Instalación y configuración de MariaDB

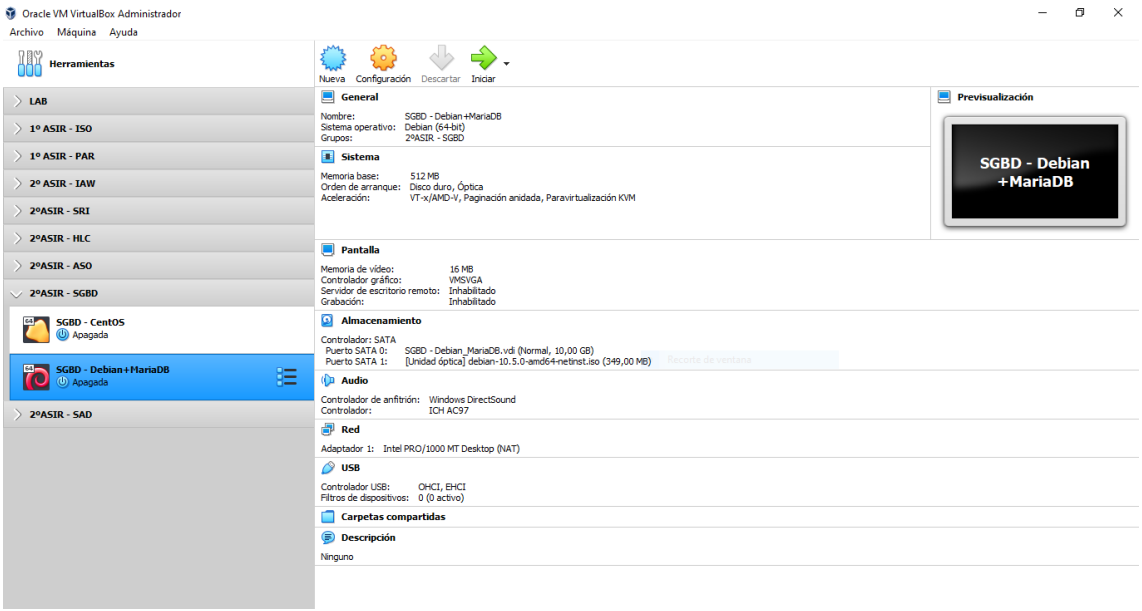
Práctica 2 de ASGBD

29/09/2020
2ºASIR
Rafael Jiménez Cobos

Contenido

Configuración del hardware de la Máquina Virtual

Esta es la configuración que he realizado en la máquina virtual.

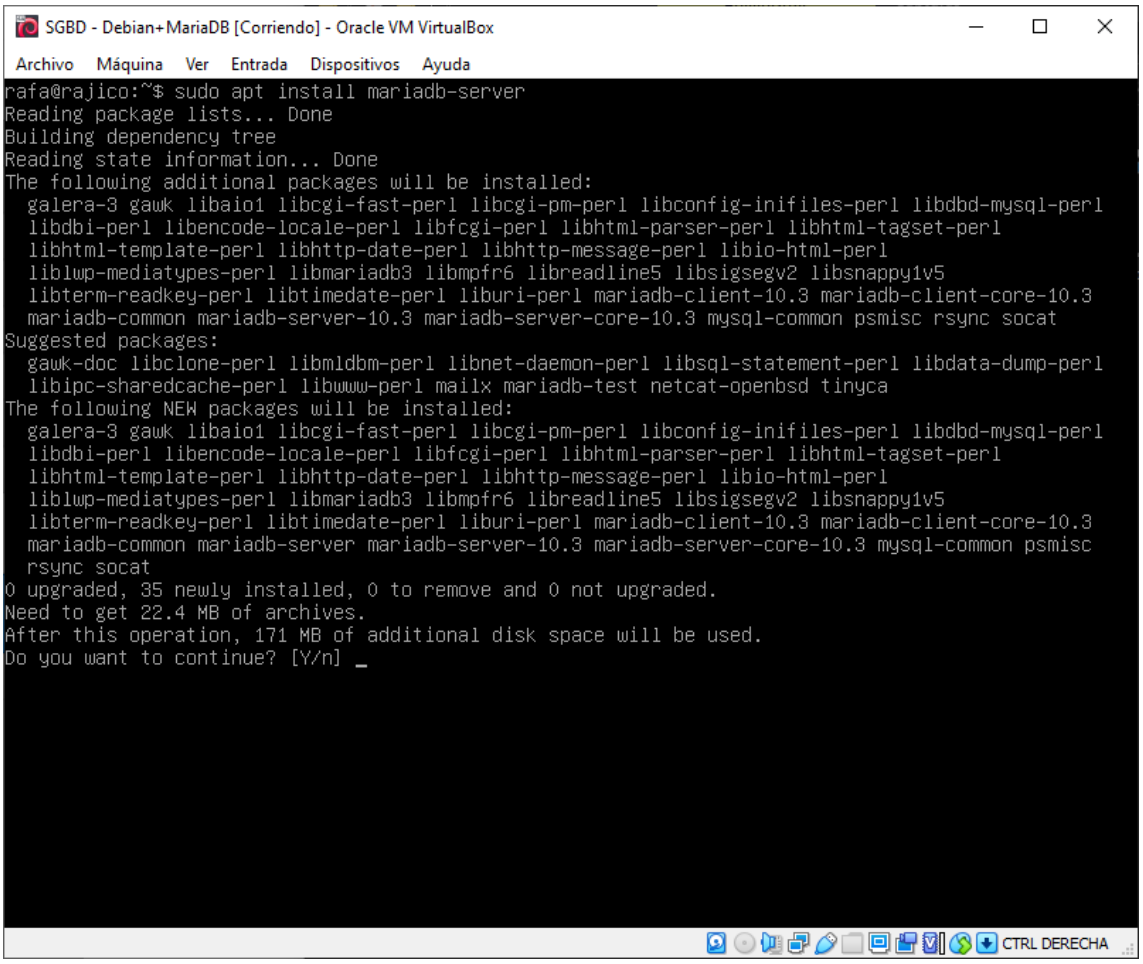


- Sistema operativo: Debian 10, especificado por el enunciado de la práctica, sin entorno gráfico. Arquitectura de 64 bits.
- Memoria RAM: 512 MB. Mínimo recomendado para Debian sin GUI y para MariaDB.
- Disco duro: 10GB. Recomendado para Debian, y es más que suficiente para MariaDB.
- El resto de parámetros son las opciones por defecto.

Instalación de MariaDB

Antes de nada, debemos actualizar el sistema operativo. Esto lo podemos hacer con los comandos apt update y apt upgrade.

