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](#_Toc159948348)

[ServerA 3](#_Toc159948349)

[Instalación del SO 3](#_Toc159948350)

[ServerB 4](#_Toc159948351)

[Instalación del SO 4](#_Toc159948352)

[Guithub 5](#_Toc159948353)

[Ansible 6](#_Toc159948354)

[Playbooks 6](#_Toc159948355)

[initial\_config.yml 7](#_Toc159948356)

[hosts\_updates.yml 8](#_Toc159948357)

[web\_ servers.yml 8](#_Toc159948358)

[reverse\_proxy.yml 9](#_Toc159948359)

[Ejecución de los Playbook: 10](#_Toc159948360)

[initial\_config.yml 10](#_Toc159948361)

[hosts\_updates.yml 12](#_Toc159948362)

[web\_servers.yml 13](#_Toc159948363)

[reverse\_proxy.yml 13](#_Toc159948364)

[Pruebas de acceso a la Web y Proxy 15](#_Toc159948365)

[Bibliografía 16](#_Toc159948366)

# 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

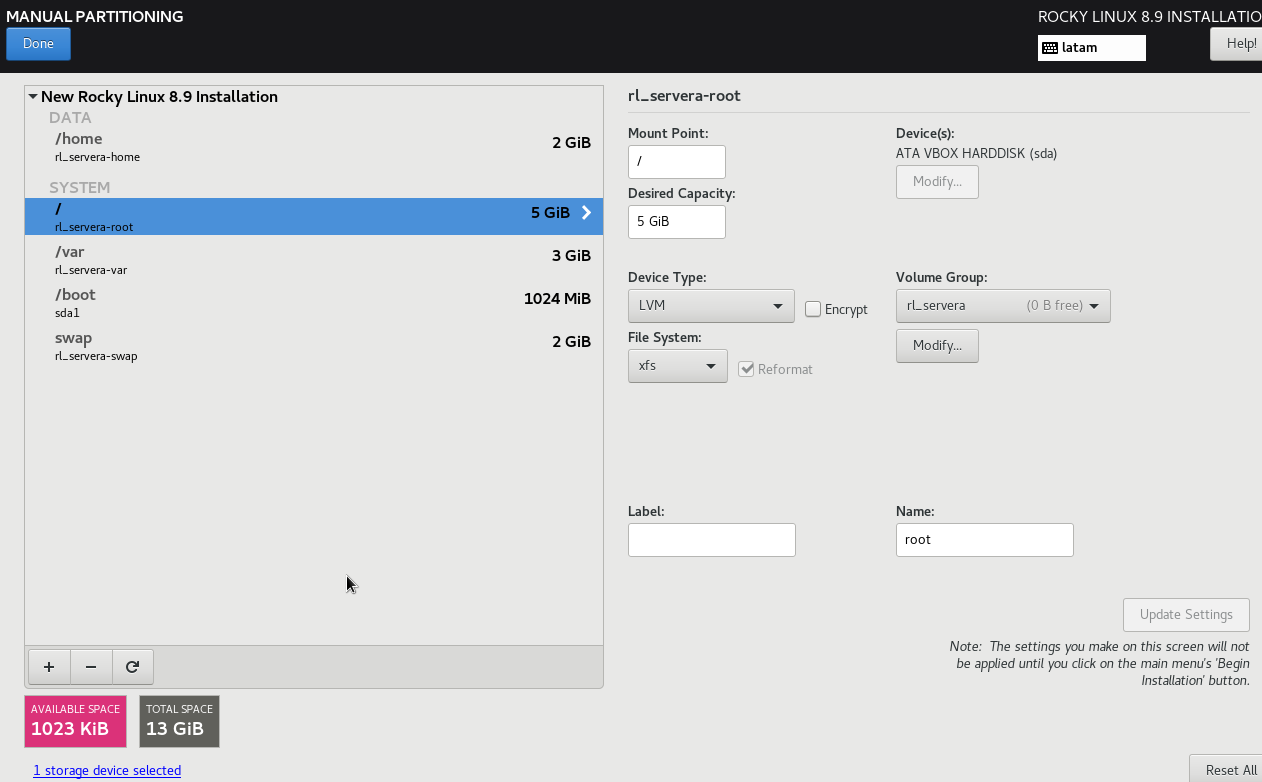
Tarjeta de red2: Adaptador sólo anfitrión IP: 192.168.56.10/24

