

Universidad ORT Uruguay

Facultad de Ingeniería

Obligatorio

Taller de Servidores Linux

Profesor: Miguel Vartabedián

Autores: Nicolás Cameto - 324408

Francisco Polido - 343332

Agosto - 2025

Índice

Tarea 1: Instalación de Servidores.....	3
Configuración de Hardware:.....	3
Particionado de disco:.....	4
Configuración de Interfaces de Red:.....	5
Configuración de Claves SSH:.....	8
Configuración del Archivo Visudo:.....	10
Tarea 2: Configurar un archivo de inventario de Ansible.....	11
Instalación de Ansible:.....	11
Configuración del inventario:.....	11
Configuración de host_vars:.....	12
Configuración de group_vars:.....	12
Configuración del archivo ansible.cfg.....	12
Tarea 3: Ejecutar comandos ad-hoc.....	15
Listar todos los usuarios en servidor Ubuntu.....	15
Mostrar el uso de memoria en todos los servidores.....	15
Que el servicio chrony esté instalado y funcionando en servidor Centos:.....	16
Tarea 4: Crear y ejecutar playbook de Ansible.....	17
nfs_setup.yml.....	17
hardening.yml.....	19
Anexo.....	24
Declaración de autoría:.....	24
Bibliografía:.....	24

Tarea 1: Instalación de Servidores

De acuerdo a lo establecido por la documentación del obligatorio se instalarán dos servidores, uno basado en CentOS Stream 9, y otro basado en Ubuntu Live Server 24.04.

A su vez se instalará un tercer equipo basado en Ubuntu Live Server 24.04. que actuará como controller de Ansible.

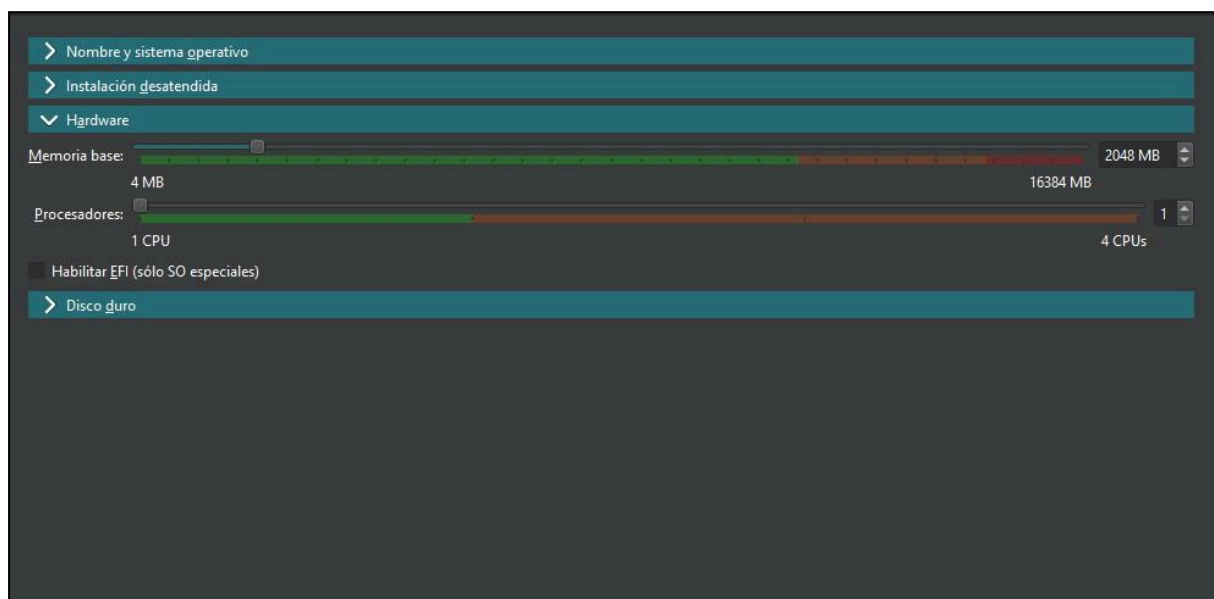
Nombre y función de los Equipos:

- Ubuntu01 - Controller de Ansible
- Ubuntu02 - Server
- CentOS01 - Web Server

Configuración de Hardware:

Los tres servidores tendrán las siguientes características de hardware:

- 1 CPU
- 2 Gb de RAM
- Un disco duro con un espacio de almacenamiento de entre 20 y 25Gb.



Particionado de disco:

Se configurarán las siguientes particiones de disco:

