

Desafío #6: Modularización con Ansible

Hernan Andres Acosta

Fecha de Entrega: 21/05/2025

Objetivo

El objetivo de este desafío fue reorganizar un proyecto existente de automatización con Ansible, modularizando su estructura para permitir la reutilización del código, facilitar el mantenimiento y mejorar la escalabilidad en futuros despliegues.

Estructura del Proyecto Modularizado

```
• hernan@andres:~/Documentos/Educacion_It/clase56/desafio6/ansible$ tree
.
├── desafio6_Readme.md
├── files
│   └── index.html
├── includes
│   └── install-apache2.yml
├── inventory.ini
├── Jenkinsfile
├── Jenkinsfile-validate
├── main.yml
├── templates
│   └── ansible_site.conf.j2
├── vars
│   └── vars-site.yml
```

Guía de Instalación

Requisitos Previos

Instalación de Multipass

`sudo snap install multipass`

instalacion de ansible

`sudo apt update`

`sudo apt install -y ansible`

Configuración del Entorno

Creación de Máquina Virtual

`multipass launch --name web1 --cpus 1 --mem 1G --disk 5G`

```
web1 Running 10.170.126.79 Ubuntu 24.04 LTS
hernan@andres:~/Documentos/Educacion_It/clase56/desafio6/ansible$
```

Configuración de SSH

Paso 1 — Creación de claves SSH

El primer paso para configurar la autenticación de clave SSH en su servidor es generar un par de claves SSH en su computadora local.

Para hacer esto, podemos usar una utilidad especial llamada

```
ssh-keygen
```

```
ssh-keygen
```

que se incluye con el conjunto estándar de herramientas OpenSSH. De forma predeterminada, esto creará un par de claves RSA de 3072 bits.

```
ssh-keygen -t ed25519 -C (mas seguro ed25519)
```

En su computadora local, genere un par de claves SSH escribiendo:

```
ssh-keygen
```

```
o
```

```
ssh-keygen -t ed25519
```

Output

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/username/.ssh/id_rsa):
```

La utilidad le pedirá que seleccione una ubicación para las claves que se generarán. De forma predeterminada, las claves se almacenarán en el ~/.ssh directorio dentro de su directorio de inicio de usuarios. Se llamará la clave privada id_rsa y se llamará la clave pública asociada id_rsa.pub.

Por lo general, es mejor seguir con la ubicación predeterminada en esta etapa. Hacerlo permitirá que su cliente SSH encuentre automáticamente sus claves SSH cuando intente autenticarse. Si desea elegir una ruta no estándar, escriba ahora, de lo contrario, presione ENTER para aceptar el valor predeterminado.

```
# Si previamente había generado un par de claves SSH, puede ver un mensaje que se ve así:
```

```
Output/home/username/.ssh/id_rsa already exists.
```

```
Overwrite (y/n)?
```

```
y= para reescribirlo, lo cual borra el contenido anterior.
```

```
n= para no sobreescribirlo.
```

Ahora tiene una clave pública y privada que puede usar para autenticarse. El siguiente paso es colocar la clave pública en su servidor para que pueda usar la autenticación de clave SSH para iniciar sesión.

Copiar Su Clave Pública Manualmente

Si no tiene acceso SSH basado en contraseña a su servidor disponible, tendrá que hacer el proceso anterior manualmente.

El contenido de tu id_rsa.pub el archivo deberá agregarse a un archivo en ~/.ssh/authorized_keys en su máquina remota de alguna manera.

Para mostrar el contenido de su id_rsa.pub clave, escriba esto en su computadora local:

```
# cat ~/.ssh/id_rsa.pub
```

```
# cat ~/.ssh/id_ed25519.pub
```

Acceda a su host remoto utilizando cualquier método que tenga disponible. Esta puede ser una consola basada en la web proporcionada por su proveedor de infraestructura.

Una vez que tenga acceso a su cuenta en el servidor remoto, debe asegurarse de que ~/.ssh se crea el directorio. Este comando creará el directorio si es necesario, o no hará nada si ya existe:

```
mkdir -p ~/.ssh
mkdir -p ~/.ssh # Crea el directorio si no existe
echo "TU_CLAVE_PÚBLICA_AQUÍ" >> ~/.ssh/authorized_keys
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

Ahora, puede crear o modificar el authorized_keys archivo dentro de este directorio. Puede agregar el contenido de su id_rsa.pub archivo al final del authorized_keys archivo, creándolo si es necesario, usando esto:

```
# echo public_key_string >> ~/.ssh/authorized_keys
```

En el comando anterior, sustituya el public_key_string con la salida de la

```
cat ~/.ssh/id_rsa.pub
cat ~/.ssh/id_ed25519.pub
```

comando que ejecutó en su sistema local. Debería empezar con ssh-rsa AAAA... o similar.

