

MySQL

Administration 2

- Installation a la main
- Installation sur Debian
- MySQL sur Docker
 - Installation de docker avec Mysql
- Fonctions avancées de l'administration
 - MySQLd Multi
 - Tuning InnoDB
 - Mysql Query Rewriter
 - my.ini
- Replication
- SSL et Utilisateurs
 - SSL
 - Utilisateur et Role
- TP 1 Prise en main
 - Démarrage de MySQL
- MySQL Cluster
 - MySQLCluster sur docker
 - MySQL Cluster On Premise
- Cluster Galera

- Cluster Galera
- Safe-To-Bootstrap
- Reprise sur incident

Installation a la main

Résolution des dépendances

MySQL dépend de la bibliothèque libaio. L'initialisation du répertoire de données et les étapes de démarrage du serveur suivantes échoueront si cette bibliothèque n'est pas installée localement.

```
pilou@ubuntu: ~$ sudo apt-get install libaio1 [sudo] password for pilou: Reading package
lists...
Done Building dependency tree Reading state information...
Done libaio1 is already the newest version (0.3.111-1).
libaio1 set to manually installed. 0 upgraded, 0 newly installed, 0 to remove and 68 not
upgraded.
```

Téléchargement de décompression de l'archive

Pour installer une distribution binaire de fichier tar compressée, décompressez-la à l'emplacement d'installation de votre choix.

```
pilou@ubuntu: ~$ mkdir mysql80

pilou@ubuntu: ~$ cd mysql80

pilou@ubuntu: ~/mysql80$ wget https://dev.mysql.com/get/Downloads/MySQL-8.0/mysql-8.0.13-
linux-glibc2.12-x86_64.tar --2018-12-30 18:10:42--

https://dev.mysql.com/get/Downloads/MySQL-8.0/mysql-8.0.13-linux-glibc2.12-x86_64.tar
Resolving dev.mysql.com (dev.mysql.com)...

137.254.60.11 Connecting to dev.mysql.com (dev.mysql.com)|137.254.60.11|:443... connected.
```

```
HTTP request sent, awaiting response... 302 Found Location:
https://cdn.mysql.com//Downloads/MySQL-8.0/mysql-8.0.13-linux-glibc2.12-x86_64.tar [following]
--2018-12-30 18:10:48-- https://cdn.mysql.com//Downloads/MySQL-8.0/mysql-8.0.13-linux-
glibc2.12-x86_64.tar Resolving cdn.mysql.com (cdn.mysql.com)... 23.210.41.222 Connecting to
cdn.mysql.com (cdn.mysql.com)|23.210.41.222|:443... connected. HTTP request sent, awaiting
response... 200 OK Length: 472883200 (451M) [application/x-tar] Saving to: 'mysql-8.0.13-
linux-glibc2.12-x86_64.tar'

mysql-8.0.13-linux-glib 100%[=====>] 450,98M 6,86MB/s in 72s

2018-12-30 18:12:01 (6,22 MB/s) - 'mysql-8.0.13-linux-glibc2.12-x86_64.tar' saved
[472883200/472883200]

pilou@ubuntu: ~/mysql80$ ls mysql-8.0.13-linux-glibc2.12-x86_64.tar
```

L'archive contient le serveur MySQL, l'outil MySQL Router ainsi qu'un banc de test pour MySQL.

```
pilou@ubuntu: ~/mysql80$ tar xvf mysql-8.0.13-linux-glibc2.12-x86_64.tar mysql-8.0.13-linux-
glibc2.12-x86_64.tar.xz mysql-router-8.0.13-linux-glibc2.12-x86_64.tar.xz mysql-test-8.0.13-
linux-glibc2.12-x86_64.tar.xz
pilou@ubuntu: ~/mysql80$ tar xvf mysql-8.0.13-linux-glibc2.12-x86_64.tar.xz
pilou@ubuntu: ~/mysql80$ ls mysql-8.0.13-linux-glibc2.12-x86_64 bin include LICENSE man
README.router support-files docs lib LICENSE.router README share
```

L'archive une fois décompresser contient:

- bin binaire mysqld, et outils client
- docs Documentation
- man Documentation
- include fichier de développement
- lib fichier de développement
- share Fichiers partagé par les bases MySQL
- support-files fichier support

NB: A ce niveau il n'y a pas de repertoire data qui a été crée.

Sous ubuntu, faire

```
cd /usr/lib/x86_64-linux-gnu
sudo ln -s libtinfo.so.6 libtinfo.so.5
```

Le répertoire bin contient:

```
pilou@pilou-pc: ~/Formation/mysql-8.0.20-linux-glibc2.12-x86_64$ ls
bin docs include lib LICENSE man README share support-files
```

Initialisation de mysql

L'initialisation de MySQL permet de créer le repertoire data et de créer un utilisateur root

```
pilou@pilou-pc: ~/Formation/mysql-8.0.20-linux-glibc2.12-x86_64$ ./bin/mysqld --defaults-
file=simplemy.ini --console --initialize
pilou@pilou-pc: ~/Formation/mysql-8.0.20-linux-glibc2.12-x86_64$ ./bin/mysqld --defaults-
file=simplemy.ini --console
```

Le fichier ini étant:

```
[mysqld]
basedir=/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64
datadir=/home/pilou/Formation/simpleinit/data
log-error=/home/pilou/Formation/simpleinit/mysqld.log
```

Dans le fichier de log on retrouve le mot de passe de l'administrateur:

```
2020-05-11T21:03:21.864864Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2020-05-11T21:03:23.439599Z 6 [Note] [MY-010454] [Server] A temporary password is generated
for root@localhost: e;ActH??b0Es
2020-05-11T21:03:33.232344Z 0 [System] [MY-010116] [Server] /home/pilou/Formation/mysql-
8.0.20-linux-glibc2.12-x86_64/bin/mysqld (mysqld 8.0.20) starting as process 13368
```

Test de MySQL

```
pilou@ubuntu: ~$ mysql -S /tmp/mysql.sock -u root -p
Enter password: Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection
id is 10 Server version: 8.0.13

Copyright (c) 2000, 2018, 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>
```

Post Installation

La première chose à faire est de changer le password du root en utilisant l'outil mysql

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'piloupilou'; Query OK, 0 rows affected
(0.04 sec)

mysql> quit
```

Pour arrêter le serveur, l'outil cli mysqladmin permet de demander l'arrêt du serveur

```
/bin/mysqladmin -h localhost -u root -p shutdown
```

Côté serveur cela donne

```
2018-12-31T17:01:56.370223Z 15 [System] [MY-013172] [Server] Received SHUTDOWN from user
root.
```

Création d'un fichier ini de base

Sur le serveur, dans le repertoire de mysql, on va créer un fichier my.cnf minimaliste afin de setter quelques variable

```
[mysqld]
port = 3306
socket = /tmp/mysql.sock skip-external-locking
key_buffer_size = 16K
max_allowed_packet = 1M
table_open_cache = 4
sort_buffer_size = 64K
read_buffer_size = 256K
read_rnd_buffer_size = 256K
net_buffer_length = 2K
thread_stack = 128K
```

Installation sur Debian

Installation

L'installation se fait via apt

```
sudo apt-get update
sudo apt-get install mysql-server
```

Après l'installation, deux utilisateurs sont créés dans le fichier `debian.cnf`

```
sudo cat /etc/mysql/debian.cnf
# Automatically generated for Debian scripts. DO NOT TOUCH!
[client]
host      = localhost
user      = debian-sys-maint
password  = J0mSbYCrYAn2NmLY
socket    = /var/run/mysqld/mysqld.sock
[mysql_upgrade]
host      = localhost
user      = debian-sys-maint
password  = J0mSbYCrYAn2NmLY
socket    = /var/run/mysqld/mysqld.sock
```

Ce qui donne après connexion:

```
pilou@pilou-pc: ~$ mysql -u debian-sys-maint -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.20-0ubuntu0.19.10.1 (Ubuntu)
```

La configuration du serveur est dans `/etc/mysql` avec deux répertoires 1 pour la configuration du serveur et 1 pour le client:

```
#
# The MySQL database server configuration file.
```

```
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file          = /var/run/mysqld/mysqld.pid
# socket            = /var/run/mysqld/mysqld.sock
# port              = 3306
# datadir           = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar\_tmpdir
# tmpdir            = /tmp
#
# 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
#
# * Fine Tuning
#
key_buffer_size     = 16M
# max_allowed_packet = 64M
# thread_stack      = 256K

# thread_cache_size = -1
```

```
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
mysam-recover-options = BACKUP

# max_connections          = 151

# table_open_cache        = 4000

#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
# general_log_file         = /var/log/mysql/query.log
# general_log              = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
# slow_query_log           = 1
# slow_query_log_file     = /var/log/mysql/mysql-slow.log
# long_query_time = 2
# 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
# binlog_expire_logs_seconds = 2592000
max_binlog_size = 100M
# binlog_do_db           = include_database_name
# binlog_ignore_db      = include_database_name
```

Securisation

L'idée est d'avoir une base de sécurisation pour MySQL

```
pilou@pilou-pc: ~$ sudo mysql_secure_installation utility
```

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords and improve security. It checks the strength of password and allows the users to set only those passwords which are secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: Y

There are three levels of password validation policy:

LOW Length >= 8

MEDIUM Length >= 8, numeric, mixed case, and special characters

STRONG Length >= 8, numeric, mixed case, special characters and dictionary file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 2

Please set the password for root here.

New password:

Re-enter new password:

Estimated strength of the password: 50

Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) :
Y

By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y

Success.

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y

Success.

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y

- Dropping test database...

Success.

- Removing privileges on test database...

Success.

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y

Success.

All done!

Securisation Systeme

Installation de Lynis

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
C80E383C3DE9F082E01391A0366C67DE91CA5D5F
sudo apt install apt-transport-https
sudo apt update
```

```
sudo apt install lynis
```

L'execution de Lynis permet de valider l'installation d'une machine

```
./lynis audit system
```

```
[ Lynis 3.0.0 ]
```

```
#####
```

```
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are  
welcome to redistribute it under the terms of the GNU General Public License.  
See the LICENSE file for details about using this software.
```

```
2007-2020, CISOfy - https://cisofy.com/lynis/
```

```
Enterprise support available (compliance, plugins, interface and tools)
```

```
#####
```

```
[+] Initializing program
```

```
-----
```

```
- Detecting OS... [ DONE ]  
- Checking profiles... [ DONE ]
```

```
-----
```

```
Program version:      3.0.0  
Operating system:     Linux  
Operating system name: Ubuntu  
Operating system version: 19.10  
Kernel version:      5.3.0  
Hardware platform:   x86_64  
Hostname:            pilou-pc
```

```
-----
```

```
Profiles:             /home/pilou/lynis/lynis/default.prf  
Log file:             /var/log/lynis.log  
Report file:         /var/log/lynis-report.dat  
Report version:      1.0  
Plugin directory:    ./plugins
```

```
-----
```

```
Auditor:              [ Not Specified ]  
Language:             en
```

Test category: all

Test group: all

- Program update status... [NO UPDATE]

[+] System Tools

- - Scanning available tools...
- Checking system binaries...

[+] Plugins (phase 1)

Note: plugins have more extensive tests and may take several minutes to complete

- Plugin: pam
[...]
- Plugin: systemd
[.....]

[+] Boot and services

- - Service Manager [systemd]
- Checking UEFI boot [DISABLED]
- Checking presence GRUB2 [FOUND]
- Checking for password protection [NONE]
- Check running services (systemctl) [DONE]
Result: found 32 running services
- Check enabled services at boot (systemctl) [DONE]
Result: found 59 enabled services
- Check startup files (permissions) [OK]
- Running 'systemd-analyze security'
- ModemManager.service: [MEDIUM]
- NetworkManager.service: [EXPOSED]
- accounts-daemon.service: [UNSAFE]
- acpid.service: [UNSAFE]
- alsa-state.service: [UNSAFE]
- anacron.service: [UNSAFE]
- appport.service: [UNSAFE]
- avahi-daemon.service: [UNSAFE]

```
- bluetooth.service: [ MEDIUM ]
- cron.service: [ UNSAFE ]
- cups-browsed.service: [ UNSAFE ]
- cups.service: [ UNSAFE ]
- dbus.service: [ UNSAFE ]
- dm-event.service: [ UNSAFE ]
- dmesg.service: [ UNSAFE ]
- emergency.service: [ UNSAFE ]
- getty@tty1.service: [ UNSAFE ]
- grub-common.service: [ UNSAFE ]
- haveged.service: [ MEDIUM ]
- irqbalance.service: [ MEDIUM ]
- kerneloops.service: [ UNSAFE ]
- lvm2-lvmpolld.service: [ UNSAFE ]
- mysql.service: [ UNSAFE ]
- networkd-dispatcher.service: [ UNSAFE ]
- ofono.service: [ UNSAFE ]
- ondemand.service: [ UNSAFE ]
- packagekit.service: [ UNSAFE ]
- plymouth-start.service: [ UNSAFE ]
- polkit.service: [ UNSAFE ]
- rc-local.service: [ UNSAFE ]
- rescue.service: [ UNSAFE ]
- rsync.service: [ UNSAFE ]
- rsyslog.service: [ UNSAFE ]
- rtkit-daemon.service: [ MEDIUM ]
- sddm.service: [ UNSAFE ]
- snapd.service: [ UNSAFE ]
- systemd-ask-password-console.service: [ UNSAFE ]
- systemd-ask-password-plymouth.service: [ UNSAFE ]
- systemd-ask-password-wall.service: [ UNSAFE ]
- systemd-fsckd.service: [ UNSAFE ]
- systemd-initctl.service: [ UNSAFE ]
- systemd-journald.service: [ OK ]
- systemd-logind.service: [ OK ]
- systemd-networkd.service: [ OK ]
- systemd-resolved.service: [ OK ]
- systemd-rfkill.service: [ UNSAFE ]
- systemd-timesyncd.service: [ OK ]
```

```
- systemd-udev.service: [ EXPOSED ]
- thermald.service: [ UNSAFE ]
- udisks2.service: [ UNSAFE ]
- unattended-upgrades.service: [ UNSAFE ]
- upower.service: [ OK ]
- user@1000.service: [ UNSAFE ]
- uuid.service: [ OK ]
- vboxadd-service.service: [ UNSAFE ]
- whoopsie.service: [ UNSAFE ]
- wpa_supplicant.service: [ UNSAFE ]
```

Optimisation

l'outil mySQLtuner permet de valider un fichier my.cnf

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

[--] Skipped version check for MySQLTuner script
Please enter your MySQL administrative login: debian-sys-maint
Please enter your MySQL administrative password: [OK] Currently running supported MySQL
version 8.0.20-0ubuntu0.19.10.1
[OK] Operating on 64-bit architecture

----- Log file Recommendations
-----

[OK] Log file /var/log/mysql/error.log exists
[--] Log file: /var/log/mysql/error.log(3K)
[OK] Log file /var/log/mysql/error.log is readable.
[OK] Log file /var/log/mysql/error.log is not empty
[OK] Log file /var/log/mysql/error.log is smaller than 32 Mb
[!!] /var/log/mysql/error.log contains 6 warning(s).
[!!] /var/log/mysql/error.log contains 3 error(s).
[--] 4 start(s) detected in /var/log/mysql/error.log
[--] 1) 2020-06-09T07:43:48.981428Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready
for connections. Version: '8.0.20-0ubuntu0.19.10.1' socket: '/var/run/mysqld/mysqld.sock'
```

```
port: 3306 (Ubuntu).
[--] 2) 2020-06-09T07:43:48.910374Z 0 [System] [MY-011323] [Server] X Plugin ready for
connections. Socket: '/var/run/mysqld/mysqld.sock' bind-address: '::' port: 33060
[--] 3) 2020-06-09T07:43:46.480121Z 0 [System] [MY-011323] [Server] X Plugin ready for
connections. Bind-address: '::' port: 33060
[--] 4) 2020-06-09T07:43:43.480774Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready
for connections. Version: '8.0.20-0ubuntu0.19.10.1' socket:
'/tmp/tmp.pYBOETvCu0/mysqld.sock' port: 0 (Ubuntu).
[--] 2 shutdown(s) detected in /var/log/mysql/error.log
[--] 1) 2020-06-09T07:43:47.628975Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown
complete (mysqld 8.0.20-0ubuntu0.19.10.1) (Ubuntu).
[--] 2) 2020-06-09T07:43:45.029894Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown
complete (mysqld 8.0.20-0ubuntu0.19.10.1) (Ubuntu).

----- Storage Engine Statistics
-----
[--] Status: +ARCHIVE +BLACKHOLE +CSV -FEDERATED +InnoDB +MEMORY +MRG_MYISAM +MyISAM
+PERFORMANCE_SCHEMA
[--] Data in InnoDB tables: 16.0K (Tables: 1)
[OK] Total fragmented tables: 0

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

----- Security Recommendations
-----
[--] Skipped due to unsupported feature for MySQL 8

----- CVE Security Recommendations
-----
[OK] NO SECURITY CVE FOUND FOR YOUR VERSION

----- Performance Metrics
-----
[--] Up for: 32m 53s (118 q [0.060 qps], 57 conn, TX: 255K, RX: 14K)
[--] Reads / Writes: 98% / 2%
[--] Binary logging is enabled (GTID MODE: OFF)
[--] Physical Memory : 2.9G
```

```
[--] Max MySQL memory      : 9.8G
[--] Other process memory: 0B
[--] Total buffers: 176.0M global + 65.1M per thread (151 max threads)
[--] P_S Max memory usage: 72B
[--] Galera GCache Max memory usage: 0B
[OK] Maximum reached memory usage: 241.1M (8.06% of installed RAM)
[!!] Maximum possible memory usage: 9.8G (334.59% of installed RAM)
[!!] Overall possible memory usage with other process exceeded memory
[OK] Slow queries: 0% (0/118)
[OK] Highest usage of available connections: 0% (1/151)
[!!] Aborted connections: 19.30% (11/57)
[!!] name resolution is active : a reverse name resolution is made for each new connection and
can reduce performance
[--] Query cache have been removed in MySQL 8
[OK] Sorts requiring temporary tables: 0% (0 temp sorts / 7 sorts)
[OK] No joins without indexes
[OK] Temporary tables created on disk: 0% (0 on disk / 11 total)
[OK] Thread cache hit rate: 96% (2 created / 57 connections)
[OK] Table cache hit rate: 81% (337 open / 416 opened)
[OK] table_definition_cache(2000) is upper than number of tables(311)
[OK] Open file limit used: 0% (6/10K)
[OK] Table locks acquired immediately: 100% (8 immediate / 8 locks)
[OK] Binlog cache memory access: 100.00% (3 Memory / 3 Total)

----- Performance schema
-----
[--] Memory used by P_S: 72B
[--] Sys schema is installed.

----- ThreadPool Metrics
-----
[--] ThreadPool stat is disabled.

----- MyISAM Metrics
-----
[--] MyISAM Metrics are disabled on last MySQL versions.

