

How to Display MariaDB Database Metric Values?

written by sysadmin | 24 December 2025

I want to see the metric values of my MariaDB database.

Problem

How to display MariaDB database metric values?

Solution

As far as I know, there are 2 tools that you can use to display the values of metrics on your MariaDB database.

1. mariadb-report

This tool is built into MariaDB, so when you install the MariaDB database or mariadb-client on your server, this tool will automatically be installed on your server. In general, use the format below to run this command:

```
mariadb-report --user=username --password your_password
```

If you want to display MariaDB database metrics using the root user, then use the command below:

```
mariadb-report --user=root --password
```

Enter the password for the root user, and a display will appear as below:

MariaDB 10.11.13-MariaDB-0u uptime 0+0:52:25 Fri Nov 28 03:14:39 2025

```
-- Key -----
Buffer used      0 of 128.00M   %Used:   0.00
  Current       23.35M           %Usage:  18.24
Write hit        0.00%
Read hit         0.00%

-- Questions -----
Total           38.14k        12.1/s
  DMS           21.86k        7.0/s   %Total:  57.32
  Com_          16.26k        5.2/s           42.64
  COM_QUIT        66         0.0/s           0.17
  -Unknown        49         0.0/s           0.13
Slow 10 s        0           0/s           0.00   %DMS:   0.00 Log: OFF
DMS              21.86k        7.0/s           57.32
  SELECT         16.78k        5.3/s           43.99           76.75
  INSERT          4.20k        1.3/s           11.02           19.23
  UPDATE          668         0.2/s           1.75           3.06
  DELETE          212         0.1/s           0.56           0.97
  REPLACE          0           0/s           0.00           0.00
Com_             16.26k        5.2/s           42.64
  commit          8.08k        2.6/s           21.18
  begin           8.08k        2.6/s           21.18
  set_option        69         0.0/s           0.18

-- Rows -----
Rows            6.82M        2.2k/s
  Using idx       3.26M        1.0k/s   %Index:  47.80
Rows/question   178.83

-- SELECT and Sort -----
Scan            7.85k        2.5/s   %SELECT:  46.81
Range           5.66k        1.8/s           33.70
Full join       1.62k        0.5/s           9.67
Range check      0           0/s           0.00
Full rng join    0           0/s           0.00
Sort scan        1.05k        0.3/s
Sort range        4.20k        1.3/s
Sort mrg pass     314         0.1/s
```



Display of mariadb-report

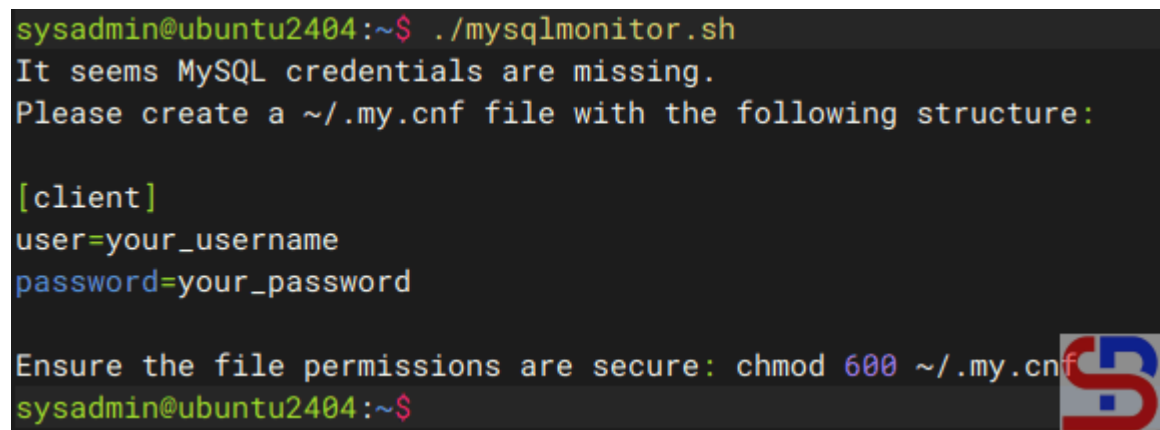
If you want to put the results into a file, you can use the command below:

```
mariadb-report --user=root --password --outfile /tmp/metric.txt
```

Enter the password, and the results will be entered into the /tmp/metric.txt file.

