



Packer build image 起手式

Daniel Lin

<https://ppt.cc/fBI2Fx>

Who am I

- Study-Area Members
- openSUSE Members
- 塵世中迷糊的便當搬運工，兼任打雜
- GitHub: [Packer_Demo](#)

Agenda

- Pre required Install Packages
- Pre required Knowledge
 - Kickstart file
 - Virtualbox
- Packer
 - Intro Packer
 - Why I choice Packer ?
 - Advantages of Packer
- Ansible
- Vagrant

Pre required Install Packages

套件	查詢指令	版本
Packer	packer version	Packer v1.4.0
Vagrant	vagrant version	Installed Version: 2.2.5
VirtualBox	VBoxManage --version	6.0.6r130049
Ansible	ansible --version	ansible 2.8.1

Pre required Knowledge - Kickstart File

Setup Static IP:

- network

- --bootproto="static" # 使用靜態 IP
- --device="Device" # 設定自己的網路卡裝置名稱
- --gateway="Gateway" # 設定 Gateway
- --ip="IP" # 設定你自己的 IP
- --nameserver="DNS" # 設定解析伺服器
- --netmask="Netmask" # 設定網路遮罩
- --noipv6 # 不啟用 IPv6
- --activate # 啟用連線

Pre required Knowledge - Kickstart File

Setup HostName:

- network
 - --hostname=baseimage # 設定主機名稱

Pre required Knowledge - Kickstart File

安裝完成後，可以自己定義後續想做的事情，如：新增使用者、安裝必要套件等。

- **%post - %end** # 這二個關鍵字為一組，請把要執行的動作寫在裡面
- 『#』字號表示為註解

Pre required Knowledge - Kickstart File

%post

Install SUDO

/usr/bin/yum -y install sudo

Create packer user

/usr/sbin/useradd -m packer

/usr/bin/echo packer | /usr/bin/passwd --stdin packer

Add packere user to SUDO

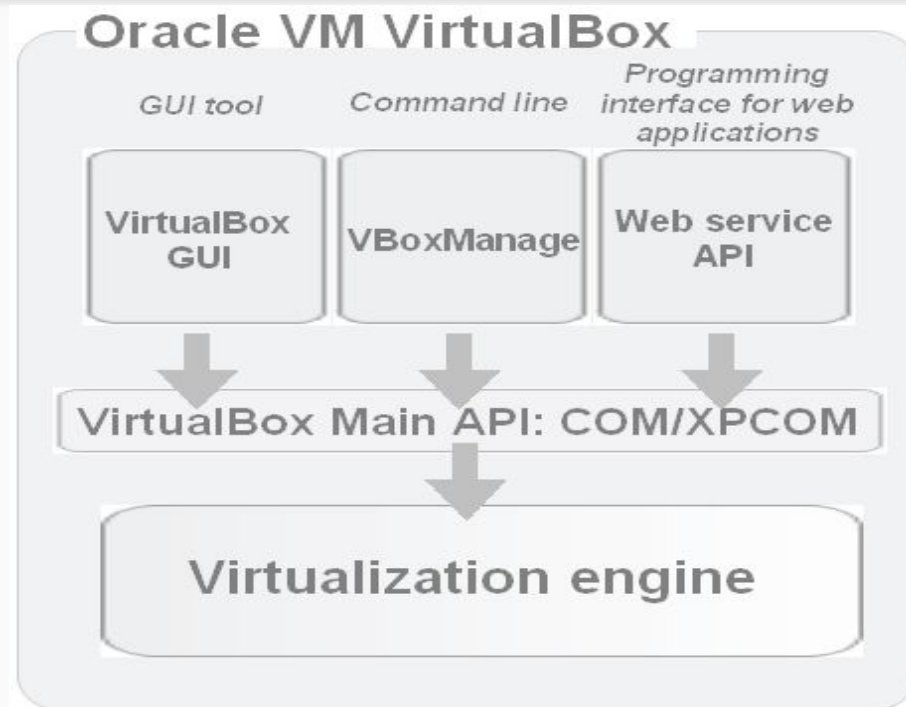
echo "packer ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers.d/packer

echo "Defaults:packer !requiretty" >> /etc/sudoers.d/packer

chmod 0440 /etc/sudoers.d/packer

%end

Virtualbox - Architecture



Virtualbox - CLI

列出所有 VM

- VBoxManage list vms

列出正在執行中的 VM

- VBoxManage list runningvms

列出支援的作業系統

- VBoxManage list ostypes

顯示 VM 的所有資訊

- VBoxManage showvminfo <uuid | vmname>

Virtualbox - CLI

啟動 VM

- `VBoxManage startvm <uuid | vmname>`

修改 VM 的 CPU 與 Memory

- `VBoxManage modifyvm <uuid | vmname> --cpus 2 --memory 2048`

關機 VM

- `VBoxManage controlvm <uuid | vmname> poweroff`
- More [Virtulbox User Manual Document](#)

