

# Trabajo Obligatorio

## Taller de servidores Linux

ORT | Analista en Infraestructura Informática  
2025

Fabrizio Ramirez - 327619

Sebastián Bentancur - 335428

# Índice

Índice .....	2
Introducción .....	3
Tarea 1: Instalar servidores .....	4
Instalación Ubuntu - Ubuntu01 .....	4
Instalación CentOS - Bastion01 .....	5
Instalación CentOS - Centos01 .....	6
Parte 2: Inventario de hosts .....	6
Parte 3: Comandos Ad-Hoc .....	8
Listar todos los usuarios en servidor Ubuntu .....	8
Mostrar el uso de memoria en todos los servidores .....	8
Que el servicio chrony esté instalado y funcionando en servidor Centos .....	9
Parte 4: Playbooks .....	10
Playbooks/nfs_setup.yaml .....	10
Playbooks/hardening.yaml .....	11
Desafíos encontrados .....	12
Reflexión de desafíos: .....	12
Comandos de IAG .....	12
Referencias utilizadas .....	12

# Introducción

El siguiente trabajo tiene como objetivo aplicar los conocimientos adquiridos en la administración de servidores Linux mediante el uso de automatizaciones Ansible.

Se desarrollaron y documentaron escenarios sobre un servidor Centos y un servidor Ubuntu, incluyendo instalación y configuración de servicios, uso de inventarios, ejecución de comandos “ad-hoc” y la creación de “playbooks” orientados a la implementación de “NFS” y configuración de “hardening”.

El trabajo realizado fue implementado mediante la utilización de un repositorio Git, el cual contiene el código de los playbooks, inventario, capturas y demás documentación solicitada.

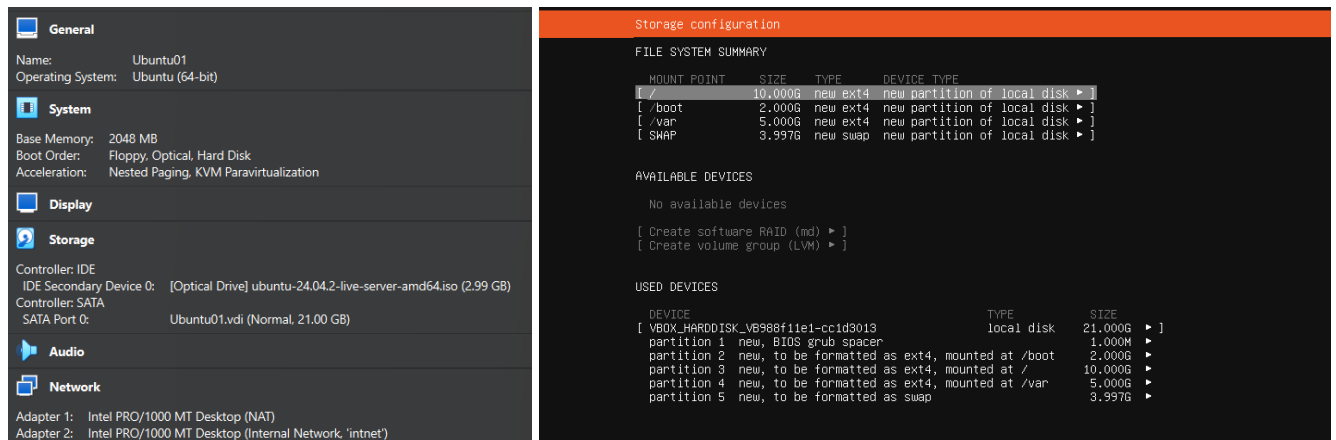
Repositorio de la tarea: <https://github.com/sbentancur/TallerASLX2025>

# Tarea 1: Instalar servidores

## Instalación Ubuntu - Ubuntu01

Se instala y configura Ubuntu siguiendo lineamientos de trabajo Obligatorio.

Registrándose hostname e IP.



```
sysadmin@ubuntu01:~$ 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    2G  0 part /boot
├─sda3       8:3    0   10G  0 part /
├─sda4       8:4    0    5G  0 part /var
└─sda5       8:5    0    4G  0 part [SWAP]
sr0         11:0    1 1024M  0 rom
```

Se configuran interfaces de redes:

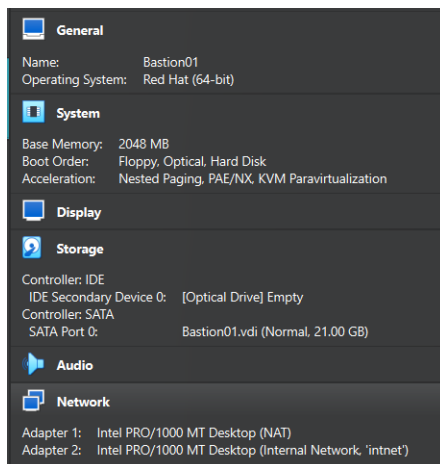
```
# sudo vi /etc/netplan/50-cloud-init.yaml
```

```
# sudo netplan apply
```

```
# sudo apt install openssh-server -y
```

```
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      dhcp6: no
      addresses: [192.168.1.10/24, 1
```

