

Compte rendu Galera Cluster

Date :
27/03/2023

COLBAUT Mathis
BTS SIO – Option SLAM

SOMMAIRE :



INTRODUCTION.....2



INSTALLATION ET MISE EN PLACE.....3



TEST DE GALERA CLUSTER.....6



CONCLUSION.....9



ANNEXE / PLAN RESEAU.....10



Qu'est-ce que Galera ?

Galera Cluster est un système de base de données distribué conçu pour offrir une haute disponibilité et une réplication synchrone à des applications critiques pour l'entreprise. Voici une procédure générale pour l'installation et l'utilisation de Galera Cluster sur 3 machines virtuelles Ubuntu.

Ce qui est appréciable avec cette solution, au-delà du fait qu'elle soit open source et gratuite, c'est qu'elle offre plusieurs avantages, notamment :

- **Cluster avec une topologie multi maîtres**, donc si un nœud tombe, les autres nœuds continuent d'assurer le service de façon transparente sans avoir besoin d'effectuer des manipulations complexes pour retrouver l'état initial (contrairement à *MySQL Replication*)
- **Cluster actif-actif** avec l'ensemble des nœuds
- **Réplication synchrone des informations**, c'est-à-dire en temps réel
- **Lecture et écriture** sur l'ensemble des nœuds
- **Souple** : fonctionne aussi bien sur le LAN que sur le WAN
- **Support des environnements géo-distribués** : plusieurs centres de données, multi cloud, etc.
- **Ajout d'un nouveau nœud** au cluster en quelques minutes (avec les bons paquets et un seul fichier de configuration)

Installation et mise en place



Avant de poursuivre, vérifiez que les trois machines virtuelles sont connectées au même réseau pour permettre une communication entre elles.

Configuration de l'infrastructure réseau :

Pour réaliser cette tâche, j'ai opté pour l'utilisation de trois machines virtuelles Ubuntu. Il est recommandé de configurer des adresses IP statiques pour chaque machine virtuelle.

Sous Ubuntu utiliser Net Plan afin d'y mettre en place les paramètres réseaux. Sur nos machines virtuelles, nous configurons également le nom d'hôtes ainsi que le pare-feu pour permettre les connexions de la base de données et les autres communications requises.

Les étapes suivantes expliquent comment installer le serveur MariaDB sur les différentes machines virtuelles.

```
sudo apt-get update  
  
sudo apt-get install mariadb-server  
  
mysql -u root -p
```

Ensuite, la requête suivante va m'afficher quel est le moteur de stockage utilisé par défaut sur cette instance :

```
show variables like 'default_storage_engine';
```

Selon la capture d'écran ci-dessous, la valeur retournée est "InnoDB". C'est une bonne nouvelle pour la suite de l'installation de notre cluster Galera.

```
root@LAMP1:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 72
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show variables like 'default_storage_engine';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| default_storage_engine | InnoDB |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [(none)]>
```

Nous allons maintenant procéder à l'installation du cluster Galera sur les trois machines virtuelles.

```
sudo apt-get install galera-4
```

Maintenant que Galera est installé, nous allons procéder à la modification du fichier de configuration sur les trois serveurs. Pour ce faire :

```
sudo nano /etc/mysql/mariadb.conf.d/60-galera.cnf
```

Lorsque vous accéderez au fichier à modifier, assurez-vous de remplacer les adresses IP existantes par celles de vos propres machines.

```
[galera]

# Mandatory settings
wsrep_on = ON
wsrep_provider = /usr/lib/galera/libgalera_smm.so
wsrep_cluster_name = "Galera_Cluster_Mathis"
wsrep_cluster_address = gcomm://192.168.100.50,192.168.100.51,192.168.100.52
binlog_format = row
default_storage_engine = InnoDB
innodb_autoinc_lock_mode = 2
innodb_force_primary_key = 1

# Allow server to accept connections on all interfaces.
bind-address = 0.0.0.0

# Optional settings
#wsrep_slave_threads = 1
#innodb_flush_log_at_trx_commit = 0
log_error = /var/log/mysql/error-galera.log
```

```
LAMP0 | LAMP1 | LAMP2 | HAPROXY |
GNU nano 6.2 /etc/mys
[galera]

# Mandatory settings
wsrep_on = ON
wsrep_provider = /usr/lib/galera/libgalera_smm.so
wsrep_cluster_name = "Galera_Cluster_Mathis"
wsrep_cluster_address = gcomm://192.168.100.50,192.168.100.51,192.168.100.52
binlog_format = row
default_storage_engine = InnoDB
innodb_autoinc_lock_mode = 2
innodb_force_primary_key = 1

# Allow server to accept connections on all interfaces.
bind-address = 0.0.0.0

# Optional settings
#wsrep_slave_threads = 1
#innodb_flush_log_at_trx_commit = 0
log_error = /var/log/mysql/error-galera.log
```

Effectuez la même modification sur vos trois serveurs et n'oubliez pas d'enregistrer en appuyant sur CTRL+X.

