

Tarea 4 Enunciado: Instalación de mysql mediante playbook

En la [dirección](#) se describe cómo instalar un servidor MySQL en Debian 12 pudiendo encontrar los archivos de configuración [aquí](#) siendo la versión más reciente del paquete la 36. Hay que elaborar un *playbook* para automatizar el proceso empleando algunos de los siguientes módulos de Ansible:

- `ansible.builtin.apt`
- `ansible.builtin.debconf`
- `ansible.builtin.shell`

Es importante saber trabajar con herramientas de **debconf** (`apt install debconf-utils`) como *debconf-set-selections* o *debconf-get-selections*, para que el proceso de instalación de software (`apt`) sea automático. Se entregará el *playbook* resultante de la instalación.

Nota: [Aquí](#) puedes encontrar como configurar MySQL paso a paso.

Instalación correcta de MySQL

1. Desinstala las versiones previas que se hayan instalado anteriormente si existen:

```
root@serverPrueba:~# apt purge -y mysql-server
```

2. Actualizar los repositorios para asegurarnos de tener los cambios más recientes:

```
root@serverPrueba:~# apt update
```

3. (Ya hecho) Descargar el paquete de configuración oficial de MySQL:

```
root@serverPrueba:~# wget https://dev.mysql.com/get/mysql-apt-config_0.8.36-1_all.deb
```

4. Instalar el paquete de configuración recién descargado:

```
root@serverPrueba:~# file mysql-apt-config_0.8.36-1_all.deb
mysql-apt-config_0.8.36-1_all.deb: Debian binary package (format 2.0), with
control.tar.xz, data compression xz
root@serverPrueba:~# apt install ./mysql-apt-config*_all.deb
```

Durante la instalación:

- Selecciona **MySQL Server and Cluster**
- Elige la versión **mysql-8.0**
- Resalta la opción **OK**, pulsa **Tab**, y luego **Enter**

5. Instalamos debconf-utils y vemos las preguntas de configuración que tenemos de mysql-ap-config

```
root@serverPrueba:~# apt install -y debconf-utils
...
root@serverPrueba:~# debconf-get-selections | grep mysql
mysql-apt-config      mysql-apt-config/connectors-component  string  mysql-
tools
mysql-apt-config      mysql-apt-config/repo-codename  select  bookworm
mysql-apt-config      mysql-apt-config/repo-distro    select  debian
mysql-apt-config      mysql-apt-config/repo-url       string
http://repo.mysql.com/apt
mysql-apt-config      mysql-apt-config/select-connectors  select  Enabled
# Choices: MySQL Server & Cluster (Currently selected: mysql-8.0), MySQL
Connectors (Currently selected: Enabled), Ok
mysql-apt-config      mysql-apt-config/select-product  select  Ok
# Choices: mysql-8.0, mysql-innovation, mysql-8.4-lts, mysql-cluster-8.0, mysql-
cluster-innovation, mysql-cluster-8.4-lts, None
mysql-apt-config      mysql-apt-config/select-server  select  mysql-8.0
mysql-apt-config      mysql-apt-config/unsupported-platform  select  abort
```

6. Instalar el servidor MySQL:

```
root@serverPrueba:~# apt update
...
root@serverPrueba:~# apt install -y mysql-server
...
root@serverPrueba:~# systemctl status mysql
• mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-09-30 16:53:49 CEST; 9s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
  Main PID: 2015 (mysqld)
    Status: "Server is operational"
     Tasks: 38 (limit: 1108)
  Memory: 368.9M
     CPU: 1.069s
   CGroup: /system.slice/mysql.service
           └─2015 /usr/sbin/mysqld
```

Nota: Si queremos purgar mysql hacemos lo siguiente.

```
root@debian64:~# apt purge mysql-server mysql-client mysql-common mysql-server-
core-* mysql-client-core-*
```

