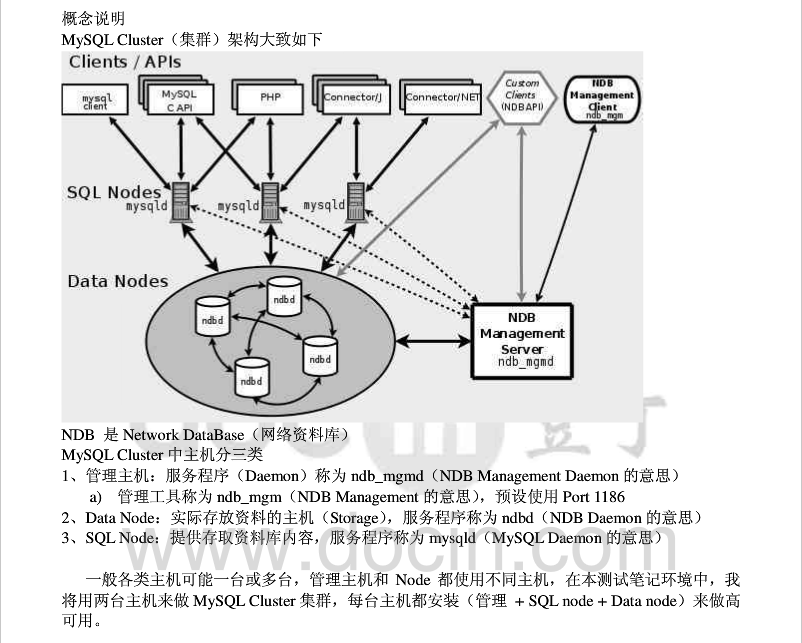
# MySQL cluster 7.2集群部署配置

1. 介绍  
   这篇文档旨在介绍如何安装配置基于2台服务器的MySQL集群。并且实现任意一台服务器出现问题或宕机时MySql集群依然能够继续运行。

  
# uname -a

# cat /etc/redhat-release

**安装环境及软件包**  
2台服务器  
软件包：mysql-cluster-gpl-7.3.4-linux-glibc2.5-x86\_64.tar.gz  
操作系统：Centos 7.2  
SerA: eth0 192.168.160.135 （管理 + SQL node + Data node）  
SerB: eth0 192.168.160.136 （管理 + SQL node + Data node）

操作系统安装完成后，请关闭防火墙

# systemctl stop firewalld.service #停止firewall

# systemctl disable firewalld.service #禁止firewall开机启动

# firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running）  
-----------------------------------------------------------------------------------------------------

二、在SerA和SerB上安装MySQL  
以下步骤需要在SerA和SerB上各做一次  
# mv mysql-cluster-gpl-7.3.4-linux-glibc2.5-x86\_64.tar.gz /usr/local/

# cd /usr/local/

添加mysql用户组及用户，修改相关目录权限

# groupadd mysql

# useradd -g mysql mysql

# tar -zxvf mysql-max-4.1.9-pc-linux-gnu-i686.tar.gz

# rm -f mysql-max-4.1.9-pc-linux-gnu-i686.tar.gz

# mv mysql-max-4.1.9-pc-linux-gnu-i686 mysql

安装MySQL cluster

# cd mysql

# scripts/mysql\_install\_db --user=mysql

# chown -R root .

# chown -R mysql data

# chgrp -R mysql .

# cp support-files/mysql.server /etc/rc.d/init.d/mysqld

# chmod +x /etc/rc.d/init.d/mysqld  
-------------------------------------------------------------------------------------------------  
三、安装并配置节点  
以下步骤需要在SerA和SerB上各做一次  
1.配置管理节点配置文件：  
# mkdir /var/lib/mysql-cluster  
# vim /var/lib/mysql-cluster/config.ini   
在config.ini中添加如下内容：  
[ndbd default]  
NoOfReplicas= 2  
MaxNoOfConcurrentOperations= 10000  
# Amount of memory required=(SizeofDatabase \* NumberOfReplicas \* 1.1 ) / NumberOfDataNodes  
DataMemory= 200M  
IndexMemory=300M  
TimeBetweenWatchDogCheck= 30000  
DataDir=/var/lib/mysql-cluster  
MaxNoOfOrderedIndexes= 512  
StartPartialTimeout=100  
StartPartitionedTimeout=100  
ArbitrationTimeout=5000  
TransactionDeadlockDetectionTimeout=5000  
HeartbeatIntervalDbDb=5000  
StopOnError=0  
  
