





KIẾM TIỀN TRÊN YOUTUBE  
NGAY HÔM NAY!  
Dùng Mã **ThachPham** Giảm 500k

✓ THU NHẬP THỤ ĐỘNG LÊN TỚI **07** con số

✓ ĐI DU LỊCH MÀ TIỀN VẪN VÀO TÀI KHOẢN

ĐĂNG KÝ  
NGAY

dgm.vn

# [Git] Đánh dấu commit với Tag



Cập nhật lúc 25/04/2015



No Comments



Công cụ , Web Development

Bài này thuộc phần 7 của 11 phần trong serie [Git cơ bản](#)

15 CHIA SẺ



Facebook

11



Twitter

2

G+ Google+

1



Pocket

1

Trong khi bạn commit các lần chỉnh sửa của mình thì mọi thứ sẽ được đều lưu vào log mà mình đã hướng dẫn bạn cách xem `commit log` ở [phần trước](#). Tuy nhiên nếu bạn commit quá nhiều thì sẽ gây khó khăn cho bạn về sau nếu cần xem lại thông tin của lần commit trước mà bạn có thể gắn thẻ đánh dấu (tag) cho mỗi commit và khi cần xem bạn chỉ cần sử dụng lệnh `git show tên_tag` là đã có thông tin rất rõ ràng, ngoài ra nó còn giúp bạn dễ dàng diff (đối chiếu) sau này khi không cần nhớ checksum (dù chỉ cần nhớ vài ký tự đầu tiên) của mỗi commit

mà chỉ cần nhớ tag, cũng như có thể tạo thêm branch từ tag để bạn thuận tiện hơn trong việc phân nhánh.

## Lightweight Tag và Annotated Tag

Trong Git có hai kiểu tag chính đó là:

- **Lightweight Tag:** Các tag này chỉ đơn thuần là đánh dấu snapshot của commit.
- **Annotated Tag:** Với tag này, bạn có thể đặt tiêu đề cho tag, và khi xem nó sẽ có thông tin về người tag, ngày tag,....

## Cách tạo Lightweight Tag

Trước tiên, bạn có thể gõ `git tag` để xem danh sách các tag có trong dự án của bạn. Sau đó để tạo thêm một tag, bạn có thể gõ `git tag tên_tag` để tạo. Ví dụ như `v1.0` chẳng hạn.

```
$ git tag v1.0
$ git tag
v1.0
```

Bây giờ bạn có thể xem thông tin của lần commit được gắn tag này bằng lệnh `git show tên_tag`. Lưu ý rằng lệnh trên nó sẽ đánh dấu lần commit cuối cùng của bạn vào tag `v1.0`.

```
$ git show v1.0
commit 05193375f7a7c1295fd26c6388d81e188f405b0b
Author: Thach Pham <contact@thachpham.com>
Date: Thu Apr 23 02:20:50 2015 -0700

    Added a new tag

diff --git a/tag.html b/tag.html
new file mode 100644
index 0000000..e69de29
```

## Cách tạo Annotated Tag

Để tạo Annotated Tag thì bạn cũng sử dụng lệnh git tag nhưng sẽ có thêm tham số `-a` và tham số `-m` để thiết lập lời nhắn cho tag này.

```
$ git tag -a v1.0-an -m "Ra mat phien ban 1.0"
$ git show v1.0-an
tag v1.0-an
Tagger: Thach Pham <contact@thachpham.com>
Date: Thu Apr 23 02:41:11 2015 -0700

Ra mat phien ban 1.0

commit d5a599e3385a8fc7a65958ed50bc8b54666b45ad
Author: Thach Pham <contact@thachpham.com>
Date: Thu Apr 23 02:40:31 2015 -0700
```

```
Commit for Annotated Tag
```

```
diff --git a/tag.html b/tag.html
index e69de29..fea03c1 100644
--- a/tag.html
+++ b/tag.html
@@ -0,0 +1 @@
+Annotated Tag
```

Bạn có thể thấy khi show ra, cái Annotated tag sẽ có nhiều thông tin hơn là so với cái tag thông thường, và đây cũng là kiểu tag bạn nên sử dụng để có nhiều thông tin hơn.

