27/11/2017

Cluster de base de données Mysql

Version 1.0 : Version Initial

Fabien MAUHOURAT
[NOM DE LA SOCIETE]

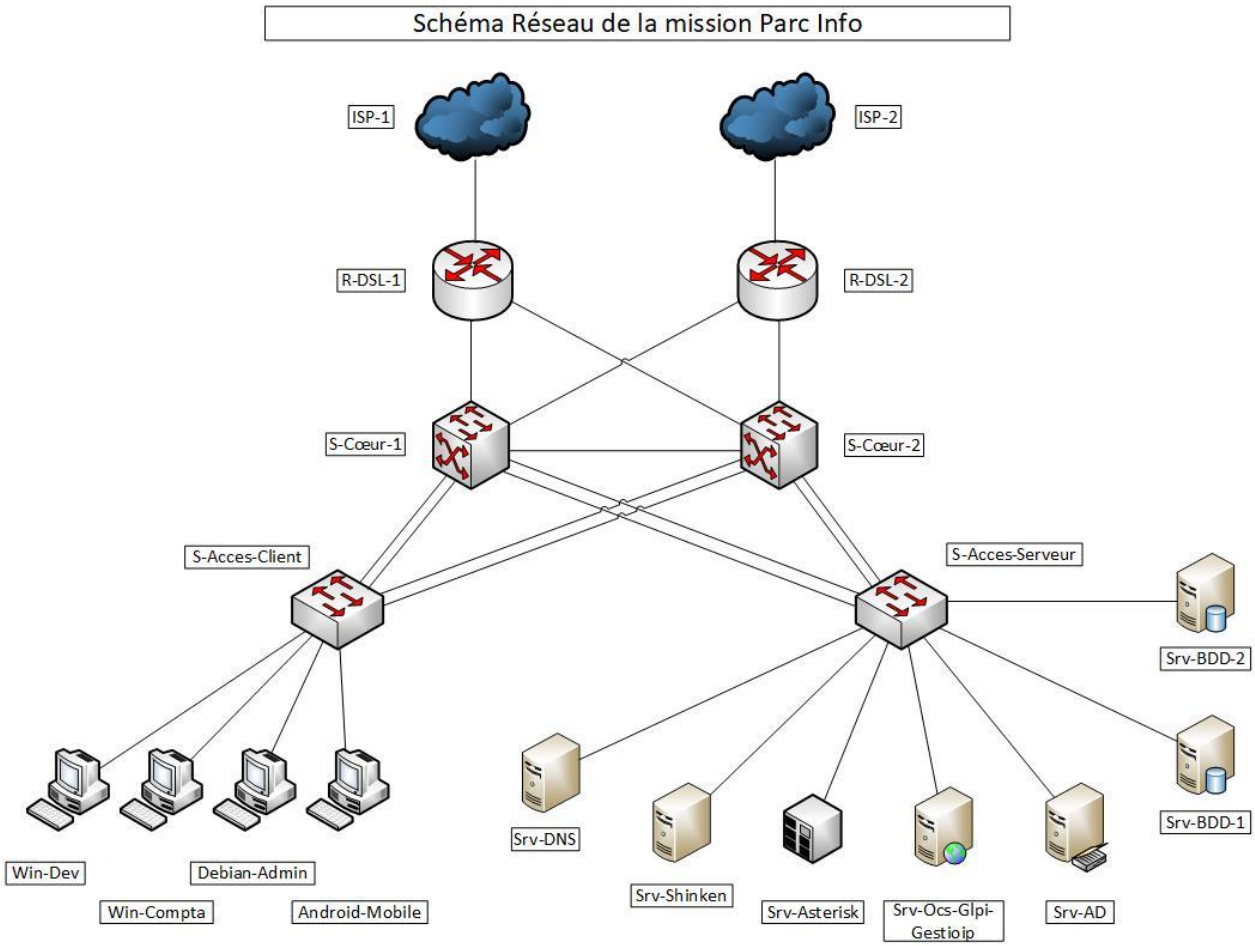
Cluster de base de données Mysql

Contexte :

Pour permettre une haute disponibilité des services critique telle que la gestion des tickets d'incident ou encore la gestion des inventaires du réseau il est nécesaire de mettre en place un cluster de base de données pour assurer une continuité d'activité.

