

**6 de agosto de 2024**

# Obligatorio TALER DE SERVIDORES LINUX

**Docente: Enrique Verdes**

---

**Autores:**

**Sergio Leonel Nicolás Giacusa N° 101906  
Matias Daniel Alvarez Ballarini N° 241314**

---

# INDICE

## Contenido

INDICE .....	1
INTRODUCCIÓN .....	2
ESCENARIO DE TRABAJO .....	2
PARTE A).....	3
Instalación y preparación de servidor controlador para poder utilizar Ansible .....	3
Instalación de los paquetes y librerías necesarias para utilizar Ansible y Git .....	4
Creamos y copiamos a los demás servidores la clave SSH: .....	6
PARTE B).....	10
PARTE C) .....	13
Creación de Playbooks para instalar, definir políticas y activar UFW en servidor Ubuntu: .....	13
Playbook para instalar JDK de Java .....	14
Playbook para instalar Tomcat.....	15
Playbook para instalar MariaDB .....	17
Playbook para instalar el Web Server.....	19
CONCLUSIONES GENERALES .....	20
Bibliografía y referencias .....	20

# INTRODUCCIÓN

En este trabajo nos enfocaremos en la preparación y configuración de un entorno automatizado utilizando Ansible para la gestión de servidores. Ansible es una herramienta poderosa para la automatización de la infraestructura de TI, que facilita la configuración, gestión y despliegue de aplicaciones en servidores de manera eficiente y repetible. A lo de la implementación, seguiremos el enfoque visto en clase para preparar y configurar los servidores, y utilizaremos Ansible para implementar y gestionar servicios en estos servidores.

## ESCENARIO DE TRABAJO

Necesitamos tener listo un servidor controlador, que será la máquina desde la cual ejecutaremos nuestros playbooks de Ansible.

Debe cumplir con los siguientes requisitos:

- Paquetes y Librerías: Debe tener instalados los paquetes y librerías necesarios para utilizar Ansible y Git.
- Autenticación SSH: El usuario que ejecutará los automatismos debe contar con sus claves pública/privada SSH para asegurar la comunicación segura entre los servidores.
- Repositorio de Código: Debe contar con un repositorio de código en Github o Gitlab para trabajar colaborativamente con el equipo.

Se instalarán dos servidores con las siguientes características y diseño de particiones:

- Disco de 13GB con el siguiente esquema de particiones:
- Partición de 1GB para /boot
- LVM de 7GB para /
- LVM de 3GB para /var
- LVM de 2GB para SWAP

Cada servidor tendrá una configuración de hardware básica de 1 CPU y 2 GB de RAM, y deberá contar con dos interfaces de red: una conectada a NAT y la otra a una red interna o Host-Only para permitir la conexión con el servidor controlador a través de Ansible.

Distribuciones para Instalar:

- Servidor Red Hat: Se sugiere utilizar CentOS Stream 8 o 9.
- Servidor Ubuntu: Utilizar la versión 24.04.

## PARTE A)

### Instalación y preparación de servidor controlador para poder utilizar Ansible

Instalamos Centos 9 GUI server con su configuración de disco y particiones:

```
sysadmin@controller:~$ sudo fdisk -l
Disk /dev/sda: 21 GiB, 22548578304 bytes, 44040192 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x8424cd0e

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sda1   *          2048    2099199    2097152     1G 83 Linux
/dev/sda2                2099200 44040191 41940992    20G 8e Linux LVM

Disk /dev/mapper/cs_10-root: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/cs_10-swap: 3.99 GiB, 4286578688 bytes, 8372224 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk /dev/mapper/cs_10-swap: 3.99 GiB, 4286578688 bytes, 8372224 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/cs_10-var: 3 GiB, 3221225472 bytes, 6291456 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/cs_10-home: 3 GiB, 3221225472 bytes, 6291456 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[sysadmin@controller ~]$
```

