

Obligatorio

Taller Linux 2025



Alumnos:

Mathias Rivero n°284729

Andrés Blanco n°183261

Docente:

Enrique Verdes

Carrera:

Analista en infraestructura informática

Fecha: 2025-08-10

Contenido

Declaracion de autoria.....	3
Abstract.....	3
Tarea 1: Instalar servidores.....	4
Instalar Ansible Core:.....	4
Acceso desde el controller (bastión):.....	4
Estructura de directorios final:.....	5
Capturas configuraciones :.....	6
Configuración particiones de los filesystem de Centos Stream 9.....	6
Configuración particiones de los filesystem de Ubuntu 24.04.....	7
Tarea 2: Archivo de inventario Ansible.....	8
Inventario por defecto:.....	9
Prueba de inventario:.....	9
ansible-inventory --list.....	9
ansible all -m ping.....	10
Tarea 3: Comandos ad-hoc.....	11
Objetivos:.....	11
Comandos utilizados:.....	11
1. Usuarios en Ubuntu.....	11
2. Uso de memoria en todos los servidores.....	12
3. Verificar instalación de chrony.....	12
4. Verificar si chronyd está activo.....	12
Tarea 4: Playbooks NFS y Hardening.....	13
Resultado al ejecutar el playbook nfs_setup.yml.....	13
NFS UTILS instalado.....	15
Servicio NFS habilitado en firewall.....	15
Exportación NFS.....	15
Resultado final del playbook.....	15
Resultado al ejecutar playbook hardening.yml.....	16
Actualizar sistema captura:.....	17
Firewall UFW captura:.....	17
Deshabilitar root login y forzar ingreso con clave ssh.....	18
Instalación y funcionamiento de fail2ban.....	19
Tarea 5: Preguntas teóricas.....	20
1. ¿Qué es Ansible?.....	20
2. ¿Qué es un playbook?.....	20
3. ¿Qué contiene un inventario?.....	20
4. ¿Qué es un módulo?.....	20
5. Ventajas de Ansible.....	20
Tarea 6: README.....	21
Conclusiones.....	22
Bibliografía.....	22

Declaracion de autoria

Nosotros, Mathias Rivero y Andrés Blanco, declaramos que el trabajo que se presenta en esta obra es de nuestra propia autoría. Podemos asegurar que:

La obra fue producida en su totalidad mientras realizamos el taller de Linux.

Cuando hemos consultado el trabajo de otros, lo hemos atribuido con claridad.

Cuando hemos citado obras de otros, hemos indicado las fuentes.

Excepto esas citas, la obra es enteramente nuestra.

En la obra hemos acusado recibo de las ayudas recibidas.

Cuando la obra se basa en trabajos conjuntos, hemos explicado claramente qué parte es nuestra.

Ninguna parte de este trabajo ha sido publicada previamente.

Abstract

Este documento detalla la implementación de un entorno de servidores Linux (CentOS Stream 9 y Ubuntu 24.04) utilizando Ansible para automatizar la configuración de servicios críticos como NFS y hardening de seguridad. Se incluyen playbooks, inventarios, comandos ad-hoc y evidencias de ejecución, siguiendo las pautas del taller. La solución demuestra cómo Ansible simplifica la gestión de infraestructura como código.

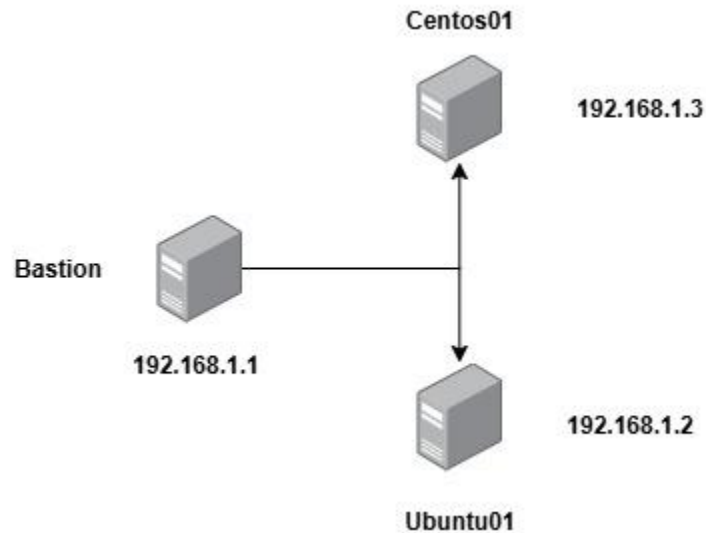
Objetivos

- ❖ Implementar servidores CentOS y Ubuntu con configuración específica.
- ❖ Automatizar la configuración de NFS en CentOS y hardening en Ubuntu mediante playbooks de Ansible.
- ❖ Documentar el proceso con capturas de pantalla, reflexiones y respuestas teóricas.

