

# Mysql 集群搭建手册 v1.1

修订时间	修订范围	修订内容	修订作者	联系方式
2013-01-19	P1~P37	初稿	ClovenFong	cloven.feng@gmail.com
2013-03-18	P38~P41	双管理节点 && 多实例	ClovenFong	cloven.feng@gmail.com

## 目录

Mysql 集群搭建手册 v1.1 .....	1
前言 .....	3
一. 集群架构原理 .....	4
二. 节点分布情况 .....	5
三. 集群安装配置 .....	6
3.1 配置 mgmd 节点 .....	8
3.2 配置 ndbd 节点 .....	9
3.3 配置 sql 节点 .....	10
四. 集群功能测试 .....	12
4.1 数据同步 .....	12
4.2 故障测试 .....	14
4.3 添加节点 .....	15
五. 集群管理运维 .....	26
5.1 集群启动与关闭 .....	26
5.2 数据备份与恢复 .....	27
六. 集群故障排查 .....	32
七. 所有脚本文件 .....	35
八. 配置双管理节点 .....	38
8.1 host 配置文件 .....	38
8.2 管理节点配置文件 .....	39
8.3 关于 ndbd 以及 mysqld 配置文件 .....	39
8.4 配置多实例 .....	40
8.5 停止一个管理节点 .....	41

## 前言

Mysql 数据库作为开源数据库中比较成功的代表，在很多功能上面已经非常完美，特别是其复制功能实现了用户数据的高度冗余，但是即使复制过程再快，从复制的原理来看，复制永远不能摆脱异步复制这个特性，即使是最新的半同步复制也不能完美解决这个问题，只要是异步复制，就有可能造成数据的不统一，所以我们需要寻求一种解决方案，实现各节点之间数据的完全实时同步，那就是 Mysql 的集群技术，本文将从集群的架构原理，集群系统的配置搭建以及管理来深入理解 Mysql 集群是如何工作的，有了对其的深入理解，我们才能够根据我们实际生产环境中的场景设计 Mysql 的架构模式，才能真正实现整套系统的最佳性能以及高可用性。

可能很多人对 Mysql 的复制已经非常了解，我们在此不妨再共同梳理一下复制的基本原理以及流程，以便跟集群架构作出对比。首先，复制功能是需要借助于二进制日志(binlog)功能才能够实现,为方便理解与交流，如下图所示

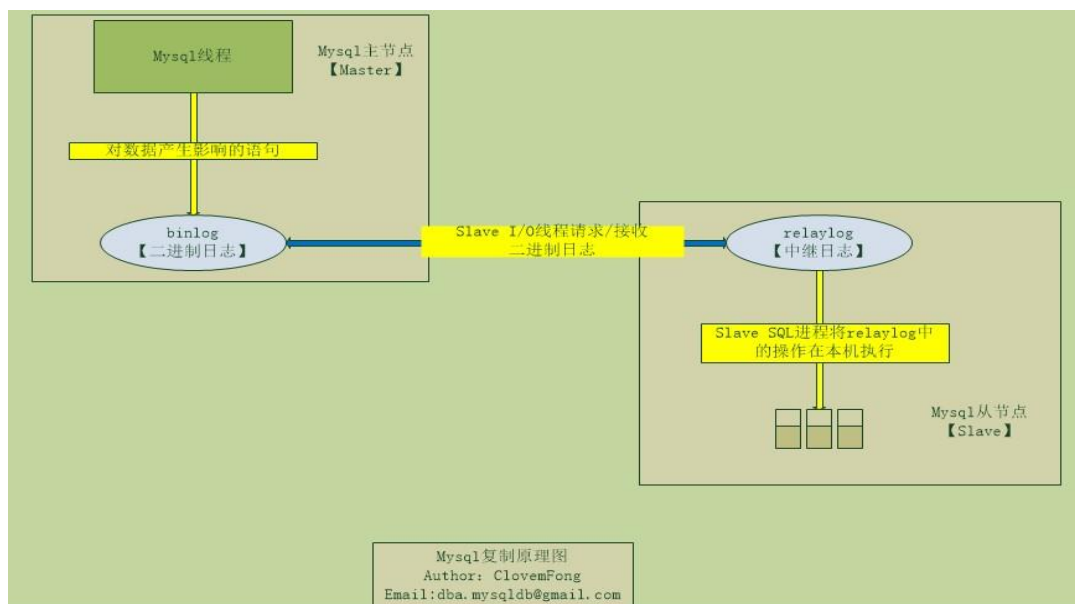


图 Mysql 复制原理图

该图比较简明得画出了 Mysql 复制中主从节点通过二进制日志文件如何进行复制的过程，主节点将对数据产生影响的语句写进本地二进制日志文件(如果二进制日志格式为 ROW 的话则是将变化的行写入二进制日志)，从节点上面的 I/O 线程通过请求与接收操作将主节点的二进制日志文件拉回本地，生成中继日志，再通过 SQL 进程将中继日志中的语句进行解析并执行，由此保证了两个节点数据的相对时间内的一致性。

以上为 Mysql 的复制原理，而关于复制跟集群的区别又在哪里呢？大致为如下两点：

1. 复制是一个服务器向另外一个服务器发送消息（数据）的方式
2. 集群包含成员，消息，冗余和自动故障转移等功能

最重要的一点：集群是无共享存储，却能保证各节点的数据实时同步。

以下正文便进入了我们对 Mysql 集群的学习与探讨,通过对集群的不断了解,我们可以将集群与复制功能同用，共同提高系统高可用性。

## 一. 集群架构原理

下面这个图是从官方文档拿来的,可以非常有层次的列出了不同节点类型之间的关系,在集群中,节点的概念并非以实际物理机器或者虚拟机,而是以集群中不同组件,如 SQL 节点,管理节点,存储节点来统计的,所以在一个物理机或者虚拟机上面可以有 N 个节点,比如同时拥有 SQL 节点,管理节点或者存储节点,但是在生产环境中,尽量保证每个节点对应一台机器,否则集群的功能就完全体现不出来了。

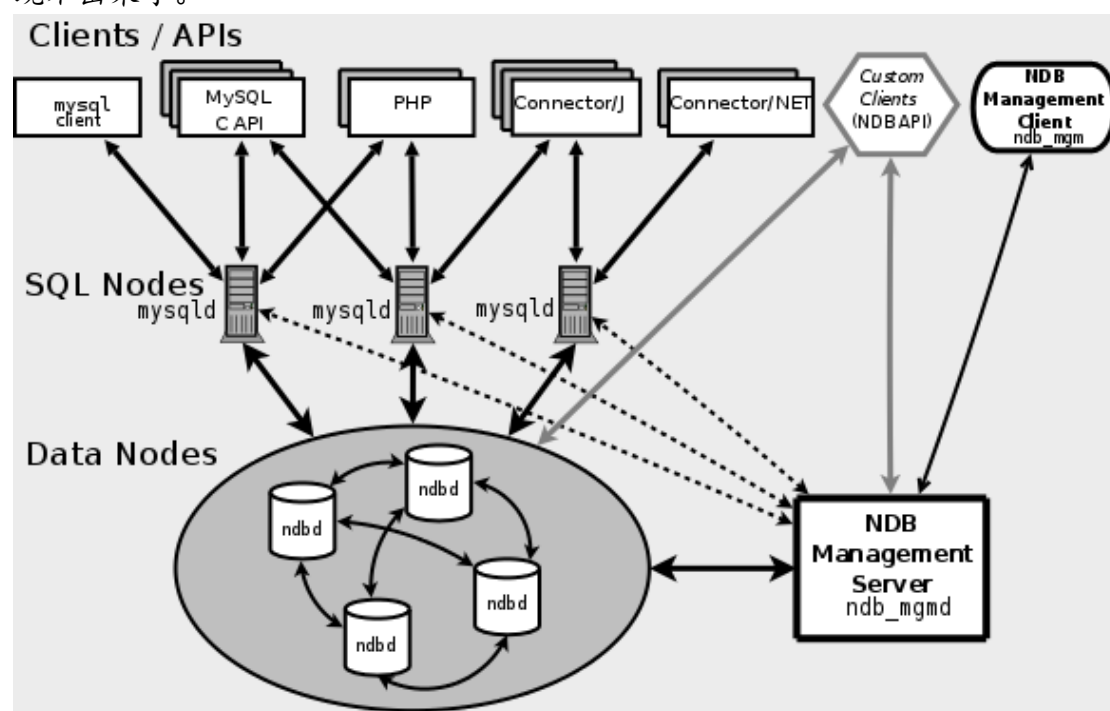


图 Mysql 集群架构图

分为四层：

各层类型	功能
应用程序层	负责与 Mysql 服务器通信的各种应用程序
MYSQL 服务器层	处理 SQL 命令，并与 NDB 存储引擎通信的 MYSQL 服务器
NDB 集群组件层	NDB 集群组件有时也称为数据节点，负责处理查询，然后将结果返回非 MYSQL 服务器
集群管理层	负责对整个集群进行监控，管理

图 集群各层功能

集群提供的功能大致如下：

1. 节点恢复

2. 日志
3. 检查点：
  - 本地检查点【用于清除部分重做日志】：本地检查点删除日志文件的尾部，当所有数据节点的日志都被刷新到磁盘的时候讲创建全局检查点
  - 全局检查点【用于不同数据节点之间的同步】：在磁盘上创建一个所有节点数据的事务一致性快照
4. 系统恢复
5. 热备份以及恢复：创建每个数据节点的同步备份，包括数据库中的对象的元数据，数据本身和当前事务日志。
6. 无单点故障
7. 故障转移
8. 数据分区
9. 联机操作：进行维护工作，正常关闭服务器或者给数据加锁，练级添加新的数据节点，改变表结构，甚至重组集群中的数据。