À présent, arrêtez MariaDB sur votre serveur principal.

```
sudo systemctl stop mariadb
```

Ensuite, démarrons le cluster Galera.

```
sudo galera_new_cluster
```

Nous allons maintenant vérifier le nombre de nœuds présents dans notre cluster. Normalement, il devrait y en avoir un seul.

```
mysql -u root -p
show status like 'wsrep_cluster_size';
```

Parfait, tout semble être en ordre pour nous. Nous avons effectivement un nœud dans notre cluster. Nous allons maintenant ajouter les deux autres nœuds.

```

root@Lamp0:~# sudo galera_new_cluster
root@Lamp0:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.6.12-MariaDB-Oubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show status like 'wsrep_cluster_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 1 |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [(none)]> 

```

Sur nos deux autres machines virtuelles, il suffit simplement de redémarrer les serveurs MariaDB (si vous avez bien effectué la modification de la configuration comme expliqué précédemment).

```
sudo systemctl restart mariadb
```

Si vous vérifiez à nouveau le nombre de nœuds, vous devriez constater qu'il y en a maintenant trois (vous pouvez le faire sur les trois nœuds et cela devrait toujours vous renvoyer trois).

```

root@Lamp0:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.6.12-MariaDB-Oubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show status like 'wsrep_cluster_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 3 |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [(none)]> 

```

Maintenant que la configuration du cluster est opérationnelle, nous allons procéder à une vérification physique. Vous pouvez utiliser la commande ci-dessous sur les trois serveurs pour vérifier si vos nœuds sont bien opérationnels entre eux.

```
show status like 'wsrep_local_state_comment';
```

```
MariaDB [(none)]> show status like 'wsrep_local_state_comment';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_local_state_comment | Synced |
+-----+-----+
1 row in set (0.001 sec)
```

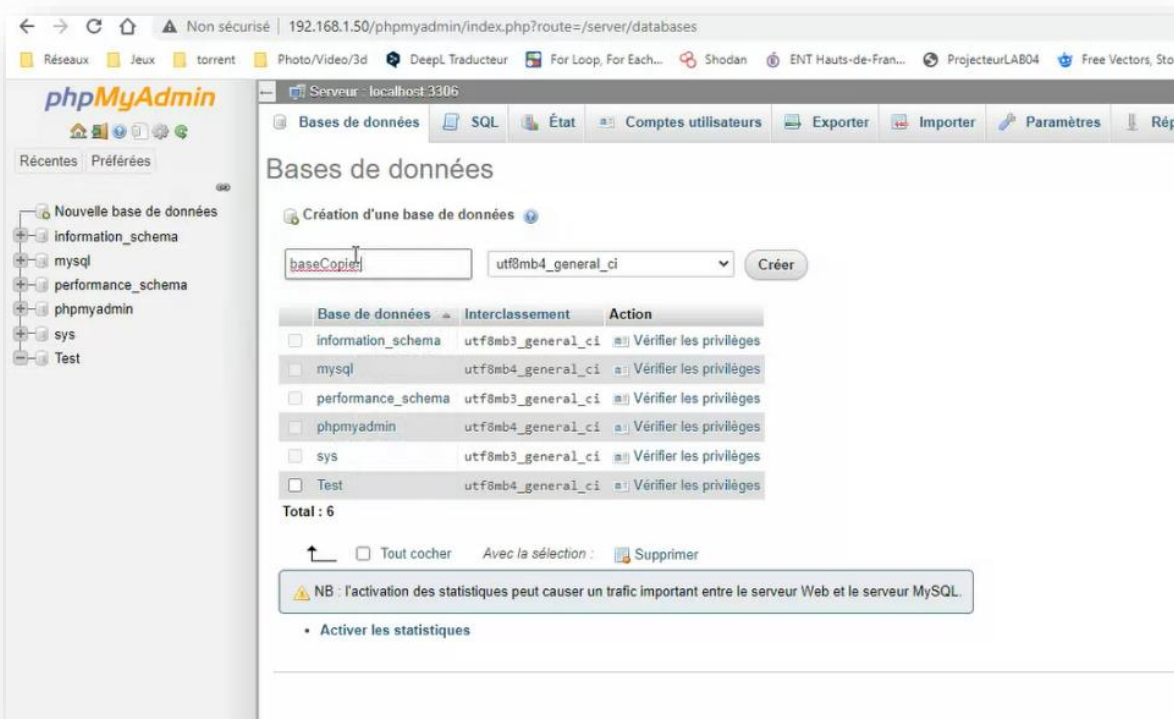
Arriver ici votre galera cluster est configuré, installé et fonctionnelle ce qui suis va vous montrer l'utilité de Galera cluster avec différent test.