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

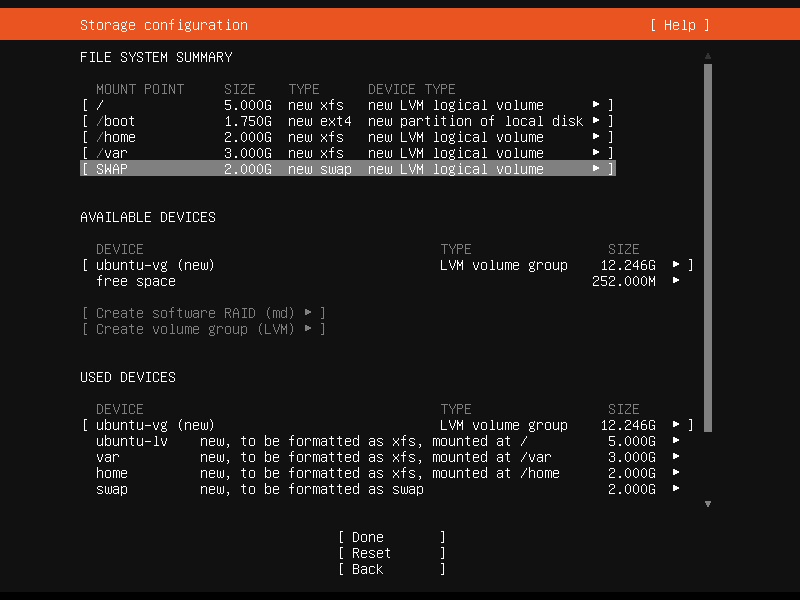
Instalación: Minimal

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](mailto: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
  + Teniendo en cuenta las configuraciones extra como habilitación de puertos en el firewall.
* Instalar y configurar un Proxy Reverso en el servidor Ubuntu
  + 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

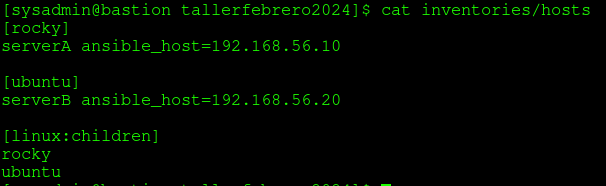
# git clone [git@github.com:gs151234/tallerfebrero2024.git](mailto:git@github.com:gs151234/tallerfebrero2024.git)

## 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.



Especificaremos a continuación detalles de las rareas que realiza cada playbook

### initial\_config.yml

El primer paso es común a todos los playbook, 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 publica a los nodos

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

- name: desabilitar login de root por ssh

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

- name: Desabilitar 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. Es scalamos 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 scalamos 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 proxi 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.