## 二. 节点分布情况

定义： clusterdir=/opt/mysql\_cluster  
 datadir=/opt/mysql\_cluster/data

操作系统: Red Hat Enterprise Linux Server release 6.3 (Santiago) 64bit

主机名	IP 地址	类型	配置文件	数据目录
mgmd	192.168.9.100	mgmd	\$clusterdir/config.ini	\$clusterdir/logs
ndbd01	192.168.9.101	ndbd	\$clusterdir /my.cnf	\$clusterdir/ndbdata
ndbd02	192.168.9.102	ndbd	\$clusterdir /my.cnf	\$clusterdir/ndbdata
mysqld01	192.168.9.201	sqld	/etc/my.cnf	\$datadir
mysqld02	192.168.9.202	sqld	/etc/my.cnf	\$datadir

每个实验节点的配置：

1. 内存：512MB
  2. 硬盘：40GB
- 生产环境以实际情况为标准。

### 三. 集群安装配置

配置前，声明一下所有节点均需要注意的设置，就是所有节点的 iptables 均需要关闭或者设置适当的规则，否则在各个节点启动或者执行的过程中均会出现非常纠结的问题。所以在每个节点上面均关闭 iptables 以及 selinux 为好，根据笔者实验，总结以下两点：

- 1.如果 mgmd 节点不关闭 iptables，则会出现 ndbd 节点启动的时候无法连接
- 2.如果 ndbd 节点不关闭 iptables，则 sql 节点在连接过程中会出现一系列问题列出几点错误，以便各位在搭建之前可以先了解，届时可以顺利解决该问题。

【1】首先在 Mgm 客户端通过 SHOW 命令查看的时候  
mysqld(API)总是不能同时在线，会出现如：

```
ndb_mgm> SHOW
^[[ACluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 (not connected, accepting connect from mysqld01)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)

ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 (not connected, accepting connect from mysqld02)
```

图 SQL 节点连接不稳定

【2】每个 SQL 节点上面进行数据操作的时候均会变得巨慢无比

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| clovem02 |
| mysql |
| ndbinfo |
| performance_schema |
| test |
+-----+
6 rows in set (0.00 sec)

mysql> DROP DATABASE clovem021
-> \c
mysql> DROP DATABASE clovem02;
Query OK, 0 rows affected (3 min 31.28 sec)
```

图 SQL 数据操作极慢

```
mysql> create database xx;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id:      5
Current database: *** NONE ***

Query OK, 1 row affected (2 min 13.47 sec)

mysql> use test
Database changed
mysql> SHOW TABLES;
Empty set (2 min 45.27 sec)

mysql> SHOW TABLES;
Empty set (0.02 sec)
```

图 操作数据库时间

### 【3】SQL 节点持续报错

```
[root@mysql01 ~]# tail -f /opt/mysql_cluster/data/mysql01.err
130115 17:05:48 [Warning] create index stats listener failed: error 4716 line 2483
130115 17:05:49 [Warning] create index stats listener failed: error 4716 line 2483
130115 17:05:50 [Warning] create index stats listener failed: error 4716 line 2483
130115 17:05:51 [Warning] create index stats listener failed: error 4716 line 2483
130115 17:05:52 [Warning] create index stats listener failed: error 4716 line 2483
130115 17:05:53 [Warning] create index stats listener failed: error 4716 line 2483
130115 17:05:54 [Warning] create index stats listener failed: error 4716 line 2483
130115 17:05:55 [Warning] create index stats listener failed: error 4716 line 2483
130115 17:05:56 [Warning] create index stats listener failed: error 4716 line 2483
```

图 SQL 节点报错

关闭防火墙之后:

```
ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
```

图 集群节点均正常连接

SQL 节点也不错报错了:

```
[root@mysql01 ~]# tail -f /opt/mysql_cluster/data/mysql01.err
2013-01-15 17:28:35 [NdbApi] INFO -- Flushing incomplete GCI:s < 3568/0
2013-01-15 17:28:35 [NdbApi] INFO -- Flushing incomplete GCI:s < 3568/0
130115 17:28:35 [Note] NDB Binlog: starting log at epoch 3568/0
130115 17:28:35 [Note] NDB Binlog: ndb tables writable
130115 17:28:35 [Note] NDB Binlog: Node: 11, subscribe from node 22, Subscriber bitmask 04000000
130115 17:28:35 [Note] NDB Binlog: Node: 12, subscribe from node 22, Subscriber bitmask 04000000
130115 17:30:40 [Note] NDB Binlog: Node: 12, unsubscribe from node 22, Subscriber bitmask 00
130115 17:30:40 [Note] NDB Binlog: Node: 11, unsubscribe from node 22, Subscriber bitmask 00
130115 17:30:48 [Note] NDB Binlog: Node: 11, subscribe from node 22, Subscriber bitmask 04000000
130115 17:30:48 [Note] NDB Binlog: Node: 12, subscribe from node 22, Subscriber bitmask 04000000
```

图 SQL 节点显示连接的 NDBD 节点

关于防火墙引起的问题已经提前声明了, 在后面的配置中由此引起相同的问题只要关闭防火墙即可, 所以在启动各个服务之前首先检查当前 ipables 状态。

### 3.1 配置 mgmd 节点

假设 [mysql-cluster-gpl-7.2.10-linux2.6-x86\\_64.tar.gz](#) 已经下载完成并置于/tmp 目录下方。

```
[root@mgmd tmp]# ll mysql-cluster-gpl-7.2.10-linux2.6-x86_64.tar.gz
-rw-r--r--. 1 root root 325846478 Jan  7 20:01 mysql-cluster-gpl-7.2.10-linux2.6-x86_64.tar.gz
[root@mgmd tmp]# useradd mysql
[root@mgmd tmp]# tar zxvf mysql-cluster-gpl-7.2.10-linux2.6-x86_64.tar.gz > /dev/null 2>&1
[root@mgmd tmp]# mv mysql-cluster-gpl-7.2.10-linux2.6-x86_64 /opt/mysql_cluster
[root@mgmd tmp]# chown mysql:mysql -R /opt/mysql_cluster/
```

```
[root@mgmd tmp]# cat /etc/hosts
```

```
192.168.9.100  mgmd
192.168.9.101  ndbd01
192.168.9.102  ndbd02
192.168.9.201  mysqld01
192.168.9.202  mysqld02
```

```
[root@mgmd ~]# mkdir /opt/mysql_cluster/logs
```

```
[root@mgmd ~]# chown mysql:mysql /opt/mysql_cluster/logs/ -R
```

```
[root@mgmd ~]# cat /opt/mysql_cluster/config.ini
```

```
[NDBD DEFAULT]
NoOfReplicas=2

[NDB_MGMD]
NodeId=1
hostname=mgmd
datadir=/opt/mysql_cluster/logs

[NDBD]
NodeId=11
hostname=ndbd01
datadir=/opt/mysql_cluster/ndbdata
[NDBD]
NodeId=12
hostname=ndbd02
datadir=/opt/mysql_cluster/ndbdata

[MYSQLD]
NodeId=21
hostname=mysqld01
[MYSQLD]
NodeId=22
hostname=mysqld02
```



```

[root@mgmd ~]# /opt/mysql_cluster/bin/ndb_mgmd -f /opt/mysql_cluster/config.ini --configdir=/opt/mysql_cluster/
MySQL Cluster Management Server mysql-5.5.29 ndb-7.2.10
[root@mgmd ~]# /opt/mysql_cluster/bin/ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 (not connected, accepting connect from ndbd01)
id=12 (not connected, accepting connect from ndbd02)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 (not connected, accepting connect from mysqld01)
id=22 (not connected, accepting connect from mysqld02)

ndb_mgm>

```

图 mgmd 节点启动成功并加入集群

```

[root@mgmd logs]# ls
ndb_1_cluster.log ndb_1_out.log ndb_1.pid
[root@mgmd logs]# cd ..
[root@mgmd mysql_cluster]# ls
bin config.ini COPYING data docs include INSTALL-BINARY lib logs man mysql-test ndb_1_config.bin.1

```

图 查看生成的日志文件以及 Pid 文件

## 3.2 配置 ndbd 节点

创建 mysql 用户，解压重命名集群目录与 mgmd 一致。  
其配置文件有所不同

```

[root@ndbd01 ~]# cat /opt/mysql_cluster/my.cnf
[mysqld]
ndbcluster
ndb-connectstring=mgmd
[mysql_cluster]
ndb-connectstring=mgmd
[root@ndbd01 ~]# ssh ndbd02 cat /opt/mysql_cluster/my.cnf
root@ndbd02's password:
[mysqld]
ndbcluster
ndb-connectstring=mgmd
[mysql_cluster]
ndb-connectstring=mgmd

```

图 两个 ndbd 节点的配置文件，一致

```

[root@ndbd01 ~]# mkdir /opt/mysql_cluster/ndbdata/
[root@ndbd01 ~]# chown -R mysql:mysql /opt/mysql_cluster/ndbdata/

```

首次启动 ndbd 节点的时候需要加上--initial-start 参数，表示初始化启动过程，后面的操作均不需要再次执行，否则会摧毁一切数据。

```

[root@ndbd01 ~]# /opt/mysql_cluster/bin/ndbd --defaults-file=/opt/mysql_cluster/my.cnf --initial-start
2013-01-16 05:30:17 [ndbd] INFO -- Angel connected to 'mgmd:1186'
2013-01-16 05:30:17 [ndbd] INFO -- Angel allocated nodeid: 11
[root@ndbd01 ~]# ssh ndbd02
root@ndbd02's password:
Last login: Wed Jan 16 07:19:00 2013
[root@ndbd02 ~]# /opt/mysql_cluster/bin/ndbd --defaults-file=/opt/mysql_cluster/my.cnf --initial-start
2013-01-16 07:24:11 [ndbd] INFO -- Angel connected to 'mgmd:1186'
2013-01-16 07:24:11 [ndbd] INFO -- Angel allocated nodeid: 12
[root@ndbd02 ~]# ssh mgmd
root@mgmd's password:
Last login: Wed Jan 16 04:26:58 2013 from 192.168.248.1
[root@mgmd ~]# /opt/mysql_cluster/bin/ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 (not connected, accepting connect from mysqld01)
id=22 (not connected, accepting connect from mysqld02)
ndb_mgm>

```

管理客户端显示ndb节点均加入集群，并且自动选举出主节点

图 启动 ndbd 节点

