

## 04 Fichero de configuración e inventario

---

### Fichero de configuración de Ansible

En este fichero definimos la forma en como le indicamos a Ansible que tiene que comportarse de una determinada manera. Esta no es la única forma de configurar su configuración ya que podemos trabajar con:

- Fichero de configuración.
- Variables de entorno.
- Opciones en línea de comandos.
- Opciones y variables de los playbooks.

Ansible tiene su configuración en el fichero *ansible.cfg* que podemos encontrarlo en:

- En cualquier ubicación definida en la variable de entorno *ANSIBLE\_CONFIG*.
- En el fichero *ansible.cfg* del directorio actual donde se encuentra el proyecto.
- En el fichero *~/.ansible.cfg* del directorio */home* del usuario que ejecuta ansible.
- En el */etc/ansible/ansible.cfg*. Si hemos hecho la instalación a través de algún gestor de paquetes como *yum* o *apt*, seguramente encontraremos este fichero en el directorio */etc/ansible* sin embargo si lo hemos construido a mano o utilizado otra utilidad como por ejemplo *pip*, tenemos que generarlo inicialmente, aunque algunos generan uno en el directorio del usuario. Se Utiliza el comando siguiente:

```
ansible-config init --disabled > ansible.cfg
```

Podemos ver que en un primer momento no tenemos fichero de configuración ya que hemos utilizado el gestor de paquetes *pip* para la instalación.

```
ansible@debian64:~$ ansible --version
ansible [core 2.19.3]
  config file = None
  configured module search path = ['/home/ansible/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.11/dist-packages/ansible
  ansible collection location =
/home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.11.2 (main, Apr 28 2025, 14:11:48) [GCC 12.2.0]
(/usr/bin/python3)
  jinja version = 3.1.6
  pyyaml version = 6.0.3 (with libyaml v0.2.5)
```

Podemos ver que en un proyecto con fichero de configuración si se nos muestra la ruta del mismo.

```

ansible@debian64:~/practicass/pr01$ ansible --version
ansible [core 2.19.3]
  config file = /home/ansible/practicass/pr01/ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.11/dist-packages/ansible
  ansible collection location =
/home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.11.2 (main, Apr 28 2025, 14:11:48) [GCC 12.2.0]
(/usr/bin/python3)
  jinja version = 3.1.6
  pyyaml version = 6.0.3 (with libyaml v0.2.5)

```

Ejemplo de fichero de configuración.

```

# config file for ansible --http://ansible.com/
# =====
# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]
# some basic default values...
inventory      = $HOME/.ansible/hosts
remote_tmp     = $HOME/.ansible/tmp
forks          = 150
sudo_user      = root
transport      = smart

```

*Nota:* Son comentarios de una línea los que comienzan por # y comentarios *in line* los que van seguidos de ;.

*Nota 2:* Se pueden poner secciones para agrupar de modo efectivo las propiedades.

Ejemplo de fichero de configuración

Vamos a crear un proyecto con su propio fichero de configuración.

```

ansible@debian64:~/prueba$ ansible-config init --disabled > ansible.cfg
ansible-config init --disabled > ansible.cfg
ansible@debian64:~/prueba$ ls -lh
total 40K
-rw-r--r-- 1 ansible ansible 40K sep 22 21:58 ansible.cfg

```

## Inventario

En la mayor parte de las prácticas vamos a hacer uso de ficheros estáticos donde definimos el conjunto de nodos a administrar y hasta este punto hemos trabajado con un inventario llamado **maquinas**.

- Un inventario nos permiten indicar los servidores a los que queremos conectarnos.
- Puede ser un único fichero, una lista o usar plugins más avanzados y dinámicos que nos permitan personalizar los datos.
- Se pueden usar distintos formatos, como YAML, INI, etc...