# Instalación CentOS - Bastion01



**General**

Name: Bastion01  
Operating System: Red Hat (64-bit)

**System**

Base Memory: 2048 MB  
Boot Order: Floppy, Optical, Hard Disk  
Acceleration: Nested Paging, PAE/NX, KVM Paravirtualization

**Display**

**Storage**

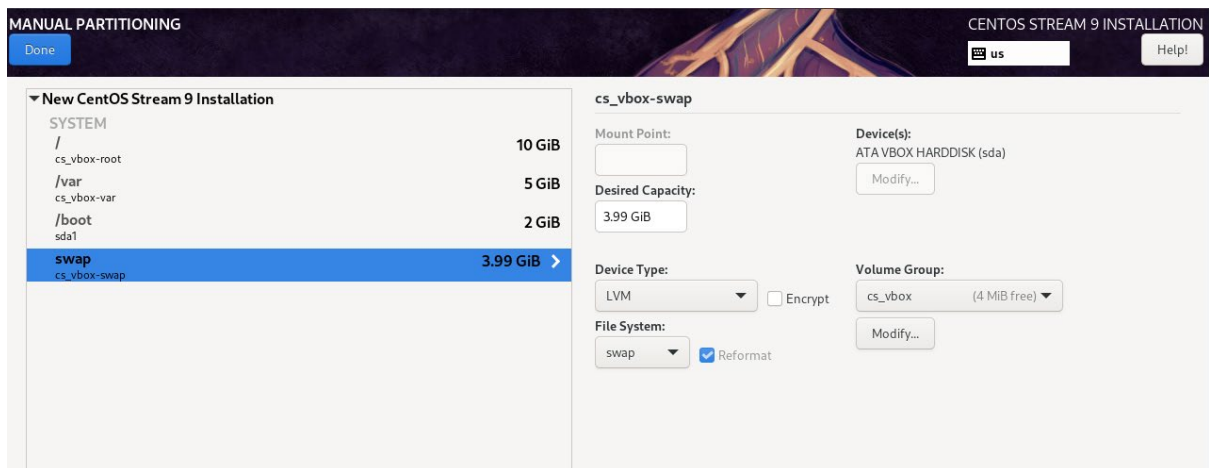
Controller: IDE  
IDE Secondary Device 0: [Optical Drive] Empty  
Controller: SATA  
SATA Port 0: Bastion01.vdi (Normal, 21.00 GB)

**Audio**

**Network**

Adapter 1: Intel PRO/1000 MT Desktop (NAT)  
Adapter 2: Intel PRO/1000 MT Desktop (Internal Network, 'intnet')

Se realiza la instalación de Bastión 1, modificándose las particiones por defecto de acuerdo con la letra de Obligatorio. Se hace configuración inicial de Red, y hostname.



**MANUAL PARTITIONING** CENTOS STREAM 9 INSTALLATION

Done us Help!

▼ New CentOS Stream 9 Installation

SYSTEM	Size
/	10 GiB
cs_vbox-root	
/var	5 GiB
cs_vbox-var	
/boot	2 GiB
sda1	
<b>swap</b>	<b>3.99 GiB</b>
cs_vbox-swap	

**cs\_vbox-swap**

Mount Point:

Device(s): ATA VBOX HARDISK (sda) Modify...

Desired Capacity: 3.99 GiB

Device Type: LVM ☐ Encrypt

File System: swap ☒ Reformat

Volume Group: cs\_vbox (4 MiB free) Modify...

```
[sysadmin@Bastion01 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   21G  0 disk
├─sda1       8:1    0    2G  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
```

## Instalación CentOS - Centos01

Se sigue la línea de la configuración realizada en los anteriores servidores y se crea y configura el servidor con el siguiente esquema de particionado.

```
[sysadmin@Centos01 ~]$ lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                  8:0    0   21G  0 disk
├─sda1               8:1    0    2G  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
```

## Parte 2: Inventario de hosts

Inventory.ini

```
[sysadmin@Bastion01 ~]$ vi TallerASLX2025/inventory.ini
[centos]
centos01      ansible_host=192.168.1.15 ansible_user=sysadmin

[ubuntu]
ubuntu01     ansible_host=192.168.1.10 ansible_user=sysadmin
```

