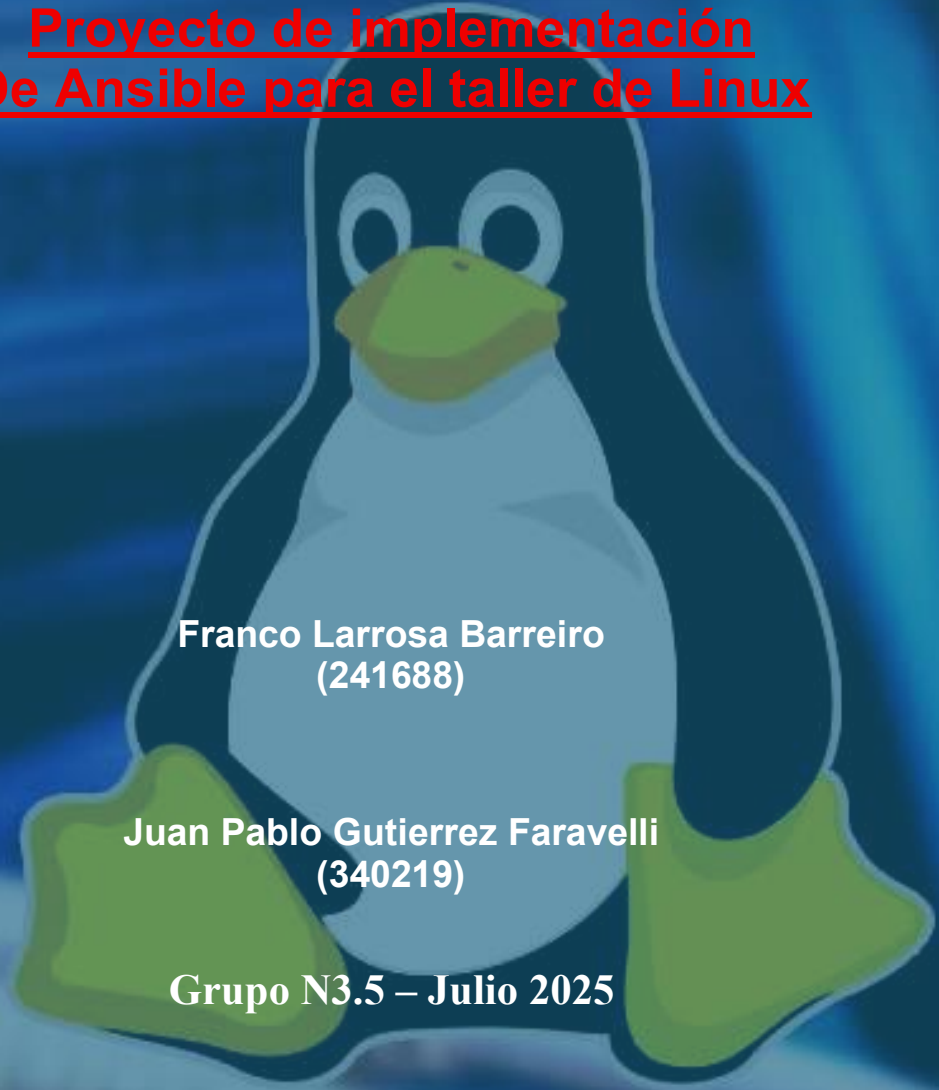


# Universidad ORT Uruguay

Facultad de Ingeniería  
Escuela de Tecnología

## Proyecto de implementación De Ansible para el taller de Linux



Franco Larrosa Barreiro  
(241688)

Juan Pablo Gutierrez Faravelli  
(340219)