[ndb\_mgmd default]  
DataDir=/var/lib/mysql-cluster  
[ndb\_mgmd]  
Id=1  
HostName= 192.168.160.135  
[ndb\_mgmd]  
Id=2  
HostName= 192.168.160.136  
[ndbd]  
Id= 3  
HostName= 192.168.160.135  
[ndbd]  
Id= 4  
HostName= 192.168.160.136  
[mysqld]  
ArbitrationRank=2 (非常重要,全靠有它,才可以形成仲裁竞争,从而当另一个机子当了时,此机还可以有知道partion完整的节点)  
[mysqld]  
ArbitrationRank=2  
[mysqld] (多出的这项是留给恢复时使用的.)  
[mysqld] (多出的这项是留给恢复时使用的.)  
[tcp default]  
PortNumber= 63132

[separator]

----------------------------------------------------------------------------------------------------  
2.配置通用my.cnf文件，mysqld及ndbd,ndb\_mgmd均使用此文件.  
# vi /etc/my.cnf  
在my.cnf中添加如下内容：  
[mysqld]  
datadir=/usr/local/var  
socket=/usr/local/var/mysql.sock  
# Default to using old password format for compatibility with mysql 3.x  
# clients (those using the mysqlclient10 compatibility package).  
old\_passwords=1  
default-storage-engine=ndbcluster  
ndbcluster  
ndb-connectstring=192.168.1.50,192.168.1.8  
  
[ndbd]  
connect-string=192.168.1.50,192.168.1.8  
  
[ndb\_mgm]  
connect-string=192.168.1.50,192.168.1.8  
  
[ndb\_mgmd]  
config-file=/var/lib/mysql-cluster/config.ini  
  
[mysql\_cluster]  
ndb-connectstring=192.168.1.50,192.168.1.8  
  
[mysql.server]  
user=mysql  
basedir=/usr/local/  
  
[mysqld\_safe]  
log-error=/var/log/mysqld.log  
#pid-file=/var/run/mysqld/mysqld.pid  
[mysql]  
#socket=/usr/local/var/mysql.sock  
[mysqladmin]  
#socket=/usr/local/var/mysql.sock  
[ndb\_restore default]  
  
  
保存退出后.  
  
  
  
-----------------------------------------------------------------------------------------------------

1. 启动管理节点SerA为：  
   [root@SerA ~]# ndb\_mgmd --ndb\_nodeid=1  
   Cluster configuration warning:  
   arbitrator with id 1 and db node with id 3 on same host 192.168.1.50  
   arbitrator with id 2 and db node with id 4 on same host 192.168.1.8  
   arbitrator with id 5 has no hostname specified  
   arbitrator with id 6 has no hostname specified  
   Running arbitrator on the same host as a database node may  
   cause complete cluster shutdown in case of host failure.  
   注:在启动时有一个警告提示  
   说节点1和3，2和4的arbitrator一样，可能引起整个集群失败。（可以不用放在心上）  
     
   启动管理节点SerB为：  
   [root@SerB ~]# ndb\_mgmd --ndb\_nodeid=2

<方法二：>

/usr/local/mysql/bin/ndb\_mgmd -f /var/lib/mysql-cluster/config.ini

其他操作：

# 查看是否有端口号为1186的监听端口 netstat -lntpu

# 查看集群状态

/usr/local/mysql/bin/ndb\_mgm -e show

# 管理节点检验  /usr/local/mysql/bin/ndb\_mgm

# 管理节点关闭  /usr/local/mysql/bin/ndb\_mgm -e shutdown  
-----------------------------------------------------------------------------------------------------  
  
五.初始化集群  
在SerA中  
[root@SerA ~]# /usr/local/mysql/bin/ndbd --initial  
在SerB中  
[root@SerB ~]# /usr/local/mysql/bin/ndbd --initial  
注：只有在第一次启动ndbd时或者对config.ini进行改动后才需要使用--initial参数！(在下面为了进行恢复实验时还再次使用到)

# 正常启动方式  /usr/local/mysql/bin/ndbd  
  
-----------------------------------------------------------------------------------------------------

检查工作状态  
在任意一台机子上启动管理终端：  
[root@SerA ~]# ndb\_mgm -e show  
Connected to Management Server at: 192.168.1.50:1186  
Cluster Configuration  
---------------------  
[ndbd(NDB)]     2 node(s)  
id=3    @192.168.1.50 (Version: 6.0.0, Nodegroup: 0, Master)  
id=4    @192.168.1.8 (Version: 6.0.0, Nodegroup: 0)  
  
