



Práctica de laboratorio 8: Utilice Ansible para automatizar la instalación de un servidor web

Versión en inglés:

<https://www.ccna6rs.com/7-4-8-lab-use-ansible-to-automate-installing-a-web-server-answers/>

Objetivos

Parte 1: Iniciar la Máquina Virtual (Virtual Machine) DEVASC.

Parte 2: Configurar Ansible.

Parte 3 Verificar las comunicaciones con el servidor web local.

Parte 4: Crear playbooks de Ansible para automatizar la instalación del servidor web.

Parte 5: Agregar opciones a su playbook de Ansible para servidores web Apache.

Aspectos básicos/Situación

En este laboratorio, primero configurará Ansible para que pueda comunicarse con una aplicación de servidor web. A continuación, creará un libro de jugadas que automatizará el proceso de instalación de Apache en el servidor web. También creará un libro de jugadas personalizado que instala Apache con instrucciones específicas.

Recursos necesarios

- 1 Computadora con sistema operativo de su elección
- Virtual Box o VMWare
- Máquina virtual (Virtual Machine) DEVASC

Instrucciones

Parte 1: Iniciar la Máquina virtual (Virtual Machine) de DEVASC

Si aún no ha completado el **laboratorio - Instalar el DEVASC-LAB**, hágalo ahora. Si se ha completado ya, inicie la máquina virtual DEVASC.

Parte 2: Configurar Ansible

La máquina virtual de DEVASC viene preinstalada con una serie de direcciones IPv4 ficticias que puede utilizar para varios escenarios y simulaciones. En esta parte, configurará Ansible para que utilice una de las direcciones IPv4 ficticias para un servidor web local.

Paso 1: Abrir una terminal en el DEVASC-LABVM.

Paso 2: Habilitar el servidor SSH.

El servidor SSH está deshabilitado en DEVASC-LABVM, junto con otros servicios que normalmente no son necesarios. Comenzar con el siguiente comando.

```
devasc @labvm: ~$ sudo systemctl start ssh
devasc@labvm:~$
```

Nota: El servidor SSH y la utilidad sshpass ya se han instalado en su máquina virtual. Para su referencia, estos se instalan mediante los siguientes comandos:

Instalar SSH

```
devasc @labvm: ~$ sudo apt-get install openssh-server
```

Instalar sshpass

```
devasc @labvm: ~$ sudo apt-get install sshpass
```

Paso 3: Abra el directorio ansible en VS Code.

- Abra **VS (Visual Studio) Code**.
- Haga clic en **Archivo > Abrir Carpeta...** y busque la carpeta **/labs/devnet-src/ansible**.
- Haga clic en **Aceptar**.
- Los dos subdirectorios de los laboratorios de Ansible ahora están cargados en el panel de VS Code **EXPLORER** para su conveniencia. En este laboratorio, trabajará con el directorio **ansible-apache**.

Paso 4: Editar el archivo de inventario de Ansible

- Abra el archivo **hosts** en el directorio **ansible-apache**.
- Agregue las siguientes líneas al archivo de **hosts** y guárdelo.
- Las credenciales **devasc** y **Cisco123!** son credenciales de administrador para la máquina virtual de DEVASC. La dirección IPv4 que utilizará para este laboratorio es 192.0.2.3. Esta es una dirección IPv4 estática en la máquina virtual bajo la interfaz dummy0, como se muestra en la salida del comando **ip addr**.

```
devasc @labvm: ~/labs/devnet-src/ansible$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    enlace/éter 08:00:27:97:ae:11 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 45882sec preferred_lft 45882sec
    inet6 fe80::a 00:27 ff:fe97:ae11/64 enlace de alcance
        valid_lft forever preferred_lft forever
```

```
3: dummy0: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group
default qlen 1000
    enlance/éter a 6:44:a7:e 8:6 a:9e brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/32 scope global dummy0
        valid_lft forever preferred_lft forever
    inet 192.0.2.2/32 scope global dummy0
        valid_lft forever preferred_lft forever
    inet 192.0.2.3/32 scope global dummy0
        valid_lft forever preferred_lft forever
    inet 192.0.2.4/32 scope global dummy0
        valid_lft forever preferred_lft forever
    inet 192.0.2.5/32 scope global dummy0
        valid_lft forever preferred_lft forever
    inet6 fe80::a444:a7ff:fee 8:6 a9e/64 link de alcance
        valid_lft forever preferred_lft forever
devasc@labvm:~/labs/devnet-src/ansible$
```

Paso 5: Editar el archivo `ansible.cfg`.

