

# OBLIGATORIO

## Taller de Servidores Linux



**Integrantes:** Diego Vazquez CI: 4.874.410-9 / N° de estudiante: 302392  
Giovanni Storti CI: 4.000.995-1 / N° de estudiante: 151234

**Docente:** Enrique Verdes

**Fecha de entrega:** 27/02/24

## Contenido

Instalación y configuración de los servidores.....	3
ServerA .....	3
Instalación del SO .....	3
ServerB .....	4
Instalación del SO .....	4
Guithub.....	5
Ansible .....	6
Playbooks .....	6
initial_config.yml .....	7
hosts_updates.yml .....	8
web_servers.yml.....	9
reverse_proxy.yml .....	10
Ejecución de los Playbook: .....	11
initial_config.yml .....	11
hosts_updates.yml .....	13
web_servers.yml .....	14
reverse_proxy.yml .....	14
Pruebas de acceso a la Web y Proxy .....	16
Bibliografía .....	17

## Instalación y configuración de los servidores

Como pide la letra se crean 2 servidores ServerA y ServerB, uno con sistema operativo Rocky Linux 8 y otro con Ubuntu 22.04 respectivamente.

Se crearon volúmenes lógicos de la siguiente manera:

- 5GB para /
- 3 GB para /var
- 2 GB para /home
- 2 GB para swap
- 1 GB para boot

Quedando los discos en un total de entre 13 y 14GB.

Se agregan las 2 interfaces de red, 1 conectada a NAT y la otra a una red Interna que le permita conectarse al equipo bastión con Ansible.

Mediante uno de los playbooks se crea el usuario Ansible en cada equipo, con permisos con SUDO sin contraseña. También se le copia la clave pública para conectarse sin contraseña.

### ServerA

#### Instalación del SO

Configuraciones generales:

Idioma del SO: Ingles

Idioma del teclado: Español Latinoamérica

Región: Montevideo

Tarjeta de red1: NAT

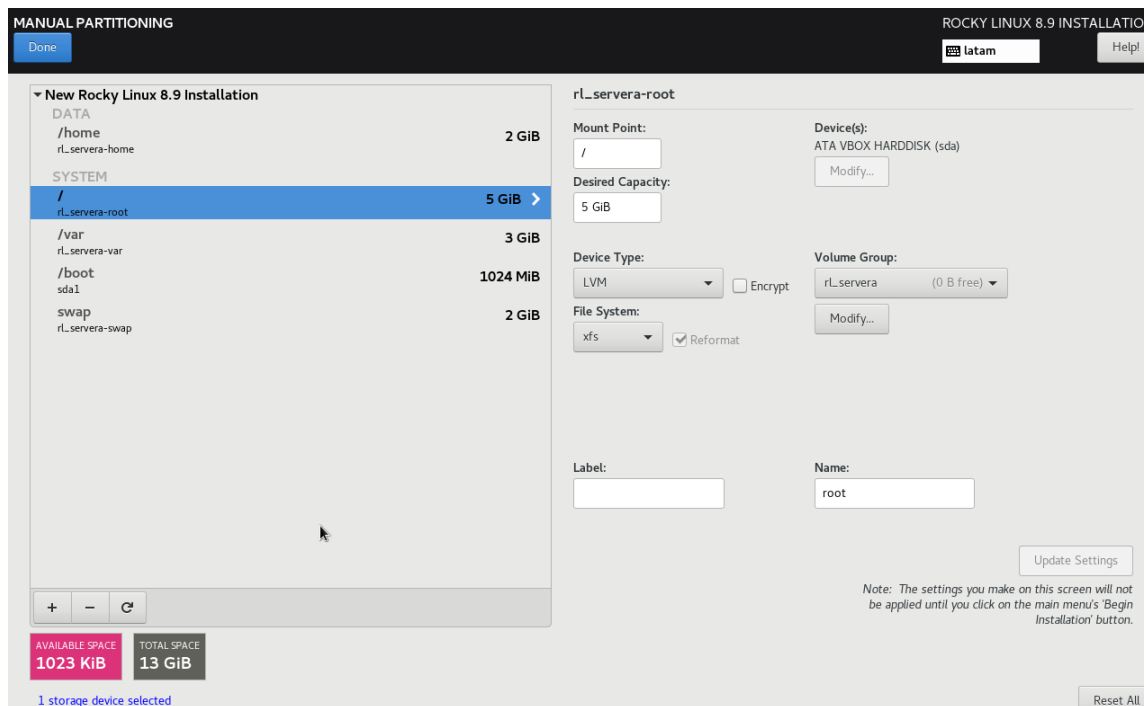
Tarjeta de red2: Adaptador sólo anfitrión IP: 192.168.56.10/24

