

<http://www.coincoin.fr.eu.org/?Installing-MariaDB-Galera-Cluster>



Installing MariaDB Galera Cluster on Debian/Ubuntu

- 6- Webographie -

Date de mise en ligne : mardi 7 janvier 2014

Copyright © L'Imp'Rock Scénette (by @_daffyduke_) - Tous droits réservés

This is a HOWTO about installing MariaDB Galera Cluster on Debian/Ubuntu.

It is because a lot of people had problems installing MariaDB Galera Cluster.

In the end *elenst* from [#maria](#) on freenode forced me to write a Howto :)

You will find out, installing MariaDB Galera Cluster is in fact quite easy and some kind of boring in the end.

This Howto is written (tested) on Debian 7.1 (Wheezy) and Ubuntu 12.04 (Precise).

What we need

In our setup we assume 3 nodes (node01, node02, node03) with one interface.

We assume following IP addresses : 172.16.8.5, 172.16.8.6 and 172.16.8.4.

We need three packages installed on all nodes :

- rsync
- galera
- mariadb-galera-server

As Galera does not ship with the distribution repositories,
go for the [repo configurator](#) and follow the instructions to include the
repository fitting your system.

Keep in mind to Choose "5.5" in Step 3 (Choose a Version). Doing this you can jump directly to *Install Packages*

Including Repository

Alternatively you can just take following steps.

Debian Wheezy

```
# apt-get install python-software-properties # apt-key adv --recv-keys --keyserver keyserver.ubuntu.com  
0xc9cb082a1bb943db # add-apt-repository 'deb http://mirror3.layerjet.com/mariadb/repo/5.5/debian wheezy  
main' # apt-get update
```

Ubuntu Precise

```
# apt-get install python-software-properties # apt-key adv --recv-keys --keyserver keyserver.ubuntu.com  
0xc9cb082a1bb943db # add-apt-repository 'deb http://mirror3.layerjet.com/mariadb/repo/5.5/ubuntu precise  
main' # apt-get update
```

Yes, they are nearly the same :)

Install Packages

(Just another shortcut for the impatient)

```
# DEBIAN_FRONTEND=noninteractive apt-get install -y rsync galera mariadb-galera-server
```

After you installed the packages you have a running MariaDB/MySQL on each node.
But none of them is configured to run in/as a Galera Cluster.

Configuring Galera

So we got to do some configuration next.

There is a MySQL/MariaDB configuration part and one part to configure Galera (starting with `wsrep_`).

As we do the most basic and simple installation in this Howto, it is sufficient you just change the IP's
(Remember : 172.16.8.5, 172.16.8.6, 172.16.8.4) with your IP's.

This will be needed to define the `wsrep_cluster_address` Variable. The list of nodes a starting mysql contacts to get into the cluster.

The following configuration file has to be distributed on all nodes.

We use a separate configuration file `/etc/mysql/conf.d/galera.cnf` with following settings :

```
[mysqld] #mysql settings binlog_format=ROW default-storage-engine=innodb innodb_autoinc_lock_mode=2
query_cache_size=0 query_cache_type=0 bind-address=0.0.0.0 #galera settings
wsrep_provider=/usr/lib/galera/libgalera_smm.so wsrep_cluster_name="my_wsrep_cluster"
wsrep_cluster_address="gcomm://172.16.8.5,172.16.8.6,172.16.8.4" wsrep_sst_method=rsync
```

FYI : The shared library for `wsrep_provider` is provided by the installed galera package.

We could also change the cluster name by changing the value of `wsrep_cluster_name` to fit our style.

This setting also works as a shared secret to control the access to the cluster.

With `wsrep_cluster_address` you see the IP addresses of our setup.

The `wsrep_sst_method` tells what method to use to synchronise the nodes. While there are also `mysqldump` and `xtrabackup` available.

I prefer rsync because it is easy to configure (i.e. it does not need any credentials set on the nodes).

There will be a time you are going to consider using the xtrabackup method. Don't forget to install xtrabackup then.

