

# Ansible

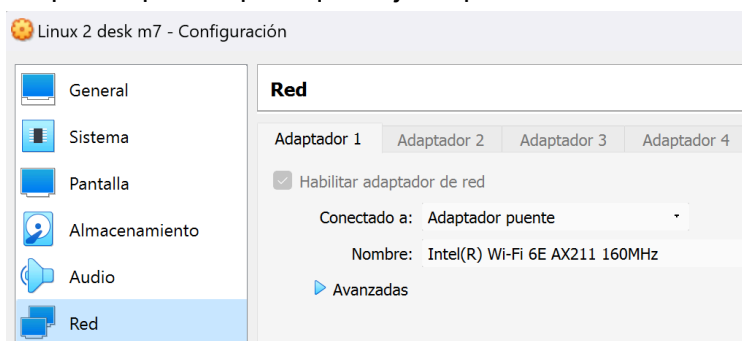


# Introduccion

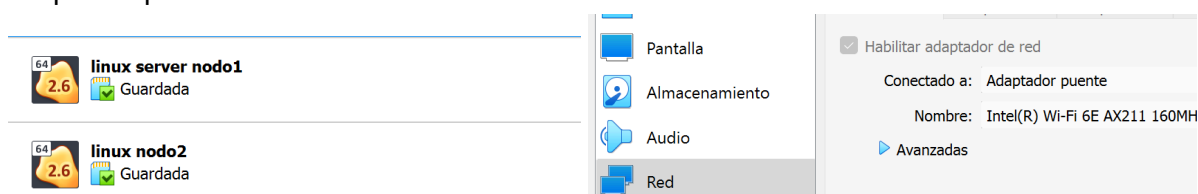
Ansible es una herramienta potente para simplificar y automatizar la gestión y el despliegue de infraestructuras y aplicaciones, reduciendo la complejidad y aumentando la eficiencia en entornos de TI. Ansible utiliza SSH para conectarse a los servidores objetivo y ejecutar las tareas definidas. Esto hace que sea fácil de implementar, ya que no requiere ningún agente permanente instalado en los servidores.

## Maquinas

Para empezar nos hará falta la máquina controladora con interfaz grafica y la red en adaptador puente para que coja la ip automáticamente



Los siguientes nodos deben ser sin interfaz por ejemplo linux server y ambos deben tener adaptador puente



# Instalacion

Comenzaremos instalando Ansible en el controlador

```
root@adnan-VirtualBox:/home/adnan# apt install ansible
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  ieee-data python-babel-localedata python3-argcomplete python3-babel
python3-distutils python3-dnspython python3-jinja2 python3-jmespath
python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
python3-packaging python3-pycryptodome python3-requests-kerberos
python3-requests-ntlm python3-requests-toolbelt python3-selinux
python3-simplejson python3-winrm python3-xlrd python3-xlsxwriter
Paquetes sugeridos:
  cowsay sshpass python3-sniffio python3-trio python-jinja2-doc ipython3
python-netaddr-docs
Se instalarán los siguientes paquetes NUEVOS:
  ansible ieee-data python-babel-localedata python3-argcomplete python3-babel
python3-distutils python3-dnspython python3-jinja2 python3-jmespath
python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
python3-packaging python3-pycryptodome python3-requests-kerberos
python3-requests-ntlm python3-requests-toolbelt python3-selinux
python3-simplejson python3-winrm python3-xlrd python3-xlsxwriter
0 actualizados, 22 nuevos se instalarán, 0 para eliminar y 218 no actualizados.
Se necesita descargar 28.2 MB de archivos.
```

**Comenta las diferencias que hay entre la versión antes y después de instalar Ansible con "pip"?**

## Version anterior

- No puedes invocar Ansible para realizar tareas de automatización o gestión de configuración.
- No se instalan las dependencias de Ansible. Esto significa que debes gestionar manualmente las dependencias requeridas por Ansible, como Python y otros módulos necesarios.

## Version nueva

- Ahora puedes utilizar Ansible para realizar tareas de automatización y gestión de configuración.
- No necesitas gestionar manualmente las dependencias, ya que Pip lo hace por ti.
- Actualizar fácilmente Ansible a una nueva versión cuando esté disponible. Esto te permite mantenerse al día con las últimas características y correcciones de errores de Ansible.
- Gestionar la versión de Ansible que deseas instalar. Puedes especificar una versión específica o instalar la última versión disponible, según tus necesidades.

