

Obligatorio Taller de Servidores Linux

Tercer semestre 2023

Alexis D'Andrea N. ° 230556

Nicolas Martins N. ° 292534

1. INDICE	
2. INTRODUCCIÓN	2
3. SERVIDORES	3
3.1 Partición en Ubuntu	3
3.2 Partición en Rocky Linux.....	4
4. BASTIÓN	5
5. GIT.....	6
6. ANSIBLE	8
7. ROLES.....	11
7.1 Apache.....	11
7.2 Firewall.debian	11
7.3 Mariadb	11
7.4 Podman.....	11
8. PLAYBOOKS.....	12
8.1 PB_Web_Server.....	12
8.2 PB_app_web.....	14
8.3 PB_mariadb	16
8.4 PB_updates.....	18
9. BIBLIOGRAFIA	19
10. ANEXO	20

2. INTRODUCCIÓN

Para concretar lo solicitado por la consigna, comenzamos con la instalación del servidor que se utilizará como bastión y todas las configuraciones necesarias en el mismo, con el objetivo de que este pueda distribuir paquetes y configuraciones a los demás servidores de la red, a través de ansible.

Luego instalamos un nuevo servidor Ubuntu y otro servidor Rocky, en los cuales realizaremos pruebas de los playbooks que vayamos creando.

A través de playbooks y roles, procedimos a instalar el contenedor en uno de los servidores, para luego disponibilizar una aplicación web con una base de datos, que será redirigida al servidor apache, con la finalidad de que pueda ser accedida mediante el mismo.

Al finalizar realizamos las verificaciones correspondientes, para asegurarnos de que las instalaciones y configuraciones realizadas a través de ansible fueran realizadas correctamente.

