

淺談 Ansible 自動化管理工具

黃俊宏 sakana / Max

<https://bit.ly/2KNaizj>

工商服務

- openSUSE x StudyArea
 - StudyArea
 - Facebook <https://www.facebook.com/sataiwan/>
 - openSUSE Taiwan
 - Facebook <https://www.facebook.com/groups/opensuse.tw/>
 - openSUSE KKTIX <http://opensuse-tw.kktix.cc/>
 - 2016/9月開始舉辦一連串workshop
 - Ansible workshop
 - Docker workshop
- DevOps Taiwan
 - Facebook <https://www.facebook.com/groups/DevOpsTaiwan/>
- Ansible Taiwan
 - <http://ansible.tw/#!index.md>

What is openSUSE project?

openSUSE是一個由SUSE所贊助的社群。

以推廣Linux 為宗旨，openSUSE.org計畫讓每個人可以免費而且輕易的使用世界上最好用的Linux 發行版本--SUSE Linux。同時，openSUSE也提供Linux開發人員和其支持者使用Linux所需的一切。

openSUSE計畫的目標：

- 打造SUSE Linux 成為最容易取得和使用的Linux發行版本，並成為最廣泛使用的開放原始碼平台。
- 提供良好的合作環境使SUSE Linux不論對新手老手而言都是世界上最好的Linux發行版本。
- 藉由大幅度地簡化和開放開發過程使 SUSE Linux成為程式設計師和Linux駭客們最喜愛的工作平台。

openSUSE 特色

- YaST

- openSUSE提供YaST(Yet another Setup Tool)實現方便的系統管理。YaST的功能強大,幾乎可以配置系統的各個方面:包括軟體管理、硬體設置、網路設置、系統服務、安全設置等諸多功能。這些功能的引入,大大簡化了配置系統的過程。

- OBS

- Build Service編譯服務,提供了openSUSE、Fedora、Ubuntu、Archlinux ...的全面編譯支援。也為openSUSE帶來了數不清的應用程式套件。
- <https://software.opensuse.org/search>

- openSUSE Tumbleweed

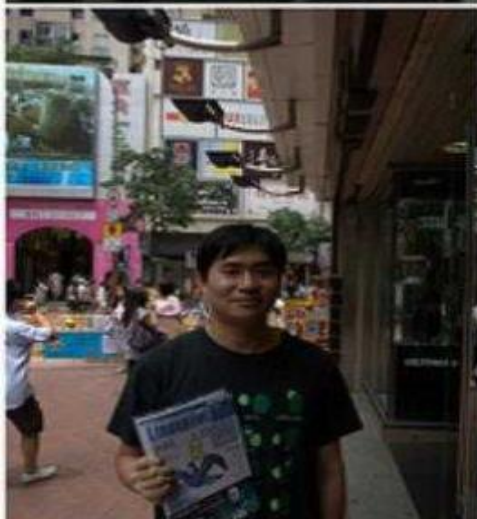
- 新的Tumbleweed 計劃旨在提供一個「滾動升級」的openSUSE 版本,滾動升級的版本即不再像openSUSE的常規版本(Leap)受限於版本開發周期,可以始終使用最新穩定版本的全部軟體。該計劃適合想使用最新的,又是穩定版的軟體的openSUSE 用戶。

Agenda

- Ansible ? Why Ansible ?
- 預備知識
- 安裝 Ansible
- The role of Ansible
- 與 shell script 的差異??
- Hello Ansible!! 第一個 ansible 指令
- Inventory file / ansible.cfg 介紹
- Ansible Module 介紹與實作
- Playbook 介紹與實做
- Reference

Who is Max?

- Study Area 小雑工
- GNOME Foundation Member
- GNOME.Asia Committee Member
- openSUSE members



About Ansible

- Ansible - configuration management tool 組態管理工具
 - 官網 <http://www.ansible.com/>
 - GitHub <https://github.com/ansible/ansible>
- Online resource 線上資源
 - Documentation 文件
 - <http://docs.ansible.com/>
 - Ansible Project Google Group 討論群組
 - <https://groups.google.com/forum/#!forum/ansible-project>
 - IRC 聊天室
 - <https://webchat.freenode.net/>
 - #ansible
 - Ansible Galaxy
 - <https://galaxy.ansible.com/>
 - 社群共享 Role 集散地

Why Ansible

- Infrastructure as Code
- 語法簡單(play book 以YAML 語法撰寫)
- 不需要安裝client (clientless)
- Push-based
 - Pull-based: Agent check to server by time. (Chef / Puppet by default)
 - Push-based: Server push change by order.
 - 你可以決定何時進行設定
- Very thin layer of abstraction 以原有的習慣進行部署
- 各個公有雲可能有相關工具, 但是我還有很多東西要一起弄 ... 我可以1個工具解決嗎?
- Loop?



