

Universidad ORT Uruguay

Facultad de Ingeniería: Escuela de Tecnología

OBLIGATORIO TALLER DE SERVIDORES LINUX

Marcelo Sosa N°141855

Hernán Pintos N°151357

Grupo: N3A

Docente: Enrique Verdes

Declaración de autoría

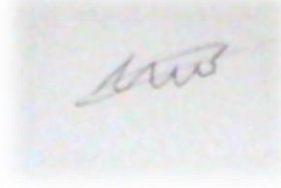
Nombres de los autores: Marcelo Sosa
Hernán Pintos

Declaración de autoría Nosotros, **Marcelo Sosa** y **Hernán Pintos**, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano. Aseguramos, bajo nuestra entera responsabilidad, que:

- La obra fue producida en su totalidad mientras realizábamos la carrera ANALISTA EN INFRAESTRUCTURA INFORMATICA;- Cuando hemos consultado trabajos publicados por otros, lo hemos atribuido con claridad;- Cuando hemos citado obras de otros autores, hemos indicado las fuentes. Con excepción de dichas citas, la obra es enteramente nuestra;- En la obra, hemos acusado recibo de las ayudas recibidas;- Cuando la obra se basa en trabajo realizado juntamente con otros no pertenecientes al equipo, hemos explicado claramente qué parte fue contribuida por dichos terceros, y qué parte fue contribuida por nosotros;- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto los casos en que se han realizado las aclaraciones correspondientes.

Hernán Pintos

Marcelo Sosa



INDICE

SECCION I - OBJETIVOS

SECCION II - ESQUEMA GENERAL

Diagrama general y configuración de red

SECCION III – INSTALACION DE MAQUINAS VIRTUALES

Configuración de Ubuntu

Configuración de CentOS

Configuración de redes

Llave publica GPG para SSH

Repositorio GITHUB

SECCION IV - Creación de estructura del repositorio y Pruebas requeridas

Pruebas de conexión

Ejecución de comandos AD-HOC

SECCION V – PLAYBOOKS

SCCION VI – PREGUNTAS

SECCION VI – RESULTADOS

PROBLEMAS ENCONTRADOS

ANEXOS:

Anexo I – Utilización de inteligencia artificial generativa

Anexo II – Playbooks

BIBLIOGRAFIA

SECCION I - OBJETIVOS

El objetivo de este documento es dar cumplimiento a las tareas solicitadas por el docente en el Taller de Aplicaciones LINUX con motivo del trabajo obligatorio del mismo.

Esto se realizará cumpliendo los siguientes ítems:

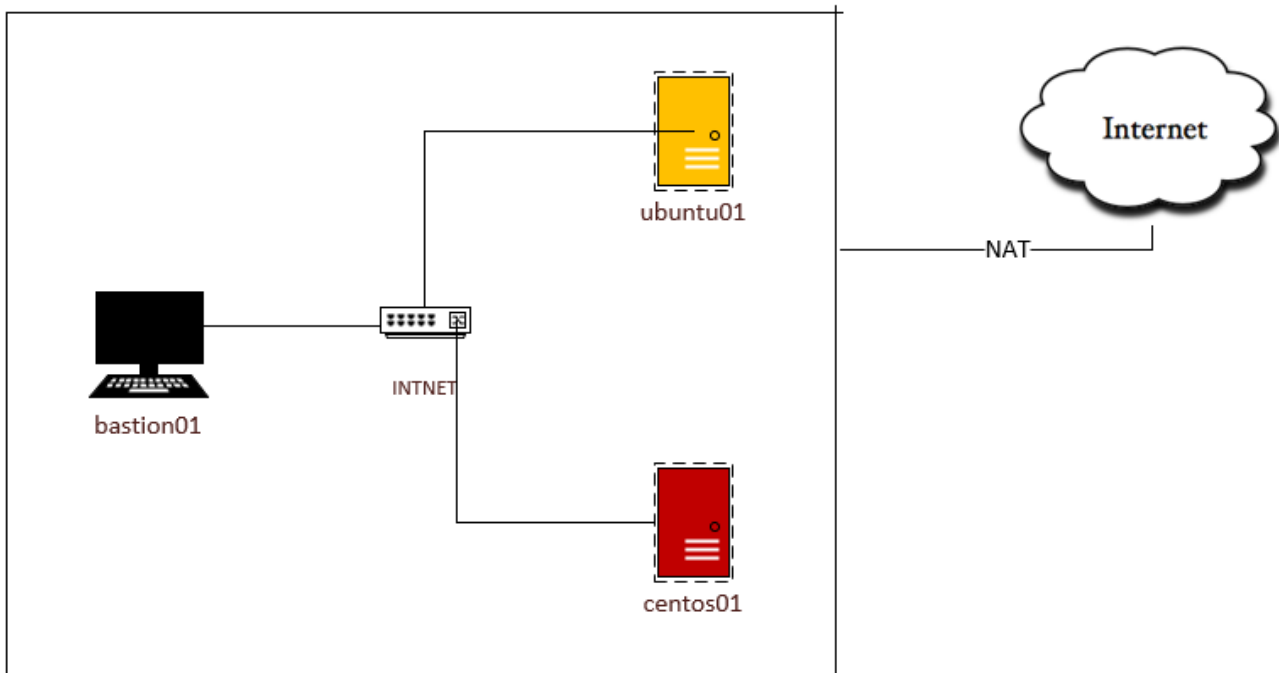
- Creación de archivo ZIP con archivos solicitados
- Creación de repositorio GitHub para realizar tareas
- Creación de Playbooks de ANSIBLE
- Documento de registros de ejecución e información
- Reflexiones sobre los desafíos planteados

Para lograr estos objetivos, utilizaremos los conocimientos y herramientas obtenidos en el taller de servidores LINUX, enfocados en la utilización de automatización con ANSIBLE a modo de satisfacer requerimientos de instalación y configuración.

