Universidad ORT Uruguay

Facultad de ingeniería

Obligatorio Taller de Servidores Linux

Entregado como requisito para la obtención de la materia
Taller de Servidores Linux

Matías Martínez – 252145

Mario Ourthe – Cabalé – 338039

Profesor: Enrique Verdes

2025

Contenido

Declaración de autoría

Nosotros, **Matias Martinez y Mario Ourthe-Cabalé**, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

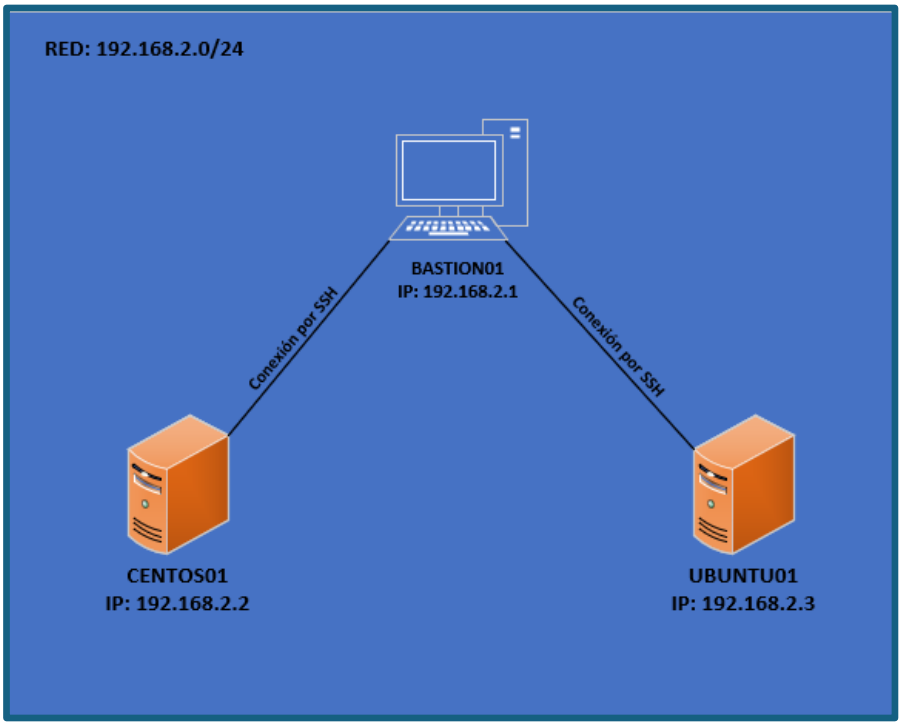
- La obra fue producida en su totalidad mientras realizábamos el obligatorio de **Taller de Servidores Linux**;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la 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 contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Introducción

En este documento se desarrollará una solución detallando como se configuran **playbooks**, comandos **ad-hoc** y como se trabaja con **repositorios de GitHub** en una infraestructura básica de administración de servidores **GNU/Linux** utilizando las distribuciones **CentOS Stream 9** y **Ubuntu 24.04**.

Diagrama de red

A continuación, se detallará un diagrama de red en el cual se realizaron las tareas del obligatorio con el fin de poder de tener un mejor contexto:



Github

El repositorio en donde se encuentra la solución es:

Obligatorio-Taller-Linux2025

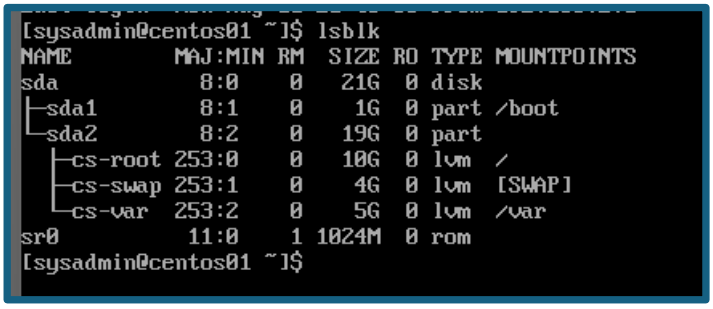
Instalación servidores

A continuación, se detallarán como fueron configurados los servidores **Centos01** y **Ubuntu01**.

