**Các bước thực hiện**

Đầu tiên: checkout sang nhánh staging – pull code mới nhất về

**1. Tạo nhánh release từ nhánh staging với tên theo format**

release-stg-<thời gian release>

Ví dụ: thời gian release vào lúc 15 giờ ngày 13/01/2019 thì tên branch sẽ là

release-stg-20190903-1000

release-prod-20190904-1100 //copy tên nhánh mẫu này đổi lại, copy đến chỗ ngày, phần thời gian xem lại trên milestone để điền cho thống nhất

**2. Tạo milestone để filter merge request**

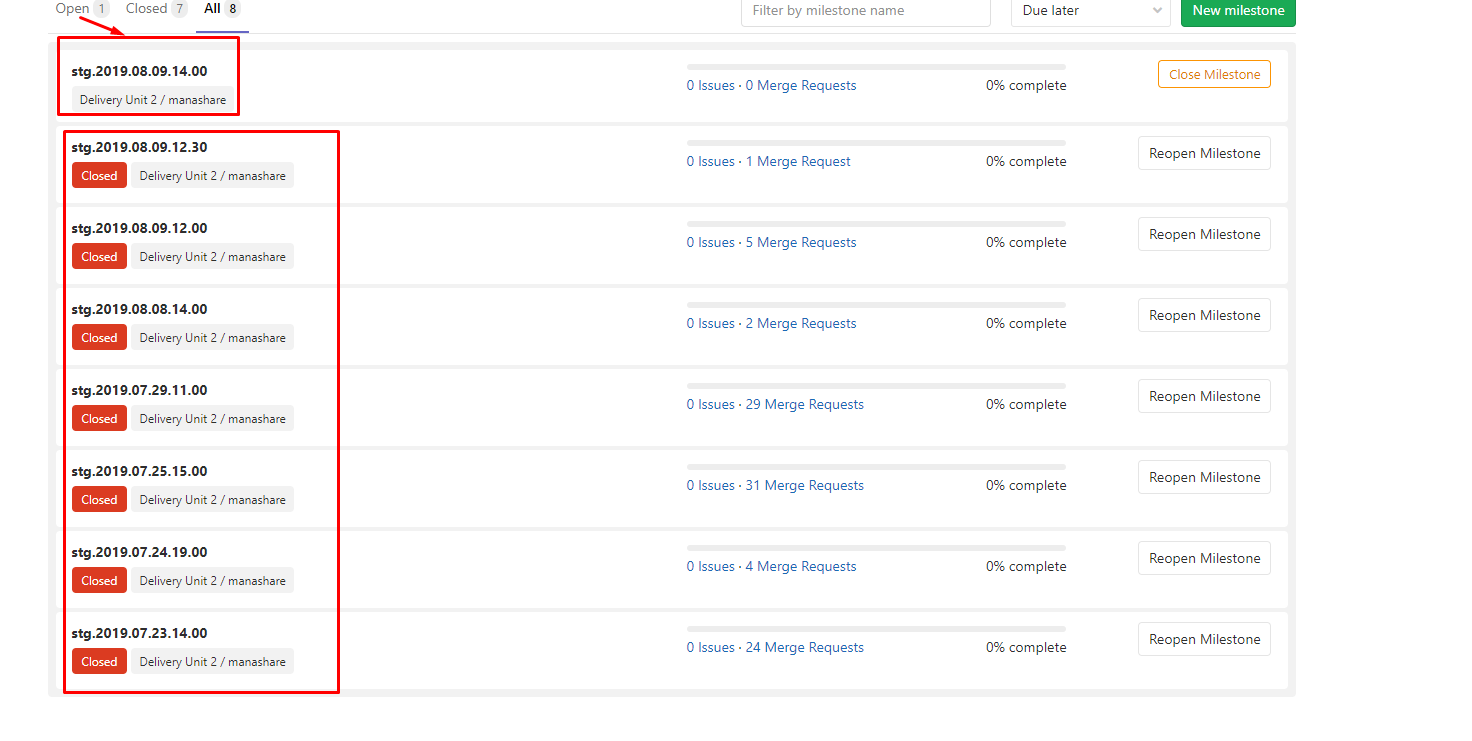
Ví dụ: milestone để release vào lúc 15 giờ ngày 13/01/2019 thì sẽ có tên như sau

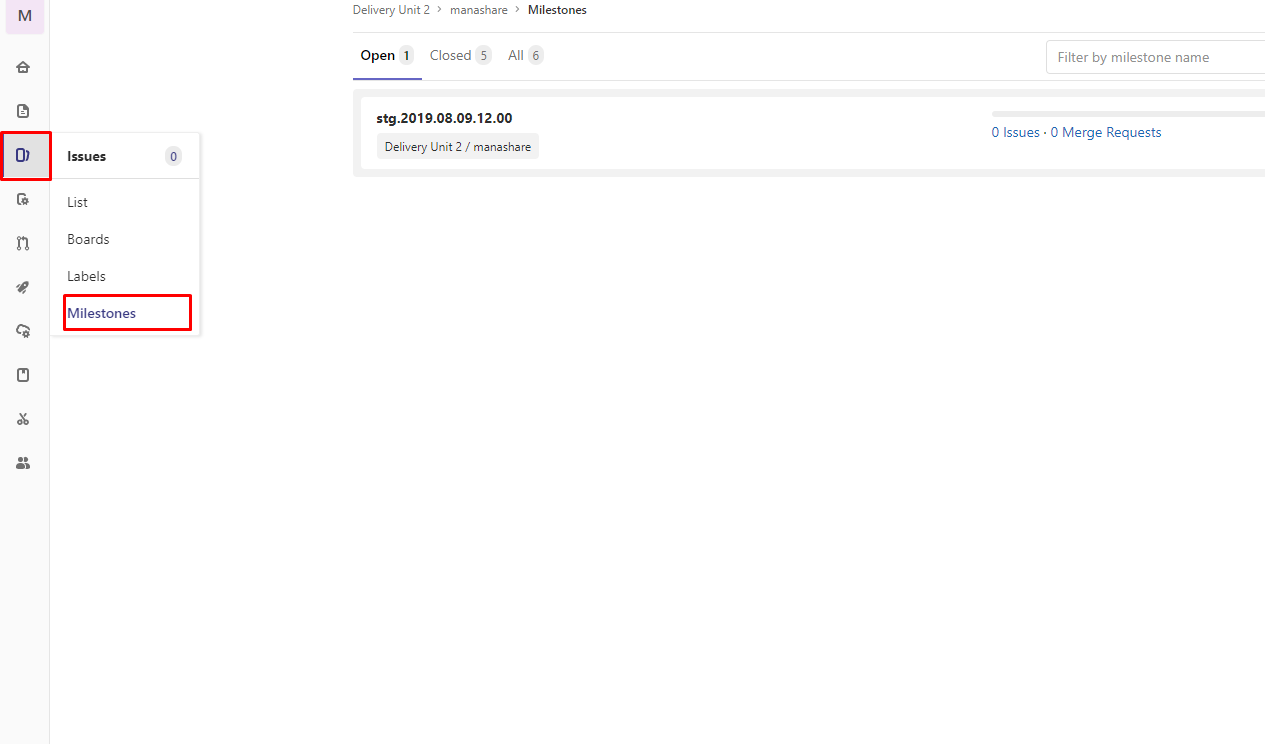
stg.2019.09.03.10.00

[prod.2019.09.03.13.00](https://source.cmcglobal.com.vn/du2/manashare/-/milestones/22)

