

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS



ÁREA DE CIENCIA DE LA COMPUTACIÓN
8° LABORATORIO - CC312

- **TÍTULO:** Utilice Ansible para automatizar la instalación de un servidor web
- **ALUMNO:**
JHONATAN POMA MARTINEZ 20182729F
- **PROFESORES:** YURI JAVIER.,CCOICCA PACASI

2023

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.

Introducción:

ANSIBLE: Ansible es una herramienta de automatización de TI (tecnología de la información), que se utiliza para administrar, configurar sistemas informáticos y redes. Permite a los administradores de sistemas automatizar tareas repetitivas, como el aprovisionamiento de servidores, la configuración de software y la implementación de aplicaciones.

Ansible utiliza un enfoque basado en lenguaje de programación para describir y automatizar la configuración de sistemas. Utiliza un lenguaje declarativo llamado YAML

PLAYBOOK de ANSIBLE: Un playbook es un archivo en formato YAML que contiene una serie de instrucciones y tareas que Ansible debe ejecutar en un conjunto de sistemas o servidores.

Un playbook de Ansible describe el estado deseado del sistema y las acciones que deben realizarse para alcanzar ese estado. Puede contener tareas como la instalación de paquetes, la configuración de archivos de configuración, la gestión de servicios, la copia de archivos, la ejecución de comandos y cualquier otra acción necesaria para configurar y administrar los sistemas de forma automatizada.

Los playbooks de Ansible lo ejecutaremos mediante el comando **`ansible-playbook`**, que toma como entrada el archivo YAML del playbook y lo ejecuta en los sistemas especificados en un inventario. Ansible se encarga de conectarse a los sistemas remotos a través de SSH y ejecutar las tareas definidas en el playbook.

APACHE2 : Es un servidor web potente y altamente configurable que permite alojar y servir contenido web en Internet. Es ampliamente utilizado debido a su estabilidad, seguridad, rendimiento y su capacidad de adaptarse a diversas necesidades y entornos.

Al instalar Apache2, obtenemos los siguientes beneficios:

1. Hospedar sitios web: Con Apache2, podemos alojar y publicar nuestros propios sitios web en Internet. Podemos crear y administrar páginas web estáticas, dinámicas o aplicaciones web complejas.
2. Procesamiento de solicitudes HTTP: Apache 2 actúa como intermediario entre el cliente (navegador web) y el servidor. Maneja las solicitudes HTTP entrantes y envía las respuestas correspondientes al cliente. Proporciona un entorno seguro y confiable para el intercambio de datos entre el servidor y los clientes.
3. Configuración y personalización: Apache2 nos ofrece una amplia gama de opciones de configuración para adaptarse a nuestras necesidades. Podemos personalizar la forma en que se sirven los archivos, configurar la seguridad, administrar el acceso a los recursos del servidor y mucho más.

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

Ya iniciado

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.

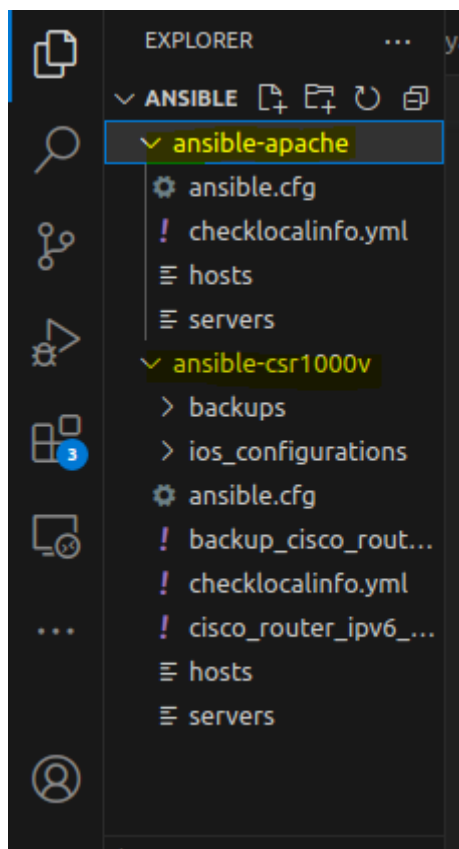
PASO 2:

