

實現 IT 自動化場景 - 以 Ansible 打造高效率工作環境

Palsys 朋昶數位科技

Leo Wang 王敬彬

2023/5/25

Agenda

- Ansible 基本介紹
- Lab 環境介紹 - 熟悉操作環境
- Workshop - 管理者的日常作業應用
- Workshop – 自動化情境串連
- Workshop - 利用 Ansible 即時反應資安事件
- Ansible 增值服務 - OPLUS Demo
- 應用情境討論

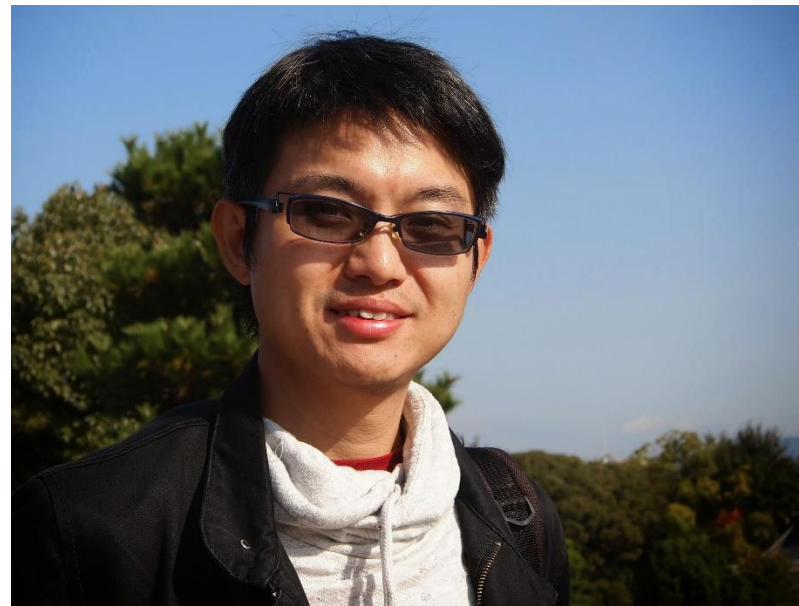
About Me

王敬彬 (Leo Wang)

朋昶數位科技 - 技術顧問
leo_wang@palsys.com.tw

- 18 年 IT 工作經驗
- 專長系統整合 RedHat / Splunk 產品線

- RedHat 認證 Ansible Sales Engineer
- Splunk 認證 Consultant & Architect



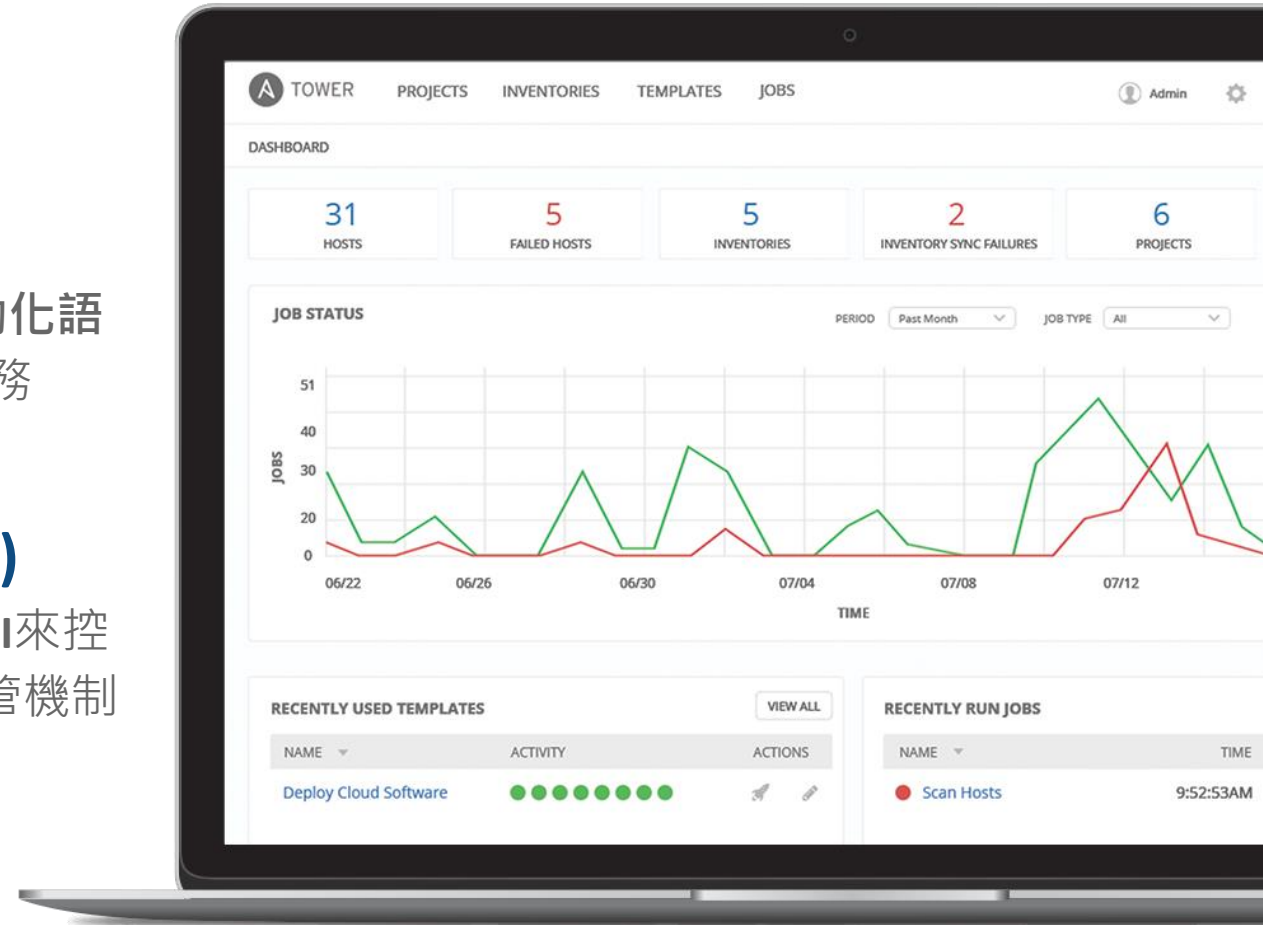
Ansible Automation Platform

Ansible Engine

是由Red Hat支援的開放原始碼社群，提供簡單自動化語言，用 Ansible Playbooks 描述要進行的各項工作任務

Ansible Tower (Automation Controller)

進階的 Ansible 管理框架，透過Web UI與RESTful API來控制管理 Ansible 的自動化; 提供 Role-Base 的權限控管機制來強化 Ansible 的安全性



ANSIBLE 的三大特色



夠簡單

- 適合人類閱讀自動化語言
- 不需特殊程式開發技能
- 依序執行工作(Tasks)提昇生產效率



夠強

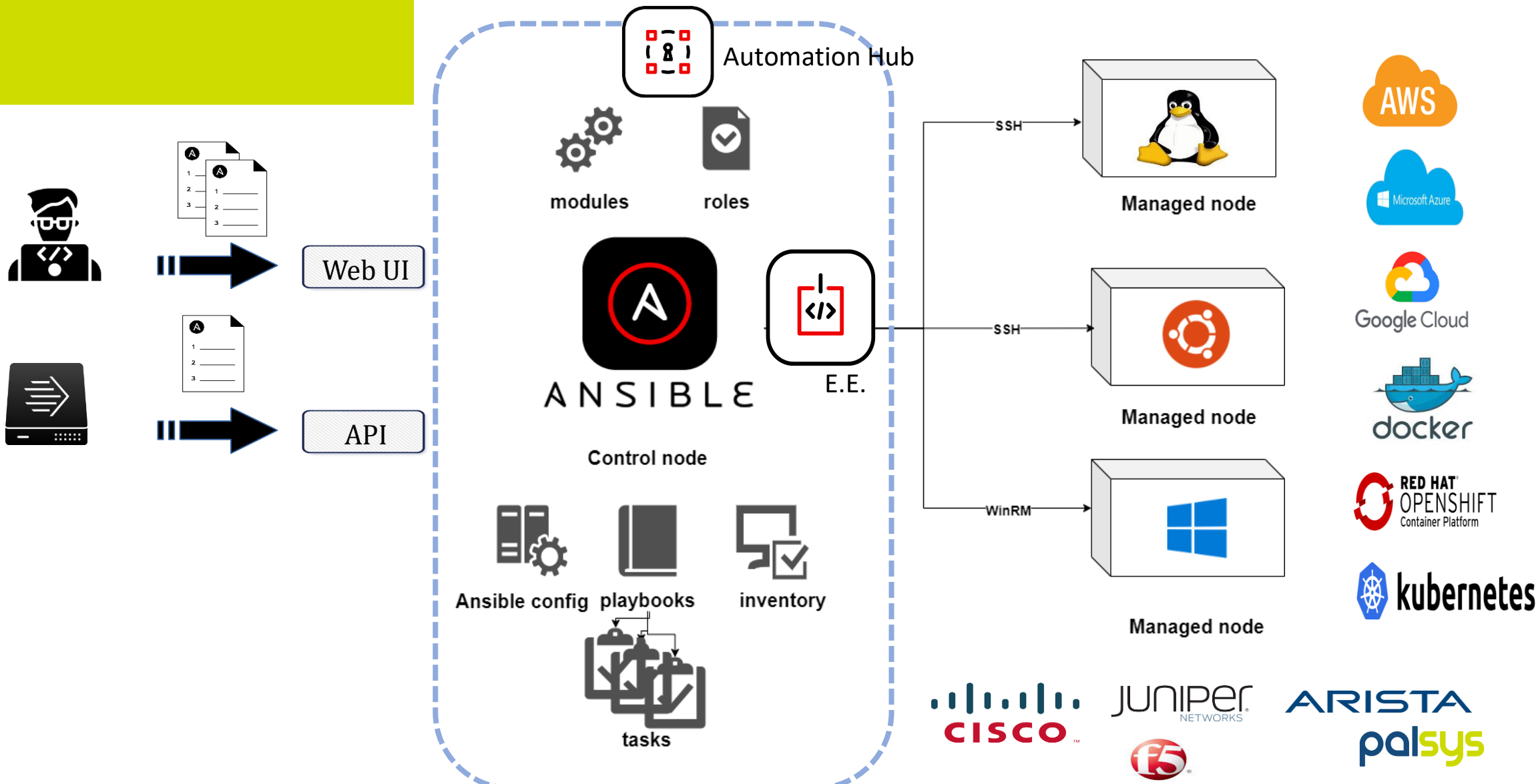
- 應用程式佈署
- 設定管理
- 流程與編配
- 網路自動化
- 編配應用程式生命週期



