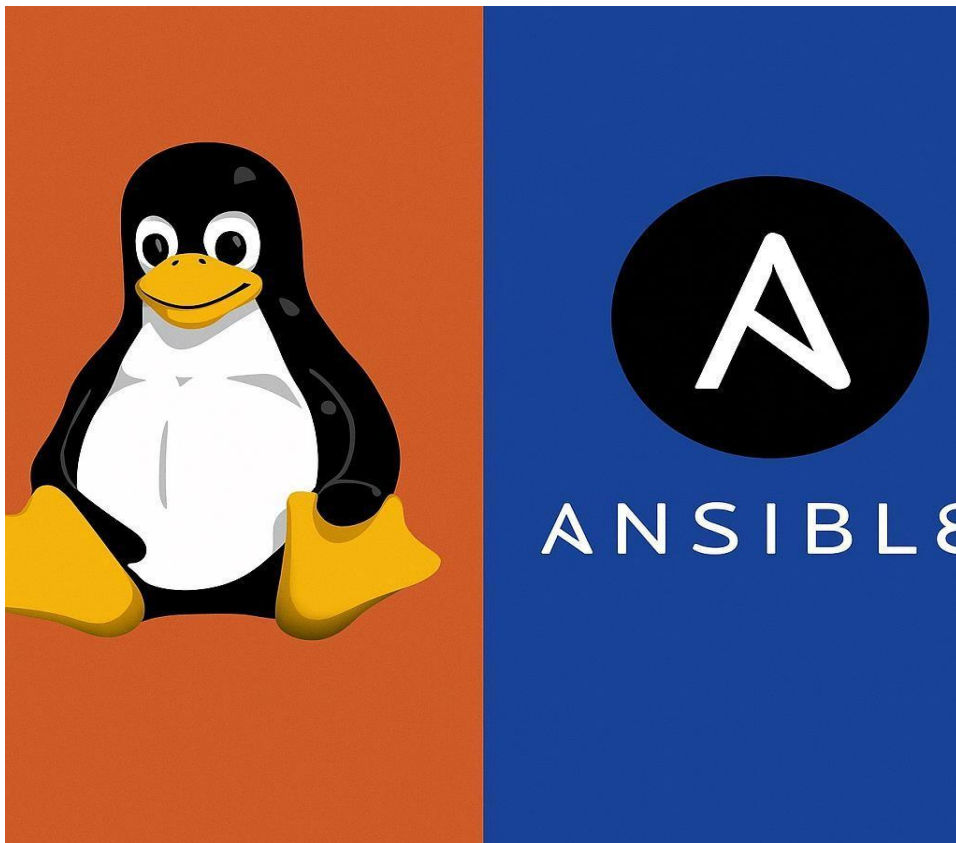


Universidad ORT Uruguay
Facultad de Ingeniería: Escuela de Tecnología

Obligatorio del Taller de servidores Linux



Autores:
Fabian Ferreira
N.º Estudiante 187374

Juan Recalde
N.º Estudiante 334170

Profesor: Enrique Verdes - Grupo N3.5A

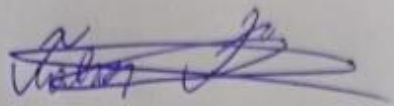
Fecha de la entrega: 12/08/2025

Declaración de Autoría

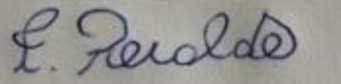
Nosotros, Juan Recalde / Fabian Ferreira, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano.

Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizamos Obligatorio de Taller de servidores Linux;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue construido por otros, y qué fue construido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



FABIAN FERREIRA



Juan Recalde

Contenido

Declaración de Autoría	2
Objetivo	4
Desafíos encontrados	4
Tarea 1: Instalar servidores.....	5
\$ssh-keygen	7
\$cat ~/.ssh/id_rsa.pub	7
\$ssh-copy-id sysadmin@192.168.1.100	7
\$ssh-copy-id sysadmin@192.168.1.101	7
Instalar GIT:	7
\$ls -a	8
\$ansible-inventory -i inventories/inventory.ini --graph	9
Tarea 2: Configurar un archivo de inventario Ansible	10
Configuración y detalles extras:	10
\$ansible-inventory -i inventory.ini --list	11
\$ansible all -i inventory.ini -m ping	12
Tarea 3: Ejecutar comandos ad-hoc	14
\$ansible ubuntu -m command -a "cut -d: -f1 /etc/passwd"	14
\$ansible-vault create secret.yml.....	16
Tarea 4: Crear y ejecutar playbook de Ansible	17
Tarea 5: Responder las siguientes preguntas.....	19
¿Qué es Ansible? Mencione dos actividades que se puedan hacer con Ansible	19
2. ¿Qué es un playbook de Ansible?	19
3. ¿Qué información contiene un inventario de Ansible?	19
4. Explique que es un módulo de Ansible y dé un ejemplo.	20
Ejemplos:	20
5. ¿Qué ventajas tiene Ansible sobre otros métodos de automatización?	20
Bibliografía:	21
Módulos utilizados para el playbook de CentOS	21
Módulos utilizados para el playbook de Ubuntu	21
Prompts utilizados en IA:	22

Objetivo

El objetivo básico del obligatorio es aplicar los conocimientos básicos de Ansible sobre dos distribuciones Linux: Centos Stream 9 y Ubuntu 24.04.

Desafíos encontrados

Uno de los principales desafíos fue lograr que los Playbooks quedaran completamente funcionales y sin errores, especialmente al aplicar configuraciones sobre ciertos servicios y aplicaciones. La instalación y ajuste de componentes como, fail2ban y la configuración de SSH, provocó fallos puntuales en la ejecución. Estos problemas exigieron revisar el orden de las tareas, adaptar parámetros a cada sistema operativo y validar que las configuraciones aplicadas se mantuvieran idempotentes.

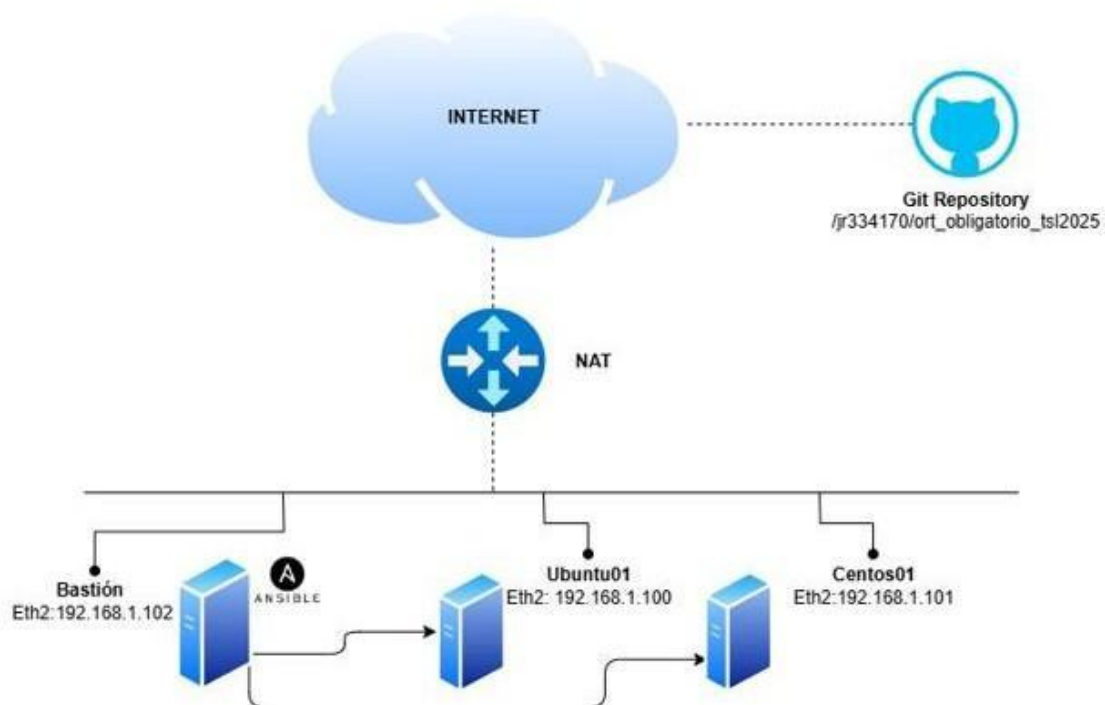
En un principio habíamos decidido utilizar módulos CORE para solucionar los requisitos del obligatorio, pero al final y luego de investigar, utilizamos módulos de terceros, lo que nos agrega mejoras notables en el código/mejoras y muchísimo más corto, sin tener que manejar/parsear datos/variables.