Các milestone trước stg.2019.04.24.14.00 theo định dạng stg.YYYY.DD.MM.hh.mm

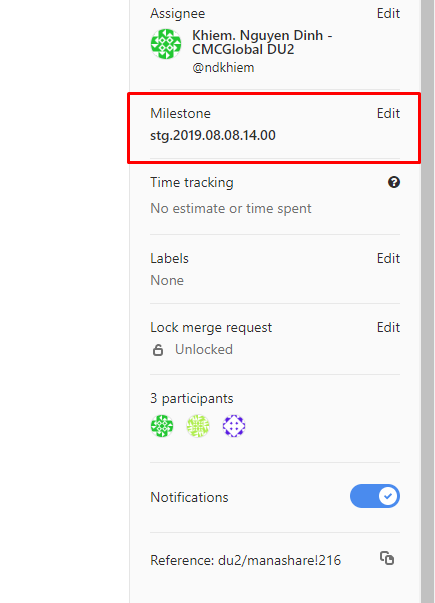
Sau khi deploy xong thì close milestone đi



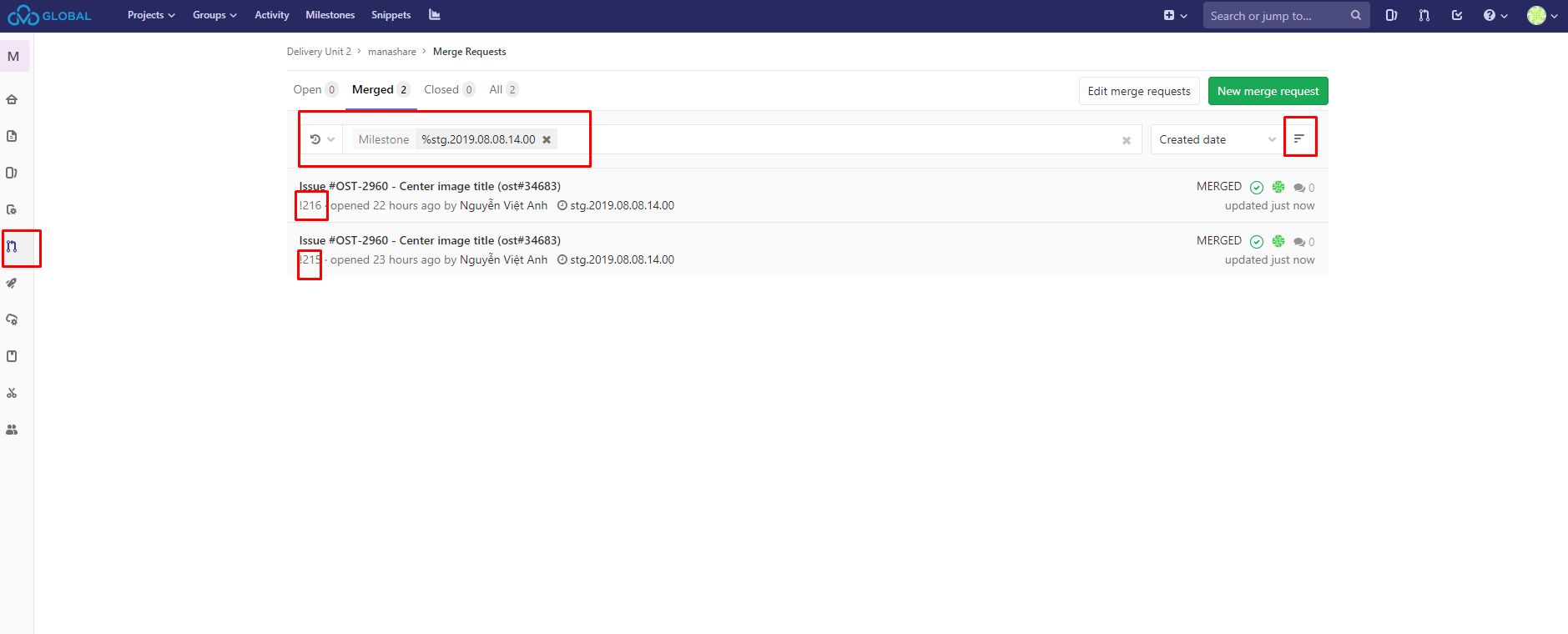


**3. Thực hiện assign các merge request của từng issue trong release note vào milestone mới được tạo ra**

Edit ở đây



**4. Filter các merge request theo milestone, sắp xếp theo thứ tự Oldest created sắp xếp các merge request theo thứ từ từ nhỏ đến lớn**



**5. Cherry pick lần lượt các commit của mỗi merge request vào nhánh release theo thứ tự từ merge request cũ nhất**

// vào từng commit trong list commit để copy mã cherry pick

// gõ câu lệnh git base như sau:

