



Git Javascript PHP Docker

Software Testing Clean Code Agile

Subscribe

Scrum Tin tuyển dụng

GIT

Giới thiệu các config option cho git thường sử dụng



tox1ngau

21 Th06 2021 • 4 min read



git

Tiếp nối bài trước, trong bài này mình sẽ giới thiệu về cách cài đặt git cùng với các config hay sử dụng.

Trước tiên là công việc cài đặt git

.....

Trên máy tính windows, truy cập link [download git](#), tải và cài đặt bằng các bước bấm next.

Trên các máy Ubuntu, các bạn thực hiện câu lệnh

```
$ sudo apt install git
```

Trên các máy MacOS, các bạn thực hiện câu lệnh

```
$ brew install git
```

Bước tiếp theo là công việc cấu hình git

1. Cấu hình git lần đầu

Bây giờ Git đã có trên hệ thống, bạn muốn tùy chỉnh một số option cho Git của bạn. Bạn chỉ cần thực hiện các bước này một lần duy nhất; chúng sẽ được ghi nhớ qua các lần cập nhật, và cũng có thể thay đổi chúng bất kỳ lúc nào.

Khi config, các giá trị đó có thể được lưu ở ba vị trí khác nhau:

- `/etc/gitconfig` : Chứa giá trị cho tất cả người dùng và kho chứa trên hệ thống bằng cách sử dụng option `--system` khi chạy git config.
- `~/.gitconfig` riêng biệt cho tài khoản của bạn bằng

cách sử dụng option `--global`.

- `File config` trong thư mục `.git` (`.git/config`) của bất kỳ Repo của bạn, chỉ áp dụng riêng cho Repo đó.

2. Config danh tính của bạn

Việc đầu tiên bạn nên làm khi cấu hình Git là chỉ định tên và địa chỉ e-mail. Điều này rất quan trọng vì mỗi Git sẽ sử dụng chúng cho mỗi lần commit, những thông tin này được gắn bất di bất dịch vào các commit:

```
$ git config --global user.name "YOUR NAME"
$ git config --global user.email "YOUR@EMAIL.com"
```

với option `--global` config đã được lưu lại trong `~/.gitconfig` nên 2 hành động trên chỉ cần làm 1 lần.

3. Config trình soạn thảo editor

```
$ git config --global core.editor nano
```

Bạn có thể lựa chọn trình editor mà mình hay sử dụng để config vào git. Ở đây mình thường xuyên sử dụng nano nên mình config nano cho editor của mình.

Mẹo: Bạn có thể dùng câu lệnh `$ git config --list` để xem những cấu hình mà mình đã cài đặt.

4. ssh-key và github

SSH Keys là một phương thức xác thực đăng nhập với máy chủ qua SSH bằng việc đối chiếu giữa một cặp keys, bao gồm private key và public key tương ứng.

SSH Keys sử dụng giao thức xác thực challenge-response mà trong đó một bên trình bày một câu hỏi – challenge và một bên khác phải cung cấp một câu trả lời hợp lệ – response để được chứng thực. Có thể hiểu đơn giản, Private Key là chìa khóa còn Public Key là ổ khóa.

Tại sao phải sử dụng ssh-key

- Bảo mật các kết nối của máy tính với server.
Không phải nhập mật khẩu github mỗi lần pull hoặc push code.
- Bảo mật các kết nối của máy tính với server.
Không phải nhập mật khẩu github mỗi lần pull hoặc push code sẽ giảm sự bất tiện.

Generate SSH Key

```
$ ssh-keygen -t rsa -b 4096 -C "your_mail@example.com"
```

Khi thực hiện câu lệnh trên, để tiện lợi nhất cho sau này, các bạn nên để các cài đặt ở mặc định, ví dụ: ssh-agent hỏi bạn

muốn lưu key của mình ở đâu thì bạn hãy gõ Enter thôi. Ví dụ trong 2 yêu cầu dưới đây khi ssh-agent có hỏi:

```
Enter file in which to save the key (/root/.ssh/id_rsa): [Press enter]
```

```
Enter passphrase (empty for no passphrase): [Type a passphrase]  
# Enter same passphrase again: [Type passphrase again]
```

Lưu ý: mật khẩu khi bạn gõ vào nó sẽ không hiển thị mấy dấu *** như bình thường, nhưng bạn cứ gõ xong rồi Enter thôi. Còn trong trường hợp của mình thì mình enter, không nhập gì cả.