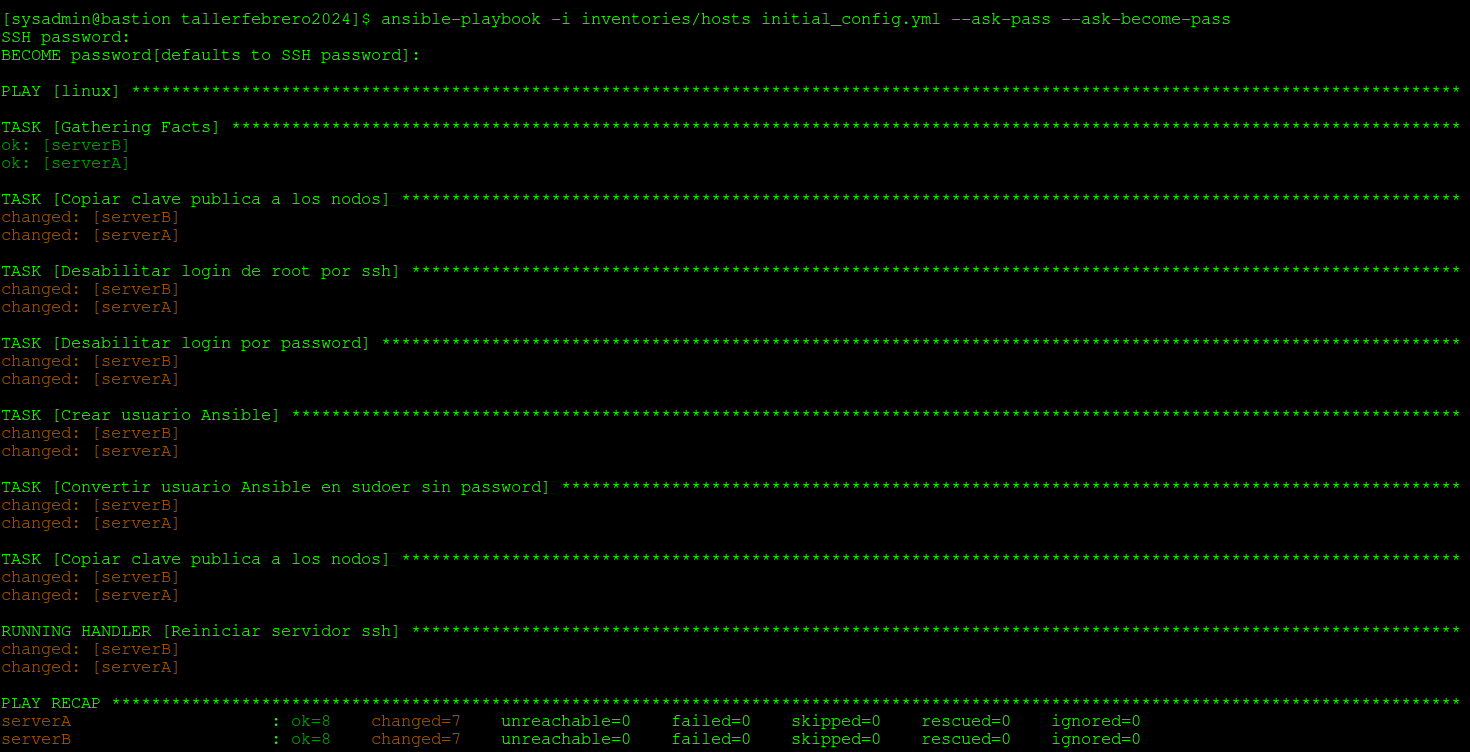
Se reinicia el SO para aplicar los cambios mediante un handler.

# Ejecución de los Playbook:

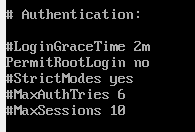
## initial\_config.yml

# ansible-playbook -i inventories/hosts initial\_config.yml --ask-pass --ask-become-pass

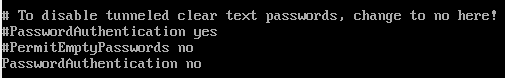
En este paso al no tener una clave publica en el usuario del servidor de destino, accedemos con la contraseña del administrador y la contraseña para escalar en permisos a SUDO.



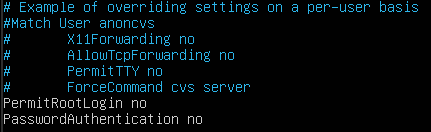
Se quita el permiso login de root por ssh en rocky



Se quita autenticación con password por ssh rocky



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



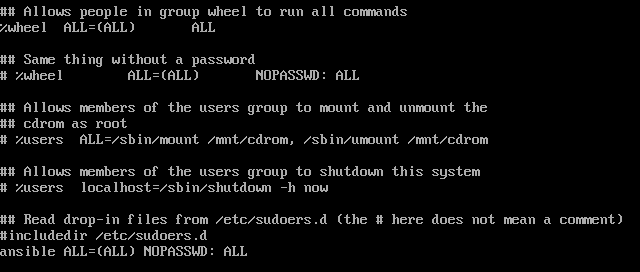
Creación del usuario ansible en rocky:



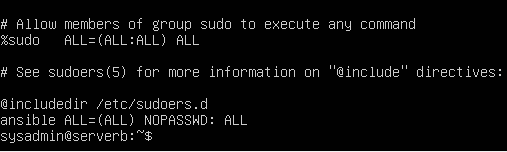
Creación del usuario ansible en ubuntu:



Permisos SUDO sin contraseña en rocky



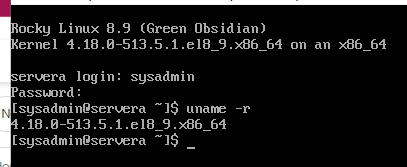
Permisos SUDO sin contraseña en Ubuntu

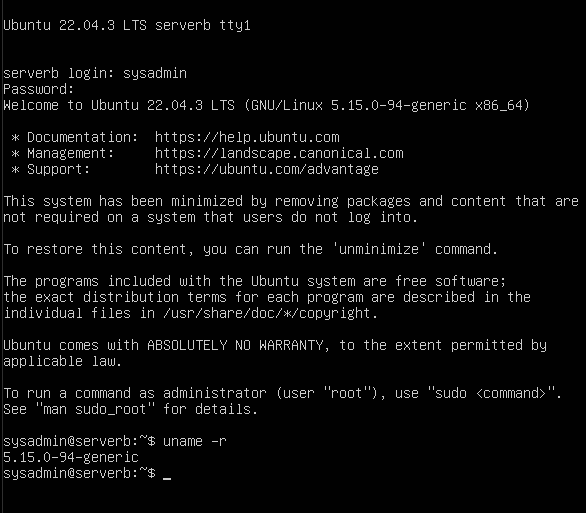


## hosts\_updates.yml

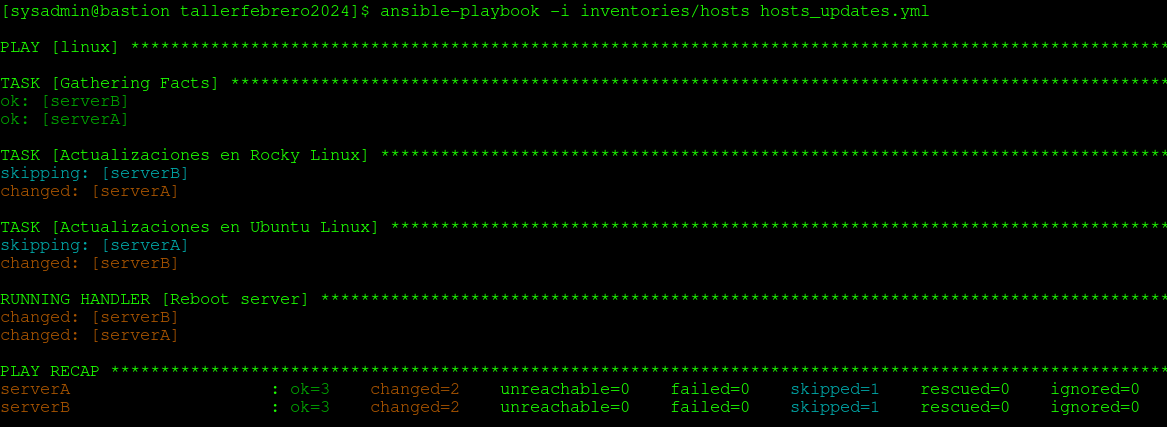
$ ansible-playbook -i inventories/hosts hosts\_updates.yml