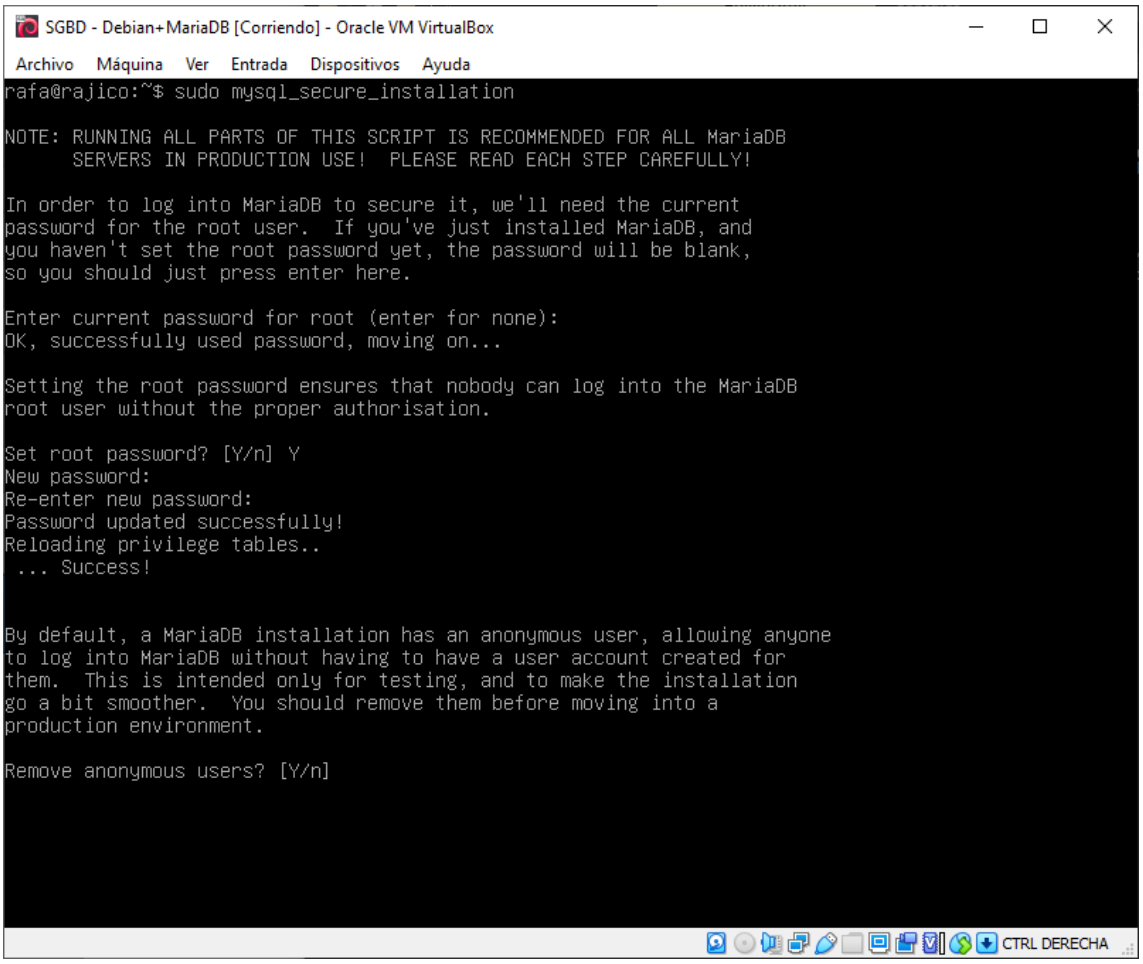
Para instalar MariaDB debemos instalar el paquete mariadb-server.



Como se puede apreciar, necesitamos una serie de paquetes adicionales que no se encuentran en el sistema. Al finalizar la instalación se levantará el servicio con la configuración por defecto.

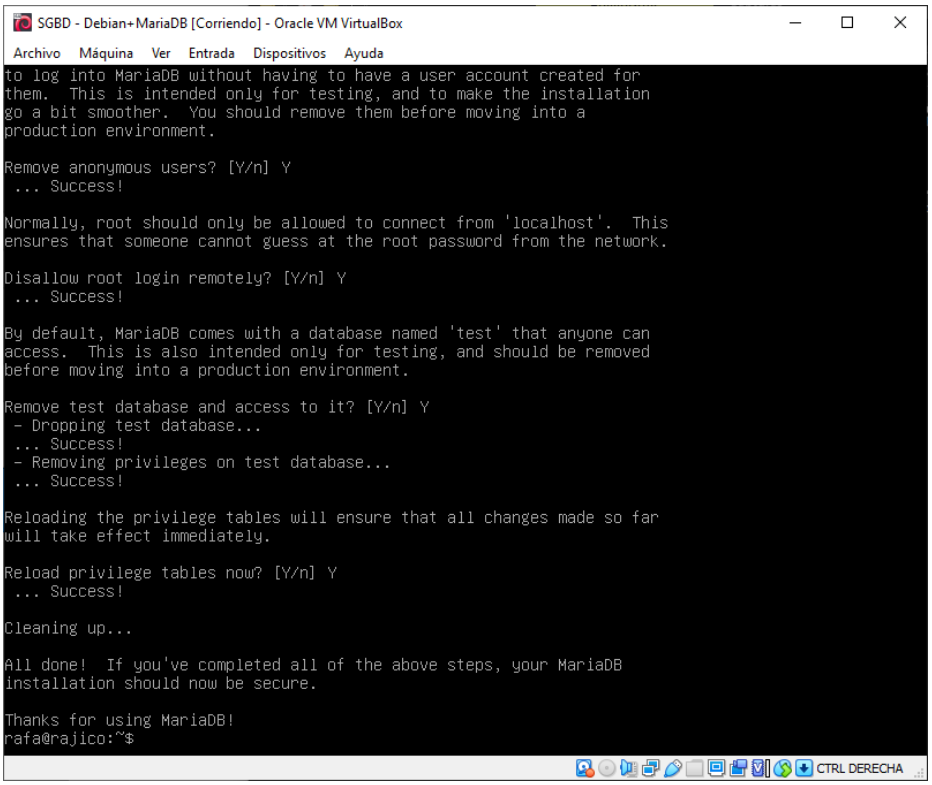
Asegurando la instalación del SGBD

Debemos darle seguridad a la instalación de MariaDB. Esto lo podemos hacer con el comando mysql_secure_installation.



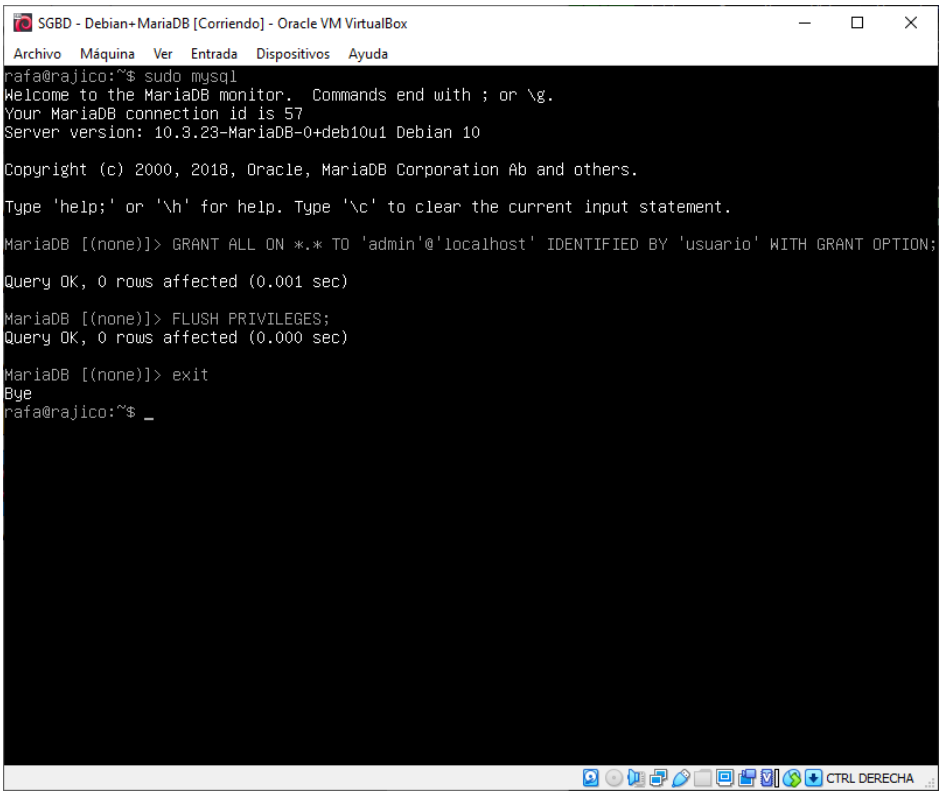
Nos hará una serie de preguntas:

- ¿Queremos asignarle contraseña al usuario root de MariaDB? Aceptamos y asignamos la contraseña.
- ¿Queremos borrar usuarios anónimos? Aceptamos, ya que de lo contrario, cualquiera podría conectarse al servidor en estos momentos.



