

13 Jinja2

Jinja2 en el contexto de Ansible es un motor de plantillas que permite crear contenido dinámico dentro de los playbooks y roles. Utilizando la sintaxis de Jinja2, se pueden insertar variables, aplicar filtros, condicionales y bucles para generar configuraciones personalizadas y adaptables según las variables definidas para cada host o entorno.

En Ansible, las plantillas basadas en Jinja2 generalmente tienen extensión `.j2` y permiten mezclar contenido estático con marcadores de posición dinámicos encerrados entre dobles llaves `{{ }}`. Esto facilita la automatización, ya que se pueden generar archivos de configuración que varían según los datos específicos de cada máquina, haciendo los playbooks más flexibles y potentes.

Además, Jinja2 permite trabajar con filtros para transformar datos (como convertir tipos, ordenar listas, aplicar valores por defecto, entre otros), y usar lógica condicional o bucles para controlar la generación del contenido de forma detallada.

Jinja2 en Ansible

Jinja2 es un motor de plantillas que permite generar contenido dinámico en Ansible. A continuación, se muestran ejemplos de **variables**, **condicionales**, **bucles** y **comentarios** en plantillas Jinja2.

Uso de variables

Las variables se insertan con `{{ }}`. Ansible reemplaza estos marcadores con los valores reales durante la ejecución.

```
# Archivo: apache.conf.j2
Listen {{ apache_port }}
ServerName {{ inventory_hostname }}
DocumentRoot {{ web_root }}
```

Si tienes definidas las variables:

```
apache_port: 8080
web_root: /var/www/html
```

El resultado será:

```
Listen 8080
ServerName nodo1
DocumentRoot /var/www/html
```

Condicionales

Con `{% if %}` puedes incluir contenido basado en condiciones.

```
# Archivo: firewall.conf.j2
{% if environment == 'production' %}
Allow from 192.168.1.0/24
Deny from all
{% elif environment == 'development' %}
Allow from all
{% else %}
Allow from 10.0.0.0/8
{% endif %}
```

Esto genera reglas de firewall distintas según el entorno.

Bucles (loops)

Con `{% for %}` puedes iterar sobre listas o diccionarios.

```
# Archivo: users.conf.j2
{% for user in allowed_users %}
AllowUser {{ user }}
{% endfor %}
```

Con la lista:

```
allowed_users:
- alice
- bob
- charlie
```

Resultado:

```
AllowUser alice
AllowUser bob
AllowUser charlie
```

Comentarios

Los comentarios en Jinja2 no aparecen en el archivo final. Se usan para documentar la plantilla.

```
{# Este es un comentario de Jinja2 - no se incluye en la salida #}
{{ apache_port }} # Puerto del servidor web #
```

A diferencia de los comentarios del sistema, que sí se generan:

```
# Este comentario SÍ aparece en el archivo de salida
Listen {{ apache_port }}
```

Ejemplos prácticos

Ejemplo de playbook con variables

```
---
- name: Uso de Jinja2
  hosts: srv1

  tasks:
    - name: Generación de plantilla
      ansible.builtin.template:
        src: plantilla.j2
        dest: /tmp/salida.txt
```

```
Hola {{ansible_hostname}}
Hoy es {{ansible_date_time.date}}
```

```
ansible@server1:~$ cat /tmp/salida.txt
Hola server1
Hoy es 2025-10-15
```

Ejemplo de playbook con condicionales

```
---
- name: Trabajar con JINJA2. Montar APACHE
  hosts: srv1
  vars:
    user_age: 30
    permiso: yes

  tasks:
    - name: Instalar Apache
      ansible.builtin.apt:
        name: apache2
        state: present
        update_cache: yes

    - name: Copiar pagina WEB
```

```
ansible.builtin.template:
  src: condicionales.j2
  dest: /var/www/html/index.html

- name: Arrancar Apache
  ansible.builtin.service:
    name: apache2
    state: started
```

```
{# Ejemplo con operadores lógicos and, or, not #-}

<h1> BIENVENIDO A MI PAGINA WEB DE ANSIBLE </h1>

{% if user_age >= 18 and not permiso %}
<h2> Eres adulto sin restricciones </h2>

{% elif user_age < 18 or permiso %}
<h2> Acceso restringido por edad o restricciones </h2>

{% else %}
<h2> Caso general o no definido </h2>
{% endif %}
```