Tarea 1: Instalar servidores

Objetivo: Tener dos máquinas virtuales funcionales: una con CentOS Stream 9 y otra con Ubuntu 24.04, con estructura de particionado específica y dos interfaces de red.

Se realizó la conexión con un repo de git tal y como se explico en clase para ir documentando el proceso: [Repo de GIT utilizado](#)



Instalar Ansible Core:

```
sudo dnf install ansible-core -y
```

Acceso desde el controller (bastión):

Ejecutar comandos:

```
ssh-keygen
```

copiar la clave pública a los otros servidores, centos y ubuntu:

```
ssh-copy-id sysadmin@192.168.1.3 y lo mismo para ubuntu 192.168.1.2
```

Colecciones de ansible:

```
ansible-galaxy collection install ansible.posix
```

```
ansible-galaxy collection install community.general
```

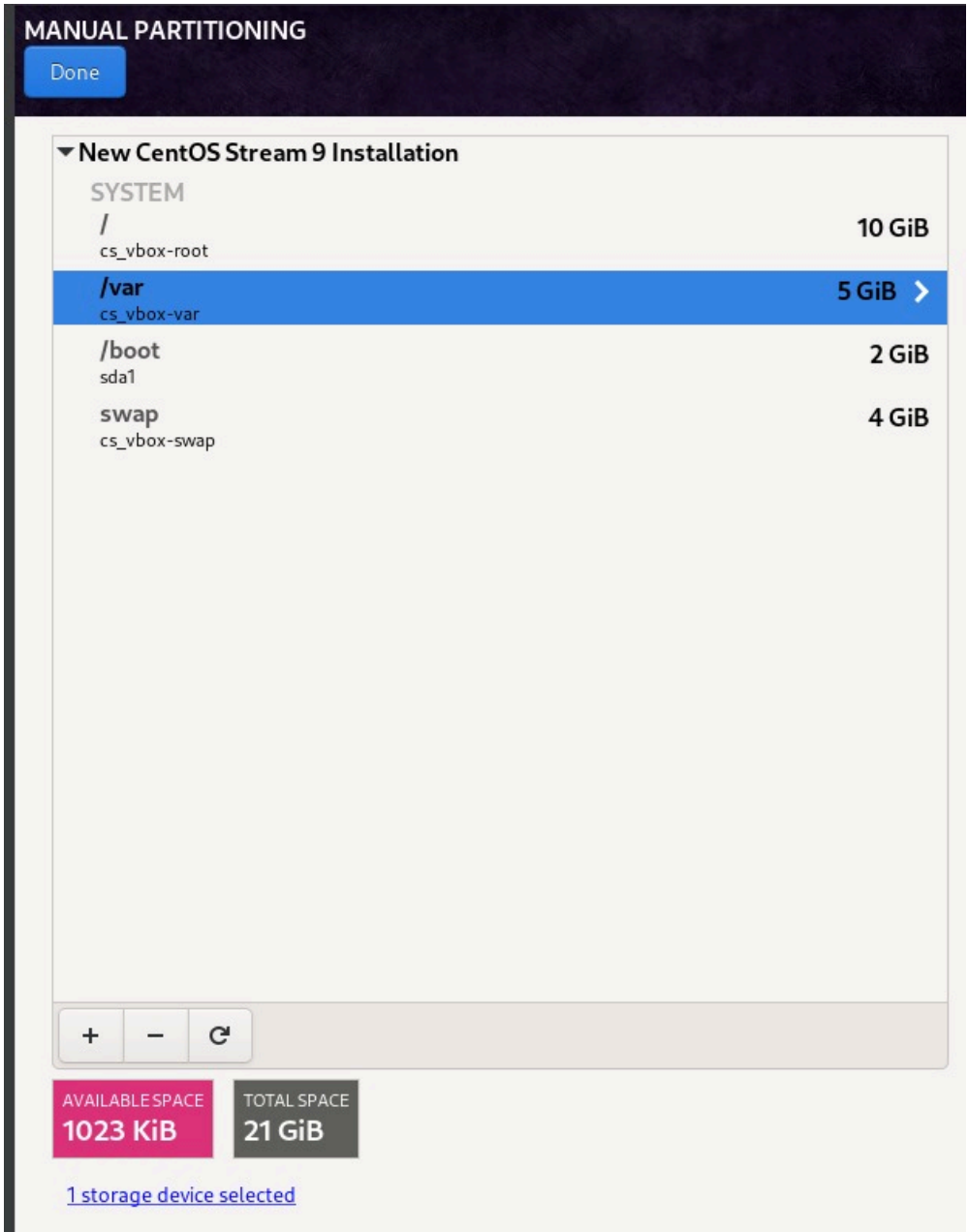
para ejecutar firwalld ya que viene en **ansible.posix** y para UFW que viene en **community.general**.