Habilitamos el servidor SSH.

```
devasc@labvm: ~  
File Edit View Search Terminal Help  
devasc@labvm:~$ sudo systemctl start ssh  
devasc@labvm:~$ sudo apt-get install openssh-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
openssh-server is already the newest version (1:8.2p1-4ubuntu0.5).  
0 upgraded, 0 newly installed, 0 to remove and 587 not upgraded.  
devasc@labvm:~$ sudo spt-get install sshpass  
sudo: spt-get: command not found  
devasc@labvm:~$ sudo apt-get install sshpass  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
sshpass is already the newest version (1.06-1).  
0 upgraded, 0 newly installed, 0 to remove and 587 not upgraded.  
devasc@labvm:~$
```

PASO 3:

Abrimos la carpeta ansible , pero esta vez usaremos el directorio “**ansible-apache**”, en el laboratorio 7 usamos el directorio “**ansible-csr1000v**”.



Paso 4: Editamos el archivo de inventario de Ansible

Editamos el archivo “hosts”

```
≡ hosts ×
ansible-apache > ≡ hosts
1 [webservers]
2 192.0.2.3 ansible_ssh_user=devasc ansible_ssh_pass=Cisco123!
3
```

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:~$ 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
    link/ether 08:00:27:e9:3d:e6 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 83988sec preferred_lft 83988sec
    inet6 fe80::a00:27ff:fee9:3de6/64 scope link
        valid_lft forever preferred_lft forever
3: dummy0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/ether 6e:04:b3:b7:67:b5 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::6c04:b3ff:feb7:67b5/64 scope link
        valid_lft forever preferred_lft forever
devasc@labvm:~$
```

Paso 5: Editar el archivo ansible.cfg.

```
hosts  ansible.cfg
ansible-apache > ansible.cfg
1  # Add to this file for the Ansible lab
2  [defaults]
3
4  # Use el archivo de hosts local en esta carpeta
5  inventory=./hosts
6
7  # No se preocupe por las huellas dactilares de RSA
8  host_key_checking = False
9
10 # No crear archivos de reintento
11 retry_files_enabled = False
```

Parte 3 Verificar las comunicaciones con el 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$ ansible webservers -m ping
192.0.2.3 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

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 enviaremos el argumento **-a "/bin/echo hola world"** para pedirle al servidor web local que responda con ¡Hola, mundo!

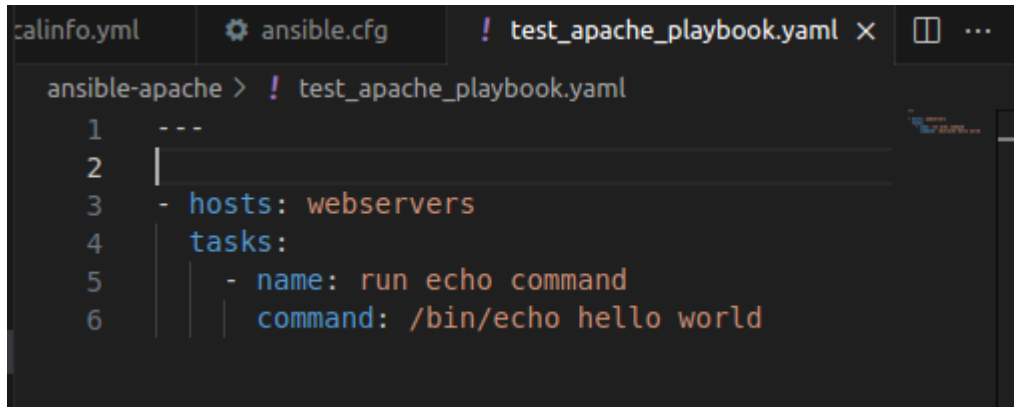
```
devasc@labvm:~/labs/devnet-src/ansible/ansible-apache$ ansible webservers -m command -a "/bin/echo hello world"
192.0.2.3 | CHANGED | rc=0 >>
hello world
devasc@labvm:~/labs/devnet-src/ansible/ansible-apache$
```

Parte 4: Crear playbooks 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 enumerará 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, se creará y automatizará la instalación del software de servidor web Apache.

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