Previo



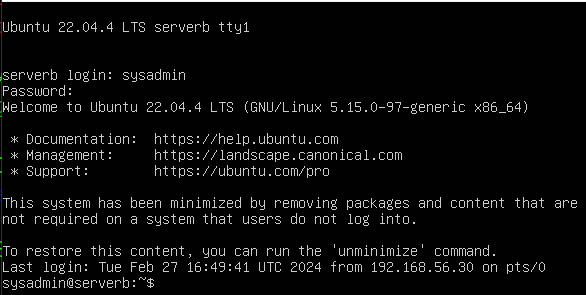


Ejecutamos el playbook



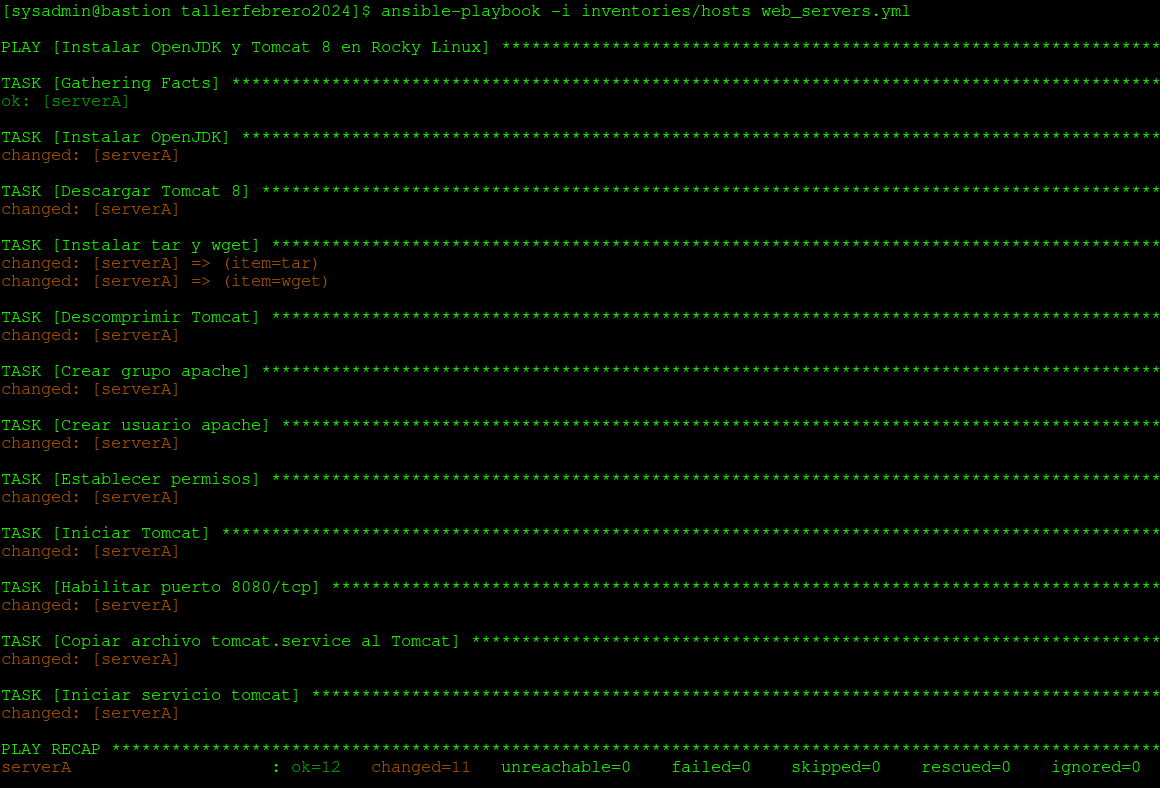
Terminada las tareas



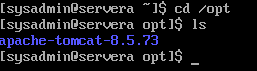


## web\_servers.yml

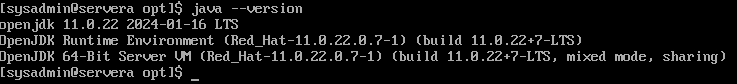
$ ansible\_playbook -I inventories/hosts web\_servers.yml