Grupo N3.5 – Julio 2025

Docente: Enrique Verdes

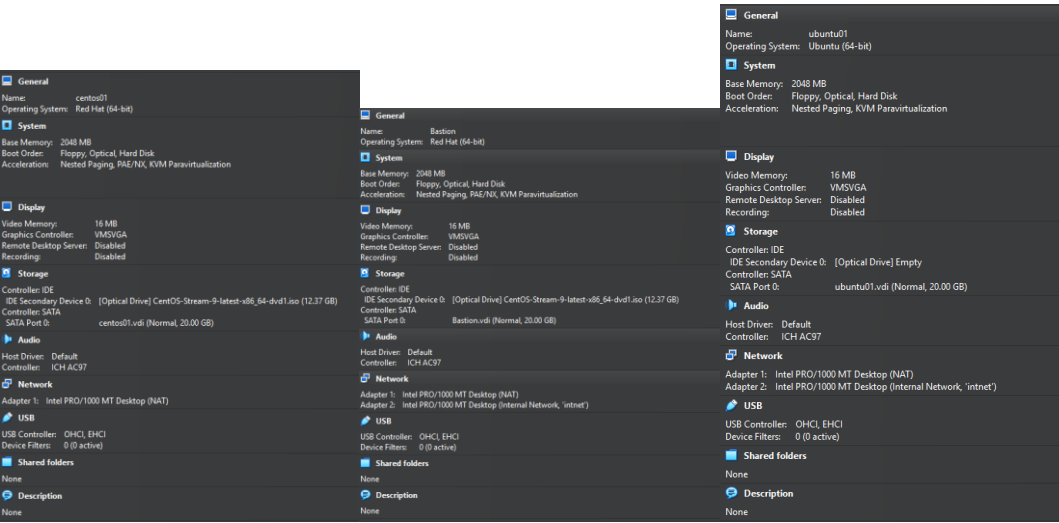
**Analista en Infraestructura informática**

## Contents

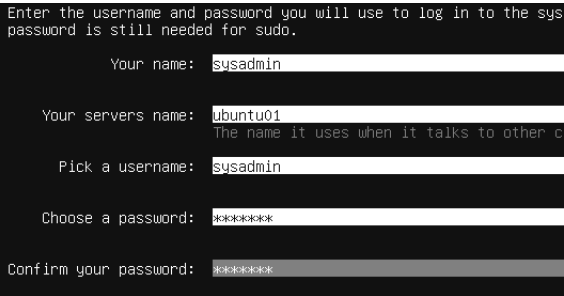
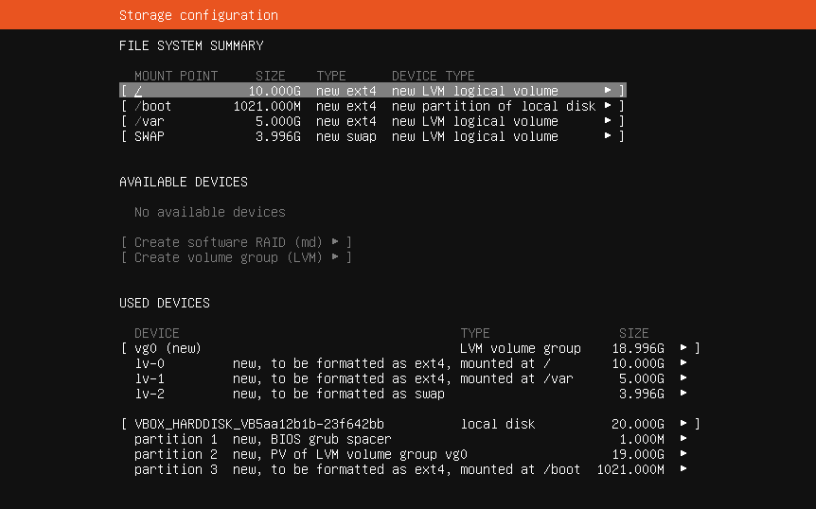
1: Configuración de los servidores .....	3
Configuración en virtualbox .....	3
Configuración de instalación y filesystem en Ubuntu01 .....	3
Configuración de instalación y filesystem en Bastion .....	4
Configuración de instalación y filesystem en centos01 .....	4
Configuración de red .....	5
Actualizar paquetes y generar clave ssh .....	6
Configuración de github .....	6
2: Configurar un archivo de inventario de Ansible .....	7
3: Ejecutar comandos ad-hoc .....	10
4: Crear y ejecutar playbook de Ansible .....	11
Complementos para los playbooks .....	11
Playbook para NFS .....	11
Playbook para hardening: .....	12