----- InnoDB Metrics
-----
[--] InnoDB is enabled.
```

```
[--] InnoDB Thread Concurrency: 0
[OK] InnoDB File per table is activated
[OK] InnoDB buffer pool / data size: 128.0M/16.0K
[!!] Ratio InnoDB log file size / InnoDB Buffer pool size (75 %): 48.0M * 2/128.0M should be
equal to 25%
[OK] InnoDB buffer pool instances: 1
[--] Number of InnoDB Buffer Pool Chunk : 1 for 1 Buffer Pool Instance(s)
[OK] Innodb_buffer_pool_size aligned with Innodb_buffer_pool_chunk_size &
Innodb_buffer_pool_instances
[OK] InnoDB Read buffer efficiency: 96.90% (27237 hits/ 28107 total)
[OK] InnoDB Write log efficiency: 90.05% (724 hits/ 804 total)
[OK] InnoDB log waits: 0.00% (0 waits / 80 writes)
```

----- AriaDB Metrics

```
-----
[--] AriaDB is disabled.
```

----- TokuDB Metrics

```
-----
[--] TokuDB is disabled.
```

----- XtraDB Metrics

```
-----
[--] XtraDB is disabled.
```

----- Galera Metrics

```
-----
[--] Galera is disabled.
```

----- Replication Metrics

```
-----
[--] Galera Synchronous replication: NO
[--] No replication slave(s) for this server.
[--] Binlog format: ROW
[--] XA support enabled: ON
[--] Semi synchronous replication Master: Not Activated
[--] Semi synchronous replication Slave: Not Activated
[--] This is a standalone server
```

----- Recommendations

General recommendations:

Control warning line(s) into /var/log/mysql/error.log file

Control error line(s) into /var/log/mysql/error.log file

MySQL was started within the last 24 hours - recommendations may be inaccurate

Reduce your overall MySQL memory footprint for system stability

Dedicate this server to your database for highest performance.

Reduce or eliminate unclosed connections and network issues

Configure your accounts with ip or subnets only, then update your configuration with skip-name-resolve=1

Before changing innodb_log_file_size and/or innodb_log_files_in_group read this:

<https://bit.ly/2TcGgtU>

Variables to adjust:

*** MySQL's maximum memory usage is dangerously high ***

*** Add RAM before increasing MySQL buffer variables ***

innodb_log_file_size should be (=16M) if possible, so InnoDB total log files size equals to 25% of buffer pool size.

MySQL sur Docker

Installation de docker avec Mysql

Le but de ce chapitre est de montrer comment on peut utiliser MySQL avec Docker et ceci afin de faciliter les tests sur une seule machine.

Sur debian, il faut commencer par installer docker.

Installation des dépendances

```
$ sudo apt-get update

$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
```

Ajout de la clef GPG

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Installation du repository Docker

```
sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

Puis enfin installation de docker en lui même

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Afin de plus avoir de sudo a faire sous Debian:

```
sudo groupadd docker
sudo gpasswd -a $USER docker
```

Installation de MySQL.

Il faut télécharger l'image de mysql

```
docker pull mysql:latest
```

Il est possible maintenant de voir les différentes images dans docker dont celle de mysql

```
sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysql/mysql-server	latest	716286be47c6	11 days ago	381MB
hello-world	latest	bf756fb1ae65	4 months ago	13.3kB

Executons maintenant un premier MySQL:

```
docker run --name=mysql1 -d mysql:8.0
3436bc797ede4432cb8e82b5ab0f1f5371adac1027011632cd304f9db6dd0d78
```

Regardons maintenant les logs de MySQL:

```
docker logs mysql1
sudo docker logs mysql1
[Entrypoint] MySQL Docker Image 8.0.20-1.1.16
[Entrypoint] No password option specified for new database.
[Entrypoint] A random onetime password will be generated.
[Entrypoint] Initializing database
2020-05-08T08:40:34.134883Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.20)
initializing of server in progress as process 20
2020-05-08T08:40:34.143161Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2020-05-08T08:40:34.908824Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2020-05-08T08:40:36.494602Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an
empty password ! Please consider switching off the --initialize-insecure option.
[Entrypoint] Database initialized
2020-05-08T08:40:41.242714Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.20)
starting as process 65
2020-05-08T08:40:41.263701Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
```

```
2020-05-08T08:40:41.463561Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2020-05-08T08:40:41.594145Z 0 [System] [MY-011323] [Server] X Plugin ready for connections.
Socket: '/var/run/mysqld/mysqld.sock'
2020-05-08T08:40:41.725532Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2020-05-08T08:40:41.754130Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.20' socket: '/var/lib/mysql/mysql.sock' port: 0 MySQL Community Server - GPL.
Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/leapseconds' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/tzdata.zi' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
[Entrypoint] GENERATED ROOT PASSWORD: Ikufok=4m0s@DAmbYmvanilILUz!
```

et faisons un premier test en lançant un bash dans le container:

```
sudo docker exec -it mysql1 bash

bash-4.2# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 8.0.20

Copyright (c) 2000, 2020, 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>
```

Arretons maintenant notre serveur:

```
sudo docker stop mysql1
sudo docker logs mysql1
```

```

2020-05-08T08:40:43.905238Z 10 [System] [MY-013172] [Server] Received SHUTDOWN from user root.
Shutting down mysqld (Version: 8.0.20).
2020-05-08T08:40:46.514959Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown
complete (mysqld 8.0.20) MySQL Community Server - GPL.
[Entrypoint] Server shut down
[Entrypoint] Setting root user as expired. Password will need to be changed before database
can be used.

[Entrypoint] MySQL init process done. Ready for start up.

[Entrypoint] Starting MySQL 8.0.20-1.1.16
2020-05-08T08:40:47.139696Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.20)
starting as process 1
2020-05-08T08:40:47.155840Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2020-05-08T08:40:47.383340Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2020-05-08T08:40:47.485572Z 0 [System] [MY-011323] [Server] X Plugin ready for connections.
Socket: '/var/run/mysqld/mysqlx.sock' bind-address: '::' port: 33060
2020-05-08T08:40:47.568434Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self
signed.
2020-05-08T08:40:47.602200Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for
connections. Version: '8.0.20' socket: '/var/lib/mysql/mysql.sock' port: 3306 MySQL
Community Server - GPL.
2020-05-08T08:52:09.535241Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown
complete (mysqld 8.0.20) MySQL Community Server - GPL.
pilou@pilou-pc: ~/Formation$ sudo docker rm mysql1
mysql1

```

un docker lancé en ces termes, nous allons lancer mysql avec la persistance des données hors du docker:

```

docker run --name=mysql2 -p 5000:3306 -v
/home/pilou/Formation/simpleserver/mysql.conf.d:/etc/my.cnf.d \
-v /home/pilou/Formation/simpleserver/datadir:/var/lib/mysql \
-e MYSQL_ROOT_PASSWORD=root -v /home/pilou/Formation/simpleserver/log:/var/log \
mysql: 5.7.30

```

Ici nous allons positionner un mysql nommé mysql2 sur le port 5000 de l'hôte avec un fichier my.ini

Fonctions avancées de l'administration

MySQLd Multi

Lors de la mise en place de la replication, il est intéressant de pouvoir lancer plusieurs MySQL en une fois. L'utilitaire `mysql_multi` est la pour ca La première étape de la configuration de `mysqld_multi` est la création de deux groupes `[mysqld]` distincts dans le fichier `my.cnf` existant.

Assurez-vous que l'utilisateur MySQL, qui arrête les services `mysqld`, a le même mot de passe pour tous les serveurs MySQL accessibles par `mysqld_multi`.

Cet utilisateur doit avoir le privilège `'Shutdown_priv'`, mais pour des raisons de sécurité raisons ne devraient pas avoir d'autres privilèges. Il est conseillé de créer un utilisateur `'multi_admin'` commun à tous les serveurs MySQL contrôlés par `mysqld_multi`.

```
GRANT SHUTDOWN ON *.* TO multi_admin @ localhost IDENTIFIÉ PAR 'password'
```

Sur les serveurs, il faut:

- Chaque serveur en nécessite un port TCP et un socket Unix unique.
- Chaque serveur doit avoir un `datadir` différents

Creation des serveurs

```
pilou@pilou-pc: ~/Formation/mysql-8.0.20-linux-glibc2.12-x86_64$ ./bin/mysqld --initialize --basedir=/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64 --datadir=/home/pilou/Formation/mysqld_multi/data1 --log-error=/home/pilou/Formation/mysqld_multi/logerror1
pilou@pilou-pc: ~/Formation/mysql-8.0.20-linux-glibc2.12-x86_64$ ./bin/mysqld --initialize --basedir=/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64 --datadir=/home/pilou/Formation/mysqld_multi/data2 --log-error=/home/pilou/Formation/mysqld_multi/logerror2
```

Sur chaque serveur, on modifie le mot de passe de root et on crée l'utilisateur `multi_admin`

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'piloupilou'
-> ;
Query OK, 0 rows affected (0.02 sec)

mysql> create user 'mysqlmulti'@'localhost' IDENTIFIED BY 'mysqlmulti'
-> ;
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT SHUTDOWN ON *.* to 'mysqlmulti'@'localhost';
Query OK, 0 rows affected (0.02 sec)
```

Fichier de configuration.

Il faut ensuite créer le fichier de configuration pour mysqld_multi

```
[mysqld_multi]
mysqld = /home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld
mysqladmin = /home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqladmin
log=/home/pilou/Formation/mysqld_multi/mysqld_multi.log
user = mysqlmulti
pass = mysqlmulti

[mysqld1]
port = 3306
mysqld = /home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld
socket = /tmp/mysql.sock1
skip-external-locking
key_buffer_size = 16K
max_allowed_packet = 1M
table_open_cache = 4
sort_buffer_size = 64K
read_buffer_size = 256K
read_rnd_buffer_size = 256K
net_buffer_length = 2K
thread_stack = 128K
table_open_cache=500
datadir = "/home/pilou/Formation/mysqld_multi/data1"
pid-file = /home/pilou/Formation/mysqld_multi/data1/mysql1.pid
log-error=/home/pilou/Formation/mysqld_multi/logerror1.err
```

```
[mysqld2]
port = 3307
mysqld = /home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld
socket = /tmp/mysql.sock2
skip-external-locking
key_buffer_size = 16K
max_allowed_packet = 1M
table_open_cache = 4
sort_buffer_size = 64K
read_buffer_size = 256K
read_rnd_buffer_size = 256K
net_buffer_length = 2K
thread_stack = 128K
table_open_cache=500
datadir = "/home/pilou/Formation/mysqld_multi/data2"
pid-file = /home/pilou/Formation/mysqld_multi/data2/mysql2.pid
log-error=/home/pilou/Formation/mysqld_multi/logerror2.err
```

Le lancement se fait ainsi

```
./bin/mysqld_multi --defaults-file=/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/mysqld_multi.ini --verbose start 1
```

et l'arret

```
./bin/mysqld_multi --defaults-file=/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/mysqld_multi.ini --verbose stop
```

Tuning InnoDB

Il est complexe de faire le tuning d'InnoDB.

Une façon de faire est de laisser mySQLtuner, un outil client vérifier la configuration.

Soit un fichier my.ini de base

```
[mysqld]
basedir=/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64
datadir=/home/pilou/Formation/simpleinit/data
log-error=/home/pilou/Formation/simpleinit/mysqld.log
port = 3306
socket = /tmp/mysql.sock skip-external-locking
key_buffer_size = 16K
max_allowed_packet = 1M
table_open_cache = 4
sort_buffer_size = 64K
read_buffer_size = 256K
read_rnd_buffer_size = 256K
net_buffer_length = 2K
thread_stack = 128K
```

et exécutons mySQLtuner au regard de ce fichier

```
wget https://raw.githubusercontent.com/major/MySQLTuner-perl/master/mysqltuner.pl
chmod +x mysqltuner.pl
wget https://raw.githubusercontent.com/major/MySQLTuner-perl/master/basic_passwords.txt -O
basic_passwords.txt
wget https://raw.githubusercontent.com/major/MySQLTuner-perl/master/vulnerabilities.csv -O
vulnerabilities.csv
```

Au premier run, nous avons:

```
/mysqltuner.pl --host 127.0.0.1 --user root --pass piloupilou
>> MySQLTuner 1.7.19 - Major Hayden <major@mhtx.net>
>> Bug reports, feature requests, and downloads at http://mysqltuner.com/
```

>> Run with '--help' for additional options and output filtering

[--] Skipped version check for MySQLTuner script

[--] Performing tests on 127.0.0.1:3306

[OK] Logged in using credentials passed on the command line

[OK] Currently running supported MySQL version 8.0.20

[OK] Operating on 64-bit architecture

----- Log file Recommendations

[OK] Log file /home/pilou/Formation/simpleinit/mysqld.log exists

[--] Log file: /home/pilou/Formation/simpleinit/mysqld.log(11K)

[OK] Log file /home/pilou/Formation/simpleinit/mysqld.log is readable.

[OK] Log file /home/pilou/Formation/simpleinit/mysqld.log is not empty

[OK] Log file /home/pilou/Formation/simpleinit/mysqld.log is smaller than 32 Mb

[!!] /home/pilou/Formation/simpleinit/mysqld.log contains 8 warning(s).

[!!] /home/pilou/Formation/simpleinit/mysqld.log contains 25 error(s).

[--] 9 start(s) detected in /home/pilou/Formation/simpleinit/mysqld.log

[--] 1) 2020-05-28T20:09:32.463843Z 0 [System] [MY-010931] [Server]

/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: ready for connections.

Version: '8.0.20' socket: '/tmp/mysql.sock skip-external-locking' port: 3306 MySQL

Community Server - GPL.

[--] 2) 2020-05-28T20:09:32.188942Z 0 [System] [MY-011323] [Server] X Plugin ready for

connections. Socket: '/tmp/mysqlx.sock' bind-address: '::' port: 33060

[--] 3) 2020-05-28T20:06:01.472567Z 0 [System] [MY-010931] [Server]

/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: ready for connections.

Version: '8.0.20' socket: '/tmp/mysql.sock skip-external-locking' port: 0 MySQL Community

Server - GPL.

[--] 4) 2020-05-28T20:06:01.309759Z 0 [System] [MY-011323] [Server] X Plugin ready for

connections. Socket: '/tmp/mysqlx.sock'

[--] 5) 2020-05-28T19:56:56.960909Z 0 [System] [MY-010931] [Server]

/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: ready for connections.

Version: '8.0.20' socket: '/tmp/mysql.sock skip-external-locking' port: 3306 MySQL

Community Server - GPL.

[--] 6) 2020-05-28T19:56:56.807408Z 0 [System] [MY-011323] [Server] X Plugin ready for

connections. Socket: '/tmp/mysqlx.sock' bind-address: '::' port: 33060

[--] 7) 2020-05-28T19:56:23.191817Z 0 [System] [MY-011323] [Server] X Plugin ready for

connections. Socket: '/tmp/mysqlx.sock'

[--] 8) 2020-05-11T21:03:33.700583Z 0 [System] [MY-010931] [Server]

/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: ready for connections.

```
Version: '8.0.20' socket: '/tmp/mysql.sock' port: 3306 MySQL Community Server - GPL.
[--] 9) 2020-05-11T21:03:33.566923Z 0 [System] [MY-011323] [Server] X Plugin ready for
connections. Socket: '/tmp/mysqlx.sock' bind-address: '::' port: 33060
[--] 11 shutdown(s) detected in /home/pilou/Formation/simpleinit/mysqld.log
[--] 1) 2020-05-28T20:08:34.761588Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.
[--] 2) 2020-05-28T20:08:12.745558Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.
[--] 3) 2020-05-28T20:07:50.132900Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.
[--] 4) 2020-05-28T20:05:52.370915Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.
[--] 5) 2020-05-28T19:56:24.617889Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.
[--] 6) 2020-05-11T21:21:48.880403Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.
[--] 7) 2020-05-11T21:03:07.368654Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.
[--] 8) 2020-05-11T21:02:10.209497Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.
[--] 9) 2020-05-11T21:02:00.245998Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.
[--] 10) 2020-05-11T21:01:56.588580Z 0 [System] [MY-010910] [Server]
/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/bin/mysqld: Shutdown complete
(mysqld 8.0.20) MySQL Community Server - GPL.

----- Storage Engine Statistics
-----
[--] Status: +ARCHIVE +BLACKHOLE +CSV -FEDERATED +InnoDB +MEMORY +MRG_MYISAM +MyISAM
+PERFORMANCE_SCHEMA
[--] Data in InnoDB tables: 16.0K (Tables: 1)
```

[OK] Total fragmented tables: 0

----- Analysis Performance Metrics

[--] innodb_stats_on_metadata: OFF

[OK] No stat updates during querying INFORMATION_SCHEMA.

----- Security Recommendations

[--] Skipped due to unsupported feature for MySQL 8

----- CVE Security Recommendations

[OK] NO SECURITY CVE FOUND FOR YOUR VERSION

----- Performance Metrics

[--] Up for: 14m 50s (102 q [0.115 qps], 43 conn, TX: 254K, RX: 11K)

[--] Reads / Writes: 100% / 0%

[--] Binary logging is enabled (GTID MODE: OFF)

[--] Physical Memory : 2.9G

[--] Max MySQL memory : 452.6M

[--] Other process memory: 0B

[--] Total buffers: 160.0M global + 1.9M per thread (151 max threads)

[--] P_S Max memory usage: 72B

[--] Galera GCache Max memory usage: 0B

[OK] Maximum reached memory usage: 163.9M (5.48% of installed RAM)

[OK] Maximum possible memory usage: 452.6M (15.12% of installed RAM)

[OK] Overall possible memory usage with other process is compatible with memory available

[OK] Slow queries: 0% (0/102)

[OK] Highest usage of available connections: 1% (2/151)

[OK] Aborted connections: 0.00% (0/43)

[!!] name resolution is active : a reverse name resolution is made for each new connection and can reduce performance

[--] Query cache have been removed in MySQL 8

[OK] Sorts requiring temporary tables: 0% (0 temp sorts / 7 sorts)

[OK] No joins without indexes

[!!] Temporary tables created on disk: 52% (12 on disk / 23 total)

[OK] Thread cache hit rate: 95% (2 created / 43 connections)

[!!] Table cache hit rate: 0% (1 open / 6K opened)

[OK] table_definition_cache(402) is upper than number of tables(311)

[OK] Open file limit used: 0% (2/5K)

[OK] Table locks acquired immediately: 100% (8 immediate / 8 locks)

[OK] Binlog cache memory access: 100.00% (1 Memory / 1 Total)

----- Performance schema

[--] Memory used by P_S: 72B

--] Sys schema is installed.

----- ThreadPool Metrics

[--] ThreadPool stat is disabled.

----- MyISAM Metrics

[--] MyISAM Metrics are disabled on last MySQL versions.

----- InnoDB Metrics

[--] InnoDB is enabled.

--] InnoDB Thread Concurrency: 0

[OK] InnoDB File per table is activated

[OK] InnoDB buffer pool / data size: 128.0M/16.0K

[!!] Ratio InnoDB log file size / InnoDB Buffer pool size (75 %): 48.0M * 2/128.0M should be equal to 25%

[OK] InnoDB buffer pool instances: 1

--] Number of InnoDB Buffer Pool Chunk : 1 for 1 Buffer Pool Instance(s)

[OK] Innodb_buffer_pool_size aligned with Innodb_buffer_pool_chunk_size & Innodb_buffer_pool_instances

[OK] InnoDB Read buffer efficiency: 97.89% (40289 hits/ 41156 total)

[OK] InnoDB Write log efficiency: 96.15% (4891 hits/ 5087 total)

[OK] InnoDB log waits: 0.00% (0 waits / 196 writes)

----- AriaDB Metrics

[--] AriaDB is disabled.