*# ansible-inventory -i inventory.ini --list*

```
[sysadmin@Bastion01 TallerASLX2025]$ ansible-inventory -i inventory.ini --list
{
  "_meta": {
    "hostvars": {
      "centos01": {
        "ansible_host": "192.168.1.15",
        "ansible_user": "sysadmin"
      },
      "ubuntu01": {
        "ansible_host": "192.168.1.10",
        "ansible_user": "sysadmin"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped",
      "centos",
      "ubuntu"
    ]
  },
  "centos": {
    "hosts": [
      "centos01"
    ]
  },
  "ubuntu": {
    "hosts": [
      "ubuntu01"
    ]
  }
}
```

*# ansible all -i inventory.ini -m ping*

```
[sysadmin@Bastion01 TallerASLX2025]$ 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"
}
```

## Parte 3: Comandos Ad-Hoc

### Listar todos los usuarios en servidor Ubuntu

*# ansible ubuntu -m shell -a "getent passwd | cut -d: -f1"*

```
[sysadmin@Bastion01 TallerASLX2025]$ ansible ubuntu -m shell -a "getent passwd | cut -d: -f1"
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
dhcpd
messagebus
systemd-resolve
pollinate
polkitd
syslog
uidd
tcpdump
tss
landscape
fwupd-refresh
usbmux
sysadmin
sshd
```

### Mostrar el uso de memoria en todos los servidores

*# ansible all -m shell -a "free -h"*

```
[sysadmin@Bastion01 TallerASLX2025]$ ansible all -m shell -a "free -h"
ubuntu01 | CHANGED | rc=0 >>
Mem:          1.9Gi       360Mi       1.1Gi       1.1Mi       589Mi       1.6Gi
Swap:         4.0Gi           0B       4.0Gi
centos01 | CHANGED | rc=0 >>
Mem:          1.7Gi       370Mi       1.2Gi       4.0Mi       303Mi       1.3Gi
Swap:         4.0Gi           0B       4.0Gi
```



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

*# ansible centos -m yum -a "name=chrony state=present" -K -b*

*# ansible centos -m systemd -a "name=chronyd enabled=yes state=started" -K -b*

```
[sysadmin@Bastion01 TallerASLX2025]$ ansible centos -m yum -a "name=chrony state=present" -K -b
BECOME password:
centos01 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "msg": "",
  "rc": 0,
  "results": [
    "Installed: chrony-4.6.1-2.el9.x86_64"
  ]
}
[sysadmin@Bastion01 TallerASLX2025]$ ansible centos -m systemd -a "name=chronyd enabled=yes state=started" -K -b
BECOME password:
centos01 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "enabled": true,
  "name": "chronyd",
  "state": "started",
  "status": {
    "AccessSELinuxContext": "system_u:object_r:chronyd_unit_file_t:s0",
    "ActiveEnterTimestamp": "Sun 2025-08-10 22:48:28 -03",
    "ActiveEnterTimestampMonotonic": "2265906334",
    "ActiveExitTimestamp": "Sun 2025-08-10 23:04:18 -03",
    "ActiveExitTimestampMonotonic": "3216634870",
    "ActiveState": "inactive",
    "After": "var.mount snmp.service -.mount tmp.mount systemd-tmpfiles-setup.service basic.target systemd-journald.socket ntpdate.service ntpd.service sysinit.target system.slice",
    "AllowIsolate": "no",
    "AssertResult": "yes",
    "AssertTimestamp": "Sun 2025-08-10 22:48:28 -03",
    "AssertTimestampMonotonic": "2265854523",
    "Before": "multi-user.target shutdown.target",
  }
}
```

```
[sysadmin@Centos01 ~]$ systemctl status chronyd
● chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-08-10 23:05:00 -03; 1h 11min ago
     Docs: man:chronyd(8)
           man:chrony.conf(5)
   Process: 4984 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 4986 (Chronyd)
     Tasks: 1 (limit: 10654)
    Memory: 1.0M
       CPU: 37ms
   CGroup: /system.slice/chronyd.service
           └─4986 /usr/sbin/chronyd -F 2

Aug 10 23:05:00 Centos01 chronyd[4986]: chronyd version 4.6.1 starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER +SIGND +ASYNCDNS +NTS +SECHASH +IPV6 +D
Aug 10 23:05:00 Centos01 chronyd[4986]: Loaded 0 symmetric keys
Aug 10 23:05:00 Centos01 chronyd[4986]: Using right/UTC timezone to obtain leap second data
Aug 10 23:05:00 Centos01 chronyd[4986]: Initial frequency -17.500 ppm
Aug 10 23:05:00 Centos01 chronyd[4986]: Loaded seccomp filter (level 2)
Aug 10 23:05:00 Centos01 systemd[1]: Started NTP client/server.
Aug 10 23:05:06 Centos01 chronyd[4986]: Selected source 200.40.115.74 (2.centos.pool.ntp.org)
Aug 10 23:05:06 Centos01 chronyd[4986]: System clock wrong by 4226.798144 seconds
Aug 11 00:15:32 Centos01 chronyd[4986]: System clock was stepped by 4226.798144 seconds
Aug 11 00:15:32 Centos01 chronyd[4986]: System clock TAI offset set to 37 seconds
```