Virtualbox - Debug

Q: VirtualBox is already locked by a session (or being locked or unlocked)

Ans: `VBoxManage startvm <uuid | vmname> --type emergencystop`

Reference:

- [VM in virtualbox is already locked for a session \(or being unlocked\)](#)

Intro Packer

Packer 是 HashiCorp 這家公司開源出來的一個打包映像檔的工具，主要是用來建立獨立的機器映像檔，可以從一個設定檔為多個平台建立映像檔，使用，如：AMIs for EC2、VMDK/VMX files for VMware、OVF exports for Virtualbox 等。

Packer 是一個很輕量化、效能也不錯的工具，可以安裝在主要常見的作業系統。



HashiCorp

Packer

Why I choice Packer ?

- 做映像檔很痛苦，不能簡單一點嗎？！
- 怎麼每個平台做的 Base Imags 格式都不同啊！
- 現在還要搞雲端，有沒有簡單一點的？



Advantages of Packer

- 快速部署基礎架構
- 多平台的高移值性
- 提高穩定性
- 更好的可測試性



Create Packer Directory

tree Packer_Demo

Packer_Demo

- |— [http/](#) → 此目錄是存放 kickstart file 的地方
- |— [output/](#) → 此目錄會產生 Virtualbox box image 的地方
- |— [template.json](#) → 簡單的 Packer 範本

2 directories, 1 file

Packer Templates Structure

Packer 是用 **JSON file** 來做為 Template, 常用的參數如下:

- builders (**require**)
- variables (option)
- provisioners (option)
- post-processors (option)
- [more information](#)

Packer CLI

查看 template.json 結構

- # packer inspect template.json

驗證 template.json 語法

- # packer validate template.json

打包 Virtualbox image

- # PACKER_LOG=1 packer build template.json

Builder Parameter

- builder
 - type
 - iso_url (required)
 - iso_checksum (required)
 - iso_checksum_type (required)
 - ssh_username (optional)
 - shutdown_command (optional)

Builder Parameter

- builder
 - vm_name (optional)
 - guest_os_type (optional)
 - disk_size (optional)
 - http_directory (optional)
 - boot_wait (optional)
 - boot_command (optional)
 - floppy_files (optional)

VBoxManage Command

- vboxmanage
 - --cpus
 - --memory
 - --audio
 - --usb

VBoxManage Command

- vboxmanage
 - --nic1
 - --nictype1
 - --cableconnected1
 - --nictrace1
 - --bridgeadapter1
 - --macaddress1

Post-Processors Parameter

- post-processors
 - type
 - override
 - virtualbox
 - output

Packer Debug

Q: Build 'virtualbox-iso' errored: Output directory exists: output-virtualbox-iso

Ans: Use the force flag to delete it prior to building.

Reference:

- Packer Github [issue 163](#)