Los fichero de inventario pueden tener formatos como son *INI* o *YAML* definidos a continuación respectivamente.

```
srv1 ansible_host=192.168.10.6 ansible_user=ansible ansible_become=yes

[debian]
ubuntu ansible_host=192.168.10.7 ansible_user=ansible ansible_become=yes

[red_hat]
centos ansible_host=192.168.10.8 ansible_user=ansible ansible_become=yes
fedora ansible_host=192.168.10.9 ansible_user=ansible ansible_become=yes
```

```
all:
  hosts:
    srv1:
      ansible_host: 192.168.10.6
      ansible_user: ansible
      ansible_become: yes

  children:
    debian:
      hosts:
        ubuntu:
          ansible_host: 192.168.10.7
          ansible_user: ansible
          ansible_become: yes

    red_hat:
      hosts:
        centos:
          ansible_host: 192.168.10.8
          ansible_user: ansible
          ansible_become: yes
        fedora:
          ansible_host: 192.168.10.9
          ansible_user: ansible
          ansible_become: yes
```

Se pueden crear Grupos de Servidores para racionalizar el uso de los despliegues y se puede tener el mismo servidor en varios grupos. Hay al menos 2 grupos predefinidos.

- **ALL:** Contiene todos los servidores dentro del inventario.
- **UNGROUPEd:** contiene los servidores que no se han asociado a ningún grupo.

También se pueden tener grupos anidados, es decir, grupos de servidores dentro de otros grupos de servidores.

```
[debian]
server1

[servers]
server2

[servers:children]
debian
```

Se pueden especificar rangos de servidores.

```
[servidores_aplicaciones]
tomcat[1:20]

[servidores_datos]
datos[1:4].empresa.com
```

```
all:
  hosts:
    datos1:
  children:
    servidores_aplicaciones:
      hosts:
        tomcat[1:20]:
    servidores_datos:
      hosts:
        datos[1:4].empresa.com:
```

También se pueden incluir varios inventarios a la hora de realizar una ejecución.

```
ansible -i inventario1 -i inventario2
```

El orden es importante, se debe tener cuidado con los nombres ya que realiza una carga en orden alfabético por lo que es recomendable tener cuidado con la nomenclatura.

```
ansible -i 01-uno -i 02-otro
```

## Formato YAML

YAML (siglas de "YAML Ain't Markup Language") es un lenguaje de serialización de datos diseñado para ser legible por humanos y fácil de escribir. Fue creado en 2001 por Clark Evans con la intención de manejar datos estructurados de manera simple y clara, sin usar etiquetas de marcado como XML. Se usa principalmente para archivos de configuración y para representar datos complejos mediante una sintaxis basada en sangrías, claves y valores, listas y mapas.

Sus principales normas incluyen usar espacios (no tabulaciones) para la indentación que marca la jerarquía, pares clave-valor separados por dos puntos seguido de espacio, listas representadas con guiones, y evitar símbolos complejos como llaves o corchetes. YAML es compatible con JSON y permite representar múltiples tipos de datos como cadenas, números, booleanos y colecciones. Este formato es muy popular en automatización, infraestructura como código y configuración de aplicaciones debido a su claridad y simplicidad.

## Práctica: Preparar fichero de configuración

- En esta práctica vamos a preparar un fichero de configuración con algunas opciones.
- Creamos un directorio.
- Copiamos el fichero de inventario que tengamos creado durante el curso.
- Dentro creamos un fichero denominado `ansible.cfg`.
- Ponemos los siguientes datos:

```
[defaults]  
inventory=./maquinas]
```

- Probamos lanzando un **ping** para todas las máquinas.
- Ansible debe haber seleccionado el fichero de inventario correspondiente.
- Ahora añadimos al fichero de configuración lo siguiente:

```
[privilege_escalation]  
become=yes  
become_user=ansible
```

- Con ello le decimos que queremos que se convierta en el usuario *ansible* al ejecutar el proceso.
- Volvemos a repetir el modulo **ping**. En este caso, debe fallar porque el usuario *ansible* no existe en las máquinas.