----- TokuDB Metrics

[--] TokuDB is disabled.

----- XtraDB Metrics

[--] XtraDB is disabled.

----- Galera Metrics

[--] Galera is disabled.

----- Replication Metrics

[--] Galera Synchronous replication: NO

[--] No replication slave(s) for this server.

[--] Binlog format: ROW

[--] XA support enabled: ON

[--] Semi synchronous replication Master: Not Activated

[--] Semi synchronous replication Slave: Not Activated

[--] This is a standalone server

----- Recommendations

General recommendations:

Control warning line(s) into /home/pilou/Formation/simpleinit/mysqld.log file

Control error line(s) into /home/pilou/Formation/simpleinit/mysqld.log file

MySQL was started within the last 24 hours - recommendations may be inaccurate

Configure your accounts with ip or subnets only, then update your configuration with skip-name-resolve=1

When making adjustments, make tmp_table_size/max_heap_table_size equal

Reduce your SELECT DISTINCT queries which have no LIMIT clause

Increase table_open_cache gradually to avoid file descriptor limits

Read this before increasing table_open_cache over 64: <https://bit.ly/2Fulv7r>

Read this before increasing for MariaDB https://mariadb.com/kb/en/library/optimizing-table_open_cache/

This is MyISAM only table_cache scalability problem, InnoDB not affected.

See more details here: <https://bugs.mysql.com/bug.php?id=49177>

This bug already fixed in MySQL 5.7.9 and newer MySQL versions.

Beware that open_files_limit (5000) variable

should be greater than table_open_cache (4)

Before changing innodb_log_file_size and/or innodb_log_files_in_group read this:

<https://bit.ly/2TcGgtU>

Variables to adjust:

tmp_table_size (> 16M)

max_heap_table_size (> 16M)

table_open_cache (> 4)

innodb_log_file_size should be (=16M) if possible, so InnoDB total log files size equals to 25% of buffer pool size.

Ajustement 1

Il n'est pas nécessaire de charger des moteurs de stockage inutile dans la base de donnée:

```
Status: +ARCHIVE +BLACKHOLE +CSV -FEDERATED +InnoDB +MEMORY +MRG_MYISAM +MyISAM
+PERFORMANCE_SCHEMA
```

On rajoute dans le fichier my.ini

```
disabled_storage_engines=" ARCHIVE, BLACKHOLE, CSV, FEDERATED, MEMORY, MRG_MYISAM, MyISAM"
default_storage_engine=InnoDB
```

```
[mysqld]
# Required Settings
basedir=/home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64
datadir=/home/pilou/Formation/simpleinit/data
log_error=/home/pilou/Formation/simpleinit/mysqld.log
bind_address            = 127.0.0.1 # Change to 0.0.0.0 to allow remote connections
max_allowed_packet     = 256M
max_connect_errors     = 1000000
pid_file                = /tmp/mysqld.pid
port                   = 3306
skip_external_locking
skip_name_resolve
```

```

socket                                = /tmp/mysqld.sock

# Enable for b/c with databases created in older MySQL/MariaDB versions (e.g. when using null
dates)
#sql_mode                              =
ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION,ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES

tmpdir                                 = /tmp

# InnoDB Settings
default_storage_engine                 = InnoDB
innodb_buffer_pool_instances           = 2      # Use 1 instance per 1GB of InnoDB pool size
innodb_buffer_pool_size                = 2G    # Use up to 70-80% of RAM
innodb_file_per_table                  = 1
innodb_flush_log_at_trx_commit        = 0
innodb_flush_method                    = O_DIRECT
innodb_log_buffer_size                 = 16M
innodb_log_file_size                  = 512M
innodb_stats_on_metadata               = 0

#innodb_temp_data_file_path            = ibtmp1: 64M: autoextend: max: 20G # Control the maximum size for
the ibtmp1 file
#innodb_thread_concurrency             = 4      # Optional: Set to the number of CPUs on your system
(minus 1 or 2) to better
                                           # contain CPU usage. E.g. if your system has 8 CPUs,
try 6 or 7 and check
                                           # the overall load produced by MySQL/MariaDB.

innodb_read_io_threads                 = 64
innodb_write_io_threads                = 64

# MyISAM Settings

#query_cache_limit                     = 4M    # UPD - Option supported by MariaDB & up to MySQL
5.7, remove this line on MySQL 8.x
#query_cache_size                      = 64M   # UPD - Option supported by MariaDB & up to MySQL
5.7, remove this line on MySQL 8.x
#query_cache_type                      = 1     # Option supported by MariaDB & up to MySQL 5.7,
remove this line on MySQL 8.x

```

```

key_buffer_size                = 32M   # UPD

low_priority_updates           = 1
concurrent_insert              = 2

# Connection Settings
max_connections                = 100   # UPD

back_log                       = 512
thread_cache_size              = 100
thread_stack                   = 192K

interactive_timeout             = 180
wait_timeout                   = 180

# For MySQL 5.7+ only (disabled by default)
#max_execution_time            = 30000 # Set a timeout limit for SELECT statements (value in
milliseconds).

                                # This option may be useful to address aggressive
crawling on large sites,

                                # but it can also cause issues (e.g. with backups). So
use with extreme caution and test!

                                # More info at:
https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_max_execution_time

# For MariaDB 10.1.1+ only (disabled by default)
#max_statement_time            = 30   # The equivalent of "max_execution_time" in MySQL 5.7+
(set above)

                                # The variable is of type double, thus you can use
subsecond timeout.

                                # For example you can use value 0.01 for 10
milliseconds timeout.

                                # More info at: https://mariadb.com/kb/en/aborting-
statements/

# Buffer Settings
join_buffer_size               = 4M   # UPD
read_buffer_size               = 3M   # UPD
read_rnd_buffer_size           = 4M   # UPD
sort_buffer_size               = 4M   # UPD

```

```

# Table Settings
# In systemd managed systems like Ubuntu 16.04+ or CentOS 7+, you need to perform an extra
action for table_open_cache & open_files_limit
# to be overridden (also see comment next to open_files_limit).
# E.g. for MySQL 5.7, please check: https://dev.mysql.com/doc/refman/5.7/en/using-systemd.html
# and for MariaDB check: https://mariadb.com/kb/en/library/systemd/
table_definition_cache      = 40000 # UPD
table_open_cache            = 40000 # UPD
open_files_limit            = 60000 # UPD - This can be 2x to 3x the table_open_cache
value or match the system's
                                # open files limit usually set in /etc/sysctl.conf or
/etc/security/limits.conf
                                # In systemd managed systems this limit must also be
set in:
                                # /etc/systemd/system/mysqld.service.d/override.conf
(for MySQL 5.7+) and
                                # /etc/systemd/system/mariadb.service.d/override.conf
(for MariaDB)

max_heap_table_size        = 128M
tmp_table_size              = 128M

# Search Settings
ft_min_word_len            = 3      # Minimum length of words to be indexed for search
results

# Logging
log_queries_not_using_indexes = 1
long_query_time             = 5
slow_query_log              = 0      # Disabled for production
slow_query_log_file         = /home/pilou/Formation/simpleinit/mysql_slow.log

[mysqldump]
# Variable reference
# For MySQL 5.7: https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html
# For MariaDB: https://mariadb.com/kb/en/library/mysqldump/
quick
quote_names
max_allowed_packet         = 64M

```


Mysql Query Rewriter

Pour installer le plug-in de réécriture de requête Rewriter, exécutez `install_rewriter.sql` situé dans le répertoire de `share` de votre installation MySQL.

```
mysql -h localhost -u root --protocol=tcp -p < /home/pilou/Formation/mysql-8.0.20-linux-glibc2.12-x86_64/share/install_rewriter.sql
```

L'installation se constate :

```
mysql> SELECT * FROM mysql.plugin;
+-----+-----+
| name   | dl     |
+-----+-----+
| rewriter | rewriter.so |
+-----+-----+
1 row in set (0.00 sec)

mysql> SHOW GLOBAL VARIABLES LIKE 'rewriter%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| rewriter_enabled | ON   |
| rewriter_verbose | 1    |
+-----+-----+
2 rows in set (0.00 sec)
```

La table `rewrite_rules` est une table persistante pour le plugin `query_rewrite`:

```
mysql> SHOW CREATE TABLE query_rewrite.rewrite_rules\G
***** 1. row *****
      Table: rewrite_rules
Create Table: CREATE TABLE `rewrite_rules` (
```

```

`id` int NOT NULL AUTO_INCREMENT,
`pattern` varchar(5000) CHARACTER SET utf8mb4 COLLATE utf8mb4_bin NOT NULL,
`pattern_database` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_bin DEFAULT NULL,
`replacement` varchar(5000) CHARACTER SET utf8mb4 COLLATE utf8mb4_bin NOT NULL,
`enabled` enum('YES','NO') CHARACTER SET utf8mb4 COLLATE utf8mb4_bin NOT NULL DEFAULT 'YES',
`message` varchar(1000) CHARACTER SET utf8mb4 COLLATE utf8mb4_bin DEFAULT NULL,
`pattern_digest` varchar(64) DEFAULT NULL,
`normalized_pattern` varchar(100) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
1 row in set (0.01 sec)

```

Vous pouvez activer le plugin soit via le my.ini soit

```

mysql> SET GLOBAL rewriter_enabled = ON;

mysql> SET GLOBAL rewriter_enabled = OFF;

```

Le plug-in de réécriture ne fonctionne qu'avec les instructions SELECT.

Commençons avec un exemple:

```

mysql> INSERT INTO query_rewrite.rewrite_rules (pattern, replacement) VALUES('SELECT ?',
'SELECT ? + 1');

```

Il est possible de les afficher sachant qu'ils ne sont pas "compilés"

```

mysql> SELECT * FROM query_rewrite.rewrite_rules\G
***** 1. row *****
      id: 1
    pattern: SELECT ?
pattern_database: NULL
  replacement: SELECT ? + 1
      enabled: YES
      message: NULL
    pattern_digest: NULL
normalized_pattern: NULL
1 row in set (0.00 sec)

```

Puis nous allons les compiler:

```
mysql> CALL query_rewrite.flush_rewrite_rules();
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM query_rewrite_rewrite_rules\G
***** 1. row *****
      id: 1
      pattern: SELECT ?
      pattern_database: NULL
      replacement: SELECT ? + 1
      enabled: YES
      message: NULL
      pattern_digest: d1b44b0c19af710b5a679907e284acd2ddc285201794bc69a2389d77baedddae
      normalized_pattern: select ?
1 row in set (0.00 sec)
```

La requete est ainsi mise :

```
mysql> select 1;
+-----+
| 1 + 1 |
+-----+
|     2 |
+-----+
1 row in set, 1 warning (0.00 sec)
```

Usage 1: Optimization

Pour des raisons de performances, il est parfois souhaitable de réécrire une requete sans pouvoir le faire

```
-> SELECT count(distinct emp_no) FROM employees.employees INNER JOIN employees.salaries
USING(emp_no) WHERE DATEDIFF(to_date, from_date) < {integer};
```

```
<= SELECT count(emp_no) FROM employees.employees WHERE emp_no IN ( SELECT emp_no
FROM employees.salaries WHERE DATEDIFF(to_date, from_date) < {integer});
```

```

INSERT INTO query_rewrite.rewrite_rules
(
pattern,
replacement
)
VALUES
(
' SELECT count(distinct emp_no) FROM employees.employees INNER JOIN employees.salaries
USING(emp_no) WHERE DATEDIFF(to_date, from_date) < ?',
' SELECT count(emp_no) FROM employees.employees WHERE emp_no IN ( SELECT emp_no FROM
employees.salaries WHERE DATEDIFF(to_date, from_date) < ?)'
);

CALL query_rewrite.flush_rewrite_rules();

```

Usage 2: Optimization

Il est possible de rajouter des optimizations en commentaires dans le SQL de MySQL. Par exemple,

```
-> SELECT count(distinct emp_no) FROM employees.employees INNER JOIN employees.salaries
USING(emp_no) WHERE salary = {integer};
```

```
<= SELECT /*+ MAX_EXECUTION_TIME(10000)*/ count(distinct emp_no) FROM
employees.employees INNER JOIN employees.salaries USING(emp_no) WHERE salary = {integer};
```

my.ini

Le but de ce chapitre est de mettre en place un bon fichier my.ini

En premier lieu, on regarde la mémoire libre ainsi que le nombre de processeur

```
cat /proc/meminfo
MemTotal:      3064328 kB
MemFree:       697228 kB
MemAvailable:  1337368 kB

cat /proc/cpuinfo
2 processeur
```

Regardons ensuite le nombre de tables (nous prenons ici l'exemple de <https://dev.mysql.com/doc/employee/en/employees-installation.html>)

Remplissons la et faisons les requetes automatiques

```
mysql < employees.sql
mysql -t < test_employees_md5.sql
```

Configuration réseau et de base

```
bind_address          = 127.0.0.1 # mettre 0.0.0.0 pour les connexions distantes
max_allowed_packet    = 256M # taille d'un packet de donnée
max_connect_errors    = 1000000 # eviction d'un client apres max_connect_errors
skip_external_locking # pour MyISAM desaloue les lock systeme
skip_name_resolve     # pas de résolution DNS
sql_mode              = "NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES"
tmpdir                = /tmp
#user                 = mysql En cas d'installation en mode service, il faut
préciser l'utilisateur qui execute le service
```

Parametrage du nombre de connexion:

Cela est issue au runtime de :

```
show status like '%connected%'
```

et faire

```
max_connections          = 151                # nombre de connexion
maximal
max_user_connections    = 145                # nombre de connexion
pour un utilisateur
thread_cache_size       = 151                # =max_connexions
```

Variable de sessions

```
sort_buffer_size        = 2M                # Could be too big for
many small sorts
tmp_table_size          = 32M                # taille des table
temporaire

read_buffer_size        = 128k               # a verifier avec le
calcul mémoire
read_rnd_buffer_size    = 256k             # a verifier avec le
calcul mémoire
join_buffer_size        = 128k             # a verifier avec le
calcul mémoire

# Other buffers and caches

table_definition_cache   = 1400            # nombre de table dans la
base de donnée au max au min show global status like 'open_tables';
table_open_cache         = 2000            # approximativement show
global status like 'opened_tables'; / max_connection
table_open_cache_instances = 16            # New default in 5.7
```

Parametrage de InnoDB

```
default_storage_engine   = InnoDB # le moteur de stockage est InnoDB
innodb_buffer_pool_instances = 1      # 1 instance par Giga, donc ici 1 pour buffer pool
chunk size de 1G
innodb_buffer_pool_size  = 410M      # 80% de la RAM
innodb_file_per_table    = 1
innodb_flush_log_at_trx_commit = 0
```

```

innodb_flush_method          = 0_DIRECT
innodb_file_format           = Barracuda # format du fichier interne
innodb_log_buffer_size       = 5M # entre 5 et 10% du log file size
innodb_log_file_size         = 64M # 25% du buffer pool
innodb_stats_on_metadata     = 0
#innodb_temp_data_file_path  = ibtmp1: 64M: autoextend: max: 20G # Control the maximum size for
the ibtmp1 file
innodb_thread_concurrency    = 4      # nombre de CPU -1/2
innodb_read_io_threads       = 8
innodb_write_io_threads      = 8

```

Les deux derniers paramètres sont issue de SHOW ENGINE INNODB STATUS et en particulier du nombre de requete pending qui doivent être inférieur a $pending * 64 < io_thread$

Parametrage de MyISAM

```

key_buffer_size              = 8M                # 25% de la RAM
mysam_recover_options        = 'BACKUP, FORCE'

```

Et enfin les logs:

```

log_warnings                 = 2                # MySQL 5.6, equivalent à
log_error_verbosity = 3
# log_error_verbosity        = 3                # MySQL 5.7, equivalent
) log_warnings = 2, a supprimer pour mariadb
innodb_print_all_deadlocks   = 1

# Slow Query Log

slow_query_log_file          = %INSTANCEDIR%/log/%UNAME%_%INSTANCE%_slow.log
slow_query_log               = 0
log_queries_not_using_indexes = 0                # pour les developpeurs
long_query_time              = 0.5
min_examined_row_limit      = 100

# Replication

#server_id                  = %SERVERID%          # id
#log_bin                    = %INSTANCEDIR%/binlog/%UNAME%_%INSTANCE%_binlog
#emplacement

```

```

# master_verify_checksum      = ON                # MySQL 5.6
#binlog_cache_size            = 1M
#binlog_stmt_cache_size       = 1M
#max_binlog_size              = 128M              # en fonction du trafic
#sync_binlog                  = 1                # Mettre a 0 pour avoir
des problemes
#expire_logs_days             = 5                # We will survive easter
holidays
#binlog_format                 = ROW              # ROW est ok pour la
replication
# binlog_row_image            = MINIMAL          # Since 5.6
# auto_increment_increment    = 2              # Pour Master/Master
mettre 2
# auto_increment_offset       = 1              # Pour Master/Master
mettre 1 et 2

```

Calcul de la mémoire

Utiliser <https://www.mysqlcalculator.com/> qui fait le calcul de la mémoire du serveur:

```

key_buffer_size
+ query_cache_size
+ tmp_table_size
+ innodb_buffer_pool_size
+ innodb_additional_mem_pool_size
+ innodb_log_buffer_size
+ max_connections
×
(sort_buffer_size
+ read_buffer_size
+ read_rnd_buffer_size
+ join_buffer_size
+ thread_stack
+ binlog_cache_size )

```

On remarquera le nombre de connexions maximums influe énormément sur la mémoire

Replication

Nous allons mettre en place une replication asynchrone.

Variables

innodb_flush_log_at_trx_commit

innodb_flush_log_at_trx_commit contrôle l'équilibre entre la conformité ACID stricte pour les opérations de validation et les performances supérieures possibles lorsque les opérations d'E / S associées à la validation sont réorganisées et effectuées par lots. Vous pouvez obtenir de meilleures performances en modifiant la valeur par défaut, mais vous pouvez ensuite perdre des transactions en cas de blocage. Le paramètre par défaut de 1 est requis pour la conformité ACID complète. Les journaux sont écrits et vidés sur le disque à chaque validation de transaction. Avec un réglage de 0, les journaux sont écrits et vidés sur le disque une fois par seconde. Les transactions pour lesquelles les journaux n'ont pas été vidés peuvent être perdues dans un crash. Avec un réglage de 2, les journaux sont écrits après chaque validation de transaction et vidés les transactions sur le disque une fois par seconde. Les transactions pour lesquelles les journaux n'ont pas été vidés peuvent être perdues dans un crash. Pour les réglages 0 et 2, la propriété ACID n'est pas garantie

sync_binlog

sync_binlog contrôle la fréquence à laquelle le serveur MySQL synchronise le journal binaire sur le disque.