Veremos en que version esta el ansible con el comando **ansible --version**

```
root@adnan-VirtualBox:/home/adnan# ansible --version
ansible 2.10.8
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
root@adnan-VirtualBox:/home/adnan#
```

Instalaremos el python3-pip para instalar la versiones más actuales de python

```
root@adnan-VirtualBox:/home/adnan# apt install python3-pip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-11 gcc gcc-11 javascript-common
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6
  libc6-dbg libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl
  libgcc-11-dev libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libnsl-dev libpython3-dev libpython3.10
  libpython3.10-dev libpython3.10-minimal libpython3.10-stdlib libquadmath0 libstdc++-11-dev libtirpc-dev libtsan0 libubsan1
  linux-libc-dev lto-disabled-list make manpages-dev python3-dev python3-setuptools python3-wheel python3.10 python3.10-dev
  python3.10-minimal rpcsvc-proto zlib1g-dev
Paquetes sugeridos:
  binutils-doc debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf automake libtool flex bison gcc-doc
  gcc-11-multilib gcc-11-locales apache2 | lighttpd | httpd glibc-doc git bzr libstdc++-11-doc make-doc python-setuptools-doc
  python3.10-venv python3.10-doc binfmt-support
Paquetes recomendados:
  libnss-nis libnss-nisplus
Se instalarán los siguientes paquetes NUEVOS:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-11 gcc gcc-11 javascript-common
```

Después instalaremos pip3 install ansible también para instalar la versiones más reciente de Ansible

```
root@adnan-VirtualBox:/home/adnan# pip3 install ansible
Collecting ansible
  Downloading ansible-9.3.0-py3-none-any.whl (46.3 MB)
    17.8/46.3 MB 3.1 MB/s eta 0:00:10
```

## Creación de claves públicas y privadas

Creamos un clave ssh pública sin utilizar el root para que se genere en el home de nuestro usuario y lo enlazaré a nuestro correo electrónico

```
root@adnan-VirtualBox:/home/adnan# ssh-keygen -t ed25519 -C e.aamranib@ies-sabadell.cat
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
```

**Quins algorismes d'enciptació existeixen a més del que acabem d'utilitzar? Quina diferència hi ha entre ells? Quins avantatges té la utilització de l'algoritme "ED" respecte al "RSA"?**

Existen varios algoritmos de cifrado además del que acabamos de usar. Algunos de los más comunes incluyen RSA, DSA, ECDSA, y AES. Cada uno de estos algoritmos tiene sus propias fortalezas y debilidades, y la elección del mejor depende de los requisitos específicos de la aplicación.

La principal diferencia entre estos algoritmos radica en cómo generan sus claves y cómo cifran y descifran los datos. Por ejemplo, RSA y DSA son algoritmos de cifrado asimétrico que utilizan un par de claves (una clave pública y una clave privada) para el cifrado y descifrado. Por otro lado, AES es un algoritmo de cifrado simétrico que utiliza la misma clave para el cifrado y descifrado.

**La utilización del algoritmo ED 25519 frente a RSA tiene varias ventajas:**

Seguridad: ED25519 es considerado uno de los algoritmos de firma digital más seguros disponibles actualmente.

Eficiencia: ED25519 es más eficiente en términos de tiempo de procesamiento y consumo de recursos que RSA.

Tamaño de clave más pequeño: Las claves generadas por ED25519 son más cortas que las claves RSA equivalentes, lo que reduce el almacenamiento y la carga de datos necesarios para la gestión de claves.

### Explica que es una «passphrase» i quin és el seu objectiu.

Una "passphrase" es una secuencia de palabras utilizada como contraseña para acceder a sistemas digitales. Su objetivo es mejorar la seguridad haciendo que la contraseña sea más difícil de adivinar.

