

Nodos gestionados

A continuación explicamos el proceso de configuración de los nodos que serán administrados desde el nodo controlador donde se ha instalado Ansible. Para la gestión de estos nodos necesitamos:

- **Paso 1:** Python instalado.
- **Paso 2:** Tener un servidor SSH.
- **Paso 3:** Crear un usuario para trabajar y que tenga permisos de SUDO para poder lanzar comandos de ROOT.
- **Paso 4:** Copiar las claves SSH desde el del usuario en el ControlNode con el usuario que quiero trabajar en estos nodos.
- **Paso 5:** En caso de hacer uso de un usuario *ansible* necesitamos crearlo y realizar configuraciones en el fichero */etc/sudoers*.

Instalación de Python

En nuestro caso, la propia máquina ya cuenta con el interprete de Python instalado como podemos comprobar a continuación.

```
user@srvTemplate:~$ python3 --version
Python 3.11.2
```

Servidor SSH

En nuestro caso, la propia máquina ya cuenta con un servidor SSH instalado. En caso de no existir dicho servicio, sería necesaria su instalación.

```
$ apt update
$ apt install openssh-server -y
```

```
user@srvTemplate:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
  Active: active (running) since Fri 2025-09-19 10:58:58 CEST; 9min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
   Process: 443 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 453 (sshd)
    Tasks: 1 (limit: 1108)
   Memory: 9.2M
      CPU: 178ms
     CGroup: /system.slice/ssh.service
             └─453 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

```
Warning: some journal files were not opened due to insufficient permissions.
```

Creación del usuario ansible en el ControlNode

Nota: Previamente a crear el usuario instalamos el paquete *whois* para poder crear la contraseña cifrada.

Creamos en el *ControlNode* y en el *srvTemplate* el usuario **ansible**.

En el *ControlNode*:

```
root@ControlNode:~# apt install -y whois
...

```

```
root@ControlNode:~# useradd -m -d /home/ansible -p "$(mkpasswd 'ansible')" -s
/bin/bash ansible
root@ControlNode:~# grep ansible /etc/passwd
ansible:x:1001:1001::/home/ansible:/bin/bash
```

En el *srvTemplate*:

```
root@srvTemplate:~# apt install -y whois
...

```

```
root@srvTemplate:~# useradd -m -d /home/ansible -p "$(mkpasswd 'ansible')" -s
/bin/bash ansible
root@srvTemplate:~# grep ansible /etc/passwd
ansible:x:1001:1001::/home/ansible:/bin/bash
```

Creacion del par de claves SSH

Creamos el par de claves SSH para poder conectarnos sin necesidad de contraseña.

```
ansible@ControlNode:~$ ssh-keygen
...
ansible@ControlNode:~$ ls .ssh/
id_rsa  id_rsa.pub
ansible@ControlNode:~$ ssh-copy-id -i .ssh/id_rsa.pub ansible@192.168.122.5
...
```

En el *srvTemplate* se ha copiado la clave pública.

```
ansible@srvTemplate:~$ ls .ssh/
authorized_keys
ansible@srvTemplate:~$ cat .ssh/authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCN0L35BTY48oCCdfx9tMXMfNe9mphXzUZBSAHEFUJhXPD+Te7w8R
7WVxrD70ekAN7rNm1viGFJ5YKw8/qLRR8DRyjZy50dzDzA422QQxemriGc8w705sqH0rXCrack1Xzb9/4
ny1kBQqjpF1PgqM39wLQ2Z8NnwaQKtBLQdrzExqh8Zkq0RZc4QcX6/vZnoy07L4JYZ3WDzg9PrAoHkS3
Z/ch1IwJQw9TwGBYxvRjYWB5f1+Mfv2o4AS8dRsUBCcAg8W3WR2hCsWzVYyoZv82dXr1CZG1QX23zRDI8S
+87mk2wd1SwXrYRGnT+HNXIRI0FeQ9emE+i862Rrnq3DLA+oKFK3ojsYWdfibzbnni8xj0YQobasTdnVnP
F1RL1KMz9noaD7yIPsD8JQHex+wOW4BddN+vyMaJBcYa7L01IM80upUmubw0VeR1USkP0Bv7GFzb1S+tHu
NrteQ07nCs1fVTyazLwdL7wuaFqluIzR7xuJzIH9VC5mi3xiNf8= ansible@ControlNode
```

En este punto puedo conectarme a la máquina remota con el usuario *ansible*.

```
ansible@ControlNode:~$ ssh 192.168.122.5
...
ansible@srvTemplate:~$
```

Nota: Esto es solo para el usuario *ansible*, el resto de usuarios necesitarán contraseña para acceder. Además es unidireccional de *ControlNode* a *srvTemplate*.

