一、环境说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 主机名 | 操作系统 | IP地址 | 角色 | 数据库 |
| Manager | CentOS6.5 | 192.168.9.101 | Mha管理节点 |  |
| Master | CentOS6.5 | 192.168.9.102 | Mha主节点 | MySQL5.6.22 |
| Slave | CentOS6.5 | 192.168.9.103 | Mha备节点 | MySQL5.6.22 |
| Slave2 | CentOS6.5 | 192.168.9.104 | Mha备节点 | MySQL5.6.22 |
|  |  | 192.168.9.120 | VIP |  |

以各主机上关闭iptables ,selinux=disabled

二、安装MySQL5.6

## 2.1 安装前的准备

创建操作系统层的MySQL专用账户和用户组，均命名为mysql：

groupadd mysql

useradd -g mysql mysql

增加主机名与IP映射：

192.168.8.101 manager

192.168.9.102 master

192.168.8.103 slave

192.168.8.104 slave2

## 2.2 目录规划

目录说明：

${port}=mysql实例启动端口

软件安装目录： /usr/local/mysql

日志目录（redolog、errlog、binlog、slowlog、relaylog）：/data/data${port}/log

配置文件目录：/usr/local/mysql${port}

临时文件目录：/data/data${port}/tmp

数据文件目录： /data/data${port}/data

## 2.3 默认字符集与校对规则

默认字符集 ：utf8mb4

默认校对规则：utf8mb4\_general\_ci

## 2.4 编译安装数据库软件

发装依赖包：

yum -y install gcc-c++ ncurses-devel cmake make perl gcc autoconf automake zlib libxml libtool bison readline readline-devel

yum install libaio-devel

使用root用户编译mysql5.6的源码包：

解压文件，进入目录：

tar zxvf mysql-5.6.22.tar.gz

cd mysql-5.6.22

执行编译命令如下：

cmake \

-DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql \

-DINSTALL\_DATADIR=/data/data3306 \

-DDEFAULT\_CHARSET=utf8 \

-DDEFAULT\_COLLATION=utf8\_general\_ci \

-DEXTRA\_CHARSETS=all \

-DWITH\_EMBEDDED\_SERVER=1 \

-DENABLED\_LOCAL\_INFILE=1 \

-DWITH\_MYISAM\_STORAGE\_ENGINE=1 \

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 \

-DWITH\_ARCHIVE\_STORAGE\_ENGINE=1 \

-DWITH\_BLACKHOLE\_STORAGE\_ENGINE=1 \

-DWITH\_FEDERATED\_STORAGE\_ENGINE=1 \

-DWITH\_PARTITION\_STORAGE\_ENGINE=1 \

-DMYSQL\_TCP\_PORT=3306 \

-DENABLED\_LOCAL\_INFILE=1 \

-DSYSCONFDIR=/etc \

-DWITH\_READLINE=on

在当前目录（mysql-5.6.23）下，安装mysql数据库软件：

make && make install

## 2.5 配置相关目录

分别在主库、从库上执行：

|  |
| --- |
| mkdir -p /usr/local/mysql  mkdir -p /usr/local/mysql3306  chown -R mysql:mysql /usr/local/mysql  chown -R mysql:mysql /usr/local/mysql3306  mkdir -p /data/data3306  mkdir -p /data/data3306/data  mkdir -p /data/data3306/tmp  mkdir -p /data/data3306/log/relaylog  mkdir -p /data/data3306/log/errlog  mkdir -p /data/data3306/log/slowlog  mkdir -p /data/data3306/log/redolog  mkdir -p /data/data3306/log/undolog  chown -R mysql:mysql /data/data3306/ |

## 2.6 编辑配置文件

将my.cnf文件上传到/usr/local/mysql3306目录下。

这两个文件的具体内容如下：

**编辑配置文件：/usr/local/mysql3360/my.cnf**

|  |
| --- |
| [client]  #password = your\_password  port= 3306  socket= /data/data3306/tmp/mysql3306.sock  #default-character-set=utf8  #safe\_updates  [mysqld]  server\_id=1  expire\_logs\_days = 3  port =3306  socket=/data/data3306/tmp/mysql3306.sock  pid\_file=/data/data3306/tmp/mysql3306.pid  lower\_case\_table\_names=0  #gtid  gtid-mode=on  enforce-gtid-consistency=true  log-slave-updates  explicit\_defaults\_for\_timestamp  master\_verify\_checksum=1  binlog\_rows\_query\_log\_events = 1  master\_info\_repository = TABLE  basedir=/usr/local/mysql  datadir=/data/data3306/data  tmpdir=/data/data3306/tmp  transaction\_isolation=READ-COMMITTED  federated  performance\_schema=on  skip\_name\_resolve  binlog\_format=ROW  binlog\_rows\_query\_log\_events = 1  #binlog\_row\_image=minimal  skip-external-locking  key\_buffer\_size = 16M  myisam\_sort\_buffer\_size = 8M  #Try number of CPU's\*2 for thread\_concurrency  back\_log = 1024  max\_connections = 4500  table\_definition\_cache=400  table\_open\_cache=256  table\_open\_cache\_instances = 8  max\_connect\_errors = 300  max\_allowed\_packet = 4M  read\_rnd\_buffer\_size = 1M  read\_buffer\_size = 1M  join\_buffer\_size = 1M  sort\_buffer\_size = 1M  query\_cache\_limit = 2M  query\_cache\_type = 1  query\_cache\_size = 10M  query\_cache\_min\_res\_unit=2k  thread\_cache\_size = 1200  thread\_stack = 256K  tmp\_table\_size = 24M  max\_tmp\_tables = 256  max\_heap\_table\_size=256M  bulk\_insert\_buffer\_size = 4M  binlog\_cache\_size = 2M  max\_binlog\_size = 1024M  max\_binlog\_cache\_size= 1024M  sync\_binlog = 1  event\_scheduler = 1  master\_verify\_checksum = 1  #log  log-bin= mysql\_bin  log\_bin\_index = mysql\_bin.index  log-error= /data/data3306/log/errlog/mysql.err  relay\_log\_purge = 1  relay\_log\_recovery = 1  slow-query-log  slow-query-log-file=/data/data3306/log/slowlog/slow.log  long\_query\_time=1  log-queries-not-using-indexes  #innodb  innodb\_buffer\_pool\_size = 20M  innodb\_data\_home\_dir = /data/data3306/data  innodb\_data\_file\_path = ibdata1:10M:autoextend  innodb\_autoextend\_increment = 100  innodb\_file\_per\_table= 1  innodb\_log\_group\_home\_dir = /data/data3306/log  innodb\_log\_files\_in\_group=2  innodb\_log\_file\_size = 20M  innodb\_log\_buffer\_size = 8M  innodb\_thread\_concurrency = 0  innodb\_lock\_wait\_timeout = 100  innodb\_flush\_method = 'O\_DIRECT'  innodb\_max\_dirty\_pages\_pct = 75  innodb\_flush\_log\_at\_trx\_commit=1  innodb\_support\_xa = 0  innodb\_io\_capacity=400  innodb\_stats\_on\_metadata=0  innodb\_strict\_mode = 1  innodb\_print\_all\_deadlocks = 1  innodb\_buffer\_pool\_dump\_at\_shutdown = 1  innodb\_buffer\_pool\_load\_at\_startup = 1  innodb\_undo\_directory=/data/data3306/undolog  innodb\_undo\_logs=128  innodb\_undo\_tablespaces=4  innodb\_autoinc\_lock\_mode=2  [mysqldump]  quick  max\_allowed\_packet = 2M  [mysql]  no-auto-rehash  # Remove the next comment character if you are not familiar with SQL  #safe-updates  [myisamchk]  key\_buffer\_size = 8M  sort\_buffer\_size = 8M  read\_buffer = 2M  write\_buffer = 2M  [mysqlhotcopy]  interactive-timeout |

## 2.7 初始化数据库

### 2.7.1 初始化数据库

|  |
| --- |
| cd /usr/local/mysql  ./scripts/mysql\_install\_db -defaults-file=/usr/local/mysql3306/my.cnf --basedir=/usr/local/mysql --datadir=/data/data3306/data --user=mysql |

### 2.7.2 初始化日志

|  |
| --- |
| [root@master mysql]# ./scripts/mysql\_install\_db -defaults-file=/usr/local/mysql3306/my.cnf --basedir=/usr/local/mysql --datadir=/data/data3306/data --user=mysql  Installing MySQL system tables...[root@master mysql]#  [root@master mysql]#  [root@master mysql]#  [root@master mysql]#  [root@master mysql]#  [root@master mysql]# ./scripts/mysql\_install\_db -defaults-file=/usr/local/mysql3306/my.cnf --basedir=/usr/local/mysql --datadir=/data/data3306/data --user=mysql  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:  /usr/local/mysql/bin/mysqladmin -u root password 'new-password'  /usr/local/mysql/bin/mysqladmin -u root -h master password 'new-password'  Alternatively you can run:  /usr/local/mysql/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 . ; /usr/local/mysql/bin/mysqld\_safe &  You can test the MySQL daemon with mysql-test-run.pl  cd mysql-test ; perl mysql-test-run.pl  Please report any problems at http://bugs.mysql.com/  The latest information about MySQL is available on the web at  http://www.mysql.com  Support MySQL by buying support/licenses at http://shop.mysql.com  WARNING: Found existing config file /usr/local/mysql/my.cnf on the system.  Because this file might be in use, it was not replaced,  but was used in bootstrap (unless you used --defaults-file)  and when you later start the server.  The new default config file was created as /usr/local/mysql/my-new.cnf,  please compare it with your file and take the changes you need.  WARNING: Default config file /etc/my.cnf exists on the system  This file will be read by default by the MySQL server  If you do not want to use this, either remove it, or use the  --defaults-file argument to mysqld\_safe when starting the server |