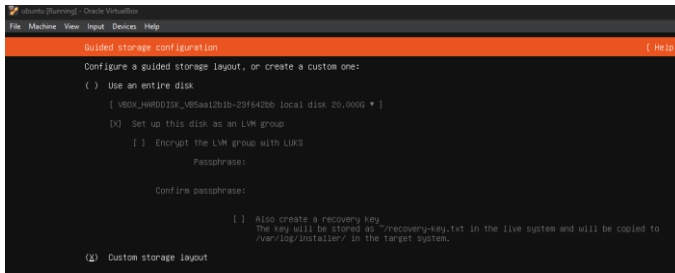
# 1:Configuración de los servidores

## Configuración en virtualbox

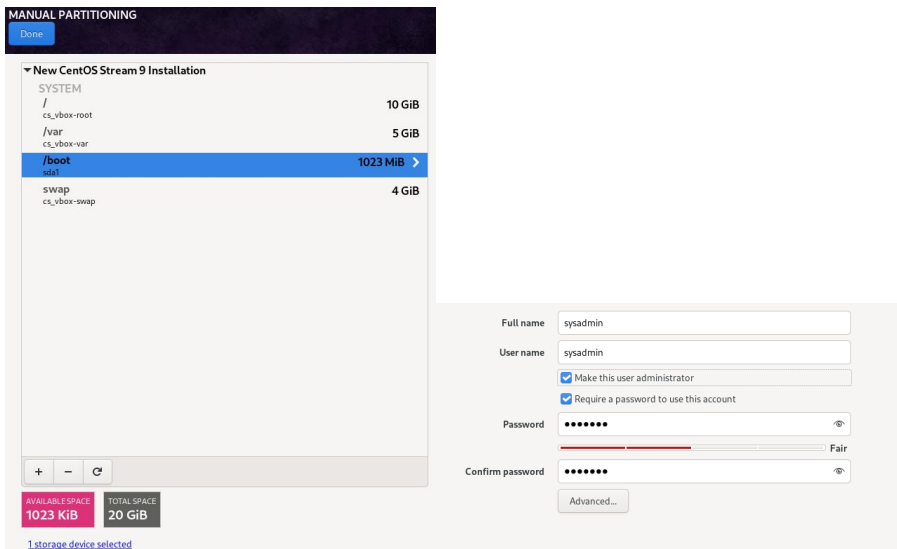


## Configuración de instalación y filesystem en Ubuntu01

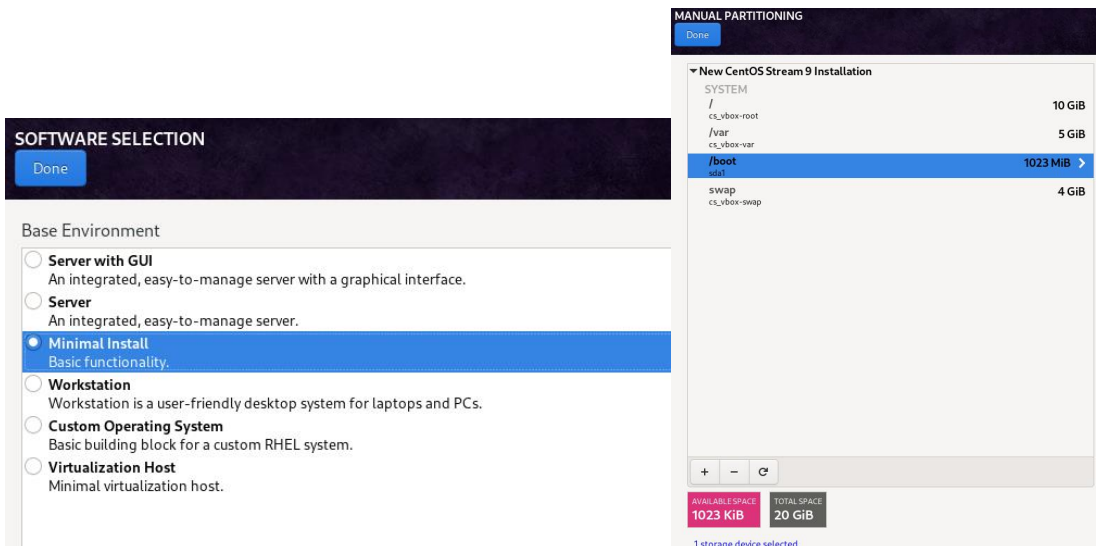




## Configuración de instalación y filesystem en Bastion



## Configuración de instalación y filesystem en centos01



## Configuración de red

### en ubuntu01:

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

Contenido:

```
network:

  version: 2

  ethernets:

    enp0s3:

      dhcp4: true

    enp0s8:

      dhcp4: false

      addresses: [192.168.1.2/24]
```