Contraseña de Root

Usuario: sysadmin (con contraseña)

Instalación: Minimal

Particiones:



## ServerB

### Instalación del SO

Configuraciones generales:

Idioma del SO: Ingles

Instalación de actualizaciones: NO (se deja para después)

Idioma del teclado: Español Latinoamérica

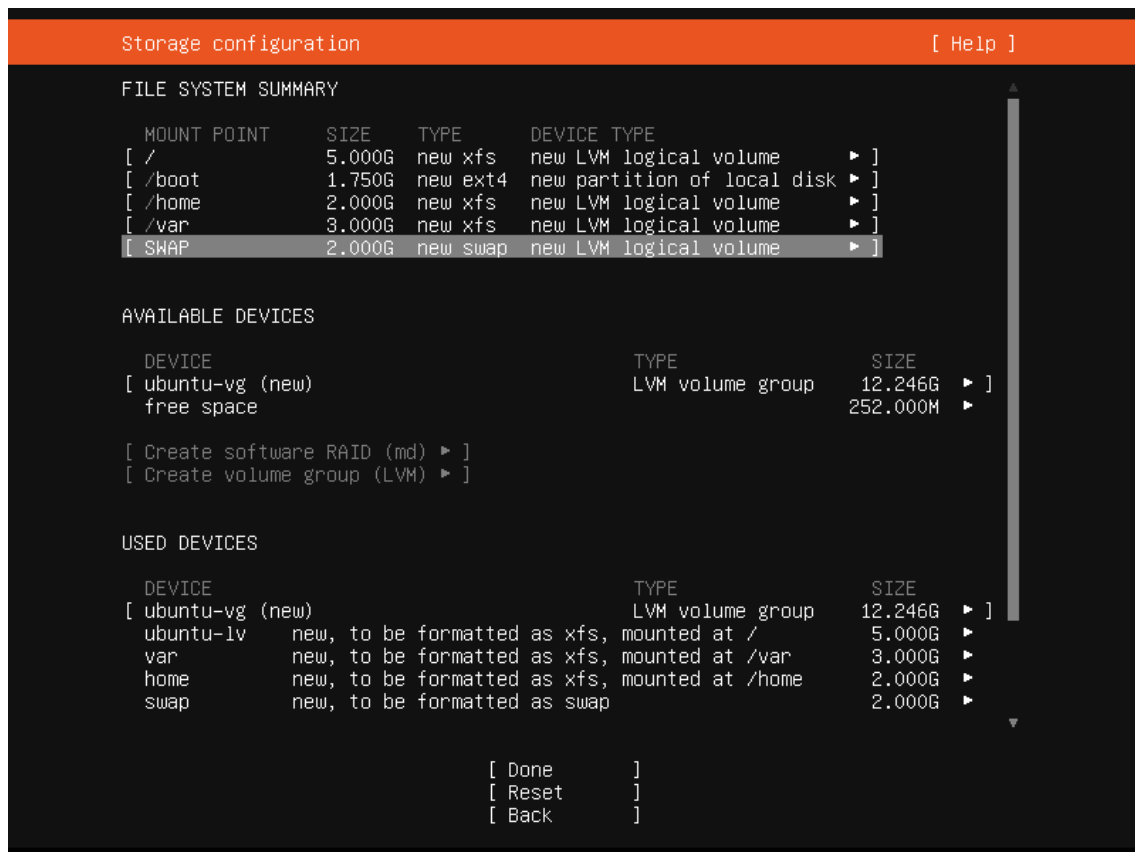
Instalación: Minimal

Tarjeta de red1: NAT

Tarjeta de red2: Adaptador sólo anfitrión IP: 192.168.56.20/24

Usuario: sysadmin (con contraseña)

Particiones:



Se instaló también un servidor Bastión para la creación, pruebas y ejecución de los playbook de Ansible.