Ubuntu01 y Ubuntu02:

```
Storage configuration

FILE SYSTEM SUMMARY

MOUNT POINT      SIZE      TYPE      DEVICE TYPE
[ /               10.000G   new ext4   new partition of local disk ► ]
[ /boot           2.000G   new ext4   new partition of local disk ► ]
[ /var            5.000G   new ext4   new partition of local disk ► ]
[ SWAP            4.000G   new swap   new partition of local disk ► ]

AVAILABLE DEVICES

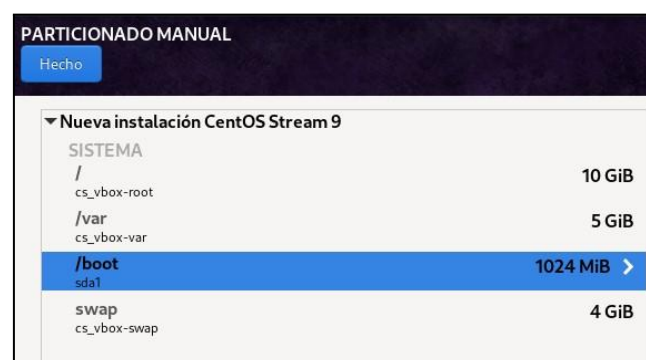
DEVICE                                     TYPE      SIZE
[ VBOX_HARDDISK_VBb88f6b26-5631d037      local disk 25.000G ► ]
  free space                               3.997G ► ]

[ Create software RAID (md) ► ]
[ Create volume group (LVM) ► ]

USED DEVICES

DEVICE                                     TYPE      SIZE
[ VBOX_HARDDISK_VBb88f6b26-5631d037      local disk 25.000G ► ]
  partition 1 new, BIOS grub spacer          1.000M ► ]
  partition 2 new, to be formatted as ext4, mounted at /boot 2.000G ► ]
  partition 3 new, to be formatted as ext4, mounted at /      10.000G ► ]
  partition 4 new, to be formatted as ext4, mounted at /var    5.000G ► ]
  partition 5 new, to be formatted as swap                      4.000G ► ]
```

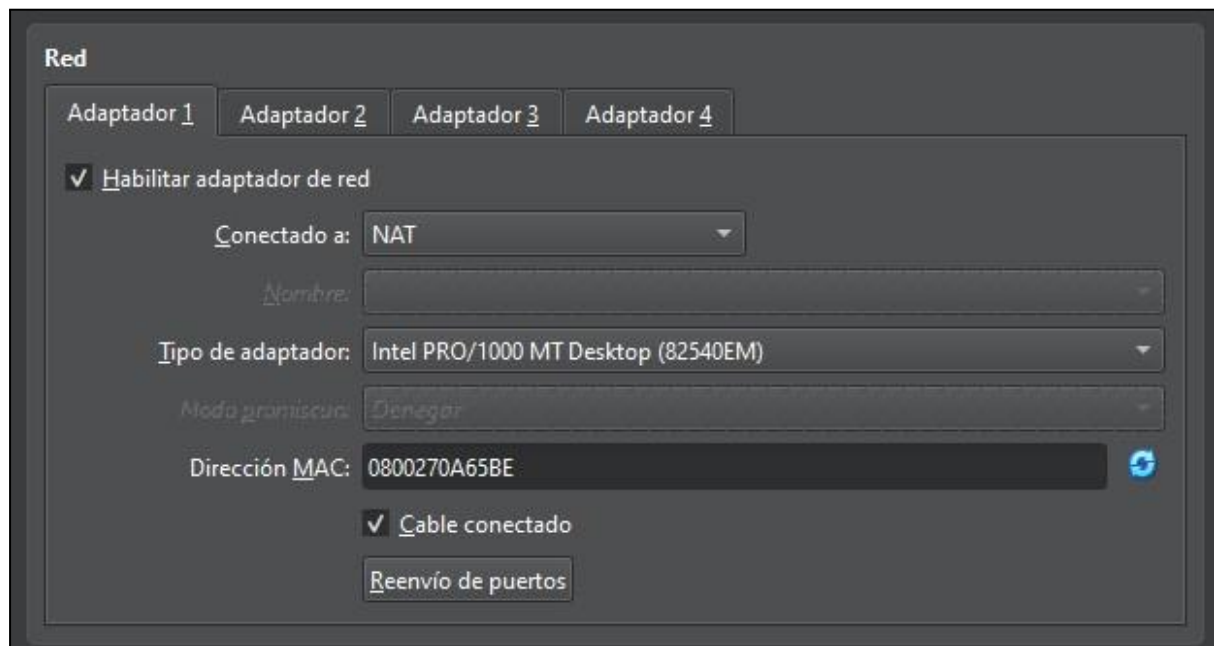
CentOS01:



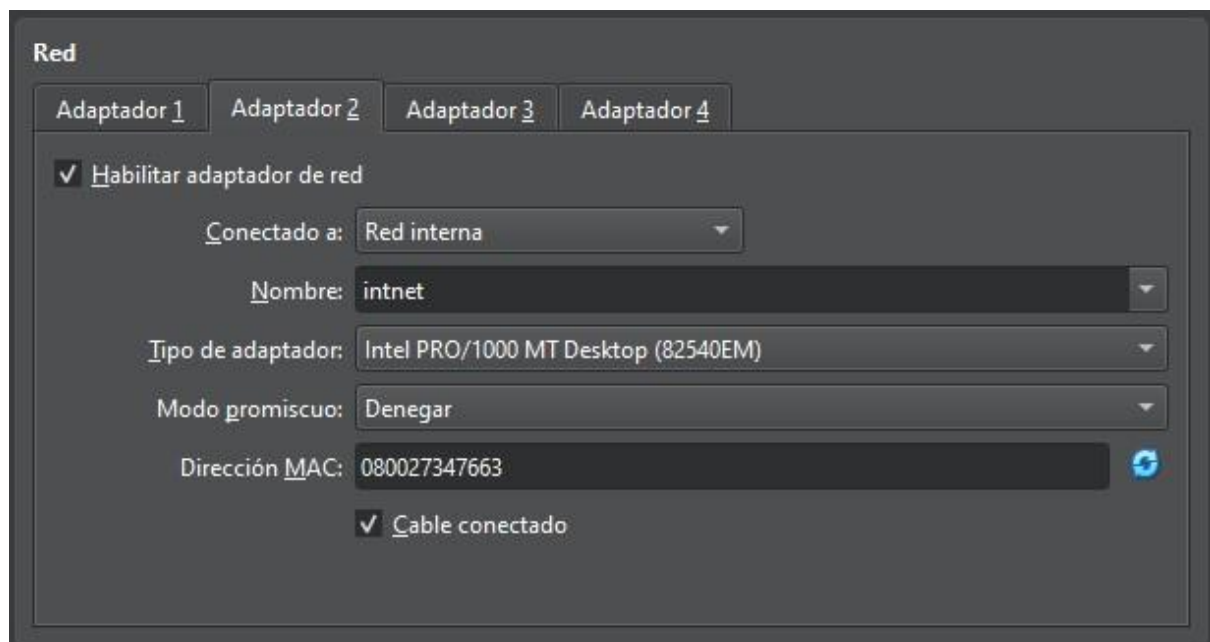
Configuración de Interfaces de Red:

A nivel de hardware, todos los servidores serán configurados de la siguiente manera:

Tendrán una interfaz conectada a NAT:



Y otra interfaz conectada a red interna:



A nivel individual, los servidores tendrán las siguientes configuraciones de red:

Ubuntu01:

```
Ubuntu01 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
sysadmin@ubuntu01 ~$ sudo cat /etc/netplan/50-cloud-init.yaml
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      dhcp6: false
      addresses: [192.168.1.1/24, ]
sysadmin@ubuntu01 ~$ _
```

```
sysadmin@ubuntu01 ~$ sudo netplan apply
sysadmin@ubuntu01 ~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ea:de:51 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86397sec preferred_lft 86397sec
    inet6 fd00::a00:27ff:feea:de51/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86398sec preferred_lft 14398sec
    inet6 fe80::a00:27ff:feea:de51/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d0:7d:e7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed0:7de7/64 scope link
        valid_lft forever preferred_lft forever
sysadmin@ubuntu01 ~$ _
```

Ubuntu02:

```
sysadmin@ubuntu02 ~$ sudo cat /etc/netplan/50-cloud-init.yaml
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      dhcp6: false
      addresses: [192.168.1.11/24, ]
```

```
sysadmin@ubuntu02 ~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:5f:94:9c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86339sec preferred_lft 86339sec
    inet6 fd00::a00:27ff:fe5f:949c/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86342sec preferred_lft 14342sec
    inet6 fe80::a00:27ff:fe5f:949c/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:29:d6:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe29:d64d/64 scope link
        valid_lft forever preferred_lft forever
sysadmin@ubuntu02 ~$
```

CentOS01:

Editar la conexión

Nombre de perfil	enp0s8		
Dispositivo	enp0s8 (08:00:27:34:76:63)		
= ETHERNET			<Mostrar>
= 802.1X SECURITY			<Mostrar>
= CONFIGURACION IPv4	<Manual>		<Ocultar>
Direcciones	192.168.1.10/24	<Retirar>	
	<Añadir■>		
Puerta de enlace	192.168.1.1		
Servidores DNS	<Añadir■>		
Búsqueda de dominios	<Añadir■>		
Enrutando (No hay rutas personalizadas) <Editar■>			

```
[sysadmin@vbox ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0a:65:be brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 86106sec preferred_lft 86106sec
    inet6 fd00::a00:27ff:fe0a:65be/64 scope global dynamic noprefixroute
        valid_lft 86107sec preferred_lft 14107sec
    inet6 fe80::a00:27ff:fe0a:65be/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:34:76:63 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
[sysadmin@vbox ~]$
```


Configuración de Claves SSH:

Para poder realizar conexiones ssh seguras, en cada equipo, generamos un par de claves ssh mediante el uso del siguiente comando: `ssh-keygen`.

Una vez generados los pares de claves en cada servidor, procedemos a copiar las claves públicas de cada uno de ellos en el servidor que actuará como ansible controller:

Copiamos llave de **Centos01**:

```
sysadmin@ubuntu01 ~$ ssh-copy-id 192.168.1.10
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sysadmin/.ssh/
The authenticity of host '192.168.1.10 (192.168.1.10)' can't be established.
ED25519 key fingerprint is SHA256:Iih27ivTcF0NemCbcJE8DAueqypw3VAADRtUWD12EMU.
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
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted r
sysadmin@192.168.1.10's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '192.168.1.10'"
and check to make sure that only the key(s) you wanted were added.

sysadmin@ubuntu01 ~$ _
```

Copiamos llave de **Ubuntu02**:

```
sysadmin@ubuntu01 ~$ ssh-copy-id 192.168.1.11
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sysadmin/
The authenticity of host '192.168.1.11 (192.168.1.11)' can't be established.
ED25519 key fingerprint is SHA256:YGD0IIOPPjU8+rvhwiA27Y7V2yK8TRi3vci/gP64eaI.
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
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
sysadmin@192.168.1.11's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '192.168.1.11'"
and check to make sure that only the key(s) you wanted were added.

sysadmin@ubuntu01 ~$
```


