

Replication SGBD avec Galera Cluster sur Debian 8.4

Sommaire

1. Introduction

- - - - 1.1 Introduction
- - - - 1.2 Prérequis
- - - - 1.3 Version des paquets installés

2. State Snapshot Transfert

- - - - 2.1 XtraBackup

3. Configuration des nœuds

- - - - 3.1 Initialisation du cluster
- - - - 3.2 XinetD et « clustercheck »

4. Load-Balancing

- - - - 3.1 HAProxy

Introduction

Introduction

Un cluster **Galera** permet de faire de la haute disponibilité à une base de données en créant un cluster possédant des nœuds. Donc on aura de la réplication maître-maître permettant ainsi de faire de la répartition de charges et d'avoir une réplication instantanée dès qu'une modification est faite sur la base de données.

Prérequis

Nom de l'hôte	Serveur SGBD: SGBDLab	Serveur SGBD: SGBD2Lab	Serveur SGBD: SGBD3Lab	Serveur Load-Balancing : HAProxyLab	Client	Client 2
Adresse IP	172.16.0.80	172.16.0.81	172.16.0.82	172.16.0.83		
OS/ES	Debian 8.4	Debian 8.4	Debian 8.4	Debian 8.4	Windows 7	Debian 8.2
Logiciel/Paquet	Apache2 MariaDB PHP PHPMYAdmin XtraBackup XinetD	Apache2 MariaDB PHP PHPMYAdmin XtraBackup XinetD	Apache2 MariaDB PHP PHPMYAdmin XtraBackup XinetD	HAProxy	Un navigateur internet: Google Chrome	Mariadb -client

Version des paquets installés

- Apache2 – 2.4.10
- MariaDB – 10.1
- PHP5 – 5.6.13
- PHPMYAdmin – 4.2.12

State Snapshot Transfert

XtraBackup

Une fois **MariaDB** installé sur chaque serveurs ainsi que les services **LAMP**, nous devons installer un **State Snapshot Transfert** pour faire une copie (Sauvegarde) complète de la base de données avec **XtraBackup**, qui permet en plus de ne pas bloquer cette base de données lors de la sauvegarde (**Les différents SST**). Pour faire cela, nous devons installer son dépôt via le téléchargement son package :

```
wget https://repo.percona.com/apt/percona-release_0.1-3.${lsb_release -sc}_all.deb
```

Puis nous installons ce package afin de rajouter le dépôt de Percona :

```
dpkg -i percona-release_0.1-3.${lsb_release -sc}_all.deb
```

Ensuite nous mettons à jour le serveur :

```
apt-get update
```

Après, nous installons XtraBackup :

```
apt-get install percona-xtrabackup
```

À la suite de cela nous devons créer un utilisateur pour **XtraBackup** dans la base de données de **MariaDB** afin qu'il puisse sauvegarder celle-ci, tout d'abord connectons nous dans la base de données :

```
mysql -uroot -p
```

Puis nous créons son utilisateur :

```
create user 'xtrauser'@'localhost' identified by 'Sio2a' ;
```

Ensuite nous lui donnons les droits sur le verrouillage des tables de la base de données ainsi que l'autorisation de savoir où se trouve les serveurs maître et esclaves :

```
grant reload, lock tables, replication client on *.* to 'xtrauser'@'localhost' ;
```

Enfin nous mettons à jour les privilèges de la table contenant les droits :

```
flush privileges ;
```

Configuration des nœuds

Initialisation du cluster

Pour que chaque serveurs soit considéré comme un nœud dans le « Cluster 1 » nous devons créer sur les serveurs un fichier de configuration, sur le premier serveur :

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

Puis nous y mettons :

```
[mysqld]

# Configuration du cluster de nœuds
wsrep_cluster_name='Cluster 1'
wsrep_cluster_address=gcomm://

# Configuration du nœud
bind-address=0.0.0.0
wsrep_node_name='SGBDLab'
wsrep_node_address='172.16.0.80'

# Paramètres obligatoires pour activer Galera
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
binlog_format=ROW
default-storage-engine=InnoDB
innodb_autoinc_lock_mode=2
query_cache_size=0
query_cache_type=0

# Configuration de la synchronisation de Galera
wsrep_sst_method=xtrabackup-v2
wsrep_sst_auth=xtrauser:Sio2a
```

Grâce à la ligne « wsrep_cluster_address=gcomm:// » on dit que c'est le nœud « SGBDLab » qui va initialiser le cluster. Pour démarrer le cluster et la configuration nœud nous devons redémarrer le service MySQL :

```
service mysql restart
```

Si tout ce passe bien le service redémarre sans erreurs. Nous pouvons configurer les autres nœuds.