git cherry-pick 1bedaa2757e22bcf039af5fe9aae81f3aa102f7d

với mỗi cherry pick được lấy từ list đã gắn nhãn milestone ở trên

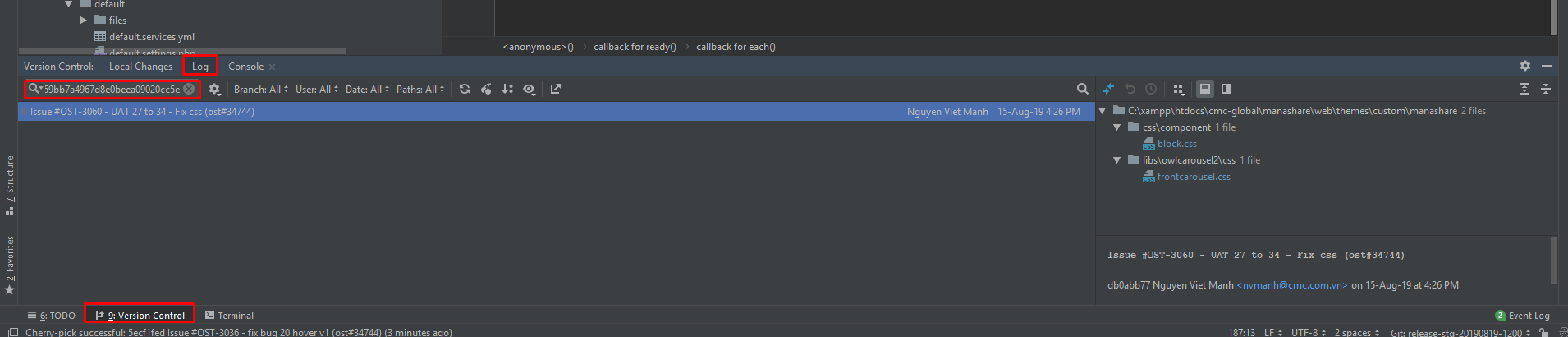
sau đó push code lên nhánh release vừa tạo

Chú ý: nếu việc cherry-pick có conflict, xử lý conflict bằng command.

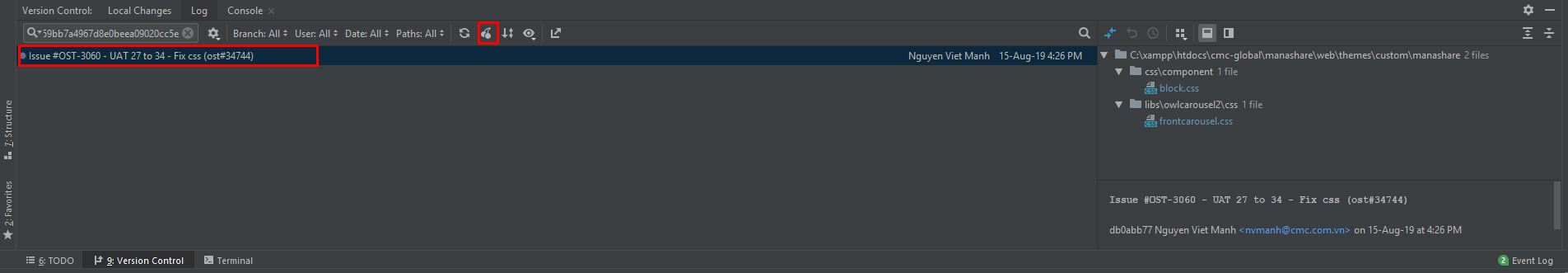
Cách 2: cherry pick sử dụng phpstorm (khi bị conflic sẽ fix dễ hơn)

B1. Mở phpstorm

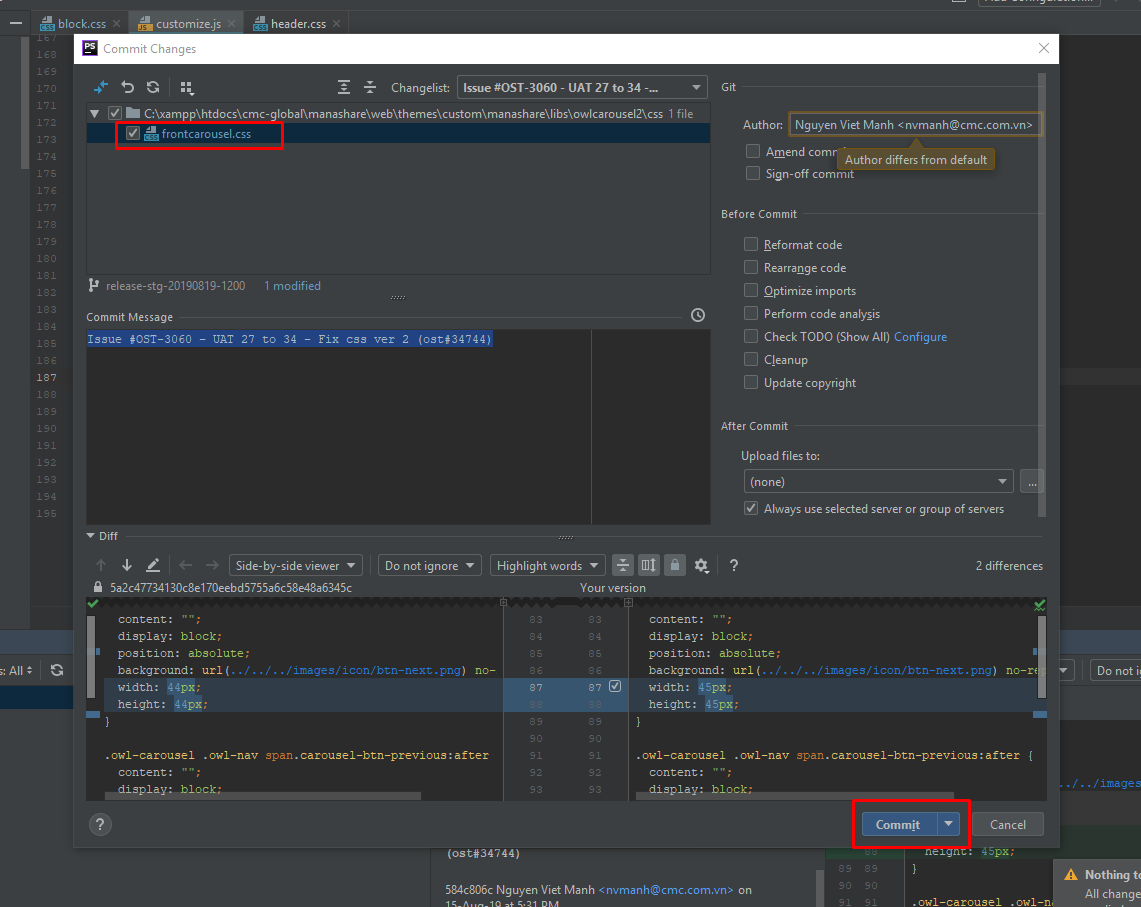
Vào version control, vào tab log, nhập mã cherry pick và bấm enter



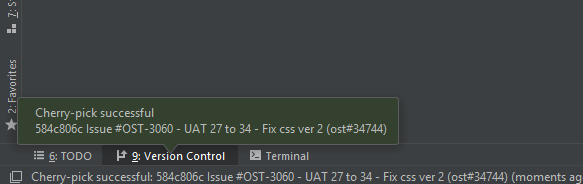
B2. Chọn vào commit và bấm vào hình quả cherry để cherry pick



