

Cluster Mysql 5.X

De Telefonica I+D - Division 6123

Explicaremos cómo instalar y configurar un Mysql 5.0 Cluster sobre Centos 5

Autor: Edwin Plauchu
mail to: j4nusz@gmail.com

Tabla de contenidos

- 1 Introduccion
- 2 Instalando
 - 2.1 Paso 1
 - 2.2 Paso 2
 - 2.3 Paso 3
 - 2.4 Paso 4
 - 2.5 Paso 5
 - 2.6 Paso 6
 - 2.7 Paso 7

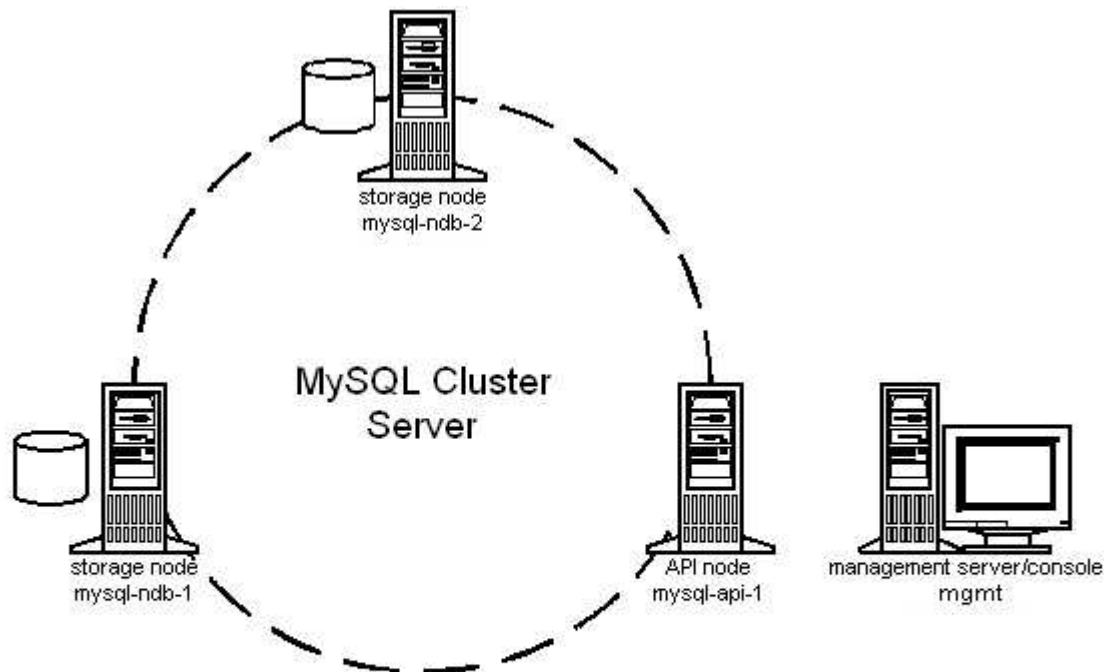
Introduccion

El servidor MySQL Cluster es:

- Tolerante a fallos.
- Redundante.
- Arquitectura escalable en base de datos.
- 99.999% rentable

En este documento describire paso a paso como instalarlo, configurarlo y probarlo con 3 mini-nodos mySQL sobre un entorno de pruebas.

- **mysql-ndb-1** --> 192.168.10.101 (*Nodo de almacenamiento*)
- **mysql-ndb-2** --> 192.168.10.102 (*Nodo de almacenamiento*)
- **mysql-api-1** --> 192.168.10.103 (*Nodo API - Interface de Aplicacion*)
- **mgmt** --> 192.168.10.104 (*Nodo de Administracion*)



