

Ejemplo típico de uso de Plantillas jinja2

Objetivo: Crear un archivo `/etc/hosts` dinámico en todos los nodos gestionados usando una plantilla Jinja2 que incluya información de todos los nodos del inventario.

Estructura del proyecto:

```
├─ playbook.yml
├─ templates/
│   └─ hosts.j2
```

1. Archivo de inventario (`hosts.inventario`):

OJO, ES UN EJEMPLO!!!! en clase ya tenemos nuestro fichero de inventario. Se puede modificar agregando sección con `[all:vars]`. En este ejemplo se agregará información al fichero host que no es ACORDE al que estamos usando

```
[webserver]
webserver1 ansible_host=192.168.33.11
webserver2 ansible_host=192.168.33.12

[databases]
dbserver1 ansible_host=192.168.33.13

[all:vars]
ansible_user=vagrant
domain=example.com # Dominio común para todos los nodos
```

2. Plantilla Jinja2 (`templates/hosts.j2`):

```
# Archivo /etc/hosts generado dinámicamente por Ansible
127.0.0.1    localhost localhost.localdomain
::1         localhost localhost.localdomain

{% for group in groups %}
{% if group != 'all' %}
# Grupo: {{ group }}
{% for host in groups[group] %}
{% Usar IP de los facts (ansible_default_ipv4) o fallback a ansible_host del
inventario #}
{{ hostvars[host].ansible_default_ipv4.address |
default(hostvars[host].ansible_host) }} {{ host }}.{{ domain }} {{ host }}
{% endfor %}
{% endif %}
{% endfor %}
```

3. Playbook (`play_jinja.yml`):

```
- name: Generar archivo /etc/hosts en todos los nodos
  hosts: all
  become: yes
  vars:
    domain: aula11.local

tasks:
  - name: Crear archivo hosts usando plantilla
    ansible.builtin.template:
      src: templates/hosts.j2
      dest: /etc/hosts
      owner: root
      group: root
      mode: '0644'
```

Explicación:

1.Plantilla Jinja2:

- Itera sobre todos los grupos del inventario (excepto `all`)
- Para cada host en cada grupo:
 - Usa la dirección IP (`ansible_host`)
 - Nombre del host (si está definido `hostname`, si no usa el nombre del nodo)
 - Crea un FQDN usando el dominio del grupo

2.Variables:

- `ansible_host`: Dirección IP del nodo (definida en el inventario)
- `domain`: Variable común para todos los nodos (definida en `all:vars`)

3.Playbook:

- Se ejecuta en todos los nodos (`hosts: all`)
 - Usa privilegios de root (`become: yes`)
 - Crea el archivo `/etc/hosts` usando la plantilla
-

Ejecución:

```
# OJO, solo si se ha creado ese inventario ficticio. Caso contrario, no se usa -i
ansible-playbook -i hosts.inventario play_jinja.yml
```

Resultado esperado en `/etc/hosts` (ej: en `webserver1`):

En el ejemplo , insisto. En clase nos dará otra información.

```
# Archivo /etc/hosts generado dinámicamente por Ansible
127.0.0.1    localhost localhost.localdomain
::1         localhost localhost.localdomain

# Grupo: webservers
192.168.1.10 webserver1 webserver1.example.com
192.168.1.11 webserver2 webserver2.example.com

# Grupo: databases
192.168.1.20 dbserver1 dbserver1.example.com
```

192.168.1.21 dbserver2 dbserver2.example.com

Verificación:

```
ansible webserver1 -i hosts.inventario -a "cat /etc/hosts"
```

Este ejercicio muestra cómo:

1. Usar variables de inventario en plantillas
2. Iterar sobre grupos y hosts
3. Generar contenido dinámico basado en la infraestructura
4. Aplicar configuración consistente en múltiples nodos