Test de Galera Cluster

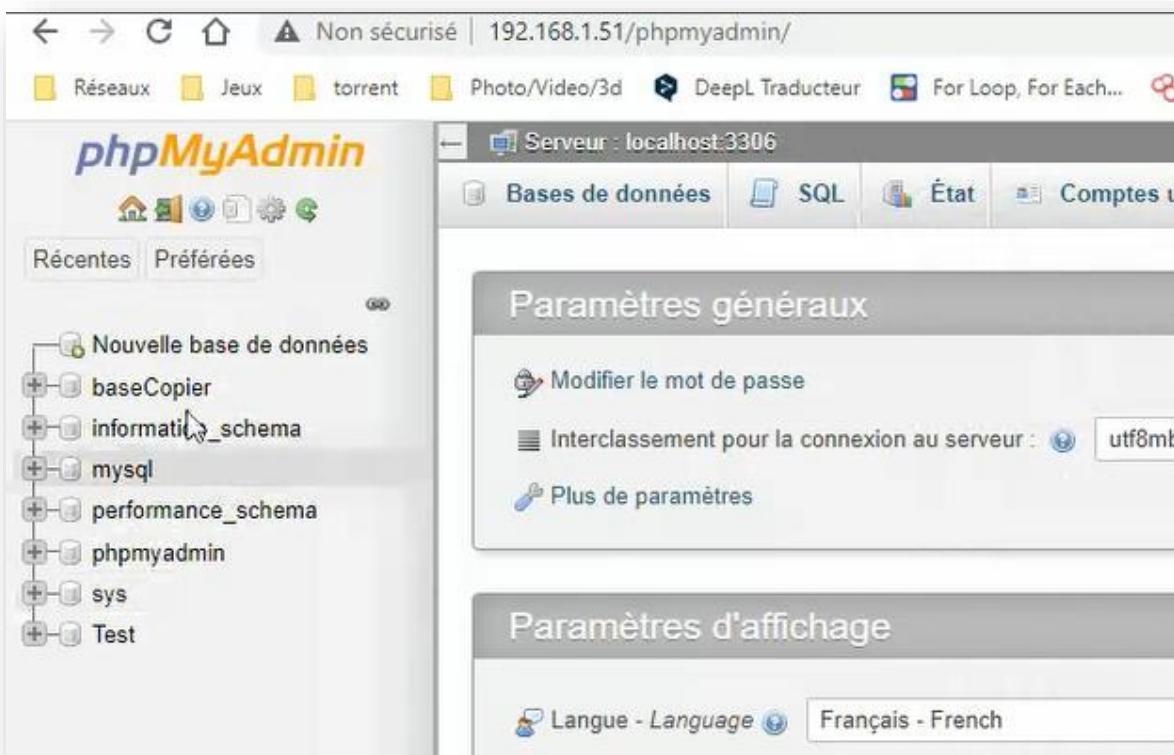


Le cluster Galera a été conçu pour assurer une haute disponibilité des données. Galera permet de répliquer toutes les bases de données d'un nœud primaire sur les nœuds secondaires, ce qui assure une haute disponibilité et une sécurité accrue des données. En effet, les données sont disponibles même en cas de panne d'un serveur, ce qui permet également de réaliser des sauvegardes. Maintenant que tout est installé et configuré correctement, il est temps de passer à la démonstration.

Nous allons créer une base de données sur l'un des serveurs MariaDB (nous avons ajouté PhpMyAdmin pour une meilleure visualisation).