Si esto funciona, puede pasar a probar su nueva autenticación SSH basada en claves.

Paso 3 — Autenticación a su Servidor Usando Claves SSH

Si ha completado con éxito uno de los procedimientos anteriores, debería poder iniciar sesión en el host remoto sin la contraseña de las cuentas remotas.

```
# ssh username@remote_host
```

en mi caso

```
ssh -v ubuntu@10.170.126.79
```

```

ubuntu@web1:~$ ls -la
total 36
drwxr-x--- 5 ubuntu ubuntu 4096 May 19 13:34 .
drwxr-xr-x 3 root root 4096 May 19 13:17 ..
drwx----- 3 ubuntu ubuntu 4096 May 19 13:26 .ansible
-rw----- 1 ubuntu ubuntu 925 May 20 12:29 .bash_history
-rw-r--r-- 1 ubuntu ubuntu 220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Mar 31 2024 .bashrc
drwx----- 2 ubuntu ubuntu 4096 May 19 13:17 .cache
-rw-r--r-- 1 ubuntu ubuntu 807 Mar 31 2024 .profile
drwx----- 2 ubuntu ubuntu 4096 May 19 13:17 .ssh
-rw-r--r-- 1 ubuntu ubuntu 0 May 19 13:26 .sudo_as_admin_successful

```

Si esta es la primera vez que se conecta a este host (si utilizó el último método anterior), puede ver algo como esto:

```

OutputThe authenticity of host '203.0.113.1 (203.0.113.1)' can't be established.
ECDSA key fingerprint is fd:fd:d4:f9:77:fe:73:84:e1:55:00:ad:d6:6d:22:fe.
Are you sure you want to continue connecting (yes/no)? yes

```

Esto significa que su computadora local no reconoce el host remoto. typee yes y luego presione ENTER para continuar.

Post de ayuda de como crear un clave privada y publica:
<https://www.digitalocean.com/community/tutorials/how-to-configure-ssh-key-based-authentication-on-a-linux-server>

Posteriormente en base a la estructura proporcionada en la clase 35 se realizaron cambios en el archivo inventory para adaptarlo a mi entorno de multipass:

```

hernan@andres:~/Documentos/Educacion_It/clase56/desafio6/ansible$ tree

```

```

.
├── desafio6_Readme.md
├── files
│   └── index.html
├── includes
│   └── install-apache2.yml
├── inventory.ini
├── Jenkinsfile
├── Jenkinsfile-validate
├── main.yml
├── templates
│   └── ansible_site.conf.j2
├── vars
│   └── vars-site.yml

```

```

# inventory.ini
#(adaptado para multipass)

```

```
[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

```
[webservers]
web1 ansible_host=10.170.126.79
```

```
[webservers:vars]
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/hernan/.ssh/id_ed25519
ansible_python_interpreter=/usr/bin/python3
```

```
*****
```

los demas archivos se mantuvieron sin cambios.

```
#files
index.html
<!DOCTYPE html>
<html>
<p>mi sitio ansible</p>
</html>
```

```
*****
```

```
includes
install-apache2.yml
- name: Actualizar el cache de paquetes
apt:
update_cache: yes
when: is_ubuntu
```

```
- name: Instalar Apache2
apt:
name: apache2
state: present
when: is_ubuntu
```

```
- name: Verificar que apache esté corriendo
service:
name: apache2
state: started
enabled: yes
when: is_ubuntu
```

```
*****
```

```
templates
ansible_site.conf.j2
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/ansible_site