- a. En el subdirectorio **ansible-apache**, abra **ansible.cfg**.
- b. Puede remover el comentario. Agregue las siguientes líneas al archivo y guárdelo. El archivo **ansible.cfg** le dice a Ansible dónde encontrar el archivo del inventario y establecer ciertos parámetros preterminados.

```
(Predeterminado: )
# Use el archivo de hosts local en esta carpeta
inventario=./hosts
# No se preocupe por las huellas dactilares de RSA
host_key_checking = False
# No crear archivos de reintento
retry_files_enabled = False
```

Parte 3: Verificar las comunicaciones con el servidor web local

En esta parte, comprobará que Ansible puede enviar comandos al servidor web local.

Paso 1: Utilice el módulo `ping` para comprobar que Ansible puede hacer ping al servidor web.

Utilice el módulo **ping** de Ansible para verificar las comunicaciones con los dispositivos enumerados en el grupo de **servidores web** del archivo de inventario de **hosts**.

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ servidores web
ansible -m ping
192.0.2.3 | SUCCESS => {
  "ansible_facts": {
    «discovered_interpreter_python»: «/usr/bin/python3»
  },
  "changed": false,
  "ping": "pong"
}
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$
```

Si varios dispositivos se incluyeran en el grupo **servidores web** en el archivo de inventario de **hosts**, el resultado indicaría información similar para cada dispositivo.

Paso 2: Utilice el módulo de comando para verificar que Ansible pueda comunicarse con el servidor web.

Utilice el módulo de **comandos** de Ansible para verificar las comunicaciones con los dispositivos enumerados en el grupo de **servidores web** del archivo de inventario de **hosts**. En este ejemplo se envía el argumento **-un «/bin/echo hola mundo»** para pedirle al servidor web local que responda con ¡Hola, mundo!

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ ansible webservers -m comando -a «/bin/echo hello world»
192.0.2.3 | CAMBIADAS | rc=0 >
Hello, World! (¡Hola, mundo!).
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$
```

Parte 4: Crear guías de Ansible para automatizar la instalación del servidor web

En esta parte, creará dos manuales de Ansible. El primer manual automatizará la prueba de eco que hizo en la parte anterior. Imagine que está conectando cien servidores web. El grupo [webserver] del archivo **hosts** enumeraría toda la información necesaria para cada servidor web. A continuación, puede utilizar un sencillo playbook para verificar las comunicaciones con todos ellos con un solo comando. En el segundo playbook, creará y automatizará la instalación del software de servidor web Apache.

Paso 1: Crear un playbook Ansible para probar su grupo de servidores web.

En este paso creará un playbook Ansible para realizar el mismo comando **echo**.

- En VS Code, cree un nuevo archivo en el directorio **ansible-apache** con el siguiente nombre:
test_apache_playbook.yaml
- Agregar la siguiente información al archivo. Asegúrese de utilizar la sangría YAML adecuada. Cada espacio y guión es significativo. Puede perder algo de formato si copia y pega.

```
---
- hosts: webservers
  tareas:
    - name: run echo command
      command: /bin/echo hello world
```

Paso 2: Ejecute el manual de Ansible para probar su grupo de servidores web.

Ejecute el playbook de Ansible utilizando el comando **ansible-playbook** con la opción **-v** verbose. Debe ver un resultado similar a lo siguiente.

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ ansible-playbook -v test_apache_playbook.yaml
Usar /home/devasc/labs/ansible/ansible-apache/ansible.cfg como archivo de configuración

PLAY [webservers] *****

TASK [Gathering Facts] *****
ok: [192.0.2.3]

TASK [run echo command] *****
changed: [192.0.2.3] => {"changed": true, "cmd": ["/bin/echo", "hello", "world"], "delta": "0:00:00.002062", "end": "2020-05-20 21:35:32.346595", "rc": 0, "start": "2020-05-20 21:35:32.344533", "stderr": "", "stderr_lines": [], "stdout": "hello world", "stdout_lines": ["hello world"]}
```

```
PLAY RECAP *****
192.0.2.3 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$
```

Paso 3: Crear un playbook Ansible para instalar Apache.

- En VS Code, cree un nuevo archivo en el directorio **ansible-apache** con el siguiente nombre:
install_apache_playbook.yaml
- Agregue la siguiente información al archivo. Asegurarse de utilizar la extensión YAML adecuada. Cada espacio y guión es significativo. Puede perder algo de formato si copia y pega. El texto resaltado se explica en el siguiente paso.