B3. check 1 lượt các file thay đổi, bấm commit để cherry-pick



Thông báo cherry pick thành công



Sau đó push code lên nhánh release tương ứng

**6. Tạo merge request từ nhánh release vào nhánh staging để kiểm tra lại thay đổi, assign merge request cho reviewer**

Tên merge request theo format

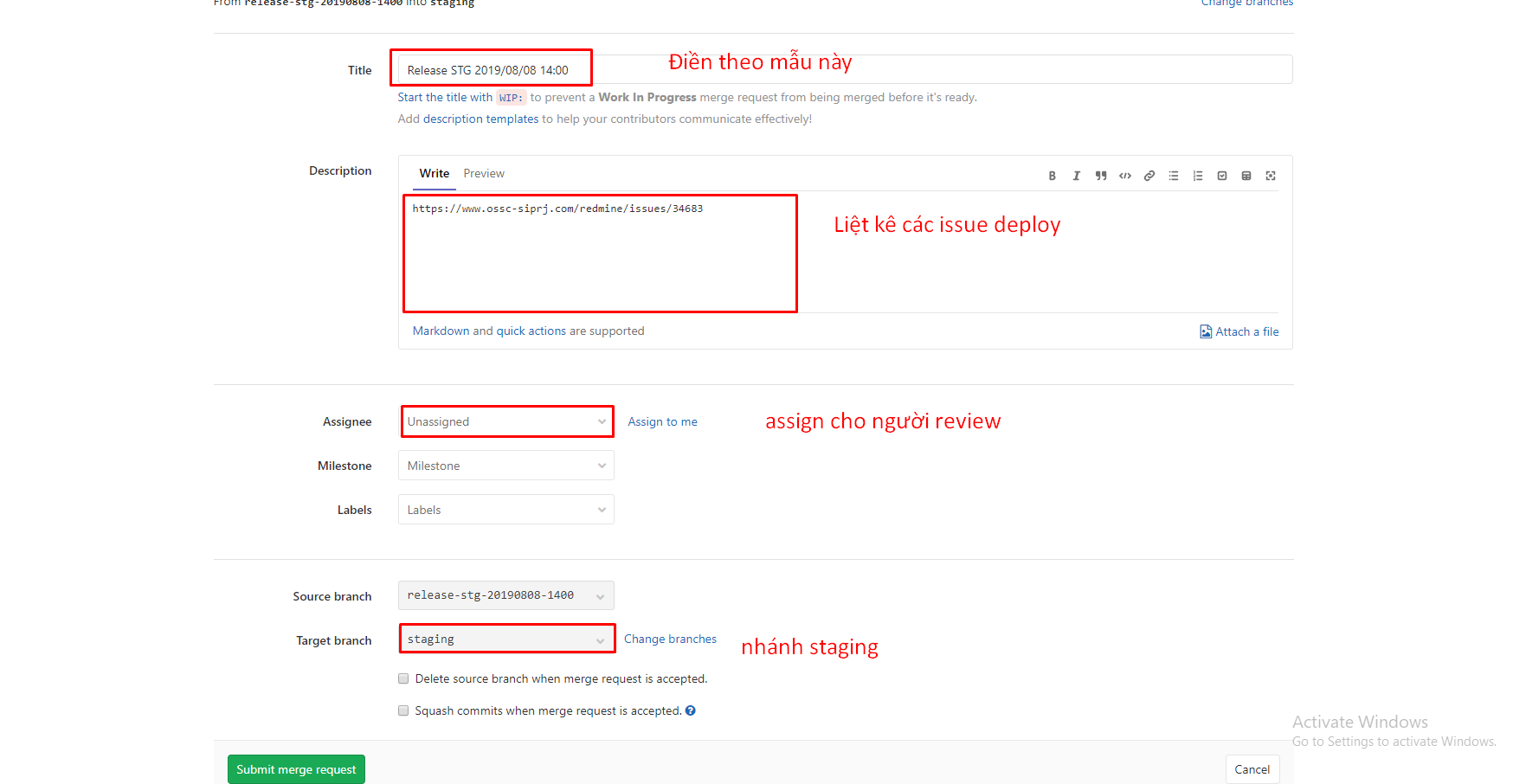
Release STG <thời gian release>

Ví dụ

Release STG 2019/09/04 11:00

Release Prod 2019/09/04 11:00 //copy ra sửa lại

Chú ý: phải gắn link của các issue vào description của merge request (để xác định được các issue đã release)



**7. Thực hiện việc merge cho merge request được tạo tại bước 6 vào nhánh staging**

**// bằng cách dùng tool xshell**

**// ví dụ dưới đây là demo cho việc deploy con manashare (cùng thư mục với kurashare nhưng kurashare sẽ trỏ đến folder www)**