## Guithub

Utilizamos el siguiente repositorio de GitHub para trabajar en el proyecto:

HTTPS: <https://github.com/gs151234/tallerfebrero2024.git>

SSH: <git@github.com:gs151234/tallerfebrero2024.git>

## Ansible

Creamos playbooks de Ansible para realizar las automatizaciones de las siguientes tareas:

- Actualizar los servidores
- Instalar OpenJDK y Tomcat en el servidor Rocky
  - o Teniendo en cuenta las configuraciones extra como habilitación de puertos en el firewall.
- Instalar y configurar un Proxy Reverso en el servidor Ubuntu
  - o Teniendo en cuenta las configuraciones extra como habilitación de puertos en el firewall.
- [Opcional] Creamos mediante Ansible el usuario “Ansible” con el que ejecutaremos las tareas.

Se instala Ansible, en este caso usamos un servidor como Bastión con Rocky Linux 8, por lo que lo hacemos mediante

```
# sudo dnf install ansible
```

Para utilizar Git, instalamos de la siguiente manera

```
# sudo dnf install git
```

Creamos un repositorio en GitHub, como se mencionó anteriormente y lo clonamos a nuestro bastión

```
# git clone git@github.com:gs151234/tallerfebrero2024.git
```

## Playbooks

Creamos 4 playbooks, donde el orden de ejecución siempre debe tener como primer playbook el **initial\_config** seguido de **hosts\_updates**, para que los SO estén asegurados a nivel de accesos y con las actualizaciones al día previo a la instalación de las aplicaciones.

Se crea una carpeta “inventories” con el archivo hosts que contiene a los equipos de la red.

Los agrupamos según la distribución, en este caso tenemos Rocky y Ubuntu. También establecimos un grupo Linux que engloba a todos los grupos.

```
[sysadmin@bastion tallerfebrero2024]$ cat inventories/hosts
[rocky]
serverA ansible_host=192.168.56.10

[ubuntu]
serverB ansible_host=192.168.56.20

[linux:children]
rocky
ubuntu
```

Especificaremos a continuación detalles de las tareas que realiza cada playbook

`initial_config.yml`

El primer paso es común a todos los playbooks, especificamos el objetivo, en este caso “linux” porque va a englobar a todos.

El usuario en este caso es sysadmin, porque es la primera conexión con el usuario administrador que creamos al instalar los equipos y con become escalamos a permisos con sudo.

- hosts: linux

user: sysadmin

become: true

Utilizamos la siguiente variable para ingresar sin que se revise la clave, ya que el paso siguiente es copiarla en el host.

vars:

ansible\_host\_key\_checking: false

tasks:

- name: Copiar clave pública a los nodos

Copia la llave pública al usuario sysadmin. Se utiliza el módulo `ansible.posix.authorized_key` que viene dentro de Ansible Packages.

- name: deshabilitar login de root por ssh

Se edita el archivo `sshd.config` para quitar el login de root por ssh.

- name: Deshabilitar login por password

Se edita el archivo `sshd.config` para quitar el login con contraseña por ssh.

- name: Crear usuario Ansible

- name: Convertir usuario Ansible en sudoer sin password

Se edita el archivo `/etc/sudoers` agregando la línea:

```
ansible ALL=(ALL) NOPASSWD=ALL
```

- name: Copiar clave publica a los nodos

Se copia la clave pública en el usuario ansible.

handlers:

- name: Reiniciar servidor ssh

hosts\_updates.yml

Especificamos el objetivo, en este caso "linux" porque va a englobar a todos.

