NSD DATABASE DAY07

1 部署集群基础环境

1.1 问题

本案例要求为MySQL集群准备基础环境，完成以下任务操作：

- 数据库授权
- 部署MySQL双主多从结构
- 配置本机hosts解析记录

1.2 方案

使用4台RHEL 6虚拟机，如图-1所示。其中192.168.4.10、192.168.4.11作为MySQL双主服务器，192.168.4.12、192.168.4.13作为主服务器的从服务器。

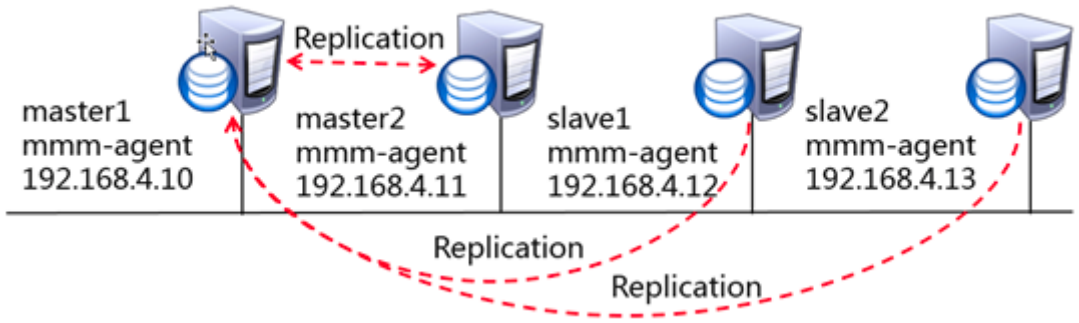

图 - 1

Top

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：准备环境

```
[root@master1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.4.10    master1         master1.tarena.com
192.168.4.11    master2         master2.tarena.com
192.168.4.12    slave1          slave1.tarena.com
192.168.4.13    slave2          slave2.tarena.com
192.168.4.100   master1         master1.tarena.com

[root@master1 ~]# ping -c 2 master1
PING master1 (192.168.4.10) 56(84) bytes of data.
64 bytes from master1 (192.168.4.10): icmp_seq=1 ttl=64 time=0.378 ms
64 bytes from master1 (192.168.4.10): icmp_seq=2 ttl=64 time=0.396 ms

--- master1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.378/0.387/0.396/0.009 ms
[root@master1 ~]#
```

步骤二：部署数据库主机

1）安装启动数据库（4台数据库主机master1，master2，slave1，slave2执行以下操作）

Top

```
[root@master1 ~]# tar xvf MySQL-5.6.15-1.el6.x86_64.rpm-bundle.tar    //解压软件包
```

```
02.  .. ..
03.  [root@master1 ~]# rpm -Uvh MySQL-*.rpm                    //安装MySQL
04.  .. ..
05.  [root@master1 ~]# service mysql start
06.  Starting MySQL.                                  [确定]
```

2）初始化配置数据库（4台数据库主机master1，master2，slave1，slave2执行以下操作）

```
01.  [root@master1 ~]# cat /root/.mysql_secret        //查看随机生成密码
02.  # The random password set for the root user at Thu May  7 22:15:47 2015 (local time): wW1BNAjD
03.
04.  [root@master1 ~]# mysql -uroot -pwW1BNAjD        //使用随机生成密码登陆
05.  Warning: Using a password on the command line interface can be insecure.
06.  Welcome to the MySQL monitor.  Commands end with ; or \g.
07.  Your MySQL connection id is 1
08.  Server version: 5.6.15
09.
10.  Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
11.
12.  Oracle is a registered trademark of Oracle Corporation and/or its
13.  affiliates. Other names may be trademarks of their respective
14.  owners.
15.
16.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
17.                                                                                    Top
18.  mysql> set password=password("pwd123");          //修改数据库root密码
19.  Query OK, 0 rows affected (0.49 sec)
```

```
20.
21.     mysql> exit
22.     Bye
23.     [root@master1 ~]#
```

