Restoring a Crashed or Stopped MySQL Galera Cluster

Target Audience

This document is designed to help Technical Support or a customer's IT or Engineering department to restore the database cluster to operation if the primary group of nodes is lost. A basic knowledge of Linux and terminal access of a Linux system is assumed.

This document should be used if all nodes in a cluster where offline at the same time due to a VM infrastructure crash, power interruption to all nodes or a switch failure interrupts all node communication.

Bootstrapping a node or a cluster is the act of forcing one node in the cluster to accept connections from other nodes by telling the bootstrapped node it has valid data and should form the 'Primary' or 'Synced' part of the cluster. As other nodes join and sync to the cluster the primary part will increase in size by the number of nodes in a synced state.

Requirements

- The servers are running CentOS 7.
- You must know the MySQL root user password.
- Terminal or SSH access of the servers is required.
- Nano editor is installed on each database node in the cluster.
  - › Run the following command to install the nano editor: `sudo yum install nano`
- You are using a user with sudo permissions on the computer or you are running as the root user.

Cluster Recovery

The following sections contain the procedures to recover your MySQL Galera cluster from a full cluster crash or shutdown.

Determining the Most Advanced Database Node

After a full cluster crash you must determine which database node in the cluster is most advanced node so you can use it to bootstrap the cluster. To determine the most advanced database node in the cluster you must check the grastate.dat file on each node. The grastate.dat file for each database node is stored in the folder path defined in the my.cnf file on the node.

To determine the most advanced database node

1. Locate the `grastate.dat` file by completing the following steps on each node in the cluster:

   a. Login in to a node of the MySQL Galera cluster.

   b. Run the following command to stop MySQL on all nodes in the cluster:

   ```
   sudo systemctl stop mysqld
   ```

   c. Run the following command to view the `my.conf` file:

   ```
   sudo cat /etc/my.cnf
   ```

d. Look for the following line in the `my.conf` file:

```
datadir=<folder path>
```

For example:

```
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
```

In the example the full data directory path to the `grastate.dat` file is `/var/lib/mysql/grastate.dat`. The full path to the `grastate.dat` file may vary on your system.

2. Run the following command to open the `grastate.dat` file in the nano editor on the current node:

```
sudo nano <folder path>
```

Where `<folder path>` is the full data directory path to the `grastate.dat` file that found in step **d** on page 4-2. For example:

```
sudo nano /var/lib/mysql/grastate.dat
```

The full path to the `grastate.dat` file may vary on your system.

3. In the open `grastate.dat` file, check the sequence number. For exmaple:

```
# GALERA saved state
version: 2.1
uuid:    25e3460b-1516-11ea-a7a4-8f6185551d7c8
sequo:   -1
safe_to_bootstrap: 0
```

Interpret the `grastate.dat` file as follows:

- A `-1` sequence number (`seqno`) indicates that the node crashed before MySQL could write the current sequence number. Complete step **5** on page 4-2 to find the sequence number of the last transition.

- A `uuid` of all `0` (zeros) indicates that the node has not yet synced or has dropped out of sync with the other nodes in the cluster. Do not use a node with `0` (zero) `uuid` unless you are starting a new cluster.

- When `safe_to_bootstrap` is `1` the current node was the last node to go down and is the node you will use to bootstrap your cluster. Skip to the procedure "**To bootstrap the cluster**" on page 4–3.

4. Press **Ctrl X** and follow the prompts at the bottom of the screen to save and the `grastate.dat` file and exit the nano text editor.

5. When your `grastate.dat` file contains a sequence number (`seqno`) of -1, run the following command to find the sequence number of the last committed transaction:

```
mysqld -wsrep-recover
```

The entered command writes the currently known sequence number (`seqno`) to the log and then quits. For example:

```
2020-03-17T12:26:12.005991Z 0 [Note] WSREP: Binlog recovery, found wsrep position
25e3460b-1516-11ea-a7a4-8f6185511d7c8:10863896
```

The sequence number (`seqno`) is to the right of the last `:` (colon). The example sequence number (`seqno`) is `10863896`.

Bootstrapping the Cluster

After determining the most advanced database node in your cluster, you can use the node to bootstrap your cluster.