Con este comando podremos ver que la clave está creada, se utiliza para generar un par de claves SSH utilizando el algoritmo de cifrado Ed25519. Este algoritmo proporciona una mayor seguridad y eficiencia.

```
adnan@adnan-VirtualBox:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/adnan/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/adnan/.ssh/id_ed25519
Your public key has been saved in /home/adnan/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:U6dBv9mwgjoAhZzKNzk+JwpyrbREHcjLC4X48z7K9Q adnan@adnan-VirtualBox
The key's randomart image is:
+--[ED25519 256]--+
| o +.      . |
| *..      . . |
| .=..... o + |
| +.**.     o + * |
| .*o=+    S o + . |
| +.X+Eo . . . |
| +*.++ o |
| . = . |
| oo. |
+-----[SHA256]-----+
```

Este comando se utilizara para obtener el PID del agente y añadir una identidad a nuestra clave privada.

```
root@adnan-VirtualBox:/home/adnan# eval $(ssh-agent -s)
Agent pid 6009
root@adnan-VirtualBox:/home/adnan#
```

# Configuración de Nodo 1

Actualizaremos el nodo 1 todos los paquetes.

```
node1@node1:~$ sudo apt update
[sudo] password for node1:
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Descargados 229 kB en 1s (171 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se pueden actualizar 61 paquetes. Ejecute «apt list --upgradable» para verlos.
node1@node1:~$ _
```

Instalaremos con el comando openssh-server como veis yo lo tengo instalado y muestro la versión que está.

```
adnan@adnan-VirtualBox:~$ ssh -V
OpenSSH_8.9p1 Ubuntu-3ubuntu0.3, OpenSSL 3.0.2 15 Mar 2022
adnan@adnan-VirtualBox:~$
```

Nos conectaremos al nodo 1 desde la máquina controladora mediante ssh con el comando ssh el nombre de la maquina@IP (Para salir solo pondremos **exit**)

```
adnan@adnan-VirtualBox:~$ ssh node1@192.168.1.183
The authenticity of host '192.168.1.183 (192.168.1.183)' can't be established.
ED25519 key fingerprint is SHA256:sh3eS9awUvDbF/9AUzwdc4z1uNp+CkvfzizY6Bza03A.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.183' (ED25519) to the list of known hosts.
node1@192.168.1.183's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of lun 25 mar 2024 12:01:49 UTC

System load:  0.080078125      Processes:            163
Usage of /:   55.4% of 9.39GB   Users logged in:     1
Memory usage: 33%             IPv4 address for enp0s3: 192.168.1.183
Swap usage:   1%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge
```

Desde la máquina controladora copiaremos la clave pública mediante ssh

```
adnan@adnan-VirtualBox:~$ ssh-copy-id -i /home/adnan/.ssh/id_ed25519.pub adnan@10.1.23.35
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/adnan/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
adnan@10.1.23.35's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'adnan@10.1.23.35'"
and check to make sure that only the key(s) you wanted were added.
```

Volveremos a conectarnos al nodo 1 (Sin contraseña) y dentro del nodo podemos comprobar creado un archivo en el controlador

```
adnan@adnan-VirtualBox:~$ ssh -l adnan 10.1.23.42
The authenticity of host '10.1.23.42 (10.1.23.42)' can't be established.
ED25519 key fingerprint is SHA256:IWzdboEIgik9ckbm0ue7khKNP0gqbiqqphdk92vEdxs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.1.23.42' (ED25519) to the list of known hosts.
adnan@10.1.23.42's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-97-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro
```

System information as of vie 01 mar 2024 18:18:33 UTC

System load:	0.04296875	Processes:	174
Usage of /:	69.4% of 6.99GB	Users logged in:	1
Memory usage:	13%	IPv4 address for enp0s3:	10.1.23.42
Swap usage:	0%		

```
root@adnan:/home/adnan# ls
hola
root@adnan:/home/adnan# _
```

```
Last login: Fri Mar  1 18:14:29 2024 from 10.1.23.42
adnan@adnan:~$ ls
hola
adnan@adnan:~$
```

Podemos comprobar que coincide con la del a máquina controladora, con el comando **cat .ssh/id\_ed25519.pub** en la controladora y que ambos coinciden.

```
nodo1@nodo1:~$ cat .ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIN72vxIAWmADbHNA6FbIQafNEHVsxhhHYrvKmaLUccve
e.aamranib@ies-sabadell.es
nodo1@nodo1:~$
```