**Command cd /data tức là trỏ tới thư mục data**

**Còn gõ cd/data sau đó tab 2 lần nó mới hiện ra các thư mục có trong data**

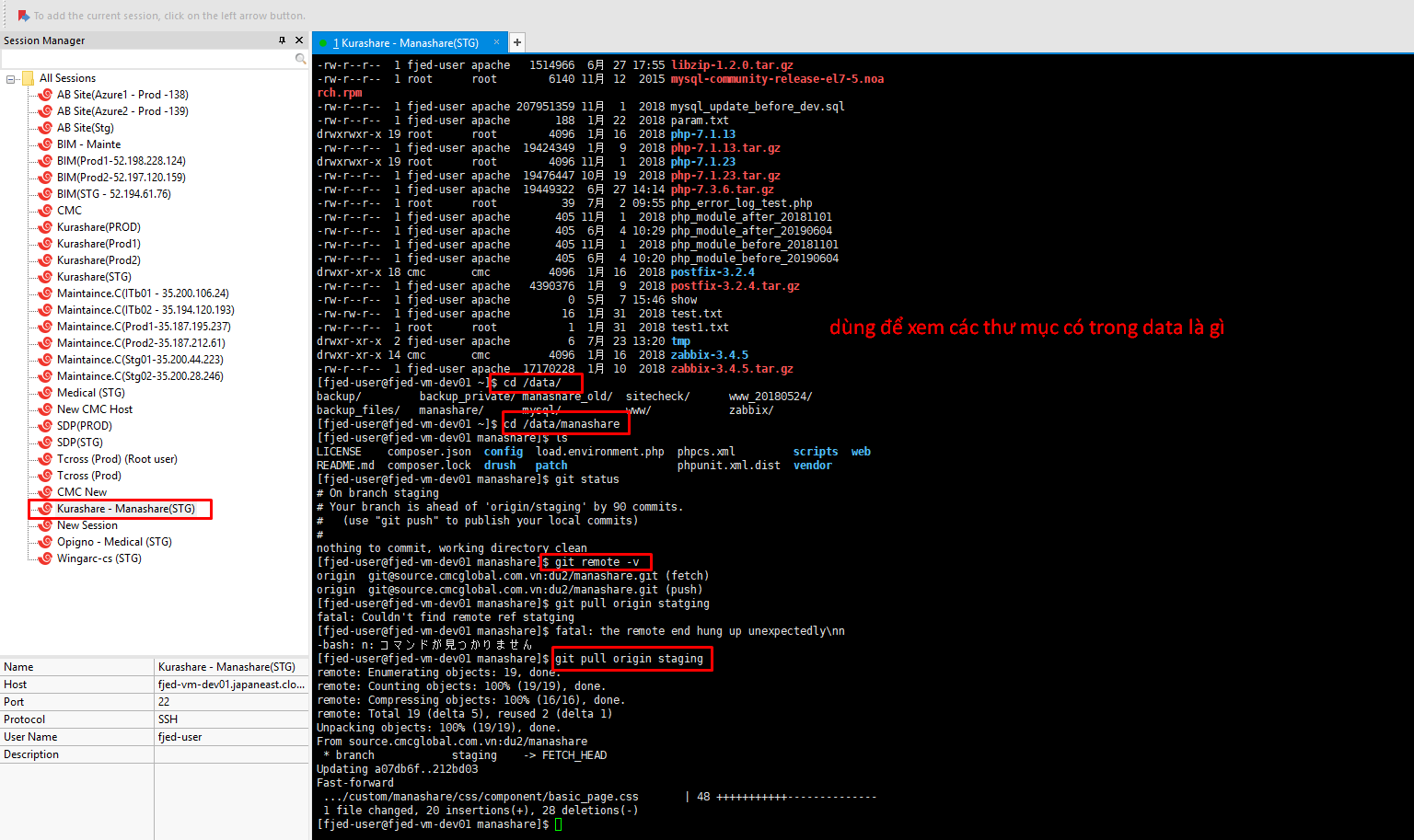
**Sau khi cd vào thư mục cần deploy thì chạy lệnh git status để kiểm tra xem khách hàng có thay đổi code gì ko, nếu có thay đổi thì phải kiểm tra lại tránh làm mất code của họ**

**Sau đó login vào môi trường của họ, vào phần configure xem họ có configure gì ko, nếu có thì cũng cần backup lại ko sẽ mất configure của họ**

* **export các file configure đó ra (export single), sau khi import các setting của mình xong thì lại import trả lại các file cofigure đó cho họ (import single, chọn đúng kiểu file, paste code và import)**

**Cuối cùng mới pull code vào nhánh staging của họ**

// chú ý: trên môi trường linux này cần viết xong tab để tránh nhầm lẫn



**Sau đó vào test lại trên môi trường tương ứng: nếu trong file deploy ko có thông tin thì vào redmine -> chọn project rồi vào wiki để tìm**

**8. Trên server Tcross, truy cập vào thư mục staging, kiểm tra tình trạng file.**

cd ~/html\_stg/

git status

Chú ý: Vì có thể có cả phần phát triển từ bên thứ 3 nên nếu git status có untracked files, cần confirm lại với khách hàng để merge cho hợp lí

**9. Nếu có setting file, cần backup database trước khi tiến hành deploy**

1. Tạo thư mục backup tại thư mục deploy, tên thư mục theo định dạng bk\_db\_stg\_<thời gian release> Ví dụ

mkdir ~/deploy/bk\_db\_stg\_20190113

1. dump db với mysqldump

mysqldump -h 127.0.0.1 -u tcross\_stg -p tcross\_stg > ~/deploy/<thư mục backup>/tcross\_stg.sql

Ví dụ

mysqldump -h 127.0.0.1 -u tcross\_stg -p tcross\_stg > ~/deploy/bk\_db\_stg\_20190113/tcross\_stg.sql

Chú ý: password của mysql lấy trong redmine wiki của project tcross

$databases['default']['default'] = array (

'database' => 'manashare\_db',

'username' => 'manashare\_user',

'password' => G#Hwtwa7LX,L

'prefix' => '',

'host' => '127.0.0.1',

'port' => '3306',

'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',

'driver' => 'mysql',

);

**10. Pull code từ nhánh stg**

cd ~/html\_stg/

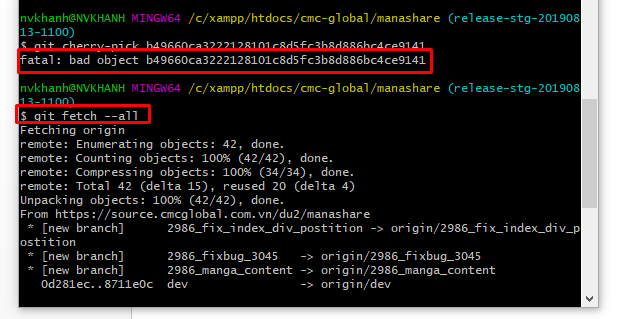
git pull cmc staging

**11. Tiến hành thực hiện setting theo release note.**

Chú ý: sau khi setting xong nên review lại hoạt động của site trên môi trường

**12. Thực hiện xóa cache nếu cần**

**13. Chú ý: Nếu cherry-pick commit bị lỗi** như sau thì xử lý như trên hình



Nguyên nhân là do commit chưa có trên local nên phải fetch all về

**14. Chú ý nếu cherry pick các commit ko theo thứ tự từ nhỏ đến lớn sẽ bị lỗi ko megre được**

1. git log --oneline để xem các commit đã được commit

2. git reset --hard origin/staging

Dùng lệnh này để trở về lúc chưa cherry pick

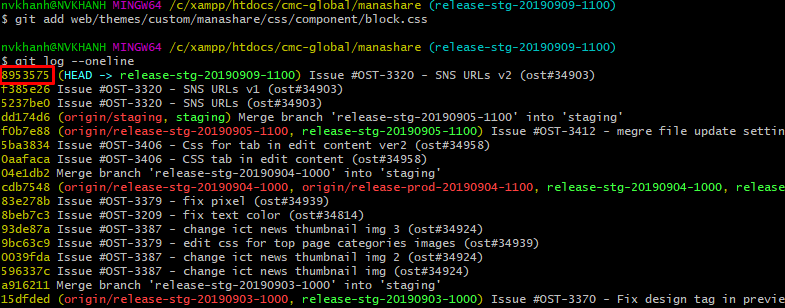
1. cherry pick như bình thường và push code dùng –f