```

[root@ndbd01 ~]# ls /opt/mysql_cluster/ndbdata/
ndb 11 fs ndb 11 out.log ndb 11.pid

```

图 查看启动生成的文件

### 3.3 配置 sqld 节点

配置 sqld 节点的时候，配置文件 my.cnf 尽量放置于/etc/目录下，否则通过 mysql.server 脚本启动的时候会报错。

```

[root@mysqld01 ~]# /opt/mysql_cluster/scripts/mysql_install_db \
--user=mysql --basedir=/opt/mysql_cluster/ --datadir=/opt/mysql_cluster/data/

```

```

Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

/opt/mysql_cluster/bin/mysqladmin -u root password 'new-password'
/opt/mysql_cluster/bin/mysqladmin -u root -h mysqld01 password 'new-password'

Alternatively you can run:
/opt/mysql_cluster/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /opt/mysql_cluster/ ; /opt/mysql_cluster/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd /opt/mysql_cluster/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /opt/mysql_cluster/scripts/mysqlbug script!

```

图 初始化数据库文件

```
[root@mysqld01 ~]# cat /etc/my.cnf
```

```
[mysqld]
ndbcluster
ndb-connectstring=mgmd
[mysql_cluster]
ndb-connectstring=mgmd
```

```
[root@mysqld01 ~]# cat /opt/mysql_cluster/support-files/mysql.server |grep -v grep
|grep basedir |grep -v '#'|head -n 2
```

```
basedir=/opt/mysql_cluster
datadir=$basedir/data
```

```
[root@mysqld01 ~]# cp /opt/mysql_cluster/support-files/mysql.server /etc/init.d/mysqld
[root@mysqld01 ~]# chkconfig --add mysqld
[root@mysqld01 ~]# service mysqld start
Starting MySQL.... [ OK ]
```

图 将 mysql.server 设置为系统服务

```
[root@mysqld01 ~]# ps -ef |grep mysql |grep -v grep
avahi      1446      1  0 03:17 ?        00:00:00 avahi-daemon: running [mysql01.l
ocal]
root       3264      1  0 07:45 pts/0    00:00:00 /bin/sh /opt/mysql_cluster/bin/m
ysqld_safe --datadir=/opt/mysql_cluster/data --pid-file=/opt/mysql_cluster/data/
mysqld01.pid
mysql     3381    3264  2 07:45 pts/0    00:00:03 /opt/mysql_cluster/bin/mysqld --
basedir=/opt/mysql_cluster --datadir=/opt/mysql_cluster/data --plugin-dir=/opt/m
ysql_cluster/lib/plugin --user=mysql --log-error=/opt/mysql_cluster/data/mysqld0
1.err --pid-file=/opt/mysql_cluster/data/mysqld01.pid
```

图 查看 mysql 进程状态

```
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -e 'SHOW DATABASES;'
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| ndbinfo |
| performance_schema |
| test |
+-----+
```

图 查看数据库

另外一个节点 mysqld02 上面的操作跟 mysqld01 上完全一致。全部启动成功之后，登录管理节点查看集群中所有节点状态。

```
[root@mgmd ~]# /opt/mysql_cluster/bin/ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)

ndb_mgm> ALL STATUS
Node 11: started (mysql-5.5.29 ndb-7.2.10)
Node 12: started (mysql-5.5.29 ndb-7.2.10)
```

图 集群所有节点状态

## 四. 集群功能测试

根据以上的所有操作, 简单的 mysql 集群系统已经基本搭建完成, 均可正常工作。  
下面测试数据同步测试以及故障测试。

### 4.1 数据同步

```
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -e "set  
password=password('passw0rd')"
```

```
[root@mysqld02 ~]# /opt/mysql_cluster/bin/mysql -e "set  
password=password('passw0rd')"
```

```
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd  
--prompt="\u@\h:\d \r:\m:\s >"
```

```
root@localhost:(none) 08:48:31 > CREATE DATABASE clovemfong;  
Query OK, 1 row affected (0.04 sec)
```

```
root@localhost:(none) 08:48:31 > CREATE TABLE clovemfong.t1 (id int  
auto_increment,name varchar(30),PRIMARY KEY(id)) ENGINE=NDBCLUSTER;  
Query OK, 0 rows affected (0.02 sec)
```

```
[root@mysqld02 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd  
--prompt="\u@\h:\d \r:\m:\s >"
```

```

root@localhost:(none) 08:50:07 >SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| clovemfong |
| mysql |
| ndbinfo |
| performance_schema |
| test |
+-----+
6 rows in set (0.00 sec)

root@localhost:(none) 08:50:13 >USE clovemfong;
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
root@localhost:clovemfong 08:50:27 >SHOW TABLES;
+-----+
| Tables_in_clovemfong |
+-----+
| t1 |
+-----+
1 row in set (0.01 sec)

root@localhost:clovemfong 08:50:30 >DESC t1;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| name | varchar(30) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

图 在 mysqld02 节点查看被创建的表

```

root@localhost:clovemfong 08:50:40 >INSERT INTO t1(name) VALUES(1),(10),(100),(1000),(10000);
Query OK, 5 rows affected (0.03 sec)
Records: 5 Duplicates: 0 Warnings: 0

root@localhost:clovemfong 08:51:06 >SELECT * FROM t1;
+-----+-----+
| id | name |
+-----+-----+
| 5 | 10000 |
| 1 | 1 |
| 2 | 10 |
| 4 | 1000 |
| 3 | 100 |
+-----+-----+
5 rows in set (0.00 sec)

```

图 插入数据并查看

```

[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd clovemfong \
> -e "SELECT * FROM t1";
+----+-----+
| id | name |
+----+-----+
| 5 | 10000 |
| 1 | 1 |
| 2 | 10 |
| 4 | 1000 |
| 3 | 100 |
+----+-----+
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd clovemfong -e "DELETE FROM t1 WHERE id not in (2,4);";
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd clovemfong -e "SELECT * FROM t1";
+----+-----+
| id | name |
+----+-----+
| 2 | 10 |
| 4 | 1000 |
+----+-----+
[root@mysqld01 ~]# ssh mysqld02
root@mysqld02's password:
Last login: Wed Jan 16 07:40:20 2013 from 192.168.248.1
[root@mysqld02 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd clovemfong -e "SELECT * FROM t1";
+----+-----+
| id | name |
+----+-----+
| 2 | 10 |
| 4 | 1000 |
+----+-----+

```

图 删除数据，查看两边数据显示

## 4.2 故障测试

从 4.1 的测试我们得知在 Mysql 集群中，实现了任何 SQL 节点操作数据的一致性，当然前提是所有的表存储引擎需要设置为 NDBCLUSTER。

本节将对集群中可能出现的可能做个排列组合进行说明，其实想想也就知道了。

### 【1】mysqld01 挂掉

这样的情况对数据完全没有影响，因为数据时保存在 ndbd 节点上面的，该类节点挂掉之后，应用程序仅仅需要将请求转发给另一 mysqld 节点即可。

### 【2】ndbd01 挂掉

这样的情况，仅仅是数据丢掉了副本而已，也不会有影响。

### 【3】mysqld01 && ndbd01 同时 挂掉

此类情况对数据而言少了一个副本，mysqld 节点少一个，不影响。此时写入数据的时候，会出现两种情况，ndbd01 此时成功启动了，这种情况下面，ndbd02 会立即同步 ndbd01 上面的数据，此时 ndbd02 如果挂了，不会影响数据的一致性，第二种情况，写完了数据，在 ndbd01 重启启动完成之前 ndbd02 也挂了，此时刚刚写好的数据必定丢失，不过这种情况不多见，这种情况就相当于连个 ndbd 同时不工作了。

### 【4】mysqld01 && mysqld02 && ndbd01 同时挂掉。

在这样的情况下面，只要重启启动好某个 mysqld 节点或者添加新的 mysqld 节点即可，关于 ndbd 节点，同样也是，要么重启 ndbd01 节点，要么添加一台新的节点进入集群。由此引申如何动态添加一个新的节点到集群中去了。

## 4.3 添加节点

所有的集群系统中都会涉及到一个动态添加节点的问题,要在保证服务不被打断的情况下做到节点的动态添加是一个集群必备的功能。

在网络上面能够找到不少关于数据节点的添加,确实在实际生产环境中,动态添加数据节点是一个必须得面对的问题,而平时负载较大的 `mysqld` 节点也有面临宕机的情况,所以考虑 `mysqld` 节点的动态添加也是有必要的。

### 4.3.1 动态加入 `mysqld` 节点

【1】修改 `/etc/hosts` 文件并同步到所有节点

```
[root@mgmd ~]# cat /etc/hosts
192.168.9.100    mgmd
192.168.9.101    ndbd01
192.168.9.102    ndbd02
192.168.9.201    mysqld01
192.168.9.202    mysqld02
192.168.9.203    mysqld03
```

图 加入新节点

【2】修改 `config.ini` 文件

```
[root@mgmd ~]# tail -n 9 /opt/mysql_cluster/config.ini
[MYSQLD]
NodeId=21
hostname=mysqld01
[MYSQLD]
NodeId=22
hostname=mysqld02
[MYSQLD]
NodeId=23
hostname=mysqld03
```

图 加入新节点信息

【3】停止管理节点

```
[root@mgmd ~]# /opt/mysql_cluster/bin/ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd (NDB)] 2 node(s)
id=11 (not connected, accepting connect from ndbd01)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)

[ndb_mgmd (MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld (API)] 2 node(s)
id=21 (not connected, accepting connect from mysqld01)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)

ndb_mgm> 1 STOP
Node 1 has shutdown.
Disconnecting to allow Management Server to shutdown
ndb_mgm>
```