Le cluster se compose de deux serveurs Mysql répliqué en mode master/master puis du service Hearbeat qui permet la mise en cluster de ces deux serveurs avec une IP virtuelle.

Voici l'architecture mise en place :



Sommaire

Réplication base de données ParcInfo

Sur le master :

/etc/mysql/my.cnf : fichier de configuration de mysql

Modifier la ligne bind-address sur le serveur maitre à savoir ubuntu

Bind-address =192.168.75.160

CREATE USER 'slave'@'192.168.75.161' IDENTIFIED BY 'toor';
GRANT REPLICATION SLAVE ON *.* TO 'slave'@'192.168.75.161';
Flush privileges ;
select host, user, password from mysql.user;
service mysql restart

Modifier le fichier de configuration my.cnf du serveur maitre :

```
[mysqld]
log_error = /var/log/mysql/error.log
log_bin  = /var/log/mysql/mysql-bin.log
server-id=1
binlog-do-db= ocsweb
```

```
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#log_slow_queries       = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
server-id               = 1
log_bin                 = /var/log/mysql/mysql-bin.log
expire_logs_days        = 10
max_binlog_size         = 100M
binlog_do_db            = ocsweb
#binlog_ignore_db       = include_database_name
```

show master status\G;

```
mysql> show master status;
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000001 |      477 | ocsweb       |                  |
+------------------+----------+--------------+------------------+
1 row in set (0.00 sec)
```

flush tables with read lock; (unlock tables; pour enlever la protection)

mysqldump –u root –p --database ocsweb > ocsweb.sql

```
fabien@Srv-BDD-1:~$ sudo mysqldump -u root -p --databases ocsweb > ocsweb.sql
Enter password:
fabien@Srv-BDD-1:~$ sudo mysqldump -u root -p --databases glpi > glpi.sql
Enter password:
fabien@Srv-BDD-1:~$ sudo mysqldump -u root -p --databases gestioip > gestioip.sql
Enter password:
fabien@Srv-BDD-1:~$ ls
gestioip.sql  glpi.sql  ocsweb.sql  UnixAgent
fabien@Srv-BDD-1:~$
```

scp ocsweb.sql sio@192.168.75.161:/home/sio

```
fabien@Srv-BDD-1:~$ scp -P 47000 ocsweb.sql fabien@192.168.75.91:/home/fabien
The authenticity of host '[192.168.75.91]:47000 ([192.168.75.91]:47000)' can't be established.
ECDSA key fingerprint is SHA256:fTHQt57gVcWXLNvvuzC46mXddMEN1DsRdiZBuouYlig.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.168.75.91]:47000' (ECDSA) to the list of known hosts.
fabien@192.168.75.91's password:
ocsweb.sql
fabien@Srv-BDD-1:~$ scp -P 47000 glpi.sql  fabien@192.168.75.91:/home/fabien
fabien@192.168.75.91's password:
glpi.sql
fabien@Srv-BDD-1:~$ scp -P 47000 gestioip.sql  fabien@192.168.75.91:/home/fabien
fabien@192.168.75.91's password:
gestioip.sql
fabien@Srv-BDD-1:~$
```

Sur le slave :

apt-get install –y mysql-server

mysql -u root -p < /home/sio/ocsweb.sql

```
fabien@Srv-BDD-2:~$ sudo mysql -u root -p < ocsweb.sql
Enter password:
fabien@Srv-BDD-2:~$ sudo mysql -u root -p < glpi.sql
Enter password:
fabien@Srv-BDD-2:~$ sudo mysql -u root -p < gestioip.sql
Enter password:
```

- Création de l’utilisateur ocs pour prendre le relai si le serveur maitre est defaillant.