**còn nếu muốn reset về issue ngay trước cái issue đang cherry pick**(do megre conflic lỗi chẳng hạn)

**-> dùng lệnh sau:**

git reset --hard 8953575

với mã hard là cái mã của cherry pick cần đẩy code về

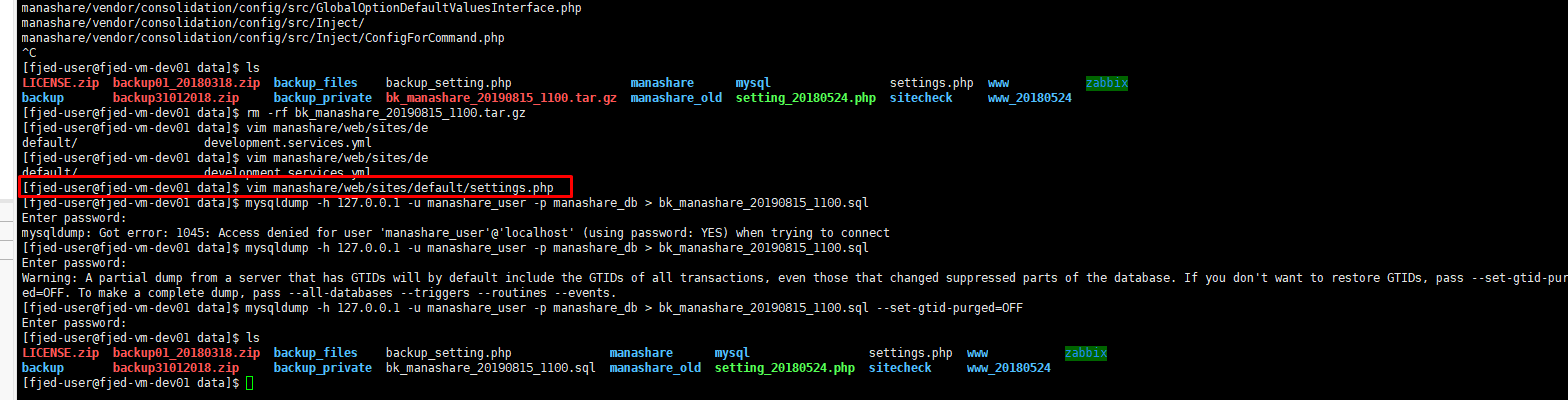


**15. chú ý: nếu có up core** thì sau khi pull code vào nhánh staging thì chạy composer install luôn rồi sau đó chạy update.php sau cùng mới import configure

**Cách backup db**

1. Từ thu mục data dùng lệnh sau để trỏ tới file setting.php

vim manashare/web/sites/default/settings.php



Phần thông tin của db sẽ hiện ra (nếu phần thông tin quá dài thì bấm shift + G để xuống tận cùng)

Copy lấy phần thông tin này để sử dụng cho việc backup db

Để thoát ra bấm nút esc sau đó gõ :q

1. Dùng câu lệnh sau để backup db

mysqldump -h 127.0.0.1 -u tcross\_stg -p tcross\_stg > ~/deploy/<thư mục backup>/tcross\_stg.sql

với

–h là host

-u là username

-p là password nhưng sẽ chưa nhập gì -> sẽ nhập sau

manashare\_db là database sẽ backup

> là biểu thị sẽ backup từ manashare\_db sang db backup được đặt tên là bk\_manashare\_20190815\_1100.sql