夠安全

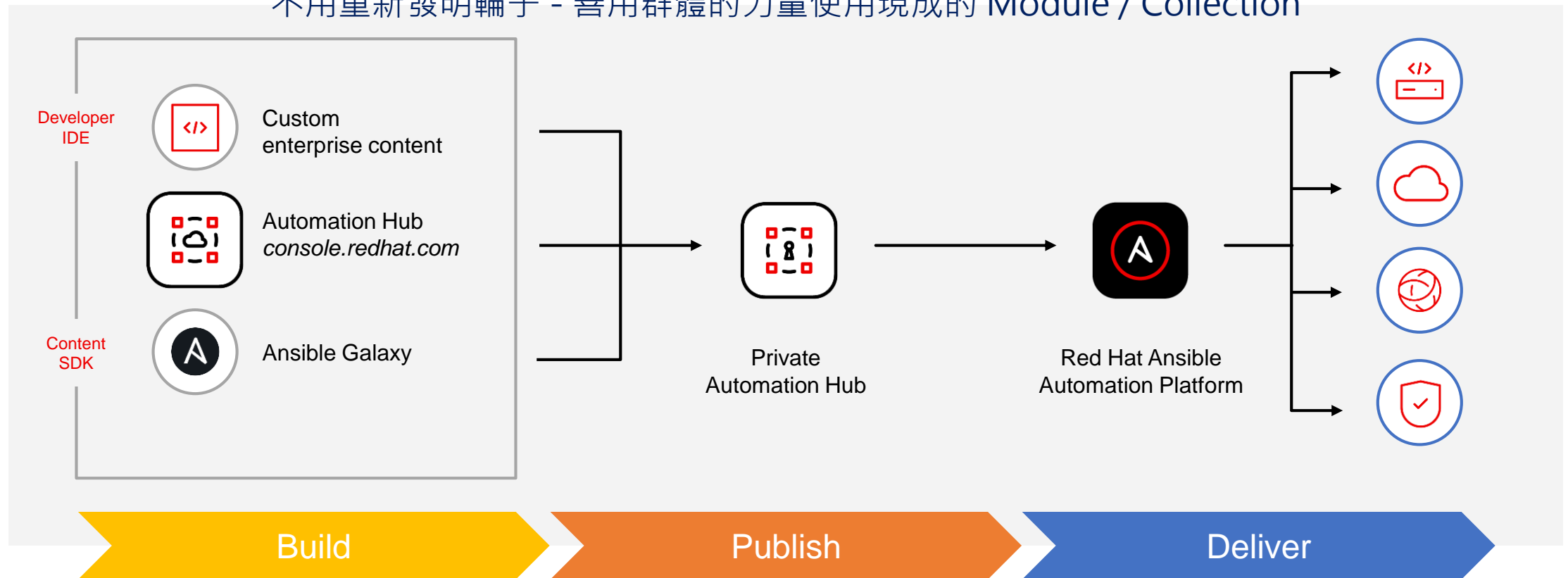
- Agentless 架構
- 使用既有的 SSH & WinRM
- 無需考慮Agent資安與更新
- 更有效率&更加安全

Ansible Automation Platform (AAP) 管理框架



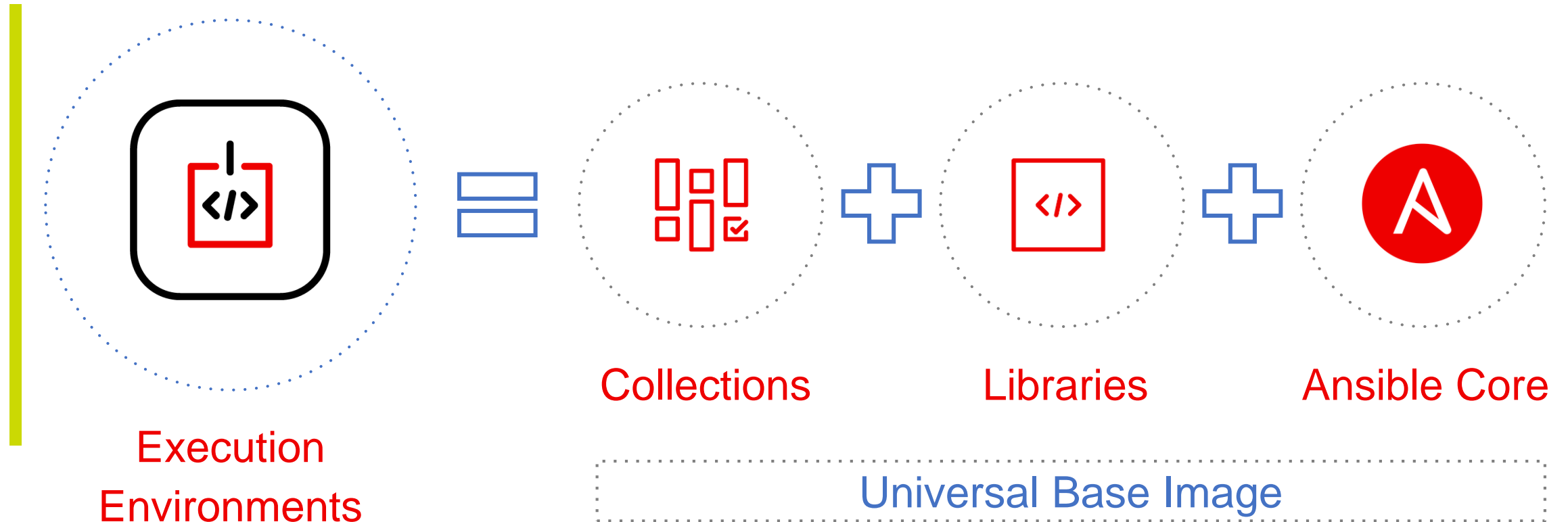
Automation Hub Architecture

不用重新發明輪子 - 善用群體的力量使用現成的 Module / Collection



Execution Environments 概念

以 Docker Image 做為執行環境，確保 Ansible 上的腳本與執行環境可以方便的移轉



Lab 環境介紹

- AAP Server (192.168.50.10) : RHEL 8
- Nodes
 - node1 (192.168.50.21) : RHEL 8
 - node2 (192.168.50.22) : RHEL 7
 - node3 (192.168.50.23) : CentOS 7
 - node4 (192.168.50.24) : RHEL 8
 - Pfsense (Firewall) (192.168.50.25) : --

Lab 環境啟動 & 檢查

本 Workshop lab 環境以 VirtualBox + Vagrant 建置, 以 vagrant 指令啟動 VM

1. 環境啟動

```
cd c:\vagrant_work\node
vagrant up
```

2. AAP 連線確認

```
ssh ansible@192.168.50.10
密碼: ansible
```

<https://192.168.50.10/>

密碼: admin / redhat123

2. 各 Node 連線確認

```
ssh ansible@192.168.50.21
密碼: ansible
```

```
cat /etc/redhat-release
>檢查各 Node 的 OS 皆不同
```

```
sudo vipw
>檢查可以 sudo 指令取得權限
```

Lab 環境啟動 & 檢查 (Firewall)

pfsense 是一種可以 VM 安裝的軟體 Firewall , 用於做為本 Workshop 的 Firewall Demo

1. 環境啟動

```
cd c:\vagrant_work\Pfsense  
vagrant up
```

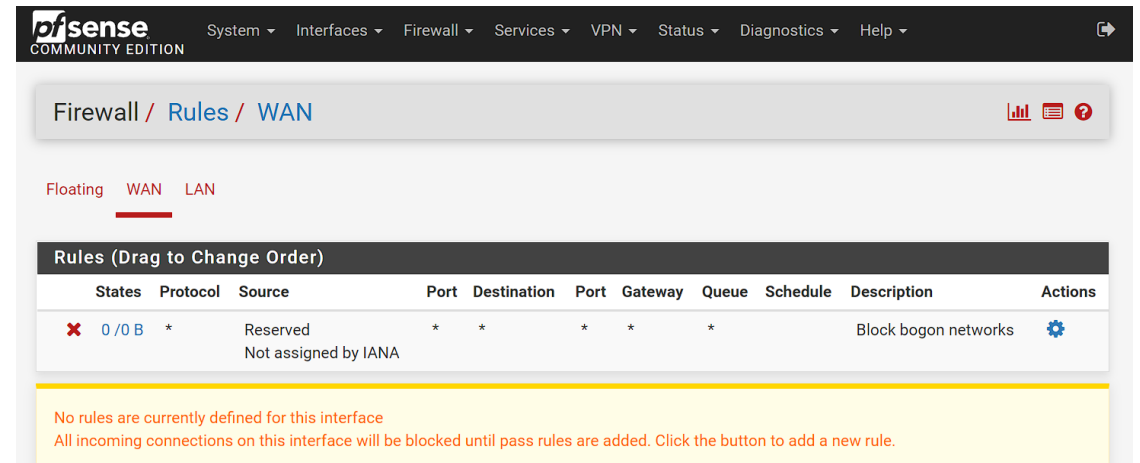
2. Pfsense 連線確認

<http://192.168.50.25/>

密碼: admin / pfsense

3. Firewall Rule 確認

Firewall → Rules



The screenshot shows the pfSense web interface for the Firewall Rules configuration page. The breadcrumb trail is "Firewall / Rules / WAN". The interface is for the "WAN" interface, with "Floating" selected. A table titled "Rules (Drag to Change Order)" is displayed with the following content:

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✘	0/0 B	*	Reserved	*	*	*	*	*	Block bogon networks	⚙️
Not assigned by IANA										

Below the table, a yellow warning box states: "No rules are currently defined for this interface. All incoming connections on this interface will be blocked until pass rules are added. Click the button to add a new rule."

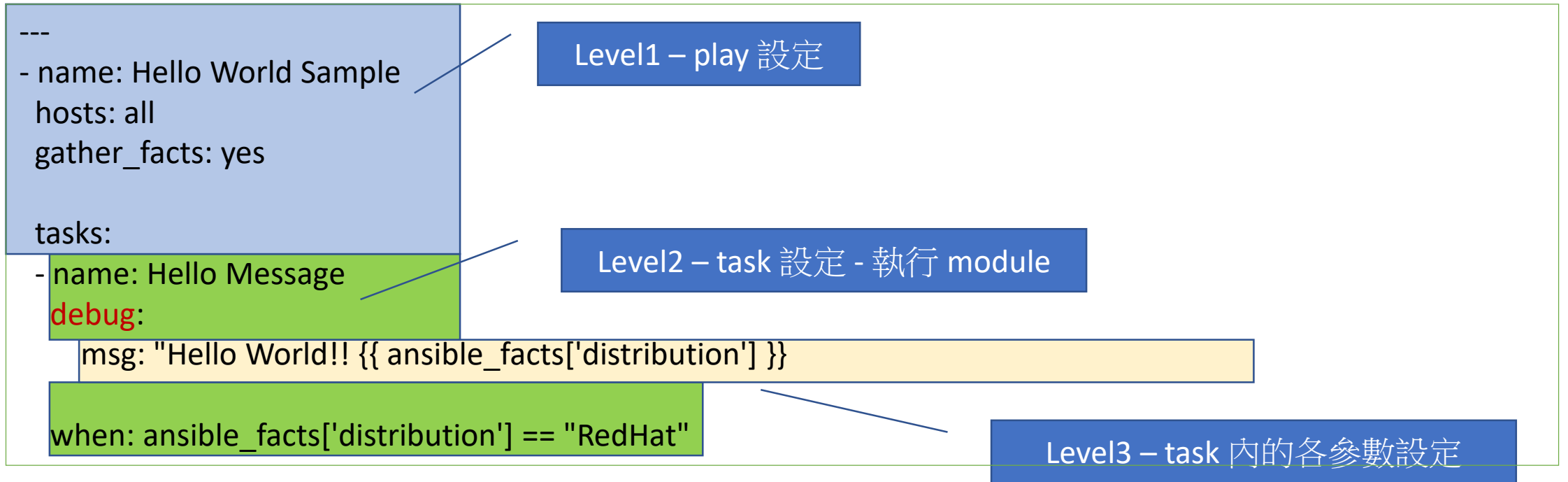
第一個 playbook - Hello World !!

```
---  
- name: Hello World Sample  
  hosts: all  
  gather_facts: yes  
  
  tasks:  
  - name: Hello Message  
    debug:  
      msg: "Hello World!! {{ ansible_facts['distribution'] }} {{ ansible_facts.distribution_version }},  
          {{ ansible_facts.kernel }} , {{ ansible_facts.hostname }}"
```

gather_facts 選項會在 connect 主機後, 自動收集該主機的細節資訊, 並儲在 ansible_facts 變數中
但使用此功能會增加一點啟動時間

```
## 以 Command 執行 Ansible playbook  
## 本 Workshop 的 playbook 置於 /var/lib/awx/projects/demo/  
  
$sudo su awx  
$cd /var/lib/awx/projects/demo/  
$ ansible-playbook hello_world.yml
```