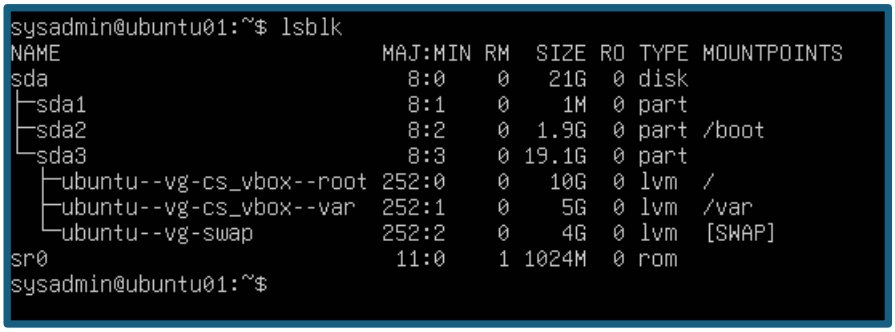
Ítems	Centos01	Ubuntu01
Sistema operativo y distribución	CentOS Stream 9	Ubuntu 24.04
CPU	1	1
RAM	2GB	2GB
Almacenamiento	21GB	21GB
Interfaces de red	2 (NAT y red interna)	2 (NAT y red interna)
IP	192.168.2.2	192.168.2.3
Interfaz gráfica	No	No

Almacenamiento		
Volumen	**Centos01**	**Ubuntu01**
/	10G	10G
/boot	2G	2G
/var	5G	5G
swap	4G	4G

LVM en **Centos01**:



LVM en **Ubuntu01**:



Inventario de Ansible

Para el caso del inventario de ansible, además de crear el archivo inventario **inventory.ini**, se creó el archivo **ansible.cfg** (en la raíz del proyecto) de manera de no tener que especificar la ruta del inventario en caso de que el comando ingresado lo requiera.

Inventory.ini

Primero creamos el directorio **inventories** donde se alojará nuestro archivo **inventory.ini**:

Comando:

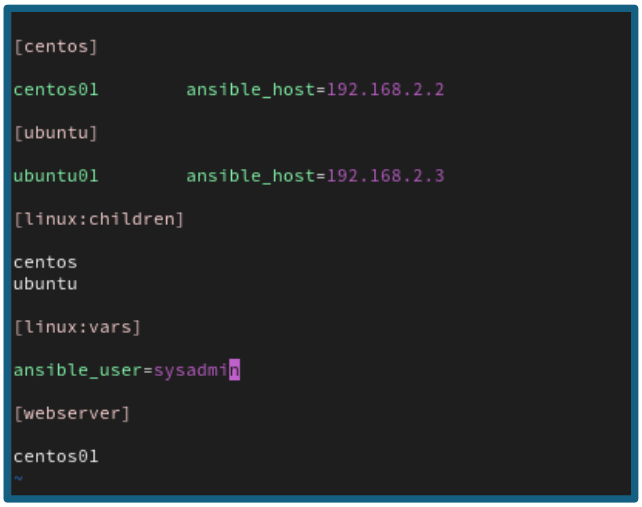
- **Mkdir inventories**

Luego procedemos a crear y configurar el archivo **inventory.ini**:

Comando:

- **Vim inventories/inventory.ini**

Configuración:



Ansible.cfg

```
[defaults]
inventory = ./inventories/inventory.ini
```

Pruebas

```
[sysadmin@localhost Obligatorio-Taller-Linux2025]$ ansible-inventory -i inventories/inventory.ini --graph
@all:
  |--@ungrouped:
  |--@linux:
  |  |--@centos:
  |  |  |--centos01
  |  |--@ubuntu:
  |  |  |--ubuntu01
  |--@webserver:
  |  |--centos01
[sysadmin@localhost Obligatorio-Taller-Linux2025]$ ansible all -i inventories/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"
}
```

