

# Ansible

- [Actualización de varios servidores con Ansible usando Vault](#)

# Actualización de varios servidores con Ansible usando Vault

Automatiza la actualización de paquetes en varios servidores usando **Ansible** y **Ansible Vault** para gestionar contraseñas de forma segura.

## Paso 1: Crear el inventario (`ansihosts`)

Define los servidores que vas a administrar:

```
[servers]
server1 ansible_host=192.0.2.1 ansible_user=myuser
server2 ansible_host=192.0.2.2 ansible_user=myuser
```

## Paso 2: Crear `host_vars` protegidos con Vault

```
mkdir host_vars
ansible-vault create host_vars/server1.yml
```

Contenido ejemplo:

```
ansible_ssh_pass: "password123"
ansible_become_pass: "rootpassword"
```

Repite para cada servidor:

```
ansible-vault create host_vars/server2.yml
```

---

## Paso 3: Crear el playbook (update.yml)

```
- name: Update and upgrade using nala
  hosts: all
  become: yes
  tasks:
    - name: Run nala update and upgrade
      shell: nala update && nala upgrade -y
      register: nala_output

    - name: Show output of nala update and upgrade
      debug:
        var: nala_output.stdout
```

“ Si no usas `nala`, cambia por `apt-get update && apt-get upgrade -y`.

---

## Paso 4: Ejecutar el playbook

```
ansible-playbook -i ansihosts update.yml --ask-vault-pass
```

---

## Paso 5: Salida detallada (verbose)

```
ansible-playbook -i ansihosts update.yml --ask-vault-pass --vvv
```

---

## Salida esperada

```
PLAY [Update and upgrade using nala] *****

TASK [Gathering Facts] *****
ok: [server1]
ok: [server2]
```

```
TASK [Run nala update and upgrade] *****
changed: [server1]
changed: [server2]

TASK [Show output of nala update and upgrade] *****
ok: [server1] => {
  "nala_output.stdout": "All packages are up to date."
}
ok: [server2] => {
  "nala_output.stdout": "All packages are up to date."
}

PLAY RECAP *****
server1 : ok=3    changed=1    unreachable=0    failed=0
server2 : ok=3    changed=1    unreachable=0    failed=0
```

## Extras útiles

### Automatizar el uso de Vault (opcional)

1. Crear archivo con la contraseña:

```
echo "contraseña" > ~/.ansible_vault_key
chmod 600 ~/.ansible_vault_key
```

2. Configurar `ansible.cfg`:

```
[defaults]
vault_password_file = ~/.ansible_vault_key
```

### Añadir más servidores

- Edita el archivo `ansihosts`.
- Crea nuevos archivos en `host_vars/` y protégelos con Vault.

### Modificar el playbook

- Puedes adaptarlo para tareas adicionales, no solo actualizaciones.

## Sobre el verbose (`-v`, `-vv`, `--vvv`)

- Usa `--vvv` para depurar comportamientos inesperados.

## Seguridad

- Protege siempre las contraseñas y datos sensibles con Vault u otro sistema seguro.
- 

## Nota personal

Uso este sistema con mi servidor principal y una Raspberry Pi. Aunque es un setup sencillo, me ahorra tiempo en tareas rutinarias. Es especialmente útil si gestionas varios servidores.