Playbook – YML 檔格式



https://docs.ansible.com/ansible/latest/reference_appendices/playbooks_keywords.html#playbook-keywords
https://docs.ansible.com/ansible/2.9/modules/debug_module.html

Ansible Fact – 變數的使用

同 Shell 上的使用概念：

JAVA_HOME = /opt/java

echo \$JAVA_HOME



Ansible Fact

JAVA_HOME: /opt/java

{{JAVA_HOME}}

有 List, Dict 概念 (同 Python) :

Arr[1]

DictObject.Name

UserList[2].Name

```
UsersList = [
  { Name: user1, Mail: user1@gritfy.com, location: chennai },
  { Name: user2, Mail: user2@gritfy.com, location: chennai },
  { Name: user3, Mail: user3@gritfy.com, location: chennai }
]
```

List of Dictionaries

可用各式 Filter 做 Fact 資料處理:

Ex: | difference , | length

https://docs.ansible.com/ansible/latest/user_guide/playbooks_filters.html

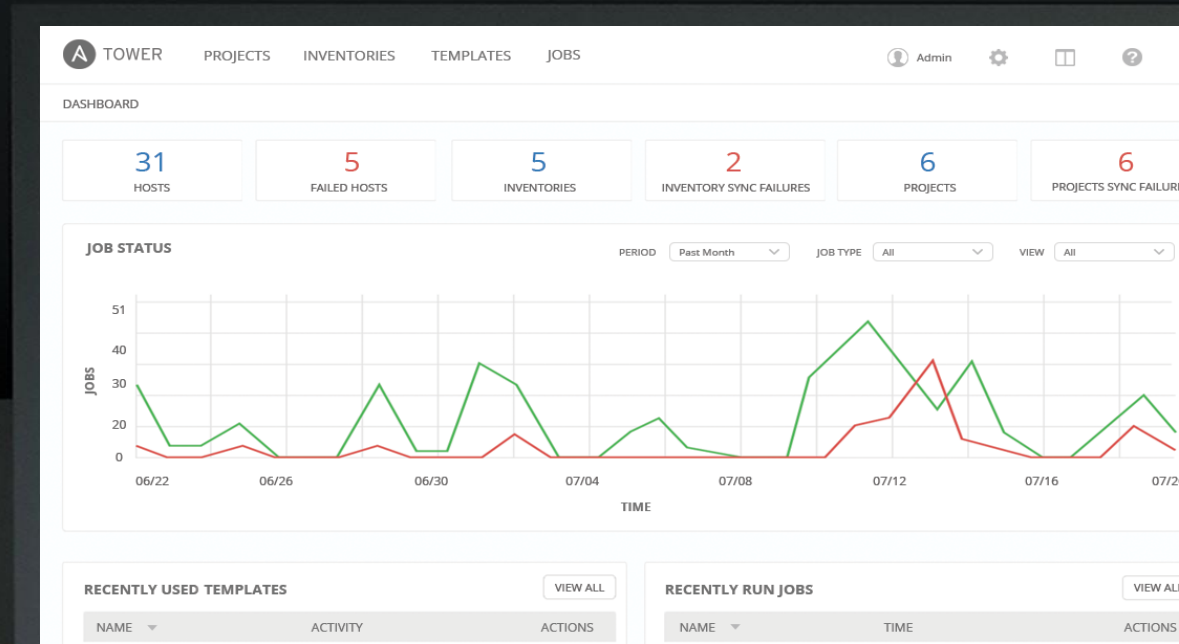
以 Ansible Web 介面進行操作



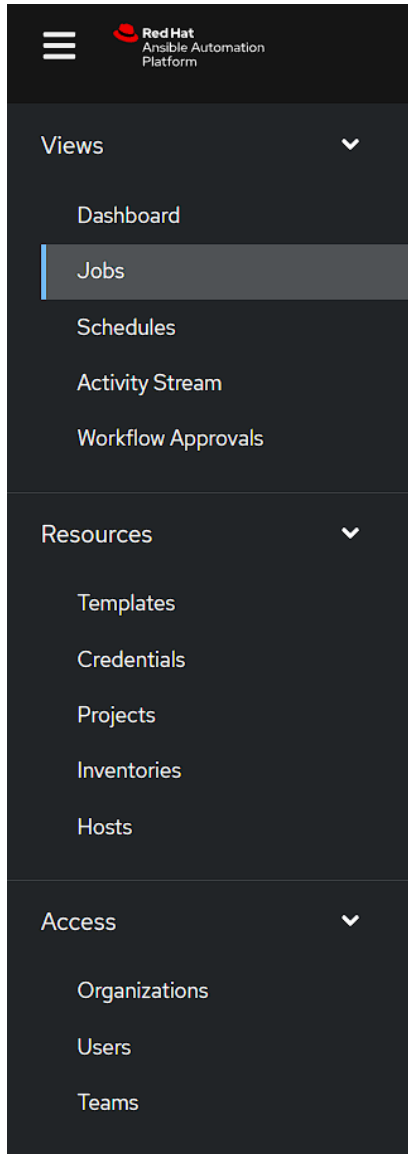
什麼是 Ansible Controller (Tower) ?

Ansible Controller / Tower是企業管理框架，透過 Web UI 與 RESTful API來控制管理並安全強化Ansible的自動化

- 角色為基礎存取控制(RBAC)
- 實現一鍵式部署或工作
- 所有自動化工作歷程都會被集中的記錄
- 提供 Restful API 進行系統間的串接整合



Ansible Web 介面 & 重點功能



- Inventories -
 - 資產清冊
- Credentials -
 - 登入的授權設定
- Template -
 - 執行 playbook 的相關設定
- Job –
 - 執行完成的 Job 記錄
- Schedule –
 - 可將 template 設定排程執行
- Project -
 - 專案, 做為管理 playbook 的群組分類
- User / Teams –
 - 使用者管理與權限控制

Ansible Web 的使用 1 - 建立 Inventories

1. 建立 Inventory , 取名 AnsibleLab

Inventories

Create new inventory

Name *	Description
<input type="text" value="AnsibleLab"/>	<input type="text"/>

Instance Groups

Labels ⓘ

Variables ⓘ YAML JSON

2. 切到 host 頁籤, 新增 4 台 host

Inventories > AnsibleLab > Hosts

Create new host

Name *	Description
<input type="text" value="node1"/>	<input type="text"/>

Variables YAML JSON

```
1 ---
2 ansible_host: 192.168.50.21
```

Press Enter to edit. Press ESC to stop editing.

3. 或是以 command 匯入

```
$ sudo tower-manage inventory_import --source=/var/lib/awx/projects/demo/inventory --inventory-name="AnsibleLab"
```

Ansible Web 的使用 2 – 建立 Credentials (ssh 登入的密碼)

Credentials

Create New Credential

1. 建立 Credential , 取名 ansible_ssh

Name * ansible_ssh Description Organization

Credential Type * Machine

Type Details

Username ansible Password Prompt on launch

SSH Private Key

Drag a file here or browse to upload

2. username / password 皆是 "ansible"

Browse... Clear

Signed SSH Certificate

Drag a file here or browse to upload

3. Escalation Method 選 "sudo" / "root"

Browse... Clear

Private Key Passphrase Prompt on launch

Privilege Escalation Method

sudo

Privilege Escalation Username

root

Privilege Escalation Password Prompt on launch

4. 填入 sudo 時的需輸入的密碼: "ansible"

Ansible Web的使用 3 - 建立 Project & Playbook

1. Web UI 上建立 Project

The screenshot shows the 'Create Project' form in the Ansible Web UI. The form is divided into several sections:

- Name ***: A text input field containing 'Demo'.
- Description**: An empty text input field.
- Organization ***: A dropdown menu showing 'Default'.
- Execution Environment**: A dropdown menu with a search icon.
- Source Control Type ***: A dropdown menu showing 'Manual'. A blue annotation box points to this field with the text: '因沒有使用 SCM, 選 "Manual"'. This indicates that 'Manual' is selected because no SCM (Source Control Management) is being used.
- Type Details**: A section containing:
 - Project Base Path**: A text input field containing '/var/lib/awx/projects'.
 - Playbook Directory ***: A dropdown menu showing 'demo'. A blue annotation box points to this field with the text: '選 "Demo" 目錄'. This indicates that the 'demo' directory is selected for the playbook directory.

At the bottom of the form, there are two buttons: 'Save' and 'Cancel'.

Ansible Web 的使用 4 – 建立 Template (for Hello World)

Name * **Description**

Job Type * Prompt on launch
 Run

Inventory * Prompt on launch

Project * Prompt on launch

Execution Environment Prompt on launch

Playbook * Prompt on launch

Credentials Prompt on launch

Labels Prompt on launch

Variables Prompt on launch

1 ---

- (1) 將之前建立的 Inventory / Project / Credential 關連上去.
- (2) Playbook 選 "hello_world.yml"

Lunch - 執行 Playbook Template

Templates > Demo 1 - Hello World

Details



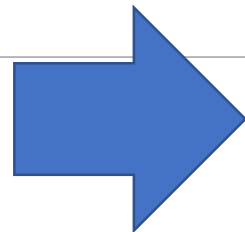
◀ Back to Templates Details Access Notifications Schedules Jobs Survey

Name	Demo 1 - Hello World	Job Type	run
Inventory	AnsibleLab	Project	Demo
Playbook	hello_world.yml	Forks	0
Timeout	0	Show Changes	Off
Created	12/1/2022, 3:36:58 PM by admin	Last Modified	12/1/2022, 6:47:37

Credentials

Variables

1 ---



Demo 1 - Hello World Successful Plays 1 Tasks 2 Hosts 4 Elapsed 00:00:13

Stdout

```
12   "msg": "Hello World!! RedHat , 8.5 , 4.18.0-348.el8.x86_64 , ansiblenode1"
13   }
14   ok: [node2] => {
15     "msg": "Hello World!! RedHat , 7.9 , 3.10.0-1160.el7.x86_64 , ansiblenode2"
16   }
17   ok: [node3] => {
18     "msg": "Hello World!! CentOS , 7.9 , 3.10.0-1160.49.1.el7.x86_64 , ansiblenode3"
19   }
20   ok: [node4] => {
21     "msg": "Hello World!! RedHat , 8.5 , 4.18.0-348.el8.x86_64 , ansiblenode4"
22   }
23
24   PLAY RECAP ***** 10:11:22
25   node1           : ok=2   changed=0   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
26   node2           : ok=2   changed=0   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
27   node3           : ok=2   changed=0   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
28   node4           : ok=2   changed=0   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
```

使用 limit & group 來限制主機範圍

Question: 這個 playbook 工作我想指定主機範圍執行, 可以怎麼做?

