1. **何谓Galera Cluster**

何谓Galera Cluster？就是集成了Galera插件的MySQL集群，是一种新型的，数据不共享的，高度冗余的高可用方案，目前Galera Cluster有两个版本，分别是Percona Xtradb Cluster及MariaDB Cluster，都是基于Galera的，所以这里都统称为Galera Cluster了，因为Galera本身是具有多主特性的，所以Galera Cluster也就是multi-master的集群架构，如图1所示：



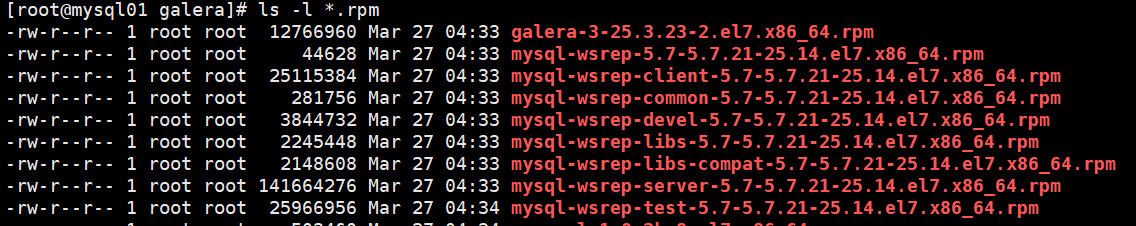
图1 Galera Cluster架构

图1中有三个实例，组成了一个集群，而这三个节点与普通的主从架构不同，它们都可以作为主节点，三个节点是对等的，这种一般称为multi-master架构，当有客户端要写入或者读取数据时，随便连接哪个实例都是一样的，读到的数据是相同的，写入某一个节点之后，集群自己会将新数据同步到其它节点上面，这种架构不共享任何数据，是一种高冗余架构。

一般的使用方法是，在这个集群上面，再搭建一个中间层，这个中间层的功能包括建立连接、管理连接池，负责使三个实例的负载基本平衡，负责在客户端与实例的连接断开之后重连，也可以负责读写分离（在机器性能不同的情况下可以做这样的优化）等等，使用这个中间层之后，由于这三个实例的架构在客户端方面是透明的，客户端只需要指定这个集群的数据源地址，连接到中间层即可，中间层会负责客户端与服务器实例连接的传递工作，由于这个架构支持多点写入，所以完全避免了主从复制经常出现的数据不一致的问题，从而可以做到主从读写切换的高度优雅，在不影响用户的情况下，离线维护等工作，MySQL的高可用，从此开始，非常完美。

1. **开发dockerfile支持docker部署Percona Xtradb Cluster**

下载Mysql galera cluster安装相关的rpm包



下载galera3

<http://releases.galeracluster.com/galera-3/centos/7/x86_64/galera-3-25.3.23-2.el7.x86_64.rpm>

下载mysql-wsrep-5.7.21-25.14

<http://releases.galeracluster.com/mysql-wsrep-5.7.21-25.14/centos/7/x86_64/>

1. **dockerfile for Percona Xtradb Cluster**

from centos

#workdir /usr/galera/

user root

COPY my.cnf /etc/mysql/my.cnf

#引入一个基础的centos镜像

add \*.rpm /

RUN yum -y install gcc-c++ make git perl libaio libaio-devel lsof iproute iproute-devel rsync which perl-DBI perl-Data-Dumper vim-enhanced less openssl openssl-devel boost boost-devel

RUN echo "$PWD"&&yes|yum localinstall -y /\*.rpm&&rm -rf /var/log/mysql&&mkdir -p /var/logs/&&chmod 777 /var/logs/

copy libgalera\_smm.so /usr/lib/galera/

copy mysql-first-time.sql /tmp

add entrypoint.sh /

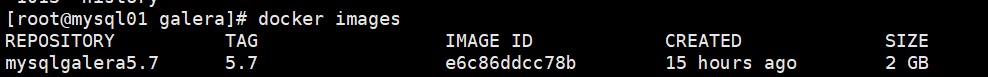
RUN chmod +x /entrypoint.sh

expose 3306

ENTRYPOINT ["/entrypoint.sh"]

1. **编译docker images for Percona Xtradb Cluster**

docker build -t mggarela:5.7 -f /usr/galera/DockerFile.txt



1. **将编译好的images拷贝至其他两个节点**
2. **Nodea节点启动**

docker run --env MYSQL\_ROOT\_PASSWORD=123 -p3339:3306 -v mysql\_galera:/var/lib/mysql -v mysql\_log:/var/logs/ -p 4567:4567 -p 4444:4444 -p 4568:4568 -d mysqlgalera5.7:5.7 --wsrep-cluster-address=gcomm:// --wsrep-node-address=192.168.0.116

**Nodeb节点启动**

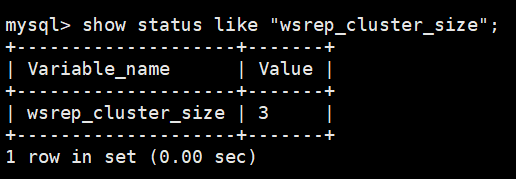
docker run --env MYSQL\_ROOT\_PASSWORD=123 -p3339:3306 -v mysql\_galera:/var/lib/mysql -v mysql\_log:/var/logs/ -p 4567:4567 -p 4444:4444 -p 4568:4568 -d mysqlgalera5.7:5.7 --wsrep-cluster-address=gcomm://192.168.0.116:4567 --wsrep-node-address=192.168.0.165

**Nodec节点启动**

docker run --env MYSQL\_ROOT\_PASSWORD=123 -p3339:3306 -v mysql\_galera:/var/lib/mysql -v mysql\_log:/var/logs/ -p 4567:4567 -p 4444:4444 -p 4568:4568 -d mysqlgalera5.7:5.7 --wsrep-cluster-address=gcomm://192.168.0.116:4567 --wsrep-node-address=192.168.0.115

1. **验证是否安装集群成功，docker登录任一节点**

mysql客户端查看集群数量状态，如果集群节点数等于状态显示节点数，则表示机群搭建正常



选择任一节点创建数据库

create database hellodb;

再查看nodea,nodeb，发现马上出现了一个数据库hellodb。

show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| hellodb | //数据库复制成功

| mysql |

| performance\_schema | |

+--------------------+

nodeb上创建表，并添加数据：

create table students (id int unsigned auto\_increment not null primary key,name char(30) not null,age tinyint unsigned,gender enum('f','m'));

insert into students (name) values ("Hello");

nodea,nodeb,nodec上分别查看：

select \* from students;

都显示：

+----+-------+------+--------+

| id | name | age | gender |

+----+-------+------+--------+

| 2 | Hello | NULL | NULL |

//如果字段时自动增加的话，则时按集群的节点数显示下一个数据

+----+-------+------+--------+

**故障恢复**

将其中一台（nodeb）关闭

docker stop nodeb

再插入数据：

insert into students (name) values ("World"); //一节点插入数据

nodea, nodec上显示：

+----+-------+------+--------+

| id | name | age | gender |

+----+-------+------+--------+

| 2 | Hello | NULL | NULL |

+----+-------+------+--------+

| 4 | World | NULL | NULL |

+----+-------+------+--------+

开启nodeb上的mysql：

docker start nodeb

再次查看数据，发现已经同步过来了。

select \* from students;

+----+-------+------+--------+

| id | name | age | gender |

+----+-------+------+--------+

| 2 | Hello | NULL | NULL |

+----+-------+------+--------+

| 4 | World | NULL | NULL |

+----+-------+------+--------+