

Nicolás Soárez
Estudiante N.º 142124

Gabriel Larrosa
Estudiante N.º 238351

Taller servidores Linux

Docente: Alejandro Vartabedián

Obligatorio 2024

| | |
|---|----|
| Objetivo | 3 |
| Propuesta de solución..... | 3 |
| Diagrama de Red | 3 |
| Instalación de Sistema Operativo..... | 4 |
| CentoOS | 4 |
| Ubuntu | 7 |
| CentOS con Ansible..... | 10 |
| Entorno Ansible | 11 |
| Herramienta Editor de código | 11 |
| Configuración de Inventario..... | 12 |
| Configuración de Playbooks..... | 13 |
| Playbooks para instalación de podman..... | 13 |
| Playbooks para instalación de tomcat..... | 15 |
| Playbooks para instalación de base de datos | 19 |
| Playbooks para instalación de properties..... | 24 |
| Copiado del archivo de aplicación todo.war..... | 25 |
| Creación de Servicio | 26 |
| Evidencia de funcionamiento..... | 28 |
| Elevar al repositorio Git | 29 |

Objetivo

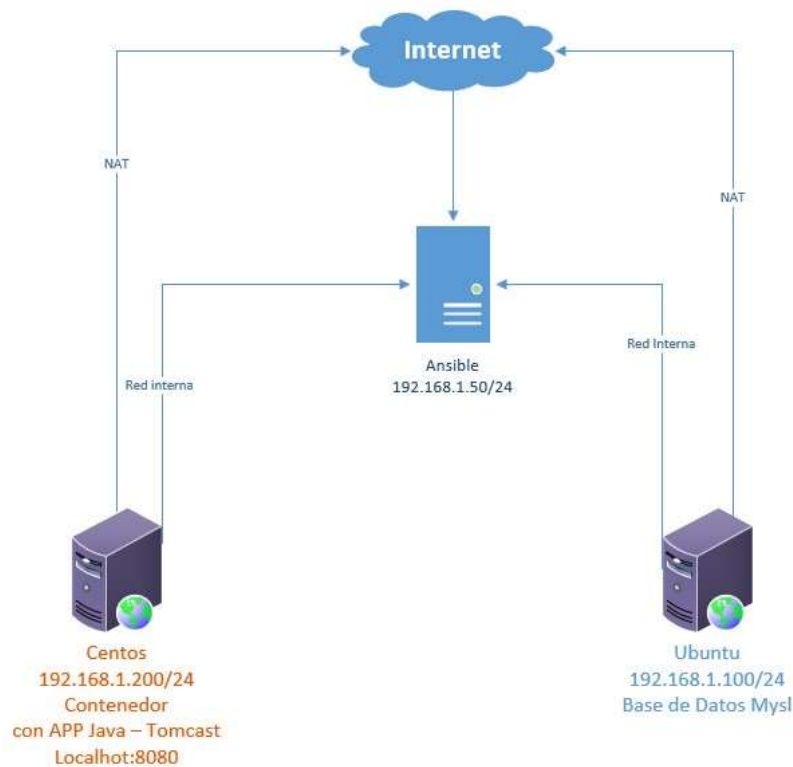
Cumplir con los requisitos solicitados en los 4 puntos del Obligatorio, logrando administrar a través de Ansible una automatización y administración de las tareas planteadas y elevar a un repositorio Git en internet.

Propuesta de solución

De acuerdo con las directrices establecidas, utilizaremos un servidor CentOS como administrador "Ansible", tendrá las claves SSH necesarias para gestionar los servidores de aplicación (CentOS) y de la base de datos (Ubuntu).

- Servidor Ansible – CentOS.
- Servidor Aplicaciones – CentOS.
- Servidor de Base de Datos – Ubuntu.

Diagrama de Red



Instalación de Sistema Operativo

CentOS

- Arranca desde el medio de instalación: Inserta el medio (USB/DVD) y arranca el sistema.
- Selección de idioma y teclado: Elige tu idioma y disposición del teclado.
- Configuración de red: Configura la red o redes.
- Configuración del disco: Define las particiones del disco.
- Creación de usuario: Crea un usuario con permisos de administrador.
- Finalización: Completa la instalación y reinicia el sistema.

Personalización de particiones de disco:

Disco de 13GB, 1CPU y 2GB RAM:

- Partición de 1GB para /boot
- LVM de 7GB para /
- LVM de 3GB para /var
- LVM de 2GB para SWAP

PARTICIONADO MANUAL INSTALACIÓN DE CENTOS STREAM 9

Hecho latam ¡Ayuda!

▼ Nueva instalación CentOS Stream 9

SISTEMA

| | | |
|-------|------|----------|
| /boot | sda1 | 1024 MiB |
| / | sda2 | 7 GiB |
| /var | sda3 | 3 GiB |
| swap | sda5 | 2 GiB |

sda1

Punto de montaje: /boot

Capacidad deseada: 1024 MiB

Tipo de dispositivo: Partición estándar ☐ Cifrar

Sistema de archivos: xfs ☒ Reformatear

Etiqueta:

Nombre: sda1

Configuración de actualizaciones

Nota: Los cambios que usted haga en esta pantalla no se aplicarán hasta que usted haga clic en el botón 'Comenzar instalación'.

ESPACIO DISPONIBLE 1,82 MiB ESPACIO TOTAL 13 GiB

1-dispositivo de almacenamiento seleccionado Descartar todos los cambios

PARTICIONADO MANUAL INSTALACIÓN DE CENTOS STREAM 9

[Hecho](#) [latam](#) [¡Ayuda!](#)

▼ Nueva instalación CentOS Stream 9

SISTEMA

| | | |
|-------|------|----------|
| /boot | sda1 | 1024 MiB |
| / | sda2 | 7 GiB |
| /var | sda3 | 3 GiB |
| swap | sda5 | 2 GiB |

sda2

Punto de montaje:

Capacidad deseada:

Tipo de dispositivo: Partición estándar ☐ Cifrar

Sistema de archivos: xfs ☒ Reformatear

Etiqueta:

Nombre:

[Configuración de actualizaciones](#)

Nota: Los cambios que usted haga en esta pantalla no se aplicarán hasta que usted haga clic en el botón 'Comenzar instalación'.