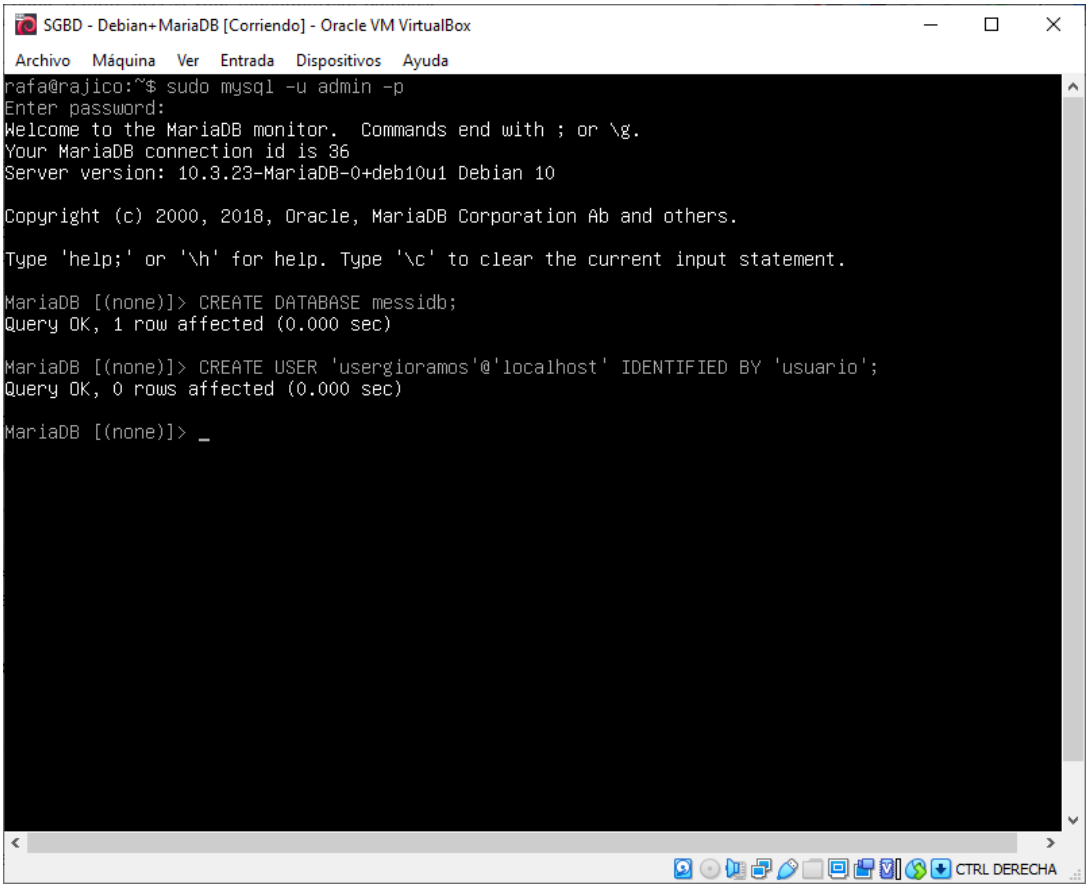
- ¿Queremos que el usuario root no pueda conectarse remotamente? Aceptamos.
- ¿Queremos eliminar la base de datos de prueba y el acceso a ella? Aceptamos.
- ¿Queremos recargar las tablas de privilegios? Aceptamos.

Para finalizar, ejecutamos sudo mysql y creamos un usuario con permisos de root, para no tener que utilizar dicha cuenta.

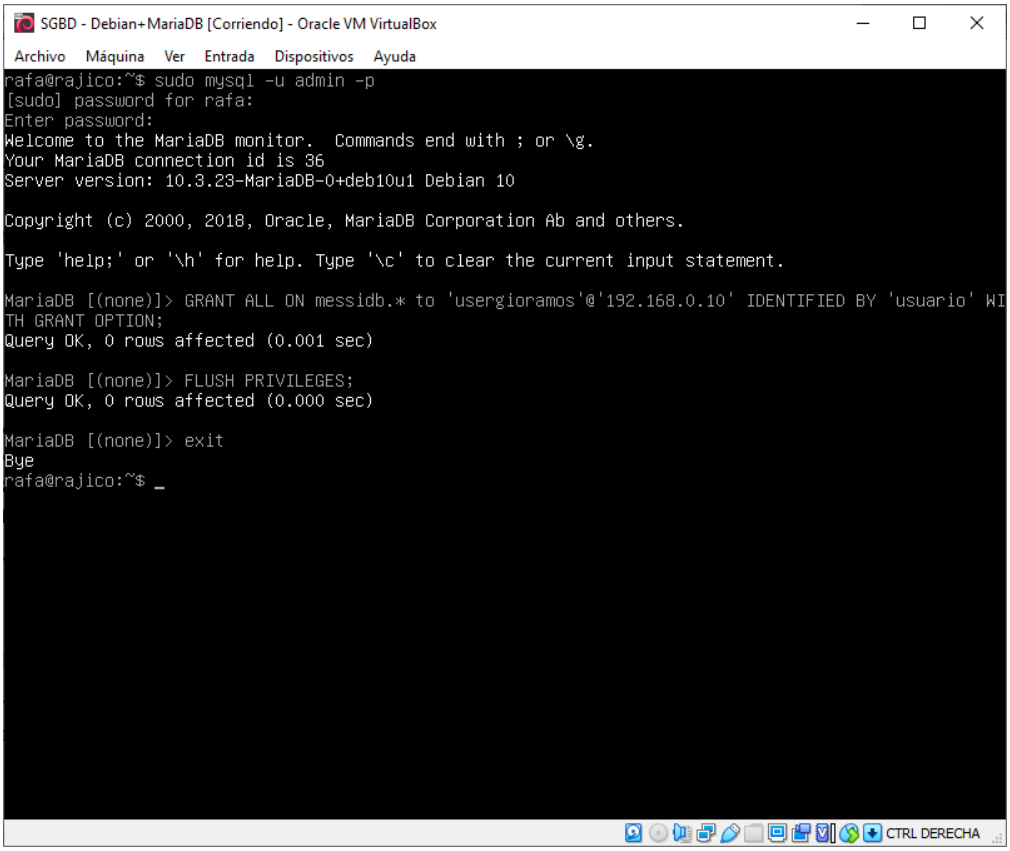


NOTA: Las contraseñas utilizadas en esta práctica no son recomendables al no ser muy seguras. Las he utilizado porque es un entorno controlado.

Una vez terminamos la instalación y aseguración de mysql, vamos a crear el usuario con el que realizaremos la práctica y la base de datos. Para ello, entramos en mysql y creamos la base de datos messidb y el usuario usergioramos.



Y por último, le asignamos permisos SOLO en la base de datos messidb, y habilitamos la conexión mediante el host 192.168.0.10:

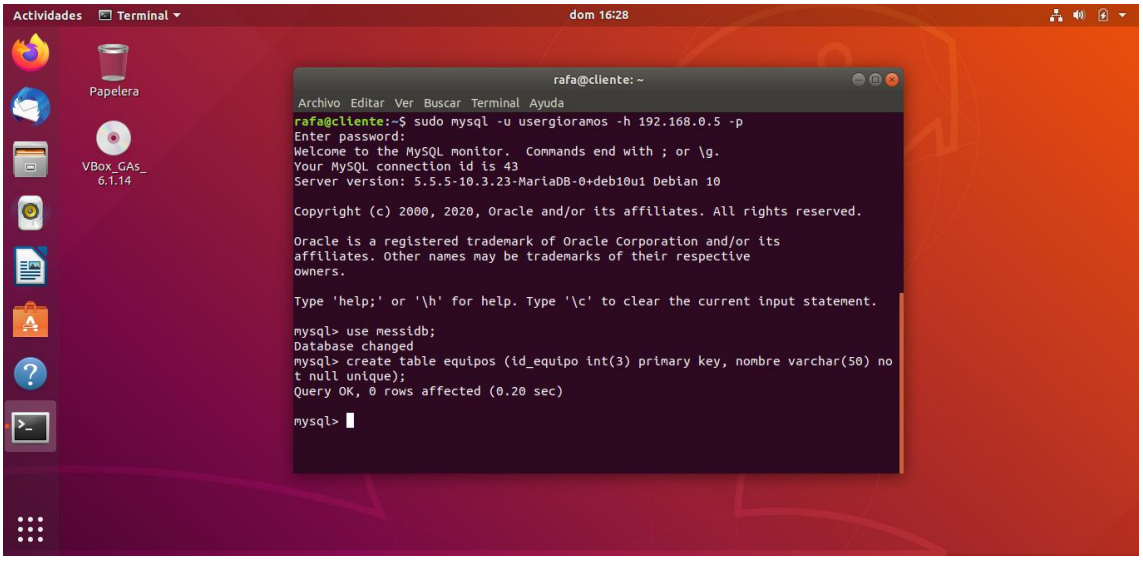


Ya tenemos lo necesario para que el usuario usergioramos se conecte remotamente y administre la base de datos.