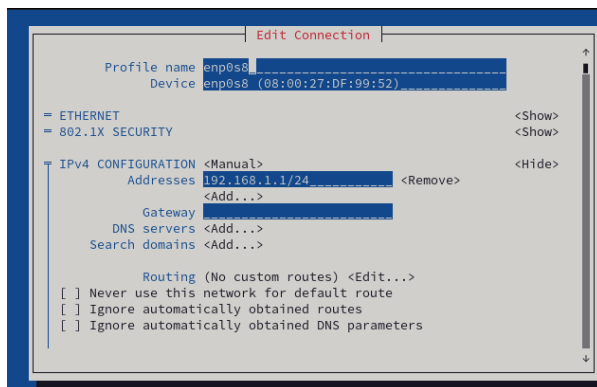
### Aplicar cambios:

```
sudo netplan apply
```

### Configuración de red en bastion:

Configuramos la red usando nmtui.

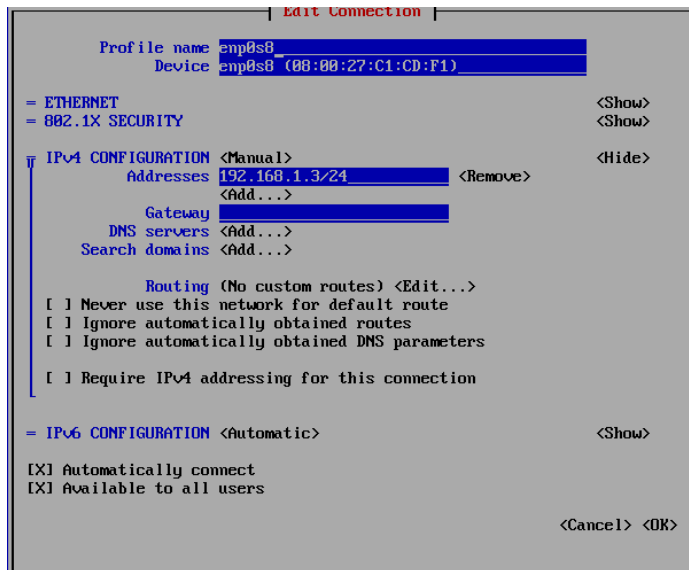
Asignándole la ip 192.168.1.1 a la interfaz de red que corresponde a la red interna.



### Configuración de red en centos01:

Configuramos la red usando nmtui.

Asignándole la ip 192.168.1.3 a la interfaz de red que corresponde a la red interna.



## Actualizar paquetes y generar clave ssh

### Actualizar paquetes:

En centos01 y bastion: `dnf update -y`

En ubuntu01: `apt update -y; apt upgrade -y`

### Copiar clave ssh a los servidores centos01 y ubuntu desde bastion:

`ssh-keygen` - generamos la clave ssh

`ssh-copy-id 192.168.1.2` - se copia la clave al ubuntu

`ssh-copy-id 192.168.1.3` - se copia la clave al centos

## Configuración de github

### Configuración del repositorio en GitHub y acceso desde bastion:

Creamos desde la web de Github un repositorio llamado TallerLinux2025.

Se agrega como colaborador al otro integrante del equipo del obligatorio, desde la pestaña "Settings" - "collaborators".

### Desde bastion nos fijamos nuestra clave pública ssh, del usuario sysadmin:

`cat /home/sysadmin/.ssh/id_rsa.pub`

En Github Configuramos "Deploy keys" en la pestaña "Security", donde asignamos las claves públicas ssh correspondientes, con permisos de read/write.

Desde bastion instalamos paquetes necesarios:

`sudo dnf install git`

**Desde bastion se clona el repositorio:**

```
git clone git@github.com:franc0lb/TallerLinux2025.git
```

**Configuramos username y email:**

```
git config --global user.name "$nombre"
```

```
git config --global user.email "$email"
```

**Verificar:**

```
git config --list
```

```
[sysadmin@bastion TallerLinux2025]$ git config --list
user.name=franc0lb
user.email=francolarrosa02@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=git@github.com:franc0lb/TallerLinux2025.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
```

## 2: Configurar un archivo de inventario de Ansible

```
mkdir TallerLinux2025/inventories
```

```
vi TallerLinux2025/inventories/inventory.ini
```