ESPACIO DISPONIBLE: 1,82 MiB
 ESPACIO TOTAL: 13 GiB

[1 - dispositivo de almacenamiento seleccionado](#) [Descartar todos los cambios](#)

PARTICIONADO MANUAL INSTALACIÓN DE CENTOS STREAM 9

[Hecho](#) [latam](#) [¡Ayuda!](#)

▼ Nueva instalación CentOS Stream 9

SISTEMA

| | | |
|-------|------|----------|
| /boot | sda1 | 1024 MiB |
| / | sda2 | 7 GiB |
| /var | sda3 | 3 GiB |
| swap | sda5 | 2 GiB |

sda3

Punto de montaje:

Capacidad deseada:

Tipo de dispositivo: Partición estándar ☐ Cifrar

Sistema de archivos: xfs ☒ Reformatear

Etiqueta:

Nombre:

[Configuración de actualizaciones](#)

Nota: Los cambios que usted haga en esta pantalla no se aplicarán hasta que usted haga clic en el botón 'Comenzar instalación'.

ESPACIO DISPONIBLE: 1,82 MiB
 ESPACIO TOTAL: 13 GiB

[1 - dispositivo de almacenamiento seleccionado](#) [Descartar todos los cambios](#)

Configuración de red

- Interface de red NAT (para salida a internet).
- Interface interna para las conexiones entre los servidores.
 - IP 192.168.1.200/24
 - Servicio de Aplicación.

```
login as: admin
admin@127.0.0.1's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Thu Aug  1 13:13:48 2024 from 10.0.2.2
[admin@localhost ~]$
[admin@localhost ~]$
[admin@localhost ~]$ 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
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:34:4e:78 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 67980sec preferred_lft 67980sec
    inet6 fe80::a00:27ff:fe34:4e78/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:81:d4:20 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.200/24 brd 192.168.1.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::8662:6806:cf3b:987e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[admin@localhost ~]$
```

Creamos el usuario soporte y lo asignamos al grupo administrador

```
[admin@localhost ~]$ useradd soporte
useradd: Permission denied.
useradd: no se pudo bloquear /etc/passwd, inténtelo de nuevo.
[admin@localhost ~]$ sudo useradd soporte
[sudo] password for admin:
[admin@localhost ~]$
[admin@localhost ~]$ sudo passwd soporte
Cambiando la contraseña del usuario soporte.
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: todos los tokens de autenticación se actualizaron exitosamente.
[admin@localhost ~]$
[admin@localhost ~]$
[admin@localhost ~]$
[admin@localhost ~]$ usermod -aG wheel soporte
usermod: Permission denied.
usermod: no se pudo bloquear /etc/passwd, inténtelo de nuevo.
[admin@localhost ~]$ sudo usermod -aG wheel soporte
[admin@localhost ~]$
```

Generamos las claves SSH pública/privada

```
[soporte@localhost ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/soporte/.ssh/id_ed25519):
Created directory '/home/soporte/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/soporte/.ssh/id_ed25519
Your public key has been saved in /home/soporte/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:YK5sHtlgFsUtykxMmLHoVUllXv274zKmYTLBdeqddSM soporte@localhost.localdomain
The key's randomart image is:
+--[ED25519 256]--+
|      .B++o..  .  |
|      .oo+.oo..  .  |
|      . o++..   .  .  |
|      . . o+o. . o  . |
|      .  = So . E oo |
|      . = o  o  .+.o |
|      = . .o + o  . |
|      o .   + .+ o  |
|      .      .o +.. |
+-----[SHA256]-----+
[soporte@localhost ~]$ ls -la /home/soporte/.ssh/
..  id_ed25519  id_ed25519.pub
[soporte@localhost ~]$
[soporte@localhost ~]$
```

Ubuntu

- Arranca desde el medio de instalación: Inserta el medio (USB/DVD) y arranca el sistema.
- Selección de idioma y teclado: Elige tu idioma y disposición del teclado.
- Configuración de red: Configura la red o redes.
- Configuración del disco: Define las particiones del disco.
- Creación de usuario: Crea un usuario con permisos de administrador.
- Finalización: Completa la instalación y reinicia el sistema.

Personalización de particiones de disco:

Disco de 13GB, 1CPU y 2GB RAM:

- Partición de 1GB para /boot
- LVM de 7GB para /
- LVM de 3GB para /var
- LVM de 2GB para SWAP


```
soporte@ubuntu-db ~> df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                     392M      1,1M   391M    1% /run
/dev/mapper/ubuntu--vg-lv--0 7,0G      2,7G    4,3G   39% /
tmpfs                     2,0G          0    2,0G    0% /dev/shm
tmpfs                     5,0M          0    5,0M    0% /run/lock
/dev/mapper/ubuntu--vg-lv--1 3,0G      848M    2,2G   29% /var
/dev/sda2                 960M     146M    815M   16% /boot
tmpfs                     392M      12K    392M    1% /run/user/1001

soporte@ubuntu-db ~> sudo lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
lv-0 ubuntu-vg -wi-ao---- 7,00g
lv-1 ubuntu-vg -wi-ao---- 3,00g
lv-2 ubuntu-vg -wi-ao---- 2,00g
soporte@ubuntu-db ~>
```

Configuración de red

- Interface de red NAT (para salida a internet).
- Interface interna para las conexiones entre los servidores.
 - IP 192.168.1.100/24.
 - Servicio de base de datos.

```
soporte@ubuntu-db ~> ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe06:8877 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:06:88:77 txqueuelen 1000 (Ethernet)
    RX packets 78871 bytes 115376577 (115.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14534 bytes 1045273 (1.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe69:6c3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:69:06:c3 txqueuelen 1000 (Ethernet)
    RX packets 5909 bytes 4717763 (4.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2267 bytes 302746 (302.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 224 bytes 23608 (23.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 224 bytes 23608 (23.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Creamos el usuario soporte y lo asignamos al grupo administrador

```
soporte@ubuntu-db ~> cut -d: -f1 /etc/passwd | grep soporte
soporte
soporte@ubuntu-db ~>
```

Generamos las claves SSH pública/privada


