CHI TIẾT
MỘT SỐ LỆNH KHI LÀM VIỆC VỚI GIT

(Tài liệu sử dụng trong buổi đào tạo nhân sự lập trình tại IMIC)

Đây là tài liệu cung cấp cho học viên ở các khóa đào tạo về kỹ năng làm việc với Git.

Hãy đảm bảo rằng bạn đã đăng ký tài khoản trên hệ thống của Github | Gitlab cũng như có sự am hiểu về những cấu hình cần thiết trên hệ thống đó. Khi bạn đã sẵn sàng hãy đi vào các nội dung chi tiết dưới đây:

1. **Please clone Remote Repositories on github to your computer**
   - ✓ $ git clone ...
2. **Let's check the configuration parameters in the git environment:**
   - ✓ $ git config --list
3. **Let's create the following branches:**
   - release
   - develop
   - feature
   - Lệnh sử dụng:
   - ✓ $ git checkout -b <branch name> 'Create a new branch
   - ✓ $ git checkout <branch name> 'switch into branch
   - ✓ $ git ls-remote 'Check the branch information on the Remote Repository
   - ✓ $ git branch -v 'list of branches
4. **Let's push the branch to Github:**
   - ✓ $ git push --set-upstream origin <branch name>
5. **Initialize the project starting with git and creating a tag:**

5.1. Initial Projec:

- ***Put resources into git databases local***
- ✓ $ git add network.txt, securities.txt,...
- ✓ $ git add Sources/*
- ✓ $ git add .
- ✓ $ git restore --staged folderA/folderB/* 'to unstage
- ✓ $ git restore --staged <file> 'to unstage
- ✓ $ git commit -am "Initialize the project structure"

- ***When you want to make a commit but don't want to log the commit many times, you can use the -amend:***

✓ $ git commit --amend

✓ $ git commit --amend -am <message>

- ***Unstage all files on Git:***

✓ $ git reset

- ***Unstage commits softly:***
  - if you want to unstage your last commit, you can the “HEAD” notation in order to revert it easily.

✓ $ git reset --soft HEAD~1

✓ $ git status

✓ $ git log

- ***Unstage commits hardly:***

✓ $ git reset --hard <commit>

- Note : be careful when using the reset hard command, you will lose all your changes when hard resetting.

5.2. Listing your tags:

- Câu lệnh $ git tag để liệt kê ra danh sách các tag trong history.
- Dấu * sẽ là các mã chưa biết sẽ giúp tìm kiếm rộng hơn.
- Lệnh sẽ là: $ git tag -l "v1.8.5*"

5.3. Creating tags:

- LightWeight tags *(nó là 1 pointer đến 1 specific commit, chứa ít thông tin).*
- Anotated tags *(lưu đầy đủ thông tin full objects trong Git Database, thông thường lựa chọn kiểu này).*

5.3.1. Annotated tags:

✓ $ git tag -a v1.0 -m "my version 1.0"

- ✓ $ git tag
- Khi muốn xem chi tiết 1 phiên bản nào đó thì sử dụng lệnh $ git show v1.0

5.3.2. Lightweight tags:

- Sử dụng lệnh sau để tạo ra 1 Lightweight tags.
- ✓ $ git tag v1.0-lw
- ✓ $ git tag

5.4. Tagging later:

- Để tạo ra tags cho lần commit nào đó dựa vào commit id thông qua các bước sau:
- **Bước 01**: Sử dụng lệnh git log để hiển thị thông tin
- ✓ $ git log --pretty=oneline
- **Bước 02**: Copy mã commit id để sử dụng cho việc tạo ra tags cho lần commit đó.
- ✓ $ git tag -a v1.2 <commit id>
- ✓ $ git tag -a v1.2 3580433de181351099b9caeee6b88aed2d335d06

5.5. Sharing tags:

- Mặc đinh tags chỉ thể hiện dưới Local Repositories. Do vậy muốn chia sẻ tags thì sử dụng các lệnh dưới đây:
- ✓ $ git push origin [tagname]
- Ví dụ: $ git push origin v1.2 (có nghĩa là chỉ đưa tags 1.2 lên Remote)
- Khi muốn đưa tất cả các tags được định nghĩa dưới Local lên Remote thì sử dụng lệnh:
- ✓ $ git push origin --tags

5.6. Checking out tags:

- Khi muốn lấy về 1 phiên bản nào đó, hoặc muốn tách từ 1 phiên bản nào đó ra thành 1 nhánh mới thì sử dụng lệnh sau:
- ✓ $ git checkout -b <tên nhánh mới> <từ phiên bản nào>
- Ví dụ: $ git checkout -b newversion v1.0

5.7. Delete tag:

- Để xóa tags ở Local: $ git tag -d {tag_name}
- Để xóa tags ở Remote: $ git push {remote_name} --delete {tag_name}
- Có thể sử dụng lệnh $ git ls-remote --tags để hiển thị ra danh sách các tags trên Remote.
- Ví dụ: $ git push origin --delete v1.0

6. Conflict when working with git:

✓ git mergetool --tool=tortoisemerge

- Choose:

✓ **Use text block from 'mine' before 'theirs'**

✓ **Use text block from 'theirs' before 'mine'**

- Complete of fix:

✓ **Delete a file *.orig (cấu hình định dạng này trong .gitignore để loại bỏ)**

✓ $ git status

✓ $ git commit -am "complete conflict handling"

✓ $ git push origin <main|...>

- **Note:**
- You need 'pull' before the update a file and 'push' into Git Server (Remote Repositories).
- You need instal tortoisegit tools for conflix error.

7. Notepadd++ Editor into Git:

✓ $ git config --list

- x64: git config --global core.editor "'C:\npp.7.6.6.bin.x64\notepad++.exe' -multiInst -nosession"

✓ $ git config core.editor *(xem lại cấu hình đã thiết lập)*