

# Obligatorio

## Taller Linux 2025



**Alumnos:**

Mathias Rivero      n°284729

Andrés Blanco      n°183261

**Docente:**

Enrique Verdes

**Carrera:**

Analista en infraestructura informática

**Fecha: 2025-07-03**

## Contenido

<b>Declaracion de autoria.....</b>	<b>3</b>
<b>Abstract.....</b>	<b>3</b>
<b>Objetivos.....</b>	<b>3</b>
<b>Tarea 1: Instalar servidores.....</b>	<b>4</b>
<b>Acceso desde el controller (bastión):.....</b>	<b>4</b>
Capturas configuraciones : .....	5
Configuración particiones de los filesystem de Centos Stream 9.....	5
Configuración particiones de los filesystem de Ubuntu 24.04.....	6
Inventario por defecto:.....	8
<b>Prueba de inventario:.....</b>	<b>8</b>
ansible-inventory --list.....	8
ansible all -m ping.....	9
<b>Objetivos:.....</b>	<b>10</b>
<b>Comandos utilizados:.....</b>	<b>10</b>
1. Usuarios en Ubuntu.....	10
2. Uso de memoria en todos los servidores.....	11
3. Verificar instalación de chrony.....	11
4. Verificar si chronyd está activo.....	11
<b>Tarea 4: Playbooks NFS y Hardening.....</b>	<b>12</b>
<b>Tarea 5: Preguntas teóricas.....</b>	<b>13</b>
1. ¿Qué es Ansible?.....	13
2. ¿Qué es un playbook?.....	13
3. ¿Qué contiene un inventario?.....	13
4. ¿Qué es un módulo?.....	13
5. Ventajas de Ansible.....	13
<b>Tarea 6: Documentación y README.....</b>	<b>14</b>

## 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

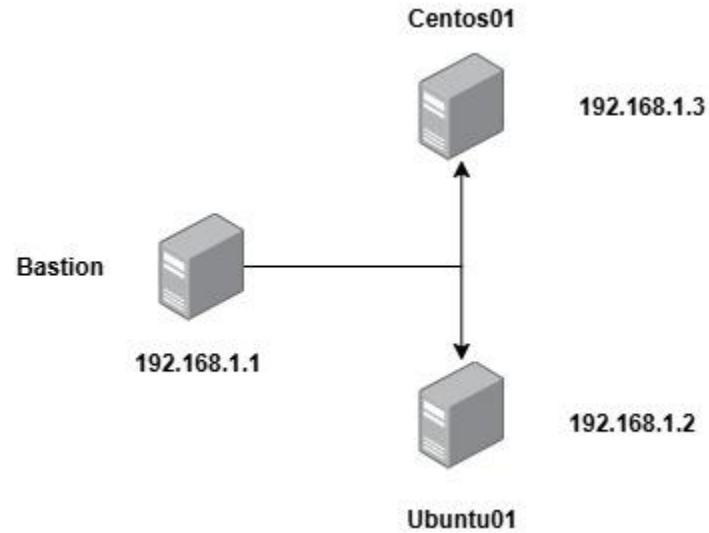
Este documento detalla la implementación de un entorno de servidores Linux (CentOS Stream 9 y Ubuntu 24.04) utilizando Ansible para automatizar la configuración de servicios críticos como NFS y hardening de seguridad. Se incluyen playbooks, inventarios, comandos ad-hoc y evidencias de ejecución, siguiendo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

## 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.



Acceso desde el controller (bastión):

