

## Parte I. Configurando Replicación de Bases de Datos (MySQL) en Ubuntu Server 18.04 LTS.

1. Cambiar al usuario con los privilegios de ROOT, cambie el nombre del servidor (coloque un nombre que le permita saber que es el servidor maestro) e instalar los paquetes mysql-server y mysql-client.

```
$ nano /etc/hostname  
$ apt install -y mysql-server mysql-client
```

2. Una vez instalado procederemos a configurar la replicación. Editaremos el archivo de configuración para configurar el servidor de bases de datos maestro.

```
$ nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
#log-queries-not-using-indexes  
#  
# The following can be used as easy to replay backup logs or for replication.  
# note: if you are setting up a replication slave, see README.Debian about  
# other settings you may need to change.  
server-id                = 1  
log_bin                  = /var/log/mysql/mysql-bin.log  
tmpdir                   = /tmp  
binlog_format            = ROW  
expire_logs_days         = 7  
max_binlog_size          = 500M  
sync_binlog              = 1  
slow_query_log  
#binlog_do_db            = include_database_name  
#binlog_ignore_db        = include_database_name  
#  
# * InnoDB  
#
```

Una vez abierto el archivo procederemos a habilitar (descomentar) las líneas de **server-id** y **log\_bin**. Y después completaremos con las otras líneas de parámetros que faltan para dejar nuestro archivo como se muestra en la figura de arriba.

Se activa la línea server-id con el número 1, ya que este nodo será el maestro, por otro lado, este conjunto de números debe de ser único y ningún otro nodo del clúster de servidores de bases de datos puede tener el mismo número.

En log\_bin, es donde estableceremos la ubicación en donde se encontrará toda la información para la replicación. Así que todos los esclavos copiaran este archivo del maestro al conectarse a él, cuando comiencen la replicación.

Terminando con lo anterior procedemos a guardar y aceptar los cambios realizados al archivo de configuración. *Recuerde todo esto se deberá de hacer con privilegios de ROOT.*

3. Reiniciar el servicio MySQL

```
root@ITM-Server01:/home/itm# systemctl restart mysql
```

4. Ahora nuestro siguiente paso es la configuración del usuario de replicación en el servidor Maestro. En la siguiente imagen se muestra cada uno de los pasos a seguir.

```
root@ITM-Server01:/home/itm# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.26-0ubuntu0.18.04.1-log (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create user repuser@10.131.35.167 identified by 'itmrepuser';
Query OK, 0 rows affected (0.05 sec)

mysql> grant replication slave on *.* to repuser@10.131.35.167;
Query OK, 0 rows affected (0.07 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.07 sec)

mysql> show grants for repuser@10.131.35.167;
+-----+
| Grants for repuser@10.131.35.167 |
+-----+
| GRANT REPLICATION SLAVE ON *.* TO 'repuser'@'10.131.35.167' |
+-----+
1 row in set (0.00 sec)

mysql> _
```

En el paso marcado en la flecha con el **número 1** lo que se hace es crear el usuario **repuser** que pertenece al host **10.131.35.167**. En el paso **número 2 y 3**, se le otorgan los privilegios de REPLICACIÓN ESCLAVO. En el paso **número 4** comprobamos que el usuario y los sus privilegios han sido creados.

5. Ahora nos dirigimos al servidor Esclavo y comenzamos con el cambio de nombre y la instalación de MySQL como se hizo en el servidor Maestro. Una vez hecho esto procederemos a cambiar los valores en el archivo de configuración de MySQL.

```
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
server-id             = 2
log_bin               = /var/log/mysql/mysql-bin.log
read_only              = 1
tmpdir                = /tmp
binlog_format          = ROW
sync_binlog            = 1
expire_logs_days      = 7
max_binlog_size       = 500M
slow_query_log         = 1
#binlog_do_db          = include_database_name
#binlog_ignore_db      = include_database_name
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
```

**Recuerde todo esto se deberá de hacer con privilegios de ROOT.**

En la configuración del esclavo podemos observar que el valor de la variable **server-id** es igual a 2, ya que este valor debe de ser único dentro del clúster de servidores y el valor en 1 es el del servidor Maestro.

Aparece una línea o variable nueva que es **read\_only = 1**, lo que hacemos con esto es establecer al esclavo en modo de solo lectura. En este caso solo los usuarios con privilegios de tipo SUPER y el que se dio de alta en el master con los privilegios de esclavo de replicación tiene derecho a modificar los datos. Garantizando así, que no existan aplicaciones que puedan modificar de manera accidental los datos del esclavo en lugar de los del maestro.

