

Configuração Cluster de Base de Dados

Instalação de Heartbeat e Percona XtraDB em CentOS 6.5

Este documento detalha os passos necessários para uma instalação de Heartbeat e Percona XtraDB em 3 sistemas CentOS 6.5.

Cada um dos sistemas será identificado pela seguinte informação:

Node #1
hostname: percona1
IP: 192.168.70.71

Node #2
hostname: percona2
IP: 192.168.70.72

Node #3
hostname: percona3
IP: 192.168.70.73

Para o endereço de VIP vamos usar o seguinte IP: 192.168.70.70

1. Heartbeat

Instalação do repositório epel:

```
# wget http://download.fedoraproject.org/pub/epel/6/\$PLATFORM/epel-release-6-8.noarch.rpm
# rpm -Uvh epel-release-6-8.noarch.rpm
```

Garantir que as configurações entre os 3 sistemas não são restringidas por nenhuma firewall. Se os sistemas tiverem ativas uma firewall local, adicionar as seguintes regras à configuração da firewall:

```
-A INPUT -s 192.168.70.71/32 -j ACCEPT
-A INPUT -s 192.168.70.72/32 -j ACCEPT
-A INPUT -s 192.168.70.73/32 -j ACCEPT
```

Instalar o pacote heartbeat:

Configuração Cluster de Base de Dados

```
# yum -y install heartbeat
```

Configurar o heartbeat:

Em cada um dos nós:

Editar o ficheiro `/etc/hosts` e incluir:

```
192.168.70.71    perconal
192.168.70.72    percona2
192.168.70.73    percona3
```

Criar o ficheiro `/etc/ha.d/authkeys` com o seguinte conteúdo:

```
auth 3
3 md5 <PASSWORD>
```

Criar o ficheiro `/etc/ha.d/haresources` com o seguinte conteúdo:

```
perconal IPaddr2::192.168.70.70/24/eth0
```

Criar o ficheiro `/etc/ha.d/ha.cf` com o seguinte conteúdo:

```
autojoin none
logfacility local0
mcast eth0 239.0.0.1 694 1 0
initdead 65
deadtime 60
warntime 30
keepalive 15
auto_failback off
node perconal
node percona2
respawn hacluster /usr/lib64/heartbeat/ipfail
apiauth ipfail gid=haclient uid=hacluster
```

No Node #1 e Node #2:

Configuração Cluster de Base de Dados

Ativar o serviço heartbeat no startup do sistema:

```
# chkconfig heartbeat on
```

No Node #3

Desativar o serviço heartbeat no startup do sistema:

```
# chkconfig heartbeat off
```

Levantar o serviço heartbeat no Node #1

```
# service heartbeat start
```

Levantar o serviço heartbeat no Node #2

```
# service heartbeat start
```

Parar o serviço heartbeat no Node #3

```
# service heartbeat stop
```

2. Percona XtraDB

NOTA: o Percona XtraDB não replica tabelas com o storage engine MyISAM.

Antes de importar um dump, garantir que as tabelas da base de dados do SugarCRM estão todas em InnoDB.

Para detetar e converter tabelas MyISAM to InnoDB, efetuar os seguintes passos antes de efetuar o dump à base de dados:

- Determinar tabelas com storage engine diferente de InnoDB:

```
mysql> SELECT concat('ALTER TABLE `',TABLE_NAME,'` ENGINE=InnoDB;')  
FROM Information_schema.TABLES  
WHERE ENGINE != 'InnoDB' AND TABLE_TYPE='BASE TABLE'  
AND TABLE_SCHEMA='sugarcrm_ama';
```

Configuração Cluster de Base de Dados

- Output do comando anterior:

```
+-----+
| concat('ALTER TABLE `',TABLE_NAME, '` ENGINE=InnoDB;') |
+-----+
| ALTER TABLE `ama_billing_docs_clusteruniqueid_invoicenumber` ENGINE=InnoDB; |
| ALTER TABLE `ama_tickets_clusteruniqueid_ticketnumber` ENGINE=InnoDB; |
| ALTER TABLE `emails_text` ENGINE=InnoDB; |
+-----+
```