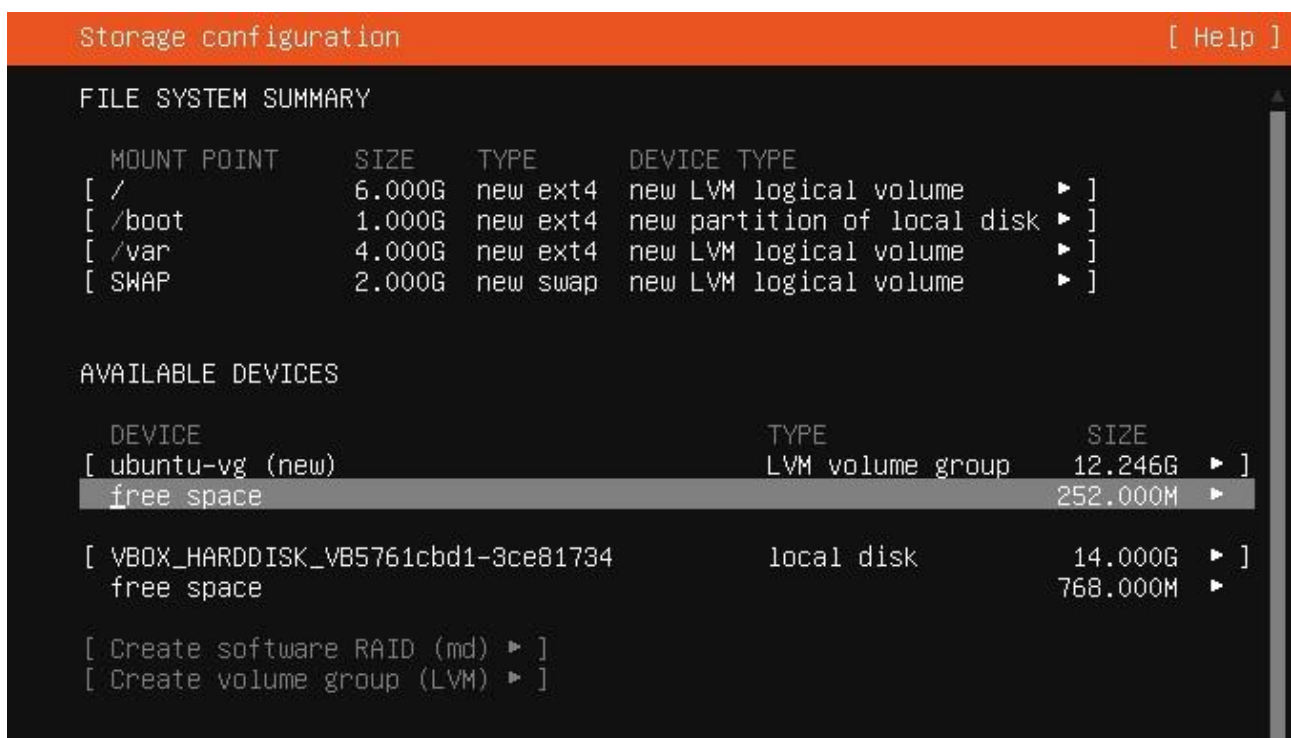
3. SERVIDORES

Nuestro primer servidor, es un Ubuntu con interfaz gráfica, 14 GB de disco, 2 GB de memoria RAM y dos interfaces de red (una interna y otra con salida a internet). En la interfaz del NAT dejamos las configuraciones por defecto para que mantenga la salida hacia internet, luego en la interfaz que va hacia la red le colocamos la IP 192.168.56.101.

Luego instalamos dos servidores con 1 GB de memoria RAM y 14 GB de disco, con una partición de 1GB para /boot y el resto del disco en un volumen lógico de 6GB para /, 4 GB para /var, 2 GB para swap. Un servidor con sistema operativo Rocky y el otro con Ubuntu. Cada uno tendrá 2 interfaces de red, 1 conectada a NAT con las configuraciones por defecto y la otra a una red Interna que le permita conectarse al equipo bastión con Ansible.

Nuestro servidor bastión se llamará bastion.taller.uy, el servidor Rocky se llamará rocky.taller.uy (IP 192.168.56.103) y el servidor Ubuntu se llamará ubuntuserver.taller.uy (IP 192.168.56.102).

3.1 Partición en Ubuntu



3.2 Partición en Rocky Linux

MANUAL PARTITIONING ROCKY LINUX 8.8 INSTALLATION

[Done](#) latam [Help!](#)

▼ New Rocky Linux 8.8 Installation

SYSTEM

/	6 GiB	>
rl-root		
/var	4 GiB	
rl-var		
/boot	1024 MiB	
sda1		
swap	2 GiB	
rl-swap		

+ **-** **↺**

AVAILABLE SPACE
1020 MiB

TOTAL SPACE
14 GiB

rl-root

Mount Point:
/

Desired Capacity:
6 GiB

Device Type:
LVM ☐ Encrypt

File System:
xfs ☒ Reformat

Device(s):
ATA VBOX HARDDISK (sda)
[Modify...](#)

Volume Group:
rl (4 MiB free)

Label:

Name:
root

4. BASTIÓN

En el servidor bastión crearemos las configuraciones necesarias de ansible para distribuir paquetes y configuraciones a los demás servidores, para ellos realizaremos las siguientes tareas:

Creamos el usuario ansible en cada uno de los servidores (excepto el bastión) con el comando `sudo useradd ansible`. Y le proporcionamos una contraseña, con el comando `sudo passwd ansible`.

Luego modificamos el archivo `/etc/sudoers`, con el comando `sudo visudo`, que verifica la sintaxis del archivo antes de guardarlo, para evitar errores que puedan bloquear el acceso a sudo.

```
Defaults        secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems) .
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere

root    ALL=(ALL)        ALL
ansible ALL=(ALL) NOPASSWD: ALL
```

Verificamos que haya quedado el usuario ansible, con el comando `sudo -l -U ansible`.

```
Matching Defaults entries for ansible on bastion:
    !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 LANGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User ansible may run the following commands on bastion:
    (ALL) NOPASSWD: ALL
```

De esta forma hacemos que el usuario ansible, tenga permisos con SUDO sin contraseña.

Luego generamos una clave publica para copiarla en los servidores destino, para poder conectarnos a estos por este medio y no necesitar ingresar credenciales.