### 2.7.3 目录结构

|  |
| --- |
| [root@master ~]# tree /data/data3306/  /data/data3306/  ├── data  │   ├── ib\_buffer\_pool  │   ├── ibdata1  │   ├── mysql  │   │   ├── columns\_priv.frm  │   │   ├── columns\_priv.MYD  │   │   ├── columns\_priv.MYI  │   │   ├── db.frm  │   │   ├── db.MYD  │   │   ├── db.MYI  │   │   ├── event.frm  │   │   ├── event.MYD  │   │   ├── event.MYI  │   │   ├── func.frm  │   │   ├── func.MYD  │   │   ├── func.MYI  │   │   ├── general\_log.CSM  │   │   ├── general\_log.CSV  │   │   ├── general\_log.frm  │   │   ├── help\_category.frm  │   │   ├── help\_category.MYD  │   │   ├── help\_category.MYI  │   │   ├── help\_keyword.frm  │   │   ├── help\_keyword.MYD  │   │   ├── help\_keyword.MYI  │   │   ├── help\_relation.frm  │   │   ├── help\_relation.MYD  │   │   ├── help\_relation.MYI  │   │   ├── help\_topic.frm  │   │   ├── help\_topic.MYD  │   │   ├── help\_topic.MYI  │   │   ├── innodb\_index\_stats.frm  │   │   ├── innodb\_index\_stats.ibd  │   │   ├── innodb\_table\_stats.frm  │   │   ├── innodb\_table\_stats.ibd  │   │   ├── ndb\_binlog\_index.frm  │   │   ├── ndb\_binlog\_index.MYD  │   │   ├── ndb\_binlog\_index.MYI  │   │   ├── plugin.frm  │   │   ├── plugin.MYD  │   │   ├── plugin.MYI  │   │   ├── proc.frm  │   │   ├── proc.MYD  │   │   ├── proc.MYI  │   │   ├── procs\_priv.frm  │   │   ├── procs\_priv.MYD  │   │   ├── procs\_priv.MYI  │   │   ├── proxies\_priv.frm  │   │   ├── proxies\_priv.MYD  │   │   ├── proxies\_priv.MYI  │   │   ├── servers.frm  │   │   ├── servers.MYD  │   │   ├── servers.MYI  │   │   ├── slave\_master\_info.frm  │   │   ├── slave\_master\_info.ibd  │   │   ├── slave\_relay\_log\_info.frm  │   │   ├── slave\_relay\_log\_info.ibd  │   │   ├── slave\_worker\_info.frm  │   │   ├── slave\_worker\_info.ibd  │   │   ├── slow\_log.CSM  │   │   ├── slow\_log.CSV  │   │   ├── slow\_log.frm  │   │   ├── tables\_priv.frm  │   │   ├── tables\_priv.MYD  │   │   ├── tables\_priv.MYI  │   │   ├── time\_zone.frm  │   │   ├── time\_zone\_leap\_second.frm  │   │   ├── time\_zone\_leap\_second.MYD  │   │   ├── time\_zone\_leap\_second.MYI  │   │   ├── time\_zone.MYD  │   │   ├── time\_zone.MYI  │   │   ├── time\_zone\_name.frm  │   │   ├── time\_zone\_name.MYD  │   │   ├── time\_zone\_name.MYI  │   │   ├── time\_zone\_transition.frm  │   │   ├── time\_zone\_transition.MYD  │   │   ├── time\_zone\_transition.MYI  │   │   ├── time\_zone\_transition\_type.frm  │   │   ├── time\_zone\_transition\_type.MYD  │   │   ├── time\_zone\_transition\_type.MYI  │   │   ├── user.frm  │   │   ├── user.MYD  │   │   └── user.MYI  │   ├── mysql\_bin.000001  │   ├── mysql\_bin.000002  │   ├── mysql\_bin.index  │   ├── performance\_schema  │   │   ├── accounts.frm  │   │   ├── cond\_instances.frm  │   │   ├── db.opt  │   │   ├── events\_stages\_current.frm  │   │   ├── events\_stages\_history.frm  │   │   ├── events\_stages\_history\_long.frm  │   │   ├── events\_stages\_summary\_by\_account\_by\_event\_name.frm  │   │   ├── events\_stages\_summary\_by\_host\_by\_event\_name.frm  │   │   ├── events\_stages\_summary\_by\_thread\_by\_event\_name.frm  │   │   ├── events\_stages\_summary\_by\_user\_by\_event\_name.frm  │   │   ├── events\_stages\_summary\_global\_by\_event\_name.frm  │   │   ├── events\_statements\_current.frm  │   │   ├── events\_statements\_history.frm  │   │   ├── events\_statements\_history\_long.frm  │   │   ├── events\_statements\_summary\_by\_account\_by\_event\_name.frm  │   │   ├── events\_statements\_summary\_by\_digest.frm  │   │   ├── events\_statements\_summary\_by\_host\_by\_event\_name.frm  │   │   ├── events\_statements\_summary\_by\_thread\_by\_event\_name.frm  │   │   ├── events\_statements\_summary\_by\_user\_by\_event\_name.frm  │   │   ├── events\_statements\_summary\_global\_by\_event\_name.frm  │   │   ├── events\_waits\_current.frm  │   │   ├── events\_waits\_history.frm  │   │   ├── events\_waits\_history\_long.frm  │   │   ├── events\_waits\_summary\_by\_account\_by\_event\_name.frm  │   │   ├── events\_waits\_summary\_by\_host\_by\_event\_name.frm  │   │   ├── events\_waits\_summary\_by\_instance.frm  │   │   ├── events\_waits\_summary\_by\_thread\_by\_event\_name.frm  │   │   ├── events\_waits\_summary\_by\_user\_by\_event\_name.frm  │   │   ├── events\_waits\_summary\_global\_by\_event\_name.frm  │   │   ├── file\_instances.frm  │   │   ├── file\_summary\_by\_event\_name.frm  │   │   ├── file\_summary\_by\_instance.frm  │   │   ├── host\_cache.frm  │   │   ├── hosts.frm  │   │   ├── mutex\_instances.frm  │   │   ├── objects\_summary\_global\_by\_type.frm  │   │   ├── performance\_timers.frm  │   │   ├── rwlock\_instances.frm  │   │   ├── session\_account\_connect\_attrs.frm  │   │   ├── session\_connect\_attrs.frm  │   │   ├── setup\_actors.frm  │   │   ├── setup\_consumers.frm  │   │   ├── setup\_instruments.frm  │   │   ├── setup\_objects.frm  │   │   ├── setup\_timers.frm  │   │   ├── socket\_instances.frm  │   │   ├── socket\_summary\_by\_event\_name.frm  │   │   ├── socket\_summary\_by\_instance.frm  │   │   ├── table\_io\_waits\_summary\_by\_index\_usage.frm  │   │   ├── table\_io\_waits\_summary\_by\_table.frm  │   │   ├── table\_lock\_waits\_summary\_by\_table.frm  │   │   ├── threads.frm  │   │   └── users.frm  │   └── test  ├── log  │   ├── errlog  │   │   └── mysql.err  │   ├── ib\_logfile0  │   ├── ib\_logfile1  │   ├── redolog  │   ├── relaylog  │   ├── slowlog  │   └── undolog  ├── tmp  └── undolog  ├── undo001  ├── undo002  ├── undo003  └── undo004  12 directories, 144 files |

### 2.7.4 启动数据库

|  |
| --- |
| /usr/local/mysql/bin/mysqld\_safe --defaults-file=/usr/local/mysql3306/my.cnf & |

### 2.7.5 连接数据库

|  |
| --- |
| /usr/local/mysql/bin/mysql -uroot -p --socket=/data/data3306/tmp/mysql3306.sock --port=3306 |

### 2.7.6 修改root密码

|  |
| --- |
| mysql> SET PASSWORD FOR 'root'@localhost = PASSWORD('123456');  mysql> flush privileges; |

## 2.8 MySQL数据库启动与关闭

### 2.8.1 建立安全登陆文件

|  |
| --- |
| /usr/local/mysql/bin/mysql\_config\_editor remove --login-path=mysql3306  /usr/local/mysql/bin/mysql\_config\_editor set --login-path=mysql3306 --user=root --host=localhost --port=3306 --password --socket=/data/data3306/tmp/mysql3306.sock  [mysql@master ~]$ /usr/local/mysql/bin/mysql\_config\_editor print --all  [mysql3306]  user = root  password = \*\*\*\*\*  host = localhost  socket = /data/data3306/tmp/mysql3306.sock  port = 3306 |

### 2.8.2 启动数据库

|  |
| --- |
| 启动数据库：  sh mysql\_startup.sh 3306  启动脚本：  cat mysql\_startup.sh  #!/bin/sh  port1=$1  /usr/local/mysql/bin/mysqld\_safe --defaults-file=/usr/local/mysql$port1/my.cnf & |

### 2.8.3 关闭数据库

|  |
| --- |
| 关闭数据库：  sh mysql\_shutdown.sh 3306  关库脚本：  cat mysql\_shutdown.sh  #!/bin/sh  port1=$1  /usr/local/mysql/bin/mysqladmin --login-path=mysql$port1 shutdown |

### 2.8.4 设置登陆快捷键

|  |
| --- |
| su – mysql  vi .bash\_profile  mysql3306="/usr/local/mysql/bin/mysql --login-path=mysql3306"  alias mysql3306=$mysql3306  source ~/.bash\_profile |

三、MySQL主从复制环境搭建

## 3.1 离线搭备库

（1）按照第二章的方法在另一台主机上，通过源码安装的方式，安装mysql备库（各种路径、参数文件与主库完全一致）。

（2）在主库创建复制账户（包含集群中所有mysql server的ip）：

grant replication slave,replication client on \*.\* to 'myrpl'@'192.168.9.102' identified by '123456';

grant replication slave,replication client on \*.\* to 'myrpl'@'192.168.9.103' identified by '123456';

（3）关闭主库

sh mysql\_shutdown.sh 3306

（4）拷贝主库所有数据文件到备库所在的主机的相应目录下（/data/data3306）。

（5）在备库做如下设置

在my.cnf参数文件中重新设置备库的server\_id参数（命名规范为ip地址后两位加端口号，例如ip为10.120.5.22，端口为3540，server\_id参数配置为：5223540）。

