

# NSD DBA2 DAY03

1. [案例1：准备MHA集群环境](#)
2. [案例2：配置MHA集群环境](#)
3. [案例3：测试MHA集群](#)

## 1 案例1：准备MHA集群环境

### 1.1 问题

- 准备6台虚拟机，并按照本节规划配置好IP参数
- 在这些虚拟机之间实现SSH免密登录
- 在相应节点上安装好MHA相关的软件包

### 1.2 方案

使用6台RHEL 7虚拟机，如图-1所示。准备集群环境，安装依赖包，授权用户，配置ssh密钥对认证登录，所有节点之间互相以root密钥对认证登录，管理主机以root密钥对认证登录所有数据节点主机，配置mha集群。

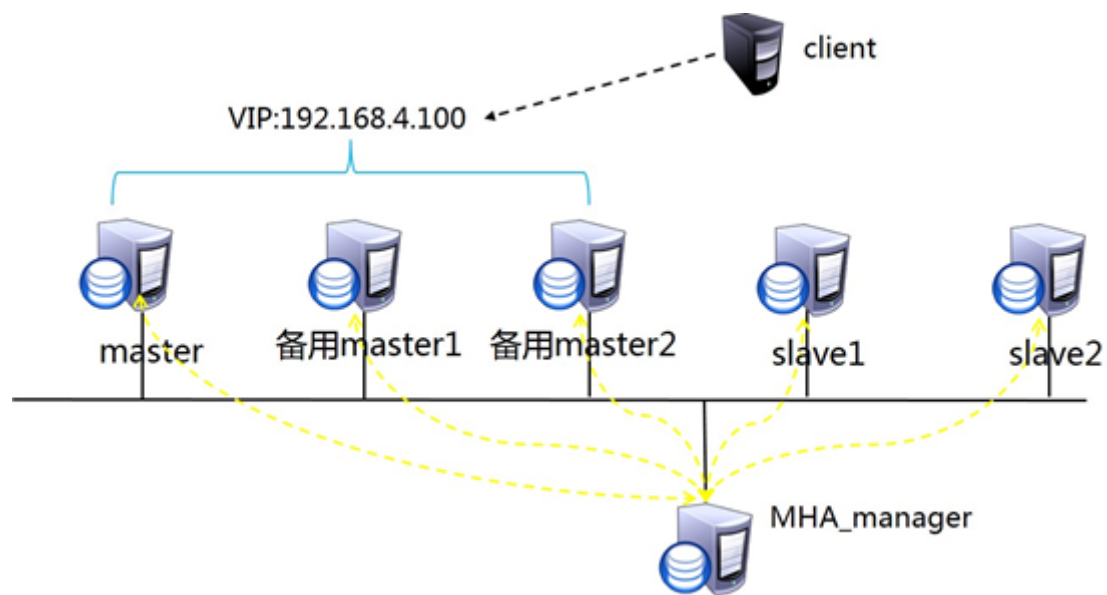


图 - 1

IP规划，如图-2所示：

[Top](#)

角色	IP地址	主机名
Master主节点服务器	192.168.4.51	master51
备用1主节点服务器	192.168.4.52	master52
备用2主节点服务器	192.168.4.53	master53
第1台 slave服务器	192.168.4.54	slave54
第2台 slave服务器	192.168.4.55	slave55
MHA_manager服务器	192.168.4.56	mgm56
VIP地址	192.168.4.100	

图-2

## 1.3

## 1.4 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：准备集群环境

1) 修改主机名，配置IP（其余几台请按照图-2修改IP和主机名，这里以master51为例）

```
01. [root@localhost ~] # echo master51 > /etc/hostname
02. [root@localhost ~] # nmcli connection modify eth0 ipv4.method manual ipv4.addresses 1
03. [root@localhost ~] # nmcli connection up eth0
```

2) 在所有主机上安装Perl依赖包（51-56操作）

```
01. [root@master51 ~] # cd mysql/mha-soft-student/
02. [root@master51 ~] # yum -y install perl-*.rpm
```

3) 在所有数据库服务器上安装mha-node包（51-55操作）

```
01. [root@master51 mha-soft-student] # yum -y install perl-DBD-mysql perl-DBI
02. [root@master51 mha-soft-student] # rpm -ivh mha4mysql-node-0.56-0.el6.noarch.rpm
03. Preparing... ##### [ 100%]
04. Updating / installing... Top
05. 1:mha4mysql-node-0.56-0.el6 ##### [ 100%]
```