Verificamos conexión a través de ssh con ambos servidores:

Centos01:

```
sysadmin@ubuntu01 ~> ssh 192.168.1.10
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Tue Aug 5 16:38:52 2025
[sysadmin@vbox ~]$
```

Ubuntu02:

```
sysadmin@ubuntu01 ~> ssh 192.168.1.11
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-71-generic x86_64)

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

System information as of mar 05 ago 2025 21:00:34 UTC

System load:          0.04
Usage of /:            21.6% of 9.75GB
Memory usage:         10%
Swap usage:           0%
Processes:            101
Users logged in:      1
IPv4 address for enp0s3: 10.0.2.15
IPv6 address for enp0s3: fd00::a00:27ff:fe5f:949c

 * 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 0 actualizaciones de forma inmediata.

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

sysadmin@ubuntu02:~$
```

Configuración del Archivo Visudo:

Con el que al ejecutar comandos, no se le solicite la contraseña al usuario sysadmin, procedemos a modificar el archivo visudo en todos los servidores.

Ubuntu 01 y Ubuntu02:

En ambos servidores debe modificarse el archivo de la siguiente manera:

```
# Allow members of group sudo to execute any command
%sudo  ALL=(ALL) NOPASSWD: ALL
```

Verificamos que no le solicite contraseña a “sysadmin” al ejecutar comandos.

```
sysadmin@ubuntu01 ~$ sudo -i
root@ubuntu01:~#
```

Centos01:

En el servidor de centos01, se comenta la línea donde dice que el grupo wheel pueda correr todos los comandos, y se descomenta la línea en la que dice que los puede correr pero sin contraseña:

```
## Allows people in group wheel to run all commands
##%wheel ALL=(ALL)        ALL

## Same thing without a password
%wheel  ALL=(ALL)        NOPASSWD: ALL
```

Verificamos que no le solicite contraseña a “sysadmin” al ejecutar comandos.

```
sysadmin@vbox ~$ sudo -i
[root@vbox ~]#
```

Tarea 2: Configurar un archivo de inventario de Ansible

Instalación de Ansible:

Primero que nada, se procede a instalar ansible en el equipo que será controller, para esto se ejecutaron los siguientes comandos:

1. `sudo apt install python3-pip`
2. `pip install pipx`
3. `pipx ensurepath`
4. `pipx install ansible-core`
5. `pipx inject ansible-core argcomplete`
6. `pipx inject ansible-core ansible-lint`
7. `activate-global-python-argcomplete --user`

Posteriormente se crea el directorio ansible, utilizando mkdir, donde se copiara el archivo de ansible.cfg

```
sysadmin@ubuntu01 ~$ find . -name 'ansible.cfg'
./local/share/pipx/venvs/ansible-core/lib/python3.12/site-packages/ansible_test/_data/ansible.cfg
./local/share/pipx/venvs/ansible-core/lib/python3.12/site-packages/ansible/galaxy/data/apb/tests/ansible.cfg
./local/share/pipx/venvs/ansible-core/lib/python3.12/site-packages/ansible/galaxy/data/container/tests/ansible.cfg
sysadmin@ubuntu01 ~$ cd ansible/
sysadmin@ubuntu01 ~/ansible$ cp ~/.local/share/pipx/venvs/ansible-core/lib/python3.12/site-packages/ansible_test/_data/ansible.cfg .
sysadmin@ubuntu01 ~/ansible$
```

Configuración del inventario:

Para la configuración de un archivo de inventario, primero se creó una directorio inventory, donde estará el archivo de inventory.ini.

```
| inventory
|   inventory.ini
```

En este último serán definidos los siguientes hosts y grupos:

```
ubuntu01
ubuntu02
centos01
```

```
[ubuntu]
ubuntu01
```

ubuntu02

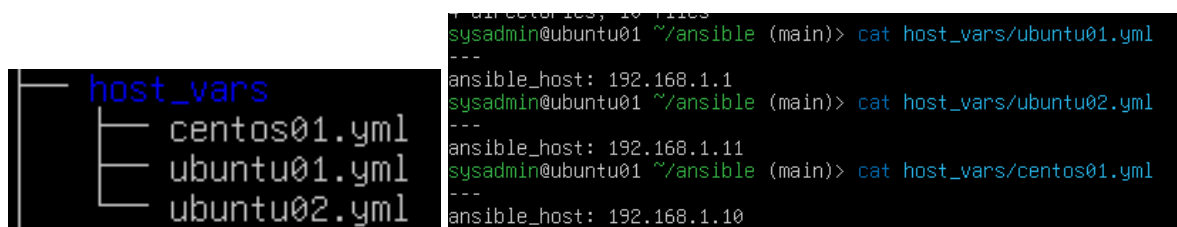
[centos]
centos01

[webservers]
centos01

[linux:children]
ubuntu
centos

Configuración de host_vars:

Para el correcto funcionamiento, se creó el directorio de host_vars donde estarán distintos yml para cada una de las máquinas con el mismo nombre, donde estarán definidas sus direcciones IP a través de la variable de ansible_host.