SECCION II – ESQUEMA GENERAL

Se definió una nomenclatura a utilizar con el fin de identificar cada uno de los equipos que se encontraran en la red, a modo de simplificar la misma, esta contendrá nombres simples al tratarse de una estructura sencilla de pruebas.

Diagrama General



Todas las maquinas estarán conectadas mediante NAT en Virtual Box.

Tabla de Direcciones IP

Nombre	Rol	IP	Mascara
bastion01	Servidor Bastion	192.168.1.1	255.255.255.0
ubuntu01	Servidor Seguro	192.168.1.2	255.255.255.0
centos01	Servidor NFS	192.168.1.3	255.255.255.0

SECCION III - INSTALACION DE MAQUINAS VIRTUALES

Para el desarrollo de este trabajo se utilizó una maquina HOST para albergar la creación de las tres máquinas virtuales requeridas.

En la maquina se creó una cuenta especial de usuario con derechos de administración para poder tener un área de trabajo ordenada y ajena a otras tareas no pertinentes al curso.

Las máquinas virtuales fueron creadas y son ejecutadas desde discos de estado sólido para mejorar el rendimiento.

Hardware Utilizado

ASUS

CPU: INTEL i7 6700HQ x8 cores


RAM: 16 GB

Almacenamiento: 1M2 256 GB, X2 1TB HDD

Sistema Operativo HOST: Windows 11 Pro

Archivos de Imagen de LINUX

Se utilizaron las imágenes de LINUX provistas por el docente para la realización del trabajo disponibles en los equipos del laboratorio.

 CentOS-Stream-9-latest-x86_64-dvd1.iso	7/7/2025 19:36	Disc Image File	12.910.592 KB
 ubuntu-24.04.1-live-server-amd64.iso	7/7/2025 18:48	Disc Image File	2.708.862 KB

Preparación de las VM

Se utilizaron los siguientes parámetros para la creación de las máquinas y servidores.

Al poseer memoria RAM suficiente en el sistema HOST, le otorgamos 4 Gb de RAM a la maquina BASTION, ya que no se especificaba en la letra y la misma contara con entorno gráfico.

Se siguieron las siguientes especificaciones para las máquinas virtuales:

Nombre Maquina	CPU	RAM	/boot	/	/var	/home	SWAP
Bastion01	X2	4Gb	1Gb	10Gb		5Gb	8Gb
ubuntu01	X1	2Gb	2Gb	10Gb	5Gb	2Gb	4Gb
centos01	X1	2Gb	2Gb	10Gb	5Gb	2Gb	4Gb

La máquina BASTION se le da 4Gb de RAM ya que tiene entorno gráfico, como SWAP utilizamos el doble de esta por buenas prácticas.

Las otras maquinas contarán con las configuraciones indicadas en la lista superior

NOTA: El servidor bastion01 será instalado directamente con entorno gráfico, en un ambiente de producción, se debería de instalar una versión minimizada e instalar el entorno grafico por separado para evitar tener paquetes innecesarios.

New CentOS Stream 9 Installation	
DATA	
/home cs_vbox-home	5 GiB
SYSTEM	
/	10 GiB
/cs_vbox-root	
/boot sda1	1024 MiB
swap cs_vbox-swap	4 GiB

Configuración de ubuntu01

Configuración Vbox

General

Nombre: ubuntu01
Sistema operativo: Ubuntu (64-bit)
Grupos: Nuevo grupo

Sistema

Memoria base: 2048 MB
Orden de arranque: Disquete, Óptica, Disco duro
Aceleración: Paginación anidada, Paravirtualización KVM

Pantalla

Memoria de vídeo: 16 MB
Controlador gráfico: VMSVGA
Servidor de escritorio remoto: Inhabilitado
Grabación: Inhabilitado

Almacenamiento

Controlador: IDE
Dispositivo IDE secundario 0: [Unidad óptica] ubuntu-24.04.1-live-
Controlador: SATA
Puerto SATA 0: ubuntu01.vdi (Normal, 20,00 GB)

Audio

Controlador de anfitrión: Predeterminado
Controlador: ICH AC97

Red

Adaptador 1: Intel PRO/1000 MT Desktop (NAT)
Adaptador 2: Intel PRO/1000 MT Desktop (Red interna, «intnet»)

Configuración File System

Para la configuración de particiones, se utiliza LVM el cual cumple con el tamaño solicitado para cada una de las mismas.