步骤三：部署双主多从结构

1）数据库授权（4台数据库主机master1，master2，slave1，slave2执行以下操作）

部署主从同步只需要授权一个主从同步用户即可，但是我们要部署MySQL-MMM架构，所以在这里我们将MySQL-MMM所需用户一并进行授权设置。再授权一个测试用户，在架构搭建完成时测试使用。

```
01.     [root@master1 ~]# mysql -uroot -ppwd123
02.     Warning: Using a password on the command line interface can be insecure.
03.     Welcome to the MySQL monitor.  Commands end with ; or \g.
04.     Your MySQL connection id is 2
05.     Server version: 5.6.15 MySQL Community Server (GPL)
06.
07.     Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
08.
09.     Oracle is a registered trademark of Oracle Corporation and/or its
10.     affiliates. Other names may be trademarks of their respective
11.     owners.
12.
13.     Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Top

数据库授权部分为了方便试验我们直接允许所有地址访问了，真实环境需谨慎

```
mysql> grant  replication slave on *.* to slaveuser@"%" identified by  "pwd123";                    //主从同步授权
Query OK, 0 rows affected (0.00 sec)

mysql> grant  replication client on *.* to monitor@"%" identified by "monitor";                     //MMM所需架构用户授权
Query OK, 0 rows affected (0.06 sec)

mysql> grant  replication client,process,super  on *.* to agent@"%" identified by "agent";                 //MMM所需架构用户授权
Query OK, 0 rows affected (0.00 sec)

mysql> grant  all on *.* to root@"%" identified by "pwd123";  //测试用户授权
Query OK, 0 rows affected (0.00 sec)

mysql>
```

2）开启主数据库binlog日志、设置server_id（master1，master2）

master1设置：

```
[root@master1 ~]# cat /etc/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
server_id=10                    //设置server_id,该值集群中不可以重复
log-bin                         //开启bin-log日志
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

```

Top

```
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

[root@master1 ~]# service mysql restart                //重启MySQL服务
Shutting down MySQL..                                  [确定]
Starting MySQL..                                       [确定]
[root@master1 ~]# ls /var/lib/mysql/master1-bin*       //查看binlog日志是否生成
/var/lib/mysql/master1-bin.000001  /var/lib/mysql/master1-bin.index
[root@master1 ~]#
```

master2设置：

```
[root@master2 ~]# cat /etc/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
server_id=11
log-bin
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
[root@master2 ~]# service mysql restart
```

Top

```
15.  Shutting down MySQL..                                 [确定]
16.  Starting MySQL.                                       [确定]
17.  [root@master2 ~]# ls /var/lib/mysql/master2-bin.*
18.  /var/lib/mysql/master2-bin.000001  /var/lib/mysql/master2-bin.index
```

3）从库设置server_id

slave1设置：

```
01.  [root@slave1 ~]# cat /etc/my.cnf
02.  [mysqld]
03.  datadir=/var/lib/mysql
04.  socket=/var/lib/mysql/mysql.sock
05.  user=mysql
06.  server_id=12
07.  # Disabling symbolic-links is recommended to prevent assorted security risks
08.  symbolic-links=0
09.
10.  [mysqld_safe]
11.  log-error=/var/log/mysqld.log
12.  pid-file=/var/run/mysqld/mysqld.pid
13.  [root@slave1 ~]# service mysql restart
14.  Shutting down MySQL..                                 [确定]
15.  Starting MySQL..                                      [确定]
16.  [root@slave1 ~]#
```

Top

slave2设置：

```
[root@slave2 ~]# cat /etc/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
server_id=13
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
[root@slave2 ~]# service mysql restart
Shutting down MySQL..                                      [确定]
Starting MySQL.                                            [确定]
[root@slave2 ~]#
```

4）配置主从从从关系

配置master2、slave1、slave2成为master1的从服务器

查看master1服务器binlong日志使用节点信息：

```
[root@master1 ~]# mysql -uroot -ppwd123
...
mysql> show master status\G
*************************** 1. row ***************************
            File: master1-bin.000001
```

Top

