

Universidad ORT Uruguay

Facultad de Ingeniería

Bernard Wand Polak



Obligatorio TSL

Agosto 2025

Bruno Rodríguez N° - 257299

**Docente: Enrique Verdes
Guillermo Ferradas**

INDICE

Contenido

Facultad de Ingeniería.....	1
Bernard Wand Polak.....	1
INDICE	2
TAREA 1 – INSTALACION DE SERVIDORES	3
TAREA 2 - ARCHIVO DE INVENTARIO ANSIBLE	4
TAREA 3 – COMANDOS AD-HOC	6
TAREA 5 – CUESTIONARIO ANSIBLE.....	8

TAREA 1 – INSTALACION DE SERVIDORES

Para la realización del obligatorio del taller de Servidores Linux se instalaron 2 máquinas virtuales, una distribución CentOS Stream 9 y Ubuntu 24.04.

Ambos equipos cuentan 1 CPU, 2GB de memoria RAM, 2 interfaces de RED y con 4 filesystem:

	CentOS Stream 9	Ubuntu 24.04
/boot	1G	2G
/	10G	10G
/var	5G	5G
SWAP	4G	4G

Además, se utilizó un servidor CentOS 9 con interface grafica como bastión desde donde se hacen las conexiones SSH y se ejecutan los playbooks.

Capturas

CentOS 9:

```
[sysadmin@tsl-centos ~]$ sudo lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda              8:0    0   21G  0 disk
├─sda1           8:1    0    1G  0 part /boot
└─sda2           8:2    0   19G  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
sr0             11:0    1 1024M  0 rom

[sysadmin@tsl-centos ~]$ 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 08:00:27:2d:48:6b 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 65387sec preferred_lft 65387sec
    inet6 fd17:625c:f037:2:a00:27ff:fe2d:486b/64 scope global dynamic noprefixroute
        valid_lft 86105sec preferred_lft 14105sec
    inet6 fe80::a00:27ff:fe2d:486b/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 08:00:27:37:7d:9f brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.51/24 brd 192.168.2.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
[sysadmin@tsl-centos ~]$
```

Ubuntu 24.04

```

sysadmin@tsl-ubuntu:~$ sudo lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   21G  0 disk
├─sda1       8:1    0    1M  0 part
├─sda2       8:2    0   10G  0 part /
├─sda3       8:3    0    2G  0 part /boot
├─sda4       8:4    0    5G  0 part /var
└─sda5       8:5    0    4G  0 part /SWAP
sr0         11:0    1 1024M  0 rom

sysadmin@tsl-ubuntu:~$ 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 08:00:27:8a:cd:4f brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 66441sec preferred_lft 66441sec
    inet6 fd17:625c:f037:2:a00:27ff:fe8a:cd4f/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86356sec preferred_lft 14356sec
    inet6 fe80::a00:27ff:fe8a:cd4f/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 08:00:27:7b:f2:55 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.50/24 brd 192.168.2.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe7b:f255/64 scope link
        valid_lft forever preferred_lft forever
sysadmin@tsl-ubuntu:~$

```

TAREA 2 - ARCHIVO DE INVENTARIO

ANSIBLE

En el servidor bastión en el directorio de nuestro proyecto se creó una carpeta “inventory” con el fichero “inventory.ini” dentro, en el mismo se definieron los grupos: Centos, Ubuntu, Linux y Fileserver.

```
# cat ./inventory/inventory.ini
```

```

[sysadmin@bastion01 OBLIGATORIO-TSL]$ cat ./inventory/inventory.ini
[Centos]
tsl-centos      ansible_host=192.168.2.51

[Ubuntu]
tsl-ubuntu      ansible_host=192.168.2.50

[Linux:children]
Centos
Ubuntu

[Fileserver]
tsl-centos

```

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

```
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible-inventory -i inventory/inventory.ini --list
{
  "Centos": {
    "hosts": [
      "tsl-centos"
    ]
  },
  "Fileserver": {
    "hosts": [
      "tsl-centos"
    ]
  },
  "Linux": {
    "children": [
      "Centos",
      "Ubuntu"
    ]
  },
  "Ubuntu": {
    "hosts": [
      "tsl-ubuntu"
    ]
  },
  "_meta": {
    "hostvars": {
      "tsl-centos": {
        "ansible_host": "192.168.2.51"
      },
      "tsl-ubuntu": {
        "ansible_host": "192.168.2.50"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped",
      "Linux",
      "Fileserver"
    ]
  }
}
```

ansible all -i inventory/inventory.ini -m ping

```
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible all -i inventory/inventory.ini -m ping
tsl-ubuntu | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
tsl-centos | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
[sysadmin@bastion01 OBLIGATORIO-TSL]$
```