A esto se sumó el aprendizaje acelerado que tuvimos que realizar en un corto período de tiempo, incorporando no solo conocimientos técnicos de Ansible y administración de servicios, sino también el manejo colaborativo de Git para versionar y sincronizar el trabajo.

Tarea 1: Instalar servidores

Repositorio en GitHub público: https://github.com/jr334170/ort_obligatorio_tsl2025

Diagrama ejemplo, solución realizada.



Ubuntu01:

```
sysadmin@ubuntu01:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           197M  1,1M  196M   1% /run
/dev/sda3       9,8G  4,1G  5,3G  44% /
tmpfs           985M    0  985M   0% /dev/shm
tmpfs           5,0M    0   5,0M   0% /run/lock
/dev/sda4       4,9G  394M  4,2G   9% /var
/dev/sda2       974M  100M  807M  11% /boot
/dev/sda5       3,9G   24K  3,7G   1% /swap
tmpfs           197M   12K  197M   1% /run/user/1000
sysadmin@ubuntu01:~$
```

```
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 08:00:27:4f:93:63 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 74831sec preferred_lft 74831sec
    inet6 fd00::83a:eb49:404c:a1ee/64 scope global temporary dynamic
        valid_lft 85038sec preferred_lft 13038sec
    inet6 fd00::a00:27ff:fe4f:9363/64 scope global dynamic mngtmpaddr
        valid_lft 85038sec preferred_lft 13038sec
    inet6 fe80::a00:27ff:fe4f:9363/64 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 08:00:27:15:99:b1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::9ca8:734b:6186:158e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Centos01:

```
[root@vbox sysadmin]# df -h
S.ficheros          Tamaño Usados  Disp Uso% Montado en
devtmpfs             4,0M      0    4,0M   0% /dev
tmpfs                886M      0    886M   0% /dev/shm
tmpfs                355M    5,0M   350M   2% /run
/dev/mapper/cs_vbox-root 9,3G    2,0G    7,3G  22% /
/dev/sda1            889M    327M   563M  37% /boot
/dev/mapper/cs_vbox-var 4,6G    138M    4,5G   3% /var
tmpfs                178M      0    178M   0% /run/user/1000
[root@vbox sysadmin]# hostname
Centos01
```

```
[sysadmin@Centos01 ~]$ 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 08:00:27:26:5f:2f 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 57153sec preferred_lft 57153sec
    inet6 fd00::a00:27ff:fe4f:9363/64 scope global dynamic noprefixroute
        valid_lft 85952sec preferred_lft 13952sec
    inet6 fe80::a00:27ff:fe4f:9363/64 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:e3:32:ca brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.101/24 brd 192.168.1.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::c3:ca36:36:368a/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```


Generamos la clave pública:

\$ssh-keygen

Para ver la clave pública:

\$cat ~/.ssh/id_rsa.pub

Copiamos las claves públicas desde el Bastión a cada host en authorized_keys:

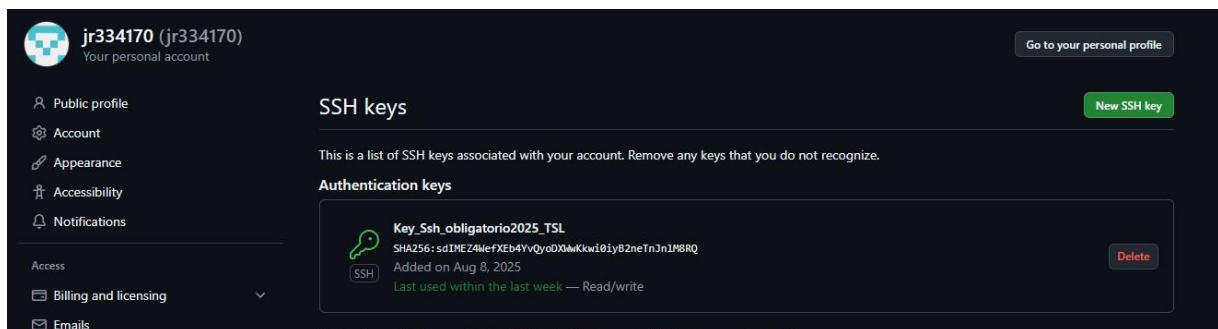
\$ssh-copy-id sysadmin@192.168.1.100

\$ssh-copy-id sysadmin@192.168.1.101

Instalar GIT:

\$sudo dnf install git

SSH Key en GitHub, cuenta utilizada para alojar repositorio.



Bastión:

```
[sysadmin@vbox ~]$ df -h
S.ficheros          Tamaño Usados  Disp Uso% Montado en
devtmpfs            -  4.0M      0  4.0M   0% /dev

[sysadmin@Bastion ~]$ 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:e3:fc:0b 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 65225sec preferred_lft 65225sec
   inet6 fe80::a00:37ff:fe01:300b/64 scope global dynamic noprefixroute
       valid_lft 86187sec preferred_lft 14187sec
   inet6 fe80::a00:37ff:fe01:300b/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 08:00:27:79:fa:fc brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.102/24 brd 192.168.1.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
   inet6 fe80::38a:13ff:fe04:8ac7/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

\$git config --list

```
[sysadmin@vbox ort_obligatorio_tsl2025]$ git config --list
user.name=Recalde Ferreira
user.email=juanreca99@hotmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=git@github.com:jr334170/ort_obligatorio_tsl2025.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
```

\$ls -a

Revisamos lo que clonó desde Git con la SSH KEY

```
[sysadmin@vbox ort_obligatorio_tsl2025]$ ls -a
.  ..  .git  LICENSE  README.md
[sysadmin@vbox ort_obligatorio_tsl2025]$
```

Instalamos ansible-core en Bastión.

\$sudo dnf install -y ansible-core

\$ansible --version

```
[sysadmin@vbox ort_obligatorio_tsl2025]$ ansible --version
ansible [core 2.14.18]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/sysadmin/.ansible/plugins/modules', '/usr/share/ansible']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /home/sysadmin/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.21 (main, Feb 10 2025, 00:00:00) [GCC 11.5.0 20240719 (Red Hat 11.5.0-5)]
  jinja version = 3.1.2
  libyaml = True
```

Instalamos ansible-galaxy, para la ejecución de módulos.

\$ansible-galaxy install -r collections/requirements.yaml

```
[sysadmin@vbox ort_obligatorio_tsl2025]$ 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-posix-2.1.0.tar.gz to /home/sysadmin/.ansible/tmp/ansible-local-319137j7j34a0/tmpx6h2zj8_/ansible-posix-2.1.0-5p7pyo3m
Installing 'ansible.posix:2.1.0' to '/home/sysadmin/.ansible/collections/ansible_collections/ansible/posix'
ansible.posix:2.1.0 was installed successfully
```

Instalación de modulo requerido para Playbook Ubuntu UFW

\$ansible-galaxy collection install community.general

\$ansible-inventory -i inventories/inventory.ini --graph

```
[sysadmin@vbox ort_obligatorio_ts12025]$ ansible-inventory -i inventories/inventory.ini --graph
@all:
  |--@ungrouped:
  |--@linux:
  | |--@centos:
  | | |--centos01
  | |--@ubuntu:
  | | |--ubuntu01
  |--@webserver:
  | |--centos01
```

Tarea 2: Configurar un archivo de inventario Ansible

Creemos el archivo inventory.ini con este formato

