 **kanamiki** @kanamiki


Theo dõi

★ 2.1K +82 49

Đã đăng vào thg 5 29, 2016 5:13 CH - 4 phút đọc

59.6K 2 53

Tập hợp những câu lệnh GIT hữu dụng

 Bài đăng này đã không được cập nhật trong 6 năm

Dưới đây là một vài ví dụ về các câu lệnh Git mà tôi thường dùng.

Git config

```
git config --global user.name "John Doe"
git config --global user.email "john@example.com"
```

--global được sử dụng để áp dụng cho tất cả các projects. Nếu bạn ko sử dụng --global thì settings sẽ chỉ dùng cho riêng project đó.

Giúp Git bỏ qua file modes

```
cd project/
git config core.filemode false
```

Câu lệnh trên hữu dụng khi chúng ta không cần quan tâm đến quyền truy cập files (ví dụ như khi sử dụng Windows).

Liệt kê những settings đang sử dụng

```
git config --list
```

Khởi tạo Git repo cho code có sẵn

```
cd existing-project/
git init
```

Clone một remote repo

```
git clone https://github.com/user/repository.git
```

Clone một remote repo tại thư mục hiện tại

```
git clone https://github.com/user/repository.git .
```

Xem thông tin trợ giúp cho một câu lệnh git

```
git help clone
```

Update và merge branch hiện tại với một remote repo

```
cd repo/  
git pull origin master
```

với `origin` là remote repo, `master` là remote branch.

Nếu bạn không muốn merge những thay đổi của bạn, hãy sử dụng `git fetch`

Liệt kê các remote urls

```
git remote -v
```

VIBLO

Tìm kiếm trên Viblo

🔍

👤

➔ Đăng nhập/Đăng ký

Thay đổi origin url

```
git remote set-url origin https://github.com/repo.git
```

Thêm remote repo

```
git remote add remote-name https://github.com/user/repo.git
```

Xem thay đổi (chưa đc add) của những file hiện tại

```
git diff
```

Xem thay đổi (đã được add, chưa commit)

```
git diff --cached
```

Xem thay đổi giữa local mà master

```
git diff origin/master
```

```
git diff COMMIT1_ID COMMIT2_ID
```

Xem những files thay đổi giữa hai commits

```
git diff --name-only COMMIT1_ID COMMIT2_ID
```

Xem những files thay đổi tại một commit bất kỳ

```
git diff-tree -no-commit-id --name-only -r COMMIT_ID
```

hoặc

```
git show --pretty="format:" --name-only COMMIT_ID
```

Xem thay đổi trước khi push

```
git diff --cached origin/master
```

Xem thông tin cụ thể của một commit

```
git show COMMIT_ID
```

Kiểm tra status của working tree

```
git status
```

Tạo vài thay đổi, rồi commit

```
git add changed_file.txt
git add folder-with-changed-files/
git commit -m "Committing changes"
```

Đổi tên/Di chuyển/Xoá files

```
git rm removeme.txt tmp/crap.txt
git mv file oldname.txt file_newname.txt
git commit -m "deleteing 2 files, renaming 1"
```

Đổi message của commit cuối

```
git commit --amend -m "New commit mesage"
```

Push local commits sang nhánh remote

Xem commit history

```
git log
```



Xem commit history cho hai commits gần nhất

```
git log -2
```



Xem commit history cho hai commits gần nhất, bao gồm cả thay đổi

```
git log -p -2
```



Xem commit history dưới dạng một dòng

```
git log --pretty=oneline
```



Revert một commit rồi push

```
git revert COMMIT_ID  
git push origin master
```



Revert đến thời điểm trước một commit

```
git reset COMMIT_ID  
git reset --soft HEAD@{1}  
git commit -m "Revert to COMMIT_ID"  
git reset --hard
```



Undo commit gần nhất, vẫn giữ thay đổi ở local

```
git reset --soft HEAD~1
```



Undo commit gần nhất, không giữ thay đổi ở local

```
git reset --hard HEAD~1
```



Undo commit gần nhất, vẫn giữ thay đổi ở index

```
git reset --mixed HEAD~1
```



hoặc

```
git reset HEAD~1
```



```
git reset origin/master
```



Reset về trạng thái của remote

```
git fetch origin  
git reset --hard origin/master
```



Xem các nhánh local

```
git branch
```



Xem tất cả các nhánh

```
git branch -a
```



Tạo một patch

```
git diff > patch-issue-1.patch
```



Thêm một file rồi tạo patch

```
git add newfile  
git diff --staged > patch-issue-2.patch
```



Thêm một file, thay đổi rồi tạo patch

```
git add newfile  
git diff HEAD > patch-issue-2.patch
```



Tạo patch từ một commit

```
git format-patch COMMIT_ID
```



Tạo patch từ hai commit cuối

```
git format-patch HEAD~2
```



Tạo patch từ tất cả những commits chưa push

```
git format-patch origin/master
```



```
git format-patch --binary --full-index origin/master
```

Apply một patch

```
git apply -v patch-name.patch
```

Apply một patch được tạo bằng format-patch

```
git am patch1.patch
```

Tạo một tag

```
git tag 7.x-1.3
```

Push một tag

```
git push origin 7.x-1.3
```

Tạo một nhánh

```
git checkout master
git branch new-branch-name
```

Lưu ý với hai câu lệnh trên thì chúng ta chưa chuyển sang nhánh mới, mà vẫn ở nhánh `master` . Phải sử dụng thêm `git checkout new-branch-name` để chuyển nhánh.