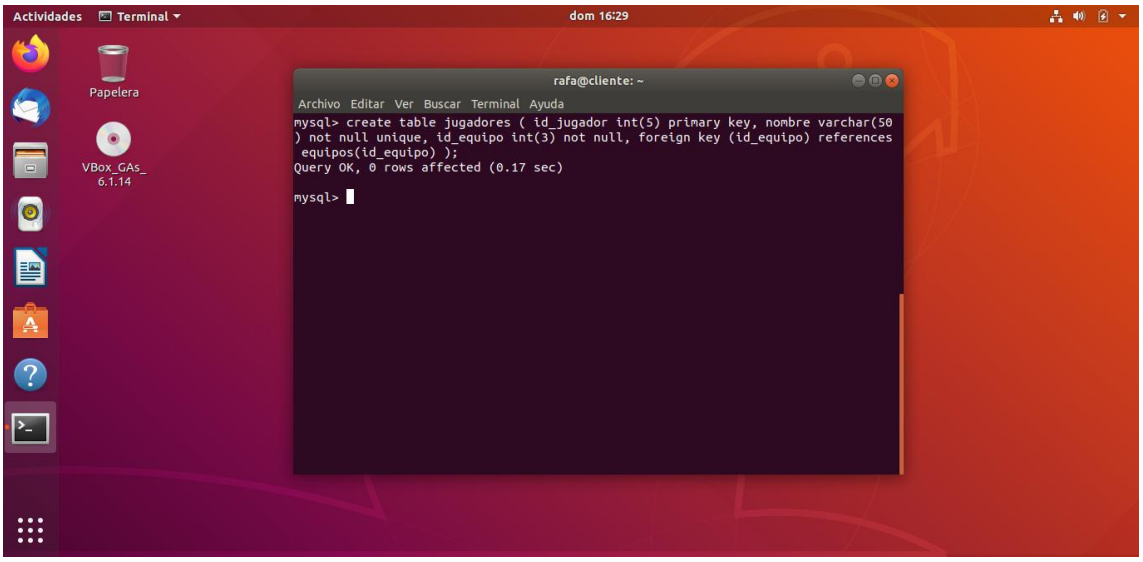
Creando las tablas e insertando datos

Nos conectamos al servidor mysql mediante el cliente.

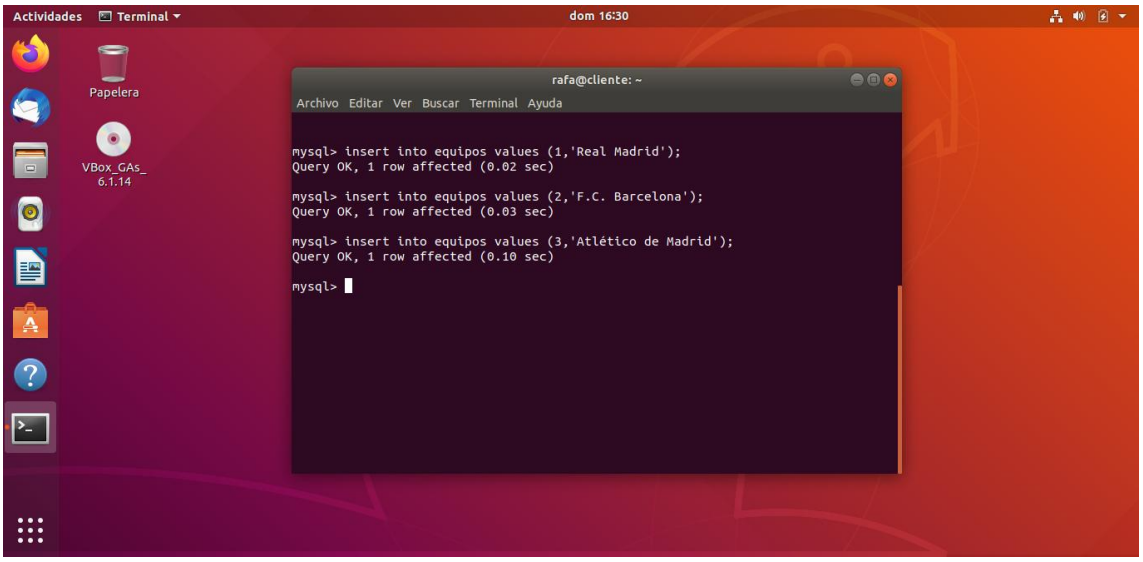
Creamos la tabla equipos:



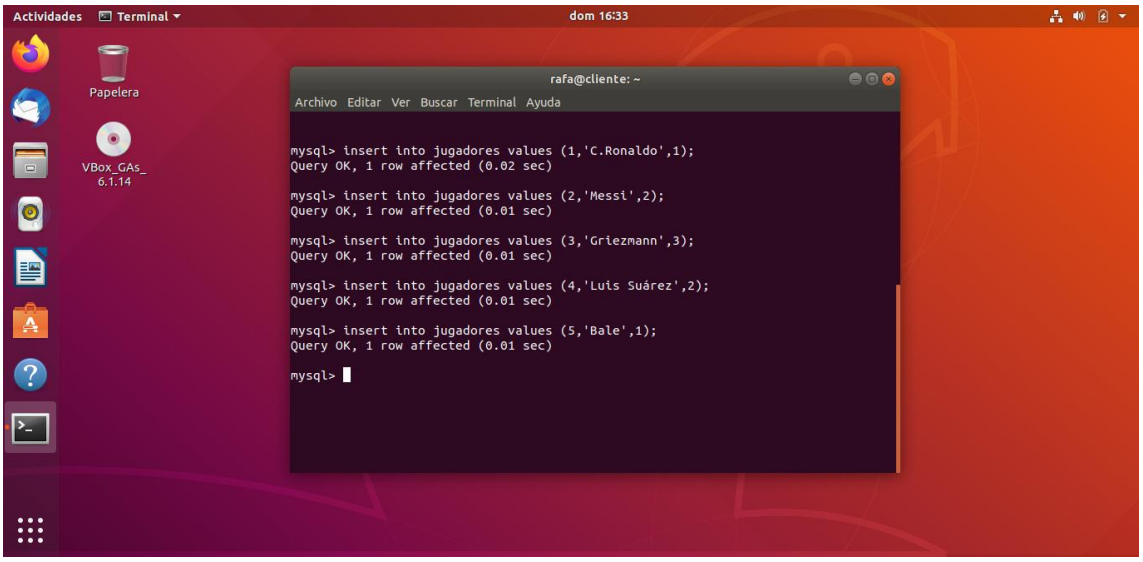
Y la tabla jugadores:



A continuación, insertamos datos en equipos:



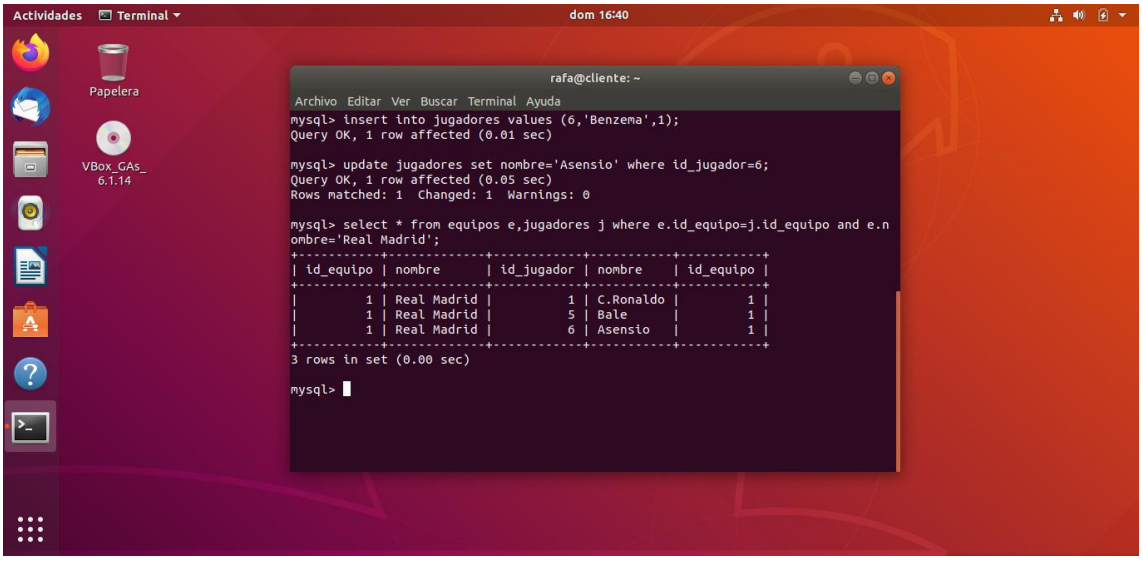
Y por último, en jugadores:



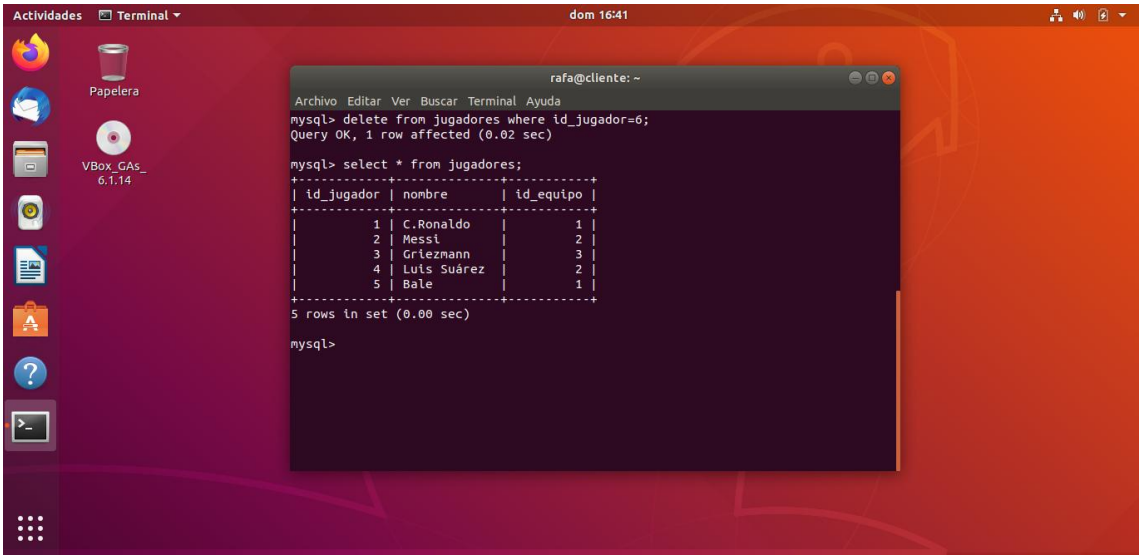
Demonstración de consultas

A continuación voy a realizar una serie de consultas.

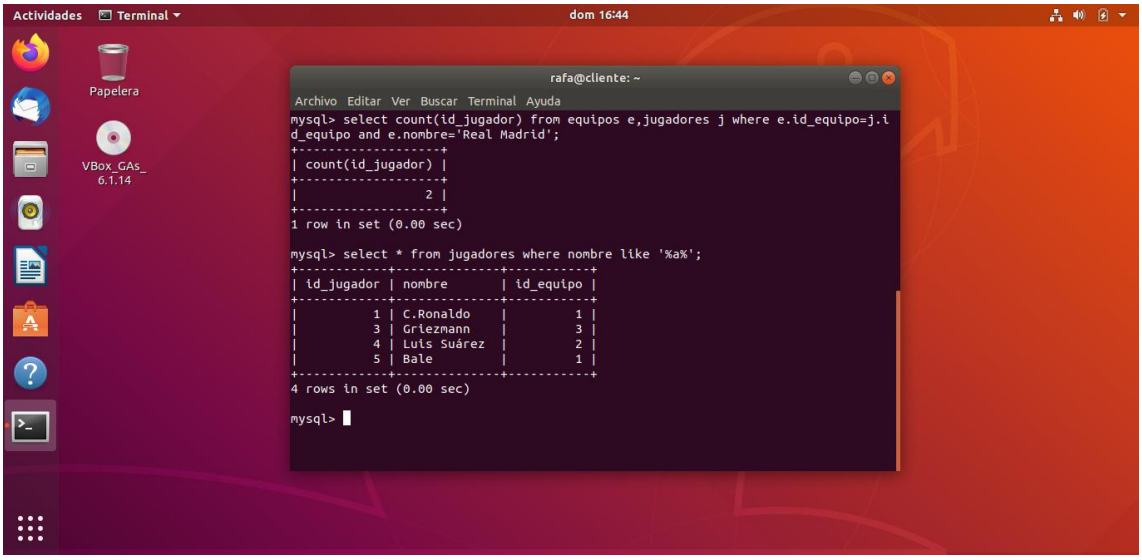
- Inserción del jugador Benzema: insert into jugadores values (6,'Benzema',1);
- Modificar el jugador Benzema para que su nombre sea Asensio: update jugadores set nombre='Asensio' where id_jugadores=6
- Mostrar todos los jugadores que estén en el Real Madrid: select * from equipos e, jugadores j where e.id_equipo=j.id_equipo and e.nombre='Real Madrid';



Borra el jugador Benzema, que es el que tiene el id 6: delete from jugadores where id_jugador=6;