Para ello utilizamos los siguientes comandos:

`ssh-keygen` (Generar las claves)

`ssh-copy-id -i /home/ansible/.ssh/id_rsa.pub ansible@192.168.56.103` (Copiar las claves)

`ssh-copy-id -i /home/ansible/.ssh/id_rsa.pub ansible@192.168.56.102` (Copiar las claves)

Instalamos el repositorio Epel con los comandos:

`dnf config-manager --set-enabled powertools`

`dnf install epel-release`

5. GIT


Utilizamos la web <https://github.com/>, para trabajar en nuestro proyecto de ansible. Para ello nos creamos una cuenta en la web, copiamos la clave pública del servidor bastión en ella y creamos un repositorio llamado tallerjulio2023 en el cual estaremos realizando el proyecto.

SSH keys

[New SSH key](#)

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

Authentication Keys



TallerLinux
SHA256: aRAg+2N8fioj1GdZeJ++uX7k4jYRuFUSDwqF7N9RYMY
Added on Aug 6, 2023
Last used within the last week — Read/write

Delete

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

En el servidor bastión instalamos GIT con el comando `sudo apt install git`, creamos el directorio en el cual iremos a trabajar con el comando `mkdir tallerjulio2023` e iniciamos GIT con el comando `git init` estando sobre el directorio `/home/sysadmin/tallerjulio2023`.

Configuramos nuestro nombre de usuario y correo electrónico de git con los comandos:

`git config --global user.name <nombre de usuario>`

`git config --global user.email <correo electrónico>`

Creamos un README.md, hicimos un commit, agregamos el origen remoto e hicimos un push del mismo, con los siguientes comandos:

```
git add README.md
git commit -m "Primer commit del taller"
git branch -M main
git remote add origin git@github.com:adandrea8/tallerjulio2023.git
git push -u origin main
```

Para finalizar con la sincronización agregamos a los colaboradores y clonamos el proyecto en sus equipos.

6. ANSIBLE

Instalamos ansible con el comando `sudo apt install`, luego hacemos `ansible-config init --disabled > ansible.cfg` para que cree el archivo `ansible.cfg` en nuestro directorio con configuraciones por defecto y deshabilitadas que luego podemos modificar en caso de querer especificar alguna configuración.

Agregamos la ruta del inventario, modificando el archivo `ansible.cfg` de la siguiente manera, para que los hosts utilizados en los playbooks puedan desplegarse sobre los grupos creados:

```
GNU nano 2.9.8                               ansible.cfg
# All playbooks and roles in the official examples repos assume the
# Changing the setting to ``merge`` applies across variable sources
# The Ansible project recommends you **avoid ``merge`` for new proj
# It is the intention of the Ansible developers to eventually depre
;hash_behaviour=replace

# (pathlist) Comma separated list of Ansible inventory sources
inventory=/home/ansible/tallerjulio2023/inventario
```