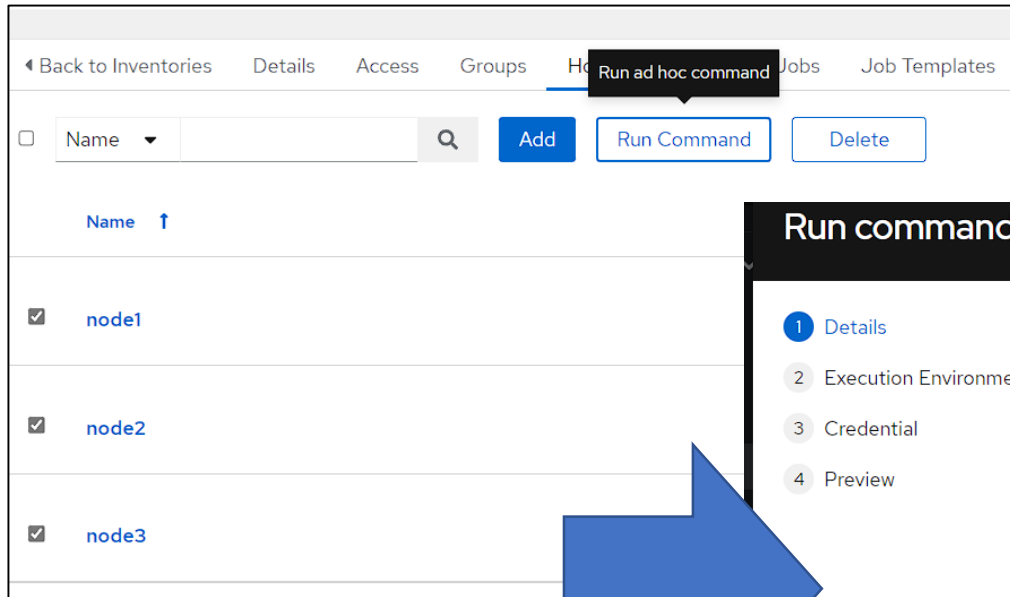
The image shows a screenshot of the AnsibleLab web interface. On the left, the 'Groups' section is visible under 'Inventories > AnsibleLab'. It shows a table with one group named 'rhel8_host'. A blue arrow points from this group to the 'Limit' field in the 'Forks' configuration section on the right. The 'Limit' field contains the text 'node1,node4'. Another blue arrow points from the 'Limit' field to the 'Groups' section, with the text '主機數多時, 可以在 Inventory 上設定 group 管理' (When there are many hosts, you can manage groups on the Inventory).

設定 limit

主機數多時, 可以在 Inventory 上設定 group 管理

臨時想對主機執行一個 Command ?

- Ansible 是個好幫手

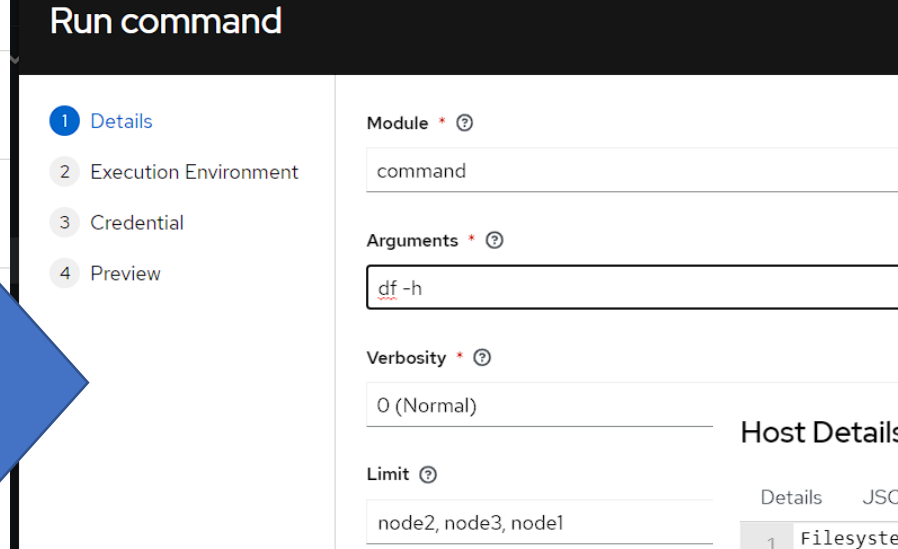


Back to Inventories Details Access Groups Hosts **Run ad hoc command** Jobs Job Templates

Name

Name ↑

- node1
- node2
- node3



Run command

- 1 Details
- 2 Execution Environment
- 3 Credential
- 4 Preview

Module * ⓘ
command

Arguments * ⓘ
df -h

Verbosity * ⓘ
0 (Normal)

Limit ⓘ
node2, node3, node1

Host Details

	Details	JSON	Standard Out	Standard Error
1	Filesystem			
2	devtmpfs		Size	Used Avail Use% Mounted on
3	tmpfs		908M	0 908M 0% /dev
4	tmpfs		919M	17M 903M 2% /run
5	tmpfs		919M	0 919M 0% /sys/fs/cgroup
6	/dev/mapper/centos_centos7-root		50G	1.5G 49G 3% /
7	/dev/sda1		1014M	131M 884M 13% /boot
8	tmpfs		184M	0 184M 0% /run/user/1001

管理者日常作業應用



任務 1 – Hostname 更名

- 情境:
 - VM 主機常是 Template 建立或直接 Clone 自其他主機, 若忘記更改 hostname 常造成管理上的問題
 - 有資產管理需求, 要將大量主機的 hostname 更名



怎麼有 10 台主機
都叫 appserver1

```
---  
- hosts: all  
  gather_facts: no  
  vars:  
    HOSTNAME: ""  
  tasks:  
    - name: "Print Information"  
      debug:  
        msg: "{{ansible_host}} Hostname change to {{HOSTNAME}}"  
  
    - name: "Change Hostname"  
      hostname:  
        name: "{{ HOSTNAME }}"  
  
    - name: "Update the hostname in /etc/hosts"  
      lineinfile:  
        path: /etc/hosts  
        regexp: "^{{ansible_host}}"  
        line: "{{ansible_host}} {{ HOSTNAME }}"  
        state: present
```

新建一個 Template : Demo 2 – Change Hostname

Templates > Demo 2 - Change Hostname

Edit Details

Name * Description Job Type * Prompt on launch

Inventory * Prompt on launch Project * Execution Environment

Playbook *

Credentials Prompt on launch

Labels

Variables

```
1 ---
2 HOSTNAME: LeoServer
```

- (1) 將之前建立的 Inventory / Project / Credential 關連上去.
- (2) Playbook 選 "change_hostname.yml"
- (3) Variables 將 HOSTNAME 的值設定上去

新建一個 Template : Demo 2 – Change Hostname (Cont.)

Forks [?] Limit [?] Prompt on launch

Job Slicing [?] Timeout [?] Show Char [?] Off

Instance Groups [?]

Job Tags [?]

Skip Tags [?]

Options

Privilege Escalation [?] Enable Webhook [?] Concurrent Jobs [?] Enable Fact Storage [?]



Demo 2 – Change Hostname

執行結果 & 驗證

```
Demo 2 - Change Hostname ✓ Successful Plays 1 Tasks 3 Hosts 1 Elapsed 00:00:08
```

```
Stdout  
```

```
1 BECOME password[defaults to SSH password]:
2
3 PLAY [all] ***** 15:03:29
4
5 TASK [Print Information] ***** 15:03:29
6 ok: [node2] => {
7   "msg": "192.168.50.22 Hostname change to LeoServer"
8 }
9
10 TASK [Change Hostname] ***** 15:03:29
11 changed: [node2]
12
13 TASK [Update the hostname in /etc/hosts] ***** 15:03:31
14 ok: [node2]
15
16 PLAY RECAP ***** 15:03:32
17 node2                : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
ansible@leoserver:~
login as: ansible
Keyboard-interactive authentication prompts from
| Password:
End of keyboard-interactive prompts from server
Last login: Tue Dec  6 06:50:43 2022 from 192.168.
[ansible@leoserver ~]$
[ansible@leoserver ~]$
[ansible@leoserver ~]$ hostname
leoserver
[ansible@leoserver ~]$
```

用 Survey Form 來設定參數 - 讓操作更簡便

1. 在 Template 設定中 → SURVEY → ADD

Edit Question

Question *

New Host Name

Description

Answer variable name * ?

HOSTNAME

Answer type * ?