删除备库的auto.cnf。

（6）启动备库

sh mysql\_startup.sh 3306

（7）登录备库

mysql3306

（8）建立主从复制关系（10.120.5.21为主库ip）：

change master to

master\_host='192.168.9.102',

master\_port=3306,

master\_user='myrpl',

master\_password='MzQxYjU5NzNlMTk3',

master\_auto\_position=1;

（9）启动复制线程

start slave;

（10）确认复制正常

show slave status\G

（11）修改从库参数

#将read\_only参数设置为ON：

set global read\_only=1;

## 3.2 在线搭备库

### 3.2.1 安装数据库

按照第三章的方法在另一台主机（192.168.9.104）上，通过源码安装的方式，安装mysql备库（各种路径、参数文件与主库完全一致）。

当前主从结构：

192.168.9.102 主库

192.168.9.103 备库（通过上一节的方法，离线搭建的备库）

192.168.9.104 准备通过在线方式搭建的备库

### 3.2.2 安装xtrabackup备份恢复软件

在192.168.9.102、192.168.9.103、192.168.9.104上分别安装xtrabackup备份恢复软件。

|  |
| --- |
| rpm -ivh libev-4.15-1.el6.rf.x86\_64.rpm  yum install perl-DBD-MySQL  rpm -ivh percona-xtrabackup-24-2.4.3-1.el6.x86\_64.rpm |

### 3.2.3 在主库创建复制账户

#包含集群中所有mysql server的ip：

grant replication slave,replication client on \*.\* to 'myrpl'@'192.168.9.102' identified by '123456';

grant replication slave,replication client on \*.\* to 'myrpl'@'192.168.9.103' identified by '123456';

grant replication slave,replication client on \*.\* to 'myrpl'@'192.168.9.104' identified by '123456';

说明：如果上一节已经执行，则这个步骤可跳过。

### 3.2.4 在主库创建备份用户并授权

GRANT RELOAD, LOCK TABLES, REPLICATION CLIENT, PROCESS ON \*.\* TO 'backup\_data'@'192.168.9.%' IDENTIFIED BY '123456';

### 3.2.5 在备库建立备份目录和恢复目录

备库192.168.9.103建立备份目录和恢复目录（在nas上建，确保192.168.9.104也能访问）。

mkdir -p /dbbk/backup/fullbackup

mkdir -p /dbbk/backup/backup\_r

### 3.2.6 在192.168.9.101上部署nfs服务：

|  |
| --- |
| #开机自启动  chkconfig rpcbind on  chkconfig nfs on  #启动服务  service nfs start  service rpcbind start  #查看服务及端口  rpcinfo –p  #建立共享目录  mkdir /dbbk  #查看nfs配置文件  # more /etc/exports  /dbbk 192.168.9.102(rw,sync,no\_root\_squash)  /dbbk 192.168.9.103(rw,sync,no\_root\_squash)  /dbbk 192.168.9.104(rw,sync,no\_root\_squash)  #重新输出共享目录格式,修改配置扣不需要重启服务  [root@manager ~]# exportfs -rv  exporting 192.168.9.102:/dbbk  exporting 192.168.9.103:/dbbk  exporting 192.168.9.104:/dbbk |

### 3.2.7 挂载NFS共享目录

在192.168.9.102/103/104三台MySQL上挂载NFS共享目录

|  |
| --- |
| #创建挂载点  mkdir /dbbk  #查看NFS服务器共享目录  [root@master tools]# showmount -e 192.168.9.101  Export list for 192.168.9.101:  /dbbk 192.168.9.104,192.168.9.103,192.168.9.102  #将NFS服务器的共享目录，挂载到挂载点  mount 192.168.9.101:/dbbk /dbbk  #查看挂载点  [root@master tools]# df -h  Filesystem Size Used Avail Use% Mounted on  /dev/mapper/vg\_slave-lv\_root 18G 5.4G 11G 33% /  tmpfs 931M 0 931M 0% /dev/shm  /dev/sda1 485M 39M 421M 9% /boot  192.168.9.101:/dbbk 18G 2.2G 15G 14% /dbbk  #卸载挂载点  umount /dbbk |

### 3.2.8 在备库192.168.9.103生成备份用的配置文件

以数据库配置文件/usr/local/mysql3306/my.cnf为准

vi /dbbk/backup/backupdata.cnf

|  |
| --- |
| [client]  user=backup\_data  password="123456"  socket=/data/data3306/tmp/mysql3306.sock  [mysqld]  datadir = /data/data3306/data  character\_set\_server = utf8  collation\_server = utf8\_general\_ci  default\_storage\_engine = InnoDB  port = 3306  socket = /data/data3306/tmp/mysql3306.sock  innodb\_data\_file\_path = ibdata1:10M:autoextend  innodb\_data\_home\_dir = /data/data3306/data  innodb\_log\_file\_size = 20M  innodb\_log\_files\_in\_group = 2  innodb\_log\_group\_home\_dir = /data/data3306/log  innodb\_undo\_directory=/data/data3306/undolog  innodb\_undo\_logs=128  innodb\_undo\_tablespaces=4  innodb\_autoinc\_lock\_mode=2 |

### 3.2.9 在备库192.168.9.103上执行数据库全备份

|  |
| --- |
| nohup innobackupex --defaults-file=/dbbk/backup/backupdata.cnf --slave-info --host=192.168.9.103 --stream=xbstream --compress --extra-lsndir=/dbbk/backup/fullbackup --compress-threads=4 --throttle=500 /tmp 1>/dbbk/backup/fullbackup/fullbackup\_data\_`date +%Y%m%d\_%H%M%S`.xbstream 2>/dbbk/backup/fullbackup/fullbackup\_data\_`date +%Y%m%d\_%H%M%S`.log &  [mysql@slave fullbackup]$ tree /dbbk/backup/  /dbbk/backup/  ├── backupdata.cnf  ├── backup\_r  └── fullbackup  ├── fullbackup\_data\_20180114\_011052.log  ├── fullbackup\_data\_20180114\_011052.xbstream  └── xtrabackup\_checkpoints  2 directories, 4 files  [mysql@slave fullbackup]$ more xtrabackup\_checkpoints  backup\_type = full-backuped  from\_lsn = 0  to\_lsn = 1532193  last\_lsn = 1532193  compact = 0  recover\_binlog\_info = 0 |

### 3.2.10 在新备库192.168.9.104上恢复数据

（1）准备全量备份数据（使用redo-only参数，只应用那些已经提交的事物，而不回滚那些未提交的事务）：

|  |
| --- |
| xbstream -x < /dbbk/backup/fullbackup/fullbackup\_data\_20180114\_011052.xbstream -C /dbbk/backup/backup\_r  #查看备份文件  [root@slave2 backup\_r]# ll /dbbk/backup/backup\_r/  总用量 200  -rw-r----- 1 root root 397 1月 14 02:26 backup-my.cnf.qp  -rw-r----- 1 root root 744 1月 14 02:26 ib\_buffer\_pool.qp  -rw-r----- 1 root root 162804 1月 14 02:26 ibdata1.qp  drwxr-x--- 2 root root 4096 1月 14 02:26 mysql  drwxr-x--- 2 root root 4096 1月 14 02:26 performance\_schema  drwxr-x--- 2 root root 4096 1月 14 02:26 test  -rw-r----- 1 root root 153 1月 14 02:26 xtrabackup\_binlog\_info.qp  -rw-r----- 1 root root 113 1月 14 02:26 xtrabackup\_checkpoints  -rw-r----- 1 root root 648 1月 14 02:26 xtrabackup\_info.qp  -rw-r----- 1 root root 522 1月 14 02:26 xtrabackup\_logfile.qp  -rw-r----- 1 root root 189 1月 14 02:26 xtrabackup\_slave\_info.qp  #通过qpress命令解压\*.qp文件  for bf in `find /dbbk/backup/backup\_r -iname "\*\.qp"`; do /usr/bin/qpress -d $bf $(dirname $bf) && rm -f $bf; done |

（2）查看xtrabackup各种日志文件

|  |
| --- |
| [root@slave2 backup\_r]# ll –l /dbbk/backup/backup\_r  总用量 10280  -rw-r--r-- 1 root root 418 1月 14 02:36 backup-my.cnf  -rw-r--r-- 1 root root 935 1月 14 02:36 ib\_buffer\_pool  -rw-r--r-- 1 root root 10485760 1月 14 02:36 ibdata1  drwxr-x--- 2 root root 4096 1月 14 02:36 mysql  drwxr-x--- 2 root root 4096 1月 14 02:36 performance\_schema  drwxr-x--- 2 root root 4096 1月 14 02:36 test  -rw-r--r-- 1 root root 63 1月 14 02:36 xtrabackup\_binlog\_info  -rw-r----- 1 root root 113 1月 14 02:26 xtrabackup\_checkpoints  -rw-r--r-- 1 root root 697 1月 14 02:36 xtrabackup\_info  -rw-r--r-- 1 root root 2560 1月 14 02:36 xtrabackup\_logfile  -rw-r--r-- 1 root root 107 1月 14 02:36 xtrabackup\_slave\_info  [root@slave2 backup\_r]# more xtrabackup\_binlog\_info  mysql\_bin.000003 1609 92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-8  [root@slave2 backup\_r]#  [root@slave2 backup\_r]# more xtrabackup\_checkpoints  backup\_type = full-backuped  from\_lsn = 0  to\_lsn = 1532193  last\_lsn = 1532193  compact = 0  recover\_binlog\_info = 0  [root@slave2 backup\_r]# more xtrabackup\_info  uuid = b6bc3e24-f884-11e7-b112-000c2992078d  name =  tool\_name = innobackupex  tool\_command = --defaults-file=/dbbk/backup/backupdata.cnf --slave-info --host=192.168.9.103 --stream=xbstr  eam --compress --extra-lsndir=/dbbk/backup/fullbackup --compress-threads=4 --throttle=500 /tmp  tool\_version = 2.4.3  ibbackup\_version = 2.4.3  server\_version = 5.6.22-log  start\_time = 2018-01-14 01:10:52  end\_time = 2018-01-14 01:10:54  lock\_time = 0  binlog\_pos = filename 'mysql\_bin.000003', position '1609', GTID of the last change '92f6e6cb-f84f-11e7-ab62  -000c291dadaa:1-8'  innodb\_from\_lsn = 0  innodb\_to\_lsn = 1532193  partial = N  incremental = N  format = xbstream  compact = N  compressed = compressed  encrypted = N  [root@slave2 backup\_r]# more xtrabackup\_logfile  testxsGEN\_CLUST\_INDEXsize©  [root@slave2 backup\_r]#  [root@slave2 backup\_r]# more xtrabackup\_slave\_info  SET GLOBAL gtid\_purged='92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-8';  CHANGE MASTER TO MASTER\_AUTO\_POSITION=1  [root@slave2 backup\_r]# file xtrabackup\_logfile  xtrabackup\_logfile: data  [root@slave2 backup\_r]# file xtrabackup\_slave\_info  xtrabackup\_slave\_info: ASCII text |

（3）只应用那些已经提交的事物，而不回滚那些未提交的事务

|  |
| --- |
| innobackupex --use-memory=128M --apply-log --redo-only /dbbk/backup/backup\_r |

报错：

InnoDB: Unable to open undo tablespace './undo001'.