```
soporte@localhost ~> ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/soporte/.ssh/id_ed25519):
Created directory '/home/soporte/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/soporte/.ssh/id_ed25519
Your public key has been saved in /home/soporte/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:S57i1mR5QpzZTIS2HP+2b2CZqMJZVhnK4cc+hF3X3GU soporte@localhost.localdomain
The key's randomart image is:
+--[ED25519 256]--+
|      o.      oE|
|      = o .  .+|
|      * / + .  |
|      % @      |
|      .S* o o   |
|      oBo= B    |
|      ..B++ + o |
|      .=.o . .  |
|      ...      o.|
+-----[SHA256]-----+
```

```
soporte@ubuntu-db:~$ ll ~/.ssh
total 8
drwx----- 2 soporte soporte 46 ago 1 23:34 ./
drwxr-x--- 4 soporte soporte 116 ago 1 23:40 ../
-rw----- 1 soporte soporte 411 ago 1 23:34 id_ed25519
-rw-r--r-- 1 soporte soporte 99 ago 1 23:34 id_ed25519.pub
```

CentOS con Ansible

Se instala los paquetes de Ansible en un servidor CentOS.

```
soporte@localhost ~> 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! ✨ ✨ ✨
soporte@localhost ~>
```

```
soporte@localhost ~> pipx inject ansible-core argcomplete
injected package argcomplete into venv ansible-core
done! ✨ ✨ ✨
soporte@localhost ~>
```

Se crean claves SSH pública/privada con el esquema de firma ed25519 y se pegan las claves públicas en los servidores de aplicación y base de datos.

`$ssh-copy-id <Servidor remoto>`

Evidencia del pegado de clave pública en CentOS de aplicación:

```
soporte@localhost ~> ssh-copy-id 192.168.1.200
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/soporte/
/id_ed25519.pub"
The authenticity of host '192.168.1.200 (192.168.1.200)' can't be established
ED25519 key fingerprint is SHA256:1LpfSUMmA/98uaKVtzzzyiWs3vqGM1f0+QL+Ly0CdxY
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 pro
d now it is to install the new keys
soporte@192.168.1.200's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '192.168.1.200'"
and check to make sure that only the key(s) you wanted were added.

soporte@localhost ~>
```

Evidencia del pegado de clave pública en Ubuntu de base de datos:

```
soporte@localhost ~> ssh-copy-id 192.168.1.100
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/soporte/.ssh/id_ed25519.pub"
The authenticity of host '192.168.1.100 (192.168.1.100)' can't be established.
ED25519 key fingerprint is SHA256:zG9KmMA2H3t6DcnLIuGhlaqaHW/ghzGVfcqQ5kPHgSU.
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
soporte@192.168.1.100's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '192.168.1.100'"
and check to make sure that only the key(s) you wanted were added.
```

Entorno Ansible

Las configuraciones se escriben en /home/soporte, usuario creado para la automatización y con permisos de ejecución.

Instalación de ansible

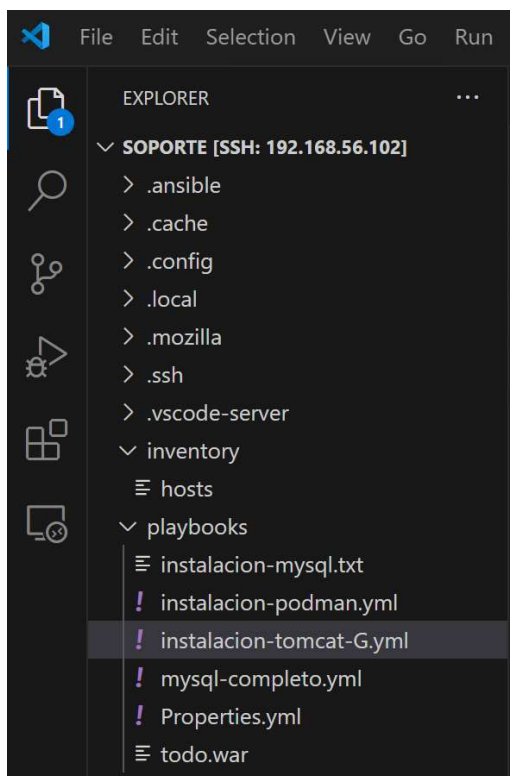
```
soporte@localhost ~> ansible --version
ansible [core 2.14.17]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/soporte/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /home/soporte/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.19 (main, Jul 18 2024, 00:00:00) [GCC 11.4.1 20231218 (Red Hat 11.4.1-3)] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
```

Validación de conexión entre ansible y remotos

