

Details

🕒

Revision #8

★

Created 1 year ago by gviabcua

✎

Updated 2 months ago by gviabcua

💎

Referenced by 1 item

Actions

🕒

Revisions

Page Navigation

• Hardware

|

• Software

After installation - System tuning

We offer you several steps to achieve maximum performance of the system.

📘 If your network is small and your server has no problem with high CPU load - you can skip this page

Hardware

The first and main recommendation is to **keep** Grusher and operating system **up-to-date**. Today not possible to use "install and forget" principle. Updates are released help to improve performance and resolve issues.

Second recommendation for highly loaded systems is to separate the web server (Application) and database (DB) on different servers or virtual machines. This will help balance processor time and memory for each machine, and there will be no competition for a shared pool of capacity.

Software

MariaDB tuning

At the time of writing this article latest version was [11.7.2](#).

MariaDB Server 11 brings many significant improvements to the query optimizer!

Read this articles

- <https://mariadb.com/kb/en/optimization-and-tuning/>
- <https://www.cloudways.com/blog/mariadb-performance-tuning/>

Also you can use clustering (Galera Cluster) from [MariaDB](#).

- <https://galeracluster.com/>
- https://www.server-world.info/en/note?os=Ubuntu_22.04&p=mariadb&f=5

MariaDB config file for performance (optional, use on your own risk)

✔ This config is optimise for VM with 6 CPU and 10 RAM. The database server is separate from the application

```
[mariadbd]
# Enable the performance schema.
performance_schema = ON
#Skip DNS resolving
skip-name-resolve = ON

#The number of file I/O threads in InnoDB is set by the options innodb_read_io_threads, innodb_write_io_threads
#usually this parameter is set to 4 or 8, on fast SSD drives set to 16.
innodb_read_io_threads = 16
innodb_write_io_threads = 16
#Enable/disable InnoDB deadlock detector (default ON). if set to OFF, deadlock detection is skipped, and we rely
innodb_deadlock_detect = 0
innodb_lock_wait_timeout = 3000
#Number of undo tablespaces to use.
innodb_undo_tablespaces=8
#Enable or Disable Truncate of UNDO tablespace.
innodb_undo_log_truncate=ON
#Desired maximum UNDO tablespace size in bytes
innodb_max_undo_log_size=2G
#Dictates rate at which UNDO records are purged. Value N means purge rollback segment(s) on every Nth iteration
innodb_purge_rseg_truncate_frequency=64
#The size of the memory buffer InnoDB uses to cache data and indexes of its tables.
innodb_buffer_pool_size = 6G
#Size of each log file in a log group.
innodb_log_file_size = 2G

#Other
#The size in bytes of MyISAM key cache, which is the buffer used for MyISAM index blocks.
key_buffer_size = 128M
#The number of cached table definitions
table_definition_cache = 405

# Logging
log_output=FILE
slow_query_log_file = /var/log/mysql/slow.log
slow_query_log
long_query_time = 10.0
log_queries_not_using_indexes=0
log_error=/var/log/mysql/mariadb.err
```

Ngnix

If you have a lot of active system users (more than 10) or you actively use the Grusher API, you need to configure php-fpm.

Otherwise, requests submitted to the web will take a long time to be processed.

- [PHP-FPM Process Calculator](#)

Grusher

- Disable poller logging. This option disable writing to disk poller logs (up to ~ -15% for CPU load in some case)
 - <https://grusher.gviabcua.in.ua/link/63#bkmrk-poller-log-settings>
- Switch session to "cookie"
 - <https://grusher.gviabcua.in.ua/link/63#bkmrk-typical-file-config>
 - SESSION_DRIVER = 'cookie'

⚠ **Do not use too many pollers.** Everything should be in moderation. Do experiment with different numbers of pollers and determine the most optimal count.

⚠ **Do not run heavy tasks too often** via scheduler

No Comments