Adicionalmente, incluimos /home con ~4Gb

SSH configuration

```
You can choose to install the OpenSSH s
[X] Install OpenSSH server
```

Storage configuration

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
[/]	10.000G	new ext4	new LVM logical volume
[/boot]	2.000G	new ext4	new partition of local disk
[/home]	3.996G	new ext4	new LVM logical volume
[/var]	5.000G	new ext4	new LVM logical volume
[SWAP]	4.000G	new swap	new LVM logical volume

Configuración CentOS

MANUAL PARTITIONING

Done

▼ New CentOS Stream 9 Installation

DATA

/home 4 GiB >
cs_vbox-home

SYSTEM

/ 10 GiB
cs_vbox-root

/var 5 GiB
cs_vbox-var

/boot 2 GiB
sda1

swap 4 GiB
cs_vbox-swap

General

Nombre: centos01
Sistema operativo: Red Hat (64-bit)
Grupos: Nuevo grupo

Sistema

Memoria base: 2048 MB
Orden de arranque: Disquete, Óptica, Disco duro
Aceleración: Paginación anidada, PAE/NX, Paravirtualización KVM

Pantalla

Memoria de vídeo: 16 MB
Controlador gráfico: VMSVGA
Servidor de escritorio remoto: Inhabilitado
Grabación: Inhabilitado

Almacenamiento

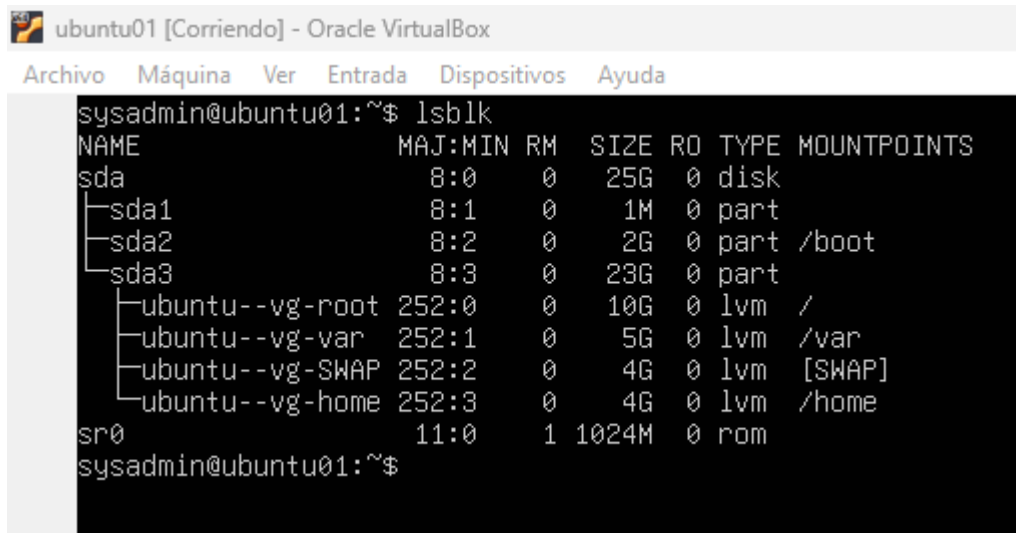
Controlador: IDE
Dispositivo IDE secundario 0: [Unidad óptica] CentOS-Stream-9-latest-x86_64-dvd1.iso (12,31 GB)
Controlador: SATA
Puerto SATA 0: centos01.vdi (Normal, 25,00 GB)

Audio

Controlador de anfitrión: Predeterminado
Controlador: ICH AC97

Red

Adaptador 1: Intel PRO/1000 MT Desktop (NAT)
Adaptador 2: Intel PRO/1000 MT Desktop (Red interna, «intnet»)

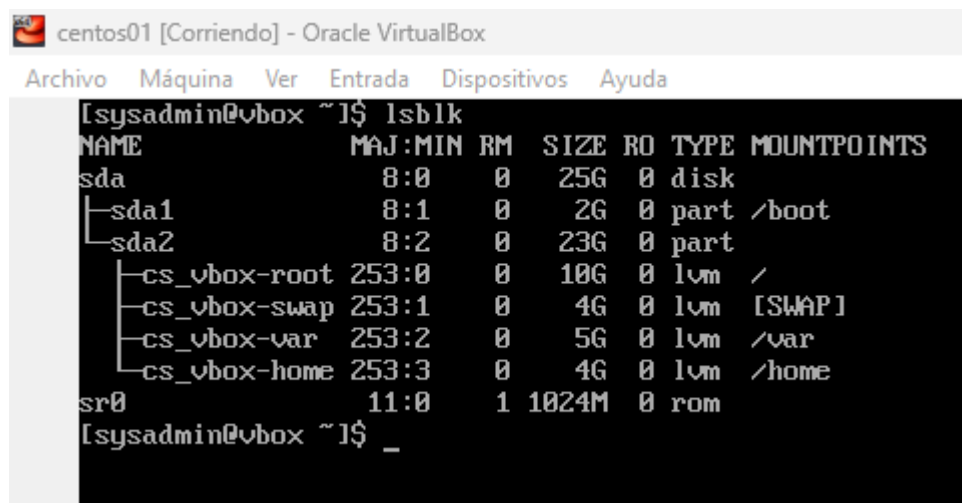


```

sysadmin@ubuntu01:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                  8:0      0   25G  0 disk
├─sda1                              8:1      0    1M  0 part
├─sda2                              8:2      0    2G  0 part /boot
├─sda3                              8:3      0   23G  0 part
│   └─ubuntu--vg-root              252:0    0   10G  0 lvm  /
│       └─ubuntu--vg-var          252:1    0    5G  0 lvm  /var
│           └─ubuntu--vg-SWAP     252:2    0    4G  0 lvm  [SWAP]
│               └─ubuntu--vg-home 252:3    0    4G  0 lvm  /home
└─sr0                              11:0     1 1024M  0 rom
sysadmin@ubuntu01:~$

