**Introducción**

MariaDB es un sistema de gestión de bases de datos relacional, el cual se deriva de MySQL, que es de código abierto y muy usado en entornos empresariales y de desarrollo. Por su parte, InnoDB es el motor de almacenamiento por defecto en MariaDB, conocido por su estabilidad, soporte a transacciones y características avanzadas como integridad referencial y recuperación ante fallos.

**Revisión Bibliográfica**

MariaDB surgió como una bifurcación de MySQL cuando Oracle adquirió MySQL, con el objetivo de mantener una base de datos libre y abierta. La comunidad y desarrolladores originales continúan su desarrollo. Muchos estudios y documentación técnica destacan su rendimiento y confiabilidad, especialmente cuando se utiliza el motor InnoDB para el almacenamiento de datos, el cual es el más utilizado.

InnoDB permite operaciones transaccionales, bloqueo a nivel de fila especifica que esta siendo modifica sin necesidad de bloquear toda la tabla y recuperación automática, siendo ideal para aplicaciones que requieren alta disponibilidad y consistencia.

**Marco Teórico**

MariaDB es una base de datos relacional libre y de código abierto derivada de MySQL, que ofrece gran compatibilidad y mejoras en rendimiento. Su motor de almacenamiento por defecto, InnoDB, es confiable, soporta transacciones y asegura la integridad de los datos.

Si hablamos de sistemas distribuidos y contenedores como Docker Swarm, MariaDB junto con InnoDB brindan una solución robusta para gestionar datos de forma eficiente y segura. La replicación activa-pasiva permite mantener un nodo principal que acepta escrituras y un nodo secundario que replica los datos en tiempo real, lo que asegura alta disponibilidad y continuidad en caso de fallos.

El uso de MariaDB con InnoDB en un clúster Docker Swarm permite aprovechar la escalabilidad y gestión de contenedores junto con una base de datos sólida y segura.

La replicación activa-pasiva garantiza que, ante una falla del nodo activo, el nodo pasivo pueda tomar control sin pérdida de datos ni interrupciones significativas en el servicio. Esto es esencial para aplicaciones críticas en la empresa que requieren alta disponibilidad.

**Guía: Configurar servidor ACTIVO MariaDB + InnoDB en Ubuntu**

**Supuestos / prerequisitos**

* Acceso sudo al servidor (root o usuario con sudo).
* Ubuntu 18.04 / 20.04 / 22.04 / 24.04
* IP estática del **activo** (ej. ACTIVE\_IP=192.0.2.10) y del **pasivo** (PASSIVE\_IP=192.0.2.11).
* El puerto de MariaDB será el estándar 3306 (abriremos sólo desde la IP del pasivo).
* Reemplaza: REPL\_USER, REPL\_PASS, ACTIVE\_IP, PASSIVE\_IP.

***Actualizar sistema e instalar MariaDB***

sudo apt update && sudo apt upgrade -y

sudo apt install mariadb-server mariadb-client ufw -y

sudo systemctl enable --now mariadb

***Verifica versión y servicio:***

mariadb --version

sudo systemctl status mariadb --no-pager

**2) Ajustes básicos de seguridad (interactivo)**

Ejecuta:

sudo mysql\_secure\_installation

Respuestas recomendadas:

* Set root password: **Yes** (o si prefieres usar auth\_socket, lo puedes omitir; la guía usa sudo mariadb para comandos root)
* Remove anonymous users: **Yes**
* Disallow root remote login: **Yes**
* Remove test database: **Yes**
* Reload privilege tables: **Yes**

Nota: en Ubuntu, el root de MariaDB suele usar unix\_socket, por eso sudo mariadb -e "SQL" funciona sin contraseña.

**3) Configuración de MariaDB para replicación (editar archivo)**

Edita /etc/mysql/mariadb.conf.d/50-server.cnf (o el archivo de configuración principal si difiere). Añade/asegúrate de estas líneas bajo [mysqld] Editar solo la parte que diga [mysqld]:

[mysqld]

server-id = 1 # ID único del maestro

bind-address = 0.0.0.0 # permitir conexiones desde la red (cuidado: proteger con firewall)

log\_bin = /var/log/mysql/mariadb-bin

log\_bin\_index = /var/log/mysql/mariadb-bin.index

binlog\_format = ROW # recomendado para replicación segura con InnoDB

innodb\_file\_per\_table = 1

# Opcionales (ajusta según RAM)

# innodb\_buffer\_pool\_size = 1G

# max\_connections = 200

expire\_logs\_days = 7 # borrar binlogs viejos automáticamente

***Crear carpeta de logs y permisos (si no existe):***

sudo mkdir -p /var/log/mysql

sudo chown mysql:mysql /var/log/mysql

sudo chmod 750 /var/log/mysql

***4) Crear el usuario de replicación (en el ACTIVO)***

Recomiendo restringir el acceso del usuario de replicación a la IP del pasivo.