Ejemplo de playbook con bucle

```
---

- name: Trabajar con JINJA2. Montar APACHE
  hosts: srv1
  vars:
    user_age: 30
    permiso: yes
    productos:
      - camisa
      - zapatos
      - camiseta

  tasks:
    - name: Instalar Apache
      ansible.builtin.apt:
        name: apache2
        state: present
        update_cache: yes

    - name: Copiar pagina WEB
      ansible.builtin.template:
        src: plantillas/bucles.j2
        dest: /var/www/html/index.html
```

```
- name: Arrancar Apache
  ansible.builtin.service:
    name: apache2
    state: started
```

```
{# Se utilizan las mismas que en Ansible
loop con una colección

for variable in Lista
endfor
#}

<h1>ESTOY EN UN BUCLE</h1>
<h2>Lista de productos</h2>
<ol>
{% for producto in productos %}
  <li> {{producto | upper}} </li>
{% endfor %}
  <li> {{productos | join(', ')}}</li>
</ol>
```

Uso de plantillas Jinja2 para crear un fichero virtualhost en apache2

Vamos a realizar la instalación de apache2 y el uso de plantillas para crear un archivo de host virtuales y las plantillas de los dos sitios almacenados.

El archivo **play.yaml** es el responsable de realizar la instalación, crear los sitios y las webs de estos.

```
---
- name: JINJA
  hosts: srv1

  tasks:
    - name: Leer configuración
      ansible.builtin.include_vars:
        file: config.yaml

    - name: Instalar apache
      ansible.builtin.apt:
        name: apache2
        update_cache: yes

    - name: Crear directorio /var/www/grupo1
      ansible.builtin.file:
        path: /var/www/grupo1
        state: directory
        owner: root
        group: root
        mode: "0755"
```

```
- name: Crear directorio /var/www/grupo2
  ansible.builtin.file:
    path: /var/www/grupo2
    state: directory
    owner: root
    group: root
    mode: "0755"

- name: Crear sitio grupo1
  ansible.builtin.template:
    src: templates/var/www/grupo1/index.html.j2
    dest: /var/www/grupo1/index.html

- name: Crear sitio grupo2
  ansible.builtin.template:
    src: templates/var/www/grupo2/index.html.j2
    dest: /var/www/grupo2/index.html

- name: Crear sitios web apache
  ansible.builtin.template:
    src: templates/etc/apache2/sites-available/sites.conf.j2
    dest: /etc/apache2/sites-available/sites.conf

- name: Deshabilitar sitio default en Apache
  ansible.builtin.shell:
    cmd: a2dissite 000-default.conf

- name: Habilitar sitios web Apache
  ansible.builtin.shell:
    cmd: a2ensite sites

- name: Restart apache2
  ansible.builtin.service:
    name: apache2
    state: restarted
```

Contenido del fichero *config.yaml*.

```
web1: Este es el sitio web 1
web2: Este es el sitio web 2

vhosts:
  - servername: grupo1.org
    documentroot: /var/www/grupo1
  - servername: grupo2.org
    documentroot: /var/www/grupo2
    serveradmin: root@gmail.com
```

Por otra parte, tenemos las plantillas de los virtualhost y de los sitios.

Plantilla: templates/etc/apache2/sites-available/sites.conf.j2 .

```
{% for vhost in vhosts %}
<VirtualHost *:80>
    ServerName {{ vhost.servername }}
    DocumentRoot {{ vhost.documentroot }}
{% if vhost.serveradmin is defined %}    ServerAdmin {{ vhost.serveradmin }}
{% endif %}
</VirtualHost>

{% endfor %}
```

Plantilla templates/var/www/grupo1/index.html.j2.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Inicio</title>
</head>
<body>
    <h1>{{ web1 }}</h1>
</body>
</html>
```

Plantilla templates/var/www/grupo2/index.html.j2.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Inicio</title>
</head>
<body>
    <h1>{{ web2 }}</h1>
</body>
</html>
```

Si ejecutamos con éxito nuestro playbook encontraremos el sitio web en las rutas: <http://grupo1.org:8080/> y <http://grupo2.org:8080/>.

Para este ejercicio es necesario recomendable editar tu fichero de hosts en tu equipo anfitrión.

```
# Configuración para Curso Ansible
127.0.0.1 grupo1.org
127.0.0.1 grupo2.org
```

Nota: En Linux está ubicado en la ruta `/etc/hosts` mientras que en Windows suele estar ubicado en `C:\Windows\System32\drivers\etc\hosts`.

Ejercicios para repasar los conceptos anteriores

Se propone la [Tarea 7](#) para repasar los conceptos vistos hasta este punto.