## Thêm tag cho các commit cũ

Ở lệnh trên thì bạn chỉ tạo ra tag cho commit cuối cùng của bạn. Còn nếu bạn có rất nhiều commit trước đó mà cần gắn tag thì chỉ cần thêm mã checksum (hoặc một đoạn mã checksum) của lần commit đó.

Để xem mã checksum của các lần commit trước đó thì bạn có thể sử dụng git log với tham số `--pretty` với giá trị `oneline` để lọc log nhé.

```
$ git log --pretty=oneline
d5a599e3385a8fc7a65958ed50bc8b54666b45ad Commit for Annotated Tag
05193375f7a7c1295fd26c6388d81e188f405b0b Added a new tag
435f642f951fbab1037fc2feef239ab26d6e6115 Added faq.html
6904d5232bf90821068279311e205e3e1ff929f1 Initial commit
```

Và bây giờ mình có thể đặt tag cho commit *Initial commit* thì sẽ khai báo một đoạn mã checksum của nó vào lệnh `git tag` như sau.

```
$ git tag -a v0.0 6904d -m "Tag for inintial commit"
$ git tag
v0.0
v1.0
v1.0-an
```

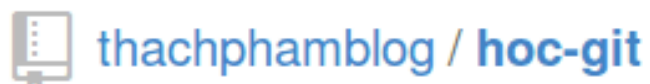
## Push Tag

Mặc định lệnh `git push` sẽ không push các tag đã tạo lên repository mà bạn có thể dùng lệnh `git push --tags` để đẩy toàn bộ tag lên repository.

```
$ git push --tags
Username for 'https://github.com': thachphamblog
Password for 'https://thachphamblog@github.com':
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 775 bytes | 0 bytes/s, done.
Total 7 (delta 1), reused 0 (delta 0)
To https://github.com/thachphamblog/hoc-git.git
 * [new tag] v0.0 -> v0.0
```

```
* [new tag] v1.0 -> v1.0
* [new tag] v1.0-an -> v1.0-an
```

Nếu bạn dùng Github hay các dịch vụ tương tự thì bây giờ bạn sẽ thấy tag xuất hiện ở repository.



Unwatch 1

Ví dụ kho chứa trong serie Git cơ bản — Edit

2 commits

1 branch

3 releases

1 contributor



branch: master

hoc-git / +



Added faq.html



thachphamblog authored an hour ago

latest commit 435f642f95

README.md

Initial commit

an hour ago

faq.html

Added faq.html

an hour ago

README.md

# Huong dan Git co ban

## Nhập tag vào branch

Mặc dù ở bài sau mình sẽ giải thích qua branch nhưng tạm thời bây giờ bạn có thể hiểu branch là **một phân nhánh trong một cây dự án** để bạn sửa mã nguồn mà không ảnh hưởng đến phân nhánh gốc (master).

Bây giờ bạn có thể truy cập vào dữ liệu mà bạn đã commit thông qua tag kèm theo việc tạo một branch mới với lệnh `git checkout -b tên_branch tên_tag`. Ví dụ mình muốn đưa cái tag `v1.0-an` vào một branch mới tên `version1` thì sẽ viết như sau:

```
$ git checkout -b version1 v1.0-an
Switched to a new branch 'version1'
```

Lúc này bạn đã tự động chuyển qua branch `version1` thay vì `master` ban đầu, kèm theo đó là dữ liệu của lần commit được gắn tag `v1.0-an`.

Bây giờ nếu bạn muốn làm việc thêm với branch này thì cứ làm việc như khi bạn ở `master`. Để chuyển về lại master, bạn gõ lệnh `git checkout master`.

Để push cái branch này lên bạn có thể sử dụng lệnh `git push origin version1`. Nếu bạn quên thì mình nhắc

lại nhé, **origin** chính là tên địa chỉ remote của repository mặc định mà Git tự đặt khi bạn clone.

Vi dụ kho chứa trong serie Git cơ bản — Edit

2 commits

2 branches

3 releases

1 contributor

Your recently pushed branches:

version1 (less than a minute ago)

Compare & pull request

Switch branches/tags

Find or create a branch...