Apache Tomcat

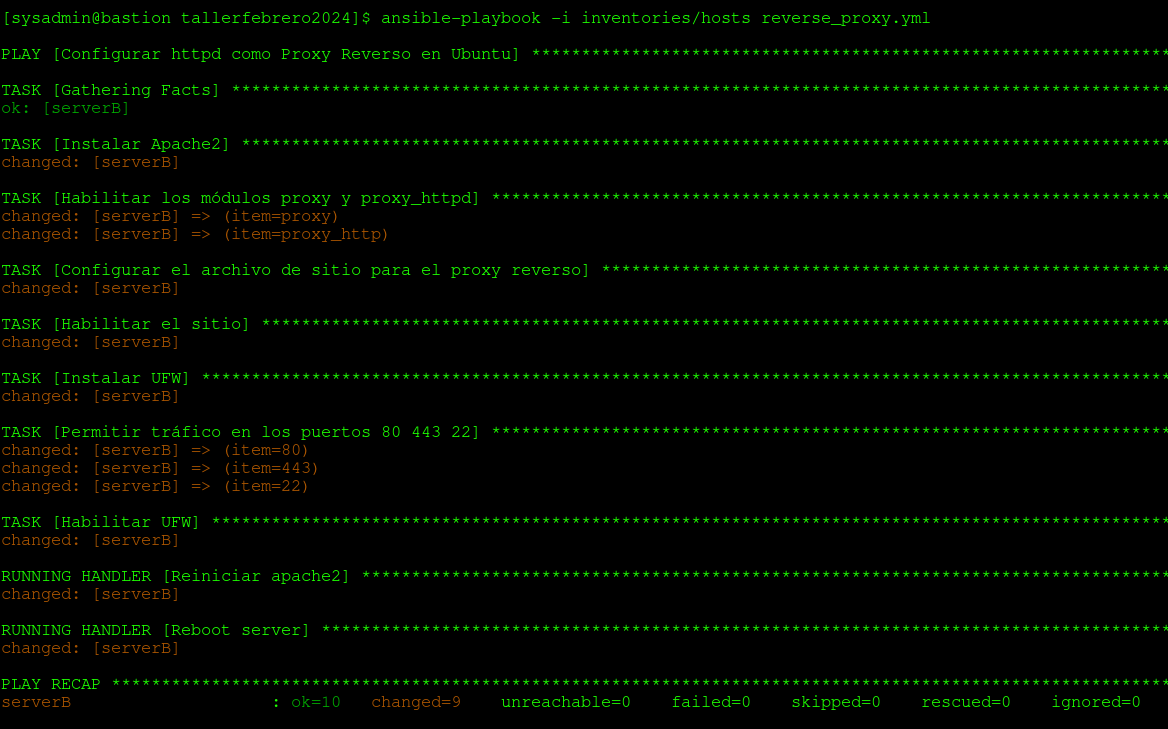


OpenJDK11

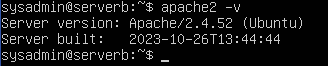


## reverse\_proxy.yml

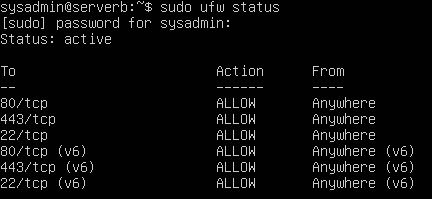
$ ansible-playbook -i inventories/hosts reverse\_proxy.yml



Apache2



UFW



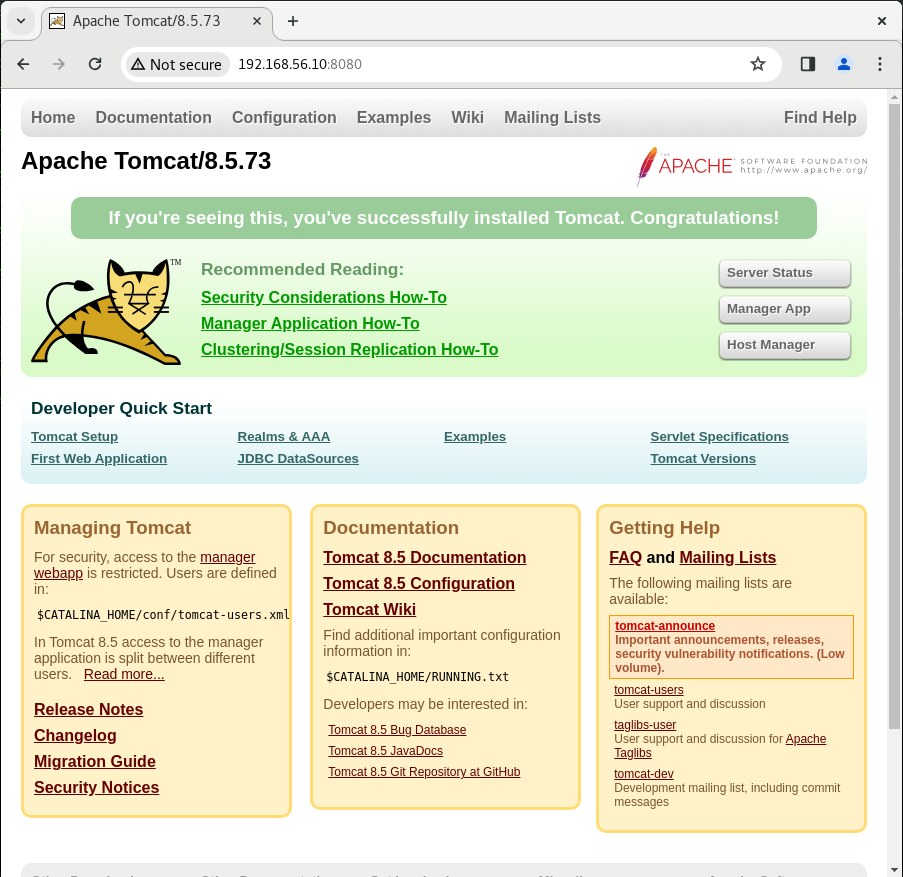
Para más evidencia de la ejecución de los playbooks en modo verbose (-v):

