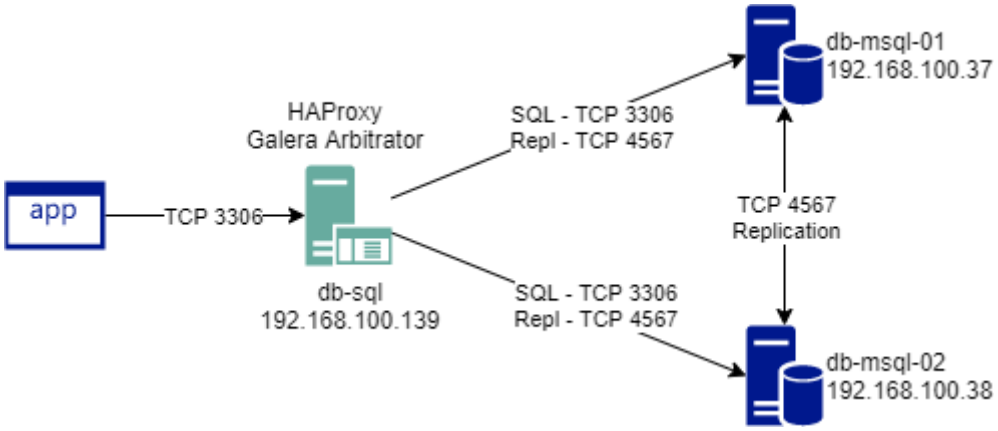
Кластер MariaDB (Galera Cluster)

Установка будет производиться на Ubuntu 20.04 LTS.
Для небольшой инсталляции будет использовано две ноды MariaDB, а для создания нечетного количества нод будет добавлен арбитр - сервис garbd (https://galeracluster.com/library/documentation/arbitrator.html)
Если планируется использовать 3 ноды репликации, то арбитр не требуется.



Сервис арбитража будет установлен на ноду балансировки нагрузки. В качестве балансировщика используется haproxy.
В данном примере будет задействован только один балансировщик, он же будет являться единственной точкой отказа. Для серьезных инсталляций лучше использовать аппаратный балансировщик (типа Citrix Netscaler), либо несколько экземпляров haproxy, на которые в свою очередь ставиться keepalived демон, который создает общий виртуальный IP адрес.

Наименование узла	**IP-адрес**	**Примечание**
db-msql-01	192.168.100.37	узел MariaBD
db-msql-02	192.168.100.38	узел MariaBD
db-sql	192.168.100.39	Балансировщик и арбитр

Установка Maria DB и включение механизма репликации

Перед инсталляцией необходимо обновить ОС и пакеты на **всех** узлах:

```
apt update && apt upgrade -yy
```

Если в сети отсутствует DNS-сервер, то прописываем на каждом узле соответствие имен серверов их IP-адресам:

```
nano /etc/hosts
```

```
192.168.100.37 db-msql-01.virtlab.local db-msql-01
192.168.100.38 db-msql-02.virtlab.local db-msql-02
192.168.100.39 db-sql.virtlab.local     db-sql
```

Установка Maria DB на нодах **db-msql-01** и **db-msql-02**:

```
apt-get install mariadb-server -y
```

Запускаем первоначальную настройку на нодах **db-msql-01** и **db-msql-02**:

```
mysql_secure_installation
```

```
Enter current password for root (enter for none):
Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Remove anonymous users? [Y/n] y
Disallow root login remotely? [Y/n]
Remove test database and access to it? [Y/n]
Reload privilege tables now? [Y/n] y
```

Создадим и отредактируем файл конфигурации galera.cnf на нодах **db-msql-01** и **db-msql-02**

```
nano /etc/mysql/conf.d/galera.cnf
```

| galera.cnf

```
[mysqld]
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0

# Galera Provider Configuration
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so

# Galera Cluster Configuration
wsrep_cluster_name="galera_cluster" #указываем наименование кластера
wsrep_cluster_address="gcomm://192.168.100.37,192.168.100.38" #сюда вписываем через запятую IP/имена всех узлов (MariaDB) кластера

# Galera Synchronization Configuration
wsrep_sst_method=rsync

# Galera Node Configuration
wsrep_node_address="192.168.100.37" # для второй ноды указываем соответственно 192.168.100.38
wsrep_node_name="db-msql-01" # для второй ноды указываем соответственно db-msql-02
```

На нодах db-msql-01 и db-msql-02 останавливаем службу mariadb:

```
systemctl stop mariadb
```

На первой ноде db-msql-01 запустить скрипт создания кластера galera:

```
galera_new_cluster
```

На второй ноде db-msql-02 запустить службу mariadb:

```
systemctl start mariadb
```

На любом из узлов db-msql-01 или db-msql-02 выполнить проверку количества узлов в кластере (количество должно быть равно 2):

```
root@db-msql-01:~# mysql -u root -p -e "SHOW STATUS LIKE
'wsrep_cluster_size'"
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 2     |
+--------------------+-------+
```