```

Comando **lsblk** el cual muestra todos puntos de montaje de cada uno de los volúmenes creados en la instalación de **ubuntu01**



```

[sysadmin@vbox ~]$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                  8:0      0   25G  0 disk
├─sda1                              8:1      0    2G  0 part /boot
├─sda2                              8:2      0   23G  0 part
│   └─cs_vbox-root                253:0    0   10G  0 lvm  /
│       └─cs_vbox-swap            253:1    0    4G  0 lvm  [SWAP]
│           └─cs_vbox-var          253:2    0    5G  0 lvm  /var
│               └─cs_vbox-home     253:3    0    4G  0 lvm  /home
└─sr0                              11:0     1 1024M  0 rom
[sysadmin@vbox ~]$ _

```

Comando **lsblk** el cual muestra todos puntos de montaje de cada uno de los volúmenes creados en la instalación de **centos01**

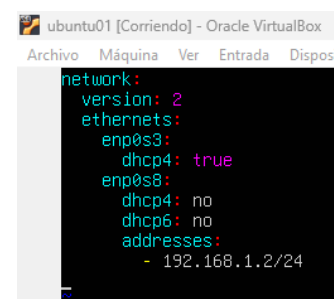
NOTA: A partir de este punto, las tareas estarán en su mayoría, descriptas en el archivo readme.md <https://github.com/aslxpintososa/obligatorioaslx2025/blob/main/README.md>

Configuración de Redes

Se procede a la configuración de la red interna la cual utilizara el bastión para comunicarse con las dos máquinas de LINUX

Tal como es mencionado en la sección anterior, centos tendrá la 192.168.1.3/24. Ubuntu01 la 192.168.1.2 y el bastion utilizara la red 192.168.1.1

Se comprueba que haya conectividad entre ellas luego de la configuración.



```

network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      dhcp6: no
  addresses:
    - 192.168.1.2/24


```


Configuración de la clave GPG en GITHUB (bastion)

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys



aslx.pintos.sosa@gmail.com
SHA256: ZrJhMcY9BKYffmRE325hfPhhJYM200qr7Nlsxh1i8Wc
Added on Aug 7, 2025
Never used — Read/write

Se instalan las siguientes versiones de GIT y ANSIBLE en el BASTION:

```
GIT
dnf install git (version git-core-2.47.3-1.el9.x86_64)
ANSIBLE
dnf install ansible core (version 1:2.14.18-1rl9)
```

Repositorio para utilizar en GIT HUB

Con motivo de tener el trabajo obligatorio lo más organizado posible, se creó una cuenta de GITHUB específica para la realización de la tarea y en la cual existirá el repositorio llamado: **obligaorioaslx2025** el cual es parte de la cuenta de GITHUB **aslxpintossosa**

Se procede a realizar la clonación de este a la maquina BASTION

1

General

Owner *

aslxpintossosa

Repository name *

obligatorioaslx2025

obligatorioaslx2025 is available.

Great repository names are short and memorable. How about **ideal-parakeet**?

Description

Repositorio para la realización del trabajo obligatorio de Taller ASLX 2025 Pintos-Sosa

87 / 350 characters

2

Configuration

Choose visibility *

Choose who can see and commit to this repository

Public

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

On

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license

Licenses explain how others can use your code. [About licenses](#)

Creative Commons Zero v1.0 Universal

SECCION IV - Creación de estructura del repositorio y Pruebas requeridas

Para satisfacer y tener organizado nuestro repositorio, se procedió a crear una estructura donde estarán configuraciones, archivos de variables y playbooks.

- Se crean los directorios collections, inventories (dentro de este, groups_vars para variables), playbooks y templates
- Se crea el archivo de configuración ansible.cfg (Para referenciar del inventario)
- Se crea archivo de inventario

```
[centos]
centos01      ansible_host=192.168.1.3      ansible_user=sysadmin

[ubuntu]
ubuntu01      ansible_host=192.168.1.2      ansible_user=sysadmin

[linux:children]
centos
ubuntu

[fileserver]
centos01
```

Prueba de Conexión Exitosa de ANSIBLE a maquinas HOSTS

```
[sysadmin@bastion01 inventories]$ ansible-inventory -i inventory.ini --list
{
  "_meta": {
    "hostvars": {
      "centos01": {
        "ansible_host": "192.168.1.3",
        "ansible_user": "sysadmin"
      },
      "ubuntu01": {
        "ansible_host": "192.168.1.2",
        "ansible_user": "sysadmin"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped",
      "linux",
      "fileserver"
    ]
  },
  "centos": {
    "hosts": [
      "centos01"
    ]
  },
  "fileserver": {
    "hosts": [
      "centos01"
    ]
  },
  "linux": {
    "children": [
      "centos",
      "ubuntu"
    ]
  },
  "ubuntu": {
    "hosts": [
      "ubuntu01"
    ]
  }
}
```

En la imagen se muestra el contenido del archivo de configuración de inventario utilizando el comando

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

En el archivo de inventario se puede observar las variables de host con IPs y las variables del usuario para conexiones SSH.