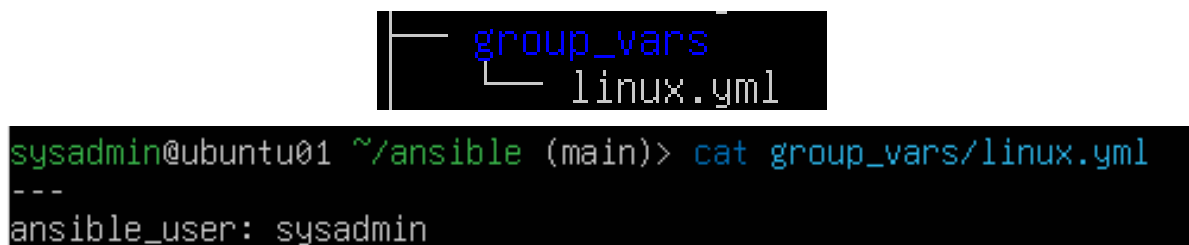


```
└─ host_vars
   └─ centos01.yml
   └─ ubuntu01.yml
   └─ ubuntu02.yml
```

```
sysadmin@ubuntu01 ~/ansible (main)> cat host_vars/ubuntu01.yml
---
ansible_host: 192.168.1.1
sysadmin@ubuntu01 ~/ansible (main)> cat host_vars/ubuntu02.yml
---
ansible_host: 192.168.1.11
sysadmin@ubuntu01 ~/ansible (main)> cat host_vars/centos01.yml
---
ansible_host: 192.168.1.10
```

Configuración de group_vars:

Además de esto, también se creó un directorio llamado group_vars donde estará un archivo yml, referenciando al grupo linux, para definir el ansible_user que usuario usará para la conexión SSH.



```
└─ group_vars
   └─ linux.yml
```

```
sysadmin@ubuntu01 ~/ansible (main)> cat group_vars/linux.yml
---
ansible_user: sysadmin
```

Configuración del archivo ansible.cfg

Por último, se editará el archivo de ansible.cfg para que utilice la ruta correcta del inventario, además de ocultar avisos de funciones obsoletas y detectar

automáticamente la ruta del intérprete de Python en el host remoto, sin mostrar advertencias.

```
[defaults]
inventory=./inventory/inventory.ini
deprecation_warnings=False
interpreter_python=auto_silent
```

Prueba de conexión:

```
$ ansible-inventory -i inventory.ini --list
```

```
sysadmin@ubuntu01 ~/ansible (main)> ansible-inventory -i inventory/inventory.ini --list
{
  "_meta": {
    "hostvars": {
      "centos01": {
        "ansible_host": "192.168.1.10",
        "ansible_user": "sysadmin"
      },
      "ubuntu01": {
        "ansible_host": "192.168.1.1",
        "ansible_user": "sysadmin"
      },
      "ubuntu02": {
        "ansible_host": "192.168.1.11",
        "ansible_user": "sysadmin"
      }
    }
  },
  "profile": "inventory_legacy"
},
{
  "all": {
    "children": [
      "ungrouped",
      "webservers",
      "linux"
    ]
  },
  "centos": {
    "hosts": [
      "centos01"
    ]
  },
  "linux": {
    "children": [
      "ubuntu",
      "centos"
    ]
  },
  "ubuntu": {
    "hosts": [
      "ubuntu01",
      "ubuntu02"
    ]
  },
  "webservers": {
    "hosts": [
      "centos01"
    ]
  }
}
```

```
$ ansible all -i inventory.ini -m ping
```

```
sysadmin@ubuntu01 ~/obligatorio (main)> ansible all -i inventory/inventory.ini -m ping
ubuntu01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
centos01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.9"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu02 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
```


Tarea 3: Ejecutar comandos ad-hoc

En esta tarea, se ejecutaron comandos ad-hoc de Ansible para realizar instrucciones rápidas sobre las máquinas del inventario.

Listar todos los usuarios en servidor Ubuntu

Para esto, se utilizó el módulo shell para ejecutar un `cat /etc/passwd` en los servidores del grupo Ubuntu ya que allí es donde está contenida la información básica de todas las cuentas de usuario. Además, se usó un `cut -d: -f1` para mostrar solo los nombres de usuario registrados en el sistema:

```
sysadmin@ubuntu01 ~/ansible (main)> ansible ubuntu -m shell -a "cut -d: -f1 /etc/passwd"
ubuntu02 | CHANGED | rc=0 >>
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
_apt
nobody
systemd-network
systemd-timesync
dhcpd
messagebus
systemd-resolve
pollinate
polkitd
syslog
uidd
tcpdump
tss
landscape
fwupd-refresh
usbmux
sshd
sysadmin
ubuntu01 | CHANGED | rc=0 >>
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
proxy
www-data
backup
list
irc
_apt
nobody
systemd-network
systemd-timesync
dhcpd
messagebus
systemd-resolve
pollinate
polkitd
syslog
uidd
tcpdump
tss
landscape
fwupd-refresh
usbmux
sshd
sysadmin
```

