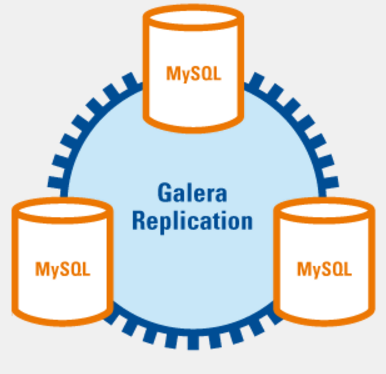
Deploy Galera Cluster

Trên MariaDB 10.0/10.1

# 1. Giới thiệu chung

Galera là một đồng bộ **multi-master** kiểu sync cho MariaDB. Hiện tại chỉ có bản trên Linux và chỉ hỗ trợ [**XtraDB/InnoDB** storage engines](https://mariadb.com/kb/en/mariadb/getting-started-with-mariadb-galera-cluster/).



Bài này sẽ hướng dẫn deploy một hệ thống Galera cluster với 3 nodes chạy trên CentOS 6.8 hoặc 7.0

**Ghi chú:**

* Câu lệnh sẽ được tô **màu xanh da trời**
* Tên riêng sẽ được tô **màu đỏ**

# 2. Các bước cấu hình

## 2.1. Thông số

IP Master 1: **103.231.188.230** (CentOS release 7.0 - 64 bits, MariaDB 10.1.17)

IP Master 2: **210.211.96.114** (CentOS release 6.8 - 64 bits, MariaDB 10.1.17)

IP Master 3: **AAA.BBB.CCC.DDD** (MariaDB 10.1.17)

MariaDB sử dụng trong bài viết có version 10.1

Để nâng cấp MariaDB từ version 10.0 lên 10.1, tham khảo bài viết số **[1]**

Galera wsrep provider version 25.3.15 **[3]**

## 2.2. Cài đặt các gói cần thiết

1. Sửa file ***/etc/yum.repos.d/mariadb.repo***, nếu chưa có thì tạo file mới

# MariaDB 10.1 CentOS repository

# http://downloads.mariadb.org/mariadb/repositories/

[mariadb]

name=MariaDB

baseurl=http://yum.mariadb.org/10.1/centos7-amd64

gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB

gpgcheck=1

Nếu cài trên hệ thống 32 bits, sửa centos-amd64 thành centos6-x86.

2. Đặt SELinux ở chế độ permissive

sudo setenforce 0

Hoặc *sed -i -e 's/SELINUX=permissive/SELINUX=disabled/g' /etc/sysconfig/selinux*

3. Cài đặt MariaDB 10.1

sudo yum install MariaDB-server MariaDB-client rsync galera

## 2.3. Cấu hình DB trên Master 1

1. Cài đặt MariaDB Security

sudo service mysql start

sudo /usr/bin/mysql\_secure\_installation

Chọn password cho root, giả sử là ‘*topica@123*’, các mục khác đều để mặc định (chọn Yes)

2. Tạo user cho Galera Cluster

Chúng ta sẽ tạo một user ‘sst\_user’ được sử dụng để authenticate in pha SST (State Transfer Snapshot)

mysql -u root -p

mysql> DELETE FROM mysql.user WHERE user='';

mysql> GRANT ALL ON \*.\* TO 'root'@'%' IDENTIFIED BY 'topica@123';

mysql> GRANT USAGE ON \*.\* to sst\_user@'%' IDENTIFIED BY 'topica@123';

mysql> GRANT ALL PRIVILEGES on \*.\* to sst\_user@'%';

mysql> FLUSH PRIVILEGES;

mysql> quit

Bạn có thể thay các dấu % bằng hostname hoặc địa chỉ IP mà được phép truy cập vào db.

