

Tarea 6 Enunciado: Creación de un rol para MySQL

En esta tarea se propone la creación de un rol para la instalación de MySQL en `srv1`.

Para el ejercicio, crearemos el rol con nombre `mysql` y tendremos los siguientes ficheros:

- **invoke_mysql.yaml:**

Playbook principal que invoca el rol `mysql` sobre el servidor `srv1`.

Carga las contraseñas desde el archivo cifrado `mypasswords_vault.yaml` y pasa las variables necesarias al rol (como `mysql_root_password` y los usuarios a crear).

```
---
- name: Instalar MYSQL con rol
  hosts: srv1
  become: yes

  tasks:
    - name: Load vault mysql users passwords
      ansible.builtin.include_vars:
        file: mypasswords_vault.yaml

    - name: Invoke mysql role
      ansible.builtin.include_role:
        name: mysql
      tags:
        - instalacion_mysql
        - usuarios_mysql
    vars:
      mysql_root_password: "{{ vault_mysql_root_pw }}"
      mysql_users:
        bob:
          password: "{{ vault_mysql_users_passwords.bob }}"
          priv:
            'db1.*': 'ALL,GRANT'
            'db2.*': 'ALL,GRANT'
        john:
          password: "{{ vault_mysql_users_passwords.john }}"
          priv:
            'db3.*': 'ALL,GRANT'
```

- **mypasswords_vault.yaml:**

Archivo que contiene las contraseñas de root y de los usuarios de las bases de datos, que serán `bob` y `john`.

Su contenido es el siguiente:

```
vault_mysql_root_pw: "toor"
vaul
```

```
bob: "bobpw"
john: "johnpw"
```

- **requirements.yaml:**

Incluye la colección necesaria para poder utilizar el módulo `community.mysql.mysql_user` en el fichero `/roles/mysql/tasks/main.yml`.

Es necesario instalar esta colección antes de ejecutar el playbook.

```
#ansible-galaxy install -r requirements.yaml
collections:
  - community.mysql
```

- **/roles/mysql/vars/main.yml:**

Contiene las variables necesarias para la instalación de MySQL, como la versión del paquete, las configuraciones iniciales y las configuraciones del servidor.

```
#SPDX-License-Identifier: MIT-0
---

# vars file for mysql

## Servicio
software: mysql-server
servicio: mysql

## Versión de paquete apt
version_package: "mysql-apt-config_0.8.34-1_all.deb"

## Paquetes MariaDB conflictivos
mariadb_conflict_packages:
  - mariadb-server
  - mariadb-server-core

## Variables de configuración de MySQL
# Variables de configuración de MySQL (diccionario de diccionarios)
config_initial:
  mysql-apt-config/repo-codename:
    value: "bookworm"
    vtype: "select"
    name: "mysql-apt-config"

  mysql-apt-config/repo-distro:
    value: "debian"
    vtype: "select"
    name: "mysql-apt-config"

  mysql-apt-config/repo-url:
    value: "http://repo.mysql.com/apt"
    vtype: "string"
```

```
name: "mysql-apt-config"

mysql-apt-config/select-connectors:
  value: "Enabled"
  vtype: "select"
  name: "mysql-apt-config"

mysql-apt-config/select-product:
  value: "Ok"
  vtype: "select"
  name: "mysql-apt-config"

mysql-apt-config/select-server:
  value: "mysql-8.0"
  vtype: "select"
  name: "mysql-apt-config"

mysql-apt-config/unsupported-platform:
  value: "abort"
  vtype: "select"
  name: "mysql-apt-config"

config_server:
  mysql-community-server/data-dir:
    value: ""
    vtype: "note"
    name: "mysql-community-server"

  mysql-community-server/root-pass:
    value: "{{ mysql_root_password }}"
    vtype: "password"
    name: "mysql-community-server"

  mysql-community-server/re-root-pass:
    value: "{{ mysql_root_password }}"
    vtype: "password"
    name: "mysql-community-server"

  mysql-community-server/remove-data-dir:
    value: "false"
    vtype: "boolean"
    name: "mysql-community-server"

  mysql-community-server/root-pass-mismatch:
    value: ""
    vtype: "error"
    name: "mysql-community-server"

  mysql-server/default-auth-override:
    value: "Use Strong Password Encryption (RECOMMENDED)"
    vtype: "select"
    name: "mysql-community-server"

  mysql-server/lowercase-table-names:
```

```

value: ""
vtype: "select"
name: "mysql-community-server"