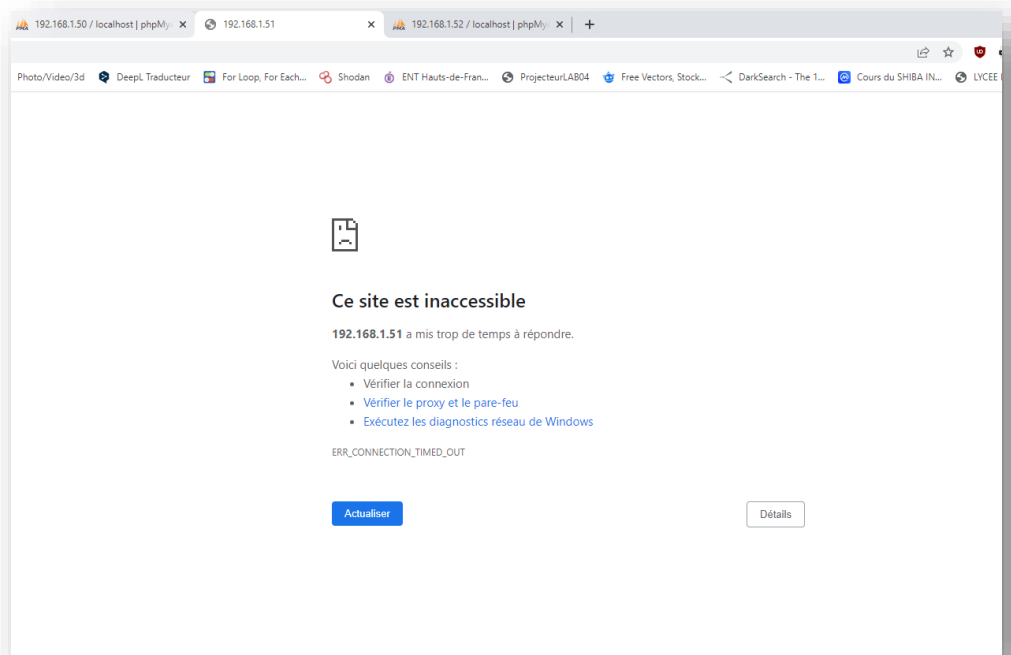
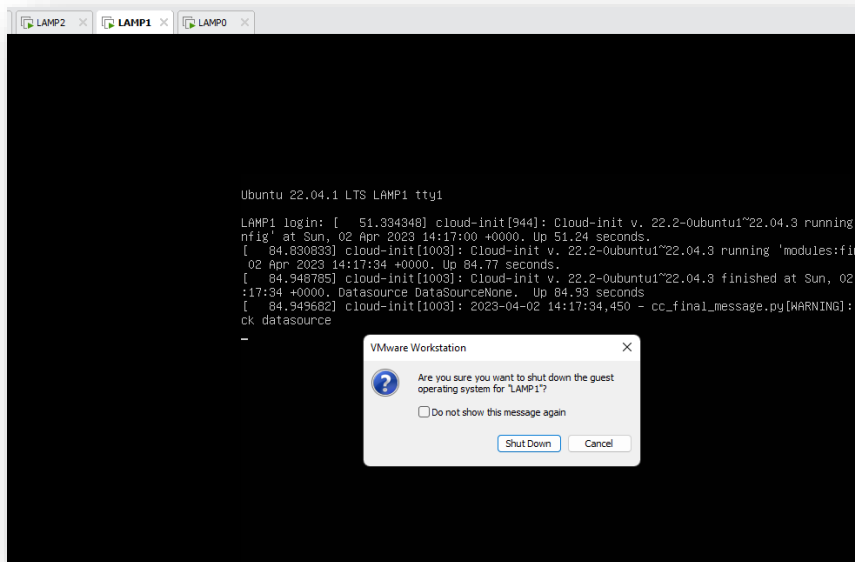


Comme vous pouvez le constater, la base de données créée sur le premier serveur n'apparaît pas sur le deuxième serveur, mais en actualisant, la base de données a bien été répliquée.