```
[sysadmin@vbox ort_obligatorio_tsl2025]$ cat inventories/inventory.ini
[centos]
centos01      ansible_host=192.168.1.2

[ubuntu]
ubuntu01      ansible_host=192.168.1.3

[linux:children]
centos
ubuntu

[fileserver]
centos01
```

\$cat inventories/inventory.ini

```
[sysadmin@vbox inventories]$ cat inventory.ini
[centos]
centos01      ansible_host=192.168.1.101

[ubuntu]
ubuntu01      ansible_host=192.168.1.100

[linux:children]
centos
ubuntu

[webserver]
centos
[sysadmin@vbox inventories]$ tree
.
├── hosts.ini
└── inventory.ini

0 directories, 2 files
```

Configuración y detalles extras:

Configuramos un directorio **group_vars**, para almacenar las variables en este caso de usuario de acceso de ansible por cada tipo de grupo.

```
├── inventories
│   └── group_vars
│       ├── centos.yml
│       └── ubuntu.yml
```

\$cat inventories/group_vars/centos.yml

```
[sysadmin@Bastion ort_obligatorio_tsl2025]$ cat inventories/group_vars/centos.yml
ansible user: sysadmin
```

\$cat inventories/group_vars/ubuntu.yml

```
[sysadmin@Bastion ort_obligatorio_tsl2025]$ cat inventories/group_vars/ubuntu.yml
ansible user: sysadmin
```

\$ansible -i inventories/inventory.ini all -m ping

```
[sysadmin@vbox ort_obligatorio_tsl2025]$ ansible -i inventories/inventory.ini all -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"
}
```

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

```
[sysadmin@Bastion inventories]$ ansible-inventory -i inventory.ini --list
{
  "_meta": {
    "hostvars": {
      "centos01": {
        "ansible_host": "192.168.1.101",
        "ansible_user": "sysadmin"
      },
      "ubuntu01": {
        "ansible_host": "192.168.1.100",
        "ansible_user": "sysadmin"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped",
      "linux",
      "webserver"
    ]
  },
  "centos": {
    "hosts": [
      "centos01"
    ]
  },
  "linux": {
    "children": [
      "centos",
      "ubuntu"
    ]
  },
  "ubuntu": {
    "hosts": [
      "ubuntu01"
    ]
  },
  "webserver": {
    "hosts": [
      "centos01"
    ]
  }
}
```

Instalamos ansible module community.general

\$ansible-galaxy collection install community.general

```
[sysadmin@Bastion ort-obligatorio-tel2025]$ ansible-galaxy collection install community.general
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/community-general-11.1.2.tar.gz to /home/sysadmin/.ansible/tmp/ansible-local-4511_no4zgfgz/tmpo0pxxov0/community-general-11.1.2-91t93fxl
Installing 'community.general:11.1.2' to '/home/sysadmin/.ansible/collections/ansible_collections/community/general'
community.general:11.1.2 was installed successfully
```

Prueba de ping ad-hoc

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

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

Tarea 3: Ejecutar comandos ad-hoc

- Listar todos los usuarios en servidor Ubuntu

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

```
[sysadmin@Bastion ort_obligatorio_ts12025]$ ansible ubuntu -m command -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
_apt
nobody
systemd-network
systemd-timesync
dhcpcd
messagebus
systemd-resolve
pollinate
polkitd
syslog
uidd
tcpdump
tss
landscape
fwupd-refresh
usbmux
sshd
sysadmin
dnsmasq
```


- Mostrar el uso de memoria en todos los servidores

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

```
[sysadmin@Bastion ort_obligatorio_tsl2025]$ ansible all -m command -a "free -h"
centos01 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    1,7Gi         404Mi       1,2Gi         4,0Mi        332Mi       1,3Gi
Swap:    3,7Gi           0B        3,7Gi
ubuntu01 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    1,9Gi         329Mi       1,4Gi         1,1Mi        316Mi       1,6Gi
Swap:    1,9Gi           0B        1,9Gi
```

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

\$ansible centos -m shell -a "dnf install -y chrony && systemctl enable --now chronyd" --become --ask-become-pass

```
[sysadmin@Bastion ort_obligatorio_tsl2025]$ ansible centos -m shell -a "dnf install -y chrony && system
ctl enable --now chronyd" --become --ask-become-pass
BECOME password:
centos01 | CHANGED | rc=0 >>
Última comprobación de caducidad de metadatos hecha hace 2:41:31, el vie 08 ago 2025 21:53:00.
El paquete chrony-4.6.1-1.el9.x86_64 ya está instalado.
Dependencias resueltas.