Branches

Tags

✓ master

version1

# Huong dan Git co ban

1  
2  
3



Bây giờ bạn thử vào xem cái version1 và cái master xem có thấy sự khác nhau của nó không? Ở bài nói về branch mình sẽ giải thích cho bạn kỹ hơn. Đại loại là bạn tạo một phân nhánh khác để sửa code riêng mà không ảnh hưởng đến code chính.

## Lời kết

Trong bài này chắc bạn đã thấy được sức mạnh của Git là như thế nào và khi nó kết hợp với branch thì sẽ rất tuyệt vời trong việc phân chia nhánh cho mã nguồn để bạn làm việc linh hoạt hơn, dễ dàng chuyển đổi qua các phiên bản mà không cần làm gì nhiều. Bạn đừng lo lắng khi chưa hiểu rõ về branch, mình sẽ giải thích ngay ở phần sau đây.

### Xem tiếp bài trong serie

Phần trước: [\[Git\] Git Log và Undo Commit](#)

Phần kế tiếp: [\[Git\] Sơ lược Remote Repository và Origin](#)

### Có liên quan

[\[Git\] Branch - Kỹ thuật phân nhánh](#)

23/04/2015

Trong "Công cụ"

[\[Git\] Hiểu thêm về Commit và](#)

[Staging Area](#)

23/04/2015

Trong "Công cụ"

[\[Git\] Git Log và Undo Commit](#)

23/04/2015

Trong "Công cụ"

[git](#)[linux](#)

## Thach Pham

Đam mê với web và lập trình, thích viết và chia sẻ, nghiện cà phê và xăm mình, hứng thú với nhạc dân ca và nhạc không lời.

[← Xem thêm bài viết](#)[RSS Subscribe](#)

**BÍ QUYẾT SEO HIỆU QUẢ**  
**Chiến lược SEO lên Top Google trong 1 tháng**  
Dùng Mã **ThachPham** Giảm 500k - [www.dgm.vn](http://www.dgm.vn)

## Đề lại bình luận

Hãy là người "bóc tem"