```
---
- hosts: webserver
  become: yes
  tareas:
    - name: INSTALL APACHE2
      apt: nombre=apache2 update_cache=yes state=latest

    - name: ENABLED MOD_REWR
      apache2_module: name=rewrite state=present
      notify:
        - RESTART APACHE2

  handlers:
    - name: RESTART APACHE2
      servicio: nombre=apache2 state=reiniciado
```

Paso 4: Examine su manual de Ansible.

La siguiente es una explicación de algunas de las líneas significativas de su playbook:

- hosts: webserver** - Esto hace referencia al grupo de dispositivos de **servidores web** en el archivo de inventario de **hosts**. Este playbook se ejecutará para todos los dispositivos con este grupo.
- Become: yes** - La palabra clave **Become** activa la ejecución del comando **sudo**, lo que permitirá tareas como instalar aplicaciones.
- apt:** - El módulo **apt** se utiliza para administrar paquetes e instalaciones de aplicaciones en Linux.
- handlers:** - Los controladores son similares a una tarea, pero no se ejecutan automáticamente. Son llamados por una tarea. Observe que la tarea **ENABLED MOD_REWRITE** llama al controlador **RESTART APACHE2**.

Paso 5: Ejecute la copia de seguridad de Ansible para instalar Apache.

Ejecute el manual de Ansible utilizando el comando **ansible-playbook** con la opción **-v** verbose. La primera vez que Apache instale en su máquina virtual, la tarea **INSTALAR APACHE2** tomará entre 30 segundos y unos minutos dependiendo de la velocidad de Internet.

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ ansible-playbook -v
install_apache_playbook.yaml
Usar /home/devasc/labs/ansible/ansible-apache/ansible.cfg como archivo de
configuración
```

```
PLAY [webservers] *****

TASK [Gathering Facts] *****
ok: [192.0.2.3]

TASK [INSTALL APACHE2] *****
ok: [192.0.2.3] => {"cache_update_time": 1590010855, «cache_updated»: true, «changed»: false}

TASK [ENABLED MOD_REWRITE] *****
ok: [192.0.2.3] => {"changed": false, «result»: «Module rewrite enabled"}

PLAY RECAP *****
192.0.2.3 : ok=3 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
=
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$
```

Al Iniciar captura, debe mostrar **ok** y **failed=0** indicando una ejecución exitosa del playbook.

Paso 6: Verificar que se haya instalado Apache.

- Utilice el siguiente comando para verificar que Apache esté instalado ahora. Presione "q" para salir.

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ sudo systemctl status apache2
```

- **apache2.service - El servidor HTTP Apache**

```
Cargado: cargado (/lib/systemd/system/apache2.service; habilitado; prese>
Activo: activo (ejecución) desde Mie 2020-05-20 03:48:49 UTC; hace 10 minutos
Documentos: https://httpd.apache.org/docs/2.4/
Proceso: 8201 ExecStart=/usr/sbin/Apachectl start (código=salido, Status=0/su>
PID principal: 8225 (apache2)
Tareas: 55 (límite: 4654)
Memoria: 5,3 m.
CGroup: /system.slice/apache2.service
        8225 /usr/sbin/apache2 -k inicio
        8229 /usr/sbin/apache2 -k inicio
        8230 /usr/sbin/apache2 -k inicio
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$
```

- Abra el navegador web Chromium e introduzca la dirección IPv4 del nuevo servidor, **192.0.2.3**, para ver la página web predeterminada de Apache2.

Parte 5: Agregar opciones a su playbook de Ansible para servidores web Apache

En un entorno de producción, la instalación predeterminada de Apache2 se suele personalizar para las características específicas que necesita la organización. Un manual de Ansible también puede ayudar a automatizar estas tareas de configuración. En esta parte, personalizará su manual especificando que el servidor Apache use un número de puerto diferente.

Paso 1: Crear el manual Ansible para instalar Apache.

- En VS Code, cree un nuevo archivo en el directorio **ansible-apache** con el siguiente nombre:
install_apache_options_playbook.yaml

- b. Agregue la siguiente información al archivo. Asegúrese de utilizar la extensión YAML adecuada. Cada espacio y guión es significativo. Puede perder algo de formato si copia y pega.