```
            Position: 120
        Binlog_Do_DB:
    Binlog_Ignore_DB:
Executed_Gtid_Set:
1 row in set (0.00 sec)

mysql>
```

设置master2为master1从：

```
[root@master2 ~]# mysql -uroot -ppwd123
...
mysql> change master to                    //设置主服务器信息
    -> master_host="192.168.4.10",         //设置主服务器IP地址
    -> master_user="slaveuser",            //设置主从同步用户
    -> master_password="pwd123",           //设置主从同步密码
    -> master_log_file="master1-bin.000001",   //设置主库binlog日志名称
    -> master_log_pos=120;                 //设置主从binlog日志使用节点
Query OK, 0 rows affected, 2 warnings (0.06 sec)

mysql> start slave;                        //启动同步进程
Query OK, 0 rows affected (0.00 sec)

mysql> show slave status\G                 //查看主从是否成功
...
```

Top

启动同步进程后查看IO节点和SQL节点是否为Yes如果均为Yes表示主从正常。

```
                Slave_IO_Running: Yes            //IO节点正常
               Slave_SQL_Running: Yes             //SQL节点正常
.. ..
mysql>
```

设置slave1为master1从：

```
[root@slave1 ~]# mysql -uroot -ppwd123
.. ..
mysql> change master to
    -> master_host="192.168.4.10",
    -> master_user="slaveuser",
    -> master_password="pwd123",
    -> master_log_file="master1-bin.000001",
    -> master_log_pos=120;
Query OK, 0 rows affected, 2 warnings (0.12 sec)

mysql> start slave;
Query OK, 0 rows affected (0.16 sec)

mysql> show slave status\G
.. ..
                Slave_IO_Running: Yes            //IO节点正常
               Slave_SQL_Running: Yes             //SQL节点正常
```

Top

```
18.  .. ..
19.  mysql>
```

设置slave2为master1从：

```
01.  [root@slave2 ~]# mysql -uroot -ppwd123
02.  .. ..
03.  mysql> change master to
04.      -> master_host="192.168.4.10",
05.      -> master_user="slaveuser",
06.      -> master_password="pwd123",
07.      -> master_log_file="master1-bin.000001",
08.      -> master_log_pos=120;
09.  Query OK, 0 rows affected, 2 warnings (0.13 sec)
10.
11.  mysql> start slave;
12.  Query OK, 0 rows affected (0.27 sec)
13.
14.  mysql> show slave status\G
15.  .. ..
16.               Slave_IO_Running: Yes            //IO节点正常
17.              Slave_SQL_Running: Yes             //SQL节点正常
18.  .. ..
19.  mysql>
```

Top

5）配置主主从从关系，将master1配置为master2的从

查看master2的binlog使用信息：

```
[root@master2 ~]# mysql -uroot -ppwd123
... ...
mysql> show master status\G
*************************** 1. row ***************************
             File: master2-bin.000001
         Position: 120
     Binlog_Do_DB:
 Binlog_Ignore_DB:
Executed_Gtid_Set:
1 row in set (0.00 sec)

mysql>
```

设置master1成为master2的从：

```
[root@master1 ~]# mysql -uroot -ppwd123
... ...
mysql> change master to
    -> master_host="192.168.4.11",
    -> master_user="slaveuser",
    -> master_password="pwd123",
    -> master_log_file="master2-bin.000001",
    -> master_log_pos=120;
Query OK, 0 rows affected, 2 warnings (0.31 sec)
```

Top

```
mysql> start slave;
Query OK, 0 rows affected (0.27 sec)

mysql> show slave status\G
... ...
             Slave_IO_Running: Yes            //IO节点正常
            Slave_SQL_Running: Yes            //SQL节点正常
... ...
mysql>
```

6）测试主从架构是否成功

master1更新数据，查看其它主机是否同步：

```
[root@master1 ~]# mysql -uroot -ppwd123
... ...
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.00 sec)

```

Top

```
mysql> create database tarena;
Query OK, 1 row affected (0.06 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| tarena             |
| test               |
+--------------------+
5 rows in set (0.00 sec)

mysql>
```

master2主机查看：

```
[root@master2 ~]# mysql -uroot -ppwd123 -e "show databases"
Warning: Using a password on the command line interface can be insecure.
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
```

Top