- sync_binlog = 0: désactive la synchronisation du journal binaire sur le disque par le serveur MySQL. À la place, le serveur MySQL s'appuie sur le système d'exploitation pour vider le journal binaire sur le disque de temps en temps, comme il le fait pour tout autre fichier. Ce paramètre offre les meilleures performances, mais en cas de panne d'alimentation ou de panne du système d'exploitation, il est possible que le serveur ait validé des transactions qui n'ont pas été synchronisées avec le journal binaire.
- sync_binlog = 1: active la synchronisation du journal binaire sur le disque avant que les transactions ne soient validées. Ce paramètre est le plus sûr, mais peut avoir un impact négatif sur les performances en raison du nombre accru d'écritures sur disque. En cas de panne d'alimentation ou de panne du système d'exploitation, les transactions manquantes dans le journal binaire sont uniquement dans un état préparé. Cela permet à

la routine de récupération automatique d'annuler les transactions, ce qui garantit qu'aucune transaction n'est perdue à partir du journal binaire.

- `sync_binlog = N`, où N est une valeur autre que 0 ou 1: le journal binaire est synchronisé sur le disque après la collecte de N groupes de validation de journal binaire. En cas de panne d'électricité ou de panne du système d'exploitation, il est possible que le serveur ait validé des transactions qui n'ont pas été vidées dans le journal binaire. Ce paramètre peut avoir un impact négatif sur les performances en raison du nombre accru d'écritures sur disque. Une valeur plus élevée améliore les performances, mais avec un risque accru de perte de données.

binlog-format

`binlog-format` contrôle la façon dont le format de log par statement, par données ou les deux. (ROW,STATEMENT,MIXED)

- Lors de l'utilisation de la journalisation binaire basée sur des instructions, le maître écrit des instructions SQL dans le journal binaire. La réplication du maître sur l'esclave fonctionne en exécutant les instructions SQL sur l'esclave. C'est ce qu'on appelle la réplication basée sur les instructions (qui peut être abrégée en SBR), ce qui correspond au format de journalisation binaire basé sur les instructions MySQL.
- Lors de l'utilisation de la journalisation basée sur les lignes, le maître écrit dans le journal binaire des événements indiquant comment les rangées individuelles d'une table sont modifiées. La réplication du maître sur l'esclave fonctionne en copiant les événements représentant les modifications apportées aux lignes du tableau vers l'esclave. C'est ce qu'on appelle la réplication basée sur les lignes (qui peut être abrégée en RBR).

log_slave_updates

Normalement, un esclave n'écrit aucune mise à jour reçue d'un serveur maître dans son propre journal binaire. Cette option amène l'esclave à écrire les mises à jour effectuées par son thread SQL dans son propre journal binaire.

master-info-file

Nom à utiliser pour le fichier dans lequel l'esclave enregistre des informations sur le maître

- La journalisation basée sur les lignes est la méthode par défaut.
- Vous pouvez également configurer MySQL pour utiliser une combinaison de consignation basée sur les instructions et basée sur les lignes, en fonction de l'option la plus appropriée pour la journalisation des modifications. C'est ce qu'on appelle la journalisation mixte. Lorsque vous utilisez une journalisation mixte, un journal basé sur des instructions est

utilisé par défaut. En fonction de certaines instructions, ainsi que du moteur de stockage utilisé, le journal est automatiquement basculé en cas de modification basée sur les lignes.

Configuration replication asynchrone

Dans cette replication, mysqld1 va servir de maître et mysqld2 sera un esclave. Dans le groupe mysqld1, nous allons associer un id de serveur ainsi que les bonnes valeurs de réplication

Configuration du master:

```
[mysqld]
port = 3306
server-id = 1
socket = /tmp/mysql.sock1
skip-external-locking
key_buffer_size = 16K
max_allowed_packet = 1M
table_open_cache = 4
sort_buffer_size = 64K
read_buffer_size = 256K
read_rnd_buffer_size = 256K
net_buffer_length = 2K
thread_stack = 128K
table_open_cache=500
datadir = /home/pilou/Formation/replication/master-slave/master/data
pid-file = /home/pilou/Formation/replication/master-slave/master/mysqlmaster.pid
log-error=/home/pilou/Formation/replication/master-slave/master/logerror.err
expire_logs_days = 10
max_binlog_size = 100M
innodb_flush_log_at_trx_commit = 1
sync_binlog = 1
binlog-format = ROW
```

et dans le serveur esclave (en readonly) nous associons <source lang='ini'>

```
datadir = /home/pilou/mysql80/datadir/mysql2
pid-file = /home/pilou/mysql80/datadir/mysql2/mysql2.pid
server-id = 2
log-bin = /home/pilou/mysql80/datadir/mysql2/mysql-bin.log
log-error = /home/pilou/mysql80/datadir/mysql2/error_slave.log
relay-log = /home/pilou/mysql80/datadir/mysql2/relay-bin
relay-log-index = /home/pilou/mysql80/datadir/mysql2/relay-bin.index
master-info-file = /home/pilou/mysql80/datadir/mysql2/master.info
relay-log-info-file = /home/pilou/mysql80/datadir/mysql2/relay-log.info
read_only = 1
```

Création d'un utilisateur de replication.

Il est nécessaire d'avoir une connectivité du slave vers le master pour faire la replication.

Sur le master faire:

```
CREATE USER 'slave'@'%' WITH mysql_native_password IDENTIFIED BY 'password';
GRANT REPLICATION SLAVE ON *.* TO 'slave'@'%';
```

Puis regarder la position dans les journaux

```
mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+-----+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| binlog.000004 |      690 |              |                  |                   |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Une autre solution est de faire un rattrapage avec position des journaux:

Nous allons faire un dump de la base du master via mysqldump

```
./bin/mysqldump -u root -p --host=127.0.0.1 --port=3306 --all-databases --master-data=2 > replicationdump.sql
```

L'utilisation du flag master-data permet d'inclure les information du master, et en particulier la position des log dans le fichier de log

```

- MySQL dump 10.13 Distrib 8.0.13, for linux-glibc2.12 (x86_64) --

-- Host: 127.0.0.1 Database: -- -----

-- Server version 8.0.13

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!50606 SET @OLD_INNODB_STATS_AUTO_RECALC=@@INNODB_STATS_AUTO_RECALC */;

/*!50606 SET GLOBAL INNODB_STATS_AUTO_RECALC=OFF */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */; /*!40014 SET
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;

/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

-- -- Position to start replication or point-in-time recovery from --

-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000004', MASTER_LOG_POS=690;

-- -- Current Database: `mysql` --

```

Sur l'esclave, connecter reinjecter les données:

```
mysql -uroot -p --host=127.0.0.1 --port=3307 < replicationdump.sql
```

Puis connecter vous sur l'esclave CHANGE MASTER TO MASTER_HOST='127.0.0.1', MASTER_USER='slave', MASTER_PASSWORD='password', MASTER_LOG_FILE='binlog.000004', MASTER_LOG_POS=973;

Mise en place de la replication

Faire un start slave et vérifier le statut de l'esclave

```
mysql> start slave;
Query OK, 0 rows affected (0.01 sec)

mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
        Master_Host: 127.0.0.1
        Master_User: slave
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: binlog.000004
  Read_Master_Log_Pos: 973
        Relay_Log_File: relay-bin.000002
        Relay_Log_Pos: 321
  Relay_Master_Log_File: binlog.000004
  Slave_IO_Running: Yes
  Slave_SQL_Running: Yes
  Replicate_Do_DB:
  Replicate_Ignore_DB:
  Replicate_Do_Table:
  Replicate_Ignore_Table:
  Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
          Last_Errno: 0
          Last_Error:
        Skip_Counter: 0
  Exec_Master_Log_Pos: 973
        Relay_Log_Space: 524
        Until_Condition: None
        Until_Log_File:
        Until_Log_Pos: 0
  Master_SSL_Allowed: No
  Master_SSL_CA_File:
  Master_SSL_CA_Path:
  Master_SSL_Cert:
```

```
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_UUID: 631f7e58-9421-11ea-b477-080027193107
Master_Info_File: mysql.slave_master_info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
Master_Retry_Count: 86400
Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
Master_public_key_path:
Get_master_public_key: 0
Network_Namespace:
1 row in set (0.00 sec)
```

Configuration replication semi-synchrone

Sur le master il faut:

- Activer le plugin de replication semi synchrone `INSTALL PLUGIN rpl_semi_sync_master SONAME 'semisync_master.so';`
- Modifier le fichier de configuration en rajoutant `rpl_semi_sync_master_enabled = 1`

Une fois le redemarrage du serveur on observe:

```
mysql> show status like "rpl_semi_sync%";
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| Rpl_semi_sync_master_clients      | 0     |
| Rpl_semi_sync_master_net_avg_wait_time | 0     |
| Rpl_semi_sync_master_net_wait_time  | 0     |
| Rpl_semi_sync_master_net_waits     | 0     |
| Rpl_semi_sync_master_no_times     | 0     |
| Rpl_semi_sync_master_no_tx        | 0     |
| Rpl_semi_sync_master_status       | ON    |
| Rpl_semi_sync_master_timefunc_failures | 0     |
| Rpl_semi_sync_master_tx_avg_wait_time | 0     |
| Rpl_semi_sync_master_tx_wait_time  | 0     |
| Rpl_semi_sync_master_tx_waits     | 0     |
| Rpl_semi_sync_master_wait_pos_backtraverse | 0     |
| Rpl_semi_sync_master_wait_sessions  | 0     |
| Rpl_semi_sync_master_yes_tx       | 0     |
+-----+-----+
14 rows in set (0.00 sec)
```

La replication est pour l'instant asynchrone car aucun client semi synchrone n'est connecté

```
mysql> SHOW VARIABLES LIKE "rpl_semi_sync_master_timeout";
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| rpl_semi_sync_master_timeout | 10000 |
+-----+-----+
1 row in set (0.02 sec)
```

Sur le slave

Installer le plugin tel que sur le master mais dans sa version salve

```
INSTALL PLUGIN rpl_semi_sync_slave SONAME 'semisync_slave.so';
```

Puis dans le fichier my.ini

```
rpl_semi_sync_slave_enabled = 1
```

Enfin il faut vérifier le statut de la réplication:

```
mysql> SHOW VARIABLES LIKE "Rpl_semi_sync_slave_%";
+-----+
| Variable_name          | Value |
+-----+
| rpl_semi_sync_slave_enabled | ON    |
| rpl_semi_sync_slave_trace_level | 32    |
+-----+
2 rows in set (0.01 sec)
```

Sur le master, il est maintenant indiqué qu'il y a un client:

```
mysql> show status like "rpl_semi_sync%";
+-----+
| Variable_name          | Value |
+-----+
| Rpl_semi_sync_master_clients | 1     |
| Rpl_semi_sync_master_net_avg_wait_time | 0     |
| Rpl_semi_sync_master_net_wait_time | 0     |
| Rpl_semi_sync_master_net_waits | 0     |
| Rpl_semi_sync_master_no_times | 0     |
| Rpl_semi_sync_master_no_tx | 0     |
| Rpl_semi_sync_master_status | ON    |
| Rpl_semi_sync_master_timefunc_failures | 0     |
| Rpl_semi_sync_master_tx_avg_wait_time | 0     |
| Rpl_semi_sync_master_tx_wait_time | 0     |
| Rpl_semi_sync_master_tx_waits | 0     |
| Rpl_semi_sync_master_wait_pos_backtraverse | 0     |
| Rpl_semi_sync_master_wait_sessions | 0     |
| Rpl_semi_sync_master_yes_tx | 0     |
+-----+
14 rows in set (0.01 sec)
```

Configuration GTID

Sur le master

```
server-id = 1
log-bin = mysql-bin
binlog_format = row
gtid-mode=ON
enforce-gtid-consistency
log-slave-updates
```

Sur le slave

```
server-id = 2
log-bin = mysql-bin
relay-log = relay-log-server
relay-log = relay-log-server
read-only = ON
gtid-mode=ON
enforce-gtid-consistency
log-slave-updates
```

Puis

```
CHANGE MASTER TO
MASTER_HOST = '54.89.xx.xx',
MASTER_PORT = 3306,
MASTER_USER = 'repl_user',
MASTER_PASSWORD = 'XXXXXXXXX',
MASTER_AUTO_POSITION = 1;
```

SSL et Utilisateurs

SSL

maximum de connection et maximum de connection utilisateur

- `max_user_connections` : Le nombre maximum de connexions simultanées autorisées sur un compte utilisateur MySQL donné. Une valeur de 0 (valeur par défaut) signifie «aucune limite». Cette variable a une valeur globale qui peut être définie au démarrage ou à l'exécution du serveur. Il a également une valeur de session en lecture seule qui indique la limite effective de connexion simultanée qui s'applique au compte associé à la session en cours.
- `max_connections` Le nombre maximum autorisé de connexions client simultanées

Par défaut dans l'installation :

15

1

```
mysql> select @@max_user_connections ;
```

2

```
+-----+
```

3

```
| @@max_user_connections |
```

4

```
+-----+
```

5

```
| 0 |
```

6

```
+-----+
```

7

```
1 row in set (0,00 sec)
```

8

```
□
```

9

```
mysql> select @@max_connections ;
```

10

```
+-----+
```

11

```
| @@max_connections |
```

12

```
+-----+
```

13

```
| 151 |
```

14

```
+-----+
```

15

1 row in set (0,00 sec)

Un bon conseil est de fixer `max_user_connections` à 50 à 75% de vos paramètres `max_connections`. Vous définissez cette valeur dans la section `mysqld` de votre `my.cnf`:

2

1

```
max_connections = 400
```

2

```
max_user_connections=200
```

maximum de connection pour un utilisateur

Le settings précédent concerne une mise en place assez globale du nombre de connection.

Il est possible de signifier des limits plus fine en terme de temps et de ressources

Il existe différents types de limites pouvant être utilisés:

- `MAX_QUERIES_PER_HOUR` Limite le compte à X requêtes par heure.
- `MAX_UPDATES_PER_HOUR` Limite le compte à X relevés UPDATE par heure.
- `MAX_CONNECTIONS_PER_HOUR` Limite le compte à un total de X connexions par heure.
- `MAX_USER_CONNECTIONS` Limite le compte à un total de X connexions simultanées pour le compte.

Par exemple, on limite le nombre de connection de myuser à 5

2

1

```
mysql> ALTER USER 'myuser'@'localhost' WITH MAX_USER_CONNECTIONS 5;
```

2

```
□
```

LOCK et Unlock Account

Account lock et Account unlock permette de verrouiller ou pas un utilisateur

9

1

```
ALTER USER 'myuser'@'localhost' ACCOUNT LOCK;
```

2

```
Query OK, 0 rows affected (0,09 sec)
```

3

```
□
```

4

```
mysql> \q
```

5

```
Bye
```

6

```
pilou@ubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64$ ./bin/mysql -u myuser -h localh
```

7

```
Enter password:
```

8

```
ERROR 3118 (HY000): Access denied for user 'myuser'@'localhost'. Account is locked.
```

9

```
□
```

Mise en place de SSL

Pour l'instant la connection entre le client et le serveur est faite en claire.

23

1

```
status
```

2

```
-----
```

3

```
./bin/mysql Ver 8.0.13 for linux-glibc2.12 on x86_64 (MySQL Community Server - GPL)
```

4

```
□
```

5

```
Connection id: 10
```

6

```
Current database:
```

7

```
Current user: root@localhost
```

8

```
SSL: Not in use
```

9

```
Current pager: stdout
```

10

```
Using outfile: ''
```

11

```
Using delimiter: ;
```

12

```
Server version: 8.0.13 MySQL Community Server - GPL
```

13

```
Protocol version: 10
```

14

Connection: Localhost via UNIX socket

15

Server characterset: utf8mb4

16

Db characterset: utf8mb4

17

Client characterset: utf8mb4

18

Conn. characterset: utf8mb4

19

UNIX socket: /tmp/mysql.sock

20

Uptime: 29 min 39 sec

21

□

22

Threads: 2 Questions: 22 Slow queries: 0 Opens: 136 Flush tables: 2 Open tables: 106 Queries

23

Création de l'autorité de certification

Exécutez les commandes suivantes pour créer les clés de l'autorité de certification (CA):

7

1

```
pilou@ubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64$ mkdir ssl_keys
```

2

```
pilou@ubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64$ openssl genrsa 2048 > ./ssl_key
```

3

```
Generating RSA private key, 2048 bit long modulus (2 primes)
```

4

```
.....+++++
```

5

```
.....+++++
```

6

```
e is 65537 (0x010001)
```

7

```
pilou@ubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64$ openssl req -sha1 -new -x509 -r
```

Création de la clef serveur et du certificat serveur

Exécutez les commandes suivantes pour créer la clé SSL et le certificat du serveur:

3

1

```
openssl req -sha1 -newkey rsa:2048 -days 3650 -nodes -keyout ./ssl_keys/server-key.pem > ./s
```

2

```
openssl x509 -sha1 -req -in ./ssl_keys/server-req.pem -days 3650 -CA ./ssl_keys/ca-cert.pem
```

3

```
openssl rsa -in ./ssl_keys/server-key.pem -out ./ssl_keys/server-key.pem
```

Création de la clef serveur et du certificat client

Exécutez les commandes suivantes pour créer la clé SSL et le certificat du client:

3

1

```
openssl req -sha1 -newkey rsa:2048 -days 3650 -nodes -keyout ./ssl_keys/client-key.pem > ./s
```

2

```
openssl x509 -sha1 -req -in ./ssl_keys/client-req.pem -days 3650 -CA ./ssl_keys/ca-cert.pem
```

3

```
openssl rsa -in ./ssl_keys/client-key.pem -out ./ssl_keys/client-key.pem
```

Sortie de OpenSSL

Pour avoir de bon certifiact, il est important de selectionner des CN différents pour les CA, server et client

77

1

```
pilou@ubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64$ ./ssl.sh
```

2

```
Generating RSA private key, 2048 bit long modulus (2 primes)
```

3

```
.....+++++
```

4

```
.....+++++
```

5

```
e is 65537 (0x010001)
```

6

```
You are about to be asked to enter information that will be incorporated
```

7

```
into your certificate request.
```

8

```
What you are about to enter is what is called a Distinguished Name or a DN.
```

9