Para ello también creamos el archivo inventario en la ruta indicada con los siguientes datos:

```
[redhat]
rocky.taller.uy ansible_host=192.168.56.103

[debian]
ubuntuserver.taller.uyansible_host=192.168.56.102

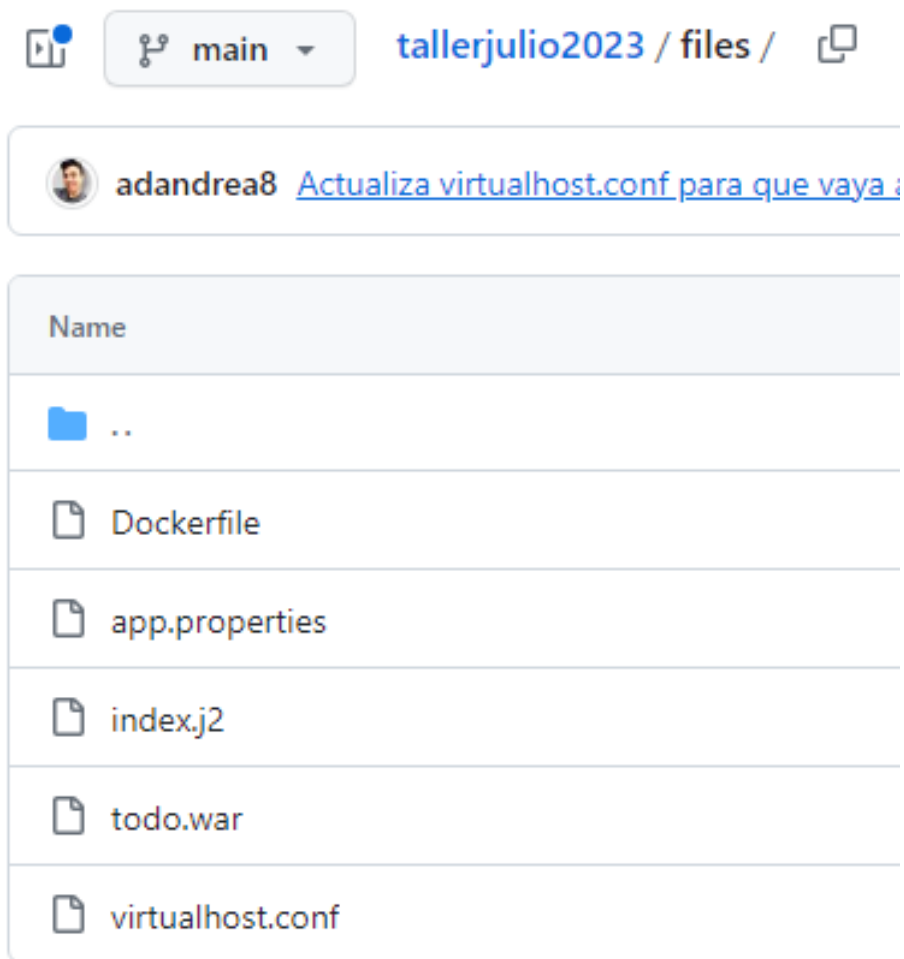
[linux:children]
redhat
debian
```

De la misma manera, creamos los archivos encriptados `pass.yml` y `todopass.yml`:

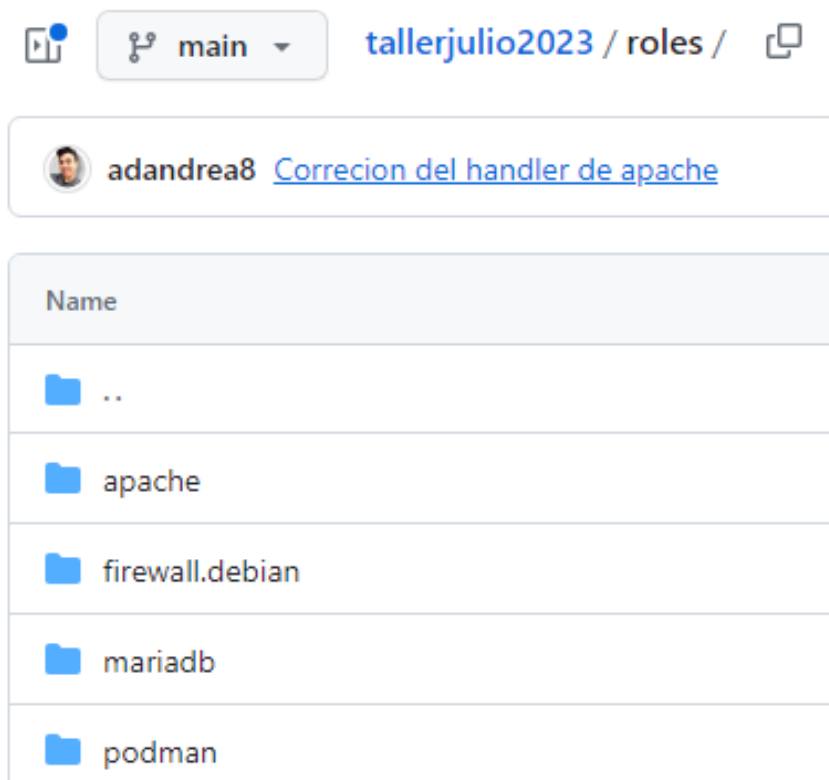
Estos archivos tendrán las variables que utilizarán contraseña y para una mayor seguridad al ser archivos sensibles estarán encriptados con vault.