```
soporte@localhost ~> ansible -i inventory/hosts linux -m shell -a "uname -a"
ubuntu-db | CHANGED | rc=0 >>
Linux ubuntu-db 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul 5 21:49:14 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
centos-app | CHANGED | rc=0 >>
Linux centos-app 5.14.0-480.el9.x86_64 #1 SMP PREEMPT_DYNAMIC Fri Jul 12 20:45:27 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
soporte@localhost ~>
```

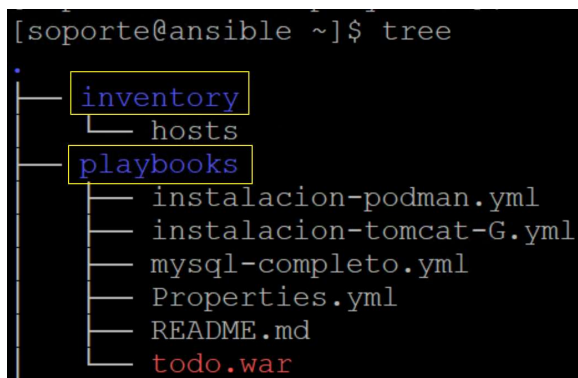
Herramienta Editor de código

Para gestionar de una forma ordenada la configuración de los playbooks de ansible, se instaló la herramienta Visual Studio Code, software de edición de código fuente gratuito. Se agrega plugin de Ansible y SSH para mejorar la tarea a desarrollar.



Se crean 2 directorios de configuración en el /home del usuario soporte:

- inventory.
- playbooks.

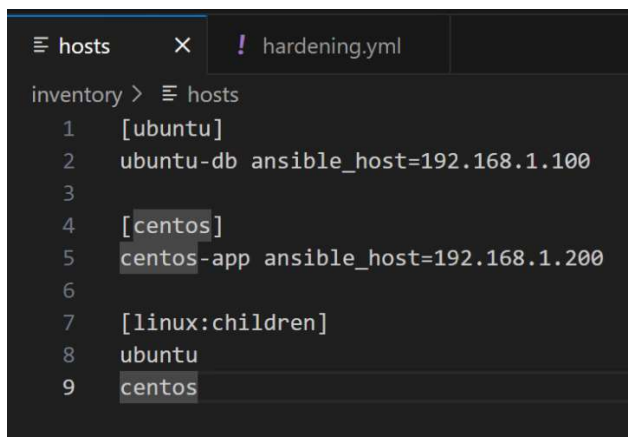


Configuración de Inventario

Dentro del directorio inventory, se creó un archivo nombrado hosts, este incluye las direcciones IP de los servidores de aplicación y base de datos, así como un grupo entre los 2 equipos.

- Nombre → ubuntu
 - IP 192.168.1.100.
 - Nombre del host → ubuntu-db.
- Nombre → centos
 - IP 192.168.1.200.

- Nombre del host → centos-app.



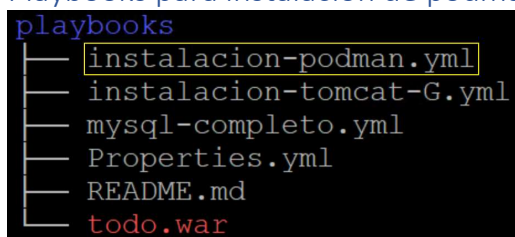
```

inventory > hosts
1  [ubuntu]
2  ubuntu-db ansible_host=192.168.1.100
3
4  [centos]
5  centos-app ansible_host=192.168.1.200
6
7  [linux:children]
8  ubuntu
9  centos
  
```

Configuración de Playbooks

Dentro del directorio playbooks, se encuentran los archivos YAML (YAML Ain't Markup Language), lenguaje de configuración utilizado por Ansible para configuración de automatización de equipos remotos, permitiendo establecer desde un hardening, modificar parámetros, instalar servicios, entre otras tareas más.

Playbooks para instalación de podman



```

playbooks
├── instalacion-podman.yml
├── instalacion-tomcat-G.yml
├── mysql-completo.yml
├── Properties.yml
├── README.md
└── todo.war
  
```

Configuración del playbook para el podman:

```

---
- name: Instalacion de Podman

  hosts: centos

  become: yes

  tasks:

    - name: Instalar el paquete Podman

      ansible.builtin.yum:

        name: podman

        state: present

    - name: Instalar herramientas adicionales para Podman (opcional)

      yum:
  
```

name:

- podman-docker
- skopeo

state: present

- name: Verificar la versión instalada

command: podman --version

register: podman_version

- name: Iniciar el servicio Podman

systemd:

name: podman.socket

state: started

enabled: yes

- name: Verificar el estado del servicio Podman

command: systemctl is-active podman

register: podman_status

ignore_errors: yes

- name: Habilitar y arrancar el servicio Podman

systemd:

name: podman

enabled: yes

state: started

when: podman_status.stdout != 'active'

- name: Mostrar la versión instalada

ansible.builtin.debug:

msg: "La versión de Podman es: {{ podman_version.stdout }}"

...


```
soporte@ansible ~ [1]> ansible-playbook -i inventory/hosts playbooks/instalacion-podman.yml --ask-become-pass
BECOME password:

PLAY [Instalacion de Podman] *****

TASK [Gathering Facts] *****
ok: [centos-app]

TASK [Instalar el paquete Podman] *****
ok: [centos-app]

TASK [Verificar la versión instalada] *****
changed: [centos-app]

TASK [Mostrar la versión instalada] *****
ok: [centos-app] => {
  "msg": "La versión de Podman es: podman version 5.1.2"
}

PLAY RECAP *****
centos-app                : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

soporte@ansible ~>
```

Playbooks para instalación de tomcat

```
playbooks
├── instalacion-podman.yml
├── instalacion-tomcat-G.yml
├── mysql-completo.yml
├── Properties.yml
├── README.md
└── todo.war
```

Configuración del playbook para el tomcat y archivo todo.war:

Para esta configuración se creó varias tareas en un orden específico:

1. Crea usuario tomcat
2. Confirmar la instalación de podman.
3. Instalar Git.
4. Crear un directorio para almacenar el git a descargar.
5. Clonar el Git.
6. Configurar el archivo Containerfile.
7. Agregar parámetros al archivo Containerfile.
8. Validamos salida a internet para instalar curl.
9. Agregar el archivo todo.war a la ruta específica.
10. Levantar tomcat.

- name: Playbook para instalar Podman y configurar Tomcat en CentOS

hosts: centos

become: yes

tasks:

- name: Crear usuario tomcat

user:

```
name: tomcat
comment: "Usuario para correr Tomcat"
home: /home/tomcat
shell: /bin/bash
createhome: yes
```

- name: Instalar Podman (DNF Backend)

```
dnf:
  name: podman
  state: present
```

- name: Instalar Git

```
dnf:
  name: git
  state: present
```

- name: Crear directorio para almacenar el git

```
file:
  path: /home/tomcat/contenedor
  state: directory
  owner: tomcat
  group: tomcat
  mode: '0755'
```

- name: Clonar el repositorio de Git

```
# git:
# repo: https://github.com/emverdes/docker-tomcat-tutorial.git
# dest: /home/tomcat/contenedor/docker-tomcat-tutorial
# clone: yes
```

- name: Configurar archivo Containerfile para Docker Tomcat