Sau khi hoàn thành các bước trên ssh-key đã được tạo theo thuật toán mã hóa RSA, độ dài 4096bit và lưu tại `~/.ssh/`. Trong đó, Private Key là `id_rsa` còn Public Key là `id_rsa.pub`.

Thêm ssh-key vào github

Bước tiếp theo là lấy key vừa tạo và add vào Github.

Xem public key bằng câu lệnh:

```
$ cat ~/.ssh/id_rsa.pub
```

Pub key có dạng:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDm+Dk0MXXQwJwxEBekcfeHePkBvhUS
```

Sau đó copy và paste vào mục **Settings key** của github. Link của mục settings key [tại đây](#).

Kiểm tra kết nối

bằng câu lệnh:

```
$ ssh -T git@github.com
```

Khi có thông báo

```
Hi your_github_username! You've successfully authenticated, but GitH
```

là bạn đã kết nối thành công.

Trên đây mình đã giới thiệu cách cài đặt git cùng với các config hay sử dụng. Trong bài tiếp theo của series này mình sẽ tiếp tục đưa thêm cách nhìn, khái niệm cơ bản của git như là Repository, Snapshot, Commit, Remote, Branch, Push, Pull. ([Link bài tiếp theo tại đây](#))

Ngoài ra, các bạn có thể xem lại bài [Giới thiệu về git](#) để củng cố thêm kiến thức về git.

0 comments

Sort by **Oldest**



Add a comment...

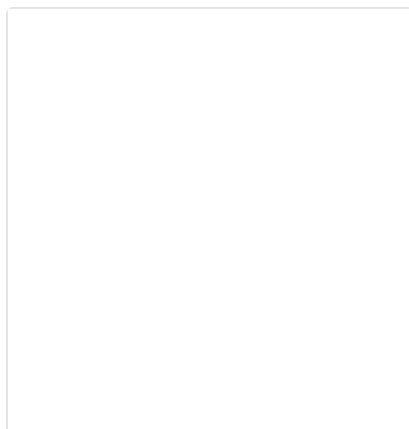


Facebook Comments Plugin

Sign up for more like this.

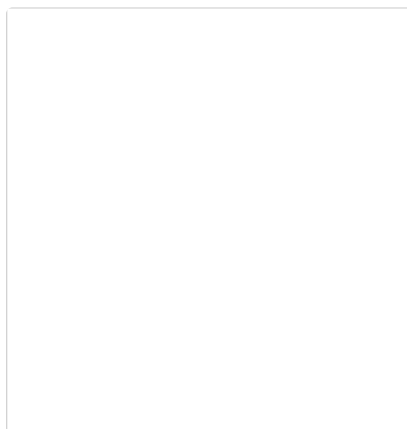
Enter your email

Subscribe



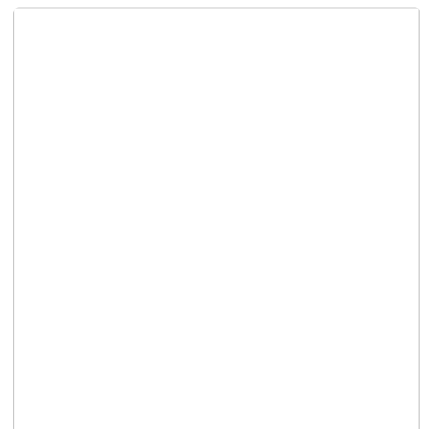
Cách deploy Nextjs app lên server

Bạn có 1 trang web NextJs, bạn để nó ở local,



Các bước áp dụng tiêu chuẩn ISO 9001 & ISO 27001 vào trong doanh nghiệp

Hiện nay việc áp dụng



Redux là gì? Giới thiệu về Redux cơ bản

Giới thiệu Redux khá là phổ biến. Tuy nhiên

nhưng bạn muốn deploy



vudo

12 Th12 2022 • 2 min
read

Hiện nay, việc áp dụng
tiêu chuẩn ISO 9001 &
ISO 27001 trong doanh



Ninhdtd

9 Th12 2022 • 6 min
read

không phải tất cả chúng



Nguyễn Anh Toàn

5 Th12 2022 • 5 min
read

Hapolog © 2022

Contact

Powered by Ghost