En el archivo `README.md` de cada directorio, estará una breve explicación de lo que contiene su directorio.

En el directorio `files`, tendremos todos los archivos que se copiarán a los servidores según la necesidad de cada playbook.

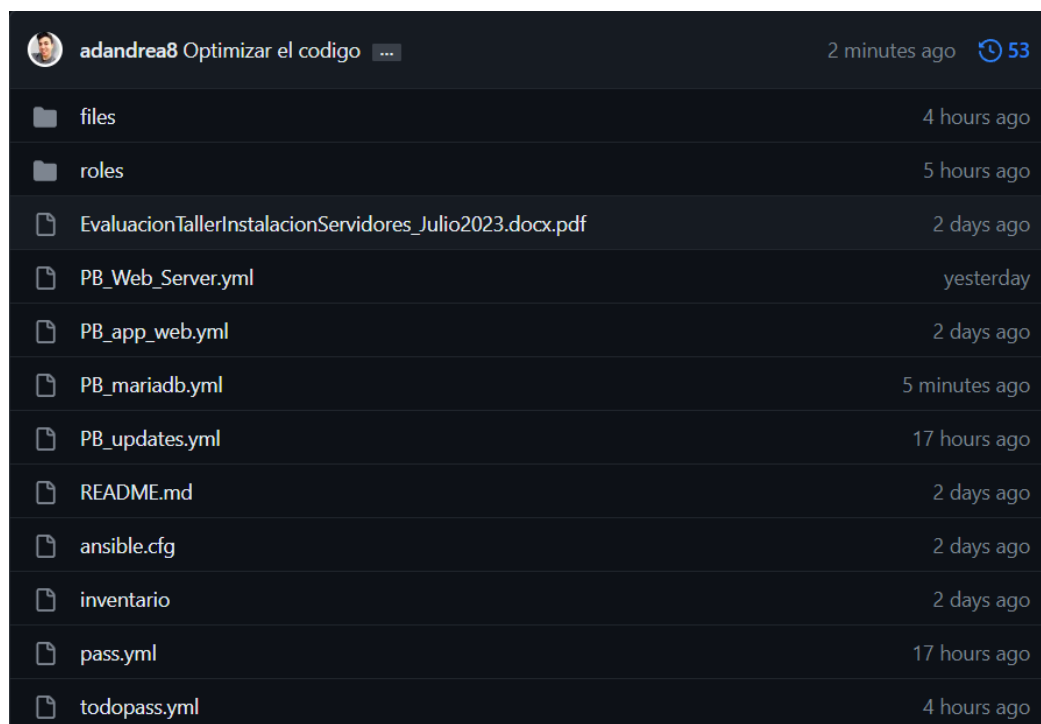


En el directorio roles crearemos todos los roles con el comando `ansible-galaxy init <nombre del rol>`, dejándolo de la siguiente manera:



El archivo `pass.yml`, contiene la contraseña encriptada con vault del usuario root de mariadb y el archivo `todopass.yml` contiene la contraseña encriptada con vault del usuario todo.

Luego crearemos los playbooks en el directorio principal, dejándolo de la siguiente manera:



7. ROLES

7.1 Apache

/home/sysadmin/tallerjulio2023/roles/apache

El rol apache, instala apache, abre los puertos en el firewall (lo habilita si es necesario), habilita el servicio de apache y lo levanta, indistintamente de cuál de las dos familias (Debian o RedHat) de sistemas operativos llamen al rol.

7.2 Firewall.debian

/home/sysadmin/tallerjulio2023/roles/firewall.debian

El rol firewall.debian, habilita toda la salida y deshabilita toda la salida de tráfico y luego habilita el firewall en los servidores de la familia Debian.

7.3 Mariadb

/home/sysadmin/tallerjulio2023/roles/mariadb

El rol mariadb, instala mariadb-server, mariadb-client, python3 y PyMySQL. Abre el puerto 3306 en el firewall y por último habilita el servicio y lo inicia indistintamente de cuál de las dos familias (Debian o RedHat) de sistemas operativos llamen al rol. Luego configura mariadb para que escuche todas las interfaces, actualiza la clave clave root de la base de datos, elimina al usuario anónimo y la base de datos de test creada por defecto.