#### 4) 在管理主机上安装mha\_node 和 mha-manager包 ( 56操作 )

```

01. [ root@mgm56 mha-soft-student ] # yum -y install perl-DBD-mysql perl-DBI
02. [ root@mgm56 mha-soft-student ] # rpm -ivh mha4mysql-node-0.56-0.el6.noarch.rpm
03. Preparing... ##### [ 100%]
04. Updating / installing...
05. 1:mha4mysql-node-0.56-0.el6 ##### [ 100%]
06. [ root@mgm56 mha-soft-student ] # yum -y install perl-ExtUtils-* perl-CPAN-*
07. [ root@mgm56 mha-soft-student ] # tar -zxf mha4mysql-manager-0.56.tar.gz
08. [ root@mgm56 mha-soft-student ] # cd mha4mysql-manager-0.56/
09. [ root@mgm56 mha4mysql-manager-0.56 ] # perl Makefile.PL
10. *** Module::AutoInstall version 1.03
11. *** Checking for Perl dependencies...
12. [ Core Features ]
13. - DBI ...loaded. ( 1.627)
14. - DBD::mysql ...loaded. ( 4.023)
15. - Time::HiRes ...loaded. ( 1.9725)
16. - Config::Tiny ...loaded. ( 2.14)
17. - Log::Dispatch ...loaded. ( 2.41)
18. - Parallel::ForkManager ...loaded. ( 1.18)
19. - MHA::NodeConst ...loaded. ( 0.56)
20. *** Module::AutoInstall configuration finished. //配置完成
21. Checking if your kit is complete...
22. Looks good
23. Writing Makefile for mha4mysql::manager
24. Writing MYMETA.yml and MYMETA.json
25. [ root@mgm56 mha4mysql-manager-0.56 ] # make
26. [ root@mgm56 mha4mysql-manager-0.56 ] # make install

```

#### 步骤二：配置ssh密钥对认证登陆

##### 1) 所有节点之间可以互相以ssh密钥对方式认证登陆以 ( 以51为例 )

```

01. [ root@master51 mha-soft-student ] # ssh-keygen
02. [ root@master51 mha-soft-student ] # ssh-copy-id 192.168.4.52
03. //除了传给52外，53，54，55也要传，52-55主机也是一样的

```

#### 6) 配置56主机 无密码ssh登录所有数据节点主机

[Top](#)

01. [ root@mgm56 mha4mysql-manager-0.56] # ssh-keygen
02. [ root@mgm56 mha4mysql-manager-0.56] # ssh-copy-id 192.168.4.51
03. //除传给51外，还要传给52-55

## 2 案例2：配置MHA集群环境

### 2.1 问题

- 配置主节点 master51
- 配置两个备用主节点 master52、master53
- 配置两个从节点 slave54、slave55
- 配置管理节点 mgm56

1.

### 2.2 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：配置mha集群环境

1) 安装数据库 (51-55同样操作，以51为例)

01. [ root@master51 ~] # cd /root/mysql
02. [ root@master51 mysql] # tar -xvf mysql-5.7.17.tar
03. [ root@master51 mysql] # yum -y install perl-JSON
04. [ root@master51 mysql] # rpm -Uvh mysql-community-\*.rpm
05. [ root@master51 mysql] # rpm -qa | grep -i mysql

2) master51 数据库服务器配置文件

01. [ root@master51 mysql] # vim /etc/my.cnf
02. plugin-load="rpl\_semi\_sync\_master=semisync\_master.so;rpl\_semi\_sync\_slave=semisync\_
03. rpl\_semi\_sync\_master-enabled=1
04. rpl\_semi\_sync\_slave-enabled=1
05. server\_id=51
06. log-bin=master51
07. binlog-format="mixed"
- 08.
09. [ root@master51 mysql] # systemctl restart mysqld
- 10.
11. [ root@master51 mysql] # mysql -u root -p123456

[Top](#)

```

12.
13. mysql> set global relay_log_purge=off; //不自动删除本机的中继日志文件
14. Query OK, 0 rows affected ( 0.00 sec)
15.
16. mysql> grant replication slave on *.* to repluser@"%" identified by "123456";
17. //添加主从同步授权用户
18. Query OK, 0 rows affected, 1 warning ( 10.01 sec)
19.
20. mysql> show master status;
21. +-----+-----+-----+-----+-----+
22. | File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
23. +-----+-----+-----+-----+-----+
24. | master51.000003 | 441      |              |                  |                    |
25. +-----+-----+-----+-----+-----+
26. 1 row in set ( 0.00 sec)