Entramos al controlador y en la carpeta **/etc/hosts** añadimos la Ip y nombre del nodo 1 “n1” para poder conectarnos mediante ssh sin poner la ip solo con el nombre

```
GNU nano 6.2
127.0.0.1 localhost
127.0.1.1 adnan
10.1.23.35 n1
# The following lines are desirable for IPv6 capable hosts
```



## Comprobamos que podemos conectarnos mediante ssh con el host

```
adnan@adnan-VirtualBox:~$ ssh n1
The authenticity of host 'n1 (10.1.23.35)' can't be established.
ED25519 key fingerprint is SHA256:tyKiv4rTHAYmKryz5m3xEIFeGZ7485RL+3RvC4fw5nc.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'n1' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-97-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of mié 06 mar 2024 16:07:17 UTC

System load:  0.0          Processes:      156
Usage of /:   48.1% of 9.75GB   Users logged in: 1
Memory usage: 27%          IPv4 address for enp0s3: 10.1.23.35
Swap usage:   0%

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 59 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Wed Mar  6 16:01:45 2024 from 10.1.23.34
adnan@adnan:~$
```

## Configuración de segundo nodo

Instalaremos el openssh-server en el nodo 2

```
node02@node02:~$ sudo apt install openssh-server
[sudo] password for node02:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server ya está en su versión más reciente (1:8.9p1-3ubuntu0.6).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 61 no actualizados.
node02@node02:~$
```

Mediante ssh nos conectamos al nodo 2 con la ip y podemos comprobar que funciona correctamente

```

connection to 192.168.1.184:
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ssh nodo2@192.168.1.184
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of lun 25 mar 2024 12:19:11 UTC

System load:  0.0673828125      Processes:           180
Usage of /:   48.2% of 9.75GB   Users logged in:    1
Memory usage: 32%              IPv4 address for enp0s3: 192.168.1.184
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

```

Ahora, también desde la máquina controladora copiaremos la clave pública mediante SSH con el comando.

```
adnanamrani@adnanamrani-VirtualBox:~$ ssh-copy-id -i /home/adnanamrani/.ssh/id_ed25519.pub nodo2@192.168.1.184
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/adnanamrani/.ssh/id_ed25519.pub"
The authenticity of host '192.168.1.184 (192.168.1.184)' can't be established.
ED25519 key fingerprint is SHA256:/ba2FL50xT37QehnXj16aotley+RhPL2IgTjYOR9mOU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
nodo2@192.168.1.184's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'nodo2@192.168.1.184'"
and check to make sure that only the key(s) you wanted were added.
```

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ssh -l nodo2 192.168.1.184
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of lun 25 mar 2024 12:25:55 UTC

System load:  0.0               Processes:    172
Usage of /:   48.2% of 9.75GB    Users logged in: 1
Memory usage: 33%              IPv4 address for enp0s3: 192.168.1.184
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 63 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Mon Mar 25 12:19:36 2024 from 192.168.1.182
nodo2@nodo2:~$
```

Podemos comprobar que coincide con la del a máquina controladora, con el comando **cat .ssh/id\_ed25519.pub** en la controladora y que ambos coinciden.

```
nodo2@nodo2:~$ cat .ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIN72vxIAWmADbHNA6FbIQafNEHVsXhhHYrvKmaLUccve e
aamranib@ies-sabadell.es
nodo2@nodo2:~$
```

Abriremos el archivo **/etc/hosts** para añadir la IP del nodo y añadirle un nombre n1

```
GNU nano 6.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 adnanamrani-VirtualBox
192.168.1.183 n1
192.168.1.184 n2
# The following lines are desirable for IPv6 capable hosts
```

Podemos ver que nos hemos conectado sin utilizar la IP solamente con el nombre de host

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ssh nodo2@n2
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of lun 25 mar 2024 12:29:25 UTC

System load:  0.0               Processes:            172
Usage of /:   48.2% of 9.75GB   Users logged in:     1
Memory usage: 33%              IPv4 address for enp0s3: 192.168.1.184
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 63 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Mon Mar 25 12:25:55 2024 from 192.168.1.182
nodo2@nodo2:~$
```