Comandos Ad-Hoc

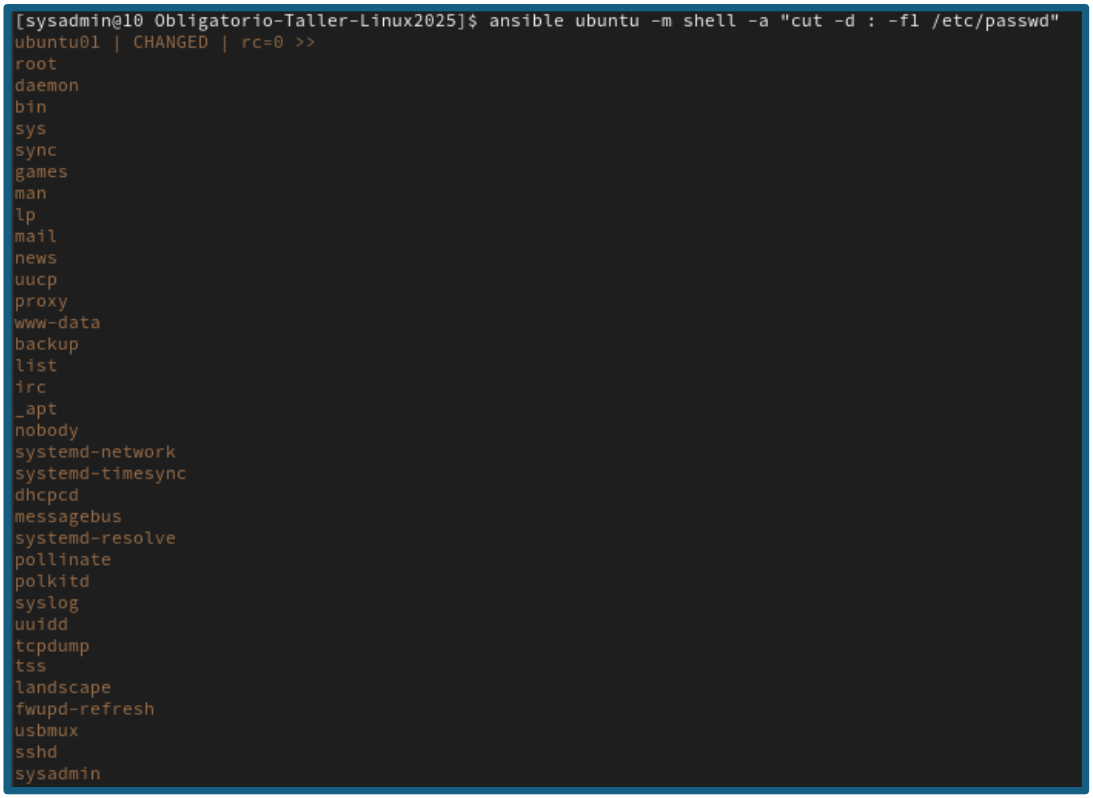
En este apartado procedemos a ejecutar algunos comandos ad-hoc buscando los siguientes resultados:

- Listar todos los usuarios en servidor Ubuntu.
- Mostrar el uso de memoria en todos los servidores.
- Que el servicio chrony esté instalado y funcionando en servidor Centos.

A continuación, se detallarán los comandos utilizados con su correspondiente captura:

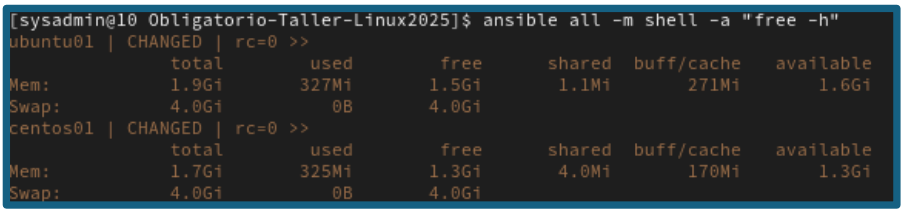
- Listar todos los usuarios en servidor Ubuntu:

```
$ ansible ubuntu -m shell -a "cut -d : -f1 /etc/passwd"
```