```
| tarena             |
| test               |
+--------------------+
[root@master2 ~]#
```

slave1主机查看：

```
[root@slave1 ~]# mysql -uroot -ppwd123 -e "show databases"
Warning: Using a password on the command line interface can be insecure.
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| tarena             |
| test               |
+--------------------+
[root@slave1 ~]#
```

slave2主机查看：

```
[root@slave2 ~]# mysql -uroot -ppwd123 -e "show databases"
Warning: Using a password on the command line interface can be insecure.
+--------------------+
| Database           |
```

Top

```
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| tarena             |
| test               |
+--------------------+
[root@slave2 ~]#
```

2 MySQL-MMM架构部署

2.1 问题

本案例要求熟悉实现MySQL-MMM的架构部署，主要包括以下任务：

1. 安装依赖包
2. 安装软件包
3. 配置MySQL-MMM

2.2 方案

使用5台RHEL 6虚拟机，如图-2所示。其中192.168.4.10、192.168.4.11作为MySQL双主服务器，192.168.4.12、192.168.4.13作为主服务器的从服务器，192.168.4.100作为MySQL-MMM架构中管理监控服务器，实现监控MySQL主从服务器的工作状态及决定故障节点的移除或恢复工作，架构搭建完成后使用客户机192.168.4.120进行访问，客户机需要安装MySQL-client软件包。

Top

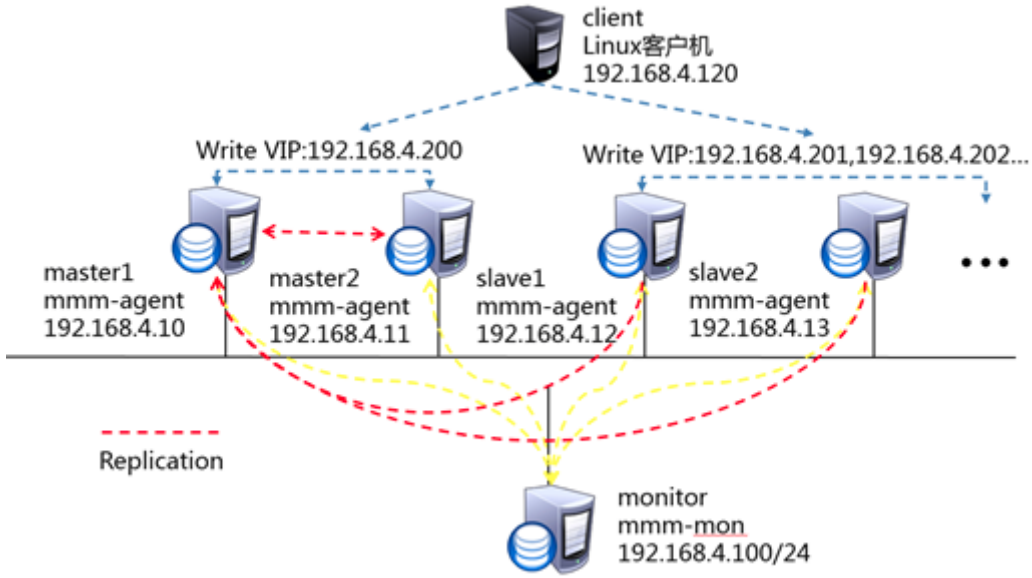

图-2

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装MySQL-MMM

1）安装依赖关系（MySQL集群内5台服务器master1，master2，slave1，slave2，monitor）均需安装

```
[root@master1 ~]# yum -y install gcc* perl-Date-Manip perl-Date-Manip perl-Date-Manip perl-XML-DOM-XPath perl-XML-Parser perl-XML-RegE
. . .
```

2）安装MySQL-MMM软件依赖包（MySQL集群内5台服务器master1，master2，slave1，slave2，monitor）均需安装，软件包讲师提供

安装安装Log-Log4perl 类

Top