## Instalación de los paquetes y librerías necesarias para utilizar Ansible y Git

Primero debemos actualizar el sistema, aplicamos el siguiente comando para hacerlo:

```
sudo dnf update -y
```

```
selinux-policy-38.1.42-1.el9.noarch          selinux-policy-targeted-38.1.42-1.el9.noarch
shadow-utils-2:4.9-9.el9.x86_64             shadow-utils-subid-2:4.9-9.el9.x86_64
slirp4netns-1.3.1-1.el9.x86_64              sos-4.7.1-3.el9.noarch
sssd-2.9.5-4.el9.x86_64                     sssd-ad-2.9.5-4.el9.x86_64
sssd-client-2.9.5-4.el9.x86_64              sssd-common-2.9.5-4.el9.x86_64
sssd-common-pac-2.9.5-4.el9.x86_64           sssd-ipa-2.9.5-4.el9.x86_64
sssd-kcm-2.9.5-4.el9.x86_64                 sssd-krb5-2.9.5-4.el9.x86_64
sssd-krb5-common-2.9.5-4.el9.x86_64          sssd-ldap-2.9.5-4.el9.x86_64
sssd-proxy-2.9.5-4.el9.x86_64               sushi-3.38.1-3.el9.x86_64
systemd-252-38.el9.x86_64                   systemd-libs-252-38.el9.x86_64
systemd-pam-252-38.el9.x86_64                systemd-rpm-macros-252-38.el9.noarch
systemd-udev-252-38.el9.x86_64               tpm2-tss-3.2.3-1.el9.x86_64
tzdata-2024a-2.el9.noarch                   udisks2-2.9.4-11.el9.x86_64
udisks2-iscsi-2.9.4-11.el9.x86_64            udisks2-lvm2-2.9.4-11.el9.x86_64
usbutils-017-1.el9.x86_64                   vulkan-loader-1.3.283.0-1.el9.x86_64
webkit2gtk3-2.44.2-1.el9.x86_64             webkit2gtk3-jsc-2.44.2-1.el9.x86_64
wget-1.21.1-8.el9.x86_64                    wpa_supplicant-1:2.10-5.el9.x86_64
xfsprogs-6.4.0-3.el9.x86_64                 xorg-x11-server-Xorg-1.20.11-26.el9.x86_64
xorg-x11-server-Xwayland-23.2.7-1.el9.x86_64 xorg-x11-server-common-1.20.11-26.el9.x86_64
yum-4.14.0-15.el9.noarch
Installed:
composefs-1.0.3-2.el9.x86_64                 composefs-libs-1.0.3-2.el9.x86_64
grub2-tools-efi-1:2.06-82.el9.x86_64         grub2-tools-extra-1:2.06-82.el9.x86_64
kernel-5.14.0-480.el9.x86_64                 kernel-core-5.14.0-480.el9.x86_64
kernel-modules-5.14.0-480.el9.x86_64         kernel-modules-core-5.14.0-480.el9.x86_64
keyutils-1.6.3-1.el9.x86_64                  ledmon-libs-1.0.0-1.el9.x86_64
libdecor-0.1.1-1.el9.x86_64                  passt-0^20240624.glee2eca-1.el9.x86_64
passt-selinux-0^20240624.glee2eca-1.el9.noarch
Complete!
[sysadmin@controller ~]$
[sysadmin@controller ~]$
```

Ahora instalamos GIT:

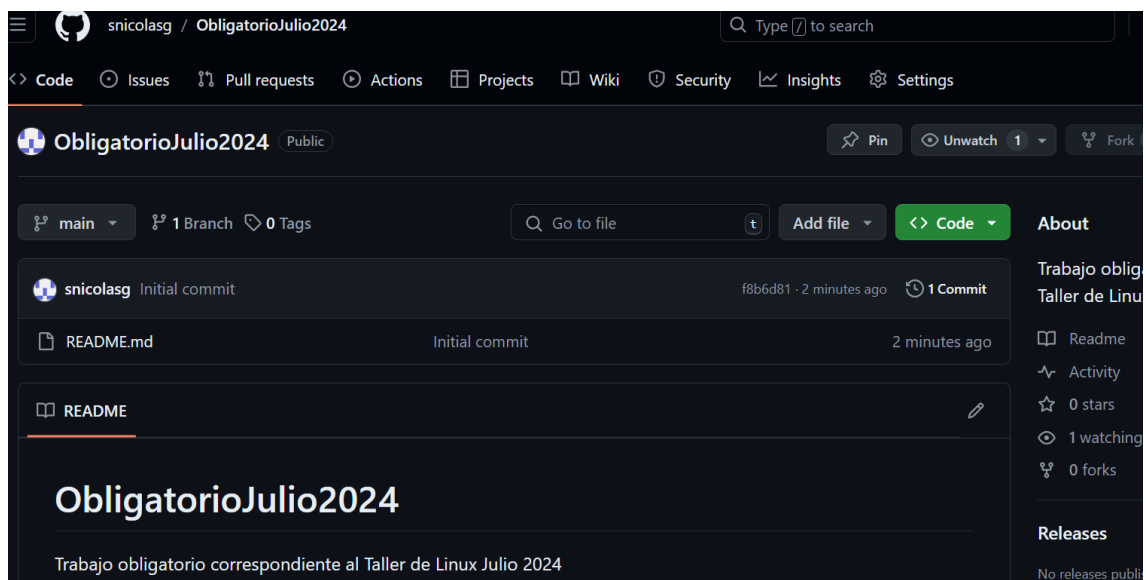
```
$ sudo dnf install git
```

Podemos verificar la versión instalada con el comando:

```
git --version
```

```
sysadmin@controller:~  
[sysadmin@controller ~]$ git --version  
git version 2.43.5  
[sysadmin@controller ~]$
```