```
calinfo.yml  ansible.cfg  ! test_apache_playbook.yaml x  ...
ansible-apache > ! test_apache_playbook.yaml
1  ---
2  |
3  - hosts: webserver
4  tasks:
5    - name: run echo command
6      command: /bin/echo hello world
```

Paso 2: Ejecuta el playbook de Ansible para probar tu grupo de servidores web



```
devasc@labvm:~/labs/devnet-src/ansible/ansible-apache$ ansible-playbook -v test_apache_playbook.yaml
Using /home/devasc/labs/devnet-src/ansible/ansible-apache/ansible.cfg as config file

PLAY [webserver] *****

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": "0:00:00.008574", "end": "2023-06-05 17:33:48.748902", "rc": 0, "start": "2023-06-05 17:33:48.740328", "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
```

Paso 3: Crear un playbook Ansible para instalar Apache.

```
test_apache_playbook.yaml  ! install_apache_playbook.yaml X ...
ansible-apache > ! install_apache_playbook.yaml
1  ---
2
3  - hosts: webserver
4    become: yes
5    tasks:
6      - name: INSTALL APACHE2
7        apt: name=apache2 update_cache=yes sta
8
9      - name: ENABLED MOD_REWRITE
10       apache2_module: name=rewrite state=pre
11       notify:
12         - RESTART APACHE2
13     handlers:
14       - name: RESTART APACHE2
15         service: name=apache2 state=restarted
```

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.

```
devasc@labvm:~/labs/devnet-src/ansible/ansible-apache$ ansible-playbook -v install_apache_playbook.yaml
Using /home/devasc/labs/devnet-src/ansible/ansible-apache/ansible.cfg as config file

PLAY [webserver] *****

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

TASK [INSTALL APACHE2] *****
changed: [192.0.2.3] => {"cache_update_time": 1685987051, "cache_updated": true, "changed": true, "stderr": "", "stderr_lines": [], "stdout": "Reading package lis
```



```
"": "init", "WantedBy": "multi-user.target", "WatchdogSignal": "6", "WatchdogTimes  
tampMonotonic": "0", "WatchdogUsec": "0"}}  
  
PLAY RECAP *****  
192.0.2.3 : ok=4 changed=3 unreachable=0 failed=0 sk  
ipped=0 rescued=0 ignored=0  
  
devasc@labvm:~/labs/devnet-src/ansible/ansible-apache$
```


Paso 6: Verificar que se haya instalado Apache.

Utilice el siguiente comando “**sudo systemctl status apache2**” para verificar que Apache esté instalado ahora. Presione "q" para salir.

```
devasc@labvm:~/labs/devnet-src/ansible/ansible-apache$ sudo systemctl status apac  
he2  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset  
   Active: active (running) since Mon 2023-06-05 17:46:03 UTC; 4min 7s ago  
     Docs: https://httpd.apache.org/docs/2.4/  
  Process: 8326 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC  
 Main PID: 8350 (apache2)  
    Tasks: 55 (limit: 3525)  
   Memory: 5.2M  
    CGroup: /system.slice/apache2.service  
            └─8350 /usr/sbin/apache2 -k start  
              └─8351 /usr/sbin/apache2 -k start  
                └─8367 /usr/sbin/apache2 -k start  
  
Jun 05 17:46:03 labvm systemd[1]: Starting The Apache HTTP Server...  
Jun 05 17:46:03 labvm systemd[1]: Started The Apache HTTP Server.  
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.