Instalando

Paso 1

Sobre ambos nodos de almacenamiento, mysql-ndb-1 (192.168.10.101) y mysql-ndb-2 (192.168.10.102), obtendremos e instalaremos la suite Mysql-5 con sus agregados para NDB engine, aclaro que me he bajado solo RPMS para x86 Linux (dynamically linked), mismos que he depositado en el directorio /root/mysql-rpm-cluster/:

```
root@mysql-ndb-1:~/mysql-rpm-cluster$ ls
MySQL-bench-5.0.45-0.glibc23.i386.rpm
MySQL-client-5.0.45-0.glibc23.i386.rpm
MySQL-debuginfo-5.0.45-0.glibc23.i386.rpm
MySQL-devel-5.0.45-0.glibc23.i386.rpm
MySQL-ndb-extra-5.0.45-0.glibc23.i386.rpm
MySQL-ndb-management-5.0.45-0.glibc23.i386.rpm
MySQL-ndb-storage-5.0.45-0.glibc23.i386.rpm
MySQL-ndb-tools-5.0.45-0.glibc23.i386.rpm
MySQL-server-5.0.45-0.glibc23.i386.rpm
MySQL-shared-5.0.45-0.glibc23.i386.rpm
MySQL-shared-compat-5.0.45-0.glibc23.i386.rpm
```

Ahora proceda a instalar los paquetes rpm en los nodos mysql-ndb-1 y mysql-ndb-2:

```
root@mysql-ndb-1:~/mysql-rpm-cluster$ rpm -Uvh /root/mysql-rpm-cluster/<paquete-rpm>
```

```
root@mysql-ndb-2:~/mysql-rpm-cluster$ rpm -Uvh /root/mysql-rpm-cluster/<paquete-rpm>
```

Nota:

No arranque los servidores mysql que acaba de instalar aun!.

Paso 2

Instale la consola de administracion sobre el host mgmt (192.168.10.104).

```
mgmt$ rpm -Uvh MySQL-ndb-management-5.0.45-0.glibc23.i386.rpm
```

Es necesario crear el directorio en que puede encontrarse el archivo de configuración y crear el archivo própiamente dicho. Por ejemplo (ejecutando como root):

```
mgmt$ mkdir /var/lib/mysql-cluster
mgmt$ cd /var/lib/mysql-cluster
mgmt$ vi config.ini
```

El nodo de administración requiere un archivo **config.ini** que le diga cuántas replicas mantener, cuánta memoria reservar para datos e índices en cada nodo de almacenamiento, dónde encontrar los nodos de almacenamiento, dónde se guardarán los datos en cada nodo de almacenamiento, y dónde encontrar los nodos SQL.

El archivo **config.ini** contendrá información acerca de la configuración del cluster:

```
[NDBD DEFAULT] # Opciones que afectaran a los procesos ndbd sobre los nodos de almacenamiento de datos:
NoOfReplicas=2 # Numero de replicas

[MYSQLD DEFAULT]

[NDB_MGMD DEFAULT]

[TCP DEFAULT]

# Servidor de administracion
[NDB_MGMD]
HostName=192.168.10.104 # Direccion IP de este servidor

# Nodos de almacenamiento
[NDBD]
HostName=192.168.10.101 # Direccion IP del nodo de almacenamiento mysql-ndb-1
DataDir=/var/lib/mysql-cluster

[NDBD]
HostName=192.168.10.102 # Direccion IP del nodo de almacenamiento mysql-ndb-2
DataDir=/var/lib/mysql-cluster

# Opciones para nodos API (clientes de el cluster)
[MYSQLD]
[MYSQLD]
[MYSQLD]
[MYSQLD]
```

Ahora arranque el demonio **ndb_mgmd** y verifique que este si se encuentra efectivamente corriendo (el demonio **ndb_mgmd** lee el archivo de configuración del cluster y distribuye esta información a todos los nodos en el cluster que la piden):

```
mgmt$ ndb_mgmd -f /var/lib/mysql-cluster/config.ini
mgmt$ ps -ax | grep ndb_mgmd
```

Paso 3

Sobre ambos nodos de almacenamiento, **mysql-ndb-1** (192.168.10.101) y **mysql-ndb-2** (192.168.10.102), configure los servidores mySQL:

```
mysql-ndb-1$ vi /etc/my.cnf
```

```
mysql-ndb-2$ vi /etc/my.cnf
```

Este es el archivo de configuración (/etc/**my.cnf**) para ambos servidores mysql sobre los nodos de

almacenamiento:

```
[mysqld] #Opciones para el proceso mysqld
ndbcluster #Activa el engine NDB
ndb-connectstring='host=192.168.10.104' # Direccion IP del servidor de administracion

[mysql_cluster] # Opciones para el proceso ndbd:
ndb-connectstring='host=192.168.10.104' # Direccion IP del servidor de administracion
```