```
[root@mysql-master1 ~]# rpm -ivh perl-Log-Log4perl-1.26-1.el6.rf.noarch.rpm
warning: perl-Log-Log4perl-1.26-1.el6.rf.noarch.rpm: Header V3 DSA/SHA1 Signature, key ID 6b8d79e6: NOKEY
Preparing...                ########################################### [100%]
1:perl-Log-Log4perl         ########################################### [100%]
```

安装Algorithm-Diff类

```
[root@mysql-master1 ~]# tar -zxvf Algorithm-Diff-1.1902.tar.gz  //解压安装包
...
[root@mysql-master1 ~]# cd Algorithm-Diff-1.1902                //切换到安装目录
[root@mysql-master1 Algorithm-Diff-1.1902]# perl Makefile.PL    //生成makefile文件
Checking if your kit is complete...
Looks good
Writing Makefile for Algorithm::Diff

[root@mysql-master1 Algorithm-Diff-1.1902]# make && make install       //编译，编译安装
...
[root@mysql-master1 Algorithm-Diff-1.1902]# cd                  //切换到软件包目录
[root@mysql-master1 ~]#
```

安装Proc-Daemon类

```
[root@mysql-master1 ~]# tar -zxvf Proc-Daemon-0.03.tar.gz    //解压安装包
...
[root@mysql-master1 ~]# cd Proc-Daemon-0.03                  //切换到安装目录
[root@mysql-master1 Proc-Daemon-0.03]# perl Makefile.PL    //生成makefile文件
```

Top

```
Checking if your kit is complete...
Looks good
Writing Makefile for Proc::Daemon
[root@mysql-master1 Proc-Daemon-0.03]# make && make install    //编译，编译安装
... ...
[root@mysql-master1 Proc-Daemon-0.03]# cd                      //切换到软件包目录
[root@mysql-master1 ~]#
```

安装Net-ARP虚拟IP分配工具：

```
[root@mysql-master1 ~]# gunzip Net-ARP-1.0.8.tgz    //使用gunzip解压tgz格式的安装包
[root@mysql-master1 ~]# tar xvf Net-ARP-1.0.8.tar          //解压tar安装包
... ...
[root@mysql-master1 ~]# cd Net-ARP-1.0.8                   //切换到安装目录
[root@mysql-master1 Net-ARP-1.0.8]# perl Makefile.PL       //生成makefile文件
Module Net::Pcap is required for make test!
Checking if your kit is complete...
Looks good
Writing Makefile for Net::ARP
[root@mysql-master1 Net-ARP-1.0.8]# make && make install    //编译，编译安装
... ...
[root@mysql-master1 Net-ARP-1.0.8]# cd                     //切换到软件包目录
[root@mysql-master1 ~]#
```

Top

安装Mysql-MMM软件包：

```
[root@mysql-master1 ~]# tar xvf mysql-mmm-2.2.1.tar.gz       //解压安装包
...
[root@mysql-master1 ~]# cd mysql-mmm-2.2.1                   //切换到安装目录
[root@mysql-master1 mysql-mmm-2.2.1]# make && make install   //编译，编译安装
...
[root@mysql-master1 mysql-mmm-2.2.1]#
```

步骤二：修改配置文件

1）修改公共配置文件

本案例中MySQL集群的5台服务器（master1、master2、slave1、slave2、monitor）都需要配置，可以先配好一台后使用scp复制。

```
[root@master1 ~]# vim  /etc/mysql-mmm/mmm_common.conf
active_master_role    writer

<host default>
    cluster_interface       eth0             //设置主从同步的用户

    pid_path                /var/run/mmm_agentd.pid
    bin_path                /usr/lib/mysql-mmm/

 replication_user       slaveuser          //设置主从同步的用户
 replication_password   pwd123             //设置主从同步用户密码

    agent_user          agent              //mmm-agent控制数据库用户
    agent_password      agent              //mmm-agent控制数据库用户密码
</host>

```

Top