El usuario en este caso es ansible, porque en el paso anterior ya creamos el usuario y le copiamos la clave para acceder sin contraseña por ssh. Escalemos a permisos con sudo, que tampoco necesitamos colocar contraseña.

- hosts: linux

user: ansible

become: true

En este caso las tareas actualizan los servidores llevando a la última versión de los paquetes de la distribución.

tasks:

- name: Update packages in Rocky Linux

Utilizamos "when:" para que la condición para instalar sea familia "RedHat"

- name: Update packages in Ubuntu Linux

Utilizamos "when:" para que la condición para instalar sea familia "Debian"

handlers:

- name: Reboot server



web\_servers.yml

Especificamos el objetivo, en este caso “rocky” porque seleccionamos este servidor para ser web server

El usuario es ansible, porque en el primer paso creamos el usuario y le copiamos la clave para acceder sin contraseña por ssh. Es escalamos a permisos con sudo, que tampoco necesitamos colocar contraseña.

- hosts: rocky

user: ansible

become: true

tasks:

- name: Instalar OpenJDK

Ejecuta los pasos de instalación

- name: Descargar Tomcat 8

Descarga de una url el archivo comprimido “apache-tomcat-8.5.73.tar.gz”

- name: Instalar tar y wget

Se instala tar para descomprimir el archivo “apache-tomcat-8.5.73.tar.gz”

Se instala wget

- name: Descomprimir Tomcat

Se descomprime en la carpeta /opt

- name: Crear grupo apache

- name: Crear usuario apache

Se crean usuario y grupo apache para darle los permisos sobre la carpeta en el siguiente paso.

- name: Establecer permisos

Se establecen los permisos sobre “/opt/apache-tomcat-8.5.73”

- name: Iniciar Tomcat

- name: Habilitar puerto 8080/tcp

Se habilita el puerto 8080/tcp para que el proxy pueda redirigir por este puerto a la aplicación.

- name: Copiar archivo tomcat.service al Tomcat

Se creó una carpeta “templates” que contiene el archivo “tomcat.service.j2” el cual es el servicio que iniciamos luego de instalado.

- name: Iniciar servicio tomcat

reverse\_proxy.yml

Especificamos el público, en este caso “ubuntu” porque seleccionamos este servidor para ser proxy reverso.

El usuario es ansible, porque en el primer paso creamos el usuario y le copiamos la clave para acceder sin contraseña por ssh. Es scalamos a permisos con sudo, que tampoco necesitamos colocar contraseña.

- hosts: ubuntu

user: ansible

become: true

La siguiente variable se utiliza para hacer referencia al archivo de configuración del sitio virtual (virtualhost)

vars:

http\_conf: 000-default.conf

tasks:

- name: Instalar Apache2

- name: Habilitar los módulos proxy y proxy\_httpd

Habilitamos los módulos para permitir que el servidor se configure como un servido proxy http.

- name: Configurar el archivo de sitio para el proxy reverso

Utilizamos un archivo creado en la carpeta “templates” llamado “httpd\_proxy.conf.j2” donde se encuentra la configuración del virtualhost.

- name: Habilitar sitio

Utilizamos el comando “a2ensite” para habilitar el sitio virtual.

- name: Instalar UFW

Instalamos el firewall.

- name: Permitir tráfico en los puertos 80 443 22

Habilitamos los puertos para permitir el tráfico antes de activar el firewall.

- name: Habilitar UFW

Se activa el firewall.

Se reinicia el SO para aplicar los cambios mediante un handler.