Hemos creado en GitHub nuestro repositorio de trabajo:



Ahora necesitamos configurar el ambiente Git con los siguientes comandos:

Nombre de usuario:

```
$ git config --global user.name "Nicolas - Alvarez"
```

Dirección de correo electrónico:

```
$ git config --global user.email "sergiong1976@gmail.com"
```

Colores en los comandos de git en la consola:

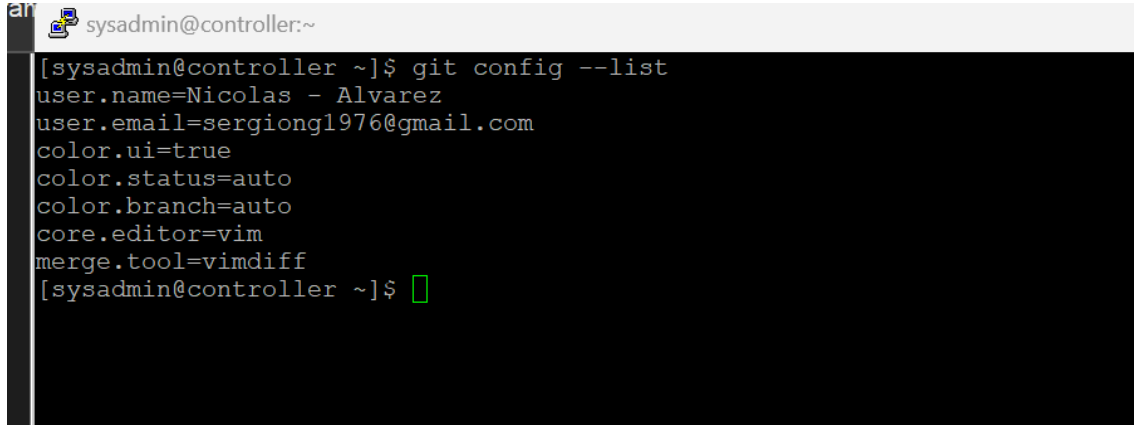
```
$ git config --global color.ui true  
$ git config --global color.status auto $ git config --global
```

Editor de textos:

```
$ git config --global core.editor vim
```

Herramienta para merges:

```
$ git config --global merge.tool vimdiff
```

A terminal window with a light gray title bar showing 'sysadmin@controller:~'. The terminal output of 'git config --list' is as follows:

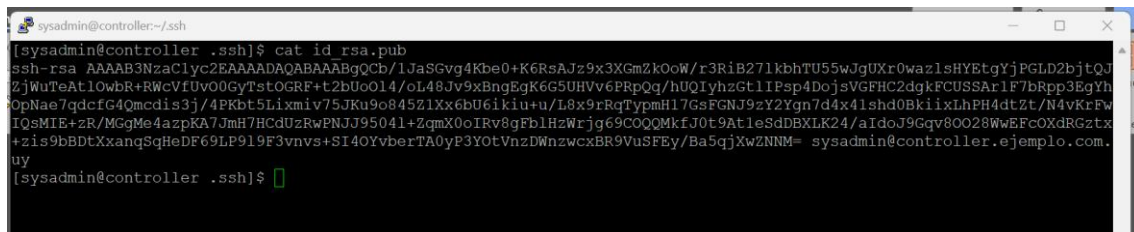
```
[sysadmin@controller ~]$ git config --list
user.name=Nicolas - Alvarez
user.email=sergiong1976@gmail.com
color.ui=true
color.status=auto
color.branch=auto
core.editor=vim
merge.tool=vimdiff
[sysadmin@controller ~]$
```