PASS="Ulatina"

PASSIVE\_IP="10.0.0.11" # reemplazar por Ip de equipo

sudo mariadb -e "CREATE USER IF NOT EXISTS 'repluser'@'${PASSIVE\_IP}' IDENTIFIED BY '${PASS}';"

sudo mariadb -e "GRANT REPLICATION SLAVE ON \*.\* TO 'repluser'@'${PASSIVE\_IP}';"

sudo mariadb -e "FLUSH PRIVILEGES;"

Si vas a permitir la replicación desde cualquier host (no recomendado), usa 'repluser'@'%' en vez de la IP.

***5) Generar snapshot inicial (recomendado para InnoDB: mysqldump)***

Opción A — **Recomendado** si la mayor parte de tus tablas son InnoDB (no requiere lock global si usamos --single-transaction):

sudo mysqldump --all-databases --single-transaction --master-data=2 --events --routines --triggers | sudo tee /root/all\_databases.sql > /dev/null

 --master-data=2 inserta la posición de binlog dentro del dump (comentada). Útil para importar en el pasivo y sincronizar coordenadas automáticamente.

 --single-transaction evita bloquear si solo InnoDB.

***6) Cómo obtener el estado actual del maestro (coordenadas)***

Para dar al servidor pasivo las coordenadas exactas (si no usaste --master-data):

sudo mariadb -e "SHOW MASTER STATUS\G"

Salida relevante:

* File: mariadb-bin.00000X
* Position: 12345

Proporciona estos valores al pasivo para configurarlo.

***7) Firewall: permitir solo al servidor pasivo conectar al puerto 3306***

Usando ufw (ejemplo):

sudo ufw allow OpenSSH

sudo ufw allow from <PASSIVE\_IP> to any port 3306 proto tcp

sudo ufw enable

sudo ufw status verbose

Reemplaza <PASSIVE\_IP> por la IP real del servidor pasivo. Esto evita exponer el puerto 3306 al resto de la red.

***8) Backups periódicos (ejemplo simple con mysqldump)***

Backup lógico diario (ejemplo con cron). Crea /usr/local/bin/backup\_mariadb.sh:

#!/bin/bash

DIR=/var/backups/mariadb

mkdir -p "$DIR"

DATE=$(date +%F\_%H%M)

mysqldump --all-databases --single-transaction --events --routines --triggers > "$DIR/all\_${DATE}.sql"

find "$DIR" -type f -mtime +7 -delete # conservar 7 días

***Hacer ejecutable y agregar cron:***

**sudo chmod +x /usr/local/bin/backup\_mariadb.sh**

**sudo crontab -l | { cat; echo "0 2 \* \* \* /usr/local/bin/backup\_mariadb.sh >/dev/null 2>&1"; } | sudo crontab –**

**Nota: para grandes bases o RTO rápidos, considera mariabackup (physical backup).**

***9) Monitoreo y mantenimiento***

* **Revisa SHOW SLAVE STATUS\G en el pasivo para verificar replicación.**
* **Revisa /var/log/mysql/error.log por problemas.**
* **Controla tamaño de binlogs y usa expire\_logs\_days para rotarlos.**
* **Prueba failover manualmente (simulación) y documenta procedimientos con tu equipo.**

***10) Script opcional (automatiza instalación y configuración del ACTIVO)***

**Guarda el siguiente script como setup\_mariadb\_master.sh, modifica las variables al principio y ejecútalo (sudo bash setup\_mariadb\_master.sh):**