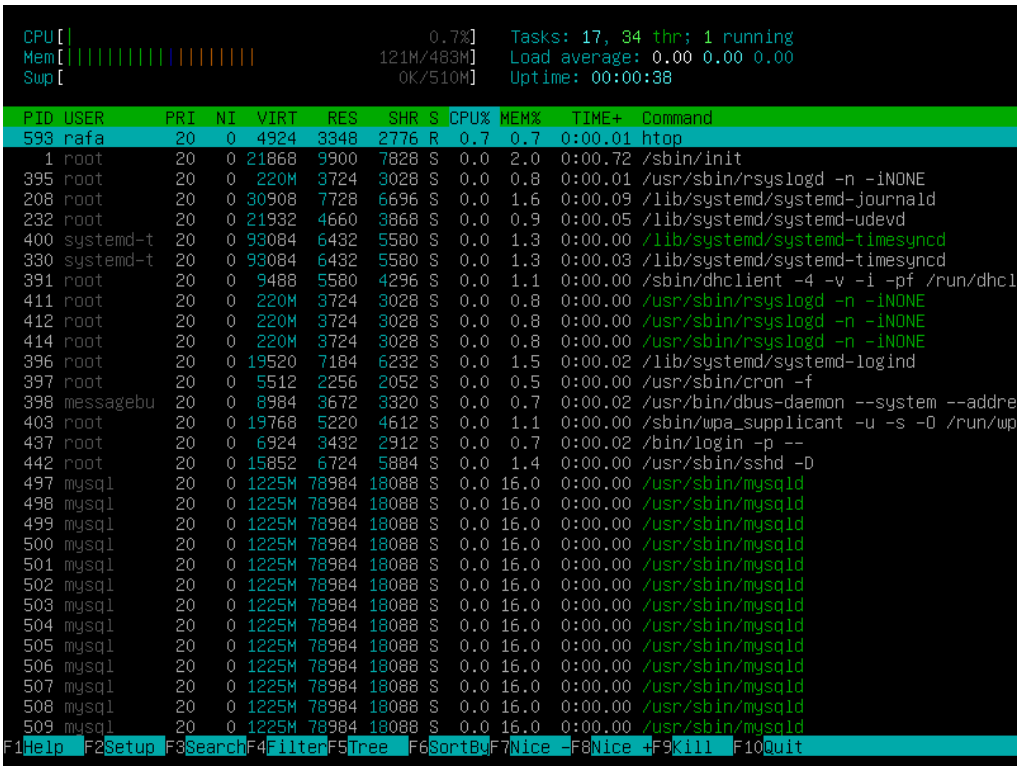


- Cuenta cuantos jugadores hay en el equipo Real Madrid: select count(id_jugador) from equipos e,jugadores j where e.id_equipo=j.id_equipo and e.nombre='Real Madrid';
- Muéstrame los jugadores cuyo nombre contenga la letra a: select * from jugadores where nombre like "%a%";

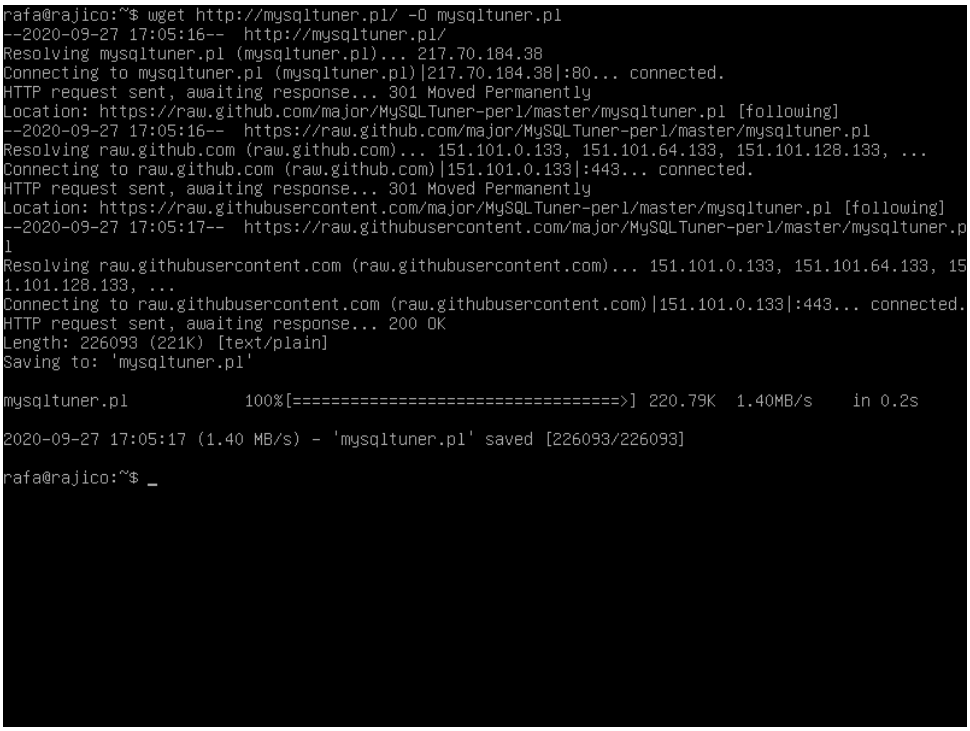


Mysqltuner

Antes de empezar la instalación de mysqltuner, vamos a comprobar el rendimiento de la máquina tras reiniciarla:



A continuación, vamos a descargar mysqltuner, una utilidad para afinar mysql y mejorar su rendimiento.



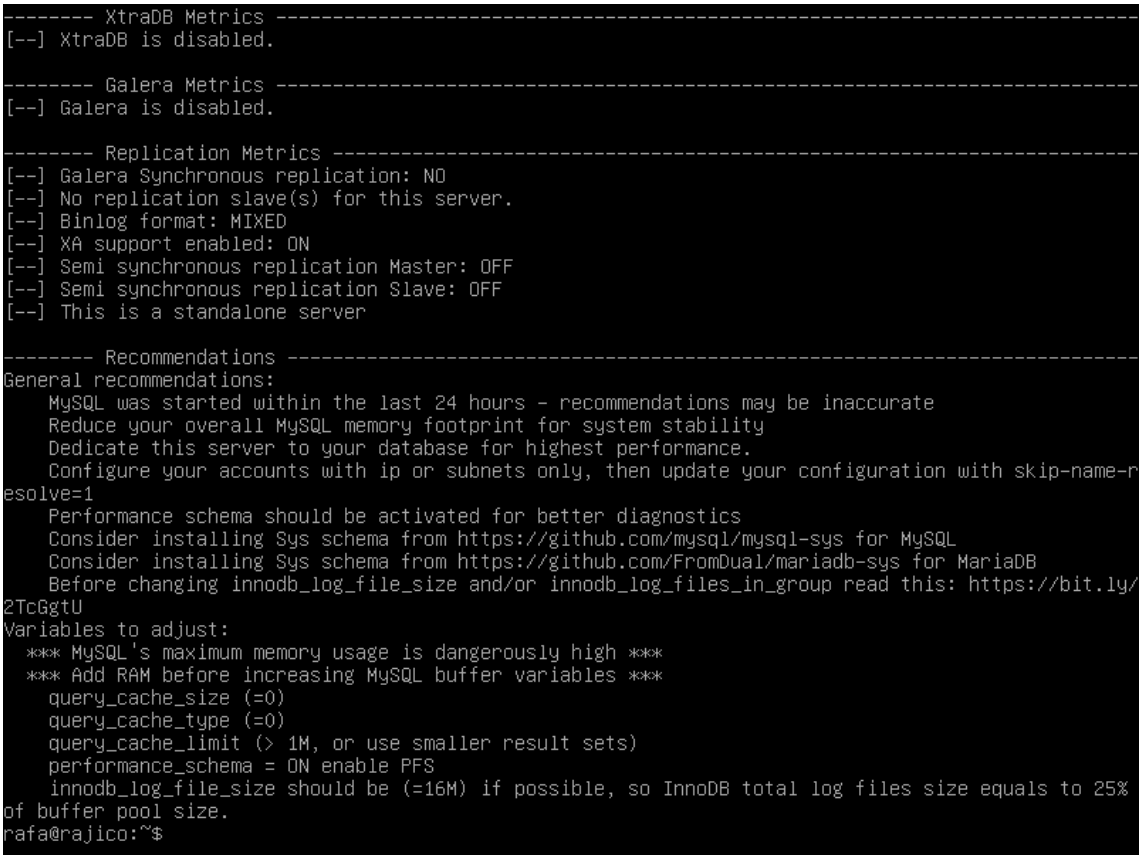
Nos bajamos también el archivo vulnerabilities.csv.

```
rafa@rajico:~$ wget https://raw.githubusercontent.com/major/MySQLTuner-perl/master/basic_passwords.t
xt -O basic_passwords.txt
--2020-09-27 17:06:22--  https://raw.githubusercontent.com/major/MySQLTuner-perl/master/basic_passwo
rds.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133, 151.101.64.133, 15
1.101.128.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.0.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3988 (3.9K) [text/plain]
Saving to: 'basic_passwords.txt'

basic_passwords.txt      100%[=================================>]   3.89K  --.-KB/s    in 0.001s

2020-09-27 17:06:22 (2.66 MB/s) - 'basic_passwords.txt' saved [3988/3988]

rafa@rajico:~$ wget https://raw.githubusercontent.com/major/MySQLTuner-perl/master/vulnerabilities.c
sv -O vulnerabilities.csv
--2020-09-27 17:06:53--  https://raw.githubusercontent.com/major/MySQLTuner-perl/master/vulnerabilit
ies.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133, 151.101.64.133, 15
1.101.128.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.0.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1185249 (1.1M) [text/plain]
Saving to: 'vulnerabilities.csv'

vulnerabilities.csv      100%[=================================>]   1.13M  3.73MB/s    in 0.3s

2020-09-27 17:06:54 (3.73 MB/s) - 'vulnerabilities.csv' saved [1185249/1185249]

rafa@rajico:~$ _
```