TAREA 3 – COMANDOS AD-HOC

a – Listar todos los usuarios en el servidor Ubuntu

```
# ansible Ubuntu -i inventory/inventory.ini -m ansible.builtin.getent -a "database=passwd"
```

Este nos muestra todos los usuarios, incluidos los de sistema, si queremos filtrar por usuarios “normales” debemos usar el modulo shell y pasar un comando como lo haríamos sin ansible, filtrando por bash que utiliza el usuario o por número de grupo, ya que por defecto empiezan a contar a partir de 1000.

```
# ansible Ubuntu -i inventory/inventory.ini -m shell -a "getent passwd | grep 'bash'"
```

```
# ansible Ubuntu -i inventory/inventory.ini -m shell -a "getent passwd | grep '1[0-9][0-9][0-9]'"
```

```
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible Ubuntu -i inventory/inventory.ini -m shell -a "'getent passwd | grep '/sbin/bash'"
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible Ubuntu -i inventory/inventory.ini -m shell -a "'getent passwd | grep 'bash'"
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible Ubuntu -i inventory/inventory.ini -m shell -a "getent passwd | grep 'bash'"
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible Ubuntu -i inventory/inventory.ini -m shell -a "getent passwd | grep '10*'"
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible Ubuntu -i inventory/inventory.ini -m shell -a "getent passwd | grep '10[0-9]'"
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible Ubuntu -i inventory/inventory.ini -m shell -a "getent passwd | grep '10[0-9][0-9]'"
[sysadmin@bastion01 OBLIGATORIO-TSL]$
```

b – Mostrar el uso de memoria en todos los servidores

```
# ansible Linux -i inventory/inventory.ini -m ansible.builtin.setup -a  
"filter=ansible_memory_*
```

c – Verificar que chrony esté instalado y corriendo en el servidor CentOS

```
# ansible centos -i inventory/inventory.ini -m  
ansible.builtin.systemd_service -a "service=chronyd state=started" | head -n  
7
```

Con este comando podemos saber si el servicio está instalado y ponerlo a correr si hace falta, pero no instalarlo, para eso habría que utilizar el módulo dnf.

"| head -n 7" se debe a que el módulo systemd_service por defecto trae el status del servicio que normalmente ocupa muchas líneas, con esto nos aseguramos de ver lo solicitado, si el servicio existe y si está corriendo.

```
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible Centos -i inventory/inventory.ini -m ansible.builtin.systemd_service -a "state=started service=chrony" | head -n 7  
ts1-centos | FAILED! => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "msg": "Could not find the requested service chrony: host"  
}  
[sysadmin@bastion01 OBLIGATORIO-TSL]$ ansible Centos -i inventory/inventory.ini -m ansible.builtin.systemd_service -a "state=started service=chronyd" | head -n 7  
ts1-centos | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "name": "chronyd",  
  "state": "started"  
}  
[sysadmin@bastion01 OBLIGATORIO-TSL]$
```

TAREA 5 – CUESTIONARIO ANSIBLE

a – ¿Qué es Ansible? Mencione dos actividades que se pueden hacer con Ansible.

r: Ansible es una herramienta de automatización OpenSource, con esta herramienta se puede automatizar tareas de aprovisionamiento, configuración y flujos de trabajo.

b – ¿Qué es un playbook de Ansible?

r: Es un documento yaml donde se establece una lista de acciones que luego va a ejecutar el o los dispositivos que queramos.

c – ¿Qué información contiene un inventario de Ansible?

r: Contiene una lista con los hosts que queramos que Ansible administre, estos los podemos juntar por grupos y ejecutar un playbook en muchos equipos con un solo llamado.

d – Explique que es un módulo de Ansible y dé un ejemplo

r: Son scripts los cuales llama Ansible para poder realizar las acciones indicadas en un comando ad-hoc o un playbook, cada módulo se utiliza para realizar tareas concretas, los módulos yum y apt se utilizan para instalar, actualizar o desinstalar paquetes.

e – ¿Qué ventajas tiene Ansible sobre otros métodos de automatización?

r: Ansible es OpenSource por lo que no requiere pagar un licenciamiento, no requiere instalar agentes en los equipos administrados, es muy sencillo de usar con una curva de aprendizaje rápido.