Estructura de directorios final:

```
[sysadmin@Bastion01 Obligatorio_2025]$ tree
.
├── ansible.cfg
├── documentos
│   ├── ansible_basics.txt
│   └── Prueba-ObligatorioTaller_Linux_2025_mriveryo_ablanco.docx.pdf
├── inventarios
│   ├── desarrollo
│   │   └── group_vars
│   │       └── variables_centos.yaml
│   └── hosts
│       └── inventory.ini
├── playbooks
│   ├── hardening.yml
│   └── nfs_setup.yml
└── README.txt

6 directories, 8 files
[sysadmin@Bastion01 Obligatorio_2025]$
```

Capturas configuraciones :
Configuración particiones de los filesystem de Centos Stream 9



Configuración particiones de los filesystem de Ubuntu 24.04

Storage configuration				[Help]
RESUMEN DEL SISTEMA DE ARCHIVOS				
PUNTO DE MONTAJE	TAMAÑO	TIPO	TIPO DE DISPOSITIVO	
[/	10.000G	new ext4	new LVM logical volume	►]
[/boot	2.000G	new ext4	new partition of disco	local ►]
[/boot/efi	1.049G	new fat32	new partition of disco	local ►]
[/swap	4.000G	new ext4	new LVM logical volume	►]
[/var	5.000G	new ext4	new LVM logical volume	►]

Configuración de red y prueba de conexión

Se asigna como se muestra en el diagrama la ip 192.168.1.1 para la red local al Bastión, el Ubuntu 24.04 ira con ip 192.168.1.2, por ultimo el centos ira con la ip 192.168.1.3 , se demuestra conectividad desde el Bastión en la siguiente captura:

```

lsysadmin@172 ~]$ 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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:01:06:1d brd ff:ff:ff:ff:ff:ff
    inet 172.28.7.143/20 brd 172.28.15.255 scope global dynamic noprefixroute eth0
        valid_lft 86000sec preferred_lft 86000sec
    inet6 fe80::215:5dff:fe01:61d/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:01:06:1e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global noprefixroute eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe01:61e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
lsysadmin@172 ~]$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=1.05 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=1.18 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=1.41 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=1.23 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=1.37 ms
^C
--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 1.053/1.247/1.409/0.130 ms
lsysadmin@172 ~]$ ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.581 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.559 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.493 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.604 ms
64 bytes from 192.168.1.3: icmp_seq=5 ttl=64 time=0.823 ms
^C
--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4099ms

```

Tarea 2: Archivo de inventario Ansible

Objetivos:

- Crear archivo **inventory.ini**:
- Probar la conectividad con los nodos usando Ping
- Captura de salida con evidencias del trabajo realizado

sudo dnf install -y ansible-core

Tener el usuario ansible creado y funcionando en los servidores Ubuntu y CentOS.

Haber hecho ssh-copy-id desde el controlador a los nodos.

Comandos:

Crear directorio:

mkdir /Obligatorio_2025

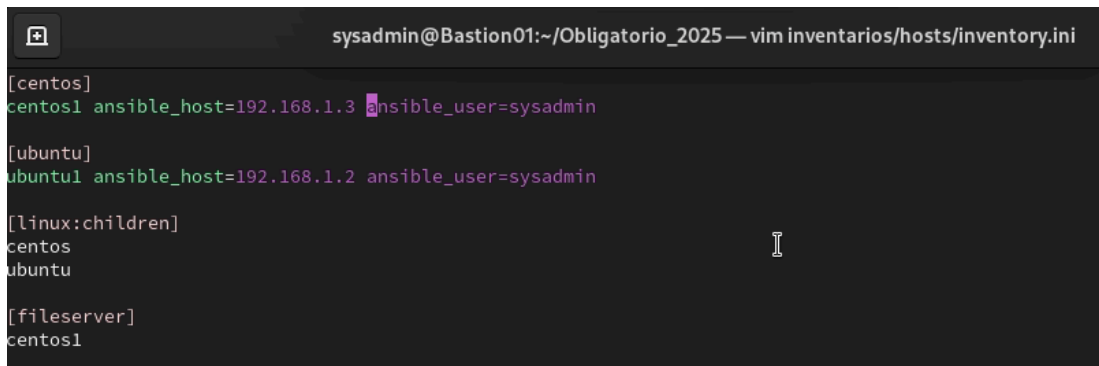
(así con el resto de carpetas hasta formar el árbol de directorios tal y como se muestra en pág. 5)