## Creamos y copiamos a los demás servidores la clave SSH:

Creamos la clave SSH con los siguientes comandos:

```
$ ssh-keygen
```

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

A terminal window with a light gray title bar showing 'sysadmin@controller:~/ssh'. The terminal output of 'cat id\_rsa.pub' is as follows:

```
[sysadmin@controller .ssh]$ cat id_rsa.pub
ssh-rsa AAAAB3NzaClyc2EAAAADAQABAAQgCbl1JaSGvg4Kbe0+K6RsAJz9x3XGmZk0oW/r3Rib271kbhTU55wJgUXr0waz1sHYEtgyjPGLD2bjtQJ
ZjWuTeAt1owBR+RWcVfUv00GyTstOGRF+t2bUo014/oL48Jv9xBngEgK6G5UHVv6PRpQg/hUQIyhzGt1IPsp4DojsVGFHC2dgkFCUSSAr1F7bRpp3EgYh
OpNae7qdcfG4Qmcdis3j/4PKbt5Lixmiv75JKu9o845Z1Xx6bU61kiu+u/L8x9rRqTypmH17GsFGNJ9zY2Ygn7d4x41shd0BkiiXhPH4dtZt/N4vKrFw
IQsMIE+zR/MGqMe4azpKA7JmH7HCdUzRwPNJJ95041+ZqmX0oIRv8gFb1HzWrjg69C0QOMkfJ0t9AtleSdDBXLK24/aIdoJ9Gqv80O28WwEFcoXDRGztX
+zis9bBDtXxangSqHeDF69LP919F3vnvs+SI40YvberTA0yP3Y0tVnZDwnzwcxBR9VuSFey/Ba5qjXwZNNM= sysadmin@controller.ejemplo.com.
uy
[sysadmin@controller .ssh]$
```

Ahora una vez creada la clave, debemos copiarla al server02 y server01. Esto lo hacemos de la siguiente manera:

```
$ ssh-copy-id 192.168.56.20
$ ssh-copy-id 192.168.56.102
```

```
[sysadmin@controller ~]$ ssh-copy-id 192.168.56.20
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sysadmin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
sysadmin@192.168.56.20's password:
Permission denied, please try again.
sysadmin@192.168.56.20's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '192.168.56.20'"
and check to make sure that only the key(s) you wanted were added.

[sysadmin@controller ~]$
```

```
sysadmin@controller:~
[sysadmin@controller ~]$ ssh-copy-id 192.168.56.102
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sysadmin/.ssh/id_rsa.pub"
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:x3gXXz5ais96ClxnMfeTqq6ppNG+RMk0+uLyz6R3cpM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
sysadmin@192.168.56.102's password:

Number of key(s) added: 1



Now try logging into the machine, with:  "ssh '192.168.56.102'"
and check to make sure that only the key(s) you wanted were added.

[sysadmin@controller ~]$
```

Esta clave SSH la vamos a copiar en nuestro GitHub

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

### Authentication keys

	<b>Taller</b> SHA256:ddPeMJOM+KzCC3nxJTvgNGRZrpykt/ZV0WmohbBzIjI Added on Jul 30, 2024 Last used within the last week — Read/write	<a href="#">Delete</a>
	<b>ObligatorioJulio2024</b> SHA256:u58N3/cOG8fVG+eKyo5awRzcVBLu9LLKNMfViPm4WZE Added on Jul 31, 2024 Never used — Read/write	<a href="#">Delete</a>

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

Clonamos ahora nuestro repositorio de GitHub utilizando la clave SSH antes cargada:

```
$ git clone git@github.com:snicolasg/ObligatorioJulio2024.git
```



```
sysadmin@controller:~  
[sysadmin@controller ~]$ git clone git@github.com:snicolasg/ObligatorioJulio2024.git  
Cloning into 'ObligatorioJulio2024'...  
The authenticity of host 'github.com (140.82.113.4)' can't be established.  
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (3/3), done.  
[sysadmin@controller ~]$
```

Tenemos el usuario sysadmin NO root con permisos para ejecutar comando de administrador.

```
sysadmin@controller:~  
[sysadmin@controller ~]$ groups sysadmin  
sysadmin : sysadmin wheel  
[sysadmin@controller ~]$ sudo -l -U sysadmin  
Matching Defaults entries for sysadmin on controller:  
    !visiblepw, always set home, match group by gid, always query group plugin, env_reset, env_keep="COLORS DISPLAY  
    HOSTNAME HISTSIZE KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",  
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC  
    LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",  
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin  
  
User sysadmin may run the following commands on controller:  
    (ALL) ALL  
[sysadmin@controller ~]$
```

Ahora procedemos a instalar Ansible aplicando los siguientes comandos:

```
$ sudo dnf install pip  
$ pip install pipx  
$ pipx ensurepath  
$ pipx install ansible-core
```

```
sysadmin@controller:~  
[sysadmin@controller ~]$ pipx install ansible-core  
installed package ansible-core 2.15.12, installed using Python 3.9.19  
These apps are now globally available  
- ansible  
- ansible-config  
- ansible-connection  
- ansible-console  
- ansible-doc  
- ansible-galaxy  
- ansible-inventory  
- ansible-playbook  
- ansible-pull  
- ansible-test  
- ansible-vault  
done! ✨ ✨ ✨  
[sysadmin@controller ~]$
```