```
---
- hosts(Usuarios): servidores web
  convertir en: sí
  tareas:
    - nombre: INSTALL APACHE2
      apt: nombre=apache2 update_cache=yes state=latest

    - name: ENABLED MOD_REWRITE
      apache2_module: name=rewrite state=present
      notify:
        - RESTART APACHE2

    - name: APACHE2 LISTEN ON PORT 8081
      lineinfile: dest=/etc/apache2/ports.conf regexp="^Listen 80"
      line="Listen 8081" state=present
      notify:
        - RESTART APACHE2

    - name: APACHE2 VIRTUALHOST ON PORT 8081
      lineinfile: dest=/etc/apache2/sites-available/000-default.conf
      regexp="^<VirtualHost *:80>" line="<VirtualHost *:8081>" state=present
      notify:
        - RESTART APACHE2

  handlers:
    - name: RESTART APACHE2
      service: name=apache2 state=restarted
```

Esta lista de reproducción es muy similar a la anterior con la adición de dos tareas que hacen que los servidores web escuchen en el puerto 8081 en lugar del puerto 80.

El módulo **lineinfile** se utiliza para reemplazar las líneas existentes en los archivos `/etc/apache2/ports.conf` y `/etc/apache2/sites-available/000-default.conf`. Puede buscar más información en la documentación de Ansible en el módulo **lineinfile**.

Paso 2: Examinar los dos archivos que serán modificados por el libro de jugadas.

Muestra los archivos `/etc/apache2/ports.conf` y `/etc/apache2/sites-available/000-default.conf`. Observe que el servidor web está escuchando actualmente en el puerto 80.

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ cat
/etc/apache2/ports.conf
# Si acaba de cambiar el puerto o añadir más puertos aquí, es probable que también
# debe cambiar la instrucción VirtualHost en
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
```

```
<IfModule ssl_module>
    Listen 443
<output omitted>
```

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ cat
/etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost *:80>
```

```
    # La directiva Nombre de Servidor establece el esquema de solicitud, nombre de
    host y puerto que
```

```
    # el servidor utiliza una para identificarse. Esto se usa al crear
```

```
    # URL(Uniform Resource Locator,Localizador de recursos uniforme) de
    redirección. En el contexto de hosts virtuales, Nombre del Servidor
```

```
<output omitted>
```

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$
```

Paso 3: Ejecute el playbook de Ansible.

- Ejecute el playbook de Ansible utilizando el comando **ansible-playbook**

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ ansible-playbook
install_apache_options_playbook.yaml
```

```
PLAY [webservers] *****
```

```
TASK [Gathering Facts] *****
ok: [192.0.2.3]
```

```
TASK [INSTALL APACHE2] *****
ok: [192.0.2.3]
```

```
TASK [ENABLED MOD_REWRITE] *****
ok: [192.0.2.3]
```

```
TASK [APACHE2 LISTEN ON PORT 8081] *****
ok: [192.0.2.3]
```

```
TASK [APACHE2 VIRTUALHOST ON PORT 8081] *****
ok: [192.0.2.3]
```

```
PLAY RECAP *****
192.0.2.3 : ok=5 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$
```

Paso 4: Verificar que se haya instalado Apache.

- Vea los archivos **/etc/apache2/ports.conf** y **/etc/apache2/sites-available/000-default.conf**. de nuevo. Observar que el manual modificó estos archivos para escucharlos en el puerto 8081.

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ cat
/etc/apache2/ports.conf
```

```
# Si acaba de cambiar el puerto o añadir más puertos aquí, es probable que también
# tiene que cambiar la instrucción VirtualHost en
```



```
# /etc/apache2/sites-enabled/000-default.conf
```

```
Listen 8081
```

```
<IfModule ssl_module>
```

```
    Listen 443
```

```
</IfModule>
```

```
<IfModule mod_gnutls.c>
```

```
    Listen 443
```

```
</IfModule>
```

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$
```

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$ cat  
/etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost *:8081>
```

```
    # La directiva Nombre de Servidor establece el esquema de solicitud, nombre de  
host y puerto que
```

```
    # el servidor utiliza para identificarse. Esto se usa al crear
```

```
    # URL(Uniform Resource Locator, Localizador de recursos uniforme) de  
redirección. En el contexto de hosts virtuales, Nombre del Servidor
```

```
<output omitted>
```

```
devasc @labvm: ~/labs/devnet-src/ansible/ansible-apache$
```

- b. Abra el navegador web Chromium e introduzca la dirección IPv4 del nuevo servidor. Pero esta vez especifique 8081 como el número de puerto, **192.0.2. 3:8081**, para ver la página web predeterminada de Apache2.

Nota: Aunque puede ver en el archivo `ports.conf` que Apache2 también está escuchando en el puerto 443, esto es para HTTP seguro. Aún no ha configurado Apache2 para un acceso seguro. Esto, por supuesto, se añadiría a su manual de Ansible, pero está fuera del alcance de este curso.