預備知識？

- 透過 **SSH** 連接遠端主機
- 使用命令列操作 (pipes and redirection).
- 安裝套件.
- 使用 **sudo** 指令.
- 管理檔案權限.
- 管理 services .
- 了解 / 設定環境變數.
- 寫 scripts (Any language)

安裝 Ansible

- http://docs.ansible.com/ansible/intro_installation.html
- openSUSE
 - `# zypper install ansible`
- Mac
 - `$ sudo easy_install pip`
 - `$ sudo pip install ansible`
- CentOS 7
 - `$ sudo yum install epel-release -y`
 - `$ sudo yum install ansible`
- Ubuntu
 - `$ sudo apt-get install ansible`

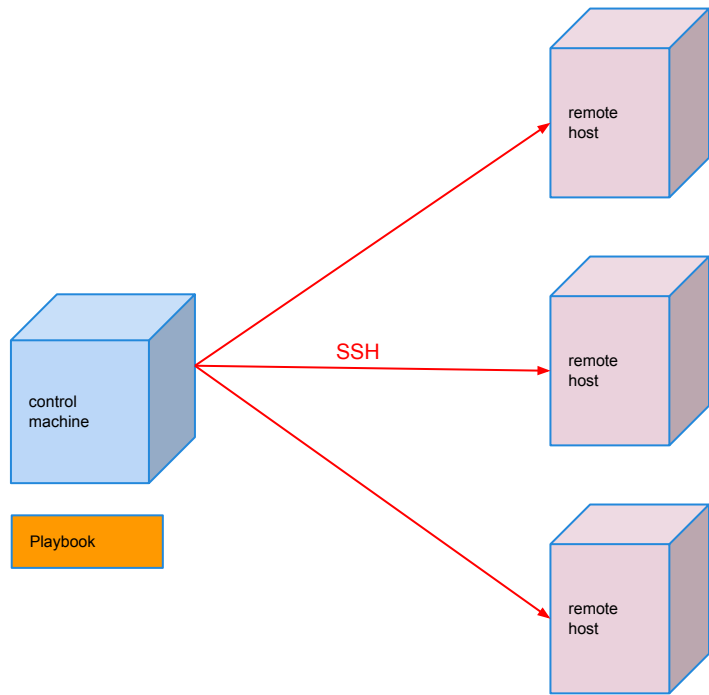
The role of Ansible

Control Machine 控制端

- 用於控制部署遠端的機器.
- 需要 python 2.6 以上版本.

Managed Node 遠端主機

- 一個或是多個遠端主機
- 需要 python 2.5 以上版本.



與shell script 的差異??

- 不用登入遠端去抓 shell script 與 執行 (中央集權)
- 針對所有主機同時進行, 按照順序執行任務
- 有許多現成的 Module 可用, 用法習慣跟原系統差不多
- 語法簡單, 容易上手(YAML)
- 可以利用 fact 與變數執行 loop 或是其他的做法.
- 有別人寫好的 Role 可以參考與套用 (Ansible-galaxy)

Hello Ansible! 第一個 ansible 指令

實驗環境

<http://play-with-docker.com>

本機 docker

```
# docker pull sakana/jupyterhub:20180527
```

```
# docker pull sakana/sshd_opensuse423
```

```
#docker run -d -p 8000:8000  
sakana/jupyterhub:20180527
```

```
#docker run -d -p 8022:22  
sakana/sshd_opensuse423
```

- JupyterHub
 - 帳號: **ansible** 密碼: 2016StudyArea
- 切換目錄到
 - **LearnJupyter-master/Notebook/20180527SA**
- 請修改 playbook目錄下 hosts
 - 填入 遠端 IP於 **ansible_host=**
 - 填入 遠端 port 於 **ansible_port=**
 - 填入 遠端 ssh 密碼 於 **ansible_ssh_pass=**
 - **openSUSE**
 - 跟預設不同, 請手動修改
 - 記得 File -- > **Save**

