

TỰ ĐỘNG HÓA QUẢN TRỊ HỆ THỐNG LINUX

web: blog.tapchilinux.com

mail: tapchilinux@gmail.com

Công cụ

- Ansible
- Chef
-
- Puppet
- Saltstack

Lựa chọn



ANSIBLE

Ansible

- Phần 1: Giới thiệu, cài đặt, cấu hình ansible
- Phần 2: Cấu trúc thư mục ansible playbooks
- Phần 3: Triển khai WordPress bằng Ansible
- Phần 4: Quản lý dữ liệu được mã hóa với Vault

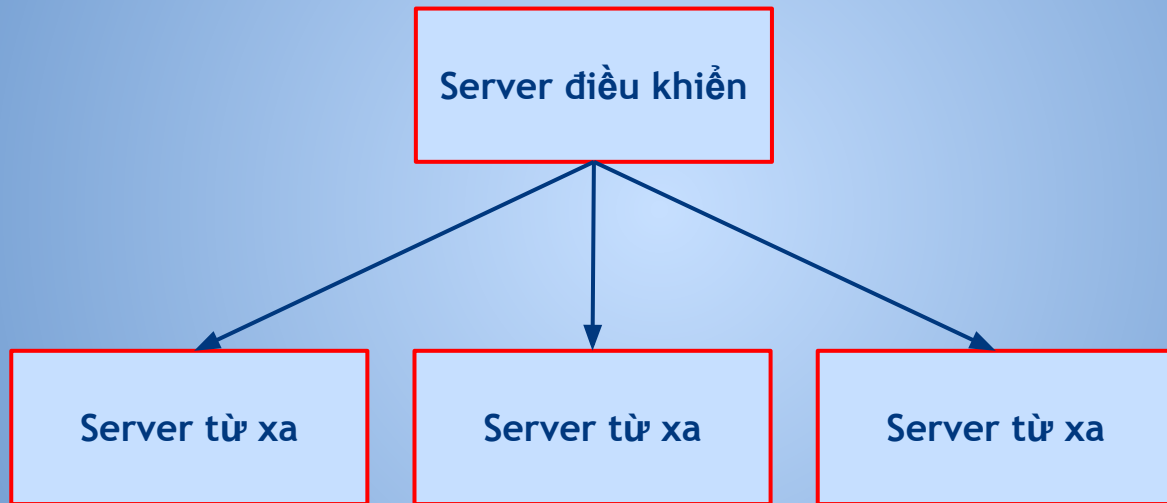
Phần 1: Giới thiệu ansible

- Công cụ mã nguồn mở
- Quản lý, cài đặt, cấu hình hệ thống tự động
- Chỉ cần sử dụng SSH + Python 2.4 (hoặc mới hơn)
- Quản lý nhiều tác vụ với YAML
- Dễ học, dễ cấu hình, dễ sử dụng

Phần 1: Yêu cầu hệ thống

- OS: RHEL/CentOS/Fedora/Ubuntu/Debian ...
- Jinja2: Là template engine cho Python, hiện đại, nhanh và dễ sử dụng
- PyYAML: Là thư viện phân tích YAML cho Python
- paramiko: Là thư viện SSHv2 cho Python
- httplib2: Là thư viện HTTP client
- sshpass: Là thư viện xác thực password ssh

Phần 1: Mô hình cài đặt demo



Phần 1: Cài đặt ansible

- Đối với Ubuntu

\$ sudo apt-add-repository ppa:ansible/ansible -y

\$ sudo apt-get update && sudo apt-get install ansible -y

- Đối với Centos

\$ Cài EPEL repo

\$ sudo yum install ansible -y

- Check lại việc cài đặt

\$ ansible --version

- Sinh cặp khóa key-pair

ssh-keygen -t rsa -b 4096 -C "admin@teamsvr.vn"

Phần 1: Cài đặt ansible (tiếp)

- Copy public key đến các server từ xa

```
# ssh-copy-id hungld@10.3.17.52
```

```
# ssh-copy-id hungld@10.3.17.53
```

```
# ssh-copy-id hungld@10.3.17.54
```

- Check lại xem việc ssh đã ok chưa

```
# ssh hungld@10.3.17.52
```

```
# ssh hungld@10.3.17.53
```

```
# ssh hungld@10.3.17.54
```

- Tạo file hosts để quản lý các server từ xa

```
# vi /etc/ansible/hosts
```

```
[test-servers]
```

```
10.3.17.52
```

```
10.3.17.53
```

```
10.3.17.54
```