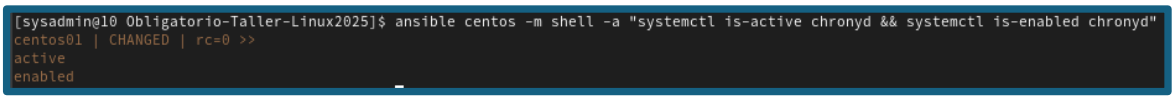
- Mostrar el uso de memoria en todos los servidores:

```
$ ansible all -m shell -a "free -h"
```



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

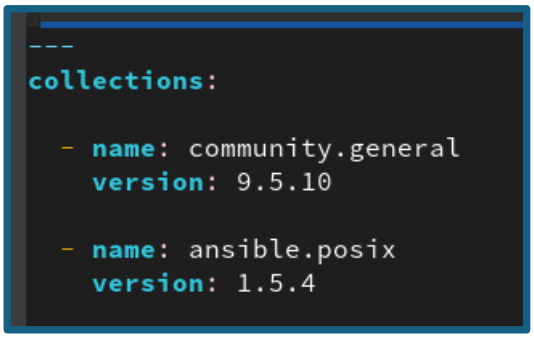
```
$ ansible centos -m shell -a “systemctl is-active chronyd
&& systemctl is-enabled chronyd”
```



Playbooks

Antes de comenzar con el detalle de los playbooks creados es importante mencionar que se debieron de instalar algunos módulos con un versionado especifico de manera de que sea compatible con la versión de **ansible (2.14.18)**, evitando de esta manera los **Warning** generados al ejecutar un playbook.

El archivo de configuración se encuentra ubicado en **collections/requirement.yaml** y su configuración quedó de la siguiente manera:



```yaml
---
collections:

  - name: community.general
    version: 9.5.10

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

Nfs_setup.yaml

En la siguiente playbok se configuraron varias plays de manera que realicen las siguientes acciones en servidores CentOS:

- 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.

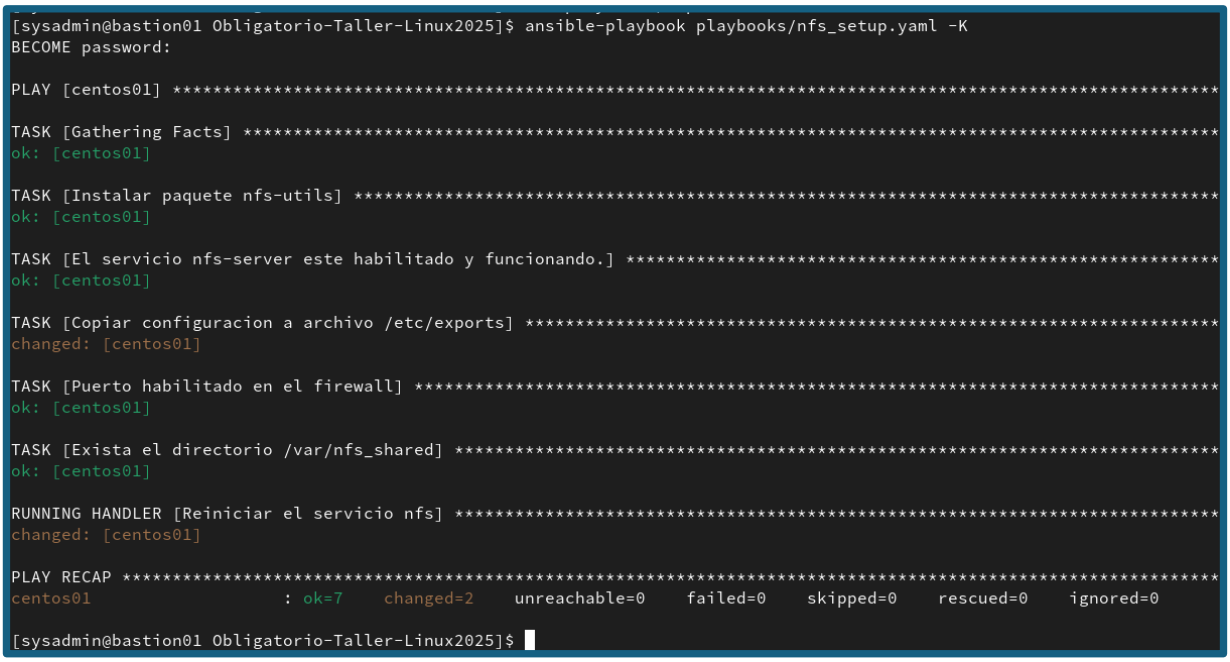
Configuración

A continuación, se detallará como fue configurada la playbook de nfs_setup.yaml