Hello Ansible! 第一個 ansible 指令

Hands on Lab

- 方式 1:
 - 開啓目錄下 Ansible_openSUSE_20180527.ipynb
 - 按 shift + Enter 執行 block
 - 嘗試執行第一個 ansible 指令
 - ansible 對象 -m ping
- 方式 2:
 - New -- > Terminal
 - 切換到 playbook 目錄下手動執行指令

Inventory file

- Ansible lists hosts in text files, called inventory files.
- 將遠端主機相關資訊以文字檔案的方式建立稱為 inventory file, 常見的檔案名稱為 hosts
- 語法 servername options
- 可以設定群組 [群組名稱] 來組織對象
- 常用選項
 - ansible_host -- Remote Host IP
 - ansible_user -- Remote SSH User Name
 - ansible_port -- ssh port
 - ansible_ssh_private_key_file -- SSH Key
 - ansible_ssh_pass -- ssh password
- 如果有定義到 ansible.cfg 的 [defaults] 就可以不列出

ansible.cfg

可以設定一些預設行為, 不需要逐一設定在 hosts

[defaults]

hostfile -- 主機 ip 對照

hostfile 即將棄用, 可以使用 inventory = hosts (看 ansible 的版本)

hostfile = hosts

remote_user -- 遠端使用者名稱

remote_user = root

private_key_file -- SSH private key path

host_key_checking -- 不詢問加入 ssh 金鑰

host_key_checking = False

設定 retry files (*.retry) 存放路徑, 預設放家目錄

我自己喜歡指定在目前目錄, 以免作完實驗家目錄一堆 .retry

retry_files_save_path = ./ansible-retry

平行處理數量, 預設是 5 個, 應該不一定會用到先記下來

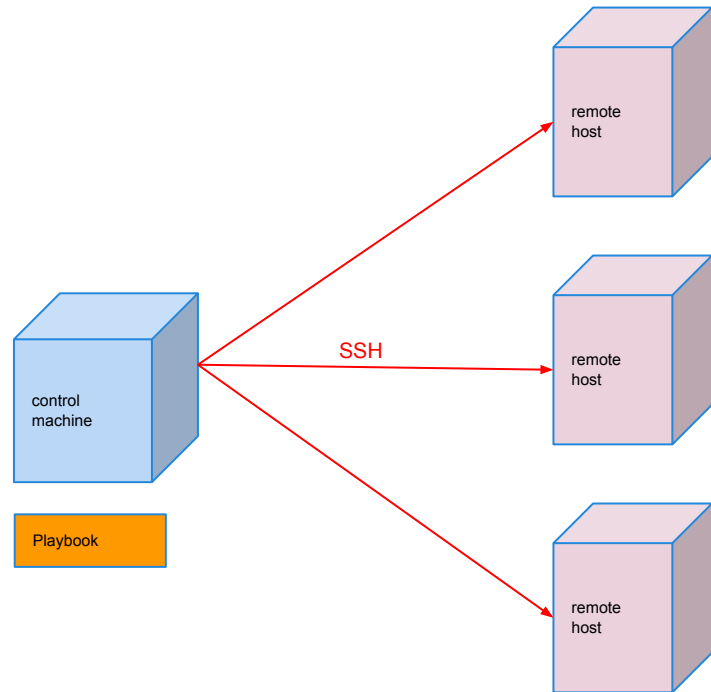
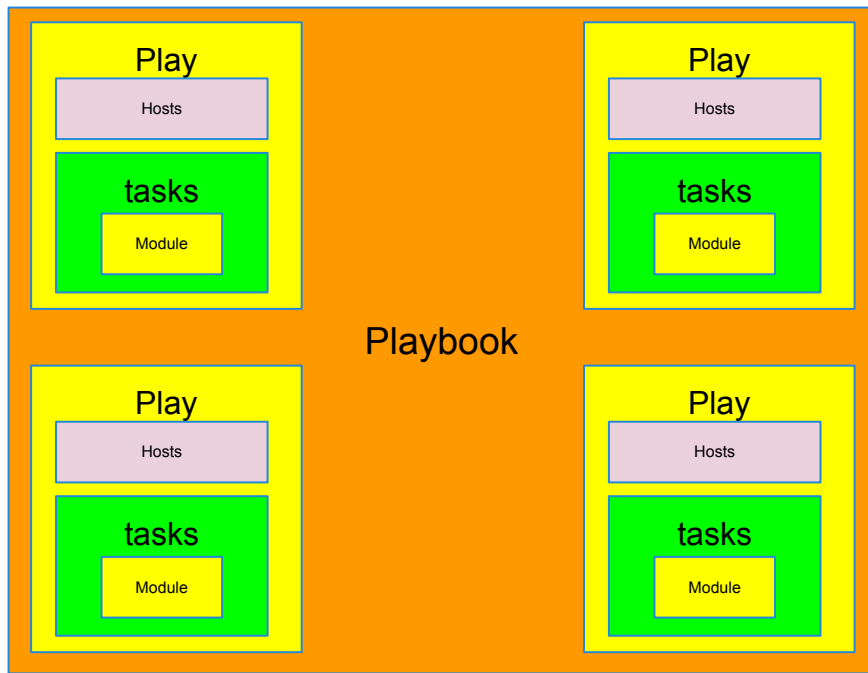
forks = 20