**'database' => 'kurashare\_dev',**

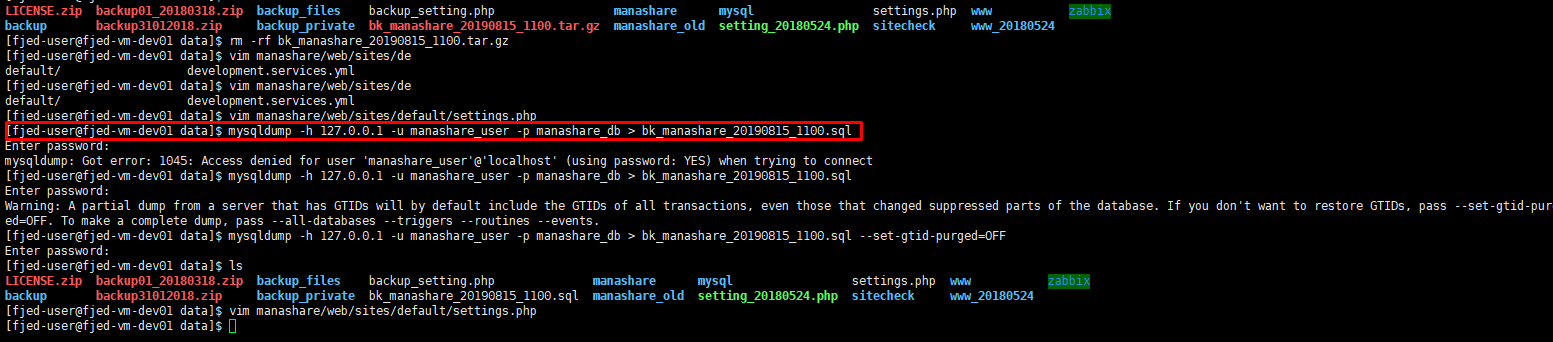
**'username' => 'drupal',**

**'password' => 9x(t.StG~\*He**

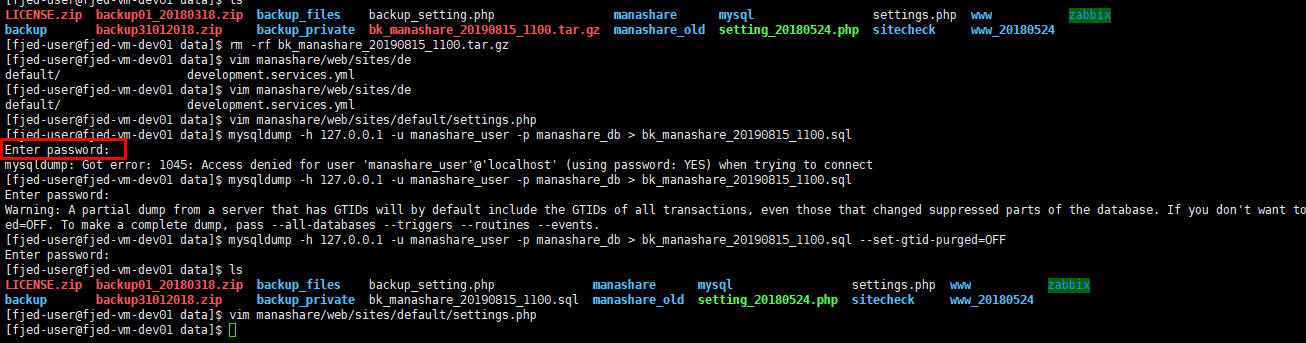
**'prefix' => '',**

**'host' => '127.0.0.1',**

**'port' => '3306',**

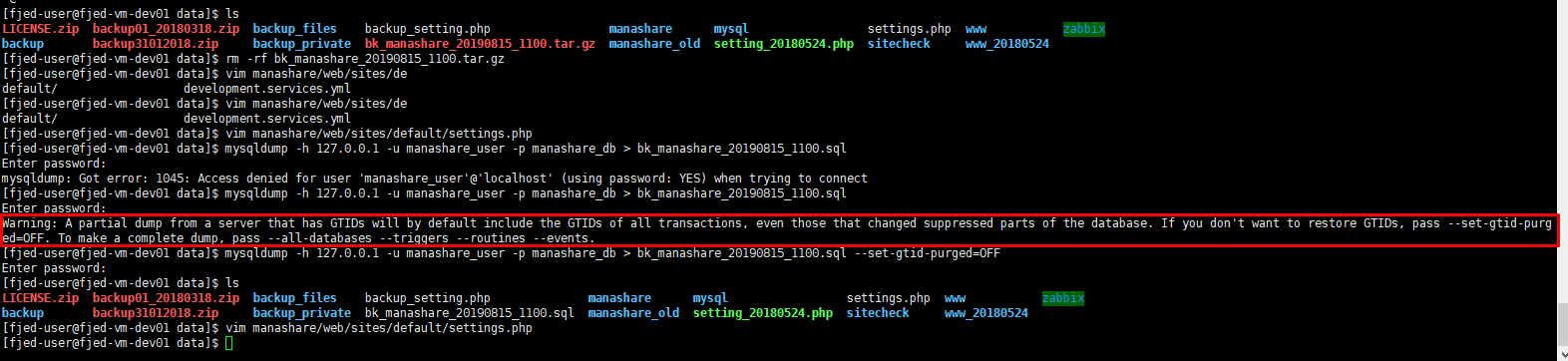


Sau đó sẽ có yêu cầu nhập password



Copy password vào (dùng shift + insert)

Sau đó nếu xuất hiện warning như sau (ở đây nó xuất hiện warning nhưng vẫn xuất db bình thường)



Thì thêm phần lệnh sau vào câu lệnh backup db là được



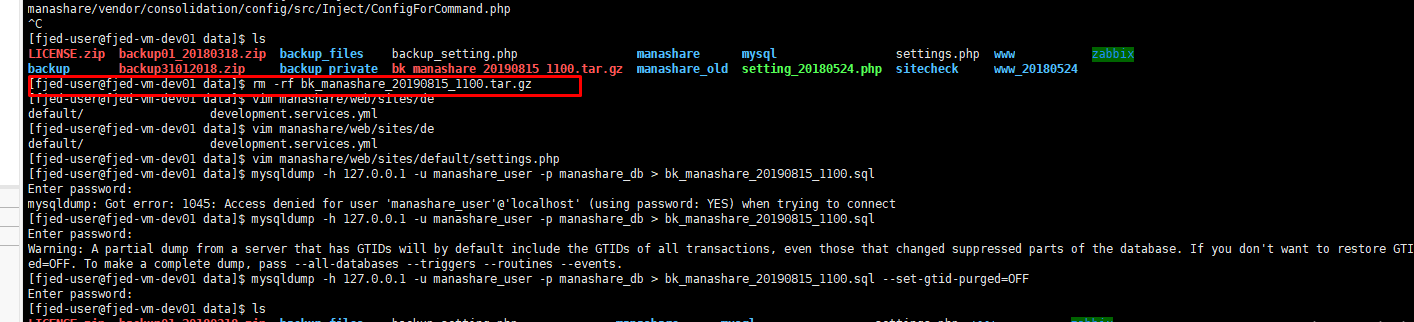
Sau khi thêm phần lệnh sau nó sẽ tự đè db mới lên db cũ do trùng tên

Sau khi deploy xong, test lại oke thì xóa phần backup đi

Làm như sau:

Trước hết gõ lệnh ls để xem chính xác thư mục cần xóa

Rồi dùng lệnh rm –rf để xóa



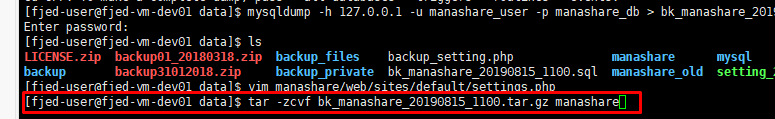
**Cách backup code**

Dùng câu lệnh sau

tar -zcvf bk\_manashare\_20190815\_1100.tar.gz manashare

với bk\_manashare\_20190815\_1100.tar.gz là tên file code backup

manashare là thư mục code backup ra



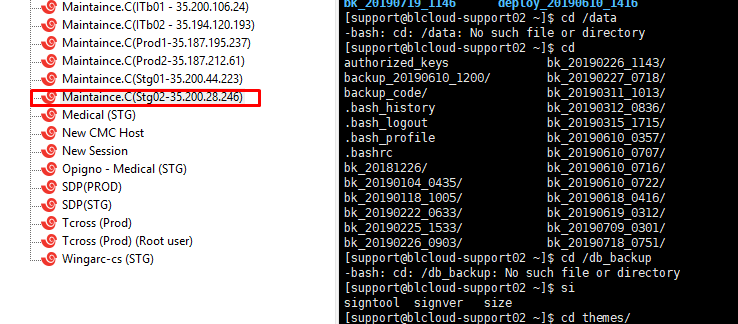
**Chú ý: Vì backup code lâu và kể cả code có lỗi thì vẫn có cách revert lại được nên về cơ bản là ko cần**