Ngoài ra có thể tạo nhánh mới và chuyển sang luôn bằng 1 câu `git checkout -b new-branch-name`

Chuyển nhánh

```
git checkout new-branch-name
```

Xem commit history so với branch hiện tại

```
git cherry -v master
```

`master` ở đây là branch mà bạn muốn so sánh

Merge commit từ branch khác

```
git checkout master
git merge branch-name
```

Ở đây chúng ta merge các commits của `branch-name` vào `master` .

Merge branch mà không commit

```
git merge branch-name --no-commit --no-ff
```

Xem thay đổi giữa state hiện tại và một branch

```
git diff branch-name
```

Xem thay đổi trong một file, giữa state hiện tại và một branch

```
git diff branch-name path/to/file
```

Xoá branch

```
git branch -d branch-name
```

Push lên một branch

```
git push origin branch-name
```

Lấy tất cả các branches

```
git fetch orgin
```

Lấy thư mục root

```
git rev-parse --show-toplevel
```

Xoá các file bị xoá ở local trên repo

```
git rm $(git ls-files --deleted)
```

Xoá toàn bộ các files chưa đc track

```
git clean -f
```

xoá cả folder:

```
git clean -f -d
```

xem các file trước khi xoá:

Unstage các files

```
git reset HEAD file.txt
```



Xem tag gần nhất

```
git describe --tags `git rev-list --tag --max-count=1`
```



Liệt kê các nhánh theo trình tự sử dụng gần nhất

```
git for-each-ref --sort=-committerdate refs/heads/ | head
```



Tar cả project, ngoại trừ thư mục .git

```
cd ..
tar cJf project.tar.xz project/ --exclude-vcs
```



Tar tất cả các files bị thay đổi ở local

```
git diff --name-only | xargs tar -cf project.tar -T -
```



Tìm conflict

```
grep -H -r "<<<" *
grep -H -r ">>>" *
grep -H -r '^=====$' *
```



Apply một patch không sử dụng git

```
patch < file.patch
```



Source: [The most useful git commands](#)

Git

All rights reserved



Bài viết liên quan

[Git Overview ...](#)

Trần Trọng Bình



↑ +46 ↓





Những lệnh Git cơ bản cần nhớ

[Nguyễn Thi Hang](#)
6 phút đọc
👁 74865 📖 37 💬 4 ⬆ 34

[Git] Bạn làm việc với Git vậy đã sử dụng Git Stash chưa?

[Le Xuan Duy](#)
0 phút đọc
👁 13958 📖 14 💬 3 ⬆ 24

10 vấn đề thường gặp của Git và cách khắc phục chúng

[Dung Ha](#)
10 phút đọc
👁 9274 📖 12 💬 1 ⬆ 9

Một số thủ thuật nhỏ với Git

[Nguyễn Duy Tá](#)
1 phút đọc
👁 1211 📖 4 💬 3 ⬆ 4

Bài viết khác từ kanamikiii

5 bước để từ ý tưởng thành sản phẩm

[kanamikiii](#)
8 phút đọc
👁 535 📖 1 💬 0 ⬆ 6

Phát triển phần mềm với Agile - Phần 2

[kanamikiii](#)
3 phút đọc
👁 315 📖 1 💬 0 ⬆ 3

Phát triển phần mềm với Agile - Phần 1

[kanamikiii](#)
7 phút đọc
👁 549 📖 0 💬 0 ⬆ 0

10 dự đoán về phần mềm năm 2022

[kanamikiii](#)
9 phút đọc
👁 308 📖 0 💬 0 ⬆ 0



Làm sao để trở thành Full Stack Developer

[kanamikiii](#)
8 phút đọc
👁 3274 📖 7 💬 0 ⬆ 7

Làm sao để tránh Async/Await Hell ?

[kanamikiii](#)
6 phút đọc
👁 1469 📖 10 💬 0 ⬆ 15

Sử dụng debugger của browser thay cho console.log().

[kanamikiii](#)
5 phút đọc
👁 1766 📖 5 💬 0 ⬆ 7

Các tình huống thường gặp khi sử dụng GIT

[kanamikiii](#)
5 phút đọc
👁 1903 📖 10 💬 0 ⬆ 7

Bình luận

💬 Đăng nhập để bình luận

[Phạm Huy](#) [@phhuy38](#)
thg 11 11, 2017 8:18 SA ✎

Nhiều lệnh hay quá. Cảm ơn bác.

^ 0 v | [Trả lời](#) [Chia sẻ](#) ⋮

[Hoàng vn](#) [@wiliamfeng](#)
thg 8 2, 2020 10:04 SA ✎

Tks bạn

^ 0 v | [Trả lời](#) [Chia sẻ](#) ⋮



↑ +46 ↓



TÀI NGUYÊN


[Bài viết](#)
[Câu hỏi](#)
[Videos](#)
[Thảo luận](#)
[Công cụ](#)
[Trạng thái hệ thống](#)

TỔ CHỨC


[Tags](#)
[Tác giả](#)
[Đề xuất hệ thống](#)
[Machine Learning](#)

DỊCH VỤ


[Viblo Code](#)

[Viblo CV](#)

[Viblo CTF](#)




[Viblo Learning](#)

ỨNG DỤNG DI ĐỘNG






LIÊN KẾT



Tiếng Việt

▼

[Về chúng tôi](#)[Phản hồi](#)[Giúp đỡ](#)[FAQs](#)[RSS](#)[Điều khoản](#)

 DMCA PROTECTED

© Viblo 2022