```
ansible@ControlNode:~$ ssh user@192.168.122.5
user@192.168.122.5's password:
```

Instalación de sudo para poder tener permisos de administración

Instalamos sudo.

```
root@srvTemplate:~# apt update && apt install -y sudo
```

Con ayuda de la herramienta *sudo* modificamos el fichero [/etc/sudoers](#) para darle permisos al usuario *ansible*.

```
root@srvTemplate:~# visudo
root@srvTemplate:~# grep ansible /etc/sudoers
ansible ALL=(ALL:ALL) NOPASSWD:ALL
```

En este punto podemos hacer uso de *sudo* con el usuario *ansible* en *srvTemplate*.

Comprobación de la conexión

Llegados a este punto se va a trabajar haciendo clonaciones enlazadas de *srvTemplate*. Vamos a crear una máquina de prueba llamada *srv1*, la cual tendrá IP por DHCP y en ella vamos a crear un primer proyecto, más adelante se verá en detalle cada uno de los pasos.

En este punto vamos a comprobar si podemos hacer ping a nuestra máquina *srv1*.

```
ansible@ControlNode:~/practicas/pr01$ cd practicas/pr01/
ansible@ControlNode:~/practicas/pr01$ ls
ansible.cfg  hosts

ansible@ControlNode:~/practicas/pr01$ cat ansible.cfg
[defaults]
inventory = hosts

ansible@ControlNode:~/practicas/pr01$ cat hosts
srv1 ansible_host=192.168.10.6 ansible_user=ansible ansible_become=yes
```

Ejecutamos el módulo ping y vemos si tenemos conexión con la máquina.

```
ansible@ControlNode:~/practicas/pr01$ ansible -i hosts srv1 -m ping
[WARNING]: Host 'srv1' is using the discovered Python interpreter at
'/usr/bin/python3.11', but future installation of another Python interpreter could
cause a different interpreter to be discovered. See
https://docs.ansible.com/ansible-core/2.19/reference\_appendices/interpreter\_discovery.html for more information.
srv1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.11"
    },
    "changed": false,
    "ping": "pong"
}
```

Vemos que tenemos una conexión exitosa con nuestro nodo *srv1*. Si hemos llegado a este punto tenemos el entorno de trabajo configurado.

Podemos ejecutar un playbook de prueba, más adelante explicaremos en detalle cual es el funcionamiento del mismo.

```
---
- name: Playbook inicial
  hosts: srv1

  tasks:
    - name: Instalar los paquetes zip y unzip
      ansible.builtin.apt:
        name:
          - zip
```

```
- unzip
  state: present

- name: Crear un fichero /home/user/ejemplo con permisos 777
  ansible.builtin.file:
    path: /home/user/ejemplo
    state: touch
    owner: root
    group: root
    mode: "0777"

- name: Añadir la cadena hola al fichero /home/user/ejemplo
  ansible.builtin.shell: |
    if test "$(tail -1 /home/user/ejemplo)" != "hola"
    then
      echo "hola" >> /home/user/ejemplo
    fi

- name: Añadir un fichero con contenido
  ansible.builtin.copy:
    dest: /home/user/copia
    content: |
      Hola que tal
      Espero que bien.
      Un saludo
      Martin
```

Si ejecutamos el playbook podemos ver que se lanza con éxito.

```
ansible@ControlNode:~/practicas/pr01$ ansible-playbook playbook.yaml

PLAY [Playbook inicial]
*****
TASK [Gathering Facts]
*****
ok: [srv1]

TASK [Instalar los paquetes zip y unzip]
*****
changed: [srv1]

TASK [Crear un fichero /home/user/ejemplo con permisos 777]
*****
changed: [srv1]

TASK [Añadir la cadena hola al fichero /home/user/ejemplo]
*****
changed: [srv1]

TASK [Añadir un fichero con contenido]
*****
```

```
changed: [srv1]
```

```
PLAY RECAP
```

```
*****
```

```
srv1 : ok=5    changed=4    unreachable=0    failed=0
      skipped=0   rescued=0   ignored=0
```

Si nos conectamos a *srv1* podemos ver el resultado de esta ejecución.

```
ansible@srv1:~$ sudo ls /home/user/ejemplo
/home/user/ejemplo
ansible@srv1:~$ sudo cat /home/user/ejemplo
hola
ansible@srv1:~$ sudo ls -l /home/user/ejemplo
-rwxrwxrwx 1 root root 5 nov 4 09:21 /home/user/ejemplo
ansible@srv1:~$ cat /home/user/copia
Hola que tal
Espero que bien.
    Un saludo
    Martin
```