[ndb\_mgmd(MGM)] 2 node(s)  
id=1    @192.168.1.50 (Version: 6.0.0)  
id=2    @192.168.1.8 (Version: 6.0.0)  
  
[mysqld(API)]   3 node(s)  
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)  
  
如果上面没有问题，现在开始加入mysqld(API)

-----------------------------------------------------------------------------------------------------

# 检验mysql是否运行  /etc/rc.d/init.d/mysqld status

# sql节点关闭service mysqld start

# sql节点关闭 service mysql stop

# 为sql指定密码

/usr/local/mysql/bin/mysqladmin -u root password 'new-password'

/usr/local/mysql/bin/mysqladmin -u root -h 'host' password 'new-password'

# 启动命令行窗口  /usr/local/mysql/bin/mysql -uroot -p

六.SQL节点启动  
注意，这篇文档对于MySQL并没有设置root密码，推荐你自己设置SerA和SerB的MySQL root密码。  
在SerA 中：  
[root@SerA ~]# service mysqld start  
在SerB 中：  
[root@SerB ~]# service mysqld start  
  
再次检查工作状态,看mysql节点是否加入成功  
[root@SerA ~]# ndb\_mgm -e show  
Connected to Management Server at: 192.168.1.50:1186  
Cluster Configuration  
---------------------  
[ndbd(NDB)]     2 node(s)  
id=3    @192.168.1.50 (Version: 6.0.0, Nodegroup: 0, Master)  
id=4    @192.168.1.8 (Version: 6.0.0, Nodegroup: 0)  
  
[ndb\_mgmd(MGM)] 2 node(s)  
id=1    @192.168.1.50 (Version: 6.0.0)  
id=2    @192.168.1.8 (Version: 6.0.0)  
  
[mysqld(API)]   3 node(s)  
id=5    @192.168.1.50 (Version: 6.0.0)  
id=6    @192.168.1.8 (Version: 6.0.0)  
id=7 (not connected, accepting connect from any host)  
-----------------------------------------------------------------------------------------------------  
七.测试：  
在SerA 中  
[root@SerA ~]# /usr/local/mysql/bin/mysql -uroot -p  
>create databases backup;  
>use backup;  
>create table dog (name varchar(10));  
>create table pig (name varchar(10));  
  
退出终端, 使用下面的命令往上面两个表内批量插入数据.  
[root@SerA ~]# mysql -uroot<sql.txt  
[root@SerB ~]# mysql -uroot<sqltest.txt  
  
这里要等上几分钟, 应为sql.txt 里有20W行记录, 而且sqltest.txt也用10W行.  
  
数据插入完毕再回到终端中检查是否有新增的数据库和表以及数据.  
[root@SerB ~]# mysql -uroot  
>show databases;  
+--------------------+  
| Database           |  
+--------------------+  
| information\_schema |  
| backup             |  
| mysql              |  
| test               |  
+--------------------+  
4 rows in set (0.00 sec)  
  
mysql> use backup  
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> show tables;  
+------------------+  
| Tables\_in\_backup |  
+------------------+  
| dog              |  
| pig              |  
+------------------+  
2 rows in set (0.00 sec)  
  
  
mysql> select \* from pig;  
...............  
...............  
  
| 144163 |  
| 173821 |  
| 188584 |  
| 45860 |  
+--------+  
200000 rows in set (1.66 sec)  
  
  
mysql> select \* from pig;  
...............  
...............  
  
| 27580 |  
| 83268 |  
| 47744 |  
| 97018 |  
+--------+  
100000 rows in set (0.83 sec)  
  
  
可以看到mysql能正常工作.  
  
  
ndb下数据备份和恢复：  
备份很简单：  
在任意的一台机子上,只需通过ndb\_mgm，运行start backup  
[root@SerB zman]# ndb\_mgm  
-- NDB Cluster -- Management Client --  
ndb\_mgm> start backup  
Connected to Management Server at: 192.168.1.50:1186  
Waiting for completed, this may take several minutes  
Node 3: Backup 1 started from node 1  
Node 3: Backup 1 started from node 1 completed  
StartGCP: 515 StopGCP: 518  
#Records: 302059 #LogRecords: 0  
Data: 8427304 bytes Log: 0 bytes  
  
这个备份很快，备份的结果是在每个数据节点上都生成一个备份. 用ll命令分别在两台机子上查看是否生成备份文件.  
  