```
There are quite a few fields but you can leave some blank
```

10

```
For some fields there will be a default value,
```

11

```
If you enter '.', the field will be left blank.
```

12

```
-----
```

13

```
Country Name (2 letter code) [AU]:
```

14

State or Province Name (full name) [Some-State]:

15

Locality Name (eg, city) []:

16

Organization Name (eg, company) [Internet Widgits Pty Ltd]:

17

Organizational Unit Name (eg, section) []:

18

Common Name (e.g. server FQDN or YOUR name) []: CA

19

Email Address []:

20

Ignoring -days; not generating a certificate

21

Generating a RSA private key

22

.....+++++

23

.....+++++

24

writing new private key to '/home/pilou/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/ssl_keys

25

26

You are about to be asked to enter information that will be incorporated

27

into your certificate request.

28

What you are about to enter is what is called a Distinguished Name or a DN.

29

There are quite a few fields but you can leave some blank

30

For some fields there will be a default value,

31

If you enter '.', the field will be left blank.

32

33

Country Name (2 letter code) [AU]:

34

State or Province Name (full name) [Some-State]:

35

Locality Name (eg, city) []:

36

Organization Name (eg, company) [Internet Widgits Pty Ltd]:

37

Organizational Unit Name (eg, section) []:

38

Common Name (e.g. server FQDN or YOUR name) []: server

39

Email Address []:

40

□

41

Please enter the following 'extra' attributes

42

to be sent with your certificate request

43

A challenge password []:

44

An optional company name []:

45

Signature ok

46

subject=C = AU, ST = Some-State, O = Internet Widgits Pty Ltd, CN = server

47

Getting CA Private Key

48

writing RSA key

49

Ignoring -days; not generating a certificate

50

Generating a RSA private key

51

.....+++++

52

.....+++++

53

writing new private key to '/home/pilou/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/ssl_keys

54

55

You are about to be asked to enter information that will be incorporated

56

into your certificate request.

57

What you are about to enter is what is called a Distinguished Name or a DN.

58

There are quite a few fields but you can leave some blank

59

For some fields there will be a default value,

60

If you enter '.', the field will be left blank.

61

62

Country Name (2 letter code) [AU]:

63

State or Province Name (full name) [Some-State]:

64

Locality Name (eg, city) []:

65

Organization Name (eg, company) [Internet Widgits Pty Ltd]:

66

Organizational Unit Name (eg, section) []:

67

Common Name (e.g. server FQDN or YOUR name) []:client

68

Email Address []:

69

□

70

Please enter the following 'extra' attributes

71

to be sent with your certificate request

72

A challenge password []:

73

An optional company name []:

74

Signature ok

75

subject=C = AU, ST = Some-State, O = Internet Widgits Pty Ltd, CN = client

76

Getting CA Private Key

77

writing RSA key

Modification de MySQL

Il faut indiquer à MySQL où se trouvent les différentes clés et certificats:

24

1

[mysqld]

2

port = 3306

3

socket = /tmp/mysql.sock

4

skip-external-locking

5

key_buffer_size = 16K

6

max_allowed_packet = 1M

7

table_open_cache = 4

8

sort_buffer_size = 64K

9

read_buffer_size = 256K

10

read_rnd_buffer_size = 256K

11

net_buffer_length = 2K

12

thread_stack = 128K

13

table_open_cache=500

14

secure_file_priv=/tmp

15

```
max_connections = 400
```

16

```
max_user_connections=200
```

17

```
ssl-ca=/home/pilou/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/ssl_keys/ca-cert.pem
```

18

```
ssl-cert=/home/pilou/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/ssl_keys/server-cert.pem
```

19

```
ssl-key=/home/pilou/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/ssl_keys/server-key.pem
```

20

```
ssl-cipher=DHE-RSA-AES256-SHA
```

21

```
[]
```

22

```
[client]
```

23

```
ssl-cert=/home/pilou/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/ssl_keys/client-cert.pem
```

24

```
ssl-key=/home/pilou/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/ssl_keys/client-key.pem
```

Après redémarrage, le serveur signale que le certificat est auto signé

2

1

```
2019-01-05T08:44:05.815408Z 0 [Warning] [MY-010068] [Server] CA certificate
```

2

```
/home/pilou/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64/ssl_keys/ca-cert.pem is self signed.
```

et les parametre SSL sont bien chargé

Test

Nous allons nous connecter en SSL sur le serveur en demandant explicitement a utiliser la connection TCP (ce qui force l'utilisation de SSL)

34

1

```
pilou@ubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64$ ./bin/mysql --defaults-file=hc
```

2

```
Enter password:
```

3

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

4

```
Your MySQL connection id is 8
```

5

```
Server version: 8.0.13 MySQL Community Server - GPL
```

6

```
□
```

7

```
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
```

8

```
□
```

9

```
Oracle is a registered trademark of Oracle Corporation and/or its
```

10

```
affiliates. Other names may be trademarks of their respective
```

11

```
owners.
```

12

```
□
```

13

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

14

□

15

```
mysql> status
```

16

```
-----
```

17

```
./bin/mysql Ver 8.0.13 for linux-glibc2.12 on x86_64 (MySQL Community Server - GPL)
```

18

□

19

```
Connection id: 8
```

20

```
Current database:
```

21

```
Current user: root@localhost
```

22

```
SSL: Cipher in use is DHE-RSA-AES256-SHA
```

23

```
Current pager: stdout
```

24

```
Using outfile: ''
```

25

```
Using delimiter: ;
```

26

```
Server version: 8.0.13 MySQL Community Server - GPL
```

27

```
Protocol version: 10
```

28

```
Connection: localhost via TCP/IP
```

29

Server charsetset: utf8mb4

30

Db charsetset: utf8mb4

31

Client charsetset: utf8mb4

32

Conn. charsetset: utf8mb4

33

TCP port: 3306

34

Uptime: 20 sec

Installation de SSL pour MariaDB

Créez un répertoire nommé ssl dans le répertoire /etc/mysql/

3

1

```
$ cd /etc/mysql
```

2

```
$ sudo mkdir ssl
```

3

```
$ cd ssl
```

4

1

La valeur du nom commun utilisée pour les certificats/clés du serveur et du client doit être

2

Nom commun de l'AC : administrateur MariaDB

3

Nom commun du serveur : serveur MariaDB

4

Nom commun du client : client MariaDB

Tapez la commande suivante pour créer une nouvelle autorité de certification:

Utilisateur et Role

MySQL implémente un système sophistiqué de contrôle d'accès et de privilèges qui vous permet de créer des règles d'accès complètes pour la gestion des opérations client et d'empêcher efficacement les clients non autorisés d'accéder au système de base de données.

Le contrôle d'accès MySQL comporte deux étapes lorsqu'un client se connecte au serveur:

- Vérification de la connexion: un client qui se connecte au serveur de base de données MySQL doit avoir un nom d'utilisateur et un mot de passe valides. De plus, l'hôte à partir duquel le client se connecte doit correspondre à l'hôte dans la table de droits MySQL.
- Vérification de la demande: une fois la connexion établie avec succès, MySQL vérifie, pour chaque instruction émise par le client, si le client dispose des privilèges suffisants pour exécuter cette instruction. MySQL peut vérifier un privilège au niveau de la base de données, de la table et du champ.

Création de comptes d'utilisateurs à l'aide de l'instruction MySQL CREATE USER

MySQL fournit l'instruction CREATE USER qui vous permet de créer un nouveau compte utilisateur. La syntaxe de l'instruction CREATE USER est la suivante:

```
CREATE USER "compte_utilisateur" IDENTIFIED BY "mot de passe";
```

Le compte utilisateur au format 'nom_utilisateur' @ 'nom_hôte' est suivi de la clause CREATE USER.

Le mot de passe est spécifié dans la clause IDENTIFIED BY. Le mot de passe doit être en texte clair. MySQL chiffrera le mot de passe avant de sauvegarder le compte utilisateur dans la table user.

```
CREATE USER myuser@localhost IDENTIFIED BY 'myuser';  
Query OK, 0 rows affected (0,06 sec)
```

Puis la connexion peut s'établir ainsi:

```
./bin/mysql -u myuser -h localhost -p
```

L'utilisateur peut se connecter mais n'a pour l'instant aucun droit

```
show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
+-----+  
1 row in set (0,00 sec)
```

Les droits de notre utilisateurs sont ainsi:

```
SHOW GRANTS FOR 'myuser'@'localhost';  
+-----+  
| Grants for myuser@localhost |  
+-----+  
| GRANT USAGE ON *.* TO `myuser`@`localhost` |  
+-----+  
1 row in set (0,00 sec)
```

creation d'un utilisateur avec un plugin d'authentification

Il est possible d'utiliser un plugin d'authentification spécifique ici un plugin sha256

```
CREATE USER 'sha256user'@'localhost'  
-> IDENTIFIED WITH sha256_password BY 'password';
```

Augmentation de la sécurité via le plugin de validation de password

Le plugin `validate_password` sert à tester les mots de passe et à améliorer la sécurité. Le plugin expose un ensemble de variables système qui vous permettent de définir une politique de mot de passe.

```
mysql> INSTALL PLUGIN validate_password SONAME 'validate_password.so';  
Query OK, 0 rows affected, 1 warning (0,07 sec)  
  
mysql> SELECT PLUGIN_NAME, PLUGIN_STATUS  
-> FROM INFORMATION_SCHEMA.PLUGINS  
-> WHERE PLUGIN_NAME LIKE 'validate%';  
+-----+-----+  
| PLUGIN_NAME | PLUGIN_STATUS |  
+-----+-----+  
| validate_password | ACTIVE |  
+-----+-----+  
1 row in set (0,00 sec)
```

On teste maintenant la création d'un mot de passe pauvre

```
create user 'bob'@'%' IDENTIFIED BY 'test';  
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
```

Il est possible de charger une stratégie de validation:

- La stratégie `LOW` teste uniquement la longueur du mot de passe. Les mots de passe doivent comporter au moins 8 caractères. Pour changer cette longueur, modifiez `validate_password_length`.

- La stratégie MEDIUM ajoute les conditions selon lesquelles les mots de passe doivent contenir au moins un caractère numérique, un caractère minuscule, un caractère majuscule et un caractère spécial (non alphanumérique). Pour modifier ces valeurs, modifiez `validate_password_number_count`, `validate_password_mixed_case_count` et `validate_password_special_char_count`.
- La stratégie STRONG ajoute la condition selon laquelle les sous-chaînes de mot de passe de longueur égale ou supérieure à 4 ne doivent pas correspondre aux mots du fichier de dictionnaire, s'il en a été spécifié. Pour spécifier le fichier de dictionnaire, modifiez `validate_password_dictionary_file`.

Ici on selectionne la stratégie medium par défaut

```
SET GLOBAL validate_password_policy = 1;
Query OK, 0 rows affected (0,00 sec)

mysql> create user 'bob'@'%' identified by 'aA!12345678';
Query OK, 0 rows affected (0,05 sec)
```

Assignment d'un droit a un utilisateur

La commande GRANT permet d'associer un droit à un utilisateur. Nous donnons ici le droit show databases à l'utilisateur myuser

```
mysql> grant show databases on *.* TO 'myuser'@'localhost';
Query OK, 0 rows affected (0,13 sec)

mysql> \q
Bye
```

Nous pouvons maintenant le tester

```
pilou@lubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64$ ./bin/mysql -u myuser -h
localhost -p
mysql> show databases;
+-----+
| Database |
```

```
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| testUtf8 |
| testdb |
+-----+
6 rows in set (0,00 sec)
```

Mise en place de Rôle

En règle générale, vous avez plusieurs utilisateurs avec le même ensemble de privilèges. Le moyen d'octroyer et de révoquer des privilèges à plusieurs utilisateurs de modifier les privilèges de chaque utilisateur individuellement demande beaucoup de temps.

Pour faciliter les choses, MySQL a fourni un nouvel objet appelé rôle, qui est une collection nommée de privilèges.

Si vous souhaitez accorder le même ensemble de privilèges à plusieurs utilisateurs, procédez comme suit:

- Tout d'abord, créez un nouveau rôle.
- Deuxièmement, accordez des privilèges au rôle.
- Troisièmement, accordez le rôle aux utilisateurs.

Si vous souhaitez modifier les privilèges des utilisateurs, vous devez modifier uniquement les privilèges du rôle attribué. Les modifications prendront effet pour tous les utilisateurs auxquels le rôle a été attribué.

Nous allons créer une base de données exemple:

```
CREATE DATABASE crm;
Query OK, 1 row affected (0,05 sec)

mysql> use crm;
Database changed
mysql> CREATE TABLE customer(
```

```
-> id INT PRIMARY KEY AUTO_INCREMENT,  
-> first_name varchar(255) NOT NULL,  
-> last_name VARCHAR(255) NOT NULL,  
-> phone VARCHAR(15) NOT NULL,  
-> email VARCHAR(255)  
-> );  
Query OK, 0 rows affected (0,11 sec)  
  
mysql> INSERT INTO customer(first_name, last_name, phone, email)  
-> VALUES(' John', ' Doe', '( 408)-987-7654', ' john.doe@mysql.org' ),  
-> (' Lily', ' Bush', '( 408)-987-7985', ' lily.bush@mysql.org' );  
Query OK, 2 rows affected (0,06 sec)  
Records: 2 Duplicates: 0 Warnings: 0
```

Puis nous créons 3 rôle permettant respectivement d'être un super utilisateur de la base, de pouvoir la lire et de pouvoir l'écrire

```
mysql> CREATE ROLE crm_dev, crm_read, crm_write;  
Query OK, 0 rows affected (0,05 sec)  
  
mysql> GRANT ALL ON crm.* TO crm_dev;  
Query OK, 0 rows affected (0,07 sec)  
  
mysql> GRANT SELECT ON crm.* TO crm_read;  
Query OK, 0 rows affected (0,11 sec)  
  
mysql> GRANT INSERT, UPDATE, DELETE ON crm.* TO crm_write;  
Query OK, 0 rows affected (0,05 sec)
```

Puis nous créons des utilisateurs

```
mysql> -- developer user  
mysql> CREATE USER crm_dev1@localhost IDENTIFIED BY 'Secure$1782';  
Query OK, 0 rows affected (0,03 sec)  
  
mysql> -- read access user  
mysql> CREATE USER crm_read1@localhost IDENTIFIED BY 'Secure$5432';
```

```
Query OK, 0 rows affected (0,01 sec)
```

```
mysql> -- read/write users
```

```
mysql> CREATE USER crm_writel@localhost IDENTIFIED BY 'Secure$9075';
```

```
Query OK, 0 rows affected (0,01 sec)
```

```
mysql> CREATE USER crm_write2@localhost IDENTIFIED BY 'Secure$3452';
```

```
Query OK, 0 rows affected (0,14 sec)
```

Enfin nous associons nos utilisateurs a nos rôle

```
mysql> GRANT crm_dev TO crm_dev1@localhost;
```

```
Query OK, 0 rows affected (0,08 sec)
```

```
mysql>
```

```
mysql> GRANT crm_read TO crm_read1@localhost;
```

```
Query OK, 0 rows affected (0,01 sec)
```

```
mysql>
```

```
mysql> GRANT crm_read, crm_write TO crm_writel@localhost, crm_write2@localhost;
```

```
Query OK, 0 rows affected (0,11 sec)
```

Il est possible de voir les droits d'un utilisateur, qui sont en fait les droits de l'utilisateur associé aux rôles:

```
SHOW GRANTS FOR crm_dev1@localhost;
```

```
+-----+
```

```
| Grants for crm_dev1@localhost |
```

```
+-----+
```

```
| GRANT USAGE ON *.* TO `crm_dev1`@`localhost` |
```

```
| GRANT `crm_dev`@`%` TO `crm_dev1`@`localhost` |
```

```
+-----+
```

```
2 rows in set (0,00 sec)
```

Pour voir les droits d'un utilisateur en fonction d'un rôle, il faut le demander explicitement via la clause USING

```
mysql> SHOW GRANTS FOR crm_writel@localhost USING crm_write;
+-----+
| Grants for crm_writel@localhost |
+-----+
| GRANT USAGE ON *.* TO `crm_writel`@`localhost` |
| GRANT INSERT, UPDATE, DELETE ON `crm`.* TO `crm_writel`@`localhost` |
| GRANT `crm_read`@`%`,`crm_write`@`%` TO `crm_writel`@`localhost` |
+-----+
3 rows in set (0,00 sec)
```

Les droits sont associé a un rôle. Lors de la connection à MySQL, l'utilisateur doit spécifier le rôle qu'il souhaite utiliser

```
pilou@lubuntu: ~/mysql80/mysql-8.0.13-linux-glibc2.12-x86_64$ ./bin/mysql -u crm_read1 -h
localhost -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.13 MySQL Community Server - GPL

Copyright (c) 2000, 2018, 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> select current_role();
+-----+
| current_role() |
+-----+
| NONE |
+-----+
1 row in set (0,00 sec)

mysql> SET ROLE crm_read;
Query OK, 0 rows affected (0,00 sec)

mysql> select current_role() ;
```

```
+-----+
| current_role() |
+-----+
| `crm_read`@`%` |
+-----+
1 row in set (0,00 sec)

mysql> use crm;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

connexion via PAM

Les plugins d'authentification sont multiples en version entreprise et sont souvent en version GPL avec mariadb.

Il est possible d'authentifier les utilisateurs via PAM en utilisant le systeme unix pour les authentifier en lieu et place de MySQL.

Sous mariadb installer le plugin auth_pam

```
INSTALL SONAME 'auth_pam';
```

Puis créer un utilisateur qui pourras se connecter via pam

```
CREATE USER 'test_pam'@'localhost' IDENTIFIED VIA pam USING 'mariadb';
```

Créer l'utilisateur sous Linux (ici ubuntu)

```
addUser test_pam
```

Dans le repertoire /etc/pam.d on va editer la configuration pam de MySQL en demandant a ce que la vérification des utilisateurs se fasse via les comptes Unix

```
#%PAM-1.0
```

```
@include common-auth
```

```
@include common-account
```

```
@include common-session-noninteractive
```

TP 1 Prise en main

Démarrage de MySQL

Le démarrage se fait via simple.sh

Faire un show engines:

```
SHOW ENGINES \G;
***** 1. row *****
      Engine: FEDERATED
      Support: NO
      Comment: Federated MySQL storage engine
Transactions: NULL
           XA: NULL
      Savepoints: NULL
***** 2. row *****
      Engine: MEMORY
      Support: YES
      Comment: Hash based, stored in memory, useful for temporary tables
Transactions: NO
           XA: NO
      Savepoints: NO
***** 3. row *****
      Engine: InnoDB
      Support: DEFAULT
      Comment: Supports transactions, row-level locking, and foreign keys
Transactions: YES
           XA: YES
      Savepoints: YES
***** 4. row *****
      Engine: PERFORMANCE_SCHEMA
      Support: YES
      Comment: Performance Schema
Transactions: NO
           XA: NO
      Savepoints: NO
***** 5. row *****
```

```

    Engine: MyISAM
    Support: YES
    Comment: MyISAM storage engine
Transactions: NO
    XA: NO
    Savepoints: NO
***** 6. row *****
    Engine: MRG_MYISAM
    Support: YES
    Comment: Collection of identical MyISAM tables
Transactions: NO
    XA: NO
    Savepoints: NO
***** 7. row *****
    Engine: BLACKHOLE
    Support: YES
    Comment: /dev/null storage engine (anything you write to it disappears)
Transactions: NO
    XA: NO
    Savepoints: NO
***** 8. row *****
    Engine: CSV
    Support: YES
    Comment: CSV storage engine
Transactions: NO
    XA: NO
    Savepoints: NO
***** 9. row *****
    Engine: ARCHIVE
    Support: YES
    Comment: Archive storage engine
Transactions: NO
    XA: NO
    Savepoints: NO
9 rows in set (0.00 sec)

```

Faire un show tables afin de voir les détails d'une tables:

```

mysql> SHOW table status \G;
***** 1. row *****

```

```
Name: titi
Engine: InnoDB
Version: 10
Row_format: Dynamic
Rows: 0
Avg_row_length: 0
Data_length: 16384
Max_data_length: 0
Index_length: 0
Data_free: 0
Auto_increment: NULL
Create_time: 2020-06-16 21:03:06
Update_time: NULL
Check_time: NULL
Collation: utf8mb4_0900_ai_ci
Checksum: NULL
Create_options:
Comment:
1 row in set (0.00 sec)
```

Faire un

```
CREATE TABLE t2 (i INT NOT NULL) ENGINE = CSV;
Query OK, 0 rows affected (0.05 sec)
```

Et vérifier que le fichier est bien là:

```
/home/pilou/Formation/simpleinit/data/t1
```