Moverse a la carpeta del directorio:

cd /Obligatorio_2025

Ya posicionado crear el archivo inventory.ini (en este caso creamos la carpeta inventario para alojarse allí tal y como está en la estructura de directorios mostrada anteriormente)

vim inventory.ini



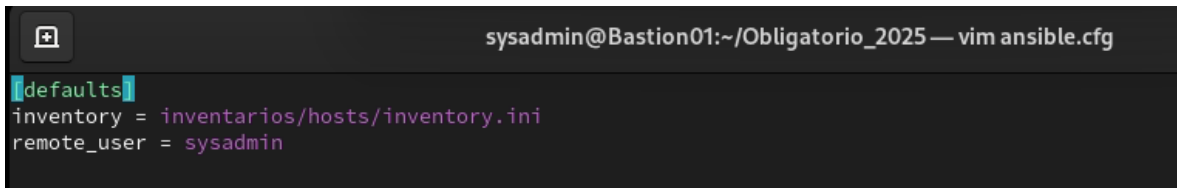
```
sysadmin@Bastion01:~/Obligatorio_2025 — vim inventarios/hosts/inventory.ini
[centos]
centos1 ansible_host=192.168.1.3 ansible_user=sysadmin

[ubuntu]
ubuntu1 ansible_host=192.168.1.2 ansible_user=sysadmin

[linux:children]
centos
ubuntu

[fileserver]
centos1
```


Inventario por defecto:



```
sysadmin@Bastion01:~/Obligatorio_2025 — vim ansible.cfg
[defaults]
inventory = inventarios/hosts/inventory.ini
remote_user = sysadmin
```

Prueba de inventario:

ansible-inventory --list



```
sysadmin@Bastion01:~/Obligatorio_2025
[sysadmin@Bastion01 Obligatorio_2025]$ ansible-inventory --list
{
  "_meta": {
    "hostvars": {
      "centos1": {
        "ansible_host": "192.168.1.3"
      },
      "ubuntu1": {
        "ansible_host": "192.168.1.2"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped",
      "linux",
      "webserver"
    ]
  },
  "centos": {
    "hosts": [
      "centos1"
    ]
  },
  "linux": {
    "children": [
      "centos",
      "ubuntu"
    ]
  },
  "ubuntu": {
    "hosts": [
      "ubuntu1"
    ]
  },
  "webserver": {
    "hosts": [
      "centos1"
    ]
  }
}
```

ansible all -m ping

```
[sysadmin@Bastion01 Obligatorio_2025]$ ansible all -m ping
centos1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
[sysadmin@Bastion01 Obligatorio_2025]$
```

Tarea 3: Comandos ad-hoc

Objetivos:

- Listar usuarios del servidor Ubuntu.
- Mostrar uso de memoria en ambos servidores.
- Verificar que el servicio chrony esté instalado y activo en CentOS.

Comandos utilizados:

1. Usuarios en Ubuntu

ansible ubuntu -m command -a "cut -d: -f1 /etc/passwd"

```
[sysadmin@Bastion01 Obligatorio_2025]$ ansible ubuntu -m command -a "cut -d: -f1 /etc/passwd"
ubuntu1 | 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
dhcpcd
messagebus
systemd-resolve
pollinate
polkitd
syslog
uidd
tcpdump
tss
landscape
fwupd-refresh
usbmux
sshd
sysadmin
[sysadmin@Bastion01 Obligatorio_2025]$
```

2. Uso de memoria en todos los servidores

ansible all -m command -a "free -h"

```
[sysadmin@Bastion01 Obligatorio_2025]$ ansible all -m command -a "free -h"
ubuntu1 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    1,9Gi       378Mi       1,4Gi       960Ki       191Mi       1,5Gi
Swap:    1,9Gi           0B        1,9Gi
centos1 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    1,7Gi       408Mi       1,2Gi        9,0Mi       254Mi       1,3Gi
Swap:    4,0Gi           0B        4,0Gi
```