```yaml
---
- hosts: centos01
  become: true

  tasks:

  - name: Instalar paquete nfs-utils
    ansible.builtin.dnf:
      name: nfs-utils
      state: present

  - name: El servicio nfs-server este habilitado y funcionando.
    ansible.builtin.systemd_service:
      name: nfs-server
      enabled: true
      state: started

  - name: Copiar configuracion a archivo /etc/exports
    ansible.builtin.copy:
      src: ../inventories/group_vars/exports
      dest: /etc/exports
      owner: root
      group: root
      mode: '0644'
    notify: Reiniciar el servicio nfs

  - name: Puerto habilitado en el firewall
    ansible.posix.firewalld:
      service: nfs
      state: enabled
      permanent: true
      immediate: true

  - name: Exista el directorio /var/nfs_shared
    ansible.builtin.file:
      path: /var/nfs_shared
      state: directory
      owner: nobody
      group: nobody
      mode: '0777'

  handlers:

  - name: Reiniciar el servicio nfs
    ansible.builtin.service:
      name: nfs-server
      state: restarted
```

Validaciones

Ejecucion del playbook:



Hardening.yaml

En la siguiente playbok se configuraron varias plays de manera que realicen las siguientes acciones en servidores Ubuntu:

- 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.
- Un handler que reinicie el sistema si se actualizan paquetes y reinicie ssh si cambia la configuración.

Configuración

A continuación, se detallará como fue configurada la playbook **hardening.yaml:**

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

  tasks:
    - name: Mantener sistema actualizado
      ansible.builtin.apt:
        name: "*"
        state: latest
        update_cache: true
      notify: Reiniciar sistema

    - name: Habilito ufw y niego trafico entrante
      community.general.ufw:
        state: enabled
        policy: deny

    - name: Habilito trafico ssh
      community.general.ufw:
        rule: limit
        port: ssh
        proto: tcp
      notify: Reiniciar daemon SSH

    - name: No permitir login como root
      ansible.builtin.lineinfile:
        path: /etc/ssh/sshd_config
        insertafter: '^#PermitRootLogin'
        line: PermitRootLogin no
      notify: Reiniciar daemon SSH

    - name: Elimino archivo /etc/ssh/sshd_config.d/50-cloud-init.conf
      ansible.builtin.file:
        path: /etc/ssh/sshd_config.d/50-cloud-init.conf
        state: absent

    - name: Deshabilito login por contraseña por ssh
      ansible.builtin.lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#PasswordAuthentication'
        line: PasswordAuthentication no
      notify: Reiniciar daemon SSH

    - name: Validamos que este habilitado el login con clave publica
      ansible.builtin.lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#PubkeyAuthentication'
        line: PubkeyAuthentication yes
        state: present
      notify: Reiniciar daemon SSH
```

```
  - name: Instalar fail2ban
    ansible.builtin.apt:
      name: fail2ban
      state: present
      update_cache: true

  - name: Validacion que fail2ban se encuentre habilitado y corriendo
    ansible.builtin.systemd_service:
      name: fail2ban
      state: started
      enabled: true

  - name: Configuracion local de fail2ban
    ansible.builtin.copy:
      dest: /etc/fail2ban/jail.local
      content: |
       [DEFAULT]
       bantime = 600
       findtime = 600
       maxretry = 3
       backend = auto

       [sshd]
       enabled = true
       port = ssh
       filter = sshd
       logpath = /var/log/auth.log
       maxretry = 3
       bantime = 600
      owner: root
      group: root
      mode: '0640'
    notify: Reiniciar servicio fail2ban

handlers:

  - name: Reiniciar sistema
    ansible.builtin.reboot:

  - name: Reiniciar daemon SSH
    ansible.builtin.systemd_service:
      name: ssh
      state: restarted

  - name: Reiniciar servicio fail2ban
    ansible.builtin.systemd_service:
      name: fail2ban
      state: restarted
      enabled: yes
```