**Hacemos pruebas:**

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

```
[sysadmin@bastion TallerLinux2025]$ ansible-inventory -i inventories/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",
      "nfsserver"
    ]
  },
  "centos": {
    "hosts": [
      "centos01"
    ]
  },
  "linux": {
    "children": [
      "ubuntu",
      "centos"
    ]
  },
  "nfsserver": {
    "hosts": [
      "centos01"
    ]
  },
  "ubuntu": {
    "hosts": [
      "ubuntu01"
    ]
  }
}
```

```
ansible-inventory -i inventories/inventory.ini --graph
```

```
[sysadmin@bastion TallerLinux2025]$ ansible-inventory -i inventories/inventory.ini --graph
@all:
  |--@ungrouped:
  |--@linux:
  |   |--@ubuntu:
  |   |   |--ubuntu01
  |   |--@centos:
  |   |   |--centos01
  |--@nfsserver:
  |   |--centos01
```

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

```
[sysadmin@bastion TallerLinux2025]$ 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"
}
[sysadmin@bastion TallerLinux2025]$
```

**Ahora debemos guardar los cambios en el repositorio de Github, para eso ejecutamos:**

`git branch` - con este comando vemos en que rama estamos, debemos estar en main.

`git status` - vemos los cambios efectuados

`git add .` - agregamos los cambios

```
[sysadmin@bastion TallerLinux2025]$ git branch
* main
[sysadmin@bastion TallerLinux2025]$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ansible.cfg
    inventories/

nothing added to commit but untracked files present (use "git add" to track)
[sysadmin@bastion TallerLinux2025]$ git add .
[sysadmin@bastion TallerLinux2025]$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   ansible.cfg
    new file:   inventories/inventory.ini
[sysadmin@bastion TallerLinux2025]$
```

**Hacemos un commit:**

`git commit -m "Creé y configuré el archivo de inventario \ 'inventories/inventory.ini'"`



```
[sysadmin@bastion TallerLinux2025]$ git commit -m "Creé y configuré el archivo de inventario 'inventories/inventory.ini'"
[main 5550cd0] Creé y configuré el archivo de inventario 'inventories/inventory.ini'
 2 files changed, 19 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventories/inventory.ini
[sysadmin@bastion TallerLinux2025]$ git config --list
user.name=franc0lb
user.email=francolarrosa02@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=git@github.com:franc0lb/TallerLinux2025.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
```

**Enviamos cambios hacia el repositorio:**

*git push origin main*

```
[sysadmin@bastion TallerLinux2025]$ git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 566 bytes | 283.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:franc0lb/TallerLinux2025.git
   1b7704c..5550cd0  main -> main
[sysadmin@bastion TallerLinux2025]$
```

**Tambien Podemos agregar que tome este inventario como predeterminado:**

*vi ansible.cfg*

**Contenido:**

```
[defaults]

inventory=./inventories/inventory.ini
```

De esta manera toma de forma automática el inventario declarado anteriormente.

**Ejemplo de comando sin necesidad de declarar el inventario**

*ansible centos -m ping*

```
[sysadmin@bastion TallerLinux2025]$ ansible centos -m ping
centos01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
[sysadmin@bastion TallerLinux2025]$
```

Aplicamos y enviamos cambios al repositorio:

*git add .*

*git commit -m "Creé y configuré el archivo ansible.cfg"*

*git push origin main*

### 3: Ejecutar comandos ad-hoc

- Listar todos los usuarios en servidor Ubuntu

```
ansible ubuntu -a "cat /etc/passwd"
```

```
[sysadmin@Bastion TallerLinux2025]$ ansible ubuntu -a "cat /etc/passwd"
ubuntu01 | CHANGED | rc=0 >>
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:992:992:systemd Resolver:/:usr/sbin/nologin
pollinate:x:102:1::/var/cache/pollinate:/bin/false
polkitd:x:991:991:User for polkitd:/:usr/sbin/nologin
usbmux:x:103:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sysadmin:x:1000:1000:System Administrator:/home/sysadmin:/bin/bash
sshd:x:104:65534::/run/ssh:/usr/sbin/nologin
```

