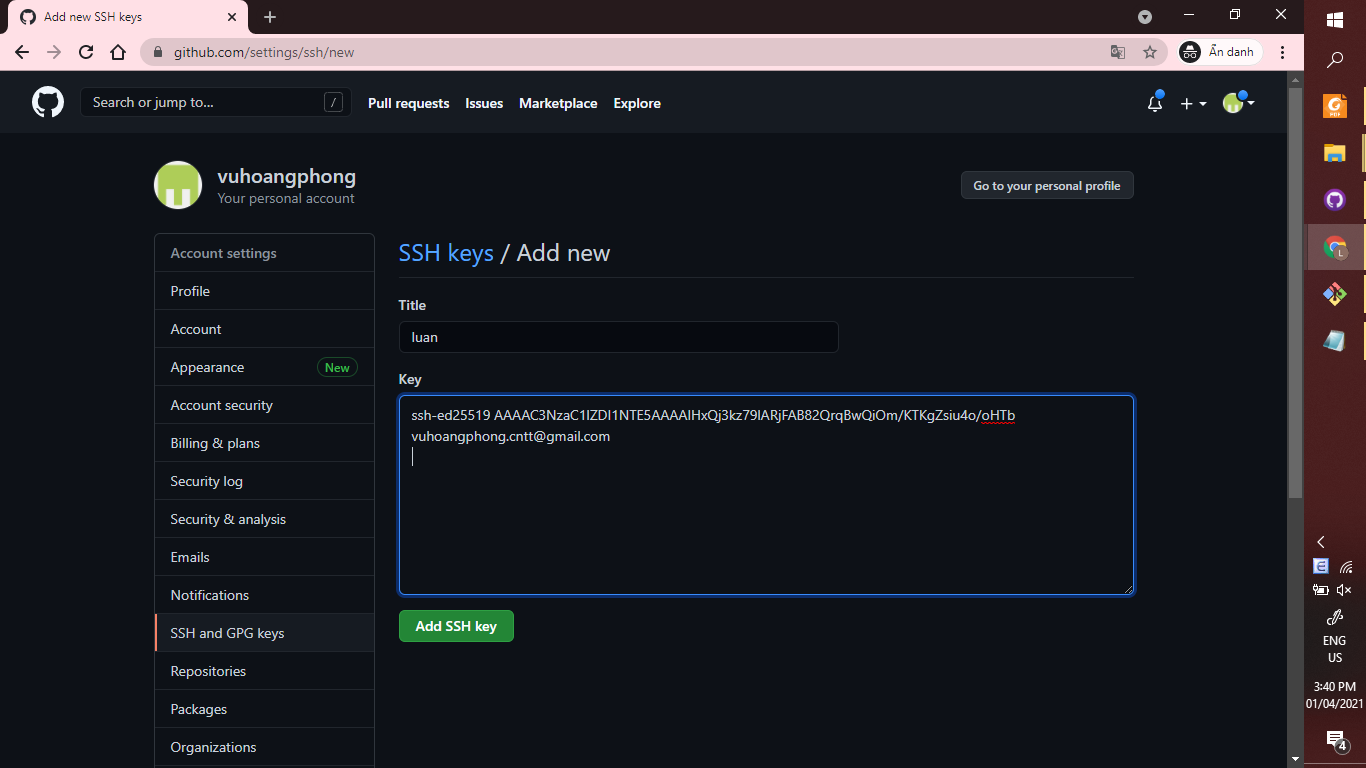
Sử dụng p4 để merge code



Chọn dòng merge và save



CÀI ĐẶT SSH KEY



**CONFIGURE TOOLING**

$ git config --global user.name "[name]"

Đặt tên người dùng khi commit lên git

$ git config --global user.email "[email address]"

Đặt email người dùng khi commit lên git

$ git config --global color.ui auto

Cho phép tô sáng dòng lệnh

CREATE REPOSITORIES

$ git init [project-name]  
Khởi tạo một Repository

$ git clone [url]  
Sao chép một kho lưu trữ từ GitHub và thêm nó vào thư mục dự án

MAKE CHANGES

$ git status  
Hiển thị trạng thái của các file sửa đổi

$ git add [file]  
Thêm file thay đổi vào staging

$ git reset [file]  
Hủy (undo) commit chưa public (chưa đẩy lên Remote Repo bằng lệnh git push) nhưng vẫn giữ nội dung

$ git diff  
Hiển thị tất cả các thay đổi chưa unstaged trong index và thư mục hiện tại

$ git diff --staged  
Hiển thị sự khác biệt giữa các file trong staging và phiên bản mới nhất

$ git commit -m "[descriptive message]"  
Commit thay đổi cùng với một thông báo tùy chỉnh

GROUP CHANGES

$ git branch  
Hiển thị tất cả các nhánh

$ git branch [branch-name]  
Tạo một nhánh mới

$ git checkout [branch-name]  
Di chuyển tới một nhánh

$ git merge [branch]  
Hợp nhất một nhánh khác với nhánh hiện tại của bạn

$ git branch -d [branch-name]  
Xóa một nhánh được chỉ định

REFACTOR FILENAMES

$ git rm [file]  
Xóa tệp khỏi thư mục làm việc và xóa các giai đoạn

$ git rm --cached [file]  
Xóa tệp khỏi kiểm soát phiên bản nhưng giữ nội dung

$ git mv [file-original] [file-renamed]  
Thay đổi tên tệp và chuẩn bị cho commit

SUPPRESS TRACKING

\*.log  
build/  
temp-\*Tệp văn bản có tên .gitignore ngăn chặn việc tạo phiên bản ngẫu nhiên của các tệp và đường dẫn khớp với các mẫu được chỉ định

$ git ls-files --other --ignored --exclude-standard  
Liệt kê tất cả các tệp bị bỏ qua trong dự án này

SAVE FRAGMENTS

$ git stash  
Bỏ tất cả các file đã sửa đổi

$ git stash list  
Xem lại danh sách đã lưu lại các thay đổi vào stash

$ git stash pop  
Truy xuất các file đã  stashed

$ git stash drop  
Xóa các file đã stashed

REVIEW HISTORY

$ git log  
Hiển thị lịch sử thay đổi

$ git log --follow [file]  
Liệt kê lịch sử phiên bản cho file, bao gồm cả tên

$ git diff [first-branch]...[second-branch]  
Cho thấy sự khác biệt về nội dung giữa hai nhánh

$ git show [commit]  
Hiển thị metadata và nội dung thay đổi của một commit

REDO COMMITS

$ git reset [commit]  
Di chuyển HEAD về vị trí commmit reset và vẫn giữ nguyên tất cả các thay đổi của file, nhưng loại bỏ các thay đổi khỏi stage

$ git reset --hard [commit]  
Di chuyển con trỏ HEAD về vị trí commmit reset và loại bỏ tất cả sự thay đổi của file sau thời điểm commit reset

SYNCHRONIZE CHANGES

$ git fetch [bookmark]  
Sao chép và tải xuống tất cả các tệp của nhánh vào thiết bị của bạn

$ git merge [bookmark]/[branch]  
Thêm các thay đổi từ nhánh này sang nhánh khác

$ git push [alias] [branch]  
Được sử dụng để cập nhật các nhánh từ xa với những thay đổi mới nhất mà bạn đã commit lên github

$ git pull  
Được sử dụng để thêm các thay đổi vào nhánh chính