Text

Required

Minimum length

0

Maximum length

1024

Default answer

Save Cancel

◀ Back to Templates Details Access Notifications Schedules Jobs Survey

Add Delete Survey Enabled

Name	Type	Default
<input type="checkbox"/> New Host Name *	text	

1. 填入 playbook 中
所需的參數值

2. Survey Enable (啟用)

3: 再一次將 Template Save & Launch

想要大量主機做批次更名？

- 情境:
 - 有資產管理需求, 要將大量主機的 hostname 更名
 - 可直接讀一個 CSV 檔做 lookup 對應
- 事前準備:
 - CSV 檔的對應清單, 可對應 IP & 新 hostname
 - /var/lib/awx/projects/demo/hostname.csv

```
IP,Hostname
192.168.50.21,dbserver
192.168.50.22,app_lab1
192.168.50.23,app_lab2
192.168.50.24,test_server
```

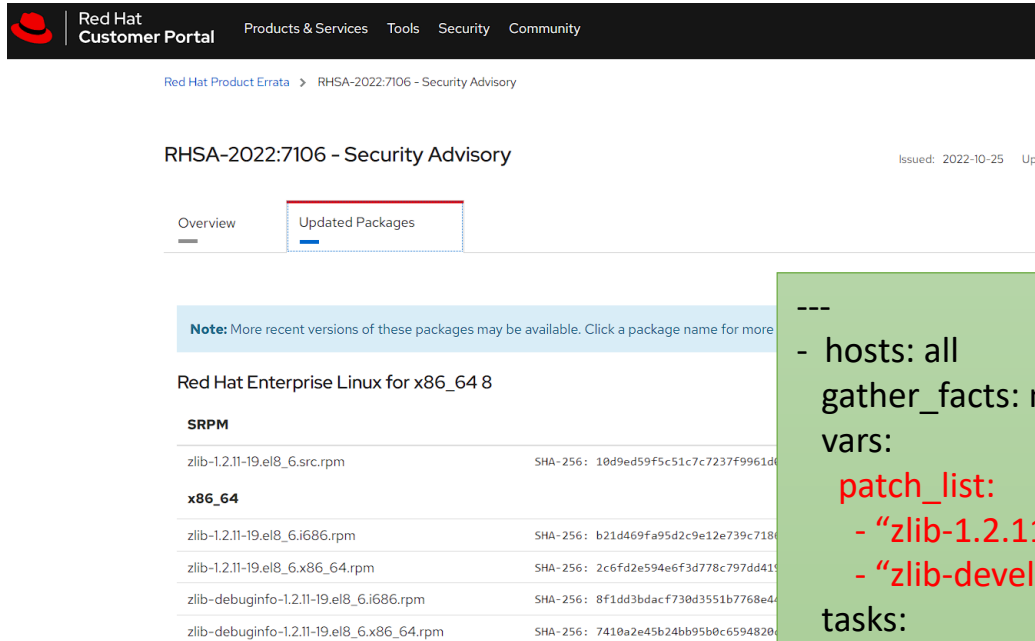
```
---
- hosts: all
gather_facts: no
vars:
  HOSTNAME: "{{ lookup('csvfile','{{ansible_host}} file=hostname.csv delimiter=',')}}"
tasks:
  - name: "Print Information"
    debug:
      msg: "{{ansible_host}} Hostname change to {{HOSTNAME}}"

  - name: "Change Hostname"
    hostname:
      name: "{{ HOSTNAME }}"

  - name: "Update the hostname in /etc/hosts"
    lineinfile:
      path: /etc/hosts
      regexp: "^{{ansible_host}}"
      line: "{{ansible_host}} {{ HOSTNAME }}"
      state: present
```

使用 lookup 機制從 CSV 檔中對應出指定的 HOSTNAME

任務 2 – 系統 Patch 更新



Red Hat Customer Portal | Products & Services | Tools | Security | Community

Red Hat Product Errata > RHSA-2022:7106 - Security Advisory

RHSA-2022:7106 - Security Advisory | Issued: 2022-10-25 | Upd:

Overview | Updated Packages

Note: More recent versions of these packages may be available. Click a package name for more

Red Hat Enterprise Linux for x86_64 8

SRPM	SHA-256
zlibc-1.2.11-19.el8_6.src.rpm	10d9ed59f5c51c7c7237f9961d
x86_64	
zlibc-1.2.11-19.el8_6.i686.rpm	b21d469fa95d2c9e12e739c718
zlibc-1.2.11-19.el8_6.x86_64.rpm	2c6fd2e594e6f3d778c797dd41
zlibc-debuginfo-1.2.11-19.el8_6.i686.rpm	8f1dd3bdacf730d3551b7768e4
zlibc-debuginfo-1.2.11-19.el8_6.x86_64.rpm	7410a2e45b24bb95b0c6594820



檔案上傳, 更新~ 上傳, 更新..... 整天做這些事就飽了

情境:

- 主機無法直接連外, 沒辦法直接用 yum update 更新
- 先整理好要更新的 rpm 清單, 由 ansible 負責上傳 & 更新的工作

```
---
- hosts: all
  gather_facts: no
  vars:
    patch_list:
      - "zlibc-1.2.11-19.el8_6.x86_64.rpm"
      - "zlibc-devel-1.2.11-19.el8_6.x86_64.rpm"
  tasks:
    - name: Copy Patch Files
      copy:
        src: patch_rpms/{{ item }}
        dest: /tmp
        with_items: "{{ patch_list }}"


    - name: Update Patches ( with yum localinstall )
      shell: yum --disablerepo=* localinstall *.rpm -y
      args:
        chdir: /tmp
```


Ansible Loop 的概念

重複性任務可以用 `with_items` 進行 Loop 循環 (Ex: 要檢查多個項目)

```
---
- hosts: ubuntu
  gather_facts: no

  tasks:
  - name: Ouput loop
    debug:
      msg: "NO. {{var_no}}"
    with_items:
      - 1
      - 2
    loop_control:
      loop_var: var_no
```



- `with_items` 定義要 loop 的項目清單 (Array) , 並通過 `loop_control` 命名變數 ,
- `loop_control` 沒指定的話 , 預設變數名為 `item` (如上一頁之範例)
- 變數內容會依次帶入同一 `task` 中執行

```
TASK [Ouput loop] *****
ok: [ubuntu18node1] => (item=1) =>
  msg: NO. 1
ok: [ubuntu18node1] => (item=2) =>
  msg: NO. 2
```

新建一個 Template : Demo 4 - Patches Update

Templates > Demo 4 - Patches Update

Edit Details

Name *

Description

Job Type * ? Prompt on launch

Inventory * ? Prompt on launch

Project * ?

Execution Environment ?

Playbook * ?

Credentials ?

Labels ?

- (1) 將之前建立的 Inventory / Project / Credential 關連上去.
- (2) Playbook 選 "update_patch.yml"
- (3) limit 設定: node4
- (4) Save & Launch

下次要更新的 Patches 不同 ? Template 可以 Re-Use

Variables ?

YAML

JSON

```
1 ---
2 patch_list:
3   - "xz-5.2.4-4.el8_6.x86_64.rpm"
4   - "xz-libs-5.2.4-4.el8_6.x86_64.rpm"
5
```

Press Enter to edit. Press ESC to stop editing.

Forks ?

0

Limit ?

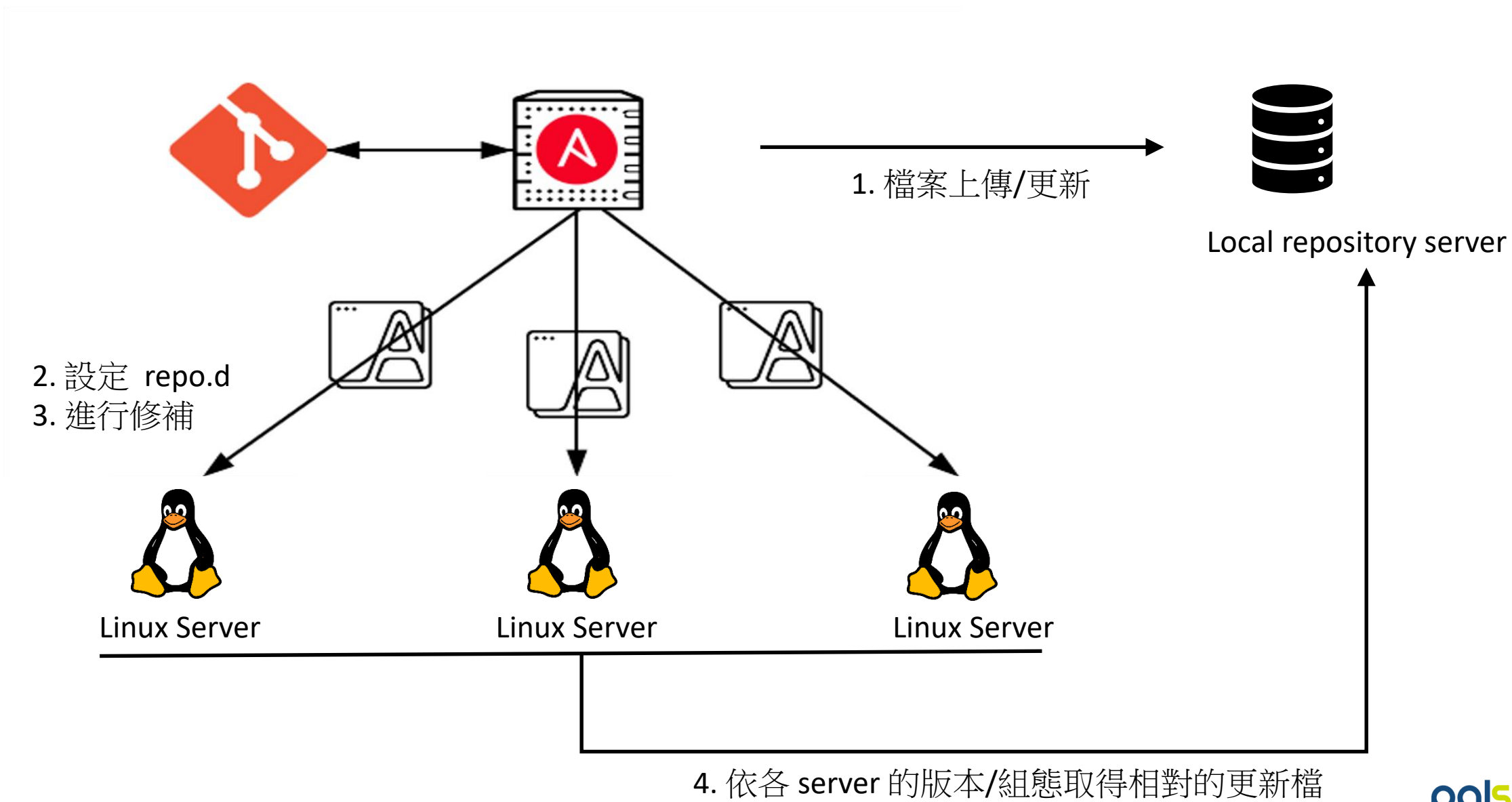
node4

patch_list:

- "xz-5.2.4-4.el8_6.x86_64.rpm"
- "xz-libs-5.2.4-4.el8_6.x86_64.rpm"

在 Variables 設定中, 設定新的清單, 會將 playbook 中定義的清單 overwrite 掉

Best Practice - 以 Local Repository 進行主機 Patch 修補



任務 3 – Firewall Policy 新增

- 情境:
 - 常常需要新增 Firewall Policy, 需要一個簡便 & 自動化的方式
- Firewall Policy 新增資訊
 - src - source ip
 - dest - dest ip
 - protocol – tcp/udp
 - dest_port -



Ansible 都幫我管一堆主機了, 那網路設備能不能也一起幫忙??

```
---
- hosts: pfsense

tasks:
  - name: "Add Internal traffic rules"
    pfsensible.core.pfsense_rule:
      name: "{{ src }} traffic to {{ dest }}"
      action: "pass"
      interface: wan
      ipprotocol: inet
      protocol: "{{ protocol }}"
      source: "{{ src }}"
      destination: "{{ dest }}"
      destination_port: "{{ dest_port }}"
      state: present      #present, absent
```

Firewall Policy 新增登入的 host / credential

1. Inventroy → AnsibleLab → host → Add

2. Credentials → Add

Inventories > AnsibleLab > Hosts > pfsense

Edit details

Name *

pfsense

Description

VM Firewall

Variables

YAML JSON

```
1 ---
2 ansible_host: 192.168.50.25
3
```

Press Enter to edit. Press ESC to stop editing.