```

- **/roles/mysql/defaults/main.yml:**

Incluye los valores de variables por defecto, como la contraseña de root en caso de que no se asigne en el fichero vault y una lista de usuarios por defecto que contiene al usuario invitado si la lista de usuarios está vacía.

```

#SPDX-License-Identifier: MIT-0
---
# defaults file for mysql

# Contraseña del usuario administrador por defecto
mysql_root_password: "abc123."
mysql_users: |
  guest:
    password: "guestpw"
  priv:

```

- **/roles/mysql/tasks/main.yml:**

Contiene las configuraciones propias del debconf, la descarga e instalación del paquete MySQL, entre otras tareas.

En este fichero se deberá hacer uso de bucles y del módulo [community.mysql.mysql_user](#) para crear los usuarios y asignarles privilegios.

```

#SPDX-License-Identifier: MIT-0
---
# tasks file for mysql
- name: Eliminar paquetes MariaDB conflictivos
  ansible.builtin.apt:
    name: "{{ paquetes_conflictivos_mariadb }}"
    state: absent
    purge: yes
  tags:
    - instalacion_mysql

- name: Configurar debconf mysql-apt-config
  ansible.builtin.debconf:
    name: "{{ initial.value.name }}"
    question: "{{ initial.key }}"
    value: "{{ initial.value.value }}"
    vtype: "{{ initial.value.vtype }}"
  loop: "{{ configuracion_inicial | dict2items }}"
  loop_control:
    loop_var: initial
  tags:

```

```
- instalacion_mysql

- name: Descargar paquete inicial
  ansible.builtin.shell:
    cmd: wget -O /tmp/{{ version_paquete }} https://dev.mysql.com/get/{{ version_paquete }}
  tags:
    - instalacion_mysql

- name: Instalar paquete inicial
  ansible.builtin.apt:
    deb: /tmp/{{ version_paquete }}
  tags:
    - instalacion_mysql

- name: Configurar debconf mysql-community-server
  ansible.builtin.debconf:
    name: "{{ server.value.name }}"
    question: "{{ server.key }}"
    value: "{{ server.value.value }}"
    vtype: "{{ server.value.vtype }}"
  loop: "{{ configuracion_servidor | dict2items }}"
  loop_control:
    loop_var: server
  tags:
    - instalacion_mysql

- name: Instalar {{ software }} y python3-pymysql
  ansible.builtin.apt:
    name:
      - "{{ software }}"
      - python3-pymysql
    update_cache: yes
  tags:
    - instalacion_mysql

- name: Asegurar que el servicio {{ servicio }} esté corriendo
  ansible.builtin.systemd:
    name: "{{ servicio }}"
    state: started
    enabled: yes
  tags:
    - instalacion_mysql

- name: Crear usuario con contraseña, todos los privilegios de base de datos y 'WITH GRANT OPTION' en db1 y db2
  community.mysql.mysql_user:
    login_user: "root"
    login_password: "{{ mysql_root_password }}"
    state: present
    name: "{{ current_mysql_user.key }}"
    password: "{{ current_mysql_user.value.password }}"
    priv: "{{ current_mysql_user.value.priv }}"
  loop: "{{ mysql_users | dict2items }}"
```

```
loop_control:  
  loop_var: current_mysql_user  
tags:  
  - usuarios_mysql
```