图 停止 Node 1—mgmd

## 【5】 重启加载配置文件

```
[root@mgmd ~]# /opt/mysql_cluster/bin/ndb_mgmd -f /opt/mysql_cluster/config.ini --configdir=/opt/mysql_cluster/ --reload
MySQL Cluster Management Server mysql-5.5.29 ndb-7.2.10
[root@mgmd ~]# /opt/mysql_cluster/bin/ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 (not connected, accepting connect from ndbd01)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 3 node(s)
id=21 (not connected, accepting connect from mysqld01)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 (not connected, accepting connect from mysqld03)

ndb_mgm>
```

图 重新加载管理节点配置文件

## 【6】 轮流重启 ndbd 节点

```
ndb_mgm> 11 restart
Node 11: Node shutdown initiated
Node 11: Node shutdown completed, restarting, no start.
Node 11 is being restarted

ndb_mgm> Node 11: Start initiated (version 7.2.10)

ndb_mgm> 12 restart
Restart failed.
* 2003: Restart failed
* 5063-Operation not allowed while nodes are starting or stopping.: Permanent error: Application error

ndb_mgm> Node 11: Started (version 7.2.10)

ndb_mgm> 12 restart
Node 12: Node shutdown initiated
Node 12: Node shutdown completed, restarting, no start.
Node 12 is being restarted

ndb_mgm>
```

图 轮流重启 ndbd 节点

## 【7】 启动新的 mysqld 节点

按照之前 mysqld 的启动流程启动即可

```
[root@mysqld03 ~]# /opt/mysql_cluster/scripts/mysql_install_db \
--user=mysql --basedir=/opt/mysql_cluster/ --datadir=/opt/mysql_cluster/data/
[root@mysqld03 data]# service mysqld restart
Shutting down MySQL.....[ OK ].....
Starting MySQL.....[ OK ]
```

## 【8】 查看状态

```
ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 3 node(s)
id=21 (not connected, accepting connect from mysqld01)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)
```

图 新节点添加成功



### 4.3.2 动态加入 ndbd 节点

根据实际项目，我们添加 ndbd 节点的场景可能有所不同，可以分为两种：

1. 添加一个或者多个节点作为新的副本，即加入原来的组中
  2. 添加 2 个或者其他偶数级别的节点作为新的组，用于分区使用。
- 一般从网络上面的文章来看第二种比较多，不过个人觉得第一种的情况非常大，因为很有可能我只想添加一台新的存储节点。

我们第一种开始进行操作：

#### 4.3.2.1 加入一个节点作为新的副本

关于这个问题，我通过实验已经得出结论。

1. 必须离线状态才可以进行该操作，即关闭整个集群
2. 原来 ndbd 节点由于检测到副本数[NoOfReplicas]需要修改重新启动的时候必须加 initial 参数，也就意味着摧毁所有原来的数据

从以上两点可以知晓，如果需要这样得添加节点，我们需要先进行数据的备份，然后再最终在所有的节点上面进行恢复才行。

#### 4.3.2 加入偶数节点作为新的分组，如添加 2 个

```
ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 3 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)

ndb_mgm> ALL STATUS
Node 11: started (mysql-5.5.29 ndb-7.2.10)
Node 12: started (mysql-5.5.29 ndb-7.2.10)

ndb_mgm>
```

图 当前节点信息

#### 【1】 修改/etc/hosts 文件并同步

```
[root@mgmd ~]# cat /etc/hosts
192.168.9.100 mgmd
192.168.9.101 ndbd01
192.168.9.102 ndbd02
192.168.9.103 ndbd03
192.168.9.104 ndbd04
192.168.9.201 mysqld01
192.168.9.202 mysqld02
192.168.9.203 mysqld03
[root@mgmd ~]# for i in `cat /etc/hosts |awk -F' ' '{print $2}'`; do scp /etc/hosts $i:/etc/hosts ; done
hosts
hosts
hosts
hosts
hosts
hosts
hosts
[root@mgmd ~]#
```

图 添加新节点信息

## 【2】 修改集群配置文件 config.ini

```
[root@mgmd ~]# cat /opt/mysql_cluster/config.ini |tail -n 26 |head -n 16
[NDBD]
NodeId=11
hostname=ndbd01
datadir=/opt/mysql_cluster/ndbdata
[NDBD]
NodeId=12
hostname=ndbd02
datadir=/opt/mysql_cluster/ndbdata
[NDBD]
NodeId=13
hostname=ndbd03
datadir=/opt/mysql_cluster/ndbdata
[NDBD]
NodeId=14
hostname=ndbd04
datadir=/opt/mysql_cluster/ndbdata
```

图 加入新节点

## 【3】 停止管理节点

```
[root@mgmd ~]# ndb_mgm -e 1 STOP
Unable to connect with connect string: nodeid=0,STOP:1186
Retrying every 5 seconds. Attempts left: 2
```

图 停止管理节点

## 【4】 重载集群配置文件

```
ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 3 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)

ndb_mgm> 1 STOP
Node 1 has shutdown.
Disconnecting to allow Management Server to shutdown

ndb_mgm> quit
[root@mgmd ~]# /opt/mysql_cluster/bin/ndb_mgmd -f /opt/mysql_cluster/config.ini --configdir=/opt/mysql_cluster --reload
MySQL Cluster Management Server mysql-5.5.29 ndb-7.2.10
[root@mgmd ~]# ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 4 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)
id=13 (not connected, accepting connect from ndbd03)
id=14 (not connected, accepting connect from ndbd04)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 3 node(s)
```

图 重载配置文件

#### 【4】 轮流重启 ndbd 节点

在重启之前，我们首先创建点测试数据：

```
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd
```

```
mysql> CREATE DATABASE db01;
Query OK, 1 row affected (0.17 sec)

mysql> CREATE TABLE db01.t1(id INT AUTO_INCREMENT PRIMARY KEY,name VARCHAR(30)) ENGINE=NDBCLUSTER;
Query OK, 0 rows affected (0.28 sec)

mysql> INSERT INTO db01.t1(name) VALUES('fengyingcong'),('liuhaicheng'),('jiangkun');
Query OK, 3 rows affected (0.11 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
[root@mysqld02 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd -e "SELECT *
FROM db01.t1;"
```

```
+----+-----+
| id | name      |
+----+-----+
| 1  | fengyingcong |
| 2  | liuhaicheng  |
| 3  | jiangkun    |
+----+-----+
```

```
[root@mysqld03 ~]# /opt/mysql_cluster/bin/mysqladmin password passwd
```

```
[root@mysqld03 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd -e "SELECT *
FROM db01.t1;"
```

```
+----+-----+
| id | name      |
+----+-----+
| 1  | fengyingcong |
| 2  | liuhaicheng  |
| 3  | jiangkun    |
+----+-----+
```

现在可以轮流重启 ndbd 节点了：

```
ndb_mgm> 11 restart
Node 11: Node shutdown initiated
Node 11: Node shutdown completed, restarting, no start.
Node 11 is being restarted

ndb_mgm> Node 11: Start initiated (version 7.2.10)

ndb_mgm> Node 11: Started (version 7.2.10)
12 restart
Node 12: Node shutdown initiated
Node 12: Node shutdown completed, restarting, no start.
Node 12 is being restarted

ndb_mgm> Node 12: Started (version 7.2.10)

ndb_mgm> ALL STATUS
Node 11: started (mysql-5.5.29 ndb-7.2.10)
Node 12: started (mysql-5.5.29 ndb-7.2.10)
Node 13: not connected
Node 14: not connected

ndb_mgm>
```

图 重启已有的 ndbd 节点

## 【5】 重启 mysqld 节点

```
[root@mgmd ~]# cat /etc/hosts |grep mysqld |awk -F' ' '{print $2}'
mysqld01
mysqld02
mysqld03
[root@mgmd ~]# for i in `cat /etc/hosts |grep mysqld |awk -F' ' '{print $2}'`
> do ssh $i service mysqld restart &
> done
[5] 6710
[6] 6711
[7] 6712
[root@mgmd ~]# Shutting down MySQL.Shutting down MySQL.Shutting down MySQL.....[ OK ]
Starting MySQL.[ OK ]
Starting MySQL.[ OK ]
Starting MySQL.....[ OK ]
[ OK ]
.[ OK ]

[5] Done          ssh $i service mysqld restart
[6] Done          ssh $i service mysqld restart
[7] Done          ssh $i service mysqld restart
```

图 重启 mysqld 节点

## 【6】 初始化新 ndbd 节点

```
[root@mgmd ~]# ssh ndbd03 "/opt/mysql_cluster/bin/ndbd --defaults-file=/opt/mysql_cluster/my.cnf --initial"
2013-01-16 18:23:26 [ndbd] INFO -- Angel connected to 'mgmd:1186'
2013-01-16 18:23:26 [ndbd] INFO -- Angel allocated nodeid: 13
[root@mgmd ~]# ssh ndbd04 "/opt/mysql_cluster/bin/ndbd --defaults-file=/opt/mysql_cluster/my.cnf --initial"
2013-01-16 18:23:33 [ndbd] INFO -- Angel connected to 'mgmd:1186'
2013-01-16 18:23:33 [ndbd] INFO -- Angel allocated nodeid: 14
```

图 初始化 ndbd03 && ndbd04

```
ndb_mgm> ALL STATUS
Node 11: started (mysql-5.5.29 ndb-7.2.10)
Node 12: started (mysql-5.5.29 ndb-7.2.10)
Node 13: not connected
Node 14: not connected

ndb_mgm> Node 13: Started (version 7.2.10)
Node 14: Started (version 7.2.10)

ndb_mgm> ALL STATUS
Node 11: started (mysql-5.5.29 ndb-7.2.10)
Node 12: started (mysql-5.5.29 ndb-7.2.10)
Node 13: started (mysql-5.5.29 ndb-7.2.10)
Node 14: started (mysql-5.5.29 ndb-7.2.10)

ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 4 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)
id=13 @192.168.9.103 (mysql-5.5.29 ndb-7.2.10, no nodegroup)
id=14 @192.168.9.104 (mysql-5.5.29 ndb-7.2.10, no nodegroup)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 3 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)
```