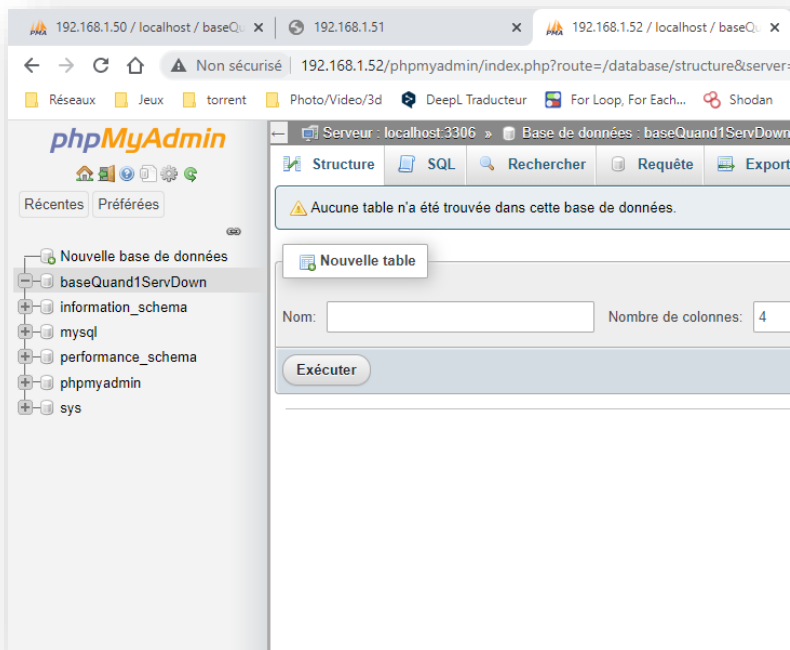
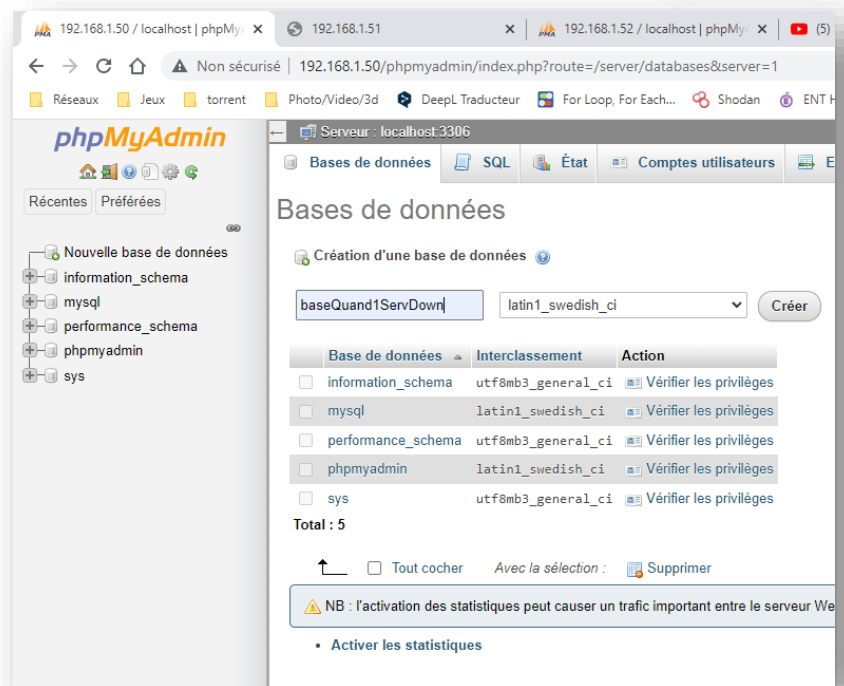


Nous avons vu que les bases de données se répliquent correctement, mais n'oublions pas que Galera est également conçu pour assurer une haute disponibilité des services, ce qui signifie que le système doit continuer à fonctionner même si un serveur tombe en panne.

Pour tester cela, nous allons éteindre l'une des machines virtuelles.



Très bien, nous allons donc créer une base de données sur le serveur 192.168.1.50.



D'après l'image ci-contre, nous pouvons constater que la réplication a bien été effectuée malgré la simulation de panne de l'une des machines virtuelles.

Comme vous pouvez le constater, la réplication sur le serveur qui été éteint a correctement fonctionné après son redémarrage.

Conclusion



En conclusion, nous avons pu installer et configurer avec succès un cluster Galera avec trois machines virtuelles sous Ubuntu et MariaDB.

Nous avons pu constater la haute disponibilité offerte par Galera Cluster grâce à la réplication en temps réel des données et même en cas de panne d'un serveur, tout a pu reprendre normalement après son redémarrage.

Ce système permet à la fois de sauvegarder les données et d'assurer une haute disponibilité, car en cas de panne d'un serveur, les autres prennent le relais pour assurer le bon fonctionnement du système.

Plan Réseau