Устанавливаем службу арбитрации на узле db-sql:

```
apt-get install galera-arbitrator-4
```

Редактируем конфигурационный файл /etc/default/garb:

| garb

```
GALERA_NODES="192.168.100.37,192.168.100.38" # указываем через запятую узлы кластера
GALERA_GROUP="galera_cluster" # указываем наименование кластера
```

Запускаем службу garbd:

```
systemctl status garbd
```

Если все сделали правильно, то при повторе запроса на размер кластера получим увеличение количества до 3:

```
root@db-msql-01:~# mysql -u root -p -e "SHOW STATUS LIKE
'wsrep_cluster_size'"
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 3     |
+--------------------+-------+
```

Установка haproxy и настройка балансировки

На узлах db-msql-01 и db-msql-02 необходимо отредактировать значение параметра **bind-address** в файле конфигурации **/etc/mysql/mariadb.conf.d/50-server.cnf** для того, что бы демон mariadb принимал запросы не только с локального интерфейса, но из других сетей:

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
```

После необходимо перезапустить службу mariadb:

```
systemctl restart mariadb.service
```

Далее на любой из нод необходимо выполнить вход в sql-кдлиент и создать пользователя для haproxy:

```
root@db-msql-01:~# mysql -u root -p
Enter password:
MariaDB [(none)]> CREATE USER 'haproxy'@'192.168.100.39';
MariaDB [(none)]> FLUSH PRIVILEGES;
```

На узле db-sql устанавливаем пакет haproxy:

```
apt-get install haproxy
```

В конфигурационный файл **/etc/haproxy/haproxy.cfg** добавляем блок:

| haproxy.cfg

```
listen galera
     bind 192.168.100.39:3306 # вешаем сервис на физический IP:порт
     balance source
     mode tcp
     option tcpka
     option mysql-check user haproxy # указываем опцию проверки sql, с
указанием имени пользователя, который будет стучаться к СУБД
     server node1 192.168.100.37:3306 check weight 1 # указываем первый и
     server node2 192.168.100.38:3306 check weight 1 # последующие
бэкенды
```

Перезагружаем службу haproxy и проверяем статус:

```
systemctl reload haproxy.service
systemctl status haproxy.service
● haproxy.service - HAProxy Load Balancer
     Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor
preset: enabled)
```

```
     Active: active (running) since Mon 2022-08-22 13:58:31 UTC; 22h ago
       Docs: man:haproxy(1)
             file:/usr/share/doc/haproxy/configuration.txt.gz
   Main PID: 23096 (haproxy)
      Tasks: 3 (limit: 4612)
     Memory: 2.7M
     CGroup: /system.slice/haproxy.service
             ├─23096 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p
/run/haproxy.pid -S /run/haproxy-master.sock -sf 23488 -x
/run/haproxy/admin.sock
             └─23953 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p
/run/haproxy.pid -S /run/haproxy-master.sock -sf 23488 -x
/run/haproxy/admin.sock

Aug 22 14:18:15 db-sql haproxy[23488]: [WARNING] 233/141815 (23488) :
Stopping frontend GLOBAL in 0 ms.
Aug 22 14:18:15 db-sql haproxy[23488]: [WARNING] 233/141815 (23488) :
Stopping proxy galera in 0 ms.
Aug 22 14:18:15 db-sql haproxy[23488]: [WARNING] 233/141815 (23488) : Proxy
GLOBAL stopped (FE: 1 conns, BE: 1 conns).
Aug 22 14:18:15 db-sql haproxy[23488]: [WARNING] 233/141815 (23488) : Proxy
galera stopped (FE: 0 conns, BE: 0 conns).
Aug 22 14:18:15 db-sql haproxy[23953]: [WARNING] 233/141815 (23953) : Server
galera/node1 is DOWN, reason: Socket error, info: "Connection reset by
peer", check duration: 1m>
Aug 22 14:18:15 db-sql haproxy[23096]: [WARNING] 233/141815 (23096) : Former
worker #1 (23488) exited with code 0 (Exit)
Aug 22 14:18:16 db-sql haproxy[23953]: [WARNING] 233/141816 (23953) : Server
galera/node2 is DOWN, reason: Socket error, info: "Connection reset by
peer", check duration: 1m>
Aug 22 14:18:16 db-sql haproxy[23953]: [ALERT] 233/141816 (23953) : proxy
'galera' has no server available!
Aug 22 14:26:19 db-sql haproxy[23953]: [WARNING] 233/142619 (23953) : Server
galera/node2 is UP, reason: Layer7 check passed, code: 0, info:
"5.5.5-10.3.34-MariaDB-0ubuntu0.>
Aug 22 14:26:20 db-sql haproxy[23953]: [WARNING] 233/142620 (23953) : Server
galera/node1 is UP, reason: Layer7 check passed, code: 0, info:
"5.5.5-10.3.34-MariaDB-0ubuntu0.>
```

From:
https://wiki.virtlab.space/ -

Permanent link:
https://wiki.virtlab.space/common_linux:galera_cluster

Last update: **2022/08/31 17:28**