Save

Cancel

Credentials

Create New Credential

Name *

pfsense_ssh

Description

for Pfsense

Organization

Q Default

Credential Type *

Machine

Type Details

Username

root

Password

Prompt on launch

SSH Private Key

Drag a file here or

Username : root
Password : pfsense

新建一個 Template : Demo 5 – Add Firewall Policy

Templates > Demo 5 - Add Firewall Policy

Edit Details



Name *	Description	Job Type * (?)	<input type="checkbox"/> Prompt on launch
Demo 5 - Add Firewall Policy		Run	
Inventory * (?) <input type="checkbox"/> Prompt on launch	Project * (?)	Execution Environment (?)	
AnsibleLab	Demo		
Playbook * (?)			
add_firewall_policy.yml			

Credentials (?)	<input type="checkbox"/> Prompt on launch
SSH: pfsense_ssh x	

Labels (?)

- (1) 將之前建立的 Inventory / Project / Credential 關連上去.
- (2) Playbook 選 "add_firewall_policy.yml"
- (3) Credentials 選 "pfsense_ssh"
- (4) Save

新建一個 Template : Demo 5 – Add Firewall Policy (Cont.)

◀ Back to Templates

Details

Access

Notifications

Schedules

Jobs

Survey



Add

Edit Order

Delete



Survey Enabled

Name	Type	Default
<input type="checkbox"/> Source IP *	text	any
<input type="checkbox"/> Dest IP *	text	
<input type="checkbox"/> Dest Port *	text	
<input type="checkbox"/> Protocol (tcp/udp) *	multiplechoice	

(1) 切到 Survey 頁籤, 將 playbook 所需的變數新增進去
src , dest , dest_port , protocol

(2) Survey Enabled
(3) Save & Launch

Launch & 結果

Launch | Demo 5 - Add Firewall Policy

1 Survey

2 Preview

Source IP *

192.168.50.10

Dest IP *

192.168.50.133

Dest Port *

1234

Protocol (tcp/udp) *

tcp

pfsense
COMMUNITY EDITION

System ▾

Interfaces ▾

Firewall ▾

Services ▾

VPN ▾

Status ▾

Diagnostics ▾

Help ▾

Firewall / Rules / WAN

Floating WAN LAN

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0 / 0 B	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
<input type="checkbox"/>	0 / 0 B	IPv4 TCP	192.168.50.10	*	192.168.50.133	1234	*	none		192.168.50.10 traffic to 192.168.50.133	

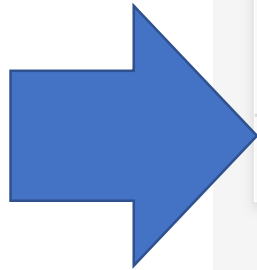
↑ Add

↓ Add

🗑 Delete

💾 Save

+ Separator



自動化情境串連



任務 X – 安裝 MySQL Server 並交付使用

- 任務流程
 - 建立 VM
 - 更改 hostname
 - 更新 Patch
 - 合規處理 (Ex: TWGCB)
 - 安裝 MySQL
 - 處理密碼問題



別人提交的一張安裝 MySQL Server 的工單, 其實裡面有一堆事要做

- 新增 Firewall Policy

```
[root@ansible01 ~]# cat /var/log/mysqld.log
2022-12-07T08:52:41.323323Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.31) initializing of server in progress as process 11365
2022-12-07T08:52:41.336502Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-12-07T08:52:42.859740Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-12-07T08:52:44.732187Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: jw1Ox(a/3_Pz
2022-12-07T08:52:49.983449Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.31) starting as process 11411
2022-12-07T08:52:49.993829Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-12-07T08:52:50.742933Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-12-07T08:52:51.267584Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2022-12-07T08:52:51.267623Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this
annel.
2022-12-07T08:52:51.293249Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.31' socket: '/var/lib/mysql/mysql.sock'
port: 3306 MySQL Community Server - GPL.
```

- 流程細節
 - Firewall 新增不是我說做就可以做
 - → 最好能有個簽核把關

建立 WorkFlow 來串接所有流程

Templates

Create New Workflow Template



Name *

WF- Provisioning MySQL Server

Description

Organization



Inventory ?

Prompt on launch

Limit ?

Prompt on launch

Source control branch ?

Prompt on launch

AnsibleLab

node1

Labels ?

Variables ?

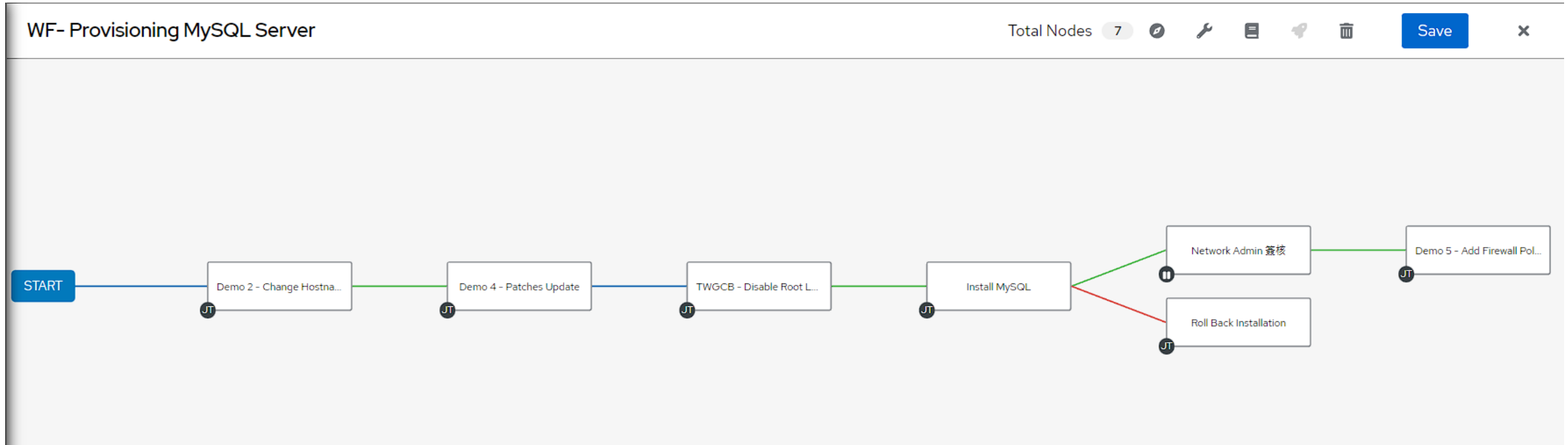
YAML JSON

Prompt on launch



1

建立 WorkFlow 來串接所有流程 (Cont.)



- (1) 以 GUI 的方式排定已定義好的 Template
- (2) 過程中, 有設定 Survey 的 Template, 會跳出表單輸入資訊
- (3) WorkFlow 可設定 Success / Failed 時執行不同的分支

AAP 內建簡易的簽核機制

Users → “network_admin” → Roles → Add (將新的 Workflow 的指定權限給此 user)

Add user permissions



- 1 Add resource type
- 2 Select items from list
- 3 Select roles to apply

Choose roles to apply to the selected resources. Note that all selected roles will be applied to all selected resources.

Selected

WF- Provisioning MyS...

Admin

Can manage all aspects of the workflow job template

Execute

May run the workflow job template

Read

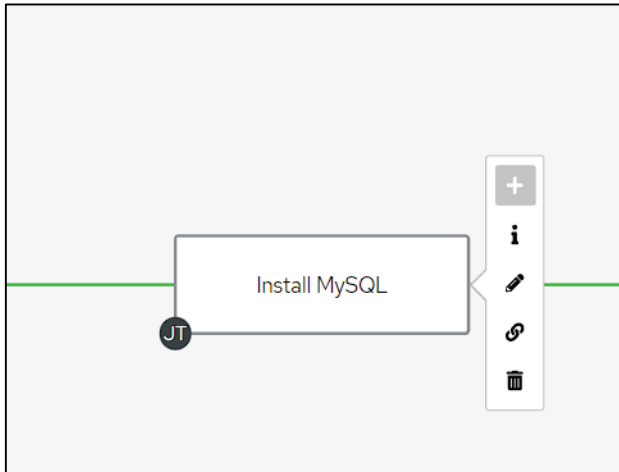
May view settings for the workflow job template

Approve

Can approve or deny a workflow approval node

AAP 內建簡易的簽核機制 - 插入一個 Approval

編輯 workflow , 在 Install MySQL 後插入一個流程



Node Type 選
“Approval”

Add Node

- 1 Run type
- 2 Node type

Node Type: Approval

Name *
Network Admin 簽核

Description
需新增 192.168.50.21 port 3306 的 firewall policy

Timeout
120 min 0

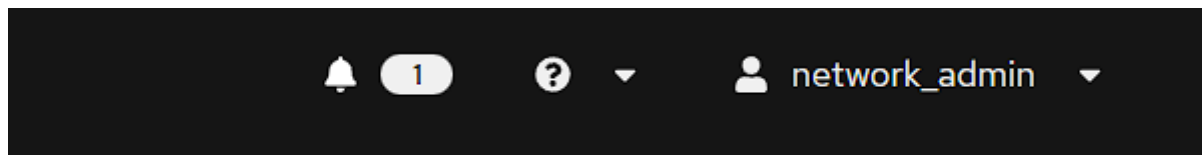
Convergence * ⓘ
Any

Node Alias ⓘ

Save Back Cancel

若有設 timeout , 則 Workflow 會在指定時間過後繼續進行 , 否則則會一直在 pending 狀態

AAP 內建簡易的簽核機制 (結果)



Workflow Approvals

Workflow 執行中 ;
以 network_admin 帳號登入 (密碼: redhat123)
點選上半部的 “通知” (小鈴噐)
可看到簽核通知

簽核後 , Workflow 才會繼續執行下去

Name

status pending

Name	Workflow Job	Started	Status	Actions
<input type="checkbox"/> 51 - 51 - Network Admin 審核	51 - WF- Provisioning MySQL Server	12/7/2022, 4:52:56 PM	Expires on 12/7/2022, 5:42:56 PM	<input type="button" value="Approve"/> <input type="button" value="Like"/> <input type="button" value="Dislike"/> <input type="button" value="More"/>

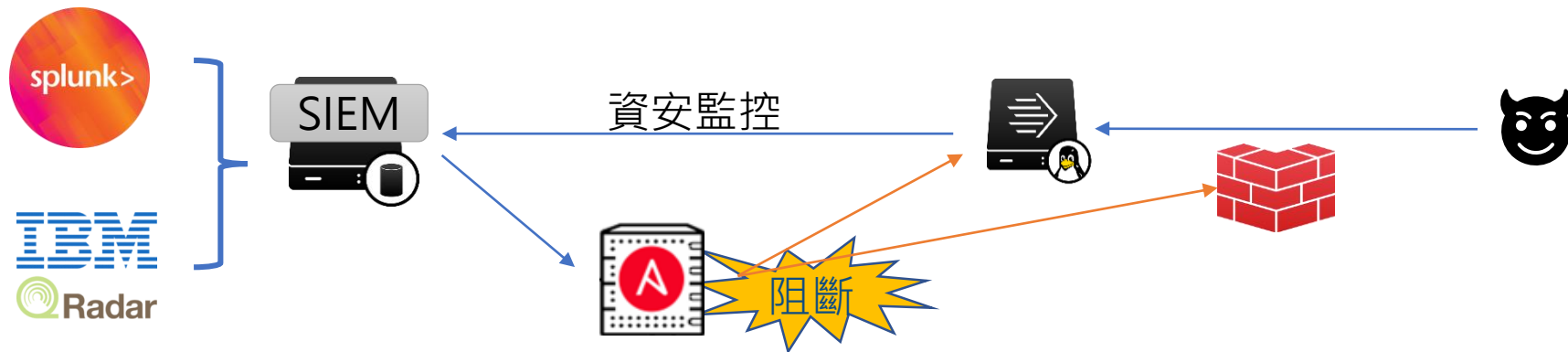
1 - 1 of 1 items 1 of 1 page

利用 Ansible 即時反應資安事件



以 Ansible 進行快速資安事件反應 (SOAR)