```

### 3 ) master52数据库服务器配置文件

```

01. [ root@master52 mysql] # vim /etc/my.cnf
02. plugin-load="rpl_semi_sync_master=semisync_master.so;rpl_semi_sync_slave=semisync_s
03. rpl_semi_sync_master-enabled=1
04. rpl_semi_sync_slave-enabled=1
05. server_id=52
06. log_bin=master52
07. binlog_format="mixed"
08.
09. [ root@master52 mysql] # systemctl restart mysqld
10. [ root@master52 mysql] # mysql -u root -p123456
11. mysql> set global relay_log_purge=off;
12. mysql> change master to
13.     -> master_host="192.168.4.51",
14.     -> master_user="repluser",
15.     -> master_password="123456",
16.     -> master_log_file="master51.000003",
17.     -> master_log_pos=441;
18. Query OK, 0 rows affected, 2 warnings ( 0.01 sec)
19. mysql> start slave;
20. Query OK, 0 rows affected ( 0.01 sec)
21. mysql> show slave status\G;

```

[Top](#)

22. ...
23. Slave\_IO\_Running: Yes
24. Slave\_SQL\_Running: Yes
25. ...

#### 4 ) master53数据库服务器配置文件

01. [ root@master53 my sql] # vim /etc/my.cnf
02. plugin-load="rpl\_semi\_sync\_master=semisync\_master.so;rpl\_semi\_sync\_slave=semisync\_slave.so"
03. rpl\_semi\_sync\_master\_enabled=1
04. rpl\_semi\_sync\_slave\_enabled=1
05. server\_id=53
06. log\_bin=master53
07. binlog\_format="mixed"
- 08.
09. [ root@master53 my sql] # systemctl restart mysqld
10. [ root@master53 my sql] # mysql -u root -p123456
11. mysql> set global relay\_log\_purge=off;
12. Query OK, 0 rows affected ( 0.00 sec)
- 13.
14. mysql> change master to
15. -> master\_host="192.168.4.51",
16. -> master\_user="repluser",
17. -> master\_password="123456",
18. -> master\_log\_file="master51.000003",
19. -> master\_log\_pos=441;
20. Query OK, 0 rows affected, 2 warnings ( 0.01 sec)
21. mysql> start slave;
22. Query OK, 0 rows affected ( 0.00 sec)
23. mysql> show slave status\G;
24. ...
25. Slave\_IO\_Running: Yes
26. Slave\_SQL\_Running: Yes
27. ...

#### 5 ) slave54 数据库服务器配置文件

[Top](#)

01. [ root@slave54 my sql] # vim /etc/my.cnf

```

02. server_id=54
03. [ root@master54 my sql] # systemctl restart mysqld
04. [ root@master54 my sql] # mysql -u root -p123456
05. mysql> change master to
06.     -> master_host="192.168.4.51",
07.     -> master_user="repluser",
08.     -> master_password="123456",
09.     -> master_log_file="master51.000003",
10.     -> master_log_pos=441;
11. Query OK, 0 rows affected, 2 warnings ( 0.01 sec)
12. mysql> start slave;
13. Query OK, 0 rows affected ( 0.00 sec)
14. mysql> show slave status\G;
15. ...
16.         Slave_IO_Running: Yes
17.         Slave_SQL_Running: Yes
18. ...

```

## 6 ) slave55 数据库服务器配置文件

```

01. [ root@slave55 my sql] # vim /etc/my.cnf
02. server_id=55
03.
04. [ root@master55 my sql] # systemctl restart mysqld
05. [ root@master55 my sql] # mysql -u root -p123456
06. mysql> change master to
07.     -> master_host="192.168.4.51",
08.     -> master_user="repluser",
09.     -> master_password="123456",
10.     -> master_log_file="master51.000003",
11.     -> master_log_pos=441;
12. Query OK, 0 rows affected, 2 warnings ( 0.01 sec)
13. mysql> start slave;
14. Query OK, 0 rows affected ( 0.00 sec)
15. mysql> show slave status\G;
16. ...
17.         Slave_IO_Running: Yes
18.         Slave_SQL_Running: Yes
19. ...