=====
Paquete      Arquitectura  Versión      Repositorio  Tam.
=====
Actualizando:
chrony        x86_64        4.6.1-2.el9  baseos       345 k
=====
Resumen de la transacción
=====
Actualizar 1 Paquete

Tamaño total de la descarga: 345 k
Descargando paquetes:
chrony-4.6.1-2.el9.x86_64.rpm                137 kB/s | 345 kB    00:02
-----
Total                                          103 kB/s | 345 kB    00:03
CentOS Stream 9 - BaseOS                      1.5 MB/s | 1.6 kB    00:00
La llave ha sido importada exitosamente
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando :                               1/1
Ejecutando scriptlet: chrony-4.6.1-2.el9.x86_64 1/2
Actualizando : chrony-4.6.1-2.el9.x86_64        1/2
Ejecutando scriptlet: chrony-4.6.1-2.el9.x86_64 1/2
Ejecutando scriptlet: chrony-4.6.1-1.el9.x86_64 2/2
Limpieza : chrony-4.6.1-1.el9.x86_64            2/2
Ejecutando scriptlet: chrony-4.6.1-1.el9.x86_64 2/2
Verificando : chrony-4.6.1-2.el9.x86_64         1/2
Verificando : chrony-4.6.1-1.el9.x86_64         2/2

Actualizado:
chrony-4.6.1-2.el9.x86_64

¡Listo!Importando llave GPG 0x8483C65D:
ID usuario: "CentOS (CentOS Official Signing Key) <security@centos.org>"
Huella : 99DB 70FA E1D7 CE22 7FB6 4882 05B5 55B3 8483 C65D
Desde : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
```

Agregamos el `--become` y el `--ask-become-pass` para que ejecute en modo sudo y solicite la password antes de ejecutar los comandos.

Adicionalmente, creamos con ANSIBLE-VAULT un archivo cifrado, que contiene la password del usuario que requiere sudo / become, por seguridad la almacenamos cifrada en dicho archivo.

Para crear el archivo secret.yml ejecutamos el siguiente comando:

\$ansible-vault create secret.yml

Y dentro del archivo escribimos: **ansible_become_password: "tupassrootaqui"**

El archivo secret.yml va a estar ubicado sobre la raíz del proyecto.

```
[sysadmin@Bastion ort_obligatorio_tsl2025]$ ls
ansible.cfg  collections  docs  inventories  LICENSE  playbooks  README.md  README.old  secret.yml
```

Cada vez que necesitemos ejecutar un comando que requiera sudo, llamamos al – extra-vars (nos permite agregar otra variable del inventario en la misma línea) y forzamos que busque dentro del archivo @secret.yml (El @ nos sirve para identificar de forma concreta el archivo y no haga una búsqueda de una variable llamada secret).