```
17.     <host master1>                    //设置第一个主服务器
18.         ip              192.168.4.10        //master1 IP 地址
19.         mode              master
20.         peer              master2           //指定另外一台主服务器
21.     </host>
22.
23.     <host master2>                    //指定另外一台主服务器
24.         ip              192.168.4.11
25.         mode              master
26.         peer              master1
27.     </host>
28.
29.     <host slave1>                      //设置第一台从服务器
30.         ip              192.168.4.12        //slave1 IP 地址
31.         mode              slave           //本段落配置的是slave服务器
32.     </host>
33.
34.     <host slave2>
35.         ip              192.168.4.13
36.         mode              slave
37.     </host>
38.
39.     <role writer>                     //设置写入服务器工作模式
40.         hosts           master1,master2     //提供写的主服务器
41.         ips             192.168.4.200       //设置VIP地址
42.         mode              exclusive         //排他模式
43.     </role>
44.
```

Top

```
<role reader>                          //设置读取服务器工作模式
    hosts           slave1,slave2      //提供读的服务器信息
    ips             192.168.4.201,192.168.4.202   //多个虚拟IP
    mode              balanced                    //均衡模式
</role>
[root@master1 ~]#
```

2）修改管理主机配置文件（monitor主机配置）

```
[root@monitor ~]# vim /etc/mysql-mmm/mmm_mon.conf
include mmm_common.conf

<monitor>
    ip                  192.168.4.100      //设置管理主机IP地址
    pid_path            /var/run/mmm_mond.pid
    bin_path            /usr/lib/mysql-mmm/
    status_path             /var/lib/misc/mmm_mond.status
    ping_ips            192.168.4.10,192.168.4.11,192.168.4.12,192.168.4.13
                                    //设置被监控数据库
</monitor>

<host default>
    monitor_user        monitor             //监控数据库MySQL用户    monitor_password      monitor             //监控数据库MySQL用户密
</host>
                                                                                                                    Top
debug 0
[root@monitor ~]#
```

3）修改客户端配置文件

master1配置

```
[root@master1 ~]# cat /etc/mysql-mmm/mmm_agent.conf
include mmm_common.conf
this master1
```

master2配置

```
[root@master2 ~]# cat /etc/mysql-mmm/mmm_agent.conf
include mmm_common.conf
this master2
```

slave1配置

```
[root@slave1 ~]# cat /etc/mysql-mmm/mmm_agent.conf
include mmm_common.conf
this slave1
```

slave2配置

Top

```
[root@slave2 ~]# cat /etc/mysql-mmm/mmm_agent.conf
```

```
include mmm_common.conf
this slave2
```

3 MySQL-MMM架构使用

3.1 问题

本案例要求基于普通版的MySQL服务器改造MMM架构，完成以下任务操作：

- 启动MMM集群架构
- 设置集群中服务器为online状态

3.2 方案

MySQL-MMM架构部署完成后需要启动，数据库端启动mmm-agent进程，管理端启动mmm-monitor进程，启动完成后设置所有数据库主机状态为online。

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：启动MMM集群架构

1）启动mmm-agent进程

master1操作：

```
[root@master1 ~]# /etc/init.d/mysql-mmm-agent start
Daemon bin: '/usr/sbin/mmm_agentd'
Daemon pid: '/var/run/mmm_agentd.pid'
Starting MMM Agent daemon... Ok
```

Top

master2操作：

```
[root@master2 ~]# /etc/init.d/mysql-mmm-agent start
Daemon bin: '/usr/sbin/mmm_agentd'
Daemon pid: '/var/run/mmm_agentd.pid'
Starting MMM Agent daemon... Ok
```

slave1操作：

```
[root@master2 ~]# /etc/init.d/mysql-mmm-agent start
Daemon bin: '/usr/sbin/mmm_agentd'
Daemon pid: '/var/run/mmm_agentd.pid'
Starting MMM Agent daemon... Ok
```

slave2操作：

```
[root@slave2 ~]# /etc/init.d/mysql-mmm-agent start
Daemon bin: '/usr/sbin/mmm_agentd'
Daemon pid: '/var/run/mmm_agentd.pid'
Starting MMM Agent daemon... Ok
```

2）启动mmm-monitor进程

monitor主机操作：

Top