Aplicamos los siguientes comandos para configurar Ansible:

```
$ pipx inject ansible-core argcomplete  
$ pipx inject ansible-core ansible-lint  
$ activate-global-python-argcomplete --user  
  
$ source /home/sysadmin/.bash_completion
```

Comprobamos funcionamiento de ansible aplicando:

```
$ ansible -i 192.168.56.20, all -m ping
```

```
sysadmin@controller:~  
[sysadmin@controller ~]$ ansible -i 192.168.56.20, all -m ping  
192.168.56.20 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
[sysadmin@controller ~]$
```

## PARTE B)

Preparamos dos nuevos servidores con los requerimientos solicitados. En uno instalamos Centos 9.0 server con su configuración de disco y particiones:

```
[sysadmin@server01 ~]$ sudo fdisk -l

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for sysadmin:

Disco /dev/sda: 13 GiB, 13958643712 bytes, 27262976 sectores
Modelo de disco: VBOX HARDDISK
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0xf59a4c81

Disposit.  Inicio Comienzo      Final Sectores Tamaño Id Tipo
/dev/sda1  *          2048  2099199   2097152     1G 83 Linux
/dev/sda2             2099200 27262975 25163776    12G 8e Linux LVM

Disco /dev/mapper/cs_server01-root: 7 GiB, 7516192768 bytes, 14680064 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes

Disco /dev/mapper/cs_server01-swap: 1,99 GiB, 2139095040 bytes, 4177920 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes

Disco /dev/mapper/cs_server01-var: 3 GiB, 3221225472 bytes, 6291456 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
[sysadmin@server01 ~]$ _
```

Tenemos el usuario sysadmin NO root con permisos para ejecutar comando de administrador.

```
sysadmin@server01:~
[sysadmin@server01 ~]$ groups sysadmin
sysadmin : sysadmin wheel
[sysadmin@server01 ~]$ sudo -l -U sysadmin
[sudo] password for sysadmin:
Matching Defaults entries for sysadmin on server01:
!visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY", secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User sysadmin may run the following commands on server01:
(ALL) ALL
[sysadmin@server01 ~]$
```

Para el otro instalamos Ubuntu Server con su configuración de disco y particiones:

```
sysadmin@server02:~$ sudo fdisk -l
[sudo] password for sysadmin:
Disk /dev/sda: 13 GiB, 13958643712 bytes, 27262976 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 5536581B-9B17-4987-AC83-8C5968974D38

Device        Start      End  Sectors  Size Type
/dev/sda1      2048      4095     2048    1M BIOS boot
/dev/sda2      4096    2101247   2097152    1G Linux filesystem
/dev/sda3    2101248  27260927  25159680   12G Linux filesystem

Disk /dev/mapper/vg0-root: 7 GiB, 7516192768 bytes, 14680064 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/vg0-var: 3 GiB, 3221225472 bytes, 6291456 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/vg0-swap: 2 GiB, 2143289344 bytes, 4186112 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
sysadmin@server02:~$
```

Tenemos el usuario sysadmin NO root con permisos para ejecutar comando de administrador.

```
sysadmin@server02: ~
sysadmin@server02:~$ groups sysadmin
sysadmin : sysadmin adm cdrom sudo dip plugdev lxd
sysadmin@server02:~$ sudo -l -U sysadmin
[sudo] password for sysadmin:
Matching Defaults entries for sysadmin on server02:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User sysadmin may run the following commands on server02:
    (ALL : ALL) ALL
sysadmin@server02:~$
```

Le cambiamos a IP fija al servidor Ubuntu y le colocamos la IP 192.168.56.20

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

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses:
        - 192.168.56.20/24
      gateway4: 192.168.56.254
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
  version: 2
~
~
~
```

```
root@server02: /etc/netplan
root@server02:/etc/netplan# 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 pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:ec:2a:78 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 86205sec preferred_lft 86205sec
    inet6 fe80::a00:27ff:feec:2a78/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:84:44:3c brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.20/24 brd 192.168.56.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe84:443c/64 scope link
        valid_lft forever preferred_lft forever
root@server02:/etc/netplan#
```

## PARTE C)

## Creación de Playbooks para instalar, definir políticas y activar UFW en servidor Ubuntu:

## Instalamos los requerimientos:

```
$ ansible-galaxy collection install -r collections/requirements.yml
```

```
[sysadmin@controller~]$ ansible-galaxy collection install -r collections/requirements.yml
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-9043lq56im09/tmp174cvhxj/ansible-posix-1.5.4-vg3 ces1
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-9.2.0.tar.gz to /home/sysadmin/.ansible/tmp/ansible-local-9043lq56im09/tmp174cvhxj/community-general-9.2.0-715dcmpr
ansible.posix:1.5.4 was installed successfully
Installing 'community.general:9.2.0' to '/home/sysadmin/.ansible/collections/ansible_collections/community/general'
community.general:9.2.0 was installed successfully
[sysadmin@controller ObligatorioJulio2024]$
```

## Ejecutamos el playbook:

```
ansible-playbook playbooks/ubuntu_firewall_ufw.yml --ask-become-pass
```

```

sysadmin@controller-~/ObligatorioJulio2024
PLAY [Configurar UFW en un servidor Ubuntu] *****
TASK [Gathering Facts] *****
ok: [server02]

TASK [UFW instalado] *****
ok: [server02]

TASK [Permitir puerto 22 en ufw] *****
changed: [server02]

TASK [Defino politicas de tráfico entrante] *****
changed: [server02]

TASK [Defino politicas de tráfico entrante] *****
ok: [server02]

TASK [servicio UFW levantado y activo] *****
ok: [server02]

TASK [Verificar el estado de UFW] *****
changed: [server02]

TASK [Mostrar estado de UFW] *****
ok: [server02] => {
    "msg": "Status: active\n\nTo          Action      From\n--          -\nALLOW       Anywhere    \nOpenSSH     (v6)        ALLOW       Anywhere   (v6)\n                                     ----\n                                     \nOpenSSH\n                                     "
}

PLAY RECAP *****
server02 : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Comprobamos en servidor Ubuntu:

```
sudo ufw status verbose
```

```
sysadmin@server02: ~  
sysadmin@server02:~$ sudo ufw status verbose  
Status: active  
Logging: on (low)  
Default: deny (incoming), allow (outgoing), disabled (routed)  
New profiles: skip  
  
To Action From  
--  
22/tcp (OpenSSH) ALLOW IN Anywhere  
22/tcp (OpenSSH (v6)) ALLOW IN Anywhere (v6)  
  
sysadmin@server02:~$
```

## Playbook para instalar JDK de Java

En <https://openjdk.org/install/> reviso la última versión de JDK

Ejecutamos el playbook de la siguiente manera:

```
$ ansible-playbook playbooks/install_Java.yml --ask-become-pass
```

```
sysadmin@controller~/ObligatorioJulio2024  
[sysadmin@controller ObligatorioJulio2024]$ ansible-playbook playbooks/install_Java.yml --ask-become-pass  
BECOME password:  
  
PLAY [Instalar JDK de Java en CentOS] *****  
TASK [Gathering Facts] *****  
ok: [server01]  
  
TASK [Actualizar el repositorio EPEL] *****  
changed: [server01]  
  
TASK [Instalo JDK de Java] *****  
changed: [server01]  
  
TASK [Verifica instalación Java] *****  
changed: [server01]  
  
TASK [Muestra version Java] *****  
ok: [server01] => {  
  "java_version.stdout_lines": []  
}  
  
PLAY RECAP *****  
server01 : ok=5 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0  
  
[sysadmin@controller ObligatorioJulio2024]$
```

En servidor1

```
sysadmin@server01:~  
[sysadmin@server01 ~]$ java -version  
openjdk version "1.8.0_362"  
OpenJDK Runtime Environment (build 1.8.0_362-b08)  
OpenJDK 64-Bit Server VM (build 25.362-b08, mixed mode)  
[sysadmin@server01 ~]$
```

## Playbook para instalar Tomcat

```
$ sudo yum install tar
```

Ejecutamos el playbook

```
$ ansible-playbook playbooks/install_tomcat.yml --ask-become-pass
```

```
TASK [Descargo el paquete tomcat] *****
changed: [server01]

TASK [Descomprimo el archivo de tomcat] *****
changed: [server01]

TASK [Seteo variable con ruta donde está el tomcat] *****
ok: [server01]

TASK [Creo enlace a tomcat] *****
changed: [server01]

TASK [Crear usuario y grupo para Tomcat] *****
ok: [server01]

TASK [Crear usuario y grupo para Tomcat] *****
ok: [server01]

TASK [Establecer permisos en el directorio de Tomcat] *****
changed: [server01]

TASK [Configurar Tomcat como servicio] *****
changed: [server01]

TASK [Desplegar la aplicación todo.war] *****
changed: [server01]

TASK [Asegurarse de que el archivo app.properties existe] *****
changed: [server01]

TASK [Configurar el acceso a la base de datos en app.properties] *****
changed: [server01]

TASK [Iniciar y habilitar Tomcat] *****
changed: [server01]

PLAY RECAP *****
server01 : ok=14  changed=9  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