Ansible Module

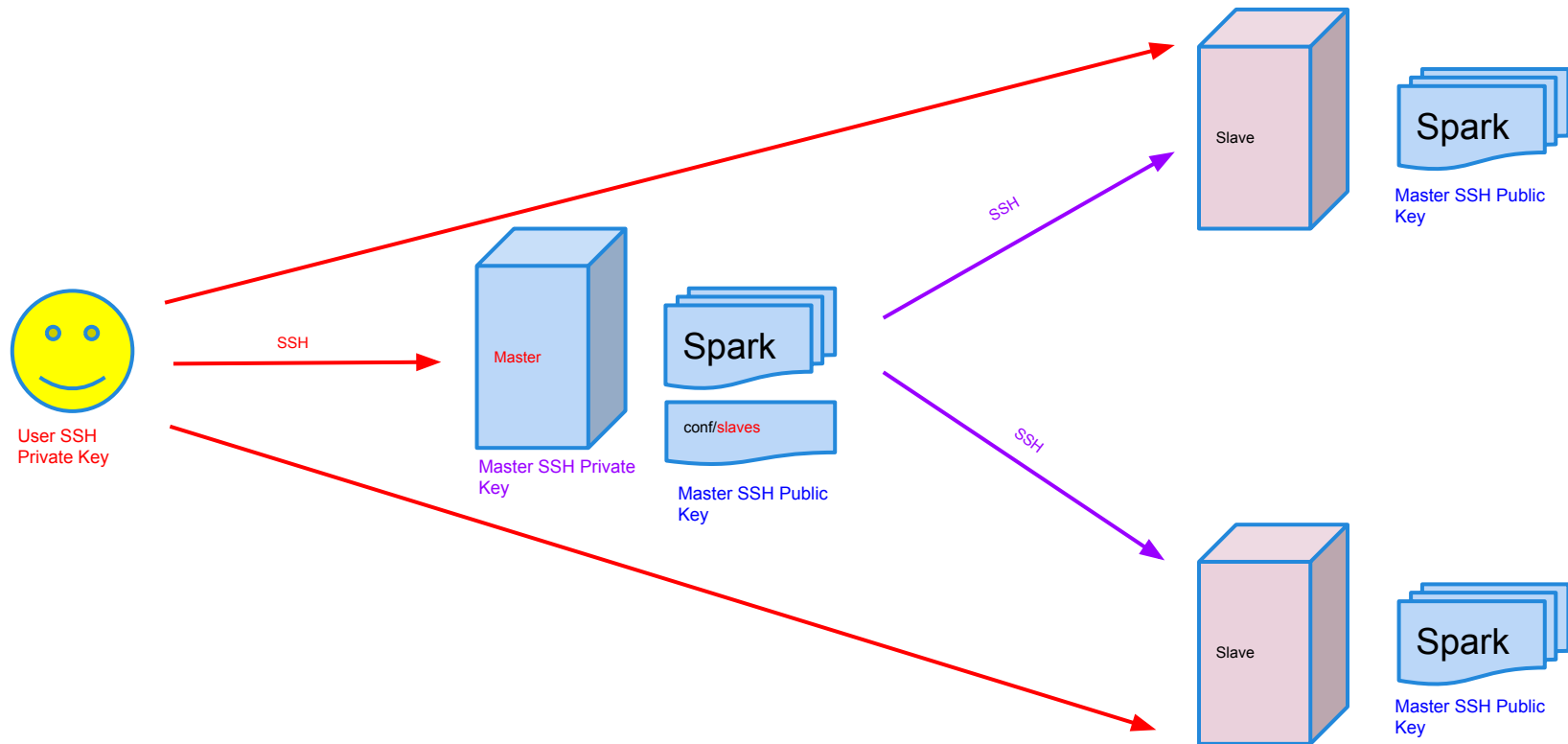
- Ansible 使用上, 會根據不同的功能呼叫不同的 Module
 - http://docs.ansible.com/ansible/modules_by_category.html
- Cloud Modules
- Clustering Modules
- Commands Modules
- Database Modules
- Files Modules
- Inventory Modules
- Messaging Modules
- Monitoring Modules
- Network Modules
- Notification Modules
- Packaging Modules
- Source Control Modules
- System Modules
- Utilities Modules
- Web Infrastructure Modules
- Windows Modules