## Ejecución de los Playbook:

initial\_config.yml

```
# ansible-playbook -i inventories/hosts initial_config.yml --ask-pass --ask-become-pass
```

En este paso al no tener una clave publica en el usuario del servidor de destino, accedemos con la contraseña del administrador y la contraseña para escalar en permisos a SUDO.

```
[sysadmin@bastion tallerfebrero2024]$ ansible-playbook -i inventories/hosts initial_config.yml --ask-pass --ask-become-pass
SSH password:
BECOME password[defaults to SSH password]:

PLAY [linux] *****

TASK [Gathering Facts] *****
ok: [serverB]
ok: [serverA]

TASK [Copiar clave publica a los nodos] *****
changed: [serverB]
changed: [serverA]

TASK [Desabilitar login de root por ssh] *****
changed: [serverB]
changed: [serverA]

TASK [Desabilitar login por password] *****
changed: [serverB]
changed: [serverA]

TASK [Crear usuario Ansible] *****
changed: [serverB]
changed: [serverA]

TASK [Convertir usuario Ansible en sudoer sin password] *****
changed: [serverB]
changed: [serverA]

TASK [Copiar clave publica a los nodos] *****
changed: [serverB]
changed: [serverA]

RUNNING HANDLER [Reiniciar servidor ssh] *****
changed: [serverB]
changed: [serverA]

PLAY RECAP *****
serverA : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
serverB : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Se quita el permiso login de root por ssh en rocky

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Se quita autenticación con password por ssh rocky

```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication no
```

Se quitan permisos de root para acceder por ssh y los permisos para acceder con password en Ubuntu.

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server
PermitRootLogin no
PasswordAuthentication no
```

Creación del usuario ansible en rocky:

```
sysadmin:x:1000:1000:System Administrator:/home/sysadmin:/bin/bash
ansible:x:1001:1001::/home/ansible:/bin/bash
```

Creación del usuario ansible en ubuntu:

```
sysadmin:x:1000:1000:System Administrator:/home/sysadmin:/bin/bash
ansible:x:1001:1001::/home/ansible:/bin/sh
sysadmin@serverb:~$
```

Permisos SUDO sin contraseña en rocky

```
## Allows people in group wheel to run all commands
wheel ALL=(ALL) ALL

## Same thing without a password
# wheel ALL=(ALL) NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#include /etc/sudoers.d
ansible ALL=(ALL) NOPASSWD: ALL
```

Permisos SUDO sin contraseña en Ubuntu

```
# Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
ansible ALL=(ALL) NOPASSWD: ALL
sysadmin@serverb:~$
```

hosts\_updates.yml

\$ ansible-playbook -i inventories/hosts hosts\_updates.yml

Previo

```
Rocky Linux 8.9 (Green Obsidian)
Kernel 4.18.0-513.5.1.el8_9.x86_64 on an x86_64
```

```
Ubuntu 22.04.3 LTS serverb tty1

serverb login: sysadmin
Password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-94-generic x86_64)
```

Ejecutamos el playbook

```
[sysadmin@bastion tallerfebrero2024]$ ansible-playbook -i inventories/hosts hosts_updates.yml
PLAY [linux] *****
TASK [Gathering Facts] *****
ok: [serverB]
ok: [serverA]

TASK [Actualizaciones en Rocky Linux] *****
skipping: [serverB]
changed: [serverA]

TASK [Actualizaciones en Ubuntu Linux] *****
skipping: [serverA]
changed: [serverB]

RUNNING HANDLER [Reboot server] *****
changed: [serverB]
changed: [serverA]

PLAY RECAP *****
serverA : ok=3    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
serverB : ok=3    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Terminada las tareas

```
Rocky Linux 8.9 (Green Obsidian)
Kernel 4.18.0-513.18.1.el8_9.x86_64 on an x86_64
```

```
Ubuntu 22.04.4 LTS serverb tty1

serverb login: sysadmin
Password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-97-generic x86_64)
```

## web\_servers.yml

\$ ansible-playbook -i inventories/hosts web\_servers.yml

```
[sysadmin@bastion tallerfebrero2024]$ ansible-playbook -i inventories/hosts web_servers.yml
PLAY [Instalar OpenJDK y Tomcat 8 en Rocky Linux] *****
TASK [Gathering Facts] *****
ok: [serverA]