```
hosts: centos
become: yes
tasks:
```

- name: Asegurarse de que el directorio existe

file:

path: /home/tomcat/contenedor/docker-tomcat-tutorial

state: directory

owner: tomcat

group: tomcat

mode: '0755'

- name: Crear archivo Containerfile con la configuración necesaria

copy:

dest: /home/tomcat/contenedor/docker-tomcat-tutorial/Containerfile

content: |

FROM quay.io/centos/centos:stream9

#LABEL maintainer="emverdes@gmail.com"

Instalar Java, iputils y curl con la opción --allowerase

RUN dnf install -y java-11-openjdk iputils curl --allowerase && dnf clean all

RUN mkdir /opt/tomcat

Probar conectividad de red con curl

#RUN curl -I https://www.google.com

Probar conectividad de red con ping

RUN ping -c 4 google.com

Descargar y extraer Tomcat

RUN curl -v -O https://ftp.unicamp.br/pub/apache/tomcat/tomcat-8/v8.5.73/bin/apache-tomcat-8.5.73.tar.gz && \

tar xvfz apache-tomcat-8.5.73.tar.gz && \

mv apache-tomcat-8.5.73/* /opt/tomcat/

Añadir archivos WAR a la carpeta de aplicaciones web de Tomcat

ADD sample.war /opt/tomcat/webapps/

ADD todo.war /opt/tomcat/webapps/

Añadir el archivo de propiedades de configuración

ADD app.properties /opt/config/

Listar el contenido del directorio de aplicaciones web de Tomcat para verificar

RUN ls -lah /opt/tomcat/webapps

Exponer el puerto 8080

EXPOSE 8080

Comando para ejecutar Tomcat

CMD ["/opt/tomcat/bin/catalina.sh", "run"]

owner: tomcat

group: tomcat

mode: '0644'

- name: Copiar archivo todo.war

copy:

src: /home/soporte/playbooks/todo.war

dest: /home/tomcat/contenedor/docker-tomcat-tutorial

owner: tomcat

group: tomcat

mode: '0644'

...

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  2
TASK [Gathering Facts] *****
ok: [centos-app]

TASK [Crear usuario tomcat] *****
changed: [centos-app]

TASK [Instalar Podman (DNF Backend)] *****
ok: [centos-app]

TASK [Instalar Git] *****
ok: [centos-app]

TASK [Crear directorio para almacenar el git] *****
changed: [centos-app]

TASK [Clonar el repositorio de Git] *****
changed: [centos-app]

PLAY [Configurar archivo Containerfile para Docker Tomcat] *****

TASK [Gathering Facts] *****
ok: [centos-app]

TASK [Asegurarse de que el directorio existe] *****
changed: [centos-app]

TASK [Crear archivo Containerfile con la configuración necesaria] *****
changed: [centos-app]

TASK [Copiar archivo todo.war] *****
changed: [centos-app]

TASK [Añadir archivos y configuraciones al Containerfile] *****
changed: [centos-app]

PLAY RECAP *****
centos-app      : ok=11  changed=7  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

soporte@ansible ~>

```

Playbooks para instalación de base de datos

```

playbooks
├── instalacion-podman.yml
├── instalacion-tomcat-G.yml
├── mysql-completo.yml
├── Properties.yml
├── README.md
└── todo.war

```

Configuración del playbook para la base de datos mysql:

Se agrega configuración en la ruta /etc/mysql/my.cnf para que el servicio acepte desde cualquier IP las conexiones que viene desde la aplicación del Centos.

- name: MySQL

hosts: ubuntu

become: yes

tasks:

- name: Crear archivo /root/.my.cnf con las credenciales de MySQL

copy:

dest: /root/.my.cnf

```
content: |  
[client]  
user=root  
password=Laboratorio2024  
owner: root  
group: root  
mode: '0600'
```

- name: Instalar dependencias de Python

```
apt:  
  name: python3-pip  
  state: present
```

- name: Instalar MySQL

```
apt:  
  name: mysql-server  
  state: present
```

- name: Instalar MySQLClient

```
apt:  
  name: mysql-client  
  state: present
```

- name: Instalar Python

```
apt:  
  name: python3-mysqldb  
  state: present
```

- name: Iniciar el servicio MySQL y asegurarse de que esté habilitado

```
service:  
  name: mysql  
  state: started  
  enabled: yes
```


- name: Establecer contraseña de root para MySQL

mysql_user:

name: root

password: "Laboratorio2024"

login_unix_socket: /var/run/mysqld/mysqld.sock

host: localhost

state: present

- name: Asegurarse de que la cuenta root tenga todos los privilegios

mysql_user:

name: root

host: localhost

priv: '*.*:ALL,GRANT'

login_unix_socket: /var/run/mysqld/mysqld.sock

state: present

- name: Crear base de datos 'todo'

mysql_db:

name: todo

state: present

login_user: root

login_password: "Laboratorio2024"

- name: Crear tabla 'users'

mysql_query:

login_user: root

login_password: "Laboratorio2024"

query: |

USE todo;

CREATE TABLE IF NOT EXISTS `users` (

`id` int(3) NOT NULL AUTO_INCREMENT,

`first_name` varchar(20) DEFAULT NULL,

`last_name` varchar(20) DEFAULT NULL,

`username` varchar(250) DEFAULT NULL,

```
`password` varchar(20) DEFAULT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

- name: Crear tabla 'todos'

mysql_query:

login_user: root

login_password: "Laboratorio2024"

query: |

```
USE todo;
```

```
CREATE TABLE IF NOT EXISTS `todos` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `description` varchar(255) DEFAULT NULL,  
  `is_done` bit(1) NOT NULL,  
  `target_date` datetime(6) DEFAULT NULL,  
  `username` varchar(255) DEFAULT NULL,  
  `title` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

- name: Crear usuario 'todo' y otorgar privilegios

mysql_user:

name: todo

host: '%'

password: "Prueba2024!"

priv: 'todo.*:ALL'

login_user: root

login_password: "Laboratorio2024"

state: present

- name: Refrescar privilegios de MySQL

mysql_query:

login_user: root

login_password: "Laboratorio2024"

query: "FLUSH PRIVILEGES;"

- name: Configure MySQL my.cnf

blockinfile:

path: /etc/mysql/my.cnf

block: |

Archivo de configuración global de MySQL

[mysqld]

bind-address = 0.0.0.0

marker: "# {mark} ANSIBLE MANAGED BLOCK"

notify: restart mysql

- name: Open MySQL port 3306 in UFW

ufw:

rule: allow

port: 3306

proto: tcp

handlers:

- name: restart mysql

service:

name: mysql

state: restarted

...