CREATE USER 'ocs'@'192.168.75.160' IDENTIFIED BY 'toor';
grant usage on *.* to 'ocs'@’192.168.75.160' ;
grant all privileges on ‘ocsweb’ to 'ocs'@’192.168.75.160' ;
Flush privileges ;

```
MariaDB [glpi]> CREATE USER 'ocs'@'%' identified by 'toor';
Query OK, 0 rows affected (0.00 sec)

MariaDB [glpi]> GRANT ALL ON ocsweb.* TO 'ocs'@'%';
Query OK, 0 rows affected (0.00 sec)

MariaDB [glpi]> CREATE USER 'glpi'@'%' identified by 'toor';
Query OK, 0 rows affected (0.00 sec)

MariaDB [glpi]> GRANT ALL ON glpi.* TO 'glpi'@'%';
Query OK, 0 rows affected (0.00 sec)

MariaDB [glpi]> CREATE USER 'gestioip'@'%' identified by 'toor';
Query OK, 0 rows affected (0.00 sec)

MariaDB [glpi]> GRANT ALL ON gestioip.* TO 'gestioip'@'%';
Query OK, 0 rows affected (0.00 sec)

MariaDB [glpi]> Flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [glpi]> select host, user, password from mysql.user;
+-----------+----------+-------------------------------------------+
| host      | user     | password                                  |
+-----------+----------+-------------------------------------------+
| localhost | root     | *9CFBBC772F3F6C1060200353B6DA5BBBF1249A11 |
| %         | ocs      | *9CFBBC772F3F6C1060200353B6DA5BBBF1249A11 |
| %         | glpi     | *9CFBBC772F3F6C1060200353B6DA5BBBF1249A11 |
| %         | gestioip | *9CFBBC772F3F6C1060200353B6DA5BBBF1249A11 |
+-----------+----------+-------------------------------------------+
4 rows in set (0.00 sec)

MariaDB [glpi]>
```

Modifier la ligne bind-address sur le serveur maitre à savoir ubuntu

Bind-address =192.168.75.161

```
[mysqld]
server-id=2
log_error = /var/log/mysql/error.log
log_bin  = /var/log/mysql/mysql-bin.log
binlog-do-db= ocsweb
```

```
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#log_slow_queries       = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
server-id               = 2
log_bin                 = /var/log/mysql/mysql-bin.log
expire_logs_days        = 10
max_binlog_size         = 100M
binlog_do_db            = ocsweb
#binlog_ignore_db       = include_database_name
```

service mysql restart

`CHANGE MASTER TO MASTER_HOST='192.168.75.160', MASTER_USER='slave', MASTER_PASSWORD='toor', MASTER_LOG_POS=120, MASTER_LOG_FILE='mysql-bin.000004';`

start slave ;

`SHOW SLAVE STATUS\G ;`

```
mysql> show slave status\G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 192.168.75.160
                  Master_User: slave
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000001
          Read_Master_Log_Pos: 477
               Relay_Log_File: mysqld-relay-bin.000002
                Relay_Log_Pos: 253
        Relay_Master_Log_File: mysql-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
```

- Test de la réplication :
  - Vérifier dans la table Hardware de la base de données ocsweb que les pc inventoriés sont bien présents :

```
mysql> select deviceid from hardware;
+-----------------------------+
| deviceid                    |
+-----------------------------+
| ubuntu14-2017-04-11-22-46-42 |
+-----------------------------+
1 row in set (0.00 sec)
```

- Effectuer une remonté d'inventaire puis vérifier à nouveau la table sur le server esclave :

```
mysql> select deviceid from hardware;
+-----------------------------------+
| deviceid                          |
+-----------------------------------+
| ubuntu14-2017-04-11-22-46-42      |
| ParcInfoMaster-2017-04-14-19-16-50 |
+-----------------------------------+
2 rows in set (0.00 sec)
```

```
# Generate a CA key and certificate with SHA1 digest
openssl genrsa 2048 > ca-key.pem
openssl req -sha1 -new -x509 -nodes -days 3650 -key ca-key.pem > ca-
cert.pem

# Create server key and certficate with SHA1 digest, sign it and convert
# the RSA key from PKCS #8 (OpenSSL 1.0 and newer) to the old PKCS #1
format
openssl req -sha1 -newkey rsa:2048 -days 730 -nodes -keyout server-key.pem
> server-req.pem
openssl x509 -sha1 -req -in server-req.pem -days 730  -CA ca-cert.pem -
CAkey ca-key.pem -set_serial 01 > server-cert.pem
openssl rsa -in server-key.pem -out server-key.pem

# Create client key and certificate with SHA digest, sign it and convert
# the RSA key from PKCS #8 (OpenSSL 1.0 and newer) to the old PKCS #1
format
openssl req -sha1 -newkey rsa:2048 -days 730 -nodes -keyout client-key.pem
> client-req.pem
openssl x509 -sha1 -req -in client-req.pem -days 730 -CA ca-cert.pem -CAkey
ca-key.pem -set_serial 01 > client-cert.pem
openssl rsa -in client-key.pem -out client-key.pem
```