<Directory /var/www/html/ansible_site>
```

```
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
*****
```

```
main.yml
```

```
- name: Deployment de un sitio estático
hosts: all
become: yes
pre_tasks:
- name: Verificar si el OS es Ubuntu
set_fact:
is_ubuntu: "{{ ansible_distribution == 'Ubuntu' }}"
vars_files:
- vars/vars-site.yml
tasks:
- name: Instalar servicio Apache
include_tasks: includes/install-apache2.yml
- name: Crear directorio para el sitio
file:
path: "{{ site_dir }}"
state: directory
owner: www-data
group: www-data
when: is_ubuntu
- name: Copiar index.html al directorio del sitio
copy:
src: files/index.html
dest: "{{ site_dir }}"
owner: www-data
group: www-data
mode: '0644'
when: is_ubuntu
- name: Configuración del sitio apache
template:
src: templates/ansible_site.conf.j2
dest: /etc/apache2/sites-available/ansible_site.conf
notify:
- Reload Apache
when: is_ubuntu
- name: Activar el nuevo sitio
command: a2ensite ansible_site.conf
notify:
- Reload Apache
when: is_ubuntu
```

- name: Deshabilitar el sitio default
command: a2dissite 000-default.conf
notify:
- Reload Apache
when: is_ubuntu
- name: Cambiar el e-mail del webmaster
lineinfile:
path: /etc/apache2/sites-available/000-default.conf
regexp: 'ServerAdmin webmaster@localhost'
line: 'ServerAdmin webmaster@educacionit.com'
when: is_ubuntu
notify:
- Reload Apache
handlers:
- name: Reload Apache
service:
name: apache2
state: reloaded
when: is_ubuntu

```
*****
```

vars
vars-site.yml
site_dir: /var/www/html/ansible_site # Ruta del sitio
apache_conf: ansible_site.conf # Nombre del archivo de configuración

```
*****  
*****
```

verificando conectividad
ansible all -i inventory.ini -m ping

hernan@andres:~/Documentos/Educacion_It/clase56/desafio6/ansible\$ ansible all -i
inventory.ini -m ping-m ping
web1 | SUCCESS => {
 "changed": false,
 "ping": "pong"

```
hernan@andres:~/Documentos/Educacion_It/clase56/desafio6/ansible$ ansible all -i inventory.ini -m ping  
web1 | SUCCESS => {  
  "changed": false,  
  "ping": "pong"  
}
```

```
*****
```

Ejecutar playbook:
ansible-playbook -i inventory.ini main.yml

```
# hernan@andres:~/Documentos/Educacion_It/clase56/desafio6/ansible$ ansible-playbook -i
inventory.ini main.yml #
```

arrojando como resultado:

```
TASK [Actualizar el cache de paquetes]
*****
changed: [web1]
```

```
TASK [Instalar Apache2]
*****
**
ok: [web1]
```

```
TASK [Verificar que apache esté corriendo]
*****
ok: [web1]
```

```
TASK [Crear directorio para el sitio]
*****
ok: [web1]
```

```
TASK [Copiar index.html al directorio del sitio]
*****
ok: [web1]
```

```
TASK [Configuración del sitio apache]
*****
ok: [web1]
```

```
TASK [Activar el nuevo sitio]
*****
changed: [web1]
```

```
TASK [Deshabilitar el sitio default]
*****
changed: [web1]
```

```
TASK [Cambiar el e-mail del webmaster]
*****
ok: [web1]
```

```
RUNNING HANDLER [Reload Apache]
*****
changed: [web1]
```

```
PLAY RECAP
*****
*****
web1 : ok=13 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```



```
*****
RUNNING HANDLER [Reload Apache] *****
changed: [web1]

PLAY RECAP *****
web1                : ok=13  changed=4  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

hernan@andres:~/Documentos/Educacion_It/clase56/desafio6/ansible$
```

```
#####
#####
```

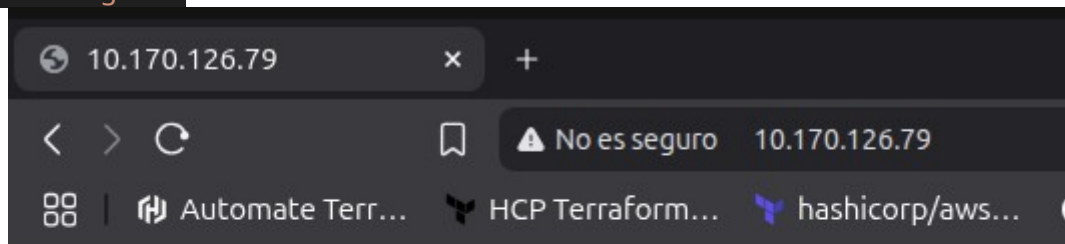
seguidamente se expuso el servidor web: ip 10.170.126.79

curl http://10.170.126.79

```
hernan@andres:~/Documentos/Educacion_It/clase56/desafio6/ansible$ curl http://10.170.126.79
<!DOCTYPE html>
<html>
  <p>mi sitio ansible</p>
  </html>