→ ↻ ⚠ Not secure | 192.0.2.3 🔍 ☆ □ 🌐 Incognito



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

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.

```

1  ---
2
3  - hosts: webservers
4    become: yes
5    tasks:
6      - name: INSTALL APACHE2
7        apt: name=apache2 update_cache=yes state=latest
8
9      - name: ENABLED MOD_REWRITE
10       apache2_module: name=rewrite state=present
11       notify:
12         - RESTART APACHE2
13
14      - name: APACHE2 LISTEN ON PORT 8081
15       lineinfile: dest=/etc/apache2/ports.conf line='Listen 8081'
16       notify:
17         - RESTART APACHE2
18
19      - name: APACHE2 VIRTUALHOST ON PORT 8081
20       lineinfile: dest=/etc/apache2/sites-available/000-default.conf line='<VirtualHost *:8081>'
21       notify:
22         - RESTART APACHE2
23
24     handlers:
25       - name: RESTART APACHE2
26         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
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<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 *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
```

Paso 3: Ejecute el playbook de Ansible.

```
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] *****
changed: [192.0.2.3]

TASK [APACHE2 VIRTUALHOST ON PORT 8081] *****
changed: [192.0.2.3]

RUNNING HANDLER [RESTART APACHE2] *****
changed: [192.0.2.3]

PLAY RECAP *****
192.0.2.3      : ok=6    changed=3    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.

Veamos 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
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /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$
```

Ahora abrimos en google , pero ingresamos 192.0.2.3:8081 :



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in** [/usr/share/doc/apache2/README.Debian.gz](#). Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the [manual](#) if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
```

RESUMEN:

PARTE 2:

En este apartado inicializamos SSH, esta conexión SSH es fundamental cuando usamos Ansible (después instalaremos Apache). Ansible utiliza esta conexión SSH para poder automatizar la gestión y configurar los servidores, mientras que SSH nos proporcionara acceso remoto y seguro para administrar y mantener servidores web con Apache. También editamos el archivo Host para que Ansible pueda conectarse y administrar, ahí le pasamos un usuario que nos servirá para la conexión SSH al servidor web y también un pass para la autenticación del usuario ssh.

También editamos el **ansible.cfg** le indicamos ciertas configuraciones como el inventario para establecer la ruta del archivo **host**(anteriormente ya le declaramos un usuario y pass y ip).

PARTE 3:

En este apartado utilizaremos el comando "ansible -m ping", para verificar la conectividad y la capacidad de respuesta de los servidores gestionados por Ansible. Realizamos una prueba de ping a los hosts definidos en ese inventario de ansible. Vemos que tras la ejecución recibimos una respuesta de éxito (pong) esto nos indica que el host es accesible y nos responderá cuando hacemos solicitudes de ansible.

PARTE 4:

“Imagine que estamos conectando cien servidores web. El grupo [webserver] del archivo hosts enumerará toda la información necesaria para cada servidor web. Pero ello podemos mejorarlo usando un playbook para verificar las comunicaciones con todos ellos con un solo comando”

Por eso creamos un archivo **test_apache_playbook.yaml** el objetivo de este playbook es ejecutar el comando "echo" con el argumento "hello world" en los servidores del grupo "webservers".

También creamos un **install_apache_playbook.yaml** que nos permitiera crear y automatizar la instalación del software de servidor web Apache.

PARTE 5:

En esta parte, personalizaremos nuestro manual especificando que el servidor Apache use un número de puerto diferente. Para eso crearemos un archivo **install_apache_options_playbook.yaml** en esta adicionaremos dos tareas que hacen que los servidores web escuchen en el puerto 8081 en lugar del puerto 80.