7. Vemos ahora las preguntas de configuración del debconf de nuevo y podemos apreciar que tenemos por una parte la configuración de los paquetes y a continuación las preguntas propias de la instalación del servidor mysql.

```
root@serverPrueba:~# debconf-get-selections | grep ^mysql
mysql-apt-config      mysql-apt-config/connectors-component  string  mysql-
tools
mysql-apt-config      mysql-apt-config/repo-codename  select  bookworm
mysql-apt-config      mysql-apt-config/repo-distro    select  debian
mysql-apt-config      mysql-apt-config/repo-url       string
http://repo.mysql.com/apt
mysql-apt-config      mysql-apt-config/select-connectors  select  Enabled
mysql-apt-config      mysql-apt-config/select-product  select  Ok
mysql-apt-config      mysql-apt-config/select-server    select  mysql-8.0
mysql-apt-config      mysql-apt-config/unsupported-platform  select  abort
mysql-community-server mysql-community-server/data-dir  note
mysql-community-server mysql-community-server/re-root-pass  password
mysql-community-server mysql-community-server/remove-data-dir  boolean  false
mysql-community-server mysql-community-server/root-pass      password
mysql-community-server mysql-community-server/root-pass-mismatch  error
mysql-community-server mysql-server/default-auth-override  select  Use Strong
Password Encryption (RECOMMENDED)
mysql-community-server mysql-server/lowercase-table-names  select

root@serverPrueba:~# debconf-get-selections | grep ^mysql >
/tmp/respuestas_mysql.txt
```

Llegados a este punto necesitamos las repuestas a las preguntas de configuración en nuestro *ControllerNode*.

```
ansible@debian64:~$ scp ansible@192.168.10.5:/tmp/respuestas_mysql.txt /tmp
respuestas_mysql.txt
100% 1053 376.5KB/s 00:00
ansible@debian64:~$ ls -l /tmp/respuestas_*
-rw-r--r-- 1 ansible ansible 1053 nov 13 09:30 /tmp/respuestas_mysql.txt
```

Playbook de instalación de MySQL

Tenemos que crear el playbook para instalar en *srv1* el servicio MySQL. Aunque el playbook como tal es mejorable, hasta este punto la mejor forma que tenemos de hacerlo es la siguiente. Este playbook será mejorado una vez que añadamos bucles.

```
---
- name: Instalación de MySQL Oracle
  hosts: srv1
  vars:
    mysql_root_password: "toor"
```