3. Verificar instalación de chrony

```
sysadmin@Bastion01:~/Obligatorio_2025

[sysadmin@Bastion01 Obligatorio_2025]$ ansible centos -m dnf -a "name=chrony state=present" --become -K
BECOME password:
centos1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Nothing to do",
  "rc": 0,
  "results": []
}
[sysadmin@Bastion01 Obligatorio_2025]$
```

4. Verificar si chronyd está activo

```
sysadmin@Bastion01:~/Obligatorio_2025

[sysadmin@Bastion01 Obligatorio_2025]$ ansible centos -m shell -a "systemctl is-active chronyd"
centos1 | CHANGED | rc=0 >>
active
[sysadmin@Bastion01 Obligatorio_2025]$ ansible centos -a "systemctl is-active chronyd"
centos1 | CHANGED | rc=0 >>
active
[sysadmin@Bastion01 Obligatorio_2025]$ ansible centos -m command -a "systemctl is-active chronyd"
centos1 | CHANGED | rc=0 >>
active
[sysadmin@Bastion01 Obligatorio_2025]$
```

Otra opción

```
sysadmin@Bastion01:~/Obligatorio_2025

[sysadmin@Bastion01 Obligatorio_2025]$ ansible centos -m service -a "name=chronyd state=started enabled=yes" --become -K
BECOME password:
centos1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "enabled": true,
  "name": "chronyd",
  "state": "started",
  "status": {
    "AccessSELinuxContext": "system_u:object_r:chronyd_unit_file_t:s0",
    "ActiveEnterTimestamp": "Thu 2025-08-07 06:41:00 -03",
    "ActiveEnterTimestampMonotonic": "11681147",
    "ActiveExitTimestampMonotonic": "0",
    "ActiveState": "active",
    "After": "systemd-journald.socket systemd-tmpfiles-setup.service basic.target -.mount tmp.mount system.slice snpt.service sysinit.target ntpd.service ntpdate.service var.mount",

```

Tarea 4: Playbooks NFS y Hardening

Resultado al ejecutar el playbook nfs_setup.yml

Comandos ejecutados previamente a la ejecución :

```

sysadmin@Bastion01:~/Obligatorio_2025
└─ README.txt
5 directories, 9 files
[sysadmin@Bastion01 Obligatorio_2025]$ vim inventarios/desarrollo/group_vars/variables_centos.yaml
[sysadmin@Bastion01 Obligatorio_2025]$ vim playbooks/nfs_setup.yml
[sysadmin@Bastion01 Obligatorio_2025]$ vim playbooks/nfs_setup.yml
[sysadmin@Bastion01 Obligatorio_2025]$ vim playbooks/nfs_setup.yml
[sysadmin@Bastion01 Obligatorio_2025]$ ansible-playbook playbooks/nfs_setup.yml --check --ask-become-pass
BECOME password:
[WARNING]: Collection ansible.posix does not support Ansible version 2.14.18

PLAY [Configurar servidor NFS en CentOS] *****

TASK [Gathering Facts] *****
ok: [centos1]

TASK [Instalar paquete nfs-utils] *****
changed: [centos1]

TASK [Permitir servicio NFS en firewall] *****
changed: [centos1]

TASK [Crear directorio compartido] *****
changed: [centos1]

TASK [Configurar exportación NFS] *****
changed: [centos1]

RUNNING HANDLER [Habilitar servicio NFS] *****
fatal: [centos1]: FAILED! => {"changed": false, "msg": "Could not find the requested service nfs-server: host"}

PLAY RECAP *****
centos1 : ok=5 changed=4 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0

[sysadmin@Bastion01 Obligatorio_2025]$ vim playbooks/nfs_setup.yml
[sysadmin@Bastion01 Obligatorio_2025]$ systemctl list-unit-files | grep nfs
[sysadmin@Bastion01 Obligatorio_2025]$ systemctl status rpcbind
Unit rpcbind.service could not be found.
[sysadmin@Bastion01 Obligatorio_2025]$ vim playbooks/nfs_setup.yml

```

Puntos a tener en cuenta: NFS requiere la instalación de la colección `ansible.posix` desde Ansible Galaxy. Para ello, se utiliza el comando:

ansible-galaxy collection install ansible.posix

Este comando descarga la colección y la instala en el directorio predeterminado de colecciones de Ansible.