图 查看集群节点状态

此时我们可以看到所有的数据节点(ndbd 节点)均启动完成,但是新加入的两个节点没有节点组, 我们需要手动给其创建一个组。

## 【7】 创建节点组

```
ndb_mgm> CREATE NODEGROUP 13,14
Nodegroup 1 created
ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 4 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)
id=13 @192.168.9.103 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)
id=14 @192.168.9.104 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 3 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)
```

图 创建节点组 1，并加入新节点

## 【8】 重新分配集群数据

当新加入数据节点时，已经存在的数据或者索引不会被自动分配到新的节点上，，查看当前数据分布情况。

```
[root@mgmd ~]# ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> ALL REPORT MEMORY;
Connected to Management Server at: localhost:1186
Node 11: Data usage is 1%(33 32K pages of total 2560)
Node 11: Index usage is 1%(26 8K pages of total 2336)
Node 12: Data usage is 1%(33 32K pages of total 2560)
Node 12: Index usage is 1%(26 8K pages of total 2336)
Node 13: Data usage is 0%(16 32K pages of total 2560)
Node 13: Index usage is 0%(0 8K pages of total 2336)
Node 14: Data usage is 0%(16 32K pages of total 2560)
Node 14: Index usage is 0%(0 8K pages of total 2336)

ndb_mgm> █
```

图 数据分布信息

```
mysql> SELECT table_schema,table_name FROM information_schema.tables WHERE engine='ndbcluster';
+-----+-----+
| table_schema | table_name |
+-----+-----+
| db01         | t1         |
| mysql        | ndb_apply_status |
| mysql        | ndb_index_stat_head |
| mysql        | ndb_index_stat_sample |
+-----+-----+
4 rows in set (0.02 sec)

mysql> USE db01;
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> ALTER ONLINE TABLE t1 REORGANIZE PARTITION;
Query OK, 0 rows affected (0.35 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

图 重新组织分区

```
ndb_mgm> ALL REPORT MEMORY;
Node 11: Data usage is 1% (33 32K pages of total 2560)
Node 11: Index usage is 1% (26 8K pages of total 2336)
Node 12: Data usage is 1% (33 32K pages of total 2560)
Node 12: Index usage is 1% (26 8K pages of total 2336)
Node 13: Data usage is 0% (17 32K pages of total 2560)
Node 13: Index usage is 0% (0 8K pages of total 2336)
Node 14: Data usage is 0% (17 32K pages of total 2560)
Node 14: Index usage is 0% (0 8K pages of total 2336)

ndb_mgm> ALL REPORT MEMORY;
Node 11: Data usage is 1% (30 32K pages of total 2560)
Node 11: Index usage is 1% (26 8K pages of total 2336)
Node 12: Data usage is 1% (30 32K pages of total 2560)
Node 12: Index usage is 1% (26 8K pages of total 2336)
Node 13: Data usage is 0% (25 32K pages of total 2560)
Node 13: Index usage is 0% (4 8K pages of total 2336)
Node 14: Data usage is 0% (25 32K pages of total 2560)
Node 14: Index usage is 0% (4 8K pages of total 2336)
```

图 数据分区变化

因为我们的测试数据非常小，所以根据数据使用率不能看出效果，可以通过后面的数字进行判别是否进行了数据分区。

新节点加入之后，对原有的数据不会产生影响，只是将数据进行了分区处理，相当于每个数据均有两个副本，而每个副本均将数据分开存放到另外一个组里面的节点上面。

通过如下的测试可以看出数据在四个节点上面是如何分布工作的，每个组中的两个节点保存相同的副本，每个完整的数据都会分布在本节点以及另外组中的一个节点。下图为我们节点中数据分布的状态变化。

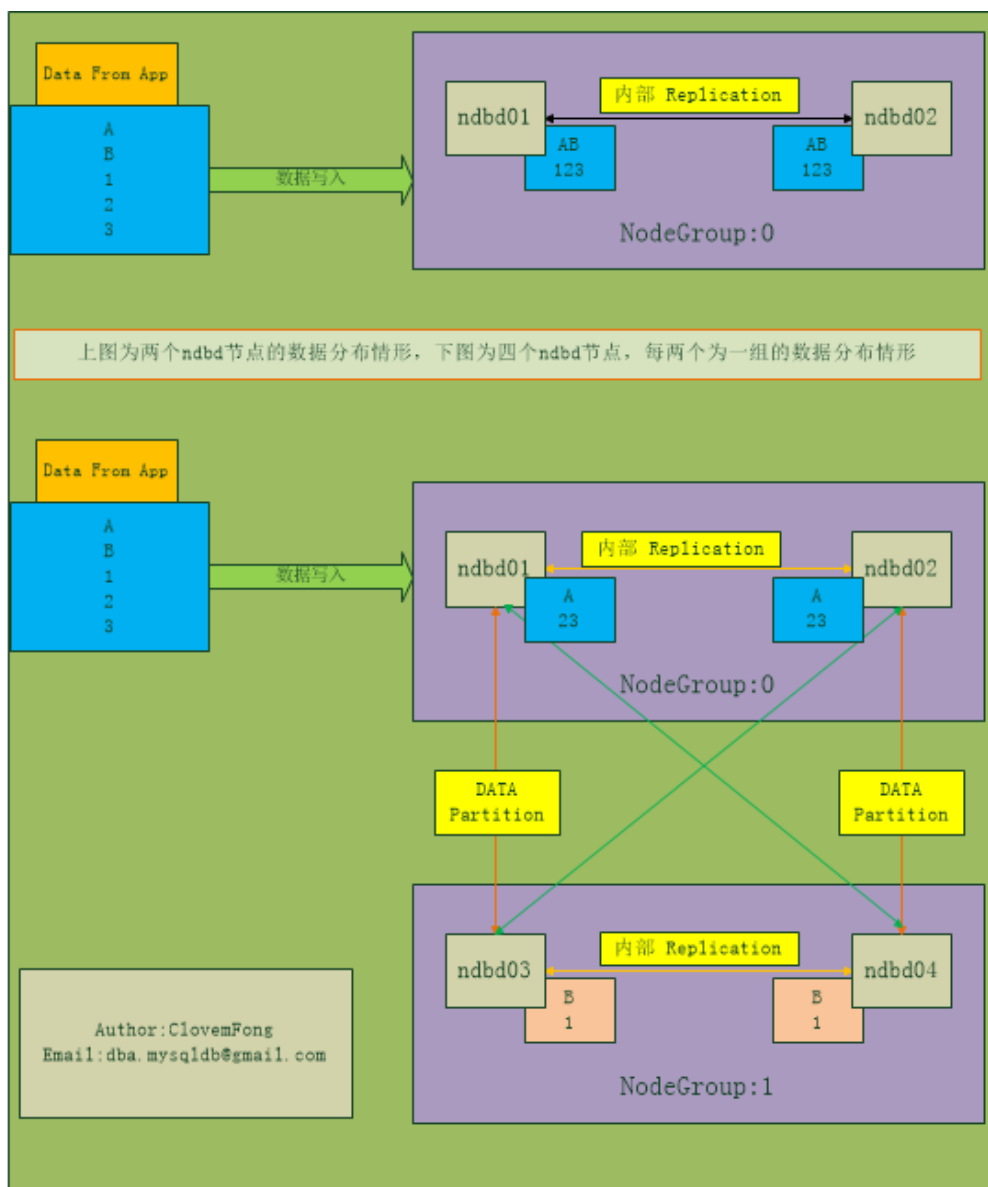


图 单组跟多组节点的数据分布

由上图【上下两张图】可以清楚得知道两种场景中能够维持集群正常运行的最少节点数。在第一张图中，由于数据有两个副本，分别位于两个存储节点之上，故允许一台的损坏，第二张图中，数据节点有了4个，但是只有组内的节点的副本是完全一样的，要保证数据的完整性，必须保证两个组中都至少需要一个存储节点能够正常工作，一旦出现一个组中所有的节点都宕机的情形，则集群失效。

下面的截图是用于证明如上所述的操作，在我们进行进行集群管理，手动对存储节点进行停止的时候，发现每个组中最后一个节点都不能够被关闭，提示会导致集群系统的 Crash.



```

ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 4 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)
id=13 @192.168.9.103 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)
id=14 @192.168.9.104 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 4 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)
id=24 (not connected, accepting connect from any host)

ndb_mgm> 11 STOP
Node 11: Node shutdown initiated
Node 11: Node shutdown completed.
Node 11 has shutdown.

ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 4 node(s)
id=11 (not connected, accepting connect from ndbd01)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=13 @192.168.9.103 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)
id=14 @192.168.9.104 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 4 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)

```

图 停止 NodeGroup:0 上面的一个节点

```

ndb_mgm> 12 STOP
Node 12: Node shutdown aborted
Shutdown failed.
* 2002: Stop failed
* Node shutdown would cause system crash: Permanent error: Application error

ndb_mgm> 13 STOP
Node 13: Node shutdown initiated
Node 13: Node shutdown completed.
Node 13 has shutdown.

ndb_mgm> ALL REPORT MEMORY
Node 12: Data usage is 6%(171 32K pages of total 2560)
Node 12: Index usage is 10%(245 8K pages of total 2336)
Node 14: Data usage is 6%(167 32K pages of total 2560)
Node 14: Index usage is 9%(226 8K pages of total 2336)

```

图 停止 NodeGroup:0 上面的另外一个节点



```

ndb_mgm> SHOW
Cluster Configuration

-----
[ndbd(NDB)] 4 node(s)
id=11 (not connected, accepting connect from ndbd01)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=13 (not connected, accepting connect from ndbd03)
id=14 @192.168.9.104 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 4 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)
id=24 (not connected, accepting connect from any host)

ndb_mgm> 14 STOP
Node 14: Node shutdown aborted
Shutdown failed.
* 2002: Stop failed
* Node shutdown would cause system crash: Permanent error: Application error

```

图 NodeGroup:1 中至少需要有一个正常节点

```

-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 4 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=13 @192.168.9.103 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)
id=14 @192.168.9.104 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 4 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)
id=24 (not connected, accepting connect from any host)

ndb_mgm> ALL STATUS
Node 11: started (mysql-5.5.29 ndb-7.2.10)
Node 12: started (mysql-5.5.29 ndb-7.2.10)
Node 13: started (mysql-5.5.29 ndb-7.2.10)
Node 14: started (mysql-5.5.29 ndb-7.2.10)

ndb_mgm> ALL REPORT MEMORY
Node 11: Data usage is 6%(172 32K pages of total 2560)
Node 11: Index usage is 10%(234 8K pages of total 2336)
Node 12: Data usage is 6%(171 32K pages of total 2560)
Node 12: Index usage is 10%(245 8K pages of total 2336)
Node 13: Data usage is 6%(167 32K pages of total 2560)
Node 13: Index usage is 9%(215 8K pages of total 2336)
Node 14: Data usage is 6%(167 32K pages of total 2560)
Node 14: Index usage is 9%(226 8K pages of total 2336)

```

图 当前集群所有节点信息

## 五. 集群管理运维

本章节共同学习如何对 Mysql 集群进行日常的管理，上面提到的动态添加节点也属于运维管理范畴，在上一章提到过加入单个数据节点到集群中去，需要离线整个集群系统，并且在启动数据节点的时候均需要初始化操作，由此带来了很大的风险，在数据库系统中，数据库的备份操作一直是非常重要的，本章会涉及到备份恢复相关操作。

### 5.1 集群启动与关闭

我们需要让一个集群上线，就需要了解其的启动过程，一般情况下面，Mysql 的启动流程如下：

【1】 启动管理节点守护进程： ndb\_mgmd

【2】 启动各个数据节点守护进程： ndbd

【3】 启动 SQL 节点的守护进程： mysqld

完成以上三步，集群就可以正常启动了，在启动过程中，可以通过查看每个节点的日志文件来解决启动过程中可能出现的问题：

mgmd: /opt/mysql\_cluster/logs/ndb\_id\_cluster.log

ndbd: /opt/mysql\_cluster/ndbdata/ndb\_id\_error.log

mysqld: /opt/mysql\_cluster/data/hostname.err

我们某些情况下面需要对整个系统进行维护操作，为了使得数据不再受到其他操作的影响，我们按照如下步骤对集群进行关闭操作：

【1】 关闭 SQL 节点守护进程： service mysqld stop

【2】 在管理客户端执行 SHUTDOWN 命令关闭所有节点

```
ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 4 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=13 @192.168.9.103 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)
id=14 @192.168.9.104 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 4 node(s)
id=21 (not connected, accepting connect from mysqld01)
id=22 (not connected, accepting connect from mysqld02)
id=23 (not connected, accepting connect from mysqld03)
id=24 (not connected, accepting connect from any host)

ndb_mgm> SHUTDOWN
Node 12: Cluster shutdown initiated
Node 13: Cluster shutdown initiated
Node 14: Cluster shutdown initiated
Node 11: Cluster shutdown initiated
Node 12: Node shutdown completed.
5 NDB Cluster node(s) have shutdown.
Disconnecting to allow management server to shutdown.
Node 14: Node shutdown completed.
ndb_mgm> QUIT
[root@mgmd ~]# lsof -i:1186
```

图 关闭所有节点

## 5.2 数据备份与恢复

在我们传统的 Mysql 数据库中，数据的备份以及恢复从通用角度来看，也就是针对大部分常用存储引擎，如 Myisam,Innodb 等，我们用到的备份命令多为：mysqldump 进行全局备份以及 select into outfile 进行表级别的备份或者通过 mysqlbinlog 进行更高级的 PITR（Point-In-Time-Recovery）的时间点恢复（可实现一定意义上的增量备份），而恢复命令多是通过：mysql 命令导入文件或者登录进数据库通过 source 命令使备份语句生效，也可以通过 load data infile，通过 mysqlbinlog 进行恢复等。

在集群系统中，就有一套自己的备份方法进行数据的备份以及恢复。这些操作均可以通过管理客户端登录管理节点执行。

我们可以为在管理节点配置文件 config.ini 中添加该参数

[ndbd]

.....

backupdatadir=/opt/mysql\_cluster/BACKUP

也可以其默认的备份目录，系统会默认在 ndbd 的数据节点目录下面创建 BACKUP 目录，并根据指定的备份号在目录下面创建 BACKUP-N 目录。

```
ndb_mgm> START BACKUP
Waiting for completed, this may take several minutes
Node 11: Backup 1 started from node 1
Node 11: Backup 1 started from node 1 completed
StartGCP: 11674 StopGCP: 11677
#Records: 231445 #LogRecords: 0
Data: 5557672 bytes Log: 0 bytes
```

图 执行备份命令

```
[root@mgmd ~]# ssh ndbd01 du -h /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/*
1.4M    /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1-0.11.Data
32K     /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.11.ct1
4.0K    /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.11.log
[root@mgmd ~]# ssh ndbd02 du -h /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/*
1.4M    /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1-0.12.Data
32K     /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.12.ct1
4.0K    /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.12.log
[root@mgmd ~]# ssh ndbd03 du -h /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/*
1.4M    /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1-0.13.Data
32K     /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.13.ct1
4.0K    /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.13.log
[root@mgmd ~]# ssh ndbd04 du -h /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/*
1.4M    /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1-0.14.Data
32K     /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.14.ct1
4.0K    /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.14.log
```

备份同片段

图 查看备份文件

```
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd db01 -e "SHOW TABLES;"
+-----+
| Tables_in_db01 |
+-----+
| t1              |
| t2              |
+-----+
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd -e "SELECT * FROM db01.t1;"
+-----+
| id | name          |
+-----+
| 2  | liuhaicheng   |
| 4  | congcong      |
| 3  | jiangkun      |
| 1  | fengyingcong  |
+-----+
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd -e "SELECT count(*) FROM db01.t2;"
+-----+
| count(*) |
+-----+
| 229376   |
+-----+
```

图 查看当前数据

```
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd -e "DROP DATABASE db01;"
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd -e "SHOW DATABASES;"
+-----+
| Database |
+-----+
| information_schema |
| mysql          |
| ndbinfo        |
| performance_schema |
| test           |
+-----+
```

图 删除数据库 db01

在 Master 存储节点上面执行如下命令，其他节点执行的时候无需加“-m”参数  
**ndbd01 节点恢复：**

```
[root@mgmd ~]# ssh ndbd01 "/opt/mysql_cluster/bin/ndb_restore -c mgmd -n11 -b1
-r -m --backup-path=/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1"
```

```
Nodeid = 11
Backup Id = 1
backup path = /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/
Opening file '/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.11.ctl'
File size 31952 bytes
Backup version in files: ndb-6.3.11 ndb version: mysql-5.5.29 ndb-7.2.10
Stop GCP of Backup: 11676
Connected to ndb!!
Successfully restored table `db01/def/t1`
Successfully restored table event REPL$db01/t1
Successfully restored table `db01/def/t2`
Successfully restored table event REPL$db01/t2
Successfully created index `PRIMARY` on `t1`
Opening file
'/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1-0.11.Data'
File size 1396932 bytes
```

```

Processing data in table: mysql/def/NDB$BLOB_7_3(8) fragment 0
-----
Processing data in table: mysql/def/ndb_index_stat_sample(5) fragment 0
-----
Processing data in table: sys/def/NDB$EVENTS_0(3) fragment 0
-----
Processing data in table: mysql/def/ndb_apply_status(9) fragment 0
-----
Processing data in table: db01/def/t1(10) fragment 0
-----
Processing data in table: mysql/def/ndb_index_stat_head(4) fragment 0
-----
Processing data in table: db01/def/t2(13) fragment 0
-----
Processing data in table: sys/def/SYSTAB_0(2) fragment 0
-----
Processing data in table: mysql/def/ndb_schema(7) fragment 0
Opening file '/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.11.log'
File size 52 bytes
Restored 57104 tuples and 0 log entries

NDBT_ProgramExit: 0 - OK

```

参数说明:

```

-c: --ndb-connectstring=name  后面接管理节点 IP 或者主机名:mgmd
-n: --nodeid=#                后面接存储节点标号
-b: --backupid=#              后面接备份号, START BACKUP ID
-r: --restore-data             指定备份文件所在目录
-m: --restore-meta             恢复元数据信息, 尽在 Master 节点添加【即通过
ndb_mgm -e SHOW 显示的存储节点中的 Master】

```

### ndbd02 节点恢复:

```

[root@mgmd ~]# ssh ndbd02 "/opt/mysql_cluster/bin/ndb_restore -c mgmd -n12 -b1
-r --backup-path=/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1"

```