- SIEM 偵測到 user 登入失敗多次後成功登入 (*Brute force attack*)
 - ➔ 觸發 playbook, 中斷此 user 所有正進行的 process, 並 lock 帳號
 - ➔ 觸發 playbook, 在 Firewall 上新增一組 policy 將此來源 IP 阻斷

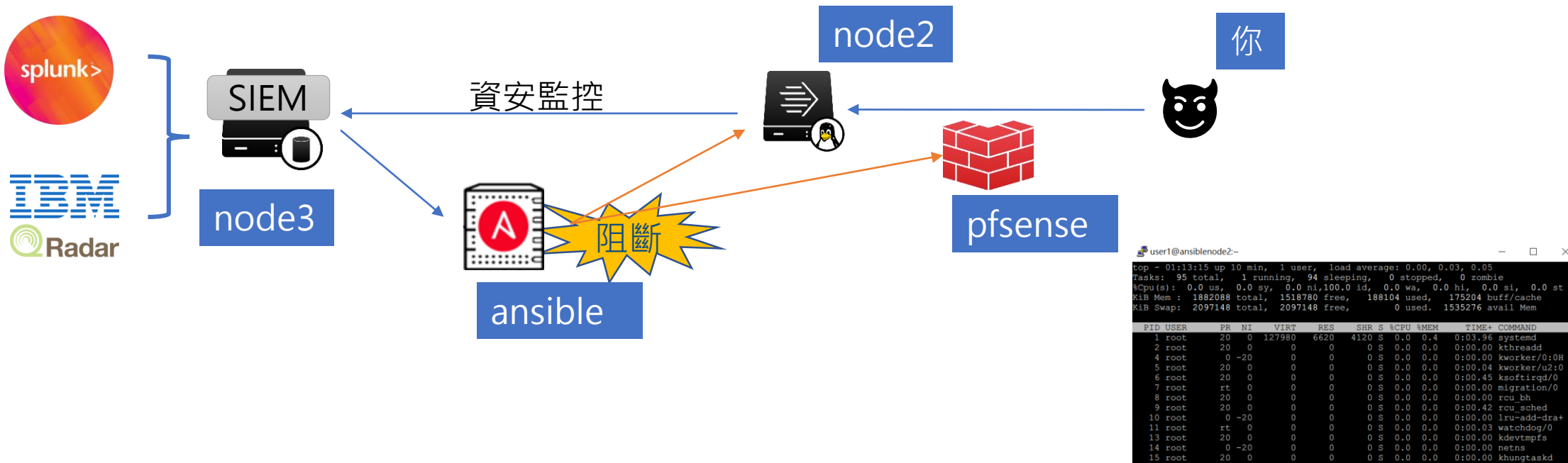


SIEM
(Event Correlation)



Ansible
(Event Action)

以 Lab 環境模擬此情境測試



1. 以 user1 帳號 (Password: user1) ssh 登入 node2 (192.168.50.22) , 執行 top 指令
假裝已入侵並執行工作
2. 在 node3 以 curl 指令呼叫 ansible api , 觸發以下兩支 playbook 執行動作
“[SOAR] kill user process and lock” , “Demo 5 – Add Firewall Policy”
3. 可發現 user1 的 top process 被中斷且 user1 被 lock , 不能再登入 ; pfsense 上可找到一筆新增的 Policy 記錄

事前動作 - 新增 Application (for SIEM 的 Trigger 使用)

Applications

Create New Application



Name *

SIEM CLI

Description

for SOAR Use Case

Organization *

Q Default

Authorization grant type * ⓘ

Resource owner password-based ▼

Redirect URIs ⓘ

Client type * ⓘ

Confidential ▼

Save

Cancel

因資安考量 - 新增 user 專門給 SIEM 觸發使用

以新增 user：“siem_ap_user” - 在 Role 頁籤給予指定的 2 支 template 權限 (Read / Execute)

Users > siem_ap_user

Roles

◀ Back to Users Details Organizations Teams Roles

Role 1 - 5 of 5

Name	Type	Role
Default	Organization	<input type="button" value="Member"/> ✕
[SOAR] Kill user process and lock	Job Template	<input type="button" value="Execute"/> ✕
[SOAR] Kill user process and lock	Job Template	<input type="button" value="Read"/> ✕
Demo 5 - Add Firewall Policy	Job Template	<input type="button" value="Execute"/> ✕
Demo 5 - Add Firewall Policy	Job Template	<input type="button" value="Read"/> ✕

因資安考量 - 新增 user 專門給 SIEM 觸發使用

改以 “siem_ap_user” 登入後, 進行 TOKEN 新增

Users > siem_ap_user > Tokens

Create user token

PS. 新增 Token 的功能需以 siem_ap_user 登入後才會顯示 (無法以 admin 帳號新增)

Application ⓘ

SIEM CLI

Description

Scope * ⓘ

Write

Save

Cancel

之後會得到一組 Token

Token information

This is the only time the token value and associated refresh token value will be shown.

Token > YIT3sXUZBI9VU3yUtE4xfYHFgpSthg

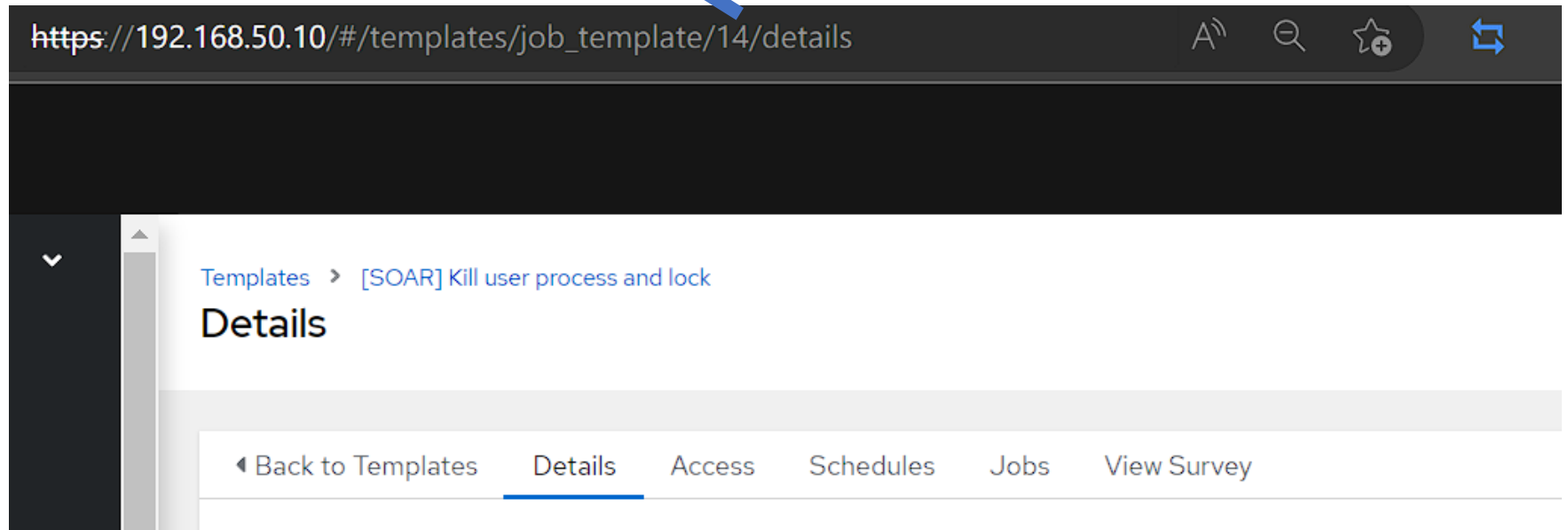
Refresh Token > j0lIfV50pZpxHtU7d3lEX6R1zwFVIf

Expires 4/9/2022, 6:39:33 PM

使用 Rest API Trigger Template (Lock User)

上一步驟取到的 Token , 用於 Restful API 的認證

```
curl -k -H "Authorization: Bearer YIT3sXUZBI9VU3yUtE4xfYHFgpSthg" -H "Content-Type: application/json" -X POST https://192.168.50.10/api/v2/job_templates/14/launch/ -d '{"limit": "node2", "extra_vars": {"target_account": "user1"}}'
```



使用 Rest API Trigger Template (Add Firewall Policy)