Intro Ansible

Ansible 是一種通用的語言，
將棘手的任務，變成可以重覆
使用的 Playbook，從此你的
生命因而變的更快樂。

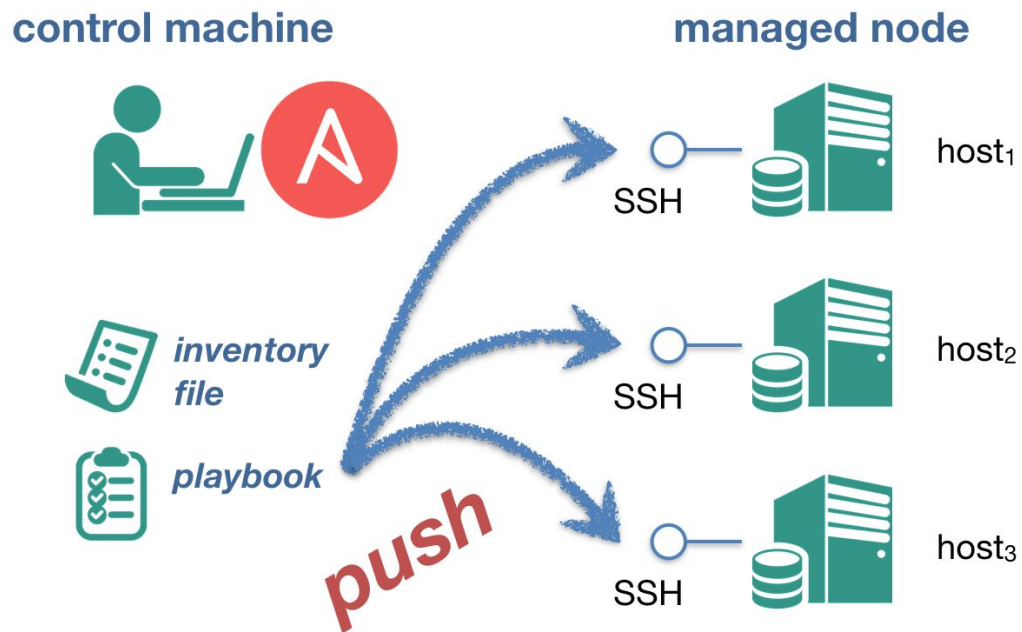


ANSIBLE

Advantages of Ansible

- 使用 YAML 檔設定
- 無需在 Remote Server 上安裝 Agent
- 可讀性高

Ansible 運作示意圖



Ansible Directory

ansible

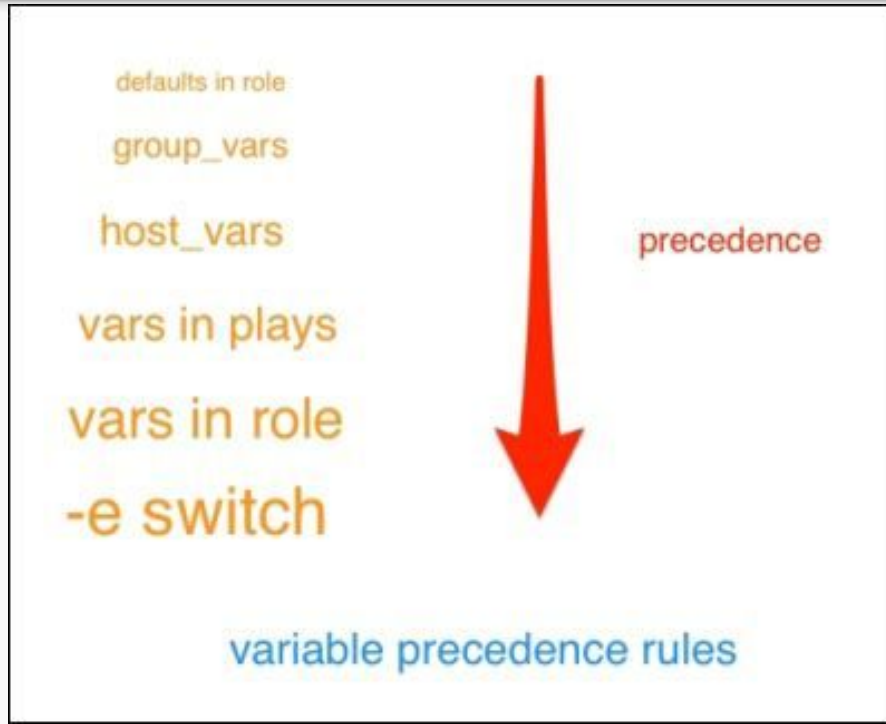
- ├── ansible.cfg → Ansible 主要的設定檔
- ├── demo.yml → Playbook 即是要執行的動作
- ├── inventory → 存放遠端伺服器的資訊
- └── [roles/](#) → 把 Playbook 模組化的資料夾

1 directory, 3 files

Ansible Roles

- demo
 - README.md → 需要用 markdown 格式寫的文件
 - **defaults** → 定義預設變數, 優先級是最低
 - **files** → 可以放檔案, 如: 設定檔
 - **handlers** → 用來觸發服務的狀態, 如: 重啟, 停止服務
 - meta → 定義 Role 的依賴性
 - **tasks** → 大部份要執行的工作, 都在寫在這
 - **templates** → 可以用 [Jinja2](#) 來寫範本
 - tests → 給你寫簡單的測試用, 需要有 inventory & test.yml
 - **vars** → 定義內部變數使用