Sobre ambos nodos de almacenamiento, heche a volar la ingenieria de almacenamiento NDB y el servidor Mysql server, verifique que ambos se encuentren corriendo:

```
mysql-ndb-1$ mkdir /var/lib/mysql-cluster
mysql-ndb-1$ /usr/sbin/ndbd --initial
mysql-ndb-1$ /etc/init.d/mysql start
mysql-ndb-1$ ps -ax | grep ndbd
mysql-ndb-1$ ps -ax | grep mysqld
```

```
mysql-ndb-2$ mkdir /var/lib/mysql-cluster
mysql-ndb-2$ /usr/sbin/ndbd --initial
mysql-ndb-2$ /etc/init.d/mysql start
mysql-ndb-2$ ps -ax | grep ndbd
mysql-ndb-2$ ps -ax | grep mysqld
```

Nota:

Si Mysql no arranco apropiadamente, cheque el archivo de log en /var/lib/mysql/data/\${HOSTNAME}.err y corrija el problema.

Paso 4

Inicialice la consola de administracion sobre el host **mgmt** y consulte el estado del cluster:

```
mgmt$ ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.10.101 (Version: 5.0.19, STARTING, Nodegroup: 0, Master)
id=3 @192.168.10.102 (Version: 5.0.19, STARTING, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.10.104 (Version: 5.0.19)

[mysqld(API)] 4 node(s)
id=4 (NOT connected, accepting connect FROM any host)
id=5 (NOT connected, accepting connect FROM any host)
id=6 (NOT connected, accepting connect FROM any host)
id=7 (NOT connected, accepting connect FROM any host)
```

Paso 5

Cree una base de datos de prueba, y llenela de datos, no olvide usar en las tablas el engine NDBCLUSTER, y verifique que todo funcione de manera correcta:

Sobre ambos nodos **mysql-ndb-1** y **mysql-ndb-2** crea la base de datos de prueba:

```
mysql-ndb-1$ mysql -u root
Welcome to the MySQL monitor. Commands END with ; OR \g.
Your MySQL connection id IS 1 to server version: 5.0.19

Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> CREATE DATABASE ejemplo;
Query OK, 1 row affected (0.09 sec)
```

```
mysql-ndb-2$ mysql -u root
Welcome to the MySQL monitor.  Commands END with ; OR \g.
Your MySQL connection id IS 6 to server version: 5.0.19

Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> CREATE DATABASE ejemplo;
Query OK, 1 row affected (0.13 sec)
```

Regresamos sobre el nodo de almacenamiento **mysql-ndb-1**, y agregamos a la base de datos una tabla que contenga datos simples:

```
mysql-ndb-1$ mysql -u root
Welcome to the MySQL monitor.  Commands END with ; OR \g.
Your MySQL connection id IS 1 to server version: 5.0.19

Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> USE ejemplo;
Database changed
mysql> CREATE TABLE pruebal (i INT) engine=ndbcluster;
Query OK, 0 rows affected (0.94 sec)

mysql> INSERT INTO pruebal () VALUES (1);
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM pruebal;
+-----+
| i     |
+-----+
| 1    |
+-----+
1 row IN SET (0.01 sec)
```

Ahora valla al nodo de almacenamiento **mysql-ndb-2** y verifique que los datos estan accesibles:

```
mysql-ndb-2$ mysql -u root
Welcome to the MySQL monitor.  Commands END with ; OR \g.
Your MySQL connection id IS 7 to server version: 5.0.19

Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> USE ejemplo;
Reading table information for completion of table AND COLUMN names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM pruebal;
+-----+
| i     |
+-----+
| 1    |
+-----+
1 row IN SET (0.00 sec)
```

Esta es una buena señal, pero actualmente no se estan replicando los datos. El nodo de almacenamiento (**mysql-ndb-2**) es tambien un nodo API del cluster, y este nodo solo muestra meramente, que solo es posible traer datos del cluster. Esto no demostro nada en base al mecanismo de almacenamiento en cluster. Demostraremos esto de una manera mas clara con la prueba siguiente.