```

Nodeid = 12
Backup Id = 1
backup path = /opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1
Opening file '/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.12.ctl'
File size 31952 bytes
Backup version in files: ndb-6.3.11 ndb version: mysql-5.5.29 ndb-7.2.10
Stop GCP of Backup: 11676
Connected to ndb!!
Opening
'/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1-0.12.Data'
File size 1401696 bytes

```

file

---

Processing data in table: mysql/def/NDB\$BLOB\_7\_3(8) fragment 1

---

Processing data in table: mysql/def/ndb\_index\_stat\_sample(5) fragment 1

---

Processing data in table: sys/def/NDB\$EVENTS\_0(3) fragment 1

---

Processing data in table: mysql/def/ndb\_apply\_status(9) fragment 1

---

Processing data in table: db01/def/t1(10) fragment 1

---

Processing data in table: mysql/def/ndb\_index\_stat\_head(4) fragment 1

---

Processing data in table: db01/def/t2(13) fragment 2

---

Processing data in table: sys/def/SYSTAB\_0(2) fragment 1

---

Processing data in table: mysql/def/ndb\_schema(7) fragment 1

Opening file '/opt/mysql\_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.12.log'

File size 52 bytes

Restored 57294 tuples and 0 log entries

NDBT\_ProgramExit: 0 - OK

### **ndbd03 节点恢复**

```
[root@mgmd ~]# ssh ndbd03 "/opt/mysql_cluster/bin/ndb_restore -c mgmd -n13 -b1  
-r --backup-path=/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1"
```

Nodeid = 13

Backup Id = 1

backup path = /opt/mysql\_cluster/ndbdata/BACKUP/BACKUP-1

Opening file '/opt/mysql\_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.13.ctl'

File size 31952 bytes

Backup version in files: ndb-6.3.11 ndb version: mysql-5.5.29 ndb-7.2.10

Stop GCP of Backup: 11676

Connected to ndb!!

Opening

file

'/opt/mysql\_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1-0.13.Data'

File size 1387156 bytes

---

Processing data in table: db01/def/t1(10) fragment 2

---

Processing data in table: db01/def/t2(13) fragment 1

Opening file '/opt/mysql\_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.13.log'

File size 52 bytes



Restored 57790 tuples and 0 log entries

NDBT\_ProgramExit: 0 - OK

#### ndbd04 节点恢复

```
[root@mgmd ~]# ssh ndbd04 "/opt/mysql_cluster/bin/ndb_restore -c mgmd -n14 -b1  
-r --backup-path=/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-1"
```

Nodeid = 14

Backup Id = 1

backup path = /opt/mysql\_cluster/ndbdata/BACKUP/BACKUP-1

Opening file '/opt/mysql\_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.14.ctl'

File size 31952 bytes

Backup version in files: ndb-6.3.11 ndb version: mysql-5.5.29 ndb-7.2.10

Stop GCP of Backup: 11676

Connected to ndb!!

Opening

file

'/opt/mysql\_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1-0.14.Data'

File size 1372808 bytes

---

Processing data in table: db01/def/t1(10) fragment 3

---

Processing data in table: db01/def/t2(13) fragment 3

Opening file '/opt/mysql\_cluster/ndbdata/BACKUP/BACKUP-1/BACKUP-1.14.log'

File size 52 bytes

Restored 57192 tuples and 0 log entries

NDBT\_ProgramExit: 0 - OK

在所有节点上面全部执行完成之后，我们通过连接 `mysqld` 节点进行数据的查看，可惜的是，不能够自己创建数据库，所以库我们还得自己创建。

```
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd -e \  
"CREATE DATABASE db01;"
```

```
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd db01 -e "SHOW TABLES;"  
+-----+  
| Tables_in_db01 |  
+-----+  
| t1              |  
| t2              |  
+-----+  
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd -e "SELECT * FROM db01.t1;"  
+-----+  
| id | name      |  
+-----+  
| 3  | jiangkun |  
| 2  | liuhaicheng |  
| 4  | congcong |  
| 1  | fengyingcong |  
+-----+  
[root@mysqld01 ~]# /opt/mysql_cluster/bin/mysql -uroot -ppassw0rd -e "SELECT count(*) FROM db01.t2;"  
+-----+  
| count(*) |  
+-----+  
| 229376   |  
+-----+
```

图 数据成功恢复

### 总结步骤：

1. 拥有备份文件，由于数据备份的时候是分布式存储的，所以任何一个数据节点中的数据都不能够丢失。
2. 保留一个暂时不被使用的 mysqld 节点，删除库中所有表，或者直接删除库，然后重建库。
3. 在每个存储节点执行数据恢复操作。
4. 在 SQL 节点查看数据恢复情况。

注意点：只有在 ndbd 的 Master 节点上面才需要使用 -m 参数，表示恢复元数据信息，如表信息等，在其他 ndbd 节点上面都无需使用该参数，否则会执行失败，再进行数据恢复之前，需要将库中的所有表进行清空或者将表所在的库直接删除，否则也会导致失败。

## 六．集群故障排查

### 【1】由配置文件引起的部分集群工具不能够正常使用

```
[root@mgmd scripts]# ndb_mgm -e SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 4 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)
id=13 @192.168.9.103 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)
id=14 @192.168.9.104 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 3 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)

[root@mgmd scripts]# ndb_desc
Configuration error: Error: Could not alloc node id at localhost port 1186: Connection done from wrong host ip 127.0.0.1.
Unable to connect to management server.

NDBT_ProgramExit: 1 - Failed
```

图 ndb\_desc 执行失败



```

ndb_mgm> SHOW
Cluster Configuration
-----
[ndbd(NDB)] 4 node(s)
id=11 @192.168.9.101 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.9.102 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)
id=13 @192.168.9.103 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)
id=14 @192.168.9.104 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 1)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.9.100 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 4 node(s)
id=21 @192.168.9.201 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.9.202 (mysql-5.5.29 ndb-7.2.10)
id=23 @192.168.9.203 (mysql-5.5.29 ndb-7.2.10)
id=24 (not connected, accepting connect from any host)

ndb_mgm> quit
[root@mgmd scripts]# ndb_de
-bash: ndb_de: command not found
[root@mgmd scripts]# ndb_desc
NDBT_ProgramExit: 0 - OK

```

图 ndb\_desc 执行成功

解决这个问题的方法就是在集群配置文件 config.ini 中加入空的[MYSQLD]标签，如下所示：

#### 1. 修改配置文件

```

.....
[MYSQLD]
NodeId=23
hostname=mysqld03
[MYSQLD]

```

图 加入 mysql 标签

#### 2. 停止整个集群系统

```
[root@mgmd ~]# ndb_mgm -e SHUTDOWN
```

#### 3. 重载配置文件

```
[root@mgmd scripts]# ./mgmd_reload.sh
```

脚本如下

```

#!/bin/bash
ndb_mgmd -f /opt/mysql_cluster/config.ini --configdir=/opt/mysql_cluster --reload

```

#### 4. 启动 ndbd 节点

```
[root@mgmd scripts]# ./ndbd_start.sh
```

脚本如下

```

#!/bin/bash
for i in `cat /etc/hosts |grep ndbd |awk -F' ' '{print $2}'`; do ssh $i
/opt/mysql_cluster/bin/ndbd --defaults-file=/opt/mysql_cluster/my.cnf & done

```

#### 5. 执行测试命令

```
[root@mgmd tmp]# ndb_desc -c mgmd -d db01 -t n t2 -p
```

参数注：

-c 后面接管理节点主机名或者 IP

-d 后面接 database name

-t 后面接 table name

-p 打印分区信息

```
[root@mgmd recovery]# ndb_desc -c mgmd -d db01 -t n t2 -p
-- t2 --
Version: 6
Fragment type: HashMapPartition
K Value: 6
Min load factor: 78
Max load factor: 80
Temporary table: no
Number of attributes: 2
Number of primary keys: 1
Length of frm data: 204
Row Checksum: 1
Row GCI: 1
SingleUserMode: 0
ForceVarPart: 1
FragmentCount: 4
ExtraRowGciBits: 0
ExtraRowAuthorBits: 0
TableStatus: Retrieved
HashMap: DEFAULT-HASHMAP-3840-4
-- Attributes --
id Int NULL AT=FIXED ST=MEMORY
$PK Bigunsigned PRIMARY KEY DISTRIBUTION KEY AT=FIXED ST=MEMORY AUTO_INCR
-- Indexes --
PRIMARY KEY($PK) - UniqueHashIndex
-- Per partition info --
Partition Row count Commit count Frag fixed memory Frag var sized memory Extent_space Free extent_space
0 48351 48351 1966080 0 0 0
1 49023 49023 1998848 0 0 0
2 48527 48527 1966080 0 0 0
3 48523 48523 1966080 0 0 0
NDBT_ProgramExit: 0 - OK
```

图 ndb\_desc 执行结果

【2】关于防火墙问题，之前已经提及，请务必注意

【3】管理节点不能够显示 mysqld 节点信息

原因：ndbd 节点出现问题

解决：检查 ndbd 节点状态

在 ndbd 节点全部启动成功之后，重启 mysqld 节点即可。

## 七. 所有脚本文件

### 【1】管理节点启动

```
#!/bin/bash
#Filename: mgmd_start.sh
ndb_mgmd -f /opt/mysql_cluster/config.ini --configdir=/opt/mysql_cluster
```

### 【2】管理节点重载配置文件

```
#!/bin/bash
#Filename: mgmd_reload.sh
ndb_mgmd -f /opt/mysql_cluster/config.ini --configdir=/opt/mysql_cluster --reload
```

### 【3】存储节点初始化

```
#!/bin/bash
#Filename: ndbd_initial.sh
for i in `cat /etc/hosts |grep ndbd |awk -F' ' '{print $2}'`
do
ssh $i /opt/mysql_cluster/bin/ndbd --defaults-file=/opt/mysql_cluster/my.cnf --initial
&
done
```

### 【4】存储节点正常启动

```
#!/bin/bash
#Filename: ndbd_start.sh
for i in `cat /etc/hosts |grep ndbd |awk -F' ' '{print $2}'`
do
ssh $i /opt/mysql_cluster/bin/ndbd --defaults-file=/opt/mysql_cluster/my.cnf &
done
```

### 【5】存储节点全部重新启动