原因：在/dbbk/backup/backupdata.cnf文件中未指定undo相关参数，增加后重新备份恢复成功！

|  |
| --- |
| [root@slave2 backup\_r]# innobackupex --use-memory=128M --apply-log --redo-only /dbbk/backup/backup\_r  180114 03:26:32 innobackupex: Starting the apply-log operation  IMPORTANT: Please check that the apply-log run completes successfully.  At the end of a successful apply-log run innobackupex  prints "completed OK!".  innobackupex version 2.4.3 based on MySQL server 5.7.11 Linux (x86\_64) (revision id: 6a46905)  xtrabackup: cd to /dbbk/backup/backup\_r  xtrabackup: This target seems to be not prepared yet.  InnoDB: Number of pools: 1  xtrabackup: xtrabackup\_logfile detected: size=8388608, start\_lsn=(1532193)  xtrabackup: using the following InnoDB configuration for recovery:  xtrabackup: innodb\_data\_home\_dir = .  xtrabackup: innodb\_data\_file\_path = ibdata1:10M:autoextend  xtrabackup: innodb\_log\_group\_home\_dir = .  xtrabackup: innodb\_log\_files\_in\_group = 1  xtrabackup: innodb\_log\_file\_size = 8388608  xtrabackup: using the following InnoDB configuration for recovery:  xtrabackup: innodb\_data\_home\_dir = .  xtrabackup: innodb\_data\_file\_path = ibdata1:10M:autoextend  xtrabackup: innodb\_log\_group\_home\_dir = .  xtrabackup: innodb\_log\_files\_in\_group = 1  xtrabackup: innodb\_log\_file\_size = 8388608  xtrabackup: Starting InnoDB instance for recovery.  xtrabackup: Using 134217728 bytes for buffer pool (set by --use-memory parameter)  InnoDB: PUNCH HOLE support available  InnoDB: Mutexes and rw\_locks use GCC atomic builtins  InnoDB: Uses event mutexes  InnoDB: GCC builtin \_\_sync\_synchronize() is used for memory barrier  InnoDB: Compressed tables use zlib 1.2.3  InnoDB: Number of pools: 1  InnoDB: Using CPU crc32 instructions  InnoDB: Initializing buffer pool, total size = 128M, instances = 1, chunk size = 128M  InnoDB: Completed initialization of buffer pool  InnoDB: page\_cleaner coordinator priority: -20  InnoDB: Opened 4 undo tablespaces  InnoDB: 4 undo tablespaces made active  InnoDB: Highest supported file format is Barracuda.  InnoDB: The log sequence number 1527064 in the system tablespace does not match the log sequence number 1532193 in the ib\_logfiles!  InnoDB: Database was not shutdown normally!  InnoDB: Starting crash recovery.  InnoDB: Doing recovery: scanned up to log sequence number 1532193 (0%)  InnoDB: xtrabackup: Last MySQL binlog file position 1609, file name mysql\_bin.000003  InnoDB: xtrabackup: Last MySQL binlog file position 1609, file name mysql\_bin.000003  xtrabackup: starting shutdown with innodb\_fast\_shutdown = 1  InnoDB: Starting shutdown...  InnoDB: Shutdown completed; log sequence number 1532202  InnoDB: Number of pools: 1  180114 03:26:33 completed OK! |

（4）在完成了所有全量备份+增量备份的数据准备后，回滚全量备份中未提交事务确保数据一致性

|  |
| --- |
| innobackupex --use-memory=128M --apply-log /dbbk/backup/backup\_r  180114 03:41:49 innobackupex: Starting the apply-log operation  IMPORTANT: Please check that the apply-log run completes successfully.  At the end of a successful apply-log run innobackupex  prints "completed OK!".  innobackupex version 2.4.3 based on MySQL server 5.7.11 Linux (x86\_64) (revision id: 6a46905)  xtrabackup: cd to /dbbk/backup/backup\_r  xtrabackup: This target seems to be already prepared with --apply-log-only.  InnoDB: Number of pools: 1  xtrabackup: notice: xtrabackup\_logfile was already used to '--prepare'.  xtrabackup: using the following InnoDB configuration for recovery:  xtrabackup: innodb\_data\_home\_dir = .  xtrabackup: innodb\_data\_file\_path = ibdata1:10M:autoextend  xtrabackup: innodb\_log\_group\_home\_dir = .  xtrabackup: innodb\_log\_files\_in\_group = 2  xtrabackup: innodb\_log\_file\_size = 20971520  xtrabackup: using the following InnoDB configuration for recovery:  xtrabackup: innodb\_data\_home\_dir = .  xtrabackup: innodb\_data\_file\_path = ibdata1:10M:autoextend  xtrabackup: innodb\_log\_group\_home\_dir = .  xtrabackup: innodb\_log\_files\_in\_group = 2  xtrabackup: innodb\_log\_file\_size = 20971520  xtrabackup: Starting InnoDB instance for recovery.  xtrabackup: Using 134217728 bytes for buffer pool (set by --use-memory parameter)  InnoDB: PUNCH HOLE support available  InnoDB: Mutexes and rw\_locks use GCC atomic builtins  InnoDB: Uses event mutexes  InnoDB: GCC builtin \_\_sync\_synchronize() is used for memory barrier  InnoDB: Compressed tables use zlib 1.2.3  InnoDB: Number of pools: 1  InnoDB: Using CPU crc32 instructions  InnoDB: Initializing buffer pool, total size = 128M, instances = 1, chunk size = 128M  InnoDB: Completed initialization of buffer pool  InnoDB: page\_cleaner coordinator priority: -20  InnoDB: Setting log file ./ib\_logfile101 size to 20 MB  InnoDB: Setting log file ./ib\_logfile1 size to 20 MB  InnoDB: Renaming log file ./ib\_logfile101 to ./ib\_logfile0  InnoDB: New log files created, LSN=1532202  InnoDB: Opened 4 undo tablespaces  InnoDB: 4 undo tablespaces made active  InnoDB: Highest supported file format is Barracuda.  InnoDB: Log scan progressed past the checkpoint lsn 1532428  InnoDB: Doing recovery: scanned up to log sequence number 1532437 (0%)  InnoDB: Doing recovery: scanned up to log sequence number 1532437 (0%)  InnoDB: Database was not shutdown normally!  InnoDB: Starting crash recovery.  InnoDB: xtrabackup: Last MySQL binlog file position 1609, file name mysql\_bin.000003  InnoDB: Creating shared tablespace for temporary tables  InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...  InnoDB: File './ibtmp1' size is now 12 MB.  InnoDB: 96 redo rollback segment(s) found. 1 redo rollback segment(s) are active.  InnoDB: 32 non-redo rollback segment(s) are active.  InnoDB: Waiting for purge to start  InnoDB: 5.7.11 started; log sequence number 1532437  InnoDB: xtrabackup: Last MySQL binlog file position 1609, file name mysql\_bin.000003  xtrabackup: starting shutdown with innodb\_fast\_shutdown = 1  InnoDB: FTS optimize thread exiting.  InnoDB: Starting shutdown...  InnoDB: Shutdown completed; log sequence number 1532456  InnoDB: Number of pools: 1  xtrabackup: using the following InnoDB configuration for recovery:  xtrabackup: innodb\_data\_home\_dir = .  xtrabackup: innodb\_data\_file\_path = ibdata1:10M:autoextend  xtrabackup: innodb\_log\_group\_home\_dir = .  xtrabackup: innodb\_log\_files\_in\_group = 2  xtrabackup: innodb\_log\_file\_size = 20971520  InnoDB: PUNCH HOLE support available  InnoDB: Mutexes and rw\_locks use GCC atomic builtins  InnoDB: Uses event mutexes  InnoDB: GCC builtin \_\_sync\_synchronize() is used for memory barrier  InnoDB: Compressed tables use zlib 1.2.3  InnoDB: Number of pools: 1  InnoDB: Using CPU crc32 instructions  InnoDB: Initializing buffer pool, total size = 128M, instances = 1, chunk size = 128M  InnoDB: Completed initialization of buffer pool  InnoDB: page\_cleaner coordinator priority: -20  InnoDB: Opened 4 undo tablespaces  InnoDB: 4 undo tablespaces made active  InnoDB: Highest supported file format is Barracuda.  InnoDB: Removed temporary tablespace data file: "ibtmp1"  InnoDB: Creating shared tablespace for temporary tables  InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...  InnoDB: File './ibtmp1' size is now 12 MB.  InnoDB: 96 redo rollback segment(s) found. 1 redo rollback segment(s) are active.  InnoDB: 32 non-redo rollback segment(s) are active.  InnoDB: Waiting for purge to start  InnoDB: 5.7.11 started; log sequence number 1532456  xtrabackup: starting shutdown with innodb\_fast\_shutdown = 1  InnoDB: FTS optimize thread exiting.  InnoDB: Starting shutdown...  InnoDB: Shutdown completed; log sequence number 1532475  180114 03:41:55 completed OK! |