TASK [Instalar OpenJDK] *****
changed: [serverA]

TASK [Descargar Tomcat 8] *****
changed: [serverA]

TASK [Instalar tar y wget] *****
changed: [serverA] => (item=tar)
changed: [serverA] => (item=wget)

TASK [Descomprimir Tomcat] *****
changed: [serverA]

TASK [Crear grupo apache] *****
changed: [serverA]

TASK [Crear usuario apache] *****
changed: [serverA]

TASK [Establecer permisos] *****
changed: [serverA]

TASK [Iniciar Tomcat] *****
changed: [serverA]

TASK [Habilitar puerto 8080/tcp] *****
changed: [serverA]

TASK [Copiar archivo tomcat.service al Tomcat] *****
changed: [serverA]

TASK [Iniciar servicio tomcat] *****
changed: [serverA]

PLAY RECAP *****
serverA                : ok=12  changed=11  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

## Apache Tomcat

```
[sysadmin@servera ~]$ cd /opt
[sysadmin@servera opt]$ ls
apache-tomcat-8.5.73
[sysadmin@servera opt]$ _
```

## OpenJDK11

```
[sysadmin@servera opt]$ java --version
openjdk 11.0.22 2024-01-16 LTS
OpenJDK Runtime Environment (Red_Hat-11.0.22.0.7-1) (build 11.0.22+7-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-11.0.22.0.7-1) (build 11.0.22+7-LTS, mixed mode, sharing)
[sysadmin@servera opt]$ _
```

## reverse\_proxy.yml

\$ ansible-playbook -i inventories/hosts reverse\_proxy.yml

```
[sysadmin@bastion tallerfebrero2024]$ ansible-playbook -i inventories/hosts reverse_proxy.yml

PLAY [Configurar httpd como Proxy Reverso en Ubuntu] *****

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

TASK [Instalar Apache2] *****
changed: [serverB]

TASK [Habilitar los módulos proxy y proxy_httpd] *****
changed: [serverB] => (item=proxy)
changed: [serverB] => (item=proxy_httpd)

TASK [Configurar el archivo de sitio para el proxy reverso] *****
changed: [serverB]

TASK [Habilitar el sitio] *****
changed: [serverB]

TASK [Instalar UFW] *****
changed: [serverB]

TASK [Permitir tráfico en los puertos 80 443 22] *****
changed: [serverB] => (item=80)
changed: [serverB] => (item=443)
changed: [serverB] => (item=22)

TASK [Habilitar UFW] *****
changed: [serverB]

RUNNING HANDLER [Reiniciar apache2] *****
changed: [serverB]

RUNNING HANDLER [Reboot server] *****
changed: [serverB]

PLAY RECAP *****
serverB : ok=10  changed=9  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

## Apache2

```
sysadmin@serverb:~$ apache2 -v
Server version: Apache/2.4.52 (Ubuntu)
Server built: 2023-10-26T13:44:44
sysadmin@serverb:~$ _
```

## UFW

```
sysadmin@serverb:~$ sudo ufw status
[sudo] password for sysadmin:
Status: active