3. Cấu hình Galera Cluster tại node này

Thêm vào file ***nano /etc/my.cnf.d/server.cnf*** những dòng sau:

# \* Galera-related settings

[galera]

# Mandatory settings

wsrep\_on=ON

wsrep\_provider=/usr/lib64/galera/libgalera\_smm.so

wsrep\_provider\_options="gcache.size=10G"

wsrep\_cluster\_address="gcomm://"

binlog\_format=row

default\_storage\_engine=InnoDB

innodb\_autoinc\_lock\_mode=2

# Allow server to accept connections on all interfaces.

bind-address=0.0.0.0

# Optional setting

#wsrep\_slave\_threads=1

#innodb\_flush\_log\_at\_trx\_commit=0

wsrep\_cluster\_name='galera\_cluster'

wsrep\_node\_address='103.231.188.230'

wsrep\_node\_name='cl230'

wsrep\_sst\_method=rsync

wsrep\_sst\_auth=sst\_user:topica@123

4. Mở cổng với firewalld trên CentOS 7

Theo mặc định, Galera Cluster sử dụng những cổng sau:

* **3306** for MySQL client connections and State Snapshot Transfer that use the mysqldump method
* **4567** for Galera Cluster replication traffic, multicast replication uses both UDP transport and TCP on this port
* **4568** for Incremental State Transfer
* **4444** for all other State Snapshot Transfer

Vì vậy chúng ta phải mở những cổng này, với những lệnh sau:

firewall-cmd --permanent --zone=public --add-port=3306/tcp

firewall-cmd --permanent --zone=public --add-port=4567/tcp

firewall-cmd --permanent --zone=public --add-port=4568/tcp

firewall-cmd --permanent --zone=public --add-port=4444/tcp

firewall-cmd --reload

5. Khởi động node đầu tiên cho Galera Cluster

sudo /etc/init.d/mysql start --wsrep-new-cluster

Bạn có thể check lại xem Galera Cluster đã hoạt động chưa bằng cách

mysql -u root -p

mysql> show status like 'wsrep%';

Nếu thấy hiện ra các thông số như sau thì quá trình tạo node đầu tiên cho Galera Cluster đã thành công

wsrep\_local\_state\_comment | Synced <= cluster is synced

wsrep\_incoming\_addresses | 103.231.188.230:3306 <= node db1 is a provider

wsrep\_cluster\_size | 1 <= cluster consists of 1 node

wsrep\_ready | ON <= good

## 2.4. Cấu hình DB trên Master 2

1. Giống bước 1 của phần 2.3 (Cấu hình DB trên Master 1)

2. Giống bước 2 của phần 2.3 (Cấu hình DB trên Master 1)

3. Cấu hình Galera Cluster tại node này

Thêm vào file ***nano /etc/my.cnf.d/server.cnf*** những dòng sau:

# \* Galera-related settings

[galera]

# Mandatory settings

wsrep\_on=ON

wsrep\_provider=/usr/lib64/galera/libgalera\_smm.so

wsrep\_provider\_options="gcache.size=10G"

wsrep\_cluster\_address="gcomm://103.231.188.230,210.211.96.114"

binlog\_format=row

default\_storage\_engine=InnoDB

innodb\_autoinc\_lock\_mode=2

# Allow server to accept connections on all interfaces.

bind-address=0.0.0.0

# Optional setting

#wsrep\_slave\_threads=1

#innodb\_flush\_log\_at\_trx\_commit=0

wsrep\_cluster\_name='galera\_cluster'

wsrep\_node\_address='210.211.96.114'

wsrep\_node\_name='cl114'

wsrep\_sst\_method=rsync

wsrep\_sst\_auth=sst\_user:topica@123

4. Mở cổng với iptables trên CentOS 6.8

iptables -A INPUT -p tcp -m tcp --dport 3306 -j ACCEPT

iptables -A INPUT -p tcp -m tcp --dport 4567 -j ACCEPT

iptables -A INPUT -p tcp -m tcp --dport 4568 -j ACCEPT

iptables -A INPUT -p tcp -m tcp --dport 4444 -j ACCEPT

service iptables save

service iptables restart

5. Khởi động node thứ 2

sudo service mysql start

Bạn cũng có thể dùng lệnh

mysql -u root -p

mysql> show status like 'wsrep%';

để kiểm tra xem node 2 này đã join vào được cluster của chúng ta hay không?