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  2

TASK [Instalar dependencias de Python] *****
changed: [ubuntu-db]

TASK [Instalar MySQL] *****
changed: [ubuntu-db]

TASK [Instalar MySQLClient] *****
changed: [ubuntu-db]

TASK [Instalar Python] *****
changed: [ubuntu-db]

TASK [Iniciar el servicio MySQL y asegurarse de que esté habilitado] *****
ok: [ubuntu-db]

TASK [Establecer contraseña de root para MySQL] *****
changed: [ubuntu-db]

TASK [Asegurarse de que la cuenta root tenga todos los privilegios] *****
ok: [ubuntu-db]

TASK [Crear base de datos 'todo'] *****
changed: [ubuntu-db]

TASK [Crear tabla 'users'] *****
ok: [ubuntu-db]

TASK [Crear tabla 'todos'] *****
ok: [ubuntu-db]

TASK [Crear usuario 'todo' y otorgar privilegios] *****
changed: [ubuntu-db]

TASK [Refrescar privilegios de MySQL] *****
ok: [ubuntu-db]

PLAY RECAP *****
ubuntu-db                : ok=14   changed=8   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

soporte@ansible ~>

```

Playbooks para instalación de properties

```

playbooks
├── instalacion-podman.yml
├── instalacion-tomcat-G.yml
├── mysql-completo.yml
├── Properties.yml
├── README.md
└── todo.war

```

Configuración del playbook para el archivo properties

Parámetros para la conexión entre la aplicación y la base de datos.

- name: Configurar archivo app.properties para Tomcat

hosts: centos

become: yes

tasks:

- name: Crear archivo app.properties con las configuraciones

copy:

dest: /home/tomcat/contenedor/docker-tomcat-tutorial/app.properties

content: |

tipoDB=mysql

jdbcURL=jdbc:mysql://192.168.1.100:3306/todo

jdbcUsername=todo

jdbcPassword=Prueba2024!

owner: tomcat

group: tomcat

mode: '0644'

...

```
soporte@ansible ~$ ansible-playbook -i inventory/hosts playbooks/Properties.yml --ask-become-pass
BECOME password:

PLAY [Configurar archivo app.properties para Tomcat] *****

TASK [Gathering Facts] *****
ok: [centos-app]

TASK [Crear archivo app.properties con las configuraciones] *****
changed: [centos-app]

PLAY RECAP *****
centos-app                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

soporte@ansible ~$
```

Copiado del archivo de aplicación todo.war

Esta configuración está dentro del playbooks donde se instaló el tomcat, nombrado anteriormente:

```
- name: Copiar archivo todo.war
  copy:
    src: /home/soporte/playbooks/todo.war
    dest: /home/tomcat/contenedor/docker-tomcat-tutorial
    owner: tomcat
    group: tomcat
    mode: '0644'
```

```
playbooks
├── instalacion-podman.yml
├── instalacion-tomcat-G.yml
├── mysql-completo.yml
├── Properties.yml
├── README.md
└── todo.war
```

Creación de Servicio

Se un YML el cual genera un servicio llamado demo1 el cual permite que levante automáticamente la APP en caso de un reinicio del sistema.

- name: Configurar inicio del servicio con SystemD

hosts: centos

become: yes

tasks:

- name: Crear el archivo de servicio demo1.service

copy:

dest: /etc/systemd/system/demo1.service

content: |

[Unit]

Description=Ejecutar el contenedor Tomcat demo1

After=network.target

[Service]

Type=simple

ExecStart=/usr/bin/podman run --replace -p 8080:8080 --name demo1
tomcat-demo:1

ExecStop=/usr/bin/podman stop demo1

ExecStopPost=/usr/bin/podman rm demo1

WorkingDirectory=/home/tomcat/contenedor/docker-tomcat-tutorial

Restart=always

RestartSec=10

User=soporte

[Install]

WantedBy=multi-user.target

- name: Recargar systemd para aplicar cambios

command: systemctl daemon-reload


```

~
~
lines 1-21/21 (END)...skipping...
● demol.service - Ejecutar el contenedor Tomcat demol
   Loaded: loaded (/etc/systemd/system/demol.service; enabled; preset: disabled)
   Active: active (running) since Tue 2024-08-06 15:30:58 -03; 14s ago
     Main PID: 34477 (podman)
        Tasks: 11 (limit: 48773)
       Memory: 36.2M
          CPU: 197ms
    CGroup: /system.slice/demol.service
            └─34477 /usr/bin/podman run --replace -p 8080:8080 --name demol tomcat-demo:1
              └─34489 catatonit -P

ago 06 15:30:58 centos-app systemd[1]: Started Ejecutar el contenedor Tomcat demol.
ago 06 15:30:58 centos-app podman[34485]: time="2024-08-06 15:30:58-03:00" level=error msg="Ref>
ago 06 15:30:58 centos-app podman[34485]: 2024-08-06 15:30:58.202443606 -0300 -03 m=+0.07003159>
ago 06 15:30:58 centos-app podman[34485]: 2024-08-06 15:30:58.587380285 -0300 -03 m=+0.45496824>
ago 06 15:30:58 centos-app podman[34485]: 2024-08-06 15:30:58.562882338 -0300 -03 m=+0.43047033>
ago 06 15:30:58 centos-app podman[34485]: 2024-08-06 15:30:58.626734791 -0300 -03 m=+0.49432275>
ago 06 15:30:58 centos-app podman[34485]: 2024-08-06 15:30:58.742193398 -0300 -03 m=+0.60978139>
ago 06 15:30:58 centos-app podman[34485]: 2024-08-06 15:30:58.749672882 -0300 -03 m=+0.61726085>
ago 06 15:30:58 centos-app podman[34485]: 2024-08-06 15:30:58.75453689 -0300 -03 m=+0.622124862>
ago 06 15:30:58 centos-app podman[34485]: NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.b>
~

```

Si vamos a CentOS vemos creado el servicio.