## Ansible

Generamos en el controlador, en la carpeta de nuestro usuario, una carpeta de nombre **01-P1** y dentro un archivo **ansible.cfg**, **inventory** y **inventory-dns**

```
adnan@adnan:~/01-P1$ ls
ansible.cfg  inventory  inventory-dns
```

Dentro de esa nueva carpeta, crearemos los archivos **ansible.cfg**, donde especificaremos la configuración de Ansible.

```
GNU nano 6.2                                ansible.cfg
[defaults]
inventory = ./inventory-dns
```

Ahora, generamos el archivo **inventory-dns**, que tendrá los nombres de los nodos, contendrá lo siguiente.

```
GNU nano 6.2                                inventory-dns
n1
n2
n3
```

Dentro del archivo **inventory**, que tendrá las IPs de los nodos, contendrá lo siguiente.

```
GNU nano 6.2 inventory
10.1.23.35
```

Comprobaremos que la configuración de los archivos de inventario están bien escritos con el comando **ansible-inventory -i inventory --list**

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ansible-inventory -i inventory --list
{
  "_meta": {
    "hostvars": {}
  },
  "all": {
    "children": [
      "ungrouped",
      "nodos"
    ]
  },
  "nodos": {
    "hosts": [
      "192.168.1.183",
      "192.168.1.184"
    ]
  }
}
```

Como podemos comprobar podemos conectarnos con el nombre del nodo.

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ssh n2
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-101-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of lun 25 mar 2024 12:44:43 UTC

System load:  0.0          Processes:           171
Usage of /:   48.2% of 9.75GB Users logged in:       1
Memory usage: 33%          IPv4 address for enp0s3: 192.168.1.184
Swap usage:   0%
```

Comprobamos que hacen ping las máquinas desde el comando ansible **all -i inventory-dns -m ping**

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ansible all -i inventory-dns -m ping
n1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
n2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
adnanamrani@adnanamrani-VirtualBox:~/01-P1$
```

- **SUCCESS:** Significa que la acción que Ansible intentó llevar a cabo en el servidor fue realizada con éxito. En otras palabras, Ansible logró hacer lo que se le pidió hacer.
- **ansible\_facts:** Esta es información extra que Ansible proporciona sobre el servidor con el que está interactuando. Por ejemplo, puede decirnos dónde se encuentra el intérprete de Python en ese servidor.
- **changed:** false: Indica que no se hicieron cambios en el servidor como resultado de la tarea ejecutada por Ansible. Es como decir que Ansible verificó el estado pero no fue necesario hacer ningún cambio.
- **ping: pong:** Es una respuesta simple que indica que el servidor está disponible y respondiendo a las solicitudes de Ansible. Es similar a un "sí" cuando preguntas si alguien está ahí.

## Inventarios y privilegios

En el archivo `inventory-dns` añadiremos lo siguiente, para trabajar con nombres de grupos en vez de con nodos individuales.

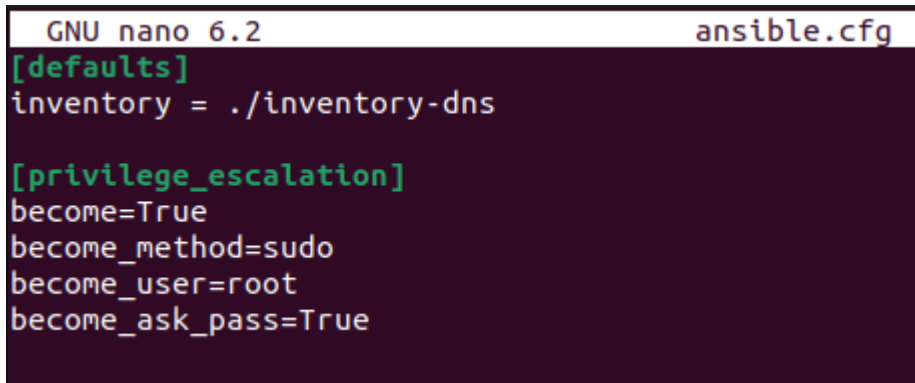
```
GNU nano 6.2                                inventory-dns
[nodos]
n1
n2