```
[sysadmin@Bastion ort_obligatorio_tsl2025]$ ansible centos -m shell -a "dnf install -y chrony && systemctl enable
--now chronyd" --become --extra-vars "@secret.yml" --ask-vault-pass
Vault password:
centos01 | CHANGED | rc=0 >>
CentOS Stream 9 - BaseOS                        34 kB/s | 26 kB    00:00
CentOS Stream 9 - AppStream                     44 kB/s | 27 kB    00:00
CentOS Stream 9 - Extras packages               171 kB/s | 30 kB    00:00
El paquete chrony-4.6.1-2.el9.x86_64 ya está instalado.
Dependencias resueltas.
Nada por hacer.
¡Listo!
```

Tarea 4: Crear y ejecutar playbook de Ansible

Crea un playbook (**nfs_setup.yml**) para CentOS que permita obtener el siguiente resultado:

- El servidor NFS esté instalado
- Se asegure que el servicio NFS esté iniciado y funcionando
- El firewall permita conexiones al puerto 2049
- Exista el directorio `/var/nfs_shared`, que pertenece al usuario/grupo `nobody/nobody` y tiene permisos `777`
- El directorio está compartido por NFS.
- Debe haber un handler que actualice relea el archivo `/etc/exports` si este cambia.

**\$ansible-playbook -i inventories/inventory.ini playbooks/nfs_setup.yml
--become --extra-vars "@secret.yml" --ask-vault-pass**

```
[jrecalde@vbox ort_obligatorio_tsl2025]$ ansible-playbook -i inventories/inventory.ini playbooks/nfs_setup.yml --become --extra-vars "@secret.yml" --ask-vault-pass
Vault password:
[WARNING]: Collection ansible.posix does not support Ansible version 2.14.18

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

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

TASK [TAREA#1 Asegurar paquetes requeridos instalados] *****
ok: [centos01]

TASK [TAREA#2 Asegurar firewalld iniciado y habilitado] *****
ok: [centos01]

TASK [TAREA#3 Permitir TCP 2049 en el firewall] *****
ok: [centos01]

TASK [TAREA#4 Permitir UDP 2049 en el firewall] *****
ok: [centos01]

TASK [TAREA#5 Creamos /var/nfs_shared con nobody:nobody y 0777] *****
ok: [centos01]

TASK [TAREA#6 Declaramos export en /etc/exports] *****
ok: [centos01]

TASK [TAREA#7 Aseguramos que el servicio NFS este iniciado y habilitado] *****
ok: [centos01]

PLAY RECAP *****
centos01 : ok=8  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

Crea un playbook (**hardening.yml**) que se va a ejecutar en los servidores Ubuntu, que realice lo siguiente:

- Actualizar todos los paquetes
- Que esté habilitado ufw, bloqueando todo el tráfico entrante y permitiendo solo ssh.
- Que solo se pueda hacer login con clave pública, y que root no pueda hacer login.
- Que esté instalado fail2ban y bloquee intentos fallidos de conexión SSH. El servicio debe quedar habilitado y activado.
- Debe haber un handler que reinicie el sistema si se actualizan paquetes.
- Debe haber un handler que reinicie ssh si cambia la configuración.

Para todos los playbooks deben estar documentadas las ejecuciones.

