



Certified Tech Developer

The Ultimate Degree

Infraestructura I

Ansible en acción

En este ejercicio te proponemos conocer un poco más la herramienta Ansible y como puede automatizarse tareas a un grupo de servidores.

Como hemos visto hasta ahora Ansible nos permite hacer sistemas mucho más robustos, podemos automatizar procesos en nuestros servidores. Podemos encontrar más información en su web oficial:

- Levantamos el entorno de trabajo que habíamos preparado en la [clase asincrónica](#). Si hicimos la versión full debería verse como muestra la imagen, si no con solo 2 server podemos probarlo.



- Luego en nuestro servidor Ansible abrimos una Terminal y corremos el siguiente comando: "ansible --version"

```
usuario@debian-10:~$ ansible --version
ansible 2.7.7
  config file = /etc/ansible/ansible.cfg
  configured module search path =
  [ '/home/usuario/.ansible/plugins/modules', '/usr/share/ansible/p
  lugins/modules' ]
  ansible python module location =
  /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.7.3 (default, Jul 25 2020, 13:03:44) [GCC
  8.3.0]
```

- Verificamos nuestro archivo de inventario con el comando: "vim /etc/ansible/hosts"

```
[desarrollo]
192.168.1.43 ansible_ssh_user=ro
192.168.1.44 ansible_ssh_user=root

[produccion]
192.168.1.45 ansible_ssh_user=root
192.168.1.42 ansible_ssh_user=root
```

Recuerden:

- Configurado los certificados de seguridad, generado y enviado las claves SSH del servidor Ansible a los HOSTs,
- Habilitar el usuario root por ssh. ([ver en clase asincrónica](#))

Vamos a comenzar nuestro primer playbook en Ansible. Para ello vamos a trabajar dentro de nuestro servidor Ansible, creando el archivo de texto en una ruta del sistema. En principio puedes colocar los playbooks en cualquier carpeta del

ordenador de control. Usaremos un archivo con extensión .yaml, por ejemplo **primer-playbook.yaml**, que tendrá un código como el siguiente:

```
---
#Primer ejemplo de Playbook en Ansible
- hosts: all
  tasks:
    - name: Hacer ping
      ping:
```

En este ejemplo hemos creado un simple ping a todos los servidores configurados dentro del archivo **/etc/ansible/hosts**.

- Toda línea que comienza con # es un comentario.
- La declaración «hosts: all» indica que queremos hacer el proceso contra todos los servidores incluidos en el archivo de hosts.
- A continuación definimos la tarea en sí que se va a ejecutar, pudiendo encadenar varias tareas a realizar en un mismo playbook. Con el código anterior estamos aportando un nombre a la tarea e indicando que consiste en ejecutar realizar el ping.

Ejecutar el playbook

Un playbook es un fichero que ejecuta las funciones de configuración e implementación de Ansible.

Para ejecutar un playbook de Ansible tenemos que escribir el comando **«ansible-playbook»** seguido de la ruta o nombre de archivo .yaml donde está el código a ejecutar. Por ejemplo:

```
usuario@debian-10:~$ cd /etc/ansible/
usuario@debian-10:/etc/ansible/$ansible-playbook primer-playbook.yaml
```

Obteniendo un resultado similar al siguiente:

```
usuario@debian-10:/etc/ansible$ansible all -m apt -a "update_cache=yes" -b
10.0.2.8 | SUCCES => {
    "cache_update_time": 1632946878,
    "cache_updated": false,
    "change": false
}
10.0.2.7 | SUCCES => {
    "cache_update_time": 1632946855,
    "cache_updated": false,
    "change": false
}
10.0.2.9 | SUCCES => {
    "cache_update_time": 1632946876,
    "cache_updated": false,
    "change": false
}
10.0.2.10 | SUCCES => {
    "cache_update_time": 1632946997,
    "cache_updated": false,
    "change": false
}
```

Instalar Apache

Una vez tenemos instalado Ansible empezamos con la instalación de Apache. Para ello creamos un nuevo playbook con la configuración necesaria para darle un click e instalarlo. Para ello vamos a nuestro directorio **/etc/ansible** y creamos un nuevo archivo con el nombre de: **apache.yml**.

Añadimos lo siguiente:

```
---  
- hosts: desarrollo  
  remote_user: root  
  
  tasks:  
    - name: "instalar Apache"  
      shell: /usr/bin/apt install apache2 -y  
      register: apache  
  
    - name: "salida apache2"  
      debug: msg="{{ apache.stdout }}"  
  
    - name: "Reiniciar servicio Apache"  
      shell: /usr/bin/systemctl restart apache2  
      register: apache-restart
```

Finalmente nos posicionamos en la carpeta donde tenemos el archivo: apache.yml y lo ejecutamos de la siguiente forma:

```
usuario@debian-10:~$ cd /etc/ansible/  
usuario@debian-10:/etc/ansible/$ ansible-playbook apache.yml
```

Y finalmente llegamos al siguiente resultado:

```

usuario@debian-10: /etc/ansible
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
usuario@debian-10:/etc/ansible$ nano apache.yml
usuario@debian-10:/etc/ansible$ ansible-playbook apache.yml

PLAY [produccion] *****

TASK [Gathering Facts] *****
ok: [10.0.2.7]
ok: [10.0.2.9]

TASK [Instalar Apache] *****
changed: [10.0.2.7]
changed: [10.0.2.9]

TASK [salida apache2] *****
ok: [10.0.2.7] => {
  "msg": "Leyendo lista de paquetes...\nCreando árbol de dependencias...\nLeyendo la información de estado...\napache2 ya está en su versión más reciente (2.4.18-3+deb10u5).\nEl paquete indicado a continuación se instaló de forma automática y ya no es necesario.\n linux-image-4.19.0-11-amd64\nUtilice «apt autoremove» para eliminarlo.\n0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados."
}
ok: [10.0.2.9] => {
  "msg": "Leyendo lista de paquetes...\nCreando árbol de dependencias...\nLeyendo la información de estado...\napache2 ya está en su versión más reciente (2.4.18-3+deb10u5).\nEl paquete indicado a continuación se instaló de forma automática y ya no es necesario.\n linux-image-4.19.0-11-amd64\nUtilice «apt autoremove» para eliminarlo.\n0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados."
}

```

Ya tenemos instalado nuestro servidor apache en los dos host que tenemos. Si accedemos desde un navegador web a cualquiera de las dos direcciones IPs veremos la pantalla por defecto de Apache2.