Ejecutamos mysqltuner:

```
rafa@rajico:~$ ./mysqltuner.pl
 >>  MySQLTuner 1.7.19 - Major Hayden <major@mhtx.net>
 >>  Bug reports, feature requests, and downloads at http://mysqltuner.pl/
 >>  Run with '--help' for additional options and output filtering

[--] Skipped version check for MySQLTuner script
Please enter your MySQL administrative login: admin
Please enter your MySQL administrative password: [OK] Currently running supported MySQL version 10.3
.23-MariaDB-0+deb10u1
[OK] Operating on 64-bit architecture

-------- Log file Recommendations ------------------------------------------------------------------
[!!] Log file /var/log/mysql/error.log doesn't exist

-------- Storage Engine Statistics -----------------------------------------------------------------
[--] Status: +Aria +CSV +InnoDB +MEMORY +MRG_MyISAM +MyISAM +PERFORMANCE_SCHEMA +SEQUENCE
[--] Data in InnoDB tables: 80.0K (Tables: 2)
[OK] Total fragmented tables: 0

-------- Analysis Performance Metrics --------------------------------------------------------------
[--] innodb_stats_on_metadata: OFF
[OK] No stat updates during querying INFORMATION_SCHEMA.

-------- Security Recommendations ------------------------------------------------------------------
[OK] There are no anonymous accounts for any database users
[OK] All database users have passwords assigned
[--] There are 620 basic passwords in the list.
```

Cuando termine de analizar el entorno y la configuración, nos dará una serie de recomendaciones y variables que ajustar.



Recomendaciones:

- No serán del todo acertadas, debido a que Mysql fue iniciado en las últimas 24 horas por primera vez.
- Reducir las huellas de memoria para estabilidad en el sistema.
- Dedicar el servidor para que la base de datos tenga el más alto rendimiento.
- Configurar las cuentas con direcciones ip o subredes y actualizar la configuración con la opción skip-name-resolve=1. (hecho)
- Activar el esquema de rendimiento para mejores diagnósticos. (hecho)
- Instalar esquema del sistema para mysql (no se puede).
- Instalar esquema del sistema para mariadb (instalado).

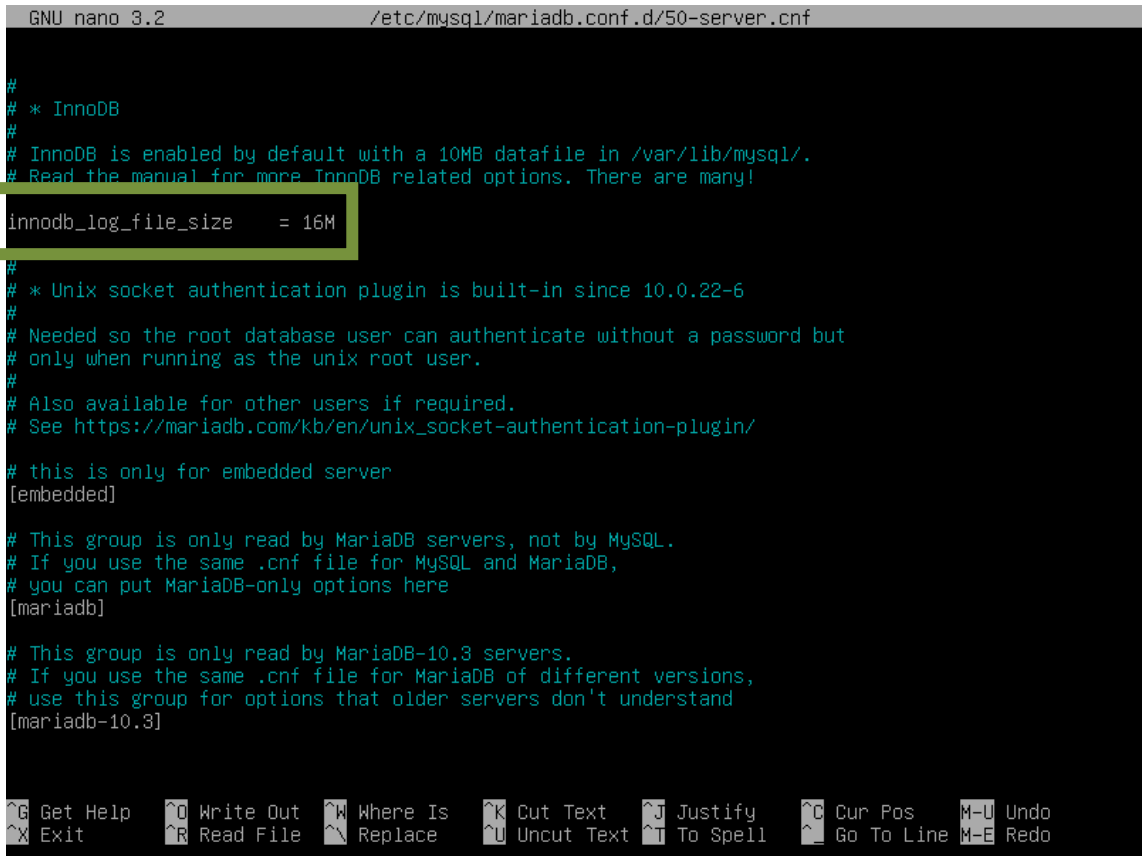
Variables que ajustar:

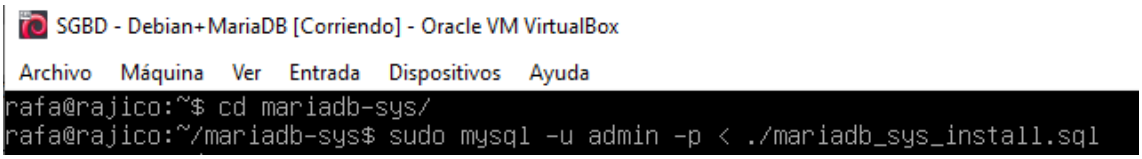
- Query_cache_size=0 (hecho).
- Query_cache_type=0(hecho).
- Query_cache_limit >1M (hecho).
- Activar esquema de rendimiento (hecho).
- Tamaño del fichero de logs innodb=16M (hecho).