**#!/usr/bin/env bash**

**set -e**

**PASSIVE\_IP="10.0.0.11" # <- reemplaza**

**REPL\_USER="repluser"**

**REPL\_PASS="TuClaveSegura123!" # <- reemplaza**

**# 1. actualizar e instalar**

**apt update && apt upgrade -y**

**apt install -y mariadb-server mariadb-client ufw**

**# 2. asegurar carpeta logs de binlog**

**mkdir -p /var/log/mysql**

**chown mysql:mysql /var/log/mysql**

**chmod 750 /var/log/mysql**

**# 3. backup config y escribir ajustes básicos**

**cp /etc/mysql/mariadb.conf.d/50-server.cnf /etc/mysql/mariadb.conf.d/50-server.cnf.bak**

**cat >> /etc/mysql/mariadb.conf.d/50-server.cnf <<'EOF'**

**# --- configuracion para replicacion ---**

**[mysqld]**

**server-id = 1**

**bind-address = 0.0.0.0**

**log\_bin = /var/log/mysql/mariadb-bin**

**log\_bin\_index = /var/log/mysql/mariadb-bin.index**

**binlog\_format = ROW**

**innodb\_file\_per\_table = 1**

**expire\_logs\_days = 7**

**EOF**

**systemctl restart mariadb**

**# 4. crear usuario replicacion (restringido a la IP del pasivo)**

**sudo mariadb -e "CREATE USER IF NOT EXISTS '${REPL\_USER}'@'${PASSIVE\_IP}' IDENTIFIED BY '${REPL\_PASS}';"**

**sudo mariadb -e "GRANT REPLICATION SLAVE ON \*.\* TO '${REPL\_USER}'@'${PASSIVE\_IP}';"**

**sudo mariadb -e "FLUSH PRIVILEGES;"**

**# 5. crear dump consistente (mysqldump con master-data)**

**sudo mysqldump --all-databases --single-transaction --master-data=2 --events --routines --triggers > /root/all\_databases.sql**

**# 6. firewall: permitir solo al pasivo en 3306**

**ufw allow OpenSSH**

**ufw allow from ${PASSIVE\_IP} to any port 3306 proto tcp**

**ufw --force enable**

**echo "Instalacion y configuracion basica completada."**

**echo "Archivo dump: /root/all\_databases.sql"**

**echo "Copia ese archivo al servidor pasivo y configura el slave con las coordenadas dentro del dump (--master-data=2 inserta las coords)."**

**IMPORTANTE: el script no ejecuta mysql\_secure\_installation (recomendado hacerlo manualmente) y asume que sudo mariadb funciona sin contraseña (modo socket).**

**11) Qué brindar al servidor pasivo (resumen para la siguiente etapa)**

**Al administrador del pasivo o para cuando lo configures deberás entregar:**

* **repluser y su contraseña.**
* **Archivo all\_databases.sql (o la copia física del dato si usaste mariabackup).**
* **Si no usaste --master-data, las coordenadas del maestro (resultado de SHOW MASTER STATUS).**

***12) Comandos útiles de verificación***

**En el maestro (activo):**

**sudo mariadb -e "SHOW MASTER STATUS\G"**

**sudo mariadb -e "SHOW VARIABLES LIKE 'server\_id';"**

**sudo tail -n 200 /var/log/mysql/error.log**

***En el pasivo (cuando lo configures):***

***SHOW SLAVE STATUS\G***

***-- Debe mostrar Slave\_IO\_Running: Yes y Slave\_SQL\_Running: Yes***

***Errores comunes y soluciones rápidas***

* **Slave\_IO\_Running: No** → revisar que el MASTER\_HOST y credenciales sean correctas, que el puerto 3306 esté accesible desde la IP del pasivo y que el usuario de replicación tenga permisos.
* **Access denied for user 'repluser'** → verificar host desde el que se permite el acceso en la creación del usuario ('repluser'@'PASIVO\_IP').
* **No aparece la BBDD nueva en el pasivo** → revisar binlog position y que el slave esté arrancado (START SLAVE;).

**3. Configuración del Nodo Pasivo**

En el nodo pasivo, ejecutar el siguiente script para instalar y configurar MariaDB como esclavo:

#!/bin/bash  
sudo apt update && sudo apt upgrade -y  
sudo apt install mariadb-server -y  
sudo systemctl enable mariadb  
sudo systemctl start mariadb  
# Configuración de replicación  
sudo bash -c 'cat >> /etc/mysql/mariadb.conf.d/50-server.cnf <<EOL  
[mysqld]  
server-id=2  
relay\_log=/var/log/mysql/mariadb-relay-bin.log  
binlog\_do\_db=mi\_basedatos  
EOL'  
sudo systemctl restart mariadb  
# Conexión al nodo activo (reemplazar X.X.X.X con IP del nodo activo)  
mysql -u root -p -e "CHANGE MASTER TO MASTER\_HOST='X.X.X.X', MASTER\_USER='repl', MASTER\_PASSWORD='password\_repl', MASTER\_LOG\_FILE='mariadb-bin.000001', MASTER\_LOG\_POS=4; START SLAVE;"