Demo / Hands on Lab

Playbook



Use case: Standalone Cluster Manager



Use case: Standalone Cluster Manager

- templates 模組

- slaves

- <https://github.com/sakanamax/LearnAnsible/blob/master/books/Oreilly-Up-And-Running-Ansible/playbooks/templates/slaves>

- known_hosts

- https://github.com/sakanamax/LearnAnsible/blob/master/books/Oreilly-Up-And-Running-Ansible/playbooks/templates/known_hosts.j

Playbook

- A script is called a playbook.
 - 包含要進行組態的主機
 - 以及順序進行的工作
 - 包含許多的 play(*.yml)
- Ansible 針對所有的主機**同時(平行)**執行tasks.
 - Ansible **會等待** 所有主機 task 完成之後才會進行下一個 task.
 - Ansible 按照 **順序** 執行 tasks.
- Ansible playbooks 以 **YAML** 語法撰寫, 簡單易讀.
- 使用 **ansible-playbook** 指令執行.
 - ansible-playbook
 - ansible-playbook --verbose 顯示詳細資訊
 - ansible-playbook --check 不實際執行 dry run

Playbook

YAML 語法

- Yet Another Markup Language

- <http://yaml.org/>
- <https://zh.wikipedia.org/wiki/YAML>

- Start of File

- 以 3 個 --- 開始, 不加上也可以

- 註解

- 以 # 來進行單行註解
- #就是註解開始

- Strings

- 不一定要加上引號
- 可是有的時候為了易讀性, 可以使用 單引號或是雙引號

- Booleans

- 使用 True 或是 Yes 都可以視為真
- 但是還是用 True 不會混亂

- Lists

- They are delimited with hyphens, like this:
- - My Fair Lady
- - Oklahoma
- # inline 格式list
- [My Fair Lady, Oklahoma, The Pirates of Penzance]

- Dictionaries

- They look like this:
- address: 742 Evergreen Terrace
- state: North Takoma
- # inline 格式
- {address: 742 Evergreen Terrace, city: Springfield, state: North Takoma}

Plays

- 每一個 play 包含
 - A set of **hosts** to configure 目標主機
 - A list of **tasks** to be executed on those hosts. 工作內容
- 常用的設定
 - **name** - 執行的 play 或是 task 名稱.
 - **become** - 要不要執行 become (sudo).
 - **vars** - 變數設定.
- Tasks
 - 要在遠端主機執行的工作.
 - Modules
 - Modules are scripts that come packaged with Ansible.
 - http://docs.ansible.com/ansible/modules_by_category.html

Demo / Hands on Lab

Reference

- O'Reilly - Ansible Up & Running
- Blog 文章 [sakananote](#)
- 整理的心智圖 [Mindmap](#)
 - <https://github.com/sakanamax/LearnAnsible/tree/master/Mindmap>
 - 心智圖推薦使用 [freemind](#)
- Github
 - <https://github.com/sakanamax>
- Docker hub
 - <https://hub.docker.com/u/sakana/>

Thanks You