Nếu kết quả trả về như sau thì chúng ta đã cấu hình thành công.

| wsrep\_local\_state\_comment | Synced |

| wsrep\_incoming\_addre sses | 210.211.96.114:3306,103.231.188.230:3306 |

| wsrep\_cluster\_size | 2 <= cluster consists of 2 nodes |

| wsrep\_connected | ON |

| wsrep\_ready | ON |

## 2.6. Join thêm node mới vào Galera Cluster

Để thêm một node mới (node thứ 3,4, ...), bạn làm các bước giống như phần 2.4. Giả sử node 3 có ip là **AAA.BBB.CCC.DDD**

Chú ý thay đổi một số dòng sau trong file config ***/etc/my.cnf.d/server.cnf***

wsrep\_cluster\_address="gcomm://103.231.188.230, AAA.BBB.CCC.DDD"

wsrep\_node\_address='AAA.BBB.CCC.DDD'

wsrep\_node\_name='<tên\_node>'

## 2.7. Tăng tốc truy cập vào DB

Thêm vào file **/etc/my.cnf.d/server.cnf**:

[mysqld]

max\_connections=1024

table\_open\_cache=2048

max\_allowed\_packet=500M

max\_heap\_table\_size=512M

read\_buffer\_size=16M

read\_rnd\_buffer\_size=32M

sort\_buffer\_size=16M

join\_buffer\_size=128M

query\_cache\_size=64M

tmp\_table\_size=512M

query\_cache\_limit=16M

expire\_logs\_days=3

key\_buffer\_size=32M

myisam\_sort\_buffer\_size=128M

myisam\_max\_sort\_file\_size=40G

innodb\_buffer\_pool\_size=4G

innodb\_lock\_wait\_timeout=120

[mysqldump]

max\_allowed\_packet=16M

[myisamchk]

key\_buffer\_size=512M

sort\_buffer\_size=512M

read\_buffer=8M

write\_buffer=8M

[mysqlhotcopy]

interactive-timeout

wait\_timeout=30

connect\_timeout=30

[mysqld\_safe]

open-files-limit=8192

local-infile=0

# 3. Giải pháp đồng bộ cho dữ liệu DB phân tán

## 3.1. Yêu cầu đặt ra

Bài toán đặt ra là muốn đồng bộ dữ liệu trên các DB phân tán trên các server khác nhau với CHỈ MỘT DB trên một server.

## 3.2. Giải pháp dùng Galera với nhiều Clusters

Như hình minh họa dưới đây, chúng ta có 3 DB phân tán trên 2 server khác nhau và muốn đồng bộ với DB trên server thứ 4.

Chúng ta sẽ tạo nhiều Maria instance trên server 4.

Một khó khăn gặp phải là Galera sử dụng 4 cổng (3306, 4444, 4567, 4568) để thực hiện đồng bộ nên nếu chúng ta tạo 3 instances của MariaDB trên server 4, sẽ có sự xung đột cổng.



Hai giải pháp đưa ra là [18]:

1. Sử dụng MySQL Sandbox
2. Sử dụng nhiều Docker container

[[22](https://www.pythian.com/blog/best-practice-setting-mysql-replication-filters/)] MySQL cung cấp 3 levels của bộ lọc (filters) để setup replication:

1. **Binlog**: bộ lọc binlog áp dụng trên master để kiểm soát cách ghi log những sự thay đổi. Do đồng bộ MySQL dựa trên binlog, nên nó là bộ lọc level đầu tiên và có độ ưu tiên cao nhất
2. **DB**: Áp dụng trên slave. Có độ ưu tiên cao hơn bộ lọc cấp Table.
3. **Table**: Áp dụng trên slave. Bên trong bộ lọc cấp Table, MySQL sẽ evaluate một vài lựa chọn theo thứ tự:
   1. *replicate-do-table*
   2. *replicate-ignore-table*
   3. *replicate-wild-do-table*
   4. *replicate-wild-ignore-table*

Một vài gợi ý:

1. Không nên cài đặt bộ lọc cấp binlog trừ khi bạn thực sự muốn và can afford losing the chance of having an extra full copy of data changes for the master.
2. Ở cấp lọc DB, sử dụng một trong hai lựa chọn sau (không sử dụng cùng lúc): *replicate-do-db* or *replicate-ignore-db*

For MariaDB replication filters within Galera cluster, it should be used with caution. As a general rule except for InnoDB DML updates, những bộ lọc sau đây are not honored in a Galera cluster (cấp binlog và table): *binlog-do-db, binlog-ignore-db, replicate-wild-do-db, replicate-wild-ignore-db.*

Tuy nhiên, bộ lọc cấp DB *replicate-do-db,replicate-ignore-db* are honored for DDL and DML for both InnoDB & MyISAM engines.