Rajoutons une ligne et remarquons qu'il faut bien flusher les tables si on veut que cela fasse qq chose:

```
mysql> select * from t2;
+---+
| i |
+---+
| 4 |
+---+
1 row in set (0.00 sec)
```

```
mysql> flush tables;
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> select * from t2;
+---+
| i |
+---+
| 4 |
| 5 |
+---+
2 rows in set (0.01 sec)
```

Le moteur de stockage peut être changer dynamiquement par défaut:

```
SET default_storage_engine=MYISAM;
```

On vérifie que MYISAM n'est pas transactionnel:

```
mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into testisam values(3);
Query OK, 1 row affected (0.01 sec)

mysql> rollback;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> select * from testisam;
+-----+
| i   |
+-----+
|  3 |
+-----+
1 row in set (0.00 sec)
```

Il est possible de modifier le storage engine de la table

```
mysql> ALTER TABLE testisam ENGINE = InnoDB;
Query OK, 1 row affected (0.07 sec)
Records: 1 Duplicates: 0 Warnings: 0
```

mysql tuner

```
[--] Status: +ARCHIVE +BLACKHOLE +CSV -FEDERATED +InnoDB +MEMORY +MRG_MYISAM +MyISAM
+PERFORMANCE_SCHEMA
>default-storage-engine=InnoDB
>disabled_storage_engines="MyISAM, BLACKHOLE, FEDERATED, ARCHIVE, CSV"

[--] Data in CSV tables: 0B (Tables: 1)
[--] Data in InnoDB tables: 49.0K (Tables: 4)
[OK] Total fragmented tables: 0

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

----- Security Recommendations
-----
[--] Skipped due to unsupported feature for MySQL 8

----- CVE Security Recommendations
-----
[OK] NO SECURITY CVE FOUND FOR YOUR VERSION

----- Performance Metrics
-----
[--] Up for: 54s (22 q [0.407 qps], 15 conn, TX: 56K, RX: 1K)
[--] Reads / Writes: 100% / 0%
[--] Binary logging is enabled (GTID MODE: OFF)
[--] Physical Memory      : 2.9G
[--] Max MySQL memory     : 9.8G
[--] Other process memory: 0B
[--] Total buffers: 168.0M global + 65.1M per thread (151 max threads)
151 est max_connections. 168M read_buffer_size + sort_buffer_size + join_buffer_size

[--] P_S Max memory usage: 72B
[--] Galera GCache Max memory usage: 0B
[OK] Maximum reached memory usage: 298.3M (9.97% of installed RAM)
```

```
[!!] Maximum possible memory usage: 9.8G (334.35% of installed RAM)
[!!] Overall possible memory usage with other process exceeded memory
[OK] Slow queries: 0% (0/22)
[OK] Highest usage of available connections: 1% (2/151)
[OK] Aborted connections: 0.00% (0/15)
[!!] name resolution is active : a reverse name resolution is made for each new connection and
can reduce performance
[--] Query cache have been removed in MySQL 8
[OK] Sorts requiring temporary tables: 0% (0 temp sorts / 2 sorts)
[!!] Joins performed without indexes: 1
[OK] Temporary tables created on disk: 0% (0 on disk / 5 total)
[OK] Thread cache hit rate: 86% (2 created / 15 connections)
[OK] Table cache hit rate: 64% (145 open / 224 opened)
[OK] table_definition_cache(2000) is upper than number of tables(315)
[OK] Open file limit used: 0% (2/8K)
[OK] Table locks acquired immediately: 100% (4 immediate / 4 locks)
[OK] Binlog cache memory access: 0% (0 Memory / 0 Total)
```

----- Performance schema

```
-----
[--] Memory used by P_S: 72B
[--] Sys schema is installed.
```

----- ThreadPool Metrics

```
-----
[--] ThreadPool stat is disabled.
```

----- MyISAM Metrics

```
-----
[--] MyISAM Metrics are disabled on last MySQL versions.
```

----- InnoDB Metrics

```
-----
[--] InnoDB is enabled.
[--] InnoDB Thread Concurrency: 0
[OK] InnoDB File per table is activated
[OK] InnoDB buffer pool / data size: 128.0M/49.0K
[!!] Ratio InnoDB log file size / InnoDB Buffer pool size (75 %): 48.0M * 2/128.0M should be
equal to 25%
[OK] InnoDB buffer pool instances: 1
```

```
[--] Number of InnoDB Buffer Pool Chunk : 1 for 1 Buffer Pool Instance(s)
[OK] Innodb_buffer_pool_size aligned with Innodb_buffer_pool_chunk_size &
Innodb_buffer_pool_instances
[OK] InnoDB Read buffer efficiency: 94.02% (13442 hits/ 14297 total)
[OK] InnoDB Write log efficiency: 97.67% (630 hits/ 645 total)
[OK] InnoDB log waits: 0.00% (0 waits / 15 writes)
```

----- AriaDB Metrics

[--] AriaDB is disabled.

----- TokuDB Metrics

[--] TokuDB is disabled.

----- XtraDB Metrics

[--] XtraDB is disabled.

----- Galera Metrics

[--] Galera is disabled.

----- Replication Metrics

[--] Galera Synchronous replication: NO
[--] No replication slave(s) for this server.
[--] Binlog format: ROW
[--] XA support enabled: ON
[--] Semi synchronous replication Master: Not Activated
[--] Semi synchronous replication Slave: Not Activated
[--] This is a standalone server

----- Recommendations

General recommendations:

Control warning line(s) into /home/pilou/Formation/simpleinit/mysqld.log file
Control error line(s) into /home/pilou/Formation/simpleinit/mysqld.log file
MySQL was started within the last 24 hours - recommendations may be inaccurate
Reduce your overall MySQL memory footprint for system stability

Dedicate this server to your database for highest performance.

Configure your accounts with ip or subnets only, then update your configuration with skip-name-resolve=1

We will suggest raising the 'join_buffer_size' until JOINS not using indexes are found.

See <https://dev.mysql.com/doc/internals/en/join-buffer-size.html>

(specially the conclusions at the bottom of the page).

Before changing innodb_log_file_size and/or innodb_log_files_in_group read this:

<https://bit.ly/2TcGgtU>

Variables to adjust:

*** MySQL's maximum memory usage is dangerously high ***

*** Add RAM before increasing MySQL buffer variables ***

join_buffer_size (> 256.0K, or always use indexes with JOINS)

innodb_log_file_size should be (=16M) if possible, so InnoDB total log files size equals to 25% of buffer pool size.

https://wiki.deimos.fr/MysqITuner:_Optimiser_votre_serveur_MySQL.html#Patch_mysqltuner

<https://www.mysqlcalculator.com/>

<https://www.percona.com/blog/2016/05/03/best-practices-for-configuring-optimal-mysql-memory-usage/>

My.ini sample

```
# MySQL Server Instance Configuration File
# -----
# Generated by the MySQL Server Instance Configuration Wizard
#
#
# Installation Instructions
# -----
#
# On Linux you can copy this file to /etc/my.cnf to set global options,
# mysql-data-dir/my.cnf to set server-specific options
# (@localstatedir@ for this installation) or to
# ~/.my.cnf to set user-specific options.
#
# On Windows you should keep this file in the installation directory
```

```
# of your server (e.g. C:\Program Files\MySQL\MySQL Server X.Y). To
# make sure the server reads the config file use the startup option
# "--defaults-file".
#
# To run the server from the command line, execute this in a
# command line shell, e.g.
# mysqld --defaults-file="C:\Program Files\MySQL\MySQL Server X.Y\my.ini"
#
# To install the server as a Windows service manually, execute this in a
# command line shell, e.g.
# mysqld --install MySQLXY --defaults-file="C:\Program Files\MySQL\MySQL Server X.Y\my.ini"
#
# And then execute this in a command line shell to start the server, e.g.
# net start MySQLXY
#
#
# Guidelines for editing this file
# -----
#
# In this file, you can use all long options that the program supports.
# If you want to know the options a program supports, start the program
# with the "--help" option.
#
# More detailed information about the individual options can also be
# found in the manual.
#
#
# CLIENT SECTION
# -----
#
# The following options will be read by MySQL client applications.
# Note that only client applications shipped by MySQL are guaranteed
# to read this section. If you want your own MySQL client program to
# honor these values, you need to specify it as an option during the
# MySQL client library initialization.
#
[client]

port=3306
```

```
[mysql]
```

```
default-character-set=latin1
```

```
# SERVER SECTION
```

```
# -----
```

```
#
```

```
# The following options will be read by the MySQL Server. Make sure that  
# you have installed the server correctly (see above) so it reads this  
# file.
```

```
#
```

```
[mysqld]
```

```
# The TCP/IP Port the MySQL Server will listen on  
port=3306
```

```
#Path to installation directory. All paths are usually resolved relative to this.  
basedir="C:/Program Files/MySQL/MySQL Server 5.1/"
```

```
#Path to the database root  
datadir="C:/ProgramData/MySQL/MySQL Server 5.1/Data/"
```

```
# The default character set that will be used when a new schema or table is  
# created and no character set is defined  
character-set-server=latin1
```

```
# The default storage engine that will be used when create new tables when  
default-storage-engine=INNODB
```

```
# Set the SQL mode to strict  
sql-mode="STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION"
```

```
# The maximum amount of concurrent sessions the MySQL server will  
# allow. One of these connections will be reserved for a user with  
# SUPER privileges to allow the administrator to login even if the  
# connection limit has been reached.  
max_connections=100
```

```
# Query cache is used to cache SELECT results and later return them
# without actual executing the same query once again. Having the query
# cache enabled may result in significant speed improvements, if your
# have a lot of identical queries and rarely changing tables. See the
# "Qcache_lowmem_prunes" status variable to check if the current value
# is high enough for your load.
# Note: In case your tables change very often or if your queries are
# textually different every time, the query cache may result in a
# slowdown instead of a performance improvement.
query_cache_size=0

# The number of open tables for all threads. Increasing this value
# increases the number of file descriptors that mysqld requires.
# Therefore you have to make sure to set the amount of open files
# allowed to at least 4096 in the variable "open-files-limit" in
# section [mysqld_safe]
table_cache=256

# Maximum size for internal (in-memory) temporary tables. If a table
# grows larger than this value, it is automatically converted to disk
# based table This limitation is for a single table. There can be many
# of them.
tmp_table_size=205M

# How many threads we should keep in a cache for reuse. When a client
# disconnects, the client's threads are put in the cache if there aren't
# more than thread_cache_size threads from before. This greatly reduces
# the amount of thread creations needed if you have a lot of new
# connections. (Normally this doesn't give a notable performance
# improvement if you have a good thread implementation.)
thread_cache_size=8

**** MyISAM Specific options

# The maximum size of the temporary file MySQL is allowed to use while
# recreating the index (during REPAIR, ALTER TABLE or LOAD DATA INFILE.
# If the file-size would be bigger than this, the index will be created
# through the key cache (which is slower).
myisam_max_sort_file_size=100G
```

```
# If the temporary file used for fast index creation would be bigger
# than using the key cache by the amount specified here, then prefer the
# key cache method. This is mainly used to force long character keys in
# large tables to use the slower key cache method to create the index.
mysam_sort_buffer_size=410M
```

```
# Size of the Key Buffer, used to cache index blocks for MyISAM tables.
# Do not set it larger than 30% of your available memory, as some memory
# is also required by the OS to cache rows. Even if you're not using
# MyISAM tables, you should still set it to 8-64M as it will also be
# used for internal temporary disk tables.
key_buffer_size=354M
```

```
# Size of the buffer used for doing full table scans of MyISAM tables.
# Allocated per thread, if a full scan is needed.
read_buffer_size=64K
read_rnd_buffer_size=256K
```

```
# This buffer is allocated when MySQL needs to rebuild the index in
# REPAIR, OPTIMIZE, ALTER table statements as well as in LOAD DATA INFILE
# into an empty table. It is allocated per thread so be careful with
# large settings.
sort_buffer_size=256K
```

```
*** INNODB Specific options ***
```

```
# Use this option if you have a MySQL server with InnoDB support enabled
# but you do not plan to use it. This will save memory and disk space
# and speed up some things.
#skip-innodb
```

```
# Additional memory pool that is used by InnoDB to store metadata
# information. If InnoDB requires more memory for this purpose it will
# start to allocate it from the OS. As this is fast enough on most
# recent operating systems, you normally do not need to change this
# value. SHOW INNODB STATUS will display the current amount used.
innodb_additional_mem_pool_size=15M
```

```
# If set to 1, InnoDB will flush (fsync) the transaction logs to the
# disk at each commit, which offers full ACID behavior. If you are
# willing to compromise this safety, and you are running small
# transactions, you may set this to 0 or 2 to reduce disk I/O to the
# logs. Value 0 means that the log is only written to the log file and
# the log file flushed to disk approximately once per second. Value 2
# means the log is written to the log file at each commit, but the log
# file is only flushed to disk approximately once per second.
innodb_flush_log_at_trx_commit=1
```

```
# The size of the buffer InnoDB uses for buffering log data. As soon as
# it is full, InnoDB will have to flush it to disk. As it is flushed
# once per second anyway, it does not make sense to have it very large
# (even with long transactions).
innodb_log_buffer_size=7M
```

```
# InnoDB, unlike MyISAM, uses a buffer pool to cache both indexes and
# row data. The bigger you set this the less disk I/O is needed to
# access data in tables. On a dedicated database server you may set this
# parameter up to 80% of the machine physical memory size. Do not set it
# too large, though, because competition of the physical memory may
# cause paging in the operating system. Note that on 32bit systems you
# might be limited to 2-3.5G of user level memory per process, so do not
# set it too high.
innodb_buffer_pool_size=686M
```

```
# Size of each log file in a log group. You should set the combined size
# of log files to about 25%-100% of your buffer pool size to avoid
# unneeded buffer pool flush activity on log file overwrite. However,
# note that a larger logfile size will increase the time needed for the
# recovery process.
innodb_log_file_size=343M
```

```
# Number of threads allowed inside the InnoDB kernel. The optimal value
# depends highly on the application, hardware as well as the OS
# scheduler properties. A too high value may lead to thread thrashing.
innodb_thread_concurrency=10
```

My.ini 2

```
#
# FromDual configuration file template for MySQL, Galera Cluster, MariaDB and Percona Server
# Location: %MYCNF%
# This template is intended to work with MySQL 5.7 and newer and MariaDB 10.0 and newer
# Get most recent updated from here:
# https://www.fromdual.com/mysql-configuration-file-sample
#

[client]

port                = %PORT%                # default 3306
socket              = %SOCKET%              # Use mysqld.sock on
Ubuntu, conflicts with AppArmor otherwise

[mysql]

no_auto_rehash

max_allowed_packet = 16M
prompt            = '\u@\h [\d]> '          # 'user@host [schema]> '
default_character_set = utf8                # Possibly this setting
is correct for most recent Linux systems

[mysqldump]

max_allowed_packet = 16M

[mysqld_safe]

obsolete with systemd                                # Becomes sooner or later

open_files_limit = 8192                        # You possibly have to
adapt your O/S settings as well

user                = mysql

log-error           = %INSTANCEDIR%/log/%UNAME%_%INSTANCE%_error.log # Adjust
AppArmor configuration: /etc/apparmor.d/local/usr.sbin.mysqld
```

```

[mysqld]

# Connection and Thread variables

port                = %PORT%                # default 3306
socket              = %SOCKET%              # Use mysqld.sock on
Ubuntu, conflicts with AppArmor otherwise
basedir             = %BASEDIR%
datadir             = %DATADIR%
# tmpdir            = '%INSTANCEDIR%/tmp'
# innodb_tmpdir     = '%INSTANCEDIR%/tmp'   # MySQL 5.7

max_allowed_packet  = 16M
default_storage_engine = InnoDB
# explicit_defaults_for_timestamp = 1      # MySQL 5.6, test
carefully! This can have an impact on application.
# disable_partition_engine_check = true    # Since MySQL 5.7.17 to
5.7.20. To get rid of nasty message in error log

# character_set_server = utf8mb4          # For modern
applications, default in MySQL 8.0
# collation_server    = utf8mb4_general_ci

max_connections     = 151                  # Values < 1000 are
typically good
max_user_connections = 145                # Limit one specific
user/application
thread_cache_size   = 151                 # Up to max_connections
makes sense

# Query Cache (does not exist in MySQL 8.0 any more!)

# query_cache_type   = 1                  # Set to 0 to avoid
global QC Mutex
# query_cache_size   = 32M                # Avoid too big (>
128M) QC because of QC clean-up lock!

```

```

# Session variables

sort_buffer_size          = 2M                # Could be too big for
many small sorts

tmp_table_size            = 32M               # Make sure your
temporary results do NOT contain BLOB/TEXT attributes

read_buffer_size          = 128k              # Resist to change this
parameter if you do not know what you are doing

read_rnd_buffer_size      = 256k             # Resist to change this
parameter if you do not know what you are doing

join_buffer_size          = 128k             # Resist to change this
parameter if you do not know what you are doing

# Other buffers and caches

table_definition_cache    = 1400             # As big as many tables
you have

table_open_cache          = 2000             # connections x
tables/connection (~2)

table_open_cache_instances = 16              # New default in 5.7

# MySQL error log

log_error                  = %INSTANCEDIR%/log/%UNAME%_%INSTANCE%_error.log # Adjust
AppArmor configuration: /etc/apparmor.d/local/usr.sbin.mysqlld

# log_timestamps            = SYSTEM           # MySQL 5.7, equivalent
to old behaviour

log_warnings               = 2                # MySQL 5.6, equivalent
to log_error_verbosity = 3

# log_error_verbosity       = 3                # MySQL 5.7, equivalent
to log_warnings = 2, MariaDB does NOT support this!

innodb_print_all_deadlocks = 1

# wsrep_log_conflicts       = 1                # for Galera only!

# Slow Query Log

```



```

holidays
binlog_format          = ROW                      # Use MIXED if you want
to experience some troubles
# binlog_row_image     = MINIMAL                  # Since 5.6
# auto_increment_increment = 2                    # For Master/Master
set-ups use 2 for both nodes
# auto_increment_offset = 1                      # For Master/Master
set-ups use 1 and 2

# Slave variables

log_slave_updates     = 1                        # Use if Slave is used
for Backup and PiTR
read_only              = 0                        # Set to 1 to prevent
writes on Slave
# super_read_only      = 0                        # Set to 1 to prevent
writes on Slave for users with SUPER privilege. Since 5.7, not in MariaDB
# skip_slave_start     = 1                        # To avoid start of
Slave thread
# relay_log            = %UNAME%_%INSTANCE%_relay-bin
# relay_log_info_repository = table                # MySQL 5.6
# master_info_repository = table                 # MySQL 5.6
# slave_load_tmpdir    = '%INSTANCEDIR%/tmp'

# Crash-safe replication Master

# binlog_checksum      = CRC32                    # default
# sync_binlog          = 1                        # default since 5.7.6,
but slow!
# innodb_support_xa    = 1                        # default, depracted
since 5.7.10

# Crash-safe replication Slave

# master_info_repository = TABLE
# relay_log_info_repository = TABLE
# relay_log_recovery     = 1

```

```

# sync_relay_log_info          = 1
# relay_log_purge              = 1                      # default
# slave_sql_verify_checksum    = 1                      # default

# GTID replication

# gtid_mode                    = ON                      # Master and Slave
# enforce_gtid_consistency     = 1                      # Master and Slave

# log_bin                      = %INSTANCEDIR%/binlog/%UNAME%_%INSTANCE%_binlog  # In 5.6
also on Slave
# log_slave_updates           = 1                      # In 5.6 also on
Slave

# Security variables

# local_infile                 = 0                      # If you are security
aware
# secure_auth                  = 1                      # If you are security
aware
# sql_mode                     =
TRADITIONAL, ONLY_FULL_GROUP_BY, NO_ENGINE_SUBSTITUTION, STRICT_ALL_TABLES, NO_ZERO_IN_DATE, NO_ZER
O_DATE, ERROR_FOR_DIVISION_BY_ZERO, NO_AUTO_CREATE_USER  # Be careful changing this afterwards
# skip_name_resolve            = 0                      # Set to 1 if you do
not trust your DNS or experience problems
# secure_file_priv             = '%INSTANCEDIR%/tmp'      # chmod 750, adjust
AppArmor configuration: /etc/apparmor.d/local/usr.sbin.mysql

# MyISAM variables

key_buffer_size                = 8M                      # Set to 25 - 33 % of RAM
if you still use MyISAM
myisam_recover_options         = 'BACKUP, FORCE'
# disabled_storage_engines     = 'MyISAM, MEMORY'      # MySQL 5.7, do NOT
during/before mysql_upgrade, good for Galera!