**\$ansible-playbook -i inventories/inventory.ini playbooks/hardening.yml
--become --extra-vars "@secret.yml" --ask-vault-pass**

```
[sysadmin@Bastion ort_obligatorio_ts12025]$ ansible-playbook -i inventories/inventory.ini playbooks/hardening.yml --become --extra-vars "@secret.yml" --ask-vault-pass
Vault password:
[WARNING]: Collection community.general does not support Ansible version 2.14.18

PLAY [Hardening de servidores Ubuntu] *****

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

TASK [TAREA#1 Actualizamos paquetes] *****
ok: [ubuntu01]

TASK [TAREA#2 Instalamos paquetes (ufw y fail2ban)] *****
ok: [ubuntu01]

TASK [TAREA#3 Política entrante DENY] *****
ok: [ubuntu01]

TASK [TAREA#4 Política saliente ALLOW] *****
ok: [ubuntu01]

TASK [TAREA#5 Permisos OpenSSH] *****
ok: [ubuntu01]

TASK [TAREA#6 Habilitamos UFW (y arranque automático)] *****
ok: [ubuntu01]

TASK [TAREA#7 SSH solo con clave pública] *****
ok: [ubuntu01]

TASK [TAREA#8 Prohibimos login de root por SSH] *****
ok: [ubuntu01]

TASK [TAREA#9 Jail SSH en fail2ban] *****
ok: [ubuntu01]

TASK [TAREA#10 Aseguramos fail2ban iniciado y habilitado] *****
ok: [ubuntu01]

PLAY RECAP *****
ubuntu01 : ok=11  changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Tarea 5: Responder las siguientes preguntas

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

Ansible es una herramienta de automatización de código abierto que reduce la complejidad de la administración y puede ejecutarse en cualquier entorno, permitiendo automatizar prácticamente cualquier tarea.

Algunas de las actividades que se pueden realizar con Ansible son:

- Eliminar la repetición y simplificar los flujos de trabajo
- Administrar y mantener la configuración del sistema
- Implementar continuamente software complejo

2. ¿Qué es un playbook de Ansible?

Un playbook de Ansible es un plano técnico en YAML que define tareas de automatización para ejecutarse con mínima intervención manual sobre inventarios o dispositivos específicos de TI.

Contiene uno o más plays (conjuntos de tareas) que se ejecutan mediante módulos, permitiendo automatizar acciones en servidores, redes, sistemas de seguridad, plataformas como Kubernetes o repositorios de código.

Se pueden guardar, compartir y reutilizar, garantizando que los procesos se realicen siempre de forma uniforme.

3. ¿Qué información contiene un inventario de Ansible?

Un inventario de Ansible contiene la lista de nodos administrados o hosts sobre los que se ejecutará la automatización, junto con las variables asociadas a cada uno. También puede incluir grupos de hosts para organizarlos, definir variables en bloque y facilitar su selección mediante patrones.

El inventario puede ser estático (por ejemplo, un archivo en `/etc/ansible/hosts`) o dinámico, generado desde fuentes como proveedores de nube, y puede combinar múltiples archivos o fuentes para mayor flexibilidad.

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

Un módulo de Ansible es una unidad de código que ejecuta tareas específicas de TI, cómo gestionar conexiones de red, configurar sistemas, administrar seguridad, usuarios, nubes o servicios. Los módulos se utilizan dentro de tareas, que forman parte de plays y playbooks, para automatizar diferentes casos prácticos.

Ejemplos:

- **ansible.builtin.dnf**: instala, actualiza o elimina paquetes y grupos utilizando el gestor de paquetes “dnf” en sistemas basados en Fedora.
- **ansible.builtin.service**: gestiona servicios en hosts remotos, permitiendo iniciarlos, detenerlos o reiniciarlos, entre otras acciones.

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

Su principal ventaja frente a otros sistemas de automatización radica en su diseño sin agentes, que elimina la necesidad de instalar software adicional en los nodos administrados y reduce la complejidad de la infraestructura. Además, su lenguaje simple basado en YAML y su facilidad de despliegue lo hacen accesible incluso para equipos con poca experiencia en programación. Estas características, junto con su capacidad para describir cómo los sistemas se interrelacionan y su amplia compatibilidad, convierten a Ansible en una solución más ligera, flexible y eficiente que muchas alternativas propietarias o basadas en agentes.

Bibliografía:

Utilizamos material brindado en aulas y lo dado en clase para la mayor parte del obligatorio, nos apoyamos en docs.ansible y toda la documentación/ejemplos que tiene.

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

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

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

https://docs.ansible.com/ansible/2.9/user_guide/vault.html

Módulos utilizados para el playbook de CentOS

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

https://docs.ansible.com/ansible/latest/collections/ansible/posix/firewalld_module.html

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

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

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

Módulos utilizados para el playbook de Ubuntu

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

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

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

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

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

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

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

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

<https://www.redhat.com/es/topics/automation-and-management/que-son-y-como-funcionan-los-playbooks-de-ansible>

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

<https://www.redhat.com/es/topics/automation-and-management/modulos-de-ansible-que-son-y-como-funcionan>

Prompts utilizados en IA:

A la opción become se le puede dar el parámetro con qué password haga sudo?

[Link](#)

ansible-vault create secret.yml explícame como usarlo.

[Link](#)

En vez de guardarla en el inventario el ansible_user, ¿puedo guardarlo en una lista de usuarios? una variable?

[Link](#)

Creación de logo de la carátula:

[Link](#)

Como subir una imagen a un readme de git:

[Link](#)