<https://github.com/gs151234/tallerfebrero2024/tree/main/documentacion/evidencias_ejecucion>

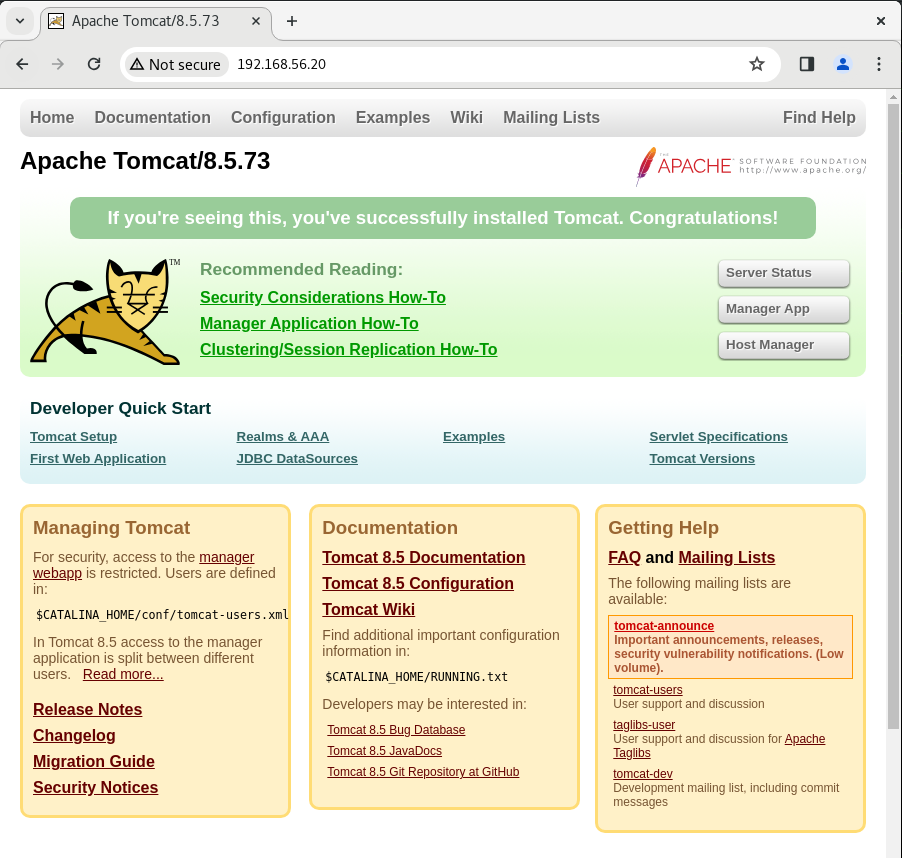
## Pruebas de acceso a la Web y Proxy

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://www.server-world.info/en/note?os=Ubuntu_22.04&p=httpd&f=2>

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