[root@SerA mysql-cluster]# ll /var/lib/mysql-cluster/BACKUP/BACKUP-1/  
total 4116  
-rw-r--r-- 1 root root 4194172 Aug 22 02:16 BACKUP-1-0.3.Data  
-rw-r--r-- 1 root root    8580 Aug 22 02:16 BACKUP-1.3.ctl  
-rw-r--r-- 1 root root      44 Aug 22 02:16 BACKUP-1.3.log  
  
[root@SerB conf]# ll /var/lib/mysql-cluster/BACKUP/BACKUP-1/  
总计 4156  
-rw-r--r-- 1 root root 4233956 08-22 14:15 BACKUP-1-0.4.Data  
-rw-r--r-- 1 root root    8580 08-22 14:15 BACKUP-1.4.ctl  
-rw-r--r-- 1 root root      44 08-22 14:15 BACKUP-1.4.log  
  
  
恢复  
恢复要通过以下几个步骤完成:  
1. 测试删除数据表，至少1个节点重新建个空的；  
2. 停止sql节点的运行，或者在配置文件中增加1个空的sql节点标志(config.ini文件里多出的一个[mysqld]项就是留个这里用 的)，否则会出现 No free node id found for mysqld(API) 错误；  
3. 首先在任意一个节点上恢复表结构，然后在每个数据节点上恢复数据  
ndb\_restore -n 4 -b 1 -m -r /var/lib/mysql-cluster/BACKUP/BACKUP-1/  
没有-m 这个恢复过程，会出现Unable to find table错误  
-n 和 -b 的数值，对应备份文件 BACKUP-n.b.ctl，这里是 BACKUP-1.4.ctl  
ndb\_restore -n 3 -b 1 -r /var/lib/mysql-cluster/BACKUP/BACKUP-1/  
  
  
  
1) 先关闭集群.  
[root@SerA ~]# mysqladmin shutdown  
[root@SerB ~]# mysqladmin shutdown  
  
[root@SerA ~]# ndb\_mgm -e shutdown  
Connected to Management Server at: 192.168.1.50:1186  
2 NDB Cluster node(s) have shutdown.  
Disconnecting to allow management server to shutdown.  
  
2)重复上面的四五六 (主要是想通过 ndbd --ndb\_nodeid=3 --initial 和 ndbd --ndb\_nodeid=4 --initial 將数据库初始化)  
  
[root@SerA ~]# mysql -uroot  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 2  
Server version: 6.0.0-alpha Source distribution  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql> use backup  
Database changed  
mysql> show tables; (可以看到数据库是空的)  
Empty set (0.00 sec)  
  
  
3)进行恢复  
在SerB上执行:  
[root@SerB ~]# ndb\_restore -n 4 -b 1 -m -r /var/lib/mysql-cluster/BACKUP/BACKUP-1  
Nodeid = 4  
Backup Id = 1  
backup path = /var/lib/mysql-cluster/BACKUP/BACKUP-1  
Ndb version in backup files: Version 6.0.0  
Connected to ndb!!  
Successfully restored table backup/def/pig  
Successfully restored table event REPL$backup/pig  
Successfully restored table backup/def/dog  
Successfully restored table event REPL$backup/dog  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: backup/def/pig(6) fragment 1  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: sys/def/NDB$EVENTS\_0(1) fragment 1  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: mysql/def/NDB$BLOB\_2\_3(3) fragment 1  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: sys/def/SYSTAB\_0(0) fragment 1  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: mysql/def/ndb\_schema(2) fragment 1  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: mysql/def/ndb\_apply\_status(4) fragment 1  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: backup/def/dog(5) fragment 1  
Restored 150727 tuples and 0 log entries  
  
NDBT\_ProgramExit: 0 - OK  
  
  
  
  
在SerA上执行:  
[root@SerA ~]# ndb\_restore -n 3 -b 1 -r /var/lib/mysql-cluster/BACKUP/BACKUP-1  
Nodeid = 3  
Backup Id = 1  
backup path = /var/lib/mysql-cluster/BACKUP/BACKUP-1  
Ndb version in backup files: Version 6.0.0  
Connected to ndb!!  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: backup/def/pig(6) fragment 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: sys/def/NDB$EVENTS\_0(1) fragment 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: mysql/def/NDB$BLOB\_2\_3(3) fragment 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: sys/def/SYSTAB\_0(0) fragment 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: mysql/def/ndb\_schema(2) fragment 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: mysql/def/ndb\_apply\_status(4) fragment 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Processing data in table: backup/def/dog(5) fragment 0  
Restored 149273 tuples and 0 log entries  
  