```
$ ansible-playbook playbooks/open_port_tomcat.yml --ask-become-pass
```

```
sysadmin@controller: ObligatorioJulio2024]$ ansible-playbook playbooks/open_port_tomcat.yml --ask-become-pass
BECOME password:

PLAY [Habilitar el puerto 8080 en el firewall de Centos] *****

TASK [Gathering Facts] *****
ok: [server01]

TASK [Comprobar que el firewall este levantado] *****
ok: [server01]

TASK [Agregamos la regla de firewall de tomcat (puerto 8080)] *****
ok: [server01]

PLAY RECAP *****
server01 : ok=3  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

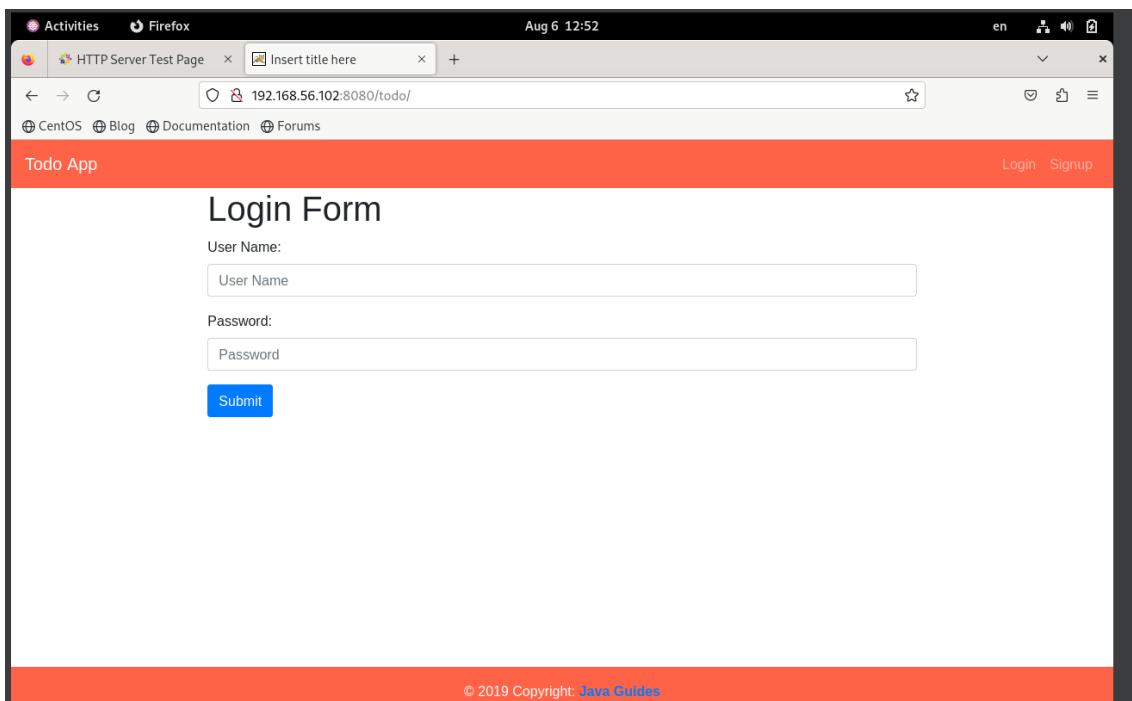


```
root@server01:~# firewall-cmd --list-port
8080/tcp
root@server01:~#
```

```
root@server01:~# sudo systemctl status tomcat
● tomcat.service
   Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; preset: disabled)
   Active: active (running) since Mon 2024-08-05 13:44:57 -03; 8min ago
     Process: 4091 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
    Main PID: 4107 (java)
      Tasks: 29 (limit: 11097)
     Memory: 226.1M
        CPU: 10.111s
    CGroup: /system.slice/tomcat.service
            └─4107 /usr/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.catalina.logging.LoggingManager

ago 05 13:44:57 server01 systemd[1]: Starting tomcat.service...
ago 05 13:44:57 server01 startup.sh[4091]: Existing PID file found during start.
ago 05 13:44:57 server01 startup.sh[4091]: Removing/clearing stale PID file.
ago 05 13:44:57 server01 startup.sh[4091]: Tomcat started.
ago 05 13:44:57 server01 systemd[1]: Started tomcat.service.
lines 1-16/16 (END)
```

Accedemos a la aplicación ToDo.war: 192.168.56.102:8080/todo



Accedemos a Signup y podemos registrar un usuario sin inconvenientes:

The screenshot shows a web browser window with the title 'Activities Firefox' and the date 'Aug 6 12:58'. The address bar displays '192.168.56.102:8080/todo/register/register.jsp'. The page has an orange header with 'Todo App' on the left and 'Login Signup' on the right. The main content area is titled 'User Register Form' and contains a green rectangular box at the top. Below this, there are four input fields: 'First Name', 'Last Name', 'User Name', and 'Password'. A blue 'Submit' button is located at the bottom of the form. The footer of the page shows '© 2019 Copyright: Todo App'.