Luego de tener esto ejecutamos el playbook:

`ansible-playbook playbooks/nfs_setup.yml -K` (el `-K` es para ejecutarlo con permisos elevados, es decir nos pedirá la contraseña del usuario `sysadmin` que está en el grupo `wheels`, para que se ejecute con privilegios elevados)

```
[sysadmin@Bastion01 Obligatorio_2025]$ ansible-playbook playbooks/nfs_setup.yml -K
BECOME password:
[WARNING]: Collection ansible.posix does not support Ansible version 2.14.18

PLAY [Configurar servidor NFS en CentOS] *****

TASK [Gathering Facts] *****
ok: [centos1]

TASK [Instalar paquete nfs-utils] *****
changed: [centos1]

TASK [Permitir servicio NFS en firewall] *****
changed: [centos1]

TASK [Crear directorio compartido] *****
changed: [centos1]

TASK [Configurar exportación NFS] *****
changed: [centos1]

RUNNING HANDLER [Habilitar servicio NFS] *****
changed: [centos1]

RUNNING HANDLER [Recargar firewall] *****
changed: [centos1]

RUNNING HANDLER [Recargar exportaciones NFS] *****
changed: [centos1]

PLAY RECAP *****
centos1 : ok=8  changed=7  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

[sysadmin@Bastion01 Obligatorio_2025]$
```

Si volvemos a ejecutar nuevamente el comando, al estar todo instalado y no ver algún cambio de estado, veremos que los handlers no son ejecutados, optimizando así para no tener que estar haciendo reinicios innecesarios o habilitar servicios que ya están corriendo.

```
[sysadmin@Bastion01 Obligatorio_2025]$ ansible-playbook playbooks/nfs_setup.yml -K
BECOME password:
[WARNING]: Collection ansible.posix does not support Ansible version 2.14.18

PLAY [Configurar servidor NFS en CentOS] *****

TASK [Gathering Facts] *****
ok: [centos1]

TASK [Instalar paquete nfs-utils] *****
ok: [centos1]

TASK [Permitir servicio NFS en firewall] *****
ok: [centos1]

TASK [Crear directorio compartido] *****
ok: [centos1]

TASK [Configurar exportación NFS] *****
ok: [centos1]

PLAY RECAP *****
centos1 : ok=5  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

[sysadmin@Bastion01 Obligatorio_2025]$
```

NFS UTILS instalado

rpm -q nfs-utils

```
sysadmin@Bastion01 Obligatorio_2025$ ssh 192.168.1.3
Last login: Tue Aug 12 13:13:16 2025 from 192.168.1.1
[sysadmin@Centos01 ~]$ rpm -q nfs-utils
nfs-utils-2.5.4-38.el9.x86_64
```

Servicio NFS habilitado en firewall

sudo firewall-cmd --list-all

```
[sysadmin@Centos01 ~]$ sudo firewall-cmd --list-all
[sudo] password for sysadmin:
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0 eth1
  sources:
  services: cockpit dhcpv6-client nfs ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Exportación NFS

sudo exportfs -v

```
[sysadmin@Centos01 ~]$ sudo exportfs -v
/var/nfs_shared
<world>(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,no_root_squash,no_all_squash)
[sysadmin@Centos01 ~]$
```

Resultado final del playbook

El recurso compartido quedó creado según lo que se sollicito en la letra del obligatorio

```
sysadmin@Centos01:~$ ls -ld /var/nfs_shared
drwxrwxrwx. 2 nobody nobody 6 ago 10 11:26 /var/nfs_shared
[sysadmin@Centos01 ~]$ cat /etc/exports
/var/nfs_shared *(rw,sync,no_root_squash)
[sysadmin@Centos01 ~]$
```

Resultado al ejecutar playbook hardening.yml

Importante, tener previamente instalada la colección de ansible community general:

ansible-galaxy collection install community.general

Ejecutar el comando con privilegios elevados:

ansible-playbook playbooks/herdening.yml -K

```
[sysadmin@Bastion01 Obligatorio_2025]$ ansible-playbook playbooks/hardening.yml -K
BECOME password:
[WARNING]: Collection community.general does not support Ansible version 2.14.18

PLAY [Aplicar hardening en servidores Ubuntu] *****

TASK [Gathering Facts] *****
ok: [ubuntu1]

TASK [Actualizar sistema] *****
changed: [ubuntu1]

TASK [Configurar firewall UFW] *****
changed: [ubuntu1]

TASK [Deshabilitar login root] *****
changed: [ubuntu1]

TASK [Forzar autenticación por clave] *****
[changed: [ubuntu1]