Configuration des nœuds

Nous créons le fichier de configuration pour les nœuds, et le cluster sur les deux serveurs « SGBD2Lab » et « SGBD3Lab » :

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

Puis nous y mettons comme configuration sur ces deux nœuds :

SGBD2Lab :	SGBD3Lab :
<pre>[mysqld] # Configuration du cluster de nœuds wsrep_cluster_name='Cluster 1' wsrep_cluster_address=gcomm://SGBD2Lab # Configuration du nœud bind-address=0.0.0.0 wsrep_node_name='SGBD2Lab' wsrep_node_address='172.16.0.81' # Paramètres obligatoires pour activer Galera wsrep_on=ON wsrep_provider=/usr/lib/galera/libgalera_smm.so binlog_format=ROW default-storage-engine=InnoDB innodb_autoinc_lock_mode=2 query_cache_size=0 query_cache_type=0 # Configuration de la synchronisation de Galera wsrep_sst_method=xtrabackup-v2 wsrep_sst_auth=xtrauser:Sio2a</pre>	<pre>[mysqld] # Configuration du cluster de nœuds wsrep_cluster_name='Cluster 1' wsrep_cluster_address=gcomm://SGBD3Lab # Configuration du nœud bind-address=0.0.0.0 wsrep_node_name='SGBD3Lab' wsrep_node_address='172.16.0.82' # Paramètres obligatoires pour activer Galera wsrep_on=ON wsrep_provider=/usr/lib/galera/libgalera_smm.so binlog_format=ROW default-storage-engine=InnoDB innodb_autoinc_lock_mode=2 query_cache_size=0 query_cache_type=0 # Configuration de la synchronisation de Galera wsrep_sst_method=xtrabackup-v2 wsrep_sst_auth=xtrauser:Sio2a</pre>

Cette fois-ci la ligne « wsrep_cluster_address=gcomm://SGBD2Lab » on dit aux deux nœuds de s'initialiser sur le nœud « SGBD2Lab ». Nous redémarrons le service MySQL pour démarrer le nœud et de se connecter au cluster :

```
service mysql restart
```

Maintenant il nous reste plus qu'à donner la configuration définitive aux trois nœuds en modifiant le paramètre « wsrep_cluster_address » ainsi :

```
wsrep_cluster_address="gcomm://SGBD2Lab,SGBD3Lab,SGBD2Lab"
```

Puis nous redémarrons sur les trois nœuds le service MySQL :

```
service mysql restart
```

Configuration des nœuds

XinetD et « clustercheck »

Nous allons télécharger un script appelé « clustercheck » sur les trois nœuds afin de connaître l'état de chaque nœuds (Si il bien connecté au Cluster et si le service MySQL est bien démarré). Pour cela nous téléchargeons un script :

```
wget -P /usr/bin/ https://raw.githubusercontent.com/olafz/percona-clustercheck/master/clustercheck
```

Nous nous donnons tous les droits sur ce fichier et nous donnons seulement les droits de lecture et d'exécutions à notre groupe et aux autres utilisateurs :

```
chmod 755 /usr/bin/clustercheck
```

Ensuite nous installons XinetD sur les trois nœuds pour créer un service qui utilisera le script de « clustercheck » afin d'envoyer les informations du script vers un port spécifique :

```
apt-get install xinetd
```

Puis nous créons le fichier du nouveau service :

```
nano /etc/xinetd.d/mysqlchk
```

Et nous y mettons :

```
# Default : ON
# Description : Surveille l'état du nœud
service mysqlchk
{
    disable      = no
    flags        = REUSE
    socket_type  = stream
    port         = 9200
    wait         = no
    user         = nobody
    server       = /usr/bin/clustercheck
    only_from    = 0.0.0.0/0
    per_source   = UNLIMITED
}
```

Configuration des nœuds

Enfin nous donnons un port pour ce service, afin d'envoyer l'état par ce port :

```
echo "mysqlchk 9200/TCP # Surveille l'état du nœud" >> etc/services
```

À la suite de cela nous devons créer un utilisateur les trois nœuds pour le script **ClusterCheck** dans la base de données de **MariaDB** afin qu'il puisse vérifier son état et si il est connecté au cluster :

```
mysql -uroot -p
```

Puis nous créons son utilisateur :

```
create user 'clustercheckuser'@'localhost' identified by 'clustercheckpassword!';
```

Ensuite nous lui donnons les droits de surveiller l'état de **MariaDB** :

```
grant process on *.* to 'clustercheckuser'@'localhost' ;
```

Et nous mettons à jour les privilèges de la table contenant les droits :

```
flush privileges ;
```

Enfin nous redémarrons le service **XinetD** :

```
service xinetd restart
```

Pour tester que le script marche bien, et ainsi vérifier l'état du nœud et de sa connexion au cluster :

```
/usr/bin/clustercheck
```

Si tout est bon, on obtient ceci :

```
HTTP/1.1 200 OK
Content-Type : text/plain
Connection : close
Content-length : 40

Percona XtraDB Cluster Node is synced
```