Mostrar el uso de memoria en todos los servidores

De la misma forma que en la parte anterior, se utilizó el módulo shell para mostrar el uso de memoria en todos los equipos:

```
sysadmin@ubuntu01 ~/ansible (main) [2]> ansible all -m shell -a "free -h"
ubuntu02 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    1.9Gi       318Mi     1.5Gi       1.1Mi       237Mi       1.6Gi
Swap:    3.2Gi         0B       3.2Gi
centos01 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    1.7Gi       375Mi     1.0Gi       5.0Mi       408Mi       1.3Gi
Swap:    4.0Gi         0B       4.0Gi
ubuntu01 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    1.9Gi       421Mi     1.4Gi       1.1Mi       292Mi       1.5Gi
Swap:    3.2Gi         0B       3.2Gi
sysadmin@ubuntu01 ~/ansible (main)>
```

Que el servicio chrony esté instalado y funcionando en servidor Centos:

Utilizando el módulo de shell, para comprobar el estado del servicio chronyd en CentOS, se ejecutó un `systemctl status chronyd`,

```
sysadmin@ubuntu01 ~/ansible (main)> ansible centos -m shell -a "systemctl status chronyd"
centos01 | CHANGED | rc=0 >>
• chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-08-05 17:30:20 -03; 1h 49min ago
     Docs: man:chronyd(8)
           man:chrony.conf(5)
   Process: 729 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 738 (chronyd)
     Tasks: 1 (limit: 10568)
    Memory: 4.0M
       CPU: 332ms
   CGroup: /system.slice/chronyd.service
           └─738 /usr/sbin/chronyd -F 2

Aug 05 17:30:20 centos01 chronyd[738]: chronyd version 4.6.1 starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER +SIGND +ASYNCDNS +NTS +SECHASH +IPV6 +DEB
UG)
Aug 05 17:30:20 centos01 chronyd[738]: Loaded 0 symmetric keys
Aug 05 17:30:20 centos01 chronyd[738]: Using right/UTC timezone to obtain leap second data
Aug 05 17:30:20 centos01 chronyd[738]: Frequency 19.995 +/- 0.395 ppm read from /var/lib/chrony/drift
Aug 05 17:30:20 centos01 chronyd[738]: Loaded seccomp filter (level 2)
Aug 05 17:30:20 centos01 systemd[1]: Started NTP client/server.
Aug 05 17:30:38 centos01 chronyd[738]: Selected source 200.40.115.74 (2.centos.pool.ntp.org)
Aug 05 17:30:38 centos01 chronyd[738]: System clock wrong by 1.208180 seconds
Aug 05 17:30:39 centos01 chronyd[738]: System clock was stepped by 1.208180 seconds
Aug 05 17:30:39 centos01 chronyd[738]: System clock TAI offset set to 37 seconds
sysadmin@ubuntu01 ~/ansible (main)>
```

Tarea 4: Crear y ejecutar playbook de Ansible

En esta parte se desarrollaron y ejecutaron dos playbooks de Ansible, uno orientado a la configuración de un servidor NFS en CentOS y otro para la implementación de medidas de hardening en servidores Ubuntu.

nfs_setup.yml

Este playbook tiene como objetivo el configurar automáticamente usando Ansible un servicio de Network File System (NFS) para CentOS y montar el recurso compartido. Para el correcto funcionamiento, se tiene que instalar la colección de `ansible.posix`

```
---
- hosts: centos
  user: sysadmin
  become: yes
  gather_facts: false

  tasks:
    - name: Instalar NFS utils
      yum:
        name: nfs-utils
        state: present

    - name: Asegurar que el servicio NFS esté iniciado y habilitado
      ansible.builtin.service:
        name: nfs-server
        state: started
        enabled: yes

    - name: Permitir puerto 2049/tcp en el firewall
      ansible.posix.firewalld:
        port: 2049/tcp
        permanent: yes
        state: enabled
        immediate: yes

    - name: Crear el directorio /var/nfs_shared con owner y permisos
      ansible.builtin.file:
        path: /var/nfs_shared
        state: directory
```

```
owner: nobody
group: nobody
mode: '0777'
```

- name: Configurar exportación en /etc/exports
ansible.builtin.lineinfile:
 path: /etc/exports
 line: /var/nfs_shared *(rw, sync, no_root_squash)
 create: yes
notify: Recargar nfs

handlers:

- name: Recargar nfs
ansible.builtin.service:
 name: nfs-server
 state: restarted

...

Playbook aplicado:

```
sysadmin@ubuntu01 ~/obligatorio (main)> ansible-playbook tasks/nfs_setup.yml
PLAY [centos] *************************************************************************************************************************************
TASK [Instalar NFS utils] *************************************************************************************************************************************
changed: [centos01]
TASK [Asegurar que el servicio NFS esté iniciado y habilitado] *********************************************************************
changed: [centos01]
TASK [Permitir puerto 2049/tcp en el firewall] *********************************************************************
changed: [centos01]
TASK [Crear el directorio /var/nfs_shared con owner y permisos] *********************************************************************
changed: [centos01]
TASK [Configurar exportación en /etc/exports] *********************************************************************
changed: [centos01]
RUNNING HANDLER [Recargar nfs] *********************************************************************
changed: [centos01]
PLAY RECAP *********************************************************************
centos01                : ok=6   changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
sysadmin@ubuntu01 ~/obligatorio (main)>
```