En la siguiente imagen, se muestra la prueba de conexión exitosa a dichos HOSTS.

```

[sysadmin@bastion01 inventories]$ ansible all -i inventory.ini -m ping
centos01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
[sysadmin@bastion01 inventories]$

```

Ejecución de comandos AD-HOC

Se procede a ejecutar los siguientes comandos AD-HOC

1 – Listar todos los usuarios del servidor Ubuntu01

ansible ubuntu01 -i inventories/inventory.ini -m shell -a "cut -d: -f1 /etc/passwd"

```

[sysadmin@bastion01 obligatorioaslx2025]$ ansible ubuntu01 -i inventories/inventory.ini -m shell -a "cut -d: -f1 /etc/passwd"
ubuntu01 | CHANGED | rc=0 >>
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
Lapt
nobody
systemd-network
systemd-timesync
dhcpcd
messagebus
systemd-resolve
pollinate
polkitd
syslog
uidd
tcpdump
tss
landscape
fwupd-refresh
usbmux
sshd
sysadmin

```

2 – Mostrar el uso de memoria en todos los servidores

ansible linux -i inventories/inventory.ini -m shell -a "free -h"

```

[sysadmin@bastion01 obligatorioaslx2025]$ ansible linux -i inventories/inventory.ini -m shell -a "free -h"
centos01 | CHANGED | rc=0 >>
total        used        free      shared  buff/cache   available
Mem:         1.7Gi         337Mi       1.2Gi       4.0Mi       260Mi       1.3Gi
Swap:        4.0Gi           0B         4.0Gi
ubuntu01 | CHANGED | rc=0 >>
total        used        free      shared  buff/cache   available
Mem:         1.9Gi         450Mi       1.4Gi       1.1Mi       239Mi       1.5Gi
Swap:        4.0Gi           0B         4.0Gi

```

3 – Verificar que Chrony este instalado y funcionando en servidores centos ansible centos01 -i inventories/inventory.ini -m shell -a "systemctl status chronyd"

```
[sysadmin@bastion01 obligatorioaslx2025]$ ansible centos01 -i inventories/inventory.ini -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 Thu 2025-08-07 20:04:10 EDT; 3h 22min ago
     Docs: man:chronyd(8)
           man:chrony.conf(5)
   Process: 677 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, status=0/SUCCESS)
  Main PID: 682 (chronyd)
    Tasks: 1 (limit: 10643)
   Memory: 4.5M
      CPU: 1.173s
   CGroup: /system.slice/chronyd.service
           └─682 /usr/sbin/chronyd -F 2

Aug 07 20:04:10 centos01 chronyd[682]: Frequency -10.121 +/- 0.867 ppm read from /var/lib/chrony/drift
Aug 07 20:04:10 centos01 chronyd[682]: Loaded seccomp filter (level 2)
Aug 07 20:04:10 centos01 systemd[1]: Started NTP client/server.
Aug 07 20:04:31 centos01 chronyd[682]: Selected source 190.64.134.53 (2.centos.pool.ntp.org)
Aug 07 20:04:31 centos01 chronyd[682]: System clock wrong by 23.157097 seconds
Aug 07 20:04:54 centos01 chronyd[682]: System clock was stepped by 23.157097 seconds
Aug 07 20:04:54 centos01 chronyd[682]: System clock TAI offset set to 37 seconds
Aug 07 20:04:56 centos01 chronyd[682]: Selected source 200.40.115.74 (2.centos.pool.ntp.org)
Aug 07 20:06:00 centos01 chronyd[682]: Selected source 164.73.232.34 (2.centos.pool.ntp.org)
Aug 07 20:06:01 centos01 chronyd[682]: Selected source 200.40.115.74 (2.centos.pool.ntp.org)
```

IMPORTANTE: Para la ejecución de los playbooks a continuación se deben instalar dos colecciones adicionales a ansible.core

ansible.posix (Version 1.5.4)
community.general (Version 10.7.2)

```
---
collections:
  - name: ansible.posix
    version: 1.5.4
  - name: community.general
    version: 10.7.2
~
~
```

El repositorio incluye el archivo
/collections/requirements.yaml

De utilizar este archivo se debe ejecutar el comando
ansible-galaxy

Ej en CentOS

ansible-galaxy install -r collections/requirements.yaml

```
[sysadmin@bastion01 obligatorioaslx2025]$ ansible-galaxy install -r collections/requirements.yaml
Starting galaxy collection install process
Process install dependency map
Starting collection install process
'ansible.posix:2.1.0' is already installed, skipping.
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifact/tmp/ansible-local-98792m4r_yu_/tmpfl06syll/community-general-10.7.2-6a3gpgy4
Installing 'community.general:10.7.2' to '/home/sysadmin/.ansible/collections/ansible_collections/
```

SECCION V - Creación de Playbooks

```
---
paquete: nfs-utils
puerto: 2049
servicio_firewall: nfs
```

Se definieron las siguientes variables para los playbooks a utilizar, presentes en `/inventories/group_vars/variables.yaml`

Las mismas se ven en los playbooks creados los cuales se muestran a continuación.