上一步驟取到的 Token , 用於 Restful API 的認證

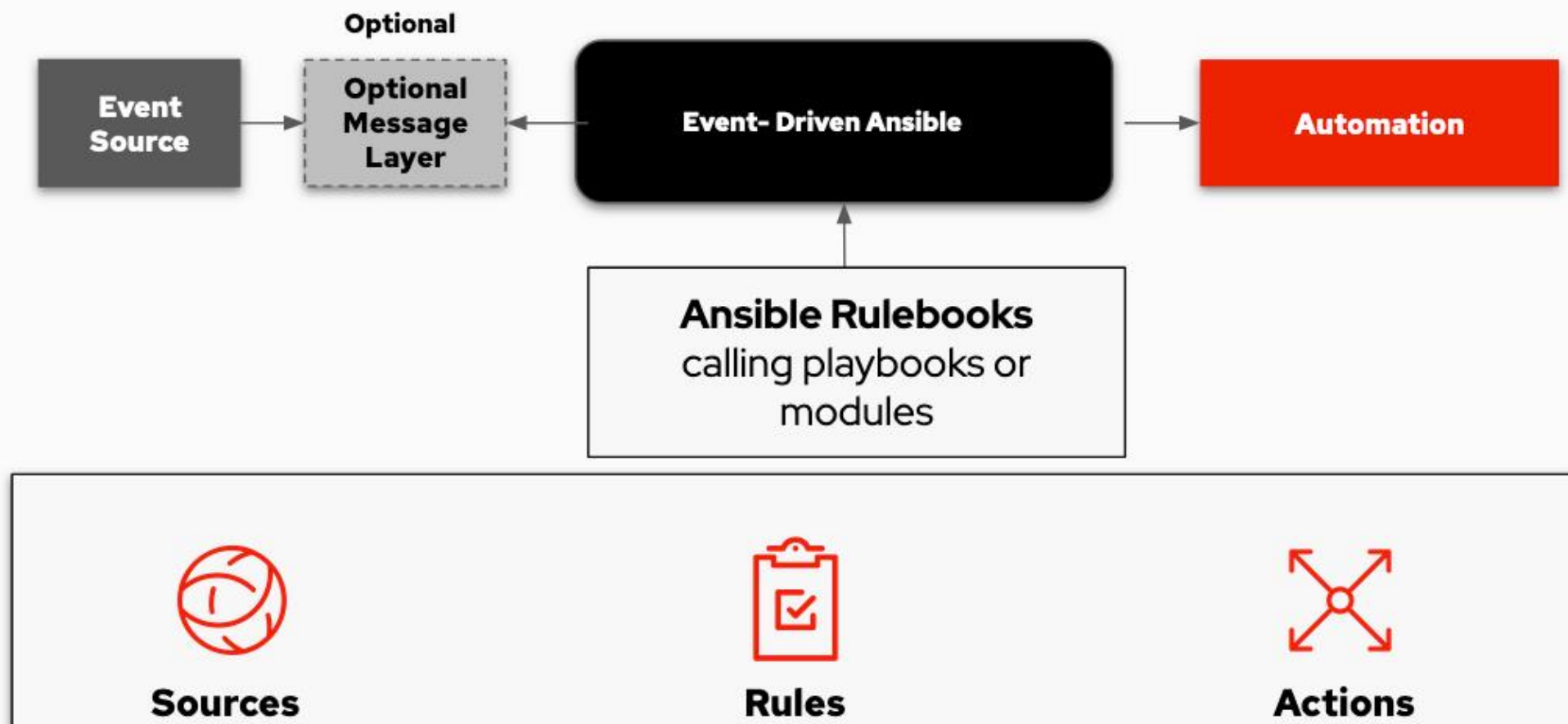
```
curl -k -H "Authorization: Bearer YIT3sXUZBI9VU3yUtE4xfYHFgpSthg" -H "Content-Type: application/json" -X POST https://192.168.50.10/api/v2/job_templates/19/launch/ -d '{"limit": "pfsense", "extra_vars": {"src": "168.95.3.3", "dest": "192.168.50.22", "protocol": "tcp", "dest_port": "22"}}'
```

將 Survey 中設定的各參數值, 以 Json 格式傳入



記得確認一下 Add Firewall Policy 的 Template ID 是否正確!!

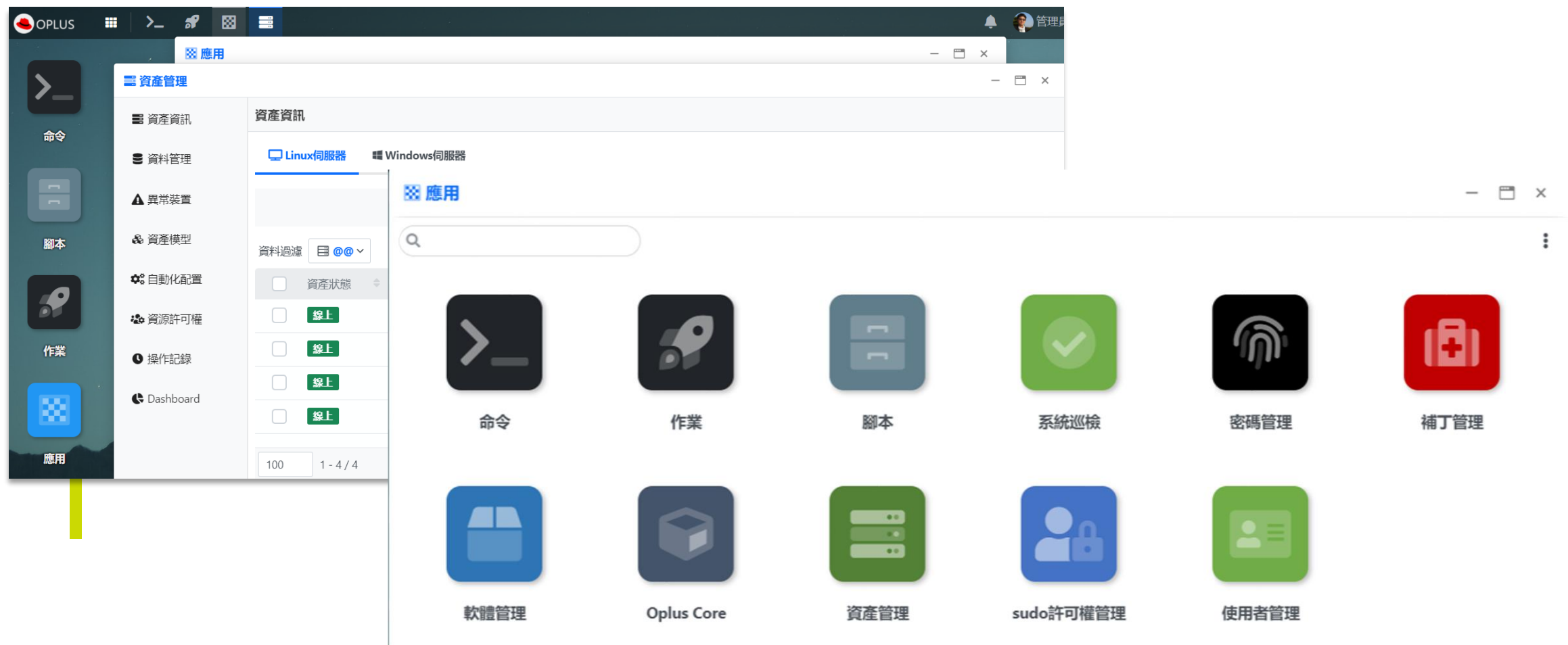
沒有事件處理平台？ 試試 Event-Driven Ansible !!



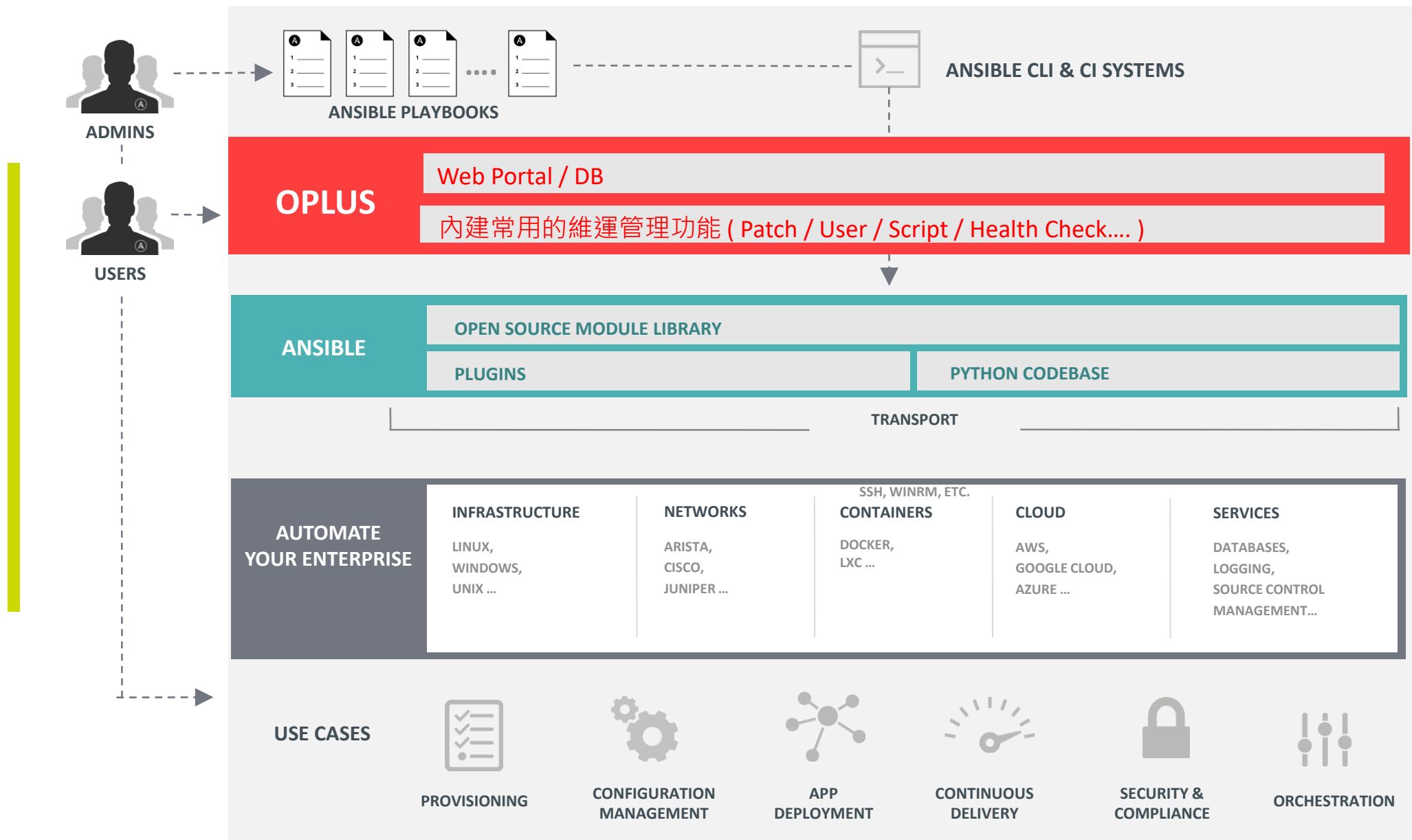
同場加映：
Ansible 增值服務 - OPLUS Demo



Ansible 增值服務: OPLUS - 讓你更省力!



OPLUS 架構概念



OPLUS 已實作常用之系統維運功能

- 資產管理
- 命令 - 執行 command / shell
- 腳本 - 執行 playbook
- 作業 - 多項 playbook 的檢查作業組合
- 系統巡檢 - 已定義好的常見系統檢查項目
- 補丁管理 - Patch 更新 (同時支援 Windows / Linux)
- 軟體管理
- 密碼管理
- 使用者管理
- sudo許可權管理
- (.... etc 持續增加)



資產管理與盤點



手上管一堆主機, 常常要調查各主機裝的 OS 版本, 還有是否已下線

VM 主機的 Spec
就失真, 常要重

有了 OPLUS 後:

(1) 進入 OPLUS

(2) “資產管理”



(3) “採集資訊” → “選取主機範圍”

(4) 得到最新的清單

Patch 盤點與更新



我常常不知道手上的主機更新 Patch 了沒？是不是有漏洞？

有新的嚴重 CVE 我很快盤出是甚麼，然後趕快處理

有了 OPLUS 後：

- (1) 進入 OPLUS
- (2) “補丁管理” 
- (3) 搜尋 CVE 編號 → 選定主機
- (4) 開始更新

User帳號管理與密碼管理



有盤點主機上

想要有類似 C
又不想導這麼

特權帳號有簽
reset 後回收

有了 OPLUS 後:

(1) 進入 OPLUS

(2) “使用者管理”



“選定主機” 盤點

(3) “密碼管理”



提交申請單

(4) 主管簽核後產生密碼

技術交流 - 應用情境分享討論



Thanks for Listening -

— Q&A

palsys