```

y desde el navegador:



mi sitio ansible

Conclusión

Este desafío demostró la importancia de estructurar proyectos de automatización de manera modular, siguiendo buenas prácticas de Infraestructura como Código (IaC). La implementación exitosa del playbook, junto con la configuración de un entorno reproducible, refuerza los conceptos clave de Ansible en un contexto de DevOps, preparando el camino para automatizaciones más complejas en entornos productivos.

Anexo

Guía de Adaptación del Proyecto a AWS

Requisitos Previos para AWS

Instalar AWS CLI (si no está instalado)

```
sudo apt install -y awscli
```

```
pip install boto3 boto
```

Autenticación en AWS

```
aws configure
```

Ingresar:

- AWS Access Key ID

- AWS Secret Access Key

- Región (ej: us-east-1)

- Formato de output (json)

Crear Par de Claves SSH

```
aws ec2 create-key-pair --key-name ansible-aws-key --query 'KeyMaterial' --output text > ~/.ssh/ansible-aws-key.pem
```

```
chmod 400 ~/.ssh/ansible-aws-key.pem
```

Modificaciones al Proyecto

Nuevo Archivo de Inventario (inventory/aws_hosts)

```
[webservers:children]
```

```
aws_ec2
```

```
[aws_ec2]
```

```
# Se llenará dinámicamente
```

```
[aws_ec2:vars]
```

```
ansible_user=ubuntu
```

```
ansible_ssh_private_key_file=~/.ssh/ansible-aws-key.pem
```

Archivo de Configuración para AWS (aws_ec2.yml)

```
plugin: aws_ec2
```

```
regions:
```

```
- us-east-1
```

```
filters:
```

```
tag:Name: webserver-ansible
```

```
keyed_groups:
```

```
- key: tags.Name
```

```
compose:
```

```
ansible_host: public_ip_address
```

Despliegue en AWS 4.1.

Crear Instancias EC2

```
aws ec2 run-instances \
```

```
--image-id ami-08c40ec9ead489470 \ # Ubuntu 22.04 LTS
--count 2 \
--instance-type t2.micro \
--key-name ansible-aws-key \
--security-group-ids sg-xxxxxxx \
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=webserver-ansible}]'
```

Configurar Security Group

```
aws ec2 authorize-security-group-ingress \
  --group-id sg-xxxxxxx \
  --protocol tcp \
  --port 22 \
  --cidr 0.0.0.0/0
```

```
aws ec2 authorize-security-group-ingress \
  --group-id sg-xxxxxxx \
  --protocol tcp \
  --port 80 \
  --cidr 0.0.0.0/0
```

Ejecutar Playbook con Inventario Dinámico

```
ansible-playbook -i aws_ec2.yml main.yml
```

Modificaciones Recomendadas para AWS

Variables Específicas para AWS (vars/aws.yml)

```
site_dir: /var/www/html/aws_site
```

```
apache_conf: aws_site.conf
```

```
server_admin_email: admin@aws.example.com
```

Actualizar el Playbook Principal

- name: Deployment para AWS

hosts: aws_ec2

become: yes

vars_files:

- vars/aws.yml

... resto del playbook igual ...

Obtener IP pública de las instancias

aws ec2 describe-instances --query "Reservations[*].Instances[*].PublicIpAddress" --
output text

Probar acceso web

curl http://<IP_PUBLICA_EC2>

Cost Optimization

Comando para detener instancias después de pruebas

aws ec2 stop-instances --instance-ids i-xxxxxxx i-yyyyyyy

Notas Clave Seguridad:

Usar IAM Roles en lugar de credenciales permanentes

Rotar claves SSH periódicamente

Escalabilidad:

Considerar usar Auto Scaling Groups

Implementar un ELB para balanceo de carga

Costos:

Usar instancias Spot para entornos de prueba

Programar paradas automáticas con AWS Instance Scheduler