```

```

# MEMORY variables

max_heap_table_size          = 64M                # Should be greater or
equal to tmp_table_size

# InnoDB variables

innodb_strict_mode           = ON

# innodb_file_format_check    = 1                # Desupported in MySQL
8.0

# innodb_file_format          = Barracuda        # For dynamic and
compressed InnoDB tables, default in 5.7

innodb_buffer_pool_size      = 128M              # Go up to 80% of your
available RAM

innodb_buffer_pool_instances = 8                # Bigger if huge InnoDB
Buffer Pool or high concurrency

innodb_file_per_table        = 1                # Is the recommended way
nowadays

# innodb_flush_method          = 0_DIRECT         # 0_DIRECT is sometimes
better for direct attached storage

# innodb_write_io_threads      = 8              # If you have a strong
I/O system or SSD

# innodb_read_io_threads      = 8              # If you have a strong
I/O system or SSD

# innodb_io_capacity           = 1000           # If you have a strong
I/O system or SSD

innodb_flush_log_at_trx_commit = 2              # 1 for durability, 0 or
2 for performance

innodb_log_buffer_size       = 8M              # Bigger if

innodb_flush_log_at_trx_commit = 0

innodb_log_file_size         = 256M           # Bigger means more write
throughput but longer recovery time

# Since MariaDB 10.0 and
MySQL 5.6

innodb_monitor_enable = all                    # Overhead < 1% according
to PeterZ/Percona

```

```

# Galera specific MySQL parameter

# default_storage_engine          = InnoDB          # Galera only works with
InnoDB
# innodb_flush_log_at_trx_commit = 2                # Durability is achieved
by committing to the Group
# innodb_autoinc_lock_mode       = 2                # For parallel applying
# binlog_format                   = row              # Galera only works with
RBR
# query_cache_type                = 0                # Use QC with Galera only
in a Master/Slave set-up
# query_cache_size                = 0

# WSREP parameter

# wsrep_on                        = on                # Only MariaDB >= 10.1
# wsrep_provider                  = /usr/lib/galera/libgalera_smm.so # Location of Galera
Plugin on Ubuntu ?
# wsrep_provider                  = /usr/lib64/galera-3/libgalera_smm.so # Location of Galera
Plugin on CentOS 7
# wsrep_provider                  = none              # Start mysqld without
Galera
# wsrep_provider_options          = 'gcache.size = 1G' # Depends on you
workload, WS kept for IST

# wsrep_cluster_name              = "My cool Galera Cluster" # Same Cluster name for
all nodes
# wsrep_cluster_address           = "gcomm: //192.168.0.2,192.168.0.3" # Start other nodes
like this

# wsrep_node_name                 = "Node A"          # Unique node name
# wsrep_node_address              = 192.168.0.1       # Our address where
replication is done
# wsrep_node_incoming_address     = 10.0.0.1         # Our external
interface where application comes from
# wsrep_sync_wait                 = 1                # If you need really
full-synchronous replication (Galera 3.6 and newer)

```

```

# wsrep_slave_threads          = 16                                # 4 - 8 per core, not
more than wsrep_cert_deps_distance

# wsrep_sst_method             = rsync                            # SST method (initial
full sync): mysqldump, rsync, rsync_wan, xtrabackup-v2
# wsrep_sst_auth               = sst:secret                       # Username/password for
sst user
# wsrep_sst_receive_address    = 192.168.2.1                     # Our address where to
receive SST

# Group Replication parameter

# default_storage_engine      = InnoDB                          # Group Replication
only works with InnoDB
# server_id                    = %SERVERID%                      # Should be different
on all 3 nodes
# log_bin                      = %INSTANCEDIR%/binlog/%UNAME%_%INSTANCE%_binlog # Locate
outside of datadir, adjust AppArmor configuration: /etc/apparmor.d/local/usr.sbin.mysql
# binlog_format                = ROW
# binlog_checksum              = NONE                            # not default!
# gtid_mode                    = ON
# enforce_gtid_consistency     = ON
# master_info_repository      = TABLE
# relay_log_info_repository    = TABLE
# log_slave_updates           = ON

# slave_parallel_workers      = <n>                              # 1-2/core, max. 10
# slave_preserve_commit_order = ON
# slave_parallel_type          = LOGICAL_CLOCK

# transaction_write_set_extraction = XXHASH64

# loose-group_replication_group_name = "$(uuidgen)"            # Must be the same on
all nodes
# loose-group_replication_start_on_boot = OFF
# loose-group_replication_local_address = "192.168.0.1"
# loose-group_replication_group_seeds   = "192.168.0.1,192.168.0.2,192.168.0.3" # All
nodes of Cluster
# loose-group_replication_bootstrap_group = OFF

```

```
# loose-group_replication_single_primary_mode = FALSE
```

```
# = multi-primary
```

MySQL Cluster

MySQLCluster sur docker

Nous allons mettre en place un cluster MySQL basé sur MySQL Cluster.

Mise en place du réseau

```
docker network create cluster --subnet=192.168.0.0/16
```

Puis récupérons les fichiers ou recopier les fichier de configuration:

```
https://github.com/mysql/mysql-docker/tree/mysql-cluster
```

Le fichier my.cnf est ainsi:

```
# Copyright (c) 2017, Oracle and/or its affiliates. All rights reserved.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; version 2 of the License.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

[mysqld]
ndbcluster
ndb-connectstring=192.168.0.2
user=mysql

[mysql_cluster]
ndb-connectstring=192.168.0.2
```

et le fichier mysqlcluster.cnf

```
# Copyright (c) 2017, Oracle and/or its affiliates. All rights reserved.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; version 2 of the License.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

[ndbd default]
NoOfReplicas=2
DataMemory=80M
IndexMemory=18M

[ndb_mgmd]
NodeId=1
hostname=192.168.0.2
datadir=/var/lib/mysql

[ndbd]
NodeId=2
hostname=192.168.0.3
datadir=/var/lib/mysql

[ndbd]
NodeId=3
hostname=192.168.0.4
datadir=/var/lib/mysql

[mysqld]
NodeId=4
```

```
hostname=192.168.0.10
```

Lancement du noeud de management

```
docker run -d --net=cluster --name=management1 --ip=192.168.0.2 mysql/mysql-cluster ndb_mgmd
```

Puis lancons les noeud de données:

```
docker run -d --net=cluster --name=ndb1 --ip=192.168.0.3 mysql/mysql-cluster ndbd
docker run -d --net=cluster --name=ndb2 --ip=192.168.0.4 mysql/mysql-cluster ndbd
```

Et enfin on peut executer le noeud SQL:

```
docker run -d --net=cluster --name=mysql1 --ip=192.168.0.10 -e MYSQL_RANDOM_ROOT_PASSWORD=true
mysql/mysql-cluster mysqld
```

Ce noeud va générer un password aléatoire :

```
docker logs mysql1 2>&1 | grep PASSWORD
[Entrypoint] GENERATED ROOT PASSWORD: =EbISQomAxv0mnam4d9EJigIjwA
```

Changement du mot de passe:

```
docker exec -it mysql1 mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.30-ndb-7.6.14-cluster-gpl MySQL Cluster Community Server (GPL)

Copyright (c) 2000, 2020, 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> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass';
Query OK, 0 rows affected (0.06 sec)
```

Le management du cluster se fait via :

```
docker run -it --net=cluster mysql/mysql-cluster ndb_mgm
```

La modification du fichier mysqlcluster.cnf permet de rajouter un noeud de donnée et de relancer

```
docker run -d --net=cluster --name=management1 --ip=192.168.0.2 -v
/home/pilou/Formation/cluster/mysql-cluster.cnf:/etc/mysql-cluster.cnf mysql/mysql-cluster
ndb_mgmd
9ac595c65ddf47568f20fa835e10d45d0e7adcaf40bab818ac94b9afd20524ec
pilou@pilou-pc: ~/Formation/cluster$ docker logs management1[Entrypoint] MySQL Docker Image
7.6.14-1.1.16-cluster
[Entrypoint] Starting ndb_mgmd
MySQL Cluster Management Server mysql-5.7.30 ndb-7.6.14
2020-06-09 20:25:07 [MgmtSrvr] INFO      -- The default config directory '/usr/mysql-cluster'
does not exist. Trying to create it...
2020-06-09 20:25:07 [MgmtSrvr] INFO      -- Successfully created config directory
2020-06-09 20:25:07 [MgmtSrvr] WARNING  -- at line 19: [DB] IndexMemory is deprecated, will
use Number bytes on each ndbd(DB) node allocated for storing indexes instead
2020-06-09 20:25:07 [MgmtSrvr] INFO      -- Got initial configuration from '/etc/mysql-
cluster.cnf', will try to set it when all ndb_mgmd(s) started
2020-06-09 20:25:07 [MgmtSrvr] INFO      -- Node 1: Node 1 Connected
2020-06-09 20:25:07 [MgmtSrvr] INFO      -- Id: 1, Command port: *:1186
==INITIAL==
2020-06-09 20:25:07 [MgmtSrvr] INFO      -- MySQL Cluster Management Server mysql-5.7.30 ndb-
7.6.14 started
2020-06-09 20:25:08 [MgmtSrvr] INFO      -- Node 1 connected
2020-06-09 20:25:08 [MgmtSrvr] INFO      -- Starting initial configuration change
2020-06-09 20:25:08 [MgmtSrvr] INFO      -- Configuration 1 committed
2020-06-09 20:25:08 [MgmtSrvr] INFO      -- Config change completed! New generation: 1
==CONFIRMED==
```

Outil NDB

- ndbinfo_select_all - Sélection dans les tables ndbinfo
- ndbmtdd - Le démon de nœud de données de cluster NDB (multi-thread)

- ndb_mgmd - Le démon du serveur de gestion de cluster NDB
- ndb_mgm - Le client de gestion de cluster NDB
- ndb_blob_tool - Vérifie et répare les colonnes BLOB et TEXT des tables de cluster NDB
- ndb_config - Extraire les informations de configuration du cluster NDB
- ndb_cpctest - Tests automatiques pour le développement NDB
- ndb_delete_all - Supprime toutes les lignes d'une table NDB
- ndb_desc - Décrit les tables NDB
- ndb_drop_index - Supprime l'index d'une table NDB
- ndb_drop_table - Supprime une table NDB
- ndb_error_reporter - Utilitaire de rapport d'erreurs NDB
- ndb_import - Importer des données CSV dans le NDB

Sizing

Executer

```
perl ./bin/ndb_size.pl --socket /tmp/mysql.sock --user=root --password=piloupilou </source>
```

Parameter	Default	4.1	5.0	5.1
NoOfAttributes		1000	3676*	3676*
NoOfUniqueHashIndexes		64	0	0
IndexMemory (KB)		18432	4592	3568
NoOfOrderedIndexes		128	233*	233*
DataMemory (KB)		81920	55328	55328
NoOfTriggers		768	1985*	1985*
NoOfTables		128	438*	438*

Fichier de configuration conseillé:

LE fichier suivant est un fichier config.ini correct pour la plupart des cluster:

```
[tcp default] SendBufferMemory=2M ReceiveBufferMemory=2M
```

```
[ndb_mgmd default] datadir=X
```

```
[ndbd default]
```

1. redundancy:

```
NoOfReplicas=2
```

1. avoid swapping:

LockPagesInMainMemory=1

1. Bypass FS cache (you should test if this works for you or not)

Odirect=1

1. DataMemory (memory for records and ordered indexes)

DataMemory=2048M

1. IndexMemory (memory for Primary key hash index and unique hash index)
2. Usually between 1/6 or 1/8 of the DataMemory is enough, but depends on the
3. number of unique hash indexes (UNIQUE in table def)

IndexMemory=256M

1. Redolog
2. size of each redo log fragment, 4 redo log fragment makes up on fragment log file.
3. A bigger Fragment log file size than the default 16M works better with high write load
4. and is strongly recommended!!

FragmentLogFileSize=256M

1. Set NoOfFragmentLogFiles to $6 \times \text{DataMemory [in MB]} / (4 \times \text{FragmentLogFileSize [in MB]})$
2. Thus, $\text{NoOfFragmentLogFiles} = 6 \times 2048 / 1024 = 12$
3. The "6xDataMemory" is a good heuristic and is STRONGLY recommended.

NoOfFragmentLogFiles=12

1. RedoBuffer of 32M should let you restore/provisioning quite a lot of data in parallel.
2. If you still have problems ("out of redobuffer"), then you probably have to slow disks and
3. increasing this will not help, but only postpone the inevitable.

RedoBuffer=32M

1. table related things

MaxNoOfTables=4096 MaxNoOfAttributes=24756 MaxNoOfOrderedIndexes=2048

MaxNoOfUniqueHashIndexes=512

1. Operation records
2. MaxNoOfConcurrentOperations=100000 means that you can load any mysqldump file into cluster.

MaxNoOfConcurrentOperations=100000

1. Checkpointing...

Diskcheckpointspeed=10M Diskcheckpointspeedinrestart=100M
TimeBetweenGlobalCheckpoints=1000

1. the default value for TimeBetweenLocalCheckpoints is very good

TimeBetweenLocalCheckpoints=20

1. Realtime extensions (only in MySQL Cluster 6.3 (CGE 6.3) , read this how to use this)
2. SchedulerSpinTimer=400
3. SchedulerExecutionTimer=100
4. RealTimeScheduler=1
5. LockMaintThreadsToCPU=[cpuid]
6. LockExecuteThreadToCPU=[cpuid]

1. If you use MySQL Cluster 6.3 (CGE 6.3) and are tight on disk space, e.g ATCA.
2. You should also then lock cpu's to a particular core.
3. CompressedLCP=1
4. CompressedBackup=1

datadir=X

[ndb_mgmd] hostname=...

1. second management server for redundancy
2. [ndb_mgmd]
3. hostname=...

[ndbd] hostname=...

[ndbd] hostname=...

[mysqld]

...

[mysqld]

MySQL Cluster On Premisse

Télécharger https://dev.mysql.com/get/Downloads/MySQL-Cluster-8.0/mysql-cluster-8.0.27-linux-glibc2.12-x86_64.tar.gz

puis:

```
tar xvf mysql-cluster-8.0.27-linux-glibc2.12-x86_64.tar.gz
```

Pour un premier cluster, commencez avec un seul serveur MySQL (mysqld), une paire de nœuds de données (ndbd) et un seul nœud de gestion (ndb_mgmd) – tous exécutés sur le même serveur. Créez des dossiers pour stocker les fichiers de configuration et les fichiers de données :

```
mkdir my_cluster my_cluster/ndb_data my_cluster/mysqld_data my_cluster/conf
```

Dans le dossier conf, créez 2 fichiers (notez que "/home/user1" doit être remplacé par votre répertoire personnel).

my.cnf

```
[mysqld]
ndbcluster
datadir=/home/pilou/Formation/my_cluster/mysqld_data
basedir=/home/pilou/Formation/mysql-cluster-8.0.27-linux-glibc2.12-x86_64
log-error=/home/pilou/Formation/my_cluster/logdir/mysqld.log
log-bin=/home/pilou/Formation/my_cluster/logdir/mysqlbin.log
port=5000
```

config.ini

```
[ndb_mgmd]
hostname=localhost
datadir=/home/pilou/Formation/my_cluster/ndb_data
NodeId=1
```

```
[ ndbd default]
noofreplicas=2
datadir=/home/pilou/Formation/my_cluster/ndb_data
[ ndbd]
hostname=localhost
NodeId=3
[ ndbd]
hostname=localhost
NodeId=4
[ mysqld]
NodeId=50
```

Tout comme n'importe quel autre serveur MySQL, le processus mysqld nécessite qu'une base de données « mysql » soit créée et remplie avec les données système essentielles :

```
/home/pilou/Formation/mysql-cluster-8.0.27-linux-glibc2.12-x86_64/bin/mysqld --initialize-
insecure --datadir=/home/pilou/Formation/my_cluster/mysqld_data --log-
error=/home/pilou/Formation/my_cluster/logdir/error.log --basedir=/home/pilou/Formation/mysql-
cluster-8.0.27-linux-glibc2.12-x86_64
```

Les processus doivent être démarrés dans l'ordre du nœud de gestion, des nœuds de données, puis du serveur MySQL :

```
/home/pilou/Formation/mysql-cluster-8.0.27-linux-glibc2.12-x86_64/bin/ndb_mgmd
-f /home/pilou/Formation/my_cluster/conf/config.ini --initial --
configdir=/home/pilou/Formation/my_cluster/conf/
```

```
/home/pilou/Formation/mysql-cluster-8.0.27-linux-glibc2.12-x86_64/bin/ndbd -c localhost:1186
--foreground
/home/pilou/Formation/mysql-cluster-8.0.27-linux-glibc2.12-x86_64/bin/ndbd -c localhost:1186
--foreground
```

```
/home/pilou/Formation/mysql-cluster-8.0.27-linux-glibc2.12-x86_64/bin/mysqld --defaults-
file=/home/pilou/Formation/my_cluster/conf/my.cnf
```

```
ndb_mgm> show;
Cluster Configuration
-----
```

[ndbd(NDB)] 2 node(s)

id=3 @127.0.0.1 (mysql-8.0.27 ndb-8.0.27, Nodegroup: 0)

id=4 @127.0.0.1 (mysql-8.0.27 ndb-8.0.27, Nodegroup: 0, *)

[ndb_mgmd(MGM)] 1 node(s)

id=1 @127.0.0.1 (mysql-8.0.27 ndb-8.0.27)

[mysqld(API)] 1 node(s)

id=50 @127.0.0.1 (mysql-8.0.27 ndb-8.0.27)

Cluster Galera

Cluster Galera

Le clustering de bases de données est le processus consistant à combiner plusieurs serveurs en les connectant à une seule base de données. Le clustering améliore la disponibilité de votre base de données en répartissant la charge sur différents serveurs. Si un serveur tombe en panne, d'autres sont rapidement disponibles pour continuer à servir.