## Parte 4: Playbooks

### Playbooks/nfs\_setup.yaml

```
[sysadmin@Bastion01 TallerASLX2025]$ ansible-playbook playbooks/nfs_setup.yaml -K
BECOME password:

PLAY [Configuracion de servidor NFS] *****

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

TASK [Instalar NFS] *****
changed: [centos01]

TASK [Habilitar e iniciar servicio NFS] *****
changed: [centos01]

TASK [Permitir conexiones al puerto 2049] *****
changed: [centos01]

TASK [Crear directorio] *****
changed: [centos01]

TASK [Configurar export] *****
changed: [centos01]

RUNNING HANDLER [Actualizar exports] *****
changed: [centos01]

PLAY RECAP *****
centos01 : ok=7  changed=6  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

## Playbooks/hardening.yaml

```
[sysadmin@Bastion01 TallerASLX2025]$ ansible-playbook playbooks/hardening.yaml -K
BECOME password:

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

TASK [Sys actualizado] *****
^[[1;5P
ok: [ubuntu01]

TASK [Valida UFW permitiendo SSH] *****
ok: [ubuntu01]

TASK [Permitir trafico SSH] *****
ok: [ubuntu01]

TASK [Configura politicas default] *****
ok: [ubuntu01] => (item={'direction': 'incoming', 'policy': 'deny'})
Leok: [ubuntu01] => (item={'direction': 'outgoing', 'policy': 'allow'})

TASK [Habilita UFW] *****
ok: [ubuntu01]

TASK [Valida OpenSSH esta instalado] *****
ok: [ubuntu01]

TASK [Deshabilitar root login] *****
ok: [ubuntu01]

TASK [Habilitar login con ssh pub] *****
ok: [ubuntu01]

TASK [Deshabilitar login por pass SSH] *****
ok: [ubuntu01]

TASK [Instalar Fail2ban] *****
ok: [ubuntu01]

TASK [Activar Jail SSH Fail2ban] *****
changed: [ubuntu01]

TASK [Habilitar y arrancar fail2ban] *****
ok: [ubuntu01]

RUNNING HANDLER [Reinicio Fail2ban] *****
changed: [ubuntu01]

PLAY RECAP *****
ubuntu01 : ok=14 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

El archivo indicado /etc/ssh/sshd\_config.d/50-cloud-init.conf y sobrescribe el /etc/sshd\_config no se encontró por lo que no se pudieron replicar cambios correspondientes a la configuración por defecto.

```
sysadmin@ubuntu01:~$ tree /etc/ssh/sshd_config.d/
/etc/ssh/sshd_config.d/

0 directories, 0 files
```

## Desafíos encontrados

### 1. Configuración de red Ubuntu:

Debido a que Ubuntu maneja diferente el módulo de red, se debe de utilizar netplan lo que dificulta la puesta a punto inicial.

### 2. Manejo de Branch en GIT:

Dado al hecho que no se ejemplificaron en clase, se nos dificultó el uso de “branches” por lo que mantuvimos un entorno de trabajo en main únicamente

### 3. Módulos de ANSIBLE:

Módulos que no son parte del “ansible.builtin” nos resultaron difíciles de encontrar documentación respecto a ellos dado que la comunidad publica módulos parecidos de nombre y funciones.

## Reflexión de desafíos:

Utilizando la basta información presente tanto en documentación oficial como en foros de la comunidad, Ansible es una gran herramienta con la cual aumentar la productividad y eficiencia de cualquier arquitectura de servidores.

## Comandos de IAG

- [Think Longer] Analiza el siguiente playbook (hardening.yaml) y recomienda posibles mejoras con respecto a seguridad y practicas comunes en automatizaciones Ansible para Ubuntu
- [Think Longer] En un entorno automatizado Ansible, ¿es recomendable utilizar un Playbook general que ejecute sub-playbooks? ¿Tiene alguna mejora en rendimiento?

## Referencias utilizadas

[Ansible: ¿qué es y cómo funciona?](#)

[Módulos de Ansible: qué son y cómo funcionan](#)

<https://www.redhat.com/en/blog/configure-hostname-linux>

[https://docs.redhat.com/en/documentation/red\\_hat\\_enterprise\\_linux/7/html/networking\\_guide/sec-configuring\\_ip\\_networking\\_with\\_nmtui](https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/networking_guide/sec-configuring_ip_networking_with_nmtui)

<https://netplan.io/>

<https://www.redhat.com/en/blog/configure-ssh-keygen>