（5）回拷数据：会将恢复的数据文件，回拷到指定配置文件中的datadir目录

|  |
| --- |
| innobackupex --defaults-file=/usr/local/mysql3306/my.cnf --copy-back /dbbk/backup/backup\_r  180114 03:43:54 innobackupex: Starting the copy-back operation  IMPORTANT: Please check that the copy-back run completes successfully.  At the end of a successful copy-back run innobackupex  prints "completed OK!".  innobackupex version 2.4.3 based on MySQL server 5.7.11 Linux (x86\_64) (revision id: 6a46905)  180114 03:43:54 [01] Copying undo001 to /data/data3306/undolog/undo001  180114 03:43:54 [01] ...done  180114 03:43:56 [01] ...done  --省略中间部分---  180114 03:43:56 [01] Copying ./ib\_buffer\_pool to /data/data3306/data/ib\_buffer\_pool  180114 03:43:56 [01] ...done  180114 03:43:56 completed OK! |

（6）检查并修改文件权限

|  |
| --- |
| [root@slave2 data]# ll /data/data3306/data  总用量 22556  -rw-r----- 1 root root 935 1月 14 03:43 ib\_buffer\_pool  -rw-r----- 1 root root 10485760 1月 14 03:43 ibdata1  -rw-r----- 1 root root 12582912 1月 14 03:43 ibtmp1  drwxr-x--- 2 root root 4096 1月 14 03:43 mysql  -rw-rw---- 1 mysql mysql 0 1月 14 03:58 mysql\_bin.index  drwxr-x--- 2 root root 4096 1月 14 03:43 performance\_schema  drwxr-x--- 2 root root 4096 1月 14 03:43 test  -rw-r----- 1 root root 22 1月 14 03:43 xtrabackup\_binlog\_pos\_innodb  -rw-r----- 1 root root 697 1月 14 03:43 xtrabackup\_info  -rw-r----- 1 root root 107 1月 14 03:43 xtrabackup\_slave\_info  chown -R mysql:mysql /data/data3306/ |

（7）恢复完数据后，在新备库192.168.9.104的datadir（datadir=/data/data3306/data）下，会生成xtrabackup\_slave\_info文件，文件内容如下：

|  |
| --- |
| cd /data/data3306/data  cat xtrabackup\_slave\_info  SET GLOBAL gtid\_purged='92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-8';  CHANGE MASTER TO MASTER\_AUTO\_POSITION=1 |

（8）启动并登录192.168.9.104上的mysql数据库，根据xtrabackup\_slave\_info文件中的内容，purge掉备份中已经包含的gtid，建立主从关系。

|  |
| --- |
| cat /data/data3306/data/xtrabackup\_slave\_info  SET GLOBAL gtid\_purged='92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-8';  CHANGE MASTER TO MASTER\_AUTO\_POSITION=1 |

|  |
| --- |
| #启动MySQL  su - mysql  [mysql@slave2 dba]$ ./mysql\_startup.sh 3306  [mysql@slave2 dba]$ 180114 04:01:32 mysqld\_safe Logging to '/data/data3306/log/errlog/mysql.err'.  180114 04:01:32 mysqld\_safe Starting mysqld daemon with databases from /data/data3306/data  #查看进程  [mysql@slave2 dba]$ ps -ef | grep mysql  root 5855 1480 0 04:01 pts/0 00:00:00 su - mysql  mysql 5856 5855 0 04:01 pts/0 00:00:00 -bash  mysql 5881 1 0 04:01 pts/0 00:00:00 /bin/sh /usr/local/mysql/bin/mysqld\_safe --defaults-file=/usr/local/mysql3306/my.cnf  mysql 6980 5881 6 04:01 pts/0 00:00:00 /usr/local/mysql/bin/mysqld --defaults-file=/usr/local/mysql3306/my.cnf --basedir=/usr/local/mysql --datadir=/data/data3306/data --plugin-dir=/usr/local/mysql/lib/plugin --log-error=/data/data3306/log/errlog/mysql.err --pid-file=/data/data3306/tmp/mysql3306.pid --socket=/data/data3306/tmp/mysql3306.sock --port=3306  mysql 7003 5856 0 04:01 pts/0 00:00:00 ps -ef  mysql 7004 5856 0 04:01 pts/0 00:00:00 grep mysql  #快捷登陆MySQL  [mysql@slave2 dba]$ mysql3306  Welcome to the MySQL monitor. Commands end with ; or \g.  Your MySQL connection id is 2  Server version: 5.6.22-log Source distribution  Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.  Oracle is a registered trademark of Oracle Corporation and/or its  affiliates. Other names may be trademarks of their respective  owners.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  mysql>  mysql> show master status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  File: mysql\_bin.000001  Position: 151  Binlog\_Do\_DB:  Binlog\_Ignore\_DB:  Executed\_Gtid\_Set:  1 row in set (0.00 sec)  说明：如果“Executed\_Gtid\_Set: ”有内容，需要执行reset master清空这一列数据  mysql> reset master;  Query OK, 0 rows affected (0.01 sec)  SET GLOBAL gtid\_purged='92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-8';  change master to  master\_host='192.168.9.102',  master\_port=3306,  master\_user='myrpl',  master\_password='123456',  master\_auto\_position=1;  #启动IO进程和SQL进程  mysql> start slave;  Query OK, 0 rows affected (0.01 sec)  mysql> show slave status\G  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Slave\_IO\_State: Waiting for master to send event  Master\_Host: 192.168.9.102  Master\_User: myrpl  Master\_Port: 3306  Connect\_Retry: 60  Master\_Log\_File: mysql\_bin.000002  Read\_Master\_Log\_Pos: 2029  Relay\_Log\_File: mysql3306-relay-bin.000002  Relay\_Log\_Pos: 408  Relay\_Master\_Log\_File: mysql\_bin.000002  Slave\_IO\_Running: Yes  Slave\_SQL\_Running: Yes  Replicate\_Do\_DB:  Replicate\_Ignore\_DB:  Replicate\_Do\_Table:  Replicate\_Ignore\_Table:  Replicate\_Wild\_Do\_Table:  Replicate\_Wild\_Ignore\_Table:  Last\_Errno: 0  Last\_Error:  Skip\_Counter: 0  Exec\_Master\_Log\_Pos: 2029  Relay\_Log\_Space: 616  Until\_Condition: None  Until\_Log\_File:  Until\_Log\_Pos: 0  Master\_SSL\_Allowed: No  Master\_SSL\_CA\_File:  Master\_SSL\_CA\_Path:  Master\_SSL\_Cert:  Master\_SSL\_Cipher:  Master\_SSL\_Key:  Seconds\_Behind\_Master: 0  Master\_SSL\_Verify\_Server\_Cert: No  Last\_IO\_Errno: 0  Last\_IO\_Error:  Last\_SQL\_Errno: 0  Last\_SQL\_Error:  Replicate\_Ignore\_Server\_Ids:  Master\_Server\_Id: 91023306  Master\_UUID: 92f6e6cb-f84f-11e7-ab62-000c291dadaa  Master\_Info\_File: mysql.slave\_master\_info  SQL\_Delay: 0  SQL\_Remaining\_Delay: NULL  Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for the slave I/O thread to update it  Master\_Retry\_Count: 86400  Master\_Bind:  Last\_IO\_Error\_Timestamp:  Last\_SQL\_Error\_Timestamp:  Master\_SSL\_Crl:  Master\_SSL\_Crlpath:  Retrieved\_Gtid\_Set:  Executed\_Gtid\_Set: 92f6e6cb-f84f-11e7-ab62-000c291dadaa:1-8  Auto\_Position: 1  1 row in set (0.00 sec) |

### 3.2.11 在修改从库参数

#将read\_only参数设置为ON：

set global read\_only=1;

## 3.3 常用日志命令

（1）查看所有binlog日志列表

mysql> show master logs;

（2）查看master状态，即最后(最新)一个binlog日志的编号名称，及其最后一个操作事件pos结束点(Position)值

mysql> show master status;

（3）刷新log日志，自此刻开始产生一个新编号的binlog日志文件

mysql> flush logs;

注：每当mysqld服务重启时，会自动执行此命令，刷新binlog日志；在mysqldump备份数据时加 -F 选项也会刷新binlog日志；

（4）重置(清空)所有binlog日志

mysql> reset master;

四、MHA部署

## 4.1 MHA原理

MHA可以说是强一致的主从切换工具 ，而且切换间隔小于30s，mha主要做以下事情：

监控主库健康状况，一旦发现主库故障，便执行failover；

从宕机崩溃的Master保存二进制日志事件(binlog event)；

识别含有最新更新的Slave；

应用差异的中继日志(relaylog)到其他Slave；

应用从Master保存的二进制日志事件；

提升一个Slave为新的Master；

使其他的Slave连接新的Master进行复制。

注：slave不要延迟过长，延迟越久，切换也越久。

Mha官网链接：

https://code.google.com/p/mysql-master-ha/

管理服务器：

无论是mysql5.6还是mysql5.7统一使用ce-db-mha-1（10.120.50.11）作为管理服务器，ce-db-mha-2（10.120.50.12）作为备用管理服务器。管理服务器不但安装mha组件，还安装Mha-helper和supervisor（用于管理和保护mha进程）。

Mha-helper是一个开源的、简化mha部署的工具，内置failover自动化脚本，用以实现mha没有实现的功能，例如：设置转移vip、发起arping等。

Mha-helper官方链接如下：

https://github.com/ovaistariq/MHA-helper

mysql服务器：

各个mysql数据库服务器都安装mha组件，但不安装Mha-helper和supervisor。

## 4.2 演示环境架构及安装条件

（1）演示环境架构

|  |
| --- |
| 192.168.9.102 mysql主库（参与切换的mha候选库）  192.168.9.103 mysql从库（参与切换的mha候选库）  192.168.9.104 mysql从库  192.168.9.101 mha管理机  192.168.9.120 vip |