# Link tham khảo

1. [**Upgrade MariaDB from 10.0 to 10.1**](https://support.qualityunit.com/009608-Upgrade-MariaDB-100x-to-MariaDB-101x-on-Centos-yum)
2. [**Setup Galera Cluster with MariaDB 10.0 in CentOS**](https://www.unixmen.com/setup-mariadb-galera-cluster-10-0-centos/)
3. [Getting started with Galera Cluster in MariaDB](https://mariadb.com/kb/en/mariadb/getting-started-with-mariadb-galera-cluster/)
4. [Installation of Galera Cluster in MariaDB](http://tunnelix.com/mariadb-galera-cluster-installation/)
5. [MySql Multi-Master Replication with Galera](https://www.sebastien-han.fr/blog/2012/04/01/mysql-multi-master-replication-with-galera/)
6. [Galera Replication Overview](https://mariadb.com/kb/en/mariadb/about-galera-replication/)
7. [Documentation of Galera Cluster](http://galeracluster.com/documentation-webpages/?id=mysql_galera_configuration)
8. [Galera cluster, MariaDB, CoreOS and Docker](http://withblue.ink/2016/03/09/galera-cluster-mariadb-coreos-and-docker-part-1.html)
9. [Wiki about Replication in general](https://en.wikipedia.org/wiki/Replication_(computing)#DATABASE)
10. [Tutorial of Galera Cluster in MySql](http://severalnines.com/tutorials/clustercontrol-mysql-galera-tutorial)
11. [Replication in MongoDB](https://docs.mongodb.com/manual/replication/)
12. [Terms and Definitions for Database Replication](http://www.postgres-r.org/documentation/terms)
13. [How to setup HAProxy as Load Balancer for MariaDB on CentOS 7](https://www.howtoforge.com/tutorial/how-to-setup-haproxy-as-load-balancer-for-mariadb-on-centos-7/)
14. [Slide] [Galera Cluster Installation and Quick Start Webinar](http://www.slideshare.net/Codership/coderships-galera-cluster-installation-and-quickstart-webinar-march-2016-59361687)
15. [Video] [Migrating to Multi-Master Galera Cluster from Master-Slave MySQL Replication](https://www.youtube.com/watch?v=u7H7C12vRpU)
16. [Video] [Galera Cluster Installation and Quick Start Webinar](https://www.youtube.com/watch?v=YX3rn5Z5ToM)
17. [Article] *Database replication: a tale of research across communities,* Kemme, Bettina and Alonso, Gustavo, 2010
18. [Running Multiple Copies of MariaDB](https://mariadb.com/kb/en/mariadb/running-multiple-copies-of-mariadb/)
19. [Installing and using MariaDB via Docker](https://mariadb.com/kb/en/mariadb/installing-and-using-mariadb-via-docker/)
20. [REPLICATION CHANNEL FAIL-OVER WITH GALERA CLUSTER FOR MYSQL](http://www.fromdual.com/replication-channel-fail-over-with-galera-cluster-for-mysql)
21. [How To Configure a Galera Cluster with MariaDB 10.1 on Ubuntu 16.04 Servers](https://www.digitalocean.com/community/tutorials/how-to-configure-a-galera-cluster-with-mariadb-10-1-on-ubuntu-16-04-servers)
22. [BEST PRACTICE FOR SETTING UP MYSQL REPLICATION FILTERS](https://www.pythian.com/blog/best-practice-setting-mysql-replication-filters/)
23. [List of MYSQL WSREP OPTIONS](http://galeracluster.com/documentation-webpages/mysqlwsrepoptions.html)