Validaciones

- **Ejecución de Playbook**:

```
[sysadmin@10 Obligatorio-Taller-Linux2025]$ ansible-playbook playbooks/hardening.yaml -K
BECOME password:

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

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

TASK [Mantener sistema actualizado] ********************************************************************************
changed: [ubuntu01]

TASK [Habilito ufw y niego trafico entrante] ***********************************************************************
changed: [ubuntu01]

TASK [Habilito trafico ssh] ****************************************************************************************
changed: [ubuntu01]

TASK [No permitir login como root] *********************************************************************************
changed: [ubuntu01]

TASK [Deshabilito login por contraseña por ssh] ********************************************************************
changed: [ubuntu01]

TASK [Validamos que este habilitado el login con clave publica] ****************************************************
changed: [ubuntu01]

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

TASK [Validacion que fail2ban se encuentre habilitado y corriendo] *************************************************
ok: [ubuntu01]

TASK [Configuracion local de fail2ban] *****************************************************************************
changed: [ubuntu01]

RUNNING HANDLER [Reiniciar sistema] ********************************************************************************
changed: [ubuntu01]

RUNNING HANDLER [Reiniciar daemon SSH] *****************************************************************************
changed: [ubuntu01]

RUNNING HANDLER [Reiniciar servicio fail2ban] **********************************************************************
changed: [ubuntu01]

PLAY RECAP *********************************************************************************************************
ubuntu01                   : ok=13   changed=11   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- **Estado de UFW**:

```
sysadmin@ubuntu01:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

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

- **Servicio Fail2ban**:

En **ubuntu01** validamos que se encuentre el servicio antes y después de ejecutar el playbook, el cual validamos que aparece una vez ejecutado el playbook:

```
sysadmin@ubuntu01:~$ ls /lib/systemd/system | grep fail2ban
sysadmin@ubuntu01:~$ ls /lib/systemd/system | grep fail2ban
fail2ban.service
```

Reflexiones

Fue un proceso de mucho trabajo, donde tuvimos que adentrarnos y comprender bien los módulos de ansible-core y otras colecciones que utilizamos.
Nos gustó mucho la herramienta por la facilidad de entendimiento cuando esta se ejecuta.
Tuvimos algunas dificultades, pero al ser tan fácil de interpretar los mensajes y la escritura del código, se nos hizo amigable superar los desafíos.
Entendemos que es una herramienta muy potente que nos va a acompañar de ahora en más en nuestras labores diarias.

Bibliografía

- ansible.builtin.copy module – Copy files to remote locations — Ansible Community Documentation
- ansible.builtin.file module – Manage files and file properties — Ansible Community Documentation
- Módulo ansible.builtin.user – Administrar cuentas de usuario — Ansible Community Documentation
- Módulo community.general.ufw – Administrar firewall con UFW — Ansible Community Documentation
- Módulo ansible.builtin.apt – Gestiona apt-packages — Ansible Comunidad Documentation
- Seguridad Linux: Proteja sus sistemas con fail2ban
- Material de la ORT.
- fail2ban/config/jail.conf at master · fail2ban/fail2ban · GitHub
- Community.General — Ansible Community Documentation
- Ansible Galaxy - community.general
- Promt: Que configuraciones debo realizar en fail2ban.