Starting the Galera Cluster

Let us stop mysql on all nodes.

```
node01# service mysql stop node02# service mysql stop node03# service mysql stop
```

As we got the configuration file (`galera.cnf`) already distributed to all nodes, we start the first mysqld. This node initialises/starts the cluster (creates a GTID).

```
node01# service mysql start --wsrep-new-cluster
```

So just have a look if everything really worked well be checking the cluster size.

```
node01# mysql -u root -e 'SELECT VARIABLE_VALUE as "cluster size" FROM
INFORMATION_SCHEMA.GLOBAL_STATUS WHERE VARIABLE_NAME="wsrep_cluster_size"'+-----+ |
cluster size | +-----+ | 1 | +-----+
```

Ok great thats what we would expect.

Ok now as the Cluster already exists, let the next nodes just start and join the cluster.

```
node2# service mysql start [ ok ] Starting MariaDB database server: mysqld . . . . . [info]
Checking for corrupt, not cleanly closed and upgrade needing tables.. node01:/home/debian# ERROR 1045
(28000): Access denied for user 'debian-sys-maint'@'localhost' (using password: YES)
```

We ignore the error for now. This node is still starting fine.

Let's do a quick check. As we run a cluster it is not important if we check on `node01` or `node02`.

```
node01# mysql -u root -e 'SELECT VARIABLE_VALUE as "cluster size" FROM
INFORMATION_SCHEMA.GLOBAL_STATUS WHERE VARIABLE_NAME="wsrep_cluster_size"'+-----+ |
cluster size | +-----+ | 2 | +-----+
```

Very nice. Now lets start the third node :

```
node3# # service mysql start [ ok ] Starting MariaDB database server: mysqld . . . . . [info]
Checking for corrupt, not cleanly closed and upgrade needing tables.. node03:/home/debian# ERROR 1045
(28000): Access denied for user 'debian-sys-maint'@'localhost' (using password: YES) node03# mysql -u root
-e 'SELECT VARIABLE_VALUE as "cluster size" FROM INFORMATION_SCHEMA.GLOBAL_STATUS WHERE
VARIABLE_NAME="wsrep_cluster_size"'+-----+ | cluster size | +-----+ | 3 |
+-----+
```

Ok we are finished. We got a running MariaDB Galera Cluster \o/

Having fun with Debian/Ubuntu init scripts

But we've got to fix some things because Debian/Ubuntu oddities.

As we saw already the error while starting the node.

What happened ?

Debian/Ubuntu uses a special user (`'debian-sys-maint'@'localhost'`) in there init script.

Where the credentials for that user are stored in `/etc/mysql/debian.cnf`.

This user is used to make some checks starting MySQL. Checks I don't think belong into a service script anyway.

We could simply ignore it, but the user user is also used to shutdown the mysqld.

This is also not required, as a SIGTERM is sufficient to shutdown the mysqld :/

As we copied the data from `node01` to all other nodes, the credentials in `/etc/mysql/debian.cnf` are not fitting on `node02` and `node03`.

Thats why we will not be able to shutdown mysql on any of these nodes.

```
node02# service mysql stop [FAIL] Stopping MariaDB database server: mysqld failed!
```

So we've got to fix it, by copying `/etc/mysql/debian.cnf` from the first node (`node01`) to all other nodes.

So data and configuration file have the same data again.

After that we are able to shutdown the daemon again :

```
node02# service mysql stop [ ok ] Stopping MariaDB database server: mysqld.
```

Great.

So if we would have a proper init script the Howto would have been even shorter ;)

Follow the [Bug](#) :)

Ok enjoy your MariaDB Galera Cluster and have fun !

Erkan Yanar

Thx to teuto.net for providing me an OpenStack tenant, so I can run the tests for that Howto.

PlanetMySQL Voting :

[Vote UP](#) /

[Vote DOWN](#)

Cet article est repris du site http://linsenraum.de/erkules_int/20...