[nografico]
n2
```

Haré un ping al grupo **nografico** con el comando **ansible nografico -m ping**, donde veremos la respuesta del nodo 2.

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ansible nografico -m ping
n2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
adnanamrani@adnanamrani-VirtualBox:~/01-P1$
```

Cuando usamos Ansible para ejecutar comandos en otros nodos como administrador, a veces necesitamos configurar ciertas opciones para asegurarnos de que todo funcione correctamente. Una de esas opciones es llamada **privilege\_escalation**



```
GNU nano 6.2                               ansible.cfg
[defaults]
inventory = ./inventory-dns

[privilege_escalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=True
```

- **become=True:** Activa la escalada de privilegios, permitiendo que las tareas se ejecuten con privilegios de superusuario en los hosts remotos.
- **become\_method=sudo:** Especifica el método que Ansible usará para realizar la escalada de privilegios. En este caso, se utiliza "sudo" para cambiar al usuario root utilizando el comando sudo.
- **become\_user=root:** Indica el usuario al que Ansible cambiará cuando realice la escalada de privilegios. Aquí se establece como "root", lo que significa que Ansible ejecutará las tareas con privilegios de superusuario.
- **become\_ask\_pass=True:** Indica que Ansible solicitará la contraseña del usuario que realiza la conexión SSH para realizar la escalada de privilegios. Esto es útil si se requiere una contraseña para ejecutar el comando sudo.

Finalmente, si ejecutamos un ping a todos los nodos con el comando **ansible all -i inventory-dns -m ping**, nos pedirá contraseña.

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ansible all -i inventory-dns -m ping
BECOME password:
n1 | SUCCESS => {
  "ansible_facts": {
    "ansible_python_interpreter": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
n2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
adnanamrani@adnanamrani-VirtualBox:~/01-P1$
```

## Ansible playbooks

Verificamos que el nodo 1 no reconoce el comando **sl**, el objetivo es instalar el **easter egg** llamado **sl** en el nodo 1 utilizando un playbook de Ansible. Después de ejecutar el playbook de Ansible, esperamos que el nodo 1 reconozca el comando "sl", lo que indicaría que se ha instalado con éxito el **easter egg** llamado **sl**. Finalmente generamos un archivo con lo siguiente.

```
GNU nano 6.2                                install_sl.yaml
name: Instalacion de "sl"
hosts: n1
tasks:
  - name: Instalado "sl"
    package:
      name: sl
      state: latest
      update_cache: true
```

Ejecutamos el comando **ansible-playbook install\_sl.yaml** y vemos que se procede a la instalación de **sl**

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ansible-playbook install_sl.yaml
BECOME password:

PLAY [Instalacion de "sl"] *****

TASK [Gathering Facts] *****
ok: [n1]

TASK [Instalado "sl"] *****
changed: [n1]

PLAY RECAP *****
n1 : ok=2  changed=1  unreachable=0  failed=0  skip
ped=0  rescued=0  ignored=0

adnanamrani@adnanamrani-VirtualBox:~/01-P1$
```

Para crear una playbook de desinstalación de **sl**, copiaremos la playbook que teníamos y le cambiaremos el nombre a **uninstall\_sl.yaml**

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ cp install_sl.yaml uninstall_sl.yaml
adnanamrani@adnanamrani-VirtualBox:~/01-P1$
```

Modificaremos el archivo de esta manera.