Mate los procesos que hacen referencia al engine NDB (**ndbd**) sobre uno de los nodos de almacenamiento (**mysql-ndb-2**), en este orden simularemos una falla en el engine de almacenamiento:

```
mysql-ndb-2$ ps -aux | grep ndb
root 3035 3034 0 17:28:41 ?        0:23 /usr/sbin/ndbd --initial
root 3034 1 0 17:28:41 ?        0:00 /usr/sbin/ndbd --initial
mysql-ndb-2$ killall ndbd
mysql-ndb-2$ ps -aux | grep ndb
```

El servidor de administracion reconocera que el nodo de almacenamiento **mysql-ndb-2** (192.168.10.102) ha fallado, pero su coneccion API esta aun activa:

```

ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.10.101 (Version: 5.0.19, Nodegroup: 0)
id=3 (NOT connected, accepting connect FROM 192.168.10.102)
[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.10.104 (Version: 5.0.19)
[mysqld(API)] 4 node(s)
id=4 @192.168.10.103 (Version: 5.0.19)
id=5 @192.168.10.105 (Version: 5.0.19)
id=6 (NOT connected, accepting connect FROM any host)
id=7 (NOT connected, accepting connect FROM any host)

```

Sobre el primer nodo de almacenamiento (**mysql-ndb-1**) agregaremos otra nueva tabla con algunos datos de prueba:

```

mysql-ndb-1$ mysql -u root
Welcome to the MySQL monitor. Commands END with ; OR \g.
Your MySQL connection id IS 4 to server version: 5.0.19
Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> USE ejemplo;
Reading table information for completion of table AND COLUMN names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE TABLE prueba2 (i INT) engine=ndbcluster;
Query OK, 0 rows affected (1.00 sec)

mysql> INSERT INTO prueba2 () VALUES (2);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM prueba2;
+----+
| i  |
+----+
|  2 |
+----+
1 row IN SET (0.01 sec)

```

Regresemos al segundo nodo de almacenamiento (**mysql-ndb-2**) y llevemos a cabo el mismo comando select:

```

mysql-ndb-2$ mysql -u root
Welcome to the MySQL monitor. Commands END with ; OR \g.
Your MySQL connection id IS 9 to server version: 5.0.19
Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> USE ejemplo;
Reading table information for completion of table AND COLUMN names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM prueba2;
+----+
| i  |
+----+
|  2 |
+----+
1 row IN SET (0.01 sec)

```

Debe quedar claro que el engine de almacenamiento y el servidor API son 2 cosas separadas, procesos distintos que no son dependientes uno del otro. Una vez que el proceso que activa el engine de almacenamiento es reiniciado sobre el segundo nodo de almacenamiento (**mysql-ndb-2**), el dato es replicado, como se muestra a continuacion.

Primero reiniciaremos el proceso ndbd(engine del almacenamiento) sobre el nodo de almacenamiento mysql-ndb-2:

```
mysql-ndb-2$ /usr/sbin/ndbd
```

Lo siguiente, apague el engine de almacenamiento sobre **mysql-ndb-1** usando la consola de administracion o

el comando kill en la shell:

```

mgmt$ ndb mgm
ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.10.101 (Version: 5.0.19, Nodegroup: 0, Master)
id=3 @192.168.10.102 (Version: 5.0.19, Nodegroup: 0)
[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.10.104 (Version: 5.0.19)
[mysqld(API)] 4 node(s)
id=4 @192.168.10.103 (Version: 5.0.19)
id=5 @192.168.10.105 (Version: 5.0.19)
id=6 (NOT connected, accepting connect FROM any host)
id=7 (NOT connected, accepting connect FROM any host)
ndb_mgm> 2 stop
Node 2 has shutdown.