MariaDB Galera est une solution de clustering multi-maîtres qui vous permet de lire et d'écrire sur n'importe quel nœud du cluster. Avec MariaDB Galera, une modification apportée à un nœud est répliquée sur tous les nœuds. MariaDB Galera prend en charge les moteurs de stockage XtraDB/InnoDB et est disponible uniquement sur Linux.

Configurer le premier serveur

Tout d'abord, connectez-vous au premier serveur et créez un fichier de configuration Galera dans la partie mysqld :

20

1

```
[mysqld]
```

2

```
binlog_format=ROW
```

3

```
default-storage-engine=innodb
```

4

```
innodb_autoinc_lock_mode=2
```

5

```
bind-address=0.0.0.0
```

6

```
□
```

7

```
# Galera Provider Configuration
```

8

```
wsrep_on=ON
```

9

```
wsrep_provider=/usr/lib/galera/libgalera_smm.so
```

10

```
□
```

11

```
# Galera Cluster Configuration
```

12

```
wsrep_cluster_name="galera_cluster"
```

13

```
wsrep_cluster_address="gcomm://192.168.0.101,192.168.0.102,192.168.0.103"
```

14

```
□
```

15

```
# Galera Synchronization Configuration
```

16

```
wsrep_sst_method=rsync
```

17

```
□
```

18

```
# Galera Node Configuration
```

19

```
wsrep_node_address="192.168.0.101"
```

20

```
wsrep_node_name="server1"
```

Configurer le deuxième serveur

Le deuxième serveur est homogène au premier:

20

1

```
[mysqld]
```

2

```
binlog_format=ROW
```

3

```
default-storage-engine=innodb
```

4

```
innodb_autoinc_lock_mode=2
```

5

```
bind-address=0.0.0.0
```

6

```
□
```

7

```
# Galera Provider Configuration
```

8

```
wsrep_on=ON
```

9

```
wsrep_provider=/usr/lib/galera/libgalera_smm.so
```

10

```
□
```

11

```
# Galera Cluster Configuration
```

12

```
wsrep_cluster_name="galera_cluster"
```

13

```
wsrep_cluster_address="gcomm://192.168.0.101,192.168.0.102,192.168.0.103"
```

14

```
□
```

15

```
# Galera Synchronization Configuration
```

16

```
wsrep_sst_method=rsync
```

17

```
□
```

18

```
# Galera Node Configuration
```

19

```
wsrep_node_address="192.168.0.102"
```

20

```
wsrep_node_name="server2"
```

Démarrage du cluster

Démarrage du premier nœud et vérification

4

1

```
sudo galera_new_cluster
```

2

```
[root@mariadb01 my.cnf.d]# ps -ef | grep mysql
```

3

```
mysql 7604 1 4 15:12 ? 00:00:00 /usr/sbin/mysqld --wsrep-new-cluster --wsrep_start_position=0000
```

4

```
root 7650 3887 0 15:12 pts/1 00:00:00 grep --color=auto mysql
```

Vérification :

8

1

```
[root@mariadb01 ~]# mysql -u dba -p
```

2

```
MariaDB [(none)]> show status like '%wsrep_cluster_size';
```

3

```
+-----+-----+
```

4

```
| Variable_name | Value |
```

5

```
+-----+-----+
```

6

```
| wsrep_cluster_size | 1 |
```

7

```
+-----+-----+
```

8

```
1 row in set (0.00 sec)
```

Pour les autres nœuds, le démarrage se fait par la commande classique :

1

1

```
systemctl start mysqld
```

Safe-To-Bootstrap

Les clusters Galera sont généralement conçus pour fonctionner en continu, il n'est donc pas nécessaire d'arrêter l'ensemble du cluster pendant le fonctionnement normal. Pourtant, s'il est nécessaire d'effectuer une telle procédure, il est important qu'elle se termine en toute sécurité et le plus rapidement possible afin d'éviter les temps d'arrêt prolongés et la perte potentielle de données.

Galera 3.19 inclut deux améliorations importantes au redémarrage de l'ensemble du cluster : la protection « Safe-to-Bootstrap » et la récupération Gcache. Dans cet article, nous allons décrire la première fonctionnalité.

REDÉMARRAGE DE L'ENSEMBLE DU CLUSTER

Tout d'abord, quelques mots sur les redémarrages de cluster en général. Qu'il s'agisse d'un arrêt ordonné ou d'un crash soudain de tous les nœuds, le redémarrage de l'ensemble du cluster est régi par les principes suivants :

Étant donné que l'ancien cluster n'existe plus logiquement, un nouveau cluster logique est en cours de création

Le premier nœud en cours de démarrage doit être amorcé

Il est important de sélectionner le nœud qui a les dernières transactions validées comme premier nœud dans le nouveau cluster

LA PROTECTION SAFE-TO-BOOTSTRAP

Dans un arrêt ordonné, le nœud qui a été arrêté en dernier sera celui qui a la dernière transaction validée et doit être choisi comme premier nœud dans le nouveau cluster. La sélection d'un autre nœud pour ce rôle peut entraîner des erreurs sur la route et ouvrir la possibilité de perdre ces dernières transactions.

Pour faciliter cette décision et éviter les choix dangereux, Galera, à partir de la version 3.19, gardera une trace de l'ordre dans lequel les nœuds sont arrêtés. Le nœud qui a été arrêté en dernier sera marqué comme "Safe-to-Bootstrap". Tous les autres nœuds seront marqués comme dangereux pour l'amorçage.

Lors de l'amorçage du nouveau cluster, Galera refusera d'utiliser comme premier nœud un nœud qui a été marqué comme dangereux pour l'amorçage.

SÉLECTION DU BON NŒUD

La procédure pour sélectionner le bon nœud à partir duquel s'amorcer dépend de la façon dont le cluster s'est terminé : via un arrêt ordonné ou un crash.

En cas d'arrêt ordonné, il suffit de suivre les recommandations de la fonction « Safe-to-Bootstrap ». Recherchez le nœud dont `gratate.dat` a `safe_to_bootstrap : 1` :

```
# GALERA saved state
version: 2.1
uuid:    9acf4d34- acdb- 11e6- bcc3- d3e36276629f
seqno:   15
safe_to_bootstrap: 1
```

et utilisez ce nœud.

En cas de crash dur, tous les nœuds auront `safe_to_bootstrap: 0` , nous devons donc consulter le moteur de stockage InnoDB pour déterminer quel nœud a validé la dernière transaction dans le cluster. Ceci est réalisé en démarrant `mysqld` avec la variable `--wsrep-recover` , qui produit une sortie comme celle-ci :

```
...
2016-11-18 01:42:15 36311 [Note] InnoDB: Database was not shutdown normally!
2016-11-18 01:42:15 36311 [Note] InnoDB: Starting crash recovery.
...
2016-11-18 01:42:16 36311 [Note] WSREP: Recovered position: 37bb872a-ad73-11e6-819f-
f3b71d9c5ada: 345628
...
2016-11-18 01:42:17 36311 [Note] /home/pilou/git/mysql-wsrep-bugs-5.6/sql/mysqld: Shutdown
complete
```

Le nombre après la chaîne UUID sur la ligne « Position récupérée » est celui à surveiller. Choisissez le nœud qui a le nombre le plus élevé et modifiez son `gratate.dat` pour définir `safe_to_bootstrap : 1` :

```
# GALERA saved state
version: 2.1
uuid:    37bb872a-ad73-11e6-819f-f3b71d9c5ada
seqno:   -1
safe_to_bootstrap: 1
```

En faisant cela, vous indiquez à Galera que vous avez volontairement sélectionné ce nœud et cela vous permettra de démarrer à partir de celui-ci.

Reprise sur incident

Récupération sur incident

Contrairement à la réplication MySQL standard, un cluster Galera agit comme une entité logique, qui contrôle le statut et la cohérence de chaque nœud ainsi que le statut de l'ensemble du cluster. Cela permet de maintenir l'intégrité des données plus efficacement qu'avec la réplication asynchrone traditionnelle sans perdre les écritures sécurisées sur plusieurs nœuds en même temps.

Cependant, il existe des scénarios où le service de base de données peut s'arrêter sans qu'aucun nœud ne puisse répondre aux demandes.

Scénario : le nœud A est correctement arrêté

Dans un cluster à trois nœuds (nœud A, nœud B, nœud C), un nœud (nœud A, par exemple) est gracieusement arrêté : à des fins de maintenance, de changement de configuration, etc.

Dans ce cas, les autres nœuds reçoivent un message « au revoir » du nœud arrêté et la taille du cluster est réduite ; certaines propriétés comme le calcul du quorum ou l'incrémement automatique sont automatiquement modifiées. Dès que le nœud A est redémarré, il rejoint le cluster en fonction de sa `wsrep_cluster_address` dans `my.cnf`.

Si le cache d'écriture (`gcache.size`) sur les nœuds B et/ou C a encore toutes les transactions exécutées pendant que le nœud A était en panne, la jointure est possible via IST . Si IST est impossible en raison de transactions manquantes dans le `gcache` du donneur, la décision de secours est prise par le donneur et SST est démarré automatiquement.

Scénario : deux nœuds sont correctement arrêtés

Similaire au scénario : le nœud A est arrêté en douceur , la taille du cluster est réduite à 1 — même le seul nœud restant C forme le composant principal et est capable de répondre aux demandes des clients. Pour remettre les nœuds dans le cluster, il vous suffit de les démarrer.

Cependant, lorsqu'un nouveau nœud rejoint le cluster, le nœud C passera à l'état « Donateur/Désynchronisé » car il doit fournir le transfert d'état au moins au premier nœud qui se joint. Il est toujours possible d'y lire/écrire pendant ce processus, mais cela peut être beaucoup plus lent, ce qui dépend de la quantité de données à envoyer pendant le transfert d'état. En outre, certains équilibres de charge peuvent considérer le nœud donneur comme non opérationnel et le

supprimer du pool. Il est donc préférable d'éviter la situation où un seul nœud est actif.

Si vous redémarrez le nœud A puis le nœud B, vous voudrez peut-être vous assurer que la note B n'utilise pas le nœud A comme donneur de transfert d'état : le nœud A peut ne pas avoir tous les jeux d'écriture nécessaires dans son gcache. Spécifiez node C node comme donneur dans votre fichier de configuration et démarrez le service mysql :

```
$ systemctl démarrer mysql
```

Voir également

Documentation Galera : option `wsrep_sst_donor`

<https://galeracluster.com/library/documentation/mysql-wsrep-options.html#wsrep-sst-donor>

Scénario : Les trois nœuds sont correctement arrêtés

Le cluster est complètement arrêté et le problème est de l'initialiser à nouveau. Il est important qu'un nœud PXC écrive sa dernière position exécutée dans le `grastate.dat` fichier.

En comparant le numéro de `seqno` dans ce fichier, vous pouvez voir quel est le nœud le plus avancé (probablement le dernier arrêté). Le cluster doit être amorcé à l'aide de ce nœud, sinon les nœuds qui avaient une position plus avancée devront effectuer le SST complet pour rejoindre le cluster initialisé à partir du moins avancé. En conséquence, certaines transactions seront perdues). Pour amorcer le premier nœud, appelez le script de démarrage comme ceci :

```
$ systemctl démarrer mysql@bootstrap.service
```

Noter

Même si vous démarrez à partir du nœud le plus avancé, les autres nœuds ont un numéro de séquence inférieur. Ils devront toujours se joindre via le SST complet car le cache Galera n'est pas conservé au redémarrage.

Pour cette raison, il est recommandé d'arrêter les écritures sur le cluster avant son arrêt complet, afin que tous les nœuds puissent s'arrêter à la même position. Voir aussi `pc.recovery`.

Scénario : Un nœud disparaît du cluster

C'est le cas lorsqu'un nœud devient indisponible en raison d'une panne de courant, d'une panne matérielle, d'une panique du noyau, d'un crash `mysqld`, sur `mysqld pid`, etc. `kill -9`

Deux nœuds restants remarquent que la connexion au nœud A est interrompue et commencent à essayer de s'y reconnecter. Après plusieurs délais d'expiration, le nœud A est supprimé du cluster. Le quorum est enregistré (2 nœuds sur 3 sont actifs), donc aucune interruption de service ne se produit. Après son redémarrage, le nœud A se joint automatiquement (comme décrit dans Scénario : le nœud A est normalement arrêté).

Scénario : Deux nœuds disparaissent du cluster

Deux nœuds ne sont pas disponibles et le nœud restant (nœud C) n'est pas en mesure de former seul le quorum. Le cluster doit basculer vers un mode non principal, où MySQL refuse de servir les requêtes SQL. Dans cet état, le `mysqld` processus sur le nœud C est toujours en cours d'exécution et peut être connecté, mais toute instruction liée aux données échoue avec une erreur

```
mysql > sélectionnez * à partir de test . sbtest1 ;
```

ERREUR 1047 (08 S01): WSREP n'a pas encore préparé le nœud pour l' utilisation de l' application
Les lectures sont possibles jusqu'à ce que le nœud C décide qu'il ne peut pas accéder aux nœuds A et B. Les nouvelles écritures sont interdites.

Dès que les autres nœuds deviennent disponibles, le cluster se reforme automatiquement. Si le nœud B et le nœud C étaient simplement séparés par le réseau du nœud A, mais qu'ils peuvent toujours se joindre, ils continueront à fonctionner car ils forment toujours le quorum.

Si les nœuds A et B tombent en panne, vous devez activer manuellement le composant principal sur le nœud C avant de pouvoir afficher les nœuds A et B. La commande pour ce faire est :

```
mysql > SET GLOBAL wsrep_provider_options = 'pc.bootstrap=true' ;
```

Cette approche ne fonctionne que si les autres nœuds sont en panne avant de le faire ! Sinon, vous vous retrouvez avec deux clusters ayant des données différentes.

Références croisées

Ajout de nœuds au cluster

Scénario : tous les nœuds sont tombés en panne sans procédure d'arrêt appropriée

Ce scénario est possible en cas de panne de courant du datacenter ou en cas de bug MySQL ou Galera. En outre, cela peut se produire en raison de la cohérence des données compromise lorsque le cluster détecte que chaque nœud a des données différentes. Le `grastate.dat` fichier n'est pas mis à jour et ne contient pas de numéro de séquence valide (`seqno`). Cela peut ressembler à ceci :

```
$ cat /var/lib/mysql/grastate.dat
```

```
# État enregistré de GALERA
```

```
version : 2.1
```

```
uuid : 220dcdcb-1629-11e4-add3-aec059ad3734
```

```
numéro de séquence : -1
```

```
safe_to_bootstrap : 0
```

Dans ce cas, vous ne pouvez pas être sûr que tous les nœuds sont cohérents les uns avec les autres. Nous ne pouvons pas utiliser la variable `safe_to_bootstrap` pour déterminer le nœud qui a la dernière transaction validée car elle est définie sur 0 pour chaque nœud. Une tentative d'amorçage à partir d'un tel nœud échouera à moins que vous ne commenciez `mysqld` avec le `--wsrep-recover` paramètre :

```
$ mysqld --wsrep-recover
```

Recherchez dans la sortie la ligne qui signale la position récupérée après l'UUID du nœud (1122 dans ce cas) :

```
...
```

```
... [Remarque] WSREP : Position récupérée : 220dcdcb-1629-11e4-add3-aec059ad3734:1122
```

```
...
```

Le nœud où la position récupérée est marquée par le plus grand nombre est le meilleur candidat bootstrap. Dans son `grastate.dat` fichier, définissez la variable `safe_to_bootstrap` sur 1 . Ensuite,

amorcez à partir de ce nœud.

Noter

Après un arrêt, vous pouvez boosterrap à partir du nœud qui est marqué comme sûr dans le `grastate.dat` fichier.

...

```
safe_to_bootstrap : 1
```

...

Voir également

Documentation Galera

Présentation de la fonctionnalité "Safe-To-Bootstrap" dans Galera Cluster

Dans les versions récentes de Galera, l'option `pc.recovery` (activée par défaut) enregistre l'état du cluster dans un fichier nommé `gwwstate.dats` sur chaque nœud membre. Comme le nom de cette option l'indique (`pc` - composant principal), elle n'enregistre qu'un cluster étant dans l'état PRIMAIRE. Un exemple de contenu de : fichier peut ressembler à ceci :

```
cat /var/lib/mysql/gwwstate.dat
my_uuid : 76de8ad9-2aac-11e4-8089-d27fd06893b9
#vwbeg
view_id : 3 6c821ecc-2aac-11e4-85a5-56fe513c651f 3
bootstrap : 0
membre : 6c821ecc-2aac-11e4-85a5-56fe513c651f 0
membre : 6d80ec1b-2aac-11e4-8d1e-b2b2f6a5018ad 0
membre : 8089-d27fd06893b9 0
#vwend
```

Nous pouvons voir un cluster à trois nœuds avec tous les membres actifs. Grâce à cette nouvelle fonctionnalité, les nœuds tenteront de restaurer le composant principal une fois que tous les membres commenceront à se voir. Cela permet au cluster PXC de récupérer automatiquement après une mise hors tension sans aucune intervention manuelle ! Dans les logs nous verrons :

Scénario : Le cluster perd son état principal en raison d'un cerveau divisé

Pour les besoins de cet exemple, supposons que nous ayons un cluster composé d'un nombre pair de nœuds : six, par exemple. Trois d'entre eux se trouvent à un endroit tandis que les trois autres se trouvent à un autre endroit et ils perdent la connectivité réseau. Il est recommandé d'éviter une telle topologie : si vous ne pouvez pas avoir un nombre impair de nœuds réels, vous pouvez utiliser un nœud arbitre supplémentaire (`garbd`) ou définir un poids `pc` plus élevé pour certains nœuds. Mais lorsque le split brain se produit de quelque manière que ce soit, aucun des groupes séparés ne peut maintenir le quorum : tous les nœuds doivent cesser de répondre aux demandes et les deux parties du cluster essaieront en permanence de se reconnecter.

Si vous souhaitez restaurer le service avant même que le lien réseau ne soit restauré, vous pouvez à nouveau rendre l'un des groupes principal à l'aide de la même commande que celle décrite dans Scénario : deux nœuds disparaissent du cluster

```
SET GLOBAL wsrep_provider_options = 'pc.bootstrap=true' ;
```

Après cela, vous êtes en mesure de travailler sur la partie restaurée manuellement du cluster, et l'autre moitié devrait pouvoir se reconnecter automatiquement à l'aide d' IST dès que le lien réseau est restauré.

Avertissement

Si vous définissez l'option d'amorçage sur les deux parties séparées, vous vous retrouverez avec deux instances de cluster vivantes, avec des données susceptibles de diverger les unes des autres. La restauration d'un lien réseau dans ce cas ne les fera pas se rejoindre tant que les nœuds ne seront pas redémarrés et que les membres spécifiés dans le fichier de configuration ne seront pas à nouveau connectés.

Ensuite, comme le modèle de réplication Galera se soucie vraiment de la cohérence des données : une fois l'incohérence détectée, les nœuds qui ne peuvent pas exécuter l'instruction de changement de ligne en raison d'une différence de données - un arrêt d'urgence sera effectué et le seul moyen de ramener les nœuds dans le cluster est via le SST complet