```
GNU nano 6.2                                uninstall_sl.yaml
- name: Desinstalacion de "sl"
  hosts: n1
  tasks:
    - name: Instalado "sl"
      package:
        name: sl
        state: latest
```

Ejecutamos este playbook con el comando **ansible-playbook uninstall\_sl.yaml**

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ansible-playbook uninstall_sl.yaml
BECOME password:

PLAY [Desinstalacion de "sl"] *****

TASK [Gathering Facts] *****
ok: [n1]

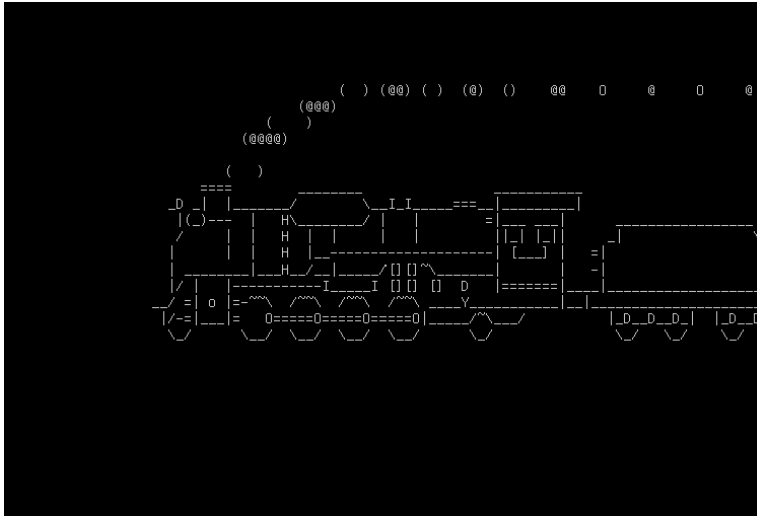
TASK [Instalado "sl"] *****
changed: [n1]

PLAY RECAP *****
n1 : ok=2  changed=1  unreachable=0  failed=0  skip
ped=0  rescued=0  ignored=0

adnanamrani@adnanamrani-VirtualBox:~/01-P1$
```



Ahora, accedemos al nodo 1 y ejecutamos el comando **sl** para comprobar que ahora sí que está instalado.



Para visualizar los cambios sin llegar a instalar **ansible-playbook --check install\_sl.yaml**

```
adnanamrani@adnanamrani-VirtualBox:~/01-P1$ ansible-playbook --check install_sl.yaml
PLAY [Instalacion de "sl"] *****
TASK [Gathering Facts] *****
ok: [n1]

TASK [Instalado "sl"] *****
ok: [n1]

PLAY RECAP *****
n1 : ok=2  changed=0  unreachable=0  failed=0  skip
ped=0  rescued=0  ignored=0

adnanamrani@adnanamrani-VirtualBox:~/01-P1$
```

## Playbook Audacity

Haremos un nuevo playbook de nombre **install\_audacity.yaml** con el siguiente contenido

```
GNU nano 6.2 install_audacity.yaml *
name: Instalacion de "Audacity"
hosts: n1
tasks:
  - name: Instalado "Audacity"
    package:
      name: audacity
      state: latest
      update_cache: true
```

Lo instalamos con el comando **ansible-playbook install\_audacity.yaml**

```
adnanamrani@adnanamrani-VirtualBox: ~/01-P1$ ansible-playbook install_audacity.yaml
BECOME password:

PLAY [Instalacion de "Audacity"] *****

TASK [Gathering Facts] *****
ok: [n1]

TASK [Instalado "Audacity"] *****
ok: [n1]

PLAY RECAP *****
n1 : ok=2 changed=0 unreachable=0 failed=0 skip
ed=0 rescued=0 ignored=0
```

Descargaremos los archivos de WinSCP

/home/adnanamrani/01-P1/				
Nombre	Tamaño	Modificado	Permisos	Propietario
..		24/03/2024 13:45:00	rw-r-x---	adnanam...
uninstall_sl.yaml	1 KB	24/03/2024 14:52:23	rw-rw-r--	adnanam...
inventory-dns	1 KB	24/03/2024 14:11:20	rw-rw-r--	adnanam...
inventory	1 KB	24/03/2024 13:54:06	rw-rw-r--	adnanam...
install_sl.yaml	1 KB	24/03/2024 14:48:19	rw-rw-r--	adnanam...
install_audacity.yaml	1 KB	24/03/2024 15:02:18	rw-rw-r--	adnanam...
ansible.cfg	1 KB	24/03/2024 14:43:46	rw-rw-r--	adnanam...

## Bibliografia

[Rsa vs. dsa vs. ecdsa: an introduction to public key cryptography \(nanotrust.io\)](https://nanotrust.io/)

[Types of Encryption Algorithms + Pros and Cons for Each | Keyfactor](#)

[RSA, ECC, ECDSA: which algorithm is better to choose when ordering a digital certificate in LeaderSSL? | LeaderSSL](#)