Load-Balancing

HAProxy

Nous allons mettre en place sur le serveur « HAProxyLab » une répartition de charges sur les services LAMP des trois serveurs et sur les requêtes SQL. Pour cela nous devons installer HAProxy :

```
apt-get install haproxy
```

Ensuite nous faisons une copie du fichier de configuration de **HAProxy** afin d'avoir une sauvegarde en cas de problème :

```
mv /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bak
```

Puis modifions ce fichier :

```
nano /etc/haproxy/haproxy.cfg
```

Et nous y mettons :

Load-Balancing

```
global
  log 127.0.0.1 local2
  maxconn 1024
  user haproxy
  group haproxy
  daemon
  stats socket /var/run/haproxy.sock mode 600 level admin

defaults
  log global
  mode http
  option tcplog
  option dontlognull
  option redispatch
  retries 3
  maxconn 1024
  timeout connect 5000ms
  timeout client 5000ms
  timeout server 5000ms

listen Service_Web 0.0.0.0:80
  mode http
  balance source
  option httpchk
  server SGBDLab 172.16.0.80:80 check
  server SGBD2Lab 172.16.0.81:80 check
  server SGBD3Lab 172.16.0.82:80 check

listen Galera_Cluster 0.0.0.0:3307
  mode tcp
  balance leastconn
  option httpchk
  server SGBDLab 172.16.0.80:3306 check port 9200
  server SGBD2Lab 172.16.0.81:3306 check port 9200
  server SGBD3Lab 172.16.0.82:3306 check port 9200

listen stats 0.0.0.0:9000
  mode http
  stats enable
  stats uri /stats
  stats realm HAProxy\ Statistiques
  stats auth haproxy:Sio2a
  stats admin if TRUE
```

Load-Balancing

Nous redémarrons le service HAProxy :

service haproxy restart

Enfin nous vérifions sur HAProxy fonctionne bien, ainsi que les services LAMP, le cluster Galera et les statistiques, en allant sur un navigateur internet, et en mettant comme adresse <http://172.16.0.83:9000/stats> :

Statistics Report for pid 1681

> General process information

pid = 1681 (process #1, nbproc = 1)
 uptime = 0d 0h00m26s
 system limits: memmax = unlimited; ulimit-n = 2087
 maxsock = 2087; maxconn = 1024; maxpipes = 0
 current conns = 5; current pipes = 0/0; conn rate = 1/sec
 Running tasks: 1/16; idle = 100 %

■ active UP ■ backup UP
■ active UP, going down ■ backup UP, going down
■ active DOWN, going up ■ backup DOWN, going up
■ active or backup DOWN ■ not checked
■ active or backup DOWN for maintenance (MAINT)
■ active or backup SOFT STOPPED for maintenance
 Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:

- Scope :
- [Hide 'DOWN' servers](#)
- [Refresh now](#)
- [CSV export](#)

External resources:

- [Primary site](#)
- [Updates \(v1.6\)](#)
- [Online manual](#)

	Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Server												
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrt		
Frontend				0	4	-	4	4	1024	8		25894	3518512	0	0	0						OPEN										
<input type="checkbox"/> SGBDLab	0	0	-	0	7		0	4	-	33	33	2s	25894	3518512		0		0	0	0	0	0	0	26s UP	L7OK/200 in 1ms	1	Y	-	0	0	0s	-
<input type="checkbox"/> SGBD2Lab	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	0	0	26s UP	L7OK/200 in 1ms	1	Y	-	0	0	0s	-
<input type="checkbox"/> SGBD3Lab	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	0	0	26s UP	L7OK/200 in 1ms	1	Y	-	0	0	0s	-
Backend	0	0		0	7		0	4	103	33	33	2s	25894	3518512	0	0	0			0	0	0	0	26s UP		3	3	0		0	0s	

Choose the action to perform on the checked servers :

Galera_Cluster																																	
	Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Server													
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrt			
Frontend				0	0	-	0	0	1024	0		0	0	0	0	0						OPEN											
<input type="checkbox"/> SGBDLab	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	0	0	26s UP	L7OK/200 in 13ms	1	Y	-	0	0	0s	-	
<input type="checkbox"/> SGBD2Lab	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	0	0	26s UP	L7OK/200 in 14ms	1	Y	-	0	0	0s	-	
<input type="checkbox"/> SGBD3Lab	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	0	0	26s UP	L7OK/200 in 13ms	1	Y	-	0	0	0s	-	
Backend	0	0		0	0		0	0	103	0	0	?	0	0		0		0	0	0	0	0	0	26s UP		3	3	0		0	0s		

Choose the action to perform on the checked servers :

stats																																
	Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Server												
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrt		
Frontend				1	1	-	1	1	1024	3		721	53592	0	0	0						OPEN										
Backend	0	0		0	0		0	0	103	0	0	0s	721	53592	0	0	0			0	0	0	0	26s UP		0	0	0		0	0s	