```

[Top](#)

## 7) 配置管理主机4.56

```
01. [root@mgm56 ~]# cd mysql/mha-soft-student/mha4mysql-manager-0.56/
02. [root@mgm56 mha4mysql-manager-0.56]# cp bin/* /usr/local/bin/
03. //提示覆盖，说明安装的时候有，没有可以拷贝过来
04. [root@mgm56 mha4mysql-manager-0.56]# mkdir /etc/mha_manager //创建工作目录
05. [root@mgm56 mha4mysql-manager-0.56]# cp samples/conf/app1.cnf /etc/mha_manag
06. //建立样板文件
07. [root@mgm56 mha4mysql-manager-0.56]# vim /etc/mha_manager/app1.cnf
08. //编辑主配置文件app1.cnf
09. [server default]
10. manager_workdir=/etc/mha_manager
11. manager_log=/etc/mha_manager/manager.log
12. master_ip_failover_script=/usr/local/bin/master_ip_failover
13.
14. ssh_user=root
15. ssh_port=22
16. repl_user=repluser
17. repl_password=123456
18. user=root
19. password=123456
20.
21. [server1]
22. hostname=192.168.4.51
23. port=3306
24.
25. [server2]
26. hostname=192.168.4.52
27. port=3306
28. candidate_master=1
29.
30. [server3]
31. hostname=192.168.4.53
32. port=3306
33. candidate_master=1
34.
35. [server4]
36. hostname=192.168.4.54
37. no_master=1
38.
39. [server5]
```

[Top](#)



```

40.  hostname=192.168.4.55
41.  no_master=1
42.  [ root@mgm56 mha4mysql-manager-0.56 ] # cp samples/scripts/master_ip_failover
43.  /usr/local/bin/      //创建故障切换的脚本

```

### 3 案例3：测试MHA集群

#### 3.1 问题

- 查看MHA集群状态
- 测试节点之间的SSH登录
- 测试集群VIP的故障切换功能

#### 3.2 步骤

实现此案例需要按照如下步骤进行。

##### 步骤一：验证配置

1) 检查配置环境，在主机52，53检查是否有同步数据的用户repluser

主机52:

```

01.  mysql> select user,host from mysql.user where user="repluser";
02.  +-----+-----+
03.  | user  | host |
04.  +-----+-----+
05.  | repluser | % |
06.  +-----+-----+
07.  1 row in set ( 0.00 sec)
08.
09.  mysql> show grants for repluser@"%";
10.  +-----+
11.  | Grants for repluser@% |
12.  +-----+
13.  | GRANT REPLICATION SLAVE ON *.* TO 'repluser'@'%' |
14.  +-----+
15.  1 row in set ( 0.00 sec)

```

主机53:

```

01.  mysql> select user,host from mysql.user where user="repluser";
02.  +-----+-----+

```

[Top](#)

```

03.  | user | host |
04.  +-----+-----+
05.  | repluser | % |
06.  +-----+-----+
07.  1 row in set ( 0.00 sec)
08.
09.  my sql> show grants for repluser@"%";
10.  +-----+-----+
11.  | Grants for repluser@% |
12.  +-----+-----+
13.  | GRANT REPLICATION SLAVE ON *.* TO 'repluser'@'%' |

+-----+

01.  1 row in set ( 0.00 sec)

```

2) 在51的主机上做root的授权，其他的会同步（如果不做，在验证数据节点的主从同步配置时会出错）

```

01.  my sql> grant all on *.* to root@"%" identified by "123456";
02.  my sql> select user,host from my sql.user where user="root";
03.  +-----+-----+
04.  | user | host |
05.  +-----+-----+
06.  | root | % |
07.  | root | localhost |
08.  +-----+-----+
09.  2 rows in set ( 0.00 sec)

```

### 3) 验证ssh 免密登陆数据节点主机

```

01.  [ root@mgm56 mha4my sql- manager- 0.56] # cd /usr/local/bin/
02.  [ root@mgm56 bin] # masterha_check_ssh --conf=/etc/mha_manager/app1.cnf
03.  Wed Sep 19 09:09:33 2018 - [ info] All SSH connection tests passed successfully.
04.  //出现这个为成功

```

[Top](#)

### 4) 验证数据节点的主从同步配置（先把自动failover时候的切换脚本注释掉）

01. [ root@mgm56 bin] # masterha\_check\_repl -- conf=/etc/mha\_manager/app1.cnf
02. My SQL Replication Health is OK. //验证成功