```
#!/bin/bash
#Filename: ndbd_restart.sh
for i in `cat /etc/hosts |grep ndbd |awk -F' ' '{print $2}'`
do
    ssh $i  pgrep ndb > ndbd_pid
    pidnum=`cat ndbd_pid |wc -l`

    if [ $pidnum -eq 0 ]
    then
        echo "Node-$i is not active!"
    else
        ssh $i "pgrep ndb |xargs skill -9  && /opt/mysql_cluster/bin/ndbd
--defaults-file=/opt/mysql_cluster/my.cnf " &
        fi
    rm -rf ndbd_pid
done
```

### 【6】 存储节点重新初始化

```
#!/bin/bash
#Filename: ndbd_reinitial.sh
for i in `cat /etc/hosts |grep ndbd |awk -F ' ' '{print $2}'`
do
    ssh $i  pgrep ndb > ndbd_pid
    pidnum=`cat ndbd_pid |wc -l`

    if [ $pidnum -eq 0 ]
    then
        echo "Node-$i is not active!"
    else
        ssh $i  "pgrep ndb |xargs skill -9  && /opt/mysql_cluster/bin/ndbd
--defaults-file=/opt/mysql_cluster/my.cnf --initial"  &
        fi
    rm -rf ndbd_pid
done
```

### 【7】 SQL 节点启动关闭重启

```
#!/bin/bash
for i in `cat /etc/hosts |grep  mysqld |awk -F ' ' '{print $2}' ` ;do ssh $i "service
mysqld start " &  done
#!/bin/bash
for i in `cat /etc/hosts |grep  mysqld |awk -F ' ' '{print $2}' ` ;do ssh $i "service
mysqld stop " &  done
#!/bin/bash
for i in `cat /etc/hosts |grep  mysqld |awk -F ' ' '{print $2}' ` ;do ssh $i "service
mysqld restart" &  done
```

### 【8】 关闭集群系统

```
#!/bin/bash
#Filename:cluster_stop.shcd sc
for i in `cat /etc/hosts |grep  mysqld |awk -F ' ' '{print $2}'`
do
    ssh $i "service mysqld stop " &
done
sleep 10
echo "Please Wait 10 Seconds..."
ndb_mgmd -e SHUTDOWN
```

### 【9】 恢复数据文件

```
#!/bin/bash
#Filename: recovery.sh

ymd=`date -d "yesterday" +"%Y-%m-%d"`
hms=`date |awk -F ' ' '{print $4}'`
```

```

logdate=${ymd}_${hms}

ndb_mgm -e SHOW |grep Master |awk -F ' ' '{print $2}' |grep -o '[0-9]*' |grep
`xargs` /etc/hosts |cut -f2 > mastername

cat /etc/hosts |grep ndbd |awk -F ' ' '{print $2}' |grep -v `cat mastername` > nomaster

ndbd_master=`cat mastername`

ndb_no_master=`cat nomaster`

masterId=`ndb_mgm -e SHOW |grep Master |awk -F ' ' '{print $1}' |grep -o
'[0-9]*`
no_masterId=`ndb_mgm -e SHOW |grep Nodegroup |grep -v Master |awk -F ' '
'{print $1}' |grep -o [0-9]*`

backupId=`ssh $ndbd_master ls -d
/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-* |cut -d- -f2 |tee backup_num`

Boo=true
while $Boo
do
    echo -e "You Have `cat backup_num |wc -l` Backups: `echo $backupId`
\nPlease Enter The BackupId You Will To Use[x]:"
    read recoveryid
    for b in $backupId
    do
        if [ $recoveryid == $b ]
        then
            Boo=false
        fi
    done
done
echo "===== Begin
Recovering=====
echo "===== View The Log in logs
=====
echo "+++++++ cd logs; tail -f BACKUP-xx.log
+++++++"
echo "===== View The Log in logs
=====

```

```
ssh $ndbd_master "/opt/mysql_cluster/bin/ndb_restore -c mgmd -n$masterId
-b$recoveryid -r -m
--backup-path=/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-$recoveryid/" >
logs/BACKUP-${recoveryid}-${logdate}.log 2>&1 &

x=0;for i in $ndb_no_master; do ndbd_array[$x]=$i; ((x=x+1));done
y=0;for j in $no_masterId ; do id_array[$y]=$j ; ((y=y+1));done

ndbd_num=${#ndbd_array[@]}

for ((m=0;m<$ndbd_num;m++))
do
    no_ndbd_master=${ndbd_array[$m]}
    no_masterId=${id_array[$m]}
    ssh $no_ndbd_master "/opt/mysql_cluster/bin/ndb_restore -c
mgmd -n$no_masterId -b$recoveryid -r
--backup-path=/opt/mysql_cluster/ndbdata/BACKUP/BACKUP-$recoveryid/" >>
logs/BACKUP-${recoveryid}-${logdate}.log 2>&1 &

done
rm -rf backup_num mastername nomaster
```

## 八. 配置双管理节点

### 8.1 host 配置文件

```
[root@node1 ~]# cat /etc/hosts
127.0.0.1    node1 localhost
192.168.3.121 mysql01 mgmd01 ndbd01
192.168.3.122 mysql02 mgmd02 ndbd02

[root@node1 ~]# ssh ndbd02 cat /etc/hosts
127.0.0.1    node2 localhost
192.168.3.121 mysql01 mgmd01 ndbd01
192.168.3.122 mysql02 mgmd02 ndbd02
```

图 host 配置文件

注意：127.0.0.1 这一行不可以删除，否则管理节点启动失败，无法连接。

## 8.2 管理节点配置文件

```
[root@node1 ~]# cat /opt/mysql_cluster/config.ini
[NDBD DEFAULT]
NoOfReplicas=2

[NDB_MGMD]
NodeId=1
hostname=mgmd01
datadir=/opt/mysql_cluster/logs
[NDB_MGMD]
NodeId=2
hostname=mgmd02
datadir=/opt/mysql_cluster/logs

[NDBD]
NodeId=11
hostname=ndbd01
datadir=/opt/mysql_cluster/ndbdata
[NDBD]
NodeId=12
hostname=ndbd02
datadir=/opt/mysql_cluster/ndbdata

[MYSQLD]
NodeId=21
hostname=mysqld01
[MYSQLD]
NodeId=22
hostname=mysqld02
```

## 8.3 关于 ndbd 以及 mysqld 配置文件

```
[root@node1 ~]# cat /etc/my.cnf
[mysql_cluster]
ndb-connectstring=mgmd01,mgmd02
[mysqld_multi]
mysqld      = /opt/mysql_cluster/bin/mysqld_safe
mysqladmin  = /opt/mysql_cluster/bin/mysqladmin

[mysqld1]
ndbcluster
ndb-connectstring=mgmd01,mgmd02
```

```

default-storage-engine=ndb
socket      = /data/AuditDB/auditdb.sock
port        = 3306
pid-file    = /data/AuditDB/auditdb.pid
basedir=/opt/mysql_cluster/
datadir     = /data/AuditDB/
user        = mysql
lower_case_table_names=1

[mysqld2]
ndbcluster
ndb-connectstring=mgmd01,mgmd02
default-storage-engine=ndb
socket      = /data/OscDB/oscdb.sock
port        = 3307
pid-file    = /data/OscDB/oscdb.pid
basedir= /opt/mysql_cluster/
datadir     = /data/OscDB/
user        = mysql
lower_case_table_names=1

```

## 8.4 配置多实例

配置文件如上所示

在启动 mysqld 进程之前，先初始化 mysqld 数据。

```
[root@node1 ~]# mkdir -p /data/AuditDB/ ; mkdir -p /data/OscDB/
```

```
[root@node1 ~]# chown mysql.mysql -R /data/
```

```

[root@node1 ~]# /opt/mysql_cluster/scripts/mysql_install_db --user=mysql --defaults-file=/etc/my.cnf \
> --basedir=/opt/mysql_cluster/ --datadir=/data/AuditDB/
[root@node1 ~]# /opt/mysql_cluster/scripts/mysql_install_db --user=mysql --defaults-file=/etc/my.cnf \
> --basedir=/opt/mysql_cluster/ --datadir=/data/OscDB/

```

图 初始化 mysql

```
[root@node1 ~]# cp /opt/mysql_cluster/bin/mysqld_multi /etc/init.d/mysqld_multi
```

```
[root@node1 ~]# /etc/init.d/mysqld_multi start //启动多实例
```

```

[root@node1 ~]# netstat -nltup |grep 330
tcp        0      0 0.0.0.0:3306          0.0.0.0:*            LISTEN     21423/mysqld
tcp        0      0 0.0.0.0:3307          0.0.0.0:*            LISTEN     21422/mysqld

```

两边都进行启动之后，查看集群状态：



```

[root@node1 ~]# /opt/mysql_cluster/bin/ndb_mgm -e SHOW
Connected to Management Server at: mgmd01:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 2 node(s)
id=1 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10)
id=2 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10)

```

图 双管理节点集群状态

## 8.5 停止一个管理节点

```

[root@node1 ~]# /opt/mysql_cluster/bin/ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: mgmd01:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 2 node(s)
id=1 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10)
id=2 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10)

ndb_mgm> 1 STOP
Node 1 has shutdown.
Disconnecting to allow Management Server to shutdown

```

图 停止 node1 的管理节点

```

[root@node2 ~]# /opt/mysql_cluster/bin/ndb_mgm -e SHOW
Connected to Management Server at: mgmd02:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 2 node(s)
id=1 (not connected, accepting connect from mgmd01)
id=2 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10)

```

图 通过 node2 查看管理节点状态

```
[root@node1 ~]# /opt/mysql_cluster/bin/ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: mgmd02:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=11 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0, Master)
id=12 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 2 node(s)
id=1 (not connected, accepting connect from mgmd01)
id=2 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10)

[mysqld(API)] 2 node(s)
id=21 @192.168.3.121 (mysql-5.5.29 ndb-7.2.10)
id=22 @192.168.3.122 (mysql-5.5.29 ndb-7.2.10)
```

图 通过 node1 登录查看集群状态