Además para las funcionalidades que no se encuentran en el `asnible-core`, se deben definir los requerimientos en el archivo `collections/requirements.yaml` como se indica en la sección anterior

Recordar ejecutar:

ansible-galaxy install -r collections/requirements.yaml

```
---
collections:
  - name: ansible.posix
    version: 1.5.4
```

```
[sysadmin@bastion01 obligatorioaslx2025]$ ansible-galaxy install -r collections/requirements.yaml
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/ansible-local-8642298eghx1/tmpesxdw2gz/ansible-posix-2.1.0-_3bd5u2_
Installing 'ansible.posix:2.1.0' to '/home/sysadmin/.ansible/collections/ansible_collections/ansible/ansible.posix:2.1.0'
ansible.posix:2.1.0 was installed successfully
```

Testeo de Puertos abiertos para NFS (y Servicio)

```
[sysadmin@bastion01 obligatorioaslx2025]$ vi playbooks/r
[sysadmin@bastion01 obligatorioaslx2025]$ ansible-playbook [root@centos01 sysadmin]# firewall-cmd --zone=public --list-services
BECOME password: cockpit dhcpv6-client ssh
[WARNING]: While constructing a mapping from /home/sysadmin/.ansible/collections/ansible_collections/ansible/ansible.posix:2.1.0 to '/home/sysadmin/.ansible/collections/ansible_collections/ansible/ansible.posix:2.1.0'
(permanent). Using last defined value only.
[WARNING]: Collection ansible.posix does not support Ansible 2.9.10
PLAY [Playbook para instalacion de FileServer (CentOS)] [root@centos01 sysadmin]# firewall-cmd --zone=public --list-ports
2049/tcp
TASK [Gathering Facts] *****
ok: [centos01]
TASK [Instalacion de Servidor NFS] *****
ok: [centos01]
TASK [Asegurar que el servicio este iniciado y habilitado] *****
ok: [centos01]
TASK [Asegurar que el puerto del Firewall este abierto e] *****
changed: [centos01]
TASK [Habilitar tambien en el firewall, NFS como servicio] *****
changed: [centos01]
PLAY RECAP *****
centos01 : ok=5 changed=2 unreachable=0 failed=0 skipped=0 rescued=0
```

Testeo de Ejecución nfs shared

```
[sysadmin@bastion01 obligatorioaslx2025]$ ansible-playbook -i inventories/inventory.ini playbooks/nfs_setup.yaml -K
BECOME password:
[WARNING]: While constructing a mapping from /home/sysadmin/obligatorioaslx2025/playbooks/nfs_setup.yaml, line 21, column 7, found a duplicate dict key
(permanent). Using last defined value only.
[WARNING]: Collection ansible.posix does not support Ansible version 2.14.18

PLAY [Playbook para instalacion de FileServer (CentOS)] *****

TASK [Gathering Facts] *****
ok: [centos01]

TASK [Instalacion de Servidor NFS] *****
ok: [centos01]

TASK [Asegurar que el servicio este iniciado y habilitado] *****
ok: [centos01]

TASK [Asegurar que el puerto del Firewall este abierto en 2049 (NFS)] *****
ok: [centos01]

TASK [Habilitar tambien en el firewall, NFS como servicio] *****
ok: [centos01]

TASK [Existe el directorio /var/nfs_shared] *****
ok: [centos01]

TASK [Mostrar Resultado] *****
skipping: [centos01]

TASK [Compartir Carpeta nfs_shared] *****
ok: [centos01]

PLAY RECAP *****
centos01 : ok=7  changed=0  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
```

NFS Instalado y corriendo:

```
[root@centos01 etc]# systemctl status nfs-utils
● nfs-utils.service - NFS server and client services
   Loaded: loaded (/usr/lib/systemd/system/nfs-utils.service; static)
   Active: active (exited) since Fri 2025-08-08 02:42:59 EDT; 3h 1min ago
   Main PID: 14916 (code=exited, status=0/SUCCESS)
   CPU: 9ms

Aug 08 02:42:59 centos01 systemd[1]: Starting NFS server and client services..
Aug 08 02:42:59 centos01 systemd[1]: Finished NFS server and client services.
[root@centos01 etc]# _
```

Firewall con servicio nfs y puerto 2049 abierto

```
[root@centos01 etc]# firewall-cmd --zone=public --list-ports
2049/tcp
[root@centos01 etc]# firewall-cmd --zone=public --list-services
cockpit dhcpv6-client nfs ssh
[root@centos01 etc]#
```

Existencia de directorio nfs_shared

```
[root@centos01 etc]# cat exports
/var/nfs_shared *(rw, sync, no_subtree_check)
[root@centos01 etc]# cd /var/nfs_shared/
```

Testeo de ejecución de hardening.yaml

```
[sysadmin@bastion01 obligatorioaslx2025]$ ansible-playbook -i inventories/inventory.ini playbooks/hardening.yaml -K
BECOME password:
[WARNING]: Collection community.general does not support Ansible version 2.14.18

PLAY [ubuntu] *****

TASK [Gathering Facts] *****
ok: [ubuntu01]

TASK [Actualizar Paquetes] *****
ok: [ubuntu01]

TASK [Instalar Fail2BAN] *****
changed: [ubuntu01]

TASK [Iniciar y habilitar Fail2BAN] *****
ok: [ubuntu01]

TASK [No Permitir Login Root] *****
ok: [ubuntu01]

TASK [No Permitir iniciar sesion con Password] *****
ok: [ubuntu01]

TASK [Asegurarse que UFW Este habilitado y ejecutandose bloqueando trafico incoming] *****
ok: [ubuntu01]

TASK [Habilitar Open SSH] *****

PLAY RECAP *****
ubuntu01 : ok=8  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

UFW Habilitado y FAIL2BAN instalado y habilitado