El archivo **mysql-bin.log**, es necesario este activo en el esclavo para que interactúe con el servidor Maestro y otra funcionalidad es que permite la recuperación puntual de la última copia de seguridad de ser necesario.

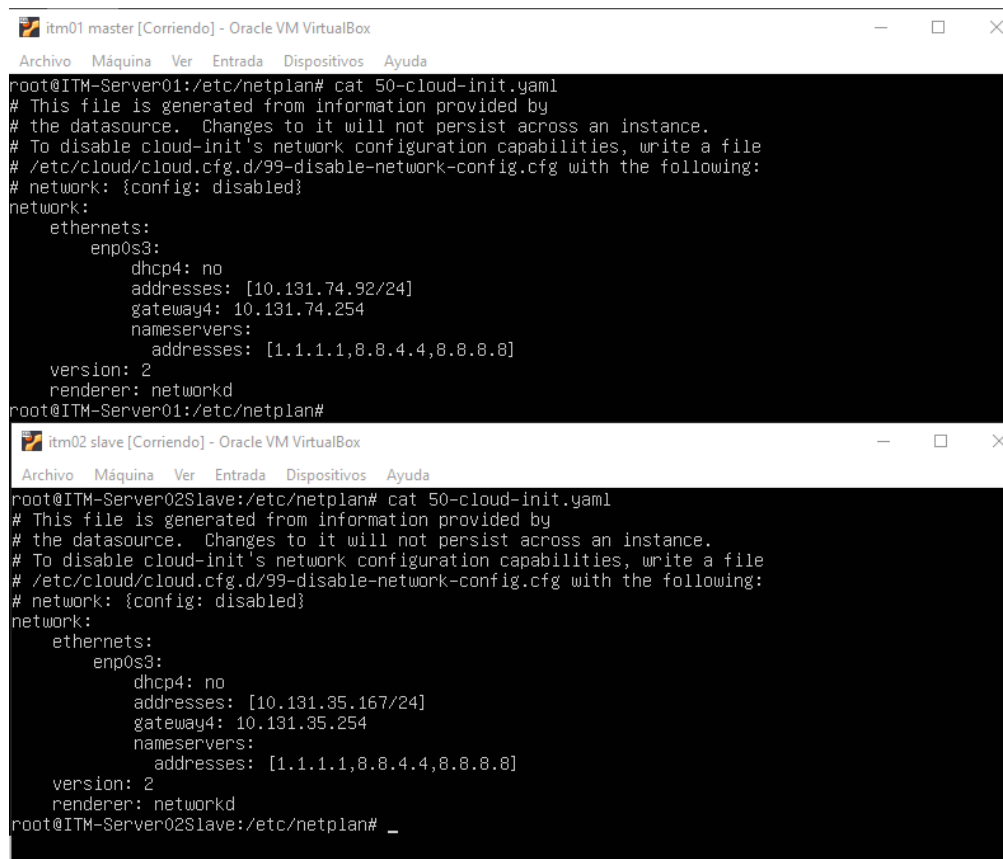
**6. Así como en el paso 3 ahora procederemos a guardar los cambios y a reiniciar el servicio de mysql.**

**7. Nos regresamos al servidor Maestro y comprobamos el estado del mismo. Esto se hace como se muestra en la siguiente imagen. Puede tomar una fotografía de estos valores o anotarlos en algún pedazo de papel, ya que con estos datos configuraremos el servidor Esclavo, los datos importantes son el archivo de log y la posición.**

```
mysql> show master status\G
***** 1. row *****
      File: mysql-bin.000001
      Position: 777
      Binlog_Do_DB:
      Binlog_Ignore_DB:
      Executed_Gtid_Set:
1 row in set (0.00 sec)

mysql> _
```