（2）安装的条件

* 各个mysql数据库的安装目录和数据文件目录必须统一
* mysql版本必须是5.5、5.6或5.7
* 各个服务器之间的ssh需要打通无密码ssh

## 4.3. MHA部署

### 4.3.1 免密登陆配置

各个服务器之间的ssh需要打通无密码ssh：

生成密钥文件（各个节点均需执行）

# mkdir -p ~/.ssh

# cd .ssh

# /usr/bin/ssh-keygen -t rsa （提示处直接回车即可）

# /usr/bin/ssh-keygen -t dsa （提示处直接回车即可）

执行完成后，在/root/.ssh 目录下会生产四个密钥文件。

在任意节点上做（此例是在192.168.56.122上做）：

# ssh 192.168.9.101 cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

# ssh 192.168.9.101 cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

# ssh 192.168.9.102 cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

# ssh 192.168.9.102 cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

# ssh 192.168.9.103 cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

# ssh 192.168.9.103 cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

# ssh 192.168.9.104 cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

# ssh 192.168.9.104 cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

scp ~/.ssh/authorized\_keys 192.168.9.102:.ssh/

scp ~/.ssh/authorized\_keys 192.168.9.103:.ssh/

scp ~/.ssh/authorized\_keys 192.168.9.104:.ssh/

修改authorized\_keys文件的权限为600（所有节点均需执行）

# chmod 600 ~/.ssh/authorized\_keys

### 4.3.2 安装文件说明

|  |  |
| --- | --- |
| 安装包 | 功能描述 |
| perl5.tgz | 包含perl5环境以及MHA Manager和MHA Node的文件。在各个节点都要安装。 |
| supervisor.tgz | supervisor安装文件。只需在管理节点（192.168.56.122）上部署。 |
| mha-helper.tgz | mha-helper安装文件。只需在管理节点（192.168.56.122）上部署。 |
| mha.tgz | 包换mha和mha-helper的各种配置文件。只需在管理节点（192.168.56.122）上部署。 |

### 4.3.3 在所有节点上安装perl5

mkdir -p /home/dba/mha

mkdir -p /home/software

cd /home/software

tar zxvf perl5.tgz -C /usr/local

### 4.3.4 添加环境变量：

cat >>~/.bashrc <<EOF

export PERLBREW\_ROOT=/usr/local/perl5/perlbrew

source /usr/local/perl5/perlbrew/etc/bashrc

perlbrew switch perl-5.18.4

export PATH=/usr/local/perl5/perlbrew/perls/perl-5.18.4/bin: $PATH

EOF

### 4.3.5 在管理节点上安装supervisor和mha-helper

#在192.168.9.101上安装mha

cd /home/dba

tar -zxvf /home/software/mha\_config.tgz && mv mha\_config mha

#安装supervisor

cd /home/software

tar -xzvf supervisor.tgz

cd supervisor

sh setup.sh

#安装mha-helper

cd /home/software

tar -xzvf mha-helper.tgz

cd mha-helper

sh setup.sh

出错：configure: error: no acceptable C compiler found in $PATH

解决：yum install gcc

### 4.3.6 在主库创建mha管理账户

|  |
| --- |
| #包含集群中所有mysql server的ip：  grant all privileges on \*.\* to 'mha'@'192.168.9.%' identified by '123456';  flush privileges; |

### 4.3.7 生成新增MHA集群的配置文件

cd /home/dba/mha/mha\_conf

**编辑**mha\_conf\_3306\_mf.ini**的配置文件生成脚本的配置文件：**

|  |
| --- |
| [root@manager mha\_conf]# more mha\_conf\_3306\_mf.ini  [mha\_manager]  ## ssh  ssh\_user=root  ssh\_port=22  ## mysql instance  port=3306  user=mha  password='123456'  repl\_user=myrpl  repl\_password='123456'  check\_repl\_delay=1  multi\_tier\_slave=0  ping\_interval=2  ping\_type=CONNECT  secondary\_check\_script ='masterha\_secondary\_check --port=22 -s 192.168.9.102 -s 192.168.9.103'  #LOCAL  #[server1]  server1\_hostname=192.168.9.102  server1\_ssh\_host=192.168.9.102  #server1\_candidate\_master=1  server1\_no\_master=0  #[server2]  server2\_hostname=192.168.9.103  server2\_ssh\_host=192.168.9.103  #server2\_candidate\_master=1  server2\_no\_master=0  [mha\_helper]  writer\_vip\_cidr='192.168.9.120/24'  cluster\_interface=eth0  report\_email='feima8@creditease.cn'  smtp\_host=localhost  cc\_email=zhdn\_791005@163.com  #email\_url=http://10.120.248.74/mail/send.php  kill\_after\_timeout=5  server1\_cluster\_interface=eth0  server2\_cluster\_interface=eth0 |

执行：mha\_conf.sh mha\_conf\_mf.ini testdb1app

生成mha和mha-helper的配置。

其中testdb1app为自定义的应用名称，用于区分各个mysql集群。

|  |
| --- |
| [root@manager mha\_conf]# ./mha\_conf.sh mha\_conf\_3306\_mf.ini testdbapp  "mha\_report" -> "/usr/bin/mha\_report" |

**修改生成的mha和mha-helper的配置：**

|  |
| --- |
| 在/home/dba/mha/manager/3306\_testdbapp/etc /mhamanager3306\_testdbapp.conf添加：  [root@manager etc]# more mhamanager3306\_testdbapp.conf  #APP  [server default]  ssh\_user=root  ssh\_port=22  port=3306  user=mha  password=123456  repl\_user=myrpl  repl\_password=123456  manager\_workdir=/home/dba/mha/manager/3306\_testdbapp  remote\_workdir=/home/dba/mha/node/3306\_testdbapp  master\_pid\_file=/home/mysql/mysql3306/tmp/mysql\_3306.pid  master\_binlog\_dir=/u01/mysql/mysql3306  client\_bindir=/home/mysql/mysql3306/bin  client\_libdir=/home/mysql/mysql3306/lib  manager\_log=/home/dba/mha/manager/3306\_testdbapp/logs/mhamanager3306\_testdbapp.log  check\_repl\_delay=1  multi\_tier\_slave=0  ping\_interval=2  ping\_type=CONNECT  secondary\_check\_script = masterha\_secondary\_check --port=22 -s 192.168.9.102 -s 192.168.9.103  master\_ip\_failover\_script = /usr/bin/master\_ip\_hard\_failover\_helper --test\_config\_path=/home/dba/mha/helper  /3306\_testdbapp  master\_ip\_online\_change\_script = /usr/bin/master\_ip\_online\_failover\_helper --test\_config\_path=/home/dba/mha  /helper/3306\_testdbapp  report\_script = /usr/bin/mha\_report  #shutdown\_script  #init\_conf\_load\_script  #LOCAL  [server1]  hostname=192.168.9.102  ssh\_host=192.168.9.102  #candidate\_master=1  no\_master=0  [server2]  hostname=192.168.9.103  ssh\_host=192.168.9.103  #candidate\_master=1  no\_master=0  [server3]  hostname=192.168.9.104  ssh\_host=192.168.9.104  #candidate\_master=1  no\_master=0 |

在/home/dba/mha/helper/3306\_testdbapp/mhahelper3306\_testdbapp.conf添加：

|  |
| --- |
| [root@manager 3306\_testdbapp]# more mhahelper3306\_testdbapp.conf  [default]  requires\_sudo = no  requires\_arping = yes  vip\_type = metal  writer\_vip\_cidr = 192.168.9.120/24  cluster\_interface = eth0  super\_read\_only = no  report\_email = feima8@creditease.cn  cc\_email = zhdn\_791005@163.com  email\_url =  smtp\_host = localhost  kill\_after\_timeout = 5  [192.168.9.102]  #cluster\_interface = eth0  [192.168.9.103]  #cluster\_interface = eth0  [192.168.9.104]  #cluster\_interface = eth0 |

编辑/home/dba/mha/bin/applist，添加新集群的ip，端口，业务名称：

|  |
| --- |
| [root@node3 /]# cat /home/dba/mha/bin/applist  port=3306;app=testdbapp |

**手动挂载vip:**

在mysql主库上挂载/home/dba/mha/manager/${port}\_${appname}/配置文件.cnf中配置的vip?