```

soporte@centos-app ~ [SIGINT]> /bin/bash
[soporte@centos-app ~]$ cat /etc/systemd/system
system/
system.conf
[soporte@centos-app ~]$ cat /etc/systemd/system/demol
cat: /etc/systemd/system/demol: No existe el fichero o el directorio
[soporte@centos-app ~]$ cat /etc/systemd/system/demol.service
[Unit]
Description=Ejecutar el contenedor Tomcat demol
After=network.target

[Service]
Type=simple
ExecStart=/usr/bin/podman run --replace -p 8080:8080 --name demol tomcat-demo:1
ExecStop=/usr/bin/podman stop demol
ExecStopPost=/usr/bin/podman rm demol
WorkingDirectory=/home/tomcat/contenedor/docker-tomcat-tutorial
Restart=always
RestartSec=10
User=soporte

[Install]
WantedBy=multi-user.target
[soporte@centos-app ~]$ █

```

Evidencia de funcionamiento

Para validar el funcionamiento se forwardo puertos desde el Hipervisor Virtualbox, puerto 8080 del tomcat → cliente puerto 8888.

http://127.0.0.1:8888/todo/



Todo App Login Form

User Name:

Password:

Submit

Pruebas de funcionamiento:

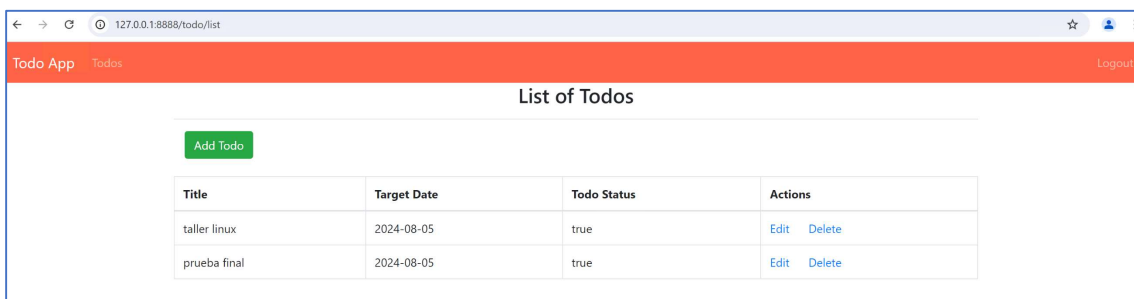


Todo App Todos

List of Todos

Add Todo

| Title | Target Date | Todo Status | Actions |
|--------------|-------------|-------------|-------------|
| taller linux | 2024-08-05 | true | Edit Delete |



Todo App Todos

List of Todos

Add Todo

| Title | Target Date | Todo Status | Actions |
|--------------|-------------|-------------|-------------|
| taller linux | 2024-08-05 | true | Edit Delete |
| prueba final | 2024-08-05 | true | Edit Delete |

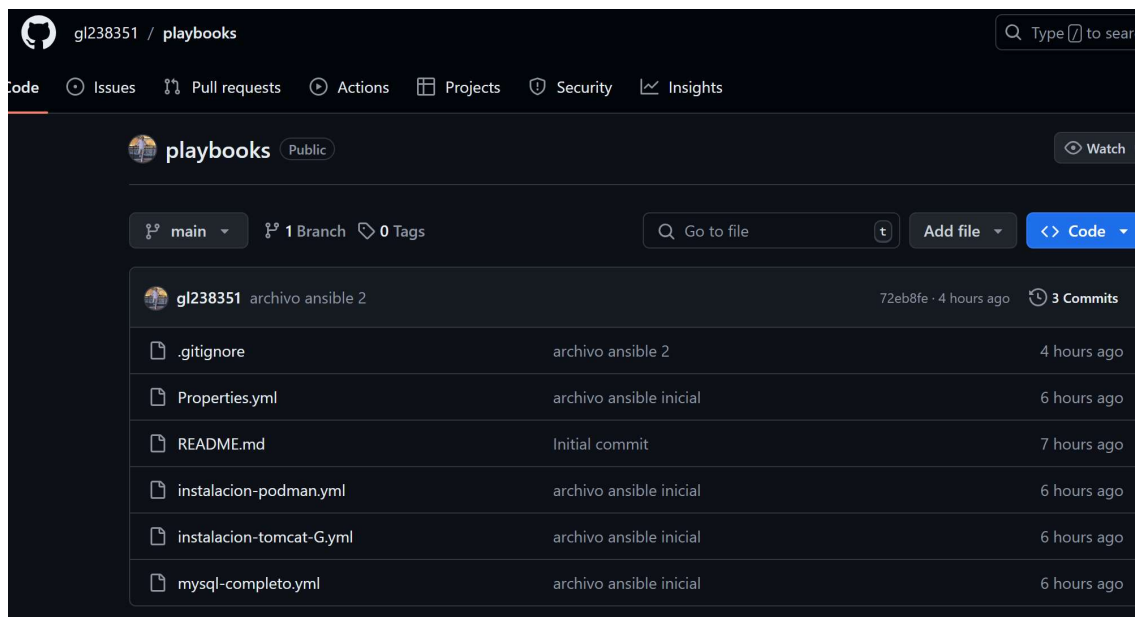
Validación del login a la base y creación de usuario y tarea.

```
The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.
com.mysql.cj.jdbc.ClientPreparedStatement: INSERT INTO users (first_name, last_name, username, password) VALUES ('Gabriel', 'Larrosa', 'Gabriel', '1234');
com.mysql.cj.jdbc.ClientPreparedStatement: INSERT INTO users (first_name, last_name, username, password) VALUES ('Nicolas', 'Soarez', 'Nicolas', '1234');
com.mysql.cj.jdbc.ClientPreparedStatement: select * from users where username = 'nicolas' and password = '1234'
INSERT INTO todos (title, username, description, target_date, is_done) VALUES (?, ?, ?, ?, ?);
com.mysql.cj.jdbc.ClientPreparedStatement: INSERT INTO todos (title, username, description, target_date, is_done) VALUES ('taller linux', null, 'taller', '2024-08-05', 1);
com.mysql.cj.jdbc.ClientPreparedStatement: select * from todos
com.mysql.cj.jdbc.ClientPreparedStatement: select * from users where username = 'nicolas' and password = '1234'
INSERT INTO todos (title, username, description, target_date, is_done) VALUES (?, ?, ?, ?, ?);
com.mysql.cj.jdbc.ClientPreparedStatement: INSERT INTO todos (title, username, description, target_date, is_done) VALUES ('prueba final', null, 'taller', '2024-08-05', 1);
com.mysql.cj.jdbc.ClientPreparedStatement: select * from todos
```

Elevar al repositorio Git

Se crea un nuevo repositorio en Github donde se almacenan los archivos de configuración playbooks e inventory.

<https://github.com/gl238351/playbooks>



Se realiza:

```
$ git status
```

Se valida que los archivos YAML no estan en seguimiento y se ejecuta:

```
$ git add *.yml →>grega los archivos a git interno.
```

```
$ git commit -m "cargar inicial YML" → realizamos un commit de los archivos agregados.
```

```
$ git push origin main → realiza el envío de los archivos a repositorio Github
```

```
[main 5814d5d] archivo ansible inicial
4 files changed, 294 insertions(+)
create mode 100644 Properties.yml
create mode 100644 instalacion-podman.yml
create mode 100644 instalacion-tomcat-G.yml
create mode 100644 mysql-completo.yml
[soporte@ansible playbooks]$ git status
En la rama main
Tu rama está adelantada a 'origin/main' por 1 commit.
(usa "git push" para publicar tus commits locales)

nada para hacer commit, el árbol de trabajo está limpio
[soporte@ansible playbooks]$ git history
git: 'history' no es un comando de git. Mira 'git --help'.
[soporte@ansible playbooks]$ git log
commit 5814d5d4c4b3042eb945562acb030f8d20e55ab4 (HEAD -> main)
Author: Gabriel <GL238351@fi365.ort.edu.uy>
Date: Mon Aug 5 14:34:52 2024 -0300

    archivo ansible inicial

commit 9db747ba9203152a37d0dd0db8281f7fal42d36e (origin/main, origin/HEAD)
Author: Gabriel Edgardo Larrosa de los Santos <GL238351@fi365.ort.edu.uy>
Date: Mon Aug 5 13:46:35 2024 -0300

    Initial commit
[soporte@ansible playbooks]$ git pull origin main
Desde github.com:gl238351/playbooks
 * branch          main      -> FETCH_HEAD
Ya está actualizado.
[soporte@ansible playbooks]$ git push origin main
Enumerando objetos: 7, listo.
Contando objetos: 100% (7/7), listo.
Compresión delta usando hasta 3 hilos.
Comprimiendo objetos: 100% (6/6), listo.
Escribiendo objetos: 100% (6/6), 3.12 KiB | 1.56 MiB/s, listo.
Total 6 (delta 0), reusados 0 (delta 0), pack-reusados 0
To github.com:gl238351/playbooks.git
   9db747b..5814d5d  main -> main
[soporte@ansible playbooks]$
```

```
soporte@ansible:~/playbooks

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que será confirmado)
Properties.yml
instalacion-podman.yml
instalacion-tomcat-G.yml
mysql-completo.yml

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
[soporte@ansible playbooks]$ git add *.yml
[soporte@ansible playbooks]$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

Cambios a ser confirmados:
(usa "git restore --staged <archivo>..." para sacar del área de stage)
nuevos archivos: Properties.yml
nuevos archivos: instalacion-podman.yml
nuevos archivos: instalacion-tomcat-G.yml
nuevos archivos: mysql-completo.yml

[soporte@ansible playbooks]$ git commit -m "archivo ansible inicial"
Identidad del autor desconocido

*** Por favor cuéntame quién eres.

Ejecuta

git config --global user.email "you@example.com"
git config --global user.name "Tu Nombre"

para configurar la identidad por defecto de tu cuenta.
Omite --global para configurar tu identidad solo en este repositorio.

fatal: no es posible auto-detectar la dirección de correo (se obtuvo 'soporte@ansible.(none)')
[soporte@ansible playbooks]$ git config --global user.email "GL238351@fi365.ort.edu.uy"
[soporte@ansible playbooks]$ git config --global user.name "Gabriel"
[soporte@ansible playbooks]$
[soporte@ansible playbooks]$ git commit -m "archivo ansible inicial"
```

Nota:

Como no queremos subir algunos archivos al repositorio de github, podemos ignorarlos gestionando el archivo .gitignore. En este caso, lo realizamos para los archivos *.war

```
soporte@ansible:~/playbooks
To github.com:gl238351/playbooks.git
 9db747b..5814d5d main -> main
[soporte@ansible playbooks]$ touch .gitignore
[soporte@ansible playbooks]$ vi .gitignore
[soporte@ansible playbooks]$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que será confirmado)
.gitignore

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
[soporte@ansible playbooks]$ git add .gitignore
[soporte@ansible playbooks]$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

Cambios a ser confirmados:
(usa "git restore --staged <archivo>..." para sacar del área de stage)
nuevos archivos: .gitignore

[soporte@ansible playbooks]$ git commit -m "archivo ansible 2"
[main 72eb8fe] archivo ansible 2
1 file changed, 1 insertion(+)
create mode 100644 .gitignore
[soporte@ansible playbooks]$ git status
En la rama main
Tu rama está adelantada a 'origin/main' por 1 commit.
(usa "git push" para publicar tus commits locales)

nada para hacer commit, el árbol de trabajo está limpio
[soporte@ansible playbooks]$ git push origin main
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 3 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 284 bytes | 142.00 KiB/s, listo.
Total 3 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:gl238351/playbooks.git
 5814d5d..72eb8fe main -> main
[soporte@ansible playbooks]$ cp ../playbooks.bkp/*.war .
[soporte@ansible playbooks]$ ls
instalacion-podman.yml  mysql-completo.yml  README.md
instalacion-tomcat-G.yml  Properties.yml      todo.war
[soporte@ansible playbooks]$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

nada para hacer commit, el árbol de trabajo está limpio
[soporte@ansible playbooks]$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

nada para hacer commit, el árbol de trabajo está limpio
[soporte@ansible playbooks]$ git status
```