**8. Ahora si procedemos a configurar nuestras máquinas virtuales para que cada una tenga una dirección IP estática.**



The image shows two terminal windows from Oracle VM VirtualBox. The top window is titled 'itm01 master [Corriendo]' and shows the command 'cat 50-cloud-init.yaml' being executed. The output is a YAML configuration for network settings, including IP address 10.131.74.92/24, gateway 10.131.74.254, and DNS servers 1.1.1.1, 8.8.4.4, and 8.8.8.8. The bottom window is titled 'itm02 slave [Corriendo]' and shows the same 'cat 50-cloud-init.yaml' command being executed. The output is a similar YAML configuration, but with IP address 10.131.35.167/24 and gateway 10.131.35.254.

```
itm01 master [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@ITM-Server01:/etc/netplan# cat 50-cloud-init.yaml
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [10.131.74.92/24]
      gateway4: 10.131.74.254
      nameservers:
        addresses: [1.1.1.1,8.8.4.4,8.8.8.8]
      version: 2
      renderer: networkd
root@ITM-Server01:/etc/netplan#

itm02 slave [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@ITM-Server02Slave:/etc/netplan# cat 50-cloud-init.yaml
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [10.131.35.167/24]
      gateway4: 10.131.35.254
      nameservers:
        addresses: [1.1.1.1,8.8.4.4,8.8.8.8]
      version: 2
      renderer: networkd
root@ITM-Server02Slave:/etc/netplan# _
```

Para llevar a cabo este paso es necesario tener privilegios de ROOT y modificar el archivo de configuración que se encuentra en el directorio `/etc/netplan` y que tiene como terminación `*.yaml`. en la siguiente imagen se muestra la configuración de los 2 servidores, observe que el **servidor Maestro** tiene la dirección **10.131.74.92** y el **servidor Esclavo** tiene **10.131.35.167**. Una vez modificado el archivo de configuración, el siguiente comando es **netplan apply** para aplicar los cambios a las interfaces de red.

9. Regresemos al servidor Esclavo y configurémoslo con los datos obtenidos del servidor Maestro en el punto anterior. Esto se hace dentro de MySQL. Vea la siguiente imagen.

```
mysql> change master to master_host='10.131.74.92',
-> master_user='repuser',
-> master_password='itmrepuser',
-> master_log_file='mysql-bin.000001',
-> master_log_pos=777;
```

10. Después de haber ingresado la configuración del punto anterior, comencemos la replicación en el servidor Esclavo.

```
mysql> start slave;
Query OK, 0 rows affected (0.06 sec)
```

11. Y por último checamos el estatus del servidor esclavo

```
mysql> show slave status\G
```

este comando nos desplegará una lista un poco extensa pero lo que nos importa para saber si está replicando son las variables **Slave\_IO\_Running** y **Slave\_SQL\_Running** con valor de **Yes**.

Hasta este momento sabemos que ya están configurados como Maestro-Esclavo para replicación de las Bases de Datos, si usted desea agregar más esclavos puede hacerlo, siempre y cuando siga los pasos de configuración necesarios.

12. Puede comprobar la replicación creando en el servidor maestro una base de datos llamada **testingdb**, y comprobar desde el servidor Esclavo que se creó también esa base datos. Ahora cree desde el Esclavo una tabla en la base de datos que acabamos de crear con el nombre **testuser**, de la misma manera compruebe en el servidor Maestro que la tabla fue creada.

```
Servidor Maestro:
mysql> CREATE DATABASE testingdb;
Servidor Esclavo:
mysql> SHOW DATABASES;
-----
Servidor Esclavo:
mysql> CREATE TABLE testingdb.testuser ( id INT, nombre VARCHAR(20));
Servidor Maestro:
mysql> USE testingdb;
mysql> SHOW TABLES;
```

## Parte II. Auditoria de Bases de Datos.

Generar una tabla de auditoria, donde guarde en cada fila los datos relevantes a las modificaciones realizadas sobre cada campo al ejecutar un comando SQL INSERT, UPDATE o DELETE sobre las tablas de origen. La tabla debe contener los siguientes campos:

**ID, NombreTabla, PK, NombreColumna, ValorAnterior, ValorActual, FechaHora, UsuarioOS/UsuarioMotor, NombreEquipo,MacAddress, IP.**

Las distintas operaciones pueden generar distintas cantidades de filas en la tabla Auditoría:

- a) Si se realiza un UPDATE sobre un solo valor de una tabla de origen, se generará en la tabla Auditoría una sola fila
- b) Si se genera un INSERT o un DELETE se generarán tantas filas como columnas existan en una tabla de origen.
- c) La finalidad de dicha tabla es poseer un historial detallado sobre los valores anteriores y posteriores a las modificaciones realizadas, especificando de forma detallada la tabla (NombreTabla), las filas (PK) y las columnas (NombreColumna) a la que pertenece cada campo. También incluye quién (UsuarioOS/UsuarioMotor), cómo y cuándo (FechaHora) se realizó cada modificación