7.4 Podman

/home/sysadmin/tallerjulio2023/roles/podman

El rol podman, instala Podman y lo habilita para que se inicie con el sistema, independientemente de que sea Debian o RedHat.

8. PLAYBOOKS

8.1 PB_Web_Server

El playbook PB_Web_Server.yml se aplicará solo en los servidores RedHat y realizará las siguientes tareas:

1. Aplica el rol apache
2. Crea el directorio para los virtualhosts (/etc/httpd/vhosts.d)
3. Copia el archivo virtualhost.conf ubicado en ./files con la configuración y lo pega en la ruta del punto anterior
4. Agrega la línea IncludeOptional en todos los archivos de configuración de virtualhost
5. Crea el directorio en donde se guardarán los index
6. Copia el archivo index ubicado en ./files y lo pega en el directorio creado en el punto anterior
7. Habilita la conexión desde afuera a la web

./files/virtualhost.conf

```
<VirtualHost *:80>
    ServerName 192.168.56.103
    ServerAdmin sysadmin@taller.uy
    ProxyPreserveHost On

    ProxyPass /todo/ http://192.168.56.103:8080/todo/
    ProxyPassReverse /todo/ http://192.168.56.103:8080/todo/

    DocumentRoot /var/www/web/html

    <Directory /var/www/web/html >
        AllowOverride none
        Options Indexes FollowSymLinks
        Require all granted
    </Directory>
</VirtualHost>
```

./files/index.j2



```
1 Este webserver esta en {{ ansible_hostname }} corriendo {{ ansible_distribution }} {{ ansible_distribution_version }}
```

Prueba de funcionamiento de playbook:

```
PLAY [redhat] *****
TASK [Gathering Facts] *****
Enter passphrase for key '/home/ansible/.ssh/id_rsa':
ok: [rocky.taller.uy]

TASK [apache : Instalar apache] *****
ok: [rocky.taller.uy]

TASK [apache : Abrir puertos del firewall] *****
ok: [rocky.taller.uy] => (item=http)
ok: [rocky.taller.uy] => (item=https)

TASK [apache : Iniciar y habilitar httpd service] *****
ok: [rocky.taller.uy]

TASK [apache : Instalar apache Debian] *****
skipping: [rocky.taller.uy]

TASK [apache : Iniciar y habilitar httpd service en Debian] *****
skipping: [rocky.taller.uy]

TASK [Crear el virtualhost con la configuracion del directorio] *****
ok: [rocky.taller.uy]

TASK [Copiar virtualhost de configuracion] *****
ok: [rocky.taller.uy]

TASK [Include vhosts directory] *****
ok: [rocky.taller.uy]

TASK [Crear el directorio para index] *****
ok: [rocky.taller.uy]

TASK [Copiar archivo index] *****
ok: [rocky.taller.uy]

TASK [SELinux] *****
ok: [rocky.taller.uy]

PLAY RECAP *****
rocky.taller.uy      : ok=10   changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Prueba de servicio corriendo:

```
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-08-07 14:13:54 -03; 3h 6min ago
     Docs: man:httpd.service(8)
  Main PID: 806 (httpd)
    Status: "Running, listening on: port 80"
     Tasks: 213 (limit: 11135)
    Memory: 48.3M
    CGroup: /system.slice/httpd.service
            └─806 /usr/sbin/httpd -DFOREGROUND
              └─826 /usr/sbin/httpd -DFOREGROUND
                └─827 /usr/sbin/httpd -DFOREGROUND
                  └─828 /usr/sbin/httpd -DFOREGROUND
                    └─829 /usr/sbin/httpd -DFOREGROUND
```

Puertos habilitados:

```
target: default
icmp-block-inversion: no
interfaces: enp0s3 enp0s8
sources:
services: cockpit dhcpv6-client http https ssh
ports:
protocols:
forward: no
```

8.2 PB_app_web

Para poder utilizar el modulo `containers.podman.podman_image` y el `containers.podman.podman_container`, tuvimos que utilizar el comando `ansible-galaxy collection install community.general` en el equipo bastión.

El playbook PB_app_web.yml se aplicará solo en los servidores RedHat y realizará las siguientes tareas:

1. Aplica el rol podman
2. Crea un directorio para el contenedor (/ansible/contenedor/appweb)
3. Copia el archivo todo.war ubicado en ./files a la ruta del punto anterior
4. Copia el archivo app.propiedades al directorio /ansible/contenedor/appweb
5. Copia el archivo Dockerfile ubicado en ./files al directorio /ansible/contenedor/appweb
6. Ejecuta y configura la imagen
7. Inicia el contenedor

./files/app.properties

```
1  tipoDB=mysql
2  jdbcURL=jdbc:mysql://192.168.56.103:3306/todo
3  jdbcUsername=todo
4  jdbcPassword=prueba2022
```

./files/Dockerfile

```
1  FROM tomcat:9.0
2
3  RUN mkdir /opt/config
4
5  COPY app.properties /opt/config
6
7  ADD todo.war /usr/local/tomcat/webapps
8
9  EXPOSE 8080
10
11 CMD ["catalina.sh", "run"]
```

Prueba de funcionamiento del playbook:

```
PLAY [Instalar y configurar Tomcat en Podman] *****

TASK [Gathering Facts] *****
ok: [rocky.taller.uy]

TASK [podman : Instalar Podman en RedHat] *****
ok: [rocky.taller.uy]

TASK [podman : Habilitar podman para que inicie con el sistema en RedHat] *****
changed: [rocky.taller.uy]

TASK [podman : Instalar Podman en Debian] *****
skipping: [rocky.taller.uy]

TASK [podman : Habilitar podman para que inicie con el sistema en Debian] *****
skipping: [rocky.taller.uy]

TASK [Crear directorio para el contenedor] *****
ok: [rocky.taller.uy]

TASK [Copiar todo.war a la ruta] *****
ok: [rocky.taller.uy]

TASK [Copiar app.properties a la ruta] *****
ok: [rocky.taller.uy]

TASK [Copiar dockerfile a la ruta] *****
ok: [rocky.taller.uy]

TASK [Configurar y ejecutar] *****
ok: [rocky.taller.uy]

TASK [Iniciar contenedor] *****
changed: [rocky.taller.uy]

PLAY RECAP *****
rocky.taller.uy      : ok=9    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Prueba de ejecución de contenedor:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost/appweb	latest	274d6clb4cl3	19 hours ago	436 MB
docker.io/library/tomcat	9.0	ee772f2dcb36	12 days ago	433 MB

Prueba de acceso a la web:

[←](#) [→](#) [↻](#) [🔒 192.168.56.103:8080/todo/](#)

Todo App

Login Form

User Name:

Password:

8.3 PB_mariadb

El playbook PB_mariadb.yml cumple con la función de instalar y configurar el motor de base de datos con instalación segura. Esta instalación se ejecutará en los hosts “Debian” y con el usuario “ansible”

- 1- Llama a los archivos cifrados con las contraseñas “pass.yml” y “todopass.yml”
- 2- Aplica los roles “mariadb” y “firewall.debian”
- 3- Crea el usuario “todo” con la contraseña almacenada en “todopass.yml”
- 4- Crea la base de datos “todo” en mariadb
- 5- Crea el directorio /ansible/sql para el repositorio SQL
- 6- Copia el statement de la base de datos todo a la ruta del paso anterior
- 7- Crea la estructura de la base de datos todo