NDBT\_ProgramExit: 0 - OK  
  
  
  
  
回到终端里查看恢复情况  
[root@SerA ~]# mysql -uroot  
mysql> use backup  
mysql> show tables;  
+------------------+  
| Tables\_in\_backup |  
+------------------+  
| dog              |  
| pig              |  
+------------------+  
2 rows in set (0.00 sec)  
  
mysql> select \* from pig;  
...............  
...............  
  
| 144163 |  
| 173821 |  
| 188584 |  
| 45860 |  
+--------+  
200000 rows in set (1.66 sec)  
  
  
mysql> select \* from pig;  
...............  
...............  
  
| 27580 |  
| 83268 |  
| 47744 |  
| 97018 |  
+--------+  
100000 rows in set (0.83 sec)  
  
OK, 数据库恢复成功.  
cluster目前，只能对所有数据进行备份和恢复，不能选择数据库，也不能进行差量备份，不知如果对应上G的数据会怎样，目前想到的是打开log-bin手工进行差量数据恢复

转载: [http://hi.baidu.com/doublelook/blog/item/e7ed781e3714246af624e4b0.html](http://hi.baidu.com/doublelook/blog/item/e7ed781e3714246af624e4b0.html" \t "http://www.cnblogs.com/hustcat/articles/_blank)

在配置中还是遇到不少问题:

最终成功版本:

下载:[http://mysql.cs.pu.edu.tw/Downloads/MySQL-5.0/mysql-5.0.85-linux-i686.tar.gz](http://mysql.cs.pu.edu.tw/Downloads/MySQL-5.0/mysql-5.0.85-linux-i686.tar.gz" \t "http://www.cnblogs.com/hustcat/articles/_blank)

# mv mysql-5.0.85-linux-i686.tar.gz /usr/local/

# cd /usr/local/

# groupadd mysql

# useradd -g mysql mysql

# tar -zxvf mysql-5.0.85-linux-i686.tar.gz

# rm -f mysql-5.0.85-linux-i686.tar.gz

# mv mysql-5.0.85-linux-i686 mysql

# cd mysql

# scripts/mysql\_install\_db --user=mysql

# chown -R root   .

# chown -R mysql data

# chgrp -R mysql .

# cp support-files/mysql.server /etc/rc.d/init.d/mysqld

# chmod +x /etc/rc.d/init.d/mysqld

# chkconfig --add mysqld

my.cnf:

[mysqld]  
datadir=/usr/local/mysql/data  
socket=/tmp/mysql.sock  
default-storage-engine=ndbcluster  
ndbcluster  
ndb-connectstring=192.168.1.229,192.168.1.191  
[ndbd]  
connect-string=192.168.1.229,192.168.1.191  
[ndb\_mgm]  
connect-string=192.168.1.229,192.168.1.191  
[ndb\_mgmd]  
config-file=/var/lib/mysql-cluster/config.ini  
[mysql\_cluster]  
ndb-connectstring= 192.168.1.229,192.168.1.191  
[mysql\_server]  
user=mysql  
basedir=/usr/local  
[mysqld\_safe]  
log-error=/var/log/mysqld.log  
[mysql]  
[mysqladmin]

config.ini:

[ndbd default]  
NoOfReplicas= 2  
MaxNoOfConcurrentOperations= 10000  
DataMemory= 128M  
IndexMemory= 24M  
TimeBetweenWatchDogCheck= 30000  
DataDir= /var/lib/mysql-cluster  
MaxNoOfOrderedIndexes= 512  
StartPartialTimeout=100  
StartPartitionedTimeout=100  
ArbitrationTimeout=5000  
TransactionDeadlockDetectionTimeout=5000  
HeartbeatIntervalDbDb=5000  
StopOnError=0

[ndb\_mgmd default]  
DataDir= /var/lib/mysql-cluster  
[ndb\_mgmd]  
Id=1  
HostName= 192.168.1.229  
[ndb\_mgmd]  
Id=2  
HostName= 192.168.1.191  
[ndbd]  
Id=3  
HostName= 192.168.1.229  
[ndbd]  
Id=4  
HostName= 192.168.1.191  
[mysqld]  
ArbitrationRank=2  
[mysqld]  
ArbitrationRank=2  
[mysqld]  
[mysqld]