```
root@ubuntu01:/home/sysadmin# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-08-10 03:33:26 UTC; 54s ago
     Docs: man:fail2ban(1)
   Main PID: 7420 (fail2ban-server)
      Tasks: 5 (limit: 2268)
     Memory: 22.1M (peak: 22.4M)
        CPU: 1.696s
    CGroup: /system.slice/fail2ban.service
            └─7420 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Aug 10 03:33:26 ubuntu01 systemd[1]: Started fail2ban.service - Fail2Ban Service.
Aug 10 03:33:27 ubuntu01 fail2ban-server[7420]: 2025-08-10 03:33:27,516 fail2ban.configreader [7420]: WARN
Aug 10 03:33:29 ubuntu01 fail2ban-server[7420]: Server ready
lines 1-14/14 (END)
^C
root@ubuntu01:/home/sysadmin# systemctl status ufw
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/usr/lib/systemd/system/ufw.service; enabled; preset: enabled)
   Active: active (exited) since Sun 2025-08-10 03:14:24 UTC; 21min ago
     Docs: man:ufw(8)
   Main PID: 527 (code=exited, status=0/SUCCESS)
        CPU: 414ms

Aug 10 03:14:19 ubuntu01 systemd[1]: Starting ufw.service - Uncomplicated firewall...
Aug 10 03:14:24 ubuntu01 systemd[1]: Finished ufw.service - Uncomplicated firewall.
root@ubuntu01:/home/sysadmin# _
```

Configuración de SSHd

```
# ForceCommand cvs server
PermitRootLogin no
PasswordAuthentication no
"/etc/ssh/ssh_config" 133L, 3562B
```

SECCION VI - PREGUNTAS:

Las siguientes preguntas se encuentran respondidas en un archivo dentro del repositorio llamado “ansible_basics.txt”

1. ¿Qué es Ansible? Mencione dos actividades que se puedan hacer con Ansible

Ansible es una herramienta de automatización de código abierto que se utiliza para reducir complejidad y ser ejecutada en varios tipos de ambientes.

Esta herramienta puede automatizar tareas de todo tipo en un sistema.

Sus características principales:

- No tiene Agente (No es necesario tener clientes instalados para su ejecución)
- Simple
- Escalable
- Idempotencia y Flexibilidad (Si el sistema está en el estado deseado, este no será alterado)

2. ¿Qué es un Playbook de Ansible?

Los playbooks son un sistema de Ansible que sirven para la configuración, implementación y aplicación compleja de tareas en sistemas.

Estas son repetibles, reutilizables y sencillas.

características principales:

- Declarar configuraciones
- Orquestrar pasos de procesos de manera ordenada
- Ejecutar tareas

3. ¿Qué información contiene un inventario de Ansible?

Un archivo de inventario en ansible contiene información principalmente de los hosts que ansible maneja.

Contiene también agrupaciones o grupos, donde los host y diferentes sistemas son agrupados por algún motivo en particular para poder luego correr playbooks específicos para esos grupos.

Estos grupos, pueden a la vez ser parte de otros grupos más grandes.

4. Explique que es un módulo de Ansible y dé un ejemplo.

Un módulo es un script que sirve para ejecutar una tarea específica, utilizado para comandos ad-hoc o playbooks, los módulos de ansible tienen una amplia gama de tareas que pueden realizar.

Existen también ampliaciones a los módulos “core” que trae incluidos ansibles los cuales pueden ser descargados adicionalmente.

5. ¿Qué ventajas tiene Ansible sobre otros métodos de automatización?

Si bien es objetivo en que puede ser mejor comparado con otros métodos, hay algunas cosas que se destacan como que no necesita cliente para funcionar, es escalable, es sencillo de configurar y ejecutar

SECCION VII - RESULTADOS

Para las conclusiones, quisiéramos destacar algunos retos y enseñanzas del taller.

Primeramente haber podido crear un modelo totalmente funcional de la solución requerida por el trabajo, el cual funciona mayoritariamente sin ningún inconveniente.

Pudimos tener una muy buena exposición, mediante ansible, a los automatismos que rodean al mundo de los sistemas operativos. En el caso de uno de los integrantes del equipo, en materia laboral, había estado brevemente trabajando con ansible, pero nunca entendiendo de la manera en la que se pudo ahondar en el taller, en el caso del otro integrante, la experiencia fue buena y se pudo comprender las básicas del sistema así como el concepto de automatizar tareas.

Los retos más importantes estuvieron a la hora de diagramar como abordar las tareas de cada uno de los playbooks, organizarlas y darles cierto orden para que no hubiese conflicto entre ellas además de buscar documentación relacionada con las mismas.

DIFICULTADES ENCONTRADAS

CONFIGURACION DE IP INCORRECTA

Al comienzo de la elaboración del trabajo, una dirección IP estaba con un error de tipeo el cual retraso bastante el trabajo hasta ser encontrado.

USUARIO NOBODY

En la creación del playbook para CentOS, cometimos el error de utilizar “nogroup” en vez de nobody en el campo de grupo para la creación del directorio, esto llevo al playbook a no funcionar correctamente.

DENEGACION TOTAL DE PUERTO 22 – UBUNTU

En la ejecución del playbook de hardening, se logró configurar el firewall de Ubuntu y que este denegara todas las conexiones entrantes, pero la sección que habilitaba el puerto 22 para el servicio SSH tenía un error de tipeo y fallo, esto hizo que perdiéramos la habilidad de arreglarlo por medio de ansible, tuvimos que habilitar manualmente el puerto 22 en el servidor y correr el playbook una vez más.

ANEXO I – Uso de Inteligencia Artificial Generativa

Para la realización de este trabajo, el uso de **IAG** fue utilizado EXCLUSIVAMENTE para la creación de gráficos específicos o conjunto de comandos básicos con motivo de verificación.

A continuación, se detalla la lista de prompts utilizados.

IAG 1 – Verificar comando de búsqueda de usuarios

<https://chatgpt.com/>

Prompt: Mostrar comando de búsqueda de usuarios dentro del archivo passwd listando solo la primera columna de este.

Resultado resumido: **cut -d: -f1 /etc/passwd**

IAG 2 – Reinicio de servicio de Firewallld para creación de handler

<https://chatgpt.com/>

Prompt: Ejemplo de handler para recargar firewallld