TASK [Instalar fail2ban] *****
changed: [ubuntu1]

TASK [Habilitar fail2ban] *****
ok: [ubuntu1]

RUNNING HANDLER [Reiniciar sistema] *****
changed: [ubuntu1]

RUNNING HANDLER [Reiniciar SSH] *****
changed: [ubuntu1]

PLAY RECAP *****
ubuntu1          : ok=9    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[sysadmin@Bastion01 Obligatorio_2025]$
```


Actualizar sistema captura:

```
2025-08-10 21:33:01 upgrade libnvmelt64:amd64 1.8-3build1 1.8-3ubuntu1
2025-08-10 21:33:01 upgrade lvm2:amd64 2.03.16-3ubuntu3.1 2.03.16-3ubuntu3.2
2025-08-10 21:33:01 upgrade pollinate:all 4.33-3.1ubuntu1 4.33-3.1ubuntu1.1
2025-08-10 21:33:02 upgrade software-properties-common:all 0.99.49.1 0.99.49.2
2025-08-10 21:33:02 upgrade python3-software-properties:all 0.99.49.1 0.99.49.2
2025-08-10 21:33:02 upgrade snapd:amd64 2.66.1+24.04 2.68.5+ubuntu24.04.1
2025-08-10 21:33:03 upgrade sosreport:amd64 4.7.2-0ubuntu1~24.04.2 4.8.2-0ubuntu0~24.04.2
sysadmin@ubuntu:~$
```

Firewall UFW captura:

```
sysadmin@ubuntu:~$ dpkg -l | grep ufw
ii ufw                                0.36.2-6                all          program for managing a Netfilter firewall
sysadmin@ubuntu:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

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

sysadmin@ubuntu:~$
```

Deshabilitar root login y forzar ingreso con clave ssh

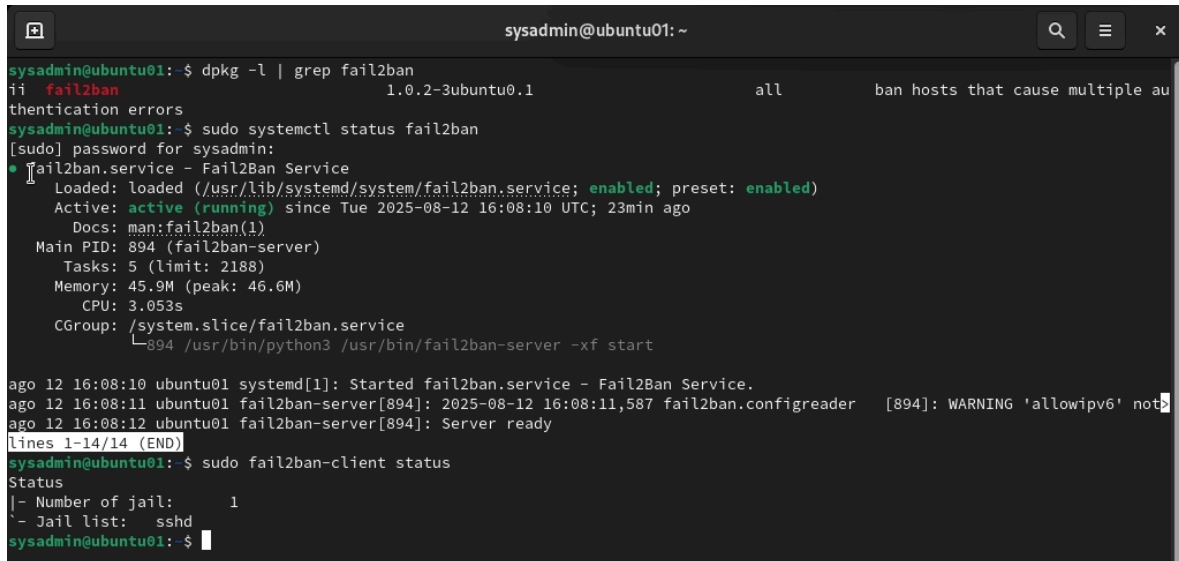
```
sysadmin@ubuntu01: ~  
#LogLevel INFO  
# Authentication:  
  
#LoginGraceTime 2m  
PermitRootLogin no  
#StrictModes yes  
#MaxAuthTries 6  
#MaxSessions 10  
  