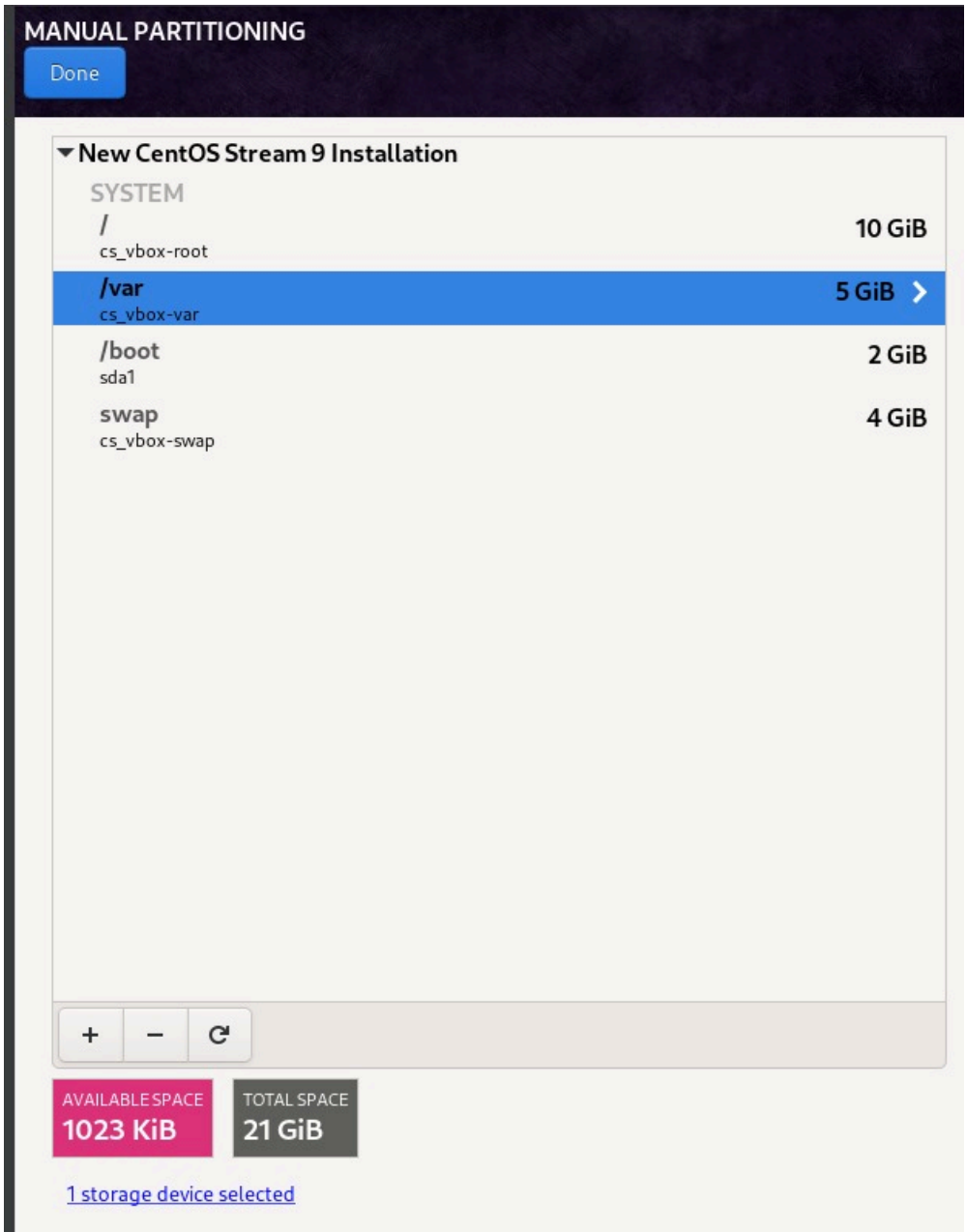
ssh-keygen

copiar la clave pública a los otros servidores

ssh sysadmin@192.168.1.3 y lo mismo para ubuntu 192.168.1.2

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 ~1$ 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 ~1$ 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 ~1$ 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:

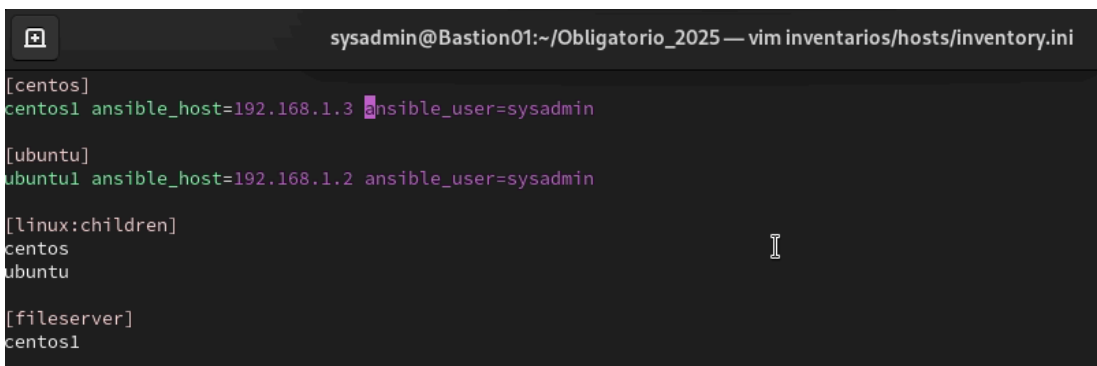
*mkdir /Obligatorio\_2025*

Moverse a la carpeta del directorio:

*cd /Obligatorio\_2025*

Ya posicionado crear el archivo inventory.ini (en este caso creamos la carpeta inventario para alojarlo allí)

*vim inventory.ini*



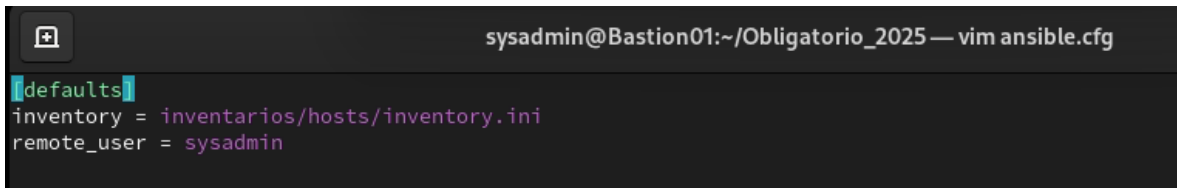
```
sysadmin@Bastion01:~/Obligatorio_2025 — vim inventarios/hosts/inventory.ini
[centos]
centos1 ansible_host=192.168.1.3 ansible_user=sysadmin

[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

```
[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

```
[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
[sysadmin@Bastion01 Obligatorio_2025]$
```

## 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 snmp.service sysinit.target ntpd.service ntpdate.service var.mount",
  }
}
[sysadmin@Bastion01 Obligatorio_2025]$
```

## Tarea 4: Playbooks NFS y Hardening

### Encabezado del Playbook

---

- name: Configurar servidor NFS en CentOS

hosts: centos

become: yes

vars:

nfs\_shared\_dir: /var/nfs\_shared

#vars: Define variables. Aquí establecemos la ruta del directorio compartido.

## 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: Documentación y README

```
[sysadmin@Bastion01 Obligatorio_2025]$ tree
.
├── ansible.cfg
├── documentos
├── inventarios
│   ├── desarrollo
│   │   └── group_vars
│   └── hosts
│       └── inventory.ini
├── playbooks
│   ├── hardening.yaml
│   └── playbooks.yaml
└── README.txt

6 directories, 5 files
[sysadmin@Bastion01 Obligatorio_2025]$
```

## 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/>

***¡Gracias por su tiempo!.***