tasks:

```
# - name: Eliminar paquetes MariaDB conflictivos
#   ansible.builtin.apt:
#     name:
#       - mariadb-server
#       - mariadb-client
#       - mariadb-client-core
#       - mariadb-server-core
#     state: absent
#     purge: yes

# Instalar paquete debconf-utils
- name: Instalar paquete debconf-utils
  ansible.builtin.apt:
    name: debconf-utils

# Configurar respuestas de debconf para mysql-apt-config
- name: Configurar connectors-component
  ansible.builtin.debconf:
    name: mysql-apt-config
    question: mysql-apt-config/repo-codename
    vtype: string
    value: mysql-tools

- name: Configurar repo-codename
  ansible.builtin.debconf:
    name: mysql-apt-config
    question: mysql-apt-config/repo-codename
    vtype: select
    value: bookworm

- name: Configurar repo-distro
  ansible.builtin.debconf:
    name: mysql-apt-config
    question: mysql-apt-config/repo-distro
    vtype: select
    value: debian

- name: Configurar repo-url
  ansible.builtin.debconf:
    name: mysql-apt-config
    question: mysql-apt-config/repo-url
    vtype: string
    value: "http://repo.mysql.com/apt"

- name: Configurar select-connectors
  ansible.builtin.debconf:
    name: mysql-apt-config
    question: mysql-apt-config/select-connectors
    vtype: select
    value: Enabled

- name: Configurar select-product
```

```
    ansible.builtin.debconf:
      name: mysql-apt-config
      question: mysql-apt-config/select-product
      vtype: select
      value: Ok

- name: Configurar select-server
  ansible.builtin.debconf:
    name: mysql-apt-config
    question: mysql-apt-config/select-server
    vtype: select
    value: mysql-8.0

- name: Configurar unsupported-platform
  ansible.builtin.debconf:
    name: mysql-apt-config
    question: mysql-apt-config/unsupported-platform
    vtype: select
    value: abort

# Descargar paquete inicial
- name: Descargar paquete inicial
  ansible.builtin.shell:
    cmd: wget -O /tmp/mysql-apt-config_0.8.36-1_all.deb
https://dev.mysql.com/get/mysql-apt-config_0.8.36-1_all.deb

# Instalar el paquete de configuración
- name: Instalar paquete inicial
  ansible.builtin.apt:
    deb: /tmp/mysql-apt-config_0.8.36-1_all.deb

# Configurar respuestas de debconf para mysql-community-server
- name: Configurar data-dir
  ansible.builtin.debconf:
    name: mysql-community-server
    question: mysql-community-server/data-dir
    vtype: note
    value: ""

- name: Configurar root-pass
  ansible.builtin.debconf:
    name: mysql-community-server
    question: mysql-community-server/root-pass
    vtype: password
    value: "{{ mysql_root_password }}"

- name: Configurar remove-data-dir
  ansible.builtin.debconf:
    name: mysql-community-server
    question: mysql-community-server/remove-data-dir
    vtype: boolean
    value: false

- name: Configurar re-root-pass
```

```
ansible.builtin.debconf:
  name: mysql-community-server
  question: mysql-community-server/re-root-pass
  vtype: password
  value: "{{ mysql_root_password }}"

- name: Configurar root-pass-mismatch
  ansible.builtin.debconf:
    name: mysql-community-server
    question: mysql-community-server/root-pass-mismatch
    vtype: error
    value: ""

- name: Configurar default-auth-override
  ansible.builtin.debconf:
    name: mysql-community-server
    question: mysql-server/default-auth-override
    vtype: select
    value: "Use Strong Password Encryption (RECOMMENDED)"

- name: Configurar lowercase-table-names
  ansible.builtin.debconf:
    name: mysql-community-server
    question: mysql-server/lowercase-table-names
    vtype: select
    value: ""

# Instalar MySQL Server
- name: Instalar mysql-server
  ansible.builtin.apt:
    name: mysql-server
    state: present
    update_cache: yes

# Asegurar que el servicio esté activo
- name: Asegurar que el servicio MySQL esté corriendo
  ansible.builtin.service:
    name: mysql
    state: started
    enabled: yes
```

Podemos ver como nuestra base de datos MySQL funciona si creamos una pequeña base de datos de ejemplo y hacemos consultas sobre ella.

```
mysql -u root -p < usuarios_coches.sql
```

```
CREATE DATABASE IF NOT EXISTS usuarios_coches;
USE usuarios_coches;
```

```

CREATE TABLE IF NOT EXISTS usuarios (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL
);

CREATE TABLE IF NOT EXISTS coches (
  id INT AUTO_INCREMENT PRIMARY KEY,
  usuario_id INT NOT NULL,
  marca VARCHAR(50) NOT NULL,
  modelo VARCHAR(50) NOT NULL,
  año INT,
  FOREIGN KEY (usuario_id) REFERENCES usuarios(id) ON DELETE CASCADE
);

INSERT INTO usuarios (nombre, email) VALUES
('Juan Pérez', 'juan@example.com'),
('Ana Gómez', 'ana@example.com');

INSERT INTO coches (usuario_id, marca, modelo, año) VALUES
(1, 'Toyota', 'Corolla', 2018),
(1, 'Honda', 'Civic', 2019),
(2, 'Ford', 'Focus', 2020);

-- Consulta para mostrar usuarios con sus coches
SELECT u.nombre, u.email, c.marca, c.modelo, c.año
FROM usuarios u
LEFT JOIN coches c ON u.id = c.usuario_id;

```

```

root@server1:~# mysql -u root -p
...
mysql> select * from usuarios_coches.coches;
+----+-----+-----+-----+-----+
| id | usuario_id | marca  | modelo | año  |
+----+-----+-----+-----+-----+
| 1  |          1 | Toyota | Corolla | 2018 |
| 2  |          1 | Honda  | Civic   | 2019 |
| 3  |          2 | Ford   | Focus   | 2020 |
+----+-----+-----+-----+-----+
3 rows in set (0,01 sec)

```