## 5) 启动管理服务MHA\_Manager

--remove\_dead\_master\_conf //删除宕机主库配置

--ignore\_last\_failover //忽略xxx.health文件

01. [ root@mgm56 bin] # masterha\_manager -- conf=/etc/mha\_manager/app1.cnf \
02. -- remove\_dead\_master\_conf -- ignore\_last\_failover
- 03.
04. Wed Sep 19 09:24:41 2018 - [ warning] Global configuration file /etc/masterha\_default.cn
05. Wed Sep 19 09:24:41 2018 - [ info] Reading application default configuration from /etc/mf
06. Wed Sep 19 09:24:41 2018 - [ info] Reading server configuration from /etc/mha\_manager,

## 6) 查看状态 ( 另开一个终端 )

01. [ root@mgm56 ~] # masterha\_check\_status -- conf=/etc/mha\_manager/app1.cnf
02. app1( pid: 15745) is running( 0: PING\_OK) , master: 192.168.4.51

## 7) 停止服务

01. [ root@mgm56 ~] # masterha\_stop -- conf=/etc/mha\_manager/app1.cnf
02. Stopped app1 successfully .

## 步骤二：测试故障转移

### 1) 在主库51上面配置VIP地址

01. [ root@master51 ~] # ifconfig eth0:1 192.168.4.100/24

### 2) 在配置文件里面把自动failover时候的切换脚本去掉注释

### 3) 修改 master\_ip\_failover 脚本，设置如下内容

01. 34 my \$vip = '192.168.4.100/24';

[Top](#)

02. 35 my \$key = "1";
03. 36 my \$ssh\_start\_vip = "/sbin/ifconfig eth0: \$key \$vip";
04. 37 my \$ssh\_stop\_vip = "/sbin/ifconfig eth0: \$key down";

#### 4) 启动服务

01. [ root@mgm56 bin] # masterha\_manager -- conf=/etc/mha\_manager/app1.cnf \
02. -- remove\_dead\_master\_conf -- ignore\_last\_failover
03. Wed Sep 19 09:50:33 2018 - [ warning] Global configuration file /etc/masterha\_default.cnf
04. Wed Sep 19 09:50:33 2018 - [ info] Reading application default configuration from /etc/ml
05. Wed Sep 19 09:50:33 2018 - [ info] Reading server configuration from /etc/mha\_manager

#### 5) 查看状态

01. [ root@mgm56 ~] # masterha\_check\_status -- conf=/etc/mha\_manager/app1.cnf
02. app1 master is down and failover is running( 50: FAILOVER\_RUNNING) . master: 192.168.4.52

验证数据节点的主从同步配置报错，如图-3所示：

01. [ root@mgm56 bin] # masterha\_check\_repl -- conf=/etc/mha\_manager/app1.cnf

```
[root@mgm56 bin]# masterha_check_repl --conf=/etc/mha_manager/app1.cnf
Wed Sep 19 09:11:56 2018 - [warning] Global configuration file /etc/masterha_default.cnf not found. Skipping.
Wed Sep 19 09:11:56 2018 - [info] Reading application default configuration from /etc/mha_manager/app1.cnf..
Wed Sep 19 09:11:56 2018 - [info] Reading server configuration from /etc/mha_manager/app1.cnf..
Wed Sep 19 09:11:56 2018 - [info] MHA::MasterMonitor version 0.56.
Wed Sep 19 09:11:56 2018 - [error][usr/local/share/perl5/MHA/ServerManager.pm, ln301] Got MySQL error when connecting 192.168.4.54(192.168.4.54:3306) :1130:Host '192.168.4.56' is not allowed to connect to this MySQL server, but this is not a MySQL crash. Check MySQL server settings.
at /usr/local/share/perl5/MHA/ServerManager.pm line 297.
Wed Sep 19 09:11:56 2018 - [error][usr/local/share/perl5/MHA/ServerManager.pm, ln301] Got MySQL error when connecting 192.168.4.55(192.168.4.55:3306) :1130:Host '192.168.4.56' is not allowed to connect to this MySQL server, but this is not a MySQL crash. Check MySQL server settings.
at /usr/local/share/perl5/MHA/ServerManager.pm line 297.
Wed Sep 19 09:11:56 2018 - [error][usr/local/share/perl5/MHA/ServerManager.pm, ln301] Got MySQL error when connecting 192.168.4.53(192.168.4.53:3306) :1045:Access denied for user 'root'@'192.168.4.56' (using password: YES), but this is not a MySQL crash. Check MySQL server settings.
at /usr/local/share/perl5/MHA/ServerManager.pm line 297.
```

图-3

解决办法：

root用户没有授权，默认只能本地连接，在主机51上面授权root用户可以远程登录，其他主机同步

[Top](#)

01. mysql> grant all on \*.\* to root@"%" identified by "123456";

[Top](#)