```

Ahora determine si los datos SQL fueron replicados cuando el engine de almacenamiento fue reiniciado, realice una consulta sobre uno o ambos hosts:

```

mysql-ndb-1$ mysql -u root
Welcome to the MySQL monitor. Commands END with ; OR \g.
Your MySQL connection id IS 5 to server version: 5.0.19
Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.
mysql> USE ejemplo;
Reading table information for completion of table AND COLUMN names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> SELECT * FROM prueba2;
+-----+
| i |
+-----+
| 2 |
+-----+
1 row IN SET (0.01 sec)

```

```

mysql-ndb-2$ mysql -u root
Welcome to the MySQL monitor. Commands END with ; OR \g.
Your MySQL connection id IS 5 to server version: 5.0.19
Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.
mysql> USE ejemplo;
Reading table information for completion of table AND COLUMN names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> SELECT * FROM prueba2;
+-----+
| i |
+-----+
| 2 |
+-----+
1 row IN SET (0.01 sec)

```

Esto es una muestra de que en realidad los datos estan siendo replicados sobre ambos nodos de almacenamiento. Arranque de nuevo el engine de almacenamiento sobre el primer nodo de almacenamiento (**mysql-ndb-1**):

```
mysql-ndb-1$ /usr/sbin/ndbd
```

Paso 6

Lo siguiente, agregaremos un nodo API. Este nodo es un miembro completo del cluster, pero no corre el engine del almacenamiento NDB. Los datos no seran replicados sobre este nodo, y su funcionalidad es esencialmente como la de un cliente del servidor cluster. Tipicamente, podriamos instalar aplicaciones que requieran acceder a datos de mysql (web servers, servicios de mensajeria empresarial, etc) sobre esta maquina. Las aplicaciones platicaran con el servidor mysql sobre el "**localhost**", el cual es un api node que

tomara a su cargo la comunicacion con el cluster en relacion a los datos solicitados.
Primero, instale el servidor mysql sobre el nodo API numero 1 (192.168.10.103):

```
root@mysql-api-1:~/ $ rpm -Uvh /root/mysql-rpm-cluster/<paquete-rpm>
```

Instala un simple archivo `"/etc/my.cnf"`:

```
[mysqld]
ndbcluster
ndb-connectstring='host=192.168.10.104' # Direccion IP del servidor de administracion mgmt
[mysql_cluster]
ndb-connectstring='host=192.168.10.104' # Direccion IP del servidor de administracion mgmt
```

Ahora inicie el servidor mysql:

```
mysql-api-1$ /etc/init.d/mysql start
```

Realice algunas consultas de prueba sobre las tablas de prueba que fueron creadas posteriormente:

```
mysql-api-1$ mysql -u root
Welcome to the MySQL monitor. Commands END with ; OR \g.
Your MySQL connection id IS 1 to server version: 5.0.19
Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.
mysql> CREATE DATABASE ejemplo;
Query OK, 1 row affected (0.11 sec)
mysql> USE ejemplo;
Reading table information for completion of table AND COLUMN names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> SELECT * FROM prueba1;
+----+
| i  |
+----+
|  1 |
+----+
1 row IN SET (0.01 sec)
mysql> SELECT * FROM prueba2;
+----+
| i  |
+----+
|  2 |
+----+
1 row IN SET (0.01 sec)
```

En este punto ya puedes checar el estado del cluster sobre la consola de administracion y verificar que el nodo API esta ya conectado:

```
ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.10.101 (Version: 5.0.19, Nodegroup: 0)
id=3 @192.168.10.102 (Version: 5.0.19, Nodegroup: 0, Master)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.10.104 (Version: 5.0.19)

[mysqld(API)] 4 node(s)
id=4 (Version: 5.0.19)
id=5 (Version: 5.0.19)
id=6 @192.168.10.103 (Version: 5.0.19)
id=7 (NOT connected, accepting connect FROM any host)
```

Nuestra configuracion ahora ya es lo que se muestra en la parte superior de este documento.

Paso 7

Finalmente, verificaremos la tolerancia a fallos de el cluster cuando se reciben querys de el nodo API.

Con el cluster arriba operando correctamente, use el nodo API para crear una nueva tabla e inserta algunos datos de prueba:

```
mysql-api-1$ mysql -u root
Welcome to the MySQL monitor.  Commands END with ; OR \g.
Your MySQL connection id IS 258519 to server version: 5.0.19

Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> USE ejemplo;
Reading table information for completion of table AND COLUMN names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE TABLE prueba3 (i INT) engine=ndbcluster;
Query OK, 0 rows affected (0.81 sec)

mysql> quit
Bye
```

