



## 1. OBJETIVO

En esta práctica trabajaremos con la herramienta de gestión de la configuración Ansible, con el fin de automatizar distintas tareas de administración sobre máquinas remotas Linux **heterogéneas**. Los módulos de Ansible (tareas) serán invocados desde la interface de comandos (modo ad-hoc) y mediante libros de jugadas (*playbooks*).

## 2. PREPARACIÓN DEL ENTORNO

Para la realización de la práctica construiremos un entorno de trabajo formado por una máquina de administración (con distribución Rocky 9.3) y cinco máquinas que serán gestionadas desde la anterior (dos con Rocky 9.3, una con Debian 11.2, otra con Ubuntu Server 22.4, y una última con OpenSUSE 15.3).

Nota: Estas instancias pueden arrancarse usando virtualización de escritorio (VMWare, VirtualBox, ...), en un proveedor de servicios *cloud* (AWS, ...), o en un entorno mixto (instancias locales y alojadas en el proveedor *cloud*). Si no es posible trabajar con las versiones del SO indicadas, se elegirán SO y versiones similares.

- Arrancar las seis máquinas virtuales e identificar sus direcciones IP.
- En los cinco equipos a gestionar comprobar que se cuenta con un servidor SSH arrancado. Crear o identificar un usuario con derechos administrativos (*sudo*) que usaremos para la administración (deberá tener *Shell* y directorio *home*, y podrá ejecutar comandos con cualquier otro perfil sin necesidad de introducir contraseña). Si es necesario, se pueden usar distintas cuentas de usuario para esta función, dependiendo del SO instalado.
- En el equipo de administración necesitamos una pareja de claves pública/privada sin frase de paso. Si el equipo administrado ya tiene inyectada la clave pública (instancias *cloud*), sólo será necesario contar con la clave privada correspondiente. En otro caso, crearemos una pareja de claves y copiaremos la clave pública al equipo administrado (utilizando la cuenta de usuario elegida para la administración remota).
- Comprobar que el usuario elegido puede acceder por SSH a las máquinas gestionadas utilizando la clave privada sin necesidad de introducir contraseña o frase de paso, y que el usuario puede ejecutar comandos con privilegios de *root*.

Nota: Incluir en la memoria los problemas encontrados para configurar las máquinas, y el modo en que se han resuelto.

## 3. CONFIGURACIÓN DE ANSIBLE

- Instalar Ansible en la máquina de administración.
- Editar el Inventario de Ansible y crear los siguientes grupos: WebServers (un equipo Rocky y los equipos Debian, Ubuntu y OpenSUSE), DNS\_Maestro (un equipo Rocky), DNS\_Esclavo (el otro equipo Rocky) y Servidores\_DNS (los dos equipos Rocky). Es posible crear otros grupos adicionales, si lo considera necesario para agrupar instancias con características comunes.
- Establecer las variables globales *ansible\_user* y *ansible\_ssh\_private\_key\_file* mediante uno o más archivos de variables (bajo el directorio *group\_vars*). Nota: Si todos los equipos administrados usan los mismos valores podemos usar el archivo *all*. En otro caso usaremos archivos específicos de cada grupo (el archivo tendrá el nombre del grupo).
- Comprobar la alcanzabilidad (capacidad de acceso y administración del equipo) usando el módulo *ping* sobre todos los equipos administrados (*all*). Probar también el módulo con la opción *-b* (acceso como *root*).



#### 4. EJECUCIÓN EN MODO AD-HOC

Probar el funcionamiento de Ansible en modo comando (*ad-hoc*) realizando sobre los equipos las siguientes tareas de configuración:

- Instalar Apache y Bind en uno de los servidores Rocky. Configurar el arranque automático al inicio de ambos servicios.
- Crear un directorio `/var/archivos` en todos los servidores y copiar un archivo `ejemplo.txt` (con un contenido arbitrario) a esa ubicación.
- Actualizar el sistema operativo del equipo Debian (Nota: puede ser un proceso lento)
- Comprobar el funcionamiento de los módulos **lineinfile** y **blockinfile**, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo `.conf`, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva `backup`. En el caso del módulo `blockinfile`, estudiar también la propiedad `Marker` e indicar cuál es su función.
- Comprobar el funcionamiento del módulo **template**. Preparar una plantilla en formato `jinja2` (`.j2`) que utilice alguna variable de prueba previamente definida en el archivo `group_vars/all`. Lanzar el módulo `template` sobre una máquina administrada y verificar el resultado.

#### 5. LIBROS DE JUGADAS

- Crear un libro de jugadas (**servidoresweb.yml**) para instalar la última versión del servidor Apache (paquete `httpd` o `apache2`, dependiendo de la distribución) y configurar un nuevo proyecto web, como servidor virtual, en las máquinas administradas.

Esta jugada se aplicará sobre el grupo `WebServers`, cuyos miembros tienen instaladas distintas distribuciones Linux: Rocky, Ubuntu, Debian y OpenSUSE.

El nombre de dominio del nuevo servidor (por ejemplo, `prueba.com`) será una variable pasada como parámetro en la llamada al *playbook* (usando `--extra-vars`). El proyecto web se ubicará en un directorio `/var/projects/<nombre del dominio>` del servidor administrado, y contendrá una página de inicio con el mensaje “Página de inicio del dominio <nombre del dominio>”.

Para facilitar la construcción se recomienda crear en la máquina administrada un archivo de configuración independiente para cada servidor virtual de Apache, con extensión `.conf`, y ubicarlo en un directorio de configuración de Apache (enlazado mediante un `include`, como los directorios `conf.d` o `sites-enabled`).

El *playbook* deberá recoger en variables las características diferenciadas de las distribuciones, para llevar a cabo la misma tarea sobre todos los equipos, independientemente de su distribución. Por ejemplo, la instalación de Apache podría hacerse de la siguiente forma con el paquete **package** (la variable `nombre_servicio` contendrá la cadena `httpd` o `apache2`)

```
- name: Instalar Apache con el modulo package
  package:
    name={{nombre_servicio}}
    state=latest
```

- Crear un libro de jugadas (**zonadns.yml**) para instalar el servidor de DNS Bind y una zona de resolución directa (cuyo nombre se pasará como parámetro con `--extra-vars` en la llamada al *playbook*) en los servidores `DNS_Maestro` y `DNS_Esclavo` (como su nombre indica, el primero mantendrá la zona maestra y el segundo una esclava de la misma zona).



Ambos servidores tendrán que configurar el servidor Bind para aceptar consultas desde cualquier origen y por cualquier interface, y no permitirán consultas recursivas.

Las zonas creadas serán muy sencillas, con un registro SOA, dos registros NS (ns1 y ns2 de la misma zona, uno para cada servidor de DNS y con las direcciones IP que figuran en el inventario o en las propias máquinas Maestra y Esclava), dos registros A para ns1 y ns2, y otros dos registros A para los servicios web de el dominio que damos de alta (con y sin www, y ambos apuntando a una dirección ficticia 10.10.10.10).

Se valorará el diseño utilizado: su estructura, originalidad, generalidad y simplicidad.

**Nota:** los aspectos significativos de la configuración realizada en los apartados 2 a 5 tendrán que reflejarse en la memoria de la práctica.

## 6. REFERENCIAS

- <https://www.ansible.com/>
- <https://docs.ansible.com/>