Resultados del playbook:

```
Aug 07 17:12:45 centos01 systemd[1]: Starting NFS server and services...
Aug 07 17:12:45 centos01 systemd[1]: Finished NFS server and services.
sysadmin@centos01 ~$ systemctl status nfs-server
nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: disabled)
   Active: active (exited) since Thu 2025-08-07 17:12:45 -03; 3min 36s ago
     Docs: man:rpc.nfsd(8)
           man:exportfs(8)
   Process: 5800 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
   Process: 5801 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
   Process: 5811 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy; then systemctl reload gssproxy ; fi (code=exited, status=0/SUCCESS)
   Main PID: 5811 (code=exited, status=0/SUCCESS)
     CPU: 76ms

Aug 07 17:12:45 centos01 systemd[1]: Starting NFS server and services...
Aug 07 17:12:45 centos01 systemd[1]: Finished NFS server and services.
sysadmin@centos01 ~$ sudo firewall-cmd --list-ports
2049/tcp
sysadmin@centos01 ~$ ls -ld /var/nfs_shared
drwxrwxrwx. 2 nobody nobody 6 Aug 7 17:12 /var/nfs_shared/
sysadmin@centos01 ~$ cat /etc/exports
/var/nfs_shared *(rw,sync,no_root_squash)
sysadmin@centos01 ~$ exportfs -v
exportfs: could not open /var/lib/nfs/.etab.lock for locking: errno 13 (Permission denied)
sysadmin@centos01 ~$ sudo exportfs -v
/var/nfs_shared
<world>(sync,udelay,hide,no_subtree_check,sec=sys,rw,secure,no_root_squash,no_all_squash)
```

hardening.yml

Este playbook aplica medidas de seguridad en sistemas Ubuntu, tales como la actualización de paquetes, la configuración de un firewall restrictivo, el endurecimiento de la autenticación SSH, y la instalación de Fail2Ban para prevenir ataques de fuerza bruta. Para el correcto funcionamiento, se tiene que instalar la colección de community.general

- hosts: ubuntu
 user: sysadmin
 become: yes
 gather_facts: false
- tasks:
 - name: Actualizar todos los paquetes
 apt:
 - update_cache: yes
 - upgrade: dist
 notify: Reiniciar el sistema
 - name: Permitir SSH (puerto 22)
 community.general.ufw:
 - rule: allow
 - port: 22
 - proto: tcp
 - name: Habilitar UFW y denegar todo por defecto

```
community.general.ufw:
  state: enabled
  policy: deny

- name: Aplico medidas de seguridad a SSH
  ansible.builtin.lineinfile:
    path: /etc/ssh/sshd_config
    regexp: "{{ item.abuscar }}"
    line: "{{ item.reemplazo }}"
  loop:
    - { abuscar: '#PermitRootLogin', reemplazo: 'PermitRootLogin
no' }
      - { abuscar: '#AuthenticationMethods', reemplazo:
'AuthenticationMethods publickey' }
      - { abuscar: '#PubkeyAuthentication', reemplazo:
'PubkeyAuthentication yes' }
  notify: Reinicio SSH

- name: Instalar fail2ban
  apt:
    name: fail2ban
    state: present

- name: Asegurar que fail2ban esté iniciado y habilitado
  ansible.builtin.service:
    name: fail2ban
    state: started
    enabled: true

handlers:

- name: Reiniciar el sistema
  reboot:
    reboot_timeout: 600

- name: Reinicio SSH
  service:
    name: ssh
    state: restarted

...
```


Playbook aplicado:

Cabe destacar que al aplicar el playbook por primera vez, este se queda trancado a la hora de habilitar el firewall en la máquina de ubuntu02, aunque la regla se aplique igual, por lo que cancelamos la tarea y esta se terminó solo en la máquina de ubuntu01.

```
sysadmin@ubuntu01 ~/obligatorio (main)> ansible-playbook tasks/hardening.yml

PLAY [ubuntu] *********************************************************************

TASK [Actualizar todos los paquetes] *********************************************************************
changed: [ubuntu02]
changed: [ubuntu01]

TASK [Permitir SSH (puerto 22)] *********************************************************************
changed: [ubuntu02]
changed: [ubuntu01]

TASK [Habilitar UFW y denegar todo por defecto] *********************************************************************
changed: [ubuntu01]
^C^C[ERROR]: Task failed: [Errno 3] No such process

Task failed.
Origin: /home/sysadmin/obligatorio/tasks/hardening.yml:20:7

18         proto: tcp
19
20     - name: Habilitar UFW y denegar todo por defecto
      ^ column 7

<<< caused by >>>

[Errno 3] No such process

fatal: [ubuntu02]: FAILED! => {"changed": false, "msg": "Task failed: [Errno 3] No such process"}

TASK [Aplico medidas de seguridad a SSH] *********************************************************************
changed: [ubuntu01] => (item={'abuscar': '#PermitRootLogin', 'reemplazo': 'PermitRootLogin no'})
changed: [ubuntu01] => (item={'abuscar': '#AuthenticationMethods', 'reemplazo': 'AuthenticationMethods publickey'})
changed: [ubuntu01] => (item={'abuscar': '#PubkeyAuthentication', 'reemplazo': 'PubkeyAuthentication yes'})

TASK [Instalar fail2ban] *********************************************************************
changed: [ubuntu01]

TASK [Asegurar que fail2ban esté iniciado y habilitado] *********************************************************************
ok: [ubuntu01]

RUNNING HANDLER [Reiniciar el sistema] *********************************************************************
```