#PubkeyAuthentication yes  
  
# Expect .ssh/authorized_keys2 to be disregarded by default in future.  
#AuthorizedKeysFile .ssh/authorized_keys .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 no  
#PermitEmptyPasswords no
```

Instalación y funcionamiento de fail2ban

`dpkg -l | grep fail2ban`

`sudo systemctl status fail2ban`

`sudo fail2ban-client status`



```
sysadmin@ubuntu01: ~
sysadmin@ubuntu01:~$ dpkg -l | grep fail2ban
ii fail2ban 1.0.2-3ubuntu0.1 all ban hosts that cause multiple authentication errors
sysadmin@ubuntu01:~$ sudo systemctl status fail2ban
[sudo] password for sysadmin:
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-08-12 16:08:10 UTC; 23min ago
     Docs: man:fail2ban(1)
    Main PID: 894 (fail2ban-server)
      Tasks: 5 (limit: 2188)
    Memory: 45.9M (peak: 46.6M)
       CPU: 3.053s
    CGroup: /system.slice/fail2ban.service
            └─894 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

ago 12 16:08:10 ubuntu01 systemd[1]: Started fail2ban.service - Fail2Ban Service.
ago 12 16:08:11 ubuntu01 fail2ban-server[894]: 2025-08-12 16:08:11,587 fail2ban.configreader [894]: WARNING 'allowip' not
ago 12 16:08:12 ubuntu01 fail2ban-server[894]: Server ready
lines 1-14/14 (END)
sysadmin@ubuntu01:~$ sudo fail2ban-client status
Status
|- Number of jail:      1
  '- Jail list:  sshd
sysadmin@ubuntu01:~$
```

Tarea 5: Preguntas teóricas

1. ¿Qué es Ansible?

Es una herramienta de automatización IT. Permite instalar software, configurar servidores, administrar usuarios, etc.

Una de sus características es que es un motor open source, que se puede utilizar para instalar software automatizar tareas cotidianas, configuraciones de elementos de infraestructura y red.

2. ¿Qué es un playbook?

Es un archivo YAML que describe tareas automatizadas a ejecutar sobre los nodos.

Se utiliza para organizar los procesos de TI, un playbook en si es un archivo .YAML el cual contiene por lo menos un play , donde sirve para definir el estado deseado de un sistema o servicio.

Se puede agregar que son conjuntos ordenados de tareas , y se ejecuta desde el archivo de inventario de ansible.

3. ¿Qué contiene un inventario?

Una lista de hosts agrupados, con IPs e información sobre como conectarse a los mismos

Sus componentes principales vistos en clase:

Hosts, grupos, variables, conexiones. El formato de este archivo puede ser "inventario.ini"

4. ¿Qué es un módulo?

Un bloque de funcionalidad, por ejemplo: `ping`, `apt`, `service`, etc.

Son pequeños programas denominados módulos, Ansible, se conecta a los nodos o hosts y les inserta estos módulos. Los módulos se utilizan para realizar tareas de automatización en Ansible.

Ansible ya viene con módulos integrados, lo que ayuda a automatizar tareas o también se pueden escribir nuevos si el módulo no existe.

5. Ventajas de Ansible

No requiere agente.

Usa SSH.

Fácil de leer y mantener.

Tarea 6: README

```
# Configuración del Laboratorio

## Configuración de red

- **Servidor Bastion**: `192.168.1.1`
- **Servidor CentOS01**: `192.168.1.3`
- **Servidor Ubuntu01**: `192.168.1.2`

## Instalaciones y preparación

# Probar conectividad con todos los hosts

ansible all -m ping

# Instalar Ansible Core

sudo dnf install ansible-core -y

# Pasar claves SSH desde Bastion a los otros servidores

ssh-copy-id usuario@192.168.1.3
ssh-copy-id usuario@192.168.1.2

# Instalar colecciones necesarias de Ansible

ansible-galaxy collection install ansible.posix
ansible-galaxy collection install community.general

# Ejecutar con privilegios elevados (-K)

ansible-playbook playbooks/nfs_setup.yml -K
ansible-playbook playbooks/hardening.yml -K
```

Conclusiones

1. Ansible demostró ser eficiente para gestionar configuraciones heterogéneas (CentOS/Ubuntu).
2. Los handlers optimizan la ejecución al agrupar acciones sensibles (ej: reinicios).
3. La automatización con playbooks reduce errores humanos y asegura consistencia.

Bibliografía

<https://www.redhat.com/es/topics/automation/learning-ansible-tutorial>

<https://docs.ansible.com/>

Uso de IA

Se utilizó IA para mejorar gramaticalmente algunas definiciones y textos.

¡Gracias por su tiempo!.