Creating a multi-master MariaDB cluster

This topic will cover the process to create a two node Maria DB cluster. The cluster will be configured to allow Soffid console to use either database node, which in turn will replicate data changes to the other one.

<table>
<tr><th>Node 1 action</th><th>Node 2 action</th></tr>
<tr><td>Create and setup a Maria DB in node 1.</td><td></td></tr>
<tr><td>Configure Maria DB to generate binary log files. Add the following lines to /etc/mysql/my.cnf:

server-id = 1
log-bin
binlog-format=row
expire_logs_days = 15
max_binlog_size = 1000M
replicate-ignore-table = soffid.SC_SEQUENCE
slave-skip-errors = 1032,1053,1062</td><td></td></tr>
<tr><td>Restart MariaDB:

<code>service mysql restart</code></td><td></td></tr>
<tr><td></td><td>Create and setup a Maria DB in node 2.</td></tr>
<tr><td></td><td>Configure Maria DB to generate binary log files. Add the following lines to /etc/mysql/my.conf:

server-id = 2
log-bin
binlog-format=row
expire_logs_days = 15
max_binlog_size = 1000M
replicate-ignore-table = soffid.SC_SEQUENCE
slave-skip-errors = 1032,1053,1062</td></tr>
<tr><td></td><td>Restart MariaDB:

<code>service mysql restart</code></td></tr>
<tr><td></td><td>Create a user for node 1 to fetch data from node 2. From mysql, execute:

grant replication slave on *.* to replication_user@<NODE1-IP>

set password for replication_user@1<NODE1-IP> = password ('<NODE1-PASS>')</td></tr>
<tr><td>Create a user for node 2 to fetch data from node 1. From mysql, execute:

grant replication slave on *.* to replication_user@<NODE2-IP>

set password for replication_user@1<NODE2-IP> = password('<NODE2-PASS>')</td><td></td></tr>
<tr><td>Query current binary log position:

MariaDB [(none)]> show master status;

The result should look like this:

<table><tr><th>File</th><th>Position</th><th>Binlog_Do_DB</th><th>Binlog_Ignore_DB</th></tr><tr><td>mysqld-bin.000030</td><td>68175</td><td></td><td></td></tr></table>
The got values will be used on node 2 to start replica process.</td><td></td></tr>
<tr><td></td><td>Start replication from node 1 to node 2. From mysql, execute the following sentence, replacing proper values:

CHANGE MASTER TO
MASTER_HOST='<NODE1-IP>',
MASTER_USER='replication_user',
MASTER_PASSWORD='<NODE2-PASS>',
MASTER_PORT=3306,
MASTER_LOG_FILE='<NODE1-FILE>' , /** i.e. mysql-bin.000030 **/
MASTER_LOG_POS=<NODE1-POSITION>, /** i.e. 68175 **/
MASTER_CONNECT_RETRY=10;</td></tr>
</table>

	Verify replica is working right, by executing *SHOW SLAVE STATUS \G* Check following lines: Slave_IO_Running: Yes Slave_SQL_Running: Yes Seconds_Behind_Master: 0
	Query current binary log position: MariaDB [(none)]> *show master status;* The result should look like this: **File** | **Position** | **Binlog_Do_DB** | **Binlog_Ignore_DB** mysqld-bin.000060 | 98325 | | The got values will be used on node 1 to start replica process.
Now, start replication from node 2 to node 1. From mysql, execute the following sentence, replacing proper values: *CHANGE MASTER TO* *MASTER_HOST='<NODE2-IP>',* *MASTER_USER='replication_user',* *MASTER_PASSWORD='<NODE1-PASS>',* *MASTER_PORT=3306,* *MASTER_LOG_FILE='<NODE2-FILE>', /** i.e. mysql-bin.000060 **/* *MASTER_LOG_POS=<NODE2-POSITION>, /** i.e. 98325 **/* *MASTER_CONNECT_RETRY=10;*	
Verify replica is working right, by executing *SHOW SLAVE STATUS \G* Check following lines: Slave_IO_Running: Yes Slave_SQL_Running: Yes Seconds_Behind_Master: 0	
Now, create and start SC_SEQUENCE table in node 1. This sequence will generate values 1, 11, 21, 31, 41, and so on: *CREATE TABLE `SC_SEQUENCE` (* *`SEQ_NEXT` bigint(20) NOT NULL,* *`SEQ_CACHE` bigint(20) NOT NULL,* *`SEQ_INCREMENT` bigint(20) NOT NULL* *);* *INSERT INTO SC_SEQUENCE VALUES (1, 100, 10);*	
	Now, create and start SC_SEQUENCE table in node 2. This sequence will generate values 2, 12, 22, 32, 42, and so on:: *CREATE TABLE `SC_SEQUENCE` (* *`SEQ_NEXT` bigint(20) NOT NULL,* *`SEQ_CACHE` bigint(20) NOT NULL,* *`SEQ_INCREMENT` bigint(20) NOT NULL* *);* *INSERT INTO SC_SEQUENCE VALUES (2, 100, 10);*
Now, install **heartbeat** to create a floating IP address to connect Soffid console to database. Create **/etc/ha.d/ha.cf** file: *autojoin none* *bcast eth0* *warntime 3* *deadtime 6* *initdead 60* *keepalive 1* *node <NODE1-NAME>* *node <NODE2-NAME>* *crm respawn*	
Create security token in node 1: *(echo -ne "auth 1\n1 sha1\n"; dd if=/dev/urandom bs=512 count=1 | openssl sha1 | cut --delimiter=' ' --fields=2) >* ***/etc/ha.d/authkeys*** ***chmod 600 /etc/ha.d/authkeys***	

	Copy both files to node 2: **/etc/ha.d/ha.cf** and **/etc/ha.d/authkeys**
Restart heartbeat service Restart pacemater service	
	Restart heartbeat service Restart pacemater service
Check cluster status executing *crm_mon -1* It should look like: *Last updated: Mon Dec 26 19:52:24 2016* *Last change: Wed Oct 21 15:11:31 2015 via cibadmin on logpmgid01v* *Stack: heartbeat* *Current DC: <node 1 name> - partition with quorum* *Version: 1.1.10-42f2063* *2 Nodes configured* *0 Resources configured* *Online: [<node 1 name> <node 2 name>]*	
	Check cluster status executing *crm_mon -1* It should look like: *Last updated: Mon Dec 26 19:52:24 2016* *Last change: Wed Oct 21 15:11:31 2015 via cibadmin on logpmgid01v* *Stack: heartbeat* *Current DC: <node 1 name>- partition with quorum* *Version: 1.1.10-42f2063* *2 Nodes configured* *0 Resources configured* *Online: [<node 1 name> <node 2 name>]*
Disable stonith: crm configure property stonith-enabled=false Add floating IP to the cluster: *crm configure* *crm(live)configure# primitive site_one_ip IPaddr params ip=<FLOATING-IP>* *cidr_netmask="<NETMASK>" nic="eth0"* *crm(live)configure# location site_one_ip_pref site_one_ip 100: <NODE1-NAME>* *crm(live)configure# commit* *crm(live)configure# exit*	
Check floating IP is up and bound to node 1	