```
[root@monitor ~]# /etc/init.d/mysql-mmm-monitor start
```

```
02.  Daemon bin: '/usr/sbin/mmm_mond'
03.  Daemon pid: '/var/run/mmm_mond.pid'
04.  Starting MMM Monitor daemon: Ok
```

步骤二：设置集群中服务器为online状态

控制命令只能在管理端monitor服务器上执行。

查看当前集群中各服务器状态：

```
01.  [root@monitor ~]# mmm_control show
02.    master1(192.168.4.10) master/AWAITING_RECOVERY. Roles:
03.    master2(192.168.4.11) master/AWAITING_RECOVERY. Roles:
04.    slave1(192.168.4.12) slave/AWAITING_RECOVERY. Roles:
05.    slave2(192.168.4.13) slave/AWAITING_RECOVERY. Roles:
```

设置4台数据库主机状态为online：

```
01.  [root@monitor ~]# mmm_control set_online master1
02.  OK: State of 'master1' changed to ONLINE. Now you can wait some time and check its new roles!
03.  [root@monitor ~]# mmm_control set_online master2
04.  OK: State of 'master2' changed to ONLINE. Now you can wait some time and check its new roles!
05.  [root@monitor ~]# mmm_control set_online slave1
06.  OK: State of 'slave1' changed to ONLINE. Now you can wait some time and check its new roles!
07.  [root@monitor ~]# mmm_control set_online slave2
08.  OK: State of 'slave2' changed to ONLINE. Now you can wait some time and check its new roles!
09.  [root@monitor ~]#
```

Top

再次查看当前集群中各服务器状态：

```
[root@monitor ~]# mmm_control show
  master1(192.168.4.10) master/ONLINE. Roles: writer(192.168.4.200)
  master2(192.168.4.11) master/ONLINE. Roles:
  slave1(192.168.4.12) slave/ONLINE. Roles: reader(192.168.4.201)
  slave2(192.168.4.13) slave/ONLINE. Roles: reader(192.168.4.202)

[root@monitor ~]#
```

步骤三：测试MySQL-MMM架构

1）客户机安装MySQL-client软件包

```
[root@client ~]# tar xvf MySQL-5.6.15-1.el6.x86_64.rpm-bundle.tar
...
[root@client ~]# rpm -ivh MySQL-client-5.6.15-1.el6.x86_64.rpm
...
```

2）MySQL-MMM虚拟IP访问测试

```
[root@client ~]# mysql -h192.168.4.200 -uroot -ppwd123 -e "show databases"
Warning: Using a password on the command line interface can be insecure.
+--------------------+
| Database           |
```

Top

```
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| tarena             |
| test               |
+--------------------+
[root@client ~]#

[root@client ~]# mysql -h192.168.4.200 -uroot -ppwd123 -e "show databases"
Warning: Using a password on the command line interface can be insecure.
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| tarena             |
| test               |
+--------------------+
[root@client ~]#

[root@client ~]# mysql -h192.168.4.202 -uroot -ppwd123 -e "show databases"
Warning: Using a password on the command line interface can be insecure.
+--------------------+
| Database           |
+--------------------+
| information_schema |
```

Top

```
| mysql              |
| performance_schema |
| tarena             |
| test               |
+--------------------+
[root@client ~]#
```

3）主数据库宕机测试

```
[root@master1 ~]# service mysql stop                    //停止master1上服务
Shutting down MySQL....                                 [确定]
[root@master1 ~]#

[root@monitor ~]# mmm_control show                      //查看集群内服务器状态
```

通过输出信息可以看到虚拟IP从master1切换到master2：

```
  master1(192.168.4.10) master/HARD_OFFLINE. Roles:
  master2(192.168.4.11) master/ONLINE. Roles: writer(192.168.4.200)
  slave1(192.168.4.12) slave/ONLINE. Roles: reader(192.168.4.201)
  slave2(192.168.4.13) slave/ONLINE. Roles: reader(192.168.4.202)
[root@monitor ~]#

[root@client ~]# mysql -h192.168.4.200 -uroot -ppwd123 -e "show databases"    //访问虚拟IP测试
Warning: Using a password on the command line interface can be insecure.
+--------------------+
```

Top

```
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| tarena             |
| test               |
+--------------------+
[root@client ~]#
```

Top