[How-To] Configure Galera Cluster on MariaDB Servers

Purpose

The purpose of this How-To is to explain the process in detail of how to take 3 MariaDB servers and cluster them together in a Multi-Master Galera cluster. For best results, it is recommended to have a minimum of 3 MariaDB servers to achieve quorum and proper redundancy. If you only cluster 2 MariaDB servers together and 1 becomes unavailable, it renders the remaining server unusable. For production environments, minimum of 3 servers and recommended 5 or more.

Prerequisites

List of prerequisites:

- Sudo user
- Minimum 3 Ubuntu 24.04 LTS VMs with MariaDB installed and configured properly (See below for instructions):
    - https://wiki.stretchpowered.com/books/mariadb/page/how-to-install-mariadb-on-ubuntu-2404-lts

Instructions

Step 1: Stop the MariaDB Service on all Servers

First step is to stop the MariaDB service during configuration of Galera as to not get weird results with setup. Run this on all MariaDB servers that will be in the cluster:

```
sudo systemctl stop mariadb
```

Step 2: Make a Backup of the MariaDB Config File on all Servers

Next important step is to make a copy of the MariaDB config file on all affected servers in case you need to go backwards. You can also achieve this with snapshots in your hypervisor:

```
sudo cp /etc/mysql/mariadb.conf.d/50-server.cnf /etc/mysql/mariadb.conf.d/50-server.cnf.bak
```

Step 3: Prepare the MariaDB Config File for Galera Cluster Settings

This step is where we actually configure the cluster. This will need to be configured on all affected MariaDB servers but is going to look slightly different on each one of them. So lets edit the config file. On each server you can access the config file by using the following command:

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Inside the config file, the following should be done on each server, adjusting ips and names of servers to be each one:

```
[mysqld]

# Basic MariaDB Settings
bind-address = 0.0.0.0
default_storage_engine = InnoDB
binlog_format = ROW
innodb_autoinc_lock_mode = 2
# query_cache_type=0
# query_cache_size=0
# log_slave_updates

# Galera Cluster Settings
wsrep_on = ON
wsrep_provider = /usr/lib/galera/libgalera_smm.so
wsrep_cluster_name = "my_galera_cluster"
wsrep_cluster_address = "gcomm://<IP-OF-DB-NODE1>,<IP-OF-DB-NODE2>,<IP-OF-DB-NODE3>"

# Node-specific Settings
wsrep_node_name = "<NAME-OF-THIS-NODE>"
wsrep_node_address = "<IP-OF-THIS-NODE>"

# SST (State Snapshot Transfer) method
wsrep_sst_method = rsync
wsrep_sst_auth = "sst_user:secure_password" # Configure in later step
```

With this config saved, you should be prepared for the Galera cluster.

Step 4: Configure SST User

We need to take a moment to configure the user that we specified for SST in the config for Galera. Run the following commands to do so:

```
sudo mariadb -u root -p
```

```
CREATE USER 'sst_user'@'%' IDENTIFIED BY 'secure_password';
```

```
GRANT RELOAD, LOCK TABLES, PROCESS, REPLICATION CLIENT ON *.* TO 'sst_user'@'%';
```

```
FLUSH PRIVILEGES;
```

```
QUIT;
```

This should be now operation for SST. Continue on to firewall steps.

Step 5: Add UFW Firewall Rules for Galera

Now that our Galera cluster is prepped, we need to open some ports on the VMs UFW firewall to allow for the Galera cluster traffic. Run the following on each MariaDB server in the Galera cluster:

```
sudo ufw allow 4444/tcp
```

```
sudo ufw allow 4567/tcp
```

```
sudo ufw allow 4568/tcp
```

```
sudo ufw reload
```

Step 6: Start the Galera Cluster

Our servers are prepped and ready to go. Time to start the cluster up. On your first node, run the following (This is not the name of your cluster, it is a built in command so don't edit this):

```
sudo galera_new_cluster
```

Now, check the cluster status by logging into MariaDB and checking the `wsrep_cluster_size` value:

```
sudo mariadb -u root -p
```

```
SHOW STATUS LIKE 'wsrep_cluster_size';
```

It should show 1, as we've only started the first. Now, on all remain nodes, run the following to start MariaDB normally:

```
sudo systemctl start mariadb
```

Now, on any of the nodes you have, run the following commands to check cluster size:

```
sudo mariadb -u root -p
```

```
SHOW STATUS LIKE 'wsrep_cluster_size';
```

If you have three nodes, it should return a 3.

Step 7: Verify Functionality and Replication

Everything should be working now so we can test by creating a test database on any of the servers. Remember, Galera is designed to be a Multi-Master clustering solution for MariaDB and allows all nodes to simultaneously be read-write active. So, on any one of your nodes, run the following:

```
sudo mariadb -u root -p
```

```
CREATE DATABASE testdb;
```

```
SHOW DATABASES;
```

You should see your DB. Now, pick another node in the cluster and run the following:

```
sudo mariadb -u root -p
```

```
SHOW DATABASES;
```

You should see your new DB have replicated to the other node. This wraps up this How-To. The next step is to get a Virtual IP that can be used to access your DBs. This can be done with keepalived or haproxy.

Revision #4
Created 6 December 2024 05:47:49 by Mike Leffring
Updated 6 December 2024 06:51:21 by Mike Leffring