Ansible Variable Precedence



Ansible Configuration

ansible.cfg

- [defaults]
 - inventory = ./inventory
 - roles_path = ./roles
 - host_key_checking = False

Ansible Inventory

- [all]
- ansible_host
- ansible_port
- ansible_user
- ansible_ssh_pass

Ansible Playbook

- hosts: all

gather_facts: no

roles:

- demo

Ansible CLI

語法檢查

- `# ansible-playbook --syntax-check demo.yml`

執行 playbook

- `# ansible-playbook -i inventory demo.yml -k`
 - `-i` inventory file
 - `-k` ask for connection password

Intro Vagrant

Vagrant 是一個建置和管理虛擬機環境的工具，通過易於使用的工作流程並專注於自動化，大大降低了開發環境的建置時間，提高了生產價值。



Advantages of Vagrant

- Ruby DSL 語法
- 易於配置
- 可重複和便攜的工作環境
- 一致性的工作流程
- 提高生產力和靈活性

Vagrantfile

這三個設定是 Vagrantfile 中最重要的

- `Vagrant.configure("2") do |config| ~ end`
 - `config.vm.provides :virtualbox do |v| ~ end`
 - `config.vm.provision :ansible do |ansible| ~ end`

Vagrantfile - config.vm

更新 Guest VM 的 /etc/hosts

- `config.vm.hostname = "base-centos7"`

使用 Virtualbox Box Image

- `config.vm.box = "CentOS"`

設定網路

- `config.vm.network "public_network", bridge: "en1: Wi-Fi (AirPort)"`

設定本機資料夾要同步到 Guest VM 的資料夾

- `config.vm.synced_folder '.', '/vagrant', disabled: true`

Vagrantfile - config.vm

設定要 Guest VM 要先做什麼事

- `config.vm.provision "shell", inline: "echo Hello, World"`

more information about [config.vm](#)

Vagrantfile - config.ssh

設定 Guest VM 的 IP

- `config.ssh.host = "Your IP Address"`

設定連接 Guest VM 的帳號, 預設是使用 **vagrant**

- `config.ssh.username = "UserName"`

設定連接 Guest VM 的密碼

- `config.ssh.password = "Password"`

Vagrantfile - config.ssh

是否要取代 Guest VM 的 SSH 密鑰, 預設是 True

- `config.ssh.insert_key = "false"`

more information about [config.ssh](#)

Vagrantfile - Ansible Provisioner

```
Vagrant.configure("2") do |config|  
  # Run Ansible from the Vagrant Host  
  config.vm.provision "ansible" do |ansible|  
    ansible.limit = "all"  
    ansible.inventory_path = "./ansible/inventory"  
    ansible.playbook = "./ansible/demo.yml"  
    ansible.host_key_checking = "false"  
    ansible.force_remote_user = "root"  
  end  
end
```

more information for [Ansible Provisioner](#)

Vagrantfile - Provider Configuration

```
Vagrant.configure("2") do |config|
  # Modify VirtualBox Parameter.
  config.vm.provider :virtualbox do |vb|
    vb.gui = true
    vb.cpus = 1
    vb.memory = 1024
    vb.customize ["modifyvm", :id, "--nic1", "bridged"]
    vb.customize ["modifyvm", :id, "--nictype1", "82540EM"]
    vb.customize ["modifyvm", :id, "--cableconnected1", "on"]
    vb.customize ["modifyvm", :id, "--nictrace1", "off"]
    vb.customize ["modifyvm", :id, "--bridgeadapter1", "en1: Wi-Fi (AirPort)"]
    vb.customize ["modifyvm", :id, "--macaddress1", "auto"]
    vb.customize ["modifyvm", :id, "--usb", "off"]
    vb.customize ["modifyvm", :id, "--audio", "none"]
  end
end
```

more information for [Provider Configuration](#)

Vagrant CLI

加入 Vagrant Box

- `# vagrant box add -name <name> /Virutalbox/output/vm.box`

列出 Vagrant Box

- `# vagrant box list`

刪除 Vagrant Box

- `# vagrant box remove <name>`

Vagrant CLI

啟動 VM

- # vagrant up

關機 VM

- # vagrant halt

刪除 VM

- # vagrant destroy <name> -f

登入 VM

- # vagrant ssh

參考來源

- [Ansible 自動化部署工具](#)
- [Ansible 實戰講堂](#)



Image credits : http://images.geronimo.com/wp-content/uploads/2018/12/happy_guy_at_pc.jpg

Q & A