Thông báo cho



Email





b i link b-quote u img ul ol li code

Bắt đầu thảo luận

# HOSTING 50.000đ

LiteSpeed | Free SSL





**Đăng ký ngay »**

AZDIGI.com

Bài mới nhất



## Hãy tắt XML-RPC khi không cần sử dụng



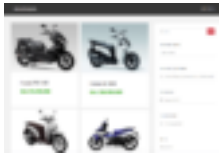
18/09/2016



## 19 themes WordPress miễn phí đẹp nhất tháng 8 – 9/2016



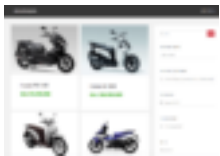
18/09/2016



## Làm trang shop đơn giản với WordPress – 07



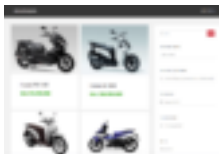
04/09/2016



## Làm trang shop đơn giản với WordPress – 06



04/09/2016



## Làm trang shop đơn giản với WordPress – 05



31/08/2016

**DGM**  
Data Marketing Training & Consulting

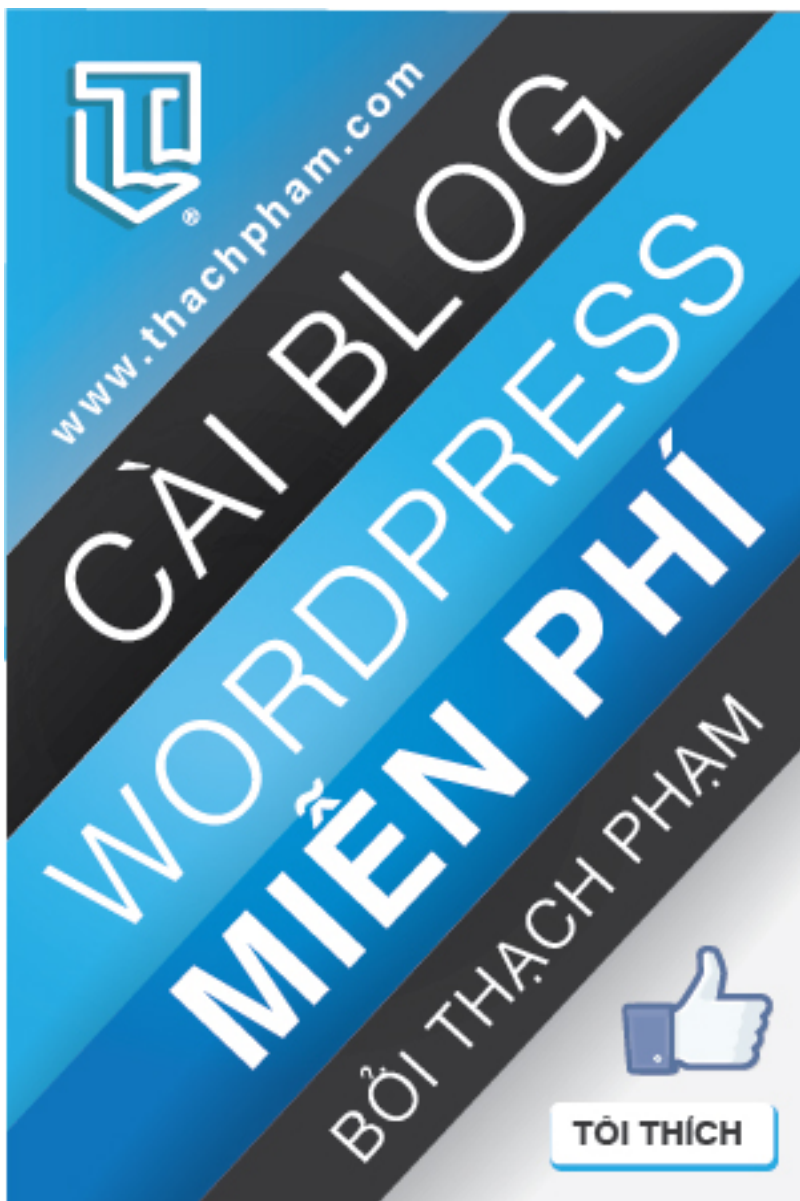


**facebook**

facebook  
**MARKETING**

DÙNG MÃ **THACHPHAM** GIẢM 500K

Bán được hàng **NGAY** trong khóa học



Dành cho người mới



[Học WordPress cơ bản](#)



[Học làm website bán hàng](#)



[Học sử dụng VPS/máy chủ Linux](#)



[Học HTML cơ bản](#)



[Học CSS cơ bản](#)

## Mới cập nhật



### Hãy tắt XML-RPC khi không cần sử dụng



Cập nhật lúc 20/09/2016



### 19 themes WordPress miễn phí đẹp nhất tháng 8 – 9/2016



Cập nhật lúc 18/09/2016



### LEMP cho VPS [Phần 1] – Cài đặt NGINX và PHP-FPM



Cập nhật lúc 17/09/2016

[Giới thiệu](#)

[Tham khảo](#)



ThachPham.Com là blog cá nhân tập trung vào việc phổ biến kiến thức WordPress nói riêng và các kiến thức làm website nói chung đến tất cả mọi người.

Phương châm hoạt động của thachpham.com đó là chia sẻ không giới hạn về những kiến thức bổ ích nhất liên quan đến quản trị website.

## Dịch vụ

---



**Cài đặt WordPress miễn phí**



**Cài đặt máy chủ**



**Thiết kế website WordPress**



**VPS SSD tốc độ cao**



**Tư vấn miễn phí**

- ★ **Các dịch vụ Shared Host tốt nhất**
- ★ **Các dịch vụ VPS tốt nhất**
- ★ **Nên mua theme WordPress ở đâu?**

## Liên kết

---

- ▶ [Trần Ngọc Chính](#)
- ▶ [IZWebz](#)
- ▶ [Vietdesigner](#)
- ▶ [Danny Nguyễn](#)
- ▶ [Ngân Sơn](#)
- ▶ [Văn Mỹ](#)
- ▶ [Lập trình Sen Việt](#)
- ▶ [Trung Đức](#)
- ▶ [Vinacode](#)
- ▶ [Giúp Bạn Làm Đẹp](#)



- ▶ [Lại Văn Đức](#)
- ▶ [Ninh Đôn](#)
- ▶ [Đinh Trang](#)

---

Xin vui lòng trích nguồn **Thach Pham Blog** khi phát hành lại nội dung trên website này.  
Giao diện thiết kế bởi [orange-themes.com](#)