**Lệnh dùng để check các commit đã được cherry pick**

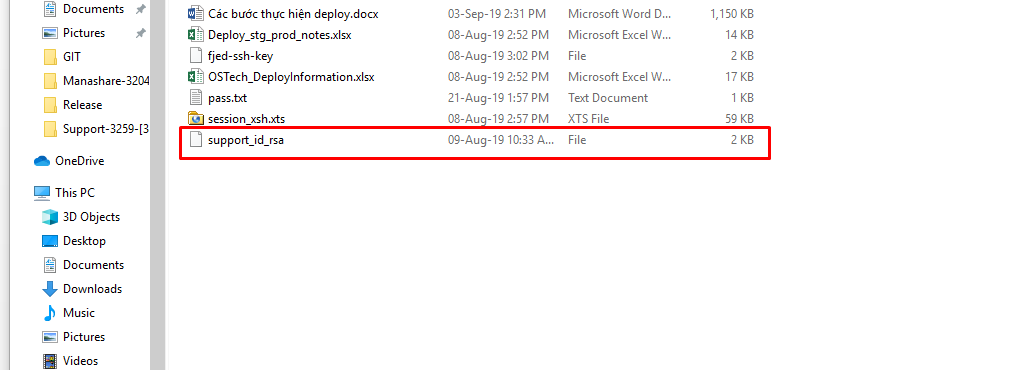
git reflog

**Backup DB con support (stg 2)**

**B1. Chọn vào mục này**



Sau đó nhập mã và password tương ứng

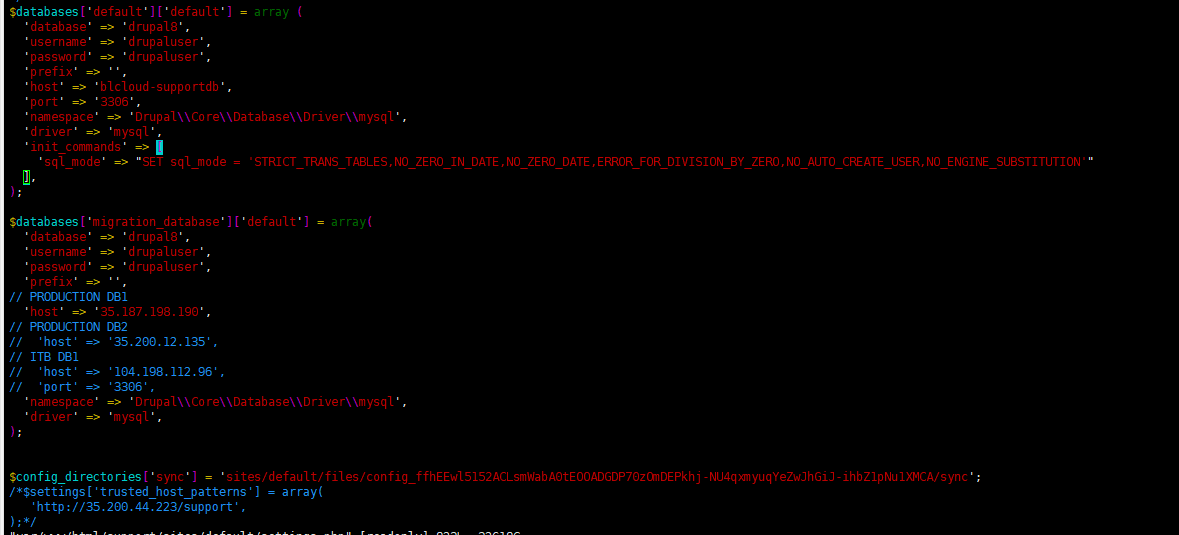


Và mình sẽ backup ở vị trí ~ vì ở vị trí này mình có quyền đọc và ghi

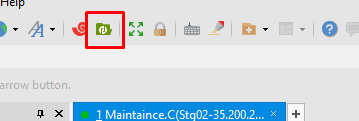


Cd / để vào thư mục /

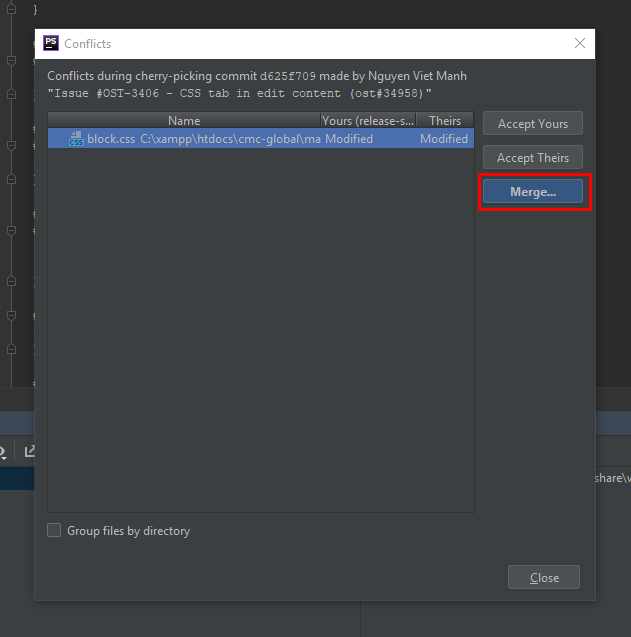
Sau đó lấy thông tin db theo đường dẫn sau: var/www/html/support/sites/default/settings.php



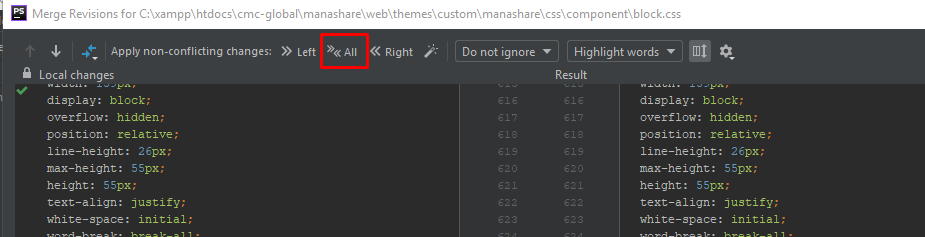
Sau khi backup xong muốn copy db ra máy sử dụng xftp



Click vào đây

**Nếu bị conflic thì bấm vào nút megre** 

**rồi chọn nút all để apply tất cả các change mà không gây conflic trước đã rồi sẽ megre conflic bằng tay các chỗ còn lại**



**Phần code bên tay trái là code ở trên local của máy mình (nó không có phần code bị báo conflic do issue đó vẫn chưa được deploy)**

**Phần code bên tay phải là code đang được cherry pick (do người làm đã pull code từ dev về nên đã có cả phần code chưa được deploy kia)**

* **Nguyên nhân: Do phần code mới đang được cherry pick (đỏ đậm) được nhận biết là ở phía dưới đoạn code bị conflic, tuy nhiên ở local chưa có phần code này nên bị báo confilc**
* **Cách sửa: đẩy hết phần code bên phải sang rồi xóa phần conflic đi**