Prueba de funcionamiento del playbook:

```
ansible@bastion:~/tallerjulio2023$ ansible-playbook PB_mariadb.yml --ask-become-pass --ask-vault-pass
BECOME password:
Vault password:

PLAY [debian] *****

TASK [Gathering Facts] *****
Enter passphrase for key '/home/ansible/.ssh/id_rsa':
ok: [ubuntuuser.taller.uy]

TASK [mariadb : Instalar mariadb en RedHat] *****
skipping: [ubuntuuser.taller.uy]

TASK [mariadb : Iniciar y habilitar en RedHat] *****
skipping: [ubuntuuser.taller.uy]

TASK [mariadb : Abrir el puerto 3306 en el firewall en RedHat] *****
skipping: [ubuntuuser.taller.uy]

TASK [mariadb : Instalar mariadb en Debian] *****
ok: [ubuntuuser.taller.uy] => (item=mariadb-server)
ok: [ubuntuuser.taller.uy] => (item=mariadb-client)
ok: [ubuntuuser.taller.uy] => (item=python3-pip)

TASK [mariadb : Instalar PyMySQL en Debian] *****
ok: [ubuntuuser.taller.uy]

TASK [mariadb : Iniciar y habilitar el servicio mariadb en Debian] *****
ok: [ubuntuuser.taller.uy]
```

```
TASK [mariadb : Abrir el puerto en el firewall en Debian] *****
changed: [ubuntuuser.taller.uy]

TASK [firewall.debian : Configuracion Firewall incoming allow en Debian] *****
changed: [ubuntuuser.taller.uy]

TASK [firewall.debian : Configuracion Firewall outcoming deny en Debian] *****
changed: [ubuntuuser.taller.uy]

TASK [firewall.debian : ufw service enable and start] *****
ok: [ubuntuuser.taller.uy]

TASK [Configure mariadb to listen on all interfaces] *****
ok: [ubuntuuser.taller.uy]

TASK [Update MariaDB root password] *****
ok: [ubuntuuser.taller.uy] => (item=127.0.0.1)
ok: [ubuntuuser.taller.uy] => (item=localhost)
ok: [ubuntuuser.taller.uy] => (item=192.168.56.%)

TASK [Delete anonymous user] *****
ok: [ubuntuuser.taller.uy]

TASK [Delete test database] *****
ok: [ubuntuuser.taller.uy]

PLAY RECAP *****
ubuntuuser.taller.uy : ok=12 changed=3 unreachable=0 failed=0 skipped=3 rescued=0 ignored=0
```

Prueba de puertos abiertos y estado de firewall:

```
ansible@ubuntuuser:~$ sudo ufw status
Status: active

To Action From
--
3306/tcp ALLOW Anywhere
Anywhere ALLOW Anywhere
3306/tcp (v6) ALLOW Anywhere (v6)
Anywhere (v6) ALLOW Anywhere (v6)

Anywhere DENY OUT Anywhere
Anywhere (v6) DENY OUT Anywhere (v6)

ansible@ubuntuuser:~$ systemctl status ufw
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enabled)
   Active: active (exited) since Tue 2023-08-08 14:35:26 UTC; 47min ago
```

Prueba de base de datos y tablas:

```
MariaDB [todo]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| todo |
+-----+
5 rows in set (0.001 sec)

MariaDB [todo]> SHOW TABLES;
+-----+
| Tables_in_todo |
+-----+
| todos |
| users |
+-----+
2 rows in set (0.000 sec)
```

8.4 PB_updates

El playbook PB_updates.yml se aplicará tanto en los servidores RedHat como en los servidores Debian y lo único que hará es revisar que tengan actualizaciones pendientes, y en el caso de que tengan alguna actualización, se hará la actualización y se reiniciará el servidor.

9. BIBLIOGRAFIA

1. Sitios Internet:

- a. https://docs.rockylinux.org/es/guides/security/ssl_keys [https/](https://)
- b. <https://www.redhat.com/es>
- c. <https://ubuntu.com/>
- d. <https://docs.ansible.com/>

- 2. Plataforma ORT Aulas / Materiales del curso
- 3. Consultas con docente Enrique Verdes
- 4. Clases grabadas en Plataforma ORT / Aulas

10. ANEXO

Este anexo trata sobre cómo redactar la declaración de autoría. Debe constar del siguiente texto:

Nosotros, Alexis D'Andrea y Nicolas Martins, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos Obligatorio del Taller de Servidores Linux;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Firma:



Alexis D'Andrea

Firma:



Nicolas Martins

Normas específicas para la presentación de trabajos finales de carrera (TFDC) Facultad de Ingeniería – 9
Escuela de Tecnología – Documento 302 ET Universidad ORT Uruguay