Ahora, insertaremos algunos datos aleatorios dentro de la tabla prueba3, para no hacerlo a mano, usaremos un script hibrido para hacerlo:

```
#!/bin/sh
for i in 1 2 3 4 5 6 7 8 9 10
do
    random=`perl -e "print int(rand(100));"`
    echo "use ejemplo; insert into prueba3 () values ($random);" | mysql -u root
done
```

Probemos un consulta sobre el nodo API:

```
mysql-api-1$ mysql -u root
Welcome to the MySQL monitor.  Commands END with ; OR \g.
Your MySQL connection id IS 258551 to server version: 5.0.19

Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> USE ejemplo;
Reading table information for completion of table AND COLUMN names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM prueba3;
+----+
| i  |
+----+
| 92 |
| 20 |
| 18 |
| 84 |
| 49 |
| 22 |
| 54 |
| 91 |
| 79 |
| 52 |
+----+
10 rows IN SET (0.02 sec)
```

Se mira bien !. Ahora desconectemos el cable de la tarjeta de red del primer nodo de almacenamiento(**mysql-ndb-1**), de esa manera provocamos una falla en el cluster. Solo pasan unos cuantos segundos para que la consola de administracion (**mgmt**) reconosca que este ha desaparecido:

```
ndb_mgm> SHOW
Cluster Configuration
+-----+
|ndbd(NDB)| 2 node(s)
id=2 (NOT connected, accepting connect FROM 192.168.10.101)
id=3 @192.168.10.102 (Version: 5.0.19, Nodegroup: 0)

|ndb_mgmd(MGM)| 1 node(s)
id=1 @192.168.10.104 (Version: 5.0.19)

|mysqld(API)| 4 node(s)
id=4 (NOT connected, accepting connect FROM any host)
id=5 (Version: 5.0.19)
id=6 @192.168.10.103 (Version: 5.0.19)
id=7 (NOT connected, accepting connect FROM any host)
```

Estan los datos del cluster aun disponibles sobre el nodo API?

```
mysql-api-1# mysql -u root
Welcome to the MySQL monitor.  Commands END with ; OR \g.
Your MySQL connection id IS 258552 to server version: 5.0.19

Type 'help;' OR '\h' for HELP. Type '\c' to clear the buffer.

mysql> USE ejemplo;
Reading table information for completion of table AND COLUMN names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM prueba3;
+-----+
| i     |
+-----+
| 54    |
| 91    |
| 79    |
| 52    |
| 92    |
| 20    |
| 18    |
| 84    |
| 49    |
| 22    |
+-----+
10 rows IN SET (0.02 sec)
```

Ahora conecta de nuevo a la red al desconectado nodo de almacenamiento uno(**mysql-ndb-1**). Esto reincorpora el nodo al cluster, pero probablemente se encuentre apagado por el servidor de administracion, y algunos mensajes apareceran en el log de error (/var/lib/mysql-cluster/ndb_2_error.log):

```
-----
Date/Time: Wednesday 02 January 2008 - 12:46:21
Type of error: error
Message: Arbitrator shutdown
Fault ID: 2305
Problem data: Arbitrator decided to shutdown this node
Object of reference: QMGR (Line: 3796) 0x0000000a
ProgramName: /usr/sbin/ndbd
ProcessID: 1185
TraceFile: /var/lib/mysql-cluster/ndb_2_trace.log.3
***EOM***
-----
```

Reinicie el engine de almacenamiento ndb sobre este nodo y verifique que el nodo se unio de nuevo al cluster apropiadamente:

```
mysql-ndb-1$ /usr/sbin/ndbd
ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.10.101 (Version: 5.0.19, Nodegroup: 0)
id=3 @192.168.10.102 (Version: 5.0.19, Nodegroup: 0, Master)
[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.10.104 (Version: 5.0.19)
[mysqld(API)] 4 node(s)
id=4 (Version: 5.0.19)
id=5 (Version: 5.0.19)
id=6 @192.168.10.103 (Version: 5.0.19)
id=7 (NOT connected, accepting connect FROM any host)
```

Obtenido de "[http://172.16.161.128/wiki/index.php/Cluster Mysql 5.X](http://172.16.161.128/wiki/index.php/Cluster_Mysql_5.X)"

- Esta página fue modificada por última vez el 16:43, 28 ene 2008.