Phần 1: Chạy qua dòng lệnh

- Chạy lệnh ping:

ansible -m ping test-servers

ansible -m ping all

ansible -i /etc/ansible/hosts -m ping all

- Chạy lệnh khác

ansible -m command -a "df -h" test-servers

ansible -i /etc/ansible/hosts -m command -a "df -h" test-servers

ansible -i /etc/ansible/hosts -m command -a "df -h" all

- Chú ý các option

all: apply cho toàn bộ các server

-m ping: sử dụng module "ping"

-s: sử dụng quyền sudo khi chạy lệnh

-k: ưu tiên sử dụng password hơn là dùng key-pair xác thực server

-u: User của server

Phần 2: Cấu trúc ansible playbooks

- Như đã nói ở trên Playbooks có thể chạy nhiều tác vụ và cung cấp một số tính năng cao cấp hơn mà chúng ta sẽ bỏ lỡ sử dụng lệnh như vừa demo.
- Do vậy chúng ta chủ yếu sẽ làm việc với ansible playbooks
- Trong ansible playbooks đều sử dụng Yaml

Phần 2: Cấu trúc playbooks

production	# file quản lý các server chạy chính thức
development	# file quản lý các server đang phát triển
group_vars/	
group1	# khai báo các biến dùng chung cho các group trong playbooks
group2	# tương tự (trong file quản lý server có host nào thì khai ở đây)
host_vars/	
hostname1	# nếu hệ thống cần khai báo các biến riêng cho từng host
hostname2	# tương tự
library/	# nếu mình cần code ra module riêng thì đặt ở đây (optional)
filter_plugins/	# nếu mình cần code plugin filler thì đặt ở đây (optional)
site.yml	# file chính của playbooks
webservers.yml	# file playbooks cho webserver
dbservers.yml	# file playbooks cho dbserver
roles/	# lưu trữ các role
common/	
webserver/	

Phần 2: Cấu trúc role

roles/

common/ # role có tên là common

tasks/ #

main.yml # <-- file này để thực thi các task vụ mà người quản trị cần làm

handlers/ #

main.yml # <-- file điều khiển khi có yêu cầu từ task gọi đến

templates/ # <-- lưu trữ các file dạng template (có thể truyền dữ liệu qua biến)

ntp.conf.j2 # <-- đây là file template

files/ #

bar.tar # <-- là file .tar

foo.sh # <-- là file script

vars/ #

main.yml # <-- khai báo các biến cho role này

defaults/ #

main.yml # <-- khai báo các biến mặc định (độ ưu tiên thấp hơn trong role)

meta/ #

main.yml # <-- Khai báo các role liên quan

Phần 3: Triển khai WordPress

- Source code Wordpress
- Web server
- Database server
- Các gói khác: PHP, ...

Phần 3: Triển khai WordPress

hosts

group_vars/

all

site.yml

roles/

common/

mysql/

nginx/

php-fpm/

wordpress/

Phần 3: Triển khai WordPress

```
$ ansible-playbook -i hosts site.yml
```

```
$ ansible-playbook -i hosts site.yml -limit '10.3.17.56'
```

```
$ ansible-playbook -i hosts site.yml --tags=install
```

```
$ ansible-playbook -i hosts site.yml --tags=config
```

```
$ ansible-playbook -i hosts site.yml --tags=iptables --limit  
'192.168.2.82,192.168.6.48'
```

```
$ ansible-playbook -i hosts site.yml --tags=iptables --limit=web
```


Phần 4: QL dữ liệu được mã hóa với Vault

- Nhiều dữ liệu riêng (password, key, ...) lưu trong templates, trong file, trong biến --> không an toàn
- Vault cho phép bạn mã hóa bất kỳ tập nào dạng YAML, thường là các file trong biến (vars, group_vars, host_vars).
- Vault không mã hóa Files và templates
- Khi tạo một file mã hóa thì nó sẽ yêu cầu mật khẩu. Mật khẩu này sẽ được sử dụng để edit, hoặc khi roles, playbooks gọi

Phần 4: QL dữ liệu được mã hóa với Vault

- VD:

\$ ansible-vault create vars/main.yml

\$ Vault Password:

\$ Confirm Vault password:

- Các option trong Vault

+ create - Tạo 1 file và mã hóa nó

+ decrypt - Tạo file không mã hóa từ file mã hóa

+ edit - Edit và luôn mã hóa file

+ encrypt - Mã hóa 1 file từ file được chưa mã hóa

+ rekey - Sét lại password cho file mã hóa

- Chạy playbooks

\$ ansible-playbook --ask-vault-pass site.yml

THANKS YOU