- Mostrar el uso de memoria en todos los servidores

```
ansible all -a "free -h"
```

```
[sysadmin@bastion TallerLinux2025]$ ansible all -a "free -h"
ubuntu01 | CHANGED | rc=0 >>
              total        used        free      shared  buff/cache   available
Mem:          1.9Gi         320Mi        1.6Gi         1.1Mi        203Mi        1.6Gi
Swap:          4.0Gi           0B         4.0Gi
centos01 | CHANGED | rc=0 >>
              total        used        free      shared  buff/cache   available
Mem:          1.7Gi         331Mi        1.3Gi         4.0Mi        170Mi        1.3Gi
Swap:          4.0Gi           0B         4.0Gi
```

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

```
Ansible centos -m package -a "name=chrony state=present" -Kb
```

#### Explicación:

con el -m indico que modulo estoy utilizando

-a define los argumentos

-kb es para ejecutar como administrador y que me pida la password,

que el resultado sea success significa que existe el servicio, el changed false y el msg nothing to do significan que estaba presente y no tuvo que cambiar nada.

```
[sysadmin@Bastion TallerLinux2025]$ ansible centos -m package -a "name=chrony state=present" -Kb
BECOME password:
centos01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Nothing to do",
  "rc": 0,
  "results": []
}
```

## 4: Crear y ejecutar playbook de Ansible

### Complementos para los playbooks

Se crea el archivo `collections/requirements.yaml`:

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

Instalamos complementos:

```
ansible-galaxy collection install -r
\ collections/requirements.yaml
```

```
[sysadmin@Bastion TallerLinux2025]$ ansible-galaxy collection 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-1.5.4.tar.gz to /home/sysadmin/.ansible/tmp/ansible-local-75964ux5wtdcm/tmpiwwlmtmt/ansible-posix-1.5.4-tmew8v_w
Installing 'ansible.posix:1.5.4' to '/home/sysadmin/.ansible/collections/ansible_collections/ansible/posix'
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-75964ux5wtdcm/tmpiwwlmtmt/community-general-11.1.2-ocysqis
ansible.posix:1.5.4 was installed successfully
Installing 'community.general:11.1.2' to '/home/sysadmin/.ansible/collections/ansible_collections/community/general'
community.general:11.1.2 was installed successfully
```

### Playbook para NFS

Se crea el archivo `playbooks/nfs_setup.yml`

Se ejecuta: `ansible-playbook playbooks/nfs_setup.yml -K`

Se comprueba que funcionó:

```
[sysadmin@Bastion TallerLinux2025]$ ansible-playbook playbooks/nfs_setup.yml -K
BECOME password:
PLAY [Instalación y configuración de nfs server en Centos] *****
TASK [Gathering Facts] *****
ok [centos01]
TASK [Asegurarse de que NFS está instalado] *****
ok [centos01]
TASK [Iniciar servicio] *****
ok [centos01]
TASK [Habilitar puerto 2049 tcp] *****
ok [centos01]
TASK [Habilitar puerto 2049 udp] *****
ok [centos01]
TASK [Crear directorio] *****
ok [centos01]
TASK [Modificar /etc/exports] *****
ok [centos01]
PLAY RECAP *****
centos01: 1 ok, 0 changed, 0 unreachable, 0 failed, 0 skipped, 0 removed, 0 ignored
```

Se guardan y envían los cambios efectuados al repositorio.

## Playbook para hardening:

Se crea *playbooks/hardening.yaml*

Se ejecuta: *ansible-playbook playbooks/hardening.yaml -K*

### Comprobación de que funcionó:

```
[sysadmin@Bastion TallerLinux2025]$ ansible-playbook playbooks/hardening.yaml -K
BECOME password:
[WARNING]: Collection community.general does not support Ansible version 2.14.18

PLAY [Hardening de servidores Ubuntu] *****
TASK [Gathering Facts] *****
ok: [ubuntu01]

TASK [Asegurarse de que UFW está instalado] *****
ok: [ubuntu01]

TASK [Habilitar y configurar ufw para permitir solo SSH] *****
ok: [ubuntu01]

TASK [Bloquear todo el tráfico entrante por defecto] *****
ok: [ubuntu01]

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

TASK [Sistema actualizado] *****
ok: [ubuntu01]

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

TASK [Inicio de sesion unicamente con llave publica] *****
ok: [ubuntu01]

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

TASK [Configurar fail2ban para bloquear intentos SSH] *****
ok: [ubuntu01]

PLAY RECAP *****
ubuntu01 : ok=10  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```