Una vez reiniciado el sistema por la actualización de paquetes, al correr de vuelta el playbook, este si se logró completar en ubuntu02, incluso indicando que no cambió nada en la parte del firewall, por lo que este ya había cambiado la primera vez que se ejecutó el playbook.

```
sysadmin@ubuntu01 ~/obligatorio (main)> ansible-playbook tasks/hardening.yml

PLAY [ubuntu] *************************************************************************************************************************************

TASK [Actualizar todos los paquetes] *********************************************************************
ok: [ubuntu01]
ok: [ubuntu02]

TASK [Permitir SSH (puerto 22)] *********************************************************************
ok: [ubuntu02]
ok: [ubuntu01]

TASK [Habilitar UFW y denegar todo por defecto] *********************************************************************
ok: [ubuntu02]
ok: [ubuntu01]

TASK [Aplico medidas de seguridad a SSH] *********************************************************************
changed: [ubuntu02] => (item={'abuscar': '#PermitRootLogin', 'reemplazo': 'PermitRootLogin no'})
ok: [ubuntu01] => (item={'abuscar': '#PermitRootLogin', 'reemplazo': 'PermitRootLogin no'})
changed: [ubuntu02] => (item={'abuscar': '#AuthenticationMethods', 'reemplazo': 'AuthenticationMethods publickey'})
ok: [ubuntu01] => (item={'abuscar': '#AuthenticationMethods', 'reemplazo': 'AuthenticationMethods publickey'})
changed: [ubuntu02] => (item={'abuscar': '#PubkeyAuthentication', 'reemplazo': 'PubkeyAuthentication yes'})
ok: [ubuntu01] => (item={'abuscar': '#PubkeyAuthentication', 'reemplazo': 'PubkeyAuthentication yes'})

TASK [Instalar fail2ban] *********************************************************************
ok: [ubuntu01]
changed: [ubuntu02]

TASK [Asegurar que fail2ban esté iniciado y habilitado] *********************************************************************
ok: [ubuntu01]
ok: [ubuntu02]

RUNNING HANDLER [Reinicio SSH] *********************************************************************
changed: [ubuntu02]

PLAY RECAP *********************************************************************
ubuntu01                : ok=6   changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu02                : ok=7   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

sysadmin@ubuntu01 ~/obligatorio (main)> _
```

Resultados del playbook:

```
sysadmin@ubuntu02 ~> sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22/tcp ALLOW IN Anywhere
22/tcp (v6) ALLOW IN Anywhere (v6)

sysadmin@ubuntu02 ~> sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-08-07 22:16:05 UTC; 14min ago
     Docs: man:fail2ban(1)
  Main PID: 3939 (fail2ban-server)
    Tasks: 5 (limit: 2268)
   Memory: 22.3M (peak: 22.6M)
      CPU: 4.821s
   CGroup: /system.slice/fail2ban.service
           └─3939 /usr/bin/python3 /usr/bin/fail2ban-server -xf start
```

Captura de /etc/ssh/sshd_config

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no
```

Anexo

Declaración de autoría:

El presente documento fue elaborado en su totalidad por los alumnos:

Nicolás Cameto - N° de estudiante: 324408

Francisco Polido - N° de estudiante 343332

Bibliografía:

- ❖ Apuntes de clase.
- ❖ Material de clase.
- ❖ [Módulos de Ansible: qué son y cómo funcionan](#)
- ❖ [ansible.builtin.command module – Execute commands on targets — Ansible Community Documentation](#)
- ❖ [ansible.builtin.service module – Manage services — Ansible Community Documentation](#)
- ❖ [ansible.builtin.dnf module – Manages packages with the dnf package manager — Ansible Community Documentation](#)
- ❖ [ansible.posix.mount module – Control active and configured mount points — Ansible Community Documentation](#)
- ❖ [Handlers: running operations on change — Ansible Community Documentation](#)
- ❖ [ansible.builtin.file module – Manage files and file properties — Ansible Community Documentation](#)
- ❖ [ansible.posix.firewalld module – Manage arbitrary ports/services with firewalld — Ansible Community Documentation](#)
- ❖ [4.6. Configuración del servidor NFS | Gestión de sistemas de archivos | Red Hat Enterprise Linux | 8 | Red Hat Documentation](#)
- ❖ [Configure NFS Server and Client using Ansible | by Meher Askri | Medium](#)
- ❖ [ansible.builtin.package module – Generic OS package manager — Ansible Community Documentation](#)
- ❖ [Ansible Galaxy - ansible.posix](#)
- ❖ [community.general.ufw module – Manage firewall with UFW — Ansible Community Documentation](#)

- ❖ [apt – Manages apt-packages — Ansible Documentation](#)
- ❖ [ansible.builtin.apt module – Manages apt-packages — Ansible Community Documentation](#)