To bootstrap the cluster

1. Log in to node that you found to be the most advanced in the cluster.

2. Use a text editor to edit the `grastate.dat` file.

3. In the `grastate.dat` file, set the `safe_to_bootstrap` parameter to `1` as follows.

   ```
   safe_to_bootstrap: 1
   ```

4. Save and close the updated `grastate.dat` file.

5. Run the following command to use the current node to bootstrap the cluster:

   ```
   sudo /usr/bin/mysqld_bootstrap
   ```

   After MySQL starts without any errors and the command prompt returns, you ready to start the other database nodes in the cluster.

Starting Remaining Nodes

After bootstrapping the most advance database node in the cluster you must start the remaining database nodes in the cluster.

To start a the remaining database nodes

1. Log in to a node to start.

2. Run the following command to start the current node:

   ```
   sudo systemctl start mysqld
   ```

3. After MySQL starts and the command prompt returns, run the following command to check the node status:

   ```
   mysql -u root -p -e "show status like 'wsrep_local_state%'"
   ```

   Before you move on to the next step the results table must display a `wrep_local_state value` of `4` or `wrep_local_state_comment value` of `Synced`. For example:

   ```
   +-------------------------+-------------------------------------+
   | Variable_name           | Value                               |
   +-------------------------+-------------------------------------+
   | wrep_local_state_uuid   | 25e3460b-1516-11ea-a7a4-8f6185551d7c8 |
   | wrep_local_state        | 4                                   |
   | wrep_local_state_comment | Synced                             |
   +-------------------------------------------------------------+
   ```

4. Verify that results table displays a **wrep_local_state value** of **4** or **wrep_local_state_comment** value of **Synced** before continuing this procedure.

5. Repeat steps **1** to **4** of this procedure on each remaining node in the cluster.

   The database is ready to use after all the nodes have started and synced.

Performing a Rolling Restart

A rolling restart is used when changes to MySQL or upgrades to the database server need to be applied and a restart of either MySQL or the server itself is required. A rolling restart is done to keep the database up and responding to served applications. There will be some user impact as the application nodes can take a moment to move on to another database node.

To perform a rolling restart

1. Log in to the first node to restart.

2. Run the following command to start the current node:

   ```
   sudo systemctl start mysqld
   ```

   If you need to restart the host system, run the following command:

   ```
   sudo restart
   ```

3. After the restart, run the following command to check the node status:

   ```
   mysql -u root -p -e "show status like 'wsrep_local_state%'"
   ```

   Before you move on to the next step the results table must display a `wrep_local_state value` of `4` or `wrep_local_state_comment value` of `Synced`. For example:

   ```
   +--------------------------+--------------------------------------+
   | Variable_name            | Value                                |
   +--------------------------+--------------------------------------+
   | wrep_local_state_uuid    | 25e3460b-1516-11ea-a7a4-8f6185551d7c8 |
   | wrep_local_state         | 4                                    |
   | wrep_local_state_comment | Synced                               |
   +--------------------------+--------------------------------------+
   ```

4. Verify that results table displays a **wrep_local_state value** of **4** or **wrep_local_state_comment** value of **Synced** before continuing this procedure.

5. Repeat steps **1** to **4** of this procedure on each remaining node in the cluster.

   The cluster is in good state after all the nodes are synced.

Contacting Technical Support

Technical Support is staffed by a team of experienced specialists ready to assist you with any question or technical issue.

Ross Video has technical support specialists strategically located around the globe to ensure a prompt response to technical inquiries. Our primary technical support center is located in Ottawa, Ontario, Canada. In addition, we have offices in The United Kingdom (London), Australia (Sydney), and Singapore with satellite locations in New York City, The Netherlands, and China. As we expand our presence globally, we are constantly evaluating other key locations to have a local technical support specialist in order to better service our customers.

North America

Our North America center located in Ottawa, Ontario, Canada and is open Monday to Friday 8:30 a.m. to 6:00 p.m. EST, with 24/7/365 on-call service after hours.

Our telephone number is: +1-613-686-1557

Toll free within North America: +1 844-652-0645

EMEA

Our EMEA center is open Monday to Friday 8:30 a.m. to 5:00 p.m. GMT. After hours support is provided by our North America location.

Our telephone number is: +44 (0)1189502446

International toll free: +800 3540 3545

If the local support specialist is not available, your call will be transferred automatically to our North America center.

Australia

Our Sydney, Australia office is located in Alexandria, NSW.

Our local support telephone number is: 1300 007 677

If the local support specialist is not available, your call will be transferred automatically to our North America center.

Online

E-mail: techsupport@rossvideo.com

Website: open a support request using the link http://www.rossvideo.com/support/tech-support.html to open a support request.

Copyright