A continuación, muestro las recomendaciones y variables que he ajustado:

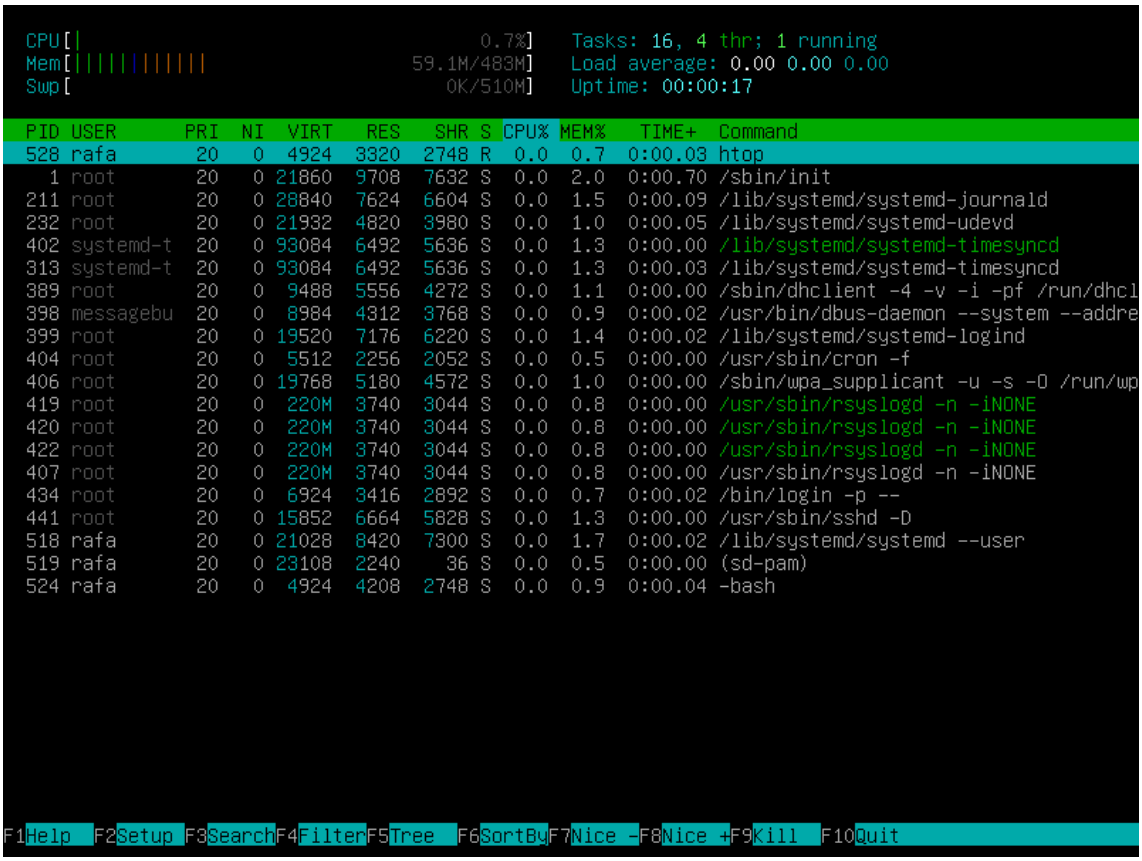
```
  GNU nano 3.2                  /etc/mysql/mariadb.conf.d/50-server.cnf

#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]
skip-name-resolve       = 1
performance_schema      = ON
#
# * Basic Settings
#
user                    = mysql
pid-file                = /run/mysqld/mysqld.pid
socket                  = /run/mysqld/mysqld.sock
#port                   = 3306
basedir                 = /usr
datadir                 = /var/lib/mysql
tmpdir                  = /tmp
lc-messages-dir         = /usr/share/mysql
#skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address            = 127.0.0.1

#
                                [ Read 138 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line M-E Redo
```

```
  GNU nano 3.2                  /etc/mysql/mariadb.conf.d/50-server.cnf


#
# * Query Cache Configuration
#
query_cache_limit       = 2M
query_cache_size        = 0M
query_cache_type        = 0

#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# As of 5.1 you can enable the log at runtime!
#general_log_file       = /var/log/mysql/mysql.log
#general_log            = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Enable the slow query log to see queries with especially long duration
#slow_query_log_file    = /var/log/mysql/mariadb-slow.log
#long_query_time        = 10
#log_slow_rate_limit    = 1000
#log_slow_verbosity     = query_plan
#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

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line M-E Redo
```


GNU nano 3.2 /etc/mysql/mariadb.conf.d/50-server.cnf
#
* InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
innodb_log_file_size = 16M
#
* Unix socket authentication plugin is built-in since 10.0.22-6
#
Needed so the root database user can authenticate without a password but
only when running as the unix root user.
#
Also available for other users if required.
See https://mariadb.com/kb/en/unix_socket-authentication-plugin/
this is only for embedded server
[embedded]
This group is only read by MariaDB servers, not by MySQL.
If you use the same .cnf file for MySQL and MariaDB,
you can put MariaDB-only options here
[mariadb]
This group is only read by MariaDB-10.3 servers.
If you use the same .cnf file for MariaDB of different versions,
use this group for options that older servers don't understand
[mariadb-10.3]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo

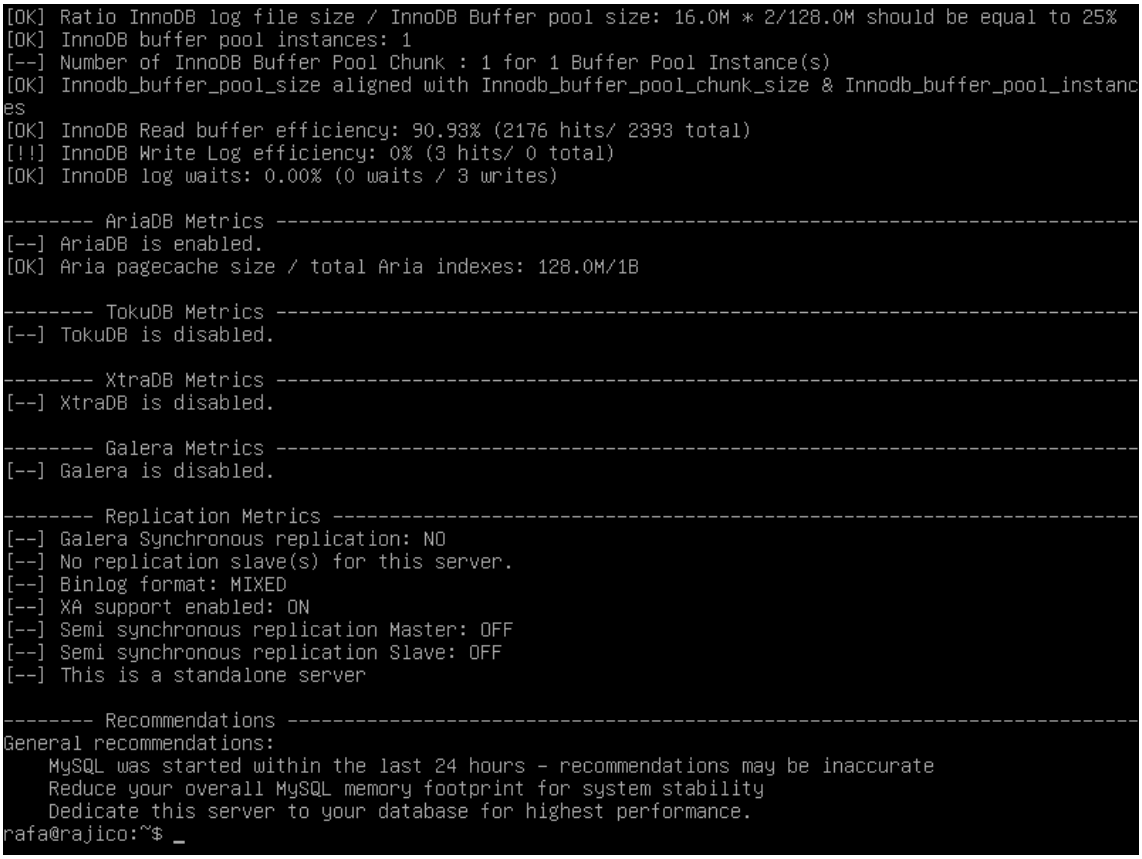

SGBD - Debian+MariaDB [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
rafa@rajico:~$ cd mariadb-sys/
rafa@rajico:~/mariadb-sys$ sudo mysql -u admin -p < ./mariadb_sys_install.sql

Tras esto, volvemos a ejecutar htop, a ver si algo ha cambiado:



Se puede apreciar que utiliza aproximadamente la mitad de memoria que usaba antes.

Por último, esta es la comparación tras ejecutar mysqltuner después de realizar la configuración recomendada:



El sistema no necesita la misma cantidad de recursos para ejecutar mariadb.