To Action From
--
80/tcp ALLOW Anywhere
443/tcp ALLOW Anywhere
22/tcp ALLOW Anywhere
80/tcp (v6) ALLOW Anywhere (v6)
443/tcp (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
```

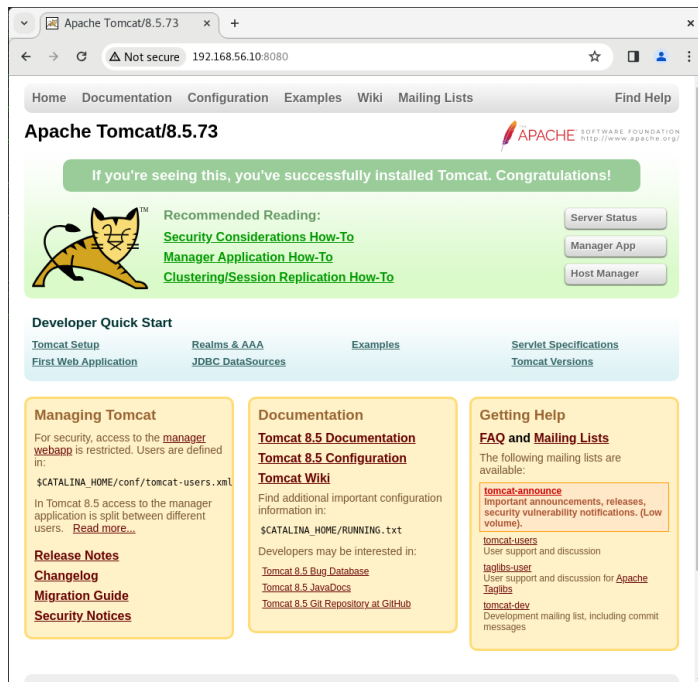
Para más evidencia de la ejecución de los playbooks en modo verbose (-v):

[https://github.com/gs151234/tallerfebrero2024/tree/main/documentacion/evidencias\\_ejecucion](https://github.com/gs151234/tallerfebrero2024/tree/main/documentacion/evidencias_ejecucion)

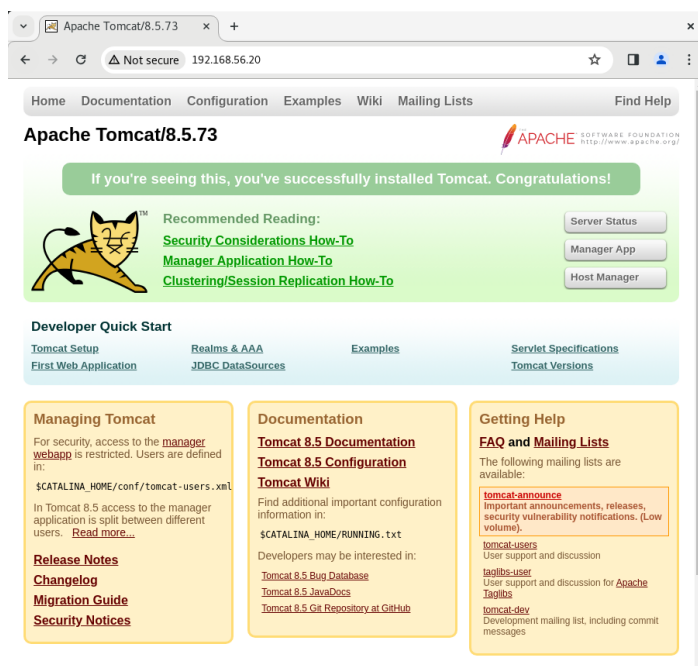
## Pruebas de acceso a la Web y Proxy

Desde el equipo Bastión se comprueba accediendo directo al servidor web <http://192.168.56.10:8080> y luego mediante el proxy <http://192.168.56.20>.

ServerA – <http://192.168.56.10:8080>



ServerB - <http://192.168.56.20>



Para esta muestra no se creó certificado para acceder con https.



## Bibliografía

Material de clase y grabaciones de las clases

ChatGPT

<https://docs.ansible.com/>

<https://www.youtube.com/@LearnLinuxTV>

<https://atareao.es/tutorial/ansible/playbooks-de-ansible/>

<https://httpd.apache.org/docs/2.4/vhosts/examples.html>

[https://access.redhat.com/documentation/es-es/red\\_hat\\_enterprise\\_linux/8/html/configuring\\_basic\\_system\\_settings/granting-sudo-access-to-a-user\\_configuring-basic-system-settings](https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/configuring_basic_system_settings/granting-sudo-access-to-a-user_configuring-basic-system-settings)

[https://www.server-world.info/en/note?os=Ubuntu\\_22.04&p=httpd&f=2](https://www.server-world.info/en/note?os=Ubuntu_22.04&p=httpd&f=2)

<https://bobcares.com/blog/a2ensite-ansible/>