12 Vault

Ansible Vault es una funcionalidad de Ansible que permite **proteger información sensible** (como contraseñas, claves API o certificados) mediante el uso de archivos cifrados que pueden integrarse fácilmente en los playbooks.

Ansible Vault encripta archivos que contienen variables delicadas, normalmente guardados en ficheros llamados `vault.yaml` o `vault.yml`.

Para cifrar este tipo de archivos se utiliza el comando:

```
ansible-vault encrypt nombre_del_fichero.yml
```

Una vez cifrado, también puede:

- **Desencriptarse**: `ansible-vault decrypt nombre_del_fichero.yml`
- **Editarse directamente** (editándolos temporalmente en texto plano y re-cifrando al guardar): `ansible-vault edit nombre_del_fichero.yml`

Podemos encriptar el contenido del fichero de la siguiente forma:

```
ansible@debian64:~/practicas/pr05/vault$ ansible-vault encrypt vault.yaml
New Vault password:
Confirm New Vault password:
Encryption successful
```

Podemos desencriptar el contenido de la siguiente forma:

```
ansible@debian64:~/practicas/pr05/vault$ ansible-vault decrypt vault.yaml
Vault password:
Decryption successful
```

Uso en playbooks

Los archivos cifrados se incluyen dentro del playbook igual que cualquier otro archivo de variables. Sin embargo, dado que el archivo está cifrado, Ansible necesita conocer la clave de descifrado **en tiempo de ejecución**. Existen dos formas de manejar la clave:

1. **Uso de un archivo de clave**:

```
ansible-playbook --vault-password-file /path/to/my/vault-password-file site.yml
```

```
ansible@debian64:~/practicas/pr05/vault$ ansible-playbook --vault-password-file
vault_password.txt playbook.yaml
```

2. **Solicitud interactiva de la clave** configurando en el archivo `ansible.cfg` la siguiente directiva:

```
ask_vault_pass = True
```

En este modo, Ansible pedirá la contraseña una sola vez al inicio de la ejecución, incluso si se usan varios vaults.

```
ansible@debian64:~/practicas/pr05/vault$ cat vault.yaml
secret_password: "abc123."
```

3. **Solicitud de clave desde linea de comandos**

Si nosotros por ejemplo tenemos un fichero con contenido encriptado y necesitamos hacer uso de estos datos en un playbook es necesario desencriptar su contenido para poder ejecutar el playbook. Para ello debemos pedir al usuario la contraseña del fichero con datos encriptados con la orden *--ask-vault-pass*.

```
ansible-playbook --ask-vault-pass playbook.yaml
```

Tenemos el fichero *vault.yaml* con el siguiente contenido.

```
$ANSIBLE_VAULT;1.1;AES256
663838656632643236643331656239383437326133346639633532623539326664303466613532316
643137396435376435643739613765356136336666643838 0a623935326133306438666263386237
306566626632353065393231613533393263393566636338316339303565333431336136326632356
262383231663435300a316436363337343632356238366131373662353231353466346631343839
373463303139363164336436383430343339616562373266663261623130333933 36
```

Tenemos el siguiente playbook que precisa de dicho conenido encriptado.

```
---
- name: Ansible Vault
  hosts: srv1
  tasks:
  - name: Incluir fichero de datos encriptados
    ansible.builtin.include_vars:
      file: vault.yaml

  - name: Mostrar el contenido de Ansible Vault
```

```
    ansible.builtin.debug:
      msg: "La contraseña una vez desencriptada es: {{ contraseña }}"
```

Es necesario que se pida la contraseña de *vault.yaml* al usuario.

```
ansible@debian64:~/practicas/pr05/vault$ ansible-playbook --ask-vault-pass
playbook.yaml
Vault password:

PLAY [Ansible Vault]
**********************************************************************************
*************************************************************

TASK [Gathering Facts]
**********************************************************************************
*************************************************************
ok: [srv1]

TASK [Incluir fichero de datos encriptados]
**********************************************************************************
****************************************
ok: [srv1]

TASK [Mostrar el contenido de Ansible Vault]
**********************************************************************************
****************************************
ok: [srv1] => {
    "msg": "La contraseña una vez desencriptada es: abc123."
}

PLAY RECAP
**********************************************************************************
**************************************************************************
srv1                       : ok=3    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Ejercicios para repasar los conceptos anteriores

Se propone la Tarea 6 para repasar los conceptos vistos hasta este punto.