2. mysqlmonitor

This is a simple bash script created to give system administrators and database administrators a fast summary of MySQL metrics. It presents essential metrics such as InnoDB buffer utilization, query efficiency, and system memory, accompanied by concise descriptions of each parameter. To run this script, you must create MariaDB credentials in the my.cnf file, because if not, there will be an error like the one below:



```
sysadmin@ubuntu2404:~$ ./mysqlmonitor.sh
It seems MySQL credentials are missing.
Please create a ~/.my.cnf file with the following structure:

[client]
user=your_username
password=your_password

Ensure the file permissions are secure: chmod 600 ~/.my.cnf
sysadmin@ubuntu2404:~$
```

Error when running mysqlmonitor

Copy the script below and insert it into the my.cnf file:

```
[client]
user=username
password=your_password
```

To download and run the script, run the commands below:

```
curl -O
https://raw.githubusercontent.com/haydenjames/mysqlmonitor-script/main/mysqlm
onitor.sh &&
chmod +x mysqlmonitor.sh &&
./mysqlmonitor.sh
```

There will be a display like the image below:

----- MySQL Runtime Metrics -----		
Aborted_clients	0	Failed client connections.
Aborted_connects	2	Failed MySQL server connections.
Connections	370	Total connection attempts.
Created_tmp_disk_tables	1	Temp tables created on disk.
Created_tmp_files	788	Temp files created by MySQL.
Created_tmp_tables	1.21K	Temp tables created in memory.
Innodb_buffer_pool_pages_free	4.68K	Free pages in InnoDB buffer pool.
Innodb_buffer_pool_read_requests	2.94M	Logical read requests to buffer.
Innodb_buffer_pool_reads	3.36K	Logical reads from disk.
Innodb_buffer_pool_wait_free	0	Waits for free pages.
Innodb_buffer_pool_write_requests	197.23K	Writes requested to InnoDB buffer.
Innodb_data_reads	3.60K	Data pages read from disk.
Innodb_data_writes	0	Data pages written to disk.
Innodb_log_waits	0	Log waits for buffer flushes.
Innodb_log_writes	7.81K	Log writes to InnoDB log file.
Key_reads	0	MyISAM disk reads (Use InnoDB).
Key_writes	0	MyISAM disk writes (Use InnoDB).
Max_used_connections	32	Max concurrent connections so far.
Open_files	58	Files currently open by MySQL.
Open_tables	147	Tables currently open.
Opened_tables	154	Total tables opened since start.
Questions (12.30 QPS)	48.33K	Total number of client requests.
Select_full_join	2.03K	Joins without usable indexes.
Select_scan	10.10K	Full table scans in SELECT queries.
Slow_queries	0	Queries exceeding long_query_time.
Sort_merge_passes	392	Merge passes performed for sorting.
Sort_range	5.25K	Range-based sort operations.
Sort_rows	562.52K	Rows sorted by MySQL.
Sort_scan	1.31K	Table-scan-based sort operations.
Table_locks_immediate	355	Locks acquired immediately.
Table_locks_waited	0	Locks that had to wait. (bad).
Table_open_cache_hits	33.98K	Cache hits for table open.
Table_open_cache_misses	155	Cache misses for table open.
Threads_cached	0	Threads in the thread cache.
Threads_connected	32	Currently open connections.
Threads_created	35	Threads created for connections.
Threads_running	1	Threads currently running queries.
Uptime (Wait 24h for accuracy)	1h 5m 28s	
----- MySQL Health Metrics -----		
InnoDB Buffer Pool Free	73 MB	Zero/low? = innodb_buffer_pool_size.
InnoDB Buffer Pool Hit Ratio	99.9%	High QPS? Aim for high hit ratio
Thread Cache Hit Ratio	90.5%	Faster connection handling. > 90%
Table Cache Hit Ratio	99.5%	Faster table access speeds. > 90%

Display of mysqlmonitor

From the image above, you can see that the results from the mysqlmonitor tool are more concise than the mariadb-report tool. To exit this tool, press the **q** key.

Note

If you want detailed metric results, then you can use the mariadb-report tool. However, if you want more concise metrics, you can use the mysqlmonitor tool.

References

linuxblog.io

mariadb.com

github.com