

mysql/mariadb - Configuration et optimisation

Cette documentation explique la configuration et la customisation d'un serveur mariadb, afin d'améliorer ses performances.

Elle fait office de mémo pour la compréhension de ces paramétrages

Ici la configuration est à destination d'un serveur avec 16G de RAM. Elle sera donc à adapter en fonction des machine, avec plus ou moins de RAM.

Fichiers de configuration

Ces paramètres sont à mettre dans les fichiers de config de mariadb.

Sous debian:

- `/etc/mysql/mariadb.cnf`
ou
- `/etc/mysql/mariadb.conf.d/50-server.cnf` (config spécifique à mariadb)

Sous Redhat Entreprise Linux et dérivée:

- `/etc/my.cnf`

Fichier de configuration d'exemple

Exemple pour un système disposant de 16 Go de RAM

```
[mysqld]
# Settings généraux
user = mysql
pid-file = /run/mysqld/mysqld.pid
basedir = /usr
datadir = /var/lib/mysql
tmpdir = /tmp
# Désactivation résolution de nom
skip-name-resolve
# Interface d'écoute
bind-address = 127.0.0.1
```

```
# Paramétrages performances
innodb_buffer_pool_size = 8G
innodb_buffer_pool_instances = 8
innodb_log_file_size = 512M
thread_cache_size = 16
query_cache_size = 128M
query_cache_type = 1
tmp_table_size = 1024M
max_heap_table_size = 1024M
table_open_cache = 4096
table_definition_cache = 4096

# Paramétrage connexions
max_connections = 500
max_user_connections = 50
```

Explication des paramétrages pour la customisation des performances:

- **innodb_buffer_pool_size:** Mémoire RAM utilisée pour le cache innodb. Cela diminue les accès disques, améliorant les performances. Il est recommandé de mettre la moitié de la RAM (ici il y a 16G sur le système)
- **innodb_buffer_pool_instances:** Nb d'instances "buffer pool" d'innodb. Mettre autant d'instance que de RAM affectée (ici 8)
- **innodb_log_file_size:** Taille fichiers journaux d'innodb. L'augmentation de ce paramètres, permet d'améliorer les perfs
- **thread_cache_size:** Nombre de threads à mettre en cache, pour les réutiliser. Cela permet de réutiliser le thread mis en cache, si un client se reconnecte
- **query_cache_size:** Cache des requêtes. Le résultat de requêtes SQL sera mis en cache, afin d'accélérer les performances
- **query_cache_type = 1:** Activation du cache de requêtes
- **max_connections:** Nombre max de connexions acceptées par le serveur. Au delà, les nouvelles connexions seront refusées
- **max_user_connections:** Nombre max d'utilisateurs connectés
- **tmp_table_size et max_heap_table_size:** Cache mémoire des tables temporaires
- **table_open_cache et table_definition_cache:** Nombre des tables SQL à garder ouverte

Après toute modification de la configuration, il faudra redémarrer le service afin qu'ils soient pris en compte:

```
systemctl restart mariadb
```

Ressources

- [Several Ways to Optimize MySQL/MariaDB for Performance](#)

Revision #1

Created 17 April 2025 08:59:22 by Lauris_Adm

Updated 17 April 2025 09:32:51 by Lauris_Adm