## Playbook para instalar MariaDB

Primero Instalamos los requerimientos en el servidor Centos:

```
$ sudo dnf install mysql -y
$ ansible-galaxy collection install -r collections/requirements.yml
$ pipx inject ansible-core pymysql
```

En el servidor Ubuntu aplicamos:

```
$ sudo apt-get install python3-pymysql
$ pip3 show PyMySQL
```

Ahora ejecutamos el playbook:

```
$ ansible-playbook playbooks/install_database.yml --ask-become-pass
```

```
$ ansible-playbook playbooks/ubuntu_firewall_ufw.yml --ask-become-pass
```

Si la base de datos ya existe, no la vuelvo a crear:

# Playbook para instalar el Web Server

Ejecutamos el playbook para instalar Web Server:

```
$ ansible-playbook playbooks/install_webserver.yml --ask-become-pass
```

```
sysadmin@controller:~/ObligatorioJulio2024
[sysadmin@controller ObligatorioJulio2024]$ ansible-playbook playbooks/install_webserver.yml --ask-become-pass
BECOME password:
PLAY [Instalar y configurar un webserver] *****
TASK [Gathering Facts] *****
ok: [server01]
TASK [Instalar apache] *****
ok: [server01]
TASK [Configurar virtualhost] *****
changed: [server01]
TASK [Crear directorio document root] *****
ok: [server01]
TASK [Copiar pagina indice del sitio] *****
ok: [server01]
TASK [Permito conexiones al puerto 80] *****
ok: [server01] => (item=http)
ok: [server01] => (item=https)
TASK [Apache levantado y habilitado] *****
changed: [server01]
RUNNING HANDLER [Reiniciar apache] *****
changed: [server01]
PLAY RECAP *****
server01 : ok=8  changed=3  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

[sysadmin@controller ObligatorioJulio2024]$
```

```
httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
Active: active (running) since Fri 2024-08-02 03:04:33 -03; 18min ago
Docs: man:httpd.service(8)
Main PID: 25654 (httpd)
Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
Tasks: 177 (limit: 11097)
Memory: 24.0M
CPU: 6.203s
CGroup: /system.slice/httpd.service
└─25654 /usr/sbin/httpd -DFOREGROUND
   └─25656 /usr/sbin/httpd -DFOREGROUND
      └─25657 /usr/sbin/httpd -DFOREGROUND
         └─25658 /usr/sbin/httpd -DFOREGROUND
            └─25659 /usr/sbin/httpd -DFOREGROUND

ago 02 03:04:25 server01 systemd[1]: Starting The Apache HTTP Server...
ago 02 03:04:27 server01 httpd[25654]: AH00558: httpd: Could not reliably determine the server's fully qualified domain
ago 02 03:04:33 server01 httpd[25654]: Server configured, listening on: port 80
ago 02 03:04:33 server01 systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
```

# CONCLUSIONES GENERALES

Al finalizar este trabajo podemos concluir que hemos quedado muy satisfechos con lo realizado y los resultados obtenidos en el transcurso del mismo. Desde el inicio se trabajó de forma ordenada y conociendo bien cada punto y paso a ejecutar, esto nos brindó la tranquilidad de llegar en tiempo y forma con el objetivo cumplido de un trabajo completo y de calidad.

Finalmente coincidimos que este es un trabajo que nos ha servido mucho en el aprendizaje y fortalecimiento de todos los conceptos dados en la materia de Servidores Linux, además de aprender nuevos conceptos y adquirir otros conocimiento gracias al repaso y puesta en práctica de los mismos nos ha dado mucha tranquilidad en nuestro conocimiento y aprendizaje que para el futuro sin duda aplicaremos.

## Bibliografía y referencias

Material del curso (clases, grabaciones y PDFs).

Como programar un debug en Ansible:

<https://chatgpt.com/share/ad599acc-f239-4317-b416-6d5901df7577>

Como hacer que Tomcat se inicie como servicio

<https://chatgpt.com/share/e4f2e458-fb09-4d2b-b8da-2d88d867b615>

<https://www.redhat.com/es/blog/system-administrators-guide-getting-started-ansible-fast>

Mariadb

<https://chatgpt.com/share/8664ecfb-95a6-4dcf-8d9d-5572a81c1524>