这个文件中没有配置VIP

ip addr add ip/mask dev 网卡

具体执行如下：

在192.168.9.102：

ip addr add 192.168.9.120/24 dev eth0

注：如果以后重启了服务器，需要重新执行上述命令。

## 4.4. MHA管理命令简介

与mha有关的管理命令都封装在mha管理服务器的/home/dba/mha/bin目录的各个shell脚本中。

check\_mha.sh 检查mha是否正常运行。

check\_rpl.sh 检查mha集群环境是否正常。

check\_ssh.sh 检查mha管理服务器与各个数据库服务器的ssh等价（是否可以实现无密码互访）

tail\_log.sh 查看mha日志。

switch\_over.sh 用于进行switch over。

命令使用举例：

./ check\_mha.sh 端口号 应用名

其中端口号和应用名，可在同级目录的applist文件中获取。

## 4.5. MHA测试

进入目录/home/dba/mha/bin

cd /home/dba/mha/bin

### 4.5.1 检查ssh连通性

./check\_ssh.sh 3306 testdbapp

|  |
| --- |
| [root@manager bin]# ./check\_ssh.sh 3306 testdbapp  Sun Jan 14 19:02:17 2018 - [warning] Global configuration file /etc/masterha\_default.cnf not found. Skipping.  Sun Jan 14 19:02:17 2018 - [info] Reading application default configuration from /home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf..  Sun Jan 14 19:02:17 2018 - [info] Reading server configuration from /home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf..  Sun Jan 14 19:02:17 2018 - [info] Starting SSH connection tests..  Sun Jan 14 19:02:18 2018 - [debug]  Sun Jan 14 19:02:17 2018 - [debug] Connecting via SSH from root@192.168.9.102(192.168.9.102:22) to root@192.168.9.103(192.168.9.103:22)..  Sun Jan 14 19:02:18 2018 - [debug] ok.  Sun Jan 14 19:02:18 2018 - [debug] Connecting via SSH from root@192.168.9.102(192.168.9.102:22) to root@192.168.9.104(192.168.9.104:22)..  Sun Jan 14 19:02:18 2018 - [debug] ok.  Sun Jan 14 19:02:19 2018 - [debug]  Sun Jan 14 19:02:18 2018 - [debug] Connecting via SSH from root@192.168.9.103(192.168.9.103:22) to root@192.168.9.102(192.168.9.102:22)..  Sun Jan 14 19:02:18 2018 - [debug] ok.  Sun Jan 14 19:02:18 2018 - [debug] Connecting via SSH from root@192.168.9.103(192.168.9.103:22) to root@192.168.9.104(192.168.9.104:22)..  Sun Jan 14 19:02:18 2018 - [debug] ok.  Sun Jan 14 19:02:20 2018 - [debug]  Sun Jan 14 19:02:18 2018 - [debug] Connecting via SSH from root@192.168.9.104(192.168.9.104:22) to root@192.168.9.102(192.168.9.102:22)..  Sun Jan 14 19:02:19 2018 - [debug] ok.  Sun Jan 14 19:02:19 2018 - [debug] Connecting via SSH from root@192.168.9.104(192.168.9.104:22) to root@192.168.9.103(192.168.9.103:22)..  Sun Jan 14 19:02:19 2018 - [debug] ok.  Sun Jan 14 19:02:20 2018 - [info] All SSH connection tests passed successfully. |

### 4.5.2 检查主从同步

./check\_rpl.sh 3306 testdbapp

第一次运行报错是因为没有将

master\_ip\_hard\_failover\_helper，master\_ip\_online\_failover\_helper脚本复制到/usr/bin下，拷贝到/usr/bin下后，执行chmod 775 master\_ip\*，再次运行通过！

|  |
| --- |
| [root@manager bin]# ./check\_rpl.sh 3306 testdbapp  Sun Jan 14 19:16:26 2018 - [warning] Global configuration file /etc/masterha\_default.cnf not found. Skipping.  Sun Jan 14 19:16:26 2018 - [info] Reading application default configuration from /home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf..  Sun Jan 14 19:16:26 2018 - [info] Reading server configuration from /home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf..  Sun Jan 14 19:16:26 2018 - [info] MHA::MasterMonitor version 0.57.  Sun Jan 14 19:16:27 2018 - [info] GTID failover mode = 1  Sun Jan 14 19:16:27 2018 - [info] Dead Servers:  Sun Jan 14 19:16:27 2018 - [info] Alive Servers:  Sun Jan 14 19:16:27 2018 - [info] 192.168.9.102(192.168.9.102:3306)  Sun Jan 14 19:16:27 2018 - [info] 192.168.9.103(192.168.9.103:3306)  Sun Jan 14 19:16:27 2018 - [info] 192.168.9.104(192.168.9.104:3306)  Sun Jan 14 19:16:27 2018 - [info] Alive Slaves:  Sun Jan 14 19:16:27 2018 - [info] 192.168.9.103(192.168.9.103:3306) Version=5.6.22-log (oldest major version between slaves) log-bin:enabled  Sun Jan 14 19:16:27 2018 - [info] GTID ON  Sun Jan 14 19:16:27 2018 - [info] Replicating from 192.168.9.102(192.168.9.102:3306)  Sun Jan 14 19:16:27 2018 - [info] 192.168.9.104(192.168.9.104:3306) Version=5.6.22-log (oldest major version between slaves) log-bin:enabled  Sun Jan 14 19:16:27 2018 - [info] GTID ON  Sun Jan 14 19:16:27 2018 - [info] Replicating from 192.168.9.102(192.168.9.102:3306)  Sun Jan 14 19:16:27 2018 - [info] Current Alive Master: 192.168.9.102(192.168.9.102:3306)  Sun Jan 14 19:16:27 2018 - [info] Checking slave configurations..  Sun Jan 14 19:16:27 2018 - [info] Checking replication filtering settings..  Sun Jan 14 19:16:27 2018 - [info] binlog\_do\_db= , binlog\_ignore\_db=  Sun Jan 14 19:16:27 2018 - [info] Replication filtering check ok.  Sun Jan 14 19:16:27 2018 - [info] GTID (with auto-pos) is supported. Skipping all SSH and Node package checking.  Sun Jan 14 19:16:27 2018 - [info] Checking SSH publickey authentication settings on the current master..  Sun Jan 14 19:16:27 2018 - [info] HealthCheck: SSH to 192.168.9.102 is reachable.  Sun Jan 14 19:16:27 2018 - [info]  192.168.9.102(192.168.9.102:3306) (current master)  +--192.168.9.103(192.168.9.103:3306)  +--192.168.9.104(192.168.9.104:3306)  Sun Jan 14 19:16:27 2018 - [info] Checking replication health on 192.168.9.103..  Sun Jan 14 19:16:27 2018 - [info] ok.  Sun Jan 14 19:16:27 2018 - [info] Checking replication health on 192.168.9.104..  Sun Jan 14 19:16:27 2018 - [info] ok.  Sun Jan 14 19:16:27 2018 - [info] Checking master\_ip\_failover\_script status:  Sun Jan 14 19:16:27 2018 - [info] /usr/bin/master\_ip\_hard\_failover\_helper --test\_config\_path=/home/dba/mha/helper/3306\_testdbapp --command=status --ssh\_user=root --orig\_master\_host=192.168.9.102 --orig\_master\_ip=192.168.9.102 --orig\_master\_port=3306  Reading config file: /home/dba/mha/helper/3306\_testdbapp/mhahelper3306\_testdbapp.conf  Checking the vip using the 'metal' provider on the original master '192.168.9.102'  Connecting to 'root'@'192.168.9.102'  Executing command on '192.168.9.102': /sbin/ip addr show dev eth0  Sun Jan 14 19:16:28 2018 - [info] OK.  Sun Jan 14 19:16:28 2018 - [warning] shutdown\_script is not defined.  Sun Jan 14 19:16:28 2018 - [info] Got exit code 0 (Not master dead).  MySQL Replication Health is OK. |

### 4.5.3 启动MHA

|  |
| --- |
| cd /home/dba/mha/bin  ./start\_mha.sh 3306 testdbapp &  [1] 65441  [root@manager bin]# Sun Jan 14 19:27:21 2018 - [warning] Global configuration file /etc/masterha\_default.cnf not found. Skipping.  Sun Jan 14 19:27:21 2018 - [info] Reading application default configuration from /home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf..  Sun Jan 14 19:27:21 2018 - [info] Reading server configuration from /home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf..  [root@manager bin]# ps -ef |grep perl  root 65464 65441 0 19:27 pts/1 00:00:00 perl /usr/local/perl5/perlbrew/perls/perl-5.18.4/bin/masterha\_manager --conf=/home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf |

### 4.5.4 检查 MHA状态

./check\_mha.sh 3306 testdbapp

|  |
| --- |
| cd /home/dba/mha/bin  [root@manager bin]# ./check\_mha.sh 3306 testdbapp  mhamanager3306\_testdbapp (pid:65464) is running(0:PING\_OK), master:192.168.9.102 |

### 4.5.5 查看MHA日志

./tail\_log.sh 3306 testdbapp

|  |
| --- |
| [root@manager bin]# ./tail\_log.sh 3306 testdbapp  Sun Jan 14 19:27:21 2018 - [info] MHA::MasterMonitor version 0.57.  Sun Jan 14 19:27:22 2018 - [info] GTID failover mode = 1  Sun Jan 14 19:27:22 2018 - [info] Dead Servers:  Sun Jan 14 19:27:22 2018 - [info] Alive Servers:  Sun Jan 14 19:27:22 2018 - [info] 192.168.9.102(192.168.9.102:3306)  Sun Jan 14 19:27:22 2018 - [info] 192.168.9.103(192.168.9.103:3306)  Sun Jan 14 19:27:22 2018 - [info] 192.168.9.104(192.168.9.104:3306)  Sun Jan 14 19:27:22 2018 - [info] Alive Slaves:  Sun Jan 14 19:27:22 2018 - [info] 192.168.9.103(192.168.9.103:3306) Version=5.6.22-log (oldest major version between slaves) log-bin:enabled  Sun Jan 14 19:27:22 2018 - [info] GTID ON  Sun Jan 14 19:27:22 2018 - [info] Replicating from 192.168.9.102(192.168.9.102:3306)  Sun Jan 14 19:27:22 2018 - [info] 192.168.9.104(192.168.9.104:3306) Version=5.6.22-log (oldest major version between slaves) log-bin:enabled  Sun Jan 14 19:27:22 2018 - [info] GTID ON  Sun Jan 14 19:27:22 2018 - [info] Replicating from 192.168.9.102(192.168.9.102:3306)  Sun Jan 14 19:27:22 2018 - [info] Current Alive Master: 192.168.9.102(192.168.9.102:3306)  Sun Jan 14 19:27:22 2018 - [info] Checking slave configurations..  Sun Jan 14 19:27:22 2018 - [info] Checking replication filtering settings..  Sun Jan 14 19:27:22 2018 - [info] binlog\_do\_db= , binlog\_ignore\_db=  Sun Jan 14 19:27:22 2018 - [info] Replication filtering check ok.  Sun Jan 14 19:27:22 2018 - [info] GTID (with auto-pos) is supported. Skipping all SSH and Node package checking.  Sun Jan 14 19:27:22 2018 - [info] Checking SSH publickey authentication settings on the current master..  Sun Jan 14 19:27:22 2018 - [info] HealthCheck: SSH to 192.168.9.102 is reachable.  Sun Jan 14 19:27:22 2018 - [info]  192.168.9.102(192.168.9.102:3306) (current master)  +--192.168.9.103(192.168.9.103:3306)  +--192.168.9.104(192.168.9.104:3306)  Sun Jan 14 19:27:22 2018 - [info] Checking master\_ip\_failover\_script status:  Sun Jan 14 19:27:22 2018 - [info] /usr/bin/master\_ip\_hard\_failover\_helper --test\_config\_path=/home/dba/mha/helper/3306\_testdbapp --command=status --ssh\_user=root --orig\_master\_host=192.168.9.102 --orig\_master\_ip=192.168.9.102 --orig\_master\_port=3306  Reading config file: /home/dba/mha/helper/3306\_testdbapp/mhahelper3306\_testdbapp.conf  Checking the vip using the 'metal' provider on the original master '192.168.9.102'  Connecting to 'root'@'192.168.9.102'  Executing command on '192.168.9.102': /sbin/ip addr show dev eth0  Sun Jan 14 19:27:22 2018 - [info] OK.  Sun Jan 14 19:27:22 2018 - [warning] shutdown\_script is not defined.  Sun Jan 14 19:27:22 2018 - [info] Set master ping interval 2 seconds.  Sun Jan 14 19:27:22 2018 - [info] Set secondary check script: masterha\_secondary\_check --port=22 -s 192.168.9.102 -s 192.168.9.103  Sun Jan 14 19:27:22 2018 - [info] Starting ping health check on 192.168.9.102(192.168.9.102:3306)..  Sun Jan 14 19:27:22 2018 - [info] Ping(CONNECT) succeeded, waiting until MySQL doesn't respond.. |