```
handlers:
  - name: Recargar firewallld
    ansible.builtin.command: firewall-cmd --reload
```

ANEXO II – Playbooks

nfs_setup.yaml

```
---
- name: Playbook para instalacion de FileServer (CentOS)
  hosts: fileserver #Se indica el grupo que contiene al fileserver en el inventario
  become: yes #Necesario para elevar el privilegio

  tasks:

    - name: Instalacion de Servidor NFS
      ansible.builtin.dnf: #el modulo dnf se encuentra en ansible core
        name: "{{ paquete }}" #Variable paquete la cual esta definida en /inventories/group_vars
        state: present #Se asegura que este presente, en caso contrario, lo instala

    - name: Asegurar que el servicio este iniciado y habilitado
      ansible.builtin.systemd_service:
        name: "{{ servicio }}"
        enabled: true
        state: started

    - name: Asegurar que el puerto del Firewall este abierto en 2049 (NFS)
      ansible.posix.firewalld:
        port: "{{ puerto }}" #La variable abre el puerto 2049, utilizado por NFS
        permanent: true
        state: enabled
        permanent: true
        immediate: true
        notify: Recargar_Firewall

    - name: Habilitar tambien en el firewall, NFS como servicio
      ansible.posix.firewalld:
        service: "{{ servicio_firewall }}" #La variable definida, corresponde a nfs
        state: enabled
        permanent: true
        immediate: true
        notify: Recargar_Firewall

    - name: Existe el directorio /var/nfs_shared
      ansible.builtin.file:
        path: /var/nfs_shared
        state: directory
        owner: nobody
        group: nobody
        mode: '0777'
        register: resultado #Guarda el resultado para ser mostrado luego por el modulo a continuacion

    - name: Mostrar Resultado
      ansible.builtin.debug:
        msg: "{{ resultado.diff.after }}"
        when: resultado.changed

    - name: Compartir Carpeta nfs_shared
      ansible.builtin.lineinfile:
        path: /etc/exports
        line: '/var/nfs_shared *(rw, sync, no_subtree_check)'
        notify: exportarfs

  handlers:

    - name: Recargar_Firewall
      ansible.builtin.command: firewall-cmd --reload

    - name: exportarfs
      ansible.builtin.command: exportfs -r
```

hardening.yml

```
---
- hosts: ubuntu
  become: yes

  tasks:

    - name: Actualizar Paquetes
      ansible.builtin.apt:
        name: "*"
        state: latest
        update_cache: true
      notify: Reinicio_Sistema

    - name: Instalar Fail2BAN
      ansible.builtin.apt:
        name: fail2ban
        state: present

    - name: Iniciar y habilitar Fail2BAN
      ansible.builtin.service:
        name: fail2ban
        enabled: yes
        state: started

    - name: No Permitir Login Root
      ansible.builtin.lineinfile:
        path: /etc/ssh/sshd_config
        insertafter: '^PermitRootLogin'
        line: PermitRootLogin no
      notify: Reinicio_Daemon_SSH

    - name: No Permitir iniciar sesion con Password
      ansible.builtin.lineinfile:
        path: /etc/ssh/sshd_config
        insertafter: '^PasswordAuthentication'
        line: PasswordAuthentication no
      notify: Reinicio_Daemon_SSH

    - name: Asegurarse que UFW Este habilitado y ejecutandose bloqueando trafico incoming
      community.general.ufw:
        state: enabled
        policy: reject
        direction: incoming
      notify: Recargar_UFW

    - name: Habilitar Open SSH
      community.general.ufw:
        rule: allow
        port: 22
        proto: tcp
      notify: Recargar_UFW

  handlers:

    - name: Reinicio_Sistema
      ansible.builtin.reboot:

    - name: Reinicio_Daemon_SSH
      ansible.builtin.systemd_service:
        name: ssh
        state: restarted

    - name: Recargar_UFW
      ansible.builtin.command: ufw reload
```

BIBLIOGRAFIA

Apartado Sobre Máquinas Virtuales

Trabajo Obligatorio CWyU – Marcelo Sosa y Hernán Pintos

Trabajo Obligatorio ASLX – Marcelo Sosa y Hernán Pintos

Creación de Inventario de ANSIBLE

https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html

```
[targets]
localhost      ansible_connection=local
other1.example.com  ansible_connection=ssh      ansible_user=myuser
other2.example.com  ansible_connection=ssh      ansible_user=myotheruser
```

Que es Ansible?

https://docs.ansible.com/ansible/latest/getting_started/introduction.html

<https://www.godaddy.com/resources/es/crearweb/que-es-ansible>

Como insertas un string al comienzo de un archivo

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/lineinfile_module.html

Comandos ANSIBLE para UFW

https://docs.ansible.com/ansible/latest/collections/community/general/ufw_module.html

Documentación Fail2BAN

<https://blog.swmansion.com/limiting-failed-ssh-login-attempts-with-fail2ban-7da15a2313b>