- Usar o output do comando anterior para converter as tabelas:

```
mysql> ALTER TABLE `ama_billing_docs_clusteruniqueid_invoicenumber` ENGINE=InnoDB;
mysql> ALTER TABLE `ama_tickets_clusteruniqueid_ticketnumber` ENGINE=InnoDB;
mysql> ALTER TABLE `emails_text` ENGINE=InnoDB;
```

A documentação do Percona XtraDB Cluster está disponível no seguinte URL:

<https://www.percona.com/doc/percona-xtradb-cluster/5.5/>

Efetuar o download dos seguintes pacotes do site da Percona (<http://www.percona.com>):

```
percona-xtrabackup-2.2.11-1.el6.x86_64.rpm
Percona-XtraDB-Cluster-server-55-5.5.41-25.11.853.el6.x86_64.rpm
Percona-XtraDB-Cluster-client-55-5.5.41-25.11.853.el6.x86_64.rpm
Percona-XtraDB-Cluster-galera-2-2.12-1.2682.rhel6.x86_64.rpm
```

Em cada um dos nós, instalar os pacotes:

```
# yum localinstall percona-xtrabackup-2.2.11-1.el6.x86_64.rpm
# yum localinstall Percona-XtraDB-Cluster-server-55-5.5.41-
25.11.853.el6.x86_64.rpm Percona-XtraDB-Cluster-client-55- 5.5.41-
25.11.853.el6.x86_64.rpm Percona-XtraDB-Cluster-galera-2-2.12-
1.2682.rhel6.x86_64.rpm
```

No Node #1:

Criar o ficheiro `/etc/my.cnf` com o seguinte conteúdo:

```
[mysqld]
```

Configuração Cluster de Base de Dados

```
datadir=/var/lib/mysql
user=mysql

# Path to Galera library
wsrep_provider=/usr/lib64/libgalera_smm.so

# Cluster connection URL contains the IPs of node#1, node#2 and node#3
wsrep_cluster_address=gcomm://192.168.70.71,192.168.70.72,192.168.70.73

# In order for Galera to work correctly binlog format should be ROW
binlog_format=ROW

# MyISAM storage engine has only experimental support
default_storage_engine=InnoDB

# This changes how InnoDB autoincrement locks are managed and is a
requirement for Galera
innodb_autoinc_lock_mode=2

# Node #1 address
wsrep_node_address=192.168.70.71

# SST method
wsrep_sst_method=xtrabackup-v2

# Cluster name
wsrep_cluster_name=my_centos_cluster

# Authentication for SST method
wsrep_sst_auth="sstuser:s3cret"
```

Levantar o mysql no Node #1 com o seguinte comando:

```
# /etc/init.d/mysql bootstrap-pxc
```

Verificar o estado do cluster com o comando:

```
mysql> show status like 'wsrep%';
```

Variable_name	Value
wsrep_local_state_uuid	c2883338-834d-11e2-0800-03c9c68e41ec
...	
wsrep_local_state	4
wsrep_local_state_comment	Synced
...	
wsrep_cluster_size	1
wsrep_cluster_status	Primary

Configuração Cluster de Base de Dados

wsrep_connected	ON	
...		
wsrep_ready	ON	
+-----+-----+		

40 rows in set (0.01 sec)

Definir a password para a conta de root:

```
mysql> UPDATE mysql.user SET password=PASSWORD("Passw0rd") where
user='root';
mysql > flush privileges;
```

Criar user para o sst:

```
mysql> CREATE USER 'sstuser'@'localhost' IDENTIFIED BY 's3cret';
mysql> GRANT RELOAD, LOCK TABLES, REPLICATION CLIENT ON *.* TO
'sstuser'@'localhost';
mysql> FLUSH PRIVILEGES;
```

No Node #2:

Criar o ficheiro /etc/my.cnf com o seguinte conteúdo:

```
[mysqld]

datadir=/var/lib/mysql
user=mysql

# Path to Galera library
wsrep_provider=/usr/lib64/libgalera_smm.so

# Cluster connection URL contains IPs of node#1, node#2 and node#3
wsrep_cluster_address=gcomm://192.168.70.71,192.168.70.72,192.168.70.73

# In order for Galera to work correctly binlog format should be ROW
binlog_format=ROW

# MyISAM storage engine has only experimental support
default_storage_engine=InnoDB

# This changes how InnoDB autoincrement locks are managed and is a
requirement for Galera
innodb_autoinc_lock_mode=2

# Node #2 address
wsrep_node_address=192.168.70.72
```

Configuração Cluster de Base de Dados

```
# Cluster name
wsrep_cluster_name=my_centos_cluster

# SST method
wsrep_sst_method=xtrabackup-v2

#Authentication for SST method
wsrep_sst_auth="sstuser:s3cret"
```

Levantar o mysql no Node #2 com o seguinte comando:

```
# /etc/init.d/mysql start
```

Depois do servidor arrancar, vai receber o sst automaticamente, por esse motivo, o nó 2 já não terá a password de root vazia. Para estabelecer a ligação à base de dados e verificar o estado do cluster, teremos de usar a password definida no Node #1.
O estado do cluster pode ser agora verificado em ambos os nós:

```
mysql> show status like 'wsrep%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_local_state_uuid | c2883338-834d-11e2-0800-03c9c68e41ec |
...
| wsrep_local_state | 4 |
| wsrep_local_state_comment | Synced |
...
| wsrep_cluster_size | 2 |
| wsrep_cluster_status | Primary |
| wsrep_connected | ON |
...
| wsrep_ready | ON |
+-----+-----+
40 rows in set (0.01 sec)
```

O output mostra que o novo nó foi adicionado com sucesso ao cluster.

No Node #3:

Criar o ficheiro `/etc/my.cnf` com o seguinte conteúdo:

```
[mysqld]
```

Configuração Cluster de Base de Dados

```
datadir=/var/lib/mysql
user=mysql

# Path to Galera library
wsrep_provider=/usr/lib64/libgalera_smm.so

# Cluster connection URL contains IPs of node#1, node#2 and node#3
wsrep_cluster_address=gcomm://192.168.70.71,192.168.70.72,192.168.70.73

# In order for Galera to work correctly binlog format should be ROW
binlog_format=ROW

# MyISAM storage engine has only experimental support
default_storage_engine=InnoDB

# This changes how InnoDB autoincrement locks are managed and is a
requirement for Galera
innodb_autoinc_lock_mode=2

# Node #3 address
wsrep_node_address=192.168.70.73

# Cluster name
wsrep_cluster_name=my_centos_cluster

# SST method
wsrep_sst_method=xtrabackup-v2

#Authentication for SST method
wsrep_sst_auth="sstuser:s3cret"
```

Levantar o mysql no Node #3 com o seguinte comando:

```
# /etc/init.d/mysql start
```

Depois de levantado, o nó deverá receber o sst tal como o Node #2, e podemos verificar o estado com o seguinte comando:

```
mysql> show status like 'wsrep%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_local_state_uuid | c2883338-834d-11e2-0800-03c9c68e41ec |
| ... | |
| wsrep_local_state | 4 |
```


Configuração Cluster de Base de Dados

```
| wsrep_local_state_comment | Synced |
...
| wsrep_cluster_size       | 3      |
| wsrep_cluster_status     | Primary|
| wsrep_connected          | ON     |
...
| wsrep_ready               | ON     |
+-----+-----+
40 rows in set (0.01 sec)
```

Testar a replicação:

- Criar base de dados no Node #2:

```
mysql> create database percona;
```

- Criar tabela example no Node #3:

```
mysql> create table example (node_id INT PRIMARY KEY, node_name
    VARCHAR(30));
```

- Inserir registo no Node #1:

```
mysql> INSERT INTO percona.example VALUES (1, 'perconal');
```

- Selecionar registos inseridos no Node #2:

```
mysql> select * from percona.example;
```