## 4.6. 通过supervisor管理mha

supervisor是用Python开发的一套通用的进程管理程序，能将一个普通的命令行进程变为后台daemon，并监控进程状态，异常退出时能自动重启。

生产环境的mha进程就是由supervisor管理的。

当一个新的mha集群建立后，需要进入/home/supervisor/etc下，新建新集群的.ini文件，格式为：mha\_应用名\_端口号.ini

|  |
| --- |
| #vi mha\_testdbapp\_3306.ini  [program:mha\_testdbapp\_3306]  command=masterha\_manager --conf=/home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf ; the program (relative uses PATH, can take args)  process\_name=%(program\_name)s ; process\_name expr (default %(program\_name)s)  numprocs=1 ; number of processes copies to start (def 1)  directory=/home/dba/mha/bin ; directory to cwd to before exec (def no cwd)  umask=022 ; umask for process (default None)  priority=999 ; the relative start priority (default 999)  autostart=false ; start at supervisord start (default: true)  autorestart=true ; whether/when to restart (default: unexpected)  startsecs=5 ; number of secs prog must stay running (def. 1)  startretries=1 ; max # of serial start failures (default 3)  exitcodes=0,2 ; 'expected' exit codes for process (default 0,2)  stopsignal=QUIT ; signal used to kill process (default TERM)  stopwaitsecs=10 ; max num secs to wait b4 SIGKILL (default 10)  stopasgroup=false ; send stop signal to the UNIX process group (default false)  killasgroup=false ; SIGKILL the UNIX process group (def false)  ;user=chrism ; setuid to this UNIX account to run the program  redirect\_stderr=true ; redirect proc stderr to stdout (default false)  stdout\_logfile=/home/dba/mha/manager/3306\_testdbapp/logs/mhamanager3306\_testdbapp.log ; stdout log path, NONE for none; default AUTO  stdout\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)  stdout\_logfile\_backups=10 ; # of stdout logfile backups (default 10)  stdout\_capture\_maxbytes=1MB ; number of bytes in 'capturemode' (default 0)  stdout\_events\_enabled=false ; emit events on stdout writes (default false)  ;stderr\_logfile=/a/path ; stderr log path, NONE for none; default AUTO  ;stderr\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)  ;stderr\_logfile\_backups=10 ; # of stderr logfile backups (default 10)  ;stderr\_capture\_maxbytes=1MB ; number of bytes in 'capturemode' (default 0)  ;stderr\_events\_enabled=false ; emit events on stderr writes (default false)  ;environment=A="1",B="2" ; process environment additions (def no adds)  serverurl=AUTO ; override serverurl computation (childutils) |

之后，还要把新的mha服务加入到group里（通过修改group\_mha.ini文件）:

|  |
| --- |
| group\_mha.ini  # cat group\_mha.ini  [group:mha]  programs=mha\_testdb1app\_3501 ; each refers to 'x' in [program:x] definitions  priority=999 ; the relative start priority (default 999)  [root@node3 etc]# |

### 4.6.1 启动supervisord服务

|  |
| --- |
| supervisord -c /etc/supervisord.conf  [root@manager tmp]# ps -ef | grep supervisor  root 67643 1 0 20:23 ? 00:00:00 /usr/local/bin/python /usr/local/bin/supervisord -c /etc/supervisord.conf |

### 4.6.2 查看supervisord帮助

|  |
| --- |
| /usr/local/bin/supervisorctl –h |

### 4.6.3 修改配置后使配置生效

|  |
| --- |
| [root@manager ~]# supervisorctl update |

运行此命令后，同组的supervisorctl服务会关闭。需要把它们逐一启动。

### 4.6.4 启动某个mha进程

|  |
| --- |
| [root@manager tmp]# supervisorctl start mha:mha\_testdbapp\_3306  mha:mha\_testdbapp\_3306: started  [root@manager tmp]# ps -ef | grep perl  root 68341 67643 0 20:44 ? 00:00:00 perl /usr/local/perl5/perlbrew/perls/perl-5.18.4/bin/masterha\_manager --conf=/home/dba/mha/manager/3306\_testdbapp/etc/mhamanager3306\_testdbapp.conf |

其格式为：supervisorctl start [group名].[program名]

group名和program名在上面提到的group\_mha.ini文件中定义。

其格式为：supervisorctl start [group名].[program名]

group名和program名在上面提到的group\_mha.ini文件中定义。

### 4.6.5 查看supervisor状态

|  |
| --- |
| [root@manager tmp]# supervisorctl status  mha:mha\_testdbapp\_3306 RUNNING pid 68341, uptime 0:01:24 |

### 4.6.6 停止mha进程

|  |
| --- |
| [root@manager tmp]# supervisorctl stop mha:mha\_testdbapp\_3306  mha:mha\_testdbapp\_3306: stopped  其格式为：supervisorctl start [group名].[program名]  group名和program名在上面提到的group\_mha.ini文件中定义。 |

## 4.7. 配置与管理注意事项

## 4.7.1 mha的管理用户

mha的管理用户是root而不是yxgly，mha管理服务器到各个数据库服务器的ssh用户等价也是基于root用户。因此，在mha管理服务器上执行各种mha管理命令都必须使用root用户，而不能使用yxgly用户，否则会报错。

## 4.7.2 arping的配置

MHA默认没有arping，这个要自己加上，否则服务器会自动等到vip缓存失效，期间VIP会有一定时间的不可用。以端口为3306的mysql服务为例，需要在mha-helper的配置文件（/home/dba/mha/helper/3306\_testdbapp/mhahelper3306\_testdbapp.conf）中加入下面参数：

requires\_arping= yes

|  |
| --- |
| [root@manager helper]# vi /home/dba/mha/helper/3306\_testdbapp/mhahelper3306\_testdbapp.conf  [default]  requires\_sudo = no  requires\_arping = yes  vip\_type = metal  writer\_vip\_cidr = 192.168.9.120/24  cluster\_interface = eth0  super\_read\_only = no  report\_email = feima8@creditease.cn  cc\_email = zhdn\_791005@163.com  email\_url =  smtp\_host = localhost  kill\_after\_timeout = 5  [192.168.9.102]  #cluster\_interface = eth0  [192.168.9.103]  #cluster\_interface = eth0  [192.168.9.104]  #cluster\_interface = eth0 |

### 4.7.3 mha的启停

当前所有mha管理服务器上的mha进程都是由supervisor管理的，请使用supervisorctl命令进行mha服务的启停，而不要用mha自己的命令。Mha管理服务器的nagios相关监控也会检查supervisor相关服务是否正常。

### 4.7.4 需要屏蔽的安全加固项

经过安全加固的服务器，会阻止root用户的远程登录，但mha管理服务器与各个数据库服务器需要基于root用户实现ssh的无密码互访。因此，必须屏蔽这一安全加固项，具体方法是：

在 /etc/ssh/sshd\_config中将PermitRootLogin（可能等于yes或no）注释掉。

重启ssh服务：

service sshd restart

### 4.7.5 当mha failover之后

已3306端口为例：

在mha做完failover后，必须删除/home/dba/mha/manager/3306\_testdbapp下的failover文件，否则下次mha无法正常启动：

rm mhamanager3306\_testdbapp.failover.complete

### 4.7.6 修改supervisor的startsecs参数

还是以3306端口为例，需要将/home/supervisor/etc下的mha\_testdbapp\_3306.ini文件中startsecs参数的值由默认的1秒改为5秒。

startsecs参数是mha管理的进程启动多少秒之后，此时状态如果是running，则我们认为启动成功了。

如果startsecs参数设置得太低，在mha做failover时，会导致mha进程被反复启动。

# 五、需要配置的权限

## 5.1. Dbant平台授权

create user puppet@'10.130.249.32' identified by "g4POngCf8lLH6iR9pyJ3";

grant ALL PRIVILEGES on \*.\* to puppet@'10.130.249.32';

update mysql.user set show\_view\_priv = 'Y' where user = 'puppet' and Host = '10.130.249.32';

flush privileges;

update mysql.user set Trigger\_priv = 'Y' where user = 'puppet' and Host = '10.130.249.32';

flush privileges;

六、附件：

|  |  |
| --- | --- |
| 附件名称 | 附件文件 |
| /usr/local/mysql3306/my.cnf |  |
| /home/mysql/dba |  |
| /home/mysql/.bash\_profile |  |
| /home/software/mha-helper |  |